

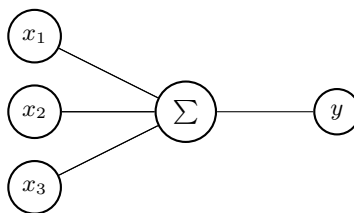
Sieci Neuronowe w Pythonie: wprowadzenie, struktura i zastosowanie w strategii tradingowej

Karol Gutkowski
Damian Kąkol
Jakub Kępka

Marzec 2024

Spis treści

1	Sieci neuronowe	2
1.1	Rodzaje sieci neuronowych	2
1.1.1	Perceptron	2
1.1.2	Wielowarstwowe perceptrony	2
1.1.3	Sieci neuronowe typu feed forward	3
1.1.4	Rekurencyjne sieci neuronowe	3
1.1.5	Konwolucyjne sieci neuronowe	4
2	Teoria rynków oraz przegląd badań	4
2.1	Przegląd metaanaliz	4
2.2	Wnioski	5
3	Trenowanie sieci	5
3.1	Spadek gradientu (Gradient Descent)	5
3.2	Propagacja wsteczna błędów (Backpropagation)	6
4	Zastosowanie w tradingu	6
4.1	Wybór rodzaju sieci	6
4.2	Źródło danych	7
4.3	Output modelu	8
4.4	Przygotowanie i analiza danych	8
4.5	Przygotowanie i trenowanie modelu	8
4.6	Porównanie modelu z rynkiem	8



Rysunek 1: Przykład perceptronu o trzech wejściach x_1 , x_2 , x_3

1 Sieci neuronowe

1.1 Rodzaje sieci neuronowych

1.1.1 Perceptron

Jest to prosty element obliczeniowy sumujący sygnały wejściowe z warstwy x z odpowiednimi wagami w dla każdego wejścia. W zależności od wyniku perceptron uznajemy za wzbudzony (stan 1) lub nie (stan 0).

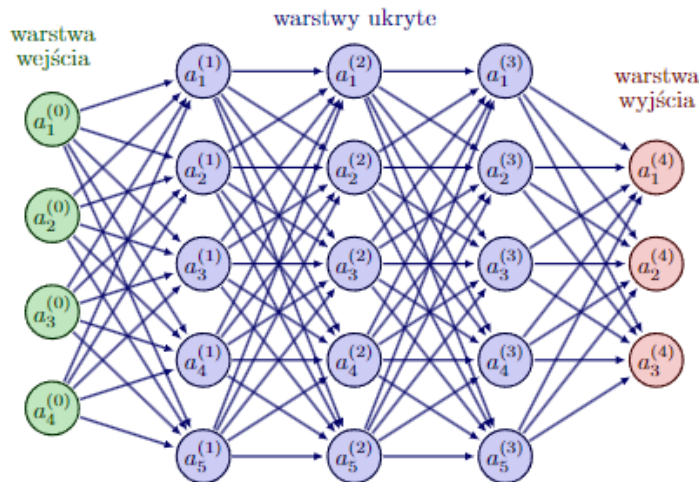
W uogólnieniu perceptron jest funkcją matematyczną obliczającą ważoną sumę wartości wejścia i dającą wyjścia w zależności od funkcji aktywacji.

$$y = \sum_{i=1}^n x_i w_i \quad (1)$$

Perceptron przez swoją prostotę nie ma zaawansowanych zastosowań ale jego idea pozwala nam na wprowadzenie bardziej skomplikowanych sieci neuronowych.

1.1.2 Wielowarstwowe perceptrony

Modelem budującym na bazie pojedynczego perceptronu jest model wielowarstwowego perceptronu. Ten model pozwala na stworzenie bardziej skomplikowanych relacji między warstwą wejściową, a wyjściową poprzez stworzenie kilku dodatkowych warstw (tzw. ukrytych z ang. hidden layers) na drodze wyliczenia wyniku w warstwie wynikowej.



Rysunek 2: Schemat wielowarstwowego perceptronu [1]

Na powyższym obrazku widać schematyczny rysunek wielowarstwowego perceptronu. Tutaj mamy 3 warstwy ukryte i warstwę wyjścia składającą się z 3 neuronów (jednak w wielu przypadkach wyjście z ostatniej warstwy mającej wiele neuronów może być zredukowane to jednego wyniku np. wybór najbardziej aktywnego ze wszystkich).

