

Przewidywanie kursów giełdowych przy pomocy sztucznych sieci neuronowych

Piotr Bródka

January 22, 2019

1 Przedstawienie problemu

Przewidywanie kursów giełdowych jest tematem zainteresowania inwestorów od długiego czasu. Sądzi się, że umiejętność grania na giełdzie można wyćwiczyć oraz, że istnieją wzorce, na podstawie których można przewidzieć przyszłe notowania (analiza techniczna). Skoro można wykorzystać do tego zadania mózg człowieka, to możliwe, że sztuczne sieci neuronowe także się do tego nadają.

Za przewidywanie kursu rozumiem przewidywanie kursu zamknięcia na kolejny dzień na podstawie kursów zamknięcia z poprzednich dni (częstotliwość danych: 1/dzień). Są dwa możliwe podejścia:

1. regresja - czyli przewidujemy konkretną wartość kursu zamknięcia,
2. klasyfikacja - przewidujemy spadek/wzrost kursu zamknięcia

W wielu pracach, dotyczących tematu, analizuje się pierwsze podejście, natomiast mnie będzie interesowała klasyfikacja. Jest to podyktowane względami praktycznymi. Gdy gramy na giełdzie, interesuje nas czy spadnie, czy wzrośnie. Można powiedzieć, że w momencie, gdy dokładność naszej predykcji przekroczy 50%, to „jesteśmy milionerami”. Oczywiście, to zdanie brzmi bardzo dobrze, ale są w nim dwa podstawowe przekłamania. Działa to pod warunkiem, że predykcja jest stabilna. Po drugie, należałoby także uwzględnić prowizję za dokonywanie transakcji.

1.1 Przegląd literatury

Istnieją źródła naukowe, omawiające ten temat. Niestety, nie ma ich dużo. Jest bardzo prawdopodobne, że udane eksperymenty przewidywania giełdy są rzadko publikowane, gdyż autorzy wolą trzymać je w tajemnicy. Przykładem artykułu naukowego, który opisuje temat jest [1].

Na blogach, dotyczących Data Science i Uczenia Maszynowego można znaleźć wiele artykułów na temat przewidywania kursów, głównie kryptowalut, ale też giełd tradycyjnych. Zdecydowana większość artykułów wskazuje na możliwość użycia sieci rekurencyjnych, których naturalnym zastosowaniem jest przewidywanie szeregów czasowych (w szczególności sieci LSTM).

2 Założenia wstępne

Będę korzystał z historii notowań indeksu WIG20 - indeksu giełdowego 20 największych spółek akcyjnych notowanych na warszawskiej Giełdzie Papierów Wartościowych. Jego historia sięga roku 1991. Poniższy wykres przedstawia historię zmian indeksu:

Do przewidywania wzrostów (i spadków) użyję dwóch typów sieci neuronowych:

1. Perceptron wielowarstwowy
2. LSTM

Zbiór danych liczy 6464 elementy (kursy z 6464 dni).

