Assignment 3 - Machine Learning

We tried two different neural network-based model, one is fully connected neural network model in notebook ‘Classification_NN.ipynb’ and ‘Hyperparameters_tune_nn’, the other one is convolutional neural networks in ‘fasion_cnn_classification’

1. Plot the training and validation error and accuracy metrics.
    a. We followed a guide from tensorflow, where we went through the guide, testing different epochs to check for the accuracy and whether or not it starts to overfit or underfit. We initially tested with 15 epochs, and then another with 50. We observed we already started overfitting before 15 epochs, and far more overfitting with 50. To find a balance, we attempted to test it with 10 epochs, even then we observed overfitting. Overfitting started at around 4 epochs. We found a balance to be around 10 epochs. https://www.tensorflow.org/tutorials/keras/classification
2. What hyperparameters did you use to tune your model and why? For example, the number of hidden layers, number of neurons, activation function, input features, etc.
    a. We referred to this guide (https://www.tensorflow.org/tutorials/keras/keras_tuner) to use Keras tuner on the hyperparameters tuning. The parameters include model parameters and algorithm parameters. These hyperparameters significantly influence the model's ability to capture patterns in the data, control overfitting, or improve training stability. By tuning them, we can optimize the balance between model complexity, performance, and training efficiency, ensuring the neural network can accurately classify images without overfitting or underfitting.
        i. For the model parameters, we tried
            1. “Number of Filters in Convolutional Layers (`conv_1_filters` and `conv_2_filters`)”,
               Why: By tuning the number of filters, we aim to balance feature extraction capability with computational efficiency.
            2. Kernel Size in Convolutional Layers (`conv_1_kernel_size` and `conv_2_kernel_size`),
               Why: Choosing the right kernel size helps the model better learn important spatial features in the data
            3. **Number of Units in the Dense Layer (`dense_units`).**
               Why:. Tuning this parameter helps to find an optimal model capacity.
            4. Dropout Rate. (`dropout_rate`)

        5. Why :Tuning the dropout rate helps in finding the best trade-off between regularization and model capacity, improving generalization on unseen data.

    ii. For the algorithm parameters, we tried

        1. Learning Rate.
           Why: Tuning the learning rate is essential to optimize the model's training process and achieve better performance in fewer epochs

Collaboration

Zoie handled writing the actual code and notes in the jupyter notebook, while Emma researched things, to help explain theory from the tensor guide, as well as being an active participant in discussing how to proceed in terms of epochs, model and the like. Initially, we were going to split up pieces into two notebooks each, but Emma had problems setting up tensor, trying to install it and run didn't work, some problems with pip with Windows. Therefore, we opted for Zoie to write and run the training, while Emma played support. Zoie did all the research and testing for the CNN model, as we noticed that the first model wasn't very good, and she wanted to explore a better model.