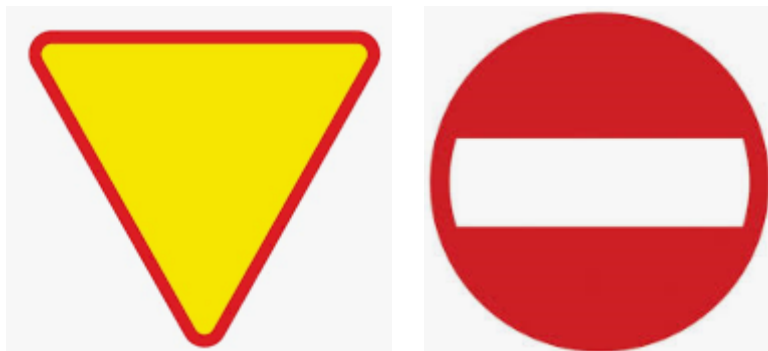


## Klasyfikacja – uczenie nadzorowane („z nauczycielem”)

Przykładowe problemy

1. Automatyczne rozpoznawanie znaków drogowych



## 2. Automatyczne określanie wiarygodności kredytowej

# Klasyfikacja

Analiza dyskryminacyjna

Badanie wiarygodności kredytowej klienta

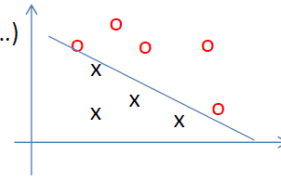
Majątek	Dochód	Zadłużenie	Liczba członków rodziny	Liczba lat pracy u obecnego pracodawcy	Splacił kredyt / nie splacił kredytu
98	35	12	4	4	1 x
65	44	5	3	1	1 x
22	50	0	2	7	1 x
8	23	12	2	1	0 o
0	15	10	4	2	0 o
21	12	28	3	2	1 x
57	39	13	5	8	0 o
...					

Nowy klient (majątek, dochód, zadłużenie, liczba członków rodziny, ...)

↓ ?  
Splacił kredyt / nie splacił kredytu

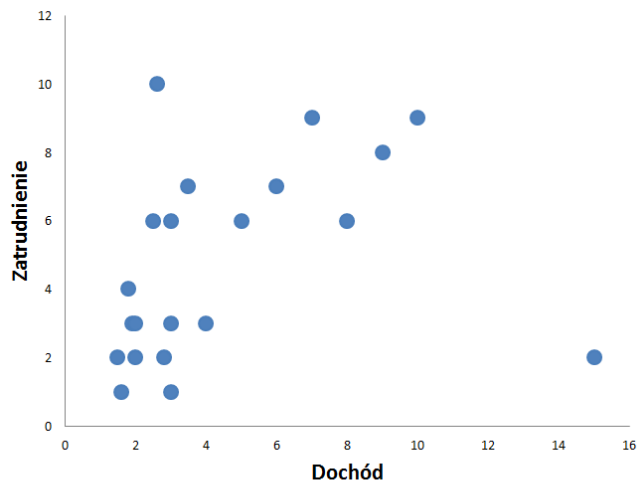
Inne przykłady:

- Email: zwykła treść / spam
- Rokowania pacjenta: pomyślne / niepomyślne



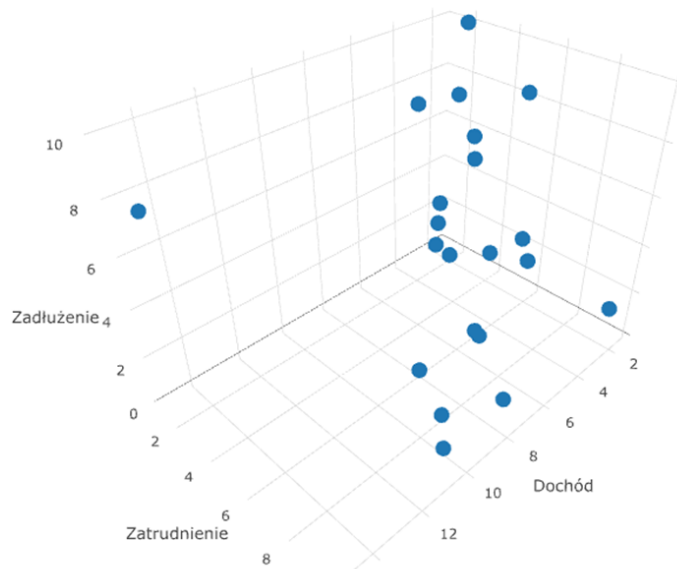
## Wizualizacja danych 2D

Dochód	Zatrudnienie
1,5	2
1,6	1
1,8	4
1,9	3
2	2
2	3
2,5	6
2,6	10
2,8	2
3	1
3	3
3	6
3,5	7
4	3
5	6
6	7
7	9
8	6
9	8
10	9
15	2

