

Praktikum 1

PIE 1412 Kecerdasan Buatan pada TI

Pengenalan Google Colab

Tujuan Praktikum

1. Memahami Google Colab sebagai tools untuk pengolahan AI dan Machine Learning berbasis cloud.
2. Menyimpan kode Python dalam file .py serta mengupload ke GitHub.
3. Menggunakan dataset sederhana (Iris Dataset) dengan model berbasis CPU.
4. Menerapkan model yang lebih kompleks (CNN) dengan GPU untuk klasifikasi gambar.

Deliverables

1. GitHub Repository (<https://classroom.github.com/a/J8rdHZuP>) berisi:
 - a. File iris_knn.py
 - b. File mnist_cnn.py
 - c. File visualisasi dari evaluasi model iris_knn
 - d. File visualisasi dari evaluasi model mnist_cnn
 - e. Python Notebook File
2. Laporan praktikum:
 - a. Menuliskan yang Anda ketahui tentang Google Colab.
 - b. Menjelaskan berbagai jenis *hardware accelerator* pada *runtime* Google Colab: CPU, TPU, dan GPU.
 - c. Klasifikasi Iris Dataset dengan KNN
 - i. Metode dan Langkah Kerja.
 - ii. Hasil dan Pembahasan.
 - d. Klasifikasi MNIST Dataset dengan CNN
 - i. Metode dan Langkah Kerja.
 - ii. Hasil dan Pembahasan
 1. Perbandingan antara CPU dan GPU.
 - e. Kesimpulan.

Google Colab

Google Colab adalah layanan berbasis cloud dari Google yang dapat menjalankan kode Python tanpa perlu menginstal library maupun perangkat lunak tambahan. Google Colab merupakan layanan tanpa biaya dan mendukung GPU/TPU untuk proses yang membutuhkan komputasi dengan kompleksitas tinggi.

Untuk mulai menggunakan Google Colab, silakan akses melalui browser:

<https://colab.research.google.com/>. Buat Notebook baru dengan nama: Lab1_NPM.ipynb. Secara default, file Python Notebook ini akan tersimpan di sebuah folder pada Google Drive masing-masing akun.

1. Setup GitHub

Membuat token GitHub

Dari GitHub.com, pilih menu Settings → Developer Settings → Personal access tokens, kemudian pilih Tokens (classic) dan Generate new token (classic). Pada form yang muncul tuliskan Note sesuai keperluan, pilih No expiration, dan centang `repo` pada bagian `scopes`.

Setelah muncul, copy token dan pastikan menyimpannya di tempat yang aman.

Setup email, username, dan link GitHub

Ketikkan perintah berikut untuk menghubungkan Google Colab dengan akun GitHub:

```
from google.colab import auth
auth.authenticate_user()

!git config --global user.email "your_email@example.com"
!git config --global user.name "Your Name"

!git clone https://[TOKEN]@github.com/[USERNAME]/[REPOSITORY].git
%cd [REPOSITORY] # Masuk ke dalam folder repo
```

Untuk mengeksekusi code yang telah diketikkan, tekan tombol play di samping code.

2. Kasus 1 – Klasifikasi Iris Dataset dengan KNN (K-Nearest Neighbours)

Dataset: Iris dataset – klasifikasi jenis bunga berdasarkan fitur fisiknya

Algoritma: K-Nearest Neighbours, yang bekerja dengan melihat jarak terdekat antar data

Komputasi: Ringan, bisa dijalankan di CPU

Menyimpan file iris_knn.py

Di Google Colab, kita bisa menyimpan kode ke dalam file .py dengan menuliskan *command*

`%%writefile iris_knn.py`:

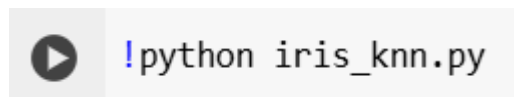
```

1 %%writefile iris_knn.py
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.datasets import load_iris
6 from sklearn.model_selection import train_test_split
7 from sklearn.preprocessing import StandardScaler
8 from sklearn.neighbors import KNeighborsClassifier
9 from sklearn.metrics import accuracy_score, confusion_matrix
10
11 # Load dataset
12 iris = load_iris()
13 X, y = iris.data, iris.target
14
15 # Split dataset
16 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
17
18 # Normalisasi fitur
19 scaler = StandardScaler()
20 X_train = scaler.fit_transform(X_train)
21 X_test = scaler.transform(X_test)
22
23 # Buat model KNN dan latih
24 model = KNeighborsClassifier(n_neighbors=5)
25 model.fit(X_train, y_train)
26
27 # Prediksi
28 y_pred = model.predict(X_test)
29
30 # Evaluasi model
31 accuracy = accuracy_score(y_test, y_pred)
32 print(f"Accuracy: {accuracy:.2f}")
33
34 # Visualisasi Confusion Matrix
35 cm = confusion_matrix(y_test, y_pred)
36
37 plt.figure(figsize=(6,4))
38 sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=iris.target_names, yticklabels=iris.target_names)
39 plt.xlabel("Predicted")
40 plt.ylabel("Actual")
41 plt.title("Confusion Matrix for KNN Model")
42
43 # Simpan gambar
44 plt.savefig("knn_confusion_matrix.png")
45 plt.show()

