

Podstawy Sztucznej Inteligencji- Sprawozdanie

Scenariusz 2

Opis ćwiczenia

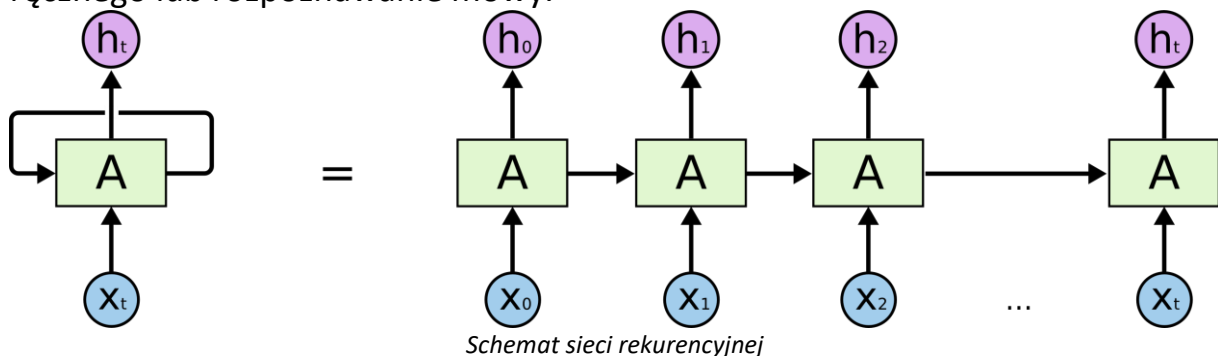
Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

Zadania do wykonania

- Wygenerowanie danych uczących i testujących, zawierających 10 dużych i 10 małych liter dowolnie wybranego alfabetu w postaci dwuwymiarowej tablicy dla jednej litery.
- Przygotowanie dwóch jednowarstwowych sieci- każda wg. Innego algorytmu podanego na wykładzie.
- Uczenie sieci przy różnych współczynnikach uczenia.
- Testowanie sieci.

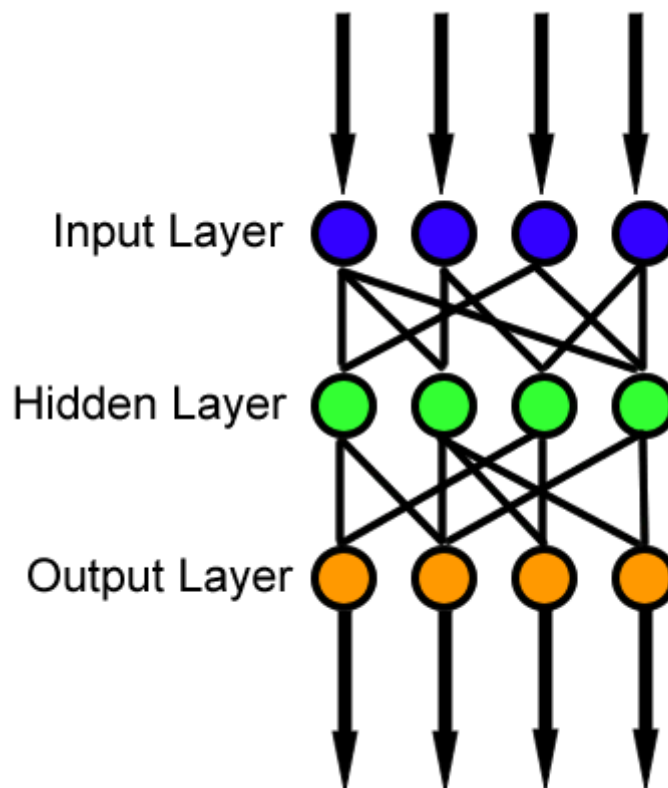
Specyfikacja

Sieci, które przygotowałem to sieć typu Recurrent i FeedForward. Sieć typu Recurrent jest klasą sztucznej sieci neuronowej, w której połączenia pomiędzy jednostkami tworzą cykl ukierunkowany. Dzięki temu może wykazywać dynamiczne tymczasowe zachowanie. W przeciwieństwie do feedforward neuronowych sieci, RNNs mogą używać ich wewnętrznej pamięci do przetwarzania dowolnych sekwencji wejść. Dzięki temu mają one zastosowanie do takich zadań, jak niesegregowane, połączone rozpoznawanie pisma ręcznego lub rozpoznawanie mowy.



Sieć typu FeedForward to sztuczna sieć neuronowa, w której połączenia nie formują cykli. Jest to pierwszy i najprostszy typ sztucznej sieci neuronowej.

