

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Daugiasluoksnio perceptrono ir radialinių bazinių
funkcijų tinklų lyginamoji analizė, besikeičiančio
uždavinio sprendime**

**Multilayer perceptron and radial basis function comparative
analysis in a changing task environment**

Bakalauro baigiamasis darbas

Atliko:	Modestas Kairys	(parašas)
Darbo vadovas:	lekt. Vytautas Valaitis	(parašas)
Recenzentas:	Linas Petkevičius	(parašas)

Vilnius – 2016

Santrauka

Darbe buvo atliekama daugiasluosnio perceptrono ir radialinės bazinės funkcijos lyginamoji analizė, besikeičiančio uždavinio sprendime. Po pirmo eksperimento, kurio tikslas buvo išmatuoti klasifikatoriaus apsimokymo greitį buvo pastebėtas radialinės bazinės funkcijos pranašumas. Radialinė funkcija gavo tikslius rezultatus, kai daugiasluoksnis perceptronas tame klupo. Kitame eksperimente buvo pastebėtas daugiasluosnio perceptrono aukštesnis patikimumas kintant klases skiriančiui atstumui, arba klasių persidengimui. Radialinė bazinė funkcija esant tam tikriems atstumams regresavo. Galiausiai paskutiniame eksperimente pastebėtas radialinės bazinės funkcijos savarankiškumas. Buvo reikalingas struktūrinis pokytis, kas reiškė svorių kiekio keitimą. Radialinė bazinė funkcija savarankiškai susitvarkė su svoriais ir parodė savo lankstumą pokyčiams.

Raktiniai žodžiai: Neuroniniai tinklai, Perceptronas, Daugiasluoksnis perceptronas, Radialinė bazinė funkcija, besikeičiantis uždavinys

Summary

In this work multilayer perceptron and radial basis function comparative analysis in a changing task environment was completed. After the first experiment, which had the goal to measure classifier's training speed, radial basis function was superior. Radial basis function computed accurate results, meanwhile multilayer perceptron failed to do so. In the next experiment the distance between data classes were altered and multilayer perceptron took the edge by performing with higher reliability. Radial basis function even regressed at some distances. Finally in the last experiment we noticed radial basis function self dependence. Change in the structure was needed, which meant change in the number of weights. Radial basis function handled the weights without any supervision and displayed flexibility.

Keywords: Neural networks, Perceptron, Multilayer perceptron, Radial basis function, changing task environment

TURINYS

ĮVADAS	5
1. NEURONINIS TINKLAS	7
2. MOKYMOSI PROCESAI	8
3. PERCEPTRONAS	10
4. DAUGIASLUOKSNIS PERCEPTRONAS	11
4.1. Daugiasluoksnio perceptrono veikimo principas	12
4.1.1. Atvirkštinio skleidimo algoritmas	12
5. RADIALINĖ BAZINĖ FUNKCIJA	14
5.1. Radialinės bazinės funkcijos veikimo principas	14
5.1.1. PSO algoritmas	15
6. EKSPERIMENTAI	16
6.1. Įeities duomenys	16
6.2. Architektūros parinkimas	16
6.3. Apmokymo greitis	17
6.4. Gebėjimas atskirti	18
6.5. ROC kreivė	21
6.6. Eksperimentas su realiais duomenimis	21
REZULTATAI IR IŠVADOS	23
LITERATŪRA	25

Įvadas

Mokslininkai jau kuris laikas yra įkvėpti žmogaus smegenų veikimu. Neuromokslininkas vardu Warren S. McCulloch ir logikas vardu Walter Pitts 1943 metais sukūrė pirmąjį koncepcinį dirbtinį neuroninio tinklo modelį. Savo darbe jie aprašė dirbtinį neuroną, kuris yra neuroninio tinklo mažiausias vienetas galintis priimti įeitį, ją apdoroti ir sugeneruoti išeitį.

Šių dviejų mokslininkų ir visų kitų mokslininkų, kurie vėliau prisidėjo, darbai nebuvo skirti tiksliai apibūdinti kaip dirba biologinės smegenys. Greičiau dirbtinis neuroninis tinklas buvo sukurtas kaip skaičiavimus atliekantis modelis, skirtas įvairioms problemoms spręsti.

Problemų, kurias gali išspręsti neuroninis tinklas, yra platus, bet ne beribis. Atskirti nuotraukoje koks gyvūnas pavaizduotas atrodo juokų darbas, tačiau tai nėra taip paprasta neuroninui tinklui. Kita vertus žmogui būtų sudėtinga klasifikuoti milijoną įrašų, kas nėra sunku neuroninui tinklui.

Neuroniniai tinklai turi aukštą pritaikomumą kasdieninėje veikloje šitaip palengvindami žmonių darbą. Vieni iš populiariausių praktikoje pritaikomų neuroninių tinklų yra daugiasluoksnis perceptronas ir radialinė bazinė funkcija. Šių dviejų tinklų pritaikymo sritis labai panaši, todėl kyla klausimas, kuris tinklas geriau prisitaiko prie kasdienės, nuolat besikeičiančios, aplinkos. Taigi šio **darbo tikslas** yra sulygtinti ir išanalizuoti daugiasluoksnį perceptroną ir radialinę bazinę funkciją besikeičiančio uždavinio sprendime.

Darbo uždaviniai:

1. Išanalizuoti daugiasluoksnio perceptrono ir radialinės bazinės funkcijos veikimo principus ir juos panaudojant suprogramuoti daugiasluoksnį perceptroną bei radialinę bazinę funkciją.
2. Išmatuoti ir palyginti daugiasluoksnio perceptrono bei radialinės bazinės funkcijos apmokymo greitį.
3. Išmatuoti ir palyginti daugiasluoksnio perceptrono bei radialinės bazinės funkcijos gebėjimą atskirti klases.
4. Suprogramuotą daugiasluoksnį perceptroną ir radialinę bazinę funkciją pritaikyti realios problemos sprendimui.

Darbo eiga: darbas buvo pradamas nuo teorinės dalies. Buvo susipažinta su neuroninių tinklų pagrindais tam kad būtų galima aiškiau suprasti tyrinėjamus objektus. Šiame darbe pateikiami bendri ir konkretūs daugiasluoksnio perceptrono ir radialinės bazinės funkcijos veikimo principai. Darbo pabaigoje aprašomi atlikti eksperimentai, užsibrėžti darbo pradžioje. Šių eksperimentų rezultatai padėjo suformuluoti darbo išvadas.

Darbe, norint suprogramuoti daugiasluoksnį perceptroną ir radialinę bazinę funkciją, buvo naudojama C# kalba ir „Visual Studio Express 2015“ kūrimo aplinka. Paveikslėlių piešimui buvo naudojamas grafikų redagavimo įrankis „Inkscape“. Tai labai patogi, nemokama, atviro kodo programa. Jinai plačiai naudojama ir kuriant sudėtingus piešinius ir paprastas diagramas. Tekstas buvo redaguojamas su dokumentų ruošimo sistema „LaTeX“.

Darbui rašyti pagrindinis šaltinis buvo S. Haykino knyga. Ši knyga ko gero populiariausias ir vienas detaliausių šaltinių apie neuroninius tinklus. Neskaitant šios knygos buvo naudotasi „stac-

kexchange“ puslapiu. Jo pagalba buvo išspręsta nemažai smulkių kilusių problemų programuojant. Taip pat buvo naudojamosi „YouTube“ [Har12] paslaugomis, žiūrimos paskaitos apie klasifikatorius ir jų pagalba buvo bandomas suvokti atvirkštinės sklaidos algoritmas. Visi kiti naudoti šaltiniai yra sužymėti Literatūros skyriuje.

Darbas sudarytas iš 6 skyrių. Pirmajame skyriuje pateikiamas bendras aprašas apie neuroninius tinklus, tai funkcionuoja kaip įžanga į temą. Antrasis skyrius skirtas mokymosi procesams aptarti. Mokymosi procesai yra esminė daugiasluoksnių perceptronų, radialinių bazinių funkcijų dalis ir apskritai visų neuroninių tinklų, todėl šis skyrius turi aukštą svarbą. Trečiajame skyriuje pateikiamas paprastojo perceptrono veikimo principas ir struktūra. Tai yra kiekvieno neuroninio tinklo statybinis blokas. Ketvirtame ir penktame skyriuose detalai aptariami mūsų nagrinėjami objektai. Juose aprašomi ir kaip buvo realizuoti šie klasifikatoriai. Galiausiai paskutiniame skyriuje atliekami eksperimentai, kurių pagalba buvo daromos išvados apie atliktą analizę.

