REPUBLIKA E SHQIPËRISË
UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I TEKNOLOGJISË SË INFORMACIONIT
DEPARTAMENTI I INXHINIERISË INFORMATIKE

DISERTACION

AGJENTË INTELIGJENTË NË SISTEMET

PYETJE-PËRGJIGJE VIZUALE

DHE DIALOGUN VIZUAL

nga

LORENA KODRA

PËR MARRJEN E GRADËS

"DOKTOR"

NË: "TEKNOLOGJINË E INFORMACIONIT DHE KOMUNIKIMIT"

Udhëheqës shkencor

PROF.ASOC. ELINDA KAJO MEÇE

Tiranë, 2018

AGJENTË INTELIGJENTË NË SISTEMET

PYETJE-PËRGJIGJE VIZUALE

DHE DIALOGUN VIZUAL

Disertacion

i paraqitur në Universitetin Politeknik të Tiranës për marrjen e gradës

"Doktor"

në:

"Teknologjinë e Informacionit dhe Komunikimit"

Nga

Znj. Lorena Kodra

2018

JURIA PËR VLERËSIMIN E DISERTACIONIT PËR FITIMIN E GRADËS SHKENCORE "DOKTOR"

Miratuar

me vendimin e Këshillit të Profesorëve të FTI-së Nr._____, datë_____:

Kryetar i Jurisë_____

Anëtar i Jurisë_____

Anëtar i Jurisë_____

Anëtar i Jurisë_____

Anëtar i Jurisë_____

Dekan i Fakultetit të Teknologjisë së Informacionit

_____

Mirënjohje

Mirënjohja për realizimin e këtij punimi shkon së pari për udhëheqësen time Prof.Asoc. Elinda Kajo Meçe, e cila me durimin dhe nxitjen e saj ka qenë mbështetja e parë për realizimin e studimit tim. I jam mirënjohëse për çdo diskutim, shkëmbim mendimesh, kritike dhe vlerësim, por mbi të gjitha për lirinë që më ka dhënë për të hulumtuar në machine learning dhe për të realizuar këtë punim.

Falenderoj stafin e Departamentit të Inxhinierisë Informatike për sugjerimet dhe orientimet e tyre të vlefshme gjatë dy prezantimeve të mia në Departament.

Së fundmi, i jam mirënjohëse familjes sime dhe të gjithë njerëzve të mi të dashur të cilët kanë dhënë mbështetjen dhe kontributin e tyre për realizimin e këtij punimi.

Faleminderit!

Lorena Kodra

Janar 2018, Tiranë, Shqipëri

Përmbajtja

Mirënjohje ..... **v**

Lista e Figurave ..... **ix**

Lista e Tabelave ..... **xi**

Abstrakt ..... **xii**

Abstract **1**

1 Hyrje **1**

1.1 Problemi 1

1.2 Objektivi i Këtij Punimi dhe Motivimi 3

1.3 Kontributet e Këtij Punimi 4

1.4 Struktura e Disertacionit 5

2 Sistemet Pyetje-Përgjigje dhe Dialogu Vizual **7**

2.1 Sistemet Pyetje-Përgjigje 7

2.2 Qasjet në Implementimin e Sistemeve Pyetje-Përgjigje 11

2.2.1 Qasja Gjuhësore 11

2.2.2 Qasja Statistikore 12

2.2.3 Qasja e Përputhjes së Modelit 12

2.2.4 Qasja Hibride 12

2.3 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje 13

2.3.1 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje me Bazë Njohurish 14

2.3.2 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje Komunitare 16

2.4 Tendencat e Kërkimit Shkencor në Sistemet Pyetje-Përgjigje 17

2.5 Sistemet Pyetje-Përgjigje Vizuale 18

2.5.1 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje Vizuale 19

2.6 Dialogu Vizual 20

2.6.1 Vlerësimi i Modeleve të Dialogut Vizual 21

3 Bazat Teorike të Deep Learning **24**

3.1 Machine Learning 24
3.1.1 Supervised Learning 25
3.1.2 Overfitting dhe Underfitting 27
3.1.3 Teorema “No free lunch” 29
3.1.4 Rregullarizimi 29
3.2 Optimizimi 30
3.2.1 Optimizimi Stokastik Sipas Gradientit 33
3.3 Algoritmi backpropagation 33
3.4 Rrjetat Neurale Artificiale 35
3.4.1 Neuroni 35
3.4.2 Feedforward Neural Network 37
3.4.3 Rrjetat Neurale Konvolucionale 38
3.4.4 Rrjetat Neurale Rekurrente 45
3.4.5 Vëmendja Neurale 50
3.4.6 Dropout 51

4 Punime të Ngjashme 53

4.1 Sistemet Pyetje-Përgjigje Vizuale 53
4.2 Vëmendja Neurale 55
4.3 Dialogu Vizual 56

5 Agjent Inteligjent në Sistemet Pyetje-Përgjigje Vizuale 58

5.1 Modeli i Agjentit 59
5.1.1 Paraqitja e Fjalëve 60
5.1.2 Paraqitja e Imazheve 61
5.1.3 Vëmendja Multimodale 61
5.2 Optimizimi, Detajet e Implementimit dhe Hiperparametrat e Modelit 64
5.3 Ambienti i Punës 65
5.4 Dataset-et dhe Metrikat e Përdorura 67
5.4.1 Dataset-i VQA 67
5.4.2 Dataset-i Visual7W 68
5.5 Rezultatet e Vlerësimit për Dataset-in VQA 69
5.5.1 Vlerësimi Sasior 70
5.5.2 Vlerësimi Cilësor 72

5.6 Rezultatet e Vlerësimit për Dataset-in Visual7W 77
5.6.1 Vlerësimi Sasior 77
5.6.2 Vlerësimi Cilësor 78
5.7 Diskutime dhe konkluzione 81

6 Agjent Inteligjent në Dialogun Vizual 83

6.1 Modeli i Agjentit 84
6.1.1 Paraqitja e Fjalëve 85
6.1.2 Paraqitja e Imazheve 86
6.1.3 Vëmendja Multimodale 86
6.2 Optimizimi, Detajet e Implementimit dhe Hiperparametrat e Modelit 88
6.3 Ambienti i Punës 89
6.3.1 Konfigurimi Hardware 89
6.3.2 Konfigurimi Software 90
6.4 Dataset-i dhe Metrikat e Përdorura 91
6.4.1 Dataset-i VisDial 91
6.4.2 Metrikat e Përdorura 92
6.5 Rezultatet e Vlerësimit Sasior 92
6.6 Rezultatet e Vlerësimit Cilësor 93
6.6.1 Vlerësimi Cilësor i Gabimeve 96
6.7 Diskutime dhe konkluzione 98

7 Konkluzione 100

7.1 Puna në të ardhmen 102

Referenca 104

Shtojca 1: Fjalor i Termave Teknikë 111

Shtojca 2: Lista e Botimeve 120

Lista e Figurave

Figura 1.1. Sfidat në njohjen dhe arsyetimin rreth imazheve 2

Figura 2.1. Përpunimi i informacionit në një sistem pyetje-përgjigje. 9

Figura 3.1.Underfitting, overfitting dhe gjeneralizimi i mirë. 28

Figura 3.2. Gradient descent 31

Figura 3.3. Gradient descent në rastet e konvergjimit të vonuar dhe divergjimit.....32

Figura 3.4. Grafi llogaritës i funksionit $y = (5x + 1)^2$ 34

Figura 3.5. Bakpropagation për funksionin $y = (5x + 1)^2$.....35

Figura 3.6. Modeli logjik i një neuroni.....36

Figura 3.7. Funksione të ndryshme aktivizimi 37

Figura 3.8. Shembull i një rrjeti neural feedforward 37

Figura 3.9. Konvolucioni sipas një filtri 5 x 5 x 3.....41

Figura 3.10. Filtrat e mësuar nga një rrjet CNN.....42

Figura 3.11. Padding. 43

Figura 3.12. Stride.....43

Figura 3.13. Pooling sipas vlerës maksimale për një filtër me përmasa 2 x 2, me hap 2. 44

Figura 3.14. Arkitektura e një rrjeti RNN. 46

Figura 3.15. Vëmendja vizuale në kohë.....51

Figura 3.16. Rrjet neural me dropout.....52

Figura 5.1: Dataflow e qelizave LSTM brenda në rrjetin e koduesit LSTM.....60

Figura 5.2: Gjenerimi i vëmendjes.....63

Figura 6.1: Dataflow i modelit.....85

Lista e Tabelave

Tabela 2.1. Krahasimi midis qasjeve në implementimin e sistemeve pyetje-përgjigje.....13

Tabela 5.1: Rezultatet e testimit për dataset-in VQA për përgjigjet e lira krahasuar me state of the art.....70

Tabela 5.2: Rezultatet e testimit për dataset-in VQA për përgjigjet me alternativa krahasuar me state of the art.....71

Tabela 5.3: Shembuj të përgjigjeve për pyetjet *Po/Jo*. 72

Tabela 5.4: Shembuj të përgjigjeve për pyetjet e numërimit të objekteve.....74

Tabela 5.5: Shembuj të përgjigjeve për pyetjet e kategorisë *Tjetër*.....75

Tabela 5.6: Rezultatet e testimit për dataset-in Visual7W krahasuar me state of the art. 77

Tabela 5.7: Shembuj të përgjigjeve të gjeneruara nga agjenti për tipe të ndryshme pyetjesh për dataset-in Visual7W.....79

Tabela 6.1: Performanca e modeleve në dataset-in VisDial v0.9 e matur me *mean reciprocal rank* (MRR) dhe *recall @ k*.....92

Tabela 6.2: Shembuj bisedash për dataset-in Visdial.....94

Tabela 6.3: Shembuj bisedash me përgjigje të gabuara për dataset-in Visdial. 96

Abstrakt

Progresi i madh në vizionin kompjuterik (*computer vision*) dhe në teknikat e përpunimit të gjuhës natyrore (d.m.th. gjuhës njerëzore) ka bërë të mundur avancimin gjithmonë e më shumë të inteligjencës artificiale duke kaluar nga sisteme të thjeshta si klasifikimi i imazheve në sisteme inteligjente të cilat janë të afta t'i përgjigjen pyetjeve rreth një imazhi apo një videoje.

Hapi i ardhshëm, gjithashtu dhe aspirata më e madhe e inteligjencës artificiale është krijimi i sistemeve të cilat të jenë të afta jo vetëm të "shohin" dhe "kuptojnë" imazhin, por edhe të zhvillojnë një dialog kuptimplotë në gjuhën natyrore me njerëzit rreth një konteksti vizual.

Në këtë disertacion paraqitet implementimi dhe testimi i dy modeleve risi të agjentëve inteligjentë të cilët i përgjigjen pyetjeve në gjuhën natyrore rreth një imazhi. Keta agjentë janë implementuar me *rrjeta neurale artificiale* dhe përdorin teknika të *computer vision* dhe *përpunimit të gjuhës natyrore*.

Fillimisht paraqitet modeli i një agjenti inteligjent për sistemet pyetje-përgjigje vizuale i cili është i aftë të përqëndrojë vëmendjen në fjalë të veçanta të pyetjes dhe zona të veçanta të imazhit në mënyrë që të rritë saktësinë e përgjigjes së gjeneruar. Ideja e përdorimit të vëmendjes së dyfishtë qëndron në faktin se, përveçse të vendosë në cilat zona të imazhit të fokusohet më shumë, agjenti vendos gjithashtu dhe cilave fjalë të pyetjes t'i kushtojë më shumë vëmendje. Mekanizmi i vëmendjes implementohet me anë të një arkitekture risi të rrjetave neurale artificiale.

Më tej paraqitet modeli i një agjenti të dialogut vizual. Agjentët kanë të përbashkët faktin që të dy i përgjigjen pyetjeve rreth një imazhi, por agjenti i dialogut

vizual qëndron në një nivel më të lartë abstraksioni. Ai është i aftë, jo vetëm t'i përgjigjet pyetjeve në gjuhë natyrore rreth një imazhi, por gjithashtu të ruajë dhe kuptojë kontekstin e bisedës (d.m.th. pyetjeve dhe përgjigjeve të mëparshme rreth të njëjtit imazh) dhe t'i përgjigjet pyetjeve vijuese të cilat kanë lidhje me kontekstin e pyetjeve dhe përgjigjeve të mëparshme. Gjatë bashkëveprimit, operatori njerëzor mund të përdorë përemra referues për objektet në imazh dhe agjenti është i aftë të kuptojë se cilit objekt i përket kjo referencë. Edhe ky agjent është i implementuar me rrjeta neurale artificiale dhe përdor vëmendjen e dyfishtë duke u fokusuar njëkohësisht në historikun e bisedës dhe zona të caktuara të imazhit. Mekanizmi i vëmendjes së dyfishtë i përdorur nga ky agjent është një risi për dialogun vizual pasi ky mekanizëm nuk është eksploruar më parë për këto sisteme.

Agjentët janë testuar duke përdorur *dataset*-e (d.m.th. bashkësi të dhënash trajnimi dhe testimi) të mirënjohura publike dhe rezultatet e testimeve analizohen në mënyrë sasiore duke u krahasuar me modele të ngjashme *state of the art*. Rezultatet e testimeve tregojnë se modelet e propozuara të agjentëve janë të suksesshme dhe përmirësojnë *state of the art* për metrikat e testimit. Përveç analizës sasiore, rezultatet e testimeve vlerësohen edhe në mënyrë cilësore dhe bëhet një diskutim për aftësitë dhe limitimet e agjentëve. Rezultatet e kësaj analize tregojnë se arkitekturat risi të propozuara janë të suksesshme dhe ndihmojnë agjentët të përmirësojnë saktësinë e përgjigjeve. Analiza cilësore tregon gjithashtu edhe limitimet e agjentëve.

Fjalë Kyçe—Sistemet pyetje-përgjigje vizuale, dialogu vizual, agjentë inteligjentë, vëmendja multimodale, deep learning, rrjeta neurale konvolucionale, rrjeta neurale rekurrente

Abstract

The great advances in computer vision and natural language (i.e. human language) processing techniques have made possible the advancement of artificial intelligence from simple systems such as image classification to intelligent systems which are capable of answering questions about an image or a video.

The next step, as well as the greatest aspiration of artificial intelligence, is to create systems that not only can "see" and "understand" the image, but also are able to sustain a meaningful conversation in natural language with humans about a visual context.

This dissertation presents the implementation and testing of two innovative models of intelligent agents that answer natural language questions about an image. These agents have been implemented with artificial neural networks and utilize computer vision and natural language processing techniques.

We first introduce the model of an intelligent agent for visual question answering that is capable of focusing attention on particular question words and particular image areas in order to increase the accuracy of the generated answer. The idea of using dual attention is that in addition to deciding which areas of the image to focus on, the agent also decides which words of question to pay more attention to. The attention mechanism is implemented through a novel architecture of artificial neural networks.

We present further the model of a visual conversation agent. The agents have in common the ability to answer questions about an image, but the agent of visual conversation stands at a higher level of abstraction. It is not only capable of answering

natural language questions about an image, but also maintaining and understanding the context of the conversation (i.e., previous questions and answers about the same image) and answering follow up questions that are related to the context of the previous questions and answers. During the conversation, the human operator can use reference pronouns for the objects in the image and the agent is capable of understanding what objects these references belong to. This agent is implemented with artificial neural networks and uses dual attention to simultaneously focus on conversation history and certain areas of the image. The dual attention mechanism used by this agent is a novelty for visual conversation as this mechanism has not been explored before for these systems.

The agents have been tested using well-known publicly available datasets (i.e. collections of data), and test results are quantitatively analyzed and compared to similar state of the art models. Test results show that the proposed agents' models are successful and improve the state of the art for the testing metrics. In addition to the quantitative analysis, the test results are also evaluated qualitatively and a discussion of agent capabilities and limitations is made. The results of this analysis show that the proposed innovative architectures are successful and help agents improve the accuracy of answers. Qualitative analysis also shows agents' limitations.

Keywords—Visual question answering systems, visual conversation, intelligent agents, multimodal attention, deep learning, convolutional neural networks, recurrent neural networks

1

Hyrje

Ky kapitull paraqet idenë kryesore dhe kontributet e studimit që përbën ky disertacion. Në të paraqitet problemi që studimi kërkon të zgjidhë, motivimi si dhe qasja e përdorur për zgjidhjen e problemit.

1.1 Problemi

Një nga synimet më të mëdha të inteligjencës artificiale është t'i mundësojë kompjuterave të "shohin" dhe "kuptojnë" botën vizuale që na rrethon dhe t'i mësojë atyre aftësinë për të komunikuar në gjuhën natyrore me njerëzit. Truri njerëzor është përshtatur në mënyrë të tillë që të mund të përpunojë lehtësisht informacionin vizual. Kjo i lejon njerëzit të kuptojnë mjedisin rreth tyre dhe të arsyetojnë në mënyrë komplekse në lidhje me të. Kështu, njerëzit e kanë të lehtë të shohin një imazh dhe të analizojnë e kuptojnë përmbajtjen e tij (*scene understanding)*; të kuptojnë lidhjet që kanë pjesët e imazhit me njëra-tjetrën. Përveç aftësisë së përpunimit të informacionit vizual, truri i njeriut ka aftësi të jashtëzakonshme për të kuptuar dhe komunikuar nëpërmjet gjuhës, gjë që e bën të veçantë dhe e dallon njeriun nga gjallesat e tjera të cilat nuk e kanë këtë aftësi. Për rrjedhojë, njerëzit jo vetëm mund të kuptojnë mjedisin rreth tyre, por dhe mund ta përshkruajnë dhe të bashkëveprojnë me të nëpërmjet komunikimit gjuhësor.

Pavarësisht këtyre aftësive të jashtëzakonshme, truri i njeriut e ka të vështirë të kryejë veprime matematikore komplekse. Nga ana tjetër, kompjuterat kanë aftësi të jashtëzakonshme të kryejnë miliona veprime matematikore nga më komplekset, por e kanë shumë të vështirë të përpunojnë dhe arsyetojnë në lidhje me informacionin vizual dhe gjuhësor, gjë që njerëzit e kanë të natyrshme dhe të lehtë. Pikërisht ky

është dallimi kryesor midis kompjuterave dhe njerëzve. Ne domain-in vizual, në një kompjuter imazhi shprehet si një matricë numrash që tregon nivelin e ndriçimit në secilin pozicion të imazhit. Matrica e numrave mund të ketë miliona elementë dhe qëllimi i inteligjencës artificiale është t'i japë kuptim këtyre numrave. Kjo është një gjë shumë e vështirë për t'u realizuar dhe vështirësohet akoma më shumë për shkak të variacioneve të ndryshme që mund të ketë një objekt (p.sh. ndriçimi, pozicioni, etj.) siç ilustrohet në figurën 1.1.

Figura 1.1. Sfidat në njohjen dhe arsyetimin rreth imazheve. Përshtatur nga [36]. Disa nga sfidat më të mëdha janë: *ndryshimet e pikëpamjes* (*viewpoint variation*) ku objektet e së njëjtës klasë mund të jenë të orientuara në drejtime të ndryshme ndaj kamerës, *shkallëzime të ndryshme (scale variation)*, *kushte të ndryshme ndriçimi* ku i njëjti objekt mund të duket ndryshe për kënde të ndryshme ndriçimi, *deformimi* ku i njëjti objekt mund të ketë pozicione të ndryshme dhe të duket sikur është një objekt i ri, *zhurma e sfondit* (*background clutter*) ku objekti mund të jetë i ngjashëm me sfondin dhe të duket sikur është një me të, *pengesat* ku një objekt mund të mos shfaqet i plotë në imazh duke vështirësuar identifikimin e tij, *variacioni brenda klasës* (*intra-class variation*) ku objekte të së njëjtës klasë mund të kenë pamje tërësisht të ndryshme duke vështirësuar identifikimin e llojit të objektit, etj.

Në domain-in gjuhësor, një fjali në kompjuter mund të shprehet si një sekuencë numrash të plotë që përfaqësojnë pozicionin e çdo fjale përbërëse të saj në fjalor. Është shumë e vështirë që këto numra të kthehen në kuptim dhe për më tepër të mund të kesh një bisedë kuptimplotë në gjuhën natyrore me një kompjuter nëpërmjet këtyre numrave.

Kombinimi i këtyre dy domain-eve, duke krijuar sisteme inteligjente për të realizuar një komunikim të plotë dhe të kuptimtë në gjuhën natyrore midis njerëzve dhe kompjuterave, është problemi dhe objektivi më i madh i inteligjencës artificiale.

1.2 Objektivi i Këtij Punimi dhe Motivimi

Pavarësisht vështirësive, vitet e fundit ka patur një progres të madh në zhvillimin e sistemeve të tilla inteligjente të cilat ndodhen në pikën e bashkimit midis domain-it vizual dhe atij gjuhësor [6], [7], [14], [17], [18], [19]. Ky progres është bërë i mundur falë rrjetave neurale artificiale dhe *deep learning*. Këto algoritma, kombinuar me rritjen e fuqisë përpunuese të hardware-it, kanë mundësuar avancimin e *state of the art* dhe arritjen e rezultateve që deri para pak vitesh konsideroheshin të paarritshme. Pavarësisht progresit të arritur, mbeten akoma shumë sfida dhe vështirësi për t'u tejkaluar.

Objektivi i këtij punimi është krijimi i arkitekturave risi të agjentëve inteligjentë për sistemet pyetje-përgjigje vizuale dhe dialogun vizual të cilët janë të aftë t'i përgjigjen pyetjeve në gjuhën natyrore rreth një imazhi. Qëllimi i krijimit të këtyre arkitekturave është përmirësimi i *state of the art* duke hedhur një hap më tej drejt inteligjencës së plotë artificiale dhe zvogëlimit të distancës midis njerëzve dhe kompjuterave.

Krijimi i sistemeve apo agjentëve të tillë inteligjentë do të sillte shumë përfitime për njerëzimin. Këta agjentë do të gjenin përdorim në shumë fusha të jetës si mjekësi, bursë, sisteme të vendimmarrjes (*decision making systems*), asistent personal, etj.

Me rritjen gjithmonë e më shumë të sasisë së informacionit që na rrethon, bëhet gjithnjë e më i vështirë përftimi në mënyrë të saktë dhe të përmbledhur i informacionit të dëshiruar. Agjentët inteligjentë përfaqësojnë një shkallë të re në evolucionin e motorëve të kërkimit të informacionit dhe do të mundësonin përftimin e informacionit të dëshiruar (*information retrieval*) në gjuhën natyrore për një pyetje të shtruar po në gjuhë natyrore, pa qenë nevoja që njerëzit të shqyrtonin manualisht sasi të mëdha të dhënash.

Në domainin vizual, këta agjentë mund të ndihmonin analistët të nxirrnin informacionin e kërkuar nga sasi të mëdha të dhënash vizuale nëpërmjet pyetjeve në gjuhën natyrore, pa qenë nevoja e përpunimit manual të tyre. Këta agjentë mund të

përdoreshin gjithashtu për të ndihmuar personat me probleme me shikimin për të kuptuar mjedisin rreth tyre. Një përdorim tjetër mund të ishte në misionet e kërkim-shpëtimit ku operatori njerëzor mund të mos kishte akses vizual në të gjithë zonën e kërkimit.

Për shkak të këtyre përfitimeve, kapërcimi i vështirësive dhe avancimi i *state of the art* për këto sisteme është një nga drejtimet kërkimore më aktuale në inteligjencën artificiale [44], [46], [64].

1.3 Kontributet e Këtij Punimi

Kontributi i parë i këtij punimi është studimi i *state of the art* të sistemeve pyetje-përgjigje vizuale dhe dialogut vizual. Rezultatet e këtij studimi paraqiten në mënyrë të strukturuar duke evidentuar, ndër të tjera, edhe sfidat në implementimin e tyre dhe tendencat e kërkimit shkencor. Përfitimi që sjell ky studim është të paturit e një tabloje të qartë rreth problemeve të këtyre sistemeve dhe mundësive për të kontribuar për zgjidhjen e tyre.

Kontributi i dytë i këtij punimi është krijimi i një agjenti inteligjent për sistemet pyetje-përgjigje vizuale i cili i përgjigjet pyetjeve të bëra në gjuhën natyrore rreth një imazhi. Ky agjent përdor mekanizmin e *vëmendjes neurale* për t'u fokusuar njëkohësisht në fjalë të veçanta të pyetjes dhe zona të caktuara të imazhit për të arsyetuar dhe gjeneruar përgjigjen e tij. Agjenti është një model i implementuar me rrjeta neurale artificiale dhe përdor *rrjetat neurale konvolucionale* (CNN) për përpunimin e imazhit dhe *rrjetat neurale rekurrente* (RNN) për përpunimin e sekuencave të fjalëve. Mekanizmi i vëmendjes implementohet me anë të një arkitekture risi *long short-term memory* (LSTM) e cila përdor vëmendjen tekstuale dhe vizuale (në vijim do i referohemi me termin *vëmendje multimodale*) në çdo qelizë të rrjetit LSTM.

Përfitimi që sjell kjo arkitekturë risi LSTM është përmirësimi i *state of the art* për sistemet pyetje përgjigje-vizuale bazuar mbi testimet e kryera mbi dy *dataset*-e të mirënjohura publike VQA [9] dhe Vizual 7W [7].

Kontributi i tretë i këtij punimi është krijimi i një agjenti inteligjent për dialogun vizual i cili i përgjigjet pyetjeve në gjuhën natyrore rreth një imazhi në kontekstin e një dialogu. Ky agjent ruan kontekstin e pyetjeve dhe përgjigjeve të dhëna më parë rreth të njëjtit imazh dhe është në gjendje t'i përgjigjet pyetjeve të reja të cilat mund të kenë lidhje me hapa të mëparshëm të dialogut. Edhe ky agjent është ndërtuar duke kombinuar rrjeta CNN për përpunimin e imazhit dhe rrjeta RNN për përpunimin e sekuencave të fjalëve.

Risia që sjell ky model është përfshirja e mekanizmit të vëmendjes multimodale. Ky mekanizëm nuk është eksploruar më parë për dialogun vizual. Përfitimi që sjell ky mekanizëm është përmirësimi i *state of the art* për të gjitha metrikat e testuara mbi datasetin VisDial [64].

Kontributi i katërt i këtij punimi është prezantimi i dy pyetjeve të reja (konkretisht *numër* dhe *ngjyrë*) për datasetin Visual7W [7] për sistemet pyetje-përgjigje vizuale. Në kapitullin 5 paraqitet motivimi dhe arsyetimi i përfshirjes së këtyre dy pyetjeve. Përfitimi që sjell përfshirja e tyre është një vlerësim më i mirë i aftësive të modeleve të sistemeve pyetje-përgjigje vizuale. Mospërfshirja e tyre nuk do të lejonte vlerësimin e drejtë të aftësive të këtyre sistemeve, veçanërisht aspektin e numërimit të objekteve e cila dihet që është një detyrë e vështirë e *computer vision*.

1.4 Struktura e Disertacionit

Në këtë disertacion paraqiten dy modele agjentësh të cilët i përgjigjen pyetjeve në gjuhën natyrore në një kontekst vizual. Agjentët implementohen me arkitektura risi rrjetash neurale artificiale për përpunimin e dy modaliteteve (vizual dhe gjuhësor) të informacionit në hyrje. Modelet trajnohen dhe testohen mbi *dataset*-e të mirënjohura publike.

Në **Kapitullin 2** paraqiten bazat teorike të sistemeve pyetje-përgjigje dhe dialogut vizual. Në pjesën e parë të tij paraqitet një studim i *state of the art* i sistemeve pyetje-përgjigje duke evidentuar ndër të tjera edhe sfidat në implementimin e tyre dhe tendencat e reja të kërkimit shkencor. Në pjesën e dytë prezantohet dialogu

vizual, ndryshimet që ka ai nga sistemet pyetje-përgjigje vizuale dhe mënyra e vlerësimit të sistemeve të dialogut vizual.

Në **Kapitullin 3** paraqiten bazat e nevojshme matematikore, teorike dhe praktike të *machine learning*, rrjetave neurale artificiale dhe *deep learning*. Gjithashtu paraqiten arkitektura dhe konfigurime të rrjetave neurale artificiale të përdorura zakonisht në praktikë, në veçanti për përpunimin e imazheve dhe tekstit.

Në **Kapitullin 4** paraqiten punimet e ngjashme dhe që kanë lidhje me temën e këtij disertacioni të cilat janë të realizuara midis vitit 2014 dhe 2017 në sistemet pyetje-përgjigje vizuale, dialogut vizual dhe vëmendjes neurale.

Kapitujt 5 dhe 6 përbëjnë kontributin kryesor të këtij punimi. Në **Kapitullin 5** paraqitet modeli i një agjenti inteligjent për sistemet pyetje-përgjigje vizuale së bashku me detajet e implementimit të tij. Gjithashtu në këtë kapitull paraqiten edhe rezultatet e testimeve si dhe bëhet një analizë sasiore dhe cilësore e tyre duke paraqitur edhe konkluzionet përkatëse.

Shkalla e abstraksionit rritet në **Kapitullin 6** në të cilin paraqitet modeli dhe implementimi i një agjenti inteligjent për dialogun vizual. Edhe për këtë agjent paraqiten rezultatet e vlerësimit sasior dhe cilësor së bashku me konkluzionet përkatëse.

Në **Kapitullin 7** paraqiten konkluzionet e këtij punimi së bashku me drejtime të mundshme të punës në të ardhmen në këtë fushë.

2

Sistemet Pyetje-Përgjigje dhe Dialogu Vizual

Ky kapitull paraqet një studim të *state of the art* të sistemeve pyetje-përgjigje. Ky studim është bazuar mbi 147 artikuj shkencorë të botuar gjatë viteve 2014-2016 në konferencat dhe revistat më të rëndësishme të information retrieval, inteligjencës artificiale, machine learning, përpunimit të gjuhës natyrore (*natural language processing*), computational linguistic, neural information processing, web intelligence, semantic web, etj. Qëllimi është të evidentohen qasjet në implementimin e këtyre sistemeve, sfidat në avancimin e tyre dhe tendencat e reja të kërkimit shkencor. Rezultatet e plota të studimit janë paraqitur në [44], [46] dhe [72].

Përveç studimit të sistemeve pyetje-përgjigje, në këtë kapitull prezantohet gjithashtu dhe dialogu vizual, mënyrat e vlerësimit të sistemeve të tilla dhe ndryshimet nga sistemet pyetje-përgjigje vizuale.

2.1 Sistemet Pyetje-Përgjigje

Sistemet pyetje-përgjigje (*question answering systems)* gjenden në pikën e takimit midis *information retrieval* (kërkimi dhe gjetja e informacionit) dhe *natural language processing* (përpunimi i gjuhës natyrore). Ato merren me detyrën e vështirë të gjetjes dhe kthimit të një përgjigjeje të saktë nga një kompjuter për një pyetje të bërë në gjuhën natyrore. Me rritjen e sasisë së informacionit rreth nesh, rritet dhe nevoja për përpunimin e tij në mënyrë efikase. Avancimi i motorëve të kërkimit ka përmirësuar dhe lehtësuar mënyrat e përpunimit dhe gjetjes së informacionit të

kërkuar. Sidoqoftë, kërkimi i informacionit me anë të këtyre motorëve është ende në fazën e përdorimit të fjalëve kyçe dhe jo një pyetjeje në gjuhën natyrore. Për më tepër, në shumicën e rasteve, motorët e kërkimit nuk janë të aftë të kthejnë një përgjigje të saktë dhe të përmbledhur edhe pas kërkimit me fjalë kyçe. Në këto raste ata kthejnë vetëm një listë dokumentash apo burimesh qe kanë një probabilitet të lartë të përmbajë informacionin e kërkuar dhe përdoruesve u duhet të kërkojnë manualisht nëpër to për të gjetur përgjigjen e duhur. Shpeshherë përgjigja mund të gjendet e shpërndarë në disa burime të ndryshme informacioni. Kjo gjë e vështirëson akoma më shumë gjetjen e informacionit të kërkuar dhe rrit kostot e tij. Detyra e sistemeve pyetje-përgjigje (SPP) është pikërisht rritja e efikasitetit të kërkimit dhe gjetjes së informacionit duke i mundësuar kompjuterave të kthejnë një përgjigje të saktë dhe të përmbledhur duke përmbushur nevojat e përdoruesit pa qenë nevoja që ky i fundit të bëjë kërkim manual nëpër burimet e informacionit. Për më tepër, synimi i SPP është gjenerimi i përgjigjes së saktë në gjuhë natyrore për një pyetje të bërë nga përdoruesi po në gjuhë natyrore, pra pa u limituar thjesht në përdorimin e fjalëve kyçe.

SPP-të gjejnë përdorim në shumë fusha të tilla si: ekstraktimi i dokumentave, bashkëveprimi njeri-kompjuter (*human-computer interaction)*, menaxhim dhe klasifikim i dokumentave, kërkimi dhe gjetja e informacionit mjekësor (*medical information retrieval*), etj.

Në figurën 2.1 paraqiten hapat e përpunimit të informacionit në një SPP. Informacioni përpunohet në tre faza kryesore [52]:

1. *Analiza e pyetjes:* Gjatë kësaj faze bëhet analiza sintaksore dhe semantikore e pyetjes duke përdorur teknika të *natural language processing* (NLP). Kjo është faza ku SPP duhet të “kuptojë” pyetjen e bërë në gjuhë natyrore dhe më pas ta përdorë atë si input për fazën e dytë të procesimit.
2. *Analiza e bazës së njohurive:* Gjatë kësaj faze bëhet ekzaminimi i një sërë dokumentash ose i një baze të dhënash e cila njihet me emrin *baza e njohurive (knowledge base*). Kjo bazë njohurish përmban gjithë informacionin mbi të cilin SPP bazohet për të gjeneruar përgjigjen.

Ekzaminimi i njohurive bëhet për të gjetur informacionin që përkon më shumë me pyetjen. Pas ekzaminimit të bazës së njohurive mblidhen një sërë dokumentash dhe përgjigjesh të cilat janë kandidate të mundshme për të qenë përgjigja e saktë.

3. *Analiza e përgjigjes:* Gjatë kësaj faze bëhet analiza e përgjigjeve kandidate dhe përpunimi i tyre për të gjeneruar përgjigjen e saktë e cila i kthehet përdoruesit. Shpeshherë, SPP kthen një sërë përgjigjesh të renditura sipas një shpërndarjeje probabilitare ku përgjigja e renditur e para ka probabilitet më të lartë për të qenë e saktë.

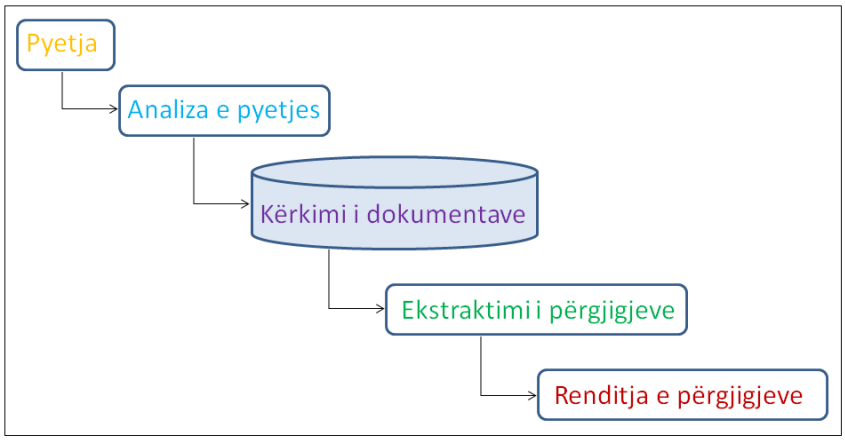


Figura 2.1. Përpunimi i informacionit në një sistem pyetje-përgjigje [52]. Informacioni përpunohet në tre faza kryesore: *analiza e pyetjes:*analiza sintaksore dhe semantikore e pyetjes, *analiza e bazës së njohurive:*gjetja e dokumentave dhe përgjigjeve kandidate, *analiza e përgjigjes:*përpunimi i përgjigjeve kandidate dhe renditja e tyre.

Sistemet pyetje-përgjigje ndahen në dy grupe kryesore: *sisteme pyetje-përgjigje tekstuale* dhe *sisteme pyetje-përgjigje vizuale*. Sistemet pyetje-përgjigje tekstuale janë sisteme në të cilat burimi i informacionit është në formë teksti dhe funksionimi i tyre bazohet mbi përpunimin e këtij informacioni tekstual. Në kontrast, në sistemet pyetje-përgjigje vizuale (SPPV) burimi i informacionit është një imazh dhe qëllimi i sistemit është të përpunojë këtë informacion vizual në mënyrë që të ktheje përgjigjen e saktë. Ekzistojnë gjithashtu edhe sisteme hibride të cilat përdorin si burim informacioni të dhëna tekstuale dhe një imazh [19], por këto janë shumë të rralla.

