Uczenie regułą perceptronu

Przemysław Klęsk & Joanna Kołodziejczyk

1 Zagadnienie

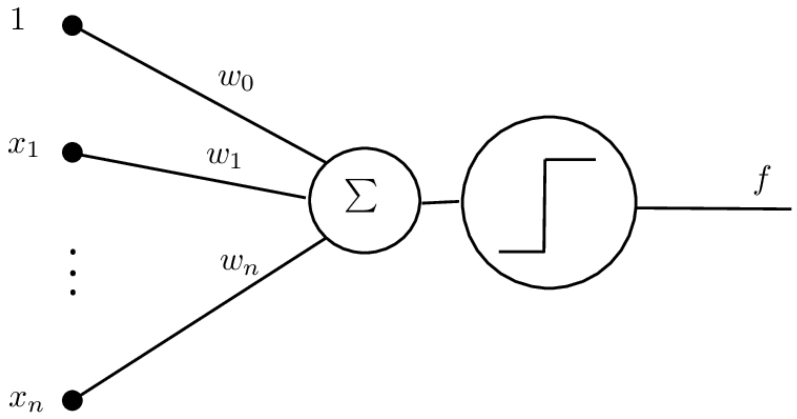
Sieć Perceptron jest najprostszym typem sztucznej sieci neuronowej.

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.

Perceptron będzie szacować klasę pewnej zadanej próbki po procesie uczenia. Celem niniejszego laboratorium jest implementacja algorytmu nazywanego 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 $x_1, \ldots, x_n$ 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.

Przyjmując jako próg decyzyjny wartość 0, omówiony schemat obliczeń można zapisać następująco:

$$s = w_0 + w_1 x_1 + \cdots + w_n x_n. \tag{1}$$

$$f(s) = \begin{cases} 1, & \text{dla } s > 0; \\ -1, & \text{dla } s \leqslant 0. \end{cases} \tag{2}$$

1.2 Algorytm uczenia perceptronu prostego regułą delta

Zakładamy, że dany jest zbiór uczący składający się z m punktów $\mathbf{x}_i$ i odpowiadające im oczekiwane wyjścia y_i.

1. Niech $\mathbf{w}(0) = (0, \ldots, 0)$
2. $k = 0$
3. Dopóki zbiór punktów uczących pozostaje błędnie klasyfikowany tj. zbiór $E = \{\mathbf{x}_i : y_i \neq f(s)\}$ pozostaje niepusty, powtarzaj:
   (a) Wylosuj ze zbioru E dowolną parę $(\mathbf{x}_i : y_i)$
   (b) Aktualizuj wagi według następującej reguły:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta y_i \mathbf{x}_i$$

   (c) $k = k + 1$

2 Zadanie

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

1. Wygeneruj zbiór danych w sposób sztuczny, kontrolując jego rozmiar m, a także margines odstępu pomiędzy klasami. Dane przechowaj w formie macierzy o wymiarach $m \times 4$, gdzie kolejne kolumny przechowują wartości: 1, x_{i1}, x_{i2}, y_i.

   Do zadania można na przykład użyć make_classification z biblioteki scikit-learn.

```
from pandas import np
from sklearn.datasets import make_classification

m = 20
X1, Y1 = make_classification(n_features=2, n_redundant=0, n_informative=2,
        n_clusters_per_class=1, n_classes=2, n_samples=m, class_sep = 2)
plt.scatter(X1[:, 0], X1[:, 1], marker='x', c=Y1, s=25, edgecolor='k')

plt.show()
X_train = np.hstack((X1, np.ones((X1.shape[0], 1), dtype=X1.dtype)))
```

2. Przedstaw dane w formie wykresu.

3. Zaimplementuj algorytm uczący jako funkcję przyjmującą jako argumenty zbiór danych 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).

4. Sprawdź wpływ następujących zmian na licznik k:

   (a) liczby przykładów (parametr m)

   (b) współczynnika uczenia (parametr η)

3 Przekazanie zadań

Kod z rozwiązaniem oraz wykresy z analizy wpływu parametrów na k proszę podpiąć w Teams. Proszę w nazwach plików źródłowych zawierać swoje nazwisko celem łatwiejszej identyfikacji.