

Algorytmy Ewolucyjne

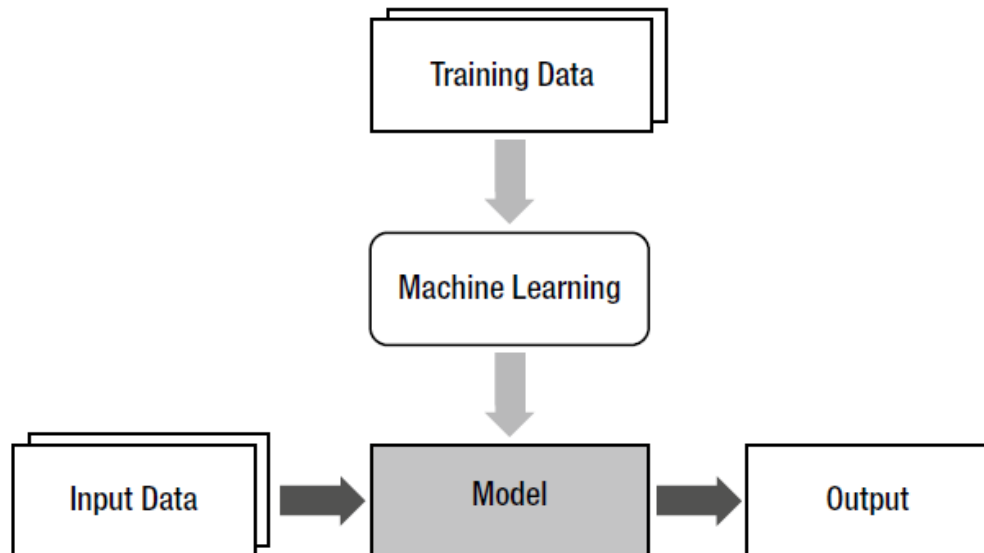
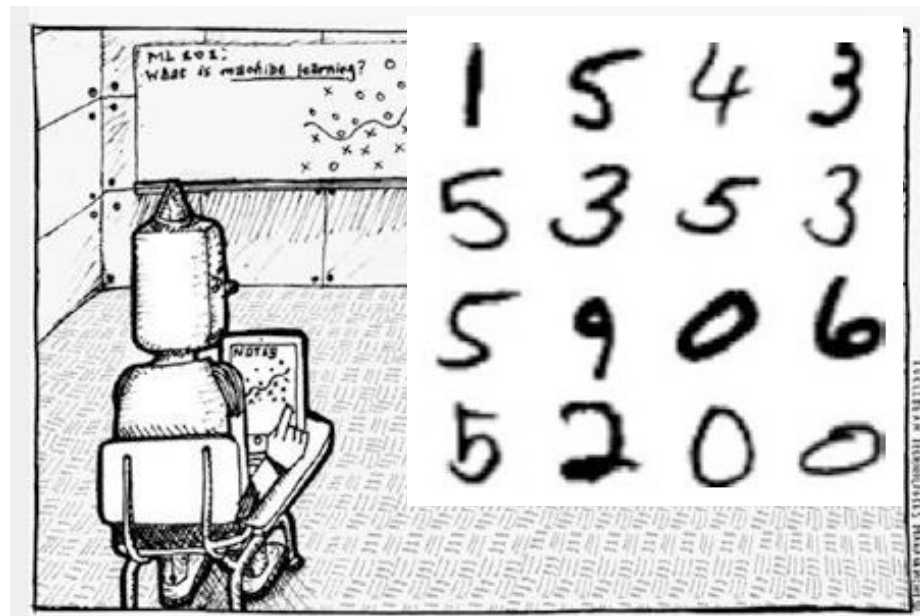
Klasyfikatory - wprowadzenie

Informacje organizacyjne

- Prowadzący: dr inż. Grzegorz Bogdan
- E-mail: G.Bogdan@ire.pw.edu.pl
- MS Teams: Grzegorz.Bogdan@pw.edu.pl
- Konsultacje: wtorki 14:15-15:00, środy 11:15-12:00, czwartki 15:15-16:00, MS Teams
- Plan części wykładowej:
 - 4 dwugodzinne wykłady poświęcone podstawowym technikom uczenia maszynowego tzn. sieciom neuronowym oraz maszynom wektorów nośnych
 - 2 prace domowe (po 2 pkt) - czas na wykonanie i złożenie krótkiego sprawozdania to 7 dni od momentu ogłoszenia pracy domowej
 - kartkówka (15 pkt)
 - projekt 3 (15 pkt)
- Literatura uzupełniająca:
 - Osowski, Stanisław., *Metody i narzędzia eksploracji danych*. Wydawnictwo BTC, 2013
- Uwaga dodatkowa:
 - 10 maja mija termin oddania projektu 2.

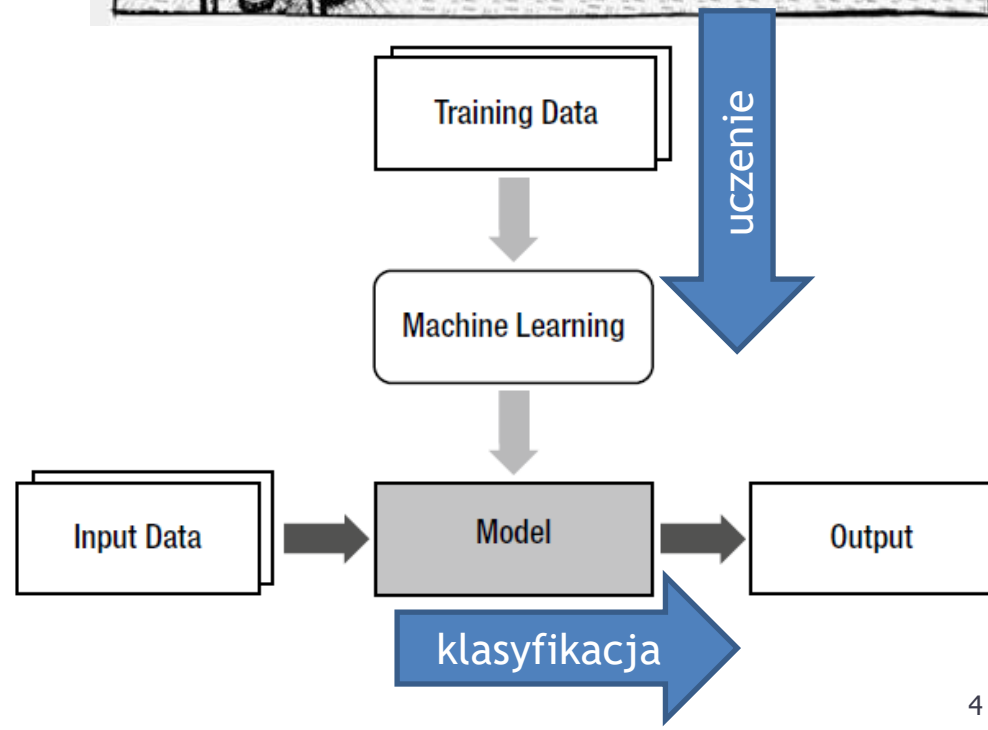
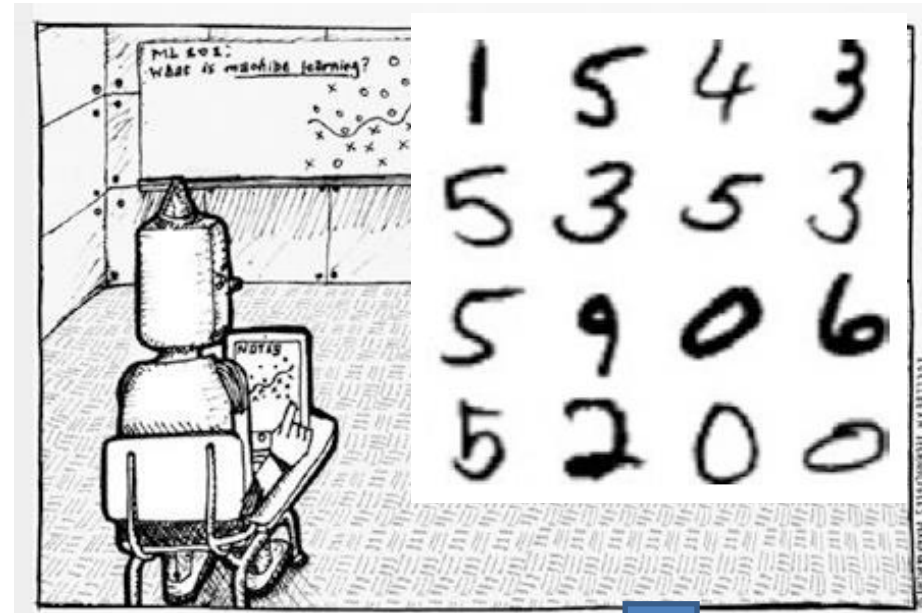
Uczenie maszynowe (ang. Machine Learning)

