

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Edvinas Bernotas

**Finansinių ir fizinių procesų prognozavimo modelių tyrimas ir
taikymas**

Magistro darbas

Darbo vadovas
prof. J.Mockus

Kaunas

2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Edvinas Bernotas

**Finansinių ir fizinių procesų prognozavimo modelių tyrimas ir
taikymas**

Magistro darbas

Recenzentas

2009-05-25

doc. dr. S. Maciulevičius

Vadovas

prof. J. Mockus
2009-05-25

Atliko

2009-05-25

IFM-3/1 gr. stud.
Edvinas Bernotas

Kaunas

2009

1. Įvadas	4
2. Dirbtinių neuronų tinklų ir jų pritaikymo prognozėms analizė	7
2.1. Bendros sąvokos	7
2.2. Dirbtinis neurono modelis	8
2.3. Tinklo modelis	10
2.4. Neuroninio tinklo mokymas	11
2.5. Mokymas ir apibendrinimo savybės	14
2.6. Funkcijos aproksimavimas	15
2.7. Funkcijos aproksimavimas daugiasluoksniu neuroniniu tinklu.....	17
2.8. Klaidų įvertinimas.....	19
2.9. Kiti prognozavimo modeliai	20
2.9.1. ARMA modelis.....	20
2.9.2. AR-ABS modelis	21
3. Naudojamų priemonių išnagrinėjimas ir modelių paruošimas	22
3.1. Įvadas	22
3.2. Biržos žaidimo modelis, Nešo pusiausvyra	23
3.3. Autoregresijos absoliutaus nuokrypio modelis (AR-ABS)	26
3.4. Tiesioginis neuroninis tinklas naudojant MatLAB ir Neural network tool	30
3.4.1. Duomenų apdorojimas	31
3.4.2. Tinklo treniravimas.....	33
3.5. Joone (Java Object Oriented Neural Engine)	36
4. Modelių eksperimentinis tyrimas.....	40
4.1. Išsamesnė peržiūra	44
5. Išvados	52
6. Literatūra.....	53
7. Terminų ir santrumpų žodynas	54

1. ĮVADAS

Prognozavimas yra procesas, kuris pateikia išėjimų aibę, kurie gauti įvedus kintamųjų aibę. Šie duomenys paprastai būna istoriniai duomenys. Iš esmės, prognozavimas paremtas prielaida, kad ateities įvykiai yra visiškai ar bent dalinai paremti anksčiau stebėtais ar praeities įvykiais. Daroma prielaida, kad kažkurie praeities dėsningumai atsikartos ateityje. Praeities ryšiai gali būti atrasti stebėjimo ir tyrimo metu. Pagrindinė prognozavimo idėja – sudaryti įėjimo ir išėjimo duomenų „žemėlapi“ (kuo labiau priartėti prie įėjimo ir išėjimo duomenų paskirstymo), tam kad būtų atrastos numanomos taisyklės, valdančios stebėtus veiksmus. Pavyzdžiui, rinkos kainų prognozavimas gali būti nusakytas tokiu būdu. Tarkime u_i nusako šiandieninę kainą, v_i nusako kainą, kuri bus po dešimties dienų. Jeigu kainos, kuri bus po dešimties dienų, prognozavimas bus atliekamas remiantis šiandienos rinkos kaina, turi būti funkcinis kelias nuo u_i iki v_i , kur $v_i = \Gamma(u_i)$. Naudojantis visomis istorinių duomenų poromis (u_i, v_i) , gali būti išvesta bendra funkcija $\Gamma()$, kuri sudaryta iš $\Gamma_i()$, t.y. $v = \Gamma(u)$. Apskritai, funkcijoje $\Gamma()$ gali būti naudojamas u , kuris turi daugiau informacijos apie esančią kainą. Kadangi NT yra universalūs aproksimatoriai, mes galime surasti NT simuliuojantį šią $\Gamma()$ funkciją. Tada apmokytas tinklas naudojamas nustatyti ateities pokyčius.

NT paremtas finansinis prognozavimas nagrinėjamas jau apie dešimtmetį. Įvairiuose tarptautiniuose žurnaluose ir konferencijų medžiagose išleista daug tiriamųjų straipsnių. Kai kurios kompanijos ir institucijos reiškia teises arba pardavinėja taip vadinamus pažangius prognozavimo įrankius arba modelius.

Siekiant sukurti patikimesnį finansinės prognozės modelį, reikia atsižvelgti į šias problemas:

- Pirma, vietoj tik prognozės tikslumo akcentavimo, reikėtų atsižvelgti ir į kitus finansinius kriterijus. Dabartiniai tyrinėtojai linkę naudotis gerumo (goodness of fit) ar panašiais kriterijais, kad įvertintų ar treniruotų savo modelius finansinėje terpėje. Teoriškai matematinių skaičiavimų požiūriu šis būdas yra teisingas. Realybėje tobulas prognozavimas yra neįmanomas. Nei vienas modelis negali pasiekti tokių idealių rezultatų. Todėl tobulo prognozavimo pasiekimas, nėra esminis tikslas. Yra įmanoma tik optimizuoti netobulus prognozes ir naudotis kitais kriterijais, kad būtų gautas kuo realesnis rezultatas.
- Antra, turi būti atitinkamas prognozavimo duomenų organizavimas ir apdorojimas. Pradinis apdorojimas ir teisingas įvedimo duomenų atrinkimas, gali turėti įtakos prognozavimo vykdymui. Rėmimasis jautrumo analize pasirenkant indikatorius kaip

- Trečia, prekybos sandorių sistemai reikia pasirinkti geriausią įrankį. NT nėra vienintelis įrankis, kuris gali būti naudojamas finansiniam prognozavimui. Taip pat negalima teigti, kad tai geriausias įrankis prognozavimui. Faktiškai, žmonės vis dar nežino kokio tipo laiko eilutės geriausiai tinka NT taikymui. Atlikus prognozavimo analizę, galima gauti modelių ir duomenų serijų tinkamumą. Tada galima nuspręsti, kokie modeliai atitinka kokias laiko eilutes. NT modelių kūrimas ir treniravimas yra bandymų ir klaidų procedūra. Kai kurie tyrinėtojai nelinkę atlikti daugiau bandymų su savo duomenų rinkiniais. Jeigu būtų atrasta sistema, kuri padėtų formalizuoti tokias varginančias tyrimo procedūras, ji būtų labai vertinga finansiniam prognozavimui.

Vietoje tik vieno sėkmingo eksperimento atlikimo, tikimybių ar patikimumo pakopa galėtų būti taikoma ir išėjimams. Duomenys suskaidomi į keletą aibių, tam kad būtų išsiaiškintos konkrečios žinios apie nagrinėjamas laiko serijas. Kaip teigia D.Wolpert ir W.Macready apie savo No-Free-Lunch teoremas, paėmus visų problemų vidurkius, visi paieškos algoritmai veikia vienodai. Tik eksperimentuojant su viena duomenų aibe, gali būti surastas NT modelis kuris veikia geriau už kitus. Tačiau, kitai duomenų aibe, kitas modelis kuris veikia geriau negu NT modelis taip pat gali būti rastas remiantis No-Free-Lunch teoremomis. Kad būtų išvengta tokio modelio kuris veikia geriau už kitus, duomenų aibė suskirstoma į kelis duomenų poaibių. Rekomenduojami NT modeliai yra tie kurie veikia geriau negu kiti, visuose duomenų poaibiuose. Kitaip tariant, tik tie modeliai kurie apjungia pakankamai lokalinių žinių, gali būti naudojami ateities prognozavimui.

Labai svarbu ir būtina pabrėžti šias tris problemas. Kriterijai naudojami moksliniuose tyrimuose (teorijoje) skiriasi nuo kriterijų naudojamų pramonėje (praktikoje). Moksliniuose tyrimuose kartais siekiama 100% tikslumo, tuo tarpu pramonėje paprastai siekiama 60% tikslumo. Taip pat, galutinis praktikų siekiamas tikslas yra pelnas, todėl į pelną orientuotas prognozuojantis modelis gali atitikti jų reikmes.

Kad būtų sukurtas NT prognozavimas reikia daug eksperimentų. Tik vienos rinkos ar tik vienos laiko eilutės testavimas nieko nereiškia. Tai nesukurs stipraus modelio paremto rankiniais, bandymų-ir-klaidų ar specialiais eksperimentais. Reikalingas dar efektyvesnis modelis ne tik vienai rinkai ar vienam laiko periodui. Dėl pramoninių modelių trūkumo ir kadangi akademinio pasaulio tyrimų nesėkmės neskelbiamos, vienas asmuo ar net grupė

tyrėjų negali gauti pakankamai informacijos ar patirties, kad galėtų sukurti gerą prognozavimo modelį. Akivaizdu, kad automatizuota sistema apdorojanti NT modelių kūrimą yra būtina. [2]

Šio darbo tikslas yra sumodeliuoti neuroninį tinklą skirtingomis priemonėmis ir sulyginti su kitais prognozės modeliais. Taip pat ištirti modelių gebėjimą prognozuoti įvairių rūšių duomenis: įvairių sektorių realius istorinius duomenis ir virtualius sugeneruotus duomenis.

2. DIRBTINIŲ NEURONINIŲ TINKLŲ IR JŲ PRITAIKYMO PROGNOZĖMS ANALIZĖ

2.1. Bendros sąvokos

Dirbtiniai neuroniniai tinklai apima informacijos apdorojimo metodus, įkvėptus imituojant ir modeliuojant gyvūnų smegenyse vykstančius procesus. Dirbtinis neuroninis tinklas yra matematinių modelių rinkinys, kurio pagalba bandoma emuluoti įvairias biologinių sistemų savybes. Labiausiai viliojantis atrodo biologinių sistemų gebėjimas mokintis, prisitaikyti ir adaptuotis.

Dirbtiniai neuroniniai tinklai sudaryti iš daugelio skaičiuojančių elementų, kurie tarpusavyje glaudžiai susiję. Šie skaičiuojantys elementai yra panašūs į biologinius neuronus ir vieni su kitais sujungiami įvairaus stiprumo jungtimis, kurios yra analogiškos biologinių neuronų sinapsėms.

Mokymosi metu smegenyse keičiasi jungčių, siejančių neuronus, stiprumas. Tai galioja ir dirbtiniams neuroniniams tinklams. Neuroninių tinklų mokymui naudojami duomenų pavyzdžiai, pagal kurios specialių algoritmų pagalba mokymo metu iteratyviai keičiami jungčių stiprumo koeficientai, vadinami svoriais. Informacija, reikalinga konkretaus uždavinio sprendimui, yra sukaupiama šių svorių reikšmėse.

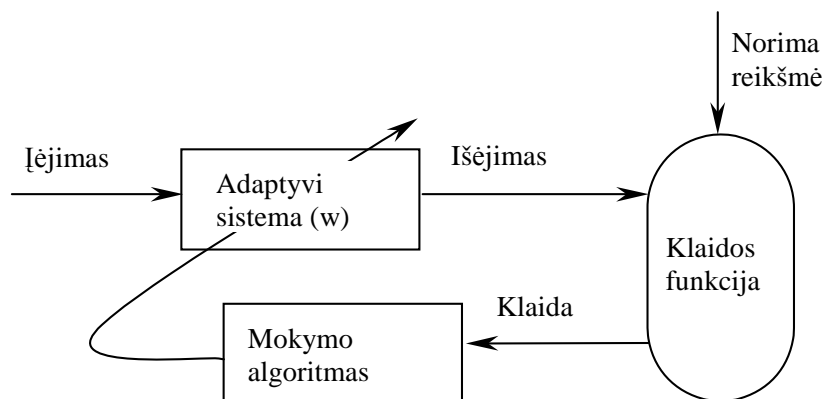
Dirbtiniai neuroniniai tinklai pradėti nagrinėti šeštajame dešimtmetyje, tačiau iki devintojo dešimtmečio vidurio jie nebuvo plačiai naudojami. Tik išradus greitus ir galingus apmokymo mechanizmus, dirbtiniai neuroniniai tinklai galėjo spręsti realius uždavinius.

Šiuo metu neuroniniai tinklai naudojami signalų apdorojimui, triukšmo eliminavimui, duomenų klasifikavimui, sistemų modeliavimui, identifikavimui, prognozei ir kontrolei. Neuroniniai tinklai naudojami daugelyje komercinių produktų, tokių kaip modamai, vaizdų apdorojimo ir atpažinimo sistemos, kalbos atpažinimas, signaliniai procesoriai ir biomedicininė įranga.

Tikimasi, kad artimoje ateityje panašus sąrašas augs eksponentiškai. Neuroninės sistemos, tinkamai jas naudojant, gali pralenkti kitus duomenų apdorojimo, modeliavimo, klasifikavimo metodus.

Neuroninių tinklų adaptyvumas yra geriausioji jų savybė (1 paveikslėlis). Neuroninės sistemos nėra sudaromos naudojant išankstines žinias pagal specifikaciją, formules ar aprašymą. Vietoje to, sistema naudoja išorinius duomenis savo parametrų nustatymui.

Neuroniniai tinklai apmokomi žinant įėjimo ir atitinkamas išėjimo reikšmes (jos dar vadinamos užduoties reikšmėmis), grąžinamas per ryšį, kuriame naudojama klaidos funkcija (1 pav.).



1 pav. Adaptyvios sistemos kūrimas.

Klaidos funkcija labai dažnai yra skirtumo funkcija tarp neuroninio tinklo išėjimo ir užduoties reikšmės. Neuroninio tinklo atsako tikslumas tiesiogiai naudojamas parametru (tinklo svorių) keitimui. Mokymo metu svoriai keičiami taip, kad sistemos išėjimo reikšmės artėtų prie norimų reikšmių (mažėtų klaida).

Neuroninių tinklų parametrai modifikuojami naudojant pasirinktą duomenų rinkinį, vadinamą duomenų imtimi. Naudojant jau apmokytą neuroninį tinklą parametrai paprastai būna fiksuoti.

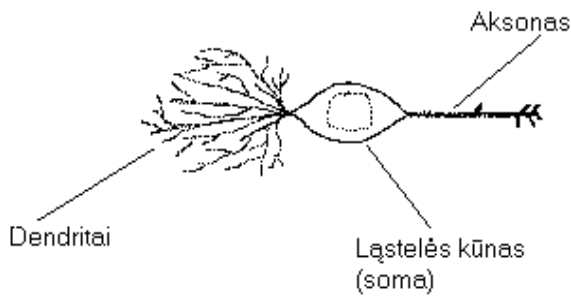
Kas žinoma šiuo metu apie dirbtinius neuroninius tinklus:

- yra žinomos topologijos, galinčios kurti universalų įėjimo-išėjimo atvaizdavimą;
- žinoma kaip sukurti adaptyvius algoritmus, galinčius išgauti informaciją iš duomenų ir koreguoti neuroninių tinklų parametrus;
- pradedamos suprasti būtinos sąlygos apibendrinimui, t.y. kaip garantuoti, kad mokymo imčiai pritaikytas neuroninis tinklas tiktų ir mokymui nenaudotiems duomenims.

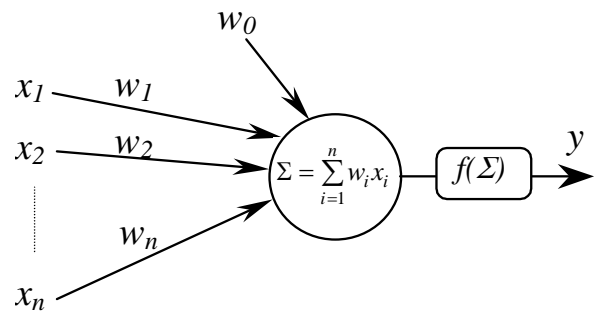
2.2. Dirbtinis neurono modelis

Biologinis neuronas (2 pav.) turi išsišakojusią įėjimo struktūrą (dendritai), ląstelės kūną (soma) ir besišakojančią išėjimo struktūrą (aksonas). Vienos ląstelės aksonas su kitos ląstelės dendritais jungiasi per sinapses. Sinapsėmis vadinamos aksono jungtys su kitų neuronų dendritais.

Siekiant išlaikyti panašumą į biologines neuronines sistemas galima apibūdinti dirbtinio neuroninio tinklo neuroną (3 pav.):



2 pav. Biologinis neuronas.



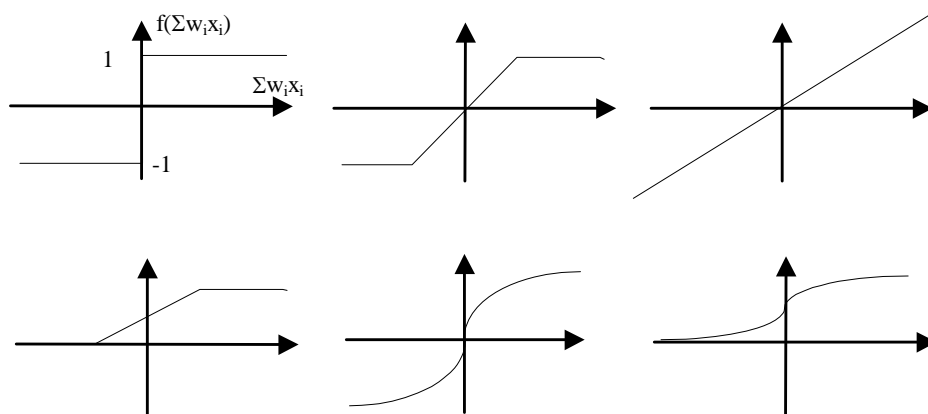
3 pav. Dirbtinio neuroninio tinklo elementas.

Neuronas gauna keletą įėjimo reikšmių. Tai gali būti viso neuroninio tinklo įėjimo reikšmės arba kitų tinklo neuronų išėjimo reikšmės. Kiekviena įėjimo jungtis turi savo perdavimo koeficientą (svorį), šie svoriai atitinka biologinio neurono sinapsių efektyvumą. Kiekvienas neuronas turi savo slenksčio reikšmę. Neurono sužadinimo reikšmė formuojama skaičiuojant svorinę įėjimo signalų sumą ir atimant slenksčio reikšmę.

Pagal sužadinimo signalą, naudojant neurono perdavimo funkciją skaičiuojama neurono išėjimo reikšmė.

Jeigu naudojama šuolinė neurono perdavimo funkcija (neurono išėjimas lygus 0, jei aktyvacijos reikšmė mažiau už nulį, ir lygus 1, jei reikšmė didesnė ar lygi nuliui), neuronas veikia kaip biologinis neuronas. Neurono svoriai gali būti neigiami. Neigiamas svoris reiškia, jog jungtis turi slopinantį, ne žadinantį efektą. Slopinantys neuronai egzistuoja ir biologinėse sistemose

3 paveikslėlyje neurono įėjimai pažymėti $x_1 \dots x_n$; atitinkami svoriai pažymėti $w_1 \dots w_n$; w_0 – slenksčio reikšmė; $f()$ – perdavimo funkcija; y – neurono išėjimas.



4 pav. Neuronų perdavimo funkcijos.

