Metody Sztucznej Inteligencji II

Joanna Kołodziejczyk

17 marca 2013

Neuron biologiczny

Neuron

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

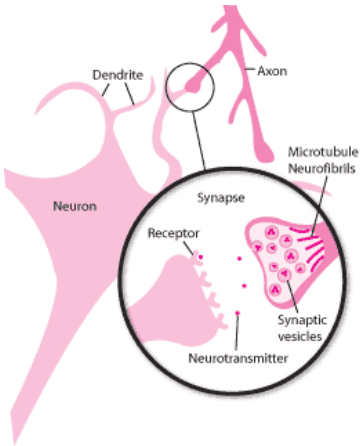
Neuron działanie

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

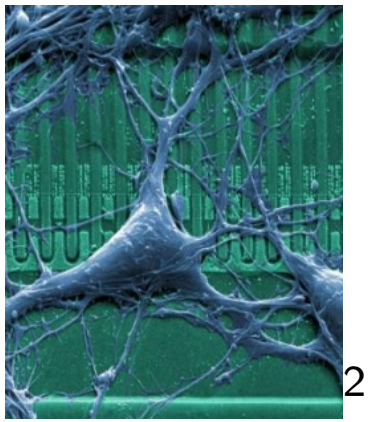
Budowa neuronu biologicznego

Neuron składa się z:

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



1



2

[1] http://www.smart-publications.com/articles/view/choline-uridine-builds-new-

Sztuczna sieć neuronowa

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

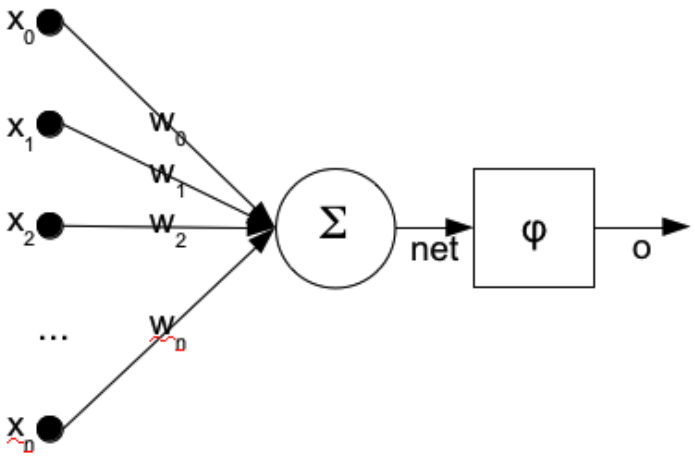
Sztuczna neuron

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

Działanie neuronu

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

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

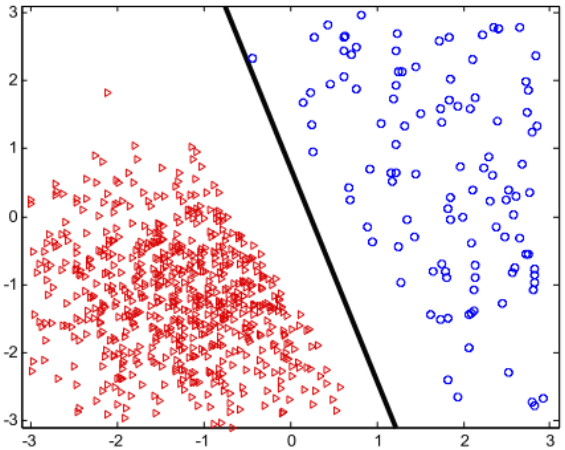


Cechy neuronu

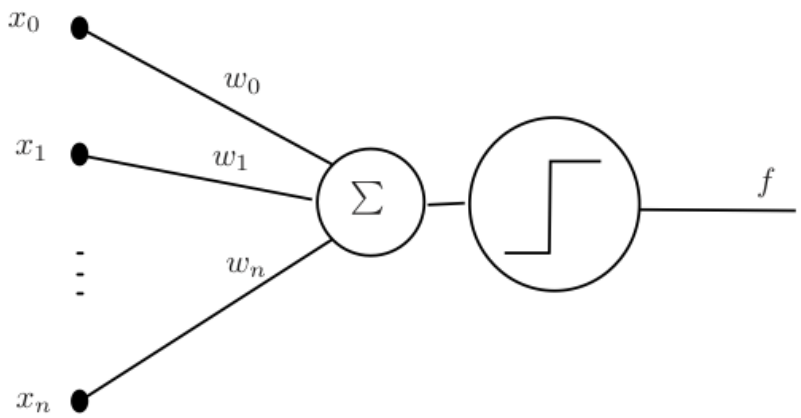
- Wejścia i wagi są liczbami rzeczywistymi dodatnimi i ujemnymi.
- Jeżeli jakaś cecha (wejście) powoduje odpalenie neuronu, waga będzie dodatnia, a jeżeli cecha ma działanie hamujące to waga jest ujemna.
- Funkcję aktywacji dobiera się do rozwiązywanego zadania.
- Wagi są dopasowywane przez pewną regułę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

Perceptron prosty (Rosenblatt 1962)

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



Perceptron — model



Funkcja aktywacji

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

Zadanie klasyfikacji binarnej

Klasyfikacja binarna

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

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

Płaszczyzna separacji klas

Równanie płaszczyzny w przestrzeni $\mathbb{R}^n$

$$w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = 0$$

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

$$\forall_{i, y_i = 1} w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n > 0$$

oraz

$$\forall_{i, y_i = 0} w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \leq 0,$$

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

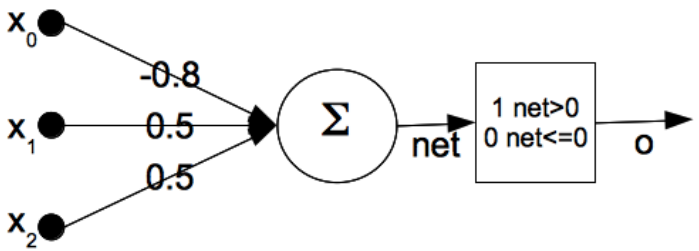
$$\langle \mathbf{w}, \mathbf{x_i} \rangle = \sum_{j=0}^{n} w_j x_{ij}$$

Przykład dwuwymiarowy

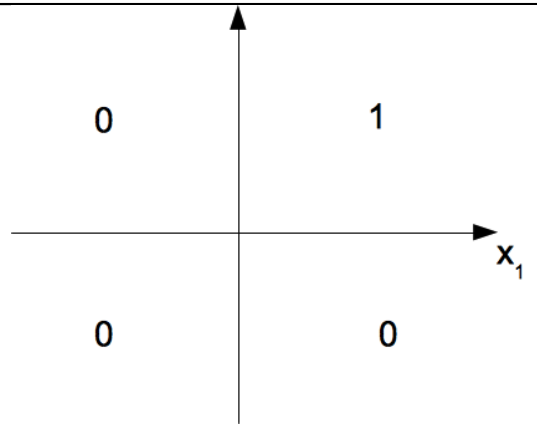
Dane uczące:

	x_1	x_2	y
p_1	1	1	1
p_2	1	0	0
p_3	0	1	0
p_4	0	0	0

Nauczony neuron (ustalone wagi):



Przestrzeń wejść:



Uczenie SSN

Uczenie

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

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

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

Algorytm uczenia perceptronu

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

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

   3. $k = k + 1$

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

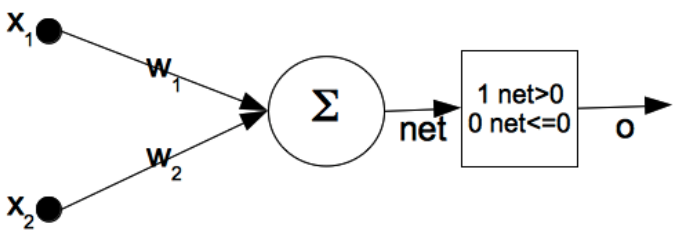
Analiza algorytmu na przykładzie testowym