- Učeniem maszynowym nazywamy pewną grupę technik Sztucznej Inteligencji
- Sieć neuronowa jest jedną z technik stosowanych w uczeniu maszynowym
- **Uczenie** rozumiane jest jako rozwiązanie **problemu** lub opracowanie **modelu** w oparciu o zbiór **danych** wejściowych (treningowych)
- Model odnosi się do sposobu przetwarzania danych (algorytmu) z kompletnym zestawem parametrów, dobranych zwykle w procesie uczenia. Jego postacią może być wzór matematyczny lub określona struktura systemu

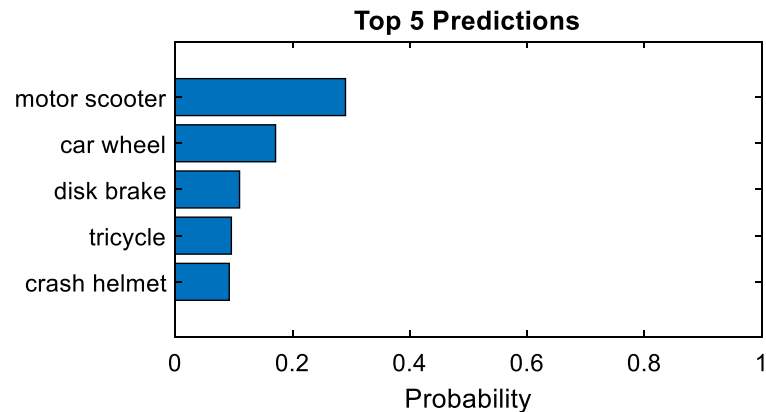
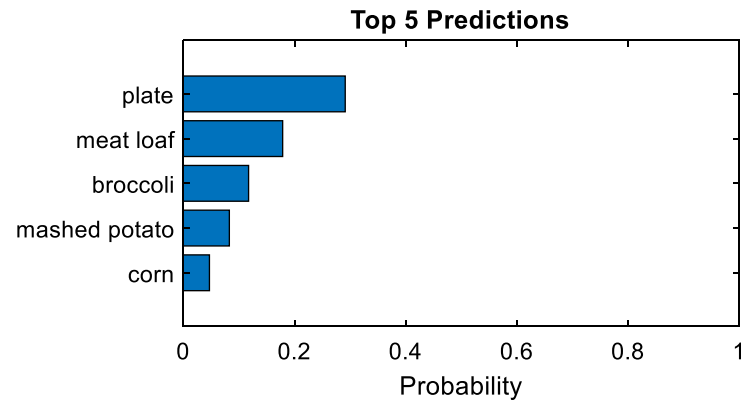
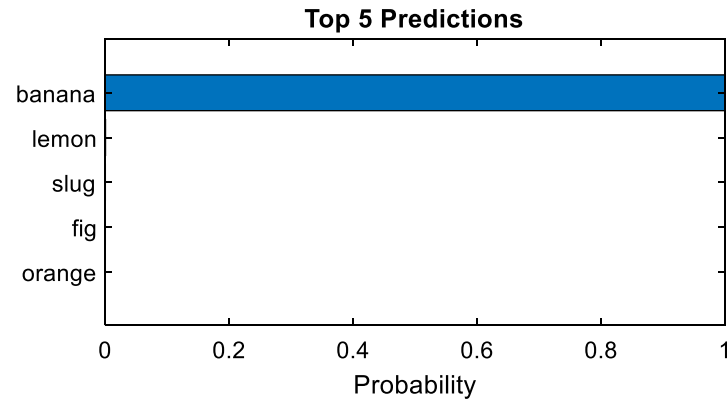


Uczenie maszynowe (ang. Machine Learning)

- Przykładowe modele:
 - prawo powszechnego ciążenia
 - wycena akcji spółek giełdowych
 - prognoza pogody
- Uczenie maszynowe stosuje się w przypadku wyznaczenia modeli, których analityczna postać jest trudna lub nawet niemożliwa do wyznaczenia (np. Spotify, Snapchat, LinkedIn)
- Jakość modelu uzyskanego za pomocą techniki uczenia maszynowego zależy od danych treningowych

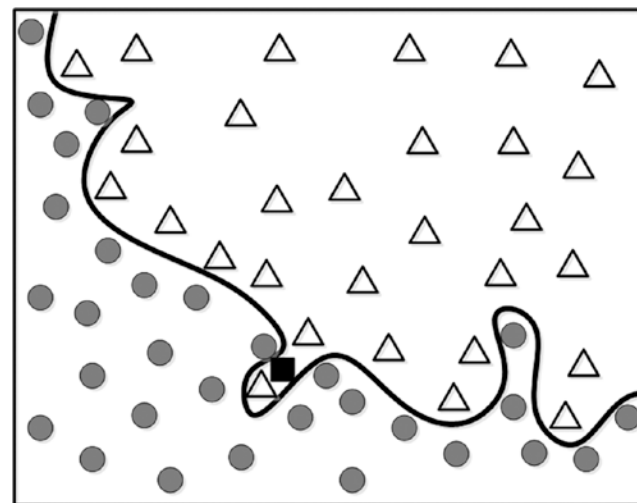
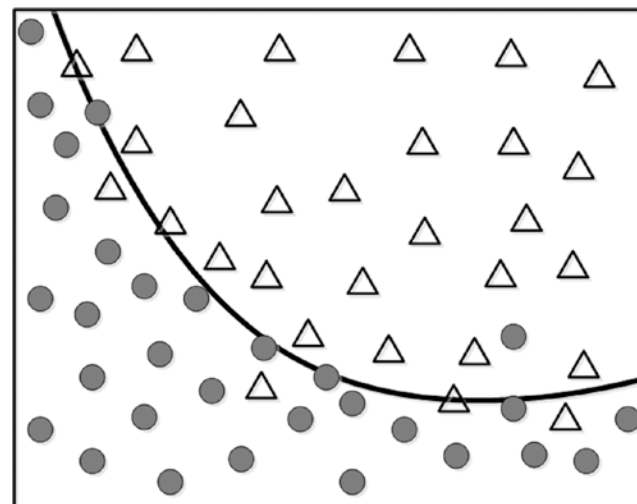