```

Menjalankan file iris_knn.py

Untuk menjalankan program Python, gunakan code berikut:



Commit dan Push ke GitHub Repository

```

!git add .
!git commit -m "Upload iris_knn.py"
!git push origin main

```

3. Kasus 2 – Klasifikasi Gambar dengan CNN (Convolutional Neural Network)

Dataset: MNIST (angka tulisan tangan 0 - 9)

Algoritma: Convolutional Neural Network

Komputasi: Tinggi, membutuhkan GPU untuk hasil lebih cepat

Menyimpan file mnist_cnn.py

```
1 #!writefile mnist_cnn.py
2 import tensorflow as tf
3 from tensorflow import keras
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 # Load dataset MNIST
8 mnist = keras.datasets.mnist
9 (X_train, y_train), (X_test, y_test) = mnist.load_data()
10
11 # Normalisasi data
12 X_train, X_test = X_train / 255.0, X_test / 255.0
13
14 # Buat model CNN
15 model = keras.Sequential([
16     keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)),
17     keras.layers.MaxPooling2D((2,2)),
18     keras.layers.Flatten(),
19     keras.layers.Dense(128, activation='relu'),
20     keras.layers.Dense(10, activation='softmax')
21 ])
22
23 # Compile model
24 model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
25               metrics=['accuracy'])
26
27 # Training model & simpan history
28 history = model.fit(X_train.reshape(-1,28,28,1), y_train, epochs=5,
29                    validation_data=(X_test.reshape(-1,28,28,1), y_test))
30
31 # Evaluasi model
32 test_loss, test_acc = model.evaluate(X_test.reshape(-1,28,28,1), y_test)
33 print(f"Test accuracy: {test_acc:.2f}")
34
35 # Simpan hasil evaluasi ke file teks
36 with open("evaluation.txt", "w") as f:
37     f.write(f"Test Loss: {test_loss:.4f}\n")
38     f.write(f"Test Accuracy: {test_acc:.4f}\n")
39
40 print("Evaluation results saved to evaluation.txt.")
41
42 # Visualisasi loss & Akurasi
43 history_dict = history.history
44 fig, ax = plt.subplots(1, 2, figsize=(12, 5))
45
46 # Plot Loss
47 ax[0].plot(history_dict['loss'], label='Train loss', marker='o')
48 ax[0].plot(history_dict['val_loss'], label='Validation loss', marker='o')
49 ax[0].set_title("Model loss")
50 ax[0].set_xlabel("Epoch")
51 ax[0].set_ylabel("Loss")
52 ax[0].legend()
53
54 # Plot Accuracy
55 ax[1].plot(history_dict['accuracy'], label='Train Accuracy', marker='o')
56 ax[1].plot(history_dict['val_accuracy'], label='Validation Accuracy', marker='o')
57 ax[1].set_title("Model Accuracy")
58 ax[1].set_xlabel("Epoch")
59 ax[1].set_ylabel("Accuracy")
60 ax[1].legend()
61
62 # Simpan grafik ke dalam file
63 plt.savefig("training_performance.png")
64 plt.show()
65
66 print("Training performance plot saved to training_performance.png.")
```

Menjalankan file mnist_cnn.py

Jalankan mnist_cnn.py dengan mode CPU, catat berapa lama waktu yang dibutuhkan hingga proses klasifikasi selesai.

Lalu jalankan mnist_cnn.py dengan mengaktifkan GPU di Google Colab dengan memilih menu **Runtime > Change runtime type > GPU**. Catat waktu yang dibutuhkan hingga proses klasifikasi selesai.

Bandingkan kedua hasil di atas, dan berikan penjelasan mengapa hal tersebut terjadi.

Commit dan Push ke GitHub

Selanjutnya commit dan push file mnist_cnn.py ke GitHub repository.

4. Menyimpan file .ipynb ke GitHub

Jangan lupa untuk melakukan save Python Notebook file secara berkala.

Untuk mengupload Python Notebook file ke GitHub, cek Lokasi Python Notebook file. Jika tersimpan di Google Drive, perlu terlebih dulu menghubungkan Google Colab dan Google Drive dengan command berikut:

```
from google.colab import drive
drive.mount('/content/drive')
```

Misalnya file ada di My Drive/Colab Notebooks/Lab1_NPM.ipynb, copy file ke repository lokal dengan perintah berikut:

```
!cp "/content/drive/My Drive/Colab Notebooks/Lab1_NPM.ipynb" "/content/[REPOSITORY]/"
```

Lalu commit dan push ke GitHub repository.

Jika ada perubahan atau penambahan aktivitas pada Python Notebook file, lakukan proses copy dan commit-push ke GitHub repository dengan cara di atas.