

Wstęp do inteligencji komputerowej – zajęcia nr 7

Jarosław Stańczak

WSISiZ

Sztuczne sieci neuronowe cz. 1:

- neuron a sztuczny neuron
- modele sieci neuronowych
- sieci warstwowe
- perceptron
- algorytm propagacji wstecznej – uczenie sieci neuronowych wielowarstwowych.

Sztuczne sieci neuronowe a układ nerwowy

Sztuczne sieci neuronowe (SSN, *ang. artificial neural networks*, ANN, NN), konstruowane od kilkudziesięciu lat przez ludzi, bez wątplenia są wzorowane na budowie i działaniu układu nerwowego zwierząt, a w szczególności ssaków. Dlatego też zanim zaczniemy je omawiać, warto będzie poświęcić chwilę na przypomnienie sobie budowy i działania pierwowzoru.

Układ nerwowy

Układ nerwowy zwierząt, a szczególnie kręgowców i ssaków jest najbardziej skomplikowanym z układów, budujących ciała zwierząt i ludzi. Jednocześnie jego budowa i działanie nie są jeszcze do końca poznane.

Jest to zbiór wyspecjalizowanych komórek, pozostających ze sobą w złożonych relacjach funkcjonalnych i strukturalnych, odpowiadający za sterowanie aktywnością organizmu. Układ nerwowy jest w stanie wykryć określone zmiany zachodzące w otoczeniu i wywołać w związku z tym odpowiednią reakcję organizmu.

(za Wikipedią)

Układ nerwowy

Tkanka nerwowa – utworzona przez **neurony (komórki nerwowe)** i komórki glejowe, tworzy **układ nerwowy**. Odbiera, przekazuje i reaguje na bodźce pochodzące ze środowiska zewnętrznego, jak na przykład dotyk, temperatura czy światło. Przewodzi impulsy (elektryczne i chemiczne) z neuronu do efektorów, od receptorów, przetwarza impulsy w adekwatne odpowiedzi, przewodzi impulsy z neuronu do innego neuronu, wytwarza substancje przekąźnikowe. Komórki nerwowe umożliwiają organizmowi normalne funkcjonowanie w danym środowisku, adekwatną odpowiedź w zależności od sytuacji w środowisku zarówno wewnętrznym jak i zewnętrznym. Neurony stale rejestrują i analizują informacje o stanie wewnętrznym organizmu jak i zewnętrznym stanie otoczenia, przez co przygotowują organizm do adekwatnej reakcji. Do neuronów należy również koordynacja aktywności intelektualnej, świadomości, podświadomości, aktywności ruchowej czy też czynności gruczołów.

(za Wikipedią)

Układ nerwowy

Z tkanki nerwowej zbudowane są:

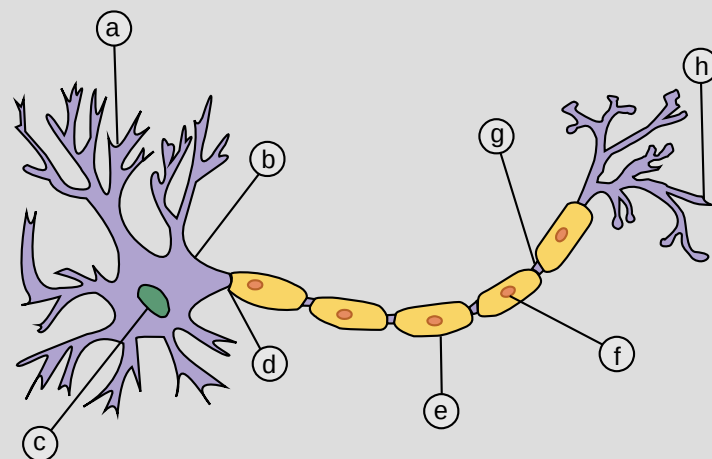
- ośrodkowy układ nerwowy
 - mózg
 - rdzeń kręgowy
- obwodowy układ nerwowy

Układ nerwowy

W mózgu człowieka znajduje się ok. 10^{11} neuronów, a liczba synaps (połączeń między neuronami) jest szacowana na 10^{14} .

Nie jest prawdą, że mózg człowieka wykorzystuje tylko 10% swoich możliwości. Przyroda nie lubi tworzyć nadmiarowych i niewykorzystywanych układów, które dodatkowo pochłaniają znaczą ilość energii (mózg - 2,5% masy ciała zużywa ok 20% energii w spoczynku!). Mózg wykonuje wiele funkcji związanych z utrzymaniem naszego ciała, tak więc nigdy nie może w 100% poświęcić się „rozmyślaniom” na jakiś temat, a zajmuje się regulacją i podtrzymaniem funkcji życiowych.

Neuron



Schemat budowy neuronu: a – dendryty, b – ciało komórki, c – jądro komórkowe, d – akson, e – otoczka mielinowa, f – komórka Schwanna, g – przewężenie Ranviera, h – zakończenia aksonu.
(źródło - Wikipedia)

