**Image Classification with Fashion MNIST Dataset**

**Submitted by:**

**ALLAN BURTON – IN13/00054/19**

**DANIEL KINUTHIA – IN13/00106/20**

**BRUCE SONYE – IN13/00078/20**

**Submission Date: 20th March 2024**

**1. Introduction**

The objective of this assignment is to develop a neural network model capable of accurately classifying images from the Fashion MNIST dataset into their respective categories. The assignment involves data preprocessing, model development, training, evaluation, and analysis.

**2. Data Preprocessing**

a) Loading the Dataset

The Fashion MNIST dataset consisting of 60,000 training images and 10,000 test images was loaded using TensorFlow.

b) Normalization

Image data was normalized to range between 0 and 1 to facilitate faster convergence during training.

c) Reshaping

Data was reshaped to fit the input requirements of the model, adding a channel dimension for convolutional layers.

d) Label Encoding

Labels were converted into one-hot encoded vectors to represent the categorical classes.

**3. Model Development**

Convolutional Neural Network (CNN)

A CNN architecture was designed comprising convolutional layers with ReLU activation, max-pooling layers, a flattening layer, dense layers with ReLU activation, and a softmax output layer.

Compilation

The model was compiled with the Adam optimizer, categorical cross-entropy loss function, and accuracy as the evaluation metric.

**4. Model Training**

Training Procedure

The model was trained on the training data using the `model.fit()` function, specifying the number of epochs.

Validation

Model performance was validated using the test data to assess its generalization ability.

**5. Evaluation and Analysis**

Performance Metrics

The model's performance on the test dataset was evaluated using the `model.evaluate()` function to obtain accuracy and loss metrics.

Visualization

Training and validation accuracy and loss curves were plotted over epochs using Matplotlib to visualize the learning process.

Sample Predictions

Sample images with their predicted and actual labels were displayed to assess the model's performance qualitatively.

**6. Discussion**

Challenges

One of the challenges faced during the assignment was understanding the inner workings of complex neural networks and diagnosing issues such as vanishing gradients or dead neurons.

Designing a CNN architecture that is sufficiently complex to capture the nuances of the dataset without overfitting was also challenging.

Solutions

To overcome these challenges, we used visualization techniques such as tensorboard to visualize model performance, gradients, and activations during training. Additionally, we used debugging tools provided by TensorFlow and Keras to identify and address issues in the model architecture or training process.

We also experimented with different CNN architectures, layer configurations, and regularization techniques such as dropout and batch normalization to mitigate overfitting.

Improvements

Possible improvements to the model's performance include implementing architectural modifications like skip connections or attention mechanisms to enable the model to focus on relevant features and reduce the impact of noise.

Experimenting with learning rate schedules such as exponential decay or cosine annealing to dynamically adjust the learning rate during training.

Fine-tune the pre-trained models on the Fashion MNIST dataset by freezing certain layers and only updating the weights of specific layers during training.

**7. Conclusion**

In conclusion, the assignment provided valuable insights into image classification using deep learning techniques. The developed CNN model demonstrated satisfactory performance on the Fashion MNIST dataset, achieving 90.9% accuracy on the test data. Further experimentation with model architectures and hyperparameters could potentially enhance performance.

**Link to our colab notebook**

https://colab.research.google.com/drive/1ofSNuEICs5QQ5NCSbRAv4V8BTsQu0ffH?authuser=0#scrollTo=X8mh0mNmhk0N