**POLITECHNIKA WARSZAWSKA**

**WYDZIAŁ ELEKTRYCZNY**

**INSTYTUT ELEKTROTECHNIKI TEORETYCZNEJ**

**I SYSTEMÓW INFORMACYJNO-POMIAROWYCH**

**PRACA DYPLOMOWA INŻYNIERSKA**

**na kierunku Elektrotechnika**

**specjalność: Automatyka i Inżynieria Komputerowa**

Arkadiusz Łyjak

Nr albumu 170330

Rok akad.: *2014/2015*

Warszawa, 23.11.2013

**Wykorzystanie sieci neuronowych do predykcji kursów walut**

Opiekun naukowy:  Kierownik Zakładu Dydaktycznego:

Dr inż. Marcin Kołodziejprof. dr hab. inż. Remigiusz Rak

Konsultant:

Termin złożenia pracy: data złożenia pracy do recenzji

Praca wykonana i zaliczona pozostaje

własnością Instytutu i nie będzie

zwrócona wykonawcy.

**Streszczenie**

Celem niniejszej pracy jest opracowanie sposobu wykorzystania sieci neuronowych w predykcji kursów giełdowych. W tym celu zastosowano model perceptronu wielowarstwowego. Zbadano predykcyjne możliwości wybranego modelu sieci.

Część teoretyczna pracy przybliża czytelnikowi:

* Omówienie zagadnień związanych z funkcjonowaniem rynków finansowych.
* Opis najważniejszych czynników mogących mieć wpływ na zachowanie się rynku.
* Kryteria wyboru danych wejściowych oraz sposobu ich przygotowania.  
    
  Część praktyczna zawiera:
* Opis i zastosowanie sieci neuronowej opartej na modelu perceptronu wielowarstwowego.
* Przedstawienie sposobu trenowania modelu oraz analizę wyników działania sieci w oparciu o dane z rynku walutowego na przykładzie pary EUR/USD.

W pracy wykorzystano oprogramowanie Matlab (Neural Network Toolbox)

**Title: Using neural networks to predict the exchange rate of the currency pairs**

**Abstract**

The purpose of this paper is to present the possibilities of using neural networks in predicting stock exchange. A MLP model of the neural network was applied. It was examined the predictive capabilities of the neural network model. Efforts were made to describe a number of important factors that may affect the outcome of this type of research.

The theoretical part includes:

* A discussion of issues related to the functioning of financial markets.
* Identify the most important factors that may have influence to the behavior of the market.
* The presentation of the input data selection criteria and the method of preparation.

The practical part includes:

* Description and application of neural network model based on multilayer perceptron.
* Demonstrate how to train a model and an analysis of the performance of the network based on the currency market data on the example of the EUR / USD.

The study was based on Matlab software (Neural Network Toolbox)

OŚWIADCZENIE

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Podziękowania

Spis treści

[1 Wstęp 10](#_Toc416251746)

[1.1 Cel i założenia pracy 12](#_Toc416251747)

[2 Opis wybranych strategii inwestycyjnych 14](#_Toc416251748)

[2.1 Wykres świecowy 14](#_Toc416251749)

[2.2 Pojęcie trendu 17](#_Toc416251750)

[3 Przegląd technicznych możliwości użycia sieci neuronowej i teorii cyfrowego przetwarzania sygnałów w predykcji kursu giełdowego 19](#_Toc416251751)

[3.1 Porównanie wykresu ceny giełdowej do sygnału fal mózgowych 19](#_Toc416251752)

[3.2 Falowa oraz harmoniczna teoria rynków 19](#_Toc416251753)

[3.3 Rozpoznawanie wzorców rynkowych 21](#_Toc416251754)

[4 Sieci neuronowe – budowa i zastosowanie 25](#_Toc416251755)

[4.1 Podstawy biologiczne neuronu 25](#_Toc416251756)

[4.2 Model matematyczny pierwszego neuronu 26](#_Toc416251757)

[4.3 Cechy sieci neuronowych do zastosowań predykcyjnych 27](#_Toc416251758)

[5 Budowanie modelu sieci neuronowej 29](#_Toc416251759)

[5.1 Schemat blokowy badań 29](#_Toc416251760)

[5.2 Przygotowanie danych wejściowych 30](#_Toc416251761)

[5.3 Podział danych na uczące i testujące 36](#_Toc416251762)

[5.4 Badanie właściwości predykcyjnych wybranych typów sieci 38](#_Toc416251763)

[5.5 Wpływ ilości danych uczących oraz ilości zmiennych na własności prognostyczne sieci neuronowej 39](#_Toc416251764)

[5.6 Badanie wpływu liczby zmiennych na maksymalny błąd prognozy 42](#_Toc416251765)

[5.7 Badanie wpływu liczby neuronów ukrytych na maksymalny błąd prognozy 43](#_Toc416251766)

[5.8 Szczegółowe przedstawienie wyników trenowania sieci o optymalnej topologii oraz parametrach wejściowych 44](#_Toc416251767)

[6 Podsumowanie 50](#_Toc416251768)

[7 Zakończenie 52](#_Toc416251769)

[Bibliografia 53](#_Toc416251770)

[Dodatek A. Kod programu komputerowego 55](#_Toc416251771)

[Dodatek B. Spis zawartości dołączonej płyty CD 64](#_Toc416251772)

# Wstęp

Istnieje wiele pojęć i definicji ~~[2, 20, 21]~~ omawiających to, czym są lub mogą być rynki finansowe [2, 20, 21]. W zależności od tego, jaką wiedzą dysponuje inwestor lub z jakiego środowiska się wywodzi, definicje te mogą się różnić. Zatem inwestorów możemy podzielić na kilka grup.

Pierwsza grupa to inwestorzy mający wykształcenie ekonomiczne i doszukujący się wpływu na rynek wydarzeń ze świata ekonomii i polityki. Drugi typ inwestorów to osoby niemające wiedzy ekonomicznej, za to dysponujące bogatą wiedzą z dziedziny analizy technicznej. I trzeci, ostatni typ inwestorów, to są osoby, które nie mają właściwie żadnej wiedzy nt. funkcjonowania rynków i korzystają z wiedzy np. biur maklerskich. Trudno jest ocenić, która grupa ma tak naprawdę rację, ponieważ w myśl pewnego bardzo popularnego powiedzenia - „każdy ma swój pomysł na rynek, a rynek ma dla każdego odpowiedz”, rację ma ten, kto zarabia i nikt więcej. ~~Wobec powyższego nie należy nikogo oceniać przedwcześnie.~~

Pozostając przy dwu pierwszych grupach inwestorów tj. stosujących analizę fundamentalną i/lub analizę techniczną, można nieco szerzej rozwinąć zagadnienie podejścia do rynków finansowych. Faktem jest, że tak naprawdę nikt na świecie nie wie, co się będzie działo na rynku i może tylko przypuszczać/zakładać/domniemywać (ujmując dość ogólnie, ponieważ nie można się skupić na tzw. „inside tradingu”, czyli sytuacji, gdzie inwestor posiada informację z „wewnątrz”, co jest niezgodne z prawem) i zawsze jest to wynik będący czymś z pogranicza rachunku prawdopodobieństwa i wypadkową przekonań na podstawie posiadanej wiedzy, a więc przypuszczeniem i prywatną opinią.

Innymi słowy, jeśli większość uważa, że cena wzrośnie i dokonają oni zakupów (w sensie otworzą pozycje długie – BUY) cena wzrośnie i odwrotnie, jeśli większość uzna, że cena jest za wysoka i zaczną sprzedawać (otwierając pozycje krótkie – SELL) cena spadnie. Jeśli obie te siły będą w równowadze rynek pozostanie w konsolidacji.

Mówi się że, **rynek jest wypadkową przekonań jego uczestników** [21]. Ważny przy tym, z punktu widzenia zwykłego inwestora, mającego do swojej dyspozycji cały szereg metod i strategii inwestycyjnych, jest **sposób przetwarzania informacji** giełdowych oraz jak najlepsze ich wykorzystanie. Jednakże problemem dla pojedynczej jednostki jest mnogość ~~owych~~ strategii oraz niezliczona ilość informacji, które należy „przetworzyć”, aby uzyskać w końcowym efekcie informację, będącą podstawą do zawarcia transakcji na rynku. Nasuwa się zatem pytanie – w jaki sposób, jak najszybciej oraz najefektywniej można te informacje przetworzyć?

Pierwszą czynnością powinno być określenie, które wg inwestora informacje są **ważne** a które można zwyczajnie pominąć. Raczej nie znajdziemy dwóch, nawet stosujących tę samą strategię inwestycyjną, traderów (inaczej graczy giełdowych), zgadzających się w stu procentach, co do aktualnej sytuacji rynkowej. Jest to kolejny dowód na to, że rynek jest tworzony przez przekonania jego uczestników.

Zazwyczaj nie jest możliwe obserwowanie więcej niż kilku instrumentów giełdowych jednocześnie oraz analizowanie zachowania się ceny, stosując kilkanaście wskaźników i kilku strategii inwestycyjnych. To ograniczenie wiąże się z możliwościami analitycznymi ludzkiego mózgu i nie jest to czymś niezwyczajnym, ponieważ, pomimo faktu, że ludzki mózg jest bardzo potężnym, (chociażby pod względem ilości komórek nerwowych w nim zawartych w porównaniu do budowy sztucznych sieci neuronowych), narzędziem, to już **trudność sprawia** rysowanie lewą ręką koła a prawą kwadratu. Mowa tutaj o podzielności uwagi oraz niemożliwości wykonywania wielu różnych czynności jednocześnie. Co oczywiście nie jest niemożliwe, ale po prostu człowiek nie jest do tego przystosowany i w dłuższej perspektywie może być bardzo stresujące.

Odpowiedzią na powyższe trudności może być tzw. automat inwestycyjny, czyli program komputerowy, który „wyręczy” niejako inwestora ze żmudnego analizowania danych i angażowania się emocjonalnego w grę giełdową. Pozostaje więc określenie programowej płaszczyzny rozwiązania problemu oraz wyboru elementów strategii będących przedmiotem rozważania.

Kolejną trudnością jest ocena skuteczności rozwiązań automatycznych. Na przestrzeni lat można było zaobserwować, że zwykłe algorytmy programowe, ze względu na swoją „sztywność” kodu, nie dają oczekiwanych rezultatów. Dzieje się tak, ponieważ wykonują one ściśle zaprogramowane czynności, bez możliwości analizy zmieniającej się dynamicznie sytuacji rynkowej. Skuteczność takowych rozwiązań (średnio) można ocenić na nie więcej, niż 55-65% (czyli mniej więcej co druga pozycja była stratna) [1], co stanowi wynik satysfakcjonujący pod warunkiem, że:

1. Otwarte i zrealizowane zyskowne pozycje były większe pod względem ekspozycji kapitału na transakcję niż te stratne,
2. Pozycje stratne były automatycznie zamykane po osiągnięciu założonego z góry poziomu ceny, przy której uznano, że transakcja powinna zostać zamknięta z minimalną stratą – Stop Loss,

Co ważne - większość dotychczasowych strategii automatycznych nie radzi sobie z gwałtownie zmieniającą się tendencją na rynku (krach, nagłe załamanie bieżącej tendencji itp.), co nierzadko kończy się całkowitym wyzerowaniem konta inwestora bez możliwości odrobienia poniesionych strat.

Nasuwa się zatem niejako samoistnie stwierdzenie, że automat powinien działać jak robot, ale mieć także zdolność analizy zachowania rynku jak człowiek. Rozwiązaniem mogą być właśnie algorytmy z użyciem sieci neuronowych, będące tematem niniejszej pracy.

## Cel i założenia pracy

Celem ~~niniejszej~~ pracy jest zbadanie możliwości wykorzystania sieci neuronowej w predykcji kursu walutowego na przykładzie pary walutowej EUR/USD. Praca składa się z części teoretycznej, opisującej zagadnienie od strony analizy technicznej rynków finansowych, oraz opisu prostej strategii inwestycyjnej na podstawie, której został zbudowany ~~odpowiedni~~ model sieci. W części praktycznej poruszono problem doboru ~~odpowiednich~~ danych wejściowych, doboru topologii sieci oraz jej typu, i co za tym idzie, sposobu trenowania. W trakcie wykonywania pracy wspomagano się dostępnymi pracami naukowymi traktującymi o problemie predykcji rynków.

Zakres pracy obejmuje:

1. Wprowadzenie teoretyczne do analizy technicznej rynków finansowych,
2. Opis najważniejszych elementów prostej strategii inwestycyjnej na podstawie, której został stworzony model sieci neuronowej,
3. Przegląd możliwych rozwiązań programowych stosowanych w handlu automatycznym na podstawie różnych teorii rynkowych takich jak np. teoria fal czy harmoniczności,
4. Omówienie podstaw działania i budowy sztucznych sieci neuronowych,
5. Wybór odpowiedniej struktury sieci na potrzeby predykcji kursów walut,
6. Badania mające na celu wskazanie optymalnych parametrów sieci takich jak:

* topologii,
* ilości neuronów ukrytych,
* wielkości zbioru uczącego,

1. Wyznaczenie błędów stworzonej sieci neuronowej.

Pracy zawiera wnioski z przeprowadzonych badań oraz ~~jednoznacznie~~ wskazuje, które cechy podawane na wejście sieci mają decydujący wpływ na prognozowanie kursu walut, a które nie mają praktycznego wpływu lub ich wpływ może zostać pominięty.

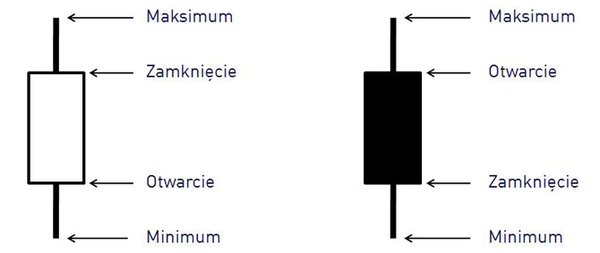
Zakończenie pracy jest próbą przedstawienia możliwych do zastosowania rozwiązań z użyciem sieci neuronowych, mających na celu jak najdokładniejsze prognozowanie kursów walut.

# Opis wybranych strategii inwestycyjnych

## Wykres świecowy

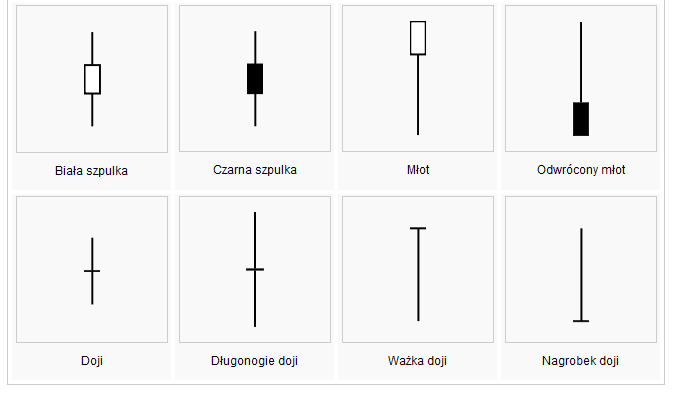
~~Kolejnym bardzo~~ ważnym elementem strategii inwestycyjnej są wykresy świecowe, które ~~również~~ są przedstawione na rys. 1.1 [17]. Z punktu widzenia analizy komputerowej zasadniczą część stanowią cztery parametry każdej ze świec tworzących wykres.

