

VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS

INFORMATIKOS KATEDRA

RIMVYDAS BUTLERIS

MAGISTRO DARBAS

Kreditų vertinimo tyrimai neuroniniais tinklais

KAUNAS, 2007

VILNIAUS UNIVERSITETAS
KAUNO HUMANITARINIS FAKULTETAS
INFORMATIKOS KATEDRA

RIMVYDAS BUTLERIS

VERSLO INFORMATIKOS MAGISTRO STUDIJŲ PROGRAMA

Kreditų vertinimo tyrimai neuroniniais tinklais

Magistro darbas

Leidžiama ginti _____
Prof. habil dr.

Magistrantas _____

Darbo vadovas: Dr. doc. Gintautas Garšva

Darbo įteikimo data _____

Registracijos Nr. _____

Kaunas, 2007

TURINYS

TURINYS.....	3
ĮVADAS.....	4
1. KREDITŲ VERTINIMO PRINCIPAI	5
1.1. Vertinimo metodai	6
1.2. Diskriminantų analizė	6
1.3. Regresija	6
1.4. Kuris metodas geriausias?.....	7
2. NEURONINIŲ TINKLŲ TAIKYMO FINANSŲ VALDYMUI ANALIZĖ.....	8
2.1. NeuroXL klasifikatorius-neuroninių tinklų finansų pritaikymas-kreditai ir paskolos.....	9
2.3. Kreditų rizikos sprendimai paremti neuroniniais tinklais	11
2.4. Kreditų rizikos analizė naudojantis ANN	13
3. BENDRIEJI TINKLO APMOKYMO ALGORITMAI.....	20
3.1. Atgalinio sklaidimo neuroninis tinklas	20
4. BENDRŲJŲ TINKLO APMOKYMO ALGORITMŲ EKSPERIMENTINIS TYRIMAS	23
4.1. batbp funkcija.....	23
4.2. igls funkcija.....	25
4.3. incbp funkcija.....	27
4.4. marq funkcija	30
4.5. marqlm funkcija	32
4.6. rpe funkcija	32
IŠVADOS.....	36
LITERATŪROS SĄRAŠAS.....	37
PRIEDAI	40

ĮVADAS

Kreditų rizikos analizės vertinimas yra labai sunki ir svarbi finansų analizės valdymo problema. Šią problemą paliečia daugybė klasifikacinių metodų. Ypač neuroniniams tinklams skirta daug dėmesio dėl jų universalios apytikslės derinimo savybės. Tačiau didžiausias trūkumas sprendimų priėmimo naudojantis neuroniniais tinklais yra jų mažos aiškumo galimybės. Kol kas jie negali paaiškinti kaip jie pasiekia aukštą numatymo teisingumo rodiklį. Neuroninių tinklų sprendimų aiškinimas, pagal aiškinimo taisyklės, kurios apima išmoktas žinias įtvirtintas tinkluose, gali padėti kreditų rizikos valdytojui aiškinant kodėl konkretus kandidatas yra klasifikuojamas į blogų arba gerų kandidatų klases. Išvadoje galima pasakyti, jog neuroninių tinklų gavybos taisyklė ir sprendimų lentelė yra galingi valdymo įrankiai kurie leidžia mums sukonstruoti pažangias ir patogias vartotojui sprendimo priėmimo sistemas kreditų rizikos įvertinime.

Tema: kreditų vertinimo tyrimai neuroniniais tinklais.

Problema: kokią įtaką kreditų vertinimui turi neuroniniai tinklai?

Objektas: kreditų vertinimo modelis.

Dalykas: vidiniai kreditų vertinimai bankų sistemoje

Tikslas: iširti kreditų vertinimą naudojant neuroninius tinklų pagalba

Uždaviniai:

- Išanalizuoti kreditų vertinimo neuroniniais tinklais modelius, nustatyti esmines detales.

Metodai:

Duomenų rinkimo metodai:

Mokslinės literatūros kaupimas ir išsami analizė.

Duomenų analizės metodas:

Surinktų duomenų analizė bus atliekama Matlab programinės įrangos pagalba.

1. KREDITŲ VERTINIMO PRINCIPAI

Kreditų vertinimas tai kreditų rizikos įvertinimo metodas, paskolų suteikimuose. Naudojantis istoriniais duomenimis ir statistine technika, kreditų vertinimas bando izoliuoti įvairių vartotojų charakteristikas nusikalstamumo ir išsipareigojimų nevykdymo atveju. Šis metodas duoda „vertinimus“, kuriuos bankas gali naudoti jo paskolų vartotojų ar skolintojų klasifikacijai, rizikos terminuose. Norėdami sukurti vertinimo modelį, kūrėjai analizuoja buvusių duotų paskolų realizacijos duomenis, kad galėtų nustatyti, kurių skolininkų charakteristikos yra naudingos, nustatant ar paskola suteikta ir realizuota tinkamai. Geras modelis turėtų suteikti aukštesnį procentą aukštų rezultatų skolininkams, kurių paskolos realizuojamos tinkamai ir didesnę procentą žemų rezultatų, kurių paskolos nebus realizuojamos tinkamai. Tačiau nė vienas modelis nėra tobulas ir keli blogi apskaičiavimai duos didesnius skaičius nei geri apskaičiavimai. Informacija apie skolininką gaunama iš jo paskolos prašymo. Duomenys, tokie kaip kandidato mėnesinės pajamos, nesumokėtos skolos, turimos lėšos, kiek ilgai asmuo dirba vienoje darbo vietoje, ar kada nors yra ėmęs paskolą ir nevykdęs išsipareigojimų, ar nuomoja gyvenamąjį būstą ar yra jo savininkas, ir kokias banko sąskaitas turi, yra potencialūs faktoriai, leidžiantys numatyti paskolos panaudojimą ir ją įvertinti. Regresijos analizė, siejanti paskolos panaudojimą su šiais faktoriais, yra naudojama parinkti, kuri faktorių kombinacija geriausiai numato išsipareigojimų nevykdymą ir, kiek svarbos turi būti suteikta kiekvienam iš šių faktorių. Kuomet duodama koreliacija tarp faktorių, visai tikėtina, kad kai kurie iš faktorių, kuriais kūrėjas pradeda, neprives jo iki galutinio modelio, kadangi jie turi mažai pridėtinės vertės kitiems modelių variantams. Iš tiesų, pasak Fai, Isaac ir Company, Inc., vadovaujantis vertinimo modelių dizaineris gali apsvarstyti 50 ar 60 variantų, kol išplėtos tipišką modelį, bet nuo 8 iki 12 gali baigtis galutiniam vertinimo modelyje, kaip lanksti ir geriausiai nuspėjama kombinacija. Daugumoje vertinimo sistemų, aukštesnis vertinimas nurodo žemesnę riziką ir paskolos davėjas nustato ribą paremtą rizikos kiekiu, kurią jis gali priimti. Griežtas prijungimas prie modelio, skolintojas pavirtintų kandidatus vertinimais virš ribos ir atmestų kurie žemiau ribos.. Netgi gera vertinimo sistema nenumatys su tikslumu bet kurio asmens paskolos realizavimo, bet turėtų suteikti apytikslį numatymą to kas gali būti tikėtina, kad kandidatas su tikslia charakteristika nevykdys išsipareigojimų. Norint sukurti gerą vertinimo modelį, kūrėjams reikia daug istorinių duomenų, kurie atspindi paskolų realizavimą geromis ir blogomis ekonominėmis sąlygomis. Kai jau modelis suskurtas pagal statistinį prašymą, modelis gali būti panaudotas naujų duomenų nustatymui. Modelio naudojimo procesas yra skirtingas nuo kūrimo proceso. Tipiškai, modelis yra naudojimas daug kartų po sukūrimo, tam, kad nustatyti skirtingas duomenų bazines. Per modelio kūrimo procesą, statistinis prašymas naudotą informaciją apie jau esančius klientus, tam kad pastatyti ir ratifikuoti modelį. Galiausiai rezultatas yra modelis kuris paimitų detales apie klientus kaip įeigą ir sukurtų išeigą [14].

1.1. Vertinimo metodai

Istoriškai diskriminantų analizė ir tiesinė regresija buvo plačiausiai naudojama technika vertinimo modelių kūrimo. Abi turi nuopelnų būdamos konceptualiai nesudėtingos ir plačiai prieinamos statistinių programų pakuotėse. Tipiškai bruožų koeficientai ir skaitiniai vertinimai yra sukombinuoti pirmiems įnašams, kurie yra pridedami gaunant visą vertinimą. Kitos technikos kurios buvo naudojamos pramonėje įskaitant ir strateginę regresiją, bandymų analizę, vienalyčius nekriterinius metodus, matematinį programavimą, Markovo grandinės modelį, rekursinis skirstymas, ekspertų sistemos, genetiniai algoritmai, neuroniniai tinklai ir sąlyginės nepriklausomybės modelis [14].

1.2. Diskriminantų analizė

Pagrindinis diskriminantų analizės tikslas yra numatyti grupės narystę, remiantis tiesinėmis intervalų kintamųjų kombinacijomis. Procedūra prasideda su stebėjimų įtvirtinimu, kur kartu grupės narystė ir intervalų kintamųjų vertė yra žinomi. Galutiniai procedūros rezultatai yra modelis, leidžiantis numatyti grupės narystę, vien tik kai žinomos intervalų variacijos. Antrasis diskriminantų analizės tikslas yra pateiktų duomenų supratimas, kaip smulki numatymo modelio analizė, kurios rezultatai gali suteikti išvalgumo grupių narystės ir kintamųjų, naudojamų numatyti grupių narystei, santykiuose [6].

1.3. Regresija

Pasikartojančios regresijos tikslas yra numatyti atskirus kintamuosius nuo vieno ar daugiau nepriklausomų kintamųjų. Pasikartojanti regresija su daugybę numatomų kintamųjų yra tiesinės regresijos su dviem numatomais kintamaisiais ištesimas. Tiesinė transformacija X kintamųjų yra padaryta taip, kad stebėtų ir numatytų Y kvadratinių nuokrypių suma yra minimumas. Skaičiavimai yra sudėtingesni, tačiau, todėl tarpusavio ryšys tarp visų kintamųjų privalo būti apimtas į apskaičiavimus į kintamųjų pusiausvyros koeficientą. pasikartojančios regresijos analizės rezultatų interpretacija yra taip sudėtingesnė dėl beveik tos pačios priežasties. Y numatymas yra baigiamas tokia lygtimi:

$$Y_i = b_0 + b_1X_{1i} + b_2X_{2i} + \dots + b_kX_{ki}$$

"b" vertės yra vadinamos regresijos svertais (pusiausvyros koeficientais) ir yra apskaičiuojami tokio būdu, kuris minimizuoja kvadratinių nuokrypių sumą tuo pačiu būdu kaip paprasta tiesinė regresija. Šiuo atveju K numatytojo kintamieji yra greičiau du ir K +1 regresijos svertas turi būti apskaičiuotas, vienas kiekvienam K numatytojo kintamajam ir vienas konstantos (b₀) laikui [4].

1.4. Kuris metodas geriausias?

Iš esmės geriausio modelio iš vis nėra. Gerumas priklauso nuo problemos detalių: duomenų struktūros, naudojamų charakteristikų, laipsnio, iki kurio galima atskirti klases naudojantis tas charakteristikas ir klasifikacijos objektus. Klasifikacijos tikslumas, kad ir kaip bebūtų matuojamas, yra vienintelis veiklos aspektas. Kiti apima klasifikacijos greitį, greitį, su kuriuo klasifikacijos metodo vertinimas gali būti pakartotas ir lengvai suprantamas ir paaiškinamas, kodėl jis pasiekė savo pabaigą. Koks greitas klasifikacijos greitis, tuo patrauklesnis yra greitas sprendimas potencialiam paskolos ėmėjui, negu tu metu, kai jis turi augti kelias dienas. Greiti pasiūlymai gali smarkiai sumažinti išsekimo rodiklius. Klasifikacijos metodai, kurie yra lengvai suprantami (tokie kaip regresija ar artimiausio kaimyno ir medžio metodai) yra daug daugiau patrauklesni vartotojams, negu metodai kurie iš esmės yra juodosios dėžės (neuroniniai tinklai). Neuroniniai tinklai yra gerai suformuoti situacijoms, kuriose mes mažai turime žinių apie duomenų struktūrą. Tiesą sakant, neuroniniai tinklai gali būti laikomi sistemomis, kurios sukombinuoja automatizuota gavybą, su klasifikaciniais procesais, t.y. jie nusprendžia kaip kombinuoti ir transformuoti neapdorotas charakteristikas duomenyse, taip pat kaip ir lengvas sprendimo parametrų apskaičiavimas. Tai reiškia, kad tokie metodai gali būti naudojami skubiai, nesukeldami didelių problemų. Tačiau bendrai, jei mes turime puikų supratimą šie duomenis ir problemą, tada metodas, kuris gauna naudos iš to supratimo, gali geriau veikti [14].

2. NEURONINIŲ TINKLŲ TAIKYMO FINANSŲ VALDYMUI ANALIZĖ

Neuroniniai tinklai yra sudėtingas matematinis modelis parengtas pagal žmogaus smegenų funkcionavimo modelį. Tai naudingas ir galingas įrankis analizuojant didelius kiekius duomenų ir mokantis ieškoti šablonų duomenyse. Šie tinklai yra ypač gerai tinkantys plėtojant tikslias kreditų vertinimo sistemas, ypač kuomet palygini su kitomis klasifikacinėmis technikomis, tokiomis kaip loginė regresija, diskriminantų analizė, sprendimų medis ir taisyklės ar kitais egzotiškais matematiniais metodais. Prieš mokantis apie neuroninius tinklus, tyrėjas privalo atnaujinti procesus duomenų nustatymuose. Tik švarus duomenys yra naudingi: trūkstami duomenys, neatitinkantys, prieštaringi ir nesuderinami duomenys sumažina neuroninių tinklų kokybę. Po procesų atnaujinimo tyrėjas gali dirbti su tinklais naudodamas taip vadinamus mokomuosius algoritmus. Galiausiai, procesas gerai apskaičiuotas, naudojant tinkamus matus atspindės sistemos tikslumą. Jeigu tikslumas ne visai tinkamas arba tam tikra prasme netobulus, tuomet tai gali atvesti iki blogų rezultatų ir sprendimų. Kreditų vertinimo sistema bus laikoma nekompetentinga tokiu atveju ir tokiu būdu nenaudojama kaip naudingas įrankis finansiniam arsenale. Keli neuroninių tinklų architektūros šablonai yra prieinami SAS® Enterprise Miner™ programoje [18].

HNC kompanija, įkurta Roberto Hecht-Nielsen'o, išplėtojo keletą neuroninių tinklų pritaikymo būdų. Vienas iš jų yra Kredito Skaičiavimo sistema, kuri padidino egzistuojančio modelio pelningumą iki 27proc. Kredito rizikos nustatymas yra labai iššaukianti (reikalaujanti daugybės žinių) ir svarbi vadybos (mokslo) problema finansinių analizių srityje. Paliesti šiai problemai literatūroje buvo pasiūlyti daugybė klasifikacijos metodų. Ypač daug dėmesio sulaukė neuroniniai tinklai dėl jų turto aproksimacijos universalumo. Tačiau, darant sprendimus pagrindinis trūkumas susijęs su neuroninių tinklų naudojimu yra jų paaiškinimo stoka (mažos paaiškinimo galimybės). Kol jie gali pasiekti labai tikslų prognozuojantį rodiklį, priešastis, kaip jie gali padaryti sprendimus yra iš tiesų neprieinama. Aiškinant neuroninių tinklų sprendimus remiantis paaiškinamosiomis taisyklėmis, kurios užfiksuoja žinias įtvirtintas tinkluose, gali padėti kreditų rizikos valdymui, aiškinant, kodėl konkretus vartotojas yra klasifikuojamas kaip arba geras arba blogas. Režimuojant, galima pasakyti, kad neuroninių tinklų taisyklės gavyba ir sprendimų lentelės yra stiprus vadybos įrankis, kuris leidžia mums pastatyti pažangias ir patogias vartotojui sistemas, su sprendimų palaikymu kredito rizikos analizei [20].

2.1. NeuroXL klasifikatorius-neuroninių tinklų finansų pritaikymas-kreditai ir paskolos

Neuroniniai tinklai turėjo plačiai paplitusį priėmimą finansuose dalinai dėl jų puikių gebėjimų klasifikuoti duomenis. Tipiškiausias finansų pritaikymo pavyzdys yra paskolų vartojimo klasifikavimas, analizė kas nevykdys paskolų įsipareigojimo ir kodėl tai daro tokia didžiulę įtaką verslui. Bankai gali išvengti žymių praradimų išgydami bent minimalų supratimą apie pareigų nevykdančius kandidatus paskoloms. Koreliacijos kuri veda link blogų paskolų apibrėžimas gali būti efektyvus būdas mažinant riziką susijusią su paskolomis, išryškinant aukštos rizikos kandidatus kurių reiktų vengti, bet kokia kaina.

