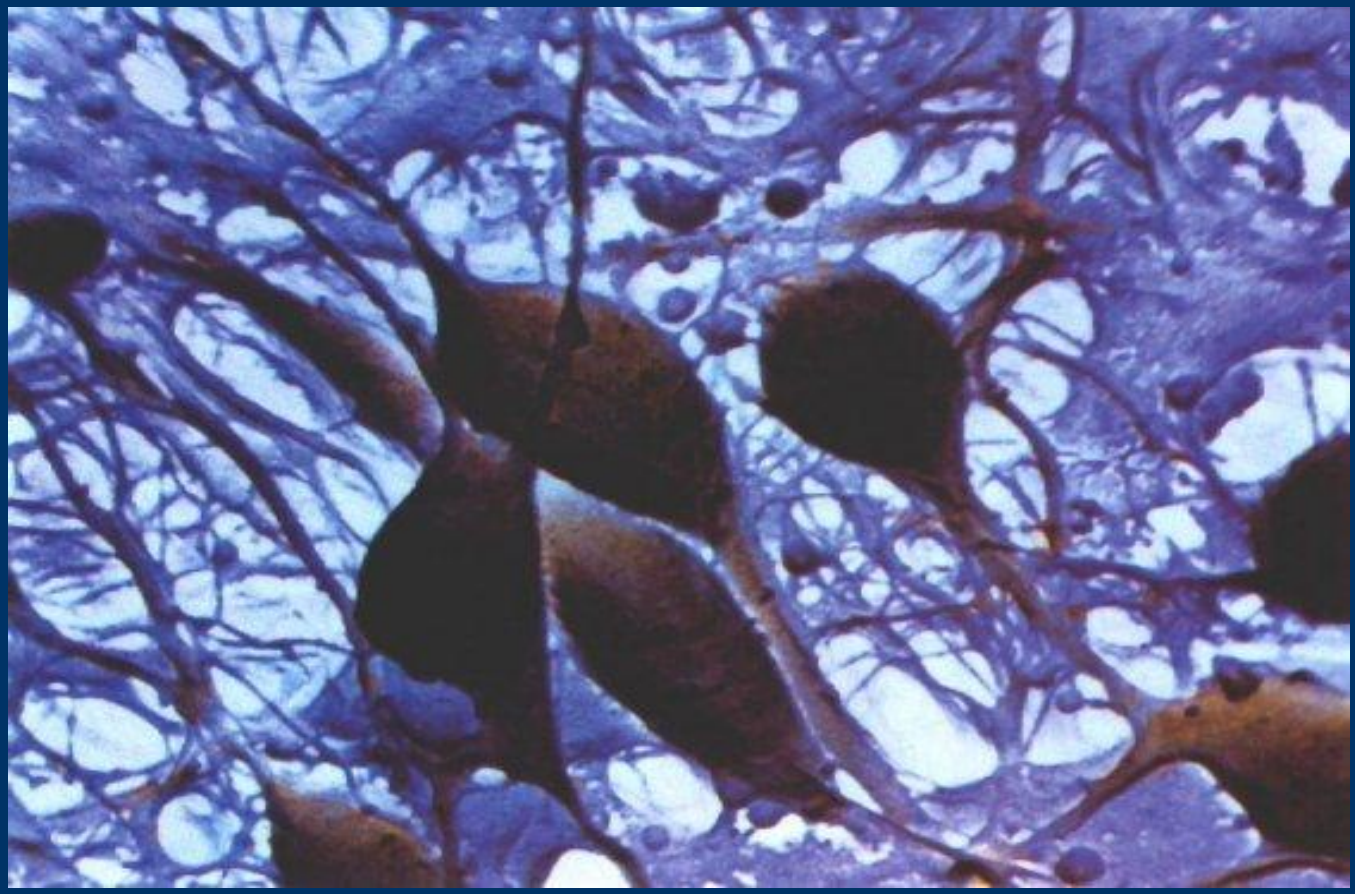
SIECI NEURONOWE

Liniowe i nieliniowe sieci neuronowe

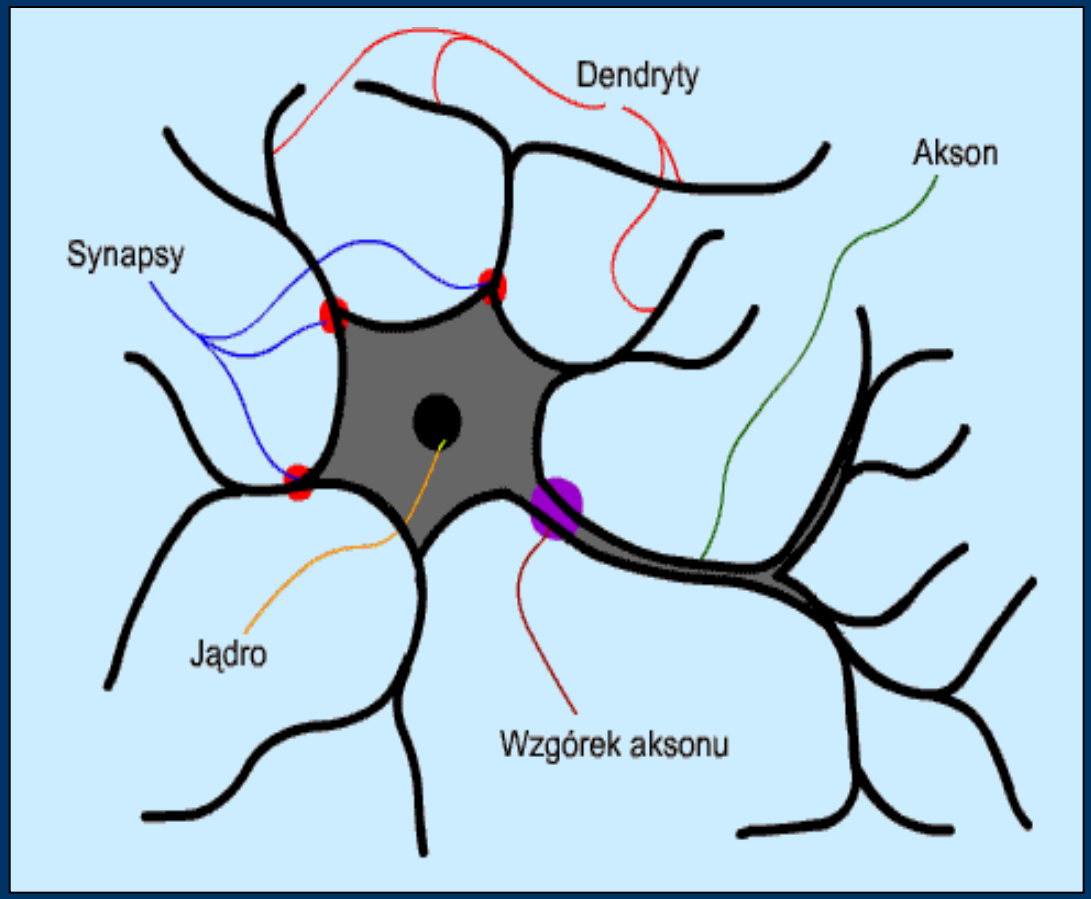


JOANNA GRABSKA-CHRZĄSTOWSKA

Wykłady w dużej mierze przygotowane w oparciu o materiały i pomysły

PROF. RYSZARDA TADEUSIEWICZA

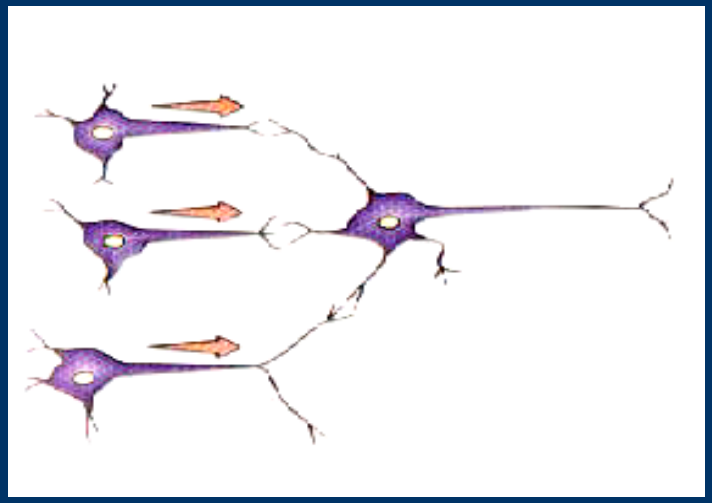
BUDOWA RZECZYWISTEGO NEURONU



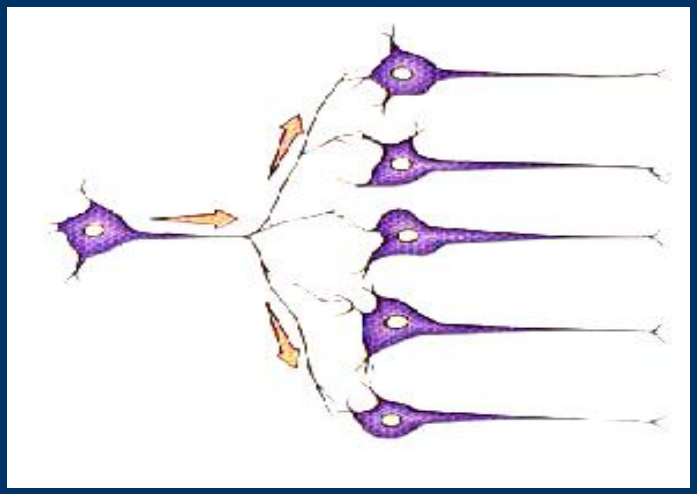
- **Jądro** - "centrum obliczeniowe" neuronu. Tutaj zachodzą procesy kluczowe dla funkcjonowania neuronu.
- **Akson** - "wyjście" neuronu. Neuron ma tylko jeden akson.
- **Wzgórek aksonu** – sumowanie przychodzących sygnałów i generowanie potencjału czynnościowego, który wędruje dalej poprzez akson.

- **Dendryt** - "wejście" neuronu. Dendrytów może być wiele - biologiczne neurony mają ich tysiące.
- **Synapsa** - jeśli dendryt jest wejściem neuronu, to synapsa jest jego furtką. Ma wpływ na moc sygnału napływającego poprzez akson.

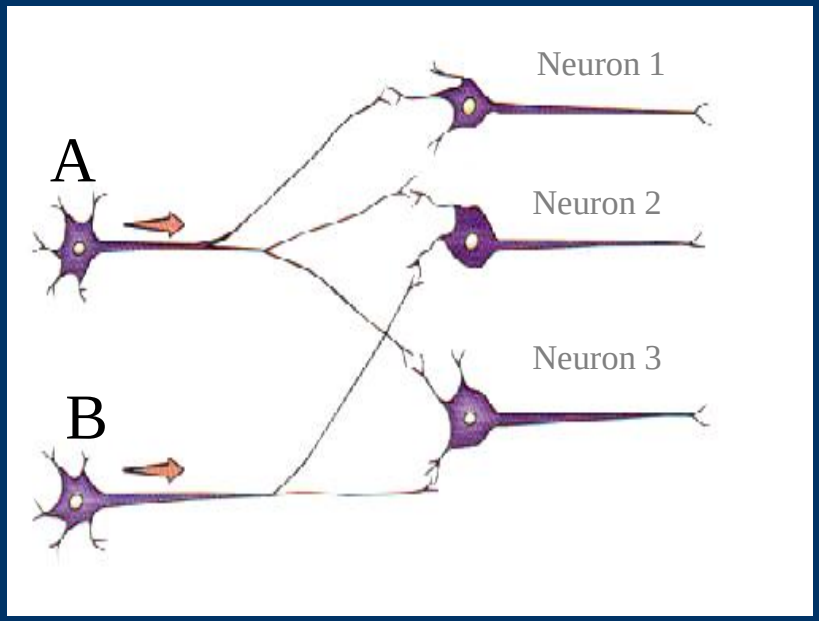
NATURALNE STRUKTURY NEURONOWE



Konwergencja - pobudzenie z wielu włókien dociera do jednego neuronu

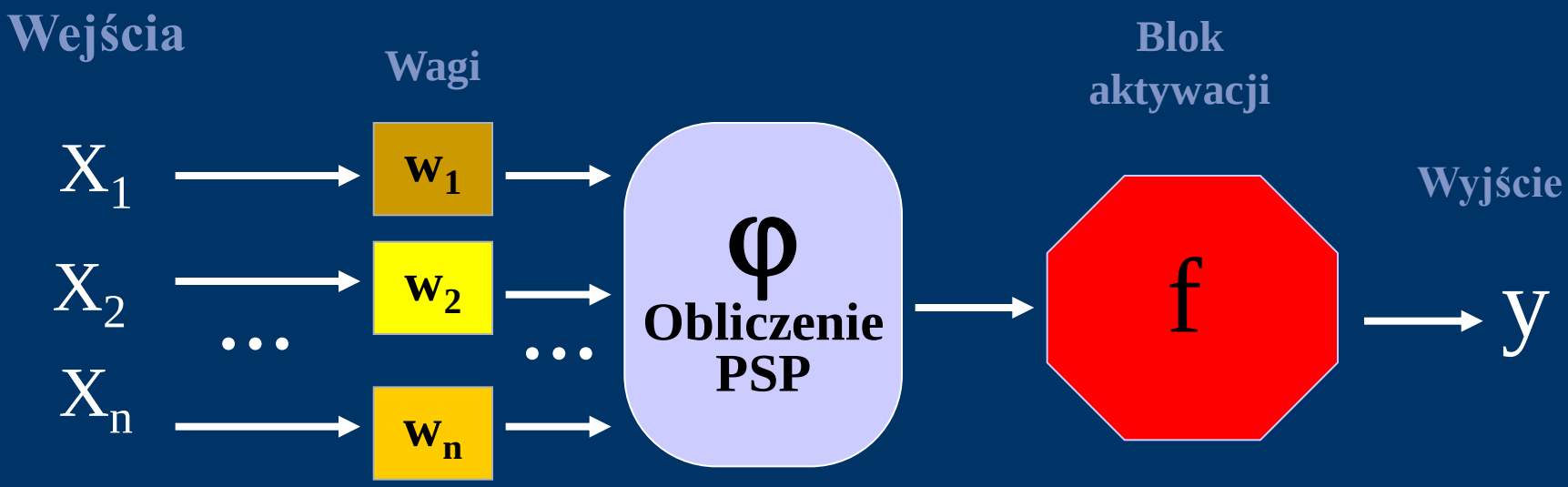


Dywergencja - pobudzenie z jednego neuronu dociera do szeregu innych neuronów.



Pobudzenie docierające z neuronów A i B nie wywołuje pobudzenia neuronów 2 lub 3. Dochodzi jednak do częściowej depolaryzacji błony komórkowej w neuronach 2 i 3 bliskiej progowej wartości potrzebnej do wzbudzenia. Kolejne pobudzenie docierające z neuronów bez trudu wzbudza potencjał czynnościowy.

