

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMATIKOS KATEDRA

## **3D objektų atpažinimas iš 2D nuotraukų**

### **3D object recognition from 2D images**

Magistro baigiamasis darbas

Atliko:	Aleksas Vaitulevičius	(parašas)
Darbo vadovas:	prof. habil. dr. Olga Kurasova	(parašas)
Recenzentas:	doc. dr. Vardauskas Pavardauskas	(parašas)

Vilnius – 2020

## 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:** Klasifikavimo uždavinys, 3D objektai, dirbtiniai neuroniniai tinklai, kapsuliniai neuroniniai tinklai, tiesioginio sklidimo neuroniniai tinklai

## Summary

Santrauka anglų kalba.

**Keywords:** classification, 3D objects, artificial neural networks, CapsNet neural networks, convolutional neural networks

## Turinys

Ivadas .....	4
1. Literatūros analizė .....	7
1.1. 3D objektų atpažinimo iš 2D nuotraukų uždavinys .....	7
1.2. 3D objektų atpažinimo iš 2D nuotraukų uždavinio sprendinių pavyzdžiai.....	7
1.3. Dirbtinių neuroninių tinklų bendrieji principai .....	8
1.3.1. Dirbtinis neuronas, perceptronas .....	8
1.3.2. Dirbtiniai neuroniniai tinklai .....	10
1.3.3. Gilieji neuroniniai tinklai .....	13
1.4. Daugiavaizdžių konvoliucinių dirbtinių neuroninių tinklų apžvalga.....	15
1.4.1. Konvoliucija .....	15
1.4.2. Konvoliucinio neuroninio tinklo sluoksnių tipai.....	16
1.4.3. Daugiavaizdis konvoliucinis neuroninis tinklas.....	19
1.5. Kapsulinių dirbtinių neuroninių tinklų apžvalga .....	19
2. Kapsulinių neuroninių tinklų modifikacijos ir parametrai .....	21
3. Eksperimentiniai tyrimai .....	22
3.1. Tyrimams naudoti duomenys .....	22
3.2. Kapsulinių neuroninių tinklų modifikacijų ir parametrų eksperimentiniai tyrimai .....	22
3.3. Kapsulinių ir daugiavaizdžių konvoliucinių neuroninių tinklų eksperimentiniai tyrimai..	22
Rezultatai ir išvados .....	23
Literatūra .....	24
Santrumpos .....	26
Priedas Nr.1	
Priedas Nr.2	

## Ivadas

Vienas iš fundamentalių kompiuterinės regos uždavinių yra informacijos apie 3 dimensijų (3D) pasaulį išgavimas naudojant 2 dimensijų (2D) nuotraukas. Šio uždavinio tikslas yra atpažinti konkrečius 3D objektus, naudojant jų, 2D nuotraukas, padarytas iš skirtingų kampų. Šiam tikslui pasiekti, yra konstruojami objektų atpažinimo algoritmai, kurie klasifikuoja 2D nuotraukas į klases, kurios atstovauja vieną iš 3D objektų modelių.

3D objektų atpažinimas iš 2D nuotraukų yra naudojamas srityse, kuriose turimi 3D objektai turi būti atpažinti iš visų galimų 2D nuotraukų, turint tik poaibį šių nuotraukų. Keli šių sričių pavyzdžiai yra automatinė objektų inspekcija - turint algoritmą, atpažįstantį konkretų objektą, kuris turi tik jam būdingą 3D formą, galima nustatyti nuotraukas, kuriose yra tas objektas, navigacijoje - turint algoritmą, atpažįstantį konkrečius objektus, esančius skirtingose vietovėse, ir tų vietovių koordinates, galima nustatyti kurioje vietovėje buvo padaryta nuotrauka. Deja, laiko ir duomenų kaštai yra per dideli, kad pasiekti absoliutų tikslumą sprendžiant šį uždavinį. Todėl taikomi metodai yra euristiniai. Dėl to renkantis metodą, spręsti 3D objektų atpažinimo iš 2D nuotraukų uždaviniui, reikia atsižvelgti į laiko kaštus ir kaip tiksliai tuo metodu pagrįstas algoritmas klasifikuoja 2D nuotraukas, sprendžiamas šį uždavinį. Šiame darbe bus atliekami tyrimai, skirti nustatyti metodą, sprendžiantį 3D objektų atpažinimo iš 2D nuotraukų uždavinį ir kuris pasiekia didžiausią tikslumą ir reikalauja mažiausiai laiko mokymui.

Gana dažnai naudojamas metodas šiam uždaviniui spręsti, yra dirbtiniai gilieji neuroniniai tinklai. 3D objektų atpažinimo iš 2D nuotraukų uždavinyje naudojami duomenys yra 2D nuotraukos, yra nestruktūrizuoti, jiems sudėtinga vykdyti požymių išgavimą. Todėl daugelis kitų sprendimų nėra tokie patrauklūs kaip dirbtiniai gilieji neuroniniai tinklai, dėl savo sugebėjimo efektyviai vykdyti automatinį požymių išgavimą iš nestruktūrizuotų duomenų. Tačiau, norint pasiekti aukštą klasifikavimo tikslumą, naudojant šį metodą, yra reikalingas didelis kiekis duomenų. Konkrečiai šiam uždaviniui reikia didelio kiekio 3D modelių. Laimei, šiuo metu egzistuoja viešai prieinamos didelės 3D repozitorijos. Tokios kaip 3D Warehouse, TurboSquid, ir Shapeways. Dėl to šiuo metu daugelis senesnių architektūrų jau yra išbandytos sprendžiant 3D objektų atpažinimo iš 2D nuotraukų uždavinį. Pavyzdžiui viena iš architektūrų, kuri buvo išbandyta, yra konvoliucinio gilaus pasitikėjimo neuroninio tinklo (angl. convolutional deep belief neural network) architektūra. Šiai architektūrai atlikti tyrimai yra aprašyti darbe [WSK<sup>+</sup>15]. Tačiau 2017 metais buvo aprašyta nauja architektūra, kapsuliniai neuroniniai tinklai. Tyrimai, parodė, kad ji yra pranašesnė tikslumo atžvilgiu už senesnes architektūras, sprendžiant uždavinius panašius į 3D objektų atpažinimo iš 2D nuotraukų uždavinį.

Šiuo metu šiam uždaviniui spręsti, optimaliausius rezultatus, laiko ir tikslumo atžvilgiu, pasiekusi dirbtinio neuroninio tinklo architektūra yra konvoliucinių neuroninių tinklų tipas - daugiavaizdžiai konvoliuciniai neuroniniai tinklai (angl. multi-view convolutional neural networks). Tyrimai, kuriuose ši architektūra buvo išbandyta, yra aprašyta darbuose [Mas18; SMK<sup>+</sup>15]. Darbe [WSK<sup>+</sup>15] atlikto tyrimo rezultatai parodo, kad pateiktas sprendimas, kuriame 3D objektų atpažinimas yra konstruojamas naudojantis tik 2D nuotraukomis, yra tikslesnis 8 %. Algoritmas, naudojantis 3D modelius, pasiekė 77 % tikslumą, o algoritmas, naudojantis tik 2D nuotraukas, pasiekė 85

% tikslumą. Todėl šiame darbe bus daromi eksperimentai su dirbtinio neuroninio tinklo architektūrų įgyvendinimais, kurie yra pagrįsti būtent šiuo metodu. Tad šiam darbui, vienas iš pasirinktų metodų yra daugiavaizdis konvoliucinis neuroninis tinklas, kurio įgyvendinimas ir tyrimai yra aprašyti darbe [SMK<sup>+</sup>15]. Mat šio darbo įgyvendinimas naudoja tik 2D nuotraukas, konstruojant 3D objektų atpažinimo algoritmą.

Kita tiriamą dirbtinio neuroninio tinklo architektūra yra kapsuliniai neuroniniai tinklai. Lyginant su konvoliuciniais neuroniniais tinklais, tai gana nauja architektūra. Aprašytos 2017 metais [SFH17] darbe Kapsulinių neuroninių tinklų architektūros veikimo principas tiksliau imituoja žmogaus regą, remiantis faktu, kad žmogaus rega ignoruoja nereikšmingas vaizdo detales, naudodama tik seką fokusuotų taškų, taip apdorojama tik dalį vaizdinės informacijos su labai aukšta rezoliucija. [SFH17] darbe atliktas tyrimas parodo, kad ši architektūra atlieka ranka rašytų skaičių klasifikavimo užduotį tiksliau nei konvoliuciniai neuroniniai tinklai. Kitas tyrimas, kuris yra atliktas darbe [MC17] su 4 duomenų rinkiniais, kuriuose yra veidai, kelio ženklai ir kasdieniai objektai, parodo, kad dabartiniai kapsuliniai neuroniniai tinklai ne visada yra pranašesni už konvoliucinius neuroninius tinklus. Parinkus geresnius parametrus ir modifikacijas (sluoksnių skaičių, neuronų skaičių kiekviename sluoksnyje, aktyvacijos funkcijos), konvoliucinis neuroninis tinklas dar vis būna pranašesnis už kapsulinį neuroninį tinklą. Tačiau taip pat darbe [MC17] yra teigiama, kad kapsuliniai neuroniniai tinklai dar nėra pasiekę pilno savo potencialo ir tolimesni tyrimai turi būti atlikti.