Klasyfikacja obrazów za pomocą sieci neuronowej



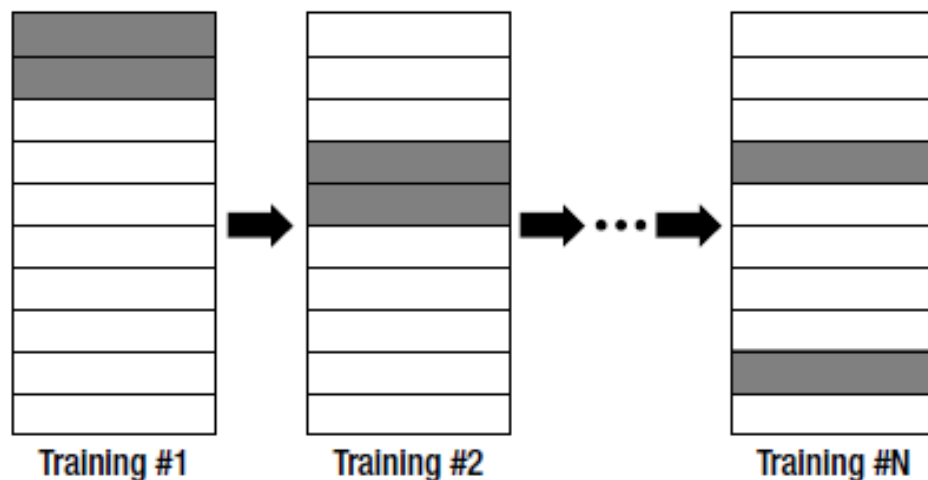
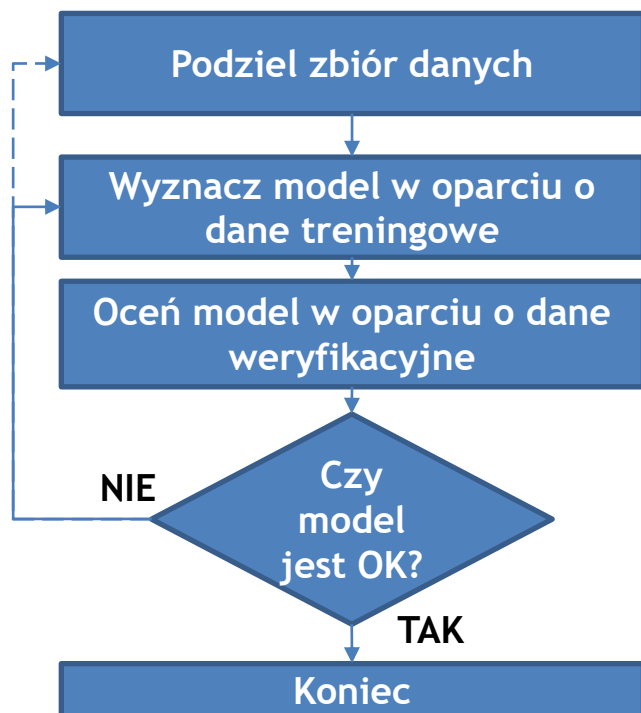
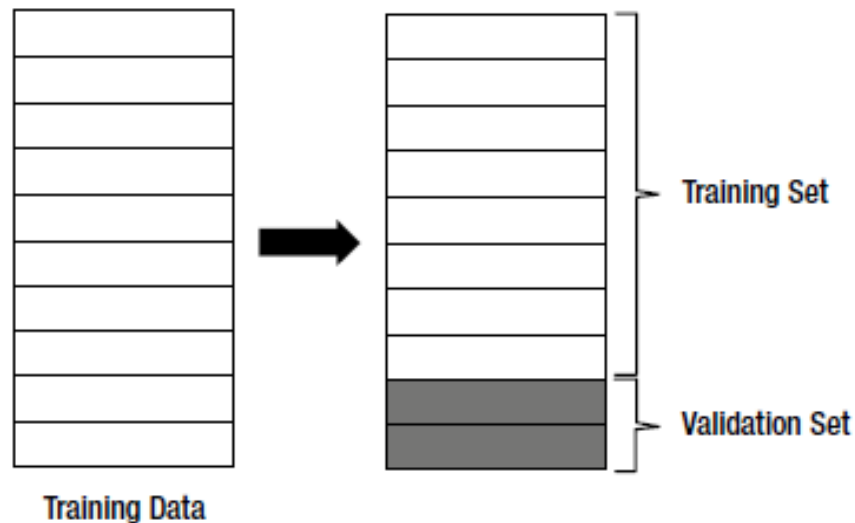
Przetrenowanie

- Celem uczenia maszynowego jest wyznaczenie modelu wiernie odpowiadającego danym treningowym
- Jednakże model, który ze 100% skutecznością klasyfikuje dane treningowe, może powodować błędy klasyfikacji danych wejściowych
- Dzieje się tak, gdyż dane wejściowe są zniekształcone (zaszumione), co prowadzi do wyznaczenia błędnego modelu
- Niepoprawne założenie o bezbłędności danych treningowych prowadzi do efektu znanego pod pojęciem przetrenowania
- Nie oznacza to jednak, że należy celowo zmniejszyć dokładność modelu względem danych wejściowych
- Dwie podstawowe techniki pozwalające ograniczyć przetrenowanie to: regularyzacja i walidacja



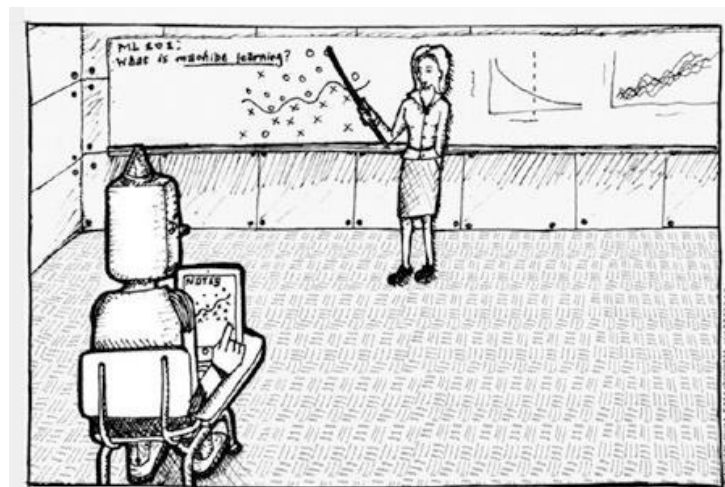
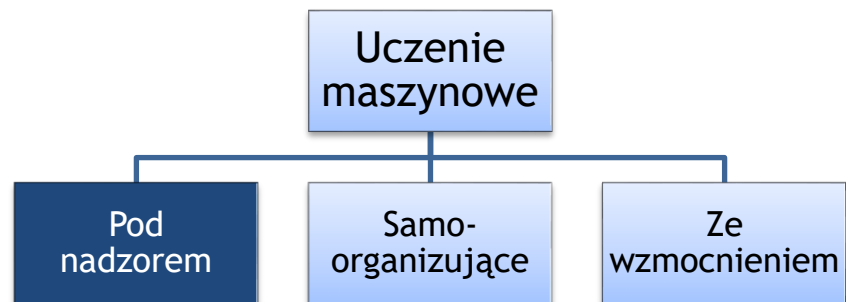
Przetrenowanie

- Regularyzacja polega na uproszczeniu modelu
- Walidacja - proces określania jakości modelu na podstawie zestawu danych weryfikujących, wydzielonych z danych uczących jako ich pewna częśćka (zwykle około 10% danych uczących)
- Walidacja skrośna (inaczej sprawdzanie krzyżowy)



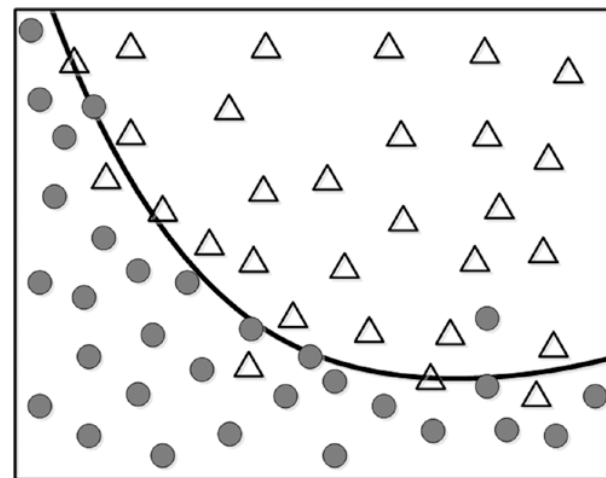
Rodzaje technik uczenia maszynowego

- Uczenie - proces maszynowego (automatycznego lub półautomatycznego) doboru wartości parametrów modelu na podstawie zbioru danych uczących (treningowych)
- Uczenie się z nauczycielem (pod nadzorem) - proces budowania modelu eksploracji na podstawie zbioru uczącego składającego się z par danych (x, d) , gdzie x oznacza wektor danych wejściowych, a d wektor wartości pożądaných pełniących funkcję nauczyciela. Zestaw treningowy musi zawierać dane wejściowe oraz oczekiwane dane wyjściowe.
- Uczenie się bez nauczyciela (samoorganizujące) - proces budowania modelu na podstawie jedynie zbioru atrybutów wejściowych opisanych za pomocą wektorów x .
- Uczenie się ze wzmocnieniem ma na celu uczenie się wiedzy proceduralnej (umiejętności). Jest to obliczeniowe podejście do uczenia się celowego zachowania na podstawie interakcji. Źródłem informacji trenującej, która ma charakter wartościujący jakość działania ucznia, jest raczej krytyk niż nauczyciel.

