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: \_\_\_\_Сорокин А.Э.\_\_\_\_

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 (клаиватура)
2. 10 + 56 = 66 (енот)
3. 10 + 21 = 31 (слон)

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

Выборка:

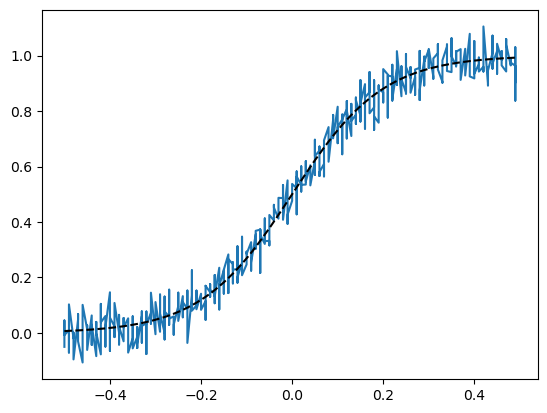


Рис. 1 – аппроксимируемая функция.

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).

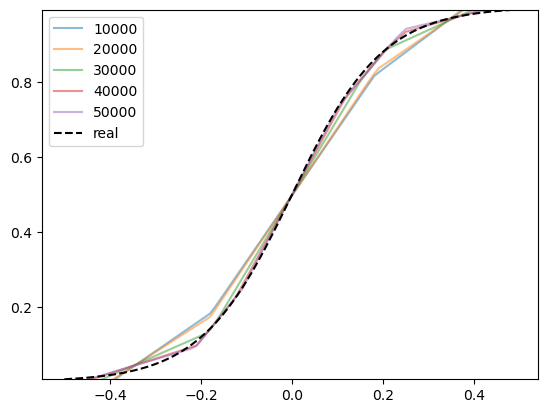


Рис. 2 – результат аппроксимации.

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.

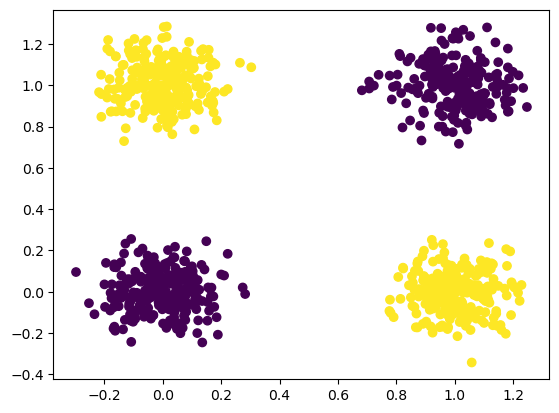


Рис. 3 – график классов.

Функция потерь – логарифмическая

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

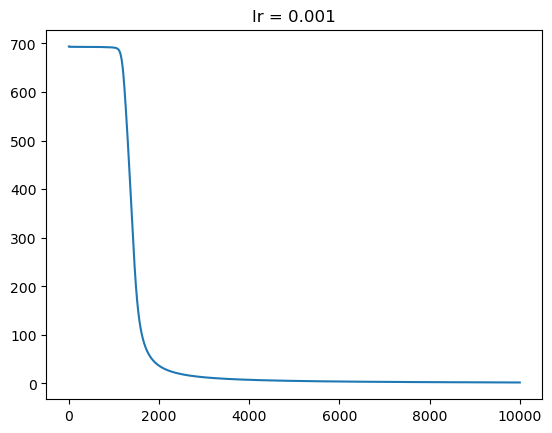


Рис. 4 – график общей функции потерь.

По рисунку видно, что после около 1.7 тысяч итераций НС начала сильно уменьшать функцию потерь, т.е. было найдено верное направление градиента. После 6000 итераций функция потерь находится у минимума. На рис.5 изображён график вывода нейронной сети в области [-0.5;0.5]

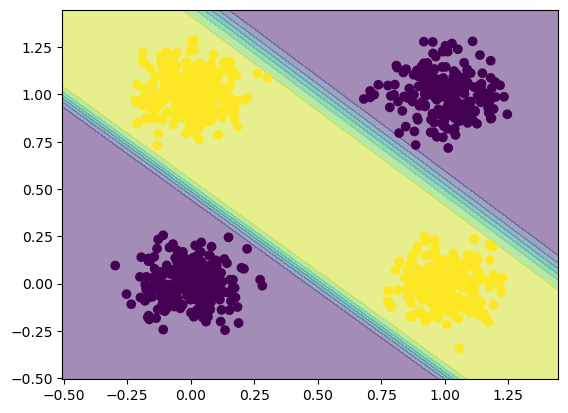


Рис. 5 – полученные области классификации.

