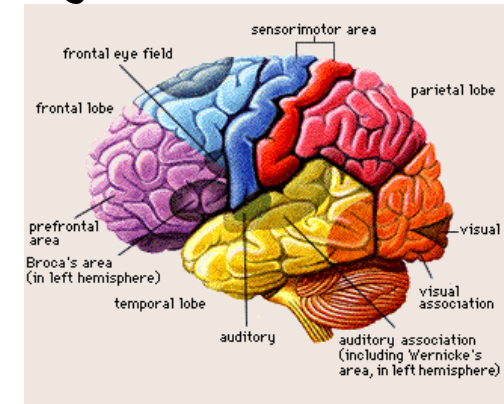


The brain - that's my second most favourite organ! - Woody Allen



Sieci neuronowe - wprowadzenie

Agnieszka Nowak - Brzezińska

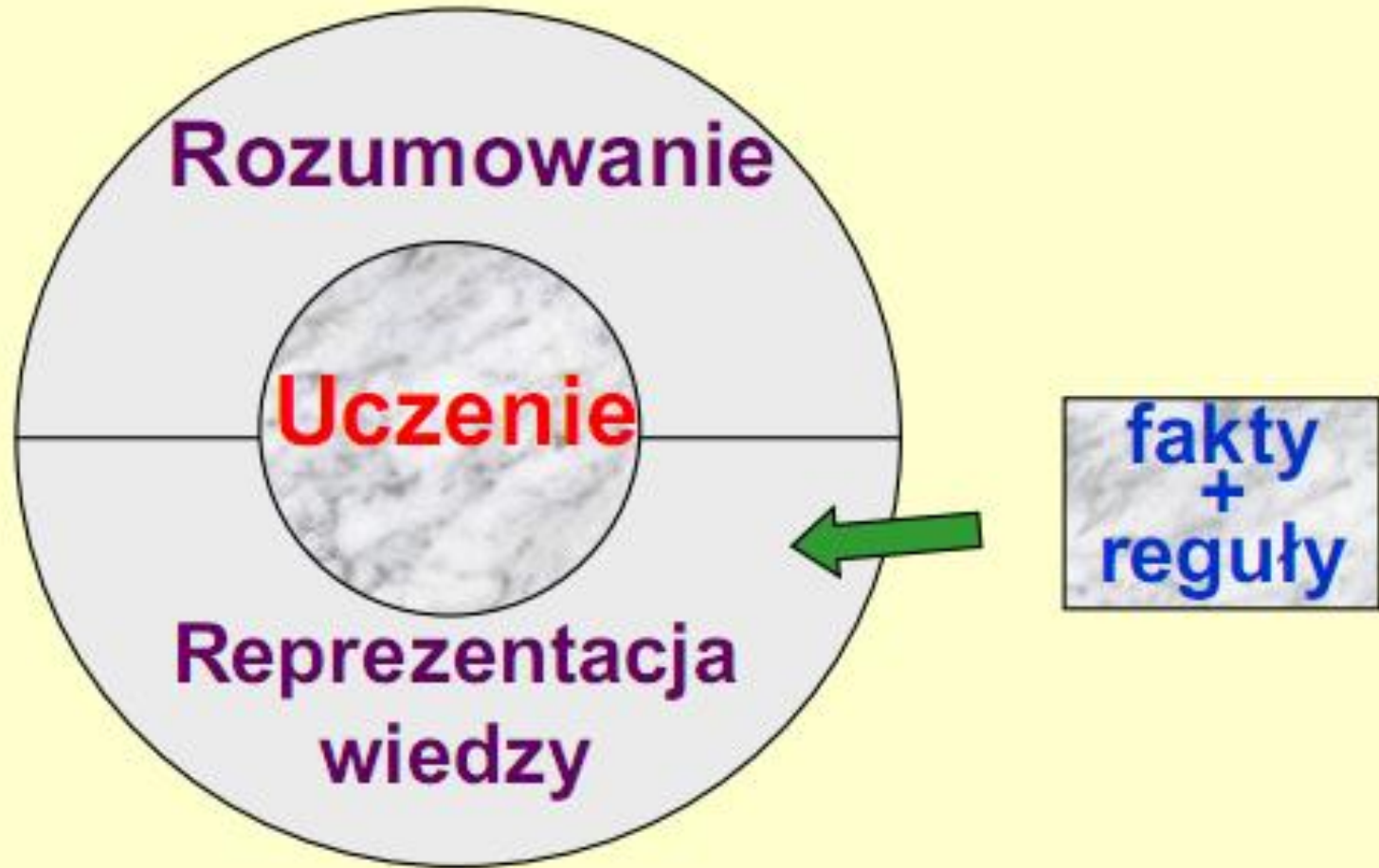
Klasyfikacja danych

- Klasyfikacja danych to jedno z podstawowych zadań wykonywanych przy użyciu sieci neuronowych. Na czym polega ?
- Celem klasyfikacji jest skojarzenie obiektu na podstawie jego cech charakterystycznych z pewną kategorią.

Cechy systemu inteligentnego

- Zdolność do uczenia się na podstawie przykładów i uogólniania przyswojonej wiedzy, w celu rozwiązywania problemów postawionych w nowym kontekście:
 - Zdolność do tworzenia reguł (skojarzeń) wiążących ze sobą oddzielne elementy systemu (obiektu)
 - Zdolność do rozpoznawania obiektów (obrazów cech) na podstawie niepełnej informacji.

Elementy Systemu Inteligentnego



By wytłumaczyć problem - kilka przykładów:

- rozpoznawanie twarzy - to klasyfikacja oparta np. na takich cechach jak: położenie oczu, zarys twarzy itp. Wbrew pozorom to bardzo ważny obszar zastosowań - pozwala na lokalizację poszukiwanej osoby w kartotece komputerowej.
- klasyfikacja obrazów graficznych- wykrywanie typu pojazdu na podstawie obrazu z kamery lub radaru.
- klasyfikacja obrazów czasowych - to klasyfikacja, w której obraz jest ciągiem danych pojawiających się w czasie. Klasyczny już przykład to sygnał mowy. W takim przypadku zadaniem sieci neuronowej jest przyporządkowanie zapisowi dźwięku tekstu wypowiedzi.

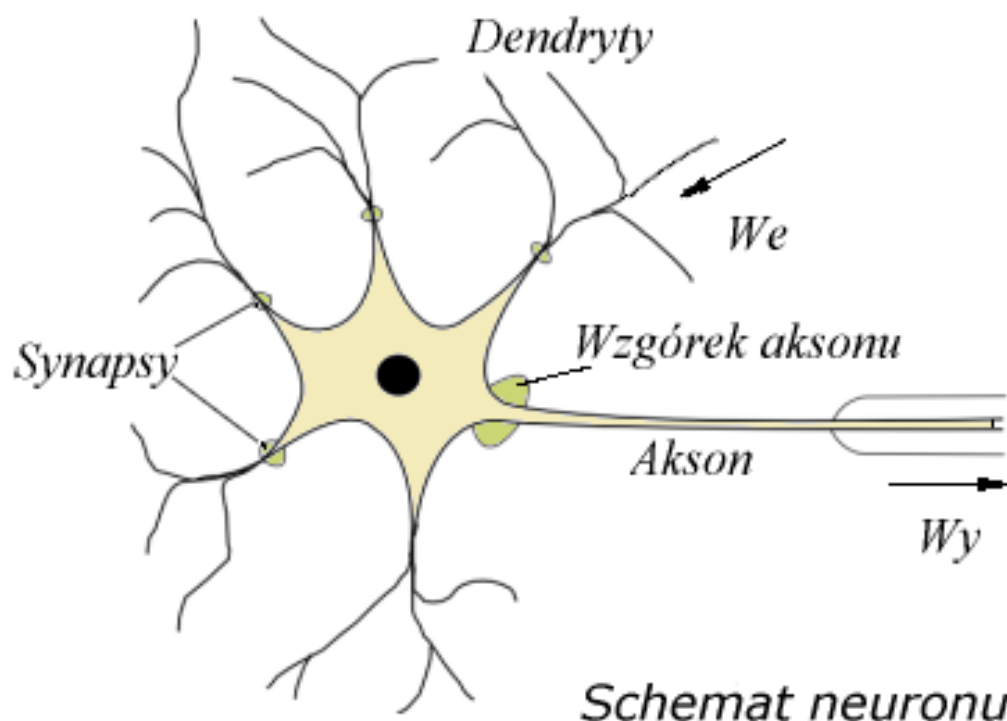
Sieci neuronowe znajdują szczególnie ważne zastosowanie tam, gdzie nie ma prostych reguł klasyfikacji. Można je bowiem uczyć na przykładach. Sam człowiek klasyfikacji danych uczy się stopniowo, korygując własne błędy i wykorzystując wcześniej zdobyte doświadczenia do poprawnego wnioskowania.

- Jeden z ciekawszych przykładów to system diagnozowania zawałów serca. Zadanie jakie postawiono sieci: *Na podstawie 41 danych na temat stanu zdrowia pacjenta - ustal czy występuje u niego zawał serca ?*
- Jaki był wynik budowy takiego systemu?
- **Sieć neuronowa potrafi postawić lepszą diagnozę niż zespół lekarzy specjalistów.** Diagnoza sieci neuronowej - odpowiednio nauczanej - poprawna w 92%, o 4% lepsza niż wynik lekarzy specjalistów. Co ciekawe, **sieć wykorzystuje do tego jedynie połowę informacji o pacjencie** i wystarcza to do postawienia tak trafnej diagnozy.
- Wniosek: uczenie sieci to sposób na system ekspercki, ale również na system, który odpowie nam - co jest podstawą dla trafnych decyzji.
- Uwaga: dokładny opis systemu: J. Żurada, M. Barski, W. Jędruch, *Sztuczne sieci neuronowe*, PWN, 1996

Słowem wstępu...

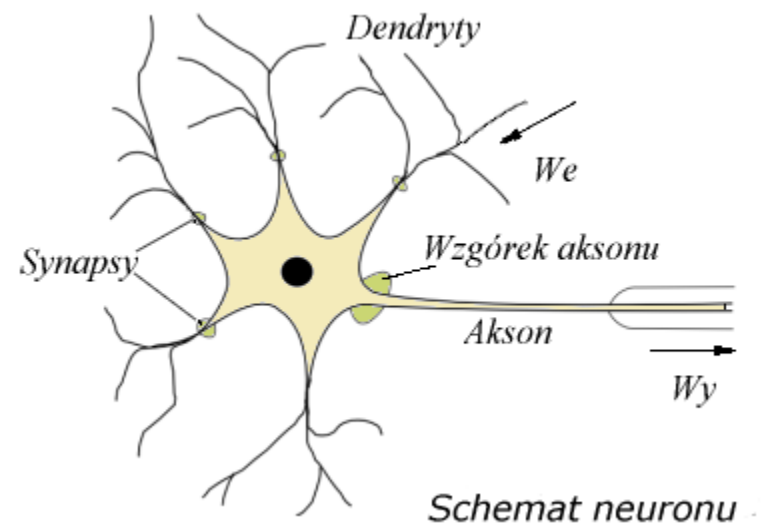
- Sieci neuronowe są systemami, których struktura jest w mniejszym lub większym stopniu wzorowana na działaniu ludzkiego systemu nerwowego i mózgu. Podstawową jednostką funkcjonalną tego systemu, zwanego biologiczną siecią neuronową, jest komórka nerwowa lub neuron. Możemy powiedzieć, że system nerwowy odbiera zmiany zachodzące zarówno w środowisku zewnętrznym jak i wewnątrz organizmu, przewodzi je i przetwarza w celu wypracowania odpowiednich reakcji.
- Bodźce odbierane z narządów zmysłów są przekazywane do ośrodkowego układu nerwowego za pomocą neuronów wstępujących (lub czuciowych).

- Na podstawie doprowadzonych do ośrodkowego układu nerwowego informacji są podejmowane decyzje dotyczące reakcji na odebrane bodźce. Schemat typowego neuronu:



Definicje SN

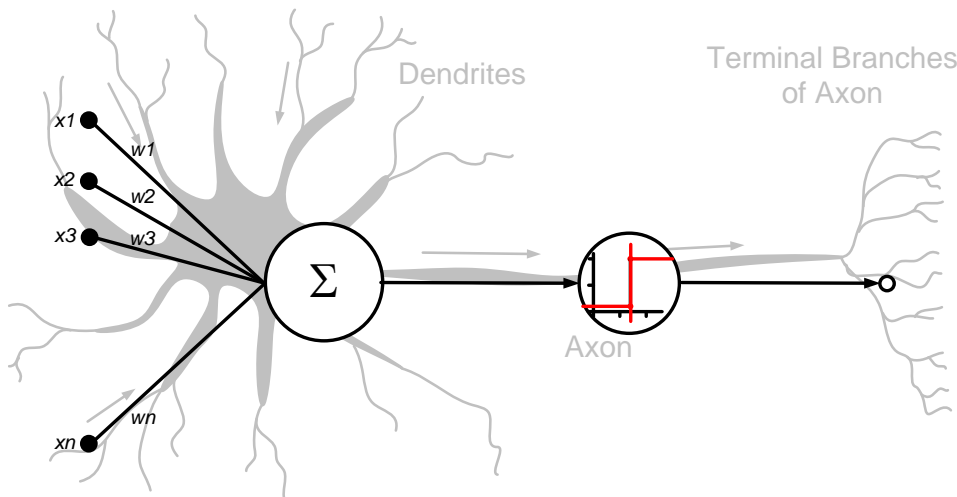
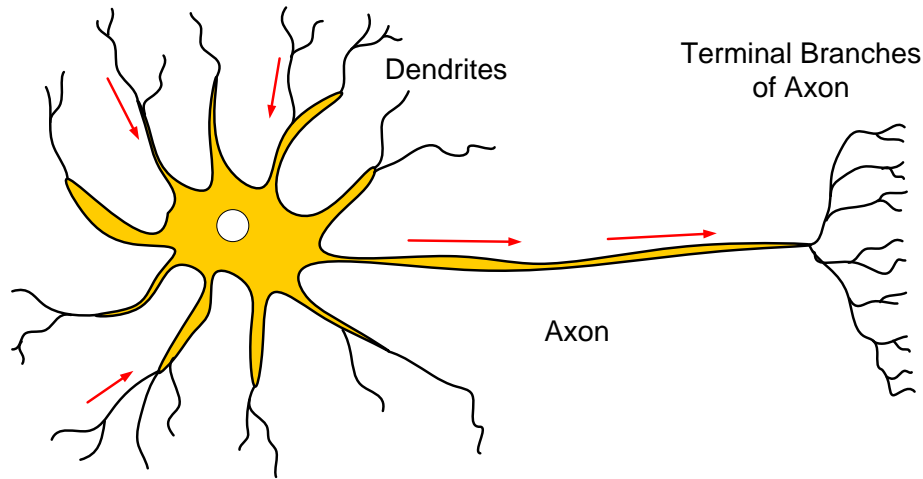
Sieć neuronowa to rodzaj architektury systemu komputerowego. Polega on na przetwarzaniu danych przez neurony pogrupowane w warstwy. Odpowiednie wyniki uzyskuje się dzięki procesowi uczenia, który polega na modyfikowaniu wag tych neuronów, które są odpowiedzialne za błąd.



- Największą częścią neuronu jest ciało komórki (**jądro**), od którego odchodzą dwa typy wypustek: **dendryty** oraz pojedynczy **akson**.
- To one uczestniczą w przekazywaniu impulsów elektrycznych o krótkim czasie trwania, zwanych impulsami nerwowymi.
- **Dendryty** są rozgałęzionymi włóknami wyspecjalizowanymi w odbieraniu bodźców od innych neuronów lub narządów zmysłu, **Akson** zaś przewodzi impulsy nerwowe z ciała komórki do innych neuronów. Może do wielu na raz.
- Przekazywanie informacji między neuronami odbywa się przez przewodzenie synaptyczne.

Inspiracja biologiczna

- Sygnały elektrochemiczne



w typowej komórce wyróżniamy 3 główne składniki: **ciało komórki**, **akson** i **dendryty**.

Dendryty tworzą gęsto rozgałęzioną pajęczynę cienkich włókien wokół ciała neuronu.

Informacja dociera od neuronu do dendrytów neuronu sąsiedniego za pośrednictwem **aksonów**.

Poprzez złącze akson-dendryt, zwane **synapsą** przekazywane są sygnały między neuronami. Sygnały dochodzące do synaps i dobierane przez dendryty są impulsami elektrycznymi.

Cechy sieci neuronowych:

- potrafi uogólniać posiadane wiadomości na przypadki których nigdy nie widziała, może więc "wymyślić" sposób rozwiązania nieznanego zadania, czego "normalny" komputer nigdy by nie zrobił.
- jest odporna na uszkodzenia, na błędne lub niepełne informacje. Potrafi działać sprawnie nawet wtedy gdy część jej elementów jest uszkodzona, a część danych została utracona.
- najlepiej nadaje się do rozwiązywania klasy zadań w których napisanie normalnego programu jest bardzo trudne lub niemożliwe, np. z braku znanego algorytmu. Gdzie posiadane dane są niepełne lub niedokładne, gdzie nie znamy sposobu rozwiązania problemu. Na pewno nie ma sensu stosować sieci do zadań typu, tabliczka mnożenia, edytor tekstu itp.
- może być połączona razem z systemem eksperckim i wtedy dostarcza wiedzy do bazy danych systemu eksperckiego.

- Kiedy SSN uczy się aproksymować pewną funkcję, musi otrzymać przykłady działania tej funkcji. Na tej podstawie SSN powoli zmienia wagi tak, by wyprodukować wyniki identyczne z wynikami podanymi w przykładach. Jest wtedy nadzieja, że kiedy SSN otrzyma inny zestaw wartości wejściowych również wyprodukuje poprawne wyniki.
- Zatem jeśli SSN ma rozpoznawać nowotwór po zdjęciu rentgenowskim, otrzyma na wstępie wiele obrazów zawierających nowotwory i wiele obrazów zawierających jedynie zdrowe tkanki. Po pewnym okresie uczenia z tymi obrazami, wagi w SSN powinny zawierać informacji pozwalające na prawidłową identyfikację nowotworów na zdjęciach rentgenowskich, które sieć analizuje po raz pierwszy.

Charakterystyczna cecha wszystkich sieci neuronowych:

Cechą wspólną wszystkich sieci neuronowych jest to, że na ich strukturę składają się neurony połączone ze sobą synapsami. Z synapsami związane są *wagi*, czyli wartości liczbowe, których interpretacja zależy od modelu.

Zastosowanie SN

- Sztuczne sieci neuronowe znajdują zastosowanie w rozpoznawaniu i klasyfikacji wzorców (przydzielaniu wzorcom kategorii), predykcji szeregów czasowych, analizie danych statystycznych, odszumianiu i kompresji obrazu i dźwięku oraz w zagadnieniach sterowania i automatyzacji.



Zastosowanie SN:

- **NIE:** do obliczeń, tabliczki mnożenia, do edytorów tekstu itp. zastosowań gdzie można łatwo zastosować znany algorytm.
- **TAK:** tam, gdzie algorytm postępowania jest bardzo trudny do osiągnięcia, gdzie dane są niekompletne lub niedokładne, gdzie przebieg badanego zjawiska jest nieliniowy itp. Tam, gdzie jest dużo danych, jakieś wyniki ale nie znamy jeszcze metody działania.

- 1.)Przetwarzanie danych giełdowych i prognozowanie kursów akcji.
- 2.)Prognozowanie trzęsień ziemi i zjawisk pogodowych.
- 3.)Rozpoznawanie w jakim języku jest napisany dany tekst.
- 4.)Przetwarzanie zeskanowanego obrazu na tekst.
- 5.)Badania psychiatryczne.
- 6.)Prognozy giełdowe, sprzedaży, Prognozy cen i danych ekonomicznych.
- 7.)Poszukiwania ropy naftowej.
- 8.)Sterowanie procesów przemysłowych.
- 9.)Prognozowanie liczb w Totka (jak na razie bez rezultatów).

Mózg ludzki

- Objętość: 1400 cm³
- Powierzchnia: 2000cm²
- Masa: 1,5 kg
- Pokrywająca półkule kora mózgowa zawiera 10¹⁰ komórek nerwowych
- Liczba połączeń między komórkami: 10¹⁵
- Komórki wysyłają i odbierają sygnały, szybkość operacji = 10¹⁸ operacji / s
- Sieć neuronowa jest uproszczonym modelem mózgu !

	processing elements	element size	energy use	processing speed	style of computation	fault tolerant	learns	intelligent, conscious
	10^{14} synapses	10^{-6} m	30 W	100 Hz	parallel, distributed	yes	yes	usually
	10^8 transistors	10^{-6} m	30 W (CPU)	10^9 Hz	serial, centralized	no	a little	not (yet)

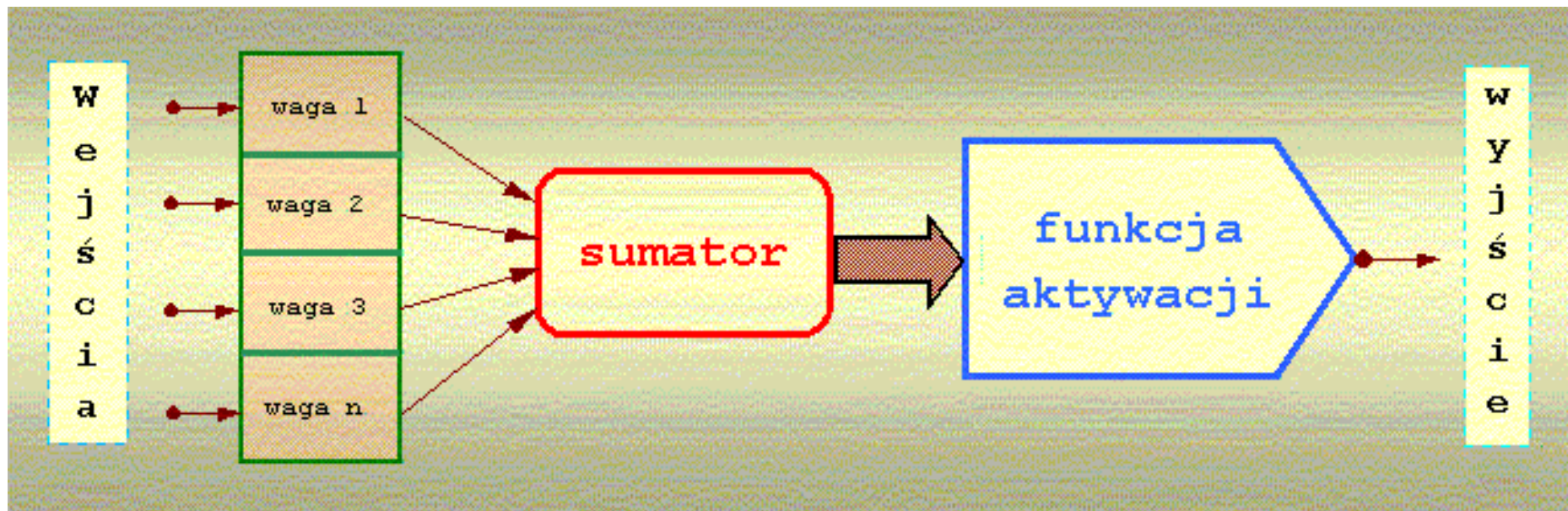
- Objętość: 1400 cm^3
- Powierzchnia: 2000 cm^2
- Masa: 1,5 kg
- Pokrywająca półkule kora mózgowa zawiera 10^{10} komórek nerwowych
- Liczba połączeń między komórkami: 10^{15}
- Komórki wysyłają i odbierają sygnały, szybkość operacji = 10^{18} operacji / s
- Sieć neuronowa jest uproszczonym modelem mózgu !

Podstawowe cechy mózgu

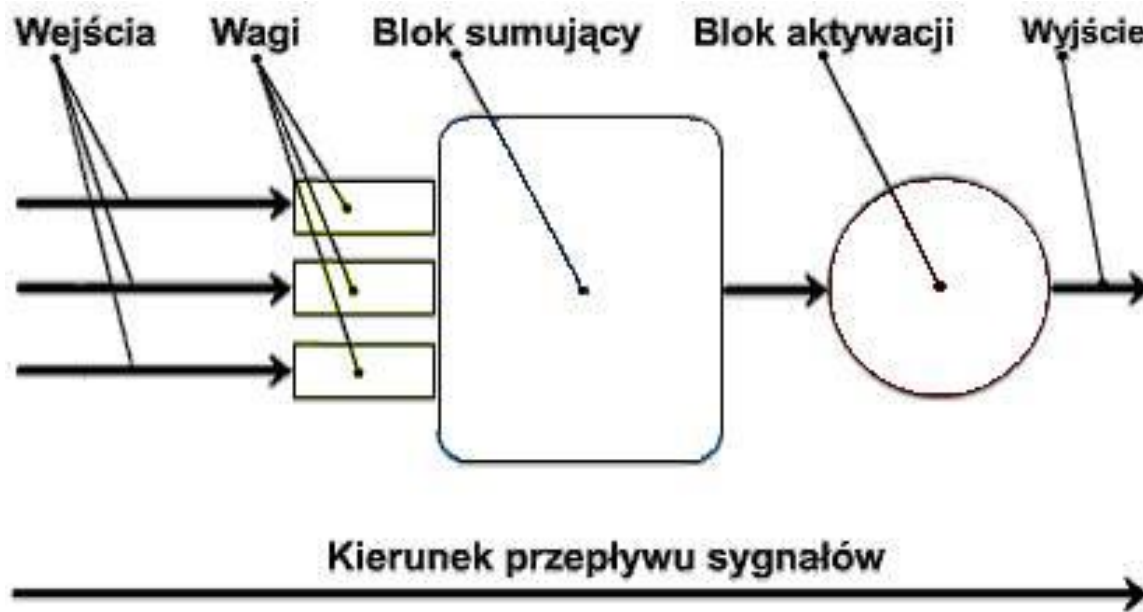
- ODPORNY NA USZKODZENIA;
- ELASTYCZNY – ŁATWO DOSTOSOWUJE SIĘ DO ZMIENNEGO OTOCZENIA;
- UCZY SIĘ - NIE MUSI BYĆ PROGRAMOWANY;
- POTRAFI RADZIĆ SOBIE Z INFORMACJĄ ROZMYTĄ, LOSOWĄ, ZASZUMIONĄ LUB NIESPÓJNĄ;
- W WYSOKIM STOPNIU RÓWNOLEGŁY;
- MAŁY, ZUŻYWA BARDZO MAŁO ENERGII.

Schemat sztucznego neuronu

- do wejść doprowadzane są sygnały dochodzące z wejść sieci lub neuronów warstwy poprzedniej. Każdy sygnał mnożony jest przez odpowiadającą mu wartość liczbową zwaną wagą. Wpływa ona na percepcję danego sygnału wejściowego i jego udział w tworzeniu sygnału wyjściowego przez neuron.
- Waga może być pobudzająca - dodatnia lub opóźniająca - ujemna;
- jeżeli nie ma połączenia między neuronami to waga jest równa zero. Zsumowane iloczyny sygnałów i wag stanowią argument funkcji aktywacji neuronu.



Uproszczony model neuronu pokazujący wyrażane jego podobieństwo do modelu naturalnego



Wejście

- sygnały przychodzące przez dendryty

Wagi

- odpowiedniki modyfikacji dokonywanych przez synapsy

Blok sumujący

- jądro

Blok aktywacji

- wzgórek aksonu

Wyjście

- akson

Formuła opisująca działanie neuronu

$$y = f(s)$$

W której

$$s = \sum_{i=0}^n x_i w_i$$

Funkcja aktywacji może przyjmować różną postać w zależności od konkretnego modelu neuronu.

