Deep Learning — Assignment 4

Anirudhan J. Rajagopalan
Department of Computer Science
New York University
New York, NY.
ajr619@nyu.edu

1 nnGraph

1.1 1. Warmup

The code for nngraph_warmup.lua can be found at `https://git.io/vwQco`

1.1.1 2. Grucell diagram

The gru cell was drawn using the following steps.

1. Code the cell in torch similar to the code in main.lua
2. Plot the code using graph.dot function passing the filename argument
3. Open the svg file in browser and remove the unwanted nodes.

The cell diagram generated is included in 2.

2 Language Modeling

2.1 Generating sequences

The query_sentences.lua can be found at `https://git.io/vwQEc`.

The query_sentences.lua does the following

1. Loads the core network of the model.
2. Builds the vocabulary map and the inverse vocabulary map.
3. Fetches the number of words to generate and the initial seed words (minimum 2).
4. Does a forward pass on the core_network for each and every word to generate the index for next word.
5. The index is generated by using a multinomial distribution over the probabilities generated by the logsoftmax layer (layer 44 in core_network)
6. Concatenates and returns the new sentence.

Steps to run the model:

1. Change the params table in query_sentences.lua accoding to the model that will be used.
2. Change the model file path to point to the right path
3. *th query_sentences.lua*

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

2.2 Improvements to the model

2.2.1 Experiments summary

A number of experiments were preformed on the model. A few of the major areas which we explored are

1. Changing the size of rnn (200, 600). The best performing model has rnn_size of 600.
2. Enabling/changing dropout. The best performing model has a dropout of
3. Changing the sequence length. The best performing model has a sequence length of 30.
4. Changing the core network to work with GRU instead of lstm (Code can be found in `https://git.io/vwQXB`)
5. Chainging the number of layers. Increasing the number of layers consistently decreased the performance of the model.
6. Changing gradient clipping. Changing the gradient clipping doesn't appear to affect the outputs much.
7. Changing the vocabulary size. This actually has no effect as the total number of words in the corpus is only 10,000.

The best performing model has a test accuracy of **86.818**. The model characteristics are

vocab_size 12000

core_network LSTM

Seq_length 30

rnn_size 600

dropout 0.4

layers 2

2.2.2 Hardware & Runtimes

Almost all of the experiments were run in NYU HPC clusters with 20 core processors, 16GB RAM.

The default model ran ran fast with wps = 2K. There was considerable reduction in the speed of the model as the rnn_size is increased. The best performing model has a wps of around 650.

2.2.3 Model file

The model file can be found at `http://cs.nyu.edu/~ajr619/lang_model.net`

2.2.4 Experiments

LSTM

seq length	layers	rnn size	dropout	vocab size	best Perplexity
20	2	200	0	10000	119.756
30	2	200	0	10000	114.548
15	2	200	0	10000	195.712
30	4	200	0	10000	120.359
40	3	200	0	15000	137.629
40	5	200	0.2	10000	135.020
40	4	400	0.2	10000	107.970
30	2	400	0.2	10000	93.449
30	4	400	0.3	10000	102.013
30	4	400	0.5	10000	113.420
30	2	400	0.5	10000	96.340
30	2	600	0.4	12000	87.741
30	2	500	0.3	10000	89.794

GRU

seq length	layers	rnn size	dropout	vocab size	best Perplexity
20	2	200	0	10000	182.217
15	2	200	0	10000	195.712
30	2	600	0.4	10000	97.056
30	2	700	0.5	10000	101.021

