

Sieci Neuronowe, wsteczna propagacja błędów

Neuron sigmoidalny, uczenie

Uczenie **wag** za pomocą
zadanych wartości na wejściu i
wyjściu

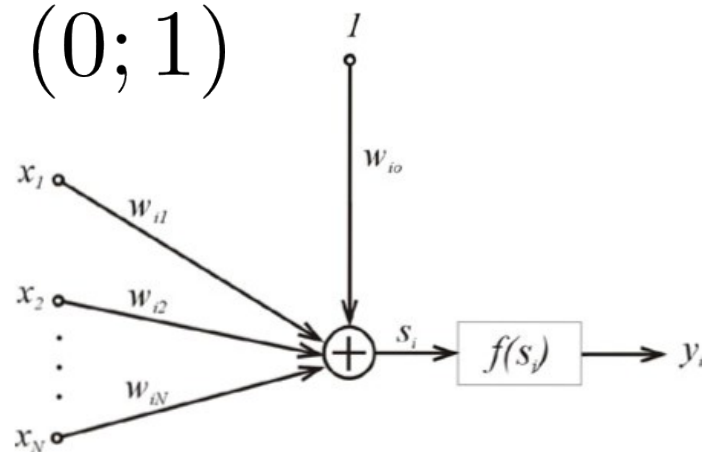
$$\Delta w_a = \mu(d - y) \frac{\partial y}{\partial w_a}$$

μ - współczynnik uczenia $\in (0; 1)$

d - wartość zadana

y - wartość otrzymana

$(d - y)$ - błąd (korekta)



Neuron sigmoidalny, uczenie

Uczenie **wag** za pomocą
zadanych wartości na wejściu i
wyjściu

$$\Delta w_a = \mu(d - y) \frac{\partial y}{\partial w_a}$$

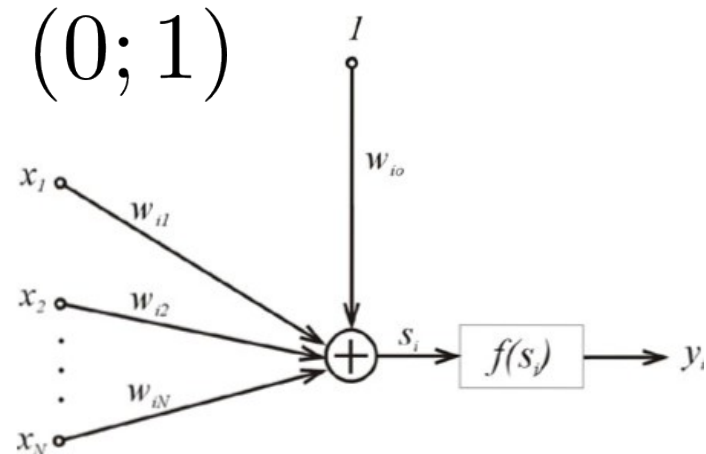
μ - współczynnik uczenia $\in (0; 1)$

d - wartość zadana

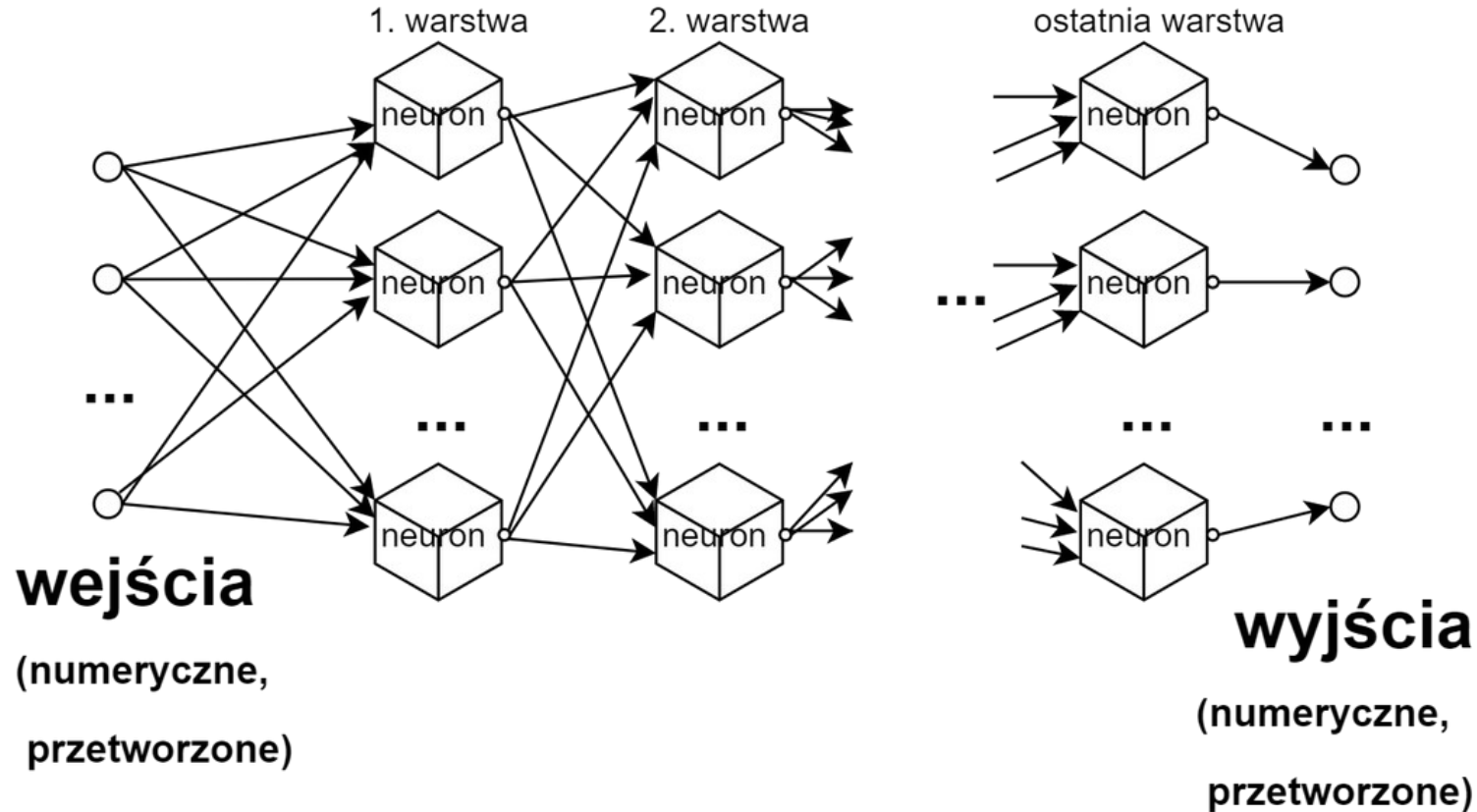
y - wartość otrzymana

$(d - y)$ - błąd (korekta)

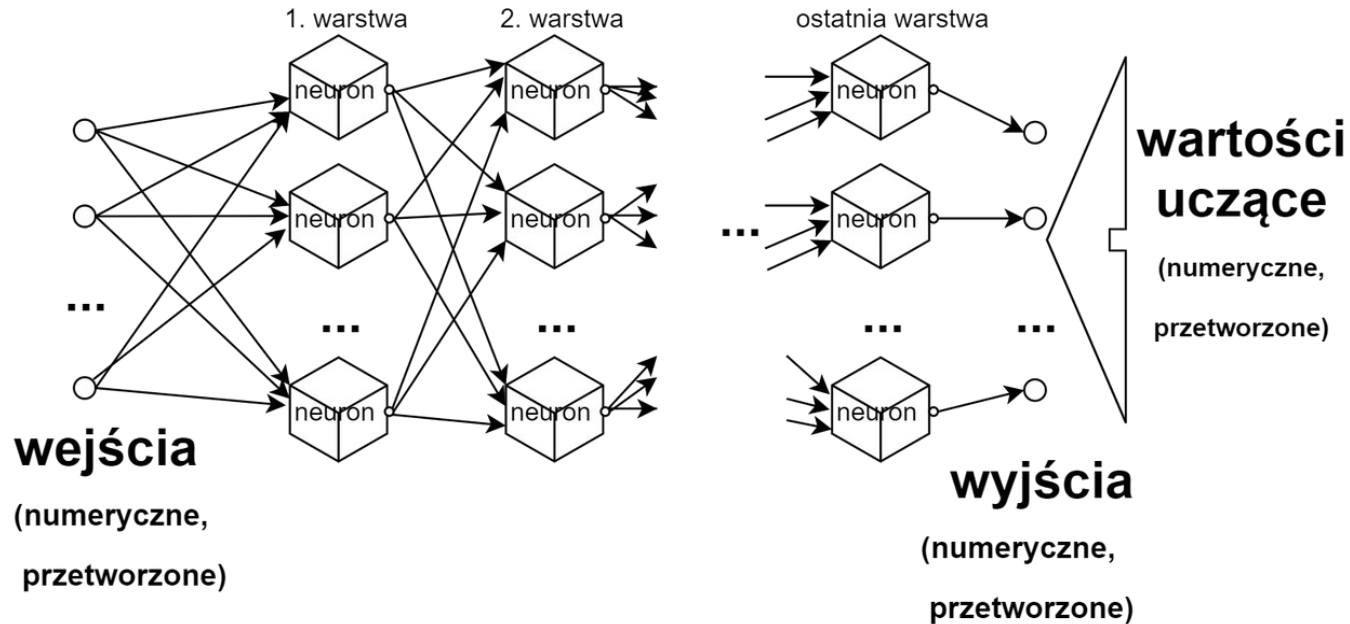
Tylko wagi (w) podlegają uczeniu



Sieć wielowarstwowa, wielowarstwowa, działanie



Sieć wielowarstwowa, wielowarstwowa, uczenie



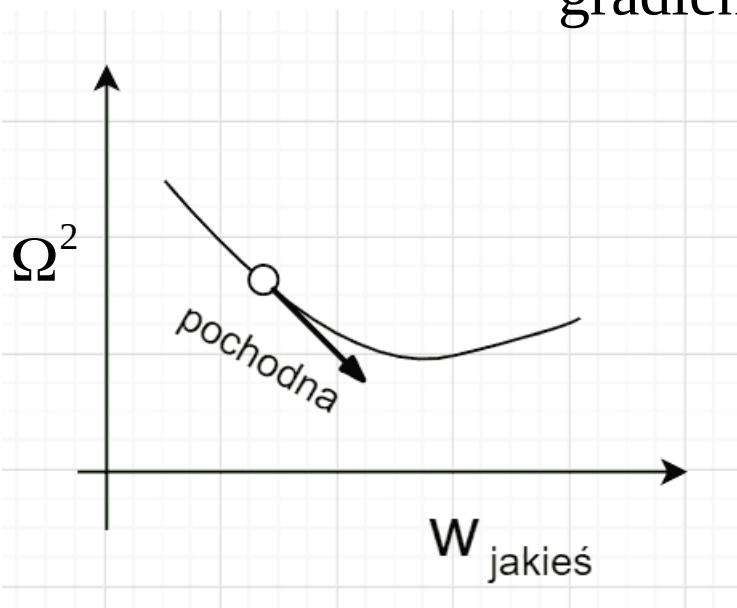
$$\Delta w_{jakies} = \mu \sum_{wyjscieNr} (d_{wyjscieNr} - y_{wyjscieNr}) \frac{\partial y_{wyjscieNr}}{\partial w_{jakies}}$$

$\mu = \text{parametr uczenia} ; \mu \in (0 ; 1)$

uczenie gradientowe

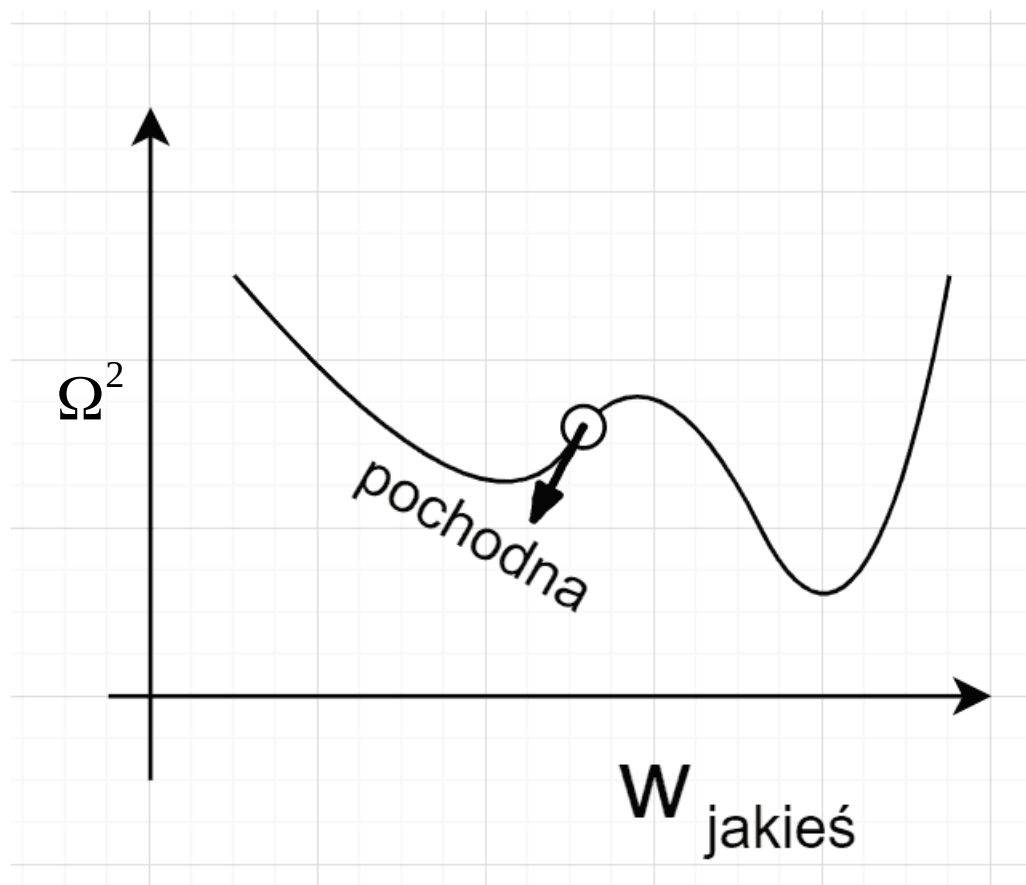
$$\text{suma błędów}^2 = \sum_{\text{wyjscieNr}} (d_{\text{wyjscieNr}} - y_{\text{wyjscieNr}})^2 = \Omega$$

$$\text{gradient błędu}^2 = \left[\frac{\partial \Omega}{\partial w_{\text{jakies1}}}, \frac{\partial \Omega}{\partial w_{\text{jakies2}}}, \dots \right]$$



- Zwykle działa
- Ciężko wyliczyć pochodne dla nietrywialnych sieci

uczenie gradientowe



- Czy również tym razem pochodna wskazuje na odpowiedni kierunek?
- Czy pochodna prawidłowo określa nie tylko kierunek, a „moc” korekty?
- Czy zmiana jednego parametru wpływa na pochodną drugiego?

