МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Лабораторная работа №\_\_1\_\_**

по дисциплине«Разработка нейронных сетей»

Тема: «Введение в DL»

ИСПОЛНИТЕЛЬ: \_\_\_\_Гаврилов Л.Я.

ФИО

группа ИУ5-24М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"16"\_\_\_02\_\_\_2024 г.

ПРЕПОДАВАТЕЛЬ: \_\_\_Канев А.И.\_\_\_\_\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_\_"\_\_\_\_\_\_\_\_\_2024 г.

Москва - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Задание

Необходимо познакомиться с фреймворком машинного обучения PyTorch и выполнить три задания:

1. Регрессия по теореме универсальной аппроксимации, ручное дифференцирование
2. Бинарная классификация с помощью автодиффиренцирования PyTorch
3. Обучить полносвязную нейронную сеть классификации 3 классов изображений из набора данных CIFAR100 по варианту из примера и затем повысить точность на тестовой выборке.

Для задания 3 нужно сформировать свою подвыборку CIFAR100 по варианту. Вы должны использовать следующие классы из CIFAR100:

1. Номер группы + 15
2. Номер варианта + 56
3. ИУ5 (Номер варианта + 21); ГУИМЦ (80); Иностранцы (90)

Полученные варианты:

1. 24 + 15 = 39
2. 14 + 56 = 70
3. 14 + 21 = 35

# Часть 1. Задача регрессии по теореме универсальной аппроксимации, ручное дифференцирование

Выборка:

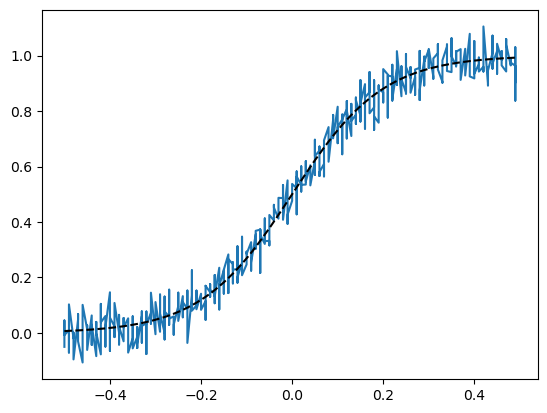


Рис. 1 – аппроксимируемая функция.

Решается задача аппроксимации функции – отмечена на рис.1 пунктиром. В значения y внесён сдвиг с ст. кв. откл. 0.05

Аппроксимация производится с помощью градиентного спуска. Функция потерь – сумма квадратов отклонения. Параметры НС: 1 вход, 1 скрытый слой из 64 нейронов, 1 выход.

На каждой итерации высчитывается производная по весам и смещению для обоих уровней НС, после чего значения сдвигаются на x\_new = x\_old - 0,0001\*grad(x).

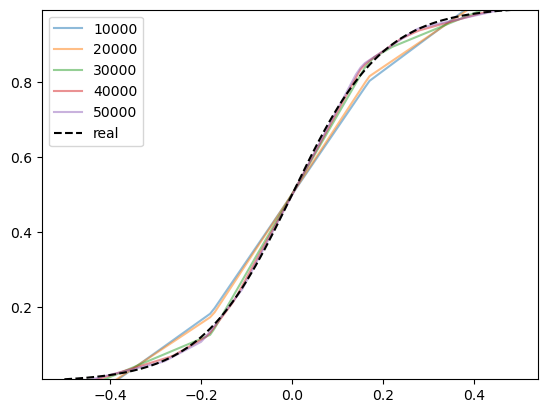


Рис. 2 – результат аппроксимации.

При прохождении 50000 итераций сумма квадратов потерь теперь составляет 0.0025866674. Результат аппроксимации указан на рис. 2.

# Часть 2. Бинарная классификация с помощью автодиффиренцирования PyTorch

Решается задача классификации. Классы отображены на рис.3. Параметры НС: 2 входа (соответственно x и y), 1 скрытый слой из 16 нейронов, 1 выход – класс.

