

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Konvoliucinių dirbtinių neuroninių tinklų
taikymas statistinėje duomenų analizėje**

**Application of Artificial Convolutional Neural Networks in
Statistical Data Analysis**

BAKALAURO DARBAS

Atliko: Edvardas Balkunas (parašas)
Darbo vadovas: asist. dr. Vytautas Valaitis (parašas)
Darbo recenzentas: lekt. Saulė Girdzijauskaitė (parašas)

Vilnius – 2018

TURINYS

SANTRAUKA	3
SUMMARY	4
IVADAS	5
Tikslas ir uždaviniai	5
Susiję darbai	6
Metodika.....	6
Naudoti įrankiai	7
1. LITERATŪROS APŽVALGA	8
1.1. Neuroniniai tinklai.....	8
1.1.1. Vidutinis abipusis rangas	8
1.1.2. Duomenų analizės metodai	8
1.1.2.1. Naivusis Bajesas.....	8
1.1.2.2. „Random Forest”	9
1.1.3. Dirbtinis neuronas	9
1.1.4. Neuroninio tinklo struktūra	10
1.1.5. „Siunčiantys į priekį” neuroniniai tinklai	11
1.1.6. Rekurentiniai neuroniniai tinklai	11
1.2. Neuroninių tinklų mokymo būdai	12
1.2.1. Mokymas be priežiūros	12
1.2.2. Mokymas su priežiūra.....	12
1.3. Permokymas	13
1.4. Gilusis mokymas.....	14
1.5. Konvoliuciniai neuroniniai tinklai.....	14
1.5.1. Konvoliucinis sluoksnis	14
1.5.2. Sujungimo sluoksnis	15
1.5.3. Netiesinis sluoksnis	15
1.5.4. Pilno sujungimo sluoksnis	16
2. DUOMENŲ RINKINYS	17
2.1. Originalus duomenų rinkinys	17
2.2. Paruoštas duomenų rinkinys	18
3. STATISTINĖS ANALIZĖS ATLIKIMAS NEURONINIO TINKLO PAGALBA	19
3.1. Pradinio taško nustatymas	19
3.2. Bibliotekos ypatumai.....	19
3.3. Eksperimento atlikimas ir tinklo parametrų įtaka tinklo tikslumui	20
3.3.1. Neuronų sluoksnių skaičius.....	20
3.3.2. Duomenų partijos dydis	21
3.3.3. Netiesiškumas	22
3.3.4. L2 generalizacija.....	23
3.3.5. Netekties funkcija	23
REZULTATAI IR IŠVADOS	25
Rezultatai	25
Išvados	25
ŠALTINIAI	26
SANTRUMPOS	29

Santrauka

Dėl didelio informacinių technologijų vartotojų generuojamo duomenų kiekio dažnai ieškoma būdų automatizuoti šių duomenų statistinės analizės atlikimą. Vienas tokių būdų – mašininio mokymosi (angl. *machine learning*) metodų taikymas. Populiariausias tokių metodų yra dirbtiniai neuroniniai tinklai. Šiame darbe nagrinėjamas konvoliucinių dirbtinių neuroninių tinklų (angl. *Convolutional artificial neural networks*), kaip vienu plačiausiai taikomų dirbtinių neuroninių tinklų tipų, taikymas statistinėje duomenų analizėje (angl. *statistical analysis*). Statistinės duomenų analizės atlikimui buvo pasirinktas duomenų rinkinys – Vilniaus viešojo transporto vėlavimo duomenys nuo 01-01-2018 iki 07-05-2018. Atlikus statistinę duomenų analizę konvoliucinio neuroninio tinklo pagalba buvo atlikta neuroninio tinklo parametrų įtaka neuroninio tinklo spėjimo tikslumui analizė. Gautos šios išvados: apmokytas neuroninis tinklas spėja geresniu tikslumu negu atsitiktinai (29% prieš 8%) arba taikant populiarius mašininio mokymo algoritmus – Naivųjį Bajesą ir "Random Forest" (apie 7.8%), aukšto tikslumo galima pasiekti naudojant nedaug neuronų sluoksnių (iki 3), pasirinktam duomenų rinkiniui geriausiai tinka „hinge” netekties funkcija, apsauga nuo persimokymo (angl. *overfitting*) teigiamai veikia spėjimo tikslumą realaus pasaulio duomenims, "relu" netiesiškumo funkcija teigiamai veikia neuroninio tinklo tikslumą.

Raktiniai žodžiai: dirbtiniai neuroniniai tinklai, statistinė duomenų analizė, konvoliuciniai dirbtiniai neuroniniai tinklai, dirbtinio neuroninio tinklo parametrai.

Summary

Nowadays, amount of data generated by IT users is growing at a significant pace. To process all of the generated data new ways statistical analysis automatization are created. One of these ways is machine learning. Probably, the most popular technique used in machine learning are artificial neural networks. This paper looks into application of convolutional neural networks, one of the most widespread types of neural networks, in statistical analysis. The dataset for the experiment is the set of Vilnius public transportation lag from 01-01-2018 to 07-05-2018. After training the convolutional neural network with chosen dataset, influence of neural network's parameters to neural network's accuracy was analyzed. The experiment has yielded following results: accuracy is substantially better than taking a random guess (29% vs 8%) or applying popular machine learning algorithms like Naive Bayes and Random Forest (about 7.8%), the maximum accuracy can be achieved with low numbers of layers (up to 3), hinge loss function is suitable for the chosen dataset, L2 regularization positively affects accuracy of the neural network, relu non-linearity function positively influences accuracy of the neural network.

Keywords: artificial neural networks, statistical analysis, convolutional artificial neural networks, parameters of artificial neural networks.

Įvadas

Esminė šio darbo tiriama tema yra konvoliucinių neuroninių tinklų (angl. *convolutional neural networks*) taikymas statistinėje duomenų analizėje. Šio tipo neuroniniai tinklai yra labai plačiai naudojami šiuolaikiniuose giliojo mokymosi architektūrose (angl. *deep learning*). Tokie tinklai sėkmingiausiai taikomi problemų, susijusių su vaizdine medžiaga, sprendimu. Pavyzdžiui, tam tikrų objektų (veidų, gyvūnų, automobilių ir t.t.) atpažinimas, ranka rašyto teksto nuskaitymas[BBG⁺16], juodai-baltų fotografijų nuspalvinimas[LMS16]. Be to, CNN sėkmingai taikomi natūralios kalbos (angl. *Natural Language Processing*) apdorojime[LK17]. Kiek rečiau, konvoliuciniai neuroniniai tinklai naudojami statistinėje duomenų analizėje (SA).

Statistinė duomenų analizė – procesas skirtas nustatyti tam tikros pasirinktos populiacijos savybes analizuojant žinomus duomenis apie šią populiaciją. Dažniausiai, norint gauti naudingus statistinės analizės rezultatus reikalingas aukštos kvalifikacijos analitikas. Jeigu imtis yra sąlyginai nedidelė ir ,svarbiausia, ji mažai kinta arba nekinta išvis, užteks vieno ar kelių analitikų. Tačiau šiuolaikiniai informacinių technologijų (IT) pasiekimai, tokie kaip „Big Data“, ženkliai padidina analizuojamą duomenų kiekį. Be to, svarbiausios „Big Data“ savybės, tokios kaip didelė apimtis, aukštas naujų duomenų gavimo dažnis ir/arba aukštas duomenų variabilumas, ženkliai apsunkina analitikų darbą.

Natūralu, kad tokios aplinkybės reikalauja naujų instrumentų atsiradusių problemų sprendimui, todėl norėtųsi automatizuoti visą arba bent jau dalį SA proceso. Šiam tikslui pasiekti gali būti naudojami kiti IT pasiekimai – bendruoju atveju tai gilusis mokymasis (DL), o konkrečiau – neuroniniai tinklai (CNN, RNN ir t.t.). Vienas praktikoje taikomų automatizavimo būdų – sukurti neuroninį tinklą, kuris nuspėtų konkretaus vartotojo (objekto) veiksmus, remdamasis anksčiau surinktais duomenimis. Dažnai, tokiam tinklui sukurti naudojami rekurentiniai neuroniniai tinklai RNN, bet bandoma taikyti ir CNN, kaip vieną plačiausiai dabar naudojamų neuroninių tinklų architektūrų. Šiame darbe bandoma parodyti, kaip pasirinktos konvoliucinio neuroninio tinklo realizacijos parametrų keitimas veikia tokių tinklų atliekamų spėjimų tikslumą.

Tikslas ir uždaviniai

Šio bakalauro darbo tikslas – patikrinti konvoliucinio neuroninio tinklo tinkamumą duomenų rinkinio apie Vilniaus viešojo transporto vėlavimus nuo 01-01-2018 iki 07-05-2018 statistinės analizės atlikimui bei nustatyti šio tinklo parametrų įtaką dirbtinio neuroninio tinklo spėjimo tikslumui. Tikslui pasiekti buvo išskirti šie uždaviniai:

1. Pasirinkti ir paruošti duomenų rinkinį.
2. Konvoliucinio neuroninio tinklo pagalba atlikti paruošto duomenų rinkinio statistinę duomenų analizę ir nustatyti dirbtinio konvoliucinio neuroninio tinklo parametrų įtaką tinklo spėjimų tikslumui.

Susiję darbai

Vienas dažniausių konvoliucinių neuroninių tinklų taikymo statistinei analizei atvejų – finansų sektorius. Egzistuoja gana didelis kiekis tyrimų, kurie siūlo būdus atlikti tam tikrų finansinių duomenų (akcijų kainų, valiutų vertės) analizę neuroninių tinklų pagalba ir, naudojantis analizės metu gautomis išvadomis, apmokyti tinklą nuspėti nagrinėjamo objekto iš finansinio sektoriaus kitimą. Dalis tokių tyrimų remiasi konvoliucinių neuroninių tinklų architektūra, pavyzdžiui, šiame darbe[BBO17] pasiūlytas būdas leidžia efektyviai nuspėti valiutų vertę nenaudojant ilgalaikių istorinių duomenų.

