Grzegorz Płaneta

**Podstawy Sztucznej Inteligencji- Sprawozdanie**

**Scenariusz 2**

**Opis ćwiczenia**

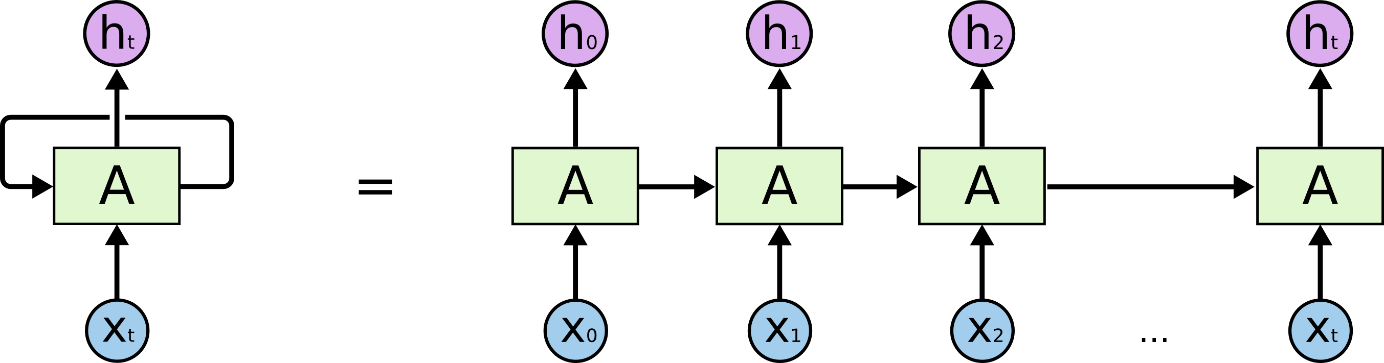
Celem ćwiczenia jest poznanie budowy i działania jednowarstwowych sieci neuronowych oraz uczenie rozpoznawania wielkości liter.

**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.

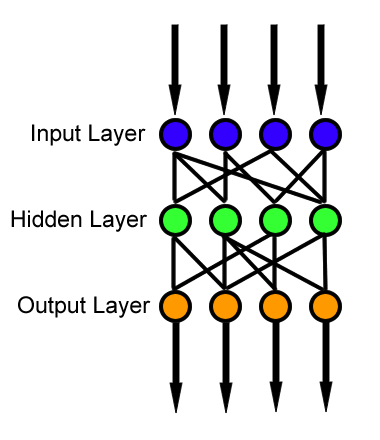
**Specyfikacja**

Sieci, które przygotowałem to sieć typu Reccurent 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.



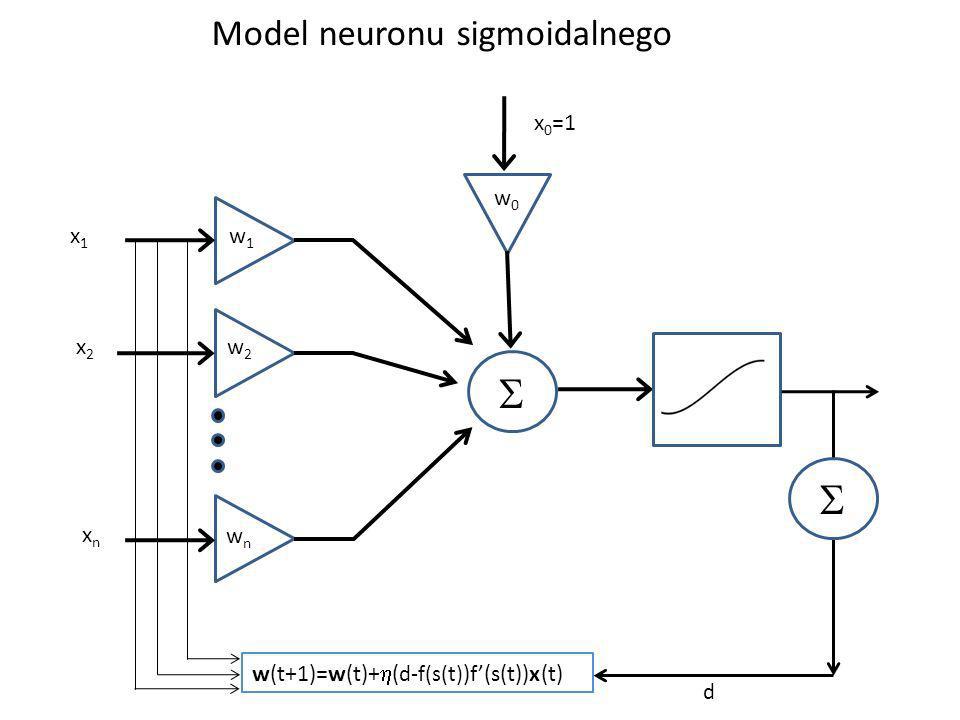
*Schemat sieci rekurencyjnej*

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.