NeuroXL klasifikatoriaus programa yra pridėtinė prie Microsoft Excel skaičiavimo programos kuri gali būti naudojama gaunant aukšto tikslumo klasifikacijos duomenis apie paskolų kandidatus. Kuomet vertinimas reikalauja paskolos, daugybė susijusių kintamųjų įtakoja galutinį sprendimą ar individas vertas gauti kreditą ar ne. Neuroxl klasifikatorius gali lengviau daryti sprendimus lengvesniu būdu klasifikuodamas paskolos kandidatus, remdamasis daugybe kompleksinių kintamųjų. Pavyzdžiui kaip įeigos ir išeigos gali būti naudojami:

➤ **įeiga**

- Namų nuosavybės statusas
- Laiko tarpas kuomet nesikeičia padėtis
- Banko sąskaitos balansas
- Pareigos

➤ **išeiga**

- Klasifikuojama kaip aukštos rizikos, žemos rizikos, vidutinės rizikos
Viskas, ką reikia analitikui padaryti, tai patikslinti įeigos duomenis ir nustatyti kelis parametrus-neuroxl klasifikatorius tuomet padaro viską, klasifikuojant paskolos kandidatus.

Pažangi klasifikacijos technologija

Neuroniniai tinklai yra gerai įtvirtinta technologija sprendžiant klasifikacijos problemas finansuose, kuri naudoja praktikavimo ir testavimo duomenis kuriant modelius. Duomenys apima istorines informacijos rinkinius kurie sudaryti iš įeigos kintamųjų arba duomenų srities, kurie derinasi su išeiga. Mokomieji duomenys yra tai ką tinklai naudoja “mokydami” šablonus ir panašumus duomenyse.

Neuroxl klasifikatorius kaip finansų analizės įrankis

Neuroxl klasifikatorius tapo gana populiarus tarp finansų analitikų, nes įrodė jų klasifikavimo galią palyginimu su kitais pritaikymais ir technikos pagalba. Naudojantis

klasifikatoriumi, visa ko reikalaujama iš vartotojo tai yra patikslinti įėjigos duomenis ir nustatyti reikiamus parametrus.

Kainos nauda

Klasifikacijos programa tokia pat sudėtinga kaip neuroxl klasifikatorius, paprastai kainuoja šimtus kartais netgi tūkstančius dolerių. Neuroxl klasifikatorius yra pirmasis ir vienintelis sprendimas siūlantis savo pažangias technologijas mažiau nei už 50 doleriu.

2.2. Kreditų rizikos sprendimai paremti neuroniniais tinklais

Visi paskolų tiekėjai nori būti informuoti apie kandidatų galimybes išmokėti paskolas laiku, nes dažniausiai dėl šios informacijos jie yra mažiausiai užtikrinti. Finansinės institucijos klausia šito savęs pakankamai dažnai, kad pradėtų ieškoti tam sprendimo. Kreditų rizikos analizės metodų plėtojimas kuris valdytojams padeda apsispręsti ar suteikti paskolą yra labai sudėtinga problema. Pirmas sprendimas kuriant SAS programas kreditu rizikos nustatymui buvo sukurti, apmokyti ir įvertinti šiuos tinklus. Kūrėjai remiasi aukštu skaičiumi pažangių algoritminių apmokymų ir jų lengvai naudojamų grafinių sąsajų [14].

Sudėtingi sprendimai su didele įtaka

Kreditų rizikos analizė yra nelengvas darbas. Vienintelis neteisingas sprendimas gali sukelti dideles pasekmės susijusias su pinigų praradimu lygiai kaip ir su turtų pastatymu. Asmuo priimančis šiuos sprendimus turi suprasti ir paaiškinti priežastingumą savo veiksmų. Neverta sakyti jog bet kuris įrankis/įrenginys kuris gali teisingai įvertinti rizika ir supaprastinti sprendimo procesą turės didelę paklausa finansų srityje. Kiekviena finansinė institucija ieško patikimos ir lengvai naudojamos sprendimų paramos sistemos. Pažangūs, tačiau kartais sudėtingi statistiniai modeliai yra šiuo metu naudojami mažinant klaidingų sprendimų skaičių. Deja, šie modeliai retai yra patogūs ir gali būti naudojami tik patyrusių specialistų. Protinga kreditų vertinimo sistema turi būti ir tiksli ir suprantama [14].

2.3. Kreditų rizikos sprendimai paremti neuroniniais tinklais

Visi paskolų tiekėjai nori būti informuoti apie kandidatų galimybes išmokėti paskolas laiku, nes dažniausiai dėl šios informacijos jie yra mažiausiai užtikrinti. Finansinės institucijos klausia šito savęs pakankamai dažnai, kad pradėtų ieškoti tam sprendimo. Kreditų rizikos analizės metodų plėtojimas kuris valdytojams padeda apsispręsti ar suteikti paskolą yra labai sudėtinga problema. Pirmas sprendimas kuriant SAS programas kreditu rizikos nustatymui buvo sukurti, apmokyti ir įvertinti šiuos tinklus. Kūrėjai remiasi aukštu skaičiumi pažangių algoritminių apmokymų ir jų lengvai naudojamų grafinių sąsajų [14].

Sudėtingi sprendimai su didele įtaka

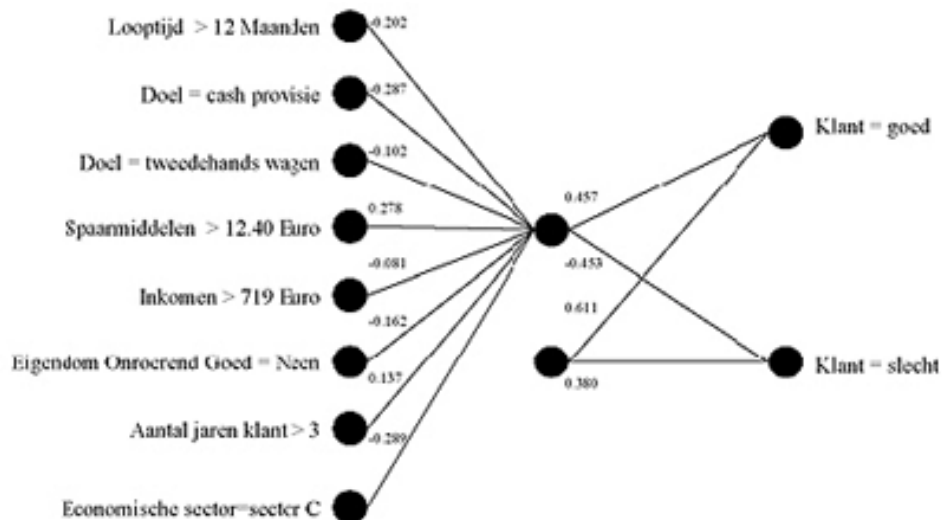
Kreditų rizikos analizė yra nelengvas darbas. Vienintelis neteisingas sprendimas gali sukelti dideles pasekmės susijusias su pinigų praradimu lygiai kaip ir su turtų pastatymu. Asmuo priimančias šiuos sprendimus turi suprasti ir paaiškinti priežastingumą savo veiksmų. Neverta sakyti jog bet kuris įrankis/įrenginys kuris gali teisingai įvertinti riziką ir supaprastinti sprendimo procesą turės didelę paklausa finansų srityje. Kiekviena finansinė institucija ieško patikimos ir lengvai naudojamos sprendimų paramos sistemos. Pažangūs, tačiau kartais sudėtingi statistiniai modeliai yra šiuo metu naudojami mažinant klaidingų sprendimų skaičių. Deja, šie modeliai retai yra patogūs ir gali būti naudojami tik patyrusių specialistų. Protinga kreditų vertinimo sistema turi būti ir tiksli ir suprantama [14].

SAS programa galingų neuroninių tinklų kūrimui

SAS® Enterprise Miner™ ir STAT® teikia pažangią paramą plėtojant aukštos kokybės kreditų vertinimo sistemas. SAS programa apima daugybę pažangių mokomųjų algoritmų neuroniniams tinklams. Kompanijoje SAS Enterprise Miner informacija dažnai yra vizualizuojama diagramomis, kurios padaro programą patrauklią ir lengvai naudojamą. Progresyvi tikslumo matų technika taip pat yra pateikiama. SAS vartotojai renkasi būtent dėl patogios grafikos, patogaus naudojimo ir galingo bei sudėtingo algoritmo supaprastinimo [14].

Rezultatas: suprantamos ir naudingos taisyklės

Galutinis šio tyrimo tikslas yra suprantamų taisyklių rinkinys vietoj sudėtingo matematinio modelio. Ši forma yra kilusi iš neuroninių tinklų modelio naudojantis gavybos taisyklę arba trimis gavybos technikomis. Glausta ir aiški sprendimų lentelė palengvina ir pagreitinga teisingų sprendimų priėmimą. Tai taip pat įrankis kreditų valdyje aiškinant klientui sprendimo priėmimą [14].



1. Savings Account	<= 12.40 Euro										> 12.40 Euro					
2. Economical sector	Sector C		other								-					
3. Purpose	-		cash provisioning				second-hand car				other		-			
4. Term	-		<= 12 months				> 12 months				-		-			
5. Years client	-		<= 3		> 3		<= 3		> 3		<= 3		> 3		-	
6. Owns Property	-		Yes		No		-		-		Yes		No		-	
7. Income	-		<= 719 Euro		> 719 Euro		-		-		<= 719 Euro		> 719 Euro		-	
1. applicant = good	-	x	x	-	x	-	x	-	x	x	-	x	x	x		
2. applicant = bad	x	-	-	x	-	x	-	x	-	-	x	-	-	-		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14		

1 pav. Glausta ir aiški sprendimų lentelė palengvina ir pagreitina teisingu sprendimu priėmimą.

Kreditų vertinimo rezultatų analizė: pagrindiniai principai papildomiems sprendimams

Kreditų suteikimas tai ne tik sprendimas suteikti paskola ar nesuteikti. Pakankamai svarbu žinoti ir tai kuriuo momentu klientas gali nebeturėti galimybių gražinti paskolos. Tai vadinama rezultatų analize ir ji gyvybiškai svarbi papildomiems faktoriams primant galutinį sprendimą. Ši analizė palengvina nustatant tokius parametrus kaip paskolos trukmė ir maksimalus kredito limitas. Tai realizuoja paskolos tiekimą ir paskolos gražinimo elgesio valdymą.

SAS ® priklauso pačiai galingiausiai prieinamai programai, kuri naudojama rezultatų analizei. Tačiau dėl to, jog neuroniniai tinklai nėra normaliai pritaikyti rezultatų analizei, yra dar toliau atliekami tyrimai prieš publikuojant šią programą šioje srityje [10].

2.4. Kreditų rizikos analizė naudojantis ANN

Kreditų rizikos analizės tokios kaip įsipareigojimų vertinimas ir bankrotų numatymas paprastai naudodavo regresijos analizę, MD Air loginę regresiją. Tokie būdai tapo standartiniu etalonu, kuriuo ANN priemonės buvo lyginamos. Iš darbų apžvelgtų žemiau aišku, kad ANN vykdymas kreditų rizikos klasifikavime yra nenusileidžiantis šioms priemonėms. Leshno ir Spector (1996) parodė, kad šios priemonės atstovauja specialius ANN atvejus ir todėl nenuostabu, jog ANN tapo žymiai svarbiau kreditų rizikos srityje.

Dutta and Shekhar (1988) pirmieji išrado ANN modelį įsipareigojimų reitingų apjungimui. Jie palygino ANN modelio veiklą su regresijos modeliu ir atrado, jog ANN nuosekliai veikia geriau įsipareigojimų vertinimo nustatyme iš duotų finansinių rodiklių. Visumoje regresijos modelio klaidų buvo žymiai daugiau ir reikšmingesnių nei ANN modelio. Numatymo rodiklio tikslumas ir teisingumas ANN modelio buvo 88.3% tuo tarpu kai tas pats rodiklis regresijos modelio buvo 64.7%. Šio reitingo nustatyme modelių objektu buvo atpažinimas, ar duoti įsipareigojimai priklausę konkrečiai reitingų grupei, pavadintai AA [11].

Tokie reikšmingi rezultatai buvo motyvuoti tolesniame. ANN įsipareigojimų vertinimo realizavime. Įsipareigojimų vertinimo apibendrinimo kontekste Surkan ir Singleton (1990) palygino ANN prieš MDA ir priėjo išvados, jog ANN atliekamas darbas žymiai geresnės kokybės. Jų įgyvendinime jie apribojo išieigos vektorius iki dviejų elementų kur AAA-A3 įsipareigojimas, ant Mood'žio vertinimo skalės buvo suklasifikuoti į tris grupavimus. Išplėtimas vertinimo klasių įskaitant ir subklasifikacijų vedė į neuroniniu tinklu nekonvergenciją. Taip galėjo būti dėl to, jog papildymo dydis buvo neadekvatus atspindėti žemesnias klases.

Jie taip pat apklause ar neuroniniai tinklai galėjo klasifikuoti tiksliai tarp gretutinių žemesnių klasių. Tai veda prie Singleton and Surkan (1995) tyrimo kuris nustatė įsipareigojimų reitingo kritimą arba pakilimą naujos finansinės informacijos šviesoje.

Kim, Weistroffer ir Redmond (1993) palygino veikimą regresinės analizės, MDA, kelintines logistinės regresijos ir taisykle paremtus metodus su ANN įsipareigojimų vertinimo klasifikavime. Teisingai suklasifikuotų įsipareigojimų procentingumas buvo toks 36.2%, 36.2%, 43.1%, 31.0%, ir 55.2% nurodyta tvarka, kur vertinimo klasės buvo suskirstytos į 6 kategorijas. Maher and Sen (1997) taip pat palygino ANN ir loginę regresijos veikimą. Jų įgyvendinimas suklasifikavo įsipareigojimus į šešių kategorijų reitingus pradedant nuo AAA iki B Mood'žio vertinimo skalėje. Sėkmingiausias regresijos modelis panaudojo tą patį nepriklausomų kintamųjų rinkinį kaip ir Kaplan'o ir Urwitz'o (1979) modelis. Kaplan'as ir Urwitz'as teisingai suklasifikavo 65.5% tuo tarpu kai Maher'o ir Sen'o (1997) regresijos modelis 61.66%. tačiau, jų ANN įgyvendinimas pasiekė 66.66% naudodamas tuos pačius įieigos kintamuosius. Daniel'iaus and

Kamp'o (1999) ANN realizavimas įsipareigojimų vertinime aiškiai atliko geriau linijinį modelį paremtą teisingų klasifikacijų procentingumu (67% prieš 53%). Jie priartėjo prie pilno skaičiaus klasių tik iki 7 ir panaudojo viena išeią, užsibaigiančią artimiausia diskrečia verte [11].

Visiškai aišku, kad ANN parode puikias perspektyvas įsipareigojimų vertinimo srityje. Tačiau anksčiau paminėtos studijos supaprastino klasifikacijos problemą varžydami klasifikacijų kategorijų skaičių atpažintų modelio ir dirbtinai kurdami duomenų rinkinius kuriuos sudaro vienodai paskirstytas stebėjimas per visas šias metaklases.

Kwon, Han ir Lee (1997) nustatė šį bruožą, nors jų ANN modelis klasifikavo tik į 5 skirtingas klases. Jų tyrime, kelintinių nepriklausomų kintamųjų rūšis buvo formaliai adresuota per kelintinę porinę dalijimo prieigą.

Dėl šio realizavimo prieigos, keturi ANN buvo panaudoti serijoje kurioje kiekvienas tinklas arba klasifikavo įsipareigojimus į klasės dalis kurias jis reprezentavo arba nukreiptas į sekančią tinklo klasifikaciją. Kiekvienas tinklas buvo apmokytas individualiai.

Eksperimento rezultatai paremia tokių apmokymų prieigos naudojimą per standartini ANN ir kartu neuroninių tinklų prieigą puikiai atliekama MDA.

Mood'is and Utans'a (1995) ištyrė pilnos 16 klasifikacinių kategorijų, palygintų su 5 ar 3, asortimento efektą.

Nors procentas teisingų klasifikacijų buvo labai sumažintas, procentas vienu skaičiumi viršijo teisingų penkių kategorijų atvejo procentingumą [11].

