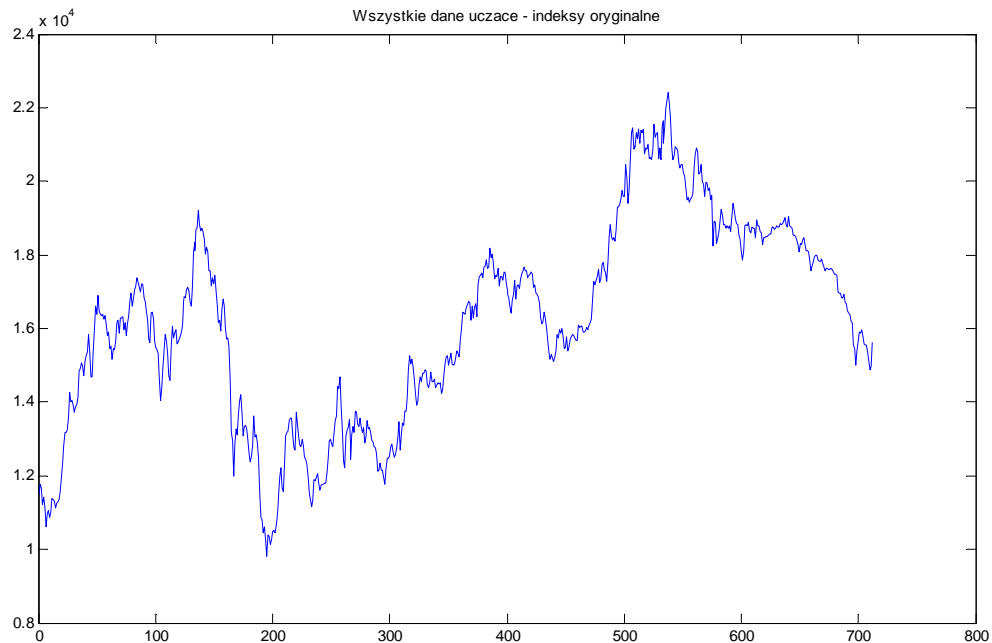


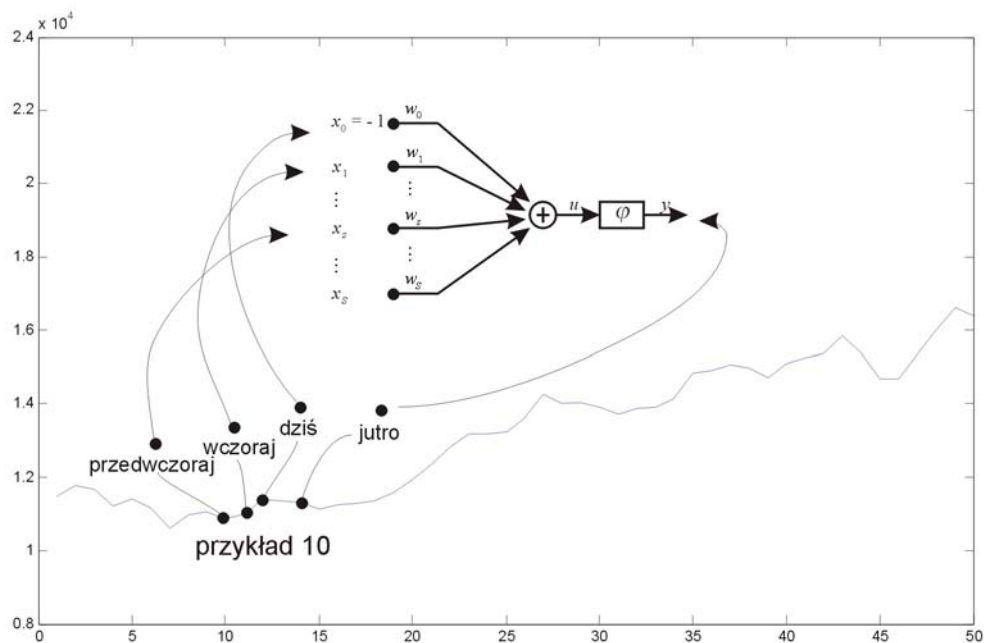
Sieci neuronowe - giełda

Zastosujemy dwuwarstwową sieć neuronową do predykcji ciągów (szeregów) czasowych. Przykładem takiego ciągu będą notowania giełdowe (WIG) z lat 1998÷2005.