Tad šio darbo **tikslas** yra įrodyti arba paneigti keliamą **hipotezę**:

*Kapsuliniai neuroniniai tinklai sprendžia 3D objektų atpažinimo iš 2D nuotraukų uždavinį efektyviau nei konvoliuciniai neuroniniai tinklai remiantis apmokymo laiko ir tikslumo kriterijais.*

Tikimasi, kad, sprendžiant 3D objektų atpažinimo iš 2D nuotraukų uždavinį, kapsulinio neuroninio tinklo mokymas truks trumpiau nei konvoliucinio neuroninio tinklo. Taip pat, kad apmokytas kapsulinis neuroninis tinklas vykdys klasifikavimą tiksliau nei daugiavaizdis konvoliucinis neuroninis tinklas.

Siekiant patikrinti iškeltą hipotezę reikia atlikti šiuos uždavinius:

1. Išanalizuoti ir nustatyti dabartinį efektyviausią 3D objektų atpažinimo iš 2D nuotraukų uždavinio sprendinį.
2. Išanalizuoti kapsulinių neuroninių tinklų veikimą.
3. Surasti duomenis, skirtus spręsti 3D objektų atpažinimo iš 2D nuotraukų uždaviniui.
4. Eksperimentiškai nustatyti efektyviausius parametrus ir modifikacijas, skirtus spręsti 3D objektų atpažinimo iš 2D nuotraukų uždaviniui, kapsulinio neuroninio tinklo implementacijai, remiantis apmokymo laiko ir tikslumo kriterijais.
5. Atlikti eksperimentus, skirtus palyginti kapsulinio neuroninio tinklo ir daugiavaizdžio konvoliucinio neuroninio tinklo tikslumą ir apmokymo laiką, sprendžiant 3D objektų atpažinimo iš 2D nuotraukų uždavinį.

Šiame darbe planuojami arba jau pasiekti rezultatai:

1. Nustatyta, kad šiuo metu efektyviausias 3D objektų atpažinimo iš 2D nuotraukų uždavinio sprendinys yra daugiavaizdžiai konvoliuciniai neuroniniai tinklai, lyginant eksperimentų, aprašytų skirtinguose literatūros šaltiniuose, rezultatus. Šiuose šaltiniuose buvo surasta daugiavaizdžio konvoliucinio neuroninio tinklo implementaciją ir duomenys skirti apmokymui ir testavimui.
2. Išanalizuotas kapsulinių neuroninių tinklų veikimas, surasta jo implementacija.
3. Planuojama eksperimentiškai nustatyti efektyviausią kapsulinio neuroninio tinklo konfigūraciją, skirtą spręsti 3D objektų atpažinimo iš 2D nuotraukų uždavinį, naudojantis apmokymo laiko ir tikslumo kriterijais.
4. Planuojama eksperimentiškai palyginti kapsulinio neuroninio tinklo tikslumą ir apmokymo laiką su konvoliuciniu neuroniniu tinklu, naudojantis apmokymo laiko ir tikslumo kriterijais.

Darbas remiasi šiomis prielaidomis:

1. Kiekvienam 2D paveikslėliui yra priskirta jam jį atitinkantis 3D objektas.
2. Kiekvienas 3D objektas turi bent po vieną jį atitinkantį 2D paveikslėlį.

Šio darbo turinys yra sudarytas iš 4 skyrių. Pirmame skyriuje yra pateikiama literatūros analizė. Jame yra pateiktas 3D objektų atpažinimo iš 2D nuotraukų uždavinio aprašymas, egzistuojančių sprendimų apžvalga, bendrieji neuroninių tinklų principai, daugiavaizdžio konvoliucinio neuroninio tinklo veikimo aprašymas ir kapsulinio neuroninio tinklo aprašymas. Tada antrame skyriuje yra pateikiami šiame darbe bandomų kapsulinių neuroninių tinklų modifikacijos ir parinkti parametrai. Trečiame skyriuje yra aprašomi tyrimams naudoti duomenys. Taip pat šiame skyriuje yra aprašomi tyrimai, skirti nustatyti kapsulinių neuroninių tinklų modifikaciją ir parametrus, kurie pasiekia didžiausią tikslumą ir reikalauja mažiausiai laiko apmokymui, sprendžiant 3D objektų atpažinimo iš 2D nuotraukų uždavinį. Galiausiai šiame skyriuje yra aprašomi tyrimai, skirti palyginti kapsulinių neuroninių tinklų ir daugiavaizdžių konvoliucinių neuroninių tinklų tikslumą ir apmokymo laiką, sprendžiant 3D objektų atpažinimo iš 2D nuotraukų uždavinį. Paskutiniame skyriuje yra pateikiami rezultatai ir išvados.

# 1. Literatūros analizė

## 1.1. 3D objektų atpažinimo iš 2D nuotraukų uždavinys

3D objektų atpažinimo iš 2D nuotraukų uždavinys - tai klasifikavimo uždavinys, kuriame pateiktos 2D nuotraukos, kuriose yra atvaizduotas 3D objektas iš atsitiktinio apžvalgos taško, turi būti priskirtas 3D modeliui, kuris yra atvaizduotas toje 2D nuotraukoje.

Klasifikavimo uždavinys - tai uždavinys, kuriame kuriamas metodas, kaip nustatyti pavyzdžio, iš tiriamos srities populiacijos, klasę. 3D objektų atpažinimo iš 2D nuotraukų uždavinio atveju, tiriamą sritį yra 2D nuotraukos, kuriose yra atvaizduotas 3D objektas iš bet kurio apžvalgos taško ir klasė - 3D objektas. Taip pat, šio darbo atveju, metodas yra dirbtinio neuroninio tinklo (kapsulinio arba konvoliucinio) apmokytas modelis.

Kaip jau minėta įvade, šiam uždaviniui spręsti efektyviausia yra naudoti mašininio mokymo metodą, kurio mokymo duomenys yra tik 2D nuotraukos, o 3D objektai bus tik duomenų klasės. Darbe [WSK<sup>+</sup>15] atlikto tyrimo rezultatai parodo, kad pateiktas sprendimas, kuriame 3D objektų atpažinimas yra konstruojamas naudojantis tik 2D nuotraukomis, yra tikslesnis 8 %. Algoritmas, naudojantis 3D modelius, pasiekė 77 % tikslumą, o algoritmas, naudojantis tik 2D nuotraukas, pasiekė 85 % tikslumą. Šaltinyje [SMK<sup>+</sup>15] yra teigiama, kad to priežastis yra reliatyviai efektyvesnis 2D nuotraukų informacijos saugojimas negu 3D modelių. Todėl, kad, nors 3D modelis turi visą informaciją apie atvaizduotą 3D objektą, tačiau tam, kad panaudoti vokselinę 3D objekto reprezentaciją mašiniame mokyme, kurio mokymas su pakankamai didele duomenų imtimi užtruktų racionalų laiko tarpą, tenka ženkliai sumažinti 3D modelio rezoliuciją. Pavyzdžiui, 3D modelio, kurio rezoliucija yra  $30 \times 30 \times 30$  vokseliai, įvesties dydis yra apytiksliai lygus 2D paveikslėlio, kurio rezoliucija yra  $164 \times 164$  pikseliai. Tad šiuo atveju, 3D modelis yra apdorojamas per tiek pat laiko kaip ir 2D paveikslėlis, bet modelio rezoliucija yra apytiksliai 5.5 karto mažesnė. Todėl mašininio mokymo metodas, kurio mokymo duomenys yra 3D modelis, gauna mažesnės raiškos įvestį, negu metodas, kurio mokymo duomenys yra 2D paveikslėliai.

