

Zastosowanie do prognoz finansowych

Wstęp

W Częściach 7 i 13 opracowano symulator wstecznej propagacji błędów. W tej Części użyjesz symulatora do rozwiązania złożonego problemu w prognozowaniu finansowym. Zastosowanie sieci neuronowych do prognozowania i modelowania finansowego jest w ostatnich latach bardzo popularne. Czasopisma i magazyny finansowe często wspominają o wykorzystaniu sieci neuronowych, a komercyjne narzędzia i symulatory są dość rozpowszechnione. Ta Część zawiera przegląd typowych kroków stosowanych przy tworzeniu modeli prognozowania finansowego. Wiele kroków zostanie uproszczonych, przez co wyniki nie będą niestety wystarczająco dobre do zastosowania w prawdziwym życiu. Mijmy jednak nadzieję, że posłuży to jako wprowadzenie do tematu z kilkoma dodatkowymi wskazówkami do dalszej lektury i zasobami dla tych, którzy chcą uzyskać bardziej szczegółowe informacje.

Kto handluje z sieciami neuronowymi?

Na Wall Street pojawiło się duże zainteresowanie sieciami neuronowymi. Bradford Lewis zarządza dwoma funduszami Fidelity, częściowo z wykorzystaniem sieci neuronowych. Ponadto LBS Capital Management (Peoria, Illinois) zarządza częścią swojego portfela za pomocą sieci neuronowych. Według Barron's (27 lutego 1995), 150 milionów dolarów funduszu LBS pokonuje średnie o trzy punkty procentowe rocznie od 1992 roku. stawki. Jednak inny model LBS radził sobie gorzej niż S&P 500 przez ostatnie pięć lat. W książce Virtual Trading Jeffrey Katz stwierdza, że większość odnoszących sukcesy systemów sieci neuronowych jest prawnie zastrzeżona i nikt o nich nie mówi. Klienci korzystający z sieci neuronowych zwykle nie chcą, aby ktokolwiek inny wiedział, co robią, z obawy przed utratą przewagi konkurencyjnej. Firmy poświęcają wiele osobołat na projektowanie inżynieryjne z wieloma cyklami procesora, aby osiągnąć praktyczne i opłacalne wyniki. Przyjrzyjmy się procesowi:

Opracowanie modelu prognozowania

Model prognozowania składa się z wielu etapów, które wymieniono poniżej.

1. Zdecyduj, jaki jest twój cel i opracuj sieć neuronową (postępując zgodnie z tymi krokami) dla każdego celu.
2. Określ ramy czasowe, które chcesz prognozować.
3. Zbierz informacje o problematycznej domenie.
4. Zbierz potrzebne dane i wyczuj relację każdego wejścia z wartością docelową.
5. Przetwórz dane, aby wyróżnić funkcje, które sieć ma rozpoznać.
6. Odpowiednio przekształć dane.
7. W razie potrzeby skaluj i ustawiaj dane dla sieci.
8. W miarę możliwości zmniejsz wymiarowość danych wejściowych.
9. Zaprojektować architekturę sieci (topologia, # warstwy, wielkość warstw, parametry, paradygmat uczenia się).
10. Przejdź przez pętlę pociągu/testu/przeprojektowania dla sieci.
11. Wyeliminuj skorelowane dane wejściowe tak bardzo, jak to możliwe, podczas wykonywania kroku 10.

12. Wdróż swoją sieć na nowych danych i przetestuj ją oraz dopracuj w razie potrzeby.

Po opracowaniu modelu prognozowania należy go zintegrować z systemem transakcyjnym. Sieć neuronową można zaprojektować tak, aby przewidywała kierunek lub wielkość, a może po prostu punkty zwrotne na określonym rynku lub coś innego. Avner Mandelman z Cereus Investments (Los Altos Hills, Kalifornia) wykorzystuje wyszkoloną sieć neuronową o dużym zasięgu, aby poinformować go, kiedy rynek osiąga szczyt, czy dół. Przejdźmy teraz do dwunastu aspektów budowania modelu: Cel i ramy czasowe.

Jaki powinien być wynik Twojej sieci neuronowej? Powiedzmy, że chcesz przewidzieć rynek akcji. Czy chcesz przewidzieć S&P 500? A może chcesz przewidzieć kierunek S&P 500? Możesz również przewidzieć zmienność S&P 500 (może jeśli jesteś graczem w opcje). Co więcej, podobnie jak pan Mandelman, możesz chcieć tylko przewidzieć szczyty i dołki, powiedzmy, dla Dow Jones Industrial Average. Musisz zdecydować o rynku lub rynkach, a także o swoich konkretnych celach.

Kolejną kluczową decyzją jest ramy czasowe, które chcesz przewidzieć. Łatwiej jest tworzyć modele sieci neuronowych do prognoz długoterminowych niż do prognoz krótkoterminowych. Możesz zobaczyć wiele szumów rynkowych lub pozornie losowych, chaotycznych zmian w coraz mniejszych rozdzielczościach skali czasowej, które mogą to wyjaśnić. Innym powodem jest to, że siły makroekonomiczne, które fundamentalnie wpływają na rynek w długich okresach, poruszają się powoli. Dolar amerykański tworzy wieloletnie trendy, kształtowane przez politykę gospodarczą rządów na całym świecie. Dla danej tolerancji błędu prognoza roczna lub miesięczna będzie wymagała mniej wysiłku z siecią neuronową niż prognoza jednodniowa.

Ekspertyza domeny

Do tej pory rozmawialiśmy o celu i ramach czasowych. Teraz jeszcze jednym ważnym aspektem budowania modeli jest znajomość domeny. Jeśli chcesz stworzyć skuteczny model prognostyczny pogody, musisz znać lub umieć odgadnąć czynniki, które wpływają na pogodę. To samo dotyczy giełdy lub innego rynku finansowego. Aby stworzyć prawdziwy, zbywalny system handlu obligacjami skarbowymi, musisz mieć dobry pomysł na to, co naprawdę napędza rynek i co działa - np. porozmawiać z handlowcem Tbond i zawrzeć w nim wiedzę na temat jego domeny!

Zbierz dane

Znając czynniki, które wpływają na wynik docelowy, możesz zbierać surowe dane. Jeśli przewidujesz indeks S&P 500, możesz wziąć pod uwagę rentowność obligacji skarbowych, rentowność 3-miesięcznych bonów skarbowych i zarobki jako niektóre z czynników. Po uzyskaniu danych możesz wykonać wykresy punktowe, aby sprawdzić, czy istnieje pewna korelacja między danymi wejściowymi a docelowymi danymi wyjściowymi. Jeśli nie jesteś zadowolony z fabuły, możesz rozważyć inny wkład w jej miejsce.

Wstępne przetwarzanie danych dla sieci

Choć może to zabrzmieć zaskakująco, najprawdopodobniej spędzisz około 90% swojego czasu jako programista sieci neuronowych na masowaniu i przekształcaniu danych w sensowną formę do trenowania sieci. Właściwie zdefiniowaliśmy trzy podetapy w tym obszarze przetwarzania wstępnego w naszej głównej liście:

- Wyróżnij funkcje
- Przekształć

- Skala i stroniczość

Podświetlanie funkcji w danych wejściowych

Powinieneś zaprezentować sieci neuronowe, tak bardzo jak to możliwe, z łatwym sposobem znajdowania wzorców w twoich danych. W przypadku danych szeregów czasowych, takich jak ceny giełdowe w czasie, możesz rozważyć przedstawienie wielkości, takich jak tempo zmian i przyspieszenie (pierwsza i druga pochodna danych wejściowych) jako przykłady. Innymi sposobami wyróżniania danych jest powiększanie niektórych wystąpień. Na przykład, jeśli uważasz interwencję banku centralnego za ważny kwalifikator kursów walutowych, możesz podać jako dane wejściowe do swojej sieci wartość 1 lub 0, aby wskazać obecność lub brak interwencji banku centralnego. Teraz, jeśli dalej uważasz, że działalność banku Rezerwy Federalnej USA jest ważna sama w sobie, możesz to podkreślić, oddzielając ją jako kolejny wkład 1/0. Używanie kodowania 1/0 do oddzielania efektów kompozytowych nazywa się kodowaniem termometru. Istnieje cały zbiór badań zachowań rynkowych, zwany analizą techniczną, z którego możesz również chcieć przedstawić analizy techniczne swoich danych. Istnieje szeroki asortyment matematycznych badań technicznych, które wykonujesz na swoich danych, takich jak na przykład średnie kroczące w celu wygładzenia danych. Istnieją również badania rozpoznawania wzorców, z których można skorzystać, takie jak formacja „podwójnego szczytu”, która rzekomo skutkuje wysokim prawdopodobieństwem znacznego spadku. Aby móc rozpoznać taki wzór, możesz chcieć przedstawić funkcję matematyczną, która pomoże w identyfikacji podwójnego szczytu. Możesz chcieć złagodzić niepożądane szумы w danych wejściowych. Jeśli zauważysz skok w swoich danych, możesz zmniejszyć jego wpływ, przepuszczając go na przykład przez filtr średniej ruchomej. Należy jednak uważać na wprowadzanie nadmiernego opóźnienia w danych wynikowych.