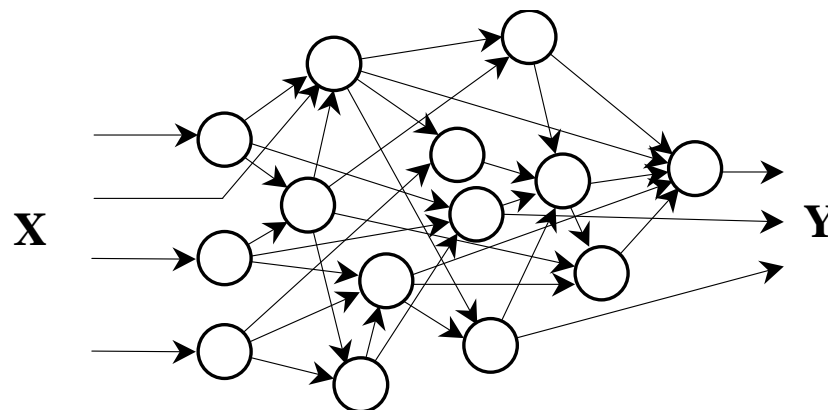
4 paveikslėlyje parodyta keletas neurono perdavimo funkcijų. Kai kurios funkcijos būdingos tam tikro tipo neuroniniams tinklams, kitos sąlygojamos mokymo taisyklių arba parenkamos pagal sprendžiamą uždavinį.

2.3. Tinklo modelis

Atskiri neuronai gali būti jungiami į neuroninį tinklą. Bet kokios paskirties neuroninis tinklas turi įėjimus, perduodančius kintamųjų reikšmes iš išorės, ir išėjimus, formuojančius tinklo atsaką. Įėjimai ir išėjimai atitinka sensorinius ir motorinius nervus, kaip pavyzdžiui, einančius iš akių ir į rankas. Dažnai egzistuoja ir tarpiniai (paslėpti) neuronai atliekantys vidinį vaidmenį tinkle. Įėjimo, paslėpti ir išėjimo neuronai jungiami vieni su kitais.

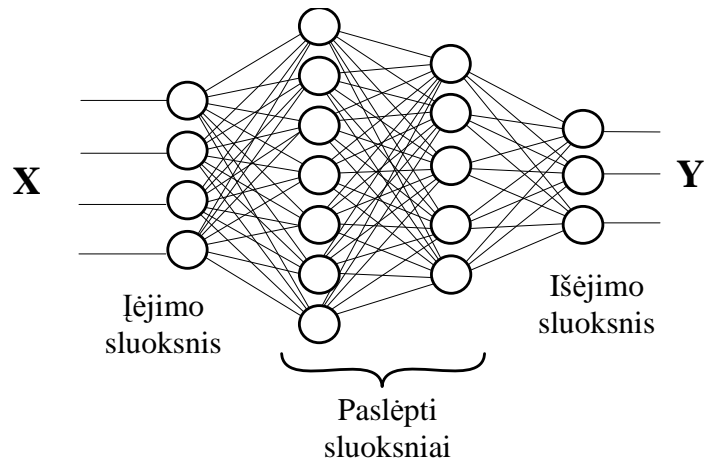
Paprastas dirbtinis neuroninis tinklas turi tiesioginio sklidimo struktūrą: signalai sklinda iš įėjimų pirmyn per visus paslėptus elementus ir pasiekia išėjimo neuronus. Tokia struktūra yra stabili. Jei neuroninis tinklas yra rekurentinis turintis jungtis atgal iš vėlesniųjų į ankstesnius neuronus, jis gali būti nestabilus ir dažniausiai turi sudėtingą dinamiką. Dirbama ir su rekurentiniais neuroniniais tinklais, tačiau realių problemų sprendimui paprastai naudojami tiesioginio sklidimo neuroniniai tinklai.

Neuroniniame tinkle neuronus įprasta grupuoti į atskirus sluoksnius. Galimi tiesioginio sklidimo neuroniniai tinklai, kuriuose neuronai nėra grupuoti sluoksniais. Tokio tinklo pavyzdys parodytas 5 paveikslėlyje. Šiame pavyzdyje tinklas yra tiesioginio sklidimo, tačiau neuronai nėra išdėstyti sluoksniais.



5 pav. Tiesioginio sklidimo neuroninis tinklas

Tipinis tiesioginio sklidimo neuroninis tinklas, kuriame neuronai išdėstyti sluoksniais, pavaizduotas 6 paveikslėlyje.



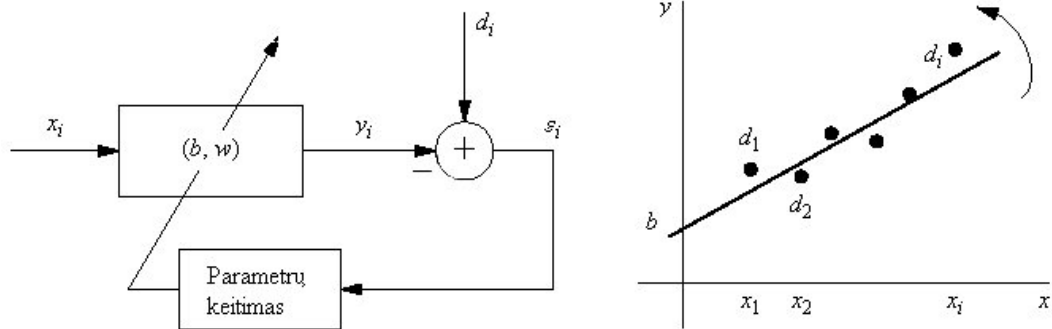
6 pav. Tiesioginio sklidimo daugiasluoksnis neuroninis tinklas.

Piešinyje parodytas įėjimo sluoksnis nėra skaičiuojančių neuronų sluoksnis. Šio sluoksnio paskirtis tėra įvesti į tinklą įėjimo kintamųjų reikšmes. Toks pirmojo sluoksnio tinkle vaizdavimas yra įprastas, kadangi prieš duomenų apdorojimą neuroniniu tinklu dažnai seka pirminio duomenų apdorojimo etapas (pavyzdžiui, normavimas, centravimas). Pavyzdyje paslėptų ir išėjimo sluoksnių neuronai sujungti su visais prieš tai esančio sluoksnio neuronais. Tai nėra taisyklė, kartais neuronai sujungiami ne su visais gretimų sluoksnių neuronais. Vieno sluoksnio neuronai dažniausiai turi tą pačią perdavimo funkciją.

Įėjimo reikšmės patenka į įėjimo sluoksnį, vėliau paeiliui skaičiuojamos paslėptų sluoksnių neuronų išėjimo reikšmės, po to išėjimo sluoksnio neuronų išėjimo reikšmės. Kiekvienas neuronas skaičiuoja svorinę prieš tai esančio sluoksnio neuronų išėjimų sumą ir atima slenksčio reikšmę, taip gaudamas sužadinimo lygį. Pagal gautąją sužadinimo reikšmę skaičiuojama neurono perdavimo funkcijos (išėjimo) reikšmė. Išėjimo sluoksnio neuronų išėjimo reikšmės laikomos neuroninio tinklo išėjimo reikšmėmis.

2.4. Neuroninio tinklo mokymas

Neuroninio tinklo kūrimas grindžiamas mokymu ar adaptacija. 7 paveikslėlyje parodyta tiesinė adaptyvi sistema.



7 pav. Tiesinės sistemos parametru adaptavimas.

Ši tiesinė sistema turi tik du adaptuojamus parametrus b ir w . Sakykime, sistemos uždavinys – pagal x reikšmę išmokti prognozuoti, kokia bus d reikšmė. Mokymo tikslas yra pritaikyti sistemos parametrus (reikšmes b ir w) taip, kad minimizuoti skirtumus tarp sistemos išėjimo reikšmės y_i ir norimo atsako d_i . 7 paveikslėlyje grafiškai parodytas tiesinės sistemos parametru adaptavimas.

Sistemos daroma klaida naudojama sistemos parametru keitimui. Suskaičiuota klaida $\varepsilon_i = d_i - y_i = d_i - (wx_i + b)$ grąžinama sistemai, ir netiesiogiai vėl įtakoja išėjimą, keisdama parametrus. Pradžioje parametrai būna netinkami, ir sistema daro dideles klaidas. Mokymo metu parametrai derinami taip, kad klaida mažėtų. Sistema “susipažindama” su duomenimis suranda geriausią parametru rinkinį. Adaptyvios sistemos pakankamai sudėtingos. Jos turi vykdyti reikalingas funkcijas, be to, jose papildomai naudojamos posistemės parametru keitimui. Reikia pastebėti, jog pasikeitus duomenims ši metodika leidžia modifikuoti sistemos parametrus optimaliam uždavinio sprendimui. Ta pati sistema gali būti taikoma įvairių problemų sprendimui.

Neuroninis tinklas negali padaryti nieko, kas negali būti padaroma tradiciniais skaičiavimo metodais. Tačiau neuroniniu tinklu galima padaryti tai, ką kitais metodais išgauti kartais labai sunku. Neuroninis tinklas gali formuoti modelį iš pavyzdžių. Tai ypač naudinga apdorojant daviklių duomenis arba duomenis iš sudėtingų (cheminių, gamybos, komercinių) procesų. Galbūt, daugeliu atvejų ir egzistuoja duomenų apdorojimo algoritmas, bet dažnai jis nėra žinomas arba turi pernelyg daug kintamųjų. Žymiai paprasčiau, naudojant turimus duomenų pavyzdžius, apmokyti neuroninį tinklą.

Keletas neuroninių tinklų naudojimo pavyzdžių:

- prognozavimas: turint duomenis apie paskutinės savaitės valiutų kurso kitimą bandoma prognozuoti rytojaus valiutų kursą; turint keleto dienų (valandų) duomenis

- vaizdų analizė: padėtas parašas (pavyzdžiui, banke) lyginamas su parašais, saugomais archyve;
- valdymas ir kontrolė: skaičiuojama, kur turi judėti robotas, kai turimas roboto kameros vaizdas; pagal matuojamus proceso parametrus skaičiuojami proceso valdymo parametrai;
- diagnozė: apdorojant vibracijos ir triukšmų lygio daviklių duomenis tikimasi anksti diagnozuoti mechaninių sistemų gedimus.

Naudojant neuroninius tinklus (kaip ir bet kurį kitą metodą), turi egzistuoti priklausomybė tarp žinomų įėjimo ir išėjimo reikšmių. Ši priklausomybė gali būti su tam tikru triukšmų lygiu, tačiau išėjimo reikšmės neturi būti sąlygojamos gryno atsitiktinumo arba faktorių, neatsispindinčių įėjimo duomenyse.

Bendru atveju, neuroniniai tinklai naudojami, kai nėra žinoma priklausomybė tarp įėjimo ir išėjimo duomenų. Jei ši priklausomybė žinoma, duomenys gali būti modeliuojami tiesiogiai. Neuroninis tinklas gali išmokti įėjimo-išėjimo duomenų priklausomybę apmokymo metu ir tai yra pagrindinė neuroninių tinklų savybė.

Įvairūs neuroninių tinklų tipai naudoja skirtingus mokymo metodus. Egzistuoja du pagrindiniai neuroninių tinklų apmokymo tipai. Tai mokymas su mokytoju ir mokymas be mokytojo, iš kurių labiau paplitęs yra mokymas su mokytoju.

7 paveikslėlyje sistemos derinimui naudojamos užduoties reikšmės, todėl tai mokymo su mokytoju pavyzdys. Mokymo su mokytoju adaptyvios sistemos kūrimo sudėtinės dalys:

- sistema su adaptuojamais parametrais;
- duomenų imtis, turinti užduoties reikšmes;
- tinklo kokybę įvertinantis kriterijus, dažnai tai tinklo daroma klaida;
- metodas, leidžiantis optimaliai parinkti sistemos parametrus.

Neuroninis tinklas apmokomas naudojant mokymo imtį. Mokyme su mokytoju mokymo imtis sudaroma iš įėjimo reikšmių kartu su atitinkamomis išėjimo (užduoties) reikšmėmis. Neuroninis tinklas mokosi (nežinomos) priklausomybės tarp jų. Jei tinklas tinkamai apmokytas, jis gali modeliuoti funkciją, siejančią įėjimo bei išėjimo kintamuosius. Tokiu būdu, neuroninis tinklas pagal įėjimo reikšmes gali išduoti nežinomas išėjimo reikšmes.

Mokymo proceso metu dažniausiai skaičiuojamas neuroninio tinklo darbą įvertinantis parametras. Tai gali būti neuroninio tinklo daroma klaida visiems duomenims, pavyzdžiui, skirtumas tarp tinklo išėjimo reikšmės ir norimos (užduoties) reikšmės arba kitoks neuroninį tinklą įvertinantis kriterijus. Toliau pagal mokymo taisyklę koreguojami tinklo svoriai ir slenksčių reikšmės taip, kad neuroninis tinklas darytų mažesnę klaidą arba gerėtų tinklo darbą vertinantis kriterijus.

Modeliuojant tradiciniais metodais (pavyzdžiui, naudojant tiesinį modeliavimą), įmanoma algoritmiškai nustatyti optimalius modelio parametrus, kurie absoliučiai minimizuoja klaidą. To dažniausiai negalima padaryti neuroniniam tinklui, kuris, bendru atveju, yra netiesinis modelis. Neuroninio tinklo svoriai nustatomi mokymo metu, iteraciniu (priartėjimo) keliu.

2.5. Mokymas ir apibendrinimo savybės

Neuroninio tinklo apmokymo tikslas nėra tiksliai atstovauti mokymo duomenų imtį. Mokant siekiama įvertinti ir sudaryti statistikinį proceso modelį, pagal kurį, labiausiai tikėtina, buvo gauti mokymo duomenys. Siekiama sudaryti proceso modelį, pagal kurį generuoti duomenys savo savybėmis būtų kiek įmanoma labiau panašūs į mokymo duomenis. Neuroninis tinklas pagal mokymo imties duomenis turi “išmokti” ir padaryti išvadas apie visus galimus ir nematytus duomenis. Sugebėjimas padaryti išvadas ir tinkamai įvertinti naujus duomenis vadinama apibendrinimo savybėmis.

Neuroninio tinklo apibendrinimo savybės labai priklauso nuo modelio sudėtingumo ir duomenų kiekio mokymo imtyje. Modelis turintis mažai parametrų gali būti per mažai lankstus, tuo tarpu modelis su pernelyg dideliu parametrų kiekiu gali labai prisitaikyti prie duomenų imties ir joje esančių triukšmų. Todėl siekiama parinkti reikiamą neuroninio tinklo sudėtingumą. Daugiasluoksnio neuroninio tinklo sudėtingumą lemia paslėptų neuronų skaičius.

Apibendrinimo klaidą patogiu išskaidyti į dvi sudėtines dalis: bazinę klaidą ir išsibarstymą. Per daug paprastas modelis turi didelę bazinę klaidą, duotajai mokymo imčiai per didelio lankstumo modelis turi didelę išsibarstymo klaidą. Geriausia apibendrinimo klaida gaunama tada, kai randamas kompromisas tarp vienas kitam prieštaraujančių reikalavimų, kad būtų maža bazinė ir maža išsibarstymo klaidos.

Norint pasiekti balansą tarp išsibarstymo ir bazinių klaidų, reikalinga kontroliuoti efektyvų modelio sudėtingumą. Neuroninio tinklo sudėtingumas keičiamas keičiant paslėptų

neuronų, tuo pačiu ir svorių, skaičių. Keletas metodų naudojamų parenkant tinkamą neuroninio tinklo sudėtingumą:

- Lyginami neuroniniai tinklai su skirtingais paslėptų neuronų skaičiais ir išrenkamas tinkamiausias.
- Kuriamas ir apmokomas palyginti didelis neuroninis tinklas ir vėliau atsisakoma mažiausiai reikšmingų jungčių ir neuronų.
- Pradedama nuo mažo tinklo ir mokymo metu pridodami papildomi neuronai iki pasiekiami optimali neuroninio tinklo struktūra.
- Apjungiami keleto neuroninių tinklų išėjimai ir formuojamas komitetas.

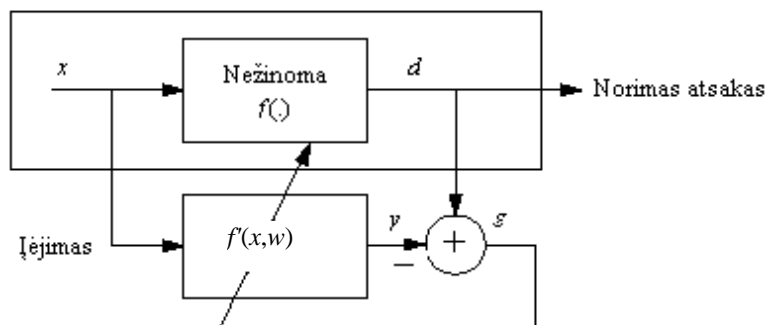
Kitas principinis metodas sudėtingumui kontroliuoti yra klaidos funkcijoje naudoti nari, baudžiantį sudėtingumą. Sudėtingumo nario įtaka bendrai klaidos funkcijai ir modelio sudėtingumui keičiama įvedant sudėtingumo nario daugiklį.

Turint tam tikro dydžio duomenų imtį ir siekiant kontroliuoti modelio sudėtingumą dažnai be mokymo imties sudaroma testinė imtis. Testinės imties pagalba gali būti nustatomas ir kontroliuojamas modelio prisitaikymas prie mokymo imties duomenų.

2.6. Funkcijos aproksimavimas

Apmokant neuroninį tinklą siekiama neuroninio tinklo išėjime išgauti kiekvienam įėjimo vektoriui užduotas reikšmes. Neuroninio tinklo išėjime pagal įėjimo reikšmes turėtų būti kaip įmanoma tiksliau imituojamos užduoties reikšmės.

Mokymo sistema, turėdama įėjimo vektorių x rinkinį ir užduoties d reikšmes, turi surasti optimalų parametrų rinkinį, kuris minimizuotų skirtumą tarp užduoties reikšmių d ir neuroninio tinklo atsako y . Tai funkcijos aproksimavimo uždavinys, kuomet priimama, kad norimos reikšmės d yra gaunamos pagal nežinomą, bet fiksuotą įėjimo funkciją $d=f(x)$, kaip parodyta 8 pav. Šiuo atveju neuroninis tinklas “išmoksta” imituoti nežinomą funkciją f .



8 pav. Funkcijos aproksimavimas mokymo su mokytoju būdu.

Mokymo sistemos tikslas – atrasti funkciją $f()$, turint baigtinį skaičių (tikėtina mažą) įėjimo – išėjimo (x,d) reikšmių porų. Mokymo sistemos išėjimas $y=f'(x,w)$ priklauso nuo sistemos parametru rinkinio w , kuris modifikuojamas taip, kad skirtumas ar neatitikimas tarp sistemos išėjimo reikšmės y ir norimo atsako d būtų minimalus. Jeigu neuroninio tinklo išėjimo reikšmės y tinkamai aproksimuoja d reikšmes, tikimasi, kad funkcija $f'(x,w)$ gali pakeisti nežinomą funkciją $f(x)$.

Aproksimuojant funkciją siekiama keletu paprastų funkcijų pagalba nusakyti labai sudėtingas funkcijas. Funkcijų aproksimavimui gali būti naudojami polinomiali, trigonometriniai polinomiali ir kitos (elementarios) funkcijos.

