

Практическое задание №6

Вариант 1.

Задание

Необходимо построить сверточную нейронную сеть, которая будет классифицировать черно-белые изображения с простыми геометрическими фигурами на них.

К каждому варианту прилагается код, который генерирует изображения.

Для генерации данных необходимо вызвать функцию `gen_data`, которая возвращает два тензора:

Тензор с изображениями ранга 3

Тензор с метками классов

Обратите внимание:

Выборки не перемешаны, то есть наблюдения классов идут по порядку

Классы характеризуются строковой меткой

Выборка изначально не разбита на обучающую, контрольную и тестовую

Скачивать необходимо оба файла. Подключать файл, который начинается с `var` (в нем и находится функция `gen_data`)

Классификация изображений с прямоугольником или кругом.

Решение

Была построена модель сверточной нейронной сети со следующими слоями

Тип слоя	Параметры
Convolution2D	32 фильтра, ядро свертки 3x3, функция активации - Relu, input_shape - (50, 50, 1)
MaxPooling2D	Размер пула - (2, 2)
Dropout	Вероятность отключения нейрона - 0.2
Convolution2D	64 фильтра, ядро свертки 2x2, функция активации - Relu
MaxPooling2D	Размер пула - (2, 2)
Flatten	
Dropout	Вероятность отключения нейрона - 0.5
Dense	1 нейрон, функция активации - sigmoid

Так как задача классификации бинарная, то используется 1 нейрон на последнем слое. Также используются параметры обучения:

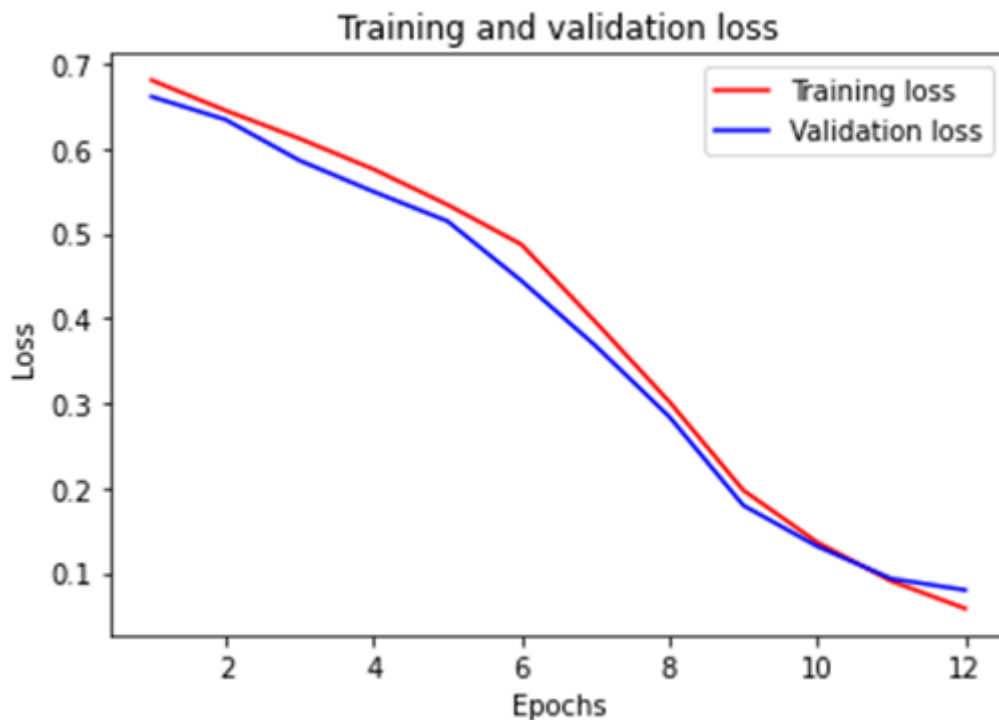
- Функция потерь - `binary_crossentropy`, подходит для бинарной классификации
- Оптимизатор - `adam`
- Число эпох - 12
- Размер батча - 16

- Соотношение тестовых данных к обучающим - 20/80
- validation_split = 0.2

Перед обучением модели данные "размешиваются" с помощью `np.random.shuffle`. Далее разбиваются на обучающие и тестовые данные.

Результаты обучения модели

Ниже приведены графики точности и потерь:



Полученные данные при вызове функции `evaluate` после обучения:

loss: 0.0207 - accuracy: 0.9937

Как видно, модель достигла точности >99% на тестовых данных.