Uczenie sieci, alg. makro

- Stwórz sieć, wylosuj wagi początkowe
- Przetwórz sygnały uczące i testowe
(normalizacja, „rozbicie atrybutów symbolicznych“)
- Ucz przez zadaną liczbę epok
 - Ucz sieć każdą próbką uczącą,
próbki uczące są wybierane w kolejności losowej

Epoka – proces pojedynczego uczenia sieci za pomocą całego zbioru próbek uczących

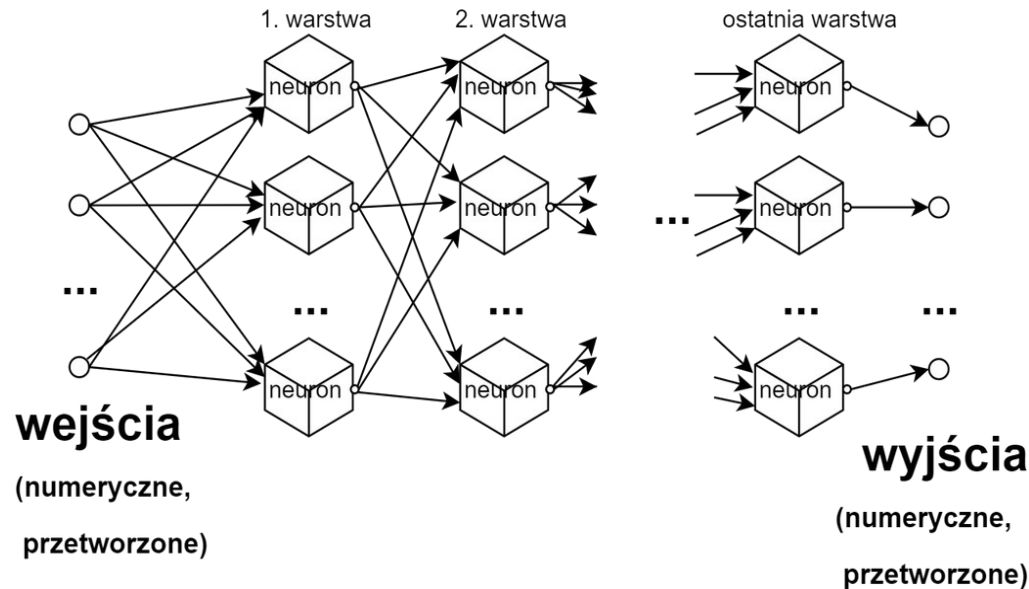
Wstępne przetwarzanie sygnałów

- Wartości numeryczne wejściowe znormalizuj
na przykład $\rightarrow [0; 1]$ albo wymuś stałe odchylenie std.
- Wartości numeryczne wyjściowe znormalizuj do zakresu wyjściowego neuronów
- Atrybuty symboliczne rozbij (prawie zawsze)
np. Atrybut symboliczny „jedna z m klas”
na m pseudoatrybuty numeryczne
np. Atrybut symboliczny „jedna z 2 klas” na binarny atr.

Wsteczna propagacja błędów, krok 1.

Wylicz wartości wyjściowe w kolejnych warstwach,

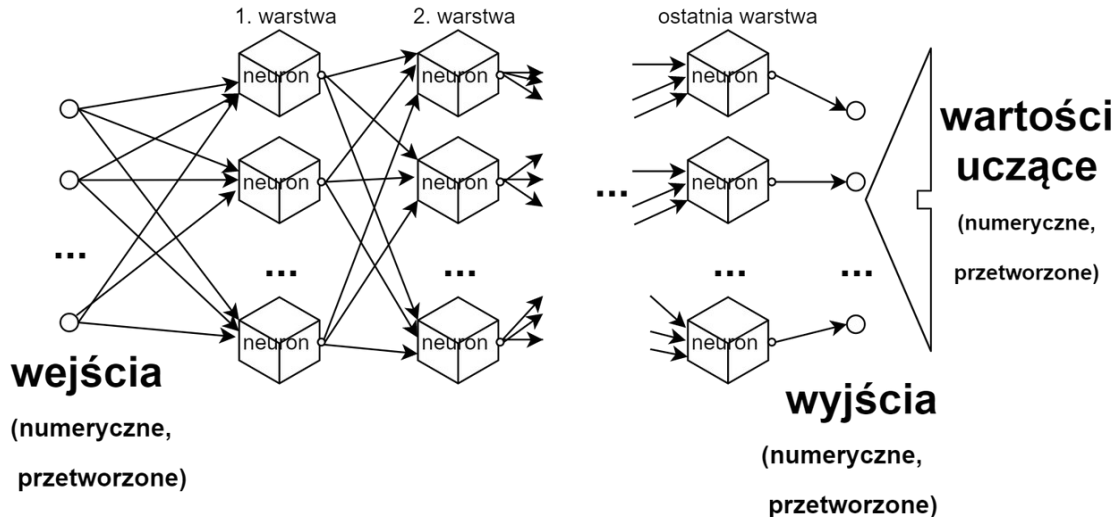
od warstwy pierwszej po ostatnią



Wsteczna propagacja błędów, krok 2.

Wylicz błędy ($d-y$) i poprawki ($\Delta w, \delta$) w kolejnych warstwach

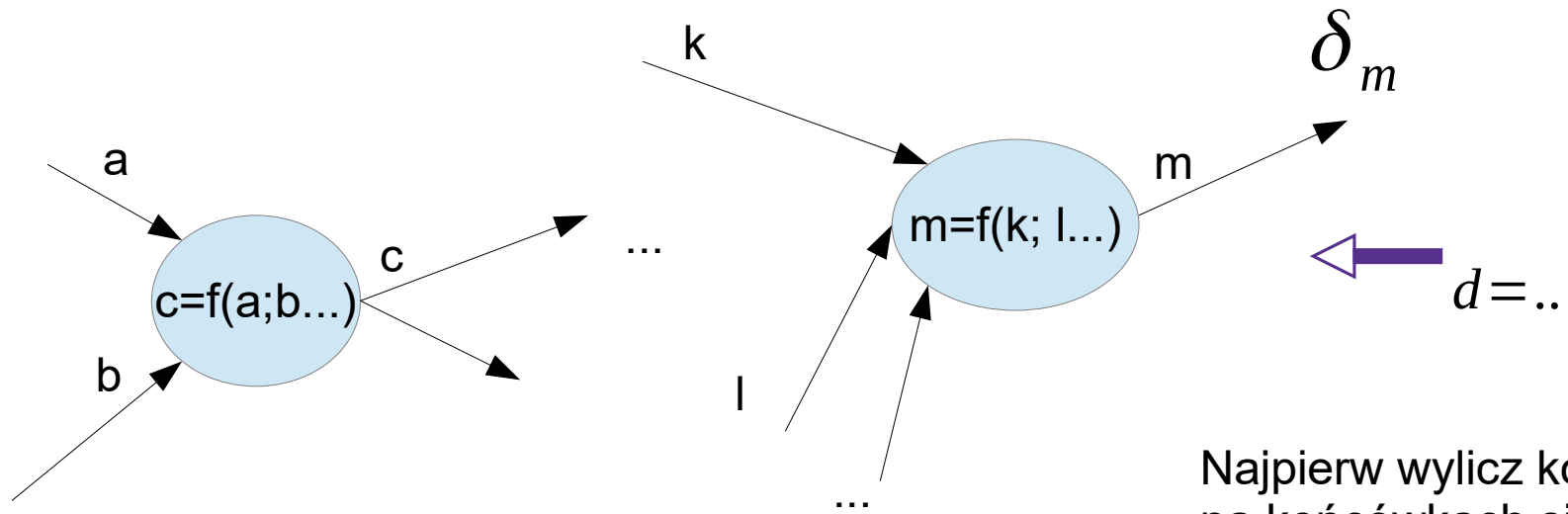
← od ostatniej warstwy do pierwszej (stąd nazwa propagacja wsteczna)



Wsteczna propagacja błędów, krok 2.

Wylicz błędy ($d-y$) i poprawki ($\Delta w, \delta$) w kolejnych warstwach

← od ostatniej warstwy do pierwszej (stąd nazwa propagacja wsteczna)



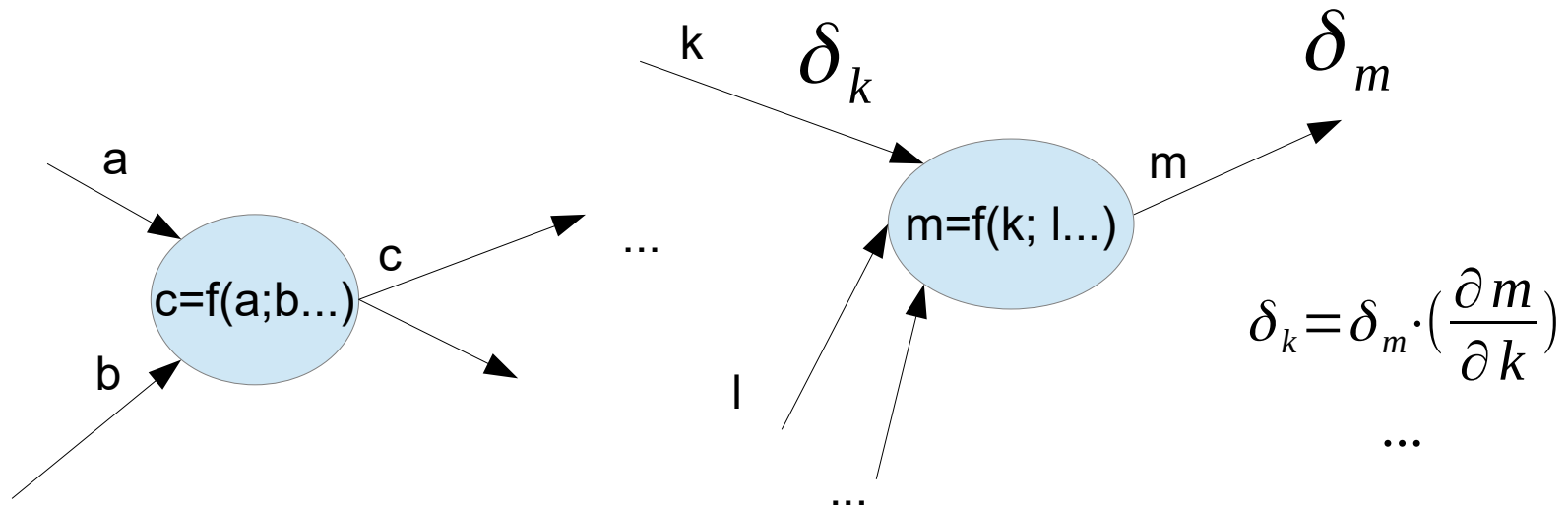
Najpierw wylicz korekty
na końcówkach sieci

$$\delta_{\text{tegoWyjścia}} = \mu \cdot (d_{\text{tegoWyjścia}} - \text{wyjście}_{\text{tegoWyjścia}})$$

Wsteczna propagacja błędów, krok 2.

Wylicz błędy (d-y) i poprawki ($\Delta w, \delta$) w kolejnych warstwach

od ostatniej warstwy do pierwszej (stąd nazwa propagacja wsteczna)



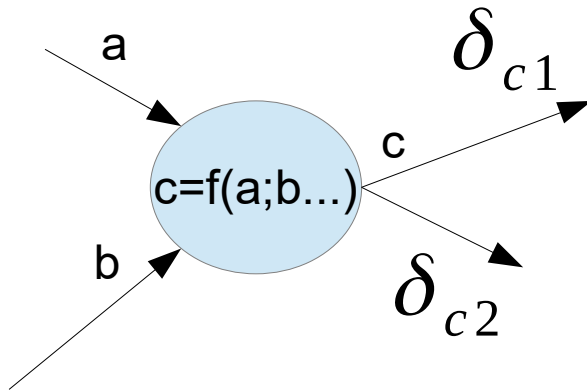
Później wszędzie propaguj
Korekty wstecz

Wsteczna propagacja błędów, krok 2.

Wylicz błędy (d-y) i poprawki ($\Delta w, \delta$) w kolejnych warstwach



od ostatniej warstwy do pierwszej (stąd nazwa propagacja wsteczna)



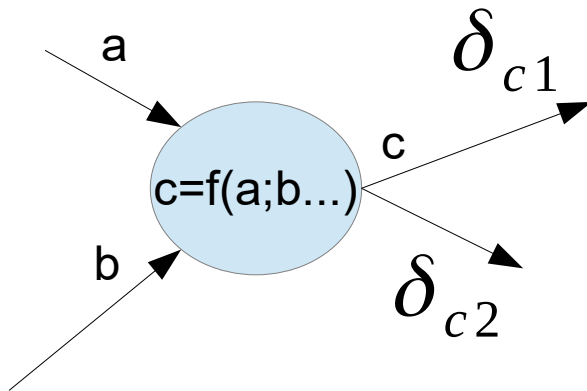
$$\delta_c = \delta_{c1} + \delta_{c2}$$

Gdyby do jednego wyjścia propagowane było wiele poprawek, to trzeba je zsumować

Wsteczna propagacja błędów, krok 2.

Wylicz błędy (d-y) i poprawki ($\Delta w, \delta$) w kolejnych warstwach

← od ostatniej warstwy do pierwszej (stąd nazwa propagacja wsteczna)



$$\delta_c = \delta_{c1} + \delta_{c2}$$

$$\delta_a = \delta_c \cdot \left(\frac{\partial c}{\partial a} \right),$$

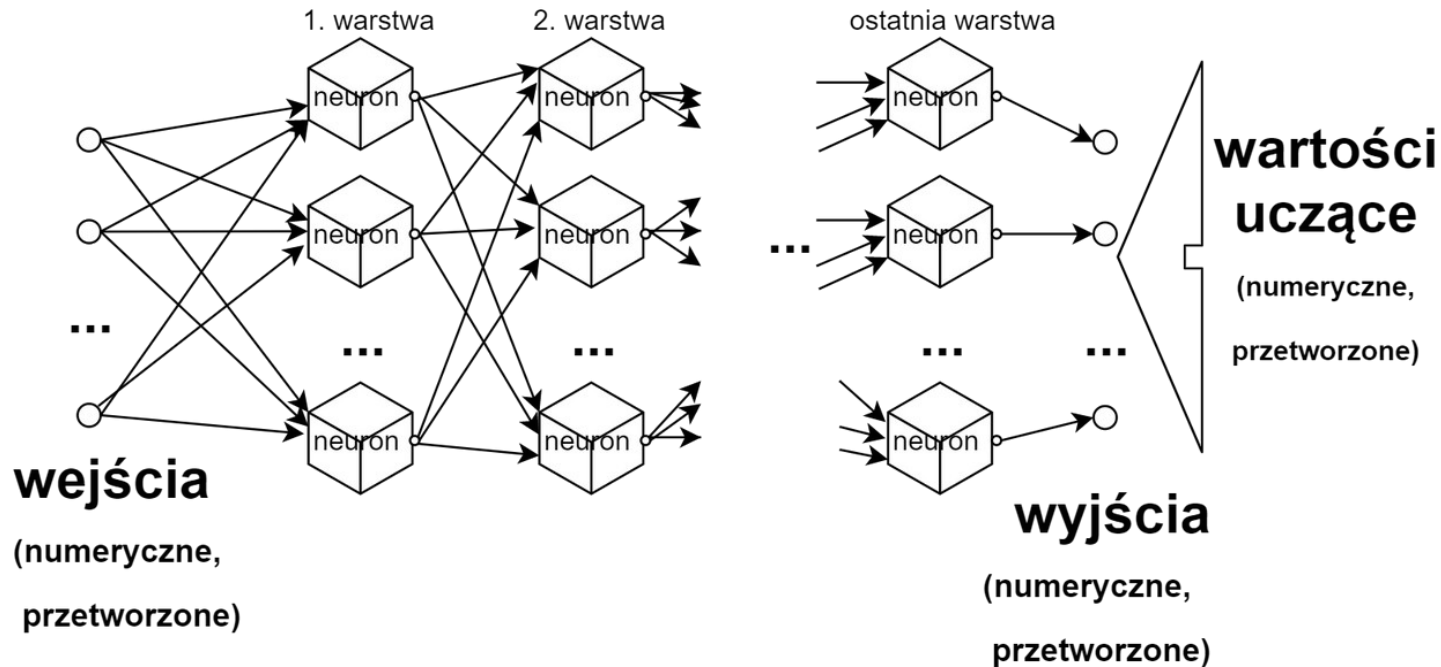
$$\delta_b = \delta_c \cdot \left(\frac{\partial c}{\partial b} \right)$$

...

