Laboratorium: Wprowadzenie do sieci neuronowych

May 9, 2023

1 Cel/Zakres

- Przypomnienie zasady działania i ograniczeń perceptronu
- Proste sieci neuronowe w Scikit-learn

2 Zadania

2.1 Perceptrony i irysy

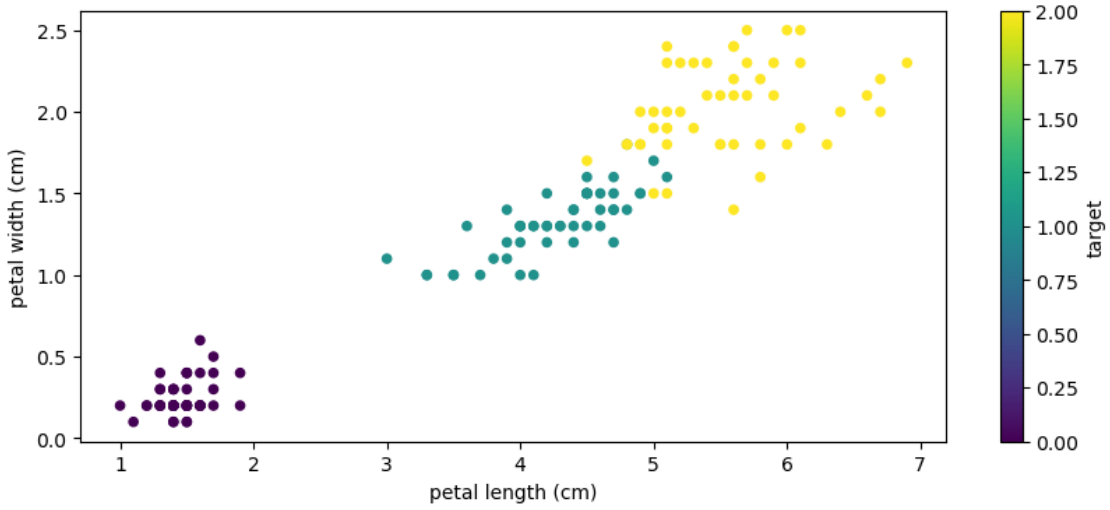
Pobierz zbiór danych Iris.

```
from sklearn.datasets import load_iris
iris = load_iris(as_frame=True)
```

Zwróć uwagę, jak długość i szerokość płatka jest powiązana z gatunkiem irysów.

```
pd.concat([iris.data, iris.target], axis=1).plot.scatter(
    x='petal length (cm)', y='petal width (cm)', c='target',
    colormap='viridis', figsize=(10,4)
)
```

```
<Axes: xlabel='petal length (cm)', ylabel='petal width (cm)'>
```



Podziel zbiór danych na zbiór uczący oraz zbiór testowy (8/2), a następnie kolejno dla każdego z gatunków $(0, 1, 2)$ zbuduj perceptron, przeprowadź jego uczenie (przy pomocy dwóch omawianych parametrów) i oceń jego dokładność dla zbioru uczącego i dla zbioru testowego.

Dla każdej z klas sprawdź wagę biasu (`w_0`) oraz poszczególnych wejść (`w_1`, `w_2`). Czy wartości te korespondują z tym co widzisz na obrazku?

Czy, patrząc na ten obrazek, potrafisz wyjaśnić różnice w dokładności dla poszczególnych klas?

Dokładność dla każdej z klas zapisz jako krotkę `(a_tr, a_te)`, a wszystkie krotki zapisz w pliku `per_acc.pkl` jako listę krotek: `[(a_tr_0, a_te_0), (a_tr_1, a_te_1), (a_tr_2, a_te_2)]`.

Podobnie, wagi dla poszczególnych trzech klas zapisz jako listę 3-elementowych krotek `(w_0, w_1, w_2)` w pliku `per_wght.pkl`.

2.2 Perceptron i XOR

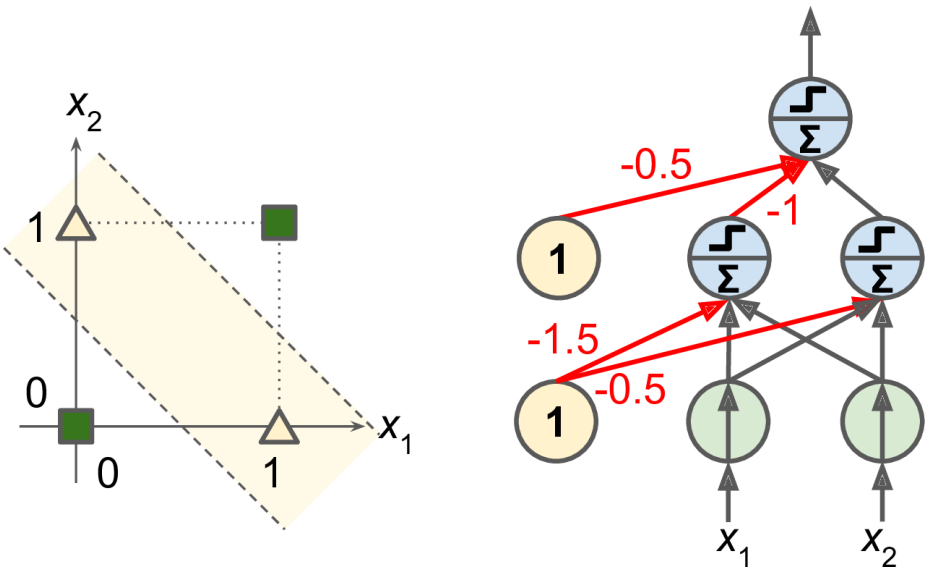
Przygotuj prosty zbiór danych modelujący operację XOR.

```
X = np.array(
    [[0, 0],
     [0, 1],
     [1, 0],
     [1, 1]])
y = np.array([0,
     1,
     1,
     0])
```

Utwórz i przeprowadź uczenie perceptronu dla tych danych. Czy jest zdolny do poprawnego przeprowadzenia predykcji? Jak wyglądają jego wagi?

2.3 XOR, drugie podejście

Problem klasyfikacji XOR teoretycznie da się rozwiązać przy pomocy sieci przedstawionej na poniższym rysunku.



Zamodeluj tę sieć neuronową (perceptron wielowarstwowy, MLP) przy pomocy klasyfikatora MLP scikit-learn.

Przeprowadź eksperymenty z różnymi wartościami parametrów:

- funkcja aktywacji neuronów warstwy ukrytej,
- aglorytm optymalizacji (*solver*),
- maksymalna liczba iteracji (epok),
- długość kroku uczenia.

Po każdym uczeniu sprawdź dokładność oraz wyjście z sieci:

```
model.score(X, y)
model.predict(X)
```

Za każdym razem sprawdź wartości wag przy połączeniach wewnątrz sieci oraz przy połączeniach neuronów biasu. Czy wartości są zbliżone do tych z rysunku?

Każdy eksperyment powtórz kilka razy. Czy wyniki są stabilne?

Pozostaw w swoim programie zestaw parametrów który uznasz za najlepszy. Utworzony model zapisz w pliku `mlp_xor.pkl`.

Teraz przypisz ręcznie wagi tak, aby odpowiadały tym z rysunku. Sprawdź działanie sieci. Zapisz model do pliku `mlp_xor_fixed.pkl`.

```
with open('mlp_xor_fixed.pkl', 'wb') as f:
    pickle.dump(model, f)
```

3 Wyślij rozwiązanie

Zapisz skrypt wykonujący wszystkie ćwiczenia w pliku `lab09/lab09.py` i wyślij go do swojego repozytorium.