

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

|  |    |
|--|----|
| IVADAS .....   | 4  |
| Gestų kalba .....                                    | 4  |
| Gestų kalbos specifika .....                         | 4  |
| Darbo tikslas .....                                  | 5  |
| Darbo uždaviniai .....                               | 5  |
| Darbo eiga .....                                     | 5  |
| Panaudotos priemonės .....                           | 6  |
| 1. APSIMOKANČIOS SISTEMOS .....                      | 7  |
| 1.1. Prižiūrimas mokymas .....                       | 7  |
| 1.2. Neprižiūrimas mokymas .....                     | 8  |
| 1.3. Praktinis mokymas .....                         | 9  |
| 2. NEURONINIAI TINKLAI .....                         | 10 |
| 2.1. Perceptronas .....                              | 10 |
| 2.2. Daugiasluosknis perceptronas .....              | 11 |
| 2.3. Dirbtiniai neuroniniai tinklai .....            | 12 |
| 2.4. Konvoliuciniai neuroniniai tinklai .....        | 12 |
| 2.4.1. Konvoliucinis sluoksnis .....                 | 12 |
| 2.4.2. Telkimo sluoksnis .....                       | 13 |
| 2.4.3. Atsisakymo sluoksnis .....                    | 14 |
| 2.5. Rekurentiniai neuroniniai tinklai .....         | 14 |
| 2.5.1. Rekurentinių neuroninių tinklų tipai .....    | 15 |
| 2.5.2. Rekurentinių neuroninių tinklų modeliai ..... | 16 |
| 2.5.2.1. LSTM .....                                  | 16 |
| 2.5.3. GRU .....                                     | 18 |
| 2.6. Apjungiamieji tinklų modeliai .....             | 18 |
| 3. EKSPERIMENTINĖ DALIS .....                        | 19 |
| 3.1. Panašūs darbai .....                            | 19 |
| 3.2. Argentiniečių gestų kalbos atpažinimas .....    | 19 |
| 3.3. Lietuvių gestų kalbos atpažinimas .....         | 20 |
| 3.3.1. Duomenų paruošimas .....                      | 20 |
| 3.3.2. Modelio apmokymas .....                       | 20 |
| 3.3.2.1. Pirmasis apmokymas .....                    | 21 |
| 3.3.2.2. Rezultatai .....                            | 21 |
| 3.3.3. Modelio testavimas .....                      | 21 |
| REZULTATAI IR IŠVADOS .....                          | 22 |
| SUTARTINIAI ŽYMĖJIMAI .....                          | 23 |
| SAVOKŲ APIBRĖŽIMAI .....                             | 24 |
| SANTRUMPOS .....                                     | 25 |
| PRIEDAI .....  | 25 |
| 1 priedas. Rankų pirštų numeracija .....             | 26 |
| 2 priedas. Konvoliucinio tinklo modelis .....        | 27 |

# 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 imtasi 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 (toliau - ASL)*) yra trečia pagal populiarumą Amerikoje ne anglų 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 kaip jau minėta amerikiečių, lietuvių (*toliau - LGK*), argentiniečių ir kitos gestų kalbos. Netgi tam tikri šalių regionai turi specifinius tos pačios kalbos dialektus, kaip, tarkime, vien Lietuvoje yra žodinės kalbos aukštaičių, žemaičių, suvalkiečių ar dzūkų tarmės.

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 iš kitos pusės 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

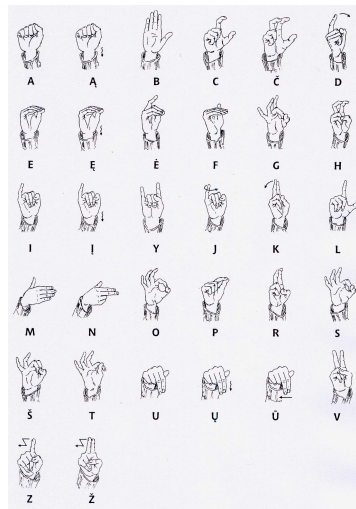
Kiekviena gestų kalba susideda iš **trijų** pagrindinių dalių:

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 „*I love you*“<sup>1</sup> 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 parašas tam, kad nebereikėtų

---

<sup>1</sup> liet. Aš tave myliu

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. 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, LGK yra „labas“, „mano“ ar „vardas“.
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 iš video srauto.

