POLITECHNIKA WARSZAWSKA

WYDZIAŁ MATEMATYKI  
 I NAUK INFORMACYJNYCH

Przewidywanie szeregów czasowych i Indeksów giełdowych

Wyniki i analiza testów

Projekt nr 2 na potrzeby przedmiotu „Sieci Neuronowe”

Maciej Kachniarz

Tomasz Luśtyk

# Cel projektu

Celem projektu jest zbadanie skuteczności przewidywania wartości kursu indeksów giełdowych przez sieci neuronowe oraz utworzenie aplikacji umożliwiającej przeprowadzanie tego typu testów.

# Wymagania funkcjonalne

Aplikacja ma spełniać poniższe warunki.

## Parametryzacja testu

Aplikacja ma mieć możliwość ustawienia następujących parametrów testu.

Parametry budowy sieci:

* Rodzaj sieci (do wyboru: MLP, Elmana, Jordana)
* Ilość neuronów w poszczególnych warstwach
* Rodzaj funkcji pobudzenia (do wyboru: unipolarna, bipolarna)

Parametry algorytmu uczącego:

* Współczynnik uczenia
* Współczynnik bezwładności
* Współczynnik podziału na zbiór testowy i treningowy

Parametry przeprowadzanego testu:

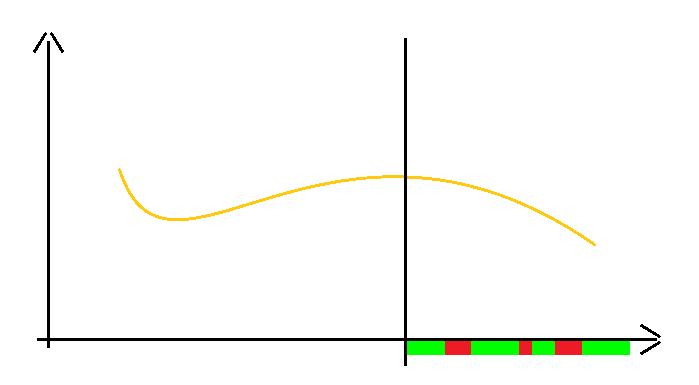
* Ilość iteracji
* Metoda doboru przykładów testowych (do wyboru: online, stochastic)

Parametry danych testowych:

* Rodzaj przewidywanego indeksu (do wyboru: WIG20, S&P 500, FTSE 100)
* Dane wejściowe analizy technicznej (do wyboru: cena otwarcia, zamknięcia, maximum, minimum, zmiana kursu, wielkość obrotu przewidywanego indeksu). Możliwość wyboru wielu opcji.
* Dane wejściowe analizy fundamentalnej (do wyboru: produkcja przemysłowa, obroty bieżące, zadłużenie, deficyt, PKB, bezrobocie, inflacja, kursy walut – złoty, euro, dolar, frank szwajcarski, funt brytyjski, WIBOR 1M/6M/12M, surowce: złoto/srebro/miedź/ropa brent). Możliwość wyboru wielu opcji.
* Okno czasowe danych wejściowych (do wyboru: z ostatniego dnia, tygodnia, 2 tygodni, miesiąca, 3 miesięcy, roku)
* Częstość próbkowania danych wejściowych (do wyboru: z każdego dnia, co 3 dni, co tydzień, co 2 tygodnie, co miesiąc)

## Analiza testu

Aplikacja ma wizualizować wyniki testu na wykresie. Wykres przedstawiać będzie cenę przewidywanego indeksu z zaznaczoną poprawnością przewidywań sieci. Miejsca pod wykresem odpowiadające danym, dla których sieć neuronowa poprawnie obliczyła zmianę kursu (przewidziała wzrost lub spadek) zaznaczone będą kolorem zielonym. Miejsca, dla których otrzymano błędną predykcję (sieć przewidywała wzrost, podczas gdy w rzeczywistości zaistniał spadek lub odwrotnie) zaznaczone będą kolorem czerwonym. Przykładowy wykres przedstawiono na rysunku 1.



Rysunek

Aplikacja prezentować będzie takie współczynniki:

* Procent poprawnych przewidzeń
* Błąd względny zmiany kursu dla poprawnych przewidzeń
* Błąd względny zmiany kursu dla błędnych przewidzeń
* Błąd względny zmiany kursu dla wszystkich przewidzeń

# Schemat działania

Schemat przeprowadzenia testu jest następujący

## Inicjalizacja

Po zdefiniowaniu przez użytkowania wszystkich parametrów testu tworzone są przykłady testowe i treningowe sparametryzowane czasem danych wejściowych oraz częstością próbkowania. W przypadku, gdy dana niedostępna będzie z żądaną częstością próbkowania obliczana będzie jej średnia wartość w danym oknie czasowym danych wejściowych i wejdzie ona w skład danych testowych, jako pojedyncza wartość. W przypadku wyboru sieci Elmana lub Jordana okno czasowe zawiera tylko jeden, ostatni dzień. Ponadto budowana jest sieć neuronowa.

## Nauka

Sieć „przechodzi” po zbiorze przykładów treningowych na wejście dostając dane z okna wejściowego a jako przewidywany wynik faktyczną wartość indeksu z kolejnego dnia. Po każdym obliczeniu wagi sieci są poprawiane zgodnie z algorytmem uczącym. Ilość przejść oraz kolejność zależne są od ilości iteracji oraz metody doboru przykładów testowych.

## Test

Sieć „przechodzi” po zbiorze przykładów testowych. Następuje obliczanie współczynników wynikowych, bazujących na różnicy wyniku obliczonego przez sieć a wyniku rzeczywistego.