Tai paremia Singleton'o and Surkan'o (1995) išvadas, jog ANN turi sunkumų nustatant skirtumą tarp gretimų kategorijų. Tačiau naudojimas visų subklasių nesumažina klasifikacijų kokybės laipsnio nuo tada kai tikslus metaklasių klasifikavimas yra panašus į vieno laipsnio klasifikavimą. Panašiam kontekste, mokslininkai tirti lemiami bankrotų veiksniai ir paskolų įsipareigojimų nevykdymai taip pat turėjo didelį pasisekimą su neuroninių tinklų įgyvendinimu. Šiais atvejais priklausomi kintamieji yra paprasčiausiai lygūs nuliui arba vienam. Tam and Kiang (1992) padarė išvadą, jog ANN geriau veikė negu linijinių diskriminantų analizė ir logistinė regresija jų tyrimuose bankų bankrotų numatyme. Kiti žinomi modeliai taip pat buvo ištirti ir veikė paliginti prasčiau. Jų rezultatai aiškiai parodė, kad finansinės metų ataskaitos apėmė vertingesnę informaciją. Jie palygino jų rezultatus su MDA, konkrečiame Z-rezultatų Altman'o (1968) modelyje, ir suprato jog ANN nepavyzdinis nustatymas buvo neįtikėtinai tikslus.

Enache and Banne (1997) ištyrė kreditų riziką vartotojų paskoloms, lygindami logistinių diskriminantų analizę, klasifikacinių medžių ir ANN veikimą.

Tyrimas parode jog ANN veikimas pagerino veikla tarp prieštaringų ir išskirtinių tvirtinimų tuo tarpu kiti metodai teikė prastesnius rezultatus. ANN ryškiai geriau veikia negu kiti metodai klasifikuojant didžiulius skaičius pajamų išmokėjimų kurie nekelia problemų, bet ne taip

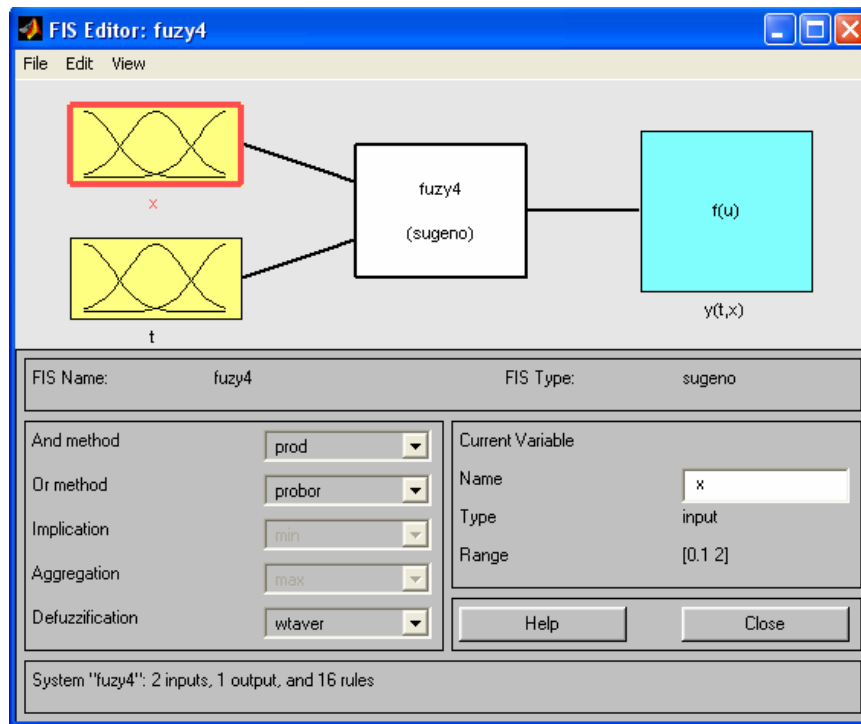
gerai sekasi numatant problematiškus paskolų teikimus. Vis dėlto ANN veikia netgi geriau nei neuro-fuzzy sistema visais galimais atvejais [11].

2.5. Fuzzy logikos naudojimas neuroninio tinklo apmokymui

Nuo tada, kai buvo paskelbtos neryškios (fuzzy) logikos teorijos ir metodologijos idėjos (Zadeh, 1965), fuzzy modeliavimas bei valdymas yra karštų diskusijų tarp modeliavimo ekspertų objektas. Palaiapsniui pripažįstama, kad neryškių aibių metodika yra gana patraukli, sprendžiant sudėtingų sistemų modeliavimo ir valdymo uždavinius. Neryškios logikos strategiją ypač tikslinga taikyti, kai sprendžiami netiesinių sistemų, kurioms būdinga neapibrėžta ar/ir nepatikima informacija apie sistemos elgseną, modeliavimo uždaviniai. Neryškios logikos modeliai yra naudojami apytikslėms sistemos reakcijoms į besikeičiančius įėjimus atvaizduoti. Dažnai fuzzy modeliavimas traktuojamas kaip duomenų “surišimas” tarp dviejų, paprastai skirtingų dimensijų įėjimo ir išėjimo vektorių. Toks duomenų “surišimas” dažnai aprašomas lingvistinėmis taisyklėmis. Jei šios taisyklės remiasi neryškiomis aibėmis ir neryškios logikos metodais, tai šį logikos taisyklių rinkinį galima laikyti proceso neryškios logikos modeliu. Kadangi, fuzzy taisyklėse dažniausiai atsispindi euristicinė patirtis, kurią inžinieriai ar vadybininkai sukaupia, nagrinėdami procesų įėjimo/išėjimo duomenis, tai tokie modeliai dažnai vadinami “pilkos dėžės” (gray - box) modeliais.

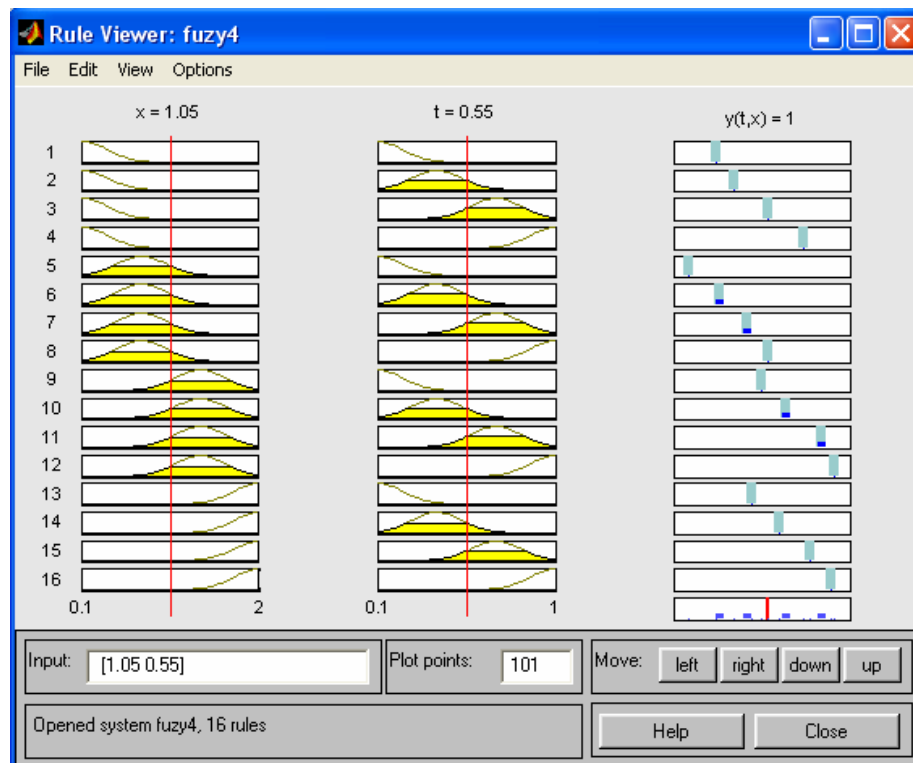
Neryškių aibių pagrindu suformuotų “pilkos dėžės” modelių privalumas prieš regresinius modelius yra tas, kad čia galima išnaudoti išankstines žinias apie proceso įėjimų ir išėjimų priklausomybes.

Žemiau pateiktame pavyzdyje parodytas neuroninio tinklo apsimokymas. Kol kas šis pavyzdys taip pat veikia tik su savo sugeneruotais duomenimis, o mano pateiktų duomenų jis nesupranta ir paklaidą ne mažina, o didina. Prognozavimas bus atliekamas naudojant fuzzy modeliavimą.



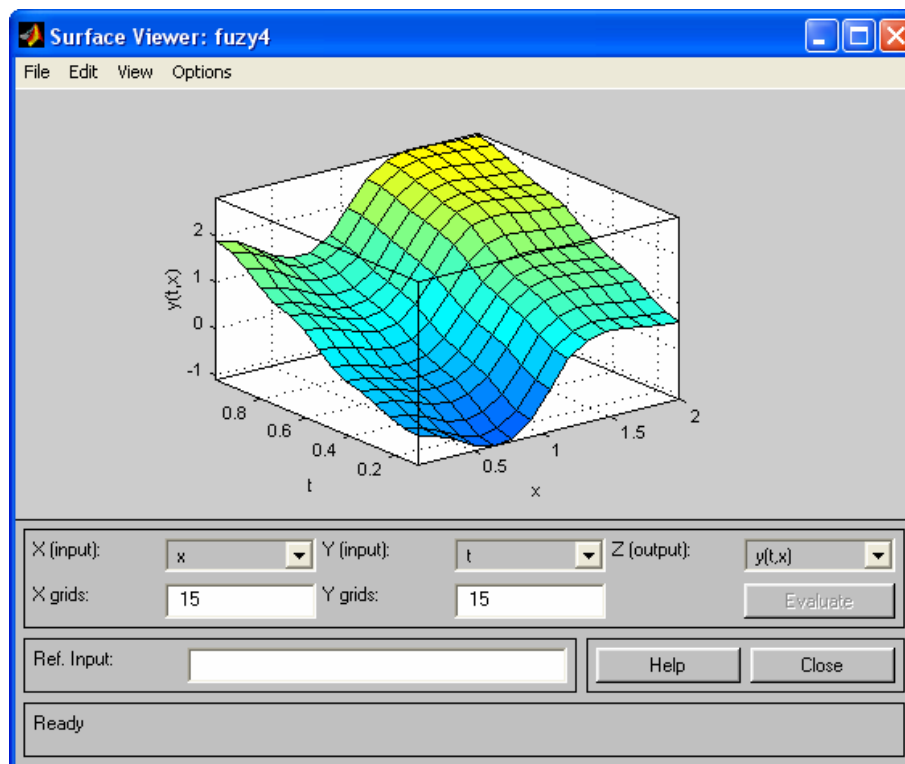
2pav. fuzy4 įėjimų nustatymo langas

Pirmiausia nustatoma kiek bus įėjimų. Pavyzdžiui parinkau du įėjimus x (finansiniai duomenys) ir t (verslo duomenys). Toliau nurodomos tam tikros taisyklės kuriomis vadovaujantis bus apmokomas neuroninis tinklas. Tarkim taisyklės sukuriamos tokios pagal kurias būtų galima prognozuoti, įmonės imančios kreditą, reitingą.



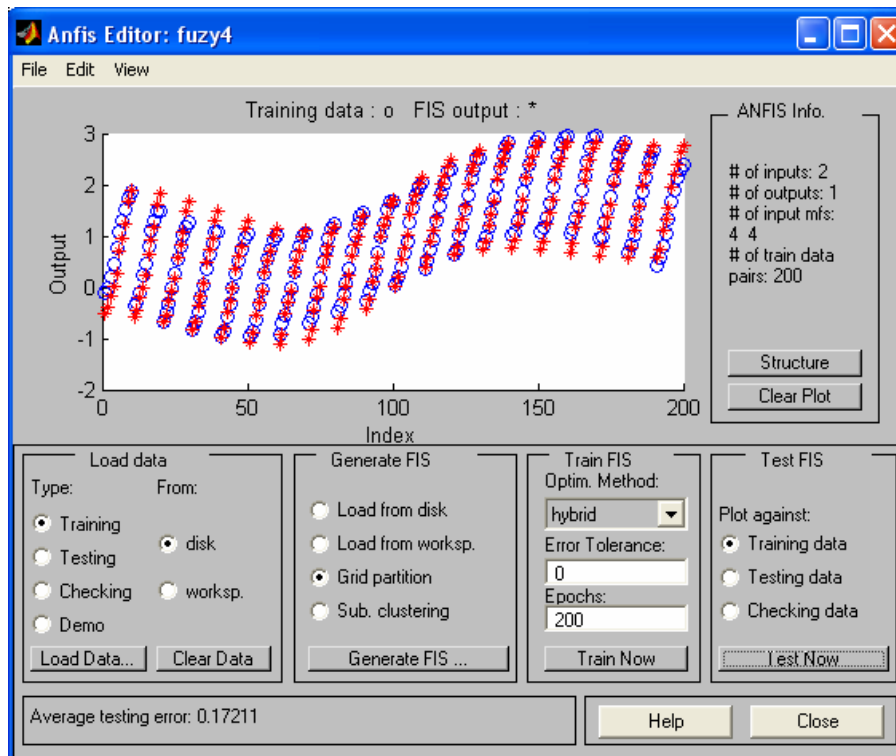
3pav. fuzy4 taisyklių peržiūrėjimo ir koregavimo langas

Pagal šias taisykles jau yra suformuotas neapmokyto neuroninio tinklo vaizdas. Jį dar rankiniu būdu pabandžiau privesti prie tikro vaizdo.

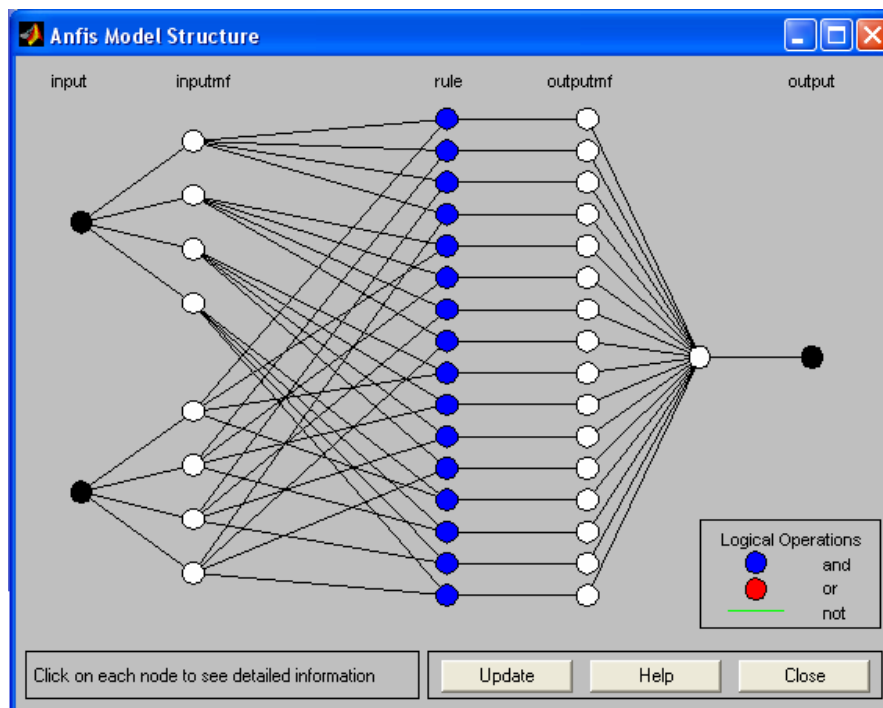


4pav. fuzy4 paviršiaus suformuoto rankiniu būdu langas

Matosi, kad rankiniu būdu apmokytas neuroninis tinklas turi nemažą paklaidą. Mūsų neuroninis tinklas bus apmokomas 200 kartų.

