



VILNIAUS PEDAGOGINIS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
INFORMACINIŲ TECHNOLOGIJŲ KATEDRA

ANDREJUS ANDREJEVAS

# Rašto ženklų atpažinimas naudojant neuroninius tinklus

Magistro darbas

Darbo vadovas: dr. O. Kurasova

Vilnius, 2011

## Turinys

1. Įvadas.....	3
1.1 Temos aktualumas .....	3
1.2 Darbo tikslas .....	3
1.3 Darbo struktūra .....	4
2 Dirbtiniai neuroniniai tinklai .....	5
2.1 Žmogaus smegenys ir biologinis neuronas.....	5
2.2 Kompiuteris ir žmogaus smegenys.....	6
2.3. Dirbtinio neurono modelis.....	7
2.4 Aktyvavimo funkcijos .....	8
2.5 Vienasluoksnis neuroninis tinklas .....	9
2.6 Daugiasluoksniai neuroniniai tinklai .....	10
2.7 Klaidos skleidimo atgal algoritmas .....	11
2.8 Dirbtiniai neuroniniai tinklai rašto ženklams atpažinti.....	13
2.8.1 Daugiasluoksnis perceptronas .....	13
2.8.2 Sąsūkinis neuroninis tinklas .....	16
2.9 Atpažinimo rezultatų vertinimas .....	18
3. Rašto ženklų atpažinimo sistemos .....	21
3.1 Sistemų analizė .....	21
3.2 Sistemų patobulinimas.....	22
3.3 Sistemos papildymas moduliu, skirtu eksperimentams atlikti.....	25
3.4 Mokymo dalimis strategija .....	27
4. Eksperimentiniai tyrimai .....	30
4.1 Atpažinimo rezultatų priklausomybė nuo svorių pradinių reikšmių parinkimo.....	30
4.2 Optimalaus intervalo parinkimas.....	35
4.3 Mokymo algoritmų palyginimas .....	35
5. Apibendrinimas ir išvados .....	37
SANTRAUKA .....	39
SUMMARY .....	40
Literatūra .....	41
Priedas .....	42

# 1. Įvadas

## 1.1 Temos aktualumas

Rašto ženklų atpažinimas jau gana seniai intensyviai nagrinėjama sritis, tačiau vis plačiau diegiant įvairias naujas technologijas (delninkai, lietimuisi jautrūs ekranai ir kt.) į kasdienę žmonių veiklą, kyla vis naujos problemos, kurių sprendimas reikalauja papildomų tyrimų. Lietuvoje dar 1968 m. buvo sukurtas skaitantis įrenginys „Rūta-701“. Šis įrenginys buvo sukurtas nuskaityti arabiškus skaitmenis, ranka rašytiems arba spausdintiems tekstams [1]. Už šį projektą prof. L. Telksnys apdovanotas Lietuvos Valstybine premija.

Šiuo metu spausdinto teksto atpažinimas, turint teksto paveikslėlius (pvz. skenuotas, spausdintas tekstas) yra galimas, jei tas paveikslėlis yra ryškus. Tikslumas tokiu atveju pasiekiamas daugiau nei 99 %. Visiškas tikslumas pasiekiamas tik po žmogaus redagavimo. Dažnai naudojamos spausdinto ar skenuoto teksto atpažinimo programos yra:

1. Komerinės: TypeReader & RTK, ABBYY FineReader, OmniPage, Readiris, Persian Reader, Kirtas Technologies Arabic OCR (gali atpažinti arabiškus simbolius), Zonal OCR, Microsoft Office Document Imaging, Microsoft Office OneNote 2007 ir kitos.
2. Nekomercinės: SimpleOCR, COCR2 (atpažįsta kinų hieroglifus), LiveOCR.

## 1.2 Darbo tikslas

Darbo tikslas – sukurti dirbtinių neuroninių tinklų, skirtų rašto ženklams atpažinti, mokymo strategiją, leisiančią sutrumpinti mokymo laiką neprarandant atpažinimo tikslumo.

Norint pasiekti šį tikslą, reikėjo išspręsti tokius uždavinius:

1. Išanalizuoti neuroniniais tinklais pagrįstus algoritmus, skirtus rašto ženklams atpažinti.
2. Realizuoti rašto ženklų atpažinimo algoritmą, pagrįstą neuroniniais tinklais, mokomais klaidos skleidimo atgal algoritmu.
3. Sukurti grafinę vartotojo sąsają, kuri leistų keisti algoritmo parametrus, tam kad nustatyti optimalias jų reikšmes, pvz., paslėptų neuronų skaičių, mokymo epochų kiekį, paklaidos slenkstį, nuo kurio priklauso mokymo trukmė, raidžių, naudojamų mokyme, skaičių.
4. Sukurti funkcijas rašto ženklo vektoriaus koregavimui – centravimui ir padidinimui.
5. Ištirti atpažinimo tikslumo priklausomybę nuo neuroninio tinklo pradinių svorių parinkimo.

6. Modifikuoti neuroninio tinklo mokymo strategiją, leisiančią sutrumpinti mokymo laiką neprarandant atpažinimo tikslumo.

### **1.3 Darbo struktūra**

Darbas sudarytas iš penkių dalių, literatūros sąrašo ir internetinių nuorodų.

**Pirma darbo dalis** – įvadas. Jame paaiškinta temos pasirinkimo motyvacija (temos aktualumas), darbo tikslas bei uždaviniai.

**Antra darbo dalis** skirta neurobiologijai, t. y., paaiškinama bendra smegenų struktūra, kas yra biologinis neuronas ir jo sudedamosios dalys (synapse, soma, aksona), smegenų ląstelių tipai, neuroniniai tinklai, bei jų darbo principai, taip pat skirtumai tarp žmogaus smegenų ir kompiuterio. Apžvelgta, kas yra dirbtinis neuronas, jo modelis, pradinių svorių vektoriai, dirbtinio neurono aktyvavimo funkcijos (slenkstinė, sigmoidinė ir hiperbolinis tangentas). Įvesta dirbtinių neuroninių tinklų (DNT) sąvoka. Aprašyta tinklo mokymo galimybė naudojant klaidos skleidimo atgal algoritmą. Taip pat supažįstama su klasifikavimo tikslumo įvertinimo matais – jautrumu ir specifiškumu.

**Trečia darbo dalis** skirta dirbtinio neuroninio tinklo struktūros bei jo mokymo strategijos pasirinkimui. Apžvelgiami keli darbai šia tema. Paaiškinti sprendžiamo uždavinio dažniausiai kylančių problemų sprendimo būdai. Pristatomas sistemos patobulinimas moduliais, skirtais palengvinti bei pagreitinti eksperimentinių tyrimų atlikimą. Išryškunami mokymo algoritmo trūkumai bei algoritmo patobulinimas, kuris leidžia greičiau ir kokybiškiau atlikti rašto ženklų atpažinimą, t. y., pristatoma darbe pasiūlyta mokymo dalimis strategija, kai mokymo aibė paduodama į neuroninį tinklą ne visa iškart, o dalinama į kelis poaibius ir mokymas vykdomas etapais.

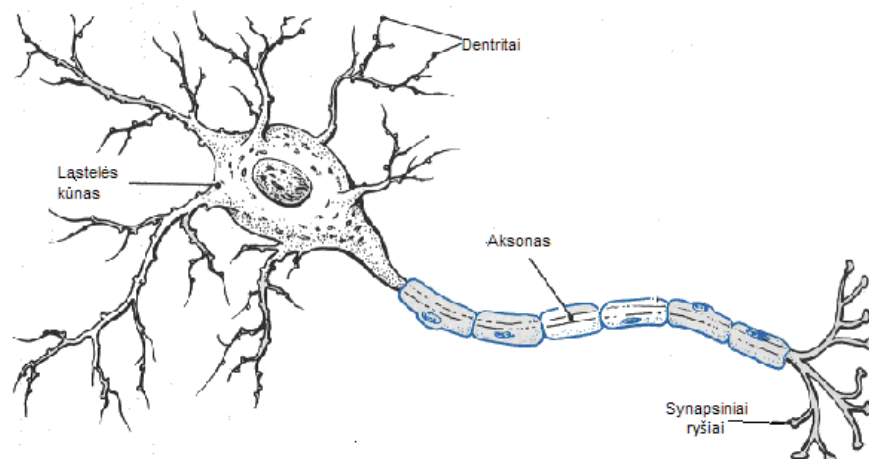
**Ketvirtoje darbo dalyje** aprašyti eksperimentiniai tyrimai: pradinių svorių generavimo intervalo pasirinkimas priklausomai nuo gautų klasifikavimo rezultatų, mokymo strategijų palyginimas, t. y., palyginamas standartinio mokymo algoritmas ir mokymas dalimis.

**Penktoji darbo dalis** skirta gautų rezultatų apibendrinimui ir išvadoms.

## 2 Dirbtiniai neuroniniai tinklai

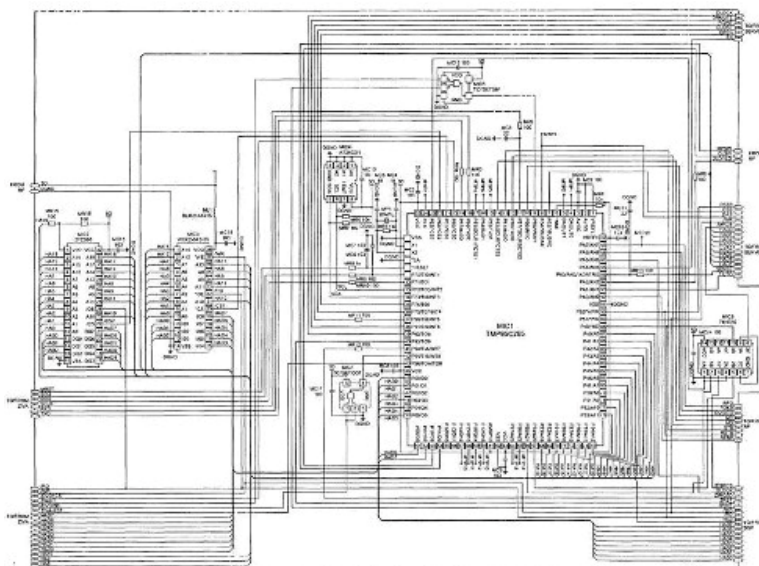
### 2.1 Žmogaus smegenys ir biologinis neuronas

Žmogaus smegenys yra biologinis kompiuteris, susidedantis iš daugelio neuronų. Šie neuronai yra sujungti vieni su kitais. Vidutiniškai smegenys turi apie 100 milijardų neuronų ir kiekvienas iš tų neuronų turi apie 10 tūkstančių jungčių. Neuronas turi išsišakojusią įėjimo struktūrą (dendritai), ląstelės kūną (soma) ir besišakojančią išėjimo struktūrą (aksonas) (žr. 2.1 pav.).



2.1 pav. Biologinis neuronas.

Neuronai tarpusavyje bendrauja labai trumpais impulsais. Žmogaus smegenys yra sudėtingiausi pasaulyje duomenų apdorojimo įrenginiai. Kompiuterio centrinis procesorius (CP) kuriamas toks, kad galėtų atlikti funkcijas, kurias atlieka žmogaus smegenys. Jame (CP) integruota milijonai tranzistorių (žr. 2.2 pav.), vyksta informacijos priėmimas, apdorojimas ir perdavimas. Už panašius veiksmus atsako biologinis neuronų tinklas, t. y., mintys, atmintis, emocijos, kūno motoriką ir kt.



2.2 pav. Centrinis procesorius.

Smegenyse nuolat miršta didelis neuronų kiekis, bet vis dėlto smegenys labai patvarios, jos ir toliau sėkmingai funkcionuoja. Biologinių neuronų tarpusavio komunikacijos greitis yra pakankamai lėtas (milijonus kartų lėčiau nei kompiuterio). Bet sudėtingi uždaviniai, tokie kaip raidžių ar veido atpažinimas, vyksta per kelias šimtas milisekundžių. Toks greitis pasiekiamas todėl, kad priimant sprendimą (pvz., veido atpažinimo) dalyvauja labai didelis kiekis neuronų, be to skaičiavimai vyksta lygiagrečiai.

Kiekvienas neuronas turi daug įvairių savybių, kai kurie iš jų yra bendri su kitais kūno elementais. Bet jis turi ir unikalias savybes, t. y., priėmimą, apdorojimą ir perdavimą elektrocheminiu signalu [2]. Biologinis neuronas susideda iš trijų dalių (žr. 2.1 pav.). Kiekviena iš jų atlieka savo funkciją. Dendritai sujungia ląstelės kūną (soma) su kitais neuronais. Šiose sujungimo taškuose (sinapse) jie gauna įėjimo signalus [3]. Priimti signalai keliauja iki ląstelės kūno, kur jie yra sumuojami. Šie signalai gali būti dvejų tipų: vieni bando sužadinti neuroną, kiti, priešingai, stengiasi sutrukdyti neurono sužadinimą. Jei suminis sužadinančių neuronų skaičius viršija tam tikrą slenkstį, tai neuronas aksonu siunčia signalą kitiems neuronams. Aišku, kad ši schema turi daug išimčių ir pasunkinimų, bet daugelis dirbtinių neuroninių tinklų modeliuoja būtent tokią schemą.

## 2.2 Kompiuteris ir žmogaus smegenys

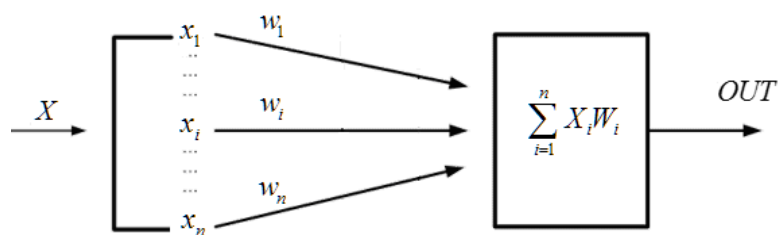
Žmogaus smegenų veikimo pagrindinę teoriją ir matematinį neurono modelį 1943 m. suformavo W.S. McCulloch ir W. Pitts. Jie pasiūlė teoriją, pagal kurią smegenys sudaryti iš daugelio neuronų, turinčių vienodą struktūrą. Į kiekvieną iš jų galima žiūrėti kaip į kažkokią slenkstinę funkciją. Jei leidžiamas slenkstis viršijamas, kiti neuronai gauna išėjimo signalą [4].