Przekształć dane, jeśli jest to właściwe

W przypadku danych szeregów czasowych można rozważyć użycie transformacji Fouriera, aby przejść do płaszczyzny fazy częstotliwości. Spowoduje to odkrycie okresowych informacji cyklicznych, jeśli takie istnieją. Transformacja Fouriera rozłoży dyskretnie serie danych wejściowych na serię pików częstotliwości, które mierzą istotność każdej składowej częstotliwości. Jeśli giełda rzeczywiście podąża za tak zwanym efektem styczniowym, w którym ceny zwykle rosną, można oczekiwać silnego rocznego komponentu w spektrum częstotliwości. Mark Jurik sugeruje próbkowanie danych z interwałami, które wychwytyują różne okresy cyklu, w swoim artykule dotyczącym przygotowania danych z sieci neuronowych. Możesz użyć innych technik przetwarzania sygnału, takich jak filtrowanie. Oprócz częstotliwości domeny, możesz również rozważyć przeniesienie do innych przestrzeni, na przykład za pomocą przekształcenia falkowego. Możesz także analizować chaotyczny składnik danych za pomocą miar chaosu. Omówienie tych technik wykracza poza zakres tej książki. Jeśli opracowujesz krótkoterminowe systemy sieci neuronowych do handlu, techniki te mogą odgrywać znaczącą rolę w twoich wysiłkach związanych z przetwarzaniem wstępnym. Wszystkie te techniki zapewnią nowe sposoby przeglądania danych pod kątem możliwych funkcji do wykrycia w innych domenach.

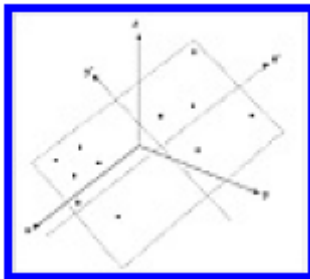
Skaluj swoje dane

Neurony lubią widzieć dane w określonym zakresie wejściowym, aby były najbardziej efektywne. Jeśli przedstawiś dane takie jak S&P 500, które wahają się od 200 do 550 (jak S&P 500 przez lata) nie będą przydatne, ponieważ środkowa warstwa neuronów ma funkcję aktywacji Sigmoid, która zgniata duże dane wejściowe do 0 lub +1. Innymi słowy, powinieneś wybrać dane, które pasują do zakresu, który nie nasycy lub nie przytłacza neuronów sieci. Dobrym pomysłem jest wybór wejść od -1 do 1 lub od 0 do 1. Z tego samego powodu należy znormalizować oczekiwane wartości dla wyjść do zakresu sigmoidalnego od 0 do 1. Ważne jest, aby zwrócić uwagę na liczbę wartości wejściowych w zestawie danych, które są bliskie zeru. Ponieważ prawo zmiany wagi jest proporcjonalne do wartości wejściowej,

to wejście bliskie zeru będzie oznaczało, że ta waga nie będzie uczestniczyć w nauce! Aby uniknąć takich sytuacji, możesz dodać stałe odchylenie do swoich danych, aby przybliżyć dane do 0,5, gdzie neurony reagują bardzo dobrze.

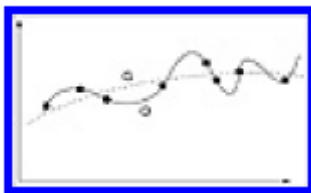
Zmniejsz wymiarowość

Powinieneś starać się eliminować dane wejściowe tam, gdzie to możliwe. Zmniejszy to wymiarowość problemu i ułatwi sieci neuronowej uogólnianie. Załóżmy, że masz trzy wejścia, x , y i z oraz jedno wyjście, o . Załóżmy teraz, że stwierdzasz, że wszystkie twoje wejścia są ograniczone tylko do jednej płaszczyzny. Możesz przedefiniować osie tak, że masz x' i y' dla nowej płaszczyzny i mapować swoje dane wejściowe do nowych współrzędnych. Zmienia to liczbę wejść do twojego problemu na 2 zamiast 3, bez utraty informacji. Ilustruje to rysunek



Generalizacja kontra zapamiętywanie

Jeśli Twój ogólny cel wykracza poza klasyfikację wzorców, musisz śledzić zdolność sieci do uogólniania. Nie tylko powinieneś spojrzeć na ogólny błąd ze zdefiniowanym zbiorem uczącym, ale także odłożyć na bok kilka przykładów uczących jako część zbioru testowego (i nie trenować z nimi), dzięki czemu możesz zobaczyć, czy sieć potrafi poprawnie przewidywać. Jeśli sieć słabo reaguje na twój zestaw testowy, wiesz, że przetrenowałeś się lub możesz powiedzieć, że sieć „zapamiętała” wzorce treningowe. Jeśli spojrzysz na arbitralną analogię dopasowywania krzywych na rysunku, zobaczysz krzywe dla uogólnionego dopasowania, oznaczonego jako G, i przesuniętego, oznaczonego jako O. błędne przewidywanie. Twoje dane testowe z pewnością wykażą duży błąd w przypadku przesadnie dopasowanego modelu.



Innym sposobem na rozważenie tej kwestii jest zastosowanie stopni swobody (DOF). Dla wielomianu:

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \dots$$

DOF równa się liczbie współczynników $a_0, a_1 \dots a_n$, czyli $N + 1$. Zatem dla równania prostej ($y = a_0 + a_1x$), DOF będzie równe 2. Dla paraboli będzie to 3 i tak dalej. Cel, aby nie przepełnić danych może być przekształcony jako cel uzyskania funkcji z najmniej DOF, który odpowiednio pasuje do danych. W przypadku modeli sieci neuronowych im większa liczba możliwych do trenowania wag (która jest funkcją liczby danych wejściowych i architektury), tym większy DOF. Uważaj na zbyt wiele (nieistotnych) danych wejściowych. Możesz znaleźć wspaniałe wyniki z danymi treningowymi, ale wyjątkowo słabe wyniki z danymi testowymi.

Wyliminuj skorelowane dane wejściowe tam, gdzie to możliwe

Widziałeś, że osiągnięcie minimalnej liczby danych wejściowych dla danego problemu jest ważne z punktu widzenia minimalizacji DOF i uproszczenia modelu. Innym sposobem na zmniejszenie wymiarowości jest poszukiwanie skorelowanych danych wejściowych i staranne wyeliminowanie nadmiarowości. Na przykład może się okazać, że frank szwajcarski i marka niemiecka są silnie skorelowane w określonym okresie zainteresowania. Możesz chcieć wyeliminować jeden z tych wejść, aby zmniejszyć wymiarowość. Musisz jednak być ostrożny w tym procesie. Może się okazać, że pozornie zbędna informacja jest w rzeczywistości bardzo ważna. Mark Jurik z Jurik Consulting w swoim artykule na temat wstępnego przetwarzania danych sugeruje, że jednym z najlepszych sposobów określenia, czy dane wejściowe są naprawdę potrzebne, jest skonstruowanie modeli sieci neuronowych z danymi wejściowymi i bez nich oraz wybranie modelu z najlepszym błędem podczas uczenia i dane testowe. Chociaż bardzo iteracyjny, możesz spróbować wyeliminować w ten sposób jak najwięcej danych wejściowych i mieć pewność, że nie wyeliminowałeś zmiennej, która naprawdę zrobiła różnicę. Innym podejściem jest analiza wrażliwości, w której nieznacznie zmieniasz jeden sygnał wejściowy, jednocześnie utrzymując wszystkie inne na stałym poziomie i odnotowując wpływ na wynik. Jeśli efekt jest niewielki, eliminujesz to wejście. To podejście jest wadliwe, ponieważ w prawdziwym świecie wszystkie dane wejściowe nie są stałe. Podejście Jurika jest bardziej czasochłonne, ale prowadzi do lepszego modelu. Proces dekorelacji lub eliminowania skorelowanych danych wejściowych może również wykorzystywać technikę algebry liniowej zwaną analizą głównych składowych. Wynikiem analizy głównych składowych jest minimalny zestaw zmiennych, który zawiera maksimum informacji. Aby uzyskać więcej informacji na temat analizy głównych składowych, należy zapoznać się z odniesieniem statystycznym lub zbadać dwie metody analizy głównych składowych: transformatę Karhunen-Loeva i transformatę Hotellinga.

Zaprojektuj architekturę sieci

Teraz nadszedł czas na zaprojektowanie sieci neuronowej. W przypadku zaprojektowanej przez nas sieci neuronowej ze sprzężeniem do przodu i propagacją wsteczną oznacza to dokonanie następujących wyborów:

1. Liczba ukrytych warstw.
2. Rozmiar ukrytych warstw.
3. Stała uczenia, β .
4. Parametr pędu, α .
5. Forma funkcji zgniatania (nie musi być esicy).
6. Punkt wyjścia, czyli początkowa macierz wag.
7. Dodanie szumu.

Niektóre z wymienionych parametrów mogą zmieniać się w zależności od liczby wykonanych cykli, podobnie do obecnej implementacji szumu. Na przykład możesz zacząć od stałej uczenia β , która jest duża i zmniejszać tę stałą w miarę postępów w nauce. Pozwala to na szybkie wstępne uczenie się na początku procesu i może przyspieszyć całkowity czas symulacji.

Pętla treningu/testu/przeprojektowania

Wiele z procesu określania najlepszych parametrów dla danej aplikacji odbywa się metodą prób i błędów. Musisz poświęcić dużo czasu na ocenę różnych opcji, aby znaleźć najlepsze rozwiązanie dla

swojego problemu. Możesz dosłownie stworzyć setki, jeśli nie tysiące sieci, ręcznie lub automatycznie, aby znaleźć najlepsze rozwiązanie. Wiele komercyjnych programów sieci neuronowych wykorzystuje algorytmy genetyczne, aby pomóc w automatycznym dotarciu do optymalnej sieci. Algorytm genetyczny tworzy możliwe rozwiązania problemu z zestawu genów początkowych. Analogicznie do ewolucji biologicznej, algorytm łączy rozwiązania genetyczne z predefiniowanym zestawem operatorów, tworząc nowe generacje rozwiązań, które przetrwają lub giną w zależności od ich zdolności do rozwiązania problemu. Kluczową zaletą algorytmów genetycznych (GA) jest możliwość przemierzania ogromnej przestrzeni poszukiwań w celu znalezienia możliwie optymalnego rozwiązania. Zaprogramowałbyś GA do wyszukiwania liczby ukrytych warstw i innych parametrów sieci i stopniowo rozwijał rozwiązanie sieci neuronowej. Niektórzy dostawcy używają GA tylko do przypisania początkowego zestawu wag do sieci, zamiast losowania wag, aby zacząć od dobrego rozwiązania.

