

# Sprawozdanie Podstawy sztucznej inteligencji

Tymoteusz Przyłucki  
gr 3

## Wstęp teoretyczny:

Model neuronu Adaline (Adaptive Linear Neuron) został podany przed Widrowa w 1960 roku. Jego istota polega na sposobie uczenia się – korekta wag odbywa się w oparciu o porównywanie oczekiwanej odpowiedzi z potencjałem membranowym neuronu co sprowadza się do zmiany we wzorze opisującym popełniany błąd ( po tej zmianie  $\delta$  nie reprezentuje wprost błędu):

$$\delta_{\mu}(t) = d_{\mu} - u_{\mu}(t)$$

Dzięki takiemu zdefiniowaniu wartości  $\delta$ , pomimo nieliniowego charakteru neuronu, możliwe jest zastosowanie gradientowego algorytmu uczenia bazującego na minimalizacji funkcji średniokwadratowej.

Propozycja Widrowa wprowadziła neuron o nieliniowej (bipolarnej) funkcji przejścia.

$$y = \operatorname{sgn}\left(\sum_{i=0}^n w_i x_i\right) = \begin{cases} 1 & \text{gdy } \sum_{i=0}^n w_i x_i > 0 \\ -1 & \text{gdy } \sum_{i=0}^n w_i x_i \leq 0 \end{cases}$$

Aktualnie w literaturze można spotkać również definicje neuronów Adaline w których jako funkcja aktywacji jest wykorzystywana funkcja liniowa, wówczas odpowiedź neuronu może być opisana wzorem:

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

Wprowadzenie współczynnika proporcjonalności  $k$  jest jedynie zmianą ilościową, a nie jakościową. W przypadku włączenia elementu Adaline w strukturę sieci wielowarstwowej jego rolę mogą przejąć wagi połączeń wychodzących z elementu.

W praktyce z neuronów Adaline buduje się najczęściej sieci jednowarstwowe zwane Madaline (Many ADALINE). W sieciach tych, każdy neuron uczony jest zgodnie z regułą adaline.

Wnioski:

Zadanie zostało wykonane w programie NeurophStudio. Dane zostały przygotowane pod konkretną sieć neuronową – MADALINE( Multi-Layer Neural Network). Uczenie powiodło się na dwuwarstwowej sieci neuronowej z jedną ukrytą która miała 150 neuronów ukrytych, co daje ponad 3000 połączeń. Liczba danych na wejściu i wyjściu wynosiła 20. Dane zostały wykonane przeze mnie w formacie CSV, oraz podane w odpowiedzi sposób jako dane do programu.

Pierwsza faza nauki przebiega szybko i dynamicznie, w dalszej części wykres nauczania zaczyna się spłaszczać ale postęp nauki wciąż trwa. Założyłem pesymistyczny scenariusz i współczynnik uczenia ustawiłem na 0.1. Przy wartości 0.4, sieć uczyła się bardzo szybko, co sprawiało trudność w odczytaniu wyników. Sieć będzie się uczyć dopóki jej błąd nie wyniesie 0.01 lub mniej, co w czasie rzeczywistym jest osiągalne po 20-30 minutach od rozpoczęcia nauki.

Zrzuty z programu. Opcje który zostały ustawione zostały wybrane doświadczalnie po około 30 próbach, dla różnych danych wejściowych.

New Neural Network

**Steps**

1. Set neural network name and type
- 2.

**Setting Multi Layer Perceptron's parameters**

Input neurons: 20

Hidden neurons: 150  
(space delimited for layers)

Output neurons: 20


☒ Use Bias Neurons

☐ Connect input to output neurons

Transfer function: Sigmoid

Learning rule: Backpropagation

< Back   Next >   **Finish**   Cancel   Help

 Training Dialog

Stopping Criteria

Max Error

0.01

☐ Limit Max Iterations

Learning Parameters

Learning Rate

0.1

Momentum

0.7

Crossvalidation

☐ Use Crossvalidation

☐ Subset count

4

☐ Subset distribution (%)

60 20 20

☐ Allow samples repetition

☐ Save all trained networks

Options

☒ Display Error Graph

Turn off for faster learning

Train

Close

