Advanced Machine Learning Workshop

Assignment 5: CarNet - Description of the Solution

Priyanka Rotte

1. **Individual Models:**

   For individual models, I'm using transfer learning from the EfficientNetV2B1 model. I tried various models available in the Keras library based on Size, Accuracy, Parameters, Time, etc. and EfficientNetV2B1 performed the best for the given data. I'm removing the fully-connected layer at the top of the model and using global average pooling on the output of the layer before. Average pooling worked better than flattening, as the output shape was getting too high after flattening.

   After EfficientNetV2B1, I'm using two separate models for vehicles and signals, as to enable the models to learn features specific to the type. After several experiments, I'm going with two dense layers of sizes 64 and 32, with ReLU activations. Finally, there's an output dense layer with 1 neuron. I'm not using any dropouts or regularization as the ensemble method with bootstrap replicates should prevent overfitting. I'm using Adam optimizer with a 0.001 learning rate, MSE loss, and early stopping.

2. **Ensemble Model (Blender):**

   I'm training 7 individual models with the same architecture with different kernel initializations and bootstrap replicates i.e., random resampling of the dataset with replacement.

   After training the individual models, I'm training an ensemble model with the outputs of these 7 models as inputs to the ensemble. Instead of just getting an average, I'm using a blender i.e., a neural network on top of the individual models. Here too, I'm using two separate models for vehicles and signals, just with one dense layer of 128 and ReLU, and one dropout (even dropout will act as an ensemble of models and prevent overfitting). This architecture gave the lowest validation MSE. Finally, there's an output dense layer with 1 neuron. The training is performed similarly as for the individual models. I'm saving the final model with the least validation loss. Finally, I'm getting a validation MSE of 4.29 and a test MSE of 4.36.

Submission:

Jupyter notebook with code for the model

Parquet file with predictions on score

Zip file

1. individual_models has individual folders for each model with their model definition, parameters and predictions on train, validate, test, and score.
2. ensemble_model has ensemble model definition, parameters, and predictions on score set. Folder inside, ensemble_alternate has the other way of creating the same ensemble.

PDF with solution description