## 1.2. 3D objektų atpažinimo iš 2D nuotraukų uždavinio sprendinių pavyzdžiai

Vienas seniausių šio uždavinio sprendinių, taikantis tokią metodologiją, yra aprašytas darbe [MN95]. Šis sprendinys atpažįsta 3D objektus lygindamas jų vaizdus, kurie buvo suformuoti iš didelės imties 2D nuotraukų, parametrizuotoje eigenerdvėje (angl. eigenspace). Šios nuotraukos buvo sugeneruotos iš 3D modelių naudojant skirtingus apžvalgos taškus ir apšvietimus. Kitas pavyzdys, kuris yra gana populiarus kompiuterinėje grafikoje, yra šviesos lauko deskriptorius (angl. light field descriptor), kuris yra aprašytas darbe [CTS<sup>+</sup>03]. Šis sprendinys išgauna geometrinius ir Fourier'io deskriptorius iš 3D objektų siluėtų, kurie buvo sugeneruoti iš 3D modelių, naudojant skirtingus apžvalgos taškus. Darbe [MSD<sup>+</sup>02] aprašytas šio uždavinio sprendimas, kuris 3D objekto siluetus išskaido į dalis ir išsaugo juos į orientuotą beciklį grafą (angl. directed acyclic graph), šoko grafą. Kitas pavyzdys aprašytas darbe [CK04], naudoja panašumo metrikas (angl. simi-



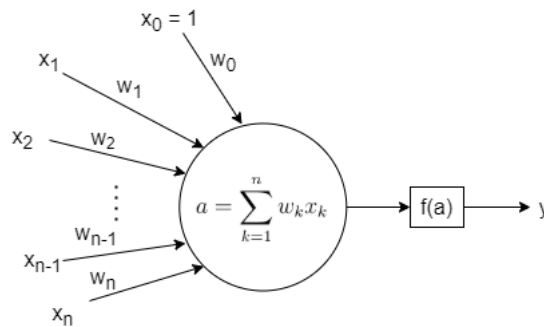
larity metrics) , kurios yra pagrįstos kreivių palyginimu (angl. curve matching) ir sugrupuotomis panašiomis 2D nuotraukomis.

Šiuo metu, 3D objektų atpažinimo iš 2D nuotraukų uždaviniui spręsti, optimaliausius rezultatus, laiko ir tikslumo atžvilgiu, pasiekęs mašininio mokymo metodu pagrįstas sprendimas yra konvoliuciniai dirbtiniai neuroniniai tinklai. Tai eksperimentu buvo įrodyta darbe [SMK<sup>+</sup>15]. Šiame eksperimente buvo palyginti įvairūs konvoliucinių neuroninių tinklų tipai sprendžiant šį uždavinį ir geriausią rezultatą pasiekęs tipas buvo daugiavaizdis (angl. multi-view convolutional network) konvoliucinis neuroninis tinklas, kurio tikslumas buvo 90.1%.

### 1.3. Dirbtinių neuroninių tinklų bendrieji principai

#### 1.3.1. Dirbtinis neuronas, perceptronas

Šiame darbe nagrinėjami dirbtiniai neuroniniai tinklai yra sudaryti iš Rosenblato darbe [Ros57] aprašytų dirbtinių neuronų, perceptronų. Perceptronas - tai iteratyviai apmokomas tiesinis klasifikatorius, kuris susideda iš  $x = \{x_0, x_1, x_2, \dots, x_p\}$  mokymo aibės, kurios dydis yra  $n$ , vektorių, vadinamais įėjimais,  $\{w_0, w_1, w_2, \dots, w_p\} \in \mathbb{R}$  perdavimo koeficientų, vadinamų svoriais, aktyvacijos (perdavimo) funkcijos  $f(a)$  ir  $\{y_0, y_1, y_2, \dots, y_n\}$  reikšmių, vadinamų išėjimais. Įėjimas  $x_0$  yra vadinamas nuliniu įėjimu ir jo reikšmė yra pastovi  $x_0 = 1$ , o  $w_0$  - nuliniu svoriu arba slenksčiu (angl. bias). Perceptronas yra atvaizduotas paveikslėlyje 1



1 pav. Perceptronas

Formulė 1 yra aktyvacijos funkcijos argumentas.

$$a = \sum_{k=1}^p w_k x_k \quad (1)$$

Dažniausiai perceptronui yra naudojamos šios aktyvacijos funkcijos: slenkstinė (angl. unit step) 2, sigmoidinė (angl. sigmoid) 3, gabalais tiesinė (angl. piecewise linear) 4, Gauso (angl. Gaussian) 5 ir tiesinė (angl. linear) 6

$$f(a) = \begin{cases} 0, & \text{jei } \beta > a \\ 1, & \text{jei } \beta \leq a \end{cases} \quad (2)$$

$$f(a) = \frac{1}{1 + \exp^{-\beta}} \quad (3)$$

$$f(a) = \begin{cases} 0, & \text{jei } a_{min} \geq \\ ma + b, & \text{jei } a_{min} < a < a_{max} \\ 1, & \text{jei } a_{max} \leq a \end{cases} \quad (4)$$

$$f(a) = \frac{1}{\sqrt{2\pi}\sigma} \exp \frac{-(a - \mu)^2}{2\sigma^2} \quad (5)$$

$$f(a) = ma + b \quad (6)$$

Perceptronas yra skirtas spręsti klasifikavimo uždavinius. Tam kad perceptronas spręstų konkretų klasifikavimo uždavinį, jis turi būti apmokytas. Perceptrono apmokymas yra iteratyvus procesas, kuriame randami svoriai  $W = \{w_0, w_1, w_2, \dots, w_n\}$ , su kuriais funkcijos 7 rezultatas įgyja apytiksliai mažiausią reikšmę. Funkcijoje 7  $y_i$  yra perceptrono i-tasis išėjimas ir  $t_i$  - i-tojo įėjimo norima klasė.

$$e(w) = \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2 \quad (7)$$

Apmokymo pradžioje pradiniai svoriai yra parenkami atsitiktinai. Toliau gradientinio nusileidimo algoritmu judant antigradiento kryptimi, svorių reikšmės perskaičiuojamos naudojantis funkcija 8, kur  $\Delta w_k(t)$  yra funkcija 9,  $t$  - iteracijos numeris,  $\eta \in [0, +\infty]$  - parinktas mokymo greitis (angl. learning rate), ir vieną įėjimo vektorių iš duomenų aibės. Svoriai yra perskaičiuojami norima skaičių kartų.

$$w_k(t+1) = w_k(t) + \Delta w_k(t) \quad (8)$$

$$\Delta w_k(t) = -\eta \frac{\partial e(w)}{\partial w_k} \quad (9)$$

Pakeitus, kai kurių kintamųjų žymėjimą, iš lygties 1 gaunama lygtis 10, kur  $a_i$  yra i-tojo įėjimo vektoriaus aktyvacijos funkcijos argumentas,  $x_{ik}$  yra i-tojo įėjimo vektoriaus k-atoji komponentė.

$$a_i = \sum_{k=1}^p w_k x_{ik} \quad (10)$$

Tad i-tasis perceptrono išėjimas  $y_i$  yra  $y_i = f(a_i)$ . Tada funkcijos 7 išvestinė yra lygtis 11.

$$\frac{\partial e(w)}{\partial w_k} = \left( \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2 \right)' = \frac{2}{n} \sum_{i=1}^n ((y_i - t_i)(f'(a_i)) \left( \sum_{k=1}^p x_{ik} \right)) \quad (11)$$

Tada bendru atveju perceptrono mokymo taisyklė 8 yra funkcija 12.

$$w_k(t+1) = w_k(t) - \eta \frac{2}{n} \sum_{i=1}^n ((y_i - t_i)(f'(a_i)) \left( \sum_{k=1}^p x_{ik} \right)) \quad (12)$$

Naudojantis apmokytu perceptronu galima nustatyti duoto duomenų vektoriaus  $\mathbf{x}'$  klasę. Klasė nustatoma randant reikšmę  $a$  iš lygties 1, kur  $\{w_0, w_1, w_2, \dots, w_p\}$  yra apmokyto perceptrono svoriai ir  $\{x_0, x_1, x_2, \dots, x_p\} = \mathbf{x}'$ .  $\mathbf{x}'$  vektoriaus klasė atitiks intervalo  $b_c$ , į kurią patenka reikšmė  $a$ , klasė  $c$ . Klasių intervalai  $b_c \in \mathbb{R}$  yra paskaičiuojami, padalinant  $\mathbb{R}$  su skiriamuoju paviršiumi (angl. decision boundary), o skiriamasis paviršius yra nustatomas pagal aktyvacijos funkciją.