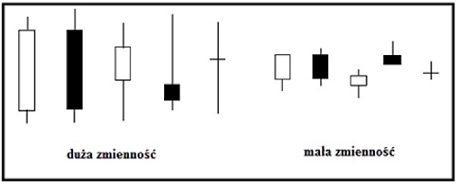
Są to: cena otwarcia, zamknięcia, maksimum – górny cień świecy, minimum – dolny cień świecy.

Rys. 2.2. Wygląd oraz najważniejsze cechy świecy japońskiej tj. ceny zamknięcia, otwarcia oraz maksimum i minimum z danej sesji.

Rys. 1.2 przedstawia dwa główne rodzaje świec japońskich. Świece „białe” - oznaczające wzrost ceny w danej sesji oraz świece „czarne” - oznaczające sesje spadkowe. Wg terminologii japońskich technik inwestycyjnych im większy korpus, czyli różnica między ceną otwarcia i zamknięcia, w obu typach świec, tym większa zmienność na rynku i jego dynamika. Innymi słowy - duże wahania cen [5, 17].

Poniżej na rys. 1.3 i 1.4 pokazane są różne rodzaje pojedynczych linii świec japońskich, które również występują na rynku. Wielkość/wysokość korpusu wyznacza jw. wspomniano dynamikę zmiany ceny. Istnieją sesje, gdzie cena zmienia się w dużym zakresie zmienności, ale są również takie sesje, gdzie cena zmienia się minimalnie – małe korpusy, tzw. szpulki, młoty, wisielce itp. lub wcale – świece typu doji.

Rys. 2.3. Wygląd oraz najważniejsze cechy świecy japońskiej tj. ceny zamknięcia, otwarcia oraz maksimum i minimum z danej sesji.

Rys. 2.4. Porównanie wielkości (wysokości) korpusów oraz ich cieni z uwzględnieniem powodowanej przez nie zmienności i dynamiki rynkowej.

Informacje niesione przez rozpiętość cenową na danej sesji, wysokość korpusu, są danymi wejściowymi dla modelu sieci neuronowej [13]. Niniejsza praca obejmuje badania rynku walutowego FOREX, a dokładnie parę walutową EUR/USD, charakteryzującą się największą płynnością oraz dynamiką zmiany ceny. Oznacza to, i to jest główne założenie przy projektowaniu modelu neuronowego, że cena zamknięcia poprzedniej sesji jest równocześnie ceną otwarcia sesji następnej. Wynika stąd, że sieć neuronowa do swojego poprawnego działania oraz prognozowania może potrzebować tylko cen zamknięcia w postaci liczbowej. Jest to w gruncie rzeczy najważniejsze założenie przy projektowaniu danej sieci.

Rys. 1.5 przestawia ~~rzeczywisty~~ wykres notowań pary walutowej EUR/USD, gdzie zaznaczono liniami poziomymi ~~zostały~~ miejsca (wybrane, aby nie zaciemniać obrazu) pokazujące, że ceny zamknięcia poprzedniej sesji (w rzeczywistości cena otwarcia następnej). ~~Oczywiście~~ jest to wyidealizowany przykład, różnice między cenami otwarcia i zamknięcia sąsiednich sesji mogą różnić się o kilka PIPS (punktów), co stanowi niewielki procent zasięgu całej sesji, czyli dla założeń projektu są pomijalnie małe.

Ze względu na ogromną płynność tej pary walutowej różnice, o których wspomniano powyżej, występują zazwyczaj w niedziele o godz. 23 czasu polskiego, kiedy zaczyna się kolejny tydzień handlowy.

Rys. 2.5. Wygląd oraz najważniejsze cechy świecy japońskiej tj. ceny zamknięcia, otwarcia oraz maksimum i minimum z danej sesji.

## Pojęcie trendu

Strategia inwestycyjna składa się z wielu elementów, zarówno tych typowo technicznych jak i psychologicznych [2, 3]. Niestety nie wszystkie zagadnienia techniczne udaje się „przełożyć na język maszynowy”, tym bardziej jest to trudne, jeśli chodzi o zagadnienia z pogranicza psychologii tradingu czy analizy behawioralnej samego inwestora. Oznacza to, że tworząc narzędzie programowe, w głównej mierze można polegać tylko na tym, co daje się opisać w sposób matematyczny.

Do dyspozycji inwestora jest około 1000 różnych wskaźników analizy technicznej, setki formacji cenowych. Nie jest w praktyce możliwe (ani tym bardziej konieczne) wykorzystanie wszystkich tych możliwości, dlatego należy skupić się na wyborze tylko tych, które mają lub mogą mieć największy wpływ na poprawną analizę zachowania się ceny danego waloru/instrumentu giełdowego.



Rys. 2.1 Poglądowy przykład prostej analizy technicznej z zaznaczeniem pojęcia trendu rynkowego oraz metodą jego określania za pomocą średnich kroczących.

Na rys. 1.1 pokazano, w pewnym uproszeniu, sposób analizy trendu rynkowego. Jest to pojęcie względne, ponieważ wielu inwestorów, przez lata praktyk rynkowych, wypracowuje swoje własne metody. Jednakże dla celów analizy komputerowej należy skupić się na takich metodach, które da się jednoznacznie opisać matematycznie.

Elementy wykresu, jako składowe strategii, można sklasyfikować i opisać następująco:

* trend spadkowy, wskazuje na średnioterminowy trend spadkowy,
* SMA7 i SMA11, średnie kroczące „opisujące” bardzo dynamiczny ruch na giełdzie. Wg teorii cena opadająca lub wzrastająca pod kątem 45° charakteryzuje uregulowany i stabilny ruch (Teoria Gann’a),
* SMA23, średnia, która wraz z dwoma poprzednimi stanowi element pomagający ocenić sytuację rynkową pod względem otwierania pozycji (zawierania transakcji) oraz ich późniejszego zamykania,
* SMA200, dwustuokresowa średnia krocząca (opisana dokładniej w rozdz. 5.2)

Przykładowo, w trendzie spadkowym jak na rys. 2.1, świece otwierające się i zamykające pod tymi średnimi w przypadku, gdy inwestor posiada otwarte pozycje krótkie (SELL), mówią, że pozycje nie są zagrożone. Jednak tylko do momentu, kiedy świece zaczynają od dołu przekraczać swoimi korpusami linie średnich i tak do momentu ich skrzyżowania (przecięcia się). Tym samym po skrzyżowaniu się linii a następnie ponownym zamykaniu się świec poniżej powstaje kolejny sygnał do otwarcia tzw. pozycji krótkich - SELL. To samo dotyczy trendu wzrostowego, lecz w odwrotnej kolejności.

Średnie kroczące to wiodący element analizy technicznej niemalże na całym świecie. Właśnie z uwagi na fakt ich ogromnej popularności mają one bardzo duże znaczenie w prognozowaniu zmian ceny instrumentów finansowych i dynamiki rynkowej [X]. Paradoksalnie rzecz biorąc, są one popularne, ponieważ są skuteczne oraz są skuteczne, ponieważ... są popularne.

~~Jest to o tyle ciekawe zagadnienie, nie tylko odnoszące się do średnich kroczących, ale do każdego elementu strategii~~. Dzieje się tak, ponieważ rynek tworzą jego uczestnicy, podejmujący decyzje inwestycyjne na podstawie tego, co obserwują na wykresie, a że na średnie kroczące patrzą niemal wszyscy, począwszy od zawodowych *traderów* pracujących dla banków czy instytucji finansowych, a kończąc na inwestorach indywidualnych. ~~Z technicznego punktu widzenia jest to, innymi słowy, pewien rodzaj „uśredniania” oraz „odszumiania” ceny rynkowej.~~

Podobne znaczenie mają także inne wskaźniki analizy technicznej [20] np. RSI (Relative Strength Index – wskaźnik siły względnej), ROC (Rate of Change) czy oscylator stochastyczny.

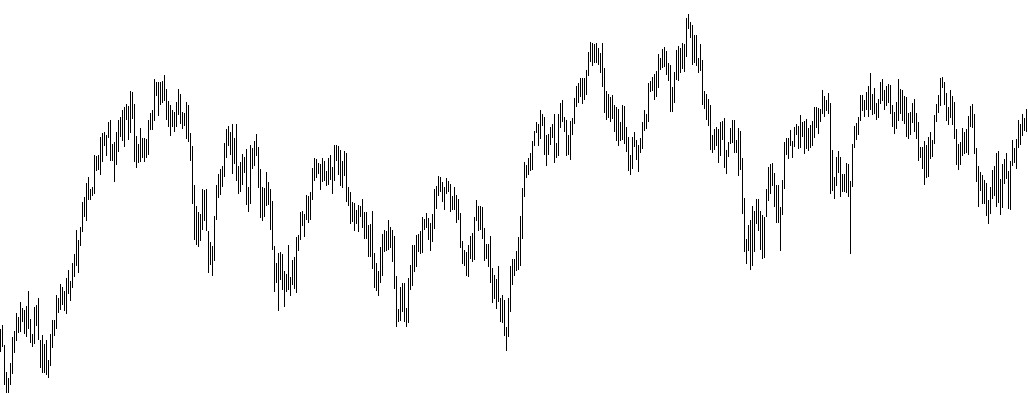
Wymienione wskaźniki analizy technicznej w pewnej mierze zawdzięczają swoja popularność temu, że niemal każdy początkujący inwestor uczy się inwestycji polegając właśnie na ich wskazaniach. Niestety z biegiem czasu utrwala się to jako nawyk, który opierając się na zasadzie wspomnianej wyżej, ma lub może mieć znaczenie w kształtowaniu się ceny giełdowej.

# **Przegląd technicznych możliwości użycia sieci neuronowej i teorii cyfrowego przetwarzania sygnałów w predykcji kursu giełdowego**

## Porównanie wykresu ceny giełdowej do sygnału fal mózgowych

Wykres ceny instrumentu giełdowego można w zasadzie potraktować jak sygnał elektryczny. Jest to śmiałe stwierdzenie, ale biorąc pod uwagę fakt, że cena na giełdzie jest tak naprawdę wypadkową decyzji (pewnego rodzaju konsensusem), wszystkich obecnych i przeszłych uczestników rynku, zaczynają się otwierać nowe perspektywy i możliwości analizy z użyciem algorytmów i technik, jakie stosuje się w analizie sygnałów elektrycznych.

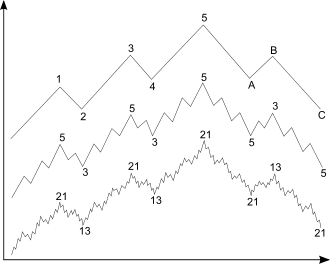
Jednakże należy już na wstępie zaznaczyć, że sygnał będący ceną rynkową jest bardzo skomplikowanym i trudnym materiałem do „obróbki”, ponieważ jak wspomniano wyżej, jest wypadkową decyzji inwestycyjnych wszystkich inwestorów i jako taki, z pewnym przybliżeniem, może być traktowany jako złożony sygnał ~~EEG~~.



Rys. 3.2. Skonsolidowany wykres ceny przedstawiający duże niezdecydowanie na rynku oraz chaotyczność notowań.

## Falowa oraz harmoniczna teoria rynków

~~Ponadto~~ wg teorii fal Elliott’a [19, 22] ceny zmieniają się falowo oraz cyklicznie [16] wg schematu pokazanego na rys. 3.3. Analiza FFT pozwala w tym przypadku na wyznaczenie cykli krótko-, średnio-, i długoterminowych, gdzie, ~~pomiędzy którymi~~ zachodzi pewna zależność harmoniczna polegająca na tym, że jeden cykl jest wielokrotnością długości innego [10]. Ma to oczywiście swoje wyjaśnienie ekonomiczne – gospodarka bierze, mówiąc potocznie, „wdechy i wydechy”, w efekcie tworząc okresy hossy i bessy.



Rys. 3.3. Schemat 5-falowego ruchu ceny giełdowej. Trzy impulsy wzrostowe oraz następujące po nich dwa impulsy spadkowe, tzw. korekty. Podstawowy falowy schemat zachowania się rynków finansowych.

W przypadku ceny giełdowej, tak samo jak w przypadku sygnału elektrycznego, będącego obiektem badań i analizy cyfrowej, ~~także~~ możemy mówić o zniekształceniu sygnału poprzez nieokreślony szum. Dokładnie widać to na trzech poniższych wykresach.

|  |
| --- |
| Ashampoo_Snap_2015  Ashampoo_Snap_2015  Ashampoo_Snap_2015 |

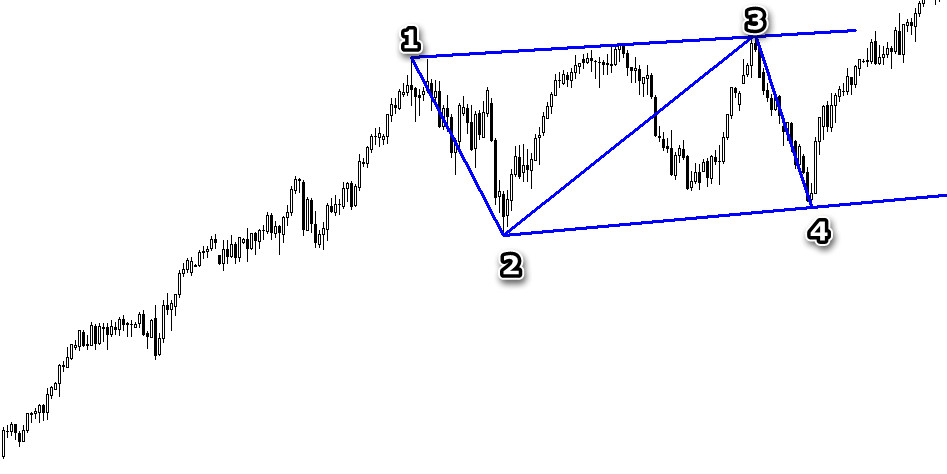
Rys. 3.4. Odszumianie poprzez zmianę podstawy czasu. Wykres pierwszy od góry – podstawa czasu 1 godz., wykres środkowy – 4 godz. i wykres najniżej 24godz. dla tego samego przedziału czasowego notowań indeksu DAX30. Im większa podstawa czasu, tym wykres bardziej zostaje „wygładzony”.