Wymagane cechy funkcji aktywacji - to:

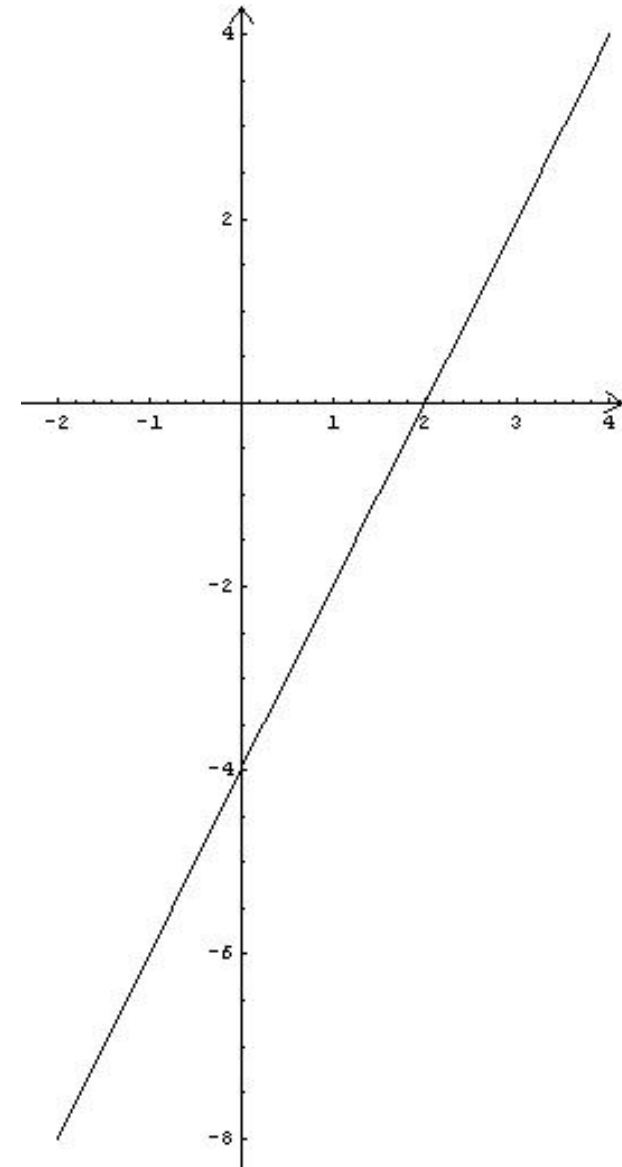
- ciągłe przejście pomiędzy swoją wartością maksymalną a minimalną,
- łatwa do obliczenia i ciągła pochodna,
- możliwość wprowadzenia do argumentu parametru α do ustalania postaci analitycznej funkcji aktywacji.

Aproksymacja funkcji

- Zasada działania SSN opiera się na aproksymowaniu pewnej funkcji, uczą się zadanej funkcji poprzez obserwowanie przykładów jej działania.
- Wartość funkcji aktywacji jest sygnałem wyjściowym neuronu i propagowana jest do neuronów warstwy następnej. Funkcja aktywacji przybiera jedną z trzech postaci:
 - skoku jednostkowego tzw. funkcja progowa
 - liniowa
 - nieliniowa

Funkcja aktywacji - **Funkcja liniowa**

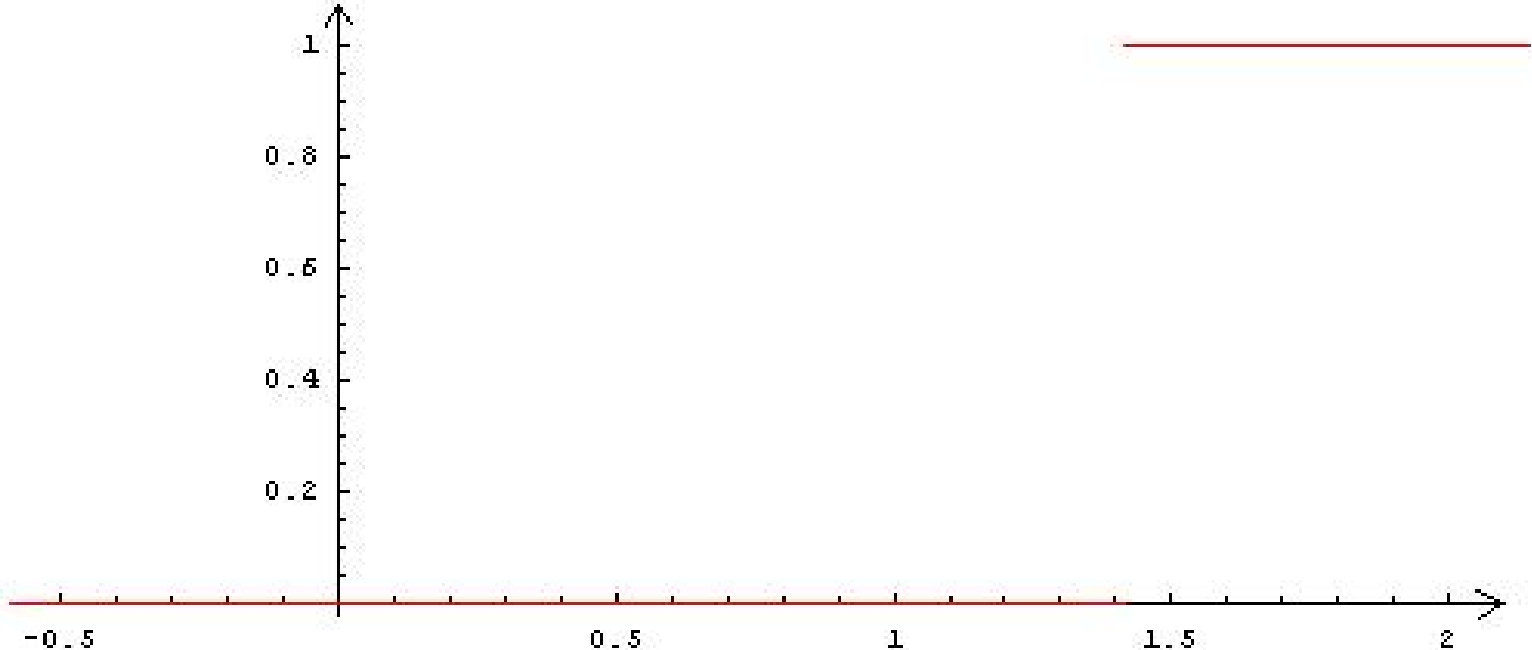
- $y = k(\phi - \phi_0)$
gdzie ϕ_0 jest wartością progową sygnału,
poniżej której wartość sygnału $y=0$,



Funkcja skoku jednostkowego - unipolarna

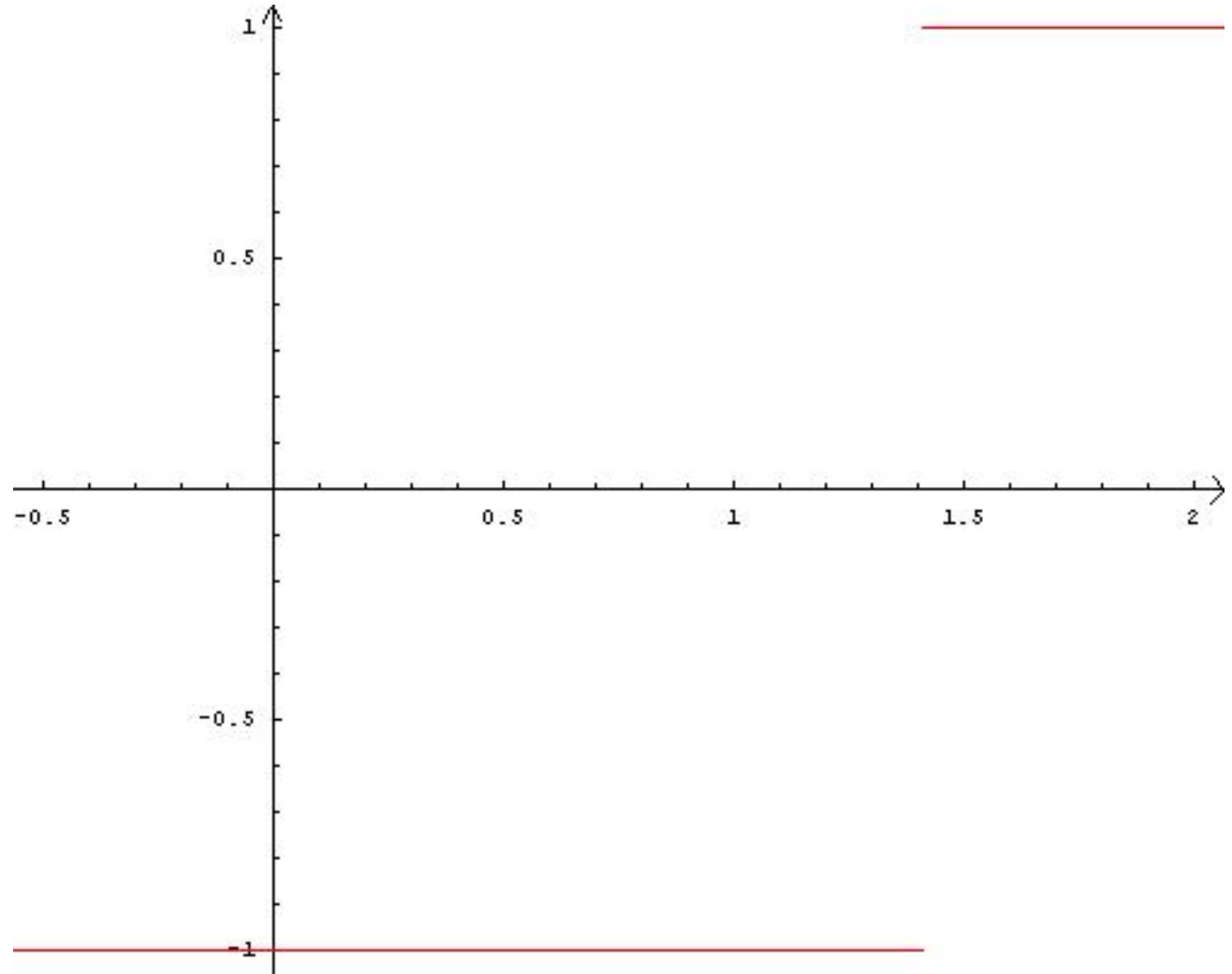
Tą funkcją było aktywowane wyjście
w modelu McCullocha i Pittsa.

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



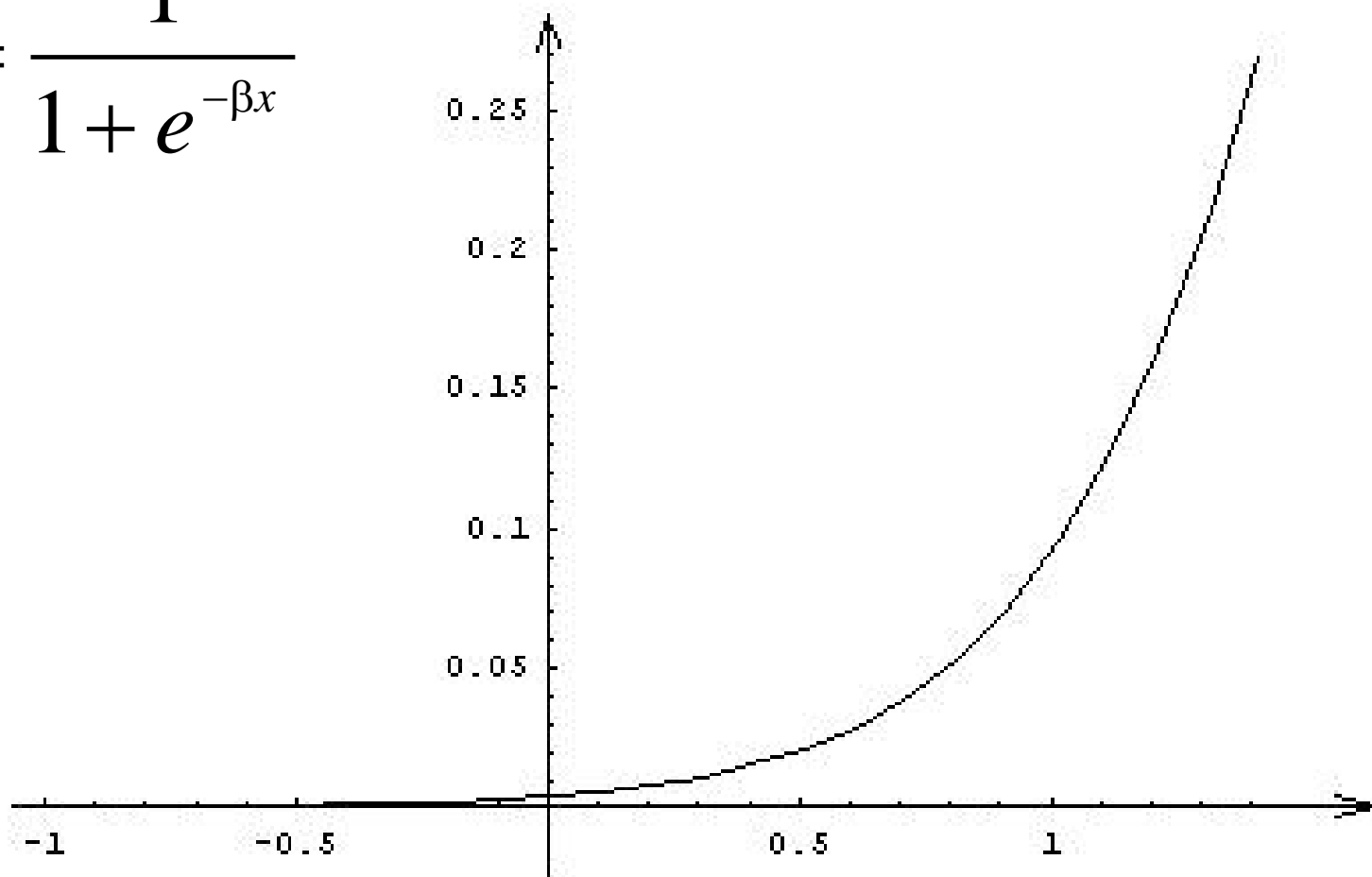
Funkcja skoku jednostkowego -bipolarna

- $y = -1$ dla $\phi < \phi_0$
 $y = 1$ dla $\phi \geq \phi_0$



Funkcja sigmoidalna - *unipolarna*

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

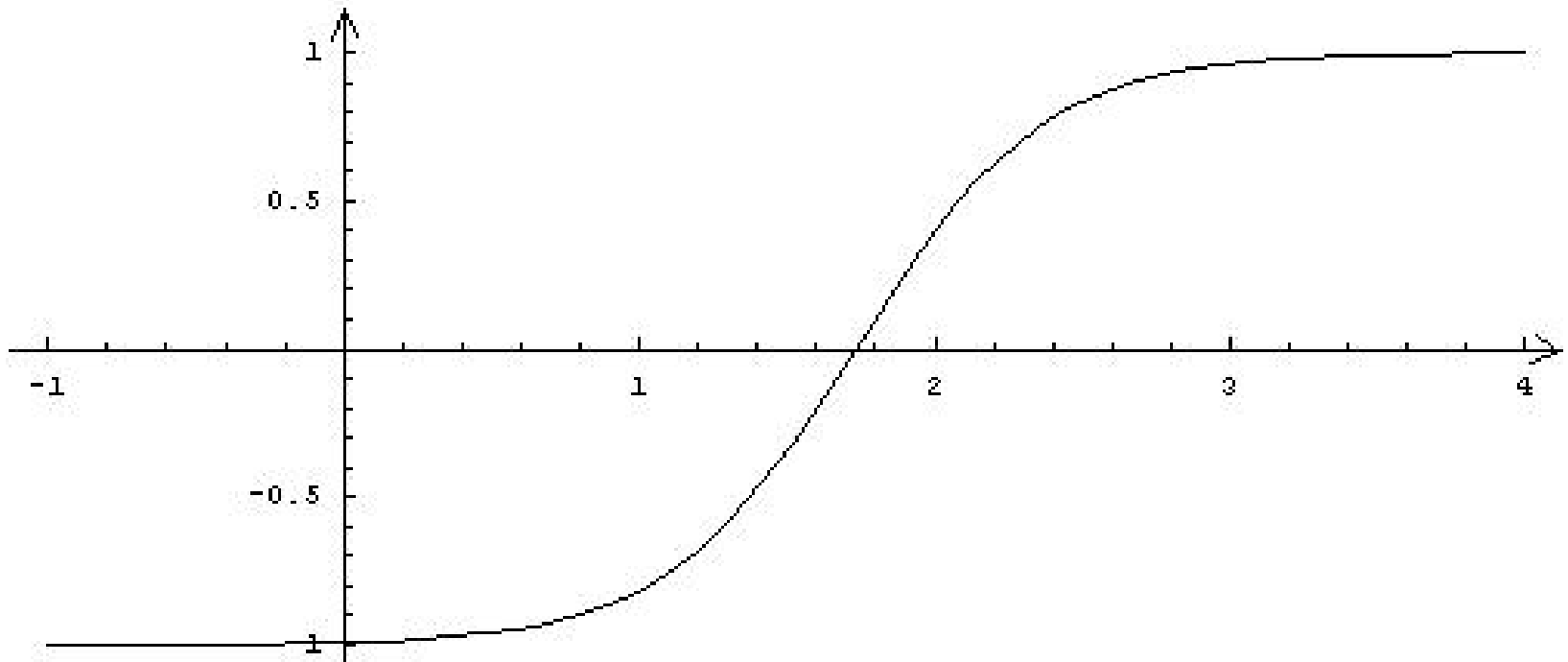


Nazywana też funkcją logistyczną, najczęściej stosowana

Funkcja sigmoidalna - *bipolarna*

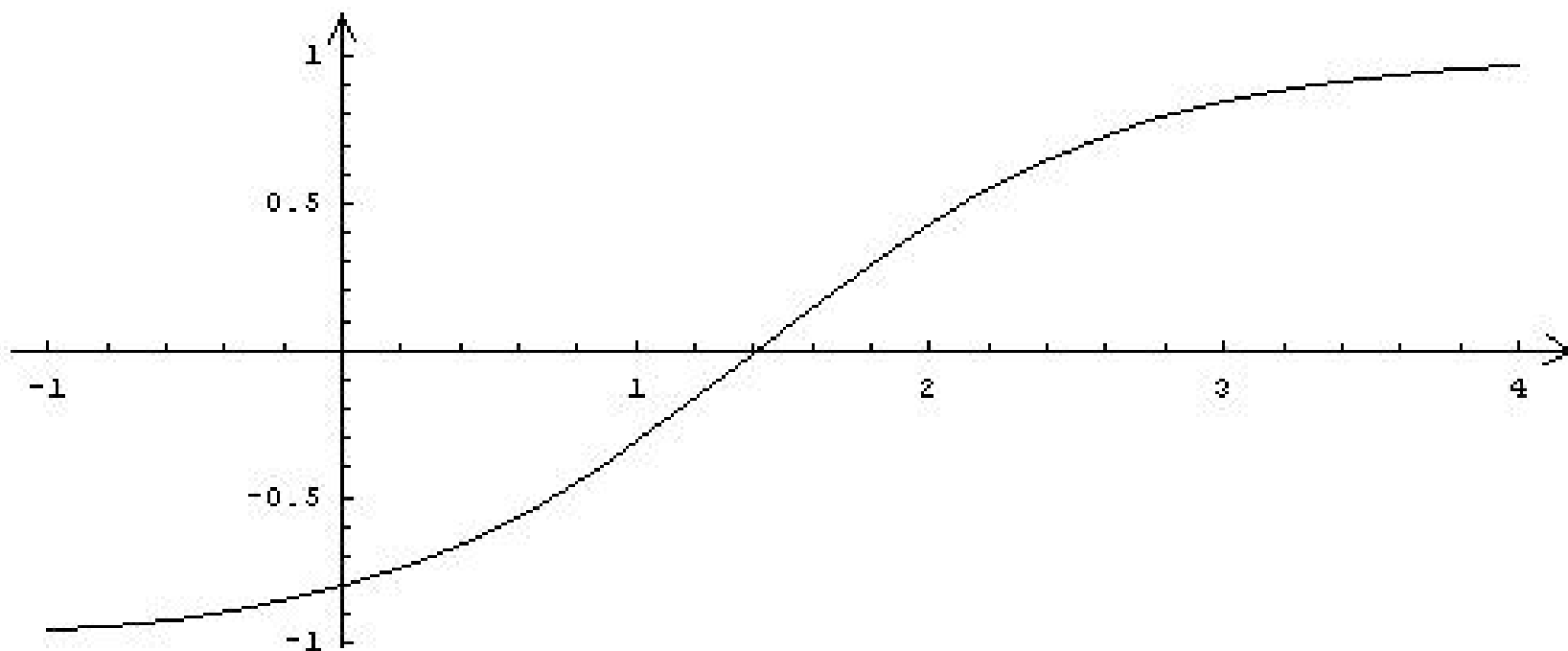
Funkcja jest przetworzeniem funkcji unipolarnej. Dzięki temu został rozszerzony przedział wartości wyjścia $(-1,1)$:

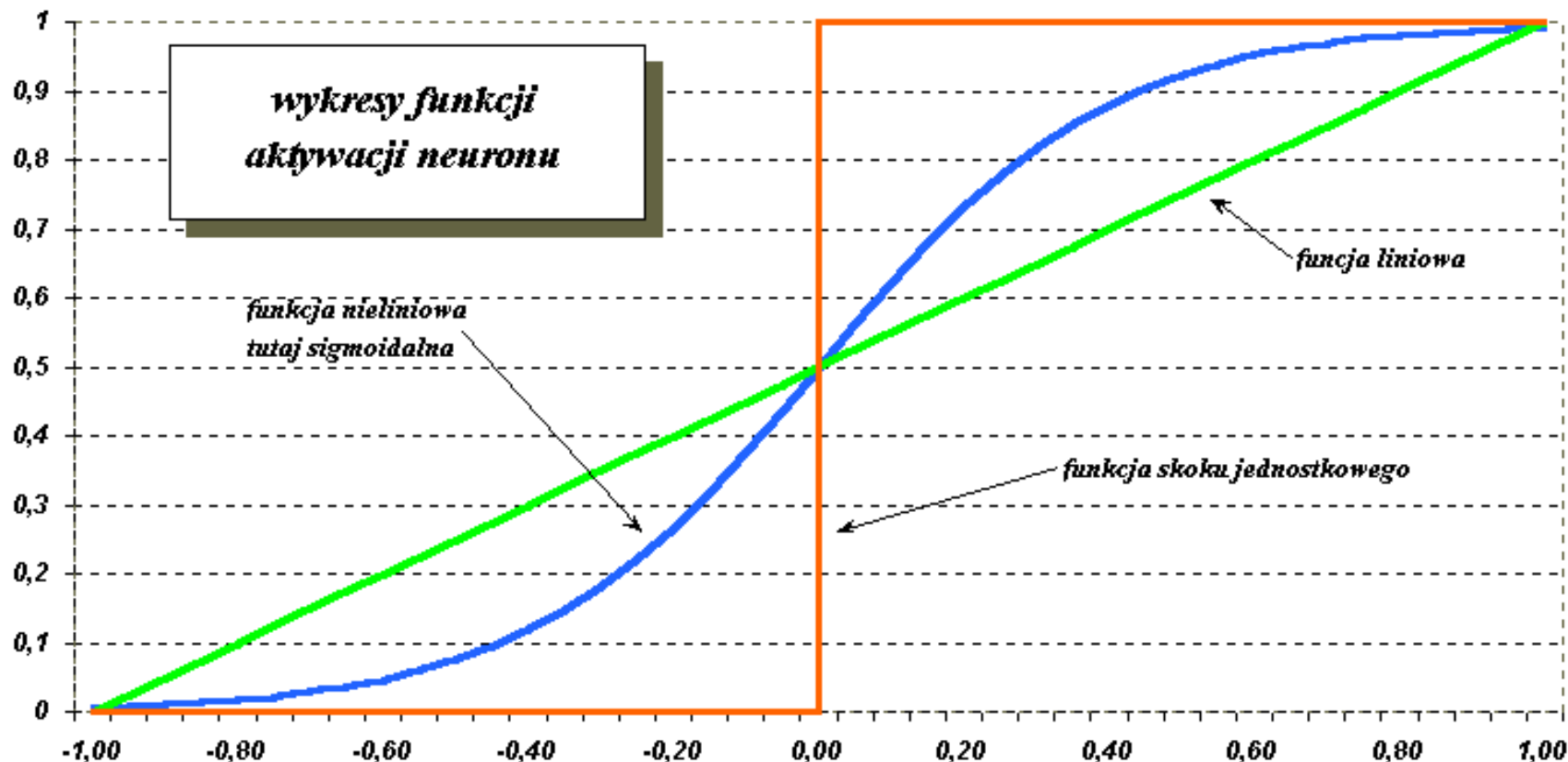
$$y = 2/(1+\exp(-\alpha(\phi-\phi_0))) - 1$$



Tangens hiperboliczny

$$f(x) = \frac{2}{1 + e^{-\beta x}} - 1 = \frac{1 - e^{-\beta x}}{1 + e^{-\beta x}}$$





- Wybór funkcji aktywacji zależy od rodzaju problemu jaki stawiamy przed siecią do rozwiązania. Dla sieci wielowarstwowych najczęściej stosowane są funkcje nieliniowe, gdyż neurony o takich charakterystykach wykazują największe zdolności do nauki. Polega to na możliwości odwzorowania dowolnej zależności pomiędzy wejściem a wyjściem sieci ale w sposób płynny. Umożliwia to otrzymanie na wyjściu sieci informacji ciągłej a nie tylko postaci: TAK - NIE.

wzory nieliniowych funkcji aktywacji i ich pochodnych

funkcja	Wzór funkcji	Wzór pochodnej
Sigmoida	$f(x) = \frac{1}{1 + e^{-(beta * x)}}$	$f'(x) = beta * (1 - f(x)) * f(x)$
Tangens hiperboliczny	$f(x) = \tanh(beta * x)$	$f'(x) = beta(1 - f^2(x))$
Sinusoida	$f(x) = \sin(beta * x)$	$f'(x) = beta \sqrt{1 - f^2(x)}$
Cosinusoida	$f(x) = \cos(beta * x)$	$f'(x) = -beta \sqrt{1 - f^2(x)}$
$\frac{x}{(1 + x)}$ (bez nazwy)	$f(x) = \frac{beta * x}{(1 + beta * x)}$	$f'(x) = \frac{beta}{1 + beta * x } - \frac{ beta * x }{(1 + beta * x)^2}$

Historia SN

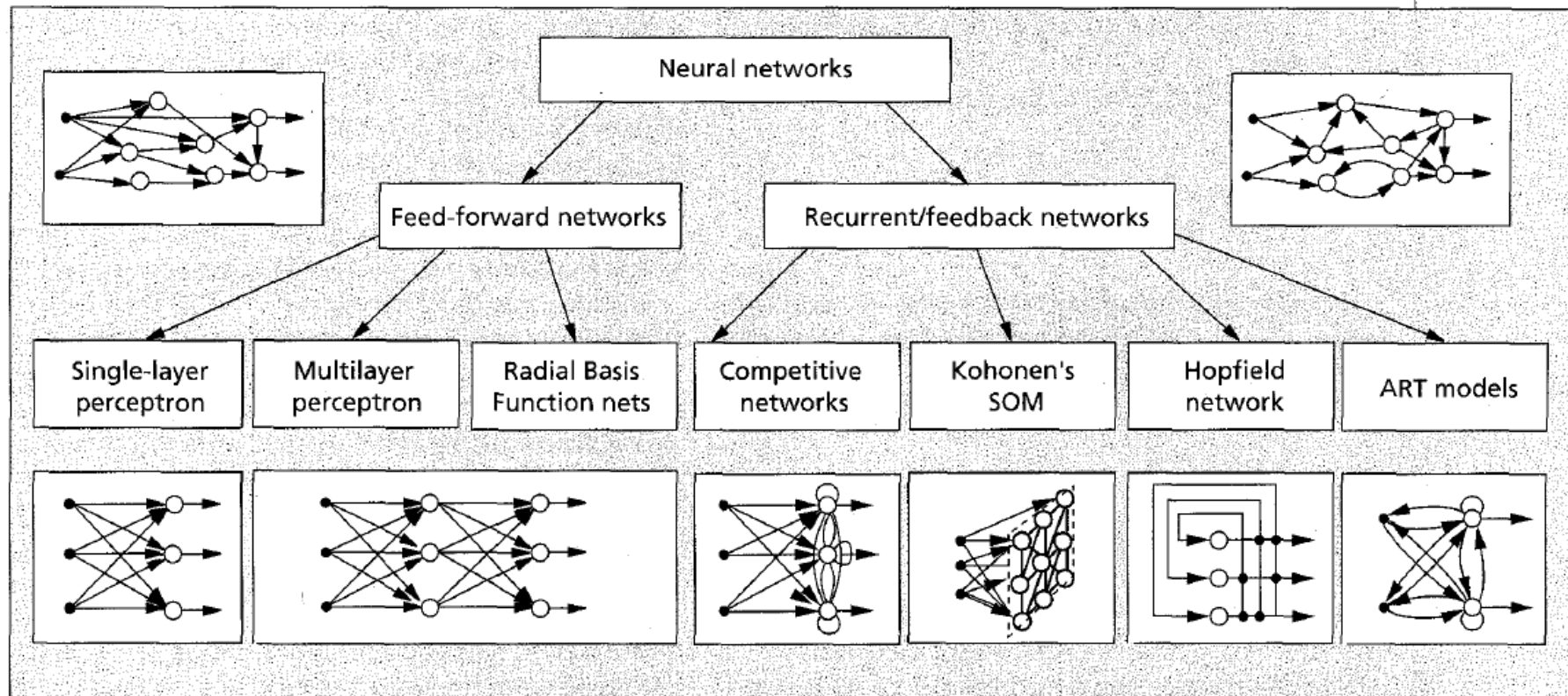
- Badania nad sieciami neuronowymi miały już swoje początki w latach czterdziestych. McCulloch i Pitts w 1943 roku przedstawili pierwszy formalny model neuronu w postaci układu arytmetyczno-logicznego. Jego idea przetrwała lata i stanowi do dzisiaj podstawowe ogniwo większości używanych modeli. Istotnym elementem tego modelu jest sumowanie sygnałów wejściowych z odpowiednią wagą i poddanie otrzymanej sumy działaniu nieliniowej funkcji aktywacji. Sieci wykorzystujące ten model nie zostały praktycznie zrealizowane w ówczesnej technice, jednak wywarły wpływ na późniejszy rozwój tej dziedziny.
- W kilka lat później Hebb (1949) odkrył, że informacja może być przechowywana w strukturze połączeń pomiędzy neuronami, i jako pierwszy zaproponował metodę uczenia sieci polegającą na zmianach wag połączeń pomiędzy neuronami. Zaproponowana metoda nazywana jest dzisiaj regułą Hebba.