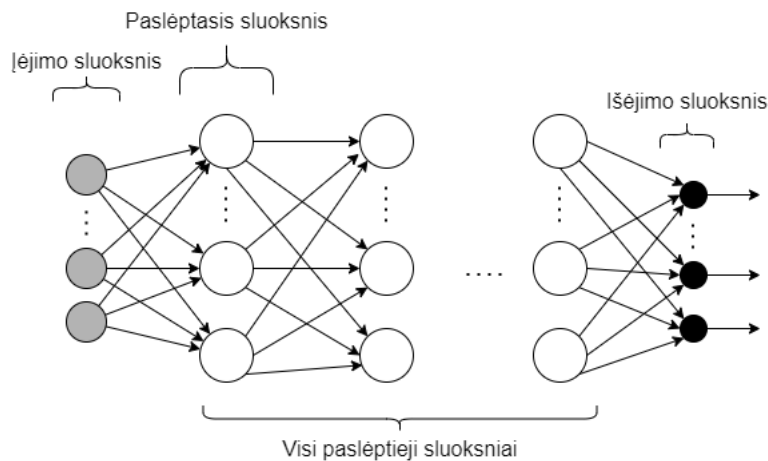
### 1.3.2. Dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai (angl. artificial neural networks) yra jungus tinklas, kurio viršūnės yra perceptronai, ir kurio orientuotos briaunos sujungia 2 perceptronus, iš kurių vieno perceptrono išėjimas yra naudojamas kaip kito perceptrono įėjimas. Dirbtinio neuroninio tinklo paskirtis yra spręsti klasifikavimo uždavinius.

Pagal tinklo struktūrą dirbtiniai neuroniniai tinklai yra skirstomi į tiesioginio sklidimo (angl. feedforward) ir grįžtamojo ryšio (angl. feedback). Grįžtamojo ryšio dirbtiniai neuroniniai tinklai turi bent vieną ciklą, o tiesioginio sklidimo neturi nei vieno ciklo. Tiesioginio sklidimo neuroniniai tinklai, dėl savo paprastumo, reikalauja trumpesnio apmokymo laiko nei grįžtamojo ryšio neuroniniai tinklai. Todėl jie yra dažniausiai naudojami praktiniuose taikymuose.

Tiesioginio sklidimo tinklai yra grupuojami į vienasluoksnius perceptronus, daugiasluoksnius perceptronus ir radialinių funkcijų tinklus. Šiame darbe tiriama dirbtiniai neuroniniai tinklai, konvoliuciniai ir kapsuliniai neuroniniai tinklai, yra daugiasluoksnių perceptronų plėtiniai. Daugiasluoksnius perceptronus (angl. multilayer perceptron) yra dirbtinis neuroninis tinklas, kurio perceptronai yra sugrupuoti į sluoksnius, kuriuose gali būti skirtingas skaičius perceptronų. Kiekviename sluoksnyje esantys perceptronai turi tą pačią aktyvavimo funkciją.

Daugiasluoksnius perceptrono sluoksniai yra išsidėstę eilėje ir kiekviename sluoksnyje esančių visų perceptronų išėjimai yra tolimesnio sluoksniu visų perceptronų įėjimai. Pirmasis sluoksnius yra vadinamas įėjimo sluoksniu ir jis susideda ne iš perceptronų, o iš mokymo duomenų vektoriaus  $\mathbf{x}$  komponentų. Paskutinis sluoksnius yra išėjimo sluoksnius ir jame yra tiek perceptronų kiek yra nagrinėjamų klasių, kiekvienai klasei yra priskiriamas ją atitinkantis perceptronas. Išėjimo sluoksniu perceptronų išėjimų reikšmės priklauso nuo juose naudojamos aktyvacijos funkcijos. Likę sluoksniai yra vadinami paslėptaisiais sluoksniais. Daugiasluoksnius perceptronas yra pavaizduotas paveikslėlyje 2.



2 pav. Daugiasluoksnis perceptronas

Daugiasluoksnis perceptronas pateiktam vektoriui  $\mathbf{x}'$  priskiria klasę, kuri buvo priskirta išėjimo sluoksnio perceptronui, kurio išėjimo reikšmė buvo didžiausia iš visų išėjimo sluoksnio perceptronų.

Tam kad daugiasluoksnis perceptronas galėtų atlikti klasifikavimą, jis turi būti apmokytas. Kaip ir perceptronas, daugiasluoksnis perceptronas yra apmokomas iteratyviai keičiant visų perceptronų svorius naudojant formulę 8. Išėjimo sluoksnio perceptronams yra naudojama bendra formulė 12. Tačiau ši funkcija, paslėptųjų sluoksnių perceptronų mokymui, yra nenaudojama, nes nėra apibrėžta nuostolių funkcija  $e(w)$ , mat funkcijai  $e(w)$  yra reikalingi perceptronų išėjimai, kurie yra nežinomi. Tad paslėptųjų perceptronų svoriai yra keičiami naudojant klaidos sklaidimo atgal algoritmą (angl. back-propagation learning algorithm).

Klaidos sklaidimo atgal algoritmas yra gradientinio nusileidimo strategijos realizacija daugiasluoksniui perceptronui. Šio algoritmo veikimo santrauka: randamas i-tosios iteracijos išėjimo vektorius  $Y_i = (y_{i1}, y_{i2}, \dots, y_{id})$ , apskaičiuojama i-tosios iteracijos išėjimo sluoksnio perceptronų paklaidos  $e_i(w)$ , apskaičiuojami paslėptųjų sluoksnių perceptronų paklaidos, kiekvienam sluoksniui naudojant jo tolimesnių sluoksnių perceptronų paklaidas, ir pakeičiant svorius naudojantis gautomis paklaidomis.

Išėjimo perceptronų paklaidos randamos naudojant formulę 7. Tad bendru atveju išėjimo sluoksnių svoriai yra keičiami naudojant formulę 12. Įvedamas lokalaus gradiento žymėjimas 13, kur  $y_{ij}$  yra gauta išėjimo reikšmė,  $t_{ij}$  - norima išėjimo reikšmė su i-tuoju įėjimo vektoriumi j-tajam perceptronui,  $a_{ij}$  apskaičiuojamas pagal lygtį 1 naudojant j-tojo perceptrono svorius ir i-tąjį įėjimo vektorių ir  $f$  - aktyvacijos funkcija. Įsistačius šį žymėjimą į formulę 12 gaunama funkcija 14.

$$\xi_i^j = (y_{ij} - t_{ij})(f'(a_{ij})) \quad (13)$$

$$w_j(t+1) = w_j(t) - \eta \frac{2}{n} \sum_{i=1}^n \xi_i^j \left( \sum_{k=1}^p x_{ik} \right) \quad (14)$$

Remiantis darbu [Hay04], lokalus gradientas  $\xi^j$  yra apskaičiuojamas pagal formulę 15, kur  $P_j$  yra aibė perceptronų, prijungtų prie j-tojo perceptrono išėjimo,  $w_{bj}$  - b-tojo ir j-tojo perceptronų

jungties svoris.

$$\xi_i^j = f'(a_{ij}) \sum_{b \in P_j} \xi_i^b w_{bj} \quad (15)$$

Tad bendru atveju daugiasluoksnio perceptrono svoriai yra keičiami naudojantis formule 16.

$$w_j(t+1) = w_j(t) - \eta \frac{2}{n} \sum_{i=1}^n (f'(a_{ij}) (\sum_{b \in P_j} \xi_i^b w_{bj}) (\sum_{k=1}^p x_{ik})) \quad (16)$$

Egzistuoja įvairūs metodai naudojami pradinių svorių parinkimui. Tačiau analizuoto šaltinio [Rou06] tyrimai neįrodė, kad vienas iš jų būtų pranašesnis už kitus. Todėl šio darbo tyrimams pradiniai svoriai yra parenkami atsitiktinai.

Procesas, kurio metu yra įvykdomas klaidos sklidimo atgal algoritmas su kiekvienu įėjimo vektoriumi iš mokymo duomenų aibės po vieną kartą, yra vadinamas epocha. Daugiasluoksnio perceptrono apmokymas gali būti vykdomas nurodytą kiekį epochų arba perceptrono apmokymo epochos gali būti vykdomos tol kol nepasiekiamas norimas tikslumas naudojantis validacijos duomenimis. Šiame darbe dirbtinių neuroninių tinklų palyginimui yra naudojami tik apmokymo laiko ir tikslumo kriterijai. Tad tyrime vykdomų epochų skaičiai yra konstantos. Naudojant tą patį epochų skaičių dviems skirtingiems dirbtiniams neuroniniams tinklams, dėl skirtingos tinklų struktūros, apmokymo epochos laikas skiriasi. Tad nurodant tą patį kiekį epochų skirtinguose dirbtinių neuroninių tinklų apmokymuose galima palyginti jų tikslumą ir apmokymo laiką.