Kitas dažnas taikymo atvejis – orų prognozė. Šiame tyrime[Rom15] statistiniai duomenys susiejami kartu su palydovinėmis nuotraukomis. Tokia kombinacija leidžia nuspėti orus tam tikruose nedidelėse vietovėse.

Vienas įdomesnių CNN taikymo atvejų SA – efektyvių ėjimų strategijų šachmatuose paieška. Šiame tyrime[OK15] CNN pagalba atliekama profesionalių šachmatų žaidėjų ėjimų statistikos analizė. Tyrimo metu gautas aukštas tikslumas – 44.4%.

Taikymas socialinėse tinkluose – vaizdo įrašų populiarumo spėjimas[TAB⁺17], politikų populiarumo prognozavimas remiantis jų nuotraukomis socialiniuose tinkluose[ABL⁺17].

Konvoliuciniai neuroniniai tinklai taikomi ir IT infrastruktūrų optimizavimui – trumpalaikio duomenų centro apkrovimo nustatymas[MOG18].

Metodika

Tiksliui pasiekti šis darbas buvo padalintas į kelis etapus:

1. Esamų CNN taikymo SA pavydžių paieška.
2. Duomenų rinkinio paieška.
3. Duomenų rinkinio paruošimas.
4. Eksperimento atlikimas.
5. Eksperimento rezultatų analizė.

Atlikus esamų sprendimų ir CNN realizacijų apžvalgą buvo pradėta duomenų rinkinio paieška. Pasirinktas duomenų rinkinys – duomenys apie viešojo transporto vėlavimus nuo laikotarpio nuo 01-01-2018 iki 07-05-2018. Šis rinkinys gautas iš Atviro Vilniaus „Atvirų Duomenų Portalo“[Ope]. Pasirinkti duomenys savo pradiniam pavidale nebuvo tinkami pritaikyti CNN mokymui – kiekvienas duomenų rinkinio įrašas neturėjo identifikatoriaus. Todėl kiekvienam įrašui atsitiktiniu būdu buvo sugeneruotas identifikatorius. 80% duomenų rinkinio buvo panaudota tinklo mokymui, 10% – testavimui, 10% – validavimui.

Po rinkinio paruošimo neuroniniam tinklui buvo nustatyti parametrų (iteracijų (angl. *epoch*) ir neuronų sluoksnių skaičius, duomenų partijos (angl. *data batch*) dydis, mokymosi greitis (angl. *learning rate*), netekties funkcija (angl. *loss function*) ir t.t.) rinkiniai. Tinklas buvo apmokytas naudojant nustatytus parametrų rinkinius. Po eksperimento buvo atlikta parametrų įtakos dirbtinio neuroninio tinklo spėjimo tikslumui analizė.

Naudoti įrankiai

Eksperimentas buvo atliktas VU MIF esančiame kompiuteryje. Esminiai kompiuterio komponentai, svarbūs eksperimento atlikimui: procesorius – AMD Ryzen 7 1700X 8 branduolių, vaizdo plokštė – NVIDIA GTX 1080 Ti. Pasinaudota CNN realizacija iš Spotlight[Kul17] sistemos (angl. *framework*).

1. Literatūros apžvalga

Šiame skyriuje atliekama darbe naudotos literatūros apžvalga, pateikiami esminiai apibrėžimai ir konceptai. Trumpai bus aptarti tokie apibrėžimai kaip dirbtinis neuroninis tinklas (angl. *artificial neural network*), neuroninio tinklo sandara: neuronų sluoksniai (angl. *neural layers*), dirbtinis neuronas, skirtumas tarp negilaus (angl. *shallow*) ir gilaus (angl. *deep*) mokymosi, plačiausiai naudojami tinklų tipai. Kiek labiau bus aptarti konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks*), kadangi jie – esminis šio darbo nagrinėjamas objektas.

1.1. Neuroniniai tinklai

„Dirbtinis neuroninis tinklas (angl. *artificial neural network*) – tai matematinis modelis, kuris bando simuliuoti biologinių neuroninių tinklų struktūrą ir funkcionalumą”[KBK11]. Kiekvieno neuroninio tinklo pagrindinis sudedamasis vienetas – dirbtinis neuronas (angl. *Artificial neuron*). Iš dirbtinių neuronų sudaryta visuma vadinama neuronų sluoksniu (angl. *neural layer*). Tokių neuronų sluoksnių ir jų tarpusavio ryšių visuma ir yra vadinama dirbtiniu neuroniniu tinklu.

1.1.1. Vidutinis abipusis rangas

Šiame darbe, neuroninio tinklo tikslumui patikrinti yra naudojamas vidutinis abipusis rangas. Vidutinis abipusis rangas (angl. *Mean reciprocal rank*) – dydis, rodantis sistemų, kurias užklauišius, gražinamas rezultatų sąrašas. Kai atsakymas į užklausą susideda iš vieno elemento, vidutinis abipusis rangas yra $\frac{1}{rangas}$, kur *rangas* yra teisingo atsakymo pozicija gražintame sąrašė. Pvz. užklauišiamas skaičius 1, gautas atsakymas – (2, 3, 4, 5, 1), vieneto pozicija yra 5, todėl vidutinis abipusis rangas bus $\frac{1}{5}$. Kai atliekamos kelios užklauišos Q , Vidutinis abipusis rangas skaičiuojamas formulėje 1:

$$MRR = \frac{1}{Q} \sum_{i=1}^Q \frac{1}{rank_i}. \quad (1)$$

1.1.2. Duomenų analizės metodai

Statistinės analizės (angl. *statistical analysis*) metodai Mašiniame mokyme (angl. *machine learning*) yra įgyvendinami tam tikrų algoritmų pagalba. Plačiaja prasme, šiame darbe yra nagrinėjamas klasifikacijos (angl. *classification*) uždavinio atvejis. Pradinio taško (angl. *baseline*) nustatymui buvo pasirinkti keli klasifikatoriai, atsižvelgiant į jų tikslumą[FCB⁺14]: „Naives Bayes” ir „Random Forest”.

1.1.2.1. Naivusis Bajesas

Naivusis Bajesas – vienas iš taip vadinamų tikimybinių klasifikatorių (angl. *probabilistic classifiers*). Šis klasifikatorius veikia remiantis Bajeso teorema. Daroma prielaida, kad duomenų rinkinyje aprašytos ypatybės (angl. *features*) yra viena nuo kitos nepriklausomos, todėl šis metodas ir pavadintas naiviu. Veikimo principas:

1. Sudaromos visos įmanomos savybių (angl. *features*) iš duomenų rinkinio kombinacijos.

2. Pagal duomenų rinkinį kiekvienai klasei nustatomos šių kombinacijų tikimybės.
3. Priklausymas klasei nustatomas pagal tai, kurioje klasėje didžiausia tikimybė sutikti užklaustą savybių kombinaciją.

1.1.2.2. „Random Forest”

Šis algoritmas yra paremtas sprendimų medžio (angl. *decision tree*) veikimu. Šis metodas veikia gana paprastai:

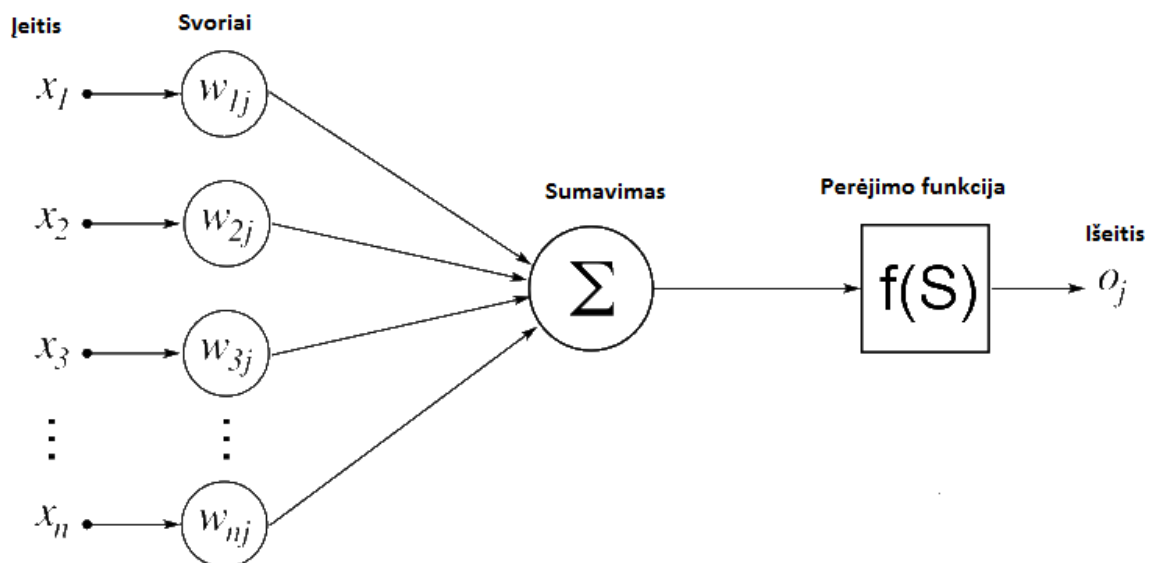
1. Iš duomenų rinkinio atsitiktiniu būdu pasirenkama n įrašų ir m rinkinyje naudojamų kintamųjų.
2. Iš šių įrašų ir kintamųjų sudaromas sprendimų medis.
3. Žingsniai 1 ir 2 kartojama tiek, kiek yra klasių. Taip gaunamas sprendimų medžių miškas.
4. Remiantis testiniu rinkiniu, kiekvienai klasei priskiriamas labiausiai tinkantis medis.