Oto ich wykres:



Predykcja szeregów czasowych polega na tym, że sieć na podstawie danych historycznych uczy się przewidywać dane z przyszłości:



Ćwiczenie to jest bardzo podobne do ćwiczenia z siecią dwuwarstwową, która uczyła się przykładu XOR. Należy jednak więcej czasu i uwagi poświęcić na przygotowanie danych w ciągu uczącym.

Najpierw należy dane znormalizować – nie jest dobrym pomysłem podawać na wejścia sieci liczby rzędu 10000. Można albo znormalizować je tak, aby notowania mieściły się w zakresie od 0 do 1 (lub od -1 do 1 w zależności od użytej funkcji aktywacji), albo wykorzystywać nie ceny akcji (tu wartości WIGu), a ich różnice między dwoma kolejnymi notowaniami – przy okazji mamy ułatwienie w normalizacji: różnica ta zawiera się między -10% a +10% - oczywiście te procenty też dobrze jest znormalizować na przedział od 0 do 1 lub od -1 do 1. To wszystko po to, aby dane, jakie podajemy na sieć, i jakich ona ma się nauczyć, pokrywały się z zakresem wartości, jakie ona jest w stanie przyjąć i zwrócić.

Przykłady ułożone są w macierzy P w kolejnych kolumnach (wymiary: liczba wierszy – tyle ile wejść sieci, liczba kolumn – tyle ile przykładów), należy odpowiednio wykorzystać dane o notowaniach indeksu, aby utworzyć przykłady, z których każdy zawiera notowanie z danego dnia, poprzedniego i przedwczorajszego, a odpowiadający mu wektor wartości żądanych – notowanie z jutra.

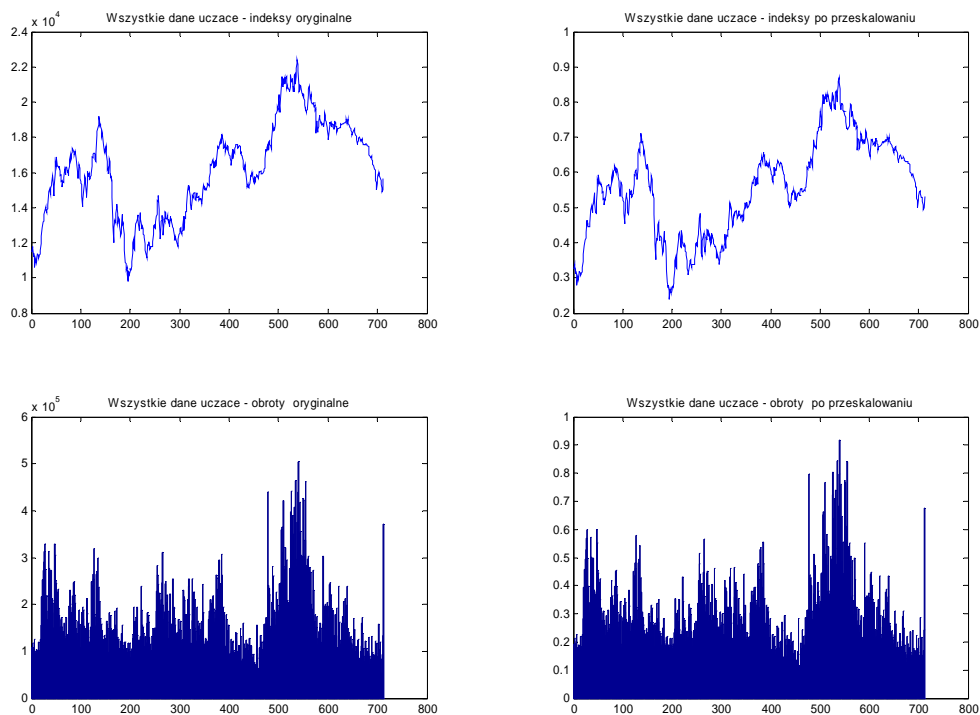
Dane uczące powinno się podzielić na ciąg uczący (zawierający przykłady wykorzystywane w procesie uczenia sieci) i ciąg testujący (np. ostatnie 25% przykładów) – na nim sprawdzać działanie sieci po nauczaniu.

Po przygotowaniu danych uczących i testujących należy sprawdzić, jak uczą się sieci o różnych architekturach. Podpowiedzi zawarte są w komentarzach w pliku zawierającym dane.