Sistemet pyetje-përgjigje tekstuale mund të klasifikohen nga tre pikëpamje të ndryshme: *1. lloji i domain-it* (i mbyllur ose i hapur), 2. *lloji i burimit të informacionit* (i strukturuar ose i pastrukturuar), *3. aspekti bashkëpunues* (komunitar ose jo-komunitar).

Lloji i domain-it. Nga kjo pikëpamje SPP-të mund te klasifikohen në dy lloje: *me domain të hapur* (*open domain*) dhe *me domain të mbyllur* (*closed domain*). Tek SPP-të me domain të hapur baza e njohurisë mund të jetë shumë e madhe dhe e pa kufizuar ne një domain specifik dhe pyetjet mund të behën rreth çdo fushe të njohurive. Tek SPP-të me domain të mbyllur informacioni është i kufizuar në një domain specifik (p.sh. mjekësia, sporti, etj.) dhe sistemi i pergjigjet pyetjeve që behën vetëm mbi këtë domain. Ky lloj sistemi është më i thjeshtë në krahasim me SPP me domain të hapur për arsye se përpunohet me pak informacion dhe baza e njohurive është më e vogël. Këto sisteme zakonisht kanë saktësi më të madhe, por mund të kërkojnë përpunim gjuhësor më të gjerë. Në anën tjetër, SPP-të me domain të hapur kanë një saktësi më te ulët (veçanërisht në rastin e sistemeve shumëgjuhëshe).

Lloji i burimit të informacionit. Ekzistojnë dy lloje burimesh informacioni: *i strukturuar* dhe *i pastrukturuar*. Në sistemet me burim të strukturuar, informacioni organizohet në formën e një baze njohurish të strukturuar ku të dhënat lidhen me anë të semantikës. Informacioni në bazën e njohurive organizohet në formën e njësive treshe (*triplets*) të përbëra nga subjekti, atributi (*predicate*) dhe objekti. Një shembull i një *triplet* është (*Mali Everest, lartësia, 8,848 m*). Atributi është një karakteristikë e subjektit ndërsa objekti është vlera e kësaj karakteristike. Në këto sisteme pyetja bëhet mbi subjektin dhe atributin ndërsa përgjigja është objekti. Pyetjet në gjuhën natyrore përkthehen në *query* në gjuhë formale që thjesht nxjerrin informacionin nga baza e strukturuar e njohurive. Procesi perfshin teknika gjuhësore si *parsing* i strukturës sintaksore të pyetjes, *POS tagging*, *tokenization*, detektimi i simetrisë semantikore, zgjidhja e dykuptimësisë (*ambiguity*) etj. Në sistemet me burim të pastrukturuar informacioni, burimi i informacionit është Web-i ose bashkësi dokumentash të përfshira brenda sistemit. Gjenerimi i përgjigjes është i bazuar në *information retrieval*. Kjo do të thotë se sistemi gjen segmente të shkurtra të tekstit nga Web-i ose brenda bashkësisë së dokumentave dhe i analizon ato në mënyrë që të gjejë përgjigjet

kandidate. Përgjigjet kandidate më pas renditen dhe përpunohen në mënyrë që të gjendet përgjigja më e vlefshme.

Aspekti bashkëpunues. Nga kjo pikëpamje ekzistojnë dy lloje sistemesh: *komunitare* dhe *jo-komunitare*. Sistemet komunitare janë sisteme bashkëpunimi moderne ku përdoruesit mbështeten në ekspertizën nga komuniteti për të marrë një përgjigje për pyetjet e tyre (p.sh. *Quora*[1], *Stack Overflow*[2], etj). Pyetjet e reja i përcillen përdoruesve që mund t'u përgjigjen atyre bazuar në nivelin e tyre të ekspertizës ndaj pyetjes. Këto lloj sistemesh përdoren kur nevojat për informacion nuk plotësohen thjesht duke parë një faqe Web [61]. Pyetjet dhe pëgjigjet e tyre korresponduese prezantohen shpesh si rezultatet kryesore të kërkimit. Ato zakonisht renditen sipas përshtatshmërisë ndaj pyetjeve të përdoruesve dhe mund të shihen nga të gjithë përdoruesit që bëjnë të njëjtën pyetje. Kjo gjë mund të veprojë si një mekanizëm për të ulur numrin e pyetjeve të dublikuara.

2.2 Qasjet në Implementimin e Sistemeve Pyetje-Përgjigje

Nevojiten qasje dhe strategji të shumta për të gjeneruar përgjigjen e saktë duke u bazuar në llojet e ndryshme të pyetjeve. Ekzistojnë tre qasje kryesore [74] në implementimin e sistemeve pyetje-përgjigje për të analizuar pyetjet në gjuhë natyrore dhe bazën e njohurive: *gjuhësore*, *statistikore* dhe *përputhja e modelit (pattern matching*). Në modelet e paraqitura në këtë disertacion është përdorur qasja statistikore e implementuar me rrjeta neurale artificiale.

2.2.1 Qasja Gjuhësore

Një sistem i tillë përdor metodat e inteligjencës artificiale të cilat integrojnë bazat e njohurive dhe teknikat e natural language processing [11], [52], [53], [54], [55], [56], [58]. Gjatë përpunimit, pyetja konvertohet në një *query* i cili aplikohet mbi bazën e njohurive për të nxjerrë përgjigjet kandidate. Shpeshherë përdoren *regex* për

[1] https://www.quora.com/
[2] https://stackoverflow.com/

të identifikuar entitetet (d.m.th. objektet rreth të cilëve bëhet pyetja) brënda pyetjes dhe karakteristikat e tyre.

2.2.2 Qasja Statistikore

Kjo qasje përdor teknika të *machine learning* të cilat punojnë mbi sasi të mëdha të dhënash. Ajo kërkon një sasi më të madhe të dhënash për trajnimin e SPP-ve, por prodhon rezultate më të mira se qasjet e tjera. Qasja statistikore përdor teknika të tilla si klasifikuesit *Bayesian*-ë, klasifikuesit *support vector machine (SVM)*, modele të entropisë maksimale (*maximum entropy models*), rrjetat neurale, etj, për të analizuar pyetjet e përdoruesve. Ata janë të pavarur nga gjuhët e strukturuara të query-ve dhe mund të formulojnë query-t në gjuhë natyrore. Gjithashtu, ata janë të pavarur nga një gjuhë e caktuar natyrore si dhe mund të përshtaten lehtësisht për domain-e të ndryshme.

2.2.3 Qasja e Përputhjes së Modelit

Qasja *pattern-matching* bazohet në njohuri gjuhësore ose leksikore në lidhje me strukturën e tekstit që do të përpunohet. Kjo njohuri ekstraktohet nga *dataset*-i duke përdorur modele (*pattern*) të paracaktuara ose të mësuara nga SPP-ja. Për shembull, pyetja “Sa i lartë është mali Everest?” ka modelin “Sa i lartë është <emri i entitetit>?” dhe modeli i përgjigjes do të jetë “<Emri i entitetit> është <lartësia>.”

2.2.4 Qasja Hibride

Pavarësisht se ekzistojnë dallime të qarta midis qasjeve te ndryshme, shpeshhere nevojitet krijimi i sistemeve të cilat e kanë të pamundur aplikimin e një qasjeje të vetme ose kërkohen sisteme më të thjeshta të cilat shfrytëzojnë karakteristikat e më shumë se një qasjeje të vetme. Së bashku me kombinimin e qasjeve të ndryshme, trashëgohen dhe avantazhet dhe disavantazhet e tyre respektive. Kjo i hap rrugën mundësive për të shfrytëzuar avantazhet e secilës teknikë. Një nga sistemet më të mirënjohura që përdor qasjen hibride është sistemi Watson QA i IBM-së [57].

Në tabelën 2.1 tregohen avantazhet dhe disavantazhet e secilës qasje.

Tabela 2.1. **Krahasimi midis qasjeve në implementimin e sistemeve pyetje-përgjigje**

Qasja	**Avantazhet**	**Disavantazhet**
Gjuhësore (NLP)	• Pyetja mund të bëhet në gjuhë natyrore në vend të query-ve të strukturuara	• Nuk ka portabilitet ndërmjet domain-eve të ndryshëm • Përdorimi i Web-it si një bazë njohurish është i pamundur sepse nuk mund të indeksohet paraprakisht • Zakonisht janë me domain të hapur • Në pyetjet në formë paragrafi është i vështirë identifikimi i entiteteve
Statistikore	• Nuk kërkon njohuri rreth domain-it • Nuk ekzistojnë probleme të NLP si p.sh. gramatika • Pyetjet komplekse përpunohen në mënyrë më efikase • Burime të dhënash heterogjene • Portabiliteti • Pavarësia nga gjuha • Mund të përdoret në pyetjet me domain të hapur	• Trajnimi kërkon kohë • Trajnimi kërkon një sasi më të madhe të dhënash • Çdo fjalë e pyetjes trajtohet në mënyrë të pavarur dhe tiparet gjuhësore të kombinimeve të fjalëve dhe shprehjeve nuk mund të identifikohen. Nuk merret parasysh semantika dhe konteksti i fjalëve dhe shprehjeve.
Sipas modelit	• Kërkon më pak të dhëna trajnimi • Nuk ndikohet nga lloji i gjuhës së query-t • Pyetjet komplekse përpunohen në mënyrë më efikase	• Kërkohet njohuri leksikore dhe njohuri rreth domain-it • Gjuha natyrore nuk ndjek model të caktuar dhe krijimi i një modeli të saktë është i vështirë.

2.3 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje

Lloje të ndryshme sistemesh kanë problematika të ndryshme. Për të kuptuar më mirë secilën prej tyre, ato janë ndarë në dy grupe: *sisteme pyetje-përgjigje me bazë njohurish* dhe *sisteme pyetje-përgjigje komunitare*.

2.3.1 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje me Bazë Njohurish

Sfidat më të mëdha në sistemet pyetje-përgjigje me bazë njohurish janë:

- **Boshllëku leksikor (*lexical gap*) midis gjuhës natyrore dhe semantikës së strukturuar të bazës së njohurive.** Ky është problemi më i madh me këto SPP dhe ka lidhje me ndryshimin midis gjuhës natyrore të pastrukturuar dhe mënyrës së paraqitjes së informacionit në bazën e njohurive . Ky problem është shpeshherë shkaktari apo përkeqësuesi i një sërë problemesh dhe sfidash të tjera. Gjuha natyrore nuk ndjek një model të caktuar. Për më tepër ekzistojnë shumë mënyra për të thënë të njëjtën gjë. Nga ana tjetër, informacioni në bazën e njohurive është i mirëorganizuar dhe i strukturuar në mënyrë semantikore. Ky është një problem i vështirë për t'u tejkaluar pasi informacioni i kërkuar mund të ekzistojë në bazën e njohurive por mënyra se si është shprehur pyetja në gjuhën natyrore mund të bëjë që SPP mos arrijë dot të vendosë korrepondencën e duhur midis pyetjes dhe informacionit të duhur. Kjo mund të rezultojë në një përgjigje të pasaktë ose sistemi SPP mos jetë në gjendje të kthejë një përgjigje.

- **Identifikimi dhe lidhja e entiteteve:** Ky problem ka lidhje me aftësinë që ka një SPP për të identifikuar entitetin e pyetjes dhe për të vendosur korrespondencë të saktë me *triplet* korrekte në bazën e njohurive. Kjo gjë vështirësohet akoma më shumë për shkak të boshllëkut leksikor midis pyetjes në gjuhë natyrore dhe shprehjes së informacionit në bazën e njohurive.

- **Gjenerimi i përgjigjes në gjuhë natyrore.** Ky problem ka lidhje me aftësinë e SPP-së për të gjeneruar përgjigjen në gjuhën natyrore duke mos u mjaftuar vetëm me gjenerimin e informacionit të kërkuar. Për shembull, për pyetjen "Sa ditë ka një vit i brishtë?" sistemi mund të përgjigjet "Një vit i brishtë ka 366 ditë." në vend të thjesht "366". Kjo është një karakteristikë e dëshirueshme për SPP-të pasi e bën sistemin më "të zgjuar".

- **Pyetjet që përfshijnë entitete të shumëfishta.** Ky problem ka lidhje me aftësinë që duhet të ketë një SPP për të identifikuar entitetet e shumëfishta në

një pyetje dhe të vendosë korrespondencën e duhur me *triplets* në bazën e njohurive. Kjo gjë vështirësohet akoma më shumë për shkak të boshllëkut leksikor midis pyetjes në gjuhë natyrore dhe shprehjes së informacionit në bazën e njohurive.

- **Pyetja në formë paragrafi.** Metodat e përpunimit të tekstit janë të efektshme për sistemet faktoide (d.m.th. sisteme ku pyetjet paraqiten kryesisht në formën e një fjalie të vetme që fillon me fjalët kush, çfarë, ku, kur, etj.). Ekziston një lloj tjetër pyetjeje e quajtur pyetje në formë paragrafi (p.sh. *quiz bowl*[3]) e cila përbëhet nga një bashkësi thëniesh që japin të dhëna rreth përgjigjes. Metodat e zakonshme të përpunimit të pyetjes janë të paefektshme kur pyetja përmban shumë pak fjalë të cilat japin të dhëna rreth përgjigjes sepse ato e trajtojnë çdo fjalë në mënyrë të pavarur dhe nuk identifikojnë lidhjen midis entiteteve. Për më tepër, identifikimi i entitetit që përbën përgjigjen është i vështirë sepse pyetja përmban thjesht fjalë që e përshkruajnë atë nga pikëpamje të ndryshme. Gjetja e përgjigjes së saktë kërkon njohuri rreth domain-it të pyetjes.
- **Përgjigja në formë paragrafi.** Kjo është një sfidë e sistemeve jo-faktoide ku përgjigja është në formë paragrafi. Në mënyrë që të gjejë përgjigjen e duhur, sistemi duhet të analizojë bazën e të dhënave dhe të mbledhë informacion nga disa *triplets*. Më pas sistemi duhet të përpunojë këtë informacion dhe të gjenerojë një përgjigje të zgjeruar në formë paragrafi për pyetjen e përdoruesit duke mos u mjaftuar vetëm me një fjalë apo fjali. Kjo është një detyrë e vështirë për t'u realizuar pasi kërkon aftësi për të analizuar informacionin që përmbajnë disa *triplets* për të kuptuar lidhjet komplekse semantikore që ekzistojnë midis tyre. Kjo detyrë vështirësohet akoma më shumë nga fakti që përgjigja duhet të jetë e zgjeruar dhe në gjuhë natyrore.

[3] https://www.naqt.com/about-quiz-bowl.html

2.3.2 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje Komunitare

Sfidat më të mëdha në sistemet pyetje-përgjigje komunitare janë:

- **Boshllëku leksikor midis pyetjeve:** Kjo është sfida më e madhe për këto lloj sistemesh. Ka të bëjë me ndryshimet në formulimin në gjuhën natyrore të pyetjeve. Shpeshherë, përdorues të ndryshëm kërkojnë të njëjtin informacion duke formuluar pyetjen në mënyra të ndryshme duke u bazuar edhe në nivelin e tyre të ekspertizës (p.sh. përdorues që janë më të familjarizuar me objektin e pyetjes mund të përdorin terma më të saktë dhe formalë krahasuar me përdorues që janë më pak të familjarizuar). Kjo gjë sjell si rezultat praninë e pyetjeve që janë semantikisht të barabarta, por ndryshojnë nga ana leksikore. Është detyrë e SPP-së që të dallojë këto raste dhe të fshijë pyetjet e dublikuara.
- **Boshllëku leksikor midis pyetjeve dhe përgjigjeve:** Ky është gjithashtu një nga problemet më të mëdha për sistemet pyetje-përgjigje komunitare. Ashtu si në rastin e boshllëkut leksikor midis pyetjeve, shpeshherë, pyetjet dhe përgjigjet mund të jenë shumë asimetrike në lidhje me informacionin që përmbajnë. Gjithashtu, ekziston edhe boshllëku i terminologjisë teknike midis pyetjeve dhe përgjigjeve. Shpeshherë pyetjet bëhen nga përdorues që nuk janë plotësisht të familjarizuar me terminologjinë teknike ndërsa përgjigjet bëhen nga përdorues më ekspertë që përdorin terminologji më të avancuar. Është përsëri detyrë e SPP-së të dallojë këto raste dhe t'i trajtojë si të sakta përgjigjet që mund të ndryshojnë në mënyrë të konsiderueshme nga pyetjet në aspektin leksikor.
- **Mungesa e të dhënave të mjaftueshme për identifikimin e ekspertëve:** Një sfidë tjetër e SPP-ve komunitare është pjesëmarrja e ulët e përdoruesve që i përgjigjen saktë pyetjeve (në vazhdim do i referohemi si *ekspertë*). Zakonisht, çdo pyetje gjen përgjigje nga pak përdorues. Si rezultat i kësaj, sistemi e ka të vështirë të identifikojë ekspertët e duhur dhe shumë pyetje të reja ngelen pa përgjigje sepse sistemi nuk mund t'i drejtojë ato drejt ekpertëve që mund t'i përgjigjen.

- **Devijimi nga pyetja:** Ky problem ka të bëjë me faktin që shpeshherë përgjigjet bëhen irrelevante ndaj pyetjes. Në sistemet komunitare përgjigjet jepen në formën e komenteve ndaj pyetjes dhe shpeshherë përdoruesit hyjnë në bashkëbisedime me njëri tjetrin dhe devijojnë nga pyetja fillestare. Është detyrë e SPP-së të dallojë këto raste dhe të gjejë e promovojë përgjigjen e saktë midis komenteve të përdoruesve.

2.4 Tendencat e Kërkimit Shkencor në Sistemet Pyetje-Përgjigje

Me avancimin e teknikave të përpunimit të gjuhës natyrore dhe *machine learning*, sistemet pyetje-përgjigje tekstuale po bëhen gjithmonë e më komplekse dhe inteligjente dhe aftësi të reja po shtohen gjithmonë e më shumë në këto sisteme. Disa nga tendencat e kërkimit shkencor për këto sisteme janë:

- **Sisteme pyetje-përgjigje jo faktoide:** Një SPP faktoid ofron përgjigje të përmbledhura për pyetje të tipit "Sa njerëz jetojnë në tokë?", "Kur është ekuinoksi i pranverës?", etj. Limitimi i këtyre sistemeve qëndron në faktin se një pjesë e konsiderueshme e pyetjeve nuk janë faktoide. Në kontrast, një sistemi pyetje-përgjigje jofaktoid mund t'i bëhet një pyetje rreth një problemi matematikor apo si ndërrohet vaji i makinës dhe përgjigja që kthen sistemi është e një natyre përshkruese. Shpeshherë, përgjigja e pyetjeve jofaktoide nuk mund të gjendet thjesht duke kryer një kërkim në një motor të thjeshtë kërkimi sepse gjenerimi i përgjigjes kërkon njohuri rreth domainit të pyetjes. Për këto arsye ka një interes dhe nevojë në rritje për krijimin e sistemeve të tilla inteligjente të cilat realisht e "kuptojnë" pyetjen [24], [34].
- **Përdorimi i bazave të njohurive të shumëfishta:** Është e zakonshme që një pyetje të përfshijë shumë aspekte të njohurisë dhe mund të bëhet jopraktik përdorimi i një baze njohurish të vetme që të mbulojë të gjitha fushat e njohurive të pyetjes. Me rritjen e vazhdueshme të web-it semantik dhe të bazave të njohurive gjithmonë e më komplekse të cilat mbulojnë një fushë të caktuar njohurish, bëhet gjithmonë e më e rëndësishme krijimi i metodave për të aksesuar informacionin nga disa baza njohurish [62]. Kjo është një tendencë

e kërkimit shkencor në SPP e nxitur nga nevoja për të krijuar sisteme më fleksibël të cilat marrin dhe validojnë përgjigjen në burime të ndryshme kur një bazë njohurish e vetme nuk mjafton.

- **Sisteme bashkëbiseduese (*Conversational Question Answering*):** Kjo është aktualisht tendenca më vizionare në SPP. Ajo ka të bëjë me krijimin e një SPP i cili është në gjendje të kthejë përgjigje për pyetjen e përdoruesit nën kontekstin e një bisede të vazhdueshme. Përdoruesi bën pyetje të njëpasnjëshme rreth fushës së tij të interesit dhe sistemi është në gjendje të ruajë kontekstin e bisedës dhe të kthejë përgjigjet e duhura. Ajo që i bën të veçantë keta sistemeve është se përdoruesi mund të vazhdojë kërkimin e tij duke përdorur përemra dhe referenca të tjera për të dhënat e pyetjeve dhe përgjigjeve të mëparshme. Sistemi duhet të jetë i aftë të kuptojë objektin e këtyre referencave dhe të kthejë përgjigjet e duhura. Përveç të kuptuarit e "temës" së bisedës, një sfidë tjetër e sistemit është të kuptojë kur "tema" ndryshon. Kërkimi për të krijuar sisteme të tilla është drejtuar drejt rrjetave neurale artificiale [59] dhe grafeve te njohurisë (*knowledge graphs*) [60].

2.5 Sistemet Pyetje-Përgjigje Vizuale

Sistemet pyetje-përgjigje vizuale janë shfaqur vitet e fundit [18] si një problem kërkimor multidisiplinar i cili qëndron në pikën e takimit midis *inteligjencës artificiale, përpunimit të gjuhës natyrore* (*natural language processing*) dhe *computer vision*. Avancimet në këto fusha dhe në teknikat e *deep learning* kanë rezultuar në krijimin e sistemeve inteligjente që kanë arritur rezultate mahnitëse në rrugën drejt arritjes së qëllimit më të madh të *computer vision*: të kuptuarit holistik të skenës (*holistic scene understanding*) [15].

Sistemet pyetje-përgjigje vizuale vlerësojnë aftësitë dhe problemet e një makine për të kuptuar skenën e imazhit. Gjithashtu ato matin inteligjencën e makinës në domainin vizual. Në qendër të një SPPV-je është një agjent inteligjent i cili i përgjigjet një pyetjeje rreth një imazhi. Pyetja bëhet nga një përdorues njerëzor në gjuhë njerëzore dhe përgjigja e agjentit duhet të jetë gjithashtu në gjuhë njerëzore.

Agjenti duhet të bëjë lidhjen midis pyetjes dhe imazhit dhe të arsyetojë rreth të dyjave në mënyrë që të mund të kthejë një përgjigje të saktë. Kjo gjë kërkon jo vetëm kuptim të mirë të imazhit, por edhe të pyetjes. Ky është një problem kompleks dhe fokusohet shumë në inteligjencën artificiale dhe veçanërisht në procesin e *inferencës* për të gjeneruar përgjigjen pasi lloje të ndryshme pyetjesh (p.sh. ngjyra, numri, vendodhja, etj.) kërkojnë lloje të ndryshme përgjigjesh. Ka gjithashtu pyetje të cilat kërkojnë arsyetim mbi mendimin praktik (*commonsense reasoning*) të tilla si "A duken njerëzit të gëzuar?" të cilat vështirësojnë akoma më shumë arsyetimin e agjentit.

2.5.1 Sfidat në Realizimin e Sistemeve Pyetje-Përgjigje Vizuale

Sfidat më të mëdha në realizimin e sistemeve pyetje-përgjigje vizuale janë:

- **Përfshirja e mendimit praktik:** Kjo është sfida dhe gjithashtu aspirata më e madhe jo vetëm e SPPV-ve, por në përgjithësi e të gjitha sistemeve të cilat përpunojnë gjuhën natyrore. Shpeshherë pyetjet dhe/ose imazhi mund të përmbajnë brenda tyre informacion rreth mendimit praktik (*common sense*) dhe kuptimi dhe shfrytëzimi i këtij mendimi praktik nga SPPV-ja për të arsyetuar me saktë rreth pyetjes është një aftësi e dëshirueshme. Për më tepër gjuha njerëzore ka tendenca dykuptimësie (*ambiguity*) pasi e njëjta fjalë apo shprehje mund të ketë kuptime të ndryshme në kontekste të ndryshme. Një model i mirë SPPV duhet të jetë në gjendje të dallojë kuptimin në të cilin është thënë një fjalë apo një shprehje në kontekstin e imazhit.
- **Numërimi i objekteve:** Kjo është një nga sfidat më të mëdha të SPPV-ve. Ajo ka lidhje me aftësinë e sistemit për të dalluar, arsyetuar dhe numëruar instanca të ndryshme të të njëjtës klasë objektesh. Kjo sfidë vështirësohet akoma më shumë kur objekte të së njëjtës klasë kanë tipare shumë të ndryshme nga njëri-tjetri sepse sistemit i duhet jo vetëm t'i dallojë ato, por edhe të arsyetojë drejt rreth klasës së tyre.
- **Dallimi nga njëri-tjetri i objekteve që në imazh shfaqen të "shkrirë" me njëri-tjetrin ose me sfondin.**

- **Identifikimi i të dhënave në pyetje dhe/ose imazh që kanë lidhje me pyetjen:** Kjo është një sfidë tjetër shumë e madhe e SPPV-ve dhe mbi të cilën po punohet shumë me anë të implementimit të mekanizmave të *vëmendjes neurale*. Ajo ka lidhje me aftësinë e sistemit për të arsyetuar rreth pyetjes dhe/apo imazhit për të identifikuar fjalë të pyetjes dhe/ose zona të imazhit të cilat mund të shërbejnë për të gjeneruar një përgjigje të saktë.
- **Implementimi i njohurive të jashtme:** Një sfidë tjetër e vështirë dhe gjithashtu aspiratë e SPPV-ve dhe në përgjithësi e sistemeve që përpunojnë informacion vizual është gjetja e një mënyre për të implementuar njohuri të jashtme. Me njohuri të jashtme, në rastin e SPPV-ve nënkuptojmë informacionin shtesë përveç imazhit i cili do t'i shërbente sistemit për të arsyetuar më mirë rreth pyetjes dhe përgjigjes. Për shembull njohuria "Bretkosat me ngjyra të ndezura janë helmuese." mund ta ndihmonte sistemin të arsyetonte më mirë rreth imazhit të një bretkose dhe pyetjes "A është helmuese bretkosa?".

2.6 Dialogu Vizual

Dialogu vizual, edhe pse është shfaqur rishtazi si një fushë kërkimi [63], [64], [65], [66], [67], është një aspiratë e vjetër e *machine learning* për të krijuar sisteme inteligjente të cilat mund të komunikojnë me njerëzit në domainin vizual dhe gjuhësor. Njësoj si sistemet pyetje-përgjigje vizuale, dialogu vizual qëndron në pikën e takimit midis *inteligjencës artificiale*, *computer vision*, dhe *përpunimit të gjuhës natyrore* (*natural language processing*). Qëllimi i kësaj fushe kërkimi është të krijojë sisteme që jo vetëm "shohin" dhe "kuptojnë" imazhin, por janë të afta të bashkëveprojnë me njerëzit me anë të një bisede kuptimplotë ku përgjigjet e sistemit të jenë të lidhura me kontekstin e bisedës.

Agjentët inteligjentë në dialogun vizual janë agjentë që kuptojnë dhe komunikojnë me njerëzit në gjuhë natyrore nën një kontekst vizual. Nga ana logjike, krahasuar me sistemet pyetje-përgjigje vizuale, sistemet e dialogut vizual janë një shkallë më lart në abstraksion dhe një hap më pranë inteligjencës artificiale të plotë.

Ndryshimi kryesor me sistemet pyetje-përgjigje vizuale është se këto të fundit përfaqësojnë vetëm një raund dialogu midis njeriut dhe makinës. Makina nuk mban të dhëna për pyetjet dhe përgjigjet e bëra më parë rreth imazhit. Kjo gjë e pengon atë të mbajë kontekstin e bisedës dhe t'i përgjigjet pyetjeve pasardhëse rreth të njëjtit imazh. Çdo pyetje konsiderohet nga sistemi si pyetja e parë dhe e vetme rreth një imazhi. Nëse bëhet një pyetje tjetër rreth të njëjtit imazh, sistemi e konsideron atë si një imazh dhe pyetje të re. Nga ana tjetër, modelet bashkëbiseduese (d.m.th. modelet e dialogut vizual) memorizojnë historikun e bisedës rreth një imazhi dhe mbajnë kontekstin e bisedës. Shpeshherë përdoruesi mund të përdorë përemra "*ai*", "*ajo*", "*atë*", etj. për t'ju referuar objekteve në imazh. Këto përemra janë burim dykuptimësish (*ambiguity*). Një model i mirë i dialogut vizual duhet të jetë në gjendje të arsyetojë në mënyrë të saktë që të zgjidhë problemet e dykuptimësisë dhe të vendosë përkatësinë e saktë midis objekteve në imazh dhe referencave të tyre.

Ekzistojnë përdorime të shumta të sistemeve të tilla. Ato variojnë nga mundësimi i përdoruesve për të komunikuar në gjuhë natyrore me asistentët inteligjentë personalë, ndihma e përdoruesve me probleme në shikim për të kuptuar ambjentin ku ndodhen, ndihma për analizimin dhe përpunimin e sasive të mëdha të informacionit video [70], [71], etj.

Edhe pse përdoret termi *dialog*, aktualisht keto sisteme janë gjendje të kthejnë vetëm përgjigje dhe jo të bëjnë pyetje. E njëjta gjë vlen edhe për agjentin e paraqitur në kapitullin 6. Krijimi i sistemeve inteligjente të cilat janë të afta të kenë një dialog të plotë me njerëzit është hapi i rradhës drejt inteligjencës së plotë artificiale.

2.6.1 Vlerësimi i Modeleve të Dialogut Vizual

Dialogu vizual qendron në mes të një spektri që varion nga *goal-oriented dialogue* (*dialog me objektiv*) [73] tek *goal-free dialogue* (*dialog pa objektiv*). Ai është i larguar mjaftueshëm nga një objektiv i caktuar dhe mund të përdoret si një test i përgjithshëm i inteligjencës artificiale [64]. Nga ana tjetër ai është i lidhur mjaftueshëm me imazhin në mënyrë që përgjigjet e tij individuale të mund të vlerësohen [64], [68]. Për rrjedhojë, për të vlerësuar një model të dialogut vizual

mund të përdoren *retrieval metrics* të cilat vlerësojnë aftësinë e tij për përzgjedhjen e një përgjigjeje të saktë nga një bashkësi përgjigjesh. Disa nga metrikat që mund të përdoren janë: *renditja mesatare reciproke (mean reciprocal rank),* dhe *recall@k.*

Mean Reciprocal Rank. Një metrikë që mund të përdoret për vlerësimin e sistemeve të dialogut vizual është *mean reciprocal rank* (MRR). MRR është renditja mesatare reciproke e përgjigjes njerëzore. Supozojmë se përgjigjet që gjenerohen nga sistemi inteligjent janë në formën e një renditjeje (d.m.th. shpërndarjeje probabilitare) ku përgjigja e renditur e para është përgjigja për të cilën sistemi ka sigurinë më të lartë (d.m.th. vlerën më të lartë të probabilitetit) për të qenë e saktë. Përgjigjet më poshtë në renditje kanë probabilitet më të ulët. Supozojmë gjithashtu se përgjigjet e sakta për pyetjet e *dataset*-it të testimit janë gjeneruar nga njerëzit. Çdo përgjigje e gjeneruar nga sistemi vlerësohet në bazë të renditjes reciproke të përgjigjes së saktë (d.m.th. të gjeneruar nga njerëzit). Për shembull, nëse sistemi ka kthyer 5 përgjigje nga të cilat 3 të parat janë të pasakta, përgjigja e saktë e renditur më lartë është përgjigja e 4-rt. Renditja reciproke (*reciprocal rank*) është inversi i pozicionit të përgjigjes së parë të saktë. Për këtë përgjigje *reciprocal rank* do të ishte ¼. Në rastet kur asnjë përgjigje nuk është e saktë, *reciprocal rank* është zero. Në rastet kur përgjigja e parë është e saktë *reciprocal rank* është 1. MRR është vlera mesatare e *reciprocal rank* për të gjitha pyetjet e *dataset*-it. Më formalisht, për një sistem i cili gjeneron një renditje të përgjigjeve për një *dataset* testimi të përbërë nga *N* pyetje, MRR përcaktohet si:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{2.1}$$

ku $rank_i$ është pozicioni i përgjigjes së parë të saktë.

Recall @ k. Një metrikë tjetër që mund të përdoret për vlerësimin e sistemeve të dialogut vizual është *Recall @ k*. Kjo metrikë shprehet në përqindje dhe mat numrin mesatar të përgjigjeve të sakta në *k* përgjigjet e renditura të para nga sistemi. Zakonisht variantet më të përdorura të kësaj metrike janë *recall @1, recall @5* dhe *recall @10*. Për një sistem të dialogut vizual, një vlere 40% e *recall@5* nënkupton se 40% e përgjigjeve të sakta të renditura nga sistemi ndodhen brenda grupit të 5 përgjigjeve të renditura të para.

Me rritjen e numrit të përgjigjeve të marra në konsideratë (d.m.th. me rritjen e parametrit *k*) rritet dhe vlera e *recall @k* sepse rritet probabiliteti që përgjigja e saktë të gjendet në *k* përgjigjet e para të renditura nga sistemi.

3

Bazat Teorike të Deep Learning

Ky kapitull paraqet bazat e nevojshme teorike dhe teknike të *machine learning* dhe rrjetave neurale artificiale. Materiali i paraqitur është bazuar në [35], [36], [37], [38], [39], [79], [80] dhe [81].

3.1 Machine Learning

Machine learning i mundëson kompjuterave të mësojnë nga eksperienca pa u programuar në mënyrë eksplicite. Çdo problem i machine learning mund të klasifikohet në tre kategori kryesore: *supervised learning* (të mësuarit e mbikqyrur), *unsupervised learning* (të mësuarit e pambikqyrur) dhe *reinforcement learning* (të mësuarit e përforcuar).

Në rastin e supervised learning, është dhënë *dataset*-i dhe dihet vlera e saktë që duhet të gjenerohet në dalje. Problemet e supervised learning kategorizohen në probleme të *regresionit* dhe *klasifikimit*. Në një problem regresioni, detyra e sistemit është të parashikojë në dalje rezultate brenda një vlere të vazhdueshme. Kjo do të thotë se duhet të vendoset një korrespondencë midis variablave të hyrjes dhe një funksioni të vazhdueshëm në dalje. ***Shembull:*** Duke pasur si të dhënë fotografinë e një personi, duhet të parashikohet mosha e tij. Në rastin e problemeve të klasifikimit, duhet të parashikohen rezultate në dalje që kanë një vlerë diskrete. Me fjalë të tjera, duhet të vendoset nje korrespondencë midis variablave të hyrjes dhe kategorive të veçanta në dalje. ***Shembull:*** Duke pasur si të dhënë fotografinë e një kafshe, duhet të parashikohet nëse është qen apo mace.

Unsupervised learning mundëson zgjidhjen e problemeve duke e ditur pak ose aspak se cili duhet të jetë rezultati i saktë në dalje. Me anë të *unsupervised learning* mund të nxirret një strukturë nga të dhëna të pastrukturuara në mënyrë eksplicite, ose të përcaktohet shpërndarja e të dhënave. Kështu, struktura mund të nxirret duke zbuluar grupe të shembujve të ngjashëm brenda të dhënave. Në *unsupervised learning* nuk ka asnjë feedback në lidhje me saktësinë e rezultateve të parashikuara në dalje. ***Shembull:*** Duke patur si të dhënë 1.000.000 gjene të ndryshme sistemi duhet të gjejë një mënyrë për grupimin automatik të këtyre gjeneve në grupe që janë të ngjashme ose të lidhura me njëra-tjetrën, si për shembull jetëgjatësia, vendndodhja, rolet dhe kështu me radhë.

Në rastin e *reinforcement learning*, sistemi ndërvepron me një mjedis dinamik në të cilin duhet të kryejë një detyrë të caktuar (të tillë si drejtimi i një automjeti ose luajtja e një loje kundër një kundërshtari). Sistemit i jepet feedback në formën e shpërblimeve apo ndëshkimeve ndërkohë që është duke kryer detyrën e caktuar, por pa i treguar në mënyrë eksplicite se ku gaboi apo se si duhet të veprojë. Qëllimi i sistemit është të maksimizojë shpërblimin në total. Disa përdorime të *reinforcement learning* janë makinat autonome (*autonomous driving*) dhe lojërat *Atari*[4] ku ndërveprimi i rradhës i sistemit me mjedisin varet nga shumë faktorë dhe nuk ka një mënyrë të realizueshme apo të lehtë për të programuar këtë ndërveprim.

