

Домашнее задание. Нейронные сети

Вопрос 1

Так как перед нами стоит задача бинарной классификации, какую функцию потерь лучше всего применить в нашем случае?

- **бинарная кросс-энтропия (binary crossentropy)**
- фокусные потери (focal loss)
- среднеквадратичная ошибка (mean squared error)
- категориальная кросс-энтропия (categorical crossentropy)

Примечание: Если вы используете активацию для выходного слоя, не требуется устанавливать `from_logits=True`.

Вопрос 2

Определите общее количество параметров в модели. Для этого примените метод `summary`.

- 9215873
- **11215873**
- 14215873
- 19215873

Model: "sequential_3"

| Layer (type) | Output Shape | Param # |
|---------------------------------------|----------------------|----------|
| conv2d_2 (Conv2D) | (None, 148, 148, 32) | 896 |
| max_pooling2d_2 (MaxPooling2D) | (None, 74, 74, 32) | 0 |
| flatten_2 (Flatten) | (None, 175232) | 0 |
| dense_4 (Dense) | (None, 64) | 11214912 |
| dense_5 (Dense) | (None, 1) | 65 |
| Total params: 11215873 (42.79 MB) | | |
| Trainable params: 11215873 (42.79 MB) | | |
| Non-trainable params: 0 (0.00 Byte) | | |

Генераторы и обучение

Для следующих двух вопросов используйте следующий генератор данных для обучающих (train) и тестовых (test) наборов:

```
ImageDataGenerator(rescale=1./255)
```

Примечание: Дополнительная предобработка изображений не требуется. При загрузке данных из каталогов обучения/тестирования убедитесь, что параметр `class_mode` установлен правильно для задачи бинарной классификации. Рекомендуемые параметры: `batch_size=20` и `shuffle=True`.

Для обучения примените метод `.fit()` со следующими параметрами:

```
model.fit(train_generator, epochs=10, validation_data=test_generator)
```

Вопрос 3

Какова медиана точности обучения по всем эпохам?

- 0.40
- 0.60 (0,59– мой ответ)
- 0.90
- 0.20
-

```
import numpy as np

# Получаем значения точности обучения для всех эпох
train_accuracy = history.history['accuracy']

# Вычисляем медиану точности обучения
median_train_accuracy = np.median(train_accuracy)

print("Медиана точности обучения по всем эпохам:", median_train_accuracy)
```

```
\ Медиана точности обучения по всем эпохам: 0.5939086079597473
```

Вопрос 4

Каково стандартное отклонение потерь в процессе обучения по всем эпохам?

Не подходит, у меня

Стандартное отклонение потерь в процессе обучения по всем эпохам:
0.061030830225842686

- 0.11 (0.02 – мой ответ)
- 0.66

- 0.99
- 0.33

```

) # Получаем значения потерь обучения для всех эпох
  train_loss = history.history['loss']

  # Вычисляем стандартное отклонение потерь обучения
  std_train_loss = np.std(train_loss)

  print("Стандартное отклонение потерь в процессе обучения по всем эпохам:", std_train_loss)
  # Оцениваем модель
  loss, accuracy = model.evaluate(test_generator)
  print(f'Потери на тесте: {loss}, Точность на тесте: {accuracy}')

; Стандартное отклонение потерь в процессе обучения по всем эпохам: 0.027303713955596436
80/80 [=====] - 20s 241ms/step - loss: 0.5816 - accuracy: 0.7641
Потери на тесте: 0.581597089767456, Точность на тесте: 0.7641154527664185

```

•

Аугментация данных

Для следующего этапа вам потребуется генерировать больше данных с помощью аугментаций.

Добавьте следующие аугментации к генератору обучающих данных:

```

rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest'

```

Вопрос 5

Обучите модель еще на 10 эпох с использованием указанного выше кода. Не создавайте модель с нуля; продолжите обучение существующей.

Каково среднее значение потерь на тестовом наборе данных по всем эпохам после аугментации?

- 0.15
- 0.77
- 0.37 (0,49)
- 0.97

```
#
# Продолжаем обучение модели еще на 10 эпох
history_additional = model.fit(
    train_generator,
    epochs=10,
    validation_data=test_generator
)

Epoch 1/10
20/20 [=====] - 33s 2s/step - loss: 0.6038 - accuracy: 0.6701 - val_loss: 0.5900 - val_accuracy: 0.6970
Epoch 2/10
20/20 [=====] - 30s 2s/step - loss: 0.5933 - accuracy: 0.7234 - val_loss: 0.5449 - val_accuracy: 0.7760
Epoch 3/10
20/20 [=====] - 33s 2s/step - loss: 0.5668 - accuracy: 0.7437 - val_loss: 0.5173 - val_accuracy: 0.7892
Epoch 4/10
20/20 [=====] - 30s 2s/step - loss: 0.5550 - accuracy: 0.7386 - val_loss: 0.5001 - val_accuracy: 0.7961
Epoch 5/10
20/20 [=====] - 30s 2s/step - loss: 0.5691 - accuracy: 0.7259 - val_loss: 0.4868 - val_accuracy: 0.8080
Epoch 6/10
20/20 [=====] - 33s 2s/step - loss: 0.5234 - accuracy: 0.7487 - val_loss: 0.4722 - val_accuracy: 0.7974
Epoch 7/10
20/20 [=====] - 33s 2s/step - loss: 0.5380 - accuracy: 0.7360 - val_loss: 0.5229 - val_accuracy: 0.7447
Epoch 8/10
20/20 [=====] - 33s 2s/step - loss: 0.5452 - accuracy: 0.7335 - val_loss: 0.4674 - val_accuracy: 0.8143
Epoch 9/10
20/20 [=====] - 33s 2s/step - loss: 0.5110 - accuracy: 0.7589 - val_loss: 0.4447 - val_accuracy: 0.8043
Epoch 10/10
20/20 [=====] - 32s 2s/step - loss: 0.5354 - accuracy: 0.7335 - val_loss: 0.4447 - val_accuracy: 0.8294
```

```
# Получаем потери на тестовом наборе данных после аугментации
test_losses = history_additional.history['val_loss']

# Вычисляем среднее значение потерь
mean_test_loss = sum(test_losses) / len(test_losses)

print("Среднее значение потерь на тестовом наборе данных после аугментации:", mean_test_loss)
```

- Среднее значение потерь на тестовом наборе данных после аугментации: 0.4991046726703644

Вопрос 6

Каково среднее значение точности на тестовом наборе данных за последние 5 эпох (с 6 по 10) после аугментации?

- 0.84(0,79 – мой ответ)
- 0.54
- 0.44
- 0.24

```
# Получаем точность на тестовом наборе данных после аугментации
test accuracies = history_additional.history['val_accuracy']

# Выбираем точности за последние 5 эпох (индексы с 5 по 9 включительно)
last_5_epoch_test accuracies = test accuracies[5:]

# Вычисляем среднее значение точности за последние 5 эпох
mean_test_accuracy_last_5_epochs = sum(last_5_epoch_test accuracies) / len(last_5_epoch_test accuracies)

print("Среднее значение точности на тестовом наборе данных за последние 5 эпох после аугментации:", mean_test_accuracy_last_5_epochs)
```

Среднее значение точности на тестовом наборе данных за последние 5 эпох после аугментации: 0.7979924798011779