Neuron - działanie

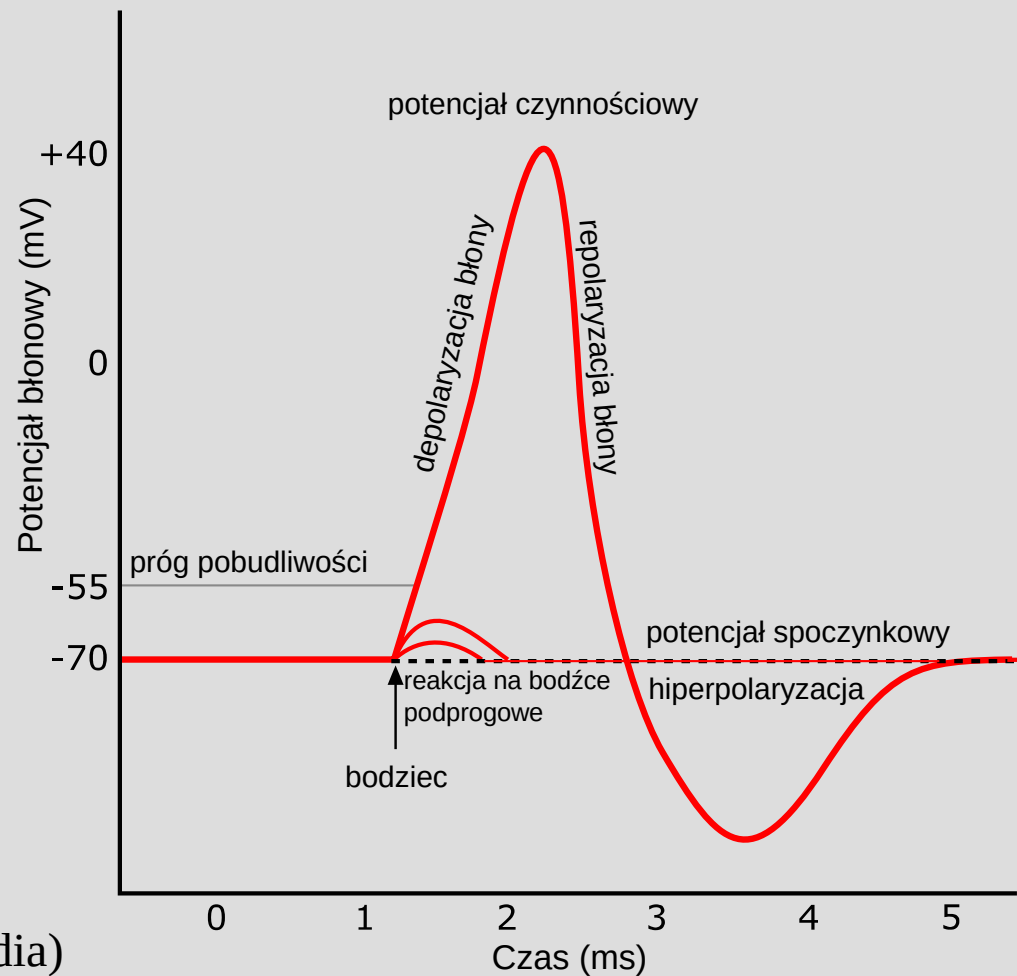
Podstawową funkcją neuronów jest przenoszenie i przetwarzanie informacji w postaci impulsów nerwowych (potencjałów czynnościowych), będących krótkotrwałymi, gwałtownymi zmianami potencjału błony komórkowej neuronu. Impulsy nerwowe w warunkach naturalnych są przewodzone tylko w jednym kierunku: od początkowego segmentu aksonu do synaps znajdujących się na jego zakończeniach.

Funkcjonalnie neuron można podzielić na cztery strefy:

- strefa wejścia – dendryty i ciało komórki, które odbierają impulsy od innych neuronów poprzez znajdujące się na nich synapsy
- strefa inicjacji – początkowy odcinek aksonu, tutaj powstaje potencjał czynnościowy neuronu
- strefa przewodzenia – akson
- strefa wyjścia – synapsy na zakończeniach aksonu.

(źródło - Wikipedia)

Neuron - działanie



(źródło rysunku - Wikipedia)

Historia powstania sztucznych sieci neuronowych (SSN)

- W 1943 r. powstał model sztucznego neuronu, podany przez McCullocha i Pittsa.
- W 1949 r. Hebb wyjaśnił mechanizm zapamiętywania i uczenia się nowych funkcji przez neurony, powstała tzw. reguła Hebba, wykorzystywana do nauki sieci także i współcześnie.
- W 1957 (lub 1958 według niektórych źródeł) r. Rosenblatt zbudował pierwszą elektromechaniczną sztuczną sieć neuronową z możliwością nauki – Perceptron, udowodnił też zbieżność stosowanego procesu nauki.
- W 1960 r. Widrow i Hoff zbudowali pierwszy neurokomputer Adaline/Madaline składający się z 8 neuronów i 128 połączeń między nimi.

Historia powstania sztucznych sieci neuronowych (SSN)

- Niestety (dla rozwoju sieci) w 1969 r. ukazała się praca Minsky'ego i Paperta, w której udowodnili oni, że sieć typu perceptronowego (jednowarstwowa lub wielowarstwowa z liniową funkcją aktywacji neuronów) może nauczyć się tylko problemów liniowo-separowalnych (np. nie może nauczyć się funkcji logicznej XOR). Spowodowało to upadek zainteresowania sieciami na prawie 20 lat, gdyż nie znano (a dokładniej już znano, ale niezbyt powszechnie) wtedy algorytmów nauki sieci wielowarstwowych z nieliniową funkcją aktywacji, w których tego ograniczenia, jak się później okazało, nie ma.
- W 1982 r. powstała sieć Hopfielda – sieć ze sprzężeniami zwrotnymi (bez warstw ukrytych), mająca właściwość pamiętania informacji.
- Ponowne zainteresowanie sieciami pojawiło się po opublikowaniu przez McClellanda i Rumelharta (1986) i Andersona i Rosenfelda (1988) prac, zawierających m. in. algorytm „bacpropagation” - algorytm nauki sieci wielowarstwowej z nieliniową lecz ciągłą funkcją aktywacji i z warstwami ukrytymi.
- W latach 1988-1990 ukazały się prace (m. in. Cybenki), dowodzące, że sieć zawierająca dwie warstwy ukryte może nauczyć się dowolnej funkcji klasyfikującej.
- Od tego czasu powstało wiele nowych rodzajów sieci, które z sukcesami zastosowano do rozwiązywania wielu problemów.

Cechy sztucznych sieci neuronowych (SSN)

Istniejące rodzaje sieci dość różnią się możliwościami i zastosowaniami, jednakże mają pewne cechy wspólne, które zdecydowały o ich popularności:

- zdolność uczenia się – wtedy gdy zostały wymyślane było to istotne novum, nie potrafiły tego robić żadne wymyślane wcześniej przez człowieka urządzenia;
- zdolność do generalizacji – sieci uczą się na przykładach, lecz generują także sensowne odpowiedzi dla danych, którymi nie były trenowane;
- odporność na uszkodzenia – sieć działa poprawnie nawet po usunięciu sporej części elementów;
- odporność na dane zniekształcone, zaszumione – sieć poprawnie przetwarza dane o znacznym stopniu odkształcenia;
- równoległe, bardzo szybkie przetwarzanie informacji;
- sieci dobrze nadają się do rozpoznawania obrazów, prognozowania, filtrowania zakłóceń, kompresji danych, modelowania działania skomplikowanych układów, rozwiązywania zadań klasy NP, rozwiązywania problemów w których posiadane dane są zaszumione i niepełne oraz wielu innych.