3.1.1 Supervised Learning

Në supervised learning, qëllimi është të gjendet një funksion $\boldsymbol{f}$: X → Y, ku X është hapësira e variablave në hyrje dhe Y është hapësira e parashikimeve në dalje [35]. Funksioni $\boldsymbol{f}$ quhet *hipoteza* dhe duhet të jetë një parashikues i "mirë" i vlerës korresponduese Y. Duke patur një *dataset* me *n* shembuj të pavarur trajnimi $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ ne mund të matim saktësinë e hipotezës duke përdorur një funksion skalar gabimi $L(\hat{y}, y)$, i quajtur ndryshe *funksion humbjeje* ose *funksion kostoje*. Ky funksion mat ndryshimin midis vlerës së parashikuar $\hat{y}$ dhe vlerës së saktë

[4] www.atari.com

y. Qëllimi i *machine learning* është gjetja e një funksioni hipotezë që minimizon humbjen (gabimin).

Vlera totale e humbjes llogaritet si mesatarja e humbjeve individuale për çdo shembull trajnimi në *dataset* dhe paraqitet në ekuacionin 3.1. Në këtë ekuacion, me *m* shënohet numri total i shembujve të trajnimit:

$$L = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 \tag{3.1}$$

Ky funksion quhet ndryshe dhe "funksioni i katrorit të gabimit" (*squared error function*) ose "katrori i gabimit mesatar" (*mean squared error*) dhe është një nga funksionet e gabimit më të përdorura në machine learning.

Në rastin e *regresionit linear* $f = wx + b$, ku *w* janë parametrat mbi të cilët do të optimizohet funksioni *f*, dhe *b* është vlera e *bias*. Funksioni i humbjes shprehet si:

$$L = \frac{1}{2m}\sum_{i=1}^{m}(wx_i + b\ - y_i)^2 \tag{3.2}$$

Në rastin e *klasifikimit*, funksioni i hipotezës duhet të klasifikojë të dhënat në hyrje në disa kategori diskrete në dalje. Një praktikë e zakonshme është përdorimi i funksionit *softmax* për gjenerimin e parashikimeve. Ky funksion merr në hyrje një vektor *x* dhe gjeneron në dalje një vektor *p* me të njëjtën përmasë, ku $p_i = \frac{e^{z^{(i)}}}{\sum_{j=0}^{k} e^{z m^{(i)}}}$ për një shembull trajnimi dhe $z = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{l=0}^{m} w_l x_l$. Vektori *p* është i tillë që elementët e tij kanë vlerë midis 0 dhe 1 dhe shuma e tyre është 1. Me fjalë të tjera, funksioni softmax gjeneron një shpërndarje probabilitare të parashikimeve për secilën kategori në varësi të të dhënave në hyrje.

Funksioni i humbjes më i përdorur në rastet e *klasifikimit* është *cross-entropy loss* (entropia e kryqëzuar) dhe formulohet si:

$$L = -\sum_{k=1}^{K} y_k \log \hat{y}_k = -\log \hat{y}_{y=1} \tag{3.3}$$

3.1.2 Overfitting dhe Underfitting

Sfida kryesore e një algoritmi të *machine learning* është se ai duhet të ketë performancë të mirë parashikimi (d.m.th. saktësi) *për të dhëna të cilat nuk i ka vrojtuar më parë*, jo vetëm për të dhënat me të cilat është trajnuar. Aftësia për të patur performancë të mirë për të dhëna të pa vrojtuara më parë quhet *gjeneralizim* (*generalization)* [39].

Gjatë trajnimit të një *modeli* (d.m.th. sistemi të krijuar për të kryer një detyrë të caktuar duke përdorur një apo disa algoritma të caktuar) në *machine learning* përdoret një *dataset* trajnimi mbi të cilin llogaritet një gabim i quajtur *gabimi i trajnimit*. Qëllimi i trajnimit është reduktimi i këtij gabimi [39]. Ekziston gjithashtu një lloj tjetër gabimi i quajtur *gabimi i gjeneralizimit*, i njohur ndryshe si *gabimi i testimit*. Gabimi i testimit llogaritet mbi një *dataset* testimi. *Dataset*-i i testimit përmban të dhëna të cilat janë mbledhur nga i njëjti burim me të dhënat e trajnimit, por të veçuara prej tyre (d.m.th. të dhënat e këtyre *dataset*-eve janë të ndryshme, por janë mbledhur tek i njëjti burim të dhënash). Gjatë trajnimit, merren të dhëna nga *dataset*-i i trajnimit, ndryshohen parametrat e modelit për të zvogëluar gabimin e trajnimit dhe më pas modeli testohet mbi *dataset*in e testimit. Gabimi i testimit pritet të jetë më i madh ose i barabartë me gabimin e trajnimit. Në këtë kontekst, një algoritëm i mirë i *machine leraning* është një algoritëm i cili ka aftësi të mirë të a) zvogëlojë gabimin e trajnimit, b) zvogëlojë diferencën midis gabimit të trajnimit dhe gabimit të testimit [39].

Si pasojë e këtyre karakteristikave të dëshiruara, rrjedhin problemi i *underfitting* dhe *overfitting*.

Underfitting ndodh kur modeli nuk është i aftë të përshtatet mirë me *dataset*-in dhe të *përshkruajë* mirë një pjesë të mirë të të dhënave të tij. Si pasojë e *underfitting*, modeli nuk është i aftë të arrijë një gabim trajnimi mjaftueshëm të vogël në *dataset*-in e trajnimit. *Overfitting* ndodh kur modeli është përshtatur më shumë seç duhet me të dhënat e *dataset*-it të trajnimit dhe nuk është i aftë për të *gjeneralizuar* mirë për të dhëna të pa vrojtuara më parë (d.m.th *dataset*-in e testimit dhe të dhënat e "botës reale"). Si pasojë e *overfitting* diferenca midis gabimit të trajnimit dhe gabimit të

testimit është shumë e madhe. Në figurën 3.1 ilustrohen grafikisht problemet e *underfitting* dhe *overfitting*. Të dhënat e *dataset*-it të trajnimit janë të paraqitura si pika me ngjyrë blu ndërsa parashikimi i algoritmit (me ngjyrë të kuqe) është paraqitur si një funksion matematikor që përshkruan të dhënat. Sa më pak variabla (*features*) të përdoren për të gjeneruar këtë funksion aq më shumë shkohet drejt *underfitting* (grafiku A). Sa më shumë *features* të përdoren për të gjeneruar këtë funksion aq më shumë shkohet drejt overfitting (grafiku C). Gjetja e funksionit të duhur që përshkruan të dhënat është sfida kryesore e *machine learning*.

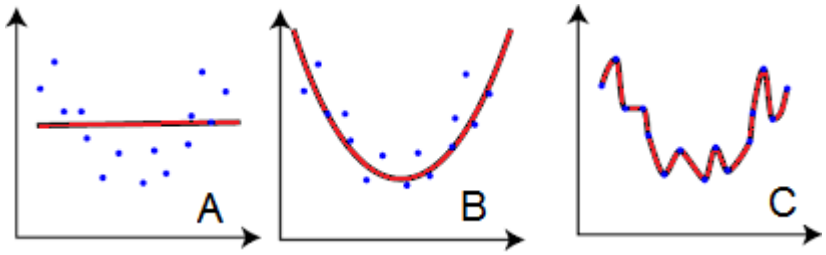


Figura 3.1. **Underfitting, overfitting dhe gjeneralizimi i mirë. Përshtatur nga [81].** Në grafikun A kemi situatën e *underfitting* ku modeli nuk është në gjendje të përshkruajë një pjesë të mirë të të dhënave. Në grafikun C kemi situatën e *overfitting* ku modeli është përshtatur me të dhënat e *dataset*-it të trajnimit dhe ka probabilitet të lartë të mos *gjeneralizojë* mirë për të dhëna të pa vrojtuara më parë. Në grafikun B kemi rastin optimal ku modeli është përshtatur mjaftueshëm me të dhënat e *dataset*-it të trajnimit sa për të përfaqësuar pjesën më të madhe të tyre, por jo të gjitha. Në këtë rast ai ka probabilitet të lartë për të *gjeneralizuar* mirë për të dhëna të pa vrojtuara më parë.

Konsiderata praktike. Në praktikë përdoret edhe një *dataset* tjetër i quajtur *dataset-i i validimit*. Ky *dataset* është i ngjashëm me *dataset*-et e trajnimit dhe testimit (d.m.th. të dhënat janë mbledhur nga i njëjti burim por janë të ndryshme nga dy *dataset*-et e tjera). *Dataset*-i i validimit përdoret gjatë trajnimit të modelit për të parë si ky i fundit do të gjeneralizojë për të dhëna të pa vrojtuara më parë. Fillimisht bëhet trajnimi i modelit me *dataset*-in e trajnimit. Trajnimi bëhet për disa *epoka*. *Një epokë* përfundon kur modeli ka përdorur të gjitha të dhënat e *dataset*-it të trajnimit për algoritmin e të mësuarit. Më pas modeli testohet me *dataset*-in e validimit për të parë gjeneralizimin e tij. Nëse gjeneralizimi nuk është në nivelet e dëshiruara trajnimi vazhdon akoma dhe vazhdimisht kryhet testimi me *dataset*-in e validimit. Pasi *gjeneralizimi* i modelit arrin nivelet e dëshiruara bëhet testimi i modelit duke përdorur

dataset-in e testimit. *Dataset*-et e trajnimit dhe të validimit përdoren disa herë gjatë trajnimit të modelit ndërsa *dataset*-i i testimit përdoret vetëm një herë gjatë testimit përfundimtar të modelit.

3.1.3 Teorema "No free lunch"

Teorema "*No free lunch"* [40] për *machine learning* thotë se, bazuar mesatarisht në të gjithë *data-generating distributions* (*shpërndarjet gjeneruese të të dhënave*, d.m.th. mbi të gjithë burimet e mundshme të të dhënave), asnjë algoritëm i *machine learning* nuk është universalisht më i mirë se të tjerët dhe se çdo algoritëm klasifikimi ka të njëjtin gabim kur klasifikon të dhëna të cilat nuk i ka vrojtuar më parë [39]. Kjo do të thotë se asnjë algoritëm i *machine learning* nuk mund të jetë automatikisht dhe universalisht i aplikueshëm për të gjitha problemet e mundshme (d.m.th. përdorimet e mundshme) dhe për të gjitha të dhënat e mundshme.

Hapësira e të gjitha problemeve të mundshme përmban të dhëna kaotike dhe që nuk kanë lidhje me njëra-tjetrën. Në kontekstin e teoremës "*no free lunch*", qëllimi i *machine learning* nuk është të krijojë një algoritëm gjithëpërfshirës i cili të jetë i aplikueshëm për të gjitha problemet e mundshme, por një algoritëm i cili të mund të zgjidhë një problem të caktuar të "botës reale" duke u bazuar ne të dhënat të cilat janë specifikë për këtë problem.

3.1.4 Rregullarizimi

Rregullarizimi (*regularization*) është çdo ndryshim që i bëhet një algoritmi të të mësuarit që ka për qëllim të reduktojë gabimin në *gjeneralizim*, por jo në trajnim. Me anë të rregullarizimit mund të mbajmë nën kontroll tendencën e një modeli për të patur *underfit* ose *overfit*. Në rastin e figurës 3.1, supozojmë se funksioni që modeli gjen për të përshkruar të dhënat në grafikun C është polinomi:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2 + \theta_3 x_3^3 + \theta_4 x_4^4 + \theta_5 x_5^5 \qquad (3.4)$$

ku θ janë parametrat e modelit dhe x janë *features*.

Një alternativë për të zvogëluar *overfitting* është zvogëlimi i shkallës së këtij polinomi duke penalizuar θ_3, θ_4 dhe θ_5. Në këtë mënyrë tre termat e fundit të polinomit do kenë shumë pak ndikim në vlerën *y* dhe grafiku i polinomit do shkojë drejt formës në grafikun B. Për ta bërë këtë, përdorim një parametër λ i cili quhet parametri i rregullarizimit (*regularization parameter*). Supozojmë se funksioni i humbjes për këtë model është sipas ekuacionit 3.1. Pas rregullarizimit të parametrave θ_3, θ_4 dhe θ_5, ky funksion bëhet:

$$L = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y_i} - y_i)^2 + \lambda\sum_{j=3}^{5}{\theta_j}^2\right] \tag{3.5}$$

Parametri λ zgjidhet me një vlerë të madhe (p.sh.1000). Kjo bën që gjatë trajnimit, në mënyrë që të zvogëlohet ndikimi (tashmë i madh) i θ_3, θ_4 dhe θ_5 modeli ndryshon në mënyrë të detyruar këto parametra në një masë më të madhe se të tjerët duke i shumëzuar me një numër shumë afër zeros.

3.2 Optimizimi

Qëllimi i supervised learning është optimizimi i funksionit të humbjes për të gjetur parametra *w* dhe *bias b* të tillë që sistemi të *gjeneralizojë* në mënyrë të saktë për të dhëna të pa vrojtuara më parë. Një nga mënyrat që përdoret më shumë në praktikë për të realizuar këtë qëllim është optimizimi sipas gradientit (*gradient descent*). *Gradient descent* është një algoritëm optimizimi për gjetjen e minimumit të një funksioni duke përdorur gradientin e tij.

Në rastin e *supervised learning*, gradienti i funksionit të humbjes është vektori i derivateve të pjesshme sipas parametrave të optimizimit *w* (për lehtësi të shprehjes së konceptit të *gradient descent*, nuk po marrim në konsideratë bias *b*) dhe tregon pjerrësinë e kurbës së këtij funksioni përgjatë dimensioneve të *w*. Optimizimi do të ketë përfunduar me sukses kur funksioni i humbjes të jetë në vlerën e tij minimale (figura 3.2).

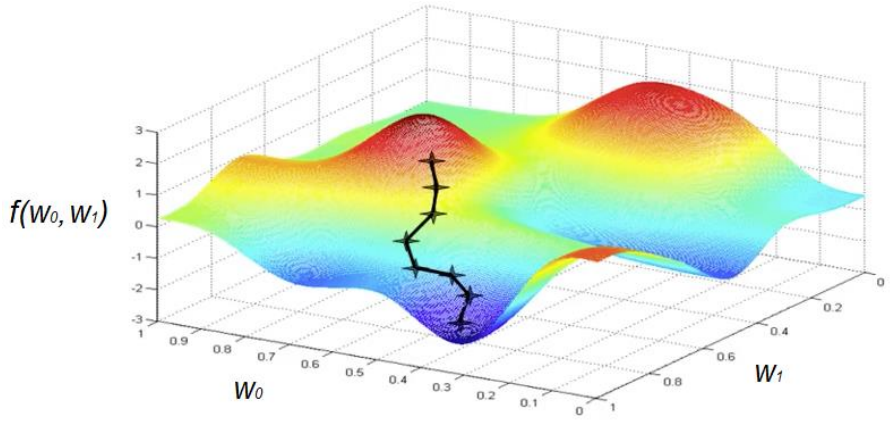


Figura 3.2. **Gradient descent. Përshtatur nga [37].** Me $f(w_0, w_1)$ është shënuar funksioni i humbjes. Për lehtësi të shprehjes së konceptit të *gradient descent* supozojmë se funksioni i humbjes parametrizohet vetëm nga w_0 dhe w_1. Qëllimi i trajnimit është gjetja e parametrave w për të cilët funksioni i humbjes ka vlerën minimale. Kjo gjë arrihet nga algoritmi *gradient descent* me anë të lëvizjes nëpër kurbën e funksionit të humbjes, duke përdorur tangenten këtij funksioni si drejtim të lëvizjes dhe *shkallën e të mësuarit* α si përmasë të hapit të lëvizjes.

Optimizimi realizohet duke përdorur derivatin sipas parametrave të optimizimit w (d.m.th. tangentja e funksionit të humbjes) si drejtim të lëvizjes në kurbën e funksionit të humbjes (figura 3.2). Lëvizja në këtë kurbë bëhet me hapa sipas drejtimit të pjerrësisë më të madhe. Në bazë të kësaj lëvizjeje bëhet ndryshimi i parametrave w duke i shtuar atyre një vlerë të vogël të drejtimit të kundërt të gradientit. Kjo vlerë parametrizohet nga *shkalla e të mësuarit* (*learning rate*) α e cila përcakton përmasën e secilit hap.

Shkalla e të mësuarit α është një parametër shumë i rëndësishëm në *machine learning*. Një vlerë e madhe e tij do të rezultonte në një hap të madh në kurbën e funksionit të humbjes. Kjo mund të çonte në divergjimin e optimizimit pasi minimumi i funksionit të humbjes mund të tejkalohej dhe hapat pasardhës mund të sillnin largimin gjithmonë e më shumë nga ky minimum. Nga ana tjetër, një vlerë e vogël do të rezultonte në një hap të vogël në kurbën e funksionit të humbjes. Kjo gjë do të vononte konvergjimin e optimizimit pasi hapi me të cilin do afroheshim drejt minimumit do të ishte i vogël. Në figurën 3.3 ilustrohen këto 2 raste problematike.

Konsiderata praktike. Siç ndodh shpeshherë me parametrat në *machine learning*, nuk ekziston një vlerë e përcaktuar e parametrit α që do të garantonte konvergjimin e algoritmit të optimizimit. Përcaktimi i këtij parametri bëhet në mënyrë empirike duke zgjedhur një vlerë mjaftueshëm të vogël që mos sjellë divergjim, por mjaftueshëm të madhe qe mos sjellë vonesa në procesin e konvergjimit. Përgjithësisht, në praktikë parametri α zvogëlohet gradualisht deri në një limit të pranueshëm që sjell konvergjim pa krijuar vonesa të konsiderueshme. Ky limit gjithashtu përcaktohet në mënyrë empirike.

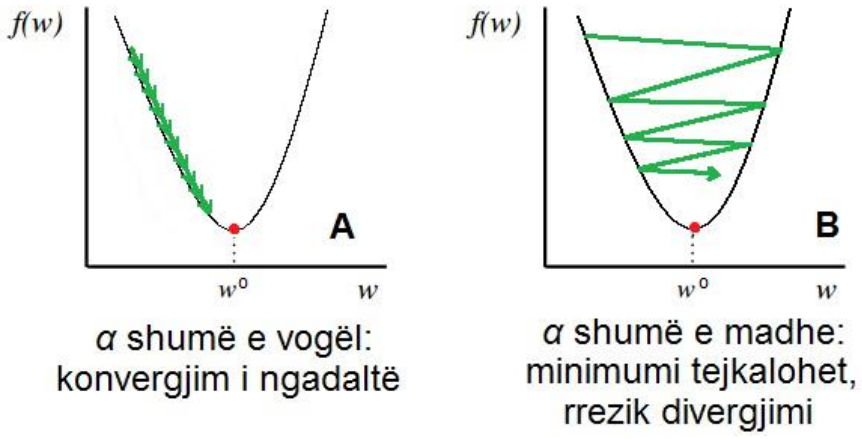


Figura 3.3. **Gradient descent në rastet e konvergjimit të vonuar dhe divergjimit. Përshtatur nga [37].** Me *f(w)* është shënuar funksioni i humbjes ndërsa me w^o është shënuar vlera optimale e parametrave *w* që sjellin në konvergjimin e optimizimit. Në rastin A, një shkallë e të mësuarit të vogël rezulton në hapa më të vegjël në kurbën e funksionit të humbjes dhe për rrjedhojë në vonesën e konvergjimit. Në rastin B, një shkallë e të mësuarit e madhe rrezulton në hapa të mëdha në kurbën e funksionit të humbjes dhe për rrjedhojë minimumi tejkalohet dhe rrezikohet divergjimi.

Hiperparametrat. Hiperparametrat (*hyperparameters*) janë parametra të një algoritmi të *machine learning* të cilat përdoren për të kontrolluar sjelljen e tij. Vlerat e hiperparametrave nuk duhen ngatërruar me parametrat *w* që kontrollohen dhe mësohen nga vetë algoritmi. Hiperparametrat janë parametra, vlerat e të cilave vendosen nga projektuesi i algoritmit dhe algoritmi nuk ka asnjë kontroll mbi to. Shembuj hiperparametrash janë parametri i rregullarizimit λ në ekuacionin 3.5 dhe shkalla e të mësuarit α.

3.2.1 Optimizimi Stokastik Sipas Gradientit

Algoritmi standard i *gradient descent* përdor të gjithë *dataset*-in për të llogaritur gradientin e funksionit të humbjes. *Dataset*-et e përdorura në praktikë përmbajnë një numër shumë të madh shembujsh trajnimi (ka raste me miliona). Kjo gjë mund të kërkojë shumë fuqi llogaritëse dhe të vonojë procesin e konvergjimit. Për këtë arsye, për llogaritjen e gradientit, në çdo hap përdoret vetëm një pjesë e vogël e *dataset*-it. Ky proces njihet si optimizim stokastik sipas gradientit (*stochastic gradient descent).* Numri i shembujve të *dataset*-it që përdoret në çdo hap njihet me termin *batch* dhe përcaktohet në bazë të fuqisë së disponueshme llogaritëse. *Stochastic gradient descent* (SGD) lejon kryerjen e ndryshimeve të shpeshta të përafruara të parametrave *w* në vend të ndryshimeve të sakta më të rralla. Në praktikë, SGD është një algoritëm që funksionon shumë mirë në shumicën e aplikimeve duke mundësuar gjithashtu edhe përdorim efikas të burimeve llogaritëse.

3.3 Algoritmi backpropagation

Llogaritja e gradientit të funksionit të humbjes mundëson optimizimin e tij dhe gjetjen e parametrave të përshtatshëm *w* dhe bias *b* të cilët do të mundësonin gjenerimin e parashikimeve të sakta $\hat{y}$ për të dhëna në hyrje *x* të panjohura më parë. Një algoritëm për llogaritjen e këtij gradienti është *backpropagation*. Në rrjetat neurale artificiale moderne, ky algoritëm mund ta bëjë trajnimin me *gradient descent* deri në 10 milion herë më të shpejtë sesa një implementim naiv i gjetjes së gradientit. Kjo përkthehet në zvogëlimin e kohës së trajnimit të një modeli nga 200,000 vjet në 1 javë [38].

Algoritmi backpropagation u soll në vëmendjen e komunitetit të *machine learning* në 1986 nga David Rumelhart, Geoffrey Hinton dhe Ronald Williams [41] dhe sot është zemra e të mësuarit në rrjetet neurale artificiale. Ai është një aplikim rekursiv i *rregullit zinxhir* (*chain rule*). Në zemër të tij është derivati i pjesshëm $\partial L/\partial w$ i funksionit të humbjes *L* në lidhje me parametrat *w* (dhe bias *b*). Ky derivat na tregon sa shpejt ndryshon humbja kur ndryshojmë parametrat *w* dhe bias *b*. Ai na tregon gjithashtu se si ndryshimi i *w* dhe *b* ndryshon sjelljen e përgjithshme të rrjetit

neural. Për të shpjeguar mënyrën e funksionimit të këtij algoritmi duhet të paraqesim më parë konceptin e *grafit llogaritës* (*computational graph*). Supozojmë se kemi shprehjen $y = (5x + 1)^2$ dhe duam të llogaritim gradientin $\partial y/\partial x$. Për të lehtësuar veprimet, përdorim disa variabla të ndërmjetëm: $a = 5x$, $b = a + 1$ dhe $y = b^2$. Për të krijuar *grafin llogaritës* të funksionit *a,* vendosim secilin veprim dhe variablat nëpër nyje si në figurën 3.4. Nyjet lidhen me një shigjetë me njëra-tjetrën kur vlera në dalje e një nyjeje shërben si hyrje për një nyje tjetër.

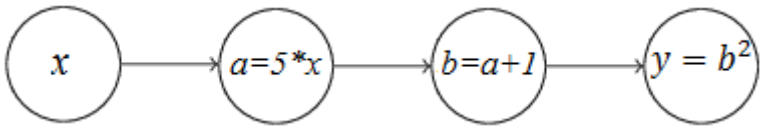


Figura 3.4. Grafi llogaritës i funksionit $y = (5x + 1)^2$. Për ta ndërtuar grafin përdorim variablat e ndërmjetëm $a = 5x$, $b = a + 1$ dhe $y = b^2$. Secilin veprim dhe variablat i vendosim nëpër nyje.

Gradienti i çdo veprimi në lidhje me variablin e tij në hyrje është: $\partial a/\partial x = 5$, $\partial b/\partial a = 1$, dhe $\partial y/\partial b = 2b$. Duke ditur gradientët e ndërmjetëm, gradienti $\partial y/\partial x$ llogaritet duke përdorur rregullin zinxhir: $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial b} * \frac{\partial b}{\partial a} * \frac{\partial a}{\partial x}$.

Në një model real të rrjetave neurale, *x* është vektori që përmban të dhënat në hyrje dhe parametrat e rrjetit dhe *y* është humbja totale e rrjetit. Parametrat e rrjetit janë ato që na interesojnë dhe gradienti i tyre na tregon se si duhet t'i modifikojmë këto parametra në mënyrë që të zvogëlojmë humbjen. Në shumë implementime në praktikë, rrjeti neural konsiderohet si një graf aciklik i drejtuar (*directed acyclic graph)* veprimesh. Në këtë graf, vektorët e të dhënave paraqiten si brinjë ndërsa nyjet përbëjnë funksione matematikore të diferencueshme. Këto funksione përdorin si input vektorë dhe i kombinojnë ato në një vektor të vetëm i cili dërgohet drejt nyjeve të tjera [35]. Llogaritjet e gradientit realizohen nëpërmjet një objekti graf (*graph)* i cili përmban nyjet dhe lidhjet midis tyre. Objekti graf dhe çdo nyje përmbajnë funksionet *forward()* dhe *backward().* Funksioni *forward()* iteron mbi të gjitha nyjet e grafit dhe çdo nyje llogarit output-in e saj. Funksioni *backward()* iteron mbi nyjet sipas rendit të kundërt topologjik dhe çdo nyjeje i jepet si input gradienti i humbjes në lidhje me outputin e saj. Në bazë të këtij gradienti, nyja llogarit gradientët sipas secilës prej

hyrjeve të saj. Ky gradient tejçohet më pas në nyjen paraardhëse duke përdorur *chain rule*, e kështu me rradhë deri në nyjet e input-it të grafit.

Figura 3.5 ilustron konceptin e *backpropagation* për grafin e funksionit në figurën 3.4.

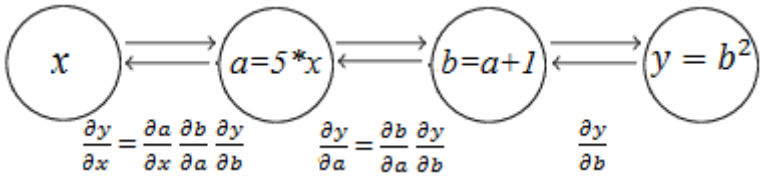


Figura 3.5. Bakpropagation për funksionin $y = (5x + 1)^2$. Gradienti tejçohet në drejtim të kundërt (d.m.th. nga output-i drejt input-it) në mënyrë rekursive duke aplikuar rregullin zinxhir për të gjetur influencën që hyrja dhe hapat e ndërmjetëm kanë mbi daljen. Sipas këtij rregulli, duke ditur *∂y/∂b* mund të llogaritim *∂y/∂a = ∂b/∂a*∂y/∂b* dhe *∂y/∂x=∂a/∂x*∂b/∂a*∂y/∂b*.

Në aplikimet në praktikë të rrjetave neurale, për të realizuar hapin *forward()*, merret një pjesë e të dhënave të trajnimit (*batch* me *m* shembuj trajnimi) $\{(x_i, y_i)\}_{i=1}^{m}$ dhe parametrat aktualë *w* dhe llogariten të gjitha vlerat e ndërmjetme të rrjetit së bashku me vlerën e humbjes. Më pas, gjatë backpropagation (d.m.th. në hapin *backward()*) procedohet nga output-i drejt input-it nëpër të gjitha hapat e ndërmjetme duke i bashkuar gradientin lokal gradientit global nëpërmjet rregullit zinxhir.

3.4 Rrjetat Neurale Artificiale

Një Rrjet Neural Artificial (RNA) është një model kompjuterik i frymëzuar nga mënyra se si rrjetet nervore biologjike të trurit të njeriut përpunojnë informacionin. Rrjetat neurale artificiale kanë mundësuar një progres të madh në *machine learning* falë rezultateve të arritura në përpunimin e tekstit, vizionit kompjuterik (*computer vision)* dhe njohjes së zërit (*speech recognition*).

3.4.1 Neuroni

Njësia bazë e përpunimit në një rrjet nervor është neuroni, shpeshherë i quajtur *nyje*. Neuroni merr input *x* nga disa nyje të tjera, ose nga një burim i jashtëm dhe llogarit një dalje *y*. Secili input ka një peshë *w*, e cila përcaktohet në bazë të rëndësisë relative të tij ndaj inputeve të tjera. Nyja aplikon një funksion ***f*** mbi shumën e ponderuar të hyrjeve të saj siç tregohet në figurën 3.6:

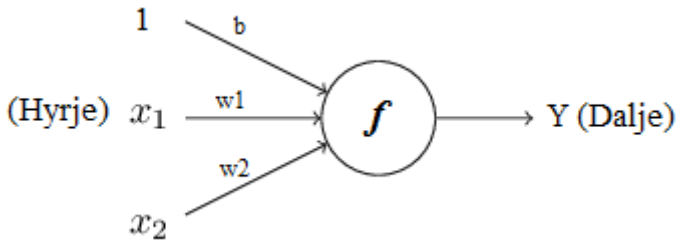


Figura 3.6. **Modeli logjik i një neuroni**. Dalja percaktohet si $Y = f(W_1x_1 + W_2x_2 + b)$

Neuroni i mësipërm merr input-e numerike x_1 dhe x_2 dhe peshat W_1 dhe W_2 respektivisht për secilën nyje. Përveç kësaj, ka një tjetër input 1 me peshën *b* të quajtur *bias*. Funksioni kryesor i bias është t'i ofrojë çdo nyjeje një vlerë konstante të trajnueshme (përveç hyrjeve normale që merr nyja). Bias lejon zhvendosjen e funksionit të aktivizimit në të majtë ose në të djathtë, gjë që mund të jetë kritike për mundësinë e trajnimit të rrjetit. Qëllimi i trajnimit të rrjetit neural është optimizimi i parametrave *w* në lidhje me një funksion humbjeje të diferencueshëm duke përdorur SGD.

Dalja *Y* nga neuroni llogaritet siç tregohet në figurën 3.6. Funksioni ***f*** është jo-linear dhe quhet *funksioni i aktivizimit*. Qëllimi i *funksionit të aktivizimit* është të mundësojë jo-linearitet në daljen e një neuroni. Prania e jo-linearitetit është e rëndësishme sepse shumica e të dhënave në problemet reale nuk janë lineare dhe ne duam që neuronet të mësojnë këto paraqitje jo lineare. Çdo *funksion aktivizimi* (ose jo-linearitet) merr një numër në hyrje dhe aplikon një funksion të caktuar matematikor mbi të. Funksionet e aktivizimit që hasen më shpesh në praktikë janë:

- **Sigmoid:** merr në hyrje një numër real dhe e transformon atë në nje vlerë midis 0 dhe 1. $\sigma(x) = 1 / (1 + e^{-x})$

- **Tanh:** merr në hyrje një numër real dhe e transformon atë në nje vlerë midis -1 dhe 1. $\tanh(x) = 2\sigma(2x) - 1$
- **ReLU:** ReLU është shkurtimi i *Rectified Linear Unit*. Ky funksion aktivizimi merr në hyrje një numër real. Vlerat negative i zëvendëson me zero ndërsa vlerat pozitive i lë të pandryshuara. $f(x) = max(0, x)$

Figura 3.7 tregon secilin nga funksionet e mësipërme të aktivizimit.

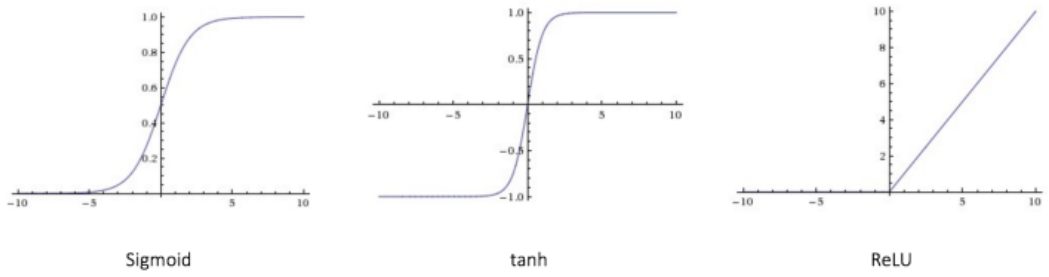


Figura 3.7. Funksione të ndryshme aktivizimi [79]

3.4.2 Feedforward Neural Network

Rrjeti neural *feedforward* është tipi i parë dhe më i thjeshtë i krijuar i rrjetit neural artificial. Ai permban një sërë neuronesh (nyjesh) të grupuara në shtresa. Nyjet e shtresave ngjitur janë të lidhura me njëra-tjetrën. Të gjitha këto lidhje kanë pesha individuale. Një shembull i një rrjeti të tillë tregohet në figurën 3.8:

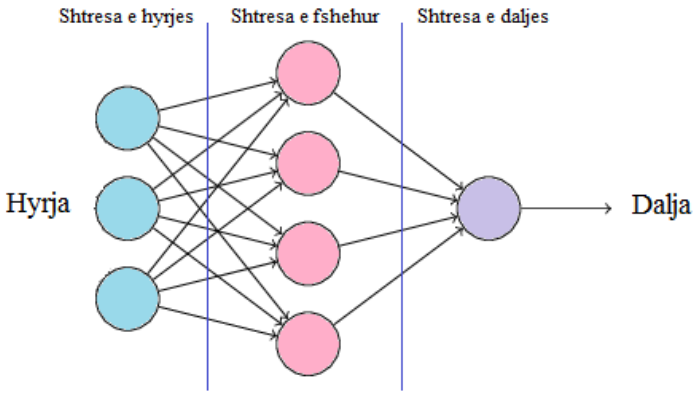


Figura 3.8. Shembull i një rrjeti neural feedforward

Një rrjet neural feedforward përbëhet nga tre tipe nyjesh:

Nyjet hyrëse - Nyjet hyrëse i sigurojnë rrjetit informacion nga bota e jashtme dhe së bashku përbëjnë atë që quhet *shtresa hyrëse.* Asnjë llogaritje nuk kryhet në asnjë nga nyjet hyrëse - ato thjesht kalojnë informacionin tek nyjet e fshehura.

Nyjet e fshehura - Nyjet e fshehura nuk kanë lidhje të drejtpërdrejtë me botën e jashtme (prandaj quhen "të fshehura"). Ato kryejnë llogaritjet dhe transferojnë informacion nga nyjet hyrëse tek nyjet dalëse. Tërësia e nyjeve te fshehura formojnë *shtresën e fshehur*. Një rrjet feedforward ka vetëm një shtresë hyrëse dhe një shtresë të vetme dalëse, por mund të ketë zero ose shumë shtresa të fshehura. Nëse ka më shumë se një shtresë të fshehur quhet *rrjet neural i thellë* (*deep neural network*).

Nyjet dalëse - Nyjet dalëse së bashku formojnë *shtresën dalëse* dhe janë përgjegjëse për llogaritjet dhe transferimin e informacionit nga rrjeti neural në botën e jashtme.

Në një rrjet *feedforward*, informacioni lëviz vetëm në një drejtim (përpara) nga nyjet hyrëse, përmes nyjeve të fshehura (nëse ka) dhe tek nyjet dalëse. Nuk ka cikle në rrjet.

Dy shembuj rrjetesh feedforward janë:

Single Layer Perceptron (Perceptron me një shtresë të vetme) – Ky është rrjeti më i thjeshtë feedforward dhe nuk përmban shtresë te fshehur.

Multi Layer Perceptron (Perceptron me shumë shtresa) – Ky rrjet përmban një apo më shumë shtresa të fshehura. Ndërkohë që një perceptron me një shtresë të vetme mund të mësojë vetëm funksione lineare, një perceptron me shumë shtresa mund të mësojnë edhe funksione jolineare.

Në çdo rast, të gjitha nyjet e një shtrese janë të lidhura me të gjitha nyjet e shtresës paraardhëse dhe shtresës pasardhëse. Ky lloj organizimi quhet *rrjet i lidhur plotësisht* (*fully connected network*).