Zastosowanie uśredniania powoduje „wygładzenie” notowań. Dzieje się tak, dlatego że przy zmianie na większy interwal czasowy np. z 1 godz. na dobowy na wykresie przestają być widoczne szumy sesyjne. ~~Można powiedzieć, że szumy te to tak jakby „wyższe harmoniczne” powstałe na skutek daytradingu, czyli gry giełdowej, gdzie gracz może otwierać od kilku do nawet kilkuset transakcji (High Frequency Trading) w ciągu jednej tylko sesji i to w obu kierunkach. Powoduje to znaczne wahania ceny w ciągu dnia sesyjnego i szum dezinformacyjny.~~

## Rozpoznawanie wzorców rynkowych

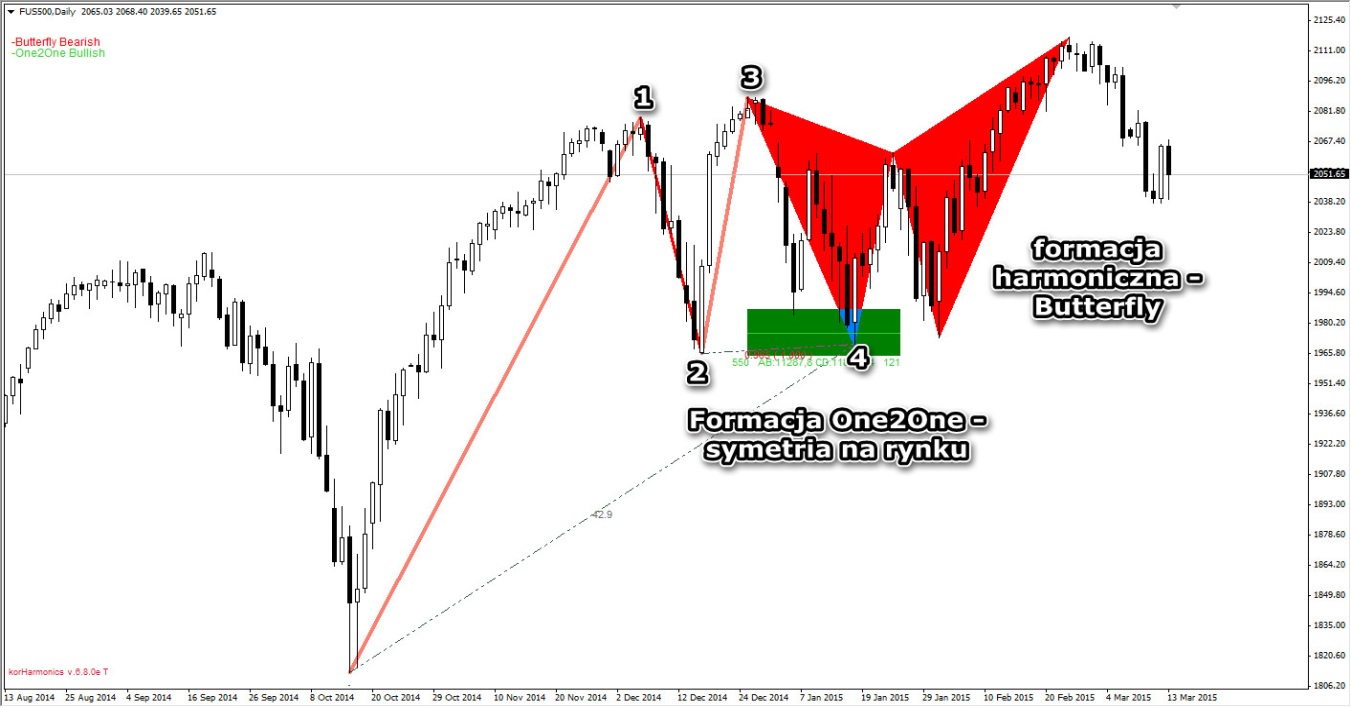
Istnieją sieci neuronowe zdolne rozpoznawać i klasyfikować wzorce. Jeżeli sieć potrafi rozpoznawać ludzką twarz, głos a nawet gestykulacje, z pewnością poradzi sobie również z wyszukiwaniem i klasyfikowaniem klasycznych formacji cenowych na wykresach giełdowych [7]. Niemniej jednak zadanie to nie jest takie proste jakby się mogło wydawać, ponieważ aby sieć poprawnie mogła taki wzorzec odnaleźć, musi się go najpierw nauczyć.

Problemem staje się odpowiedni opis matematyczny danego układu rynkowego, gdyż każdy wierzchołek formacji (np. 1,2,3,4 na wykresie poniżej) jest ściśle określony przez poziomy ceny występujące w danym czasie na danym ~~poziomie oraz będących względem siebie w pewnej zależności matematycznej.~~



Rys. 3.5. Wzorzec cenowy będący złożeniem formacji harmonicznej 121 (symetria, kolejność fale: 1-2 i 3-4), korekty pędzącej wg teorii Elliotta (1-2-3-4) oraz klasycznej formacji prostokąta (górna krawędź poprowadzona przez pkt 1-3 i dolna poprowadzona przez pkt 2-4).

Oznacza to, że bez wcześniejszego przygotowania danych wejściowych nie jest możliwe nauczenie sieci odpowiedniego zachowania, czyli rozpoznawania wzorców geometrycznych. Wprawdzie w tym jak i przypadku przedstawionym na rys. 3.6 problem sprowadza się do odpowiedniego algorytmicznego opisu danych formacji. Następnie tak przygotowane dane można potraktować jako dane wejściowe sieci neuronowej.



Rys. 3.6. Wykres przedstawiający złożenie dwóch wzorców harmonicznych tj. formacji 121 (symetria 1-2 i 3-4, równość tych fal) oraz popularnej formacji Butterfly

Najczęściej spotykanym sposobem wykorzystania sieci neuronowych w predykcji kursów giełdowych są sieci korzystające z prostego, choć starannie wyselekcjonowanego zestawu danych wejściowych - cech [4, 6] . Najczęściej, stosuje się dane liczbowe dotyczące danych sesyjnych, takich jak:

* Cena:
* Otwarcia
* Zamknięcia
* Maksimum sesji
* Minimum sesji
* Dane wolumenowe
* Liczba otwartych pozycji
* Wskaźniki i oscylatory np.
* RSI
* ATR
* czy CCI
* Średnie kroczące (wykładnicze, ważone, proste itp)
* Korelacje międzyrynkowe

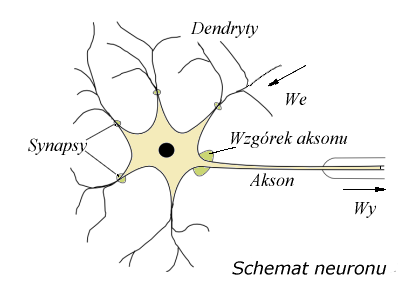
Przy czym w zależności od potrzeb i oczekiwać stosuje się odpowiednie kombinacje tych danych, np. niektóre dane można opóźniać bądź przyspieszać względem innych, lub uśredniać np. średnia cena zamknięcia za 5 ostatnich sesji. Mnogość tych działań jest zależna tylko i wyłącznie od twórcy sieci.

# Sieci neuronowe – budowa i zastosowanie

## Podstawy biologiczne neuronu

Sztuczne sieci neuronowe to programy lub urządzenia fizyczne naśladujące pracę centralnego układu nerwowego u organizmów żywych [14]. Przy czym najważniejszą kwestią, przy ich projektowaniu, było zaobserwowanie sposobu, w jaki żywe komórki radzą sobie z przetwarzaniem informacji a następnie przeniesienie to na platformę programową lub sprzętową.

Aby lepiej zrozumieć zagadnienie sztucznych sieci neuronowych warto posłużyć się prostymi modelami, zarówno pojedynczej komórki nerwowej – rys. 2.1 jak i jej matematycznego (oczywiście bardzo uproszczonego) odpowiednika – rys. 2.2.



Rys. 4.1. Wygląd oraz najważniejsze cechy świecy japońskiej tj. ceny zamknięcia, otwarcia oraz maksimum i minimum z danej sesji.

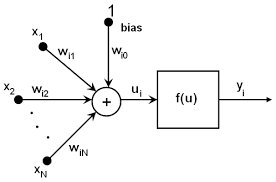
Komórka nerwowa składa się z ciała komórki (inaczej somy), dendrytów oraz synaps. W ciele komórki znajduje się jądro, w którym realizowane są w większości funkcje przetwarzania informacji. Sygnały wejściowe doprowadzane są do komórki nerwowej za pomocą synaps i dendrytów (przy czym ilość dendrytów przypadających na pojedynczą komórkę waha się w granicach 103 – 104), natomiast sygnał wyjściowy odprowadzany jest przez aksony oraz ich odgałęzienia zwane kolateralami. Sam akson rozgałęzia się na wiele mniejszych włókien tworząc tzw. drzewko wyjściowe neuronu [15].

Sygnał wyjściowy, przetworzony przez komórkę, i wychodzący przez zakończenia aksonu, jest dalej propagowany, bez osłabienia, jako ciąg impulsów. Impulsy te są przekazywane do innych komórek, gdzie również są przetwarzane.

Biorąc pod uwagę fakt, że w ludzkim mózgu znajduje się ok. 1011 neuronów pełniących proste funkcje sumowania sygnałów wejściowych, a następnie porównujących uzyskaną wielkość z progiem, oraz że liczba połączeń miedzy neuronami to ok. 1000 na każdy neuron, możemy stwierdzić, że mózg jest olbrzymią maszyną obliczeniową o wręcz niewyobrażalnych możliwościach. Długi cykl pojedynczej komórki rekompensowany jest **przetwarzaniem równoległym**, ~~czyli kolejną bardzo ważną z pkt. widzenia zastosowania w technice, cechą sieci neuronowych.~~

## Model matematyczny pierwszego neuronu

Pierwszy matematyczny model neuronu, zbudowany na podstawie biologicznego neuronu został zaproponowany w roku 1943. Zasadę jego działania z pewnym niewielkim uproszczeniem przedstawia rys. 2.2. [15, 18].



Rys. 4.2. Pierwszy matematyczny model komórki nerwowej zaproponowany w roku 1943 przez McCullocha i Pitts’a.

Opis matematyczny pierwszego modelu neuronu przypomina zasadę działania prostego procesora, który działa wg poniższego wzoru:

|  |  | 1. (4.1) |
| --- | --- | --- |

gdzie:jest argumentem funkcji. Funkcja aktywacji w modelu McCulloch i Pitts’a jest funkcją skokową.

Ten prosty model neuronu oblicza sumę sygnałów wejściowych, przemnożonych wcześniej przez odpowiednie wagi, i porównuje z pewną wartością progową zadaną poprzez funkcję aktywacji. Współczynniki wij występujące we wzorze (2.1) symbolizują wagę połączeń synaptycznych w rzeczywistym modelu neuronu.

## Cechy sieci neuronowych do zastosowań predykcyjnych

Wybór sieci neuronowych do rozwiązania zadania predykcji kursu giełdowego zostało podyktowane tym, że posiadają one szereg szczególnych cech, które są szczególnie istotne w tego typu zadaniach. Tymi cechami z pewnością są:

* zdolność do generalizacji wiedzy i uczenia się
* oraz równoległego przetwarzania informacji

Generalizacja polega na tym, że sieć wytrenowana na pewnej grupie (zbiorze) danych uczących potrafi tę „wiedzę” wykorzystać na danych niebiorących udziału w procesie uczenia.

Ponadto predykcja kursu giełdowego wymaga przetworzenia ogromnego zbioru danych wejściowych w jak najkrótszym czasie i jak najmniejszym błędem. Biorąc pod uwagę fakt, że tymi danymi wejściowymi mogą być, i zazwyczaj są:

* dane historyczne sięgające wstecz nawet kilkadziesiąt lat (na Forex do 30 lat)
* dane pochodzące ze wskaźników analizy technicznej np. RSI, SMA, CCI itp.
* lub/i przedstawione w odpowiedniej formie dane makroekonomiczne,

to niezwykle istotnym z tego punktu widzenia atutem sieci jest **przetwarzanie równolegle**.

Powoduje to znaczące przyspieszenie procesu trenowania sieci oraz uzyskania pożądanych wyników jej pracy.

Samo zagadnienie predykcji sygnału polega na tym, aby sieć na podstawie danych historycznych potrafiła jak najdokładniej przewidzieć wartości przyszłe. W procesie uczenia (treningu sieci), wykorzystuje się wektor danych wejściowych xi oraz wektor wartości zadanych di. Innymi słowy, mając dane wartości próbek poprzednich *x(k-1), x(k-2), x(k-3), …, x(k-N)* sieć ~~musi podjąć decyzję, jaka będzie~~ wartość estymowana próbki na wyjściu w chwili aktualnej.

Wagi w procesie predykcji są aktualizowane biorąc pod uwagę bieżący błąd predykcji wyrażony wzorem:

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

oraz poprzednich wartości błędu predykcji.

# Budowanie modelu sieci neuronowej

## Schemat blokowy badań

NIE

TAK

zakończenie badań

zwiększenie/zmniejszenie l. neuronów

optymalna l. neuronów

Czy zwiększenie/zmniejszenie l. neuronów ukrytych polepsza analizę?

trening

badanie wpływu liczby neuronów ukrytych

mały błąd prognozy

zwiększenie ilości danych

duży błąd prognozy

trening

badanie wybranych topologii sieci

badanie wpływu ilości danych uczących

cascadeforwardnet

feedforwardnet

elmannet

import danych wejściowych

## Przygotowanie danych wejściowych

W celu zapewnienia prawidłowego przebiegu nauczania sieci neuronowej oraz prezentacji wyników, konieczne jest odpowiednie przygotowanie macierzy danych wejściowych [9, 11, 14]. Dane giełdowe użyte do nauki sieci mogą pochodzić bezpośrednio z platformy brokerskiej, bądź ze specjalizowanych serwisów giełdowych. W zależności od źródła mogą różnić się pod względem ilości danych historycznych oraz/i liczba zmiennych.

Przykładowe dane historyczne dla pary walutowej EUR/USD pobrane z serwisu STOOQ.PL prezentują się następująco:

Tabela 5.1. Dane zaimportowane z serwisu Stooq.pl zawierające niezbędne informacje odnośnie wszystkich czterech parametrów ceny, czyli open, high, low i close



Prezentowany powyżej schemat danych składa się z następujących pozycji:

* Date - data w formacie „yyyy-mm-dd”, czyli rok, miesiąc i dzień
* Open - cena otwarcia sesji
* High - maksymalna cena w sesji
* Low - minimalna cena w sesji
* Close - cena zamknięcia sesji

W przypadku importu danych bezpośrednio z platformy brokerskiej, w zależności od brokera, dane te mogą być dodatkowo uzupełnione o ~~dodatkowe~~ pola charakteryzujące dany instrument giełdowy.

Ze względu na to, że do tego typu badań im większa liczba przykładów uczących tym lepiej, dla celów pracy skorzystano z danych historycznych pary walutowej EUR/USD zaczynających się w 4-01-1971r. do 5-03-2015r. tj. od czasu pierwszego notowania na giełdzie walutowej FOREX do bieżącego miesiąca (w chwili pisania tego tekstu jest marzec 2015r.).

