

Applied Deep Learning Assignment 2

Tung-Chun, Chiang R05922027

November 9, 2016

1 Recursive Neural Network

1.1 Abstract

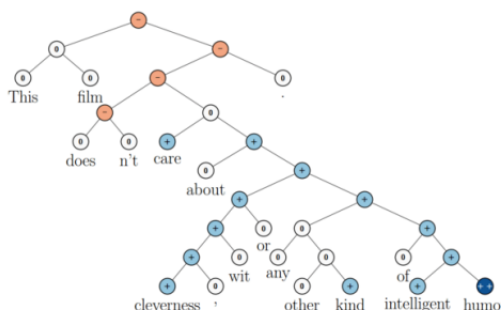


Figure 1: RvNN structure example

Recursive Neural Network is an NN in tree structure, the cost will be propagated from root to leaves(Figure 1). In this assignment, we use RvNN for sentence semantic analysis task: given tree structures of sentences, use RvNN to predict the semantic of the whole sentence(labels on roots).

1.2 Implementation challenges

In this assignment, we implement RvNN with Tensorflow. The largest challenge is: all operations we declare will be stored in the Graph, and that leads to large memory usage when we build tree structure logits. We try two methods to solve this problem:

1.2.1 Reset graph and Build logits for each tree

We build logits of a tree dynamically: in SGD, when we train a tree, we first create a new Graph(so that the memory in old Graph is free), build logits of the tree, create loss and optimizer and then run optimize operation. This method saves lots of memory, but is slow very much.

1.2.2 Pre-compute and Share the logits

To save training time, we want to pre-compute logits, loss, optimizer of all trees before doing SGD. But it costs lots of memory(because tensorflow stores all operations in Graph). To solve this problem, we share logits of the same trees(including subtrees). This methods reduces half the number of logits and memory cost from 222GB to 50GB, which is small enough to be stored in common 128GB RAM workstation.