Deep Learning Teaching Kit
Lab 4, Sample Solution

1 nngraph

2.

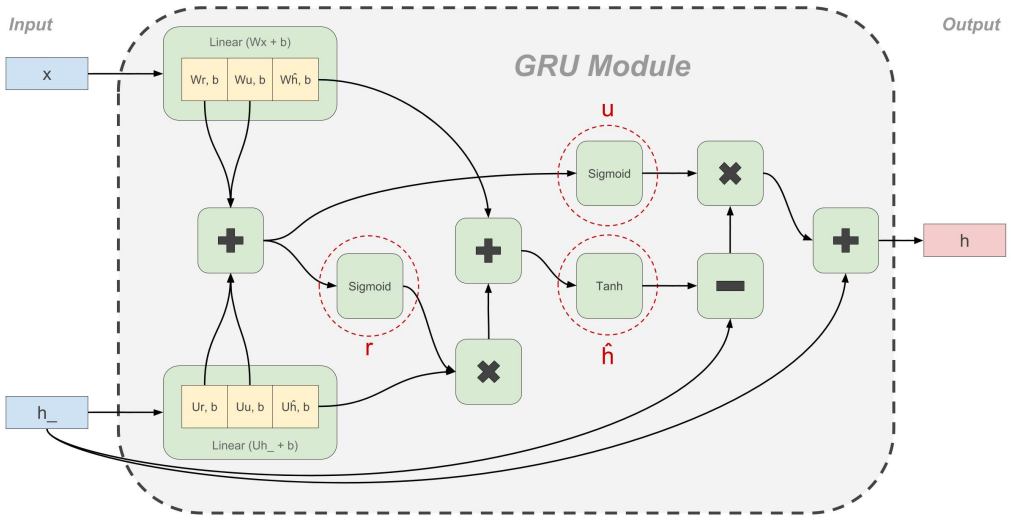

Figure 1: A diagram of a GRU module in the `nngraph` implementation.

Figure 1 shows the dataflow in the GRU cell. The GRU cell takes x_t and h_{t-1} as inputs and computes h_t as below.

$$
\begin{aligned}
r &= \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \\
u &= \sigma\left(W_u x_t + U_u h_{t-1} + b_u\right) \\
\tilde{h}_t &= tanh\left(W_{\tilde{h}} x_t + U_{\tilde{h}}\left(r \odot h_{t-1}\right) + b_{\tilde{h}}\right) \\
h_t &= (1-u) \odot h_{t-1} + u \odot \tilde{h}_t
\end{aligned}
$$

In this implementation, parameters $[W_r; W_u; W_{\tilde{h}}]$ and $[U_r; U_u; U_{\tilde{h}}]$ are stacked together to reduce the number of dot product operation. x_t and h_{t-1} are transformed in linear modules. For u and r gates, the outputs from the linear modules are summed up and performed element-wise sigmoid operation. For $\tilde{h}_t$, element-wise multiplication $r \odot h_{t-1}$ is first performed, and it is summed with $W_{\tilde{h}} x_t$. Then, element-wise $tanh$ operation is performed. Combining all outcomes so far, h_t is computed as the equation shows.

2 Language Modeling

Architecture

In this experiment, I have tested two different recurrent language model architectures. One has LSTM as recurrent units, and another one has GRU as recurrent units. Both LSTM and GRU models have 2 layers. The number of hidden units is 650 for both models. The model configurations are same as an experiment in Woijciech et al. (2015).

Learning Techniques

Dropout is a widely used regularization method in the neural network community. As shown in Woijciech et al. (2015), dropout is very effective especially if it is used to the non-recurrent connections. In this experiment, 50 % dropout is applied on each of inputs, outputs from the first layer, and outputs from the second layer. See Figures 1 reprinted from the original paper.

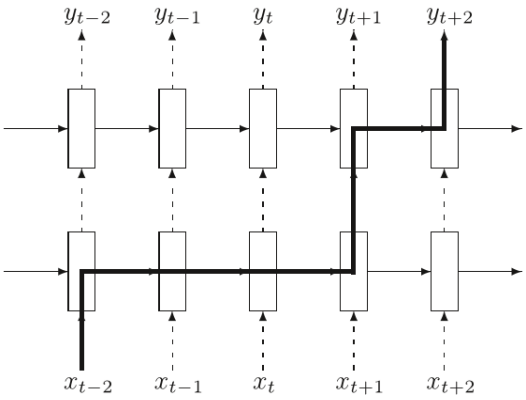

Figure 2: An information flow in LSTM. The dashed lines indicate the connections affected by dropout.

