

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Gestų kalbos vienetų atpažinimas iš video srauto

Recognition of Sign language units from a video stream

Bakalauro darbas

Atliko:	Pranciškus Ambrazas	(parašas)
Darbo vadovas:	j. asist. Linas Petkevičius	(parašas)
Darbo recenzentas:	dr. Vytautas Valaitis	(parašas)

Vilnius – 2018

Santrauka

Šiame darbe nagrinėjamas lietuvių gestų kalbos atpažinimas iš video srauto. Buvo susikurta lietuvių gestų kalbos duomenų bazė, kurią sudaro 22 gestai po 20 vaizdo įrašų kiekvienai klasei ir 3 gestai po 50 vaizdo įrašų kiekvienai klasei. Klasių, kurios turėjo po 20 vaizdo įrašų, video srautai buvo dauginami, kad kiekviena klasė turėtų po 50 vaizdinių įrašų. Remiantis šiais duomenimis buvo apmokomas konvoliucinis neuroninis tinklas pasinaudojant *Inception v3* modeliu. Toliau, remiantis konvoliucinio neuroninio tinklo rezultatais, buvo apmokomas rekurentinis neuroninis tinklas. Galiausiai buvo sukurti trys skirtingi modeliai, gebantys atitinkamai atpažinti 3, 10 ir 25 skirtingas lietuvių gestų kalbos klases. Geriausius rezultatus davė 10 klasių modelis (87,34%), tačiau didesnę panaudojamumą turi 25 lietuvių gestų kalbos klasių modelis, kuris 79,18% tikslumu atspėja rodomo gesto klasę. Taigi galima daryti išvadą, kad gauti rezultatai duoda pakankamą tikslumą norint šį modelį pritaikyti realiame pasaulyje. Apmokius jį atpažinti daugiau klasių, panaudojamumas taptų žymiai didesnis.

Raktiniai žodžiai: konvoliuciniai neuroniniai tinklai, rekurentiniai neuroniniai tinklai, lietuvių gestų kalba, apsimokančios sistemos

Summary

In this study the recognition of Sign language units from a video stream is being analyzed. A dataset of 22 sign units with 20 videos for each class and 3 units with 50 videos for each class was created. Classes which had 20 videos was augmented to make 50 videos for each one of them. Based on this data, the convolutional neural network using Inception v3 model was trained. Further, based on convolutional neural network results, recurrent neural network was trained. Finally, three different models were created which are able to detect 3, 10 and 25 Lithuanian Sign language classes. The best results were presented by 10 classes model (87,34%), but 25 signs model has a higher usability within 79,18% accuracy. To sum up, these results give a possibility to adapt Lithuanian Sign language models in a real world. The usability of this model can be increased by further studies and trainings.

Keywords: convolutional neural networks, recurrent neural networks, lithuanian sign language, machine learning

TURINYS

Išvadas	5
Gestų kalba	5
Gestų kalbos specifika	5
Darbo tikslas	6
Darbo uždaviniai	6
Darbo eiga	6
Panaudotos priemonės	7
1. APSIMOKANČIOS SISTEMOS	8
1.1. Prižiūrėjimas mokymas	8
1.2. Neprižiūrėjimas mokymas	9
1.3. Praktinis mokymas	10
2. NEURONINIAI TINKLAI	11
2.1. Perceptronas	11
2.2. Daugiasluoksnius perceptronas	12
2.3. Dirbtiniai neuroniniai tinklai	13
2.4. Konvoliuciniai neuroniniai tinklai	13
2.4.1. Konvoliucinis sluoksnis	13
2.4.2. Telkimo sluoksnis	14
2.4.3. Atsisakymo sluoksnis	15
2.5. Rekurentiniai neuroniniai tinklai	15
2.5.1. Rekurentinių neuroninių tinklų tipai	16
2.5.2. Rekurentinių neuroninių tinklų modeliai	17
2.5.2.1. LSTM	17
2.5.2.2. GRU	18
2.5.2.3. LSTM ir GRU skirtumai	19
2.6. Apjungiamieji tinklų modeliai	19
3. EKSPERIMENTINĖ DALIS	21
3.1. Panašūs darbai	21
3.2. Argentiniečių gestų kalbos atpažinimas	21
3.2.1. Sistemos savybės	21
3.2.2. Bandymai	21
3.3. Lietuvių gestų kalbos atpažinimas	22
3.3.1. Duomenų paruošimas	22
3.3.2. Modelio apmokymas	23
3.3.2.1. 3 klasių apmokymas	24
3.3.2.2. 10 klasių apmokymas	25
3.3.2.3. 25 klasių apmokymas	27
3.3.2.4. Apibendrinimas	27
3.3.3. Modelio testavimas	28
REZULTATAI IR IŠVADOS	30
SANTRUMPOS	32

PRIEDAI	32
1 priedas. Rankų pirštų numeracija	33
2 priedas. Konvoliucinio tinklo modelis	34
3 priedas. 3 klasių apmokymas.....	35
4 priedas. 10 klasių apmokymas	36
5 priedas. 25 klasių apmokymas	38

Įvadas

Pasaulyje yra virš 7 milijardų žmonių, kurie kasdien tarpusavyje komunikuoja. Net 5% visos žmonijos populiacijos sudaro žmonės, turintys klausos problemų, dėl kurių jiems sunkiau komuniikuoti tarpusavyje. Vien 34 milijonai iš jų yra vaikai, iš kurių net 60% praradusių klausą vaikystėje galėjo būti girdintys dabar, jei būtų imtasi atitinkamų prevencinių priemonių. Paskaičiuota, kad iki 2050 metų žmonių, turinčių šias problemas, skaičius išaugs netgi iki 900 milijonų. Vien šiuo metu 1,1 milijardo jaunų žmonių nuo 11 iki 35 metų amžiaus yra ant klausos praradimo ribos dėl per didelio triukšmo [WhoInt].

Gestų kalba

Gestų kalba – tai geriausias būdas klausos negalią turintiems žmonėms bendrauti tarpusavyje. Ja pasaulyje bendrauja didžioji dalis klausos sutrikimus turinčiųjų, o amerikiečių gestų kalba (*angl. American Sign Language (toliau - ASL)*) yra trečia pagal populiarumą Amerikoje ne anglų kalba po ispanų ir kinų kalbų ir ketvirta apskirtai, kuria kalba virš 500 tūkstančių žmonių [Gall].

Kiekviena šalis turi savo valstybinę kalbą - lietuvių, anglų, ispanų, rusų ar kitą. Lygiai taip pat kiekviena šalis turi ir savo gestų kalbą. Yra tokios kalbos gestų kalbos kaip amerikiečių, lietuvių (*toliau - LGK*), argentiniečių ir kitos. Netgi tam tikri šalių regionai turi specifinius tos pačios kalbos dialektus, kaip, tarkime, vien Lietuvoje yra žodinės kalbos tarmės kaip aukštaičių, žemaičių, suvalkiečių ar dzūkų.

Kiekviena gestų kalba turi skirtingą gramatiką ir sintaksę. Skirtingos gestų kalbos skiriasi tiek abėcėlėmis tiek pačiais gestais, dėl to skiriasi ir gramatika. Taip yra dėl to, kad nėra bendrinės gestų kalbos - vien Amerikoje yra virš 35 skirtingų gestų kalbų.

Vienas gestas gali turėti kelias prasmes. Kaip ir lietuvių kalboje žodis „kasa“ turi tris skirtingas reikšmes, taip ir gestų kalboje vienas gestas gali turėti keletą reikšmių. Tačiau, gestas, parodytas truputėlį kitaip, gali turėti visiškai priešingą reikšmę. Tarkime, ASL gestai „geras“ ir „blogas“ skiriasi tik puse į kurią atsuktas delnas, tačiau daugiau neturi jokių skirtumų.

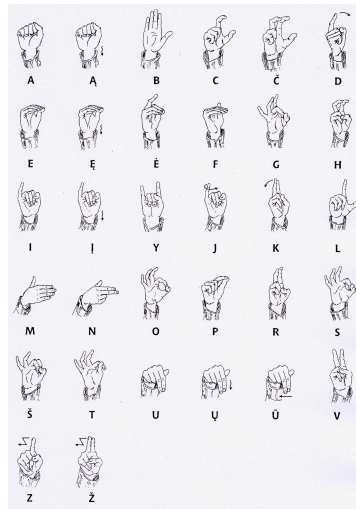
Gestų kalbos specifika

Kiekviena gestų kalba susideda iš **trijų** pagrindinių dalių:

1. **Statinė gestų kalba** - dar kitaip vadinama *pirštų kalba* (*angl. fingerspelling*). Tai įvairūs gestai rodomi vienos (ASL, LGK) ar net ir dviejų (britų ar vokiečių gestų kalba) rankų pagalba. Tai dažniausiai statiniai gestai, rodantys vieną raidę (*žr. 1 pav.*) ar net vieną žodį, kaip, pavyzdžiui, ASL yra „*I love you*“¹ gestas. Yra galimybė žodžius išreikšti ir abėcėliškai. Lygiai taip pat žmonės kasdieninėje kalboje turi galimybę pasakyti paraidžiui. Tačiau yra įprasta jungti raides į žodžius. O žodžius galiausiai į sakinius. Vienas iš variantų, kuomet naudojama gestų kalba paraidžiui tai vardų pasakyme. Tačiau svarbu paminėti tai, kad dažniausiai gestakalbiai prisistatydami parodo gestą, kuris priklauso tik jiems. Tai tarsi pa-

¹ liet. Aš tave myliu

rašas tam, kad nebereikėtų kreipiantis ar apibūdinant žmogų jo vardo sakyti paraidžiui. Toks gestas nebūtinai turi būti statinis.



1 pav. Amerikiečių gestų kalbos abėcėlė

2. **Dinaminė gestų kalba** - tai žodžių lygio gestų kalba. Nesunku pastebėti, kad 1 paveikslėlyje yra „A“, „D“, „I“, „J“, „K“, „U“, „Z“ ar „Ž“ raidės, kurios priskiriamos dinaminių judesių klasei. Taip pat yra žodžių, kurie priskiriami statinei gestų kalbai dėl savo kilmės, taip ir yra raidžių, kurios priskiriamos dinaminei gestų kalbai. Dinaminiais judesiais yra išreiškiami įvairūs gestų kalbos žodžiai tokie, kaip, pavyzdžiui, LGK yra „labas“, „mano“ ar „vardas“ ar kiti.
3. **Kitos ypatybės** - emocijos veide, liežuvis, burna ir kūno laikysena. Tai taip pat labai svarbios gestų kalbos ypatybės. Pavyzdžiui, klausiant gestų kalba klausimo, jei bus pakelti antakiai, tai reikš, kad laukiamas atsakymas „taip“ arba „ne“. Tačiau, jei antakiai bus suraukti, tai reikš, kad klausiama su paaiškinimu „kas“, „kur“, „kaip“ ar „ką“.

Darbo tikslas

Išanalizuoti ir praktiškai išbandyti gestų kalbos vienetų atpažinimo galimybes iš video srauto.

Darbo uždaviniai

- Išsiaiškinti, kokie yra paruošti duomenų rinkiniai, kuriais naudojantis būtų galima apmokyti neuroninius tinklus;
- Nustatyti, kokių tikslumu modeliai gali atpažinti gestus iš video srauto;
- Išanalizuoti konvoliucinius ir rekurentinius neuroninius tinklus bei jų panaudojimo būdus.

Darbo eiga

- Teorinės medžiagos analizė;
- Panašių ir jau įgyvendintų projektų paieška;

- Gestų kalbos video srautų paieška ir naujų vaizdo įrašų rengimas;
- Modelio apmokymai;
- Rezultatų palyginimai.

Panaudotos priemonės

- Python – programavimo kalba;
- TensorFlow – skirta darbui su apsimokančiomis sistemomis (*angl. machine learning*);
- Inception v3 – konvoliucinių neuroninių tinklų modelis;
- OpenCV – Python biblioteka darbui su vaizdais;

1. Apsimokančios sistemos

Apsimokančios sistemos (*angl. machine learning*) – metodų rinkinys, kuris sugeba automatiškai analizuoti duomenų struktūras ir tuomet apdoroti nematytus modelius, kad prognozuotų duomenis arba priimtų sprendimus kitais būdais [doi:10.1080/09332480.2014.914768].

