НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Системи штучного інтелекту

MNIST: розпізнавання рукописних цифр

Практична робота #4

Виконав:
Ростислав Коваль
Група:
ІО-11мн
Курс:
1

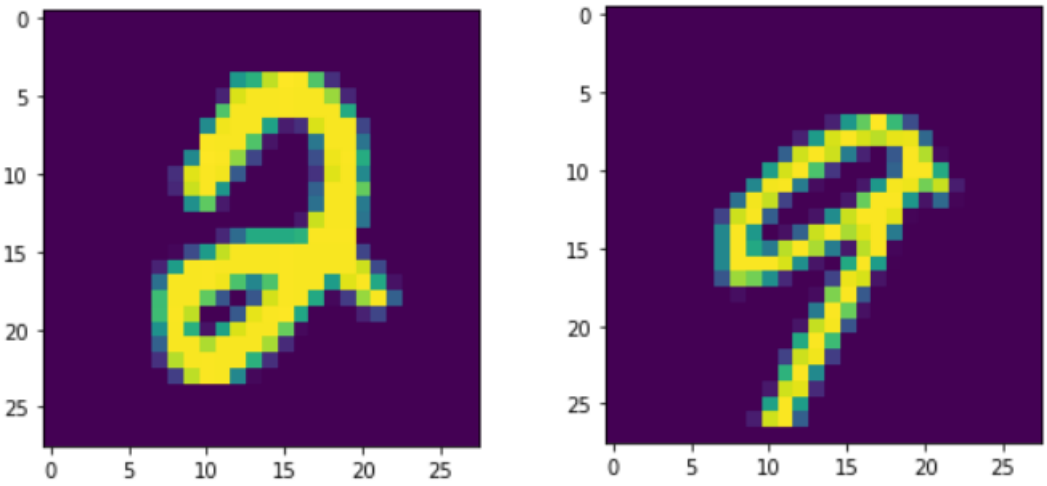
21 листопада 2021 р.

1 MNIST: розпізнавання рукописних цифр

1.1 Набір даних

Для виконання цього завдання було використано набір даних MNIST in CSV [1].

Набір даних: для навчання було використано 58471 рядків з CSV файлу даних, для валідації 1000 рядків. Остаточне тестування проходило на валідаційному наборі даних теж з 1000 значень. Роздільна здатність зображень - (28, 28, 1). Дані не ділив на 255, залишив їх в діапазоні від 0 до 255. Була виконана попередня обробка даних: змінив розмір файлів, перетворив їх з (786,) - на розмір (28, 28, 1). Нормалізація і збільшення даних не використовувалося. Приклади зображень:



1.2 Структура моделі

Моя модель має 3 шари, перший для перетворення даних в одновимірний масив, 2 наступні - повнозв'язні шари, в першому повнозв'язному 128 нейронів та функція активації relu та в останньому 10 нейронів та активація softmax. 10 нейронів відображають кількість наших класів. Для функції витрат була використана SparseCategoricalCrossentropy так як лейбли не були у форматі one-hot encoding. Оптимізатор RMSprop з дефолтними параметрами. Модель була натренована протягом 100 епох з batch-size = 64.

Лістинг 1: Модель

```
# Creating model architecture, setting parameters: loss function, metrics and optimizer.
# Start training for 100 epochs.
mdl = tf.keras.Sequential([tf.keras.layers.Flatten(input_shape=(pix, pix, 1)), tf.keras.
                                                    layers.Dense(128, activation='relu'),tf.keras.
                                                    layers.Dense(10, activation="softmax")])
mdl.compile(optimizer=tf.keras.optimizers.RMSprop(), loss=tf.keras.losses.
                                                    SparseCategoricalCrossentropy(), metrics=['
                                                    accuracy'])
mdl.fit(x_train, y_train, epochs=100, batch_size=64)
```

1.3 Результати експериментів

У даній роботі я не змінював гіперпараметри, використовувалася параметри за замовчуванням. Наприклад, learning rate = 0.001. Основна метрика у моїй роботі - це точність (accuracy). Вона визначається як відношення кількості зображень, які правильно визначила модель до всієї кількості фотографій.

1.4 Допомога

Не отримував допомоги від інших людей, використовував наступні матеріали: [2, 3, 4, 5, 6].

1.5 Висновки

Дана архітектура була обрана, бо це найпростіший варіант для вирішення задачі класифікації на датасеті MNIST. Я не змінював гіперпараметри, вони використовували значення за замовчуванням. Також я зміг отримати більшу точність при збільшенні кількості епох з 50 до 100, якщо обирати кількість більшу за 100 - точність залишається майже тою ж самою. В результаті точність визначення цифр на kaggle становить 0.96936.

Література

[1] Mnist handwritten digit recognition. Kaggle. [Online]. Available: https://www.kaggle.com/c/dmnist-2021/data

[2] T. Phan. Cnn keras 98 accuracy. Kaggle. [Online]. Available: https://www.kaggle.com/phanttan/cnn-keras-98-accuracy

[3] Pascal. Starter: Mnist in csv 5b700bc9-3. Kaggle. [Online]. Available: https://www.kaggle.com/pyim59/starter-mnist-in-csv-5b700bc9-3

[4] P. Varshney. Mnist classification: Eda, pca, cnn 99.7 score. Kaggle. [Online]. Available: https://www.kaggle.com/blurredmachine/mnist-classification-eda-pca-cnn-99-7-score

[5] T. Kumar. Digit recognition-tutorial (cnn) 99.67 accuracy. Kaggle. [Online]. Available: https://www.kaggle.com/tarunkr/digit-recognition-tutorial-cnn-99-67-accuracy

[6] TensorFlow. Tensorflow 2 quickstart for beginners. tensorflow.org. [Online]. Available: https://www.tensorflow.org/tutorials/quickstart/beginner