

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

Лабораторная работа №5
“Применение сверточных нейронных сетей (бинарная классификация)”
по учебной дисциплине “Машинное обучение”

Выполнил:

Студент гр. 956241 Дубовик Н.О.

Минск 2020

Данные: Набор данных DogsVsCats, который состоит из изображений различной размерности, содержащих фотографии собак и кошек. Обучающая выборка включает в себя 25 тыс. изображений (12,5 тыс. кошек: cat.0.jpg, ..., cat.12499.jpg и 12,5 тыс. собак: dog.0.jpg, ..., dog.12499.jpg), а контрольная выборка содержит 12,5 тыс. неразмеченных изображений. Скачать данные, а также проверить качество классификатора на тестовой выборке можно на сайте Kaggle -> <https://www.kaggle.com/c/dogs-vs-cats/data>

Результат выполнения заданий опишите в отчете.

Задание 1.

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

Данные были разделены в отношении 60/20/20.

Задание 2.

Реализуйте глубокую нейронную сеть с как минимум тремя сверточными слоями. Какое качество классификации получено?

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 32 фильтра, второй содержит 64 фильтра и третий слой содержит 128 фильтров;
- каждый сверточный слой связан с пулинг слоем с функцией среднего;
- третий сверточный слой связан с полносвязным слоем, с функцией активацией ReLU и содержащий 512 нейронов;
- выходной слой состоит из 2 нейронов с функцией активацией “softmax”;
- использовался оптимизатор Adam и функция потерь categorical_crossentropy.

Реализация, данная модель обучалась на протяжении 10 эпох:

```
tf.keras.layers.Conv2D(32,(3,3),activation='relu',input_shape=(IMAGE_WIDTH, IMAGE_HEIGHT, IMAGE_CHANNELS)),
tf.keras.layers.AveragePooling2D(),
tf.keras.layers.Dropout(0.2),

tf.keras.layers.Conv2D(64,(3,3),activation='relu'),
tf.keras.layers.AveragePooling2D(),
tf.keras.layers.Dropout(0.2),
```

```

tf.keras.layers.Conv2D(128,(3,3),activation='relu'),
tf.keras.layers.AveragePooling2D(),
tf.keras.layers.Dropout(0.2),

tf.keras.layers.Flatten(),
tf.keras.layers.Dense(512, activation='relu'),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.Dense(2, activation='softmax')
])

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

```

Получены следующие результаты: 98.72% на обучающей выборке, 78.55% на тестовой выборке и 74.06% на валидационной выборке, рисунок 1.

