

Inteligencja obliczeniowa

Laboratorium 6: Sieci neuronowe.

Na dzisiejszych laboratoriach poznamy kolejny algorytm inspirowany biologicznie (wcześniej mieliśmy algorytmy genetyczne), który pozwoli na klasyfikowanie rekordów, ale jest też stosowany do szacunków, obliczeń, przewidywań. Są to (sztuczne) sieci neuronowe, które przypominają nieco graf, a nieco również sieć neuronów w naszych mózgach.

Zadanie 1

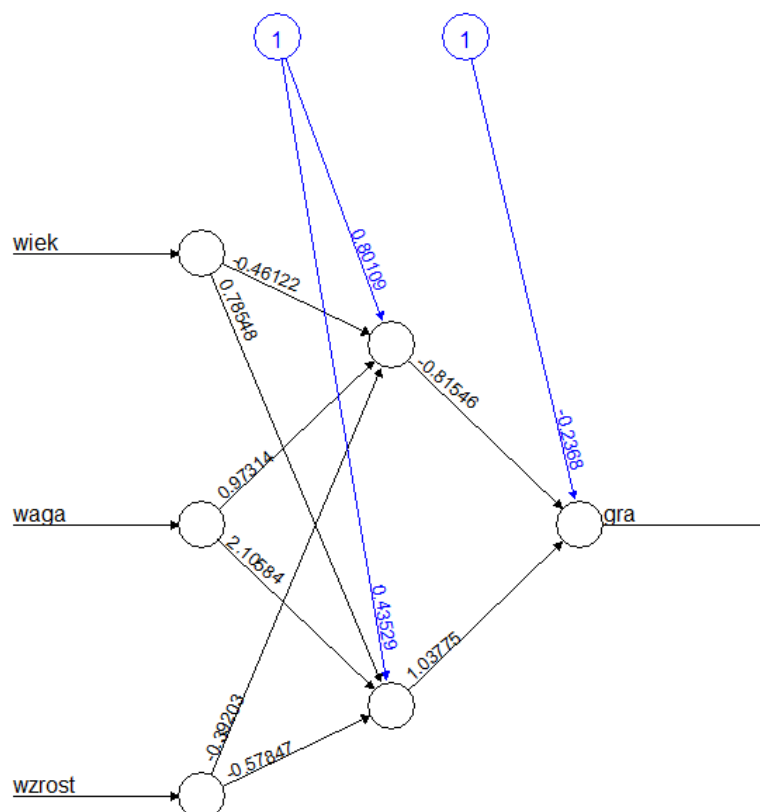
Zapoznamy się z działaniem sieci neuronowej na prostym przykładzie. Mamy małą bazę danych ludzi, których zachęcaliśmy do gry w siatkówkę na plaży (a przy okazji popytaliśmy o parametry ich organizmu). Część ludzi zgodziła się zagrać, a część nie.

wiek	waga	wzrost	gra
23	75	176	TRUE
25	67	180	TRUE
28	120	175	FALSE
22	65	165	TRUE
46	70	187	TRUE
50	68	180	FALSE
48	97	178	FALSE

Korzystając z R i paczki neuralnet:

```
siatk.nn <- neuralnet(gra~wiek+waga+wzrost, siatk.dane, hidden=2, lifesign="full")  
plot(siatk.nn)
```

stworzyliśmy sieć neuronową o następującej strukturze (nie musisz wykonywać tego kroku, przeczytaj najpierw zadanie):



Error: 0.40071 Steps: 114

Sieć neuronowa, niczym mały sztuczny mózg, miała nauczyć się rozpoznawać, które osoby będą chciały grać w siatkówkę. Sprawdziliśmy jak sieć poradziła sobie z naszymi siedmioma osobami. Korzystając z komendy:

```
siatk.predict <- compute(siatk.nn, siatk.dane[1:3])
```

i otrzymaliśmy wynik (tego kroku też nie musisz wykonywać):

[1,]	0.79852846757
[2,]	0.80094344704
[3,]	-0.01451814786
[4,]	0.80092650708
[5,]	0.80094353050
[6,]	0.80094353051
[7,]	0.01518550709

Widać, że gdyby zaokrąglić wartości do 1 (TRUE) i 0 (FALSE) to nasza sieć wypadła całkiem nieźle. Nie zgadła jedynie rekordu 6.

Pytanie: jak obliczone zostały powyższe wartości?

Sieć neuronowa dostaje trzy dane na wejściu. Wartości liczbowe są mnożone przez wagi na strzałkach i przekazywane kolejnym neuronom, które sumują wszystko co dostają. Dodane są też do tego wartości dodatkowe (bias) zaznaczone na niebiesko. Zsumowane wartości przepuszczone są przez funkcję aktywacji (fct.act) wynoszącą w tym przypadku $fct.act(x) = 1/(1+\exp(-x))$, po czym neuron przesyła je dalej (funkcja aktywacja działa tylko dla neuronów ukrytych).