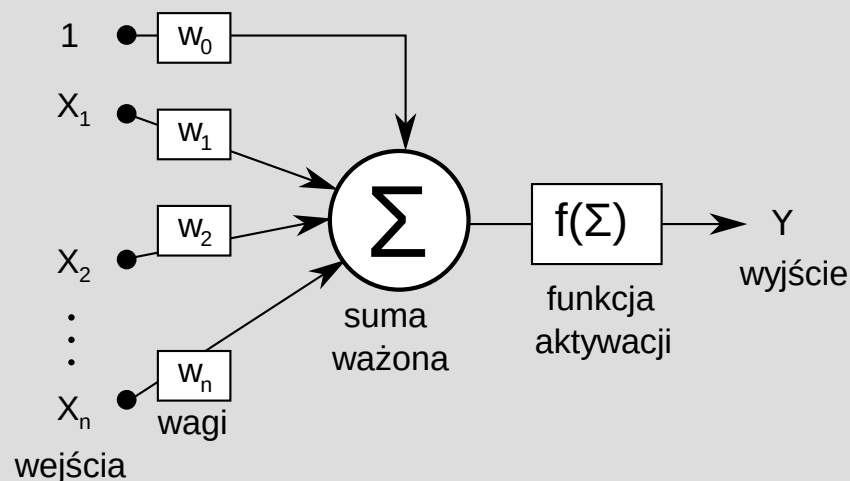
Cechy sztucznych sieci neuronowych (SSN)

Te unikalne cechy dały im w pewnym momencie (lata 90 XX w. i początek XXI w.), wielki rozgłos i mnogość zastosowań. Obecnie ta popularność nieco osłabła z powodu powstania wielu innych, często dużo lepszych (lub może z lepszym wytłumaczeniem teoretycznym, łatwiejszą analizą sposobu działania oraz łatwiejszą możliwością wykorzystania wiedzy zgromadzonej przez takie metody) metod rozwiązywania takich trudnych problemów.

Wydaje się jednak, że sztuczne sieci neuronowe w obecnej postaci nieco zawiodły oczekiwania twórców, gdyż nie udało się jak dotąd zbudować przy ich użyciu sztucznego mózgu, mającego cechy podobne do mózgów ludzi lub zwierząt. Różne rodzaje sztucznych sieci neuronowych posiadają pewne umiejętności charakterystyczne dla mózgu, jednakże żadna z nich nie posiada ich wszystkich jednocześnie, a tylko taka się umożliwiłaby budowę sztucznego mózgu.

Możliwe, że nie odkryto jeszcze wszystkich tajemnic działania żywych neuronów i ich sieci.

Sztuczny neuron McCullocha-Pittsa



(źródło rysunku – Wikipedia)

$$Y = f\left(\sum_{i=0}^n w_i * x_i\right) \quad (1)$$

w_i – wagi, ich modyfikacja umożliwia naukę neuronu (w_0 – waga skojarzona z sygnałem stałym, może być uznawana za tzw. wartość progową), X_i – sygnały wejściowe: wyjścia innych neuronów lub sygnały z zewnątrz ($X_0=1$ – wejściowa wartość stała), $f(...)$ – funkcja aktywacji najczęściej o postaci skokowej lub z nasyceniem (sigmoidalna) dla wartości niewzbudzonej i wzbudzonej, Y – sygnał wyjściowy neuronu.

Funkcje aktywacji w sztucznym neuronie McCullocha-Pittsa

Najczęściej stosowane funkcje aktywacji:

- skokowa lub funkcja signum

$$y(x) = \begin{cases} 1 & \text{dla } x \geq 0 \\ 0 & \text{dla } x < 0 \end{cases} \quad \text{lub} \quad y(x) = \text{sgn}(x) = \begin{cases} -1 & \text{dla } x < 0 \\ 0 & \text{dla } x = 0 \\ 1 & \text{dla } x > 0 \end{cases}$$

- sigmoidalna

$$y(x) = \frac{1}{1 + \exp(-\beta x)} \quad \text{lub} \quad y(x) = \tanh(\beta x)$$

- liniowa lub liniowa z nasyceniem

$$y(x) = a * x \quad \text{lub} \quad y(x) = \begin{cases} 0 & \text{dla } x < 0 \\ a * x & \text{dla } 0 \leq x < c/a \\ c & \text{dla } x \geq c/a \end{cases}$$

Neuron rzeczywisty a sztuczny neuron McCullocha-Pittsa

Praktycznie wszystkie istniejące sieci neuronowe buduje się wykorzystując ten prosty model neuronu McCullocha-Pittsa. Jest on uważany za uniwersalny i mogący posłużyć za podstawową jednostkę budującą dowolnie skomplikowaną maszynę obliczeniową. Jednakże nie odzwierciedla on niektórych cech prawdziwego neuronu:

- rzeczywiste neurony nigdy nie mają charakterystyki progowej, zawsze nieliniową lecz ciągłą – ta cecha oczywiście często jest uwzględniona w budowanych modelach;
- połączenia wejściowe (odpowiedniki wag) do neuronu nie muszą być liniowe, mogą same z siebie realizować już pewne funkcje przetwarzające, cecha ta jest zazwyczaj pomijana w konstruowanych SSN;

Neuron rzeczywisty a sztuczny neuron McCullocha-Pittsa

- wyjście wzbudzonego neuronu nigdy nie jest sygnałem stałym a impulsem bądź ciągiem impulsów, w których znaczenie może mieć wartość, faza, częstotliwość, opóźnienie sygnałów, czas działania, synchronizacja – właściwości te nie zostały całkowicie przebadane i zazwyczaj nie są uwzględniane w SSN, a ich rola w działaniu sieci może być duża;
- działanie synaps (połączeń między neuronami) nie jest do końca deterministyczne, a w każdym razie wpływ na przekazywanie sygnału mają różne czynniki chemiczne, hormony, leki, substancje psychoaktywne, zmęczenie, itp., również najczęściej nie są to zjawiska modelowane w SSN.

