

Podstawy sztucznej inteligencji

wykład 5

Sztuczne sieci neuronowe (SSN)

Joanna Kołodziejczyk

28 listopad 2012

Plan wykładu

- 1 Biologiczne wzorce sztucznej sieci neuronowej
- 2 Sztuczny neuron
- 3 Perceptron
- 4 Architektury jednokierunkowych sztucznych sieci neuronowych
- 5 Przykłady zastosowania SSN w biznesie

Neuron biologiczny

Neuron

Jest podstawowym budulcem układu nerwowego. Jest komórką, która jest w stanie odbierać i przekazywać sygnały elektryczne.

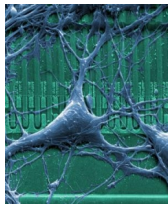
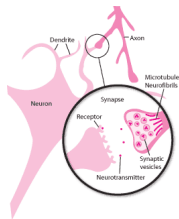
Neuron działanie

Jeżeli wartość sygnału elektrycznego przekazywana do neuronu przekracza pewną wartość, neuron ulega depolaryzacji, powodując pobudzenie komórki nerwowej. Pobudzenie polega na wyładowaniu neuronu, a powstały sygnał przesyłany jest do innych neuronów. Siła wyładowania neuronu jest niezależna od bieżącej depolaryzacji, siły, która go spowodowała. Dopóki prąd przekracza określony próg, wielkość wyładowanie pozostanie taka sama (w danym neuronie).

Budowa neuronu biologicznego

Neuron składa się z:

- 1 Wielu dendrytów, których celem jest pobieranie impulsów od innych neuronów.
- 2 Ciała komórki z jądrem.
- 3 Jednego aksonu, który przekazuje wyładowanie do kolejnych komórek.
- 4 Synaps – neuroprzekaźników osłabiających lub wzmacniających sygnał wyjściowy.



Plan wykładu

- 1 Biologiczne wzorce sztucznej sieci neuronowej
- 2 Sztuczny neuron**
- 3 Perceptron
- 4 Architektury jednokierunkowych sztucznych sieci neuronowych
- 5 Przykłady zastosowania SSN w biznesie

Sztuczna sieć neuronowa

Sztuczna sieć neuronowa

Sztuczne sieci neuronowe (SSN) łączą w sobie małe procesory (jednostki przetwarzania), a każdy naśladuje funkcje pojedynczego neuronu biologicznego. W ten sposób tworzą sztuczny „mózg” do przetwarzania informacji. Dzięki połączeniu wielu takich sztucznych neuronów, z których każdy posiada jedno wyjście, SSN może wykonywać złożone funkcje.

Sztuczna neuron

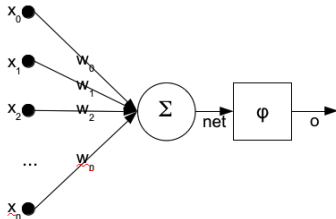
Każdy procesor, podobnie jak neuron, ma poziom progowy, który musi być przekroczony nim nastąpi transmisja sygnału. Binarny charakter procesora można porównać do włącznika światła.

Sztuczny Neuron

Działanie neuronu

$$o = \varphi(\text{net}) = \varphi\left(\sum_{i=0}^n x_i * w_i\right)$$

gdzie x_i — wejścia i $x_0 = 1$, w_i — wagi, φ — funkcja aktywacji, net — pobudzenie neuronu, o — wyjście z neuronu



Cechy neuronu






- Wejścia i wagi są liczbami rzeczywistymi dodatnimi i ujemnymi.
- Jeżeli jakaś cecha (wejście) powoduje odpalenie neuronu, waga będzie dodatnia, a jeżeli cecha ma działanie hamujące to waga jest ujemna.
- Funkcję aktywacji dobiera się do rozwiązywanego zadania.
- Wagi są dopasowywane przez pewną regułę uczenia tak, by zależność wejście/wyjście w neuronie spełniało pewien określony cel.

