ASSIGNMENT 3
Build Model for FASHION MNIST Data Set

YangDongJae
[1]Korea Tech University
Department of Computer Science Engineering
2021136150
ydj9805@koreatech.ac.kr

Abstract

This study applies convolutional neural networks (CNNs) to classify 70,000 grayscale images from the Fashion MNIST dataset. It emphasizes data normalization, optimal preprocessing, and a PyTorch-based pipeline. The goal is to achieve over 91% validation and 90% test accuracy, using tools like Google Colab and Wandb for training and visualization, and analyzing test predictions to assess model performance.

This study explores the application of deep learning techniques to the Fashion MNIST dataset, a collection of 60,000 28x28 grayscale images of clothing items across 10 categories. We focus on the utilization of convolutional neural networks (CNNs) for classifying these images into their respective categories. The dataset is divided into 55,000 training images and 5,000 test images, ensuring a comprehensive evaluation of the model's performance. Our approach involves data normalization, identifying the optimal mean and standard deviation values for image preprocessing, and implementing a robust data loading pipeline using PyTorch. We also delve into the intricacies of training a CNN on the Fashion MNIST dataset, emphasizing the importance of various hyperparameters such as learning rate, batch size, early stopping criteria, weight decay, normalization type, and dropout rate. The study aims to achieve a validation accuracy of over 91% and a test accuracy of over 90%, adhering to stringent performance metrics. Additionally, we explore the utility of image augmentation techniques and the effective use of tools like Google Colab and Wandb for model training and performance visualization. The assignment also includes the analysis of sample test data predictions to understand the model's strengths and limitations. Overall, this study provides a comprehensive guide for applying deep learning techniques to fashion image classification, showcasing the potential of CNNs in practical applications.

Assignment Requirements

- **Dataset Utilization**: Employ the Fashion MNIST dataset, which includes 60,000 28x28 grayscale images of clothing items, divided into 55,000 training images and 5,000 test images.
- **Deep Learning Technique**: Apply convolutional neural networks (CNNs) for the classification of images into 10 categories.
- **Data Normalization**: Determine and implement the optimal mean and standard deviation values for image preprocessing.
- **Data Loading Pipeline**: Develop a data loading pipeline using PyTorch to handle the dataset efficiently.
- **CNN Training**: Focus on training a CNN, paying special attention to hyperparameters such as learning rate, batch size, early stopping criteria, weight decay, normalization type, and dropout rate.
- **Performance Goals**: Aim to achieve a validation accuracy of over 91% and a test accuracy of over 90%.
- **Image Augmentation Techniques**: Explore and implement image augmentation methods to enhance the model's performance.
- **Use of Tools**: Leverage tools like Google Colab for model training and Wandb for performance visualization and analysis.
- **Sample Test Data Analysis**: Analyze sample test data predictions to evaluate the model's strengths and limitations.
- **Comprehensive Guide Creation**: Assemble a detailed guide for applying deep learning techniques to fashion image classification.

Method

Data Preprocessing

The primary dataset for this study is the Fashion MNIST dataset, consisting of 60,000 training and 10,000 test grayscale images, each of 28x28 pixel size. Each image is categorized into one of 10 fashion classes.

Data augmentation was chosen as a critical preprocessing step to enhance the robustness and generalization of the models. This approach mitigates the risk of overfitting, especially crucial given the relatively simple and repetitive nature of the Fashion MNIST dataset. The augmentation techniques included:

- **Random Rotations:** Images were rotated by up to 15 degrees to simulate variations in orientation.

- **Width and Height Shifts:** Images were randomly shifted horizontally and vertically by up to 20% of their total width or height.
- **Horizontal Flipping:** Images were mirrored horizontally to simulate different perspectives.

These methods collectively introduce sufficient variability in the dataset, mimicking a more realistic scenario where fashion items can be viewed from different angles and orientations.

Approach Methods

Our approach involved the implementation and comparative analysis of four distinct CNN architectures, each with unique features:

1. **Basic CNN (4 Convolutional Layers + 2 Fully Connected Layers + Pooling + Dropout + Batch Normalization):** Referred to as `CNNModel5`, this baseline model included four convolutional layers with increasing filter sizes (64, 64, 128, 256), two max-pooling layers, dropout layers for regularization, and batch normalization for training stability.
2. **Modified GoogleNet (Inception Module + Adam Optimizer + Learning Rate Decay):** Adaptation of the GoogleNet architecture for the Fashion MNIST dataset, employing Inception modules and trained using the Adam optimizer with learning rate decay.
3. **GoogleNet with Stochastic Gradient Descent (SGD) Optimizer:** A variant of GoogleNet, trained using the SGD optimizer to compare its convergence and generalization capabilities against the Adam optimizer.
4. **3 Convolutional Layers + 1 Fully Connected Layer Model with Pooling, L2 Regularization, and Batch Normalization:** Named `CNNModel4`, this compact architecture included three convolutional layers, one fully connected layer, and batch normalization, with L2 regularization to prevent overfitting.

Model Architecture and Training

Each model was implemented in PyTorch and trained using either the Adam or SGD optimizer. Training parameters were set based on preliminary experiments. Models were trained until convergence, with early stopping to prevent overfitting. Performance evaluation was based on classification accuracy on the test set, focusing on the impact of architectural differences and optimization strategies.

Experimental Setup

Experiments were conducted using a standard machine learning setup, with data loaders created for the Fashion MNIST dataset to manage training and validation batches. Training processes were monitored using Weights & Biases (wandb), facilitating real-time tracking of metrics. Models were evaluated using a separate test set to ensure an unbiased assessment of their generalization capabilities.

Analysis of Misclassification

An observed misclassification by the model has been presented for the provided image. The model misclassified an item labeled as 4 (Coat) as 2 (Pullover). Below are the potential reasons for this error:

- The resolution and quality of the image may be low, leading to the model's inability to accurately recognize the texture and pattern of the garment.
- Coats and pullovers share similar shapes and contours which might cause confusion for the model between these two categories.
- The distinction between the two categories may not have been sufficiently learned during training, which could be due to data imbalance or insufficient feature learning.
- Specific design elements of the garment (like sleeve length or neckline) could differ from the patterns commonly encountered by the model, resulting in a misprediction.

These factors could influence the performance of the model and lead to errors. Therefore, to enhance the model's accuracy, improvement in data quality, learning more diverse features, and addressing class imbalance is necessary.

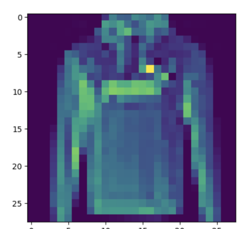


Figure 1: coat

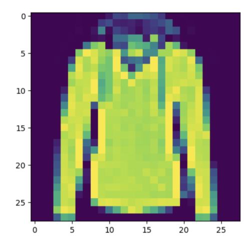
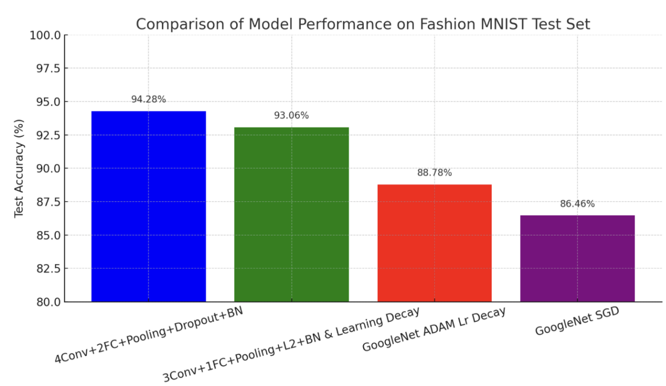


Figure 2: pullover

Results

Model Performance

The performance of four distinct convolutional neural network (CNN) models on the Fashion MNIST test dataset was evaluated. The models varied in complexity and optimization strategies. The test accuracy of each model is as follows:



- **4Conv+2FC+Pooling+Dropout+BN:** Achieved the highest test accuracy of 94.280%. This model's architecture, encompassing four convolutional layers, two fully connected layers, pooling, dropout, and batch normalization, proved to be highly effective for the task.
- **3Conv+1FC+Pooling+L2+BN & Learning Decay:** With a slightly simpler architecture, this model achieved a test accuracy of 93.060%. It suggests that even with fewer layers, the model could capture essential features for classification.
- **GoogleNet ADAM Lr Decay:** Utilizing a modified GoogleNet architecture with a learning rate decay strategy and the Adam optimizer, this model achieved a test accuracy of 88.780%. The result indicates that while the inception modules in GoogleNet are powerful, they might be more complex than necessary for a dataset like Fashion MNIST.
- **GoogleNet SGD:** This variant of the GoogleNet, optimized with Stochastic Gradient Descent (SGD), recorded a test accuracy of 86.460%. The lower performance compared to the Adam-optimized version suggests that the choice of optimizer can significantly impact model performance.

The bar chart clearly illustrates these differences in performance, with the highest accuracy achieved by the 4Conv+2FC model underscoring the effectiveness of a well-balanced architecture.

Discussion

The results demonstrate that more complex architectures like GoogleNet do not always lead to better performance on relatively simpler datasets like Fashion MNIST. Instead, a well-tuned balance between depth (number of layers), regularization techniques (like dropout and batch normalization), and the right choice of optimizer plays a crucial role in achieving high accuracy.

This study highlights the importance of model architecture choices and optimization strategies in the field of image classification, providing insights into the trade-offs between model complexity and performance.

Future Work

Further research could explore the impact of additional regularization techniques, different activation functions, or more sophisticated learning rate schedules. Applying these models to more complex datasets would provide a deeper understanding of their scalability and generalization capabilities.

Home Work#3 Review

In this project, my goal was to explore state-of-the-art (SOTA) models like SAM and GoogleNet, with an emphasis on enhancing their structures for optimal performance. Initially, I believed that modifications to these advanced models would yield the best results. This approach, however, was put to test against a surprising contender: a basic Convolutional Neural Network (CNN).

The Paradox of Complexity

- My experiments with complex models unveiled a significant trade-off. Increased complexity resulted in longer times to minimize loss. This highlighted a crucial aspect of AI development: the balance between computational power and task efficiency.
- In environments with substantial resources, such as FAANG companies, there are still limits to computational capabilities. The effectiveness of a model is not just in its complexity but in how efficiently it uses the available resources.

The Power of Simplicity

- The performance of GoogleNet was initially disappointing. However, switching to a basic CNN model led to astonishing results. The CNN not only learned faster but also delivered the highest performance on the Fashion MNIST task.
- This superior performance was due to the simpler nature of the Fashion MNIST dataset, featuring smaller sizes and fewer channels compared to more complex datasets like CIFAR-10. The basic CNN efficiently captured and represented these features.

Conclusion

Through this project, I learned an invaluable lesson in AI development: the choice of model should be specifically tailored to the task's requirements. A complex model isn't inherently superior; its effectiveness depends on the nature of the data and the objectives of the task. This project has reshaped my approach to AI modeling, emphasizing the importance of strategic and context-driven methodology over mere complexity.