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

Glaustai aprašomas darbo turinys: pristatoma nagrinėta problema ir padarytos išvados. Santraukos apimtis ne didesnė nei 0,5 puslapio. Santraukų gale nurodomi darbo raktiniai žodžiai.

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

Summary

Santrauka anglų kalba. Santraukos apimtis ne didesnė nei 0,5 puslapio.

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

TURINYS

Įvadas

Pasaulyje yra virš 7 milijardų žmonių, kurie kasdien tarpusavyje komunikuoja. Netgi 5% visos žmonijos populiacijos sudaro žmonės, turintys klausos problemų. Vien 34 milijonai iš jų yra vaikai, iš kurių net 60% praradusių klausą vaikystėje galėjo būti girdintys dabar, jei būtų imtąsi atitinkamų prevencinių priemonių. Paskaičiuota, kad iki 2050 metų žmonių, turinčių šias problemas, skaičius išaugs netgi iki 900 milijonų, o 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 (ASL)*) yra trečia pagal populiarumą Amerikoje po anglų ir ispanų kalbų, kuria kalba virš 500 tūkstančių žmonių. Kiekviena šalis turi savo valstybinę kalbą - lietuvių, anglų, ispanų, rusų. Lygiai taip pat kiekviena šalis turi ir savo gestų kalbą. Tai yra tiek jau minėta amerikiečių gestų kalba (ASL), lietuvių, argentiniečių ir kitos gestų kalbos. Netgi tam tikri šalių regionai turi specifinius tos pačios kalbos dialektus, kaip, tarkime, vien Lietuvoje yra aukštaičių, žemaičių, suvalkiečių ar dzūkų tarmės.

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

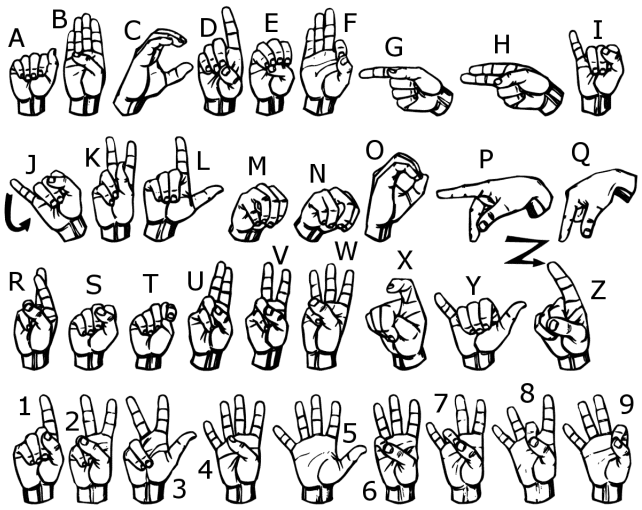
Vienas gestas turi turėti kelias prasmes. Kaip ir lietuvių kalboje žodis „kasa“ turi tris skirtingas reikšmes, taip ir gestų kalboje vienas gestas gali turėti keletą reikšmių. Tačiau iš kitos pusės gestas, parodytas truputėlį kitaip gali turėti visiškai priešingą reikšmę. Tarkime, ASL gestai „geras“ ir „blogas“ skiriasi tik puse į kurią atsuktas deltas, 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 nejudantys gestai, rodantys vieną raidę (*žr. 1 pav.*) ar net vieną žodį, kaip, pavyzdžiui, ASL „*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 gestakalbiai prisistatydami parodo gestą, kuris priklauso tik jiems. Tai tarsi parašas tam, kad nebereikėtų kreipiantis ar apibūdinant žmogų jo vardo sakyti paraidžiui.

[1] liet. Aš tave myliu



1 pav. Amerikiečių gestų kalbos abėcėlė

2. **Dinaminė gestų kalba** - tai žodžių lygio gestų kalba. Nesunku pastebėti, kad 1 paveikslėlyje yra „J“ ar „Z“ raidės, kurios priskiriama dinaminių judesių klasei. Kaip ir 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, ASL yra „labas“, „gerai“ ar „blogai“.
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 ataskymas „taip“ arba „ne“. Tačiau, jei antakiai bus suraukti, tai reikš, kad klausiama su paaiškinimu „kas“, „kur“, „kaip“, „ką“.