Historia SN

- W latach pięćdziesiątych powstały pierwsze sieci neuronowe. Szeroko znanym przykładem działającej sieci jest **perceptron**. Zbudowany został w roku **1957** przez Franka **Rosenblatta** i Charlesa **Wightmana** w Cornell Aeronautical Laboratory. Została ona określona jako maszyna klasyfikująca obrazy i zdolna do uczenia się poprzez modyfikację połączeń prowadzących do układów progowych. Te prace stworzyły zaczątki wielu do dzisiaj stosowanych metod uczenia.
- W latach **1982-1986** opublikowano szereg oryginalnych i ważnych prac, które w znaczący sposób wpłynęły na dalszy rozwój badań nad sieciami neuronowymi. Okres renesansu rozpoczął się, gdy **Hopfield** (**1982, 1984**) wprowadził **rekurencyjną architekturę pamięci neuronowych**. W swoich pracach opisał własności obliczeniowe ze sprzężeniem zwrotnym.

Historia SN

- W Polsce wydano szereg książek zajmujących się problematyką sieci neuronowych, dotyczących szczególnie rozpoznawania obrazów i metod uczenia, wśród których należy wymienić monografie Gawrońskiego (1970), Kulikowskiego (1972) i Brodziaka (1974).
- Nowa fala zainteresowania sieciami neuronowymi, która rozpoczęła się w połowie lat osiemdziesiątych, znajduje też odbicie w Polsce, gdzie wydano monografię Tadeusiewicza (1992), Korbicza, Obuchowicza i Ucińskiego (1994), Kacprzaka i Ślota (1995) oraz przetłumaczono monografię Hertza, Krogha i Palmera (1993).
- W maju 1994 roku odbyła się też zorganizowana przez Politechnikę Częstochowską pierwsza krajowa konferencja naukowa poświęcona sieciom neuronowym (KKSNIŻ, 1994).

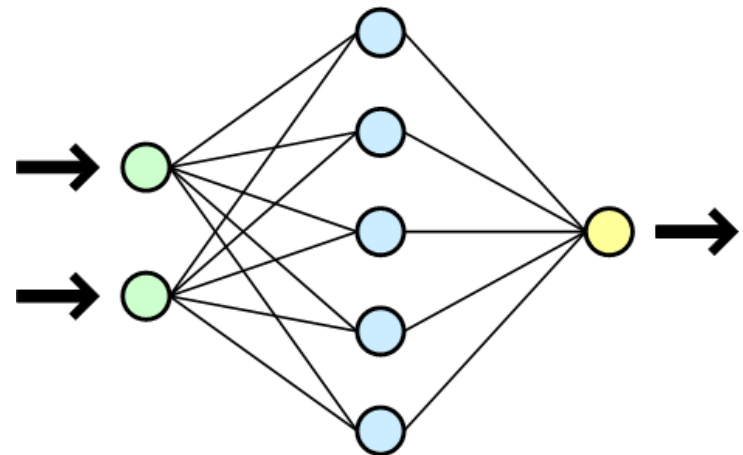


Typy sieci neuronowych

- **Jednokierunkowe** (jedno błądź wielowarstwowe) np. sieć typu Perceptron
- **Rekurencyjne** np. sieć dyskretna Hopfielda

Jak rozróżniać sieci ?

- Modele sieci neuronowych różnią się zarówno rodzajem neuronów, z których są zbudowane jak i topologią (układem) połączeń międzyneuralnych oraz sposobem przesyłania sygnałów w obrębie sieci.

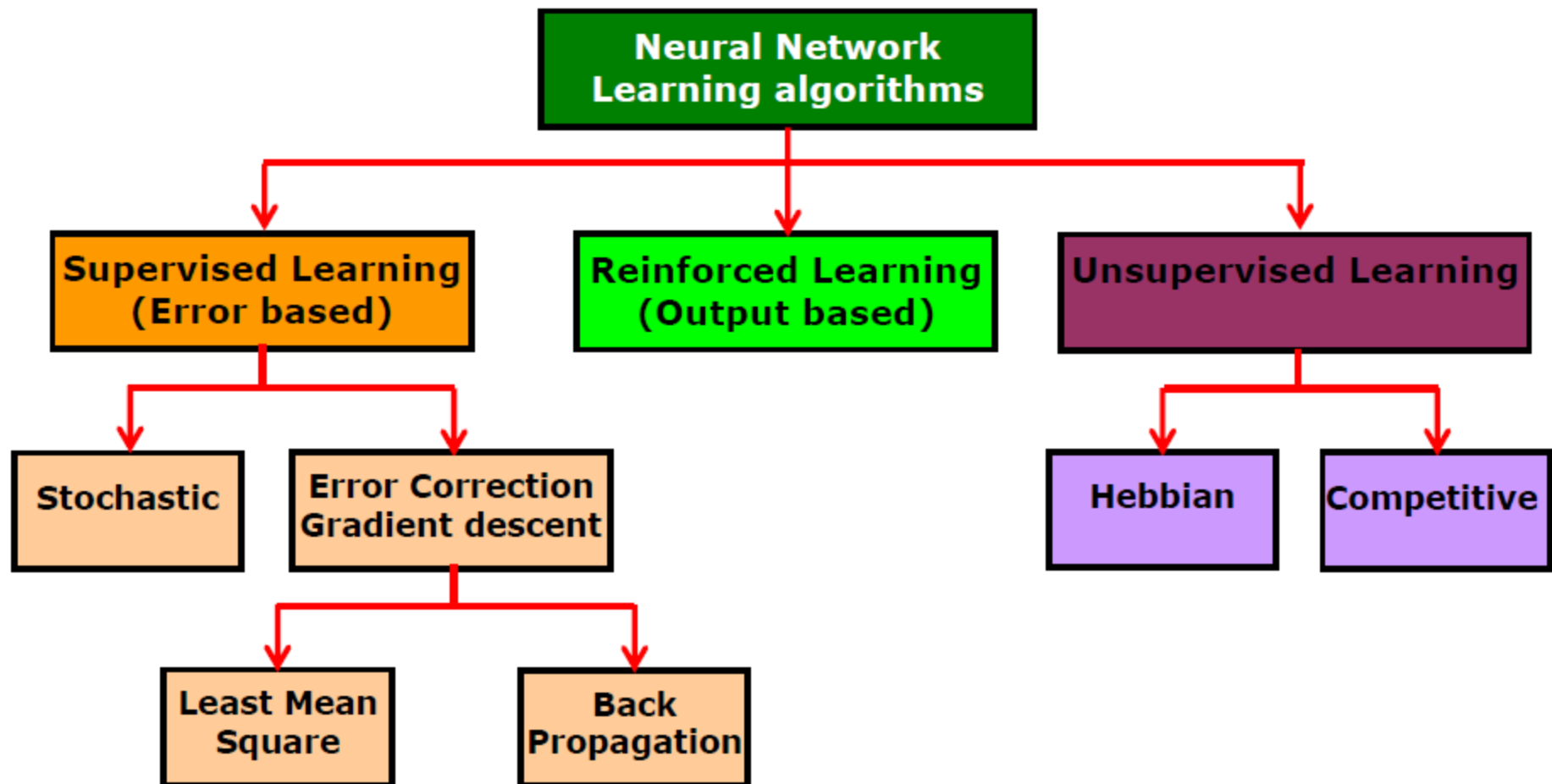


Uczenie się SN

- sieci neuronowe mają zdolność do uczenia się, czyli zdolność do samodzielnego dostosowywania współczynników wagowych. czyli inaczej mówiąc uczenie sieci jest to wymuszenie na niej określonego zareagowania na sygnały wejściowe. Dzięki temu mówi się że mają one właśnie charakter AI, bo potrafią samodzielnie dostosować się do zmieniających się warunków. **Celem uczenia jest taki dobór wag w poszczególnych neuronach aby sieci mogła rozwiązywać stawiane przed nią problemy.**
- Z punktu widzenia systemu uczącego sieci neuronowej można wyróżnić:
 - uczenie pod nadzorem (z nauczycielem),
 - uczenie z krytykiem,
 - uczenie samoorganizujące się (bez nadzoru).

Uczenie polega na modyfikacji wag.

- Działanie sieci zależy od przyjętego modelu neuronu, ale i od wartości parametrów i od topologii.
- Wiedza zawarta jest w parametrach (wagach) bo od nich zależy z jaką siłą dana zmienna będzie wprowadzona do neuronu. Nie ma reguły na to czy lepsze są dodatnie czy ujemne wartości wag.
- Neurony w danej warstwie się nie komunikują.



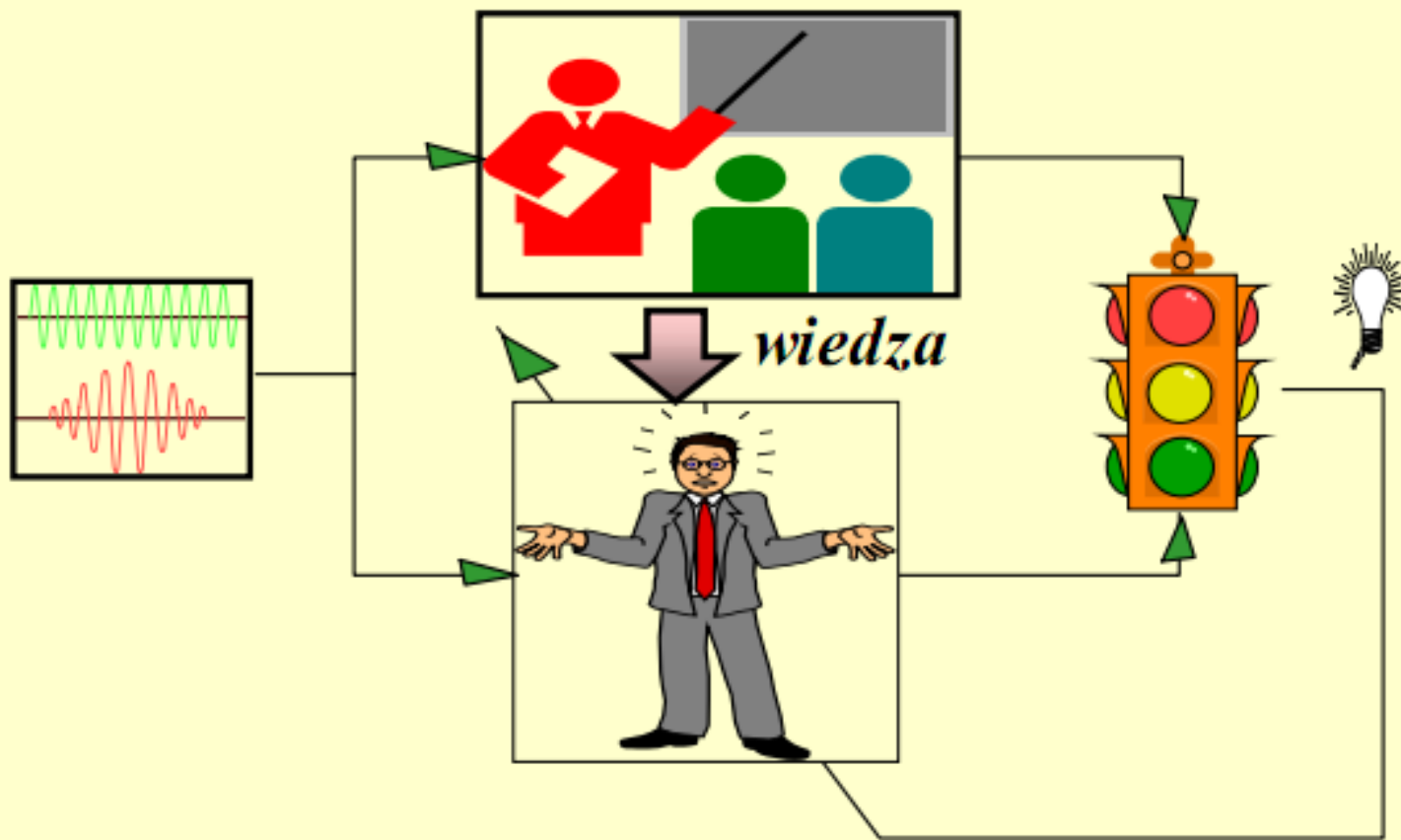
Typy uczenia się

- **Uczenie nadzorowane** – uczenie maszynowe, które zakłada obecność ludzkiego nadzoru nad tworzeniem funkcji odwzorowującej wejście systemu na jego wyjście.
- Nadzór polega na stworzeniu zestawu danych uczących, czyli par:
- wejściowy obiekt uczący (np. wektor);
- pożądana przez nadzorcę (nauczyciela) odpowiedź (np. jakaś konkretna wartość liczbową).
- Zadaniem systemu jest nauczenie się przewidywania prawidłowej odpowiedzi na zadane pobudzenie oraz generalizacja przypadków wyuczonych na przypadki, z którymi system jeszcze się nie zetknął. Do modelowania procesów technicznych zwykle wykorzystuje się uczenie nadzorowane.
- **Uczenie nienadzorowane** – uczenie maszynowe, które zakłada brak obecności ludzkiego nadzoru nad tworzeniem funkcji odwzorowującej wejście systemu na jego wyjście.

uczenie pod nadzorem (z nauczycielem)

- Uczenie z nadzorem odbywa się pod nadzorem zewnętrznego nauczyciela. On otrzymuje informację jaka powinna być na wyjściu sieci dla podanego na wejściu przykłady. W kolejnym kroku sprawdza czy uzyskana odpowiedź z sieci jest właściwa. Jeśli tak to wynik jest odpowiedzialna a jeśli nie to nauczyciel udziela odpowiedzi (zmiana wag). Proces jest powtarzany tak długo jak nie będzie prawidłowej odpowiedzi sieci.
- Podczas uczenia z nauczycielem, przy nowych danych wejściowych nauczyciel podpowiada jaka powinna być poprawna odpowiedź, dla danego przykładu.

Schemat uczenia nadzorowanego



W trakcie uczenia, wiedza nauczyciela przekazywana jest to obiektu uczącego się.

uczenie pod nadzorem (z nauczycielem)

- Umożliwia mu to ocenę, czy aktualny stan sieci neuronowej na wyjściu jest właściwy, czy nie. Jeśli jest właściwy to nie następuje żadne działanie, jeśli jednak nie, wówczas następuje zmiana wag W aby różnice między poprawną, a rzeczywistą odpowiedzią były jak najmniejsze.
- W kolejnych cyklach uczących sieć dobiera wagi w taki sposób aby jej odpowiedzi były możliwie najbardziej dokładne z wzorcami uczącymi. Istotną cechą tego procesu jest istnienie sprzężenia zwrotnego, umożliwiającego korelację wag w sieci.
- Odległość pomiędzy rzeczywistą, a pożądaną odpowiedzią sieci jest miarą błędu używaną do korekcji parametrów sieci. Dostrajanie elementów macierzy wag W może być oparte na systemie nagród i kar stosowanych przez nauczyciela w celu zmniejszenia błędu, jak to ma miejsce w naturalnym nauczaniu.
- Zestaw obrazów wejściowych i wyjściowych użytych w czasie nauki nazywamy zbiorem uczącym. Często zbiór uczący jest realizacją procesu przypadkowego i procedura minimalizacji błędu musi uwzględniać jego własności statystyczne. W rezultacie większość algorytmów uczenia z nauczycielem sprowadza się do statystycznej minimalizacji błędu w wielowymiarowej przestrzeni wag.

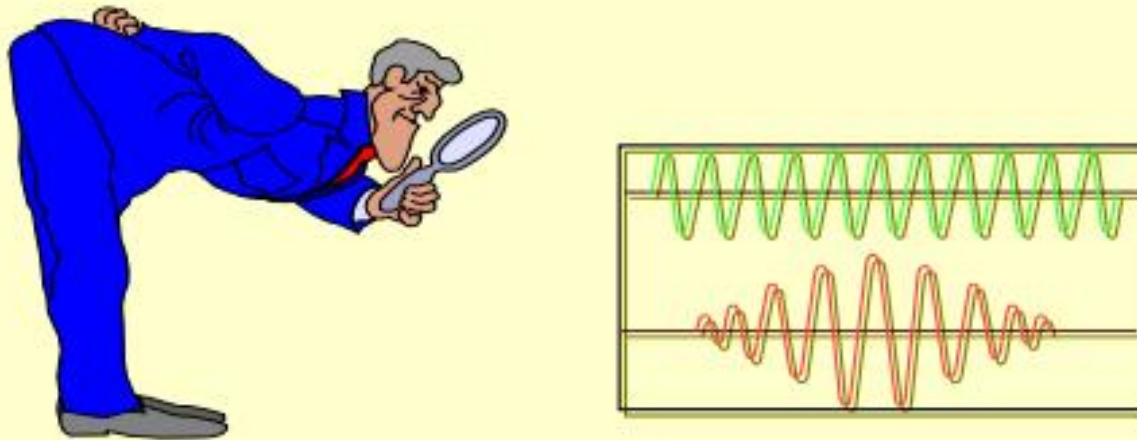
Uczenie bez nadzoru

- uczenie bez nadzoru - nie ma możliwości śledzenia i oceny poprawności odpowiedzi. Nie ma ani nauczyciela ani krytyka.
- Uczenie następuje zgodnie z określonym działaniem sieci, umożliwiającym jej samodzielne wykrywanie wszelkich regularności, linii podziału i innych ogólnych charakterystyk danych wejściowych.
- W trakcie ich wykrywania parametry sieci podlegają zmianom co nazywany samoorganizacją.
- Jej zdolności do wykrywania skupisk obrazów wejściowych są wykorzystywane do ich klasyfikacji w przypadkach gdy klasy nie są z góry ustalone.

Uczenie nienadzorowane

- Samouczenie nie wymaga żadnej jawnie podawanej do sieci neuronowej zewnętrznej wiedzy, a sieć zgromadzi wszystkie potrzebne informacje i wiadomości.
- sieci pokazuje się kolejne przykłady sygnałów wejściowych, nie podając żadnych informacji o tym, co z tymi sygnałami należy zrobić. Sieć obserwuje otoczenie i odbiera różne sygnały. Nikt nie określa jednak, jakie znaczenie mają pokazujące się obiekty i jakie są pomiędzy nimi zależności.
- Sieć na podstawie obserwacji występujących sygnałów stopniowo sama odkrywa, jakie jest ich znaczenie i również sama ustala zachodzące między sygnałami zależności.
- Po podaniu do sieci neuronowej każdego kolejnego zestawu sygnałów wejściowych tworzy się w niej pewien rozkład sygnałów wyjściowych – różnie są neurony pobudzone: słabiej, bądź bardzo silnie, a więc niektóre neurony "rozpoznają" podawane sygnały jako "własne" (czyli takie, które są skłonne akceptować), inne traktują je "obojętnie", zaś jeszcze u innych neuronów wzbudzają one wręcz "awersję".

Schemat uczenia nienadzorowanego



Metody analizy danych są wypracowywane samodzielnie przez system uczący się, na podstawie informacji o dostarczonych danych

Uczenie z krytykiem

- uczenie z krytykiem, zwane również ze wzmocnieniem jest odmianą uczenia się pod nadzorem, w którym nie występuje inf. o wartościach żądanych na wyjściu a jedynie informacja czy podjęta przez system akcja (np. zmiana wag) daje wyniki pozytywne czy negatywne. jeśli działanie daje wynik pozytywny to następuje wzmocnienie tendencji do właściwego zachowania się systemu w podobnych sytuacjach w przyszłości. w przeciwnym przypadku gdy rezultat jest negatywny to należy tak modyfikować wartości wag aby tę tendencję osłabić.
- Uczenie z krytykiem w odróżnieniu od uczenia pod nadzorem ocenia skutki podjętej akcji w i zależności od tego oraz aktualnej bazy danych podejmuje decyzje co do dalszej akcji. Jest znacznie bardziej uniwersalne w zastosowaniu gdyż nie wymaga obecności sygnałów żądanych na wyjściu systemu. jednocześnie jego realizacja praktyczna jest bardziej skomplikowana.

Wady samouczenia się

- W porównaniu z procesem uczenia z nauczycielem samouczenie jest zwykle znacznie powolniejsze.
- Bez nauczyciela nie można z góry określić, który neuron wyspecjalizuje się w rozpoznawania której klasy sygnałów. Stanowi to pewną trudność przy wykorzystywaniu i interpretacji wyników pracy sieci.
- Nie można określić, czy sieć uczona w ten sposób nauczy się wszystkich prezentowanych jej wzorców. Dlatego sieć przeznaczona do samouczenia musi być większa niż sieć wykonująca to samo zadanie, ale trenowana w sposób klasyczny, z udziałem nauczyciela. Szacunkowo sieć powinna mieć co najmniej trzykrotnie więcej elementów warstwy wyjściowej niż wynosi oczekiwana liczba różnych wzorów, które sieć ma rozpoznawać.

Reguły uczenia SN

- Neuron ma zdolności adaptacji. Jego wagi podlegają modyfikacji podczas uczenia.
- Ogólna zasada nauki przyjęta dla sieci brzmi:

wektor wag w_i rośnie proporcjonalnie do iloczynu sygnałów wejściowego x i uczącego r . Sygnał uczący r jest w ogólnej postaci funkcją w_i , x i czasami sygnału nauczyciela d_i .

Reguły uczenia SN

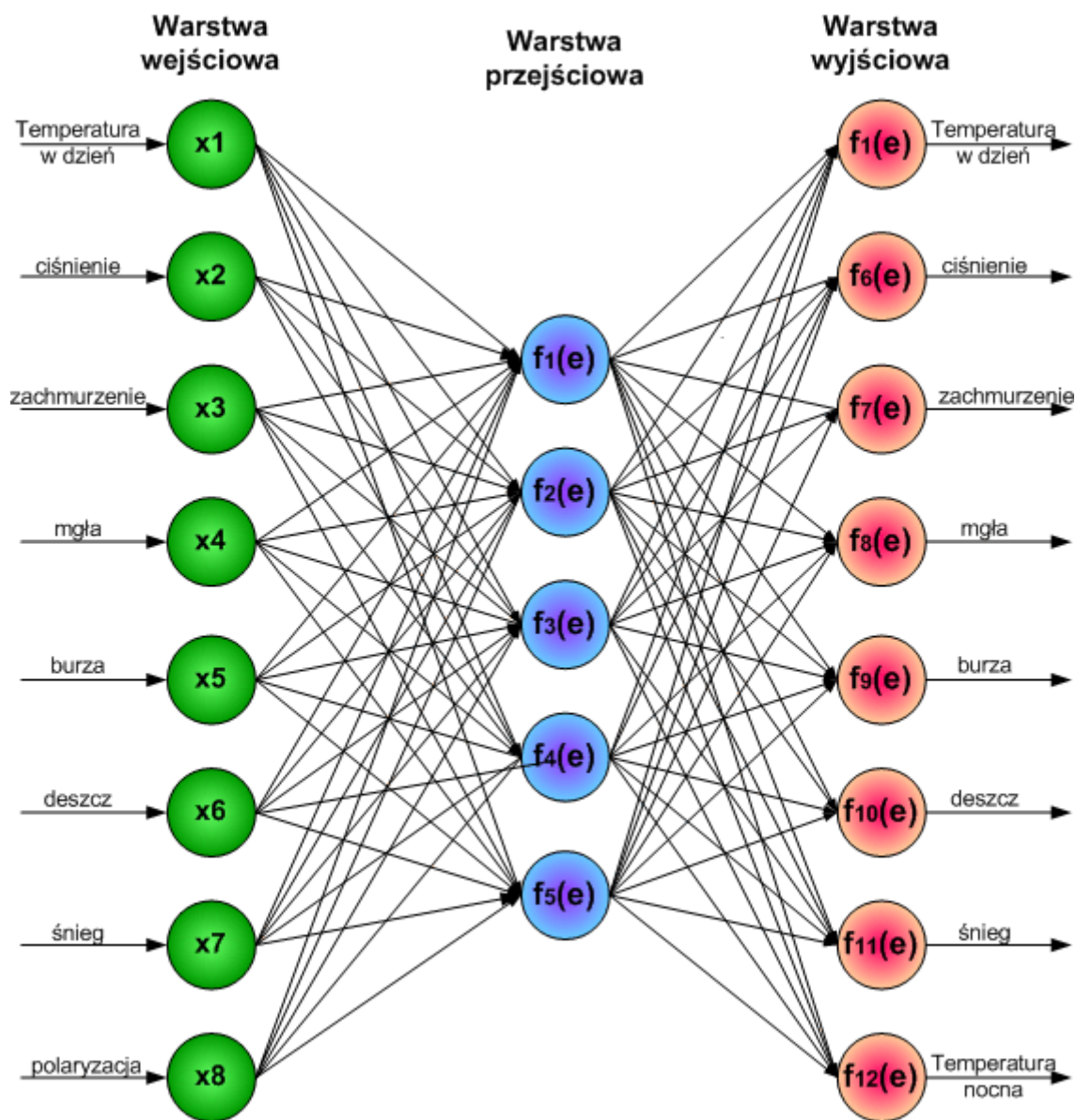
Wyróżniamy:

- **reguła Hebba** (bez nauczyciela, sygnałem uczącym jest sygnał wyjściowy)
- **reguła perceptronowa** (z nauczycielem, sygnał uczący jest różnicą między wartością rzeczywistą a pożądaną)
- **reguła delta** (dla neuronów z ciągłymi funkcjami aktywacji i nadzorowania trybu uczenia). Chodzi o minimalizację kwadratowego kryterium błędu.
- **reguła korelacyjna** (poprawka każdej składowej wektora wag jest proporcjonalna do iloczynu odpowiedniej składowej obrazu wejściowego i pożądanego przy tym wzorca wyjścia)
- **Reguła 'wygrywający bierze wszystko'** różni się zdecydowanie od pozostałych tu opisanych. Jest ona przykładem nauki z rywalizacją stosowanej zazwyczaj do poznawania własności statystycznych sygnałów wejściowych w trybie bez nauczyciela.

