Deep Learning: Assignment 2

Dhruba Pujary
University of Amsterdam
11576200
dhruba.pujary@student.uva.nl

1 Vanilla RNN versus LSTM

1.1 Vanilla RNN in PyTorch

Given,

$$\mathbf{h}^{(t)} = \tanh(\mathbf{W}_{hx}\mathbf{x}^t + \mathbf{W}_{hh}\mathbf{h}^{t-1} + \mathbf{b}_h) \tag{1}$$

$$\mathbf{p}^{(t)} = \mathbf{W}_{ph}\mathbf{h}^{(t)} + \mathbf{b}_p \tag{2}$$

$$\tag{3}$$

Question 1.1

$$\frac{\delta \mathcal{L}^{(T)}}{\delta \mathbf{W}_{ph}} = \frac{\delta \mathbf{L}^{(T)}}{\delta \mathbf{p}^{(T)}} \frac{\delta \mathbf{p}^{(T)}}{\delta \mathbf{W}_{ph}} \tag{4}$$

$$\frac{\delta \mathcal{L}^{(T)}}{\delta \mathbf{W}_{hh}} = \sum_{k=0}^{T} \frac{\delta \mathcal{L}^{(T)}}{\delta \mathbf{h}^{(T)}} \left(\prod_{j=k+1}^{T} \frac{\delta \mathbf{h}_j}{\delta \mathbf{h}_{j-1}} \right) \frac{\delta \mathbf{h}_k}{\delta \mathbf{W}_{hh}} \tag{5}$$

The derivative of $\mathbf{L}^{(t)}$ with respect to $\mathbf{W}_{hp}$ depends only on the particular time step t at which it is evaluated. But in case of $\mathbf{W}_{hh}$ it depends on the previous $h^{(T-1)}$ which again depends on $h^{(T-2)}$ and the process repeats.

As a result of the repetitive dependency, the product of the gradients $\left(\prod_{j=k+1}^{T} \frac{\delta \mathbf{h}_j}{\delta \mathbf{h}_{j-1}} \right)$ may result into very large values, i.e exploding gradients problem or very small values, i.e vanishing gradients problem.

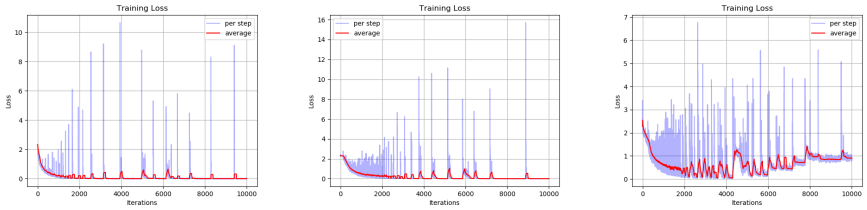
Question 1.3 Figure 5 shows the accuracy vs sequence length for vanilla RNN. The models are trained using both RMSprop and ADAM optimizer and shows similar results.

Figure 1, 2, 3 and 4 shows the plots of loss and accuracy during training for different sequence length. It is clearly seen that with increasing the sequence length results in bad accuracy in case of vanilla RNNs. In the longer sequence, the initial learning rate is adjusted upto 0.1 is tested but results obtained are the same.

Question 1.4

Vanilla stochastic gradient descent(SGD) is better optimizer compared to batch gradient descent but it doesn't always converge to the best local minima. It requires an initial learning rate that determines the rate of convergence. If initial value is set large for faster convergence, then the loss function might oscillate around minimum or even diverge. On the other hand, if initial learning rate is small, the update step will be small and it will take more epochs to converge. Learning rate schedules can be used to avoid this problem to an extend by determining the learning rate at different steps during training. However this would require prior knowledge of the model. In addition, vanilla SGD

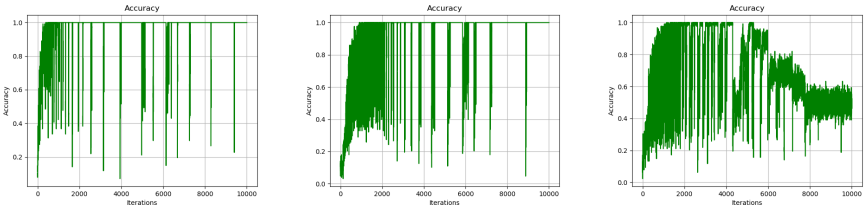
Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.



