EX2 - RNN

Nimrod Curtis, 311230924, nimrodcurtis@mail.tau.ac.il
Alon Mizrahi, 312284706, alonmizrahi2@mail.tau.ac.il

Spring 2023

1 Theory

1.1 Q1

(a) **Trimming -** Sequence truncation involves shortening each input to a specified length. This can be performed either at the beginning or the end of the sequence.

Padding - Padding a sequence involves adding zeros or other special tokens to the inputs to ensure that all sequences have the same length before they are fed into a deep learning network.

(b) **Beam Search with Length Penalty -** One common approach is to use beam search with a length penalty. Beam search generates multiple possible output sequences and selects the most likely one based on a score. In this case, a length penalty is added to the score to encourage shorter output sequences. This helps to prevent the model from generating overly long outputs while still allowing for some variability in the length of the output.

Conditional Generation - Another approach is to use conditional generation, where the model is trained to predict the length of the output sequence as well as the content. This can be done using a two-stage approach, where the model first predicts the length of the output sequence, and then generates the content of the sequence based on that length.

1.2 Q2

Two advantages of GRU over LSTM are:

1. GRU has a simpler architecture than LSTM, making it easier and faster to train.
2. GRU is better at handling short-term dependencies due to its reset gate.

1.3 Q3

Based on the question description, we can infer that the current state and input have a size of 200, resulting in weight matrices of size 200x200. For each gate, there are two weight matrices and a bias vector, making the total number of parameters to be calculated as follow:
$4 * (200 + 2 * 200x200) = 320,800$

1.4 Q4

To calculate the gradients of GRU for backpropagation at the second time stamp, we can use the chain rule to derive the expressions for the gradients with respect to the loss function. (given to us - $\frac{\partial \epsilon_{(2)}}{\partial h_{(2)}}$)

(a) $\frac{\partial \epsilon_{(2)}}{\partial W_{xz}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial z_{(2)}} \cdot \frac{\partial z_{(2)}}{\partial W_{xz}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (h_{(t-1)} - g_{(t)}) \cdot \sigma_{z_2} \cdot (1 - \sigma_{z_2}) \cdot x_{(2)}$

(b) $\frac{\partial \epsilon_{(2)}}{\partial W_{hz}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial z_{(2)}} \cdot \frac{\partial z_{(2)}}{\partial W_{hz}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (h_{(t-1)} - g_{(t)}) \cdot \sigma_{z_2} \cdot (1 - \sigma_{z_2}) \cdot h_{(1)}$

(c) $\frac{\partial \epsilon_{(2)}}{\partial W_{xg}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial g_{(2)}} \cdot \frac{\partial g_{(2)}}{\partial W_{xg}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (1 - z_{(2)}) \cdot (1 - tanh^2_{g_t}) \cdot x_{(2)}$

(d) $\frac{\partial \epsilon_{(2)}}{\partial W_{hg}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial g_{(2)}} \cdot \frac{\partial g_{(2)}}{\partial W_{hg}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (1 - z_{(2)}) \cdot (1 - tanh^2_{g_t}) \cdot r_{(2)} \cdot h_{(1)}$

(e) $\frac{\partial \epsilon_{(2)}}{\partial W_{xr}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial g_{(2)}} \cdot \frac{\partial g_{(2)}}{\partial r_2} \cdot \frac{\partial r_{(2)}}{\partial W_{xr}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (1 - z_{(2)}) \cdot (1 - tanh^2_{g_t}) \cdot (W_{hg} \cdot h_{(1)})(\sigma_{r_2} \cdot (1 - \sigma_{r_2}) \cdot x_{(2)})$

(f) $\frac{\partial \epsilon_{(2)}}{\partial W_{hr}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot \frac{\partial h_{(2)}}{\partial g_{(2)}} \cdot \frac{\partial g_{(2)}}{\partial r_2} \cdot \frac{\partial r_{(2)}}{\partial W_{hr}} = \frac{\partial \epsilon_{(2)}}{\partial h_{(2)}} \cdot (1 - z_{(2)}) \cdot (1 - tanh^2_{g_t}) \cdot (W_{hg} \cdot h_{(1)})(\sigma_{r_2} \cdot (1 - \sigma_{r_2}) \cdot h_{(1)})$

2 Practical

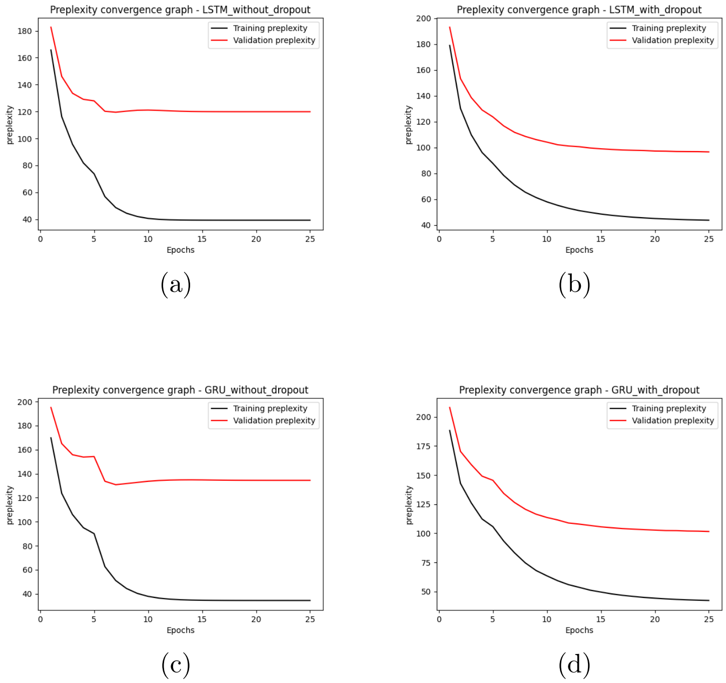
In this section, we conducted multiple training sessions of the Regularized $LSTM$ neural network by Zaremba et al. (2014).
We used the following hyper parameters:

1. hidden layers number = 2
2. hidden size = 200
3. dropout = 0.25 (when in used)
4. batch size = 20
5. seq length = 20
6. learning rate = 1 (in LSTM) / 0.8 (in GRU)
7. total epochs = 25
8. factor (dividing the lr param) = 1.8 (with dropout) / 1.2 (without dropout)

Throughout all the experiments, we used a GD optimizer and the Perplexity as the loss function.
Our evaluation include a comparison of:

1. LSTM without dropout
2. LSTM with dropout
3. GRU without dropout
4. GRU with dropout

The training outcomes are visualized through convergence graphs depicting the model losses, and a final loss table.



(a)

(b)

(c)

(d)

Figure 1: (a) LSTM without dropout, (b) LSTM with dropout (c) GRU without dropout (d) GRU with dropout.

Type	Train	Validation	Test
LSTM without dropout	39.25	119.84	115.73
LSTM with dropout	43.80	96.59	92.67
GRU without dropout	34.400	134.39	129.95
GRU with dropout	42.221	101.53	97.45

Table 1: Convergence results

Conclusions:

- It was discovered that the regularized configurations of LSTM and GRU yielded better results than the non-regularized ones. However, when dropout was applied, the convergence was slower.
- Based on the given parameters, it was observed that the LSTM outperformed the GRU in terms of loss. Furthermore, it was necessary to adjust the learning rate of the GRU to enhance its performance. The reason for this can be that GRU architecture has less trainable parameters.