Homework 4 Report

Erika, Richard, Leo, Ciny

# Softmax Properties

# 

These outputs confirm that adding a constant offset to the input vector does not change the result of the softmax function

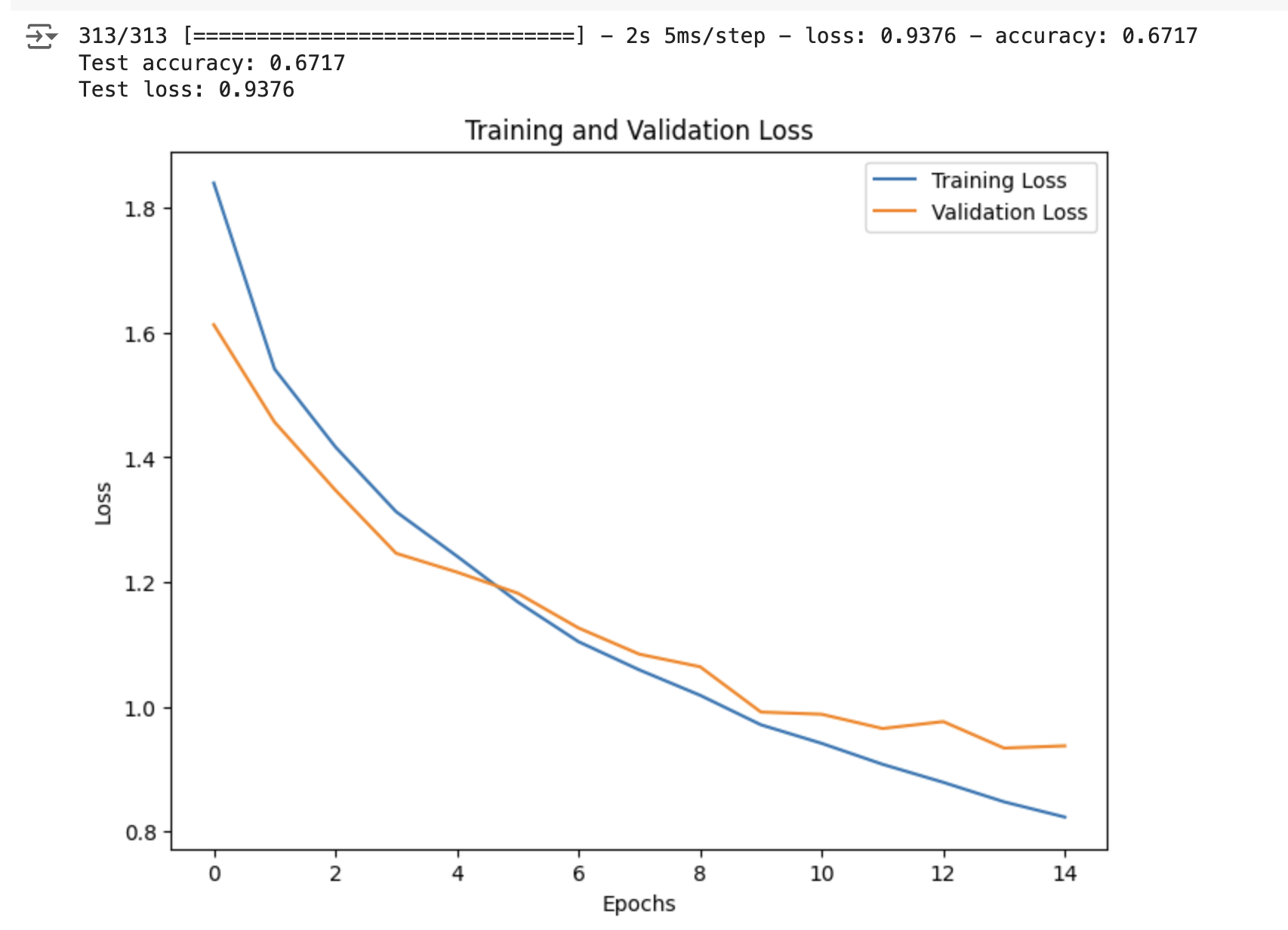
The invariance of the softmax function to constant offsets is significant in practical implementations of neural networks. It ensures that the function's output remains consistent regardless of uniform shifts in input values, thus improving numerical stability and preventing overflow or underflow during computations.