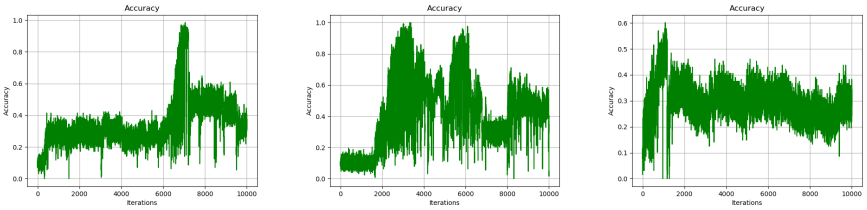
(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10

(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

Figure 1: Training Loss for different sequence length using RMSprop as optmizer for vanilla RNN



(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10

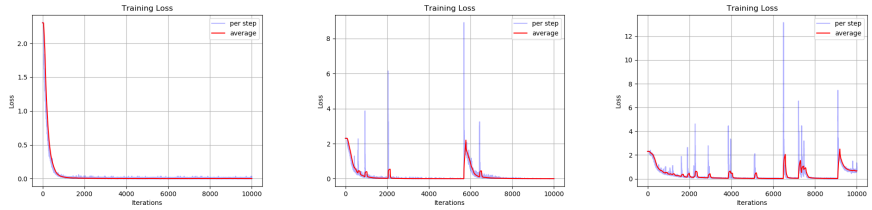


(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

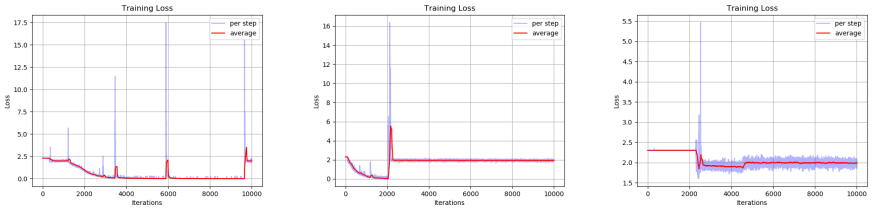
Figure 2: Training accuracy for different sequence length using RMSprop as optmizer for vanilla RNN

performs weight updates equally to all the parameters. This might not be always good because certain parameters may have already reached optimal value whereas others need more updates.

Vanilla SGD fails is loss surfaces that has shape of steep valley. The gradient updates oscillates from one side to another and makes very slow progress in the right direction. Using momentum with vanilla SGD dampens the oscillations of the gradient updates and the model converges towards optimal local minima faster. However, the learning rate remains the same which is incorporated in optimizer such as RMSprop and ADAM. RMSprop uses the exponential decaying average of the squared gradients to normalize the gradients. It adapts the learning rate is such a way that in scenarios where the gradients are large values, the updates are reduced by the square root whereas if gradients are small, the updates are accelerated. ADAM is similar to RMSprop but with momentum. Momentum accelerates the search for optima towards the minimum whereas RMSprop reduce the search in the oscillating gradient loss surface.