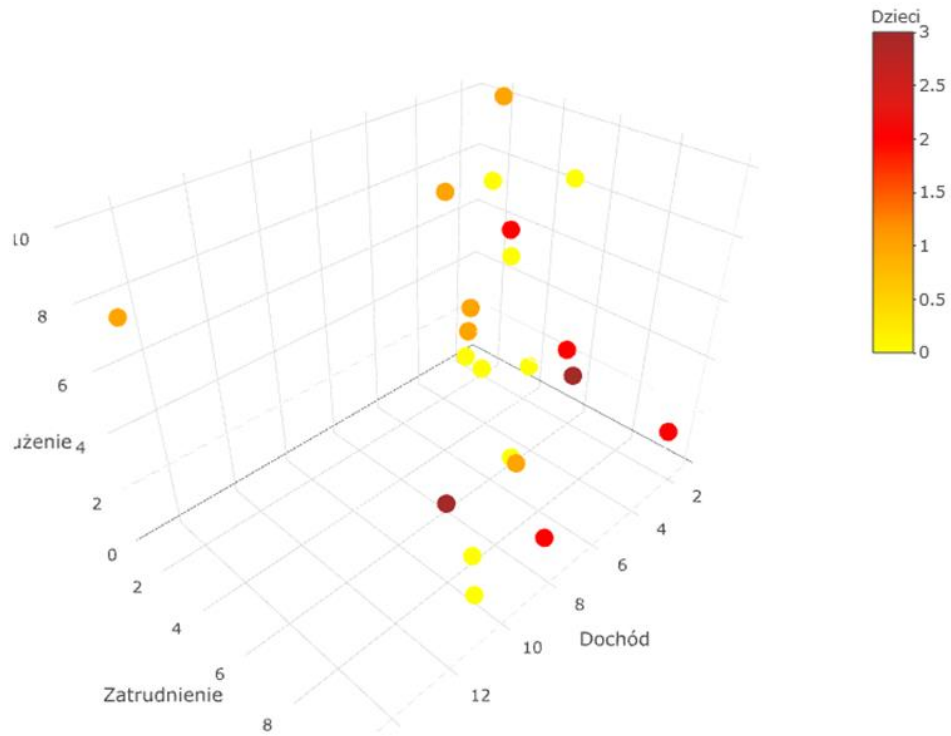


## Wizualizacja danych 3D

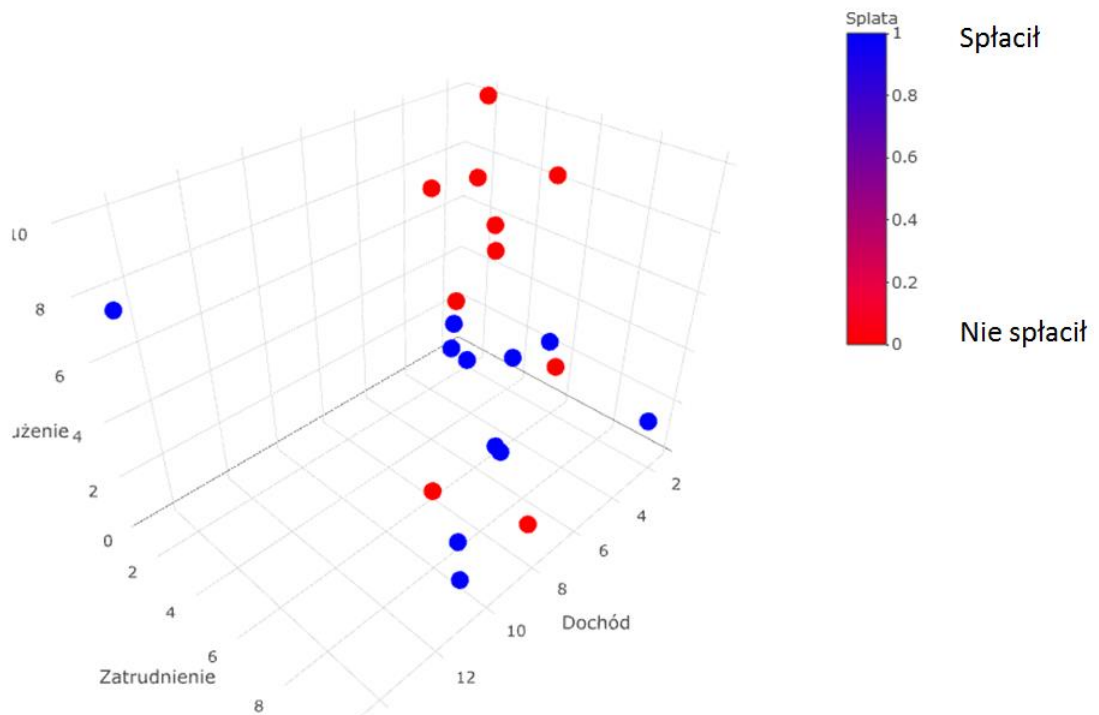
Dochód	Zatrudnienie	Zadłużenie
1,5	2	10
1,6	1	2
1,8	4	1
1,9	3	5
2	2	0
2	3	6
2,5	6	3
2,6	10	2
2,8	2	2
3	1	7
3	3	8
3	6	9
3,5	7	3
4	3	2
5	6	0
6	7	1
7	9	0
8	6	0
9	8	0
10	9	0
15	2	8