Niestety nie ma jednoznacznej odpowiedzi na pytanie, czy pominięcie tych cech nie zaburza w istotny sposób SSN w stosunku do pierwowzoru w takim stopniu, że nie będzie możliwe zbudowanie z nich układu analogicznego do ludzkiego mózgu. Z pewnością potrzebne są tu dalsze badania problemu.

Architektura SSN

SSN mogą być budowane ze sztucznych neuronów na różne sposoby.

Najczęściej przyjmuje się trzy główne architektury sieci:

- jednokierunkowe (typu perceptronowego), w których wyjścia z warstwy poprzedzającej są wejściami warstwy następnej, czyli sygnał wejściowy jest przekazywany od wejścia do wyjścia (nie dotyczy to fazy uczenia), np. sieć BP, Perceptron;
- rekurencyjne, zawierające sprzężenia zwrotne (połączenia wyjść neuronów z wejściami), sieci takie mają właściwość posiadania stanu, czyli historii działania sieci, może być ona uznawana za rodzaj pamięci, np. sieć Hopfielda, Boltzmannna, BAM;
- sieci komórkowe, w których każdy neuron jest połączony lub oddziałuje tylko z kilkoma swoimi sąsiadami, np. sieć Kohonena, ART.

Oczywiście można sobie wyobrazić różne inne sposoby łączenia sztucznych neuronów, jednakże nie są one w praktyce używane z uwagi na trudną analizę działania i problemy z opracowaniem metody uczenia.

Architektura, uczenie i funkcje SSN

Przyjęta architektura sieci w dużej mierze determinuje jej funkcje, sposób wykorzystania i metodę uczenia:

- sieci jednokierunkowe najlepiej nadają się do przetwarzania sygnałów, najczęściej wykorzystywane jest tu uczenie z nauczycielem przy użyciu różnych wariantów metody BP (dla ciągłej funkcji aktywacji) lub metod zaczerpniętych z innych dziedzin SI;
- sieci rekurencyjne mają zdolność pamiętania, wykorzystywane są jako pamięci adresowane zawartością, potrafią odtworzyć całość zapisanej informacji na podstawie jej fragmentu (autoasocjacja), informacja jest w nich zapisywana jako wagi wyliczane z pewnych funkcji, przypominających stany energetyczne cząstek, których minima to zapamiętane wzorce, czasem stosuje się uczenie metodą Hebba;
- sieci komórkowe służą najczęściej jako klasyfikatory danych, uczą się bez nauczyciela, przetwarzając nadchodzące dane, wykrywając na bieżąco ich cechy i klasyfikując do utworzonych kategorii.

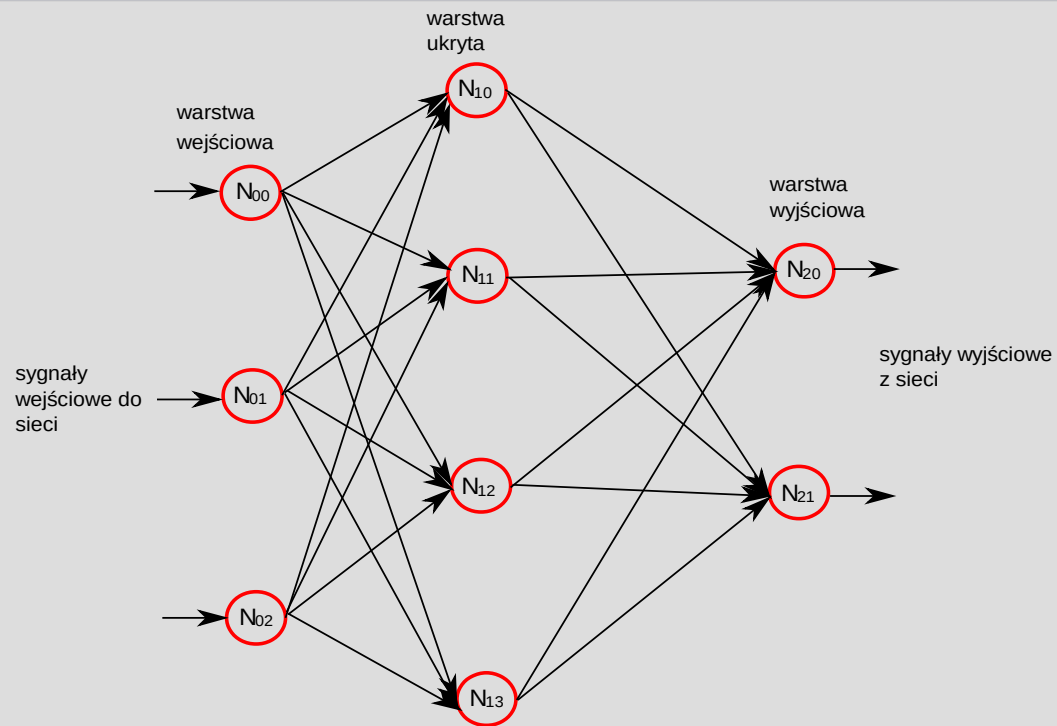
Sieci jednokierunkowe

Sieci jednokierunkowe składają się z warstw. Mogą mieć warstwę wejściową, dowolną liczbę warstw ukrytych (nieдоступnych bezpośrednio ani od strony wejść, ani od strony wyjść) i powinny mieć warstwę wyjściową (w szczególności mogą mieć też tylko jedną warstwę, która pełni wszystkie funkcje, lecz taka sieć nie jest uniwersalna). Sygnał wyjściowy z każdego neuronu warstwy poprzedniej jest doprowadzany do wejść każdego neuronu warstwy następnej (przy niektórych metodach uczenia niektóre słabsze połączenia mogą być eliminowane dla uproszczenia sieci).

W sieciach jednokierunkowych sygnał (poza fazą uczenia) przebiega wyłącznie od wejść do wyjść, bez sprzężeń zwrotnych i wymiany między neuronami w obrębie tej samej warstwy lub z pominięciem którejś z warstw.

W fazie uczenia sygnał błędu jest przekazywany w kierunku od wyjścia do wejścia, (propagacja wsteczna błędu).