Stosowane w SSN funkcje aktywacji

Name	Input/Output Relation	Icon
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$	
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	
Linear	$a = n$	
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$	

3

Stosowane w SSN funkcje aktywacji

Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$	
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$	
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$	

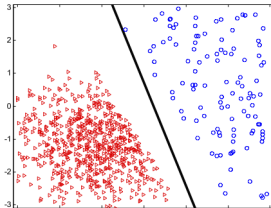
4

Plan wykładu

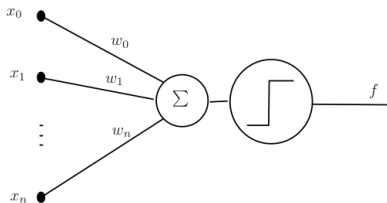
- 1 Biologiczne wzorce sztucznej sieci neuronowej
- 2 Sztuczny neuron
- 3 Perceptron**
- 4 Architektury jednokierunkowych sztucznych sieci neuronowych
- 5 Przykłady zastosowania SSN w biznesie

Perceptron prosty (Rosenblatt 1962)

Perceptron prosty jest najprostszą SSN, używaną do klasyfikacji binarnej. Perceptron prosty składa się z pojedynczego neuronu z regulacją wag. Rosenblatt zaproponowała twierdzenie zbieżności perceptronu, stwierdzając, że konwergencja jest gwarantowana wtedy i tylko wtedy, gdy wzorce wykorzystywane do uczenia perceptronu tworzą dwie separowalne klasy. Powierzchnia decyzyjna zostanie umieszczona w formie hiperpłaszczyzny gdzieś pomiędzy tymi dwiema grupami.



Perceptron — model



Funkcja aktywacji

$$f(\langle \mathbf{w}, \mathbf{x} \rangle) = \begin{cases} 1 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle > 0; \\ 0 & \text{if } \langle \mathbf{w}, \mathbf{x} \rangle \leq 0. \end{cases}$$

Zadanie klasyfikacji binarnej

Klasyfikacja binarna

Klasyfikacja polega na przydzieleniu obiektu do pewnej klasy na podstawie jego atrybutów (danych wejściowych). W klasyfikacji binarnej wyjściem jest dwupunktowe wyjście (możliwe dwie klasy na wyjściu).

Niech dany będzie zbiór uczący (zbiór par) (\mathbf{x}_i, y_i) $i = 1, \dots, l$, gdzie $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in}) \in \mathbb{R}^n$ są danymi wejściowymi a $y_i \in \{0, 1\}$ są skojarzonymi z nimi klasami (wyjściami).

Płaszczyzna separacji klas

Równanie płaszczyzny w przestrzeni \mathbb{R}^n

$$w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n = 0$$

czyli należy znaleźć taki zestaw $w_0, w_1, w_2, \dots, w_n$, że

$$\forall_{i,y_i=1} w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0$$

oraz

$$\forall_{i,y_i=0} w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n \leq 0,$$

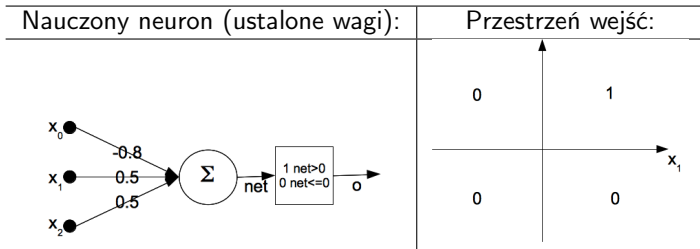
gdy dodamy jeszcze jeden punkt do wejść $x_i = (1, x_{i1}, x_{i2}, \dots, x_{in})$,
to równanie płaszczyzny:

$$\langle \mathbf{w}, \mathbf{x}_i \rangle = \sum_{j=0}^n w_j x_{ij}$$

Przykład dwuwymiarowy

Dane uczące:

x_1	x_2	y
1	1	1
1	0	0
0	1	0
0	0	0



Uczenie SSN

Uczenie

Uczenie polega na automatycznym doborze wag w SSN, na podstawie zbioru przykładów nazwanym zbiorem uczącym. Zaczyna się z losowymi małymi wagami i iteracyjnie zmienia się wagi, dopóki wszystkie przykłady uczące nie zostaną poprawnie zaklasyfikowane (lub z niewielkim błędem).

Wyróżnia się dwa typy uczenia, zależy od wykorzystanej sieci i przykładów (zbioru uczącego):

- 1 nadzorowane (z nauczycielem), gdy w zbiorze danych do każdej próbki podana jest poprawna klasa
- 2 nienadzorowane (bez nauczyciela), zbiór danych nie ma wektora odpowiedzi.