1. It is important because it allows us to shift the pre-softmax values by any constant value without changing the final output probabilities. This can be useful for numerical stability when dealing with very large or very small logit values, as adding or subtracting a constant can prevent overflow or underflow issues during computation.

# Training a CNN using CIFAR-10 Data



The images are very vague when just loaded.

CNN with three hidden layers:

In this exercise, we implemented and trained a Convolutional Neural Network (CNN) on the CIFAR-10 dataset, which consists of 60,000 32x32 color images in 10 classes. The goal was to train a model from scratch without using any pre-trained networks.

The CNN architecture consists of three hidden convolutional layers, each using the ReLU activation function. The specific configuration is as follows:

* First Convolutional Layer: 64 filters of size 11x11, followed by a 2x2 max pooling layer with a stride of 2.
* Second and Third Convolutional Layers: Each has 128 filters of size 3x3 with ReLU activation.
* Average Pooling Layer: Pools across the preceding feature map.
* Output Layer: Dense layer with softmax activation for classification into 10 categories.

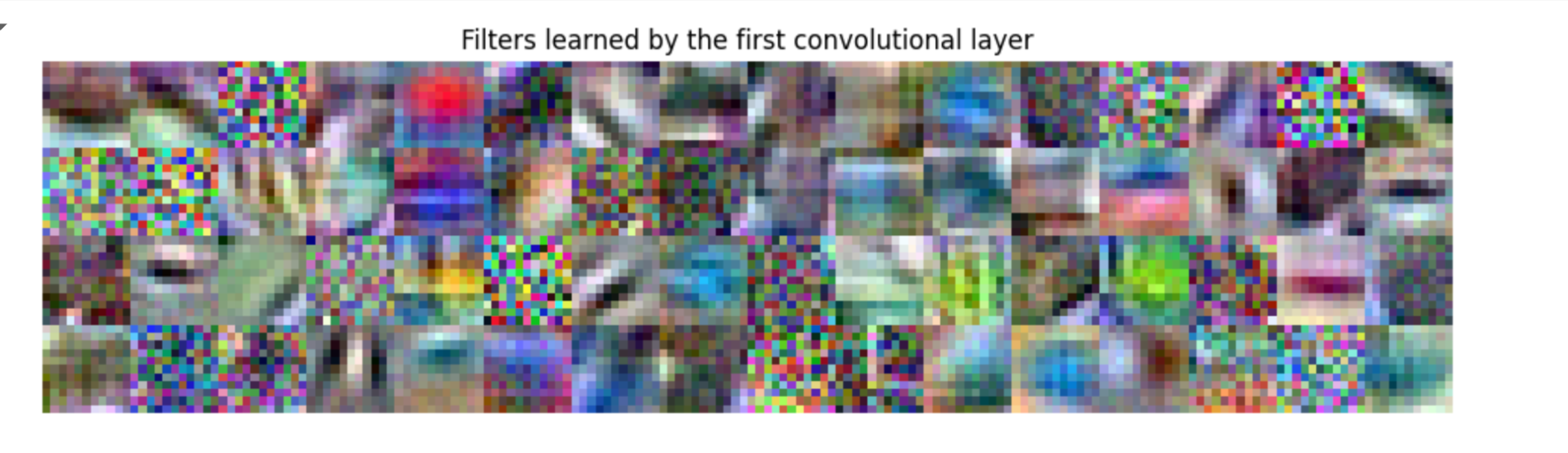
The model was trained for 15 epochs with a batch size of 128. The Adam optimizer was used with categorical cross-entropy as the loss function. The training loss and validation loss were plotted across epochs, showing a decreasing trend, which indicates effective learning.