1. Neuroninis tinklas

Visi darbai, susiję su neuroniais tinklais, buvo, ir yra, motyvuojami nuo pripažinimo, kad žmogaus smegenys veikia visiškai skirtingai negu tradicinis kompiuteris. Žmogaus smegenys sudėtingas, ne linijinis ir paralelus kompiuteris, arba kitais žodžiais, informacijos apdorojimo sistema. Mūsų smegenys turi gebėjimą manipuluoti jas sudarančiais neuronais, kad šie atliktų tam tikrus skaičiavimus. Tai leidžia mūsų smegenims atpažinti struktūras, įgauti suvokimą ir valdyti motorinius įgūdžius. Smegenų neuronuose atliekami skaičiavimai vyksta daug greičiau negu bet kokiame dabar egzistuojančiame kompiuteryje. Kaip pavyzdį paimkim regėjimą. Tai yra funkcija, kuri suteikia aplinkos reprezentaciją, ir šitaip suteikia mums informaciją, kaip mes galime su ta aplinka elgtis. Detaliau pasakius, smegenys be sustojimo atlieka atpažinimo ir suvokimo užduotis (pavyzdžiui, atpažinti pažįstamą veidą nepažįstamoje aplinkoje) per apytiksliai 100-200 milisekundžių, tuo tarpu galingas kompiuteris mažesnio sudėtingumo užduotis atlieka daug ilgiau [Hay09]. Kitam pavyzdžiui paimkim šikšnosparnio sonarą. Sonaras yra ekolokacijos sistema, kuri šikšnosparniui leidžia nustatyti taikinio greitį, dydį, azimutą ir aukštį. Visi atlikti sudėtingi neuroniniai skaičiavimai interpretuojant gautą aidą nuo taikinio yra atliekami smegenyse, kurios nėra didesnės už slyvą. Tokia ekolokacijos sistema yra geresnė negu bet koks žmogaus sukurtas radaras ar sonaras [Hay09].

Kyla klausimas - kaip smegenims tai pavyksta? Smegenys nuo pat gimimo turi gebėjimą mokytis iš patirties. Patirtis yra renkama palaipsniui, bėgant laikui, smegenys pagal tai sudarinėja taisykles, kurias vėliau taiko. Didžioji dalis žmogaus smegenų vystimosi įvyksta per pirmuosius du metus, tačiau smegenų tobulėjimas ties tuo nesustoja, bet vyksta nuolatos.

Tobulėjanti nervų sistema yra sinonimiška plastiškoms smegenims. Plastiškumas leidžia nervų sistemai prisitaikyti prie supančios aplinkos [Hay09]. Plastiškumas yra svarbus neuronų funkcionalumui žmogaus smegenyse, taigi natūralu, kad jis taipogi svarbus ir neuroninius tinklus sudarančių dirbtinių neuronų funkcionalumui. Bendrai paėmus, neuroninis tinklas yra mašina, kuri yra pritaikyta imituoti žmogaus smegenis atliekant įvairias užduotis ar funkcijas [Hay09]. Neuroninis tinklas paprastai yra realizuojamas naudojant elektroninius komponentus arba tiesiog yra simuliuojamas tam tikroje programinėje įrangoje. Detalesnis neuroninio tinklo apibrėžimas skambėtų taip - neuroninis tinklas yra procesorius, kurį sudaro paprasti apdorojimo vienetai, turintys polinkį saugoti iš patirties įgautas žinias ir leidžiantys tas žinias panaudoti [Hay09]. Neuroninis tinklas panašus į žmogaus smegenis šiais dviem aspektais:

1. Žinios yra įgaunamos naudojantis mokymosi procesus [Hay09].
2. Tarpneuroninių sujungimų jėgos, žinomos kaip sinapsiniai svoriai, yra naudojamos saugoti įgautas žinias [Hay09].

2. Mokymosi procesai

Viena iš įdomiausių neuroninių tinklų gebėjimų yra mokymasis. Nors ir mūsų supratimas kaip veikia žmogaus smegenys yra menkas ir nežinom tiksliai kaip jos atlieka mokymosi procesą, tačiau turim bent bendrą suvokimą kaip tai vyksta. Teigiama, kad vykstant mokymosi procesui yra keičiama smegenų neuroninė struktūra. Struktūra keičiama, kai yra didinama ar mažinama sinapsinių jungčių jėga, priklausomai nuo veiklos. Dėl šios priežasties svarbesnė, dažniau naudojama, informacija yra lengviau atsimenama negu kad informacija, kuri retai naudojama. Svarbesnė informacija turės stipresnes sinapsines jungtis, tuo tarpu mažiau svarbios informacijos sinapsinės jungtys per laiką silpnės. Neuroniniai tinklai šį procesą imituoja, nors ir labai supaprastintai, keičiant neuronus jungiančių svorių reikšmes. Žmogus norėdamas mokytis iš jį supančios aplinkos turi keletą skirtingų metodų. Neuroniniams tinklams ši taisyklė taipogi galioja. Mokymosi procesai, pagal kuriuos neuroniniai tinklai gali mokytis iš aplinkos, skirstomi į šias pagrindines kategorijas [Jac14] [Hay09]:

1. Mokymasis su mokytoju.
 - (a) Mokymasis duomenų paketais.
 - (b) Mokymasis imant po vieną duomenį.
2. Mokymasis be mokytojo.
3. Sustiprinamasis mokymasis.

Mokymasis su mokytoju yra kai neuroninio tinklo algoritmas priima ne tik įeities duomenis, bet ir norimos išeities. Pateikus šių dviejų tipų duomenis galima apskaičiuoti išeitį su turima įeitim ir patikrint ar gauta išeities buvo toli nuo norimos. Pagal šią reikšmę neuroninis tinklas gali daryti išvadas ir atitinkamai keisti savo svorius [Jac14].

Mokymasis su mokytoju taipogi turi dar du mokymosi pogrupius [Hay09]. Vienas iš tų pogrupių yra mokymasis paketais (angl. batch learning). Šis mokymosi pogrupis naudoja parinktą parametą nuo kurio priklauso paketų dydis. Formuojamas naujas paketas ir duodamas neuronui tinklui treniruotis. Pagrindiniai šio mokymosi pogrupio minusai [Hol13]:

1. Reikia nustatyti paketo dydžio parametą.
2. Reikia pratrinti senus treniruočių duomenis tam kad padaryti vietos naujiems.
3. Negali mokytis iš naujausių duomenų pavyzdžių kol neužsipildė visas paketas.

Kadangi reikia ištrinti aptreniruotas reikšmes, tai šis mokymosi pogrupis mažina tinklo galimybę įgauti bendrą nagrinėjamo duomenų modelio supratimą. Taipogi negalėjimas mokytis iš naujausių duomenų pavyzdžių neigiamai paveikia metodo gebėjimą reaguoti į naują sąvoką [Hol13].

Mokantis imant po vieną duomenį, šis pogrupis yra pranašus tuo, kad jis nieko nelaukdamas mokosi iš kiekvieno naujo atkeliavusio duomens. Dėl šios priežasties šis mokymosi pogrupis yra dažniau pasirenkamas negu, kad mokymasis su paketais, tačiau net ir šis turi minusą, t.y. reikalauja didelio kiekio duomenų, kad galėtų efektyviai apsimokyti [Hol13].

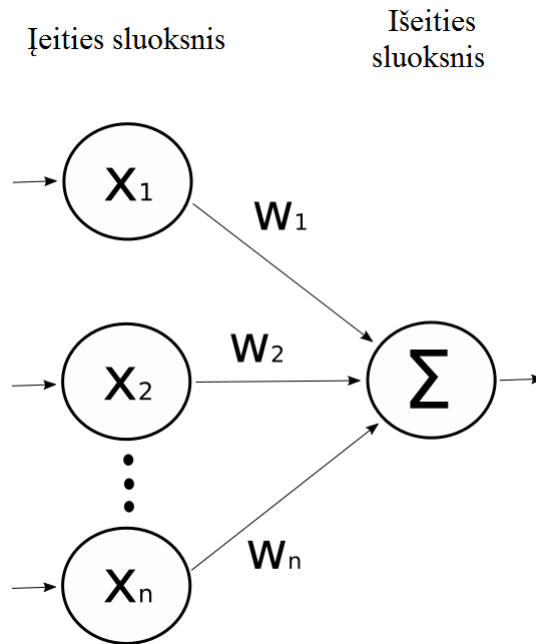
Mokymasis be mokytojo yra kai neuroninio tinklo algoritmas priima įeities duomenis ir bando tuose duomenyse rasti sąsają tarp jų. Tokio tipo mokymasis yra dažnai naudojamas duomenų

gavyboje (angl. data mining) ir taip pat dažnai naudojamas įvairiuose rekomendacijų algoritmuose, kur bandoma nuspėti naudotojo tendencijas, naudojant panašių naudotojų tendencijas [Jac14].

Sustiprinamasis mokymasis yra panašus į mokymą su mokytoju, nes gaunamas atgalinis ryšys (angl. feedback), tačiau vietoj norimos reikšmės atgalinis ryšys suteikia įvertinimą ar gerai neuroninis tinklas atliko darbą. Sustiprinamojo mokymosi esmė yra maksimizuoti suteikiamus teigiamus įvertinimus [Jac14]. Šis mokymosi tipas yra panašus į mokymą vykstantį gamtoje, gyvūnai geriau atsimena veiksmus, kurie jiems padėjo gauti maisto.