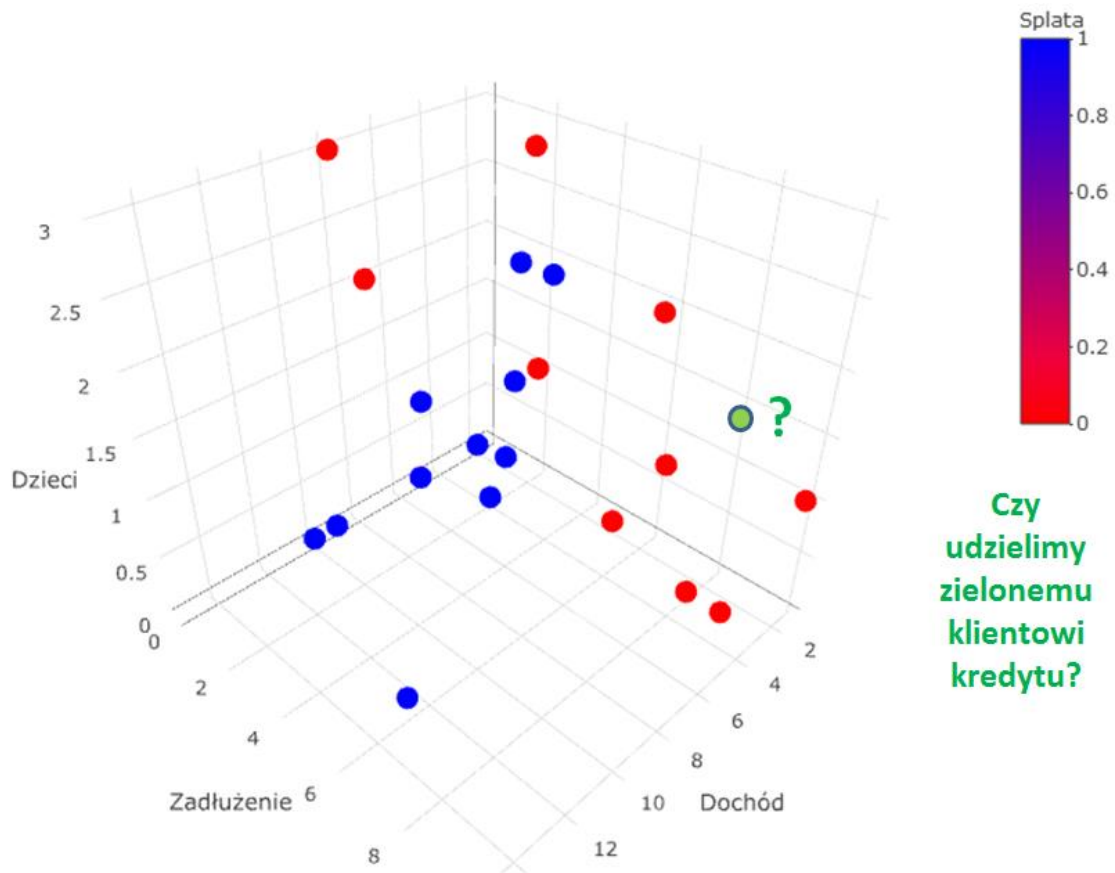


# Wizualizacja danych 4D

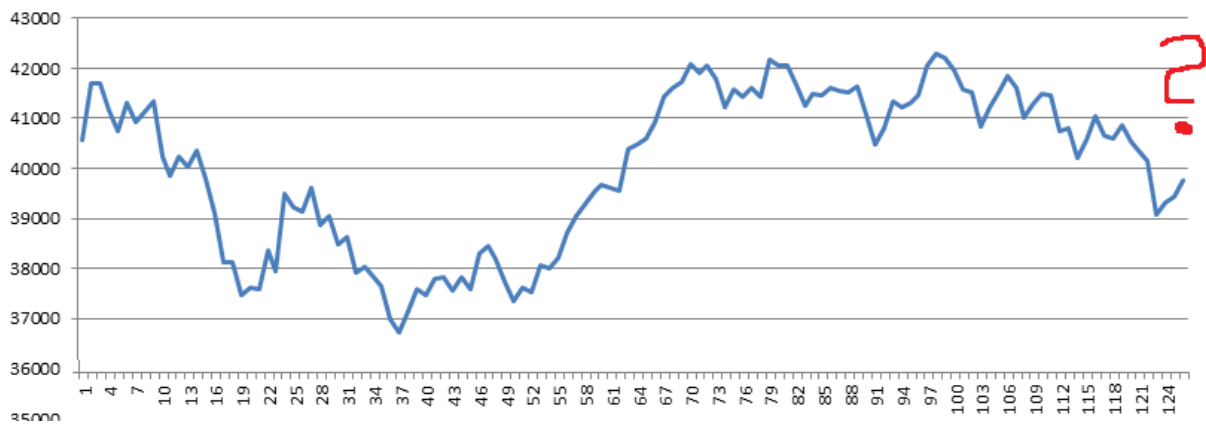


## Splata – czy klient spłacił kredyt





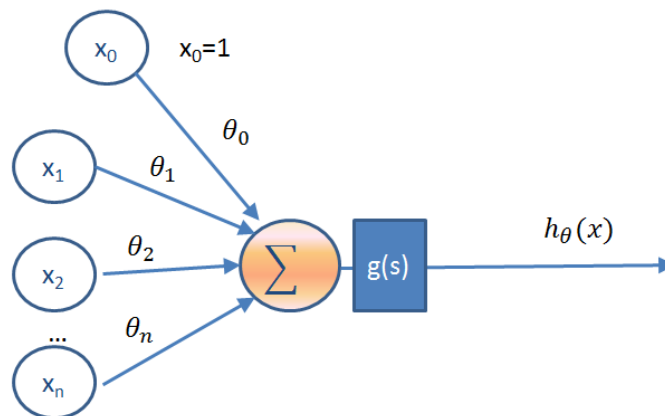
## Prognozowanie



## Sieci neuronowe

Sieci neuronowe stanowią wyrefinowaną technikę modelowania, inspirowaną działaniem ludzkiego mózgu, zdolną do odwzorowania nadzwyczaj złożonych funkcji. Sieci neuronowe składają się ze wzajemnie połączonych sztucznych neuronów, które realizują proste przekształcenia sygnałów wejściowych w wyjściowe.

Schemat blokowy pojedynczego neuronu przedstawiono na rysunku 1. Sztuczny neuron posiada wejścia  $x_0, x_1, \dots, x_n$ , którym przypisane są wagi  $\theta_0, \theta_1, \dots, \theta_n$ . Jądro neuronu działa na zasadzie sumatora sygnałów wejściowych przemnożonych przez wagi przypisane do ich połączeń. Obliczona suma przekazywana jest jako argument do funkcji aktywacji neuronu  $g(s)$  (rys. 2), generującej sygnał wyjściowy  $h_\theta(x)$ .

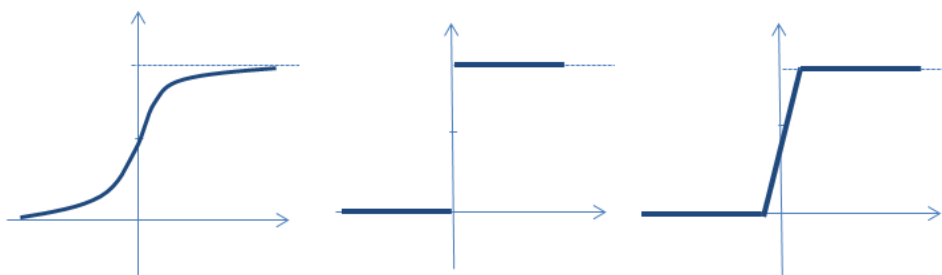


Rys. 1. Schemat blokowy sztucznego neuronu

W zapisie macierzowym działanie neuronu można przedstawić następująco:

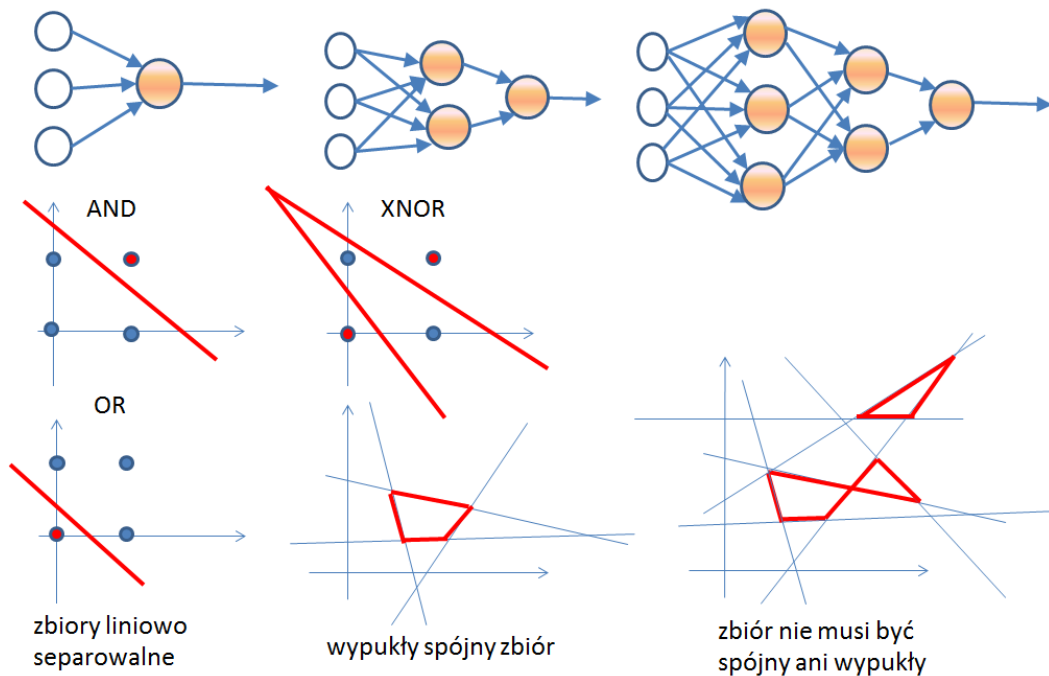
$$s = \theta^T x = \sum_{i=0}^n \theta_i x_i$$

gdzie:  $x = \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$  – wektor sygnałów wejściowych,  $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix}$  – macierz wag.