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

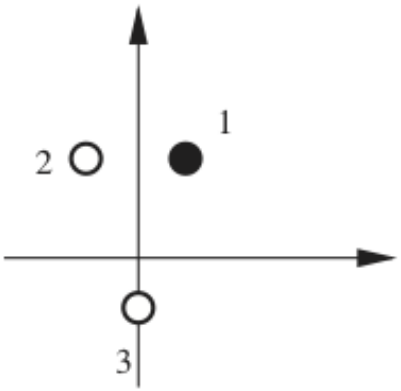
Zbiór uczący

$\{\mathbf{p}_1 = (1, 2), y_1 = 1\}$ $\{\mathbf{p}_2 = (-1, 2), y_2 = 0\}$
$\{\mathbf{p}_3 = (0, -1), y_3 = 0\}$

Architektura sieci:



Przestrzeń wejść:

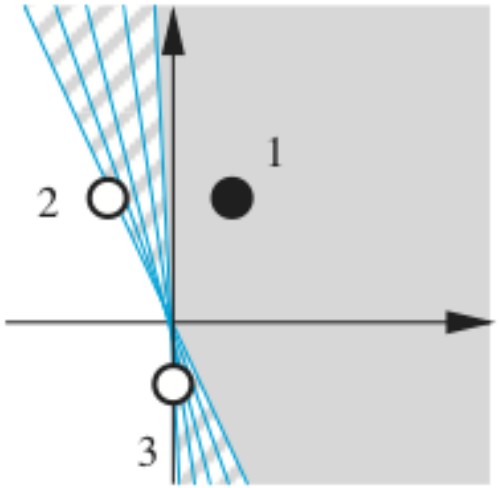


Konsekwencje braku wejścia x_0

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

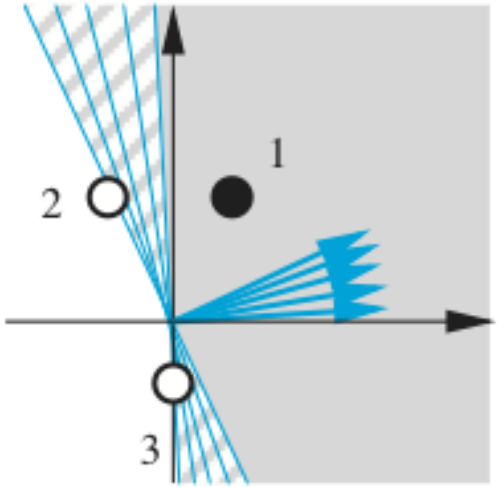
$$w_1 x_1 + w_2 x_2 = 0$$

to równanie prostej przechodzącej przez punkt $(0, 0)$.
Trzeba być pewnym, że taka sieć rozwiąże problem separacji punktu x_1 od punktów x_2 i x_3.
Jak widać rozwiązań jest nieskończenie wiele.



Cel uczenia

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

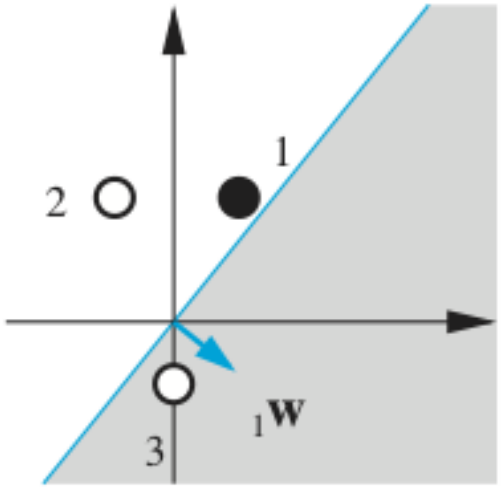


Ustalenie wartości początkowych wag

Zakłada się losowo ustalenie wartości wag początkowych i tak:

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

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



Prezentacja pierwszego punktu ze zbioru uczącego

Pierwszym prezentowanym punktem jest $\mathbf{x}_1$.

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

Wiemy, że $y_1 = 1$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu $\mathbf{x}_1$.
Jak widać na rysunku z poprzedniego slajdu granica decyzyjna jest nieprawidłowo położona w stosunku do punktu uczącego. Powinno się zatem przesunąć wektor wag nieznacznie w kierunku punktu $\mathbf{x}_1$. Można tego dokonać dodając do wektora wag wektor $\mathbf{x}_i$.

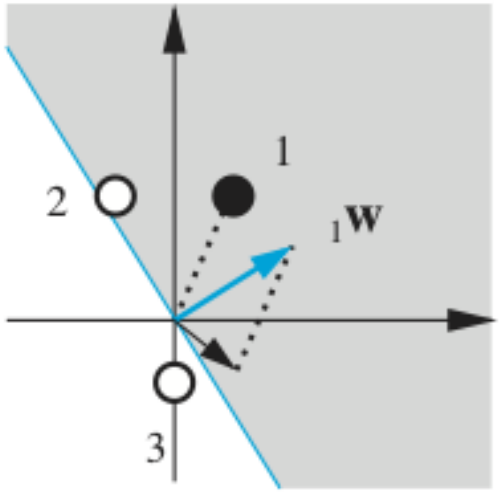
Wnioski co do poprawki wag

Reguła korekty wag (1)

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

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

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



Prezentacja drugiego punktu ze zbioru uczącego

Drugim prezentowanym punktem jest $\mathbf{x}_2$.

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

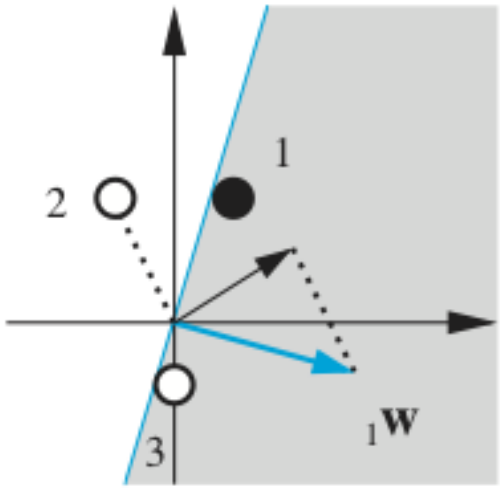
Wiemy, że $y_2 = 0$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu $\mathbf{x}_2$.
Jak widać na rysunku z poprzedniego slajdu granica decyzyjna jest nieprawidłowo położona w stosunku do punktu uczącego. Powinno się zatem odsunąć wektor wag nieznacznie od punktu $\mathbf{x}_2$.
Można tego dokonać odejmując do wektora wag wektor $\mathbf{x}_i$.

Reguła korekty wag (2)

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

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

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

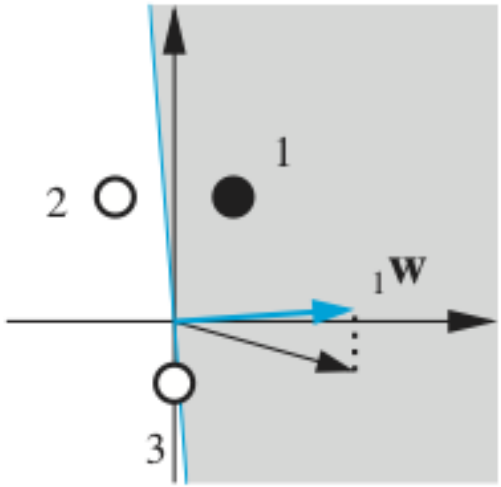


Prezentacja trzeciego punktu ze zbioru uczącego

Trzecim prezentowanym punktem jest $\mathbf{x}_3$.

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

Wiemy, że $y_2 = 0$, zatem sieć popełniła błąd i wskazała złą klasę dla punktu $\mathbf{x}_2$. Stosując regułę 2 poprawia się wagi:
$\mathbf{w}(3) = (3 - 0, (-0.8) - (-1)) = (3, 0.2)$



Kolejna seria prezentacji wzorców uczących

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

Korekty wag

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

Reguła uczenia perceptronu

Potrójna reguła na korektę wag może być zapisana w postaci pojedynczego równania.
Niech błąd popełniany przez sieć neuronową:

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

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

Korekta regułą perceptronu (delta)

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

Perceptron prosty jednowarstowy

Pojedyncza warstwa neuronów (Single layer perceptron) - cechy

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