Przemysław GARSZTKA, Marek ŁAŻEWSKI

WYKORZYSTANIE SIECI NEURONOWYCH DO PROGNOZOWANIA SZEREGÓW FINANSOWYCH O WYSOKIEJ CZĘSTOTLIWOŚCI, NA PRZYŁADZIE NOTOWAŃ AKCJI NA GPW W WARSZAWIE

Wstęp

W ciągu ostatnich kilkunastu lat gwałtowny rozwój rynków kapitałowych oraz technologii informatycznych spowodował możliwość gromadzenia informacji, dotyczących cen instrumentów finansowych, obserwowanych w odstępach czasowych krótszych niż dzień (ang. *high frequency data*). Obecnie dostępne są bazy danych, zawierające dane rejestrowane w zasadzie w dowolnych odstępach czasu, a także ceny wszystkich transakcji zawieranych na rynku giełdowym w systemie notowań ciągłych[1]. Poza cenami archiwizowane są w zasadzie wszystkie informacje o zdarzeniach, zaistniałych podczas sesji giełdowych, między innymi dane określające wolumen obrotu, ilości i ceny złożone w oczekujących ofertach kupna – sprzedaży, czas zajścia transakcji. Globalizacja rynku kapitałowego oraz to że handel giełdowy na kuli ziemskiej trwa praktycznie 24 godziny na dobę wywołały zmianę nastawienia uczestników rynku względem ryzyka związanego z posiadaniem instrumentów finansowych, a także możliwości wykorzystania danych o wysokiej częstotliwości w procesie prognostycznym struktury czasowej analizowanych szeregów czasowych [18]. Jedną z pierwszych instytucji komercyjnych (prowadzącą ponadto działalność

[1] Na przykład http://www.olsendata.com; baza danych stworzona w Katedrze Ekonometrii Akademii Ekonomicznej w Poznaniu, http://www.katek.ae.poznan.pl.

badawczą) zajmującą się wykorzystywaniem danych o wysokiej częstotliwości, między innymi tworzeniem automatycznych systemów transakcyjnych, jest *Olsen & Associates*, mająca siedzibę w Zurichu, której współpracownicy wydali jedną z pierwszych monografii w tym zakresie [4].

Wszechstronne analizy danych o wysokiej częstotliwości wskazały na specyficzne cechy badanych szeregów czasowych, do których można zaliczyć:

- istnienie grubych ogonów w empirycznej funkcji rozkładu prawdopodobieństwa oraz znacząco większej kurtozy w stosunku do rozkładu normalnego (ang. *heavy – tailed and peakedness*), opisywane najczęściej za pomocą klasy rozkładów stabilnych [14],
- zmienność wariancji w czasie (ang. *time – varying volatility*), która jest opisywana przeważnie przez modele grupy ARCH i GARCH [5],
- występowanie efektów długotrwałej pamięci i długotrwałych zależności (ang. *long memory, long – range dependence*) [13],
- kointegrację szeregów czasowych [11].

Jednoczesne uwzględnianie wszystkich wymienionych cech szeregów czasowych (które w swojej istocie wywoływane są nieliniowymi oddziaływaniami uczestników rynku) poprzez budowę i estymację wielorównaniowych modeli ekonometrycznych, wiążących zmienne endogeniczne i egzogeniczne, jest zadaniem praktycznie niewykonalnym na obecnym etapie rozwoju ekonometrii finansowej oraz ekonomii matematycznej [1]. Problemy związane z budową wielorównaniowych nieliniowych modeli przyczynowo-skutkowych mają swoje konsekwencje w niewystarczającej jakości prognoz generowanych przez takie modele. Alternatywą w tym przypadku jest wykorzystanie do prognozowania sieci neuronowych oraz tak zwanych sztucznych inteligencji, których zastosowania można wskazać w wielu dziedzinach nauki, począwszy od meteorologii, poprzez analizę i rozpoznawanie obrazów, senergetykę, powstawanie i rozwój struktur samoorganizujących się, nieliniowe systemy złożone, a skończywszy na procesach chaotycznych oraz dynamice molekularnej [23].

1. Sieci neuronowe

Sztuczna sieć neuronowa jest siecią, zawierającą dużą liczbę prostych procesorów sumujących (neuronów), zorganizowanych w równoległe struktury (warstwy), komunikujące się między sobą poprzez sprzężenia, którym przypisane są wagi. Model sztucznej sieci neuronowej oparty jest na uproszczonym ujęciu

procesów, zachodzących podczas przetwarzania informacji w mózgu ludzkim [12]. Sieci neuronowe są, oprócz systemów eksperckich i algorytmów genetycznych, narzędziami sztucznej inteligencji.

Do rozwiązywania większości problemów wystarczają sieci trójwarstwowe, składające się z warstwy wejściowej, która przyjmuje sygnały warstwy ukrytej i warstwy wyjściowej dającej wyniki [16]. Warstwa wyjściowa powinna liczyć tyle neuronów, ile wymaga odpowiedź na postawiony problem. Natomiast nie ma jednoznacznej odpowiedzi, ile neuronów powinna liczyć warstwa ukryta. Często można spotkać się z regułą, według której liczba neuronów w warstwie ukrytej powinna równać się pierwiastkowi z iloczynu liczby neuronów warstwy wejściowej oraz wyjściowej [12]. Inna reguła została wskazana przez Bauma oraz Hausslera [3]. Mówi ona, że należy znaleźć taką architekturę, w której liczba neuronów w warstwie ukrytej będzie jak największa, ale równocześnie nie wystąpi podczas trenowania problem osiągania przez sieć tzw. minimów lokalnych.

Oprócz warstw, sieci mogą również różnić się układem połączeń między neuronami. Można wyróżnić dwie główne grupy:

- sieci jednokierunkowe – sygnały są podawane na wejście, następnie przechodzą przez warstwę ukrytą i w końcu zostają przetworzone w warstwie wyjściowej,
- sieci ze sprzężeniem zwrotnym – uzyskuje się je przez dodanie połączeń w kierunku od wyjść neuronów do ich wejść. Zamknięcie pętli sprzężenia zwrotnego powoduje, że sygnały na wyjściach zależą zarówno od aktualnego stanu wejść, jak i od poprzednich sygnałów wejściowych [23].