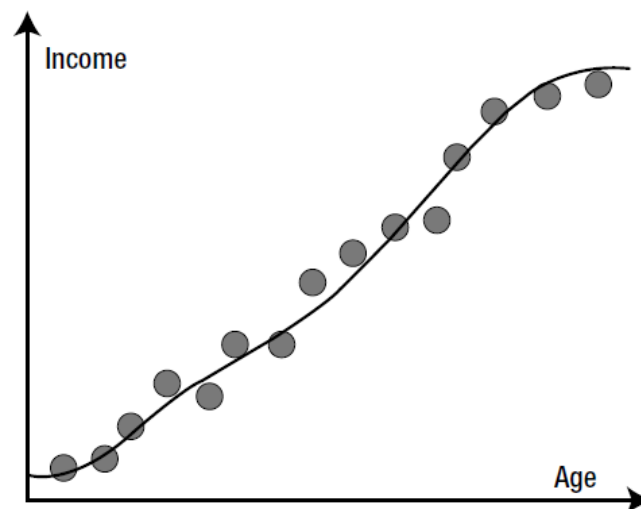


Zastosowanie uczenia się z nadzorem

- **Klasyfikacja** - proces regresji, w którym na podstawie analizy i przetwarzania atrybutów wejściowych generowana jest etykieta klasy, np. klasyfikacja poczty (aplikacja antyspamowa), klasyfikacja liter i cyfr na zdjęciu (OCR). Daną wyjściową jest klasa.



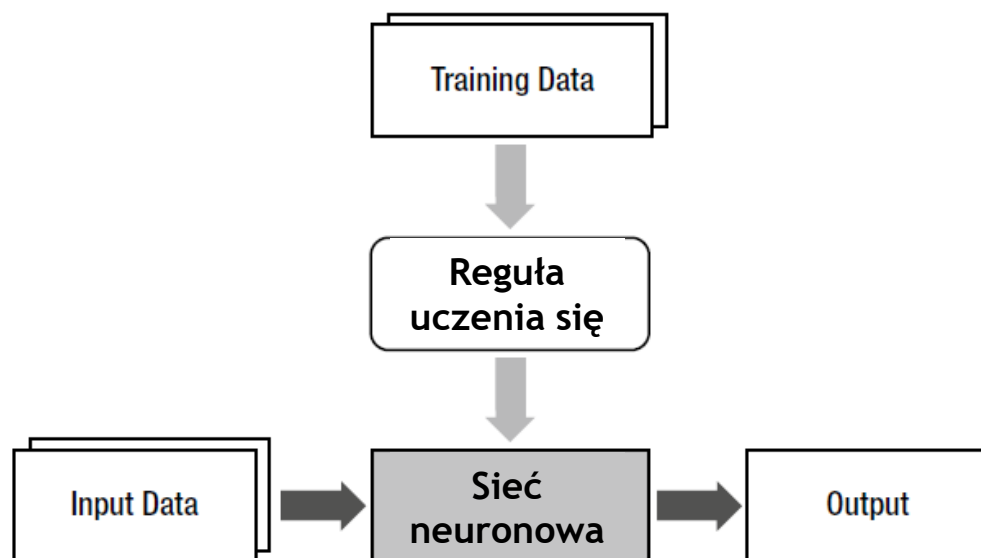
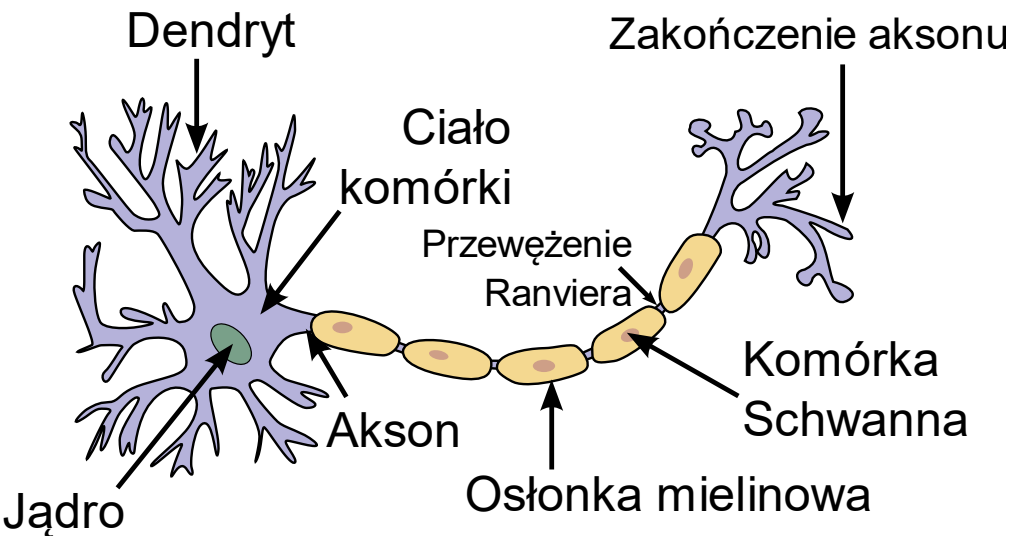
- **Regresja** - proces określania odpowiedzi systemu na pobudzenie w postaci zadanego zbioru atrybutów. W ogólnym rozumieniu matematycznym regresja obejmuje zarówno problem klasyfikacyjny, jak i aproksymacyjny dla dowolnej funkcji. Najczęściej jednak problem regresji jest zawężany do zagadnienia aproksymacji, pozostawiając klasyfikację jako oddzielny problem obliczeniowy. Daną wyjściową jest wartość.



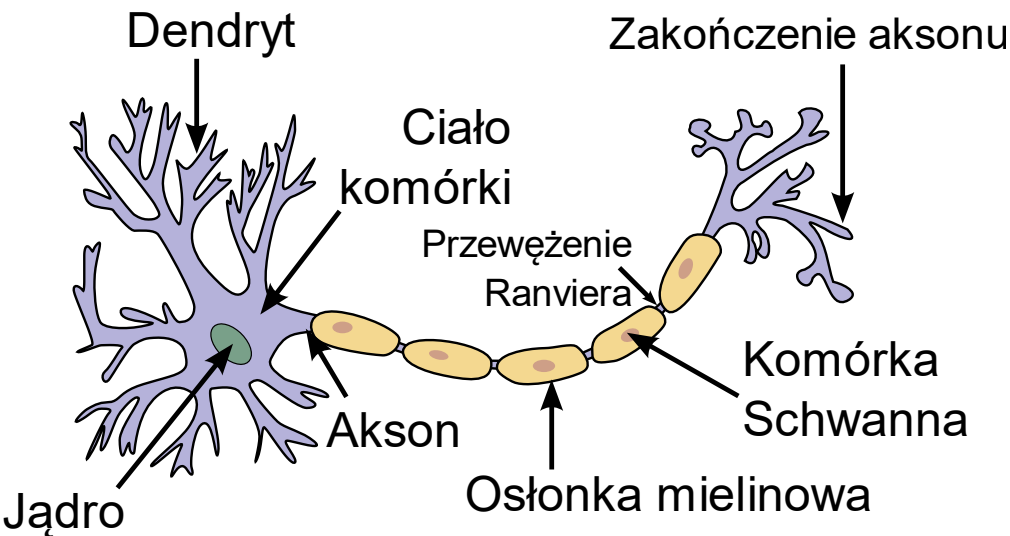
Algorytmy Ewolucyjne

Klasyfikatory - Sieci Neuronowe

Neuron a sztuczna sieć neuronowa (perceptron)



Neuron a sztuczna sieć neuronowa (perceptron)