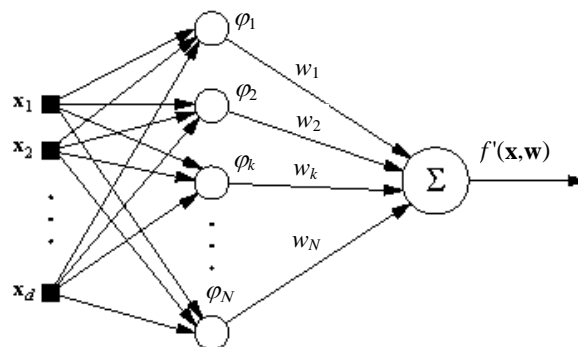
Tegu $f(x)$ yra realių reikšmių vektoriaus $x=[x_1 \ x_2 \dots x_d]$ funkcija. Funkcijos aproksimavimo uždavinys yra aprašyti sudėtingą funkciją $f(x)$ kompaktinėje įėjimo erdvės srityje kaip paprastų funkcijų $\varphi(x)$ kombinaciją:

$$f'(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^N w_i \varphi_i(\mathbf{x}), \quad (1)$$

kur w_i yra koeficientų vektoriaus w komponentės. Siekiama, kad aproksimacija tenkintų sąlygą

$$|f(\mathbf{x}) - f'(\mathbf{x}, \mathbf{w})| < \varepsilon, \quad (2)$$

kur klaida ε parenkama kiek norima maža. Aproksimavimo sistemos blokinė diagrama parodyta 9 pav.



9 pav. Funkcijos aproksimavimas:blokinė diagrama.

Išraiška (2) konstatuojama, kad funkcijos reikšmė gali būti gaunama tiesiškai kombinuojant elementarias funkcijas. Elementarių funkcijų rinkinys $\varphi(x)$ yra universalus, jeigu įmanoma rasti koeficientus w_i , kuriems esant klaida ε yra pakankamai maža bet kokiai funkcijai $f()$ apibrėžtoje srityje. Aproksimuojant funkciją susiduriama su keletu uždavinių:

- elementarių funkcijų rinkinio $\varphi()$ pasirinkimas;

- elementarių funkcijų skaičiaus pasirinkimas.
- svorių w_i skaičiavimas;

Aproksimavimui gali būti naudojamos labai įvairios elementarios funkcijos. Elementarių funkcijų vaidmenį gali atlikti daugiasluoksnio perceptrono paslėpto sluoksnio neuronai.

Uždavinys, kaip skaičiuoti koeficientus w_i , priklauso nuo to, kaip matuojamas skirtumas ar neatitikimas tarp $f(x)$ ir $f'(x, w)$. Ši problema gali būti sprendžiama minimizuojant kvadratinę klaidą tarp $f'(x, w)$ ir $f(x)$. Mažiausių kvadratų metodas taip pat gali būti naudojamas analitiniam w_i reikšmių skaičiavimui. Jeigu įėjimo vektorių x skaičius yra lygus M , elementarių funkcijų skaičius – N , formalios lygtys gali būti užrašomos:

$$\begin{bmatrix} \varphi_1(\mathbf{x}^1) & \varphi_2(\mathbf{x}^1) & \dots & \varphi_N(\mathbf{x}^1) \\ \varphi_1(\mathbf{x}^2) & \varphi_2(\mathbf{x}^2) & \dots & \varphi_N(\mathbf{x}^2) \\ \dots & \dots & \dots & \dots \\ \varphi_1(\mathbf{x}^M) & \varphi_2(\mathbf{x}^M) & \dots & \varphi_N(\mathbf{x}^M) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_N \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}^1) \\ f(\mathbf{x}^2) \\ \dots \\ f(\mathbf{x}^M) \end{bmatrix} \quad (3)$$

Sprendinys tampa

$$\mathbf{w} = \Phi^{-1} \mathbf{f}, \quad (4)$$

kur w yra koeficientų vektorius, f yra reikšmių vektorius, sudarytas iš funkcijos reikšmių visuose M taškuose, Φ yra elementarių funkcijų reikšmės taškuose. Sprendinys gali būti randamas, jei egzistuoja inversinė Φ matrica, ir jos skaičiavimas nėra labai ilgas.

2.7. Funkcijos aproksimavimas daugiasluoksnio neuroniniu tinklu

Jeigu funkcija $f(x)$ (pav.7) yra netiesinė, nėra taip paprasta parinkti bazines funkcijas iš kurių galima suformuoti funkciją $f(x)$. Pasiūlyta nemaža metodų: Voltera eilutės, splainai, skleidimas polinomais. Įrodyta, kad polinomais galima aproksimuoti bet kokią funkciją užduotose ribose. Tačiau norint gauti tikslų aproksimavimą, reikalinga daug narių. Bazinės funkcijos tinkamos, jeigu jos patogios naudoti ir leidžia aproksimuoti bet kokią funkciją.

Elementarios funkcijos skirstomos į dvi rūšis: lokalias ir globalias. Elementari funkcija yra globalinė, jei jos atsakas formuojamas visoje įėjimo srityje. Lokali funkcija “dirba” tik tam tikroje įėjimo erdvės srityje.

Aproksimuojančios sistemos architektūrą (pav. 4) nesunku susieti su neuroniniu tinklu, turinčiu vieną paslėptą sluoksnį, su tiesiniais neuronais išėjime. Tuo atveju, elementarios funkcijos yra paslėptų tinklo neuronų išėjimai:

$$\varphi_i(\mathbf{x}) = g\left(\sum_k w_{ik} x_k + b_i\right), \quad (5)$$

kur $g()$ yra viena iš sigmoidinių funkcijų. Sistemos išėjimas $y = \sum w_i \varphi_i$. Pirmo sluoksnio svoriai keičia $\varphi_i(x)$ formą.

Taigi, vieno paslėpto sluoksnio neuroninis tinklas su tiesiniais elementais išėjime gali būti traktuojamas kaip funkcijų aproksimavimo sistema, kurioje paslėptų neuronų funkcijos yra bazinės aproksimatoriaus funkcijos. Sigmoidinis neuronas formuoja atsaką visoje x erdvėje (1, -1 ar tarpinės reikšmės), todėl jo funkcija yra globalinė.

Funkcijas aproksimuojančio daugiasluoksnio neuroninio tinklo elementarios funkcijos yra adaptyvios, ir yra derinamos pagal įėjimo duomenis. Tai reiškia, kad jos nėra nustatytos iš anksto, kaip sinc, bangelės, ar sinusinės funkcijos, naudojamos Furjė analizėje.

Neuroninio tinklo naudojamos elementarios funkcijos priklauso nuo pirmo sluoksnio svorių ir įėjimo. Skirtingų sluoksnių svoriai šiuo atveju atlieka skirtingus vaidmenis. Pirmojo sluoksnio svoriai nusako elementarių funkcijų formą, antrojo sluoksnio svoriai apjungia elementarias funkcijas. Sistemos su adaptyviomis elementariomis funkcijomis apmokymas tampa sudėtingesnis. Reikalinga ne tik kombinuoti elementarių funkcijų atsakus bet ir tinkamai priderinti pačių funkcijų parametrus.

Įrodyta, kad daugiasluoksnis neuroninis tinklas yra universali aproksimuojanti sistema. Be įrodymų, kad neuroninis tinklas gali aproksimuoti bet kokią funkciją norimu tikslumu, labai svarbūs tyrimai kaip kinta aproksimavimo tikslumas, keičiantis problemos sudėtingumui, įėjimo erdvės matavimų skaičiui ir elementarių funkcijų kiekiui. Įrodyta, kad asimptotinis aproksimavimo daugiasluoksniu neuroniniu tinklu tikslumas apytikriai nėra priklausomas nuo įėjimo erdvės dimensijos.

Tuo tarpu aproksimavimo polinomais tikslumas eksponentiškai priklausomas nuo įėjimo erdvės matavimų skaičiaus (klaida eksponentiškai mažėja mažėjant įėjimo erdvės dimensijai arba polinomo narių skaičius turi didėti eksponentiškai, didėjant erdvės matavimų skaičiui norint išlaikyti tą patį aproksimavimo tikslumą). Daugiasluoksnis neuroninis tinklas funkcijų aproksimavimui yra žymiai efektyvesnis nei polinomi aukštų matavimų erdvėse.

Funkciją aproksimuojančiame neuroniniame tinkle išėjimo neuronai paprastai turi tiesinę perdavimo funkciją. Tačiau gali būti naudojama ir bet kuri kita funkcija. Tuomet svarbu tinkamai parinkti užduoties reikšmių dinaminį intervalą. Aproksimuojant funkcijas neuronų perdavimo funkcijos nebūna įsisotinusios, kadangi paprastai siekiama išgauti glotnias funkcijas. Klasifikuojant, kur išėjime siekiama gauti 1 ar 0, neuronai paprastai turi didelius svorius ir dirba soties režimu.

2.8. Klaidų įvertinimas

Neuroninių tinklų mokymas grindžiamas klaidos minimizavimo metodais. Suminė kvadratinė klaida yra paprasčiausia klaidos funkcija ir tinkamiausia sprendžiant regresijos problemas. Funkcija gali būti naudojama klasifikavimo problemoms spręsti. SSE gaunama sumuojant klaidas visuose tinklo išėjimuose visai duomenų imčiai:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^c \{y_k(\mathbf{x}^n; \mathbf{w}) - t_k^n\}^2, \quad (6)$$

Statistikoje vidutinės kvadratinės klaidos (MSE) įvertinimas yra vienas iš būdų kiekybiškai apibrėžti skirtumą tarp tikrosios vertės ir apskaičiuotosios. Tas skirtumas susidaro dėl atsitiktinumo arba dėl to, kad skaičiuotojas (estimator) neturi informacijos, kuri leistų atlikti tikslesnį skaičiavimą.

Skaičiuotojo $\hat{\theta}$ (estimator) MSE skaičiuojamas taip:

$$\text{MSE}(\hat{\theta}) = E((\hat{\theta} - \theta)^2). \quad (7)$$

MSE gali būti skaičiuojamas kaip variacijos ir paklaidos kvadrato suma:

$$\text{MSE}(\hat{\theta}) = \text{Var}(\hat{\theta}) + (\text{Bias}(\hat{\theta}, \theta))^2. \quad (8)$$

Vidutinis kvadratinės šaknies nuokrypis (RMSD) (dar žinomas kaip Vidutinė kvadratinės šaknies klaida (RMSE)), yra dažnai naudojamas palyginti vertes, kurias prognozavo modelis, su tomis, kuriomis naudotasi tą modelį sukurti.

Skaičiuotojo $\hat{\theta}$ (estimator) RMSD skaičiuojamas taip:

$$\text{RMSD}(\hat{\theta}) = \sqrt{\text{MSE}(\hat{\theta})} = \sqrt{E((\hat{\theta} - \theta)^2)}, \quad (9)$$

Objektyviam, nešališkam įvertinimui RMSE yra variacijos kvadratinė šaknis, dar žinoma kaip standartinė klaida. Kai kur RMSD naudojamas palyginti du skirtingus dalykus, kurie abu skiriasi nuo „standarto“. Pavyzdžiui, matuojant vidutinį atstumą tarp dviejų stačiakampių objektų, apibrėžtų atsitiktiniais vektoriais:

$$\theta_1 = \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,n} \end{bmatrix} \quad \text{and} \quad \theta_2 = \begin{bmatrix} x_{2,1} \\ x_{2,2} \\ \vdots \\ x_{2,n} \end{bmatrix} \quad (10)$$

Tada formulė bus tokia [10]:

$$\text{RMSD}(\theta_1, \theta_2) = \sqrt{\text{MSE}(\theta_1, \theta_2)} = \sqrt{\text{E}((\theta_1 - \theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^n (x_{1,i} - x_{2,i})^2}{n}} \quad (11)$$

2.9. Kiti prognozavimo modeliai

2.9.1. ARMA modelis

Tai auto-regresijos slenkančio vidurkio modelis

$$w_t = \sum_{i=1}^p a_i w_{t-i} + \sum_{j=1}^q b_j \epsilon_{t-j} + \epsilon_t. \quad (12)$$

w_t akcijų kursas rytoj

w_{t-1} akcijų kursas šiandien,

p dienų kiekis, kiek atsimenam į praeitį

ϵ_t atsitiktinė paklaida rytoj,

a_i, b_j įtakos koeficientai,

juos nustatom minimizuodami

$$f(x) = \sum_{t=1}^T \epsilon_t^2, \quad (13)$$

kur

$$x = (x_1, \dots, x_{p+q}), \quad x_i = a_i, \quad i = 1, \dots, p, \quad x_i = b_{i-p}, \quad i = p+1, \dots, p+q. \quad (14)$$

ARMA modelyje prognozuojamos reikšmės gavimą galima būtų padalinti į trijų dalių sumą: tam tikro kiekio p prieš tai buvusių reikšmių w_{t-i} padaugintų iš optimizavimo metu rastų koeficientų a_i sumos, vadinamos auto regresijos dalimi, metodo duodamų e_{t-i} paklaidų sandaugos su atitinkamais koeficientais b_i sumos, vadinamos slenkančio vidurkio dalimi ir e_t

nežinomų aplinkos veiksnių dar vadinamų baltuoju triukšmu. Susidaro išpūdis, jog gauti prognozę yra visai paprasta, tereikia turėti duomenų seką t.y. laiko eilutę, tačiau čia svarbu pabrėžti jog prognozės tikslumas labai priklausys nuo a ir b koeficientų arba metodo, kurio pagalba jie bus rasti. Žinoma nereikia pamiršti jog svarbi ir pati laiko eilutė ar tai prognozuojami duomenys, ar tiesiog atsitiktinių skaičių seka, kuomet joks metodas patenkinamai nesuprognozuos koks bus sekantis atsitiktinai suprognozuotas skaičius.

2.9.2. AR-ABS modelis

Auto regresinis mažiausių absoliutinių nukrypimų modelis (AR-ABS) tai ARMA modelio modifikacija. Kadangi mažiausių kvadratų metodas yra labai jautrus dideliems nukrypimams duomenyse pvz.: minimizuojant kvadratinius nuokrypius didelis nuokrypis toks kaip 100 turi tokią pat įtaką kaip ir dešimt tūkstančių mažų lygių vienetui nuokrypių. Todėl kilo idėja mažiausių kvadratinių nuokrypių minimizavimą pakeisti nuokrypių absoliutiniais didumais minimizavimu.

$$w_t = \sum_{i=1}^p a_i w_{t-i} + \epsilon_t \quad (15)$$

kur

w_t prognozė rytoj

w_{t-1} ši diena,

p dienų kiekis, kiek atsimenam į praeitį

ϵ_t atsitiktinė paklaida rytoj,

a_i įtakos koeficientai,

juos nustatom minimizuodami

$$f(x) = \sum_{t=1}^T |\epsilon_t|. \quad (16)$$

Prognozuojamas dydis priklauso nuo prieš tai buvusių reikšmių ir tam tikro aplinkos poveikio, kuris apibrėžiamas kaip atsitiktinė skaičių seka, pasiskirsčiusi pagal Gauso dėsnį. Trumpai tariant tai yra AR modelis ir tik a koeficientų radimui čia bus taikomas ne mažiausių kvadratų, o absoliutinių didumų metodas. [8]

3. NAUDOJAMŲ PRIEMONIŲ IŠNAGRINĖJIMAS IR MODELIŲ PARUOŠIMAS

3.1. Įvadas

Pastaruoju metu rinkos tapo labiau prieinama investavimo priemonė ne tik strateginiams investuotojams, bet ir eiliniams žmonėms. Taigi rinka yra ne tik susijusi su makroekonominiais veiksniais, bet jie tiesioginiu būdu įtakoja kasdienį gyvenimą. Tai yra mechanizmas, kuris daro svarbų tiesioginį socialinį poveikį. Skirtingi asmenys žvelgia į šią problemą savaip. Svarbus interesų skirtumų šaltinis yra skirtingas požiūris į riziką: priešiški rizikai, neutralūs rizikai, linkę rizikuoti. Apžvelgsime keturias konkrečias priemones naudojamas moksliniams tyrimams. Biržos žaidimo modelis (Stock Exchange Game Model, SEGM) ir autoregresijos absoliutaus nuokrypio modelis (AR-ABS), yra įgyvendinamos Java kalba, skelbiamos atviroje svetainėje, ir gali būti paleisti bet kokia naršykle su Java palaikymu. Trečiasis ir ketvirtasis įrankiai Artificial Neural Networks (ANN) modeliai. Vienas jų sukurtas naudojant komercinę MATLAB programinę įrangą, o kitas nemokamą JOONE paketą.

SEGM modelis modeliuoja N didelių turto turėtojų elgesį, kai pirkimo-pardavimo strategijos yra apibrėžtos naudojant autoregresijos (AR), arba atsitiktinės eigos (RW) modelių prognozes. AR-ABS modelis skiriasi nuo tradicinių AR tuo, kad minimizuoja absoliučius nukrypimus vietoj mažiausių kvadratų nukrypimų. ANN modeliai naudoja klasikinę dirbtinio neuroninio tinklo daugiasluoksnio tiesioginio sklidimo perceptrono versiją. Vertybinių popierių rinkos prognozavimo užduotis skirsto mokslininkus į dvi grupes. Pirmoji grupė yra tie, kurie tiki, kad mes galime sukurti mechanizmus prognozuoti rinką pagal statistikę ir fundamentalią analizę. Statistinės analizės pavyzdys yra slenkančio vidurkio autoregresija (Autoregressive Moving Average, ARMA) skirtas trumpalaikiam prognozavimui, o dalinės integracijos (ARFIMA) modelis vykdo ilgo laikotarpio prognozes. Fundamentalieji analitikai tiria vidinės vertės turtą ir investuoja į jį, jei jie mano, kad jo dabartinė vertė yra mažesnė už jo tikrąją vertę.

Kita grupė mokslininkų pripažįsta efektyvios rinkos hipotezę (EMH). Jie mano, kad rinka yra veiksminga ir kai paaiškėja nauja informacija, rinkos absorbuoja ją koreguojant save. Prognozuoti naudojamas RW modelis. Matematinė priemonė įgyvendinant RW yra Wiener'io modelis, kuris reiškia, kad geriausia prognozė rytojui yra šiandienos vertė.

Birža laikoma, kaip žaidimas, kuriame konkuruojantys žaidėjai bando padidinti laimėjimus ir perka turtą mažomis kainomis ir pardavinėja didesnėmis kainomis. Pardavimo/pirkimo lygiai grindžiami prognozėmis. Taigi žaidėjai gauna pelną, jei jie numatė teisingai. Nešo pusiausvyra pasiekama, jei visų pagrindinių žaidėjų prognozuojama turto kaina pagal Wiener'io modelį. Tai reiškia, kad žaidėjas negali gauti daugiau tikėtino pelno naudojant autoregresijos modelio prognozes, kai visi pagrindiniai žaidėjai naudoja Wiener'io modelį.

SEGM sumodeliuoti duomenys yra lyginami su realių duomenų prognozėmis. Palyginimui naudojami dirbtinio neuroninio tinklo (ANN) ir autoregresijos pagal absoliučias klaidas (AR-ABS) algoritmai. AR-ABS modelis taikomas laiko eilutėms prognozuoti minimizuojant vidutinę absoliučią klaidą (MAE). Absoliutaus nuokrypio minimizavimas išplaukia iš prielaidos, kad vartotojo naudingumo funkcija yra tiesinė. Todėl AR-ABS modelis atspindi neutralų rizikai elgesį. Tai pagrindinė priežastis naudoti MAE. Papildoma priežastis yra kai kurios palankios MAE įverčių statistinės savybės.