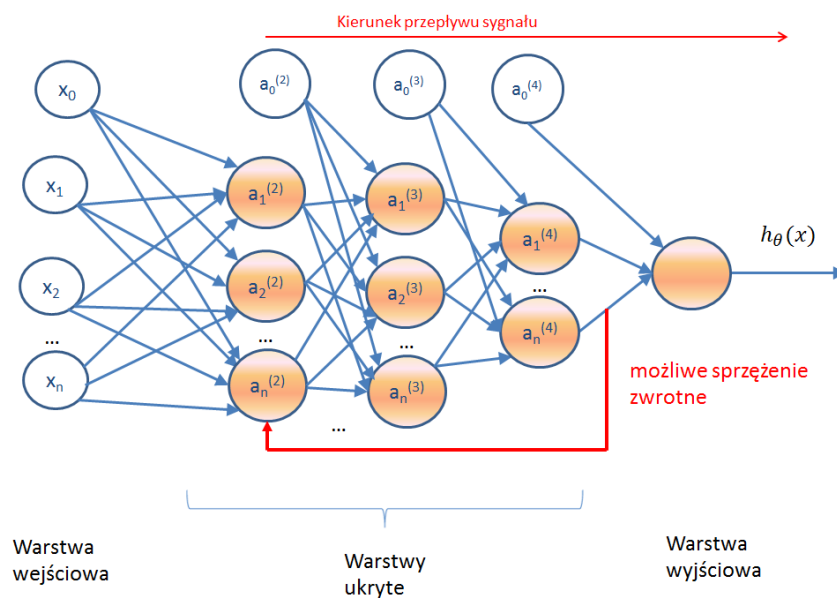


Rys. 2. Przykładowe funkcje aktywacji neuronu  $g(s)$

Pojedynczy neuron, z odpowiednio dobranymi wagami, może naśladować działanie bramki logicznej AND lub OR oraz klasyfikować zbiory liniowo separowalne (rys. 3). Do realizacji funkcji XNOR, której obrazem na płaszczyźnie jest spójny zbiór wypukły, potrzebna jest już sieć złożona z dwóch warstw neuronów  $[n\ 2\ 1]$ , czyli zawierająca minimum 3 neurony. Sieć złożona z minimum trzech  $[n\ 3\ 2\ 1]$  i większej liczby warstw jest w stanie klasyfikować dowolne zbiory (rys. 4).



Rys. 3. Możliwości sieci neuronowej w zależności od architektury



Rys. 4. Schemat blokowy sztucznej sieci neuronowej

Na rysunku 4 przedstawiono schemat sztucznej sieci neuronowej złożonej z 4 warstw neuronów (3 warstwy ukryte i 1 warstwa wyjściowa). Kluczową kwestią przy wyborze struktury sieci jest decyzja dotycząca obecności sprzężenia zwrotnego. W najczęściej stosowanych sieciach jednokierunkowych (bez sprzężenia zwrotnego) sygnał przepływa tylko w jednym kierunku, co zapewnia stabilne zachowanie sieci. Sprzężenie zwrotne zwiększa możliwości obliczeniowe sieci (obsługa rekurencji), ale wprowadza chaos deterministyczny.

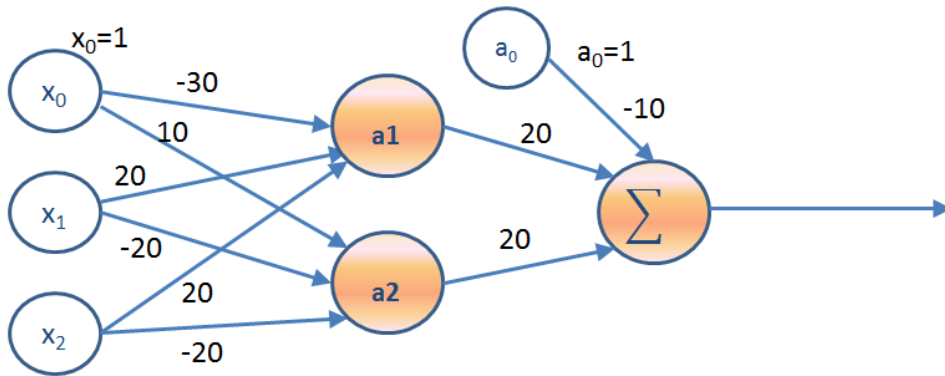
Najważniejszą cechą sieci neuronowych jest możliwość uczenia, tzn. ustalania optymalnych wag sieci na podstawie zbioru uczącego. Zbiór uczący budowany jest z danych pierwotnych, które najczęściej podlegają wstępnemu przetwarzaniu, obejmującemu standaryzację, normalizację, redukcję wymiaru poprzez eliminację współliniowości zmiennych niezależnych, filtrację lub inne czynności mające na celu poprawę jakości danych, wzmocnienie kontrastów, uwypuklenie różnic, eliminację niepożądanych składowych (szumu, zakłóceń). Zakres czynności wykonywanych w ramach wstępnego przetwarzania danych zależy od rodzaju i jakości danych pierwotnych oraz charakteru modelowanego problemu. Wstępne przetwarzanie danych oraz budowa zbioru uczącego wywiera decydujący wpływ na efektywność konstruowanej sieci neuronowej i jest etapem wymagającym doświadczenia oraz wiedzy dziedzinowej. Kolejne etapy obejmujące dobór architektury, przeprowadzenie procesu uczenia, testowania i walidacji modelu mają charakter stricte techniczny, niezależny od dziedziny problemu, a ich realizacja wspierana jest przez dostępne narzędzia.

W doborze optymalnej architektury sieci neuronowej, ustalaniu optymalnych wag sieci oraz doborze optymalnego zestawu zmiennych niezależnych podawanych na wejścia sieci, obok tradycyjnych rozwiązań, z powodzeniem stosowane są algorytmy genetyczne.

Szeroki zakres zastosowań sieci neuronowych wynika z ich efektywności oraz łatwości tworzenia złożonych modeli nieliniowych i ich użytkowania. Sieci neuronowe najczęściej stosowane są w zagadnieniach klasyfikacyjnych oraz związanych z prognozowaniem. Wykazują również przydatność w zagadnieniach optymalizacyjnych oraz zagadnieniach optymalnego sterowania.



# Sieć neuronowa – czym jest?



- zespół prostych, wzajemnie połączonych jednostek (neuronów) równoległe przetwarzających informację
- jednostki (neurony) wykonują sumowanie i nieliniowe odwzorowanie sygnałów wejściowych
- oddziaływanie każdego połączenia wyrażone jest wagą, która może być modyfikowana

Modyfikacja wag następuje w procesie uczenia

System przetwarzania informacji, uniwersalny dzięki możliwości uczenia.