3.4.3 Rrjetat Neurale Konvolucionale

Rrjetat neurale konvolucionale (*Convolutional Neural Networks*) [42] janë rrjeta neurale të projektuara për të kryer llogaritje mbi të dhëna që kanë topologji hapësinore (p.sh. imazhe, video, zë). Input-i i këtyre rrjetave është një matricë

shumëdimensionale e quajtur *tensor*. Supozojmë se kemi një imazh me ngjyra me përmasa 256 x 256 piksela. Në këtë rast, input-i i rrjetit një është një *tensor* 256 x 256 x 3 dimensional (për 3 ngjyra: e kuqe, jeshile, blu). Për këtë imazh, dimensionaliteti është shumë i lartë (196,608 vlera numerike) dhe nuk do të ishte i leverdisshëm (në termat e numrit të parametrave të rrjetit dhe kohës së përpunimit) përdorimi i një rrjeti plotësisht të lidhur. Në këto raste përdoren rrjeta neurale konvolucionale të cilat ruajnë informacion mbi strukturën topologjike të të dhënave në hyrje dhe enkodojnë karakteristikat e input-it brenda akritekturës së tyre duke krijuar mundësinë e implementimit më efikas të funksionit *forward()* dhe duke zvogëluar numrin e parametrave të rrjetit.

Rrjetat neurale konvolucionale (CNN) janë shumë të ngjashme me rrjetat neurale "të zakonshme" të trajtuara më sipër: ato përbëhen nga neuronet që kanë pesha dhe bias që mund të mësohen. Problemet që këto lloj rrjetash synojnë të zgjidhin janë probleme klasifikimi. Secili neuron merr disa input-e, kryen një veprim matematikor linear mbi to (dhe opsionalisht një veprim jo-linear). I gjithë rrjeti përsëri shpreh një funksion të vetëm të diferencueshëm $\boldsymbol{f}$: nga pikselat e imazhit në hyrje në një klasifikim në dalje. Rrjeti përsëri ka një funksion humbjeje (p.sh. Softmax) në shtresën e fundit (plotësisht të lidhur) dhe mbi të aplikohen algoritmat dhe parimet e trajtuara më sipër për trajnimin e rrjetave neurale.

Në këtë disertacion rrjetet CNN janë përdorur për të përpunuar imazhe. Duke përfituar nga fakti që input-i është imazh, arkitektura e CNN-ve është e organizuar në mënyrë të tillë që të optimizohet numri i parametrave të përdorur. Neuronet e tyre janë të organizuara në 3 dimensione: *gjerësi, lartësi dhe thellësi*. Duke qenë se input-i i çdo shtrese është 3-dimensional, atij i referohemi si *vëllim në hyrje* (*input volume*). Thellësia i referohet dimensionit të tretë të vëllimit dhe jo thellësisë totale të rrjetit neural. Neuronet e çdo shtrese e transformojnë vëllimin në hyrje në një *vëllim aktivizimi* (*activation volume*). Duke qenë se këto rrjeta përdoren për probleme klasifikimi, në dalje, arkitektura e rrjetit e redukton imazhin në një vektor të vetëm që përmban shpërndarjen probabilitare për secilën klasë.

Një arkitekturë tipike CNN është e përbërë nga disa shtresa dhe secila prej tyre e transformon një vëllim aktivizimi në një tjetër nëpërmjet një funksioni të diferencueshëm. Ka 3 tipe kryesore shtresash: **shtresa konvolucionale, shtresa pooling, dhe shtresa e lidhur plotësisht (*fully connected*).**

Shtresa konvolucionale. Kjo është shtresa kryesore e një rrjeti CNN. Në këtë shtresë kryhen shumica e llogaritjeve. Kjo shtresë merr një *tensor* hyrjeje dhe gjeneron një *tensor* daljeje duke kryer *konvolucionin* e hyrjes me një bashkësi filtrash. Filtri është një *tensor* parametrash *w* dhe qëllimi i trajnimit të rrjetit CNN është mësimi i këtyre parametrave në mënyrë që të gjenerohet një klasifikim i saktë i imazhit në dalje. Filtrat trajnohen duke përdorur algoritmin *backpropagation*. Çdo filtër zë vetëm një pjesë të gjërësisë dhe lartësisë së vëllimit në hyrje, por shtrihet përgjatë gjithë thellësisë së tij. Për shembull, një filtër tipik në shtresën e parë të një CNN-je mund të jetë një *tensor* me përmasën 5 x 5 x 3 (d.m.th 5 elementë për pikselat në gjerësi dhe lartësi dhe 3 për çdo ngjyrë të imazhit). Këtë filtër mund ta zhvendosim (konvoluojmë) mbi gjithë pozicionet e mundshme të vëllimit të hyrjes duke kryer shumëzimin matricor (*dot product*) midis *tensorit* të vëllimit të hyrjes dhe tensorit të filtrit. Pas konvolucionit do të gjenerohet një hartë aktivizimi 2-dimensionale që tregon aktivizimin e çdo filtri në pozicione të caktuara të vëllimit të hyrjes. Këto harta aktivizimi kombinohen më pas për të gjeneruar tensorin 3-dimensional në dalje ku dimensioni i tretë është i barabartë me numrin e filtrave të përdorur. Çdo filtër ka aftësinë të dallojë karakteristika të veçanta në mënyrën e organizimit të informacionit në vëllimin e hyrjes. Për shembull, në rastin e imazheve, filtrat janë të aftë të dallojnë pozicionime të caktuara të pikselave të cilat formojnë një hark, një vijë, etj. Gjithashtu, të gjithë neuronet e të njëjtit filtër përdorin të njëjtat pesha *w*. Në këtë mënyrë realizohet reduktimi masiv i numrit të parametrave në një shtresë konvolucionale, gjë që ndihmon për të mbajtur nën kontroll *overfitting*. Për shembull, supozojmë se kemi një *tensor* si në figurën 3.9 me përmasë 227 x 227 x 3 që përpunohet nga një shtresë konvolucionale me 32 filtra me përmasa 5 x 5 x 3. Output-i do të ishte një *tensor* me përmasa 223 x 223 x 32. Numri 223 është numri i pozicioneve unike që një filtër me 5 elementë mund të pozicionohet mbi një input me përmasë 227. Kështu, kjo shtresë prodhon 223 * 223 * 32 =1,591,328 vlera numerike

në dalje duke përdorur vetëm 5 * 5 * 3 * 32 + 32[5] = 2,432 parametra (pesha dhe bias). Nëse për përpunimin e *tensorit* do të përdornim një shtresë *fully connected* do të na duhej të përdornim 1,591,328 * (223 * 223 * 3 + 1[6]) = 237,407,041,664 parametra (pesha dhe bias).

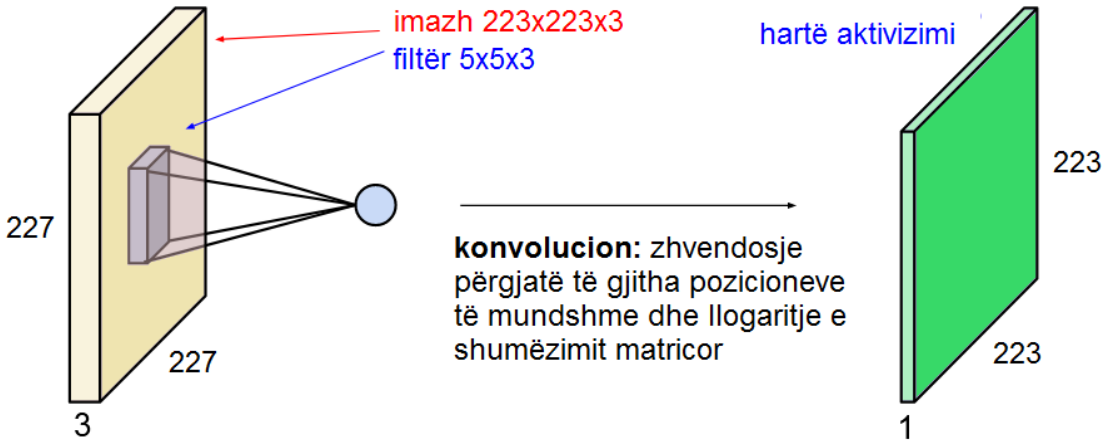


Figura 3.9. Konvolucioni sipas një filtri 5 x 5 x 3. Përshtatur nga [36]. Ekzistojnë 223 pozicione unike për një filtër 5 x 5 x 3 në një input 227 x 227 x 3. Konvolucioni prodhon një hartë aktivizimi me përmasa 223 x 223 ku çdo element është rezultati i shumëzimit matricor (*dot product*) të hyrjes me filtrin. Hartat e aktivizimit të çdo filtri kombinohen më pas për të gjeneruar një tensor 3-dimensional në dalje ku dimensioni i tretë është i barabartë me numrin e filtrave të përdorur. Nëse për shembull përdorim 32 filtra, tensori në dalje të shtresës konvolucionale do ketë përmasa 223 x 223 x 32.

Një arsye tjetër e përdorimit të parametrave të njëjtë për neuronet e të njëjtit filtër qëndron në faktin se nëse një filtër mëson të dallojë një tipar të imazhit (p.sh. vijë horizontale) në një pjesë të tij, për shkak të invariancës translacionale (*translational invariance*) të imazheve [43], nuk është e nevojshme të mësojë nga e para të detektojë këtë tipar në pjesë të tjera të imazhit. Në figurën 3.10 tregohen filtrat që një rrjet CNN [51] ka mësuar për një imazh në hyrje me dimensione 227 x 227 piksela x 3 ngjyra.

[5] 32 bias (nga 1 për secilin filtër)
[6] 1 bias për çdo konvoluim

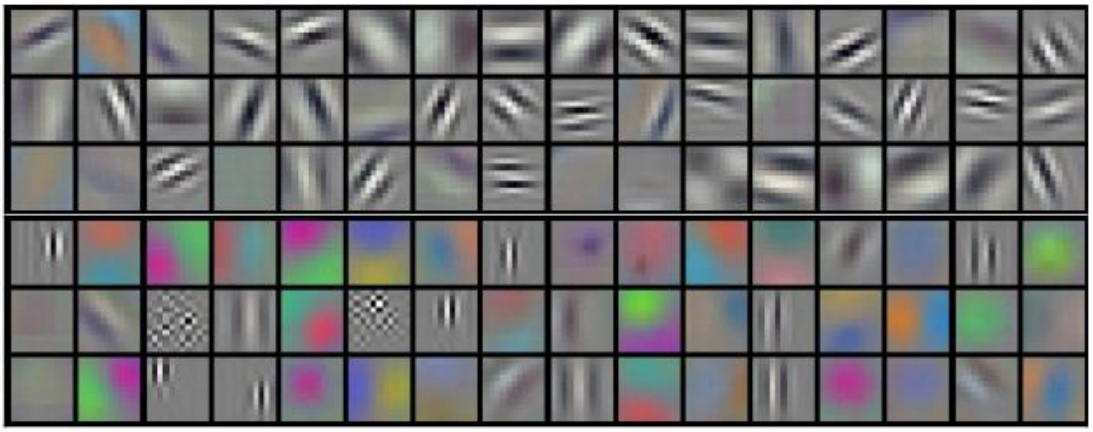


Figura 3.10. Filtrat e mësuar nga një rrjet CNN [51]. Secili nga 96 filtrat ka dimensionin 11 x 11 x 3. Këto filtra përdoren nga secili prej 55 x 55 neuroneve përgjatë dimensionit të thellësisë.

Një praktikë e zakonshme është vendosja e një bordure (*padding*) me zero rreth e rrotull të dhënave për të kontrolluar dimensionin e output-it të shtresës konvolucionale. Për shembull, *padding* me një bordurë me zero me trashësi 2 në rastin tonë do të gjeneronte një hartë aktivizimi 227 x 227 dimensionale për çdo filtër. Gjithashtu është i mundur përdorimi i filtrave me një *hap* (*stride*) të caktuar. Hapi përcakton sa pozicione kapërcehen gjatë zhvendosjes së filtrit. Për shembull, përdorimi i një filtri 5 x 5 x 3 me padding zero, me stride 2 në një tensor 227 x 227 x 3 do të gjeneronte një hartë aktivizimi 112 x 112 dimensionale. Figurat 3.11 dhe 3.12 ilustojnë përkatësisht konceptin e *padding* dhe *stride*.

Ekziston një formulë e cila mundëson llogaritjen e përmasave të vëllimit të daljes së një shtrese konvolucionale. Kështu, për një vëllim në hyrje me përmasa $H_1 \times W_1 \times D_1$, dhe N filtra me përmasa $F \times F$ të cilat përdoren me hap S dhe padding P, dimensionet e vëllimit të daljes janë $H_2 \times W_2 \times D_2$ ku: $H_2 = (H_1 - F + 2P) / S + 1$, $W_2 = (W_1 - F + 2P) / S + 1$ dhe $D_2 = N$. Në çdo rast, përmasat e filtrave, padding dhe stride zgjidhen në mënyrë të tillë që dimensionet e vëllimit të daljes të jenë gjithmonë numër i plotë.

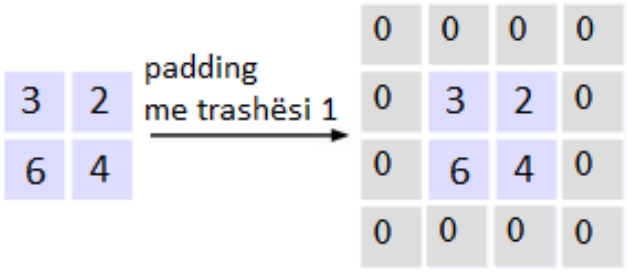


Figura 3.11. Padding. Përshtatur nga [36]. Përreth të dhënave vendoset një bordurë me trashësi të caktuar (në këtë rast 1) për të rritur dimensionin e vëllimit të tyre. Në këtë rast të dhënat origjinale kishin përmasë 2 x 2 dhe pas padding përmasa u rrit në 4 x 4.

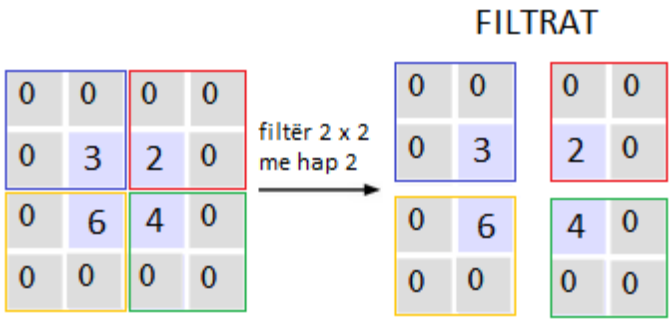


Figura 3.12. Stride. Përshtatur nga [36]. Në vëllimin e të dhënave aplikohet një filtër me përmasë fikse (në këtë rast 2 x 2) me një hap (*stride*) të caktuar (në këtë rast 2) i cili zhvendoset përgjatë dimensionit të gjatësisë dhe gjerësisë.

Shtresa pooling. Përveç shtresave konvolucionale, një praktikë e zakonshme është përdorimi i shtresave *pooling* për të kontrolluar *overfitting*. Këto shtresa vendosen midis dy shtresave konvolucionale të njëpasnjëshme. Funksioni i një shtrese pooling është të reduktojë vëllimin e të dhënave në mënyrë që të zvogëlohet numri i parametrave dhe veprimeve llogaritëse në rrjet, duke mbajtur kështu nën kontroll *overfitting*. Shtresa *pooling* vepron në mënyrë të pavarur për çdo hartë aktivizimi duke zvogëluar dimensionet e saj. Zvogëlimi i dimensioneve vjen duke përzgjedhur vetëm një pjesë të elementëve të hartës së aktivizimit sipas një kriteri të caktuar. Në çdo rast kriteri është i pandryshueshëm dhe veprimi i pooling nuk përdor parametra *w*. Kriteri më i përdorur në praktikë është vlera maksimale (*max pooling*). Kështu, kjo shtresë përdor një filtër me një hap të caktuar duke përzgjedhur elementin maksimal brenda dritares së krijuar nga filtri përgjatë dimensionit të gjerësisë dhe lartësisë. Duhet theksuar se dimensioni i thellësisë së vëllimit të të dhenave mbetet i pa

ndryshuar. Filtri më i përdorur është 2 x 2 me hap 2. Ashtu siç tregohet në figurën 3.13, për këtë lloj filtri, çdo veprim i pooling përzgjedh vlerën maksimale midis 4 vlerave të dritares së filtrit (2 x 2) duke eleminuar 75% të vlerave të tjera. Kjo gjë sjell reduktim të dimensionit të vëllimit të të dhënave duke zvogëluar numrin e llogaritjeve në rrjet si dhe mban nën kontroll *overfitting*. Përveç *max pooling*, përdorim të gjerë gjen edhe pooling sipas vlerës mesatare (*average pooling*). Në këtë rast përzgjidhet mesatarja e 4 vlerave të dritares së filtrit. Historikisht, pooling sipas vlerës mesatare ka qenë më i përdorur, por së fundmi pooling sipas vlerës maksimale po gjen përdorim më të gjerë pasi ka rezultuar më i suksesshëm në praktikë. Logjikshëm mund të rrjedhë ideja se reduktimi i vëllimit të të dhënave mund të çojë në humbje informacioni dhe të dëmtojë saktësinë e parashikimeve të rrjetit apo trajnimin e tij. Në praktikë, ky reduktim nuk sjell pasoja dhe përdorimi i vlerës maksimale ka rezultuar i mjaftueshëm për të përfaqësuar tiparet e të dhënave pa ndikuar në saktësinë apo mundësinë e rrjetit për t'u trajnuar.

Figura 3.13. Pooling sipas vlerës maksimale për një filtër me përmasa 2 x 2, me hap 2. Përshtatur nga [36]. Çdo veprim i pooling përzgjedh vlerën maksimale midis 4 vlerave të dritares së filtrit (2 x 2) duke eleminuar 75% të vlerave të tjera.

Arkitekturat CNN. Rrjetat neurale konvolucionale krijohen duke vendosur njëra mbi tjetrën shtresat konvolucionale dhe shtresat RELU (d.m.th. shtresë me neurone që përdorin funksionin ReLU si funksion aktivizimi), dhe duke futur shtresa pooling të ndërmjetme për të mbajtur nën kontroll kompleksitetin e rrjetit. Kjo vendosje përsëritet derisa imazhi të jetë reduktuar në një përmasë të vogël. Në arkitekturë vendosen gjithashtu shtresa *fully connected*. Shtresa e fundit e këtij lloji përmban rezultatin e klasifikimit. Arkitektura tipike e një rrjeti CNN është si më poshtë [36]:

INPUT -> [[CONV -> RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC

Në shprehjen e mësipërme: CONV = shtresë konvolucionale, FC = shtresë fully connected, "*" tregon përsëritje, N <=3, M>=0, dhe K>=0 (zakonisht K<3).

3.4.4 Rrjetat Neurale Rekurrente

Shpeshherë, të dhënat mund të jenë sekuenciale dhe të kenë vartësi në kohë nga njëra-tjetra (p.sh. një tekst apo sekuencat e një filmi). Një rrjet neural tradicional apo një CNN nuk mund "arsyetojë" në lidhje me sekuencat e kaluara. Për të realizuar këtë gjë përdoren rrjetat neurale rekurrente (*recurrent neural networks - RNN*). Arsyeja e projektimit dhe përdorimit të RNN-ve qëndron në nevojën për të përpunuar të dhëna të cilat kanë vartësi kontekstuale nga njëra-tjetra. Për shembull, kur ne shohim një video, ne kuptojmë çdo *frame* të saj duke u bazuar në *frame*-t e mëparshme. Ne nuk fshijmë çdo gjë nga kujtesa dhe rifillojmë arsyetimin nga e para për çdo *frame*, por mbartim kontekstin e *frame*-ve të kaluara në *frame*-in pasardhës dhe arsyetojmë në lidhje me këtë kontekst dhe *frame*-n aktual. Në këtë disertacion do të përdorim rrjetat RNN për të modeluar sekuencat e fjalëve. Rjetat RNN përmbajnë cikle në arkitekturën e tyre, gjë që i lejon të përpunojnë të dhëna sekuenciale në hyrje. Rrjetat neurale të zakonshme dhe CNN-të kërkojnë të dhëna në hyrje me përmasa fikse. Kjo gjë nuk është e domosdosshme për rrjetat neurale rekurrente (RNN) sepse ato pranojnë në hyrje sekuenca me gjatësi arbitrare.

RNN-të përpunojnë një sekuencë të dhënash në hyrje $\{x_1, \cdots, x_T\}$ duke përdorur një *formulë rekurrence* $h_t = f_\theta(h_{t-1}, x_t)$, ku $\boldsymbol{f}$ është një funksion i cili përdor të njëjtat parametra θ në çdo hap *t*. Termi h_t quhet *gjendje e fshehur* (*hidden state*) dhe përfaqëson kontekstin e sekuencës deri në hapin *t*. Për çdo input x_t, rrjeti llogarit gjendjen e fshehur h_t duke u bazuar tek gjendja e meparshme h_{t-1} dhe input-i x_t (ekuacioni 3.6). Parashikimi në dalje të rrjetit $\hat{y}$ varet kështu nga gjithë sekuenca e të dhënave në hyrje (ekuacioni 3.7). Sekuenca e parashikimeve në dalje formohet hap pas hapi duke gjeneruar secilën fjalë $\hat{y}$ një nga një. Për thjeshtësinë e paraqitjes së ekuacioneve, termat e bias nuk janë përfshirë. Funksioni i humbjes më i përdorur në rrjetat RNN është *cross-entropy* dhe ato trajnohen duke përdorur *backpropagation*.

Një praktikë e zakonshme, e përdorur për gjenerimin e sekuencave të fjalëve, është përdorimi i fjalës së sapo gjeneruar nga rrjeti RNN si input për parashikimin e rradhës.

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1}) \quad (3.6)$$

$$\hat{y}_t = \text{softmax}\ (W_y h_t) \quad (3.7)$$

Në ekuacionin 3.6, termi x_t është pjesë e vektorëvë të fjalëve $x_1, \cdots, x_{t-1}, x_t, x_{t+1}, \cdots, x_T$ që i korrespondojnë korpusit me T fjalë; termat W_{xh}, dhe W_{hh} janë përkatësisht matricat e parametrave që kushtëzojnë vektorin e fjalëve në hyrje x_t dhe të dhënat e hapit të mëparshëm h_{t-1}. Në vend të funksionit *tanh()* mund të përdoret funsioni tjetër jolinear ReLU. Në ekuacionin 3.7, $\hat{y}_t$është shpërndarja probabilitare e fjalëve të fjalorit në çdo hap *t*. Në thelb $\hat{y}_t$ është fjala e parashikuar duke marrë si input kontekstin aktual (d.m.th. h_{t-1}) dhe fjalën e fundit në hyrje x_t. Termi $W_y$parametrizon parashikimin $\hat{y}_t$. Qëllimi i trajnimit të rrjetit RNN është mësimi i parametrave *W*.

Në figurën 3.14 paraqitet arkitektura e një rrjeti RNN. Çdo drejtkëndësh përfaqëson një shtresë të rrjetit në hapin *t*. Çdo shtresë ka një grup neuronesh që përdorin një funksion të njëjtë aktivizimi (p.sh. *tanh()*) dhe përdorin të njëjtat parametra *W*.

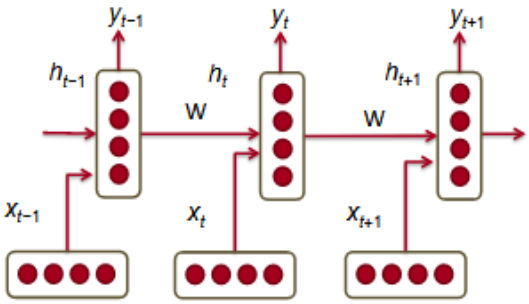


Figura 3.14. **Arkitektura e një rrjeti RNN [80]**. Çdo kuti përfaqëson një shtresë të rrjetit në hapin *t*. Në këtë rast janë paraqitur vetëm 3 hapa. Çdo shtresë ka një grup neuronesh që përdorin një funksion të njëjtë aktivizimi (p.sh. tanh()) dhe përdorin të njëjtat parametra *W*. Në çdo hap, output-i nga hapi i mëparshëm së bashku me fjalën e rradhës x_t përdoren si input për shtresën tjetër e cila gjeneron një parashikim $\hat{y}_t$ dhe kontekstin e rradhës h_t.

Problemet e zhdukjes dhe shpërthimit të gradientit. Qëllimi i rrjetave RNN është të mundësojnë shtrirjen e kontekstit në hapa kohorë të largët. Për ta bërë këtë, ato tejçojnë matricat e parametrave nga njëri hap në tjetrin. Le të marrim si shembull 2 fjalitë e mëposhtme:

1. Personi A hyri në dhomë. Personi B gjithashtu. Personi A përshëndeti ____.
2. Personi A hyri në dhomë. Personi B gjithashtu. Ishte pasdite dhe të gjithë po kthehesin në shtëpi nga një ditë e gjatë pune. Personi A përshëndeti____.

Në të dyja rastet, duke u bazuar në kontekstin e fjalisë, përgjigja për të dy vendet bosh është "personi B". Në teori, RNN-ja duhet të parashikojë "personi B", personi që është shfaqur disa hapa më parë në kontekstin e fjalisë, në të dyja rastet. Në praktikë nuk ndodh kështu. RNN-të janë më të prirura të bëjnë parashikim më të saktë për rastin e parë sesa për rastin e dytë. Kjo vjen për shkak se, gjatë fazës së *backpropagation*, kontributi i gradientit gradualisht zhduket gjatë tejçimit të tij në hapa të mëparshëm. Për rrjedhojë, për fjali më të gjata, probabiliteti që "personi B" të gjenerohet si parashikim i rrjetit zvogëlohet me rritjen e përmasës së kontekstit (d.m.th. gjatësisë së fjalisë). Ky problem njihet me emrin *zhdukja e gradientit* (*vanishing gradient*).

Duke patur parasysh ekuacionet 3.6 dhe 3.7, për të llogaritur gradientin e gabimit *dE/dW*, mbledhim vlerat e gabimit dE_t/dW për çdo hap kohor *t*. Kështu *dE/dW* llogaritet si:

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W} \tag{3.8}$$

Gabimi për çdo hap kohor *t* llogaritet duke përdorur rregullin zinxhir për ekuacionet 3.6 dhe 3.7. Ekuacioni 3.9 tregon si llogaritet ky gabim. Termi $dh_t/\, dh_k$ i referohet derivatit të pjesshëm të h_t në raport me *k* hapat e mëparshëm.

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W} \tag{3.9}$$

Problemi i *zhdukjes së gradientit* vjen për shkak të funksioneve të aktivizimit që neuronet e rrjetit RNN përdorin (*sigmoid*, *tanh*). Derivati i këtyre funksioneve është një vlerë më e vogël se 1. Sipas rregullit zinxhir në ekuacionin 3.9, kemi shumëzim të gradientëve që kanë vlerë më të vogël se 1. Ky zinxhir shumëzimesh gjeneron një vlerë gjithmonë e më të vogël derisa vlera e gradientit bëhet zero. Për rrjetin RNN kjo përkthehet në zerim të ndikimit që neuronet e fillimit kanë në rrjet, gjë që bën që hapat e hershëm në sekuencën e hyrjes të kenë pak ose aspak ndikim në kontekstin aktual dhe gjendjen h_t. Përveç kësaj, problemi i zhdukjes së gradientit bën që shtresat e para të rrjetit të jenë më të ngadalta për t'u trajnuar. Gjithashtu, meqë output-i i rrjetit RNN varet nga të gjitha shtresat e mëparshme, gabimi në shtresat e para tejçohet në të gjithë rrjetin dhe kjo sjell parashikime të pasakta.

Një alternativë për të zgjidhur problemin e zhdukjes së gradientit është përdorimi i funksionit të aktivizimit ReLU. Derivatet e këtij funksioni nuk janë midis vlerës 0 dhe 1, prandaj gradienti nuk zhduket. Sidoqoftë, edhe ky funksion ka një problem: output-i i tij është 0 kur në hyrje vendosen vlera negative. Në shumë raste kjo gjë mund të bllokojë plotësisht backpropagation sepse, pas një input-i negativ në funksionin ReLU, gradientët pasardhës do të jenë 0. Ky problem zgjidhet duke përdorur një funksion të quajtur *leaky ReLU* i cili është i ngjashëm me funksioni ReLU, por me ndryshimin që, për vlera negative në hyrje, output-i është një vlerë pozitive shumë afër me zeron (p.sh. 0.01), por jo zero. Kjo bën që gradienti të mos jetë zero dhe rrjeti mund të vazhdojë më tej trajnimin. Në teori ky funksion duket premtues për zgjidhjen e problemit të *zhdukjes së gradientit*, por në praktikë ai ka rezultuar problematik. Kjo vjen për shkak të përdorimit të konstantes me vlerë të vogël shumë afër zeros e cila mund të çojë përsëri në zhdukjen e gradientit. Gjithashtu, duke qenë se vlerat e funksionit ReLU/leaky ReLU nuk janë të kufizuara midis 0 dhe 1 apo -1 dhe 1 si në funksionet tanh/sigmoid, vlerat në dalje të këtij funksioni mund të marrin vlera të mëdha duke e bërë rrjetin RNN jo stabël dhe penguar procesin e të mësuarit.

Një problem tjetër i rrjetave RNN është *shpërthimi i gradientit* (*exploding gradient*). Ky shpërthim ndodh për shkak të shumëzimit (sipas rregullit zinxhir) të gradientëve që janë më të mëdhenj se 1. Për shkak të një numri të madh shtresash me neurone (zakonisht me qindra), rezultati bëhet një numër gjithmonë e më i madh duke prishur stabilitetin numerik të rrjetit dhe duke e bërë të pamundur trajnimin e tij. Për ta zgjidhur këtë problem aplikohet teknika e *prerjes së gradientit* (*gradient clipping*). Sipas kësaj teknike, nëse gradienti arrin një vlerë pragu, bëhet "prerja" e tij duke e zvogëluar vlerën dhe duke e sjellë brenda pragut [45].

Long Short-Term Memory (LSTM). Në vitet e fundit ka patur një sukses shumë të madh përdorimi i RNN-ve për një sërë problemesh si: njohja e zërit (*speech recognition*) [12], [75], modelimi gjuhësor [76], përkthimi neural [3], [4], [5], përshkrimi i imazheve (*image captioning*) [2], [27], [28], [29], [30], etj. Ky sukses është arritur në sajë të përdorimit të një lloji të veçantë rrjetash RNN të quajtura *Long Short-Term Memory* (*LSTM*) [8]. Këto rrjeta nuk vuajnë nga problemi i *zhdukjes së gradientit* dhe për rrjedhojë janë më të trajnueshme se RNN-të e thjeshta. Formula e tyre e rrekurrencës është e tillë që i lejon input-et x_t dhe h_{t-1}të bashkëveprojnë në një mënyrë matematikisht më komplekse dhe tejçimi i gradientit në hapat e mëparshëm në kohë bëhet në mënyrë më efektive. Edhe pse rrjetat e thjeshta RNN në teori mund të përpunojnë input me vartësi sekuenciale të shtrirë në kohë, në praktikë ato janë shumë të vështira për t'u trajnuar. LSTM-të janë më të efektshme për shkak të përdorimit të njësive më komplekse të aktivizimit. Kështu, përveç vektorit të gjendjes së fshehur h_t, LSTM-të përdorin gjithashtu dhe një vektor kujtese c_t. Në çdo hap kohor *t*, LSTM-ja zgjedh të lexojë, shkruajë apo reset-ojë gjendjen e saj duke përdorur mekanizma të mirëpërcaktuar që luajnë rolin e portave të cilat lejojnë ose jo futjen e informacionit brenda në "qelizën" LSTM.

LSTM-të janë projektuar për të patur "kujtesë" më të qëndrueshme duke e bërë më të lehtë përpunimin e input-it me vartësi sekuenciale të shtrirë më shumë në kohë. Formulimi matematikor për çdo qelizë LSTM është si më poshtë:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad \text{(Porta e hyrjes)} \quad (3.10)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad \text{(Porta e harresës)} \quad (3.11)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad \text{(Porta e daljes)} \quad (3.12)$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \quad \text{(Qeliza e kujtesës së re)} \quad (3.13)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t \quad \text{(Qeliza e kujtesës përfundimtare)} \quad (3.14)$$

$$h_t = o_t \circ \tanh(c_t) \quad \text{(Konteksti i ri)} \quad (3.15)$$

Në ekuacionet e mësipërme, termat *W* dhe *U* përfaqësojnë parametrat e rrjetit, ndërsa *b* përfaqësojnë bias. Qëllimi i trajnimit të rrjetit është mësimi i këtyre termave. Simboli "σ" përfaqëson funksionin sigmoid ndërsa "$\circ$" përfaqëson shumëzimin *element-wise* (d.m.th. shumëzimin e thjeshtë midis elementëve korrespondues të matricave, jo shumëzimin matricor).

Qeliza LSTM është e projektuar si një njësi me porta hyrëse dhe dalëse. Qëllimi i përpunimit të informacionit nga kjo qelizë është gjenerimi i kontekstit të ri h_t (d.m.th. gjendjes së re të qelizës) duke u bazuar në kontekstin ekzistues h_{t-1} (d.m.th. gjendjes së mëparshme që për rrjetin LSTM është output-i nga qeliza e mëparshme) dhe të dhënat e reja në hyrje. Vektorët *i, f* dhe *o* janë konceptuar si porta binare të cilat kontrollojnë nëse kujtesa e qelizës do të azhornohet, zerohet apo nëse ajo do të shfaqet në vektorin e gjendjes së fshehur (d.m.th. kontekstin). Funksionet e aktivizimit të këtyre portave janë sigmoidale dhe output-i varion nga 0 ne 1. Vektori *g* varion nga -1 ne 1 dhe mundëson modifikimin e përmbajtjes së kujtesës së qelizës. Kjo gjë realizohet nëpërmjet veprimit të mbledhjes dhe është një tipar shumë i rëndësishëm i rrjetave LSTM sepse gjatë *backpropagation* shmanget problemi i *zhdukjes së gradientëve* pasi veprimi i mbledhjes thjesht shperndan gradientin në rrjet dhe nuk bën shumëzimin e gradientëve. Kjo lejon gradientin qelizës së kujtesës *c* të shpërndahet mbrapsht në kohë (d.m.th. në rrjet) i pa ndërprerë për një kohë të gjatë derisa rrjedha e tij të ndërpritet nga një *portë e harresës* (*forget gate*).

3.4.5 Vëmendja Neurale

Vëmendja neurale i ndihmon modelet të fokusohen në pjesë të veçanta të input-it për të rritur saktësinë e output-it. Avantazhet e përdorimit të këtij mekanizmi janë të dyfishta. Së pari, ai redukton sasinë e informacionit që duhet përpunuar. Së dyti, duke qenë se informacioni për t'u përpunuar reduktohet, modelet gjenerojnë

output-e më të sakta pasi ka më pak informacion që mund të jetë irrelevant për input-in. Në figurën 3.15 ilustrohet koncepti i vëmendjes neurale për një model që merr një imazh në hyrje dhe gjeneron një përshkrim të tij në dalje. Në çdo hap të gjenerimit të fjalëve të përshkrimit, modeli fokusohet në pjesë të ndryshme të imazhit.

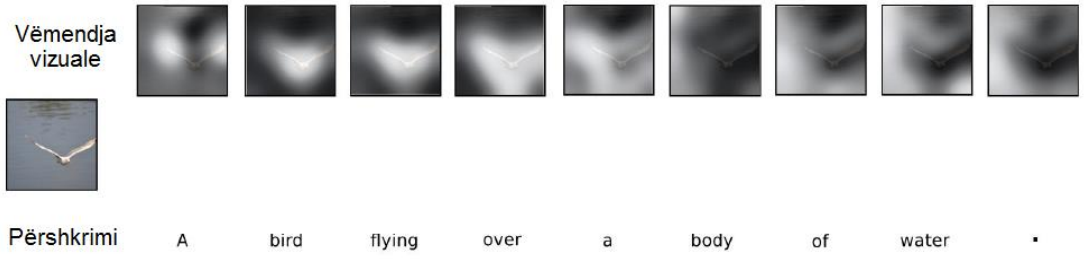


Figura 3.15. Vëmendja vizuale në kohë [28]. Gjatë gjenerimit të çdo fjale nga modeli, vëmendja e tij ndryshon për të reflektuar pjesët relevante të imazhit.

Në rastin e sistemeve pyetje-përgjigje vizuale dhe dialogut vizual, mekanizmat e vëmendjes neurale i ndihmojnë modelet të fokusohen në pjesë të veçanta të input-it tekstual (d.m.th. pyetjes aktuale në rastin e sistemeve pyetje-përgjigje dhe/ose historikut të bisedës në rastin e dialogut vizual) dhe vizual (d.m.th. imazhit). Duke shfrytëzuar këto modalitete (d.m.th. lloje) të vëmendjes modeli mund të arsyetojë më mirë dhe të gjenerojë përgjigje më të sakta. Kur modelet përdorin të dyja modalitetet e vëmendjes brenda së njëjtës arkitekture, atëherë vëmendjes i referohemi me termin *vëmendje multimodale*.

