

# Intro to Deep Learning Homework 5 & 6

## Written Questions: 1, 2.7, 3, 5

Nathan Inkawich

[Duke Community Standard \(http://integrity.duke.edu/standard.html\)](http://integrity.duke.edu/standard.html): By typing your name below, you are certifying that you have adhered to the Duke Community Standard in completing this assignment.

Name: Nathan Inkawich

## Problem 1: Mathematical Analysis of RNNs (25 points)

(A)

For the hidden state update equation with no non-linearity, the problem is similar to the one of compounding interest. If the norm of  $W$  is greater than 1, repeated multiplication by  $W$  may lead to explosion, or exponential increase in the output value, exactly the same as compounding interest. In [1], Goodfellow also describes the explosion problem with eigenvalues of  $W$ . Here,  $W$  is decomposed to  $Q\Lambda Q^T$  so the hidden state update becomes

$$h_t = Q^T \Lambda^t Q h_{t-1} + A x_t$$

where we see the eigen values of  $W$  are raised to the power of  $t$ . Thus, eigenvalues greater than 1 may cause explosion of  $h_t$ . With this, we can say that a general requirement for  $W$  for a stable sequence with no nonlinearity is that the absolute value of the largest eigenvalue be near 1. [2] also mentions keeping the L1/L2 norm of  $W$  near 1 to avoid explosion of values.

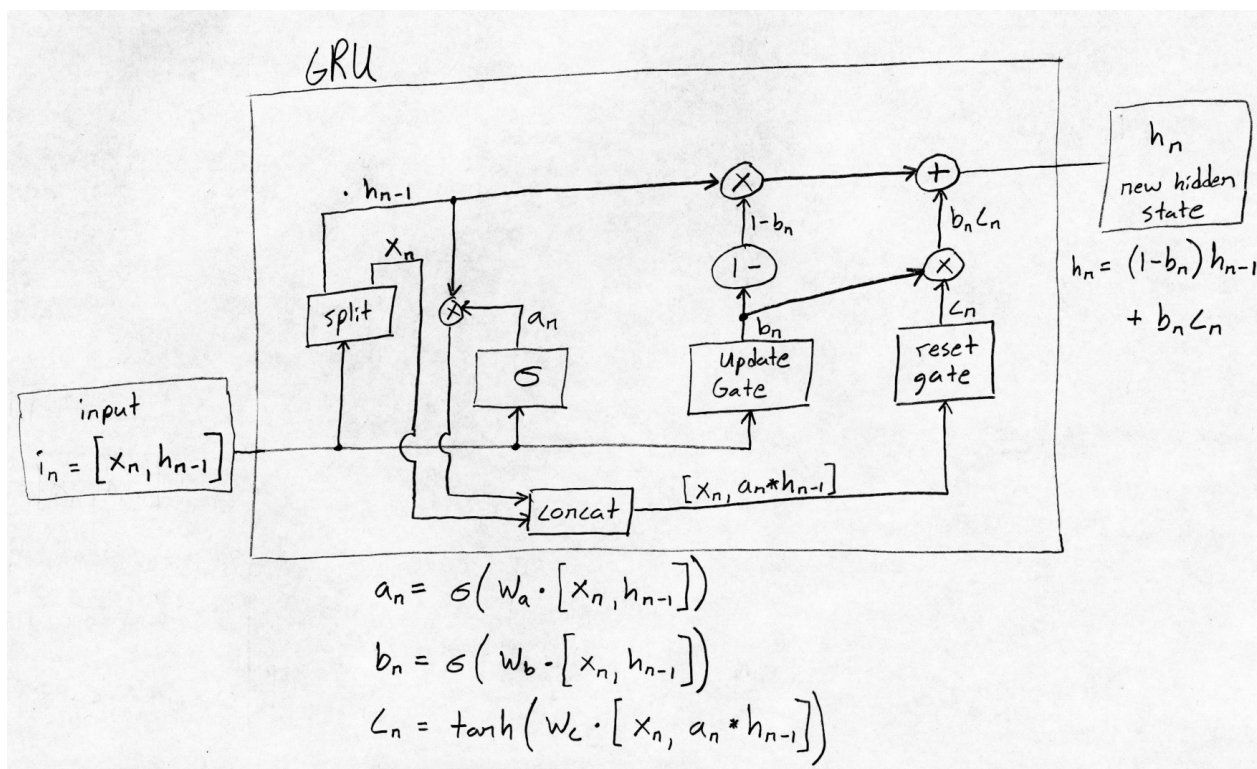
(B)