Populiarūs ir patogūs įrankiai nelineinėms prognozėms yra ANN. Kaip ir AR-ABS atveju ANN versija dirba minimizuodama MAE atspindėdama vartotojams neutralų rizikos elgesį. Neuroninio tinklo tipas tiesioginio daugiasluoksnio perceptrono su sigmoidės aktyvavimo funkcijomis ir MAE nuostolio funkcijomis. Dirbant su faktiniais duomenimis ir vidutinės kvadratinės klaidos (MSE), ir vidutinės absoliučios klaidos (MAE) apskaičiavimai buvo taikomi atstovauti skirtingų vartotojų požiūriui.

Mažai tikėtina, kad EMH užtikrina tikslų aprašymą, kaip elgiasi akcijų rinkos kainos. Tačiau modelis gali būti naudingas, net jei jis tiksliai neatitinka faktų. Taigi nors ir kainų pokyčiai gali būti ne tik nepriklausomi, modelis yra pradinės aproksimacijos pavyzdys ir gali būti pagerintas gaunant papildomos informacijos apie pagrindines vertybes ir sekant kai kurių modelių akcijų kurso pokyčius.

3.2. Biržos žaidimo modelis, Nešo pusiausvyra

Mokslinis šio modelio tikslas yra patikrinti hipotezę, kad biržą galima apytikriai apibūdinti kaip kelių žaidėjų žaidimą, naudojant strategijas, grindžiamas Nešo pusiausvyra. Sumodeliuotos vertybių kainos yra žaidimo rezultatas, susidedantis iš pagrindinių turto turėtojų plus atsitiktinio kintamojo, atstovaujamojo didelio kiekio mažųjų. Investiciniai sprendimai priklausys nuo žaidėjo prognozės dėl būsimos turto kainos ir laukiamų dividendų. Šio modelio tikslas iširti, kas gali būti išmokta apie žaidimo pagrindus ir rinkos teorijas su šiuo paprastu biržos modelio pavyzdžiu.

Daroma prielaida, kad kiekvienas žaidėjas prognozuoja turto kainą pagal autoregresijos modelį. AR modelio parametrai apskaičiuojami standartiniu mažiausių kvadratų algoritmu naudojant skirtingus p . Naujasis metodas yra taikomas siekiant įvertinti parametrus p : jie gaunami minimizuojant vidutinį nuokrypį nuo Nešo pusiausvyros sąlygos. Pusiausvyra dažnai pasiekama, kai visi pagrindiniai žaidėjai prognozuoja turto kainas Wiener'io modeliu, kur $p = 1$, ir $a = 1$. Tai reiškia, kad vidutinis pelnas negali padidėti naudojant AR modelius, su didesniais atminties parametrais $p > 1$. Tai naujas paaiškinimas, kodėl RW modelus prognozuoja "rytoj, kaip šiandien".

SEGM pagrindinė idėja apsvarstant paprasčiausią atvejį: vienas brokeris, I pagrindinių klientų $i = 1 \dots I$, ir J akcinių bendrovių $j = 1 \dots J$.

Tarkime, akcinės bendrovės j akcijos kaina $z^j(t)$ momentu t lygi svarbiausių klientų $i = 1 \dots I$ paskutiniam sandoriui, pridėjus kažkokį atsitiktinį skaičių $\varepsilon^j(t)$, įtakotą smulkių klientų. Tuo laiko momentu $t = 1 \dots T$, klientas i užsako biržos maklerį pirkti skaičių $n = n_i^j(t)$ bendrovės j akcijų, jei

$$z^j(t) \leq z_i^{j_{buy}}(n, t) \quad (17)$$

Čia $z_i^{j_{buy}}(n, t)$ yra kliento i skaičiaus n akcijų j pirkimo riba metu t . Tuo laiko momentu $t = 1 \dots T$ klientas i užsako biržos makleriui parduoti skaičių $n = n_i^j(t)$ akcijų j , jei

$$z^j(t) \geq z_i^{j_{sell}}(n, t) \quad (18)$$

Čia $z_i^{j_{sell}}(n, t)$ yra kliento i skaičiaus n akcijų j pardavimo riba metu t . Pirkimo riba yra įmanoma, jei

$$n \leq N_i(t) \quad (19)$$

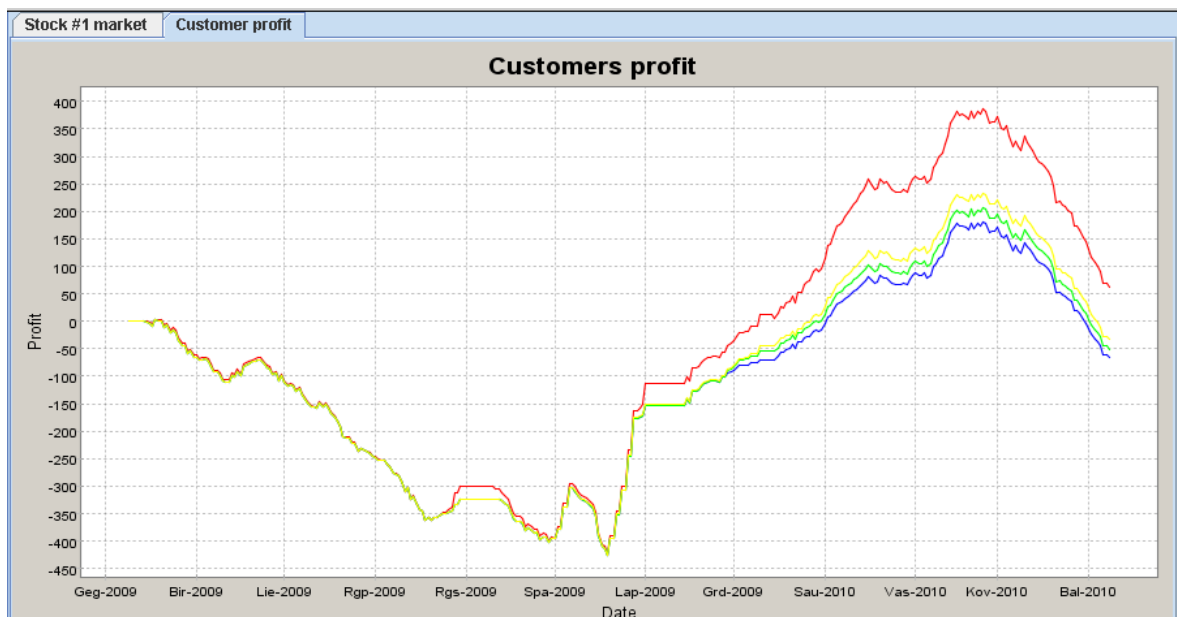
Čia $N_i(t)$ yra kliento i pirkimo galia metu t . Tai apibrėžia, pavyzdžiui, pradinį turtą N_i plius turto balansą momentu t . Vartotojas gauna pelną, lygų skirtumui tarp pirkimo ir pardavimo kainų, jei jo prognozės yra teisingos.

SEG modelis įgyvendintas Java kalba ir yra prieinamas internete. Jokių specialių įgūdžių nereikia dirbant ir pritaikant modelį. Todėl šis modelis yra patogus įrankis modeliuoti ir tyrinėti elgesį biržoje kaip N asmenų žaidimą. Be to, visus rezultatus galima savarankiškai išbandyti ir palyginti su kitais modeliais. Programa įgyvendina biržos modelį su $I \leq 8$ stambių klientų ir $J \leq 8$ akcinių bendrovių skaičium. Grafinė sąsaja skirta skaičių p sąveikaujančiai paieškai, kurie minimizuotą vidutinį nukrypimą nuo Nešo pusiausvyros sąlygos. 10 paveikslas rodo 4 pagrindinių žaidėjų pradinius parametrus [6].

Customer #0		Customer #1		Customer #2		Customer #3	
Method:	WIENER	Method:	AR1	Method:	AR2	Method:	AR3
Has stock #1:	0	Has stock #1:	0	Has stock #1:	0	Has stock #1:	0
Has stock #2:	0	Has stock #2:	0	Has stock #2:	0	Has stock #2:	0
Has stock #3:	0	Has stock #3:	0	Has stock #3:	0	Has stock #3:	0
Has stock #4:	0	Has stock #4:	0	Has stock #4:	0	Has stock #4:	0
Has stock #5:	0	Has stock #5:	0	Has stock #5:	0	Has stock #5:	0
Kbuy2:	0.08	Kbuy2:	0.08	Kbuy2:	0.08	Kbuy2:	0.08
Kbuy1:	0.05	Kbuy1:	0.05	Kbuy1:	0.05	Kbuy1:	0.05
Kbuy:	0.025	Kbuy:	0.025	Kbuy:	0.025	Kbuy:	0.025
Ksell:	-0.025	Ksell:	-0.025	Ksell:	-0.025	Ksell:	-0.025
Ksell1:	-0.05	Ksell1:	-0.05	Ksell1:	-0.05	Ksell1:	-0.05
Ksell2:	-0.08	Ksell2:	-0.08	Ksell2:	-0.08	Ksell2:	-0.08

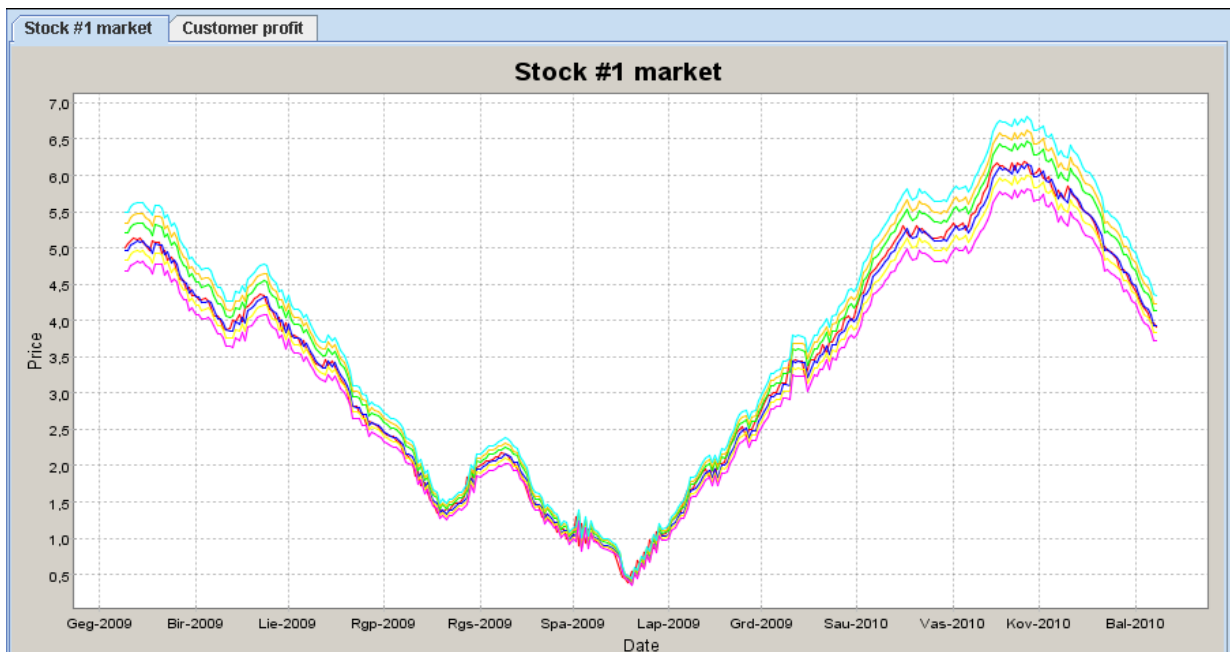
10 pav. Žaidėjų pradiniai parametrai

11 paveiksle pavaizduota skirtingų žaidėjų pelnas sumodeliuotame žaidime, kai pirmasis žaidėjas numato turto kainas remdamasis Wiener'io modeliu, o likę žaidėjai prognozuoja naudodami AR modelį.



11 pav. Skirtingų žaidėjų pelnas

Raudona linija rodo pirmojo žaidėjo pelną. Kitos linijos rodo likusių dalyvių pelną. 12 paveikslas rodo eilutes, kurios rodo sumodeliuotas turto kainas ir skirtingų veikėjų nustatytus pirkimo/pardavimo apribojimus. [7]



12 pav. Sumodeliuotos turto kainos ir pirkimo/pardavimo apribojimai

3.3. Autoregresijos absoliutaus nuokrypio modelis (AR-ABS)

Laiko eilučių modeliai yra vieni tinkamiausių prognozavimo uždaviniams spręsti. Vienas seniausių yra auto regresinis slenkančio vidurkio modelis ARMA. Iš šio modelio išsirutuliojo visa eilė daugiau ar mažiau ištirtų modifikacijų.

AR-ABS - mažiausių absoliučių nukrypimų modelis yra viena šio modelio, modifikacijų, tiksliau tai AR modelis, kur vietoj mažiausių kvadratų metodo paklaidų minimizavimo naudojamas paklaidų absoliutiniais didumais minimizavimas.

Mažiausių kvadratų metodas yra jautrus dideliems nukrypimams. Pavyzdžiui, kai minimizuojam kvadrato nuokrypį, didelis nukrypimas šimtas turi tokį patį poveikį kaip dešimt tūkstančių nedidelių nukrypimų po vieną. Todėl, pakeisti kvadratinius nuokrypius į absoliučias vertes yra naudinga, jei naudingumo funkcija yra tiesinė. Linijinė naudingumo funkcija atspindi neutralios rizikos elgesį. Apytikslis sprendimas gali būti gautas skirtinguose prognozavimo modeliuose minimizavus absoliutinių nukrypimų sumą, naudojant globalios optimizacijos algoritmus. AR-ABS modelis atrodytų taip:

$$w_t = \sum_{i=1}^p a_i w_{t-i} + \varepsilon_t \quad (20)$$

Manome, kad

$$w_{t-i} = 0, \varepsilon_{t-i} = 0, \text{ jei } t \leq i \quad (21)$$

Čia w_t yra kitos dienos prognozė, w_{t-1} yra tikroji vertė šiandien, w_{t-2} yra vakarykštė vertė, ε_t nepriklausomas atsitiktinis skaičius gaunamas pagal Gauso pasiskirstymą.

Skaičiuojam liekanas:

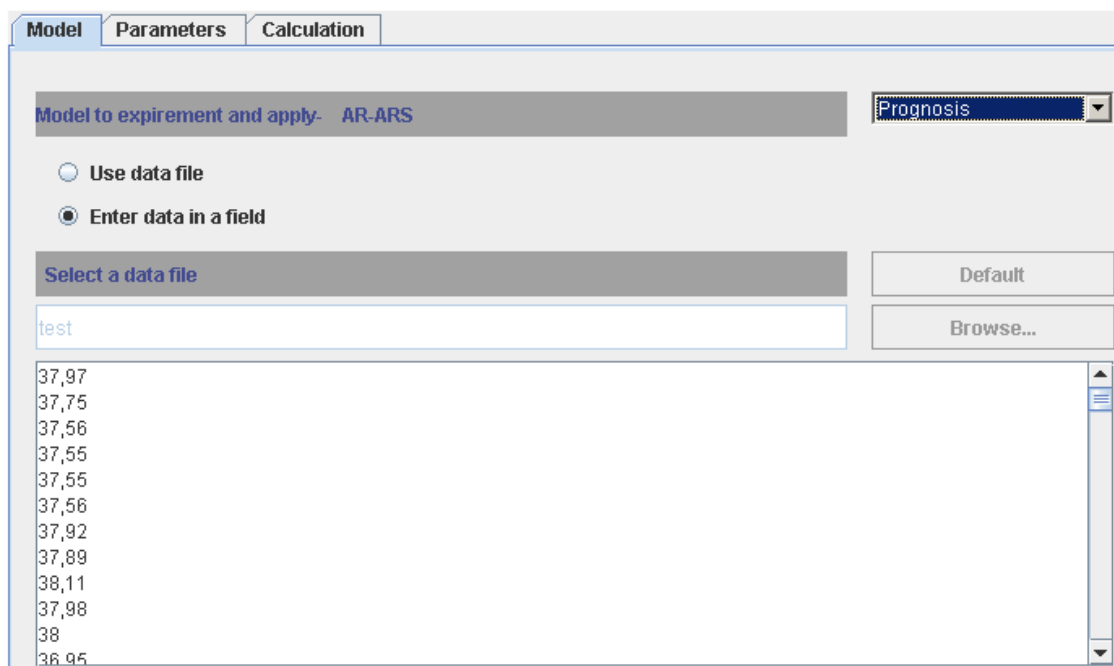
$$\begin{aligned} \varepsilon_1 &= w_1 \\ \varepsilon_2 &= w_2 - a_1 w_1 \\ &\dots\dots\dots \\ \varepsilon_t &= w_t - a_1 w_{t-1} - \dots - a_p w_{t-p} \end{aligned} \quad (22)$$

Paskui sumą

$$f(x) = \sum_{t=1}^T |\varepsilon_t| \quad (23)$$

minimizuojam. Čia $x = (a_i, i = 1 \dots p)$ ir T yra dabartinis laikas.

AR-ABS programa gali būti paleista per interneto naršyklę. Pradiniam lange reikia pasirinkti duomenų failą arba įrašyti duomenis į tam tikrą lauką. [8]



13 pav. Duomenų parinkimo langas

Parametrų langas atrodo taip:

Property	Value
Number of factors (M) :	1
AR parameter (p) :	2
Number of data entrees (T) :	10
Indikator for one p evaluation (0), increasing p(1) :	1
Indikator for one step evaluation(0), multistep(1) :	1
Indikator for multi days prognose (1) :	0
Number of multi days prognose repetitions K :	0

14 pav. Parametrų parinkimo langas

Parametrai:

1. “Number of factors (M):” – įvedimo lauke galima įvesti skaičių iš intervalo [1; 32767]. Įvestas skaičius apibrėš išorinių faktorių kiekį, kurie bus įvertinami tiek sudarant konkretų AR-ABS modelį tiek prognozuojant sekančiam laiko momentui. $M=1$ reikš jog atsižvelgiama tik į patį prognozuojamą dydį ir jo reikšmes praeities momentais. Rekomenduojama, kad išorinių faktorių kiekis nebūtų didesnis nei duomenų failo eilutės elementų kiekis, priešingu atveju duomenys apibrėšiantys išorinius faktorius bus imami iš sekančių failo eilučių.
2. “AR parameter (p):” – įvedimo lauke galima įvesti skaičių iš intervalo [0;32767], jis apibrėžia laiko momentų į praeitį kiekį, kuriuos mes atsimeiname.
3. “Number of data entrees (T):” – įvedimo lauke galima įvesti skaičių iš intervalo [1;32767]. Įvesta reikšmė apibrėš tariamai turimų duomenų kiekį. Duomenų kiekis (T) turi būti didesnis už antrajame lauke įvestą dydį.
4. “Indicator for one p evaluation (0), increasing p(1)” – įvedimo lauke gali būti įvesta reikšmė 0 arba 1.
 - “0” – reikš, jog skaičiavimai bus atlikti vieną kartą su pasirinkta p reikšme.
 - “1” – reikš, jog skaičiavimai bus atlikti p kartų, pirmą kartą $p=1$, antrą $p=2$ ir t.t. Bus rastas optimalus p parametras, tai yra toks su kuriuo gaunami mažiausi nuokrypai prognozuojamų dydžių nuo realių.
5. “Indicator for one step evaluation (0), multistep (1)” – šiame įvedimo lauke taip pat gali būti įvestas tik 0 arba 1.
 - “0” – reikš, jog prognozuosim tik vienam sekančiam laiko momentui $T+1$.

