

Лабораторная работа №3. Реализация сверточной нейронной сети

Данные: В работе предлагается использовать набор данных notMNIST, который состоит из изображений размерностью 28×28 первых 10 букв латинского алфавита (A ... J, соответственно). Обучающая выборка содержит порядка 500 тыс. изображений, а тестовая – около 19 тыс.

Ход работы:

Для реализации лабораторной работы используется библиотека **Swift for TensorFlow** <https://github.com/tensorflow/swift>

Исходный код: <https://github.com/Stunba/MachineLearning2/tree/master/lab3/>

Код структуры модели с 2умя сверточными слоями:

```
struct SimpleConvolutionalModel: Layer {
    var conv1 = Conv2D<Float>(filterShape: (5, 5, 1, 2), padding: .same, activation: relu)
    var conv2 = Conv2D<Float>(filterShape: (5, 5, 2, 4), activation: relu)
    var flatten = Flatten<Float>()
    var fc = Dense<Float>(inputSize: 2304, outputSize: 10)

    @differentiable
    public func callAsFunction(_ input: Tensor<Float>) -> Tensor<Float> {
        let convolved = input.sequenced(through: conv1, conv2)
        return convolved.sequenced(through: flatten, fc)
    }
}
```

Точность данной модели 94%.

Код структуры модели со сверточным слоем и слоем пулинга:

```
struct ConvolutionalWithPooling: Layer {
    var conv = Conv2D<Float>(filterShape: (5, 5, 1, 4), padding: .same, activation: relu)
    var pool = AvgPool2D<Float>(poolSize: (2, 2), strides: (2, 2))
    var flatten = Flatten<Float>()
    var fc = Dense<Float>(inputSize: 784, outputSize: 10)

    @differentiable
    public func callAsFunction(_ input: Tensor<Float>) -> Tensor<Float> {
        let convolved = input.sequenced(through: conv, pool)
        return convolved.sequenced(through: flatten, fc)
    }
}
```

Точность данной модели 94%.

Реализация модели LeNet-5 (<http://yann.lecun.com/exdb/lenet/>).

```

struct LeNet: Layer {
    var conv1 = Conv2D<Float>(filterShape: (5, 5, 1, 6), padding: .same, activation: relu)
    var pool1 = AvgPool2D<Float>(poolSize: (2, 2), strides: (2, 2))
    var conv2 = Conv2D<Float>(filterShape: (5, 5, 6, 16), activation: relu)
    var pool2 = AvgPool2D<Float>(poolSize: (2, 2), strides: (2, 2))
    var flatten = Flatten<Float>()
    var fc1 = Dense<Float>(inputSize: 400, outputSize: 120, activation: relu)
    var fc2 = Dense<Float>(inputSize: 120, outputSize: 84, activation: relu)
    var fc3 = Dense<Float>(inputSize: 84, outputSize: 10)

    public init() {}

    @differentiable
    public func callAsFunction(_ input: Tensor<Float>) -> Tensor<Float> {
        let convolved = input.sequenced(through: conv1, pool1, conv2, pool2)
        return convolved.sequenced(through: flatten, fc1, fc2, fc3)
    }
}

```

Точность данной модели 96%.

Используя простую логистическую регрессию получили модель с точностью 85%.

Используя полносвязную глубокую сеть получили точность 95%. Для модели LeNet-5 получили точность 96%.

Вывод:

В данной лабораторной работе была построена и обучена сверточная нейронная сеть используя набор данных notMNIST, реализована модель сети LeNet-5.