OGÓLNA DEFINICJA NEURONU

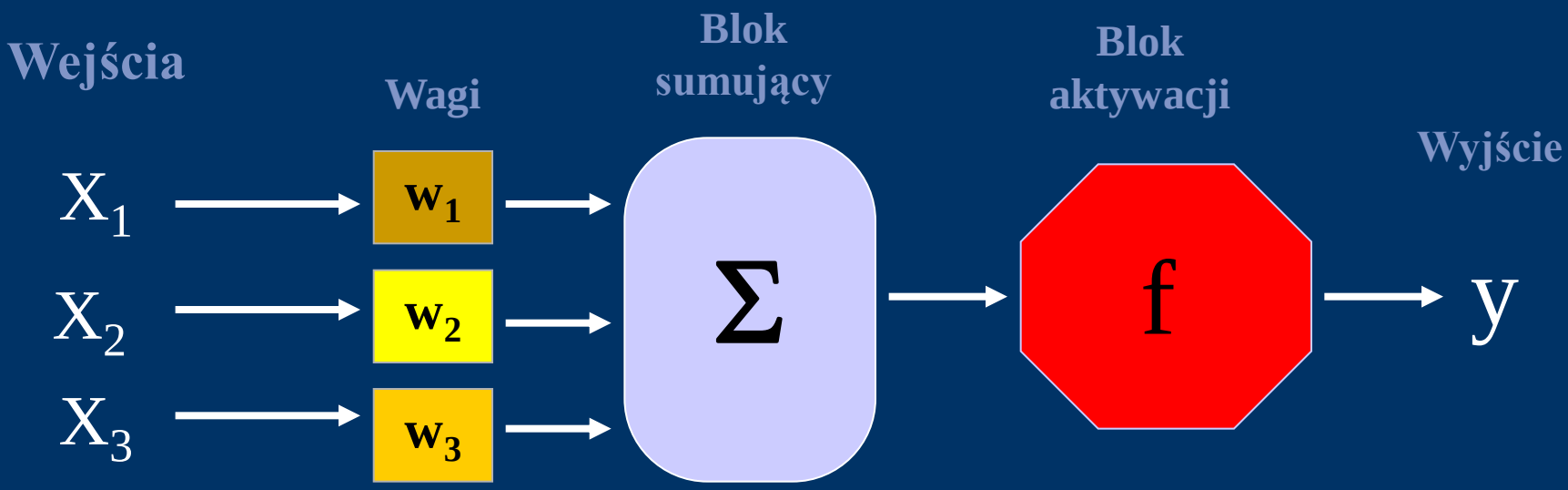


$$y = f\ (\ \varphi\ (\ x_i * w_i\ ))$$

PSP – Post-Synaptic Potential
(Potencjał Postsynaptyczny)

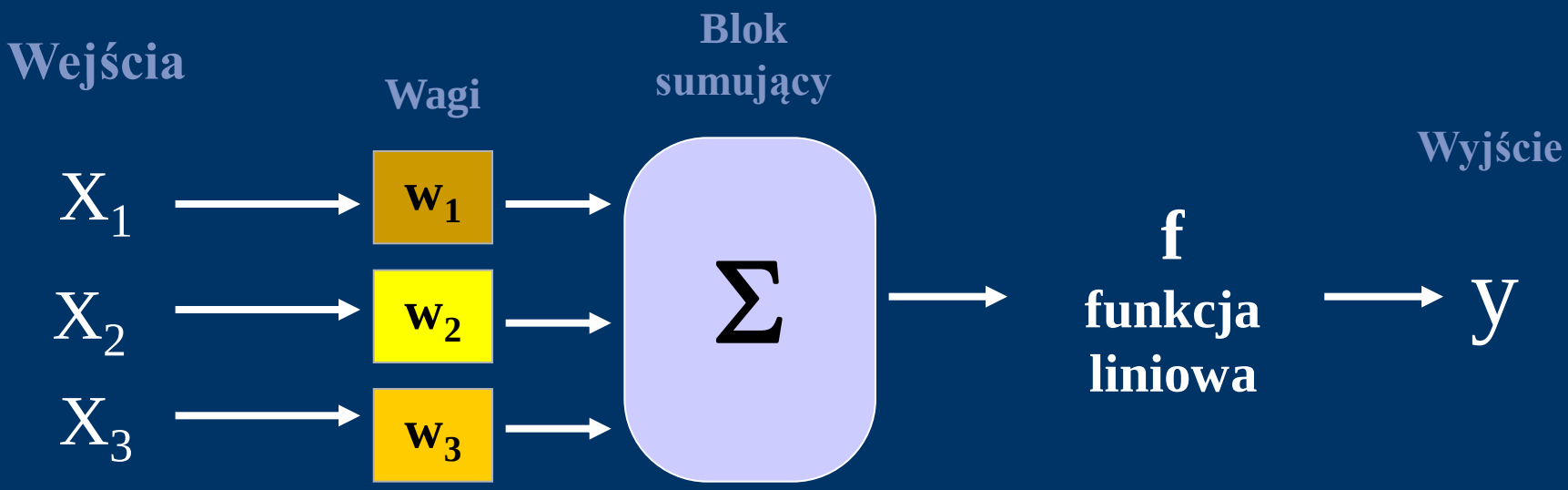
f - funkcja aktywacji

DEFINICJA NEURONU



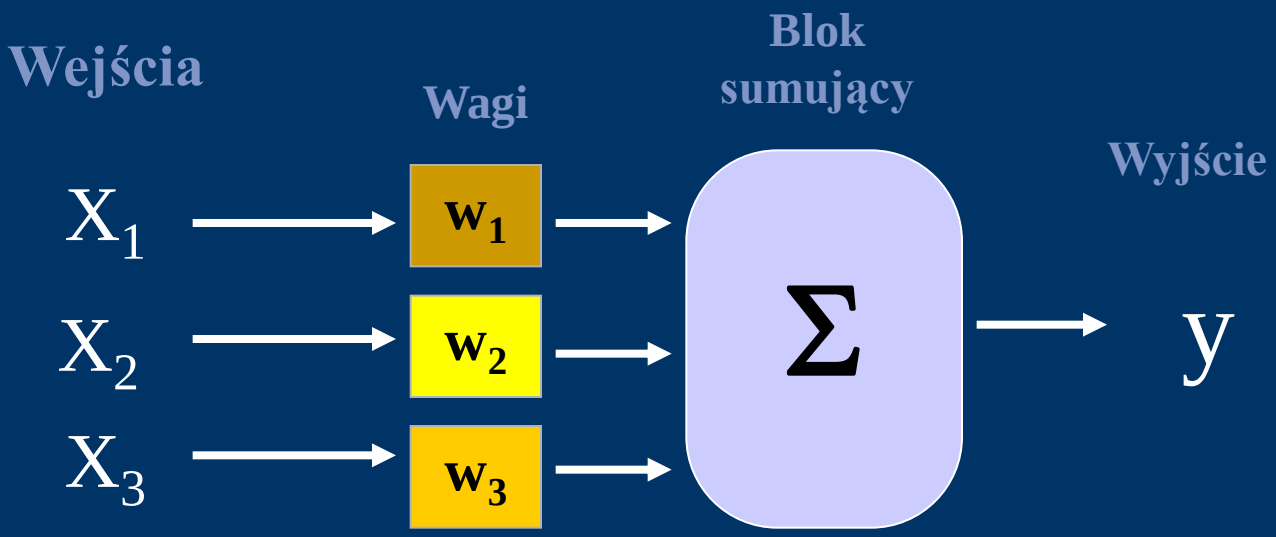
$$y = f\ (\ \Sigma\ (\ x_i * w_i\ ))$$

NEURON LINIOWY



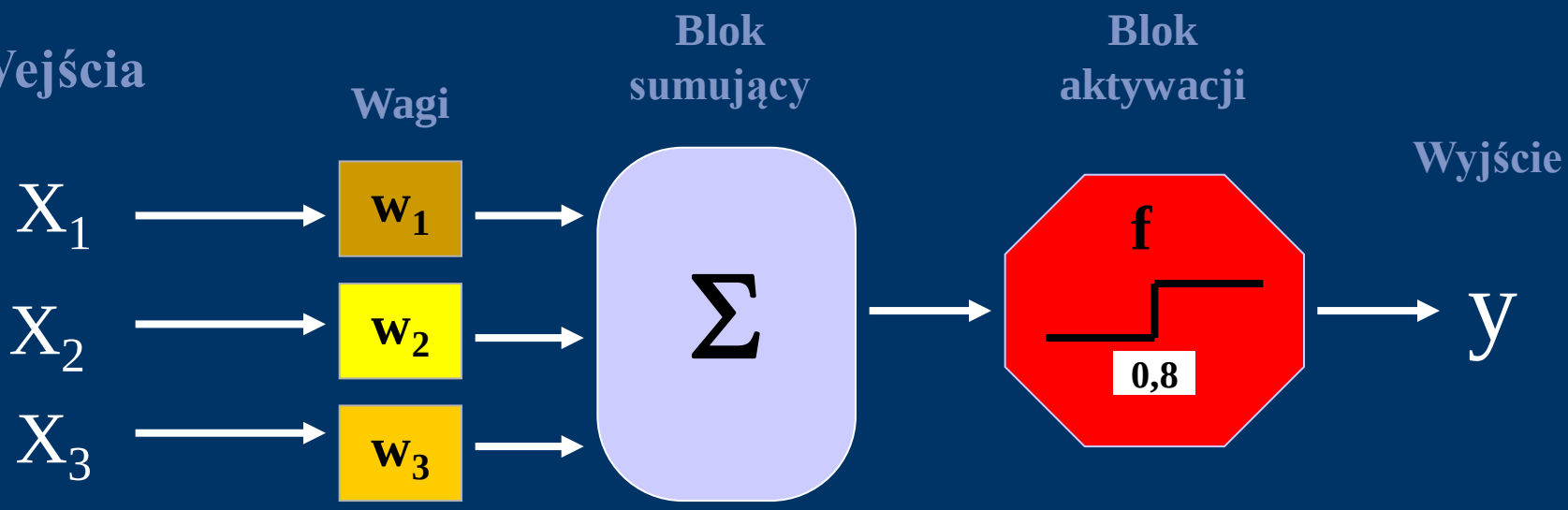
$$y = \Sigma (x_i * w_i)$$

NEURON LINIOWY



$$y = \Sigma (x_i * w_i)$$

ZASADA DZIAŁANIA NEURONU Z FUNKCJĄ SKOKOWĄ (NIELINIOWEGO)



$$y = f\ (\ \Sigma\ (\ x_i * w_i\ ))$$

PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ

Trzy neurony zmysłowe (siatkówki oka żaby) kodują trzy różne reakcje organizmu:

- **odskoczenie (wyprostowanie mięśnia uda)**
- **zjadanie (poruszanie mięśnia języka)**
- **trawienie (wydzielanie soku żołądkowego).**

Wzorce muszą zostać rozpoznane tzn.

kiedy z siatkówki oka do mózgu przesyłany jest :

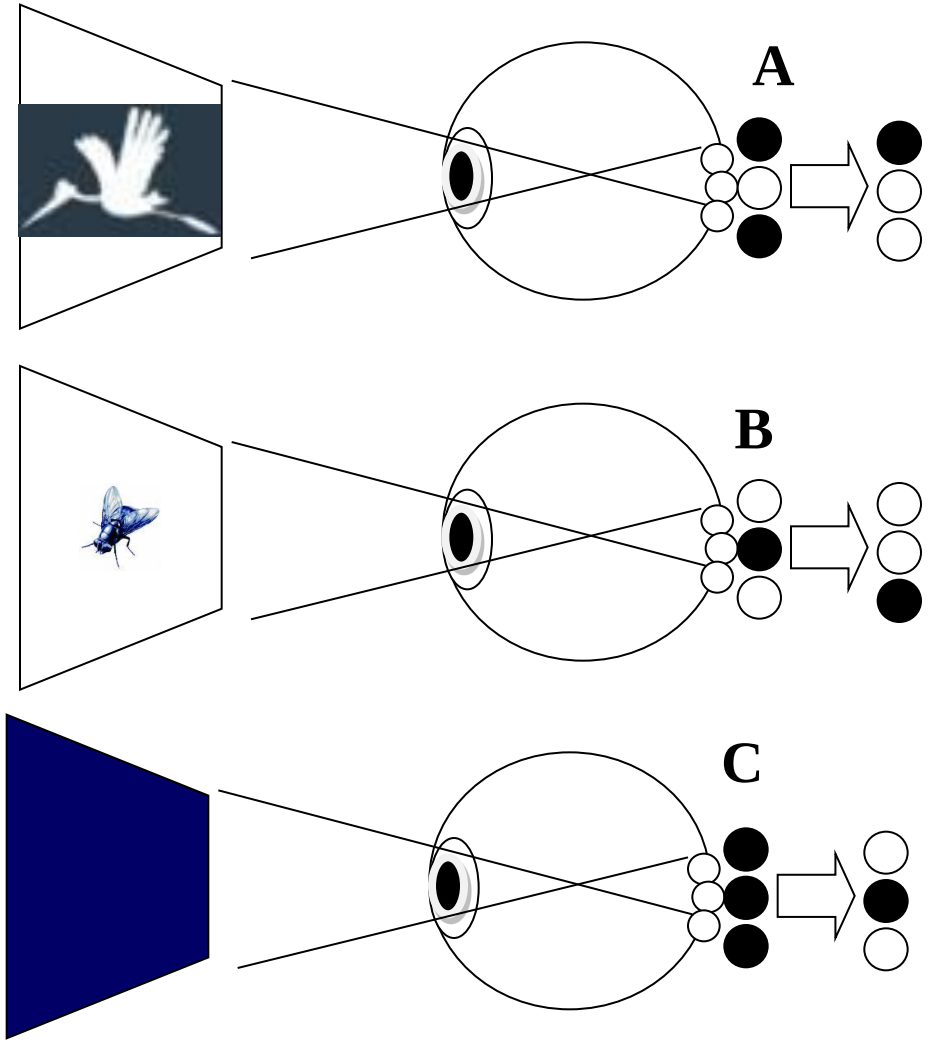
wzorzec A (bocian) , powinien się uaktywnić neuron wyjściowy **1**, który aktywuje mięsień uda i sprawia, ze żaba odskakuje;

wzorzec B (mucha), to powinien się uaktywnić neuron wyjściowy **3** pobudzający mięsień języka.

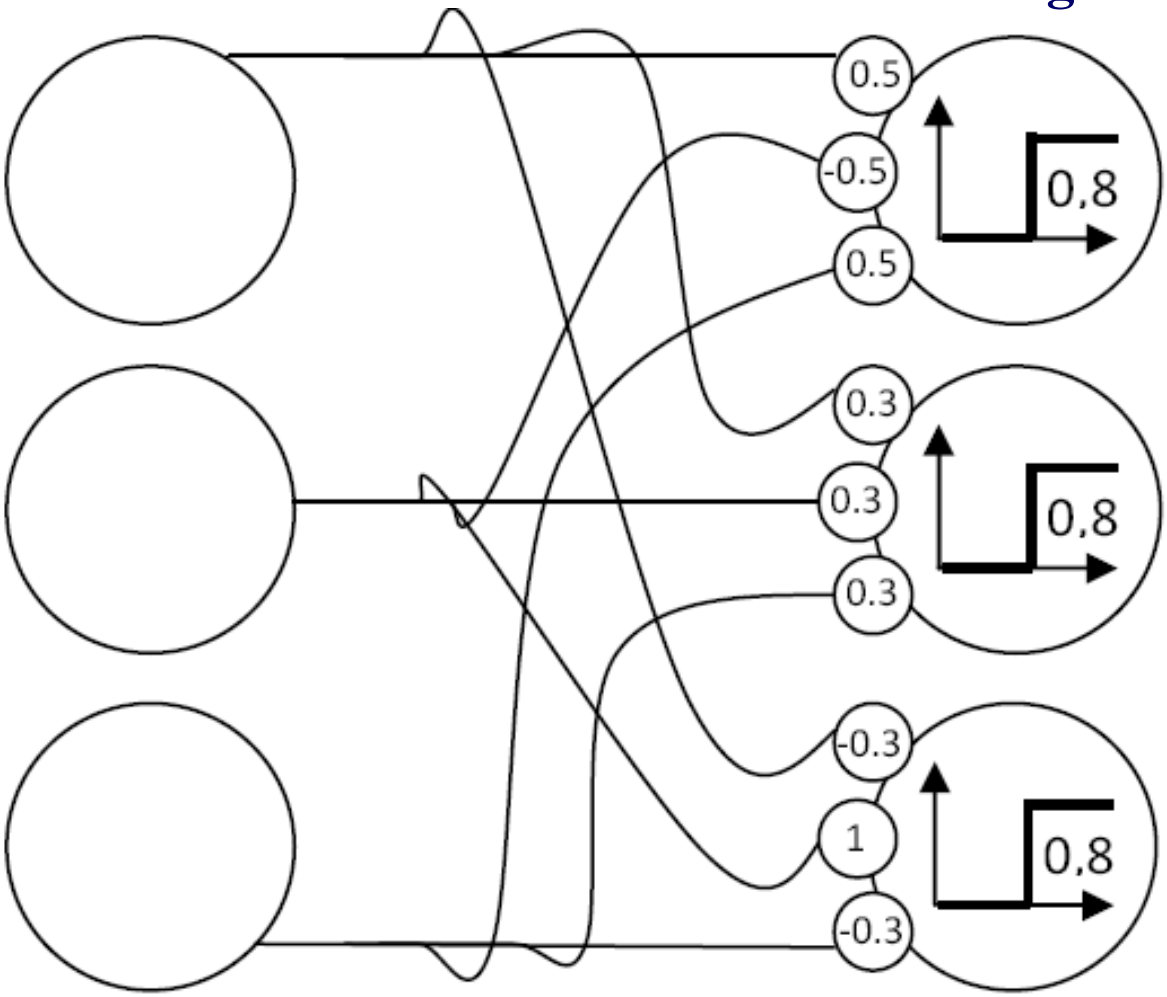
widok błękitnego nieba na siatkówce uaktywnia się **wzorzec C**, to wtedy aktywny powinien być neuron **2**, którego zadaniem jest stymulacja procesu trawienia.

Manfred Spitzer 1996

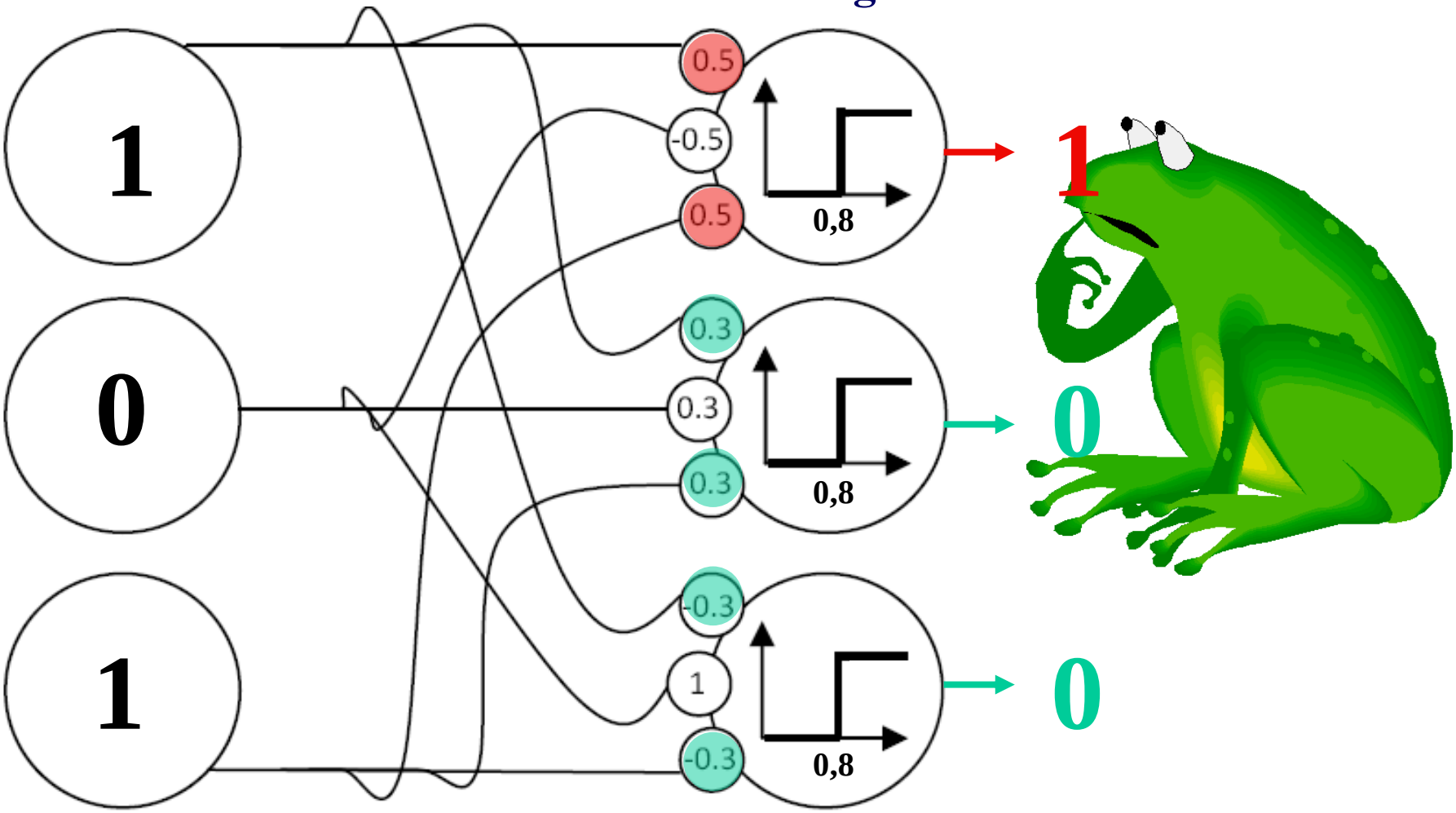
PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



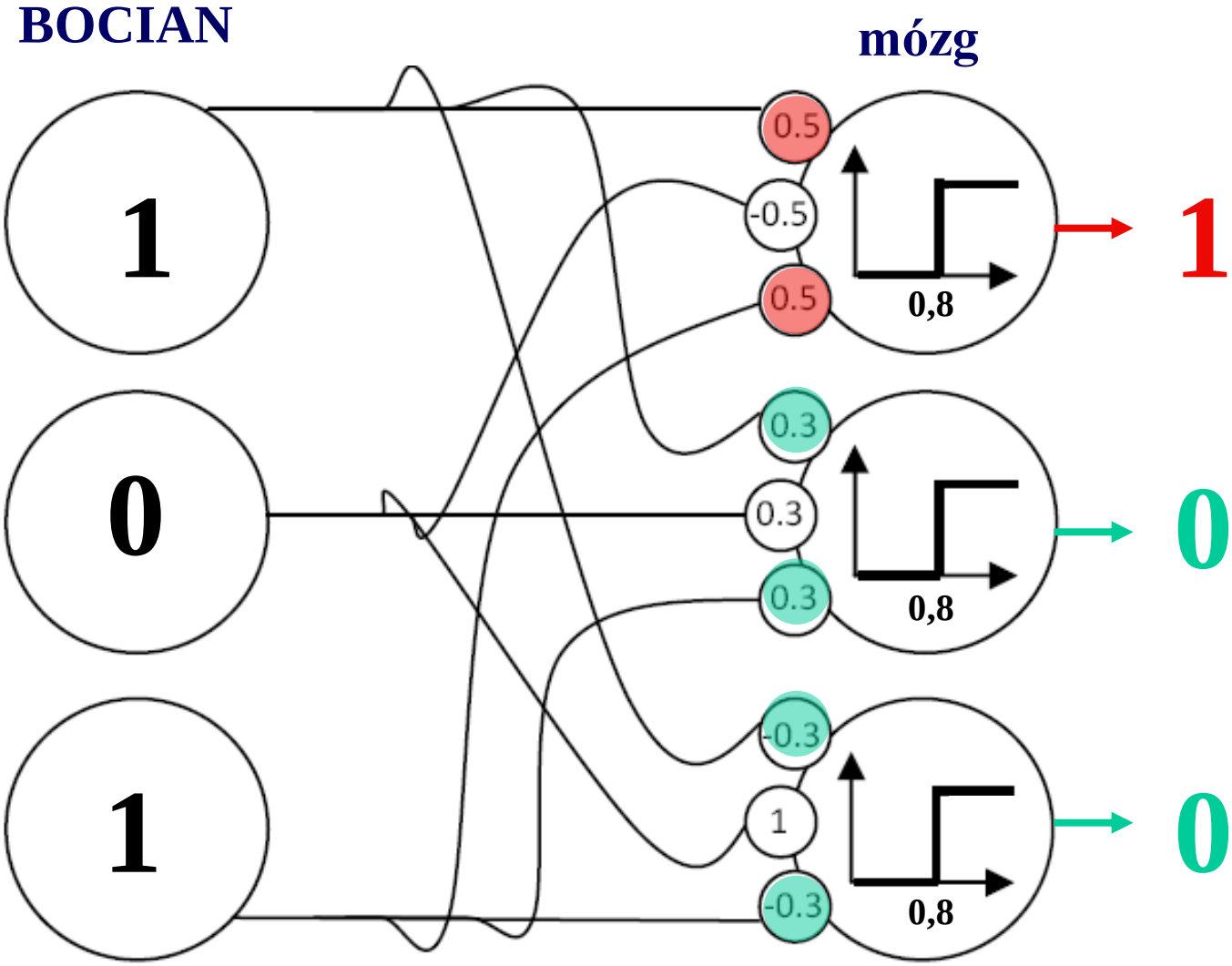
PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ

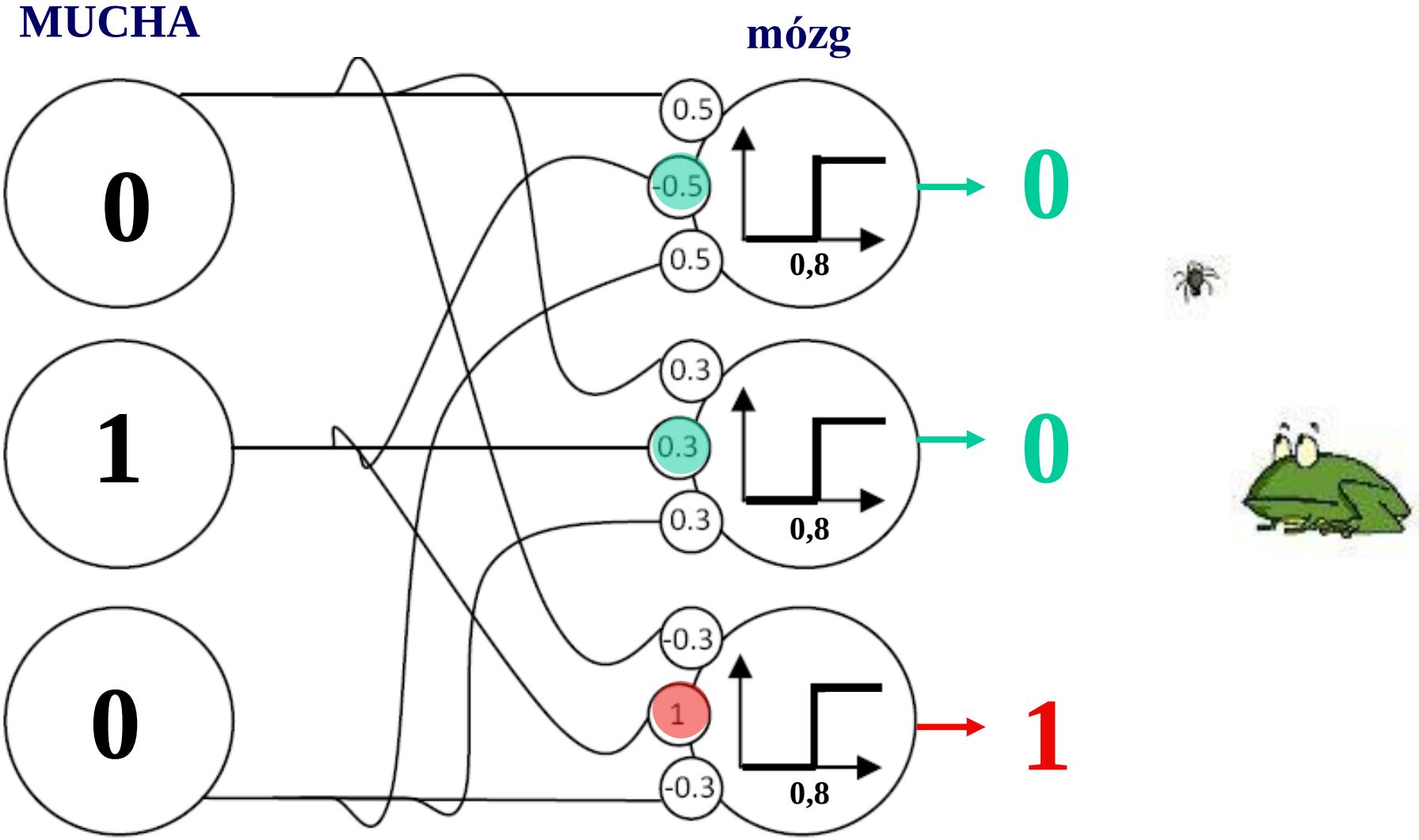


PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ

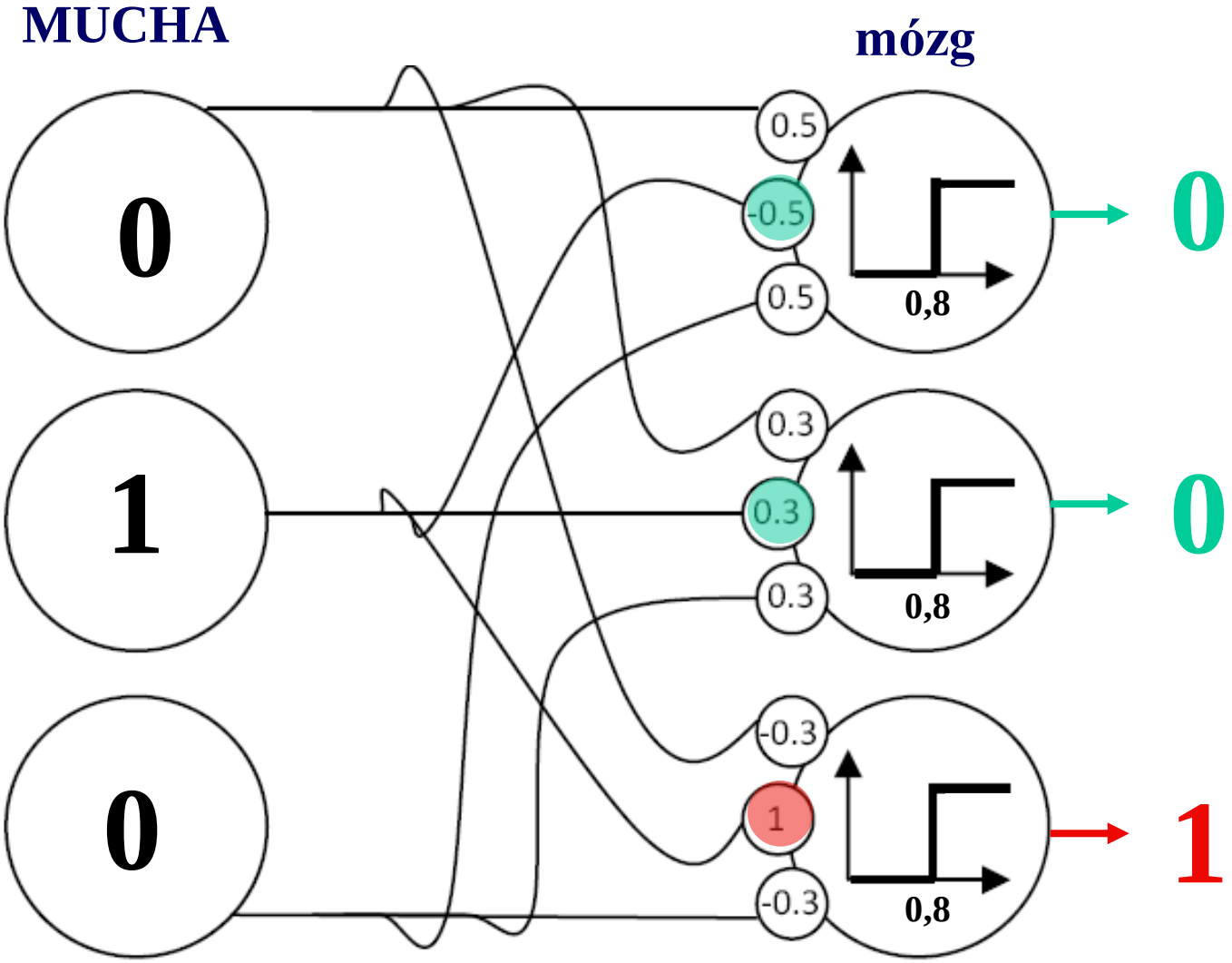


żaba odskakuje

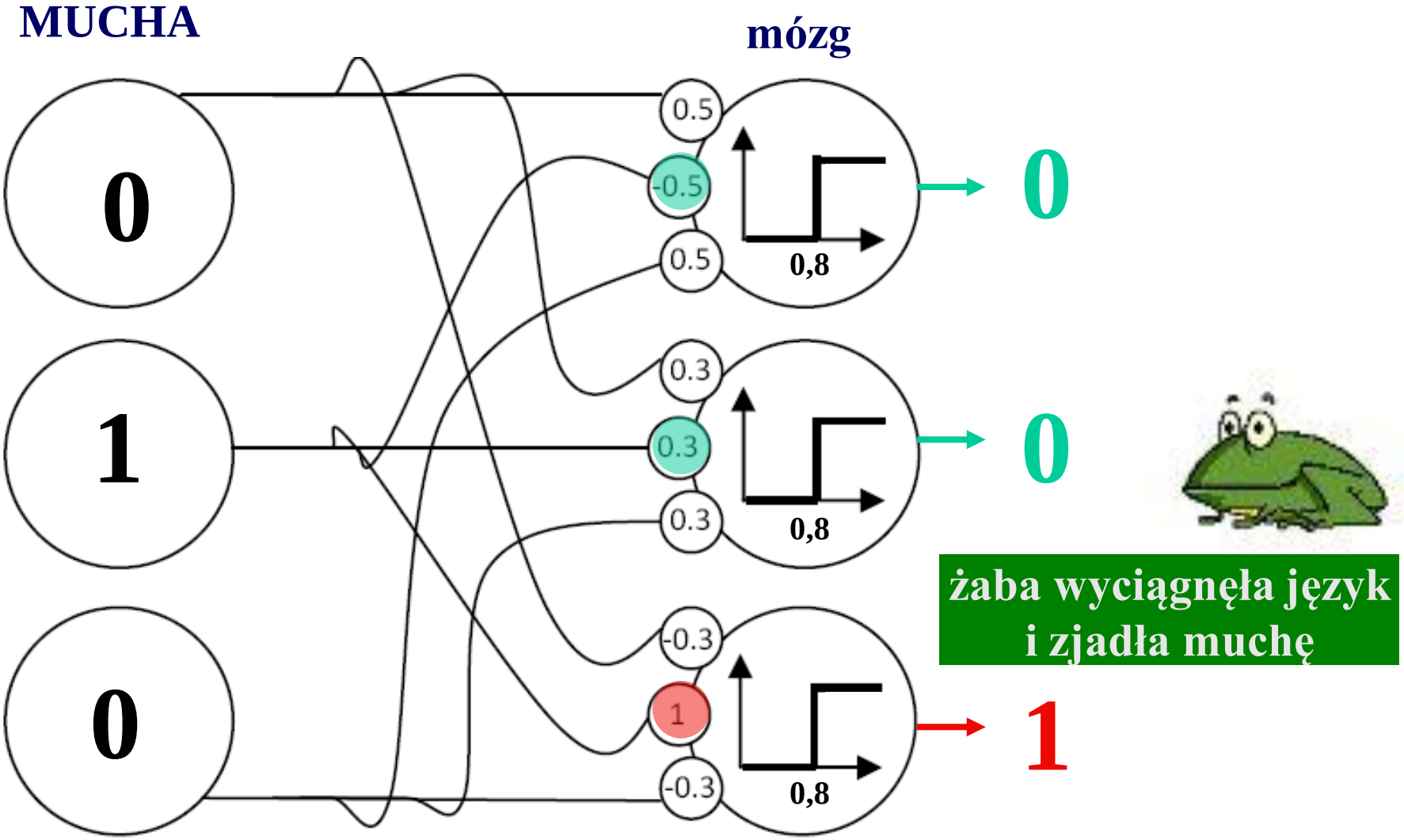
PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



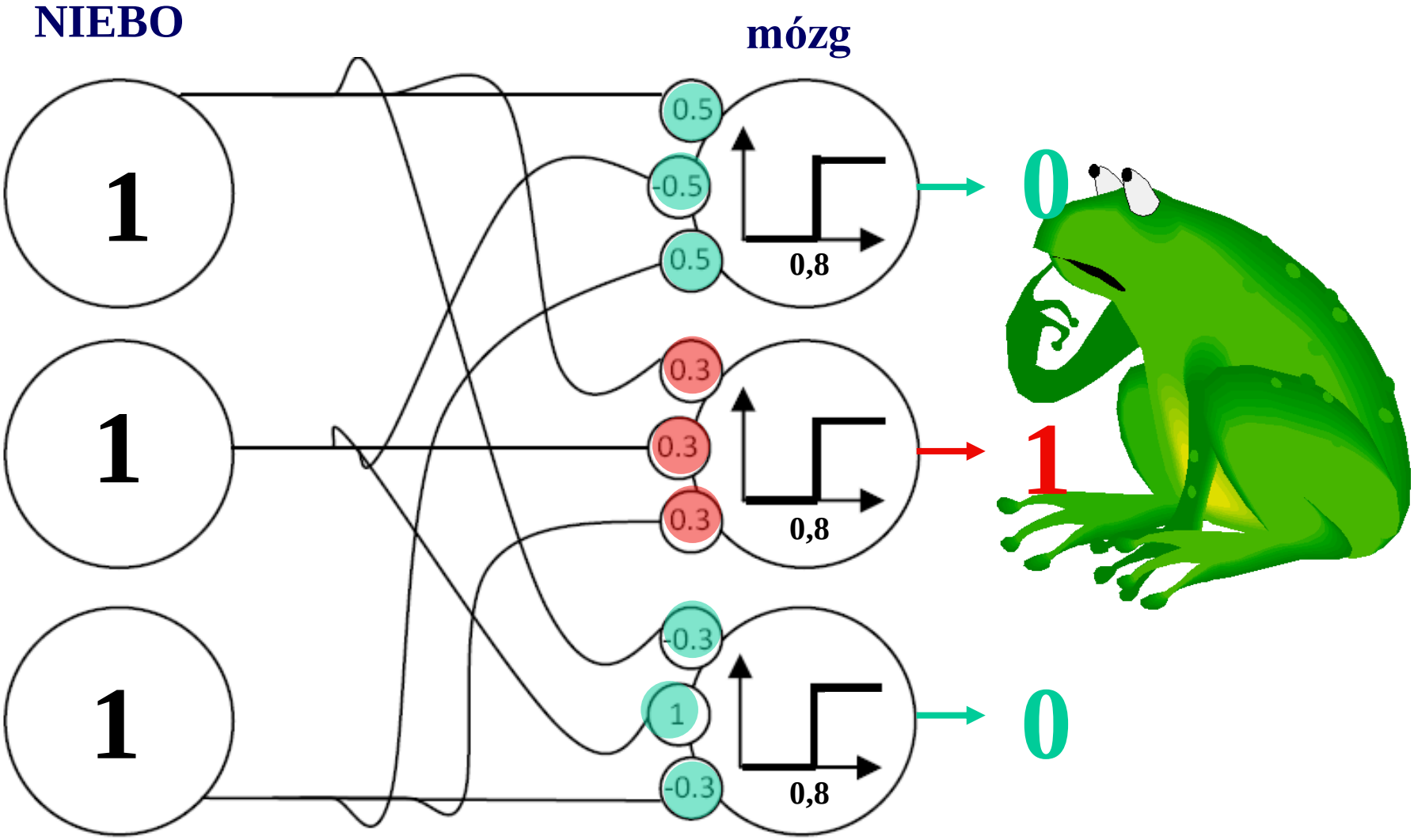
PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



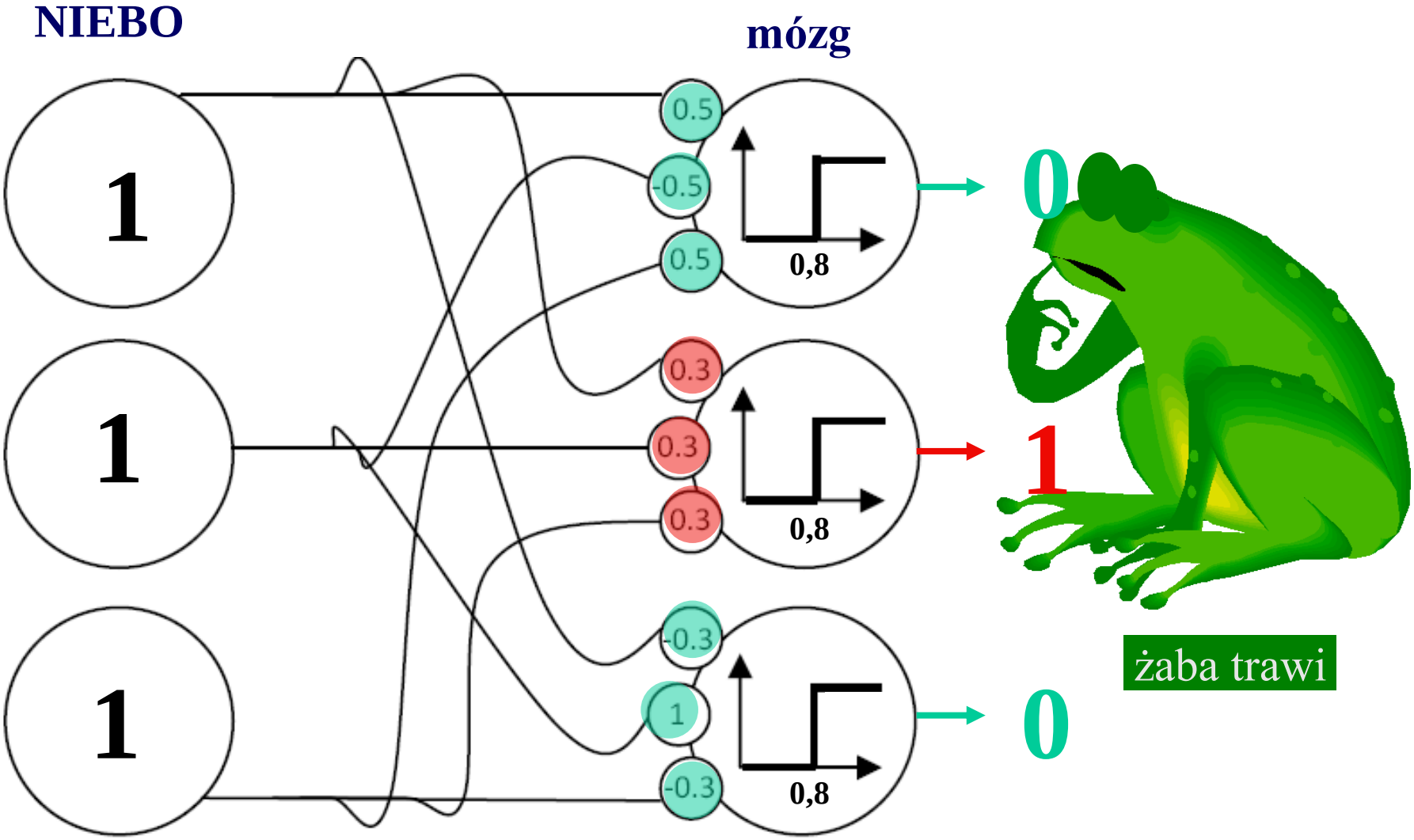
PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



PROSTY PRZYKŁAD DZIAŁANIA SIECI NEURONOWEJ



BIAS (przesunięcie) i PRÓG (threshold)

Jeżeli neuron posiada n wejść to wektor jego sygnałów wejściowych można przedstawić w postaci:

$$x = (x_1, x_2, \ldots, x_n)$$

a wektor wag połączeń jako wektor

$$w = (w_1, w_2, \ldots, w_n)$$

Neuron oblicza sumę ważoną tych wartości

$$e = b + \sum_{i=1}^{n} x_i w_i$$

gdzie b to waga połączenia bias .

Możemy przyjąć, że $w_0 = b$ i wtedy połączenie **bias** traktujemy dokładnie tak jak każdą inną wagę z tym, że sygnał wejściowy x_0 tego połączenia wynosi **1.**

Zatem łączne pobudzenie neuronu możemy zapisać w prostszej formie:

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

Funkcję aktywacji neuronu często przyjmujemy jako

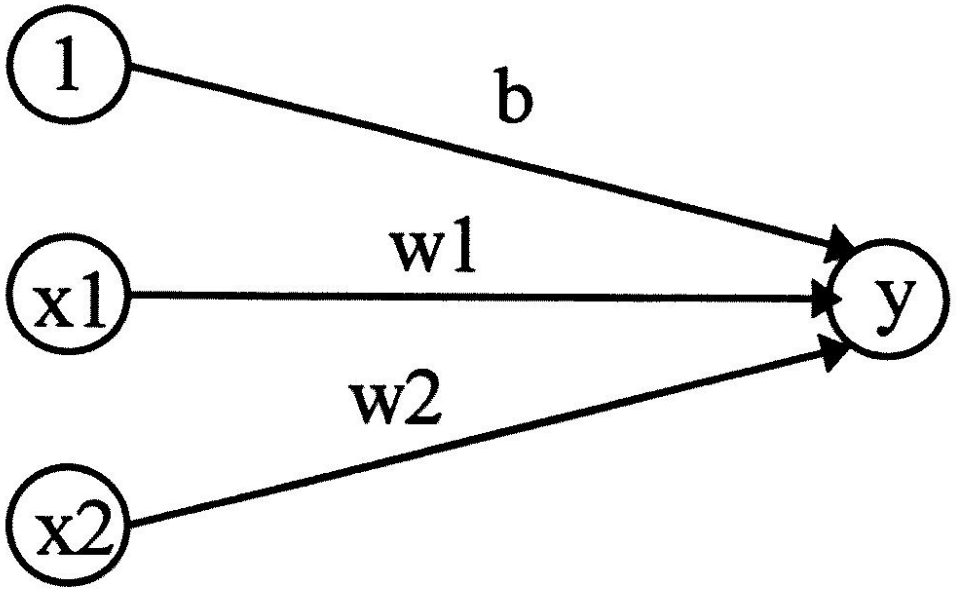
$$f(e) = \begin{cases} 1 & \text{gdy } e >= 0 \\ 0 & \text{gdy } e < 0 \end{cases}$$

Czasami zamiast elementu bias stosuje się stały **próg** i wtedy

$$f(e) = \begin{cases} 1 & \text{gdy } e >= \theta \\ 0 & \text{gdy } e < \theta \end{cases}$$

gdzie $e = \sum_{i=1}^{n} x_i w_i$

Zauważmy, że warunek $\Sigma\ \mathbf{x_i\ w_i = 0}$ i=1, ..., n dla n=2 oznacza równanie prostej, dla n=3 równanie płaszczyzny a ogólnie dla n>3 rozmaitość liniową stopnia n-1 czyli hiperpłaszczyznę w przestrzeni sygnałów wejściowych.