Su kiekviena epocha dirbtinis neuroninis tinklas vis tiksliau klasifikuoja apmokymo duomenų vektorius. Tačiau apmokymo duomenys tėra tik poaibis realaus pasaulio duomenų, vadinamų populiacija. Apmokymo duomenys niekada nėra pilnai reprezentatyvūs populiacijai. Todėl po tam tikro skaičiaus epochų, dirbtinio neuroninio tinklo tikslumas priskiriant populiacijos vektorius, kurie nepriklauso apmokymo duomenims, ima mažėti. Ši problema yra vadinama persimokymu (angl. overfitting). Tad dirbtinio neuroninio tinklo tikslumo, kuris būtų reprezentatyvus populiacijai, nustatymui, yra taikomi įvairūs metodai. Du populiariausi šių metodų pavyzdžiai yra n-perlenkimų kryžminis validavimas (angl. n-folds cross validation) ir validavimas su testiniais duomenimis. Šiame darbe naudojamas metodas yra validavimas su testiniais duomenimis, nes šiuo metodu apskaičiuotų dviejų dirbtinių neuroninių tinklų tikslumų palyginimas yra lengviau interpretuojamas.

Apmokytų Dirbtinių neuroninių tinklų tikslumo nustatymui, naudojant validavimą su testiniais duomenimis, turimų duomenų aibė yra padalinama į du poaibius: apmokymo duomenis ir testavimo duomenis. Apmokymo duomenys yra naudojami tik apmokymo procesui. Tuo metu testavimo duomenys yra naudojami apskaičiuoti tikslumui. Šių duomenų vektoriams apmokyti dirbtiniai neuroniniai tinklai priskiria po klasę, kuri yra palyginama su tikrąja vektoriaus klase. Tada apskaičiuojamas tikslumas, naudojantis formule  $acc = \frac{m}{n}$ , kur  $m$  yra skaičius vektorių, kuriems priskirta klasė sutapo su jų tikrąja klase ir  $n$  - testavimo duomenų vektorių kiekis.

### 1.3.3. Gilieji neuroniniai tinklai

Kaip jau minėta praeitame poskyryje tiek konvoliuciniai tiek kapsuliniai neuroniniai tinklai yra daugiasluoksnio perceptrono plėtiniai. Abu šie plėtiniai priklauso daugiasluoksnio perceptrono plėtinių klasei, giliesiems neuroniniams tinklams (angl. deep neural networks). Pirmasis giliojo dirbtinio neuroninio tinklo aprašymas yra pateiktas darbe [IL65]. Gilusis neuroninis tinklas tai daugiasluoksnis perceptronas, turintis daugiau nei vieną paslėptąjį sluoksnį. Šie neuroniniai tinklai dažniausiai būna žymiai sudėtingesni nei paprasti daugiasluoksniai perceptronai. Tad jų apmokymas trunka ilgiau, bet jų tikslumas yra ženkliai didesnis nei paprastų daugiasluoksnių perceptronų.

Šių neuroninių tinklų apmokymuose gradientinio nusileidimo strategija grįstas algoritmas, klaidos sklidimo atgal, gali būti pakeistas algoritmu, grįstu stochastinio gradientinio nusileidimo strategija. Pagrindinis skirtumas tarp algoritmų grįstų gradientinio nusileidimo strategija ir stochastinio gradientinio nusileidimo strategija yra funkcija naudojama svorių keitimui. Gradientinio nusileidimo strategija naudoja funkciją 9, o stochastinio gradientinio nusileidimo strategija grįsti algoritmai naudoja funkciją 17, kur gradientas  $\Delta e_i(w)$ , yra  $i$ -tojo įėjimo vektoriaus paklaidos išvestinė svoriams  $\Delta e_i(w) = \frac{\partial e(w)}{\partial w}$ .

$$w_k(t+1) = w_k(t) - \eta \Delta e_i(w) \quad (17)$$

Labai dažnai giliųjų neuronų apmokymuose stochastinio gradientinio nusileidimo strategija yra praplečiama inercijos (angl. momentum) metodu. Tokiu atveju svoriai keičiami pagal funkciją 18, kur  $\Delta w$  yra svorių pokytis praeitoje iteracijoje, o  $\alpha$  - teigiama konstanta intervale (0, 1), vadinama inercijos konstanta (angl. momentum constant).

$$w_k(t+1) = w_k(t) - \eta \Delta e_i(w) + \alpha \Delta w \quad (18)$$

Taip pat labai dažnai šių neuroninių tinklų vienoje iteracijoje yra naudojamas ne vienas įėjimo vektorius, bet apmokymo duomenų poaibis, vadinamas duomenų rinkiniu (angl. batch). Duomenų rinkiniai sudaromi parenkant iš apmokymo duomenų nurodytą skaičių vektorių, kurie nebuvo naudojami praeitose iteracijose  $n$ -tąjį kartą. Jei tokių vektorių nėra arba yra nepakankamai, kad sudaryti naują duomenų rinkinį su nurodytu dydžiu, tai trūkstami vektoriai yra parenkami tokie, kurie nebuvo naudoti praeitose iteracijose  $(n+1)$ -tąjį kartą.  $n$  pradedamas skaičiuoti nuo vieneto. Šiame darbe apmokymai yra vykdomi naudojant duomenų rinkinius.

Jei dirbtinio neuroninio tinklo apmokyme yra naudojami duomenų rinkiniai, kurių dydis yra didesnis negu 1, tai tada yra naudojama modifikuota stochastinio gradientinio nusileidimo strategija, kurioje gradientas yra apskaičiuojamas naudojantis ne visus apmokymo duomenis, bet tik iteracijos duomenų rinkinio. Stochastinio gradientinio nusileidimo strategija ir jos modifikacijos yra vadinamos optimizavimo algoritmais, nes šių algoritmų paskirtis yra optimizuoti nuostolių funkciją (angl. loss function, cost function arba objective function). Nuo pasirinkto algoritmo priklauso apmokymo trukmė. Šių algoritmų pavyzdžiai yra AdaGrad, prisitaikančio gradiento (angl. adaptive gradient); RMSProp, šaknies vidurkio kvadrato išskleidimo (angl. root mean square propagation); AdaDelta ir Adam, prisitaikančios inercijos apskaičiavimo (angl. adaptive moment estimation),

algoritmai. Šiame darbe yra naudojamas vienas populiariausių modifikacijų - Adam algoritmas.

Adam algoritmas yra aprašytas darbe [KB15]. Šis algoritmas apjungia algoritmų RMSProp ir AdaGrad privalumus bei darbe [KB15] yra pateikiamas empirinis įrodymas, kad Adam algoritmas veikimas yra sparčiausias iš tame darbe nagrinėtų algoritmų. Pirmas Adam algoritmo žingsnis yra apskaičiuoti gradientą  $t$ -ajai iteracijai  $g_t$ , naudojantis formulę 19, kur  $\Delta l f(g_{t-1})$  yra funkcijos  $l f$  išvestinė svoriams ir  $l f$  yra nuostolių funkcija.

$$g_t = \Delta l f(g_{t-1}) \quad (19)$$

Antras šio algoritmo žingsnis yra apskaičiuoti  $t$ -ajai iteracijai eksponentiškai mažėjančią praeitų kvadratu pakeltų gradientų vidurkį  $v_t$ , naudojantis formulę 21, ir eksponentiškai mažėjančią praeitų gradientų vidurkį, naudojantis formulę 20, kur  $\beta_1$  ir  $\beta_2$  yra nurodomos konstantos.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (20)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (21)$$

Pirmojoje iteracijoje  $t = 1$  naudojami  $m_0 = 0$  ir  $v_0 = 0$ . Darbe [KB15] pateikiama, kad parametrai  $v_t$  ir  $m_t$  yra linkę konverguoti į 0. Tad tolimesnis Adam algoritmo žingsnis apskaičiuoja koreguotas šių parametrų reikšmes  $\hat{v}_t$  ir  $\hat{m}_t$  naudodamas formules 23 ir 22.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (22)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (23)$$

Paskutinis šio algoritmo žingsnis yra svorių korekcija naudojantis formule 24, kur  $w_t$  yra  $t$ -osios iteracijos svorių aibė,  $\alpha$  nurodyta konstanta, vadinama mokymosi žingsnio dydžiu (angl. learning step size), ir  $\epsilon$  yra labai mažas skaičius, naudojamas tam kad išvengti dalybos iš 0.

$$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (24)$$

Darbe [KB15] atlikti tyrimai parodė, kad Adam algoritmas veikia sparčiausiai su parametrų reikšmėmis -  $\alpha = 0.001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  ir  $\epsilon = 10^{-8}$ . Tad šiame darbe yra naudojamos būtent šitos parametrų reikšmės.

Naudojamos nuostolių funkcijos optimizavimo algoritmuose priklauso nuo giliojo neuroninio tinklo tipo ir išėjimo sluoksnio aktyvacijos funkcijos. Pavyzdžiui, konvoliucinis neuroninis tinklas su išėjimo sluoksnio aktyvacijos funkcija SoftMax naudoja kryžminės entropijos (angl. cross-entropy) nuostolių funkciją. Giliojo neuroninio tinklo apmokymas yra svorių radimas, su kuriais nuostolių funkcija įgyja apytiksliai mažiausią reikšmę.

## 1.4. Daugiavaizdžių konvoliucinių dirbtinių neuroninių tinklų apžvalga

### 1.4.1. Konvoliucija

Konvoliuciniai neuroniniai tinklai yra vieni populiariausių giliųjų neuroninių tinklų tipas. Pirmą kartą sėkmingai įgyvendintas konvoliucinis neuroninis tinklas yra aprašytas darbe [LBD<sup>+</sup>89]. Šis tinklas yra skirtas ranka rašytiems pašto kodams atpažinti. Konvoliucinis neuroninis tinklas - tai gilusis neuroninis tinklas, kurio bent viename sluoksnyje yra naudojama konvoliucijos operacija, dar vadinama sąsuka. Konvoliucija - tai matematinė operacija, kurios operandai yra dvi funkcijos  $f$  ir  $g$ , ir kurios rezultatas yra funkcija, kuri apibūdina kaip viena funkcija keičia kitą. Ši operacija yra žymima  $f * g$  ir ji yra apibrėžiama kaip integralinės transformacijos rūšis pavaizduota formulėje 25, kur  $a$  ir  $b$  nurodo funkcijų  $f$  ir  $g$  apibrėžimo sritį.

$$(f * g)(t) = \int_a^b f(\tau)g(t - \tau)d\tau \quad (25)$$

Konvoliucijos algebros savybės yra komutatyvumas ( $f * g = g * f$ ), asociatyvumas ( $f * (g * h) = (f * g) * h$ ), distributyvumas ( $f * (g + h) = (f * g) + (f * h)$ ), vienetinis elementas  $f * \delta = \delta * f = f$  ir daugybos su skaliaru asociatyvumas ( $a(f * g) = (af) * g = f * (ag)$ , kur  $a \in \mathbb{R}$ ).

Dažniausiai konvoliuciniuose neuroniniuose tinkluose yra vykdoma konvoliucija diskrečioms funkcijoms. Konvoliucija, kurios operandai yra diskrečios funkcijos yra vadinama diskreti konvoliucija ir ji yra apibrėžiama kaip formulė 26, kur  $a$  ir  $b$  nurodo funkcijų  $f$  ir  $g$  apibrėžimo sritį.

$$(f * g)(t) = \sum_{\tau=a}^b f(\tau)g(t - \tau) \quad (26)$$

Šio darbo tyrimuose yra naudojamos 2D nuotraukos, kurios yra saugomos kaip dviejų dimensijų vaizdai, vadinamos matricomis. Diskreti konvoliucija matricoms yra atliekama naudojantis formulę 27.

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (27)$$

Konvoliuciniuose neuroniniuose tinkluose matrica  $I$  yra vadinama įvestimi, o matrica  $K$  - branduoliu arba filtru. Konvoliucija yra komutatyvi, todėl formulė 27 gali būti išreikšta kaip 28.

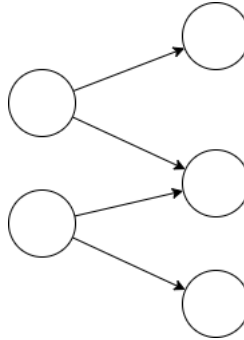
$$(K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (28)$$

Dažniausiai ši išraiška yra naudojama konvoliuciniuose neuroniniuose tinkluose. Branduolys  $K$  dažniausiai yra žymiai mažesnio dydžio nei įvesties matrica  $I$  ir  $K$  dažniausiai yra išretinta matrica (angl. sparse matrix). Išretinta matrica yra matrica, kurios didžioji dalis elementų yra lygūs 0. Konvoliucijos naudojimas giliuosiuose neuroniniuose patobulina mokymosi procesą dėl konvoliucijos principų - išretintos sąveikos (angl. sparsity), parametrų pasidalinimo ir ekvivalentiško atvaizdavimo.

Paprastame daugiasluoksniame perceptrone kiekvieno sluoksnio visi perceptronai turi po vie-



ną jungtį su tolimesnio sluoksnio kiekvienu perceptronu. Tuo metu konvoliuciniuose tinkluose yra taikoma išretinta sąveika. Išretinta sąveika - tai konvoliucijos padarinys dirbtiniam neuroniniam tinklui, dėl kurios sluoksniuose, kuriuose taikoma konvoliucija, sumažėja jungčių skaičius su tolimesnio sluoksnio perceptronais. Išretintos sąveikos tarp dviejų sluoksnių pavyzdys yra pateiktas paveikslėlyje 3



3 pav. Išretinta sąveika

Dėl išretintos sąveikos konvoliuciniai neuroniniai tinklai atsižvelgia tik į reikšmingus požymius. Todėl konvoliucinių neuroninių tinklų mokymas trunka trumpiau ir triukšmas turi mažesnę įtaką rezultatui.

Kitas principas dėl kurio konvoliuciniai neuroniniai tinklai yra spartesni nei daugiasluoksniai perceptronai yra parametrų pasidalinimas. Parametrų pasidalinimas - tai principas, kuriuo remiantis bent vienas parametras yra panaudojamas daugiau nei vienai funkcijai. Konvoliucinių neuroninių tinklų atveju kiekvienas branduolio narys yra naudojamas kiekvienam įvesties nariui. Tuo metu daugiasluoksniame perceptrone, kurio svoriai yra matricos, kiekvienas svorio matricos narys yra naudojamas tik vienam įvesties nariui. Todėl svoriai užima mažiau vietos kompiuterio atmintyje.

Paskutinis principas yra ekvivalentiškumas. Funkcija  $f(x)$  yra ekvivalenti funkcijai  $g(x)$  jei patenkinama lygybė  $f(g(x)) = g(f(x))$ . Konvoliucija yra ekvivalenti daugeliui matricos transformacijos funkcijų. Šio principo nauda pasireiškia kei turima nedidelio skaičiaus kaimyninių pikselių funkcija yra naudinga, kai ji yra pritaikoma daugelyje įvesties vietų.

#### 1.4.2. Konvoliucinio neuroninio tinklo sluoksnių tipai

Konvoliucinio neuroninio tinklo sluoksniai yra skirstomi į tipus. Pagrindinis konvoliucinio neuroninio tinklo sluoksnio tipas yra konvoliucijos sluoksnis. Šio sluoksnio paskirtis yra padalinti kiekvieną įėjimo vaizdą į lokalius fragmentus ir nustatyti kiekvieno fragmento atitikimą kiekvienam požymiui naudojantis konvoliucija. Konvoliucijos sluoksnio parametrai yra filtrų skaičius ir jų dimensijos bei lango žingsnis - per kiek pozicijų langas yra paslenkamas kuriant lokalius vaizdo fragmentus. Šio sluoksnio rezultatas yra požymių žemėlapis (angl. feature map). Šiuose žemėlapiuose yra saugoma informacija - kiekvieno fragmento atitikimas konkrečiam požymiui. Kiekvienas požymių žemėlapis yra sudaromas naudojantis unikalų, tik jam priskirtą filtrą. I-tasis požymių žemėlapis  $Y_i^{(l)}$ , priklausantis l-tajam sluoksniui, yra apskaičiuojamas pagal formulę 29, kur  $Y_j^{l-1}$  yra j-tasis praeitas sluoksnis,  $m_1^{(l-1)}$  - praeito sluoksnio perceptronų išėjimų skaičius,  $K_{i,j}^{(l)}$  - nau-

dojamas filtras l-tajame sluoksnyje apskaičiuoti i-tąjį požymių žemėlapij j-tajam įėjimo vaizdui, ir  $B_i^{(l)}$  - tai i-toji l-sluoksnio postūmio matrica (angl. bias matrix).

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{l-1} \quad (29)$$

Konvoliucijos sluoksnio perceptronų išėjimai yra šie žemėlapiai.