Gdyby do jednego wyjścia propagowane było wiele poprawek, to trzeba je zsumować, a reszta działa normalnie

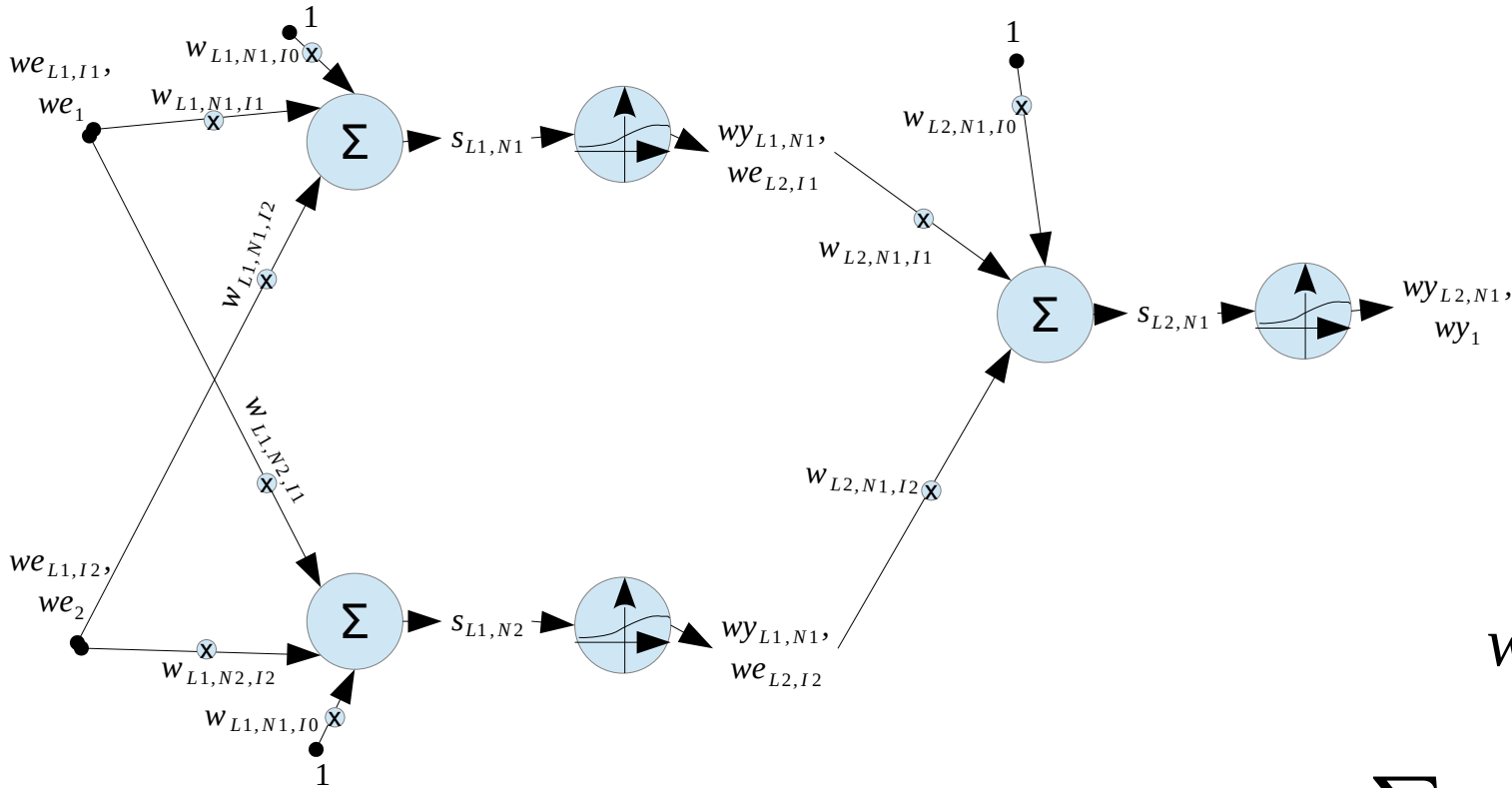
Wsteczna propagacja błędów, krok 3.

Wprowadź poprawki (Δw) w każdym z parametrów (każdej wadze)



Sieć neuronowa 2-2-1 uni-pol, sig. problem XOR

parametr $\beta > 0$
domyślnie $\beta = 1$

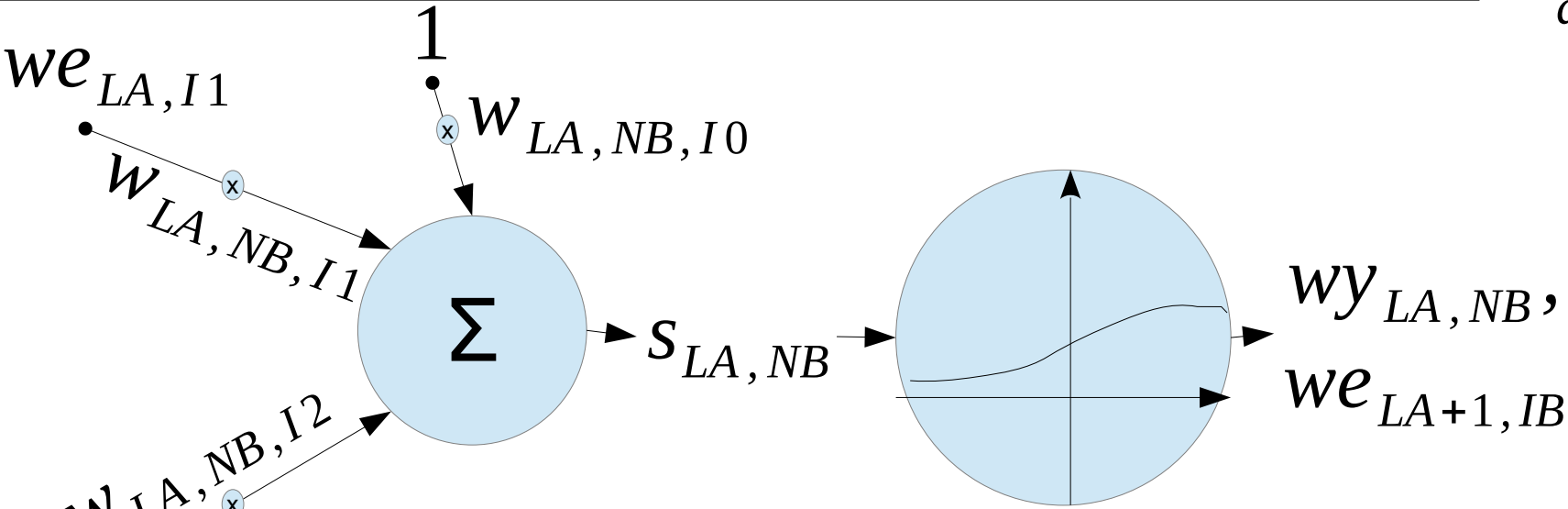


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot s_{LA,NB}}}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

