LM Report (NLU Assignment 1)

Silvano Vento Maddonni (247370)

University of Trento

silvano.maddonni@studenti.unitn.it

1. Introduction

In this project, we start from a baseline LM-RNN and we modify it by adding a set of techniques to improve the performance. For each experiment, we evaluate the performance using the Perplexity (PPL), trying to minimize it. The work is composed of two parts. In part one, we start by replacing the RNN with a Long-Short Term Memory (LSTM) network. Then we add two dropout layers: one after the embedding layer, one before the last linear layer. Finally, we replace SGD with AdamW. In the second part, we apply some regularization techniques and check the results with each of them. In order: Weight Tying, Variational Dropout and Non-monotonically Triggered AvSGD.

2. Implementation details

Part 1

1.1 - LSTM

Replacing the RNN with the LSTM is pretty straightforward: we keep the same architecture and structure and swap the RNN with the LSTM. This results in an improvement, as LSTMs are able to handle long-term dependencies much better than RNNs.

1.2 - Dropout Layers

Two dropout layers are added: one after the embedding layer and one before the last linear layer. This has a big impact, and it is reflected in the PPL. Dropout is a very common technique that is almost always implemented. Different dropout probabilities have been tried, but we stick with 0.1.

1.3 - AdamW

On top of adding the dropout layers, a different optimizer is also tried. AdamW replaces SGD. AdamW features Adaptive Learning Rates and Weight Decay Regularization, and it is preferred in the LM domain because it's faster to converge, more robust, and it's able to handle better sparse gradients.

Part 2

2.1 - Weight Tying

Weight tying is used to reduce model size and improve generalization. Here, we set `output.weight = embedding.weight` in the initialization of the model. This sharing of the weights also requires that the weights matrixes of these two layers must be of the same size. Weight tying can improve performance, especially in LM where there can be many logical connections tied together.

2.2 - Variational Dropout

The key idea behind variational dropout is to introduce a random variable for each weight in the network, which is then sampled from a Bernoulli distribution during each forward pass. This sampling process introduces noise into the network, which helps prevent overfitting and encourages robustness.

To implement this, a VariationalDropout class is created and used in place of the two precedent dropout layers. Inside it, a binary mask of the same shape as x (the input tensor) is created, where each element of the mask is randomly sampled from a Bernoulli distribution with a probability of (1 - *dropout_prob*) of being 1 and *dropout_prob* of being 0.

The mask is then scaled by dividing it by (1 - *dropout_prob*). This scaling is necessary to maintain the expected value of the output during training, accounting for the dropped elements, ensuring that the expected value of the tensor remains the same regardless of whether dropout is applied or not.

The dropout probability is set as 0.3 for both the layers.

2.3 - Non-monotonically Triggered AvSGD

Non-monotonically Triggered AvSGD is an extension of SGD that incorporates averaging over multiple updates to stabilize convergence and improve generalization. To implement this technique, we defined two optimizers instead of one: SGD and ASGD and a switch is created in the train function. It will start with a regular SGD (*lr*=1.5) and switch to ASGD (*lr*=1.5, *t0*=0, *lambd*=0, *weight_decay*=1.2e-6) when the performance of the model stagnates (or worsen). The number of steps to wait to see if the model was not improving anymore before switching the optimizer is set to 5.

3. Results

In each section of Part 1 and Part 2 different modifications were added and tested. Although it could have happened that not all of them provided improvements, that was not the case. We got enhanced performance by gradually adding each of them. We present now the results, with respect to the evaluation metric used: the Perplexity (PPL). Each consecutive part was build on top of the previous.

Table 1: *Summary of Experimental Results.*

	Model	**PPL**
–	Base RNN (SGD, no dropouts)	163.3
1.1	LSTM (SGD, no dropouts)	143.5
1.2	LSTM + dropout layers	122.1
1.3	AdamW	120.0
2.1	Weight Tying with model from 1.1	122.9
2.1	Weight Tying with model from 1.3	108.9
2.2	Variational Dropout	101.8
2.3	Non-monotonically trig AvSGD	99.8