1.1.3. Dirbtinis neuronas

Kaip ir buvo minėta, dirbtinis neuronas – tai pati mažiausia dirbtinio neuroninio tinklo sudedamoji dalis. „Jo struktūra ir funkcionalumas yra sukurti remiantis biologinio neurono struktūromis, kuris, savo ruožtu, yra biologinių neuroninių tinklų (sistemų), tokių kaip smegenys, stuburas ar periferinė ganglija, pamatinis vienetas.”[KBK11]. Dirbtinio neurono modelis susideda iš trijų matematinių veiksmų – daugybos, sudėties ir perėjimo funkcijos (angl. *transfer function*) taikymo. Tokio tipo neuronas veikia gana paprastai:

1. Visi įėjties duomenys ir padauginami iš tam tikrų svorių (angl. *weights*).
2. Įėjties duomenų ir atitinkamų svorių sandaugos yra sumuojamos.
3. Gautas rezultatas transformuojamas naudojant perėjimo funkciją.

Vizualiai dirbtinio neurono veikimo principas pavaizduotas 1 pav.:



1 pav. Dirbtinis neuronas[Pre].

Visą aukščiau aprašytą ir pavaizduotą procesą galima aprašyti formule 2:

$$y(k) = f\left(\sum_{i=0}^n w_i(k) \cdot x_i(k)\right). \quad (2)$$

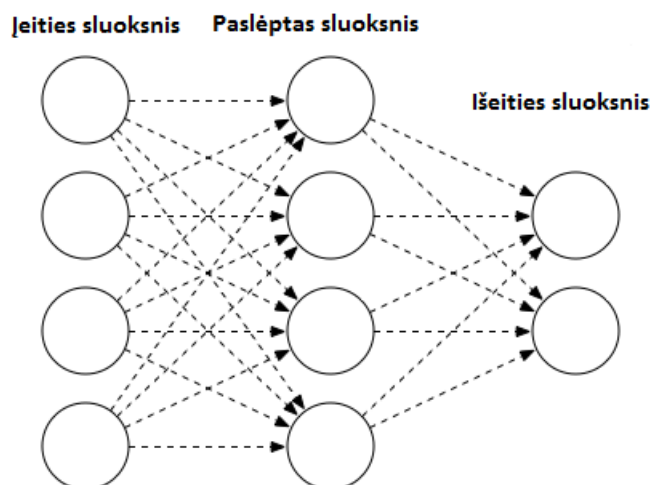
Kur:

- $x_i(k)$ - įeities reikšmė laiko momentu k
- $w_i(k)$ - svorio reikšmė laiko momentu k
- f - perėjimo funkcija
- y_i - išeities reikšmė laiko momentu k

„Kaip matyti iš dirbtinio neurono modelio ir jį aprašančios formulės, svarbiausias šio modelio nežinomas yra perėjimo funkcija f . Perėjimo funkcija nustato dirbtinio neurono savybes ir gali būti bet kokia matematinė funkcija. Mes pasirenkame ją priklausomai nuo dirbtinio neurono (dirbtinio neuroninio tinklo) sprendžiamos problemos ir, dažniausiai, pasirenkame ją iš šių funkcijų aibių: Laiptinių (angl. *step*) funkcijų, tiesinių (angl. *linear*) ir netiesinių (Sigmoidinių) (angl. *non-linear (Sigmoid)*) funkcijų.[KBK11]

1.1.4. Neuroninio tinklo struktūra

„Kai sukombinuojame du arba daugiau dirbtinių neuronų, mes gauname dirbtinį neuroninį tinklą. Jeigu vienas dirbtinis neuronas neturi beveik jokios naudos realaus pasaulio problemų sprendimams, tai dirbtinis neuroninis tinklas tiekia tokią naudą. Iš tikrųjų, dirbtiniai neuroniniai tinklai geba spręsti sudėtingas realaus pasaulio problemas apdorodami informaciją savo bazinėse sudedamosiose dalyse (dirbtiniuose neuronuose) netiesiškai (angl. *non-linear*), paskirstytai (angl. *distributed*), lygiagrečiai (angl. *parallel*) ir lokaliai.”[KBK11]. Paprasčiausiu atveju, neuroninis tinklas gali būti pavaizduotas 2 pav.



2 pav. Neuroninio tinklo sandara[Hil].

Kadangi visi tinkle esantys dirbtiniai neuronai gali būti sujungti tarpusavyje, galimų tinklo variantų kiekis greitai auga priklausomai nuo neuronų skaičiaus. Tai, kaip neuronai sujungti

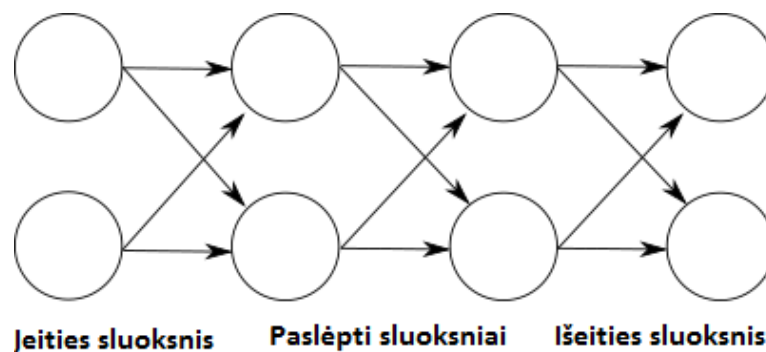
tarpusavyje yra vadinama neuroninio tinklo topologija (angl. *topology*). Dažniausiai šiuo metu naudojamos šios topologijos: „siunčiantys į priekį“ neuroniniai tinklai (angl. *Feed-forward neural network*) ir rekurentiniai neuroniniai tinklai (angl. *Recurrent neural networks*).

1.1.5. „Siunčiantys į priekį“ neuroniniai tinklai

„Siunčiantys į priekį“ neuroniniai tinklai (angl. *feed-forward neural networks*) susideda iš mažiausiai trijų neuronų sluoksnių (angl. *layers*): įėjties sluoksnio (angl. *input layer*), paslėpto sluoksnio (angl. *hidden layer*) ir išėjties sluoksnio (angl. *output layer*). Paslėptų sluoksnių gali būti daugiau negu vienas. Esminė savybė, skirianti „siunčiantį į priekį“ neuroninį tinklą nuo kitų tipų tinklų yra tai, kad kiekvienas dirbtinis neuronas iš esamo sluoksnio yra sujungtas su kiekvienu dirbtiniu neuronu iš praeito sluoksnio. Be to, duomenys negali būti siunčiami į praeitus sluoksnius, todėl tokio tipo tinkluose nėra ciklų. Kiekviena jungtis tarp neuronų turi skirtingus skaitinius įverčius arba svorius (angl. *weight*). „Siunčiančio į priekį“ neuroninio tinklo veikimas gali būti aprašytas šia veiksmų seka:

1. Duomenys patenka į įėjties sluoksnį. Šiame sluoksnyje gauti duomenys nėra keičiami, o tiesiog perduodami į tolimesnį – paslėptą sluoksnį.
2. Duomenys keliauja per kiekvieną paslėptą sluoksnį.
3. Atvykusiems į išėjties sluoksnį duomenims taikoma perėjimo funkcija (angl. *transfer function*).

Įprastiniu atveju, kai tokios topologijos tinklas veikia kaip klasifikatorius (angl. *classifier*), sluoksnyje n gautas rezultatas nėra siunčiamas $n - 1$ sluoksniui. Todėl toks tinklas yra vadinamas „siunčiančiu į priekį“ neuroniniu tinklu. Aukščiau aprašytas veikimo principas pavaizduotas 3 pav.

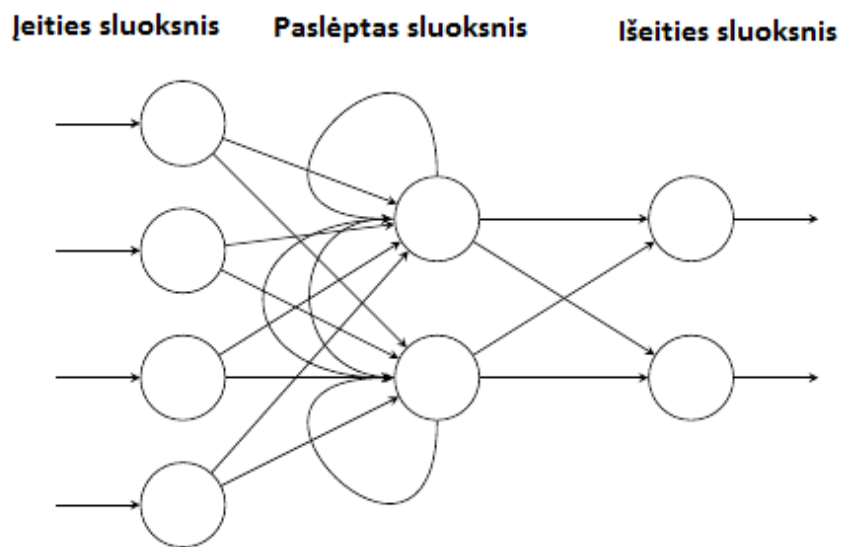


3 pav. „Siunčiantis į priekį“ neuroninis tinklas[Wik].

1.1.6. Rekurentiniai neuroniniai tinklai

„Dirbtiniai neuroniniai tinklai su rekurentine topologija yra vadinami Rekurentiniu dirbtiniu neuroniniu tinklu.”[KBK11]. Rekurentiniai tinklai skiriasi nuo „siunčiančių į priekį“ tinklų tuo, kad rekurentinėje topologijoje yra leidžiami ciklai. Ciklų būvimas sukuria tam tikrą vidinę

neuroninio tinklo būseną, kuri leidžia numatyti neakivaizdžius atvejus. Iš esmės, ši tinklo būseną imituoja atmintį, kurios turinys gali būti pritaikomas tolimesniems įėjties duomenims. Viena populiariausių rekurentinio neuroninio tinklo topologijų yra „Pilnai rekurentinė“ (angl. *Fully recurrent*), kurioje kiekvienas dirbtinis neuronas yra sujungtas su **kiekvienu** kitu dirbtiniu neuronu (įskaitant ir save). Rekurentiniai neuroniniai tinklai gerai tinka spręsti regresinius (angl. *regression*) uždavinius. Kiti naudojami rekurentinių neuroninių tinklų tipai: Hopfildo (angl. *Hopfield*), Elmano (angl. *Elman*), Džordano (angl. *Jordan*), bi-direkciniai (angl. *bi-directional*) ir t.t. Rekurentinio neuroninio tinklo topologija vaizduojama 4 pav.