Przyjrzyjmy się teraz krokom:

1. Podziel swoje dane. Najpierw podziel zbiór danych na trzy części, zbiór uczący, zbiór testowy i ślepy zbiór testowy. Użyj około 80% swoich rekordów danych dla swojego zestawu treningowego, 10% dla swojego zestawu testowego i 10% dla swojego ślepego zestawu testowego.
2. Trenuj i testuj. Następnie zacznij od topologii sieci i trenuj swoją sieć na podstawie danych zestawu treningowego. Gdy osiągniesz zadowalający minimalny błąd, zapisz swoje wagi i zastosuj wytrenowaną sieć do danych testowych i zanotuj błąd. Teraz zrestartuj proces z tą samą topologią sieci dla innego zestawu początkowych wag i sprawdź, czy możesz osiągnąć lepszy błąd w zestawach uczących i testowych. Rozumowanie: mogłeś znaleźć lokalne minimum przy pierwszej próbie, a losowanie początkowych wag spowoduje, że zaczniesz od innego, być może lepszego rozwiązania.
3. Wyeliminuj skorelowane wejścia. Możesz opcjonalnie spróbować w tym momencie, aby sprawdzić, czy możesz wyeliminować skorelowane dane wejściowe, jak wspomniano wcześniej, przez iteracyjne usuwanie każdego wejścia i notowanie najlepszego błędu, jaki możesz osiągnąć w zestawach uczących i testowych dla każdego z tych przypadków. Wybierz przypadek, który prowadzi do najlepszego błędu i wyeliminuj dane wejściowe (jeśli istnieją), które go osiągnęły. Możesz powtórzyć cały ten proces ponownie, aby spróbować wyeliminować inną zmienną wejściową.
4. Iteracyjnie trenuj i testuj. Teraz możesz wypróbować inne parametry sieci i powtórzyć proces pociągu i testowania, aby osiągnąć lepszy wynik.
5. Wdróż swoją sieć. Teraz możesz użyć ślepego zestawu danych testowych, aby zobaczyć, jak działa zoptymalizowana sieć. Jeśli błąd nie jest zadowalający, musisz ponownie przejść do fazy projektowania lub do fazy pociągu i testów.
6. Przyjrzyj się projektowi sieci, gdy zmienią się warunki. Musisz przeszkolić swoją sieć, gdy masz powody, by sądzić, że masz nowe informacje związane z modelowanym problemem. Jeśli masz sieć neuronową, która próbuje przewidzieć tygodniową zmianę w S&P 500, prawdopodobnie będziesz musiał przeszkolić swoją sieć co najmniej raz w miesiącu, jeśli nie raz w tygodniu. Jeśli okaże się, że sieć nie uogólnia już dobrze nowych informacji, musisz ponownie wejść w fazę projektowania.

Jeśli brzmi to jak dużo pracy, to jest! Teraz spróbujmy szczęścia w prognozowaniu, przechodząc przez podzbiór przedstawionych kroków

Prognozowanie S&P 500

Indeks S&P 500 jest powszechnie stosowaną średnią giełdową, podobnie jak Dow Jones Industrial Average (DJIA). Ma szerszą reprezentację rynku akcji, ponieważ ta średnia opiera się na 500 akcjach, podczas gdy DJIA opiera się na tylko 30. Problemem, do którego należy podejść w tym rozdziale, jest przewidzenie indeksu S&P 500, biorąc pod uwagę różne wskaźniki i dane za poprzednie tygodnie.

Wybór właściwych wyników i celów

Naszym celem jest prognozowanie indeksu S&P 500 za dziesięć tygodni od teraz. Podczas gdy celem może być przewidzenie poziomu S&P 500, ważne jest, aby uprościć pracę sieci, prosząc o zmianę poziomu, a nie o bezwzględny poziom indeksu. To, co chcesz zrobić, to dać sieci możliwość wygodnego dopasowania problemu w przestrzeni wyjściowej warstwy wyjściowej. Praktycznie rzecz biorąc, wiesz, że dane wyjściowe z sieci nie mogą znajdować się poza zakresem od 0 do 1, ponieważ użyliśmy funkcji aktywacji sigmoidalnej. Możesz na przykład wziąć indeks S&P 500 i przeskalować ten bezwzględny poziom cen do tego zakresu. Jednak prawdopodobnie otrzymasz bardzo małe liczby, które mają niewielki zakres zmienności. Z drugiej strony różnica z tygodnia na tydzień ma znacznie mniejszy ogólny zakres, a gdy te różnice są skalowane do zakresu od 0 do 1, masz znacznie większą zmienność. Wybrany przez nas wyjściem jest zmiana S&P 500 z bieżącego tygodnia na 10 tygodni od teraz jako procent wartości z bieżącego tygodnia.

Wybór odpowiednich wejść

Dane wejściowe do sieci muszą być cotygodniowymi zmianami wskaźników, które mają pewne znaczenie dla indeksu S&P 500. Jest to złożony problem prognozowania i możemy tylko zgadywać niektóre zależności. Jest to jedna z nieodłącznych zalet korzystania z sieci neuronowych do prognozowania; jeśli związek jest słaby, sieć nauczy się go automatycznie ignorować. Pamiętaj jednak, że chcesz zminimalizować DOF, jak wspomniano wcześniej. W tym przykładzie wybieramy zestaw danych, który reprezentuje stan rynków finansowych i gospodarki. Wybrane wejścia są wymienione jako:

- Poprzednia akcja cenowa w indeksie S&P 500, w tym zamknięcie lub końcowa wartość indeksu
- Wskaźniki szerokości dla rynku akcji, w tym liczba postępujących i spadających emisji akcji na giełdzie nowojorskiej (NYSE)
- Inne wskaźniki techniczne, w tym liczba nowych maksimum i nowych minimum osiągniętych w ciągu tygodnia dla rynku NYSE. Daje to pewną wskazówkę na temat siły trendu wzrostowego lub spadkowego.
- Stopy procentowe, w tym stopy krótkoterminowe w Rentowności Trzymiesięcznych Bonów Skarbowych oraz długoterminowe w Rentowności 30-letnich Obligacji Skarbowych.

Innymi możliwymi danymi wejściowymi mogły być statystyki rządowe, takie jak wskaźnik cen konsumpcyjnych, liczba rozpoczętych budów i stopa bezrobocia. Nie zostały one wybrane, ponieważ długoterminowe i krótkoterminowe stopy procentowe zwykle obejmują już te dane. Zachęcamy do eksperymentowania z innymi danymi wejściowymi i pomysłami. Wszystkie wymienione dane można uzyskać w domenie publicznej, np. z publikacji finansowych oraz z witryn FTP w Internecie dla Departamentu Handlu i Rezerwy Federalnej, a także z sprzedawcy komercyjni. W Internecie cały czas pojawiają się nowe źródła.

Wybór architektury sieci

Warstwy wejściowe i wyjściowe są ustalone przez liczbę wejść i wyjść, których używamy. W naszym przypadku wyjściem jest pojedyncza liczba, oczekiwana zmiana indeksu S&P 500 za 10 tygodni.

Rozmiar warstwy wejściowej będzie podyktowany liczbą danych wejściowych, które mamy po wstępnym przetworzeniu. Więcej na ten temat zobaczysz wkrótce. Warstwy środkowe mogą mieć wartość 1 lub 2. Najlepiej wybrać najmniejszą liczbę neuronów możliwą dla danego problemu, aby umożliwić uogólnienie. Jeśli neuronów jest za dużo, będziesz miał tendencję do zapamiętywania wzorów. Użyjemy jednej ukrytej warstwy. Ogólnie zaleca się, aby rozmiar pierwszej warstwy ukrytej wynosił od połowy do trzech razy rozmiar warstwy wejściowej. Jeśli obecna jest druga warstwa ukryta, możesz mieć od trzech do dziesięciu razy więcej neuronów wyjściowych. Najlepszym sposobem określenia optymalnego rozmiaru jest metoda prób i błędów.

UWAGA: Powinieneś upewnić się, że istnieje wystarczająca liczba przykładów treningu dla twoich ciężarów, które można trenować. Innymi słowy, twoja architektura może być podyktowana liczbą przykładów szkolenia danych wejściowych lub faktów, które masz. W idealnym świecie chciałbyś mieć około 10 lub więcej faktów na każdą wagę. W przypadku architektury 10-10-1 istnieje ($10 \times 10 + 10 \times 1 = 110$ wag), więc powinieneś dążyć do około 1100 faktów. Im mniejszy stosunek faktów do wag, tym większe prawdopodobieństwo niedotrenowania swojej sieci, co prowadzi do bardzo słabej zdolności generalizacji.

Wstępne przetwarzanie danych

Rozpoczynamy teraz wstępne przetwarzanie. Jak wspomniano wcześniej, najprawdopodobniej to właśnie tam, jako projektant sieci neuronowych, spędzisz większość swojego czasu.

Widok surowych danych

Spójrzmy na surowe dane dotyczące problemu, który chcemy rozwiązać. Istnieje kilka sposobów na rozpoczęcie wstępnego przetwarzania danych w celu zmniejszenia liczby danych wejściowych i zwiększenia zmienności danych:

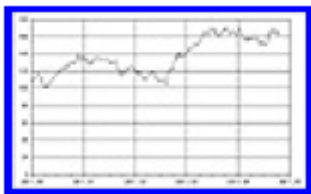
- Użyj współczynnika postępow/spadków zamiast każdej wartości osobno.
- Stosuj stosunek Nowe maks./nowe minim. zamiast każdej wartości osobno.