3. Perceptronas

Perceptrono idėjos autorius - psichologas Frenkas Rosenblatas. Perceptronas yra paprasčiausia neuroninio tinklo forma, kuri yra naudojama nepersidengiančių struktūrų (besirandančių priešingose hiperplokštumos pusėse) klasifikavimui. Perceptroną sudaro vienas neuronas, kuris turi kintančius sinapsinius svorius ir poslinkį. Perceptronas sudarytas iš vieno neurono turi limituotas galimybes, jis gali atlikti klasifikavimą tik tokios struktūros, kuri yra sudaryta iš dviejų klasių. Išplėsdami perceptroną, pridėdami jam daugiau neuronų, mes atitinkamai praplečiam perceptrono galimybes, tampa galimas klasifikavimas su daugiau nei dviem klasėmis [Hay09].



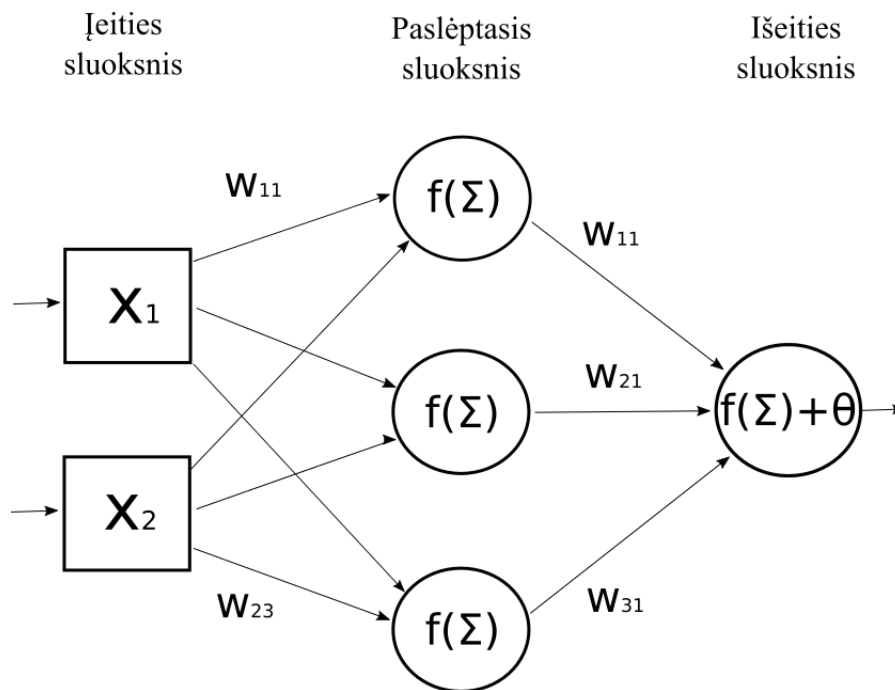
1 pav. Perceptronas

Kaip parodyta 1 pav., perceptrono įeities gali būti multidimensinė, pvz. įeitis gali būti vektorius, mūsų atveju įeitis yra $x = (x_1, x_2, x_3, x_4, \dots, x_n)$. Įeities sluoksnio taškai yra prijungti prie išeities sluoksnyje esančio taško. Šiame taške apskaičiuojama visų įeičių, padaugintų iš svorių $(w_1, w_2, w_3, w_4, \dots, w_n)$, suma. Sumos formulė - $\sum_i^n x_i w_i$. Perceptronas, pagal taisyklę, turi tam tikrą skaičių t , kuris funkcionuoja kaip riba. Jeigu $\sum_i^n x_i w_i \geq t$ tada $y = 1$. Priešingu atveju, jeigu $\sum_i^n x_i w_i < t$ tada $y = 0$. Perceptronas paprasčiausiai įeitį suskirsto į dvi kategorijas, arba kitais žodžiais - brėžia liniją [Hum02].

4. Daugiasluoksnis perceptronas

Daugiasluoksnis perceptronas yra neuroninių tinklų struktūra, kurią sudaro perceptronas su daugiau nei vienu neuronu. Daugiasluoksniui perceptronui būdingi pagrindiniai bruožai:

- Kiekvieno tinkle esančio neurono modelį sudaro nepersidengianti aktyvavimo funkcija, kuri yra atskirtina [Hay09].
- Tinklą sudaro vienas arba daugiau sluoksnių, kurie yra paslėpti nuo įėjties ir išeities taškų [Hay09].
- Tinklas turi aukšto lygio sujungiamumą, kurį nusako tinklo sinapsiniai svoriai [Hay09].



2 pav. Daugiasluoksnis perceptronas

Daugiasluoksnį perceptroną pavaizduotą 1 pav. sudaro šios dalys:

- **Įeities sluoksnis** yra vieta, kur paduodama tinklo įeitis. Įeities sluoksnio neuronų kiekis priklauso nuo to, kiek mes norime, kad mūsų tinklas turėtų įeičių [Kar09].
- Vienas arba daugiau **paslėptųjų sluoksnių**. Diagramoje pavaizduotas tik vienas paslėptasis sluoksnis, tačiau šio tipo sluoksnių skaičius yra kintamas. Paslėptąjo sluoksnio paskirtis yra užkoduoti paduotą jam įeitį ir paskirtį į toliau einančią išeitį. Yra įrodyta, kad daugiasluoksnis perceptronas tik su vienu paslėptuoju sluoksniu gali apskaičiuoti bet kokią funkciją jungiančią įeities ir išeities sluoksnius, su ta sąlyga, kad tokia funkcija egzistuoja. Todėl daugumai problemų išspręsti galima išsiversti tik su vienu paslėptuoju sluoksniu [Kar09].
- **Išeities sluoksnis** yra vieta, kur matomas tinklo rezultatas. Išeities sluoksnio neuronų kiekis priklauso nuo uždavinio, kuri norime, kad mūsų neuroninis tinklas išspręstų [Kar09].

Daugiasluoksnis perceptronas žymiai skiriasi nuo paprasto perceptrono. Ta pati, juos siejanti, dalis yra svorių atsitiktinis parinkimas. Visiems svoriams yra duodamos atsitiktinai parinktos reikšmės iš tam tikro intervalo, paprastai tas intervalas yra $[-0.5, 0.5]$ [Kar09].

4.1. Daugiasluoksnio perceptrono veikimo principas

Daugiasluoksnio perceptrono išeities apskaičiavimas veikia panašiu principu kaip ir perceptronas. Kiekvienam sluoksniui mes apskaičiuojame kiekvieno, jame esančio, neurono reikšmę pamatant visas į tą neuroną einančias reikšmes, jas padauginam iš atitinkamų svorių ir kartu sudedame [Hay09]. Šitokiu būdu mes po truputį judame link išeities sluoksnio ir sukuriame išeities reikšmes. Kaip ir perceptrone, tos reikšmės po pirmos iteracijos paprastai yra gan toli nuo norimos reikšmės. Svarbu pabrėžti tai, kad daugiasluoksniame perceptrone pradiniai svoriai turi nemažą svarbą. Net ir aukšto sudėtingumo mokymosi procesai negali kompensuoti žalos, kai yra parenkami netinkami pradiniai svoriai ar poslinkiai. Tuo tarpu su gerais parinktais svoriais ir paprastas mokymosi algoritmas sugeba efektyviai minimizuoti klaidą [Vra08] [McC12].

4.1.1. Atvirkštinio skleidimo algoritmas

Daugiasluoksnio perceptrono naudojančio atvirkštinio skleidimo algoritmą kiekvieną iteraciją sudaro 2 etapai [McC12]:

1. Išeitis apskaičiuojama naudojant gautus įeities duomenis, svorius ir poslinkius.
2. Pasitelkiamas atvirkštinio skleidimo algoritmas tam kad pakeisti svorius ir poslinkius.