4 pav. Rekurentinis neuroninis tinklas[Joh].

1.2. Neuroninių tinklų mokymo būdai

Šiame poskyryje kalbama apie neuroninių tinklų apmokymo būdus. Bus trumpai aptarti du esminiai mokymo būdai – mokymas su priežiūra (angl. *Supervised learning*) ir mokymasis be priežiūros (angl. *Unsupervised learning*).

1.2.1. Mokymas be priežiūros

Mokymas be priežiūros – neuroninio tinklo mokymo būdas kai kartu su įėjties duomenimis nėra pateikiami teisingi išeities duomenys. Šiuo atveju, tinklui yra pateikiami įėjties duomenys ir kainos (angl. *cost*) funkcija, kurią reikia minimizuoti. Kainos funkcija priklauso nuo nagrinėjamo uždavinio. „Kažkuria prasme, į mokymą be priežiūros gali būti žiūrima kaip į šablonų (angl. *patterns*) paiešką pateiktuose duomenyse, bet ne nestruktūrizuotame triukšme”[Gha04].

1.2.2. Mokymas su priežiūra

„Mokymasis su priežiūra (angl. *supervised learning*) arba Asociatyvus mokymasis (angl. *Associative learning*), kuris treniruoja tinklą pateikdamas jam įeitį ir atitinkamus išeities šablonus

(angl. *pattern*). Šios įeities-išeities poros gali būti suteiktos išorinio mokytojo arba sistemos, kuriame yra neuroninis tinklas (savipriežiūra (angl. *self-supervised*)).”[KS96]. Tradiciškai, įeities ir išeities duomenys yra atvaizduojami vektoriais. Mokymasis su priežiūra, iš esmės, gali būti traktuojamas kaip klasifikacijos uždavinys, turintis platų klasifikatorių spektrą, kurių kiekvienas turi savo silpnybes ir stiprybes. Tinkamo klasifikatoriaus pasirinkimas yra atskiras ir sudėtingas uždavinys. Kai kurie klasifikatoriai:

- Daugiasluksniai perceptronai (angl. *Multilayer Perceptrons*)
- k-artimiausių kaimynų algoritmas (angl. *k-nearest neighbors*)
- Gauso
- Sprendimų medis (angl. *Decision tree*)

Norint išspręsti problemą taikant mokymą su priežiūra, atliekami šie žingsniai:

- Nustatyti mokymo pavyzdžių tipą
- Sudaryti duomenų rinkinį, tinkamai apibūdinantį nagrinėjamą problemą
- Paruošti duomenų rinkinį taip, kad jis būtų suprantamas pasirinktam neuroniniam tinklui.
- Mokyti tinklą ir patikrinti, kaip gerai tinklas įsisavino duomenų rinkinį.

1.3. Permokymas

Mašiniame mokyme (angl. *machine learning*) egzistuoja dvi su duomenų rinkiniu susijusios problemos - permokymas (angl. *overfitting*) ir nepakankamas išmokimas (angl. *underfitting*). Permokymas atsiranda, kai mašininio mokymo algoritmas be reikalingų duomenų pasiima ir jų triukšmą (angl. *noise*). Trumpai tariant, šis reiškinys įvyksta, kai mokymo algoritmas per stipriai prisitaiko prie duotų duomenų[CJ13]. Permokymas įvyksta, kai modelyje naudojamo algoritmo „šališkumas”(angl. *bias*) yra žemas, o variabilumas (angl. *variance*) - didelis. Praktikoje tai reiškia, kad mokymui parinktas duomenų rinkinys yra per didelis duomenų įvairovės prasme. Permokymo kompensavimui taikomi šie būdai:

- „Cross-validation” - padalinti esamą duomenų rinkinį į mažesnius. Tuomet neuroninis tinklas apmokomas su kiekvienu „mini-rinkiniu”.
- Ankstyvus stabdymas (angl. *early stopping*) - tam tikras mokymo iteracijų skaičius didina gero spėjimo tikimybę. Po tam tikro kiekio tikslumas pradeda mažėti. Šio metodo esmė - sustoti prieš pasiekiant šį tašką.

Šiame darbe naudotas permokymo mažinimo būdas - L2 regularizacija, kartais tai vadinama L2 bauda (angl. *penalty*). Pagrindinė regularizatoriaus užduotis - sumažinti duomenų rinkinio sudėtingumą, bet palikti tą patį įrašų skaičių. L2 regularizacija pasiekia šį efektą uždedant svorio baudas (angl. *penalties*) aukštai svyruojančiams parametrams[Velb]. Regularizatoriaus naudojimas prideda dydžio, aprašyto formule 3, kainą prie netekties (angl. *loss*) funkcijos.

$$\frac{\lambda}{2} \|\vec{\omega}\|^2 = \frac{\lambda}{2} \sum_{i=0}^W \omega_i^2. \quad (3)$$

Čia $\vec{\omega}$ - svorių vektorius, λ - pasirinkta bauda.

1.4. Gilusis mokymas

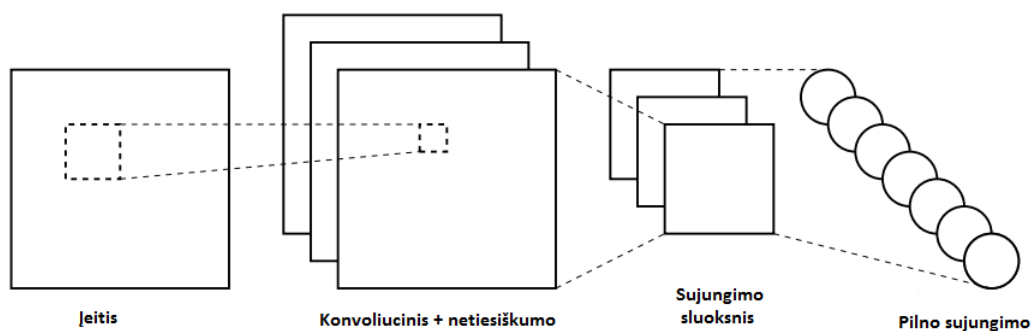
Gilusis mokymas (angl. *deep learning*) arba gilusis struktūrizuotas mokymas (angl. *deep structured learning*) nėra tiksliai apibrėžtas, todėl egzistuoja keletas jo apibrėžimų:

- „Mašininio mokymosi (angl. *Machine learning*) poaibis, kuris remiasi algoritmais, skirtais išmokyti kelis duomenų vaizdavimo lygius, kad būtų sumodeliuoti sudėtingi ryšiai tarp duomenų”[DY13].
- „Gilusis mokymasis yra mašininio mokymosi algoritmų rinkinys, kuriais bandoma mokyti keliais lygmenimis, atitinkančiais skirtingus abstrakcijos lygius”[DY13].

Žinoma, yra ir kitų giliojo mokymosi apibrėžimų, bet visi jie turi du esminius aspektus - (1) modeliai susideda iš kelių netiesinio informacijos apdorojimo sluoksnių arba stadijų; ir (2) iš metodų mokymui su priežiūra arba mokymui be priežiūros aukštesniuose ir labiau abstrakčiuose sluoksniuose.[DY13].

1.5. Konvoliuciniai neuroniniai tinklai

Šiame poskyryje bus kalbama apie Konvoliucinius neuroninius tinklus (angl. *Convolutional neural networks*). Konvoliuciniai neuroniniai tinklai - biologijos įkvėptų neuroninių tinklų klasė, kuri sprendžia mokymosi su priežiūra problemą perleisdama X per konvoliucinių filtrų ir paprastų netiesinių funkcijų sluoksnius[ON15]. CNN priklauso „siunčiančių į priekį” neuroninių tinklų klasei. Šio tipo tinklai parodė puikius rezultatus daugelyje mašininio mokymosi problemų. Konvoliucinis neuroninis tinklas susideda iš šių sluoksnių: įeities (angl. *input*), konvoliucinio (angl. *convolutional*), netiesiškumo (angl. *non-linearity*), jungimo (angl. *pooling*), pilnai-sujungto (angl. *fully-connected*). Bendra CNN schema pavaizduota 5 pav.

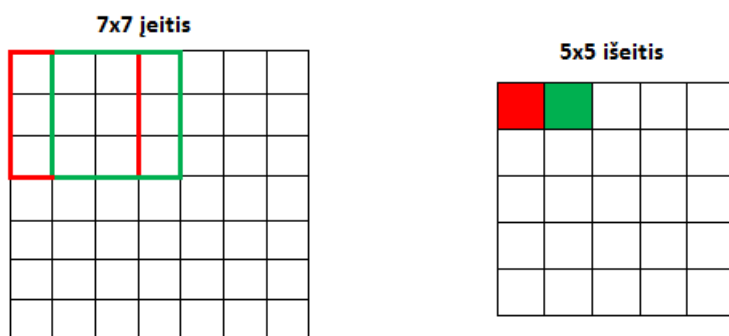


5 pav. Konvoliucinio neuroninio tinklo sandara[Sre].