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?
   1. Tak- przejdź dalej.
   2. Nie- wróć do 2 punktu.
6. Koniec

1 przebieg algorytmu nazywa się epoką.

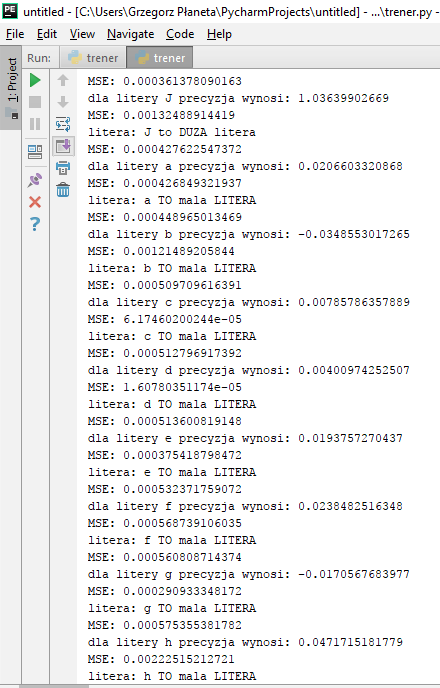
**Wykonanie zadania**

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.

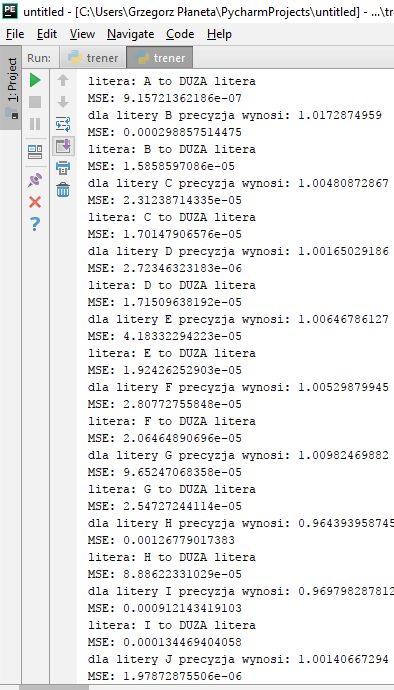
**Wyniki**

Przykładowe zrzuty ekranu z procesu testowania:

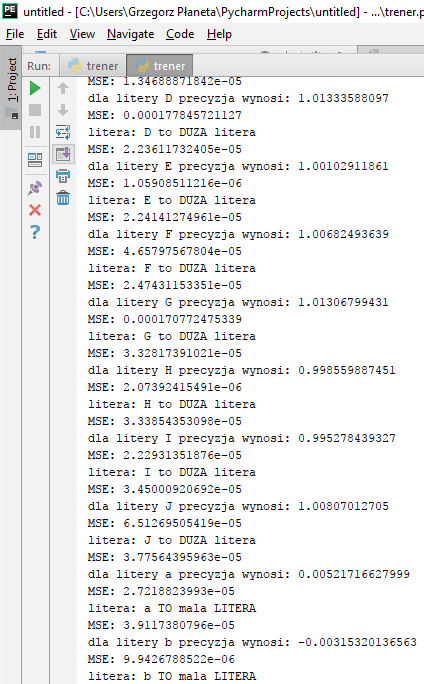
Learning rate=0.1, epoki=100



Learning rate=0.1, epoki=500



Learing rate=0,1, epoki=1000



**Analiza**

Pierwszą siecią, która była testowana to sieć rekurencyjna z neuronami typu tanh. Learnign 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:

* Learing rate=0.1, optymalna ilość epok=1400
* Learing rate=0.5, optymalna ilość epok=300

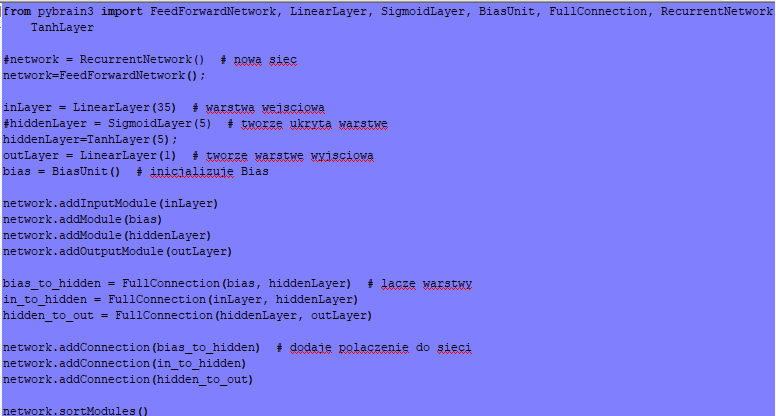
Następna konfiguracja to sieć FeedForward z neuronami sigmoidalnymi. Przy learning rate wynoszącym 0.1 odpowiednio niski błąd (ok. 0.001 lub mniej) był otrzymywany przy 900 epokach. Dla learing rate wynoszącego 0.5 pożądany wynik był osiągany dla 150 epok. Dla neuronów typu tanh:

* Learning rate=0.1, optymalna ilość epok=800
* Learing rate=0.5, optymalna ilość epok=400

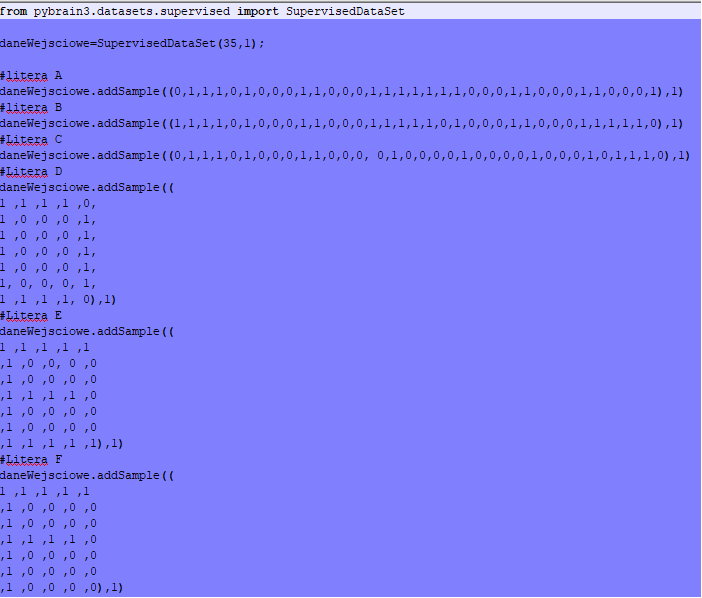
**Wnioski**

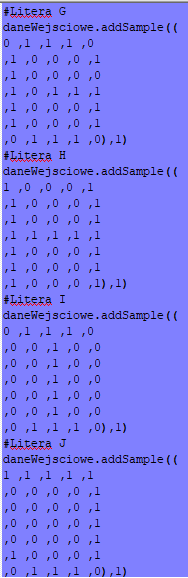
Na podstawie wyników, które otrzymałem i przeanalizowałem mogę stwierdzić, że do tego typu zastosowań sieć rekurencyjna nie jest optymalna, gdyż działa wolniej od sieci FeedForward. Oprócz tego można zauważyć, że neurony sigmoidalne lepiej zachowywały się przy wysokim learning rate. Za to przy małym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

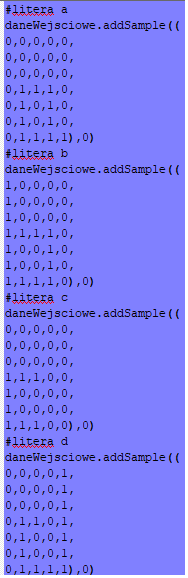
**Kod programu:**

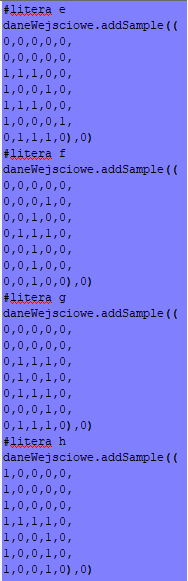
**Klasa „siec”: **

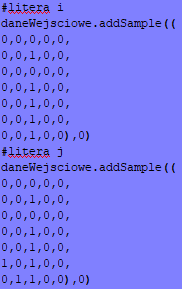
**Klasa „dane”:**

****

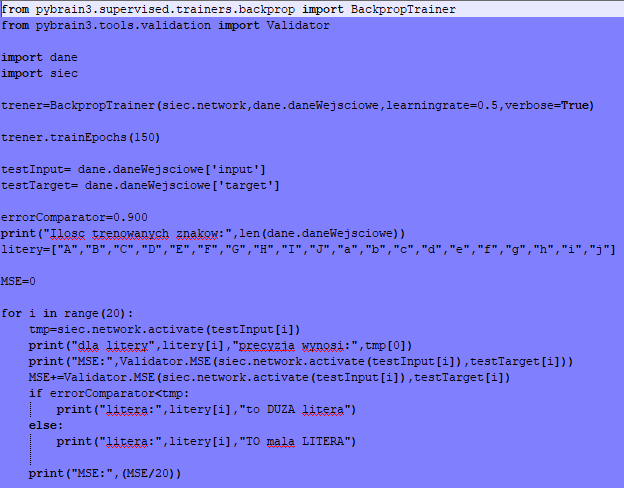
****

****

****

****

**Klasa „trener”:**

****