## Projekt sieci neuronowej

Architektura sieci	<ul style="list-style-type: none"><li>• liczba warstw</li><li>• liczba neuronów w warstwie</li><li>• połączenia zwrotne</li><li>• kodowanie sygnału wejściowego</li><li>• kodowanie sygnału wyjściowego</li></ul>
Funkcja aktywacji	np. funkcja sigmoidalna, $\text{sgn}(x)$
Wagi	Uczenie sieci neuronowej Wiedza sieci neuronowej tkwi w jej wagach

# Proces uczenia

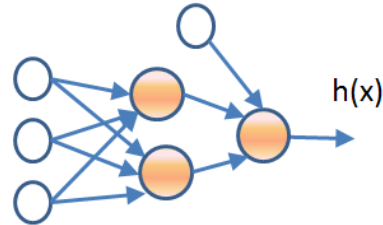
1. Losowe wygenerowanie wag inicjalnych
2. Dane wejściowe – ciąg uczący

$$\theta^{(1)} = \begin{bmatrix} \theta_{10}^{(1)} & \theta_{11}^{(1)} & \theta_{12}^{(1)} \\ \theta_{20}^{(1)} & \theta_{21}^{(1)} & \theta_{22}^{(1)} \end{bmatrix}$$

$$\theta^{(2)} = \begin{bmatrix} \theta_{10}^{(2)} & \theta_{11}^{(2)} & \theta_{12}^{(2)} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

$m$  próbek o  $n$  cechach i wyniku  $y_m$

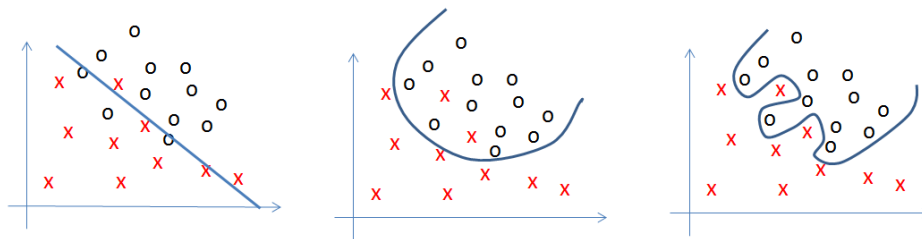


3. Podaj wektor  $X_m$  na wejście i oblicz  $h(X_m)$
4. Porównaj otrzymane  $h(X_m)$  z  $Y_m$  i skoryguj wagi

Powtórz  $m$  razy

Algorytmy korekty wag (np. reguła uczenia perceptronu, propagacja wsteczna błędu)

$N = [3 \ 20 \ 20 \ 20 \ 1]$ ;



$$h_{\Theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

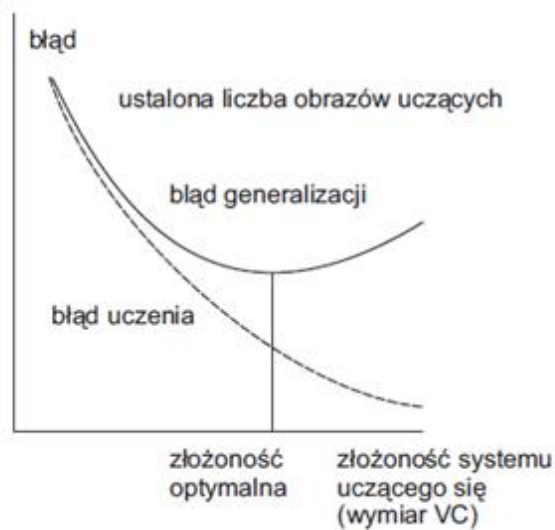
$$h_{\Theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

- Problem przeuczenia sieci (*overfitting*) – utrata zdolności generalizujących

**Generalizacja** – zdolność systemu do udzielania poprawnych odpowiedzi na sygnały wejściowe, które nie wystąpiły w zbiorze uczącym.

Brzytwa Ockhama – „Nie należy mnożyć bytów ponad potrzebę”

Podstawowy cel uczenia – nabycie przez system zdolności generalizujących.



Ustalona liczba obrazów uczących

## Zastosowania sieci neuronowych

- Klasyfikacja
- Grupowanie
- Optymalizacja
- Pamięć asocjacyjna
- Predykcja (generalizacja)
- ...

Równoległy, rozproszony system obliczeniowy

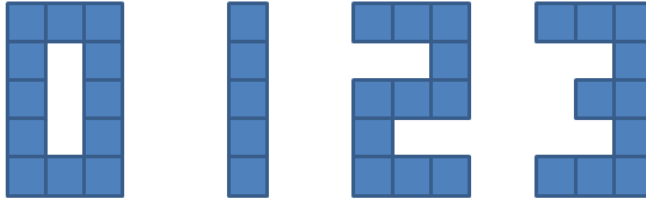
programowanie liniowe  
problem komiwojażera  
rozpoznawanie obrazów  
kompresja danych  
....

- Modelowanie przebiegu i skutków chorób neurodegradacyjnych
- Modelowanie wpływu substancji chemicznych na mózg
- Modelowanie procesów uczenia
- ...

Sztuczny mózg

## Przykład – rozpoznawanie cyfr

Wzorce



Kodowanie – macierze 5 x 3 z wartościami {0,1}

1 1 1	0 1 0	1 1 1	1 1 1
1 0 1	0 1 0	0 0 1	0 0 1
1 0 1	0 1 0	1 1 1	0 1 1
1 0 1	0 1 0	1 0 0	0 0 1
1 1 1	0 1 0	1 1 1	1 1 1

Wyjście

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

## Przykład – rozpoznawanie cyfr

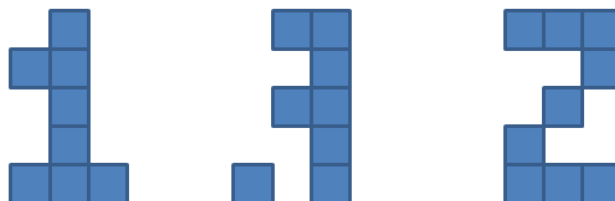
SciLab – moduł ANN (*Artificial Neural Network*)

Wybór architektury sieci     `N = [15, 15, 15, 15, 4];`     Wiedza sieci zawarta  
w macierzy wag

Inicjalizacja wag     `W = ann_FF_init (N);`

Wybór algorytmu uczenia     `[W, sW] = ann_FF_Mom_online (P, T, N, W, lp, Te);`

Podajemy     `Y = ann_FF_run (cyfra (:), N, W);`



Otrzymujemy Y =

0.0722130	0.1388426	0.2046233
0.7243399	0.1692661	0.2022744
0.0884518	0.1202711	0.2706005
0.2594783	0.6177673	0.2746726

**Przykład 1. Rozpoznawanie wzorców. Przykładowy plik sieci1.sci. Exec('sieci1.sci')**

1) Należy przygotować 4 własne wzorce (cyfry, litery) jako macierze o elementach {0,1}.

Poniżej przykład 4 wzorców 5x3 definiujących cyfry 0,1,2,3

1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0
1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0

```
T0=[1 1 1; 1 0 1; 1 0 1; 1 0 1; 1 1 1];
```

```
T1=[0 1 0; 0 1 0; 0 1 0; 0 1 0; 0 1 0];
```

```
T2=[1 1 1; 0 0 1; 1 1 1; 1 0 0; 1 1 1];
```

```
T3=[1 1 1; 0 0 1; 0 1 1; 0 0 1; 1 1 1];
```

2) Przygotowane wzorce należy umieścić we wspólnej macierzy tak, by każdy wzorzec stanowił kolumnę macierzy.

```
P=[T0(:) T1(:) T2(:) T3(:)];
```