# Architektura sieci neuronowej

## Sieć MLP

Sieć Multi-Layer-Perceptron to sieć zbudowana z warstw, w których znajdują się neurony. Każdy z neuronów oblicza sumę swoich wejść, a następnie wartość ta (jako poziom pobudzenia) przekazywana jest do funkcji aktywacji, jako argument. Wartość funkcji aktywacji stanowi o wyjściu neuronu. Każdy z neuronów warstwy *i* połączony jest ze wszystkimi neuronami warstwy *i+1* a każde z tych połączeń ma ustaloną wagę.

W przypadku predykcji giełdy, liczba neuronów wejściowych i wyjściowych jest zdeterminowana przez ten problem. Sieć będzie miała jeden neuron w warstwie wyjściowej, który będzie zwracał przewidywaną wartość wybranego indeksu giełdowego w momencie zamknięcia na następny dzień. Liczba neuronów wejściowych będzie zaś zależna od wybranego przypadku:

1. Liczba neuronów warstwy wejściowej wynosi *C*, gdzie *C* jest sumą wybranych danych wejściowych analizy technicznej i danych wejściowych analizy fundamentalnej.
2. Liczba neuronów warstwy wejściowej wynosi *P\*C*, gdzie *C* jest zdefiniowane jak w powyższym podpunkcie, a *P* oznacza liczbę dni historycznych rozpatrywanych przez nas.

Dni historyczne nie muszą być kolejnymi dniami. Możliwe jest branie pod uwagę np. dnia ostatniego, tydzień temu i 2 tygodnie temu. W przypadku, gdy dana niedostępna będzie z żądaną częstością próbkowania liczba neuronów warstwy wejściowej wynosi *P\*(C-N)+N* gdzie *N* oznacza ilość danych o niedostępnej żadnej częstości.

Wydaje się, iż jako że ten typ sieci nie dysponuje pamięcią (nie posiada połączeń rekurencyjnych), lepsze byłoby zastosowanie punktu nr 2. Dodatkowo podawanie danych historycznych zrekompensuje brak rekurencyjnych połączeń w sieci. Niemniej jednak, brak doświadczenia i chęć głębszego poznania i porównania, jakości działania obu sposobów podejścia do reprezentacji wejścia sieci powoduje, że punkt nr 1. również zostanie przez nas zaimplementowany.

Ilość neuronów w warstwach pośrednich oraz ilość warstw pośrednich będzie możliwa do ustalenia przez użytkownika programu z poziomu GUI.

### Sposób uczenia sieci MLP

Do uczenia sieci wykorzystamy zaimplementowaną już momentową metodę wstecznej propagacji błędu. Sprowadza się ona do uzależnienia zmian wartości wag w bieżącym kroku od gradientu funkcji celu oraz zmiany tych samych wag w kroku poprzednim (bezwładności). Dzięki uwzględnieniu w uczeniu sieci bezwładności, znacznie przyśpieszony zostaje proces uczenia na płaskich obszarach funkcji błędu oraz utrzymywanie względnie stałego kierunku zmian w przestrzeni wag.

## Sieć Elmana

Sieć Simple Recurrent Networks (SRN) jest modyfikacją sieci perceptronowej MLP. Bazuje ona na strukturze sieci Jordana. Jej cechą charakterystyczną jest częściowa rekurencja w postaci sprzężenia zwrotnego między warstwą ukrytą, a warstwą wejściową. Sieć bazuje na sprzężeniu zwrotnym.

Każdy neuron w warstwie ukrytej będzie miał swój odpowiednik w warstwie kontekstowej, stanowiącej wspólnie z wejściami sieci warstwę wejściową sieci. Wartym podkreślenia jest fakt, iż przekazane sygnały do warstwy kontekstowej nie mają wag.

Warstwa wyjściowa

Warstwa wejściowa

Warstwa kontekstowa

Warstwa ukryta

Liczba neuronów w sieci wyjściowej wynosi 1 (predykcja wartości zamknięcia giełdy), w warstwie ukrytej będziemy mieć *K* neuronów, natomiast w wejściowej: C *+ K*, gdzie *C* to liczba cech (czynników) dla pojedynczego dnia, natomiast *K* reprezentować będzie pamięć naszej sieci.

Wartość *K* możliwa do ustalenia przez użytkownika programu z poziomu GUI.

### Sposób uczenia sieci Elmana

W uczeniu sieci Elmana spróbujemy wykorzystać gradientową metodę uczenia. Będziemy obliczać gradient funkcji celu względem każdej wagi sieci. Będziemy inicjalizować wagi neuronów w warstwie ukrytej i wyjściowej za pomocą rozkładu równomiernego z przedziału [-1;1] bądź [0;1] (w zależności od wybranej funkcji aktywacji).

## Sieć Jordana

Sieć Simple Recurrent Networks (SRN) jest bardzo podobna do sieci Elmana. Różni się jedynie faktem, że warstwa kontekstowa na wejściu dostaje dane z warstwy wyjściowej a nie ukrytej.

### Sposób uczenia sieci Jordana

Sposób uczenia sieci Jordana jest analogiczny do uczenia sieci Elmana.

# Przewidywane trudności

Najprawdopodobniej podczas trwania projektu konieczne będzie zmierzenie się z takimi zadaniami jak:

* Implementacja GUI
* Implementacja logiki biznesowej – struktury sieci neuronowych oraz metod nauki
* Pozyskanie i przetworzenie do odpowiedniego formatu danych testowych
* Przeprowadzenie testów oraz analiza zaobserwowanych zależności