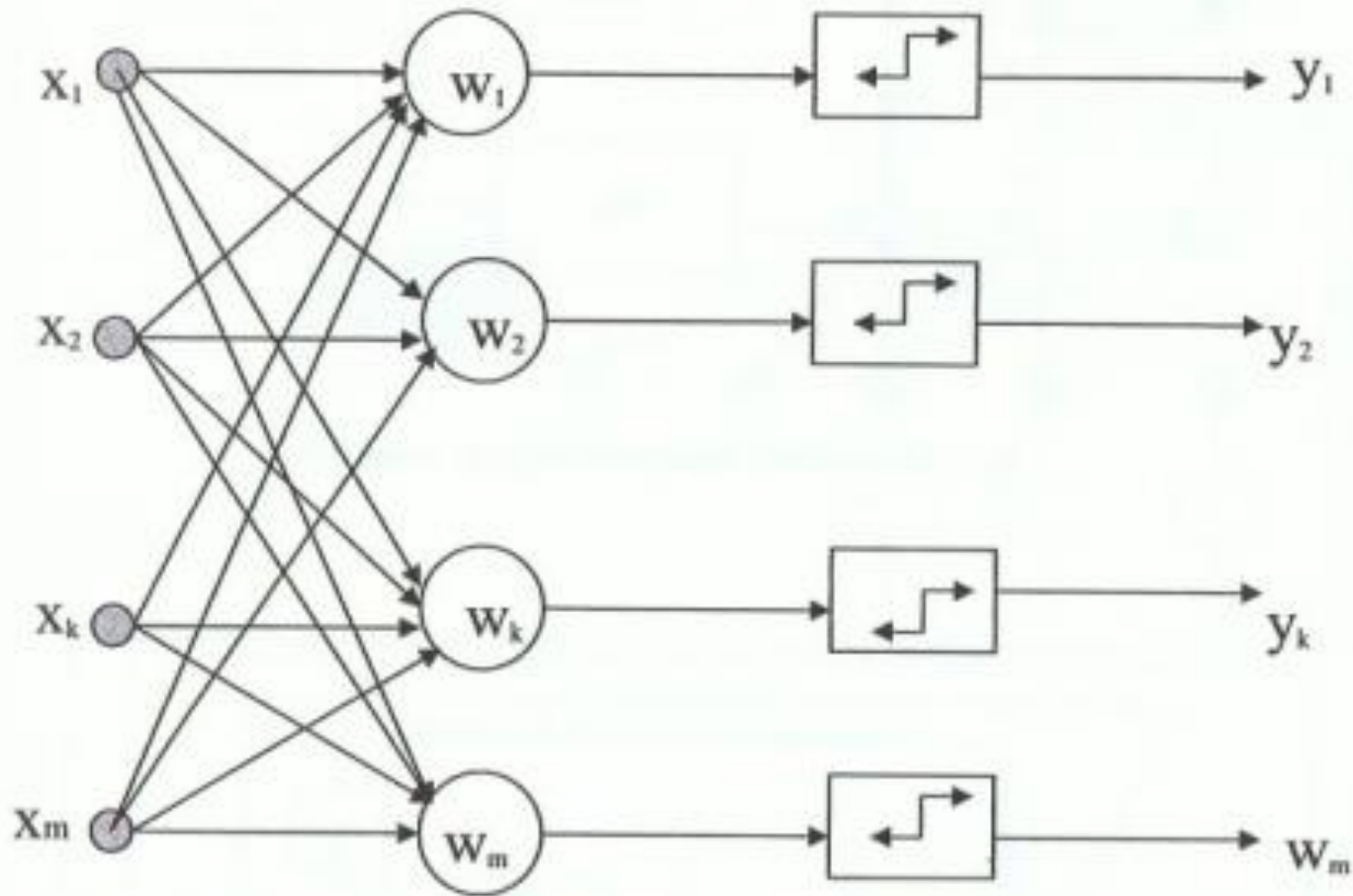
Paradigm	Learning rule	Architecture	Learning algorithm	Task
Supervised	Error-correction	Single- or multilayer perceptron	Perceptron learning algorithms	Pattern classification
			Back-propagation	Function approximation
			Adaline and Madaline	Prediction, control
	Boltzmann	Recurrent	Boltzmann learning algorithm	Pattern classification
	Hebbian	Multilayer feed-forward	Linear discriminant analysis	Data analysis Pattern classification
Unsupervised	Competitive	Competitive	Learning vector quantization	Within-class categorization
				Data compression
		ART network	ARTMap	Pattern classification
				Within-class categorization
	Error-correction	Multilayer feed-forward	Sammon's projection	Data analysis
Unsupervised	Hebbian	Feed-forward or competitive	Principal component analysis	Data analysis
				Data compression
		Hopfield Network	Associative memory learning	Associative memory
	Competitive	Competitive	Vector quantization	Categorization
				Data compression
	Kohonen's SOM	Kohonen's SOM	Categorization	
			Data analysis	
Hybrid	Error-correction and competitive	ART networks	ART1, ART2	Categorization
		RBF network	RBF learning algorithm	Pattern classification
				Function approximation
			Prediction, control	

Sieci jednokierunkowe (np. Perceptron)

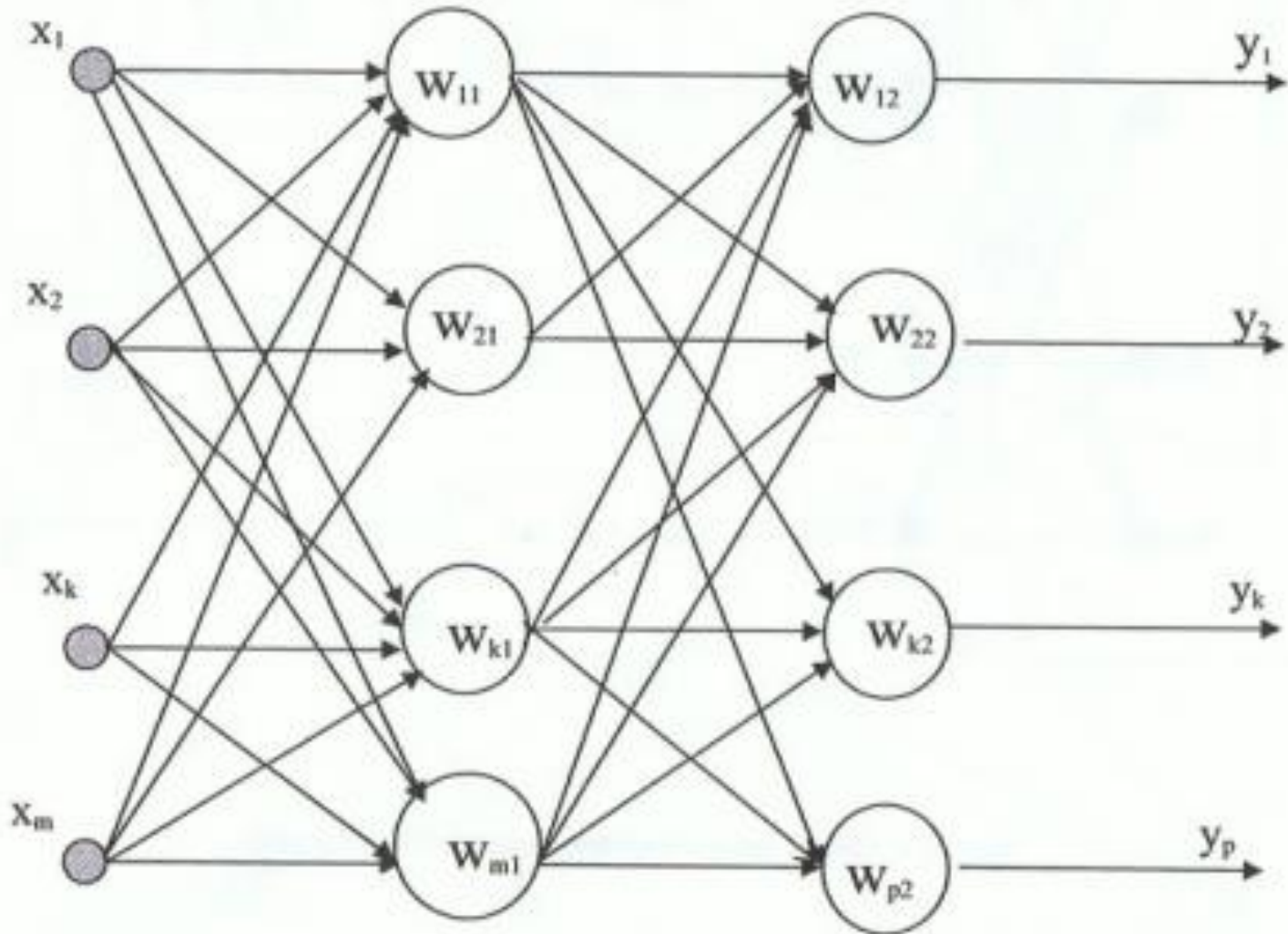
- zbudowane są z jednej bądź kilku warstw.
- Przepływ sygnału w tego typu sieciach przebiega zawsze w ściśle określonym kierunku: od warstwy wejściowej do warstwy wyjściowej.
- Każda dodatkowa warstwa pomiędzy warstwami wejściową i wyjściową nazywana jest *warstwą ukrytą* z uwagi na to, że jej działalność nie może być obserwowana bezpośrednio ani na wejściu sieci ani na jej wyjściu.
- Zwykle wszystkie neurony warstwy poprzedniej połączone są ze wszystkimi neuronami warstwy następnej.
- Do uczenia perceptronów wielowarstwowych stosuje się algorytmy spadku gradientowego, między innymi algorytm propagacji wstecznej.



Sieć jednokierunkowa jednowarstwowa

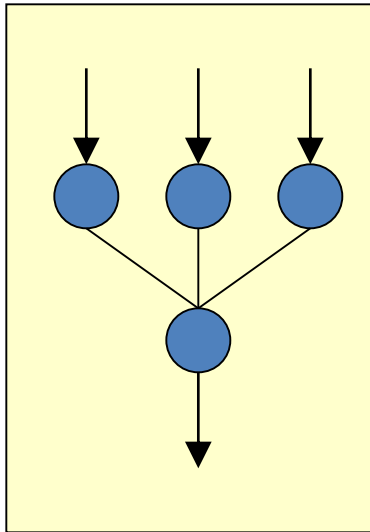


Sieć jednokierunkowa dwuwarstwowa

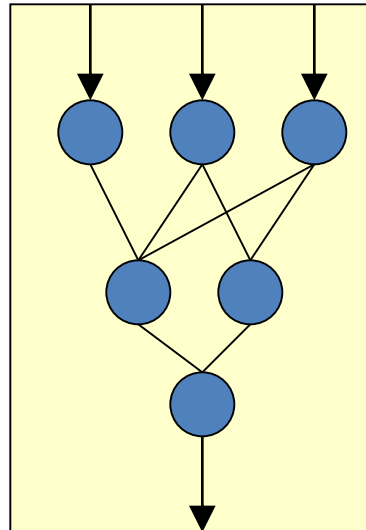


Różne liczby warstw ukrytych

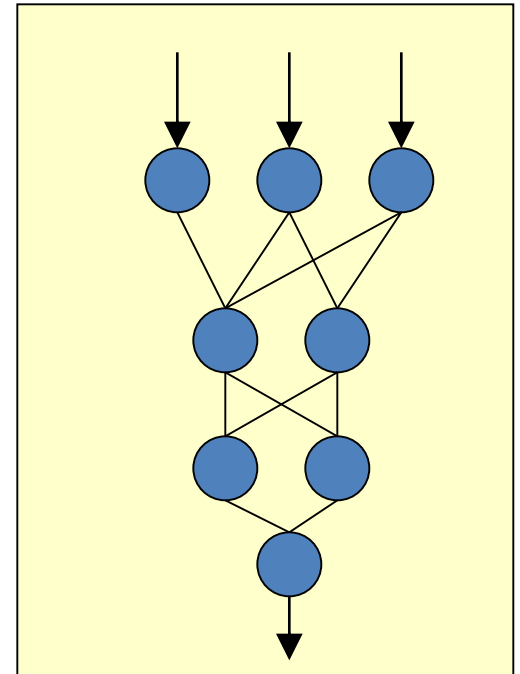
brak



jedna



więcej

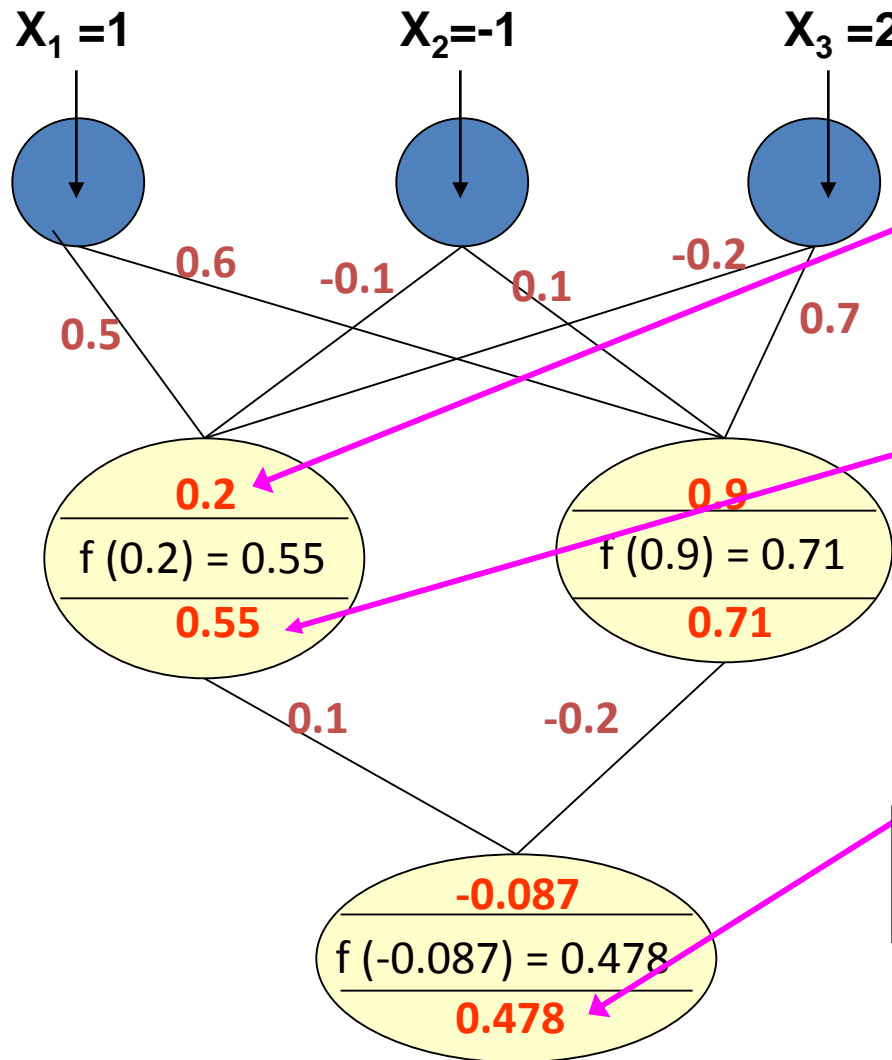


Predykcja

Wejścia: X_1 X_2 X_3

Wyjścia: Y

Model: $Y = f(X_1 \ X_2 \ X_3)$



$$0.2 = 0.5 * 1 - 0.1 * (-1) - 0.2 * 2$$

$$f(x) = e^x / (1 + e^x)$$
$$f(0.2) = e^{0.2} / (1 + e^{0.2}) = 0.55$$

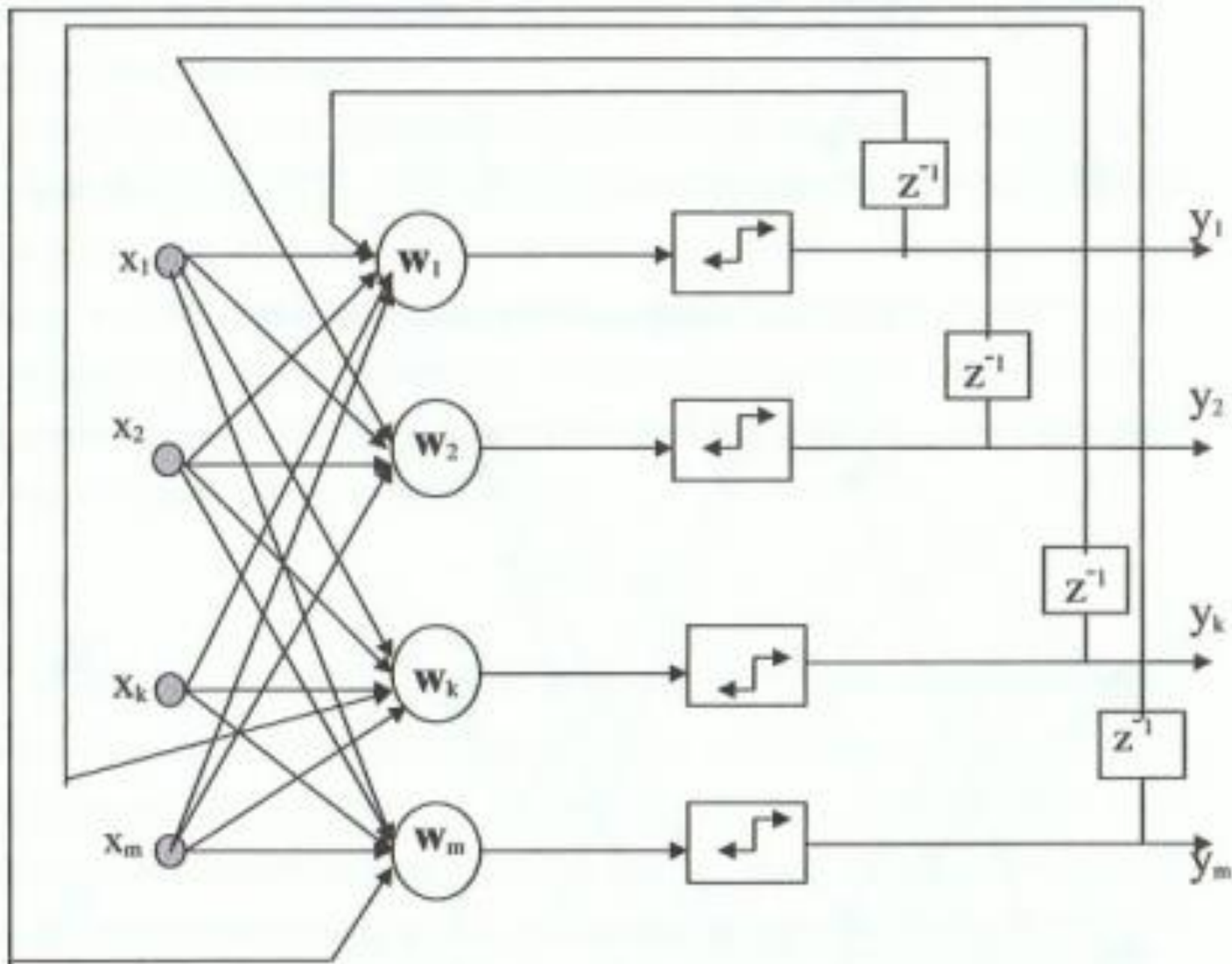
Predykcja $Y = 0.478$

Jeśli aktualnie $Y = 2$
To błąd predykcji $= (2 - 0.478) = 1.522$

Sieci rekurencyjne

- Połączenia między neuronami stanowią graf z cyklami (obieg zamknięty) tzn. sygnały z warstwy wyjściowej sieci podawane są z powrotem do warstwy wejściowej. Wśród różnorodności modeli rekurencyjnych sztucznych sieci neuronowych wyróżnić można:
 - sieć Hopfielda – układ gęsto połączonych ze sobą neuronów (każdy z każdym, ale bez połączeń zwrotnych) realizującą dynamikę gwarantującą zbieżność do preferowanych wzorców
 - maszyna Boltzmanna –modyfikacja sieci Hopfielda; pozwoliła na uczenie neuronów ukrytych i likwidację wzorców pasożytniczych kosztem zwiększenia czasu symulacji.
- Sieci Hopfielda i maszyny Boltzmanna stosuje się jako pamięci adresowane kontekstowo, do rozpoznawania obrazów, rozpoznawania mowy, a także do rozwiązywania problemów minimalizacji (np. problemu komiwojażera).

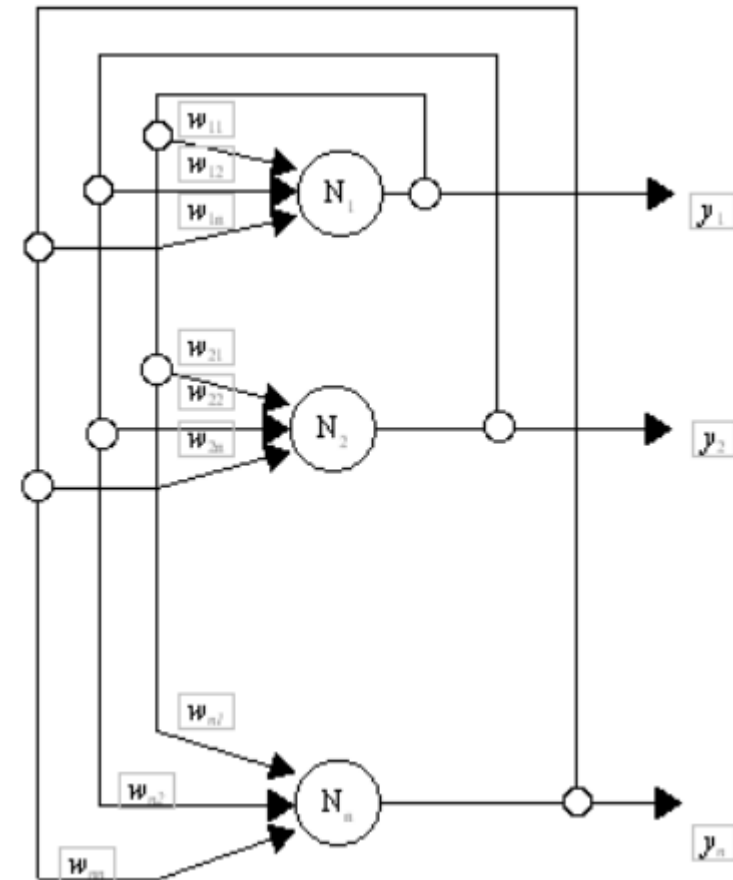
Model sieci rekurencyjnej jednowarstwowej



Sieć Hopfielda

- architektura sieci charakteryzuje się sprzężeniem zwrotnym między warstwami sieci.
- zależności dynamiczne występują na każdym etapie działania, a zmiana stanu jednego neuronu przenosi się na całą sieć.
- składa się z n neuronów, przy czym w danej chwili czasu t aktywny jest tylko jeden. Każdy neuron wybierany jest z jednakowym prawdopodobieństwem, a zmiany stanu neuronu następują w dyskretnych chwilach czasu.
- stan i -tego neuronu może być opisany następująco:

$$y_i(t+1) = f\left(\sum_{j=0}^n (w_{ij} y_j(t) + b_i)\right)$$



przy warunku początkowym $y_j(0)=x_j$, przy czym: x_j oznacza wektor wejściowy, $y_i(t+1)$ oznacza stan i -tego neuronu w chwili $t+1$, w_{ij} oznacza macierz wag, b_i oznacza bias, f oznacza funkcję aktywacji.

Sieć Hopfielda c.d.

- Jako funkcję aktywacji przyjmuje się funkcję signum:

$$f(y_j(t)) = \begin{cases} 1, & \text{gdy } \sum_{i=0}^n w_{ij} y_i(t) + b_j > 0 \\ 0, & \text{gdy } \sum_{i=0}^n w_{ij} y_i(t) + b_j \leq 0 \end{cases}$$

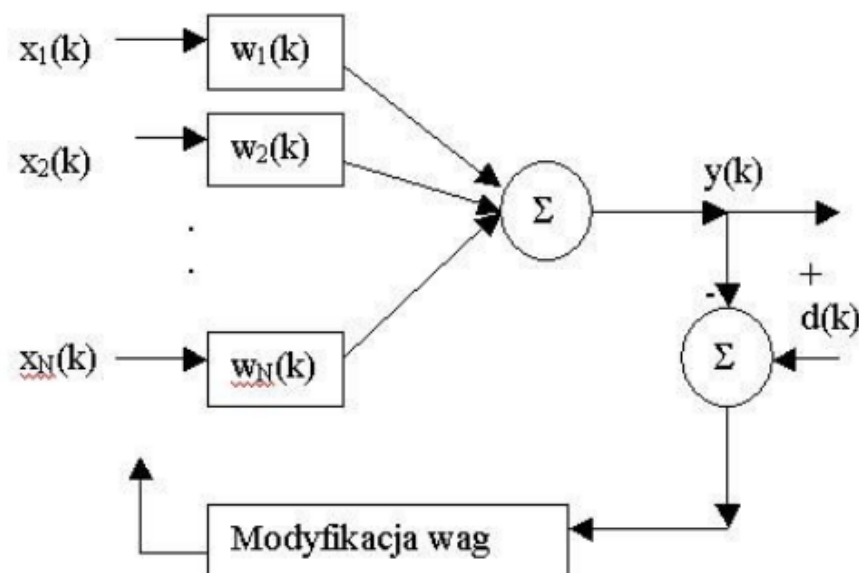
- W działaniu sieci Hopfielda wyróżnia się dwa tryby:
 - Uczenia - na podstawie dostarczonych do sieci wzorców uczących x_j dobierane są wagi w_{ij} . Proces uczenia sieci kształtuje obszary przyciągania (atrakcji) poszczególnych punktów równowagi, które odpowiadają danym uczącym.
 - i odtwarzania - po ustaleniu wartości wag następuje proces przebiegający według zależności, kończący się w określonym minimum lokalnym, w którym $y(t+1)=y(t)$.

Sieć Adaline (ang. Adaptive Linear Neuron)

- Układ został zaproponowany w 1960 roku przez Widrowa i Hoffa.
- Nazywany jest również adaptacyjnym liniowym sumatorem ważonym

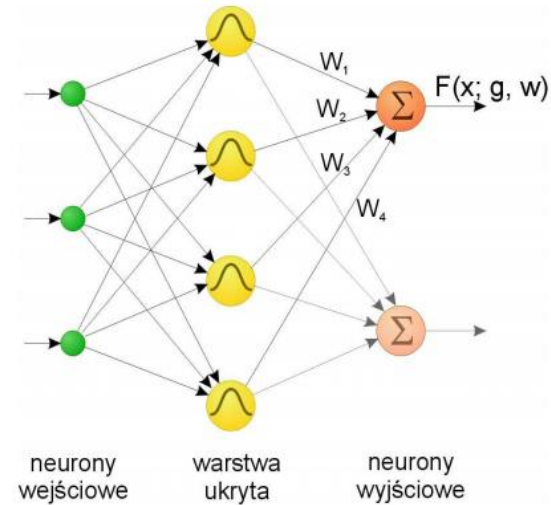
Algorytm modyfikacji wag ma charakter uczenia pod nadzorem. Sygnał wyjściowy y sumatora porównywany jest z sygnałem wzorcowym d . Uzyskany w ten sposób błąd sumatora wykorzystany jest do zmodyfikowania wartości wag zgodnie ze wzorem:

$$w(k+1) = w(k) + \eta x(k)[d(k) - w^T(k)x(k)]$$



Radialne sieci neuronowe (RBF, radial based function)

- 1988 r. Broomhead i Lowe.
- W sieci takiej znajdują się neurony, których pobudzenie zależy od odległości (a więc podobieństwa) sygnału wyjściowego od pewnego centrum, które jest jednym z parametrów neuronu. Czyli, dany neuron reaguje jedynie na bodźce podobne do pewnego z góry ustalonego bodźca zapisanego w neuronie.
- Sieć taka może zawierać dowolną liczbę N wejść, określoną liczbę neuronów ukrytych o radialnej funkcji aktywacji oraz tyle neuronów wyjściowych, ile wynosi wymiar wektora zadanego d , wynikający ze specyfiki problemu.
- Neurony wyjściowe są zwykle liniowe i każdy z nich realizuje sumę wagową sygnałów pochodzących z neuronów radialnych warstwy ukrytej.



Funkcja aktywacji $\phi(x)$ neuronu ukrytego w sieciach RBF dokonuje odwzorowania oryginalnego punktu zadanego lub grupy takich punktów stanowiących grupę wokół centrum funkcji gaussowskiej.

Uczenie się sieci radialnej

Neurony warstwy ukrytej wzbudzane są za pomocą funkcji zależnej od odległości pomiędzy danym punktem x a pewnym centrum c , które jest jednym z parametrów neuronu ustalanych w procesie nauki.

Wzbudzenie i -tego neuronu jest najczęściej funkcją gaussowską określoną wzorem:

$$g(x, c, b) = e^{-\|x - c\|^2 / 2b^2}$$

Czasami stosuje się inne funkcje bazowe np.:

B – to wsp. rozmycia (zasięg funkcji),
 $\alpha > 0$, oraz $0 < \beta < 1$ są dodatkowymi
parametrami neuronu.

$$g(x, c, b) = \|x - c\| / b^2$$

$$g(x, c, b) = (b^2 + \|x - c\|^2)^{-\alpha}$$

$$g(x, c, b) = (b^2 + \|x - c\|^2)^\beta$$

$$g(x, c, b) = (b\|x - c\|)^2 \ln(b\|x - c\|)$$

Różnice między MLP a RBF

- RBF obok sieci jednokierunkowej wielowarstwowej to najpopularniejsza sieć, wykorzystująca gaussowską funkcję radialną.
- Sieć ta różni się od MLP tym, że:
 - w MLP jest aproksymacja typu globalnego (każdy neuron ma wpływ na wynik odwzorowania w całej przestrzeni danych)
 - w RBF – typu lokalnego – neuron jest aktywny tylko w ograniczonym obszarze skupionym wokół centrum funkcji gaussowskiej.