3.4.6 Dropout

Rrjetat neurale të thella (*deep neural networks*) përmbajnë shumë shtresa të fshehura dhe kjo i bën ato algoritma shumë të fuqishëm të cilët mund të mësojnë lidhje shumë të komplikuara midis vlerave të tyre në hyrje dhe vlerave në dalje [47]. Këto rrjeta përmbajnë një numër shumë të madh parametrash, fakt i cili sjell rrezikun e *overfitting* të rrjetit. *Overfitting* është një problem shumë serioz për këto rrjeta. Duke qenë të mëdha, ato janë të ngadalta për t'u përdorur, gjë që vështirëson akoma më shumë shmangien e *overfitting*, sidomos në rastet kur kemi të bëjmë me sisteme të cilat kombinojnë disa rrjeta. Ekzistojnë disa teknika për të shmangur *overfitting*, ndër të cilat mund të përmenden *rregullarizimi* dhe ndalimi i trajnimit në momentin që

performanca e validimit fillon të bjerë. Një teknikë tjetër për të adresuar këtë problem është *dropout* [47]. Ideja bazë është të shkëputësh nga rrjeti neural gjatë trajnimit disa nyje të përzgjedhura në mënyrë të rastësishme së bashku me lidhjet që ato kanë me nyjet e tjera. Kjo gjë e ndalon rrjetin të përshtatet së tepërmi me të dhënat e *dataset*-it. Zakonisht përzgjedhja e nyjeve që do të shkëputen nga rrjeti bëhet duke përdorur një probabilitet fiks *p.* Probabiliteti i shkëputjes së një nyjeje është i pavarur nga gjithë nyjet e tjera të rrjetit. Në praktikë vlera e *p* që ka rezultuar më e suksesshme ka qenë 0.5. Në figurën 3.16 ilustrohet koncepti i *dropout*.

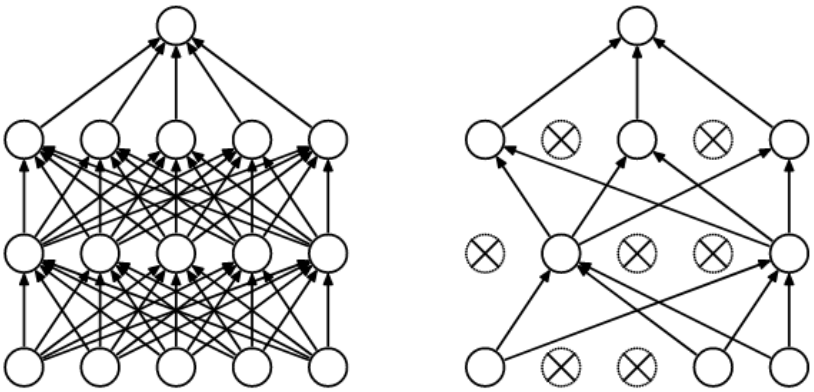


Figura 3.16. **Rrjet neural me dropout [47]**. **Në të majtë**: Një rrjet neural standard *fully connected* me 2 shtresa të fshehura. **Në të dhjathtë**: Një rrjet neural në të cilin është aplikuar *dropout*. Nyjet e shkëputura nga rrjeti janë shënuar me simbolin "⊗".

Gjatë testimit përdoret rrjeti i plotë duke bërë korrigjimin e parametrave përkatës. Kështu, nëse gjatë trajnimit një nyje mbahet në rrjet sipas probabilitetit *p,* gjatë testimit parametrat dalëse nga kjo nyje shumëzohen me *p*. Në këtë mënyrë garantohet që për çdo nyje të fshehur output-i i pritur sipas konfigurimit të rrjetit gjatë trajnimit të jetë i njëjtë me output-in gjatë testimit.

4

Punime të Ngjashme

Në këtë kapitull paraqiten punime të ngjashme dhe që kanë lidhje me teknikat e përdorura për implementimin e agjentëve të propozuar në këtë disertacion. Këto punime ndahen në tre grupe: *sistemet pyetje-përgjigje vizuale*, *vëmendja neurale* dhe *dialogu vizual*. Të gjitha modelet e propozuara në to janë të implementuar me rrjeta neurale artificiale. Për modelet që janë më të ngjashme me modelet e propozuara në këtë disertacion, përvec paraqitjes së tyre, vihet gjithashtu në dukje ku ndryshojnë ato nga modelet tona.

4.1 Sistemet Pyetje-Përgjigje Vizuale

Qasjet e bazuara në *deep learning* kanë treguar performancë të kënaqshme për sistemet pyetje-përgjigje vizuale [9], [21], [23], [25], [26]. Shumica e këtyre modeleve përdorin kombinime të rrjetave neurale CNN dhe RNN të lidhura në kaskadë me njëra-tjetrën. Rrjetat neurale CNN kanë rezultuar të suksesshme për përpunimin dhe nxjerrjen e tipareve të imazheve [1]. Nga ana tjeter, rrjetat neurale RNN kanë rezultuar të suksesshme për modelimin e sekuencave të fjalëve [7], [14], [15], [18], [19], [20], [21], [22]. Qasje të tjera përdorin *bag of words embedding* të pyetjes [17] ose *multilayer perceptrons* (MLP) [16]. Të gjitha qasjet e trajtojnë procesin e përpunimit të informacionit në hyrje (d.m.th. imazhin dhe pyetjen) dhe kthimin e përgjigjes si një problem klasifikimi dhe trajnojnë një klasifikues *softmax* për të gjeneruar përgjigjen. Procesi i gjenerimit të përgjigjes bëhet në disa hapa (çdo hap për një fjalë të përgjigjes). Ky klasifikues nxjerr si output një shpërndarje probabilitare të fjalëve të përgjigjes dhe si fjalë e saktë konsiderohet fjala që ka vlerën më të madhe në këtë shpërndarje probabilitare.

Janë të shumta mekanizmat dhe teknikat që janë propozuar për sistemet pyetje-përgjigje vizuale. Autorët në [15] përdorin një rrjet neural RNN i cili është trajnuar të ndryshojë parametrat e tij në mënyrë dinamike. Këto parametra ndryshohen duke u bazuar në pyetjen në hyrje. Kjo e lejon sistemin që të arsyetojë në mënyrë të ndryshme për pyetje të ndryshme. Arsyeja e implementimit të një qasjeje të tillë është që pyetje të ndryshme kërkojnë lloje dhe nivele të ndryshme të analizimit të imazhit në mënyrë që të gjenerohet përgjigja e saktë. Disa pyetje janë më të vështira për t'iu përgjigjur dhe kërkojnë arsyetim më të thellë. Për shembull pyetjet që kanë të bëjnë me numërimin e instancave të të njëjtës kategori objekti janë më të vështira për t'iu përgjigjur sesa pyetjet në lidhje me ngjyrën e një objekti.

Një model tjetër i propozuar në [20] është një arsyetues neural i bazuar në një *multilayer perceptron* (MLP). Ky arsyetues është në gjendje të ndryshojë pyetjen e përdoruesit në mënyrë iterative duke shtuar në të gjithmonë e më shumë detaje të cilat ndihmojnë në gjenerimin e një përgjigjeje sa më të saktë. Këto ndryshime bëhen duke arsyetuar rreth imazhit. Modeli e realizon këtë gjë duke përzgjedhur zona të imazhit të cilat janë relevante ndaj pyetjes dhe mëson të gjenerojë përgjigjen e saktë. I gjithë arsyetimi i modelit realizohet nëpërmjet shtresave të shumëfishta arsyetuese të cilat e ndihmojnë atë ta bëjë pyetjen më specifike se pyetja origjinale duke u fokusuar automatikisht në pjesë të rëndësishme të imazhit.

Autorët në [22] propozojnë një metodë kompakte bilineare të *pooling*. Kjo metodë kombinon tiparet e ekstraktuara nga një rrjet neural CNN për imazhin, kombinuar me një rrjet neural LSTM i cili përpunon pyetjen. Mekanizmi i propozuar i *pooling* redukton dimensionalitetin e paraqitjes së përbashkët të informacionit të ekstraktuar nga rrjeti CNN dhe LSTM. Reduktimi i dimensionalitetit bën që modeli përfundimtar të ketë më pak parametra dhe për rrjedhojë të jetë më i lehtë për t'u trajnuar.

Një alternativë tjetër janë sistemet multimodale të përbëra nga rrjeta neurale CNN dhe RNN të cilat trajnohen si një sistem i tërë (end-to-end) për të ekstraktuar informacionin e pyetjes dhe tiparet e imazhit. Gjithashtu ato ruajnë kontekstin gjuhësor të përgjigjes. Në fund, këto rrjeta kombinojnë këtë informacion për të gjeneruar përgjigjen për një pyetje të bërë në gjuhe natyrore [21].

4.2 Vëmendja Neurale

Mekanizmat e vëmendjes i lejojnë modelet e realizuara me rrjeta neurale që të fokusohen në pjesë të veçanta të informacionit në hyrje. Kjo ide është implementuar kohët e fundit me sukses në disa fusha si përshkrimi i imazheve (*image captioning*) [2], [27], [28], [29], [30], përkthimi neural [3], [4], [5] dhe sistemet pyetje-përgjigje vizuale [6], [7], [14], [17], [18], [19]. Në rastin e sistemeve pyetje-përgjigje vizuale, mekanizmat e vëmendjes neurale i lejojnë modelet që, në çdo hap të gjenerimit të përgjigjes, të fokusohen në pjesë të caktuara të inputit vizual (d.m.th. imazhit) dhe/ose tekstual (d.m.th. pyetjes) të cilat kanë lidhje me kontekstin e përgjigjes.

Në vend që të analizojnë të gjithë imazhin, modelet e vëmendjes vizuale fokusohen në mënyrë selektive në zona të ndryshme të imazhit për të ekstraktuar karakteristikat e tij të cilat kanë lidhje me pyetjen. Kjo gjë bëhet gjithashtu për të zvogëluar sasinë e informacionit që duhet përpunuar. Nga ana tjetër, mekanizmat e vëmendjes tekstuale gjejnë lidhje sintaksore dhe semantikore midis fjalëve. Të gjitha këto realizohen me anë të një arkitekture kodim-dekodimi. Gjatë kodimit informacioni tekstual kodohet në një gjendje të fshehur të një koduesi LSTM dhe gjatë fazës së dekodimit, dekoduesi LSTM e përkthen këtë gjendje të fshehur në një parashikim (d.m.th. shpërndarje probabilitare për çdo fjalë të përgjigjes).

Në mënyrë që të përmbushin detyrat e tyre, sistemet pyetje-përgjigje vizuale të paraqitura në disa studime realizojnë vëmendjen vizuale në disa hapa. Në [18] autorët propozojnë një rrjet vëmendjeje vizuale me disa hapa e cila e analizon imazhin disa herë për të arsyetuar rreth përgjigjes në mënyrë progresive. Modeli i propozuar përdor paraqitjen semantikore të pyetjes si një pikënisje për të identifikuar zonat e imazhit të cilat kanë lidhje me përgjigjen. Autorët në [17] propozojnë një skemë tjetër për vëmendjen vizuale me disa hapa. Në hapin e parë vendoset korrespondenca midis fjalëve të pyetjes dhe imazhit. Më pas, në hapin e dytë përdoret e gjithë paraqitja e pyetjes për të gjeneruar hartat e vëmendjes vizuale.

Ideja e përfshirjes së vëmendjes brenda arkitekturës standarde RNN është eksploruar më parë në modelet e propozuara në [7], dhe [19]. Xiong *et al*. në [19] përmirësojnë rrjetat dinamike të memorjes (*dynamic memory networks*) me një shtresë të re e cila përdor *gated recurrent units* (GRU) bidireksionale (d.m.th. dydrejtimshe).

Gjithashtu autorët propozojnë një arkitekturë të re për GRU e cila arsyeton për gjenerimin e përgjigjes. Zhu *et al* [7] përfshijnë vëmendjen vizuale në arkitekturën standarte LSTM për arsyetimin dhe gjenerimin e përgjigjes. Sidoqoftë, modelet e përmendura më sipër përdorin vetëm vëmendjen vizuale.

Hyeonseob *et al* [6] propozojnë rrjeta neurale të vëmendjes së dyfishtë të cilat fokusohen në zona specifike të imazhit dhe fjalëve të pyetjes. Ky fokusim bëhet në disa hapa për të mbledhur informacionin e duhur nga të dyja modalitetet e vëmendjes. Lu *et al* [14] propozojnë një vëmendje hierarkike e cila arsyeton njëkohësisht në lidhje me vëmendjen vizuale dhe tekstuale. Modeli i paraqitur në këtë disertacion për sistemet pyetje-përgjigje vizuale është vijimi i kësaj linje kërkimi si dhe idesë së eksploruar në [7] dhe [19] duke propozuar një arkitekturë risi LSTM. Në ndryshim nga modelet e [7] dhe [19] të cilat përdorin vetëm vëmendjen vizuale, modeli ynë përdor vëmendjen vizuale dhe tekstuale në çdo portë të qelizave të rrjetit LSTM. Çdo hap i vëmendjes varet nga gjendja e mëparshme e rrjetit LSTM dhe fokusimi aktual në fjalë të veçanta të pyetjes dhe zona të veçanta të imazhit. Përdorimi i kësaj arkitekture risi sjell përmirësimin e *state of the art* për sistemet pyetje-përgjigje vizuale.

4.3 Dialogu Vizual

Sistemet e dialogut vizual janë prezantuar dhe eksploruar rishtazi në [63], [64], [65], [66], [67] dhe [77].

Aktualisht *state of the art* për dialogun vizual është modeli i propozuar në [64]. Për të realizuar një sistem të tillë, autorët propozojnë disa zgjidhje të cilat të gjitha përdorin një arkitekturë kodimi-dekodimi të informacionit në hyrje (d.m.th. imazhi dhe pyetjes). Modeli i tyre më i suksesshëm është një rrjet memorizues LSTM i cili realizon një paraqitje të përbashkët midis informacionit tekstual dhe vizual. Në mënyrë që të përmirësojë saktësinë e përgjigjes, modeli përdor vëmendje ndaj historikut të bisedës e cila drejtohet nga pyetja dhe një paraqitje e përbashkët e pyetjes dhe imazhit. Për të paraqitur imazhin është marrë output-i nga shtresa e parafundit e VGG-16 [31].

Autorët në [64] përdorin vetëm vëmendjen tekstuale. Në ndryshim nga ky model, modeli i paraqitur në këtë disertacion për dialogun vizual përdor edhe vëmendjen vizuale e cila drejtohet nga pyetja aktuale. Mekanizmi i vëmendjes vizuale nuk është eksploruar më parë për sistemet e dialogut vizual. Për më tepër, në ndryshim nga modeli i [64], për të bërë paraqitjen (d.m.th. kodimin) e imazheve, modeli i paraqitur në këtë disertacion përdor hartat konvolucionale të tipareve (*convolutional feature maps*) të gjeneruara nga shtresa e katërt konvolucionale e VGG-16 [31]. Kjo shtresë gjeneron një hartë tiparesh të përbërë nga 196 elementë që përfaqësojnë 196 zona të imazhit. Çdo element është një vektor 512 dimensional. Kjo hartë përdoret për të llogaritur vëmendjen vizuale ku secila nga 196 zonat e imazhit kontribuon në masë të ndryshme në vëmendjen totale.

Së bashku me vëmendjen ndaj historisë së bisedës, propozohet një arkitekturë risi me vëmendje multimodale dhe agjenti shfrytëzon të dyja modalitetet e vëmendjes për të arsyetuar rreth përgjigjes dhe për të rritur saktësinë e saj. Përdorimi i kësaj arkitekture sjell përmirësimin e *state of the art*.

5

Agjent Inteligjent

në Sistemet Pyetje-Përgjigje Vizuale

Në këtë kapitull paraqitet modeli, implementimi dhe testimi i një agjenti inteligjent i cili i përgjigjet pyetjeve në një kontekst vizual. Konkretisht, duke marrë si input një imazh dhe një pyetje të shprehur në gjuhë natyrore, agjenti është i aftë t'i përgjigjet pyetjes rreth imazhit.

Për realizimin e këtij modeli është përdorur qasja statistikore (*data-driven)* për arsye se, ashtu siç është diskutuar në kapitullin 2, kjo qasje prodhon rezultate më të mira se qasjet e tjera. Agjenti është implementuar si një model i një rrjeti neural hibrid i cili merr në hyrje një imazh dhe një pyetje në formën e një sekuence fjalësh dhe nxjerr si rezultat një sekuencë shpërndarjesh probabilitare që përcaktojnë fjalët e përgjigjes. Ky model përbëhet nga tipe të ndryshme rrjetash neurale të cilët lidhen në kaskadë me njëri-tjetrin. Për të bërë përpunimin e imazheve përdoret një rrjet neural CNN, ndërsa pyetja dhe përgjigja përpunohen nga një rrjet neural RNN. Modeli i propozuar trajnohet si një i tërë (*end-to-end*) me *dataset*-et publike VQA [9] dhe Visual7W [7] të cilat janë *dataset*-et më të njohura dhe më të përdorura për sistemet pyetje-përgjigje vizuale. Duke qenë se këto *dataset*-e janë në gjuhën angleze, modeli njeh dhe gjeneron fjalë në gjuhën angleze. Nëse do kërkohej që modeli të njihte një gjuhë tjetër, do të ishte i mjaftueshëm trajnimi i tij me një *dataset* në gjuhën e dëshiruar pa modifikuar asgjë në arkitekturën e tij. Ky është një avantazh i përdorimit të qasjes statistikore dhe një arsye tjetër pse është përdorur kjo qasje për realizimin e modelit.

Pas trajnimit, modeli testohet mbi të njëjtat *dataset*-e dhe bëhet një analizë sasiore dhe cilësore mbi rezultatet e testimeve. Për të kuptuar më mirë sa e efektshme është arkitektura risi e propozuar, rezultatet e testimeve krahasohen me modele të ngjashme *state of the art* të cilat gjithashtu përdorin mekanizmin e vëmendjes neurale. Arkitektura e këtij agjenti së bashku me rezultatet e testimit dhe konkluzionet përkatëse paraqiten gjithashtu në [82], [83] dhe [84].

5.1 Modeli i Agjentit

Arsyeja e përdorimit të vëmendjes multimodale qëndron në faktin se, përveçse të vendosë në cilat zona të imazhit të fokusohet më shumë, agjenti vendos gjithashtu dhe se cilave fjalë të pyetjes t'i kushtojë më shumë vëmendje.

Ideja e përfshirjes së *vëmendjes multimodale* në SPPV është eksploruar së fundmi në [6] dhe [14]. Ndryshimi kryesor midis këtyre modeleve dhe modelit të propozuar në këtë disertacion është se ne përfshijmë *vëmendjen* si përbërës të çdo porte LSTM siç tregohet në figurën 5.1. Arkitektura standarde LSTM [8] është modifikuar për të akomoduar mekanizmin e vëmendjes multimodale. Ideja është që duke u fokusuar njëkohësisht në zona specifike të imazhit dhe në fjalë specifike të pyetjes, agjenti mund të vendosë se çfarë të harrojë nga kujtesa e tij aktuale, cila do të jetë kujtesa e re, apo cilën fjalë të gjenerojë në vazhdim. Përdorimi i kontekstit aktual (d.m.th. gjendja e mëparshme e fshehur e LSTM-së) ndihmon për të drejtuar vëmendjen në mënyrë korrekte dhe për të përmirësuar saktësinë e përgjigjes. Për implementimin e këtij agjenti janë përzgjedhur rrjetat neurale LSTM sepse këto të fundit kanë arritur performancë *state of the art* në disa fusha të përpunimit të sekuencave [30], [32], përfshirë këtu edhe SPPV [9], [21], [25].

Procesi i përpunimit të informacionit në hyrje dhe gjenerimit të përgjigjes në dalje konsiderohet si një proces me dy faza: *kodim* dhe *dekodim* [21], [25]. Në figurën 5.1 paraqiten këto dy faza. Në fazën e kodimit, agjenti memorizon të dhënat në hyrje (d.m.th. imazhin dhe pyetjen) nëpërmjet transformimit të tyre në një vektor të gjendjes së fshehur të një rrjeti LSTM (koduesi LSTM në figurën 5.1). Në fazën e

dekodimit, ky vektor dekodohet në një shpërndarje probabilitare të përgjigjeve kandidate (dekoduesi *softmax* në figurën 5.1).

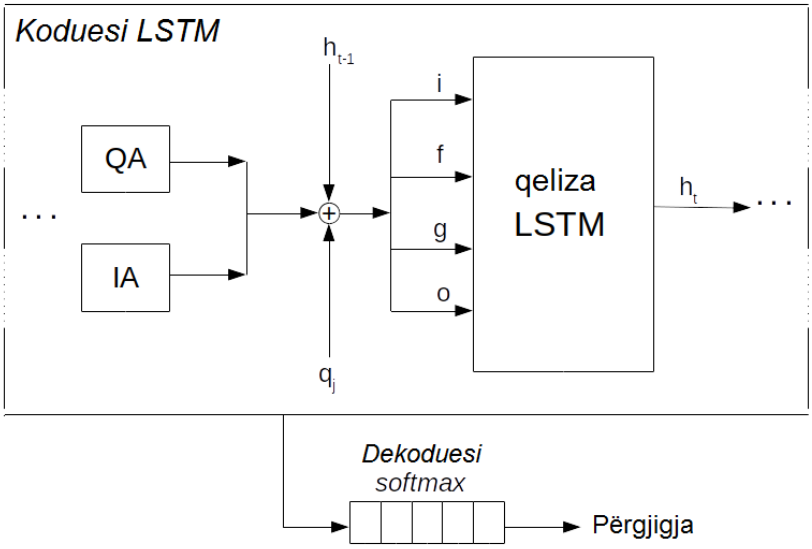


Figura 5.1: Dataflow e qelizave LSTM brenda në rrjetin e koduesit LSTM. Vëmendja tekstuale (QA), vëmendja vizuale (IA), gjendja e mëparshme e LSTM-së (h_{t-1}) dhe token-i (d.m.th. fjala) aktual i pyetjes (q_j) përdoren në çdo portë të çdo qelize LSTM për të gjeneruar kontekstin (h_t) që do të përdoret nga qeliza pasardhëse LSTM. Një klasifikues *softmax* përdoret si dekodues në dalje të rrjetit LSTM për të gjeneruar një nga një çdo fjalë *y* të përgjigjes.

5.1.1 Paraqitja e Fjalëve

Në modelin e propozuar, input-i është një imazh i përmasave 224 x 224 piksela dhe një pyetje e përbërë nga një sekuencë me gjatësi të ndryshueshme fjalësh.

Çdo fjalë fillimisht transformohet në një vektor *one-hot.* Vektori *one-hot* është një vektor që ka përmasën e fjalorit dhe të gjithë elementët i ka të barabartë me 0 përveç një elementi, pozicioni i të cilit është i njëjtë me pozicionin e fjalës në fjalor. Vlera e këtij elementi është e barabartë me 1 (për këtë arsye quhet *one-hot*). Çdo vektor *one-hot* transformohet më tej (*embedded*) në një vektor me vlera reale $Q = \{q_j | \, q_j \in R^D, j = 1, \dots, N\}$ ku *N* është numri i fjalëve të pyetjes, *D* është dimensioni i hapësirës së transformimit (*embedding space)* dhe $Q \in R^{DxN}$.

5.1.2 Paraqitja e Imazheve

Për të bërë paraqitjen e imazheve, është marrë output-i nga shtresa e fundit *fully connected* (fc7) e një rrjeti të paratrajnuar CNN, konkretisht VGG-16 [31]. Kjo shtresë e transformon imazhin *I*, në një vektor 4096-dimensional që përmban tiparet e imazhit (*image features*). Ky vektor bëhet *embed* më tej në një vektor *D* dimensional $V = \{v_i | v_i \in R^D, i = 1, \dots, M\}$ ku M është numri i *image features*, D është dimensioni i *embedding space* dhe $V \in R^{DxM}$.

Dimensioni i *embedding space* është 512, i njëjtë për të dyja paraqitjet (d.m.th. fjalët dhe imazhet). Modeli trajnohet si një i tërë (*end-to-end*) dhe gjatë trajnimit ai mëson edhe *embeddings* të fjalëve dhe imazheve.

5.1.3 Vëmendja Multimodale

Imazhi i përpunuar nga rrjeti CNN dhe i bërë *embed* në një vektor trajtohet si *token*-i i parë në hyrje të koduesit LSTM. Më pas, koduesi merr në hyrje *tokens-at* e fjalëve të pyetjes të cilat gjithashtu janë bërë *embed* në një vektor derisa arrin *token-in* e fundit të pyetjes. Procesi i gjenerimit të përgjigjes trajtohet si një detyrë *klasifikimi*. Për këtë arsye përdoret si dekodues në klasifikues *softmax* për gjenerimin dhe përzgjedhjen e fjalëve të përgjigjes. Gjatë trajnimit, modeli merr gjithashtu në hyrje *tokens*-at e përgjigjes (të bëra *embed* në një vektor). Qëllimi i trajnimit është maksimizimi i probabilitetit (*log-likehood)* të këtyre *tokens*-ave (d.m.th. në shpërndarjen probabilitare të gjeneruar nga modeli, këto *tokens*-a të kenë probabilitetin më të lartë). Gjatë testimit, shpërndarja probabilitare përdoret për të renditur përgjigjet kandidate dhe përgjigja me probabilitet më të lartë përzgjidhet si e saktë.

Ekuacionet e përpunimit të informacionit nga koduesi LSTM janë si më poshtë:

$$i_t = \sigma(W_{iv} v_t + W_{ih} h_{t-1} + W_{it}^{txt} a_t^{txt} + W_{it}^{img} a_t^{img} + b_i) \quad (5.1)$$

$$f_t = \sigma(W_{fv} v_t + W_{fh} h_{t-1} + W_{ft}^{txt} a_t^{txt} + W_{ft}^{img} a_t^{img} + b_f) \quad (5.2)$$

$$o_t = \sigma(W_{ov} v_t + W_{oh} h_{t-1} + W_{ot}^{txt} a_t^{txt} + W_{ot}^{img} a_t^{img} + b_o) \quad (5.3)$$

$$g_t = \tanh(W_{cv} v_t + W_{ch} h_{t-1} + W_{ct}^{txt} a_t^{txt} + W_{ct}^{img} a_t^{img} + b_c) \quad (5.4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t \quad (5.5)$$

$$h_t = o_t \circ \tanh(c_t) \quad (5.6)$$

ku "σ" është funksioni i aktivizimit sigmoidal dhe "∘" është produkt *element-wise*.

Autorët në [7] përdorin vetëm vëmendjen vizuale. Në ndryshim nga ta, modeli i propozuar në këtë disertacion integron gjithashtu vëmendjen tekstuale (d.m.th. vëmendjen ndaj fjalëve të caktuara të pyetjes) në çdo portë LSTM. Vemendja vizuale dhe tekstuale përfaqësohen respektivisht nga termi a_t^{img} dhe a_t^{txt}. Gjatë trajnimit rrjeti neural mëson këto terma. Në [14] autorët përdorin shumëzimin matricor (*dot product*) të *embeddings* të pyetjes dhe imazhit për të gjeneruar një *matricë ngjashmërie* e cila i shtohet *embeddings* të pyetjes ose imazhit për të drejtuar përkatësisht vëmendjen tekstuale ose vizuale. Në ndryshim nga ta, modeli i propozuar në këtë disertacion përdor gjendjen e mëparshme të fshehur të LSTM-së (h_{t-1}) dhe *embeddings* të pyetjes ose imazhit për të drejtuar respektivisht vëmendjen tekstuale ose vizuale. Vëmendja vizuale formulohet si më poshtë:

$$l_t^{img} = tanh(W_{lh}^{img} h_{t-1} + W_{lq}^{img} CNN(I) + b_{img}) \quad (5.7)$$

$$r_t^{img} = softmax\ (W_{img}^T l_t^{img}) \quad (5.8)$$

$$a_t^{img} = r_t^{img} CNN(I) \quad (5.9)$$

Për të gjeneruar vëmendjen vizuale përdoret output-i i shtresës së katërt konvolucionale të VGG-16 [31]. Kjo shtresë gjeneron një hartë konvolucionale karakteristikash (*convolutional feature map*) të imazhit *I*, përfaqësuar nga termi $CNN(I)$ në ekuacionet 5.7 dhe 5.9, e përbërë nga 196 (14 x 14) pjesë ku secila prej tyre është një vektor me 512 elementë. Kjo hartë karakteristikash do të përdoret për të llogaritur vëmendjen vizuale ku secila nga 196 zonat e imazhit kontribuon në vlerë të ndryshme. Një vlerë më e madhe nënkupton një vëmendje më të madhe në atë zonë. Koduesi LSTM gjeneron vektorët e vëmëndjes tekstuale dhe vizuale duke u bazuar në

kontekstin aktual (d.m.th. gjendja e mëparshme e fshehur e LSTM-së) dhe në këtë hartë konvolucionale.

Termi r_t^{img} në ekuacionet 5.8 dhe 5.9 përfaqëson shpërndarjen probabilitare të vëmendjes në çdo zonë të imazhit. Vëmendja vizuale llogaritet si një shumë e ponderuar e këtyre probabiliteteve dhe hartës konvolucionale. Ajo shprehet me anë të termit a_t^{img} i cili është një vektor 196-dimensional që përcakton kontributin e secilës zonë të imazhit në hapin kohor *t*.

Vëmendja tekstuale llogaritet si më poshtë:

$$l_t^{txt} = \tanh(W_{lh}^{txt} h_{t-1} + W_{lq}^{txt} Q + b_{txt}) \quad (5.10)$$

$$r_t^{txt} = softmax\,(W_{txt}^T l_t^{txt}) \quad (5.11)$$

$$a_t^{txt} = r_t^{txt} Q \quad (5.12)$$

Termi r_t^{txt} përfaqëson shpërndarjen probabilitare të vëmendjes për çdo fjalë të pyetjes. Duke u bazuar mbi këto probabilitete, vektori i vëmendjes tekstuale llogaritet si shumë e ponderuar e tyre me *embedding* të pyetjes. Vëmendja tekstuale shprehet me anë të termit a_t^{txt} në ekuacionet 5.11 dhe 5.12, i cili është një vektor *N*-dimensional që përcakton kontributin e çdo fjale të pyetjes në hapin kohor *t*.

Në të gjithë ekuacionet e mësipërme, termat *W* përfaqësojnë parametrat e rrjetit LSTM, ndërsa termat *b* përfaqësojnë *bias*-et e tij. Qëllimi i trajnimit është mësimi (d.m.th. llogaritja) e këtyre termave.

Në figurën 5.2 ilustrohet *dataflow* e gjenerimit të secilit modalitet të vëmendjes.

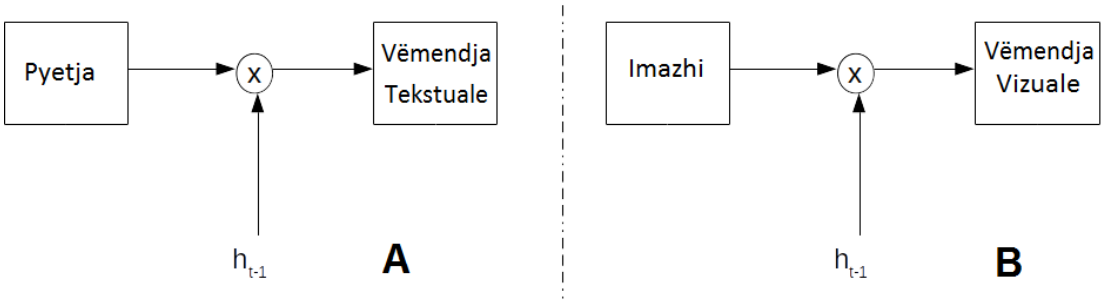


Figura 5.2: Gjenerimi i vëmendjes. (A) Në çdo hap, vëmendja tekstuale gjenerohet duke kombinuar kontekstin aktual (gjendjen e mëparshme të fshehur të LSTM-së h_{t-1}) dhe *embedding* të pyetjes. (B) Në çdo hap, vëmendja vizuale gjenerohet duke kombinuar kontekstin aktual (gjendjen e mëparshme të fshehur të LSTM-së h_{t-1}) dhe *embedding* të imazhit.

Në çdo hap, koduesi LSTM gjeneron vektorë të rinj të vëmëndjes tekstuale dhe vizuale dhe agjenti shfrytëzon këta vektorë për të gjeneruar fjalën e rradhës së përgjigjes. Arsyeja e gjenerimit të vektorëve të rinj të vëmendjes është se agjentit mund t'i duhet të fokusohet në pjesë të ndryshme të imazhit ose fjalë të ndryshme të pyetjes për të gjeneruar fjalën e rradhës të përgjigjes. Autorët në [6] përdorin vëmendjen akumulative në modelin e tyre për të mbajtur informacion mbi pjesët e fokusuara më herët dhe për të drejtuar vëmendjen në të ardhmen. Një vëmendje akumulative mund të vuajë nga futja e gabimeve në faza të hershme të cilat mund të tejçohen në hapat e ardhshëm të gjenerimit të vëmendjes. Në kontrast, modeli i prezantuar në këtë disertacion gjeneron vëmendje të pavarur në çdo hap dhe nuk vuan nga ky problem.

5.2 Optimizimi, Detajet e Implementimit dhe Hiperparametrat e Modelit

Për të implementuar modelin e propozuar është përdorur framework-u publik për *machine learning* Torch [10] dhe Python. Gjuha e programimit që Torch përdor është Lua [48]. Përpara trajnimit, fillimisht të gjitha pyetjet konvertohen në shkronja të vogla dhe pikëpyetjet së bashku me të gjitha shenjat e tjera të pikësimit hiqen. Modeli inicializohet sipas algoritmit Xavier [13]. Përjashtim bëjnë *embeddings* të cilat inicializohen duke përdorur një shpërndarje uniforme. Arsyeja e përdorimit të algoritmit Xavier është inicializimi i parametrave të rrjetit në mënyrë të tillë që derivatet e funksioneve të aktivizimit (tanh dhe sigmoid) të neuroneve në hapat fillestarë të mos kenë vlera të cilat do të pengonin trajnimin e rrjetit gjatë *backpropagation* (përkatësisht -1 dhe 1 për tanh dhe 0 për sigmoid). Kështu rrjeti inicializohet sipas një shpërndarjeje Gausiane me mesatare 0. Modeli trajnohet duke përdorur *backpropagation.* Shkalla globale e të mësuarit (*learning rate*) merret 10^{-4}. Si funksion humbjeje përdoret entropia e kryqëzuar (*cross-entropy*). Gjatë testimit

përzgjidhet përgjigja kandidate e cila ka probabilitet më të madh. Gjatë trajnimit, përmasa e batch-it të të dhënave (*batch size*) vendoset 128. Trajnimi bëhet për 40 epoka duke aplikuar teknikën *early stopping* (d.m.th ndërprerja e parakohëshme e trajnimit) nëse saktësia e modelit gjatë validimit nuk është përmirësuar në 5 epokat e fundit. Validimi bëhet pas përfundimit të çdo epoke trajnimi për të vendosur nëse trajnimi do të vazhdojë akoma për numrin e përzgjedhur të epokave apo do të ndërpritet.

Thellësia e rrjetit LSTM vendoset 512 për të gjitha eksperimentet. Dimensioni i hapësirës së *embeddings* të imazhit dhe fjalëve është 512. Duke ndjekur praktikën e zakonshme të vendosur në komunitetin e *machine learning*, për të mbajtur nën kontroll *overfitting*, përdoret *dropout* në rrjetin LSTM me probabilitet 0.5 për çdo shtresë. Gjithashtu për të parandaluar shpërthimin e gradientëve në rrjetin LSTM përdoret teknika e *gradient clipping.*

Për arsye të mungesës së burimeve të nevojshme hardware për trajnimin e një rrjeti CNN, është përdorur një rrjet i paratrajnuar VGG-16 [31] i cili është nga rrjetet më të përdorura nga modelet e SPPV. Paratrajnimi i tij është bërë mbi *dataset*-in ImageNet [49]. Një arsye tjetër e përdorimit të këtij rrjeti CNN është realizimi i një krahasimi sa më të saktë dhe të ndershëm me modelet *state of the art* të cilat përdorin të gjitha këtë rrjet. Nëse do përdorej një rrjet i ndryshëm CNN, atëherë do të ishte e pamundur të kuptohej nëse kontributi në rezultatin e testimeve do të vinte në mënyrë ortogonale nga një mënyrë më e mirë e përfaqësimit të imazheve dhe ekstraktimit të tipareve të tyre nga CNN apo nga një model më i mirë i agjentit.