Zadaniem neuronu jest zsumowanie iloczynów wartości wejściowych i wag przypisanych poszczególnym synapsom, czyli obliczenie wyniku funkcji postsynaptycznego potencjału neuronu, której wynikiem jest potencjał (poziom) aktywacji neuronu. Poziom aktywacji neuronu wykorzystywany jest jako argument funkcji aktywacji neuronu. Funkcja aktywacji neuronu $y=\varphi(\mathbf{e})$, przekształca potencjał aktywacji neuronu (wektor $\mathbf{e}$ wartości wejściowych) w jego sygnał wyjściowy. Jako funkcję aktywacji najczęściej stosuje się funkcje sigmoidalne, wywodzące się z funkcji logistycznej, na przykład:

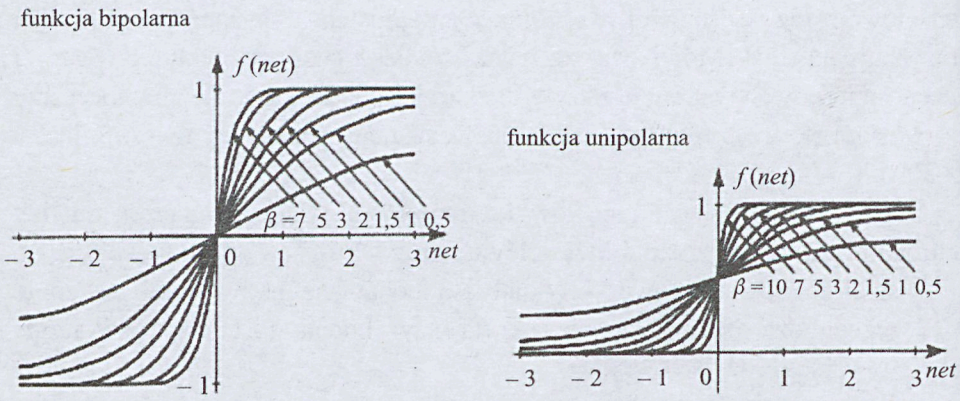
- bipolarną (tzn. przyjmującą wartości z przedziału $[-1, 1]$):

$$\varphi(\mathbf{e})=\frac{2}{1+\exp(-\beta\mathbf{e})}-1, \tag{1}$$

– unipolarną (wartości z przedziału [0, 1]):

$$\varphi(\mathbf{e}) = \frac{1}{1+\exp(-\beta \mathbf{e})}, \tag{2}$$

gdzie β oznacza parametr decydujący o szybkości, z jaką funkcja dąży do krańców przedziału.

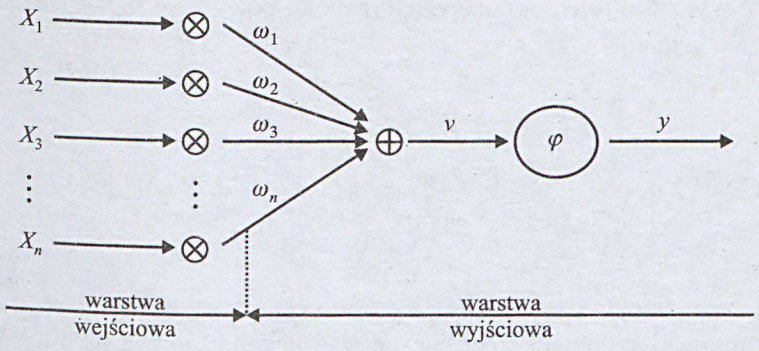


Rys. 1. Kształtowanie się sigmoidalnej funkcji aktywacji

Źródło: [21], s. 6, 7

Wektor **e** wartości wejściowych nazywany jest często presynaptyczną aktywnością neuronu. Każdej z synaps przyporządkowana jest odpowiednia waga ω_1. Poziom aktywacji neuronu jest iloczynem wektora presynaptycznej aktywności i wektora wag, czyli wynikiem funkcji postsynaptycznego potencjału

$$PSP(\boldsymbol{\omega}, \mathbf{X}) = v = [\omega_1\ \omega_2\ \ldots\ \omega_n] \cdot \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} = \sum_{i=1}^{n} \omega_i X_i. \tag{3}$$



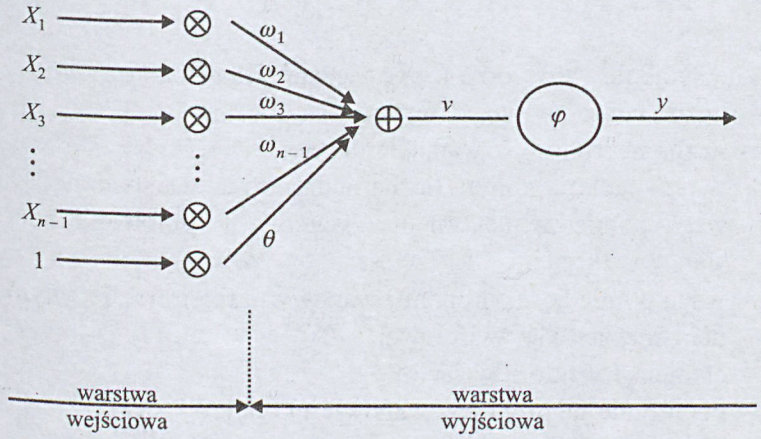
⊗ synapsy

⊕ miejsce sumowania (miejsce obliczenia funkcji postsynaptycznego potencjału neuronu)

Rys. 2. Schemat działania neuronu

Wartość wyjściowa neuronu jest elementem wektora presynaptycznej aktywności dla neuronów należących do kolejnej warstwy, z którymi dany neuron jest połączony.

Czasami wartości wejściowe neuronu są poszerzane o tak zwane obciążenie θ. Wtedy model neuronu przybiera nieco inną postać.