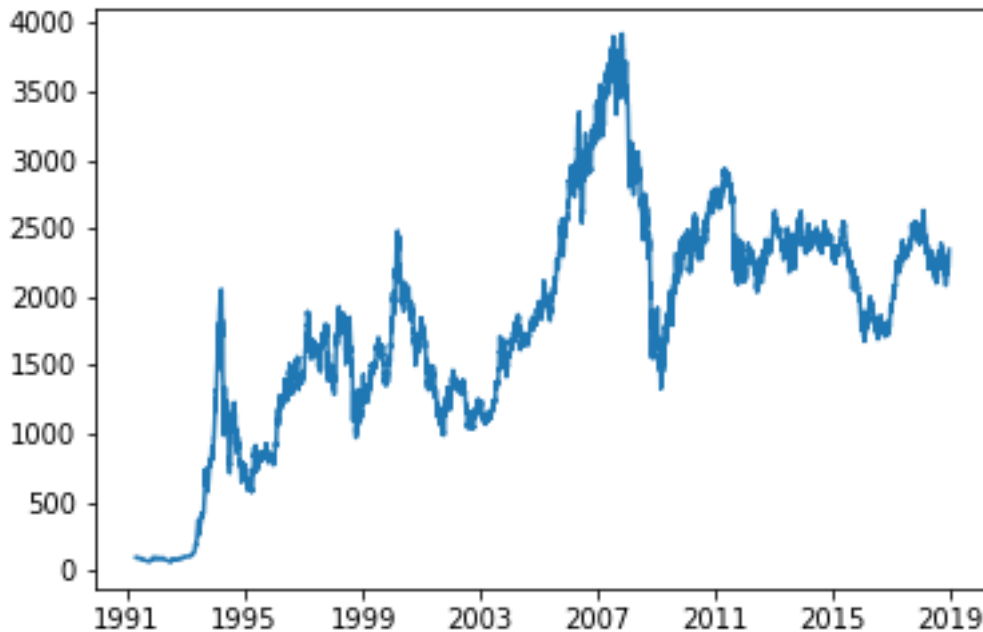


Figure 1: Historia zmian kursu WIG20

3 Przygotowanie danych

Na wejściu mamy t elementów, wartości kursów z danych dni. Ponieważ będę przewidywał szeregi czasowe, muszę przetransformować dane do postaci:

Dane na podstawie których będę przewidywał:

$t_1 \ t_2 \ \dots \ t_k$
 $t_2 \ t_3 \ \dots \ t_{k+1}$
 $\dots\dots\dots$

Gdzie k jest długością szeregu, dla którego przewidyuję.

Dane, które będę przewidywał.

t_{k+1}
 t_{k+1+1}
 t_{k+1+2}
 $\dots\dots$

Gdzie l to ilość dni, na które patrzymy w przyszłość. U mnie ustalam l jako 1.

Na wykresie widać, że wielkości kursów w różnych okresach różnią się znacznie od siebie. W takim wypadku potrzebna jest regularyzacja danych. Są dwa możliwe sposoby regularyzacji:

1. Dane będą reprezentowały procentowy wzrost/spadek w stosunku do dnia poprzedniego:

$$val_t = \frac{(val_t - val_{t-1})}{val_{t-1}}$$

Ten sposób regularyzacji wykorzystam przy klasyfikacji.

2. Dane będą stosunkiem kursu zamknięcia z danego dnia do początkowego kursu zamknięcia

$$val_t = \frac{(val_t - val_0)}{val_0}$$

Ten sposób regularyzacji wykorzystam przy klasyfikacji.

Dodatkowo - z danych usunąłem wszystkie rekordy przed 1994 rokiem jako, że fakt początku funkcjonowania giełdy mógłby popsuć wyniki.

4 Perceptron

Sieć feed-forward. Zakładając, że obecnie mamy dzień n i chcemy przewidzieć ruch indeksu na dzień kolejny na podstawie k poprzednich dni, to do warstwy wejściowej wprowadzane są indeksy giełdy z dni od $n-k+1$ do n . Na wyjściu mamy:

1. w przypadku regresji - wartość przewidywana kursu,
2. w przypadku klasyfikacji - wyjście binarne: 1 - jeśli przewidywany jest wzrost kursu, 0 - jeśli przewidywane jest utrzymanie bądź spadek kursu.

4.1 Parametry i hiperparametry

W przypadku tej sieci jest wiele parametrów / hiperparametrów, od których może zależeć jakość predykcji:

1. Ilość warstw ukrytych
2. Wielkość warstw ukrytych
3. Na jak długim zakresie danych przewidujemy (na danych z ilu dni wstecz)?
4. Jak daleko przewidujemy? Ustaliłem, że przewidujemy na następny dzień, ale nie musi być to z góry nauczane.
5. Czy posługujemy się regresją czy klasyfikacją?