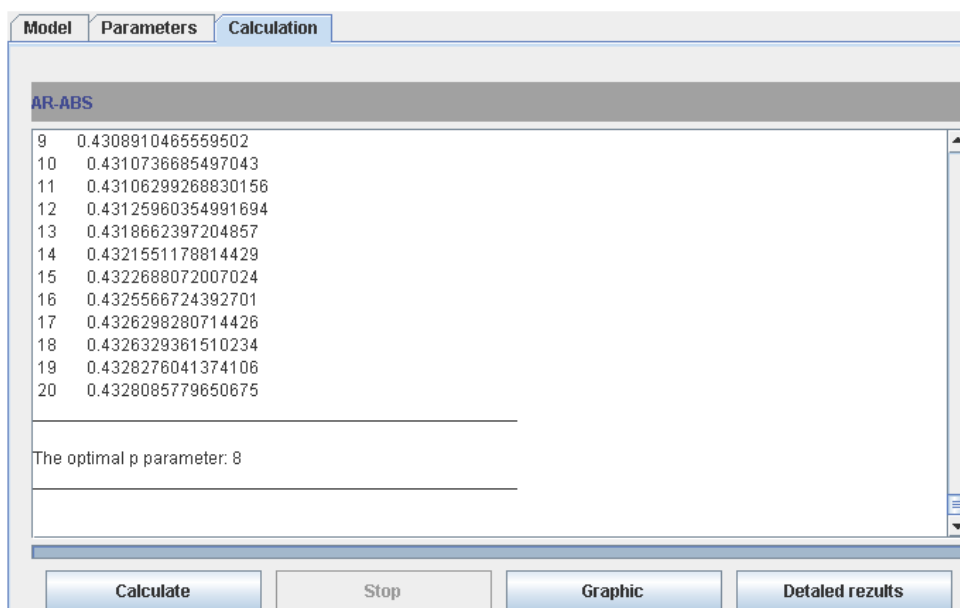
- “1” – reikš, jog prognozuosim tokiam pat kiekiui laiko momentų, kokia buvo duomenų imtis T , t.y. prognozuosim $T+1$, $T+2$, $T+3...T+T$ laiko momentams. Bus rastas vidutinis nuokrypis absoliutiniu didumu prie tam tikro parametro p .
6. “Indicator for multi days prognose (1):“ – šiame įvedimo lauke taip pat gali būti įvestas tik 0 arba 1.
- “1” – reikš, jog prognozuosim tokiam pat kiekiui laiko momentų į ateitį kokia buvo duomenų imtis T .
7. “Number of multi days prognose repetitions K :” – reikš kiek kartų, prognozuojant eilę laiko momentų į ateitį, kiekvienam laiko momentui bus kartojama prognozė.

Programa vykdo griežtą šių parametrų įvedimo kontrolę, neteisingai įvesta reikšmė yra atstatoma į paskutinę teisingą ir pranešama atitinkama perspėjimo žinute.

Skaičiavimų (“Calculation”) lange esančių mygtukų pagalba atliekami veiksmai:

- Paspaudus mygtuką “Calculate” vykdomi skaičiavimai;
- Mygtuku “Stop” skaičiavimai nutraukiami. Skaičiavimai nutraukiami baigus loginį skaičiavimų vieneta, o ne iš karto.
- Paspaudus mygtuką “Graphic” – atskirame lange bus pateiktas skaičiavimo rezultatų grafikas.
- Paspaudus mygtuką “Detailed results” atskirame lange, bus pateikti detalūs rezultatai.

Vykdam skaičiavimus galimi pranešimai apie problemas atsiradusias skaitant duomenis iš failo.



15 pav. Skaičiavimų langas

Gauti skaičiavimo rezultatai prognozuojant analizuojami taip:

“p” – atsiminimo į praeitį momentų kiekis;

“Predicted” – sekantis prognozuojamas dydis;

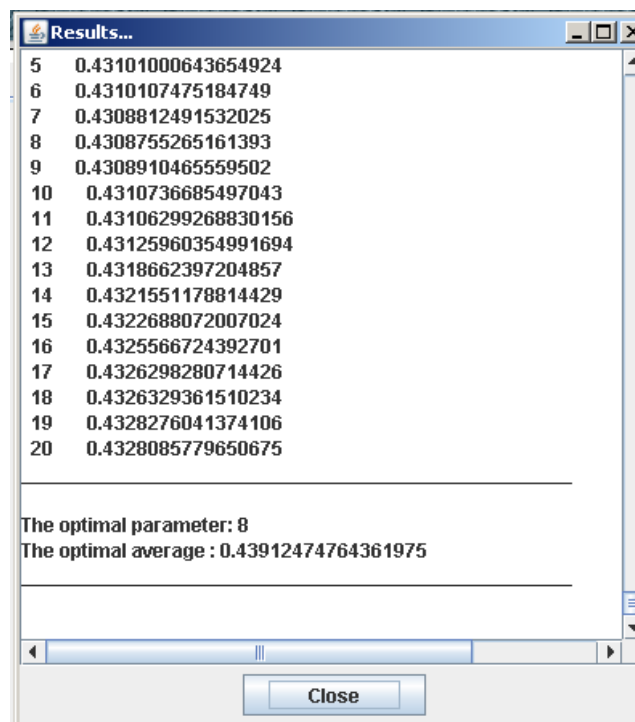
“Should be” – koks iš tikrųjų dydis turėtų būti;

“Residuals” – skirtumas tarp prognozuojamo ir tikrojo dydžio;

“Sum of all residuals” – visų skirtumų absoliutiniu didumu suma;

“The average of residuals” – vidutinis skirtumas;

“The optimal p parameter:” – optimali p parametro reikšmė.



Index	Value
5	0.43101000643654924
6	0.4310107475184749
7	0.4308812491532025
8	0.4308755265161393
9	0.4308910465559502
10	0.4310736685497043
11	0.43106299268830156
12	0.43125960354991694
13	0.4318662397204857
14	0.4321551178814429
15	0.4322688072007024
16	0.4325566724392701
17	0.4326298280714426
18	0.4326329361510234
19	0.4328276041374106
20	0.4328085779650675

The optimal parameter: 8
The optimal average : 0.43912474764361975

Close

16 pav. Rezultatų langas

3.4. Tiesioginis neuroninis tinklas naudojant MatLAB ir Neural network tool

Ekspimente buvo naudojama klasikinė dirbtinio neuroninio tinklo versija - daugiasluoksnis tiesioginio sklidimo perceptronas. Tinklas susideda iš 3 sluoksnių. Pirmasis sluoksnis (įėjimų) yra virtualus sluoksnis, per kurį paduodami duomenys. Antrasis (paslėptas) ir trečiasis (išėjimų) sluoksniai pagaminti iš neuronų. Tinklas yra regresijos tipo neuroninis tinklas, kur įėjimai yra laiko eilutės ankstesnės reikšmės o išėjimas yra sekanti vertė.

Sukurti tiesioginį tinklą galima naudojant funkciją NEWFF. Sintaksė:

```
net = newff(PR,[S1 S2...SN1],{TF1 TF2...TFN1},BTF,BLF,PF)
```

Parametrai:

PR – įėjimų reikšmių minimali ir maksimali reikšmės

Si – i-tojo sluoksnio dydis

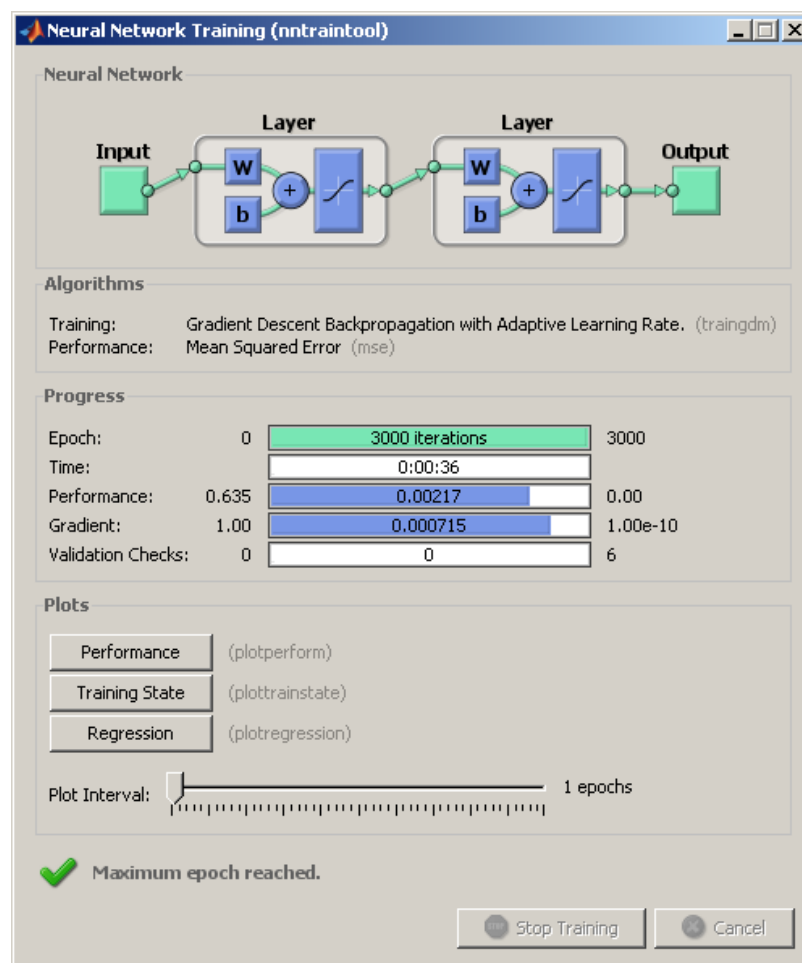
TFi – i-tojo sluoksnio perdavimo funkcija, pagal nutylėjimą 'tansig'

BTF – treniravimo funkcija, pagal nutylėjimą 'trainlm'.

BLF – svorių parinkimo funkcija, pagal nutylėjimą 'learnngdm'.

PF – klaidos funkcija, pagal nutylėjimą 'mse'

Gražina N sluoksnių tiesioginį tinklą.



17 pav. Neural network toolbox

3.4.1. Duomenų apdorojimas

Kai kurioms perdavimo funkcijoms būtina, kad įėjimai ir siekiami pasiekti rezultatai tilptų tam tikrame intervale. Siekiant įvykdyti šį reikalavimą reikia iš anksto apdoroti duomenis.

prestd: apdoroja duomenis taip, kad vidurkis būtų 0 o standartinis nuokrypis būtų 1

Sintaksė:

```
[pn,meanp,stdp,tn,meant,stdt] = prestd(p,t)
```

premmx: apdoroja duomenis taip, kad minimumas būtų -1, maksimumas 1.

Sintaksė:

```
[pn,minp,maxp,tn,mint,maxt] = premmx(p,t)
```

Jei mes iš anksto apdorojam duomenis iki eksperimentų, tada po modeliavimo norint pamatyti rezultatus, turime konvertuoti atgal originaliu masteliu. Atitinkamos atvirkštinės funkcijos: *prestd* ir *postd*, *premmx* ir *postmmx*.

Tarkime turime tokia skaičių eilutę:

```
8 10 9 7 6 6 5 6 8 9 11 7 6 5 4 3 4 7 9 8 9 10 9 8 10
```

Pradžiai padarom lentelę kad pirmam stulpelyje būtų duomenys pagal, kuriuos prognozuosim, o antram norima gauti reikšmę,

```
for i=1:k+1
temp(i,1:s2-k)=data2(i:s2-k+i-1) %padarome matrica (k+1)x(s2-k)
end
```

ir gauname:

```
      8      10
     10      9
      9      7
      7      6
      6      6
      6      5
      5      6
      6      8
      8      9
      9     11
     11      7
      7      6
      6      5
```



```

5      4
4      3
3      4
4      7
7      9
9      8
8      9
9      10
10     9
9      8
8      10

```

Pirmus 15 duomenų rinkinių naudosime apmokymui, likusius 10 testavimui.

```

P=temp(1:s1-k+1,1:k)
T=temp(1:s1-k+1,k+1)
a=temp(s1:s2-k,1:k)
s=temp(s1:s2-k,k+1)

```

Poprogramė *premnmx* naudojama apdoroti duomenis taip, kad jie patektų į intervalą [-1,1]. Reikia atkreipti dėmesį, kad mes turime konvertuoti savo duomenis į eilutes, kad galėtų būti taikomas *premnmx*.

```

[pn,minp,maxp,tn,mint,maxt]=premnmx(P',T');
[an,mina,maxa,sn,mins,maxs]=premnmx(a',s');

```

3.4.2. Tinklo treniravimas

Tinklo sukūrimas su 5 neuronais paslėptajame sluoksnyje, panaudojus tangento sigmoid perdavimo funkciją:

```

net=newff(minmax(pn),[5 1],{'tansig','tansig'},'traingdm')

```

Tinklo parametrų parinkimas:

```

net.trainParam.epochs=3000;%iteracijų skaičius
net.trainParam.lr=0.3;      %mokymo koef.
net.trainParam.mc=0.6;      %momento konstanta
net=train (net,pn,tn);      %tinklo treniravimas

```

Po treniravimo galima atkurti rezultatus:

```

y=sim(net,an)

```

Gaunam:

y =

Columns 1 through 8

```
-0.4962    -0.9773    -0.4962     0.6130     0.4532     0.5299     0.4532
0.0029
```

Columns 9 through 10

```
0.4532    0.5299
```

Kad y reikšmę atversti atgal į originalų dydį, naudojama ši komanda:

```
t=postmmx(y',mins,maxs);
```

Pasižiūrime rezultatus:

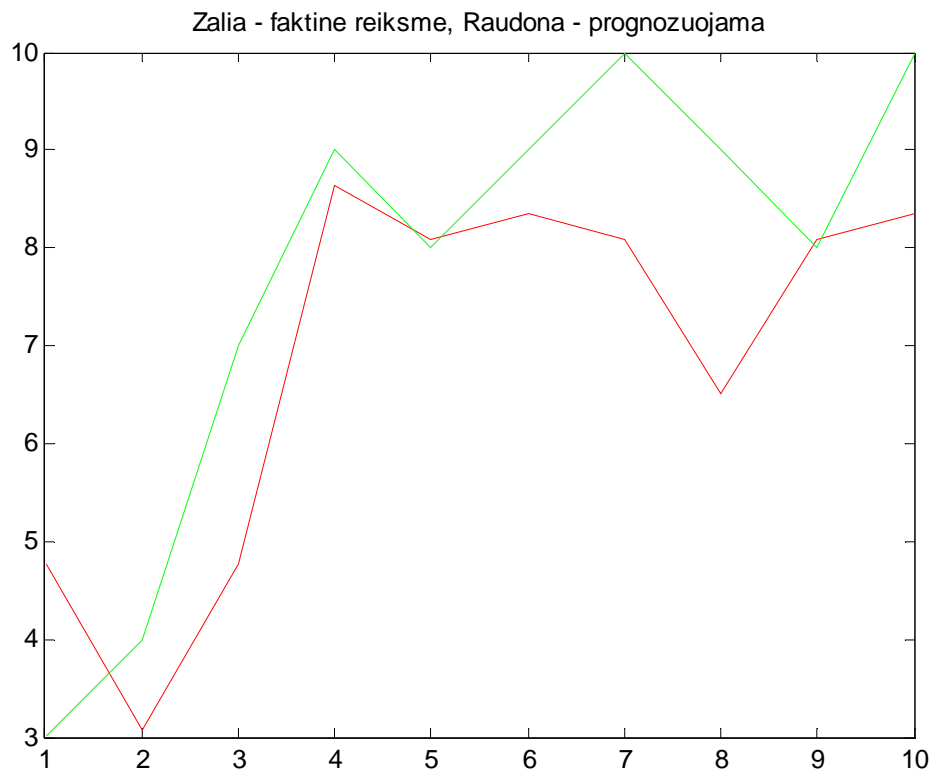
```
[T S]
```

```
ans =
```

```
4.7633    3.0000
3.0795    4.0000
4.7633    7.0000
8.6454    9.0000
8.0861    8.0000
8.3547    9.0000
8.0861   10.0000
6.5100    9.0000
8.0861    8.0000
8.3547   10.0000
```

Atvaizduojame reikšmes:

```
[t s]
plot(t,'r')
hold
plot(s,'g')
hold
title('Zalia - faktine reikšme, Raudona - prognozuojama')
```



18 pav. Reikšmių grafikas

Kad paskaičiuoti vidutinę kvadratinę paklaidą (MSE) ir vidutinę kvadratinės šaknies klaidą (RMSE) reikia atlikti tokius skaičiavimus:

```
d=[t-s].^2;
mse=mean(d)
rmse=sqrt(mse)
```

Gaunam:

mse =

2.2086

rmse =

1.4861

Kad įvertinti mūsų neuroninio tinklo darbą taip pat galime panaudoti ir regresinę analizę:

```
[m,b,r]=postreg(t',s')
```

M – tiesinės regresijos pasvirimas

B – atidėtos tiesinės regresijos Y reikšmės

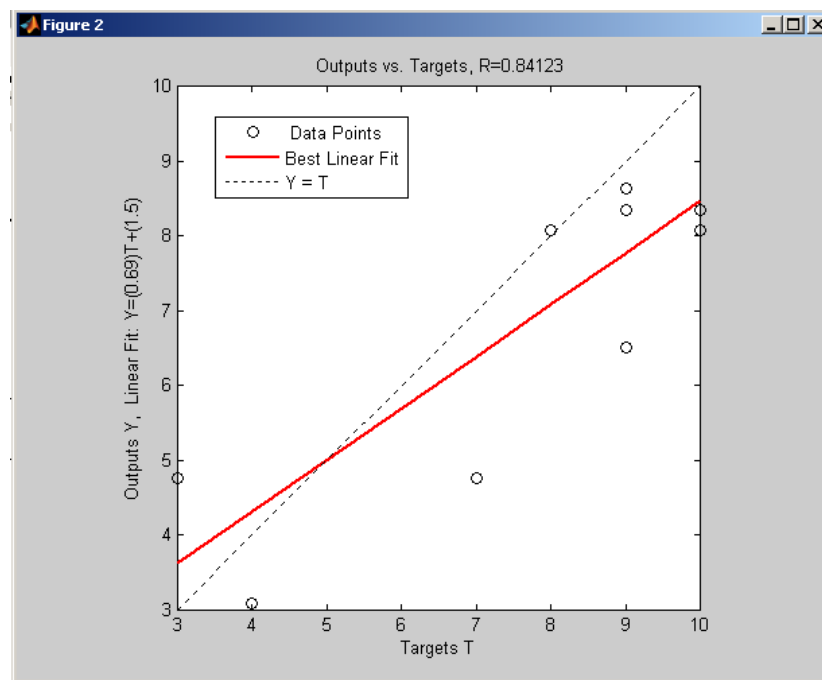
R – regresijos reikšmė (jei R=1 tai reiškia tikslią koreliaciją)