Darbo tikslas

Išanalizuoti gestų kalbos vienetų atpažinimo galimybes ir video srauto.

Darbo uždaviniai

- Gestų kalbos video srautų paieška ir mokomosios medžiagos neuroniniams tinklams surinkimas
- Susipažinimas su rekurentiniais neuroniniais tinklais
- Gestų kalbos vienetų atpažinimas iš video srauto pasinaudojant rekurentiniais neuroniniais tinklais.

Darbo eiga

- Panašių ir jau įgyvendintų projektų paieška
- Esamos sistemos patobulinimai
- Rezultatų palyginimai

Panaudotos priemonės

- Python – programavimo kalba
- TensorFlow – skirta darbui su apsimokančiomis sistemomis[2]

Įvade nurodomas darbo tikslas ir uždaviniai, kuriais bus įgyvendinamas tikslas, aprašomas temos aktualumas, apibrėžiamas tiriamasis objektas akcentuojant neapibrėžtumą, kuris bus išspręstas darbe, aptariamos teorinės darbo prielaidos bei metodika, apibūdinami su tema susiję literatūros ar kitokie šaltiniai, temos analizės tvarka, darbo atlikimo aplinkybės, pateikiama žinių apie naudojamus instrumentus (programas ir kt., jei darbe yra eksperimentinė dalis). Darbo įvadas neturi būti dėstymo santrauka. Įvado apimtis 2–4 puslapiai.

[2]angl. Machine learning

1. Apsimokančios sistemos

Apsimokančios sistemos (*angl. machine learning*)

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 *b*. Tai būdas, kuris veikia medžio principu.

1 lentelė. Pavyzdinis prižiūrimo mokymo apmokymui paruoštų duomenų rinkinys

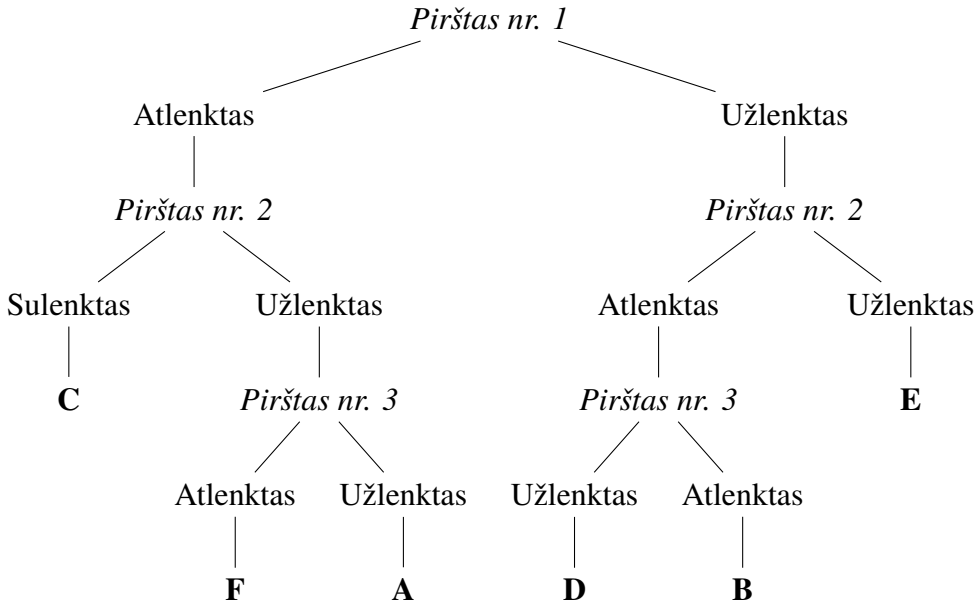
Nr.	**Pirštas nr. 1**	**Pirštas nr. 2**	**Pirštas nr. 3**	**Pirštas nr. 4**	**Pirštas nr. 5**	**Raidė**
1.	Atlenktas	Užlenktas	Užlenktas	Užlenktas	Užlenktas	**A**
2.	Užlenktas	Atlenktas	Atlenktas	Atlenktas	Atlenktas	**B**
3.	Atlenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas	**C**
4.	Užlenktas	Atlenktas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