Wektor wejściowy

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Wektor współczynników wagowych

$$\mathbf{w} = [w_{11} \quad w_{12} \quad w_{13}]$$

Składnik stały

$$b_1$$

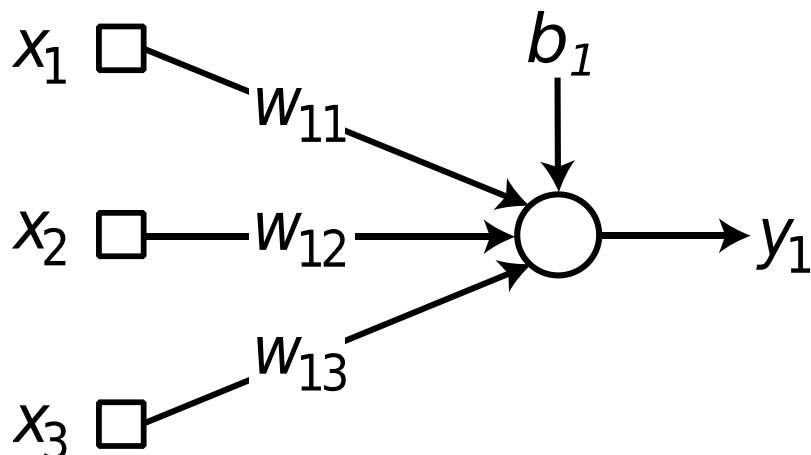
Suma ważona sygnałów

$$v = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$\mathbf{v} = \mathbf{w}\mathbf{x} + \mathbf{b}$$

Sygnał wyjściowy i funkcja aktywacji

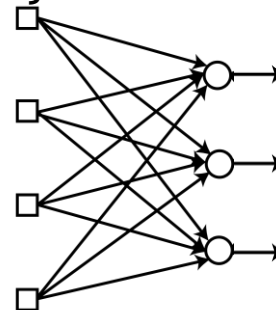
$$\mathbf{Y} = \varphi(\mathbf{v}) = \varphi(\mathbf{w}\mathbf{x} + \mathbf{b})$$



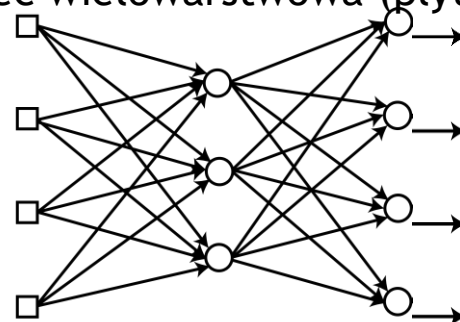
Warstwy sieci neuronowej

- Sieć neuronowa to sieć węzłów
- Warstwa wejściowa
 - odpowiada za przekazywanie sygnałów wejściowych do kolejnych węzłów
 - nie odpowiada za obliczanie sum ważonych oraz funkcji aktywacji
- Warstwa wyjściowa
 - sygnały na wyjściu tej warstwy stanowią odpowiedź sieci neuronowej
- Warstwy ukryte
 - umieszczone pomiędzy warstwą wejściową a warstwą wyjściową
 - dostęp do tych warstw spoza sieci neuronowej jest niemożliwy
- Klasyfikacja sieci neuronowych
 - jednowarstwowe (1 warstwa wejściowa i 1 warstwa wyjściowa)
 - wielowarstwowe (zawierają dodatkowo warstwy ukryte):
 - płytkie (1 warstwa ukryta)
 - głębokie (przynajmniej 2 warstwy ukryte)
- Obecnie najczęściej wykorzystywanymi | w praktyce są głębokie sieci neuronowe

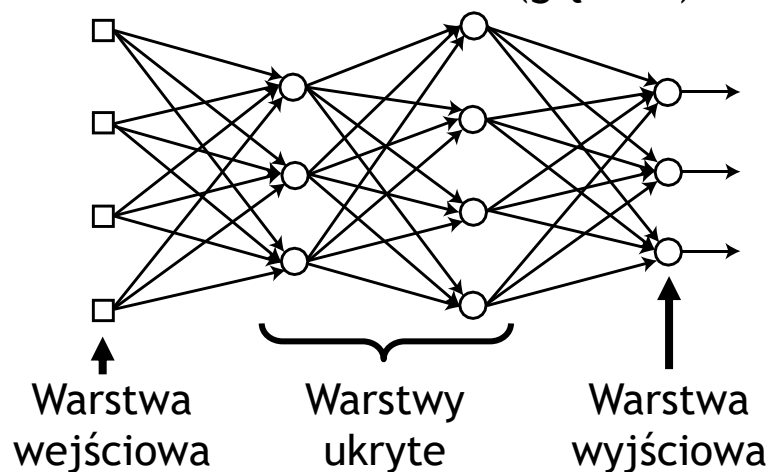
Sieć jednowarstwowa



Sieć wielowarstwowa (płytką)

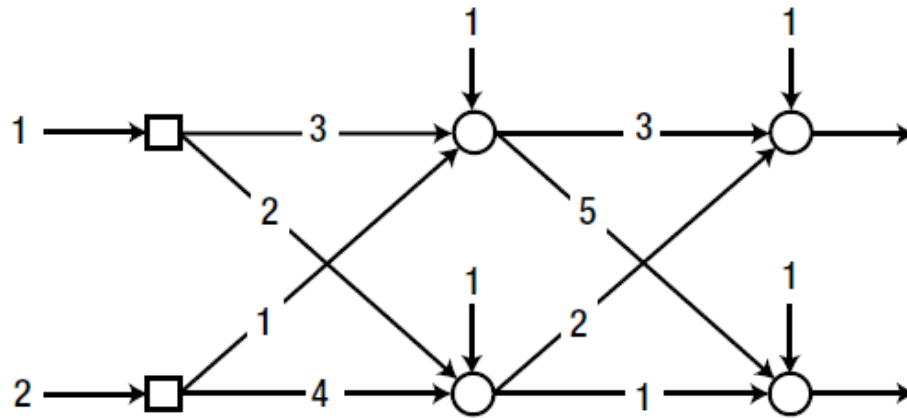


Sieć wielowarstwowa (głęboka)