Gaunam reikšmes ir grafiką:

m =
0.6931

b =
1.5357

r =
0.8412

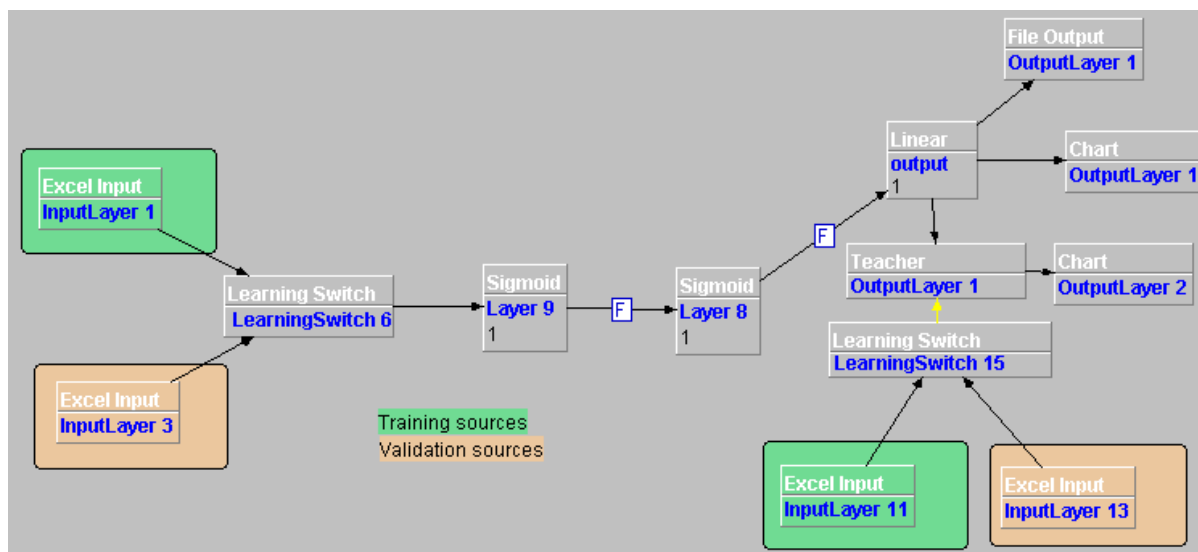


19 pav. Regresijos grafikas

3.5. Joone (Java Object Oriented Neural Engine)

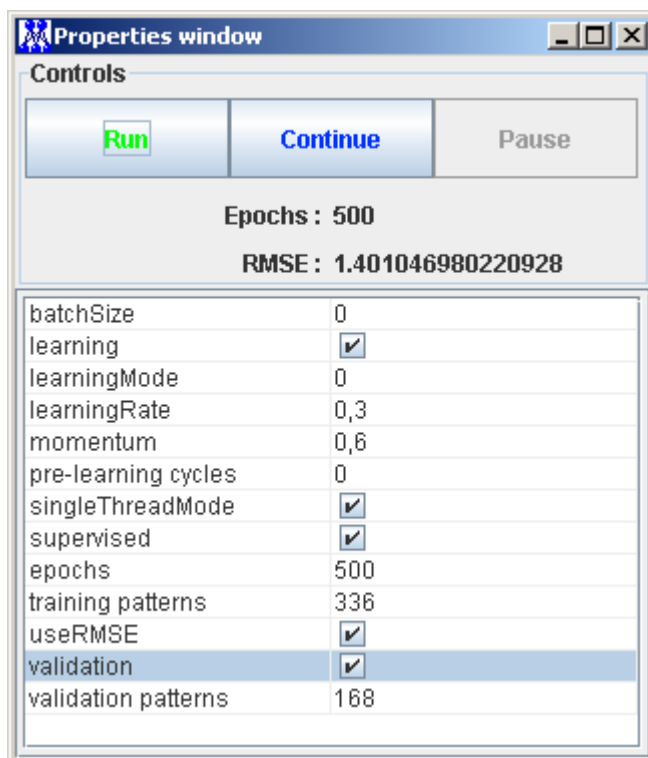
Joone yra nemokama neuroninio tinklo kūrimo, treniravimo ir testavimo sistema. Jos tikslas yra sukurti galingą aplinką tiek mėgėjams, tiek profesionaliems vartotojams, naudojant naujausias Java technologijas. Joone susideda iš centrinio variklio, kuris yra atramos taškas visoms programoms, kurios yra sukurtos Joone pagalba. Joone neuronų tinklai gali būti sukurti vietiniame kompiuteryje, apmokyti tam tikroje aplinkoje ir veikti kitame norimame prietaise. Kiekvienas gali rašyti naujus modulius, siekiant įgyvendinti naujus algoritmus ar panaudoti naujas architektūras. Pagrindinė idėja yra sukurti pagrindą siekiant skatinti dirbtinio intelekto programų kūrimą. [4]

Tinklas sukurtas panaudojus tangento sigmoid perdavimo funkciją, įėjimo elementus (atskiri apmokymo ir patvirtinimo duomenys), perjungiklius, išėjimo elementus (grafinius ir į failą).



20 pav. Joone paketu sumodeliuotas neuroninis tinklas

Tinklo valdymas vyksta per valdymo skydą (Tools->Control Panel):



21 pav. Valdymo langas

Valdymo lango mygtukai:

1 lentelė. Valdymo mygtukai

Run:	Pradedama tinklo darba
Continue:	Tęsia tinklo darba
Stop:	Sustabdo neuroninį tinklą.

Valdymo lango parametrai:

2 lentelė. Valdymo parametrai

BatchSize:	Jei naudojamas paketų režimas tai čia reik įrašyti skaičių nuo 1 iki <i>TrainingPatterns</i> . Šis parametras leidžia nustatyti mišrų režimą, kuriuo tinklas mokinasi nuo 1 iki "batch size" paketų režimu, ir tiesioginiu būdu likusį iteracijų skaičių.
Epochs:	Ciklų skaičius, per kurį išmokomas neuroninis tinklas.
TrainingPatterns:	Įėjimo eilučių skaičius, iš kurio mokinasi tinklas.
ValidationPatterns:	Įėjimo eilučių skaičius, iš kurio patvirtinamas išmokintas tinklas. Naudojamas kartu su <i>Validation</i> parametro nustatyta reikšme <i>true</i> .
Momentum:	Momento reikšmė
Learning Rate:	Mokinimosi koeficientas
Learning:	<i>True</i> jei tinklas apmokomas ir <i>false</i> jei ne
LearningMode	Apmokymo algoritmas 0 = Tiesioginis Backprop (pagal nutylėjimą) 1 = Paketais paremtas Backprop 2 = Elastingas BackProp (RPROP)
Supervised:	<i>True</i> jei tinklas mokomas su priežiūra. Dažniausiai šito parametro nereikia nustatyti, nes programa patikrina ar yra mokytojo komponentas ir nustato parametą pati.
Validation:	<i>True</i> jei tinklas testuojamas su patvirtinimo duomenimis.

	Naudojamas tik kartu su <i>Learning Switch</i> komponentu.
Pre-Learning:	Ciklų skaičius, kurie mokymo metu praleidžiami. Įprastai šis parametras nustatomas 0, ir naudojamas tik tada kai tinkle yra <i>DelayLayer</i> komponentas. Šiuo atveju <i>pre-learning</i> parametras turi būti lygus <i>tap</i> skaičiui, kad buferis užsipildytų reikiamu kiekiu duomenų prieš pradėdant mokymo ciklą.
UseRMSE	Jei <i>true</i> , tai skaičiuojama vidutinė kvadratinės šaknies klaida (RMSE), jei ne tai vidutinė kvadratinė klaida (MSE)

4. MODELIŲ EKSPERIMENTINIS TYRIMAS

Šio eksperimento tikslas patyrinėti neuroninių tinklų elgseną ir palyginti su kitais modeliais. Atliekant eksperimentą, buvo naudojamos įvairių rūšių duomenų eilutės. Duomenys buvo pasirinkti iš skirtingų sektorių:

3 lentelė duomenys

Sektorius	Pavadinimas	Laikotarpis	Sutrumpinimas	Duomenų kiekis	Šaltinis
IT	Intel Corporation	2007.01.01-2008.12.31	intc	504	[3]
Maisto pramonė	Kraft Foods Inc.	2007.01.01-2008.12.31	kft	504	[3]
Obligacijų fondas	Templeton Global Bond A	2007.01.01-2008.12.31	tpinx	504	[3]
Žaliavos	United States Oil	2007.01.01-2008.12.31	uso	504	[3]
Valiutos	Euro-Dollar	2007.01.01-2008.05.18	eurusd	504	[1]
Upės debitas	Amazon	1928-1946 ir 1970-1992	amazon	504	[10]
Virtualūs duomenys	Stock Exchange Model Game 1		semg1	504	[9]
Virtualūs duomenys	Stock Exchange Model Game 2		semg2	504	[9]

Duomenų kiekis paimtas vienodas – 504 reikšmės. Iš jų 336 apmokymui, 168 testavimui. Virtualūs duomenys buvo sugeneruoti naudojant biržos žaidimo modelį. Pirmuoju atveju biržoje dalyvavo Wiener'io tipo žaidėjai, antruoju – autoregresijos. Neuroninis tinklas sukurtas su Matlab buvo bandomas su skirtingu atsimenamų į praeitį reikšmių skaičiumi (Matlab1, Matlab2, Matlab9). Indeksas žymi, iš kelių reikšmių atliekama prognozė. Modelių efektyvumui įvertinti buvo pasirinktas RMSE.

	AR-ABS	Matlab1	Matlab2	Matlab9	Joone	RW
Amazon	9909,27871	23295,9223	11252,46	10414,166	57270,8718	23906,3928
eurusd	0,00596434	0,00628635	0,0068168	0,0075912	0,01314196	0,00596985
intc	0,60588591	0,65398143	0,5970041	0,6118131	1,10190389 (0,6663992)*	0,59364642
kft	0,48094791	0,47764691	0,4820794	0,4899439	0,66398502 (0,5707888)*	0,48042578
tpinx	0,06163321	0,06180526	0,0639438	0,0665873	0,10246529	0,06023268
uso	2,64055519	2,82221603	2,7851995	3,1147905	7,53996168 (3,2403194)*	2,58843835
segm1	0,08711664	0,0887736	0,0971285	0,0920069	0,11287238	0,0872447
segm2	0,09542771	0,0950738	0,0971027	0,0980124	0,1221507	0,09465276
Laimėjimai	3	1	0	0	0	4

AR-ABS – prognozavimas, naudojant AR-ABS paketą

Matlab1 – prognozavimas, naudojant matlab paketą, prognozę atliekant iš paskutinės reikšmės

Matlab2 – prognozavimas, naudojant matlab paketą, prognozę atliekant iš paskutinių 2 reikšmių

Matlab9 – prognozavimas, naudojant matlab paketą, prognozę atliekant iš paskutinių 9 reikšmių

Joone – prognozavimas, naudojant Joone paketą

RW (random walk) – prognozavimas, kai rytdienos prognozė yra šiandienos vertė

4 lentelėje suvestos RMSE reikšmės gautos skaičiavimų metu. Paryškintos mažiausios reikšmės, kurios reiškia, kad tam modeliui geriausiai sekėsi prognozuoti reikšmes. Iš lentelės matosi, kad geriausius rezultatus gavo AR-ABS ir RW modeliai.

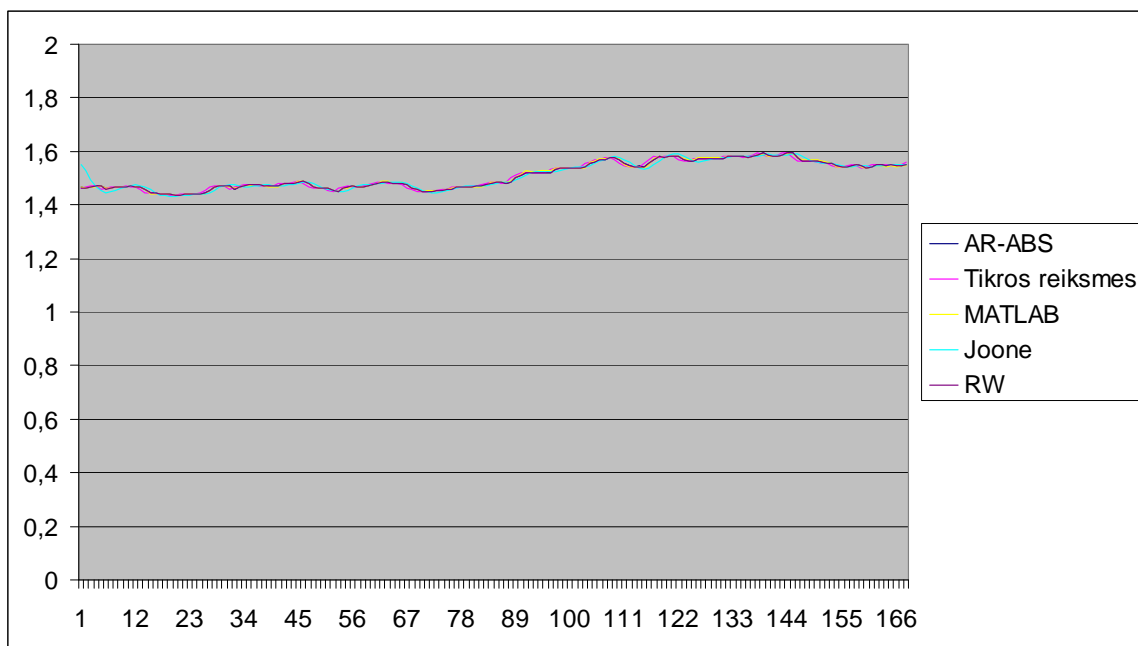
Iš šios lentelės matosi tik tai, kokiam modeliui geriau sekėsi prognozuoti įvairių rūšių duomenis, bet negalima nuspręsti, kurie duomenys prognozuojami lengviau, o kurie sunkiau. To negalima padaryti, nes naudojami duomenys yra pakankamai iš skirtingų diapazonų: pvz amazon reikšmės svyruoja nuo 86109 iki 306317, o euro-dolerio kursas svyruoja 1,4359-1,5952. Tai įtakoja ir tokius pakankamai didelius skirtumus tarp RMSE reikšmių skirtingoms duomenų eilutėms. Tam, kad nuspręsti kurie duomenys lengviau prognozuojami, buvo išreikštos santykinės reikšmės t.y. RMSE procentinė dalis nuo vidutinės prognozuojamo dydžio reikšmės. Rezultatai procentais pateikiami 5 lentelėje.

5 lentelė. RMSE%

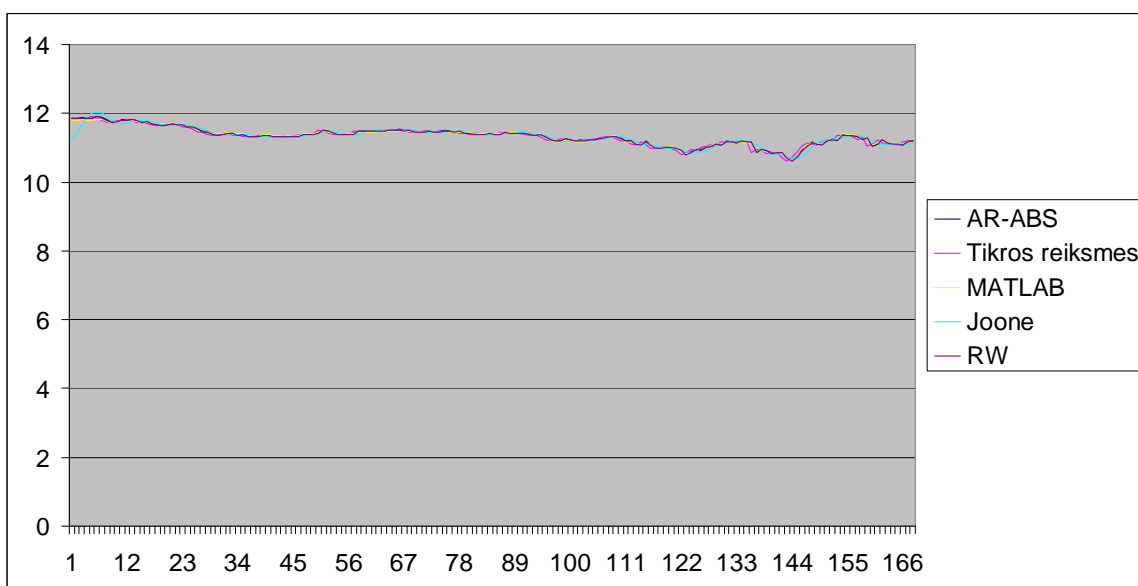
	AR-ABS	Matlab1	Matlab2	Matlab9	Joone	RW
Amazon	5,983	14,064	6,793	6,287	34,576	14,433
eurusd	0,395	0,416	0,451	0,503	0,870	0,395
intc	3,141	3,391	3,095	3,172	5,713	3,078
kft	1,439	1,429	1,442	1,466	1,986	1,437

tpinx	0,544	0,546	0,565	0,588	0,905	0,532
uso	3,305	3,532	3,486	3,898	9,437	3,240
segm1	1,471	1,499	1,641	1,554	1,906	1,474
segm2	3,635	3,622	3,699	3,734	4,654	3,606

Iš šios lentelės galima spręsti, kad geriausiai visiems modeliams sekėsi prognozuoti euro-dolerio kursą, paskui obligacijų fondo kursą. Iš grafikų matosi, kad šių duomenų išsibarstymas yra pakankamai mažas.



22 pav. eurUSD grafikas



23 pav. tpinx grafikas

4 ir 5 lentelėse pateiktuose rezultatuose neatsispindi modelių efektyvumas atsižvelgiant į duomenų išsibarstymą (dispersiją):

$$D(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (24)$$

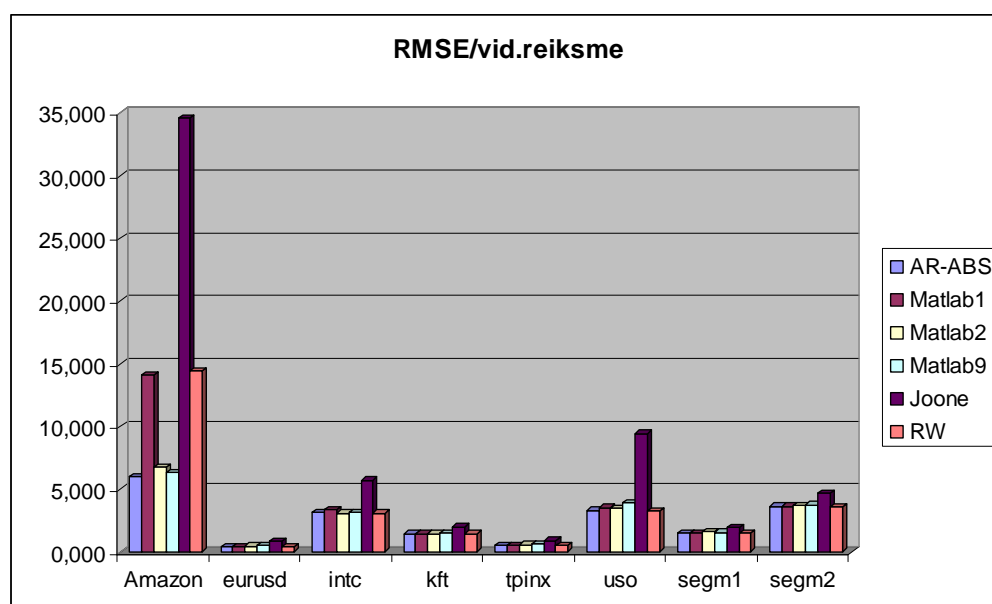
Pagal 5 lentelę matosi, kad modeliams geriau pavyksta prognozės mažiau issibarsčiusiems duomenims (eurusd ir tpinx). Tam, kad objektyviai įvertinti modelių efektyvumą buvo pasirinktas RMSE klaidos ir šaknies iš dispersijos santykis išreikštas procentais.

