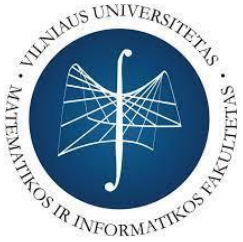


VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

DUOMENŲ MOKSLO BAKALAURO STUDIJOS

Tiesioginio sklidimo DNT naudojant sistemą WEKA

Forward Propagation NN with WEKA

tiriamasis darbas

Atliko: Vainius Gataveckas

VU el.p.: vainius.gataveckas@mif.stud.vu.lt

Vertintojas: Dr. Viktor Medvedev

Vilnius

2021

Darbo tikslas - Išmokyti neuroninį tinklą teisingai klasifikuoti duomenis naudojant sistemą WEKA.

WEKA, tai programa, kuri suteikia galimybę atlikti duomenų analizę per vizualiai patogią vartotojo sąsają. Vietoj to, kad būtų rašomas kodas, analizės etapus galima dėlioti kaip grafą ir juos sujungti pagal atitinkamas užduotis. Tokiu būdų dirbtinio intelekto modeliai patampa dar labiau „black box“, todėl yra prasminga įsitikinti skaičiavimų tikslumu panaudojus kitą programinę įrangą.

Darbo uždaviniai:

1. Paruošti duomenų rinkinius modelio taikymui
2. Naudojant WEKA programinę sistemą realizuoti modelį ir rasti geriausius hiperparametrus
3. Sukurti modelį naudojant mokymo ir testavimo duomenis.
4. Panaudojus tuos pačius svorius patikrinti modelio skaičiavimus su MS „Excel“.

1. Duomenys.

Irisų duomenų aibė suskirstoma į mokymo ir testavimo aibes. Kiekvienos, Setosa, Versicolor ir Virginica, klasės po keturiasdešimt įrašų priskiriama mokymo ir po dešimt testavimo aibei. Tai atliekama su „Python“ ir galutiniai duomenų masyvai išsaugojami kaip „.arff“ formato failai, su pavadinimais „iris_train_test.arff“ ir „iris_new.arff“.

1 kodo fragmentas. Duomenų nuskaitymas ir pertvarkymas.

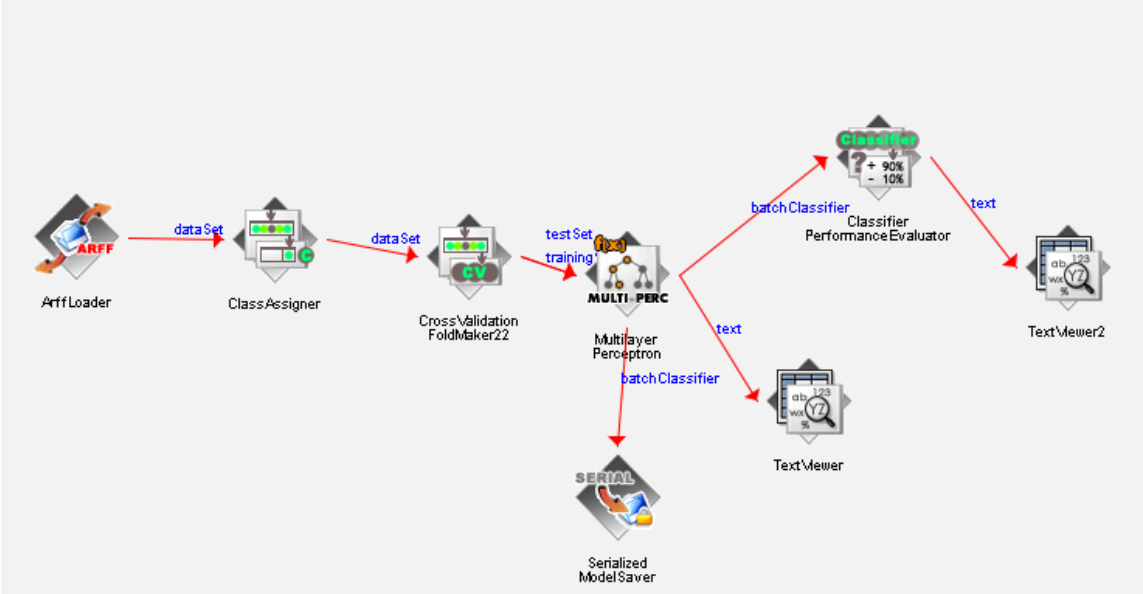
```
import pandas as pd
from sklearn.model_selection import train_test_split
import arff
data = sciarff.loadarff('iris.arff')
df = pd.DataFrame(data[0])
print(df)

df.columns =
["Taurelapio_ilgis","Taurelapio_plotis","Ziedlapio_ilgis","Ziedlapio_ploti
s","klase"]
df["klase"] = df['klase'].str.decode('utf-8')
train, test =
train_test_split(df,train_size=0.8,test_size=0.2,stratify=df["klase"])
check_train = train.klase.value_counts()
check_test = test.klase.value_counts()
print(check_train)
print(check_test)

train_arff = []
test_arff = []
for index, row in train.iterrows():
    train_arff.append([row["Taurelapio_ilgis"], row["Taurelapio_plotis"],
row["Ziedlapio_ilgis"], row["Ziedlapio_plotis"],str(row["klase"])])
for index, row in test.iterrows():
    test_arff.append([row["Taurelapio_ilgis"], row["Taurelapio_plotis"],
row["Ziedlapio_ilgis"], row["Ziedlapio_plotis"],str(row["klase"])])

print(train_arff)
arff.dump('iris_train_test.arff'
      , train_arff
      , relation='iris_train_test'
      , names=df.columns)
arff.dump('iris_new.arff'
      , test_arff
      , relation='iris_new'
      , names=df.columns)
```