Sieci jednokierunkowe

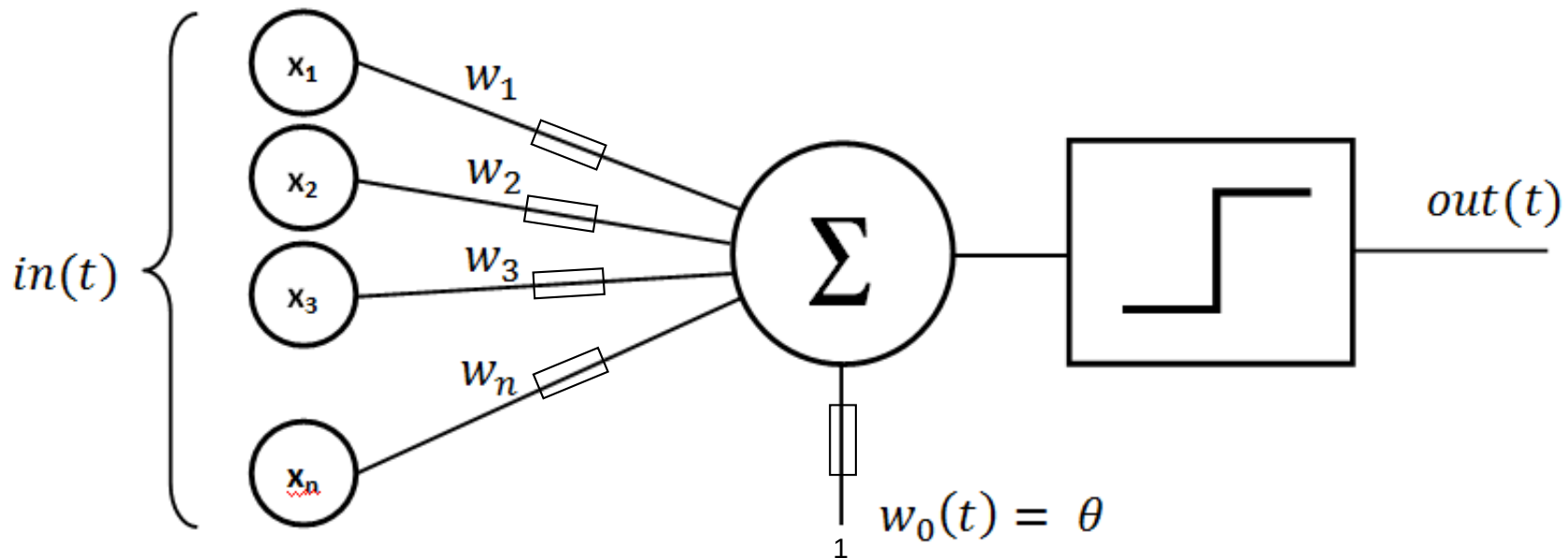


Sztuczna sieć neuronowa o 3 neuronach wejściowych, 4 w warstwie ukrytej i 2 w warstwie wyjściowej.

Sieci jednokierunkowe

perceptron

Perceptron był jedną z pierwszych realizacji (Rosenblatt-Wightman, 1957) sztucznej sieci neuronowej zbudowanej na bazie neuronu McCullocha-Pittsa. Była to realizacja sprzętowa, elektromechaniczna z wagami w postaci potencjometrów, poruszanych silniczkami (komputerowa symulacja takiego urządzenia była ówczesnie raczej niemożliwa), składała się z jednego, a później kilku właściwie niezależnych neuronów.



Sieci jednokierunkowe

perceptron

Niestety nie mogła to być sieć wielowarstwowa, gdyż nie potrafią uczyć warstw ukrytych. Wzór na działanie perceptronu można zapisać jako:

$$y(t) = \begin{cases} 1 & \text{if } \sum_{i=0}^n w_i * x_i(t) > 0 \\ 0 & \text{if } \sum_{i=0}^n w_i * x_i(t) \leq 0 \end{cases} \quad (2)$$

gdzie: w_0 – odpowiada progowi, $w_1 \dots w_n$ – wagom na wejściach $x_1(t) \dots x_n(t)$, $y(t)$ to sygnał wyjściowy.

Sieci jednokierunkowe

uczenie perceptronu

Reguła uczenia perceptronu wygląda następująco:

1. Wagi startowe powinny być zainicjowane niewielkimi wartościami rzeczywistymi lub zerami*.
2. Dla każdego z s uczonych wzorców d_j ($j=1 \dots s$) należy:
 - obliczyć wyjście sieci $y_j(t)$ na podstawie wzoru (2);
 - skorygować wagi według wzoru $w_i(t+1)=w_i(t)+\eta*(d_j-y_j(t))*x_{ij}$, gdzie η to współczynnik uczenia;
3. Powtarzać pkt. 2 aż miara błędu $\frac{1}{s} * \sum_{j=1}^s |d_j - y_j(t)| < \varepsilon$, gdzie ε to wartość ustalona przez użytkownika.

* W niektórych publikacjach zerowanie wag startowych nie jest zalecane.

Sieci jednokierunkowe

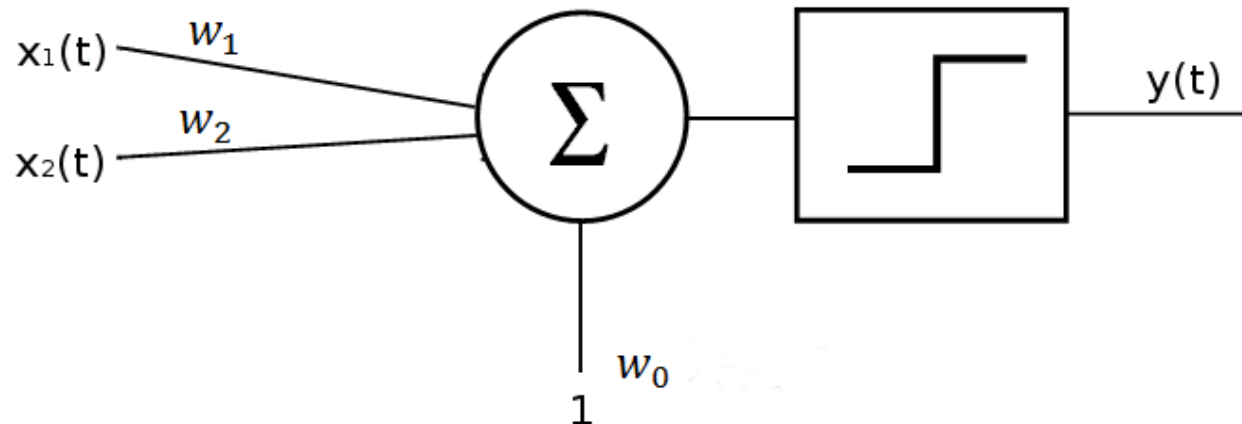
ćwiczenie uczenia perceptronu

Proszę dobrać wagi perceptronu, realizującego funkcję logiczną OR z $\varepsilon < 0,5$, współczynnik uczenia η przyjąć jako 0,3, wagi startowe równe 0.