1.5.1. Konvoliucinis sluoksnis

Konvoliucinis sluoksnis yra esminė sudedamoji CNN dalis. Šis sluoksnis susideda iš apmokomų sudedamųjų dalių arba filtrų. Iš esmės, šio sluoksnio pagrindinė užduotis yra „apibendrinti” duomenis iš tam tikro duomenų rinkinio sluoksnio. Šis apibendrinimas pasiekiamas taikant taip vadinamą konvoliucijos žingsnį (angl. *stride*). Toks žingsnis veikia taip: įeities duomenys ir

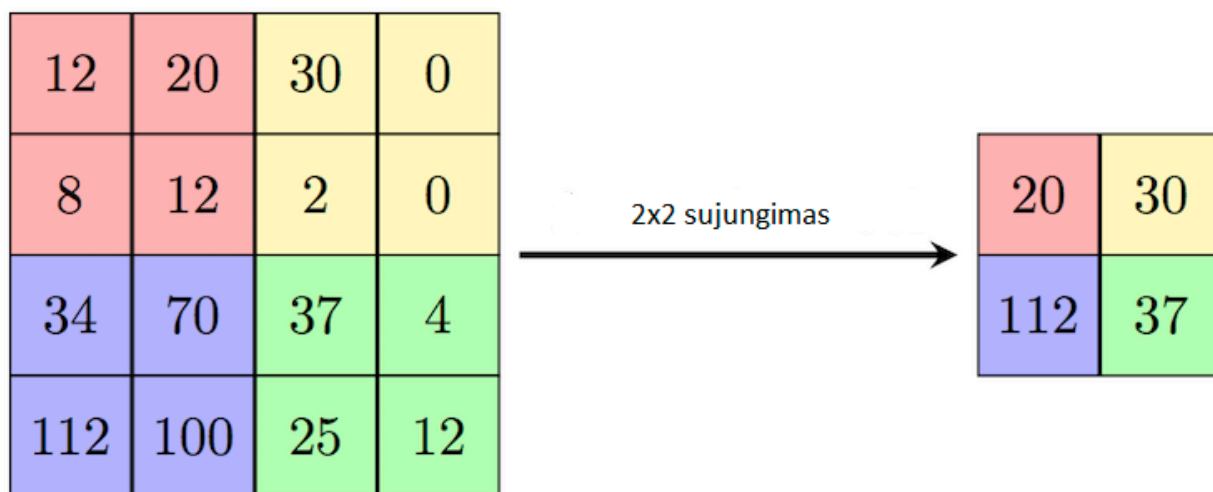
filtrai vaizduojami matricos pavidalu, tuomet filtrai slenka (angl. *convolve*) per duomenų matričą. Pasirinkta duomenų matricos dalis sudauginama su filtru. Gautas rezultatas saugomas išeities duomenyse. Žingsnio veikimas pavaizduotas 6 pav.



6 pav. Konvoliucinis žingsnis[Des].

1.5.2. Sujungimo sluoksnis

Šio sluoksnio funkcija yra progresiškai mažinti erdvinį atvaizdavimo (angl. *representation*) dydį, kad sumažinti parametrų ir skaičiavimų skaičių tinkle ir tokiu būdu kontroliuoti permokymą (angl. *overfitting*)[Kar]. Dažniausiai, sujungimui yra naudojama MAX funkcija. Be MAX tipo sujungimo yra naudojamas ir vidutinis jungimas (angl. *average pooling*). Veikimo principas pavaizduotas 7 pav.



7 pav. Sujungimo sluoksnio veikimo principas[Velb].

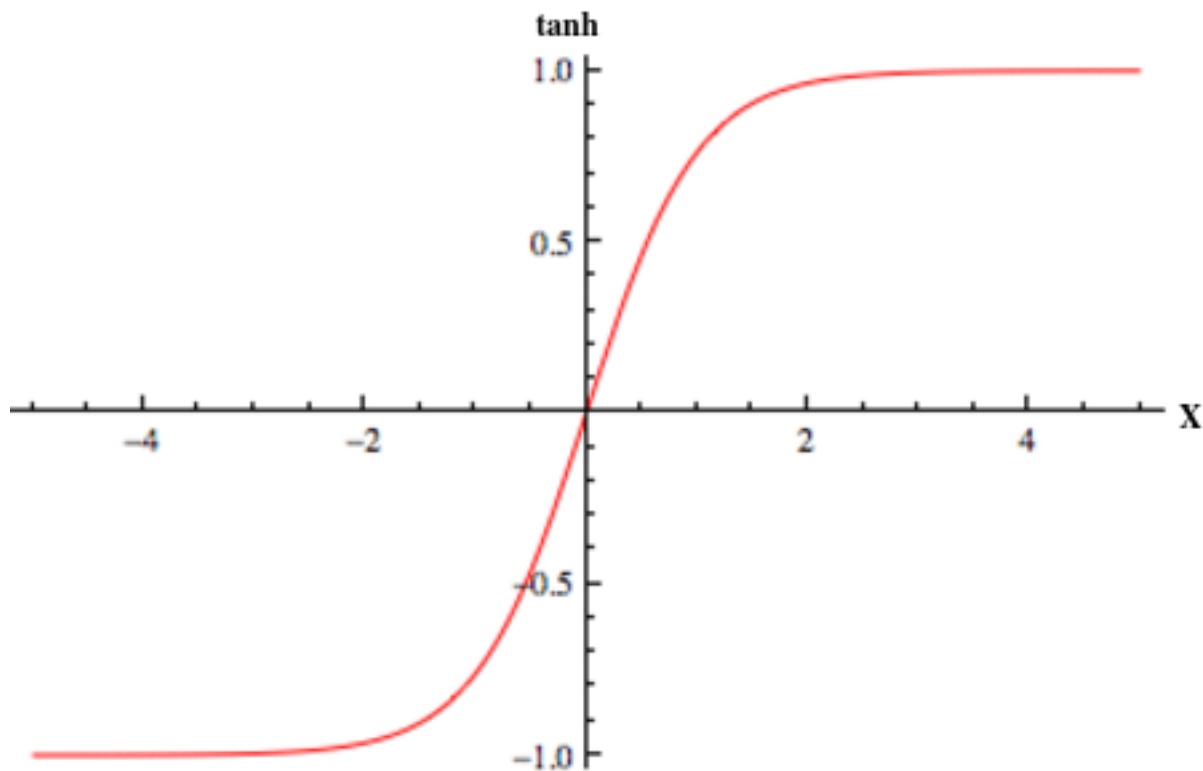
1.5.3. Netiesinis sluoksnis

Kiekvieną konvoliucinį sluoksnį seka papildomas sluoksnis – netiesinis (angl. *non-linear*). Iš esmės – tai tiesiog pasirinkta netiesinė aktyvavimo (angl. *activation*) funkcija. Šis sluoksnis naudojamas įvesti netiesiškumą neuroniniame tinkle[Des]. Dažniausiai netiesiškumui sukurti naudojama funkcija – „Rectified Linear Unit (ReLU)”[NH10]. ReLU veikimas vaizduojamas formule

4.

$$f(x) = \max(0, x). \quad (4)$$

Be ReLU ir jos variacijų („Noisy ReLU“, „Leaky ReLU“), netiesiškumui suteikti naudojamas ir hiperbolinis tangentas \tanh . \tanh „suspaudžia“ realius skaičius į $[-1; 1]$ intervalą. \tanh vaizduojamas 8 pav.



8 pav. Hiperbolinis tangentas[Wei].

1.5.4. Pilno sujungimo sluoksnis

Tarpiniai CNN sluoksniai grąžina daug išskirtų duomenų rinkinio savybių. Šias savybes reikia sudėti į grupes arba klasifikuoti (angl. *classify*). Šis sluoksnis gauna rezultatus iš praeito sluoksnio ir grąžina N dydžio vektorių, N - galimų klasių skaičius. Kitaip tariant, sluoksnis vykdo klasifikatoriaus (angl. *classifier*) funkciją. Šitame sluoksnyje taikomas FNN su tam tikra (dažniausiai SoftMax) aktyvavimo funkcija.

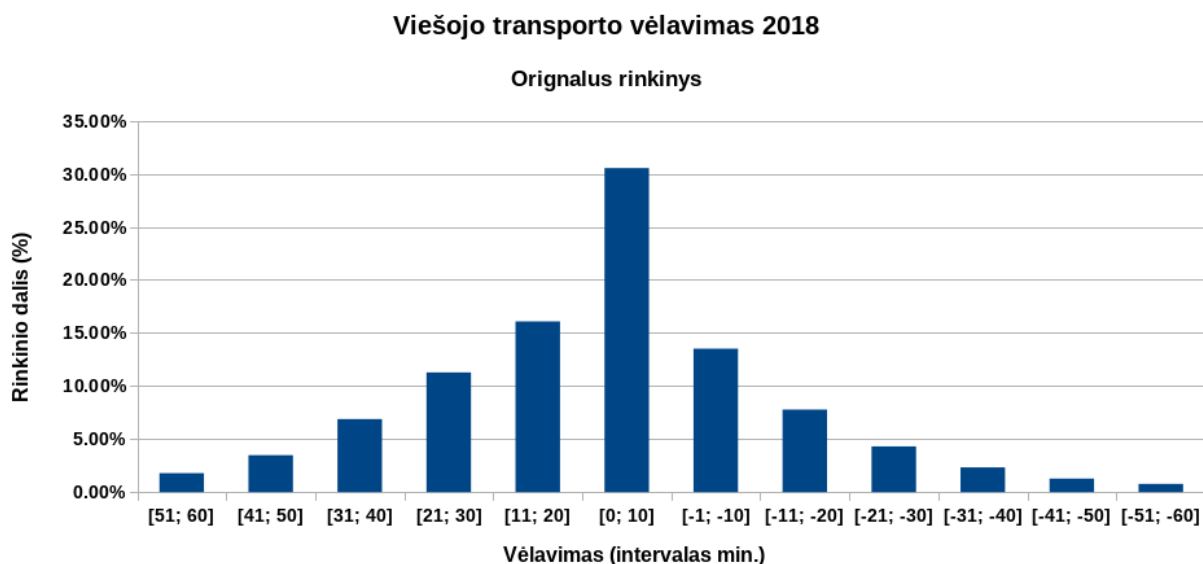
2. Duomenų rinkinys

2.1. Originalus duomenų rinkinys

Eksperimento atlikimui buvo pasirinktas duomenų rinkinys iš Atviro Vilniaus „Atviro duomenų portalo“ [Ope]. Pasirinktas rinkinys – duomenys apie viešojo transporto vėlavimus. Duomenys repozitorijoje yra skirstomi pagal dieną. Esminiai įrašo elementai:

- Kelionės identifikatorius (RunID).
- Transporto priemonės numeris (GPSNum).
- Garažo numeris (GarNum).
- Kelionės tipas – iš A į B, ar atvirkščiai.
- Pavadinimas – maršruto pavadinimas.
- Planuotas kelionės pradžios laikas (planTripStartTime).
- Kelionės pradžios laikas (tripStartTime).
- Kiek pavėluota į sekančią stotelę.
- Sekančios stotelės pavadinimas.