Tabela 5.2. Dane zaimportowane z platformy brokerskiej Metatrader 4. Zawiera dodatkowe dwie kolumny typu Volume i OpenInt, które oznaczają kolejno: wolumen transakcji oraz liczbę otwartych pozycji.



Format danych jak powyżej zawiera dodatkowe kolumny:

* Volume
* OpenInt

Powyższe dwie zmienne Volume i OpenInt nie będą stosowane w analizie działania prezentowanej w niniejszej pracy sieci neuronowej, stąd na wstępie przygotowania danych należy je wyrugować ze zmiennych. Działanie to jest podyktowane faktem, że obie te wartości nie są w rzeczywistości wartościami prezentującymi właściwe dane. Dzieje się tak, dlatego że rynek walutowy Forex jest rynkiem **zdecentralizowanym** i nie ma jednego źródła wszystkich danych, lecz są to tzw. dane lokalne, czyli np. pochodzące od jednego dostawcy płynności (danych) i nieuwzględniających całości rynku[[1]](#footnote-1).

Import danych a następnie ich formatowanie polega w głównej mierze na wyrugowaniu danych niebędących w zamyśle użytecznymi w procesie predykcji oraz posegregowanie wg wytycznych Matlaba.

Za prawidłowy proces importu danych oraz ich wstępnego podziału odpowiada następująca formuła kodu programu napisana zgodnie ze składnią języka skryptowego pakietu Matlab:

imported\_data = importdata('EURUSD\_STOOQ.csv');

[w1, k1] = size(imported\_data.data);

[w2,k2] = size(imported\_data.textdata);

Data\_Num = imported\_data.data(1:w1,1:4);

Data\_Text = imported\_data.textdata(2:w2, 1);

Ze względu na to, że pliki z danymi giełdowymi zawierają struktury tekstowe jak i liczbowe, należy je ~~również~~ odseparować od siebie, gdyż zmienne tekstowe nie stanowią danych, które można poddać analizie.

Zastosowany w badaniach predykcyjnych model sieci neuronowej w warstwie wejściowej ma macierz o wymiarach 13x11280 w skład, której wchodzą następujące wiersze danych: *Date2Num, WeekDayNum, DayOfMonth, MonthNum, OpenPrice, HighPrice, LowPrice, ClosePrice, DiffCloseOpen, SMA7, SMA11, SMA23, SMA200*. Liczba 13 oznacza liczbę zmiennych, a liczba 11280 oznacza liczbę przykładów uczących (zależy od ilości danych historycznych). Zastosowanie średnich kroczących, jako zmiennych, zostało dokładniej wyjaśnione w rodziale 2.2.

Tabela 5.3. Tabela opisująca znaczenie poszczególnych zmiennych zastosowanych w zadaniu predykcji

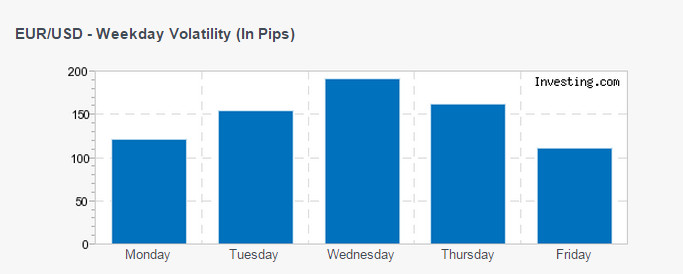
|  |  |
| --- | --- |
| *Date2Num* | data zamieniona na liczbę np. 19.05.2000 w formacie 'dd.mm.yyyy’ to 730625 |
| *WeekDayNum* | dzień tygodnia np. poniedziałek, wtorek |
| *DayOfMonth* | nr dnia miesiąca np. 1,2,3 … 29, 30, 31 |
| *MonthNum* | nazwa miesiąca np. styczeń, luty |
| *OpenPrice* | cena otwarcia danej sesji |
| *HighPrice* | cena maksymalna na danej sesji |
| *LowPrice* | cena minimalna na danej sesji |
| *ClosePrice* | cena zamknięcia danej sesji |
| *DiffCloseOpen* | różnica miedzy cena zamknięcia i otwarcia |
| *SMA7* | zwykła średnia krocząca (Simple Moving Average) o okresie 7 z cen zamknięcia |
| *SMA11* | SMA o okresie 11 z cen zamknięcia |
| *SMA23* | SMA o okresie 23 z cen zamknięcia |
| *SMA200* | SMA o okresie 200 z cen zamknięcia |

Kod programu odpowiadający za złożenie wielu segmentów danych w odpowiadającą strukturze sieci macierz danych wejściowych Inputs.

Inputs = [Date2Num WeekDayNum DayOfMonth MonthNum OpenPrice HighPriceLowPrice ClosePrice DiffCloseOpen SMA7 SMA11 SMA23 SMA200];

Wybór właśnie tych, a nie innych zmiennych został podyktowany następującymi przesłankami. Notowania rynku Forex odbywają się od niedzieli od godz. 23 do piątku do godz. 23. Zmienność i dynamika ruchu ceny zależy od dnia tygodnia oraz miesiąca. Dzieje się tak dlatego, że zazwyczaj po dniach wolnych od notowań, czyli sobocie i niedzieli, inwestorzy wstrzymują się od podejmowania decyzji inwestycyjnych (poniedziałek) i czekają, aż inni uczestnicy rynku „odsłonią” swoje zamiary (kulminacja w środę), co może oznaczać małą dynamikę i zmienność do tego czasu, co widać na rys. 5.1.

Pomaga to uniknąć nieprzewidywalnych ruchów ceny oraz ominąć ewentualne luki cenowe, niebezpieczne dla otwartych pozycji. Podobna sytuacja ma miejsce w piątki, ponieważ większość inwestorów woli nie zostawiać otwartych pozycji przez weekend w obawie o wspomniane wyżej luki cenowe.



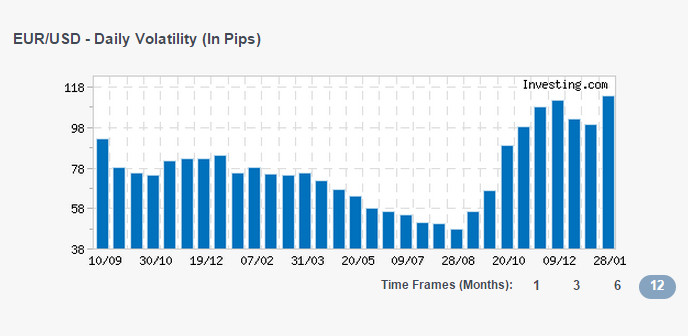
Rys. 5.1. Dzienna zmienność w PIPS pary walutowej EURUSD w odniesieniu do poszczególnych handlowych dni tygodnia. Wykresy pochodzą ze strony http://www.investing.com/forex-tools/volatility-calculator

Stąd zasadne było użycie zmiennych wejściowych, które wskazywały jednoznacznie na:

* dzień tygodnia - WeekDayNum
* dzień miesiąca - DayOfMonth
* miesiąc – MonthNum

A także Date2Num – czyli przedstawienie daty w formacie liczbowym w celu zawarcia w danych uczących cykliczności ww. zjawisk oraz opisanych poniżej (dane historyczne od 1971r.)

Rys. 5.2 przedstawia średnią dzienną zmienność pary walutowej EURUSD na przestrzeni dwunastu miesięcy. Wyraźnie można dostrzec okresy spadku zmienności (mała liczba transakcji) w okresie letnim, wakacyjnym – miesiące lipiec i sierpień.



Rys. 5.2. Dzienna zmienność pary walutowej EUR/USD w perspektywie dwunastu miesięcy. Średnia liczona za okres dziesięciu poprzednich tygodni. Wykresy pochodzą ze strony: http://www.investing.com/forex-tools/volatility-calculator

Zmienne OpenPrice, HighPrice, LowPrice, ClosePrice są standardowymi danymi charakteryzującymi każdą sesję wszystkich instrumentów giełdowych. Jednak przedstawienie ich na wykresie świecowym (japanese candlestick) obrazuje nie tylko zmianę ceny, ale także wprawnemu inwestorowi pozwala zauważyć **psychologię uczestników rynku**. Nieoficjalnie uznaje się informacje niesione poprzez ten rodzaj wykresów, jako znamienne w prawidłowej ocenie sytuacji rynkowej. Wszystkie te parametry mają więc ogromne znaczenie w predykcji.

Zmienna DiffCloseOpen oznacza różnicę między ceną otwarcia a zamknięcia. Jest to parametr mówiący o tym o ile PIPS zmieniła się cena na danej sesji.

Każdemu wektorowi danych wejściowych odpowiada wektor danych wzorcowych tzw. targetów. Programowo jest to zwykłe przesunięcie o jedną pozycję wektora cen zamknięcia w taki sposób, aby sieć dostosowywała swoje wagi w procesie uczenia, wzorując się na cenie zamknięcia z dnia następnego. Za wspomniane wyżej przesunięcie wektora odpowiada poniższa formuła programowa:

Targets = zeros(length(Data\_Num), 1);

for d = 1:Row\_Dt\_Num % 11280

if d < Row\_Dt\_Num

Targets(d, 1) = Data\_Num(d+1, 4);

else

Targets(d, 1) = Data\_Num(d, 4);

end

end

Jak wspomniano w rozdziale pierwszym, elementami strategii są również średnie kroczące. Ich wartości liczbowe reprezentowane są przez wektory: SMA7, SMA11, SMA23, SMA200.

Trzy pierwsze średnie kroczące są stosowane jako pomoc w określeniu trendu. Jeśli cena znajduje się ponad nimi oznacza to trend wzrostowy i prognozowanie powinno przeważać w stronę dodatnią. Jeśli cena znajduje się poniżej oznacza to trend spadkowy, a zatem prognozowanie powinno przeważać w stronę ujemną.

Wszystkie średnie użyte w procesie predykcji są zwykłymi średnimi arytmetycznymi i obliczanymi zgodnie ze wzorem:

|  |  |  |
| --- | --- | --- |
|  |  | (5.1) |

gdzie: n – oznacza liczbę okresów, a p0 ostatnią wartość ceny zamknięcia.

Ciekawą opcją jest średnia krocząca SMA200 (obliczana średnia arytmetyczna z cen zamknięcia za n = 200 poprzednich okresów). Jest to jeden z tych niewielu elementów w analizie technicznej rynków finansowych, któremu przypisuje się duże znaczenie, choć nie zostało to jak dotąd racjonalnie i dostatecznie zbadane ani wyjaśnione (potwierdzone naukowo). Jedynym wyjaśnieniem może być to, że określa ona pewien **nieznany cykl giełdowy** i może się to odbywać na takiej samej zasadzie jak np. używanie średnich 30 czy 50 okresowych na rynkach terminowych, gdzie liczba 56 oznacza cykl notowania danej serii kontraktów a następnie ich wygasanie i rozpoczęcie nowego cyklu.

W zasadzie o użyteczności danego wskaźnika (dowolnego) analizy technicznej mniej decyduje jego skomplikowana struktura, a bardziej popularność i powszechność. Jest to potwierdzenie założenia, że rynek jest tworzony przez ludzi a **cena jest wypadkową** ich **aktualnych poglądów**.

Ostatecznie macierz danych wejściowych prezentuje się jako macierz o rozmiarze R[13, N], gdzie N jest liczbą kolumn i zależy od ilości zaimportowanych danych historycznych.

Tabela 5.4. Gotowa macierz transponowana danych wejściowych jako dane uczące dla sieci neuronowej. Dane w każdym kolejnym wierszu, licząc od góry do dołu, zawierają rekordy danych w kolejności: Date2Num, WeekDayNum, DayOfMonth, MonthNum, OpenPrice, HighPrice, LowPrice, ClosePrice, DiffCloseOpen, SMA7, SMA11, SMA23, SMA200.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date2Num | 732338 | 732339 | 732340 | 732343 | 732344 | 732345 | 732346 | 732347 | 732350 |
| WeekDayNum | 4 | 5 | 6 | 2 | 3 | 4 | 5 | 6 | 2 |
| DayOfMonth | 26 | 27 | 28 | 31 | 1 | 2 | 3 | 4 | 7 |
| MonthNum | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| OpenPrice | 1,2973 | 1,3072 | 1,3039 | 1,3011 | 1,3043 | 1,3042 | 1,3030 | 1,2974 | 1,2853 |
| HighPrice | 1,3105 | 1,3123 | 1,3078 | 1,3056 | 1,3064 | 1,3092 | 1,3036 | 1,3035 | 1,2863 |
| LowPrice | 1,2963 | 1,3010 | 1,2983 | 1,2974 | 1,2999 | 1,3009 | 1,2940 | 1,2862 | 1,2732 |
| ClosePrice | 1,3070 | 1,3037 | 1,3042 | 1,3047 | 1,3041 | 1,3031 | 1,2976 | 1,2862 | 1,2750 |
| DiffCloseOpen | 0,0097 | -0,0035 | 0,0003 | 0,0036 | -0,0002 | -0,0011 | -0,0054 | -0,0112 | -0,0103 |
| SMA7 | 1,3009 | 1,3012 | 1,3017 | 1,3031 | 1,3036 | 1,3034 | 1,3035 | 1,3005 | 1,2964 |
| SMA11 | 1,3061 | 1,3042 | 1,3026 | 1,3022 | 1,3021 | 1,3023 | 1,3020 | 1,3011 | 1,2988 |
| SMA23 | 1,3224 | 1,3199 | 1,3174 | 1,3150 | 1,3124 | 1,3101 | 1,3080 | 1,3062 | 1,3040 |
| SMA200 | 1,2514 | 1,2520 | 1,2526 | 1,2532 | 1,2538 | 1,2543 | 1,2549 | 1,2553 | 1,2557 |

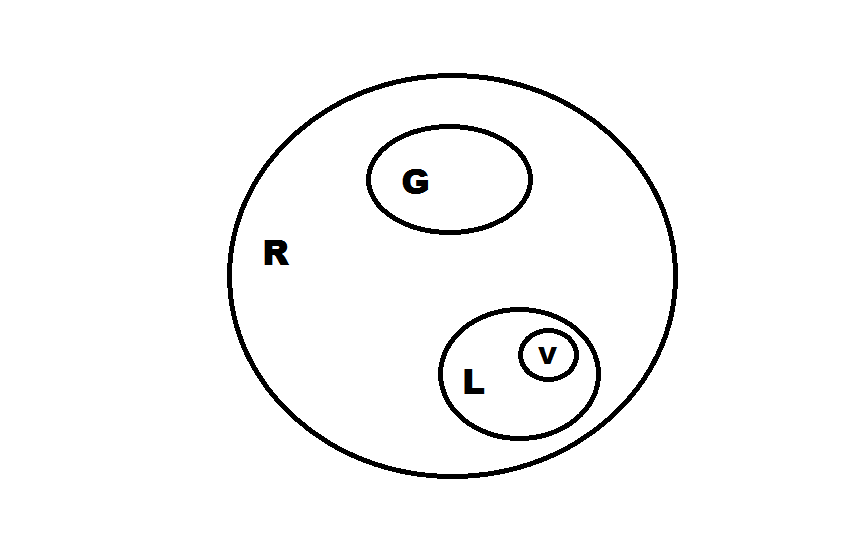
## Podział danych na uczące i testujące

W procesie nauczania sieci neuronowej dane uczące są dzielone wg schematu:

1. trenujące,
2. walidacyjne,
3. testujące.

