Reguła DELTA

Jeżeli neutron nie jest nauczony (sygnał wyjściowy jest inny niż oczekiwany) należy ocenić wielkość błędu w następujący sposób:

## **δ = z – y**

Następnie należy skorygować wektor wag **W** w następujący sposób:

**W’ = W + ηδX**

gdzie **η** jest współczynnikiem liczbowym, decydującym o szybkości nauki,

**X** wektor wejściowy.

Algorytm backpropagation

Wykorzystywany jest od w sieciach wielowarstwowych. Jeżeli neutron nie jest nauczony (sygnał wyjściowy jest inny niż oczekiwany) należy ocenić wielkość błędu w następujący sposób:

## **δ = z – y**

Następnie dla każdego neuronu należy wyznaczyć wagę błędu poprzez "rzutowanie" błędu wstecz do wszystkich neuronów. Współczynniki wagowe *wmn* wykorzystywane przy wstecznym rzutowaniu błędów są te same przez które mnożone były przesyłane sygnały, tyle tylko, że kierunek przesyłania informacji zostaje w tym wypadku odwrócony (zamiast od wejścia do wyjścia przesyła się je od wyjścia kolejno w kierunku wejścia). Następnie po wyznaczeniu wszystkich błędów przystępujemy do obliczania nowych wag wykorzystując odpowiedni wzór.

W ramach ćwiczeń przy użyciu gotowego programu (zamieszczonego w przykładach na <http://www.aforgenet.com/framework/>) dokonałem przykładowej klasyfikacji punktów. Dodatkowo przy użyciu programu NeurophStudio rozwiązałem program bramki XOR przy wykorzystaniu sieci wielowarstwowej. Rozwiązanie tego problemu nie jest możliwe przy wykorzystaniu sieci jednowarstwowej.

Przeczytałem/przejrzałem wszystkie zamieszczone materiały.