Pozostały nam następujące wskaźniki:

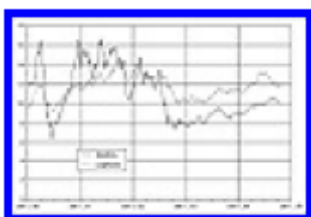
1. Rentowność trzymiesięcznych bonów skarbowych
2. Rentowność 30-letnich obligacji skarbowych
3. Problemy z awansem/spadkiem NYSE
4. NYSE Nowe wloty/nowe upadki
5. Cena zamknięcia S&P 500

Surowe dane za okres od 4 stycznia 1980 r. do 28 października 1983 r. są traktowane jako okres uczący, łącznie 200 tygodni danych. Kolejne 50 tygodni jest trzymane w rezerwie na okres testowy, aby sprawdzić, czy prognozy są ważne poza interwałem treningowym. Ostatnia data tego okresu to 19 października 1984 r. Przyjrzyjmy się teraz surowym danym. (Otrzymujesz dane na dysku dostępnym w tej książce, które obejmują okres od stycznia 1980 r. do grudnia 1992 r.) Na poniższych rysunkach zobaczysz szereg tych wskaźników wykreślonych w interwałach treningowych i testowych:

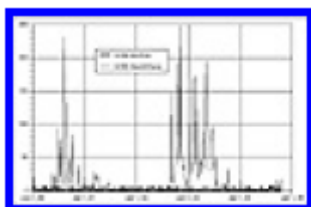
- Rysunek przedstawia indeks giełdowy S&P 500.



- Wykres przedstawia obligacje długoterminowe oraz krótkoterminowe oprocentowanie 3-miesięcznych bonów skarbowych.



- Wykres przedstawia niektóre wskaźniki szerokości na NYSE, liczbę akcji na plusie/liczbę akcji spadających, a także stosunek nowych maksimów do nowych minimów na NYSE



Próbka kilku linii wygląda jak następujące dane w tabeli 14.1. Zauważ, że kolejność parametrów jest taka sama, jak wymieniona powyżej.

Date	3Mo TBills	30YrTBonds	NYSE-Adv/Dec	NYSE-NewH/NewL	SP-Close
1/4/80	12.11	9.64	4.209459	2.764706	106.52
1/11/80	11.94	9.73	1.649573	21.28571	109.92
1/18/80	11.9	9.8	0.881335	4.210526	111.07
1/25/80	12.19	9.93	0.793269	3.606061	113.61
2/1/80	12.04	10.2	1.16293	2.088235	115.12
2/8/80	12.09	10.48	1.338415	2.936508	117.95
2/15/80	12.31	10.96	0.338053	0.134615	115.41
2/22/80	13.16	11.25	0.32381	0.109091	115.04
2/29/80	13.7	12.14	1.676895	0.179245	113.66
3/7/80	15.14	12.1	0.282591	0	106.9
3/14/80	15.38	12.01	0.690286	0.011628	105.43
3/21/80	15.05	11.73	0.486267	0.027933	102.31
3/28/80	16.53	11.67	5.247191	0.011628	100.68
4/3/80	15.04	12.06	0.983562	0.117647	102.15
4/11/80	14.42	11.81	1.565854	0.310345	103.79
4/18/80	13.82	11.23	1.113287	0.146341	100.55
4/25/80	12.73	10.59	0.849807	0.473684	105.16
5/2/80	10.79	10.42	1.147465	1.857143	105.58
5/9/80	9.73	10.15	0.513052	0.473684	104.72
5/16/80	8.6	9.7	1.342444	6.75	107.35
5/23/80	8.95	9.87	3.110825	26	110.62

Wyróżnij funkcje w danych

Dla każdego z pięciu danych wejściowych chcemy użyć funkcji do podświetlenia typu szybkości zmian funkcji. W tym celu użyjemy następującej funkcji (jak pierwotnie zaproponował Jurik).

$$ROC(n) = (\text{wejście}(t) - BA(t - n)) / (\text{wejście}(t) + BA(t - n))$$

gdzie: input(t) to bieżąca wartość wejścia, a BA(t - n) to pięciojednostkowa średnia blokowa sąsiednich wartości wyśrodkowanych wokół wartości sprzed n okresów. Teraz musimy zdecydować, ile z tych funkcji potrzebujemy. Ponieważ prognozujemy 10 tygodni w przyszłość, będziemy pobierać dane nawet za 10 tygodni. To będzie ROC(10). Wykorzystamy również jeszcze jedną stopę zmian, ROC(3). Dodaliśmy teraz $5 * 2 = 10$ wejść do naszej sieci, łącznie 15. Całe wstępne przetwarzanie można wykonać za pomocą arkusza kalkulacyjnego. Oto, co otrzymujemy (Tabela 14.2) po wykonaniu średnich blokowych. Przykład: BA3MoBills za 1/18/80 = (3MoBills (1/4/80) + 3MoBills (1/11/80) + 3MoBills (1/18/80) + 3MoBills(1/25/80) + 3MoBills(2 /1/80))/5.

Date	3MoBills	LngBonds	NYSE-
1/4/80	12.11	9.64	4.209459
1/11/80	11.94	9.73	1.649573
1/18/80	11.9	9.8	0.881335
1/25/80	12.19	9.93	0.793269
2/1/80	12.04	10.2	1.16293
2/8/80	12.09	10.48	1.338415
2/15/80	12.31	10.96	0.338053
2/22/80	13.16	11.25	0.32381
2/29/80	13.7	12.14	1.676895
3/7/80	15.14	12.1	0.282591
3/14/80	15.38	12.01	0.690286
3/21/80	15.05	11.73	0.486267
3/28/80	16.53	11.67	5.247191
4/3/80	15.04	12.06	0.983562
4/11/80	14.42	11.81	1.565854
4/18/80	13.82	11.23	1.113287
4/25/80	12.73	10.59	0.849807
5/2/80	10.79	10.42	1.147465

5/9/80	9.73	10.15	0.513052
5/16/80	8.6	9.7	1.342444
5/23/80	8.95	9.87	3.110825
NYSE-Adv/Dec	SP-Close	BA3MoB	BALngBnd
	NewH/NewL		
2.764706	106.52		
21.28571	109.92		
4.210526	111.07	12.036	9.86
3.606061	113.61	12.032	10.028
2.088235	115.12	12.106	10.274
2.936508	117.95	12.358	10.564
0.134615	115.41	12.66	11.006
0.109091	115.04	13.28	11.386
0.179245	113.66	13.938	11.692
0	106.9	14.486	11.846
0.011628	105.43	15.16	11.93
0.027933	102.31	15.428	11.914
0.011628	100.68	15.284	11.856
0.117647	102.15	14.972	11.7
0.310345	103.79	14.508	11.472
0.146341	100.55	13.36	11.222
0.473684	105.16	12.298	10.84
1.857143	105.58	11.134	10.418
0.473684	104.72	10.16	10.146
6.75	107.35	7.614	8.028
26	110.62	5.456	5.944
BAA/D	BAH/L	BAClose	
1.739313	6.791048	111.248	
1.165104	6.825408	113.534	
0.9028	2.595189	114.632	
0.791295	1.774902	115.426	

0.968021	1.089539	115.436
0.791953	0.671892	113.792
0.662327	0.086916	111.288
0.69197	0.065579	108.668
1.676646	0.046087	105.796
1.537979	0.033767	103.494
1.794632	0.095836	102.872
1.879232	0.122779	101.896
1.95194	0.211929	102.466
1.131995	0.581032	103.446
1.037893	0.652239	103.96
0.993211	1.94017	104.672
1.392719	7.110902	106.686
1.222757	7.016165	85.654
0.993264	6.644737	64.538

Przyjrzyjmy się teraz dalszej części tej tabeli, która składa się z nowych 10 wartości wskaźników ROC

Date	ROC3_3Mo	ROC3_Bond	ROC10_AD	ROC3_H/L	ROC3_SPC
1/4/80					
1/11/80					
1/18/80					
1/25/80					
2/1/80					
2/8/80	0.002238	0.030482	-0.13026	-0.39625	0.029241
2/15/80	0.011421	0.044406	-0.55021	-0.96132	0.008194
2/22/80	0.041716	0.045345	-0.47202	-0.91932	0.001776
2/29/80	0.0515	0.069415	0.358805	-0.81655	-0.00771
3/7/80	0.089209	0.047347	-0.54808	-1	-0.03839
3/14/80	0.073273	0.026671	-0.06859	-0.96598	-0.03814
3/21/80	0.038361	0.001622	-0.15328	-0.51357	-0.04203
3/28/80	0.065901	-0.00748	0.766981	-0.69879	-0.03816
4/3/80	-0.00397	0.005419	-0.26054	0.437052	-0.01753
4/11/80	-0.03377	-0.00438	0.008981	0.437052	-0.01753
4/18/80	-0.0503	-0.02712	-0.23431	0.803743	0.001428
4/25/80	-0.08093	-0.0498	-0.37721	0.58831	0.015764
5/2/80	-0.14697	-0.04805	-0.25956	0.795146	0.014968
5/9/80	-0.15721	-0.05016	-0.37625	-0.10178	0.00612

5/16/80	-0.17695	-0.0555	0.127944	0.823772	0.016043
5/23/80	-0.10874	-0.02701	0.515983	0.86112	0.027628
ROC10_3Mo	ROC10_Bnd	ROC10_AD	ROC10_HL	ROC10_SP	

0.15732	0.084069	0.502093	-0.99658	-0.04987
0.111111	0.091996	-0.08449	-0.96611	-0.05278
0.087235	0.069553	0.268589	-0.78638	-0.04964
0.055848	0.030559	0.169062	-0.84766	-0.06888
0.002757	-0.01926	-0.06503	-0.39396	-0.04658
-0.10345	-0.0443	0.183309	0.468658	-0.03743
-0.17779	-0.0706	-0.127	0.689919	-0.03041
-0.25496	-0.0996	0.319735	0.980756	-0.0061
-0.25757	-0.0945	0.299569	0.996461	0.02229

UWAGA: Pamiętaj, że nie otrzymasz wypełnionych wierszy do 28.03.90, ponieważ mamy wskaźnik ROC zależny od wartości średniej bloku 10 tygodni przed nim. Pierwsza średnia wartość bloku jest generowana 1/1/80, dwa tygodnie po uruchomieniu zestawu danych. Wszystko to wskazuje, że będziesz musiał odrzucić pierwszych 12 wartości w zestawie danych, aby uzyskać pełne wiersze, zwane również kompletnymi faktami.