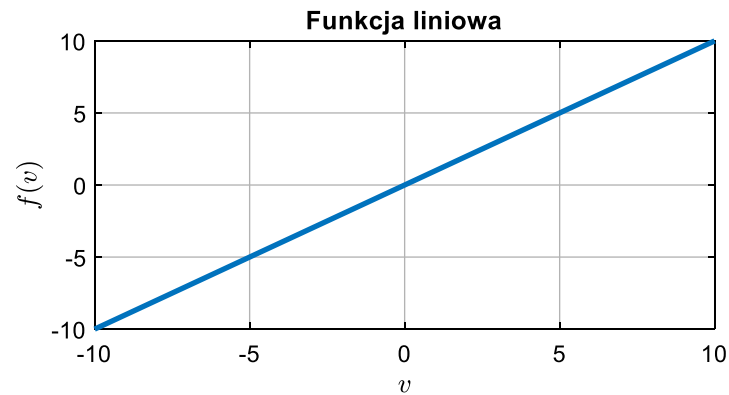


Przykład

Oblicz wartości na wyjściu płytkiej sieci neuronowej

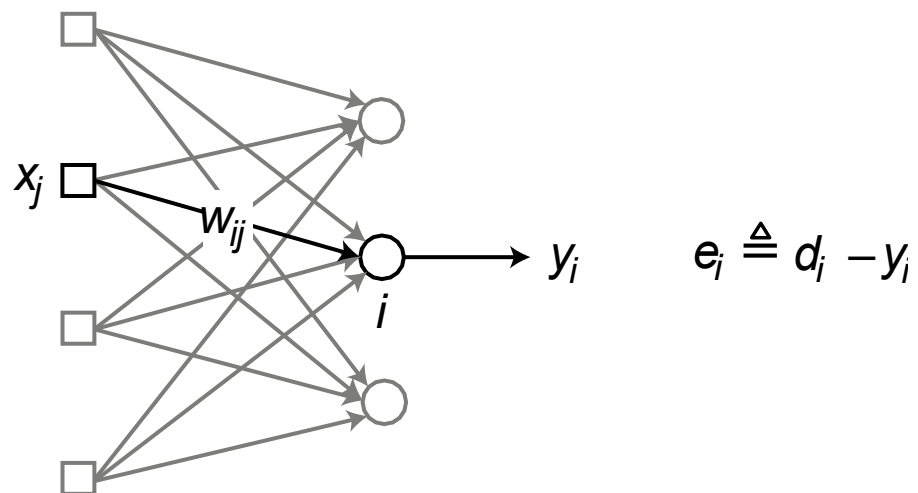
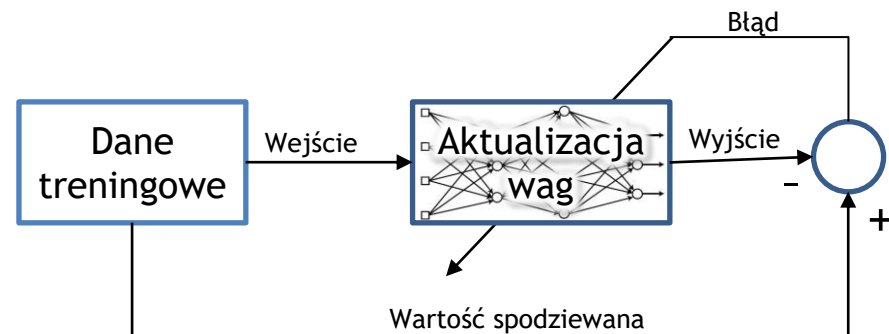


o funkcji aktywacji $\varphi(v) = v$



Nadzorowane uczenie się sieci neuronowej

- Dane treningowe - pary sygnałów wejściowych oraz sygnałów oczekiwanych na wyjściu sieci neuronowej
- Sieć neuronowa przechowuje informację w postaci wartości współczynników wagowych
- Uczenie się sieci polega na zmianie wartości współczynników wagowych zgodnie z założonymi kryteriami
- Uczenie się sieci neuronowej pod nadzorem opiera się na redukcji wartości błędu pomiędzy wartością na wyjściu sieci neuronowej a wartością spodziewaną



Nadzorowane uczenie się sieci neuronowej

1. Inicjalizacja wartości wag sieci neuronowej
2. Przetworzenie elementu ze zbioru danych treningowych przez sieć neuronową
3. Obliczenie różnicy pomiędzy wyjściem sieci neuronowej a wartością spodziewaną

$$e_i = d_i - y_i$$

4. Obliczenie korekty współczynników wagowych

$$\Delta w_{ij} = \alpha e_i x_j$$

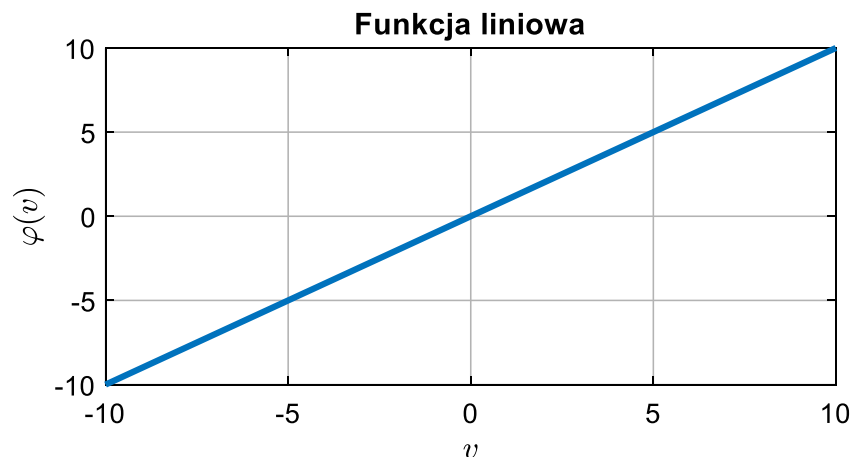
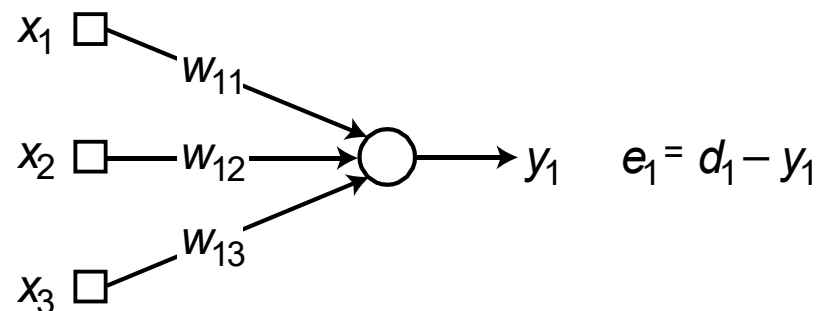
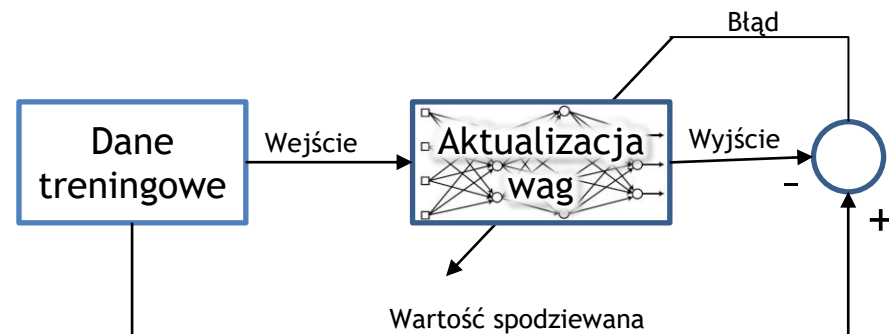
5. Korekta wartości wag w oparciu o wartość błędu

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

6. Powtórzenie kroków 2-5 dla pozostałych danych treningowych

7. Powtórzenie kroków 2-6 do moment uzyskania odpowiednio małej wartości błędu

- x_j - wartość j-tego sygnału wejściowego
- e_i - wartość błędu zmierzona na i-tym wyjściu
- w_{ij} - wartość współczynnika wagowego pomiędzy j-tym węzłem wejściowym a i-tym węzłem wyjściowym
- α - współczynnik uczenia ($0 < \alpha \leq 1$)



Nadzorowane uczenie się metodą „delty”

1. Inicjalizacja wartości wag sieci neuronowej
2. Przetworzenie elementu ze zbioru danych treningowych przez sieć neuronową
3. Obliczenie różnicy pomiędzy wyjściem sieci neuronowej a wartością spodziewaną