1.1. Prižiūrimas mokymas

Prižiūrimas mokymas (*angl. supervised learning*) - tai apsimokančių sistemų apmokymo būdas, kuomet duomenys mokymui yra paruošiami taip, kad kiekvienas duomuo turėtų ir atitinkamą rezultatą. Kitaip tariant, jei yra duomuo *a*, tai yra ir jį atitinkantis rezultatas, arba dar vadinama etiketė (*angl. label*) *b*. Tai būdas, kuris veikia medžio principu.

1 lentelė. Pavyzdinis prižiūrimo mokymo apmokymui paruoštų duomenų rinkinys

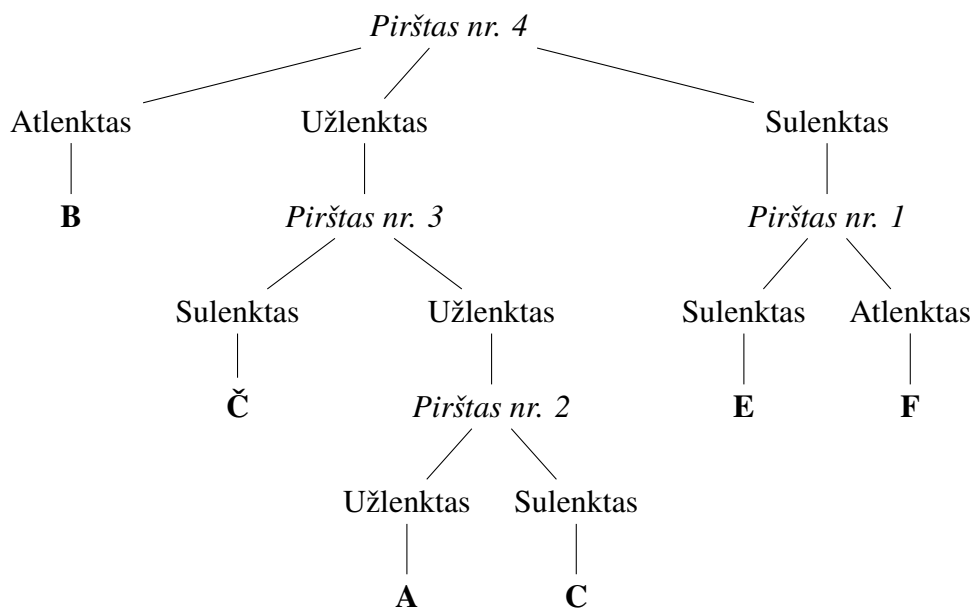
Nr.	Pirštas nr. 1	Pirštas nr. 2	Pirštas nr. 3	Pirštas nr. 4	Pirštas nr. 5	Raidė
1.	Atlenktas	Užlenktas	Užlenktas	Užlenktas	Užlenktas	A
2.	Atlenktas	Atlenktas	Atlenktas	Atlenktas	Atlenktas	B
3.	Atlenktas	Sulenktas	Užlenktas	Užlenktas	Užlenktas	C
4.	Atlenktas	Sulenktas	Sulenktas	Užlenktas	Užlenktas	Č
5.	Sulenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas	E
6.	Atlenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas	F

1 lentelėje pateikiamas pavyzdys su supaprastinta lietuvių gestų kalbos abėcėle. Lentelėje pateikiamos pirštų padėties, o pirštai numeruojami pagal 1 priede pateikiamą pirštų numeraciją. Kiekvieno piršto padėtis šiame pavyzdyje gali būti: *atlenktas*, *sulenktas* ar *užlenktas*. Ir kiekvienai padėčiai esant pateikiamas rezultatas, arba kitaip - etiketė, kokią raidę abėcėlėje atitinka pavaizduotos pirštų padėties.

2 lentelė. Pavyzdinė praktinė užduotis

Nr.	Pirštas nr. 1	Pirštas nr. 2	Pirštas nr. 3	Pirštas nr. 4	Pirštas nr. 5	Raidė
1.	Atlenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas	?

2 lentelėje pateikiamas uždavinys, kuriame nurodoma ta pati informacija, kuri buvo pateikta 1 lentelėje, tačiau rezultatas nėra pateiktas, o jis randamas medžio principu.



2 pav. Galimybių medis

Vien iš šio galimybių medžio galima matyti, kad pilnai užtenka sprendimui nustatyti 3 pirštų, kadangi rezultatų nėra daug. Jei būtų imama visa abėcėlės aibė, tuomet rezultato nustatymui būtų naudojama, galimai, visų pirštų padėtys.

1.2. Neprižiūrimas mokymas

Neprižiūrimas mokymas (*angl. unsupervised learning*) - mokymas, kuomet duomenims nėra priskiriamos teisingos etiketės ar teisingi rezultatai. Pavyzdžiui, tai galėtų atitikti naujos kalbos mokymąsi be mokytojo ar bet kokio žodyno. Kuomet pastoviai matomas vis tas pats tekstas, žodžiai tampa atpažįstami, tačiau išversti jų neįmanoma. Tačiau tai nesukelia jokių nepatogumų, jei į tekstą reikia įrašyti tinkamą žodį, kuomet dėl daugybės duomenų yra aišku koks žodis su kokia galūne turėtų būti įrašytas.

3 lentelė. Pavyzdinis neprižiūrimo mokymo apmokymui paruoštų duomenų rinkinys

Nr.	Pirštas nr. 1	Pirštas nr. 2	Pirštas nr. 3	Pirštas nr. 4	Pirštas nr. 5
1.	Atlenktas	Užlenktas	Užlenktas	Užlenktas	Užlenktas
2.	Atlenktas	Atlenktas	Atlenktas	Atlenktas	Atlenktas
3.	Atlenktas	Sulenktas	Užlenktas	Užlenktas	Užlenktas
4.	Atlenktas	Sulenktas	Sulenktas	Užlenktas	Užlenktas
5.	Sulenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas
6.	Atlenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas

3 lentelėje pateikiamas pavyzdinis neprižiūrimam mokymui apmokyti paruoštų duomenų rinkinys. Duomenys tokie patys, kaip ir 1 lentelėje, tačiau nėra teisingo atsakymo sulpelio „**Raidė**“. Apmokius tokią sistemą būtent tokiais duomenimis vienas iš tikėtinų scenarijų, kur galima būtų panaudoti tokią sistemą, tai nuspėti, kokios raidės yra labiausiai tikėtinos ar tiesiog numatyti, kokia labiausiai tikėtina raidžių seka bus rodoma.

1.3. Praktinis mokymas

Praktinis mokymas (*angl. reinforcement learning*) - labiausiai dirbtinį intelektą atitinkančių apsimokančių sistemų apmokymo modelis. Šis mokymas pagrįstas praktiniais bandymais. Kiekvienas teisingai gautas rezultatas yra būdas, kuriuo reikėtų sekti, ir kiekvienas blogai gautas rezultatas, yra būdas, kurio vertėtų atsisakyti. Dažniausiai šis apmokymo būdas naudojamas sistemą apmokant žaisti žaidimus. Vienas iš labiausiai žinomų būtent šiuo apmokymo būdu apmokytų modelių yra *AlphaZero*, kuris sugeba laimėti prieš pasaulio šachmatų čempionus. Tai puikus pavyzdys to, kaip kompiuteris iš laimėjimų, už kuriuos gauna taškus, ir pralaimėjimų, už kuriuos jam taškai atimami, sugeba rasti laimėjimo strategijas kiekviename žingsnyje ir taip, nuolatos tobulėdamas, laimėti dvikovas ar apskritai spręsti uždavinius, kuriuose reikalingas pastabumas ir strategijų kūrimas.

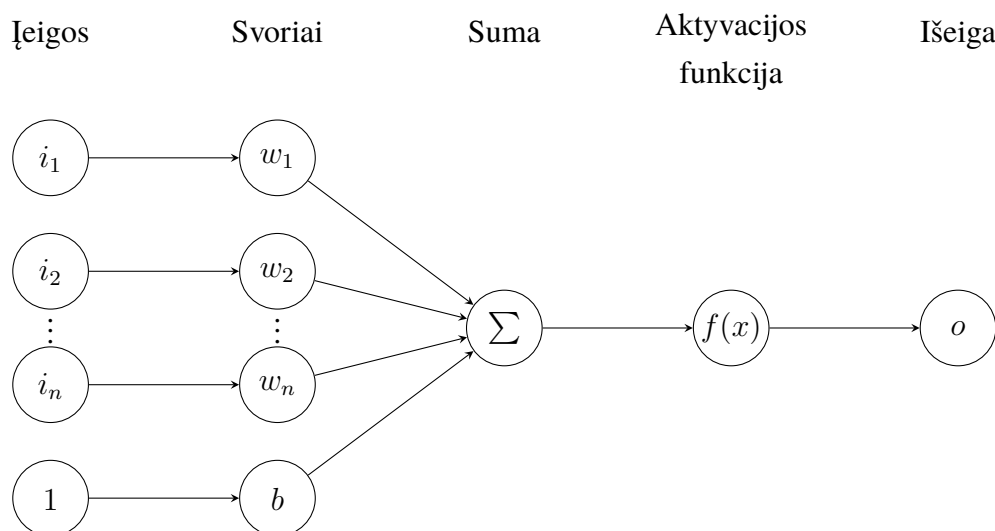
2. Neuroniniai tinklai

Žmogaus smegenys yra labai sudėtingas, nelineinis ir paralelinis kompiuteris [Hay09]. Kiekvieno žmogaus kūnas yra sudarytas iš milijardų nervinių ląstelių vadinamų neuronais. Jie sukuria ir/arba perduoda elektrocheminius impulsus. Neuronai tarpusavyje yra sujungti dendritais, ant kurių yra sinapsės.

Kiekvienas sužadintas neuronas dėl pasikeitusios temperatūros, spaudimo, skausmo ar kitų veiksnių, perduoda informaciją į smegenis dėl sprendimo, ką daryti, priėmimo. Neuronų paskirtis - siųsti signalą iš vieno neurono į kitą, kol galiausiai signalas pasiekia smegenis. Svarbu ir tai, kad kiekvienas neuronas yra nepriklausomas nuo kito. Tai tik grandis, kuri yra atsakinga už signalo priėmimą ir perdavimą. Smegenims gavus signalą, jį apdorojus ir priėmus sprendimą, signalas tuo pačiu keliu siunčiamas atgal, kol pirmąjį sužadimą gavęs neuronas sulaukia atsakymo.

2.1. Perceptronas

Perceptronas (*angl. perceptron*) – kompiuterinis modelis, skirtas atkartoti žmogaus kūne esančių neuronų darbą. Jis, kaip ir neuronas, gali gauti vieną ar kelias įėjimas ir generuoti vieną išėjimą. Neuronas yra skirtas tik perduoti impulsą, o perceptronas pritaiko aktyvacijos funkciją. Šis skirtumas tarp neurono ir perceptrono yra todėl, kad žmogus turi smegenis, o perceptronas atlieka smegenų vaidinį. Toliau pateikiamas perceptrono pavyzdys.



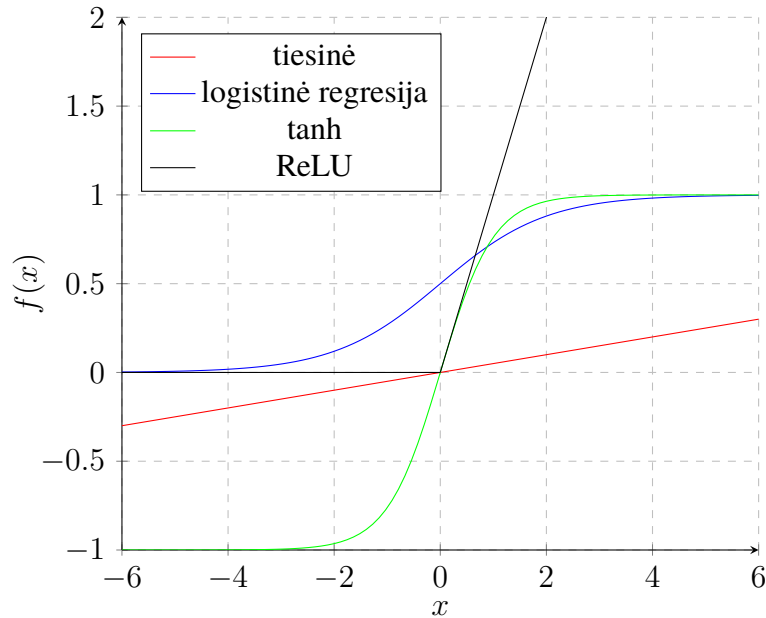
3 pav. Perceptrono pavyzdys

3 paveikslėlyje pavaizduotame pavyzdyje esančią išėjimą galima aprašyti formule:

$$o = f\left(\left(\sum_{j=0}^n i_j \cdot w_j\right) + 1 \cdot b\right) \quad (1)$$

1 formulėje i_j - j -toji įėjiga, w_j - j -tosios įėjigos svorinis koeficientas, b - poslinkio koeficientas, kuris dažniausiai kaip įėjigos vertę turi 1.