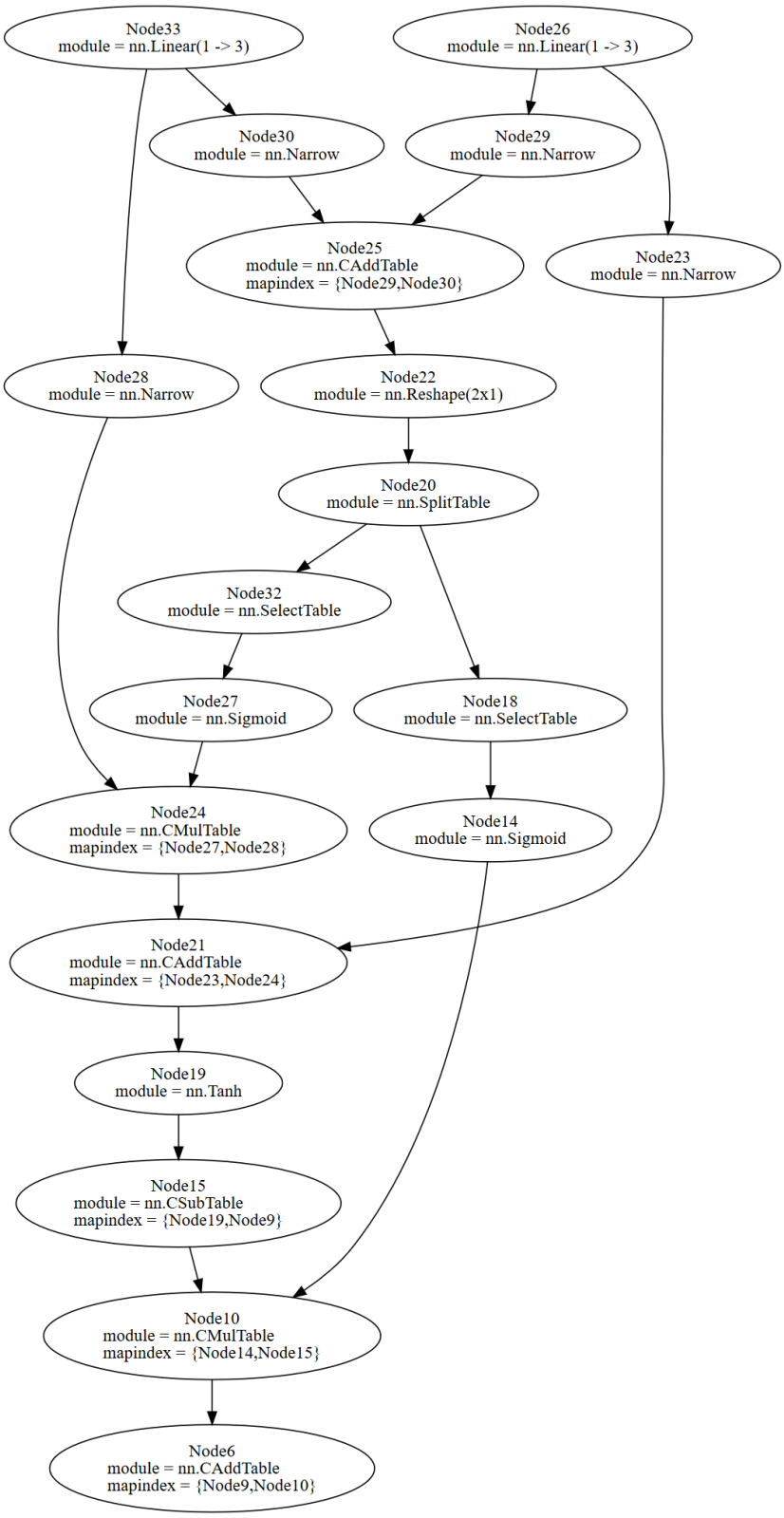

Figure 1: GRUCell given in slide 32 of talk by Armand Joulin

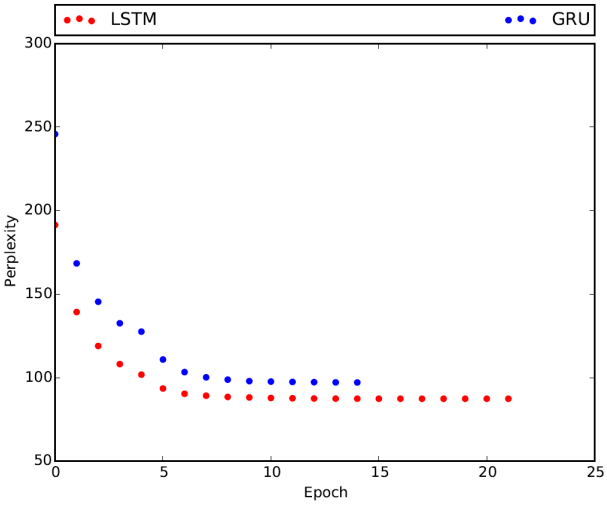

Figure 2: LSTM vs GRU sample comparison plot for their corresponding best performing model.