

# SIECI NEURONOWE

## Sprawozdanie - Ćwiczenie 1

Aleksander Poławski  
Grupa - Poniedziałek 18:15  
Prowadzący - mgr inż. Jan Jakubik

18 października, 2020

### 1 Cel ćwiczenia

Celem ćwiczenia pierwszego laboratoriów kursu Sieci Neuronowe było poznanie podstawowych funkcji wykonywanych przez pojedynczy neuron, obserwacja zachowania neuronu przy różnych funkcjach przejścia oraz określenie wielkości, które mają wpływ na szybkość uczenia neuronu.

### 2 Plan ćwiczenia oraz badań

- a) stworzenie programu symulującego działanie pojedynczego neuronu, w formie perceptronu prostego realizującego logiczną funkcję AND oraz przeprowadzenie eksperymentów badających szybkość i skuteczność uczenia się tego perceptronu w zależności od:
  - wartości progu aktywacji i użycia dynamicznego progu (bias)
  - zakresu wartości początkowych losowych wag
  - wartości współczynnika uczenia  $\alpha$
  - zastosowanej funkcji przejścia neuronu (funkcji progowej unipolarnej lub funkcji progowej bipolarnej)
- b) modyfikacja zaimplementowanego perceptronu prostego do Adaline oraz przeprowadzenie podobnych badań jak w przypadku perceptronu prostego - tj. badań szybkości uczenia się Adaline w zależności od zakresu początkowych, losowych wag oraz współczynnika uczenia  $\alpha$
- c) porównanie własności, skuteczności, wyników badań perceptronu prostego i Adaline

### 3 Opis zaimplementowanego programu

Program zaimplementowano w środowisku Visual Studio w języku C#. Składa się z następujących klas:

- DataSets - umożliwia przechowywanie zbiorów uczących i testowych. Klasa pozwala na definiowanie i obsługę zbiorów o dowolnej liczbie wektorów wejściowych o dowolnej długości wektora wejściowego
- Entry - klasa opisująca pojedynczy wektor wejściowy
- PerceptronSettings - instancje tej klasy pozwalają na przechowywanie ustawień perceptronu takich jak: zakres wartości wag początkowych, rodzaj funkcji przejścia, użycie dynamicznego biasu, wartość progu aktywacji, rodzaj perceptronu (prosty lub adaline), wartość współczynnika uczenia  $\alpha$  itd.
- Perceptron - klasa zawierająca logikę uczenia perceptronu o ustawieniach zdefiniowanych w instancji PerceptronSettings
- Program - klasa główna - manager - organizuje kolejność wykonywania zadań programu, zawiera predefiniowane testy potrzebne do wykonania badań przewidzianych w ćwiczeniu

Program w sposób prosty i intuicyjny umożliwia wykonanie wszystkich zaplanowanych w ćwiczeniu badań. Jego zaletą jest też możliwość rozwiązywania innych problemów niż "AND", dzięki implementacji obsługi danych wejściowych o różnych długościach wektorów wejściowych.

## 4 Badania

W poniższej sekcji zamieszczono i opisano wyniki badań.

### 4.1 Wpływ wartości progu aktywacji na wydajność uczenia perceptronu prostego

#### a) Założenia

- rodzaj perceptronu: [prosty]
- funkcja przejścia: [unipolarna]
- zakresu wartości początkowych wag: od  $[-0.8]$  do  $[0.8]$
- wartość współczynnika uczenia  $\alpha$ :  $[0.01]$
- brak dynamicznego progu - biasu
- **zmienna wartość progu aktywacji:  $[-0.50]$ ,  $[-0.25]$ ,  $[0.00]$ ,  $[0.25]$ ,  $[0.50]$ ,  $[0.75]$ ,  $[0.90]$**
- koniec uczenia po 10000 iteracjach lub przy sumie błędów równej 0 dla całej epoki

#### b) Przebieg

- Dla każdej wartości progu wykonanych zostało 100 procedur uczenia, a następnie (dla każdej wartości) obliczona została średnia ilość iteracji potrzebnych do wyuczenia perceptronu.
- Dodatkowo, obok badań dla danych wejściowych wyrażenia "AND" wykonano także badania dla wyrażenia "OR" w celu potwierdzenia czy optymalna wartość progu zależy od zestawu wejściowego.

#### c) Otrzymane wyniki

Próg aktywacji	AND - Średnia liczba iteracji ze 100 prób	OR - Średnia liczba iteracji ze 100 prób
-0.50	nie wyuczono	nie wyuczono
-0.25	nie wyuczono	nie wyuczono
0.00	nie wyuczono	nie wyuczono
0.25	65.45	54.82
0.50	55.98	67.56
0.75	57.54	92.45
0.90	46.02	103.20

Tabela 1: wpływ wartości progu aktywacji na wydajność uczenia perceptronu prostego

#### d) Komentarz

- Wydajność perceptronu prostego jest silnie zależna od dobranej wartości progu aktywacji. Ponadto, optymalna wartość tego progu jest również mocno zależna od charakterystyki zestawu danych treningowych.
- Wykonano podobne badanie tym razem używając dynamicznego progu (bias). Pomyślne wyuczenie następuje w tej konfiguracji średnio po 60 iteracjach (dla "OR" i "AND").
- Zalety wykorzystania automatycznego progu są nieocenione, a spadek wydajności znikomy. Alternatywą jego użycia byłoby strojenie progu aktywacji każdorazowo przy zmianie charakterystyki danych treningowych.

## 4.2 Wpływ wartości zakresu wag początkowych na wydajność uczenia perceptronu prostego

### a) Założenia

- rodzaj perceptronu: [prosty]
- funkcja przejścia: [unipolarna]
- wartość współczynnika uczenia  $\alpha$ : [0.01]
- dynamiczny próg - bias
- **zmienny zakres wartości początkowych wag:** [-1.0] do [1.0], [-0.8] do [0.8], [-0.6] do [0.6], [-0.4] do [0.4], [-0.2] do [0.2], [-0.1] do [0.1], [-0.01] do [0.01]
- koniec uczenia po 10000 iteracjach lub przy sumie błędów równej 0 dla całej epoki

### b) Przebieg

- Dla każdej wartości zmiennej wykonanych zostało 100 procedur uczenia, a następnie (dla każdej wartości) obliczona została średnia ilość iteracji potrzebnych do wyuczenia perceptronu.

### c) Otrzymane wyniki

Zakres wag początkowych	Średnia liczba iteracji ze 100 prób
-1.0 do 1.0	67.88
-0.8 do 0.8	54.08
-0.6 do 0.6	36.66
-0.4 do 0.4	28.96
-0.2 do 0.2	15.79
-0.1 do 0.1	9.09
-0.01 do 0.01	3.88

Tabela 2: wpływ wartości zakresu wag początkowych na wydajność uczenia perceptronu prostego

### d) Komentarz

- Wydajność perceptronu prostego jest silnie zależna od dobranej wartości progu aktywacji. Ponadto, optymalna wartość tego progu jest również mocno zależna od charakterystyki zestawu danych treningowych.
- Wykonano podobne badanie tym razem używając dynamicznego progu (bias). Pomyślne wyuczenie następuje w tej konfiguracji średnio po 60 iteracjach (dla "OR" i "AND").
- Zalety wykorzystania automatycznego progu są nieocenione, a spadek wydajności znikomy. Alternatywą jego użycia byłoby strojenie progu aktywacji każdorazowo przy zmianie charakterystyki danych treningowych.