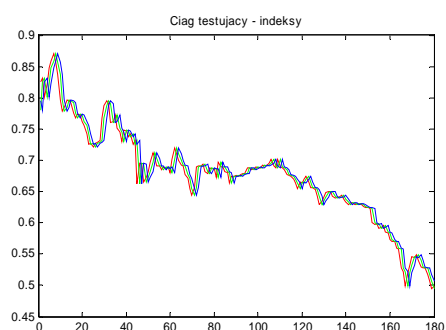
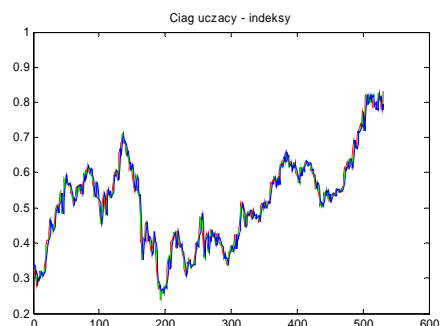
Oto, jak zachowuje się skrypt napisany przeze mnie.

Jest w nim dużo wykresów pokazujących ciąg uczący, parametry sieci, ich błędy itp.

Takie okienko z wykresami można wydrukować w celu pokazania (i sprawdzenia) danych przed i po normalizacji (przeskalowaniu, tu do zakresu 0÷1):



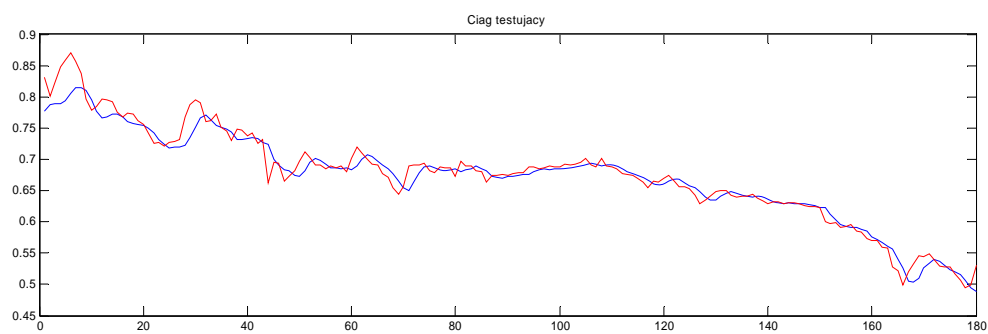
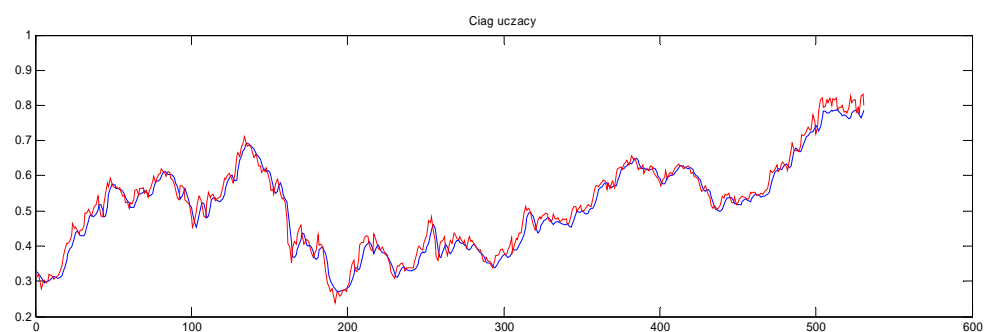
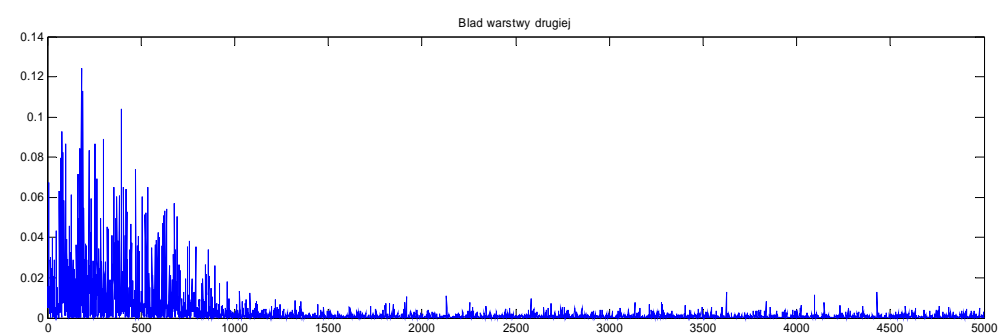
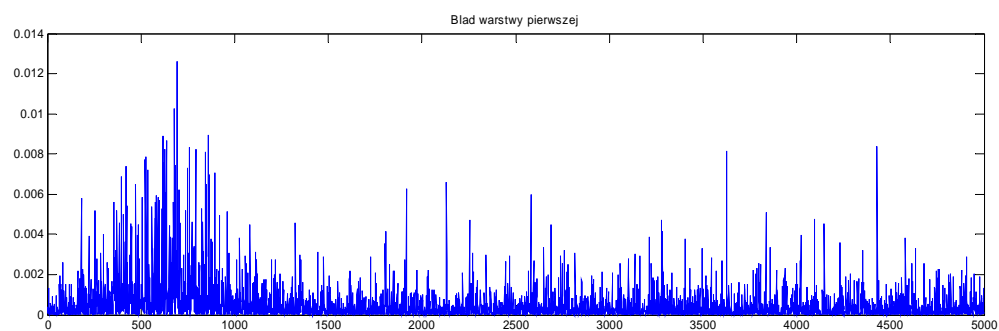
W ten sposób mój skrypt pozwala sprawdzić, czy dobrze są ułożone dane w ciągach uczącym i testującym (w kolejnych kolumnach powinny być one przesunięte względem siebie – można to sprawdzić naocznie rysując je wszystkie na jednym wykresie, ale w tzw. międzyczasie czekając na naciśnięcie czegoś):