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

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 medžio galimybių medžio galima matyti, kad pilnai užtenka sprendimui nusakyti 3 pirštų, kadangi rezultatų nėra daug. Jei būtų imama visa abėcėlės aibė, tuomet rezultato nustatymui būtų naudojama galimai visų pirštų padėtys. Tačiau net ir šį medį optimizavus galima būtų, tarkime, C raidė atsakymą gauti tik iš vieno piršto padėties, kadangi tik ši vienintelė raidė turi sulenkto piršto padėtį. Galiausiai iš šio medžio galima pastebėti, kad 1 lentelėje pateikto pavyzdžio atsakymas yra raidė **C**.

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 ir bet kokio žodyno. Kuomet pastoviai matomas vis tas pats tekstas, žodžiai tampa atpažįstami, tačiau išversti jų neišeina. 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.	Užlenktas	Atlenktas	Atlenktas	Atlenktas	Atlenktas
3.	Atlenktas	Sulenktas	Sulenktas	Sulenktas	Sulenktas
4.	Užlenktas	Atlenktas	Užlenktas	Užlenktas	Užlenktas
5.	Užlenktas	Užlenktas	Užlenktas	Užlenktas	Užlenktas
6.	Atlenktas	Užlenktas	Atlenktas	Atlenktas	Atlenktas

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.

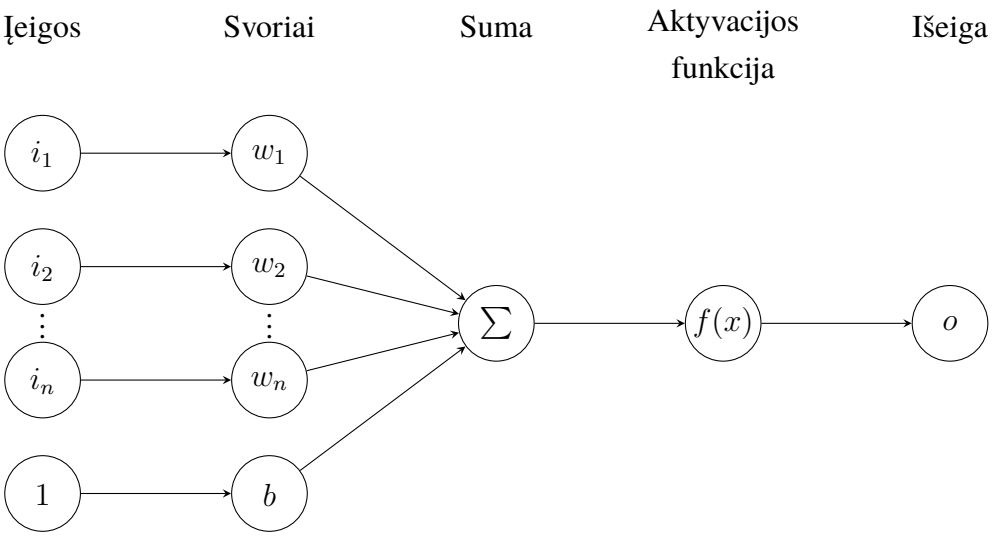
2. Neuroniniai tinklai

Žmogaus smegenys yra labai sudėtingas, nelijinis ir paralelinis kompiuteris [**Hay09**]. Kiekvieno kūnas yra sudarytos 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. Tai, kaip ir buvo paminėta, yra 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 sužadinimą gavęs neuronas sulaukia atsakymo.

2.1. Perceptronas

Parceptronas (*angl. perceptron*) – kompiuterinis modelis, skirtas atkartoti žmogaus smegenų darbą. Toliau pateikiamas perceptrono pavyzdys.