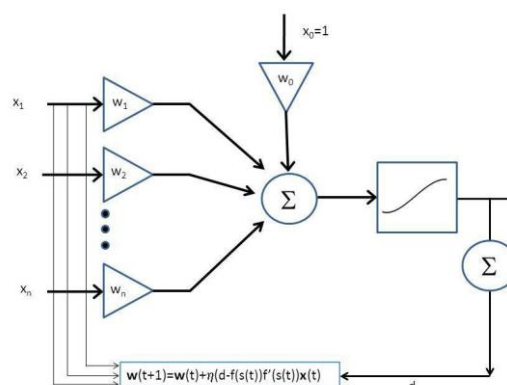
Informacje w niej poruszają tylko w jednym kierunku, poprzez warstwę wprowadzającą, poprzez warstwę ukrytą do warstwy wyjścia.



Schemat sieci typu FeedForward

W programie korzystałem z dwóch typów neuronów w warstwie ukrytej- tanh i sgmoid. Neuron tanh jest modyfikacją neurona liniowego, jednak funkcja aktywacji jest hyperboliczna, a dokładnie jest to tangens hiperboliczny. Neuron sigmoidalny jest zbliżony do perceptronu. Jego funkcje aktywacji przybierają postać funkcji sigmoidalnej unipolarnej lub bipolarnej. Są to funkcję ciągłe. Zaletą takich neuronów jest różniczkowalność funkcji aktywacji.

Model neuronu sigmoidalnego



Algorytm, którego użyłem to propagacja wsteczna. To podstawowy algorytm uczenia nadzorowanego wielowarstwowych, jednokierunkowych sieci neuronowych. Oparty jest na minimalizacji sumy kwadratów błędów uczenia z wykorzystaniem optymalizacyjnej metody największego spadku.

Schemat krokowy tej funkcji wygląda tak:

- 1) Inicjalizacja sieci i algorytmu.
- 2) Obliczanie wartości wyjściowej sieci na podstawie danych.
- 3) Obliczanie błędu sieci.
- 4) Korekcja wag.
- 5) Czy sieć jest nauczona?
 - a) Tak- przejdź dalej.
 - b) Nie- wróć do 2 punktu.
- 6) Koniec

1 przebieg algorytmu nazywa się epoką.

Wykonanie zadania

Do wykonania ćwiczenia użyto wykonanego zestawu danych z małymi i dużymi literami. W Obu sieciach do rozpoznawania wielkości liter użyto 5 neuronów. Algorytm, który został zastosowany to algorytm propagacji wstecznej.

Wyniki

Przykładowe zrzuty ekranu z procesu testowania:

Learning rate=0.1, epoki=100

untitled - [C:\Users\Grzegorz Planeta\PycharmProjects\untitled] - ...|trener.py

File Edit View Navigate Code Help

Run: trener

```

MSE: 0.000361378090163
dia litera J precyzja wynosi: 1.03639902669
MSE: 0.00132488914419
litera: J to DUZA litera
MSE: 0.000427622547372
dia litera a precyzja wynosi: 0.0206603320868
MSE: 0.00042684321537
litera: a to mala LITERA
MSE: 0.000448965013469
dia litery b precyzja wynosi: -0.0348553017265
MSE: 0.00121489205844
litera: b to mala LITERA
MSE: 0.000509709616391
dia litery c precyzja wynosi: 0.00785786357889
MSE: 6.17460200244e-05
litera: c to mala LITERA
MSE: 0.000512796917392
dia litery d precyzja wynosi: 0.00400974252507
MSE: 1.60780351174e-05
litera: d to mala LITERA
MSE: 0.000513600819148
dia litera e precyzja wynosi: 0.0193757270437
MSE: 0.000375418798472
litera: e to mala LITERA
MSE: 0.000533291759072
dia litery f precyzja wynosi: 0.0238482516348
MSE: 0.000568739106035
litera: f to mala LITERA
MSE: 0.0005680808714374
dia litery g precyzja wynosi: -0.0170567683977
MSE: 0.000290933348172
litera: g to mala LITERA
MSE: 0.000575355381782
dia litery h precyzja wynosi: 0.0471715181779
MSE: 0.00222515212721
litera: h to mala LITERA