4.2 Plan eksperymentów

Przypomnę, że chodzi mi o klasyfikację (czyli, czy wartość kursu wzrośnie czy spadnie). Można ją zrealizować na dwa sposoby:

1. za pomocą regresji - To znaczy przewidzieć wartość kursu i na tej podstawie policzyć, czy kurs wzrósł czy spadł
2. za pomocą klasyfikacji - Sieć na wejściu dostanie procentowe wzrosty i spadki. Na tej podstawie wyznaczy, czy będzie wzrost czy spadek w przyszłości.

Eksperymenty będą zakładały sprawdzenie, czy architektura sieci jest optymalna (czy sieć nie jest za mała albo czy nie jest zbyt duża).

Postawię hipotezę, że zmiany giełdy są lokalne. To znaczy, że ucząc się na pewnym fragmencie notowań, możemy uzyskać złe wyniki, testując model na innym fragmencie. Może się to wiązać ze zmianą warunków makroekonomicznych, których nie jesteśmy w stanie przewidzieć. Dlatego zastosuję stworzone modele do uczenia się, walidacji i przewidywania - ale na krótszych okresach (10% całości okresu).

5 LSTM

Wiele źródeł (jak na przykład [5]) wskazuje na dobre efekty użycia rekurencyjnych sieci neuronowych w celu przewidywania szeregów czasowych. Rekurencyjną sieć neuronową (RNN) można interpretować, jako wiele instancji tej samej sieci, komunikujących się ze sobą w czasie. Dzięki temu, możemy powiedzieć, że sieć ma pamięć. Częstym problemem tradycyjnych RNN jest tzn. zanikający gradient. Problem ten stara się rozwiązać sieć LSTM (Long Short Term Memory) Szczegółowy opis, jak działa LSTM można znaleźć na przykład tutaj: [8]

5.1 Plan eksperymentów

Wykonam eksperyment dla regresji (wejściowy ciąg: wzrosty i spadki w stosunku do dnia poprzedniego), wyjściem będzie spadek bądź wzrost.

Architektura sieci będzie ustalona: jedna warstwa ukryta LSTM.

Wykonam eksperyment dla różnych wielkości sekwencji wejściowej na podstawie której przewidujemy przyszłe ruchy giełdy.

6 Wyniki eksperymentu - perceptron

Na początku eksperymenty uruchamiałem dla całego zbioru. Za pomocą regresji i klasyfikacji. Sprawdziłem różne konfiguracje: 10, 5, 20 warstw ukrytych, rozmiary tych warstw: 10, 5, 20. Sprawdziałem dla różnych k - długości ciągów dni na podstawie których przewidujemy dzień kolejny.

Niestety przy wielokrotnym uruchomieniu eksperymentów (od 50 do 100) dokładność była bardzo zbliżona do 0.5. Odchylenie było rzędu 0.005. Nie załączyłem tutaj tych wyników, gdyż byłyby to bardzo dużo wartości zbliżonych do 0.5.

Zmieniłem metodykę i losowałem ciągłe fragmenty zbioru wejściowego (o długości 10% zbioru wejściowego). Oto wyniki, jakie uzyskałem dla regresji (10 warstw ukrytych, po 10 neuronów każda):

Kolejne kolumny oznaczają przewidywania na podstawie różnych długości ciągu wejściowego na podstawie którego przewidujemy: 5, 10, 15, 20, 30, 40, 50.

W kolumnie mamy najpierw wyniki pierwszych 15 eksperymentów. Ostatni element każdej kolumny to średnia ze stu eksperymentów.