Rys. 3. Schemat działania neuronu z obciążeniem

W tym wypadku funkcja postsynaptycznego potencjału może zostać przedstawiona w następujący sposób:

$$\mathbf{X}=\begin{bmatrix} X_1 \\ \vdots \\ X_{n-1} \\ 1 \end{bmatrix}; \quad PSP(\boldsymbol{\omega}, \mathbf{X}, \theta)=v=\left(\sum_{i=1}^{n-1} \omega_i X_i\right)+\theta. \tag{4}$$

Następnym etapem budowy sieci neuronowej jest jej uczenie. W przypadku jednokierunkowej sztucznej sieci neuronowej uczenie polega na odpowiednim ustawieniu wag we wszystkich warstwach oprócz wejściowej. Proces uczenia jest dostosowywaniem wag (i obciążenia) przypisanych poszczególnym synapsom wszystkich neuronów w sieci. Dostosowanie to odbywa się na podstawie zbioru uczącego, który zawiera wartości wektorów wejściowych **X** oraz odpowiadające tym wartościom oczekiwane wzorcowe wyniki. Celem uczenia jest zminimalizowanie rozbieżności pomiędzy wynikami obliczonymi przez sieć neuronową dla poszczególnych wejść i wynikami odpowiadającymi tym wejściom w zbiorze uczącym.

Perceptron wielowarstwowy jest jednym z najprostszych modeli sieci neuronowych. Jego prosta budowa pozwala na analityczne wyznaczenie pierwszej pochodnej cząstkowej funkcji reprezentującej wartość wyjściową perceptronu. Postać funkcyjna perceptronu wielowarstwowego może zostać przedstawiona w następujący sposób [23]:

$$MLP(I)=f\left(\sum_{j=1}^{J} \omega_2(j, 1)\cdot f\left(\sum_{i=1}^{N} \omega_1(i, j)\cdot X_i+\theta_1(j)\right)+\theta_2(1)\right). \tag{5}$$

Poszczególne symbole we wzorze (5) oznaczają odpowiednio:

N – liczba neuronów wejściowych,
J – liczba neuronów w warstwie ukrytej,
1 – w przypadku ogólnym liczba neuronów w warstwie wyjściowej,
$\omega_1(i, j)$ – waga pomiędzy neuronem i warstwy wejściowej i neuronem j warstwy ukrytej,
$\omega_2(j, 1)$ – waga pomiędzy neuronem j warstwy ukrytej oraz jedynym neuronem w warstwie wyjściowej,
X_i – element i wektora wejściowego,
$\theta_1(j)$ – obciążenie neuronu j w warstwie ukrytej,
$\theta_2(1)$ – obciążenie jedynego neuronu warstwy wyjściowej,
f – funkcja aktywacji (rosnąca i różniczkowalna).

Ze względu na to, że wszystkie sieci neuronowe stworzone w ramach niniejszego opracowania będą uczone z wykorzystaniem algorytmu wstecznej propagacji błędu (ang. *back propagation*, BP), poniżej zostanie przedstawiona jego krótka charakterystyka [22]. Przyjęte zostaną następujące oznaczenia:

$\mathbf{X}=[x_1\ x_2\ \ldots\ x_N]^T$ – wektor wartości wejściowych,
$\mathbf{H}=[h_1\ h_2\ \ldots\ h_J]^T$ – wektor wartości wyjściowych z warstwy ukrytej,
$\mathbf{O}=[o_1\ o_2\ \ldots\ o_M]^T$ – wektor wartości wyjściowych z warstwy wyjściowej,
$\mathbf{T}_p=[t_{p1}\ t_{p2}\ \ldots\ t_{pM}]^T$ – wektor rzeczywistych wartości wyjściowych odpowiadający p-temu wektorowi wartości wejściowych,
W_{jk} – waga pomiędzy neuronem j warstwy wejściowej oraz neuronem k warstwy ukrytej,
ω_{ij} – waga pomiędzy neuronem i warstwy ukrytej oraz neuronem j warstwy wyjściowej,
a_i – wartość funkcji postsynaptycznego potencjału dla neuronu i,
g – funkcja aktywacji neuronu.

Korzystając z podanych oznaczeń wartość wyjściową neuronu i warstwy wyjściowej można zapisać w następujący sposób:

$$o_i=g\left(\sum_{j=1}^{L}\omega_{ij}h_j\right)=g\left(\sum_{j=1}^{L}\omega_{ij}g\left(\sum_{k=1}^{N}W_{jk}x_k\right)\right). \tag{6}$$

Dostosowanie wag może odbywać się w dwóch trybach. W pierwszym z nich wagi dostosowywane są na podstawie błędu obliczonego dla wszystkich wektorów sieci. W drugim wagi są dostosowywane w sposób ciągły, po każdym wektorze uczącym.

W trybie, polegającym na dostosowywaniu wag w sposób ciągły, błąd oszacowania wartości wyjściowych, obliczonych z wykorzystaniem sieci neuronowej, reprezentowany jest jako suma kwadratów różnic pomiędzy tymi wartościami i wartościami rzeczywistymi (elementami wektora $\mathbf{T}_p$). Błąd ten obliczany jest z wykorzystaniem następującej funkcji:

$$E=\frac{1}{2}\sum_{i=1}^{M}(t_{pi}-o_i)^2=\frac{1}{2}\sum_{i=1}^{M}\left(t_{pi}-g\left(\sum_{j=1}^{L}\omega_{ij}g\left(\sum_{k=1}^{N}W_{jk}x_k\right)\right)\right)^2. \tag{7}$$

Celem uczenia sieci jest minimalizacja wartości funkcji E. Proces ten odbywa się przez modyfikowanie wag. Dla wag pomiędzy warstwą wejściową i ukrytą sieci neuronowej zmiana wartości poszczególnych wag określona jest następującym wzorem:

$$\Delta W_{jk} = -\eta \frac{\partial E}{\partial o_i} \cdot \frac{\partial o_i}{h_j} \cdot \frac{\partial h_j}{\partial W_{jk}} = \eta \sum_{i=1}^{M} (t_i - o_i) \cdot g'(a_i) \cdot \omega_{ij} \cdot g'(a_j) \cdot x_k, \tag{8}$$

$$\delta_i = (t_i - o_i) \cdot g'(a_i), \tag{9}$$

w którym η jest parametrem uczenia.

