



Intelligent Systems

Laboratory activity 2016-2017

Adrian Groza, Anca Marginean and Radu Razvan Slavescu
Tool:

Name: Dreghiciu Anca
Group: 30235
Email: anca_dreghiciu_1997@yahoo.com

Assoc. Prof. dr. eng. Adrian Groza
Adrian.Groza@cs.utcluj.ro



Contents

1	Descrierea proiectului (W_5)	3
1.1	Descrierea Problemei Perceptron	3
1.2	Descrierea Retelelor Neuronale	4
1.3	Descrierea Arborilor Decizionali	4
1.4	Descrierea K-Means	5
1.5	KNN	5
1.6	SVM	5
A	Your original code	6
A.1	Perceptron	6
A.2	Retele Neuronale	6
A.3	Decision Tree	6

Chapter 1

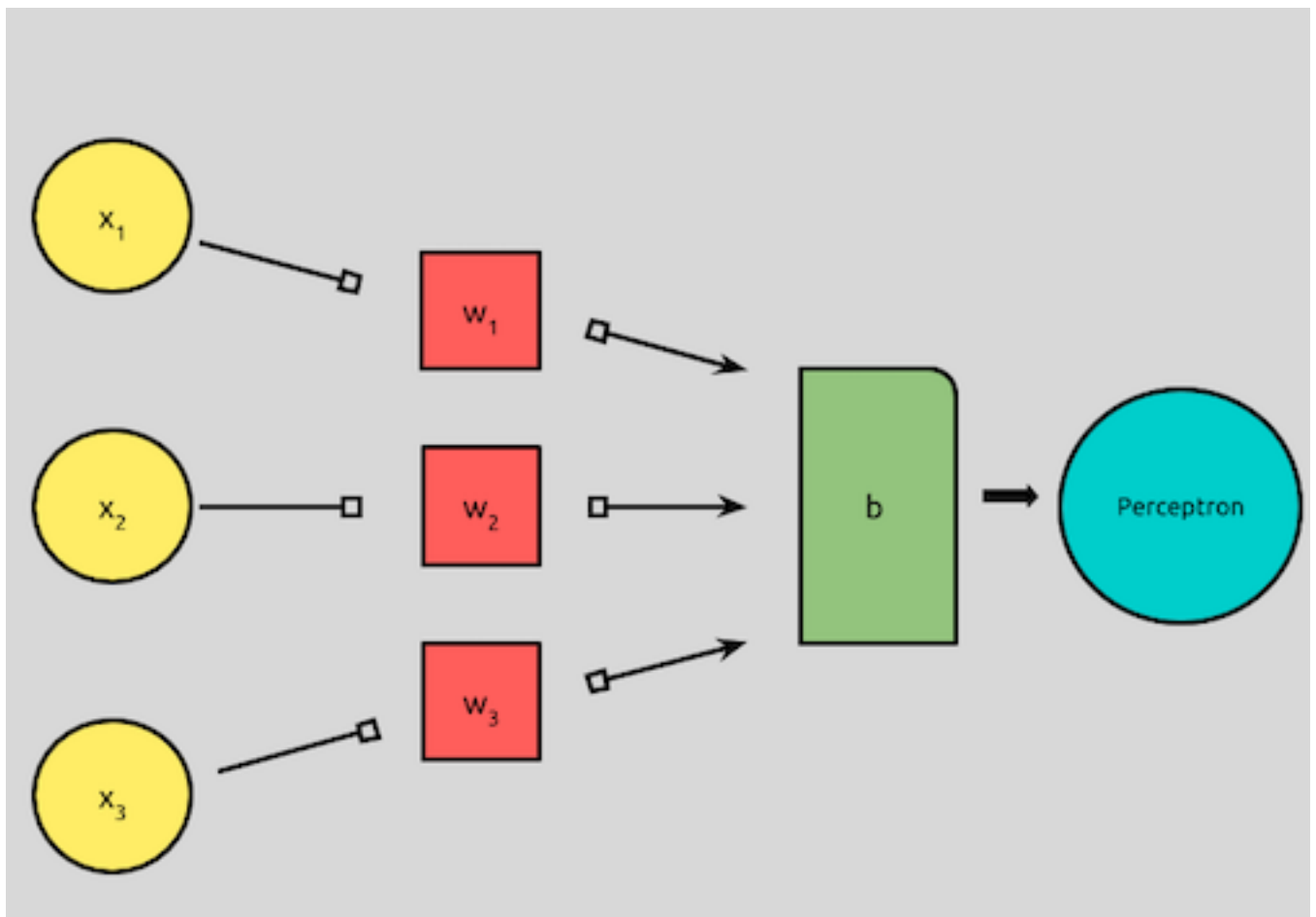
Descrierea proiectului (W_5)