Kiekvienas perceptronas gali gauti vieną ar kelias įeigas (*angl. input*). Visų šių įeigų svorių suma yra sudedama ir paskui apdorojama aktyvacijos funkcija. Pritaikius aktyvacijos funkciją yra gaunama išeiga (*angl. output*). Yra keletas skirtingų aktyvacijos funkcijų. Pačios populiariausios pateikiamos 4 diagramoje.



4 pav. Aktyvacijos funkcijos

4 diagramoje pateikiamos šios funkcijos:

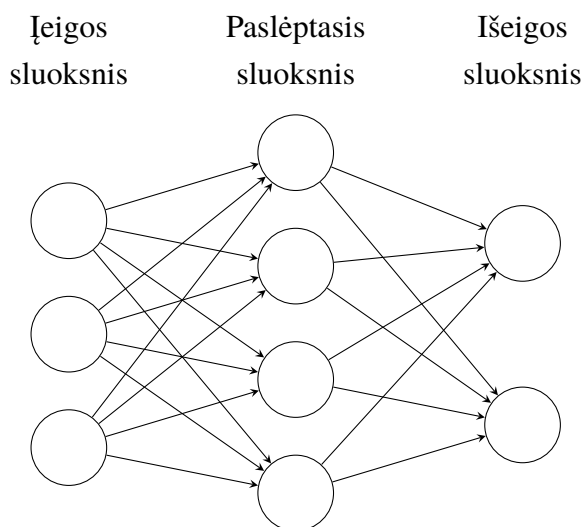
- Tiesinė – $f(x) = a \cdot x$;
- Logistinės regresijos – $f(x) = \frac{1}{1+e^{-x}}$;
- Tanh – $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$;
- ReLU – $f(x) = \begin{cases} 0 & , \text{kai } x < 0 \\ x & , \text{kai } x \geq 0 \end{cases}$.

2.2. Daugiasluosknis perceptronas

Daugiasluosknis perceptronas (*angl. multilayer perceptron*) – struktūra, sudaryta iš kelių sluoksnių perceptronų.

Dažniausiai daugiasluosknis perceptronas turi tris ar daugiau sluoksnius – įeigos (*input layer*), paslėptąjį (*hidden layer*) ir išeigos (*output layer*). Svarbu pastebėti, kad paslėptajame sluoksnyje gali būti daugiau nei vienas sluoksnis. Daugiasluosknis perceptronas kaip aktyvacijos funkciją naudoja nelineines aktyvacijos funkcijas. Dažniausiai tai būna *tanh* ar loginės regresijos funkcijos. Kiekvienas sluoksnio elementas yra sujungtas su kito sluoksnio elementu, todėl tai sudaro pilnai apjungtą (*angl. fully connected*) tinklą. Yra pavyzdžių, kur daugiasluoskniai perceptronai naudojami atpažinti žodinę kalbą ar versti tekstus.

Toliau pateikiamas daugiasluoksnio perceptrono pavyzdys.



5 pav. Daugiasluoksnio perceptrono pavyzdys

2.3. Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai (angl. *artificial neural networks*) – struktūra, sukurta remiantis žmogaus nervinės sistemos darbu. Dirbtiniai neuroniniai tinklai gali būti išmokinti atlikti klasifikavimo, spėjimo, sprendimų priėmimo ir kitas užduotis.

Dirbtiniai neuroniniai tinklai remiasi daugiasluoksnio perceptrono principu ir susideda iš šių sluoksnių - įeigos, paslėptojo, kuris gali būti sudarytas iš kelių sluoksnių, ir išeigos.

2.4. Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks*) – specialios rūšies vienpusiai (angl. *feed-forward*) neuroniniai tinklai, kurie remiasi daugiasluoksnio perceptrono principu. Šie tinklai, kurie remiasi *ReLU* principu yra kelis kartus greitesni, nei tie, kurie remiasi kitais principais, pavyzdžiui, *tanh* [NIPS2012_4824].

Toliau aptariami keli pagrindiniai konvoliucinių neuroninių tinklų sluoksniai.

2.4.1. Konvoliucinis sluoksnis

Konvoliucinis sluoksnis (angl. *convolution layer*) – sluoksnis, skirtas išskirti savybes. Šio sluoksnio pritaikymą galima skaidyti į tokias operacijas:

1. **Įeiga**, susidedanti iš $W_1 \times H_1 \times D_1$, kur W_1 - plotis, H_1 - aukštis ir D_1 - gylis;
2. **Parametrai**, kurie susideda iš F , K , P ir S , kur:
 - F - filtro dydis (dažniausiai taikomas 3×3 filtras);
 - K - filtrų skaičius (dažniausiai naudojamas 2^n , kur n - natūralusis skaičius);
 - P - papildomas rėmelis matricai, sudarytas iš 0. Dažniausiai naudojama $M = \frac{F-1}{2}$, kur M yra iš kiekvienos matricos pusės pridedamų eilučių ar stulpelių skaičius, sudarytas iš 0, tam, kad matrica nepakeistų savo dydžio po šio sluoksnio pritaikymo;
 - S - žingsnis, per kiek paslenkamas filtras (dažniausiai naudojamas 1);

3. **Išeiga**, susidedanti iš $W_2 \times H_2 \times D_2$, kur $W_2 = \frac{W_1 - F + 2P}{S} + 1$ - plotis, $H_2 = \frac{H_1 - F + 2P}{S} + 1$ - aukštis ir $D_2 = K$ - gylis

4 lentelė. Pavyzdinės konvoliucinio sluoksnio užduoties ypatybės

Įeiga			Parametrai				Išeiga		
W_1	H_1	D_1	F	K	P	S	W_2	H_2	D_2
3	3	1	3×3	1	1	1	3	3	1

Toliau, 2 formulėje pateikiamas pavyzdys, kuriame naudojamos 4 lentelėje pateiktos pavyzdinės konvoliucinio sluoksnio užduoties ypatybės. Spalvos šioje formulėje žymi skirtingų matricių elementus, kur geltona - įeigos matricos elementų spalva, raudona - papildomo rėmelio P spalva, mėlyna - filtro matricos spalva, o žalia - išeigos matricos elemento spalva. 3 ir 4 formulėse pateikiami konkretūs pavyzdžiai, kuriais remiantis buvo gautos 2 formulės reikšmės.

$$\begin{bmatrix} 1 & 8 & 6 \\ 9 & 2 & 4 \\ 3 & 7 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 21 & 8 \\ 24 & 17 & 19 \\ 5 & 20 & 7 \end{bmatrix} \quad (2)$$

$$0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 8 \cdot 0 + 0 \cdot 1 + 9 \cdot 0 + 2 \cdot 1 = 3 \quad (3)$$

$$0 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 8 \cdot 1 + 6 \cdot 0 + 9 \cdot 1 + 2 \cdot 0 + 4 \cdot 1 = 21 \quad (4)$$

2.4.2. Telkimo sluoksnis

Telkimo sluoksnis (*angl. pooling layer*) – sluoksnis, skirtas sumažinti matricą, paliekant tik svarbiausias jos dalis. Dažniausiai naudojamos vidutinės (*angl. average pooling*) arba didžiausios (*angl. max pooling*) reikšmės operacijos.

Telkimo sluoksnio operacijas galima skaidyti į tokias dalis:

1. **Įeiga**, susidedanti iš $W_1 \times H_1 \times D_1$, kur W_1 - plotis, H_1 - aukštis ir D_1 - gylis
2. **Parametrai**, kurie susideda iš F ir S , kur F - filtro dydis (dažniausiai taikomas 2×2 filtras) ir S - žingsnis, per kiek paslenkamas filtras (dažniausiai naudojamas 2)
3. **Išeiga**, susidedanti iš $W_2 \times H_2 \times D_2$, kur $W_2 = \frac{W_1 - F}{S} + 1$ - plotis, $H_2 = \frac{H_1 - F}{S} + 1$ - aukštis ir $D_2 = D_1$ - gylis

5 lentelė. Pavyzdinės telkimo sluoksnio užduoties ypatybės

Įeiga			Parametrai		Išeiga		
W_1	H_1	D_1	F	S	W_2	H_2	D_2
4	4	1	2×2	2	3	3	1

Toliau, 5 formulėje pateikiamas pavyzdys, kuriame naudojamos 5 lentelėje pateiktos pavyzdinės telkimo sluoksnio užduoties ypatybės. Spalvos šioje formulėje žymi filtro su žingsniu prita-

kytas operacijas gauti išeigai. 6 ir 7 formulėse pateikiami konkretūs pavyzdžiai, kuriais remiantis buvo gautos 5 formulės reikšmės.

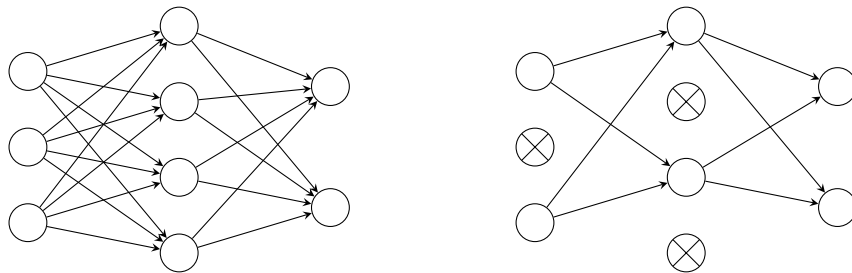
$$\begin{bmatrix} 1 & 3 & 1 & 3 \\ 2 & 5 & 4 & 2 \\ 4 & 3 & 2 & 5 \\ 2 & 5 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 5 & 5 \end{bmatrix} \quad (5)$$

$$\max\left(\begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}\right) = 5 \quad (6)$$

$$\max\left(\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}\right) = 4 \quad (7)$$

2.4.3. Atsisakymo sluoksnis

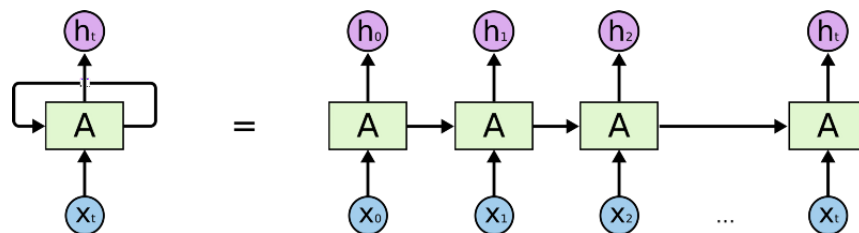
Atsisakymo sluoksnis (*angl. dropout layer*) – konvoliucinių tinklų sluoksnis, skirtas normalizuoti ir sureguliuoti tarpusavyje susijusių neuronų sąryšius, skirtus perduoti signalus. Mokymo fazėje dažniausiai ištrinamos neuronuose esančios reikšmės tam, kad šis per naują apsimokytų. Galimai netgi atsisakoma tam tikrų neuronų darbo [DBLP:journals/corr/abs-1207-0580].



6 pav. Standartinis neuroninis tinklas 7 pav. Tinklas po atsisakymo sluoksnio

2.5. Rekurentiniai neuroniniai tinklai

Rekurentiniai neuroniniai tinklai (*angl. recurrent neural networks*) – vienpusiai neuroniniai tinklai, kurie remiasi daugiasluoksnio perceptrono principu. Šie tinklai, apima kitų laiko vienetų apdorotą informaciją ir bendrą kitimą laike [DBLP:journals/corr/Lipton15].