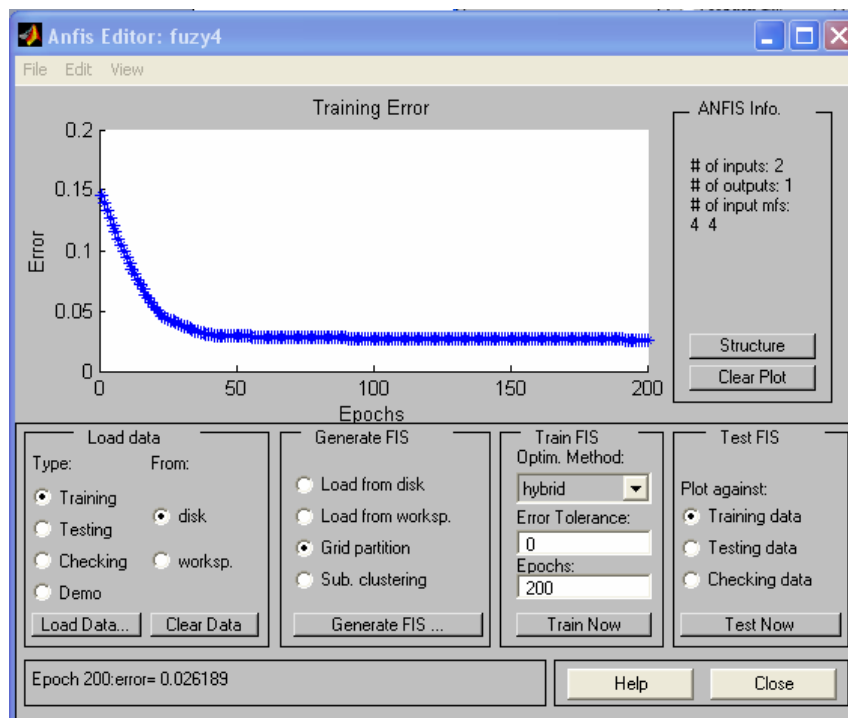


5 pav. fuzy4 apmokymo, testavimo langas

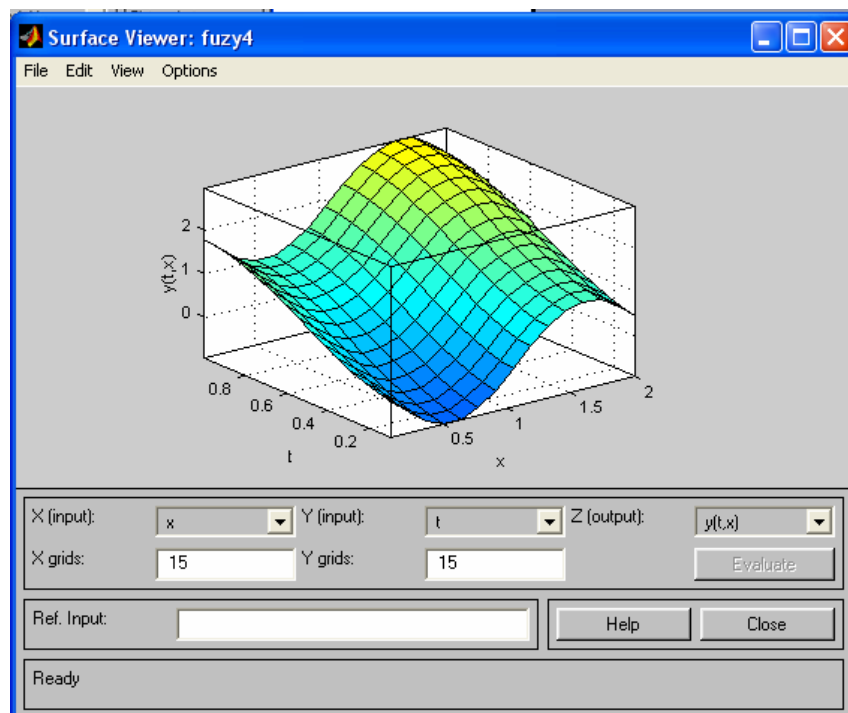


6 pav. suformuoto modelio struktūra

Apmokome neuroninį tinklą. Prieš tinklo apmokymą matome kad rankiniu būdu paklaida pavyko sumažinti iki 1,5%. Per 60 iteracijų paklaida sumažėjo beveik iki 0,26% per likusias iteracija paklaida mažėjo labai nežymiai.



7pav. fuzy4 apmokymų langas



8pav. fuzy4 treniruotas paviršius

3. BENDRIEJI TINKLO APMOKYMO ALGORITMAI

3.1. Atgalinio sklaidimo neuroninis tinklas

Atgalinio sklaidimo modelį lengva suprasti ir jis lengvai gali būti naudojamas programinių poreikių modeliavimui.

Apie NN sritį galima galvoti kaip susijusią su dirbtiniu intelektu, mašinų (mechanizmų) nagrinėjimu, paralelinių apdorojimų, statistika ir kitomis sritimis. NN patrauklumas yra tas, kad jie labiausiai tinka problemų, kurias sunku išspręsti tradiciniais skaičiavimo metodais, sprendimui.

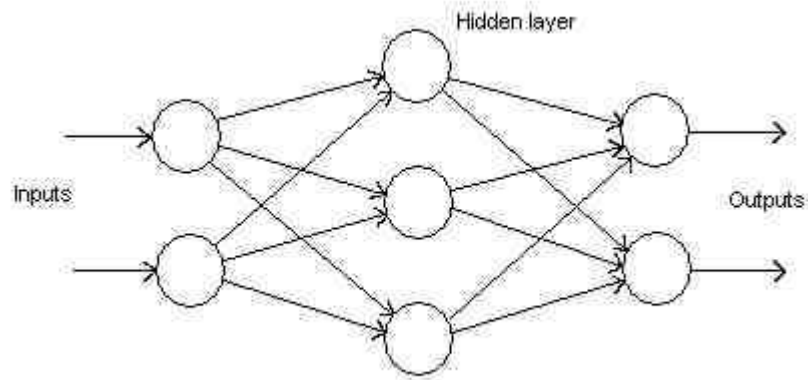
Įsivaizduokime vaizdo apdorojimo uždavinį, tokį kaip kasdienio objekto, projektuojamo kitų objektų fone, atpažinimą. Tai - užduotis, kurią net mažo vaiko smegenys gali išspręsti per kelias dešimtas sekundes. Bet sukurti taip pat gerai veikiančią įprastą nuoseklią (serijinę) mašią (mechanizmą) yra nepaprastai sudėtinga. Tačiau tas pats vaikas gali nesugebėti suskaičiuoti $2+2=4$, kai tuo tarpu nuosekli mašina išsprendžia tai per kelias nanosekundes.

Esminis skirtumas tarp vaizdo pažinimo problemos ir sudėties problemos yra tas, kad pirmoji geriau išsprendžiama paraleliniu būdu, o paprasta matematinė - nuosekliu būdu. Neurologai tiki, kad smegenys yra kaip plataus masto paralelinis analoginis kompiuteris, turintis 10^{10} paprastų procesorių, kurių kiekvienam reikia kelių milisekundžių atsakyti į įvestą informaciją. Su NN technologija, mes galime naudoti paralelinio apdorojimo metodus realaus pasaulio problemų sprendimui, nes jame labai sunku taikyti nuoseklų algoritmą.

Jeigu mes suvokiame žmogaus smegenis kaip "didžiulį" NN, tai geriausia būtų, kad norėtume sukurti įrenginį, imituojantį žmogaus smegenų funkcijas. Tačiau, dėl technologijos ribotumo, mes turime apsiriboti daug paprastesniu modeliu. Akivaizdus požiūris būtų sukurti mažą elektroninį pretaisą, turintį perdavimo funkciją, panašią į biologinį nervą, o po to sujungti kiekvieną nervą su daugybe kitų nervų, naudojant RLC tinklus imituoti dendritus, aksonus ir sinapses. Šis elektroninio modelio tipas vis dar gana sudėtingas įgyvendinti, ir kils sunkumų "mokant" tinklą daryti ką nors naudinga. Tolimesni suvaržymai yra reikalingi padaryti projektą labiau valdomą. Pirmiausia, reikia keisti nervų jungtis/sujungimus taip, kad jie būtų skirtinguose sluoksniuose, ir, kad kiekvienas nervas vename sluoksnyje būtų sujungtas su visais nervais (su kiekvienu) kitame sluoksnyje. Po to, nustatoma, kad signalų srautas tekėtų tinkle tik viena kryptimi. Supaprastinamas nervo ir sinapsės modelis, kad veiktų kaip analogo komparatoriai (nešami kitų nervų) per paprastus rezistorius (varžas).

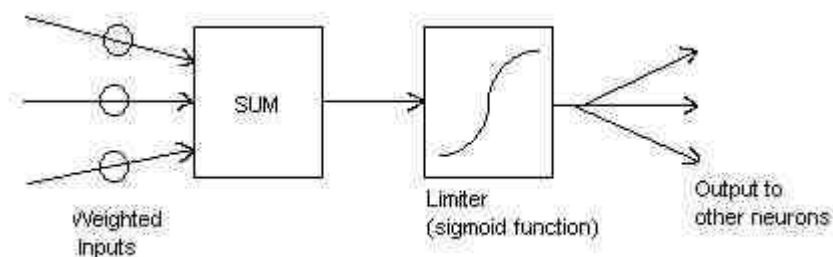
Devintame ir dešimtame paveiksluose, tinklas veikia taip: kiekvienas nervas gauna signalą iš nervų, esančių ankstesniame sluoksnyje, ir kiekvienas signalas yra padauginamas atskiro svorio verte. Pasverti duomenys yra susumuojami ir perduodami per ribojimo funkciją, kuri matuoja

išeinančius duomenis nustatytų verčių diapazonu. Išėjė duomenys yra perduodami visiems nervams kitame sluoksnyje. Taigi, kad naudoti tinklą spręsti problemą, pritaikomos įvadinių duomenų vertės prie pirmojo sluoksnio įvadinių duomenų, leidžiama signalams sklirti tinklu, ir skaitomi išeinantys duomenis.



9pav. Apibendrintas generalizuotas tinklas

9 paveikslas. Apibendrintas (generalizuotas) tinklas. Pirmojo suoksnio įvadiniams duomenims yra taikomas stimuliavimas ir signalai sklinda viduriniiais (paslėptais) sluoksniais į produkcijos (galutinių duomenų) sluoksnį. Kiekvienas ryšys tarp neuronų turi unikalią vienintelę apkrovimo vertę.



10pav. Neurono struktūra

10 paveikslas. Neurono struktūra. Įvadiniai duomenys iš vieno ar daugiau ankstesnių neuronų yra atskirai sveriami, po to susumuojami. Resultatas yra išmatuojamas nelinejiniai tarp 0 ir +1, ir produkcijos vertė perduodama kito sluoksnio neuronams.

Kadangi realus tinklo unikalumas arba "intelektas" egzistuoja svorio tarp neuronų vertėse, mums reikia svorio derinimo metodo, kad išspręsti atitinkamą problemą. Šiam tinklo tipui įprasčiausias mokymosi algoritmas yra vadinamas Atgaliniu Sklidimu (BP). BP mokosi iš

pavyzdžių, t.y., turi būti pateiktas mokymosi rinkinys, susidedantis iš įvadinių duomenų pavyzdžių ir žinomų-teisingų galutinių duomenų kiekvienam atvejui. Taigi, naudojamas įvadinių-galutinių duomenų pavyzdys parodytu tinklui kokio veikimo yra tikimasi, o BP algoritmas leidžia tinklui prisitaikyti.

BP mokymosi procesas veikia mažais pasikartojančiais žingsneliais: vienas iš pavyzdžių atvejų įvedamas į tinklą ir tinklas pateikia galutinius duomenis, paremtus esama jo sinapsiniu svorių sandara (iš pradžių, duomenys bus atsitiktiniai). Šie duomenys yra lyginami su žinomais-gerais duomenimis ir vidutiniu paklaidos signalu yra apskaičiuojami. Paklaidos vertė yra perduodama atgal per tinklą ir yra padaromos mažos svorio pataisos kiekviename sluoksnyje. Svorio pataisos yra apskaičiuojamos sumažinti klaidos signalą to atvejo klausime. Visas procesas yra kartojamas kiekvienam pavyzdžio atvejui, tada atgal prie pirmojo pavyzdžio ir t.t. Ciklas yra kartojamas kol klaidos vertė visiškai sumažinama žemiau iš anksto apibrėžto lygio. Šiame taške sakoma, kad tinklas išmoko problemą "pakankamai gerai" - tinklas niekada tiksliai neišmoks idealios funkcijos.

4. BENDRŲJŲ TINKLO APMOKYMO ALGORITMŲ EKSPERIMENTINIS TYRIMAS

4.1. batbp funkcija

Aprašymas

[skaitant atitinkamą rinkinį įėjimo porų – išėjimo ir pradinio tinklo [W1, W2, critvec, praėjimas] = batbp (NetDef, W1, W2, PHI, Y, trparams) apsimoko tinklas su apdirbtais pasiskirščiusiais pasklidimais. Aktivacijos funkcijos turi būti linijinės arba hiperbolinė. Tinklo architektūra apibūdina matricą. „NetDef“ susidedančio iš dviejų eilių. Pirmoji eilė apibūdina paslėptą sluoksnį, tuometu kai antra eilė apibūdina išėjimo sluoksnį.

Pavyzdžiui: NetDef = ['LHHHH'LL---']

(L = Linijinis, H = hiperbolinė)

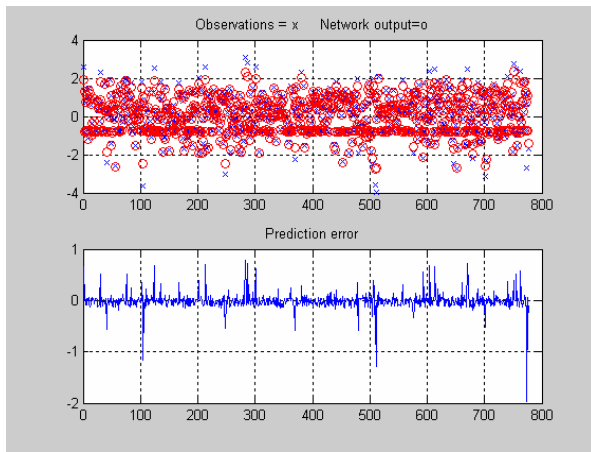
Algoritmas

Atgalinis sklidimas – gradiento paleidimo algoritmas, kur nuskaitymas vyksta paprastu būdu, naudojant specialią architektūrą savo interesais. Šitame atlikime yra nustatytas žingsnio dydis [27].

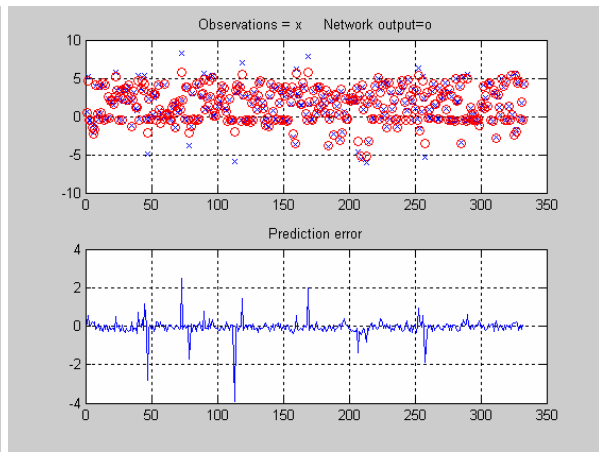
Naudojant **BATBP** funkciją buvo atlikta keletas neuroninio tinklo apmokymų ir gautos paklaidos. Programinis kodas 5 priede.

