📝 Documentary Report: Deep Learning for Fashion MNIST Classification

---

📘 Introduction

The Fashion MNIST dataset provides a practical platform to explore deep learning in image classification. Unlike traditional datasets like MNIST digits, Fashion MNIST consists of grayscale images of clothing items, making it slightly more complex and realistic. This project aims to train a deep learning model to recognize and classify these images into one of ten fashion categories using a supervised learning approach.

🧠 Dataset Overview

- **Total Samples:** 70,000 images
- **Training Set:** 60,000 images
- **Test Set:** 10,000 images
- **Image Format:** Grayscale, 28×28 pixels
- **Number of Classes:** 10

Each image belongs to one of the following categories:

1. T-shirt/top
2. Trouser
3. Pullover
4. Dress
5. Coat
6. Sandal
7. Shirt
8. Sneaker
9. Bag
10. Ankle boot

🔄 Data Preprocessing

Before training, each image is normalized so that pixel values fall between 0 and 1. This ensures more efficient learning. The images are also reshaped from 2D arrays (28×28) into 1D vectors, allowing them to be passed into a fully connected neural network.

📐 Model Design

The model used in this project is a simple feedforward neural network. It includes:

- An **input layer** that takes the flattened image vector
- A **hidden layer** with an activation function (ReLU)
- A **dropout layer** to reduce overfitting
- An **output layer** with 10 units for the 10 fashion classes

The model uses an appropriate activation function in the output layer to produce class probabilities.

⚙️ Training Process

The model was trained using a common optimizer and a loss function suitable for multi-class classification. Training occurred over **twenty-eight epochs**, where the model gradually improved its predictions by minimizing the error between predicted and actual classes.

📈 Model Performance

Once trained, the model was evaluated using the test dataset. Key performance results include:

- **Training Accuracy:** ~91%
- **Test Accuracy:** ~88–89%
- **Training Loss:** Decreased steadily with each epoch
- **Test Loss:** Slightly higher than training loss, indicating minor overfitting but strong generalization overall

These results confirm that the model is well-trained and performs reliably on unseen data.

▶️ How to Run the Project

To run this project from start to finish:

1. **Set Up the Environment**
   Install essential Python libraries such as Pandas, NumPy, and Matplotlib.
2. **Load the Dataset**
   Use the Fashion MNIST dataset provided through Keras' built-in datasets module.
3. **Preprocess the Data**
   Normalize image pixel values and reshape the image arrays to match model input requirements.
4. **Build the Neural Network Model**
   Define the model structure with input, hidden, and output layers.

5. **Compile and Train the Model**
   Choose the optimizer, loss function, and metrics, then run training over a chosen number of epochs.
6. **Evaluate the Model**
   Test the model on unseen data and analyze accuracy and loss to validate performance.
7. **Make Predictions**
   Use the trained model to predict labels of new images.

This project can be executed within a Jupyter notebook, with results including accuracy metrics and prediction outputs.

📌 Conclusion

This project demonstrates the complete pipeline of using deep learning to classify fashion items. Through data normalization, network design, training, and evaluation, a well-performing model was achieved. With nearly 90% accuracy on test data, this project showcases how neural networks can be effectively used for image classification tasks in modern datasets.