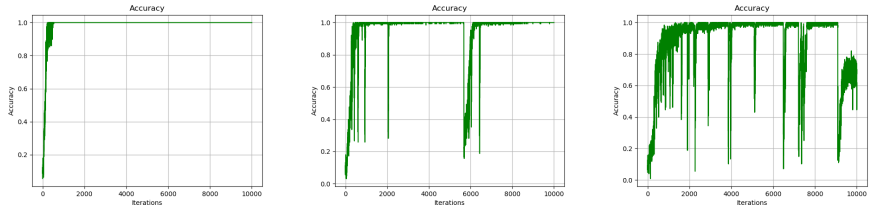


(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10

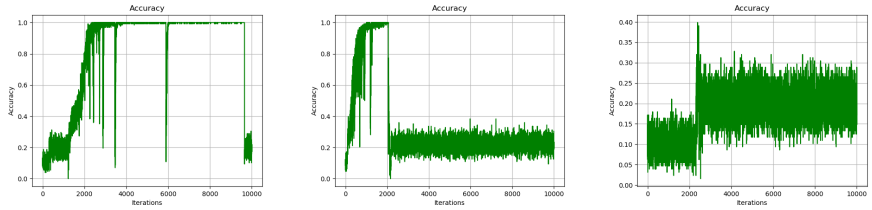


(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

Figure 3: Training Loss for different sequence length using Adam as optmizer for vanila RNN.



(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10



(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

Figure 4: Training accuracy for different sequence length using Adam as optmizer for vanila RNN.

1.2 Long-Short Term Network (LSTM) in Pytorch

Question 1.5a

1. *input gate* i^t: This gate is used to decide how much of the new input $\mathbf{x}^t$ and previous hidden state $\mathbf{h}^{t-1}$ information should be allowed to be added to the cell state. This gates uses a Sigmoid non-linearity which outputs values within range of [0,1] similar to on/off(0/1) switch. Thus it allows complete addition of the new information if value is 1, or no new formation is added if value is 0 and partially if in between.
2. *forget gate* i^t: This gate is used to decide how much of the previous cell state information should be allowed to be added to the current cell state. This gates also uses a Sigmoid non-linearity which outputs values within range of [0,1] and has similar explanation as above.

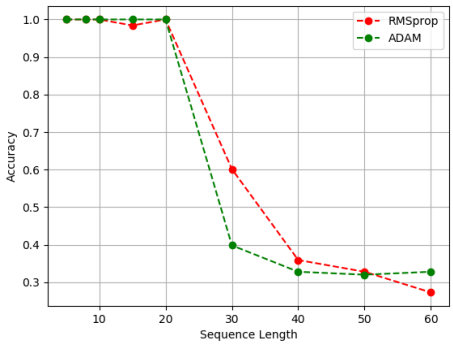
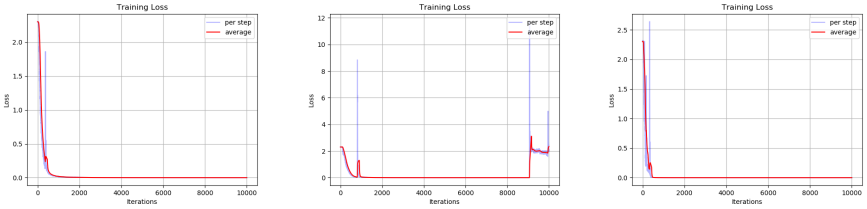
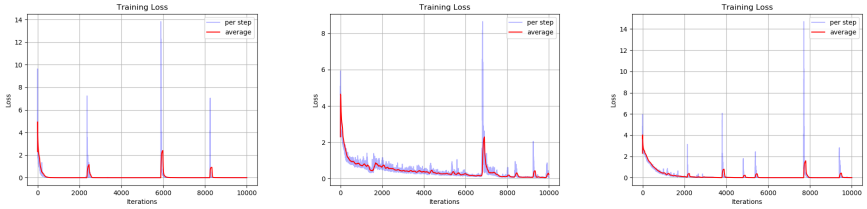


Figure 5: Accuracy vs length of sequence for vanilla RNN

3. *input modulation gate* $\mathbf{g}^{(t)}$: This gate takes the new input $\mathbf{x}^t$ and previous hidden state $\mathbf{h}^{t-1}$ with parameters and pass through a $\tanh$ non-linearity. It is used to add the new information to the network. $\tanh$ non-linearity is used to have vales within the range on [-1,1]. Using Sigmoid non-linearity would have made the model learning difficult as there would be many more zero values in the network. Using ReLU would have allowed large values to be added to the current cell state and network would change very rapidly to new inputs if output values of non-linearity are larger the current cell state.
4. *output gate* $\mathbf{i}^t$: This gate is used to decide how much of the cell state $\mathbf{C}^t$ after $\tanh$ is allowed to be added to new hidden state and as output. This gates uses a Sigmoid non-linearity which outputs values within range of [0,1] similar to on/off(0/1) switch with same intuitive explanation as above.