Pentru acest proiect am pus in aplicare diferite tipuri de probleme din machine learning. Avand mai multe categorii, am parcurs pentru fiecare cateva exemple. Aceste categorii de probleme sunt: probleme bazate pe invatare supervizata, invatare nesupervizata si reinforcement learning.

Proiectul se bazeaza pe dezvoltarea capacitatii de a intelege si a putea prelucra date, sub diferite forme, si de a alege modalitatea cea mai eficienta de a rezolva anumite probleme.

1.1 Descrierea Problemei Perceptron

Pentru o prima abordare a retelelor neurale, am implementat un Perceptron. Un perceptron este un algoritm ce implementeaza o functie liniara. Acesta poate avea oricate "inputuri", dar produce o singura data de iesire, care se mai numeste si activator. Fiecare input are o greutate/prioritate. Pentru a activa perceptronul se va face suma greutatilor si se va verifica daca este peste un prag.

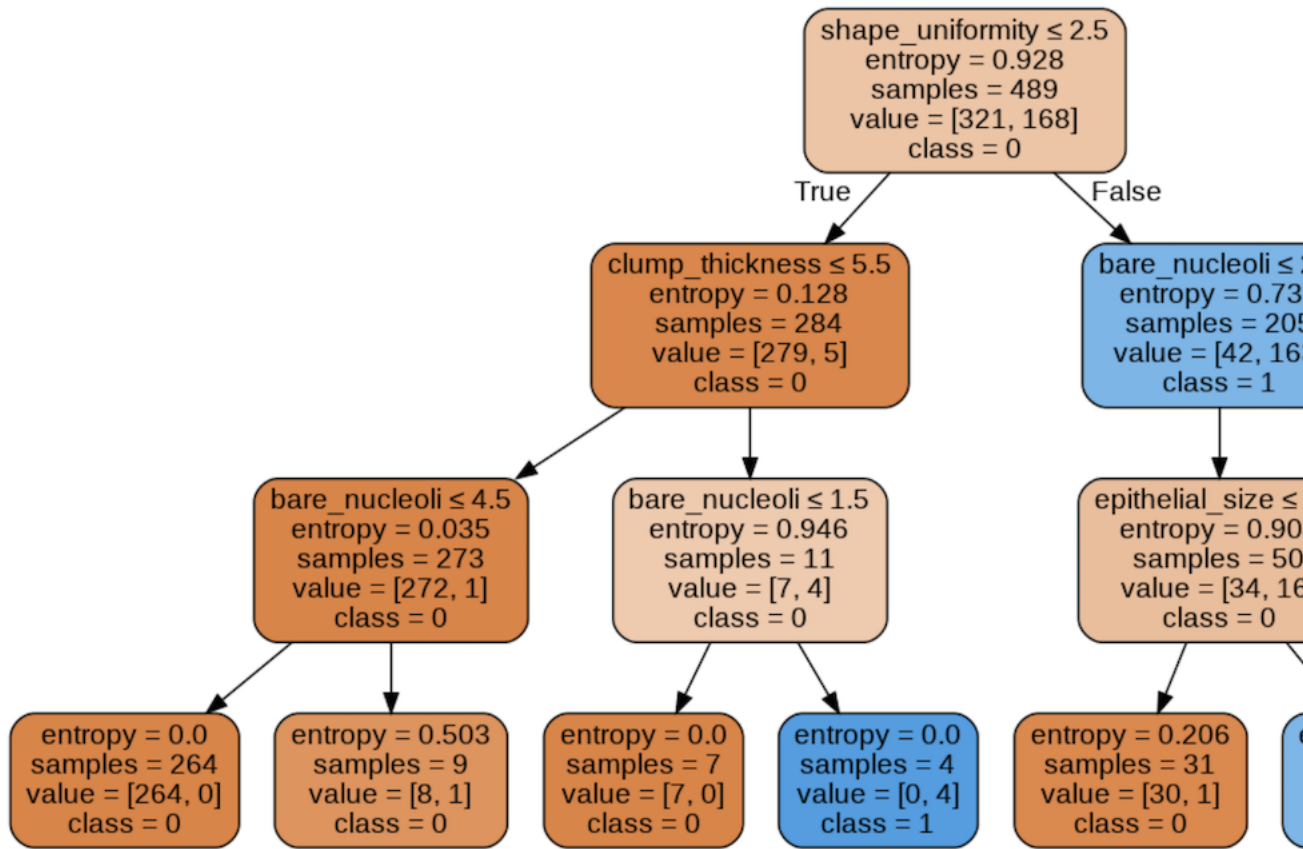


1.2 Descrierea Retelelor Neuronale

O retea neuronală este o combinație de mai mulți neuroni conectați între ei. O retea poate avea mai multe nivele, unul de input și unul de output, ce sunt necesare, iar între acestea se mai pot construi mai multe nivele numite nivele ascunse. Aceasta se poate învăța din datele pe care le primește, astfel se pot calcula pierderile pe care le are această retea, adică cât de bine se descurcă rețeaua. Prin învățare se încearcă obținerea unei pierderi cât mai mici. Noi am folosit MSE (Mean Squared Error) care este diferența dintre data așteptată și data prezisă la pătrat. Se poate itera de mai multe ori prin date. Aceste iterații complete se numesc epoci. În cazul acestui context intra și problemele de regresie liniară și predicția diabetului.

1.3 Descrierea Arborilor Decizionali

Un arbore decizional este sub forma unui arbore flow-chart, unde fiecare nod reprezintă un feature și o ramură reprezintă o decizie, iar frunzele reprezintă rezultatul tuturor deciziilor. Prin aceasta se poate realiza învățarea mai rapid decât printr-o rețea neuronală. Acest algoritm selectează datele relevante din setul de date, și compară datele. Selectarea datelor se poate face după Information Gain, Gain Ratio, și Gini Index.



