

Sztuczna Inteligencja

Temat projektu – Sieć Neuronowa

Ilość osób w grupie: 3

Stwórz projekt implementujący jednokierunkową sztuczną neuronową złożoną z neuronów typu sigmoidalnego z algorytmem uczenia opartym na wstecznej propagacji błędów z wykorzystaniem metody największego spadku (metoda gradientowa).

Sieć powinna umożliwiać:

- definiowanie współczynnika uczenia,
- konfigurację ilości warstw ukrytych,
- ustalenie ilości neuronów w poszczególnych warstwach,
- odczyt danych treningowych i testujących ze zbiorów *.tab oraz *.arff,
- wizualizację błędów w trakcie procesu uczenia,
- ustalenie warunku zatrzymania procesu uczenia (maksymalny błąd lub ilość iteracji),
- wybór metody największego spadku z członem momentum,
- wykorzystanie innych funkcji aktywacji neuronów (unipolarna, bipolarna).

Pomoc do projektu

Działanie algorytmu dla pojedynczej epoki rozpoczyna się od podania pierwszego wzorca uczącego na wejście sieci. Najpierw jest on przetwarzany przez pierwszą, następnie przez kolejną k -tą warstwę neuronów, gdzie przetworzenie przez pojedynczy i -ty neuron w tej warstwie dane jest zależnością:

$$y_i^k = f(s_i^k(t)) = f\left(\sum_{j=0}^{N_{k-1}} w_{ij}^k(t) x_j^k(t)\right)$$

Otrzymane w ten sposób sygnały z warstwy poprzedniej (k -tej) stają się sygnałami wejściowymi dla warstwy kolejnej ($k+1$). Znając sygnały wyjściowe warstwy ostatniej (L) oraz sygnał wzorcowy d_i (pożądany sygnał warstwy wyjściowej dla danej próbki), można obliczyć błąd na wyjściu sieci zgodnie z zależnością:

$$Q_i^L(t) = d_i^L(t) - y_i^L(t)$$

Teraz można zmodyfikować wagi neuronów warstwy ostatniej korzystając z reguły delta. Modyfikacja wagi j -tej neuronu i -tego przebiega zgodnie z wzorem:

$$w_{ij}^L(t+1) = w_{ij}^L(t) + 2\eta \delta_i^L(t) x_j^L(t), \text{ gdzie } \eta - \text{współczynnik uczenia } (0,1)$$

$$\delta_i^L = Q_i^L(t) f'(s_i^L(t)), \text{ gdzie } f' - \text{pochodna przyjętej funkcji aktywacji}$$

Po tym kroku następuje modyfikacja wag neuronów warstw poprzednich (wsteczna propagacja błędów) zgodnie z zależnościami:

$$w_{ij}^k(t+1) = w_{ij}^k(t) + 2\eta \delta_i^k(t) x_j^k(t)$$

$$\delta_i^k = Q_i^k(t) f'(s_i^k(t))$$

$$Q_i^k(t) = \sum_{m=1}^{N_{k+1}} \delta_m^{k+1}(t) w_{mi}^{k+1}(t)$$

Po modyfikacji wag neuronów w warstwie pierwszej, na wejście sieci podawana jest kolejna próbka.

Dodanie członu momentum polega na przebudowaniu wzoru do modyfikacji wag:

$$w_{ij}^k(t+1) = w_{ij}^k(t) + 2\eta \delta_i^k(t) x_j^k(t) + \alpha [w_{ij}^k(t) - w_{ij}^k(t-1)], \text{ gdzie } \alpha - \text{momentum } (0,1)$$

Jak łatwo zauważyć, człon momentum modyfikuje daną wagę w zależności od wielkości jej zmiany w poprzednim kroku.

Funkcje unipolarna i bipolarna neuronu sigmoidalnego dane są wzorami:

- unipolarna: $f(x) = \frac{1}{1 + e^{-\beta x}}$
- bipolarna: $f(x) = \frac{1 - e^{\beta x}}{1 + e^{-\beta x}}$

Pierwsze pochodne tych funkcji dane są wzorami:

- unipolarna: $f'(x) = \beta f(x)(1 - f(x))$
- bipolarna: $f'(x) = \beta(1 - f^2(x))$

Podsumowanie wykorzystywanych oznaczeń:

- i, m – numer neuronu,
- j – numer wagi neuronu,
- k – numer warstwy,
- L – warstwa wyjściowa (numer ostatniej warstwy),
- t – iteracja w epoce, numer próbki uczącej,
- x – wartość wejściowa neuronu.