

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Лабораторная работа №2

по дисциплине «Машинное обучение»

Выполнил
студент гр. 3530904/00103

Плетнева А.Д.

Руководитель

Селин И.А.

Санкт-Петербург
2023

Оглавление

Задание3

 Задание 14

 Задание 25

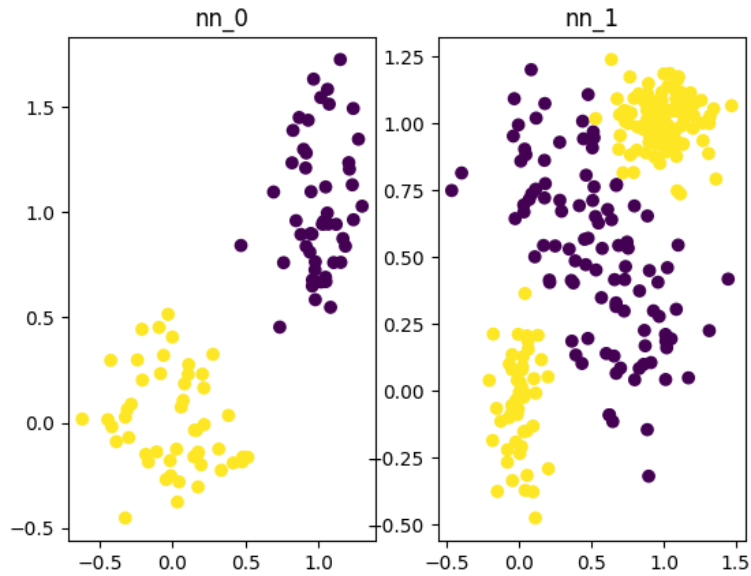
 Задание 36

Задание

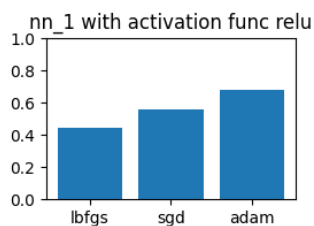
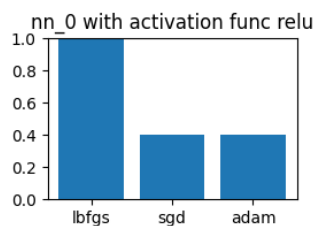
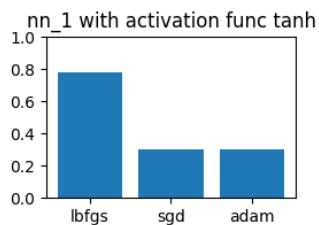
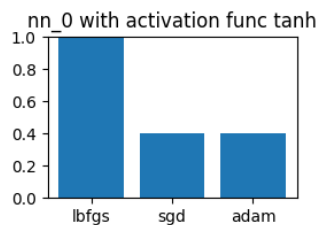
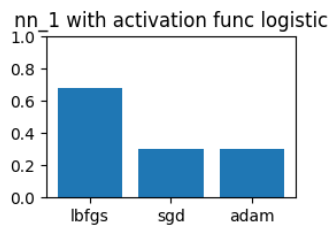
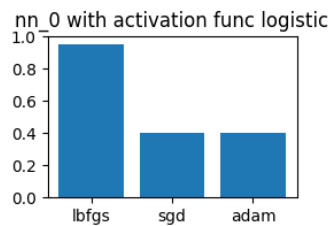
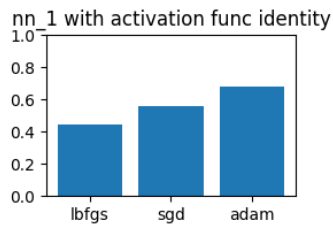
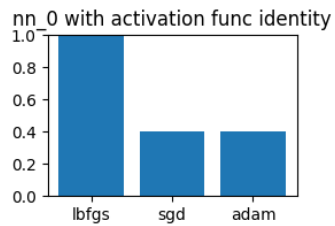
1. Постройте нейронную сеть из одного нейрона и обучите её на датасетах `mn_0.csv` и `mn_1.csv`. Насколько отличается результат обучения и почему? Сколько потребовалось эпох для обучения? Попробуйте различные функции активации и оптимизаторы.
2. Модифицируйте нейронную сеть из пункта 1, чтобы достичь минимальной ошибки на датасете `mn_1.csv`. Почему были выбраны именно такие гиперпараметры?
3. Создайте классификатор на базе нейронной сети для набора данных [MNIST](#) (так же можно загрузить с помощью `torchvision.datasets.MNIST`, `tensorflow.keras.datasets.mnist.load_data` и пр.). Оцените качество классификации.