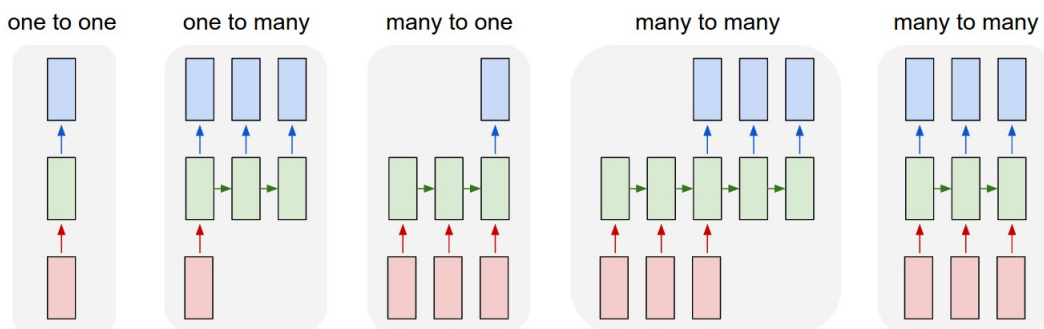
8 pav. Rekurentinių neuroninių tinklų veikimo principas

8 paveikselyje yra pavaizduotas bendrinis rekurentinių neuroninių tinklų veikimo principas, kurį galima užrašyti formule:

$$h_t = f_w(h_{t-1}, x_t) \quad (8)$$

Kur h_t - paslėpto sluoksnio būsena laiko momentu t , kurią dar būtų galima vadinti t žingsnio išeiga, f_w - aktyvacijos funkcija f su parametrais w , h_{t-1} - praėjusio žingsnio būsena, o x_t - įėjimo vektorius. Iš šios formulės galima pastebėti, kad kiekviena būsena gauna praeito žingsnio būseną, kuri yra reikalinga norint stebėti būsenas kintant laike.

2.5.1. Rekurentinių neuroninių tinklų tipai



9 pav. Rekurentinių neuroninių tinklų tipai

9 paveikslėlyje pavaizduoti keturi skirtingi būdai, kuriais naudojantis rekurentiniai neuroniniai tinklai veikia. Rausvos spalvos kvadratai reiškia įėją, žalsvas - paslėptuosius sluoksnius, o melsvas - išeigą. Pateikiami šie būdai:

- **Vienas su vienu** (*angl. one to one*) – būdas, kuriame yra viena įėjga, paslėptasis sluoksnis ir išeiga. Šis būdas dažniausiai taikomas konstruojant konvoliucinius neuroninius tinklus. Kaip pavyzdį galima pateikti paveikslėlio atpažinimą. Tai galėtų būti statinės gestų kalbos atpažinimas.
- **Vienas su daug** (*angl. one to many*) – būdas, kuriame yra viena įėjga ir kelios išeigos. Vienas iš panaudojimo būdų galėtų būti sakinio suformavimas iš paveikslėlio. Toks tinklas ne tik atpažįsta pagrindinį objektą kadre, bet ir apibūdina esančią aplinką, daro kitus sprendimus;
- **Daug su vienu** (*angl. many to one*) – būdas, kuriame yra daug įėjgų, bet tik viena išeiga. Tokio būdo pavyzdys galėtų būti vieno žodžio, tarkime, „labas“ atpažinimas iš video srauto.
- **Daug su daug** (*angl. many to many*) – būdas, kuriame yra daug įėjgų ir daug išeigų. Šis būdas gali būti skaidomas į dvi dalis:
 - **Priklausomas** - įėjgų skaičius sutampa su išeigų skaičiumi. Kiekviena įėjga turi savo išeigos atitikmenį. Tai būtų dalinai galima gretitinti su *vienas su vienu* būdu.
 - **Nepriklausomas** - įėjgų skaičius galimai nesutampa su išeigų skaičiumi. Kiekviena įėjga yra nepriklausoma ir išeigos dėliojamos pagal tam tikrus aspektus. Tokio būdo pavyzdys galėtų būti neuroniniai tinklai, kurie atlieka vertėjo funkcijas, pavyzdžiui, iš anglų į lietuvių kalbas, nes skiriasi tiek gramatika, tiek sakinių stilistika. Šis būdas taip pat galėtų būti pritaikomas versti gestų kalbą.

2.5.2. Rekurentinių neuroninių tinklų modeliai

Viena pagrindinių problemų, su kuria susiduria paprastieji rekurentiniai neuroniniai tinklai yra nykstančių gradientų problema (*angl. vanishing gradient problem*). Tai problema, kurios metu kiekvieno laiko momentu perceptronas apskaičiuoja naujas reikšmes iš praeitame žingsnyje turimų duomenų ir kaip įeiga priima praeito laiko momento išeią.

$$f(w_n \cdot h_n) = h_{n+1} \quad (9)$$

Šioje formulėje w_n - n -tojo sluoksnio svoris, h_n - n -tojo sluoksnio paslėptoji būseną, o $f(x)$ - aktyvacijos funkcija.

Tinklo pabaigoje gaunamas praradimas (*angl. loss*) arba kitaip - skirtumas tarp to, kas turėjo būti gauta ir ką tinklas gauna. Sakysime, kad $f(h_n)$ bus funkcija f , kurios parametras h_n yra paskutinio sluoksnio paslėptoji būseną.

Norint pakeisti w_n (n -tojo elemento svorį), tai galima padaryti apskaičiuojant gradientą atsižvelgiant į w_n .

$$\frac{\partial Loss}{\partial w_n} = \frac{\partial Loss}{\partial f(h_n)} \cdot \frac{\partial f(h_n)}{\partial w_n} = \frac{\partial Loss}{\partial f(h_n)} \cdot f'(h_n) \cdot w_n \quad (10)$$

Čia $\frac{\partial Loss}{\partial f(h_n)}$ yra dalinė išvestinė to, kaip skaičiuojamas praradimas. Svarbu tai, kad jis skaičiuojamas iš $f(h_n)$, todėl tai bus pastovi grįžtamojo ryšio (*angl. backpropagation*) lygtis.

Tęsiant toliau, pirmojo svorio w_1 reikšmę galima apskaičiuoti pagal šią lygtį:

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial f(h_n)} \cdot \frac{\partial f(h_n)}{\partial h_{n-1}} \dots \frac{\partial f(h_2)}{\partial h_1} \cdot \frac{\partial f(h_1)}{\partial w_1} = \frac{\partial f(Loss)}{\partial f(h_n)} \cdot f'(h_n) \cdot w_n \dots f'(h_1) * w_1 \quad (11)$$

Dėl šios priežasties ilgainiui dėl per naują skaičiuojamų svorių, perceptronas susiduria su problema, kad „pamiršta“, kas buvo prieš daugiau nei vieną laiko momentą - w_1 palaipsniui pradeda nebekisti dėl ilgų skaičiavimų ir tampa labai mažas. Tai reiškia, kad rekurentiniai neuroniniai tinklai paprasčiausiai vadovaujasi trumpalaikės atminties principu. Dėl šios priežasties buvo sukurtos keletas architektūrų, kurios sugebėtų atsiminti ir teisingai įvertinti esamą situaciją. Toliau pateikiama dažniausiai šiuo metu naudojamos RNN architektūros spręsti šią problemą.

2.5.2.1. LSTM

1997 metais Hochreiter ir Schmidhuber pristatė LSTM modelį, kuris, buvo manyta, galės išspręsti nykstančiųjų gradientų problemą. Šis modelis ypatingas tuo, jog kiekvienas įprastas paslėptąjo sluoksnio mazgas (*angl. node*) yra pakeistas atminties ląstele [DBLP:journals/corr/Lipton15].

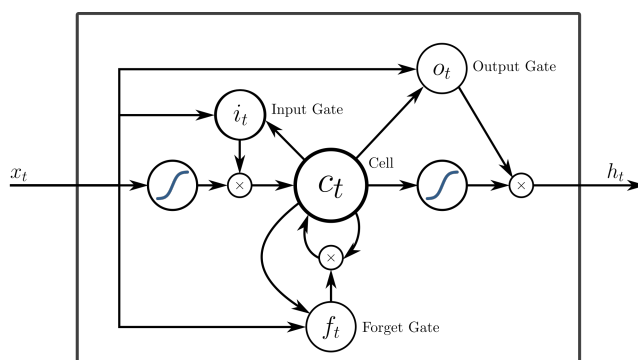
LSTM – ilga trumpalaikė atmintis (*angl. long short-term memory*) - RNN architektūra, kuri sugeba atsiminti informaciją ilgesniam laiko tarpui.

Paprasti RNN priima buvusią paslėptąją būseną, pritaiko aktyvacijos funkciją ir grąžina naują būseną. LSTM daro beveik tą patį, tik priima dar ir savo buvusią būseną ir grąžina savo naują

būseną.

LSTM įveda naują sąvoką - vartai (*angl. gate*). LSTM turi trijų skirtingų tipų vartus:

- **Užmaršties vartai** (*angl. forget gate*) - juose apdorojama praeita paslėptoji būseną ir dabartinė įeiga. Šių vartų išeiga - nuosprendis, ką vertėtų pasilikti ląstelės būsenoje, o ką - užmiršti. Kuo vertė artimesnė 1 - tuo tai labiau verta atsiminti, o arčiau 0 - pamiršti;
- **Įeigos vartai** (*angl. input gate*) - įeigos funkcija atnaujiną ląstelės būseną;
- **Išeigos vartai** (*angl. output gate*) nusprendžia, kurios ląstelės būsenos reikšmės bus pridėdamos į paslėptąją būseną, kuri bus visos ląstelės išeiga. Taip pat labai svarbu paminėti ir faktą, kad pasiliekomos ir tos reikšmės ar būsenos, kurios manoma, kad bus reikalingos ateityje.



10 pav. Atminties mazgo pavyzdys

Toliau pateikiamos *Tensorflow* LSTM variacijos tinklo gavimui:

- **Paprastas tinklas**
 - Du sluoksniai
 - Pirmajame - atsisakymo lygis 0,8
 - 128 - sluoksnių vienetų skaičius
- **Gilus tinklas**
 - Trys sluoksniai
 - Atsisakymo lygis 0,2 visuose lygiuose
 - 64 - sluoksnių vienetų skaičius
- **Platus tinklas**
 - Vienas sluoksnis
 - Atsisakymo lygis 0,2
 - 256 - sluoksnių vienetų skaičius
- **Platesnis tinklas**
 - Vienas sluoksnis
 - Atsisakymo lygis 0,2
 - 512 - sluoksnių vienetų skaičius

2.5.2.2. GRU

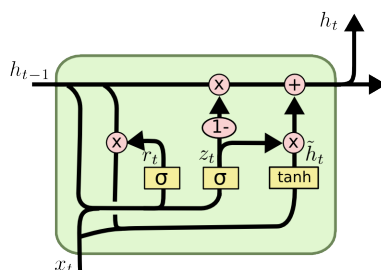
2014 metais Cho pristatė GRU modelį, kuris, buvo manyta, galės išspręsti nykstančiųjų gradientų problemą [DBLP:journals/corr/ChoMGBSB14]. Šis modelis yra panašus į LSTM ir abu

modeliai dažnai duoda labai panašius rezultatus.

GRU – uždaromas pasikartojantis vienetas (*angl. gated recurrent unit*) - RNN architektūra, kuri sugeba atsiminti informaciją ilgesniam laiko tarpui.

Kaip ir LSTM taip ir GRU naudojami vartai. Šiuo atveju yra dviejų tipų vartai:

- **Atnaujinimo vartai** (*angl. update gate*) - juose apdorojama nauja įeiga sudauginta su savo svoriu ir praėjusio laiko momento paslėptoji būsena su savo svoriu. Šie vartai nusprendžia, kiek aktuali praeituose laiko momentuose turima informacija ateityje;
- **Atstatymo vartai** (*angl. reset gate*) - šie vartai savo veiksmu niekuo nesiskiria nuo atnaujinimo vartų, tačiau jie skirti tam, kad nustatytų, kurių kitų laiko momentų informacijos verta atsisakyti.