Z wykorzystaniem zapisu sformułowanego w (9) funkcja (8) może zostać zapisana w następujący sposób:

$$\Delta W_{jk} = \eta \cdot g'(a_j) \cdot x_k \sum_{i=1}^{M} \delta_i \omega_{ij}. \tag{10}$$

Pomiędzy warstwą ukrytą i warstwą wyjściową zmiana wartości poszczególnych wag określona jest następującym wzorem:

$$\Delta \omega_{ij} = -\eta \cdot \frac{\partial E}{\partial \omega_{ij}} = -\eta \cdot \frac{\partial E}{\partial o_i} \cdot \frac{\partial o_i}{\partial \omega_{ij}} = \eta \cdot (t_1 - o_i) \cdot g'(a_i) \cdot h_j. \tag{11}$$

Funkcja (11) może zostać zapisana w następujący sposób:

$$\Delta \omega_{ij} = \eta \cdot \delta_i \cdot h_j. \tag{12}$$

Wagi zmieniane są w niżej objaśniony sposób dopóki nie zostanie osiągnięty określony, tzw. warunek stopu (osiągnięcie określonej dokładności w oszacowaniu wartości wyjściowych za pomocą sieci, ustabilizowanie się wag itd.).

Oczywiście

$$W_{p+1} = W_p + \Delta W, \qquad \omega_{p+1} = \omega_p + \Delta \omega. \tag{13}$$

2. Prognozowanie finansowych szeregów czasowych za pomocą sieci neuronowych

Możliwość opisu nieliniowej dynamiki finansowych szeregów czasowych za pomocą sieci neuronowych jest wykorzystywana do prognozowania zachowania takich systemów w przyszłości. W ostatniej dekadzie można znaleźć wiele prac naukowych, poświęconych tej tematyce. Można tutaj wymienić pracę Honga i Rognvaldssona, którzy wykorzystali niestacjonarną sztuczna sieć neuronową do predykcji cen kontraktów terminowych [10]. System prognostyczny rynku akcji oparty na sieciach neuronowych z połączeniem modeli średnich arytmetycznych i geometrycznych stóp zwrotu zbudowany został przez Haefkego i Halmensteina, kórzy wykorzystali go do zbudowania strategii inwestycyjnej, dającej lepsze wyniki niż na przykład standardowa strategia kup i trzymaj [8]. Bardzo szeroki przegląd możliwości zastosowania sieci neuro-

nowych na rozwiniętych rynkach kapitałowych można znaleźć również w pracy Gatleya [6]. Problem klasyfikacji spółek giełdowych – z punktu widzenia ich pozycji finansowej – na Giełdzie Papierów Wartościowych w Warszawie z wykorzystaniem różnych sztucznych sieci neuronowych poruszył Tarczyński, który analizował 20 spółek, będących podstawą indeksu WIG20 [17]. Inne zastosowanie sieci neuronowych do prognozowania cen akcji z wykorzystaniem rekurencyjnych, wielowarstwowych architektur zaprezentował Bassi [2]. Badania dotyczące możliwości zastosowania sieci neronowych do analizy szeregów czasowych notowań pochodzących z rynków wschodzących przeprowadził Siriopoulos, który zastosował zarówno jedno-, jak i wielowymiarowe sieci i wykorzystał dwie różne odmiany algorytmu wstecznej propagacji błędu [15]. Szereg badań dotyczyło porównań prognoz uzyskiwanych na podstawie standardowych metod statystycznych oraz prognoz generowanych przez różnej złożoności sieci neuronowe. Eksperymenty przeprowadzone przez Wonga, Tsibourisa i Zeidenberga oraz Hiemstra pozwoliły na wyciągnięcie wniosków, że w analizowanych szeregach czasowych istnieją znaczące związki nieliniowe, które z powodzeniem można opisać za pomocą odpowiednich sieci neuronowych ([9], [20], [22]).

3. Opis eksperymentu

3.1. Wybór próby empirycznej

Przy wyborze próby do badań empirycznych autorzy kierowali się przede wszystkim kryterium płynności danego waloru, rozumianego jako prawdopodobieństwo zawarcia transakcji [9]. Wybrane spółki reprezentują akcje bardzo płynne – Agora SA, o średniej płynności – Amica SA oraz mało płynne – WBK SA. Ze względu na złożoność obliczeń nie udało się zbadać zachowania większej populacji akcji.

Dane empiryczne użyte do trenowania sieci zostały wyodrębnione z notowań danej spółki, umieszczonych w systemie WARSET. Dla Agory SA zbiór uczący obejmował okres od 14 maja 2001 roku do 23 lipca 2002 roku. Zbiór testowy obejmował okres następnych trzech miesięcy, tzn. okres pomiędzy 24 lipca a 24 października 2002 roku. Zbiór uczący zawierał 65 000 obserwacji, natomiast zbiór testowy zawierał 19 400 obserwacji. W przypadku notowań spółki Amica SA zbiór uczący obejmował okres od 17 listopada 2001 roku do 23 lipca 2002 roku. Zbiór testowy obejmował okres następnych trzech miesięcy, tzn. pomiędzy 24 lipca a 24 października 2002 roku. Dla zbioru uczącego znaleziono 14 200 obserwacji, natomiast dla zbioru testowego 1700 obserwacji. Ostatnia spółka – WBK SA – była notowana pod tą nazwą tylko do 12 czerwca

2001 roku. W związku z tym zbiór uczący obejmował okres od 17 listopada 2000 roku do 23 maja 2001 roku, natomiast zbiór testowy obejmował okres od 24 maja do 12 czerwca 2001 roku. Dla WBK SA zbiór uczący zawierał 20 000 obserwacji, zbiór testowy natomiast1 100 obserwacji.

