Wstęp do Sztucznej Inteligencji
Laboratorium 4 - Perceptron prosty

mgr inż. Andrii Shekhovtsov

16 kwietnia 2025

Zasady oceniania

Program, który powstał w ramach tego zadania, powinien zostać przesłany za pośrednictwem Moodle jako plik tekstowy z kodem w Python w formacie `.py`. W przesyłanym pliku z kodem proszę umieścić na pierwszej linii komentarz ze swoim imieniem, nazwiskiem, numerem albumu oraz numerem grupy. Plik proszę nazwać `wdsi_lab4.py`.

Plik ten należy przesłać za pośrednictwem systemu Moodle w wyznaczonym tam terminie.

W przesyłanym pliku z kodem proszę umieścić na pierwszej linii komentarz ze swoim imieniem, nazwiskiem i numerem albumu.

Przykładowe formatowanie pliku:

```
# Jan Kowalski, nr. alb. 12345

# tutaj umieszczamy cały kod programu...
```

UWAGA: Termin oddania zadania jest ustawiony w systemie Moodle. W przypadku nieoddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia. Zadania oddawane później niż miesiąc po terminie ustawionym na Moodle mogą zostać niesprawdzone lub ocenione na ocenę niedostateczną.

UWAGA: W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie `.zip` lub `.pdf` tam, gdzie był wymagany plik tekstowy z rozszerzeniem `.py`.

1 Materiały pomocniczy

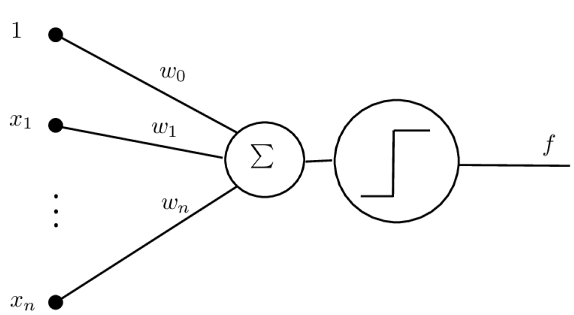
Sieć Perceptron jest najprostszym typem sztucznej sieci neuronowej.

Jest to model pojedynczego neuronu, który może być wykorzystany do rozwiązywania problemów klasyfikacyjnych do dwóch klas i stanowi podstawę do późniejszego rozwoju znacznie większych sieci.

Celem niniejszego laboratorium jest implementacja algorytmu nazywanego regułą perceptronu (regułą delta) do zmieniania parametrów sieci (wag).

1.1 Schemat graficzny

Obliczenia w perceptronie prostym są realizowane zgodnie ze schematem przedstawionym na Rysunku w kierunku od lewej do prawej.



Sygnały wejściowe oznaczone jako $x_1, \dots, x_n$ reprezentują zaobserwowane lub zmierzone cechy pewnego obiektu. Sygnały te są mnożone przez odpowiednie współczynniki wagowe, a następnie sumowane. Obliczona suma wpływa na sygnał wyjściowy, oznaczony jako f, czyli odpowiedź układu.

W związku z złożeniem o klasyfikacji binarnej, możliwe są tylko dwa stany odpowiedzi, a jako popularną konwencję przyjmuje się wartości $\{-1, 1\}$. Jeżeli obliczona suma jest powyżej pewnego ustalonego progu, to układ odpowiada wartością 1, w przeciwnym razie wartością -1. Ten element obliczeń nazywany jest *funkcją aktywacji neuronu*, a w przypadku perceptronu prostego jest to funkcja schodkowa.

Wprowadzimy teraz notację wektorów $\mathbf{x}_i$ rozszerzoną o dodatkową cechę $x_0 = 1$ (dla wygody dalszych zapisów matematycznych):

$$\mathbf{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{in}).$$

Dzięki temu będziemy mogli iloczyn skalarny wektora wag $\mathbf{w}$ i wektora cech $\mathbf{x}_i$ zapisać krótko jako parę: $\langle \mathbf{w}, \mathbf{x}_i \rangle$:

$$\langle \mathbf{w}, \mathbf{x}_i \rangle = \sum_{j=0}^{n} w_j x_{ij}. \tag{1}$$

Zatem funkcje odpowiedzi perceptronu $f(\langle \mathbf{w}, \mathbf{x}_i \rangle)$ możemy przedstawić jako:

$$f(\langle \mathbf{w}, \mathbf{x}_i \rangle) = \begin{cases} 1, & \text{dla } \langle \mathbf{w}, \mathbf{x}_i \rangle > 0 \\ -1, & \text{dla } \langle \mathbf{w}, \mathbf{x}_i \rangle \leq 0 \end{cases} \tag{2}$$

Poniżej jest przedstawiony algorytm uczenia wag za pomocą „Reguły Perceptronu”. Proszę zwrócić uwagę, że zbiór danych D zawiera pary wartości $\mathbf{x}_i$, y_i: $D = \{(\mathbf{x}_i, y_i)\}$.