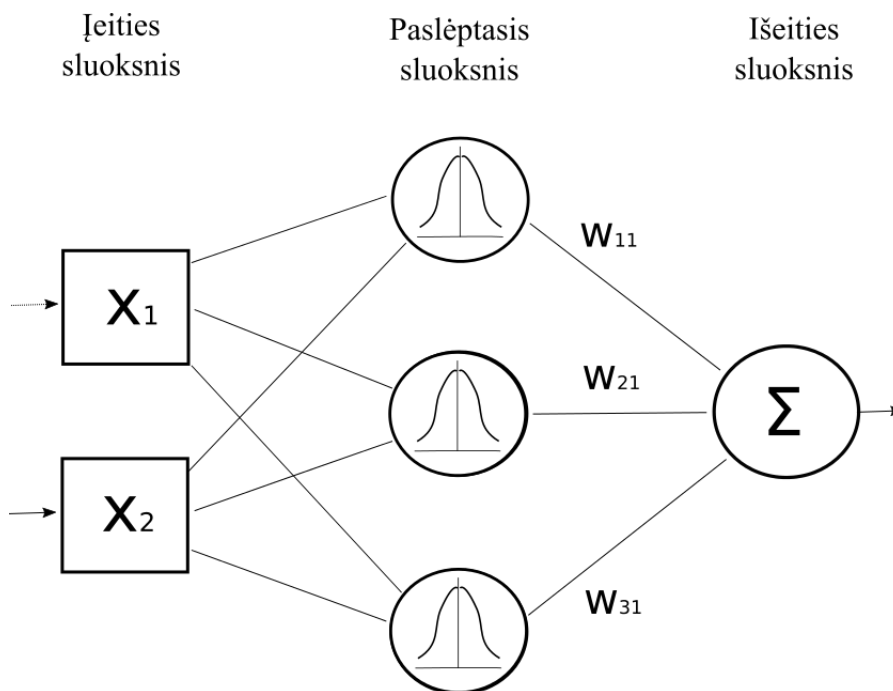
Įeities sluoksnyje pateikti duomenis dažniausiai yra niekaip nekeičiami. Dėl šios priežasties neretai neuroninių tinklų literatūroje šio sluoksnio neuronai yra žymimi kita figūra negu kad kiti neuronai, dažniausiai kvadratu, kaip pažymėta 2 paveikslėlyje. Toliau kiekvienas paslėptojo sluoksnio neuronas apdoroja ateinančius duomenis iš įeities sluoksnio. Kiekvienas paslėptojo sluoksnio neuronas yra sujungtas su kiekvienu įeities sluoksnio neuronu. Veikimas yra panašus į perceptrono veikimą, nes tai daug perceptronų sujungtų kartu, tiesiog reikia įsivaizduoti daugiasluoksnio perceptrono kiekvieną paslėptojo sluoksnio neuroną kaip perceptrono išeities neuroną. Perceptrone įeitys sudauginamos su svoriais ir sudedamos, tačiau daugiasluoksniame perceptrone tuo viskas nesibaigia, toliau tai sumai reikia taikyti aktyvacijos funkciją. Mūsų atveju naudosime sigmoidinę funkciją $\sigma(x) = \frac{1}{1 + e^{-x}}$. Egzistuoja kiti pasirinkimai aktyvacijos funkcijai, kita populiari funkcija yra tangentinė, tačiau eksperimentams atlikti buvo pasirinkta sigmoidinė aktyvacijos funkcija. Išeities sluoksnio neuronai panašiai apskaičiuoja savo rezultatą. Vienintelis skirtumas - prieš taikant aktyvacijos funkciją yra pridamas poslinkis, žymimas θ . Pridėjus poslinkį ir pritaikius aktyvacijos funkciją užbaigiamas pirmasis daugiasluoksnio perceptrono naudojančio atvirkštinio skleidimo algoritmą iteracijos etapas [McC12].

Atkeliavus iki išeities sluoksnio toliau yra skaičiuojami gradientai kiekvienam išeities sluoksnio neuronui. Išeities sluoksnyje skaičiuojami kitaip negu paslėptajame sluoksnyje. Šis gradientas yra gaunamas atimant iš norimo gauto rezultato apskaičiuotą rezultatą ir padauginant iš aktyvacijos funkcijos išvestinės į kurią įsistatoma neurono apskaičiuota reikšmė. Toliau šis apskaičiuotas gradientas grįžta tais pačiais svoriais, kurie jungia neuronus. Kitais žodžiais, signalas yra skleidžiamas atgal, arba atvirkščiai. Paslėptojo sluoksnio neuronai gradientą skaičiuos pirmiausia surinkdami gradientus sklindančius atgal iš išeities neuronų. Tie gradientai susidaukina su svoriais, kurie jungia neuronus. Tokiu būdu atsklidusios reikšmės sumuojamos ir dauginamos iš paslėptojo neurono

aktyvacijos funkcijos išvestinės į kurią įstatomas to neurono apskaičiuota išeitis. Šis naujas gautas gradientas vėl siunčiamas atgal per neuronus jungiančius svorius ir jau bus skaičiuojama δ reikšmė. Sudauginę η su gradientu ir jungiama įeities reikšmės gausim svorio δ . η rolė yra kontroliuoti apmokymo greitį. Kuo didesnis η tuo didesnis pokytis δ reikšmėje. Su didele η apsimokyt galima greičiau, bet tuo pačiu kyla rizika prašauti teisingą atsakymą. Turint kiekvieno svorio δ , svoris sujungiamas su savo δ ir gaunamas naujas svoris. Dar vienas parametras pridedantis atvirkštinio sklidimo algoritmui gylio yra inercija. Inercija dauginama su praeitos iteracijos δ ir pridedama prie naujo svorio. Inercijos rolė yra padėti svoriams pasiekti optimaliausią reikšmę [McC12].

5. Radialinė bazinė funkcija

Radialinė bazinė funkcija yra neuroninių tinklų struktūra, kuri naudoja radialinę bazinę funkciją kaip savo aktyvacijos funkciją. Radialinės bazinės funkcijos tinklas yra panašus į daugiasluksnį perceptroną. Kaip ir daugiasluksnis perceptronas, radialinė bazinė funkcija priima vieną ar daugiau įėjčių ir generuoja vieną ar daugiau skaitinių išeičių. Išėjties priklauso nuo įėjties duomenų, svorių, poslinkių. Papildomai radialinė bazinė funkcija dar remiasi centrų ir pločių reikšmėmis. Visai kaip daugiasluksnis perceptronas, radialinės bazinės funkcijos struktūrą tipiškai sudaro trys sluoksniai: įėjties, paslėptasis ir išėjties.



3 pav. Radialinė bazinė funkcija

5.1. Radialinės bazinės funkcijos veikimo principas

Skaičiavimas žinoma prasideda nuo įėjties sluoksnio. Kiekvienas paslėptajame sluoksnyje esantis neuronas gauna visas įėjties sluoksnio reikšmes. Gavę šie reikšmes jie apskaičiuoja savo lokalią išeitį naudojant savo centrą ir plotį. Šios gautos reikšmės keliauja į išėjties neuronus per neuronus jungiančius svorius. Tai reiškia, kad svoriai sudauginami iš atitinkamų svorių. Visos paslėptųjų neuronų išeičių sandaugos sudedamos ir prie sudėties pridedamas poslinkis. Viskam skamba ganėtinai paprasta, bet tikrasis klausimas yra kaip radialinė bazinė funkcija apskaičiuoja centrus, pločius, svorius ir poslinkius [McC13] [Man16].

Pradedant nuo centrų, radialinė bazinės funkcija turi pasirinkti centrus iš įėjties duomenų. Vienas iš metodų tą padaryti yra k -vidurkio algoritmą, kuris iteruodamas skirsto duomenis į grupes. Algoritmui baigus darbą kiekviena grupė turi save reprezentuojančią reikšmę, kuri gali būti naudojama centro vietoje. Kitas būdas parinkti centrus yra atsitiktiniu būdu. Šis metodas yra kiek nepatikimas, nes per dažnai gali būti pasirinktas blogas centras. Mūsų naudojamas centrų parinkimo būdas yra pastarųjų dviejų hibridas. Iš duomenų atsitiktinio parinkimo būdu paimama grupė,

kurią sudaro 4 koordinačių poros. Iš šių koordinačių norime gauti reprezentuojančias reikšmes, todėl nereiktų rinktis reikšmių, kurios yra viena šalia kitos. Taigi skaičiuojami atstumai tarp reikšmių. Atstumai skaičiuojami tik tarp kaimyninių porų, o ne tarp visų įmanomų. Tiems atstumams skaičiuoti naudojamas Manheteno (angl. Manhattan) atstumas. Vadinas jeigu turim pasirinkęs taškus $x = (a, b)$ ir $y = (c, d)$, tai atstumas bus skaičiuojamas $(|a - c| + |b - d|)/2$. Šis centrų parinkimo procesas kartojamas nurodytą kiekį kartų ir rezultate centrais bus parinktas didžiausias vidutinis atstumas tarp pasirinktų grupių. Verta pastebėti tai, kad centrų apskaičiavimas yra mokymasis be prižiūrėjimo. Tai reiškia, kad centrai patogiai ir greitai nustatomi. Norint apskaičiuoti plotius, reikia juos apskaičiuoti kiekvienam paslėptojo sluoksnio neuronui. Atlikti šią užduotį yra nemažai būdų, tačiau mūsų pasirinktas metodas paprastas - suskaičiuoti bendrą plotį visiems paslėptojo sluoksnio neuronams. Bendras plotis lygus vidutiniam Euklido atstumui tarp visų centrų porų [McC13] [Man16].