6 Lentelė. RMSE/D(x)%

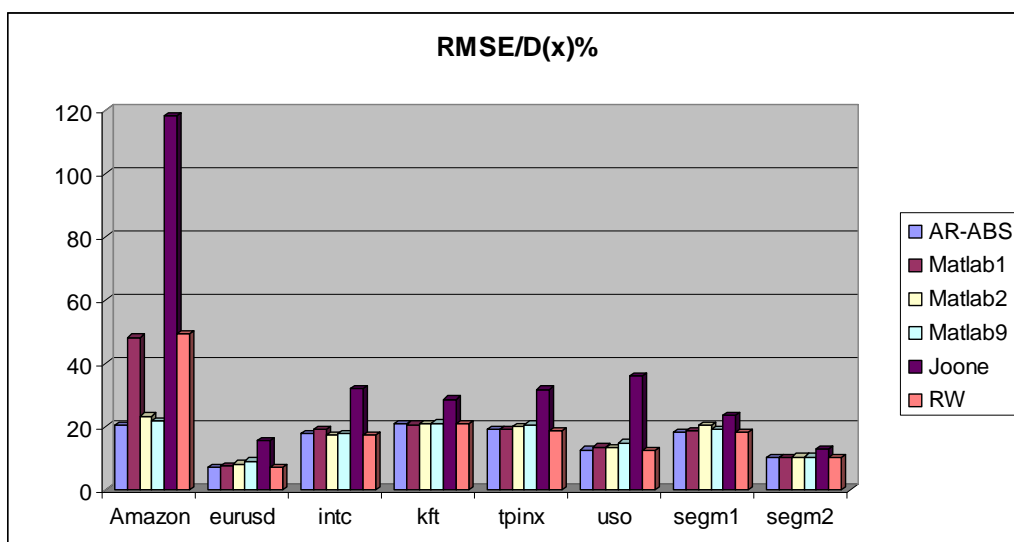
	AR-ABS	Matlab1	Matlab2	Matlab9	Joone	RW
Amazon	20,50131184	48,1969457	23,280221	21,545873	118,487736	49,4599484
eurusd	7,045390783	7,4257683	8,0523499	8,9671041	15,5239775	7,05190001
intc	17,54500182	18,9377326	17,287807	17,71664	31,9084923	17,1905757
kft	20,6639285	20,5221008	20,712543	21,050443	28,5281186	20,641495
tpinx	19,078932	19,1321894	19,794178	20,612486	31,7187477	18,6453904
uso	12,56074052	13,424875	13,248793	14,81661	35,8665111	12,3128282
segm1	18,14018603	18,4852107	20,224938	19,158479	23,5032699	18,1668514
segm2	10,10089348	10,0634319	10,278183	10,374483	12,9294851	10,0188658

Iš šios lentelės matosi, kad euro-dolerio kursas išliko lengviausiai prognozuojamas, bet obligacijas (tpinx) aplenkė nafta (uso), abu virtualios biržos sugeneruoti duomenys (segm1, segm2), ir dar Intel (intc) akcijos. Taip pat pastebėta, kad sumažėjo skirtumas tarp amazonės debito (amazon) prognozės ir kitų akcijų. Pvz. prognozuojant amazon su AR-ABS (20,50) gaunami panašūs santykiai kaip ir prognozuojant kft (20,66) ar tpinx (19,07).

Grafiškai šie sulyginimai atrodo taip:



24 pav. RMSE/vid.reikšmė, %



25 pav. RMSE/D(x), %

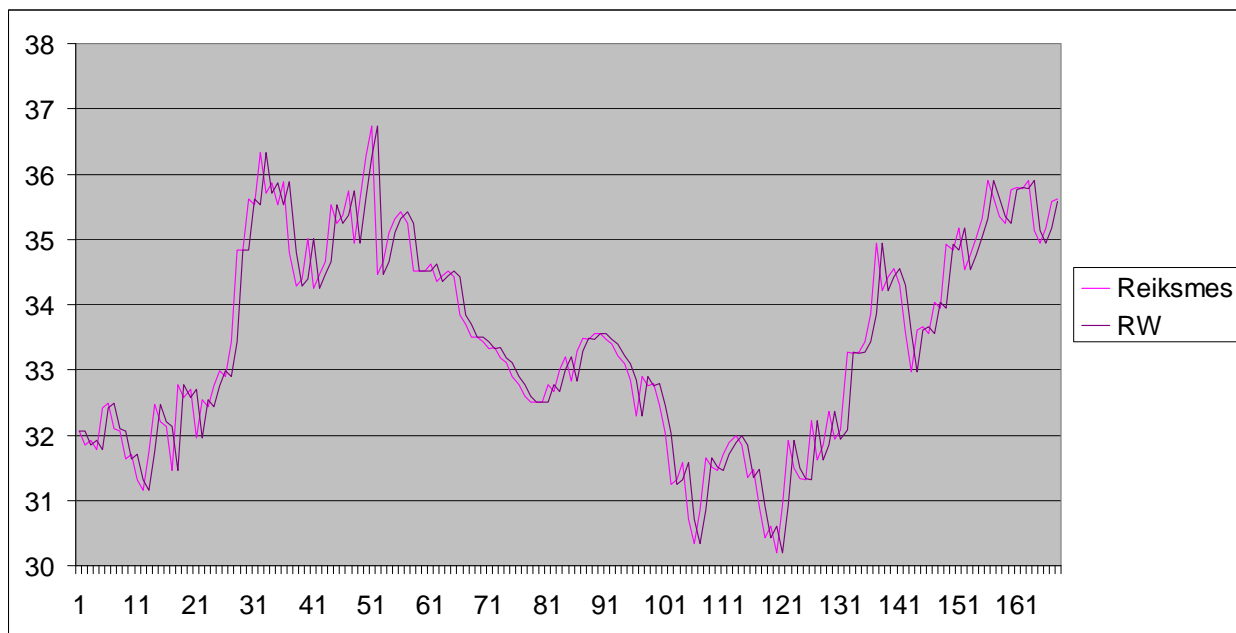
4.1. Išsamesnė peržiūra

Išsamesniam tyrimui pasirinktos Kraft Foods Inc. akcijos, kurias prognozuoti pakankamai vienodai sekėsi visiems modeliams, ir Amazonės upės debito mėnesinės reikšmės kurias prognozuoti modeliams sekėsi pakankamai skirtingai.

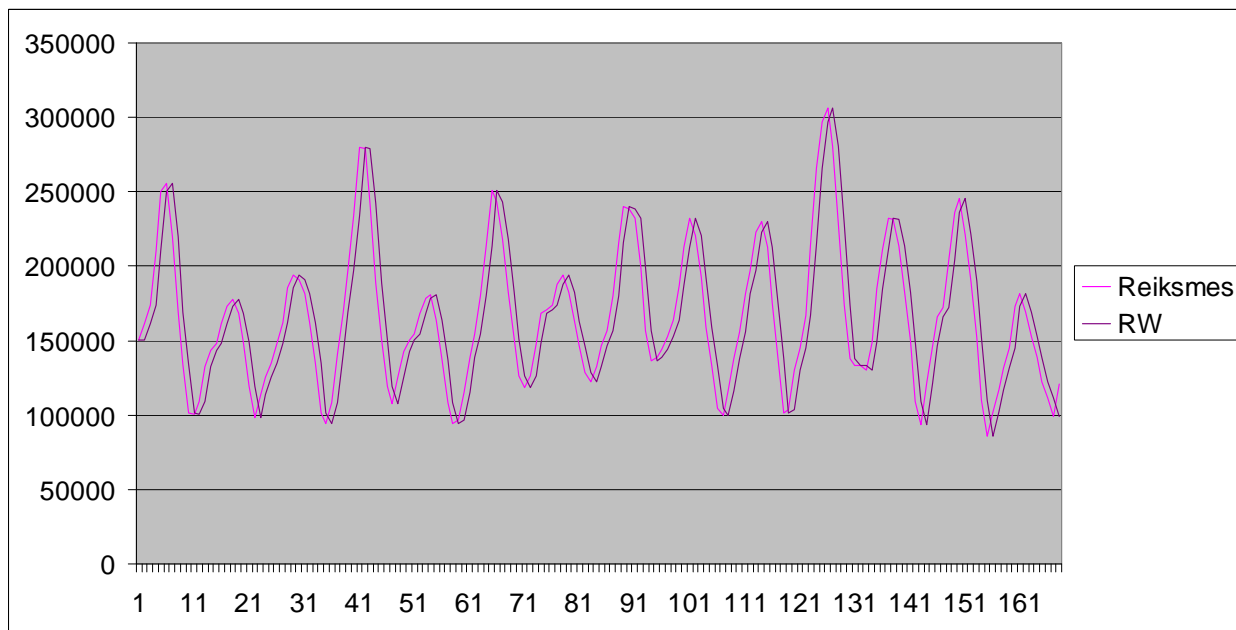
7 Lentelė. amazon ir kft RMSE

	AR-ABS	Matlab1	Matlab2	Matlab9	Joone	RW
Amazon	9909,27871	23295,9223	11252,46	10414,166	57270,8718	23906,3928
Amazon%	5,983	14,064	6,793	6,287	34,576	14,433
kft	0,48094791	0,47764691	0,4820794	0,4899439	0,66398502 (0,5707888)*	0,48042578
kft%	1,439	1,429	1,442	1,466	1,986	1,437

Paprasčiausias ir mažiausiai analizės reikalaujantis metodas yra RW, kuris grafiškai tik pasislenka per vieną laiko reikšmę į dešinę pusę.

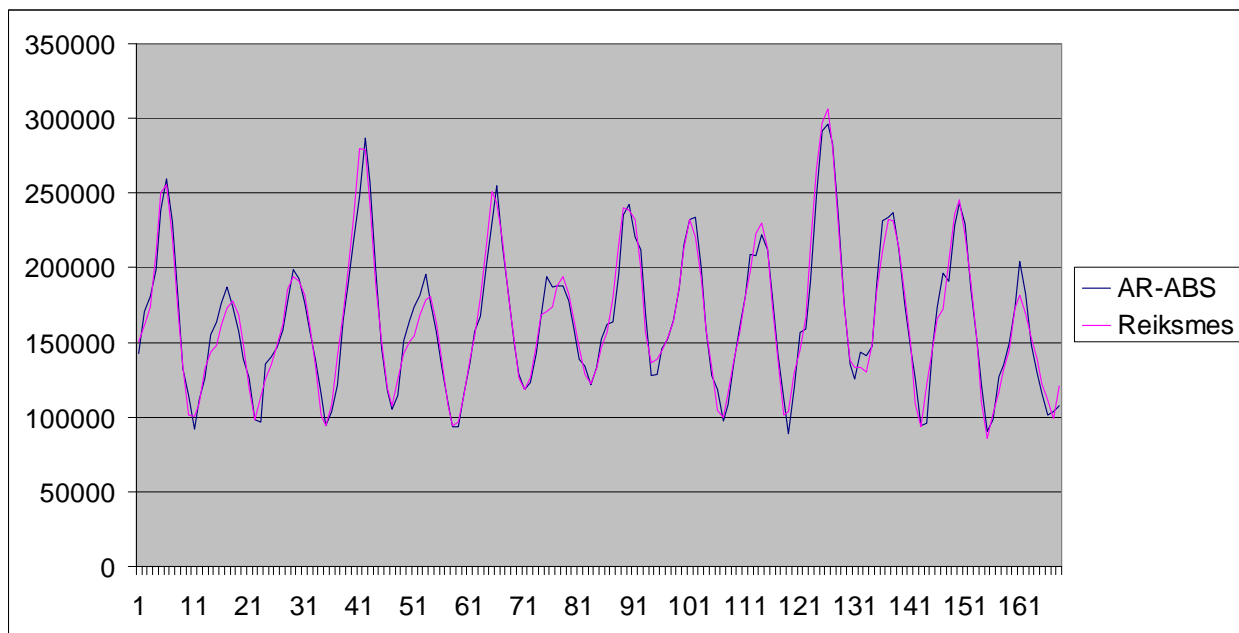


26 pav. kft grafikas RW metodu



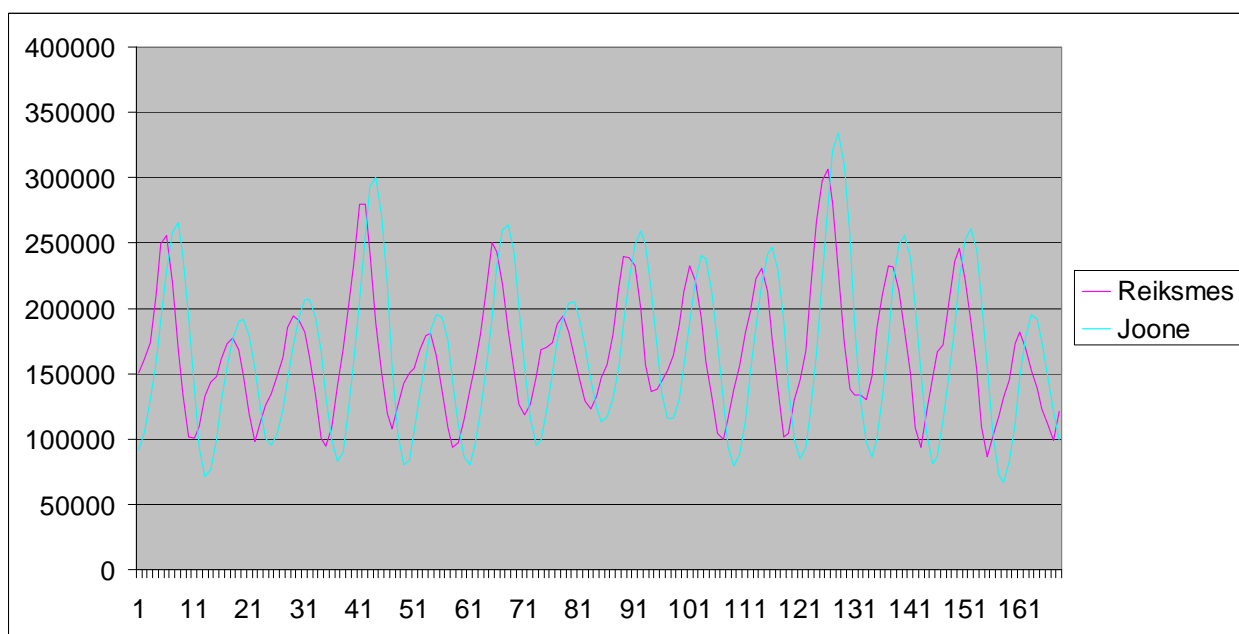
27 pav. amazon grafikas RW metodu

Atliekant bandymą su AR-ABS metodu išryškėja šio pranašumas atliekant upės debito prognozę. Matosi gan didelis spėjimų ir tikrųjų reikšmių atitikimas.



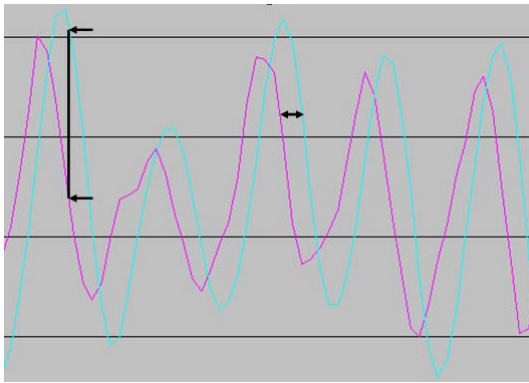
28 pav. amazon grafikas AR-ABS metodu

Prasčiausius rezultatus parodė neuroninis tinklas sukurtas Joone paketo pagalba.

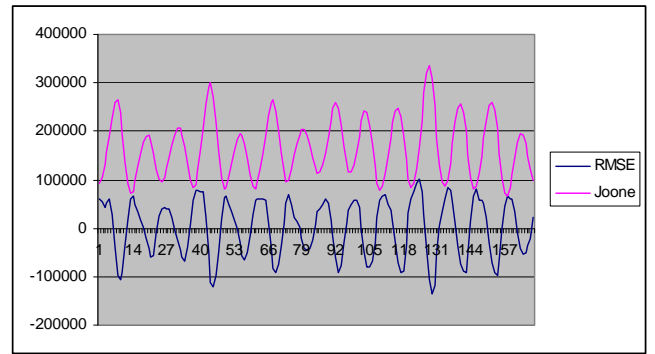


29 pav. amazon grafikas Joone metodu

Matosi gan ryškus vėlavimas ir tas iššaukia dideles klaidas kiekviename žingsnyje (RMSE = 57271).

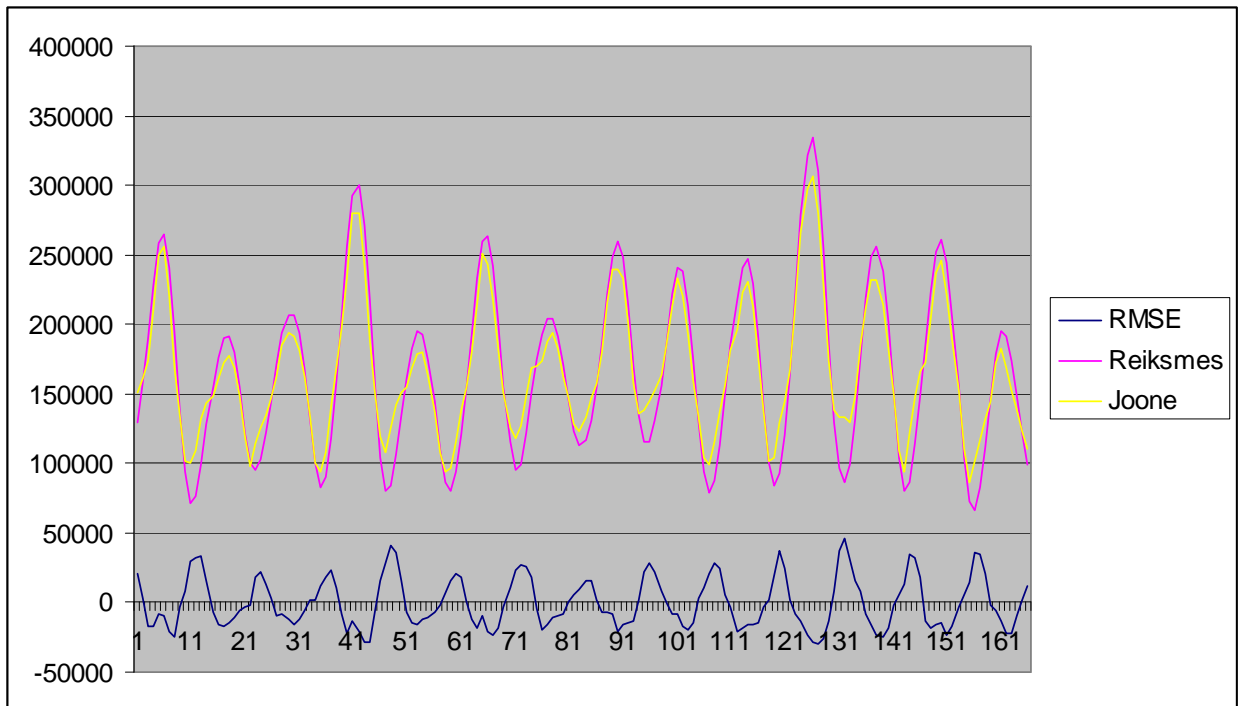


30 pav. Vėlavimas



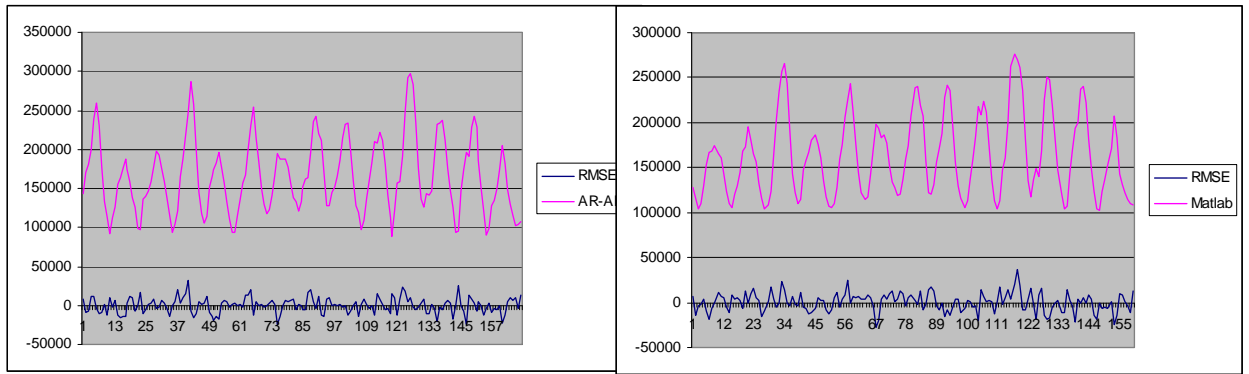
31 pav. Klaida

Eliminavus šį vėlavimą gaunamas žymiai geresnis rezultatas $RMSE = 17932$.