11 pav. GRU mazgo pavyzdys

2.5.2.3. LSTM ir GRU skirtumai

LSTM ir GRU skiriasi tik skirtingais skaičiavimais neuronuose. Skiriasi vartų skaičius ir juose atliekami skaičiavimai. Tačiau rezultatas tarp LSTM ir GRU yra minimalus. Dažnai šie du modeliai grąžina arba labai panašius arba netgi identiškus rezultatus ir abu puikiai sprendžia nykstamųjų gradientų problemą. Abu apdoroja praeituose laiko momentuose sukauptą informaciją, sprendžia, ką reikėtų pasilikti, ko reikėtų atsisakyti, priimami sprendimai remiantis prieš daugiau nei vieną laiko surinkta informacija.

2.6. Apjungiamieji tinklų modeliai

Apjungiamieji tinklų modeliai - modeliai, kuriuose yra apjungiami konvoliuciniai ir rekurentiniai neuroniniai tinklai. Pagal RNN specifikacijas to daryti neturėtų būti prasmės, tačiau pagal dabartines KNN ir RNN galimybes, KNN kur kas geriau atpažįsta tam tikras pasikartojančias savybes, o RNN - jų kitimą laike. Todėl tokie apjungiamieji tinklų modeliai dažniausiai naudojami atpažįstant šnekamąją kalbą. Taip pat tokie modeliai puikiausiai tinka apjungiant įeigos sekas ir išvedant statines išeigas.

Yra du tipai apjungiamųjų tinklų modelių:

- Viena įeiga - daug išeigų. Tokiu būdu iš vieno kadro RNN sugeba aprašyti kadrą pateikiant ne vieną tame kadre matomą objektą. Pavyzdžiui, pateikiant jūros su laivu vaizdą galima gauti aprašymą, kad matomas laivas, kuris plaukia jūra.
- Daug įeigų - viena išeiga. Tokiu būdu iš kadrų sekos RNN sugeba generuoti vieną išeigą. Kitaip tariant duodant, pavyzdžiui, apdorojant video srautą, bus gaunama konkrečios klasės

išeiga.

Galima pastebėti, kad būtent šiuo atveju, modelis, kuris apdoro video srautą ir nuspręs, kurios klasės išeiga bus gaunama yra labai naudingas. Yra žinoma, kad netgi gerai apmokius sistemą KNN ji iš kadro gali nuspręsti koks veiksmas atliekamas ar kokiai klasei yra priskiriamas kadras. Šiuo atveju norint apmokyti sistemą atpažinti gestų kalbą galima pasinaudoti KNN skirstyti gestus pakadriui ir tuomet juos apjungus RNN modeliu galima išvesti vieną bendrą klasę, kuri ir bus bendra viso vaizdo srauto klasė.

3. Eksperimentinė dalis

Šioje dalyje bus aprašomi visi atlikti eksperimentai ir juose gauti rezultatai.

3.1. Panašūs darbai

Dar prieš metus, sistemų, kurios atpažintų gestų kalbą konvoliucinių ar rekurentinių tinklų pagalba, beveik nebuvo. 2017 metais Harish Chandra Thuwal ir Adhyan Srivastava iš Jamia Millia Islamia universiteto Naujajame Delyje sukūrė konvoliucinių ir rekurentinių neuroninių tinklų modeliais paremtą sistemą, kuri sugeba atpažinti gestų kalbą iš video srauto. Šiame darbe jie vaizdo įrašą verčia į kadrų seką ir apmoko konvoliucinį tinklą. Vėliau iš šių duomenų apmoko rekurentinį neuroninį tinklą. Svarbu paminėti, kad šie du studentai pasinaudojo argentiniečių gestų kalbos duomenų rinkiniu, kuriame ant kiekvienos rankos žmonės, kurie rodė gestus, buvo užsidėję skirtingų spalvų pirštines. Taip jie iš vaizdo įrašo kadrų ištrindavo visą foną ir palikdavo tik rankas, taip apmokydami sistemą be papildomų trikdžių (*angl. noise*).

3.2. Argentiniečių gestų kalbos atpažinimas

Buvo pasirinkta apmokyti jau esamą Harish Chandra Thuwal ir Adhyan Srivastava sukurta modelį, jį tobulinant.

3.2.1. Sistemos savybės

- KNN - Inception v3 modelis
- RNN - LSTM architektūra

3.2.2. Bandymai

6 lentelė. Argentiniečių gestų kalbos bandymai su 3 klasėmis

Bandymo Nr.	Klasių skaičius	Apmokymo tikslumas	Epochų skaičius	Tikslumas	Praradimas	Testavimas
1.	3	100%	10	81.27%	0.6431	85.32%
2.	3	99.99%	100	89.27%	0.4422	93.33%

Pats pirmasis bandymas buvo atliktas su trimis klasėmis, apmokant sistemą ir skaidant vaizdo įrašo kadrus kaip paveikslėlius. Kiekvienam iš jų buvo nuimamas fonas (*angl. background*) ir paliekamos tik rankų plaštakos, todėl buvo toks didelis apmokymo tikslumas.

Antruoju bandymu buvo atsisakyta nuimti foną ir palikti kadrus tokius, kokie yra. Dėl padidinto epochų skaičiaus rezultatai gauti geresni nei pirmuoju bandymu.

7 lentelė. Argentiniečių gestų kalbos bandymai su 25 klasėmis

Bandymo Nr.	Klasių skaičius	Apmokymo tikslumas	Epochų skaičius	RNN apmokymo tipas	Tikslumas	Praradimas
1.	25	91.90%	100	Platus	91.99%	0.6839
2.	25	91.90%	100	Platesnis	91.95%	0.6255
3.	25	91.90%	100	Gilus	16.55%	2.0566
4.	25	91.90%	10	Paprastas	97.61%	0.2814
5.	25	91.90%	100	Paprastas	92.66%	0.5539

7 lentelėje pateikiami dar 5 bandymai atlikti su argentiniečių gestų kalba. Šiuo atveju sistemos KNN modelis buvo apmokytas vieną kartą, o toliau buvo keičiami RNN apmokymo būdai. Šie būdai buvo paremti LSTM modeliu. Galima pastebėti, kad giliuoju (*angl. deep*) būdu rezultatai buvo prasčiausi.

3.3. Lietuvių gestų kalbos atpažinimas

Nėra jokių oficialių duomenų lietuvių gestų kalbos atpažinimui naudojantis konvoliuciniais ar rekurentiniais neuroniniais tinklais. Šie bandymai, manoma, kad yra pirmieji naudojantis rekurentiniais neuroniniais tinklais atpažinti LGK. Konvoliucinių neuroninių tinklų pagalba tokį darbą dariau prieš metus, skirtą atpažinti statinę gestų kalbos abėcėlę pasinaudojant Inception v3 modeliu.

Šių eksperimentų tikslas - išbandyti galimybes atpažinti gestų kalbą, nes lietuvių gestų kalbos atpažinimas iš video srauto galėtų palengvinti ne tik gestų kalbos nesuprantantiems bendrauti su gestakalbiais, bet ir, pavyzdžiui, turėti galimybę versti žodinę kalbą į gestų kalbą.

Toliau pateikiamas visas darbas padarytas su lietuvių gestų kalbos atpažinimu, duomenų rinkimu ir gautais rezultatais.

3.3.1. Duomenų paruošimas

Oficialiame lietuvių gestų kalbos žodyne, kurį pristato **neįgaliųjų reikalų departamentas prie socialinės apsaugos ir darbo ministerijos**, pateikiama apie 9000 gestų. Žodynas rengiamas nuo 2004 metų kurčiųjų ir girdinčiųjų komandos. Šiame žodyne gestus galima rasti pagal žodį, gesto formą ar temą. Taip pat galima pasirinkti ar gesto ieškoti kaip atitinkamo žodžio ar naudojimo pavyzdžiuose. Susiradus tinkamą žodį yra aprašomos tokios specifikos kaip plaštakos forma, lūpų judesys, žodžio ar sakinio reikšmė.

Naudojantis šiuo žodynu iškyla viena pagrindinė problema - kiekvienas gestas turi tik po vieną video įrašą atitinkantį tą žodį. Toks kiekis duomenų yra per mažas, norint apmokyti sistemą RNN būdu. Galima iš sakinių, kuriuose yra žodžio naudojimo pavyzdžiai, taip pat išskirti gestus, atitinkančius norimą gestą. Tačiau tai padidintų kiekvienos klasės duomenų kiekį iki daugiausiai 5 vaizdo įrašų. Net ir toks duomenų kiekis yra per mažas.

Nuspręsta duomenis susikurti. Teko pramokti lietuvių gestų kalbos gestus. Įsigilinti į gestų kalbos specifiką. Pirmiesiems bandymams buvo nufilmuota 3 skirtingų žodžių klasių gestai po 50

vaizdo įrašų kiekvienam iš jų, kas yra tapatu 150 video įrašų. Filmuota buvo mobiliuoju telefonu rodančiajam atsistojus prie gelsvos sienos. Filmuoti buvo du skirtingi asmenys, kurių kiekvienas atliko po 25 vaizdo įrašus kiekvienai klasei. Buvo pasirinkta pirmiesiems bandymams pasinaudoti „labas“, „mano“, „vardas“ žodžių klasėmis. Kiekvienas vaizdo įrašas truko ne ilgiau nei 3 sekundes.

Vėliau buvo nufilmuotos dar 10 skirtingų gestų klasių. Šiuo atveju buvo nufilmuota nebe 50, o 20 kiekvieno gesto įrašų. Kaip ir pirmuoju filmavimu, taip ir šiuo, buvo filmuojami du skirtingi asmenys. Galiausiai buvo nufilmuota dar 12 papildomų klasių, todėl bendras duomenų rinkinys buvo sudarytas iš 25 skirtingų lietuvių gestų kalbos gestų klasių.

Rezultatas – duomenų bazė sudaryta iš 25 skirtingų LGK klasių po 20-50 video įrašų. Kiekvieno video trukmė ne ilgesnė nei 4 sekundės, kas tapatu ne daugiau nei 120 kadrų² kiekvienam įrašui. 8 lentelėje pateikiamos visos nufilmuotos klasės ir vaizdo įrašų kiekis paruoštas kiekvienam gestui.

8 lentelė. LGK paruošti duomenys

Nr.	Klasė	Kiekis	Nr.	Klasė	Kiekis	Nr.	Klasė	Kiekis
1.	labas	50	10.	radijas	20	19.	juodas	20
2.	mano	50	11.	šuo	20	20.	lėktuvas	20
3.	vardas	50	12.	katė	20	21.	autobusas	20
4.	jėga	20	13.	višta	20	22.	limonadas	20
5.	nuoširdus	20	14.	laikrodis	20	23.	kava	20
6.	nuostabus	20	15.	kapinės	20	24.	pietūs	20
7.	mama	20	16.	laimėti	20	25.	žaizda	20
8.	tėtis	20	17.	vestuvės	20			
9.	kompiuteris	20	18.	gydytojas	20			

3.3.2. Modelio apmokymas

Modelį buvo nuspręsta apmokyti pasinaudojant Harish Chandra Thuwal ir Adhyan Srivastava jau sukurtu modeliu, jau atliktus jam patobulinimus išbandant argentiniečių gestų kalbos atpažinimą. Vienas iš svarbiausių pakeitimų - nenutrinti fono kadruose. Taip pat pakeistas epochų skaičius, RNN LSTM tinklo apmokymo tipas, perdarytas testavimo mechanizmas.

Visų pirma buvo apmokyta sistema su trimis klasėmis „labas“, „mano“ ir „vardas“, turinčiomis po 50 vaizdo įrašų (žr. 8 lentelė). Šie gestai visi atliekami dešiniąja ranka, todėl tai sistemai, buvo manyta, turėtų šiek tiek palengvinti darbą su duomenimis.

Pirmiausiai kiekvienas vaizdo įrašas buvo išskaidomas į kadrus. Dažniausiai gesto vaizdo įrašas truko apie 2 sekundes, bet ne ilgiau 3 sekundžių. Tai reiškia, kad filmuojant mobiliuoju įrenginiu pasirinkus 30 kadrų per sekundę būdą, kiekvienas gestas turėdavo apie 60 kadrų, daugiausiai 90 kadrų. Buvo nuspręsta padaryti vienodus kiekius kadrų kiekvienam gestui. Tokiu atveju kadrų kiekis buvo pakeltas iki 120 kadrų kiekvienam vaizdo įrašui, kas lygu 4 sekundėms vaizdo įrašo.

