Deep Learning A4
Language Models

Mehmet Ugurbil
Courant Institute
New York University
mu388@nyu.edu

Abstract

1 Questions

The function lstm(i, prev_c, prev_h) is related to the equations in the paper via

$$\text{i}= D(h_t^{l-1}), \text{ prev\_c}= c_{t-1}^l, \text{ prev\_h}= h_{t-1}^l$$

The function create_network() returns a single time instance (at $t = t_0$, initial time) of a recurrent neural network with long-short term memory with a linear then a softmax layer on top. Error is given by the negative log likelihood function. It is not unrolled network because we only have one time, we do not unroll it to other time steps.

The model._s keeps the whole batch and its time steps while model._ds keeps the current time of the batch, and model.start_s keeps the first time step of the batch from the last time step of the previous batch. It is reset to 0 after it is transfered to model._s.

Gradient clipping is used for gradient normalization.

Back prop through time is used for the optimization.

To deal with the extra output in the backward pass we just added a tensor of zeros.

2 ConvNet

The character level architecture uses two layers. The size of the recurrent neural network is 200. There are 50 time steps.

References

[1] LeCun, Y., Bottou, L., Genevieve B.O., & Muller K.R. (1998) *Efficient BackProp* New York: Springer.

[2] Graves, A. (2014) *Generating Sequences with Recurrent Neural Networks*, ArXiv.

[3] Zaremba, W. (2015) *Recurrent Neural Network Regularization*, ICLR.