Министерство образования Республики Беларусь

Учреждение образования

"Белорусский Государственный университет информатики

и радиоэлектроники"

Лабораторная работа №3

“**Реализация сверточной нейронной сети”**

по учебной дисциплине “Машинное обучение”

|  |  |
| --- | --- |
| Выполнил: | Студент гр. 956241 Дубовик Н.О. |
|  |  |

Минск 2020

**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.

Данные можно скачать по ссылке:

* <https://commondatastorage.googleapis.com/books1000/notMNIST_large.tar.gz> (большой набор данных);
* [https://commondatastorage.googleapis.com/books1000/notMNIST\_small.tar.gz](https://commondatastorage.googleapis.com/books1000/notMNIST_large.tar.gz) (маленький набор данных);

Описание данных на английском языке доступно по ссылке:

<http://yaroslavvb.blogspot.sg/2011/09/notmnist-dataset.html>

Результат выполнения заданий опишите в отчете.

**Задание 1.**

Реализуйте нейронную сеть с двумя сверточными слоями, и одним полносвязным с нейронами с кусочно-линейной функцией активации. Какова точность построенное модели?

Была создана нейронная сеть, состоящая из двух сверточных слоев, с фильтрами размера 3х3 и функцией активации ReLU. Первой сверточный слой состоит из 8 фильтров, в то время как второй из 16.

Второй сверточный слой связан с полносвязным слоем, состоящим из 128 нейронов с функцией активации ReLU.

Выходной слой состоит из 10 нейронов с функцией активации “softmax”.

Был использован оптимизатор Adam и функция потерь “sparse\_categorical\_crossentropy”.

Реализация, здесь количество эпох равно 50:

def train\_conv(x\_train,y\_train,epochs,img\_height,img\_width,save\_path):

    model=tf.keras.models.Sequential([

        tf.keras.layers.Conv2D(8,(3,3),activation='relu',input\_shape=(img\_height, img\_width,1)),

        tf.keras.layers.Conv2D(16,(3,3),activation='relu'),

        tf.keras.layers.Flatten(),

        tf.keras.layers.Dense(128, activation='relu'),

        tf.keras.layers.Dense(10, activation='softmax')

    ])

    model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

    model.fit(x\_train, y\_train, epochs=epochs)

    model.save(save\_path)

После обучения получились следующие показатели: 99.41% и 92.48% на обучающей и тестовой выборках соответственно.

**Задание 2.**

Замените один из сверточных слоев на слой, реализующий операцию пулинга (Pooling) с функцией максимума или среднего. Как это повлияло на точность классификатора?

def train\_conv\_pool(x\_train,y\_train,epochs,img\_height,img\_width,save\_path):

    model=tf.keras.models.Sequential([

        tf.keras.layers.Conv2D(8,(3,3),activation='relu',input\_shape=(img\_height, img\_width,1)),

        tf.keras.layers.AveragePooling2D(),

        tf.keras.layers.Flatten(),

        tf.keras.layers.Dense(128, activation='relu'),

        tf.keras.layers.Dense(10, activation='softmax')

    ])

    model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

    model.fit(x\_train, y\_train, epochs=epochs)

    model.save(save\_path)

После замены были достигнуты следующие показатели: 98.41% на обучающей и 91.73% на тестовой выборке.

**Задание 3.**

Реализуйте классическую архитектуру сверточных сетей LeNet-5 (http://yann.lecun.com/exdb/lenet/).

Архитектура реализованной сети получилась следующая:

* сверточный слой с 6-ю фильтрами размера 5х5 и функцией активацией ReLU;
* слой, реализующий операцию пулинга, с функцией среднего;
* сверточный слой с 16-ю фильтрами размера 5х5 и функцией активацией ReLU;
* слой, реализующий операцию пулинга, с функцией среднего;
* полносвязанный слой с 120 нейронами и функцией активацией ReLU;
* полносвязанный слой с 84 нейронами и функцией активацией ReLU;
* выходной слой с 10 нейронами и функцией активацией "softmax".

def train\_LeNet5(x\_train,y\_train,epochs,save\_path):

    model=tf.keras.models.Sequential([

        tf.keras.layers.Conv2D(filters=6, kernel\_size=(5, 5), activation='relu', input\_shape=(32,32,1)),

        tf.keras.layers.AveragePooling2D(),

        tf.keras.layers.Conv2D(filters=16, kernel\_size=(5, 5), activation='relu'),

        tf.keras.layers.AveragePooling2D(),

        tf.keras.layers.Flatten(),

        tf.keras.layers.Dense(120, activation='relu'),

        tf.keras.layers.Dense(84, activation='relu'),

        tf.keras.layers.Dense(10, activation='softmax')

    ])

    model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

    model.fit(x\_train, y\_train, epochs=epochs)

    model.save(save\_path)

После обучения получились следующие показатели: 95.58% и 92.77% на обучающей и тестовой выборках соответственно.

**Задание 4.**

Сравните максимальные точности моделей, построенных в лабораторных работах 1-3. Как можно объяснить полученные различия?

Исходя их показателей, полученных в ходе лабораторных работ, можно сделать вывод, что сверточные нейронный сети показывают лучший результат из-за более точного извлечения свойств изображения