## Darbo uždaviniai

- Gestų kalbos video srautų paieška ir rengimas bei 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 (*angl. machine learning*);
- Inception v3 – konvoliucinių neuroninių tinklų modelis;
- LSTM – rekurentinių neuroninių tinklų metodas;
- OpenCV – Python įrankis 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     |

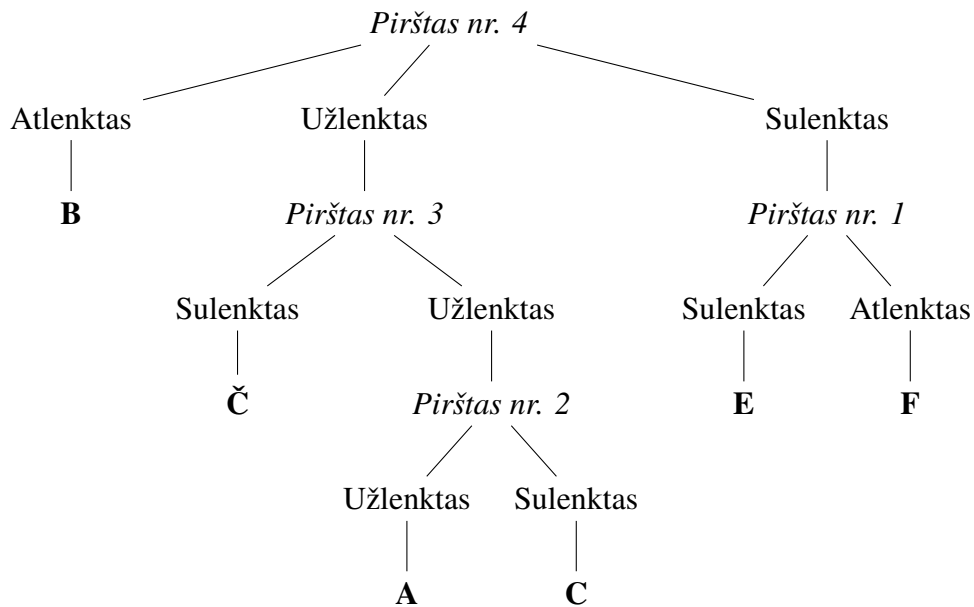
1 lentelėje pateikiamas pavyzdys su supaprastinta lietuvių gestų kalbos abėcėle. Lentelėje pateikiamos pirštų padėties, o pirštai numeruojami pagal 1 priede pateikiamą pirštų numeraciją. Kiekvieno piršto padėtis šiame pavyzdyje gali būti: *atlenktas*, *sulenktas*, *užlenktas*. Ir kiekvienai padėčiai 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 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.

## 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 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.  | 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.