²1 sekundė = 30 kadrų

Kadangi visi įrašai skyrėsi savo ilgiu buvo nuspręsta, jei vaizdo įrašas per trumpas, paskutinį kadrą kartoti tiek kartų, kad visi įrašai turėtų vienodą kiekį kadrų.

Buvo priimti tokie sprendimai, remiantis jau turimu argentiniečių gestų kalbos patobulintu apjungtu neuroniniu tinklu:

1. Vaizdo įrašą skaidyti į 120 kadrų (vietoje 200 turimame mokymo mechanizme). Jei įrašas per trumpas - paskutinį kadrą kartoti n kartų, kol bus 120 kadrų;
2. Konvoliucinį neuroninį tinklą apmokyti Inception v3 modeliu (žr. 2 priede);
3. RNN apmokyti pasirinkus paprastą modelį, kuris naudoja LSTM architektūrą;
4. Mokymo - testavimo aibę skaidyti į 80% mokymui ir 20% testavimui, todėl mokymui buvo skirta 40 vaizdo įrašų, o testavimui - 10.

3.3.2.1. 3 klasių apmokymas

Duomenys:

- 3 klasės - „labas“, „mano“, „vardas“;
- 40 vaizdo įrašai kiekvienai klasei. Viso - 120;
- 10 vaizdo įrašų kiekvienos klasės testavimui.

Pirmasis bandymas

Rezultatai:

- KNN: galutinis tikslumas (*angl. final test accuracy*) - 98.9%;
- Buvo atlikti spėjimai su KNN, kuris teisingai atspėjo 14307 iš 14400 kadrų (tikslumas - 99.35%);
- RNN: Tikslumas 86.93% ir 0.5081 praradimas;
- Mokymosi grafikai pateikiami 3 priede.

Antrasis bandymas

Pakeitimai:

Nuspręsta, jei vaizdo įrašas yra mažesnis nei 120 kadrų, kartoti kadrus nebe gale, kaip buvo ligi tol, bet juos pakartoti viduryje. Šio metodo išbandymui buvo pasinaudota pakartoti kadrus daugiausiai du kartus. Pavyzdžiui, jei įrašas yra 60 kadrų, tai kiekvieną kadrą pakartoti po du kartus. Jei įrašas 90 kadrų, tai kartoti 45-75 kadrus po du kartus. Tačiau buvo nuspręsta, kad jei vaizdo įrašas trumpesnis nei 60 kadrų, kartoti likusius kadrus 60 – n , kur n - kadrų skaičius, pasinaudojant paskutiniu kadru.

Rezultatai:

- KNN: galutinis tikslumas - 99.0%;
- Buvo atlikti spėjimai su KNN, kuris teisingai atspėjo 14270 iš 14400 kadrų (tikslumas - 99.10%);
- RNN: Tikslumas 87.5% ir 0.4995 praradimas;
- Mokymosi grafikai pateikiami 3 priede.

Trečiasis bandymas

Pakeitimai:

Nuspręsta patobulinti antruoju bandymu padarytą pakeitimą. Padaryta, kad jei vaizdo įrašas yra trumpesnis nei 60 kadrų, tada kartoti kadrus po 3 kartus. Jei ilgesnis nei 60 ir trumpesnis nei 120 - 2 kartus. Kadangi trumpesnių vaizdo įrašų nei 40 kadrų nebuvo, tai tikslo kartoti kadrus po 4 kartus taip pat nebuvo.

Rezultatai:

- KNN: galutinis tikslumas - 99.1%;
- Buvo atlikti spėjimai su KNN, kuris teisingai atspėjo 14307 iš 14400 kadrų (tikslumas - 99.35%);
- RNN: Tikslumas 88.33% ir 0.4832 praradimas;
- Mokymosi grafikai pateikiami 3 priede.

Apibendrinimas

Šiais trimis bandymais naudojantis trimis skirtingomis LGK klasėmis buvo pastebėta, kad geresni rezultatai gaunami tuomet, kai kartojamas gestas yra viduryje, kuomet ir yra svarbiausia jį pastebėti ir atpažinti. Toks sprendimas buvo priimtas dėl KNN apsimokymo galimybių ir bendrų savybių išskyrimo. Kartojimas tose vietose, kuriose labiausiai pasireiškia gesto savybės ir jo atskyrimas nuo kitų klasių, yra geresnis sprendimas apmokant KNN, nei kartojimas pačiame vaizdo įrašo gale, pasinaudojant paskutiniu kadru, nes tuomet rodantysis stovi nuleidęs rankas ar pozicijoje, kurioje didžiojoje klasių dalyje yra pradedami ir pabaigiami gestai.

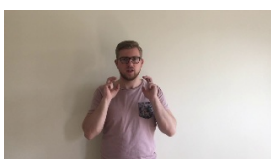
Pastebėta, kad rezultatai ypatingai nesiskyrė, bet gauti geriausi rezultatai, nors ir minimalūs, paskutiniu bandymu. Nuspręsta naudotis šiuo (trečiuoju bandymu) susikurtu metodu, vykdant tolimesnius apmokymus pasinaudojant daugiau klasių.

3.3.2.2. 10 klasių apmokymas

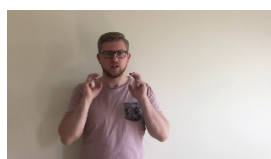
Antruoju apmokymu buvo pasirinktos 10 klasių (4-13 klasės (žr. 8 lentelę)). Kadangi buvo nufilmuota po 20 kiekvienos iš šių klasių vaizdo įrašų, tai buvo nuspręsta pasinaudojus *OpenCV* bibliotekos savybėmis, skaidant vaizdo įrašus į kadrus praplėsti vaizdo įrašų aibę iki 50.

Kiekvieną vaizdo įrašą skaidant į gestus, jei gestas per trumpas, gesto vaizdo įrašo centrinė dalis kartojama 2-3 kartus (*plačiau - 3.3.2.1. 3 klasių apmokymas. Trečiasis bandymas*).

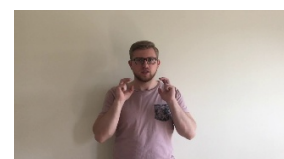
Prieš skaidant vaizdo įrašus į kadrus, iš eilės, jei vaizdo įrašų kiekis klasėje yra mažesnis nei 50, kiekvienas vaizdo įrašas yra pakartojamas antrąjį kartą. Jei mažesnis nei 25 - vaizdo įrašas pakartojamas trečiąjį kartą. Taip daroma tol, kol bendras skaičius vaizdo įrašų pasiekia 50. Tačiau kiekvienam vaizdo įrašo pakartojimui yra pritaikomos modifikacijos. Vaizdo įrašo kadrai yra ištempiami ir suspaudžiami pasinaudojus atsitiktiniais metodais.



12 pav. Originalus vaizdo įrašo kadras



13 pav. Modifikuotas 12 pav. kadras



14 pav. Modifikuotas 12 pav. kadras

Kiekvienas kadras vaizdo įrašė buvo 1280×720 taškų. Anksčiau kiekvienas kadras ir būdavo saugomas tokiu pačiu dydžiu. Nuspręsta sumažinti kadrus iki 320×180 . Tai padidino greitį ir labai sumažino užimamos vietos kiekį, kadangi taškų kiekis kadre buvo sumažintas 16 kartų.

Taip pat, šiame žingsnyje buvo pritaikytas automatinis LGK klasių vaizdo įrašų išskaidymas į mokymo-testavimo aibes. Generuojant naujus vaizdo įrašus paskutiniai 20% vaizdo klasės įrašų buvo perkeliama į testavimo aplanką (*angl. folder*).

Šio apmokymo metu visi bandymai buvo atlikti konvoliucinį neuroninį tinklą apmokius vieną kartą.

Buvo nuspręsta pasinaudoti gilioju RNN tinklu vietoje paprasto.

Duomenys:

- 10 klasių;
- 20 vaizdo įrašų kiekvienai klasei;
- Sugeneruota 40 vaizdo įrašų kiekvienai klasei;
- Sugeneruota 10 vaizdo įrašų kiekvienos klasės testavimui.

Pirmasis bandymas

Rezultatai:

- KNN: galutinis tikslumas - 96.3%;
- RNN: Tikslumas 93.58% ir 0.3727 praradimas;
- Mokymosi grafikai pateikiami 4 priede.

Antrasis bandymas

Pakeitimai:

Nuspręsta padidinti RNN epochų skaičių nuo 50 iki 100 ir pakeisti iš gilaus tiklo į paprastą tinklą.

Rezultatai:

- KNN: galutinis tikslumas - 96.3%;
- RNN: Tikslumas 45.61% ir 1.419 praradimas;
- Mokymosi grafikai pateikiami 4 priede.

Trečiasis bandymas

Pakeitimai:

Pastebėjus, kad gauti labai prasti rezultatai antruoju bandymu, buvo nuspręsta vėl pasinaudoti giliu tinklu, bet palikti 100 epochų skaičių.

Rezultatai:

- KNN: galutinis tikslumas - 96.3%;
- RNN: Tikslumas 94.28% ir 0.3563 praradimas;
- Mokymosi grafikai pateikiami 4 priede.

Apibendrinimas

Pastebėta, kad didinant epochų skaičių (pirmas ir trečias bandymai) galutinis RNN tikslumas kinta, bet nėra didelis. Šio apmokymo pabaigoje buvo nuspręsta, kad geriau palikti 50 epochų

skaičių vietoje 100, kadangi rezultatas beveik nekinta, o laiko užima 2 kartus daugiau. Taip pat pastebėta, kad 3 klasių apmokyje paprastas RNN LSTM tinklas buvo puikus sprendimas, duodantis po 10 epochų labai aukštus rezultatus, tačiau padidinus klasių skaičių nuo 3 iki 10, paprastas tinklas grąžina prastus rezultatus. To pasekoje buvo nuspręsta geriau naudoti gilųjį tinklą vietoj paprastojo tęsiant bandymus.

3.3.2.3. 25 klasių apmokymas

Trečiojo apmokymo metu buvo pasinaudota visomis duomenų paruošimo stadijoje pasiruštomis klasėmis (žr. 8 lentelę). Kiekviena klasė turėjo 20 (22 klasės) arba 50 (3 klasės) vaizdo įrašų. Naudojantis antruoju apmokymo metu susikurtu duomenų dauginimo mechanizmu, kuris leido 20 vaizdo įrašų duomenis paversti į 50. Taip pat automatiškai buvo išskaidoma mokymo-testavimo aibė. Reikėtų pastebėti ir tai, kad buvo pasinaudota ir 10 klasių apmokymo stadijoje sumažintu kadrų taškų skaičiumi. Tai vėlgi sutaupė labai daug užimamos vietos, bet neturėjo jokios įtakos tikslumui.

Duomenys:

- 25 klasės;
- 20-50 vaizdo įrašų kiekvienai klasei;
- Sugeneruota 40 vaizdo įrašų kiekvienai klasei;
- Sugeneruota 10 vaizdo įrašų kiekvienos klasės testavimui.

Rezultatai:

- KNN: galutinis tikslumas (*angl. final test accuracy*) - 89.1%;
- RNN: Tikslumas 92.89% ir 0.4361 praradimas;
- Mokymosi grafikai pateikiami 5 priede.

3.3.2.4. Apibendrinimas

Toliau, 9 lentelėje pateikiami visų trijų apmokymų su lietuvių gestų kalba geriausi gautų modelių rezultatai.

9 lentelė. Lietuvių gestų kalbos apmokymų rezultatai

Bandymo Nr.	Klasių skaičius	Vaizdo įrašų skaičius	KNN	RNN		
			Tikslumas	Epochų skaičius	Tikslumas	Praradimas
1.	3	120	99.1%	100	88.33%	0.4832
2.	10	400	96.3%	100	94.28%	0.3563
3.	25	1000	89.1%	50	92.89%	0.4361

Galima pastebėti, kad nėra didelio KNN ir RNN tikslumų ir praradimų skirtumo tarp modelių juos apmokius. Pastebėta, kad epochų skaičius apmokant RNN - 50 ar 100 - neturi didelės įtakos

bendro modelio tikslumo gerinimui. Taip pat verta pastebėti, kad klasių skaičius nedarė didelės įtakos KNN ir RNN tikslumams.