Tačiau eksperimento atlikimui reikalingi tik du stulpeliai iš viso rinkinio: transporto priemonės numeris (GPSNum) ir kiek pavėluota į sekančią stotelę (ahead). „ahead” stulpelio reikšmės buvo suapvalintos iki artimiausio sveikojo skaičiaus. Eksperimento atlikimui buvo paimti duomenys nuo 01-01-2018 iki 2018-05-07. Sujungus visus duomenis apie pasirinktą laikotarpį gavosi apie 290000 įrašų turintis rinkinys. Duomenų rinkinio sudėtis vaizduojama šia diagrama:

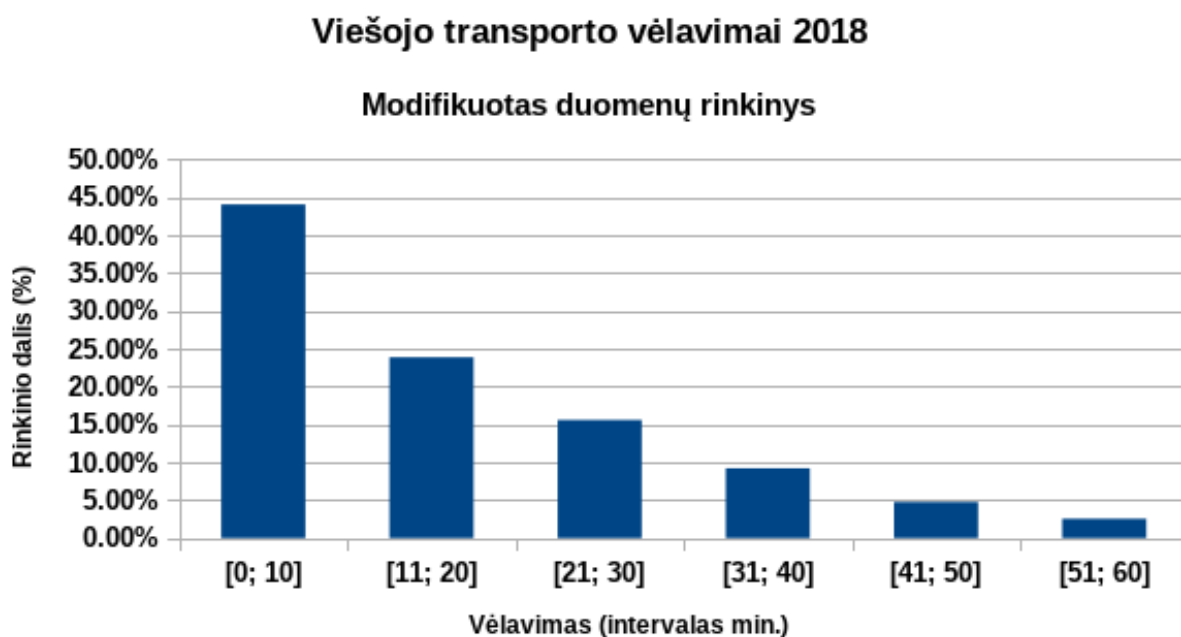


9 pav. Originalus duomenų rinkinys. Viena kategorija reiškia 10 minučių intervalą.

Šioje diagramoje, teigiamos kategorijos reiškia viešosios transporto priemonės vėlavimą, o neigiamos reiškia, kad viešasis transportas atvažiavo per anksti.

2.2. Paruoštas duomenų rinkinys

Originalus duomenų rinkinys savo pirminiame pavidale netinka eksperimento atlikimui – „Spotlight”[Kul17] bibliotekoje siūlomos CNN realizacijos priima specifinius duomenų rinkinių tipus: kiekvienas rinkinio įrašas turi susidėti iš (1)vartotojo id, (2)daikto (klasės) id ir (3)laiko žymės UNIX formatu. Be to, kiekviena reikšmė duomenų rinkinyje turi būti teigiama. Visoms neigiamoms „ahead” stulpelio reikšmėms iš praeito duomenų rinkinio buvo pritaikyta modulio funkcija, reikšmės buvo suapvalintos iki sveikąjį skaičių ir 0 pakeistas į 1. Kiekvienam naujo duomenų rinkinio įrašui buvo pridėta UNIX formato laiko žymė, reiškianti dieną, kurią buvo surinktas įrašas. Gauto duomenų rinkinio sudėtis:



10 pav. Modifikuotas duomenų rinkinys. Viena kategorija reiškia 10 minučių intervalą.

3. Statistinės analizės atlikimas neuroninio tinklo pagalba

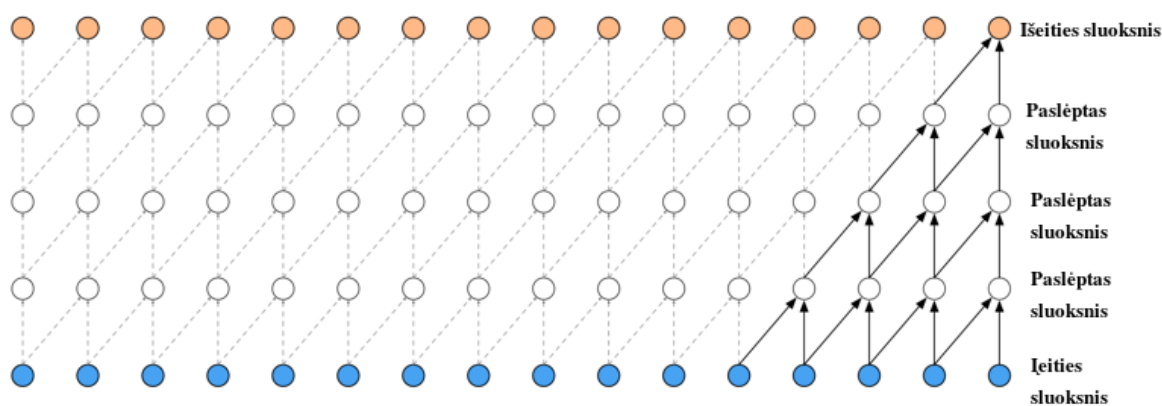
Šiame skyriuje aprašomas statistinės duomenų analizės atlikimas konvoliucinio neuroninio tinklo pagalba. Bus kalbama apie naudojamos bibliotekos ypatumus, eksperimento atlikimo eigą bei naudotus neuroninio tinklo parametrus.

3.1. Pradinio taško nustatymas

Dirbtinio konvoliucinio neuroninio tinklo įvertinti buvo nustatyti pradinis taškas (angl. *baseline*). Pradinio taško nustatymui buvo naudoti du klasifikavimo metodai: Naivusis Bajesas (skyrius 1.1.2.1) ir „Random Forest“ (skyrius 1.1.2.2). Duomenų analizė buvo atlikta „Weka“ [FHW16] programos pagalba. Naiviojo Bajeso tikslumas - 7.7073%, „Random Forest“ - 7.7087%.

3.2. Bibliotekos ypatumai

Esminis skirtumas tarp konvoliucinio neuroninio tinklo šioje bibliotekoje ir kitų konvoliucinio tinklo realizacijų yra kitoks naudojamų konvoliucinių sluoksnių tipas. „Spotlight“ [Kul17] bibliotekoje naudojamos taip vadinamos „priežastinės“ (angl. *causal*) konvoliucijos. „Bet rekurentiniai neuroniniai tinklai nėra vienintelis būdas efektyviai vaizduoti sekas: konvoliucijos taip gali susitvarkyti su šiuo darbu. Mes galime panaudoti priežastines konvoliucijas: konvoliuciniai filtrai pritaikomi sekai iš kairės į dešinę, taip išmesdami atvaizdą kiekviename žingsnyje.“ [Kul17]. Iš esmės, tokio tipo konvoliucijos vadinamos „priežastinėmis“ todėl, kad išeitis (angl. *output*) laiko momentu t priklauso išeities laiko momentu $t - 1$. Be to, neuronų sluoksniai sujungiami naudojant tanh (skyrius 1.5.3) ir ReLU (skyrius 1.5.3) netiesiškumą (angl. *non-linearity*). Priežastinių konvoliucijų veikimas parodytas 11 pav.



11 pav. Priežastinių konvoliucijų veikimo principas. [vdODZ⁺16]

Netekties (angl. *loss*) funkcijos - funkcijos, skirtos nustatyti neteisingo neuroninio tinklo spėjimo kainą (angl. *cost*). Šioje bibliotekoje naudojamos netekties funkcijos:

- „Hinge” apibrėžiama formule 5

$$\max(0, 1 - t \cdot y), t = \pm 1. \quad (5)$$

, kur t - ar teisingas klasės spėjimas, y - klasifikatoriaus priskirta reikšmė

- „Adaptive hinge” - „Hinge” variacija. Pirmą, iš anksčiau atliktų neigiamų spėjimų išrenkami turintys didžiausią vertę. Antra, Visiems šiems spėjimams taikoma „Hinge” funkcija.
- „Pointwise” apibrėžime taikoma sigmoidinė funkcija 6

$$f(x) = \frac{1}{1 + \exp(-x)}. \quad (6)$$

Tegul P - teigiami spėjimai, o N - neigiami, $f(X)$ - sigmoidinė funkcija 6. „Pointwise” netektis apibrėžiama formulėje 7:

$$1 - f(P) + f(N). \quad (7)$$

- „BPR” - iš teigiamų ir neigiamų spėjimų sudaromos poros. Kiekvienam iš šių porų skirtumui d taikoma sigmoidinė funkcija 6.