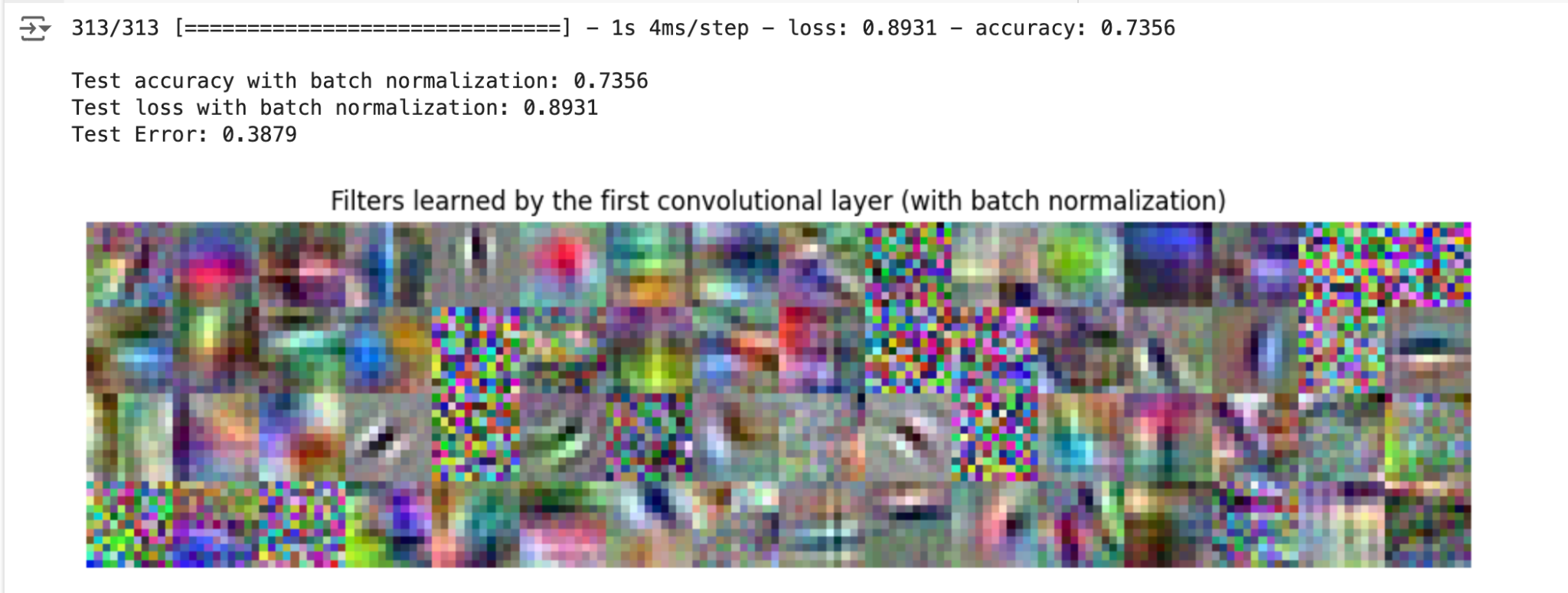
The final test accuracy achieved was 67.17%, with a corresponding test loss of 0.9376. This performance demonstrates the model's ability to generalize from the training data to unseen test data, though there is room for further optimization.