Algorytm 1 „Reguła perceptronu”

procedure PerceptronLearningRule(D, η)
  $\mathbf{w}(0) := (0, 0, \dots, 0)$ ▷ początkowy wektor wag, może być także losowy
  $k := 0$ ▷ licznik kroków
  while zbiór błędnie sklasyfikowanych punktów $E = \{(\mathbf{x}_i, y_i) \colon\ y_i \neq f(\langle \mathbf{w}(k), \mathbf{x}_i \rangle)\}$ jest niepusty **do**
    wylosuj ze zbioru E dowolną parę $(\mathbf{x}_i, y_i)$
    popraw wektor wag wg wzoru: $\mathbf{w}(k+1) := \mathbf{w}(k) + \eta y_i \mathbf{x}_i$
    $k := k + 1$
  return $\mathbf{w}(k)$

W przypadku gdy dane są liniowo separowalne, algorytm zawsze powinien zatrzymać się po wykonaniu skończonej liczby kroków. Natomiast, do powyższego algorytmu dodać można „hamulec bezpieczeństwa”, czyli ograniczyć odgórnie liczbę wykonywanych iteracji. Zagwarantuje to skończenie działania programu nawet w przypadku, gdyby dane nie były separowalne liniowo.

Do wygenerowania danych do testowania implementacji proszę wykorzystać funkcje z poniższego przykładu:

```python
import numpy as np
import matplotlib.pyplot as plt

# Funkcja do generowania próbek (tylko przypadek 2 wymiarowy)
def generate(n=100, seed=None):
    np.random.seed(seed)
    x1 = 2 * np.random.rand(n) - 1
    x2 = 2 * np.random.rand(n) - 1

    a = 4 * np.random.rand() - 2
    b = 0.1 * np.random.rand() - 0.05

    y = (a * x1 + b) >= x2
    y = 2 * y - 1
    return np.array([x1, x2]).T, y

# Wyświetlenie 100 wygenerowanych próbek
X, y = generate(100)
```

```
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()

# Jeżeli podamy seed będzie generować te same dane za każdym razem
X, y = generate(100, seed=42)
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```

2 Zadania do wykonania

Celem jest implementacja algorytmu uczenia regułą perceptronu do klasyfikacji binarnej separowalnego liniowo zbioru.

1. Zaimplementuj algorytm uczący jako funkcję przyjmującą jako argumenty zbiór danych $D = \{(\mathbf{x}_i, y_i)\}$ i współczynnik uczenia $\eta \in [0, 1]$.

   Uwaga: funkcja może być (nie musi) ogólna, tzn. pracować dla danych dowolnej wymiarowości. Jako rezultaty zwróć otrzymany wektor wag oraz liczbę wykonanych kroków aktualizacyjnych (licznik k).

2. Sprawdź wpływ następujących zmian na liczbę wykonywanych iteracji k:

   (a) liczby przykładów uczących

   (b) współczynnika uczenia (parametr η)

   Przygotuj wykresy, ilustrujące te zależności.

3. Na końcu pliku z kodem podaj wnioski z przeprowadzonego eksperymentu (wpływ liczby przykładów uczących i współczynnika uczenia na liczbę iteracji).

4. W przypadku danych dwuwymiarowych możliwe jest wygenerowanie wykresu przedstawiającego dane oraz znalezioną linię separacji klas. Przygotuj taką wizualizację za pomocą implementacji własnej.

   Podpowiedź: linię separacji można zdefiniować przekształcając równanie:

   $$w_1x_1 + w_2x_2 + w_0 = 0$$

   Należy przedstawić to równanie w postaci $x_2 = ax_1 + b$, gdzie a i b będą współczynnikami definiującymi linię separacji na płaszczyźnie, obliczone za pomocą w_0, w_1 oraz w_2.

Skala ocen

- Na ocenę 3.0: Zadania 1
- Na ocenę 4.0: Zadania 1-3
- Na ocenę 5.0: Zadania 1-4