Sieć neuronowa 2-2-1 uni-pol, sig. problem XOR

parametr $\beta > 0$
domyślnie $\beta = 1$

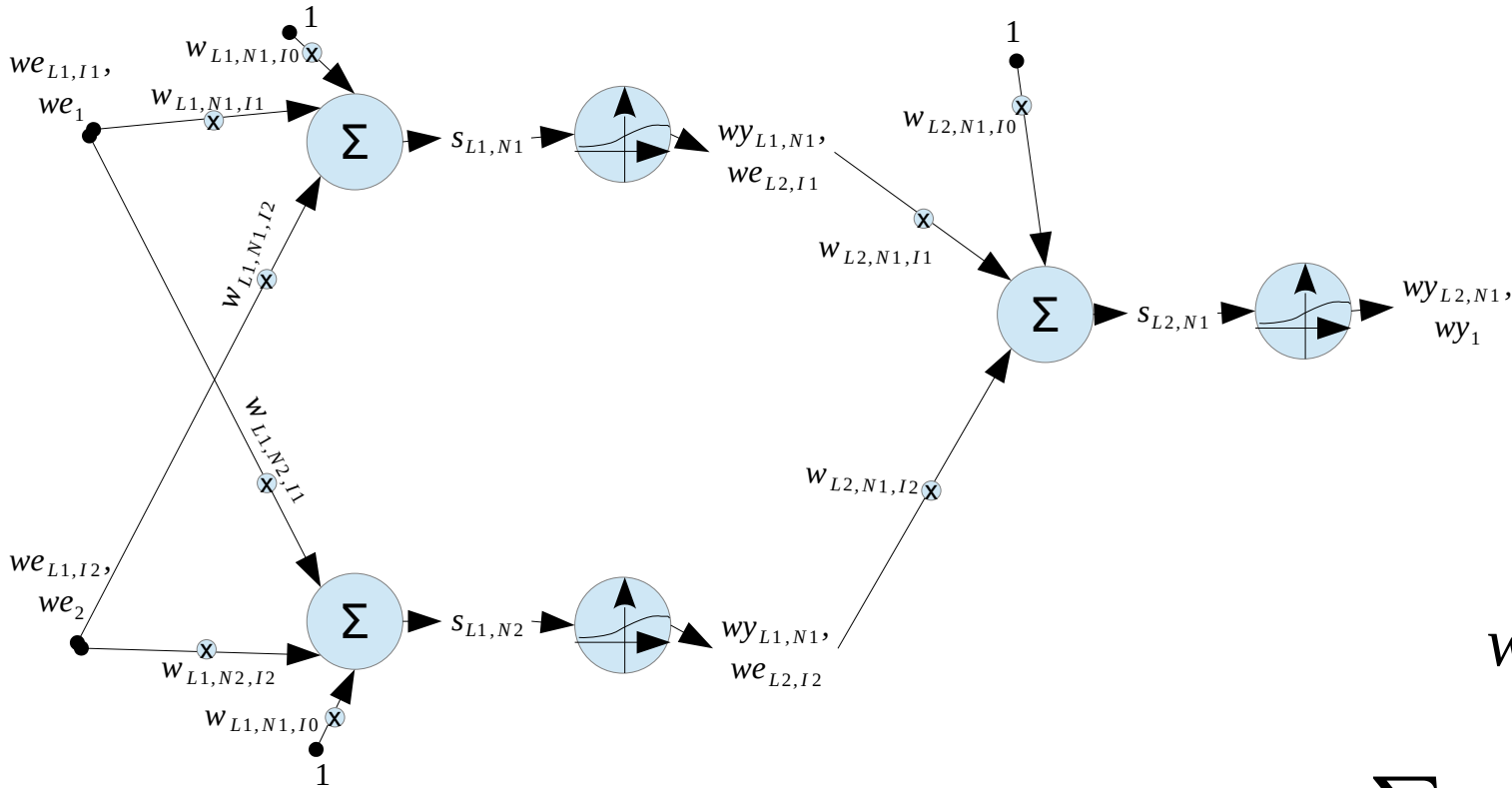


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

Sieć neuronowa 2-2-1 uni-pol, sig. problem XOR

parametr $\beta > 0$
domyślnie $\beta = 1$

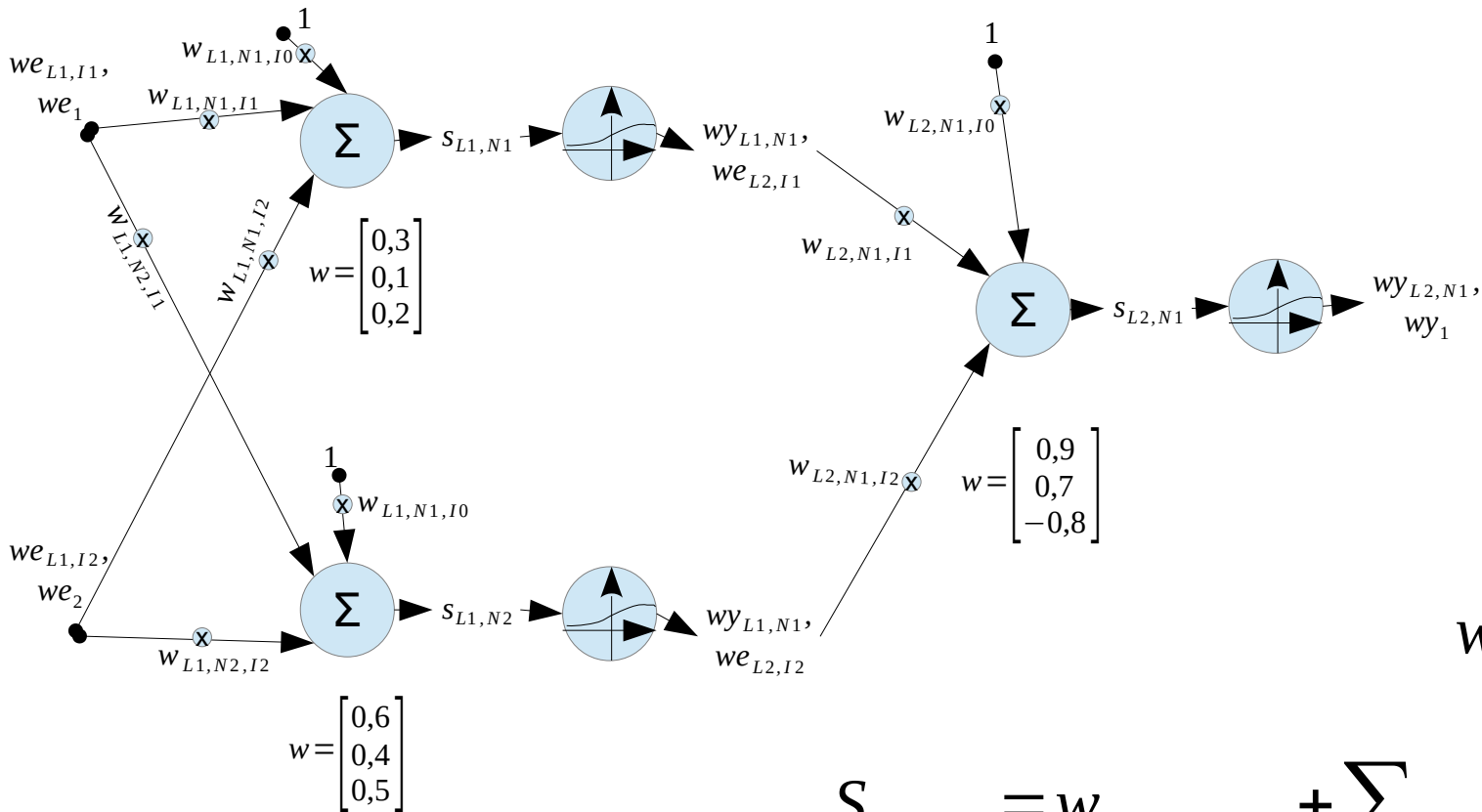


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, start, losowanie wag

$$\beta = 1$$

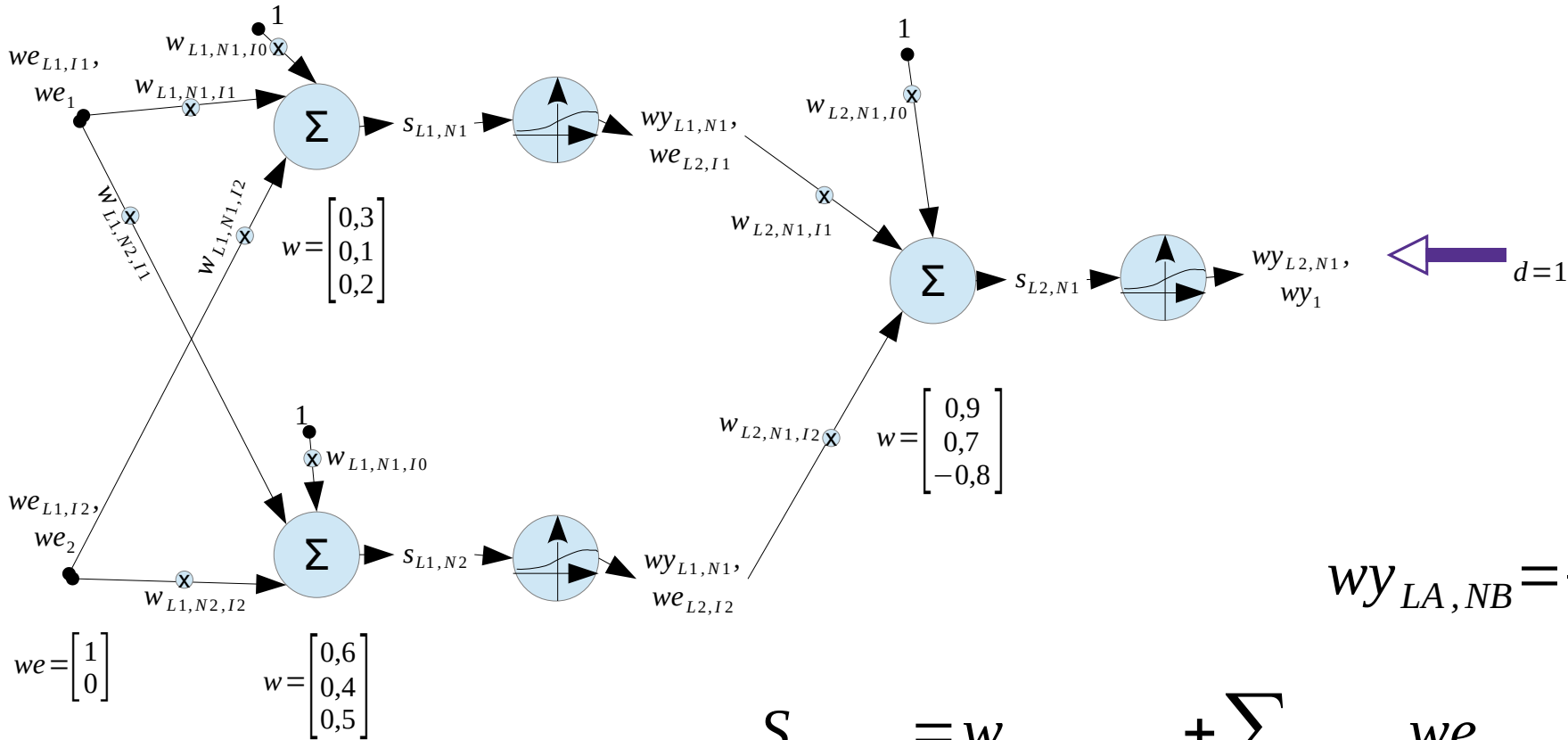


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$\beta = 1$

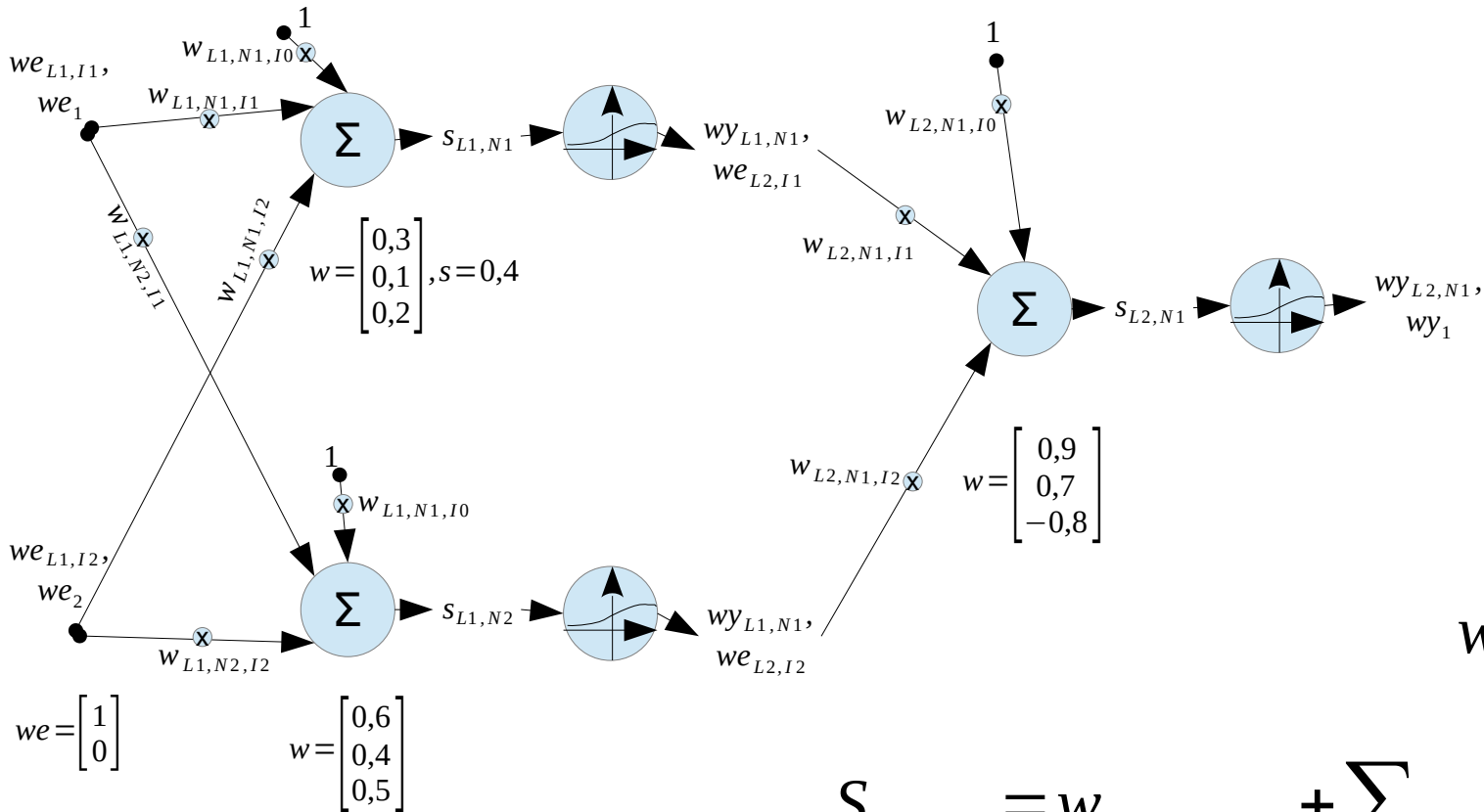


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$$\beta = 1$$

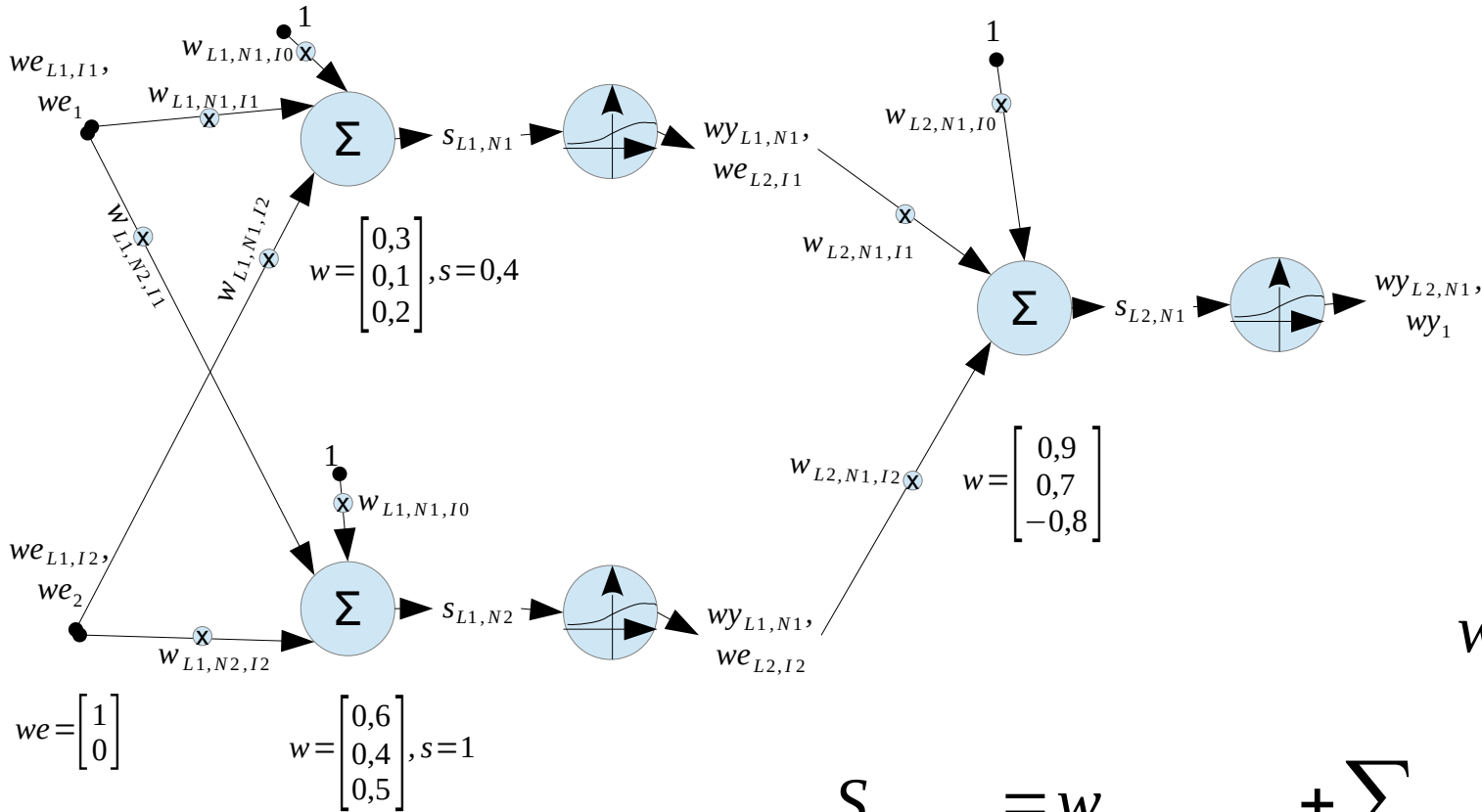


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$$\beta = 1$$

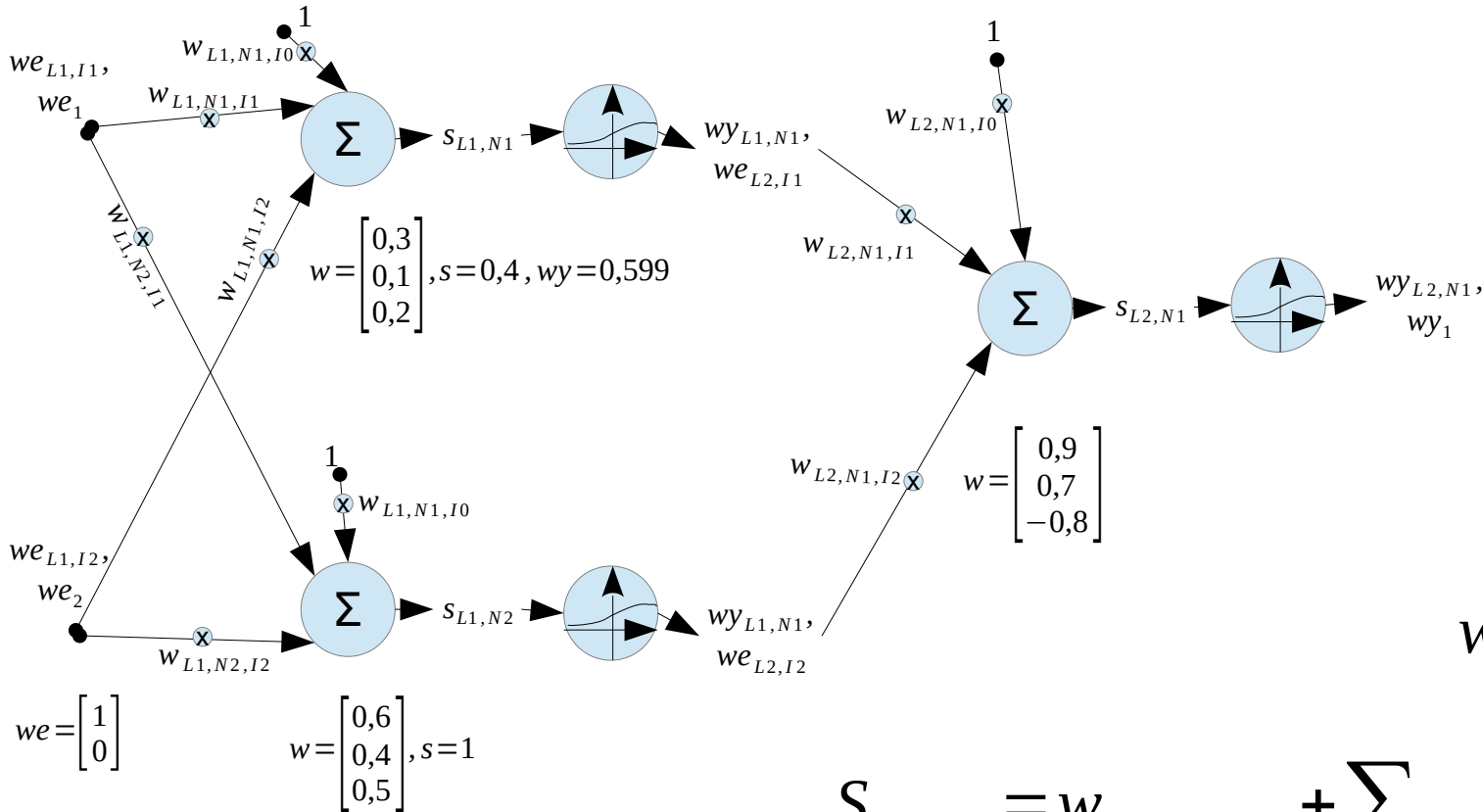


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$$\beta = 1$$

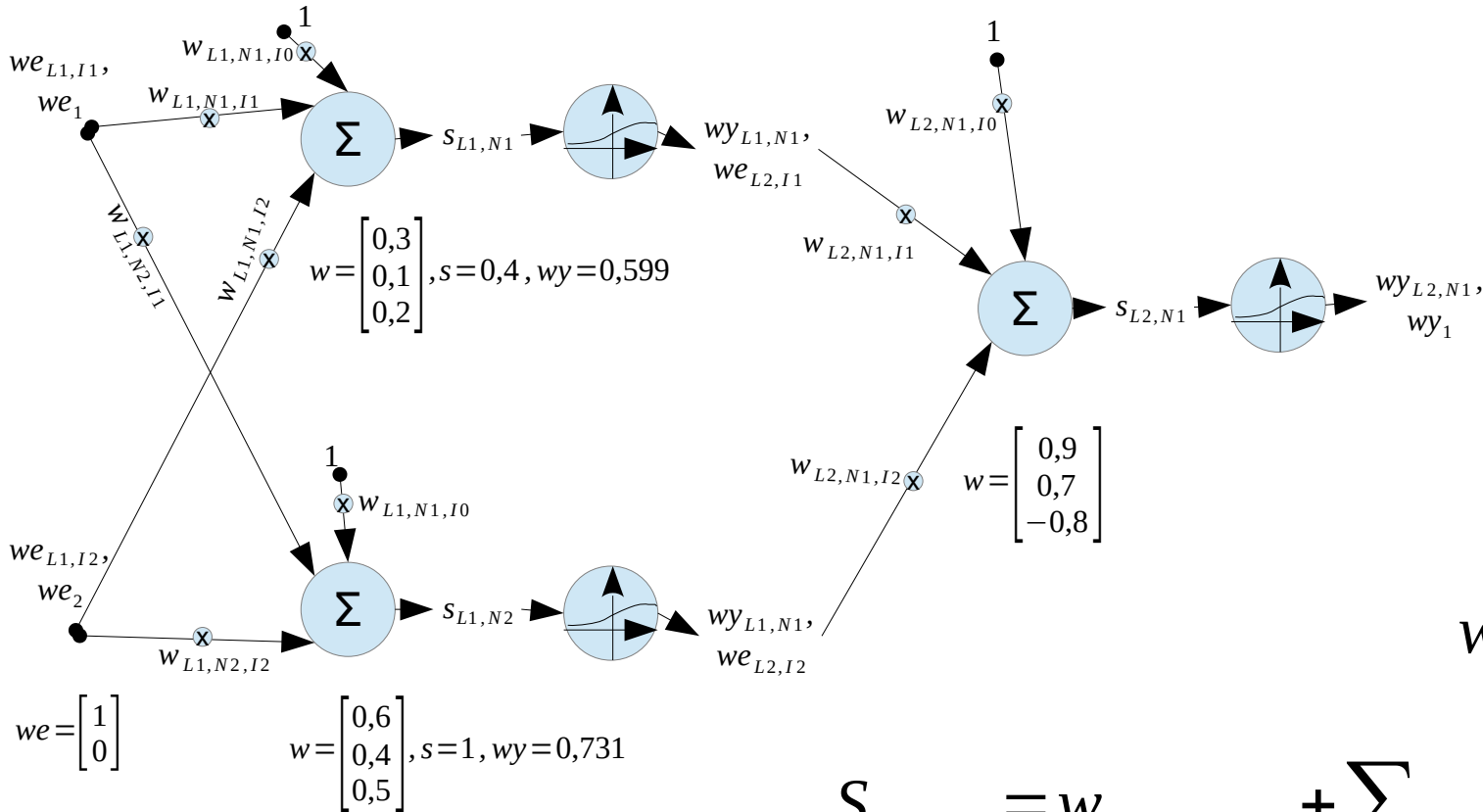


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$$\beta = 1$$

