Часть 4. Обучение нейронных сетей

Датасет: https://www.kaggle.com/competitions/digit-recognizer/data

Для представленного датасета: необходимо:

1. Решить задачу многоклассовой классификации с помощью многослойной нейронной сети, состоящей из полносвязных слоев.

2. Вычислить градиент целевой функции по обучаемым параметрам нейронной сети, сравнить полученные значения со значениями, вычисленными с помощью библиотечных функций

3. Определить параметры нейронной сети, при которых точность классификации максимальна

Решение

Подготовка данных

```
import zipfile
import os

# File path
file_path = '/mnt/data/digit-recognizer.zip'
extract_folder = '/mnt/data/digit-recognizer'

# Extracting the zip file
with zipfile.ZipFile(file_path, 'r') as zip_ref:
    zip_ref.extractall(extract_folder)

# Listing extracted files
extracted_files = os.listdir(extract_folder)
extracted_files
```

Результат

```
['sample_submission.csv', 'test.csv', 'train.csv']
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

# Загрузка данныех для обучения
train_data_path = os.path.join(extract_folder, 'train.csv')
train_data = pd.read_csv(train_data_path)

# Разделиние данныех на элементы и метки
X = train_data.drop('label', axis=1).values
y = train_data['label'].values

# Нормализация характеристики
scaler = StandardScaler()
X_normalized = scaler.fit_transform(X)
```

```
# Однократное кодирование меток
y_categorical = to_categorical(y)

# Разделите данные на обучающий и проверочный наборы
X_train, X_val, y_train, y_val = train_test_split(X_normalized,
y_categorical, test_size=0.2, random_state=42)

# Построение нейронной сети
model = Sequential([
    Dense(128, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')  # 10 classes for digits 0-9
])

# Компилирование модели
model.compile(optimizer=Adam(), loss='categorical_crossentropy',
metrics=['accuracy'])

model.summary()
```