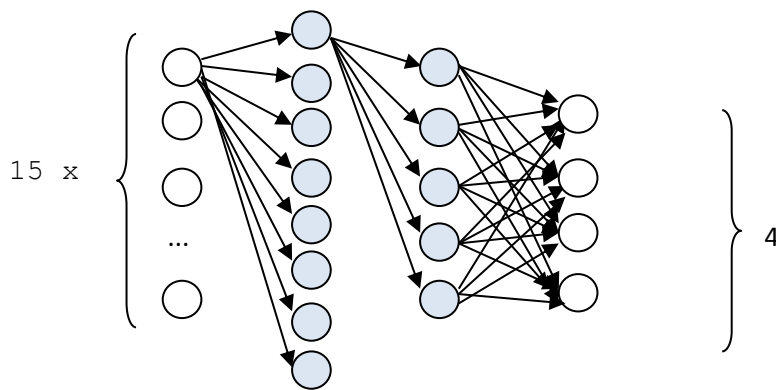
3) W oddzielnej macierzy należy przygotować zakodowaną postać wyniku odpowiadającego poszczególnym wzorcom.

```
T=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
```

4) Zdefiniować architekturę sztucznej sieci neuronowej

```
N=[15 8 5 4];
```

```
W=ann_FF_init(N);
```



Każdy z 15 neuronów wejściowych jest połączony z każdym z 8 neuronów warstwy 1 (na rysunku narysowano połączenia jedynie 1-go neuronu wejściowego). Każdy z 8 neuronów warstwy 1-szej jest połączony z każdym z 5 neuronów warstwy 2-giej.

5) Określić parametry algorytmu uczenia sieci

```
lp=[0.1 0.05 0.5 0.1];
```

```
Te=100;
```

6) Przeprowadzić uczenie sieci

```
[W,sW]=ann_FF_Mom_online(P,T,N,W,lp,Te);
```

7) Zbadać jak sieć zareaguje na podanie wzorców

```
Y=ann_FF_run(P,N,W);
```

8) Przygotować 3-4 przykłady zaburzonych cyfr/liter i sprawdzić odpowiedź sieci.

```
W1=[0 1 0;1 1 0;0 1 0; 0 1 0; 1 1 1];
```

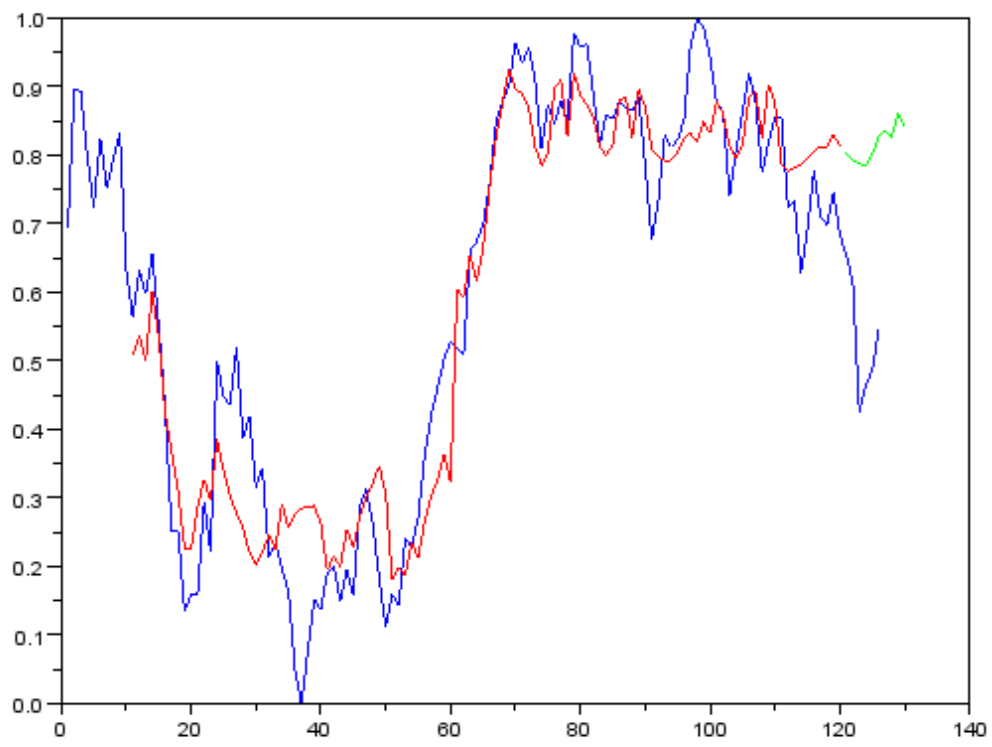
```
Y=ann_FF_run(W1(:),N,W);
```

9) Zmienić wybraną architekturę sieci i zbadać wpływ zmiany na wyniki sieci

**Przykład 2. Predykcja zmian indeksu WIG na podstawie danych historycznych w plikach WIG\_dane.xls, dane\_wig.txt i dane\_obrot.txt.**

**Przykładowe pliki sieci2.sci i sieci3.sci. Exec('sieci2.sci'). Exec('sieci3.sci').**

- 1) Wczytanie danych dotyczących historycznych notowań indeksu WIG  
`Z=read('c:\laborat\mmt01\dane_wig.txt',126,1);`
- 2) Normalizacja danych  
`Z=(Z-min(Z))/(max(Z)-min(Z));`
- 3) Wybór architektury sieci  
`N=[10 10 10 10 10];`
- 4) Określenie parametrów uczenia sieci  
`lp=[0.1 0.05 0.5 0.1];`  
`Te=500;`
- 5) Przygotowanie wzorców i odpowiedzi  
`P=zeros(10,11);`  
`T=zeros(10,11);`  
`for j=1:11`  
    `for i=1:10`  
        `P(i,j)=Z((j-1)*10+i,1);`  
        `T(i,j)=Z(j*10+i,1);`  
    `end`  
`end`
- 6) Inicjalizacja wag sieci  
`W=ann_FF_init(N);`
- 7) Uczenie sieci  
`[W,sW]=ann_FF_Mom_online(P,T,N,W,lp,Te);`
- 8) Uzyskanie odpowiedzi na wzorzec  
`Y1=ann_FF_run(P,N,W);`
- 9) Przygotowanie nowego wzorca  
`D=Z(111:120,1);`
- 10) Uzyskanie odpowiedzi na nowy wzorzec (predykcja)  
`Y2=ann_FF_run(D,N,W);`
- 11) wykresy  
`xx=1:1:130;`  
`plot(xx(1:126),Z);plot(xx(11:120),Y1(:),'r');`  
`plot(xx(121:130),Y2,'g');`





**Przykład 3** Przygotowanie prognozy dot. liczby wypadków / zabitych / rannych dla wybranego województwa

## Krok 1 Wczytanie danych

Dane przykładowe (źródło GUS)

Lp	Województwo	Ludność	PKB [mln zł]	Wynagr. Brutto [zł]	Liczba pojazdów	Liczba stacji paliw	Drogi publiczne	Drogi publiczne na 100 km <sup>2</sup>
		[1]	[2]	[3]	[4]	[5]	[6]	[7]
1	Dolnośląskie		110448	3627.02	1655833	612	23468.0	117.7
2	Kujawsko-pomorskie		61721	3139.23	1253158	574	26480.8	147.3
3	Lubelskie		51082	3279.39	1322252	668	34012.7	135.4
4	Lubuskie		30358	3132.90	609231	338	13218.2	94.5
5	Łódzkie		81869	3301.92	1568366	900	25556.1	140.3
6	Małopolskie		99509	3460.83	1898517	673	30369.3	200.0
7	Mazowieckie		292974	4598.69	2541044	1508	52302.9	147.1