Normalizacja zasięgu

Mamy teraz wartości w oryginalnych pięciu kolumnach danych, które mają bardzo duży zakres. Chcielibyśmy w jakiś sposób zmniejszyć zasięg. Korzystamy z następującej funkcji:

nowa wartość = (stara wartość - średnia) / (maksymalny zakres)

Odnosi się to do odległości od średniej wartości w kolumnie jako ułamek maksymalnego zakresu dla tej kolumny. Należy zanotować wartość Zakresu maksymalnego i Średnia, aby można było odnormalizować dane po otrzymaniu wyniku.

Cel

Zadbaliśmy o wszystkie nasze wejścia, które mają numer 15. Ostatnią informacją jest cel. Celem określonym na początku tego ćwiczenia jest przewidzenie zmiany procentowej 10 tygodni w przyszłość. Musimy przesunąć czas zamknięcia S&P 500 o 10 tygodni wstecz, a następnie obliczyć wartość jako zmianę procentową w następujący sposób:

Wynik = $100 \times ((\text{S\&P 10 tygodni do przodu}) - (\text{S\&P w tym tygodniu})) / (\text{S\&P w tym tygodniu})$.

Daje nam to wartość, która waha się od -14,8 do + 33,7. To nie jest jeszcze w takiej formie, jakiej potrzebujemy. Jak pamiętasz, wyjście pochodzi z funkcji sigmoid, która jest ograniczona do 0 do +1.

Najpierw dodamy 14,8 do wszystkich wartości, a następnie przeskalujemy je o współczynnik 0,02. Spowoduje to skalowanie celu, który waha się od 0 do 1.

przeskalowany cel = (wynik + 14,8) X 0,02

Ostateczny plik danych ze skalowanym celem pokazanym wraz z przeskalowanymi oryginalnymi sześcioma kolumnami danych przedstawiono poniżej.

Date	S_3MOBill	S_LngBnd	S_A/D
3/28/80	0.534853	-0.01616	0.765273
4/3/80	0.391308	0.055271	-0.06356
4/11/80	0.331578	0.009483	0.049635
4/18/80	0.273774	-0.09674	-0.03834
4/25/80	0.168765	-0.21396	-0.08956
5/2/80	-0.01813	-0.2451	-0.0317
5/9/80	-0.12025	-0.29455	-0.15503
5/16/80	-0.22912	-0.37696	0.006205
5/23/80	-0.1954	-0.34583	0.349971

S_H/L	S_SPC	Result	scaled Target
-0.07089	-0.51328	12.43544	0.544709
-0.07046	-0.49236	12.88302	0.55366
-0.06969	-0.46901	9.89498	0.4939
-0.07035	-0.51513	15.36549	0.60331
-0.06903	-0.44951	11.71548	0.53031
-0.06345	-0.44353	11.61205	0.528241
-0.06903	-0.45577	16.53934	0.626787
-0.04372	-0.41833	12.51048	0.54621
0.033901	-0.37179	9.573314	0.487466

Przechowywanie danych w różnych plikach

Należy umieścić pierwsze 200 wierszy w pliku training.dat (dostarczonym na dołączonej dyskietce), a kolejne 40 wierszy danych w innym pliku test.dat do wykorzystania w testach. Więcej na ten temat przeczytasz wkrótce. Na tej dyskietce znajduje się również więcej danych w postaci surowej, abyś mógł przeprowadzić dalsze eksperymenty.

Szkolenia i testy

Mając dostępne dane treningowe, konfigurujemy symulację. Liczba wejść to 15, a wyjść to 1. Łącznie używane są trzy warstwy ze środkową warstwą o rozmiarze 5. Ta liczba powinna być jak najmniejsza,

aby uzyskać akceptowalne wyniki. Optymalne rozmiary i liczbę warstw można znaleźć tylko metodą prób i błędów. Po każdym biegu możesz spojrzeć na błąd ze zbioru treningowego i ze zbioru testowego.

Używanie symulatora do obliczania błędu

Błąd zestawu testowego uzyskuje się uruchamiając symulator w trybie Trening (trzeba tymczasowo skopiować dane testowe z oczekiwanymi wyjściami do pliku training.dat) na jeden cykl z obciążnikami załadowanymi z pliku obciążników. Ponieważ jest to ostatni i jedyny cykl, wagi nie są modyfikowane i można uzyskać odczyt średniego błędu. Więcej informacji na temat trybów testowych i treningowych symulatora można znaleźć w Części 13. Takie podejście przyjęto w przypadku pięciu przebiegów symulatora po 500 cykli każdy. Tabela podsumowuje wyniki wraz z zastosowanymi parametrami. Błąd staje się coraz lepszy z każdym przebiegiem do przebiegu nr 4. W przebiegu nr 5 błąd zestawu treningowego zmniejsza się, ale błąd zestawu testowego wzrasta, wskazując na początek zapamiętywania. Przebieg nr 4 jest używany do ostatecznych wyników sieci, pokazując błąd RMS zestawu testowego wynoszący 13,9% i błąd zestawu treningowego wynoszący 6,9%.

Run#	Tolerance	Beta	Alpha	NF	max cycles	cycles run	training set error	test set error
1	0.001	0.5	0.001	0.0005	500	500	0.150938	0.25429
2	0.001	0.4	0.001	0.0005	500	500	0.114948	0.185828
3	0.001	0.3	0	0	500	500	0.0936422	0.148541
4	0.001	0.2	0	0	500	500	0.068976	0.139230
5	0.001	0.1	0	0	500	500	0.0621412	0.143430

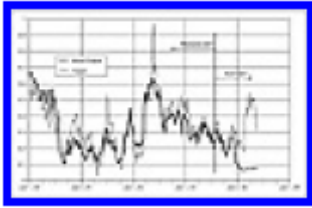
UWAGA: Jeśli stwierdzisz, że błąd zestawu testowego nie zmniejsza się znacznie, podczas gdy błąd zestawu treningowego nadal powoduje znaczne postępy, oznacza to, że zaczyna się zapamiętywanie (w przykładzie przebieg nr 5). Ważne jest, aby monitorować zestawy testowe, których używasz podczas treningu, aby upewnić się, że ma miejsce dobre, uogólnione uczenie się, a nie zapamiętywanie lub nadmierne dopasowanie danych. W pokazanym przypadku, błąd zestawu testowego poprawiał się aż do przebiegu nr 5, w którym błąd zestawu testowego uległ degradacji. Musisz ponownie przejść do 12-etapowego procesu projektowania modelu prognozowania, aby wprowadzić dalsze ulepszenia wykraczające poza to, co zostało osiągnięte.

Aby zobaczyć dokładną korelację, możesz skopiować dowolny okres, z usuniętymi polami wyjściowymi wartości oczekiwanej, do pliku test.dat. Następnie uruchamiasz symulator w trybie testowym i otrzymujesz z symulatora wartość wyjściową dla każdego wektora wejściowego. Następnie możesz to porównać z oczekiwaną wartością w zestawie treningowym lub zestawie testowym. Teraz, gdy skończyłeś, musisz cofnąć normalizację danych, aby uzyskać odpowiedź w zakresie zmiany indeksu S&P 500. To, co udało Ci się osiągnąć, to sposób, w jaki możesz uzyskać dane z gazet finansowych, takich jak Barron's lub Investor's Business Daily, i wprowadzić dane z bieżącego tygodnia do wytrenowanej sieci neuronowej, aby uzyskać prognozę tego, co indeks S&P 500 prawdopodobnie zrobi w dziesięć tygodni od teraz. Oto kroki, aby cofnąć normalizację:

1. Weź przewidywaną przeskalowaną wartość docelową i oblicz wartość wyniku jako Wynik = (Skalowana wartość docelowa / 0,02) - 14,8
2. Weź wynik z kroku 1 (który jest zmianą procentową za 10 tygodni od teraz) i oblicz przewidywaną wartość, Przewidywany S&P za 10 tygodni = (Wartość S&P z tego tygodnia) (1 + Wynik / 100)

Tylko początek

To tylko bardzo krótka ilustracja (nie przeznaczona do handlu!) tego, co można zrobić z sieciami neuronowymi w prognozowaniu finansowym. Musisz dalej analizować dane, dostarczać więcej wskaźników predykcyjnych i optymalizować/przeprojektować architekturę sieci neuronowej, aby uzyskać lepszą generalizację i mniejszy błąd. Musisz przedstawić wiele, wiele więcej przypadków testowych reprezentujących różne warunki rynkowe, aby mieć solidny predyktor, z którym można handlować. Wykres oczekiwanych i przewidywanych wyników dla zbioru testowego i zbioru uczącego pokazano na rysunku. Tutaj do wyjścia używane są znormalizowane wartości. Zauważ, że błąd wynosi średnio około 13,9% w zestawie testowym i 6,9% w zestawie uczącym. Widać, że zestaw testowy na początku spisywał się dobrze, ale w ostatnich tygodniach wykazywał duże rozbieżności.



