**Deep Learning Project 2**

**Chaitanya Sai Juturi**

**16355022**

VideoLink: <https://d2y36twrtb17ty.cloudfront.net/sessions/6639612d-0089-4512-973c-b151004d9626/76fcca7e-e313-4021-8f2e-b151004d9633-6f4043ee-a48e-40e0-8f31-b151004ed687.mp4?invocationId=0e94e3fc-50f9-ee11-8291-12c206d2fd2b>

GitHubLink: <https://github.com/Chay039/DeepLearningProject2>

1. **A 1 page of discussion – your observations when adjusting the parameters.**

When adjusting the parameters for the fully connected networks (FC) and convolutional neural networks (CNN) on the Fashion MNIST dataset, I made the following observations:

**Fully Connected Networks**

**Hidden Layer Sizes and Number of Layers**

- Increasing the number of hidden layers and nodes generally improved performance up to a certain point, after which overfitting became an issue.

- Smaller networks with fewer hidden layers and nodes tended to underfit the data, resulting in lower accuracy.

- The optimal network size seemed to be around 3-4 hidden layers with moderate node counts (e.g., 64-128 nodes per layer).

**Learning Rate**

- The learning rate had a significant impact on the training process and final performance.

- Lower learning rates (e.g., 0.001) tended to converge more slowly but achieved better final accuracy, as they allowed for more stable updates and avoided overshooting local minima.

- Higher learning rates (e.g., 0.003) converged faster initially but often got stuck in suboptimal solutions or diverged entirely.

**Optimizer**

- The choice of optimizer also influenced the training process and final performance.

- Adam and RMSprop generally performed better than SGD, as they adapt the learning rate for each parameter and handle sparse gradients more effectively.

- SGD with a carefully tuned learning rate and momentum could still achieve good results, but it was more sensitive to the hyperparameter settings.

**Convolutional Neural Networks**

**Kernel Size and Max Pooling**

- Smaller kernel sizes (3x3) and max pooling sizes (2x2) tended to work better than larger sizes for this dataset.

- Larger kernel and pooling sizes (e.g., 5x5 and 3x3) led to excessive information loss and poorer performance.

**Activation Function**

- The ReLU activation function consistently outperformed other activation functions (e.g., sigmoid, tanh) for the convolutional and dense layers.

- ReLU's non-saturating behavior and sparse activations seemed to be well-suited for this task.

**Regularization**

- While not explicitly explored in this experiment, regularization techniques like dropout and L2 regularization could potentially improve performance and prevent overfitting, especially for larger CNN architectures.

Overall, finding the right balance between model complexity and generalization was crucial for achieving good performance on the Fashion MNIST dataset. Larger networks with more parameters had the potential for better accuracy but also risked overfitting without proper regularization and hyperparameter tuning. Smaller networks were more prone to underfitting but were less computationally expensive and easier to train.

1. **1-2 pages of figures, tables, and graphs which support the observations made in the discussion.**

3. Compare the results of your experiments for Part 1 and Part 2 – use the values from your

recorded model performance to generate at least (2) meaningful figures related to your results.

a. Display the results of the test performance for each experiment in a single graph.

(preferred).

A screenshot of a graph

Description automatically generated

b. Provide a table or plot showing how complexity of the model contributed to challenges.

when training both the FC and CNN implementation. (Did you overfit or stop learning?)

A screenshot of a graph

Description automatically generated

4.

## Best Performing Model

Based on the results, the best performing model appears to be the one with the following configuration:

```

{'kernel\_size': (3, 3), 'pool\_size': (2, 2), 'activation': 'relu'}

```

This configuration achieved the highest test accuracy among all the models trained. The combination of a 3x3 kernel size, 2x2 max pooling, and ReLU activation function seems to work well for the Fashion MNIST dataset.

## Worst Performing Model

The worst performing model in this experiment was the one with the following configuration:

```

{'kernel\_size': (5, 5), 'pool\_size': (3, 3), 'activation': 'sigmoid'}

```

This configuration likely struggled due to the larger kernel size (5x5) and max pooling size (3x3), which may have caused too much information loss during the convolution and pooling operations. Additionally, the sigmoid activation function can be less effective than ReLU for deep neural networks.

**a. Were larger networks (structures with more hidden nodes) worth the trade off in training time?**

In the case of the fully connected networks (FC) trained on the Fashion MNIST dataset, it's difficult to draw a definitive conclusion about whether larger networks were worth the trade-off in training time. The results show that some larger networks performed better than smaller ones, but there were also cases where smaller networks outperformed larger ones.

Generally, larger networks have more parameters and can potentially learn more complex representations, but they also require more training time and are more prone to overfitting. The optimal network size depends on the complexity of the task and the amount of available training data.

For the Fashion MNIST dataset, which is relatively simple, it's possible that the performance gains from larger networks were not significant enough to justify the increased training time and complexity.

**b. While the performance between FC and CNN can be large, does the training time and complexity of the CNN seem necessary for this task?**

Based on the results, the convolutional neural networks (CNNs) achieved significantly higher accuracy compared to the fully connected networks (FC) on the Fashion MNIST dataset. This is not surprising, as CNNs are particularly well-suited for image classification tasks due to their ability to learn and extract relevant features from the input images.

While CNNs are generally more complex and require more training time compared to FC networks, the performance gains they provide for image classification tasks often justify their use. The Fashion MNIST dataset, although relatively simple, still benefits from the feature extraction capabilities of CNNs.

However, it's important to note that the training time and complexity of CNNs can vary depending on the specific architecture and hyperparameters used. In this experiment, the CNN models were relatively small and simple, with only a few convolutional and pooling layers. As a result, their training time and complexity may not have been significantly higher than the FC networks.

For more complex image classification tasks or larger datasets, the performance benefits of CNNs would likely be even more pronounced, and the additional training time and complexity would be justified by the improved accuracy.