Wczytanie danych z pliku tekstowego do macierzy o podanych wymiarach 16x13

→ dane=read('c:\laborat\mmt\wypadki01.txt',16,13)

dane =

```
2877840. 110448. 3627.02 1655833. 612. 23468. 117.7 3936. 242.6 2393.
2069543. 61721. 3139.23 1253158. 574. 26480.8 147.3 1714. 189.7 1414.
2151895. 51082. 3279.39 1322252. 668. 34012.7 135.4 1718. 98.9 1266.
1011024. 30358. 3132.9 609231. 338. 13218.2 94.5 1334. 45.4 1051.
2534357. 81869. 3301.92 1568366. 900. 25556.1 140.3 5040. 189.9 1013.
3310094. 99509. 3460.83 1898517. 673. 30369.3 200. 5053. 389.7 1388.
5242911. 293974. 4598.69 3541044. 1508. 52302.9 147.1 8139. 1117. 3038.
1028585. 29680. 3345.88 630900. 207. 11445.5 121.6 579. 28.7 495.
2103505. 50684. 3093.91 1174565. 636. 18370.3 102.9 2793. 58.2 1124.
1188329. 30903. 3241.73 697228. 292. 24195.4 119.9 1108. 99.5 1012.
2240319. 76243. 3654.94 1313756. 448. 22338.7 122. 3375. 303.1 2106.
4635882. 175324. 3764.92 2521534. 799. 25683.8 208.3 11257. 461. 2889.
1266014. 34747. 3214.38 783698. 337. 16915.8 144.4 2298. 46.1 759.
1427241. 37076. 3082.25 782601. 355. 22569. 93.4 1470. 73.6 1328.
3419426. 127361. 3364.8 2346895. 936. 40277.4 135. 4856. 381.5 2393.
1693072. 52389. 3341. 937571. 455. 18918.2 82.6 1617. 180. 2146.
```

Powstała macierz o nazwie *dane* i wymiarach 16 x 13. Macierz zawiera m=16 próbek o n=13 cechach.

[1] Ludność stan na 31.12.2010

[2] Produkt Krajowy Brutto w 2010 w mln zł

[3] Przeciętne m-czne wynagrodzenie brutto w gospodarce narodowej w I-III kw. 2011 w zł

[4] Pojazdy samochodowe i motorowery stan na 31.12.2010

[5] Liczba stacji paliw w 2009

[6] Drogi publiczne w km

[7] Drogi publiczne na 100km<sup>2</sup>

[8] Linie komunikacji miejskiej w km

[9] Przewozy pasażerów komunikacją miejską w mln w 2010

[10] Przewozy ładunków transportem samochodowym w tyś ton (nadano do przewozu wewnątrz województwa) w 2010

[11] Liczba wypadków w 2011

[12] Liczba zabitych w 2011

[13] Liczba rannych w 2011

## Krok 2 Normalizacja danych

Wyjściowy zbiór zmiennych						Y
x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	
23	0.1	20000	2	120	24.3	
12	0.4	5000	3	234	12.1	
26	0.25	1000	20	13	10.9	

Podzbiór zmiennych diagnostycznych



Normalizacja

$$x_{ij} = \frac{x_{ij} - \min_i x_{ij}}{\max_i x_{ij} - \min_i x_{ij}}$$

Znormalizowany zbiór zmiennych						Y
x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	
	0	1	0			
	1	0.2105	0.0505			
	0.5	0	1			

Wybór zmiennych wejściowych i wyjściowych

Normalizacja wybranych zmiennych

```
-->A=dane(:,1); // wybór pierwszej kolumny Ludność
```

```
-->A=(A-min(A))/(max(A)-min(A)) // normalizacja
```

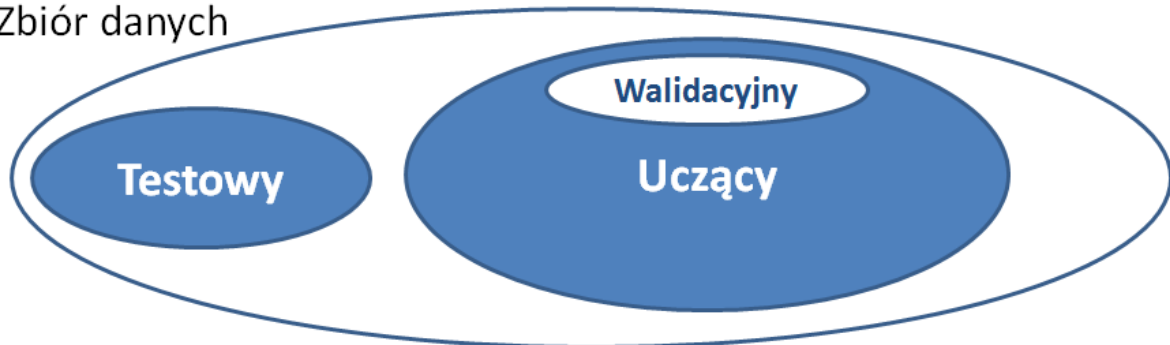
```
A =
```

```
0.4411309
0.2501293
0.2695892
0.
0.3599654
0.5432730
1.
0.0041497
0.2581546
0.0418974
0.2904839
0.8565583
0.0602544
0.0983526
0.5691083
0.1611688
```

B,C,D ... i zmiennej wyjściowej W

### Krok 3 Podział zbioru danych

Zbiór danych



Dokonyjemy podziału np. wiersze 1-12 zbiór uczący, 13-15 zbiór walidacyjny i 16 zbiór testowy

```
->Aucz=A(1:12,1)
Aucz =

    0.4411309
    0.2501293
    0.2695892
    0.
    0.3599654
    0.5432730
    1.
    0.0041497
    0.2581546
    0.0418974
    0.2904839
    0.8565583

-->Awal=A(13:15,1)
Awal =

    0.0602544
    0.0983526
    0.5691083

-->Atest=A(16,1)
Atest =

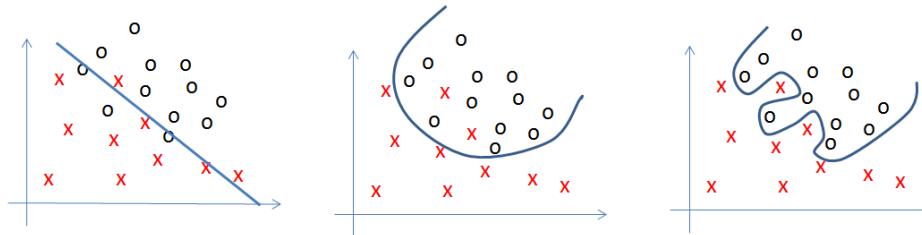
    0.1611688

-->Zucz=[Aucz B(1:12,1) C(1:12,1)]
// utworzenie macierzy zawierającej znormalizowane wybrane cechy
Zucz =

    0.4411309  0.3569812  0.7463626
    0.2501293  0.2196344  0.3613842
    0.2695892  0.2432014  0.3031852
    0.    0.    0.2186394
    0.3599654  0.3271474  0.2036964
    0.5432730  0.4397572  0.3511600
    1.    1.    1.
    0.0041497  0.0073910  0.
    0.2581546  0.1928274  0.2473457
    0.0418974  0.0300145  0.2033032
    0.2904839  0.2403035  0.6335037
    0.8565583  0.6522595  0.9414078
```