Zalety sieci radialnych:

- Uproszczony algorytm uczenia,
- istnieje tylko jedna warstwa ukryta,
- istnieje ścisłe powiązanie neuronu z odpowiednim obszarem przestrzeni danych uczących, punkt startowy uczenia może być położony znacznie bliżej rozwiązania optymalnego.

Samooorganizujące się mapy

- Samooorganizujące się mapy (*Self Organizing Maps*, SOM), zwane też sieciami Kohonena, to sieci neuronów, z którymi są stowarzyszone współrzędne na prostej, płaszczyźnie lub w dowolnej n -wymiarowej przestrzeni.
- Uczenie tego rodzaju sieci polega na zmianach współrzędnych neuronów, tak, by dążyły one do wzorca zgodnego ze strukturą analizowanych danych. Sieci zatem "rozpinają się" wokół zbiorów danych, dopasowując do nich swoją strukturę.
- Sieci te stosowane są do klasyfikacji wzorców, np. głosek mowy ciągłej, tekstu, muzyki. Do najciekawszych zastosowań należy rozpinanie siatki wokół komputerowego modelu skanowanego obiektu.

Sieci nadzorowane:

- Jednokierunkowe:
 - Liniowe:
 - Hebb - Hebb (1949), Fausett (1994);
 - Perceptron - Rosenblatt (1958), Minsky i Papert (1969/1988), Fausett (1994);
 - Adaline - Widrow i Hoff (1960), Fausett (1994);
 - wyższego uporządkowania (ang. Higher Order) - Bishop (1995);
 - połączenie funkcjonalne (ang. Functional Link) - Pao (1989).
 - MLP (ang. Multi-Layered Perceptron) wielowarstwowy perceptron - Bishop (1995), Reed i Marks (1999), Fausett (1994):
 - wstecznej propagacji (ang. Back Propagation) - Rumelhart, Hinton i Williams (1986);
 - korelacji kaskadowej (ang. Cascade Correlation) - Fahlman i Lebiere (1990), Fausett (1994);
 - szybkiej propagacji (ang. Quickprop) - Fahlman (1989);
 - RPROP (ang. Resilient Back PROPagation) - Riedmiller i Braun (1993).
 - Sieci wyłącznie klasyfikujące:
 - LVQ (ang. Learning Vector Quantization) - Kohonen (1988), Fausett (1994);
 - PNN (ang. Probabilistic Neural Network) - Specht (1990), Masters (1993), Hand (1982), Fausett (1994).
- Rekurencyjne - Hertz, Krogh i Palmer (1991), Medsker i Jain (2000):
 - BAM (ang. Bidirectional Associative Memory) Dwukierunkowa Pamięć Asocjacyjna - Kosko (1992), Fausett (1994);
 - Maszyna Boltzmana (ang. Boltzman Machine) - Ackley (1985), Fausett (1994).
- Oparte na zasadzie rywalizacji:
 - ARTMAP, sieć rezonansowa - Carpenter, Grossberg i Reynolds (1991);
 - CP (ang. Counterpropagation) "Sieć przesyłająca żetony" - Hecht-Nielsen (1987; 1988; 1990), Fausett (1994);
 - Neocognitron - Fukushima, Miyake i Ito (1983), Fukushima, (1988), Fausett (1994).

Sieci nienadzorowane - Hertz, Krogh i Palmer (1991)

- Oparte na zasadzie rywalizacji:
 - VQ (ang. Vector Quantization):
 - Grossberga - Grossberg (1976);
 - Kohonena - Kohonen (1984).
 - SOM (ang. Self-Organizing Map) "Mapa Cech" - Kohonen (1995), Fausett (1994);
 - ART (ang. Adaptive Resonance Theory), sieci rezonansowe:
 - ART 1 - Carpenter and Grossberg (1987a), Moore (1988), Fausett (1994);
 - ART 2 - Carpenter i Grossberg (1987b), Fausett (1994);
 - ART 2-A - Carpenter, Grossberg i Rosen (1991a);
 - Rozmyty ART - Carpenter, Grossberg i Rosen (1991b).
- Autoasocjacyjne:
 - Liniowa sieć autoasocjacyjna - Anderson (1977), Fausett (1994);
 - Hopfielda - Hopfield (1982), Fausett (1994).

Klasyfikacja ze względu na funkcję przejścia:

- liniowe (Adaline)
- nieliniowe (perceptron wielowarstwowy, sieć z radialnymi funkcjami bazowymi)

Klasyfikacja ze względu na topologię:

- ze sprzężeniami (pamięć auto asocjacyjna Hopfielda)
- bez sprzężeń (perceptron wielowarstwowy, mapa cech Kohonena, sieć z radialnymi funkcjami bazowymi)

Modele neuronu

- W roku 1943 W.McCulloch i W. Pitts podali pierwszy model neuronu.
- Model ten można opisać zależnością:

- Gdzie $f(\cdot)$ to funkcja aktywacji.
$$y = f\left(\sum_{i=1}^N w_i u_i - \theta\right)$$
- Współczynniki w_i są wagami połączeń synaptycznych
- θ - to próg wrażliwości neuronu

Model neuronu

- W roku 1943 W. McCulloch i W. Pitts podali pierwszy model neuronu.
- Model ten można opisać zależnością:

Gdzie $f(\cdot)$ to funkcja aktywacji.

Współczynniki w_i są wagami połączeń synaptycznych

θ - to próg wrażliwości neuronu

$$y = f\left(\sum_{i=1}^N w_i u_i - \theta\right)$$

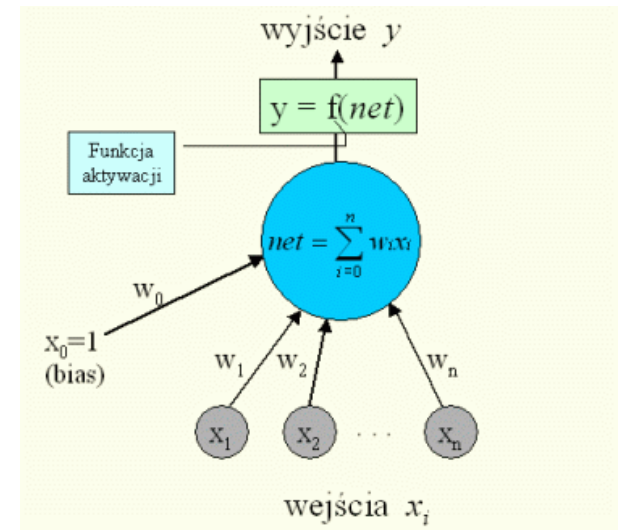
- Argument funkcji to potencjał aktywacji
- Oryginalnie była to funkcja:

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Wtedy mówimy, że neuron jest w stanie aktywnym gdy suma ważona sygnałów wejściowych

$$u_i \geq \theta$$

W zależności od znaku wagi w_i odpowiadające jej wejście jest typu pobudzającego lub hamującego. Ponieważ wyjście to 0 lub 1 to zasada „wszystko albo nic”.



Algorytm wstecznej propagacji błędów

Algorytm wstecznej propagacji błędów polega na takiej zmianie wag sygnałów wejściowych każdego neuronu w każdej warstwie, by wartość błędu dla kolejnych par uczących zawartych w zbiorze uczącym była jak najmniejsza. W tym celu wykorzystywana jest metoda gradientowa najszybszego spadku. Schemat krokowy można przedstawić następująco:

1. Inicjalizacja sieci i algorytmu
2. Obliczanie wartości wyjściowej sieci na podstawie danych
3. Obliczanie błędu sieci
4. Korekcja wag
5. Czy sieć nauczona?
 1. TAK – przejdź dalej
 2. NIE – wróć do punktu 2
6. Koniec

Przebieg algorytmu dla wszystkich elementów ciągu uczącego nazywa się **epoką**. Algorytm wstecznej propagacji błędów przedstawia sposób uczenia z nadzorem, lub inaczej- z nauczycielem. Określa on sposób, w jaki dobierane są współczynniki wagowe w sieciach wielowarstwowych na podstawie znanej wartości popełnionych błędów.

Schemat uczenia się



Backpropagation

- To jedna z najczęściej stosowanych technik uczenia się w sieciach neuronowych.



Jak wyznaczyć błąd predykcji sieci ?

$$Error_i = Output_i(1 - Output_i)(Actual_i - Output_i)$$

Gdzie:

- $Error_i$ jest błędem osiągniętym w węźle i-tym,
- $Output_i$ jest wartością przewidzianą przez sieć,
- $Actual_i$ jest wartością rzeczywistą (której sieć powinna się nauczyć).

$$Error_i = Output_i(1 - Output_i) \sum_{j=1}^n Error_j w_{ij}$$

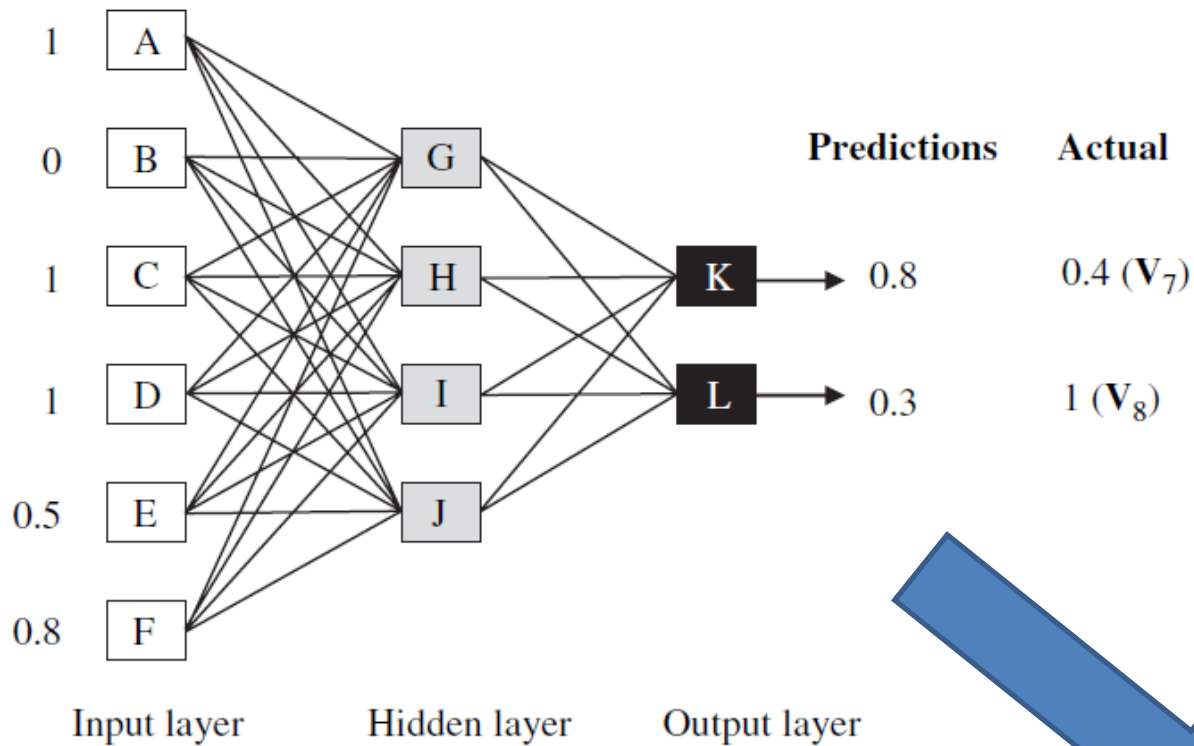
Zmiana wag

$$w_{ij} = w_{ij} + l \times Error_j \times Output_i$$

L- jest to tzw. **Współczynnik uczenia sieci**, najczęściej z przedziału [0,1]

Im mniejsza wartość tego współczynnika tym wolniejszy jest proces uczenia sieci neuronowej.

Często współczynnik ten jest ustawiany na najwyższą wartość początkowo, a następnie jest redukowany przy zmianie wag sieci.



Node K:

$$Error_K = Output_K(1 - Output_K)(Actual_K - Output_K)$$

$$Error_K = 0.8 \times (1 - 0.8) \times (0.4 - 0.8)$$

$$Error_K = -0.064$$

Node L:

$$Error_L = Output_L(1 - Output_L)(Actual_L - Output_L)$$

$$Error_L = 0.3 \times (1 - 0.3) \times (1 - 0.3)$$

$$Error_L = 0.147$$

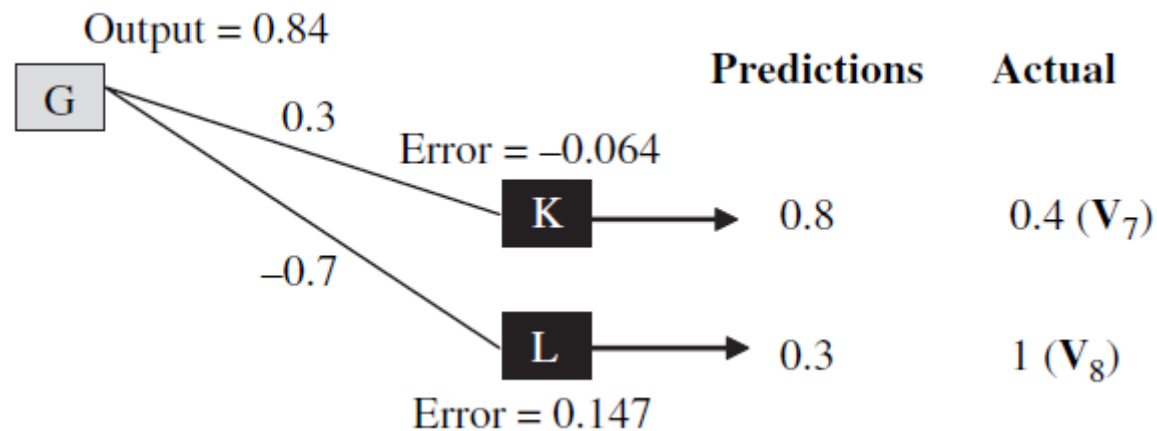
Przykład

Node G:

$$Error_G = Output_G(1 - Output_G)((Error_K \times w_{GK}) + (Error_L \times w_{GL}))$$

$$Error_G = 0.84 \times (1 - 0.84) \times ((-0.064 \times 0.3) + (0.147 \times -0.7))$$

$$Error_G = 0.0112$$



Zmiana wag

$$w_{ij} = w_{ij} + l \times Error_j \times Output_i$$

L- jest to tzw. **Współczynnik uczenia sieci**, najczęściej z przedziału [0,1]

Im mniejsza wartość tego współczynnika tym wolniejszy jest proces uczenia sieci neuronowej.

Często współczynnik ten jest ustawiany na najwyższą wartość początkowo, a następnie jest redukowany przy zmianie wag sieci.

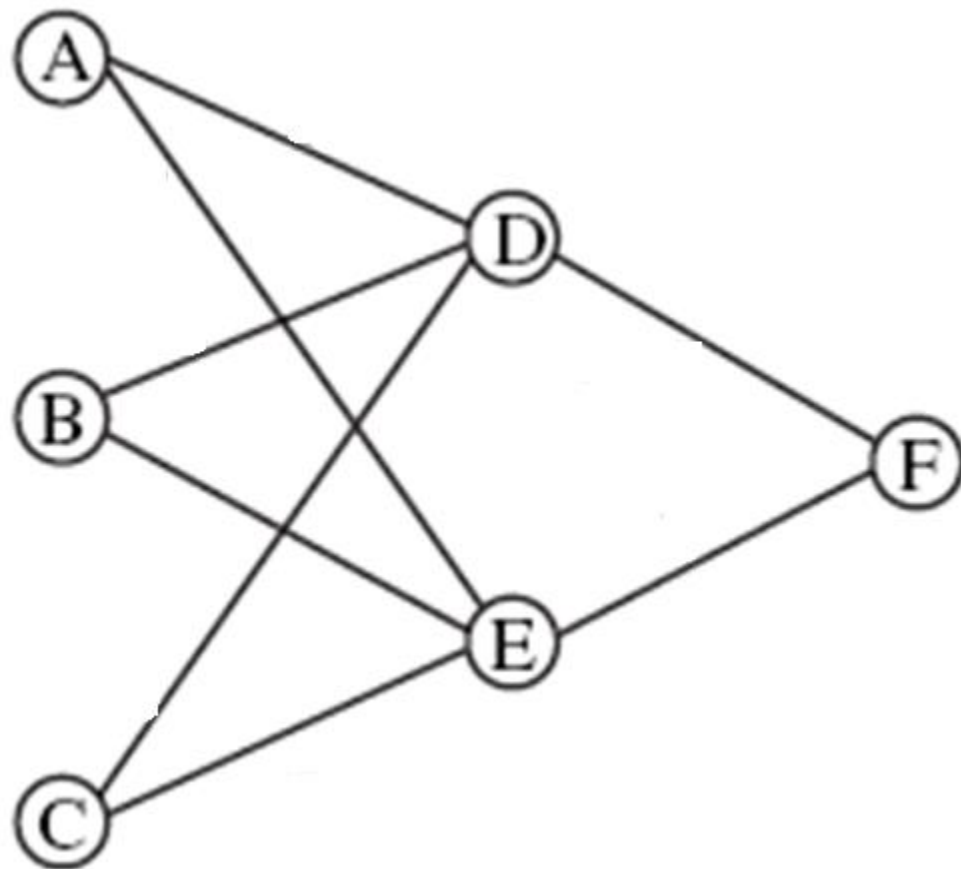
$$w_{GK} = w_{GK} + l \times Error_K \times Output_G$$

$$w_{GK} = 0.3 + 0.2 \times -0.064 \times 0.84$$

$$w_{GK} = 0.276$$

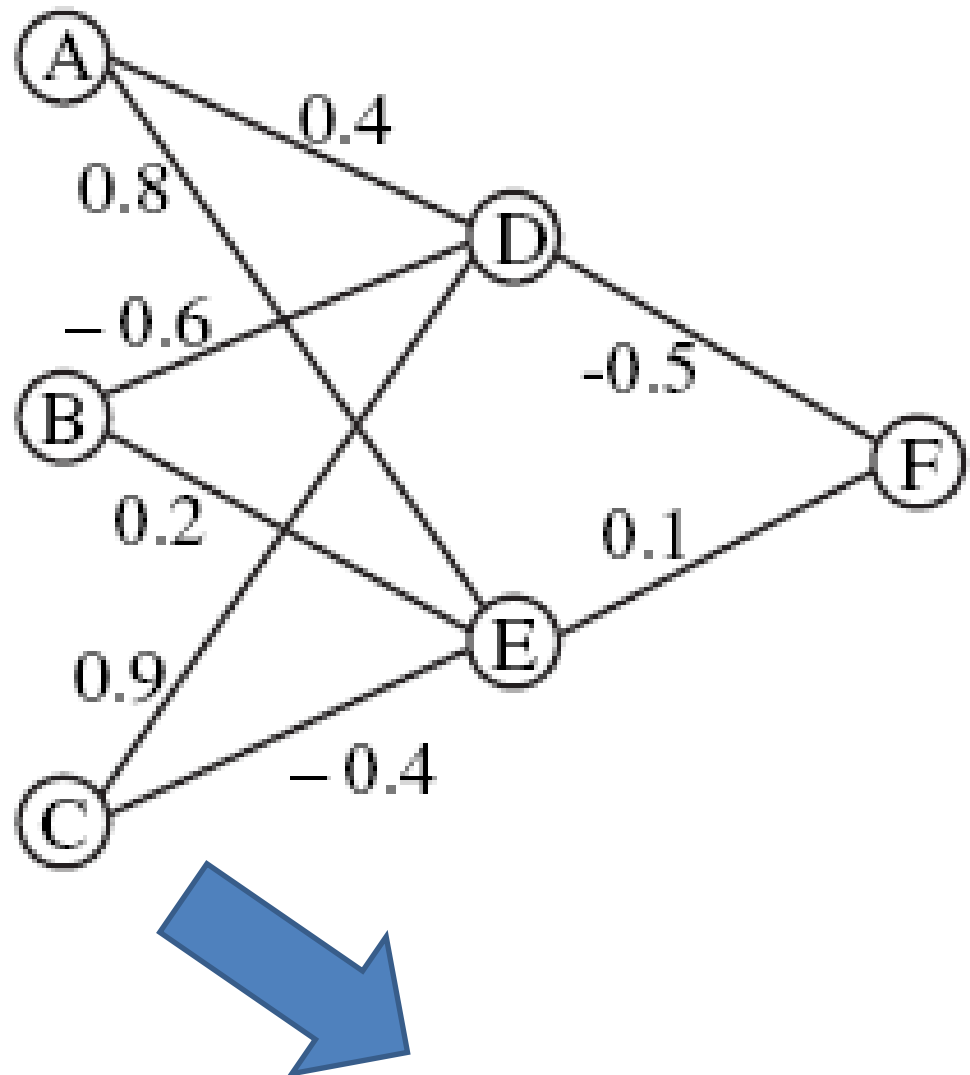
Przykład „krok po kroku”

- 1 warstwa ukryta: D, E
- Warstwa wejściowa: A, B, C
- Warstwa wyjściowa: F



Step 1

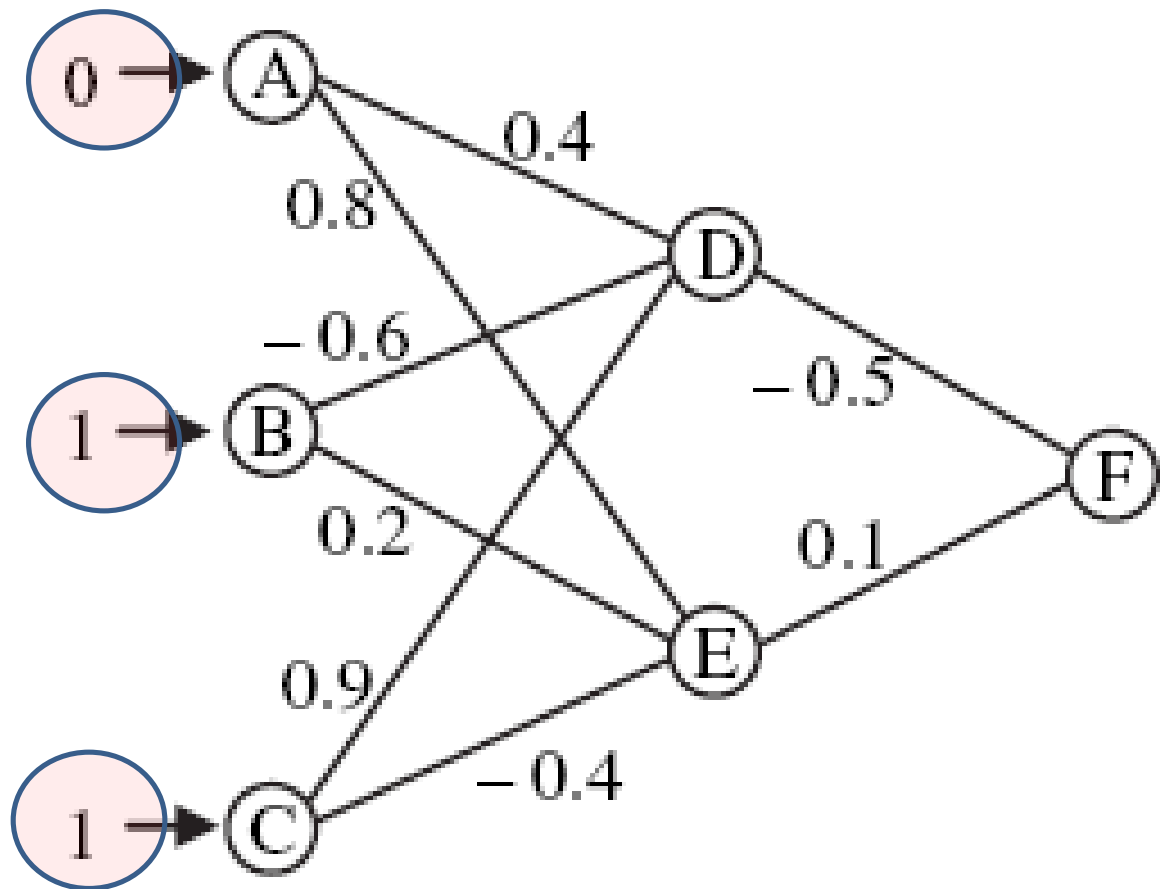
	I_1	I_2	I_3	O
i	1	1	1	1
ii	0	0	0	0
iii	1	0	0	1
iv	0	0	1	0
v	0	1	1	0