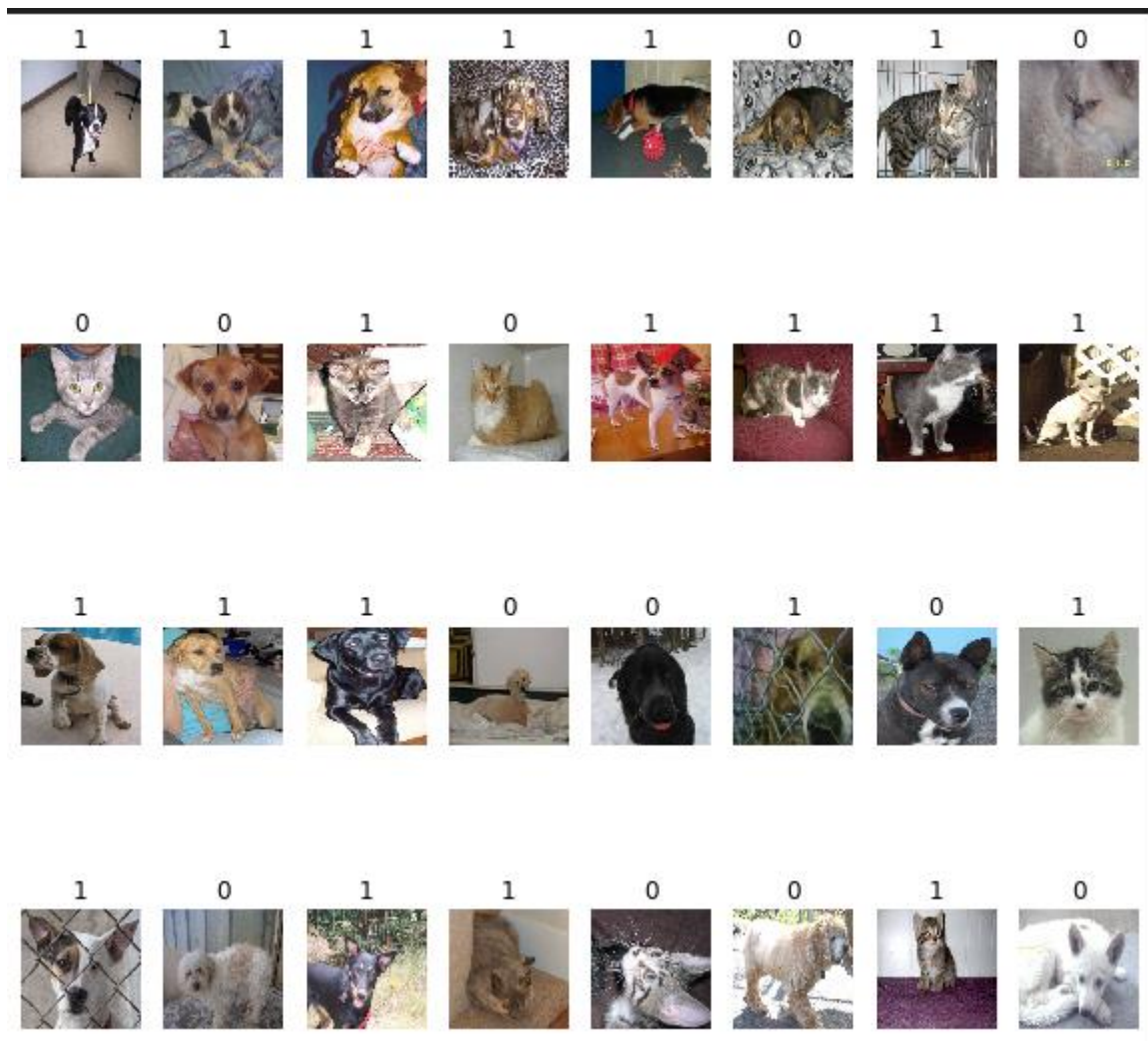


Рисунок 1 – Результат работы глубокой нейронной сети

Задание 3.

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

Были использованы техники вращения и сдвига изображения:

```
img_augmentation_generator=tf.keras.preprocessing.image.ImageDataGenerator(  
    rotation_range=45,  
    shear_range=2,  
    rescale=1./255  
)
```

После чего были получены следующие результаты: 96.64% на обучающей выборке, 89.6% на тестовой выборке и 85.16% на валидационной выборке, рисунок 2.