Napisz prostą funkcję, która sprawdzi czy sieć działa dobrze tj. funkcja na input dostanie wiek, wagę, wzrost, a na output zwróci liczbę przewidyującą granie w siatkówkę.

```
forwardPass <- function(wiek, waga, wzrost){  
  hiddenNeuron1 = (tutaj wzór – mnożenie, sumowanie z wykorzystaniem wag ze schematu sieci neuronowej, funkcja aktywacji)  
  hiddenNeuron2 = (tutaj wzór – mnożenie, sumowanie z wykorzystaniem wag ze schematu sieci neuronowej, funkcja aktywacji)  
  output = (tutaj wzór – mnożenie, sumowanie)  
  return(output)  
}
```

Przykład działania: `forwardPass(23,75,176) = 0.798528`

Sprawdź jej działanie dla dwóch wybranych przez ciebie rekordów.

W razie problemów zachęcam do skorzystania z Google / Youtube (wyszukanie: sieć neuronowa, perceptron, neural network) lub z wykładów.

Praca domowa

Zadanie 2

Dokonyamy klasyfikacji irysów za pomocą sieci neuronowej.

a) Znormalizuj kolumny liczbowe bazy danych irysów, wzorem $(x-\min(x))/(\max(x)-\min(x))$.

b) Podziel zbiór irysów na `iris.train`, `iris.test` tak, jak robiliśmy to przy innych algorytmach klasyfikujących. Proporcje mogą być 67/33.

c) Zainstaluj i załaduj paczkę `neuralnet`. Uruchom komendę `neuralnet` na zbiorze treningowym. Problem może stanowić klasa, która nie ma danych logicznych (TRUE/FALSE), dlatego możesz posłkować się następującym pomysłem:

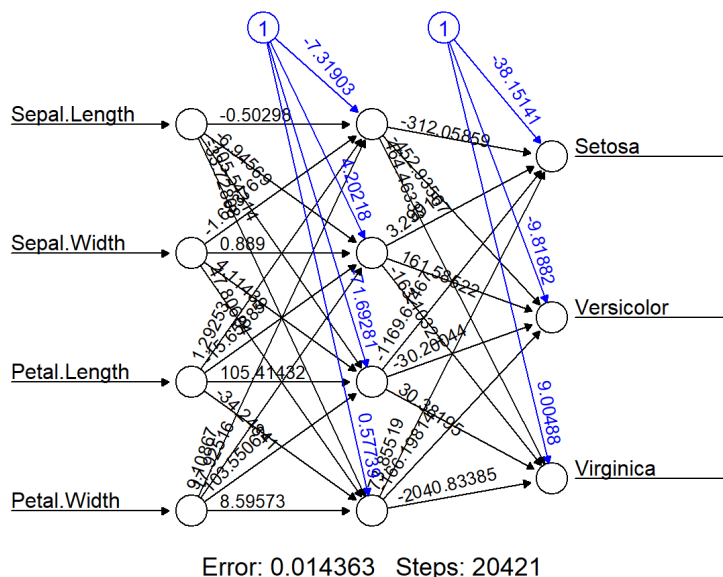
- Rozbij kolumnę `Species` na kolumnę `Setosa`, `Versicolor`, `Virginica`. Łącznie będzie w bazie 7

kolumn, 4 numeryczne i 3 logiczne.

- W kolumnie Setosa będą wartości 0 (dany irys nie jest setosą) i 1 (dany irys jest setosą). Podobnie będą działały pozostałe kolumny z gatunkami.
- Wszystkie trzy kolumny będą brane pod uwagę jako klasa. Należy uwzględnić je uruchamiając sieć neuronową:

```
neuralnet(Setosa + Versicolor + Virginica ~ Sepal.Length + Sepal.Width +  
Petal.Length + Petal.Width, ...
```

Wówczas sieć neuronowa będzie miała trzy neurony wyjściowe.



d) Zewaluuuj klasyfikator uruchamiając komendę compute na zbiorze testowym. Podaj macierz błędu i dokładność klasyfikatora.

Zadanie 3

Skorzystaj z sieci neuronowych, by dokonać klasyfikacji na zbiorze diabetes.csv, po wcześniejszym znormalizowaniu danych. Poeksperymentuj z liczbą ukrytych neuronów (hidden). Zewaluuuj klasyfikator i nanieś go na wykres (FP-Rate, TP-Rate) obok wcześniej poznanych klasyfikatorów. Czy jest lepszy?

*Zadanie 4 (dla chętnych)

Bardzo modną dziedziną jest tzw. Deep Learning, czyli wykorzystywanie sieci neuronowych z wieloma warstwami, np. konwolucyjnych sieci neuronowych, do rozwiązywania skomplikowanych zadań. Wykorzystaj wybraną paczkę w R i bazę danych obrazków, by przetestować jak działają techniki typu deep learning/convolutional neural networks do rozpoznawania obrazków.

Przydatne linki:

<https://www.r-bloggers.com/image-recognition-tutorial-in-r-using-deep-convolutional-neural-networks-mxnet-package/>

<http://dmlc.ml/rstats/2015/11/03/training-deep-net-with-R.html>

<https://www.r-bloggers.com/deep-learning-in-r-2/>