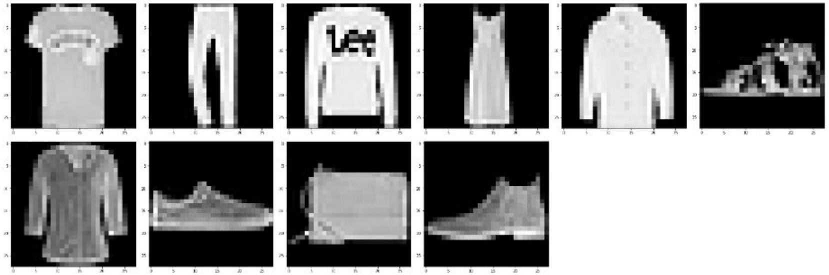
COEN 140L Lab 7 Intro

Xuyang Wu

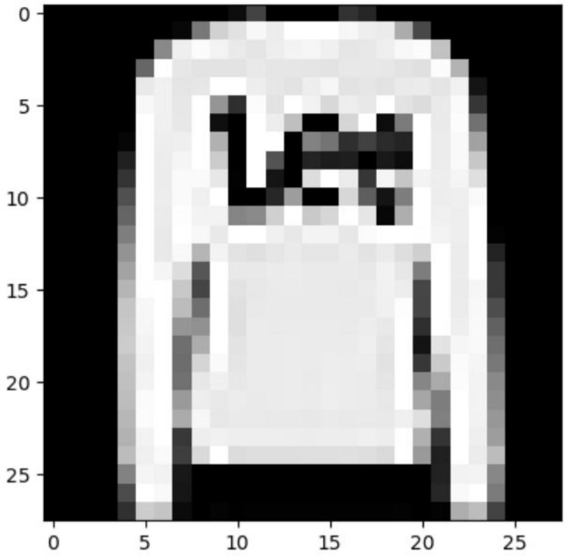
Dataset



```
import tensorflow as tf
from tensorflow import keras

# import the data
mnist = keras.datasets.fashion_mnist

# load the data, train = (60000,28,28), test = (10000,28,28)
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

# normalize the pixel values to be real numbers in [0,1], It’s fine if you don’t normalize them
x_train, x_test = x_train /255.0, x_test / 255.0
```

Plot grayscale image

```
import matplotlib.pyplot as plt

plt.imshow(x_testo[0, :, :], cmap='gray', vmin=0, vmax=255)
plt.show()
```



Explanation of example code

Based on the example code(diabetes_nn.py), you need your own modification version:

- **Add layers for a neural network, and set the activation function such as Relu, or softmax.**
  - **two-layer neural network**, hidden layer has 512 nodes, and adopts the ReLU activation function; the output layer has 10 nodes, and adopts the softmax activation function.
  - **Flatten layer**, as the image has 2 dimensions.
- **Set the loss function 'sparse_categorical_crossentropy'**
  - https://keras.io/api/losses/probabilistic_losses/#sparse_categorical_crossentropy-function
- **Calculate confusion matrix**

  from sklearn.metrics import confusion_matrix
  CM = confusion_matrix(y_test, y_test_hat, labels=range(10))
  CM_array = CM
- **calculate accuracy rate for test data**
  - You could used the solution such as last week, with y_test_hat = model.predict_classes(x_test).
  - You also could try loss_test, acc_test = model.evaluate(x_test, y_test)

Calculate the number of parameters and number of multiplications

- Generate **one neuron** in the hidden layer, we need a weighted sum of **all neurons in the previous (input) layer**, **plus a bias term**.
- Using **model.summary()** in the code to get the number of model parameters to **check your calculation**.
- Correctly calculating the number of parameters and multiplications is very important.

Include in the report

1. Display **10 selected images** from the test set, as gray-scale images, each with a different class label.
2. Give the recognition **accuracy rate** of the whole test set, and **show the confusion matrix**.
3. **Manually calculate** the **number of parameters** in the neural network model. That is, the number of weight elements (**include the bias terms**). Show you work. Verify whether your results are the same as those given by model.summary().
4. Manually calculate the **number of multiplications** required to generate the hidden layer and the output layer. The **weights have bias terms**. You only need to consider the multiplications required to calculate the weighted sums. You don't need to consider the multiplications involved in the softmax function. Show you work.

Lab Tasks

- **Need demo** for week 7 assignment(10% points).
- Submit to Camino a pdf report with answers(60% points), the report contains some **results** which required by lab document, you also need to add some **observations** for the questions.
- Submit all the source code needed to generate these answers to Camino(30% points).