Schemat działania pojedynczych neuronów jest analogiczny jak w przypadku perceptronu. Sumują one wejścia z poprzednich warstw wraz z odpowiadającymi im wagami oraz tzw. bias-em i wynik ostateczny jest wprowadzany do funkcji aktywacyjnej neuronu (np. sigmoid albo ReLU).

1.1.3 Sieci neuronowe typu feed forward

Sieci neuronowe typu feed forward to obok sieci rekurencyjnych, jeden z najistotniejszych rodzajów sieci neuronowych. Sieci tego typu wyróżnia ukierunkowany przepływ danych tj.: od węzłów wejściowych, poprzez ukryte aż do wyjścia. W sieci nie występują żadne cykle w przepływie danych. Do uczenia tej sieci wykorzystuje się metodę propagacji wstecznej

1.1.4 Rekurencyjne sieci neuronowe

Rekurencyjne sieci neuronowe zostały zdefiniowane w kontraście do sieci typu feed forward jako te, które umożliwiają dwukierunkowy przepływ informacji. W strukturze takiej sieci znajdują się bowiem węzły, których wyjście może wpływać na późniejsze wejście tych samych węzłów. Rekurencyjne sieci znajdują zastosowanie w takich dziedzinach jak rozpoznawanie mowy lub pisma ręcznego. Sieci te dobrze radzą sobie w procesowaniu języków ponieważ są w stanie uwzględniać kontekst z poprzednich kroków czasowych.

1.1.5 Konwolucyjne sieci neuronowe

Konwolucyjna sieć neuronowa to sieć typu feed forward. Została stworzona do procesowania danych w postaci tabelarycznej (np. zdjęcia). Klasycznie taka sieć składa się z warst konwolucyjnych, poolingowych, spłaszczających, odrzucenia, gęstych, oraz funkcji aktywacji. Warstwa konwolucyjna korzysta z operacji konwolucji, która w uproszczeniu to operacja stosowana na dwóch funkcjach, która w wyniku daje 3 funkcję. W przypadku zdjęć (danych 2 - wymiarowych) operacja konwolucji jest reprezentowana przez filtr macierzowy, który jest nakładanych na dane w warstwach konwolucyjnych. Warstwy poolingowe służą do zmniejszania rozmiarów modelu oraz ekstrakcję cech, realizowane są poprzez wyciąganie maksimum lub średnich z sąsiadujących pikselów. Funkcje aktywacji dodają modelowi porządkanej nieliniowości. Warstwa spłaszczająca redukuje wymiar wektora danych i w rezultacie otrzymujemy zwykłą sieć neuronową, która korzysta z warstw gęstych do wyznaczenia outputu. Aby zachować odpowiednią generalizację modelu stosuje się również warstwy odrzucenia, które przeciwdziałają overfittingowi poprzez losowe ignorowanie pewnych neuronów.

2 Teoria rynków oraz przegląd badań

W tym rozdziale przyjrzymy się teorii rynków, zrobimy przegląd metaanaliz skuteczności działań algorytmów sztucznej inteligencji i w szczególności sieci neuronowych. Postaramy odpowiedzieć się na pytanie w jakich sytuacjach, jakie algorytmy i do jakiego stopnia sieci neuronowe mają zastosowania w strategiach tradingowych. To czy modele sieci neuronowych mogą mieć zastosowanie w trading'u nie jest wcale oczywiste. Wydaje się że ich skuteczność wymaga istnienia praw rynku. Co nie jest oczywiste, jest i było to kwestią pewnych kontrowersji nawet już w XIX wieku¹. Kolejnym problemem są nieprzewidywalne, mało prawdopodobne zdarzenia z ogromnymi konsekwencjami (przykładowo wielkie kryzysy lub wojny). Czy modele są w stanie przewidzieć takie zjawiska? Jeżeli nie to czy mogą się przed tym jakoś zabezpieczyć? Czy w danych historycznych, wykresach zawarta jest informacja na temat przyszłości cen akcji? Czy możliwa jest analiza techniczna? Wiele z tych pytań nadal jest kwestią wielu sporów, jednak my postaramy odwołując się do meta analiz przynajmniej odrzucić tezy i rozwiązania które z dużym prawdopodobieństwem zdają się być fałszywe albo potencjalnie bardzo szkodliwe w zastosowaniu praktycznym.