5.1.1. PSO algoritmas

Nustačius centrus ir plotius, paskutinis žingsnis yra nustatyti svorius ir poslinkius. Programoje naudosime PSO algoritmą (angl. Particle Swarm Optimization). PSO algoritmas yra alternatyvus metodas atvirkštinės sklaidos metodui, kai norima apmokyti neuroninį tinklą. Algoritmas prasideda apsirašant atsitiktinio parinkimo objektą. Jis naudojamas, kad būtų sukuriamos dalelytės su pozicijos ir greičio reikšmėmis. Dalelių pozicijos yra potencialūs sprendimai mūsų uždaviniui surasti geriausią svorį ar poslinkį. Sukurtos dalelytės taip pat bus naikinamos atsitiktiniu būdu. Kiekvienos sukurtos dalelytės objektas turi savo geriausią rastą poziciją, arčiausiai norimos. Taip pat yra globalus kintamasis, kad saugoti absoliučiai geriausią poziciją. Kitas žingsnis yra skaičiuoti klaidas, kaip toli kiekviena dalelytė iki tikslo. Kai visos dalelytės yra šitaip sukuriamos tikrinama ar galima atnaujinti geriausios pozicijos globalųjį kintamąjį. Toliau vyksta dalelių migravimas. Iš pat pradžių dalelytės bus toli nuo tikslo, todėl naudojamas greičio parametras, kuris reguliuoja kaip greita dalelytė artės link tikslo. Šis parametras po kiekvienos iteracijos kinta, nes jis priklauso nuo dalelytės geriausios žinomos pozicijos, dabartinio greičio ir globaliai geriausios pozicijos. Taip sukančios iteracijomis dalelių debesis artėja link tikslo ir randamas optimalus svoris ar poslinkis. Svarbu paminėti dalelių naikinimą. Pagrindinis šios funkcijos tikslas yra neleisti dalelytėms apspisti neoptimalios pozicijos. Šis parametras negali būti per aukštas, nes dalelytės gali per anksti būti sunaikintos dar neradus optimalaus atsakymo ir negali būti per žemas, nes pozicijos keitimas bus labai lėtas [McC13] [Man16].

6. Eksperimentai

Norint palyginti ir išanalizuoti daugiasluoksnį perceptroną ir radialinę bazinę funkciją būtų labai palanku tai atlikti praktiškai. Apsirašyti multislauksnį perceptroną ir radialinę funkciją nėra taip sunku, todėl mes šiame skyriuje bandysime apsirašyti šias neuroninių tinklų struktūras. Apsirašius atliksime eksperimentus, kurių tikslas bus sulyginti struktūras iš praktinės pusės. Nors ir multislauksnio perceptrono ir radialinės funkcijos parametrai nėra visiškai vienodi, tačiau yra panašumų ir pagal juos vadovaujantis lyginsime šiuos du tinklus.

6.1. Įeities duomenys

Įeities duomenis bus generuojami atsitiktinės parinkties būdu, remiantis Box-Muller transformacijos principu. Koordinačių generavimo formulės: $x = \sqrt{-2 \ln x_1} \cos(2\pi r_2)$ ir $y = \sqrt{-2 \ln x_1} \sin(2\pi r_2)$, kur r_1 ir r_2 atsitiktinai sugeneruoti skaičiai intervale $[0;1]$. Šiomis formulėmis sukurtomis koordinatėmis bus manipuluojama, kad sukurti nepersidengiančias ir persidengiančias duomenų klases.

Eksperimentui su realiais duomenimis bus naudojamas Fišerio vilkdagių duomenų rinkinys (angl. Fisher's Iris data set). Duomenis sudaro 3 vilkdagių rūšys. Kiekvieną rūšį sudaro 50 matavimų rinkinių, viso 150 matavimų rinkinių. Vieną matavimų rinkinį sudaro 4 matavimai: taurėlapio ir žiedlapio ilgiai ir pločiai. Šis duomenų rinkinys yra plačiai naudojamas klasifikatoriams testuoti, nes pagal minėtus 4 matavimus galima identifikuoti vilkdagio rūšį.

6.2. Architektūros parinkimas

Prieš pradėdant programuoti reikia nuspręsti kokios struktūros bus neuroniniai tinklai. Kitais žodžiais, kiek ir kokių sluoksnių, bei neuronų prireiks norint ištirti apsibrėžtus duomenis. Kadangi įeities duomenys yra dvimačiai ir juos sudaro dvi įeitys x ir y , tai dėl to mes pasirinksim įeities sluoksnyje naudoti du neuronus.

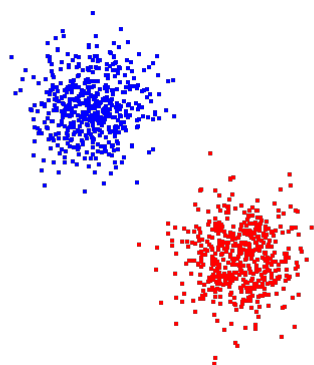
Išeities sluoksniu irgi nesudėtingai apsirendžiamas, kadangi mūsų tikslas duomenis klasifikuoti į dvi klases, tai mums tereikia tik dviejų galimų išeičių. Pasirinkus vieną neuroną išeities sluoksniui tikėsimės iš jo dvejetainės išeities - 0 arba 1. Duomenų paskirstymas nebus idealus, t.y. gautas skaičius bus realus intervale $[0;1]$, todėl naudosime apvalinimo funkciją, kad gauti norimą dvejetainį formatą. Išeities sluoksnyje naudosime vieną neuroną.

Paslėptojo sluoksniu sandara yra diskutuotina. Nėra nustatyta jokių griežtų taisyklių kiek paslėptųjų sluoksnių, ir kiek juose esančių neuronų, turi įeiti į daugiasluoksnį perceptroną. Optimaliausias paslėptųjų sluoksnių ir juose esančių neuronų skaičius nustatomas atliekant bandymus ir mokantis iš rezultatų, klaidų. Tačiau, kaip anksčiau minėta, dažniausiai vieno sluoksniu pilnai užtenka. Neuronų skaičiui pasirinkti yra siūloma rinktis įeities ir išeities neuronų skaičiaus sumos vidurkį. Pagal šį siūlymą naudosime 2 neuronus [faq14].

6.3. Apmokymo greitis

Apmokymo greitis yra svarbus faktorius norint apmokyti klasifikatorių. Duomenų kiekis, kurį reikia klasifikuoti, dažnai skiriasi. Ne visada yra turimas idealus duomenų kiekis apmokymui. Klasifikatoriaus vienas iš esminių principų yra surasti tam tikrus, norimai idealius ar tenkinančius, parametrus su kuriais būtų galima priskirti duomenims klasę. Ir kuo greičiau tie norimi parametrai surandami, tuo labiau kyla klasifikatoriaus vertė. Šiame eksperimente bus testuojamas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos gebėjimas pasiekti užsibrėžtą klasifikavimo tikslumą. Taip pat duomenys bus sukiojami norint susimuliuoti besikeičiančią aplinką ar uždavinį.

Pagal aprašytas funkcijas susigeneruojame dvi taškų grupes, kurias išdėstome plokštumoje taip, kad jos būtų linijiškai atskiriamos, arba nepersidengiančios. 3 pav. pažymėtus duomenis sudaro viso 10000 taškų arba 500 kiekvienoje klasėje. Klasę indikuoja taško spalva.



4 pav. Nepersidengiančių klasių įeities duomenys

1 lentelė. Klasifikavimo tikslumas

Iteracijos	Daugiasluoksnis perceptronas	Radialinė bazinė funkcija
10	0.500	0.989
50	0.986	0.994
100	0.998	0.999
200	0.999	0.999
500	0.999	1.000

Iš rezultatų akivaizdžiai matomas radialinės funkcijos pranašumas. Išskirtina yra tai, kad radialinė bazinė funkcija sugebėjo apsimokyti gan taikliai vos su 10 iteracijų, padaryta tik 11 klaidų (pav.). Tuo tarpu daugiasluoksnis perceptronas turėjo sunkumų su mažu duomenų kiekiu, jam išvis nepavyko apsimokyti su 10 iteracijų. Tačiau treniruočių iteracijoms kylant natūraliai klasifikavimo taiklumas išsilygino.

Tolesniame eksperimente įeities duomenys yra pasukami nedideliu, 15° , kampų. Šiuo eksperimentu yra keičiama neuroninius tinklus supanti aplinka ir šio eksperimento tikslas yra patikrinti kaip neuroniniai tinklai prisitaiko prie lėtai besikeičiančios aplinkos.

2 lentelė. Klasifikavimo tikslumas pasukus duomenis 15° kampu

Iteracijos	Daugiasluoksnis perceptronas	Radialinė bazinė funkcija
10	0.501	0.999
50	0.993	1.000
100	0.999	1.000
200	0.999	0.999
500	0.999	0.999

Iš 2 lentelėje pateiktų duomenų matome, kad radialinė bazinė funkcija rodė geresnius rezultatus. Jinai prisitaikė su mažesniu kiekiu duomenų, kol daugiasluoksniui perceptronui to padaryti nepavyko.

