

Laboratorium 2

Programowanie Sieciowe

Klasyfikacja wieloklasowa

Marek Bazan

III rok

Semestr letni 2021/2022

Plan zajęć

1. Dane, na których będziemy pracować (1),(2)
2. Klasyfikacja wieloklasowa za pomocą perceptronu wielowarstwowego
3. Przeszukiwanie siatki hiper-parametrów
4. Zadania do wykonania
 - 4.1 Klasyfikacja Iris Data
 - 4.2 Klasyfikacja na zbiorze mnist
 - 4.3 Dowolne zadanie z kaggle'a
5. Dane, na których będziemy pracować (3)

Zadanie 1 i 2

- ▶ Iris
- ▶ Klasyfikacja cyfr pisanych odręcznie

https://en.wikipedia.org/wiki/MNIST_database



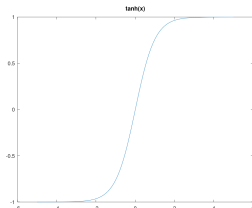
```
import listings
(trainX, trainy), (testX, testy) = mnist.load_data()
```

Klasyfikacja wieloklasowa - struktura sieci (1)

Model perceptronu wielowarstwowego z jedną warstwą ukrytą

https://scikit-learn.org/stable/modules/neural_networks_supervised.html

1. Wejście - kodowanie liter – wektor o długości $28 * 28$ z obrazkiem.
2. Warstwy ukryte – funkcja aktywacji $\tanh(x)$



3. Wyjście – kodowanie wyjścia typu hot one
 n -ta klasa na wyjściu kodowana $(0, \dots, \underbrace{1}_{n\text{-ta składowa}}, \dots, 0)$

Klasyfikacja wieloklasowa - struktura sieci (2)

1. Funkcja softmax na wyjściu sieci dla klasyfikacji wieloklasowej
<https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax>

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ dla } i = 1, \dots, K \text{ i } z = (z_1, \dots, z_K) \in \mathbb{R}^K$$

2. Klasyfikator oparty o MLP – patrz

[https:](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)

[//scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)

Aby zdefiniować na wyjściu funkcję softmax należy definiując klasyfikator użyć podstawienia:

your_model.out_activation_ = 'softmax'.

Klasyfikacja wieloklasowa - uczenie sieci - funkcja celu

https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html

Uczenia z doboorem hiper-parametrów

1. Z modułu scikit-learn

```
class sklearn.neural_network.MLPClassifier(  
    hidden_layer_sizes=(100, ),  
    activation="tanh", solver="adam",  
    batch_size="auto", learning_rate="constant",  
    learning_rate_init=0.001,  
    max_iter=200, shuffle=True,  
    warm_start=False, momentum=0.9,  
    validation_fraction=0.1,  
    epsilon=1e-08,  
    n_iter_no_change=10)
```

2. GridSearchCV – [https:](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)

[//scikit-learn.org/stable/modules/generated/
sklearn.model_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)

Algorytm uczenia perceptronu wielowarstwowego

`solver` : 'lbfgs', 'sgd', 'adam', default 'adam'

Mamy do wyboru następujące optymalizatory:

1. 'lbfgs' procedura optymalizacyjna z rodziny metod quasi-newtonowskich,
2. 'sgd' algorytm minimalizacji względów gradientu przy czym, które wzorce mają wpływ na kierunek w danej iteracji jest losowane,
3. 'adam' jedna ze skuteczniejszych wersji minimalizacji względów stochastycznego gradientu zaproponowana przez Kingma, Diederik, and Jimmy Ba

Przegląd zupełny przestrzeni parametrów

1. Liczba neuronów (`hidden_layer_sizes=(N,)`) –
 $N \in \{20, 40, \dots, 100\}$,
2. Współczynnik uczenia sieci (`learning_rate_init = η`) –
 $\eta = \{0.1, 0.01, 0.001\}$,
3. Algorytm optymalizacji wag (`solver = name`) –
`name` $\in \{\text{adam}, \text{lbfgs}, \text{sgd}\}$.

Zadania

1. Iris
2. MNist
3. Kaggle.com – klasyfikacja

W sprawozdaniu wynik każdego zadania proszę zwizualizować za pomocą macierzy pomyłek (ang. confusion matrix)