Po prawej stronie można umieścić takie same wykresy dla obrotów – ja tego nie zrobiłem, bo nie wykorzystuję obrotów jako danych wejściowych.

A tak uczyły się moje sieci:

Sieć 5-1, 5000 epok, 1 pokaz w epoce:



Sieć 5-1, 5000 epok, 5 pokazów w epoce:

Sieć 5-1, 5000 epok, 10 pokazów w epoce:

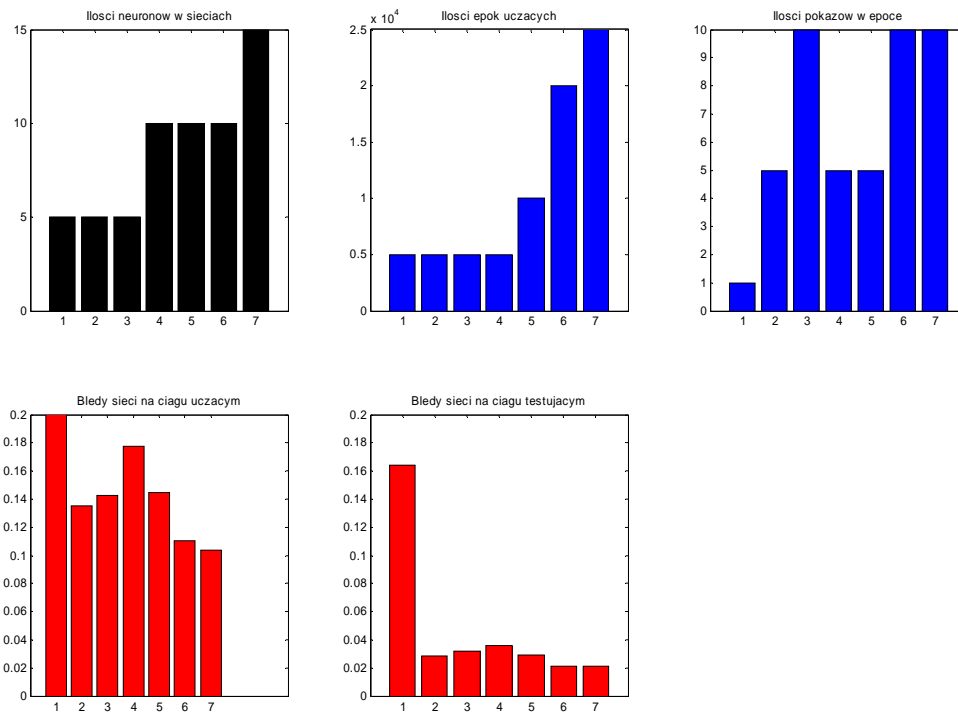
Sieć 10-1, 5000 epok, 5 pokazów w epoce:

Sieć 10-1, 10000 epok, 5 pokazów w epoce:

Sieć 10-1, 20000 epok, 10 pokazów w epoce:

Sieć 15-1, 25000 epok, 10 pokazów w epoce:

A oto jak mój skrypt pokazał porównanie wszystkich sieci poddanych uczeniu:



A tak odbywała się konwersacja:

```
Uczymy sieć nr 1
Podaj ilość neuronów w warstwie ukrytej: 5
Podaj ilość epok uczących: 5000
Podaj ilość pokazów w epoce: 1
Uczymy dalej?t
```

```
Uczymy sieć nr 2
Podaj ilość neuronów w warstwie ukrytej: 5
Podaj ilość epok uczących: 5000
Podaj ilość pokazów w epoce: 5
Uczymy dalej?t
```

```
Uczymy sieć nr 3
Podaj ilość neuronów w warstwie ukrytej: 5
Podaj ilość epok uczących: 5000
Podaj ilość pokazów w epoce: 10
Uczymy dalej?t
```

Uczymy sieć nr 4
Podaj ilość neuronów w warstwie ukrytej: 10
Podaj ilość epok uczących: 5000
Podaj ilość pokazów w epoce: 5
Uczymy dalej?t

Uczymy sieć nr 5
Podaj ilość neuronów w warstwie ukrytej: 10
Podaj ilość epok uczących: 10000
Podaj ilość pokazów w epoce: 5
Uczymy dalej?t

Uczymy sieć nr 6
Podaj ilość neuronów w warstwie ukrytej: 10
Podaj ilość epok uczących: 20000
Podaj ilość pokazów w epoce: 10
Uczymy dalej?t

Uczymy sieć nr 7
Podaj ilość neuronów w warstwie ukrytej: 15
Podaj ilość epok uczących: 25000
Podaj ilość pokazów w epoce: 10
Uczymy dalej?n

W trakcie uczenia tych sieci funkcja ucz2u (u oznacza unipolarną funkcję aktywacji: od 0 do 1) losowała w każdej epoce przykłady uczące. Można ją zmodyfikować tak, aby w zależności od dodatkowego parametru albo losowała, albo przeglądała cały ciąg uczący sekwencyjnie.

Sieć 15-1 (250000 epok, 1 pokaz w epoce) uczona sekwencyjnie dała takie wyniki:

Po 500000 epok takie:

Po 1000000 (słownie: jednym milionie) epok takie:

Sieć 15-1 (25000 epok, 10 pokazów, odpowiednik pierwszej sieci) takie:

Sieć 15-1 (50000 epok, 10 pokazów, odpowiednik drugiej sieci) takie:

Sieć 15-1 (100000 epok, 10 pokazów, odpowiednik trzeciej sieci) takie:

Wyniki dla tych sieci:

Ostatnia próba – dla większej sieci: 25-1 (500000 epok, 20 pokazów w epoce):

Najlepiej samemu naocznie wypróbować, jak uczą się sieci. Można poprobować np. przewidywać notowania na 5 dni naprzód, na 10 dni naprzód, czy co Wam tylko inwencja podpowie.

Podsumowanie - co można zrobić samemu:

- Zbadać różne architektury sieci (różna liczba neuronów w warstwie ukrytej)
- Inny zakres czasowy notowań na wejściu, np. notowania sprzed tygodnia, sprzed dwóch i sprzed miesiąca, albo średnią z ostatniego miesiąca
- Dodać jeszcze na wejście obroty z dzisiaj, albo z dzisiaj i wczoraj itd.
- Użyć wstępnej obróbki danych wejściowych – np. podawać na wejścia nie wartość indeksu (przeskalowaną od 0 do 1 lub od -1 do +1), ale zmianę od wartości poprzedniej – ona zwykle mieści się w granicach od -15% do +15% i sieć może wtedy dokładniej nauczyć się tych zmian (je oczywiście też skalujemy na zakres od 0 do 1 lub od -1 do +1)

- Sprawdzić, jak uczy się sieć, dla której w każdej epoce losujemy przykłady, a jak taka, której przykłady pokazujemy sekwencyjnie
- Sprawdzić, jak uczy się sieć, której w jednej epoce pokazujemy kilka przykładów uczących (dla każdego osobno obliczamy poprawki, pamiętamy je, a na koniec epoki uśredniamy i modyfikujemy wagi)
- Użyć innej funkcji aktywacji (bipolarnej) – trzeba wtedy uważnie przerobić obie funkcje – działaj i ucz, oraz przerobić dane uczące na zakres od -1 do +1
- Zbadać zachowanie sieci dla różnych współczynników nachylenia funkcji aktywacji beta
- Przerobić wagi początkowe na zakres od +0.4 do +0.6, dane wejściowe na zakres od 0 do 1, a funkcję aktywacji tak, aby zakres jej wejścia był od 0 do 1 – wtedy wszystko będzie od 0 do 1
- To co podałem jako zadania dodatkowe dla sieci jedno- i dwuwarstwowych - adaptacyjny współczynnik uczenia, sprawdzenie po nauczaniu sieci, czy osiągnęła minimum funkcji błędu itp.