Tiriant žmogaus smegenis buvo sumodeliuota ir dirbtinių neuroninių tinklų struktūra. Dirbtiniai neuroniniai tinklai skiriasi vieni nuo kitų funkcionalumu, konfigūracijomis ir taikymo sritimis.

Kaip jau minėta yra analogija tarp žmogaus smegenų ir kompiuterio. Žmogaus smegenys, kaip ir kompiuteris, sudaryti iš daugelio paprastų elementų. Vykdo „skaičiavimo“ funkcijas, operuoja elektroniniais signalais, turi galimybę mokytis, priimti abstrakčius sprendimus ir daryti klaidas. Tačiau galima rasti ir fundamentalių skirtumų tarp žmogaus smegenų ir kompiuterio [5]. Nerviniai neuronų impulsai yra gana lėti, jei mes jas lyginsime su šiuolaikinio kompiuterio galimybėmis. Kiekvienas smegenų neuronas reikalauja milisekundinio užlaikymo (apdorojant gautą informaciją), tuo tarpu šiuolaikiniai kompiuteriai reikalauja nanosekundinio [6]. Bet galimybė vykdyti skaičiavimus lygiagrečiai suteikia žmogui pranašumą ir greitį. Klaidos – dar vienas skirtumas tarp smegenų ir kompiuterio. Klaidų darymas būdingas žmogui, bet ne kompiuteriu. Kompiuteris nedaro klaidų, jei programinė įranga, įėjimo parametrai yra tikslūs ir nesugadinti, o žmogaus smegenys gana dažnai daro klaidas, bet tų klaidų kiekis turi garantuoti žmonijos išgyvenimą [7].

### 2.3. Dirbtinio neurono modelis

Dirbtinis neuronas bando imituoti biologinį neuroną. Įėjime turime signalą  $X$  (vektorių), kurį mes gauname kaip pradinius duomenis arba kaip kitų neuronų išėjimus. Kiekviena šio vektoriaus komponentė  $x_i$  dauginama iš atitinkamos svorio  $w_i$  ir visos šios sandaugos sumuojamos (žr. 2.3 pav.). Ši suma ir lemia neurono aktyvavimo lygį.

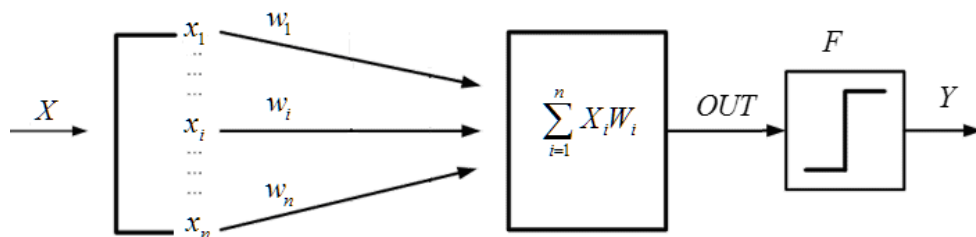


2.3 pav. Dirbtinio neurono modelis.

Dirbtinio neurono modelis parodytas 2.3 paveiksle. Įėjimo vektorius  $X(x_1, x_2, \dots, x_n)$  dauginamas iš atitinkamo svorio  $W(w_1, w_2, \dots, w_n)$  ir visos gaunamos sandaugos sumuojamos.

## 2.4 Aktyvavimo funkcijos

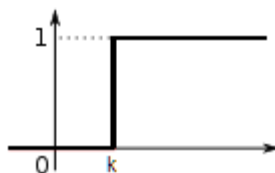
Pagal sužadavimo signalą  $OUT$  (žr. 2.3 pav.) naudojant aktyvavimo funkciją  $F$ , gaunamas neurono išėjimo signalas  $Y$  (žr. 2.4 pav.).



2.4 pav. Dirbtinio neurono modelis.

Galimi keli aktyvavimo funkcijos variantai. Tai gali būti slenkstinė funkcija  $F(OUT)$  (2.1), kuri skaičiuoja neurono išėjimo reikšmę  $Y$ . Ši reikšmė tampa lygi vienam, jei suma viršija arba yra lygi slenkstinei reikšmei  $k$ , priešingu atveju, nuliui (žr. 2.5 pav.).

$$Y = f(OUT) = \begin{cases} 1, & \text{jeigu } OUT \geq k \\ 0, & \text{jeigu } OUT < k \end{cases} \quad (2.1)$$

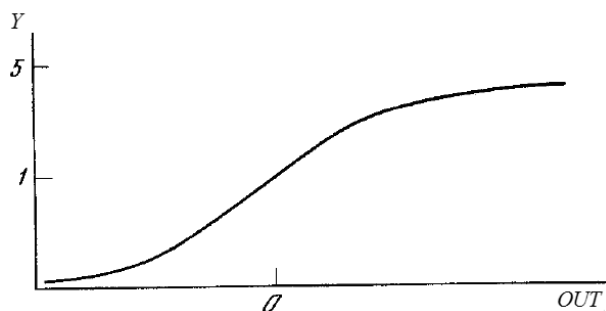


2.5 pav. Slenkstinė funkcija.

Dirbtinio neurono modelyje naudojama ne tik (2.1) funkcija. Kita dažnai naudojama aktyvacijos funkcija yra sigmoidinė funkcija (2.2) (žr. 2.6 pav.).

$$Y = f(OUT) = \frac{1}{1 + e^{-OUT}} \quad (2.2)$$

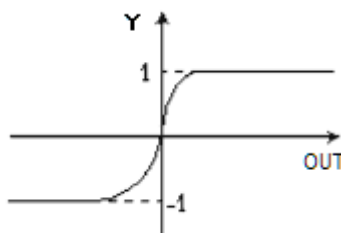




2.6 pav. Sigmoidinė funkcija.

Hiperbolinis tangentas yra dar viena dažnai naudojama aktyvavimo funkcija (2.3) (žr. 2.7 pav.). Pagal savo formą ji panaši į sigmoidinę funkciją, dažnai naudojama biologų, kaip neurono matematinis modelis. Ši funkcija užrašoma taip:

$$Y = f(OUT) = \tanh(OUT) = \frac{e^{OUT} - e^{-OUT}}{e^{OUT} + e^{-OUT}} \quad (2.3)$$



2.7 pav. Hiperbolinis tangentas.

Matome, kad hiperbolinis tangentas panašus į sigmoidinę funkciją, bet jis yra simetrinis koordinačių pradžios atžvilgiu. Taške  $OUT = 0$  išėjimo signalo  $Y$  reikšmė ligi 0 (žr. 2.7 pav.). Dar vienas svarbesnių skirtumų yra tai, kad išėjimo signalas  $Y$  gali įgauti ir teigiamą ir neigiamą reikšmę, o tai suteikia privalumą sprendžiant kai kuriuos uždavinius, susijusias su dirbtiniais neuroniniais tinklais.

## 2.5 Vienasluoksnis neuroninis tinklas

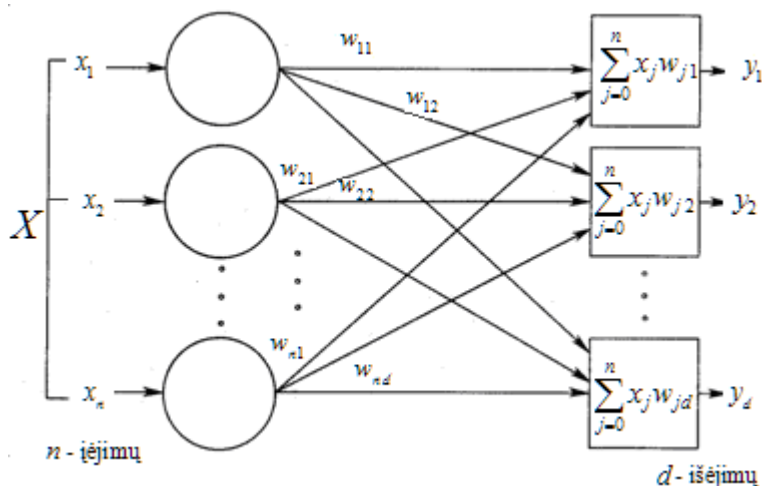
Nors ir vienas dirbtinis neuronas teoriškai gali atlikti paprastus klasifikavimo (atpažinimo) uždavinius, bet visą neuroninių tinklų galią galima pamatyti jungiant neuronus į tinklą. Paprasčiausios architektūros neuroninis tinklas – perceptronas [8]. Jis susideda iš vieno sluoksnio neuronų (žr. 2.8 pav.). Apskritimai paveikslo kairėje pusėje – tai įėjimo signalai (paskirstytojai), kurie nevykdo jokių skaičiavimų. Kiekvienas įėjimo sluoksnio elementas  $X$ , perduodamas neuronui, turi atitinkamą svorį  $w_{ij}$ . Šis užrašas  $w_{ij}$  reiškia jungties iš  $i$ -tosios įėjimo vektoriaus

komponentės į  $j$ -tąjį išėjimą svorį. Kiekvienas išėjimas  $y_i$  yra įėjimo vektoriaus  $X = (x_1, x_2, \dots, x_n)$  funkcija, kuri skaičiuojama pagal (2.4) formulę [15].

$$y_i = f(OUT_i) = f\left(\sum_{j=0}^n x_j w_{ji}\right), i = 1, \dots, d \quad (2.4)$$

čia  $d$  yra neuroninio tinklo išėjimų skaičius.

Funkcija  $f$  yra aktyvavimo funkcija. Dažniausiai ji skaičiuojama pagal (2.1) (2.2) (2.3) formules.



2.8 pav. Vienasluoksnis neuroninis tinklas (perceptronas).

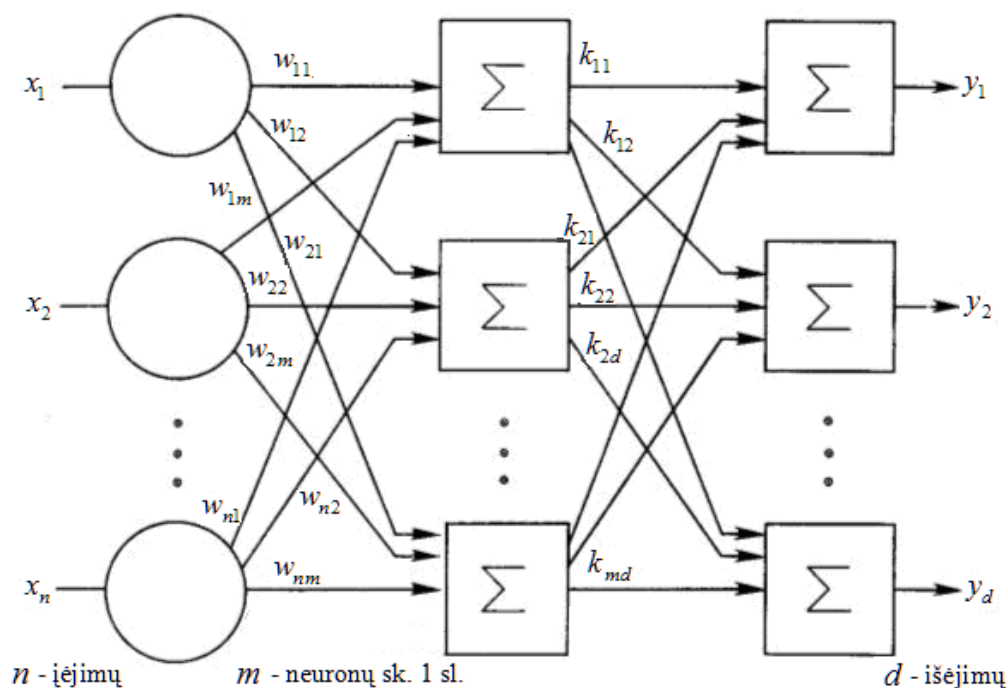
Tegu įėjimo vektorius  $X = (x_1, x_2, \dots, x_n)$  yra susietas su trokštamų reikšmių vektoriumi  $T = (t_1, t_2, \dots, t_d)$ , čia  $T$  – norimo išėjimo reakcija į  $X$ . Vienasluoksnio perceptrono mokymo tikslas – rasti tokius svorius, kad skirtumas tarp gaunamų išėjimo reikšmių  $y_i$  ir norimų reikšmių  $t_i$  būtų kuo mažesnis, t. y., minimizuojama tinklo paklaida  $E(W)$ , kuri dažnai išreiškiama formule (2.5).

$$E(W) = \frac{1}{2} \sum_{i=1}^d (y_i - t_i)^2 \quad (2.5)$$

## 2.6 Daugiasluoksniai neuroniniai tinklai

Neuronų tinklai, turintys daugiau nei vieną neuronų sluoksnį, vadinami daugiasluoksniais neuroniniais tinklais. Šie tinklai turi ir didesnių atpažinimo galimybių [9].

Kiekvienas toks tinklas sudarytas iš įėjimų aibės, išėjimų neuronų sluoksnio ir paslėptų neuronų sluoksnių tarp įėjimų ir išėjimų (žr. 2.9 pav.).



2.9 pav. Dviejų sluoksnių neuroninis tinklas.

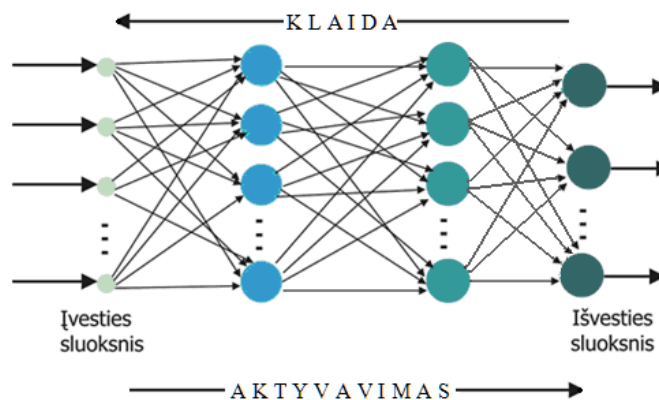
2.9 paveiksle  $w_{ij}$  yra pirmo neuroninio sluoksniu svorių koeficientai,  $k_{ij}$  antro neuroninio sluoksniu svorių koeficientai. Dažnai tokiam tinklui apmokyti naudojamas klaidos sklaidimo atgal algoritmas (angl. *error back propagation*).