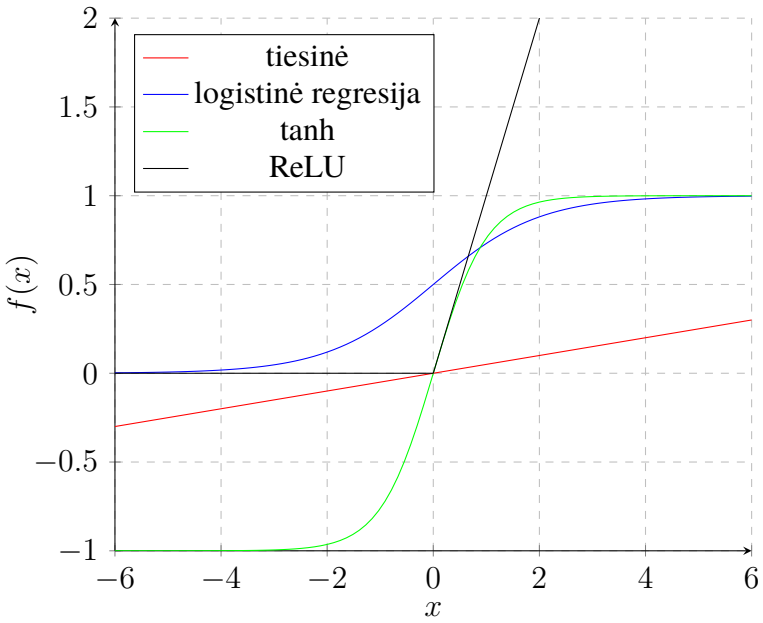


3 pav. Perceptrono pavyzdys

3 paveikslėlyje pavaizduotame pavyzdyje esančią išeigą galima aprašyti formule:

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

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



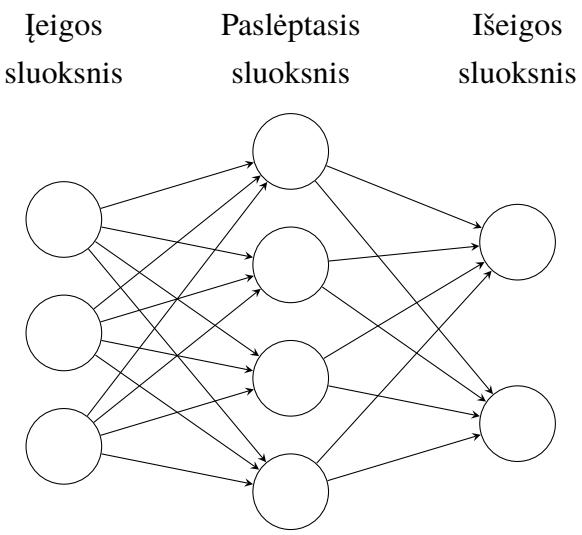
4 pav. Aktyvacijos funkcijos

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

Daugiasluoksnis perceptronas (*angl. multilayer perceptron*) – strukūra, sudaryta iš kelių sluoksnių perceptronų.



5 pav. Daugiasluoksnio perceptrono pavyzdys

Dažniausiai daugiasluoksnis perceptronas turi tris ar daugiau sluoksnių – įeigos (*input layer*), paslėptasis (*hidden layer*) ir išeigos (*output layer*) sluoksnių. Paslėptajame sluoksnyje gali būti daugiau nei vienas sluoksnis. Daugiasluoksnis perceptronas kaip aktyvacijos funkciją naudoja nelinijines 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 daugiausluoksniai perceptronai naudojami atpažinti žodinę kalbą ar versti tekstus.

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. convoliutional 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. convoliution 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 matricų 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.

