

Sieci neuronowe w zastosowaniach biomedycznych - analiza danych i koncepcja realizacji

Milena Kuna - 325033
Karol Franczuk - 325001
Zespół 16, temat 12

12 kwietnia 2025

Temat projektu

Klasyfikacja kwiatów za pomocą sieci uczonej bez nauczyciela (SOM) – katalog: *iris*.

1. Analiza danych

1.1 Opis zbioru danych

Zbiór danych *Iris* (*Fisher's Iris Dataset*) zawiera pomiary 150 egzemplarzy trzech gatunków irysów: *Iris-setosa*, *Iris-versicolor*, *Iris-virginica*. Każda próbka opisana jest za pomocą czterech cech numerycznych:

Nr	Nazwa cechy	Jednostka	Zakres wartości
1	Długość działki kielicha	cm	4.3 – 7.9
2	Szerokość działki kielicha	cm	2.0 – 4.4
3	Długość płatka	cm	1.0 – 6.9
4	Szerokość płatka	cm	0.1 – 2.5

Dodatkową cechą jest etykieta klasy (gatunek kwiatu), wykorzystywana jedynie w celu oceny działania klasyfikatora.

1.2 Statystyki opisowe cech

Cecha	Średnia	Odchylenie standardowe
Długość działki kielicha	5.84	0.83
Szerokość działki kielicha	3.05	0.43
Długość płatka	3.76	1.76
Szerokość płatka	1.20	0.76

2. Wstępne przetwarzanie danych

- **Skalowanie cech:** cechy wejściowe zostaną przeskalowane do zakresu $[0, 1]$ (normalizacja) lub znormalizowane (średnia = 0, odchylenie = 1), w zależności od wymagań zastosowanej implementacji SOM.
- **Kodowanie danych nienumerycznych:** zbiór zawiera tylko cechy numeryczne, więc kodowanie nie jest wymagane.

- **Podział danych:** w przypadku SOM, jako sieci uczącej się bez nauczyciela, nie jest wymagany podział na zbiór treningowy i testowy. Etykiety klas będą używane jedynie do wizualizacji i oceny jakości klasteryzacji.

3. Koncepcja realizacji sieci neuronowej

3.1 Typ sieci

Zastosowana zostanie sieć typu **Self-Organizing Map (SOM)**, znana również jako mapa Kohonena. Jest to sieć neuronowa ucząca się bez nadzoru, wykorzystywana głównie do grupowania danych i redukcji wymiarowości.

3.2 Architektura i parametry sieci

- **Liczba neuronów wejściowych:** 4 (odpowiadające czterem cechom kwiatów).
- **Struktura warstwy SOM:** siatka 10x10 neuronów (100 neuronów).
- **Liczba epok:** wstępnie 10 (może ulec zmianie).
- **Funkcja sąsiedztwa:** gaussowska.
- **Metryka odległości:** odległość euklidesowa.
- **Algorytm uczenia:** klasyczny algorytm Kohonena z dynamicznym zmniejszaniem promienia sąsiedztwa i współczynnika uczenia.

Funkcja aktywacji: W klasycznej sieci SOM nie stosuje się funkcji aktywacji w tradycyjnym sensie. Zamiast tego, „aktywacja” neuronu to jego wybór jako zwycięzcy (BMU – *Best Matching Unit*) na podstawie najmniejszej odległości euklidesowej między wagami a wektorem wejściowym.

3.3 Algorytm uczenia

Uczenie odbywa się iteracyjnie poprzez dopasowywanie wag neuronów do danych wejściowych. W każdej iteracji:

1. Wybierany jest wektor wejściowy \mathbf{x} .
2. Znajdowany jest neuron BMU – neuron o najmniejszej odległości od \mathbf{x} .
3. Aktualizowane są wagi BMU i jego sąsiadów według wzoru:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) \cdot h_{ci}(t) \cdot (\mathbf{x}(t) - \mathbf{w}_i(t))$$

gdzie:

- $\eta(t)$ – współczynnik uczenia malejący w czasie,
- $h_{ci}(t)$ – funkcja sąsiedztwa, zwykle gaussowska.

3.4 Ocena wyników

Po zakończeniu treningu, neurony SOM przypisują próbki do klastrów. Dla wizualizacji wyników planujemy wykorzystać:

- Mapę U-Matrix (pokazującą odległości między neuronami).
- Mapę aktywacji neuronów z zaznaczonymi klasami etykiet.
- Porównanie przypisania klas do neuronów względem rzeczywistych klas w zbiorze danych.

4. Potencjalne rozszerzenia

Na etapie 2 planujemy:

- Wykonać własną implementację SOM w Pythonie.
- Eksperymentować z różnymi strukturami/parametrami SOM (np. 5x5, 15x15).
- Testować różne wartości współczynnika uczenia.
- Obliczyć metryki skuteczności klasyfikacji: **czułość** i **specyficzność**.