Imazhet e *dataset*-eve ridimensionohen në përmasat 224 x 224 piksela sepse këto janë përmasat që pranon rrjeti VGG-16 [31]. Për të mësuar *embeddings* të imazheve përdoret output-i i shtresës së fundit konvolucionale (fc7) të këtij rrjeti Duke ndjekur praktikën e zakonshme në komunitetin e *machine learning*, për llogaritjen e vëmëndjes vizuale përdoret output-i nga shtresa e katërt konvolucionale e të njëjtit rrjet CNN.

5.3 Ambienti i Punës

Trajnimi i rrjetave neurale artificiale është një proces që kërkon shumë fuqi llogaritëse. Ato janë algoritma të vjetra, por që vetëm vitet e fundit po fillojnë të gjejnë zbatim dhe zhvillim. Kjo ka ndodhur për shkak të përmirësimeve të fuqisë llogaritëse të hardware-it të cilat kanë mundësuar trajnimin e rrjetave gjithmonë e më komplekse dhe gjithmonë e më inteligjente. Ndikimi i hardware-it në algoritmat e *deep learning* vihet re në kohën e trajnimit të tyre. Në këtë kuptim, ndryshimi nga një hardware në tjetrin është koha e trajnimit të algoritmit që mund të jetë nga minuta ose orë në disa ditë apo muaj. Nga ana tjetër, vlera e rezultatit të gjeneruar nga algoritmi është e pavarur nga hardware.

Hardware i përdorur për trajnimin dhe testimin e e agjentit është një server HP Proliant DL360 G7 [50]. Ky model ka procesor Intel(R) Xeon(R) CPU L5630 @2.13 GHz. Kujtesa RAM e instaluar është 48GB (@1333MHz, RDIMM). Hard disqe të instaluar janë 2 x 300GB.

Sistemi operativ i instaluar dhe konfiguruar është Ubuntu 16.04.3 LTS x86_64. Ky është sistemi operativ i cili është kompatibël me paketat software të përdorura për trajnimin dhe testimin e modelit.

Gjithashtu edhe për shkak të kufizimeve të kujtesës RAM të serverit dhe kërkesës së lartë për RAM nga modeli ynë, gjatë trajnimit dhe testimit është konfiguruar një hapësirë me madhësi prej 26GB në hard disqet lokale si hapësirë Swap e përdorur nga sistemi operativ Linux.

Paketat e mëposhtme software janë instaluar (së bashku me *dependencies* të tyre përkatëse), konfiguruar dhe përdorur në trajnimet dhe testimet e modelit:

- Torch 7
- Python 2.7 (versioni 2.7.12 i cili vjen si default me instalimin e sistemit operativ Ubuntu 16.04.3)
  - paketa të nevojshme: h5py, numpy, skimage
- Luajit 2.1.0-beta1 (e instaluar si paketë pas instalimit të Torch)
  - paketa të nevojshme: torch, nn, nngraph, hdf5, loadcaffe, cjson, image

5.4 Dataset-et dhe Metrikat e Përdorura

Trajnimi dhe testimi i agjentit është bërë mbi dy *dataset*-e publike: VQA [9] dhe Visual7W [7]. Meqënëse procesi i gjenerimit të përgjigjes trajtohet si një detyrë klasifikimi, metrika që përdoret për të dy *dataset*-et për vlerësimin e agjentit të propozuar është *saktësia mesatare* (*average accuracy*). Kjo është gjithashtu metrika e përdorur nga modelet *state of the art* për këto *dataset*-e. Ajo llogaritet si raporti i përgjigjeve të sakta me numrin total të përgjigjeve të gjeneruara nga agjenti.

5.4.1 Dataset-i VQA

Dataset-i VQA u përdor pasi është *dataset*-i më i madh dhe më kompleks për sistemet pyetje-përgjigje vizuale. Versioni VQA-v1 (i vitit 2015) u përzgjodh për të patur mundësinë e një krahasimi sa më të saktë dhe të ndershëm me modelet e tjera *state of the art*.

Dataset-i VQA-v1 është ndërtuar duke përdorur *dataset*-in Microsoft COCO [33] dhe përmban 123,287 imazhe trajnimi/validimi dhe 81,434 imazhe testimi. Çdo imazh ka disa pyetje rreth tij dhe secila prej tyre ka përgjigje nga disa njerëz. Ky *dataset* përmban 248,349 pyetje trajnimi, 121,512 pyetje validimi dhe 244,302 pyetje testimi. Numri total i imazheve, pyetjeve dhe përgjigjeve janë: 204,721 imazhe COCO (trajnim/validim/testim), 614,163 pyetje, 6,141,630 përgjigje të sakta dhe 1,842,489 përgjigje të pranueshme. Me përgjigje të pranueshme nënkuptohet një përgjigje e cila nuk bën pjesë në grupin e përgjigjeve të sakta, por përsëri është një përgjigje e vlefshme dhe nuk konsiderohet e pasaktë.

Dataset-i VQA ka dy kategori testimi: *përgjigje të lira* (*open-ended*) dhe *përgjigje me alternativa* (*multiple-choice*). Vlerësimi i modelit të propozuar është bërë për të dy keta skenarë. Duke ndjekur një praktikë shumë të përdorur në komunitetin e *machine learning*, janë përzgjedhur 1,000 përgjigjet më të shpeshta në set-in e trajnimit dhe validimit si përgjigje kandidate. Për rrjedhojë, nga *dataset*-i i plotë mbahen vetëm shembujt, përgjigjet e të cilëve bëjnë pjesë në këto 1,000 përgjigje të cilat përbëjnë 86.54% të përgjigjeve të trajnimit dhe validimit. Fjalori i

pyetjeve është i përbërë nga 7477 fjalë unike dhe secila prej tyre përdoret të paktën 3 herë në *dataset*.

Pyetjet e këtij *dataset*-i janë të ndara në disa grupe:

- (*Po/Jo*) – Janë pyetje që pranojnë vetëm përgjigje "Po" ose "Jo"
- *Numër* – Janë pyetje që kërkojnë numërimin e objekteve
- *Tjetër* – Janë të gjithë pyetjet e tjera të cilat nuk bëjnë pjesë në asnjë nga dy grupet e para

Përveç ndarjes së pyetjeve në grupet e mësipërme, në rezultatet e testimit raportohet edhe saktësia e përgjithshme e modelit.

5.4.2 Dataset-i Visual7W

Dataset-i Visual7W është një *dataset* i ri për sistemet pyetje-përgjigje vizuale i propozuar nga Zhu et. Al [7] dhe po fiton popullaritet në komunitetin *e machine learning* [22]. Ky *dataset* është ndërtuar duke u bazuar në *dataset*-in Microsoft COCO [33] dhe përmban 327, 939 çifte pyetje-përgjigje në lidhje me 47,300 imazhe COCO. Gjithashtu ai përmban 1,311,756 alternativa përgjigjesh të gjeneruara nga operatorë njerëzorë.

Pyetjet e këtij *dataset*-i ndahen në dy grupe të mëdha: treguese (*telling*) dhe drejtuese (*pointing*). Pyetjet që bëjnë pjesë në kategorinë treguese janë pyetjet që fillojnë me fjalët *çfarë, ku, kur, kush, pse, si* (*what, where, when, who, why dhe how*). Pyetjet që bëjnë pjesë në kategorinë drejtuese janë pyetjet që fillojnë me fjalën *cili* (*which*). Në total janë 7 lloje pyetjesh të cilat fillojnë me gërmën *w* në anglisht (përveç *how*). Për këtë arsye ky *dataset* ka marrë emrin *Visual7W*. Testimet e modelit të propozuar të agjentit janë bërë për kategorinë e pyetjeve treguese. Për këtë kategori, *dataset*-i përmban 14,366 imazhe trajnimi, 5,678 imazhe validimi dhe 8,609 imazhe testimi. Gjithashtu, për këtë kategori, ky *dataset* përmban 22,933 çifte pyetje-përgjigje *where*, 66,689 çifte pyetje-përgjigje *what*, 6,412 çifte pyetje-përgjigje *when*, 14,169 çifte pyetje-përgjigje *who*, 20,781 çifte pyetje-përgjigje *how* dhe 8,884 çifte pyetje-përgjigje *why*.

Pyetjet e Reja për Dataset-in Visual7W

Përveç pyetjeve standarde të këtij *dataset*-i ne propozojmë një ndarje të re të pyetjeve duke shtuar dy lloje të reja pyetjesh: *ngjyra* dhe *numri* në mënyrë që të përmirësohet vlerësimi i modelit. Arsyeja është se ekziston një ndryshim midis pyetjes "Çfarë loje po luajnë fëmijët?" dhe "Çfarë ngjyre është çadra?". Të dyja këto pyetje fillojnë me fjalën *çfarë* (*what*) dhe në versionin origjinal të Visual7W ato trajtohen si të të njëjtit lloj, por janë logjikisht të ndryshme. E njëjta gjë mund të thuhet për ndryshimin midis pyetjes "Si është koha?" (*How is the weather?*) dhe "Sa zogj ka në pemë?" (*How many birds are in the tree?*). Në gjuhën angleze këto dy pyetje fillojnë me të njëjtën fjalë dhe për rrjedhojë në *dataset* trajtohen si të së njëjtit lloj, por janë logjikisht të ndryshme. Motivimi për të propozuar këto dy lloje të reja pyetjesh, përveç ndryshimeve të tyre logjike, është se ato paraqesin shkallë të ndryshme vështirësie për t'u përpunuar dhe për të kthyer përgjigjen e saktë. Për shembull, pyetjet që bëhen rreth numrit të objekteve janë veçanërisht të vështira për t'iu përgjigjur saktë sepse sistemit i duhet të identifikojë numrin e instancave të ndryshme të së njëjtës kategori objekti brenda imazhit. Në versionin origjinal të *dataset*-it, për këtë lloj pyetjeje, vlerësimi i sistemit do të bëhej sikur ajo t'i përkiste kategorisë *si* (*how*) dhe nuk do të mund të arriheshin në përfundime të sakta në lidhje me aftësitë numëruese të sistemit.

Një arsye tjetër e propozimit të këtyre dy llojeve të reja të pyetjeve është sasia e konsiderueshme e tyre dhe impakti që ato kanë në vlerësimin e sistemit. Kështu, në *dataset*-in Visual7W ekzistojnë 15,919 pyetje për *ngjyrën* (*what color*) nga 66,689 pyetje të llojit *what* të cilat përbëjnë rreth 23,87% të pyetjeve të këtij lloji. Për sa i përket pyetjeve për *numrin* (*how many*), ekzistojnë 15,539 pyetje të tilla nga 20,781 pyetje të llojit *how* të cilat përbëjnë rreth 74,77% të pyetjeve të këtij lloji. Këto të dhëna tregojnë prezencën influencuese që këto pyetje kanë në *dataset* dhe impaktin e tyre të madh në një vlerësim sa më korrekt të SPPV-ve.

5.5 Rezultatet e Vlerësimit për Dataset-in VQA

Performanca e modelit të propozuar është krahasuar me modelet aktuale të *state of the art*. Termi *performancë* i referohet rezultateve të testimeve bazuar në metrikat e përdorura. Në rastin konkret, termi *performancë* i referohet *saktësisë mesatare (average accuracy)* të përgjigjeve të gjeneruara nga agjenti. Rezultatet e testimeve raportohen për të gjitha llojet e pyetjeve në mënyrë që të tregohen pikat e forta dhe të dobëta të modelit.

5.5.1 Vlerësimi Sasior

Rezultatet e testimit për kategorinë e përgjigjeve të lira paraqiten në tabelën 5.1.

Tabela 5.1: Rezultatet e testimit për dataset-in VQA për përgjigjet e lira krahasuar me state of the art. Saktësia paraqitet në %.

Modeli	**Pyetja**			
	Po/Jo	***Numër***	***Tjetër***	***Të gjitha***
HieCo [14]	79.5	**38.7**	48.3	60.1
D-NMN [16]	80.5	37.4	43.1	57.9
SAN(2, LSTM) [18]	79.3	36.6	46.1	58.7
SMem-VQA [17]	80.87	37.32	43.12	57.99
Modeli ynë	**81.9**	37.51	**49.1**	**61.08**

Nga rezultatet e tabelës 5.1 vihet re se të gjithë modelet arrijnë saktësinë më të lartë për pyetjet që kërkojnë përgjigje *Po/Jo*. Kjo gjë justifikohet nga fakti që ka vetëm dy përgjigje të mundshme dhe mundësia për gjenerimin e një përgjigjeje të pasaktë zvogëlohet.

Modeli që ngjan më shumë me modelin tonë është HieCo[14]. Nga të dhënat e tabelës 5.1 mund të shohim se qasja jonë ka performancë më të mirë dhe e përmirëson *state of the art* nga 60.1% (HieCo [14]) në 61.8% (modeli ynë). Për pyetjet që kërkojnë përgjigje *Po/Jo* dhe pyetjet e kategorisë *Tjetër* përmirësimi është respektivisht 1.03% në krahasim me [17] dhe 0.8% në krahasim me [14]. Këto rezultate tregojnë se agjenti ynë përfiton nga vëmendja multimodale dhe nga pavarësia e modaliteteve jo vetëm nga njëri-tjetri, por dhe nga hapat e mëparshëm të

vëmendjes. Për pyetjet e *Numërimit* rezultatet tregojnë se aftësia numëruese e modelit tonë dobësohet. Kjo gjë tregon se të paturit e modaliteteve të vëmendjes të lidhura me njëri-tjetrin si në HieCo[14] ndihmon në arritjen e një performance më të mirë në numërimin e objekteve. Rezultatet tregojnë se performanca e të gjithë modeleve bie për këtë lloj pyetjeje. Në fakt numërimi i objekteve është një problem i njohur në *computer vision* i cili ende konsiderohet i pazgjidhur.

Në tabelën 5.2 tregohen rezultatet e testimit për përgjigjet me alternativa (*multiple-choice*). Të dhënat ishin të disponueshme për krahasim vetëm me HieCo [14].

Tabela 5.2: Rezultatet e testimit për dataset-in VQA për përgjigjet me alternativa krahasuar me state of the art. Saktësia paraqitet në %.

Modeli	**Pyetja**			
	Po/Jo	***Numër***	***Të tjera***	***Të gjitha***
HieCo[14]	79.5	**39.8**	57.4	64.6
Modeli ynë	**82.1**	38.68	**58.61**	**66.08**

Nga rezultatet e tabelës 5.2 vihet re se modelet kanë performancë më të mirë për këtë kategori përgjigjesh. Kjo gjë vjen nga fakti se modelet shfrytëzojnë *bias*-et në secilën alternativë të përgjigjes (d.m.th ndihmohen nga fjalët e alternativave të përgjigjes). Sidoqoftë është e debatueshme nëse këto rezultate tregojnë progres apo jo pasi në aplikimet e modeleve në situata "reale" alternativat e përgjigjes nuk njihen paraprakisht.

Rezultatet e tabelës 5.2 tregojnë se qasja multimodale e modelit tonë ka performancë më të mirë dhe e përmirëson *state of the art* me 1.48% nga 64.6% në 66.08%. Gjithashtu vemë re se modeli ynë performon 1.03% më mirë se *state of the art* për pyetjet *Po/Jo*. Ashtu si në rastin e përgjigjeve të lira, modelet arrijnë performancën më të mirë për pyetjet *Po/Jo* dhe performancën më të ulët për pyetjet që kërkojnë numërimin e objekteve. Për këto pyetje, njësoj si në rastin e përgjigjeve të lira, të paturit e modaliteteve të vëmendjes të lidhura me njëri-tjetrin si në HieCo[14]

e ndihmon modelin të fokusohet më mirë në zonat e duhura të imazhit dhe të gjenerojë përgjigje më të sakta.

5.5.2 Vlerësimi Cilësor

Çdo pyetje kërkon lloje dhe nivele të ndryshme të të kuptuarit dhe të vëmendjes në mënyrë që të gjenerohet përgjigja e saktë. Për të krijuar një ide më të mirë të sjelljes së modelit të propozuar, dhe për të parë se si mund të përmirësohet ai në të ardhmen u përzgjodhën në mënyrë të rastësishme dhe u analizuan përgjigjet e gjeneruara nga agjenti. Në tabelat 5.3, 5.4 dhe 5.5 paraqiten disa shembuj për tipe të ndryshme pyetjesh për *dataset*-in VQA. Duke qenë se modeli është trajnuar gjuhën angleze, shembujt e përgjigjeve janë lënë në gjuhën origjinale.

Tabela 5.3: Shembuj të përgjigjeve për pyetjet *Po/Jo*. Pyetjet shënohen me "Q", përgjigjet e agjentit shënohen me "A" dhe përgjigjet e sakta shënohen me "GT"

		
1. Q: Is the horse eating? A: No GT: No	2. Q: Is there a bench? A: No GT: Yes	3. Q: Is there a red sandal here? A: Yes GT: Yes

4. Is the road paved?

A: Yes

GT: No

5. Q: Are there lights on in the two buildings?

A: Yes

GT: Yes

6.Q: Are these zebra confined?

A: Yes

GT: Yes

7. Q: Is the kitchen cluttered?

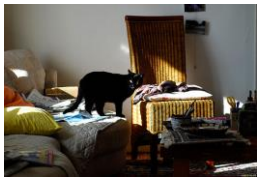
A: No

GT: Yes

8. Q: Is the beach crowded?

A: Yes

GT: Yes



9. Q: Are there any scissors in this picture?

A: No

GT: Yes

10. Q: Is this animal in a zoo?

A: No

GT: No

Nga përgjigjet e gjeneruara nga agjenti për pyetjet *Po/Jo* vihen re karakteristikat e mëposhtme:

+ Modeli fokusohet në mënyrë të saktë, identifikon dhe arsyeton në lidhje me objektet në plan të parë (*foreground*) dhe karakteristikat e tyre (p.sh. imazhet 1 dhe 3).
– ***Vështirësi për të arsyetuar rreth objekteve në sfond***. Modeli ka vështirësi në identifikimin në mënyrë të saktë të objekteve në sfond. Për këtë lloj objektesh

aftësia e fokusimit të vëmendjes në mënyrë të saktë dobësohet dhe modeli nuk mund të arsyetojë në mënyrë të saktë rreth tyre (p.sh. imazhet 9 dhe 4).

– ***Vështirësi në identifikimin e objekteve jo të plota.*** Aftësia për të identifikuar objektet të cilat shfaqen jo të plota në imazh dobësohet dhe agjenti nuk mund të arsyetojë në mënyrë të saktë rreth tyre (p.sh. imazhet 2 dhe 7).

Tabela 5.4: Shembuj të përgjigjeve për pyetjet e numërimit të objekteve. Pyetjet shënohen me "Q", përgjigjet e agjentit shënohen me "A" dhe përgjigjet e sakta shënohen me "GT"

 1. Q: How many street signs are shown? A: Two GT: Four	 2. Q: How many horses are there? A: Two GT: Two	 3. Q: How many bikes are there? A:Two GT: One
 4. Q: How many people do you see? A:Two GT: None	 5. Q: How many people are there? A: None GT: None	 6. Q: How many yellow planes are there? A: One GT: Three
 7. Q: How many giraffes are in this picture? A: Two GT: Two	 8. Q: How many jets are there? A:One GT: Two	 9. Q: How many birds? A: Two GT: None

 10. Q: How many buses are there? A: One GT: One

Nga përgjigjet e gjeneruara nga agjenti për pyetjet *Numër* vihen re karakteristikat e mëposhtme:

+ ***Modeli fokusohet në mënyrë të saktë, identifikon dhe arsyeton në lidhje me objektet në plan të parë (foreground) dhe karakteristikat e tyre*** (p.sh. imazhet 1, 7 dhe 10).
+ ***Modeli fokusohet në mënyrë të saktë dhe identifikon objektet në sfond të cilët në imazh nuk shfaqen "të shkrirë" me njëri-tjetrin*** (p.sh. imazhi 2).
– ***Vështirësi për të identifikuar objektet në sfond***. Modeli ka vështirësi në identifikimin në mënyrë të saktë të objekteve në sfond. Për këtë lloj objektesh aftësia e fokusimit të vëmendjes në mënyrë të saktë dobësohet dhe modeli nuk mund të arsyetojë në mënyrë të saktë rreth tyre (p.sh. imazhet 3, 4 dhe 9)
– ***Vështirësi në dallimin nga njëri-tjetri të objekteve në sfond që në imazh shfaqen "të shkrirë" me njëri-tjetrin.*** Aftësia për t'u fokusuar në mënyrë të saktë dobësohet për këto lloj objektesh dhe modeli nuk mund të arsyetojë në mënyrë të saktë rreth tyre dhe as t'i numërojë ato (p.sh. imazhet 1 dhe 8) .

Tabela 5.5: Shembuj të përgjigjeve për pyetjet e kategorisë *Tjetër*. Pyetjet shënohen me "Q", përgjigjet e agjentit shënohen me "A" dhe përgjigjet e sakta shënohen me "GT"

1. Q: Who is with the giraffes?
A: No one
GT: No one

2. Q: What is the woman in front sitting on?
A: A bicycle
GT: A bicycle

3. Q: What color are the walls?
A: yellow
GT: yellow

4. Q: where are the engines?
A: in the middle of the plane
GT: Behind the wings toward the back of the fuselage.

5. Q: what has a purple border?
A: the window
GT: The box truck.

6. Q: What kind of flooring is in the room?
A: White tile.
GT: Gray marble tile.

7. Q: What angle was the picture taken from?
A: From the left side of the sign
GT: Below the sign, looking up at it

8. Q: Where was this photo taken?
A: At a tennis court
GT: At a tennis court

9. Q: how is the food served?
A: in a basket
GT: in a basket

10. Q: Where was this photo taken?
A: At the park
GT: At the park

Nga përgjigjet e gjeneruara nga agjenti për pyetjet e kategorisë *Tjetër* vihen re karakteristikat e mëposhtme:

+ ***Modeli fokusohet në mënyrë të saktë, identifikon dhe arsyeton në lidhje me objektet në plan të parë (foreground) dhe karakteristikat e tyre*** (p.sh. imazhet 1, 2 dhe 9).
+ ***Modeli fokusohet në mënyrë të saktë dhe identifikon objektet në sfond të cilët në imazh janë qartësisht të dallueshëm nga njëri-tjetri*** (p.sh. imazhet 3, 8 dhe 10).
– ***Vështirësi në dallimin nga njëri-tjetri të objekteve në sfond që në imazh shfaqen "të shkrirë" me njëri-tjetrin.*** Aftësia për t'u fokusuar në mënyrë të saktë dobësohet për këto lloj objektesh dhe modeli nuk mund të arsyetojë në mënyrë të saktë rreth tyre (p.sh. imazhet 4, 5 dhe 6).

5.6 Rezultatet e Vlerësimit për Dataset-in Visual7W

Performanca e modelit të propozuar është krahasuar me *state of the art*. Të dhënat e krahasimit për këtë *dataset* ishin të disponueshme vetëm për LSTM-Att [7]. Edhe për këtë *dataset*, termi *performancë* i referohet *saktësisë mesatare (average accuracy)* të përgjigjeve të gjeneruara nga agjenti. Rezultatet e testimeve raportohen për të gjitha llojet e pyetjeve në mënyrë që të tregohen pikat e forta dhe të dobëta të modelit.

5.6.1 Vlerësimi Sasior

Rezultatet e testimit për këtë *dataset* paraqiten në tabelën 5.6.

Tabela 5.6: Rezultatet e testimit për dataset-in Visual7W krahasuar me state of the art. Saktësia paraqitet në %.

Modeli	**Lloji i pyetjes**								
	Çfarë	***Ku***	***Kur***	***Kush***	***Pse***	***Si***	***Ngjyrë***	***Numër***	***Të gjitha***
LSTM-Att [7]	50.6	53.1	71.2	58.1	49.6	41.8	44.5	51.0	51.7
Modeli ynë	**53.7**	**56.8**	**75.1**	**61.7**	**52.0**	**45.0**	**47.1**	**53.8**	**54.8**

Nga tabela 5.6 vemë re se modeli ynë ka një saktësi të përgjithshme prej 3.1% më lart se *state of the art*. Gjithashtu, modeli ynë performon të paktën 2.4% më mirë se *state of the art* për të gjithë llojet e pyetjeve. Kjo gjë tregon se agjenti përfiton nga

vëmendja multimodale. Të dy këto modele arrijnë saktësinë më të madhe për pyetjet *kur*. Shumica e pyetjeve *kur* në *dataset* pyesin në lidhje me kohën kur është bërë fotografia prandaj është më e lehtë për modelin të përgjigjet në mënyrë të saktë pasi ekzistojnë vetëm dy alternativa logjike: ditë dhe natë. Agjenti ynë arrin përmirësimin maksimal të performancës (3.9%) për këtë lloj pyetje. Kjo tregon se përfshirja e vëmendjes multimodale në arkitekturën LSTM e ndihmon modelin të arsyetojë më mirë dhe të rrisë saktësinë e përgjigjes.

Llojet e pyetjeve që modelet kthejnë përgjigje të pasaktë në pjesën më të madhe të rasteve janë *ngjyra, pse* dhe *si*, ku saktësia më e ulët arrihet për këtë të fundit. Këto janë pyetje të vështira sepse kërkojnë një nivel të lartë të të kuptuarit të imazhit dhe arsyetimit rreth tij. Për shembull, në rastin e pyetjeve *pse*, modeli jo vetëm duhet të lidhë pyetjen me pjesën e duhur të imazhit, por edhe të arsyetojë në lidhje me mendimin praktik (*common sense*) që përmban pyetja dhe imazhi. Për këto pyetje të vështira agjenti ynë ka saktësi 2.4% më të lartë. Kjo gjë tregon se përfshirja e vëmendjes multimodale brenda në arkitekturën LSTM ndihmon për të arsyetuar në lidhje me *common sense*. Në fakt, përfshirja e *common sense* në modelet e *deep learning* është aktualisht një temë kërkimi shkencor shumë aktuale në komunitetin e *machine learning.*

5.6.2 Vlerësimi Cilësor

Edhe për këtë *dataset*, për të krijuar një ide më të mirë të sjelljes së modelit të propozuar, dhe për të parë se si mund të përmirësohet ai në të ardhmen u përzgjodhën në mënyrë të rastësishme dhe u analizuan përgjigjet e gjeneruara nga agjenti. Në tabelën 5.7 paraqiten disa shembuj për tipe të ndryshme pyetjesh për *dataset*-in Visual7W.

Tabela 5.7: Shembuj të përgjigjeve të gjeneruara nga agjenti për tipe të ndryshme pyetjesh për dataset-in Visual7W. Pyetjet shënohen me "Q", përgjigjet e agjentit shënohen me "A" dhe përgjigjet e sakta shënohen me "GT"

 1. Q: what is on the ground between the man's legs? A: A backpack. GT: A backpack.	 2. Q: What has been cut? A: A pizza. GT: A cake.	 3. Q: Where is the elephant? A: Next to the tree. GT: Next to the tree.
 4. Q: Where is a person sitting? A: In a boat. GT: In a boat.	 5. Q: When was the picture taken? A: Daytime. GT: Daytime.	 6.Q: When was this photo taken? A: Night. GT: Night.
 7. Q: Who is not wearing a jacket? A: The man on the right. GT: The man on the far left.	 8. Q: who is playing tennis? A: Two men. GT: Two men.	 9. Q: Why are there trajns? A: For people to ride. GT: For people to ride.

 10. Q: Why are they staring down? A: Looking at phone. GT: Looking at computer.	 11. Q: How is the day? A: Sunny. GT: Clear and beautiful.	 12. Q: How is the photo? A: Clear. GT: Clear.
 13. Q: What color is the man's board? A: Black. GT: Orange.	 14. Q: What color is the sauce? A: The sauce is red. GT: The sauce is red.	 15. Q: How many dogs are there? A: One. GT: Three.
 16. Q: How many people are pictured? A: One. GT: One.		

Nga përgjigjet e gjeneruara nga agjenti vihen re karakteristikat e mëposhtme:

+ ***Modeli fokusohet në mënyrë të saktë, identifikon dhe arsyeton në lidhje me objektet në plan të parë (foreground) dhe karakteristikat e tyre*** (p.sh. imazhet 1, 3, 4 dhe 14).
+ ***Modeli fokusohet në mënyrë të saktë dhe identifikon objektet në sfond të cilët në imazh janë qartësisht të dallueshëm nga njëri-tjetri*** (p.sh. imazhet 8, 11 dhe 12).
– ***Vështirësi në numërimin e objekteve*****.** Rezultatet tregojnë se agjenti nuk nxjerr në mënyrë të saktë informacionin e duhur nga imazhi për të realizuar

numërimin e objekteve (p.sh. imazhi 15). Agjenti i përgjigjet me “asnjë”, “një” ose “dy” shumicës së pyetjeve që bëhen për numërimin e objekteve.

- ***Vështirësi në dallimin e objekteve jo të plota.*** Aftësia për të identifikuar objektet të cilat shfaqen jo të plota në imazh dobësohet dhe agjenti nuk mund të arsyetojë në mënyrë të saktë rreth tyre (p.sh. imazhi 2).
- ***Vështirësi për t’iu përgjigjur saktë pyetjeve që kanë lidhje me ngjyrën.*** Për pyetje që kanë lidhje me ngjyrën, agjenti ka tendencë të kthejë si përgjigje ngjyrën që shfaqet më shumë në imazh dhe ka vështirësi për të fokusuar vëmendjen e tij mbi objektet më të vogla në mënyrë që të dallojë ngjyrën e tyre (p.sh. imazhi 13)
- ***Vështirësi për t’iu përgjigjur saktë pyetjeve “pse”.*** Agjenti ka vështirësi për të arsyetuar në mënyrë të saktë rreth pyetjeve “pse” (p.sh. imazhi 10). Kjo lloj pyetjeje kërkon nivele të larta arsyetimi dhe shfrytëzimi i *mendimit praktik* (*common sense*) që përmban pyetja nuk mjafton për tu përgjigjur saktë.

5.7 Diskutime dhe konkluzione

Në këtë kapitull u paraqit arkitektura dhe implementimi i një agjenti inteligjent me vëmendje multimodale për sistemet pyetje-përgjigje vizuale. Agjenti shfrytëzon njëkohësisht vëmendjen tekstuale (d.m.th. vëmendjen ndaj fjalëve të pyetjes) dhe vëmendjen vizuale (d.m.th vëmendjen ndaj imazhit) për të identifikuar entitetet e pyetjes dhe për t’i lidhur ato me imazhin. Ai mëson t’i përgjigjet pyetjeve duke gjeneruar modalitete të vëmendjes të pavarura nga njëra tjetra në çdo hap të gjenerimit të përgjigjes (d.m.th. për çdo fjalë të përgjigjes).

Agjenti u testua dhe vlerësua në mënyrë sasiore mbi *dataset*-et publike VQA and Visual 7W dhe rezultatet u krahasuan me modelet e ngjashme *state of the art* të cilat përdorin mekanizmin e vëmendjes neurale. Si metrikë vlerësimi u përdor *saktësia mesatare.* Rezultatet e testimit treguan se modeli i propozuar ka performancë më të mirë se modelet e ngjashme dhe përmirëson *state of the art* për të dy *dataset*-et sipas rezultateve të tabelave 5.1, 5.2 dhe 5.6. Kjo tregon se integrimi i vëmendjes multimodale në arkitekturën LSTM ndihmon agjentin të arsyetojë më mirë dhe të përmirësojë saktësinë e përgjigjes. Rezultatet tregojnë gjithashtu se të paturit e

modaliteteve të pavarura të vëmendjes ndihmon në saktësinë e përgjigjes. Përjashtim nga ky rast bëjnë pyetjet të cilat kanë të bëjnë me numërimin e objekteve në imazh. Për këto pyetje, arkitektura e propozuar nuk është mjaftueshëm e efektshme për të gjeneruar përgjigje të sakta.

Për *dataset*-in Visual7W u propozuan dy lloje të reja pyetjesh (*ngjyra* dhe *numri*) të cilat ndihmojnë në një vlerësim më të mirë të modeleve si dhe u diskutua rreth arsyeve të përfshirjes së këtyre pyetje dhe përfitimet që vijnë nga përfshirja e tyre.

Përgjigjet dhe gabimet e agjentit u vlerësuan gjithashtu në mënyrë cilësore për të kuptuar më mirë sjelljen e tij dhe si mund të përmirësohet në të ardhmen. Nga analiza e këtyre rezultateve u arrit në konkluzionin se agjenti i propozuar është i aftë të përdorë në mënyrë të saktë dhe përfiton nga vëmendja multimodale në disa aspekte. Ai është i aftë:

- Të fokusojë vëmendjen, identifikojë dhe arsyetojë në mënyrë të saktë rreth objekteve në plan të parë (*foreground*) dhe karakteristikave të tyre.
- Të fokusojë vëmendjen, identifikojë dhe arsyetojë rreth objekteve në sfond të cilat janë qartësisht të dallueshëm nga njëri-tjetri.

Nga analiza e përgjigjeve të gabuara u arrit në konkluzionin se agjenti gjithashtu ka limitime dhe vështirësi:

- Të numërojë objektet.
- Të fokusojë vëmendjen, identifikojë dhe arsyetojë rreth objekteve të cilët nuk shfaqen të plotë në imazh ose shfaqen të “shkrirë” me njëri-tjetrin dhe/ose planin e parë/sfondin
- Të arsyetojë në mënyrë të saktë rreth ngjyrës së objekteve.
- Të arsyetojë në mënyrë të saktë në lidhje me pyetjet *pse*.

Këto probleme tregojnë nevojën për të përmirësuar mekanizmin e vëmendjes dhe tejkalimi i tyre është subjekt i punës në të ardhmen.

6

Agjent Inteligjent

në Dialogun Vizual

Në këtë kapitull paraqitet modeli, implementimi dhe testimi i një agjenti inteligjent i cili është i aftë të bëjë një dialog me njerëzit në gjuhën natyrore në një kontekst vizual. Me *dialog* nënkuptohet procesi, gjatë së cilit, duke marrë si input një imazh dhe një pyetje të shprehur në gjuhë natyrore, agjenti është i aftë t'i përgjigjet pyetjeve rreth imazhit të cilat janë të lidhura me kontekstin e pyetjeve të mëparshme. Ashtu si u diskutua në kapitullin 2, edhe pse përdoret termi *dialog*, aktualisht ky agjent është në gjendje të kthejë vetëm përgjigje dhe jo të bëjë pyetje.

Ndryshimi kryesor midis këtij agjenti dhe atij të paraqitur në kapitullin 5 është se ky agjent mban informacion në lidhje me historikun dhe kontekstin e bisedës. Agjenti jo vetëm mban informacion, por ai është gjithashtu në gjendje të kuptojë kontekstin e bisedës dhe t'i përgjigjet pyetjeve që kanë lidhje më këtë kontekst. Operatori njerëzor mund të përdorë përemra të tillë si "*ai*", "*ajo*", "*ata*", etj., për t'iu referuar objekteve në imazh (*coreference*). Në këto raste agjenti duhet të jetë i aftë të zgjidhë dykuptimësitë (*ambiguity*) që mund të sjellë përdorimi i këtyre përemrave dhe të vendosë referencat e sakta me objektet në imazh.

Për realizimin e këtij modeli, njësoj si për modelin e paraqitur në kapitullin 5, është përdorur qasja statistikore (*data-driven)*. Agjenti është implementuar si një model i një rrjeti neural i cili merr në hyrje një imazh, historikun e bisedës dhe një pyetje rreth imazhit dhe nxjerr si rezultat një sekuencë shpërndarjesh probabilitare që përcaktojnë fjalët e përgjigjes. Modeli përbëhet nga tipe të ndryshme rrjetash neurale të cilët lidhen në kaskadë me njëri-tjetrin. Përpunimi i imazheve bëhet nga një rrjet

neural CNN, ndërsa historiku i bisedës, pyetja aktuale dhe përgjigja përpunohen nga një rrjet neural RNN. Modeli i propozuar trajnohet si një i tërë (*end-to-end*) me *dataset*-in publik VisDial [64]. Ashtu si *dataset*-et e kapitullit 5, edhe ky *dataset* është në gjuhën angleze. Për rrjedhojë modeli njeh dhe gjeneron fjalë në gjuhën angleze, por arkitektura e tij është gjithashtu e pavarur nga gjuha dhe mund të trajnohet në një gjuhë tjetër pa bërë asnjë ndryshim në të.

Pas trajnimit, modeli testohet mbi të njëjtin *dataset* dhe bëhet një analizë sasiore dhe cilësore mbi rezultatet e testimeve. Për të kuptuar më mirë se sa e efektshme është zgjidhja e propozuar, rezultatet e testimeve krahasohen me *state of the art* që gjithashtu përdor mekanizmin e vëmendjes neurale. Arkitektura e këtij agjenti së bashku me rezultatet e testimit dhe konkluzionet përkatëse paraqiten gjithashtu në [85].