3.2. Zastosowane architektury sieci

Do utworzenia sieci neuronowych, obliczeń i symulacji użyte zostało środowisko SNNS utworzone na Uniwersytecie w Stuttgarcie[2]. Ponieważ zasadniczym celem pracy było zbadanie możliwości zastosowania sztucznych sieci neuronowych w procesie prognozowania szeregów czasowych o wysokiej częstotliwości, autorzy do każdego zestawu obserwacji zastosowali kilka sieci neuronowych, różniących się architekturą. W badaniach zostały wykorzystane jednokierunkowe sieci trójwarstwowe o różnych liczbach neuronów w warstwie ukrytej oraz sieci częściowo rekurencyjne również o różnych liczbach neuronów w warstwie ukrytej. Jako sieć częściowo rekurencyjna użyta została sieć Jordana, w której liczba neuronów ze sprzężeniem zwrotnym równa jest liczbie neuronów wyjściowych.

Dla notowań spółki Agora SA badano sieci jednokierunkowe, zawierające między 8 a 12 neuronów w warstwie ukrytej, 23 neurony w warstwie wejściowej oraz 3 neurony w warstwie wyjściowej. Jako dane wejściowe zastosowano informacje, dotyczące bieżącej ceny akcji, wolumenu, czasu, jaki upłynął od pierwszego notowania w danym dniu mierzonego w sekundach oraz informację o pozostałych pięciu najlepszych ofertach kupna i sprzedaży (zawierających oferowaną cenę i liczbę akcji w ofercie). Dane wyjściowe opisywały średnią cenę, wolumen i liczbę transakcji w ciągu następnego dnia.

Dla notowań spółek Amica SA oraz WBK SA użyto sieci jednokierunkowe z 15 neuronami wejściowymi, 3 neuronami wyjściowymi oraz pomiędzy 6 a 9 neuronami w warstwie ukrytej. Zakres danych wejściowych obejmował bieżącą cenę akcji, wolumen, czas, jaki upłynął od pierwszego notowania w danym dniu mierzonego w sekundach oraz informację o pozostałych trzech najlepszych ofertach kupna i sprzedaży (informacja zawierała cenę i liczbę akcji w ofercie). Dane wyjściowe opisywały średnią cenę, wolumen i liczbę transakcji w ciągu następnego dnia.

Badano również częściowo rekurencyjne sieci Jordana. W przypadku wszystkich spółek zakres danych wejściowych i wyjściowych pozostał ten sam.

[2] Stuttgart Neural Network Simulator (SNNS), (1990-1995), SNNS Group, Institute for Parallel and Distributed High-Performance Systems (IPVR), University of Stuttgart, Breitwiesenstrasse 20-22, 70565 Stuttgart, Fed. Rep. of Germany.

Dla Agory SA w warstwie ukrytej rozpatrywano pomiędzy 3 a 8 neuronami, dla pozostałych spółek pomiedzy 3 a 6 neuronami w warstwie ukrytej.

Do trenowania sieci we wszystkich przypadkach użyto algorytmu wstecznej propagacji błędu. Dla sieci jednokierunkowych zastosowano tzw. algorytm *vanilia backpropagation* (standardowy algorytm wstecznej propagacji błędu), natomiast dla sieci Jordana w algorytmie uwzględniono trzy neurony ze sprzężeniem zwrotnym. Jako funkcję aktywacji dla neuronów w warstwie ukrytej zastosowano funkcję unipolarną, natomiast dla neuronów w warstwie wejściowej i wyjściowej funkcję tożsamościową. Zastosowane funkcje aktywacji wymusiły unormowanie obserwacji do przedziału [0, 1].

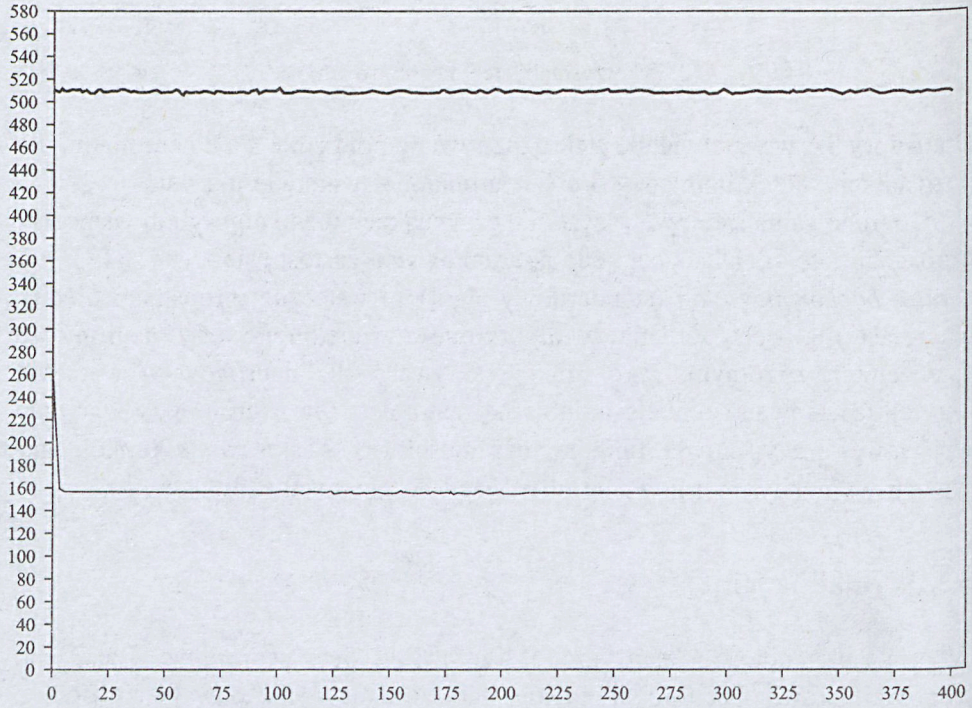
3.3. Wyniki badań

W przypadku notowań akcji spółki Agora S.A. sieci neuronowe – zarówno jednokierunkowe, jak i rekurencyjne sieci Jordana – nie dawały się dobrze nauczać. Na rysunku 4 pokazane zostały błędy, jakie wykazywała sieć jednokierunkowa o 8 neuronach w warstwie ukrytej uzyskane dla zbioru uczącego (cienka linia) i zbioru testowego (półgruba linia). Przypominamy, że błąd był sumą kwadratów różnic pomiędzy obserwowanymi wartościami a wartościami znalezionymi przez sieć neuronową dla każdego neuronu wyjściowego. Zaznaczony został na osi pionowej. Na osi poziomej zaznaczono kolejne iteracje (epoki). Jak widać, po kilku pierwszych iteracjach (około 10) sieć nie poprawiała błędu (nie uczyła się). Podobne wyniki uzyskano niezależnie od liczby neuronów znajdujących się w warstwie ukrytej, oraz niezależnie od współczynnika uczenia. Można w tym miejscu dodać, że średnio-kwadratowy błąd dla zbioru testowego był równy około 0,08, co odpowiada między 10 a 30% wartości prognozowanych.

