redline communications

ITU-ML5G-PS-007: AUTOMATIC MODULATION CLASSIFICATION

December 2, 2021

RED GECKO TEAM

Members:

- Brad Stimpson
- Shahram Mollahasani (Postdoc at Dr. Melike Erol-Kantarci's lab. at University of Ottawa)
- Armand Kamary

Advisors:

- Alex Shatsky
- Felix Dujmenovic
- Ali Arad

LITERATURE REVIEW AND STUDY

LIST OF ARTICLES

- Real-time Automatic Modulation Classification using RFSoC, Stephen Tridgell *et al.*
- Automatic Modulation Classification Using CNN-LSTM Based Dual-Stream Structure, Zufan Zhang *et al.*
- AMC-IoT: Automatic Modulation Classification Using Efficient Convolutional Neural Networks for Low Powered IoT Devices, Muhammad Usman *et al.*
- Deep Learning Techniques for Automatic Modulation Classification: A Systematic Literature Review, Sara Ali Ghunaim *et al.*
- Automatic Modulation Classification Using Gated Recurrent Residual Network, Sai Huang *et al.*
- Acceleration of Deep Convolutional Neural Networks Using Adaptive Filter Pruning, Pravendra Singh *et al.*
- Hardware Implementation of Automatic Modulation Classification with Deep Learning, Satish Kumar *et al.*

ALTERNATIVE NEURAL NETWORKS

- Thinking out of the box

ALTERNATIVE NETWORKS

Our team has implemented several alternative automatic modulation classification neural networks such as

- VGG-16
- Ternary Weight Network (TWN) based on Stephen Tridgell *et al.* research

RED GECKO TEAM APPROACH

SNR BASED OUTLIER REJECTION

- Low SNR samples act as outliers and reduce the accuracy

- We manually removed/pruned the low SNR samples

```
# you could train on a subset of the data, e.g. based on the SNR
# here we use all available training samples
if snr_idx >= 4:
   train_indices.extend(train_indices_subclass)
test_indices.extend(test_indices_subclass)
```

CHANGE THE NETWORK HYPERPARAMETERS

- We set number of input quantization, weights and activation bits to **5**
- We downsized the kernel size of our convolutional layers to **28**
- Following the above change, we downsized the dense layer input size to **28**

PRUNING

We use *L1 unstructured pruning* technique with ratio 0.5 on all convolutional layers.

```
def prune_model_l1_unstructured(model, layer_type, proportion):
    for module in model.modules():
        if isinstance(module, layer_type):
            prune.l1_unstructured(module, 'weight', proportion)
            prune.remove(module, 'weight')
    return model

model = prune_model_l1_unstructured(model, qnn.QuantConv1d, 0.5)
```

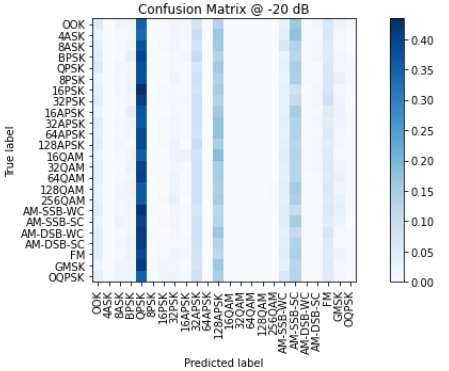
TRAINING IMPROVEMENTS

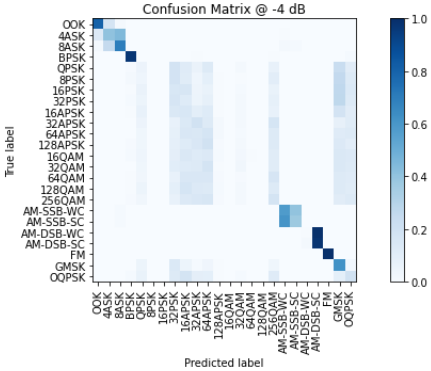
- We changed the starting learning rate of *Adam* optimization algorithm to **0.1**

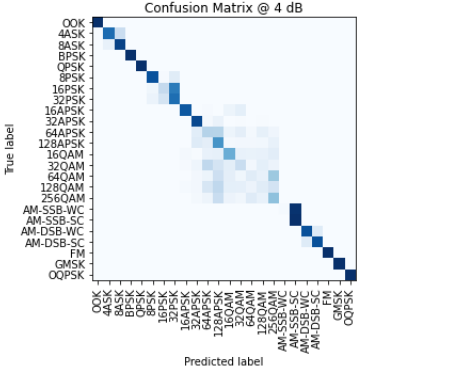
- We further changed the learning rate scheduler:
  - For epochs less than 100 we used **Cosine Annealing** learning rate scheduler
  - For epochs beyond 100 we used **Stochastic Weight Averaging (SWA)** learning rate scheduler

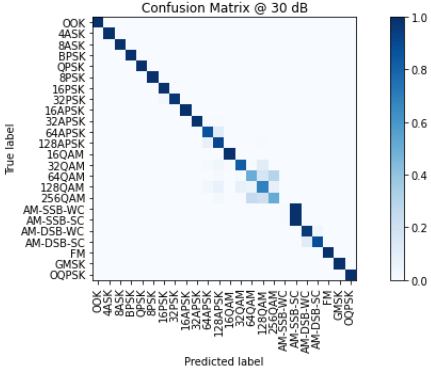
```
optimizer = torch.optim.Adam(model.parameters(), lr=0.01)
swa_model = torch.optim.swa_utils.AveragedModel(model)
swa_scheduler = torch.optim.swa_utils.SWALR(optimizer, swa_lr=0.001)
lr_scheduler = torch.optim.lr_scheduler.CosineAnnealingLR(optimizer, T_max=num_epochs)
```