6.1 Modeli i Agjentit

Mekanizmi i vëmendjes neurale është përfshirë së fundmi në sistemet e dialogut vizual në [64]. Për të përmirësuar saktësinë e përgjigjeve të gjeneruara nga modeli i tyre, autorët përdorin vëmendjen ndaj historikut të mëparshëm të bisedës. Kjo vëmendje drejtohet nga pyetja aktuale dhe nga nga një *embedding* e përbashkët midis imazhit dhe pyetjes aktuale. Ndryshimi midis këtij modeli dhe modelit të propozuar në këtë disertacion, është se, përveç vëmendjes ndaj historikut të bisedës, agjenti ynë përdor edhe vëmendje ndaj pjesëve të caktuara të imazhit.

Arkitektura e vëmendjes multimodale i mundëson agjentit të fokusohet njëkohësisht në pjesë të veçanta të historikut të bisedës të cilat kanë lidhje me pyetjen aktuale si dhe në zona të veçanta të imazhit për të arsyetuar rreth përgjigjes duke u bazuar në kontekstin e bisedës. Kjo qasje nuk është implementuar më parë në dialogun vizual.

Agjenti ynë inteligjent konsiston në një model i cili kryen arsyetimin rreth imazhit, pyetjes dhe përgjigjes me dy faza: *kodim* dhe *dekodim*. Gjatë fazës së

kodimit, agjenti i transformon të dhënat në hyrje (d.m.th. imazhin, historikun e bisedës dhe pyetjen aktuale) duke i bërë ato *embedd* në një vektor. Gjatë fazës së dekodimit, dekoduesi e konverton këtë vektor në një parashikim në dalje i cili shprehet në formën e një shpërndarjeje probabilitare të përgjigjeve kandidate. Koduesi dhe dekoduesi janë implementuar me anë të rrjetave neurale RNN të cilat lidhen në kaskadë me njëra-tjetrën. Në figurën 6.1 paraqitet dataflow i modelit të propozuar.

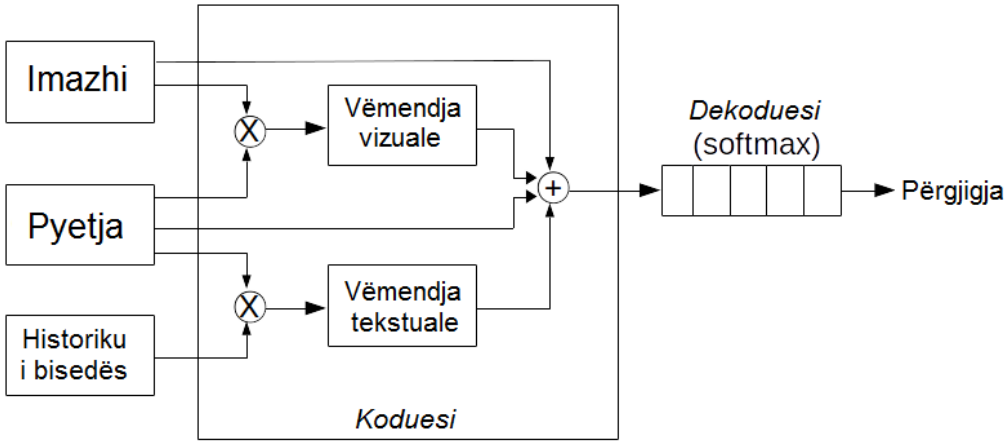


Figura 6.1: Dataflow i modelit. Koduesi përdor imazhin, pyetjen dhe historikun e bisedës për të gjeneruar secilën nga modalitetet e vëmendjes të cilat shtohen në output-in e tij. Dekoduesi përdor një klasifikues *softmax* për të gjeneruar një nga një fjalët e përgjigjes.

6.1.1 Paraqitja e Fjalëve

Në modelin e propozuar, input-i është një imazh i përmasave 224 x 224 piksela, historiku i pyetjeve dhe përgjigjeve të mëparshme, si dhe një pyetje e përbërë nga një sekuencë me gjatësi të ndryshueshme fjalësh.

Gjatë testimit, modeli gjeneron një përshkrim të imazhit (*image caption*) duke përdorur [30]. Në mënyrë që t'i afrohemi sa më shumë situatave reale ku operatori njerëzor mund të ketë probleme shikimi, ai ka informacion vetëm për *image caption* dhe i bën pyetjet agjentit në mënyrë që të kuptojë skenën e imazhit.

Gjatë trajnimit, modeli merr si input historikun e bisedës H, pyetjen Q dhe 100 përgjigje kandidate $A_t = \left\{A_t^{(1)}, \dots, A_t^{(100)}\right\}$. Historiku H përbëhet nga *image caption* C i

ndjekur nga çiftet e mëparshme pyetje-përgjigje $H = (\underbrace{C}_{H_0}, (\underbrace{Q_1, A_1}_{H_1}), \dots, (\underbrace{Q_{t-1}, A_{t-1}}_{H_{t-1}}))$. Çdo fjalë fillimisht transformohet në një vektor *one-hot.* Çdo vektor *one-hot* transformohet më tej (*embedded*) në një vektor me vlera reale $B = \{b_j | \, b_j \in R^D, j = 1, \dots, N\}$ ku N është numri i fjalëve D është dimensioni i hapësirës së transformimit (*embedding space)* dhe $B \in R^{DxN}$. E njëjta procedurë ndiqet për historikun H, pyetjen Q dhe përgjigjet kandidate A_t. Dimensioni i *embedding space* është 300 dhe *embedding* i fjalëve mësohet gjatë trajnimit të modelit.

6.1.2 Paraqitja e Imazheve

Për të bërë paraqitjen e imazheve, është marrë output-i nga shtresa e katërt konvolucionale e rrjetit CNN VGG-16 [31]. Kjo shtresë gjeneron një hartë konvolucionale karakteristikash (*convolutional feature map*) I të imazhit në hyrje, e përbërë nga 196 (14 x 14) pjesë ku secila prej tyre është një vektor me 512 elementë. Kjo hartë karakteristikash do të përdoret për të llogaritur vëmendjen vizuale ku secila nga 196 zonat e imazhit kontribuon në vlerë të ndryshme.

6.1.3 Vëmendja Multimodale

Ashtu si në modelin e paraqitur në kapitullin 5, modalitetet e vëmendjes (d.m.th. tekstuale dhe vizuale) janë të pavarura nga njëra-tjetra dhe agjenti i shfrytëzon këto modalitete për të arsyetuar rreth pyetjes dhe për të vendosur se cila është përgjigja e duhur.

Gjatë trajnimit, modeli merr si input imazhin *I,* historikun e bisedës H, pyetjen Q dhe 100 përgjigje kandidate $A_t = \{A_t^{(1)}, \dots, A_t^{(100)}\}$ dhe i kërkohet të gjenerojë një shpërndarje probabilitare të elementëve të A_t. Imazhi *I* konsiderohet si *token* i parë në hyrje. Ai ndiqet nga përshkrimi i imazhit *C* dhe më tej nga çiftet e mëparshme pyetje-përgjigje që përbëjnë historikun e bisedës. Në fund modeli merr si input një nga një *tokens*-at (d.m.th. fjalët) e pyetjes.

Procesi i gjenerimit të përgjigjes trajtohet si një detyrë klasifikimi. Për këtë arsye përdoret si output një klasifikues *softmax* për gjenerimin dhe përzgjedhjen e

fjalëve të përgjigjes. Gjatë trajnimit, modeli merr gjithashtu në hyrje *tokens-at* (d.m.th. fjalët) e përgjigjes (të bëra *embed* në një vektor). Qëllimi i trajnimit është maksimizimi i probabilitetit (*log-likehood)* të këtyre *tokens*-ave (d.m.th. në shpërndarjen probabilitare të gjeneruar nga modeli, këto *tokens*-a të kenë probabilitetin më të lartë). Gjatë testimit, shpërndarja probabilitare përdoret për të renditur përgjigjet kandidate dhe përgjigja me probabilitet më të lartë përzgjidhet si e saktë.

Si tek [64] vëmendja tekstuale llogaritet si:

$$qh_t = Q_t H_t \tag{6.1}$$

$$r_t^{txt} = softmax\,(W_r^{txt} qh_t)\,. \tag{6.2}$$

$$a_t^{txt} = r_t^{txt} H_t \tag{6.3}$$

$$l_t^{txt} = \tanh(W_a^{txt} a_t^{txt} + b_{txt}) \tag{6.4}$$

Termi r_t^{txt} në ekuacionet 6.2 dhe 6.3 përfaqëson shpërndarjen probabilitare të vëmendjes për çdo raund të bisedës (d.m.th. çdo çift pyetje-përgjigje). Duke u bazuar në këtë shpërndarje probabilitare, vektori i vëmendjes ndaj historisë të bisedës llogaritet si shumë e ponderuar e kësaj shpërndarjeje me *embedding* të historikut të bisedës. Vëmendja ndaj historikut të bisedës (d.m.th. tekstuale) shprehet me anë të termit a_t^{txt} i cili vendos për kontributin e çdo raundi të historikut së bisedës në hapin kohor *t.* Më tej vëmendja tekstuale kalon nëpër një shtresë *fully connected* duke gjeneruar termin l_t^{txt} në ekuacionin 6.4 për të krijuar dimensionalitetin e duhur për t'u mbledhur me vëmendjen vizuale.

Vëmendja vizuale llogaritet si:

$$qi_t = Q_t I \tag{6.5}$$

$$r_t^{img} = softmax\,(W_r^{img} qi_t) \tag{6.6}$$

$$a_t^{img} = r_t^{img} I \tag{6.7}$$

$$l_t^{img} = \tanh(W_a^{img} a_t^{img} + b_{img}) \tag{6.8}$$

Termi r_t^{img} në ekuacionet 6.6 dhe 6.7 përfaqëson shpërndarjen probabilitare të vëmendjes për çdo zonë të imazhit. Duke u bazuar në këtë shpërndarje probabilitare, vektori i vëmendjes ndaj imazhit llogaritet si shumë e ponderuar e kësaj

shpërndarjeje me hartën konvolucionale *I* të karakteristikave të imazhit. Vëmendja vizuale shprehet me anë të termit a_t^{img} i cili është një vektor 196-dimensional dhe vendos për kontributin e çdo karakteristike të imazhit (*image feature*) në hapin kohor *t.* Më tej, vëmendja vizuale kalon nëpër një shtresë *fully connected* duke gjeneruar termin l_t^{img} në ekuacionin 6.8 për të krijuar dimensionalitetin e duhur për t'u mbledhur me vëmendjen tekstuale.

Të dy modalitetet e vëmendjes mblidhen më paraqitjen e pyetjes dhe imazhit. Ekuacioni 6.9 tregon si kodohet input-i i modelit:

$$e_t = \tanh(W_l^{txt} l_t^{txt} \oplus W_l^{img} l_t^{img} \oplus W_e^{txt} Q_t \oplus W_e^{img} I + b) \quad (6.9)$$

Simboli "$\oplus$" përfaqëson mbledhjen e thjeshtë midis elementëve korrespondues të matricave përkatëse (*element-wise*). Output-i i koduesit dërgohet drejt rrjetit LSTM dekodues. Ky i fundit gjeneron sekuencën (d.m.th. fjalët e përgjigjes) në dalje.

Në të gjithë ekuacionet e mësipërme, termat *W* përfaqësojnë parametrat e rrjetit LSTM, ndërsa termat *b* përfaqësojnë *bias*-et e tij. Qëllimi i trajnimit është mësimi (d.m.th. llogaritja) e këtyre termave.

6.2 Optimizimi, Detajet e Implementimit dhe Hiperparametrat e Modelit

Një pjesë e madhe hiperparametrave dhe detajeve të optimizimit të këtij agjenti janë të njëjta me ato të modelit të paraqitur në kapitullin 5. Implementimi i agjentit është bërë në Torch [10] dhe në Python. Përpara trajnimit, fillimisht të gjitha pyetjet konvertohen në shkronja të vogla dhe pikëpyetjet së bashku me të gjitha shenjat e tjera të pikësimit hiqen. Ashtu si për modelin e paraqitur në kapitullin 5, edhe ky model inicializohet sipas algoritmit Xavier [13] për të njëjtën arsye, përveç *embeddings* të cilat inicializohen duke përdorur një shpërndarje uniforme.

Modeli trajnohet me *backpropagation* dhe si funksion humbjeje përdoret entropia e kryqëzuar (*cross-entropy*). Shkalla minimale e të mësuarit (*learning rate*) merret $5 * 10^{-5}$. Gjatë testimit përzgjidhet përgjigja kandidate e cila ka probabilitet më të madh. Gjatë trajnimit, përmasa e batch-it të të dhënave (*batch size*) vendoset 16.

Trajnimi bëhet për 40 epoka duke aplikuar teknikën *early stopping* nëse saktësia e modelit gjatë validimit nuk është përmirësuar në 5 epokat e fundit.

Dimensioni i hapësirës së *embeddings* të fjalëve është 300. Thellësia e të dy rrjeteve LSTM (koduesi dhe dekoduesi) është 380 për të gjitha eksperimentet. Njësoj si për modelin e paraqitur ne kapitullin 5, për të mbajtur nën kontroll overfitting, përdoret *dropout* në rrjetin LSTM me probabilitet 0.5 për çdo shtresë. Gjithashtu për të parandaluar shpërthimin e gradientëve në rrjetin LSTM përdoret teknika e *gradient clipping.*

Për sa i përket rrjetit CNN, përsëri për arsye të mungesës së burimeve të nevojshme hardware për trajnimin e tij, është përdorur një rrjet i paratrajnuar VGG-16 [31] mbi *dataset*-in ImageNet [49].

Imazhet e *dataset*-it ridimensionohen në përmasat 224 x 224 piksela. Për llogaritjen e vëmëndjes vizuale përdoret output-i nga shtresa e katërt konvolucionale e të njëjtit rrjet CNN.

6.3 Ambienti i Punës

Ashtu si u diskutua në kapitullin 5, trajnimi i algoritmave të deep learning kërkon hardware me fuqi të lartë përpunuese. Për këtë arsye, për të realizuar trajnimin e këtyre algoritmave në komunitetin e *machine learning* përdoren gjërësisht hardware të specializuar të cilët shfrytëzojnë fuqinë përpunuese të GPU-ve.

6.3.1 Konfigurimi Hardware

Hardware i përdorur për trajnimin dhe testimin e agjentit është Nvidia Jetson TX1[7] [69]. Moduli Jetson TX1 përmban katër bërthama ARM Cortex-A57 64-bit së bashku me një GPU që përmban 256 bërthama CUDA (Nvidia Tegra X1 GPU) që jep më shumë se 1 TFLOPS fuqi llogaritëse. Moduli ka 4GB kujtesë RAM LPDDR4. Kujtesa RAM nuk ka mundësi shtimi. Moduli ka gjithashtu 16GB hapësirë ruajtje eMMC.

[7] Përdorimi i këtij hardware për trajnimin dhe testimin agjentit të SPPV në kapitullin 5 është i pamundur pasi algoritmi i tij kërkon një kapacitet të kujtesës RAM më të madh nga sa mund të ofrojë Nvidia Jetson TX1.

Për shkak të kufizimeve të modulit nga ana e hapësirës së ruajtjes së të dhënave, një Hard Disc me madhësi prej 256GB SSD është shtuar në një nga ndërfaqet SATA dhe është përdorur si hapësire ruajtje për modelin tonë si dhe për rezultatet e gjeneruara gjatë testimit dhe trajnimit. Gjithashtu edhe për shkak të kufizimeve të kujtesës RAM së modulit dhe kërkesës së lartë për RAM nga modeli ynë, gjatë trajnimit dhe testimeve është konfiguruar një Flash Drive me madhësi prej 16GB në portën USB 3.0 si hapësirë Swap e përdorur nga sistemi operativ Linux.

6.3.2 Konfigurimi Software

Nvidia[8] ofron një paketë software e cila përfshin të gjithë software-t që nevojiten në mënyrë që të shfrytëzohet maksimalisht moduli TX1. Paketat e mëposhtme janë instaluar, konfiguruar dhe përdorur në testimet dhe eksperimentet e bëra:

- **JetPack L4T 2.3.1.** Është një paketë instalimi e përdorur për të flash-uar Jetson Development Kits me imazhin më të fundit të sistemit operativ, tools-e zhvillimi, libraritë, dokumentacionin dhe shembuj të ndryshëm.
- **CUDA Toolkit for Ubuntu 14.04 v8.0.34**. Është një platformë llogaritjeje paralele dhe API e krijuar nga Nvidia. Ajo lejon zhvilluesit software të përdorin njësitë e përpunimit grafik (GPU) që suportojnë CUDA, të krijojnë software që kërkon fuqi të lartë llogaritëse.
- **Linux for Tegra (TX1 64-bit Ubuntu 16.04 LTS aarch64, kernel version 3.10.96-tegra)**
- **Driver for OS v24.2.1**
- **FileSystem v24.2.1**
- **cuDNN v5.1**

Këto paketa software janë përdorur sepse ato janë versionet më të fundit të përshtatshme (compatible) më paketat software të përdorura për implementimin, trajnimin dhe testimin e modelit.

[8] http://www.nvidia.com

Paketat software të mëposhtme janë instaluar (së bashku me *dependencies* të tyre përkatëse) dhe përdorur për implementimin, trajnimin dhe testimin e modelit:

- Torch 7
- Python 2.7
  - paketa të nevojshme: h5py, numpy, skimage
- Luajit 2.1.0-beta1 (e instaluar si paketë pas instalimit të Torch)
  - paketa të nevojshme: torch, nn, nngraph, hdf5, loadcaffe, cjson, image
  - paketa opsionale: cutorch, cunn, cudnn (për suport GPU)

6.4 Dataset-i dhe Metrikat e Përdorura

Trajnimi dhe testimi i agjentit është bërë mbi *dataset*-in publik VisDial v0.9 [64]. Ky *dataset* është përdorur për të patur një krahasim sa më të ndershëm me modelin aktual të *state of the art* [64]. Gjithashtu ky është *dataset*-i i vetëm i disponueshëm për dialogun vizual deri në momentin e realizimit të këtij punimi. *Dataset*-et VQA dhe Visual7W janë krijuar posaçërisht për sistemet pyetje-përgjigje vizuale dhe nuk janë të përshtatshme për trajnimin apo testimin e modeleve të dialogut dialogut vizual pasi në këto *dataset*-e nuk ekziston koncepti i dialogut me pyetje të vazhdueshme rreth të njëjtit imazh dhe çdo pyetje trajtohet si e pavarur nga të tjerat. Për këtë arsye, në këto *dataset*-e nuk ekzistojnë shembuj dialogësh me anë të të cilëve të mund të realizohet trajnimi dhe testimi i modeleve të agjentëve të dialogut vizual.

6.4.1 Dataset-i VisDial

Dataset-i VisDial v0.9 është ndërtuar duke përdorur *dataset*-in Microsoft COCO [33]. Ai përmban 123,287 imazhe. Ekziston nga 1 dialog për çdo imazh. Çdo dialog përmban 10 çifte pyetje-përgjigje. Në total, *dataset*-i përmban 1,232,870 çifte pyetje përgjigje. Ky *dataset* është i ndarë në dy pjesë: *trajninim* dhe *validim*. Pjesa e trajnimit përmban 82,783 imazhe që zënë një hapësire prej 235 MB. Pjesa e testimit përmban 40,504 imazhe të cilat zënë një hapësire prej 108 MB.

6.4.2 Metrikat e Përdorura

Në vend që të bëjmë vlerësimin e agjentit në lidhje me suksesin e realizimit të një detyrë te caktuar, njësoj siç do bënim vlerësimin e një dialogu i cili do kishte një objektiv të caktuar (*goal-oriented dialogue*), apo të vlerësojmë në përgjithësi cilësinë e bisedës, njësoj siç do bënim vlerësimin e një dialogu i cili nuk do kishte një objektiv të caktuar (*goal-free dialogue*), ne vlerësojmë përgjigjet për çdo raund të bisedës. Ne përdorim metrikat e propozuara nga Das et. al [64] për të vlerësuar agjentin e propozuar. Këto metrika klasifikohen si *retrieval metrics*. Ato përdoren për të vlerësuar aftësinë e modelit për përzgjedhjen e një përgjigjeje të saktë nga një bashkësi përgjigjesh. Më konkretisht, përdorim ***recall @ k*** (ekzistenca e përgjigjes njerëzore në *k* përgjigjet e para në listë të renditura nga agjenti) dhe ***mean reciprocal rank*** (MRR) (renditja mesatare reciproke) e përgjigjes njerëzore (sa më e lartë të jetë vlera, aq më i mirë është rezultati). Të gjitha metrikat ***recall @ k*** shprehen në përqindje.

6.5 Rezultatet e Vlerësimit Sasior

Rezultatet e testimit dhe krahasimi me *state of the art* paraqiten në tabelën 6.1. Nga këto rezultate vihet re se modeli i propozuar ka performancë më të mirë se *state of the art* për të gjitha metrikat. Termi *performancë* i referohet rezultateve të testimeve bazuar në metrikat e përdorura.

Tabela 6.1: Performanca e modeleve në dataset-in VisDial v0.9 e matur me *mean reciprocal rank* (MRR) dhe *recall @ k*. Për *MRR* dhe *recall @ k*, vlera më e lartë tregon performancë më të mirë. Vlerat e *recall@k* paraqiten në përqindje.

Modeli	MRR	R@1	R@5	R@10
Visdial [64]	0.4599	35.28	55.66	60.70
Modeli ynë	**0.5190**	**41.89**	**61.57**	**67.29**

MRR është rendi mesatar reciprok i përgjigjes së parë të saktë në shpërndarjet probabilitare të përgjigjeve të gjeneruara nga modeli. Një MRR me vlerë 1 do të thotë se mesatarisht përgjigja e saktë është përgjigja e renditur e para në shpërndarjen

probabilitare të përgjigjeve të gjeneruara nga modeli. Nga tabela 6.1 mund të shihet që modeli ynë ka një performancë 12.8% më të lartë se *state of the art*. Kjo tregon se pozicioni i përgjigjes së saktë të gjeneruar nga modeli ynë është mesatarisht 5.9% më lart në renditje dhe më pranë përgjigjes së renditur e para.

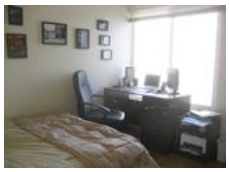
Recall @ k mat praninë e përgjigjes njerëzore (d.m.th. përgjigjes së saktë) në *k* përgjigjet e para në renditjen e gjeneruar nga agjenti. Rezultatet tregojnë se modeli ynë ka një performancë më të lartë në këtë metrikë krahasuar me *state of the art* me të paktën 5.9%. Sipas tabelës 6.1, në 41.89% të rasteve përgjigja e renditur e para nga modeli ynë është e njëjtë me përgjigjen njerëzore. Kjo është ekuivalente me 6.61% përmirësim të *state of the art*. Ashtu siç pritet, me rritjen e numrit të përgjigjeve të marra në konsideratë, prania e përgjigjes njerëzore rritet në 61.57% dhe 67.29% respektivisht për 5 dhe 10 përgjigjet e para të renditura nga agjenti për çdo pyetje. Krahasuar me *state of the art*, kjo përbën një përmirësim prej 5.91% dhe 6.59% respektivisht për 5 dhe 10 përgjigjet e para.

Rezultatet e mësipërme tregojnë se integrimi i vëmendjes multimodale është i suksesshëm dhe ndihmon agjentin të arsyetojë më mirë dhe të përmirësojë saktësinë e përgjigjes.

6.6 Rezultatet e Vlerësimit Cilësor

Në mënyrë që të krijohet një ide më e mirë e sjelljes se modelit të propozuar, dhe për të parë se si mund të përmirësohet ai në të ardhmen, nevojitet një vlerësim cilësor i tij duke analizuar përgjigjet e gjeneruara nga agjenti gjatë bisedave. Në tabelën 6.2 tregohen disa prej këtyre përgjigjeve të përzgjedhura në mënyrë rastësore. Përshkrimi i imazheve (*caption*) është gjeneruar duke përdorur [30]. Duke qenë se modeli është trajnuar gjuhën angleze, shembujt e bisedës janë lënë në gjuhën origjinale.

Tabela 6.2: Shembuj bisedash për dataset-in Visdial.

Imazhi + caption	Biseda
 1. A dog is sitting near a bike in the street	1. Is it a color photo? Yes 2. What color is the dog? White 3. What size is the dog? Small 4. Are you able to tell what breed? No 5. Does the puppy have a human with him? No 6. Does he have a collar? No 7. Does he have a leash? No 8. Is the puppy facing the camera? No 9. Are you able to see the color of his eyes? No 10. Is the setting in a park? No
 2. A baseball player is swinging at a ball	1. What is the weather like? Sunny 2. Do the fans look happy? Yes 3. Are people wearing sunglasses? No 4. What color are the baseball players' uniforms? Blue and white 5. Can you see the whole ball field? No 6. Can you see the score of the game? No 7. Can you see the pitcher? No 8. Are there kids there watching? Yes 9. Do you see any cars? No 10. Is it sunny out? Yes
 3. Bikes parked near metallic bench	1. Are the bikes the same color? No 2. Can you see any buildings? Yes 3. What color are the bikes? Black 4. Can you see any people? No 5. Can you see the street? Yes 6. Are there lots of buildings? Yes 7. Can you see any vehicles? No 8. Does it look nice out? Yes 9. Can you see the sky? No 10. Is it sunny? Yes
	1. Is the photo in color? Yes 2. Is it a professional photo? No 3. Is it well lit? Yes 4. Is it daytime? Yes 5. What color is the chair? Brown 6. What's the chair made of? Wood

4. A bedroom with a bed and a desk	7. Does the chair have arms? Yes 8. Is the desk large? No 9. Does the desk look expensive? No 10. Is there anything on the desk? Yes
 5. A group of birds sitting on a tree branch	1. Is the photo in color? Yes 2. Is there any people? No 3. Are the birds the same size? Yes 4. Are they the same color? Yes 5. Are they in a cage? No 6. Is this at a zoo or in the wild? Can't tell 7. Is there any buildings? No 8. Is it daytime? Yes 9. Is there any poles around? No 10. Is the sky visible? No

Nga përgjigjet e gjeneruara nga agjenti në këta shembuj bisedash vihen re karakteristikat e mëposhtme:

- Agjenti është i aftë të kuptojë mirë situatën e imazhit (*image scene understanding*) dhe lidhjet e objekteve me njëri-tjetrin (p.sh. imazhi 2: pyetjet 5, 6 dhe 7, imazhi 3: pyetjet 4, 5, 6, 7 dhe 8, imazhi 5: pyetja 6).
- ***Kuptim i mirë i kontekstit të bisedës dhe koreferencës***. Agjenti është i aftë të mbajë kontekstin e bisedës dhe të integrojë në imazh në mënyrë të saktë referencat e objekteve të bëra duke përdorur përemra. Agjenti zgjidh referencën dhe është i aftë të fokusojë vëmendjen e tij në objektin e duhur dhe të japë përgjigje të saktë (p.sh. imazhi 1: pyetjet 4, 6, dhe 7, imazhi 4: pyetja 3, imazhi 5: pyetjet 4 dhe 5).
- ***Aftësi e mirë për të dalluar objektet në plan të parë (foreground) dhe karakteristikat e tyre.*** Agjenti është i aftë të fokusojë në mënyrë të saktë vëmendjen e tij për të dalluar objektet në plan të parë nga ato në sfond dhe për të dalluar karakteristikat e tyre si ngjyra, përmasa, etj.
- ***Aftësi e mirë për të kuptuar organizimin e ngjyrave të imazhit.*** Nga rezultatet vihet re se agjenti i përgjigjet në mënyrë të saktë pyetjeve që kanë lidhje me ngjyra të imazhit (p.sh. imazhi 1: pyetja 1, imazhi 4: pyetja 1, imazhi 5: pyetja 1).

Nga rezultatet e tabelës 6.2 vihet re se ka disa gabime të pranishme në përgjigjet e gjeneruara nga agjenti gjatë bisedave. Kështu, agjenti nuk arsyeton në mënyrë të saktë për objektet që janë në sfond (p.sh. imazhi 2: pyetja 3, imazhi 3: pyetjet 1 dhe 3, imazhi 4: pyetjet 5 dhe 6). Kjo tregon se vëmendja funksionon më mirë për objektet në plan të parë dhe dobësohet për objektet në sfond ose objektet që duket sikur shkrihen me sfondin.

6.6.1 Vlerësimi Cilësor i Gabimeve

Në mënyrë që të krijohet një ide më e mirë e gabimeve të bëra nga agjenti dhe për të parë se si mund të përmirësohet ai në të ardhmen, është bërë një analizë e bisedave ku ai kthen përgjigje të gabuara. Tabela 6.3 tregon disa prej këtyre bisedave të përzgjedhura në mënyrë rastësore.

Tabela 6.3: Shembuj bisedash me përgjigje të gabuara për dataset-in Visdial.

Imazhi + caption	Biseda
 1. A group of bulls standing in a barn	1. Is this in color? Yes 2. Do you see any people? There are no people 3. How many bulls? 2 4. Do any have horns? No 5. What color are the bulls? Brown 6. Are they indoors? Yes 7. What color are the walls? White 8. Do you see a fence? Yes 9. Are the bulls standing? Yes they are 10. Is the photo well lit? Yes it is
 2. A man riding a skateboard on a ramp	1. How old is the skateboarder? Can't tell 2. What is the skateboarder's gender? Male 3. What is the skateboarder wearing? Shirt and pants 4. Do you see any other people? No 5. How many other people? 1 6. Are the people watching the skateboarder? No 7. Is the skateboarder in the air? Yes 8. Is this at a skateboard park? No 9. Is it a sunny day? Yes 10. Does the skateboarder look like he's going to have a smooth landing? No

 3. A group of people riding skis down a snow covered slope	1. Is the photo in color? Yes 2. How many people? 2 3. Is the sun out? Yes 4. How is the weather? Sunny 5. What color is his board? Black 6. What color is his coat? Black 7. Is he wearing pants? Yes 8. What color pants? Black 9. Is there trees? Yes 10. Are there leaves on the trees? Yes
 4. A group of children sitting at a table with a cake	1. How many kids? 3 2. Is this birthday party? No 3. Any adults? No 4. Is this outdoors? No 5. Is table round? Yes 6. Is there tablecloth? No 7. Are kids wearing hats? No 8. Are smiling? Yes 9. Does cake have candles? No 10. Are they lit? No
 5. A train is coming down the tracks in a city	1. Is this a passenger train? Yes 2. Is it moving? No 3. Is it at a station? No 4. Any people around? No 5. Does the station have a roof? No 6. What color is the platform? Green 7. Any seats? No 8. Is it daytime? Yes 9. Is it sunny? Yes 10. Any lights on the train? Yes

Nga përgjigjet e gjeneruara nga agjenti në këta shembuj përgjigjesh të pasakta vihen re karakteristikat e mëposhtme:

- ***Vështirësi për të identifikuar objektet në sfond***. Agjenti nuk është i aftë të dallojë nga njëri-tjetri në mënyrë të saktë objektet në sfond. Vëmendja e fokusimit dobësohet për këto lloj objektesh dhe agjenti nuk mund të arsyetojë saktë rreth tyre (p.sh. imazhi 1: pyetja 3, imazhi 2: pyetjet 4 dhe 6).

- ***Vështirësi për të numëruar objektet në sfond.*** Agjenti fokusohet dhe numëron në mënyrë të saktë objektet në plan të parë, por aftësia numëruese zbehet ndërsa objektet bëhen njësh me sfondin (p.sh. imazhi 1: pyetja 3, imazhi 3: pyetja 2). Ky problem vjen gjithashtu edhe nga vështirësia e identifikimit të objekteve në sfond.
- ***Aftësia e numërimit dobësohet për objektet jo të plota.*** Agjenti është i aftë të fokusohet, identifikojë dhe numërojë objektet e plota në imazh, por ka vështirësi për të dalluar dhe arsyetuar rreth objekteve të cilat shfaqen jo të plota në imazh (p.sh. imazhi 4: pyetja 1 dhe 3).
- ***Vështirësi për të dalluar nga njëri-tjetri objektet që duken "të shkrirë" me njëri-tjetrin.*** Vëmendja nuk funksionon në mënyrë të saktë për të bërë dallimin dhe arsyetimin e duhur rreth këtyre objekteve në imazh (p.sh. imazhi 1: pyetja 3 dhe 4, imazhi 5: pyetja 5 dhe 6, imazhi 4: pyetja 9).

6.7 Diskutime dhe konkluzione

Në këtë kapitull u paraqit arkitektura dhe implementimi i një agjenti inteligjent me vëmendje multimodale për dialogun vizual. Agjenti shfrytëzon njëkohësisht vëmendjen tekstuale (d.m.th. vëmendjen ndaj historikut të bisedës) dhe vëmendjen vizuale (d.m.th vëmendjen ndaj imazhit) për të mbajtur kontekstin e bisedës dhe për ta lidhur atë më imazhin.

Agjenti u vlerësua në mënyrë sasiore mbi *dataset*-in publik VisDial duke përdorur *retrieval metrics* dhe rezultatet e testimeve tregojnë se ai ka performancë më të mirë dhe përmirëson *state of the art* për të gjitha metrikat sipas rezultateve të tabelës 6.1. Përmirësimin më të madh të *state of the art* modeli e arrin për metrikën *recall@1* me 6.61%. Këto rezultate tregojnë se arkitektura risi e propozuar me vëmendje multimodale është e suksesshme dhe ndihmon agjentin të arsyetojë më mirë dhe të përmirësojë saktësinë e përgjigjes.

Përgjigjet dhe gabimet e agjentit u vlerësuan gjithashtu në mënyrë cilësore për të kuptuar më mirë sjelljen e tij dhe si mund të përmirësohet në të ardhmen. Nga

analiza e këtyre rezultateve u arrit në konkluzionin se agjenti i propozuar është i aftë të përdorë në mënyrë të saktë dhe përfiton nga vëmendja multimodale në disa aspekte. Ai është i aftë:

- Të kuptojë skenën e imazhit (*image scene understanding*) dhe lidhjen që kanë objektet me njëri-tjetrin.
- Të kuptojë kontekstin e bisedës dhe *coreference*-n dhe të bëje lidhje të saktë midis referencave ndaj objekteve (p.sh. përemrat) dhe objekteve në imazh.
- Të dallojë objektet në plan të parë dhe karakteristikat e tyre si përmasa, ngjyra, etj.
- Të dallojë organizimin e ngjyrave në imazh.

Nga analiza e përgjigjeve të gabuara u arrit në konkluzionin se agjenti ka gjithashtu edhe disa limitime. Kështu, ai ka vështirësi:

- Të identifikojë dhe numërojë objektet në sfond.
- Të numërojë objekte të cilët nuk shfaqen të plotë në imazh.
- Të dallojë nga njëri-tjetri objektet të cilët në imazh shfaqen si “të shkrirë” me njëri-tjetrin.

Këto vështirësi tregojnë nevojën për të përmirësuar mekanizmin e vëmendjes dhe tejkalimi i tyre është objekt i punës në të ardhmen.

7

Konkluzione

dhe Puna në të Ardhmen

Në këtë disertacion u paraqit modeli, implementimi dhe testimi i dy agjentëve inteligjentë të cilët realizojnë detyra që qëndrojnë në pikën e takimit midis *vizionit kompjuterik, machine learning* dhe *natural language processing*. Fillimisht u paraqit një agjent i sistemeve pyetje-përgjigje vizuale cili është i aftë t'i përgjigjet pyetjeve rreth një imazhi të bëra në gjuhë natyrore nga një operator njerëzor. Agjenti është një model i ndërtuar me rrjeta neurale artificiale CNN dhe RNN dhe përdor mekanizmin e vëmendjes neurale multimodale për t'u fokusuar njëkohësisht në fjalë të caktuara të pyetjes dhe në zona të caktuara të imazhit për të arsyetuar dhe për të përmirësuar saktësinë e përgjigjes. Vëmendja multimodale u integrua në një arkitekturë risi LSTM e cila përfshin vëmendjen vizuale dhe tekstuale në portat e çdo qelize LSTM. Përgjigjet e gjeneruara nga agjenti duke përdorur këtë arkitekturë risi u vlerësuan në mënyrë sasiore dhe cilësore për *dataset*-et publike VQA dhe Visual 7W, si dhe u krahasuan me modele *state of the art*. Rezultatet e vlerësimit sasior treguan se modeli i propozuar është më i efektshëm dhe përmirëson *state of the art* për të gjitha metrikat e testimit dhe llojet e pyetjeve. Përjashtim bën rasti i pyetjeve rreth numrit të objekteve në të cilin modeli i propozuar has vështirësi për të fokusuar vëmendjen dhe për të arsyetuar në mënyrë të saktë. Përmirësimi i aftësisë numëruese të agjentit është subjekt i punës në të ardhmen. Përgjigjet e agjentit u vlerësuan në mënyrë cilësore për të kuptuar më mirë sjelljen e tij dhe për të parë ku mund të përmirësohej ai në të ardhmen. Rezultatet e vlerësimit treguan se agjenti përdor në mënyrë të saktë dhe përfiton nga arkitektura e propozuar dhe është i aftë:

- Të fokusojë vëmendjen, identifikojë dhe arsyetojë në mënyrë të saktë rreth objekteve në plan të parë (*foreground*) dhe karakteristikave të tyre.
- Të fokusojë vëmendjen, identifikojë dhe arsyetojë rreth objekteve në sfond të cilat janë qartësisht të dallueshëm nga njëri-tjetri.

