31.01.2020

Sprawozdanie z projektu

Temat projektu: Przewidywanie czasów retencji peptydów na podstawie ich sekwencji przy użyciu sieci neuronowej.

1. Wstęp i realizacja projektu
   Projekt miał na celu stworzenie aplikacji, która będzie służyła do przewidywania czasów retencji peptydów na podstawie ich sekwencji przy wykorzystaniu sieci neuronowych.

2. Implementacja programu
   Program został napisany w języku Python, składa się z czterech plików:
   1. main.py→ głowny plik wykonawczy programu
   2. readpeptides.py→ implementacja odczytu peptydów z pliku oraz podział względem białek oraz przypisanie wartości do tablic
   3. multilayeredneuralnetwork.py→ zaimplementowana sieć neuronowa składająca się z dwóch klas:
      a. class NeuronLayer()→ zdefiniowanie rozmiaru sieci poprzez podanie liczby neuronów oraz ilości wejść do każdego neuronu
      b. class NeuralNetwork()→ implementacja obliczeń wykonywanych przez sieć
   4. savereadsynapticweights.py→ zaimplementowano funkcje zapisujące oraz odczytujące obliczone wagi dla neuronów do plików

3. Dane uczące
   Dane uczące zostały udostępnionne wraz z projektem na gitlab. Zbiór uczący pochodzi z pliku krokhin.txt znajdującym się w folderze rt_pred_data.
   Jako dane uczące posłużyło pierwsze 150 peptydów. Cały plik składa się 230 peptydów, więc pozostałe 80 peptydów posłużyło za zbiór testujący.

4. Procedura testowa
   a. współczynnik korelacji - określa w jakim stopniu zmienne są współzależne.
   b. wyznaczenie błędu średniokwadratowego (MSE)

5. Parametry sieci
    - Sieć opracowana w tym projekcie była oparta na składzie reszt aminokwasowych. Składa się z 20 węzłów wejściowych, 2 ukrytych węzłów i 1 węzła wyjściowego. W tym 21 wejściem jest stała liczba 1. Pierwsza warstwa posiada dwa neurony, druga warstwa przyjmuje trzy wejścia z tego dwa to rezultaty z neuronów z pierwszej warstwy i dodatkowy neuron, który jest wartością 1 do optymalizacji regresji.

    - Wykorzystane zostały dwie funkcje aktywacji. W pierwszej warstwie użyto funkcji sigmoidalnej, która definiowana jest jako krzywa w kształcie litery S, której środkowy fragment jest zbliżony do linii prostej, a fragmenty skrajne przyjmują kształt krzywej nasycenia. W drugiej warstwie użyta została funkcja liniowa, która polega na bezpośrednim przekazaniu wartości wyrażającej łączne pobudzenie neuronu na jego wyjście.

    - Pierwsze 20 wyników opisuje wagi danych białek dla pierwszej warstwy neuronowej. Dodatkowy jeden wynik to 21 neuron, który jest dodatkowym dla stałej jedynki niezależnej od białek.

    - Aminokwasy białkowe analizowane w projekcie:
        - Leucyna
        - Fenyloalanina
        - Izoleucyna
        - Tryptofan
        - Metionina
        - Walina
        - Tyrozyna
        - Alanina
        - Glutamina
        - Prolina
        - Cysteina
        - Asparagina
        - Treonina
        - Glicyna
        - Arginina
        - Asparagina
        - Glutamina
        - Seryna
        - Lizyna
        - Histydyna

      Wagi przed nauką wybrane losowo:

```
[ 4.29683237e-01 -4.10776307e-01]
[ 1.59375261e+00  1.06208188e-01]
```

```
[ 2.22803708e-01 -2.50521509e-01]
[-4.63610911e-02  1.71844031e-01]
[ 4.06008497e+00  5.31560448e-01]
[ 1.57109041e-03  1.37194856e-01]
[ 9.82309246e-02 -8.66373651e-01]
[-9.44583538e-01  2.87178284e+00]
[-2.64844214e-01 -3.85732656e-01]
[ 1.93631593e+00  8.18595871e-01]
[ 6.01489137e-01  9.36523151e-01]
[ 4.83992061e-01 -6.61084830e-01]
[ 8.96251378e-01 -2.61426889e-01]
[-3.53517691e-01 -1.21702799e-01]
[ 3.33666207e-01 -4.77876763e-01]
[-1.29588996e+00  1.09465213e+00]
[ 5.48261641e-01 -2.40583888e-01]
[ 1.31544827e+00  6.57755139e-01]
[ 2.74174481e+00  2.42680389e+00]
[ 1.73739402e+00  3.74053169e-01]
[-4.88569951e+00 -4.50237052e+00]
```