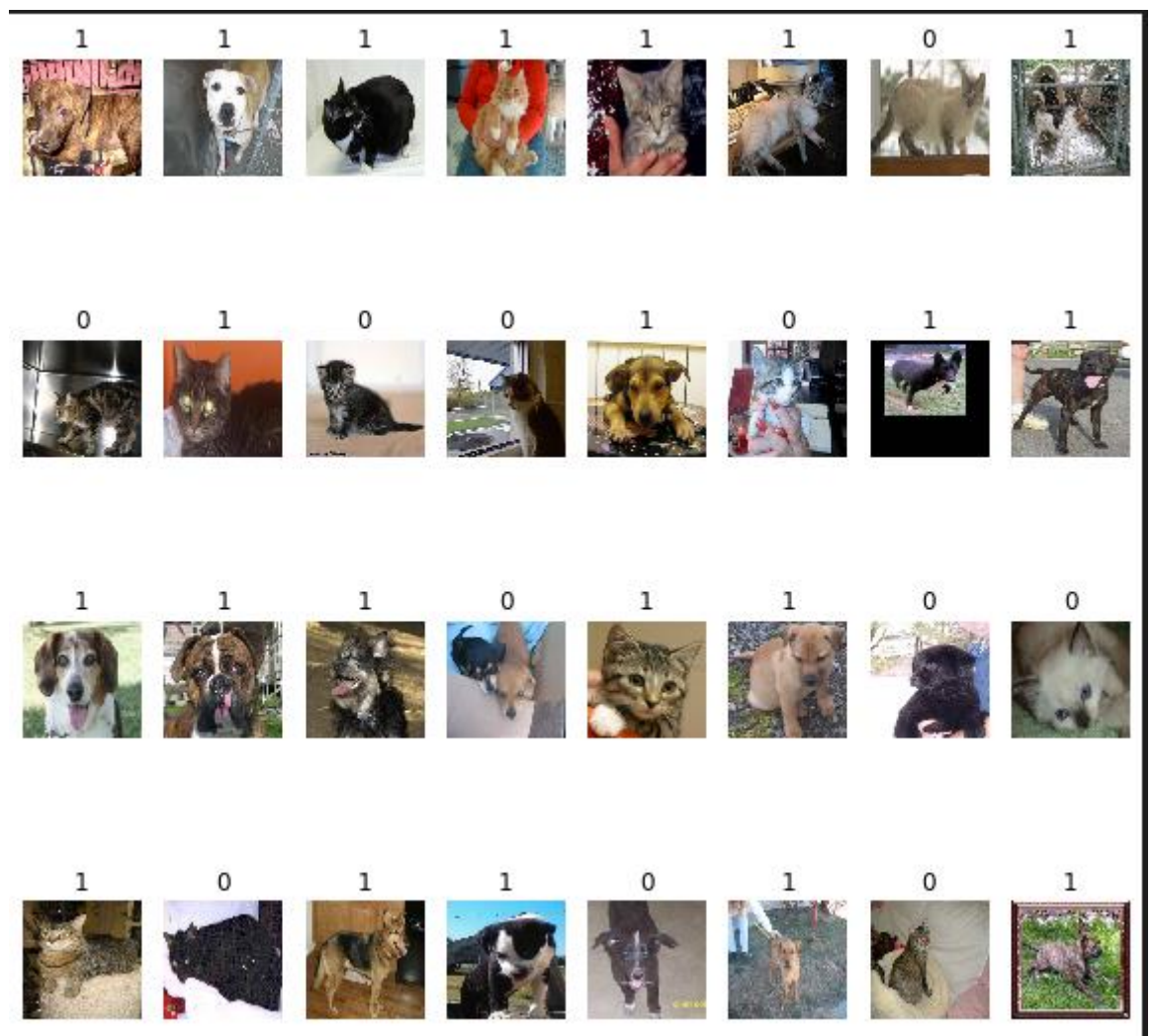


Рисунок 2 – Результат работы глубокой нейронной сети после дополнения данных

Задание 4.

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

Какой максимальный результат удалось получить на сайте Kaggle? Почему?

Было использовано передаточное обучение на архитектуре VGG16.