Wybieramy losowo jedną z obserwacji

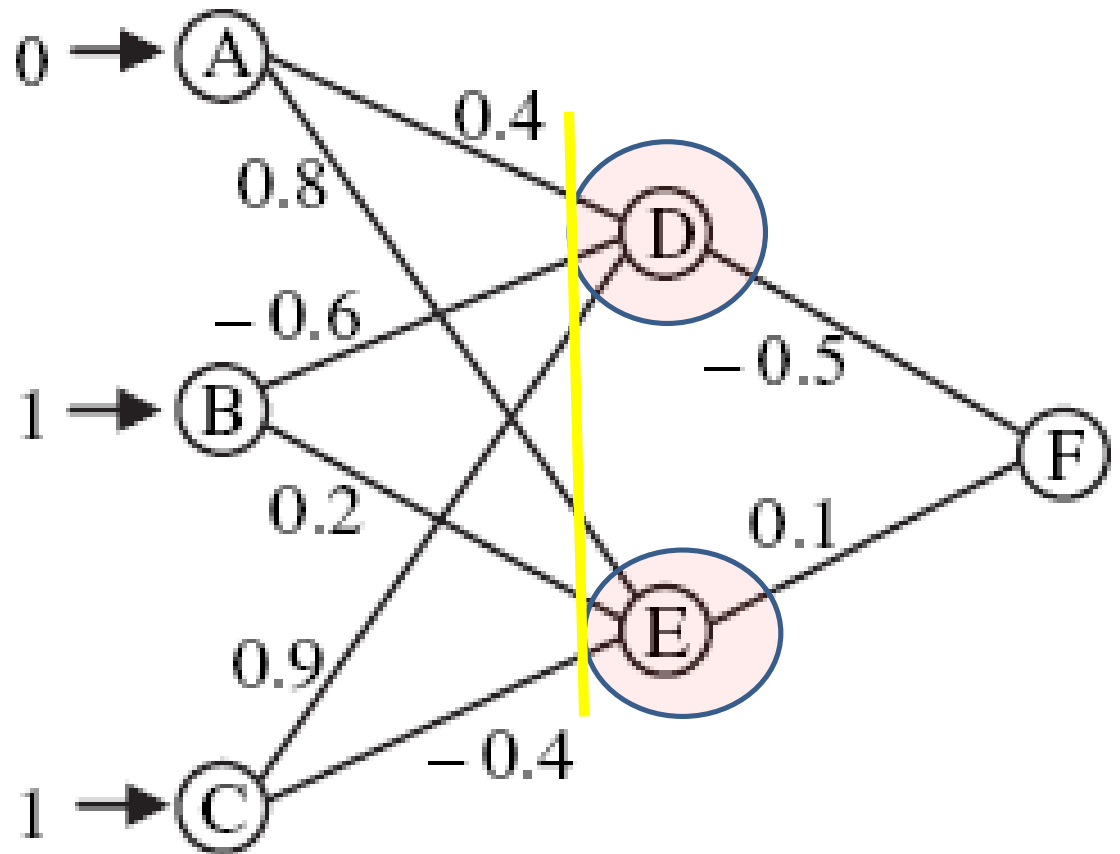
Step 2

	I₁	I₂	I₃	O
i	1	1	1	1
ii	0	0	0	0
iii	1	0	0	1
iv	0	0	1	0
v	0	1	1	0



Step 2

	I₁	I₂	I₃	O
i	1	1	1	1
ii	0	0	0	0
iii	1	0	0	1
iv	0	0	1	0
v	0	1	1	0



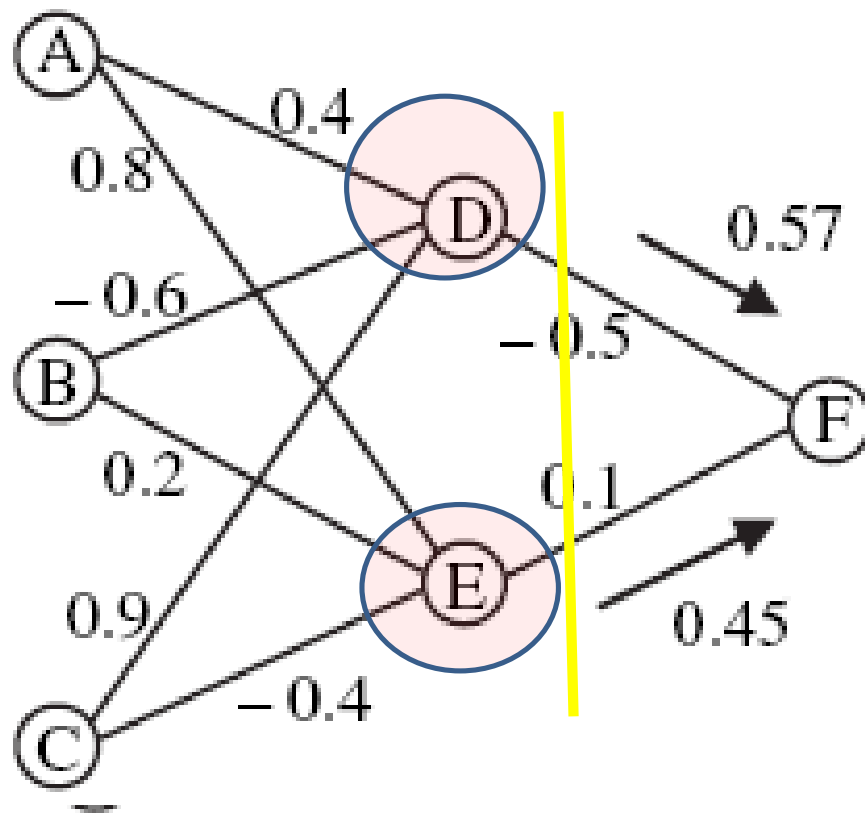
$$Input_D = I_1 \times w_{AD} + I_2 \times w_{BD} + I_3 \times w_{CD}$$

$$Input_D = (0 \times 0.4) + (1 \times -0.6) + (1 \times 0.9) = 0.3$$

$$Input_E = I_1 \times w_{AE} + I_2 \times w_{BE} + I_3 \times w_{CE}$$

$$Input_E = (0 \times 0.8) + (1 \times 0.2) + (1 \times -0.4) = -0.2$$

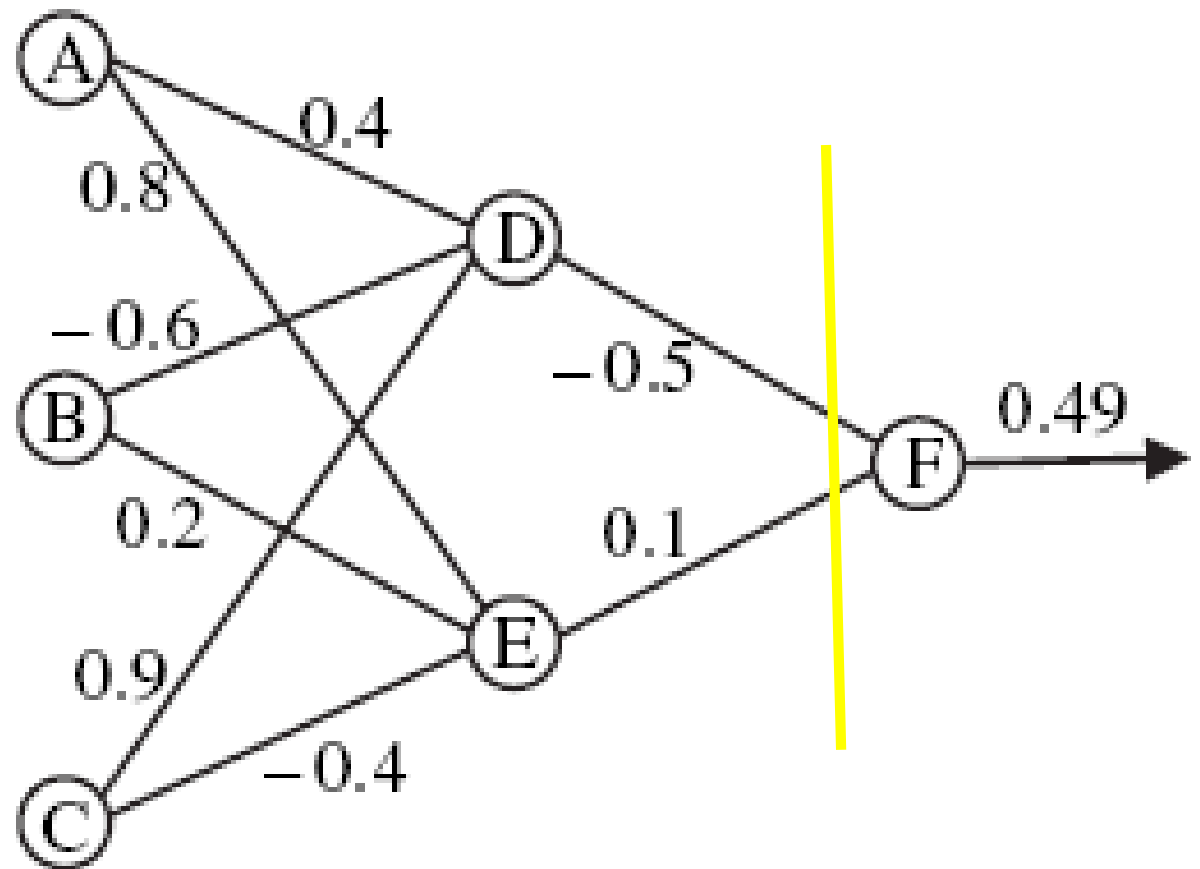
Step 3



$$Output_D = \frac{1}{1 + e^{-Input_D}} = \frac{1}{1 + e^{-0.3}} = 0.57$$

$$Output_E = \frac{1}{1 + e^{-Input_E}} = \frac{1}{1 + e^{0.2}} = 0.45$$

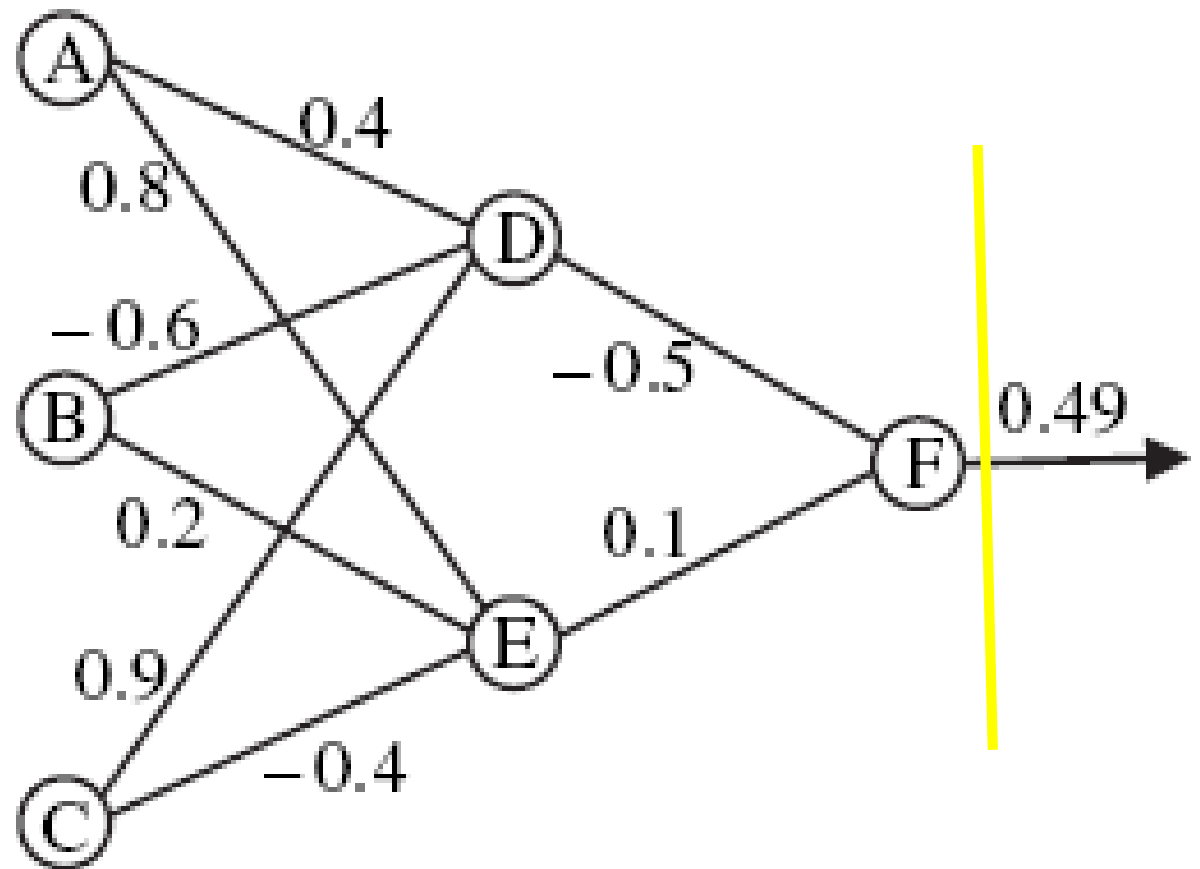
Step 4



$$Input_F = Output_D \times w_{DF} + Output_E \times w_{EF}$$

$$Input_F = (0.57 \times -0.5) + (0.45 \times 0.1) = -0.24$$

Step 4

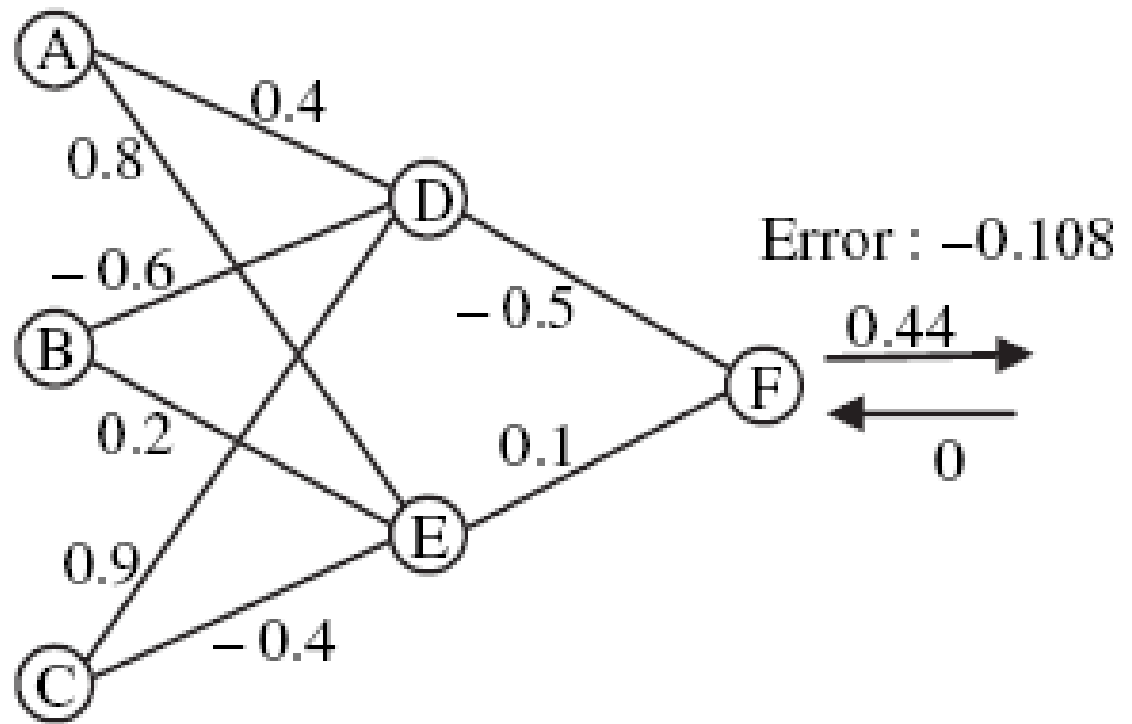


$$Input_F = Output_D \times w_{DF} + Output_E \times w_{EF}$$

$$Input_F = (0.57 \times -0.5) + (0.45 \times 0.1) = -0.24$$

$$Output_F = \frac{1}{1 + e^{-Input_F}} = \frac{1}{1 + e^{0.24}} = 0.44$$

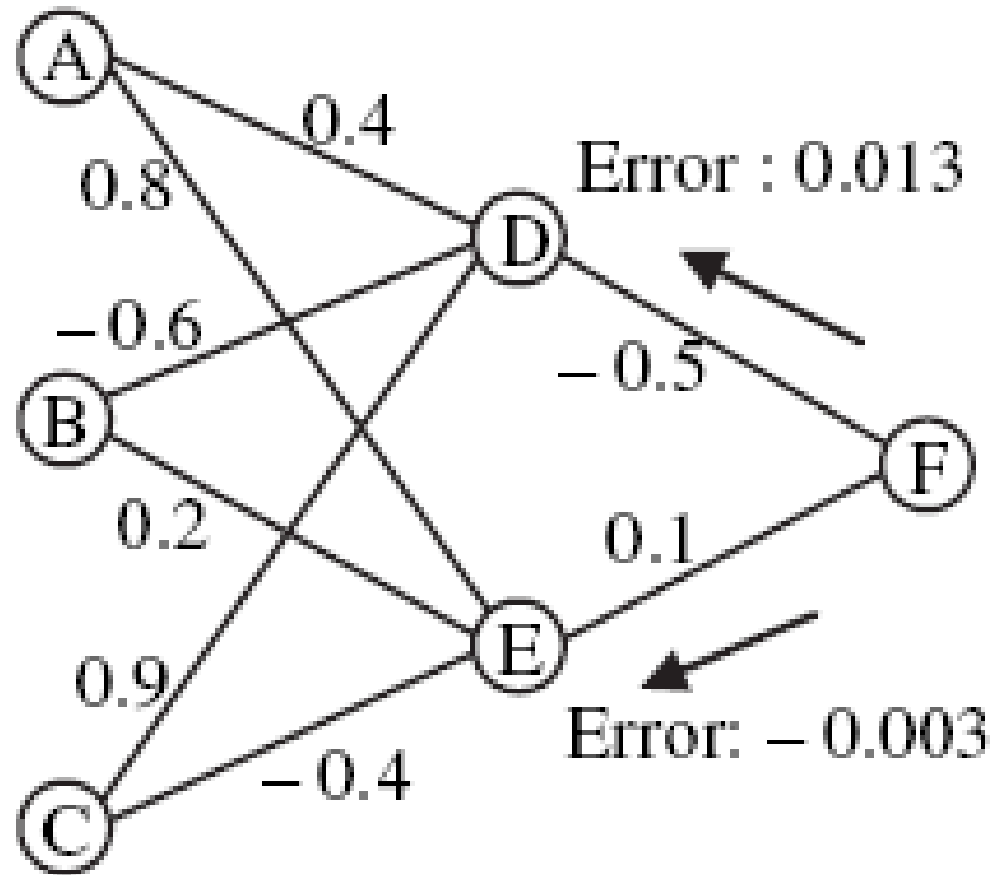
Step 5



$$Error_F = Output_F(1 - Output_F)(Actual_F - Output_F)$$

$$Error_F = 0.44(1 - 0.44)(0 - 0.44) = -0.108$$

Step 6



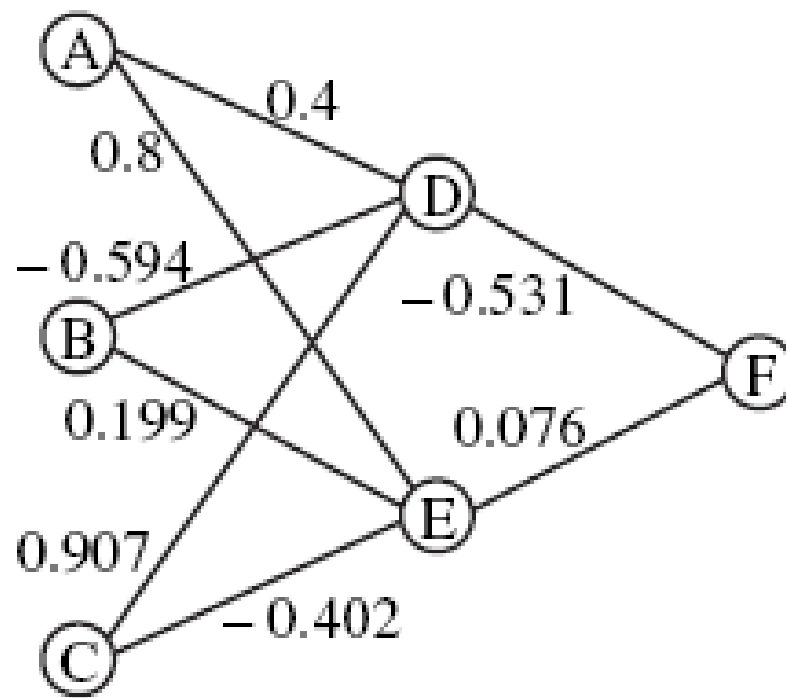
$$Error_D = Output_D(1 - Output_D)(Error_F \times w_{DF})$$

$$Error_D = 0.57(1 - 0.57)(-0.108 \times -0.5) = 0.013$$

$$Error_E = Output_E(1 - Output_E)(Error_F \times w_{EF})$$

$$Error_E = 0.45(1 - 0.45)(-0.108 \times 0.1) = -0.003$$

Step 7



$$w_{ij} = w_{ij} + l \times Error_j \times Output_i$$

$$w_{AD} = 0.4 + 0.5 \times 0.013 \times 0 = 0.4$$

$$w_{AE} = 0.8 + 0.5 \times -0.003 \times 0 = 0.8$$

$$w_{BD} = -0.6 + 0.5 \times 0.013 \times 1 = -0.594$$

$$w_{BE} = 0.2 + 0.5 \times -0.003 \times 1 = 0.199$$

$$w_{CD} = 0.9 + 0.5 \times 0.013 \times 1 = 0.907$$

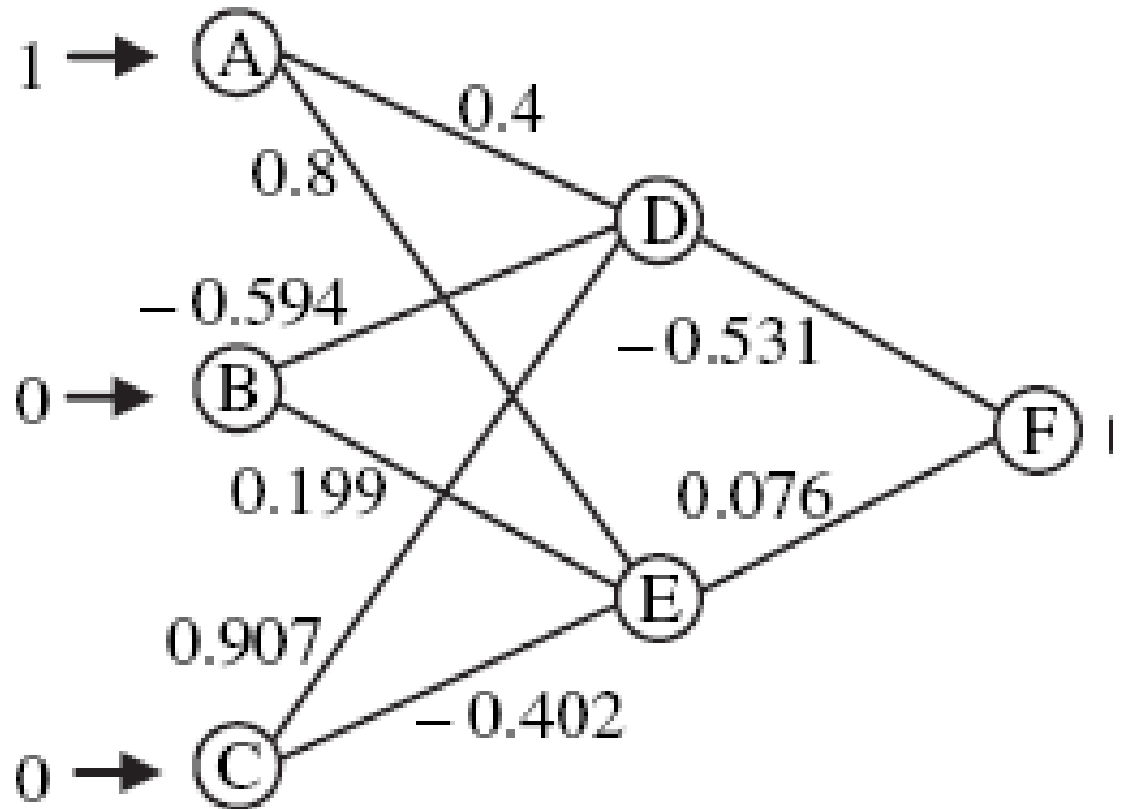
$$w_{CE} = -0.4 + 0.5 \times -0.003 \times 1 = -0.402$$

$$w_{DF} = -0.5 + 0.5 \times -0.108 \times 0.57 = -0.531$$

$$w_{EF} = 0.1 + 0.5 \times -0.108 \times 0.45 = 0.076$$

Step 8

	I_1	I_2	I_3	O
i	1	1	1	1
ii	0	0	0	0
iii	1	0	0	1
iv	0	0	1	0
v	0	1	1	0



It'd....

Rozmiar sieci neuronowej ?

- Za duże w praktyce uznaje się sieci mające kilka tysięcy neuronów bądź nawet więcej
- Ich liczba powinna zależeć od typu zadania które sieć ma do wykonania
- Siła sieci leży właśnie w liczbie neuronów, gęstości połączeń między nimi oraz odpowiednim dobraniu wag tych połączeń.

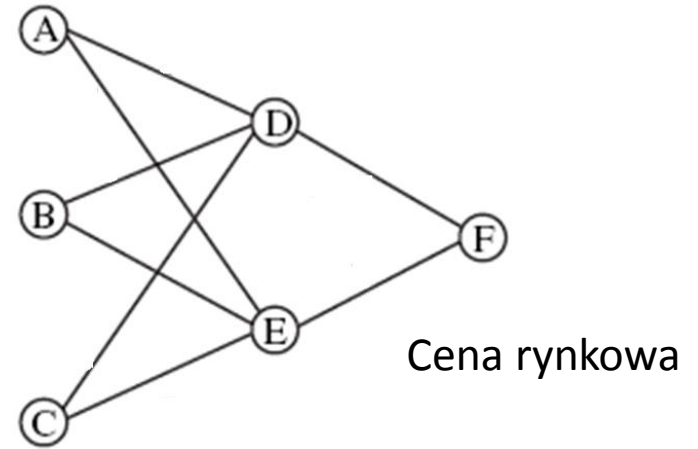
Ile warstw ukrytych ?

- **Liczba warstw ukrytych** jest zazwyczaj nie większa od dwóch. To w warstwach ukrytych następuje fuzja sygnałów sieci.
- **Warstwa wejściowa** zazwyczaj odpowiada jedynie za wstępne przygotowanie danych wejściowych.
- **Warstwa wyjściowa** odpowiada za finalną agregację pobudzeń neuronów z warstw ukrytych oraz prezentację końcowego wyniku działania sieci na wyjściach jej neuronów, stanowiących jednocześnie wyjścia całej sieci.

- **Liczba neuronów w warstwie wejściowej** z reguły zależy od liczby i typu atrybutów w zbiorze danych.
- **Liczba warstw ukrytych** oraz **liczba neuronów w każdej z tych warstw** zależą tylko od użytkownika (jego doświadczenia).
- **Warstwa wyjściowa** może składać się z więcej niż jednego neuronu, jej wielkość zależy od danego zadania klasyfikacji (typu problemu).
- Ile neuronów w ukrytej warstwie ? Większa liczba neuronów powoduje zwiększenie mocy obliczeniowej i elastyczności sieci neuronowej przy poznawaniu skomplikowanych wzorców, ale zbyt duża liczba prowadzi do przeuczenia, czyli zbyt dobrego dopasowania sieci do zbioru uczącego oraz utraty zdolności uogólniania na zbiorze testującym.
- Jeśli wystąpiło przeuczenie trzeba zmniejszyć liczbę neuronów.

- Zależności regresyjne

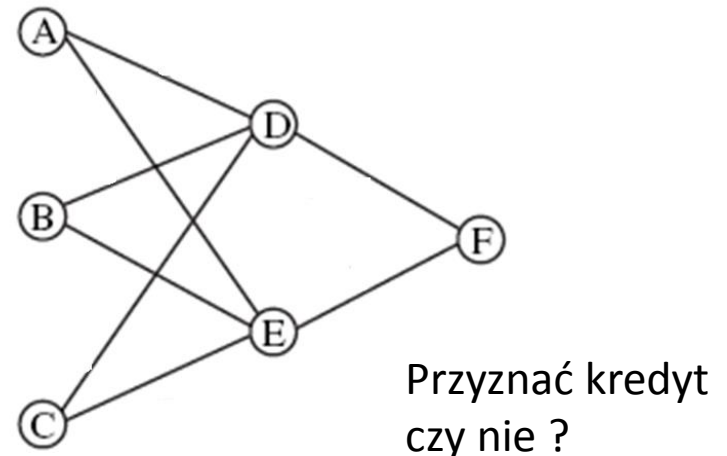
Powierzchnia
Garaż
Wiek
Ogrzewanie
położenie
Piętro
....



Na wyjściu będzie szacowana **wartość ceny rynkowej**

- Klasyfikacja wzorcowa

Dochody
Zabezpieczenie
Wiek
Stan cywilny
zatrudnienie
....



Na wyjściu będzie prawdopodobieństwo przynależności do klasy TAK/NIE

- Przypadek: analiza powikłań pooperacyjnych, 7 różnych klas. Można użyć modelu sieci, w której będą tylko dwie klasy: zgon i pozostałe. Wtedy sieć podaje tylko ryzyko zgonu.
- Zwykle w prostych problemach wystarcza jedna warstwa ukryta.
- Kryterium wyboru to zawsze warstwa wyjściowa a nie ukryta.
- Jakość dopasowania się nie zmienia gdy mamy funkcję typu: tangens, czy sigmoidalną.
- Wybieramy funkcję aktywacji liniową w warstwie wyjściowej. A w ukrytej – na jakość wpływa wybór funkcji aktywacji, więc spośród różnych funkcji aktywacji jest tangens hiperboliczny bo taka funkcja zawsze coś mówi: 0 – tylko w jednym przypadku, a poza tym zawsze jakaś wartość z przedziału $-1...+1$.

Gdy mamy zmienne jakościowe:

- Pochodzenie : {Azja, Europa, Ameryka}
- To dla tej jednej zmiennej jakościowej potrzebujemy aż trzech neuronów:
- Azja 1 0 0
- Europa 0 1 0
- Ameryka 0 0 1
- To rodzi wiele dodatkowych wag.
- Warto rozważyć scalanie kategorii – w mniejszą liczbę.
- Zwykle liczba wag powinna być 10 razy mniejsza niż liczba przypadków uczących.

Regresja dla zbioru Energia.sta

Zbiór zawiera dane o zapotrzebowaniu na energię, zebrane przez pewne Zakłady Energetyczne. Dane dotyczą okresu od listopada do stycznia następnego roku. Dla zachowania poufności wartości niektórych zmiennych zostały zmienione.											
	1 Miesiąc	2 Dzień miesiąca	3 Dzień tygodnia	4 Godzina	5 Zapotrzebowanie	6 Średnia temp. dobowa					
81	Listopad	4	Poniedziałek	9	353,718	-1,0					
82	Listopad	4	Poniedziałek	10	354,383	-1,0					
83	Listopad	4	Poniedziałek	11	364,066	-1,0					
84	Listopad	4	Poniedziałek	12	366,068	-1,0					
85	Listopad	4	Poniedziałek	13	369,205	-1,0					
86	Listopad	4	Poniedziałek	14	373,167	-1,0					
87	Listopad	4	Poniedziałek	15	366,521	-1,0					
88	Listopad	4	Poniedziałek	16	366,963	-1,0					
89	Listopad	4	Poniedziałek	17	392,660	-1,0					
90	Listopad	4	Poniedziałek	18	409,691	-1,0					
91	Listopad	4	Poniedziałek	19	413,232	-1,0					
92	Listopad	4	Poniedziałek	20	408,409	-1,0					
93	Listopad	4	Poniedziałek	21	396,544	-1,0					
94	Listopad	4	Poniedziałek	22	380,513	-1,0					
95	Listopad	4	Poniedziałek	23	358,139	-1,0					
96	Listopad	4	Poniedziałek	24	340,744	-1,0					
97	Listopad	5	Wtorek	1	322,712	-4,0					
98	Listopad	5	Wtorek	2	309,716	-4,0					
99	Listopad	5	Wtorek	3	303,944	-4,0					

STATISTICA - [Dane: Dane_energia_kurs.sta (6 zm. * 2208 prz.)]