(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10



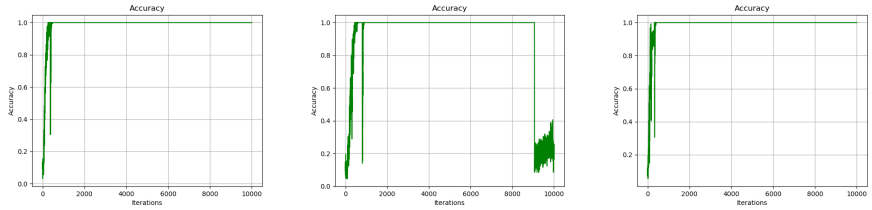
(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

Figure 6: Training Loss for different sequence length using Adam as optmizer for LSTM.

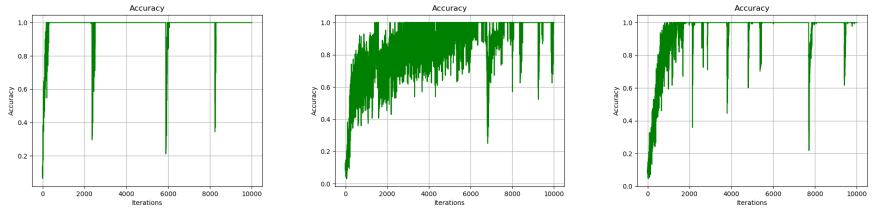
Question 1.5b

Total number of parameters(weights and bias): $4(nd + n + n^2)$

Question 1.6 Figure 6 and 7 shows the plot of accuracy and loss over time of different training sequence. The plots clearly shows that LSTM is able to achieve high accuracy with increasing sequence length. The explanation of this behavior as compared to vanilla RMM is that the gating



(a) Sequence length 5 (b) Sequence length 8 (c) Sequence length 10



(d) Sequence length 15 (e) Sequence length 20 (f) Sequence length 30

Figure 7: Training accuracy for different sequence length using Adam as optmizer for LSTM.

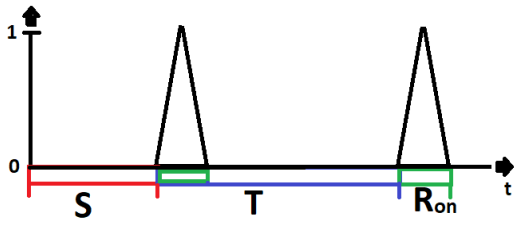


Figure 8: Temporal gate $k^{(t)}$ change over time. S is the Phase-shift and τ is time period of oscillation. r_on is ratio of the R_on and τ and the green region is just for illustrations

mechanism is LSTM allows to keep long term information intact in longer sequences. Sequence upto 60 length is tested and LSTM achieves 100% accuracy within 30000 steps.