Задание 1

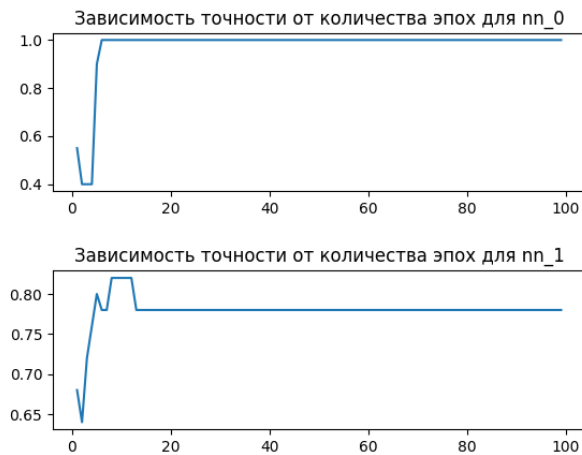
Визуализируем данные из обоих датасетов, видно, что данные из первого набора можно разделить линейно, а из второго - нет



Для данных nn_0 лучшим оптимизатором оказался lbfgs, а функции активации для этой выборки не сыграли роли. Для nn_1 лучшими функцией активации и оптимизатором оказались tanh и lbfgs



При tanh и lbfgs для обеих выборок установили зависимость точности от количества эпох обучения – уже на 6 эпохе мы получаем максимальную точность для nn_0, для второй выборки точность меньше, так как данные этой выборки нельзя разделить линейно, как данные первого сета

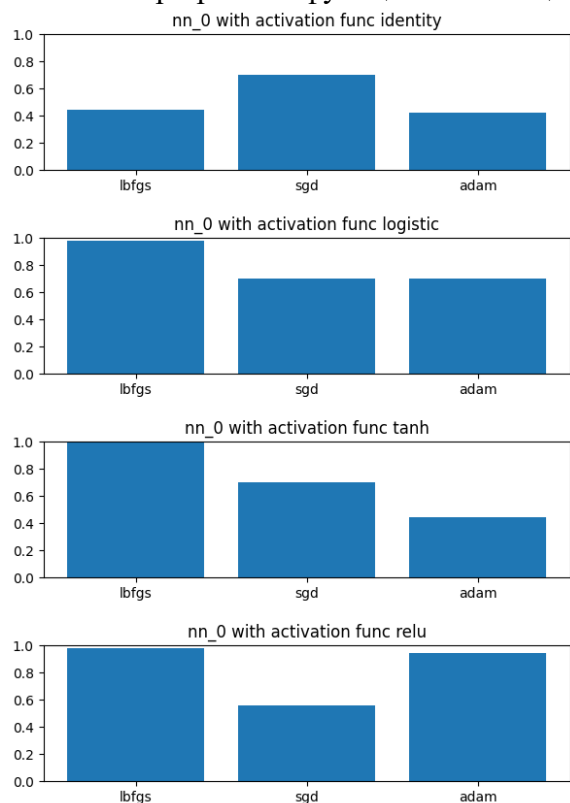


Задание 2

Так как данные nn_1 не являются линейно разделимыми, то использование одного нейрона недостаточно для достижения высокой точности. Сделаем два слоя нейронов, возьмем максимальное количество итераций равным 14 и найдем с помощью поиска по сетке оптимальное количество нейронов в каждом слое. В библиотеке sklearn он реализован с помощью класса GridSearchCV. Будем выбирать оптимальное количество нейронов в каждом слое из вариантов от 1 до 25 нейронов для каждого слоя. Получаем:

```
{'hidden_layer_sizes': (14, 13)}
```

Точность при разных функциях активации и оптимизаторов:



Получаем точность 100% при 50 итерациях, двух скрытых слоях с 14 и 13 ядрами, функции активации – tanh и оптимизаторе lbfgs

Задание 3

Создали классификатор на базе нейронной сети для набора данных [MNIST](#), сделали 1 скрытый слой с 64 нейронами с функцией активации – relu и оптимизатором – adam

```
Max_iter=1: 0.8369285714285715
Max_iter=2: 0.8526428571428571
Max_iter=3: 0.8841428571428571
Max_iter=4: 0.9024285714285715
Max_iter=5: 0.9076428571428572
Max_iter=6: 0.9135
Max_iter=7: 0.9216428571428571
Max_iter=8: 0.9217142857142857
Max_iter=9: 0.9306428571428571
Max_iter=10: 0.9347142857142857
```

Для 10 эпох получаем 93,5% точности, что является довольно хорошим результатом