Machine Learning for IOT - Homework 2

Sepehr Alidousti Shahraki, Shadi Nikneshan, Ali Ghasemi Group 22

Exercise 1

In this exercise we first create our own window generator function to create the dataset we want to feed to our model (more information about the code is available in the comments in the code file). We tested three different models for two scenarios in the homework file. We test LSTM, MLP and, CNN (LSTM model didn't give us good enough results so we avoided using it in the report). For both output steps of 3 and 9, we used the CNN model. Before the fitting process, normalization was performed on the data. After the fitting process, the pruning procedure was performed to reduce the size of the TFLITE model file as much as possible. We have reduced the number of units using the value alpha to reduce the complexity of our model. The best value available for alpha was 0.25. This value is obtained after experimenting with different values in the range of 0.25 to 0.75. The values for metrics and loss in the fitting process are as mentioned in the lab sessions and for the pruning process the values for initial sparsity and final sparsity have been found after experimenting with different possible values. We also had to tweak the architecture of the CNN model to decrease the size of the TFLITE model, we removed some layers of the CNN in order to achieve that. You can find the results of the models in the table below. The MAE values are measured after the pruning process.

Model	MAE(Temperature)	MAE(Humidity)	Output size(KB)	Steps
CNN	0.033284843	0.070087954	1.445	3
CNN	0.06273821	0.1319452	1.477	9

Exercise 2

In this exercise we first chose different values for frame length, frame step, number of Mel bins, frame steps and, MFCC (using it or not) to get the best combination for the data that we're using for our model. After this step different ML and Deep Learning models (DS-CNN, CNN, MLP) had to be tested in order to find the best model and the best hyperparameters for them. The best combinations of models and their hyperparameters and the best combination (that satisfy the requirements for the problem) for the parameters that create our data and their results (accuracy and latency), and their sizes are mentioned in the table below. The value for alpha is the value that we use to control the number of units in each layer of our CNN and DSCNN models. We used magnitude-based pruning techniques to reduce the size of our TFLITE model and after that, compression was performed to reduce the file size even more. It is also good to mention that at the beginning we kept on using fixed values for frame length and frame steps but to get the desired criteria for the "C" version of our file we had to try different values for them.

Model	Problem number	Alpha	Momentum	Epochs	Frame Length	Frame Steps	Number of Mel Bins	Number of Coefficients	Number of Filters	Accuracy	Size (KB)	Latency (ms)
CNN	a	0.17	0.3	20	650	350	30	10	512	92.75	125.738	-
DSCNN	b	0.26	0.9	20	650	350	30	10	512	92.5	41.282	56
CNN	c	0.11	0.9	25	512	322	30	10	256	91.125	16.628	23.15

Command for latency (b): python kws_latency.py --mfcc --length 650 --stride 350 --bins 30 --coeff 10

Command for latency (c): python kws_latency.py --mfcc --length 512 --stride 322 --bins 30 --coeff 10