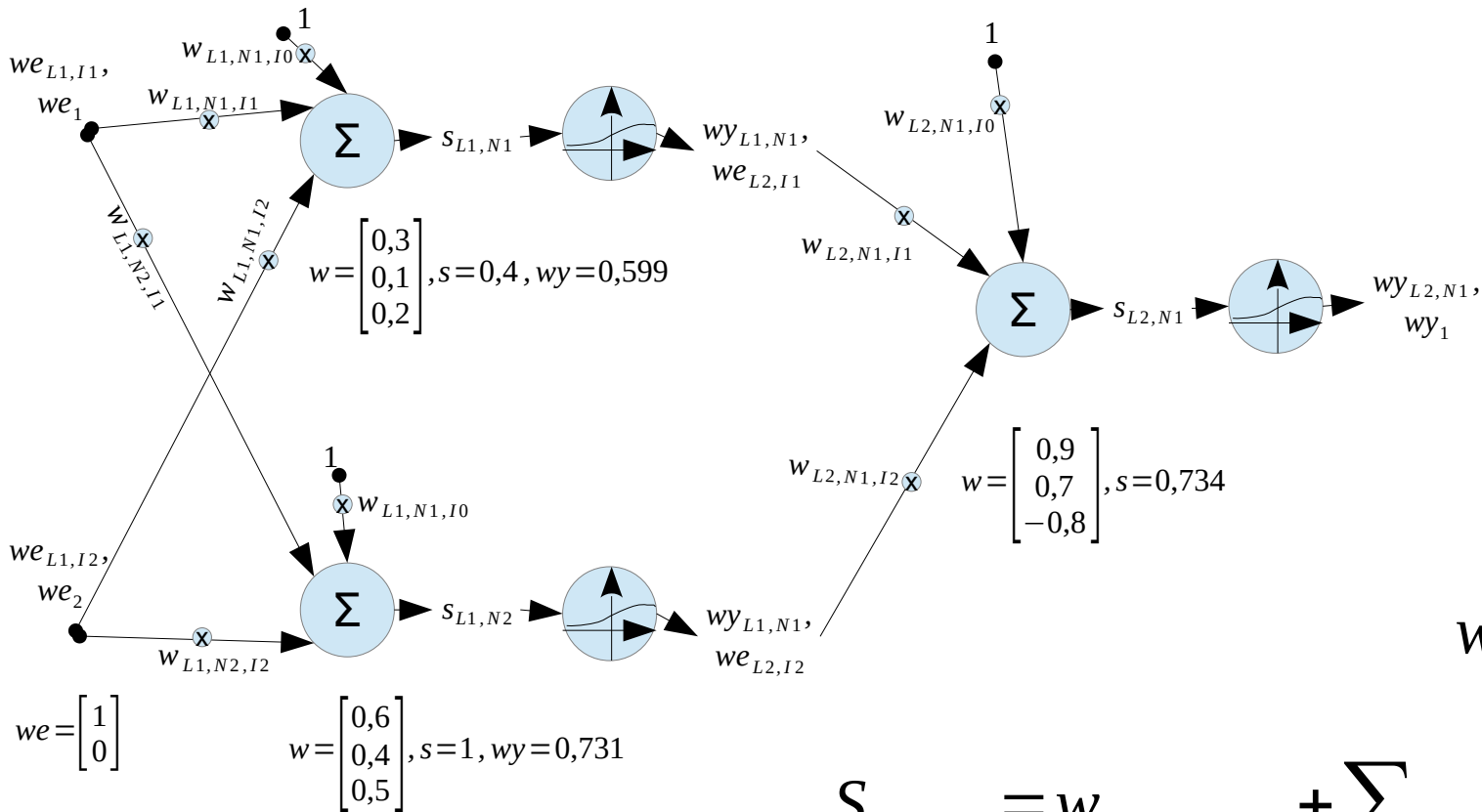


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wyliczanie wyjścia, próbka [1 0 1]

$$\beta = 1$$

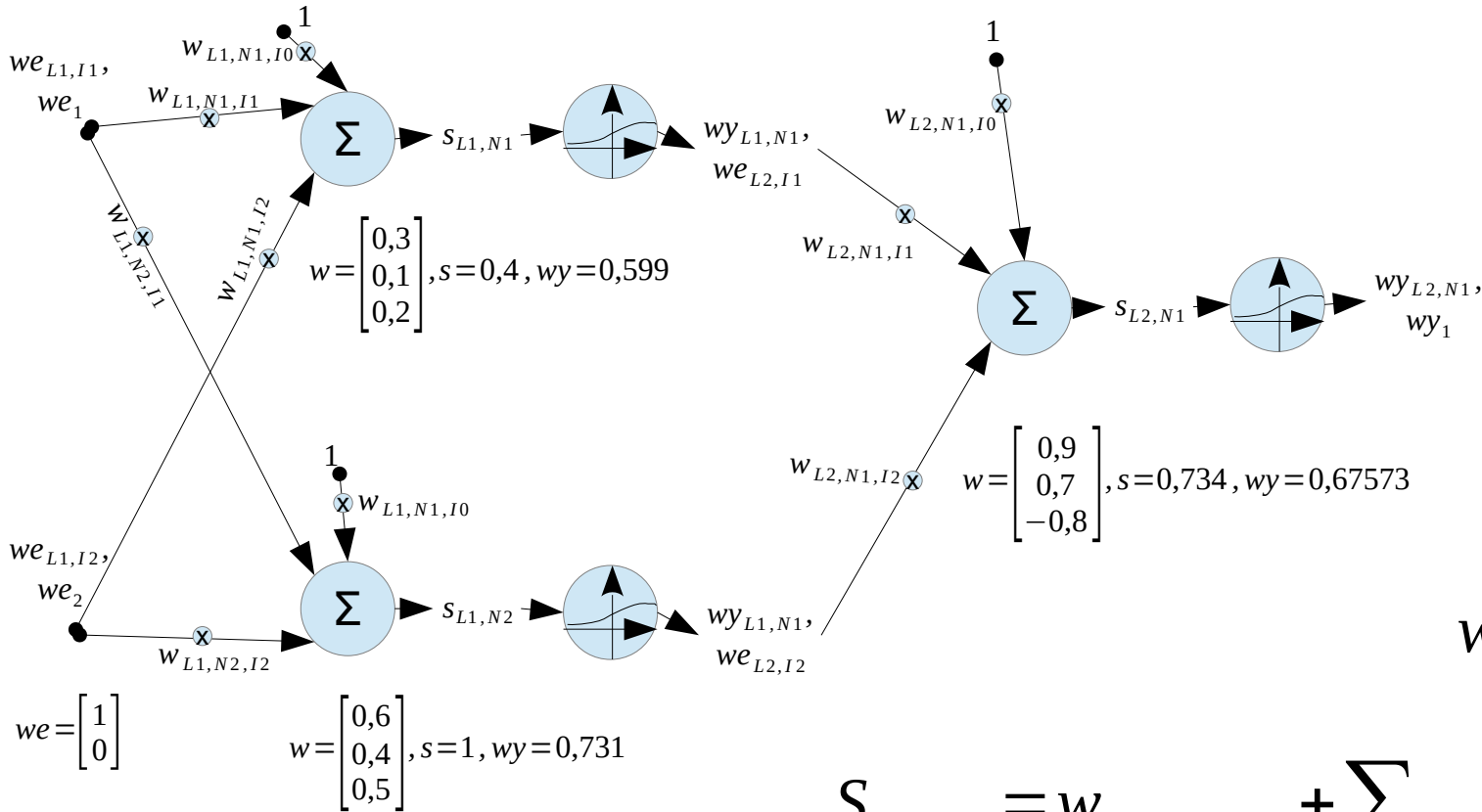


$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot s_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, wylczenie wyjścia, próbka [1 0 1]

$$\beta = 1$$



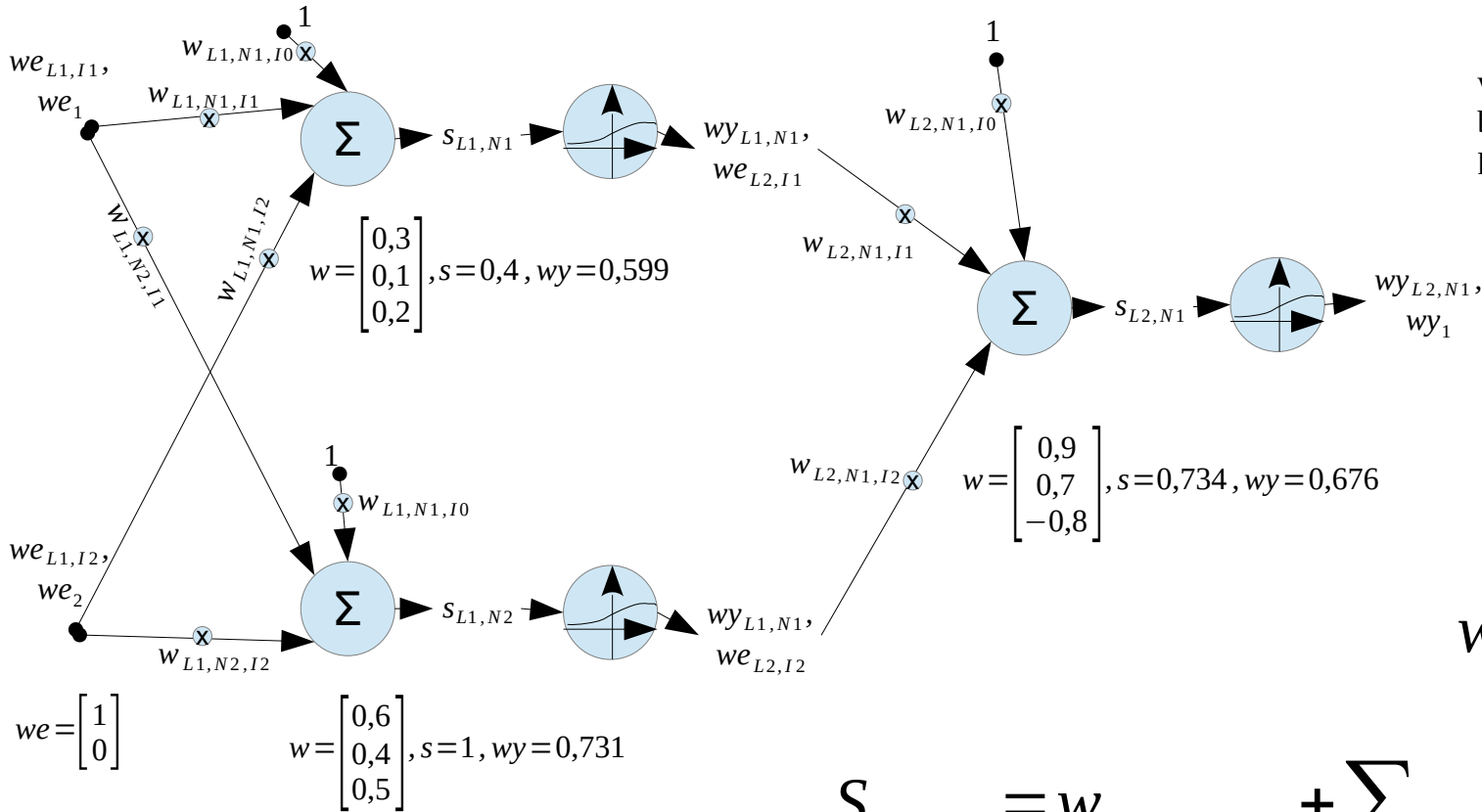
$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot s_{LA,NB}}}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, próbka [1 0 1] wyliczenie błędu i poprawki

$$\beta = 1$$

$$\mu = 0,1$$



wartość zadana = $d_1 = 1$
 błąd = $(d_1 - wy_1) = 0,324$
 korekta = $\mu * (d_1 - wy_1) = \delta_{wy,L2,N1} = 0,032$



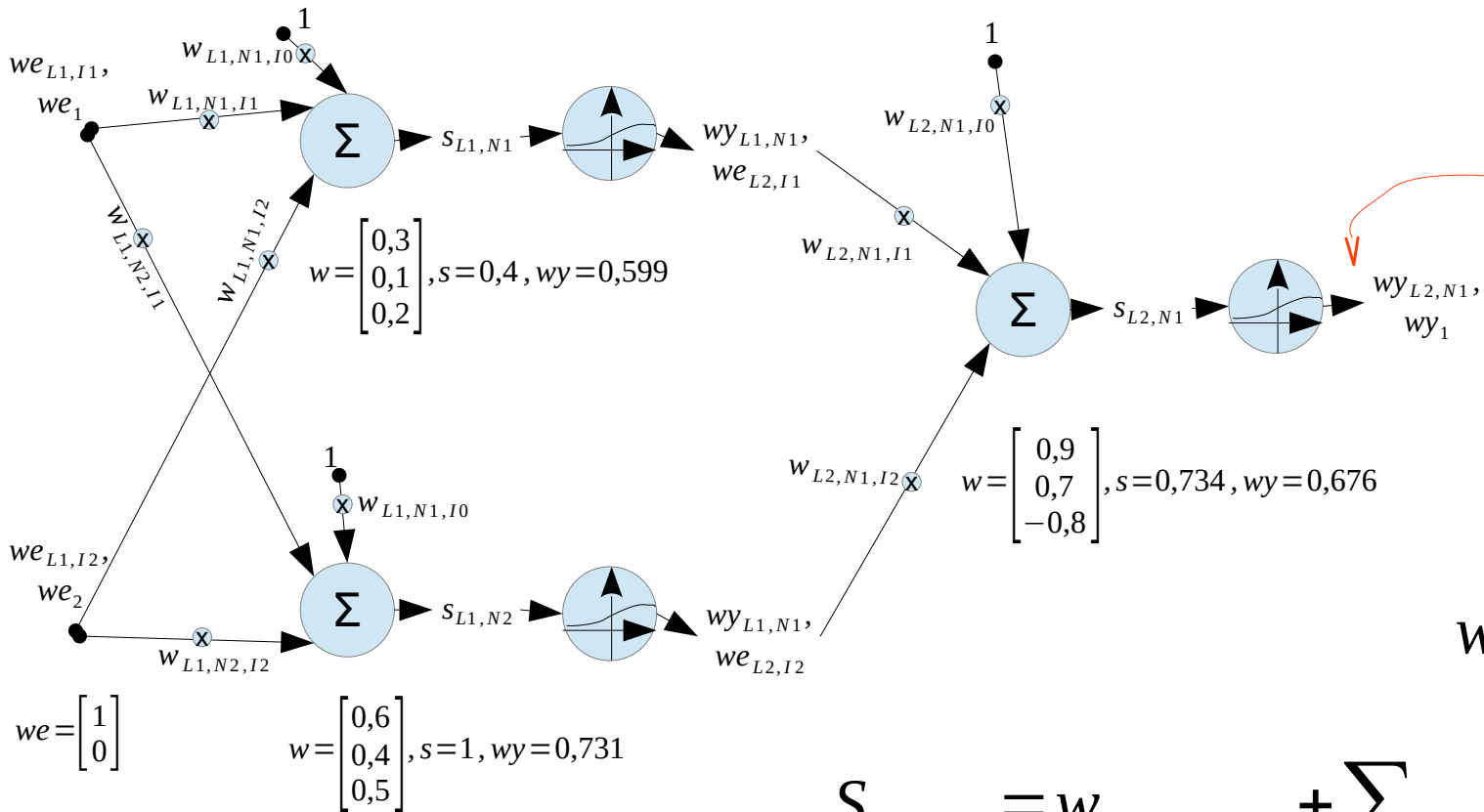
$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot s_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, próbka [1 0 1] wyliczenie błędu i poprawki