Dobrze wytrenowana sieć na danych uczących powinna dawać poprawne wyniki na danych testujących, przy czym dane te (testujące) nie powinny być wcześniej jej znane, choć powinny należeć do tego samego zbioru danych, który nazwiemy zbiorem R. Zbiór ten został podzielony na dwa kolejne podzbiory L oraz G. W zbiorze L wydzielone zostały dane V weryfikujące stopień nauczenia sieci, czyli tzw. dane walidacyjne. Zdolność do odtwarzania zbioru L charakteryzuje zdolność sieci do zapamiętywania danych uczących, natomiast zdolność do rozpoznawania danych ze zbioru G, których sieć nie widziała w procesie nauki, jest nazywana zdolnością do generalizacji.

W zbiorze R powinny znaleźć się dane typowe dla danego zagadnienia. Oznacza to, że zbiór ten powinien zawierać dane charakteryzujące np. rynek walutowy Forex oraz parę walutową EUR/USD. W przeciwnym razie testując sieć na danych pochodzących z innych rynków, np. kontraktów terminowych bądź akcyjnych, sieć będzie dawała błędne wyniki. Regułę te prezentuje obrazowo rys. 5.1.



Rys. 5.3. Podział danych wejściowych na zbiór wartości typowych R oraz podzbiory – uczący L, weryfikujący stopień nauczenia V oraz testowy G, który nie był znany sieci w procesie uczenia.

Podział na dane w Matlabie może być ustawiony standardowo (default), lub jeśli zajdzie taka potrzeba, można każdy z tych parametrów ustawiać indywidualnie. Kod programu pozwalający na podział zbioru danych uczących na odpowiednie podzbiory:

net.divideParam.trainRatio = 0.7;

net.divideParam.valRatio = 0.15;

net.divideParam.testRatio = 0.15;

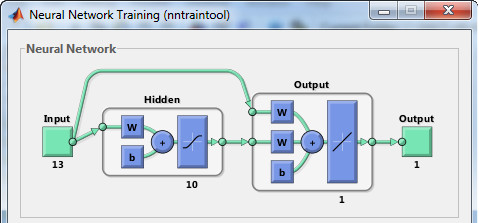
Gdzie zbiór uczący zawiera 70% danych, zbiór sprawdzający stopień nauczenia sieci 15% i zbiór testujący 15%.

## Badanie właściwości predykcyjnych wybranych typów sieci

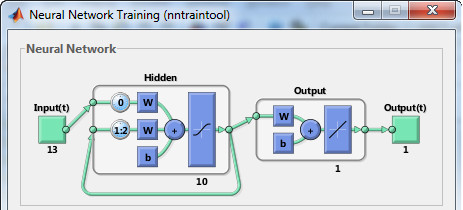
W trakcie badania możliwości predykcyjnych sieci neuronowej przebadano możliwości poprawy predykcji pod względem szybkości obliczeń oraz minimalizacji błędu prognozy dla różnych parametrów sieci takich jak [8]:

1. struktura sieci
   1. sieć bez sprzężenia zwrotnego
   2. sieć ze sprzężeniem warstwy ukrytej i warstwy wejściowej
   3. sieć z połączeniem typu forward
2. liczby danych tj. wielkości zbioru danych historycznych
3. ilości zmiennych

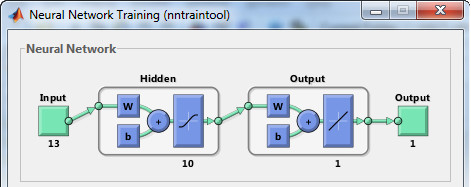
Schematy zastosowanych topologii sieci:



Rys. 5.4. Topologia sieci typu Cascadeforwardnet



Rys. 5.5. Topologia sieci typu Elmannet



Rys. 5.6. Topologia sieci typu Feedforwardnet

Tabela 5.4. Zestawienie wyników badania różnych form sieci, w tym sieci bez sprzężenia zwrotnego (feedforwardnet), ze sprzężeniem zwrotnym warstwy ukrytej i warstwy wejściowej (elmannet) oraz sieci kaskadowej, gdzie dane uczące były przekazywane bezpośrednio również do warstwy wyjściowej (cascadeforwardnet)

|  |  |  |  |
| --- | --- | --- | --- |
|  | feedforwardnet | elmannet (layerdelays 1:2) | cascadeforwardnet |
| Obliczenia dla danych uczących:  Max\_Teach\_Down - Maksymalne odchylenie prognozy w dół  Max\_Teach\_Up - Maksymalne odchylenie prognozy w górę | | | |
| Max\_Teach\_Up | +0.0656 | +0.0774 | +0.0641 |
| Max\_Teach\_Down | -0.0362 | -0.0717 | -0.0454 |
| Obliczenia dla danych testujących:  Max\_Test\_Down - Maksymalne odchylenie prognozy w dół  Max\_Test\_Up - Maksymalne odchylenie prognozy w górę | | | |
| Max\_Test\_Up | +0.0310 | +0.0507 | +0.0306 |
| Max\_Test\_Down | -0.0290 | -0.0298 | -0.0291 |
| l. epok | 15 (Best: 9) | 1000(Best:998) | 661(Best: 655) |
| gradient | 5.0255e-05 | 1.1697e-03 | 1.0684e-04 |
| czas | 9.43 sek. | 295.78 sek. | 313.20 sek. |

## Wpływ ilości danych uczących oraz ilości zmiennych na własności prognostyczne sieci neuronowej

W rozdziale 5.3 dokonano próby oceny poprawności działania trzech rodzajów sieci neuronowych. Zastosowano podział pod względem budowy modelu i zastosowanych wzajemnych połączeń pomiędzy poszczególnymi warstwami sieci. Wyniki tych badań jednoznacznie wskazują, że do predykcji kursu giełdowego najlepiej nadają się sieci bez sprzężeń zwrotnych, o jednokierunkowym przesyle informacji.

Oczywiście należy również wziąć pod uwagę sposób przedstawienia danych wejściowych, co ma decydujący wpływ nie tylko na działanie sieci, ale również predestynuje jej wybór oraz topologię.

Dalsze badania sieci neuronowej można podzielić na następujące etapy:

* badanie wpływu ilości danych uczących,
* badanie wpływu liczby zmiennych wejściowych,
* badanie wpływu liczby neuronów ukrytych .

Zbiór danych wejściowych jest macierzą o rozmiarze *xi = [N, K]* = [13, 11280], która zawiera sformatowane dane giełdowe za okres od 1971-01-04 do 2015-03-05, gdzie N jest liczbą wierszy a *K* liczbą kolumn.

Podział ze względu na ilość danych uczących i testujących jest wykonywany programowo za pomocą poniższego kodu:

% Podzial na dane:

% uczace i testujace - dane uczace X, testujace Y.

% Div\_Scale = 1/10 oznacza 90%/10%, uczace / testujace.

% Div\_Scale = 2/10 oznacza 80%/20%, uczace / testujace.

% Div\_Scale = 3/10 oznacza 70%/30%, uczace / testujace.

% Div\_Scale = 4/10 oznacza 60%/40%, uczace / testujace.

% Div\_Scale = 5/10 oznacza 50%/50%, uczace / testujace.

[Row\_Inputs\_Size, Col\_Inputs\_Size] = size(Inputs);

relative\_num = 1;

Div\_Scale = relative\_num / 10;

Col\_Resizer = (Col\_Inputs\_Size-Col\_Inputs\_Size\*(Div\_Scale));

Inputs\_Teach = Inputs(:, 1:Col\_Resizer);

Targets\_Teach = Targets(1, 1:Col\_Resizer);

[Row\_Data\_Teach, Col\_Data\_Teach] = size(Inputs\_Teach);

Inputs\_Test = Inputs(:, Col\_Resizer:end);

Targets\_Test = Targets(1, Col\_Resizer:end);

Tabela 5.5. Tabela przedstawiająca podział danych na dane uczące i testujące z całego zbioru liczącego 11280 rekordów historycznych pary EUR/USD.

|  |  |  |
| --- | --- | --- |
| Współczynnik podziału | Dane uczące [%] | Dane testujące [%] |
| 0.1  0.2  0.3  0.4  0.5  0.6  0.7 | 90  80  70  60  50  40  30 | 10  20  30  40  50  60  70 |

Tabela 5.6. Zestawienie wyników prognoz w zależności od ilości danych uczących.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Wpół. podziału danych | Rozmiar macierzy danych uczących | Rozmiar macierzy danych testujących | Maksymalne błędy prognozy dla danych uczących | | Maksymalne błędy prognozy dla danych testujących | |
| prognoza w górę | prognoza w dół | prognoza w górę | prognoza w dół |
| 0.7 | [13,3384] | [13,7897] | +0.0637 | -0.0389 | +0.0180 | -0.2287 |
| 0.6 | [13,4512] | [13,6769] | +0.0647 | -0.0387 | +0.0445 | -0.0575 |
| 0.5 | [13,5640] | [13,5641] | +0.0647 | -0.0391 | +0.0263 | -0.1169 |
| 0.4 | [13,6768] | [13,4513] | +0.0641 | -0.0403 | +0.0307 | -0.0716 |
| 0.3 | [13,7896] | [13,3385] | +0.0643 | -0.0390 | +0.0361 | -0.0530 |
| 0.2 | [13,9024] | [13,2257] | +0.0629 | -0.0396 | +0.0360 | -0.0468 |
| **0.1** | **[13,1012]** | **[13,1129]** | **+0.0638** | **-0.0447** | **+0.0309** | **-0.0282** |

Z przeprowadzonych badań trenowania sieci na różnej ilości danych uczących wynika, że **im większa liczba danych, tym wyniki prognozy są lepsze**. W przypadku prognozowania danych giełdowych należy sprawdzic czy dane bedace uczącymi w całym swoim zbiorze maja taka sama strukturę.

Ujmując to w ramach czasowych oznacza to, że poprawne prognozy są możliwe tylko przy dysponowaniu zakresem danych historycznych większym niż **10 lat dla danych EOD (End of Day), czyli danych dziennych.**

## Badanie wpływu liczby zmiennych na maksymalny błąd prognozy

## Badanie wpływu liczby neuronów ukrytych na maksymalny błąd prognozy

Tabela 5.7. Zestawienie wyników prognoz w zależności od ilości danych uczących.

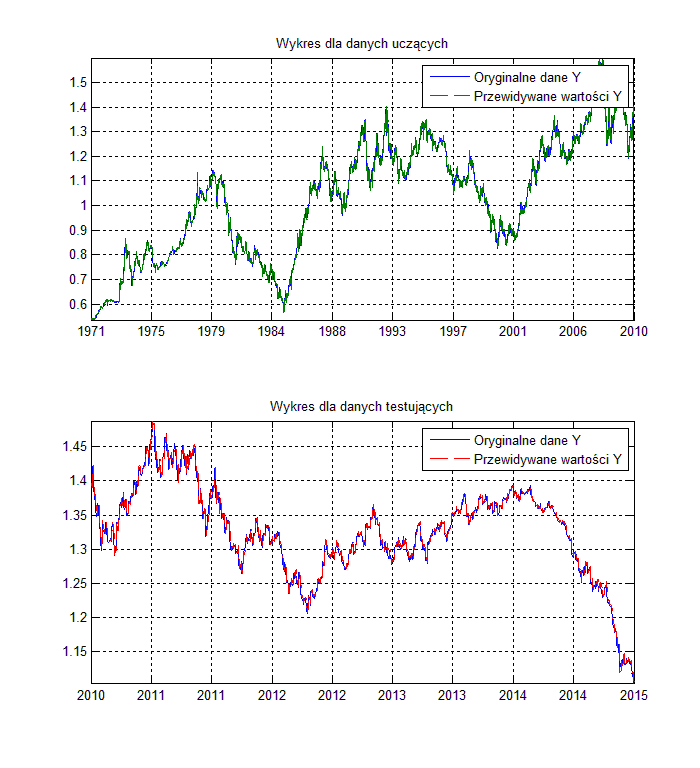
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Liczba neuronów w warstwie ukrytej | Czas uczenia [sek.] | Liczba epok uczenia | Maksymalne błędy prognozy dla danych uczących | | Maksymalne błędy prognozy dla danych testujących | |
| prognoza w górę | prognoza w dół | prognoza w górę | prognoza w dół |
| 10 | 16.27 | 22 | +0.0621 | -0.0467 | +0.0335 | -0.0273 |
| 11 | 42.09 | 61 | +0.0639 | -0.0458 | +0.0308 | -0.0291 |
| 12 | 15.33 | 20 | +0.0644 | -0.0457 | +0.0302 | -0.0279 |
| 13 | 12.28 | 15 | +0.0641 | -0.0467 | +0.0310 | -0.0278 |
| 14 | 11.56 | 13 | +0.0617 | -0.0471 | +0.0322 | -0.0300 |
| 15 | 13.08 | 14 | +0.0637 | -0.0453 | +0.0308 | -0.0293 |
| 25 | 55.19 | 34 | +0.0640 | -0.0470 | +0.0304 | -0.0309 |
| 50 | 54.59 | 16 | +0.0541 | -0.0415 | +0.0399 | -0.0303 |
| 100 | 158.02 | 14 | +0.0486 | -0.0629 | +0.0408 | -0.0340 |

W badaniu wpływu liczby neuronów ukrytych na przebieg uczenia oraz wartości prognozowanych posłużono się metodą **doświadczalną** doboru ich ilości. Zaczęto od standardowej wielkości 10 neuronów i przebadano właściwości sieci zwiększając tę liczbę do 100 neuronów w kolejności oraz liczbie jak przedstawiono w pierwszej kolumnie tabeli 5.7. Z analizy uzyskanych danych wynika, że **optymalna liczba neuronów** ukrytych to 13.

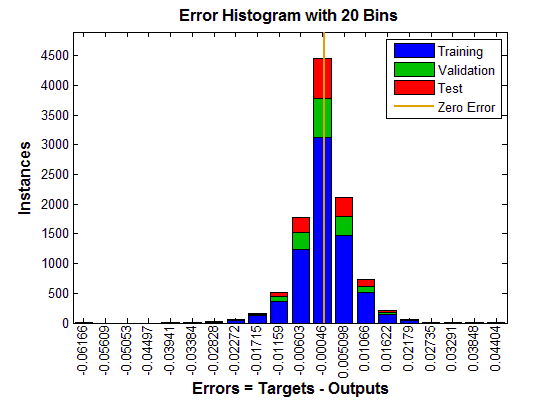
Wybór został podyktowany oceną parametrów sieci i wyników takich jak (w kolejności pod względem ważności):

1. błąd prognozy dla danych testujących,
2. liczba epok uczenia,
3. czas przetwarzania danych (trenowania),.