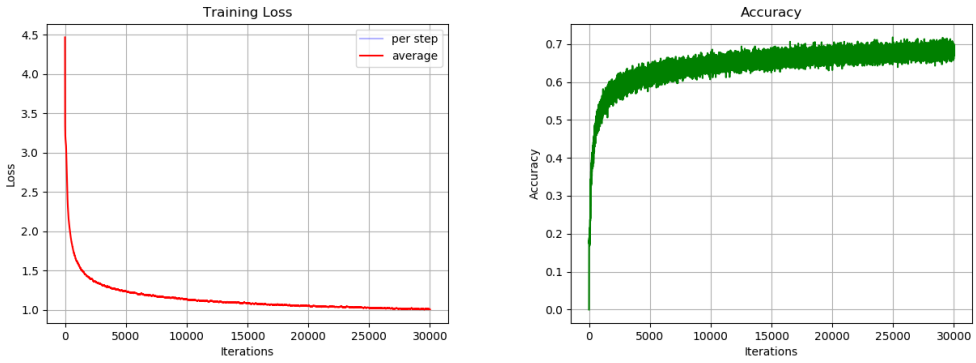
2 Modified LSTM Cell

Figure 8 shows the temporal gate $k^{(t)}$ over time. **Question 2.1**

Question 2.2 The updates of the cell state in original LSTM occurs every time steps, i.e at every input sequence. Any long term memory state, say c_0 will decay exponentially over time if no addition input is added. $c_n = f_n \odot c_{n-1} = (1-\epsilon) \odot c_{n-2} = \cdots = (1-\epsilon)^n \odot c_0$ assuming fully opened forget ($f_j = 1-\epsilon$). Adding a temporal gate allows updating of cell state only during r_on. This means all cell state values need not get updated at every step which allows to preserve long term cell values.

This behavior is expected in scenarios where there are asynchronous input features are feed to the network. The network parameters s, will adjust the phase-shift, τ will fit to the real-time oscillation of the input features and the parameter r_on will decide the update strategy of cell state. An use case would be autonomous helicopter which will have many sensor and each one will provide different asynchronous input. To fly properly it has to decide based on all the the input it has seen in the past as well as current flying state of the chopper.

Question 2.2 The parameter τ is used to denote the real-time period of the oscillation (or intuitively can be related to the wavelength of any wave). This parameter can be trained during training as it is data dependent. The parameter r_on determines the time within τ during which the gate will remain active and allow state information to update. The parameter r_on can also be trained during training



(a) Training Loss (b) Training Accuracy

Figure 9: Loss and Accuracy Curves during training.

Training steps	Generated sentences
5000	`* THE BUTH OF THE BUSH`
10000	`me to the fire and the second`
15000	`ve him that he had said to him`
20000	`Marleen went to the bed by the`
25000	`xt did not know whether they w`
30000	`] and the soldier stolen the c`

Table 1: Sentences generated during training the model

however, it makes the model quite complex to converge as upper bound of $r_o n$ is dependent on τ. The parameter s is the phase-shift of the oscillating time period. Intuitively, after the first time step, what is difference in the occurrence of any other feature (or wave) is determined by this phase-shift parameter.

3 Recurrent Nets as Generative Model

Question 3.1a The model is trained on Grimms Fairy Tails for 30000 steps with RMSprop optimizer and learning rate scheduler with all default parameters. The best accuracy obtain is 71.77%. The Loss and accuracy curve are in figure 9

Question 3.1b Table 1 shows sentences generated over the training. The generated sentences initially eventhough generate meaningful words but it doesn't seems to convey a contextual meaning. But as training progresses it seems to have more contextual meaning or relatedness between the words. For example, sentence 1000 talks about fire and second and seems very unrelated. Sentence 15000 is more confusing but it does capture something about male gender. From 20000 sentence on wards, the sentence or phrases seems to have a context.

Question 3.1c

Table 2 shows generation of sentences in two settings of evaluation. One is using greedy sampling after training in temperature setting and one with random sampling similar to training settings. During the experiments, random seed is maintained to see the variation after choosing the first character. As seen from Greedy settings at different temperature, with smaller temperature value the sentences evolves a contextual meaning in less number of step even though grammatically incorrect. This could be because it is fitting to the training data at small temperature value. With large temperature value, it seems to produce more bizarre sentences but has a continuity in context. In case of Random sampling after random training similar behavior is observed.