Lentelė Nr.1

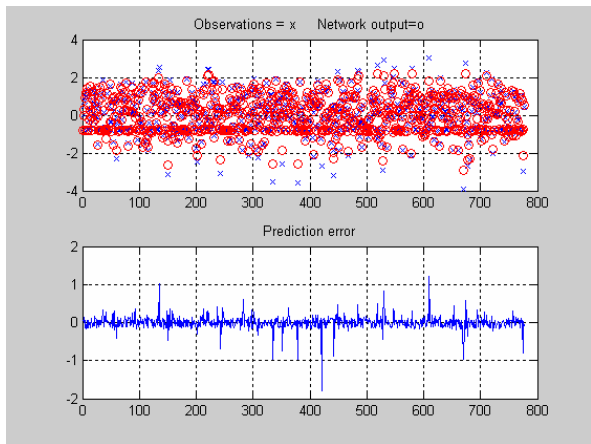
Iteracijų skaičius	Tinklo parametrai	NSSE	NSSE2
30	'HHLHHLHHHH';L-----'	0,0138	0,0872
50	'HHLHHLHH';L-----'	0,0148	0,0431
80	'HLHLLLH';L-----'	0,0030	0,0023
30	'HLHLLH';L-----'	0,0762	0,4477
100	'HLLLLH';L-----'	0,0047	0,0138
VIDURKIS		0,0225	0,5941



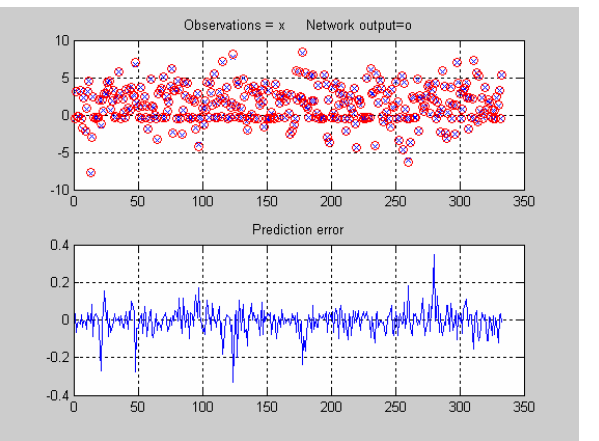
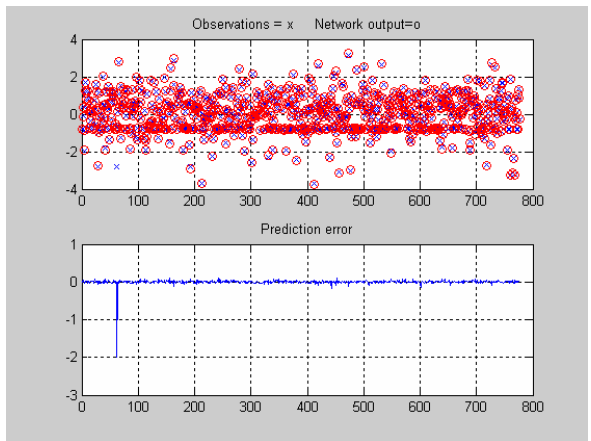
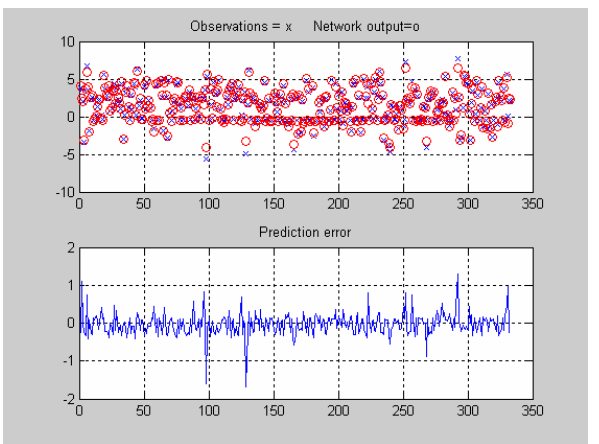
10-11 pav. Neuroninio tinklo apsimokymas

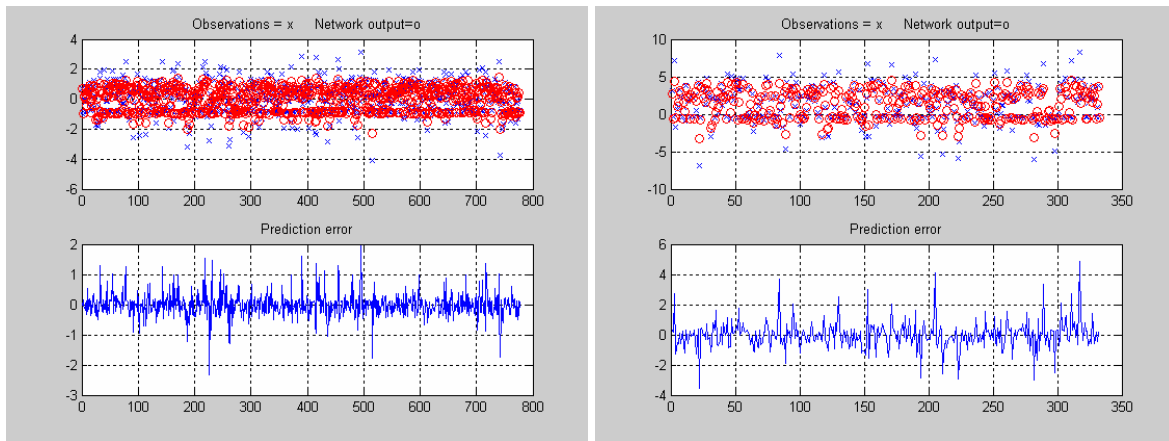


12-13 pav. Neuroninio tinklo apsimokymas

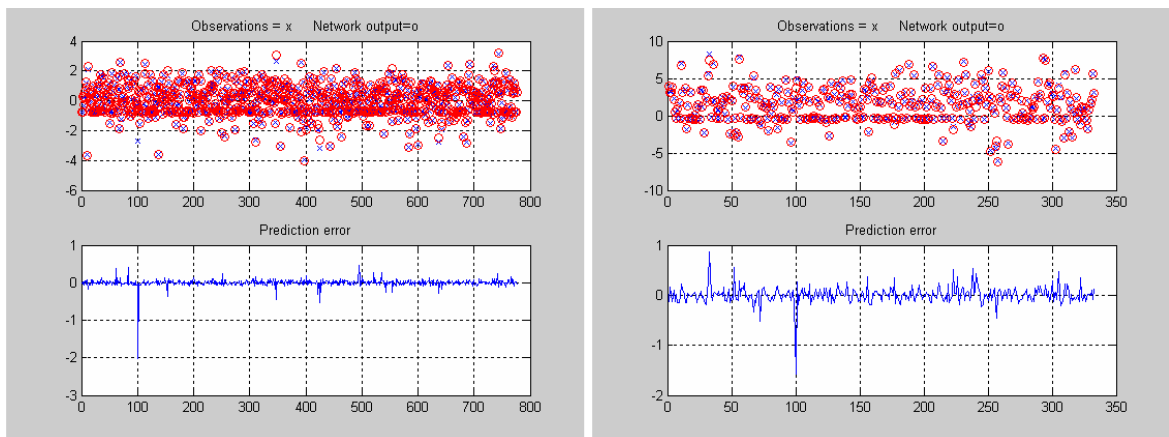


14-15 pav. Neuroninio tinklo apsimokymas





16-17pav. Neuroninio tinklo apsimokymas



18-19pav. Neuroninio tinklo apsimokymas

4.2. igls funkcija

Tikslas

Pakartotinis apmokymas nemažų kvadratų neuroninių tinklų su daugkartiniais įėjimais.

Aprašymas

$[w1, w2, liambda, gama] = igls(NetDef, W1, W2, trparms, Gamma0, PHI, Y)$

Įėjimas

NetDef, W1, W2, trparms, PHI, Y: žr. funkcija MARQ.

Gama 0 pirminis matricos kovariacijos skaičius dėl triukšmo. Jeigu perduota kaip [] tai nustatyta matricos identifikavime.

Trparms: vektorius, kurio parametrai susiję su apmokymu (žr. MARQ funkcija). Nemokėjimo vertybės (gautas jeigu trparms = []): trparms = [50 0 10]

Produkcija {išėjimas}

w1, w2, liambda: žiūrėti funkciją MARQ.

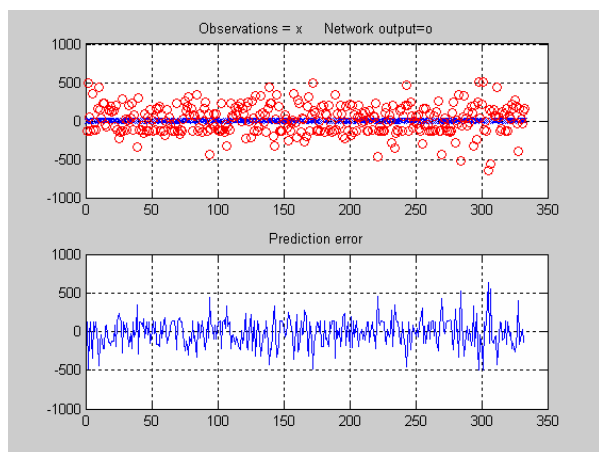
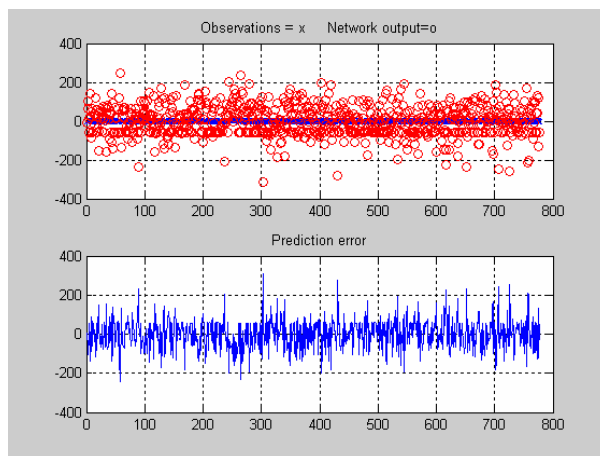
Multiprodukcija (Multiišėjimas) besimokantis tinklas ir aptriukšmintą kovariacijos matrica, įvertinta su pasikartojančia atsipalaidavimo procedūra. Būtina padauginti išėjimus iš SQRTM kad

gauti neišskaičiuojamus nusakymus. Jeigu tinklas turi vienintelį linijinį išėjimą, galima jo vietoje išmatuoti paslėptų sluoksnių svorį: $W2 = \sqrt{\Gammaamma} * W2$ [27].

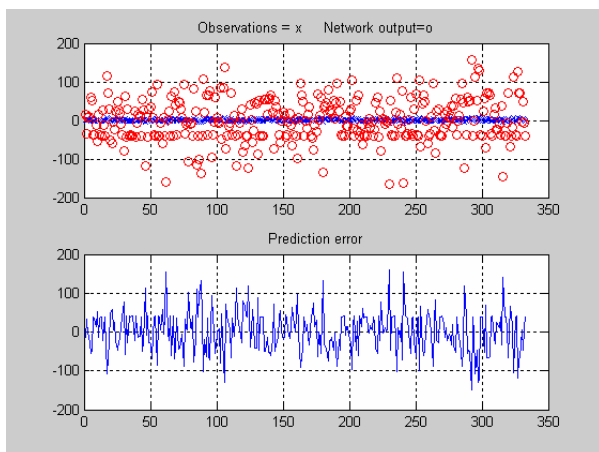
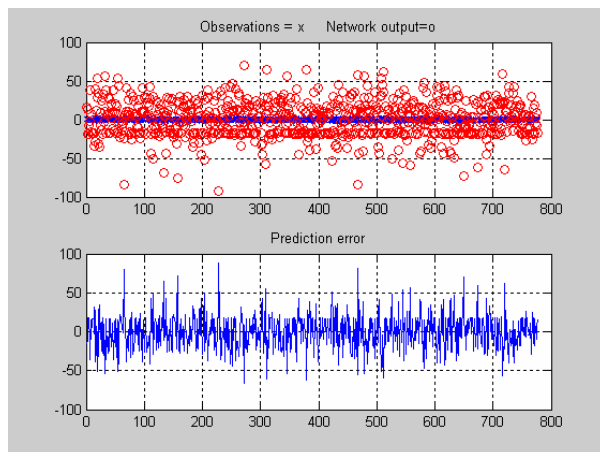
Naudojant **IGLS** funkciją buvo atlikta keletas neuroninio tinklo apmokymų ir gautos paklaidos. Programinis kodas 4 priede.

Lentelė Nr.2

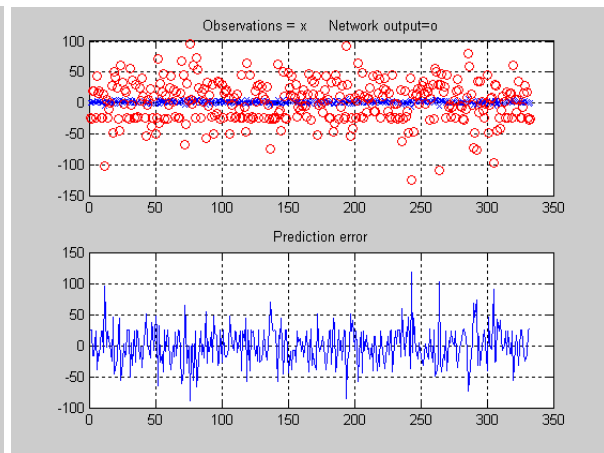
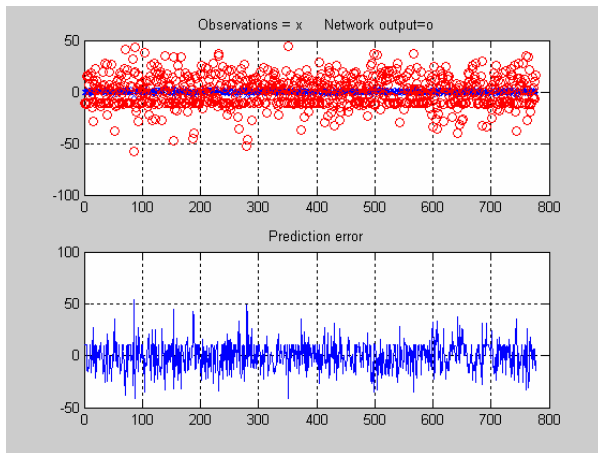
Iteracijų skaičius	Tinklo parametrai	NSSE	NSSE2
30	'HHLHHLHHHH';'L-----'	25,5238	7,3439
50	'HHLHHLHH';'L-----'	242,1242	2,5741
80	'HLHLLLH';'L-----'	94,2994	468,1521
30	'HLHLLH';'L-----'	92,2063	524,2173
100	'HLLLLH';'L-----'	1025,9170	196,9687
VIDURKIS		659,3371	1041,68114



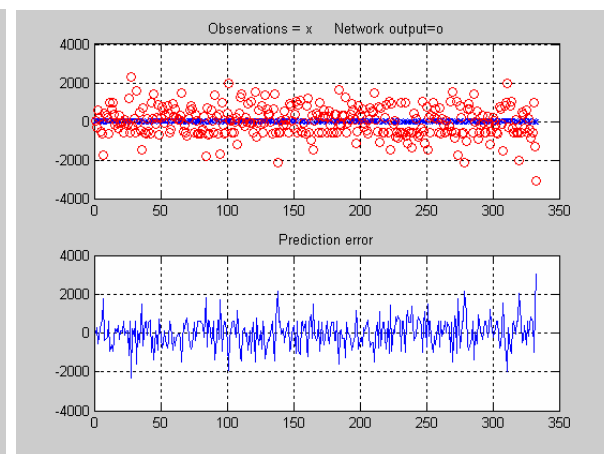
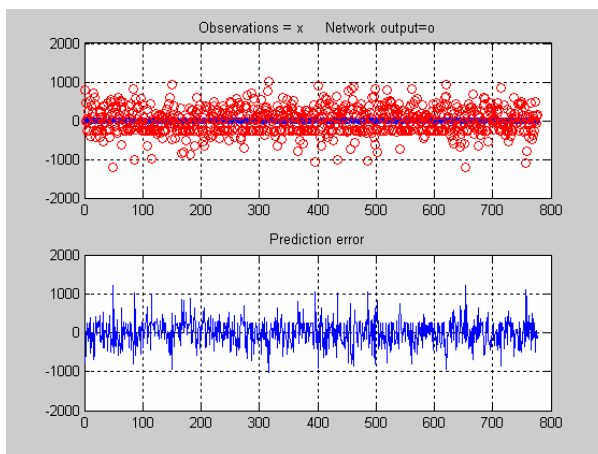
20-21pav. Neuroninio tinklo apsimokymas



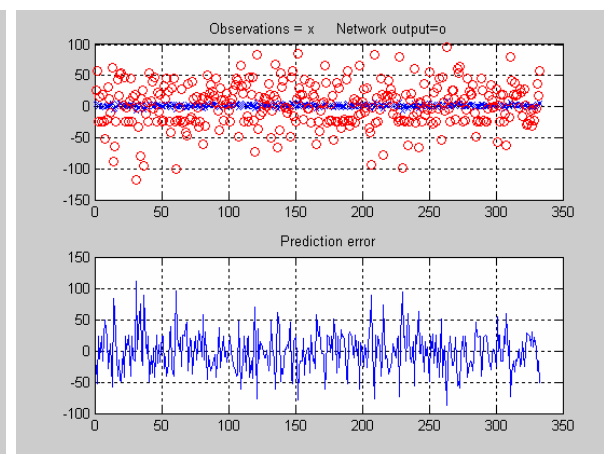
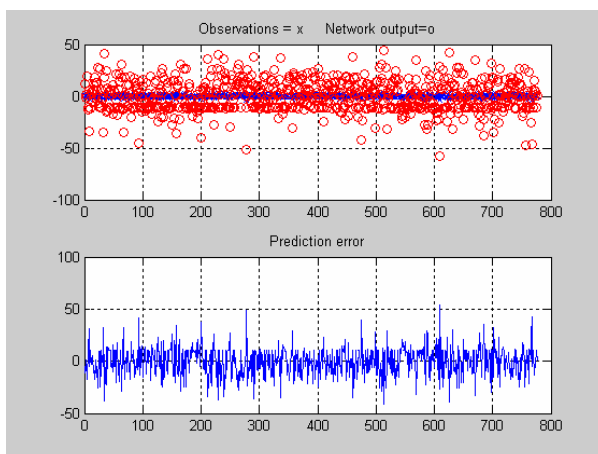
22-23pav. Neuroninio tinklo apsimokymas



24-25pav. Neuroninio tinklo apsimokymas



26-27pav. Neuroninio tinklo apsimokymas



28-29pav. Neuroninio tinklo apsimokymas

4.3. incbp funkcija

Tikslas

Atgalinio sklidimo algoritmo rekursyvinė versija.