$$\beta = 1$$

$$\mu = 0,1$$



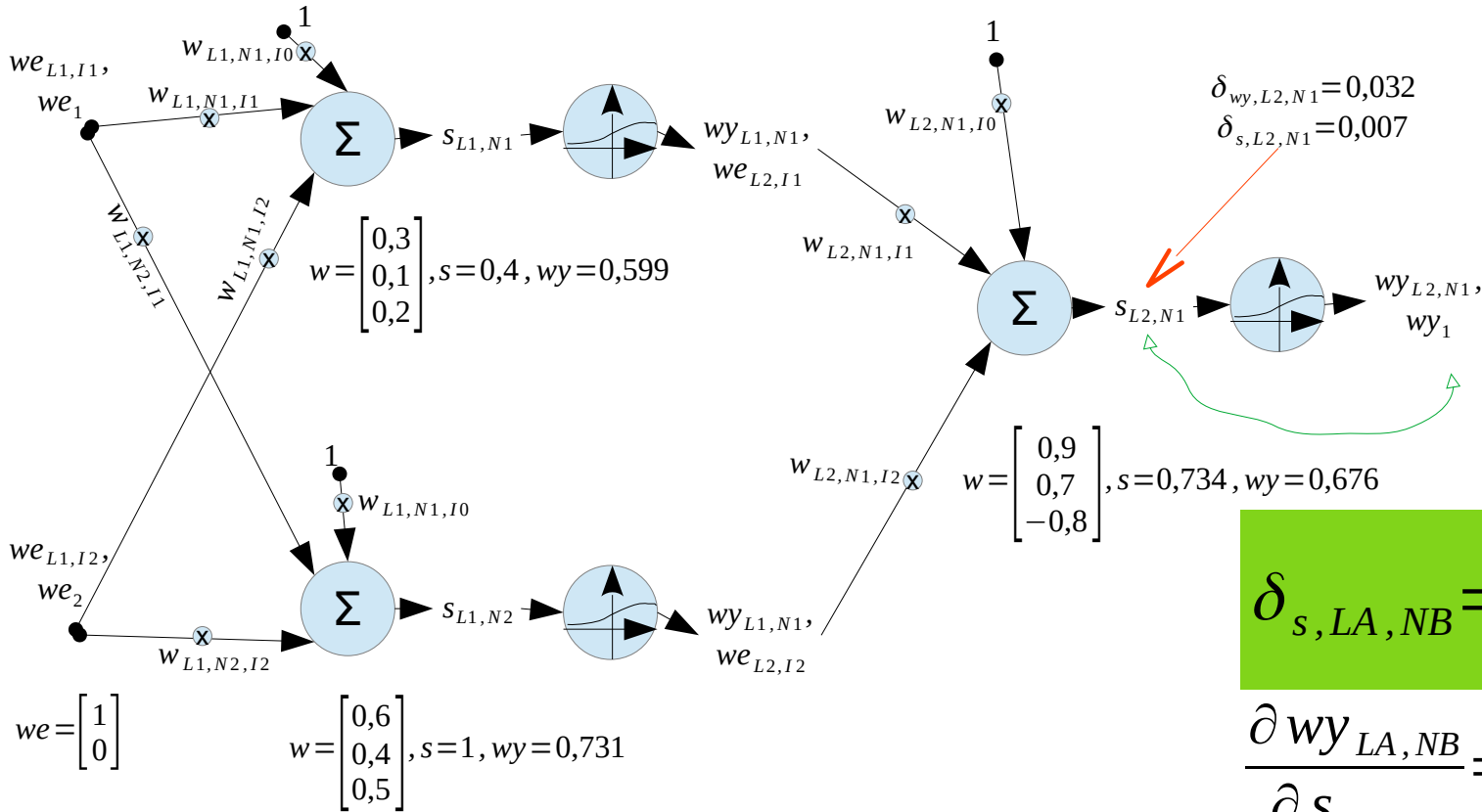
wartość zadana = $d_1 = 1$
 błąd = $(d_1 - wy_1) = 0,324$
 korekta = $\mu * (d_1 - wy_1) = \delta_{wy,L2,N1} = 0,032$

$$wy_{LA,NB} = \frac{1}{1 + e^{-\beta \cdot S_{LA,NB}}}$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1, \dots} we_{LA,I1} * w_{LA,NB,I1}$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



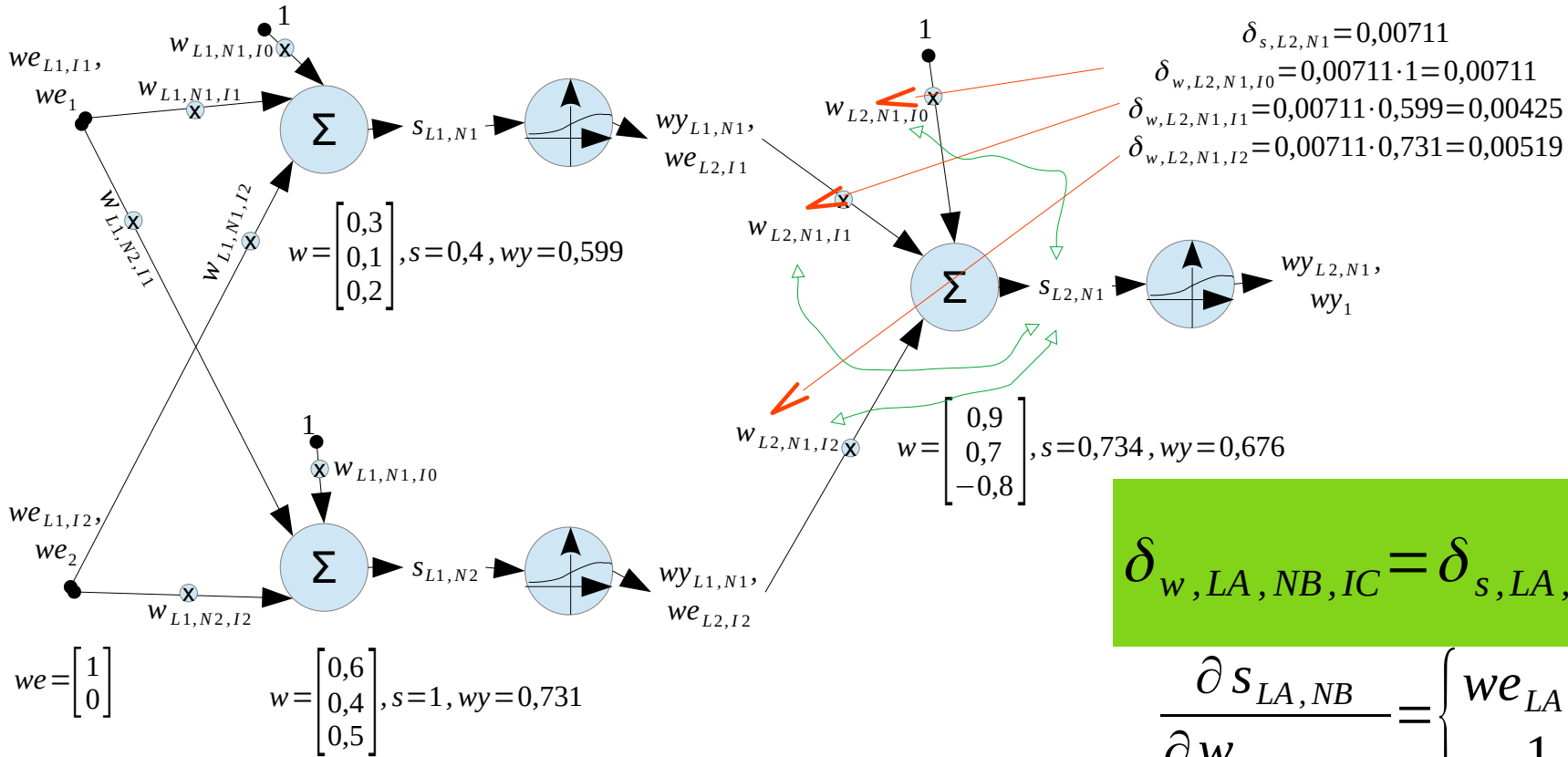
$$\delta_{s, LA, NB} = \delta_{wy, LA, NB} \cdot \frac{\partial wy_{LA, NB}}{\partial s_{LA, LB}}$$

$$\frac{\partial wy_{LA, NB}}{\partial s_{LA, LB}} = \beta \cdot wy_{LA, NB} \cdot (1 - wy_{LA, NB})$$

$$S_{LA, NB} = w_{LA, NB, I0} + \sum_{i=1}^n we_{LA, Ii} * w_{LA, NB, Ii} \quad wy_{LA, NB} = 1 / (1 + e^{-\beta \cdot S_{LA, NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



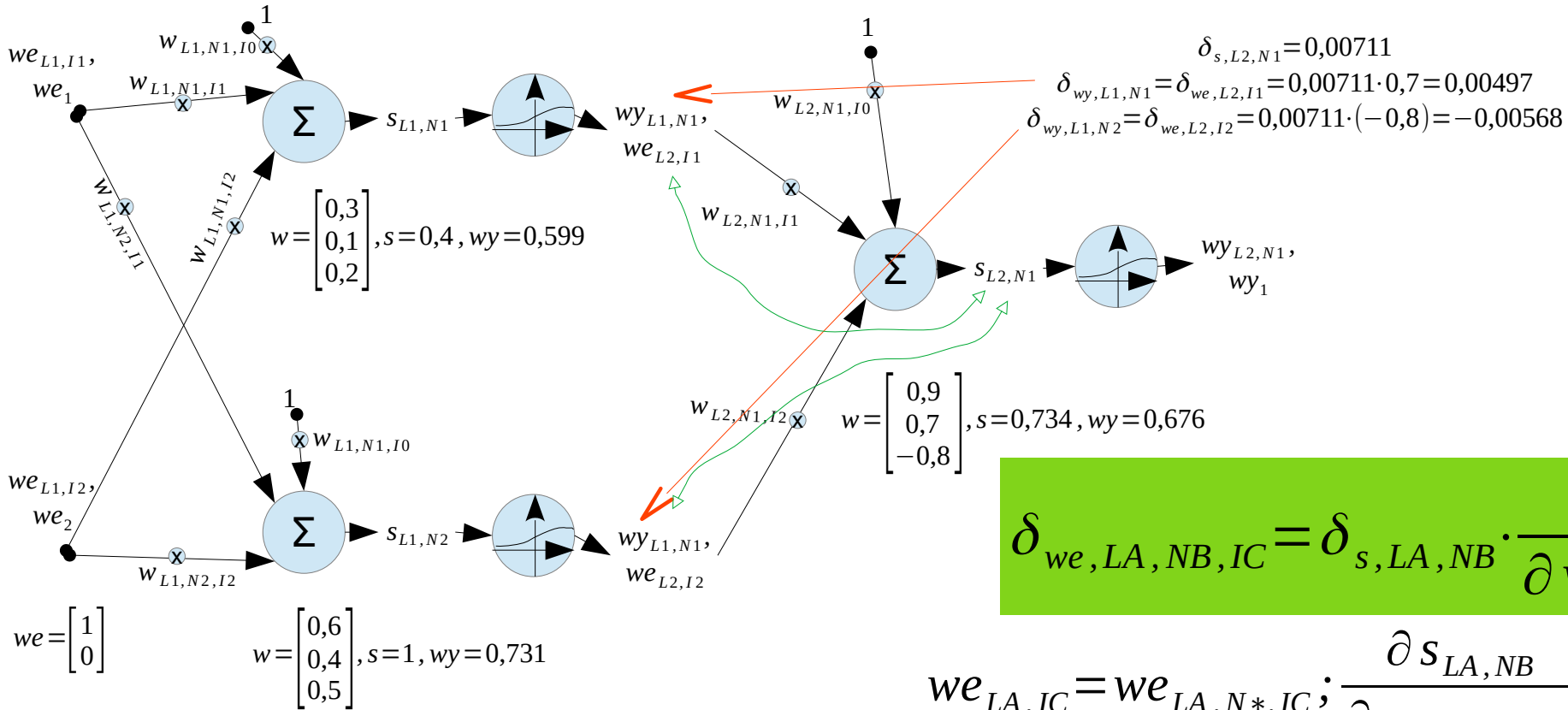
$$\delta_{w,LA,NB,IC} = \delta_{s,LA,NB} \cdot \frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}}$$

$$\frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}} = \begin{cases} we_{LA,IC} & \text{jeśli } C > 0 \\ 1 & \text{jeśli } C = 0 \end{cases}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1} we_{LA,I1} * w_{LA,NB,I1} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot s_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



$$\delta_{we,LA,NB,IC} = \delta_{s,LA,NB} \cdot \frac{\partial s_{LA,NB}}{\partial we_{LA,NB,IC}}$$

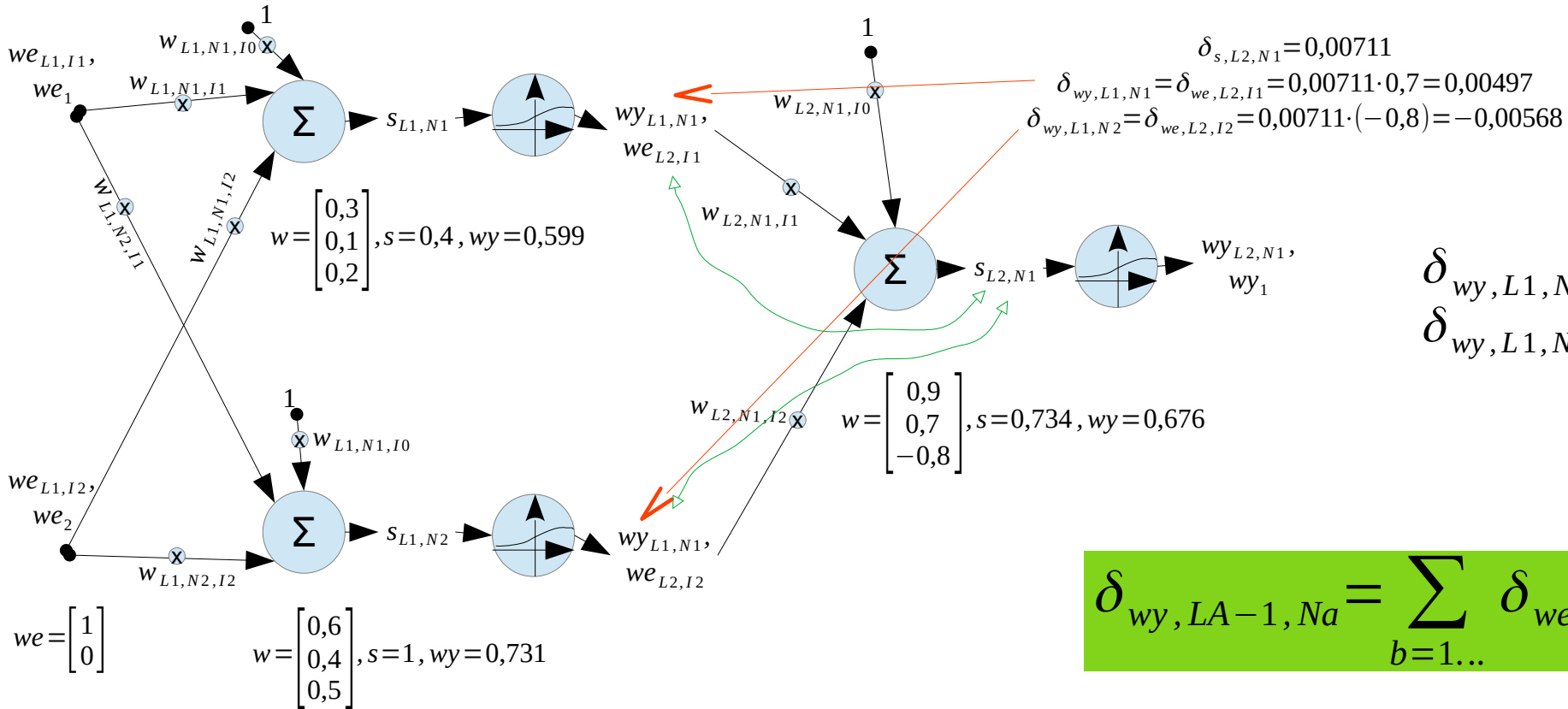
$$we_{LA,IC} = we_{LA,N*,IC}; \frac{\partial s_{LA,NB}}{\partial we_{LA,NB,IC}} = w_{LA,NB,IC}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1} we_{LA,I1} * w_{LA,NB,I1} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot s_{LA,NB}})$$

przykład XOR, próbka [1 0 1]