32 pav. Pakoreguota Joone prognozė

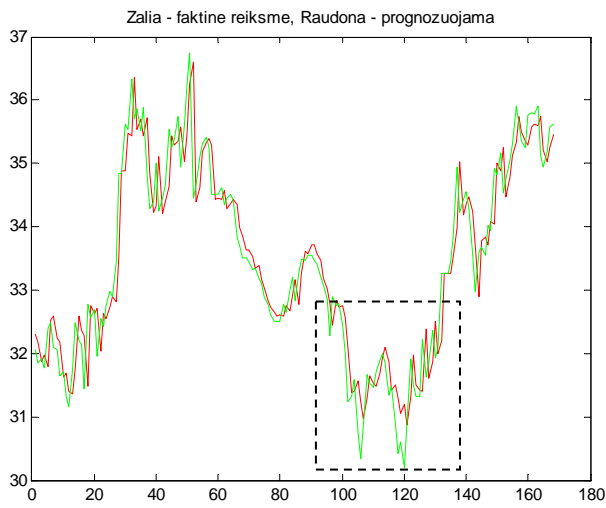
Iš grafiko matyti, kad pakoreguota prognozė gauna žymiai geresnį rezultatą ($RMSE = 17932$, prieš koregavimą $RMSE = 57271$), bet vistiek atsilieka nuo AR-ABS (9909) ar Matlab (10414).



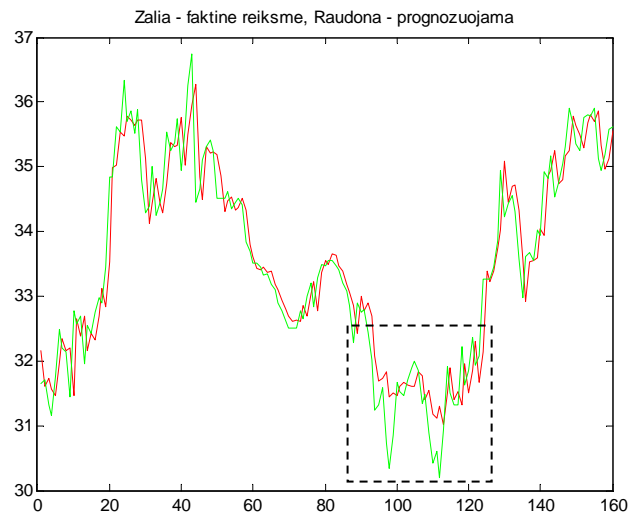
33 pav. amazon grafikas AR-ABS metodu

34 pav. amazon grafikas Matlab metodu

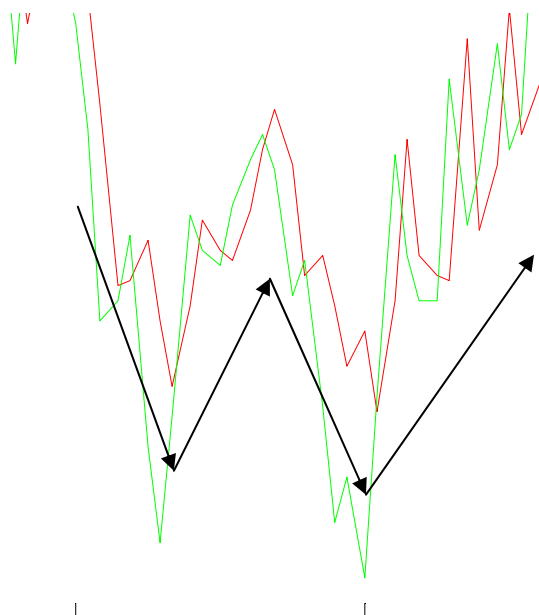
Tolesniuose grafikuose - kft akcijų prognozavimas su Matlab įrankiu. Juose matosi, kad didinant atsimenamų į praeitį reikšmių sunkiau sekasi prognozuoti tose vietose kur svyravimai ypatingai dideli tiek į vieną pusę, tiek į kitą.



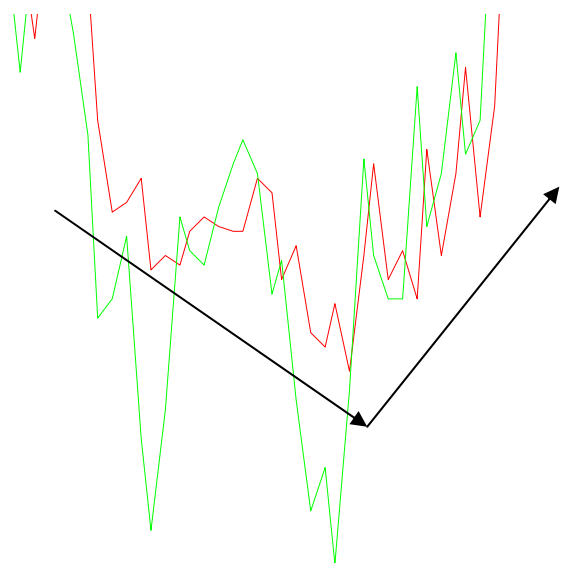
35 pav. kft grafikas Matlab1 metodu



37 pav. kft grafikas Matlab9 metodu



36 pav. Matlab1 metodo jautrumas



38 pav. Matlab9 metodo jautrumas

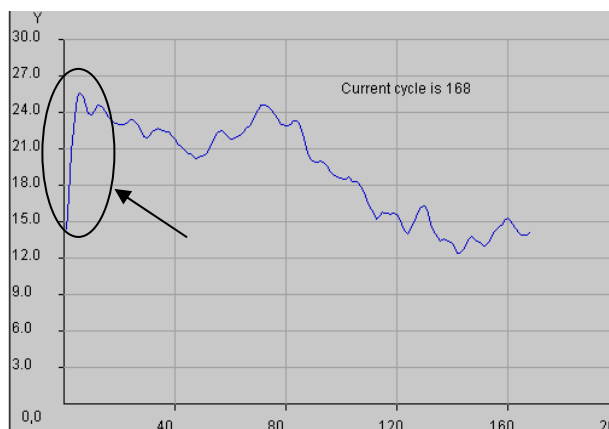
Iš šių grafikų galima daryti išvadą, kad neuroniniams tinklams sėkmingiau sekas atlikti prognozes su labiau išsibarsčiusiais duomenis (kft) atsimenant mažiau praeities reikšmių. O su tolygiai kintančia duomenų eilute neuroniniam tinklui geriau sekas didinant atsimenamų reikšmių skaičių. Amazonės upės debito prognozės įvertinimo rodiklis mažėjo (23296->11252->10414) didinant reikšmių skaičių (1->2->9), iš kurių buvo atliekama prognozė.

Atliekant eksperimentą su Joone buvo pastebėtas ne tik prognozių vėlavimas, bet ir pirmųjų prognozių neatitikimas.



39 pav. kft grafikas su Joone

Toks reiškinys pastebėtas ir su kitais duomenimis:



40 pav. intc grafikas su Joone



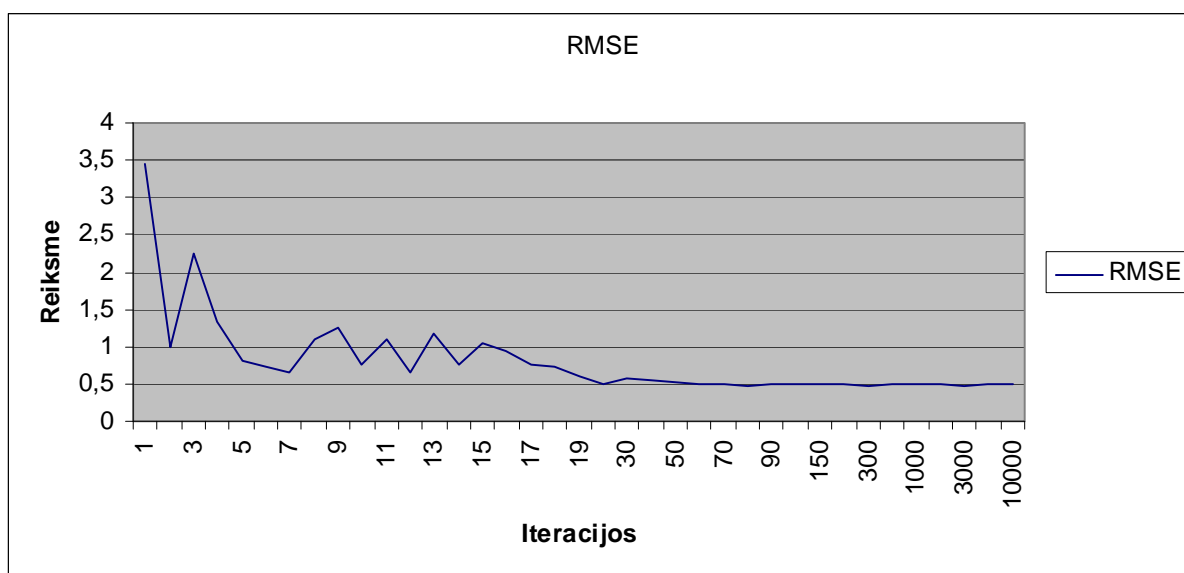
41 pav. uso grafikas su Joone

Eliminavus šias neteisingas prognozes pasikeičia ir Joone rodikliai:

8 Lentelė Joone RMSE

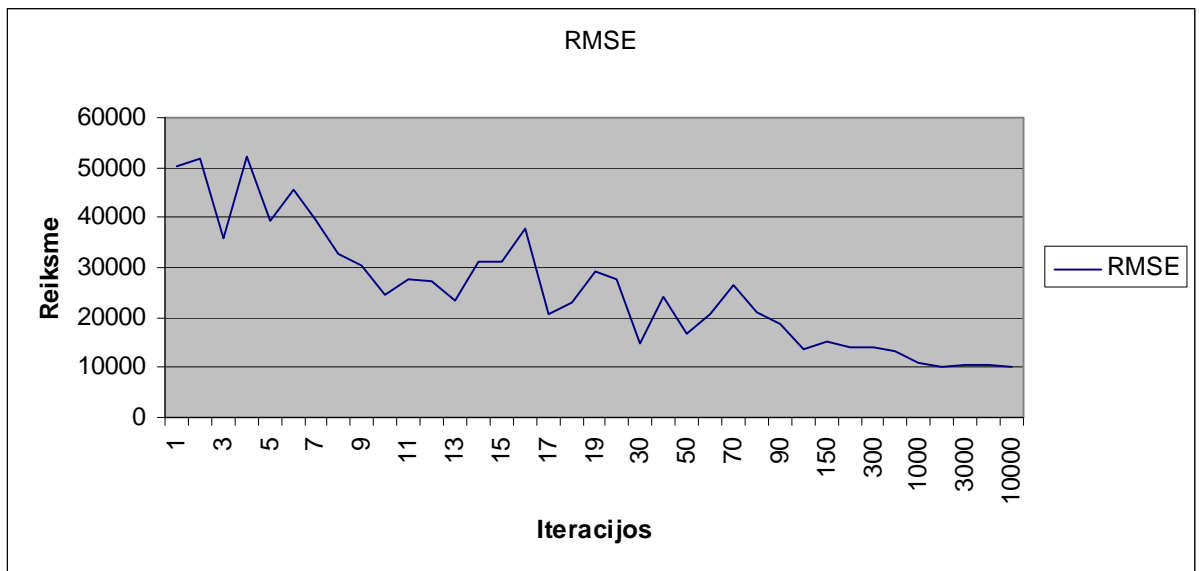
Akcija	Prieš pakeitimus	Po pakeitimų
intc	1,10190389	0,6663992
kft	0,66398502	0,5707888
uso	7,53996168	3,2403194

Naudojant neuroninius tinklus atlikti prognozes taip pat svarbu atsižvelgti į laiką, reikalingą tam tinklui išmokinti. 42 paveiksle matoma kft akcijų RMSE. Pakankamai tikslus rezultatas pasiekiamas jau po 20 iteracijų.



42 pav. kft RMSE nuo iteracijų skaičiaus

Tuo tarpu prognozuoti amazon duomenis žymiai sunkiau. Geras rezultatas pasiekiamas tik po 1000 iteracijų.



43 pav. amazon RMSE nuo iteracijų skaičiaus

Iš šių grafikų galima spręsti, kad apmokyti tinklą prognozuoti fizinius mėnesinius duomenis reikia 50 kartų daugiau laiko negu finansinius.

IŠVADOS

- Ištirtas ir sutvarkytas biržos žaidimo modelis (SEGM), su kuriuo buvo generuojami duomenys tolimesniam tyrimui.
- Darbe sumodeliuoti 2 neuroniniai tinklai dviem skirtingom priemonėm. Naudotos priemonės: komercinis produktas MatLAB kartu su integruotu Neural network tool įrankiu ir nemokama JOONE (Java Object Oriented Neural Engine) programa.
- Neuroninių tinklų elgsenai pasirinkti duomenys iš skirtingų sričių: IT, maisto pramonė, obligacijų fondas, žaliavos, valiutos, upės debitas ir virtualūs duomenys gauti, panaudojus SEGM. Duomenų imtis vienoda visiems duomenims: 504 reikšmės, iš jų 336 panaudotos apmokymui, 168 – testavimui.
- Modelių efektyvumui įvertinti pasirinkta vidutinė kvadratinės šaknies klaida (RMSE). Gauti rezultatai buvo sulyginami su AR-ABS ir RW metodais. Iš gautų rezultatų galima spręsti, kad neuroniniai tinklai parodė prastesnius rezultatus nei AR-ABS ir RW metodai. Bet neuroniniai tinklai kai kuriais atvejais pranoko šiuos metodus, kas leidžia manyti, kad atitinkamai pritaikius neuroninį tinklą galima gauti geresnes prognozes negu AR-ABS ir RW metodų.
- Norint nustatyti kuriuos duomenis, modeliams geriau sekasi prognozuoti, buvo paskaičiuotas santykis tarp RMSE ir vidutinės duomenų reikšmės, kas leido pamatyti, kad lengviausiai prognozuojami yra mažiausiai kintantys kursai (euro-dolerio, obligacijų fondo)
- Norint objektyviai įvertinti duomenų prognozes teko atsižvelgti ir į duomenų išsibarstymą. Apskaičiavus RMSE ir šaknies iš dispersijos santykį, nustatyta, kad lengviausiai prognozuojamas valiutų kursas. Taip pat pastebėta, kad sumažėjo skirtumas tarp amazonės debito (amazon) prognozės ir kitų akcijų
- Išsamiau patyrinėjus prongozių ir realių duomenų grafikus pastebėti JOONE programa sumodeliuoto neuroninio tinklo elgsenos trūkumai. Prognozuojant pagal amazon duomenų eilutę pastebėtas pakankamai ryškus vėlavimas (Pav. 29 ir Pav. 30). Taip pat pastebėtas pirmųjų prognozių gan ryškus neatitikimas (Pav. 39 - Pav. 41).
- MatLAB paketu išbandytas tinklo jautrumas didelio duomenų svyravimo vietose (Pav. 35 - Pav. 38). Taip pat išbandyta RMSE priklausomybė nuo iteracijų skaičiaus.

5. LITERATŪRA

1. FXHistory – Historical Currency Exchange Rates [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://www.oanda.com/convert/fxhistory>
2. Guidelines for Financial Forecasting with Neural Networks [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: http://www2.cs.uregina.ca/~jtyao/Papers/guide_iconip01.pdf
3. Yahoo! Finance – Business Finance, Stock Market, Quotes, News [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://finance.yahoo.com/>
4. Joone: neural networks, Java implementation [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://proin.ktu.lt/~mockus/>
5. Mockus, J. A set of examples of global and discrete optimization 2 [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://proin.ktu.lt/~mockus/>
6. Mockus, J. Raudys, A. On the Efficient-Market Hypothesis and Stock Exchange Game Model 2009.03.15
7. Nash Equilibrium, examples of global optimization in a simplest stock exchange model [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://proin.ktu.lt/~mockus/>
8. Prediction and Diagnosis by AR-ABS GMJ2 that Minimizes Absolute Deviations [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: <http://proin.ktu.lt/~mockus/>
9. The Global River Discharge Database (RivDIS v1.1)
10. Wikipedia, The free encyclopedia [interaktyvus]. [žiūrėta 2009.05.19]. Prieiga per internetą: http://en.wikipedia.org/wiki/Main_Page

6. TERMINŲ IR SANTRUMPŲ ŽODYNAS

amazon – Amazonės upės debito istoriniai duomenys

ANN (Artificial Neural Networks) – Dirbtiniai neuroniniai tinklai

AR – Auto regresija

AR-ABS (Auto regressive absolute values) – Auto regresijos mažiausių absoliutinių nukrypimų modelis

ARMA (Auto Regressive Moving Average) – Auto regresijos slenkančio vidurkio modelis

EMH (Efficient-Market Hypothesis) – Efektyvios rinkos hipotezė

eurusd – Euro-Dolerio keitimo kursas

intc (Intel Corporation) – Intel korporacijos akcijų kursas

JOONE (Java Object Oriented Neural Engine) – Java orientuotas neuroninis variklis

kft (Kraft Foods Inc.) – Kraft Foods korporacijos akcijų kursas

MAE (Mean Average Error) – Vidutinė absoliučioji klaida.

MSE (Mean Squared Error) – Vidutinė kvadratinė klaida

NE (Nash Equilibrium) – Nešo pusiausvyra

NT – Neuroninis tinklas

RMSD (Root Mean Squared Deviation) – Vidutinis kvadratinės šaknies nuokrypis

RMSE (Root Mean Squared Error) – Vidutinė kvadratinės šaknies klaida

RW (Random Walk) – atsitiktinės eigos modelis

SEGM (Stock Exchange Game Model) – Biržos žaidimo modelis

tpinx (Templeton Global Bond A) – Templeton obligacijų fondo kursas

uso (United States Oil) – JAV naftos fondo kursas