Funkcja OR:

| $x_1 \setminus x_2$ | 0 | 1 |
|---------------------|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

Architektura perceptronu (rys. Wikipedia)



Sieci jednokierunkowe

działanie w fazie eksploatacji

Działanie sieci jednokierunkowej w fazie eksploatacji jest proste: po przyłożeniu sygnałów na wejściu, neurony wejściowe obliczają swoje wyjścia zgodnie ze wzorem (1) (czasem neurony wejściowe pełnią tylko funkcje dystrybucyjne i nie modyfikują sygnału) i podają je na wejścia neuronów warstwy kolejnej (najczęściej ukrytej), tu sytuacja się powtarza aż sygnał dotrze do wyjść sieci. Zdarza się, że funkcje aktywacji w różnych warstwach sieci są różnego typu.

W wersjach sprzętowych sieci neurony z pewnością muszą być odpowiednio synchronizowane aby różnice w czasie propagacji sygnału nie powodowały problemów, w symulacjach programowych sieci ten problem nie występuje. Zdecydowanie inaczej sieć działa w fazie uczenia. Temu zagadnieniu poświęcone są kolejne slajdy.

Sieci jednokierunkowe

faza uczenia

Aby sieć jednokierunkowa mogła pracować w fazie eksploatacji, najpierw należy ją nauczyć wykonywać swoją pracę. Uczenie w skrócie polega na przetwarzaniu sygnałów zgodnie z odpowiednimi przykładami zawierającymi wejścia i pożądane dla nich wyjścia. Przykłady te trzeba najpierw sieci zaprezentować. Faza uczenia polega na wielokrotnym prezentowaniu przykładów, obliczaniu odpowiedzi sieci i korekcji wag tak, aż błąd między wyjściem sieci a wzorcem spadnie poniżej określonego minimum. Przykłady można prezentować po kolei, ale znacznie lepsze efekty daje prezentacja w kolejności losowej, należy jednak pamiętać, aby wszystkie przykłady były pokazywane tak samo często i z jednakową częstością w czasie. Wagi startowe sieci przed uczeniem są **losowane** najczęściej jako nieduże liczby rzeczywiste.

Wagi startowe w sieciach BP nie mogą być jednakowe!
Sieć z jednakowymi wagami startowymi nie będzie w stanie się czegokolwiek nauczyć!

Sieci jednokierunkowe

faza uczenia

Początkowo sieci neuronowe (np. PERCEPTRON, ADALINE, MADALINE) budowane były z neuronów o skokowej funkcji aktywacji (poniżej pewnego progu na wyjściu było 0, powyżej 1 – była to swoista analogia do techniki cyfrowej). Jednakże powstał tu pewien problem, neurony takie można było uczyć np. zgodnie z tzw. regułą Hebba (o której więcej na następnym slajdzie), jednakże znamy dla nich wartości sygnałów wzorcowych tylko dla warstwy wyjściowej, nie wiadomo jakie sygnały wzorcowe powinny mieć warstwy ukryte. Dlatego też pierwsze sieci jednokierunkowe miały tylko jedną warstwę. Wspomniani już Minsky i Pappert wykazali, że sieć taka ma mocno ograniczone możliwości – może działać jako klasyfikator, w którym klasy są liniowo-separowalne, w praktyce oznacza to, że sieć typu perceptronowego nie mogła nauczyć się realizować np. funkcji logicznej XOR. Praca ta spowodowała prawie dwudziestoletni regres w badaniach nad sieciami. Jednakże problemy udało się przezwyciężyć.

Sieci jednokierunkowe

faza uczenia – reguły Hebba i Oji

Neurony w zasadzie wszelkich sieci neuronowych uczą zgodnie z zasadami tzw. reguły Hebba, która głosi, że jeśli neuron B jest cyklicznie pobudzany przez neuron A, to staje się on bardziej czuły na to pobudzanie, co można przedstawić jako wzrost wartości wagi łączącej te neurony:

$$\Delta w_{AB}(t+1) = \alpha * x_A(t) * y_B(t) \quad (3)$$

Jednakże okazało się, że reguła ta może prowadzić do nieograniczonego wzrostu wagi łączącej neurony, gdyby zbadać graniczne właściwości tak uczonej sieci. Oczywiście w praktyce nigdy nie stosuje się nieskończonej liczby powtórzeń sygnałów uczących, jednak postarano się wyeliminować tę niedogodność. Można to zrobić przez sztuczne ograniczenie wielkości wag lub też przez normalizację wektora wag po każdej iteracji, co prowadzi do reguły uczenia Oji:

$$\Delta w_{AB}(t+1) = \alpha * y_B(t) * (x_A(t) - y_B(t) * w_{AB}(t)) \quad (4)$$

$w_{AB}(t+1)$, $\Delta w_{AB}(t+1)$ – waga i przyrost wagi połączenia między neuronami, α - niewielka stała uczenia, $y_B(t)$ – wzor_B(t) - $x_B(t)$ – sygnał uczący, $x_A(t)$, $x_B(t)$ – sygnały wyjściowe neuronów, A i B oznaczają tu neurony z kolejnych warstwy sieci (A – poprzedniej, B – następnej).

