### Fashion MNIST Project Report

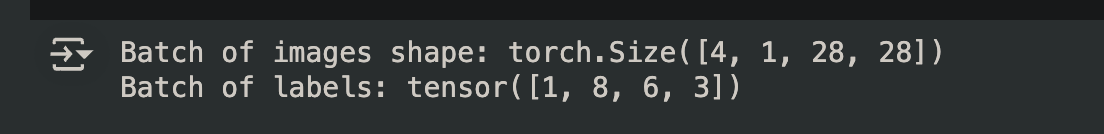
In this project, I implemented a convolutional neural network (CNN) to classify images from the Fashion MNIST dataset. The steps involved in the project are summarized below:

**Library Installation**: I began by installing the necessary libraries, including PyTorch, torchvision, and other dependencies.

**Setup and Configuration**: I set up the device for GPU usage if available, ensuring reproducibility by seeding the random number generators.

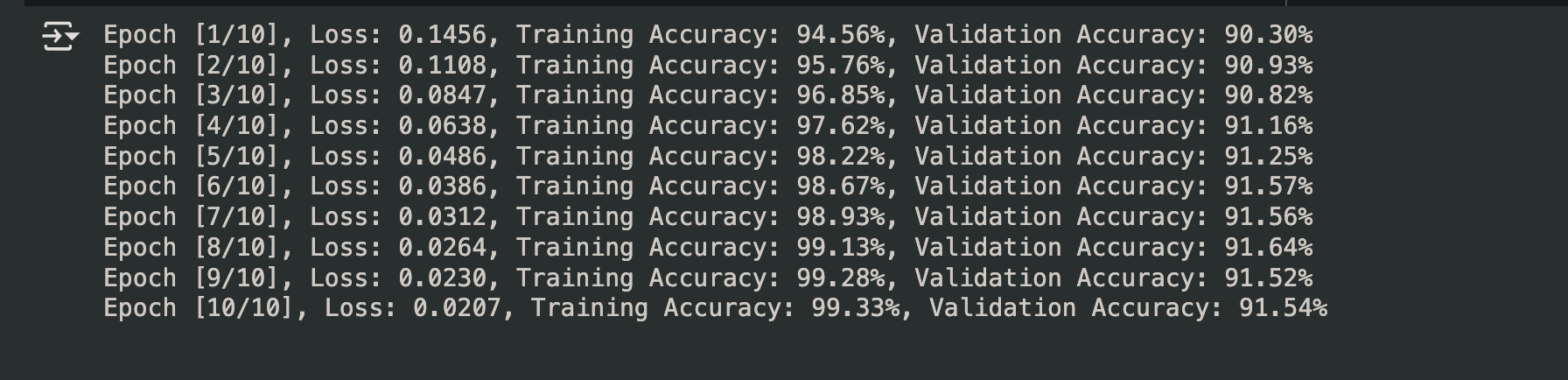
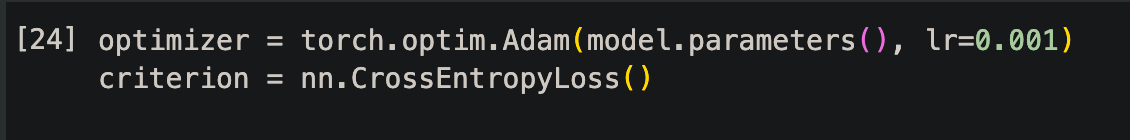
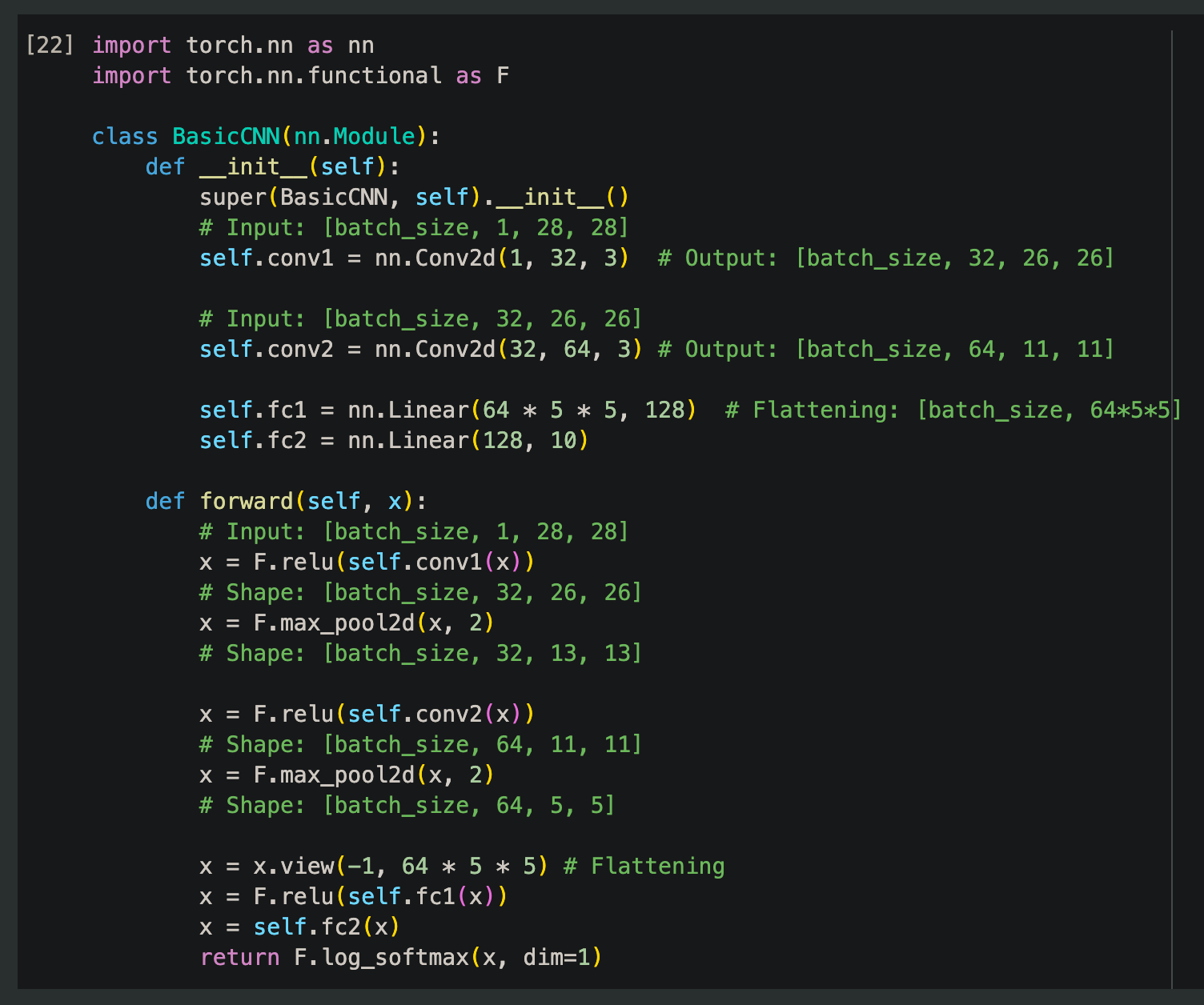
**Data Loading and Processing**:

* 1. I loaded the Fashion MNIST training and test datasets, applying transformations to convert images into PyTorch tensor format.
  2. The training data was organized into batches with a batch size of 4, which produced a shape of (4, 1, 28, 28) for images and a corresponding tensor for labels.



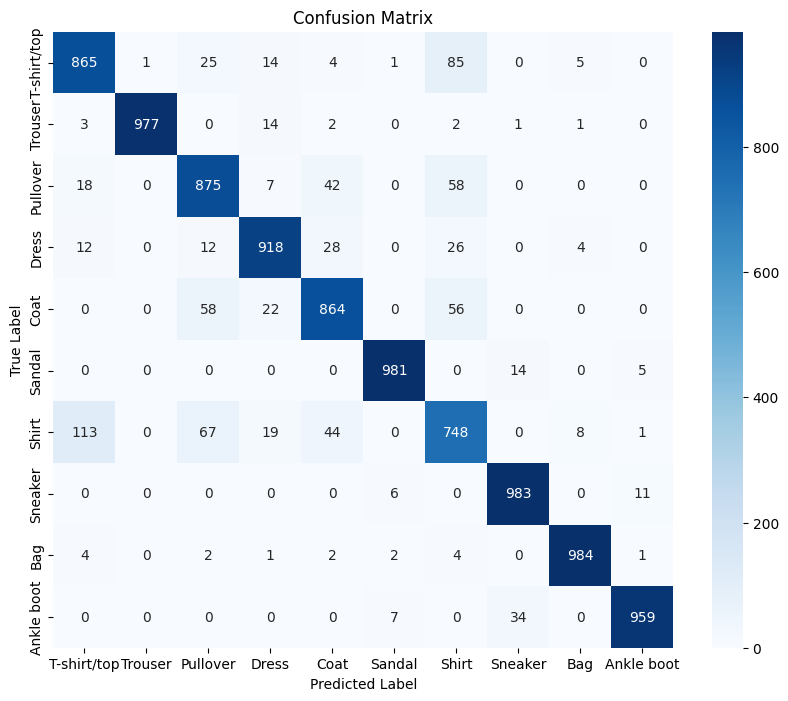
**Model Design**:

* 1. I designed a CNN model consisting of two convolutional layers and two fully connected layers.
  2. The model was trained using the Adam optimizer with a learning rate of 0.001 for 10 epochs.
  3. Initial accuracy results indicated a strong performance, with a training accuracy reaching 99.33% and a validation accuracy of 91.54%.



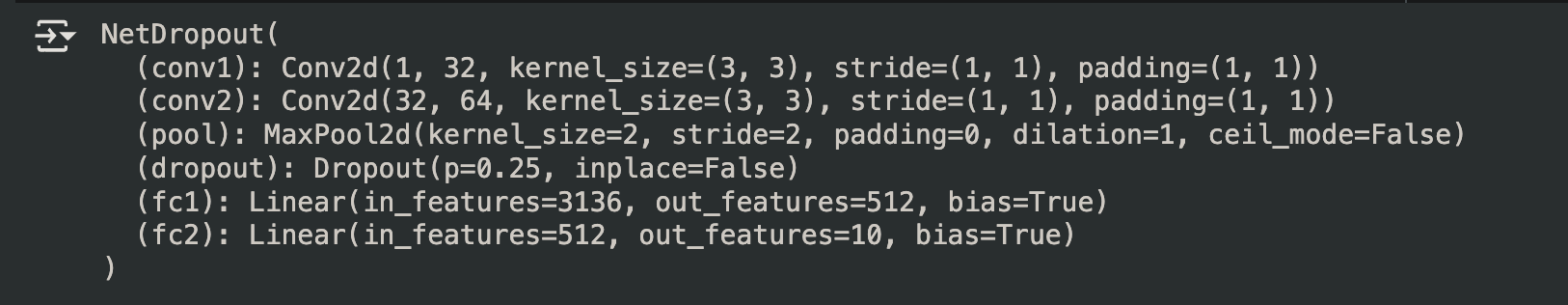
**Evaluation**:

* 1. I implemented a confusion matrix for a comprehensive evaluation of the model's performance across different classes.
  2. The confusion matrix showed precision and recall values that varied among classes, indicating areas for improvement.

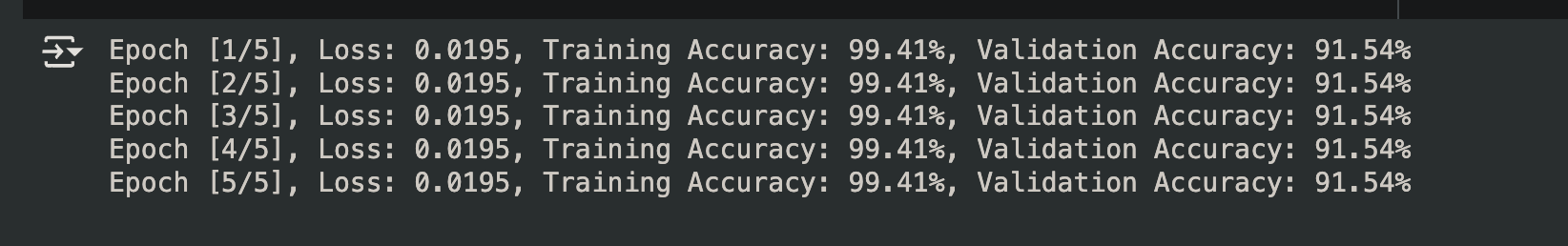


**Regularization**:

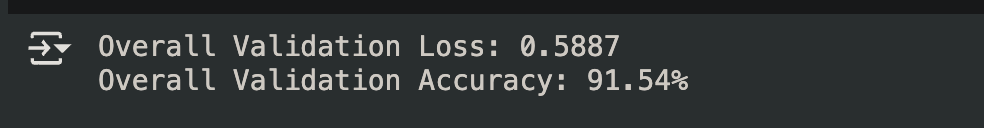
* 1. To further enhance model performance and reduce overfitting, I applied dropout regularization, which helped maintain accuracy during training.



1. Enhanced result after regularization



1. Overall accuracy and Loss



### Comparison of Results

#### First Attempt:

* **Batch Size**: 4
* **Optimizer**: Adam
* **Model**: CNN
* **Layers**: 2
* **Learning Rate**: 0.001
* **Epochs**: 10

**Results**:

* Epoch [1/10]: Loss: 0.1456, Training Accuracy: 94.56%, Validation Accuracy: 90.30%
* Epoch [10/10]: Loss: 0.0207, Training Accuracy: 99.33%, Validation Accuracy: 91.54%
* **Overall Validation Loss**: 0.5887
* **Overall Validation Accuracy**: 91.54%

#### Second Attempt:

* **Batch Size**: 64
* **Optimizer**: SGD
* **Model**: Updated CNN model with additional layer
* **Layers**: 3
* **Learning Rate**: 0.01
* **Epochs**: 10

**Results**: Very poor performance.

* Loss: 2.3151, Accuracy: 8.66%

#### Third Attempt:

* **Batch Size**: 128
* **Optimizer**: Adam
* **Model**: Updated CNN model with additional layer
* **Layers**: 3
* **Learning Rate**: 0.01
* **Epochs**: 5
* **Dataset Type**: Numpy Array

**Results**: Still poor performance.

* Loss: 0.4903, Accuracy: 86.90%
* Subsequent losses and accuracies fluctuated but did not improve significantly.