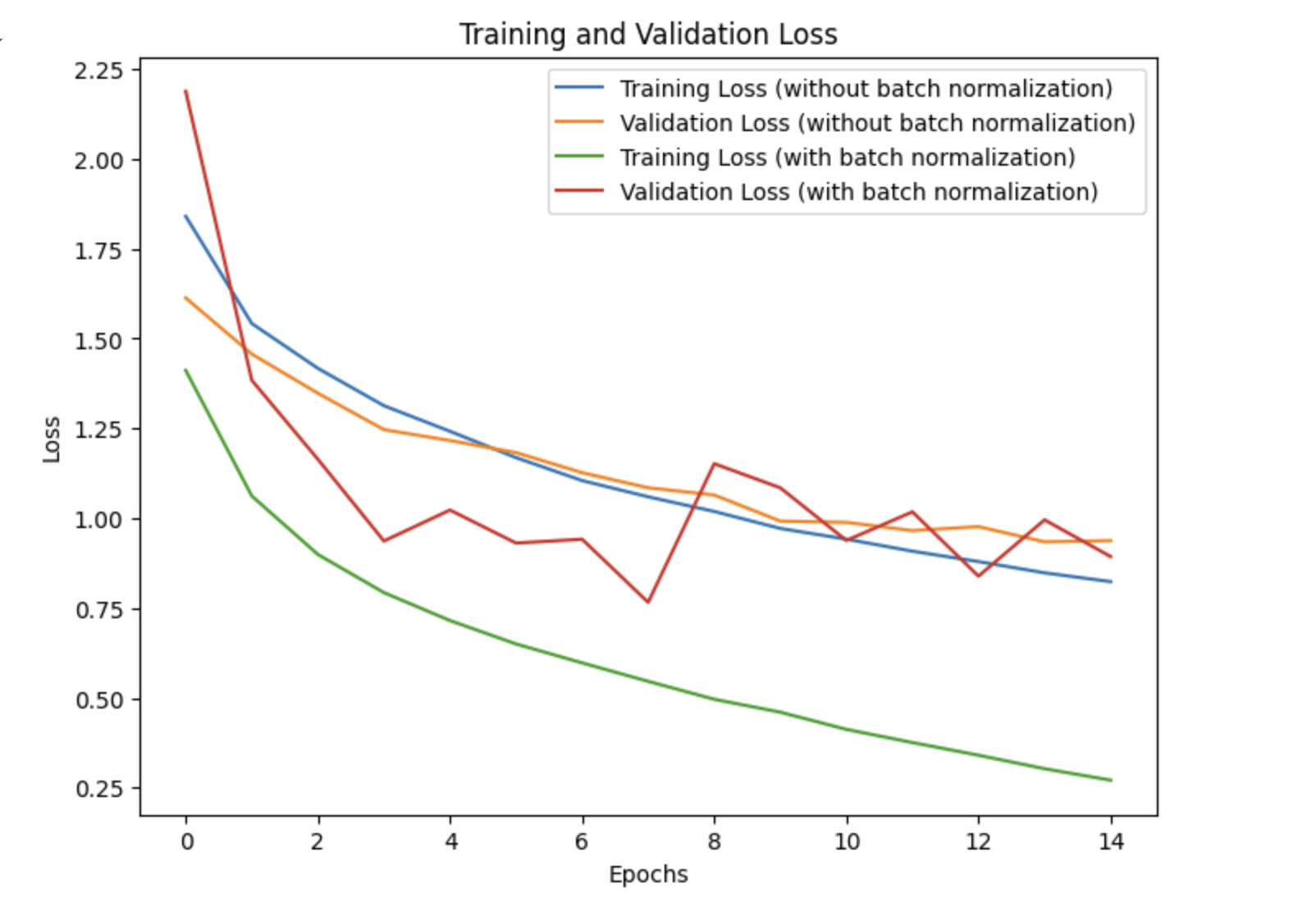
Hyperparameters

* Batch size: 128 (chosen for efficient training and manageable memory usage).
* Epochs: 15 (to ensure sufficient training without overfitting).
* Optimizer: Adam (for its adaptive learning rate properties).
* Loss function: Categorical cross-entropy (appropriate for multi-class classification problems).

The training process highlighted the effectiveness of the chosen architecture and hyperparameters in handling the CIFAR-10 dataset, with a balanced trade-off between complexity and performance. Future improvements could explore deeper networks, data augmentation, and regularization techniques to further enhance accuracy.







#### **Model Architecture**

The original CNN architecture included three hidden convolutional layers with ReLU activation, followed by a max pooling layer, average pooling layer, and a final dense layer with softmax activation for classification. For this experiment, batch normalization was added between each of the hidden layers to assess its effect.