When using a nonlinearity in the hidden state update equation, it is common to experience vanishing gradients. This is due to the nature and shape of the nonlinearity. For the case of the tanh function, the largest gradient value is 1 and occurs in the unstable region at input 0. This means almost all gradient values when backpropagating through tanh are less than one. Thus, repeated multiplication by a value less than 1 leads to exponential decrease in the value. In RNNs, if we are backpropagating through several layers of the unfolded graph with tanh activations, the gradients are continually multiplied by values less than 1 and vanish quickly. Also, in the regions of saturation for tanh, the gradient values approach 0, meaning in the backward pass the gradients for a saturated neuron would be multiplied by a very small value, not helping the vanishing gradients problem.

This problem is not solved with the sigmoid non-linearity, and may be in fact worse. The largest slope value of the sigmoid is 1/4 in the unstable region which means that when backpropagating through a sigmoid activation, the derivative will always be less than one. The sigmoid function also has the same problem as tanh in regions of saturation.

(C)

Following the structures used in the class notes, below is a sketch of the GRU.



There are a few core differences between the LSTM and GRU.

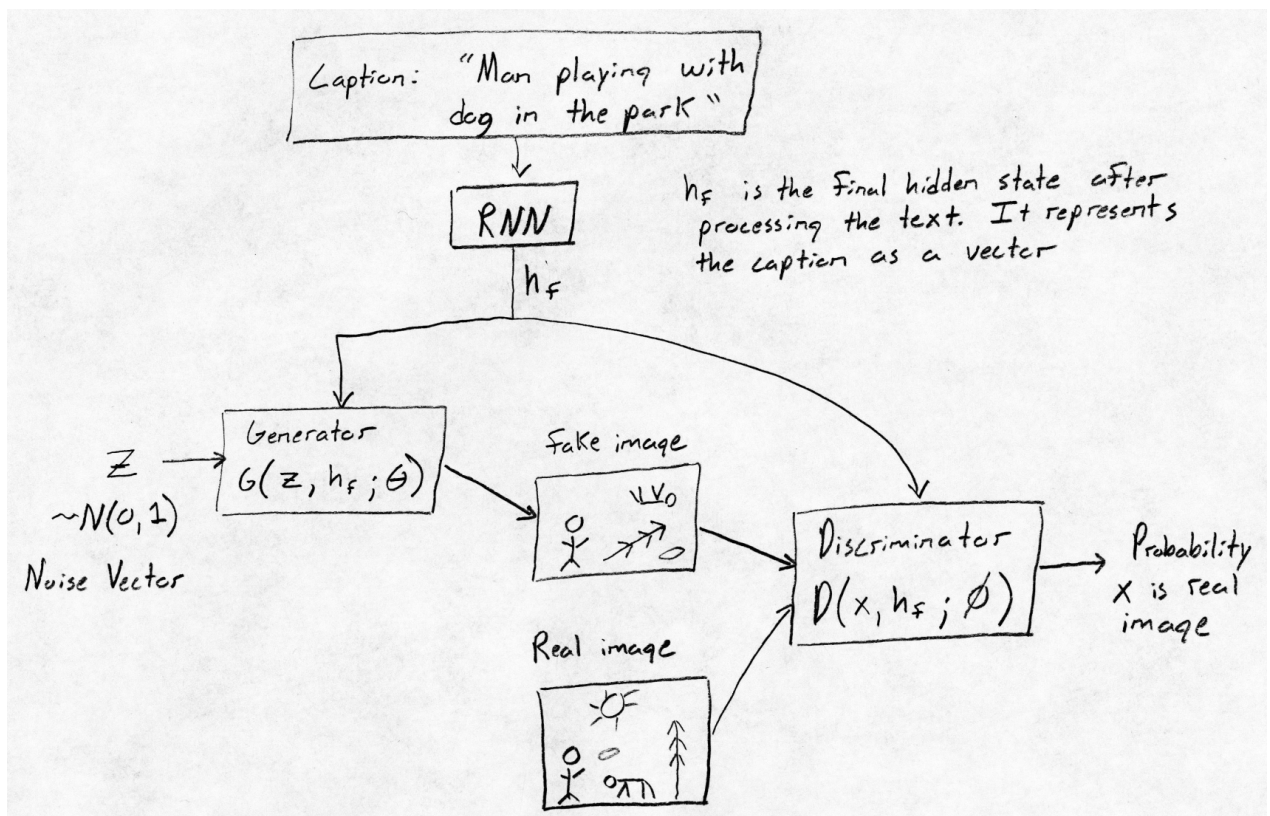
- GRU is less computationally expensive and simpler because it only has two gates (update gate & reset gate) as compared to LSTM's 3 gates (forget gate, input gate, output gate)
- The GRU does not have a cell state like LSTM does. Rather, it only has a hidden state  $h_n$ .
- GRU has less trainable parameters than LSTM