Dla drugiej warstwy mamy 3 wejścia więc trzy wagi.

```
[28.54171949 25.98849676 21.78104622]
```

Po nauce:

Pierwsza warstwa:

```
[-0.16595599  0.44064899]
[-0.99977125 -0.39533485]
[-0.70648822 -0.81532281]
[-0.62747958 -0.30887855]
[-0.20646505  0.07763347]
[-0.16161097  0.370439  ]
[-0.5910955   0.75623487]
[-0.94522481  0.34093502]
```

```
[-0.1653904   0.11737966]
[-0.71922612 -0.60379702]
[ 0.60148914  0.93652315]
[-0.37315164  0.38464523]
[ 0.7527783   0.78921333]
[-0.82991158 -0.92189043]
[-0.66033916  0.75628501]
[-0.80330633 -0.15778475]
[ 0.91577906  0.06633057]
[ 0.38375423 -0.36896874]
[ 0.37300186  0.66925134]
[-0.96342345  0.50028863]
[ 0.97772218  0.49633131]
```

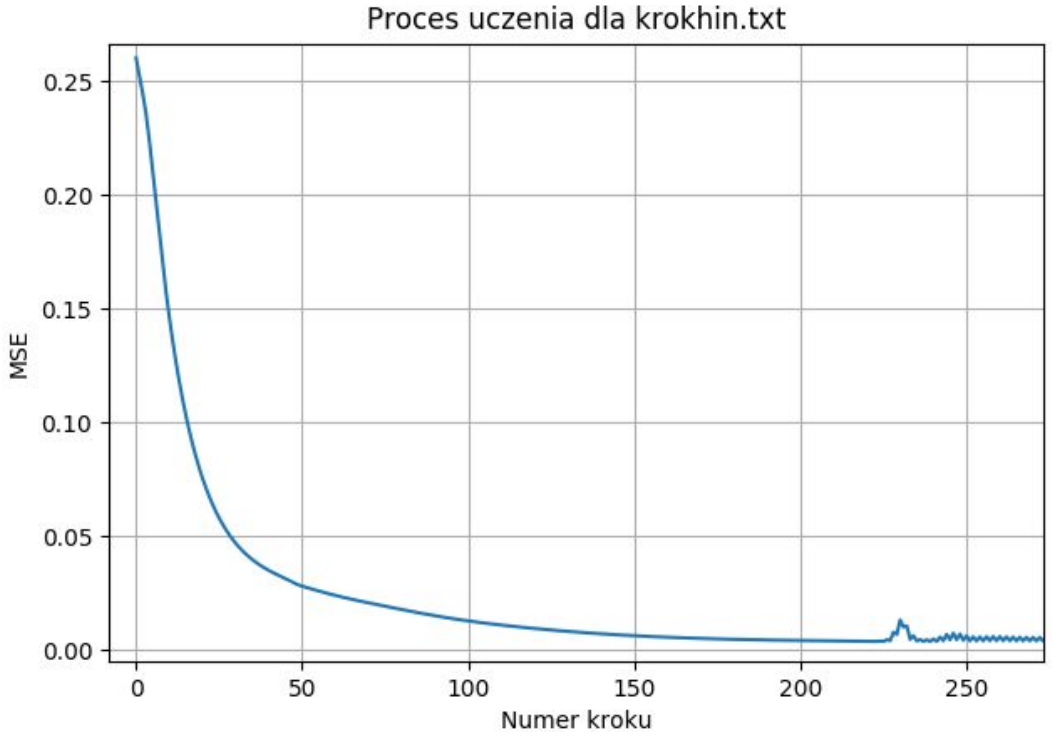
Druga warstwa:

```
[[-0.43911202]
 [ 0.57855866]
 [-0.79354799]]
```

