Vilniaus Universitetas

Tiesioginio sklidimo DNT naudojant sistemą WEKA

Dirbtinio intelekto pagrindai

3 užduotis

Darbą atliko:

Dovydas Martinkus

Duomenų Mokslas 4 kursas 1gr.

Vilnius, 2022

Turinys

1 Tikslas ir uždaviniai

Tikslas: Išmokyti neuroninį tinklą teisingai klasifikuoti duomenis naudojant sistemą WEKA.

Uždaviniai:

Analizuojamų duomenų paruošimas ir aprašymas.

Užduočių sekų sudarymas ir jų vaizdinis pateikimas su komentarais kam skirta kiekviena naudojama komponentė.

Neuroninio tinklo parametrų parinkimas: optimalių paslėptų neuronų skaičiaus, mokymo greičio parametro bei momentum reikšmių radimas pateikiant klasifikavimo tikslumo įverčius visiems tirtiems atvejams.

Naujų duomenų klasifikavimas naudojant išsaugotą tinklo modelį.

Neuronų išėjimo reikšmių perskaičiavimas „rankiniu“ būdu pasirinktoje programoje. Išvadų apie rezultatų sutapimą padarymas.

2 Duomenys

Užduotyje naudojamą irisų duomenų aibę sudaro 150 stebėjimų, iš kurių kiekvienas turi po 4 skaitinius požymius ir klasę, kuriai priklauso. Šiuose duomenyse yra trys klasės: Setosa, Versicolor ir Virginica. Duomenų aibėje yra po 50 kiekvienos klasės stebėjimų.

Analizei naudotas *.arff* failas su šia duomenų aibe, kuris yra įrašomas į kompiuterį įdiegus sistemą WEKA.

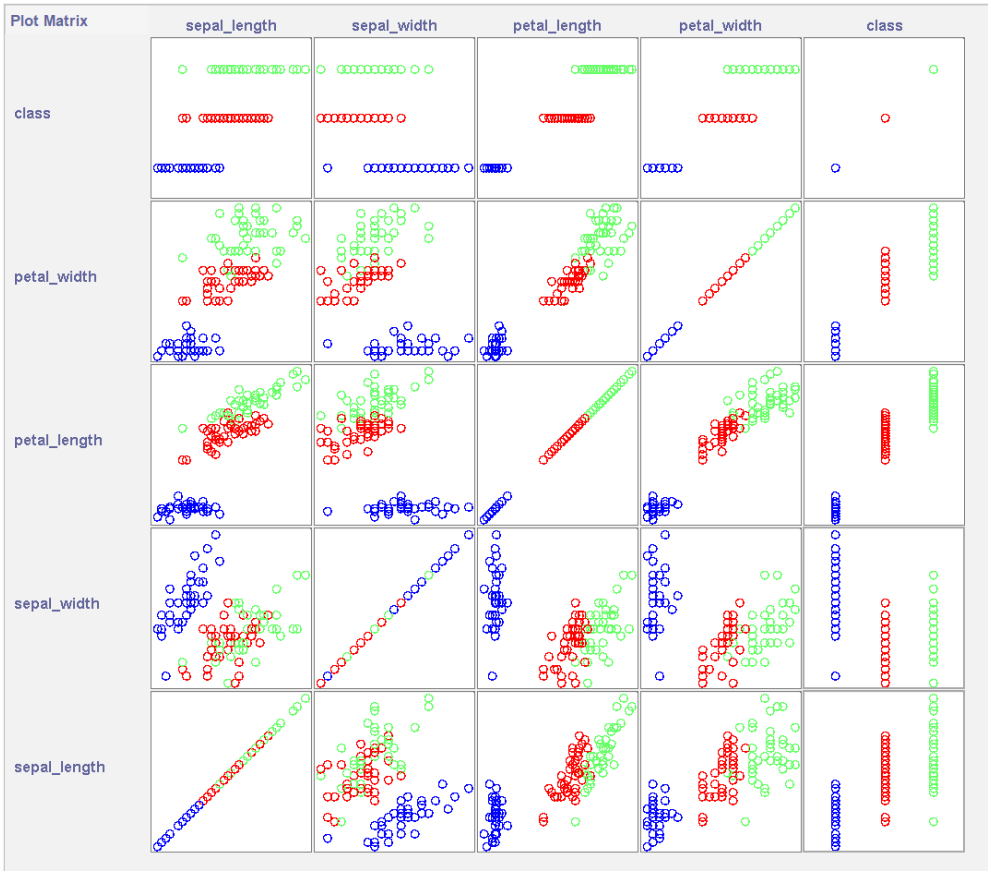
Papildomai prieiga prie duomenų rinkinio per internetą: https://archive.ics.uci.edu/ml/datasets/iris.

3 Užduoties ataskaita

Duomenų pertvarkymas atliktas naudojant programavimo kaklbą „Python“. Duomenys padalinti į mokymo ir testavimo aibes santykiu 80-20 abiejose aibėse išlaikant tokį patį skirtingų klasių stebėjimų skaičiaus santykį (mokymo aibėje tokiu būdu paliekant po 40 kiekvienos klasės stebėjimų). Mokymo ir testavimo aibės atitinkamai išsaugotos į *iris_train_test.arff* ir *iris_new.arff* failus:

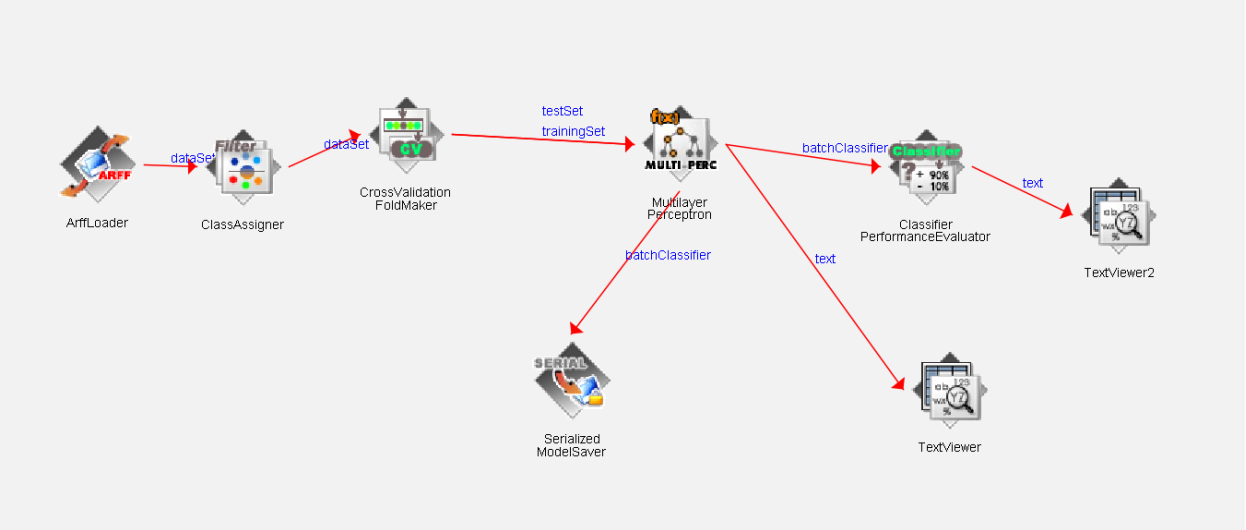
```
from scipy.io import arff as sciarff
import pandas as pd
from sklearn.model_selection import train_test_split
from collections import Counter

data = sciarff.loadarff('iris.arff')
df = pd.DataFrame(data[0])


df.columns = ["sepal_length","sepal_width","petal_length","petal_width","class"]

df = df.replace({b'Iris-setosa': 'setosa', b'Iris-virginica': 'virginica', b'Iris-
versicolor' : 'versicolor'})
train, test = train_test_split(df,train_size=0.8,test_size=0.2,stratify=df["class"])


from pandas2arff import pandas2arff
pandas2arff(train, "iris_train_test.arff", wekaname="iris_train_test")
pandas2arff(test, "iris_new.arff", wekaname="iris_new")
```

Žemiau pateikta skaitinių kintamųjų sklaidos diagramų matrica testavimo aibėje esantiems stebėjimams:



1 pav. Duomenų aibėje esančių stebėjimų skaitinių požymių sklaidos diagramų matrica

Žemiau pavaizduota WEKA sistemoje sudaryta užduočių seka, skirta surasti geriausius neuroninio tinklo parametrų rinkinius, klasifikavimo vertinimui vietoje testavimo aibės naudojant kryžminę validaciją:



2 pav. Pirmos užduočių sekų vaizdas

Užduočių sekoje naudojami šie komponentai:

„ClassAssigner“ - skirtas sekančioms komponentėms pažymėti, kuris duomenų stulpelis skirtas klasei.

„CrossValidationFoldMaker“ – atlieka kryžminę validaciją.

„TextViewer“ ir „TextViewer2“ – atitinkamai skirti peržiūrėti gautus neuroninio tinklo svorius ir klasifikavimo rezultatų įvertinimus.

„SerializedModelSaver“ – išsaugo gautą modelį.

„ClassifierPerfomanceEvaluator“ – įvertina klasifikavimo rezultatus.

„MultilayerPerceptron“ – sudaro ir apmoko daugiasluoksnį perceptroną su pasirinktais parametrais.

Naudojant šią seką keisti mokymosi greičio, momentum ir paslėptų neuronų skaičiaus parametrai, siekiant surasti parametrų reikšmių rinkinį, su kuriais gaunami geriausi klasifikavimo rezultatai. Kaip matome iš žemiau pateiktų paveikslėlių, geriausi klasifikavimo rezultatai gaunami naudojant 2 paslėptų neuronų sluoksnius, 0.1 mokymo greitį ir 0.1 momentum reikšmę (taip pat ir naudojant 0.3 momentum reikšmę):

```
Scheme: MultilayerPerceptron
Options: -L 0.1 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H "3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances         116               96.6667 %
Incorrectly Classified Instances         4                3.3333 %
Kappa statistic                          0.95
Mean absolute error                      0.0574
Root mean squared error                  0.136
Relative absolute error                 12.9232 %
Root relative squared error             28.8557 %
Total Number of Instances              120
```

3 pav. Klasifikavimo rezultatai (2 paslėpti sluoksniai, 0.1 mokymo greitis, 0.1 momentum)

```
Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.3 -N 500 -V 0 -S 0 -E 20 -H "3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances         115               95.8333 %
Incorrectly Classified Instances         5                4.1667 %
Kappa statistic                          0.9375
Mean absolute error                      0.0439
Root mean squared error                  0.1653
Relative absolute error                  9.8872 %
Root relative squared error             35.0623 %
Total Number of Instances              120
```

4 pav. Klasifikavimo rezultatai (2 paslėpti sluoksniai, 0.3 mokymo greitis, 0.3 momentum)

```
Scheme: MultilayerPerceptron
Options: -L 0.1 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H "3, 3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances          40               33.3333 %
Incorrectly Classified Instances        80               66.6667 %
Kappa statistic                          0
Mean absolute error                      0.4444
Root mean squared error                  0.4715
Relative absolute error                100      %
Root relative squared error            100.0146 %
Total Number of Instances              120
```

5 pav. Klasifikavimo rezultatai (3 paslėpti sluoksniai, 0.1 mokymo greitis, 0.1 momentum)

```
Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.3 -N 500 -V 0 -S 0 -E 20 -H "3, 3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances         114               95      %
Incorrectly Classified Instances         6                5      %
Kappa statistic                          0.925
Mean absolute error                      0.0886
Root mean squared error                  0.1906
Relative absolute error                 19.9322 %
Root relative squared error             40.441  %
Total Number of Instances              120
```

6 pav. Klasifikavimo rezultatai (3 paslėpti sluoksniai, 0.3 mokymo greitis, 0.3 momentum)

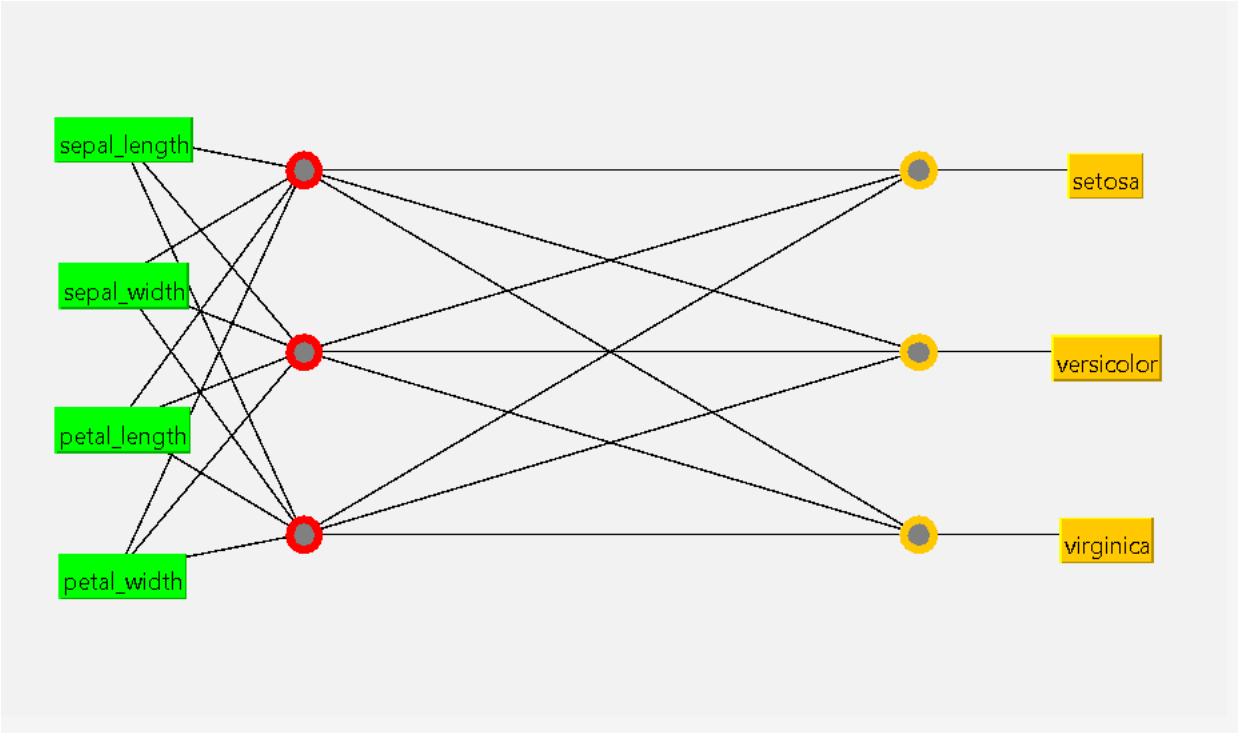
```
Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H "3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances         115               95.8333 %
Incorrectly Classified Instances         5                4.1667 %
Kappa statistic                          0.9375
Mean absolute error                      0.0454
Root mean squared error                  0.1606
Relative absolute error                 10.2051 %
Root relative squared error             34.0753 %
Total Number of Instances              120
```

7 pav. Klasifikavimo rezultatai (2 paslėpti sluoksniai, 0.3 mokymo greitis, 0.1 momentum)

```
Scheme: MultilayerPerceptron
Options: -L 0.1 -M 0.3 -N 500 -V 0 -S 0 -E 20 -H "3, 3" -R
Relation: iris_train_test-weka.filters.unsupervised.attribute.ClassAssigner-Clast

=== Summary ===

Correctly Classified Instances         116               96.6667 %
Incorrectly Classified Instances         4                3.3333 %
Kappa statistic                          0.95
Mean absolute error                      0.0517
Root mean squared error                  0.1435
Relative absolute error                 11.6287 %
Root relative squared error             30.4427 %
Total Number of Instances              120
```

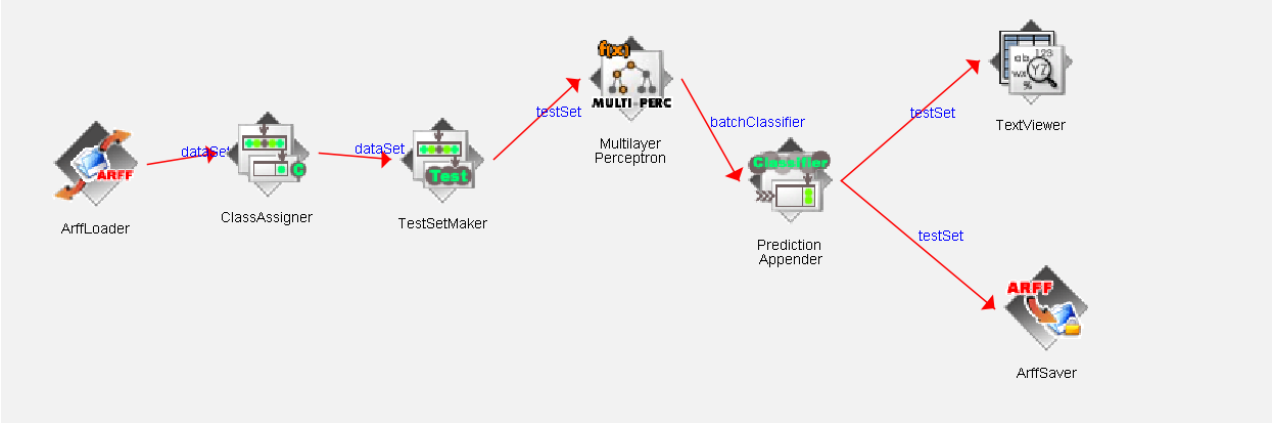
8 pav. Klasifikavimo rezultatai (2 paslėpti sluoksniai, 0.1 mokymo greitis, 0.3 momentum)

Žemiau pateiktas geriausius rezultatus pasiekusio neuroninio tinklo grafinis vaizdas:



9 pav. Sudaryto dirbtinio neuroninio tinklo vaizdas

Sudaryta ir žemiau pavaizduota WEKA užduočių seka, skirta klasifikuoti prieš tai nematytus duomenis naudojant praeitoje užduočių sekoje išsaugotą modelį (bet kurį iš gautų kryžminės validacijos metu):



10 pav. Antros užduočių sekų vaizdas

Prieš tai nenaudoti užduočių sekų komponentai:

„TestSetMaker“ – paverčia duomenų aibę į testavimo (sekančioms komponentėms).

„PredictionAppender“ – skirtas papildyti duomenis prognozuotos klasės stulpeliu.

„ArffSaver“ – išsaugo duomenis, papildytus prognozuotos klasės stulpeliu.

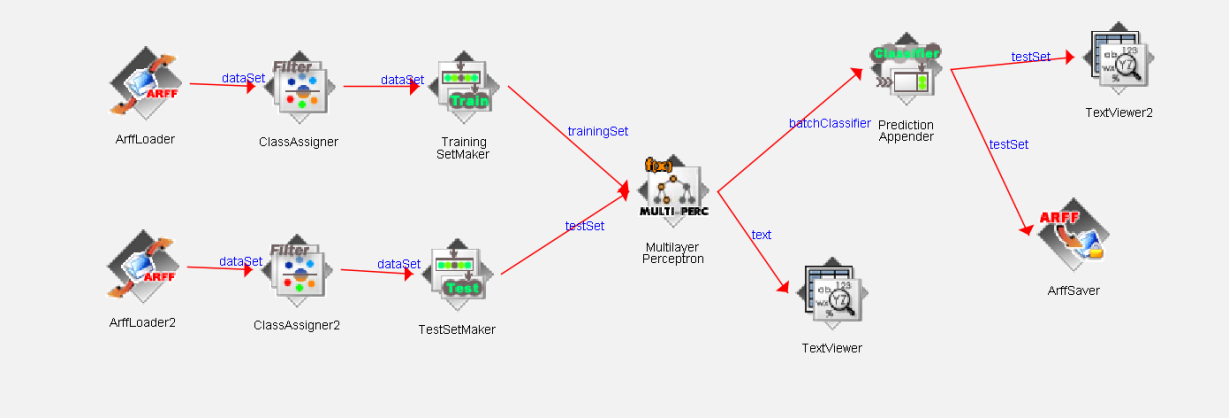
Lentelėje palyginimui pateiktos tikros klasės ir prognozuotos klasės, gautos naudojant išsaugotą modelį:

1 lentelė Tikros ir prognozuotos klasės testavimo aibei naudojant išsaugotą modelį

Tikra klasė	Prognozuota klasė
setosa	setosa
setosa	setosa
versicolor	versicolor
versicolor	versicolor
versicolor	versicolor
versicolor	versicolor
setosa	setosa
virginica	virginica
setosa	setosa
versicolor	versicolor
virginica	virginica
versicolor	versicolor
versicolor	versicolor
virginica	virginica
setosa	setosa
versicolor	versicolor
setosa	setosa
virginica	virginica

virginica	virginica
virginica	virginica
setosa	setosa
virginica	virginica
setosa	setosa
setosa	setosa
virginica	virginica
virginica	virginica
virginica	virginica
versicolor	versicolor
setosa	setosa
versicolor	versicolor

Žemiau pateiktas užduočių sekos, sudarytos siekiant apmokyti modelį naudojant visą testavimo aibę ir klasifikuoti stebėjimus testavimo aibėje, vaizdas:



11 pav. Trečios užduočių sekų vaizdas

Šioje sekoje yra tik 1 prieš tai nenaudota komponentė:

„TrainingSetMaker“ – iš duomenų sudaro mokymo aibę (sekančioms komponentėms).

Pasirinkta sudaryti neuroninį tinklą su 1 paslėptų sluoksniu, kurį sudaro 5 neuronai. Į išvesties failą išsaugotos modeliu gautos spėjamos klasės testavimo duomenims kartu su tikimybėmis priklausyti kiekvienai klasei.

Lentelėse pateikti gauti neuroninio tinklo neuronų svoriai (atskirai paslėptam ir išvesties sluoksniams):

2 lentelė Išvesties sluoksnio svoriai skirtingiems neuronams

	0	1	2
Poslinkis	1.032	-3.560	-3.690
3	3.657	-0.823	-7.646
4	-3.694	4.242	0.244
5	-1.770	-6.863	5.331
6	-1.841	-7.726	5.833
7	-5.342	7.392	0.609

3 lentelė Paslėpto sluoksnio svoriai skirtingiems neuronams

	3	4	5	6	7
Poslinkis	2.701	1.898	-5.584	-6.084	2.482
sepal_length	1.514	1.150	-2.324	-2.725	1.313
sepal_width	0.983	-1.943	-1.814	-2.089	-2.547
petal_length	-5.153	2.916	8.838	9.825	3.782
petal_width	-2.476	2.514	4.868	5.145	3.393

Naudojant šiuos išsaugotus svorius „Excel“ programa rankiniu būtu atkartotas WEKA atliktas klasių prognozių gavimas.

Duomenys, esantys testavimo aibėje perkelti į „Excel“, kartu su išsaugotais svoriais. Testavimo aibės duomenys normuoti intervale [-1 ; 1].

Skaičiavimai atlikti naudojant tokią tvarką:

Pirmiausia kiekvienam iš 5 paslėpto sluoksnio neuronų susumuotos duomenų įėjimo vektorių (4 įvestys) ir paslėptų neuronų svorių vektorių (4 svoriai ir poslinkis) sandaugos.

Šios reikšmės pateiktos sigmoidinei funkcijai. Taip gautos paslėpto sluoksnio išvesties reikšmės.

Šios 5 reikšmės naudotos kaip įvestys kiekvienam iš 3 neuronų, esančių paskutiniame (išvesties) sluoksnyje.

Galutinės priklausimo klasei tikimybės gautos pateikus 3 praėjusiame žingsnyje gautas reikšmes softmax funkcijai.

Klasės priskyrimas atliktas pasirenkant klasę, kuriai gauta didžiausia tikimybe.

Kaip matome iš žemiau esančios lentelės, „Excel“ atliktais skaičiavimais gauti rezultatai visiems testavimo aibės elementams sutapo su rezultatais, gautais naudojant WEKA sistemą:

4 lentelė Excel atliktais skaičiavimais ir WEKA gautų klasių palyginimas

Skaičiavimais gauta klasė	WEKA gauta klasė
setosa	setosa
setosa	setosa
versicolor	versicolor
versicolor	versicolor
versicolor	versicolor
versicolor	versicolor
setosa	setosa
virginica	virginica
setosa	setosa
versicolor	versicolor
virginica	virginica
versicolor	versicolor
versicolor	versicolor
virginica	virginica
setosa	setosa
versicolor	versicolor
setosa	setosa
virginica	virginica
virginica	virginica
virginica	virginica
setosa	setosa
versicolor	versicolor
setosa	setosa
setosa	setosa
virginica	virginica
virginica	virginica
virginica	virginica
versicolor	versicolor
setosa	setosa
versicolor	versicolor

4 Išvados

Geriausi klasifikavimo rezultatai buvo gauti naudojant 2 paslėptų neuronų sluoksnius, 0.1 mokymo greitį ir 0.1 momentum reikšmę (taip pat ir naudojant 0.3 momentum reikšmę).

Keičiant mokymosi greičio ir momentum reikšmes, pastebėta, kad daugeliu atveju naudojant mažesnės mokymosi greičio ir momentum reikšmes gauti geresni rezultatai. Taip pat pastebėtas faktas, kad naudojant didesnį neuronų sluoksnių skaičių galima gauti prastesnius rezultatus, negu naudojant paprastesnį dirbtinį neuroninį tinklą.

„Excel“ programoje atliktų skaičiavimų rezultatai sutapo su WEKA sistemos. Šis rezultatas leidžia teigti, kad teorinis dirbtinio neuroninio tinklo supratimas sutapo su modeliu, kuris buvo implementuotas naudojant WEKA sistemą.