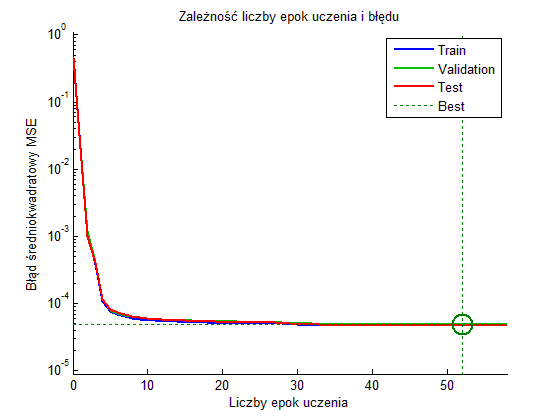
## Szczegółowe przedstawienie wyników trenowania sieci o optymalnej topologii oraz parametrach wejściowych



Rys. 5.7. Wykres górny przedstawiający dane rzeczywiste i dane prognozowane dla danych uczących. Charakterystyczną cechą otrzymanych wyników jest fakt, że największe błędy prognozy mają tendencję do pojawiania się w newralgicznych miejscach wykresu, gdzie następuje zmiana trendu (szczyty i dołki).



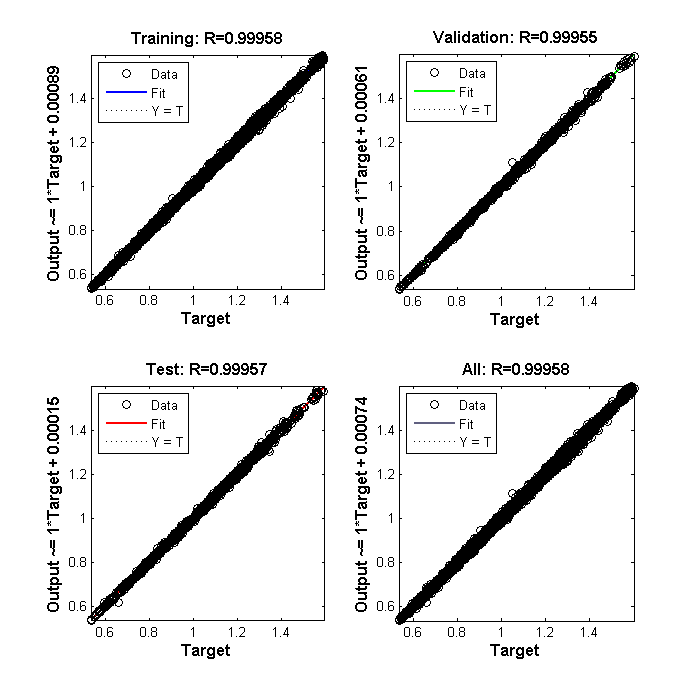
Rys. 5.8. Histogram przedstawiający rozkład błędów na zbiorze uczącym z podziałem na dane trenujące (dla wybranego przykładu 70%), sprawdzające (15%) oraz testujące (15%). Wyraźna tendencja do minimalizacji dużych błędów prognoz – mała liczba wystąpień.



Rys. 5.9. Histogram przedstawiający rozkład błędów na zbiorze uczącym z podziałem na dane trenujące (dla wybranego przykładu 70%), sprawdzające (15%) oraz testujące (15%).

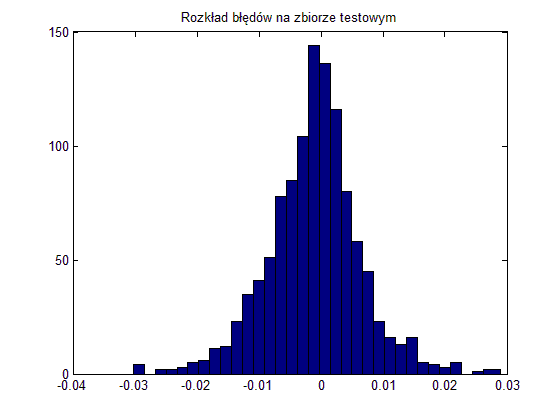
Uczenie sieci przebiega bardzo płynnie i równomiernie. Brak gwałtowanych oscylacji o dużej amplitudzie, co wskazuje na małe skoki zmian wag. Oznacza to, że współczynniki uczenia, momentum oraz **funkcje aktywacji są dobrane prawidłowo**.

Współczynnik uczenia odpowiada za szybkość uczenia i ma bezpośredni związek z gradientem. Punkt poruszający się po powierzchni funkcji błędu, w zależności od wielkości skoku (zmiany) może albo poruszać się po wierzchołkach lub utknąć w minimum lokalnym, nie odnajdując minimum globalnego. Dodatkowo zmiana wartości wag może być regulowana za pomocą współczynnika momentum.



Rys. 5.10. Wykres regresji przedstawiający dane wyjściowe sieci (prognoza – outputs) w stosunku do odpowiadających im danych wejściowych (targets).

Wykres regresji pokazany na rys. 5.10 przedstawia dane wyjściowe sieci (prognoza – outputs) w stosunku do odpowiadających im danych wejściowych (targets). Współczynnik R>9.3 oznacza dobre dopasowanie obydwu zbiorów. Idealna regresja R=1 nie występuje w praktyce i oznacza idealne dopasowanie.

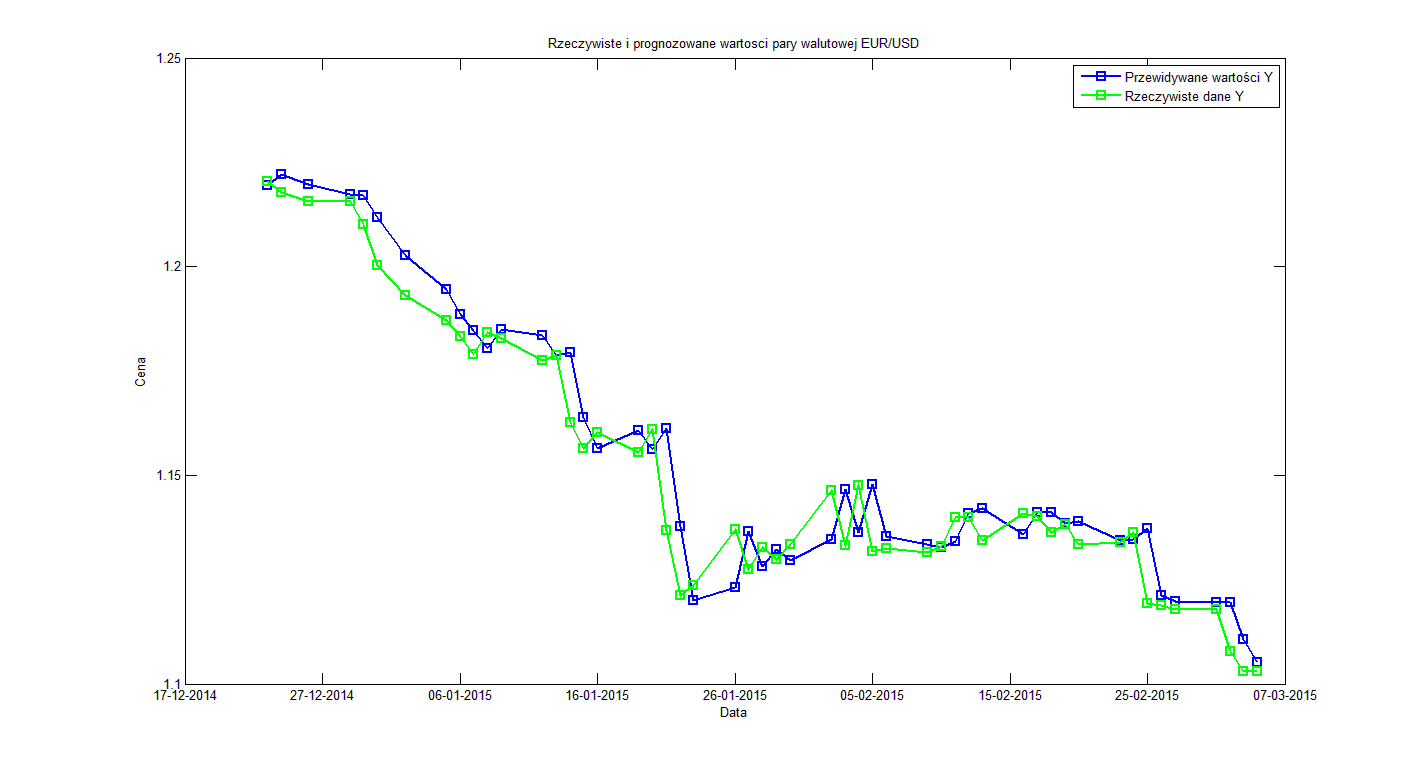


Rys. 5.12. Rozkład błędów prognozy na zbiorze testowym. Liczba wystąpień błędów najmniejszych jest największa (największych najmniejsza) z całego zbioru, co dobrze rokuje na prognozę.

Tabela 5.8. Zestawienie przykładowych wartości rzeczywistych i prognozowanych.

|  |  |  |
| --- | --- | --- |
| Data | Wartości rzeczywiste | Wartości prognozowane |
| 2012-09-27  2012-09-28  2012-10-01  2012-10-02  2012-10-03  2012-10-04  2012-10-05  2012-10-08  2012-10-09  2012-10-10  2012-10-11  2012-10-12  2012-10-15  2012-10-16  2012-10-17  2012-10-18  2012-10-19  2012-10-22  2012-10-23  2012-10-24  2012-10-25  2012-10-26  2012-10-29  2012-10-30  2012-10-31  2012-11-01 | 1.2858  1.2885  1.2916  1.2908  1.3015  1.3032  1.2966  1.2883  1.2850  1.2929  1.2951  1.2945  1.3098  1.3114  1.3068  1.3019  1.3064  1.2983  1.2969  1.2941  1.2941  1.2903  1.2966  1.2957  1.2946  1.2836 | 1.2920  1.2861  1.2887  1.2912  1.2910  1.2984  1.3009  1.2984  1.2904  1.2867  1.2924  1.2942  1.2964  1.3117  1.3121  1.3074  1.3024  1.3082  1.3003  1.2987  1.2949  1.2945  1.2918  1.2979  1.2964  1.2928 |

Tabela 5.8 przedstawia przykładowe wartości rzeczywiste i prognozowane za okres od 27-09-2012 do 1-11-2012. Graficzna prezentacja danych z tabeli znajduje się na rys. 5.13 z dodatkowymi danymi w celu dokładniejszego zaprezentowania wyników wizualnie.



Rys. 5.13. Przykładowy wykres przestawiający wzajemne położenie wartości rzeczywistych oraz prognozowanych dla pary walutowej EURUSD.

# Podsumowanie

Zastosowany w pracy model prostej sieci jednokierunkowej, składającej się z dwu warstw: wejściowej i ukrytej o trzynastu neuronach ukrytych i trzynastu wektorach wejściowych, wykazuje prawidłowe rozpoznawanie w ceny na jedną sesję w przód. Przy czym wyraźnie widać, że największy wpływ na prognozę ma najbliższa prognozowana wartość z sesji poprzedniej. Jest to również zaobserwowana zbiorze wejściowym odpowiednich wartości, pozwalających na dość dokładne prognozowanie wartości prawidłowość i reguła rynkowa, ponieważ ceny instrumentów giełdowych nie zmieniają się w sposób całkowicie nieokreślony, lecz w statystycznie określonym przedziale wartości, co zostało przedstawione na rysunkach 5.1 oraz 5.2 w sposób obrazowy.

Z badań wpływu ilości danych giełdowych do nauczania sieci można wywnioskować, że im większy zbiór danych historycznych, tym lepsza prognoza, przy czym zbiór zbyt mały (tabela 5.5 i 5.6) nie pozwala sieci na wytrenowanie z akceptowalnym marginesem błędu.

Badanie związane z doborem topologii sieci oraz zastosowaniem sprzężeń zwrotnych i innych wzajemnych połączeń pomiędzy poszczególnymi warstwami sieci, wykazało, że najlepszym rozwiązaniem pod względem szybkości obliczeń, liczby epok uczenia oraz powstałego błędu prognozy, będą sieci jednokierunkowe bez sprzężeń i połączeń pomiędzy warstwami.

Zastosowano zatem model feedforwardnet z algorytmem uczenia **Levenberga-Marquardta** i **wsteczną propagacją błędów**.

Kolejnym ważnym czynnikiem w tego typu badaniach jest odpowiedni dobór liczby neuronów ukrytych.

Doświadczalnie ustalono, że liczba 13 neuronów jest liczbą optymalną oraz, że:

* zbyt mała liczba neuronów ukrytych (13<) powoduje zwiększanie liczby epok uczenia oraz czasu obliczeń bez zauważalnego wpływu na błąd prognozy,
* zbyt duża liczba neuronów ukrytych (>15) znacząco wydłuża czas obliczeń, przy stosunkowo niezmiennej liczbie epok uczenia, lecz zwiększając błąd prognozy,

Rozkład błędów na zbiorze testowym wskazuje, że większość błędów prognozy znajduje się w przedziale od 0 do -60 PIPS dla błędu prognozy w kierunku spadkowym i od 0 do +40 PIPS dla błędu prognozy w kierunku wzrostowym (rys.5.12), czyli najwięcej było błędów małych a najmniej błędów dużych. Biorąc pod uwagę statystyczne wielkości zmian dziennych ceny, gdzie średnia zmienność wynosiła od 110 PIPS do 190 PIPS dla całego tygodnia handlowego, wyniki prognoz są do zaakceptowania.

# Zakończenie

Zagadnienie wykorzystania sieci neuronowych do predykcji kursów giełdowych jest zagadnieniem bardzo trudnym i stanowi duże wyzwanie dla współczesnej analizy technicznej rynków finansowych. Jakkolwiek sztuczne sieci neuronowe wykazują tendencje do prawidłowego radzenia sobie z problemem predykcji, tak prognozowanie dokładnej wartości liczbowej analizowanego instrumentu giełdowego nie jest optymalnym rozwiązaniem w przeciwieństwie do budowy modelu decyzyjnego, który miałby za zadanie wskazywanie jedynie kierunku przewidywanej zmiany ceny, tj. (1) – kup, (0) – poza rynkiem, (-1) – sprzedaj.