Из данной сети был убран исходящий слой и были добавлены полносвязанный слой, с функцией активацией ReLU и содержащий 512 нейронов, и выходной слой из 2 нейронов с функцией активацией “softmax”. Набор данных обучался в течение 10 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.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(2, activation='softmax')
])

pretrained_model.compile(optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'])
```

Были получены следующие результаты: точность на обучающей выборке составляет 98.03%, 95.34% на тестовой выборке и 91.5% на валидационной выборке, рисунок 3.

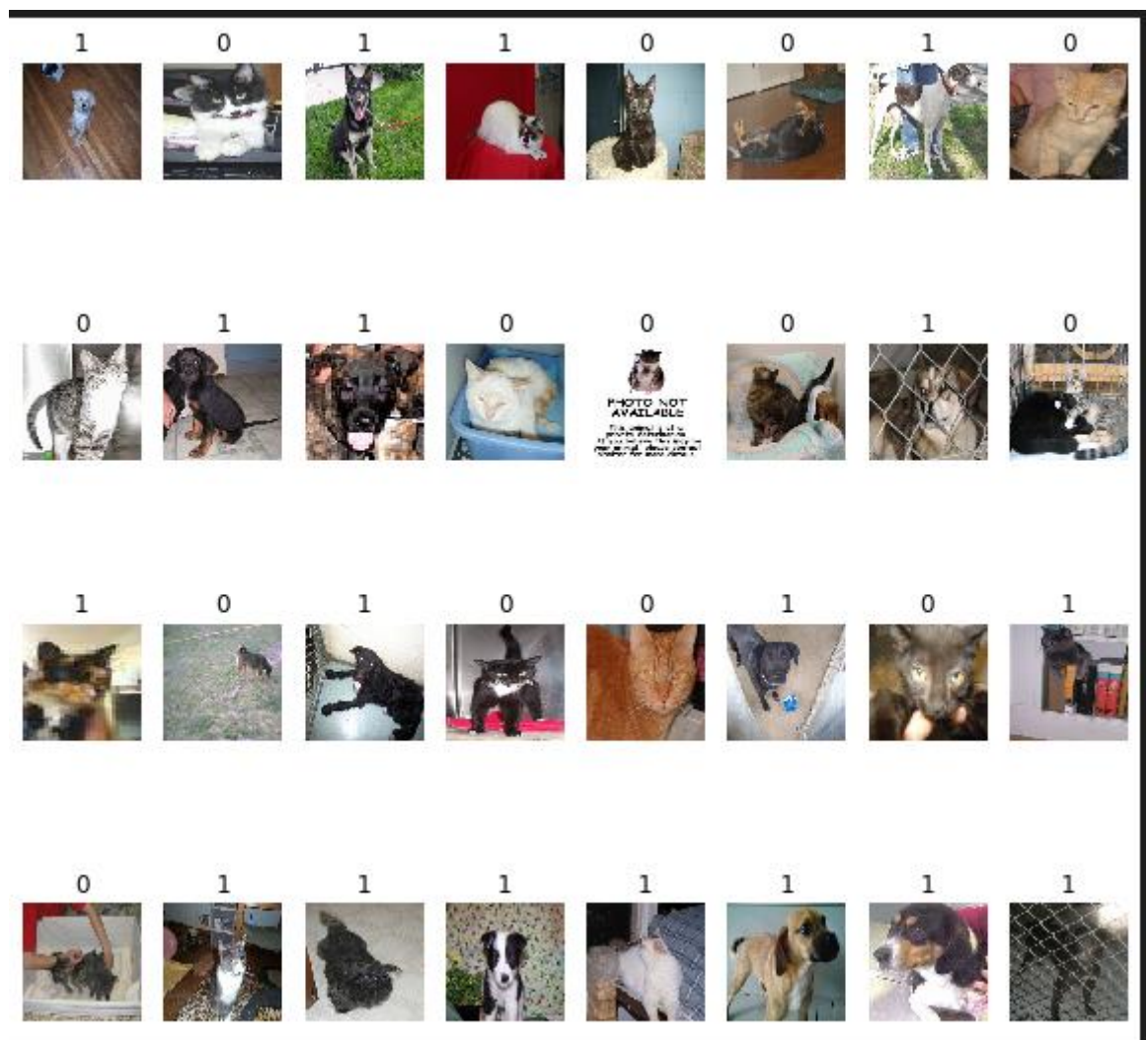


Рисунок 3 – Использование архитектуры VGG16