Plik Edycja Wzrost Wstaw Format Statystyka Data Mining Wykresy Narzędzia Dane Okno Pomoc

Arial 10 B I U

Zbiór zawiera dane o zapotrzebowaniu. Dane dotyczą okresu od listopada do ... zmiennych zostały zmienione.

	1 Miesiąc	2 Dzień miesiąca	3 Dzień tygodnia
1	Listopad		1 Piątek
2	Listopad		1 Piątek
3	Listopad		1 Piątek
4	Listopad		1 Piątek
5	Listopad		1 Piątek
6	Listopad		1 Piątek
7	Listopad		1 Piątek
8	Listopad		1 Piątek
9	Listopad		1 Piątek
10	Listopad		1 Piątek
11	Listopad		1 Piątek
12	Listopad		1 Piątek
13	Listopad		1 Piątek
14	Listopad		1 Piątek
15	Listopad		1 Piątek
16	Listopad		1 Piątek

Przepisy Data Miner

- Ogólne modele drzew klasyfikacyjnych i regresyjnych
- Ogólne modele CHAID
- Drzewa interakcyjne (C&RT, CHAID)
- Wzmacniane drzewa klasyfikacyjne i regresyjne
- Łosowy las (regresja i klasyfikacja)
- Uogólnione modele addytywne
- MARSplines (Multivariate Adaptive Regression Splines)
- Analiza skupień uogólniona metoda EM i k-średnich
- Automatyczne sieci neuronowe**
- Inne metody uczenia maszyn
- Analiza składowych niezależnych
- Text & Document Mining
- Web Crawling, Document Retrieval
- Analiza koszykowa
- Analiza sekwencji, asocjacji i połączeń
- Szybkie wdrażanie modeli predykcyjnych (PMML)
- Dobroć dopasowania, klasyfikacji, predykcji
- Dobór i eliminacja zmiennych
- Łączenie klas(grup) dla predykcyjnego Data-Mining
- Przestrzenie robocze
- Optymalizacja procesu

Zbiór zawiera dane o zapotrzebowaniu na energię, zebrane przez pewne Zakłady Energetyczne. Dane dotyczą okresu od listopada do stycznia następnego roku. Dla zachowania poufności wartości niektórych zmiennych zostały zmienione.

	1 Miesiąc	2 Dzień miesiąca	3 Dzień tygodnia	4 Godzina	5 Zapotrzebowanie	6 Średnia temp. dobowa
1	Listopad	1				
2	Listopad	1				
3	Listopad	1				
4	Listopad	1				
5	Listopad	1				
6	Listopad	1				
7	Listopad	1				
8	Listopad	1				
9	Listopad	1				
10	Listopad	1				
11	Listopad	1				
12	Listopad	1				
13	Listopad	1				
14	Listopad	1				
15	Listopad	1				
16	Listopad	1				
17	Listopad	1				
18	Listopad	1	Piatek	18	285,577	5,0

[illegible]

4
Godzina

5
Zapotrzebowanie

6
Średnia temp.

1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21	293,734	5,0
22	287,581	5,0

SANN - Wybór danych: Dane_energia_kurs.sta

Podstawowe

Zmienne analizy

Wyjściowe ilościowe

Wyjściowe jakościowe

Wejściowe ilościowe

Wejściowe jakościowe

Sposób tworzenia

☒ Automatyczny

☐ Projekt sieci

☐ Wielokrotny

Wyjścia ilościowe:

6

Wejścia ilościowe:

4-5

Wejścia jakościowe:

1 3

☒ Pokazuj tylko zmienne o odpowiedniej skali

Proszę wybrać zmienne

2 - Dzień miesiąca

4 - Godzina

5 - Zapotrzebowanie

6 - Średnia temp. dobowe

2 - Dzień miesiąca

4 - Godzina

5 - Zapotrzebowanie

6 - Średnia temp. dobowe

1 - Miesiąc

3 - Dzień tygodnia

Rozwiń

Przybliź

Rozwiń

Przybliź

Rozwiń

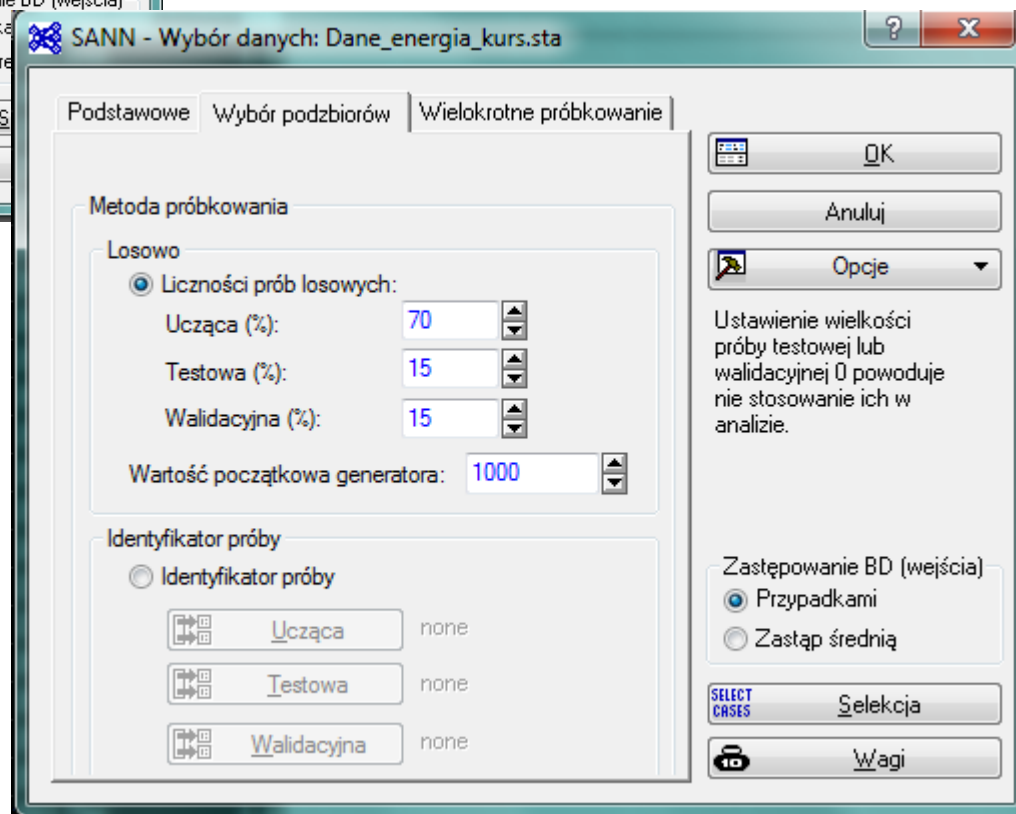
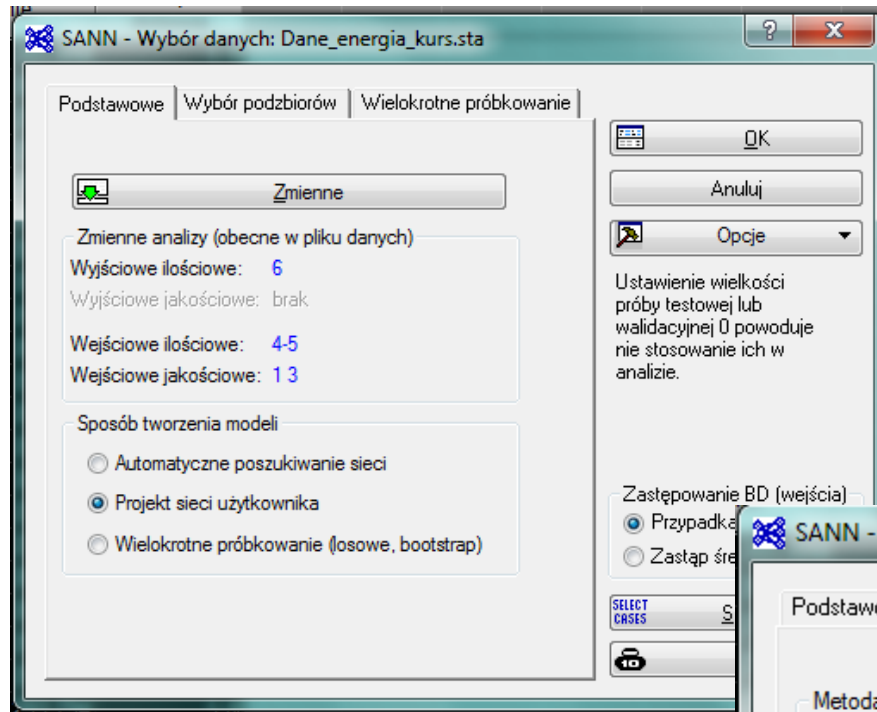
Przybliź

OK

Anuluj

[Zestawy]...

Włącz opcję "Pokazuj tylko zmienne o odpowiedniej skali" aby na listach, w zależności od potrzeby, pojawiały się tylko zmienne ilościowe albo jakościowe. Naciśnij F1 aby uzyskać więcej informacji.



SANN - Projekt użytkownika: Dane_energia_kurs.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd

Podstawowe | Perceptron | Redukcja wag | Inicjalizacja | Wykres uczenia | Uczenie

Typ sieci

☒ Perceptron wielowarstwowy

☐ Sieć RBF

Funkcja błędu

☒ Suma kwadratów

☐ Entropia wzajemna

Funkcja aktywacji

Neurony ukryte: Neurony wyjściowe:

Liczba sieci: Liczba neuronów:

SANN - Projekt użytkownika: Dane_energia_kurs.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd

Podstawowe | Perceptron | Redukcja wag | Inicjalizacja | Wykres uczenia | Uczenie

Algorytm uczenia

Algorytm:

Liczba epok:

Szybkość uczenia:

Bezwładność:

Inicjalizacja sieci

☒ Losowa, gaussowska

☐ Losowa, równomierna

Średnia/Min.:

Wariancja/Maks.:

Warunki zatrzymania

☒ Włącz warunki zatrzymania

Zmiana błędu: Okno:

Uczenie

Idź do wyników

Zapisz sieci

Statystyki danych

Podsumowanie

Anuluj

Opcje

Klikając „uczenie”

SANN - Wyniki: Dane_energia_kurs.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd
1	MLP 12-8-1	0,636447	0,598197	0,606783	BFGS 20	SOS

Wybór sieci Usuwanie sieci

Więcej (projekt użytkownika) Więcej (projekt automatyczny) Więcej (wielokrotne próbkowanie)

Przewidywania | Wykresy | Szczegóły | Przewidywania dla nowych danych

Arkusz predykcji

Sposób predykcji

☒ Pojedyncze sieci

☐ Zespół

☐ Pojedyncze sieci i zespół

Umieść w arkuszu

☐ Wejścia ☐ Moduły reszt

☒ Zmn. zal. ☐ Kwadraty reszt

☒ Wyjścia ☐ Przedziały aktywacji

☐ Reszty ☐ Inne zmienne

☐ Standaryzowane reszty

Predykcja

Podsumowanie

Zapisz sieci

Anuluj

Opcje

Próby

☒ Uczenie

☐ Test

☐ Walidacja

☐ Brakujące

STATISTICA - [Skoroszyt1* - Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)]

Plik Edycja Wzrost Wstaw Format Statystyki Data Mining Wykresy Narzędzia Dane Skoroszyt Okno Pomoc

Arial 10 B I U [Formatowanie] Dodaj do skoroszytu Dodaj do raportu Dodaj do MS Word [Wstawianie] Zmienne Przypadki [Wykresy]

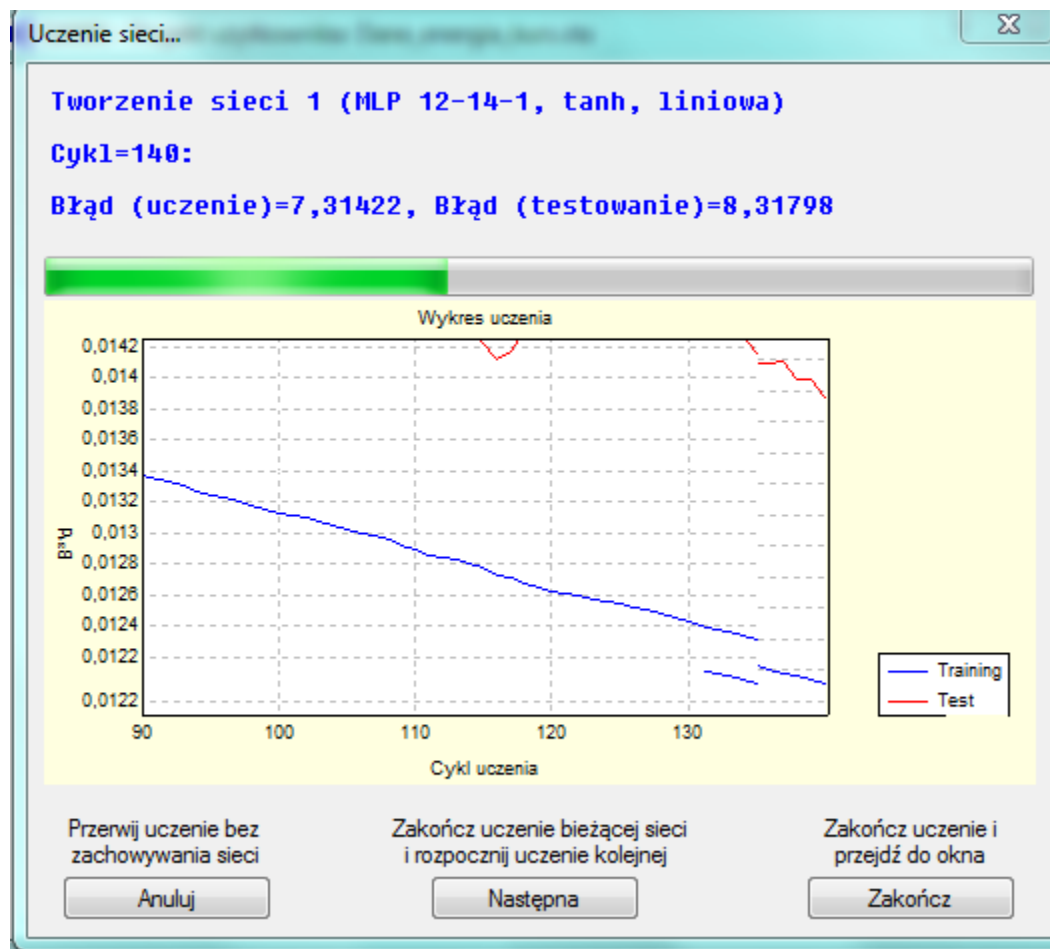
Skoroszyt1*

SANN (Dane_energia_kurs.sta)

Okno wyboru wariantów

Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)

Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Błąd (uczenie)	Błąd (testowanie)	Błąd (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
1	MLP 12-8-1	0,636447	0,598197	0,606783	9,931587	9,617678	10,77772	BFGS 20	SOS	Tanh	Liniowa



Zwiększyć liczbę neuronów z 8 na 14
Zwiększyć liczbę epok z 20 na 300

Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)											
Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Błąd (uczenie)	Błąd (testowanie)	Błąd (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
2	MLP 12-14-1	0,753996	0,681856	0,706956	7,201541	8,107286	8,542372	BFGS 146	SOS	Tanh	Liniowa

porównanie

STATISTICA - [Skoroszyt1* - Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)]

Plik Edycja Wzrost Wstaw Format Statystyka Data Mining Wykresy Narzędzia Dane Skoroszyt Okno Pomoc

Dodaj do skoroszytu Dodaj do raportu Dodaj do MS Word

Arial 10 B I U

Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)

Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Błąd (uczenie)	Błąd (testowanie)	Błąd (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
1	MLP 12-8-1	0,636447	0,598197	0,606783	9,931587	9,617678	10,77772	BFGS 20	SOS	Tanh	Liniowa

Zwiększono liczbę neuronów z 8 na 14

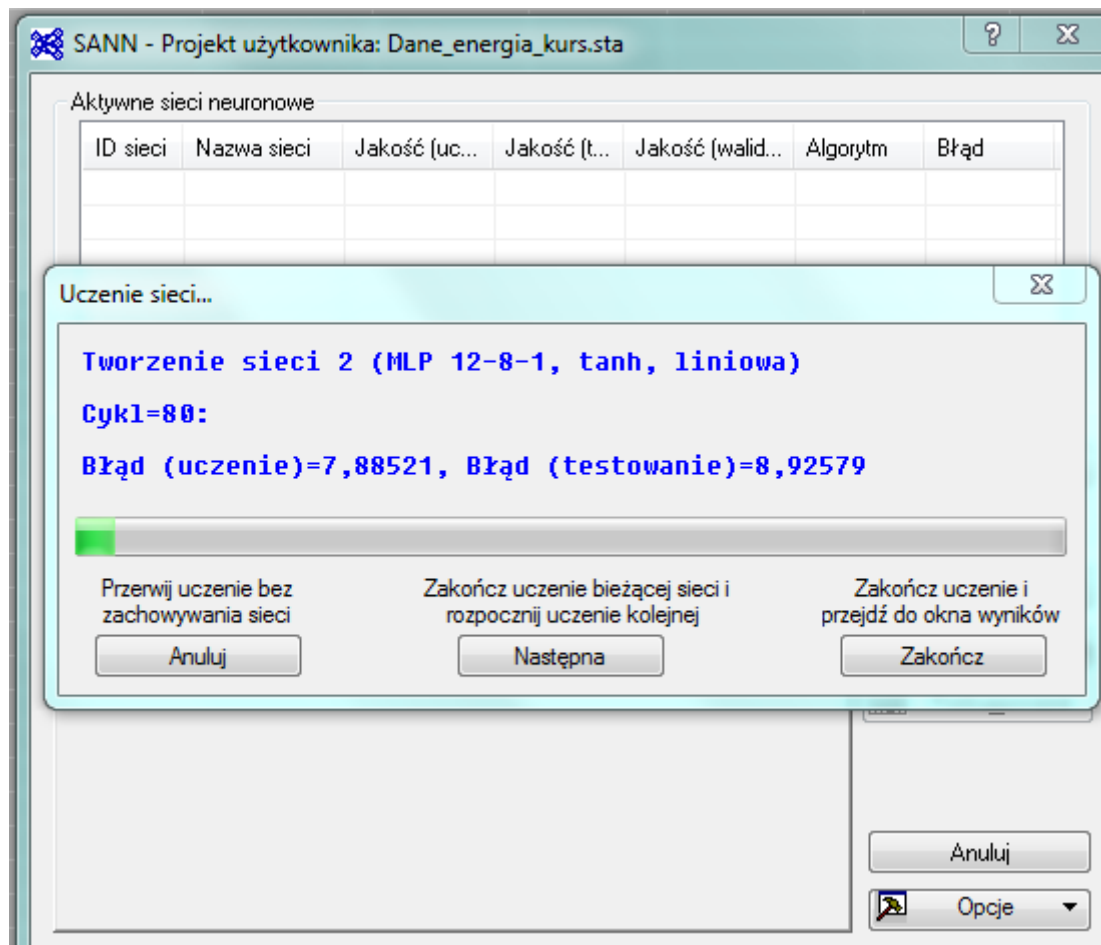
Zwiększono liczbę epok z 20 na 300

10 B I U

Podsumowanie aktywnych sieci (Dane_energia_kurs.sta)

Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Błąd (uczenie)	Błąd (testowanie)	Błąd (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
2	MLP 12-14-1	0,753996	0,681856	0,706956	7,201541	8,107286	8,542372	BFGS 146	SOS	Tanh	Liniowa

4 sieci, po 8 neuronów



Więcej sieci – automatycznie generowane

SANN - Automatyczny projekt sieci: Dane_energia_kurs.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd
1	MLP 12-8-1	0,725314	0,643154	0,655931	BFGS 107	SOS
2	MLP 12-8-1	0,746849	0,671905	0,679137	BFGS 174	SOS
3	MLP 12-8-1	0,733532	0,674084	0,682096	BFGS 114	SOS
4	MLP 12-8-1	0,707696	0,643881	0,655788	BFGS 62	SOS

Podstawowe | Funkcje aktywacji sieci MLP | Redukcja wag | Inicjalizacja

Typy sieci

☒ MLP:

Min. liczba ukrytych: 3

Maks. liczba ukrytych: 10

☐ RBF:

Min. liczba ukrytych: 21

Maks. liczba ukrytych: 30

Uczenie i zachowywanie

Liczba sieci: 5

Zachowywanych: 5

Funkcja błędu

☒ Suma kwadratów

☐ Entropia wzajemna

Uczenie

Idź do wyników

Zapisz sieci

Statystyki danych

Uczenie sieci...

Tworzenie sieci 2 (MLP 12-9-1, logisticzna, exp)

Cykl=115:

Błąd (uczenie)=7,63624, Błąd (testowanie)=8,85819

Przerwij uczenie bez zachowywania sieci

Zakończ uczenie bieżącej sieci i rozpocznij uczenie kolejnej

Zakończ uczenie i przejdź do okna wyników

Anuluj

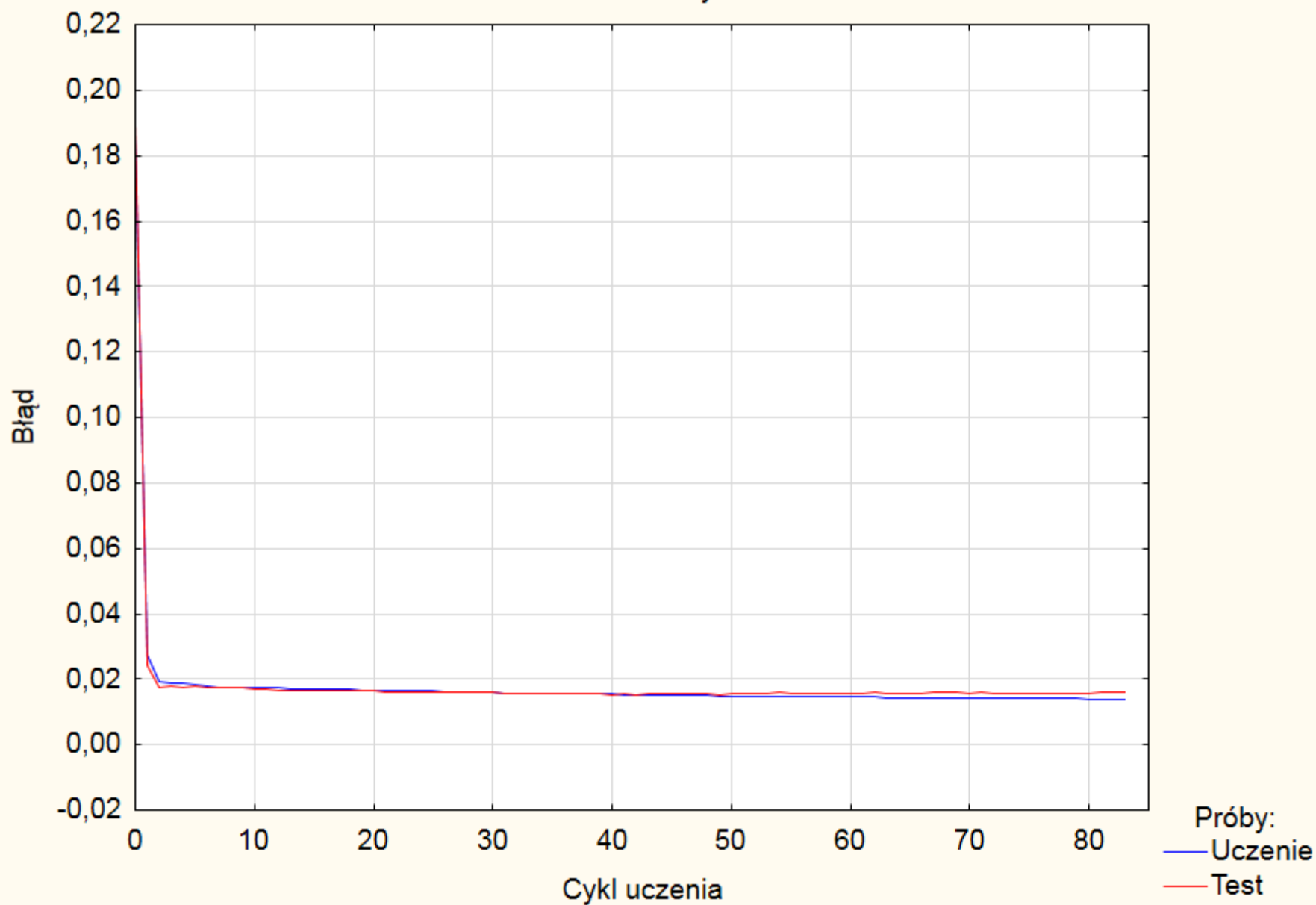
Następna

Zakończ

predykcja


Przypadek nazwa	Arkusz predykcji dla Średnia temp. dobowa (Dane_energia_kurs.sta) Próby: Uczenie				
	Średnia temp. dobowa Zm.zal	Średnia temp. dobowa - Wyjście 2. MLP 12-8-1	Średnia temp. dobowa - Wyjście 4. MLP 12-8-1	Średnia temp. dobowa - Wyjście 6. MLP 12-9-1	
1	5,0000	5,1916	5,7293	5,5648	
2	5,0000	5,1650	6,0081	5,6070	
3	5,0000	5,1210	6,1263	5,6051	
4	5,0000	5,0315	6,1117	5,6043	
6	5,0000	4,9210	6,2400	5,5949	
8	5,0000	4,8204	6,3588	5,5802	
10	5,0000	4,6163	6,2194	5,5856	
11	5,0000	4,5733	6,2669	5,5780	
12	5,0000	4,5452	6,2789	5,5731	
14	5,0000	4,8720	6,3240	5,5594	
15	5,0000	5,2022	6,4014	5,5406	
16	5,0000	5,8390	6,4427	5,5228	
17	5,0000	6,6588	6,1963	5,5682	
19	5,0000	6,3680	5,9030	5,5949	
23	5,0000	6,1140	6,1962	5,5719	
25	5,0000	4,7367	4,9320	5,6240	
26	5,0000	4,8136	4,9594	5,6250	
27	5,0000	5,0320	4,9741	5,6253	
28	5,0000	5,5479	4,9831	5,6253	
30	5,0000	6,8343	5,0659	5,6248	
31	5,0000	6,7672	5,1008	5,6247	
33	5,0000	6,4176	5,1957	5,6206	
34	5,0000	6,2591	5,2218	5,6146	
35	5,0000	6,1464	5,2571	5,6126	
37	5,0000	5,9270	5,3301	5,6091	
39	5,0000	5,8201	5,4258	5,6213	
40	5,0000	5,7333	5,4633	5,6234	

Wykres uczenia dla [13.MLP 12-8-1]
Sieć odnaleziono w cyklu uczenia 49



Klasyfikacja wzorcowa

	1	2	3	4	5	6	7	8	9	10	11	
	Pracujący	Przeznaczenie	Plec	Stan cywilny	Zagrożenie	Wiek	Stan konta	Rata	Liczba rat	Staż	Decyzja	
1	Tak	komputer	K	samotny	nie	18	200	20	15	1	tak	
2	Tak	komputer	K	samotny	nie	20	100	20	20	2	tak	
3	Nie	komputer	K	rodzina	tak	25	50	40	12	0	nie	
4	Tak	komputer	K	rodzina	nie	40	50	70	12	2	tak	
5	Tak	komputer	K	samotny	tak	50	50	40	12	25	tak	
6	Tak	komputer	M	samotny	nie	18	100	50	8	1	tak	
7	Tak	komputer	M	samotny	nie	22	100	30	8	4	tak	
8	Tak	komputer	M	rodzina	nie	28	150	40	10	5	tak	
9	Tak	komputer	M	rodzina	nie	40	200	20	20	15	tak	
10	Nie	komputer	M	rodzina	nie	50	50	40	12	0	nie	
11	Tak	samochod	K	samotny	nie	18	500	80	20	1	nie	
12	Nie	samochod	K	rodzina	nie	20	500	100	20	2	nie	
13	Tak	samochod	K	samotny	nie	25	500	50	20	5	nie	
14	Tak	samochod	K	samotny	nie	38	1500	100	20	15	tak	
15	Tak	samochod	K	rodzina	nie	50	500	150	20		tak	
16	Tak	samochod	M	samotny	nie	19	500	70	20	2	nie	
17	Tak	samochod	M	rodzina	nie	21	1500	30	20	3	tak	
18	Tak	samochod	M	samotny	nie	25	1500	100	20	2	tak	
19	Tak	samochod	M	rodzina	nie	38	1000	100	20	15	tak	
20	Tak	samochod	M	rodzina	nie	50	500	100	30	2	nie	
21	Tak	audio	M	rodzina	nie	42	900	20	6	7	tak	
22	Tak	audio	M	samotny	nie	28	100	100	5	3	tak	
23	Nie	audio	K	rodzina	tak	55	300	100	8	0	nie	
24	Tak	audio	K	rodzina	nie	21	100	10	12	1	tak	
25	Tak	audio	K	samotny	nie	81	200	40	10	0	nie	
26	Tak	audio	K	samotny	nie	23	800	100	6	1	nie	
27	Tak	audio	M	rodzina	nie	35	70	100	4	5	nie	
28	Tak	audio	K	samotny	nie	47	50	30	7	8	tak	
29	Tak	audio	M	rodzina	nie	98	100	150	5	0	tak	