Kroki przetwarzania wstępnego przedstawione w tym rozdziale powinny służyć jako jeden przykład rodzajów kroków, których możesz użyć. Istnieje wiele różnych metod analitycznych i statystycznych, które można wykorzystać w przetwarzaniu wstępnym. Aby zastosować dane rozmyte, możesz użyć programu takiego jak program fuzzifier, który został opracowany w rozdziale 3, aby wstępnie przetworzyć niektóre dane.

Co dalej?

Od tego momentu możesz przeprowadzić wiele innych eksperymentów. Wybrany przykład dotyczył prognozowania finansowego. Ale z pewnością możesz wypróbować symulator w innych problemach, takich jak prognozowanie sprzedaży, a może nawet prognozowanie pogody. Kluczem do wszystkich aplikacji jest jednak sposób prezentacji i ulepszania danych oraz praca z wyborem parametrów metodą prób i błędów. Przed zakończeniem tego rozdziału omówimy więcej tematów związanych z przetwarzaniem wstępnym i przedstawimy kilka studiów przypadku dotyczących prognoz finansowych. Jeśli jesteś bardziej zainteresowany tym obszarem, powinieneś rozważyć sugestie zawarte w 12-etapowym podejściu do projektowania modeli prognozowania i zbadać niektóre z zasobów wymienionych na końcu tego rozdziału.

Analiza techniczna i wstępne przetwarzanie sieci neuronowych

Nie możemy przecenić znaczenia przetwarzania wstępnego w opracowaniu modelu prognostycznego. Istnieje wiele informacji związanych z badaniem zachowań rynków finansowych, zwanych analizą techniczną. Możesz użyć badań matematycznych zdefiniowanych przez analizę techniczną, aby wstępnie przetworzyć dane wejściowe w celu ujawnienia funkcji predykcyjnych. Przedstawimy próbkę badań analizy technicznej, które można wykorzystać, wraz ze wzorami i wykresami.

Średnie kroczące

Średnie kroczące są bardzo szeroko stosowane do uchwycenia podstawowego trendu ruchu cenowego. Średnie kroczące to proste filtry, które uśredniają dane w ruchomym oknie. Popularne średnie kroczące obejmują 5-, 10- i 20-okresowe średnie kroczące. Poniżej przedstawiono wzór dla prostej średniej ruchomej SMA:

$$SMA_t = (P_t + P_{t-1} + \dots + P_{t-n}) / n$$

where n = the number of time periods back
 P_{-n} = price at n time periods back

Wykładnicza średnia krocząca to ważona średnia krocząca, która przywiązuje większą wagę do najnowszych danych. Wzór na ten wskaźnik, EMA, jest następujący:

$$EMA_t = (1 - a)P_t + a (EMA_{t-1})$$

where a = smoothing constant (typical 0.10)
 P_t = price at time t

Pęd i tempo zmian

Momentum to tak naprawdę prędkość, czyli tempo zmiany ceny w czasie. Wzór na to jest

$$M_t = (P_t - P_{t-a})$$

where a = lookback parameter
 for a 5-day momentum value, a = 5

Wskaźnik Rate of Change jest w rzeczywistości stosunkiem. Jest to aktualna cena podzielona przez cenę z jakiegoś przedziału, a, temu podzielona przez stałą. Konkretnie,

$$ROC = P_t / P_{t-a} \times 1000$$

Wskaźnik siły względnej

Wskaźnik siły względnej (RSI) to siła zamknięcia w górę w stosunku do zamknięcia w dół w określonym przedziale czasu. Jest obliczany w przedziale czasu T jako :

$$RSI = 100 - [100 / (1 + RS)]$$

where RS = average of x days' up closes/ average of x days' down closes

Typowy przedział czasu, T, wynosi 14 dni. Założenie przy użyciu RSI jest takie, że wyższe wartości zamknięć w górę w stosunku do zamknięć w dół wskazują na silny rynek, a odwrotnie wskazuje na rynki słabe.

Procent R

Ten wskaźnik mierzy, gdzie w ostatnim przedziale cen dzisiejsza cena spada. Wskaźnik zakłada, że ceny powracają do swojej średniej. Niski %R wskazuje, że ceny osiągnęły górną granicę zakresu, a wysoki %R wskazuje, że ceny są najniższe w zakresie. Formuła to:

$$\%R = 100 \times (HighX - P) / (HighX - LowX)$$

where HighX is the highest price over the price interval of interest
 LowX is the lowest price over the price interval of interest
 P is the current price

Wskaźnik wypłat Herricka

Ten wskaźnik wykorzystuje inne dane rynkowe, które są dostępne oprócz informacji o cenie. Wykorzystuje wielkość papieru wartościowego, która w przypadku akcji jest liczbą akcji w obrocie na akcje w określonym przedziale czasu. Wykorzystuje również otwarte zainteresowanie, które jest wartością całkowitej liczby otwartych transakcji w określonym czasie. W przypadku kontraktów terminowych na towary jest to liczba otwartych pozycji krótkich i długich. Niniejsze badanie próbuje zmierzyć przepływ pieniędzy na rynek i z rynku. Wzór na to jest następujący (zauważ, że tick to najmniejszy dopuszczalny ruch na danym rynku):

Let MP = mean price over a particular interval
 OI = the larger of yesterday's or today's open interest

wtedy

$$K = [(MP_t - MP_{t-1}) \times \text{dollar value of 1 tick move} \times \text{volume}] \times [1 \pm 2/OI]$$

$$HPI_t = HPI_{t-1} + [0.1 \times (K - HPI_{t-1})] / 100,000$$

MACD

Wskaźnik MACD (ang. Moving Average Convergence Divergence) jest różnicą między dwiema średnimi ruchomymi i informuje, kiedy na rynku występują krótkoterminowe warunki wykupienia lub wyprzedania. Wzór wygląda następująco:

Let OSC = EMA1 - EMA2,
 where EMA1 is for one smoothing constant and time period, for example
 0.15 and 12 weeks
 EMA2 is for another smoothing constant and time period,
 for example
 0.075 and 26 weeks

wtedy

$$MACD_t = MACD_{t-1} + K \times (OSC_t - MACD_{t-1})$$

where K is a smoothing constant, for example, 0.2

Ostateczna formuła skutecznie wykonuje kolejne wykładnicze wygładzanie różnicy dwóch średnich ruchomych, na przykład w okresie 9 tygodni.

„Stochastyka”

Ten wskaźnik nie ma absolutnie nic wspólnego z procesami stochastycznymi. Powód nazwy jest tajemnicą, ale wskaźnik składa się z dwóch części: %K i %D, która jest średnią ruchomą %K. Skrzyżowanie tych linii wskazuje obszary wykupienia i wyprzedania. Formuły są następujące:

$$\text{Raw \%K} = 100 \times (P - \text{LowX}) / (\text{HighX} - \text{LowX})$$

$$\%K_t = [(\%K_{t-1} \times 2) + \text{Raw \%K}_t] / 3$$

$$\%D_t = [(\%D_{t-1} \times 2) + \%K_t] / 3$$

Objętość bilansowa

Wskaźnik wolumenu bilansowego (OBV) został stworzony, aby spróbować odkryć wzorce akumulacji i dystrybucji dużego gracza na giełdzie. Jest to skumulowana suma danych dotyczących wolumenu, określona w następujący sposób:

Jeśli dzisiejsze zamknięcie jest większe niż wczorajsze zamknięcie

$$OBV_t = OBV_{t-1} + 1$$

Jeśli dzisiejsze zamknięcie jest mniejsze niż wczorajsze

$$OBV_t = OBV_{t-1} - 1$$

Wartość bezwzględna indeksu nie jest ważna; zwraca się uwagę tylko na kierunek i trend.

Akumulacja-Dystrybucja

Ten wskaźnik robi dla ceny to, co OBV dla wolumenu.

Jeśli dzisiejsze zamknięcie jest większe niż wczorajsze zamknięcie:

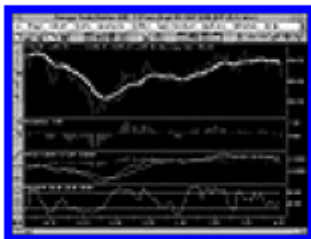
$$AD_t = AD_{t-1} + (Close_t - Low_t)$$

Jeśli dzisiejsze zamknięcie jest mniejsze niż wczorajsze

$$AD_t = AD_{t-1} + (High_t - Close_t)$$

Przyjrzyjmy się teraz, jak wyglądają te wskaźniki. Rysunek 14.7 przedstawia wykres słupkowy, który jest wykresem danych cenowych w funkcji czasu, wraz z następującymi wskaźnikami:

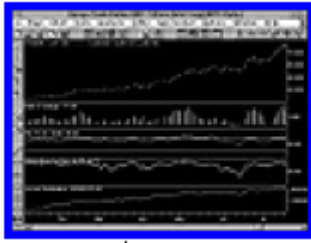
- Dziesięciojednostkowa średnia krocząca
- Dziesięciojednostkowa wykładnicza średnia krocząca
- Pęd
- MACD
- Procent R



Przedstawiony okres to 5-minutowe słupki dla kontraktu futures S&P 500 z września 1995 roku. Górna część każdego paska wskazuje najwyższą wartość („wysoką”) dla tego przedziału czasu, dolna wskazuje najniższą wartość („niska”), a poziome linie na pasku wskazują początkową („otwartą”) i końcową („zamkniętą”) wartości dla przedziału czasu. Rysunek przedstawia kolejny wykres słupkowy akcji Intel Corporation w okresie od grudnia 1994 do lipca 1995, gdzie każdy słupek reprezentuje dzień aktywności. Wyświetlane są również następujące wskaźniki.