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

$$0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 8 \cdot 0 + 0 \cdot 1 + 9 \cdot 0 + 2 \cdot 1 = 3 \tag{3}$$

$$0 \cdot 1 + 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 0 + 8 \cdot 1 + 6 \cdot 0 + 9 \cdot 1 + 2 \cdot 0 + 4 \cdot 1 = 21 \tag{4}$$

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

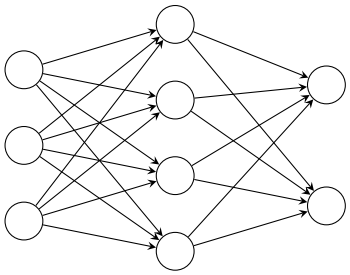
$$\begin{bmatrix} 1 & 3 & 1 & 3 \\ 2 & 5 & 4 & 2 \\ 4 & 3 & 2 & 5 \\ 2 & 5 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 5 & 5 \end{bmatrix} \tag{5}$$

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

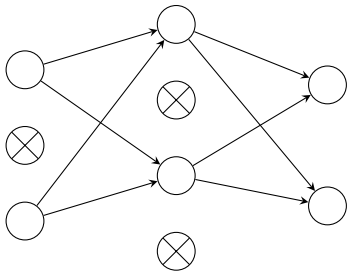
$$max(\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}) = 4 \tag{7}$$

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**].



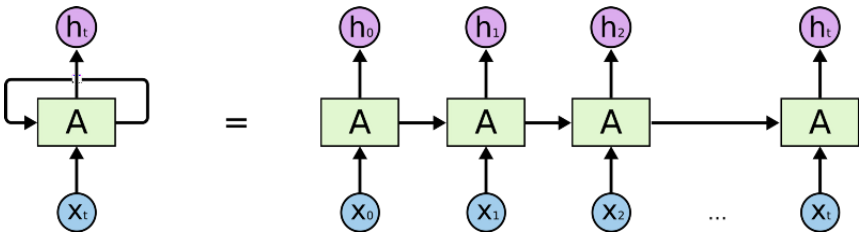
6 pav. Standartinis neuroninis tinklas



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**].



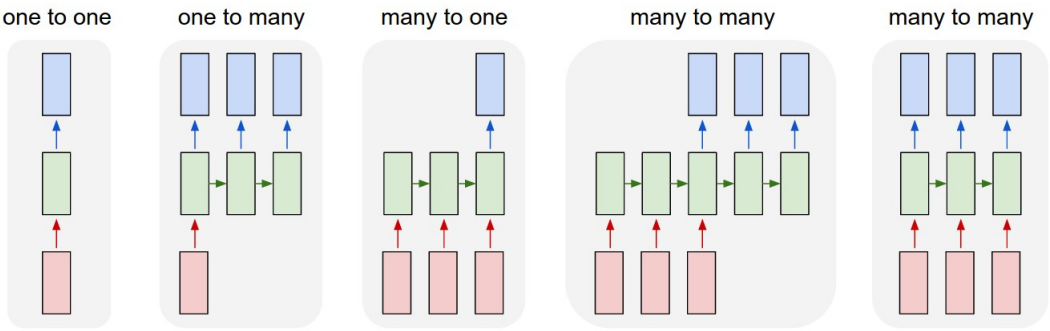
8 pav. Rekurentinių neuroninių tinklų veikimo principas

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

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

Kur h_t - paslėpto sluoksnio būsena laiko momentu t, kurią dar būtų galima vadinti t žingsnio išeiga, f_w - funkcija f su parametrais w, h_{t-1} - praėjusio žingsnio būsena, o x_t - įeigos vektorius. Iš šios formulės galima pastebėti, kad kiekviena būsena gauna praeito žingsnio būsena, 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 kvadratėlis reiškia įeigą, žalsvas - paslėptuosius sluoksnius, o melsvas - išeigą. Pateikiami šie būdai:

- **Vienas su vienu** (*angl. one to one*) – būdas, kuriame yra viena įeiga, 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 įeiga, bet 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 įeigų, bet tik viena išeiga. Tokio būdo pavyzdys galėtų būti vieno žodžio, tarkime, „labas“ atpažinimas iš video sraudo.
- **Daug su daug** (*angl. many to many*) – būdas, kuriame yra daug įeigų ir daug išeigų. Šis būdas gali būti skaidomas į dvi dalis:
  - **Priklausomas** - įeigų skaičius sutampa su išeigų skaičiumi. Kiekviena įeiga turi savo išeigos atitikmenį. Tai būtų dalinai galima gretitinti su *vienas su vienu* būdu. Pavyzdys šios atšakos galėtų būti video srauto klasifikacija pagal kiekvieną kadrą - nuolatinis atnaujinimas, to kas galėjo būti pasakyta, pavyzdžiui, gestų kalboje.
  - **Nepriklausomas** - įeigos skaičius galimai nesutampa su išeigų skaičiumi. Kiekviena įeiga 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.

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 reiškmes iš praeitame žingsnyje turimų duomenų ir kaip įeiga priima praeito laiko momento išeigą.

$$f(w_n \cdot o_n) = o_{n+1} \tag{9}$$

Šioje formulėje w_n - n-tojo sluoksnio svoris, o_n - n-tojo sluoksnio išeiga, 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 $f(o_n)$ yra praradimas bus funkcija f, kurios parametras o_n yra paskutinio sluoksnio išeiga.

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

$$\frac{\partial Loss}{\partial w_n} = \frac{\partial Loss}{\partial f(o_n)} \cdot \frac{\partial f(o_n)}{\partial w_n} = \frac{\partial Loss}{\partial f(o_n)} \cdot f'(o_n) \cdot w_n \tag{10}$$

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

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

$$\frac{\partial Loss}{\partial w_1} = \frac{\partial Loss}{\partial f(o_n)} \cdot \frac{\partial f(o_n)}{\partial o_{n-1}} \cdots \frac{\partial f(o_2)}{\partial o_1} \cdot \frac{\partial f(o_1)}{\partial w_1} = \frac{\partial f(Loss)}{\partial f(o_n)} \cdot f'(o_n) \cdot w_n \cdots f'(o_1) * w_1 \tag{11}$$

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 pateikiami keletos iš tokių neuroninių tinklų architektūrų pavyzdžių.

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 paslėptojo sluoksnio mazags (*angl. node*) yra pakeistas atminties ląstele [**DBLP:journals/corr/Lipton15**].

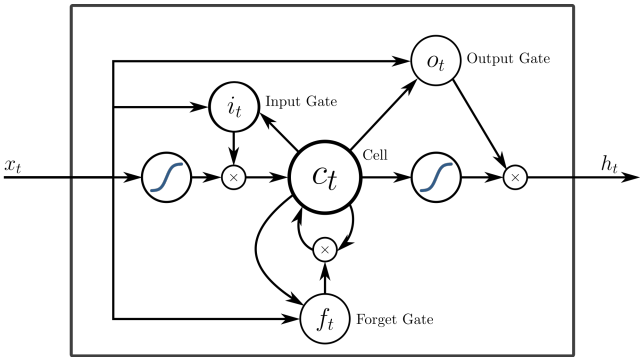
LSTM – ilga trumpalaikė atmintis (*angl. long short-term memory*) - RNN architektūra, kuri sugeba atsiminti informaciją ilgesniam laiko tarpui.

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 būsena 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 atnaujina ląstelės būseną.
- Išeigos vartai (*angl. output gate*) nusprendžia, kurios ląstelės būsenos reikšmės bus pridedamas į paslėptąją būseną, kuri bus visos ląselės išeiga. Taip pat labai svarbu paminėti ir faktą, kad pasiliekamos ir tos reikšmės ar būsenos, kurios manoma, kad bus reikalingos ateityje.



10 pav. Atminties mazgo pavyzdys

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 būdų iš kadrų sekos RNN sugeba generuoti vieną išeigą. Kitaip tariant duodant, pavyzdžiui, video srautą, bus gaunama konkrečios klasės išeiga.

Galima pastebėti, kad būtent šiuo atveju, modelis, kuris apdoros video srautą ir nuspręs, kurios klasės įeiga buvo įeitimi 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 modeliu 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 savybių.