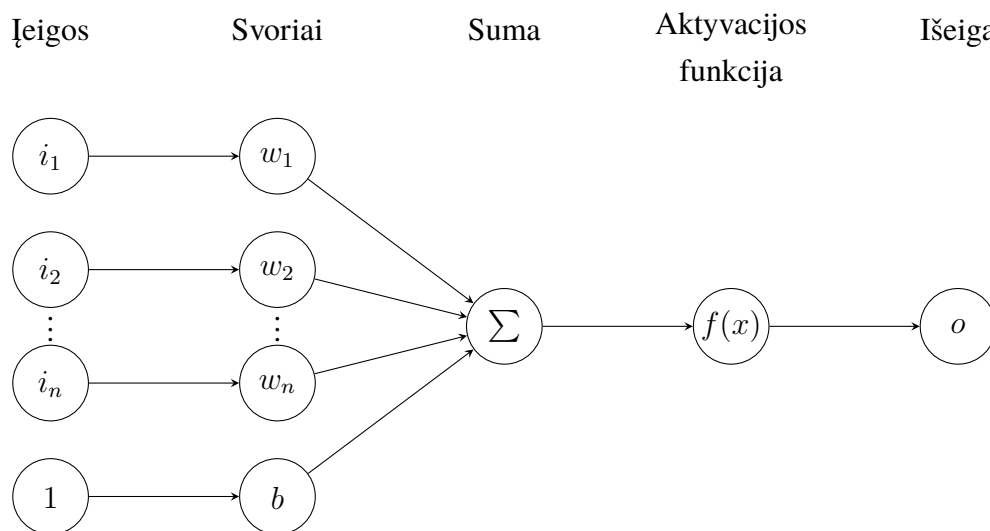
## 2. Neuroniniai tinklai

Žmogaus smegenys yra labai sudėtingas, nelijinis 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. Tai, kaip ir buvo paminėta, 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 sužadinimą gavęs neuronas sulaukia atsakymo.

### 2.1. Perceptronas

Perceptronas (*angl. perceptron*) – kompiuterinis modelis, skirtas atkartoti žmogaus kūne esančių neuronų darbą. Toliau pateikiamas perceptrono pavyzdys.



3 pav. Perceptrono pavyzdys

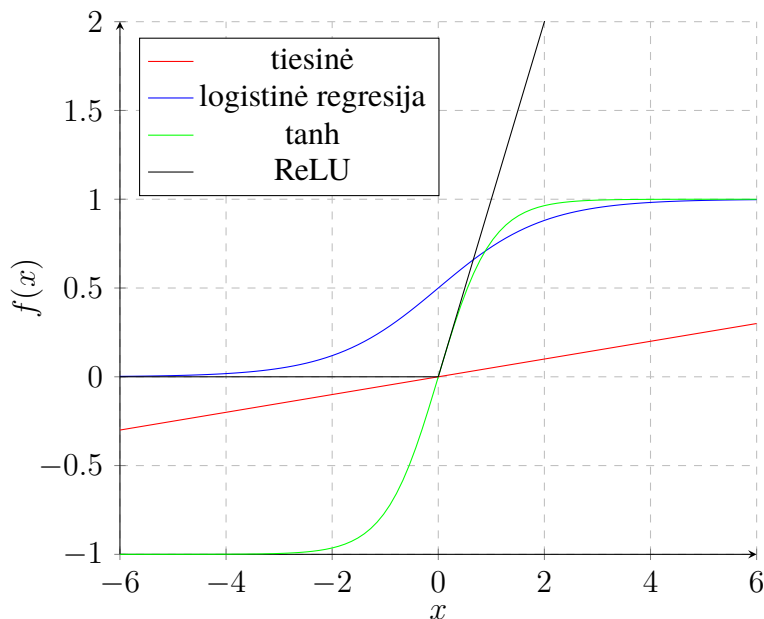
3 paveikslėlyje pavaizduotame pavyzdyje esančią išėigą galima aprašyti formule:

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

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 įėjigas (*angl. input*). Visų šių įėjigų 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:

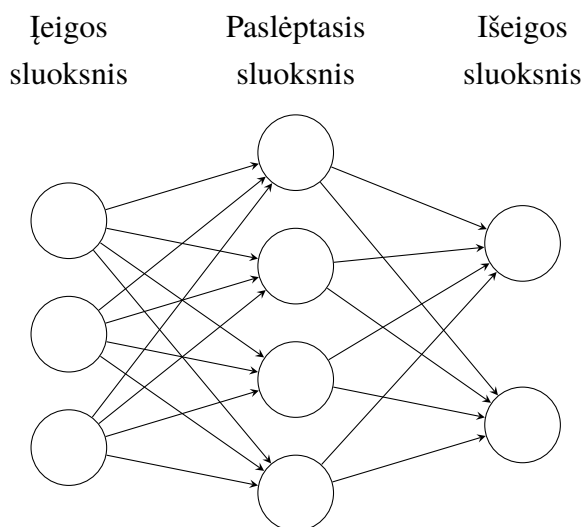
- Tiesinė –  $f(x) = a \cdot x$ ;
- Logistinės regresijos –  $f(x) = \frac{1}{1+e^{-x}}$ ;
- Tanh –  $f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$ ;
- ReLU –  $f(x) = \begin{cases} 0 & , \text{kai } x < 0 \\ x & , \text{kai } x \geq 0 \end{cases}$ .

## 2.2. Daugiasluosknis perceptronas

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

Dažniausiai daugiasluosknis perceptronas turi tris ar daugiau sluoksnių – įėjimo (*input layer*), paslėptasis (*hidden layer*) ir išėjimo (*output layer*) sluoksnius. 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 paveikslėlyje.



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  $3 \times 3$  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     | $3 \times 3$ | 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  $2 \times 2$  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     | $2 \times 2$ | 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.

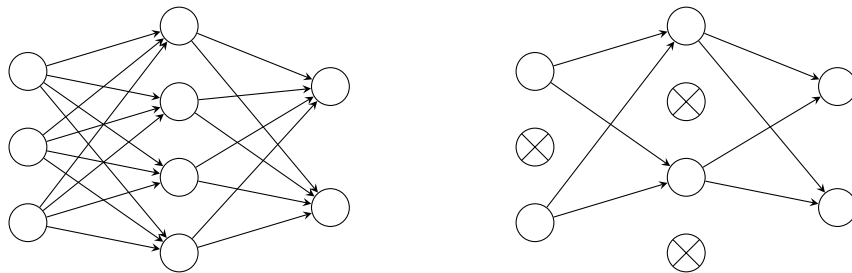
$$\begin{bmatrix} 1 & 3 & 1 & 3 \\ 2 & 5 & 4 & 2 \\ 4 & 3 & 2 & 5 \\ 2 & 5 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 5 & 5 \end{bmatrix} \quad (5)$$

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

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

### 2.4.3. Atsisakymo sluoksnis

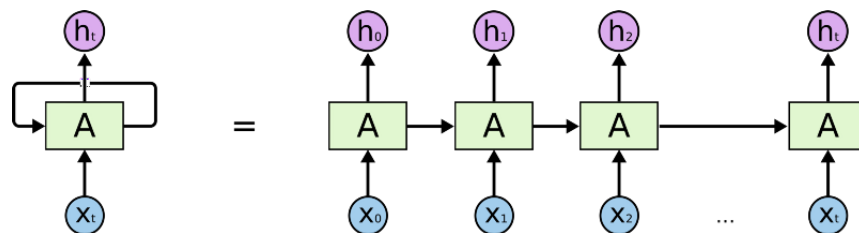
**Atsisakymo sluoksnis** (*angl. dropout layer*) – konvoliucinių tinklų sluoksnis, skirtas normalizuoti ir sureguliuoti tarpusavyje susijusių neuronų sąryšius, skirtus perduoti signalus. Mokymo fazėje dažniausiai ištrinamos neuronuose esančios reikšmės tam, kad šis per naują apsimokytų. Galimai netgi 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*) – vienus neuroninius tinklus, kurie remiasi daugiasluoksnio perceptrono principu. Šie tinklai, apima kitų laiko vienetų apdorotą informaciją ir bendrą kitimą laike [DBLP:journals/corr/Lipton15].