Visi neuroniniai tinklai, apie kurias buvo iki šiol kalbėta, neturėjo grįžtamųjų ryšių. Jie yra vadinami tiesioginio sklaidimo neuroniniai tinklai. Šie tinklai neturi atminties ir išėjimo reikšmės priklauso nuo įėjimo reikšmių ir svorinių reikšmių.

## 2.7 Klaidos sklaidimo atgal algoritmas

Jei neuroninis tinklas turi vieną neuronų sluoksniu, tai šiuo atveju mokymo algoritmas aiškus, nes išėjimo sluoksniu teisingi atsakymai yra žinomi. Taigi šiuo atveju tinklo svorių reguliavimas vykdomas minimizuojant klaidą išėjimo sluoksnyje (2.5). Būtent šis principas yra naudojamas vienasluoksniu perceptrono mokyme. Kai turime daugiasluoksniu neuroninį tinklą, žinomos paskutinio sluoksniu išėjimo reikšmės, bet mums nežinomos kitų sluoksniu išėjimo reikšmės. Todėl šio metodo taikyti tiesiogiai yra neįmanoma.

Vienas iš sprendimo būdų taikyti klaidos sklaidimo atgal algoritmą. Algoritmo pavadinimas kilęs nuo to, kad jis „paskirsto“ klaidą tarp neuroninio tinklo elementų pradedant nuo išėjimo sluoksniu tolyn prie įėjimo sluoksniu (žr. 2.10 pav.).



2.10 pav. Daugiasluoksnis DNT klaidos skleidimas atgal.

Taigi, klaidos skleidimo atgal algoritmą sudaro du žingsniai:

1. įėjimų reikšmių „skleidimas pirmyn“ iš įėjimų į išėjimų sluoksnį;
2. paklaidos „skleidimas atgal“ iš išėjimų į įėjimų sluoksnį.

Šis algoritmas naudoja mokymo su mokytoju strategija. Klaida skaičiuojama pagal (2.5) formulę. Svoriai keičiami pagal (2.6) formulę:

$$\Delta w_{jk}(t) = -\eta \frac{\partial E(t)}{\partial w_{jk}}, \quad (2.6)$$

čia  $\eta$  – teigiamas daugiklis, jis dar yra vadinamas *mokymo greičiu* (angl. *learning rate*),  $t$  – iteracijos numeris. Pirmame žingsnyje apskaičiuojamas išėjimų vektorius  $Y_i$  ir įvertinama paklaidos funkcija  $E_i(W)$ , išėjimų sluoksnyje  $L$  („skleidimo pirmyn“ fazė). Jei paklaidos funkcija  $E_i(W) \neq 0$ , reikia keisti jungčių svorius  $w_{jk}$ , kurie jungia  $k$ -tąjį neuroną  $(l-1)$ -ajame sluoksnyje su  $j$ -uoju neuronu  $l$ -ajame sluoksnyje. Keitimui naudojame (2.7) formulę:

$$\Delta w_{jk}^i = -\eta \frac{\partial E_i}{\partial w_{jk}}. \quad (2.7)$$

Dalinės išvestinės išreiškiamos pagal formulę:

$$\frac{\partial E_i}{\partial w_{jk}} = \frac{\partial E_i}{\partial out_{ij}} \cdot \frac{\partial out_{ij}}{\partial w_{jk}}. \quad (2.8)$$

$$\frac{\partial out_{ij}}{\partial w_{jk}} = y_{ik}. \quad (2.9)$$

Tegul

$$\delta_{ij} = \frac{\partial E_i}{\partial out_{ij}}. \quad (2.10)$$

Istatę (2.10) ir (2.9) į (2.8) ir (2.7) formules, gauname:

$$\begin{aligned}\frac{\partial E_i}{\partial w_{jk}} &= \delta_{ij} y_{ik}, \\ \Delta w_{jk}^i &= -\eta \delta_{ij} y_{ik}.\end{aligned}\quad (2.11)$$

čia  $j$ -asis neuronas priklauso  $l$ -ajam sluoksniui,  $k$ -asis neuronas priklauso  $(l-1)$ -ajam sluoksniui.

Išėjimų sluoksnyje

$$\delta_{ij} = \frac{\partial E_i}{\partial out_{ij}} = f'(out_{ij})(y_{ij} - t_{ij}) \quad (2.12)$$

čia  $j$ -asis neuronas priklauso išėjimų sluoksniui  $L$ .

Rasime  $\frac{\partial E_i}{\partial out_{ij}}$  paslėptiems neuronams. Naudojantis dalinėmis išvestinėmis, gauname:

$$\delta_{ij} = \frac{\partial E_i}{\partial out_{ij}} = \sum_{s=1}^{n_{l+1}} \frac{\partial E_i}{\partial out_{is}} \cdot \frac{\partial out_{is}}{\partial out_{ij}},$$

čia  $n_{l+1}$  žymi neuronų  $(l+1)$ -ajame sluoksnyje skaičių. Išraiška  $\frac{\partial E_i}{\partial out_{is}}$  yra lygi dydžiui  $\delta_{is}$ , apibrėžtam  $s$ -ajam neuronui  $(l+1)$ -ajame sluoksnyje.

Toliau gauname

$$\frac{\partial out_{is}}{\partial out_{ij}} = f'(out_{ij}) w_{is}.$$

Tada paslėptųjų  $j$ -ųjų neuronų

$$\partial_{ij} = f'(out_{ij}) \sum_{s=1}^{n_{l+1}} w_{is} \delta_{is} \quad (2.13)$$

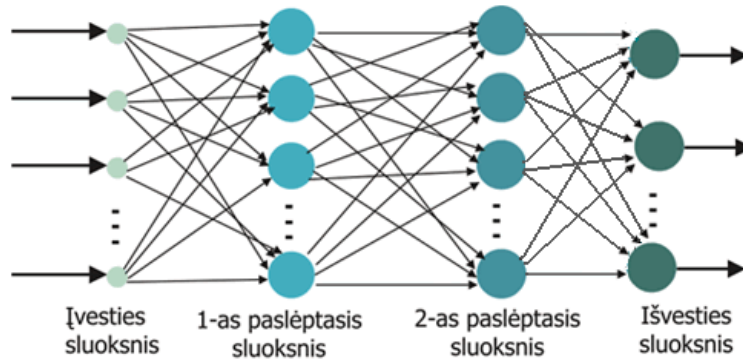
čia  $j$ -asis neuronas priklauso sluoksniui  $l < L$ ,  $s$ -asis neuronas priklauso sluoksniui  $l+1$ .

Iš pradžių apskaičiuojamas  $\delta_{ij}$  reikšmės išėjimų sluoksnyje  $L$  pagal (2.12) formulę. Tada pagal formulę (2.13) palaipsniui skaičiuojamos  $\delta_{ij}$  reikšmės tarpiniuose sluoksniuose. Visi svoriai keičiami pagal formulę (2.11). Tada į tinklą paduodamas naujas vektorius  $X_i$  ir mokymo algoritmas kartojamas iš naujo [15].

## ***2.8 Dirbtiniai neuroniniai tinklai rašto ženklams atpažinti***

### **2.8.1 Daugiasluoksnis perceptronas**

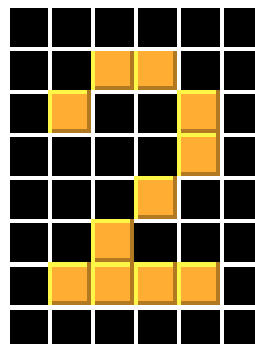
Daugiasluoksnius perceptronas (žr. 2.1 pav.), mokomas klaidos skleidimo atgal algoritmu (žr. 2.7 d.), gali būti taikomas rašto ženklams atpažinti.



2.11 pav. Daugiasluoksnis DNT.

Toliau bus paaiškinta, kaip iš ženklų gauti vektorius, kurie bus naudojami neuroniniams tinklams mokyti. Taip pat aptartas raidžių centravimas ir padidinimas.

Tarkime turime tinklą, sudalintą į  $a$  stulpelių ir  $b$  eilučių. Raidžių vektorius yra dvimatis masyvas  $X' = (x'_{11}, x'_{12}, \dots, x'_{ab})$ . Reikšmės  $x'_{ij}$  priklauso nuo tinklelio langelio „uždažymo“ ryškumo. Galimos reikšmės priklauso intervalui nuo 0 iki 1. Kuo ryškesnė spalva, tuo didesnę reikšmę įgyja  $x'_{ij}$ . 2.12 paveikslėlyje matome simbolį, o 1 lentelėje pavaizduota, kokias reikšmes įgyja atitinkamas langelis. Šiuo atveju langelis gali įgyti reikšmę 0 arba 1, priklausomai nuo to, ar jis yra „uždažytas“, ar ne.



2.12 pav. Simbolis (skaitmuo „2“).

0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	1	1	1	1	0
0	0	0	0	0	0

1 lentelė. Simbolio vektorius.

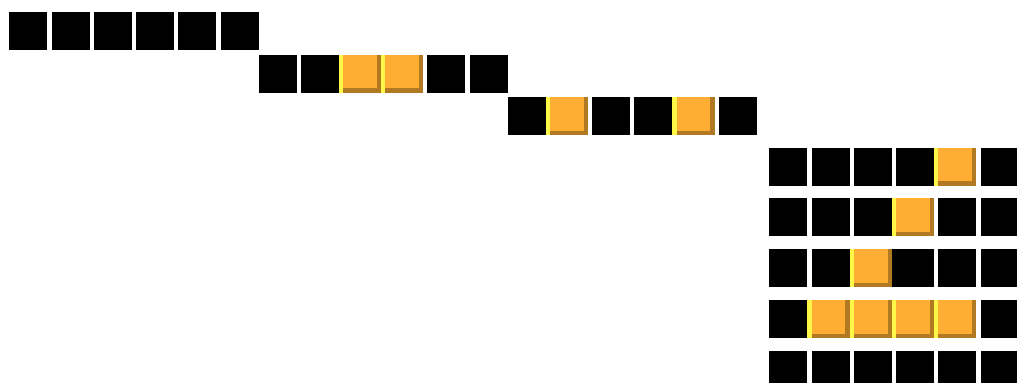
Turime matricą (t. y., dvimatį masyvą), iš kurio turime padaryti vienmatį masyvą, nes daugiasluoksnio perceptrono įėjimai turi būti vienmatis masyvus  $X = (x_1, x_2, \dots, x_n)$ . Žemiau parodytas funkcijos pseudo kodas, pagal kurį iš dvimačio masyvo gaunamas vienmatis.

```

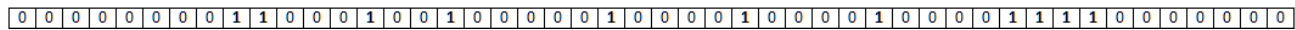
function vektorius( $X'$ )
// įėjimas:  $X'$  – raidės vektorius
// išėjimas:  $X$  – įėjimo vektorius
BEGIN
FOR  $i=1$  TO  $a$                                 // stulpelių skaičius
  FOR  $j=1$  TO  $b$                                 // eilučių skaičius
     $x[b(i-1)+j] = x'[i][j]$ 
  END
END
RETURN  $X$ 
END

```

Funkcija  $\text{vektorius}(X')$  įėjime gauna simbolio vektorių (žr. 2.12 pav.) (1 lentelė). Gautas dvimatis  $a \times b$  vektorius perrašomas eilutėmis (žr. 2.13 pav.). Išėjime gaunamas vienmatis vektorius, kurio ilgis yra  $ab$  (žr. 2.14 pav.)

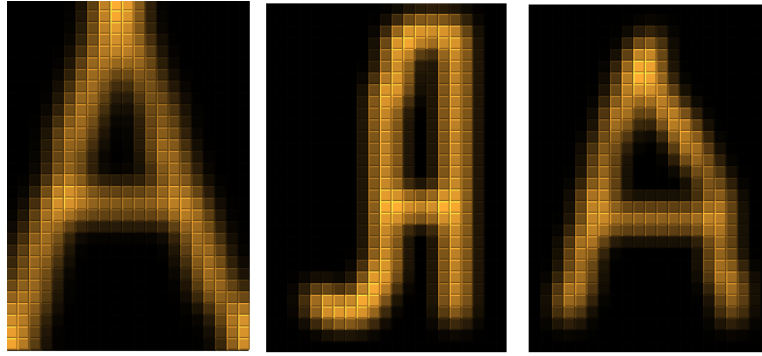


2.13 pav. Vektoriaus formavimas.



2.14 pav. Gautas vektorius (skaitmuo „2“).

Išvesties sluoksnis formuojamas priklausomai nuo pradinių duomenų, t. y., jei tinklas mokomas atpažinti  $q$  raidžių ir kiekvienos raidės yra po  $c$  variantų, tai  $Y = (y_1, y_2, \dots, y_{qc})$ . 2.15 paveiksle parodyti skirtingi raidės „A“ užrašymo variantai. Įėjimo vektorius  $X = (x_1, x_2, \dots, x_n)$  yra susietas su trokštamų reikšmių vektoriumi  $T = (t_1, t_2, \dots, t_d)$ , čia  $T$  – norimo išėjimo reakcija į  $X$ . Skaičiuojant paklaidą naudojama (2.5) formule.



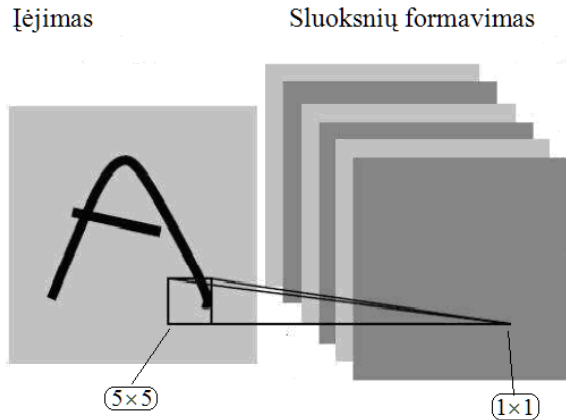
2.15 pav. Skirtingi tos pačios raidės užrašymo variantai.

### 2.8.2 Sąsūkinis neuroninis tinklas

Kitas dažnai pastaruoju metu naudojamas neuroninis tinklas yra sąsūkinis [10]. Sąsūkinio dirbtinio neuroninio tinklo (angl. *convolutional neural network*) pagrindą sudaro trys mechanizmai:

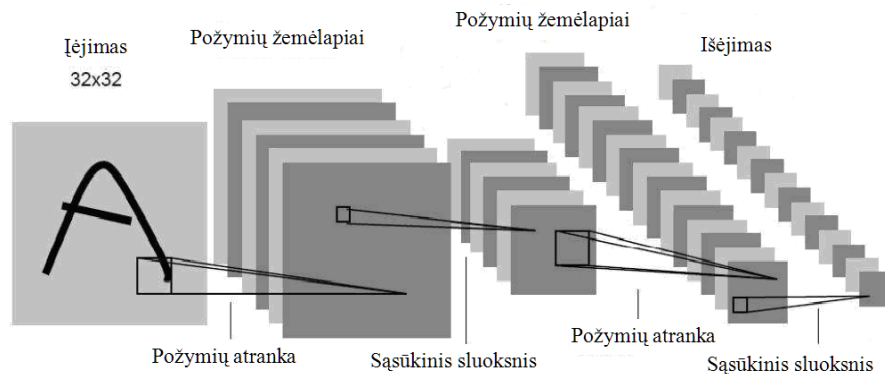
1. **Lokaliųjų požymių išskyrimas.** Kiekvienas neuronas gauna įėjimo signalą iš prieš tai buvusio sluoksnio lokalaus lauko. Tokiu būdu yra ištraukiami lokalieji požymiai (žr. 2.16 pav.). Kai tik požymis yra ištrauktas, jo buvimo vieta jau yra nesvarbi, nes jau žinoma jo vieta atžvelgiant į kitus požymius.
2. **Sluoksnių formavimas.** Kiekvienas sluoksnis turi kelių plokštumų rinkinius (žr. 2.16 pav.). Sluoksnyje vienos plokštumos neuronai turi tuos pačius svorių rinkinius, jie sujungti su visomis prieš tai einančio sluoksnio sritimis. Tokia forma daro tinklo struktūrą sudėtingą, bet turinčią daug privalumų.
3. **Požymių atranka.** Po kiekvieno sąsūkinio sluoksnio eina sluoksnis, kuris vykdo požymių atranką. Ši operacija leidžia sumažinti išeinančio signalo jautrumą rakurso keitimui ir nedideliems iškraipymams.





2.16 pav. Sąsūkinio DNT sluoksnių formavimas.

Naudojant šiuos tris mechanizmus pasiekiamas sumažintas rakurso ir mastelio jautrumas keitimui, bei nedidelių iškrypimų dalinis pastovumas [10].



2.17 pav. Sąsūkinis DNT.

Į įėjimo sluoksnį paduodamas centruotas raidės vektorius (žr. 2.17 pav.). Nagrinėjamo pavyzdžio raidės vektoriai yra  $21 \times 30$  (pikselių). Jie centruojami paveiksle, sudarytame iš  $32 \times 32$  pikselių.

Algoritmo schema:

1. Pirmas paslėptas sluoksnis yra sąsūkinis. Jis sudarytas iš 6 požymių žemėlapių, kurie yra  $28 \times 28$  pikselių. Pažiūrėkim atidžiau, kaip yra formuojamas šis sąsūkinis sluoksnis.

Kiekvienas požymio žemėlapiio elementas yra sujungtas su įėjimo sluoksnio dalimi  $5 \times 5$ . Iš čia seka, kad kiekvienas elementas turi 25 apmokomus koeficientus, bei apmokomą poslinkį. Požymio žemėlapiio elementai skaičiuojami pagal formulę:

$$X_i^{h,l} = f \left( \sum_{k=1}^{n_l-1} \sum_{j=-\infty}^{\infty} X_{i-j}^{k,l} W_{i-j,i}^{h,k,l} + B_i^{h,l} \right), \quad (2.14)$$

čia  $X_i^{h,l}$  elemento  $i$  reikšmė, esanti požymių žemėlapyje  $h$ , sluoksnyje  $l$ .  $n_l$  – požymių žemėlapių kiekis sluoksnyje  $l$ .  $B_i^{h,l}$  – poslinkio  $i$  reikšmė, esanti požymių žemėlapyje  $h$  sluoksnyje  $l$ .  $W_{i-j,i}^{h,k,l}$  – svorio koeficientas tarp elemento  $i$  požymių žemėlapyje  $h$ , sluoksnyje  $l$  ir elemento  $i-j$  žemėlapyje  $k$ , sluoksnyje  $l-1$ . Visi kiti sąsūkiniai sluoksniai yra formuojami pagal tą patį principą. Šis sluoksnis turi 122304 ryšių ir 156 apmokomus parametrus. Taupydami kompiuterio atmintį ir skaičiavimo kaštus, vienam požymio žemėlapiui bendrai naudojami svoriai  $W$ .

2. Antras paslėptas sluoksnis – požymių atrankos. Jį sudaro 6 žemėlapiai 14x14. Kiekvienas iš šio sluoksnio elementų yra sujungtas su prieš tai buvusių sąsūkinių sluoksnių (pirmuoju paslėptu sluoksniu) dalimi 2x2. Požymių atrankos sluoksnio elementai taip pat skaičiuojami pagal (2.14) formulę. Atkreipkite dėmesį, kad skaičiuojant šio sluoksnio laukus, daromas žingsnis lygus 2. Todėl laukai nepersidengia ir gauname 2 kartus mažiau eilučių ir stulpelių, nei prieš tai buvusiam sluoksnyje. Šio sluoksnio užduotis yra požymių atranka, todėl galime naudoti vieną svorių rinkinį ir postūmio koeficientą [11]. Šis sluoksnis turi 5880 ryšių ir 12 apmokomų parametrų.

Toliau viską vykdome pagal prieš tai buvusį algoritmą (žingsniai kartojasi). Sąsūkinis sluoksnis, o po jo eina požymių atrankos. Būtina atkreipti dėmesį, kad kiekvienas tyrėjas, pats nusprendžia kaip bus jungiami antro ir trečio sluoksnio požymių žemėlapiai.

Kaip ir klasikiniuose neuroniniuose tinkluose, kiekvienas perceptronas įėjime gauna skaliarinės sandaugos tarp įėjimo ir svorių vektorių sumą. Ši skaliarinė sandauga toliau paduodama į aktyvavimo funkciją (1.2) arba (1.3).

Paskutinis yra išėjimo sluoksnis. Šio sluoksnio matmenys priklauso nuo sprendžiamo uždavinio. Pavyzdžiui, analizuojant lotynų abėcėlę, kurioje yra 23 raidės, tinklo išėjimų bus 23.

## **2.9 Atpažinimo rezultatų vertinimas**

Sprendžiamas atpažinimo uždavinys yra klasifikavimo uždavinys. Klasifikavimo algoritmai dažniausiai vertinami pagal klasifikavimo tikslumą. Klasifikavimo tikslumas vertinamas naudojant kitą duomenų aibę, vadinamą testavimo aibe, kuri nebuvo naudojama klasifikatoriaus mokyme. Vertinant klasifikavimo tikslumą dažniausiai remiamasi tuo faktu, kad elementai yra tinkamose klasėse. Bet visiškai nėra atitvėlgama į tai, kiek elementų priskirti netinkamai klasei.

Tarkime turime dvi aibes: duomenų aibę  $X = (X_1, X_2, \dots, X_k)$  ir klasių aibę  $C = (C_1, C_2, C_3, C_4)$ . Galimi du atvejai, kai  $X_j$  priskiriamas tinkamai klasei arba kai priskiriamas netinkamai klasei. Pats objektas  $X_j$  gali priklausyti klasei (ji vadinsime teigiamu objektu) arba nepriklausyti jai (vadinsime neigiamu). Klasifikuojant objektas priskiriamas klasei, tas priskyrimas gali būti teisingas (teigiamas) arba neteisingas (neigiamas).

Apibrėžkime pagrindines sąvokas:

*Tikrai teigiamas* (TT, angl. *true positive TP*), jei objektas  $X_j$  priskirtas klasei, kuriai jis priklauso.

*Klaidingai teigiamas* (KT, angl. *false positive FP*), jei objektas  $X_j$  priskirtas klasei, bet jis jai nepriklauso.

*Tikrai neigiamas* (TN, angl. *true negative TN*), jei objektas  $X_j$  nepriskirtas klasei ir jis jai nepriklauso.

*Klaidingai neigiamas* (KN, angl. *false negative FN*), jei objektas  $X_j$  nepriskirtas klasei, bet jis jai priklauso.

		Klasės	
		<i>Priklauso</i>	<i>Nepriklauso</i>
Klasifikuota testuojant	<i>Teisingai</i>	<b>Tikrai Teigiamas (True Positive) (TT) = 20</b>	<b>Klaidingai Teigiamas (False Positive) (KT) = 180</b>
	<i>Neteisingai</i>	<b>Klaidingai Neigiamas (False Negative) (KN) = 10</b>	<b>Teisingai Neigiamas (True Negative) (TN) = 1820</b>
		Jautrumas (Sensitivity) = $TT / (TT + KN)$ = $20 / (20 + 10)$ = $20 / 30$ = <b>66.67%</b>	Specifiškumas (Specificity) = $TN / (KT + TN)$ = $1820 / (180 + 1820)$ = $1820 / 2000$ = <b>91%</b>

2.18 pav. Klasifikavimo tikslumo įvertinimas.

Tarkime, turime grupę objektų  $X = (X_1, X_2, \dots, X_k)$ . Atliekamas tyrimas klasifikavimo tikslumo įvertinimui (žr. 2.18 pav.). Keli objektai  $Z = (Z_1, Z_2, \dots, Z_m)$ , kur  $m < k$ ,  $Y \subset X$  tikrai teigiami (TT). Kita dalis objektų nebuvo priskirta šiai grupei, bet priklauso jai. Vadinsime klaidingai neigiamais (KN). Dar dalis aibės  $X = (X_1, X_2, \dots, X_k)$  objektų buvo klasifikuoti kaip tikrai neigiami, t. y., nepriklausantys aibei  $Z = (Z_1, Z_2, \dots, Z_m)$  ir jie išties nepriklauso jai (TN). Paskutinė dalis objektų buvo klasifikuoti kaip klaidingai teigiami, t. y., priskirti aibei  $Z = (Z_1, Z_2, \dots, Z_m)$ , bet jie nepriklauso šiai aibei (KT).

Jautrumo matas (2.15) (angl. *sensitivity*) parodo santykį tarp objektų, tikrai turinčius mus dominančią savybę, kuri yra patvirtinta su visa teigiamai klasifikuota objektų aibe. Kuo didesnis jautrumo matas, tuo mažesnė tikimybė, kad objektas neturintis mums dominančių savybių bus klasifikuotas kaip teigiamas.

$$Jautrumas = \frac{TT}{TT + KN} \quad (2.15)$$

Specifiškumo matas (2.16) (angl. *specificity*) parodo santykį tarp objektų tikrai neturinčių mums dominančių savybių su visa neigiamai klasifikuota objektų aibe. Kuo didesnė specifiškumo reikšmė, tuo mažesnis skaičius, tikrai teigiamų objektų, bus blogai klasifikuotas.

$$Specifiškumas = \frac{TN}{KT + TN} \quad (2.16)$$

Taip pat tinklo apsimokymo kokybė gali būti vertinama ir pagal formulę:

$$E(W) = \frac{1}{2} \sum_{i=1}^d (y_i - t_i)^2,$$

čia  $y_i$  yra tinklo išėjimai, o  $t_i$  – trokšamos reikšmės.

## 3. Rašto ženklų atpažinimo sistemos

### 3.1 Sistemų analizė

Apžvelgiant kitus mokslinius darbus šia tema, buvo išskirti dviejų autorių darbai – rašto ženklų atpažinimo algoritmų, pagrįstų neuroniniais tinklais, programinės realizacijos. Vieną darbą atliko J. Tiscione. Programa realizuota Java applet aplinkoje [12]. Grafinis vartotojo sąsajos vaizdas pateiktas 3.1 paveiksle.

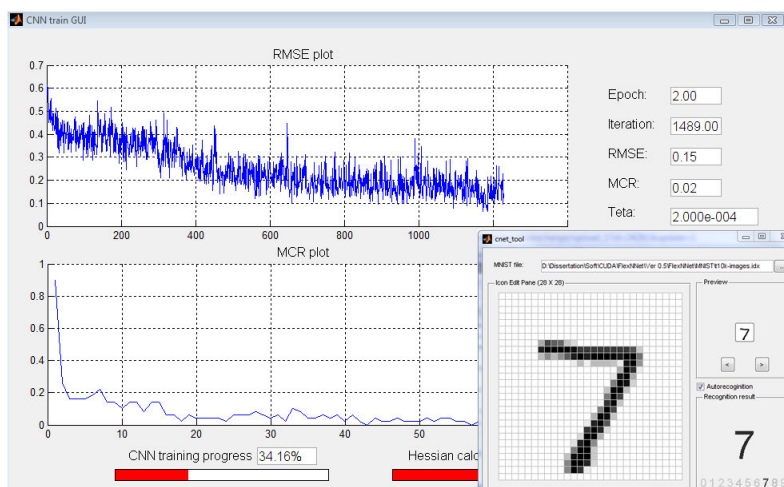


3.1 pav. Java applet programos grafinė vartotojo sąsaja.