Naudojantis jų jau sukaupta patirtimi tobulinama sistema galiausiai atpažinti ir lietuvių kalbą.

3.2. Argentiniečių gestų kalbos atpažinimas

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

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 tikslumas.

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

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 buvo sistema buvo apmokyta KNN Inception v3 modeliu vieną kartą, o toliau buvo keičiami RNN apmokymo būdai. Galima pastebėti, kad giliuoju (*angl. deep*) būdu rezultatai buvo prasčiausi.

Toliau bus aptariami skirtingi skirtingi RNN tinklo apmokymo tipai. Visi iš jų buvo apmokomi naudojantis LSTM modeliu. Visi keturi penki būdai naudojasi *categorical crossentropy* praradimo apskaičiavimo metodu, *softmax* aktyvacijos būdu ir *adam* optimizatoriumi.

3.2.1. Paprastas tinklas

Sudėtis:

- Du sluoksniai
- Pirmajame - atsisakymo lygis 0,8
- 128 - sluoksnių vienetų skaičius

3.2.2. Gilus tinklas

Sudėtis:

- Trys sluoksniai
- Atsisakymo lygis 0,2 visuose lygiuose
- 64 - sluoksnių vienetų skaičius

3.2.3. Platus tinklas

Sudėtis:

- Vienas sluoksnis
- Atsisakymo lygis 0,2
- 256 - sluoksnių vienetų skaičius

3.2.4. Platesnis tinklas

Sudėtis:

- Vienas sluoksnis
- Atsisakymo lygis 0,2
- 512 - sluoksnių vienetų skaičius

3.3. Lietuvių gestų kalbos atpažinimas

Lietuvių kalbai su gestų kalbos atpažinimu naudojantis konvoliuciniais ar rekurentiniais neuroniniais tinklais oficialaus nieko nėra. Todėl toliau pateikiama, kas buvo atlikta ir kokie rezultatai buvo gauti naudojantis šiais metodais jau anksčiau aptartu ir perdarytu modeliu.

3.3.1. Duomenų paruošimas

Lietuvių gestų kalbos žodyne 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. Tačiau iškyla viena problema - kiekvienas gestas turi tik po vieną video įrašą atitinkantį tą žodį. Toks kiekis duomenų yra per mažas, norint apmokyti sistemą. 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. Filmuota buvo mobiliuoju telefonu atsistojus prie gelsvos sienos. Filmuoti buvo du skirtingi asmenys, kurių kiekvienas atliko po 25 vaizdo įrašus kiekvienai klasei.

3.3.2. Modelio apmokymas

Modelį buvo nuspręsta apmokyti pasinaudojant jau turimomis žiniomis ir jau turimu sukurtu modeliu jį patobulinant. Visų pirma buvo apmokyta sistema su trimis klasėmis „labas“, „mano“ ir „vardas“. Šie gestai visi atliekami dešiniąja ranka, todėl tai sistemai buvo manyta, kad turėtų šiek tiek palengvinti darbą su duomenimis.

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

3.3.2.1. Pirmasis apmokymas

Duomenys:

- 3 klasės
- 45 vaizdo įrašai kiekvienai klasei

Rezultatai:

- 3 klasės 45 kiekvienai klasei. bendrai 145
- 121 training samples, 14 validation, 100 epochs, 86.93% accuracy, 0.5081 total loss
- 92.31% tikslumas iš nematytų 13 video.

3.3.2.2. Rezultatai

8 lentelė. Lietuvių gestų kalbos bandymų rezultatai

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

3.3.3. Modelio testavimas

4. Medžiagos darbo tema dėstymo skyriai

Medžiagos darbo tema dėstymo skyriuose išsamiai pateikiamos nagrinėjamos temos detalės: pradiniai duomenys, jų analizės ir apdorojimo metodai, sprendimų įgyvendinimas, gautų rezultatų apibendrinimas.