$$e_i = d_i - y_i$$

4. Obliczenie korekty współczynników wagowych

$$\Delta w_{ij} = \alpha \delta_i x_j$$

5. Korekta wartości wag w oparciu o wartość błędu

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

6. Powtórzenie kroków 2-5 dla pozostałych danych treningowych

7. Powtórzenie kroków 2-6 do moment uzyskania odpowiednio małej wartości błędu

$$\delta_i = \varphi'(v_i) e_i$$

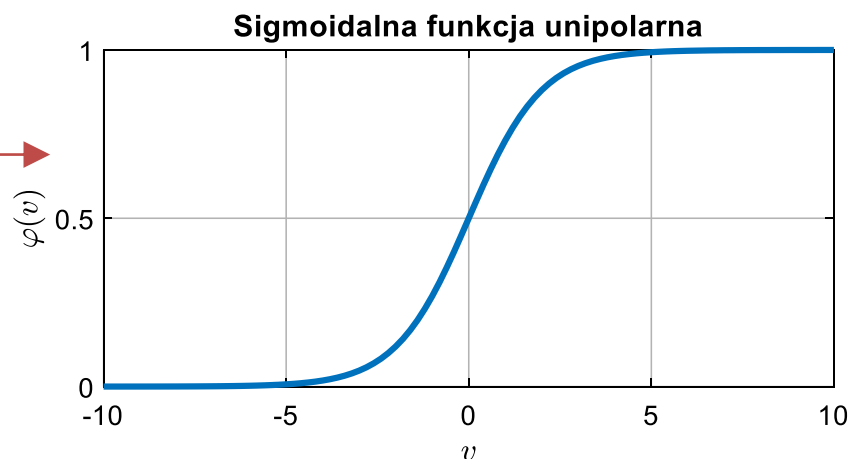
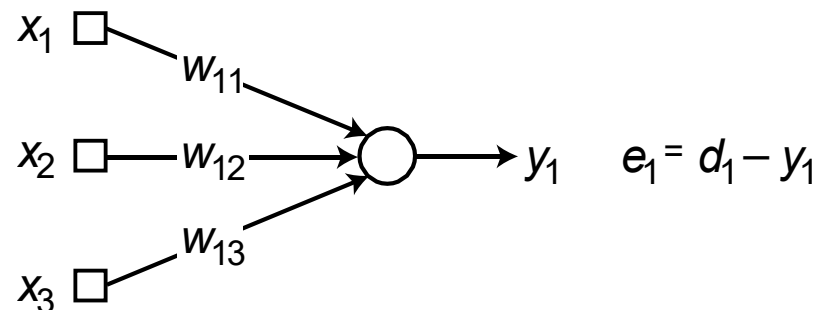
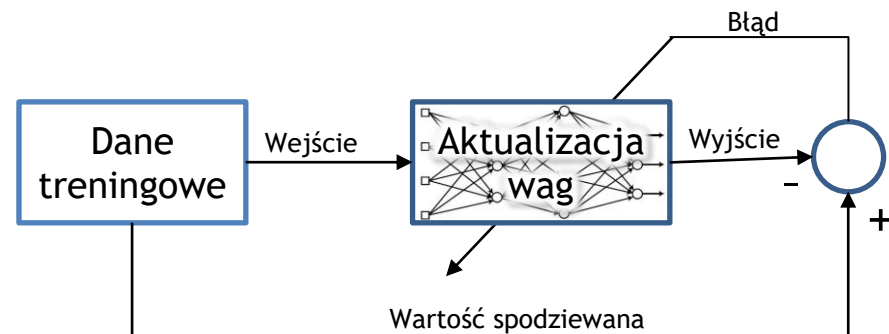
$$\varphi(v_i) = \frac{1}{1 + e^{-v_i}}$$

$$\varphi'(v_i) = \varphi(v_i)(1 - \varphi(v_i))$$

$$\Delta w_{ij} = \alpha \varphi(v_i)(1 - \varphi(v_i)) e_i x_j$$

v_i - suma ważona i-tego węzła wyjściowego

$\varphi'(v_i)$ - pochodna funkcji aktywacji $\varphi(v_i)$



Metody aktualizacji współczynników wagowych

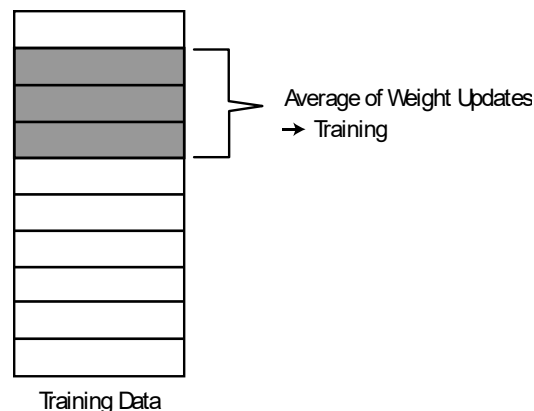
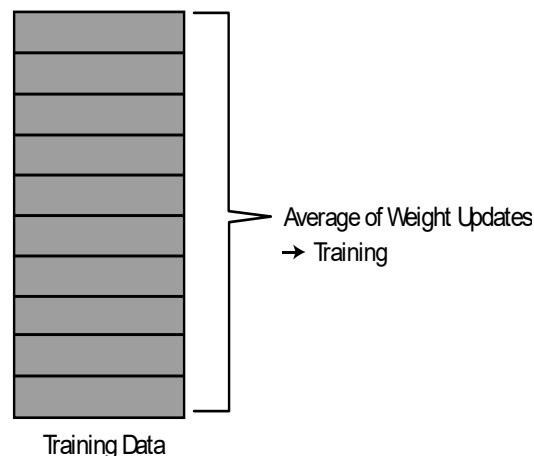
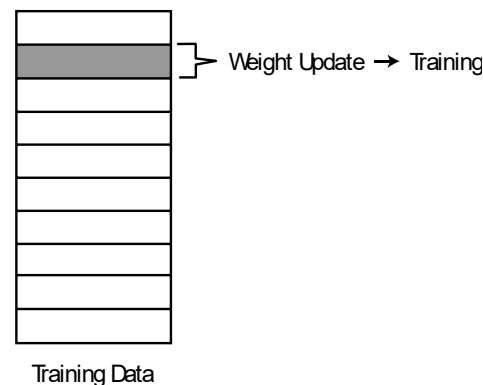
- SGD (ang. *stochastic gradient descent*) - obliczenie wartości błędu a następnie aktualizacja wartości współczynników wagowych zachodzi kolejno dla każdego elementu z zestawu danych treningowych

$$\Delta w_{ij} = \alpha x_j \delta_i$$

- Przetwarzanie wsadowe - współczynnik aktualizacji wag jest średnią ze współczynników aktualizacyjnych wyznaczonych dla całego zestawu danych treningowych o rozmiarze N

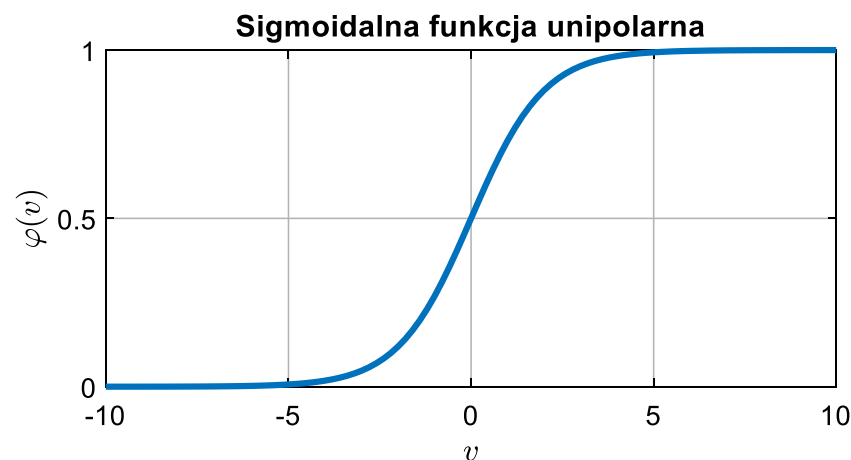
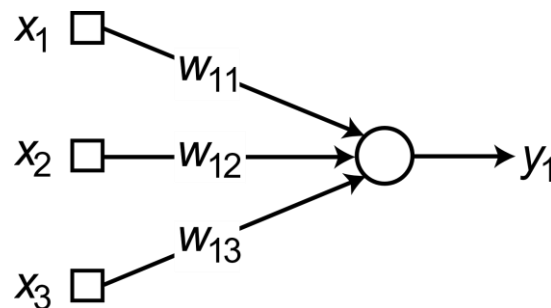
$$\Delta w_{ij} = \frac{1}{N} \sum_{k=1}^N \Delta w_{ij}(k)$$

- Metoda pośrednia - uruchamiana kilkakrotnie dla porcji danych, łączy korzyści obu metod szybkość metody SGD i stabilność metody wsadowej, dlatego jest często stosowana w procesie uczenia się sieci neuronowych



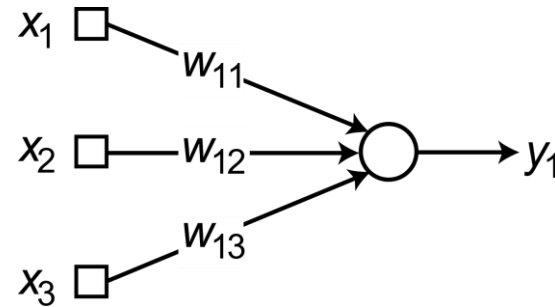
Przykład

```
X=[ 0 0 1;  
    0 1 1;  
    1 0 1;  
    1 1 1;  
];  
D=[ 0  
    0  
    1  
    1  
];  
  
W = 2*rand(1,3) - 1;  
for epoch = 1:10000  
    W = DeltaSGD(W, X, D);  
end  
  
N = size(X,1);  
for k = 1:N  
    x = X(k, :)' ;  
    v = W*x;  
    y = Sigmoid(v);  
    fprintf('%g\n',y)  
end
```



