Neural Network Image Classification Using Fashion MNIST

Introduction

This project explores the application of feedforward neural networks for image classification using the Fashion MNIST dataset. The goal is to demonstrate how deep learning models can outperform traditional machine learning approaches in recognizing complex visual patterns. Fashion MNIST contains 70,000 grayscale images of clothing items across 10 categories, making it suitable for evaluating neural network performance on real-world data.

Methodology

Dataset and Preprocessing

The Fashion MNIST dataset was loaded using TensorFlow/Keras. Each image was normalized to a pixel range of [0, 1] to improve training stability. The images were flattened from 28×28 into 784-dimensional vectors to fit the feedforward architecture. Labels were one-hot encoded for multi-class classification.

Model Architecture

A simple feedforward neural network was constructed using Keras' Sequential API. The architecture included:

- Input layer: 784 neurons
- Hidden layers: Two layers with 128 and 64 neurons, ReLU activation
- Output layer: 10 neurons with Softmax activation

The model was compiled with the Adam optimizer, categorical crossentropy loss, and accuracy as the evaluation metric.

Training

The model was trained for 15 epochs with a batch size of 64. A validation split of 20% was used to monitor performance during training. Training and validation accuracy/loss were plotted to assess convergence and detect overfitting.

Results Analysis

The trained feedforward neural network achieved an overall **accuracy of 88%** on the test set, demonstrating strong performance across most fashion categories.

Classification Report Insights

- **Highest performing classes**:
    - *Trouser*: Precision and recall of 0.98, indicating near-perfect classification.
    - *Sandal*, *Bag*, and *Ankle boot*: F1-scores above 0.95, showing excellent model confidence and consistency.

- **Challenging classes**:
- *Shirt*: F1-score of 0.68, with lower recall (0.62), suggesting frequent misclassification—likely due to visual similarity with *T-shirt/top* and *Coat*.
- *Pullover* and *Coat*: Moderate precision and recall, indicating some confusion with adjacent categories.

Confusion Matrix Highlights

- The confusion matrix confirms that *Shirt* is often misclassified as *T-shirt/top*, *Pullover*, or *Coat*.
- *Sneaker* and *Trouser* show minimal misclassification, reinforcing their high precision and recall.
- Most errors occur between visually similar items, which is expected in fashion datasets.

Training Curves

Training and validation accuracy curves showed stable convergence, with no signs of overfitting. Validation accuracy closely tracked training accuracy, indicating good generalization.

Practical Application

In a fashion retail setting, this model could automate product tagging by classifying user-uploaded images. This would streamline inventory management and enhance search functionality. Operational challenges include scaling the model to handle large volumes of data, ensuring real-time inference, and integrating with existing systems. Solutions include deploying the model via TensorFlow Serving, optimizing for latency, and wrapping it in an API for seamless integration.

Conclusion

This project successfully demonstrated the use of a feedforward neural network for image classification on the Fashion MNIST dataset. The model achieved **88% test accuracy**, with strong precision and recall across most categories. Visual analysis confirmed stable training behavior and effective generalization, while the confusion matrix highlighted challenges in distinguishing visually similar items like shirts and pullovers.

These results validate the model's potential for real-world deployment in domains such as fashion retail or digital media tagging. Future improvements could include experimenting with convolutional architectures, applying data augmentation, and optimizing for deployment on scalable platforms. Addressing class confusion through enhanced feature extraction or ensemble methods could further boost performance.