**Results**

The comparison of training loss and validation loss with and without batch normalization is illustrated in the graph provided. The model with batch normalization demonstrated a significantly lower training loss and validation loss compared to the model without batch normalization. This indicates that batch normalization helped the model converge faster and reduced overfitting.

The final test accuracy of the model with batch normalization was 73.56%, with a test loss of 0.8931 and a test error of 0.3879. This is a notable improvement over the model without batch normalization, which had a lower test accuracy.

**Conclusion**

Batch normalization had a significant positive impact on the performance of the CNN. It not only improved the convergence speed and reduced the training and validation loss but also enhanced the overall test accuracy and reduced the test error. This experiment underscores the importance ofbatch normalization in training deep neural networks, particularly for image classification tasks such as those involving the CIFAR-10 dataset. Future work could explore further optimizations and different normalization techniques to continue improving model performance.

GAN using Fasion-MNIST Data

## Vanilla GAN

**Model Architecture**

* Dense layer with 256 units, followed by Batch Normalization and ReLU activation.
* Dense layer with 512 units, followed by Batch Normalization and ReLU activation.
* Dense layer with 1024 units, followed by Batch Normalization and ReLU activation.
* Dense layer with 28x28x1 units and tanh activation, reshaped to (28, 28, 1).

**Discriminator Model**

* Flatten layer to convert the 28x28x1 input into a 784-dimensional vector.
* Dense layer with 1024 units and ReLU activation.
* Dense layer with 512 units and ReLU activation.
* Dense layer with 256 units and ReLU activation.
* Dense layer with 1 unit and sigmoid activation.

**Conclusion**

The Vanilla GAN successfully learned to generate realistic images from the Fashion-MNIST dataset. The inclusion of Batch Normalization in the generator model helped stabilize training and improve the quality of generated images. This experiment highlights the effectiveness of GANs in generating realistic data and their potential applications in various fields such as image synthesis and augmentation. Future work could explore advanced GAN architectures, such as Conditional GANs or Wasserstein GANs, to further enhance performance and address challenges like mode collapse.

1. Mode Collapse in GANs

The histogram of the predicted labels for the 3000 generated samples is shown below. The original Fashion-MNIST dataset has 10 classes that are equally distributed, but the histogram of the generated samples is not close to uniform. This indicates that the GAN suffers from mode collapse, as it fails to generate a diverse set of samples across all classes.

Exploring the Capabilities of ChatGPT

1. Mental health issues are rising globally, and many people often lack access to immediate support. Traditional mental health support systems often have limited resources, and long waiting time. Many people experience mental health crises outside regular hours when professional help may not be available.

ChatGPT-4 can offer immediate, personalized mental health support and counseling. Imagine it in a simple, easy-to-use app where people can get real-time help tailored to their needs. It can chat with users, listen empathetically, suggest coping strategies, and provide useful resources. While it can't replace a professional therapist, it can be a valuable support during tough times and guide users to professional help when needed.

1. Potential Impact: More Accessibility: This app can give people 24/7 mental health support, especially those living in remote areas or who can't easily access professional services. Immediate Assistance: It offers quick help during mental health crises, which might prevent harm and provide comfort until someone can reach professional help. Less Stigma: It provides a private way to seek help, which can reduce the stigma around mental health issues.Supporting Professionals: By providing initial support, it can lighten the load for mental health professionals, letting them concentrate on more severe cases.
2. Deep learning and language models like ChatGPT have massive potential across various fields. They're great at understanding and generating human-like text, which makes them perfect for applications that need natural language processing. The model can be used in healthcare, education, customer service, content creation, and even legal assistance.

In our future work, we aim to integrate deep learning and language models into applications that need advanced language understanding. We thought to integrate deep learning and lanuguage models into smart chatbots for healthcare, education, and customer service to provide quick, accurate responses. We also expect it to be human's personal assistants to manage schedules, provide reminders, and offer personalized recommendations.