Aprašymas

Įskaitant atitinkamą rinkinį įėjimo porų – išėjimo ir pradinio tinklo (INCBP) apsimoko tinklas su apdirbtais pasiskirščiusiais pasklidimais. . Aktivacijos funkcijos turi būti linijinės arba hiperbolinė. Tinklo architektūra apibūdina matricą, „NetDef“ susidedančio iš dviejų eilių. Pirmoji eilė apibūdina paslėptą sluoksnį, tuometu kai antra eilė apibūdina išėjimo sluoksnį.

Pavyzdžiui: NetDef = ['LHHHH'LL---']

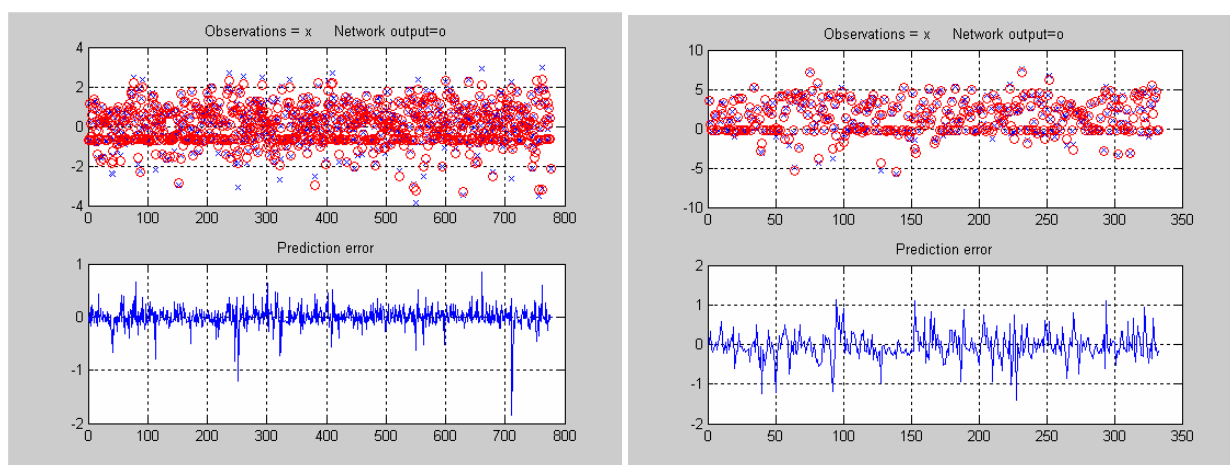
(L = Linijinis, H = hiperbilinis).

Atkreipkite dėmesį į tai, kad kryptis įtraukta kaip paskutinė kolona matricos svoryje [27].

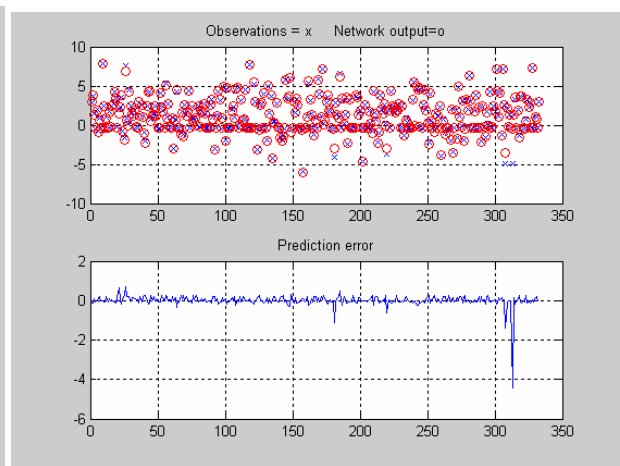
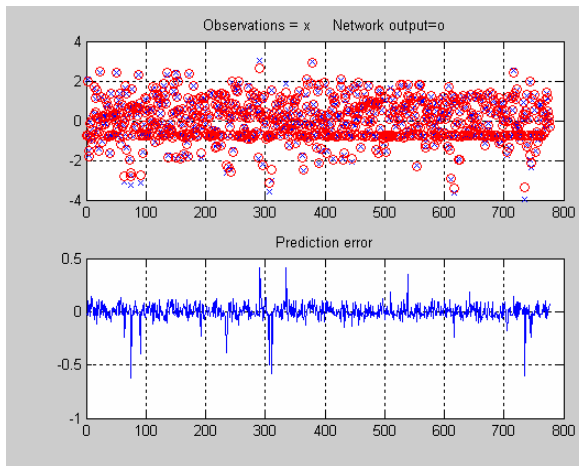
Naudojant **INCBP** funkciją buvo atlikta keletas neuroninio tinklo apmokymų ir gautos paklaidos. Programinis kodas 2 priede.

Lentelė Nr.3

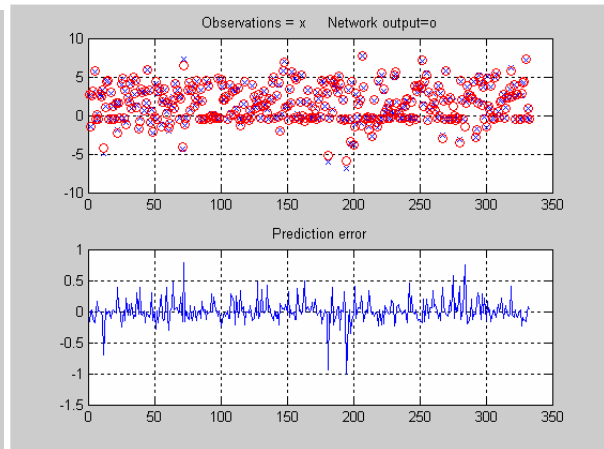
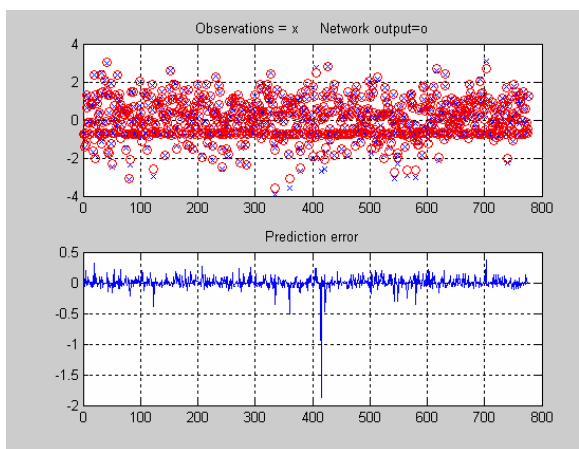
Iteracijų skaičius	Tinklo parametrai	NSSE	NSSE2
30	'HHLHHLHHHH';'L-----'	0,0172	0,0653
50	'HHLHHLHH';'L-----'	0,0031	0,0450
80	'HLHLLLH';'L-----'	0,0056	0,0168
30	'HLHLLH';'L-----'	0,0187	0,1837
100	'HLLLLH';'L-----'	0,0060	0,0600
VIDURKIS		0,01012	0,13872



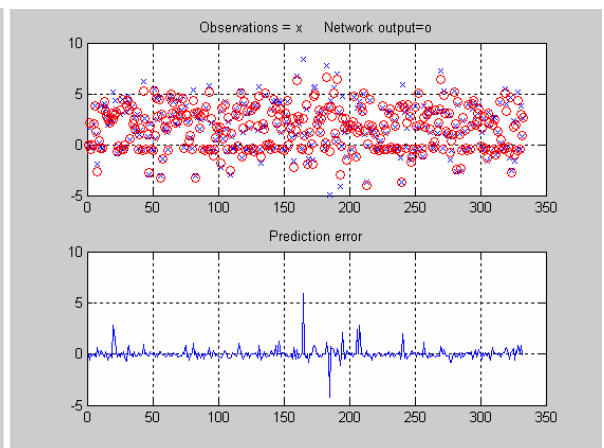
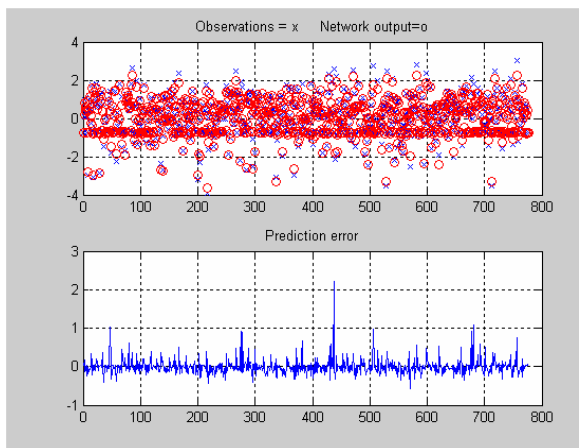
30-31pav. Neuroninio tinklo apsimokymas



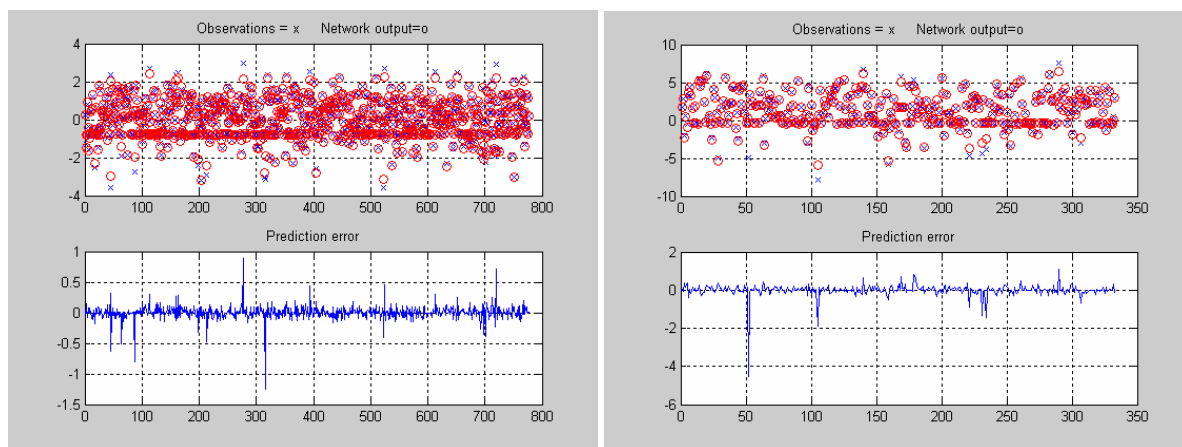
32-33pav. Neuroninio tinklo apsimokymas



34-35pav. Neuroninio tinklo apsimokymas



36-37pav. Neuroninio tinklo apsimokymas



38-39pav. Neuroninio tinklo apsimokymas

4.4. marq funkcija

Tikslas

Apmokyti tinklą su Levenbergo Markovo metodu

Aprašymas

Įskaitant atitinkamą rinkinį įėjimo porų – išėjimo ir pradinio tinklo, dviejų neuroninių tinklų sluoksnių apmokymas su Levenbergo Markovo metodu. Jeigu norima, galima naudoti reguliarizaciją kuri išskiria pagal svorį. Taip pat apmokyti gali būti nepilnai susiję tinklai. Aktyvacijos funkcijos gali būti linijinės arba tanh. Tinklo architektūra apibūdina matricą. „NetDef“ susidedančio iš dviejų eilių. Pirmoji eilė apibūdina paslėptą sluoksnį, tuometu kai antra eilė apibūdina išėjimo sluoksnį. Atkreipkite dėmesį į tai, kad kryptis įtraukta kaip paskutinė kolona matricos svoryje ir kad svoris sumažintas kalibruojant jį iki nulio. Galima apmokyti tinklą su paprastu svorių išdėstymu reguliarizuojant [27].

Algoritmas

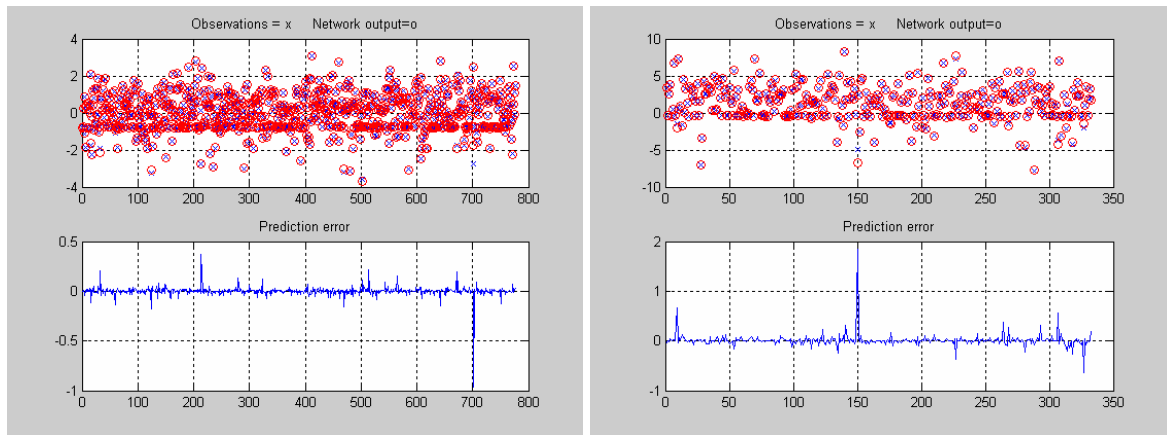
Algoritmas – standartinis Levenbergo – Markovo metodas.

Naudojant **MARQ** funkciją buvo atlikta keletas neuroninio tinklo apmokymų ir gautos paklaidos. Programinis kodas 1 priede.

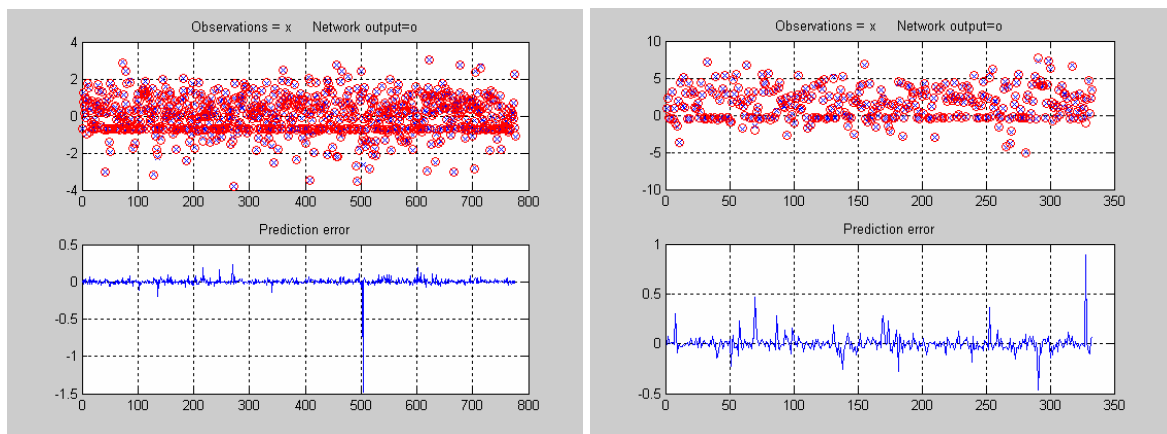
Lentelė Nr.4

Iteracijų skaičius	Tinklo parametrai	NSSE	NSSE2
30	'HHLHHLHHH';'L-----'	0,0011	0,0093
50	'HHLHHLHH';'L-----'	0,0019	0,0043

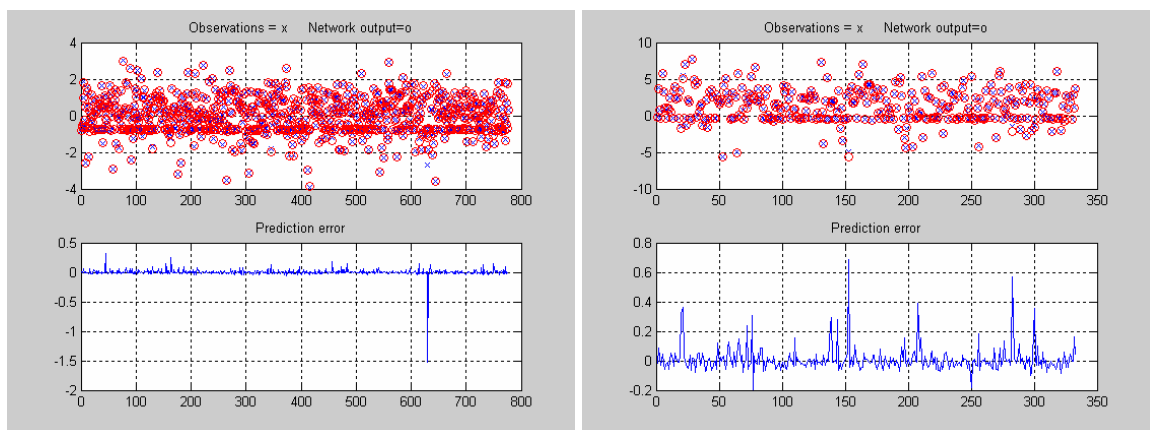
80	'HLHLLLH';'L-----'	0,0020	0,0037
30	'HLHLLH';'L-----'	0,0021	0,0016
100	'HLLLLH';'L-----'	0,0011	0,0286
VIDURKIS		0,00164	0,02462