2.1 Przegląd metaanaliz

Metaanaliza z 2021 roku robię przegląd 148 badań skupiających się na ocenie skuteczności sieci neuronowych w przewidywaniu cen na rynku. Rynek charakteryzuje się nieliniowością, nagłymi zmianami oraz niepewnym środowiskiem. Autorzy artykułu twierdzą, że sztuczna inteligencja radzi sobie dobrze z takimi

¹Miedzy innymi John Stuart Mill próbował odpowiedzieć na to pytanie.

charakterystykami rynku. W dodatku twierdzą że skuteczność AI może obalać lub stawiać opór teorii efektywnych rynków. Najlepszymi modelami okazują się sieci neuronowe oraz sieci hybrydowe, jednak nie istnieją określone struktury i sposobu trenowania sieci, wymagają one uwzględnienia kontekstu oraz metody prób i błędów.²

Następne badanie stwierdza, że fundusze korzystające z rozwiązań AI oraz ML mają statystycznie większe zyski od funduszy niekorzystających z takich rozwiązań. W dodatku taka zależność wydaje się być trwała.³

Kolejne badanie z 2021 przeprowadzone na 150 firmach wskazuje na pozytywny wpływ algorytmów AI na działanie firmy. Jednak w tym badaniu większa część AI była używana do redukcji błędów, zwiększenia użyteczności audytów i oszczędzania kosztów szkolenia nowych pracowników. Zatem te badania nie odpowiadają na żadne nasze pytanie, mogą jednak stanowić ciekawostkę.⁴

2.2 Wnioski

Biorąc pod uwagę powyższe badania, możemy stwierdzić że nieliniowość zjawisk na rynku jest dobrym argumentem na rzecz skuteczności modeli AI używanych do trading'u. Z rozwiązań z dziedziny AI sztuczne sieci neuronowa (również Hybrid neural network) mają wyróżniającą się skuteczność. Sieci neuronowe wykorzystywane są efektywnie przez fundusze trading'owe, więc już znajdują praktyczne zastosowania które można śledzić i wykorzystać w naszych modelach. Inne pytania wymagają dalszych badań.

3 Trenowanie sieci

Dzięki połączeniu gradient descent i backpropagation sieci neuronowe są w stanie uczyć się, poprawiając swoje predykcje na podstawie dostępnych danych treningowych.

3.1 Spadek gradientu (Gradient Descent)

Gradient descent to metoda optymalizacji używana do minimalizacji funkcji kosztu w procesie uczenia maszynowego. W kontekście sieci neuronowych, funk-

²Chopra, R.; Sharma, G.D. Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. *J. Risk Financial Manag.* 2021, 14, 526. <https://doi.org/10.3390/jrfm14110526>

³Niang, Joachim Amath Fabian Artificial intelligence and hedge fund performance : An analysis of hedge fund trading styles (2021), <https://osuva.uwasa.fi/handle/10024/12786>

⁴Berdiyeva, Oguljan Islam, Muhammad Saeedi, Mitra. (2021). Artificial Intelligence in Accounting and Finance: Meta-Analysis. *International Business Review.* 3. 56-79. 10.37435/NBR21032502.

cja kosztu określa różnicę między predykcją sieci, a rzeczywistymi wartościami oczekiwanymi.

Proces ten polega na iteracyjnym aktualizowaniu parametrów sieci neuronowej w kierunku przeciwnym do gradientu funkcji kosztu. Celem jest znalezienie minimum funkcji kosztu, gdzie błąd predykcji jest minimalny.

Algorytm można opisać w następujących krokach:

- Oblicz gradient funkcji kosztu względem parametrów sieci.
- Zaktualizuj parametry sieci, przesuając się w kierunku przeciwnym do gradientu, pomnożonym przez szybkość uczenia (learning rate).
- Powtarzaj te kroki aż do zbieżności lub osiągnięcia maksymalnej liczby iteracji.

3.2 Propagacja wsteczna błędów (Backpropagation)

Backpropagation jest algorytmem wykorzystywanym do obliczania gradientów funkcji kosztu względem parametrów sieci neuronowej.

Proces ten polega na propagowaniu błędu od końcowej warstwy sieci wstecz, przez wszystkie warstwy, w celu obliczenia gradientów funkcji kosztu.

Algorytm można opisać w następujących krokach:

- Wykonaj krok do przodu, przetwarzając dane wejściowe przez sieć neuronową, aby uzyskać predykcję.
- Oblicz błąd pomiędzy predykcją a rzeczywistymi wartościami.
- Przeprowadź propagację wsteczną, obliczając gradienty funkcji kosztu za pomocą reguły łańcuchowej.
- Aktualizuj parametry sieci, wykorzystując gradient descent.