3.3. Eksperimento atlikimas ir tinklo parametrų įtaka tinklo tikslumui

Šiame skyriuje aprašoma pagrindinių neuroninio tinklo parametrų įtaka spėjimo tikslumo rezultatui. Kiekvieno nagrinėjamo parametro rezultatų lentelė aprašoma atitinkamame poskyryje. Eksperimento atlikimo metu buvo keičiami šie neuroninio tinklo parametrai:

- Neuronų sluoksnių skaičius. Reikšmės - nuo 1 iki 4.
- Epochų (angl. *epoch*) skaičius - nuo 1 iki 9.
- Netekties (angl. *loss*) funkcija - „bpr”, „hinge”, „adaptive-hinge”, „pointwise”.
- Netiesiškumas (angl. *non-linearity*) - ReLU arba tanh.
- Konvoliucinių filtrų skaičius sluoksnyje - 8, 16, 32.
- Duomenų partijos (angl. *data batch*) dydis - 64, 128.
- L2 (skyrius 1.3) generalizacija - 0, 0.01, 1.

Remiantis literatūros šaltiniais[Vela][GBC11], svarbiausi neuroninio tinklo parametrai yra: neuronų sluoksnių skaičius, netekties funkcija, duomenų partijos dydis, L2 generalizacija bei netiesiškumas.

3.3.1. Neuronų sluoksnių skaičius

Šiame poskyryje nagrinėjama neuronų sluoksnių skaičiaus įtaka maksimaliam ir minimaliam konvoliucinio neuroninio tinklo spėjimo tikslumui. Neuronų sluoksnių skaičiaus įtaka spėjimo tikslumui:

Sluoksniai	MRR	Netiesiškumas	Netekties f-cija	Partijos dydis	L2
1	0.27986	tanh	hinge	32	1
2	0.28263	relu	hinge	64	1
3	0.28948	relu	adaptive_hinge	64	1
4	0.28001	tanh	adaptive_hinge	64	0

1 lentelė. Didžiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal sluoksnių skaičių.

Iš rezultatų lentelės matyti, kad pasirinktam uždaviniui su pasirinktu duomenų rinkiniu, neuronų sluoksnių skaičius daro teigiamą įtaką kol yra sąlyginai mažas (iki 3). Be to, neuronų sluoksnių skaičius gerai kombinuojasi su „hinge” arba „adaptive-hinge” netekties funkcijomis.

Sluoksniai	MRR	Netiesiškumas	Netekties f-cija	Partijos dydis	L2
1	0.09648	relu	pointwise	128	0.01
2	0.09243	tanh	bpr	64	0.01
3	0.09519	relu	pointwise	128	1
4	0.09305	relu	bpr	128	0

2 lentelė. Mažiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal sluoksnių skaičių.

Minimalaus tikslumo atveju, didesnis neuronų sluoksnių skaičius turi neigiamą įtaką.

3.3.2. Duomenų partijos dydis

Šiame poskyryje nagrinėjama duomenų partijos dydžio įtaka maksimaliam ir minimaliam konvoliucinio neuroninio tinklo spėjimo tikslumui. Duomenų partijos dydžio įtaka spėjimo tikslumui:

Partijos dydis	MRR	Netiesiškumas	Netekties f-cija	Sluoksnių skaičius	L2
64	0.28948	relu	adaptive_hinge	3	1
128	0.28516	tanh	adaptive_hinge	3	0

3 lentelė. Didžiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal duomenų partijos dydį.

Matyti, kad sprendžiamam uždaviniui tinkamas duomenų partijos dydis yra 64 įrašai. Duomenų partijos dydis gerai kombinuojamas su „adaptive-hinge” netekties funkcija.

Partijos dydis	MRR	Netiesiškumas	Netekties f-cija	Sluoksnių skaičius	L2
64	0.09243	relu	bpr	2	0.01
128	0.09305	relu	bpr	4	0

4 lentelė. Mažiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal duomenų partijos dydį.

Naudojant „bpr” netekties funkciją mažesnis duomenų partijos dydis turi neigiamą įtaką tikslumui.

3.3.3. Netiesiškumas

Šiame poskyryje nagrinėjama netiesiškumo (angl. *non-linearity*) įtaka maksimaliam ir minimaliam konvoliucinio neuroninio tinklo spėjimo tikslumui. Netiesiškumo įtaka spėjimo tikslumui:

Netiesiškumas	MRR	Partijos dydis	Netekties f-cija	Sluoksnių skaičius	L2
relu	0.28948	64	adaptive_hinge	3	1
tanh	0.28001	64	hinge	4	0

5 lentelė. Didžiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal netiesiškumo funkciją.

Geriausias rezultatas pasiekiamas naudojant ReLU netiesiškumą. Kaip ir anksčiau nagrinėtų parametrų atvejais, „hinge” ir „adaptive-hinge” suteikia geriausius rezultatus.

Netiesiškumas	MRR	Partijos dydis	Netekties f-cija	Sluoksnių skaičius	L2
relu	0.09305	128	bpr	4	0
tanh	0.09243	64	bpr	2	0

6 lentelė. Mažiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal netiesiškumo funkciją.

Kaip ir netiesiškumo įtakos maksimaliam tikslumui nagrinėjimo atveju, ReLU leidžia pasiekti geresnius rezultatus.

3.3.4. L2 generalizacija

Šiame poskyryje nagrinėjama L2 baudos (angl. *penalty*) (skyrius 1.3) įtaka maksimaliam ir minimaliam konvoliucinio neuroninio tinklo spėjimo tikslumui. L2 baudos įtaka spėjimo tikslumui:

L2	MRR	Partijos dydis	Netekties f-cija	Sluoksnių skaičius	Netiesiškumas
0	0.28001	64	adaptive_hinge	4	tanh
0.01	0.26177	128	hinge	1	tanh
1	0.28948	64	adaptive-hinge	3	relu

7 lentelė. Didžiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal netekties funkciją.

Iš rezultatų matyti, kad esant $L2 = 0.01$ gerokai krenta spėjimo tikslumas. Bendroju atveju, didesnis $L2$ turi neigiamą įtaką spėjimo tikslumui. Taip gali būti dėl duomenų rinkinio sudėties (skyrius 2.2).

L2	MRR	Partijos dydis	Netekties f-cija	Sluoksnių skaičius	Netiesiškumas
0	0.09305	64	bpr	3	tanh
0.01	0.09243	64	bpr	2	tanh
1	0.09572	64	bpr	4	relu

8 lentelė. Mažiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal L2 baudą.

Kaip ir L2 įtakos maksimalaus spėjimo tikslumui nagrinėjimo atveju, mažiausias tikslumas yra esant $L2 = 0.01$.

3.3.5. Netekties funkcija

Šiame poskyryje nagrinėjama netiekties (angl. *loss*) funkcijos įtaka maksimaliam ir minimaliam konvoliucinio neuroninio tinklo spėjimo tikslumui. Netekties funkcijos įtaka spėjimo tikslumui:

Netekties f-cija	MRR	Partijos dydis	Sluoksnių skaičius	Netiesiškumas	L2
adaptive-hinge	0.28948	64	3	relu	1
hinge	0.28263	64	2	relu	0.01
bpr	0.26910	64	1	tanh	0.01
pointwise	0.24030	64	1	relu	0

9 lentelė. Didžiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal netekties funkciją.

Rezultatai rodo, kad geriausias tikslumas pasiekiamas naudojant „hinge” arba jos variaciją - „adaptive-hinge”.

Netekties f-cija	MRR	Partijos dydis	Sluoksnių skaičius	Netiesiškumas	L2
adaptive-hinge	0.12413	128	4	tanh	0
hinge	0.10154	128	3	relu	0.01
bpr	0.09243	64	2	tanh	0.01
pointwise	0.09450	128	4	relu	0.01

10 lentelė. Mažiausią spėjimo tikslumą turinčios neuroninio tinklo konfigūracijos pagal netekties funkciją.

Kaip ir netekties funkcijos įtakos maksimaliam spėjimo tikslumui nagrinėjimo atveju, geriausias tikslumas pasiekiamas taikant „hinge” ir „adaptive-hinge”. Matyti, kad didesnis duomenų partijos dydis turi neigiamą įtaką spėjimo tikslumui.

Rezultatai ir išvados

Darbai atlikti buvo įvykdyti šie etapai: duomenų rinkinio pasirinkimas ir šio duomenų rinkinio paruošimas eksperimentui (skyrius 2.2), eksperimento atlikimas, konvoliucinio dirbtinio neuroninio tinklo parametrų įtakos spėjimo tikslumui analizė (skyrius 3.3). Duomenų rinkinys - Vilniaus viešojo transporto vėlavimai nuo 01-01-2018 iki 07-05-2018. Originaliame duomenų rinkinyje transporto vėlavimo reikšmės buvo suapvalintos iki sveikojo skaičiaus. Norint sumažinti galimų klasių kiekį, kiekvienai gautai vėlavimo reikšmei buvo pritaikyta modulio funkcija (neigiamą vėlavimo reikšmę reikšdavo, kad transportas atvažiavo anksčiau). Norint nustatyti pradžios tašką, „Weka”[FHW16] programos pagalba, gautam rinkiniui buvo pritaikyta pora mašiniame mokyme (angl. *machine learning*) dažnai naudojamų klasifikavimo metodų (skyrius 3.1) - Naivusis Bajesas ir „Random Forest”. Gauti spėjimo tikslumai: Bajesas - 7.7073%, „Random Forest” - 7.7087%. Šio duomenų rinkinio pagalba buvo apmokyta daug „WaveNet”[vdODZ⁺16] architektūros konvoliucinių neuroninių tinklų su skirtingais parametrais - neuronų sluoksnių, iteracijų, konvoliucinių filtrų skaičius, duomenų partijos dydžiu, netiesiškumu, netekties funkcija, L2 generalizacija. Neuroninio tinklo spėjimo tikslumas buvo matuojamas vidutinio abipusio rango (angl. *Mean Reciprocal Rank*) pagalba. Atlikus eksperimentą, buvo atlikta neuroninio tinklo parametrų įtakos neuroninio tinklo spėjimo tikslumui analizė.