[1] - Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. The MIT Press.

[2] - Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28 (ICML'13), Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. JMLR.org III-1310-III-1318. (<http://proceedings.mlr.press/v28/pascanu13.pdf>) (<http://proceedings.mlr.press/v28/pascanu13.pdf>)

## Problem 2.7: Diagram for Generating Images to Match Text

Below is a diagram for how we may design a system to generate images from text captions.



The design is a composition of two main units, a RNN and a conditional GAN. The RNN here is some pretrained text encoder which is used to transform the english caption to a vector. In this case, we may process the text and use the final hidden state vector  $h_f$  as the encoded caption. The GAN is then conditioned on  $h_f$ , as we feed this encoding to both the generator and discriminator as inputs. During inference when we want to generate new images from text, we will first encode the caption with the RNN, then feed the encoding and the random noise vector to the Generator as input. The output will be an image representation of the input caption.

### Problem 3: Understanding the decay term in Q-learning (30 points)

(A)

(i) - When  $\gamma = 1$  the agent prioritizes long term reward and does not care about short term reward. Therefore, the optimal policy would be to move to state 2, as the future reward is 4 which is greater than the short term reward of 1 if we were to stay in state 1.

(ii) = When  $\gamma = 0$  the agent prioritizes short term reward and is not concerned with long term reward. Therefore, the optimal policy would be to stay in state 1 because the short term reward is 1 vs 0 for transitioning to state 2.

(iii) - We must first setup the expected future reward for both situations. For staying in state 1, the future reward is 1 but for transitioning to state 2 the future reward is  $0 + 0\gamma + 4\gamma^2$ . Thus, if we solve  $1 = 0 + 0\gamma + 4\gamma^2$  for  $\gamma$  we find the value where each action is equally valuable is  $\gamma = 1/2$

(B)

The calculation of future reward in this situation becomes  $0 + 1\gamma = 0 + 0\gamma + 3\gamma^2$ . So the critical value is  $\gamma = 1/3$ . In this case, if  $\gamma < 1/3$  the system would prioritize short term reward and the optimal policy would be to move to  $s = 0$ . However, if  $\gamma > 1/3$  the policy would prioritize long term reward and the agent would move to  $s = 2$ .

**(C)**

For the top system, the equation for future reward is  $1 = 0 + 0\gamma + 0\gamma^2 + 6\gamma^3$ . So the breakeven gamma value is  $\gamma = \sqrt[3]{1/6}$ .

For the bottom system, the future reward is  $1 = 0 + 0\gamma + 0\gamma^2 + \dots + 110\gamma^{99}$ . So the breakeven gamma value is  $\gamma = \sqrt[99]{1/110}$ .

**(D)**

We might prefer the constant reward rather than the larger reward less frequently because it may aid the learning process. If the reward signal is very sparse, it may be very difficult to get the agent to learn the task because it has to take so many actions before getting any type of reward. If the reward signal is more constant, the agent gets more regular feedback about what actions are good and which are bad. With this, it may take the agent longer to learn a good policy when the reward is very sparse because it has to take so many more actions just to get feedback.

## Problem 5: Bookkeeping (5 points)

**(A)**

This assignment took 20-25 hours.

**(B)**

"I adhered to the Duke Community Standard in the completion of this assignment" - NAI

In [ ]: