

Министерство образования Республики Беларусь
Учреждение образования
"Белорусский Государственный университет информатики
и радиоэлектроники"

Лабораторная работа №6
**“Применение сверточных нейронных сетей (многоклассовая
классификация)”**
по учебной дисциплине “Машинное обучение”

Выполнил:

Студент гр. 956241 Дубовик Н.О.

Минск 2020

Данные: Набор данных для распознавания языка жестов, который состоит из изображений размерности 28x28 в оттенках серого (значение пикселя от 0 до 255). Каждое из изображений обозначает букву латинского алфавита, обозначенную с помощью жеста, как показано на рисунке ниже (рисунок цветной, а изображения в наборе данных в оттенках серого). Обучающая выборка включает в себя 27,455 изображений, а контрольная выборка содержит 7172 изображения. Данные в виде csv-файлов можно скачать на сайте Kaggle -> <https://www.kaggle.com/datamunge/sign-language-mnist>

Результат выполнения заданий опишите в отчете.

Задание 1.

Загрузите данные. Разделите исходный набор данных на обучающую и валидационную выборки.

Данные были разделены в отношении 80 к 20.

Задание 2.

Реализуйте глубокую нейронную сеть со сверточными слоями. Какое качество классификации получено? Какая архитектура сети была использована?

Была создана следующая архитектура нейронной сети:

- 3 сверточных слоя с функцией активации ReLU в каждом;
- в каждом слое есть фильтры размера 3x3, первый слой содержит 8 фильтров, второй содержит 16 фильтров и третий слой содержит 32 фильтра;
- каждый сверточный слой связан с пулингом с функцией среднего;
- третий сверточный слой связан с полносвязным слоем, с функцией активацией ReLU и содержащий 128 нейронов;
- выходной слой состоит из 25 нейронов с функцией активацией “softmax”;
- использовался оптимизатор Adam и функция потерь `sparse_categorical_crossentropy`.

Реализация, данная модель обучалась на протяжении 30 эпох:

```
model=tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(8,(3,3),activation='relu',input_shape=(IMAGE_WIDTH,
    IMAGE_HEIGHT, IMAGE_CHANNELS)),
    tf.keras.layers.AveragePooling2D(),
```

```

tf.keras.layers.Conv2D(16,(3,3),activation='relu'),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.AveragePooling2D(),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.AveragePooling2D(),

tf.keras.layers.Flatten(),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dense(25, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

```

Получены следующие результаты: 95.01% на обучающей выборке и 99.06% на валидационной.

Задание 3.

Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?

Были использованы техники вращения и сдвига изображения:

```

img_rotation_generator=tf.keras.preprocessing.image.ImageDataGenerator(rotation_range=90)
img_shear_generator=tf.keras.preprocessing.image.ImageDataGenerator(shear_range=45.0)

```

После чего были получены следующие результаты: 93.13% на обучающей выборке и 99.64% на валидационной.

Задание 4.

Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Можно ли было обойтись без него?

Какой максимальный результат удалось получить на контрольной выборке?

Было использовано передаточное обучение на архитектуре VGG16.

Из данной сети был убран исходящий слой и были добавлены пулинг слой с функцией среднего, и выходной слой из 25 нейронов с функцией активацией “softmax”. Набор данных обучался в течение 30 эпох.

```
base_model=tf.keras.applications.VGG16(  
    input_shape=(IMAGE_WIDTH,IMAGE_HEIGHT,IMAGE_CHANNELS),  
    include_top=False,  
    weights='imagenet'  
)  
base_model.trainable = False  
pretrained_model=tf.keras.models.Sequential([  
    base_model,  
    tf.keras.layers.GlobalAveragePooling2D(),  
    tf.keras.layers.Dense(25, activation='softmax')  
)  
  
pretrained_model.compile(optimizer='adam',  
    loss='sparse_categorical_crossentropy',  
    metrics=['accuracy'])
```

Были получены следующие результаты: точность на обучающей выборке составляет 97.72% и 97.78% на валидационной.