Po konvoliucijos sluoksnio tolimesnio sluoksnio tipas konvoliuciniame neuroniniame tinkle yra netiesiškumo sluoksnis (angl. non-linearity layer). Šis sluoksnis yra sudarytas iš aktyvacijos funkcijos ir šio sluoksnio rezultatas yra šios funkcijos rezultatas, vadinamas aktyvacijos žemėlapiu. Kitaip tariant netiesiškumo sluoksnis yra formulė  $Y_i^{(l)} = f(Y_i^{(l-1)})$ . Netiesiškumo sluoksnio aktyvacijos funkcija gali būti bet kuri funkcija, kuri taip pat yra naudojama ir vienasluoksniame perceptrone, ir kurios argumentas yra matrica ir rezultatas yra matrica, kurios matmenys yra lygūs argumento matricos matmenims. Dažniausiai aktyvacijos funkcijos yra sigmoidinė, hiperbolinio tangento ir ištaisymo tiesinė (angl. rectified linear function (ReLU)) funkcijos.

Po netiesiškumo sluoksnio tolimesnis sluoksnis konvoliuciniame neuroniniame yra ištaisymo sluoksnis (angl. rectification layer). Šiame sluoksnyje aktyvacijos žemėlapio reikšmės yra padaromos absoliučiomis. Kitaip tariant ištaisymo sluoksnis yra formulė  $Y_i^{(l)} = |Y_i^{(l-1)}|$ .

Dažniausiai konvoliuciniuose tinkluose netiesiškumo ir ištaisymo sluoksniai yra apjungiami į vieną. Pats paprasčiausias šių sluoksnių apjungimas yra formulė  $Y_i^{(l)} = |f(Y_i^{(l-1)})|$ . Kitas labai populiarus apjungimo metodas yra naudoti ištaisymo tiesinę funkciją (angl. rectified linear unit (ReLU)) kaip aktyvacijos funkciją. Ištaisymo tiesinė funkcija yra  $f(x) = \max(0, x)$ . Ši funkcija yra gana paprasta. Tad jos naudojimas paspartina apmokymo procesą. Taip pat ištaisymo tiesinė funkcija padaro išretinimą dar aktualesniu, nes neigiamas reikšmės ši funkcija pakeičia nuliais. Dažniausiai konvoliuciniuose neuroniniuose tinkluose yra taikomos ištaisymo tiesinės funkcijos variacijos - su triukšmu (angl. noisy ReLU), pavaizduota lygtyje 30, kur  $Y$  yra normalioji distribucija ( $Y \sim \mathcal{N}(0, \sigma(x))$ ), ir nesandari (angl. leaky ReLU), pavaizduota formulėje 31, ištaisymo tiesinės funkcijos.

$$f(x) = \max(0, x + Y) \quad (30)$$

$$f(x) = \begin{cases} x, & \text{jei } x > 0 \\ 0.01x, & \text{jei } x \leq 0 \end{cases} \quad (31)$$

Taip pat dažnai kaip aktyvacijos funkcija yra naudojama softplus funkcija. Softplus funkcija yra  $f(x) = \ln(1 + e^x)$ . Šios funkcijos patenkina sąlygą  $f(x) > 0$ . Tad šių funkcijų rezultatai yra lygūs absoliučioms reikšmėms. Todėl ištaisymo sluoksnis patampa nereikšmingas.

Po kelių konvoliucinių, netiesiškumo ir ištaisymo sluoksnių dažniausiai konvoliuciniame neuroniniame tinkle tolimesnis sluoksnis yra sujungimo sluoksnis (angl. pooling layer). Šio sluoksnio tikslas yra sumažinti aktyvacijos žemėlapių dydžius. Tokiu būdu pagreitinant apmokymo procesą ir sumažinant persimokymą. Sujungimo sluoksnis turi 2 parametrus - filtrą  $F^{(l)} \in \mathbb{N}$  ir žingsnio dydį

$S^{(l)} \in \mathbb{N}$ . Šio sluoksnio įvestis yra trimatis vaizdas, kurio dimensijos yra  $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$ , ir išvestis yra trimatis vaizdas, kurio dimensijos yra  $m_1^{(l)} \times m_2^{(l)} \times m_3^{(l)}$ , kurios yra paskaičiuojamos pagal formules 32.

$$\begin{aligned} m_1^{(l)} &= m_1^{(l-1)} \\ m_2^{(l)} &= \frac{m_2^{(l-1)} - F^{(l)}}{S^{(l)}} + 1 \\ m_3^{(l)} &= \frac{m_3^{(l-1)} - F^{(l)}}{S^{(l)}} + 1 \end{aligned} \quad (32)$$

Išvesties vaizdo skaičiavimas vyksta sukuriant  $F^{(l)} \times F^{(l)}$  dydžio langą ir slenkant jį per kiekvieną įvesties matricą per  $S^{(l)}$  pozicijų. Kiekvienoje lango pozicijoje yra apskaičiuojama viena skaliarinė reikšmė. Dažniausiai ši reikšmė gaunama naudojant maksimalaus sujungimo metodą (angl. max pooling). Naudojant šį metodą apskaičiuojama reikšmė lange yra lygi maksimaliai reikšmei, kuri yra lange. Kitas dažnai naudojamas metodas yra vidutinis sujungimas (angl. average pooling). Naudojant šį metodą apskaičiuojama reikšmė lange yra lygi visų lango reikšmių vidurkiui. Maksimalaus sujungimo metodas yra pranašesnis, nes apmokymo procesas yra spartesnis ir tikslesnis, kai yra naudojamas šis metodas sujungimo sluoksniuose. Jei yra patenkinama sąlyga  $F^{(l)} > S^{(l)}$ , tai langai persidengs ir sujungimo sluoksnis su persidengiančiais langais yra vadinamas persidengiančiu sujungimo sluoksniu (angl. overlapping pooling layer). Kitu atveju - langai nepersidengia ir toks sluoksnis vadinamas nepersidengiančiu sujungimo sluoksniu (angl. non-overlapping pooling layer).

Kitas sluoksnio tipas naudojamas konvoliuciniame neuroniniame tinkle yra pilnai sujungtas sluoksnis (angl. fully connected layer). Šis sluoksnis yra daugiasluoksnis perceptronas, kurio įvestis yra trimatis vaizdas. Jei praeitas sluoksnis irgi yra pilnai sujungtas sluoksnis, tai aktyvacijos funkcijos argumentas  $a_i^{(l)}$  yra gaunamas pagal formulę 33, kitu atveju naudojama formulė 34.

$$a_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} w_{ij}^{(l)} y_i^{(l-1)} \quad (33)$$

$$a_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} \sum_{r=1}^{m_2^{(l-1)}} \sum_{s=1}^{m_3^{(l-1)}} w_{ijrs}^{(l)} (y_i^{(l-1)})_{rs} \quad (34)$$

Pilnai sujungto sluoksnio rezultatas yra klasių tikimybių distribucija.

Konvoliucinio neuroninio tinklo apmokymas tai procesas, kurio metu yra nustatomos konvoliucijos sluoksnių filtrų ir pilnai sujungtų sluoksnių svorių reikšmės su kuriomis nuostolių funkcija įgyja apytiksliai mažiausias reikšmes. Šiame darbe tiriamo konvoliucinio neuroninio tinklo išėjimo pilnai sujungto sluoksnio aktyvacijos funkcija yra softmax. Ši funkcija yra atvaizduota formulėje 35, kur  $i = 1, 2, \dots, K - 1, K$ ;  $K$  - klasių skaičius ir funkcijos argumentas  $\mathbf{z} = (z_1, z_2, \dots, z_{K-1}, z_K) \in \mathbb{R}^K$ .

$$f_i(\mathbf{z}) = \frac{e^{z_i}}{\sum_j^K e^{z_j}} \quad (35)$$

Softmax funkcija yra logistinės regresijos apibendrinimas daugeliui klasių, kuri mažesnes reikšmes sumažina, o didesnes padidina. Šiai aktyvacijos funkcijai yra taikoma kryžminės entropijos (angl. cross-entropy) nuostolių funkcija, atvaizduota formulėje 36.

$$L_i = -\log\left(\frac{e^{z_i}}{\sum_j^K e^{z_j}}\right) \quad (36)$$

Kryžminė entropija tarp tikrojo diskretaus pasiskirstymo  $p(x)$ , kur  $p$  yra vektorius sudarytas iš 0 ir vieno 1 pozicijoje  $x$ , ir gauto diskretaus pasiskirstymo  $q(x)$ , atvaizduoto funkcijoje 38, yra apibrėžta formulėje 37.

$$H(p, q) = -\sum_{x=1}^K p(x) \log(q(x)) \quad (37)$$

$$q(x) = \frac{e^{z_x}}{\sum_j^K e^{z_j}} \quad (38)$$

### 1.4.3. Daugiavaizdis konvoliucinis neuroninis tinklas

## 1.5. Kapsulinių dirbtinių neuroninių tinklų apžvalga

Kapsuliniai neuroniniai tinklai yra aprašyti darbe [SFH17]. Kapsuliniai neuroniniai tinklai yra giliųjų neuroninių tinklų tipas, kurio sluoksnio perceptronai yra grupuojami į kapsules. Kiekviena kapsulė apskaičiuoja tikimybę, kad paveikslėlyje pavaizduotas objektas priklauso kažkuriai klasei, ir išgauna informaciją apie tokio objekto bruožus kaip pozicija, orientacija, mastelis, deformacija, spalva ir kitus panašius objekto bruožus. Pirminių kapsulių sluoksniuose nagrinėjami objektai yra paprastos geometrinės figūros. Tolimesniuose sluoksniuose objektai darosi sudėtingesni, jie ima atitikti realaus pasaulio objektus. Kapsulės tarp sluoksnių yra sujungiamos į hierarchiją. Taip kapsulinis neuroninis tinklas sukuria hierarchinę vaizdo reprezentaciją.