1.4 Descrierea K-Means

Algoritmul K-Means Clustering este un algoritm de învățare nesupervizată, acesta încearcă găsirea de similitudini între date, nu are Output ci doar input. Clusterizarea este gruparea unui set de obiecte în diferite cluster, care au mai multe similitudini cu obiectele din același cluster decât cu cele din alt cluster. Similaritatea este o metrică ce reflectă puterea unei relații între 2 obiecte. În principal este un algoritm bazat pe clusterizarea centrozilor, funcționând iterativ, în care similaritatea este dată de cât de aproape este data căutată, de centroidul unui cluster.

1.5 KNN

K Nearest Neighbor este un algoritm de clasificare, non-parametric. Structura modelului este determinată de datele sale. Algoritmul ia un număr k pentru un punct și caută cele mai apropiate puncte față de acel punct inițial, iar k este numărul vecinilor. Dacă k este un număr mic, va avea un zgomot mare ce se va reflecta în rezultat, iar un k mai mare va face ca costul calculului să crească.

1.6 SVM

Support Vector Machine este un algoritm de clasificare, de învățare supervizată. El separă datele folosind un plan cu cea mai largă margine, astfel el împarte datele în cel puțin 2 categorii. Vectorii suport sunt cei mai apropiați de plan, aceștia definesc linia, pentru a calcula marginile planului. Planul este un spațiu de decizie care împarte datele în clasele sale.

Appendix A

Your original code

A.1 Perceptron

```
training_inputs = [] training_inputs.append(np.array([1, 1])) training_inputs.append(np.array([1, 0])) training_inputs.append(np.array([0, 1])) training_inputs.append(np.array([0, 0]))
labels = np.array([0, 1, 1, 0])
perceptron = Perceptron(2) perceptron.train(training_inputs, labels)
inputs = np.array([1, 1]) print(perceptron.predict(inputs)) = 1
inputs = np.array([0, 1]) print(perceptron.predict(inputs)) = 0
```

A.2 Retele Neuronale

Pentru retele neuronale am schimbat functia sigmoid cu cea Relu si functia sa derivativa. Apoi am adaugat inca un nivel ascuns in retea si avand astfel 5 neuroni ascunsi.

A.3 Decision Tree

Pentru aceasta functie am schimbat setul de date cu cel pentru Cancer Mamar.

Bibliography

Intelligent Systems Group