Nga analiza e përgjigjeve të gabuara u arrit në konkluzionin se agjenti gjithashtu ka limitime dhe vështirësi:

- Të fokusojë vëmendjen, identifikojë dhe arsyetojë rreth objekteve të cilët nuk shfaqen të plotë në imazh ose shfaqen "të shkrirë" me njëri-tjetrin dhe/ose planin e parë/sfondin.
- Të arsyetojë në mënyrë të saktë rreth ngjyrës së objekteve.
- Të arsyetojë në mënyrë të saktë në lidhje me pyetjet *pse*.

Tejkalimi i këtyre limitimeve është gjithashtu subjekt i punës në të ardhmen.

Modeli i dytë i paraqitur është një agjent i dialogut vizual. Ai qëndron në një nivel më të lartë abstraksioni se agjenti i parë pasi ai jo vetëm i përgjigjet pyetjeve në gjuhën natyrore rreth një imazhi, por gjithashtu është i aftë të ruajë kontekstin e pyetjeve të mëparshme në mënyrë që t'i përgjigjet pyetjeve pasardhëse të cilat mund të jenë vijueshmëri logjike e pyetjeve të mëparshme. Duke u bazuar në kontekstin e historikut të bisedës (d.m.th. kontekstin e pyetjeve dhe përgjigjeve të mëparshme) agjenti është i aftë të vendosë gjithashtu përkatësinë midis objekteve të imazhit dhe referencave të tyre në pyetjet e përdoruesit të cilat mund të jenë të shprehura me përemrat "*ai*", "*ajo*", "*ata*", etj. Agjenti është implementuar me rrjeta neurale CNN dhe RNN dhe përdor gjithashtu mekanizmin e vëmendjes multimodale. Ky mekanizëm u propozua si risi për dialogun vizual pasi ai nuk ishte eksploruar më parë për këto lloj sistemesh. Agjenti u vlerësua në mënyrë sasiore dhe cilësore mbi *dataset*-in VisDial dhe rezultatet e testimeve u krahasuan me *state of the art*. Nga vlerësimi sasior u arrit në konkluzionin se arkitektura e propozuar është e suksesshme dhe përmirëson *state of the art* në të gjitha metrikat e vlerësimit. Për sa i përket vlerësimit cilësor, nga analizat e përgjigjeve të gjeneruara prej modelit të propozuar u arrit në

konkluzionin se ai përdor në mënyrë të saktë dhe përfiton nga vëmendja multimodale duke qenë i aftë:

- Të kuptojë skenën e imazhit (*image scene understanding*) dhe lidhjen që kanë objektet me njëri-tjetrin.
- Të kuptojë kontekstin e bisedës dhe *coreference*-n dhe të bëje lidhje të saktë midis referencave ndaj objekteve (p.sh. përemrat) dhe objekteve në imazh.
- Të dallojë objektet në plan të parë dhe karakteristikat e tyre si përmasa, ngjyra, etj.
- Të dallojë organizimin e ngjyrave në imazh.

Analiza e përgjigjeve të gabuara nxorri në pah të metat dhe vështirësitë e agjentit për:

- Të identifikuar dhe numëruar objektet në sfond.
- Të numëruar objekte të cilët nuk shfaqen të plotë në imazh.
- Të dalluar nga njëri-tjetri objektet të cilët në imazh shfaqen si "të shkrirë" me njëri-tjetrin.

7.1 Puna në të ardhmen

Ky punim mund të zgjerohet më tej në të ardhmen në disa drejtime:

1. Krijimi i *dataset*-eve në gjuhën shqipe për sistemet pyetje-përgjigje vizuale dhe dialogut vizual.
2. Trajnimi dhe testimi i agjentëve mbi *dataset*-e në gjuhën shqipe.
3. Përmirësimi i aftësisë numëruese të objekteve në plan të parë dhe në sfond si dhe objekteve që në imazh shfaqen të paplotë.
4. Përmirësimi i aftësisë shquese të objekteve në sfond.
5. Përmirësimi i aftësise shquese të objekteve të cilat në imazh shfaqen "të shkrirë" me njëri-tjetrin.
6. Përmirësimi i aftësise shquese të objekteve te cilat në imazh shfaqen të paplotë.

7. Hulumtimi i mundësive për të futur mendimin praktik (*common sense*) në të dyja modelet e agjentëve në mënyrë që të rritet saktësia e përgjigjes.

Referenca

[1] K. He, X. Zhang, Sh. Ren, and J. Su, “Deep residual learning for image recognition”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[2] L. Anne Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, and T. Darrell, “Deep compositional captioning: Describing novel object categories without paired training data”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[3] J. B. Delbrouck, S. Dupont, “Multimodal compact bilinear pooling for multimodal neural machine translation”, International Conference on Learning Representations (ICLR), 2017

[4] P-Y. Huang, F. Liu, Sz-R. Shiang, J. Oh, and C. Dyer, “Attention-based multimodal neural machine translation”, Proceedings of the First Conference on Machine Translation (WMT), 2016.

[5] O. Caglayan, W. Aransa, Y. Wang, M. Masana, M. García-Martínez, F. Bougares, L. Barrault, and J. van de Weijer, “Does multimodality help human and machine for translation and image captioning?”, arXiv preprint arXiv:1605.09186, 2016.

[6] N.Hyeonseob, H.Jung-Woo, K. Jeonghee, “Dual Attention Networks for Multimodal Reasoning and Matching”, arXiv:1611.00471, 2017

[7] Y.Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, “Visual7w: Grounded question answering in images”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory. Neural computation”, 9(8):1735–1780, 1997.

[9] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “Vqa: Visual question answering”, arXiv preprint arXiv:1505.00468, 2015.

[10] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning”, BigLearn, Conference on Neural Information Processing Systems (NIPS) Workshop, 2011.

[11] B. F. Green, Jr., A. K. Wolf, C. Chomsky, and K. Laughery. “Baseball: An automatic question-answerer”, Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, 1961.

[12] Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao,K., Gonina, K., Jaitly, N., Li, B., Chorowski, J., Bacchiani, M., "State-of-the-art speech recognitionwith sequence-to-sequence models.", arXiv:1712.01769 (2017)

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 249–256, 2010.

[14] J. Lu, J. Yang, Dh. Batra, and D. Parikh, "Hierarchical Question-Image Co-Attention for Visual Question Answering", 30th Conference on Neural Information Processing Systems (NIPS), 2016.

[15] H. Noh, P. Hongsuck Seo, and B. Han, "Image Question Answering using Convolutional Neural Network with Dynamic Parameter Prediction,", Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to Compose Neural Networks for Question Answering" Proceedings of NAACL-HLT, pp. 1545–1554, 2016.

[17] H. Xu, and K. Saenko, "Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering", European Conference on Computer Vision (ECCV), 2016.

[18] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked Attention Networks for Image Question Answering", Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[19] C. Xiong, S. Merity, and R. Socher, "Dynamic Memory Networks for Visual and Textual Question Answering", International Conference on Machine Learning (ICML), 2016.

[20] R. Li, and J. Jia, "Visual Question Answering with Question Representation Update (QRU)", 30th Conference on Neural Information Processing Systems (NIPS), 2016.

[21] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask Your Neurons: A Neural-Based Approach to Answering Questions about Images", IEEE International Conference on Computer Vision (ICCV), 2015.

[22] A. Fukui, D. Huk Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding", Empirical Methods in Natural Language Processing (EMNLP), 2016.

[23] M. Ren, R. Kiros, and R. S. Zemel, "Exploring models and data for image question answering", Conference on Neural Information Processing Systems (NIPS), 2015.

[24] Clark, P., Etzioni, O., Khot, T., Sabharwal, A., Tafjord, O., Turney, P. “Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions”, AAAI Conference on Artificial Intelligence, 2016.

[25] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, “Are you talking to a machine? Dataset and methods for multilingual image question answering”, Conference on Neural Information Processing Systems (NIPS), 2015.

[26] L. Ma, Z. Lu, and H. Li, “Learning to answer questions from image using convolutional neural network,” AAAI Conference on Artificial Intelligence, 2016.

[27] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator”, Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3156–3164, 2015.

[28] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell:Neural image caption generation with visual attention”, arXiv preprint arXiv:1502.03044, 2015.

[29] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al, “From captions to visual concepts and back”, Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 1473–1482, 2015.

[30] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3128–3137, 2015.

[31] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, International Conference on Learning Representations (ICLR), 2014.

[32] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description”, Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[33] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C.Lawrence Zitnick, “Microsoft COCO: Common objects in context”, European Conference on Computer Vision (ECCV), pp. 740-755, 2014.

[34] Pang, L., Lan, Y., Guo, J., Xu, J., Cheng, X.: SPAN, “Understanding a Question with Its Support Answers”, AAAI Conference on Artificial Intelligence (2016).

[35] “Connecting images and natural language”, Disertacion nga Andrej Karpathy, Universiteti i Stanfordit, 2016 (https://purl.stanford.edu/wf528qt3314 aksesuar në 08.01.2018).

[36] CS231n: Convolutional Neural Networks for Visual Recognition (http://cs231n.stanford.edu/index.html aksesuar në 08.01.2018).

[37] Machine Learning (https://www.coursera.org/learn/machine-learning aksesuar në 08.01.2018).

[38] Rrjetat Neurale Artificiale (http://colah.github.io/ aksesuar në 08.01.2018).

[39] Ian Goodfellow Yoshua Bengio dhe Aaron Courville. "Deep learning" (http://www.deeplearningbook.org aksesuar në 08.01.2018).

[40] Wolpert, David ,"The Lack of *A Priori* Distinctions between Learning Algorithms", *Neural Computation*, (1996) pp. 1341-1390

[41] David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams, "Learning representations by back-propagating errors", Jornal Nature, 1986

[42] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, 86(11):2278–2324, 1998.

[43] http://cs231n.stanford.edu/reports/2016/pdfs/107_Report.pdf (aksesuar në 08.01.2018)

[44] L., Kodra, E. K. Meçe, "Question Answering Systems: A Review on present developments, challenges and trends", International Journal of Advanced Computer Science and Applications (IJACSA) - Volume 8, Number 9, September 2017

[45] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks", Proceedings of the International Conference on Machine Learning (ICML), 2013.

[46] L., Kodra, E., K,. Meçe, "A Review on Neural Network Question Answering Systems", International Journal of Artificial Intelligence and Applications (IJAIA), Volume 8, Number 2, March 2017

[47] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", Journal of Machine Learning Research 15 (2014) 1929-1958

[48] Lua (http://www.lua.org/ aksesuar në 08.01.2018).

[49] ImageNet Dataset (www.image-net.org aksesuar në 08.01.2018).

[50] HP ProLiant DL360 Generation 7 https://h20195.www2.hpe.com/v2/getpdf.aspx/c04284501.pdf?ver=37 (aksesuar në 08.01.2018).

[51] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012

[52] P. Gupta and V. Gupta, “A survey of text question answering techniques” International Journal of Computer Applications, 53(4):1-8, 2012.

[53] S. R. Petrick, “On natural language based computer systems”, IBM J. Res. Dev., 20(4):314-325, July 1976.

[54] G. M. Mazzeo, C. Zaniolo , “Answering Controlled Natural Language Questions on RDF Knowledge Bases”, 19th International Conference on Extending Database Technology (EDBT), 2016

[55] RDF 1.1 Concepts and Abstract Syntax https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225 (Aksesuar në 08.01.2018)

[56] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix, and C. Lange, “Qanary – A Methodology for Vocabulary-Driven Open Question Answering Systems”, The Semantic Web. Latest Advances and New Domains, Volume 9678 of the series Lecture Notes in Computer Science pp 625-641, 14 May 2016

[57] Watson https://www.lt.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_LangTech/data/Watson2015_Prak_intro_1.pdf (Aksesuar në 08.01.2018)

[58] “A Natural Language Question and Answer System”, Chris Callison-Burch and Philip Shilane June 3, 2000 http://nlp.stanford.edu/courses/cs224n/2000/philips/project.pdf (Aksesuar në 08.01.2018)

[59] R. Yan, Y. Song, H. Wu ,“Learning to Respond with Deep Neural Networks for Retrieval-Based Human-Computer Conversation System”, Special Interest Group on Information Retrieval (SIGIR), 2016.

[60] B. Hixon, P. Clark, H. Hajishirzi, “Learning Knowledge Graphs for Question Answering through Conversational Dialog”, Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics (ACL), pages 851–861, 2015.

[61] Zhao, Zhou & Yang, Qifan & Cai, Deng & He, Xiaofei & Zhuang, Yueting “Expert Finding for Community-Based Question Answering via Ranking Metric Network Learning”, International Joint Conferences on Artificial Intelligence (IJCAI), 2016.

[62] Unger, C.; Freitas, A.; and Cimiano, P., “An introduction to question answering over linked data”, Reasoning on the Web in the Big Data Era. 100–140, 2014.

[63] Das, A., Kottur, S., Moura, J. M.F., Lee, S., Batra, D., “Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning”, International Conference on Computer Vision (ICCV), 2017.

[64] Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M.F., Parikh, D., Batra, D., "Visual Dialog", International Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[65] de Vries, H., Strub, F., Chandar, S., Pietquin, O., Larochelle, H., Courville, A. "GuessWhat?! Visual object discovery through multi-modal dialogue", International Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[66] Strub, F., de Vries, H., Mary, J., Piot, B., Courville, A., Pietquin, O. "End-to-end optimization of goal-driven and visually grounded dialogue systems" arXiv:1703.05423, 2017.

[67] Chattopadhyay, P., Yadav, D., Prabhu, V., Chandrasekaran, A., Das, A., Lee, S., Batra, D., Parikh, D.,"Evaluating Visual Conversational Agents via Cooperative Human-AI Games", International Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[68] J. Weizenbaum. ELIZA. (http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm aksesuar në 08.01.2018)

[69] Berkeley Design Technology: A Test Drive of the NVIDIA Jetson TX1 Developer Kit for Deep Learning and Computer Vision Applications, https://www.bdti.com/MyBDTI/pubs:Nvidia_JetsonTX1_Kit.pdf (aksesuar në 08.01.2018)

[70] Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler. S., "MovieQA: Understanding Stories in Movies through Question-Answering" International Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[71] Tu, K., Meng, M., Lee, M. W., Choe, T. E., Zhu, S. C., "Joint Video and Text Parsing for Understanding Events and Answering Queries", IEEE MultiMedia, 2014.

[72] L., Kodra, E., K., Meçe, "A review on current research in Question Answering Systems and future trends", 11th Doctoral Student Conference Thessaloniki, Greece, 2017.

[73] T. Paek, "Empirical methods for evaluating dialog systems", In Proceedings of the workshop on Evaluation for Language and Dialogue Systems-Volume 9, 2001.

[74] Sanjay K Dwivedia, Vaishali Singhb, "Research and reviews in question answering system", International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), 2013.

[75] Zhang, Y., Suda, N., Lai, L., Chandra, V., "Hello edge: Keyword spotting on microcontrollers." arXiv:1711.07128, 2017.

[76] C. Chelba, M. Norouzi, S. Bengio, "N-gram language modeling using recurrent neural network estimation", arXiv:1703.10724, 2017.

[77] N. Mostafazadeh, C. Brockett, B. Dolan, M. Galley, J. Gao,G. P. Spithourakis, and L. Vanderwende., “Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation.”, arXiv:1701.08251, 2017.

[78] H. Mei, M. Bansal, and M. R. Walter, “Listen, attend, and walk: Neural mapping of navigational instructions to actionsequences” AAAI Conference on Artificial Intelligence, 2016.

[79] A Quick Introduction to Neural Networks (https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/ aksesuar në 08.01.2018)

[80] CS224d: Deep Learning for Natural Language Processing (http://cs224d.stanford.edu/syllabus.html aksesuar në 08.01.2018).

[81] Memorizing is not learning (https://hackernoon.com/memorizing-is-not-learning-6-tricks-to-prevent-overfitting-in-machine-learning-820b091dc42 aksesuar në 08.01.2018)

[82] L., Kodra, E. K. Meçe, “Multimodal Attention for Visual Question Answering”, Computing Conference 2018, London, United Kingdom, July 10-12, 2018

[83] L., Kodra, E. K. Meçe, “Multimodal Attention in Recurrent Neural Networks for Visual Question Answering”, Global Journal of Computer Science and Technology (GJCST), Volume XVII Issue I, 2017

[84] L., Kodra, E. K. Meçe, “Visual Question Answering Agent with Visual and Textual Attention”, 7th International Conference on Frontiers of Information Technology, (ICFIT 2018), Paris, France, April 14-16, 2018

[85] L., Kodra, E. K. Meçe, “Multimodal Attention Agents in Visual Conversation”, 6th International Conference on Emerging Internet, Data & Web Technologies (EIDWT), Tirana, Albania, March 15-17, 2018

Shtojca 1

Fjalor i Termave Teknikë

Ky fjalor përmban termat teknikë në gjuhën angleze të përdorura në këtë disertacion së bashku me një përshkrim të shkurtër të tyre.

- **Activation function** – Funksion që përdoret nga një neuron për të transformuar input-in në output.
- **Activation volume** – Vëllimi i të dhënave në dalje të një shtrese konvolucionale brenda një rrjeti neural konvolucional. Termi vëllim nuk i referohet sasisë së të dhënave, por formës 3-dimensionale të tyre.
- **Average accuracy** – Metrikë për vlerësimin e modeleve neurale. Matet si raporti i numrit përgjigjeve të sakta mbi numrin total të përgjigjeve.
- **Background clutter** (fig 1.1) – Zhurmë vizuale që paraqitet në sfond dhe vështirëson dallimin e objekteve në plan të parë nga sfondi.
- **Backpropagation** – Algoritmi më i përdorur i trajnimit të rrjetave neurale artificiale.
- **Bag of words** – Mënyrë e paraqitjes së tekstit. Ky i fundit konsiderohet si një "çantë" me fjalë. Lidhjet gramatikore apo renditja e fjalëve nuk merren parasysh.
- **Batch** – Pjesë e të dhënave të një dataseti të cilat përdoren në një hap të ekzekutimi të algoritmit Stochastic Gradient Descent.
- **Batch size** – Përmasa e batch-it të të dhënave.
- **Bias** – Parametër shtesë që e lejon neuronin të ndryshojë output-in e tij për t'iu përshtatur më mirë kërkesave të trajnimit.
- **Chain rule** – Teknikë për llogaritjen e gradientit të rrjetit neural gjatë backpropagation.

- **Chatbot** – Algoritëm i ndërtuar me synimin për të zhvilluar një bisedë me operatorin njerëzor në gjuhë natyrore.
- **Classification** – Algoritëm i përdorur për të gjeneruar dhe shprehur output-in e një algoritmi të machine learning. Ky output shprehet si një renditje sipas probabiliteteve që input t'i përkasë një klase të caktuar.
- **Common sense (knowledge)** – Mendim praktik që është i zakonshëm për shumicën e njerëzve dhe që ka lidhje me çështje të përditshme ose aftësi bazë për të perceptuar, kuptuar dhe arsyetuar.
- **Computational graph** – Grafi llogaritës i një funksioni.
- **Computational linguistic** – Fushë që merret me modelimin e gjuhës natyrore nga pikëpamja llogaritëse.
- **Computer vision** – Fushë kërkimi e cila merret me zhvillimin e teknikave që i mundësojnë kompjuterat të përpunojnë, analizojnë dhe kuptojnë imazhet dhe videot.
- **Conversational question answering** – Sistem pyetje-përgjigje ku kërkimi për informacion zhvillohet nëpërmjet pyetjeve të njëpasnjëshme. Pyetjet e bëra nga përdoruesi njerëzor mund të kenë lidhje dhe/ose t'i referohen pyetjeve dhe përgjigjeve të mëparshme. Sistemi duhet të kuptojë kontekstin dhe tematikën e informacionit që kërkohet dhe të zgjidhë në mënyrë të saktë referencat ndaj informacionit të pyetjeve dhe përgjigjeve të mëparshme.
- **Convolutional feature map** – Hartë e tipareve të imazhit e cila gjenerohet nga rrjeti CNN pas përpunimit të imazhit.
- **Convolutional layer** – Lloji më i rëndësishëm i shtresave të një rrjeti neural konvolucional në të cilën kryen shumica e llogaritjeve. Kjo shtresë merr një tensor hyrjeje dhe gjeneron një tensor daljeje duke kryer konvolucionin e hyrjes me një tensor parametrash. Qëllimi i trajnimit të rrjetit konvolucional është mësimi i këtyre parametrave.
- **Convolutional Neural Network (CNN)** – Lloj rrjetash neurale artificiale të cilat përdoren gjerësisht për përpunimin e imazheve.
- **Coreference** – Referenca gjatë dialogut për objekte të imazhit duke përdorur përemra.

- **Corpus (of words)** – Tërësia e dokumentave që përbëjnë një sistem mbi të cilat bëhet përpunim gjuhësor.
- **Cross-entropy** – Funksion për llogaritjen e humbjes (d.m.th. gabimin) së një algoritmi të machine learning.
- **Data features** – Tipare të të dhënave të cilat përdoren për të gjeneruar funksionin që përfaqëson këto të dhëna.
- **Data generating distribution** – Burimi i të dhënave. Shpërndarje nga e cila merren të dhëna për të krijuar dataset-e për trajnimin, validimin dhe testimin e algoritmave të machine learning.
- **Data-driven (approach)** – Qasje statistikore për të zgjidhur një problem në machine learning.
- **Dataset** – Bashkësi të dhënash të grumbulluara nga e njëjta shpërndarje gjeneruese të dhënash dhe që përdoret për trajnimin, validimin dhe testimin e algoritmave të machine learning.
- **Decision making systems** – Sisteme të cilat përdorin algoritma kompjuterikë për të analizuar të dhënat, identifikuar dhe zgjidhur problemet si dhe marrjen e vendimeve rreth këtyre problemeve.
- **Deep learning** – Procesi i trajnimit të rrjetave neurale artificiale që përmbajnë më shumë se 1 shtresë të fshehur.
- **Deep neural networks** – Rrjeta neurale artificiale të cilat përmbajnë më shumë se 1 shtresë të fshehur.
- **Dependencies** – Paketa software që duhen instaluar paraprakisht.
- **Dropout** – Teknikë që përdoret për të kontrolluar overfitting. Gjatë trajnimit, shkëputen në mënyrë të rastësishme disa nyje nga rrjeti neural sipas një probabiliteti të caktuar. Kjo bëhet për të penguar rrjetin të përshtatet më shumë seç duhet me dataset-in.
- **Early stopping** – Teknikë që ndërpret para kohe trajnimin e një algoritmi kur performanca e tij gjatë validimit është brenda limiteve të dëshiruara.
- **Element-wise (addition, multiplication)** – Veprimi matematikor i bërë mbi elementët korrespondues të dy matricave.

- **Embedding space** – Hapësira vektoriale për paraqitjen numerike në trajtë matricore të fjalëve ose imazhit.
- **End-to-end (training)** – Trajnim i pjesëve të ndryshme të një rrjeti neural si një njësi e vetme.
- **Exploding gradient** – Problemi i zmadhimit të pakontrolluar të gradientit të një rrjeti neural artificial gjatë backpropagation. Ky problem mund ta bëje gradientin matematikisht të paqëndrueshëm dhe të pengojë trajnimin e rrjetit. Zgjidhet duke aplikuar teknikën gradient clipping.
- **Factoid (question)** – Pyetje, përgjigja e së cilës mund të shprehet thjesht nëpërmjet një fakti.
- **Feedforward neural network** – Rrjet neural artificial, lidhjet e neuroneve të të cilit nuk formojnë cikle.
- **Fully-connected layer** – Shtresë në të cilën të gjitha neuronet janë të lidhura me të gjitha neuronet e shtresës paraardhëse.
- **Gated Recurrent Units (GRU)** – Variant arkitekture e rrjetave neurale rekurrente.
- **Generalization** – Aftësia e një modeli për të patur performancë të mirë për të dhëna të pa vrojtuara më parë.
- **Goal-free dialogue** – Dialog midis një makine dhe një njeriu që nuk ka si objektiv realizimin e një detyre të caktuar, por thjesht angazhimin e njeriut në dialog për një kohë sa më të gjatë
- **Goal-oriented dialogue** – Dialog midis një makine dhe një njeriu që ka për qëllim realizimin e një detyre të caktuar (p.sh. prenotimin e një dhome hoteli).
- **Gradient clipping** – Teknikë për zgjidhjen e problemit të exploding gradient duke bërë prerjen (d.m.th. zvogëlimin manual) sipas një algoritmi të caktuar të gradientit që kalon një prag të caktuar.
- **Hidden state** – Gjendja e brendshme e një rrjeti neural artificial rekurrent.
- **Human-computer interaction** – Ndërveprim midis njerëzve dhe kompjuterave.
- **Hyperparameter** – Parametra të jashtëm që përdoren për të kontrolluar trajnimin e algoritmit të machine learning. Këto parametra vendosen nga

projektuesi dhe algoritmi nuk ka asnjë kontroll mbi to. Nuk duhet të ngatërrohen me parametrat të cilat algoritmi i mëson për të gjeneruar output-in e tij. Një shembull hiperparametri është shkalla e të mësuarit.

- **Image captioning** – Procesi i gjenerimit të përshkrimit të një imazhi duke përdorur gjuhën natyrore.
- **Image features** – Tipare të ekstraktuara nga një imazh me anë një rrjeti neural konvolucional. Tipari mund të jetë p.sh. vizë, hark, etj.
- **Inference** – Një konkluzion i arritur mbi bazën e provave dhe arsyetimit.
- **Information retrieval** – Procesi i përftimit të informacionit relevant ndaj nevojës për informacion duke u nisur nga një bashkësi burimesh informacioni
- **Input volume** – Vëllimi i të dhënave në hyrje të një shtrese konvolucionale brenda një rrjeti neural konvolucional. Termi vëllim nuk i referohet sasisë së të dhënave, por formës 3-dimensionale të tyre.
- **Intra-class variation** (fig 1.1) – Ndryshimet vizuale që mund të kenë nga njëri-tjetri objekte të së njëjtës klasë.
- **Knowledge base** – Baza e njohurive.
- **Leaky ReLU** – Variant i funksionit ReLU i cili nxjerr output të ndryshëm nga zero kur input-i është negativ me qëllim që të zgjidhë problemin e vanishing gradient.
- **Learning rate** – Hiperparametër që përcakton shkallën e të mësuarit të një algoritmi të machine learning. Është një nga hiperparametrat më të rëndësishëm. Kontrollon masën e ndryshimit të parametrave të algoritmit dhe në mënyrë indirekte konvergjimin e algoritmit të trajnimit.
- **Log-likehood** – Probabilitet logaritmik. Maksimizohet gjatë trajnimit të rrjetit neural.
- **Machine learning** – Fushë e shkencave kompjuterike e cila i jep kompjuterave aftësinë të mësojnë pa qenë të programuar në mënyrë eksplicite.
- **Mapping rules** (**of knowledge base**) – Rregulla korrespondence midis njohurisë dhe paraqitjes së saj në një bazë njohurish.
- **Mean reciprocal rank (MRR)** – Metrikë e information retrieval përdorur për të vlerësuar modelin e agjentit të dialogut vizual.

- **Natural language processing** – Përpunimi i gjuhës natyrore.
- **Neural information processing** – Përpunimi i informacionit me anë të rrjetave neurale artificiale.
- **"No free lunch" Theorem** – Teoremë sipas së cilës asnjë algoritëm i *machine learning* nuk është universalisht më i mirë se të tjerët dhe për rrjedhojë nuk mund të jetë automatikisht dhe universalisht i aplikueshëm për të gjitha përdorimet e mundshme dhe për të gjitha të dhënat e mundshme.
- **Non-factoid (question)** – Pyetje, përgjigja e së cilës nuk mund të shprehet thjesht nëpërmjet një fakti, por kërkon një formë më deskriptive.
- **One-hot vector** – Një vektor që të gjithë elementët i ka të barabartë me 0 përveç një elementi i cili është i barabartë me 1.
- **Ontology** – Emërtimi formal dhe përcaktimi i llojeve, karakteristikave dhe lidhjeve midis entiteteve që ekzistojnë në një domain të veçantë.
- **Open-ended (answer)** – Përgjigje e lirë e cila nuk kufizohet në disa alternativa.
- **Overfitting** – Situatë ku ndryshimi midis gabimit të testimit dhe gabimit të trajnimit është i madh. Kjo do të thotë se algoritmi i machine learning është përshtatur së tepërmi me dataset-in e trajnimit dhe nuk gjeneralizon mirë për të dhënat e dataset-it të testimit.
- **Padding** – Teknikë për kontrollimin e vëllimit të të dhënave duke shtuar rreth tyre një shtresë me vlera zero me trashësi të caktuar .
- **Pattern matching** – Përputhja e modelit ose përputhja gjuhësore.
- **Pattern-based** – I bazuar mbi një model.
- **Pooling (layer)** – Shtresë e rrjetit konvolucional që kryen reduktimin e vëllimit të të dhënave dhe parametrave të rrjetit sipas një kriteri të caktuar. Kjo shtresë nuk përdor parametra të cilat duhen mësuar gjatë trajnimit.
- **Pooling (operation)** – Veprimi i reduktimit të dimensionalitetit të një vëllimi të dhënash bazuar në një kriter të caktuar.
- **POS (Part Of Speech) tagging** – Procesi i kategorizimit të një fjale në një kategori të caktuar gramatikore – p.sh. emër, folje, etj.
- **Predicate** – Pjesë e RDF triplet.

- **Question answering systems** – Sistem inteligjent i cili plotëson nevojat për informacion të operatorit njerëzor nëpërmjet kthimit të përgjigjeve në gjuhën natyrore për pyetje të bëra nga ky i fundit po në gjuhë natyrore.
- **Recall@k** – Metrikë e information retrieval përdorur për të vlerësuar modelin e agjentit të dialogut vizual.
- **Recurrent Neural Networks (RNN)** – Arkitekturë ciklike rrjetash neurale artificiale që përdoret gjerësisht për përpunimin e sekuencave të informacionit që kanë vartësi nga njëra-tjetra.
- **Regex** – Shkurtim për "Regular Expression". Një sekuencë karakteresh që përcakton një model të caktuar kërkimi.
- **Regularization parameter** – Parametër që përdoret për të mbajtur nën kontroll overfitting duke penalizuar në mënyrë selektive disa parametra të algoritmit machine learning për të zvogëluar ndikimin e tyre në output-in e algoritmit.
- **Reinforcement learning** – Teknikë hibride midis supervised dhe unsupervised learning sipas së cilës algoritmi nuk e di paraprakisht çfarë output-i duhet të gjenerojë por atij i jepen "shpërblime" nëse output-i është i saktë. Qëllimi i algoritmit është të mësojë parametrat në mënyrë që të maksimizojë shpërblimet. Kjo teknikë është shumë e përdorur gjatë zhvillimit të algoritmave që mësojnë lojra Atari.
- **Retrieval metrics** – Metrika të Information Retrieval.
- **Scene understanding (image)** – Aftësia e algoritmave për të kuptuar imazhin dhe lidhjet që objektet apo pjesë të ndryshme të imazhit kanë me njëra-tjetrën
- **Semantic web** – Zgjerim i rrjetit aktual World Wide Web në të cilin informacionit i është dhënë kuptim i mirëpërcaktuar dhe që mundëson kompjuterat dhe njerëzit të bashkëpunojnë.
- **Softmax** – Funksion i përdorur për të shprehur output-in e një algoritmi klasifikimi në machine learning. Ky output shprehet në formën e një shpërndarjeje probabilitare të të gjithë klasave të mundshme.
- **Speech recognition** – Fushë kërkimi e cila merret me zhvillimin e teknikave që i mundësojnë kompjuterat të transformojnë gjuhën e folur në tekst.

- **State of the art** – Niveli më i lartë i zhvillimit të një teknike apo fushe shkencore në një moment të caktuar kohe.
- **Stride** – Hapi me të cilin filtri i parametrave konvoluohet me vëllimin e të dhënave në një rrjet neural konvolucional.
- **Supervised learning** – Teknikë trajnimi e machine learning sipas së cilës algoritmi e di paraprakisht output-in që duhet të gjenerojë dhe mëson parametrat të cilat do të gjeneronin këtë output.
- **Support Vector Machine** – Algoritëm i tipit supervised learning që analizon të dhëna për problemat e klasifikimit dhe regresit linear.
- **Tensor** – Termi me të cilin i referohen të dhënave në hyrje dhe në brendësi të një rrjeti neural artificial. Tensori është një matricë shumëdimensionale të dhënash që kalon nga njëra shtresë në tjetrën gjatë përpunimit të informacionit nga rrjeti neural.
- **Token (input)** – Një pjesë përbërëse e informacionit që përpunohet nga algoritmi. Në rastin e modeleve të propozuara në këtë disertacion, token-i i input-it është imazhi dhe fjalët përbërëse të pyetjes.
- **Tokenization** – Procesi i shpërbërjes së një sasie informacioni në pjesë individuale të quajtura ndryshe "token".
- **Translational invariance (of images)** – Veçori e imazheve sipas së cilës pikselat respektivë të imazhit mbeten të pandryshuar edhe nëse imazhi rrotullohet apo spostohet.
- **Triplets (RDF)** – Mënyrë e organizimit të informacionit formën e njësive treshe (triplets të përbëra nga subjekti, atributi (predicate) dhe objekti) në bazën e njohurive.
- **Underfitting** – Situatë ku gabimi i trajnimit është i madh dhe algoritmi nuk është në gjendje të përfaqësojë një pjesë të madhe të të dhënave në dataset.
- **Unsupervised learning** – Teknikë trajnimi e machine learning sipas së cilës algoritmi duhet të zbulojë lidhjet që kanë të dhënat me njëra-tjetrën dhe ti grupojë ato në kategori të paracaktuara. Algoritmi nuk i di paraprakisht kategoritë përkatëse të të dhënave.

- **Vanishing gradient** – Problemi i zhdukjes (d.m.th. zerimit) të gradientit të një rrjeti neural artificial gjatë backpropagation. Ky problem mund të pengojë ose ta bëjë të pamundur trajnimin e rrjetit.
- **Web intelligence** – Fushë e kërkimit që shqyrton të dhënat dhe përdor inteligjencën artificiale dhe teknologjitë e informacionit për të krijuar produkte dhe shërbime të reja të cilat do të operojnë në World Wide Web.
- **Word embeddings** – Mënyrë për paraqitjen numerike në trajtë matricore të fjalëve.

Shtojca 2

Lista e Botimeve

Konferenca:

1. **Lorena Kodra,** Elinda Kajo Meçe, “A review on current research in Question Answering Systems and future trends”, 11th Doctoral Student Conference, May 17-19, 2017, Thessaloniki, Greece
2. **Lorena Kodra,** Elinda Kajo Meçe, “Multimodal Attention for Visual Question Answering”, Computing Conference 2018, July 10-12, 2018, London, United Kingdom
3. **Lorena Kodra,** Elinda Kajo Meçe, “Multimodal Attention Agents in Visual Conversation”, 6th International Conference on Emerging Internet, Data & Web Technologies (EIDWT), March 15-17, 2018, Tirana, Albania
4. **Lorena Kodra,** Elinda Kajo Meçe, “Visual Question Answering Agent with Visual and Textual Attention”, 7th International Conference on Frontiers of Information Technology, (ICFIT 2018), April 14-16, 2018, Paris, France
   (Botuar në proceedings të: 2018 International Conference on e-business and mobile commerce (ICEMC 2018), May 21-23, Chengdu, China)

Revista:

1. **Lorena Kodra,** Elinda Kajo Meçe, “A Review on Neural Network Question Answering Systems”, International Journal of Artificial Intelligence and Applications (IJAIA), Vol.8, No.2, March 2017
2. **Lorena Kodra,** Elinda Kajo Meçe, “Question Answering Systems: A Review on Present Developments, Challenges and Trends”, International Journal of Advanced Computer Science and Applications (IJACSA), Volume 8 Issue 9, 2017
3. **Lorena Kodra,** Elinda Kajo Meçe, “Multimodal Attention in Recurrent Neural Networks for Visual Question Answering”, Global Journal of Computer Science and Technology (GJCST), Volume XVII Issue I, 2017