Algorytm uczenia perceptronu

- 1 Niech $\mathbf{w}(0) = (0, \dots, 0)$ lub wartości losowe z przedziału $[-1, 1]$, $k = 0$
- 2 Dopóki zbiór punktów uczących pozostaje błędnie klasyfikowany tj. zbiór $A = \{\mathbf{x}_i : y_i \neq f(\langle \mathbf{w}, \mathbf{x}_i \rangle)\}$ pozostaje niepusty, powtarzaj:
 - 1 Wylosuj ze zbioru A dowolny punkt
 - 2 Aktualizuj wagi według następującej reguły:

$$\mathbf{w}(k + 1) = \mathbf{w}(k) + \eta e \mathbf{x}_i$$

- 3 $k = k + 1$

Gdzie η współczynnik uczenia $\eta \in (0, 1]$. A e jest wartością błędu popełnianego na prezentowanej próbce.

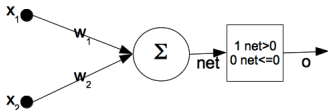
Analiza algorytmu na przykładzie testowym

Przykład pochodzi z książki „Neural Network Design”, M. T. Hagan, H. B. Demuth, M. H. Beale.

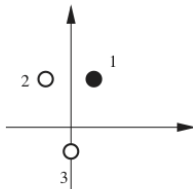
Zbiór uczący

$$\{x_1 = (1, 2), y_1 = 1\} \quad \{x_2 = (-1, 2), y_2 = 0\}$$
$$\{x_3 = (0, -1), y_3 = 0\}$$

Architektura sieci:



Przestrzeń wejść:



Konsekwencje braku wejścia x_0

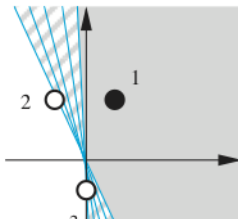
Usunięcie dodatkowego wejścia powoduje, że płaszczyzna decyzyjna musi przejść przez początek układu współrzędnych, bo:

$$w_1x_1 + w_2x_2 = 0$$

to równanie prostej przechodzącej przez punkt $(0,0)$.

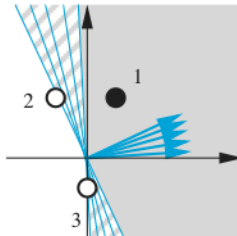
Trzeba być pewnym, że taka sieć rozwiąże problem separacji punktu x_1 od punktów x_2 i x_3 .

Jak widać rozwiązań jest nieskończenie wiele.



Cel uczenia

Rysunek pokazuje wektory wag, które odpowiadają granicom decyzyjnym. Wektor jest prostopadły do granicy decyzyjnej. Reguła ucząca ma znaleźć wektor wag, który wskazuje w jednym ze wskazanych kierunków. Długość wektora wag nie ma znaczenia, ważny jest tylko kierunek.

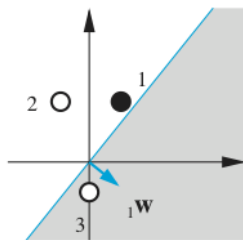


Ustalenie wartości początkowych wag

Zakłada się losowo ustalenie wartości wag początkowych i tak:

$$\mathbf{w}(0) = (1, -0.8)$$

Na wykresie zaprezentowano początkowy wektor wag i granicę decyzyjną dla takich wag.



Prezentacja pierwszego punktu ze zbioru uczącego

Pierwszym prezentowanym punktem jest x_1 .

$$f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = f(1 \cdot 1 + (-0.8) \cdot 2) = f(-0.6) = 0$$

Wiemy, że $y_1 = 1$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu x_1 .

Jak widać na rysunku z poprzedniego slajdu granica decyzyjna jest nieprawidłowo położona w stosunku do punktu uczącego. Powinno się zatem przesunąć wektor wag nieznacznie w kierunku punktu x_1 . Można tego dokonać dodając do wektora wag wektor x_i .