Galiausiai reikia patikrinti kaip neuronai tinklai prisitaiko prie staigiai kintančios aplinkos. Norint susimuliuoti šį eksperimentą mes duomenis pasuksime 60° kampu.

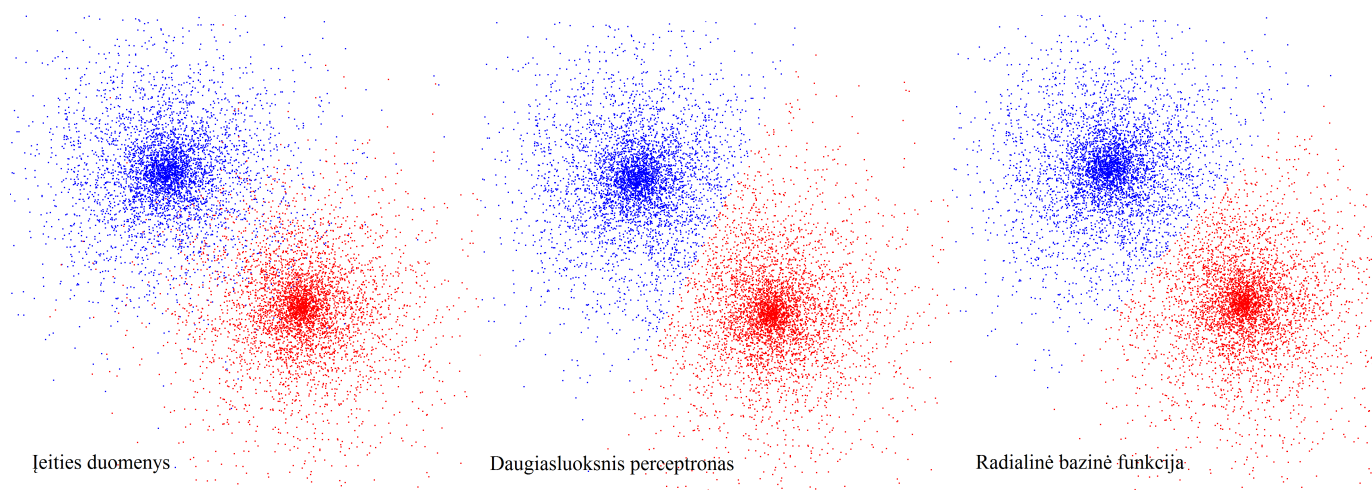
3 lentelė. Klasifikavimo tikslumas pasukus duomenis 60° kampu

Iteracijos	Daugiasluoksnis perceptronas	Radialinė bazinė funkcija
10	0.931	0.971
50	0.987	1.000
100	0.997	0.993
200	0.999	0.999
500	1.000	0.999

Po staigaus aplinkos pakeitimo daugiasluoksnis pasirodė žymiai geriau. Su šiuo duomenų rinkinio pasukimu jis sugebėjo pakankamai tiksliai apsimokyti per pirmas 10 iteracijų. Tačiau radialinė bazinė funkcija vistiek pasirodė geriau.

6.4. Gebėjimas atskirti

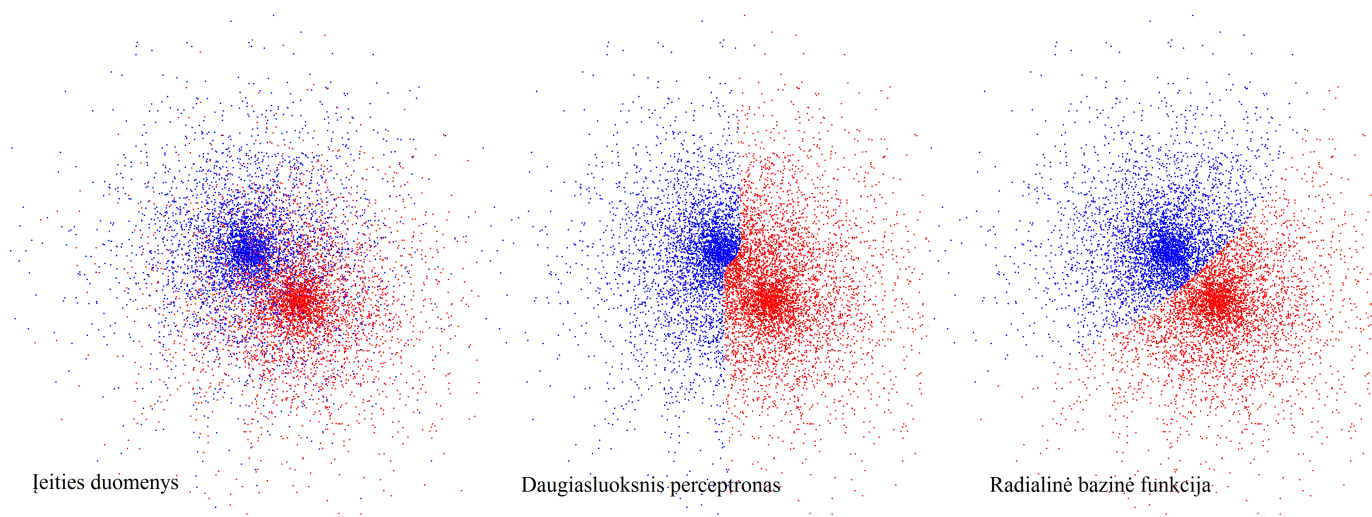
Prieš tai buvusiame eksperimente lyginome kaip mūsų lyginami neuronai tinklai tvarkosi su linijiškai atskiriamos klasėmis. Rezultatai buvo labai aukšti ir ne kartą tinklai atskyrė klases idealiai. Tačiau duomenys ne visada būna tokie paprasti, o kartais persidengia. Įvykstant duomenų persidengimui klasifikatoriaus kyla išsūkis priskirti duomenį teisingai klasei. Taigi šio eksperimento metu yra tikrinamas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos gebėjimas ir tikslumas atskirti persidengiančių klasių duomenis.



5 pav. Klasifikavimo rezultatai su netankiai persidengiančiomis klasėmis

Šiam eksperimentui atlikti pasirinkta padidinti įeities taškų skaičių iki 10000, kad geriau matytumėme liniją, kurią brėžia daugiasluoksnis perceptronas ir radialinė bazinė funkcija. Iš praeito eksperimento rezultatų buvo pastebėta, kad su mažesniais duomenų kiekiais radialinė bazinė funkcija dirba efektyviau, tačiau duodant treniruotėms penktadalį ar pusę įeities duomenų, abu tinklai tuomet beveik susivienodina tikslumu. Darome išvadą, kad geriausias duomenų kiekis aptreniravimui, kuomet nėra per mažai, ar per daug, yra dešimtadalis įeities duomenų.

Pirmam eksperimentui su persidengiančiomis klasėmis buvo duota daugiasluoksniui perceptronui ir radialinei funkcijai atskirti netankiai persidengiančias klases. 4 paveikslėlyje matome, kad klasių centrai aiškiai nutolę vienas nuo kito. Persipynę yra tik klasių taškai, kurie nutolę nuo centro. Abu klasifikatoriai su netankiai persidengiančiomis klasėmis susidoroja gerai, abiejų tikslumas buvo virš 0,96.



6 pav. Klasifikavimo rezultatai su tankiai persidengiančiomis klasėmis

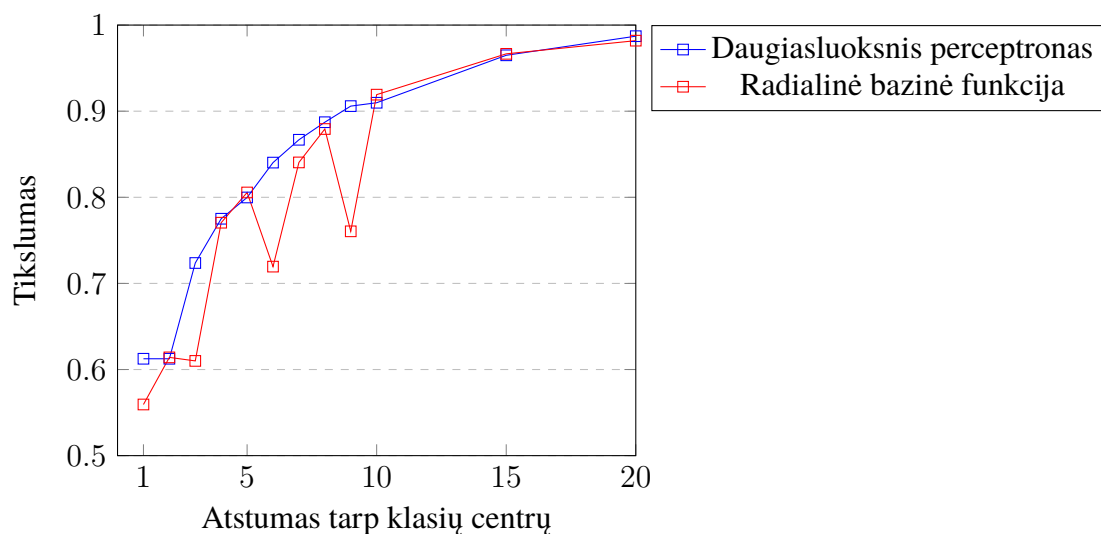
Įdomūs išsiskirimai pradeda matytis su tankiai persidengiančiomis klasėmis. 5 paveikslėlyje matomi skirtumai kokią liniją tarp klasių brėžia daugiasluoksnis perceptronas ir kokią radialinė bazinė funkcija. Svarbu paminėti, kad šitaip persipynusias klases yra labai sudėtinga atskirti, nes

yra didelis kiekis taškų, kurie nutolę nuo savo klasės centro į kitą klasės centrą šitaip klaidindami apmokymą. Todėl esant mažam atstumui tarp klasių tikslumas yra netoli 0,5.