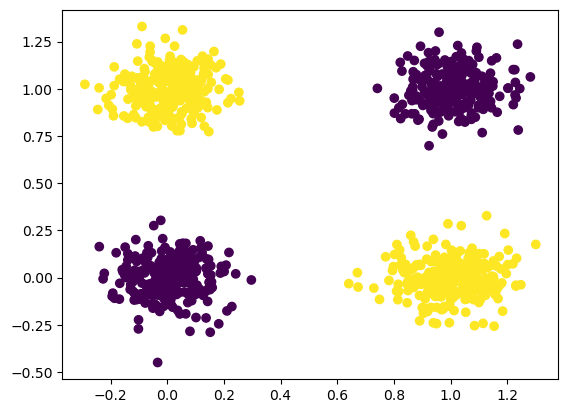


Рис. 3 – график классов.

Функция потерь – логарифмическая

чем дальше значение от , тем ближе значение к 1. Соответственно для всего объёма значений берём сумму всех потерь.

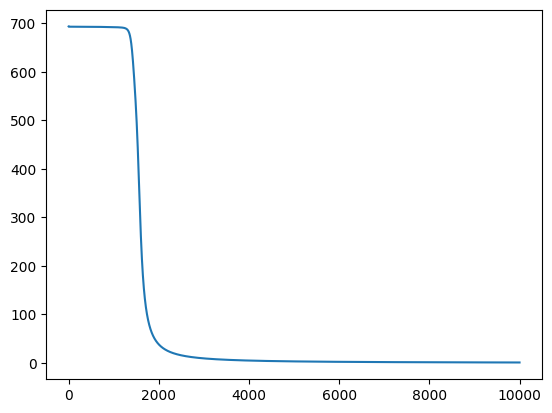


Рис. 4 – график общей функции потерь.

По рисунку видно, что после около 2.4 тысяч итераций НС начала сильно уменьшать функцию потерь, т.е. было найдено верное направление градиента. После 8000 итераций функция потерь находится у минимума. На рис.5 изображён график вывода нейронной сети в области [-0.5;0.5]

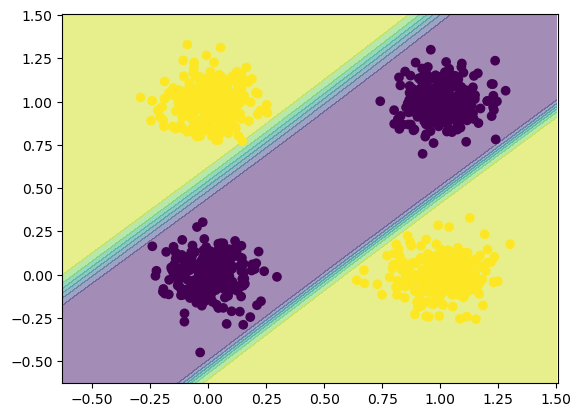


Рис. 5 – полученные области классификации.

Поскольку вначале графика потерь видим плато, можно говорить о потенциальном сокращении числа итераций до нахождения минимума, путём увеличения learning rate.

|  |  |  |
| --- | --- | --- |
| **C:\Users\ksarb\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A21FE299.tmp** | C:\Users\ksarb\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DB7334F5.tmp |  |
| Рис. 6 – функция ошибок с lr 1e-2 | Рис. 7 – функция ошибок с lr 2e-3 |  |

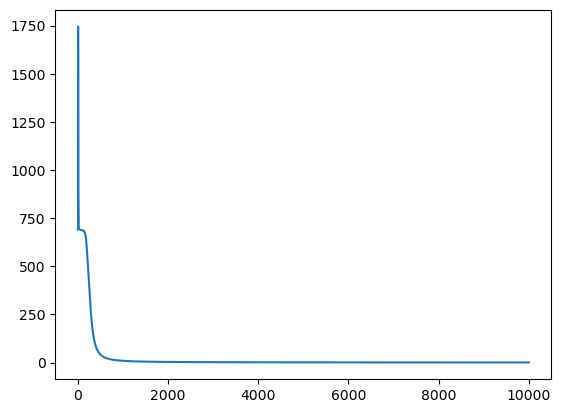


Рис. 8 – функция ошибок с lr 3e-3

Для learning rate 0,003 (рис. 8) видим почти мгновенное начало уменьшения функции ошибок, причём не попадаем в локальный минимум, наблюдаемый при lr1e-2 (рис.6), и не наблюдая плато, как на рис.7.