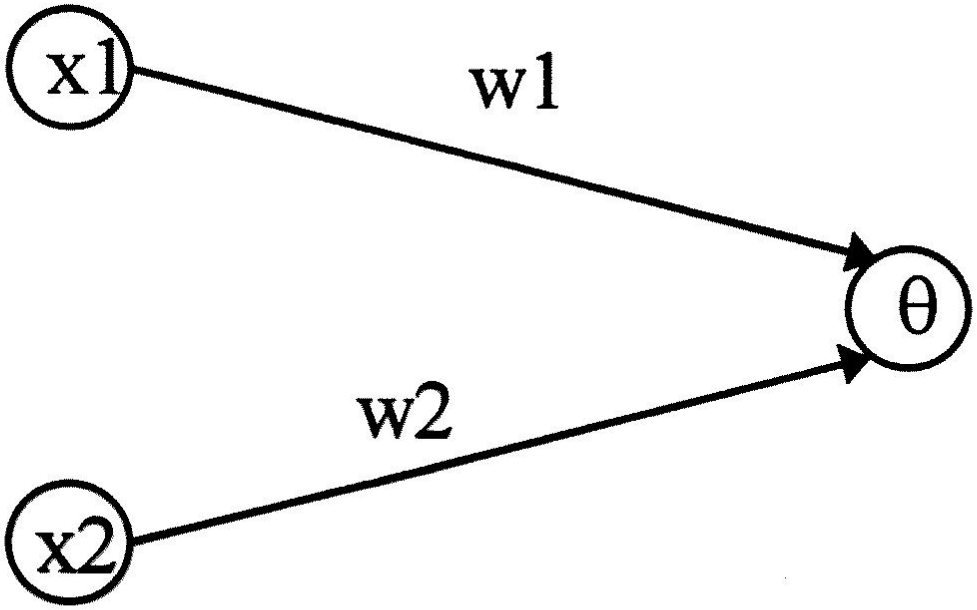


Granica pomiędzy wartościami, dla których sieć daje wartość 1 i 0 jest prostą

$$b + x_1 w_1 + x_2 w_2 = 0$$

lub (zakładając w_2 różne od 0)

$$x_2 = - w_1 / w_2 \, x_1 - b / w_2.$$



W przypadku wprowadzenia progu analogiczne równanie ma postać

$$x_1 w_1 + x_2 w_2 = \theta$$

lub (zakładając w_2 różne od 0)

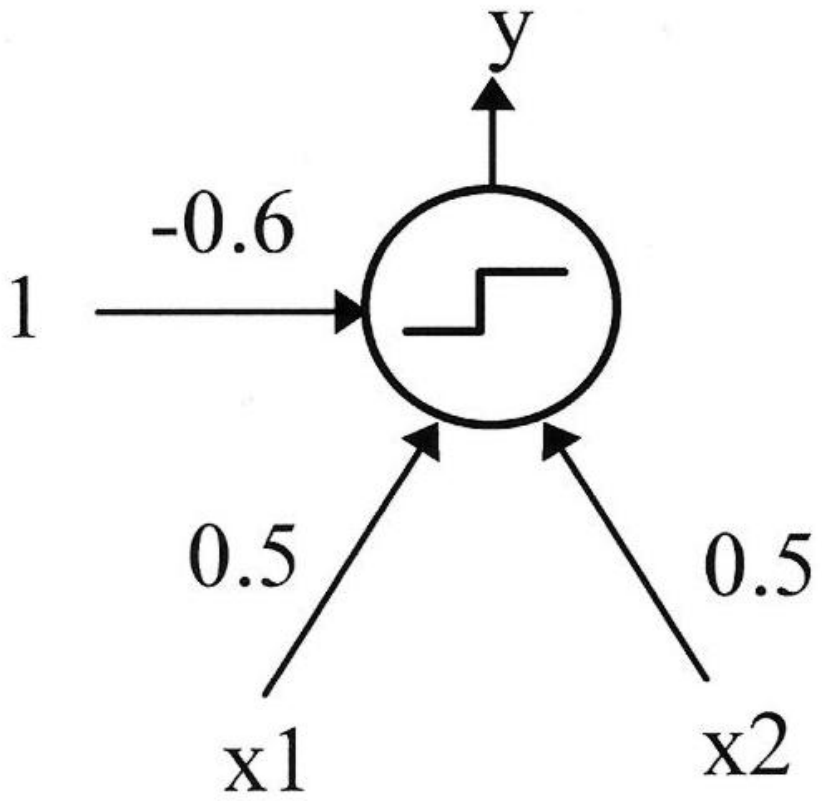
$$x_2 = - w_1 / w_2 \, x_1 + \theta / w_2.$$

$$x_2 = -w_1/w_2 - b/w_2 \qquad \textbf{BIAS}$$

$$x_2 = -w_1/w_2 + \theta/w_2 \qquad \textbf{PRÓG}$$

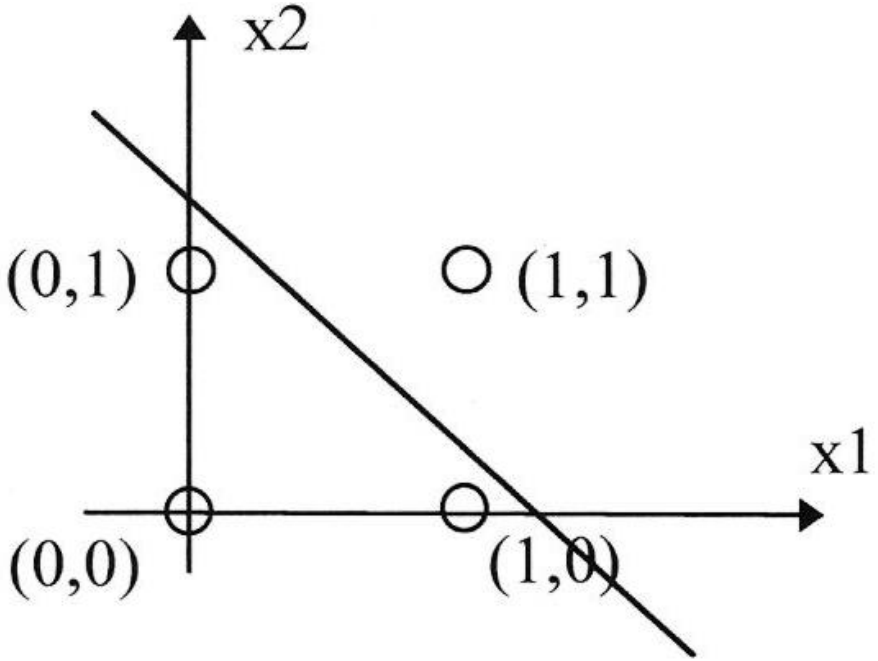
Przy porównaniu, równania z elementem bias i progiem wyglądają bardzo podobnie. Gdyby w obu sieciach wagi były takie same to b równałoby się -θ. Trzeba jednak pamiętać, że w trakcie uczenia wartość wagi bias jest modyfikowana natomiast próg θ pozostaje niezmienny. Korzystanie z jednej lub drugiej metody uzależnione jest od rozwiązywanego problemu.

ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU NIELINIOWEGO (Z FUNKCJĄ SKOKOWĄ)



Ta sieć realizuje funkcję logiczną AND

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

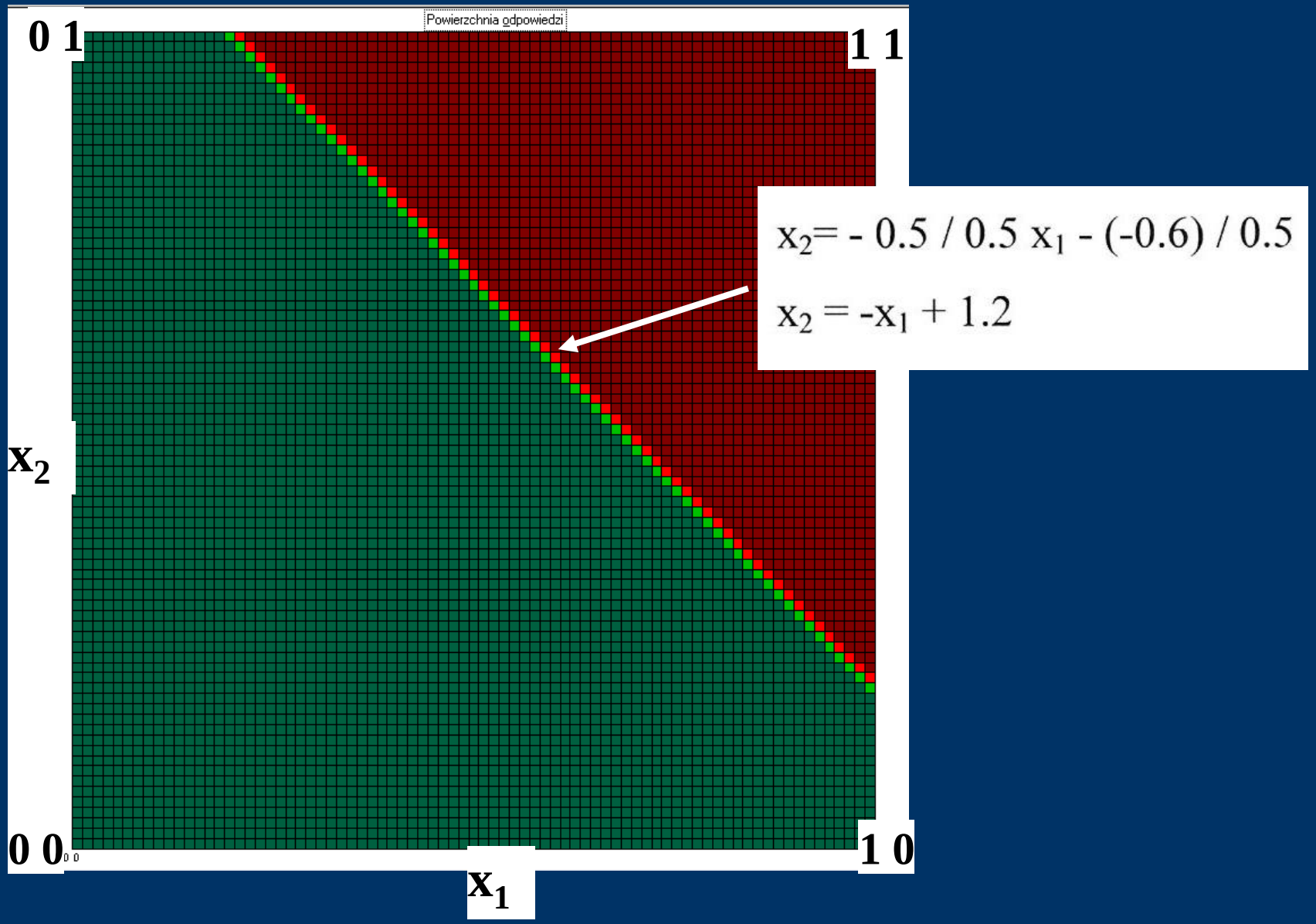


Prostą rozdzielającą punkty w przestrzeni sygnałów wejściowych równanie:

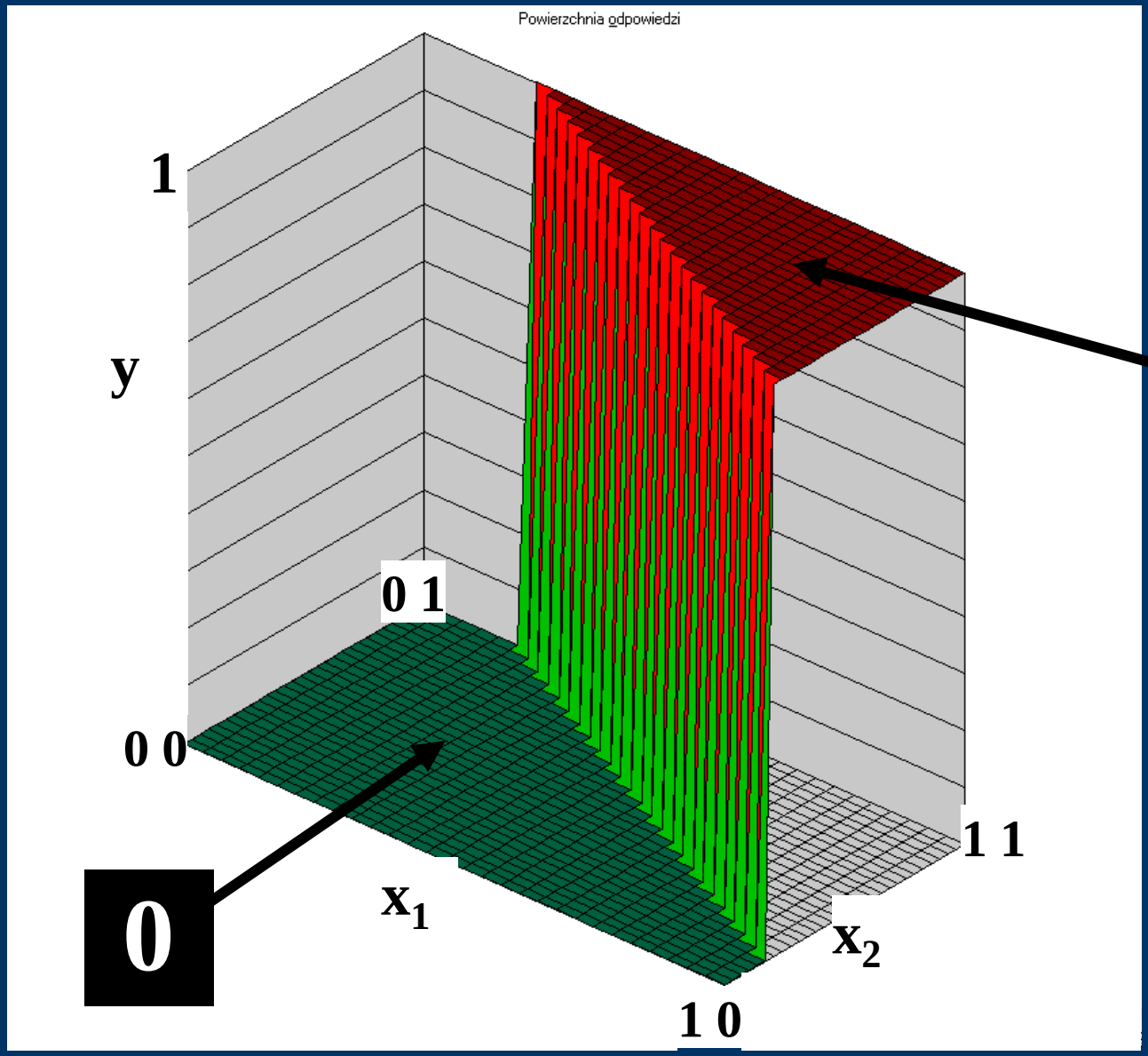
$$x_2 = - 0.5 / 0.5 \, x_1 - (-0.6) / 0.5$$

$$x_2 = -x_1 + 1.2$$

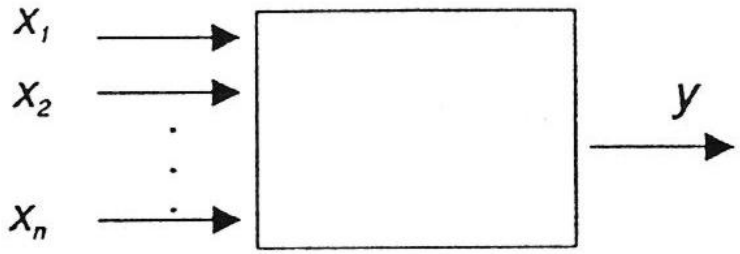
DYSKRYMINACJA LINIOWA



POWIERZCHNIA ODPOWIEDZI



NEURON LINIOWY



Sygnały wejściowe $x_i (i = 1, 2, \ldots, n)$ oraz sygnał wyjściowy y mogą przyjmować wart
z pewnego ograniczonego przedziału; z dokładnością do prostej funkcji skalującej mo
przyjąć, że

$$x_1 \in [-1,\ 1]$$

dla każdego i, a także

$$y \in [-1,\ 1]$$

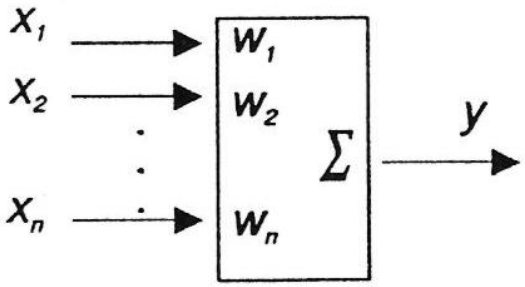
Zależność

$$y = f(x_1, x_2, \ldots, x_n)$$

w najprostszym przypadku może być rozważana jako liniowa:

$$y = \sum_{i=1}^{n} w_i\, x_i$$

NEURON LINIOWY



Aby to wyjaśnić, wystarczy zastosować notację wektorową. Niech zestaw sygnałów wejściowych neuronu tworzy wektor