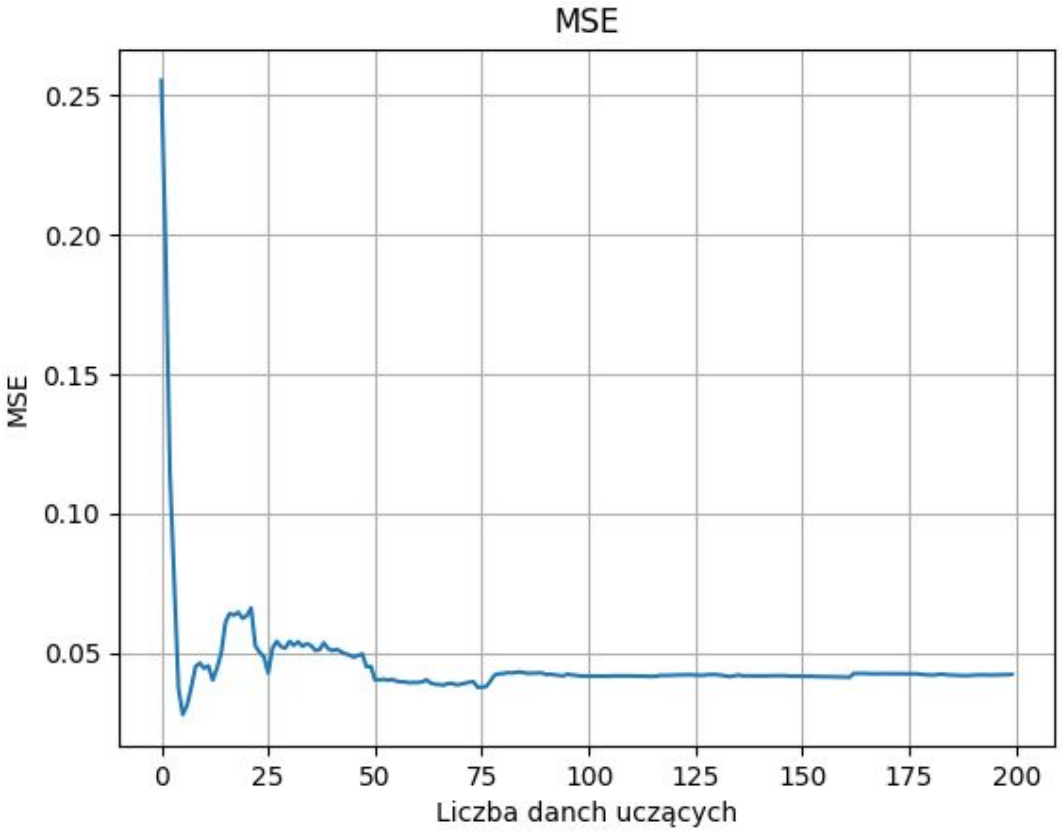
6. Wyniki pomiarów

   Proces uczenia zmiana błędu MSE od numeru iteracji (od epoki) i dla 150 danych wejściowych dla krokhin.txt

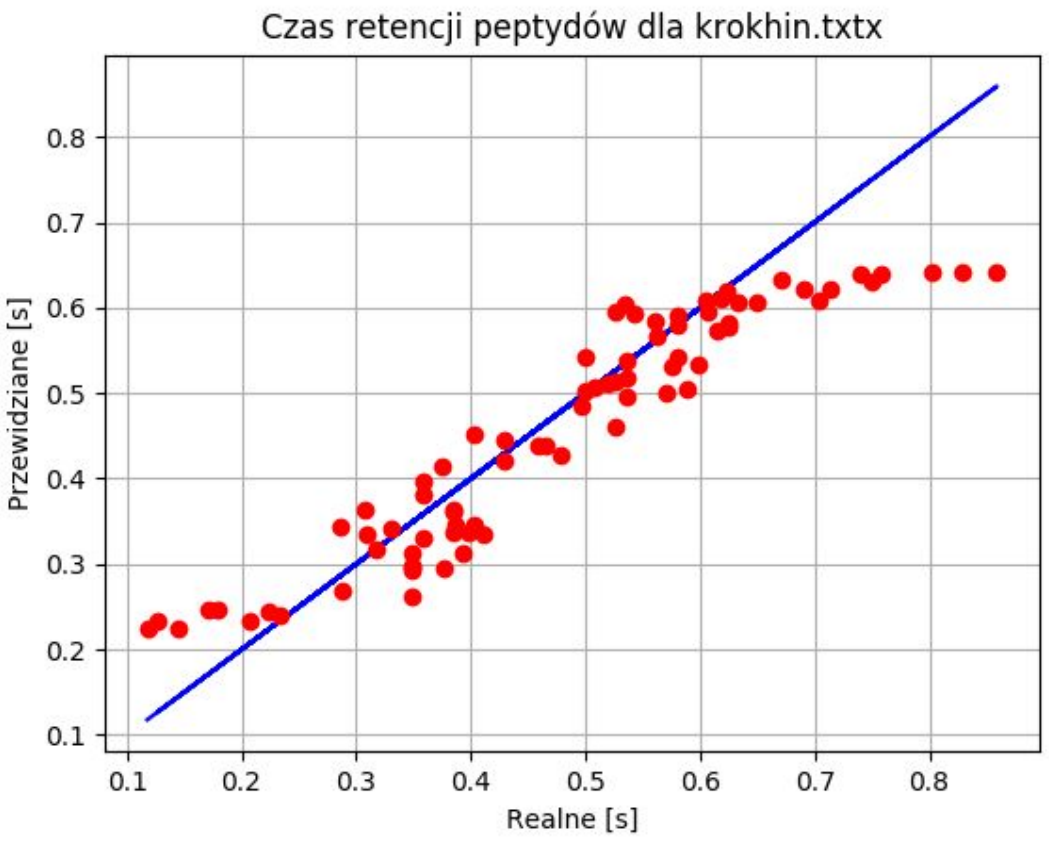
   Na Rysunkach 1,2,3,4 przedstawiono proces uczenia oraz uzyskane wyniki przewidywania czasów retencji dla peptydów z pliku krokhin.txt. Dane zostały podzielone na dwie części. Pierwszych 150 peptydów to dane uczące natomiast 80 peptydów do dane testujące.