2. Modelio realizacija su WEKA.

Sukuriamas modelis, kuris naudoja WEKA vartotojo sąsają. Naudota komponentų tvarka iš kairės į dešinę:

- Užkraunami duomenys
- Duomenų stulpeliai priskiri atitinkamai požymiams ir klasei
- Atliekama kryžminė validacija (5)
- Pasirinkus skirtingus hiperparametrus apmokomas modelis
- Įvertinami klasifikavimo rezultatai

Papildomai panaudota „Serialized Model Saver“- komponentas kuris išsaugo modelį atmintyje, ir „Text Viewer“ – komponentai kurie leidžia peržiūrėti kitų komponentų išvedama informacija. Šiuo atveju „Text Viewer“ pridėti prie „Multilayer Perceptron“ leidžia pamatyti svorių reikšmes kiekvienam neuronui. O likęs „Text Viewer2“ suteikia galimybę pamatyti galutinius modelio klasifikavimo tikslumo rezultatus.



1 paveikslas. Modelio hiperparametrų parinkimas naudojant WEKA

Bandant skirtingą neuroninio tinklo architektūrą, momementum ir mokymo greičio reikšmes, gauti skirtingi neuroniniai tinklai, kurie pasiekia tokį patį klasifikavimo tikslumą: teisingai klasifikuojama 96,6667 % reikšmių.

```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.5 -M 0.5 -N 500 -V 0 -S 0 -E 20 -H "4, 4, 4" -R
Relation: iris_train_test

=== Summary ===

Correctly Classified Instances         116               96.6667 %
```

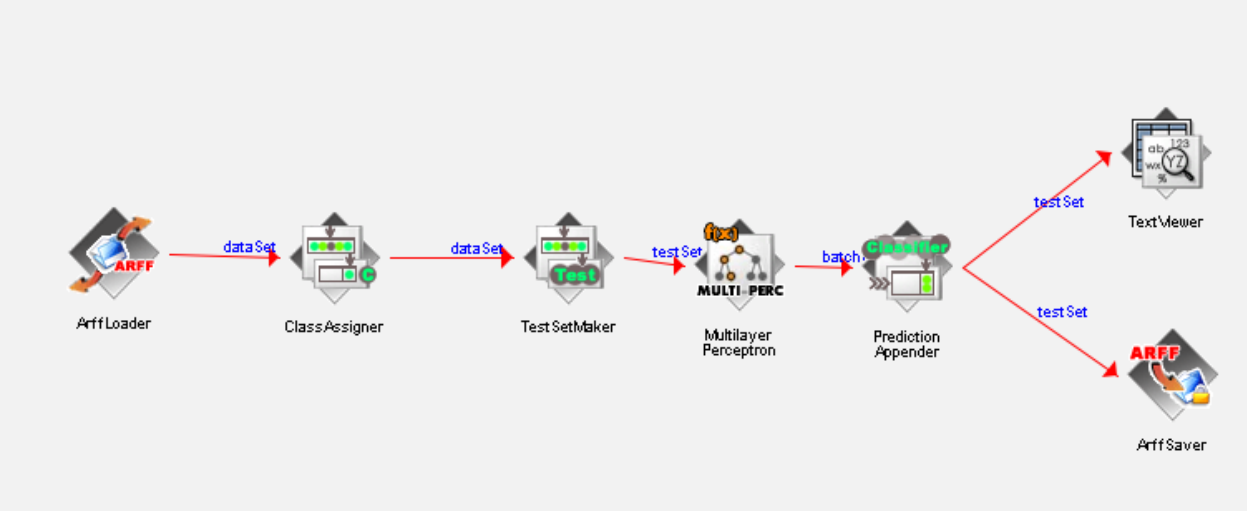
Trys paslėpti sluoksniai po 4 n.; 0,5 m. g.; 0.5 m.

```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.2 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H "4, 4" -G -R
Relation: iris_train_test

=== Summary ===

Correctly Classified Instances         116               96.6667 %
```