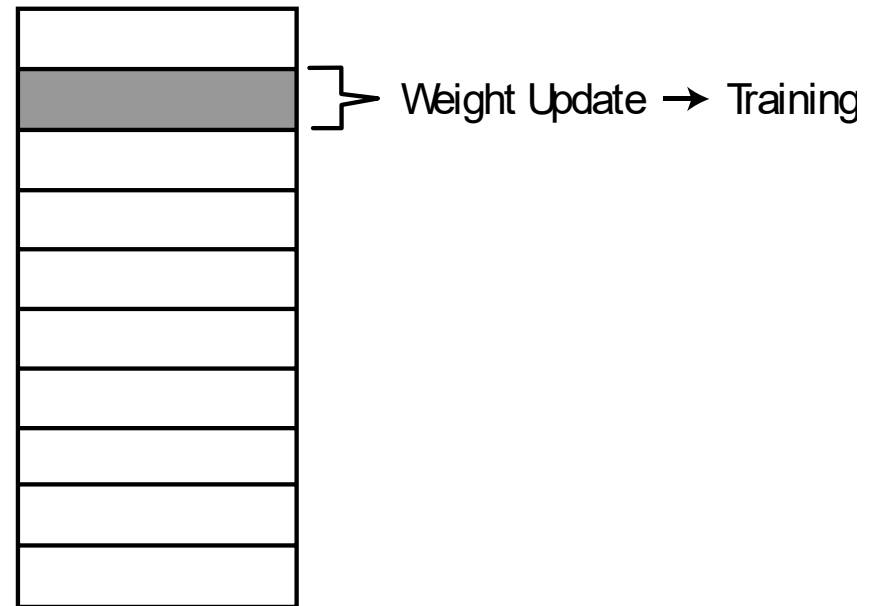
DeltaSGD(W, X, D)

```
function W = DeltaSGD(W, X, D)
alpha = 0.9;
N = 4;
for k = 1:N
    x = X(k, :)';
    d = D(k);
    v = W*x;
    y = Sigmoid(v);
    e = d - y;
    delta = y*(1-y)*e;
    dw = alpha*delta*x;
    w(1) = w(1) + dw(1);
    w(2) = w(2) + dw(2);
    w(3) = w(3) + dw(3);
end
end
```



$$\delta_i = \varphi(v_i)(1 - \varphi(v_i))e_i$$

$$\Delta w_{ij} = \alpha \delta_i x_j$$



Training Data

Przykład

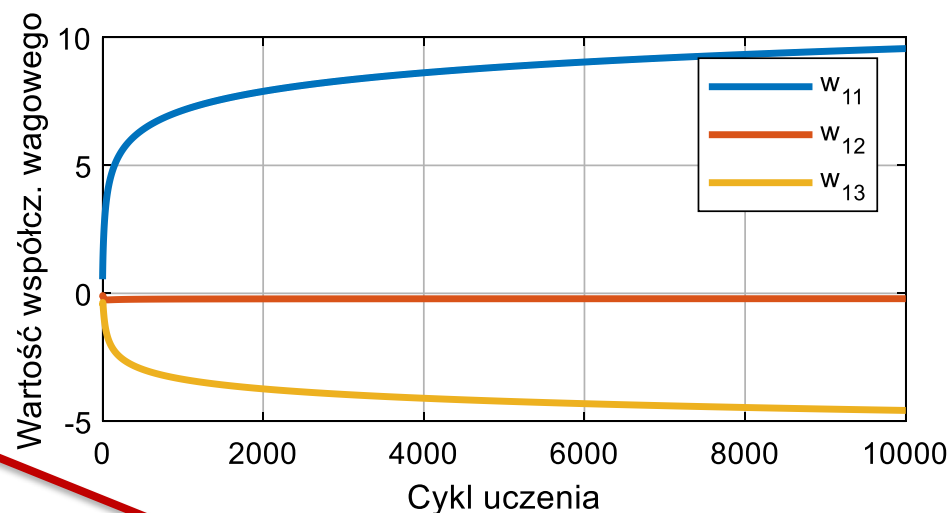
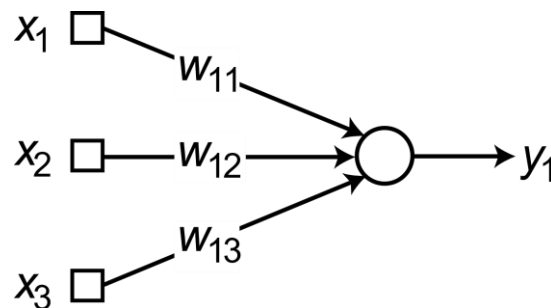
```
X=[ 0 0 1;  
    0 1 1;  
    1 0 1;  
    1 1 1;
```

```
];
```

```
D=[ 0  
    0  
    1  
    1  
];
```

```
W = 2*rand(1,3) - 1;  
for cykl = 1:10000  
    W = DeltaSGD(W, X, D);  
end
```

```
N = size(X,1);  
for k = 1:N  
    x = X(k, :);  
    v = W*x;  
    y = Sigmoid(v);  
    fprintf('%g\n',y)  
end
```



```
0.010  
0.008  
0.993  
0.992
```

Przykład 1

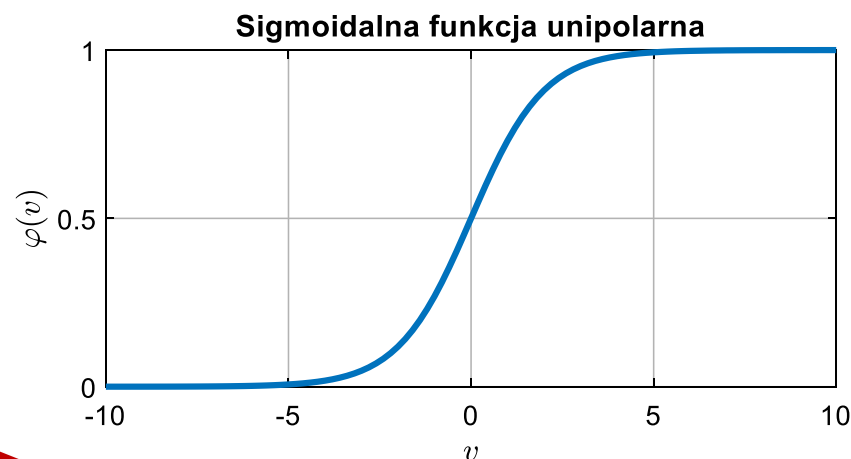
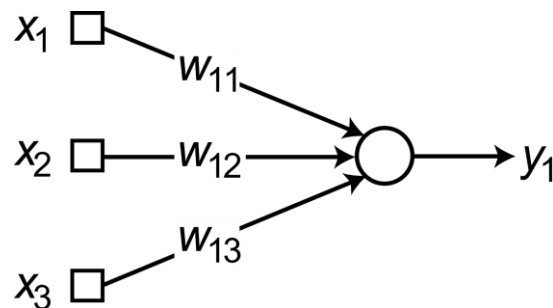
```
X=[ 0 0 1;  
    0 1 1;  
    1 0 1;  
    1 1 1;
```

```
];
```

```
D=[ 0  
    0  
    1  
    1  
];
```

```
W = 2*rand(1,3) - 1;  
for cykl = 1:10000  
    W = DeltaSGD(W, X, D);  
end
```

```
N = size(X,1);  
for k = 1:N  
    x = X(k, :)' ;  
    v = W*x;  
    y = Sigmoid(v);  
    fprintf('%g\n',y)  
end
```



```
0.010  
0.008  
0.993  
0.992
```


Praca domowa 1

- Porównaj szybkość uczenia się metodą SGD oraz metodą wsadową:
 - w MATLAB-ie napisz funkcję `DeltaBatch(W, X, D)` realizującą uczenie się sieci neuronowej metodą wsadową
 - wyniki przedstaw w postaci wykresu