W trakcie propagacji wstecznej wykorzystuje się pochodne cząstkowe funkcji kosztu względem każdego parametru sieci, aby określić, jak bardzo dany parametr wpływa na błąd predykcji. Te pochodne są obliczane za pomocą reguły łańcuchowej.

4 Zastosowanie w tradingu

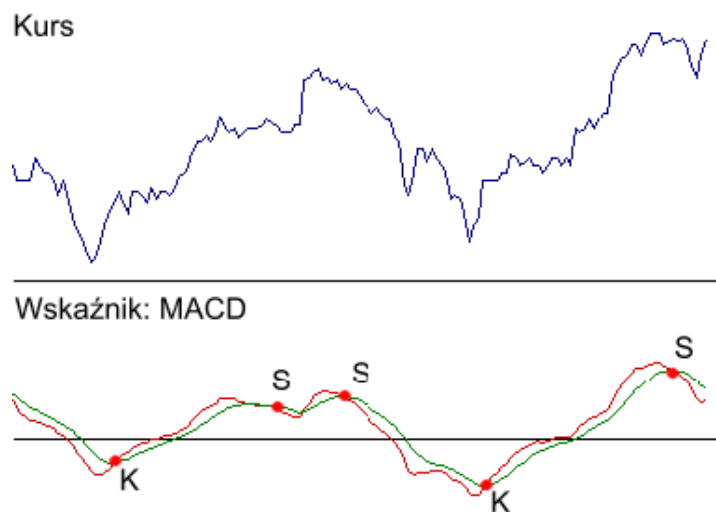
4.1 Wybór rodzaju sieci

Będziemy chcieli zrealizować dwa modele, jeden oparty o sieć rekurencyjną drugi w oparciu o konwolucyjną sieć.

4.2 Źródło danych

Dane giełdowe mogą zostać pobrane z platformy kaggle. Ułatwia to proces pozyskania czystych i spójnych w formacie danych. Platforma kaggle znana jest z konkurencji data science i posiadania szerokiego zbioru dostępnych danych także notowań np. największej światowej giełdy nasdaq.

- cena otwarcia
- cena zamknięcia
- najniższa cena
- najwyższa cena
- wolumen
- średnia krocząca
- wskaźnik MACD (dwie linie, jedna powstaje poprzez różnicę dwóch średnich wykładniczych, natomiast druga to linia sygnałowa (również średnia wykładnicza), przecięcie się lini wyznacza podejmowanie decyzji)
- RSI (wskaźnik siły względnej), obliczany jest w następujący sposób:
$$RSI = 100 - \frac{100}{1 + RS}$$
, gdzie RS to stosunek ważonych średnich ruchomych wzrostu i spadku cen zamknięcia akcji z danych dni
- dane fundamentalne



Rysunek 3: Przykład MACD na wybranym kursie akcji

4.3 Output modelu

Na wyjściu będzie znajdować się wartość binarna. 0 oznacza predykcja spadku ceny, natomiast 1 oznacza predykcja wzrostu ceny akcji. Narażenie jako horyzont czasowy predykcji zostanie przyjęty jeden dzień

4.4 Przygotowanie i analiza danych

Dane zostaną podzielone na trzy części. Do trenowania do weryfikacji i do testowania. Pierwsza z nich będzie używana do nauki modelu, druga do oceny efektywności, a ostatnia do sprawdzenia finalnego modelu przed finalizacją implementacji.

4.5 Przygotowanie i trenowanie modelu

Do trenowania modelu skorzystamy z jednej z dwóch najpopularniejszych bibliotek w python (pytorch albo tensorflow(keras)). Pozwala to na skupienie się na wysokopoziomowym myśleniu o architekturze modelu zamiast na szczegółach implementacji sieci neuronowej.

4.6 Porównanie modelu z rynkiem

Aby móc ocenić model będziemy porównywali, z realnym zachowaniem rynku na podstawie danych testowych. Jeżeli model na wyjściu wygeneruje 0 to zostanie zrealizowany short na danej akcji, jeśli wygeneruje 1 to zostanie zajęta pozycja long. Podjęte operacje zostają zamykane po upływie jednego dnia.

Literatura

- [1] https://tikz.net/neural_networks/
- [2] https://en.wikipedia.org/wiki/Convolutional_neural_network
- [3] https://en.wikipedia.org/wiki/Feedforward_neural_network
- [4] https://en.wikipedia.org/wiki/Recurrent_neural_network