Stephanie Lew

Week 11 - Advanced Architectures and Memory Networks

MOTIVATION FOR ADVANCED ARCHITECTURES

Rarely do we use vanilla models - Bag of Vectors, RNN, CNN, TreeRNNs - as it is. How can we combine and extend the models creatively to increase performance/accuracy?

QUASI-RNN

RNNs do not interact well with GPUs. RNNs have an inherently sequential nature that prevents full utilization of parallel hardware. This unlike LSTMs and GRUs, which are now the go-to architecture for many tasks including text classification, language modeling, and the analysis of time-series data.

- QRNN tries to achieve similar performance to LSTM with much less training time
- Consists of a convolution layer and a pooling layer.
- Convolutional layer computes intermediate representations, while pooling layer handles sequential dependencies.
- dependency between the computation of hidden states is removed
- For character language modelling, QRNNs underperformed LSTMs, whereas they performed well for word level language modeling. Character level language models require much more complex long-term interactions, and therefore being able to transform each hidden state by a whole matrix multiplication before computing the next hidden state was crucial in obtaining good performance

NEURAL ARCHITECTURE SEARCH

- Attempts to combining predictions of training
- NAS aggregates classifications together Ensemble weight
- Reweighting how important the training examples are.
    - Recall AdaNet - Boosting: build a classifier, modify a classifier after it's built
    - Adaboost changes how important the instances are

DYNAMIC MEMORY NETWORK

- Input sequences and questions, forms episodic memories, and generates relevant answers based on question fed to the module

DISCUSSION

- Why is max pooling used in Q-RNN?
- How is Q-RNN Chaining coreferencing in LSTM
- What is the role of boosting in NAS?
- How is Sentiment analysis and attention related to dynamic metwork network?