```

Learning rate=0.1, epoki=500

```
untitled - [C:\Users\Grzegorz Planeta\PycharmProjects\untitled] - ...tr
File Edit View Navigate Code Help
Run: trener trener
1: Project
litera: A to DUZA litera
MSE: 9.15721362186e-07
dla litery B precyzja wynosi: 1.0172874959
MSE: 0.000298857514475
litera: B to DUZA litera
MSE: 1.5858597086e-05
dla litery C precyzja wynosi: 1.00480872867
MSE: 2.31238714335e-05
litera: C to DUZA litera
MSE: 1.70147906576e-05
dla litery D precyzja wynosi: 1.00165029186
MSE: 2.72346323183e-06
litera: D to DUZA litera
MSE: 1.71509638192e-05
dla litery E precyzja wynosi: 1.00646786127
MSE: 4.18332294223e-05
litera: E to DUZA litera
MSE: 1.92426252903e-05
dla litery F precyzja wynosi: 1.00529879945
MSE: 2.80772755848e-05
litera: F to DUZA litera
MSE: 2.06464890696e-05
dla litery G precyzja wynosi: 1.00982469882
MSE: 9.65247068358e-05
litera: G to DUZA litera
MSE: 2.54727244114e-05
dla litery H precyzja wynosi: 0.964393958745
MSE: 0.00126779017383
litera: H to DUZA litera
MSE: 8.88622331029e-05
dla litery I precyzja wynosi: 0.969798287812
MSE: 0.000912143419103
litera: I to DUZA litera
MSE: 0.000134469404058
dla litery J precyzja wynosi: 1.00140667294
MSE: 1.97872875506e-06
```

Learning rate=0,1, epoki=1000

```
untitled - [C:\Users\Grzegorz Planeta\PycharmProjects\untitled] - ...trener.p
File Edit View Navigate Code Help
Run: trener trener
1: Project
MSE: 1.34688871842e-05
dla litery D precyzja wynosi: 1.01333588097
MSE: 0.000177845721127
litera: D to DUZA litera
MSE: 2.23611732405e-05
dla litery E precyzja wynosi: 1.00102911861
MSE: 1.05908511216e-06
litera: E to DUZA litera
MSE: 2.24141274961e-05
dla litery F precyzja wynosi: 1.00682493639
MSE: 4.65797567804e-05
litera: F to DUZA litera
MSE: 2.47431153351e-05
dla litery G precyzja wynosi: 1.01306799431
MSE: 0.000170772475339
litera: G to DUZA litera
MSE: 3.32817391021e-05
dla litery H precyzja wynosi: 0.998559887451
MSE: 2.07392415491e-06
litera: H to DUZA litera
MSE: 3.33854353098e-05
dla litery I precyzja wynosi: 0.995278439327
MSE: 2.22931351876e-05
litera: I to DUZA litera
MSE: 3.45000920692e-05
dla litery J precyzja wynosi: 1.00807012705
MSE: 6.51269505419e-05
litera: J to DUZA litera
MSE: 3.77564395963e-05
dla litery a precyzja wynosi: 0.00521716627999
MSE: 2.7218823993e-05
litera: a TO mala LITERA
MSE: 3.9117380796e-05
dla litery b precyzja wynosi: -0.00315320136563
MSE: 9.9426788522e-06
litera: b TO mala LITERA
```

Analiza

Pierwszą siecią, która była testowana to sieć rekurencyjna z neuronami typu tanh. Learning rate=0.1 sprawił, że otrzymanie wystarczająco niskiego błędu było możliwe przy 1100 epokach. Przy learning rate wynoszącym 0.5 po 600 epokach sieć była odpowiednio nauczona. Sieć za każdym razem poprawnie rozpoznawała wielkość liter. Dla neuronów sigmoidalnych:

- Learning rate=0.1, optymalna ilość epok=1400
- Learning rate=0.5, optymalna ilość epok=300

Następna konfiguracja to sieć FeedForward z neuronami sigmoidalnymi. Przy learning rate wynoszącym 0.1 odpowiednio niski błąd (ok. 0.001 lub mniej) był otrzymywany przy 900 epokach. Dla learning rate wynoszącego 0.5 pożądany wynik był osiąganym dla 150 epok. Dla neuronów typu tanh:

- Learning rate=0.1, optymalna ilość epok=800
- Learning rate=0.5, optymalna ilość epok=400

Wnioski

Na podstawie wyników, które otrzymałem i przeanalizowałem mogę stwierdzić, że do tego typu zastosowań sieć rekurencyjna nie jest optymalna, gdyż działa wolniej od sieci FeedForward. Oprócz tego można zauważyć, że neurony sigmoidalne lepiej zachowywały się przy wysokim learning rate. Za to przy małym learning rate lepiej zachowywały się neurony typu tanh.