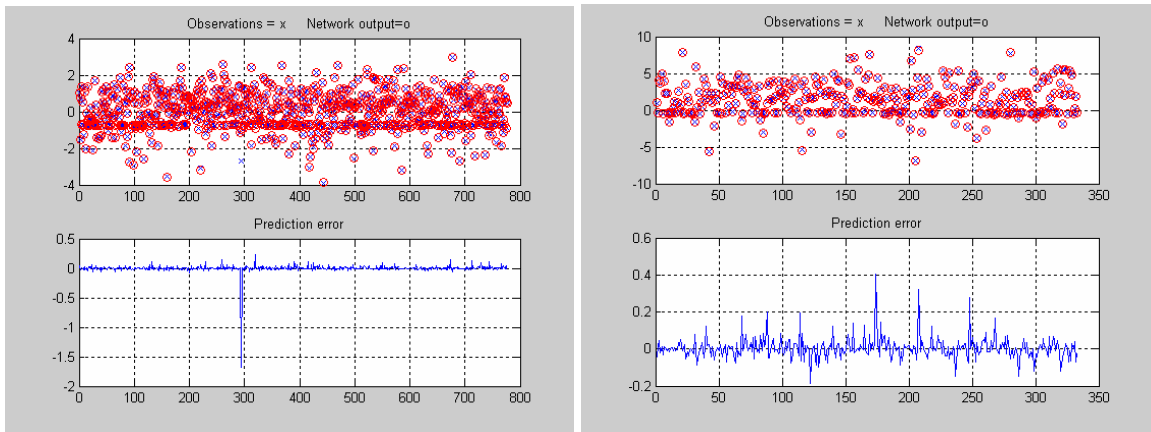
40-41pav. Neuroninio tinklo apsimokymas



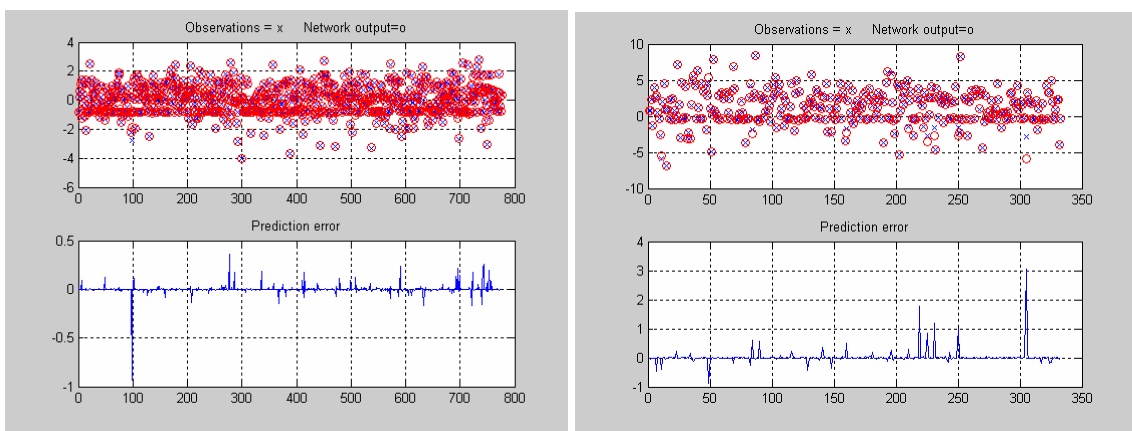
42-43pav. Neuroninio tinklo apsimokymas



44-45pav. Neuroninio tinklo apsimokymas



46-47pav. Neuroninio tinklo apsimokymas



48-49pav. Neuroninio tinklo apsimokymas

4.5. marqlm funkcija

Tikslas

Levenbergo – Markovo metodo atlikimas, naudojant mažiau atminties nei MARQ.

Aprašymas

Mažiau naudojanti atminties Levenbergo - Markovo algoritmo apmokymo versija, įgyvendinta MARQ. Greičio skirtumai atsiranda todėl, kad funkcija mažiau vektorizuota (tai ir yra Matlabo problema), o taip pat todėl, kad kai kurie nuskaitymai gali būti pakartoti [27].

4.6. rpe funkcija

Aprašymas

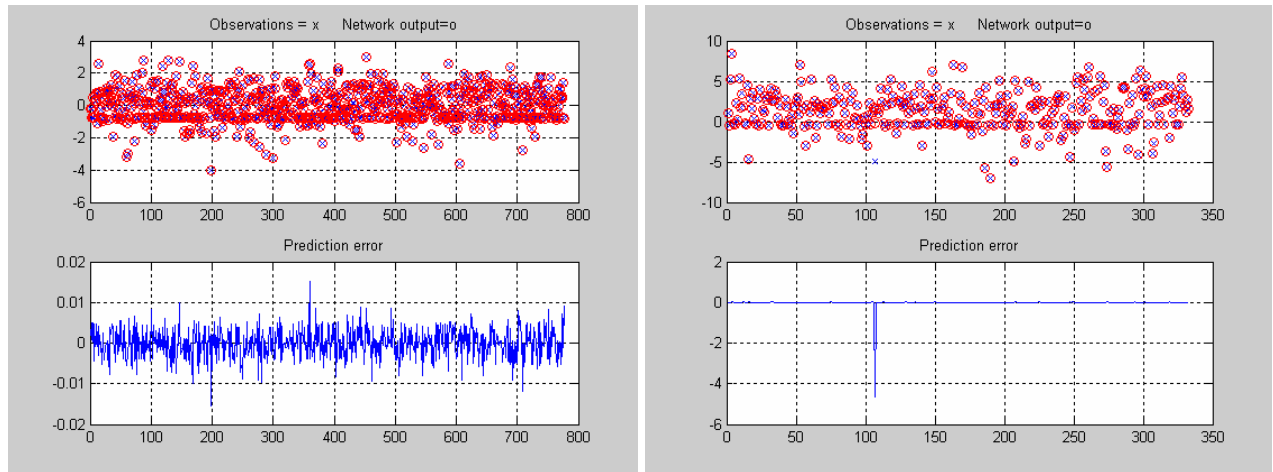
Įskaitant atitinkamą rinkinį įėjimo porų – išėjimo ir pradinio tinklo, dviejų neuroninių tinklų sluoksnių apmokymas su rekursyviu klaidos nusakymo metodu. („rekursyvinis Gauso-Niutono“). Taip pat sutrumpintas, t.y. nepilnai susijęs tinklas gali būti apmokytas. Rekursyviniai metodai gali

būti pritaikyti labai dideliems tinklams + duomenims, statomas, kur atminties trūkumas yra problema arba kai yra labai didelė paklaida duomenyse. Skirtingi metodai buvo įgyvendinti su įkvėpimu iš adaptuotos kontrolės{valdymo}: teigiamas nuolydis, nuolatinis pėdsakas ir taip vadinamas teigiamas nuolydis ir algoritmo perstatymas (EFRA). Nemokėjimo metodas – teigiamas algoritmo faktoriaus nuolydis [27].

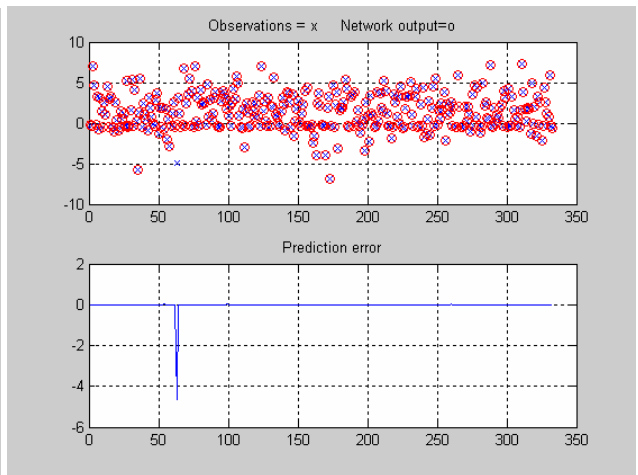
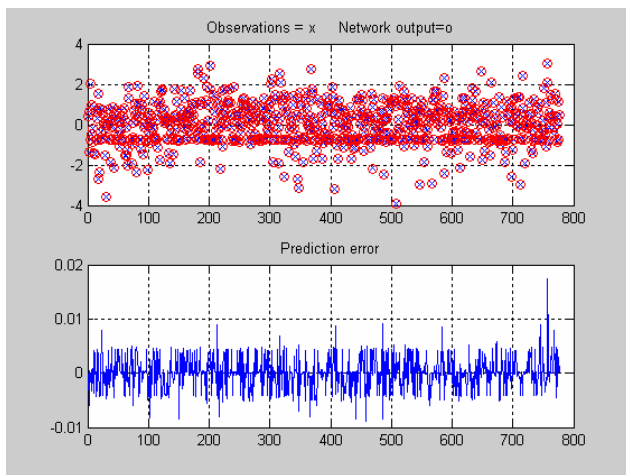
Naudojant **RPE** funkciją buvo atlikta keletas neuroninio tinklo apmokymų ir gautos paklaidos. Programinis kodas 3 priede.

Lentelė Nr.5

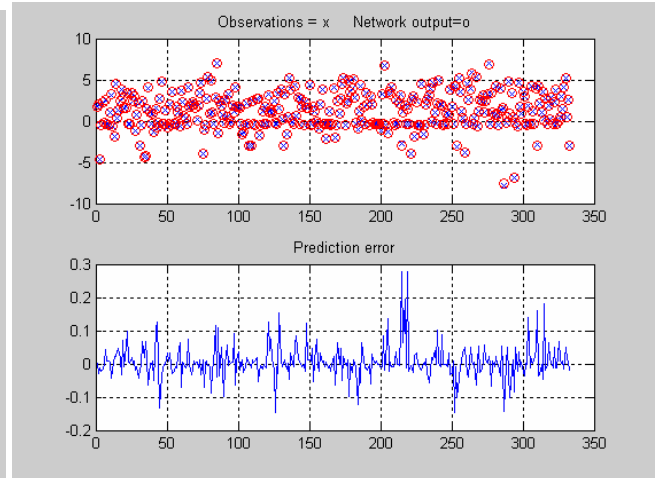
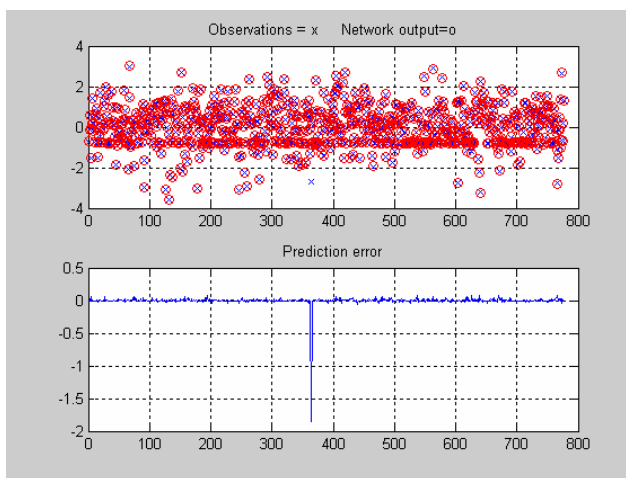
Iteracijų skaičius	Tinklo parametrai	NSSE	NSSE2
30	'HHLHHLHHHH';'L-----'	0.0023	0.0017
50	'HHLHHLHH';'L-----'	0,0025	0.0328
80	'HLHLLLH';'L-----'	0,0024	0,0013
30	'HLHLLH';'L-----'	0,0025	0,0413
100	'HLLLLH';'L-----'	0,0026	0,0018
VIDURKIS		0,01022	0,07746



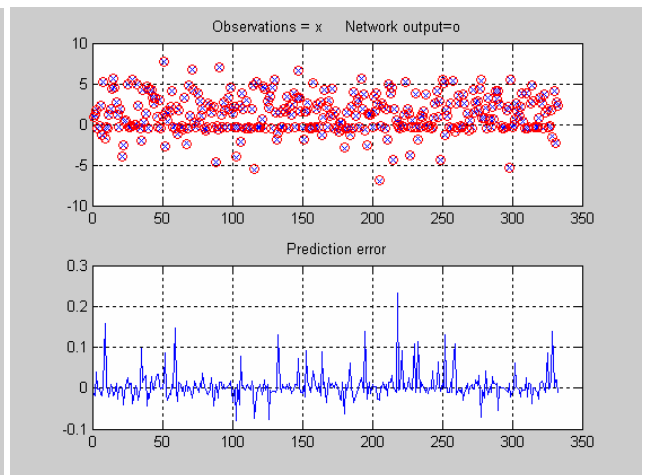
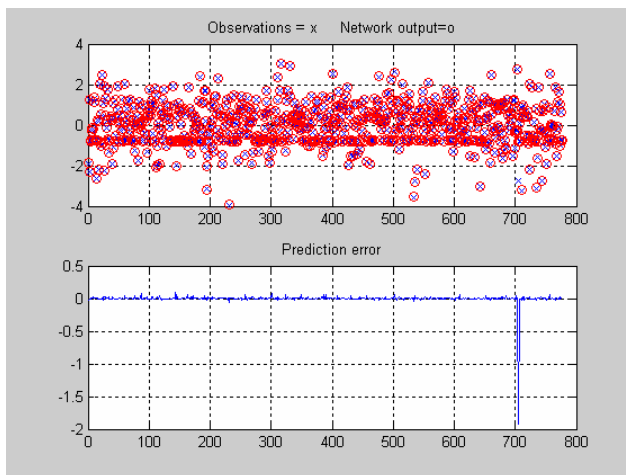
50-51pav. Neuroninio tinklo apsimokymas



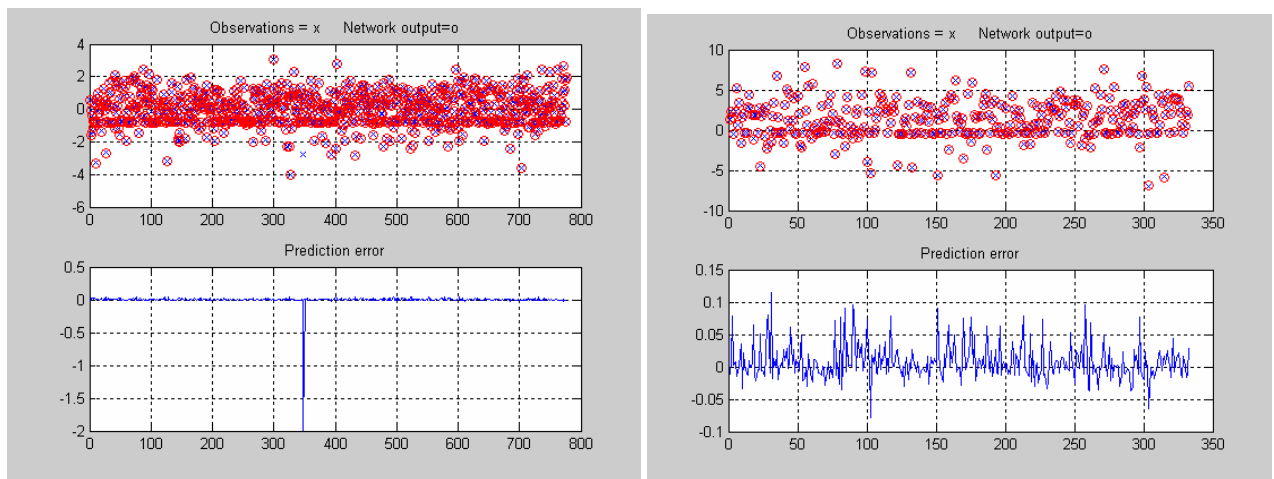
52-53pav. Neuroninio tinklo apsimokymas



54-55pav. Neuroninio tinklo apsimokymas



56-57pav. Neuroninio tinklo apsimokymas



58-59pav. Neuroninio tinklo apsimokymas

Funkcijų rezultatų palyginimas

Lentelė Nr.6

Funkcijos pavadinimas	NSSE	NSSE2
batbp	0,0225	0,5941
igls	659,3371	1041,68114
incbp	0,01012	0,13872
marq	0,00164	0,02462
rpe	0,01022	0,07746

Kaip matoma lentelėje Nr.6 mažiausia paklaidą daro Markovo Levenbergo metodas. Batbp, incbp ir rpe funkcijos duoda labai panašią paklaidą.

