**COMP\_SCI-5567-0001 – Deep Learning**

**Deep Learning Project 2**

**A Comparison of Architectures Feature Extraction and Classification Performance**

**Fully Connected ANNs vs CNN using fashionMNIST**

**[composed with Keras]**

**Submitted by:**

**Sai Deeksha Enukonda**

**ID: 16343736**

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

Here are my observations when adjusting the parameters for the CNN models on the Fashion-MNIST dataset:

* **Convolutional Layers**: Exploring 2-3 layers revealed a trend where configurations with 3 layers generally outperformed those with 2. This suggests that the added depth allowed the model to capture more intricate features, contributing to enhanced performance.
* **Filters**: With counts ranging from 16 to 64, configurations with higher filter numbers, such as 32 and 64, consistently demonstrated better performance. This indicates that a richer set of features was captured, facilitating improved discrimination between clothing categories.
* **Kernel Sizes**: Utilizing 3x3 kernels across all configurations is a common practice for image tasks. While the impact on performance couldn't be conclusively assessed due to uniformity, this choice aligns with standard practices in convolutional neural networks.
* **Pool Sizes**: Employing 2x2 max pooling uniformly, the influence of pool size variation couldn't be determined from the results. Nevertheless, this choice aligns with typical down sampling strategies in CNNs for preserving important features while reducing spatial dimensions.
* **Dense Layer Sizes**: All configurations featured a single dense layer with 128 units, indicating that this size was sufficient for capturing high-level abstractions from the extracted features. Larger dense layers might have led to overfitting or increased computational complexity without significant performance gains.
* **Learning Rates and Optimizers**: Configurations with lower learning rates, particularly 0.001 for Adam, tended to perform better. The Adam optimizer consistently outperformed SGD and RMSprop, indicating its effectiveness in optimizing the model parameters for this task.
* **Batch Sizes**: Ranging from 64 to 128, there was no clear superiority of one batch size over the other. Both batch sizes produced satisfactory results, suggesting that the choice of batch size may not have been a critical factor for this particular task.

In summary, configurations achieving around 90% accuracy typically featured 2-3 convolutional layers, 32-64 filters, 3x3 kernels, 2x2 pooling, a 128-unit dense layer, lower learning rates (e.g., 0.001 for Adam), and the Adam optimizer. However, the absence of a consistently superior configuration underscores the nuanced influence of parameter interplay on model performance.

Top of Form

Bottom of Form

**Discussion:**

**1. While the MNIST dataset is perhaps the most frequently utilized dataset in ML courses, FashionMNIST is considered much more challenging. Explain why the samples in the dataset you used in this exercise seem harder to classify than the numerical identification tasks.**

**Ans:**

FashionMNIST dataset is considered more challenging than the traditional MNIST dataset due to several reasons:

1. **Complexity of Patterns:**

FashionMNIST contains images of clothing items and accessories, which have more intricate patterns and textures compared to handwritten digits in MNIST. This complexity makes it harder for the models to discern distinct features for classification.

1. **Variability in Appearance:**

Clothing items come in various shapes, sizes, colors, and styles. This variability introduces more diversity and ambiguity into the dataset, making it challenging for models to generalize well across different classes.

1. **Intra-class Variability:**

Within each class (e.g., shirts, shoes, dresses), there can be significant variations in appearance, such as different styles, textures, and orientations. This intra-class variability poses a challenge for models to learn representative features for each class.

1. **Imbalance and Overlapping Classes:**

FashionMNIST dataset may have class imbalances and overlapping classes, where certain classes have fewer samples or share similar visual characteristics with other classes. This makes it difficult for the model to distinguish between these classes accurately.

1. **Real-world Relevance:**

Classifying fashion items has real-world applications, such as e-commerce, image search, and fashion recommendation systems. Therefore, the task is more practical and challenging compared to digit recognition.