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рис. 6 – функция ошибок с lr 2e-3 | Рис. 7 – функция ошибок с lr 5e-3 |  |

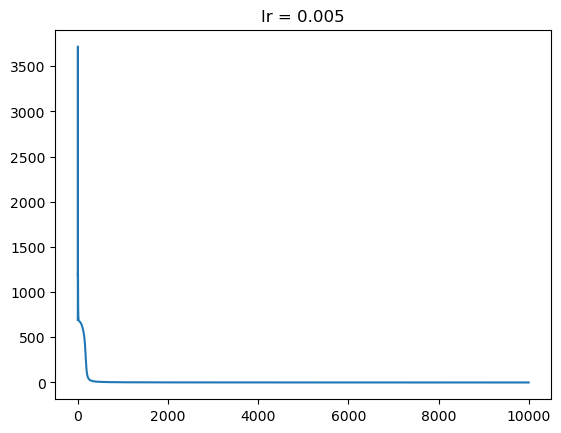


Рис. 8 – функция ошибок с lr 1e-2

Для learning rate 0,005 (рис. 8) видим почти мгновенное начало уменьшения функции ошибок, причём не попадаем в локальный минимум, наблюдаемый при lr2e-3 (рис.6), и не наблюдаем колебания, как на рис.7.

Проверка для lr 3e-3:

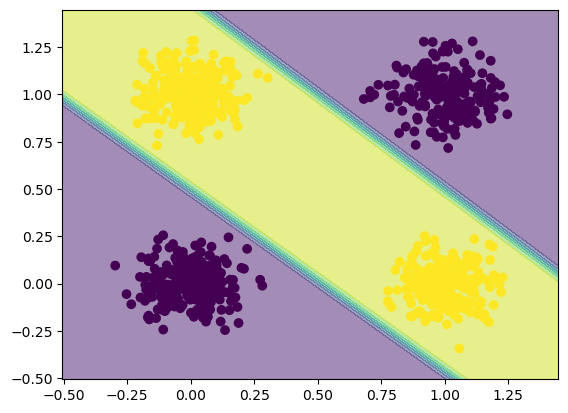


Рис. 9 – полученные области для lr 3e-3.

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, 66, 31]

Клавиатуры, енот, слон

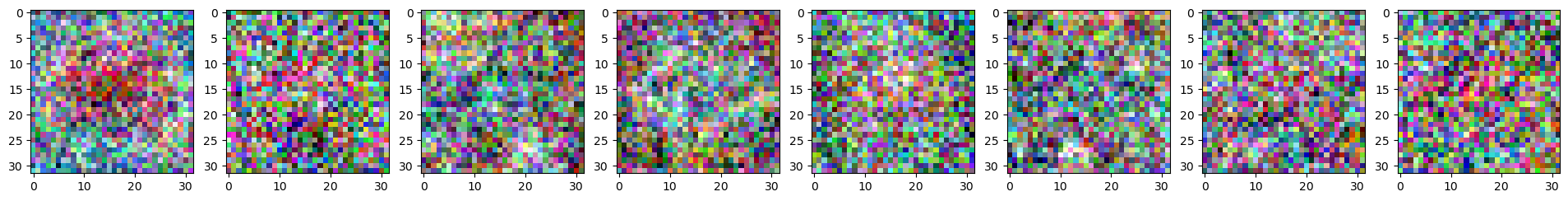


Рис. 10 – визуализация весов базовой модели.

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 | Название |
| Train | 39 | 0.9780 | 0.9800 | 0.9790 | Клавиатуры |
| Train | 61 | 0.9781 | 0.9820 | 0.9800 | Енот |
| Train | 26 | 0.9839 | 0.9780 | 0.9809 | Слон |
| Test | 39 | 0.7216 | 0.7000 | 0.7107 | Клавиатуры |
| Test | 61 | 0.7172 | 0.7100 | 0.7136 | Енот |
| Test | 26 | 0.6154 | 0.6400 | 0.6275 | Слон |

Общая точность на обучающей выборке 98%, а тестовой выборке 70%. Хуже всего распознался класс со слоном, среди метрик этого класса самый низкий recall, это значит что на метрики влияет то, что класс не обнаруживается в принципе.