## Krok 4 Wybór architektury sieci

$N=[3\ 20\ 20\ 20\ 1]$ ;



$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

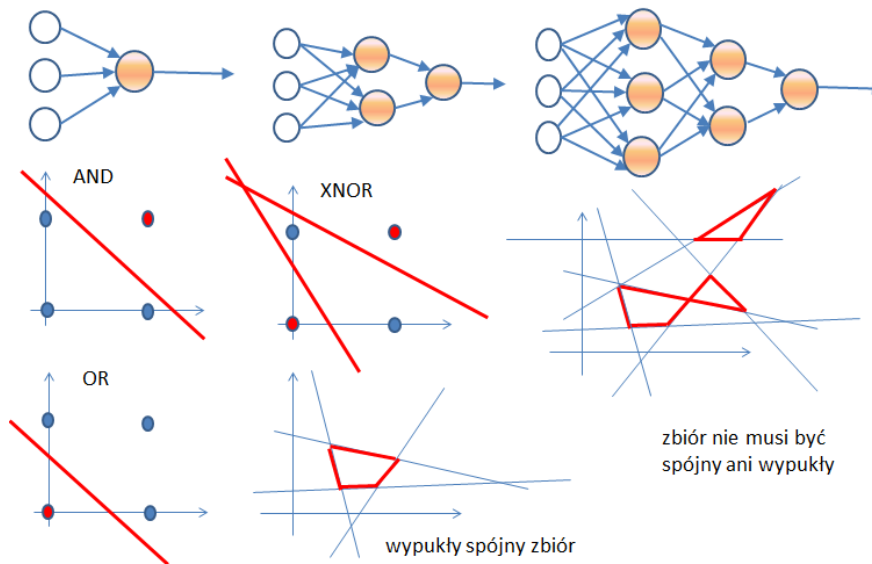
- Problem przeuczenia sieci (*overfitting*) – utrata zdolności generalizujących

**Generalizacja** – zdolność systemu do udzielania poprawnych odpowiedzi na sygnały wejściowe, które nie wystąpiły w zbiorze uczącym.

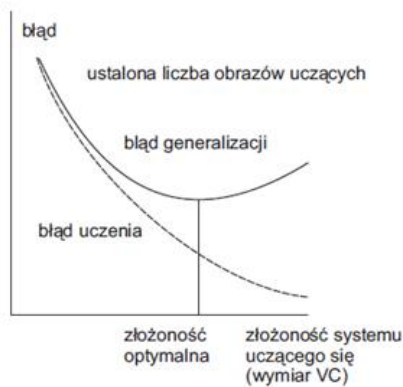
Brzytwa Ockhama – „Nie należy mnożyć bytów ponad potrzebę”

Podstawowy cel uczenia – nabycie przez system zdolności generalizujących.

### Separowalność



## Krok 5 Uczenie sieci



Ustalona liczba obrazów uczących

```
W=ann_FF_init(N);  
lp=[0.1 0.05 0.5 0.1];  
Te=200;  
[W,sW]=ann_FF_Mom_online(Zucz',Yucz',N,W,lp,Te);
```

Ocena błędu uczenia:

```
Y1=ann_FF_run(Zucz',N,W)  
blad=(Y1-Yucz');  
bladucz=sum(blad.^2)/12
```

Ocena błędu generalizacji:

```
Y2=ann_FF_run(Zwal',N,W)  
blad=(Y2-Ywal');  
bladwal=sum(blad.^2)/3
```

## Krok 6 Prognoza

```
Y3=ann_FF_run(Ztest',N,W)  
Y(16,1)
```

## Zadania do wykonania

### Zadanie 1 (15 pkt). Rozpoznawanie wzorców. Przykładowy plik sieci1.sci.

Zadanie polega na :

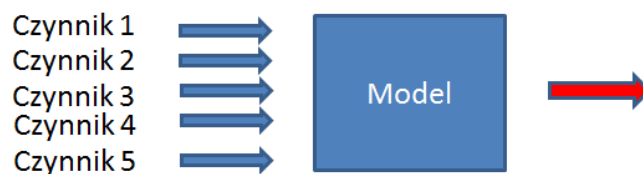
- 2) przygotowaniu 4 własnych wzorców liter jako macierzy o elementach {0,1}.
- 3) przygotowaniu 4-6 własnych liter 'zaburzonych' jako macierzy o elementach {0,1}.
- 4) przeprowadzeniu procesu uczenia sieci o różnych architekturach (zmienna lub stała liczba epok uczenia  $T_e$ ) i udokumentowaniu wyników (odpowiedzi sieci na uczone wzorce) w postaci krzywej uczenia
- 5) przetestowaniu odpowiedzi sieci o różnych architekturach na litery 'zaburzone' i udokumentowaniu wyników w postaci krzywej generalizacji
- 6) wyborze architektury optymalnej
- 7) skomentowaniu wyników

### Zadanie 2. Budowa modelu BRD (10 pkt) Przykładowy plik sieci\_wypadki.sci, wypadki01.txt

Zestaw 1	dane wejściowe kolumny 1-5, dana wyjściowa kolumna 11 (liczba wypadków)
Zestaw 2	dane wejściowe kolumny 6-10, dana wyjściowa kolumna 11 (liczba wypadków)
Zestaw 3	dane wejściowe kolumny 3-7, dana wyjściowa kolumna 12 (liczba zabitych)
Zestaw 4	dane wejściowe kolumny 1,2,8-10, dana wyjściowa kolumna 12 (liczba zabitych)
Zestaw 5	dane wejściowe kolumny 3,4,8-10, dana wyjściowa kolumna 13 (liczba rannych)

Etapy:

- 1) normalizacja danych wejściowych i wyjściowych
- 2) podział zbioru danych na zbiór uczący i testowy
- 3) przeprowadzenie procesu uczenia, wybór architektury optymalnej modelu na podstawie analizy średniego błędu odpowiedzi na zbiór uczący i zbiór testowy
- 4) przeprowadzenie badania istotności wpływu czynników wejściowych na modelowaną miarę BRD



Można wyznaczyć wartość średnią i odchylenie standardowe każdego z czynników i modelowanej miary. Zmieniając wartości czynników (procentowe lub +/- wielokrotność odchylenia standardowego) można badać zmiany odpowiedzi modelu. Porównanie wpływu zmian każdego z czynników może stanowić podstawę do wyciągnięcia wniosków dotyczących istotności.

Uwaga: Standardowa procedura budowy modelu jest bardziej złożona od. Obejmuje dodatkowo selekcję zmiennych wejściowych, mającą na celu pominięcie zmiennych skorelowanych, analizę reszt i inne kroki.