Zmiana ceny na giełdzie to wynik wielu czynników, zarówno natury ekonomicznej i gospodarczej, jak również wielu innych teorii rynkowych takich jak np. teoria fal Elliotta, teoria cykli, teoria harmoniczności, geometrii i symetrii. To nasuwa stwierdzenie, że każda z tych teorii mogłaby stanowić osobny element do analizy przez odpowiednio zaprojektowaną sieć (wiele sieci równocześnie), z tym, że nie każdy czynnik giełdowy może być w prosty sposób, jeśli w ogóle, przedstawiony w postaci algorytmicznej (matematyczno-programowej). Stąd nie tylko trudność w odpowiednim przygotowaniu modeli neuronowych, ale odpowiednim opisie matematycznym problemu.

Sugerowanym rozwiązaniem mogłoby być zastosowanie dodatkowo elementów, jakie stosuje się w klasycznym tradingu automatycznym, zaadoptowanych dla potrzeb sieci. Czynnikami tymi mogłyby być informacje o formacjach cenowych występujących na wykresach giełdowych. Zarówno formacjach świecowych jak i klasycznych czy harmonicznych.

Przykładowy model mógłby zatem zawierać sieci klasyfikujące SVM (rozwiazywanie zadań klasyfikacji danych separowanych) i samoorganizujących się SOM [12] do segregacji świec japońskich [7, 13] pod względem wielkości korpusu i maksymalnej zmienności (cena maksymalna i minimalna z sesji widoczna w postaci cieni świec).

# Bibliografia

[1] LeBeau Ch., Lucas D. W., Komputerowa analiza rynków terminowych, wyd. WIG PRESS 1999r.

[2] Murphy J. J., Międzyrynkowa analiza techniczna. Strategie inwestycyjne na rynkach akcji, obligacji, towarów i walut, wyd. WIG PRESS 1998r.

[3] Plummer T., Psychologia rynków finansowych, wyd. WIG PRESS 1995r.

[4] Gately E., Sieci neuronowe Prognozowania finansowe i projektowanie systemów transakcyjnych, wyd. WIG PRESS 1999r.

[5] Bulkowski N., Encyclopedia of Chart Patterns, 2nd Edition. John Wiley and Sons, 2005

[6] F. W., Landt, Stock Price Prediction using Neural Networks, Master thesis Leiden University, 1997

[7] Ken-ichi Kainijo, Tetsuji Tanigawa, Stock Price Pattern Recognition - A Recurrent Neural Network Approach, C&C Information Technology Research Laboratories,  
NEC Corporation

[8] Lawrence R., Using Neural Networks to Forecast Stock Market Prices, Department of Computer Science University of Manitoba, December 12, 1997

[9] Domaradzki R., Zastosowanie sieci neuronowych do generowania strategii decyzyjnych (inwestycyjnych) na przykładzie Giełdy Papierów Wartościowych, Rozprawa Doktorska

[10] Borowski K., Zastosowanie szybkiej transformaty Fouriera do analizy cykli cen akcji notowanych na Giełdzie Papierów Wartościowych w Warszawie i wchodzących w skład Indeksów giełdowych: WIG20, mWIG40 I sWIG80, Warsaw School of Economics, 2013

[11] Jasiński G., Zastosowanie sieci neuronowych do prognozowania wskaźników finansowych, Uniwersytet Jagielloński Akademia Ekonomiczna w Krakowie

[12] Ciskowski P., Zaton M., Neural pattern recognition with self-organizing maps for efficient processing of forex market data streams, Institute of Computer Engineering, Control and Robotics, Wroclaw University of Technology

[13] Donn S. Fishbein, MD, PhD, Identifying Short Term Market Turns Using Neural Networks and Genetic Algorithms, Neuroquant.com

[14] Duzinkiewicz K., Grochowski M., Sztuczne sieci neuronowe – wprowadzenie oraz dziedziny i przykłady zastosowań, Katedra Inżynierii systemów Sterowania, 2008-2009

[15] Osowski S., Sieci neuronowe do przetwarzania informacji, Oficyna Wydawnicza Politechniki Warszawskiej, 2006

[16] Bernstein J., Cykle giełdowe, Wydawnictwo WIG PRESS, ISBN: 8390329689, 1996

[17] Nison S., Świece i inne japońskie techniki analizowania wykresów, Wydawnictwo WIG , ISBN: 8390329646, 1996

[18] Tadeusiewicz R., Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#, Polska Akademia Umiejętności, Kraków 2007.

[19] Fischer R., Liczby Fibonacciego na giełdzie, Wydawnictwo WIG PRESS, ISBN: 8387014001, 1996

[20] Murphy J. J., Analiza techniczna rynków finansowych, , Wydawnictwo WIG PRESS, ISBN: 8387014222, 1999

[21] Elder A., Zawód inwestor giełdowy - Psychologia rynków. Taktyka inwestycyjna. Zarządzanie portfelem, Wydawnictwo Wolters Kluwer, ISBN: 9788326407772, 2012

[22] Frost A. J., Prechter R. R., Teoria fal Elliotta, Wydawnictwo WIG PRESS, ISBN: 839032962X, 1995

[23] Danielewicz P., Geometria Fibonacciego, Wydawnictwo WIG PRESS, ISBN: 8389241366, 2006

# Dodatek A. Kod programu komputerowego

clc;

clear;

close all;

% format shortG ,format compact

format long ,format compact

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% Import i konwersja danych wejsciowych %%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

file\_import = 'EURUSD\_STOOQ.csv';

[imported\_data, Data\_Text, Data\_Num, w1, k1, w2 ,k2] = data\_import(file\_import);

[Date2Num, WeekDayNum, MonthNum, DayOfMonth, OpenPrice, HighPrice, LowPrice, ClosePrice, ...

Row\_Dt\_Num, Col\_Dt\_Num, SMA7, SMA11, SMA23, SMA200, DiffCloseOpen, ...

Targets] = ConvDates(w1, w2, Data\_Text, Data\_Num);

% % Zlozenie danych liczbowych w macierz danych wejsciowych Inputs w

% % formacie:

% % (1) Data przekonwertowana na liczbe

% % (2) Dzien tygodnia, 1 - niedziale, 2 - pon., 3 - sroda, ...

% % (3) Dzien miesiaca, 1, 2, 3, 4, 5 ...

% % (4) Miesiac, 1 - styczen, 2 - luty, ...

% % (5) OpenPrice - cena otwarcia sesji

% % (6) HighPrice - najwyzsza cena sesji

% % (7) LowPrice - najnizsza cena sesji

% % (8) ClosePrice - cena zamkniecia sesji

% % (9) Roznica miedzy cena otwarcia a cena zamkniecia (Close - Open)

% % (10) SMA7

% % (11) SMA11

% % (12) SMA23

% % (13) SMA200

% % (1) 7.199e+05 7.199e+05 7.199e+05 7.199e+05 7.199e+05

% % (2) 2 3 4 5 6

% % (3) 4 5 6 7 8

% % (4) 1 1 1 1 1

% % (5) 0.5353 0.535 0.5352 0.5353 0.5354

% % (6) 0.5353 0.535 0.5352 0.5353 0.5354

% % (7) 0.5353 0.535 0.5352 0.5353 0.5354

% % (8) 0.5353 0.535 0.5352 0.5353 0.5354

% % (9) 0 0 0 0 0

% % (10) 0.5353 0.53515 0.53517 0.5352 0.53524

% % (11) 0.5353 0.53515 0.53517 0.5352 0.53524

% % (12) 0.5353 0.53515 0.53517 0.5352 0.53524

% % (13) 0.5353 0.53515 0.53517 0.5352 0.53524

Inputs = [Date2Num WeekDayNum DayOfMonth MonthNum OpenPrice HighPrice ...

LowPrice ClosePrice DiffCloseOpen SMA7 SMA11 SMA23 SMA200];

Inputs = Inputs';

Targets = Targets';

% % Podzial na dane: uczace i testujace - dane uczace X %, testujace Y %.

% % Div\_Scale = 1 / 10 oznacza podział 90% / 10%, uczące / testujące.

% % Div\_Scale = 2 / 10 oznacza podział 80% / 20%, uczące / testujące.

% % Div\_Scale = 3 / 10 oznacza podział 70% / 30%, uczące / testujące.

% % Div\_Scale = 4 / 10 oznacza podział 60% / 40%, uczące / testujące.

% % Div\_Scale = 5 / 10 oznacza podział 50% / 50%, uczące / testujące.

[Row\_Inputs\_Size, Col\_Inputs\_Size] = size(Inputs(:, :));

relative\_num = 1; % Im weeks, tym mniej danych uczacych a wiecej testujacych!

Div\_Scale = relative\_num / 10;

Col\_Resizer = (Col\_Inputs\_Size - Col\_Inputs\_Size \* (Div\_Scale));

Inputs\_Teach = Inputs(:, 1:Col\_Resizer);

Targets\_Teach = Targets(1, 1:Col\_Resizer);

[Row\_Data\_Teach, Col\_Data\_Teach] = size(Inputs\_Teach);

Inputs\_Test = Inputs(:, Col\_Resizer:end);

Targets\_Test = Targets(1, Col\_Resizer:end);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% Utworzenie i trenowanie sieci neuronowej %%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% hiddenLayerSize = input('Ilosc neuronow w warstwie ukrytej: ');

hiddenLayerSize = 10;

% trainFcn = input('Algorytm trenujacy (default:trainlm): ', 's');

trainFcn = 'trainlm';

net = feedforwardnet(hiddenLayerSize, trainFcn);

% net = cascadeforwardnet(hiddenLayerSize, trainFcn);

% % W tym, w danych uczacych nastepuje kolejny wewnetrzny podzial:

net.divideParam.trainRatio = 0.7;

net.divideParam.valRatio = 0.15;

net.divideParam.testRatio = 0.15;

% % zmiana funkcji aktywacji w warstwie ukrytej i wyjsciowej

net.layers{1}.transferFcn = 'tansig'; % <- logsig, tansig, purelin

net.layers{2}.transferFcn = 'purelin'; % <- logsig, tansig, purelin

[netw1, tr] = train(net, Inputs\_Teach, Targets\_Teach);

Outputs\_Teach = netw1(Inputs\_Teach);

Differ\_Teach = Outputs\_Teach-Targets\_Teach;

Errors\_Teach = gsubtract(Targets\_Teach, Outputs\_Teach);

Performance\_Teach = perform(netw1, Targets\_Teach, Outputs\_Teach);

% % view(net);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Testowanie wytrenowanej sieci na nowych, nieznanych dotad danych %%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Outputs\_Test = netw1(Inputs\_Test);

Differ\_Test = Outputs\_Test-Targets\_Test;

Errors\_Test = gsubtract(Targets\_Test, Outputs\_Test);

Performance\_Test = perform(netw1, Targets\_Test, Outputs\_Test);

Outputs\_Test = Outputs\_Test';

fig\_position(tr, Targets\_Test, Targets\_Teach, Inputs\_Teach, Inputs\_Test, ...

Outputs\_Teach, Outputs\_Test, Errors\_Test);

disp2screen(Differ\_Teach, Differ\_Test, Inputs\_Test, Targets\_Test, Outputs\_Test);

disp2file(Differ\_Teach, Differ\_Test, Div\_Scale, hiddenLayerSize, trainFcn, net, tr);

disp('####### KONIEC OBLICZEŃ #######')

­­

function [SMA7, SMA11, SMA23, SMA200] = mov\_averages(Data\_Num\_f, Row\_Dt\_Num\_f)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% Srednie kroczace oraz wskazniki (ewentualnie) %%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % % Srednia kroczaca SMA7 z cen zamkniecia

Col\_Price = 4; % Kolumna danych z 1-Open, 2-High, 3-Low, 4-Close

SMA7(1, 1) = Data\_Num\_f(1, Col\_Price); % Wyliczenie/Przypisanie pierwszej pozycji

SMA\_Period\_1 = 7;

for j = 2:Row\_Dt\_Num\_f

if j <= SMA\_Period\_1

k = j;

else

k = SMA\_Period\_1;

end

SMA7(j, 1) = 0;

for i = (j-k+1):j

SMA7(j, 1) = SMA7(j, 1) + Data\_Num\_f(i, Col\_Price);

end

SMA7(j, 1) = SMA7(j, 1) / k;

end

% % % Srednia wykladnicza SMA11

SMA11(1, 1) = Data\_Num\_f(1, Col\_Price); % Wyliczenie/Przypisanie pierwszej pozycji

SMA\_Period\_2 = 11;

for j = 2:Row\_Dt\_Num\_f

if j <= SMA\_Period\_2

k = j;

else

k = SMA\_Period\_2;

end

SMA11(j, 1) = 0;

for i = (j-k+1):j

SMA11(j, 1) = SMA11(j, 1) + Data\_Num\_f(i, Col\_Price);

end

SMA11(j, 1) = SMA11(j, 1) / k;

end

% % % Srednia wykladnicza SMA23

SMA23(1, 1) = Data\_Num\_f(1, Col\_Price); % Wyliczenie/Przypisanie pierwszej pozycji

SMA\_Period\_3 = 23;

for j = 2:Row\_Dt\_Num\_f

if j <= SMA\_Period\_3

k = j;

else

k = SMA\_Period\_3;

end

SMA23(j, 1) = 0;

for i = (j-k+1):j

SMA23(j, 1) = SMA23(j, 1) + Data\_Num\_f(i, Col\_Price);

end

SMA23(j, 1) = SMA23(j, 1) / k;

end

% % % Srednia wykladnicza SMA200

SMA200(1, 1) = Data\_Num\_f(1, Col\_Price); % Wyliczenie/Przypisanie pierwszej pozycji

SMA\_Period\_4 = 200;

for j = 2:Row\_Dt\_Num\_f

if j <= SMA\_Period\_4

k = j;

else

k = SMA\_Period\_4;

end

SMA200(j, 1) = 0;

for i = (j-k+1):j

SMA200(j, 1) = SMA200(j, 1) + Data\_Num\_f(i, Col\_Price);

end

SMA200(j, 1) = SMA200(j, 1) / k;

end

end

function [Date2Num, WeekDayNum, MonthNum, DayOfMonth, OpenPrice, HighPrice, LowPrice, ClosePrice, ...

Row\_Dt\_Num, Col\_Dt\_Num, SMA7, SMA11, SMA23, SMA200, DiffCloseOpen, Targets] = ConvDates(w1, w2, Data\_Text, Data\_Num)

% % Konwersja daty na liczbe - [732654;732655;732657;] ...

formatIn = 'yyyy-mm-dd';

[Date2Num] = ConvDate2Num(formatIn, w2, Data\_Text);

% Wyliczenie numeru dnia tygodnia - 1, 2, 3, 4, 5, 6, 7 ...