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation

Wraz z opracowaniem i opublikowaniem w 1986 r. metody BP (metoda ta została wynaleziona niezależnie przez kilku badaczy, po raz pierwszy już w 1969 r., lecz nie doczekała się wtedy rozgłosu) badania nad sieciami nabrały nowego rozpędu. Wreszcie można było konstruować w pełni uniwersalne sieci wielowarstwowe. Warunkiem było zastosowanie w nich neuronów o różniczkowalnych funkcjach aktywacji. Wtedy możliwe stało się wykorzystanie typowych, gradientowych metod minimalizacji funkcji błędu popełnianego przez sieć, np. o postaci (5), a następnie propagacja tego błędu do kolejnych warstw ukrytych zmodyfikowanego przez wartości pochodnych funkcji aktywacji neuronów. Uczenie sieci w dalszym ciągu polega na prezentacji jej w odpowiedniej (np. losowej) kolejności wejść z wymaganymi wyjściami, obliczenie wartości błędu, a następnie jego propagacja w kierunku odwrotnym (wstecznym) do normalnej propagacji sygnału i korekcja wag na podstawie sygnału błędu.

Minimalizowane kryterium jakości:

$$Q(W) = \frac{1}{2} \sum_{i=0}^n \sum_{j=0}^{m_i} (Y_{ij}^s - X_{ij}^s)^2 \quad (5)$$

$Q(W)$ – funkcja jakości zależna od zbioru wag (W) sieci, Y_{ij}^s – wzorcowe wartości wyjść z sieci, n – liczba wzorców, i – indeks wzorca, j – indeks neuronu wyjściowego, m_s – liczba neuronów warstwy s (wyjściowej), X_{ij}^s – wyjścia neuronów warstwy s (wyjściowej) dla uczonych wzorców, $Y_{ij}^s - X_{ij}^s$ – błąd j -tego neuronu wyjściowego sieci w odpowiedzi na wzorzec i .

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation

Wzory na korekcje wag w tej metodzie można otrzymać różniczkując kryterium (5) względem wag, po podstawieniu do niego wzorów na obliczenie wyjścia z sieci (1). Jest to tzw. **reguła delta**.

- Dla warstwy wyjściowej ($k=s$) będzie to:

$$\Delta W_{jl}^s = \eta * \sum_{i=0}^n \delta_{ij}^s * X_{il}^{s-1} \quad \delta_{ij}^s = g'_{ij} \left(\sum_{p=0}^{m_{s-1}} W_{pj}^s * X_{ip}^{s-1} \right) * (Y_{ij}^s - X_{ij}^s) \quad (6).$$

- Dla warstw ukrytych ($k=2 \dots s-1$):

$$\Delta W_{jl}^k = \eta * \sum_{i=0}^n \delta_{ij}^k * X_{il}^{k-1} \quad \delta_{ij}^k = g'_{ij} \left(\sum_{q=0}^{m_{k-1}} W_{jq}^k * X_{iq}^{k-1} \right) * \sum_{p=0}^{m_{k+1}} W_{pj}^{k+1} * \delta_{ip}^{k+1} \quad (7).$$

- Dla warstwy wejściowej ($k=1$):

$$\Delta W_{jl}^1 = \eta * \sum_{i=0}^n \delta_{ij}^1 * We_{il}^0 \quad \delta_{ij}^1 = g'_{ij} \left(\sum_{q=0}^{m_0} W_{jq}^1 * We_{iq}^0 \right) * \sum_{p=0}^{m_2} W_{pj}^2 * \delta_{ip}^2 \quad (8).$$

n – liczba wzorców, i – indeks wzorca, j – indeks neuronu w warstwie, m_k – liczba neuronów warstwy k , m_0 – liczba wejść do sieci, Y_{ij}^s – wzorcowe wartości wyjść z sieci, We_{il}^0 – wzorcowe wartości wejść do sieci, X_{ij}^k – wyjścia neuronów

warstwy k dla uczonych wzorców, ΔW_{jl}^k , W_{jl}^k – zmiana wagi, waga łącząca neurony l w warstwie $k-1$ i j w warstwie k , η – współczynnik nauki, $g'_{ij}(\dots)$ – wartość pochodnej funkcji aktywacji $g_{ij}(\dots)$ neuronu j .

Sygnał δ jest odpowiednikiem błędu propagującego od warstwy wyjściowej sieci do warstwy wejściowej.

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation i jej rozszerzenia

Wzory (6-8) mogą wyglądać nieco inaczej w różnych opracowaniach nie tylko z powodu użycia innych oznaczeń, ale także dlatego, że:

- czasem optymalizuje się nieco inną postać kryterium jakości niż (5),
- możliwe jest użycie bardziej skomplikowanych metod metod optymalizacji niż metoda gradientowa największego spadku, np. metody Levenberga-Marquardta lub zmiennej metryki, są to metody o szybszej zbieżności lecz wymagają np. obliczania drugich pochodnych funkcji błędu i oczywiście powodują jeszcze większą komplikację wzorów (6-8), co np. przy dużych rozmiarach sieci może być problemem,

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation i jej rozszerzenia

- możliwe jest użycie dodatkowych technik poprawiających zbieżność, np. metody momentum (uśredniania kolejnych zmian wag), co znacząco zmniejsza fluktuacje wag w trakcie uczenia i znacząco przyspiesza ten proces; wracając do wzoru (7), pokazującego korekcję wag, potrzebne będzie uwzględnienie czasu (t) i poprzednich zmian wag:

$$\Delta W_{jl}^k(t) = \eta * \sum_{i=0}^n \delta_{ij}^k * X_{il}^{k-1} + \mu * \Delta W_{jl}^k(t-1) \quad (9)$$

oznaczenia jak w (7), μ – współczynnik momentum, przyjmowany z zakresu (0,1).

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation i jej rozszerzenia

Tabela wyników na podstawie L. Rutkowski, „Metody i techniki sztucznej inteligencji”

| Algorytm uczący | Liczba epok potrzebna do nauczania sieci |
|-----------------------|--|
| Największego spadku | 415 |
| Metoda momentum | 250 |
| Zmiennej metryki | 8 |
| Levenberga-Marquardta | 3 |

Sieci jednokierunkowe

faza uczenia – metoda Backpropagation i jej rozszerzenia

- stosuje się adaptację parametrów – strojenie wartości współczynnika uczenia η , w zależności od efektów uczenia, jeśli błąd maleje, to powoli zwiększa się wartość η , gdy natomiast błąd zaczyna rosnąć zmniejsza się go – ma to przyspieszyć uczenie sieci,
- używa się technik pozwalających pokonać minima lokalne występujące w optymalizowanej funkcji błędu, np. heurystyk,
- stosuje się też czasem metody heurystyczne do znajdowania wag sieci (działają one również dla sieci z progową funkcją aktywacji, dla których nie ma metod gradientowych), np. algorytm wzrostu, algorytmy ewolucyjne, symulowane wyżarzanie, metody rojowe.