Du paslėpti sluoksniai po 4 n.; 0,2 m. g.; 0.1 m.

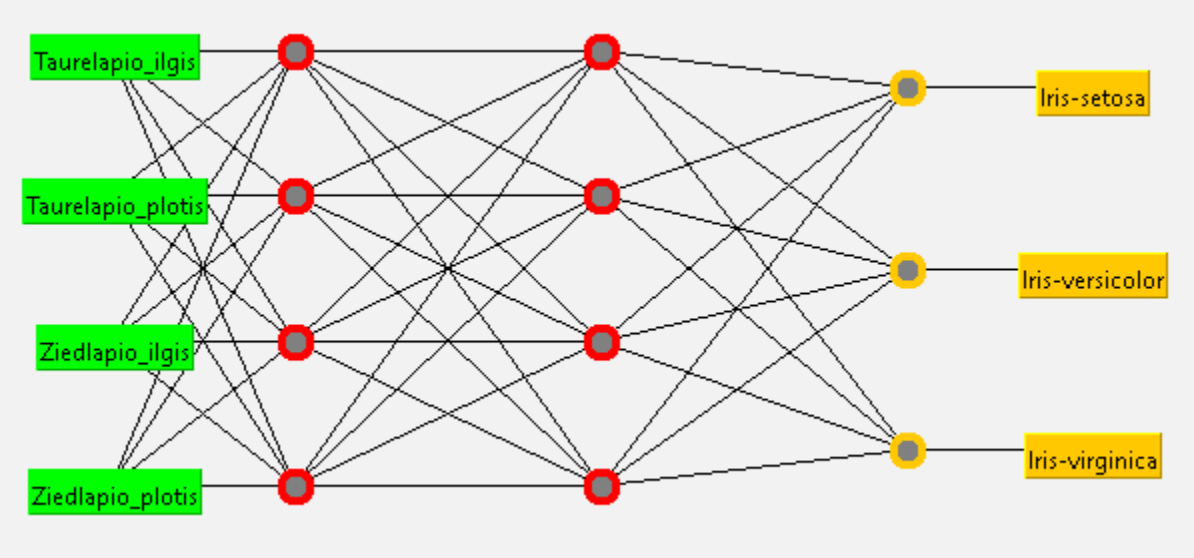
```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.1 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H 2 -G -R
Relation: iris_train_test

=== Summary ===

Correctly Classified Instances         116               96.6667 %
```

2 paslėpto sluoksnio neuronai ; 0,1 mokymo greitis; 0.1 momentum.



2 paveikslas. Modelio patikrinimo su validavimo duomenimis schema.

Modelio kokybė validavimo duomenims įvertinta su antruoju modeliu. Tai modelis su dvejais paslėptais sluoksniais po keturis neuronus.



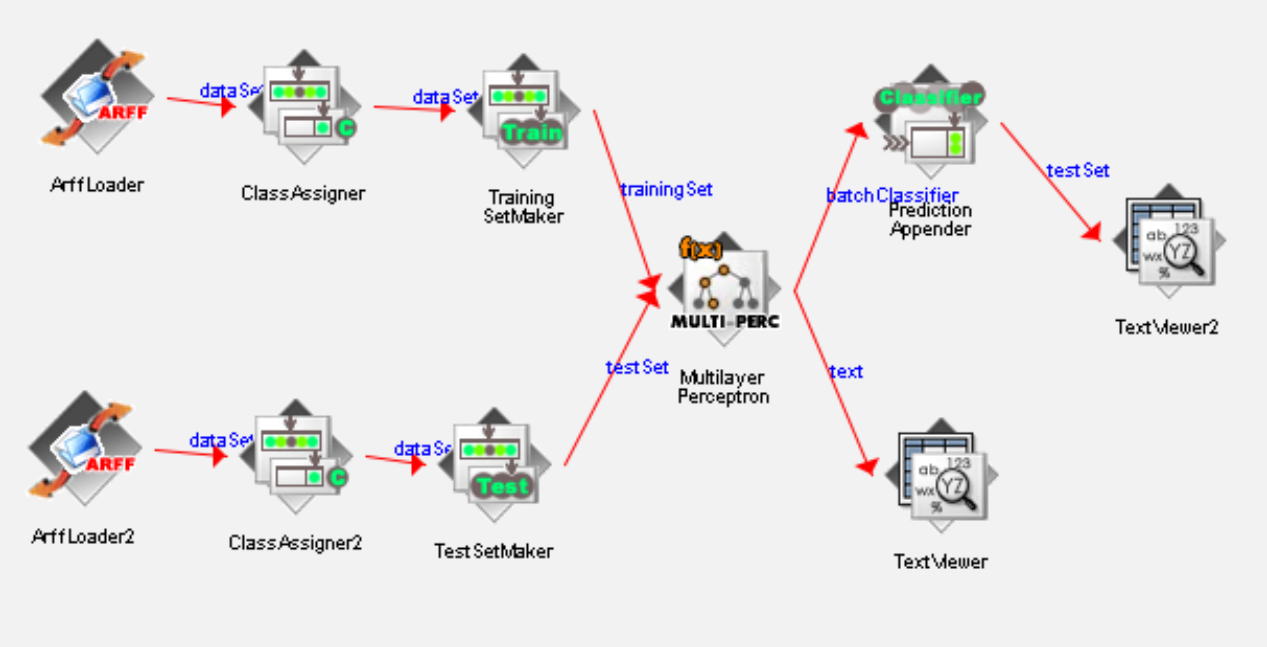
3 paveikslas. Modelio schema.

Gauti rezultatai (1 lentelė) leidžia daryti išvada, kad modelis gerai išmoko duomenis. Testavimo aibėje, rezultatai yra panašūs į rezultatus gautus apmokant modelį.

Tikroji reikšmė	Spėjama reikšmė
Iris-versicolor	Iris-versicolor
Iris-versicolor	Iris-versicolor
Iris-setosa	Iris-setosa
Iris-versicolor	Iris-versicolor
Iris-setosa	Iris-setosa
Iris-versicolor	Iris-virginica
Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa
Iris-versicolor	Iris-versicolor
Iris-virginica	Iris-virginica
Iris-versicolor	Iris-versicolor
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa
Iris-virginica	Iris-virginica
Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa
Iris-virginica	Iris-virginica
Iris-versicolor	Iris-virginica
Iris-setosa	Iris-setosa
Iris-setosa	Iris-setosa
Iris-virginica	Iris-virginica
Iris-versicolor	Iris-versicolor
Iris-virginica	Iris-virginica
Iris-versicolor	Iris-versicolor
Iris-setosa	Iris-setosa
Iris-virginica	Iris-virginica
Iris-setosa	Iris-setosa
Iris-setosa	Iris-setosa
Iris-versicolor	Iris-versicolor

1 Lentelė. Testavimo aibės rezultatai.

3. Modelis su testavimo ir mokymo duomenimis



4 paveikslas. Modelis kuriam pateikiami abu duomenų failai.

Sukonstruojamas modelis su vienu paslėptu neuronų sluoksniu su penkiais neuronais. Šiam modeliui pateikiami abu duomenų failai. Vienas perduodamas naudojant „Training Set Maker“ komponentą, kitas „Test Set Maker“ komponentą.

Modelis gražina testavimo aibės rezultatus kaip tikrosios duomenų įrašo klasės tikimybes (2 lentelė).

'Iris-setosa'	'Iris-versicolor'	'Iris-virginica'
0,002476	0,965634	0,03189
0,006578	0,990813	0,002609
0,987026	0,012961	0,000013
0,003777	0,988176	0,008048
0,986162	0,013825	0,000013
0,000313	0,103862	0,895825

2 lentelė. Modelio gražinamos tikimybės.

Panaudojus šio modelio svorius (3 ir 4 lentelė) atliekamas eksperimentas patikrinti ar programos skaičiavimas sutampa su teoriniu neurono veikimo principu.

	svoriai					
	Poslinkis	node 3	node 4	node 5	node 6	node 7
node 0	0,270752	4,707632741	-1,950592369	-1,479361994	-1,725329871	-4,679372289
node 1	-0,59994	-4,163169692	2,183208674	-5,486631941	-8,504183513	6,059846784
node 2	-4,76379	-6,523296709	-0,842197884	4,849734787	7,055220532	0,233912698

3 lentelė. Išėjimų neuronų svoriai.

	svoris				
	Poslinkis	Taurelapio_ilgis	Taurelapio_plotis	Ziedlapio_ilgis	Ziedlapio_plotis
node 3	-0,2776	0,536372035	1,880258891	-2,530880377	-2,426148219
node 4	1,418462	1,02688576	-1,18415481	1,591323863	1,483697438
node 5	-3,69432	-1,232042976	-2,605975175	6,161171396	3,922295699
node 6	-4,95946	-2,209870008	-4,436275936	9,084946367	4,711680947
node 7	2,535927	0,848973093	-2,324192115	3,222015134	3,265504801

4 lentelė. Paslėpto sluoksnio neuronų įėjimų svoriai.

4. Skaičiavimų patikrinimas su MS „Excel“

Naudojant WEKA sukonstruoto modelio svorius MS „Excel“ programoje suskaičiuojamos išėjimo neuronų reikšmės bei pritaikoma sigmoidinė aktyvacijos funkcija. Šiuos reikšmės paverčiamos tikimybėmis. Kadangi visos reikšmės teigiamos, naudojama formulė $\frac{x_i}{\sum_n^j x_j}$.

Gautos tikimybės sulyginamos. Pastebima, kad tikimybės neatitinka tiksliai, todėl bandomas tam tikras skaičius po kablelio (5 lentelė). Tikėtina, kad WEKA naudoja kitokią formulę išėjimo neuronų vertes versdama į tikimybes.

Suapvalinta iki 3-jų skaičių po kablelio			Suapvalinta iki 2-jų skaičių po kablelio		
0,003	0,968	0,029	0	0,97	0,03
0,006	0,991	0,003	0,01	0,99	0
0,989	0,011	0	0,99	0,01	0
0,004	0,989	0,007	0	0,99	0,01
0,988	0,012	0	0,99	0,01	0
0	0,032	0,968	0	0,03	0,97
0	0,002	0,998	0	0	1
0,991	0,009	0	0,99	0,01	0
0,008	0,99	0,001	0,01	0,99	0
0	0,003	0,997	0	0	1
0,002	0,938	0,06	0	0,94	0,06
0	0,002	0,998	0	0	1
0	0,01	0,99	0	0,01	0,99
0,988	0,012	0	0,99	0,01	0
0	0,002	0,998	0	0	1
0	0,002	0,998	0	0	1
0,986	0,014	0	0,99	0,01	0
0	0,002	0,998	0	0	1
0	0,033	0,967	0	0,03	0,97
0,989	0,011	0	0,99	0,01	0
0,988	0,012	0	0,99	0,01	0
0	0,003	0,997	0	0	1
0,012	0,987	0,001	0,01	0,99	0
0	0,057	0,942	0	0,06	0,94
0,009	0,99	0,002	0,01	0,99	0
0,989	0,011	0	0,99	0,01	0
0	0,002	0,998	0	0	1
0,989	0,011	0	0,99	0,01	0
0,987	0,013	0	0,99	0,01	0
0,007	0,991	0,003	0,01	0,99	0

5 lentelė. „Excel“ skaičiavimų rezultatas. Žaliai pažymėtas visiškas rezultatų sutapimas.

Išvados.

WEKA yra teisingai veikiantis įrankis duomenų analizės procese. Vartotojas yra apribotas naudoti tam tikro formato failus, todėl duomenų pateikimas reikalauja programavimo žinių duomenų tvarkyme ir transformavime. Nepaisant to, hiperparametrų parinkimas, kryžminė validacija, modelio ir duomenų vizualizavimas (1 priedas) patogiai integruotas į vartotojo sąsają. Perskaičiavus gautus modelio rezultatus su MS „Excel“ pastebėta, kad išėjimo reikmės paverčiamos į tikimybes, kurių bendra suma yra lygi 1. Pabandžius tai atkartoti pastebimas tik dalinis atitikimas (daugumoje atvejų, atitinka tik suapvalinus iki 2 skaičių po kablelio). Nėra pagrindo manyti, kad skaičiavimai yra neteisingi, nes galutinis klasių priskyrimas sutampa su teoriniais skaičiavimais.

1 priedas. Duomenų priklausomybės vizualizacija.