wsteczna propagacja korekty

$$\beta = 1$$



$$\delta_{wy,L1,N1} = \delta_{we,L2,I1}$$

$$\delta_{wy,L1,N2} = \delta_{we,L2,I2}$$

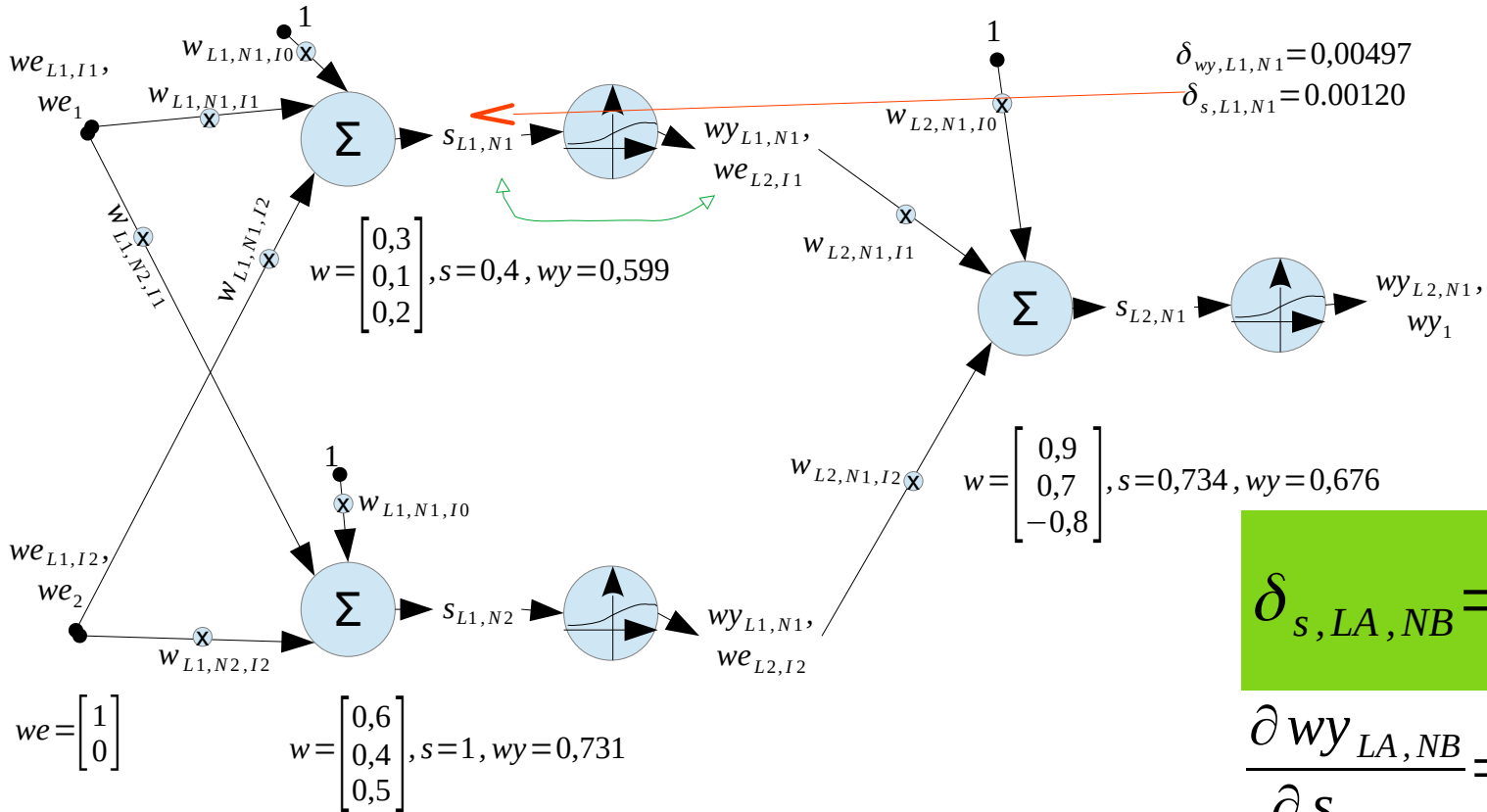
$$\delta_{wy,LA-1,Na} = \sum_{b=1..} \delta_{we,LA,Nb, Ia}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1} we_{LA,Ii} * w_{LA,NB,Ii}$$

$$wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot s_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



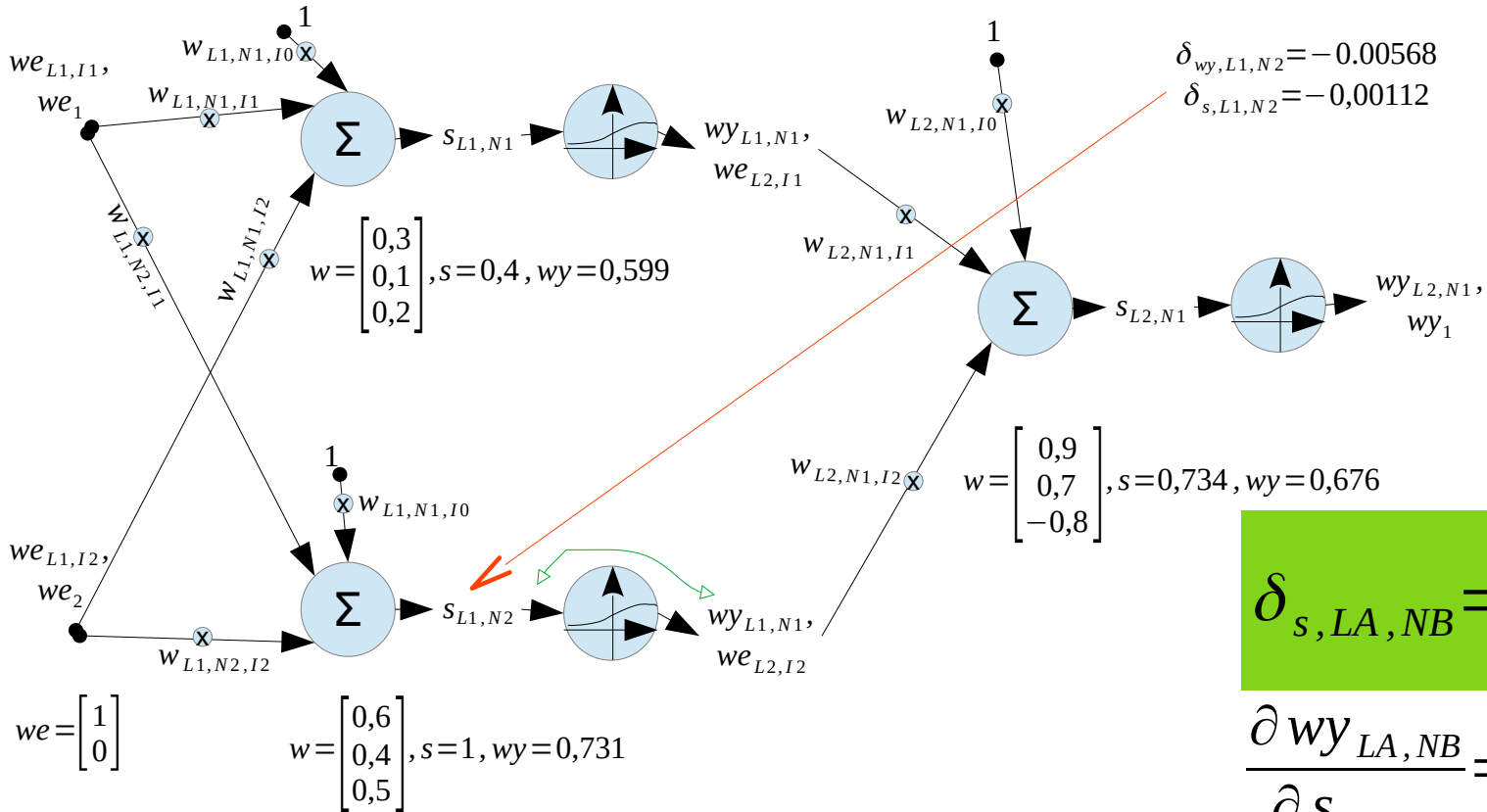
$$\delta_{s,LA,NB} = \delta_{wy,LA,NB} \cdot \frac{\partial wy_{LA,NB}}{\partial s_{LA,LB}}$$

$$\frac{\partial wy_{LA,NB}}{\partial s_{LA,LB}} = \beta \cdot wy_{LA,NB} \cdot (1 - wy_{LA,NB})$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1}^n we_{LA,Ii} * w_{LA,NB,Ii} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot S_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$$\beta = 1$$



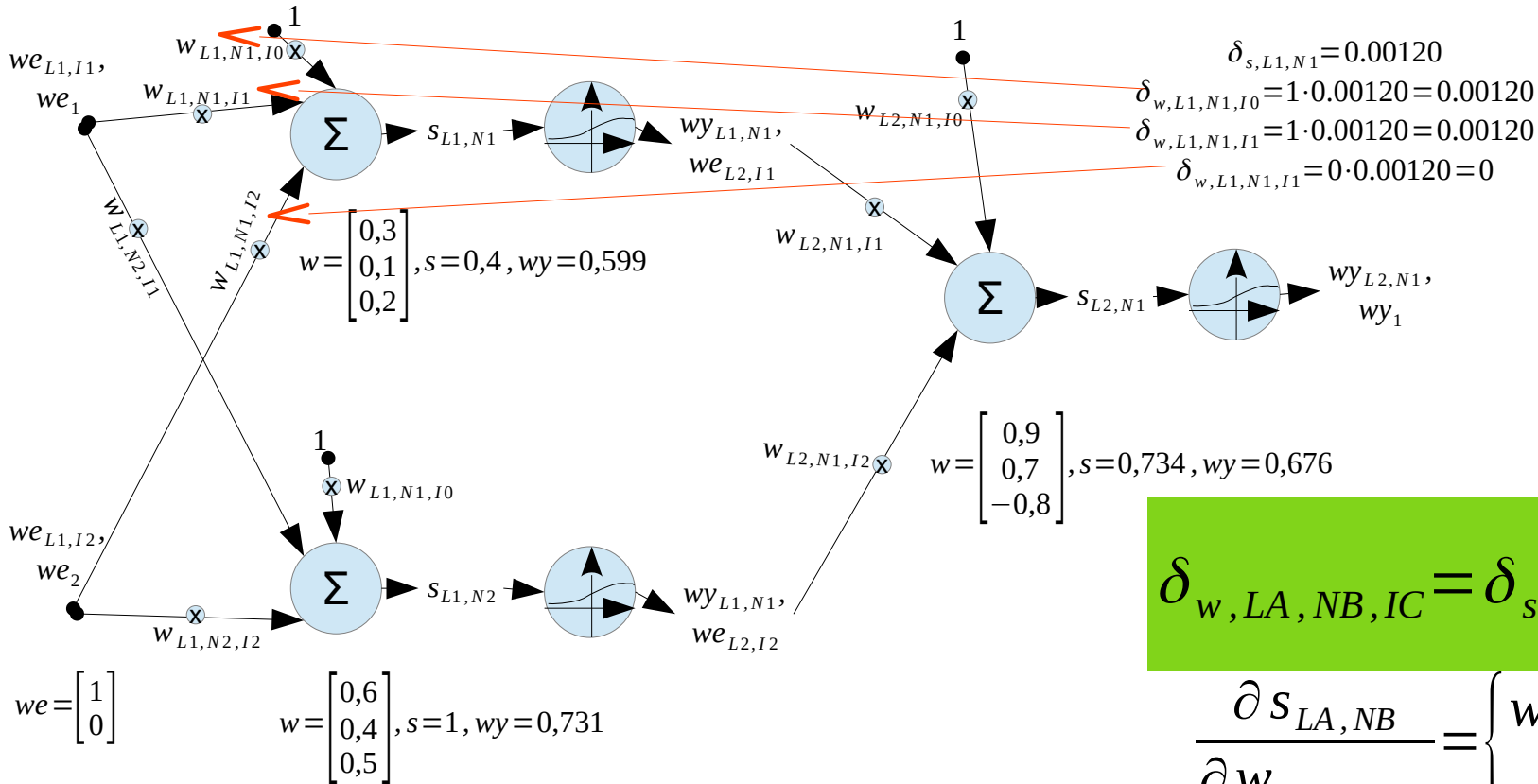
$$\delta_{s,LA,NB} = \delta_{wy,LA,NB} \cdot \frac{\partial wy_{LA,NB}}{\partial s_{LA,LB}}$$

$$\frac{\partial wy_{LA,NB}}{\partial s_{LA,LB}} = \beta \cdot wy_{LA,NB} \cdot (1 - wy_{LA,NB})$$

$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1}^n we_{LA,I1} * w_{LA,NB,I1} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot S_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



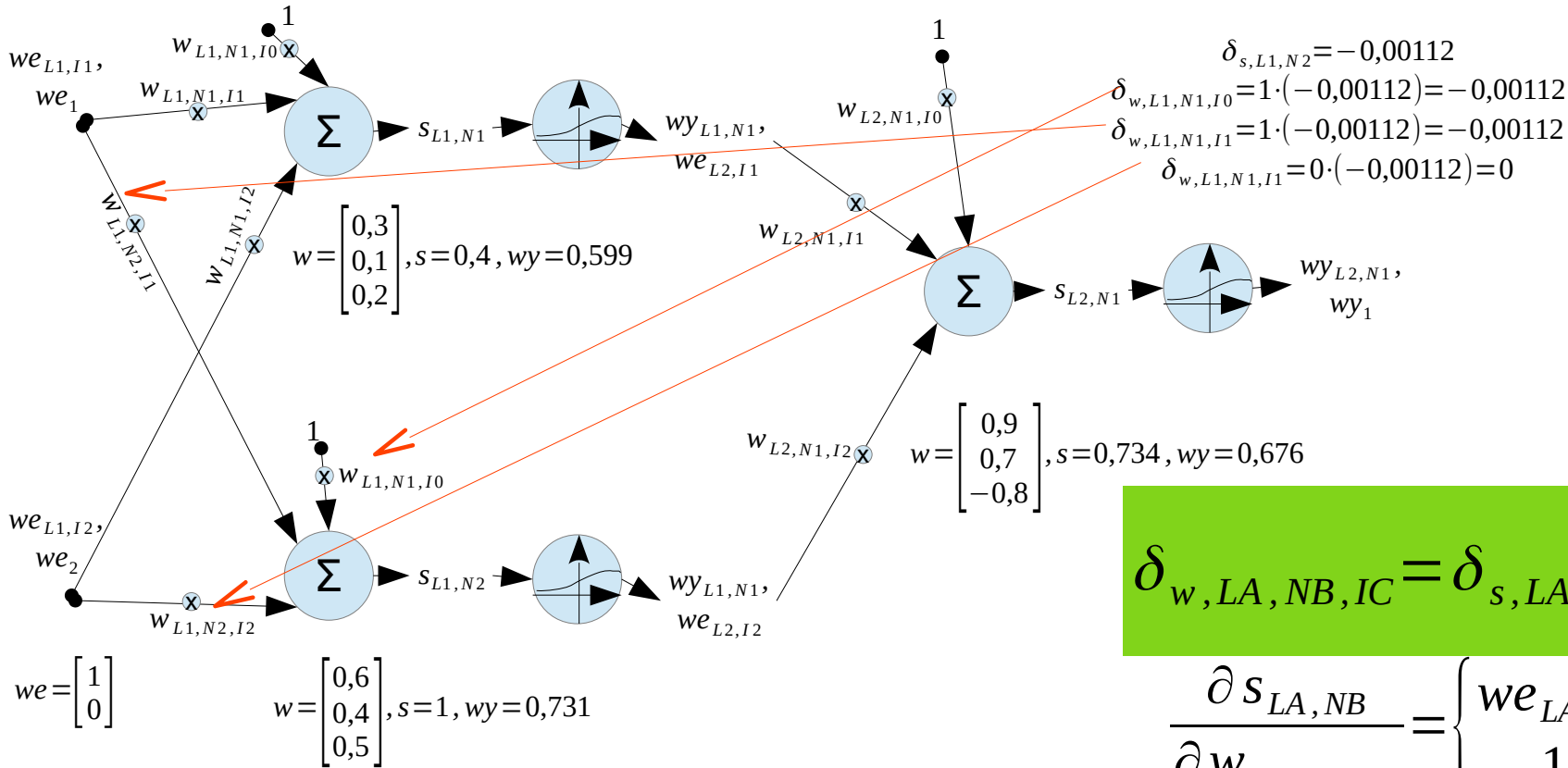
$$\delta_{w,LA,NB,IC} = \delta_{s,LA,NB} \cdot \frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}}$$

$$\frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}} = \begin{cases} we_{LA,IC} & \text{jeśli } C > 0 \\ 1 & \text{jeśli } C = 0 \end{cases}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1}^n we_{LA,Ii} * w_{LA,NB,Ii} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot s_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wsteczna propagacja korekty