На точность могло повлиять, что нп изображениях со слонами само животное попадет обрезано, только частично, что приводит к ошибкам и отнесению изображения к классу енотов. Лучше всего НС распознает енотов, т.к. ей проще определять «серые пятна» с характерной формой мордочки. Примеры фото из классов приведены на рис 11-13



Рис. 11 – класс с клавиатурами.



Рис. 12 – класс со слоном.



Рис. 13 – класс с енотом.

Вероятно, происходит переобучение. Исследуя output (рис. 15) получим min(val\_loss) на 63 эпохе, значение val\_loss = 0.667.

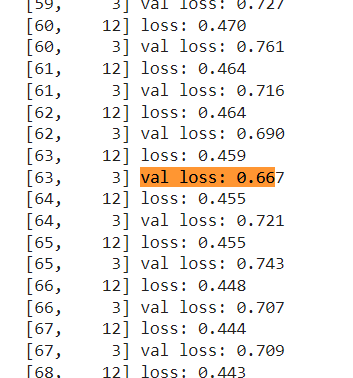


Рис. 14 – значения val loss для базовой модели.

Соответственно, обучим модель до границы переобучения – установим кол-во эпох на 63. Получим точность test = 70%, train = 85%, т.е. точность немного стала лучше.

Постараемся выполнить улучшение НС с помощью уменьшения скорости обучения в 5 раз. Соответственно увеличим кол-во эпох в 5 раз. Полученная модель имеет точность 70%. При поиске по val\_loss обнаружим минимум в 50 – используем его как новое количество эпох.

При lr = 0.001, batch\_size = 128, epochs = 300 получаем точность 71% - максимум на текущий момент. Попробуем увеличить добавить кол-во слоев и подобрать лучшие гиперпараметры. В модели FC(30), FC(30) при значениях lr = 0.005, batch\_size = 256, epochs =156 получаем наилучший результат 73%.

Таблица 2 – итоговая таблица.

| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| --- | --- | --- | --- |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 69%, train = 98% | Базовый вариант |
| FC(10), FC(3) | lr = 0.005, batch\_size = 128, epochs = 63 | test = 70%, train = 85% | В базовой модели было переобучение – уменьшили количество эпох. |
| FC(10), FC(3) | lr = 0.001, batch\_size = 128, epochs = 315 | test = 71%, train = 82% | Уменьшили learning rate для повышения точности. Соответственно увеличили количество эпох. |
| FC(30), FC(10) | lr = 0.005, batch\_size = 128, epochs = 250 | test = 71%, train = 98% | Добавим больше нейронов 1 и 2 слой |
| FC(30), FC(10) | lr = 0.001, batch\_size = 128, epochs = 420 | test = 72%, train = 81% | Обнаружено переобучение для прошлой итерации. Снизим скорость и компенсируем кол-во эпох. |
| FC(30), FC(10) | lr = 0.005, batch\_size = 128, epochs = 84 | test = 70%, train = 83% | Попробуем поднять скорость обучения |
| FC(30), FC(30) | lr = 0.005, batch\_size = 128, epochs = 84 | test = 71%, train = 83% | Добавим ещё нейронов во второй слой |
| FC(30), FC(30) | lr = 0.001, batch\_size = 128, epochs = 420 | test = 72%, train = 82% | Обнаружено переобучение для прошлой итерации. Снизим скорость и компенсируем кол-во эпох. |
| FC(30), FC(30) | lr = 0.005, batch\_size = 256, epochs =156 | test= 73%,  train = 94% | Приведение к оптимальным гиперпараметрам. |

**Вывод**

Во время выполнения лабораторной работы мы рассмотрели архитектуру нашей полносвязной НС, а также научились с помощью гиперпарметров повышать точность.

Класс со слонами определяется хуже всего из-за того, что на многих картинках в датасете слоны обрезаны и часто скрыты за листвой. Енот определяется лучше всего, т.к. обладает явными простыми признаками такими как цвет и форма мордочки.

Изменение гиперпараметров и добавление нейронов в слои позволило увеличить общую точность обучения, что согласуется с теорией. Самым эффективным способом увеличения точности распознавании оказалось усложнение нейронной сети, а также уменьшение эпох до момента переобучения, что согласуется с теорией.