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

zapisywany przez nas jako kolumna o n składowych. Wektor ten interpretować można także jako punkt w n-wymiarowej przestrzeni $\mathbf{X}$, nazywanej przestrzenią wejść. Wektor ten (i inne wektory) zapisywać niekiedy będziemy w wygodniejszej postaci

$$\mathbf{X} = < x_1, x_2, \ldots, x_n >^T$$

gdzie T jest symbolem transpozycji.

NEURON LINIOWY

Zestaw n współczynników wagowych także można rozpatrywać jako wektor

$$\mathbf{W} =< w_1, w_2, \ldots, w_n >^T$$

wyznaczający punkt w n-wymiarowej przestrzeni $\mathbf{W}$, nazywanej przestrzenią wag. Przy tych założeniach równanie neuronu wyrazić można jako iloczyn skalarny wektora wejść i wektora wag:

$$y = \mathbf{W} * \mathbf{X}$$

Z formalnego punktu widzenia wygodniej będzie zastąpić iloczyn skalarny (zapisany wyżej z pomocą specjalnego operatora: gwiazdki $*$) zwyklym iloczynem *transponowanego* wektora $\mathbf{W}$ i wektora $\mathbf{X}$

$$y = \mathbf{W}^T \mathbf{X}$$

Z ogólnie znanych właściwości iloczynu skalarnego wynika, że sygnał wyjściowy neuronu y będzie tym większy, im bardziej położenie wektora wejściowego $\mathbf{X}$ w przestrzeni $\mathbf{X}$ przypominać będzie położenie wektora wag $\mathbf{W}$ w przestrzeni $\mathbf{W}$. W ten sposób można powiedzieć, że neuron rozpoznaje[1] sygnały wejściowe, wyróżniając te, które są podobne do jego wektora wag. Aby to jeszcze silniej zaakcentować załóżmy, że wektory $\mathbf{X}$ i $\mathbf{W}$ są znormalizowane, to znaczy

$$\|\mathbf{X}\| = \mathbf{X}^T \mathbf{X} = 1$$

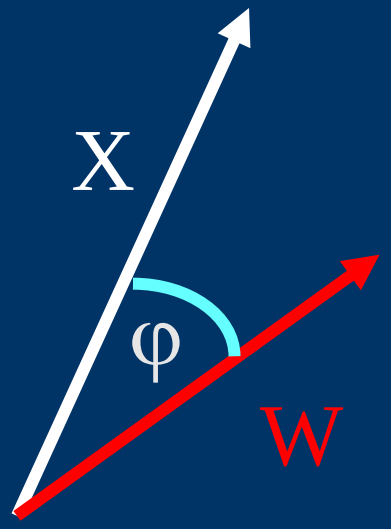
oraz

$$\|\mathbf{W}\| = \mathbf{W}^T \mathbf{W} = 1$$

W tym wypadku sygnał wyjściowy neuronu wyznaczyć można ze wzoru

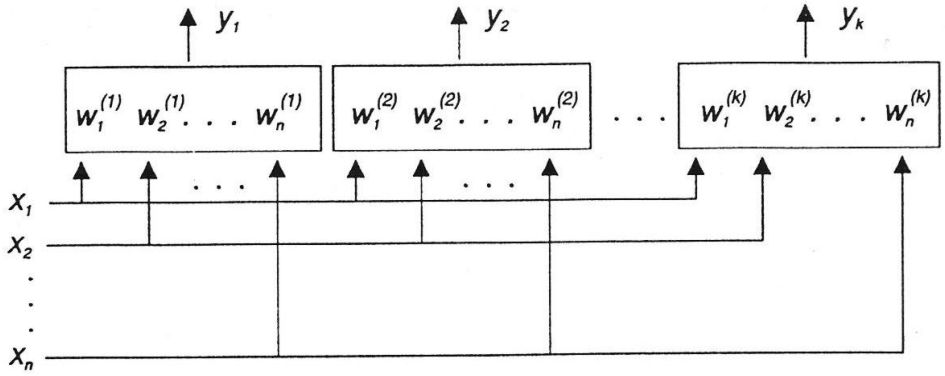
$$y = \cos\varphi$$

gdzie φ jest kątem pomiędzy wektorami $\mathbf{W}$ i $\mathbf{X}$. Jeśli na wejście rozważanego neuronu podawać będziemy różne sygnały $\mathbf{X}$, to wyjście neuronu y będzie miało tym większą wartość, im bardziej podany sygnał $\mathbf{X}$ będzie podobny do „wzorcowego" sygnału, który neuron pamięta w postaci swojego zestawu wag $\mathbf{W}$.



$$y = \cos\varphi$$

SIEĆ LINIOWA



Rozważmy teraz *warstwę* neuronów, z których każdy ma ten sam zestaw sygnałów wejściowych $\mathbf{X} =< x_1, x_2, \ldots, x_n >^T$, natomiast każdy ma swój własny wektor wag. Ponumerujmy neurony w warstwie i oznaczmy jako $\mathbf{W}^{(m)} =< w_1^{(m)}, w_2^{(m)}, \ldots, w_n^{(m)} >^T$ wektor wag m-tego neuronu ($m = 1, 2, \ldots, k$). Wówczas oczywiście sygnał wyjściowy m-tego nauronu można wyznaczyć ze wzoru

$$y^{(m)} = \mathbf{W}^{(m)} * \mathbf{X} = \sum_{i=1}^{n} w_i^{(m)} x_i$$

Stosując konsekwentnie notację wektorową można sygnały wyjściowe z rozważanej warstwy neuronów zebrać w formie wektora

$$\mathbf{Y} =< y_1, y_2, \ldots, y_k >^T$$

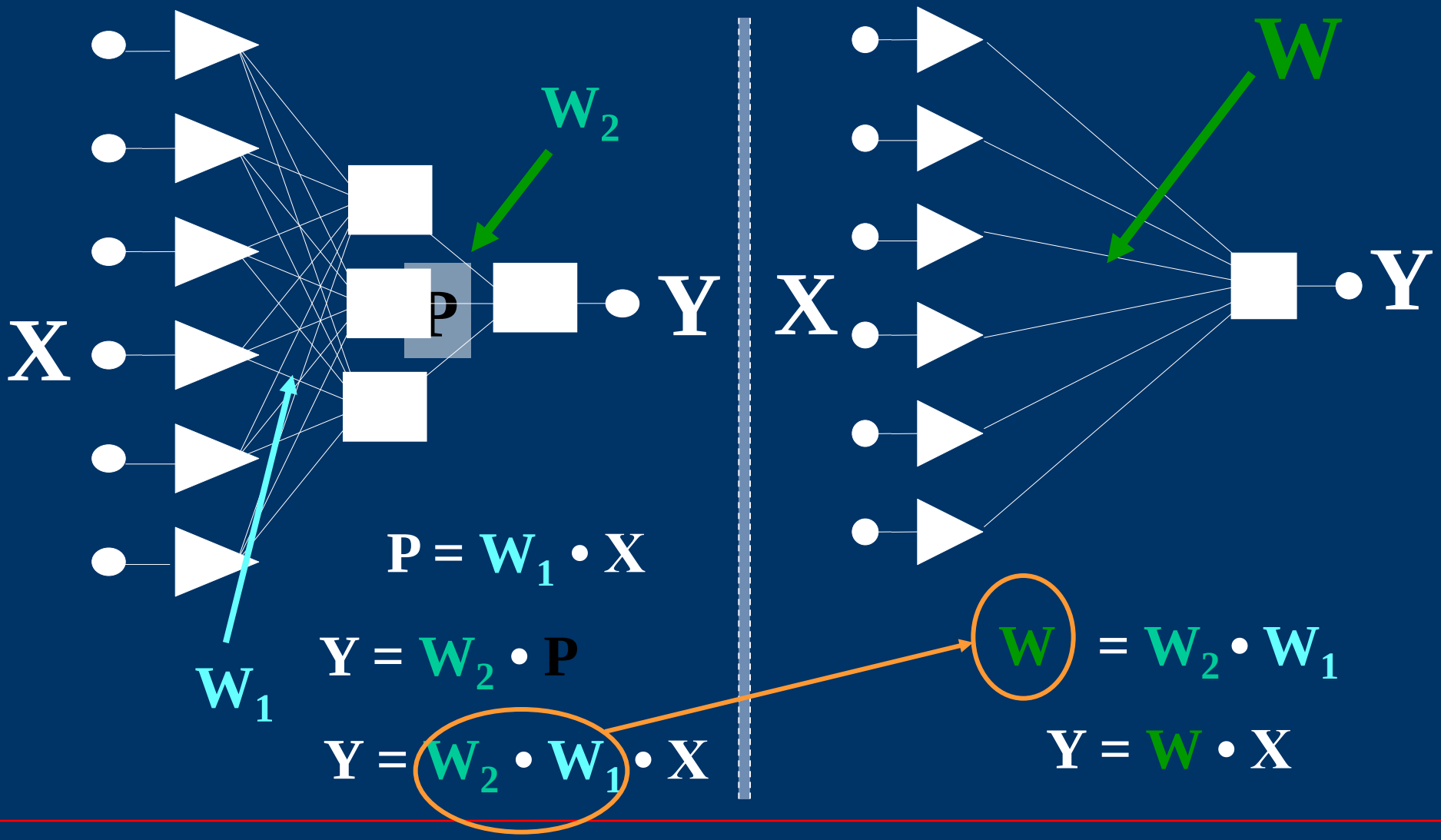
Wektor ten można wyznaczyć mnożąc wektor wejściowy $\mathbf{X}$ przez macierz $\mathbf{W}_k$ o wymiarach $[k \times n]$, utworzoną w taki sposób, że jej kolejnymi wierszami są (transponowane) kolejne wektory $\mathbf{W}^{(m)}$ (dla $m = 1, 2, \ldots, k$). Macierz $\mathbf{W}_k$ ma więc następującą budowę:

$$\mathbf{W_k} = \begin{bmatrix} w_1^{(1)} & w_2^{(1)} & \ldots & w_n^{(1)} \\ w_1^{(2)} & w_2^{(2)} & \ldots & w_n^{(2)} \\ \vdots & \vdots & & \vdots \\ w_1^{(k)} & w_2^{(k)} & \ldots & w_n^{(k)} \end{bmatrix}$$

Wykorzystując macierz $\mathbf{W}_k$ można zapisać funkcje realizowane przez całą sieć w formie jednego zwartego wzoru:

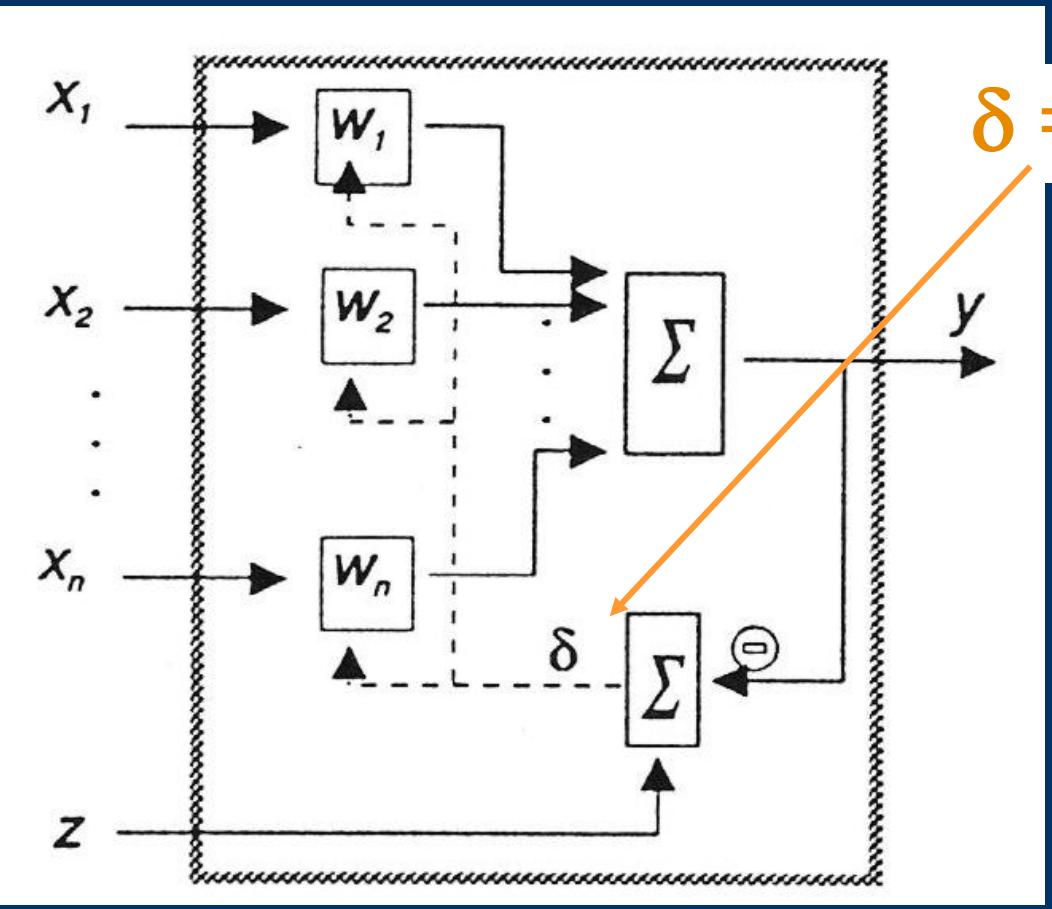
$$\boldsymbol{Y} = \boldsymbol{W}_k \; \boldsymbol{X}$$

LINIOWA SIEĆ NEURONOWA



Sieć liniowa z zasady nie posiada warstw ukrytych, bo nawet jeśli się je wprowadzi, to nie wzbogacą one zachowania sieci

ADALINE (ADAptive LINear Element)