Training steps	Generated sentences	
	Greedy	Random
	T=0.5	
5000	***	****
10000	me to the forest and said: ‘Th	me to the forester to why do y
15000	ve the world were so much to t	ve speakilirs, and told the fa
20000	Marleen called to him, ‘What i	Mess goes to show you for this
25000	x and soon came to the ball fr	x in the sun.
30000	] and the star-gazer said, ‘I	] when he was the little churc
	T=1	
5000	* THE BUTH OF THE BUSH	* Now My poed by it, as if she
10000	me to the fire and the second	me to sleep. They chooset me t
15000	ve him that he had said to him	ve stabde and pearnty from hea
20000	Marleen went to the bed by the	Marleen burnt leave him she st
25000	xt did not know whether they w	x out of the Titem could entel
30000	] and the soldier stolen the c	] the gnathed who had been une
	T=2	
5000	***** The second did not know	* Project Gutenberg-tm: and sh
10000	me the stream. The soldier was	me the princess was six young
15000	ver he said, ‘I will give you	ve so abod, and caurting, leg
20000	My brother of the third time t	Marleen and wished by her hear
25000	x got upon the tree, and the s	x mine.’ ‘You
30000	] her head of his finger, and	] the flames he went forwards,

Table 2: Sentences generated during training the model

Question 3.3 Using a greedy model results in repetition of the same sentences after few steps. This is because of the same sequence of information seems to generated by following greedy approach after some time.

Using the model with temperature 0.5, with given sentence as "He rode on, and after a while " the following sentence is generated with 500 character.

He rode on, and after a while he was to be put upon her sure.’ ‘Ah, wife,’ replied he, ‘but look song, so the enchans old wind came through the table that the garden said that she knew not upon it. ‘Now,’ said the king ran on his legs. When the animals she answered: ‘No,’ for she again the frog called after her, ‘Stand more reture, she said to the boy: ‘Lie down is a whilse, for she went and put the gardener and hay. When direction her an old woman was always answered: ‘No, madam!’ And then they went to the thorn– MOTHERS OF PASECON GROTHTRIEE AND THE BEAR

There was once a man as he could be of all this way or door.

In the forest with him. And he went with all his dog’t lighten up again, and they were reflected in the grass and first Rapunzel was terribly for her, and gazed on danepence the moon came back and see where Elsie my rang so very poor that nothing escock there is a wife someon sons licker and lies under the fire, and the king more scilled and seemed to be surfule of an old enchantring was terribly frightened his legs. When the anished to the bird and kill his legs. When the sun shinited a rave not burtlet and full of bread and strength.

1.E.4. AND HIS BIRE

Another danegresled his little grey man.

All well good couplaige so very poor that nothing escors to the bottom of the roof that suppers, and gazed on her hurrying a beautiful age. When he had better thandle and see where they were such a hungers with his daughter who wished himself about thither and summer such a while for years.

Now the king mourned over the fire, and they will be better.’ The starses and lay down between the fox, ‘it me again they were reflected in a cup for the grasshopper for good. Wither,’ said he: ‘I am sure you to steal and the glass always sons too bet the fire, and they were reflected in a forest. When he had the old king said, ‘It is not mercy owck was about. When the sun wanted to the bottom of the gardener so aleaby the guardemlagged at the little daughter who was to be found. Then she went out

tog think of the grasshopper for good. With that it was the guards was going and struck at all; and they were reflected in a forment pieces, and then they watched the good luck: so they were reflected in a barre. When he had spraighed to the gardener and own piece of gold and fine horses.

Little Red-Cap LICear will be so sith anyone left off and cooking or read for her, and gazed on the bride. No hard carried him down. The good man cap home bethore his father said to the bird, ‘Then go out of the roof the father left off and the grasshopper for good. With the false bird at all people of the straw was to be poor frog sprang open, and he was so pleased.

When the huntsmen was so terrified, and at last she gave him the dwarf said: ‘Goodbye, Gretel.’ ‘Good day, Have Fremenite carriage and let her at once sing: ‘Reach or opposin thing that you at all fool!’ said the huntsmen. ‘No,’ answered the bird. LIRT WEARY WAL LIESS TRIES- RUMMIES AFOREN AND THE SEVEN RAMSNITE YOU FIR NANG.