Проверка для lr 3e-3:

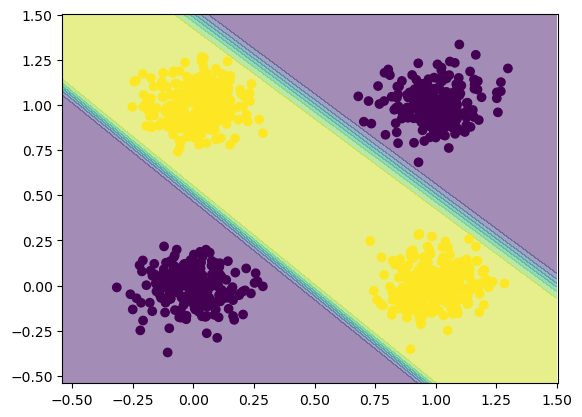


Рис. 9 – полученные области для lr 3e-3.

Сравнивая рис. 5 и рис. 9 можно наблюдать один из главных плюсов нейронных сетей – возможность поиска оптимального решения несколькими путями. Поскольку наши классы расположены симметрично относительно вращения на 180° относительно точки (0.5, 0.5), то и найденные решения также симметричны.

# Часть 3. Классификация изображений CIFAR100

Вариант: CLASSES = [39, 70, 35]

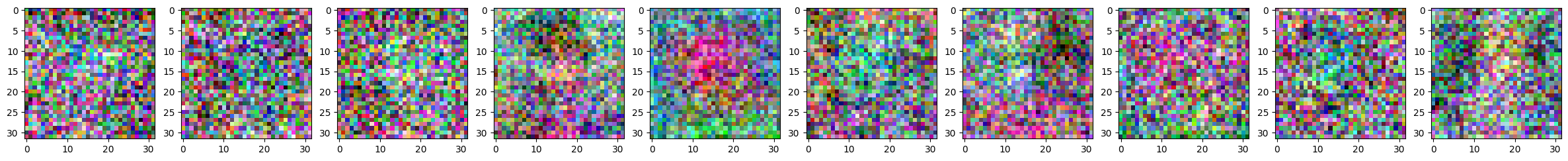


Рис. 10 – визуализация весов базовой модели.

Для базового варианта получим метрики, указанные в таб. 1., визуализация весов на рис.10.

Таблица 1 – базовые метрики модели.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выборка | Класс | precision | recall | f1-score |
| Train | клавиатура | 0.9959 | 0.9680 | 0.9817 |
| Train | роза | 0.9682 | 0.9740 | 0.9711 |
| Train | девочка | 0.9746 | 0.9960 | 0.9852 |
| Test | клавиатура | 0.7683 | 0.6300 | 0.6923 |
| Test | роза | 0.7333 | 0.6600 | 0.6947 |
| Test | девочка | 0.6875 | 0.8800 | 0.7719 |

По значениям видно, что метрики для классов клавиатура и девочка сильно падают при проверке на тестовой выборке. Общая точность на тестовой выборке – 71%.

Вариант: CLASSES = [39, 70, 35]

Класс keyboard (клавиатура).

Класс с rose (роза).

Класс с girl (девочка).

Вероятно, происходит либо недообучение, либо переобучение. Исследуя output (рис. 11) получим min(val\_loss) на 50 эпохе, значение val\_loss = 0.515.

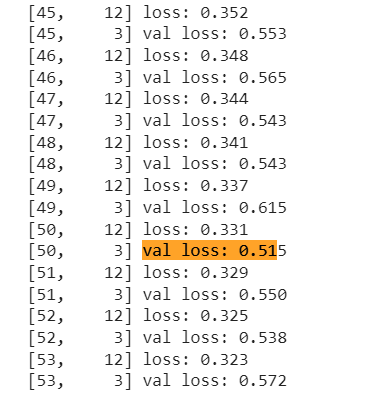


Рис. 11 – значения val loss для базовой модели.

Соответственно, обучим модель до границы переобучения – установим кол-во эпох на 50. Получим точность test = 75%, train = 90%, т.е. точность немного снизилась.

Постараемся выполнить улучшение НС с помощью уменьшения скорости обучения в 5 раз. Соответственно увеличим кол-во эпох в 5 раз. Полученная модель имеет точность 75%. При поиске по val\_loss обнаружим минимум в 100 – используем его как новое количество эпох.

При lr = 0.001, batch\_size = 128, epochs = 100 получаем точность 73% - максимум на текущий момент. Попробуем увеличить в 1.5 раза батч и количество эпох, для сохранения итераций.