Išvados

1. Eksperimentas buvo atliekamas su penkiomis neuroninių tinklų apmokymo funkcijomis. Buvo naudojami tokie patys penki neuroninių tinklų konfigūravimai, su tiek pat apmokymų. Atliktas eksperimentas parodė, kad Markovo Levenbergo metodas duoda mažiausią paklaidą.
2. Funkcijų batbp , incbp ir rpe pagalba gauti rezultatai rodo, kad šios trys funkcijos duoda labai panašią paklaidą. Šių trijų funkcijų komandinėje eilutėje naudojami tokie patys kintamieji.
3. Praščiausius rezultatus padarė igls funkcija. Tai gali įtakoti daugkartinis funkcijos kreipimasis į neuroninio tinklo įėjimus.

LITERATŪROS SĄRAŠAS

1. Neural Network Applications. [interaktyvus] [žiūrėta 2004 m. lapkričio 30 d.]. Prieiga per internetą: <http://www.singaporegateway.com/products/nfga/intronn.htm>
2. Asian risk. [interaktyvus] [žiūrėta 2004 m. lapkričio 30 d.]. Prieiga per internetą: <http://www.asiarisk.com.hk/japanrisk/newangle3.htm>
3. Bankruptcy Prediction for Credit Risk Using Neural Networks: A Survey and New Results. [interaktyvus] [žiūrėta 2004 m. gruodžio 2 d.]. Prieiga per internetą: <http://alumnus.caltech.edu/~amir/bankr.pdf>
4. Credit risk assessment for a small business, Using multidimensional membership functions: design and applications. [interaktyvus] [žiūrėta 2004 m. gruodžio 2 d.]. Prieiga per internetą: <http://www.cs.montana.edu/angryk/pages/MSinCS.htm>
5. Inter Science. [interaktyvus] [žiūrėta 2004 m. gruodžio 2 d.]. Prieiga per internetą: <http://www3.interscience.wiley.com/cgi-bin/abstract/99018885/ABSTRACT>
6. The Bundesbank's method of assessing the creditworthiness of business enterprises. [interaktyvus] [žiūrėta 2004 m. gruodžio 5 d.]. Prieiga per internetą: http://www.bundesbank.de/download/volkswirtschaft/mba/1999/199901mba_art03_credwor th.pdf
7. Ccas: an intelligent decision support system for credit card Assessment. [interaktyvus] [žiūrėta 2004 m. gruodžio 5 d.]. Prieiga per internetą: <http://www4.fe.uc.pt/mcda56/docs/Papers/MATSATSI.PDF>
8. Modelling Sovereign Credit Ratings: Neural Networks versus Ordered Probit. [interaktyvus] [žiūrėta 2004 m. gruodžio 5 d.]. Prieiga per internetą: <http://www.management.soton.ac.uk/research/publications/documents/AF03-11.pdf>
9. The Relevance of Qualitative Information in Credit Rating. [interaktyvus] [žiūrėta 2004 m. gruodžio 14 d.]. Prieiga per internetą: http://www.pfingsttagung.unizh.ch/Papers/Bina_Lehmann.pdf
10. A fuzzy neural network for assessing the risk of fraudulent financial reporting. [interaktyvus] [žiūrėta 2004 m. gruodžio 14 d.]. Prieiga per internetą: <http://murphylibrary.uwlax.edu/ereserves/Burrowes/ACC%20327%20Spr04/Fuzzy%20neur al%20network%20for%20assessing%20the%20risk%20of.pdf>
11. The Relevance of Qualitative Information in Credit Rating. [interaktyvus] [žiūrėta 2004 m. gruodžio 17 d.]. Prieiga per internetą: <http://www.efmaefm.org/AcceptedPapers2003/LehmannBina/LehmannBina.pdf>
12. Fuzzy Neural Networks and Evolving Connectionist Systems for Intelligent Decision Support. [interaktyvus] [žiūrėta 2004 m. gruodžio 17 d.]. Prieiga per internetą:

- http://www.aut.ac.nz/research_showcase/research_activity_areas/kedri/downloads/pdf/ifs98tx.pdf
13. Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation . [interaktyvus] [žiūrėta 2004 m. gruodžio 17 d.]. Prieiga per internetą: http://pubsonline.informs.org/main/pdfstore/cecbaae209_abstract.pdf
 14. NeuroXL Classifier - Neural Network Finance Application - Credit and Loans. [interaktyvus] [žiūrėta 2004 m. gruodžio 17 d.]. Prieiga per internetą: http://www.neuroxl.com/finance_neural_network.htm
 15. Credit-risk solutions based on neural networks. [interaktyvus] [žiūrėta 2004 m. gruodžio 30 d.]. Prieiga per internetą: http://www.sas.com/offices/europe/belux/customer/cust_kul2.html
 16. DBMS, Data Mining Solutions Supplement. [interaktyvus] [žiūrėta 2004 m. gruodžio 30 d.]. Prieiga per internetą: <http://www.dbmsmag.com/9807m06.html>
 17. Data Mining. [interaktyvus] [žiūrėta 2004 m. sausio 5 d.]. Prieiga per internetą: <http://www.uni-koblenz.de/~moeh/lehre/ws0405/dm51.pdf>
 18. Sas[®] credit risk management. [interaktyvus] [žiūrėta 2004 m. sausio 5 d.]. Prieiga per internetą: <http://www.sas.com/industry/banking/credit/>
 19. Using Business Intelligence for Factoring Credit Analysis. [interaktyvus] [žiūrėta 2004 m. sausio 5 d.]. Prieiga per internetą: <http://www.mitgmbh.de/e/solutions/bertelsmann/bertelsmannenglisch.htm>
 20. Kreditų skaičiavimas,[interaktyvus] – [žiūrėta 2004 11 26], prieiga per internetą <http://www.calsci.com/CreditScoring.html>
 21. Merkevičius, Egidijus. *IT modelis banko vidinių reitingų sistemoje*. Kaunas: 2003. 67 p.
 22. Vilniaus universitetas. *Informacijos mokslai*. Kaunas: 2005. 344p. Merkevičius Egidijus, Garšva Gintautas. *Kreditavimo taisyklių reikšmingumo įvertinimas savitvarkių žemėlapių metodu*. [160p. – 165p.]
 23. Profesoriaus Rimvydo Simučio paskaitų medžiaga 2004-2005m.
 24. Self-Organizing Maps Applet [interaktyvus] [žiūrėta 2005 m. gruodžio 30 d.]. Prieiga per internetą: <http://davis.wpi.edu/~matt/courses/soms/applet.html>
 25. Understanding and Predicting Your Customers' Behavior [interaktyvus] [žiūrėta 2005 m. gruodžio 30 d.]. Prieiga per internetą: <http://www.eudaptics.de/home/index.php?sprache=en>
 26. "Risk Scoring with BizTalk Server 2004 and the Business Rules Framework" [interaktyvus] [žiūrėta 2005 m. gruodžio 30 d.]. Prieiga per internetą: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bts_2004wp/html/87e1b104-c39c-4b0d-a74e-494d672e9779.asp

27. „General Network Training Algorithms“ [interaktyvus] [žiūrėta 2005 m. gruodžio 30 d.].
Prieiga per internetą: <http://www.iau.dtu.dk/research/control/nlib/manual1.pdf>

PRIEDAI

1 Priedas

```
clear all, close all          % isvalo langus
load duomenys.txt;           % uzkrauna duomenis is tekstinio failo
[id,jd]=size(duomenys);      % matricos apimtis i susideda uzkrautus duomenis
% duomenu padalinimas 30:70 cross-validation
N=size(duomenys,1);
    crossVP=0.7; % Cross-validation dalinimo procentas
    NTRN = floor(crossVP*N); % select 70% for training
RandCols = randperm(N); % Randomize the vectors
    RandCols=RandCols'; % Transpose matrix
    ptrn = duomenys(RandCols(1:NTRN),:); % training i/p vector
    ptst = duomenys(RandCols(NTRN+1:N),:); % testing i/p vector
% duomenys paruosti apmokymui ir testavimui
itrc=100;
% svoriai random`u
W1=rand(6,6)-0.5; %
W2=rand(1,7)-0.5; %
X=ptrn;
% duomenys normalizuojami
[Xn,Skal]=dscale(X');
% apibreziamas NN tinklas
NetDef=['HLHLLH';'L-----'];
PHI=Xn(1:end-1,:);
Y=Xn(end,:);
% apmokymo iteraciju skaicius
trparms.maxiter = itrc;
% sukuriamas backproagation NN
[W1,W2,critvec,iter]=marq(NetDef,W1,W2,PHI,Y,trparms);
% apskaiciuojama NN paklaida
[Yhat,E,NSSE] = nneval(NetDef,W1,W2,PHI,Y);
Uscale=Skal(1:end-1,:); Yscale=Skal(end,:);
% keiciami svoriai
[AW1,AW2]=wrescale('marq',W1,W2,Uscale,Yscale);
Xtest=ptst;
Xa=Xtest';
Input=Xa(1:end-1,:);
Output=Xa(end,:);
[Yhat2,E,NSSE2] = nneval(NetDef,AW1,AW2,Input,Output);
```


2 Priedas

```
clear all, close all          % isvalo langus
load duomenys.txt;           % uzkrauna duomenis is tekstinio failo
[id,jd]=size(duomenys);      % matricos apimtis i susideda uzkrautus duomenis
% duomeniu padalinimas 30:70 cross-validation
N=size(duomenys,1);
    crossVP=0.7; % Cross-validation dalinimo procentas
    NTRN = floor(crossVP*N); % select 70% for training
RandCols = randperm(N); % Randomize the vectors
    RandCols=RandCols'; % Transpose matrix
    ptrn = duomenys(RandCols(1:NTRN),:); % training i/p vector
    ptst = duomenys(RandCols(NTRN+1:N),:); % testing i/p vector
% duomenys paruosti apmokymui ir testavimui
itrc=100;
% svoriai random`u
W1=rand(6,6)-0.5; %
W2=rand(1,7)-0.5; %
X=ptrn;
% duomenys normalizuojami
[Xn,Skal]=dscale(X');
% apibreziamas NN tinklas
NetDef=['HLHLLH';'L-----'];
PHI=Xn(1:end-1,:);
Y=Xn(end,:);
% apmokymo iteraciju skaicius
trparms.maxiter = itrc;
% sukuriamas backpropagation NN
[W1,W2,critvec,iter]=incbp(NetDef,W1,W2,PHI,Y,trparms);
% apskaiciuojama NN paklaida
[Yhat,E,NSSE] = nneval(NetDef,W1,W2,PHI,Y);
Uscale=Skal(1:end-1,:); Yscale=Skal(end,:);
% keiciami svoriai
[AW1,AW2]=wrescale('incbp',W1,W2,Uscale,Yscale);
Xtest=ptst;
Xa=Xtest';
Input=Xa(1:end-1,:);
Output=Xa(end,:);
[Yhat2,E,NSSE2] = nneval(NetDef,AW1,AW2,Input,Output);
```

3 Priedas

```
clear all, close all          % isvalo langus
load duomenys.txt;           % uzkrauna duomenis is tekstinio failo
[id,jd]=size(duomenys);      % matricos apimtis i susideda uzkrautus duomenis
% duomeniu padalinimas 30:70 cross-validation
N=size(duomenys,1);
    crossVP=0.7; % Cross-validation dalinimo procentas
    NTRN = floor(crossVP*N); % select 70% for training
RandCols = randperm(N); % Randomize the vectors
    RandCols=RandCols'; % Transpose matrix
    ptrn = duomenys(RandCols(1:NTRN),:); % training i/p vector
    ptst = duomenys(RandCols(NTRN+1:N),:); % testing i/p vector
% duomenys paruosti apmokymui ir testavimui
itrc=100;
% svoriai random`u
W1=rand(6,6)-0.5; %
W2=rand(1,7)-0.5; %
X=ptrn;
% duomenys normalizuojami
[Xn,Skal]=dscale(X');
% apibreziamas NN tinklas
NetDef=['HLLLLH';'L-----'];
PHI=Xn(1:end-1,:);
Y=Xn(end,:);
% apmokymo iteraciju skaicius
trparms.maxiter = itrc;
% sukuriamas backproagation NN
[W1,W2,critvec,iter]=rpe(NetDef,W1,W2,PHI,Y,trparms);
% apskaiciuojama NN paklaida
[Yhat,E,NSSE] = nneval(NetDef,W1,W2,PHI,Y);
Uscale=Skal(1:end-1,:); Yscale=Skal(end,:);
% keiciami svoriai
[AW1,AW2]=wrescale('rpe',W1,W2,Uscale,Yscale);
Xtest=ptst;
Xa=Xtest';
Input=Xa(1:end-1,:);
Output=Xa(end,:);
[Yhat2,E,NSSE2] = nneval(NetDef,AW1,AW2,Input,Output);
```

4 Priedas

```
clear all, close all          % isvalo langus
load duomenys.txt;           % uzkrauna duomenis is tekstinio failo
[id,jd]=size(duomenys);      % matricos apimtis i susideda uzkrautus duomenis
% duomeniu padalinimas 30:70 cross-validation
N=size(duomenys,1);
    crossVP=0.7; % Cross-validation dalinimo procentas
    NTRN = floor(crossVP*N); % select 70% for training
RandCols = randperm(N); % Randomize the vectors
    RandCols=RandCols'; % Transpose matrix
    ptrn = duomenys(RandCols(1:NTRN),:); % training i/p vector
    ptst = duomenys(RandCols(NTRN+1:N),:); % testing i/p vector
% duomenys paruosti apmokymui ir testavimui
itrc=30;
% svoriai random`u
W1=rand(10,6)-0.5; %
W2=rand(1,11)-0.5; %
X=ptrn;
% duomenys normalizuojami
[Xn,Skal]=dscale(X');
% apibreziamas NN tinklas
NetDef=['HHLHHLHHHH';'L-----'];
PHI=Xn(1:end-1,:);
Y=Xn(end,:);
% apmokymo iteraciju skaicius
trparms.maxiter = itrc;
% sukuriamas backproagation NN
[W1,W2,lambda,Gamma]=igls(NetDef,W1,W2,trparms,[],PHI,Y);
% apskaiciuojama NN paklaida
[Yhat,E,NSSE] = nneval(NetDef,W1,W2,PHI,Y);
Uscale=Skal(1:end-1,:); Yscale=Skal(end,:);
% keiciami svoriai
[AW1,AW2]=wrescale('igls',W1,W2,Uscale,Yscale);
Xtest=ptst;
Xa=Xtest';
Input=Xa(1:end-1,:);
Output=Xa(end,:);
[Yhat2,E,NSSE2] = nneval(NetDef,AW1,AW2,Input,Output);
```

5 Priedas

```
clear all, close all          % isvalo langus
load duomenys.txt;           % uzkrauna duomenis is tekstinio failo
[id,jd]=size(duomenys);      % matricos apimtis i susideda uzkrautus duomenis
% duomeniu padalinimas 30:70 cross-validation
N=size(duomenys,1);
    crossVP=0.7; % Cross-validation dalinimo procentas
    NTRN = floor(crossVP*N); % select 70% for training
RandCols = randperm(N); % Randomize the vectors
    RandCols=RandCols'; % Transpose matrix
    ptrn = duomenys(RandCols(1:NTRN),:); % training i/p vector
    ptst = duomenys(RandCols(NTRN+1:N),:); % testing i/p vector
% duomenys paruosti apmokymui ir testavimui
itrc=100;
% svoriai random`u
W1=rand(6,6)-0.5; %
W2=rand(1,7)-0.5; %
X=ptrn;
% duomenys normalizuojami
[Xn,Skal]=dscale(X');
% apibreziamas NN tinklas
NetDef=['HLLLLLH';'L-----'];
PHI=Xn(1:end-1,:);
Y=Xn(end,:);
% apmokymo iteraciju skaicius
trparms.maxiter = itrc;
% sukuriamas backpropagation NN
[W1,W2,critvec,iter]=batbp(NetDef,W1,W2,PHI,Y,trparms);
% apskaiciuojama NN paklaida
[Yhat,E,NSSE] = nneval(NetDef,W1,W2,PHI,Y);
Uscale=Skal(1:end-1,:); Yscale=Skal(end,:);
% keiciami svoriai
[AW1,AW2]=wrescale('batbp',W1,W2,Uscale,Yscale);
Xtest=ptst;
Xa=Xtest';
Input=Xa(1:end-1,:);
Output=Xa(end,:);
[Yhat2,E,NSSE2] = nneval(NetDef,AW1,AW2,Input,Output);
```