- Tempo zmian
- Siła względna

- Stochastyka
- Akumulacja-Dystrybucja



Widziałeś kilka z setek wskaźników technicznych, które do tej pory wymyślono. Nowe wskaźniki powstają szybko, ponieważ dziedzina analizy technicznej zyskuje na popularności i naśladowaniu. Istnieją również badania rozpoznawania wzorców, takie jak formacje przypominające flagi lub proporczyki, a także bardziej egzotyczne rodzaje badań, takie jak liczenie fal Elliota. Więcej informacji na temat tych i innych badań można znaleźć w książkach poświęconych analizie technicznej (np. Murphy). Wstępne przetwarzanie neuronowe za pomocą narzędzi analizy technicznej, a także tradycyjnych narzędzi analizy inżynierskiej, takich jak szeregi Fouriera, falki i fraktale, może być bardzo przydatne w znajdowaniu wzorców predykcyjnych do prognozowania.

Co zgłosili inni

W tej końcowej części przedstawiamy kilka studiów przypadków udokumentowanych w czasopiśmie i książkach, aby dać wyobrażenie o dotychczasowych sukcesach lub niepowodzeniach sieci neuronowych w prognozowaniu finansowym. Należy pamiętać, że najlepsze (= najbardziej opłacalne) wyniki zwykle nigdy nie są zgłaszane (aby nie stracić przewagi konkurencyjnej)! Pamiętaj też, że nieefektywność rynku, którą wykorzystano wczoraj, może już nie być taka sama, jak dziś.

Czy trzylatek może handlować towarami?

Cóż, Hillary Clinton z pewnością może handlować towarami, ale trzylatka też? W swoim artykule „Handel towarami z trzylatką” J.E. Collard opisuje sieć neuronową z rzekomą inteligencją trzylatki. Aplikacja wykorzystywała sieć ze sprzężeniem zwrotnym z propagacją wsteczną o architekturze 37-30-1. Sieć została przeszkolona, aby kupować („zajmować pozycję długą”) lub sprzedawać („zajmować pozycję krótką”) na rynku kontraktów terminowych na żywe bydło. Zestaw szkoleniowy składał się z 789 faktów dla dni handlowych w latach 1988, 1989, 1990 i 1991. Każdy wektor wejściowy składał się z 18 podstawowych wskaźników i sześciu rynkowych zmiennych technicznych (otwarte, wysokie, niskie, zamknięte, otwarte zainteresowanie, wolumen). Sieć mogłaby zostać przeszkolona w celu uzyskania prawidłowych wyników dla wszystkich z wyjątkiem 11 z 789 faktów. W pełni wyszkolona sieć została wykorzystana przez 178 kolejnych dni handlowych w 1991 roku. Skumulowany zysk wzrósł w tym okresie z 0 USD do 1547,50 USD dzięki wymianie jednego kontraktu na żywe bydło. Największa strata w transakcji wyniosła 601,74 USD, a największy zysk w transakcji 648,30 USD.

Prognozowanie rentowności bonów skarbowych i obligacji skarbowych

Milam Aiken zaprojektował sieć propagacji wstecznej ze sprzężeniem do przodu, która przewidywała stawki bonów skarbowych i porównywała uzyskaną przez niego prognozę z prognozami czołowych amerykańskich ekonomistów. Wyniki pokazały, że sieć neuronowa, biorąc pod uwagę te same dane, wykonała lepsze prognozy (błąd bezwzględny 0,18 w porównaniu z 0,71). Aiken wykorzystał 250 serii danych ekonomicznych, aby zobaczyć korelację z rachunkami skarbowymi i wykorzystał tylko te serie, które wykazały wiodącą korelację: indeks wiodących wskaźników ekonomicznych Departamentu

Handlu, krótki indeks kompozytowy Centrum Badań nad Międzynarodowym Cyklem Biznesowym (CIBCR) oraz CIBCR Long Leading Composite Index. Wcześniejsze dane dla tych trzech wskaźników za ostatnie cztery lata (łącznie 12 wejść) wykorzystano do przewidzenia średniej rocznej stopy bonów skarbowych (jeden wynik) na bieżący rok. Guido Deboeck i Masud Cader zaprojektowali rentowne systemy transakcyjne dla dwuletnich i dziesięcioletnich skarbowych papierów wartościowych. Wykorzystali sieci neuronowe ze sprzężeniem do przodu z algorytmem uczenia zwanym rozszerzoną delta-bar-delta (EDBD), który jest wariantem propagacji wstecznej. Próbkę szkoleniową składającą się ze 100 faktów zostały wybrane z 1120 dni handlowych w okresie od 1 lipca 1989 do 30 czerwca 1992. Okres testowy obejmował ponad 150 dni handlowych od 1 lipca 1992 do 30 grudnia 1992. Wydajność zestawu testowego była monitorowana co N tysięcy cykli treningowych, a procedura treningowa była zatrzymywana, gdy wydajność na zestawie testowym uległa pogorszeniu. (Jest to ta sama procedura, którą zastosowaliśmy przy opracowywaniu modelu dla S&P 500.) Kryterium używanym do oceny wydajności modelu był stosunek średniego zysku do maksymalnej wartości wypłaty, która jest największą niezrealizowaną stratą odnotowaną w okresie handlu. Portfel oddzielnych zaprojektowanych systemów transakcyjnych dla dwuletnich i dziesięcioletnich papierów wartościowych dał następujące wyniki: W okresie 4,5 roku portfel zawierał łącznie 133 transakcje z 65% zyskownymi transakcjami i maksymalnym wykorzystaniem 64 punktów bazowych, czyli tysięcy jednostek dla rentowności obligacji. Całkowity zysk wyniósł w tym okresie 677 punktów bazowych, przy czym maksymalny zysk w jednej transakcji wyniósł 52 punkty bazowe, a maksymalna strata w jednej transakcji 47 punktów bazowych.

Stabilność i niezawodność tego systemu sprawdzono przy użyciu ponad 1000 ruchomych okien czasowych o czasie trwania 3-miesięcznym, 6-miesięcznym i 12-miesięcznym w przedziale 4,5 roku oraz odnotowując odchylenia standardowe w zyskach i maksymalnym wykorzystaniu środków. Maksymalny spadek wahał się od 30 do 48 punktów bazowych. Sieci neuronowe a prognozowanie szeregów czasowych Boxa-Jenkinsa Ramesh Sharda i Rajendra Patil użyli standardowej sieci propagacji wstecznej ze sprzężeniem do przodu 12-12-1 i porównali wyniki z metodologią Box-Jenkinsa do prognozowania szeregów czasowych. Prognozowanie Box-Jenkinsa jest tradycyjną techniką prognozowania szeregów czasowych. Do oceny autorzy wykorzystali 75 różnych szeregów czasowych. Wyniki pokazały, że sieci neuronowe osiągnęły lepsze MAPE (średni bezwzględny błąd procentowy) ze średnią dla wszystkich 75 MAPE szeregów czasowych wynoszącą 14,67 w porównaniu z 15,94 dla podejścia Box-Jenkinsa.

Sieci neuronowe a analiza regresji

Leorey Marquez i inni porównał modelowanie sieci neuronowych ze standardową analizą regresji. Autorzy zastosowali sieć ze sprzężeniem do przodu z propagacją wsteczną o strukturze 1-6-1. Wykorzystali trzy formy funkcjonalne znalezione w analizie regresji:

1. $Y = B_0 + B_1 X + e$

2. $Y = B_0 + B_1 \log(X) + e$

3. $Y = B_0 + B_1/X + e$

Dla każdego z tych formularzy wygenerowano 100 par danych (x,y) dla tego „prawdziwego” modelu. Teraz sieć neuronowa została wytrenowana na tych 100 parach danych. Sieć wygenerowała dodatkowe 100 punktów danych w celu przetestowania zdolności prognozowania sieci. Wyniki pokazały, że sieć neuronowa osiągnęła MAPE w granicach 0,6% prawdziwego modelu, co jest wynikiem bardzo dobrym. Model sieci neuronowej najlepiej przybliżył model liniowy. Przeprowadzono również eksperyment z celową błędną specyfikacją niektórych punktów danych. Model sieci neuronowej

również wypadł dobrze w tych przypadkach, ale porównywalnie gorzej w przypadku modelu odwrotnego.

Hierarchiczna sieć neuronowa

Mendelsohn opracował wielopoziomową sieć neuronową, jak pokazano na rysunku. Tutaj pięć sieci neuronowych jest rozmieszczonych w taki sposób, że cztery wyjścia sieciowe zasilają tę końcową sieć. Cztery sieci są przeszkolone do generowania trendu wysokiego, niskiego, krótkoterminowego i średnioterminowego dla konkretnego instrumentu finansowego. Ostateczna sieć bierze te cztery wyjścia jako dane wejściowe i tworzy wskaźnik punktu zwrotnego.



Każda sieć została przeszkolona i przetestowana przez 1200 dni w okresie od 1988 do 1992 roku (33% wykorzystane do testów). Wstępne przetwarzanie zostało zrealizowane przy użyciu różnic danych wejściowych i pewnych technicznych analiz analitycznych:

- Średnie kroczące
- Wykładnicze średnie kroczące
- Wskaźniki stochastyczne

W przypadku sieci, która generuje przewidywaną wysoką wartość, średni błąd wahał się od 7,04% do 7,65% dla różnych rynków finansowych w okresie testowym, w tym kontraktów terminowych na obligacje skarbowe, eurodolara, jena japońskiego i S&P 500.