Pirmieji du sluoksniai kapsuliniame neuroniniame tinkle yra konvoliucijos sluoksnis ir apjungtas netiesiškumo ir ištaisymo sluoksnis su ištaisymo tiesine aktyvacijos funkcija. Šių sluoksnių tikslas yra išgauti pagrindinius požymius, kurie tolimesniame sluoksnyje yra naudojami objektų konstrukcijai.

Tolimesnio sluoksnio tipas yra pirminės kapsulės (angl. primary capsules). Šiame sluoksnyje aktyvacijos žemėlapiai yra konvertuojami į vektorius. Toliau kiekvienas vektorius atskirai yra pateikiamas squash funkcijai kaip argumentai. Squash funkcija yra formulė ??, kur  $\|s\|$  yra visų matricos  $s$  narių suma.

$$squash(s) = \frac{\|s\|^2}{1 + \|s\|^2} \frac{s}{\|s\|} \quad (39)$$

Tolimesnių sluoksnių tipai yra kapsulės. Šiuose sluoksniuose yra vykdomas dinaminis marš-

rutizavimas tarp kapsulių (angl. dynamic routing between capsules). Dinaminis maršrutizavimas tarp kapsulių - tai iteratyvus procesas, kurio paskirtis yra apjungti kapsules tarp dviejų sluoksnių. Prieš pradėdant iteratyvią proceso dalį, kiekvienai sluoksnio  $l$  kapsulei  $i$  ir sluoksnio  $(l + 1)$  kapsulei  $j$  yra inicializuojami kintamieji  $b_{ij}$  su reikšme 0. Taip pat kiekvienai kapsulių  $i$  ir  $j$  porai yra apskaičiuojami vektoriai  $\hat{u}_{j|i}$  pagal formulę 40, kur  $W_{ij}$  yra svorio matrica tarp kapsulių  $i$  ir  $j$  bei  $u_i$  - tai kapsulės  $i$  išvestis.

$$\hat{u}_{j|i} = W_{ij}u_{ij} \quad (40)$$

Tada pirmasis iteratyvaus proceso žingsnis yra apskaičiuoti apjungimo koeficientus  $c_{ij}$  kiekvienai kapsulių  $i$  ir  $j$  porai pagal softmax funkciją atvaizduota formulėje 41, kur  $n$  yra sluoksnio  $(l + 1)$  kapsulių skaičius.

$$c_{ij} = \frac{\exp^{b_{ij}}}{\sum_{k=1}^n \exp^{b_{ik}}} \quad (41)$$

Tolimesnis žingsnis yra apskaičiuoti svertines sumas  $s_j$  kiekvienai kapsulei  $j$  naudojantis formulę 42, kur  $m$  yra kapsulių skaičius sluoksnyje  $l$ .

$$s_j = \sum_{i=1}^m c_{ij}\hat{u}_{j|i} \quad (42)$$

Toliau yra apskaičiuojami vektoriai  $v_j$  kiekvienai kapsulei  $j$  naudojantis softmax funkcija su argumentu  $s_j$ . Kitaip tariant yra apskaičiuojama formulė  $v_j = \text{softmax}(s_j)$ . Paskutinis iteratyvios dalies žingsnis yra pakeisti kintamųjų  $b_{ij}$  reikšmes naudojantis formulę  $b_{ij} = b_{ij} + \hat{u}_{j|i}v_j$ .

Iteratyvi dinaminio maršrutizavimo tarp kapsulių proceso dalis yra kartojama nurodyta skaičių iteracijų ir šio proceso rezultatas yra vektorius  $v_j$ . Šiame vektoriuje yra tikimybės, kad objektas, nagrinėjamas kapsulės  $i$ , yra dalis objekto, nagrinėjamo kapsulės  $j$ .

## **2. Kapsulinių neuroninių tinklų modifikacijos ir parametrai**

### **3. Eksperimentiniai tyrimai**

#### **3.1. Tyrimams naudoti duomenys**

#### **3.2. Kapsulinių neuroninių tinklų modifikacijų ir parametrų eksperimentiniai tyrimai**

#### **3.3. Kapsulinių ir daugiavaizdžių konvoliucinių neuroninių tinklų eksperimentiniai tyrimai**

## **Rezultatai ir išvados**

Rezultatų ir išvadų dalyje išdėstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas įdiegta), pateikiamos išvados (daromi nagrinėtų problemų sprendimo metodų palyginimai, siūlomos rekomendacijos, akcentuojamos naujovės).



## Literatūra

- [CK04] C. M. Cyr and B. B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.124.4344&rep=rep1&type=pdf>, 2004. 3820 KB, accessed 2020-08-15.
- [CTS<sup>+</sup>03] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. <http://www.cs.jhu.edu/~misha/Papers/Chen03.pdf>, 2003. 1910 KB, accessed 2020-08-15.
- [Hay04] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 2004. volume 2.
- [IL65] A. G. Ivakhnenko and V. G. Lapa. *Cybernetic Predicting Devices*. CCM Information Corporation, 1965.
- [KB15] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. <https://arxiv.org/pdf/1412.6980.pdf>, 2015. 571 KB, accessed 2020-09-12.
- [LBD<sup>+</sup>89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. <http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf>, 1989. 5530 KB, accessed 2020-09-14.
- [Mas18] F. Vitor Suzano Massa. Relating images and 3d models with convolutional neural networks. <https://pastel.archives-ouvertes.fr/tel-01762533/document>, 2018. 7.917 KB, accessed 2020-05-23.
- [MC17] R. Mukhometzianov and J. Carrillo. Capsnet comparative performance evaluation for image classification. <https://arxiv.org/ftp/arxiv/papers/1805/1805.11195.pdf>, 2017. 899 KB, accessed 2020-05-23.
- [MN95] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. <http://murase.m.is.nagoya-u.ac.jp/~murase/pdf/704-pdf.pdf>, 1995. 3041 KB, accessed 2020-08-15.
- [MSD<sup>+</sup>02] D. Macrini, A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. View-based 3-d object recognition using shock graphs. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.852&rep=rep1&type=pdf>, 2002. 155 KB, accessed 2020-08-15.
- [Ros57] F. Rosenblatt. *The Perceptron—a perceiving and recognizing automaton*. Cornell Aeronautical Laboratory., 1957. 386–408 p.
- [Rou06] N. Rountree. *Initialising Neural Networks with Prior Knowledge*. PhD thesis, 2006.
- [SFH17] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. <https://arxiv.org/pdf/1710.09829.pdf>, 2017. 899 KB, accessed 2020-05-23.

- [SMK<sup>+</sup>15] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. [https://www.cv-foundation.org/openaccess/content\\_iccv\\_2015/papers/Su\\_Multi-View\\_Convolutional\\_Neural\\_ICCV\\_2015\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Su_Multi-View_Convolutional_Neural_ICCV_2015_paper.pdf), 2015. 1.439 KB, accessed 2020-05-23.
- [WSK<sup>+</sup>15] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: a deep representation for volumetric shapes. [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Wu\\_3D\\_ShapeNets\\_A\\_2015\\_CVPR\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Wu_3D_ShapeNets_A_2015_CVPR_paper.pdf), 2015. 2.451 KB, accessed 2020-05-23.

## **Santrumpos**

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

## **Priedas Nr. 1**

### **Niauroninio tinklo struktūra**

## Priedas Nr. 2

### Eksperimentinio palyginimo rezultatai

1 lentelė. Lentelės pavyzdys

Algoritmas	$\bar{x}$	$\sigma^2$
Algoritmas A	1.6335	0.5584
Algoritmas B	1.7395	0.5647