Medžiaga turi būti dėstoma aiškiai, pateikiant argumentus. Tekste dėstomas trečiuoju asmeniu, t.y. rašoma ne „aš manau“, bet „autorius mano“, „autoriaus nuomone“. Reikėtų vengti informacijos nesuteikiančių frazių, pvz., „...kaip jau buvo minėta...“, „...kaip visiems žinoma...“ ir pan., vengti grožinės literatūros ar publicistinio stiliaus, gausių metaforų ar panašių meninės išraiškos priemonių.

Skyriai gali turėti poskyrius ir smulkesnes sudėtines dalis, kaip punktus ir papunkčius.

Rezultatai ir išvados

Rezultatų ir išvadų dalyje išdėstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas įdiegta), toliau pateikiamos išvados (daromi nagrinėtų problemų sprendimo metodų palyginimai, siūlomos rekomendacijos, akcentuojamos naujovės). Rezultatai ir išvados pateikiami sunumeruotų (gali būti hierarchiniai) sąrašų pavidalu. Darbo rezultatai turi atitikti darbo tikslą.

Sutartiniai žymėjimai

- i_t – įeiga laiko momentu t
- o_t – išeiga laiko momentu t
- h_t – būsena laiko momentu t

Sąvokų apibrėžimai

- Dirbtiniai neuroniniai tinklai - artificial neural networks
- Inception v3 - Google modelis
- Išeiga - output
- Įeiga - input
- Konvoliuciniai neuroniniai tinklai - convolutional neural networks
- Neuroniniai tinklai - neural networks
- Paslėptasis sluoksnis - hidden layer
- Rekurentiniai neuroniniai tinklai - recurrent neural networks
- Sluoksnis - layer
- Vienpusiai neuroniniai tinklai - Feed-Forward neural networks

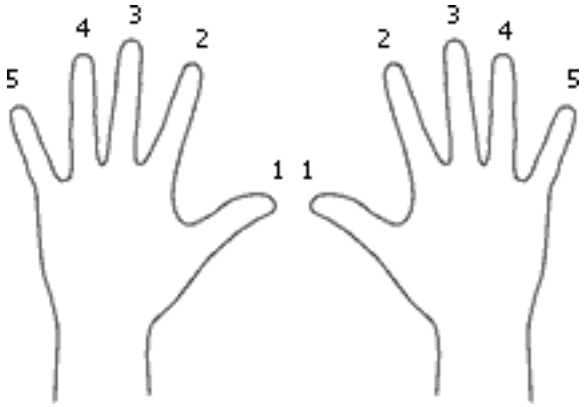
Santrumpos

- KNN - konvoliuciniai neuroniniai tinklai
- NN - neuroniniai tinklai
- RNN - Rekurentiniai neuroniniai tinklai

Sąvokų apibrėžimai ir santrumpų sąrašas sudaromas tada, kai darbo tekste vartojami specialūs paaiškinimo reikalaujantys terminai ir rečiau sutinkamos santrumpos.

Priedas 1

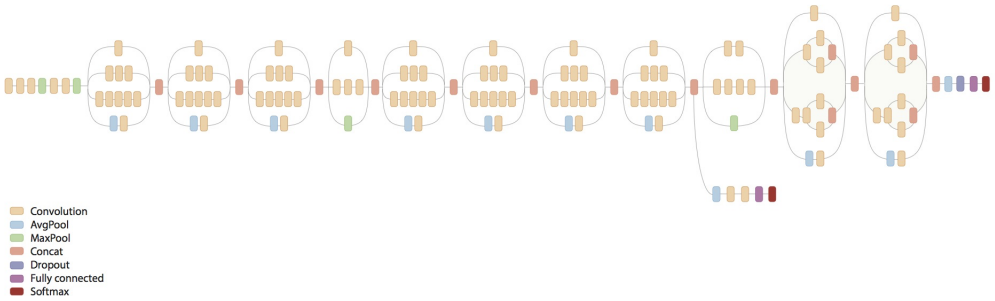
Rankų pirštų numeracija



11 pav. Kairės ir dešinės rankų pirštų numeracija

Priedas 2

Konvoliucinio tinklo modelis



12 pav. Konvoliucinio tinklo modelis „Inception v3“