VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Ekstremalaus mašininio mokymo efektyvumo tyrimas

Investigation of performance of Extreme Learning Machine

Magistro baigiamasis darbas

Atliko: Povilas Vabalas (parašas)

Darbo vadovas: asist. dr. Valdas Dičiūnas (parašas)

Recenzentas: lekt. dr. Rimantas Kybartas (parašas)

Vilnius – 2018

Santrauka

Šiame darbe apžvelgiama literatūra apie ekstremalaus mašinų mokymo (ELM) schemą ir aprašomas jos efektyvumo tyrimas kartu su eksperimentų rezultatais. ELM sudaryta iš vieno paslėptojo sluoksnio dirbtinio neuroninio tinklo ir jo mokymo algoritmo. Ši schema pristatyta 2004 m.; lyginant su ankstesnėmis mašinų mokymo schemomis ji pasižymi supaprastintu mokymo algoritmu.

ELM inicializuoja paslėptojo sluoksnio įvesties svorius ir postūmius atsitiktinėmis reikšmėmis, o jo išvesties svoriai ir postūmiai nustatomi sprendžiant tiesinių lygčių sistemą. Generalizavimo efektyvumas maksimizuojamas minimizuojant mokymo klaidą ir paslėptojo sluoksnio išvesties svorių normą, o naudojant praktiniuose taikymuose — ieškant pusiausvyros tarp mokymo klaidos ir paslėptojo sluoksnio dydžių.

Tyrimas atliktas su didelės dimensijos tekstiniais duomenimis. ELM ir jos modifikacijos R-ELM klasifikavimo efektyvumui tirti panaudoti Reuters-21578 ir 20 Newsgroups duomenų rinkiniai. Suskaičiuoti duomenų tfidf įverčiai, tuomet nuo pradinės nemažinant duomenų dimensijos iš jų sudarytas VSM (angl. *Vector Space Model*).

ELM ir R-ELM generalizavimo rezultatai palyginti su SVM. Klasifikuojant Reuters-21578 rinkinio duomenis SVM pateikė klasifikavimo klaidos požiūriu su ELM konkurencingus rezultatus, bet sugaišo $\sim 5 - 7$ kartus daugiau laiko. Su didesnės dimensijos ir apimties 20 Newsgroups rinkiniu ELM buvo tikslesnė ir greitesnė už SVM. Bandymas ELM rezultatus palyginti su BP buvo nesėkmingas dėl BP mokymui reikalingų laiko sąnaudų. Kartu išbandytas OP-ELM mokymo algoritmas taip pat pasirodė per daug imlus laikui.

Raktiniai žodžiai: ELM, R-ELM, OP-ELM, BP, SVM, tekstų klasifikavimas.

Summary

This thesis contains a review of literature on Extreme Learning Machine (ELM) and presents an investigation of it's performance together with the results. ELM comprises single hidden layer artificial neural network and it's training algorithm. This scheme was first proposed in year 2004; in comparison with earlier machine learning methods, it's training algorithm is much simplified.

ELM initialises hidden layer's input weights and biases with randomly picked values, and calculates it's output weights by solving a system of linear equations. ELM's generalization efficiency is maximized by simultaneously minimising it's training error and it's norm of output weights. This is achieved in practice by balancing it's training error and hidden layer size.

This investigation was performed on high-dimensional textual data, which comprises well known Reuters-21578 and 20 Newsgroups datasets. These datasets were transformed into their tfidf values and then mapped into Vector Space Model representation. The initial dimensionality of both datasets was preserved.

Performance results of ELM and R-ELM were compared to those of SVM. In case of Reuters-21578 dataset, both methods performed similarly, however, SVM was approx. 5-7 times slower. With higher-dimensional and higher-volume data of 20 Newsgroups dataset ELM performed better in terms of both correctness and speed.

Attempt to compare ELM's efficiency to BP's was not successful because of latter's much higher training times. This was also true in OP-ELM's case.

Keywords: ELM, R-ELM, OP-ELM, BP, SVM, classification of textual data.

Turinys

1. Įvadas

Šiame darbe rašoma apie ekstremalaus mašinų mokymo schemą (angl. *Extreme Learning Machine*, ELM). Ji pristatyta 2004 m.; lyginant su tradicinėmis mašinų mokymo schemomis, ELM mokymo algoritmas yra radikaliai supaprastintas. Dėl to ženkliai sutrumpėja ELM mokymo laikas, o generalizavimo efektyvumas stipriai nenukenčia. Šiame darbe tiriamos ELM savybės klasifikuojant didelės dimensijos tekstinius duomenis.

Šio tyrimo tikslas yra: 1) ištirti kaip ELM efektyvumas priklauso nuo schemos laisvųjų parametrų (pvz., paslėptojo sluoksnio dydžio); 2) ištirti kaip ELM efektyvumas priklauso nuo tiriamų duomenų savybių (pvz., mokymo imties dydžio, duomenų dimensiškumo); 3) palyginti ELM generalizavimo efektyvumą su kitomis schemomis (pvz., atraminių vektorių mašina (angl. *Support Vector Machine*, SVM)); 4) paruošti ELM mokymo rekomendacijas naudotojui.

Darbo uždaviniai: 1) išnagrinėti ELM pristatančią ir tiriančią medžiagą; 2) sudaryti schemos programinę realizaciją; 3) parinkti ir parengti mokymo ir testavimo duomenis; 4) atlikti ELM mokymą ir testavimą; 5) palyginti ELM rezultatus su kitais metodais; 6) parengti išvadas apie ELM efektyvumą.

Darbas sudarytas iš dviejų dalių. Pirmoji dalis (2. skyrius) skirta literatūros apie ELM apžvalgai. Apžvelgiamos temos: klasikinis schemos apibrėžimas (2.1. poskyris), ją grindžiantys teoriniai teiginiai (2.2. poskyris), ankstesni už ELM dirbtinių neuroninių tinklų (DNT) parametrų inicializavimo atsitiktinėmis reikšmėmis tyrimai (2.3. poskyris), įvairios ELM modifikacijos (2.4. poskyris). Antroji dalis (3. skyrius) skirta rengiant darbą atlikto tyrimo ataskaitai. Joje rašoma apie ELM generalizavimo efektyvumą klasifikuojant tekstinius duomenis. Ataskaita sudaryta iš metodinės ir tiriamosios dalių. Metodinę sudaro trys pirmieji smulkesnieji poskyriai: mašininiam apdorojimui tinkamos tekstinių duomenų reprezentacijos sudarymas (3.1.1. poskyris), klasifikavimo metodų parinkimas (3.1.2. poskyris), tyrimo rezultatų apdorojimas ir vertinimas (3.1.3. poskyris). Tiriamąją dalį sudaro trumpa naudotų įrankių apžvalga (3.2.1. poskyris), duomenų rinkinio aprašymas (3.2.2. poskyris), duomenų parengimo ir klasifikavimo procedūros aprašymas (3.2.3. poskyris) ir tyrimo rezultatai (3.2.4. poskyris). Prie darbo pridedamas diskas su tyrimo metu sudarytu programiniu kodu.

(2.1. poskyris) Klasikinis ELM variantas pristatytas [HZS04] ir detalizuotas [HZS06] straipsniuose. ELM schema sudaryta iš vieno paslėptojo sluoksnio tiesioginio sklidimo DNT (angl. *Single hidden Layer Feedforward neural Network*, SLFN) ir jo mokymo algoritmo. Matematinis paslėptojo sluoksnio modelis — įvesties matrica $\mathbf{H}$ ir išvesties svorių matrica $\mathbf{V}$. ELM mokymo algoritmas turi tris žingsnius: 1) paslėptojo sluoksnio įvesties svorių $\mathbf{W}$ ir postūmių $\mathbf{B}$ inicializavimas pagal pasirinktą tikimybinį skirstinį generuojamomis atsitiktinėmis reikšmėmis, 2) paslėptojo sluoksnio įvesties matricos $\mathbf{H}$ sudarymas naudojant mokymo duomenų rinkinį $\mathbf{X}$ ir pirmame žingsnyje parinktus svorius ir postūmius, 3) paslėptojo sluoksnio išvesties svorių $\mathbf{V}$ apskaičiavimas, naudojant ankstesniame žingsnyje sudarytą $\mathbf{H}$ ir mokymo tikslinių reikšmių matricą $\mathbf{T}$.

Paslėptojo sluoksnio matrica naudojama atlikti netiesinei įvesties duomenų transformacijai į požymių erdvės (angl. *feature space*) taškus. Aktyvavimo funkcijos, kurios gali būti įvairios dalimis tolydžios (angl. *piecewise continuous*), suspaudžia transformuotų duomenų reikšmių intervalų

ribas.

Pagrindinės ELM savybės [HZS04]:

- ELM mokymo laikas yra trumpesnis už gradientinio nusileidimo metodą mokymo klaidos reikšmei minimizuoti naudojančių tradicinių algoritmų;
- ELM mokymo algoritmas kartu minimizuoja mokymo klaidą ir paslėptojo sluoksnio išvesties svorių normas, todėl pasižymi stabilumu;
- ELM aktyvavimo funkcijų pasirinkimas yra didesnis negu tradicinių gradientinio nusileidimo metodą naudojančių algoritmų, nes nėra griežtų reikalavimų funkcijų diferencijuojamumui;
- ELM mokymo algoritmas, skirtingai nuo gradientinį nusileidimą naudojančių metodų, neįstringa lokaliuose minimumuose, jį naudojant nereikia eksperimentiškai parinkti mokymosi spartos parametro reikšmės, jį nesudėtinga realizuoti.

Didžiausias ELM schemos trūkumas yra didesnė negu tradicinių metodų generalizavimo fazės trukmė; dėl atsitiktinio paslėptojo sluoksnio matricos **H** pobūdžio jį tenka naudoti didesnį, todėl didėja ir skaičiavimų apimtis. Kaip trūkumą reikia paminėti ir paslėptojo sluoksnio dydžio parinkimo būtinybę, kadangi nuo jo priklauso mokymo ir generalizavimo klaidos dydis, bet tas pats būdinga ir kitoms mašinų mokymo schemoms.

Kai kurie ELM variantai naudoja tam tikras SLFN kombinacijas (angl. *Multi hidden Layer Feedforward neural Network*, MLFN). ELM galima naudoti klasifikavimui ir regresijai.

2.2. poskyryje rašoma apie teorines ELM prielaidas. ELM generalizavimo efektyvumas apibrėžiamas ir pagrindžiamas trimis aspektais: 1) interpoliavimo (apibrėžia kiek efektyvų duomenų modelį ELM sudaro iš mokymo imties); 2) universalaus funkcijų aproksimavimo (kokios yra ELM modelių sudėtingumo ir įvairovės ribos); 3) generalizavimo klaidos ribos (kokia yra apibrėžto didumo generalizavimo klaidos tikimybės riba kai žinomos kitų ELM laisvųjų parametrų reikšmės).

Pagrindinė ELM generalizavimo efektyvumą apibrėžianti teorija — P. Bartlett teorija apie generalizavimo klaidos ryšį su mokymo klaidos ir išvesties svorių normos dydžiais. Ši teorija, savo ruožtu, remiasi V. Vapnik ir A. Červonenkis statistine mokymo teorija. Esminis P. Bartlett teorijos teiginys — siekiant minimizuoti generalizavimo klaidą, reikia kartu minimizuoti mokymo klaidą ir DNT išvesties svorių normą.

2.3. poskyryje pristatomi du už ELM ankstesni tyrimai, kuriais siekta sumažinti optimizuojamų DNT parametrų skaičių. Vienas šių tyrimų pristatytas 1992 m., jis atliktas su atsitiktiniais neuronų svoriais, bet apskaičiuojamais postūmiais [SKD92], kitas — 1994 m. pristatytas tyrimas (RVFL) — su sigmoidinio arba RBF tipo mazgais, kurių svoriai arba centrai buvo parenkami atsitiktiniai, bet postūmiai arba reikšmingumo veiksniai (angl. *impact factor*) buvo optimizuojami [PPS94]. Esminis ELM skirtumas nuo šių tyrimų — visi laisvieji įvesties parametrai yra atsitiktinai generuojami.

Likusioje skyriaus dalyje apžvelgiama ELM modifikacijų ir taikymų literatūra. Dalis jų — R-ELM ir OP-ELM šiame tyrime lyginami su klasikiniu ELM ir kitais mašinų mokymo metodais.

Tiriamoji dalis (3. skyrius). Tekstų klasifikavimą naudojant mašinų mokymą galima skirti į tris etapus [ZQL13]: 1) mašininiam apdorojimui tinkamos teksto reprezentacijos sudarymą; 2) klasifikavimo metodo parinkimą; 3) klasifikavimo rezultatų apdorojimą ir vertinimą. Trečiojo skyriaus dalys:

1) metodinėje dalyje rašoma apie tyrimui pasirinktus tekstinių duomenų parengimo būdus ir jų pasirinkimo motyvus. Toliau rašoma apie ELM naudojimą tekstiniams duomenims klasifikuoti ir apie klasikinės ELM (angl. *Extreme Learning Machine*) schemos mokymo algoritmo efektyvumą didinančių modifikacijų — R-ELM (angl. *Regularized ELM*) ir OP-ELM (angl. *Optimally Pruned ELM*) pasirinkimą šiam tyrimui. Po to rašoma apie klasifikavimo rezultatų vertinimo būdus, argumentuojamas DP (daugiasluoksnio perceptrono) su BP (angl. *Backpropagation*) mokymo algoritmu ir SVM mašinų mokymo schemų pasirinkimas su kuriomis šie rezultatai lyginami;

2) tiriamojoje dalyje pristatomas tyrimui panaudotas duomenų rinkinys, su mokymo duomenimis atliktos generalizavimo efektyvumą didinančios transformacijos, tiriamų mašinų mokymo schemų (ELM, R-ELM, OP-ELM) rezultatai. Po to tyrimo rezultatai palyginti su kitomis mašinų mokymo schemomis — DP (su BP) ir SVM;

3) tyrimo ataskaitos gale suformuluotos tyrimo išvados ir ELM naudojimo tekstams klasifikuoti rekomendacijos;

4) prieduose pridedamas tyrimo metu naudotų mašinų mokymo schemų programų kodas Octave programavimo kalba.

Šis tyrimas naujas tuo, kad jame naudojami tekstų klasifikavimo metodai leidžia atlikti didelio dimensiškumo duomenų apdorojimą nepasitelkiant sudėtingų pradinės tekstų analizės metodų ir nemažinant pradinių duomenų dimensiškumo. Naudojamos mašinų mokymo schemos — ELM rezultatai: 1) už tradicinius metodus didesnis darbinės atminties naudojimas mokymo ir generalizavimo metu; 2) didesni kompiuterio skaičiavimo galios reikalavimai generalizavimo metu; 3) santykinai paprastas naudojimas; 4) didesnis DNT mokymo greitis. Generalizavimo efektyvumas priėmus tokias sąlygas yra panašus į tradicinių schemų.

Šio tyrimo indėlis mokslinio reikšmingumo prasme yra teiginio, kad galima atsisakyti dalies DP laisvųjų parametrų optimizavimo juos fiksuojant (atsitiktinėmis ar kokiu nors kitu būdu parinktomis reikšmėmis) tyrimas ir aiškinimasis kaip tai veikia DNT mokymo algoritmo sudarymą ir generalizavimo efektyvumą. Yra žinoma, kad optimizuojant visus DNT laisvuosius parametrus, kaip BP atveju, reikia ilgai trunkančio sudėtingos mokymo procedūros taikymo, kuris negali garantuoti optimalaus sprendinio suradimo. BP algoritmas reikalauja ir tinkamų naudotojo įgūdžių — parengti duomenis, parinkti pradines svorių, postūmių ir kt. parametrų reikšmes, pasirinkti aktyvavimo funkcijas, kurios privalo būti diferencijuojamos ne mažiau kartų, nei tinklo gylio reikšmė, nuspręsti dėl reikalingo mokymo epochų skaičiaus. Bent iš dalies šių parametrų parinkimą galima automatizuoti, bet taip dar labiau padidėja mokymo trukmė. Tuo tarpu pagrindiniam ELM variantui pakanka parengti duomenis, parinkti atsitiktinai generuojamų parametrų reikšmių intervalus, skirstinius ir paslėptojo sluoksnio dydį. Aktyvavimo funkcijos gali būti ir nediferencijuojamos.

ELM generalizavimo efektyvumo kontekste yra svarbus paslėptojo sluoksnio dydžio klausimas. Nuo jo priklauso tinklo išmokstamo modelio sudėtingumas, bet nuo jo priklauso ir generalizavimo stabilumas, t.p., ir skaičiavimų apimtis testavimo (generalizavimo) metu. R-ELM sprendžia stabilumo klausimą minimizuodama išvesties sluoksnio svorių normą, bet už tai yra aukojama mokymo klaida ir nėra sumažinama skaičiavimų apimtis (kitaip tariant, tinklas išmoksta galimai perteklinę ir nestabilią modelio reprezentaciją, ir tik išvesties sluoksnyje ją papildomai stabilizuoja). OP-ELM tiesiogiai sprendžia išmokstamo modelio sudėtingumo ir generalizavimo efektyvumo klausimą ribodama paslėptojo sluoksnio dydį (genėdama mažai reikšmingus neuronus). OP-ELM schemos kryptimi vykdomi tyrimai reikšmingi ir tuo, kad prisideda sprendžiant pagrindinę ELM komplikaciją — paslėptojo sluoksnio dydžio klausimą, kurį paprastai tenka spręsti empiriškai.

Didelio dimensiškumo, bet santykinai mažos mokymo imties atveju ELM schema t.p. susiduria su generalizavimo stabilumo sunkumais, bet dėl priešingos priežasties — modelio nepakankamumo. R-ELM sprendimas šiuo atveju t.p. tinkamas, bet OP-ELM efektyvumą tokiomis sąlygomis reikia patikrinti, nes nepakankamo modelio atveju visi arba didžioji dauguma paslėptojo sluoksnio neuronų yra generalizavimo rezultatų atžvilgiu statistiškai reikšmingi.

Kitas šio tyrimo rezultatas — ELM schemos naudojimo tekstų klasifikavimui rekomendacijos: 1) pirminio duomenų parengimo (centravimas, standartizavimas, dekoreliavimas ir kt.); 2) tinklo laisvųjų parametrų parinkimo (pradinių svorių ir postūmių reikšmių intervalai ir skirstiniai, paslėptojo sluoksnio dydis, reguliarizavimo konstantos dydis, jei naudojama R-ELM, o OP-ELM atveju — neurono reikšmingumo rodiklio slenkstis).

Prie tyrimo pridedamas ir jo metu parašytas programinis kodas su ELM schemos ir naudotų jos modifikacijų realizacijomis. Tyrimo metu taip pat parašyta ir tekstams klasifikuoti naudota BP algoritmo realizacija.

2. Literatūros apžvalga

2.1. Klasikinis ELM apibrėžimas

Klasikinis ELM mašinų mokymo schemos variantas pristatytas [HZS04]. 2006 m. straipsnyje [HZS06] pateiktas išsamus jos aprašymas, kuris vėliau tikslintas ir pildytas [HHS+15; Hua15].

Šia schema siekiama sutrumpinti DNT mokymo laiką atsisakant gradientinio nusileidimo klaidos reikšmės minimizavimo metodo bei dalį tinklo parametrų inicializuojant atsitiktinėmis reikšmėmis.

Klasikiniame ELM apibrėžime [HZS06] ši schema suformuluota kaip tiesioginio sklidimo dirbtinis neuroninis tinklas su vienu paslėptu sluoksniu (SLFN) ir jo mokymo algoritmas. Yra kitokių ELM tinklų topologijos variantų tyrimų, bei ELM ir kitokių mašinų mokymo schemų kombinavimo tyrimų. Šia tema daugiau – 2.4. poskyryje (ELM tyrimai).

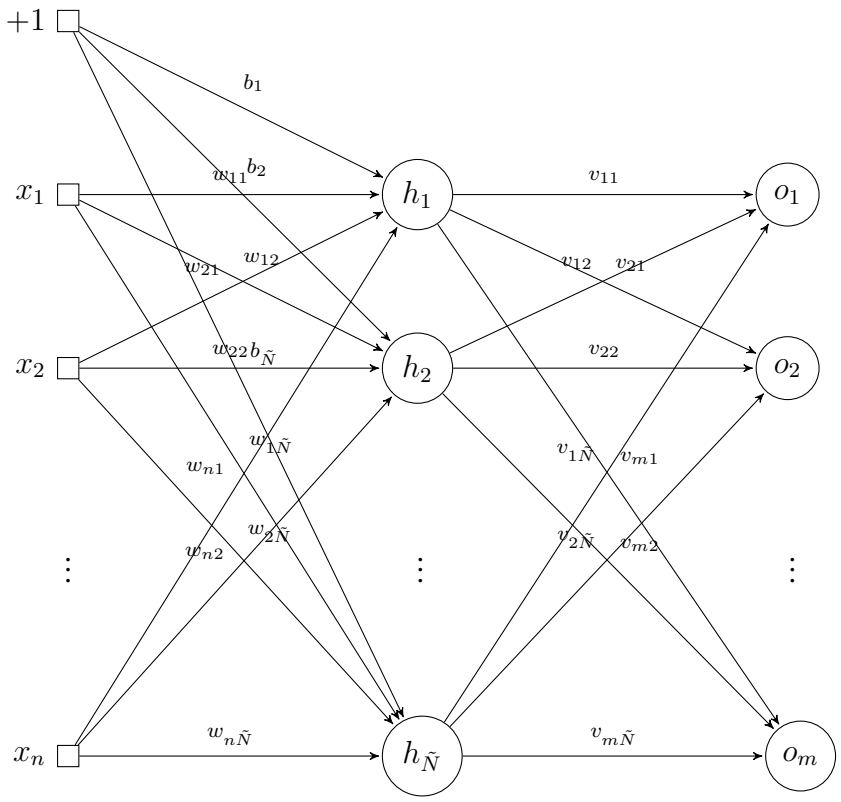
Šio tinklo paslėptojo sluoksnio neuronų įvesties jungčių poslinkiai ir svoriai yra parenkami atsitiktinai, o išvesties jungtys turi svorius, kurie yra apskaičiuojami, bet neturi poslinkių. ELM mokymo algoritmas yra greitesnis už tradicinius, o juo apmokyti tinklai pasiekia su tradiciniais sulyginamą generalizavimo efektyvumą su nedaug didesnėmis skaičiavimų sąnaudomis [HZS06]. Kitaip nei atbulinio klaidos skleidimo (angl. *back-propagation (BP)*) mokymo algoritmas, ELM mokymo algoritmas nereikalauja, kad neuronų aktyvavimo funkcijos būtų diferencijuojamos, todėl paslėptojo ir išvesties sluoksnių neuronų aktyvavimo funkcijos gali būti įvairesnės nei tradicinių algoritmų [HZS04]. Toliau pristatomi klasikinės ELM schemos elementai: dirbtinis neuroninis tinklas ir jo mokymo algoritmas.

Tiesioginio sklidimo dirbtinis neuroninis tinklas su vienu paslėptu sluoksniu. Naudojamas tiesioginio sklidimo dirbtinis neuroninis tinklas su vienu paslėptu sluoksniu. Tinklo architektūrinės diagramos pavyzdį žr. 1 pav. Grafo viršūnės $x_p,\ p = 1, \ldots, n$ žymi įvesties sluoksnio elementus, $h_q,\ q = 1, \ldots, \tilde{N}$ – paslėptojo sluoksnio neuronus, $o_r,\ r = 1, \ldots, m$ – išvesties sluoksnio neuronus. Grafo lankai $w_{qp},\ p = 1, \ldots, n,\ q = 1, \ldots, \tilde{N}$ žymi paslėptojo sluoksnio neuronų jungčių svorius, $v_{qr},\ q = 1, \ldots, \tilde{N},\ r = 1, \ldots, m$ – išvesties sluoksnio neuronų jungčių svorius. Nagrinėjamo tinklo architektūra – $n + \tilde{N} + m$.

Tegul $L = \{(\mathbf{x}_i, \mathbf{t}_i)\}, i = 1, \ldots, N$ – mokymo duomenų imtis, kur $\mathbf{x}_i = [x_{i1}, \ldots, x_{in}]^T \in \mathbb{R}^n$, $\mathbf{t}_i = [t_{i1}, \ldots, t_{im}]^T \in \mathbb{R}^m$ (čia ir toliau naudojamos mokymo tikslinių reikšmių $\mathbf{t}_i$ ir išvesties $\mathbf{o}_i$ vektorinės išraiškos, sudarytos iš išvesties sluoksnio neuronų aktyvavimo funkcijų rezultatų vektorių [HZS06]). Nagrinėjamo neuroninio tinklo paslėptojo sluoksnio neuronų skaičius yra $\tilde{N}$, o $g(x)$ yra neurono aktyvavimo funkcija. Tokio tinklo matematinė išraiška:

$$\sum_{j=1}^{\tilde{N}} \mathbf{v}_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = \mathbf{o}_i, \quad i = 1, \ldots, N.$$

Čia $\mathbf{w}_j = [w_{j1}, \ldots, w_{jn}]^T$ – paslėptojo sluoksnio j-ojo neurono jungčių svorių vektorius, $\mathbf{v}_j = [v_{1j}, \ldots, v_{mj}]^T$ – paslėptojo sluoksnio j-ąjį neuroną su išvesties sluoksniu jungiančių jungčių svorių



1 pav. Tiesioginio sklidimo dirbtinis neuroninis tinklas su vienu paslėptu sluoksniu

vektorius, b_j – paslėptojo sluoksnio j-ojo neurono poslinkio didumas.

Ekstremalaus mašininio mokymo algoritmas.

1. Paslėptojo sluoksnio inicializavimas. Paslėptojo sluoksnio neuronų jungčių svoriams $\mathbf{w}_j$ ir poslinkiams b_j priskiriamos atsitiktinės reikšmės iš pasirinkto intervalo (šių reikšmių tikimybinis skirstinys – laisvasis algoritmo parametras).

2. Paslėptojo sluoksnio išėjimo matricos H ir išėjimo sluoksnio svorių matricos V radimas. Šiame žingsnyje naudojant anksčiau priskirtus parametrus apskaičiuojama paslėptojo sluoksnio išėjimo matrica $\mathbf{H}$. Toliau tiesioginio sklidimo dirbtinį neuroninį tinklą su vienu paslėptu sluoksniu galima laikyti tiesine sistema $\mathbf{HV} = \mathbf{T}$. Čia

$$\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \vdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}},$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}.$$

Tada $\mathbf{V} = \mathbf{H}^+\mathbf{T}$, kur $\mathbf{H}^+$ yra Moore-Penrose pseudoatvirkštinė matrica [Hay09].

3. Tiesinių lygčių sistemos $\mathbf{HV} = \mathbf{T}$ minimalios normos mažiausiųjų kvadratų sprendinio radimas. Šiame mokymo algoritmo žingsnyje atliekamas klaidos funkcijos minimizavimas ir išvesties sluoksnio jungčių svorių normos minimizavimas [HZS04]. Jei paslėptojo sluoksnio dydis yra mažesnis už mokymo imties įrašų skaičių – $\tilde{N} < N$ – ne visais atvejais galima pasiekti nulinę mokymo klaidą (apie tai daugiau – 2.2. poskyryje (Teoriniai ELM pagrindai))

$$E = \sum_{i=1}^{N} \left(\sum_{j=1}^{\tilde{N}} \mathbf{v}_j g(\mathbf{w}_j \cdot \mathbf{x}_i - b_j) - \mathbf{t}_i \right)^2 .$$

Tuomet klaida turi būti minimizuojama randant tinkamus parametrus $\hat{\mathbf{w}}_i$, $\hat{b}_i$ ir $\hat{\mathbf{V}}$:

$$\|\mathbf{H}(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{\tilde{N}}, \hat{b}_1, \ldots, \hat{b}_{\tilde{N}})\hat{\mathbf{V}} - \mathbf{T}\| = \min_{\mathbf{w}_i, b_i, \mathbf{V}} \|\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\mathbf{V} - \mathbf{T}\|.$$

Tačiau 1 žingsnyje $\mathbf{w}_i$ ir b_i priskyrus atsitiktines reikšmes minimizuojama funkcija pertvarkoma taip:

$$\|\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\hat{\mathbf{V}} - \mathbf{T}\| = \min_{\mathbf{V}} \|\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\mathbf{V} - \mathbf{T}\|.$$

Lygčių sistemos $\mathbf{HV} = \mathbf{T}$ sprendinys yra

$$\hat{\mathbf{V}} = \mathbf{H}^{+}\mathbf{T}.$$

Sprendinio savybės [HZS04]:

1. minimali mokymo klaida

$$\|\mathbf{H}\hat{\mathbf{V}} - \mathbf{T}\| = \|\mathbf{H}\mathbf{H}^{+}\mathbf{T} - \mathbf{T}\| = \min_{\hat{\mathbf{V}}} \|\mathbf{H}\mathbf{V} - \mathbf{T}\|;$$

2. minimali išvesties sluoksnio neuronų jungčių svorių norma

$$\|\hat{\mathbf{V}}\| = \|\mathbf{H}^{+}\mathbf{T}\| \leq \|\mathbf{V}\|, \forall \mathbf{V} \in \{\mathbf{V} : \|\mathbf{H}\mathbf{V} - \mathbf{T}\| \leq \|\mathbf{H}\mathbf{Z} - \mathbf{T}\|, \forall \mathbf{Z} \in \mathbb{R}^{\tilde{N} \times N}\};$$

3. $\hat{\mathbf{V}} = \mathbf{H}^{+}\mathbf{T}$ yra unikalus tiesinių lygčių sistemos $\mathbf{HV} = \mathbf{T}$ sprendinys.

ELM algoritmu apmokytas neuroninis tinklas testavimo metu veikia kaip tiesinė sistema. Kadangi kartu minimizuojamos neuroninio tinklo mokymo klaida ir jungčių svorių norma, tuo pačiu minimizuojama ir testavimo klaida [Bar97; Bar98].

Su atitinkamai parinktais parametrais ELM gali klasifikuoti norimu tikslumu ir aproksimuoja uždarose aibėse bet kokias funkcijas [HZS04].

2.2. Teoriniai ELM pagrindai

ELM schemos teorinis pagrindimas formuluojamas trimis aspektais [HHS+15]:

1. interpoliavimo;
2. universalaus aproksimavimo;
3. generalizavimo efektyvumo ribos.

Interpoliavimas. ELM schema gali aproksimuoti norimai mažą mokymo klaidą $\epsilon > 0$, jei tenkinamos šios sąlygos:

- paslėptojo sluoksnio neuronų aktyvavimo funkcijos yra be galo diferencijuojamos (glodžios) bet kuriame intervale;
- mokymo imtis $\{(\mathbf{x}_i, \mathbf{t}_i)\}, \mathbf{x}_i \in \mathbb{R}^n, \mathbf{t}_i \in \mathbb{R}^m$, sudaryta iš N skirtingų elementų;
- paslėptojo sluoksnio dydis yra $\tilde{N} < N$;
- atsitiktinai pagal bet kokį tolydųjį tikimybinį skirstinį parenkami paslėptojo sluoksnio svoriai ir poslinkiai $\{\mathbf{w}_j, b_j\}_{j=1}^{\tilde{N}}, \mathbf{w}_j \in \mathbb{R}^N, b_j \in \mathbb{R}$,

tai yra teisinga nelygybė $\|\mathbf{HV}-\mathbf{T}\| < \epsilon$. Jei paslėptojo sluoksnio dydis $\tilde{N} = N$, tada $\|\mathbf{HV}-\mathbf{T}\| = 0$ [HHS+15; HZS06; TT97].

Šis teiginys susieja ELM mokymo klaidos didumą su paslėptojo sluoksnio dydžiu.

Universalus funkcijų aproksimavimas. ELM nuo tradicinių mašinų mokymo schemų skiriasi tuo, kad paslėptojo sluoksnio įvesties svoriai ir poslinkiai nėra pritaikomi prie mokymo imties duomenų, todėl, viena vertus, sutrumpėja paslėptojo sluoksnio parengimas darbui, kita vertus, tinklo mokymui tampa nebeprivaloma naudoti glodžias aktyvavimo funkcijas, nes nenaudojamas klaidos atgalinis skleidimas. Atsitiktinis paslėptojo sluoksnio inicializavimas ir laisvesnis aktyvavimo funkcijų pasirinkimas nesutrukdo ELM aproksimuoti tokių pačių funkcijų, kaip ir tradicinėms schemoms.

Jei tenkinamos šios sąlygos:

- paslėptojo sluoksnio aktyvavimo funkcija $g : \mathbb{R}^n \mapsto \mathbb{R}$ yra intervaluose tolydi ir ne konstanta;
- apvalkalas (angl. *span*) $\{g(\mathbf{w}_j, b_j, \mathbf{x}_i)\}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, \tilde{N}$ yra tankus erdvėje L^2;
- $\{g(\mathbf{w}_j, b_j, \mathbf{x}_i)\}$ inicializuojama atsitiktiniais, pagal kokį nors tolydųjį tikimybinį skirstinį parenkamais parametrais;
- tikslo funkcija f yra bet kokia diferencijuojama funkcija,

tai teisinga lygybė $\lim_{\tilde{N}\to\infty} \|f - f_{\tilde{N}}\| = 0$, jei išvesties sluoksnio jungčių svoriai $\mathbf{v}_j$ apskaičiuojami minimizuojant $\|f(\mathbf{x}) - \sum_{j=1}^{\tilde{N}} \mathbf{v}_j g(\mathbf{w}_j, b_j, \mathbf{x})\|$, kur $f_{\tilde{N}} = \sum_{j=1}^{\tilde{N}} \mathbf{v}_j g(\mathbf{w}_j, b_j, \mathbf{x})$.

Taigi, jei ELM naudoja intervaluose tolydžias aktyvavimo funkcijas, kurios yra tankios erdvėje L^2 ir aproksimuoja tolydžias tikslo funkcijas, bei turi atsitiktinai pagal tolydųjį skirstinį inicializuojamą paslėptojo sluoksnio matricą, tikslo funkciją galima aproksimuoti su norimai maža klaida, jei

paslėptojo sluoksnio dydis (modeliuojamų funkcijų-hipotezių skaičius) yra pakankamas [HHS+15]. Aktyvavimo funkcijos neprivalo būti tolydžios visoje jų apibrėžimo srityje, todėl galima naudoti, pvz., slenkstinę aktyvavimo funkciją.

Šis teiginys reiškia, kad ELM su pakankamai dideliu paslėptuoju sluoksniu gali aproksimuoti norimo sudėtingumo skiriamąjį hiperpaviršių.

Generalizavimo klaidos riba. Ši poskyro dalis sprendžia klausimą kaip testavimo klaida priklauso nuo mokymo klaidos, tinklo parametrų skaičiaus (kitaip tariant, paslėptojo sluoksnio didumo), mokymo imties didumo. Gretimas klausimas – išvesties svorių normos didumo ryšys su testavimo klaida.

Atsakant į šiuos klausimus panaudotas mokymo klaidos reprezentatyvumo testavimo klaidos įverčio atžvilgiu priklausomybės nuo mokymo imties dydžio sprendimas. Statistinės mokymo teorijos požiūriu, mokymo imties dydis turėtų būti tiesiškai proporcingas schemos naudojamų funkcijų klasės VC (Vapnik-Červonenkis) dimensijai, kuri, savo ruožtu, yra ne mažesnė, nei sistemos (DNT) parametrų skaičius. Testavimo klaida tada yra proporcinga VC dimensijos ir mokymo imties dydžio santykiui.

Generalizavimo klaidos minimizavimo požiūriu svarbu sudaryti su mažiausia klaida duomenų pasiskirstymą į klases modeliuojančią hipotezę. Tą hipotezę reikia parametrizuoti taip, kad ši minimizuotų testavimo klaidą turint tik mokymo aibę.

Toliau kalbama apie teoriją, kuri siekia pateikti šias sąlygas tenkinančios hipotezių klasės tam tikro didumo testavimo klaidos tikimybės įverčius. Čia pateikiami ELM veikimui paaiškinti reikalingi straipsnio [Bar98] rezultatai, bet jų įrodymai nenagrinėjami. Įrodymai pateikti minėtame straipsnyje. Be to, dalis šioje dalyje naudojamų žymėjimų (tų, kurie naudojami kalbant apie minėto straipsnio [Bar98] rezultatus) skiriasi nuo anksčiau naudotų. Visi tokie žymėjimai yra papildomai paaiškinti.

Sprendžiamas DNT sudarymo klasifikavimui uždavinys. Tariama, kad duomenų erdvėje išdėstyta duomenų aibė yra padalinta į poaibius-klases. DNT mokymo algoritmo užduotis – iš DNT modeliuojamos funkcijų-hipotezių klasės išrinkti tokią, kuri su minimalia klaida sukurtų duomenų aibės poaibių dengiančiuosius poaibius-klases. Generalizavimo efektyvumo požiūriu gali būti neparanku tiksliai modeliuoti mokymo aibės poaibius, nes testavimo aibės poaibiai gali nuo jų tam tikru dydžiu skirtis. Todėl mokymo užduotis yra suformuoti klasifikavimo hipotezę ne tik atsižvelgiant į mokymo aibės dėsningumus, bet ir iš šių sukurti prielaidas apie tikėtinus mokymo ir testavimo aibių neatitikimus. Vienas iš generalizavimo gerinimo būdų yra hipotezių apie duomenų poaibius sudarymas šių poaibių ribas apskaičiuojant su papildomu intervalu, kitaip tariant, skaičiuoti jų FS (angl. *fat-shattering*) dimensiją.

Neuroniniai tinklai gali apskaičiuoti realiąsias išvesties reikšmes. Tegul tinklo išvesties ženklas žymi duomenų įrašo klasę. Tada galima sudaryti tokį DNT mokymo algoritmą, kuris paslenka klasifikavimo rezultatus nuo 0 (reikšmių aibės vidurinės reikšmės – skiriamojo paviršiaus vidurio) per tam tikrą intervalą, taip pastorindamas atstumą tarp modeliuojamos duomenų erdvės poerdvių-klasių. Tokio mokymo algoritmo tikslas yra ne tik minimizuoti mokymo klaidą, bet ir maksimizuoti šį atstumą tarp modeliuojamų klasių (kitaip tariant, rasti geriausią mokymo algoritmo jautrumo ir

stabilumo derinį). Todėl testavimo klaidos tikimybės skaičiavimuose vietoje DNT modeliuojamos klasifikavimo hipotezių aibės VC dimensijos galima naudoti jos atmainą – FS dimensiją. Tada DNT testavimo klaidos tikimybė proporcinga FS dimensijos ir mokymo imties dydžio santykiui fat_H/m. Pati FS dimensija yra proporcinga tinklo gyliui (sluoksnių skaičiui) ℓ ir išvesties sluoksnio neuronų jungčių svorių normai A. FS dimensija nepriklauso nuo neuronų skaičiaus $\tilde{N}$.

DNT testavimo klaidos įvertinimui P. L. Bartlett pritaikė testavimo klaidos įvertį pagal Vapnik ir Červonenkis

$$\Pr\left(\text{er}_P(h) \leq \hat{\text{er}}_z(h) + \sqrt{\frac{1}{m}\left(D\left(\log\left(\frac{2m}{D}\right)+1\right) - \log\left(\frac{\delta}{4}\right)\right)}\right) = 1-\delta,$$

čia $\text{er}_P(h)$ yra testavimo naudojant hipotezę h, esant duomenų tikimybiniam skirstiniui P, klaidos įvertis, $\hat{\text{er}}_z(h)$ yra testavimo klaidos įvertis, kur z yra mokymo imtis, D – modeliuojamos hipotezių klasės VC dimensija, m – mokymo aibės dydis, $0 \leq \delta \leq 1$.

DNT modeliuojamų funkcijų-hipotezių aibės VC dimensija šiuose skaičiavimuose netinka, nes ji gali būti begalinė. Vietoje VC dimensijos P. L. Bartlett panaudojo FS dimensiją, kuri apibrėžiama panašiai į VC dimensiją, tačiau FS dimensijos skiriamasis paviršius į abi puses pastorinamas per intervalą γ, o pati FS dimensija tampa baigtine.

Jei $H : X \mapsto \mathbb{R}$ yra hipotezių aibė, X – duomenų aibė, tada H γ-suskaido taškų seką $(x_1, \ldots, x_m) \in X^m$ per $\gamma > 0$, jei $\exists h \in H$, kad $\forall b = (b_1, \ldots, b_m) \in \{-1, 1\}^m$ ir $r = (r_1, \ldots, r_m) \in \mathbb{R}^m$ yra teisinga $(h(x_i) - r_i)b_i \geq \gamma$. b_i čia atitinka taško x_i klasę, o r_i gali būti lygu 0. Tada hipotezių klasės H FS dimensija yra

$$\text{fat}_H(\gamma) = \max\left\{m : \exists x \in X^m, \text{ir } H\,\gamma - \text{suskaido}\, x\right\}.$$

Jei yra pseudometrinė erdvė (S, ρ) ir aibė $A \subseteq S$, tai aibė $T \subseteq S$ yra aibės A ϵ-dengiančioji aibė metrikos ρ atžvilgiu, jei tenkinama sąlyga $\forall a \in A, \quad \exists t \in T, \quad \rho(t, a) < \epsilon$. Tada $\mathcal{N}(A, \epsilon, \rho)$ yra mažiausios A ϵ-dengiančios aibės dydis.

Atstumas tarp hipotezių $f, g \in F$, kur $F : X \mapsto \mathbb{R}$, o $x_i \in X, i = 1, \ldots m$, pagal Čebyšovo metriką ρ apibrėžiamas taip

$$d_{\ell_\infty(x)}(f, g) = \max_i |f(x_i) - g(x_i)|.$$

Šis atstumas reikalingas įvertinti tikėtinam skirtumui tarp modeliuojamosios (pagal kurią bus sudaroma dengiančioji aibė) ir tikrosios (tačiau neišreikštos) duomenų pasiskirstymo f-jos.

Didžiausia dengiančioji aibė:

$$\max_{x \in X^m} \mathcal{N}(A, \epsilon, d_{\ell_\infty(x)}) = \mathcal{N}_\infty(A, \epsilon, m).$$

Toliau sudaroma hipotezės h reikšmių intervalą ribojanti funkcija

$$\pi_\gamma(\alpha) = \begin{cases} \gamma, & \text{jei } \alpha \geq \gamma \\ -\gamma, & \text{jei } \alpha \leq -\gamma \\ \alpha, & \text{jei } -\gamma < \alpha < \gamma \end{cases}$$

ir apibrėžiama $\pi_\gamma(H) = \{\pi_\gamma \circ h : h \in H\}$.

Tada apibrėžiamas testavimo klaidos įvertis pritaikant naują pagal FS dimensiją išreikštą dengiančiosios aibės dydį VC dimensijos formulėje. Hipotezės $h \in H$ testavimo klaidos tikimybė:

$$\mathrm{er}_P(h) < \hat{\mathrm{er}}_z^\gamma(h) + \sqrt{\frac{2}{m}\ln\left(\frac{2\mathcal{N}_\infty(\pi_\gamma(H), \gamma/2, 2m)}{\delta}\right)},$$

čia $0 \leq \delta \leq 1$.

Taigi, kuo mažiau prie dengiamosios prigludusi ir netikslesnė yra dengiančioji aibė, tuo didesnė ir teorinė klaidos tikimybė. (Dengiančiosios aibės tikslumas priklauso nuo pasirinktos hipotezės $h \in H$ ir FS dimensijos parametro γ reikšmės. Abu šie parametrai turi būti priderinami prie klasifikuojamų duomenų.)

Poskyryje apie klasikinę ELM schemą minėta, kad išvesties sluoksnio neuronų svoriai priderinami prie mokymo imties duomenų išsprendžiant tiesinių lygčių sistemą ir kad šis sprendinys minimizuoja mokymo klaidą. Pagal teoremos teiginį, jei kartu su mokymo klaida minimizuojama ir išvesties sluoksnio neuronų jungčių svorių norma, maksimizuojamas generalizavimo efektyvumas [Bar97; Bar98; HZS04].

Šie skaičiavimai, pritaikyti DNT, pateikia žemiau aprašytą rezultatą. Tegul H yra dvisluoksnių DNT su paslėptaisiais neuronais iš F klasė:

$$H = \left\{ \sum_{i=0}^{N} w_i f_i : N \in \mathbb{N}, f_i \in F, \sum_{i=0}^{N} |w_i| \leq A \right\},$$

F yra $f : X \mapsto \left[-\frac{M}{2}, \frac{M}{2}\right]$ aibė, o A yra išvesties sluoksnio neurono jungčių svorių absoliučių didumų riba. Tada

$$\mathrm{fat}_H(\gamma) \leq \frac{cM^2A^2d}{\gamma^2}\log^2\left(\frac{MAd}{\gamma}\right),$$

kur $d = \mathrm{fat}_F\left(\frac{\gamma}{32A}\right) \geq 1$, o c – konstanta.

Tada testavimo klaidos įvertis skaičiuojamas

$$\mathrm{er}_P(h) < \hat{\mathrm{er}}_z^\gamma(h) + \sqrt{\frac{c}{m}\left(\frac{A^2n}{\gamma^2}\right)\log\left(\frac{A}{\gamma}\right)\log^2 m + \log(1/\delta)}.$$

Jei įvesties duomenų aibė $X = \{x \in \mathbb{R}^n : \|x\|_\infty \leq B\}$, o sluoksnių skaičius yra ℓ, tada testa-

vimo klaida skaičiuojama

$$\mathrm{er}_P(h) < \hat{\mathrm{er}}_z^{\gamma}(h) + \sqrt{\frac{c}{m}\left(\frac{B^2(AL)^{\ell(\ell+1)}}{\gamma^{2\ell}}\right)\log n \log^2 m + \log(1/\delta)}.$$

Šie skaičiavimai su įrodymais pateikti [Bar98].

Dvisluoksnio DNT su neapibrėžtu paslėptojo sluoksnio dydžiu ir sigmoidine aktyvavimo funkcija, bei įvesties vektoriaus dimensija d, mokymo imties dydis proporcingas $A^2 d/\epsilon^2$, mokymo klaida α, testavimo klaida $\alpha + \epsilon$. Dvisluoksnio DNT su neapibrėžtu paslėptojo sluoksnio dydžiu ir sigmoidine aktyvavimo funkcija, bei duomenimis iš $[-B, B]^d$, mokymo imties dydis proporcingas $A^6 B^2 \log d/\epsilon^2$, mokymo klaida α, testavimo klaida $\alpha + \epsilon$.

2.3. Ankstesni tyrimai

Iki ELM pristatymo 2004 m. [HZS04] yra pasirodę dvi mašinų mokymo schemos, naudojančios dalies parametrų inicializavimą atsitiktiniais dydžiais. 1992 m. buvo pristatytas vieno paslėptojo sluoksnio tiesioginio sklidimo neuroninis tinklas su atsitiktiniais dydžiais inicializuojamais paslėptojo sluoksnio neuronų svoriais [SKD92]. Kita schema – RVFL (angl. *Random vector functional-link net*) [PPS94] pristatyta 1994 m., naudoja paslėptojo sluoksnio neuronus su atsitiktiniais svoriais, arba atsitiktiniais centrais, jei pasirenkamos RBF aktyvavimo funkcijos.

Nuo ELM šios schemos skiriasi tuo, kad jose nebuvo pasiūlyti visiškai atsitiktiniai paslėptojo sluoksnio neuronai. RBF RVFL atveju panaudoti atsitiktiniai centrai w_j, bet įtakos veiksniai (angl. *impact factor*) b_j apskaičiuojami kaip centrų ir įvesties x_i funkcija. Sigmoidinis RVFL tinklų tipas atitinkamai naudoja atsitiktinius svorius $\mathbf{w}_j$, bet postūmiai b_j apskaičiuojami pagal mokymo imties duomenis x_i ir svorius $\mathbf{w}_j$ [Hua15].

Schmidt [SKD92] tirti tinklai su iš dalies atsitiktiniais neuronais buvo griežtai dvisluoksniai su postūmiais išvesties sluoksnyje (todėl neatitinka SLFN apibrėžimo). Jų išraiška

$$f_{\tilde{N}}(\mathbf{x}_i) = \sum_{j=1}^{\tilde{N}} \mathbf{v}_j g_{\text{sig}}(\mathbf{a}_j \cdot \mathbf{x}_i + b_j) + b.$$

RVFL tinklai apibrėžti su jungtimi iš įvesties sluoksnio į išvesties sluoksnį

$$f_{\tilde{N}}(\mathbf{x}_i) = \sum_{j=1}^{\tilde{N}} \mathbf{v}_j g_{\text{sig arba RBF}}(\mathbf{w}_j, b_j, \mathbf{x}_i) + \alpha \cdot \mathbf{x}_i.$$

Tuo tarpu ELM schema naudoja SLFN tinklą

$$f_{\tilde{N}}(\mathbf{x}_i) = \sum_{j=1}^{\tilde{N}} \mathbf{v}_j g(\mathbf{w}_j, b_j, \mathbf{x}_i).$$

ELM atsitiktiniu būdu inicializuojami įvairių tipų paslėptojo sluoksnio neuronai. Yra tyrimų

su RBF, sumuojančiais neuronais, Fourier eilutėmis ir kt. ELM neuronai nepriklauso nuo mokymo duomenų ir nuo kitų neuronų parametrų. Be to, kiekvienas neuronas gali naudoti skirtingą nuo kitų aktyvavimo funkciją. Taip pat, ELM tinkluose atsitiktinumas paskirstytas keliais būdais:

1. paslėptojo sluoksnio neuronai inicializuojami atsitiktiniais parametrais;
2. jungtys tarp įvesties ir paslėptojo sluoksnio gali būti generuojamos atsitiktinai;
3. paslėptojo sluoksnio neuronas pats gali būti sudarytas iš kito potinklio.

Minėtieji ankstesni tyrimai šių sąlygų netenkina [Hua15].

2.4. ELM modifikacijos

2.4.1. Taikymai įvairioms DNT architektūroms

Klasikinė ELM schema apibrėžta su vieno paslėptojo sluoksnio tiesioginio sklidimo neuroniniu tinklu (angl. *Generalized single hidden layer feedforward network, Generalized SLFN*), bet vėliau pristatyti ir jos taikymai daugiau nei vieno paslėptojo sluoksnio tiesioginio sklidimo neuroniniams tinklams (angl. *Generalized multi-hidden layer feedforward network, Generalized MLFN*). ELM schemos naudojamas tinklas nuo įprastinių vieno paslėptojo sluoksnio tiesioginio sklidimo tinklų skiriasi išvesties neuronų forma — ELM išvesties neuronai neturi poslinkių, kai, tuo tarpu, tradicinių tinklų išvesties neuronai juos gali turėti.

ELM taikymai daugiasluoksnėms tinklų topologijoms pasižymi tokiomis savybėmis:

1. tarp sluoksnių informacija turi būti perduodama tikslingai, ne atsitiktiniu būdu, t.y. kiekvienas toks paslėptas sluoksnis gali turėti įterptą ELM, atliekančią apibrėžtą užduotį (pvz.: duomenų kompresiją, požymių išmokimą (angl. *feature learning*), klasterizavimą, regresiją ar klasifikavimą), kurią atlikęs perduoda duomenis kitiems sluoksniams toliau apdoroti. Perduodant informaciją tarp sluoksnių atsitiktiniu būdu nebūtų galima prognozuoti kokie veiksmai su ja bus atliekami;
2. neuronų parametrai gali būti vieną kartą priskiriami ir po to nebekeičiami:
   (a) dalis neuronų gali būti generuojami atsitiktinai;
   (b) po jų esančių kitų sluoksnių neuronų parametrai gali būti apskaičiuojami kaip anksčiau esančių atsitiktinių neuronų funkcija.
3. tokiu būdu vieno paslėptojo sluoksnio ELM tinklus galima naudoti kaip didesnės schemos modulius, atliekančius jiems priskirtas (neatsitiktines) užduotis. Kitaip tariant, iš ELM modulių galima konstruoti sutvarkytas sistemas;
4. ELM galima tokiose sistemose naudoti kartu su kitomis mašinų mokymo schemomis, arba (ELM požymių išmokimo ar klasterizavimo atveju) kaip jungiančiąsias grandis tarp tokių skirtingų schemų [Hua15].

2.4.2. Paslėptojo ir išvesties sluoksnių neuronų tipai

Didžiausias skirtumas tarp dviejų ELM schemos DNT sluoksnių yra tai, kad paslėptojo sluoksnio neuronai gali (bet neprivalo) būti inicializuoti atsitiktiniais jungčių svoriais ir poslinkiais. Išvesties sluoksnio neuronų jungčių svoriai yra randami sprendžiant mokymo klaidos minimizavimo uždavinį. Jo ypatumas — minimumas randamas neiteraciniu skaičiavimu, skirtingai nuo pvz., gradientinio nusileidimo metodo. Kitas išvesties sluoksnio neuronų ypatumas yra tai, kad jie neturi poslinkių. Aktyvavimo funkcijos priklauso nuo konkretaus taikymo pobūdžio (nuo jų pavidalo priklauso generalizavimo rezultatas), bet bazinis ELM variantas apibrėžiamas su sigmoidine aktyvavimo funkcija (logistine). Skirtinguose sluoksniuose (netgi skirtinguose to paties sluoksnio neuronuose) gali būti naudojamos skirtingos aktyvavimo funkcijos, jei to reikalauja taikymo sąlygos.

2.4.3. Klasifikavimas ir regresija

Čia apžvelgiamos populiariausios klasifikavimui ir regresijai skirtos ELM modifikacijos.

ELM stabilumo didinimo tyrimai. ELM mokymo algoritmą galima pritaikyti spręsti reguliarizuotą mokymo klaidos minimizavimo uždavinį

$$\min_{\mathbf{V}\in\mathbb{R}^{\tilde{N}\times m}} \|\mathbf{V}\|_p^{\sigma_1} + C\|\mathbf{HV}-\mathbf{T}\|_q^{\sigma_2},$$

čia $\sigma_1 > 0, \sigma_2 > 0$ yra atitinkamos normos laipsnio rodiklis, o $p, q = 0, \frac{1}{2}, 1, 2, \ldots, +\infty$ yra normos tipas. Pvz., kai $\sigma_1 = \sigma_2 = p = q = 2$

$$\min_{\mathbf{V}\in\mathbb{R}^{\tilde{N}\times m}} \frac{1}{2}\|\mathbf{V}\|^2 + \frac{C}{2}\|\mathbf{HV}-\mathbf{T}\|^2.$$

[HHS+15] pateikti sprendimai su pilno ir nepilno rango paslėptojo sluoksnio matricomis.

ELM mokymo algoritmas, prieš apskaičiuodamas išvesties sluoksnio parametrus turi rasti atvirkštinę paslėptojo sluoksnio matricą. Tam, kad ši būtų teigiamai apibrėžta, galima pasinaudoti reguliarizavimo koeficientu C, tačiau jį didinant mažėja mokymo algoritmo jautrumas. Tam DNT pritaikymas prie mokymo imties duomenų išplečiamas ir į paslėptojo sluoksnio sudarymo žingsnį. Vietoje atsitiktinio paslėptojo sluoksnio generavimo pasiūlytas paslėptojo sluoksnio matricos parametrų parinkimo algoritmas [WCY11]. Įrodyta, kad naudojant RBF aktyvavimo funkcijas, taip sudaryta paslėptojo sluoksnio matrica bus pilno eilučių arba stulpelių rango [HHS+15].

ELM kompaktiškumo klausimas. Dėl atsitiktinio paslėptojo sluoksnio parametrų pobūdžio ELM reikalauja didesnio neuronų skaičiaus už kai kuriuos klasikinius algoritmus (pvz., BP). Dėl to didėja skaičiavimų apimtis testuojant. Vienas sprendimas — palaipsniui didinti paslėptojo sluoksnio dydį, prijungiant neuronus iš atsitiktinai inicializuotų neuronų aibės, kurių išvesties svoriai yra tinkami, o netinkamų neįtraukiant. Pasiūlyti tokio algoritmo variantai: I-ELM (angl. *incremental ELM*) — griežtai inkrementinis algoritmas — pridedant naują neuroną, esami paslėptojo

sluoksnio neuronai fiksuojami ir daugiau nebekeičiami. CI-ELM (angl. *convex incremental ELM*) modifikuoja šią ankstesniojo algoritmo sąlygą ir leidžia keisti anksčiau parinktus neuronus [HC07], B-ELM (angl. *bidirectional ELM*) pridedamus nelyginio numerio neuronus inicializuoja atsitiktiniais parametrais (kaip ir ELM), bet lyginiai neuronai inicializuojami proporcingai iki jų pridėjimo sukonstruoto DNT klaidos funkcijos reikšmei [YWY12], OP-ELM (angl. *optimally pruned ELM*) konstruoja paslėptąjį sluoksnį taip pat, kaip ir klasikinis ELM, bet vėliau atliekamas paslėptojo sluoksnio genėjimas, kurio metu pagal naudingumą ranguojami neuronai ir pašalinami išėjime darantys didžiausią klaidą [MSB+10]. AG-ELM (angl. *adaptive growth ELM*) algoritmas mokymo metu konstruoja paslėptąjį sluoksnį jį dinamiškai didindamas ir mažindamas [ZLH+12].

Taip pat pasiūlytas algoritmas, kuris optimizuoja paslėptojo sluoksnio parametrus gradientinio nusileidimo būdu minimizuodamas bendrą tinklo kvadratinę klaidą. Taikant šį algoritmą reikia naudoti diferencijuojamas paslėptojo sluoksnio aktyvavimo funkcijas [YD12].

Keli ELM mokymo algoritmai konstruoja išretintą (angl. *sparse*) paslėptojo arba išvesties sluoksnio matricą [HHS+15]. Vienas jų — OS-ELM — sumažina reikšmingų parametrų skaičių, vietoj optimizavimo apribojimo-lygybės $\mathbf{h}(\mathbf{x}_i)\mathbf{v} = \mathbf{t}_i - \mathbf{e}_i$ naudodamas nelygybę $\mathbf{t}_i\mathbf{h}(\mathbf{x}_i)\mathbf{v} \geq 1 - \mathbf{e}_i$ [BHW+14].

ELM variantai stochastiniam (angl. *online sequential*) mokymui. Vienas jų — OS-ELM (angl. *online sequential ELM*) [LHS+06]. Tinka mokymui pavieniais vektoriais (angl. *one-by-one*) arba jų rinkiniais (angl. *chunk-by-chunk*). Paslėptojo sluoksnio matrica $\mathbf{H}$ pildoma inkrementiškai, $k+1$-osios iteracijos metu išvesties sluoksnį $\mathbf{V}$ perskaičiuojant kaip $k+1$-osios iteracijos paslėptojo sluoksnio matricos ir k-osios iteracijos išvesties sluoksnio funkciją. FOS-ELM (angl. *forgetting OS-ELM*) algoritmas skirtas lokalių laiko atžvilgiu įvesties dėsningumų išmokimui. Praėjus tam tikram laikui, anksčiau išmoktus dėsningumus keičia naujesni. Yra variantai stacionariems ir nestacionariems duomenims [HHS+15].

ELM mokymo algoritmo modifikacija nesubalansuotiems duomenims. W-ELM (angl. *weighted ELM*) [ZHC13] pasiūlytas naudoti su mokymo imtimis, sudarytomis iš nevienodo didumo klasių arba nevienodo svorio (svarbos) įrašų. Šis algoritmas su mokymo įrašais susieja baudos koeficientus C_i (didesnės reikšmės mažumos klasėms, mažesnės – daugumos) ir sprendžia reguliarizuotą mažiausių kvadratų optimizavimo uždavinį

$$\min_{\mathbf{V}\in\mathbb{R}^{\tilde{N}\times m}} \frac{1}{2}\|\mathbf{V}\|^2 + \frac{1}{2}\sum_{i=1}^{N} C_i\|\mathbf{HV} - \mathbf{T}\|^2$$

Yra šio algoritmo modifikacija sumažinti taškų-atsiskyrėlių įtakai DNT mokymui. Kitame tyrime šis algoritmas panaudotas kaip pagrindas iš dalies prižiūrimam (angl. *semi-supervised*) mokymui įgyvendinti [HHS+15].

ELM variantas triukšmingiems duomenims ir duomenims su trūkstamais požymiais. FIR-ELM (angl. *finite impulse response filter ELM*) panaudoja paslėptąjį sluoksnį kaip duomenų

parengimo (angl., *preprocessing*) žingsnį triukšmui iš signalo pašalinti [MLW+11]. Pasinaudojama tokia išvesties sluoksnio jautrumo įvesties pokyčiams priklausomybe

$$\frac{\|\Delta \mathbf{V}\|}{\|\mathbf{V}\|} \approx \frac{\|\Delta \mathbf{V}\|}{\|\mathbf{V} + \Delta \mathbf{V}\|} \leq \|\mathbf{H}^{+}\|\|\Delta \mathbf{H}\| = \bar{\kappa}(\mathbf{H}) \frac{\|\Delta \mathbf{H}\|}{\|\mathbf{H}\|},$$

kur $\bar{\kappa}(\mathbf{H}) = \|\mathbf{H}^{+}\|\|\mathbf{H}\|$.

Kitas, DFT-ELM, algoritmas naudoja diskrečiąsias Fourier transformacijas triukšmui iš įvesties signalo pašalinti.

Keli šios grupės algoritmai — IRWLS-ELM, MLTS-ELM ir RMLTS-ELM — skirti įrašų-atsiskyrėlių įtakai mokymo rezultatams sumažinti [HCS13]. Šie algoritmai mokymo įrašams priskiria svorius, o tada apmoko tinklą naudodami aukščiau aprašytą W-ELM algoritmą. Įrašams-atsiskyrėliams suteikus mažus svorius, apribojamas jų reikšmingumas.

TROP-ELM algoritmas skirtas apdoroti duomenims su trūkstamais požymiais [YME+13]. Mokymo duomenys transformuojami į jų atstumų matricą, trūkstami požymiai įrašuose užpildomi kuriuo nors jų vidurinės reikšmės variantu. Po to naudojamas mokymo algoritmas su paslėptojo sluoksnio neuronų rangavimu ir genėjimu, siekiant padidinti paslėptojo sluoksnio kompaktiškumą [HHS+15].

Inkrementiškai paslėptąjį sluoksnį konstruojantys ELM mokymo algoritmai. Pirmasis toks algoritmas buvo aukščiau minėtas I-ELM (angl. *incremental ELM*) [HC07]. Po to pasiūlyti EI-ELM (angl. *enhanced I-ELM*) [HC08] ir EM-ELM (angl. *error minimized ELM*) [FHL+09]. I-ELM paslėptąjį sluoksnį sudaro kiekvienoje iteracijoje pridėdamas po vieną neuroną. Šis algoritmas savo grupėje išsiskiria tuo, kad gali naudoti įvairesnes aktyvavimo funkcijas (reikalaujama, kad jos būtų tankios L^2 erdvėje ir tolydžios intervaluose). EI-ELM konstruoja paslėptąjį sluoksnį pridėdamas po vieną neuroną iš sugeneruotos neuronų grupės, pasirinkdamas tą, kuris labiausiai sumažina mokymo klaidą. Taip gaunamas kompaktiškesnis paslėptasis sluoksnis negu sukonstruotas I-ELM algoritmo. EM-ELM konstruoja paslėptąjį sluoksnį iš generuojamų neuronų grupių iš $k \geq 1$ neuronų. Neuronų skaičius k skirtingose iteracijose gali būti skirtingas. EM-ELM algoritmo pradiniai apribojimai — maksimalus paslėptojo sluoksnio dydis ir leistinas mokymo klaidos dydis [HHS+15].

ELM grupės. EOS-ELM (angl. *ensemble of OS-ELM*) [LSH09] generalizavimo rezultatą apskaičiuoja kaip K nepriklausomai aukščiau minėtu OS-ELM algoritmu apmokytų tinklų rezultatų vidurkį. V-ELM (angl. *voting based ELM*) [CLH+12] apmoko K nepriklausomų tinklų ir išrenka daugumos rezultatą [HHS+15].

ELM taikymas rangavimo uždaviniams spręsti. ELM schema pritaikyta dviems mokymo metodams: taškiniam (angl. *pointwise RankELM*) ir poriniam (angl. *pairwise RankELM*). Taškinio mokymo atveju tinklui pateikiami duomenys, sudaryti iš užklausų ir dokumentų porų, bei jų atitikimą žyminčių tikslinių reikšmių. Tinklas mokomas aproksimuoti šias tikslines reikšmes. Porinio rangavimo esmė – sudaryti dokumentų atitikimo užklausai įverčius, tuomet sudaryti visas

šių įverčių poras ir galiausiai tuos įverčius porose palyginti. Jei poroje kuris nors įvertis svaresnis (pvz., didesnis), laikoma, kad tas dokumentas šioje poroje labiau atitinka užklausą. Tinklas mokomas modeliuoti šių palyginimų rezultatus [HHS+15; ZH14].

2.4.4. ELM mokymas naudojant dalį nežymėtų mokymo duomenų (angl. *semi-supervised learning*)

Pristatytas mokymo algoritmas, kuris iteratyviai papildo mokymo imtį ankstesnėje iteracijoje didžiausia tikimybe klasifikuotais nežymėtais įrašais [LZX+13]. Kitas sprendimas — SS-ELM (angl. *semi-supervised ELM*) [HSG+14] klasikinį ELM mokymo algoritmą papildo reguliarizavimu ant daugdaros (sudarytos iš žymėtų ir nežymėtų mokymo duomenų) paviršiaus [HHS+15].

2.4.5. Mokymasis be mokytojo

Netiesinis duomenų dimensijos mažinimas su duomenų projektavimu į mažesnės dimensijos erdvę (angl. *embedding*) ir duomenų klasterizavimas projekcinėje erdvėje. Šiai užduočiai atlikti pasiūlytas US-ELM (angl. *unsupervised ELM*) algoritmas [HSG+14], pagrįstas reguliarizuotu mokymo klaidos minimizavimu ant daugdaros (sudarytos iš mokymo duomenų) paviršiaus. Minimizuojamas reiškinys

$$\min_{\mathbf{V}\in\mathbb{R}^{\tilde{N}\times m}} \|\mathbf{V}\|^2 + \lambda \mathrm{Tr}(\mathbf{V}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\mathbf{V}),$$

čia $\mathbf{L}$ yra mokymo duomenų Laplace'o matrica, apskaičiuojama taip $L = D - A$, kur $A = [a_{ij}]$ yra duomenų gretimumo (angl. *similarity*) matrica, a_{ij} yra koks nors įrašų $\mathbf{x}_i$ ir $\mathbf{x}_j$ gretimumo indikatorius (pvz. $\mathbf{x}_i$ ir $\mathbf{x}_j$ yra tarp vienas kito k artimiausių kaimynų), o $D_{ii} = \sum_{j=1}^{N} \mathbf{a}_{i,j}$ matrica, kurios įstrižainė yra duomenų įrašo artimumo kitiems įrašams įvertis (pvz., artimiausių kaimynų skaičius) [HSG+14]. Skaičiuojant $\mathrm{Tr}(\mathbf{V}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\mathbf{V})$ gaunama ELM atliktos duomenų projekcijos suminė dispersija, kurią minimizuojant randama projekcija, kurioje duomenys yra labiausiai susitelkę į klasterius. Toliau galima naudoti pasirinktą klasterizavimo metodą [HHS+15].

Reprezentacinis mokymas. Užduotis, kuriai atlikti paprastai naudojami (angl. *deep belief*) tinklai ir Boltzmann mašinos. Kiekvienas sluoksnis išskiria tam tikrus duomenų požymius (angl. *feature representation*) ir konvejerio principu perduoda juos apdoroti gilesniam sluoksniui, kol išvesties sluoksnis apskaičiuoja galutinį generalizavimo rezultatą [HHS+15].

2.4.6. ELM taikymas reikšmingiausių įrašo požymių išrinkimui

FS-ELM (angl. *feature selection ELM*) algoritmu minimizuojant mokymo klaidos funkciją

$$\min_{\mathbf{s},\mathbf{V}} \frac{1}{N}\sum_{i=1}^{N}(\mathbf{T}_i - f(\mathbf{s}^T\mathbf{x}_i;\mathbf{V}))^2,$$

kur $\|\mathbf{s}\|_0 = d_s \leq d, \quad \mathbf{s} \in \{0,1\}^d$, sudaromi reikšmingų požymių rinkiniai. Čia $\mathbf{s}$ naudojamas požymių poaibiui išrinkti. Ši klaidos funkcijos išraiška nėra diferencijuojama, todėl praktiškai

naudojama tolydi jos modifikacija, kuri sprendinio paiešką atlieka pagal mokymo klaidos funkcijos gradientą. Čia iškylanti lokalių minimumų problema sprendžiama vykdant paiešką daugiau nei vieną kartą, algoritmą inicializuojant atsitiktiniais pradiniais parametrais. Testavimo klaida įvertinama pagal kryžminio testavimo metodą [HHS+15].

2.4.7. ELM realizacijos

Lygiagrečioji realizacija. Sudarytos realizacijos bendros atminties lygiagrečiųjų skaičiavimų sistemoms (CPU ir GPU), taip pat paskirstytiesiems skaičiavimams (MapReduce metodas ir kt.). ELM mokymo fazė čia išsiskiria tuo, kad mokymo duomenis skaičiavimų mazgams užtenka paskirstyti vieną kartą ir prie tų pačių duomenų mokymo sesijoje grįžti nebereikia. Taip pat nereikalingas grįžtamasis ryšys iš gilesniųjų sluoksnių į aukštesniuosius. Kiekvieno ELM schemos sluoksnio neuronai yra tarpusavyje nepriklausomi, jų individualūs skaičiavimai gali būti atliekami atskirai ir lygiagrečiai, užtenka sinchronizuoti skaičiavimų rezultatų perdavimą tarp sluoksnių. Testavimo metu ELM lygiagretaus vykdymo savybės nesiskiria nuo tradicinių DNT [HHS+15].

Aparatinė realizacija. Nesudėtingus ir gerai lygiagretinamus ELM schemos skaičiavimus galima įgyvendinti aparatiškai. Sukurtos realizacijos programiškai apibrėžiamos logikos įrenginiams (CPLD (angl. *Complex Programmable Logic Device*) ir FPGA (angl. *Field Programmable Gate Array*)). Taip pat sukurta ir realizacija specializuotiems, biologinių neuronų veikimą modeliuojantiems įrenginiams (angl. *spiking neural circuits*) [HHS+15].

3. Tyrimo ataskaita

3.1. Tyrimo metodai

Metodinė darbo dalis nustato mašinų mokymo naudojimo principus tekstų klasifikavimui šio tyrimo kontekste. Iš pradžių rašoma apie tekstinių duomenų parengimą klasifikavimui, tuomet pristatomi šiame tyrime naudojami duomenų rinkiniai. Toliau rašoma kaip jiems klasifikuoti galima naudoti ELM ir R-ELM. Po to rašoma apie tekstų klasifikavimo rezultatų vertinimo būdus.

3.1.1. Duomenų parengimas, požymių identifikavimas ir išrinkimas

Teksto struktūra svarbi pasirenkant pirminio apdorojimo pobūdį ir reikšminių požymių identifikavimo (angl. *feature selection*) ir išrinkimo (angl. *feature extraction*) metodus. Galima skirti du tekstinių duomenų pavidalus:

1) nestruktūruotas tekstas. Semantinių ryšių paiešką tokiu atveju reikia atlikti tiesiogiai pačiame tekste, nes jis nėra įrėmintas kokia nors papildoma tarnybine informacija (pvz., HTML žymomis). Tokius duomenis galima atvaizduoti į VSM (angl. *vector space model*) [ZWB+11] ir toliau apdoroti skaitiniais metodais;

2) struktūruotas tekstas (pvz., su HTML, XML žymomis, programų kodas, teisės aktai ir kt.). Iš anksto neapdorota tarnybinė informacija gali trukdyti semantinei paieškai, bet galima pasinaudoti duomenų išreikštiniu struktūravimu identifikuojant ir išrenkant informatyvius požymius, o mažareikšmius palikant jau šiame etape [ZWB+11], tada vėl atlikti atvaizdavimą į kokį nors VSM variantą, ar kitą, skaitiniams metodams tinkamą reprezentaciją.

Nustačius teksto struktūros savybes galima imti spręsti klausimus apie duomenų parengimą tyrimui:

1) kokią semantinę tekstinių duomenų reprezentaciją būtų naudingiausia sudaryti. Efektyvios semantinės teksto reprezentacijos privalumai — duomenų atributų informatyvumo teksto klasės nustatymui maksimizavimas ir galimybė minimizuoti duomenų dimensiškumą. Trūkumai — pasirinkus turimiems duomenims netinkamą reprezentaciją ji gali trukdyti identifikuoti informatyviausius požymius, o pasirinkus ypač sudėtingą atvaizdavimą — užduoties automatinis atlikimas gali tapti daug sudėtingesnis už patį klasifikavimą. Vienas paprasčiau automatizuojamų ir dažnai naudojamų būdų yra LSA (angl. *Latent Semantic Analysis*) [ZQL13], sudarytas iš VSM (angl. *Vector Space Model*) derinio su PCA (angl. *Principal Component Analysis*) ar kokiais nors kitais duomenų dimensiškumo mažinimo būdais;

2) kaip parengti duomenų atributus. Keli iš galimų būdų:

    1) žodžių ar kitų teksto vienetų dažnis tf (angl. *term frequency*). Sudaromas tekstų rinkinio D žodynas T. Kiekvienas rinkinio dokumentas atvaizduojamas į vektorių $d_i = [d_{i1}, \ldots, d_{im}]$, kur $\text{tf}(d_i, t_j) = d_{ij}$, $d_i \in \text{D}$, $t_j \in \text{T}$, $j = 1, \ldots, m$, $m = |\text{T}|$. Čia $\text{tf}(d_i, t_j) \in \{0, \mathbb{N}\}$ yra žodžio t_j pasikartojimų dokumente d_i skaičius [ZWB+11];

2) tarpusavio informacija mi (angl. *mutual information*). Naudojantis šiuo rodikliu galima įvertinti žodžių, kaip duomenų vektorių atributų informatyvumą. Kuo didesnė atributų tarpusavio informacija, tuo labiau koreliuoti jų pasikartojimai dokumentų rinkinyje ir tuo labiau verta kurio nors jų atsisakyti, jei siekiama minimizuoti duomenų dimensiškumą. Iš kitos pusės, kuo didesnė atributo koreliacija su dokumento klase, tuo jis informatyvesnis ir vertingesnis klasifikuojant.

$$I(D_k; D_l) = \sum_{d_p \in D_k} \sum_{d_q \in D_l} p(d_{pr}, d_{qs}) \log_b \frac{p(d_{pr}, d_{qs})}{p(d_{pr}) p(d_{qs})},$$

Dažniausiai logaritmo pagrindas $b \in \{2, e, 10\}$. r ir s, $r \neq s$ čia yra atributo pozicija duomenų vektoriuje. Diskretaus arba nežinomo tolydaus skirstinio atveju duomenų vektorių atributus galima suskirstyti į reikšmių intervalus, o žinomo tolydaus skirstinio atveju vietoje čia pavaizduoto sumos skaičiavimo galima integruoti pagal atributų skirstinių tankių funkcijas;

3) χ^2 statistika gali būti naudojama panašiai į tarpusavio informacijos įvertį. Suskaičiuojamas vidutinis dokumentų rinkinio įrašų atributų reikšmių nuokrypis nuo jų reikšmių vidurkio

$$\chi^2 = \sum_{d_{pj} \in D_k} \frac{(d_{pj} - \overline{d_{pj}})^2}{\overline{d_{pj}}}.$$

Čia

$$\overline{d_{pj}} = \frac{1}{|D_k|} \sum_{d_{pj} \in D_k} d_{pj}.$$

Kuo didesnis yra nuokrypis, tuo informatyvesnis yra atributas skirstant rinkinio dokumentus į klases;

4) informacijos prieaugis ig (angl. *information gain*). Matuoja dokumento $d_i \in \mathrm{D}$ priklausymo klasei $c_k \in \mathrm{C}$ entropijos dydžio H sumažėjimą kurio nors žodžio $t_j \in \mathrm{T}$ atžvilgiu.

$$\mathrm{ig}(\mathrm{D}, t_j) = H(\mathrm{D}) - \sum_{d_{ij}} \frac{|D_j|}{|\mathrm{D}|} H(D_j),$$

kur $D_j = \{d_i \in \mathrm{D} \mid \mathrm{tf}(d_i, t_j) = d_{ij}\}$. Kuo didesnė žodžio t_j informacijos prieaugio reikšmė dokumentų rinkinio D atžvilgiu, tuo didesnė šio žodžio reikšmė sprendžiant tekstų klasifikavimo uždavinį;

5) VSM reprezentacija gali būti sudaroma ir suskaičiuojant standratizuotą $\mathrm{tfidf}(d_i, t_j)$ — žodžio t_j reikšmingumą dokumento d_i atžvilgiu (kartu ir dokumentų rinkinio atžvilgiu):

$$\mathrm{tfidf}(d_i, t_j) = \mathrm{tf}(d_i, t_j)\, \mathrm{idf}(d_i, t_j),$$

kur

$$\mathrm{idf}(d_i, t_j) = \log \frac{N}{\mathrm{df}(t_j)},$$

$N = |\mathrm{D}|, \mathrm{df}(t_j) = |\{d_{ij} > 0\}|, d_{ij} = \mathrm{tf}(d_i, t_j), d_i \in \mathbf{D}, t_j \in \mathbf{T}$ [ZQL13]. tfidf dydis yra žodžio dažnio dokumente ir jo retumo dokumentų rinkinyje jungtinis įvertis. Kuo žodis dokumente dažnesnis, tuo labiau jis tam dokumentui ir dokumentų klasei būdingas. Kuo žodis yra dokumentų rinkinyje retesnis, tuo labiau jo pasitaikymai informatyvūs, tuo labiau jis būdingas dokumentams, kuriuose yra sutinkamas, o kartu ir tų dokumentų klasei. Tuomet sudaromi dokumentų tfidf įverčių vektoriai $d_i = (d_{i1}, \ldots, d_{im})$, kur

$$d_{ij} = \frac{\mathrm{tfidf}(\mathrm{d_i}, \mathrm{t_j})}{\sqrt{\sum_j^m \mathrm{tfidf}(d_i, t_j)^2}}.$$

Galiausiai iš d_i vektorių galima sudaryti matricą $\mathbf{F} = (d_1, \ldots, d_n)^{\mathrm{T}} \in \mathbb{R}^{n \times m}$, tuomet rasti jos tikrines reikšmes ir vektorius. Atrinkus pasirinktą skaičių didžiausias tikrines reikšmes atitinkančių tikrinių vektorių galima atlikti duomenų dimensijos sumažinimą (PCA metodas). Tam tinka matricų SVD (angl. *Singular Value Decomposition*) faktorizavimas $\mathbf{F} = \mathbf{U} \times \Sigma \times \mathbf{V}^{\mathrm{T}}$. $\mathbf{F}_k = \mathbf{U}_k \times \Sigma_k \times \mathbf{V}_k^{\mathrm{T}}$ yra matricos $\mathbf{F}$ pirmosios k pagrindinės komponentės. Tada sumažintos dimensijos dokumento vektorius gaunamas $\hat{d_i} = d\mathbf{U}_k\Sigma_k^{-1}$ [ZQL13].

ELM mokymo algoritmo spartos ir naudojimo paprastumo kaina — generalizavimo metu jam reikia didesnio paslėptojo sluoksnio negu, pvz., BP. Skaičiavimų apimtis lyginant su BP apmokytu DP išauga testavimo metu ir generalizuojant, ypač apdorojant daugiamačius duomenis. Tekstų klasifikavimas ir pasižymi duomenų daugiamatiškumu [ZQL13]. Naudojant paprastą žodžių dažnių tf įvertį duomenų įrašų atributų sudarymui PCA metodas gali būti neefektyvus mažinant duomenų dimensiškumą, nes žodžių dažnių reikšmės visame dokumentų rinkinyje gali būti pasiskirsčiusios nedideliame intervale, ir tuomet gali nebūti žymiai didesnes dispersijos reikšmes turinčių komponenčių, kurias reikėtų pasirinkti projekcijos atlikimui, ir mažesnes, kurių projekcijai galima nepasirinkti.

Klasifikuojant tekstus gali pasitaikyti, kad neatlikus semantinės analizės ir dimensiškumo mažinimo, duomenų atributų skaičius (vektorių dimensiškumas) viršija įrašų (vektorių) skaičių. Pvz., šis tyrimas atliktas su mokymo duomenų rinkiniais, kurių atributų skaičius — 10-100 tūkst., o įrašų skaičius — 3-12 tūkst. Tai, atsižvelgiant į mokymo algoritmo ypatumus, gali lemti nepakankamai apibrėžto modelio sudarymą. Didelis duomenų dimensiškumas lyginant su imties dydžiu sumažina ELM mokymo algoritmo stabilumą, todėl rekomenduotina naudoti reguliarizuojantį klaidos minimizavimo funkcijos apribojimą. Tam yra skirta ELM modifikacija — R-ELM [ZQL13], bet ir ji negarantuoja optimalaus generalizavimo, nes paslėptojo sluoksnio dydžio parinkimą palieka schemos naudotojo nuožiūrai. (Su šiame tyrime naudotais duomenimis geriausi testavimo rezultatai pasiekti esant paslėptojo sluoksnio neuronų skaičiaus santykiui su mokymo įrašų skaičiumi $\sim 2/5$). Kitą išvesties sluoksnio svorių normos minimizavimo būdą naudoja OP-ELM schema. Šį metodą taikant reikia ieškoti generalizavimo stabilumo ir tinklo informacinės talpos — paslėptojo sluoksnio dydžio — pusiausvyros. OP-ELM schemos naudojama nenaudingų neuronų genėjimo procedūra (sudaryta iš MRSR (angl. *Multi-Response Sparse Regression*) algoritmu atliekamo neuronų rangavimo pagal koreliacijos su mokymo tikslų reikšmėmis absoliutųjį didumą ir LOO (angl. *Leave One Out*) neuronų poaibio atrinkimo) viena vertus, minimizuoja neuronų skaičių ir didina

generalizavimo stabilumą, kita vertus, ji reikalauja didelių skaičiavimų ir dėl to netinka apdoroti didelės dimensijos duomenims. Yra būdų OP-ELM naudojamo atvirkštinės matricos skaičiavimo sudėtingumą nuo kubinio sumažinti iki kvadratinio (kai matricos tenkina tam tikrus reikalavimus [Dic98]), bet šiame darbe toks tyrimas nebuvo atliktas. Be to, paslėptojo sluoksnio dydžio parinkimo OP-ELM neatlieka, o nuo jo labiausiai priklauso ir bendra tinklo informacinė talpa, ir atskirų neuronų efektyvumas [CL15]. Panašiais privalumais ir trūkumais pasižymi ir iteracijomis paslėptąjį sluoksnį didinančios schemos, apie kurias rašyta ankstesniame skyriuje.

Siekiant sutrumpinti mokymo laiką galima sumažinti skaičiavimų apimtį panaudojant išretintas (angl. *sparse*) duomenų reprezentacijas ir atitinkamus algoritmus — (angl. *sparse ELM*) [BHW+14]. Atliekant šį tyrimą buvo pastebėtas mokymo ir testavimo algoritmų greitaveikos padidėjimas vien dėl išretintų matricų naudojimo, net neatlikus jokių papildomų mokymo schemų modifikacijų. Toks rezultatas pasiektas dėl santykinio tekstinių duomenų retumo (99,753% r8 ir 99,853% ng20 atributų reikšmių yra nuliai (r8 ir ng20 duomenų rinkiniai aprašyti 3.2. poskyryje)) ir dėl algebrinių veiksmų su išretintomis matricomis efektyvumo naudotoje programavimo aplinkoje.

3.1.2. Mašinų mokymo metodų taikymas tekstų klasifikavimui

Mašininis tekstų klasifikavimas gali būti naudojamas atskirti dvi klases (pvz., teisingą nuo klaidingos) (angl. *binary classification*) arba daugiau nei dvi klases (angl. *multiclass classification*). Jeigu reikia klasifikuoti daugiau nei į dvi klases, galima rinktis vienos klasės atskyrimo OAA (angl. *one against all*) arba visų klasių porų atskyrimo OAO (angl. *one against one*) procedūrą. Pirmoji reikalauja eksponentinio rezultatų palyginimų skaičiaus, kol nustatomos jų klasės, o antroji — $M(M-1)/2,\ M = |\mathbf{C}|$ nepriklausomų mokymo iteracijų ir rezultatų palyginimų ($\mathbf{C} = \{c_1, \ldots, c_M\}$ yra klasių aibė) [ZWB+11]. Daugiaklasio klasifikavimo procedūra gali reikalauti tolesnio rezultatų apdorojimo: 1) pakartotinio klasifikavimo esant lygiems maksimaliems įvertinimams — REV (angl. *Revoting of Equal Votes*); 2) nepakankamai tikslius klasifikavimo rezultatus gavusių klasių pakartotinis klasifikavimas — RCC (angl. *Revoting of Confusing Classes*). Yra sukurtų ELM schemos modifikacijų, kurias galima taikyti daugelio klasių atskyrimui (pvz. V-ELM (angl. *Voting ELM*)) [CLH+12] ir kt., bet šiame darbe testuotas tik dviejų klasių atskyrimo efektyvumas.

Šiame darbe pasirinkta naudoti ELM ir R-ELM modifikaciją, o jų rezultatus palyginti su SVM. Taip pat sudarytos OP-ELM ir DP (2 sluoksnių) su BP mokymo algoritmu realizacijos, bet jų efektyvumo rodiklių palyginti su minėtomis schemomis nepavyko dėl mažos greitaveikos apdorojant pasirinktus duomenis.

ELM. ELM pristatyta literatūros apžvalgoje. Čia galima parašyti apie paslėptojo sluoksnio dydžio parinkimą, duomenų dimensiškumo reikšmę, aktyvavimo funkcijas ir svorių skirstinį.

R-ELM. R-ELM schema klasikinės ELM išvesties svorių radimo lygčių sistemą papildo išvesties svorius minimizuojančiu apribojimu. Tokiu būdu šiek tiek padidinama mokymo klaida, bet

testavimo metu neuroninis tinklas veikia stabiliau. R-ELM mokymo ir testavimo algoritmai yra tokie pat kaip ir klasikinio ELM varianto. Kitokia yra tik klaidos funkcija:

$$\|\mathbf{H}\hat{\mathbf{V}} - \mathbf{T}\| = \min_{\mathbf{V}}(\|\mathbf{H}\mathbf{V} - \mathbf{T}\| + \lambda\|\mathbf{V}\|^2),$$

čia $\lambda \in \mathbb{R}^+$ — reguliarizavimo konstanta. Tuomet mokymo metu R-ELM išvesties sluoksnio svoriai randami pagal tokią formulę:

$$\hat{\mathbf{V}} = (\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T}.$$

OP-ELM. Rengiant šį darbą buvo sudaryta OP-ELM schemos realizacija, bet atlikti jos efektyvumo su pasirinktais duomenimis tyrimo nepavyko. OP-ELM [ROT+08] (angl. *Optimally-pruned ELM*) naudojama paslėptojo sluoksnio sudarymo procedūra, lyginant su ELM ir R-ELM, reikalauja daug didesnių skaičiavimų. OP-ELM ranguoja paslėptojo sluoksnio neuronus pagal jų išvesčių koreliacijos su mokymo tikslinėmis reikšmėmis dydį. Šiame žingsnyje naudojamas MRSR algoritmas išrinkdamas iš paslėptojo sluoksnio neuronų aibės geriausią neuroną atlieka atvirkštinės matricos skaičiavimą, tuomet kiekvienam kitam atrenkamam neuronui toks skaičiavimas turi būti pakartotas su viena eilute ir stulpeliu mažesne matrica. Dėl tiriamų duomenų rinkinių apimties tiesioginis atvirkštinės matricos skaičiavimas kiekvienoje iteracijoje turi būti keičiamas našesniais metodais, kaip aprašyta [Dic98]. Kitame OP-ELM žingsnyje pasirinkto reikšmingumo slenksčio neperžengiantys neuronai yra šalinami iš tinklo. Tokiu būdu pasiekiama kompaktiškesnė neuroninio tinklo architektūra ir padidinamas generalizavimo stabilumas

OP-ELM mokymo algoritmas.

1) Paslėptojo sluoksnio svorių $\mathbf{W}$ ir postūmių $\mathbf{B}$ inicializavimas.

2) Paslėptojo sluoksnio išėjimo matricos $\mathbf{H}(\mathbf{X}, \mathbf{W}, \mathbf{B})$ apskaičiavimas, naudojant mokymo imtį $\mathbf{X}$.

3) Paslėptojo sluoksnio neuronų rangavimas naudojant MRSR.

4) Reikšmingumo slenksčio neperžengiančių neuronų pašalinimas naudojant LOO.

5) Išėjimo sluoksnio svorių matricos $\mathbf{V} = \mathbf{H}^{+}\mathbf{T}$ apskaičiavimas.

BP ir SVM. ELM ir jos modifikacijų rezultatus šiame darbe planuota palyginti su klasikinėmis DP(BP) ir SVM mašinų mokymo schemomis. BP pasirinkimas motyvuotas tuo, kad šis mokymo algoritmas, kaip ir ELM, gali būti naudojamas vieno paslėptojo sluoksnio DNT, be to, BP savybės yra gerai ištirtos, todėl naudinga įvertinti ELM rezultatus jo atžvilgiu.

ELM ir SVM yra skirtingų klasių mašinų mokymo schemos, SVM nėra DNT, bet ji yra viena efektyviausių klasifikuojant tekstus, todėl naudinga palyginti ELM ir SVM efektyvumą. Šiame tyrime panaudota libSVM realizacija [CL11]. Bandymui parinkti parametrai nurodyti tyrimo aprašyme (3.2. poskyris).

DP(BP) bandymai su vienu paslėptuoju sluoksniu su pasirinktais duomenų rinkiniais buvo nesėkmingi. Per ~ 100 val. trukusį bandymą įvykdžius 40 mokymo epochų, nebuvo pastebėta reikšmingo mokymo klaidos mažėjimo.

3.1.3. Rezultatų įvertinimo ir palyginimo metodai

Be įprastų mokymo ir testavimo klaidos didumo ir trukmės rodiklių, klasifikavimo efektyvumui įvertinti šiame darbe dar naudojami šie įverčiai [ZQL13]:

1) atpažinimo pilnumas (angl. *Recall*). Komplementarus pirmojo tipo statistinei klaidai rodiklis. Parodo, kuri dalis klasės įrašų klasifikavimo metu buvo teisingai atpažinta

$$\mathrm{re}_i = \frac{|TP_i|}{|TP_i| + |FN_i|}.$$

Čia ir toliau TP_i žymi klasei i teisingai priskirtų jos įrašų poaibį (angl. *True positive*); FP_i žymi klasei i neteisingai priskirtų kitos klasės įrašų poaibį (angl. *False positive*); TN_i žymi klasei i teisingai nepriskirtų kitos klasės įrašų poaibį (angl. *True negative*); FN_i žymi klasei i neteisingai nepriskirtų jos įrašų poaibį (angl. *False negative*);

2) atpažinimo tikslumas (angl. *Precision*). Komplementarus antrojo tipo statistinei klaidai rodiklis. Parodo, kuri dalis visų klasifikavimo metu klasei priskirtų įrašų jai iš tiesų priklauso

$$\mathrm{pr}_i = \frac{|TP_i|}{|TP_i| + |FP_i|};$$

3) F_1 harmoninis vidurkis apibendrina atpažinimo pilnumo ir tikslumo rodiklius. Šis rodiklis dažnai naudojamas tekstų klasifikavimo efektyvumui vertinti

$$\mathrm{F}_{1i} = \frac{2\,\mathrm{re}_i\,\mathrm{pr}_i}{\mathrm{re}_i + \mathrm{pr}_i};$$

4) mF_1 harmoninis mikro vidurkis įvertina klasifikavimo efektyvumą viso duomenų rinkinio atžvilgiu

$$\mathrm{mF}_1 = \frac{2\,\widehat{\mathrm{re}}^U\,\widehat{\mathrm{pr}}^U}{\widehat{\mathrm{re}}^U + \widehat{\mathrm{pr}}^U},$$

kur

$$\widehat{\mathrm{re}}^U = \frac{\sum_{i\in\mathbf{C}} |TP_i|}{\sum_{i\in\mathbf{C}}(|TP_i| + |FN_i|)},$$

ir

$$\widehat{\mathrm{pr}}^U = \frac{\sum_{i\in\mathbf{C}} |TP_i|}{\sum_{i\in\mathbf{C}}(|TP_i| + |FP_i|)};$$

5) MF_1 harmoninis makro vidurkis parodo vidutinę klasės c_i F_1 rodiklio reikšmę duomenų rinkinio kontekste

$$\mathrm{MF}_1 = \frac{\sum_{i\in\mathbf{C}} \mathrm{F}_{1i}}{|\mathbf{C}|}.$$

3.2. Tyrimo aprašymas

Šiame darbe tiriamas ELM ir jos modifikacijos R-ELM tekstų klasifikavimo efektyvumas ir lyginamas su atitinkamais SVM rezultatais. Tam sudaryti tekstų rinkinių žodžių sąrašai ir suskaičiuoti jų dažniai rinkinių tekstuose. Po to šie duomenys konvertuoti į tekstų VSM su tfidf, kurie tuomet panaudoti šių schemų mokymui ir generalizavimo efektyvumo vertinimui bei palyginimui.

3.2.1. Panaudoti įrankiai

Pradiniam duomenų apdorojimui ir mašinų mokymo schemų realizacijų sudarymui buvo naudojama programavimo aplinka Octave [EBH+17], kuri, savo ruožtu, naudoja OpenBLAS [QXY+] projekto tiesinės algebros bibliotekų BLAS ir LAPACK realizacijas. Octave paketas yra sukonfigūruotas kaip vienos gijos programa, bet OpenBLAS biblioteka be papildomo konfigūravimo skaičiavimams gali naudoti daugiau negu vieną giją. Šiame tyrime taip pat panaudota libSVM projekto SVM realizacija [CL11].

Bandymams parengtas kompiuteris su Intel i5 6500 CPU ir 32 GiB RAM. Naudota openSUSE Leap 42.3 OS su 64b Linux branduoliu ir bibliotekomis.

3.2.2. Duomenų rinkiniai

Tyrimui pasirinkti dažnai klasifikatorių našumui vertinti naudojami duomenų rinkiniai — Reuters-21578 (toliau šiame darbe vadinamas r8) ir 20 Newsgroups (20ng) [Car07].

Rinkinys Reuters-21578 sudarytas iš 1987 m. naujienų agentūros Reuters skelbtų straipsnių. 1990 m. rinkinys tapo prieinamas viešai. Dokumentai į klases suskirstyti pagal temas, kurioms jie priklauso. Originaliame rinkinyje yra dokumentų, kurie priklauso kelioms klasėms, arba nepriklauso nei vienai. Šiame tyrime naudojama apdorota rinkinio versija, kurioje palikti dokumentai, priklausantys tik vienai kategorijai. Rinkinyje pateikti du dokumentų suskirstymai — į 8 ir į 52 klases. Čia naudotas pirmasis skirstymas. Dokumentų suskirstymą į klases žr. 1 lentelėje. r8 rinkinio mokymo ir testavimo aibėse skirtingų klasių dydžiai skiriasi. Tai reikšmingas parametras renkantis duomenų parengimo atitinkamai mašinų mokymo schemai būdus ir vertinant mokymo ir testavimo rezultatus. Pvz., reikia atkreipti dėmesį, kad mokymo ir testavimo imtyse duomenų pasiskirstymai pagal klases būtų proporcingi. r8 rinkinio skirtingų klasių duomenys yra tarpusavyje sumaišyti. Duomenų pateikimo tvarka svarbi kai kurioms mašinų mokymo schemoms, pvz. stochastinio mokymo BP. ELM ir R-ELM nėra jautrios duomenų pateikimo tvarkai, bet joms reikšmingi klasių dažniai mokymo imtyje.

Klasė	Mokymo imties dydis	Testavimo imties dydis	Viso
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Viso	5485	2189	7674

1 lentelė. Reuters-21578 duomenų rinkinys

20 Newsgroups duomenų rinkinyje surinkta apie 20000 straipsnių iš 20 panašaus dydžio klasių. Duomenų šaltinis — teminėse interneto diskusijų grupėse skelbtos žinutės. Rinkinys suskirstytas į klases pagal grupes, kuriose jo dokumentai buvo skelbti. Kai kurios grupės yra tarpusavyje labiau susiję negu kitos:

1) dalyje grupių skelbiami straipsniai apie kompiuterių aparatinę ir programinę įrangą (comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x);

2) dalis temų — apie sporto šakas (rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey);

3) apie mokslą (sci.crypt, sci.electronics, sci.med, sci.space);

4) kita dalis skirta diskusijoms apie religinius dalykus (alt.atheism, soc.religion.christian, talk.religion.misc);

5) apie politiką (talk.politics.guns, talk.politics.mideast, talk.politics.misc);

6) dalis grupių su kitomis nesusijusios (misc.forsale).

Iš dokumentų pašalinta juos su diskusijų grupe susiejanti informacija. Taip pat pašalintos daugiau negu vienoje iš šių grupių pakartotinai skelbtos žinutės. Dokumentų suskirstymą į klases žr. 2 lentelėje.

Klasė	Mokymo imties dydis	Testavimo imties dydis	Viso
alt.atheism	480	319	799
comp.graphics	584	389	973
comp.os.ms-windows.misc	572	394	966
comp.sys.ibm.pc.hardware	590	392	982
comp.sys.mac.hardware	578	385	963
comp.windows.x	593	392	985
misc.forsale	585	390	975
rec.autos	594	395	989
rec.motorcycles	598	398	996
rec.sport.baseball	597	397	994
rec.sport.hockey	600	399	999
sci.crypt	595	396	991
sci.electronics	591	393	984
sci.med	594	396	990
sci.space	593	394	987
soc.religion.christian	598	398	996
talk.politics.guns	545	364	909
talk.politics.mideast	564	376	940
talk.politics.misc	465	310	775
talk.religion.misc	377	251	628
Viso	11293	7528	18821

2 lentelė. 20 Newsgroups duomenų rinkinys

Abiejuose rinkiniuose pateikti keturiais būdais apdoroti dokumentai:

1) su **all-terms** tipo dokumentais atliktos tokios transformacijos:
    1) originalaus teksto tabuliacijos (`TAB`), naujos eilutės (`NL`) ir grįžimo į eilutės pradžią (`CR`) simboliai pakeisti tarpo simboliais;
    2) ne abėcėlės simboliai (pvz., skaičiai, skyrybos ženklai) pakeisti tarpo simboliais;
    3) didžiosios raidės pakeistos į mažąsias;
    4) sekos iš daugiau nei vieno tarpo simbolio pakeistos vienu tarpo simboliu;
    5) kiekvieno dokumento pavadinimas įterptas į jo teksto pradžią.
2) **no-short** iš **all-terms** rinkinio pašalinti trumpesni nei 3 simbolių žodžiai;
3) **no-stop** iš **no-short** rinkinio pašalinti 524 SMART žodžių rinkinio žodžiai [LYR+04];
4) **stemmed no-stop** rinkinio žodžiai apdoroti Porter Stemmer algoritmu [VRP80].

Iš Reuters-21578 rinkinio tyrimui naudotas **r8-all-terms** variantas. Mokymo imties dydis — 5485 įrašai, testavimo imties dydis — 2189 įrašai. Įrašai sudaryti iš 19984 atributų. Duomenys — 8 klasių. Iš 20 Newsgroups rinkinio naudotas **20ng-all-terms** variantas. Mokymo imties dydis — 11293 įrašai, testavimo imties — 7528, įrašų dydis — 93864 atributų. Klasių skaičius — 20.

Pasirinkti minimaliai apdoroti duomenų rinkiniai. Juose pateikiami visi žodžiai, net ir dažnai besikartojantys, ir trumpesni negu trijų raidžių; žodžiai netrumpinti. Tokie tekstai praktiškai dažnai pasitaiko, juose dažnai būna trumpinių ir žargono, kuris yra svarbus teksto priklausymo kokiai nors klasei požymis.

ELM mokymo algoritmo realizacija yra jautri duomenų standartizavimui ir dekoreliavimui. Lyginant su tf, tfidf reprezentacijos naudojimas leido pasiekti mažesnę išvesties svorių normą ir geresnius mokymo ir testavimo rezultus. tfidf pavidalo duomenys yra pasiskirstę mažesniame intervale už tf.

3.2.3. Duomenų klasifikavimas

Rinkinyje duomenys pateikti suskirstytų pagal klases straipsnių pavidalu. Iš tekstų pašalinti skyrybos ženklai, žodžių eilės tvarka tekste nekeista. Kiekvienam straipsniui rinkmenoje skiriama atskira eilutė. Toliau duomenys apdoroti tokia tvarka:

1) neapdoroti duomenys įkrauti į darbinę atmintinę (žr. `loadData.m`);

2) sudarytas žodynas (žr. `compileDictionary.m`);

3) suskaičiuoti žodžių dažniai (žr. `tf.m`);

4) VSM su tfidf skaičiavimas (žr. `tfidf.m`);

5) pasirinktų mašinų mokymo schemų realizacijų (ELM, R-ELM, SVM) mokymas ir testavimas (ELM žr. `train.m` ir `test.m`, R-ELM žr. `trainRElm.m` ir `test.m`);

6) suskaičiuoti mF_1 ir MF_1 harmoniniai vidurkiai.

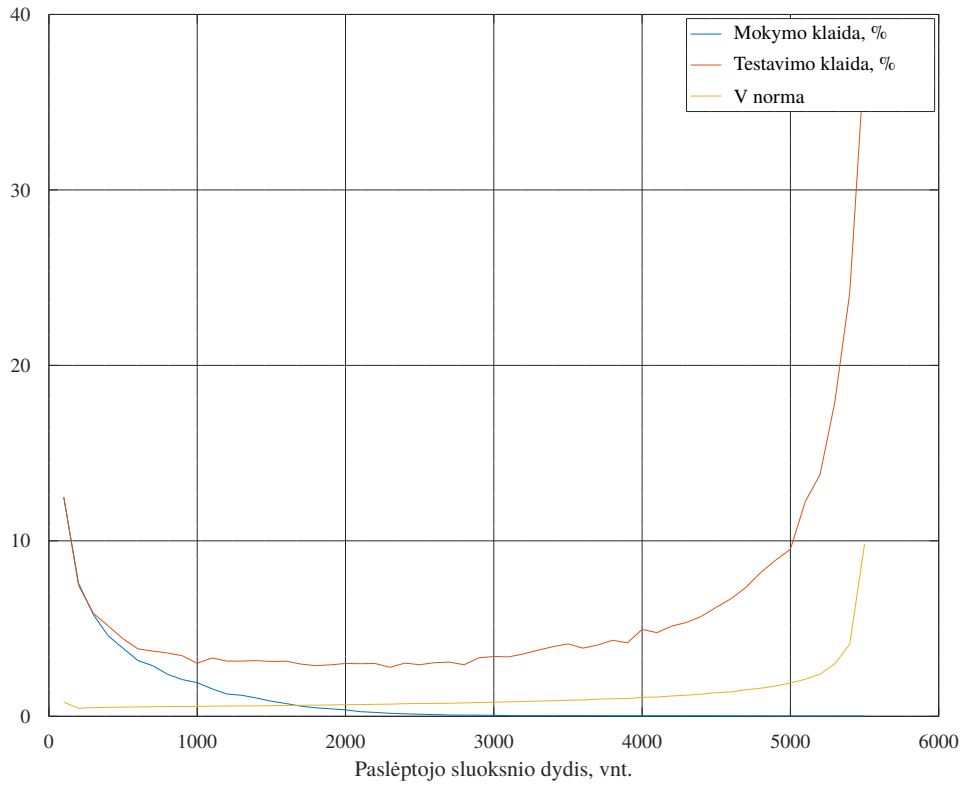
3.2.4. Rezultatų įvertinimas ir palyginimas

Teorijoje teigiama, kad ELM generalizavimo efektyvumas kitoms sąlygoms nekintant priklauso nuo dviejų veiksnių: mokymo klaidos didumo ir išvesties sluoksnio svorių normos didumo. Atlikus tyrimą su r8 ir ng20 duomenų rinkiniais (kiekviename bandyme viena iš rinkinio klasių pasirinkta kaip teigiama, visos likusios — kaip neigiama, tada apskaičiuoti visų tokių rezultatų mokymo

$$\bar{e}_{train} = \frac{1}{|C|} \sum_{\forall c \in C} e^c_{train},$$

testavimo

$$\bar{e}_{test} = \frac{1}{|C|} \sum_{\forall c \in C} e^c_{test},$$



2 pav. ELM mokymo ir testavimo klaidų ir išvesties svorių normos vidurkiai (duomenys — r8 VSM su tfidf)

klaidų ir išvesties svorių normos

$$\|\bar{V}\| = \frac{1}{|C|} \sum_{\forall c \in C} \|V^c\|$$

aritmetiniai vidurkiai).

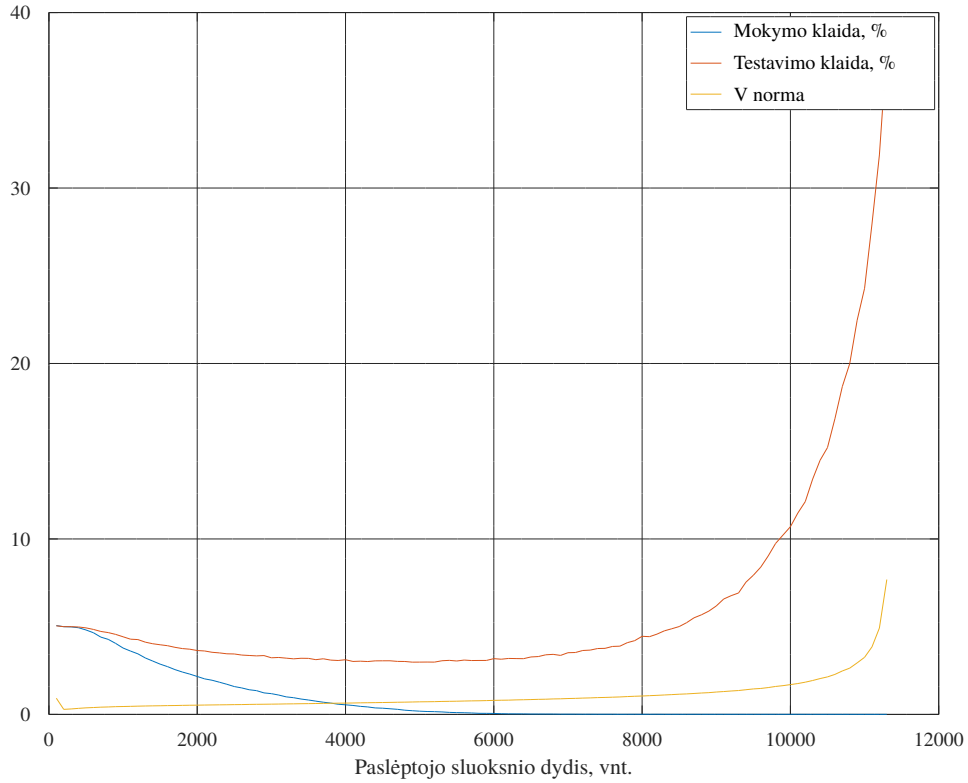
Su r8 rinkiniu atlikti matavimai 51 taške, pradedant vienu ir kas 100 didinant paslėptojo sluoksnio neuronų skaičių. Su 20ng rinkiniu atliktas 131 toks matavimas. r8 atveju kiekviename taške išmatuotos minėtos reikšmės kiekvienai iš 8 klasių, ir pavaizduotas šių klasių rezultatų aritmetinis vidurkis. 20ng atveju tokie matavimai atlikti su 20 klasių. Gauti r8 rinkinio mokymo, testavimo klaidų ir išvesties svorių normos matavimo rezultatai pavaizduoti 2 diagramoje. Tokie patys 20ng rinkinio rezultatai pavaizduoti 3 diagramoje.

Toliau pateikiamos abiejų tirtų rinkinių klasifikavimo rezultatų klasikiniu ir reguliarizuotu ELM variantais suvestinės. Pateikiami mokymo ir testavimo rezultatai, matavimai su kiekvienu paslėptojo sluoksnio dydžiu atlikti 10 kartų, kiekvienam matavimui skaičiuojant visų klasių rezultatų aritmetinį vidurkį, tuomet apskaičiuotas šių 10 matavimų rezultatų aritmetinis vidurkis. Bandymai atlikti su pilnomis mokymo ir testavimo imtimis.

Harmoniniai vidurkiai mF_1 ir MF_1 skaičiuojami tik iš testavimo imties rezultatų.

Paslėptojo sluoksnio svoriai $\mathbf{W}$ ir postūmiai $\mathbf{b}$ generuoti tolygiai pasiskirstę intervale $[-1, 1]$. Paslėptojo sluoksnio įvesties aktyvavimo funkcija — logistinė

$$g(\mathbf{x}_i, \mathbf{w}_j, b_j) = \frac{1}{1 + e^{-(x_i \cdot \mathbf{w}_j + b_j)}},$$



3 pav. ELM mokymo ir testavimo klaidų ir išvesties svorių normos vidurkiai (duomenys — 20ng VSM su tfidf)

$\tilde{N}$	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
150	0,245	6,43	0,199	6,39	0,100	0,70543	0,21675
500	0,813	3,26	0,646	4,12	0,318	0,82097	0,48136
1000	1,679	1,58	1,292	3,22	0,629	0,86339	0,64399
2000	3,642	0,28	2,575	2,90	1,255	0,87965	0,73611
4000	8,534	0,03	5,143	4,87	2,506	0,80558	0,74217

3 lentelė. ELM r8

$\tilde{N}$	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
150	0,243	6,60	0,201	6,59	0,101	0,69534	0,21481
500	0,812	3,31	0,651	4,07	0,319	0,82323	0,48997
1000	1,683	1,57	1,294	3,20	0,629	0,86487	0,65062
2000	3,642	0,28	2,581	2,98	1,260	0,87656	0,74015
4000	8,497	0,03	5,153	4,58	2,509	0,81661	0,74445

4 lentelė. R-ELM ($\lambda = 2$) r8

$\tilde{N}$	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
150	1,048	5,00	0,882	5,00	0,635	0,00013	0,00012
500	3,514	4,65	2,919	4,85	2,109	0,06553	0,06140
1000	7,094	3,61	5,821	4,31	4,197	0,25871	0,24332
2000	14,585	2,05	11,564	3,60	8,345	0,46061	0,44174
4000	31,108	0,50	23,033	3,07	16,596	0,59318	0,57815
6000	49,991	0,04	34,517	3,15	24,941	0,61780	0,60865
8000	71,897	0,01	46,062	4,46	33,230	0,54923	0,54987
10000	95,919	0,01	57,631	11,46	41,562	0,32623	0,33230

5 lentelė. ELM 20ng

$\tilde{N}$	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
150	1,045	5,00	0,888	5,00	0,640	0,00007	0,05007
500	3,483	4,66	2,915	4,85	2,098	0,06356	0,10967
1000	7,028	3,61	5,788	4,32	4,182	0,25515	0,24120
2000	14,525	2,04	11,530	3,61	8,333	0,45905	0,44072
4000	31,158	0,49	23,045	3,07	16,633	0,59338	0,57910
6000	49,714	0,04	34,511	3,14	24,891	0,61850	0,60918
8000	71,345	0,01	46,023	4,35	33,231	0,55417	0,55351
10000	95,248	0,01	57,465	9,93	41,443	0,35911	0,36659

6 lentelė. R-ELM ($\lambda = 2$) 20ng

išvesties — tiesinė

$$f(\mathbf{h}_i, \mathbf{v}_j) = \mathbf{h}_i \cdot \mathbf{v}_j.$$

Iš ELM ir R-ELM rezultatų matyti, kad esant nedideliam paslėptajam sluoksniui, reguliarizavimo poveikis yra nežymus, nes $\|V\|$ yra maža, bet didėjant paslėptojo sluoksnio dydžiui $\tilde{N}$ didėja ir $\|V\|$, todėl labiau pastebimas tampa ir reguliarizavimo poveikis. Reguliarizavimo konstanta λ parinkta taip, kad R-ELM paslėptojo sluoksnio didumui esant arti optimalaus, pagerintų geriausią ELM rezultatą, bet keičiant λ reikšmę galima R-ELM optimizuoti įvairaus didumo paslėptiesiems sluoksniams.

Mokymo ir testavimo klaidų dydžiai gali neparodyti esminių generalizavimo trūkumų, nes neinterpretuoja klaidų pobūdžio. Tam reikia surinkti informaciją apie pirmo ir antro tipo statistines klaidas rezultatuose. Pvz., 20ng duomenų rinkinio atveju daugelis klasių yra 5% dydžio lyginant su visu duomenų rinkiniu. Trivialus klasifikatorius, visuomet spėjantis neigiamą klasę, lengvai galėtų pasiekti 95% teisingų spėjimų, tačiau susidarytų tikrojo neatitinkantį duomenų modelį. mF_1 ir MF_1 rodikliai leidžia įvertinti klasifikatoriaus efektyvumą klasifikuojant rinkinio duomenis bendrai ir atpažįstant atskiras rinkinio klases atitinkamai. Tai svarbu ir dėl to, kad DNT apmokomi įgyja inerciją klasių dydžio atžvilgiu — didesnė klasė spėjama su didesne tikimybe. Naudojant mF_1 ir MF_1 rodiklius galima įvertinti DNT daromas tokio pobūdžio generalizavimo klaidas. Bandymų su ELM rezultatus žr. 3 ir 5 lentelėse. Bandymų su R-ELM rezultatus žr. 4 ir 6 lentelėse.

Atliekant bandymą su SVM griausi rezultatai pasiekti naudojant RBF pavidalo branduolius $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$, baudos parametrą $C = 2$, branduolio parametrą $\gamma = 1/|X|$, kur X yra duomenų įrašų skaičius. SVM bandymų rezultatus žr. 7 lentelėje.

Duomenų rinkinys	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
r8	29,224	1,65	26,660	2,29	10,181	0,89998	0,56716
ng20	243,080	4,09	211,790	4,57	140,620	0,16642	0,15297

7 lentelė. SVM r8 ir 20ng

Geriausi ELM, R-ELM ir SVM rezultatai klasifikuojant r8 palyginti 8 lentelėje. Atitinkamus rezultatus su 20ng žr. 9 lentelėje.

Schema	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
ELM	3,642	0,28	2,575	2,90	1,255	0,87965	0,73611
R-ELM	3,642	0,28	2,581	2,98	1,260	0,87656	0,74015
SVM	29,224	1,65	26,660	2,29	10,181	0,89998	0,56716

8 lentelė. ELM, R-ELM ir SVM r8

Schema	$\bar{t}_{train}, s$	$\bar{e}_{train}, \%$	$\bar{t}_{test}^{train}, s$	$\bar{e}_{test}, \%$	$\bar{t}_{test}^{test}, s$	mF_1	MF_1
ELM	49,991	0,04	34,517	3,15	24,941	0,61780	0,60865
R-ELM	49,714	0,04	34,511	3,14	24,891	0,61850	0,60918
SVM	243,080	4,09	211,790	4,57	140,620	0,16642	0,15297

9 lentelė. ELM, R-ELM ir SVM 20ng

SVM paprastai užtrunka $\sim 5-7$ kartus ilgiau mokymo ir testavimo fazėse, bet pasiekia panašias testavimo klaidų reikšmes. Su didelės dimensijos duomenimis ELM ir R-ELM klasifikavimo efektyvumas lenkia SVM. ELM ir R-ELM rezultatai naudojant artimą optimalaus dydžio $\tilde{N}$ paslėptąjį sluoksnį žymiai nesiskiria.

4. Rezultatai ir išvados

Atlikta literatūros apžvalga ir ELM generalizavimo efektyvumo tyrimas leidžia teigti, kad pagrindiniai ELM schemos generalizavimo efektyvumo veiksniai yra mokymo klaidos didumas ir paslėptojo sluoksnio neuronų išvesties jungčių svorių normos didumas. Jų dydžius galima reguliuoti keičiant paslėptojo sluoksnio didumą (neuronų skaičių) ir kokybę (pvz., pagal neuronų svorių ir mokymo tikslinių reikšmių atitikimą) ir papildant ELM mokymo algoritmą reguliarizavimu.

Paslėptojo sluoksnio dydį ir jo išvesties svorių normą sieja atvirkštinis ryšys: didinant paslėptąjį sluoksnį, nemažėja jo išvesties svorių norma. Be to, didinant paslėptąjį sluoksnį, proporcingai didėja skaičiavimų sudėtingumas mokymo ir testavimo metu.

Kaip pažymi P. L. Bartlett teorija, išvesties sluoksnio neuronų jungčių svorių normai nekintant, paslėptojo sluoksnio didumas reikšmingas tik mokymo klaidos mažinimui. G.-B. Huang ir kt. taip pat pažymi, kad didinant paslėptąjį sluoksnį ELM VC dimensija atitinkamai nemažėja, todėl nukenčia generalizavimo efektyvumas ir reikia daugiau mokymo duomenų, bet kuo didesnis yra paslėptasis sluoksnis, tuo mažesnė tampa mokymo klaida.

Dėl šių priežasčių ELM generalizavimo efektyvumas maksimizuojamas ieškant pusiausvyros tarp paslėptojo sluoksnio dydžio ir paslėptojo sluoksnio neuronų išvesties jungčių svorių normos dydžio. Išvesties svorių normos reguliarizavimas padeda priartėti prie šios pusiausvyros, bet bendrų teorinių teiginių apie šios pusiausvyros radimą nėra, todėl konkretiems ELM taikymams reikia atlikti individualius tyrimus. Šio tyrimo atveju reguliarizavimas prie generalizavimo efektyvumo didinimo prisidėjo tik esant dideliam paslėptajam sluoksniui, ir atitinkamai, jo normai.

ELM dėl atsitiktinių paslėptojo sluoksnio įvesties svorių ir postūmių reikalauja santykinai didelio paslėptojo sluoksnio. Tyrimo metu šio teiginio patikrinti nepavyko, nes rezultatų palyginimui pasirinktas BP algoritmas reikalavo per didelių laiko sąnaudų mokymui. Didžiausias ELM privalumas prieš kitas mašinų mokymo schemas yra jos trumpesnis mokymo laikas, bet ELM generalizavimo efektyvumas yra panašus į konkuruojančių schemų, o testavimo laikas netgi ilgesnis. Daliai praktinių taikymų būtent generalizavimo efektyvumo ir laiko derinys yra svarbesnis už mokymo trukmę. Dėl šių aplinkybių ELM schema turėtų būti labiausiai konkurencinga tuose praktiniuose taikymuose, kuriuose svarbus trumpas mokymo laikas — įvairiose greito reagavimo sistemose ir pan., todėl turėtų būti svarbūs ELM plėtiniai stochastiniam mokymui, bei ELM kompaktiškumo didinimo tyrimai. ELM taip pat paprasčiau naudoti už kitas tirtas schemas, todėl ji tinka pradiniam duomenų tyrimui, ji kelia mažiau reikalavimų naudotojų pasirengimui, ja galima tirti didelės dimensijos ir apimties duomenis.

Šiame tyrime atliktos užduotys:

1) tyrimui surinkta informacija apie DNT (ELM ir BP) ir SVM naudojimą mašininiam tekstų klasifikavimui;

2) tyrimui surinkta informacija apie tekstų klasifikavimui parengimą ir klasifikavimo rezultatų vertinimą;

3) sudaryta ir atlikta tekstinių duomenų parengimo mokymui ir klasifikavimui procedūra;

4) sudarytos ELM ir R-ELM realizacijos, taip pat sudarytos OP-ELM ir BP realizacijos;

5) atliktas tekstinių duomenų klasifikavimas ELM, R-ELM, SVM mašinų mokymo schemomis, išmatuoti ir įvertinti jų efektyvumo rodikliai.

Su tirtais duomenų rinkiniais Reuters-21578 ir 20 Newsgroups ELM generalizavimo efektyvumą maksimizuojantis parametrų rinkinys:

1) paslėptojo sluoksnio įvesties svoriai ir postūmiai tolygiai pasiskirstę, standartizuoti, intervalas $[-k, k], k \in \mathbb{R}^+$, šiame tyrime naudota $k = 1$, kitiems panašaus pobūdžio duomenims rekomenduojama pradėti šio parametro reikšmės paiešką intervale $0 < k \leq 5$;

2) mokymo ir testavimo duomenų intervalas $[0, l], l \in \mathbb{R}^+$;

3) mokymo ir testavimo duomenų matricos išretintos;

4) artimas optimaliam paslėptojo sluoksnio dydis klasifikuojant didelės dimensijos duomenis $\sim 2/5$ mokymo imties dydžio;

5) ELM klaidos funkcijos reguliarizavimas efektyvus esant didesniam už optimalų paslėptojo sluoksnio dydžiui, nes ima ryškėti persimokymo požymiai (mokymo klaidos mažėjimą atsveria testavimo klaidos didėjimas), mažėja generalizavimo stabilumas.

Atlikto tekstinių duomenų klasifikavimo efektyvumo tyrimo metu ELM pasiekė artimus SVM testavimo klaidos rezultatus (2,90% ELM ir 2,29% SVM), bet jos mokymo ir testavimo trukmė buvo $\sim 5-7$ kartus trumpesnė, kai testuota su Reuters-21578 duomenų rinkiniu. Su didesnės dimensijos ir apimties 20 Newsgroups rinkiniu ELM pasiekė geresnius testavimo klaidos rezultatus (3,15% ELM ir 4,57% SVM) ir išsaugojo greičio pranašumą. Antruoju atveju ELM sudarė ir ženkliai tikslesnį duomenų modelį už SVM (žr. 9 lentelę).

Literatūra

[Bar97] Peter L. Bartlett. For valid generalization, the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems 1996*, 9:134–140, 1997. M. Mozer, M. Jordan, and T. Petsche, editors.

[Bar98] Peter L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

[BHW+14] Zuo Bai, Guang-Bin Huang, Danwei Wang, Han Wang, and M. Brandon Westover. Sparse extreme learning machine for classification. *IEEE Transactions on Cybernetics*, 44(10):1858–1870, 2014.

[Car07] Ana Cardoso-Cachopo. Improving methods for single-label text categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

[CL11] Chih-Chung Chang ir Chih-Jen Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 3, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CL15] Jiuwen Cao and Zhiping Lin. Extreme learning machines on high dimensional and large data applications: a survey. *Mathematical Problems in Engineering*, 2015, 2015.

[CLH+12] Jiuwen Cao, Zhiping Lin, Guang-Bin Huang, and Nan Liu. Voting based extreme learning machine. *Information Sciences*, 185(1):66–77, 2012.

[Dic98] Valdas Diciunas. Simply invertible matrices and fast prediction. *Informatica, Lith. Acad. Sci.*, 9, 1998-01.

[EBH+17] John W. Eaton, David Bateman, Søren Hauberg ir Rik Wehbring. *GNU Octave version 4.2.1 manual: a high-level interactive language for numerical computations*. 2017. URL: `https://www.gnu.org/software/octave/doc/v4.2.1/`.

[FHL+09] Guorui Feng, Guang-Bin Huang, Qingping Lin, and Robert Gay. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, 20(8):1352–1357, 2009.

[Hay09] S. S. Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2009.

[HC07] Guang-Bin Huang and Lei Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16):3056–3062, 2007.

[HC08] Guang-Bin Huang and Lei Chen. Enhanced random search based incremental extreme learning machine. *Neurocomputing*, 71(16):3460–3468, 2008.

[HCS13] Punyaphol Horata, Sirapat Chiewchanwattana, and Khamron Sunat. Robust extreme learning machine. *Neurocomputing*, 102:31–44, 2013.

[HHS+15] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: a review. *Neural Networks*, 61:32–48, 2015.

[HSG+14] Gao Huang, Shiji Song, Jatinder N. D. Gupta, and Cheng Wu. Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, 44(12):2405–2417, 2014.

[Hua15] Guang-Bin Huang. What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation*, 7(3):263–278, 2015.

[HZS04] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN2004)*, vol. 2, pp. 985–990. IEEE, 2004.

[HZS06] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[YD12] Dong Yu and Li Deng. Efficient and effective algorithms for training single-hidden-layer neural networks. *Pattern Recognition Letters*, 33(5):554–558, 2012.

[YME+13] Qi Yu, Yoan Miche, Emil Eirola, Mark Van Heeswijk, Eric SéVerin, and Amaury Lendasse. Regularized extreme learning machine for regression with missing data. *Neurocomputing*, 102:45–51, 2013.

[YWY12] Yimin Yang, Yaonan Wang, and Xiaofang Yuan. Bidirectional extreme learning machine for regression problem and its learning effectiveness. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1498–1505, 2012.

[LHS+06] Nan-Ying Liang, Guang-Bin Huang, Paramasivan Saratchandran, and Narasimhan Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural networks*, 17(6):1411–1423, 2006.

[LYR+04] David D Lewis, Yiming Yang, Tony G Rose ir Fan Li. Rcv1: a new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

[LSH09] Yuan Lan, Yeng Chai Soh, and Guang-Bin Huang. Ensemble of online sequential extreme learning machine. *Neurocomputing*, 72(13):3391–3395, 2009.

[LZX+13] Kunlun Li, Juan Zhang, Hongyu Xu, Shangzong Luo, and Hexin Li. A semi-supervised extreme learning machine method based on co-training. *Journal of Computational Information Systems*, 9(1):207–214, 2013.

[MLW+11] Zhihong Man, Kevin Lee, Dianhui Wang, Zhenwei Cao, and Chunyan Miao. A new robust training algorithm for a class of single-hidden layer feedforward neural networks. *Neurocomputing*, 74(16):2491–2501, 2011.

[MSB+10] Yoan Miche, Antti Sorjamaa, Patrick Bas, Olli Simula, Christian Jutten, and Amaury Lendasse. OP-ELM: optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, 21(1):158–162, 2010.

[PPS94] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994.

[QXY+] Wang Qian, Zhang Xianyi, Zhang Yunquan, and Qing Yi. AUGEM: automatically generate high performance dense linear algebra kernels on x86 cpus. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13), Denver CO, November 2013*.

[ROT+08] Hai-Jun Rong, Yew-Soon Ong, Ah-Hwee Tan, and Zexuan Zhu. A fast pruned-extreme learning machine for classification problem. *Neurocomputing*, 72(1):359–366, 2008.

[SKD92] Wouter F Schmidt, Martin A Kraaijveld, and Robert PW Duin. Feedforward neural networks with random weights. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pp. 1–4. IEEE, 1992.

[TT97] Shin'ichi Tamura and Masahiko Tateishi. Capabilities of a four-layered feedforward neural network: four layers versus three. *IEEE Transactions on Neural Networks*, 8(2):251–255, 1997.

[VRP80] Cornelis J. Van Rijsbergen, Stephen Edward Robertson ir Martin F Porter. *New models in probabilistic information retrieval*. British Library Research ir Development Department London, 1980.

[WCY11] Yuguang Wang, Feilong Cao, and Yubo Yuan. A study on effectiveness of extreme learning machine. *Neurocomputing*, 74(16):2483–2490, 2011.

[ZH14] Weiwei Zong and Guang-Bin Huang. Learning to rank with extreme learning machine. *Neural Processing Letters*, 39(2):155–166, 2014.

[ZHC13] Weiwei Zong, Guang-Bin Huang, and Yiqiang Chen. Weighted extreme learning machine for imbalance learning. *Neurocomputing*, 101:229–242, 2013.

[ZLH+12] Rui Zhang, Yuan Lan, Guang-bin Huang, and Zong-Ben Xu. Universal approximation of extreme learning machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):365–371, 2012.

[ZQL13] Wenbin Zheng, Yuntao Qian, and Huijuan Lu. Text categorization based on regularization extreme learning machine. *Neural Computing & Applications*, 22(3-4):447–456, 2013.

[ZWB+11] Xiang-guo Zhao, Guoren Wang, Xin Bi, Peizhen Gong, and Yuhai Zhao. Xml document classification based on elm. *Neurocomputing*, 74(16):2444–2451, 2011.