Learning Procedure

First, model parameters are initialized using uniform random numbers between -0.05 and 0.05. The optimization method is mini-batch stochastic gradient descent (SGD). The batch size is 20. For the LSTM model, a learning rate is 1 in the first six epochs. For the GRU model, a learning rate is 1 in the first four epochs. After then, the learning rate is shrunk by dividing by 1.2 after every epoch. The gradient is rescaled (clipped) if the gradient norm exceeds 5.

The Penn Tree Bank (PTB) dataset preprocessed by Mikolov is used in this experiment. The vocabulary size is $10,000$. Most frequent 10,000 words are in the dataset, and less frequent words are replaced by `<unk>` token. The same split as the original data is used for the training, validation, and test sets. Those datasets contains $42,068$, $3,370$, and $3,761$ sentences respectively.

The objective function is the cross entropy shown as below.

$$L = -\frac{1}{|B|}\sum_{i=1}^{|B|}\sum_{j=1}^{|T|} y_j^i \log p_{w_j|w_{<j}},$$

where $|B|$ is the number of samples in a batch, and $|T|$ is the number of time steps or sentence length. In this experiment, both are fixed 20. During the training phase, parameters are chosen as they minimize the cost of the objective function. Each $p_{w_j|w_{<j}}$ is an element of the output vector from the softmax layer, that normalize number into the range between 0 and 1. Minimizing this objective function is equivalent to assigning higher probability to the ground truth.

The evaluation metric is a perplexity. The below equation is implemented in the code used in this experiment.

$$PPL = \exp^{-\frac{1}{N}\sum_{n=1}^{N} \log p_{w_n|w_{<n}}},$$

where N is the number of words in the validation or test set. The validation is computed after every epoch. The test perplexity is computed at the end of training.

Both the LSTM and GRU models are trained for 50 epochs. Training one network takes approximately 25 hours on my 15 inch MacBook Pro.

Results

Under the almost same configurations, the LSTM model achieved a lower perplexity than the GRU model. Both models seem converged at around the 30th epoch although the validation perplexities are still slowly decreasing after the 30th epoch. The LSTM model's performance matches what reported in Woijciech et al. (2015). At around the 5th epoch, the GRU model's gradient starts exploding; thus, the learning rate decay schedule is adjusted to avoid this happens. Though the GRU model's performance is not as strong as the LSTM model, I believe that more hyperparameter tuning is required to utilize the GRU's potential.

Model	Validation	Test
LSTM	86.52	82.52
GRU	96.43	93.15

Table 1: Perplexity on Penn Tree Bank dataset.

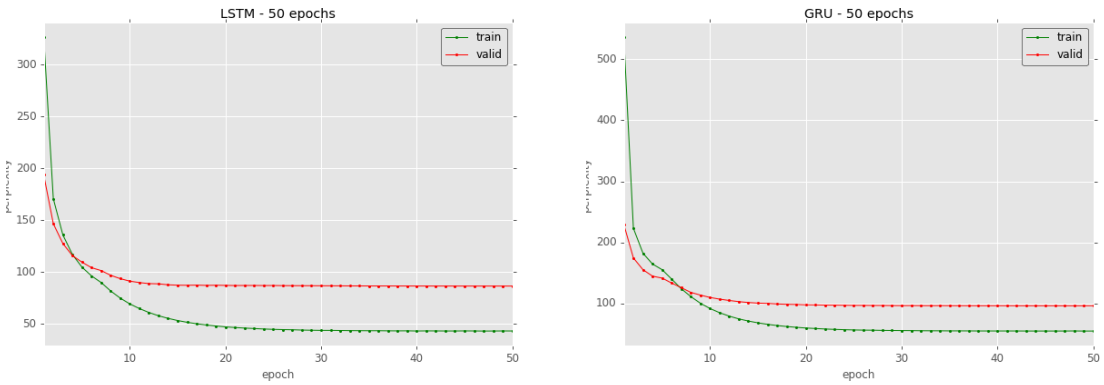

Figure 3: Training and Validation Perplexities: LSTM (left), GRU (right)

References

[1] Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Recurrent neural network regularization.