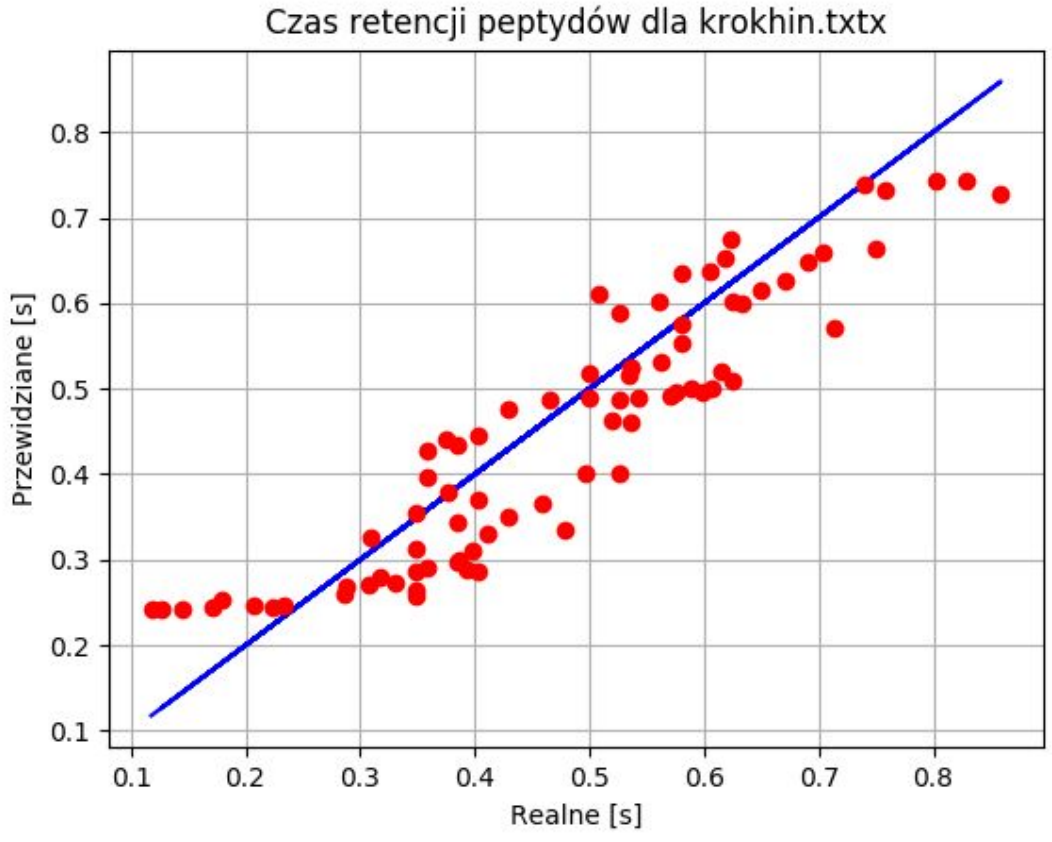
Rysunek 1. Wartość błędu średniokwadratowego od numeru epoki



Rysunek 2. MSE od ilości danych uczących



Rysunek 3. Czas przewidywane czasy retencji peptydów po 225 iteracjach. Współczynnik korelacji 0.89



Rysunek 4. Czas przewidywane czasy retencji peptydów po 1000 iteracjach. Współczynnik korelacji 0.86