 SANN - Projekt użytkownika: Kredyty1.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd

Podstawowe

Perceptron

Redukcja wag

Inicjalizacja

Wykres uczenia

Typ sieci

☒ Perceptron wielowarstwowy

☐ Sieć RBF

Funkcja błędu

☐ Suma kwadratów

☒ Entropia wzajemna

Funkcja aktywacji

Neurony ukryte:

Tanh

Neurony wyjściowe:

Softmax

Liczba sieci:

1

Liczba neuronów:

8

Uczenie

Idź do wyników

Zapisz sieci

Statystyki danych

Podsumowanie

Anuluj

Opcje

SANN - Wyniki: Kredyty1.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd
1	MLP 20-8-2	68,181818	77,777778	77,777778	BFGS 2	CE

Wybór sieci Usuwanie sieci

Więcej (projekt użytkownika) Więcej (projekt automatyczny) Więcej (wielokrotne próbkowanie)

Wykresy przyrostu i ROC Przewidywania dla nowych danych

Przewidywania Wykresy Szczegóły

Arkusz predykcji

Sposób predykcji

☒ Pojedyncze sieci

☐ Zespół

☐ Pojedyncze sieci i zespół

Umieść w arkuszu

☐ Wejścia ☐ Moduły reszt

☒ Zmn. zal. ☐ Kwadraty reszt

☒ Wyjścia ☐ Poziomy aktywacji

☐ Trafność ☐ Inne zmienne

☐ Standaryzowane reszty

Predykcja

Podsumowanie

Zapisz sieci

Anuluj

Opcje

Próby

☒ Uczenie

☐ Test

☐ Walidacja

☐ Brakujące

roszyt1* - Arkusz predykcji dla Decyzja (Kredyty1.sta)]

Widok Wstaw Format Statystyka Data Mining Wykresy Narzędz

Dodaj do skoroszytu Dodaj

10 B I U

energi
y1.sta
oru v
pred

Arkusz predykcji dla Decyzja (Kredyty1.sta)			
Próby: Uczenie			
Przypadek nazwa	Decyzja Zm.zal	Decyzja - Wyjście 1. MLP 20-8-2	
1	tak	tak	
2	tak	tak	
4	tak	tak	
6	tak	tak	
8	tak	tak	
9	tak	tak	
11	nie	tak	
12	nie	tak	
13	nie	tak	
14	tak	tak	
16	nie	tak	
17	tak	tak	
19	tak	tak	
20	nie	tak	
22	tak	tak	
24	tak	tak	
25	nie	tak	
26	nie	tak	
27	nie	tak	

STATISTICA - [Skoroszyt1* - Arkusz predykcji dla Decyzja (Kredyty1.sta)]

Plik Edycja Wzrost Wstaw Format Statystyka Data Mining Wykresy Narzędzia Dane Skoroszyt Okno Pomoc

Dodaj do skoroszytu Dodaj do raportu Dodaj do MS Word

Arial 10 B I U

Skoroszyt1*

SANN (Dane_energ

SANN (Kredyty1.sta

Okno wyboru v

Arkusz pred

Arkusz pred


Arkusz predykcji dla Decyzja (Kredyty1.sta)													
Próby: Uczenie													
Przypadek nazwa	Wiek Wejście	Stan konta Wejście	Rata Wejście	Liczba rat Wejście	Staz Wejście	Pracujący Wejście	Przeznaczenie Wejście	Plec Wejście	Stan cywilny Wejście	Zagrożenie Wejście	Decyzja Zm.zal	Decyzja - Wyjście 1. MLP 20-8-2	Decyzja - Trafność 1. MLP 20-8-2
1	18,00000	200,000	20,0000	15,00000	1,00000	Tak	komputer	K	samotny	nie	tak	tak	Poprawne
2	20,00000	100,000	20,0000	20,00000	2,00000	Tak	komputer	K	samotny	nie	tak	tak	Poprawne
4	40,00000	50,000	70,0000	12,00000	2,00000	Tak	komputer	K	rodzina	nie	tak	tak	Poprawne
6	18,00000	100,000	50,0000	8,00000	1,00000	Tak	komputer	M	samotny	nie	tak	tak	Poprawne
8	28,00000	150,000	40,0000	10,00000	5,00000	Tak	komputer	M	rodzina	nie	tak	tak	Poprawne
9	40,00000	200,000	20,0000	20,00000	15,00000	Tak	komputer	M	rodzina	nie	tak	tak	Poprawne
11	18,00000	500,000	80,0000	20,00000	1,00000	Tak	samochod	K	samotny	nie	nie	tak	Niepoprawne
12	20,00000	500,000	100,0000	20,00000	2,00000	Nie	samochod	K	rodzina	nie	nie	tak	Niepoprawne
13	25,00000	500,000	50,0000	20,00000	5,00000	Tak	samochod	K	samotny	nie	nie	tak	Niepoprawne
14	38,00000	1500,000	100,0000	20,00000	15,00000	Tak	samochod	K	samotny	nie	tak	tak	Poprawne
16	19,00000	500,000	70,0000	20,00000	2,00000	Tak	samochod	M	samotny	nie	nie	tak	Niepoprawne
17	21,00000	1500,000	30,0000	20,00000	3,00000	Tak	samochod	M	rodzina	nie	tak	tak	Poprawne
19	38,00000	1000,000	100,0000	20,00000	15,00000	Tak	samochod	M	rodzina	nie	tak	tak	Poprawne
20	50,00000	500,000	100,0000	30,00000	2,00000	Tak	samochod	M	rodzina	nie	nie	tak	Niepoprawne
22	28,00000	100,000	100,0000	5,00000	3,00000	Tak	audio	M	samotny	nie	tak	tak	Poprawne
24	21,00000	100,000	10,0000	12,00000	1,00000	Tak	103 jo audio	K	rodzina	nie	tak	tak	Poprawne
25	81,00000	200,000	40,0000	10,00000	0,00000	Tak	audio	K	samotny	nie	nie	tak	Niepoprawne
26	23,00000	800,000	100,0000	6,00000	1,00000	Tak	audio	K	samotny	nie	nie	tak	Niepoprawne
27	35,00000	70,000	100,0000	4,00000	5,00000	Tak	audio	M	rodzina	nie	nie	tak	Niepoprawne
28	47,00000	50,000	30,0000	7,00000	8,00000	Tak	audio	K	samotny	nie	tak	tak	Poprawne
29	98,00000	100,000	150,0000	5,00000	0,00000	Tak	audio	M	rodzina	nie	tak	tak	Poprawne
30	68,00000	20,000	70,0000	8,00000	0,00000	Tak	audio	K	samotny	tak	nie	tak	Niepoprawne
31	27,00000	60,000	100,0000	6,00000	4,00000	Tak	audio	K	samotny	nie	tak	tak	Poprawne
32	19,00000	50,000	20,0000	20,00000	1,00000	Tak	audio	K	samotny	nie	tak	tak	Poprawne
35	31,00000	100,000	20,0000	10,00000	9,00000	Tak	audio	K	rodzina	nie	tak	tak	Poprawne
37	20,00000	100,000	20,0000	20,00000	2,00000	Tak	audio	M	samotny	nie	nie	tak	Niepoprawne
38	32,00000	50,000	40,0000	10,00000	14,00000	Tak	audio	M	rodzina	nie	tak	tak	Poprawne

Macierz pomyłek

Decyzja (Podsumowanie klasyfikacji) (Kredyty1.sta)					
Próby: Uczenie					
		Decyzja-nie	Decyzja-tak	Decyzja-Wszystkie	
1.MLP 20-8-2	Razem	29,00000	59,0000	88,00000	
	Poprawne	1,00000	59,0000	60,00000	
	Niepoprawne	28,00000	0,0000	28,00000	
	Poprawne (%)	3,44828	100,0000	68,18182	
	Niepoprawne (%)	96,55172	0,0000	31,81818	



Podsumowanie aktywnych sieci (Kredyty1.sta)								
Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
1	MLP 20-8-2	68,18182	77,77778	77,77778	BFGS 2	Entropia	Tanh	Softmax

Funkcja SSE

 SANN - Wyniki: Kredyty1.sta

Aktywne sieci neuronowe

ID sieci	Nazwa sieci	Jakość (uc...	Jakość (t...	Jakość (walid...	Algorytm	Błąd
3	MLP 20-8-2	86,363636	77,777778	77,777778	BFGS 9	SOS

 Wybór sieci  Usuwanie sieci

[Więcej \(projekt użytkownika\)](#) [Więcej \(projekt automatyczny\)](#) [Więcej \(wielokrotne próbkowanie\)](#)

[Wykresy przyrostu i ROC](#) [Przewidywania dla nowych danych](#)

[Przewidywania](#) [Wykresy](#) [Szczegóły](#)


Arkusz predykcji

Sposób predykcji

☒ Pojedyncze sieci

☐ Zespół

☐ Pojedyncze sieci i zespół

 Predykcja

Umieść w arkuszu


☒ Wejścia ☐ Moduły reszt


☒ Zmn. zal. ☐ Kwadraty reszt

☒ Wyjścia ☐ Poziomy aktywacji


☒ Trafność ☐ Inne zmienne

☐ Standaryzowane reszty

 Podsumowanie

 Zapisz sieci ▾

[Anuluj](#)

 Opcje ▾

Próby

☒ Uczenie

☐ Test

☐ Walidacja

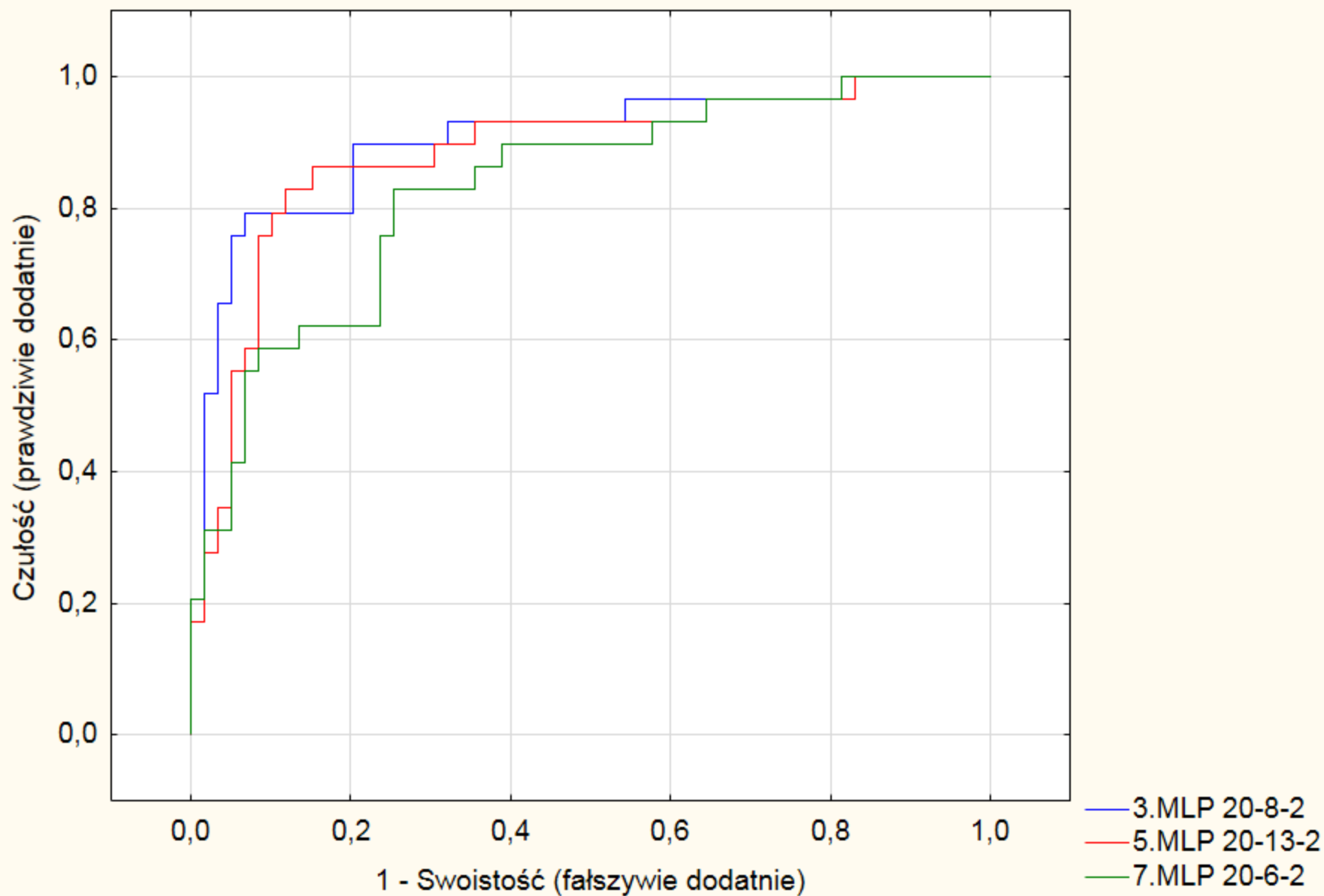
☐ Brakujące

Entropia a SOS

Podsumowanie aktywnych sieci (Kredyty1.sta)								
Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
1	MLP 20-8-2	68,18182	77,77778	77,77778	BFGS 2	Entropia	Tanh	Softmax

Podsumowanie aktywnych sieci (Kredyty1.sta)								
Id sieci	Nazwa sieci	Jakość (uczenie)	Jakość (testowanie)	Jakość (walidacja)	Algorytm uczenia	Funkcja błędu	Aktywacja (ukryte)	Aktywacja (wyjściowe)
3	MLP 20-8-2	86,36364	77,77778	77,77778	BFGS 9	SOS	Tanh	Liniowa

Krzywa ROC
Próby: Uczenie



Neuronowe Systemy Ekspertowe

- Sieci neuronowe można wykorzystać do budowy systemów ekspertowych, które na przykład mogą mieć zastosowanie do diagnostyki medycznej.
- Konwencjonalne podejście do budowy systemów ekspertowych wymaga stworzenia bazy wiedzy zawierającej sformułowane przez człowieka reguły wyrażające wiedzę z dziedziny ekspertyzy. Reguły te i dane wejściowe wykorzystywane są w procesie wnioskowania do otrzymania odpowiedzi na postawione systemowi pytania. Liczba reguł, które może zawierać system ekspertowy, może sięgać tysięcy.

- Oprócz ścisłych modeli danej dziedziny ekspert posiada wiedzę intuicyjną lub niejawną. Wiedza niejawna jest trudna do sformalizowania, a czasem nawet do werbalnego wyrażenia. Często próby przetłumaczenia wiedzy niejawnej na jawną prowadzą do utraty lub zniekształcenia informacji. Z tego względu dziedziny zawierające wiedzę niejawną są trudne do dokładnego przetworzenia na reguły. Poza tym inżynier wiedzy konstruujący system reguł musi wszechstronnie zrozumieć dziedzinę, której dotyczy budowany system, oraz sposób rozumowania eksperta z tej dziedziny, aby móc efektywnie wydobywać reguły dotyczące systemu. Wszystko to powoduje, że wyodrębnienie wiedzy jest istotnym problemem utrudniającym konstrukcję efektywnych systemów ekspertowych, a ich budowa pozostaje trudnym, kosztownym i czasochłonnym zadaniem.
- Ponieważ uczenie sieci neuronowych często zastępuje znajomość formalnych związków pomiędzy przetwarzanymi danymi, neuronowe systemy ekspertowe mogą stanowić atrakcyjną alternatywę dla systemów regułowych, zastępując uczeniem zadanie ekstrakcji wiedzy. Systemy ekspertowe oparte na sieciach neuronowych nazywane są systemami konekcyjnymi.
- Jednym z mankamentów systemów konekcyjnych są trudności w wyposażaniu ich w blok wyjaśniający. W konwencjonalnych systemach ekspertowych blok taki na pytanie użytkownika podaje przyczynę podjęcia określonej decyzji.

neuronowe systemy diagnostyki medycznej

- W obecnych czasach zbudowano już szereg bardzo dobrze działających systemów diagnozy komputerowej. Systemy regułowe wymagają jawnego sformułowania reguł diagnozy, co wiąże się z problemem ekstrakcji wiedzy. Wyrażenie wiedzy w postaci warunkowych gęstości prawdopodobieństw także jest trudne ze względu na często istotne, ale nieznane zależności pomiędzy danymi oraz ze względu na trudności określenia prawdopodobieństw występowania poszczególnych symptomów. Dopiero ostatnio rozwijające się zastosowania teorii zbiorów rozmytych mogą stanowić pomoc przy reprezentacji wiedzy niejawnej.
- W systemach konekcyjnych na węzły wejściowe sieci warstwowej podawane są informacje o wybranych symptomach i ich parametrach. Mogą to być wartości liczbowe przyporządkowane symptomom, результатам testów i medycznej historii pacjenta. W najprostszej wersji sygnały wejściowe mogą być binarne, przyjmujące wartość +1, jeżeli odpowiedni symptom występuje, lub -1 w przeciwnym przypadku. Brakowi danych może być przyporządkowana liczba 0. Liczba identyfikowanych chorób może być równa liczbie węzłów wyjściowych

Przykład: system do diagnozowania chorób skóry

- System ten dotyczy systemu ekspertowego dokonującego diagnozę chorób skóry typu grudkowo-złuszczającego. System ten zaprojektowany został głównie w celu wspomagania nauczania studentów medycyny.
- Sygnałami wejściowymi sieci były między innymi następujące symptomy i ich parametry: umiejscowienie, rozmieszczenie, kształt, liczba uszkodzeń, obecność aktywnych granic, liczba złuszczeń, wysokość brodawek, barwa, zmiana pigmentacji, swędzenie, krostowatość, limfadenopatia skóropochodna, zgrubienia na dłoniach, wyniki badań mikroskopowych, obecność wykwitu pierwotnego oraz wyniki KOH przy teście mikologicznym. Ponadto na wejście sieci podawano czas trwania objawów chorobowych, rozróżniając dni i tygodnie.

- System miał łącznie 97 wejść, w tym jedno wejście z sygnałem stałym, oraz 10 wyjść. Sygnały wejściowe przyjmowały trzy wartości: 0, 1, i 0.5 reprezentujące odpowiednio: występowanie poszczególnych symptomów, ich niewystępowanie oraz brak danych o symptomie. Sygnały wyjściowe poszczególnych neuronów w warstwie wyjściowej przedstawiają w reprezentacji lokalnej następujące 10 schorzeń: łuszczyca, łupież mieszkowy czerwony, liszaj płaski, łupież różowaty, łupież pstry, grzybica skóry, chłoniak z komórek T, wtórna kiła, przewlekłe kontaktowe zapalenie skóry oraz zapalenie łojotokowe skóry.
- Dane uczące pochodziły od 250 pacjentów. Sieć nauczono standardową metodą propagacji wstecznej. Jakość nauczonej sieci była testowana za pomocą danych nie używanych w procesie nauczania, zebranych od 99 pacjentów. Poprawną diagnozę otrzymano w 70% przypadków.
- Systemy neuronowe dochodzą do konkluzji na drodze wielu złożonych, nieliniowych i równoległych transformacji i wpływ na konkluzję mają wszystkie neurony. W związku z tym odtworzenie w nich i wyrażenie w postaci werbalnej łańcucha wnioskowania, wskazującego wpływ indywidualnego wejścia lub grupy wejść na ostateczną konkluzję jest bardzo trudne. Oczywiście, można badać eksperymentalnie wpływ wartości wejść na konkluzję, ale nie jest to równoważne ze znajomością łańcucha wnioskowania.

Sieć neuronowa	komputer
rozpoznawanie kojarzenie informacji klasyfikacja danych	obliczenia arytmetyczne
równoległe przetwarzanie danych - wiele neuronów działających w tym samym czasie	bardzo krótki czas przetwarzania jednego polecenia
zdolność do rekonstrukcji i odtworzenia sygnałów	
odporność na uszkodzenia	
zdolność przetwarzania informacji niepełnej i obciążonej błędami	wysoka precyzja obliczeń

Cechy SN:

- Adaptacyjność – zdolność uczenia się przez dostosowanie parametrów i struktury sieci do zmian otoczenia.
- Uogólnianie – zdolność wypracowania rozsądnych decyzji w sytuacjach, z którymi sieć nie miała dotąd styczności.
- Równoległość obliczeń
- Nieliniowość – rozumiana jako zdolność modelowania nieliniowych zależności wyjściowo- wejściowych przez sieć
- Odporność na błędy, czyli zdolność działania mimo pewnych uszkodzeń sieci, oraz zdolność uczenia na podstawie danych błędnych, niepełnych, rozmytych, nieprecyzyjnych.

Zadania sieci neuronowych:

- **Predykcja:** Sieci neuronowe są często wykorzystywane, aby na podstawie pewnych danych wejściowych przewidywać dane wyjściowe. Ważną zaletą jest to, że sieć może się nauczyć przewidywania sygnałów wyjściowych bez jawnego definiowania związku między danymi wejściowymi a wyjściowymi,
- **Klasyfikacja i rozpoznawanie:** Zadanie polega na przewidywaniu identyfikatora klasy, do której dany obiekt należy .
- **Kojarzenie danych:** Klasyczne systemy komputerowe mogą gromadzić duże zbiory danych bez możliwości ich kojarzenia. Sieci neuronowe, dzięki zdolności uczenia się i uogólniania doświadczeń, pozwalają zautomatyzować procesy wnioskowania i pomagają wykrywać istotne powiązania pomiędzy danymi.

Zadania sieci neuronowych:

Analiza danych: Zadanie polega na znalezieniu związków pomiędzy danymi. Realizacja tego zadania przez sieci neuronowe daje nowe możliwości w zakresie prowadzenia analiz ekonomicznych.

Filtracja sygnałów: Dane gospodarcze pochodzące z różnych źródeł są zakłócone. Klasyczne metody eliminacji szumów pozwalają usunąć zakłócenia o charakterze losowym, lecz nie dają podstaw do eliminacji przekłamań systematycznych.

Optymalizacja: Sieci neuronowe-zwłaszcza sieci Hopfielda- dobrze nadają się do optymalizacji decyzji gospodarczych. Doświadczalnie potwierdzono możliwości sieci do rozwiązywania zadań optymalizacji statycznej i dynamicznej. Szczególnie ciekawe jest zastosowanie sieci do optymalizacji kombinatorycznej i zagadnień bardzo trudnych obliczeniowo (również np- zupełnych), które mogą być rozwiązane w krótkim czasie dzięki współbieżnym wyliczaniu przez wiele elementów sieci. Dobrym przykładem jest się rozwiązującą klasyczny problem komiwojażera

Rozpoznawanie wzorców


Co można uznać za wzorzec ?

- Wzorcem może być zeskanowany obraz litery przy rozpoznawaniu znaków, przy czym litera może być drukowana lub pisana ręcznie np. do automatycznego rozpoznawania kodów pocztowych na listach. Wzorcem może być kształt części maszyny w fabryce wykorzystującej automaty, które muszą w odpowiedni sposób ją złapać w procesie produkcyjnym. Wzorcem może być sygnał akustyczny wytwarzany przez śrubę napędową okrętu podwodnego albo formacja w analizie technicznej kursów akcji z giełdy papierów wartościowych. Analizując problematykę udzielania kredytów bankowych w kontekście zdolności kredytowej podmiotów czy też wystawiania polis ubezpieczeniowych widać, że wszędzie występują wzorce.

Co zatem można rozpoznawać ?

- To już zależy od konkretnej potrzeby i zastosowania. Można rozpoznawać czy szum śrub napędowych okrętu podwodnego wskazuje na swój czy obcy ? Czy analiza danych dostarczonych przez klienta do banku potwierdzi jego wiarygodność jako kredytobiorcy ? do kompanii ubezpieczeniowej jako ubezpieczanego na konkretnych warunkach ? Czy dana formacja techniczna budowana ze zmian na kilku ostatnich sesjach w połączeniu z inną, a może jeszcze z innymi pozwoli na wskazanie zmian kursów akcji w przyszłości ?

A dlaczego należy rozpoznawać ?

- Odpowiedź  jest prosta. Wymienione wyżej przykładowe problemy charakteryzuje jedna cecha wspólna. **Wszystkie one nie mają rozwiązania algorytmicznego, a jeżeli ono nawet istnieje to dziś jest zbyt skomplikowane do znalezienia.** Dlatego Sztuczne Sieci Neuronowe są w takich przypadkach niezastąpione.

Możliwe zastosowania

- NASA wykorzystuje sieci neuronowe do sterowania ramieniem manipulatora działającego w ładowni promów kosmicznych, co pozwala na utrzymywanie manipulowanych obiektów w niezmiennym położeniu w warunkach nieważkości,
- Uczeni z New York University Medical Center zastosowali sieć jako alternatywę dla złożonych i czasochłonnych tensorowych obliczeń parametrów ruchu robota, co umożliwiło przyspieszenie działania systemu sterowania a przez to pracę robota w czasie rzeczywistym,
- Firma General Dynamics opracowała dla US Navy system opraty na sieci neuronowej klasyfikujący i rozpoznający sygnały sonarowe, pozwalający na identyfikację jednostki pływającej a nawet obiektów nadwodnych (np. helikopter unoszący się nad powierzchnią oceanu),
- Szpital Anderson Memorial Hospital w południowej Karolinie wykorzystał sieci neuronowe dla celów optymalizacji leczenia, pozwoliło na uzyskanie znacznych oszczędności z przede wszystkim na uratowanie życia kilkudziesięciu pacjentów,

Możliwe zastosowania c.d.

- Producent ракет Genera Devices Space Systems Division użył sieci neuronowych do sterowania pracą 150 zaworów doprowadzających paliwo i tlen do silników rakiety Atlas, co pozwoliło na zastąpienie dotychczas stosowanego kosztownego i zawodnego, złożonego systemu automatyki opartego na setkach sensorów,
- Firma Eaton Corporation wykorzystwała sieć neuronową w układzie sterowania wspomagającym pracę kierowcy dużej ciężarówki (pięć osi, osiemnaście kół) przy wykonywaniu niektórych szczególnie trudnych manewrów (np. cofanie z naczepą),
- Amerykańskie siły powietrzne (US Air Force) używają sieci neuronowe do rozwoju symulatorów lotu,
- Koncern Ford Motor Company przygotował nowy system diagnostyczny dla silników,
- Linie TWA stosują sieci neuronowe do lokalizacji bomb w swoim terminalu na lotnisku JFK w Nowym Yorku,
- W energetyce, np. w elektrowni BC Hydro w Vancouver sieci neuronowe znalazły zastosowanie do prognozowania zapotrzebowania na moc elektryczną,
- Firma Halliburton wykorzystuje sieci neuronowe do identyfikacji typu skał napotykanych podczas prowadzenia odwiertów przy poszukiwaniu złóż ropy i gazu.

Wady i zalety sieci neuronowej

Zalety:

- pozwala rozwiązywać problemy bez znajomości analitycznej zależności między danymi wejściowymi a oczekiwanymi wyjściami
- skuteczna w rozwiązywaniu problemów nieseparowalnych
- zdolność generalizacji
- różnorodność zastosowań: rozpoznawanie pisma, mowy, analizy finansowe rynku.

Wady:

- kłopotliwe dla niedoświadczonego użytkownika, jeśli sam ustala algorytm, parametry algorytmu oraz architekturę sieci
- niebezpieczeństwo przetrenowania lub niedouczenia sieci