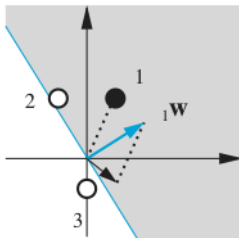
Wnioski co do poprawki wag

Reguła korekty wag (1)

Jeżeli $y_i = 1$ i $f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = 0$ to

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}_i$$

Wagi po korekcie: $\mathbf{w}(1) = (1 + 1, -0.8 + 2) = (2, 1.2)$



Prezentacja drugiego punktu ze zbioru uczącego

Drugim prezentowanym punktem jest \mathbf{x}_2 .

$$f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = f(2 \cdot (-1) + 1.2 \cdot 2) = f(0.4) = 1$$

Wiemy, że $y_2 = 0$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu \mathbf{x}_2 .

Jak widać na rysunku z poprzedniego slajdu granica decyzyjna jest nieprawidłowo położona w stosunku do punktu uczącego. Powinno się zatem odsunąć wektor wag nieznacznie od punktu \mathbf{x}_2 .

Można tego dokonać odejmując do wektora wag wektor \mathbf{x}_i .

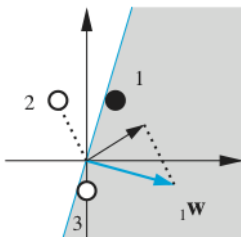
Wnioski co do poprawki wag

Reguła korekty wag (2)

Jeżeli $y_i = 0$ i $f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = 1$ to

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{x}_i$$

Wagi po korekcie: $\mathbf{w}(2) = (2 - (-1), 1.2 - 2) = (3, -0.8)$



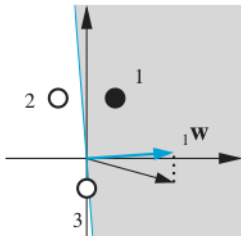
Prezentacja trzeciego punktu ze zbioru uczącego

Trzecim prezentowanym punktem jest x_3 .

$$f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = f(3 \cdot 0 + (-0.8) \cdot (-1)) = f(0.8) = 1$$

Wiemy, że $y_2 = 0$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu x_2 . Stosując regułę 2 poprawia się wagi:

$$\mathbf{w}(3) = (3 - 0, (-0.8) - (-1)) = (3, 0.2)$$



Kolejna seria prezentacji wzorców uczących

Prezentując ponownie wszystkie punkty uczące uzyskujemy bezbłędne sklasyfikowanie wszystkich 3 punktów. Oznacza to, że wagi $\mathbf{w} = (3, 0.2)$ są rozwiązaniem dla uczonego klasyfikatora.

Korekty wag

- 1 Jeżeli $y_i = 1$ i $f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = 0$ to $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}_i$
- 2 Jeżeli $y_i = 0$ i $f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = 1$ to $\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{x}_i$
- 3 Jeżeli $y_i = f(\langle \mathbf{w}, \mathbf{x}_i \rangle)$ to $\mathbf{w}(k+1) = \mathbf{w}(k)$

Reguła uczenia perceptronu

Potrójna reguła na korektę wag może być zapisana w postaci pojedynczego równania.

Niech błąd popełniany przez sieć neuronową:

$$e = y_i - f(\langle \mathbf{w}, \mathbf{x}_i \rangle)$$

- 1 Jeżeli $e = 1$ to $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}_i$
- 2 Jeżeli $e = -1$ to $\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{x}_i$
- 3 Jeżeli $e = 0$ to $\mathbf{w}(k+1) = \mathbf{w}(k)$

Korekta regułą perceptronu (delta)

$$\mathbf{w}(k+1) = \mathbf{w}(k) + e\mathbf{x}_i$$

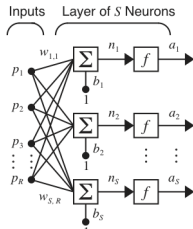
Plan wykładu

- 1 Biologiczne wzorce sztucznej sieci neuronowej
- 2 Sztuczny neuron
- 3 Perceptron
- 4 Architektury jednokierunkowych sztucznych sieci neuronowych**
- 5 Przykłady zastosowania SSN w biznesie