**2. Utilizing the template attached, provide the calculations for map dimensions, number of weights and number of bias terms for your top performing CNN model.**

| **Layer** | **Activation Map Dimensions** | **Number of Weights** | **Number of Biases** |
| --- | --- | --- | --- |
| **Input** | 28x28x1 | 0 | 0 |
| **CONV2D (dims)** | 26x26x16 | 416 | 16 |
| **POOL-2** | 13x13x16 | 0 | 0 |
| **CONV2D (dims)** | 11x11x32 | 4640 | 32 |
| **POOL-2** | 5x5x32 | 0 | 0 |
| **FC-128** | 1x1x128 | 12800 | 128 |
| **FC-10** | 1x1x10 | 1290 | 10 |

This table represents the layer-wise dimensions, weights, and biases for Configuration 9 as it is the top performing Configuration. The convolutional layers reduce spatial dimensions while increasing depth, with pooling layers further down sampling. Fully connected layers contribute significantly to the total weights and biases, providing the network's output.

**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 graph with orange and blue lines

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 graph with blue and orange lines

Description automatically generated

A graph with blue and orange lines

Description automatically generated

**4. Discuss in a few sentences the results of your best and worst performing model.**

**Ans:**

**Best and Worst Performing Models**

* Based on the results, the best performing model was the MLP with the following configuration:

**hidden\_layers: [512, 256, 128, 64]**

**batch\_size: 128**

**lr: 0.001**

**optimizer: Adam**

This model achieved an accuracy of around 89.5% on the test set.

* On the other hand, the worst performing model was the MLP with the following configuration:

**hidden\_layers: [128, 64, 32]**

**batch\_size: 512**

**lr: 0.0005**

**optimizer: SGD**

This model achieved an accuracy of around 86.5% on the test set.

**a. Were larger networks worth the trade off in training time?**

In general, larger networks with more hidden nodes tend to have higher representational capacity and can potentially achieve better performance on complex tasks. However, this comes at the cost of increased training time and computational resources.

For the MLP models in this task, the configurations with larger hidden layer sizes (e.g., [512, 256, 128, 64]) generally performed better than those with smaller hidden layer sizes (e.g., [128, 64, 32]). This suggests that the added complexity of larger networks was beneficial for this task, and the trade-off in training time was justified by the improved performance.

However, it's worth noting that the performance gains from increasing the network size beyond a certain point may diminish or become negligible. Therefore, it's important to strike a balance between model complexity and training time based on the specific requirements of the task and the available computational resources.

**b. Performance and Complexity of CNN vs. FC**

CNNs consistently outperformed MLPs on Fashion-MNIST, achieving around 92% accuracy compared to MLPs' 89.5%. CNNs excel at capturing spatial features crucial for image tasks like Fashion-MNIST. Despite their complexity, CNNs in this case were manageable, with the best model featuring 3 conv layers, 64 filters, and a single dense layer.

The necessity of CNN complexity depends on task complexity; for more intricate tasks or data, deeper architectures may be essential. Data augmentation could further boost performance, especially for CNNs, which leverage spatial information effectively.

Thus, while CNNs may require more computational resources, their ability to handle complex tasks and leverage augmented data makes them worthwhile in many scenarios.

**Video Link:**

[**https://d2y36twrtb17ty.cloudfront.net/sessions/b41890b6-fbf4-43a5-a1e3-b151004c3e95/f7cd5d2c-6e7c-4afc-8672-b151004c3e9e-07f1612d-67e9-42fb-b26a-b151004f18ce.mp4?invocationId=81749127-51f9-ee11-8291-12c206d2fd2b**](https://d2y36twrtb17ty.cloudfront.net/sessions/b41890b6-fbf4-43a5-a1e3-b151004c3e95/f7cd5d2c-6e7c-4afc-8672-b151004c3e9e-07f1612d-67e9-42fb-b26a-b151004f18ce.mp4?invocationId=81749127-51f9-ee11-8291-12c206d2fd2b)