Sieci jednokierunkowe

architektura i projektowanie sieci

Oprócz uczenia sieci, dużym problemem jest też jej zaprojektowanie, czyli dobranie liczby warstw i liczb neuronów w warstwach. Generalnie nie ma tu zbyt wielu wytycznych. Warstwa wejściowa i wyjściowa mają liczby neuronów zgodne z liczbą sygnałów w danych trenujących. Wiadomo, że w pełni uniwersalna sieć powinna mieć przynajmniej jedną (w niektórych pracach podane jest, że przynajmniej dwie, zależy to od tego, jakiego rodzaju aproksymowane przez sieć funkcje są brane pod uwagę: tylko ciągłe, czy dowolne i czy warstwa wejściowa pełni już jakieś funkcje, czy jedynie dystrybuuje sygnał) warstwę ukrytą z np. $2n+1$ neuronami, gdzie n to liczba wejść. Zazwyczaj stosuje się jednak mniejsze sieci.

Niektórzy badacze twierdzą, że zwiększenie liczby warstw ukrytych z niewielką liczbą neuronów w każdej, daje lepsze efekty niż jedna warstwa ukryta z dużą liczbą neuronów.

Sieci jednokierunkowe

architektura i projektowanie sieci

Niektóre wersje algorytmów nauki sieci mają możliwość likwidacji wag a nawet całych neuronów, których wkład do działania sieci jest niski – np. algorytm Optimal Brain Damage, wykorzystujący drugie pochodne kryterium jakości liczone po wagach. Współczynnik istotności zależny od tych pochodnych pokazuje, które neurony można usunąć.

Tu również można zastosować algorytmy heurystyczne nie tylko do uczenia sieci, ale też do znalezienia najlepszej architektury sieci (nie tylko wagi, ale też i architektura sieci – liczby neuronów i warstw ukrytych są wtedy optymalizowane).

Sieci jednokierunkowe

algorytm Optimal Brain Damage

1. Wybierz architekturę sieci – liczba warstw ukrytych i neuronów w warstwach, może być nieco nadmiarowa, np.: n (wejścia) – $2n + 1$ (warstwa ukryta) – m (wyjścia), gdzie n – liczba wejść, m – liczba wyjść.
2. Naucz sieć z założonym błędem.
3. Oblicz hesjan (macierz 2 pochodnych po wagach w sieci).
4. Oblicz istotności wag według wzoru (h_{kk} – odpowiednie wyrazy hesjanu, u_k – sygnały wejściowe):

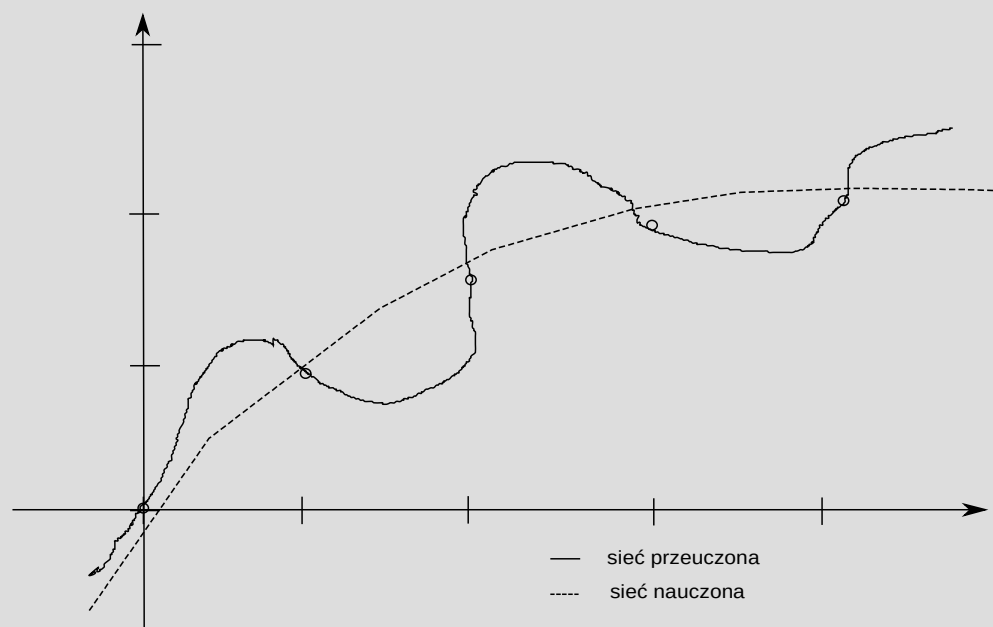
$$s_k = \frac{h_{kk} * u_k^2}{2}$$

5. Posortuj wagi według istotności i usuń te (ustaw ich wartości na stałe jako 0), które mają najmniejszą jej wartość i przejdź do pkt. 2 lub uznaj sieć za ustaloną, nauczoną i zakończ.

Sieci jednokierunkowe

problem przeuczenia sieci

W sztucznych sieciach neuronowych typu BP czasem zauważa się efekt przeuczenia sieci. Polega on na tym, że sieć bardzo dobrze i z małym błędem nauczyła się trenowanych przykładów, jednakże traci zdolność generalizacji, czyli sensownych odpowiedzi na przykłady nieprezentowane. Niewielka zmiana sygnałów na wejściu, daje znaczne zmiany na wyjściu. Jest to efekt niekorzystny i jak widać należy zachować umiar w długości uczenia, czyli „pogonią” za zmniejszeniem błędu nauczania sieci.



Sieci jednokierunkowe

zastosowania

Sieci jednokierunkowe najczęściej stosuje się do:

- rozpoznawania pisma i obrazów
- predykcji i prognozowania sygnałów, pogody, kursów giełdowych, itp.
- kompresji obrazów, sygnałów
- w algorytmach kierowania pojazdami
- nauki strategii w grach
- rozpoznawania mowy
- modelowania skomplikowanych obiektów np. w automatyce.

Dziękuję za uwagę!