acc_5	acc_10	acc_15	acc_20	acc_30	acc_40	acc_50
0.488	0.480	0.496	0.537	0.496	0.436	0.496
0.488	0.480	0.496	0.537	0.496	0.436	0.496
0.480	0.512	0.438	0.529	0.487	0.504	0.530
0.485	0.491	0.477	0.534	0.493	0.459	0.507
0.569	0.439	0.545	0.521	0.496	0.504	0.487
0.502	0.480	0.490	0.532	0.494	0.468	0.503
0.577	0.537	0.529	0.471	0.403	0.538	0.487
0.513	0.488	0.496	0.523	0.481	0.478	0.501
0.496	0.610	0.488	0.479	0.496	0.547	0.470
0.511	0.502	0.495	0.518	0.482	0.486	0.497
0.545	0.593	0.512	0.413	0.538	0.590	0.530
0.514	0.510	0.497	0.509	0.487	0.495	0.500
0.488	0.488	0.554	0.463	0.504	0.530	0.391
0.512	0.508	0.501	0.505	0.489	0.498	0.492
0.537	0.480	0.479	0.512	0.462	0.487	0.496
0.514	0.506	0.499	0.506	0.487	0.497	0.492
0.472	0.520	0.488	0.504	0.462	0.487	0.539
0.511	0.507	0.499	0.506	0.485	0.496	0.495
0.480	0.463	0.504	0.545	0.496	0.556	0.487
0.509	0.507	0.499	0.508	0.486	0.498	0.495

Warto zauważyć, że po raz pierwszy widzimy wartości większe od połowy (dla rozsądnej ilości eksperymentów po której można uśrednić), większe od połowy o prawie jeden punkt procentowy.

W przypadku badania za pomocą regresji niestety nie otrzymałem lepszych wyników.

Na obrazku poniżej możemy zobaczyć wynik dla regresji przy następującej konfiguracji: 10 warstw ukrytych, po 10 neuronów każda, uczymy, walidujemy i testujemy na całym zbiorze. Na wykresie widzimy przewidywania dla zbioru testowego (czyli końca całego okresu):

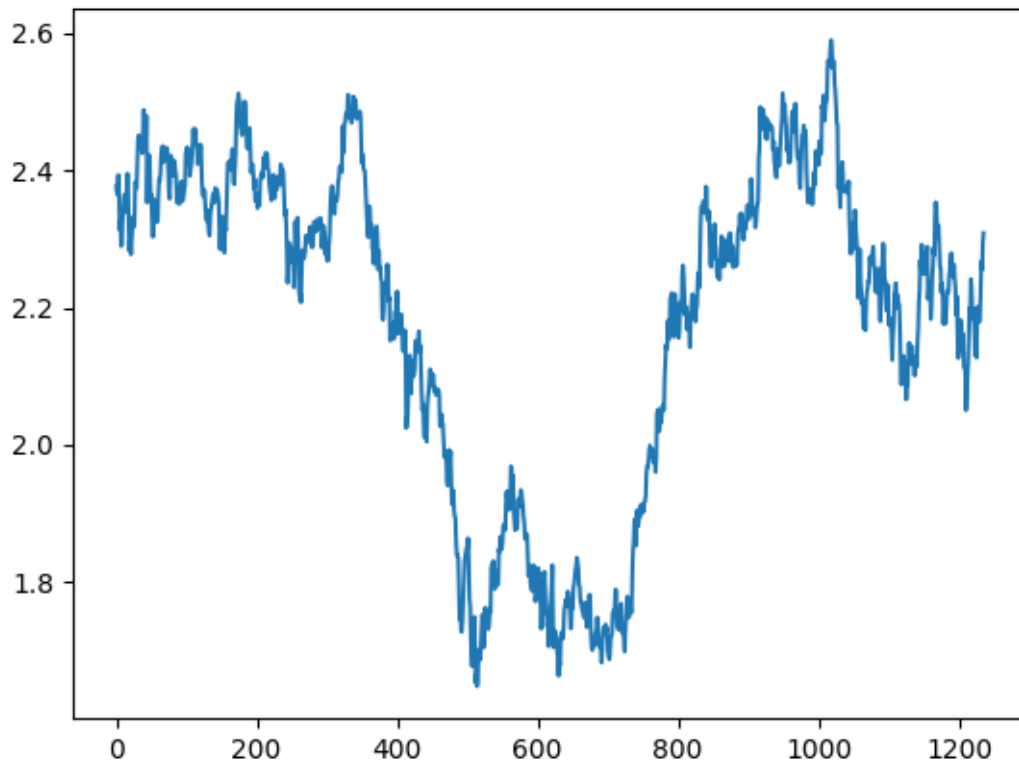


Figure 2: Przewidywane zmiany kursu za pomocą regresji

Widać, że odwzorowanie jest bardzo dokładne. Ale nie powinno nas to dziwić, gdyż przewidując na podstawie k poprzednich dni kolejny dzień, wartość kolejnego dnia będzie zbliżona do wartości z poprzednich dni. Nie znaczy to oczywiście, że jesteśmy w stanie przewidzieć wzrosty i spadki.