Size\_Dt\_Num = size(Date2Num);

for n0 = 1:Size\_Dt\_Num

WeekDayNum(n0, 1) = weekday(Date2Num(n0, 1));

end

% % % Wyliczenie numeru miesiaca. Styczen = 1, Luty = 2 itd.

for n1 = 1:Size\_Dt\_Num

MonthNum(n1, 1) = month(Date2Num(n1, 1));

end

% % % Wyliczenie dnia miesiaca. 1, 2, 3 ... 30, 31 ...

for n2 = 1:Size\_Dt\_Num

DayOfMonth(n2, 1) = day(Date2Num(n2, 1));

end

OpenPrice = Data\_Num(:, 1);

HighPrice = Data\_Num(:, 2);

LowPrice = Data\_Num(:, 3);

ClosePrice = Data\_Num(:, 4);

[Row\_Dt\_Num Col\_Dt\_Num] = size(Data\_Num); % dane do obliczen pobierane z Data\_Num

[SMA7, SMA11, SMA23, SMA200] = mov\_averages(Data\_Num, Row\_Dt\_Num);

% % % Roznica miedzy cena otwarcia a cena zamkniecia (Close - Open)

for j = 1:w1

DiffCloseOpen(j, 1) = Data\_Num(j, 4) - Data\_Num(j, 1);

end

% % Obliczenie targetu. Target jest cena zamkniecia z dnia "nastepnego"

for d = 1:Row\_Dt\_Num

if d < Row\_Dt\_Num

Targets(d, 1) = Data\_Num(d+1, 4);

else

Targets(d, 1) = Data\_Num(d, 4);

end

end

% Konwersja daty na liczbe - [732654;732655;732657;] ...

% [Date2Num] = ConvDate2Num(formatIn, w2, Data\_Text);

function Date2Num = ConvDate2Num(formatIn, w2, Data\_Text)

for j = 1:w2-1

Date2Num(j,1) = datenum(Data\_Text(j,1), formatIn);

end

function fig\_position(tr, Targets\_Test, Targets\_Teach, Inputs\_Teach, Inputs\_Test, ...

Outputs\_Teach, Outputs\_Test, Errors\_Test, Date2Num)

% % Sterowanie wyswietlaniem na ekranie - wielkosc i rozm. okien oraz

% % rozmieszczenie okien na calym ekranie

% Wykres dla danych uczących - zestawienie danych uczacych Targets\_Teach

% z odpowiedzia sieci Outputs\_Teach

fig1 = figure;

plot(Inputs\_Teach(1, :), Targets\_Teach, Inputs\_Teach(1, :), Outputs\_Teach, '--');

title('Wykres dla danych uczących');

legend('Oryginalne dane Y', 'Przewidywane wartości Y');

% Zależność liczby epok uczenia i błędu

fig2 = figure;

plotperform(tr);

title('Zależność liczby epok uczenia i błędu');

xlabel('Liczby epok uczenia');

ylabel('Błąd średniokwadratowy MSE')

% Wykres dla danych testujacych - zestawienie danych uczacych Targets\_Test

% z odpowiedzia sieci Outputs\_Test

fig3 = figure;

plot(Inputs\_Test(1, :), Targets\_Test, Inputs\_Test(1, :), Outputs\_Test, 'r--');

title('Wykres dla danych testujących');

legend('Oryginalne dane Y', 'Przewidywane wartości Y');

fig4 = figure;

k\_sqrt = sqrt(length(Errors\_Test));

hist(Errors\_Test, k\_sqrt);

title('Rozkład błędów na zbiorze testowym');

fig5 = figure;

plottrainstate(tr);

fig6 = figure;

plotregression(Targets\_Teach,Outputs\_Teach);

fig7 = figure;

DataNumber = Date2Num(10651:10676, 1);

Prognosis = Outputs\_Test(500:525, 1);

RealNumber = Inputs\_Test(8, 500:525);

hh = plot(DataNumber, Prognosis, DataNumber, RealNumber);

datetick('x','dd-mm-yyyy', 'keeplimits', 'keepticks')

set(hh,'LineWidth', 2, {'LineStyle'}, {'-';'-'})

set(hh,{'Marker'}, {'square'; 'square'})

set(hh,{'Color'},{'b'; 'g'})

xlabel('Data')

ylabel('Cena')

title('Rzeczywiste i prognozowane wartosci pary walutowej EUR/USD (27-09-2012 do 1-11-2012)');

legend('Rzeczywiste dane Y', 'Przewidywane wartości Y');

set(0,'Units','pixels')

get(0,'ScreenSize');

scnsize = get(0,'ScreenSize');

position = get(fig1,'Position');

outerpos = get(fig1,'OuterPosition');

borders = outerpos - position;

edge = -borders(1)/2;

task\_bar\_down\_px = 40;

pos1 = [edge, ...

(scnsize(4) \* (1/2)) + 20, ...

scnsize(3)/3 - edge, ...

(scnsize(4)/2) - 20];

pos2 = [scnsize(3)/3 + edge, ...

pos1(2), ...

pos1(3), ...

pos1(4)];

pos3 = [edge, ...

task\_bar\_down\_px, ...

pos1(3), ...

pos1(4)];

pos4 = [scnsize(3)/3 + edge, ...

task\_bar\_down\_px, ...

pos1(3), ...

pos1(4)];

pos5 = [scnsize(3) - scnsize(3)/3 + edge, ...

pos1(2), ...

pos1(3), ...

pos1(4)];

pos6 = [scnsize(3) - scnsize(3)/3 + edge, ...

task\_bar\_down\_px, ...

pos1(3), ...

pos1(4)];

pos7 = [scnsize(3) - scnsize(3)/1.25 + edge, ...

task\_bar\_down\_px + 10, ...

scnsize(3)/1.75, ...

scnsize(3)/2];

set(fig1,'OuterPosition',pos1);

set(fig2,'OuterPosition',pos2);

set(fig3,'OuterPosition',pos3);

set(fig4,'OuterPosition',pos4);

set(fig5,'OuterPosition',pos5);

set(fig6,'OuterPosition',pos6);

set(fig7,'OuterPosition',pos7);

function disp2screen(Differ\_Teach, Differ\_Test, Inputs\_Test, Targets\_Test, Outputs\_Test)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%% Wypisanie wynikow na ekran %%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% for i = 500:525

% fprintf('%1.4f\n', Inputs\_Test(8, i))

% end

% for i = 500:525

% fprintf('%1.4f\n', Outputs\_Test(i, 1))

% end

disp('originalne prognoza roznica rozpietosc(max-min) korpus')

for i = 1:length(Targets\_Test)

max\_min(1, i) = (Inputs\_Test(6, i) - Inputs\_Test(7, i));

corpses(1, i) = (Inputs\_Test(5, i) - Inputs\_Test(8, i));

if (Differ\_Test(1, i) > 0 && corpses(1, i) > 0)

% fprintf('Differ\_Test(1, i = %d) = %1.4f corpses(1, i) = %1.4f\n', i, Differ\_Test(1, i), corpses(1, i));

fprintf('%1.4f %1.4f +%1.4f %1.4f +%1.4f\n', ...

Targets\_Test(i), Outputs\_Test(i), Differ\_Test(i), max\_min(1, i), ...

corpses(1, i));

elseif (Differ\_Test(1, i) < 0 && corpses(1, i) < 0)

% fprintf('Differ\_Test(1, i = %d) = %1.4f corpses(1, i) = %1.4f\n', i, Differ\_Test(1, i), corpses(1, i));

fprintf('%1.4f %1.4f %1.4f %1.4f %1.4f\n', ...

Targets\_Test(i), Outputs\_Test(i), Differ\_Test(i), max\_min(1, i), ...

corpses(1, i));

elseif (Differ\_Test(1, i) > 0 && corpses(1, i) < 0)

% fprintf('Differ\_Test(1, i = %d) = %1.4f corpses(1, i) = %1.4f\n', i, Differ\_Test(1, i), corpses(1, i));

fprintf('%1.4f %1.4f +%1.4f %1.4f %1.4f\n', ...

Targets\_Test(i), Outputs\_Test(i), Differ\_Test(i), max\_min(1, i), ...

corpses(1, i));

elseif (Differ\_Test(1, i) < 0 && corpses(1, i) > 0)

% fprintf('Differ\_Test(1, i = %d) = %1.4f corpses(1, i) = %1.4f\n', i, Differ\_Test(1, i), corpses(1, i));

fprintf('%1.4f %1.4f %1.4f %1.4f +%1.4f\n', ...

Targets\_Test(i), Outputs\_Test(i), Differ\_Test(i), max\_min(1, i), ...

corpses(1, i));

end

if i >= length(Targets\_Test)

fprintf('\n')

end

end

disp('Obliczenia dla danych uczacych')

fprintf('[Uczenie] Maksymalne odchylenie prognozy w gore: +%1.4f\n', max(Differ\_Teach));

fprintf('[Uczenie] Maksymalne odchylenie prognozy w dol: %1.4f\n\n', min(Differ\_Teach));

disp('Obliczenia dla danych testujacych')

fprintf('[Test] Maksymalne odchylenie prognozy w gore: +%1.4f\n', max(Differ\_Test));

fprintf('[Test] Maksymalne odchylenie prognozy w dol: %1.4f\n\n', min(Differ\_Test));

function disp2file(Differ\_Teach, Differ\_Test, Div\_Scale, hiddenLayerSize, trainFcn, net, tr)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%% Zapisywanie wynikow do pliku %%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fileID = fopen('zrzut\_danych.txt', 'w'); % 'a' - dodaj na koniec pliku

message = ferror(fileID);

fseek(fileID, 0, 'eof');

fprintf(fileID, '\n');

% % disp('Obliczenia dla danych uczacych')

fprintf(fileID, '[Uczenie] Maksymalne odchylenie prognozy w gore: +%1.4f\n', max(Differ\_Teach));

fprintf(fileID, '[Uczenie] Maksymalne odchylenie prognozy w dol: %1.4f\n\n', min(Differ\_Teach));

% % disp('Obliczenia dla danych testujacych')

fprintf(fileID, '[Test] Maksymalne odchylenie prognozy w gore: +%1.4f\n', max(Differ\_Test));

fprintf(fileID, '[Test] Maksymalne odchylenie prognozy w dol: %1.4f\n', min(Differ\_Test));

fprintf(fileID, '\nWspol. podzialu danych: %1.1f', Div\_Scale);

fprintf(fileID, '\nDane uczace: %1.0f%%', (1 - Div\_Scale) \* 100);

fprintf(fileID, '\nDane testujace: %1.0f%%', (1 - (1 - Div\_Scale)) \* 100);

fprintf(fileID, '\n');

fprintf(fileID, '\nNeurony w warstwie ukrytej: %d', hiddenLayerSize);

fprintf(fileID, '\nAlgorytm trenujacy trainFcn: %s', trainFcn);

fprintf(fileID, '\nnet.divideParam.trainRatio: %1.2f', net.divideParam.trainRatio);

fprintf(fileID, '\nnet.divideParam.valRatio: %1.2f', net.divideParam.valRatio);

fprintf(fileID, '\nnet.divideParam.testRatio: %1.2f', net.divideParam.testRatio);

fprintf(fileID, '\nnet.layers{1}.transferFcn: %s', net.layers{1}.transferFcn);

fprintf(fileID, '\nnet.layers{2}.transferFcn: %s', net.layers{2}.transferFcn);

fprintf(fileID, '\n');

fprintf(fileID, '\nLiczba epok uczenia: %d (Best: %d)', tr.num\_epochs, tr.best\_epoch);

fprintf(fileID, '\nCzas uczenia sieci: %1.2f sek.', tr.time(1, length(tr.time)));

fprintf(fileID, '\nGradient: %1.4e', tr.gradient(length(tr.gradient)));

fprintf(fileID, '\n############################################################');

fclose(fileID);

function [imported\_data, Data\_Text, Data\_Num, w1, k1, w2, k2] = data\_import(~)

% Import danych z formatu CSV (MT4 RT Loader from MT4 - EOD data)

% Dane z plikow CSV sa pobierane z katalogu roboczego np. C:\MATLAB\R2012a\bin\work

% Dane uzyte do testow i trenowania ponizszej sieci zostaly pobrane z serwisu STOOQ.PL.

% Dane historyczne zawieraja dane od poczatku notowan pary EUR/USD.

% Do testow mozna takze uzyc danych wyeksportowanych z platformy brokerskiej

% np. MT4 za pomoca narzedzia MT4 RT Loader w formacie jak ponizej:

% Date,Open,High,Low,Close,Volume,OpenInt

% 2005-12-08,1.17130,1.18500,1.17010,1.18110,13288,0.0

% 2005-12-09,1.18100,1.18370,1.17640,1.18120,6400,0.0

% 2005-12-11,1.17870,1.17870,1.17870,1.17870,1,0.0

% 2005-12-12,1.17870,1.19760,1.17870,1.19550,17709,0.0

imported\_data = importdata('EURUSD\_STOOQ.csv');

[w1, k1] = size(imported\_data.data);

[w2,k2] = size(imported\_data.textdata);

Data\_Num = imported\_data.data(1:w1,1:4);

Data\_Text = imported\_data.textdata(2:w2, 1);

function static\_loop()

% Siec podczas uczenia ma tendencje do wytrenowania z bledem "grubym",

% na skutek, którego wynik obliczeń/treningu może znacząco odbiegać od

% wyników statystycznych osiąganych poprzez

% wielokrotne próby trenowania. W celu jego wyeliminowania siec można

% trenować wielokrotnie za pomocą pętli do osiągniecia błędu wyników

% mniejszego od statystycznego.

clear maxx minn mean\_maxx mean\_minn

n = 15;

for k = 1:n

clear tr netw1

[netw1, tr] = train(net, Inputs\_Teach, Targets\_Teach);

Outputs\_Teach = netw1(Inputs\_Teach);

Differ\_Teach = Outputs\_Teach-Targets\_Teach;

maxx(k, 1) = max(Differ\_Teach);

minn(k, 1) = min(Differ\_Teach);

mean\_maxx = mean(maxx);

mean\_minn = mean(minn);

% Zapisywanie do pliku txt

fileIDD = fopen('max\_min\_Differ\_Teach.txt', 'a');

fseek(fileIDD, 0, 'eof');

fprintf(fileIDD, '%1.4f %1.4f', maxx(k, 1), minn(k, 1));

fprintf(fileIDD, '\n');

if (maxx(k, 1) > 0.0639 || minn(k, 1) < -0.0394) && k < 10

fprintf('k = %d || max %1.4f mean\_max %1.4f || min %1.4f mean\_min %1.4f\n', ...

k, maxx(k, 1), mean\_maxx, minn(k, 1), mean\_minn);

elseif (maxx(k, 1) > 0.0639 || minn(k, 1) < -0.0394) && (k >= 10 && k <=15)

fprintf('k = %d || max %1.4f mean\_max %1.4f || min %1.4f mean\_min %1.4f\n', ...

k, maxx(k, 1), mean\_maxx, minn(k, 1), mean\_minn);

end

if (maxx(k, 1) <= 0.0639 && minn(k, 1) >= -0.0394)

break

end

end

fprintf('\n');

fclose(fileIDD);

# Dodatek B. Spis zawartości dołączonej płyty CD

Praca – niniejszy dokument

www – witryny internetowe wykorzystane w pracy

1. Dotyczy to tylko danych odnośnie wolumenu oraz ewentualnie liczby otwartych pozycji. Parametry Open, High, Low i Close są z reguły bardzo zbliżone do siebie. [↑](#footnote-ref-1)