Rezultatai

Atlikus darbą buvo pasiekti šie rezultatai:

1. Paruoštas duomenų rinkinys (skyrius 2.2) - duomenys apie Vilniaus viešojo transporto vėlavimus nuo 01-01-2018 iki 07-05-2018.
2. „WaveNet”[vdODZ⁺16] architektūros konvoliucinio neuroninio tinklo pagalba atlikta statistinė duomenų pasirinkto rinkinio analizė ir nustatyta parametrų įtaka neuroninio tinklo spėjimų tikslumui.

Išvados

Atlikus darbą buvo padarytos šios išvados:

1. Parinkus tinkamus parametrus, konvoliucinis neuroninis tinklas spėjimus atlieka gerokai tiksliau (apie 29%) negu spėjant atsitiktinai (apie 8%) arba taikant populiarius mašininio mokymo algoritmus - Naivųjį Bajesą ir „Random Forest” (apie 7.7%).
2. Esant netolygiai pasiskirsčiusiam duomenų rinkiniui (skyrius 2.2), geriausiai tinka „hinge” netekties (angl. *loss*) funkcija ir jos variacijos („adaptive-hinge”).
3. Nagrinėtam uždaviniui spręsti tinka ir nedaug sluoksnių turintys tinklai (iki 3 sluoksnių).
4. Didesnė apsauga (L2 bauda) nuo persimokymo (angl. *overfitting*) teigiamai veikia spėjimo tikslumą realaus pasaulio duomenims.
5. Nagrinėtam uždaviniui spręsti labiau tinka ReLU netiesiškumas.
6. Esant nedideliame duomenų rinkiniui (šiuo atveju, apie 290000 įrašų), reikėtų išlaikyti pakankamai mažą duomenų partijos (angl. *data batch*) dydį.

Šaltiniai

- [ABL⁺17] L. Jason Anastasopoulos, Dhruvil Badani, Crystal Lee, Shiry Ginosar ir Jake Ryland Williams. Political image analysis with deep neural networks:1–56, 2017.
- [BBG⁺16] Ashwin Bhandare, Maithili Bhide, Pranav Gokhale ir Rohan Chandavarkar. Applications of Convolutional Neural Networks. *International Journal of Computer Science and Information Technologies*, 7(5):2206–2215, 2016. URL: <http://ijcsit.com/docs/Volume%207/vol7issue5/ijcsit20160705014.pdf>.
- [BBO17] Anastasia Borovykh, Sander Bohte ir Cornelis W. Oosterlee. Conditional Time Series Forecasting with Convolutional Neural Networks:1–19, 2017. arXiv: 1703.04691. URL: <http://arxiv.org/abs/1703.04691>.
- [CJ13] Paul R Cohen ir David Jensen. Overfitting Explained. (April 2000), 2013. URL: https://www.researchgate.net/publication/2475394_Overfitting_Explained.
- [Des] Adit Deshpande. A beginner's guide to understanding convolutional neural networks part 2. <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>. Tikrinta: 2018-05-10.
- [DY13] Li Deng ir Dong Yu. Deep Learning: Methods and Applications. *Foundations and Trends[®] in Signal Processing*, 7(3-4):197–387, 2013. DOI: 10.1136/bmj.319.7209.0a. arXiv: 1309.1501.
- [FCB⁺14] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, Dinani Amorim ir Dinani Amorim Fernández-Delgado. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014. ISSN: 1532-4435. DOI: 10.1016/j.csda.2008.10.033.
- [FHW16] Eibe Frank, Mark A Hall ir Ian H Witten. The WEKA Workbench. *Morgan Kaufmann, Fourth Edition*:553–571, 2016. DOI: 10.1016/B978-0-12-804291-5.00024-6. URL: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.
- [GBC11] Ian Goodfellow, Yoshua Bengio ir Aaron Courville. *Deep Learning*. 2011, p. 1–800. DOI: 10.1038/nmeth.3707. arXiv: arXiv:1312.6184v5. URL: www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf.
- [Gha04] Zoubin Ghahramani. Unsupervised Learning BT - Advanced Lectures on Machine Learning. *Advanced Lectures on Machine Learning*, 3176(Chapter 5):72–112, 2004. ISSN: 1745-1337. DOI: 10.1007/978-3-540-28650-9_5. arXiv: 1512.00567.
- [Hil] Aaron Hillegass. Neuroninio tinklo struktūra. URL: https://www.bignerdranch.com/assets/img/blog/2016/06/neural_network_diagram.png (tikrinta 2018-05-10).

- [Joh] Stephen Johnson. Rekurentinio neuroninio tinklo struktūra. URL: <https://i.stack.imgur.com/KmrmP.png> (tikrinta 2018-05-10).
- [Kar] Andrej Karpathy. Convolutional neural networks (cnns / convnets). <http://cs231n.github.io/convolutional-networks/>. Tikrinta: 2018-05-10.
- [KBK11] Andrej Krenker, Janez Bešter ir Andrej Kos. Introduction to the Artificial Neural Networks. *European journal of gastroenterology & hepatology*, 19(12):1046–1054, 2011. ISSN: 0954-691X. DOI: 10.1097/MEG.0b013e3282f198a0. URL: <http://cdn.intechopen.com/pdfs-wm/14881.pdf>.
- [KS96] Ben Krose ir Patrick Van Der Smagt. introduction to Neural. *Networks*, (November), 1996. URL: <https://www.infor.uva.es/~teodoro/neuro-intro.pdf>.
- [Kul17] Maciej Kula. Spotlight. <https://github.com/maciejkula/spotlight>, 2017.
- [LK17] Marc Moreno Lopez ir Jugal Kalita. Deep Learning applied to NLP, 2017. arXiv: 1703.03091. URL: <http://arxiv.org/abs/1703.03091>.
- [LMS16] Gustav Larsson, Michael Maire ir Gregory Shakhnarovich. Learning representations for automatic colorization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS:577–593, 2016. ISSN: 16113349. DOI: 10.1007/978-3-319-46493-0_35. arXiv: 1603.06668.
- [MOG18] A. Mozo, B. Ordozgoiti ir S. Gómez-Canaval. *Forecasting short-term data center network traffic load with convolutional neural networks*, tom. 13 numeris 2. 2018, p. 1–31. DOI: 10.1371/journal.pone.0191939.
- [NH10] Vinod Nair ir Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3):807–814, 2010. ISSN: 1935-8237. DOI: 10.1.1.165.6419. arXiv: 1111.6189v1.
- [OK15] Barak Oshri ir Nishith Khandwala. Predicting Moves in Chess using Convolutional Neural Networks, 2015.
- [ON15] Keiron O’Shea ir Ryan Nash. An Introduction to Convolutional Neural Networks. (November), 2015. arXiv: 1511.08458. URL: <http://arxiv.org/abs/1511.08458>.
- [Ope] OpenVLN. Atviro vilniaus ”atvirų duomenų portalas”. <https://github.com/vilnius>. Tikrinta: 2018-05-10.
- [Pre] Preetham. Backpropagation—how neural networks learn complex behaviors. URL: <https://medium.com/autonomous-agents/backpropagation-how-neural-networks-learn-complex-behaviors-9572ac161670> (tikrinta 2018-05-10).
- [Rom15] Lukasz Romaszko. Signal Correlation Prediction Using Convolutional Neural Networks. *JMLR: Workshop and Conference Proceedings*, 46:45–56, 2015.

- [Sre] Rahtanak Sreang. Intro to deep learning. URL: <https://imgs.laptrinhx.com/c071d519-f58d-417e-9bc8-0d45785a6573.png> (tikrinta 2018-05-10).
- [TAB⁺17] Tomasz Trzcinski, Pawel Andruszkiewicz, Tomasz Bochenski ir Przemyslaw Rokita. Recurrent Neural Networks for Online Video Popularity Prediction:1–8, 2017. DOI: 10.1007/978-3-319-60438-1_15. arXiv: 1707.06807. URL: <http://arxiv.org/abs/1707.06807>.
- [vdODZ⁺16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior ir Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio:1–15, 2016. DOI: 10.1109/ICASSP.2009.4960364. arXiv: 1609.03499. URL: <http://arxiv.org/abs/1609.03499>.
- [Vela] Fernando Velasco. Introduction to deep learning part 2: parameters and configuration. <http://www.stratio.com/blog/deep-learning-2-parameters-and-configuration/>. Tikrinta: 2018-05-10.
- [Velb] Petar Veličković. Deep learning for complete beginners: neural network fine-tuning techniques. <https://cambridgespark.com/content/tutorials/neural-networks-tuning-techniques/index.html>. Tikrinta: 2018-05-10.
- [Wei] Eric W Weisstein. "hyperbolic tangent." URL: <http://mathworld.wolfram.com/HyperbolicTangent.html>. Tikrinta: 2018-05-10.
- [Wik] Wikibooks. „siunčiančio į priekį“ neuroninio tinklo struktūra. URL: https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Feed-Forward_Networks#/media/File:Feedforward_network.svg (tikrinta 2018-05-10).

Santrumpos

CNN Konvoliuciniai neuroniniai tinklai (angl. *Convolutional neural networks*). 5–7, 14, 16, 18

DL Gilusis mokymasis (angl. *Deep learning*). 5

FNN "Siunčiantys į priekį" neuroniniai tinklai (angl. *Feed-forward neural networks*). 16

IT Informacinės technologijos (angl. *Information Technology*). 5, 6

ReLU "Rectified Linear Unit". 15, 16, 19, 20, 22, 25

RNN Rekurentiniai neuroniniai tinklai (angl. *Recurrent neural networks*). 5

SA Statistinė analizė (angl. *Statistical Analysis*). 5, 6

tanh Hiperbolinis tangentas . 16, 19, 20