4 lentelė. Klasifikavimo tikslumas esant skirtingo tankio persidengimui

Atstumas tarp klasių	Daugiasluoksnis perceptronas	Radialinė bazinė funkcija
1	0.6125	0.5594
2	0.6852	0.6141
3	0.7237	0.6100
4	0.7751	0.7705
5	0.7998	0.8056
6	0.8403	0.7195
7	0.8668	0.8405
8	0.8872	0.8793
9	0.9059	0.7605
10	0.9098	0.9192
15	0.9649	0.9666
20	0.9871	0.9819

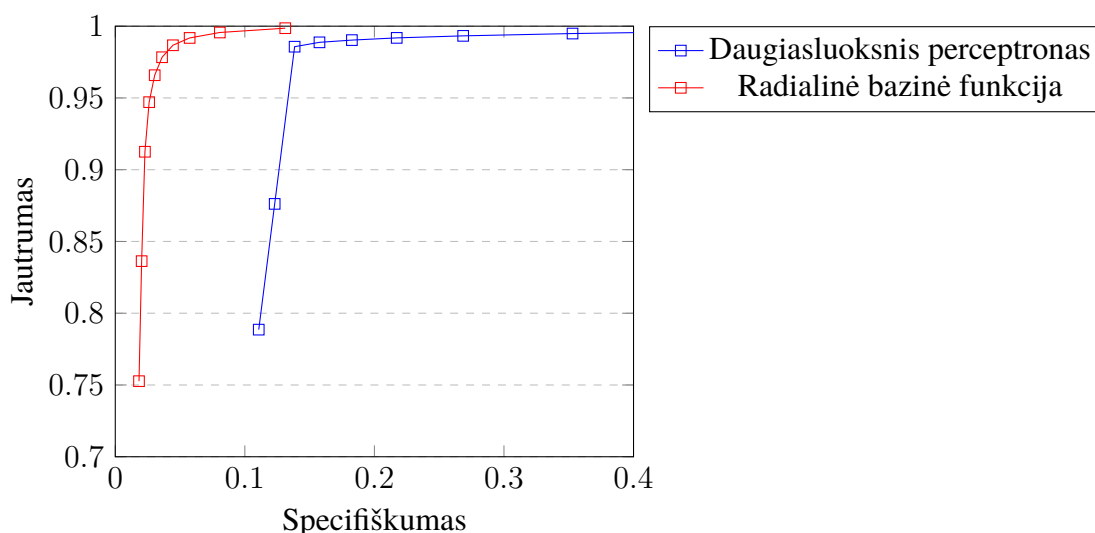
Iš detalių rezultatų pažymėtų 4 lentelėje yra nuo pat pradžių matomas daugiasluoksnio perceptrono pranašumas skirstant duomenis klasėmis. Atstumuose nuo 1 iki 3 daugiasluoksnis perceptronas žymiai tiksliau atliko klasifikavimą ir yra matomas laipsniškas tobulėjimas. Tuo tarpu radialinė bazinė funkcija turi sunkumų atskirti tame atstumų intervale esančias klases. Pagal 6 paveikslėlyje nubrėžtą kreivę iškart pastebimas radialinės bazinės funkcijos nestabilumas. Jinai neatlieka tokio laipsniško tobulėjimo ir didėjant atstumui porą kartų blogiau apsimoko negu kad su tankiau persipynusiomis klasėmis. Galiausiai atstumui tarp klasių esant 10 ir didesniai, klasifikavimo kokybė suvienodėja.



7 pav. Klasifikavimo tikslumas esant skirtingo tankio persidengimui

6.5. ROC kreivė

Dar vienas metodas palyginti daugiasluoksnį perceptroną su radialine bazine funkcija yra ROC kreivė (angl. Receiver operating characteristic). Visų pirma, kas yra ROC kreivė? ROC kreivė yra kreivė parodanti klasifikatoriaus jautrumo ir specifiškumo sąryšį. Šios kreivės principas yra gan paprastas. Kreivė remiasi slenksčio (angl. threshold) parametru ir turimais duomenimis. Jinai pateikia vieną iš dviejų galimų alternatyvių išvadų. Kiekvienai galimai šio slenksčio reikšmei, abiejose ašyse atidedamas teigiamų klasifikatoriaus spėjimų dalis, vertikalioje - teisingų teigiamų, horizontalioje - klaidingų teigiamų. Esant didelėms slenksčio reikšmėms, klasifikatorius beveik nepateikia teisingų spėjimų. Mažinant slenkstį, abiejų spėjimų dalis priartėja prie vieneto. Visiškai atsitiktinai atsakymą spėjančio klasifikatoriaus ROC kreivė bus įstriža nuo taško $[0;0]$ iki $[1;1]$. Klasifikatorius laikomas tuo geresnis, kuo aukščiau jis yra virš įstrižinės. Klasifikatorius laikomas blogu jeigu brėžia liniją žemiau įstrižinės linijos, nes tada klasifikatorius dažniau nustato blogą klasę, negu kad gerą [Hop11].



8 pav. ROC kreivė

ROC kreivei nubrėžimui naudoti duomenys buvo su netankiai persidengiančiomis klasėmis ir apmokymams buvo skirta 1000 iteracijų (dešimtadalis visų įeities duomenis sudarančių taškų). Pastaba, kad 7 paveikslėlyje nupiešta diagrama yra priartinta norint turėti aiškesnį vaizdą. Abi kreivės yra gerokai virš įstrižinės ir tai rodo, kad abu lyginami klasifikatoriai yra efektyvūs ir geriau naudoti juos negu, kad spėlioti klasę atsitiktiniu būdu. Norint nustatyti, kuris klasifikatorius yra geresnis, reikia nustatyti kurio klasifikatoriaus kreivė užima didesnį plotą. Didesnis plotą užima radialinė bazinė funkcija, kas indikuoja šio klasifikatoriaus pranašumą specifiškumo ir jautrumo santykio atžvilgiu.

6.6. Eksperimentas su realiais duomenimis

Šio eksperimento metu bus atliekamas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos darbas su realiais duomenimis, Fišerio vilkdagių duomenų rinkiniu. Eksperimento tikslas

patikrinti kaip klasifikatoriai spęš realią problemą ir kaip gerai prisitaikys prie reikalingo struktūrinio pakeitimo.

Kaip ir anksčiau minėta, Fišerio duomenų rinkinį sudaro matavimo duomenų rinkiniai sudaryti iš 4 matavimų. Visi šie 4 matavimai yra svarbūs norint atpažinti vilkdagio rūšį, todėl bus reikalingas neuroninių tinklų struktūros pakeitimas, t.y. reikės pridėti neuronų. Pradedant nuo įėjties sluoksnio, jį turės sudaryti nebe 2 neuronai, o 4, nes reikia priimti 4 įėjties. Išėjties sluoksniui taip pat reikalingi papildomi 2 neuronai. Anksčiau išėjties klasifikavimui priskirdavo vieną iš dviejų klasių, tačiau dabar reikia rinktis iš 3 klasių. Pridėjus papildomą neuroną išėjties klasės bus koduojamos šiuo susitarimu:

5 lentelė. Klasių išėjties sutartinis kodavimas

Nr.	x_1	x_2	x_3	Pavadinimas
1	1	0	0	<i>Iris setosa</i>
2	0	1	0	<i>Iris versicolor</i>
3	0	0	1	<i>Iris virginica</i>

Galiausiai prie paslėptojo sluoksnio bus pridedamas vienas neuronas. Yra žinoma, kad šio duomenų rinkinio klasės yra atskiriamos naudojant mokymasi be mokytojo. Tai reiškia, kad klasės turi šabloniškumą ir įmanomas klasifikavimas su aukšto lygio tikslumu.