$\beta = 1$



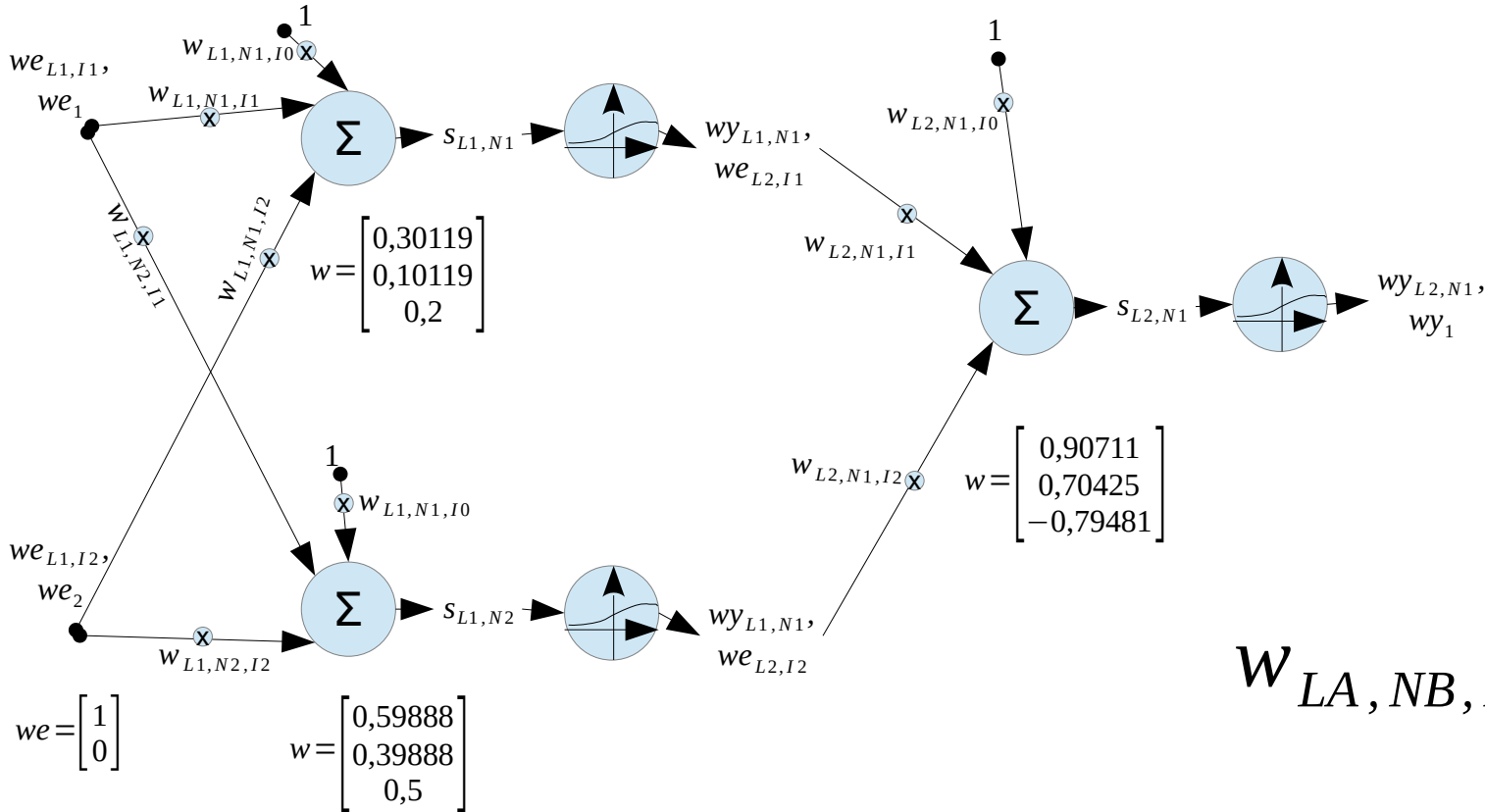
$$\delta_{w,LA,NB,IC} = \delta_{s,LA,NB} \cdot \frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}}$$

$$\frac{\partial s_{LA,NB}}{\partial w_{LA,NB,IC}} = \begin{cases} we_{LA,IC} & \text{jeśli } C > 0 \\ 1 & \text{jeśli } C = 0 \end{cases}$$

$$s_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1}^n we_{LA,I1} * w_{LA,NB,I1} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot s_{LA,NB}})$$

przykład XOR, próbka [1 0 1] wprowadzanie korekt

$\beta = 1$



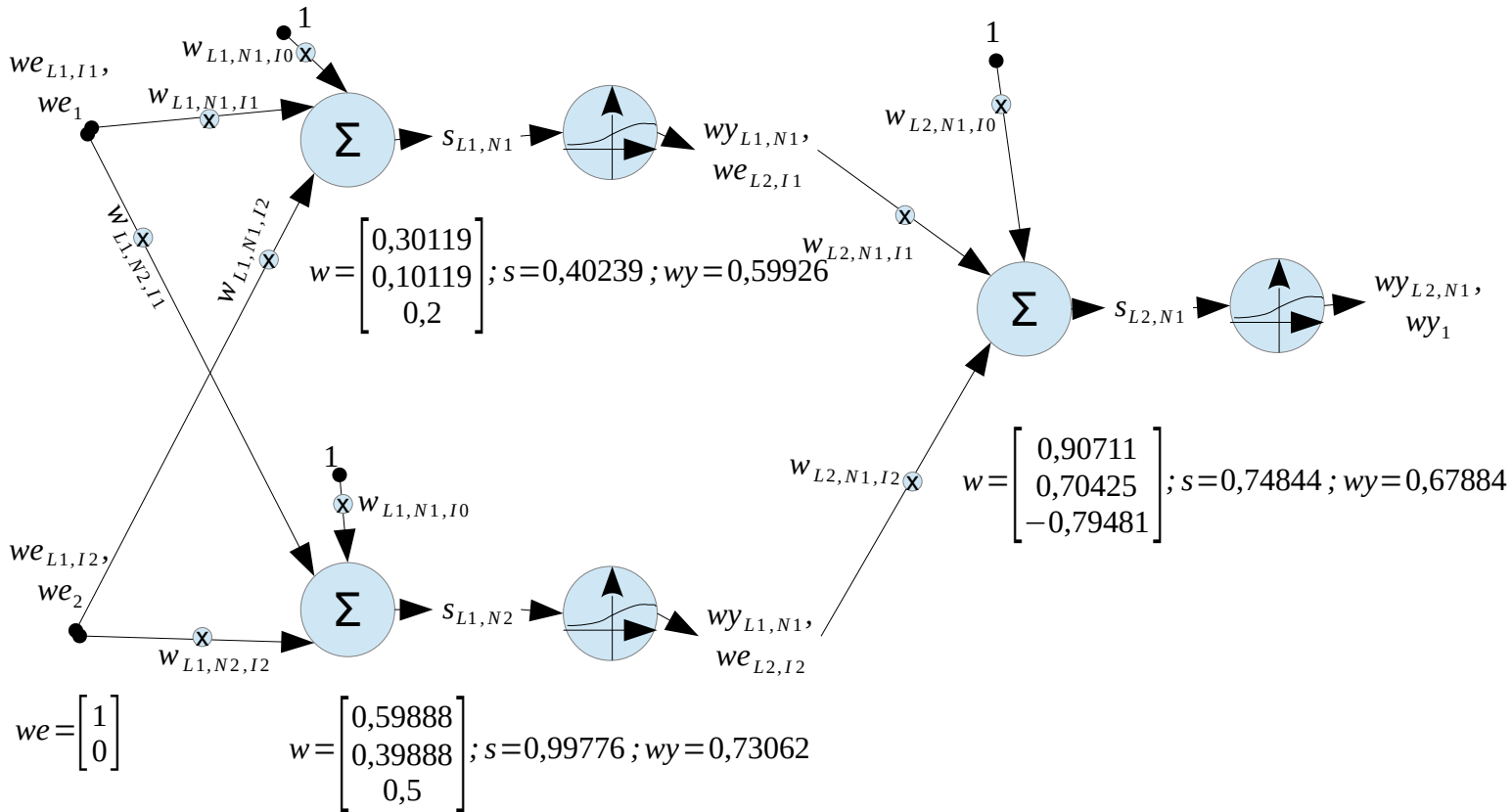
$$W_{LA, NB, IC} += \delta_{w, LA, NB, IC}$$

$$S_{LA, NB} = w_{LA, NB, I0} + \sum_{i=1}^n we_{LA, Ii} * w_{LA, NB, Ii} \quad wy_{LA, NB} = 1 / (1 + e^{-\beta \cdot S_{LA, NB}})$$

przykład XOR, próbka [1 0 1]

sprawdzenie efektów uczenia (krok niekonieczny)

$\beta = 1$



$$S_{LA,NB} = w_{LA,NB,I0} + \sum_{i=1}^n we_{LA,I1} * w_{LA,NB,I1} \quad wy_{LA,NB} = 1 / (1 + e^{-\beta \cdot S_{LA,NB}})$$

przykład XOR, próbka [1 0 1]

sprawdzenie efektów uczenia (krok niekonieczny)

wartość pożądana = 1

wyście przed uczeniem = 0,67573

wyście po uczeniu = 0,67884

Wsteczna propagacja błędów, inne implementacje

$$\delta_{\text{wyjścieNr}} = (y_{\text{wyjścieNr}} - d_{\text{wyjścieNr}})$$

zamiast

$$\delta_{\text{wyjścieNr}} = \mu \cdot (d_{\text{wyjścieNr}} - y_{\text{wyjścieNr}})$$

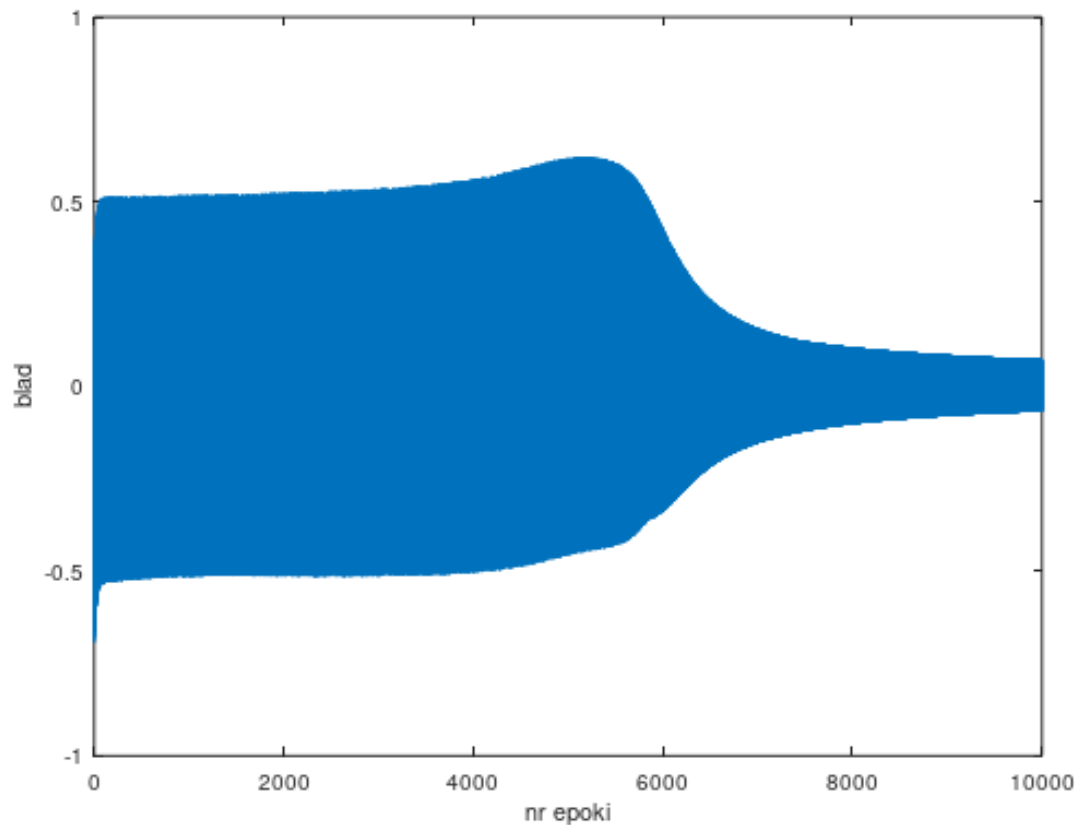
$$\Delta w_{\text{jakaś}} = -\mu \cdot \delta_{\text{jakaś}}$$

zamiast

$$\Delta w_{\text{jakaś}} = \delta_{\text{jakaś}}$$

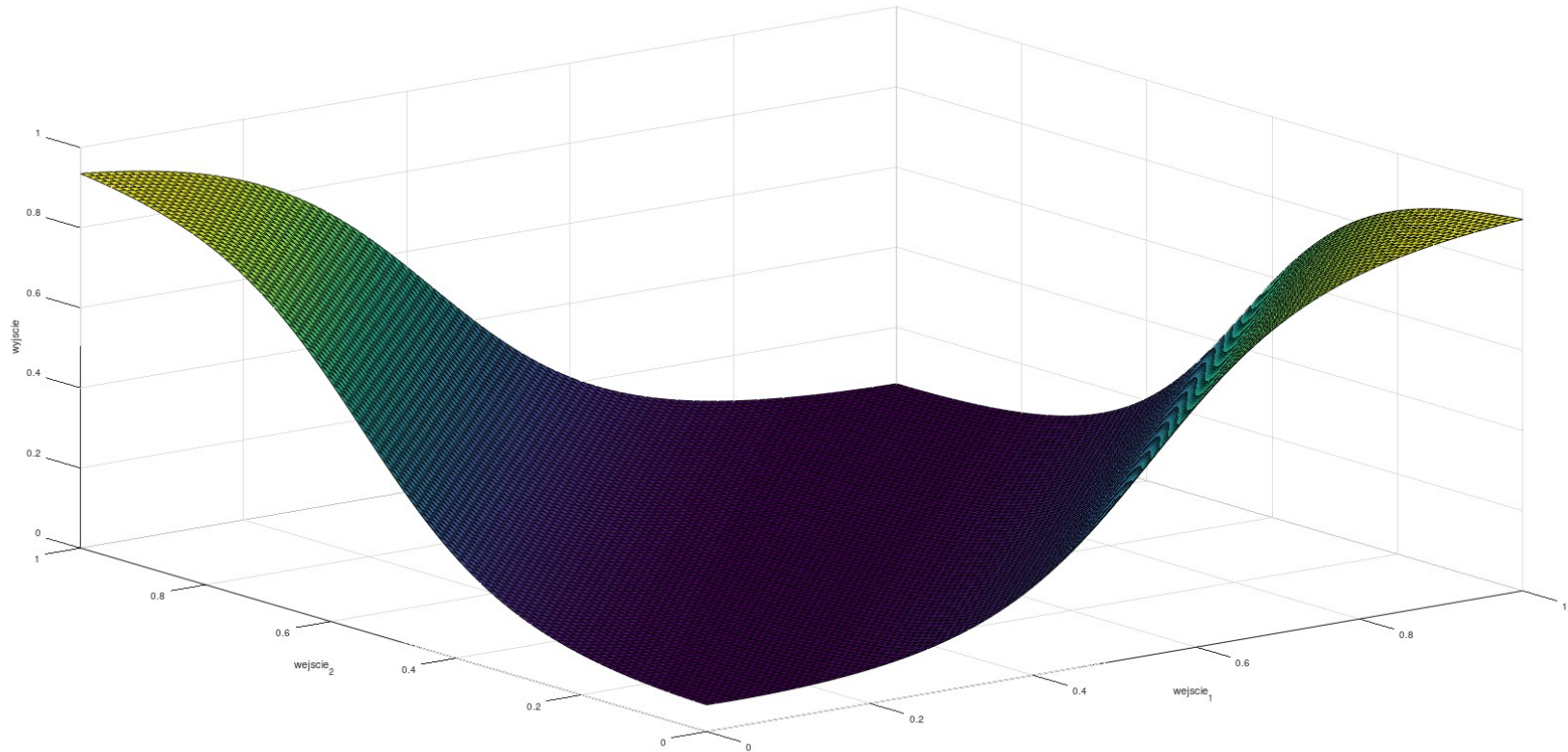
przykład XOR, wynik eksperymentalne

$$\beta = 1; \mu = 0.1$$



przykład XOR, wynik eksperymentalne po 10000 epokach

$$\beta = 1; \mu = 0.1$$



Momentum, dodanie bezwładności

$$w_{\text{jakieś}} += \delta_{\text{jakieś}} + \lambda \cdot \delta_{\text{jakieś}}^{(\text{poprzednie})}$$

zamiast

$$w_{\text{jakieś}} += \delta_{\text{jakieś}}$$

$\lambda \in (0; 1)$ = wsp. bezwładności,

$\delta_{\text{jakieś}}^{(\text{poprzednie})}$ - korekta dla poprzedniej iteracji uczenia

momentum