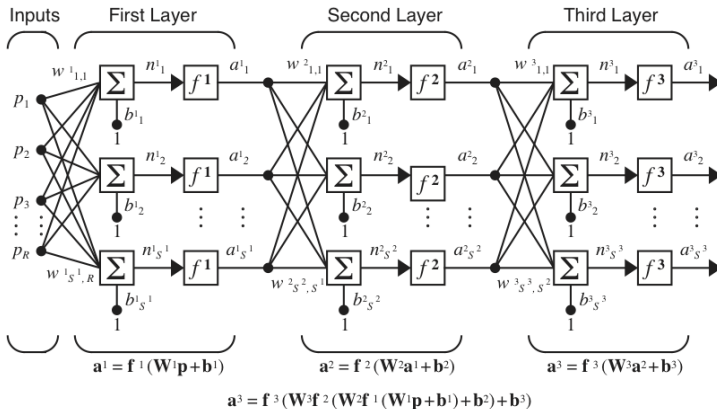
Perceptron prosty jednowarstwowy

Pojedyncza warstwa neuronów (Single layer perceptron) - cechy

- 1 Pewna liczba neuronów S jest ułożona w warstwie i wszystkie wejścia R sieci zasilają wszystkie neurony. $S \neq R$.
- 2 Wyjście z sieci jest wektorem. Wagi są macierzą $S \times R$.
- 3 Umożliwia klasyfikację do większej liczby klas.
- 4 Zazwyczaj wszystkie neurony mają tę samą funkcję aktywacji.



Sieć wielowarstwowa MLP

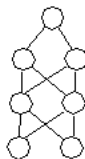
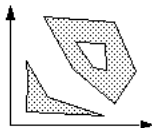
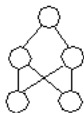
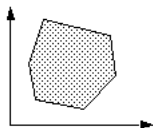
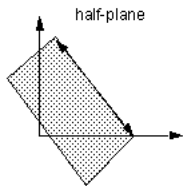


Cechy sieci MLP

Wielowarstwowy perceptron

- 1 Ostatnia warstwa nazywana jest wyjściową, a wewnętrzne warstwami ukrytymi.
- 2 MLP ma dużo większe zastosowania niż pojedyncza warstwa.
- 3 Liczba wejść i neuronów wyjściowych jest określona przez rozwiązywane zadanie (zbiór danych wejściowych).
- 4 Liczba warstw ukrytych powinna być ograniczona do jednej lub dwóch. Wyjaśnienie na kolejnym slajdzie.
- 5 Nie ma wskazania ile powinno być neuronów w każdej warstwie. Źle dobrana ta liczba będzie skutkować złym dopasowaniem (nadmiernym lub za słabym) sieci do danych uczących.

Graficzna prezentacja możliwości separowania klas przez MLP



Dane uczące dla SSN

- 1 Należy zebrać dane niezbędne w procesie uczenia. Mogą pochodzić z baz danych.
- 2 Wybrać zmienne mające znaczenia dla rozwiązywanego zadania. Zazwyczaj na początku przyjmuje się wszystkie a potem dokonuje się redukcji.
- 3 Zebrana liczba rekordów powinna być reprezentatywna. Powinno się zadbać, by pokazać dane obejmujące pełne dziedziny zmiennych.
- 4 W rzeczywistości liczba potrzebnych rekordów jest również uzależniona od złożoności zależności funkcyjnej poddawanej modelowaniu.

Dane uczące dla SSN - obróbka wstępna

- 1 Dane numeryczne są przeskalowywane do właściwego dla sieci przedziału (normalizacja).
- 2 Wartości brakujące są zastępowane wartościami średnimi (lub innymi statystykami) obliczonymi na podstawie wartości zmiennych dostępnych w ciągu uczącym.
- 3 Typ danych nominalnych takich jak Płeć = Mężczyzna, Kobieta zamienia się na wartości numeryczne. Przy dużej liczbie klas łączy się je w grupy.
- 4 Wartości ciągłe poddaje się procesowi dyskretyzacji.

Plan wykładu

- 1 Biologiczne wzorce sztucznej sieci neuronowej
- 2 Sztuczny neuron
- 3 Perceptron
- 4 Architektury jednokierunkowych sztucznych sieci neuronowych
- 5 Przykłady zastosowania SSN w biznesie

Trzy główne dziedziny zastosowań sieci neuronowych

- 1 Prognozowanie (predykcja): Wykorzystuje istotne informacje zgromadzone w czasie przeszłym by prognozować szanse, zmiany w przyszłości. Sieci neuronowe są używane by ułatwić proces prognozowanie i podejmowanie decyzji.
 - giełda
 - waluty
 - nieruchomości
- 2 obniżanie kosztów
- 3 ewaluacja (ocena).