Matomas tikslumo mažėjimas KNN modelio didėjant klasių skaičiui. Šis tikslumas mažėja, manoma, todėl, kad didesnis klasių skaičius ir tarpusavyje tarp klasių esantys bendri panašumai sunkina gesto atpažinimą iš kadro. Juolab svarbu ir tai, kad KNN mokosi iš kiekvieno kadro atskirai ir jų kitimas laike turi įtakos bendram rezultatui.

Kintant klasių skaičiui didėja RNN modelio tikslumas. Manoma, taip yra todėl, kad didesnis bendras duomenų kiekis leidžia geriau apsimokyti sistemai, kadangi apmokant 25 klasių modelį buvo sukurta daugiau nei 120 tūkstančių kadrų, iš kurių mokėsi modelis (palyginimui - 3 klasių modeliui buvo sukurti virš 14 tūkstančių kadrų).

Lyginant rezultatus, manoma, kad didesniai nei 88% daug įtakos tikslumui turėjo ir gerai paruošta LGK duomenų bazė. Toks bazės paruošimas palengvina apsimokymo procesą, pagerina rezultatus. Daug triukšmo ir netinkamų duomenų turintys rinkiniai, turėtų daug įtakos blogesniems rezultatams.

Reikėtų labiau įsigilinti į rezultatų priklausomybę nuo klasių ir apmokymo nustatymų. Reikėtų išbandyti RNN LSTM platų, platesnį būdus su geriausiais modeliais ir stebėti bendrus pasikeitimus. Taip pat reikėtų išbandyti didesnę duomenų augmentaciją, mokymui skirtą duomenų bazę praplėsti, išbandyti su daugiau klasių ir mokymo rinkinį paruošti iš daugiau skirtingų rodančiųjų duomenų bei daugiau trikdžių ir triukšmo turinčioje aplinkoje, pavyzdžiui, miške, prie spalvotų sienų ar kitur.

3.3.3. Modelio testavimas

Turimi visi trys modeliai buvo ištestuoti su 20% visų duomenų. Kiekvienas testavimo vaizdo įrašas buvo skaidomas į kadrus kaip ir apmokymo metu. Tačiau sistema KNN pagalba tik spėjo pagal kadrus, koks tai gestas galėtų būti ir tuomet visa surinkta informacija, buvo duodama RNN galutiniam sprendimui padaryti. Toliau pateikiami visų trijų apmokymų testavimai ir jų rezultatai.

10 lentelė. Lietuvių gestų kalbos modelių testavimo rezultatai

Bandymo Nr.	Nematytų duomenų kiekis	KNN pasirinkimas	RNN pasirinkimas
1.	30	83.92%	79.31%
2.	100	90.26%	87.34%
3.	250	88.17%	79.18%

10 lentelėje pateikiami lietuvių gestų kalbos sukurtų modelių testavimo rezultatai. Duomenys (vaizdo įrašai) duoti apmokymui ir testavimui beveik nesiskyrė. Kiekvieno vaizdo įrašas buvo automatiškai išskirstomas ar tai bus skirtas testavimui ar apmokymui.

Galima pastebėti, kad modeliai su dideliu tikslumu teisingai nusprendžia rodomo gesto klasę. Įdomu tai, kad kiekvienas modelis galutinį teisingą atsakymą atspėdavo 4 kartus iš 5. Tai rodo, kad modeliui didelės įtakos nedarė ar jis apmokytas atpažinti 3 ar 25 klases.

25 gestų klasių testavimo metu buvo pastebėta, kad nebuvo nei vieno teisingo galutinio atsakymo testuojant „višta“ klasę. Visais atvejais modelis spėjo „tėtis“ klasę, kadangi gestai yra ganėtinai panašūs - rodomi dešine ranka, rodomuoju (2) pirštu pradedant gestą nuo smakro. Skyrėsi tai, kad „vištos“ geste 1 ir 2 pirštai imituoja vištos snapą, o „tėtis“ klasėje 2 pirštas nuo smakro pakeliamas iki kaktos.

Geresniems ir tikslesniems testavimo rezultatams gauti reikėtų pirmiausia pabandyti ištestuoti su daugiau gestų pavyzdžių. Taip pat reikėtų išbandyti šių modelių testavimą su kitų žmonių rodomais gestais. Manoma, kad nufilmavus gestus trikdžių pilnoje aplinkoje, testavimo rezultatai būtų kur kas prastesni.

Rezultatai ir išvados

Rezultatai

1. Išanalizuotos konvoliucinių neuroninių tinklų ypatybės. Pastebėta, kad vienas plačiausiai naudojamų KNN modelis yra Inception v3. Neuroniniai tinklai labai gerai sugeba išskirti ir rasti pasikartojančias savybes;
2. Išanalizuotos rekurentinių neuroninių tinklų ypatybės. Išsiaiškinta, kad RNN sprendžia kitimo laike problemas, o paprastieji RNN turi nykstančiųjų gradientų problemą. Rastos ir išnagrinėtos architektūros, sprendžiančios šią problemą. Populiariausia iš jų - LSTM;
3. Išsiaiškinta, kad yra skirtingų tipų RNN, kurie gali generuoti skirtingą kiekį išeigų priimdami skirtingą kiekį įeigų. Iš šių savybių buvo pastebėta, kad labiausiai tinkantis RNN būdas šio darbo gestų kalbos atpažinimui - daug su vienu sąryšis (daug kadrų įeigų ir vienos klasės išeiga);
4. Išsiaiškinta, kad apjungtieji neuroniniai tinklai, kurie apjungia KNN ir RNN, gali būti plačiai pritaikomi tuose uždaviniuose, kuriuose reikalinga rasti savybes kiekviename laiko momente ir tas savybes apjungti į bendrą visumą. Taip pat išsiaiškinta, kad idealiu atveju tokių apjungiųjų nereikėtų, tačiau dabar RNN akcentuojasi labiau į kitimą laike, nei į savybių atpažinimą;
5. Išanalizuotos gestų kalbos ypatybės. Pastebėta, kad dinaminė gestų kalba ir jos mokymasis yra sudėtingas procesas, reikalaujantis labai daug laiko ir pastangų;
6. Išanalizuotos gestų kalbos vienetų atpažinimo iš video srauto galimybės. Pastebėta, kad geriausias būdas tai daryti - pasinaudoti apjungtuoju neuroninių tinklų modeliu;
7. Sukurta lietuvių gestų kalbos 25 skirtingų gestų duomenų bazė, susidedanti iš 22 klasių, kurių kiekviena turi po 20 vaizdo įrašų, ir 3 klasių, kurių kiekviena turi po 50 vaizdo įrašų;
8. Sukurti trys skirtingi lietuvių gestų kalbos modeliai, kurie atpažįsta 3, 10 ir 25 skirtingas LGK klases;
9. Sukurti trys lietuvių gestų kalbos modeliai buvo apmokyti su 80% ir ištestuoti su 20% visų duomenų;
10. Geriausius rezultatus davė 10 klasių modelis (87,34%), tačiau didesnę panaudojamumą turi 25 klasių modelis, kuris 79,18% tikslumu atspėdavo rodomo gesto klasę.

Išvados

1. Apjungtieji neuroniniai tinklai, kurie remiasi KNN ir RNN apjungtais modeliais yra labai galingas ir puikiai panaudojamas įrankis gestų kalbos atpažinime;
2. Pastebėta, kad apmokant RNN, jei epochų skaičius didesnis nei 50, rezultatuose nebus žymaus skirtumo;
3. Išsiaiškinta, kad vaizdo įrašus skaidant į kadrus ypač svarbu atkreipti dėmesį į kadro dydį, nes tai turi labai daug įtakos apsimokymo laikui ir užimamos vietos kiekiui, tačiau kadrų mažinimas beveik neturi įtakos modelio tikslumui;
4. Nustatyta, kad klasių skaičius (3, 10 ar 25) neturi didelės įtakos apsimokymo tikslumui ir

testavimo rezultatams;

5. Gauti aukšti rezultatai rodo gerai sudarytą apmokymui skirtų duomenų rinkinį, tačiau nėra aišku, koks tikslumas gautųsi testuojant ar apmokant su *nešvariais* duomenimis;
6. Norint lietuvių gestų kalbos modelį pritaikyti realiame pasaulyje, reikia daugiau studijų šioje srityje ir apmokyti jį su kelis kartus didesniu LGK klasių kiekiu;
7. Reikėtų kaupti lietuvių gestų kalbos gestų vaizdo įrašus, kad šiuos modelius būtų galima apmokyti atpažinti daugiau klasių. Surinkus užtektinai didelį duomenų kiekį, sistema būtų pritaikyta versti gestų kalbą į rašytinę kalbą ir atvirkščiai.

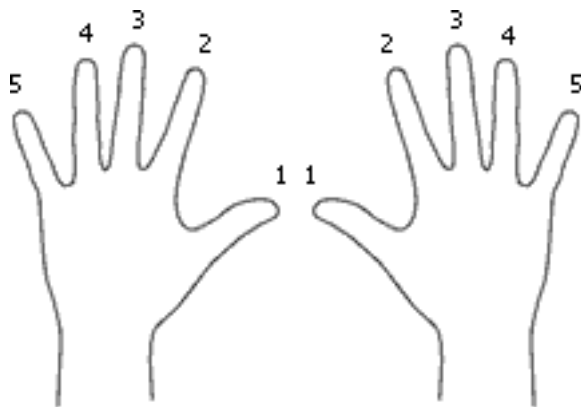
Santrumpos

- KNN - konvoliuciniai neuroniniai tinklai
- LGK - lietuvių gestų kalba
- LSTM - *long short-term memory*³
- NN - neuroniniai tinklai
- RNN - Rekurentiniai neuroniniai tinklai

³ilga trumpalaikė atmintis

Priedas 1

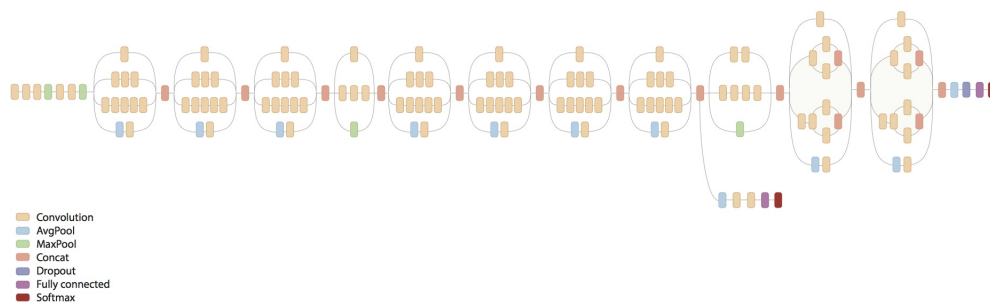
Rankų pirštų numeracija



15 pav. Kairės ir dešinės rankų pirštų numeracija

Priedas 2

Konvoliucinio tinklo modelis



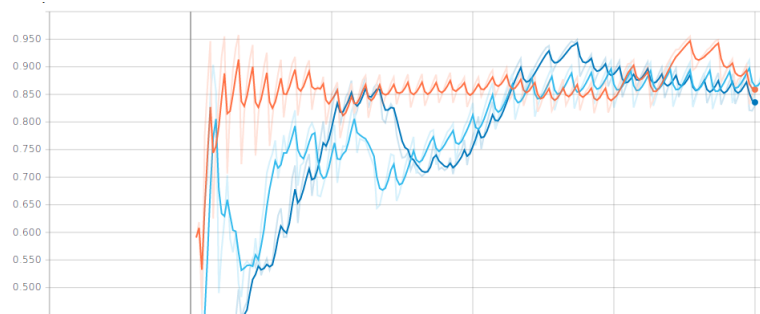
16 pav. Konvoliucinio tinklo modelis „Inception v3“