6 lentelė. Klasifikavimo tikslumas klasifikuojant vilkdagių rūšis

Itracijos	Daugiasluoksnis perceptronas	Radialinė bazinė funkcija
5	0.3333	0.6552
10	0.4667	0.6000
15	0.7362	0.9037
25	0.8333	0.8960

Kaip matoma iš 5 lentelėje pateiktų duomenų, radialinė bazinė funkcija buvo pranašesnė kiekvienam etape. Į vieną iteraciją įeina 3 matavimo rinkiniai, po vieną iš kiekvienos klasės. Tai, gi radialinė bazinė funkcija sugebėjo tiksliai klasifikuoti vilkdagių rūšis po 15 iteracijų arba 45 matavimų rinkinių. Pakėlus iteracijas iki 25 tikslumas išliko panašus. Tuo tarpu daugiasluoksnis perceptronas turėjo sunkumų apsimokant ir klasifikuojant. Net su 25 iteracijom nebuvo pavytas radialinės bazinės funkcijos tikslumas. Viena iš priežasčių, kodėl daugiasluoksnis perceptronas nesugebėjo dirbti aukštesniu tikslumu, tai pradiniai svoriai ir poslinkiai. Yra pripažinta, kad daugiasluoksnis perceptronas jautrus pradiniais svoriams ir kad jie turi įtakos rezultatams [Pol90]. Svoriai buvo parinkti atsitiktiniu būdu pradžioje ir vėliau juos buvo bandoma gerinti. Tuo tarpu radialinė bazinė funkcija apskaičiavo savo svorius be priežiūros.

Rezultatai ir išvados

Darbo metu buvo atliekama daugiasluoksnio perceptrono ir radialinės bazinės funkcijos lyginamoji analizė besikeičiančio uždavinio sprendime. Tai buvo pasiekta įvykdžius šiuos uždavinius:

1. Išanalizuotas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos veikimo principas. Šios analizės metu buvo susipažinta su daugiasluoksnio perceptrono ir radialinės bazinės funkcijos struktūra ir veikimo principais. Taipogi buvo susipažinta su atvirkštinės sklaidos algoritmu, kuris yra glaudžiai susijęs su daugiasluoksnio perceptronu, nes atlieka jame mokymosi procesą. Detaliai išanalizuotas PSO algoritmas, kuris yra atsakingas už radialinės bazinės funkcijos apmokymą.
2. Darbo metu buvo suprogramuota daugiasluoksnis perceptronas ir bazinė radialinė funkcija. Šis uždavinys leido praktiškai pritaikyti įgautas žinias ir padėti pagrindą tolesniems uždaviniams.
3. Išmatuotas ir palygintas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos apmokymo greitis.
4. Išmatuotas ir palygintas daugiasluoksnio perceptrono ir radialinės bazinės funkcijos gebėjimas atskirti.
5. Suprogramuotas daugiasluoksnis perceptronas ir bazinė radialinė funkcija buvo pritaikyti realios problemos sprendimui.

Darbo metu, aprašinėjant daugiasluoksnį perceptroną ir radialinę bazinę funkciją, bei atliekant eksperimentus buvo padarytos šios išvados:

1. Pirmo eksperimento metu, kuomet kintanti aplinka buvo išreiškiama per kintantį apmokymams galimų duomenų kiekį, buvo pastebėtas bazinės radialinės funkcijos pranašumas. Ji sugebėjo apsimokyti tik su 10 iteracijų ir bandymą atliko tik su 11 klaidų iš 1000 galimų. Tuo tarpu daugiasluoksniui perceptronui prireikė didesnio iteracijų skaičiaus, kad pasiektų tenkinamą tikslumą. Jam neužteko 10 iteracijų, kad įvykdyti apmokymą aukštu tikslumu. Radialinė bazinė funkcija išlaikė pranašumą, net kai buvo simuliuojamas aplinkos pokytis, t.y. duomenų pasukimas. Prie lėtai besikeičiančios aplinkos prisitaikė net geriau, tačiau verta paminėti, kad prie staigiai besikeičiančios aplinkos daugiasluoksnis perceptronas sugebėjo apsimokyt su 10 iteracijų, kas prieš tai nebuvo jam pavykę. Apibendrinus galima daryti išvadą, kad radialinė bazinė funkcija apsimoko greičiau.
2. Antro eksperimento metu, kuomet kintanti aplinka buvo išreiškiama per kintantį klasių persidengiamumą, buvo pastebėtas nežymus daugiasluoksnio perceptrono pranašumas. Eksperimento metu buvo didinamas atstumas tarp klasių pradedant nuo 1 ir žiūrima kaip klasifikatoriai prisitaikys. 7 paveikslėlyje nubrėžtoje kreivėje matoma, kad kreivės daugelyje vietų sutampa, tačiau tam tikrose vietose radialinė bazinė funkcija suklumpa. Radialinė bazinė funkcija kildama atstumu kartais regresuodavo. Daroma išvada, kad tankiai persidengiančias klases geriau skiria daugiasluoksnis perceptronas, tuo tarpu radialinė bazinė funkcija šioje užduotyje gali suklupti ir rodo nepatikimumą.

3. Brėžiant ROC kreivę buvo pastebėtas radialinės bazinės funkcijos didesnis jautrumas duomenims. Ji sugebėjo su mažu specifiškumu atlikti tikslesnį darbą. Tuo tarpu daugiasluoksnis perceptronas nedaug atsiliko ir jo kreivės užimamas plotas nežymiai mažesnis.
4. Atliekant eksperimentą su realiais duomenimis buvo tiriama ne tik kaip bus išspręsta problema, bet ir kaip klasifikatoriais susitvarkys su struktūros pokyčiais. Radialinė bazinė funkcija žymiai geriau susidorojo su problema ir sėkmingiau paskirstė duomenis į tris klases. Didelė šios eksperimento radialinės bazinės funkcijos sėkmės priežastis yra svorių apskaičiavimas be mokytojo. Šitaip klasifikatorius tampa savarankiškesnis ir lankstesnis. Tuo tarpu daugiasluoksnis perceptronas turėjo sunkumų prisitaikyti, nes pridėti papildomi svoriai buvo atsitiktinai parinkti. Pradiniai svoriai daugiasluoksniui perceptronui yra didelės svarbos parametrai ir norint juos tobulinti reikia mokytojo priežiūros. Šio eksperimento daroma išvada, kad radialinė bazinė yra savarankiškesnė ir struktūriškai lankstesnė.

Šio darbo metu visi užsibrėžti uždaviniai buvo atlikti, todėl galima teigti, kad tikslas buvo pasiektas - gauti daugiasluoksnio perceptrono ir radialinės bazinės funkcijos lyginamosios analizės rezultatai besikeičiančiame uždavinyje. Vis dėlto ši analizė gali būti atliekama dar detaliau. Galimi praplėtimai: palyginti klasifikatorius su skirtingomis aktyvacijos funkcijomis, palyginti klasifikatorius su skirtingais apmokymo algoritmais.

Literatūra

- [faq14] faqs.org. How do mlps compare with rbfs? <http://www.faqs.org/faqs/ai-faq/neural-nets/part2/section-14.html>. 2014.
- [Har12] R. Harris. Neural network tutorial: the back-propagation algorithm. <https://www.youtube.com/watch?v=aVId8KMsUU>. 2012.
- [Hay09] S. Haykin. *Neural networks and learning machines*. Pearson Education Inc., Upper Saddle River, New Jersey, 07458, 2009.
- [Hol13] J. Read; A. Bifet; B. Pfahringer; G. Holmes. Batch-incremental versus instance-incremental learning in dynamic and evolving data. <http://albertbifet.com/wp-content/uploads/2013/10/IDA2012.pdf>. 2013.
- [Hop11] L. Hopley. The magnificent roc. <http://www.anaesthetist.com/mnm/stats/roc/Findex.htm>. 2011.
- [Hum02] M. Humphrys. Single-layer neural networks (perceptrons). <http://computing.dcu.ie/~humphrys/Notes/Neural/single.neural.html>. 2002.
- [Jac14] L. Jacobson. Introduction to artificial neural networks part 2 - learning. <http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-2-learning/8>. 2014.
- [Kar09] L. Karapetsas. Multi-layer perceptron tutorial. <http://blog.refu.co/?p=931>. 2009.
- [Man16] A. P. Markopoulos; S. Georgiopoulos; D. E. Manolakos. On the use of back propagation and radial basis function neural networks in surface roughness prediction. <http://link.springer.com/article/10.1007/s40092-016-0146-x>. 2016.
- [McC12] J. McCaffrey. Neural network back-propagation for programmers. <https://msdn.microsoft.com/magazine/jj658979.aspx>. 2012.
- [McC13] J. McCaffrey. Particle swarm optimization using c#. <https://visualstudiomagazine.com/articles/2013/11/01/particle-swarm-optimization.aspx>. 2013.
- [Pol90] J. F. Kolen; J. B. Pollack. Back propagation is sensitive to initial conditions. <http://www.complex-systems.com/pdf/04-3-2.pdf>. 1990.