7 Wyniki eksperymentu - LSTM

Podobnie jak w poprzednim eksperymencie, przedstawiam pierwsze 15 wyników oraz średnią stu wyników w ostatnim rzędzie.

acc_10	acc_20	acc_30
0.520325	0.561983	0.529412
0.471545	0.531680	0.521008
0.455285	0.487603	0.546218
0.468293	0.522865	0.526050
0.455285	0.512397	0.478992
0.479675	0.438017	0.571429
0.467906	0.512108	0.525117
0.455285	0.495868	0.546218
0.466758	0.510632	0.527035
0.528455	0.537190	0.470588
0.471504	0.512675	0.522693
0.495935	0.454545	0.529605
0.473133	0.508799	0.522020
0.528455	0.512397	0.462185
0.536585	0.545455	0.512605
0.489555	0.510929	0.518190

Zauważmy, że uzyskaliśmy wyniki o 1 punkt procentowy większe niż połowa. Jest to obiecujące. Niestety uczenie sieci LSTM trwa długo i nie ma wyników dla k większego niż 30.

8 Wnioski

Perceptron przy użyciu regresji radzi sobie dobrze z przewidywaniem ogólnego trendu na rynku. Natomiast wyniki przewidywania spadku/wzrostu są niezbyt zadowalające. Z drugiej strony, nie spodziewałem się, że będą. Stwierdzam, że perceptron wielowarstwowy nie nadaje się do modelowania szeregów czasowych. Jego podstawową wadą w tym zakresie jest to, że traktuje dane z poprzednich dni po prostu jako pewien wektor danych wejściowych. "Nie ma świadomości", że dane przychodzą do niego jako sekwencja.

Wydaje się, że obiecującym kierunkiem do eksploracji są sieci rekurencyjne. Nie zmienia to faktu, że przewidywanie ruchów giełdy jest zadaniem bardzo trudnym. Obiecujące mogłoby być przewidywanie nie tylko na podstawie szeregu czasowego, ale także na podstawie innych czynników, takich jak na przykład analiza sentymentu.

References

- [1] Yue-Gang Song, Yu-Long Zhou, Ren-Jie Han, *Neural networks for stock price prediction*
<https://arxiv.org/abs/1805.11317>
- [2] Orhan Gazi Yalçın, *Predict Tomorrow's Bitcoin (BTC) Price with Recurrent Neural Networks*
<https://towardsdatascience.com/using-recurrent-neural-networks-to-predict-bitcoin-btc-prices-c>
- [3] Devadoss, Ligor, *Forecasting of Stock Prices Using Multi Layer Perceptron (International Journal of Computing Algorithm)*
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.679.2370rep=rep1type=pdf>
- [4] Martin Längkvist, *A review of unsupervised feature learning and deep learning for time-series modeling*
<https://www.sciencedirect.com/science/article/pii/S0167865514000221>
- [5] Vivek Palaniappan, *Neural Networks to Predict the Market*
<https://towardsdatascience.com/neural-networks-to-predict-the-market-c4861b649371>
- [6] Siraj Raval, *Stock Market Prediction*
<https://www.youtube.com/watch?v=JuLCL3wCEAk>
- [7] Yusuf Aktan, *Using NLP and Deep Learning to Predict Stock Price Movements*
<https://towardsdatascience.com/using-nlp-and-deep-learning-to-predict-the-stock-market-64eb922>
- [8] Christopher Olah, *Understanding LSTM Networks*
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>