

## Задания по лекции

### Задание 1. Работа с тензорами

1. Создайте два тензора: один с равномерным распределением значений, другой — с нормальным.
  2. Выполните между ними элементные арифметические операции: сложение, умножение.
  3. Преобразуйте результат в массив NumPy и выведите его.
- 

### Задание 2. Индексация и манипуляции с тензорами

1. Создайте тензор размера  $4 \times 4$ , заполненный случайными числами.
  2. Получите центральный  $2 \times 2$  блок, выделив его срезами.
  3. Измените значения элементов в этом блоке на 1.
- 

### Задание 3. Работа с градиентами

1. Создайте тензор  $x = 3.0$  с `requires_grad=True`.
  2. Определите функцию  $y = 5x^3 - 2x^2 + x + 1$ .
  3. Вычислите градиент функции  $y$  в точке  $x = 3.0$ .
- 

### Задание 4. Создание собственной нейронной сети

1. Реализуйте класс нейронной сети с двумя полносвязными слоями (`nn.Linear`) и активацией ReLU.
  2. Задайте входное и выходное количество нейронов: 10 и 2 соответственно.
  3. Проверьте, что модель возвращает выходной тензор нужной размерности, передав случайный вход.
- 

### Задание 5. Обучение нейронной сети на наборе данных MNIST

1. Загрузите данные MNIST с использованием `torchvision.datasets`.
  2. Нормализуйте данные и создайте DataLoader.
  3. Обучите простую нейронную сеть с двумя линейными слоями (используйте ReLU и CrossEntropyLoss) на этих данных.
- 

### Задание 6. Добавление Dropout

1. Расширьте модель из задания 5, добавив слой Dropout между слоями.
  2. Обучите новую модель и сравните точность на тестовом наборе с базовой моделью.
- 

### Задание 7. Аугментация данных

1. Реализуйте трансформации для набора данных MNIST:
  - Случайные повороты (до 15 градусов).

- Случайные горизонтальные отражения.
2. Проверьте, как изменились данные, визуализировав первые 5 изображений после аугментации.
- 

### Задание 8. Оптимизаторы

1. Создайте модель и обучите её с использованием SGD с моментумом.
  2. Повторите обучение, используя Adam. Сравните время обучения и точность моделей.
- 

### Задание 9. Визуализация с помощью TensorBoard

1. Добавьте использование TensorBoard в процессе обучения модели из задания 5.
  2. Визуализируйте графики функции потерь и точности на обучающей и тестовой выборках.
- 

### Задание 10. Пользовательский датасет

1. Создайте пользовательский датасет, содержащий массивы чисел и их суммы.
2. Реализуйте класс, наследующий `torch.utils.data.Dataset`, для обработки этих данных.
3. Напишите скрипт, который обучит модель предсказывать сумму чисел.

### Дополнительное задание со звездочкой

#### Задание 11. Реализация сверточной нейронной сети (CNN) для классификации изображений\*

1. Реализуйте сверточную нейронную сеть с архитектурой:
  - Входной сверточный слой: 16 фильтров, ядро  $3 \times 3$ , активация ReLU.
  - Слой подвыборки (MaxPooling) с ядром  $2 \times 2$ .
  - Второй сверточный слой: 32 фильтра, ядро  $3 \times 3$ , активация ReLU.
  - Слой подвыборки (MaxPooling) с ядром  $2 \times 2$ .
  - Полносвязный слой с 128 нейронами и активацией ReLU.
  - Полносвязный выходной слой с количеством классов, равным числу категорий в наборе данных.
2. Используйте набор данных CIFAR-10 (или любой другой похожий набор) для обучения модели.
3. Добавьте поддержку использования GPU (если доступно) для ускорения обучения.
4. Визуализируйте:
  - Карты активаций первых двух сверточных слоев для нескольких изображений из тестовой выборки.
  - Матрицу ошибок (confusion matrix) на тестовых данных.