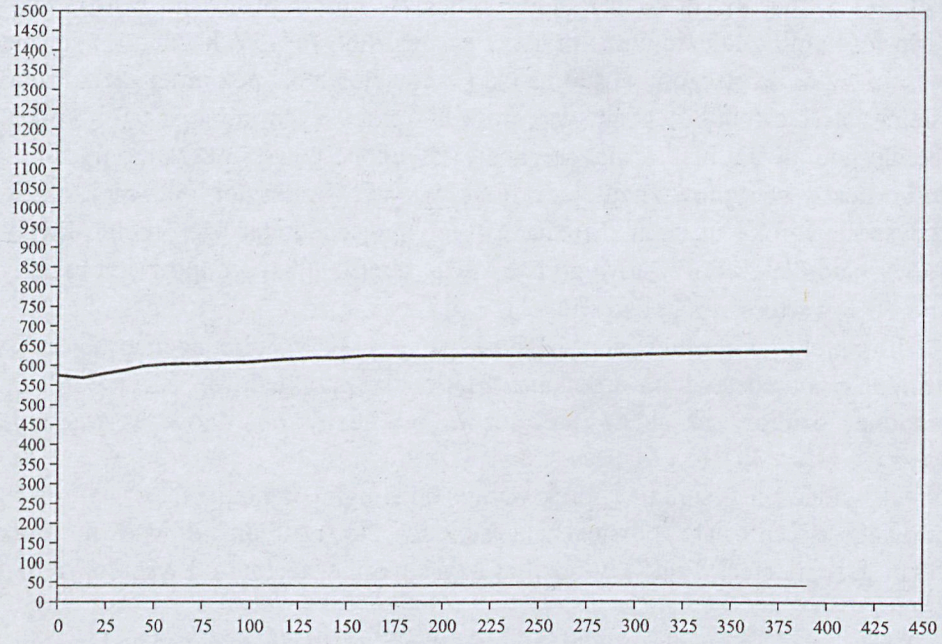
Brak efektów uczenia zanotowano również w przypadku sieci Jordana. Na rysunku 5 przedstawiono uzyskane błędy dla przykładowej sieci Jordana. Podobne rezultaty uzyskano niezależnie od liczby neuronów w warstwie ukrytej.

Jak widać na rysunku 5, praktycznie od drugiej iteracji błędy dla zbioru testowego wolno, ale powiększają się. Jest to pośredni dowód na brak efektu uczenia sieci. Podobny rezultat uzyskano niezależnie od współczynnika uczenia.

Również w przypadku spółki Amica SA nie udało się osiągnąć zadowalających rezultatów. Sieci jednokierunkowe zachowywały się bardzo podobnie do sieci trenowanych na podstawie notowań Agory SA. Zmieniła się jedynie skala

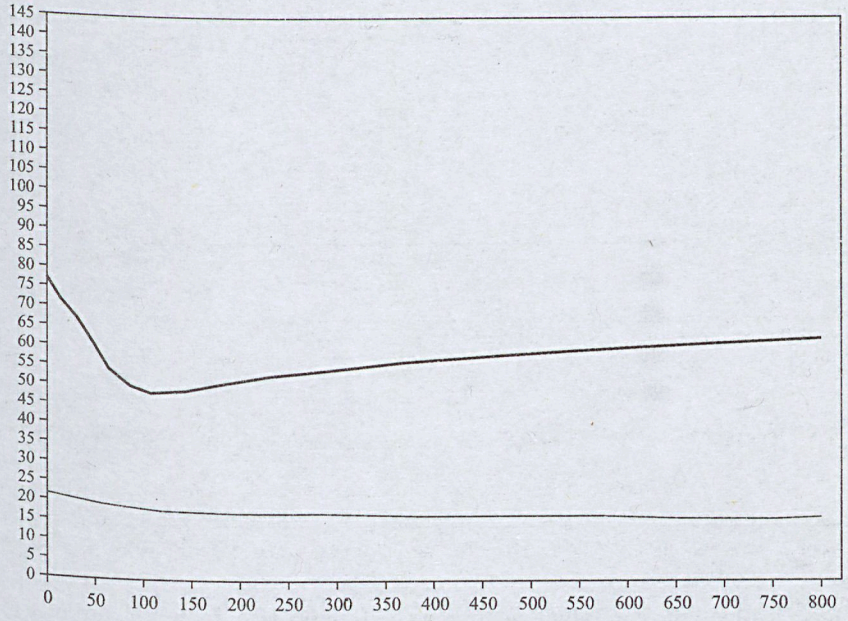


Rys. 4. Błędy przypadające na neuron wyjściowy sieci jednokierunkowej o 8 neuronach w warstwie ukrytej trenowanej na podstawie notowań spółki Agora SA



Rys. 5. Błędy przypadające na neuron wyjściowy sieci Jordana o 6 neuronach w warstwie ukrytej trenowanej na podstawie notowań spółki Agora SA