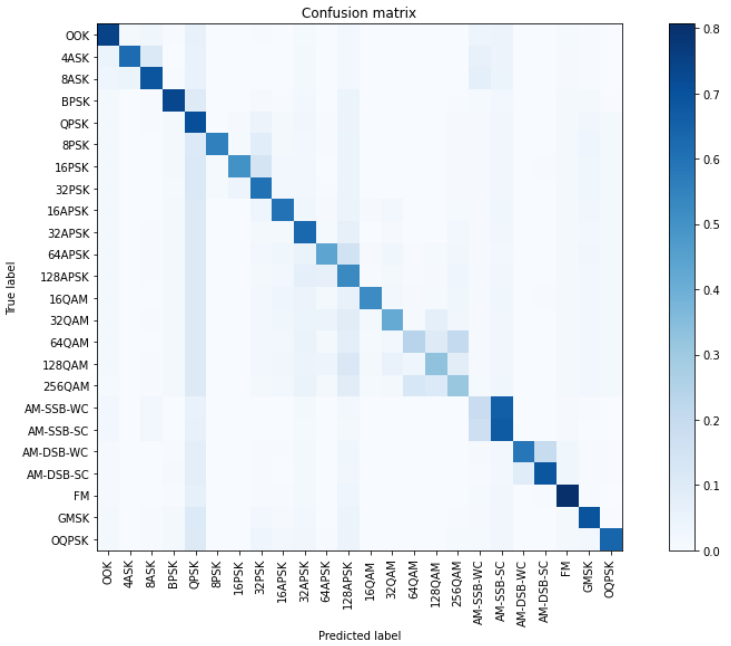
RESULTS





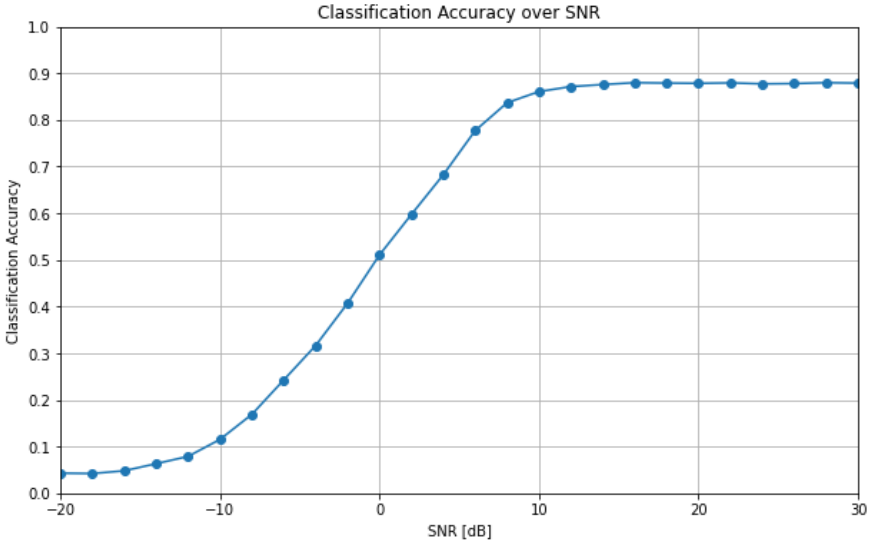


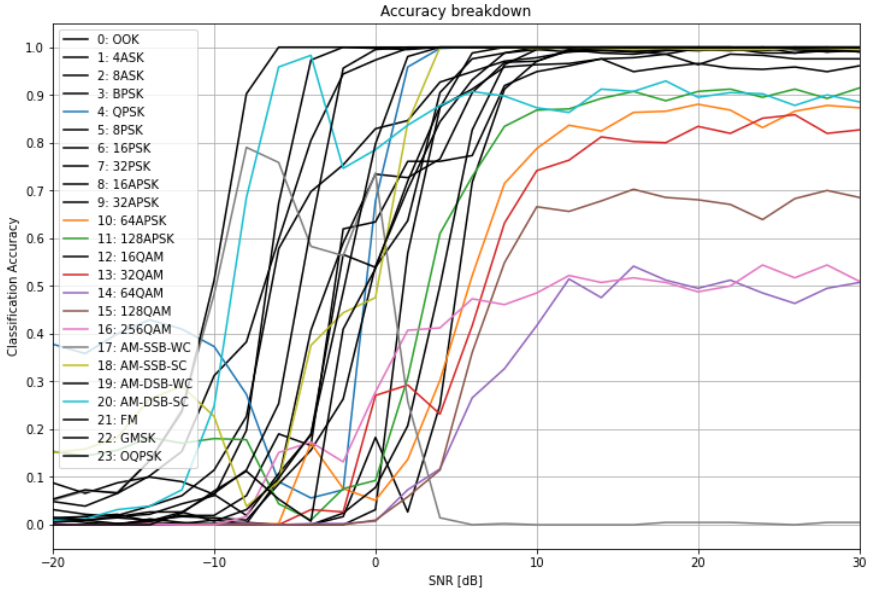




RESULTS

```
Accuracy @ highest SNR (+30 dB): 0.878963
Accuracy overall: 0.560381
```





Thank You for Your Time.

Any Questions?

Red Gecko

Redline Communications
302 Town Centre Blvd.,
Markham, ON L3R 0E8 Canada

Tel: +1.905.479.8344
Fax: +1.905.479.5331
Toll Free in North America:
+1.866.633.6669