

Praktikum Sistem Cerdas



NRP	: 3223600019
Nama	: Muhammad Bimo Fachrizky
Materi	: Membuat Aplikasi Deep Learning tensorflow
Tanggal	: Senin, 28 April 2025

Praktikum 9

Membuat Aplikasi Deep Learning tensorflow

I. Tujuan Pembelajaran

- Mahasiswa dapat memahami dan menjelaskan konsep Deep Learning
- Mahasiswa dapat menjelaskan model Deep Learning
- Mahasiswa dapat membuat aplikasi Deep Learning tensorflow

Software yang di perlukan

- Microsoft Visual C++
- PyCharm

II. Langkah percobaan

1. Simple Regression

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras

def plot_data(x_data, y_data):
    fig, ax = plt.subplots()
    ax.plot(x_data, y_data, 'ro')
    plt.pause(1)
    return None

if __name__ == "__main__":
    # Define and compile the neural networks
    model = tf.keras.Sequential([
        keras.layers.Dense(units=1, input_shape=[1])
    ])
    model.compile(optimizer="sgd", loss="mean_squared_error")

    # Providing the data
    xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0])
    ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0]) # y = 2x - 1

    # Plotting data will help us to understand faster
    # visually about the data
    plot_data(xs, ys)

    # Train the neural network
    model.fit(xs, ys, epochs=500)
```

```
# Predict using the model
print(model.predict(np.array([10.0])))
```

2. Fashion_Classifier

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import tensorflow as tf

if __name__ == "__main__":
    # Load the dataset
    mnist = tf.keras.datasets.fashion_mnist
    (training_images, training_labels), (test_images, test_labels) =
mnist.load_data()

    # Plot an image from the dataset
    plt.imshow(training_images[0], cmap="Greys")
    plt.pause(1) # Increase the pause time to give time for plotting
    print(training_labels[0])
    print(training_images[0])

    # Normalizing image intensities to [0, 1]
    training_images = training_images / 255.0
    test_images = test_images / 255.0

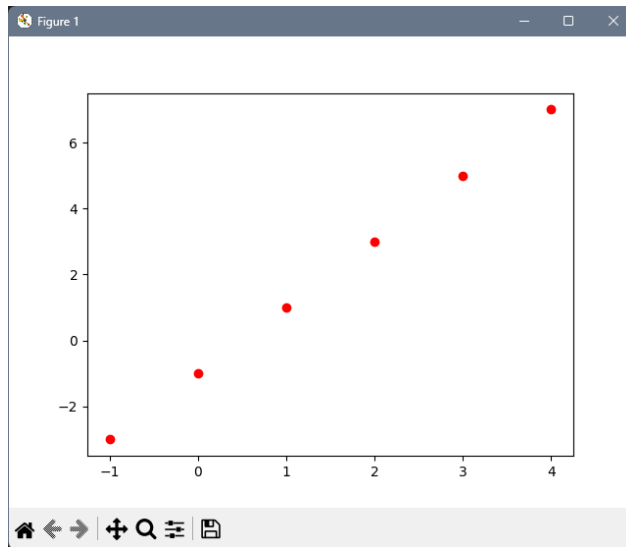
    # Create a model
    model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128, activation=tf.nn.relu),
        tf.keras.layers.Dense(10, activation=tf.nn.softmax)
    ])

    # Compile the model and train it to the dataset
    model.compile(
        optimizer=tf.optimizers.Adam(),
        loss="sparse_categorical_crossentropy",
        metrics=["accuracy"]
    )
    model.fit(training_images, training_labels, epochs=5)

    model.evaluate(test_images, test_labels)
```

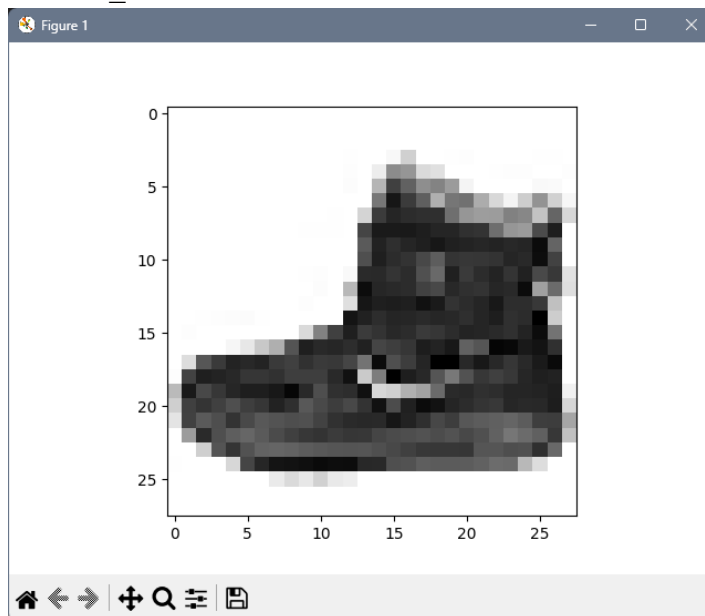
III. Hasil Percobaan

1. Simple_Regression



```
Epoch 497/500  
1/1 ————— 0s 137ms/step - loss: 4.7738e-05  
Epoch 498/500  
1/1 ————— 0s 195ms/step - loss: 4.6757e-05  
Epoch 499/500  
1/1 ————— 0s 167ms/step - loss: 4.5798e-05  
Epoch 500/500  
1/1 ————— 0s 129ms/step - loss: 4.4856e-05  
1/1 ————— 0s 123ms/step  
[[18.98046]]
```

2. Fashion_Classifier



dibangun memiliki tiga lapisan: Flatten() untuk mengubah input 2D menjadi 1D, Dense(128, activation=tf.nn.relu) sebagai hidden layer untuk mendeteksi fitur kompleks, dan Dense(10, activation=tf.nn.softmax) sebagai output layer dengan 10 neuron yang mewakili masing-masing kelas pakaian. Model dilatih (fit) selama 5 epoch pada data pelatihan, dan setelah pelatihan selesai, model dievaluasi (evaluate) terhadap data pengujian untuk menghitung akurasi dan loss pada data yang belum pernah dilihat sebelumnya.

V. Kesimpulan

TensorFlow dan Keras merupakan tools powerful dalam pengembangan model pembelajaran mesin, baik untuk kasus regresi sederhana maupun klasifikasi gambar kompleks. Program pertama menunjukkan bahwa model neural network bahkan dengan arsitektur paling dasar sekalipun (satu neuron) mampu mempelajari hubungan linear antara input dan output, dengan tingkat akurasi yang tinggi setelah melalui proses pelatihan berulang (epoch). Ini membuktikan kemampuan jaringan saraf dalam menyelesaikan masalah regresi secara efisien. Sementara itu, program kedua memperlihatkan proses pembuatan sistem klasifikasi gambar dengan dataset Fashion MNIST, yang mencakup langkah-langkah penting seperti normalisasi data, pembuatan arsitektur jaringan dengan hidden layer, serta evaluasi performa model terhadap data uji. Dengan menerapkan prinsip-prinsip deep learning yang tepat, model mampu mencapai tingkat akurasi tinggi dalam mengenali objek visual dari gambar 2D. Kedua program juga mempertegas pentingnya pemahaman preprocessing data dan pemilihan fungsi aktivasi, loss, dan optimizer yang sesuai dengan jenis permasalahan.