pip install tensorflow numpy matplotlib

import numpy as np

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.datasets import cifar10

import matplotlib.pyplot as plt

# Загрузка данных

(x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()

x\_train, x\_test = x\_train / 255.0, x\_test / 255.0 # Нормализация данных

# Создание модели

model = models.Sequential([

layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.MaxPooling2D((2, 2)),

layers.Conv2D(64, (3, 3), activation='relu'),

layers.Flatten(),

layers.Dense(64, activation='relu'),

layers.Dense(10)

])

# Компиляция модели

model.compile(optimizer='adam',

loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),

metrics=['accuracy'])

# Обучение модели

history = model.fit(x\_train, y\_train, epochs=10, validation\_data=(x\_test, y\_test))

# Оценка модели

test\_loss, test\_acc = model.evaluate(x\_test, y\_test, verbose=2)

print('\\nTest accuracy:', test\_acc)

# Визуализация потерь и точности

def plot\_history(history):

acc = history.history['accuracy']

val\_acc = history.history['val\_accuracy']

loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs = range(1, len(acc) + 1)

plt.figure(figsize=(12, 6))

# Плотим точность

plt.subplot(1, 2, 1)

plt.plot(epochs, acc, 'b', label='Точность на обуч. наборе')

plt.plot(epochs, val\_acc, 'r', label='Точность на вал. наборе')

plt.title('Точность модели')

plt.xlabel('Эпохи')

plt.ylabel('Точность')

plt.legend()

# Плотим потери

plt.subplot(1, 2, 2)

plt.plot(epochs, loss, 'b', label='Потери на обуч. наборе')

plt.plot(epochs, val\_loss, 'r', label='Потери на вал. наборе')

plt.title('Потери модели')

plt.xlabel('Эпохи')

plt.ylabel('Потери')

plt.legend()

plt.tight\_layout()

plt.show()

plot\_history(history)

**Описание кода:**

1. **Загрузка данных**: Код загружает набор данных CIFAR-10, который содержит 60,000 32x32 цветных изображений в 10 различных классах.
2. **Создание модели**: Модель состоит из нескольких свёрточных и подвыборочных слоёв, панелей (Flatten) и полносвязного слоя.
3. **Компиляция модели**: Выбираем оптимизатор и функцию потерь.
4. **Обучение модели**: Модель обучается на тренировочных данных и проверяется на валидационных.
5. **Оценка модели**: После обучения мы проверяем точность модели на тестовом наборе данных.
6. **Визуализация**: Отрисовываем графики потерь и точности.