3.1 paveiksle matome dešimt simbolių (skaitmenų), kuriais bus apmokomas neuroninis tinklas. Įėjimo vektorius sudarytas iš 12 stulpelių ir 16 eilučių, tačiau jis paverstas į vienmatį masyvą (apie tai skaitykite 2.4.1 skyrelyje). Mokymo duomenys sudaro langeliuose užrašyti skaičiai. Programoje realizuotas daugiasluoksnio perceptrono modelis, kuris yra mokomas naudojant klaidos skleidimo atgal algoritmą. Gali būti keičiamas pirmo paslėpto neuronų sluoksnio kiekis (inputs neurons) ir antro paslėpto neuronų sluoksnio kiekis (hidden neurons). Tinklo mokymas prasideda paspaudus mygtuką „Start training“. Mokymo metu keičiasi epochų skaičius (training epochs) ir kvadratinė paklaidų suma (sum squared error). Mokymo epocha – tai mokymo proceso dalis, kurios metu visi įėjimo vektoriai pateikiami į tinklą vieną kartą. Pasiekus tenkinančią paklaidą, mokymas stabdomas, paspaudus mygtuką „Stop training“. Spaudžiant virš skaičių, kuriais buvo mokomas neuroninis tinklas, mygtukus „test0 test1 ... test9“ tikrinama, kaip tinklas yra išmokytas. Jei skaitmuo yra gerai atpažintas, po juo esantis langelis tampa ryškiai raudonas, o langeliai, esantys po kitais skaičiais, turi likti juodos spalvos. Po mokymo įvedimo langelyje įrašius

simbolį, kuriuo bus testuojamas išmokytas tinklas, ir paspaudus „test“ mygtuką, vienas iš langelių, esančių po viršuje „nupaišytais“ skaitmenimis tampa raudonu. Tai reiškia, kad tinklas atpažino įvestą skaitmenį, kaip tą, kuris yra virš raudono langelio. Jei nuspalvinami keli langeliai, tai reiškia kad tinklas negali tiksliai atpažinti įvesto skaičiaus (žr. 3.1 pav.).

Darbe taip pat nagrinėjamas M. Sirotenko darbas [13]. Programa realizuota MatLab aplinkoje (žr. 3.2 pav.). Joje realizuotas sąsūkinis neuroninis tinklas. Apmokymui ir testavimui naudojama rašytinių simbolių duomenų bazė MNIST [14]. Skaičiai mokymui ir testavimui surašyti dvejuose tekstiniuose failuose. Taip pat į du tekstinius failus surašomi ir teisingi atsakymai. Teisingi atsakymai, t. y., trokštamos reikšmės, jei paduodamas simbolis yra 7 (žr. 2.2 pav.).

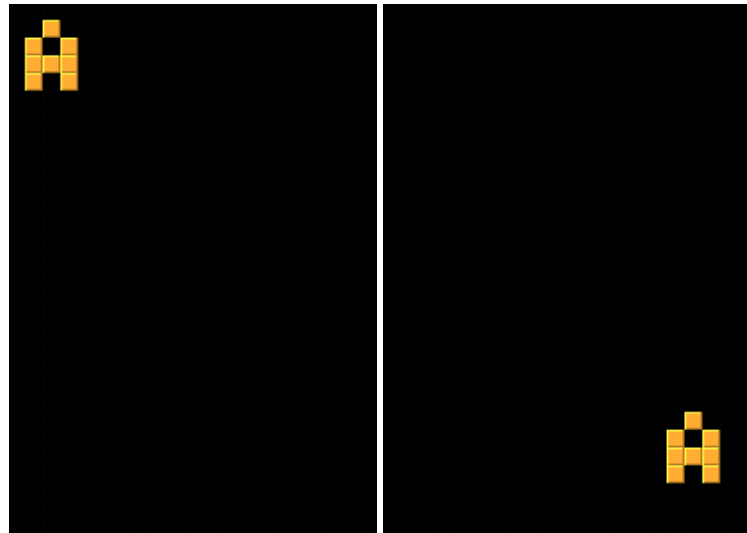


2.2 pav. MatLab programos grafinė vartotojo sąsaja.

### 3.2 Sistemų patobulinimas

Realizuojant dirbtinį neuroninį tinklą buvo pasirinktas daugiasluoksnio perceptrono modelis, bei klaidos skleidimo atgal algoritmas (žr. 2.7 d.). Raidžių vektoriai surašyti tekstiniuose failuose. Pirmas failas skirtas tinklo apmokymui, antras testavimui.

Pirma problema, su kuria buvo susidurta, tai raidės užrašymo galimybė. Buvo pastebėta, kad egzistuoja vienos tos pačios raidės daug užrašymo variantų (žr. 3.3 pav.).



a)

b)

3.3 pav. Raidės A užrašymo variantai.

Kaip matome 3.3 paveiksle egzistuoja daug galimų užrašymo vietų, ir jei žmonių rašto suvienodinti negalime, galime pareikalauti, kad raidės būtų paveikslo centre ir kaip įmanomą didesnės, t. y., normuotos.

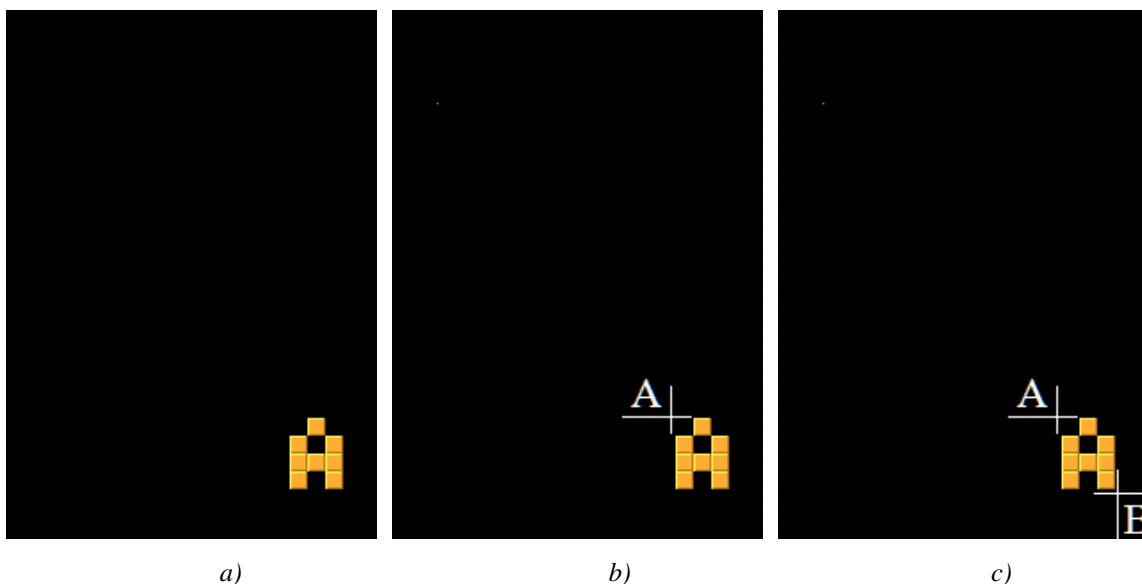
Taigi nesvarbu kur bus parašyta raidė (žr. 3.3 pav.), atlikus raidės centravimą ir padidinimą ji turi būti paveikslo centre ir kaip įmanoma ištempta į visas puses. Tam, kad atlikti raidės centravimą ir padidinimą, buvo sukurtos dvi funkcijos. Pirmos funkcijos pseudo kodas pateiktas žemiau:

```

function taskai(X)
// įėjimas: X – raidės vektorius
// išėjimas: du taškai A(x1, y1), B(x2, y2)
BEGIN
A(0,0)
B(0,0)
FOR i=1 TO 30                // eilučių skaičius
  FOR j=1 TO 21                // stulpelių skaičius
    If(X(i, j)≠0) then A(i, j)    // randamas pirmas stulpelis ir pirmą eilutę, kurioje užrašytas simbolis
    If(A(i, j)≠0 & X(i, j)=0) then B(i, j) // randamas paskutinis stulpelis ir paskutinę eilutę, kurioje yra simbolis
  END
END
RETURN (A,B)
END

```

Pažiūrėkime atidžiau, ką atlieka ši funkcija **taskai**(X). Tarkime, kurioje nors ekrano vietoje užrašyta raidė (žr. 3.4 a pav.). Pirmiausiai surandamas taškas A (žr. 3.4 b pav.), kuris yra pirmos eilutės ir stulpelio, kuriame randamas simbolis, pradžios koordinatės. Tada ieškomas taškas B (žr. 3.4 c pav.), kuris yra paskutinės eilutės ir paskutinio stulpelio, kuriame yra simbolis susikirtimo koordinatės.



3.4 pav. Raidės A centravimas.

Gauti du taškai perduodami funkcijai „centras“. Žemiau matome jos pseudo kodą:

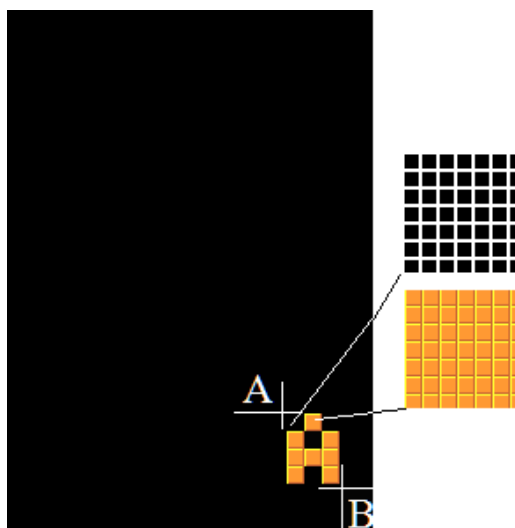
```

function centras (A,B )
// įėjimas: du taškai A( x1, y1 ), B( x2, y2 )
// išėjimas: X – raidės vektorius
BEGIN
C = Min(  $\frac{21}{x_2 - x_1}, \frac{30}{y_2 - y_1}$  ) // randame kiek kartų galime padidinti simbolį
FOR i=1 TO 30 // stulpelių skaičius
  FOR j=1 TO 21 // eilučių skaičius
    X'(i, j)=X(x1 + i mod C, y1 + j mod C) // perrašom raidės vektorius
  END
END
X = X'
RETURN X
END

```

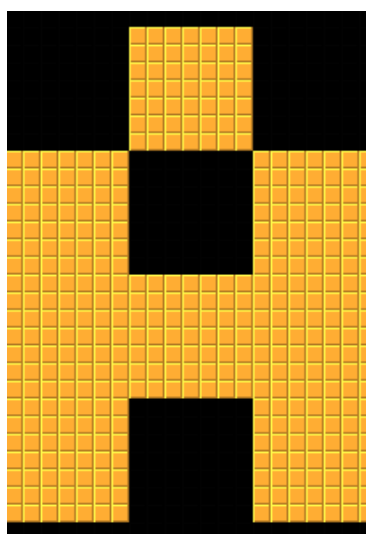
Ši funkcija centruoja ir padidina raidę, t. y., raidės vektorius yra „perrašomas“. Sakykime, kad turime raidę, kuri užrašyta trejuose stulpeliuose ir keturiose eilutėse (žr. 3.4 a pav.). Reikia šią raidę perrašyti taip, kad ji būtų per visą turimą langelį (žr. 3.6 pav.). Vektorius paduodamas į funkciją, kuriai nurodomos raidės pradžios ir pabaigos koordinatės. Randamas skaičius (koeficientas) C, kurio reikšmė nurodo, kiek kartų galime padidinti raidę. 3.4 paveiksle pateiktu atveju raidė užima 3 stulpelius ir 4 eilutes, o turimas langas yra sudarytas iš 21 stulpelio ir 30 eilučių. Taigi, pagal stulpelius galime padidinti 7 kartus, o pagal eilutes 7,5 karto. Pasirenkamas mažesnis iš šių skaičių, jei skaičius nėra sveikasis, visi skaičiai po kablelio atmetami.





3.5 pav. Raidės A perrašymas.

Perrašant raidę kiekvienas langelis, kuris yra  $1 \times 1$ , padalijamas į langelį  $7 \times 7$  (žr. 3.5 pav.), tokiu būdu gaunamas naujas vektorius, kurio matmenys jau yra  $21 \times 28$ . Kadangi reikia, kad eilučių skaičius būtų lygus 30, pridedam dar po vieną eilutę iš viršaus ir apačios. Gautoji raidė yra lango centre ir ištempta į visas puses (žr. 3.6 pav.).



3.6 pav. Perrašyta A raidė.

### 3.3 Sistemos papildymas modulių, skirtu eksperimentams atlikti

Mokant dirbtinį neuroninį tinklą labai svarbus pradinis duomenų parinkimas, t. y., svorių intervalas, iš kurio generuojamas pradinės svorių reikšmės, neuronų kiekis pirmame ir antrame sluoksnyje ir mokymo aibės dydis. Tam, kad galima būtų atlikti eksperimentus, buvo sukurtos kelios funkcijos, kurios leidžia programoje nustatyti pirmo ir antro neuronų sluoksnio svorių pradines reikšmes (3.7 pav. Start), žingsnį, kuriuo bus didinamas neuronų tinklas

(3.7 pav. *Žingsnis*), stabdymo sąlygą, t. y., kokį neuronų skaičių pasiekus tinklas nebus didinamas (3.7 pav. *Stop*), paklaidą, iki kurios bus mokomas tinklas, mokymo epochų skaičių ir mokymo aibės dydį (žr. 3.7 pav.).

Kintamasis	Reikšmė	Start	Žingsnis	Stop
Input neurons:	10	5	5	30
Hidden neurons:	30	5	5	50
Training epochs <	10000			
Kol error >	0.001			
Kiek raidž. naud. apm. ?	30			

3.7 pav. Programos fragmentas.

Šios programos fragmento pseudo kodas pateiktas žemiau:

```

function tyrimas ()
BEGIN
S1.....// pirmo sluoksnio neuronų skaičius „Start“
S2.....// antro sluoksnio neuronų skaičius „Start“
S1_žingsnis.....// pirmo sluoksnio žingsnis
S2_žingsnis      // antro sluoksnio žingsnis
S1_stop.....// pirmo sluoksnio neuronų skaičius „Stop“
S2_stop         // antro sluoksnio neuronų skaičius „Stop“

FOR S2 TO S2_stop           // čia žingsnis lygus S2_žingsnis
  FOR S1 TO S1_stop       // čia žingsnis lygus S1_žingsnis
    Mokymas (S1, S2, X)
    Print (rezultatai)     // spausdinamas laikas, paklaida, klasifikavimo matai
  END
END
END

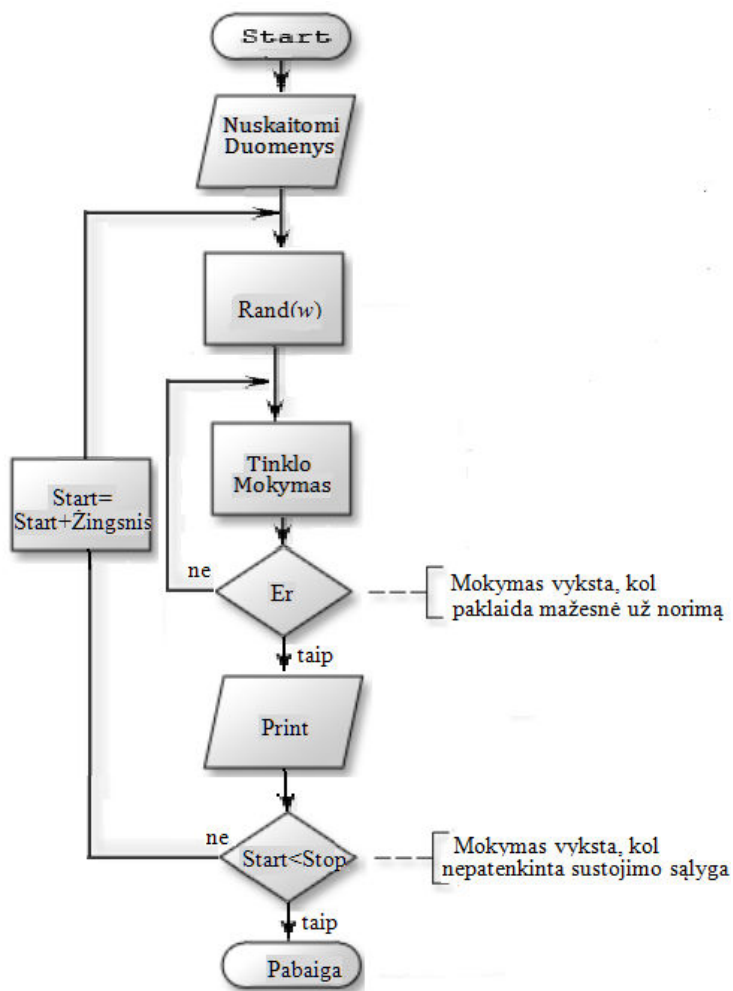
```

Funkcija „tyrimas“ nuskaityto įvestus pradinis duomenys. Tada dvigubame cikle vyksta mokymas. Mokymas vykdomas tol, kol nebus pasiektas programoje nustatytas paklaidos slenkstis arba mokymo epochų skaičius. Tada išvedami gauti rezultatai, mokymo laikas, paklaida, klasifikavimo matai: jautrumas ir specifiškumas. Toliau mokymas vykdomas esant kitiems neuronų kiekiams.

Ši funkcija naudojama norint iširti nedidelę duomenų aibę (pvz., apie 30 raidžių) ir nustatyti pradinis duomenis (neuronų kiekiai, svorių generavimo intervalas), geriausiai tinkančius sprendžiamam uždaviniui.

Svorių generavimo intervalas nustatomas pačioje programoje. Tame intervale generuojamos atsitiktinės svorių reikšmės.

Algoritmo blokinė schema pateikta 3.8 paveiksle.

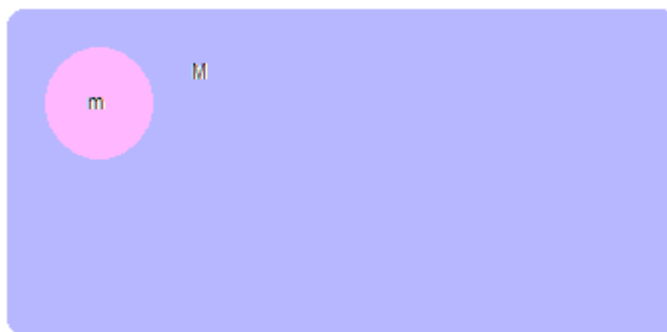


3.8 pav. Modulio skirto eksperimentiniams tyrimams blokinė schema.

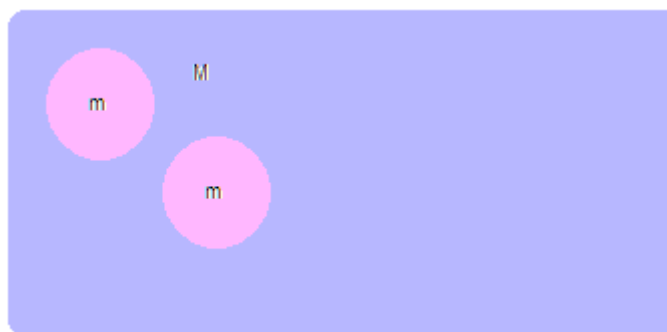
### 3.4 Mokymo dalimis strategija

Vykdamas mokymą naudojant didesnę duomenų aibę (pvz., apie 140 raidžių), klasikinis daugiasluoksnių perceptrono modelis (apie tai skaitykite 2.4.1 skyrelyje) turi keletą trūkumų (tinklo paralyžius, lokalieji minimumai ir kt.) [16]. Tam, kad išspręsti šią problemą darbe buvo pasiūlytas algoritmo patobulinimas. Patobulinimo esmė – mokyti tinklą naudojant ne visą duomenų aibę, kurios elementų skaičius  $M$ , o tik tos aibės nedidelį poaibį, kurios elementų skaičius  $m$  (žr. 3.9 pav.). Algoritmo blokinė schema pateikta 3.11 paveiksle. Mokymas stabdomas dvejais atvejais: kai pasiekta mus tenkinanti paklaida arba tam tikras epochų skaičius. Paklaidos slenkstis nustatomas gana didelis, neuroninis tinklas turi mokėti atpažinti apie 80 % tikslumu. Mokymo epochos ribojimai nustatyti tam, kad nesėkmės atveju (užstrigimas lokaliame minimume) galėtume apriboti mokymo laiką. Kai dirbtinis neuroninis tinklas išmoksta atpažinti to poaibio raides su mus tenkinančia paklaida arba pasiekiamas ribinis epochų skaičius, poaibis papildomas, t. y., mokymo aibės elementų skaičius tampa lygus  $m+m$  (žr. 3.10 pav.). Šiuo atveju pridedame tiek pat raidžių, kiek turėjome iš pradžių. Tinklo mokymas vykdomas toliau su jau turimais (ankstesniame etape

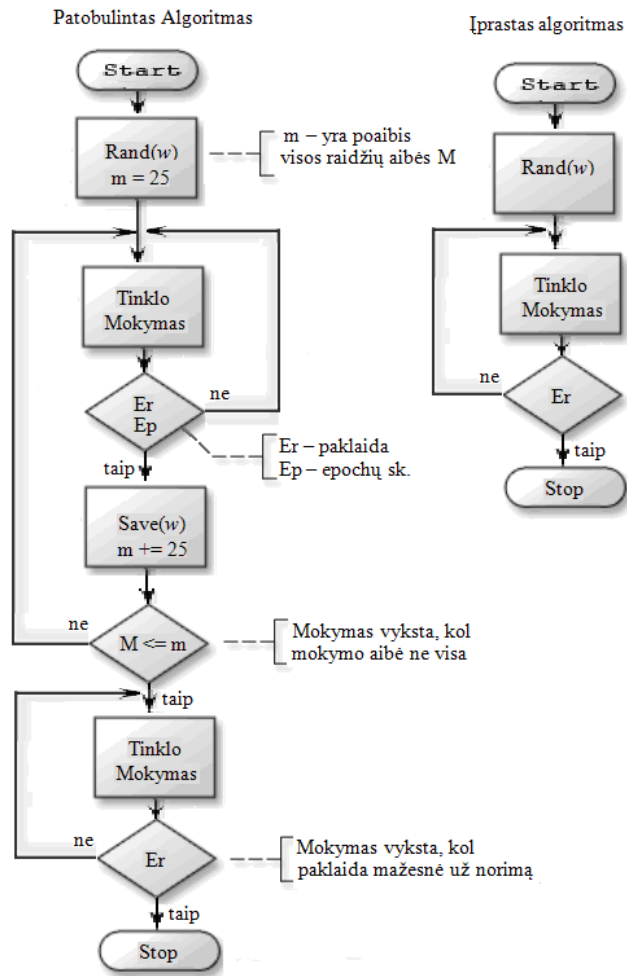
gautais) svorių vektoriais  $W$ . Tokiu būdu tinklas jau moka atpažinti pirmą poaibį ir jam lieka išmokti atpažinti antrą poaibį. Toliau mokymas vyksta tol, kol nebus pasiektas paklaidos slenkstis arba nustatytų epochų skaičius. Tada padidinamas mokomų raidžių poaibis ir tinklas mokomas toliau su turimais svoriais. Taip daroma tol, kol apmokomų raidžių poaibis nebus lygus visai aibei. Kai apmokomų raidžių poaibis lygus visai apmokomų raidžių aibei, tinklo mokymas vyksta pagal įprastą algoritmą, tik vietoj to, kad generuoti pradinių svorių rinkinį naudojamas turimas svorių rinkinys, kuris jau moka atpažinti  $M - m$  aibės dalį.



3.9 pav. Mokymo aibė ir jos poaibis,  $m < M$ .



3.10 pav. Mokymo aibės papildymas.



3.11 pav. Mokymo dalimis ir standartinio algoritmo blokinė schema.

Taip patobulintas neuroninio tinklo turi pranašumą: įvestas mokymo  $\lambda$  laiko „sąlyginis“ ribojimas, t. y., apribojamas poaibio mokymo laikas, įvedant mokymo epochų ribą. Tuo pačiu išsprendžiama ir kita dirbtinių tinklų problema – lokaliųjų minimumų išvengimas. Sakykime, kad mokomas tinklas pataikė į lokalų minimumą. Apribodami mokymo epochų skaičių, apribojame ir laiką iki tol, kol tinklas bus papildomas naujais duomenimis. Papildžius tinklą naujais raidžių vektoriais, ši problema dingsta ir mokymas sėkmingai tęsiamas toliau.

## 4. Eksperimentiniai tyrimai

### 4.1 Atpažinimo rezultatų priklausomybė nuo svorių pradinių reikšmių parinkimo

Sprendžiant rašto ženklų atpažinimo uždavinį, labai svarbu pradinius svorių vektorius generuoti iš tinkamo intervalo. Nuo to priklauso tinklo mokymo trukmė, atpažinimo tikslumas. Taip pat svarbu nustatyti optimalų neuronų skaičių pirmame ir antrame neuronų sluoksniuose. Galimi optimalumo kriterijai: laikas, paklaida, klasifikavimo matai (jautrumas, specifiškumas). Tam kad parinkti tinkamą intervalą ir nustatyti optimalų neuronų skaičių pradžioje buvo atlikti tyrimai nedidelei raidžių aibei. Mokymui naudota 6 raidžių 5 variantai, t. y., aibė buvo sudaryta iš 30 raidžių. Testavimui naudota 6 raidžių 2 variantai, viso 12 raidžių. Tyrimuose buvo naudojami 30 kompiuterių, kurių pagrindiniai parametrai yra:

Intel (IR) Core(TM)2 CPU

4400 @ 2.00GHz

2.01 GHz 1.00 GB of RAM

Buvo pasirinkti tokie pradinių svorių vektorių  $W$  generavimo intervalai:  $(-1; 1)$ ,  $(-1; 0)$ ,  $(-0,7; 0,3)$ ,  $(-0,3; 0,7)$ ,  $(0; 1)$ . Neuronų skaičiaus pirmame ir antrame sluoksnyje nustatymui naudojamas 3.3 skyriuje aprašytas algoritmas. Pirmas neuronų sluoksnis pradėtas mokyti ir testuoti nuo penkių neuronų, keičiant jų skaičių žingsniu lygiu penkiems. Procesas baigiamas, kai pasiekiamas dvidešimties neuronų kiekis. Antras neuronų sluoksnis pradėtas mokyti ir testuoti nuo penkių neuronų, keičiant jų skaičių lygiu penkiems, baigiant pasiekus trisdešimt neuronų skaičių. Vertinimui naudojami šie matai: paklaida, laikas, jautrumas ir specifiškumas.

Geriausi rezultatai buvo pasiekti su svorių vektoriais, kurie buvo generuojami iš intervalų  $(-1; 1)$  (žr. 4.1 a lentelė) ir  $(-0,7; 0,3)$  (žr. 4.1 b lentelė). Lentelėse matome neuroninio tinklo įvertinimo matų reikšmes: jautrumas, specifiškumas, paklaida, mokymo ir testavimo laikas (sekundėmis), pirmo bei antro neuronų sluoksnio neuronų skaičiai  $n_1$  ir  $n_2$ .

Paklaida				
n2 \ n1	5	10	15	20
5	10.42	1.491773	1.74	1.395126
10	0.529084	0.000998	0.000996	0.000995
15	0.017703	0.000995	0.000976	0.000971
20	0.002908	0.000987	0.000988	0.000974
25	0.000999	0.000981	0.000975	0.00098
30	0.001054	0.000982	0.000982	0.00097

Paklaida				
n2 \ n1	5	10	15	20
5	11.11708	2.658004	1.866339	1.365048
10	0.344686	0.000998	0.000999	0.000997
15	0.10065	0.000997	0.000994	0.000989
20	0.62156	0.000996	0.000988	0.000983
25	0.001172	0.000995	0.000993	0.000974
30	0.178463	0.000996	0.000984	0.000979

Jautrumas				
n2 \ n1	5	10	15	20
5	0.86	0.96	0.93	0.97
10	0.94	0.97	0.99	0.98
15	0.95	0.99	0.97	0.99
20	0.94	0.99	0.99	0.99
25	0.95	0.99	0.99	1.00
30	0.94	0.99	0.99	1.00

Jautrumas				
n2 \ n1	5	10	15	20
5	0.79	0.96	0.96	0.96
10	0.94	0.97	0.97	0.98
15	0.96	0.98	0.98	0.99
20	0.94	0.98	0.97	0.99
25	0.97	0.98	0.99	0.99
30	0.94	0.97	0.99	0.98

Specifiškumas				
n2 \ n1	5	10	15	20
5	0.97	0.99	0.99	0.99
10	0.99	0.99	1.00	1.00
15	0.99	1.00	0.99	1.00
20	0.99	1.00	1.00	1.00
25	0.99	1.00	1.00	1.00
30	0.99	1.00	1.00	1.00

Specifiškumas				
n2 \ n1	5	10	15	20
5	0.97	0.99	0.99	0.99
10	0.99	0.99	0.99	1.00
15	0.99	1.00	1.00	1.00
20	0.99	1.00	0.99	1.00
25	0.99	1.00	1.00	1.00
30	0.99	0.99	1.00	1.00

Laikas				
n2 \ n1	5	10	15	20
5	479	646	62	1000
10	455	84	62	48
15	273	24	9	8
20	248	12	7	8
25	133	22	7	8
30	249	14	7	9

Laikas				
n2 \ n1	5	10	15	20
5	296	463	49	778
10	149	105	112	62
15	327	80	43	30
20	229	57	44	22
25	211	54	38	40
30	228	60	23	19

a)

b)

4.1 lentelė. Atpažinimo tikslumas ir mokymo laikas: a)  $W(-1; 1)$ , b)  $W(-0,7; 0,3)$ .

Naudojant šiuos du intervalus (4.1 lentelė), gauti panašūs mokymo rezultatai. Skiriasi tik neuroninio tinklo mokymo laikai. Visi klasifikavimo matai parodė beveik vienodus rezultatus. Santykis tarp objektų, tikrai turinčių mus dominančią savybę (kuri jau yra patvirtinta) su visa teigiamai klasifikuotų objektų aibe (jautrumas), kai neuronų kiekiai pirmame ir antrame sluoksnyje daugiau nei 5, vidutiniškai  $\sim 0,99$ . Kai kuriais atvejais jautrumas lygus ir vienetui (žr. 4.1 a lentelė Jautrumas). Kuo didesnis jautrumo matas, tuo mažesnė tikimybė, kad objektas neturintis mums dominančių savybių bus klasifikuotas kaip teigiamas. Santykis tarp objektų tikrai neturinčių mus dominančių savybių su visa neigiamai klasifikuota objektų aibe (specifiškumas), vidutiniškai  $\sim 0,999$ . Dažniausiai specifiškumo reikšmė lygi vienetui (24 kartus iš 30) (žr. 4.1 lentelė Specifiškumas). Kuo didesnė specifiškumo reikšmė, tuo mažesnis skaičius tikrai teigiamų objektų bus blogai klasifikuotas. Mažesnę mokymo laiką parodė intervalas  $W(-1; 1)$ .

Ne tokie tikslūs mokymo rezultatai gauti naudojant neuroninį tinklą, kai pradiniai svoriai buvo generuojami intervale  $(-0,3; 0,7)$  (žr. 4.2 pav.).

Paklaida				
n2 \ n1	5	10	15	20
5	7.66	8.835022	7.50	8.278408
10	0.359965	0.000999	0.000999	0.243983
15	0.002953	0.000995	0.000988	0.15476
20	0.156868	0.00099	2.53907	3.769609
25	0.07848	3.308373	5.462061	16.23077
30	6.280846	14.07709	13.92315	16.23077

Jautrumas				
n2 \ n1	5	10	15	20
5	0.85	0.87	0.87	0.81
10	0.94	0.98	0.99	0.96
15	0.96	0.96	0.97	0.97
20	0.94	0.97	0.91	0.90
25	0.94	0.88	0.78	0.54
30	0.79	0.62	0.63	0.58

Specifiškumas				
n2 \ n1	5	10	15	20
5	0.97	0.97	0.97	0.96
10	0.99	1.00	1.00	0.99
15	0.99	0.99	0.99	0.99
20	0.99	0.99	0.98	0.98
25	0.99	0.98	0.96	0.91
30	0.96	0.92	0.93	0.92

Laikas				
n2 \ n1	5	10	15	20
5	293	474	52	779
10	358	155	106	320
15	207	32	43	110
20	172	44	305	560
25	252	251	405	828
30	352	503	660	841

4.2 lentelė Optimalumo vertinimas  $W(-0,3;0,7)$ .

Intervale  $W(-0,3;0,7)$ , kai paslėptų neuronų kiekiai pirmame ir antrame neuronų sluoksnyje vidutiniai, t. y., 10 – 15 neuronų pirmame, 10 – 20 neuronų antrame sluoksnyje, tinklas parodė gana neblogus mokymo rezultatus (4.2 lentelė). Santykis tarp objektų, tikrai turinčių mus dominančią savybę (kuri jau yra patvirtinta) su visa teigiamai klasifikuotų objektų aibe (jautrumas), vidutiniškai  $\sim 0,974$  (žr. 4.2 lentelė Jautrumas). Santykis tarp objektų, tikrai neturinčių mus dominančios savybės su visa neigiamai klasifikuota objektų aibe (specifiškumas), vidutiniškai  $\sim 0,994$  (žr. 4.2 lentelė Specifiškumas). Laiko sąnaudos apytiksliai tokios pat kaip ir intervale  $W(-0,7;0,3)$ .

Blogiausiai apmokymo rezultatai buvo gauti intervaluose  $W(-1;0)$  (4.3 a lentelė) ir  $W(0;1)$  (4.3 b lentelė). Apmokymas visada buvo stabdomas priverstinai, atlikus 10 tūkstančių mokymo epochų.



Paklaida				
n2 \ n1	5	10	15	20
5	22.89	8.083278	8.71	6.056647
10	14.24298	3.876784	3.265942	0.510805
15	12.50969	0.708932	2.346116	0.101932
20	8.541695	0.024544	0.615042	0.937572
25	10.05572	4.309054	3.360015	0.14576
30	11.07319	3.771266	2.749228	3.613803

Paklaida				
n2 \ n1	5	10	15	20
5	16.78	16.97865	26.78	30
10	16.39781	26.23124	29.30785	30
15	26.15422	29.15385	30	30
20	30	30	30	30
25	30	30	30	30
30	30	30	30	30

Jautrumas				
n2 \ n1	5	10	15	20
5	0.38	0.88	0.84	0.90
10	0.65	0.88	0.90	0.96
15	0.66	0.93	0.92	0.94
20	0.81	0.95	0.90	0.94
25	0.66	0.85	0.88	0.94
30	0.69	0.85	0.87	0.83

Jautrumas				
n2 \ n1	5	10	15	20
5	0.65	0.692308	0.294872	0.166667
10	0.628205	0.275641	0.185897	0.166667
15	0.275641	0.192308	0.166667	0.166667
20	0.166667	0.166667	0.166667	0.166667
25	0.166667	0.166667	0.166667	0.166667
30	0.166667	0.166667	0.166667	0.166667

Specifiškumas				
n2 \ n1	5	10	15	20
5	0.88	0.98	0.97	0.98
10	0.93	0.98	0.98	0.99
15	0.93	0.99	0.98	0.99
20	0.96	0.99	0.98	0.99
25	0.93	0.97	0.98	0.99
30	0.94	0.97	0.97	0.97

Specifiškumas				
n2 \ n1	5	10	15	20
5	0.93	0.938462	0.86	0.833333
10	0.925641	0.855128	0.837179	0.833333
15	0.855128	0.838462	0.833333	0.833333
20	0.833333	0.833333	0.833333	0.833333
25	0.833333	0.833333	0.833333	0.833333
30	0.833333	0.833333	0.833333	0.833333

Laikas				
n2 \ n1	5	10	15	20
5	471	642	808	995
10	490	653	756	720
15	500	587	573	679
20	516	591	679	563
25	526	656	538	696
30	570	581	773	635

Laikas				
n2 \ n1	5	10	15	20
5	287	444	621	755
10	312	459	686	820
15	466	735	822	974
20	717	579	800	819
25	353	380	457	837
30	370	580	711	835

a)

b)

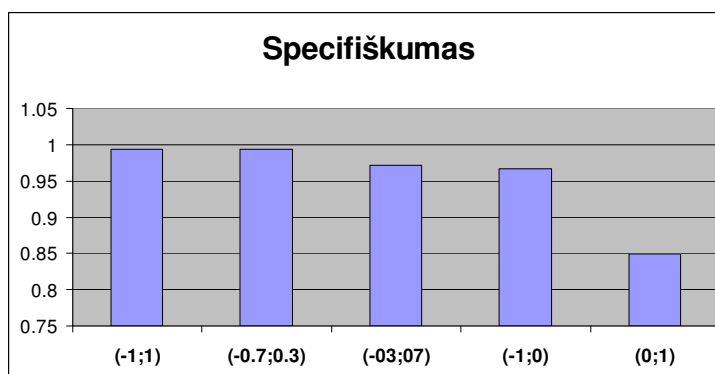
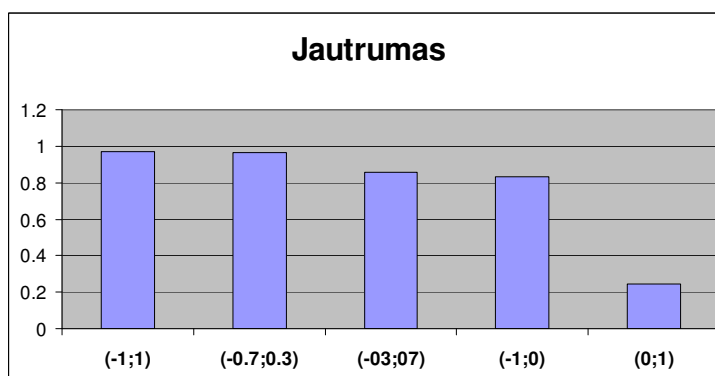
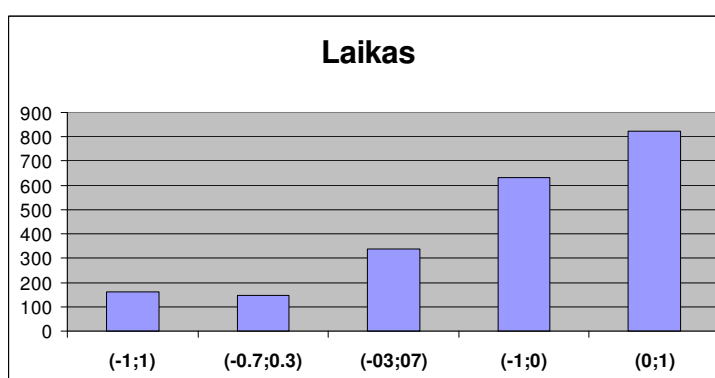
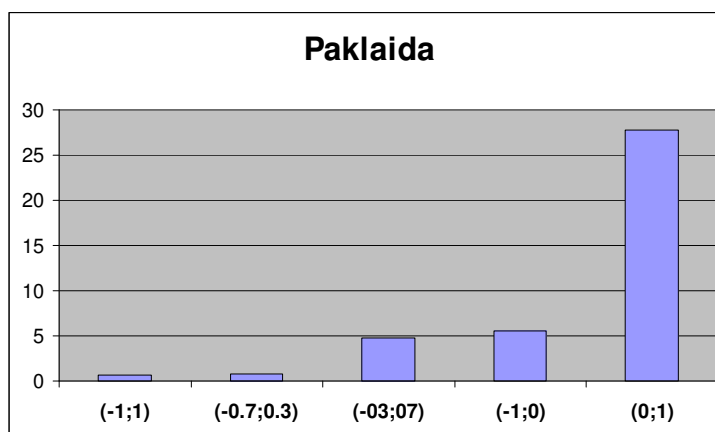
4.3 lentelė Optimalumo vertinimas a)  $W(-1;0)$ , b)  $W(0;1)$ .

4.3 a lentelė, kai pradinių svorių vektoriai generuojami tik iš neigiamo intervalo, nors neuroninis tinklas ir parodė neblogus atpažinimo rezultatus (~ 94,4), jautrumo ir specifiškumo matai vidutiniškai atitinkamai ~ 0,83 ir ~ 0,97, bet mus tenkinanti paklaida nei karto taip ir nebuvo pasiekta. Tinklas visada stabdomas priverstinai, todėl ir laiko sąnaudos gana didelės.

Kaip matome 4.3 b lentelė pradinių svorių vektoriai, kurie generuojami tik iš teigiamų intervalų, visiškai netinka sprendžiant rašto atpažinimo uždavinį, naudojant daugiasluksnį perceptroną, mokomą klaidos skleidimo atgal algoritmu. Santykis, tarp objektų tikrai turinčius mus dominančią savybę (kuri jau yra patvirtinta) su visa teigiamai klasifikuotų objektų aibe (jautrumas), net ir esant didesniai neuronų skaičiui pirmame ir antrame neuronų sluoksniuose (daugiau nei 5 neuronai), vidutiniškai ~ 0,166667 (žr. 4.3 b lentelė Jautrumas). O tai reiškia, kad ~ 84 % objektų, neturinčių mus dominančios savybės bus blogai klasifikuotos. Santykis tarp objektų tikrai neturinčių mus dominančių savybių su visa neigiamai klasifikuota objektų aibe (specifiškumas), net ir esant didesniai neuronų skaičiui pirmame ir antrame neuronų sluoksniuose (daugiau nei 5 neuronai), vidutiniškai ~ 0,833333 (žr. 4.3 b pav. Specifiškumas). Laiko sąnaudos gana didelės, skaičiavimai visada stabdomi priverstinai atlikus 10 tūkstančių epochų.

Apibendrinant šį tyrimą pavaizduosime gautus rezultatus diagramomis. Diagramose bus vaizduojamas bendras vidurkis, apskaičiuotas iš rezultatų, gautų visų šiame tyrime dalyvavusių

kompiuterių. Tai leis geriau pamatyti ir palyginti gautus rezultatus, bei apsispręsti, kuris pradinių svorių generavimo intervalas  $W$  geriausiai tinkamas šiam uždaviniui spręsti.



Iš diagramų galima matyti, kad geriausiai atpažinimo rezultatai gauti, kai pradinių svorių intervalai generuojami iš intervalų  $W(-1; 1)$  ir  $W(-0,7; 0,3)$ .

## 4.2 Optimalaus intervalo parinkimas

Kai pradiniai svoriai generuojami iš intervalo  $W(-1; 1)$  ir  $W(-0,7; 0,3)$  buvo gauti geri klasifikavimo rezultatai (žr. 4.1 pav.). Reikia atlikti dar vieną tyrimą su didesne raidžių aibe tam, kad nustatyti, koks intervalas sprendžiant šį uždavinį yra tinkamesnis. Šiam tyrimui atlikti buvo pasirinkta aibė iš 7 raidžių po 20 variantų (140 raidžių), neuroninio tinklo mokymo rezultatai bus vertinami atsižvelgiant į šiuos kriterijus: laikas (valandos, minutės, sekundės), bei paklaida, kuri vertinama pagal (2.5) formulę (apie tai skaitykite 2.3.3 skyrelyje). Jautrumas ir specifiškumas šį kartą nebus vertinami. Nes kaip parodė ankstesni tyrimai, kai pasiekiamas mus tenkinantis paklaidos slenkstis, jautrumo ir specifiškumo rezultatai gana geri (daugiau nei 0,99) (žr. 4.4 lentelė).

	<b>Paklaida</b>	<b>Laikas</b>
<b>W(-1; 1)</b>	140	09:00:00
<b>W(-0,7;0,3)</b>	0,000988	03:26:52

4.4 lentelė. Paklaida ir laikas didelei duomenų aibei.

Padidinus mokomų raidžių aibę, geriausi klasifikavimo tikslumo rezultatai gauti, kai pradinių svorių intervalas  $W(-0,7, 0,3)$ . Kaip matoma iš 4.4 paveikslėlio, vidutiniškai per 3 val. 27 min. buvo pasiektas nustatytas paklaidos slenkstis. Visai priešingus rezultatus parodė pradinių svorių intervalas  $W(-1, 1)$ . Mokymas visada buvo stabdomas priverstinai, vidutiniškai po 9 val. Paklaida, kaip matome 4.4 lentelėje liko didelė.

Atlikus šį tyrimą galime konstatuoti, kad naudojant daugiasluoksnio perceptrono modelį, bei klaidos skleidimo atgal algoritmą, geriausiai tinkantis pradinių svorių rinkinys  $W(-0,7, 0,3)$ .

## 4.3 Mokymo algoritmų palyginimas

Šiame skyriuje palyginsime „įprasto“ klaidos skleidimo atgal mokymo algoritmą ir patobulinto mokymo algoritmą – mokymą dalimis (žr. 3.4 skyrių). Mokymo aibė sudaryta iš 23 didžiųjų ir 23 mažųjų lotynų kalbos abėcėlės raidžių. Kiekviena raidė turi po 20 variantų, išskyrus „O“ ir „P“ (po 10 variantų), nes šių simbolių didžiųjų ir mažųjų raidžių užrašymo variantai nesiskiria. Viso mokymui naudojama 440 raidžių. Pradinių svorių generavimo intervalas buvo pasirinktas  $W(-0,7; 0,3)$ , nes jis (intervalas) pasirodė geriausiai tinkamas sprendžiant rašto atpažinimo problemą.

Šis tyrimas buvo atliekamas naudojant dvidešimt kompiuterių. Dešimt kompiuterių skaičiavo pagal įprastą algoritmą, o kiti dešimt kompiuterių skaičiavo pagal patobulintą mokymo algoritmą.

	<b>Paklaida</b>	<b>Laikas</b>
<b>Standartinis mokymas</b>	440	09:00:00
<b>Mokymas dalimis</b>	0,000988	03:46:36

4.5 lentelė. Paklaida ir laikas mokant tinklą standartiniu ir mokymo dalimis algoritmus.

Standartinio mokymas ir mokymo dalimis rezultatai (vidutinės paklaidos ir mokymo laikas) pateikti 4.5 lentelėje. Kaip matome, neuroninis tinklas mokomas pagal standartinį klaidos skleidimo atgal algoritmą, parodė blogus klasifikavimo rezultatus: didelė paklaida (440), mokymas visada buvo stabdomas priverstinai (po 9 val.). Visai priešingus rezultatus parodė mokymo dalimis strategija. Vidutiniškai po 3 val. 40 min. buvo pasiekta mus tenkinanti paklaida (mažiau nei 0,001) (žr. 4.5 pav.). Būtina atkreipti dėmesį, kad trokšamas paklaidos slenkstis yra labai mažas ( $< 0,0001$ ), taip pat kiekvienas mokymo raidės vektorius  $X$  turi atskirą trokštamą išėjimo vektoriaus  $T$  reikšmę. Tai reiškia, kad tuos pačios raidės skirtingus variantus neuroninis tinklas turi atpažinti kaip skirtingas raides, viena vertus mokymo laikas yra padidinamas, bet atpažinimo tikslumas pagerinamas. Vidutinis laikas, gautas suvidurkinus duomenis iš dešimties kompiuterių, bet 80 % atvejų neuroninio tinklo mokymas užtrukdavo mažiau nei 2 val. 30 min. Dirbtinio neuroninio tinklo mokymas dalimis parodė tokius pranašumus: žymų laiko „sąnaudų“ taupymą, palyginus su standartiniu mokymo algoritmu, lokaliųjų minimumų, bei tinklo paralyžiaus išvengimų. Jei mokant dalimis neuroninis tinklas vidutiniškai po 3 val. 40 min. (dažniausiai ir greičiau) buvo apmokomas atpažinti visą lotynų kalbos abėcėlę (žr. 4.5 pav.) (440 raidžių), tai mokant tinklą pagal standartinį algoritmą po 3 val. 27 min. tinklas mokėjo atpažinti tik dalį abėcėlės (žr. 4.4 pav.) (120 raidžių).

## 5. Apibendrinimas ir išvados

Darbe analizuojami rašto ženklų atpažinimo algoritmai, pagrįsti dirbtiniais neuroniniais tinklais. Išsamiai nagrinėjamas daugiasluoksnis tiesioginio sklidimo neuroninis tinklas, mokomas klaidos skleidimo atgal algoritmu. Darbe patobulinta Java applet sistema, kurios pagalba galima atlikti eksperimentus, kurių metu nustatomi tinkami svorių pradinių reikšmių generavimo intervalai, neuronų skaičius paslėptuose sluoksniuose. Atpažinimo (klasifikavimo) tikslumui vertinti realizuoti jautrumo ir specifiškumo matai. Java applet sistemos privalumas toks, kad nereikia jokios papildomos įrangos, užtenka tik interneto naršyklės. Darbe sukurta neuroninio tinklo mokymo strategija, kuri leidžia žymiai sumažinti tinklo mokymo laiką neprarandant klasifikavimo tikslumo.

Eksperimentinių tyrimų rezultatai leido daryti šias išvadas:

1. Kai pradinių svorių vektorius  $W$  generuojamas tik iš teigiamų reikšmių, gaunami netikslūs klasifikavimo rezultatai (jautrumas, specifiškumas). Dažnai paklaida lieka didelė (aibei 30 raidžių, paklaida ligi 30) atlikus daug iteracijų (10 tūkstančių). Tuomet tinklo mokymą tenka stabdyti priverstinai, nepriklausomai nuo mokymo aibės dydžio (30 raidžių, 140 raidžių ar 440 raidžių variantai).
2. Kai pradinių svorių vektorius  $W$  generuojamas ir iš teigiamų ir neigiamų reikšmių, gaunami tikslesni klasifikavimo rezultatai (daugiau nei 90 %). Geriausi klasifikavimo (atpažinimo) rezultatai gaunami, kai svorių vektoriai yra generuojami iš intervalo, kuriame neigiamų reikšmių nemažiau nei 50 %. Analizuojant nedidelę aibę (pvz., 30 raidžių), kai  $W \in (-1; 1)$  arba  $W \in (-0,7; 0,3)$ , gaunami labai tikslūs klasifikavimo rezultatai ( $\approx 99$  %).
3. Kai paslėptų sluoksnių neuronų kiekis yra didesnis (daugiau nei 9 neuronai), mokymas vyksta greičiau nei esant mažiems jų kiekiams.
4. Esant didesniam įėjimo vektorių skaičiui (raidžių kiekiui) (pvz., 140), geriausi atpažinimo rezultatai gaunami, kai pradiniai svoriai generuojami iš intervalo  $W \in (-0,7; 0,3)$ .
5. Naudojant standartinį algoritmą didelei raidžių aibei, t. y., 440 raidžių mokymui, gaunami netikslūs atpažinimo rezultatai (paklaida ligi 440). Mokymas visada buvo stabdomas priverstinai vidutiniškai po 9 valandų.
6. Kai mokymo aibė skaidoma į mažesnius poaibius ir mokymas vyksta dalimis, pasiekiami tikslūs atpažinimo rezultatai (paklaida  $\approx 0,00098$ ) žymiai greičiau, t. y., sutrumpinamas mokymo laikas. Palyginus su standartiniu algoritmu, mokymas pagreitėja daugiau nei tris kartus.

Dalis tyrimų rezultatų atlikti Studentų mokslinės praktikos metu (2010-06-21 – 2010-08-20), organizuojamos Lietuvos mokslo tarybos pagal projektą „Studentų mokslinės veiklos skatinimas“, finansuojamo pagal Žmogiškųjų išteklių plėtros veiksmų programą (kodas VP1-3.1-MM-01-V-01-002).

Taip pat rezultatai pristatyti 3-oje Lietuvos jaunųjų mokslininkų konferencijoje „Operacijų tyrimai verslui ir socialiniams procesams“ (2010-10-01) bei SMT ir SMP 2011 konferencijoje. Dalis rezultatų publikuoti darbe:

A. Andrejevas, O. Kurasova. Rašto ženklų atpažinimas naudojant neuroninius tinklus. *Studentų moksliniai darbai 2010–2011, II dalis, konferencijos medžiaga*. Vilnius, 2011 m. kovo 17 d. ISBN 978-609-417-023-2.

## SANTRAUKA

Magistriniame darbe tiriamos rašto ženklų atpažinimo problemos, nagrinėjami neuroniniai tinklai skirti rašto ženklams atpažinti. Apžvelgiamos problemos, kylančios sprendžiant rašto atpažinimo uždavinius, įvairūs problemų sprendimų būdai. Pasiūlytas dirbtinio neuroninio tinklo mokymo strategijos, pagrįstos klaidos skleidimo atgal algoritmu, patobulinimas. Patobulinimo esmė yra ta, kad mokymo aibė į tinklą paduodama ne visa iškart, o dalimis. Kai neuroninis tinklas išmoksta atpažinti tą dalį, mokymo aibė papildoma naujais duomenimis, bet pradinių svorių vektoriai nesikeičia ir tinklas mokomas toliau. Šis algoritmo patobulinimas leidžia ženkliai sumažinti apmokymo laiką neprarandant tikslumo. Kai kuriais atvejais neuroninis tinklas, mokomas pagal įprastą klaidos skleidimo atgal algoritmą, nerodė jokių mokymo perspektyvų, po ~ 9 val. paklaida nesikeisdavo, neuroninis tinklas negalėdavo teisingai atpažinti nei vienos raidės. Panaudojus patobulintą strategiją, mus tenkinanti paklaida pasiekama po ~ 3 val. 30 min.

Taip pat darbe tirama atpažinimo tikslumo priklausomybė nuo svorių pradinių reikšmių ir neuronų skaičiaus paslėptuose sluoksniuose. Nustatyti intervalai, kuriuose turi būti generuojamos svorių pradinės reikšmės, siekiant gauti tikslius atpažinimo rezultatus. Neuronų skaičius paslėptuose sluoksniuose turi būti daugiau nei penki.

# SUMMARY

## Handwriting character recognition using neural networks

The master thesis presents investigations of the problems of the optical character recognition. It also deals with the artificial neural networks that are designed for the optical character recognition. The work surveys the problems that emerge during the process of the optical character recognition. Various solutions are investigated. The improvement of a strategy for teaching the neural network that is based on the error back propagation is suggested. The essence of the improvement is that the training data set is divided into some parts and these parts are presented to the network one by one. When the neural network learns to recognize a part, the next part is presented to the network without any changes of the initial weight vectors and the network is trained further. This improvement allows us to reduce the training time significantly without losing the recognition accuracy. In some cases, the neural network that is trained according the ordinary error back propagation algorithm does not show any prospects. After ~ 9 hours, the error remains the same, the neural networks cannot recognize any letters. Using the improved strategy, the error satisfied is reached after ~3 hours 30 minutes.

The dependence of the recognition accuracy on the values of the initial weight vectors and on the number of neurons in hidden layers is also investigated. The intervals in which the values of the initial weight vectors must be generated are identified, in order to get the correct results of the recognition. The number of neurons in hidden layers must be more than five.



## Literatūra

1. Laimutis Telksnys, Antanas Žilinskas, Computers in Lithuania, IEEE Annals of the History of Computing 21, 31-37, (1999)
2. Купарадзе М.Р. Структура и функции нервной системы, Москва, Сборник докладов, 1962
3. Grossberg S. 1974. Classical and instrumental learning by neural networks. Progress in theoretical biology, vol. 3, pp. 51–141. New York: Academic Press.
4. Горбань А.Н., Методы Нейроинформатики, сборник научных трудов, Красноярск КГТУ, 1998
5. Амосова Н.М. Нейрокомпьютеры и интеллектуальные роботы. Киев, 1991.
6. Чернухин Ю.В. Микропроцессорное и нейрокомпьютерное управление адаптивными мобильными роботами. Таганрог, 1993.
7. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. 276 с.
8. Rosenblatt F. 1962. Principles of Neurodynamics. New York: Spartan Books.
9. Senashova Masha Yu., Gorban Alexander N., and Wunsch Donald, “Back-Propagation of Accuracy”// Proc. IEEE/INNS International Conference of Neural Networks, Houston, IEEE, 1997, pp.1998-2001
10. Lawrence S., Giles C.L., Tsoi A.C., Back A.D. Face Recognition : A Convolutional Network Approach // IEE Transactions on Networks, Special Issue on Neural Networks and Pattern Recognition. 1977.p.1-24.
11. Simard P.Y. “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis”, Microsoft . – 1998. – P. 23 – 24.
12. OCHRE: Optical Character Recognition <http://www.sund.de/netze/applets/BPN/bpn2/ochre.html> (žiūrėta 2011. gegužė 3 d.)
13. Convolutional neural network <http://sites.google.com/site/mihailsirotenko/projects/convolutional-neural-network-class> (žiūrėta 2011 m. gegužė 3 d.)
14. The mnist database of handwritten digits <http://yann.lecun.com/exdb/mnist/index.html> (žiūrėta 2011 m. gegužė 3 d.)
15. G. Dzemyda, O. Kurasova, J. Žilinskas. Daugiamatčių duomenų vizualizavimo metodai, Vilnius: Mokslo Aidai, 2008
16. Š. Raudys (2001). Statistical and Neural Classifiers: an Integrated Approach to Design. Advances in Pattern Recognition. Springer-Verlag.

## **Priedas**

A. Andrejevas, O. Kurasova. Rašto ženklų atpažinimas naudojant neuroninius tinklus. *Studentų moksliniai darbai 2010–2011, II dalis, konferencijos medžiaga*. Vilnius, 2011 m. kovo 17 d. ISBN 978-609-417-023-2