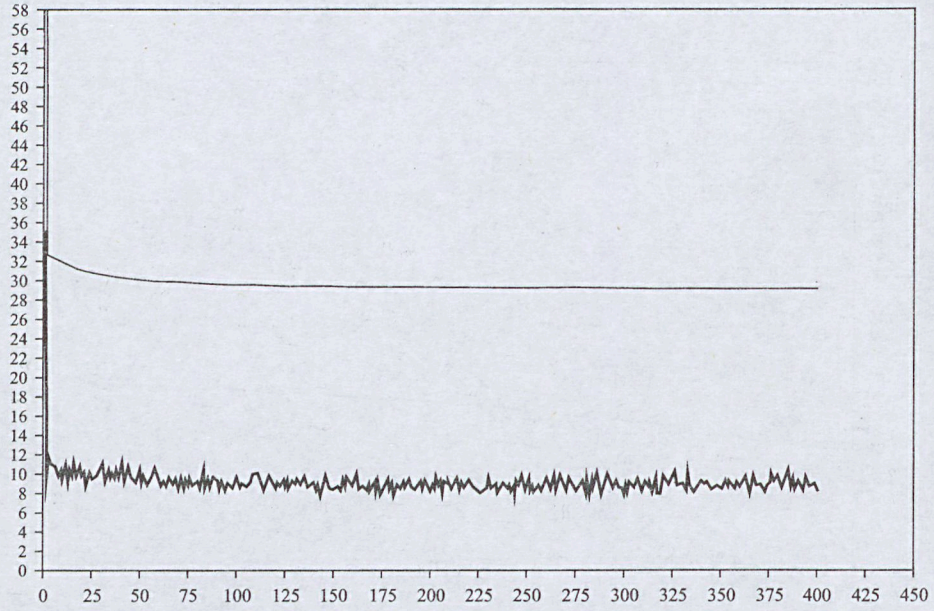
błędu, co spowodowane było zdecydowanie mniejszą liczbę obserwacji użytych do badań. Nieco lepsze rezultaty osiągnięto dla sieci Jordana. Na rysunku 6 widać, że błędy dla zbioru testowego początkowo maleją, osiągając minimum po około 100 iteracjach i następnie zaczynają rosnąć. Chociaż błędy dla zbioru uczącego nadal maleją, należało przerwać uczenie sieci w momencie, gdy sieć osiągnęła najmniejszy błąd dla zbioru testowego. W tym przypadku można stwierdzić, że sieć została, w pewnym stopniu, wytrenowana. Dla innych sieci Jordana (o mniejszej liczbie neuronów w warstwie ukrytej) zachowanie sieci było podobne. Zwiększała się jedynie liczba iteracji niezbędnych do wytrenowania sieci (około 350 dla trzech neuronów w warstwie ukrytej).



Rys. 6. Błędy przypadające na neuron wyjściowy sieci Jordana o 6 neuronach w warstwie ukrytej trenowanej na podstawie notowań spółki Amica SA

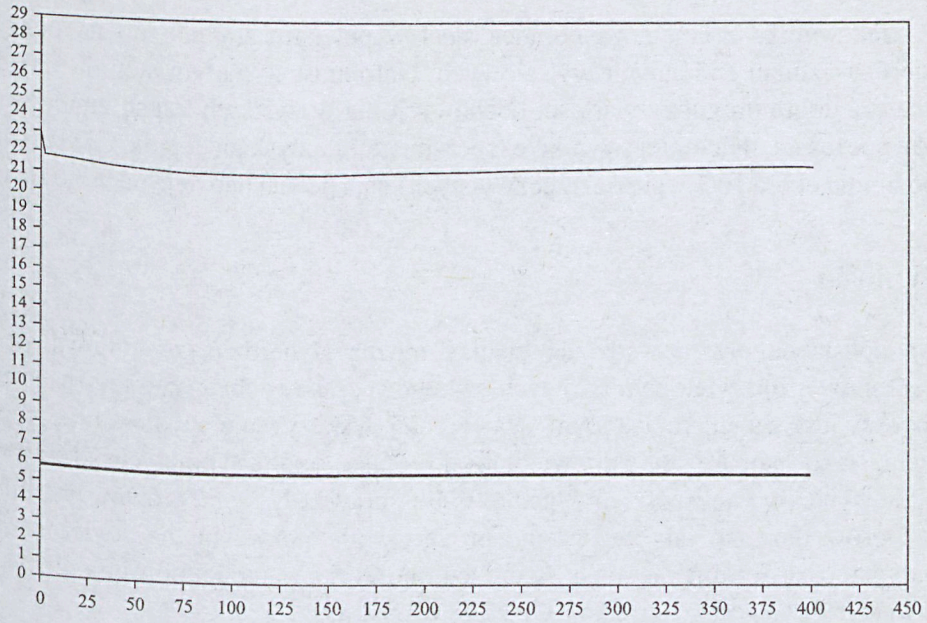
Najlepsze efekty trenowania sieci uzyskano dla szeregów notowań spółki WBK SA. Rysunek 7 przedstawia błędy uzyskane dla sieci jednokierunkowej o 8 neuronach w warstwie ukrytej. Jak widać, błędy dla zbioru testowego są mniejsze niż uzyskane dla zbioru uczącego. Sieć początkowo uczy się, zmniejszając błąd, i po około 90 iteracjach można stwierdzić, że sieć została dobrze wytrenowana. Dalsze uczenie co prawda nie poprawia w znaczącym

stopniu błędu, ale sieć zachowuje zdolność uogólniania. Wskazują na to niepowiększające się błędy zbioru testowego. Uzyskany rezultat sugeruje zatem, że sieć jednokierunkowa może być dobrym narzędziem prognozy dla tego szeregu czasowego. Podobne rezultaty uzyskano niezależnie od liczby neuronów w warstwie ukrytej. Dla sieci o 6 neuronach w warstwie ukrytej RMSE był równy 0,08, co stanowiło około 10% wartości zmiennych prognozowanych, natomiast dla sieci o 8 neuronach w warstwie ukrytej RMSE osiągnął poziom 0,068.



Rys. 7. Błędy przypadające na neuron wyjściowy sieci jednokierunkowej o 8 neuronach w warstwie ukrytej trenowanej na podstawie notowań spółki WBK SA

Również w przypadku sieci Jordana trenowanej na podstawie szeregów notowań WBK SA można zauważyć, że błędy dla zbioru testowego są mniejsze niż błędy dla zbioru uczącego. Niestety, jak widać na rysunku 8, błędy dla zbioru testowego nie wykazują tendencji malejącej. Ponieważ błędy dla zbioru uczącego maleją do pewnego minimum i następnie, po około 150 iteracjach, rosną, należy przerwać trenowanie sieci w tym momencie. Można stwierdzić, że sieć została w pewnym stopniu wytrenowana. Aby sprawdzić jakość uzys-



Rys. 8. Błędy przypadające na neuron wyjściowy sieci Jordana o 3 neuronach w warstwie ukrytej trenowanej dla notowań spółki WBK SA

kanych prognoz, porównano prognozy wygenerowane przez sieć z wartościami rzeczywistymi dla kilku wybranych dni. Wyniki zostały przedstawione w poniższej tabeli.