При увеличении батча до 192 и количества итераций до 150 значимого увеличения точности не произошло.

Теперь обучим НС FC(10), FC(3) и FC(30), FC(10), FC(3).

Таблица 2 – итоговая таблица.

| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| --- | --- | --- | --- |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 77%, train = 99% | Базовый вариант |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 50 | test = 75%, train = 90% | В базовой модели было переобучение – уменьшили количество эпох. |
| FC(10), FC(3) | lr = 0.001, batch\_size = 128, epochs = 250 | test = 75%, train = 89% | Уменьшили learning rate для повышения точности. Соответственно увеличили количество эпох |
| FC(30), FC(10), FC(3). | lr = 0.005, batch\_size = 128, epochs = 250 | test = 80%, train = 99% | Создали новую модель FC(30), FC(10), FC(3). |
| FC(30), FC(10), FC(3). | lr = 0.001, batch\_size = 128, epochs = 550 | test = 81%, train = 95% | Приведение к оптимальным гиперпараметрам. |

**Вывод:** при обучении нейронных сетей очень важны гиперпараметры. Очень важно грамотно определить момент наступления переобучения, и по возможности уменьшить learning rate – это приводит к увеличению качества НС.

Класс клавиатура определяется хуже всего (recall = 0.63), из-за того, что на многих картинках в датасете клавиатура со всевозможными картинками и из-за размытие клавиатур в данном классе сложнее определить признаки т.к. отсутствует точность признаков. Класс девочка определяется лучше всего (recall = 0.88), т.к обладает явными простыми признаками т.к. там девочка и больше признаков.

Самым эффективным способом увеличения точности распознавании оказалось усложнение нейронной сети, было увеличено кол-во скрытых слоев и кол-ва нейронов в самих слоях. После приведения гиперпарамметров этой, но к оптимальным

(lr = 0.001, batch\_size = 128, epochs = 550) была достигнута максимальная точность 81%. Это согласуется с теорией.

есть набор данных где 50000 в обучающей выборке в тестовой 10000 каждое изображение цветное 32\*32 пикселя, необходимо определить общее количество итераций обучения нейронной сети 200 эпох в размером батча 250 скорость обучения 1/100

nn.Linear(32\*32\*3, 8),

            nn.ReLU(),

            nn.Linear(8, 8),

            nn.ReLU(),

            nn.Linear(8, classes),

в нейроне сети 2 скрытх слоя по 10 нейронов, нейронная сеть для классификации изображений сшафк100 указать количество нейронов по слоям количество параметров и количество весов

cifar100 200 эпох определить размер батча если было совершенно 20 000 итераций

| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| --- | --- | --- | --- |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 77%, train = 99% | Базовый вариант |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 30 | test = 72%, train = 84% | В базовой модели было переобучение – уменьшили количество эпох. |
| FC(10), FC(3) | lr = 0.001, batch\_size = 128, epochs = 170 | test = 73%, train = 82% | Уменьшили learning rate для повышения точности. Соответственно увеличили количество эпох |
| FC(10), FC(3) | lr = 0.001, batch\_size = 128, epochs = 100 | test = 72%, train = 79% | Обнаружено переобучение для прошлой итерации. Снизили количество эпох до 100. |
| FC(10), FC(3) | lr = 0.001, batch\_size = 192, epochs = 100 | test = 74%, train = 80% | Увеличение батч сайз в 1.5 раз не дало значимого выигрыша. |
| FC(10), FC(3) | lr = 0.001, batch\_size = 192, epochs = 170 | test = 72%, train = 79% | Обнаружено переобучение для прошлой итерации. Снизили количество эпох до 170. |
| FC(8), FC(3) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 76%, train = 98% | Создали новую модель FC(8), FC(3). |
| FC(8), FC(3) | lr = 0.001, batch\_size = 192, epochs = 270 | test = 74%, train = 83% | Приведение к оптимальным гиперпараметрам. |
| FC(8), FC(8), FC(3) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 75%, train = 98% | Создали новую модель FC(8), FC(8), FC(3). |
| FC(8), FC(8), FC(3) | lr = 0.001, batch\_size = 128, epochs =565 | test = 75%, train = 87% | Приведение к оптимальным гиперпараметрам. |