Metodologia przewidywania rynku „walk-forward”

Metodologia, która jest czasami stosowana w projektowaniu sieci neuronowych, to szkolenie i testowanie typu walk-forward. Oznacza to, że wybierasz przedział czasu (np. sześć miesięcy), w którym trenujesz sieć neuronową i testujesz sieć w kolejnym przedziale. Następnie przesuwasz okno treningowe i testowe do przodu, na przykład o jeden miesiąc, i powtarzasz ćwiczenie. Robisz to przez interesujący Cię okres, aby zobaczyć wyniki prognozy. Zaletą tego podejścia jest maksymalizacja zdolności sieci do modelowania niedawnej przeszłości podczas prognozowania. Wadą jest to, że sieć zapomina o cechach rynku, które miały miejsce przed oknem szkoleniowym. Takashi Kimoto i in. wykorzystali metodologię walk forward w projektowaniu systemu transakcyjnego dla Fujitsu i Nikko Papiery Wartościowe. Oni również, podobnie jak Mendelsohn, używają hierarchicznej sieci neuronowej złożonej z indywidualnych sieci neuronowych ze sprzężeniem do przodu. Prognozę TOPIX, która jest japońskim odpowiednikiem Dow Jones Industrial Average, została wykonana przez 33 miesiące od stycznia 1987 do września 1980. Na pierwszym poziomie hierarchii wykorzystano cztery sieci wyszkolone na danych cenowych i danych ekonomicznych. Wyniki zostały przekazane do końcowej sieci, która generowała sygnały kupna i sprzedaży. Wydajność systemu transakcyjnego osiągnęła wynik o 20% lepszy niż strategia kup i trzymaj dla TOPIX.

System transakcyjny z podwójnym potwierdzeniem

Jeremy Konstenius omawia system transakcyjny dla indeksu S&P 400 z holograficzną siecią neuronową, która różni się od sieci neuronowej ze sprzężeniem do przodu z propagacją wsteczną. Sieć holograficzna wykorzystuje liczby zespolone do wprowadzania i wyprowadzania danych z neuronów,

które są matematycznie bardziej złożone niż neurony sieci ze sprzężeniem do przodu. Autor wykorzystuje dwie wytrenowane sieci do prognozowania kierunku na następny dzień na podstawie danych z ostatnich 10 dni. Każda sieć korzysta z danych wejściowych, które są hamowane, odejmując średnią ruchomą od danych. Sieć 1 używa detrendowanych wartości zamknięcia. Sieć 2 używa detrended High wartości. Jeśli obie sieci zgadzają się lub potwierdzają się nawzajem, następuje wymiana. Inaczej nie ma handlu. Sieć 1 wykazała dokładność 61,9% dla pięciomiesięcznego okresu testowego (okres szkolenia obejmował dwa lata przed okresem testowym), podczas gdy Sieć 2 również wykazała dokładność 61,9%. Łącząc obie sieci, Konstenius osiągnął dokładność 65,82%.

Predyktor punktu zwrotnego

To podejście do sieci neuronowych zostało omówione przez Michitaka Kosaka i in. (1991). Omawiają zastosowanie sieci propagacji wstecznej ze sprzężeniem do przodu do opracowania sygnałów kupna/sprzedaży papierów wartościowych. Zbierałbyś szeregi czasowe dotyczące cen akcji i chciałbyś znaleźć trendy w danych, aby zmiany kierunku trendu dostarczały ci punktów zwrotnych, które interpretujesz jako sygnały kupna lub sprzedaży. Musisz wymienić te czynniki, które Twoim zdaniem mają jakikolwiek wpływ na cenę badanego papieru wartościowego. Musisz również określić, w jaki sposób mierzysz te czynniki. Następnie formułujesz nieliniową funkcję łączącą czynniki z Twojej listy i przeszłe, niezależnie od wielu cen Twojego papieru wartościowego (twoje szeregi czasowe). Funkcja ma postać, jak Michitaka Kosaka, et al. (1991) ujął to,

```
p(t + h) = F(x(t), x(t -1), ... , f1, f2, ... )
where
f1, f2, represent factors on your list,
x(t) is the price of your stock at time t,
p(t + h) is the turning point of security price at time t + h, and
p(t + h) = -1 for a turn from downward to upward,
p(t + h) = +1 for a turn from upward to downward,
p(t + h) = 0 for no change and therefore no turn
```

Tutaj zmieniasz h przez wartości 1, 2 itd., przechodząc w przyszłość o jeden dzień (okres) na raz. Zauważ, że szczegółowa postać funkcji F nie jest podana. To jest do skonfigurowania według własnego uznania.

Możesz ustawić podobną funkcję dla x(t + h), ceny akcji w czasie t + h i mieć oddzielną sieć obliczającą ją przy użyciu paradygmatu propagacji wstecznej. Będziesz wtedy generował przyszłe ceny akcji i przyszłe sygnały kupna/sprzedaży ręka w rękę, ale równolegle. Michitaka Kosaka i in. (1991) donoszą, że wykorzystali dane szeregów czasowych z pięciu lat do identyfikacji modelu sieci oraz dane szeregów czasowych z okresu jednego roku do oceny wydajności prognozowania modelu, ze wskaźnikiem sukcesu 65% dla punktów zwrotnych.

Prognozy S&P 500 i plamy słoneczne

Michael Azoff w swojej książce o prognozowaniu szeregów czasowych za pomocą sieci neuronowych (patrz referencje) tworzy systemy sieci neuronowych do przewidywania indeksu S&P 500, a także do przewidywania chaotycznych szeregów czasowych, takich jak występowanie plam słonecznych. Azoff wykorzystuje sieci propagacji wstecznej ze sprzężeniem do przodu, z algorytmem treningowym zwanym adaptacyjnym, stromym zejściem, odmianą standardowego algorytmu. Dla szeregów czasowych plam słonecznych, architektury 6-5-1 i stosunku wektorów uczących do trenowalnych wag wynoszącym 5,1 osiąga błąd zestawu uczącego 12,9% i błąd zestawu testowego 21,4%. Seria ta składała się z rocznych liczb plam słonecznych z lat 1706-1914. Sześć lat kolejnych danych rocznych zostało

wprowadzonych do sieci. Jedną z sieci wykorzystanych przez Azoff do prognozowania indeksu S&P 500 była sieć 17-7-1. Stosunek wektorów treningowych do trenowalnych ciężarów wynosił 6,1. Osiągnięty błąd zestawu treningowego wyniósł 3,29%, a błąd zestawu testowego 4,67%. Dane wejściowe do tej sieci obejmowały dane cenowe, wskaźnik zmienności, który jest funkcją zakresu ruchu cen, oraz wskaźnik błędzenia losowego, badanie analizy technicznej. Krytyka prognozowania szeregów czasowych sieci neuronowych dla handlu Michael de la Maza i Deniz Yuret, menedżerowie Redfire Capital Management Group, sugerują, że miarą powinien być zwrot skorygowany o ryzyko, a nie błąd średniej kwadratowej do optymalizacji w aplikacji sieci neuronowej do handlu. Wskazują również, że w przypadku sieci neuronowych, podobnie jak w przypadku metod statystycznych, takich jak regresja liniowa, faktów danych, które wydają się niewytłumaczalne, nie można ignorować, nawet jeśli chcesz, aby tak było. Nie ma odpowiednika warunku „nie obchodzi” dla wyjścia sieci neuronowej. Ten rodzaj warunku może być ważną opcją dla środowisk handlowych, które nie mają „wykrywalnej regularności”, jak to ujęli autorzy, i dlatego tak naprawdę nie można nimi handlować. Niektóre rozwiązania dwóch postawionych problemów podano w następujący sposób:

- Użyj algorytmu innego niż wsteczna propagacja, który pozwala na maksymalizację zwrotu skorygowanego o ryzyko, takiego jak symulowane wyzarczenie lub algorytmy genetyczne.
- Przekształć dane wprowadzane do sieci tak, aby minimalizacja błędu średniej kwadratowej stała się równoznaczną z maksymalizacją zwrotu skorygowanego o ryzyko.
- Użyj hierarchii (patrz hierarchiczna sieć neuronowa wcześniej w tej sekcji) sieci neuronowych, przy czym każda sieć jest odpowiedzialna za wykrywanie cech lub prawidłowości jednego składnika danych.

Podsumowanie

W tej Części przedstawiono zastosowanie sieci neuronowych w prognozowaniu finansowym. Jako przykład kroków potrzebnych do opracowania modelu prognozowania sieci neuronowych, zmiana indeksu giełdowego Standard & Poor's 500 została przewidziana 10 tygodni na podstawie danych tygodniowych dla pięciu wskaźników. Pokazano kilka przykładów preprocessingu danych dla sieci oraz zagadnienia w szkoleniu. Pod koniec okresu uczącego zaobserwowano, że zachodzi zapamiętywanie, ponieważ błąd w danych testowych uległ pogorszeniu, natomiast poprawił się błąd w zbiorze uczącym. Ważne jest, aby monitorować błąd w danych testowych (bez zmian masy ciała) podczas treningu, aby zapewnić utrzymanie zdolności uogólniania. Ostateczna sieć dała średni błąd RMS 6,9% w zestawie uczącym i 13,9% w zestawie testowym. Przykład prognozowania przedstawiony w tym rozdziale podkreśla łatwość użycia i szerokie zastosowanie algorytmu wstecznej propagacji błędów w przypadku dużych, złożonych problemów i zbiorów danych. Przedstawiono kilka przykładów badań w zakresie prognozowania finansowego, a także przedstawiono szereg pomysłów i rzeczywistych metodologii. Analiza techniczna została pokrótce omówiona z przykładami badań, które mogą być przydatne we wstępnym przetwarzaniu danych dla sieci neuronowych.