Zestawienie wartości rzeczywistych i prognoz na podstawie sieci Jordana o 3 neuronach w warstwie ukrytej

Data (orientacyjna)	Wartości rzeczywiste w następnym notowaniu			Prognozy		
	cena	wolumen	liczba transakcji	cena	wolumen	liczba transakcji
17.11.2000	259,45	5599	89	271,26	2359	65
10.01.2001	257,26	2759	54	269,28	2490	62
24.05.2001	272,44	2235	16	270,89	2308	65
12.06.2001	260,35	711	26	267,88	2813	67

Jak wynika z tabeli, za pomocą sieci w pewnym stopniu można odkryć kierunek zmian zmiennych wyjściowych. Natomiast w małym stopniu można znaleźć dobre prognozy wartości liczbowych dla wszystkich trzech zmiennych równocześnie. Niemniej wyniki eksperymentu i uzyskane błędy (RMSE na poziomie około 10% wartości rzeczywistych) dają pewną nadzieję na przyszłość.

Wnioski

Na podstawie przeprowadzonej analizy można stwierdzić, że budowa sieci neuronowej dla wielowymiarowych szeregów czasowych o wysokiej częstotliwości jest zadaniem trudnym. Najczęściej wskazywane w literaturze typy sieci, jako najbardziej odpowiednie do opisu ogólnie pojętych szeregów czasowych, w większości przypadków nie sprawdziły się. Zarówno przebieg procesów uczenia, jak i uzyskane prognozy nie pozwalają na stwierdzenie przydatności wykorzystanych sieci neuronowych do podejmowania decyzji inwestycyjnych na rynku akcji GPW w Warszawie.

Literatura

[1] Barnett W.A., Kirman A.P., Salmon M. (eds.), *Nonlinear Dynamics and Economics*, Cambridge University Press, Cambridge 1996

[2] Basii D., *Stock Price Predictions by Recurrent Multi-layer Neural Network Architectures*, Neural Networks in Financial Engineering 1996, 2, 331-340

[3] Baum E., Haussler D., *What Size Net Gives Valid Generalization*, Neural, Computation 1998, 1

[4] Dacogorana M.M., Gencay R., Müller U.A., Olsen R.B., Pictet O.V., *An Introduction to High Frequency Finance*, Harcourt's Academic Press, 2000

[5] Engle F. R., *Autoregessive Conditional Heteroscedascity with Estimates of the Variance of United Kingdom Inflation*, Econometrica 50, 1982, 4, 997-1007

[6] Garsztka P., Matuszewski, P., Wieloch K., *Analiza płynności papierów wartościowych notowanych w systemie WARSET*, red. W. Milo, Acta Universitatis Lodziensis, Folia Oeconomica 2003, 166, 225-239

[7] Gatley E., *Sieci neuronowe. Prognozowanie finansowe i projektowanie systemów transakcyjnych*, Wig-Press, Warszawa 1999

[8] Haefke C., Helmenstein C., *Predicting Stock Market Averages to Anhance Profitable Trading Strategies*, Neural Networks in Financial Engineering 1997, 2, 378-289

[9] Hiemstra Y., *Linear Regression versus Backpropagation Networks to Predict Quarterly Stock Market Excess Returns*, Computational Economics 1996, 9, 67-76

[10] Hong P., Rognvaldsson T., *A Neural Network Approach to Futures Trading*, Neural Networks in Financial Engineering, 1995, 2, 17-25
[11] Lütkepohl H., *Introduction to Multiple Time Series Analysis*, Springer-Verlag, Heidelberg 1991
[12] Masters T., *Sieci neuronowe w praktyce*, Wydawnictwo Naukowe PWN, Warszawa 1996
[13] Peters E.E., *Teoria chaosu a rynki kapitałowe*, Wig-Press, Warszawa 1997
[14] Rachev S., Mittnik S., *Stable Paretian Models in Finance*, Wiley, New York 2000
[15] Siriopoulos G., et. al., *Applications of Artificial Neural Networks in Emerging Financial Markets*, Neural Networks in Financial Engineering 1995, 2, 284-302
[16] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa 1993
[17] Tarczyński W., *Fundamentalny portfel papierów wartościowych*, PWE, Warszawa 2002
[18] The Intertek Group, *Real-Time Data, Analytics and Forecasting Systems*, The Intertek Groups, Paris 1997
[19] The Swedish Neural Network Society, http://www.nada.kth.se/sans/neuronet/SNNS.html
[20] Tsibouris G., Zeidenberg M., *Testing the Efficient Market Hypothesis with Gradient Descent Algorithms*, w: *Neural Networks in the Capital Markets*, ed. A.P. References, Wiley, NewYork 1995
[21] Witkowska D., *Sztuczne sieci neuronowe i metody statystyczne*, Wydawnictwo C.H. Beck, Warszawa 2002
[22] Wong F.S., *Time Series Forecasting Using Backpropagation Neural Networks*, Neurocomputing 1990, 2, 147-159
[23] Żurada J., Barski M., Jędruch W., *Sztuczne sieci neuronowe. Podstawy teorii i zastosowania*, Wydawnictwo Naukowe PWN, Warszawa 1996

APPLICATION OF NEURAL NETWORKS TO PREDICTION OF FINANCIAL TIME – SERIES OF HIGH FREQUENCY OF STOCK LISTED ON WARSAW STOCK EXCHANGE

Summary

In this paper there is undertaken an attempt at prediction of stock prices using ANN (Artificial Neural Network) on the basis of high frequency time – series from Warsaw Stock Exchange. ANN has been used to arrive at forecast of size, value, and number of transactions during next day. The most popular types of ANN have been used in the paper.