

Fashion MNIST Classification Using Fully Connected Neural Network (FCNN)

Introduction

This project focuses on classifying images from the Fashion MNIST dataset using a Fully Connected Neural Network (FCNN). The dataset consists of 60,000 training images and 10,000 test images, each belonging to one of 10 fashion categories. The goal is to build and train a model that accurately predicts the category of a given fashion item.

Dataset and Preprocessing

- The dataset contains grayscale images of size 28x28 pixels.
- Labels are mapped to 10 different fashion categories, including T-shirts, trousers, dresses, and shoes.
- Normalization: Pixel values are scaled between 0 and 1 to improve learning efficiency.
- One-Hot Encoding: Labels are converted into categorical format for multi-class classification.

Model Architecture

The Fully Connected Neural Network (FCNN) consists of:

1. Flatten Layer: Converts the 28x28 pixel matrix into a 1D array.
2. Dense Layer (256 neurons, ReLU activation): First hidden layer.
3. Dense Layer (128 neurons, ReLU activation): Second hidden layer.
4. Output Layer (10 neurons, Softmax activation): Predicts one of the 10 categories.

Training and Optimization

- Optimizer: Stochastic Gradient Descent (SGD) is used to minimize loss.
- Loss Function: Categorical Cross-Entropy is used for multi-class classification.
- The model is trained for 15 epochs with a batch size of 32.

Evaluation and Results

- The model achieves a final test accuracy displayed after training.
- Plots for training vs. validation loss and training vs. validation accuracy show model performance over epochs.

Prediction on New Images

- A function is implemented to predict and visualize a single image. The model's prediction is compared with the actual label.

Conclusion

This project demonstrates a simple yet effective FCNN for Fashion MNIST classification. The model achieves good accuracy, and further improvements can be made by experimenting with different optimizers, adding dropout layers, or using a deeper network.