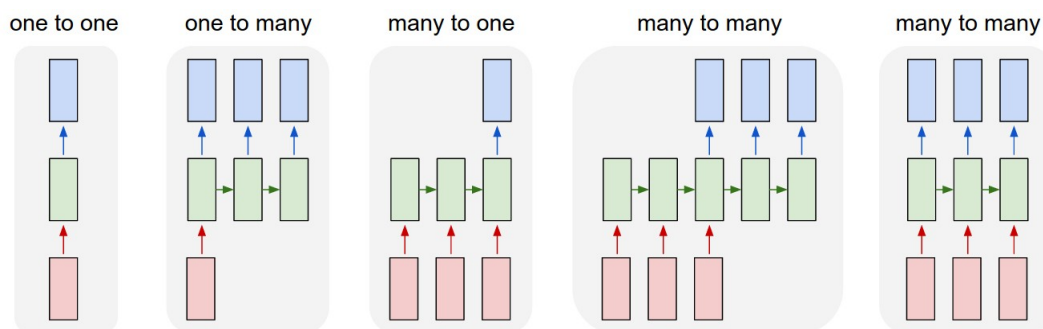
8 pav. Rekurentinių neuroninių tinklų veikimo principas

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,  $f_w$  - 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 įėjimą, žalsvas - paslėptuosius sluoksnius, o mėlsvas - išeigą. Pateikiami šie būdai:

- **Vienas su vienu** (*angl. one to one*) – būdas, kuriame yra viena įėjimas, 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. Dažnai naudojami KNN;
- **Vienas su daug** (*angl. one to many*) – būdas, kuriame yra viena įėjimas, bet kelios išeigos. Vienas iš panaudojimo būdų galėtų būti sakinių 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 įėjimų, 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 įėjimų ir daug išeigų. Šis būdas gali būti skaidomas į dvi dalis:
  - **Priklausomas** - įėjimų skaičius sutampa su išeigų skaičiumi. Kiekviena įėjimas turi savo išeigos atitikmenį. Tai būtų dalinai galima gretintinti 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** - įėjimo skaičius galimai nesutampa su išeigų skaičiumi. Kiekviena įėjimas 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 reikšmes iš praeitame žingsnyje turimų duomenų ir kaip įeiga priima praeito laiko momento išeigą.

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

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 naudojama RNN architektūra 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].

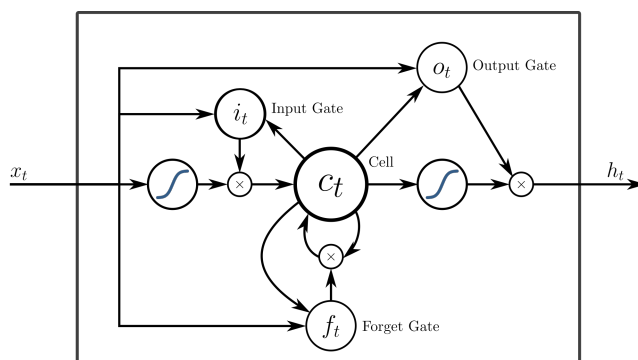
**LSTM** – ilga trumpalaikė atmintis (*angl. long short-term memory*) - RNN architektūra, kuri sugeba atsiminti informaciją ilgesniam laiko tarpui.

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

### Paprastas tinklas

Sudėtis:

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

### Gilus tinklas

Sudėtis:

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

### Platus tinklas

Sudėtis:

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

### Platesnis tinklas

Sudėtis:

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

### 2.5.2.2. GRU

## 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, video srautą, bus gaunama konkrečios klasės išeiga.

Galima pastebėti, kad būtent šiuo atveju, modelis, kuris apdoro video srautą ir nuspręs, kurios klasės įeiga buvo įeitimui 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 klase.

### 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 naudojami *categorical crossentropy* praradimo apskaičiavimo metodu, *softmax* aktyvacijos būdu ir *adam* optimizatoriumi.

### 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.

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

1. Vaizdo įrašą skaidyti į 120 kadrų. Jei įrašas per trumpas - paskutinį kadrą kartoti  $n$  kartų, kol kadrų bus 120;
2. Konvoliucinį neuroninį tinklą apmokyti Inception v3 modeliu (žr. 2 priede);
3. RNN apmokyti pasirinkus paprastą modelį;
4. Mokymo - testavimo aibę skaidyti į 80% mokymui ir 20% testavimui.

### 3.3.2.1. Pirmasis apmokymas

#### Duomenys:

- 3 klasės;
- 45 vaizdo įrašai kiekvienai klasei. Viso - 120.

#### Mokymas:

- KNN: skaidymas;
- RNN: 108 mokymo ir 12 patikrinimo vaizdo įrašų.

DIAGRAMOS!!!!

#### Rezultatai:

- KNN: ....;
- RNN: Tikslumas 86.93% ir 0.5081 praradimas.

### 3.3.2.2. Rezultatai

8 lentelė. Lietuvių gestų kalbos apmokymų rezultatai

| Bandymo Nr. | Klasių skaičius | Vaizdo įrašų skaičius | KNN       | RNN             |           |            |
|-------------|-----------------|-----------------------|-----------|-----------------|-----------|------------|
|             |                 |                       | Tikslumas | Epochų skaičius | Tikslumas | Praradimas |
| 1.          | 3               | 120                   | ...       | 100             | 86.93%    | 0.5081     |

### 3.3.3. Modelio testavimas

- 3 klasės 120 video
- 92.31% tikslumas iš nematytų 30 video.

## **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  $t$
- $h_t$  – būseną laiko momentu  $t$



## Sąvokų apibrėžimai

- Dirbtiniai neuroniniai tinklai - artificial neural networks
- Inception v3 - Google modelis
- Išėja - output
- Įėja - 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

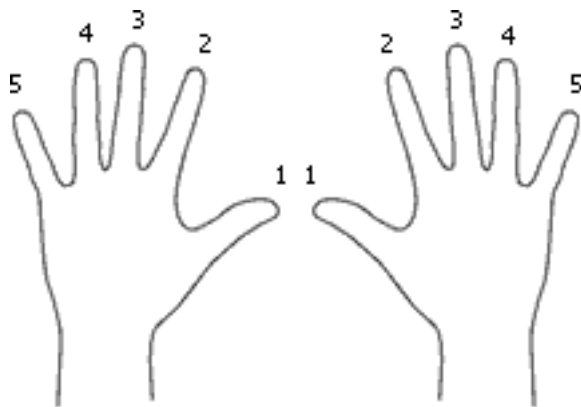
## Santrumpos

- 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**

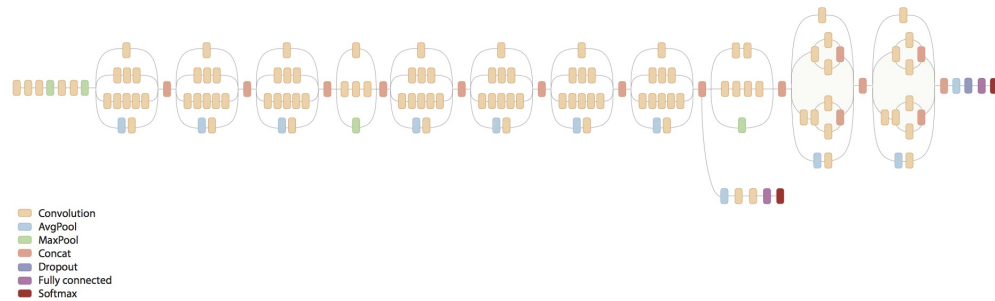
### **Rankų pirštų numeracija**



11 pav. Kairės ir dešinės rankų pirštų numeracija

## Priedas 2

### Konvoliucinio tinklo modelis



12 pav. Konvoliucinio tinklo modelis „Inception v3“