Priedas 3

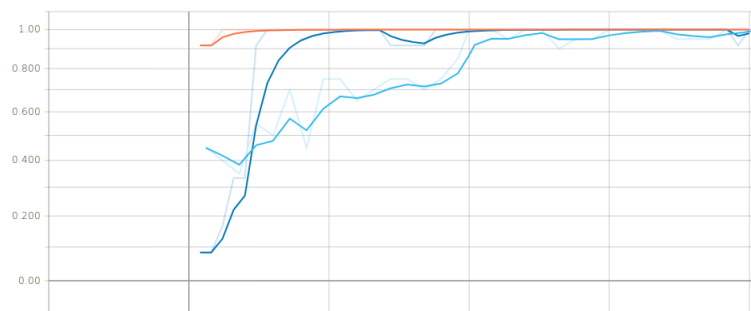
3 klasių apmokymas

RNN:

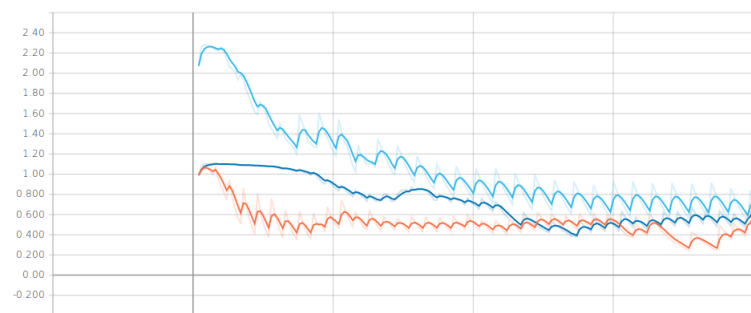
- Pirmas bandymas - oranžinė linija
- Antras bandymas - mėlyna linija
- Trečias bandymas - žydra linija



17 pav. RNN: Tikslumo grafikas



18 pav. RNN: Pasitikrinimo grafikas



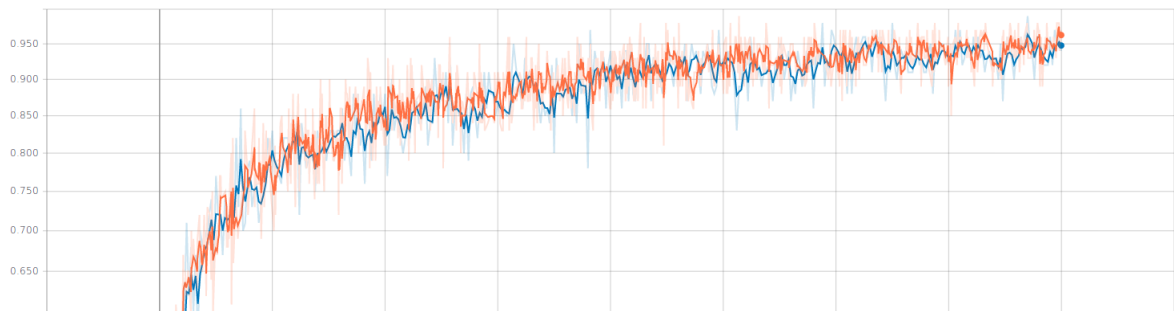
19 pav. RNN: Praradimo grafikas

Priedas 4

10 klasių apmokymas

KNN:

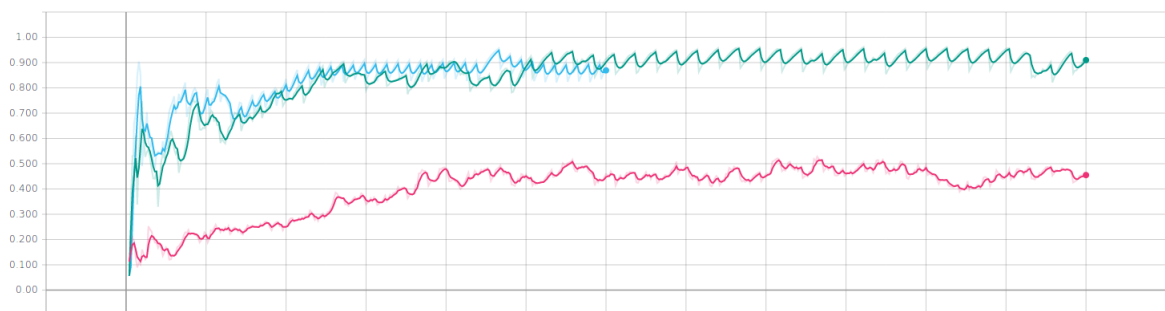
- Oranžinė linija - mokymas
- Mėlyna linija - pasitikrinimas



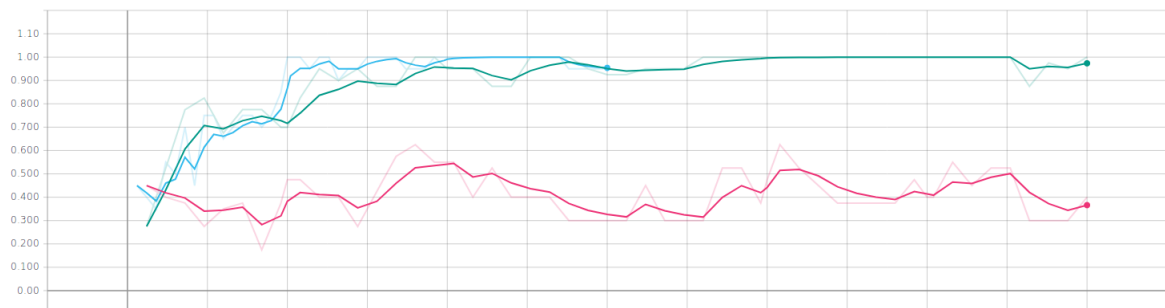
20 pav. KNN: Tikslumo grafikas

RNN:

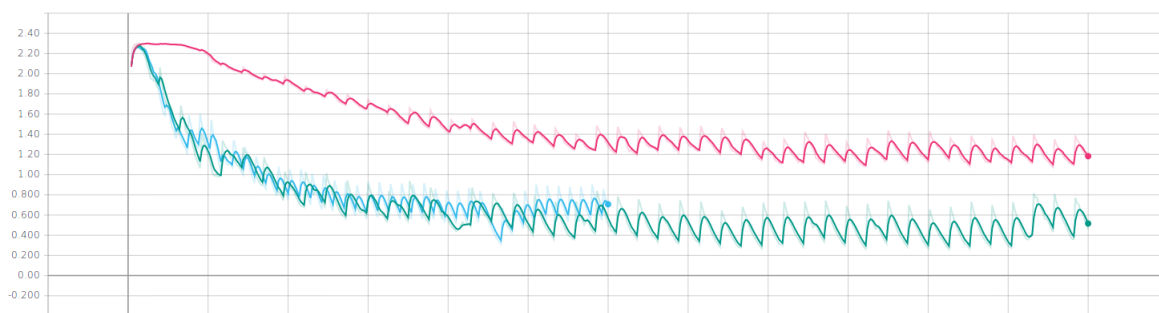
- Pirmas bandymas - mėlyna linija
- Antras bandymas - rausva linija
- Trečias bandymas - žalia linija



21 pav. RNN: Tikslumo grafikas



22 pav. RNN: Pasitikrinimo grafikas



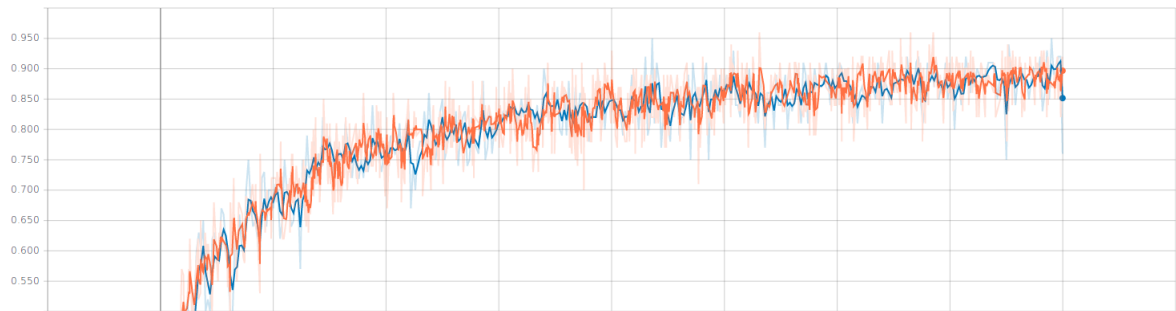
23 pav. RNN: Praradimo grafikas

Priedas 5

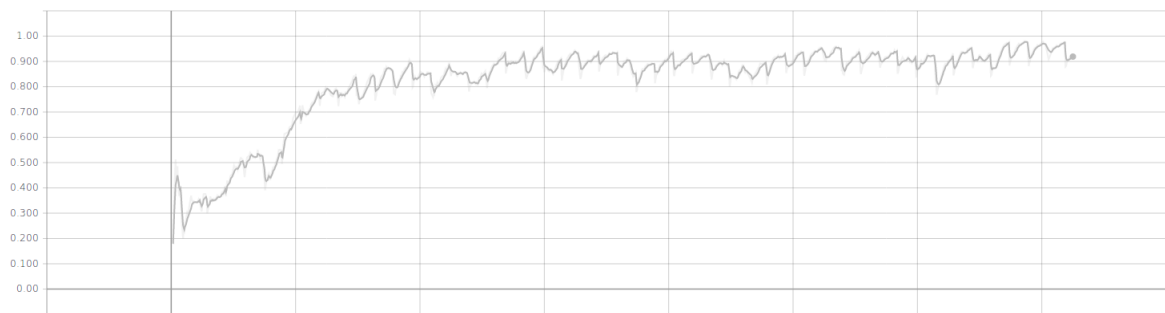
25 klasių apmokymas

KNN:

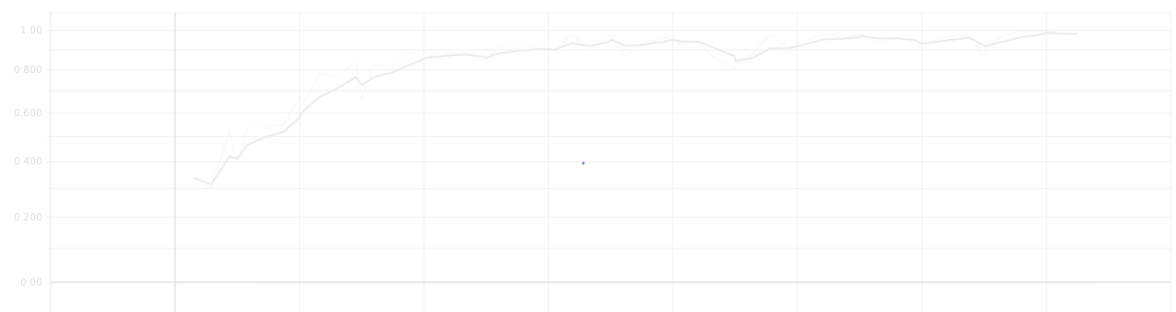
- Oranžinė linija - mokymas
- Mėlyna linija - pasitikrinimas



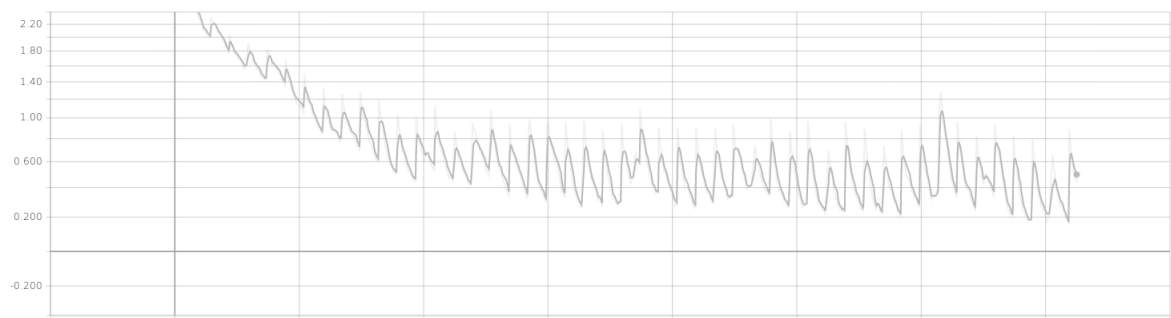
24 pav. KNN: Tikslumo grafikas



25 pav. RNN: Tikslumo grafikas



26 pav. RNN: Pasitikrinimo grafikas



27 pav. RNN: Praradimo grafikas