Prognozowanie — giełda

WRRAT <http://www.calsci.com/Stock.html>

Walkrich Investment Advisors używa sieci neuronowych BrainMaker do określenia średniej premii (rabatu) jaką rynek alokuje do poszczególnych branż. Następnie używa tego standardu w analizie neuronowej industry-by-industry w celu ustalenia, które akcje sprzedaje się poniżej ich wartości rynkowej. W tej analizie, sieć neuronowa będzie oceniać każdą akcję, dając wycenę (na podstawie ceny / zysk i stopa dywidendy), która jest dostosowana do rozmiaru, przemysłu, obrotu giełdowego oraz wpływu instytucjonalnego. Szacunkowa cena z sieci neuronowej jest porównywana do średniej w branży, dając miarę względną wartość każdej akcji - w skrócie, czy zasoby te są niedoszacowane lub przecenione przez rynek.

Prognozowanie (predykcja) — giełda

Capital Management LBS

Capital Management LBS wykorzystuje oprogramowanie ze SSN, by przewidzieć S&P 500 Index. Firma wykorzystuje system ekspertowy do zapewnienia instrukcji do sieci neuronowych, która następnie odpowiednio z regułami przetwarza dane. SSN testowana setkami danych z poprzednich dni przewiduje S&P 500 z dokładnością do około 95%.

Przewidywanie waluty

O'Sullivan Investments

Wykorzystuje wiele sieci neuronowych do przewidywania trendów na rynku. James O'Sullivan jest autorem artykułu o sieciach neuronowych „Practical Primer, AI in finance” z 1994, gdzie przedstawia niektóre z używanych sieci. Jednym z najważniejszych czynników, które wpływają na opracowanie dobrze działającej SSN jest zadanie jej odpowiedniego pytania. Zamiast po prostu zapytać, jaka jest przewidywana cena waluty, można zapytać, czy cena rynkowa będzie się zmieniać w jednym, czy drugim kierunku itp.

Przewidywanie cen gazu

Northern Natural Gas

Northern Natural Gas jest dostawcą gazu ziemnego. Na potrzeby sprzedaży muszą opracować i zgłosić stawkę za gaz, który sprzedają na podstawie średniego kosztu gazu. Poprzez wykorzystanie sieci neuronowej, które bierze pod uwagę takie czynniki jak kwartał roku, pora roku, temperatury itd. by przewidywać stawki paliwa w przyszłym miesiącu, firma była w stanie lepiej planować stawki.

Wycena nieruchomości z sieci neuronowych

Richard Borst

Kilka sieci neuronowych zostało wykorzystywanych do przewidywania cen sprzedaży domów aby wspomagać rzeczoznawców i sprzedawców w szacowaniu ceny ofertowej. Richard Borst nauczył sieć neuronową wyceny nieruchomości w Nowym Jorku. Jego sieć wykorzystuje prawie 20 zmiennych, w tym stopy kwadratowe powierzchni mieszkalnej, wiek itd. Wykorzystał ponad 200 rekordów sprzedaży z lat 1988 i 1989, aby wytrenować sieć z około 90% dokładnością.

Obniżanie kosztów — zastosowanie w marketingu

Microsoft - marketing bezpośredni

Firma Microsoft używa sieci neuronowych w celu maksymalizacji skuteczności kampanii marketingowej. Każdego roku firma wysłała mail do swoich zarejestrowanych klientów. Większość maili oferowała uaktualnienia lub nowe oprogramowanie, ale odsetek odpowiedzi był raczej niski. Firma wykorzystała sieci neuronowe, do której wprowadzano różne zmienne, np. jak dawno nastąpiła rejestracja, ile produktów zakupił klient itd. by efektywniej wskazywać użytkownikom. Wyniki wykazały do 35% oszczędności.

Punktacja kredytów

Dobry/zły kredytobiorca

Badania prowadzone przez dr Herberta Jensena wykazały, że „budowanie sieci neuronowej zdolnej do analizy zdolności kredytowej wnioskodawców kredytowych jest bardzo praktyczne i może być wykonane dość łatwo”. Sieci neuronowe został wyszkolony na nie więcej niż 100 wnioskach kredytowych gdzie podano takie dane jak: zawód, staż pracy itd. Pomimo stosunkowo niewielkiego zbioru uczącego sieć osiągnęła 75-80% poprawnych odpowiedzi. Dla porównania tradycyjne metody punktacji osiągają 75% poprawności.