$\delta = z - y$

Reguła WIDROW-HOFFA

$$W' = W + \eta \, \delta \, X$$

η - szybkość uczenia

albo obliczamy wynik stosując pseudoinwersję macierzy W

$$W' = Z \cdot X^{-1}$$

bo $Z = W \cdot X$

REGUŁA WIDROW-HOFFA

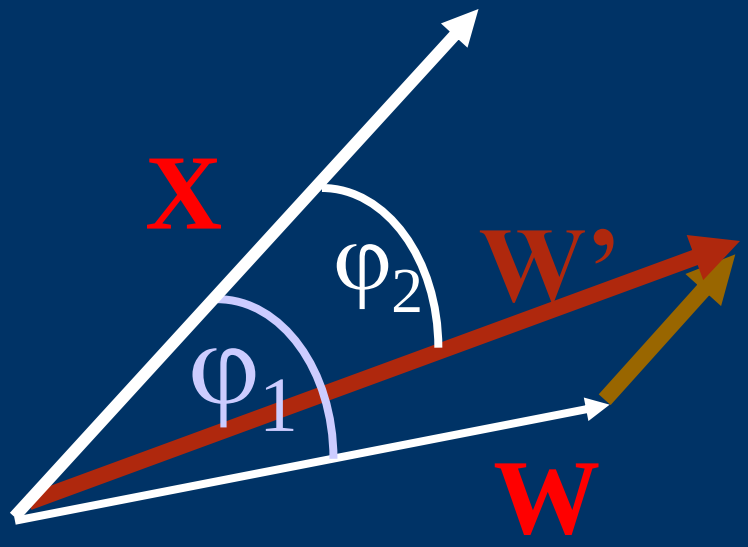
$$W' = W + \eta \delta X$$

$\eta >= 0$

$\delta = z - y$

$\delta >= 0$ ⇒ $z - y >= 0$ ⇒ $z > y$

czyli odpowiedź sieci jest **ZA MAŁA**



$\varphi_2 < \varphi_1$

$\cos \varphi_2 > \cos \varphi_1$

$y_2 > y_1$

REGUŁA WIDROW-HOFFA

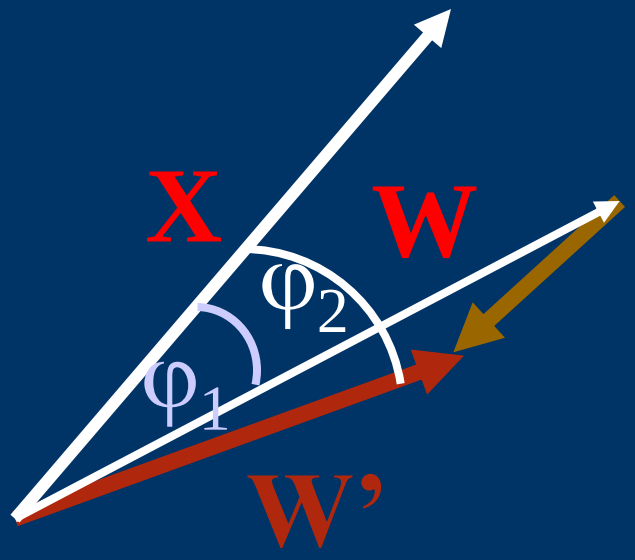
$$W' = W + \eta \delta X$$

$\eta >= 0$

$\delta = z - y$

$\delta <= 0$ ⇒ $z - y <= 0$ ⇒ $z < y$

czyli odpowiedź sieci jest **ZA DUŻA**



$\varphi_2 > \varphi_1$

$\cos \varphi_2 < \cos \varphi_1$

$y_2 < y_1$

PROCES UCZENIA
matematyczne aspekty

CIĄG UCZĄCY

$$U = << \mathbf{X}^{(1)}, z^{(1)} >, < \mathbf{X}^{(2)}, z^{(2)} >, \ldots, < \mathbf{X}^{(N)}, z^{(N)} >>$$

PROCES UCZENIA
matematyczne aspekty

reguła DELTA

zastosowana w j+1 kroku

$$W^{(j+1)} = W^{(j)} + \eta^{(j)} \, \delta^{(j)} \, X^{(j)}$$

BŁĄD UCZENIA

$$\delta^{(j)} = z^{(j)} - y^{(j)}$$

ODPOWIEDŹ SIECI

$$y^{(j)} = W^{(j)} * X^{(j)}$$

CEL UCZENIA

OPTYMALIZACJA WAG

ZGODNOŚĆ ODPOWIEDZI NEURONU Z WYMAGANYMI WARTOŚCIAMI TZN. MINIMALIZACJA FUNKCJI BŁĘDU CZYLI PEWNEJ FUNKCJI KRYTERIALNEJ:

$$Q = \frac{1}{2} \sum_{j=1}^{N} (z^{(j)} - y^{(j)})^2$$

j – numer elementu uczącego

$$Q = \sum_{j=1}^{N} Q^{(j)}$$

gdzie $Q^{(j)} = \frac{1}{2} (z^{(j)} - y^{(j)})^2$

szukanie minimum metodą gradientową

$$Q = Q(W)$$

$$w'_i - w_i = \Delta w_i = -\eta \frac{\partial Q^{(j)}}{\partial w_i}$$

$$\frac{\partial Q^{(j)}}{\partial w_i} = \frac{\partial Q^{(j)}}{\partial y^{(j)}} \frac{\partial y^{(j)}}{\partial w_i}$$

$$\frac{\partial Q^{(j)}}{\partial y^{(j)}} = \frac{\partial (\tfrac{1}{2} (z^{(j)} - y^{(j)})^2)}{\partial y^{(j)}} = -(z^{(j)} - y^{(j)}) = -\delta^{(j)}$$

$$\frac{\partial Q^{(j)}}{\partial y_i} = \frac{\partial \left(\tfrac{1}{2} \left(z^{(j)} - y^{(j)}\right)^2\right)}{\partial y_i} = -\left(z^{(j)} - y^{(j)}\right) = -\delta^{(j)}$$

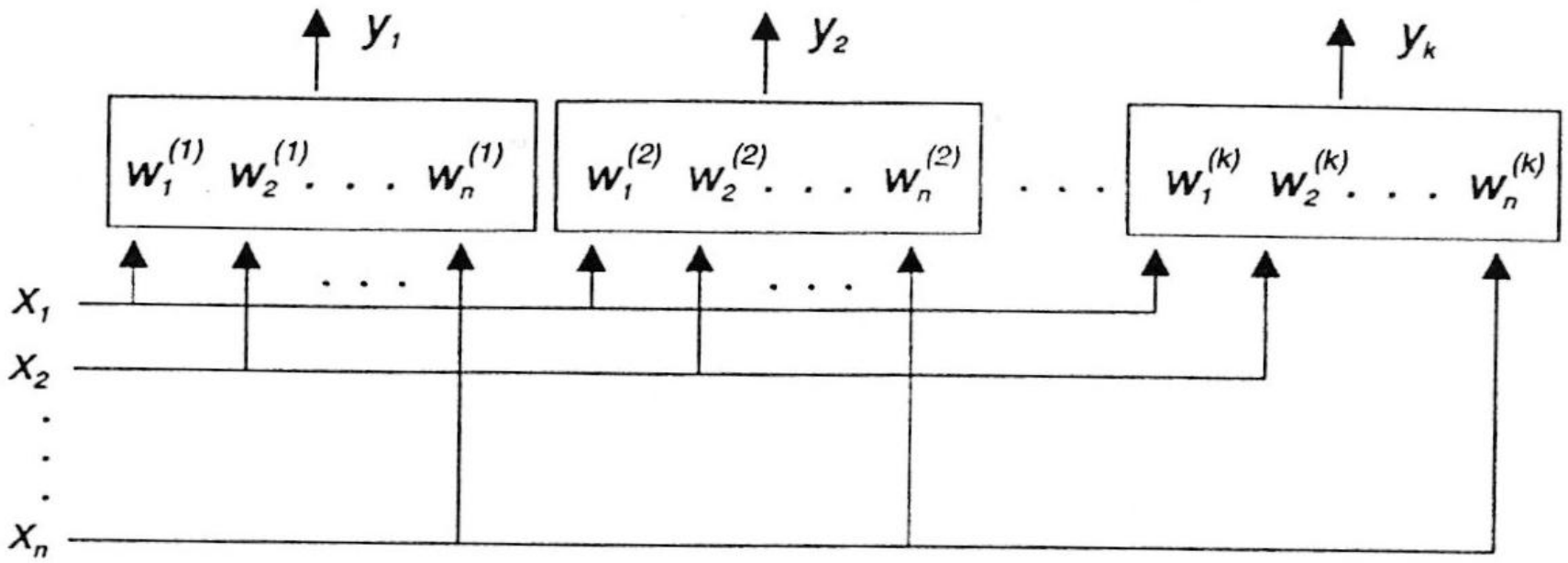
$$\frac{\partial y^{(j)}}{\partial w_i} = \frac{\partial \left(\sum x_k w_k\right)^{(j)}}{\partial w_i} = x_i^{(j)}$$

$$\Delta w_i = -\eta \frac{\partial Q^{(j)}}{\partial w_i} = -\eta \left(-\delta^{(j)}\right) x_i^{(j)}$$

$$= \eta \, \delta^{(j)} \, x_i^{(j)}$$

co należało wykazać

MADALINE (Many ADAptive LINear Elements – Many ADALINEs)



MADALINE (Many ADAptive LINear Elements – Many ADALINEs)

Obiektem podlegającym uczeniu jest w tym wypadku *macierz* $\mathbf{W}_k$, a ciąg uczący i
postać

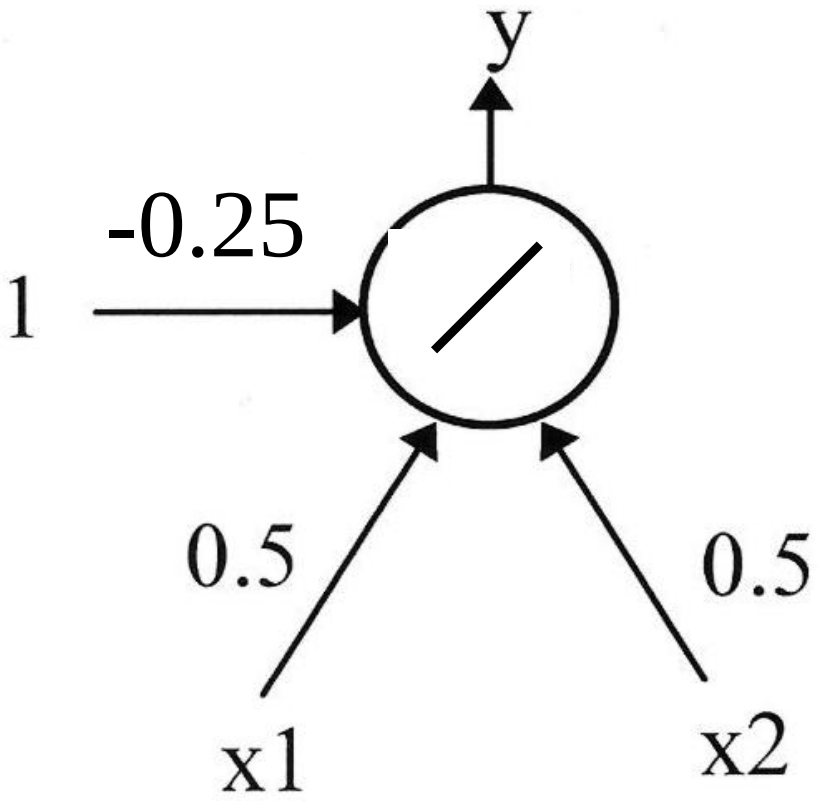
$$U = << \mathbf{X}^{(1)}, \mathbf{Z}^{(1)} >, < \mathbf{X}^{(2)}, \mathbf{Z}^{(2)} >, \ldots, < \mathbf{X}^{(N)}, \mathbf{Z}^{(N)} >>$$

gdzie $\mathbf{Z}^{(j)}$ są k-elementowymi wektorami oznaczającymi wymagane zestawy odpowiedzi si
na wymuszenia danych odpowiednimi wektorami $\mathbf{X}^{(j)}$. Sieć taka w literaturze nazywana j
MADALINE (*Many* ADALINE *s*). Uczenie sieci MADALINE odbywa się w sposób całkowi
analogiczny do wyżej opisanego, z tą tylko różnicą, że formula uczenia ma w tym wypad
postać macierzową:

REGUŁA WIDROW - HOFFA

$$\mathbf{W}_k^{(j+1)} = \mathbf{W}_k^{(j)} + \eta \left(\mathbf{Z}^{(j)} - \mathbf{Y}^{(j)}\right)\left(\mathbf{X}^{(j)}\right)^T$$

ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU LINIOWEGO

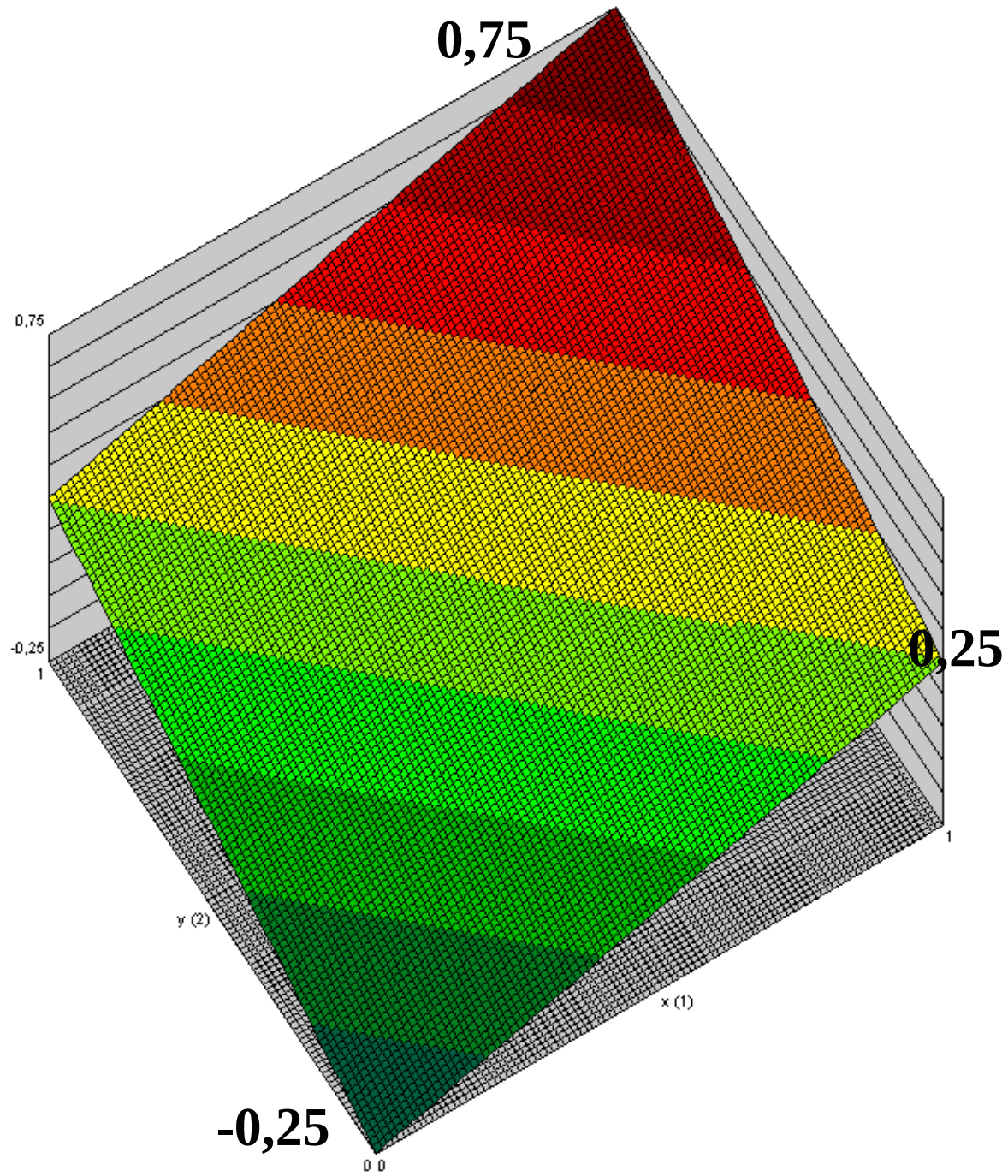


Ta sieć przybliża funkcję AND przy pomocy regresji liniowej

x_1	x_2	y
0	0	**-0.25**
0	1	**0.25**
1	0	**0.25**
1	1	**0.75**

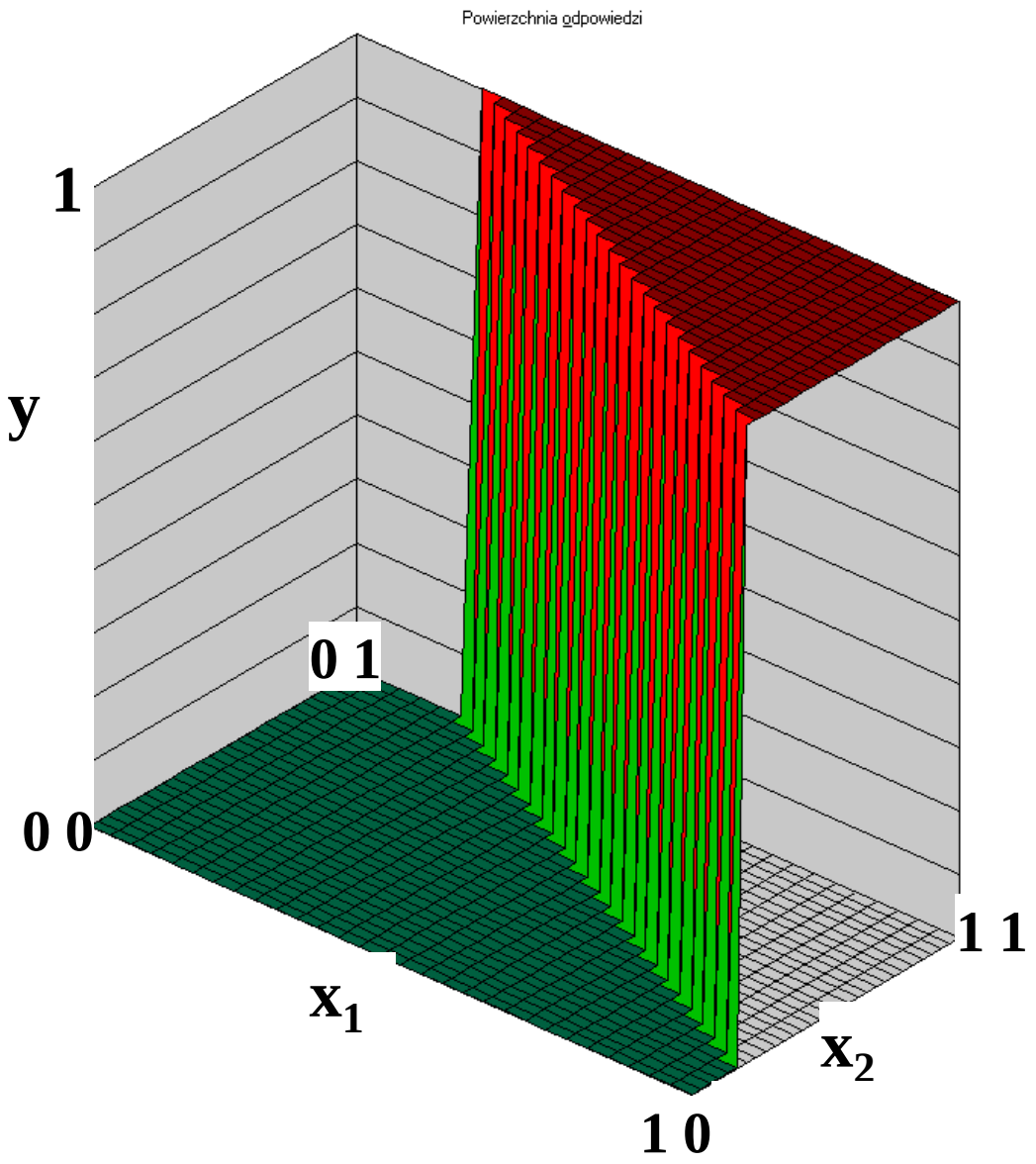
ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU LINIOWEGO

NEURON LINIOWY WYTWARZA W ODPOWIEDZI PŁASZCZYZNĘ



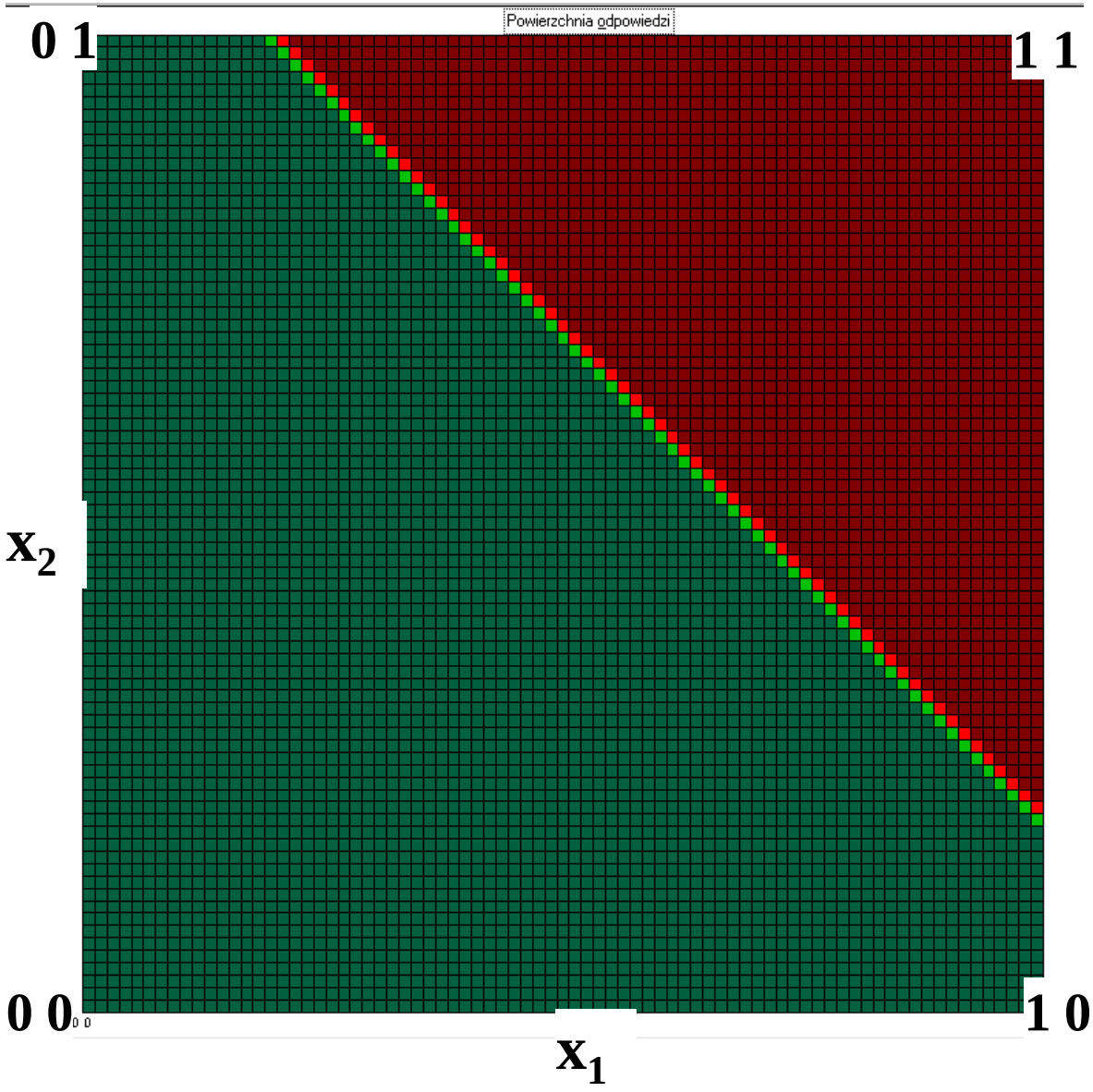
ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU SKOKOWEGO

POWIERZCHNIA ODPOWIEDZI

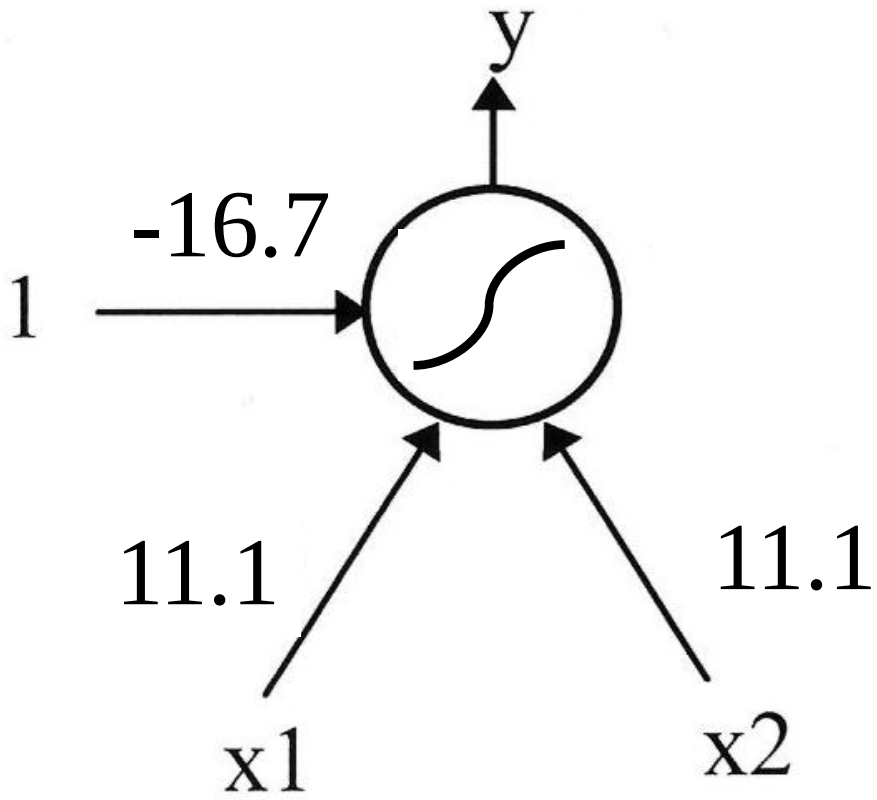


ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU SKOKOWEGO

RZUT POWIERZCHNIA ODPOWIEDZI NA PŁASZCZYZNĘ X_1X_2



ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU NIELINIOWEGO LOGISTYCZNEGO (SIGMOIDY)

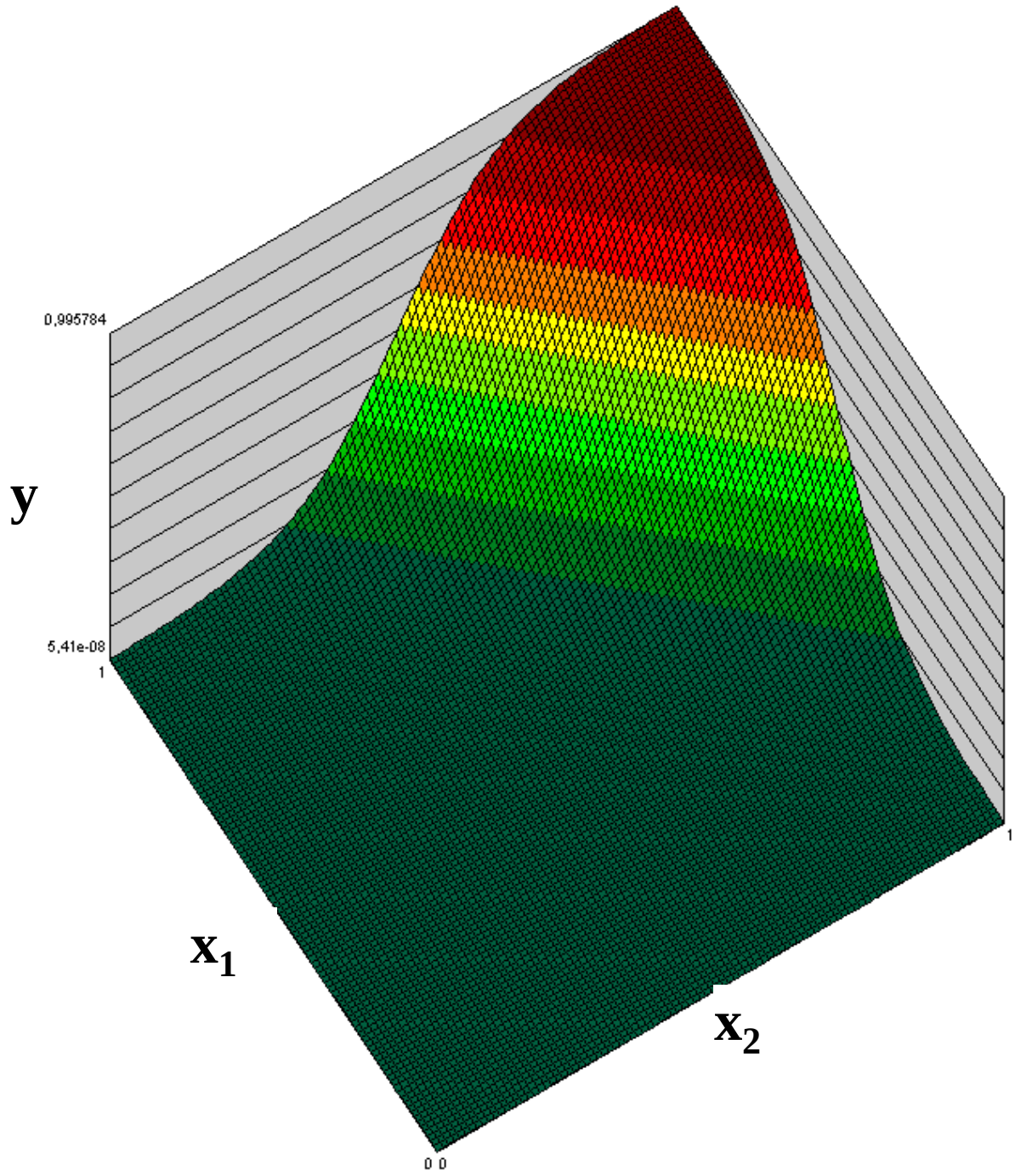


Ta sieć realizuje funkcję logiczną AND (z dowolną dokładnością)

x_1	x_2	y
0	0	**0**
0	1	**0.004**
1	0	**0.004**
1	1	**0.996**

ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU LOGISTYCZNEGO (SIGMOIDY

POWIERZCHNIA ODPOWIEDZI



ROZWIĄZANIE PROBLEMU AND PRZY POMOCY NEURONU LOGISTYCZNEGO (SIGMOIDY

RZUT POWIERZCHNI ODPOWIEDZI NA PŁĄSZCZYZNĘ X_1X_2