ECE 590 Intro to Deep Learning HW5+6

Yifan Li yl506@duke.edu.

Problem 1:

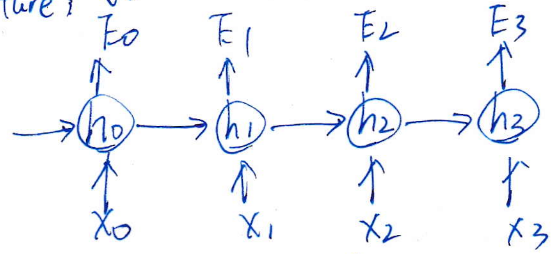
(a) Based on the expression of $h_t = \vec{w} h_{t-1} + \vec{A} x_t$, we have the recursion pattern.

At t+1, $h_{t+1} = \vec{w} h_t + \vec{A} x_{t+1} = \vec{w} \cdot (\vec{w} h_{t-1} + \vec{A} x_t) + \vec{A} x_{t+1}$ and we can see that

if the magnitude/norm of $\vec{w}$ is greater than 1, a long recursion would introduce enormous multiplication of $\vec{w}$. $\|\vec{w}\|^t$ will be very large for $\|\vec{w}\| > 1$ and $t >> 1$.
So the general requirement on $\vec{w}$ for a stable sequence is keeping the norm of $\vec{w}$ to be close to 1 to prevent the exponential increase in the $\|\vec{w}\|^t$.

(b) $h_t = \tanh(\vec{W} h_{t-1} + \vec{A} x_t)$. In this case, consider a simple RNN with following architecture: $\hat{y}_t = \text{softmax}(V h_t)$

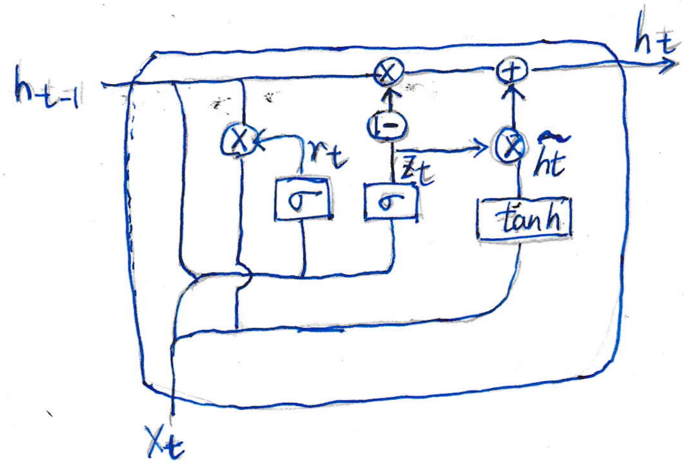
E denotes the error function. During backprop, $E = \sum_t E_t(y_t, \hat{y}_t)$



we have $\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \frac{\partial h_3}{\partial h_k} \frac{\partial h_k}{\partial W}$, $\frac{\partial h_3}{\partial h_k}$ is a chain rule, ~~so~~ $\cancel{\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3}}$

so $\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial h_3} \left(\prod_{j=k+1}^{3} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W}$, and the 2-norm of the Jacobian matrix has an upper bound of 1. Since tanh or sigmoid activation function maps all values in to a range between −1 and 1 (0 and 1 for sigmoid), the derivative is bounded as well. Thus, with multiple multiplications, the gradients shrink exponentially. So this problem (gradient vanishing) always exists for RNN, no matter if the activation function is sigmoid or hyperbolic tangent.

(c)



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

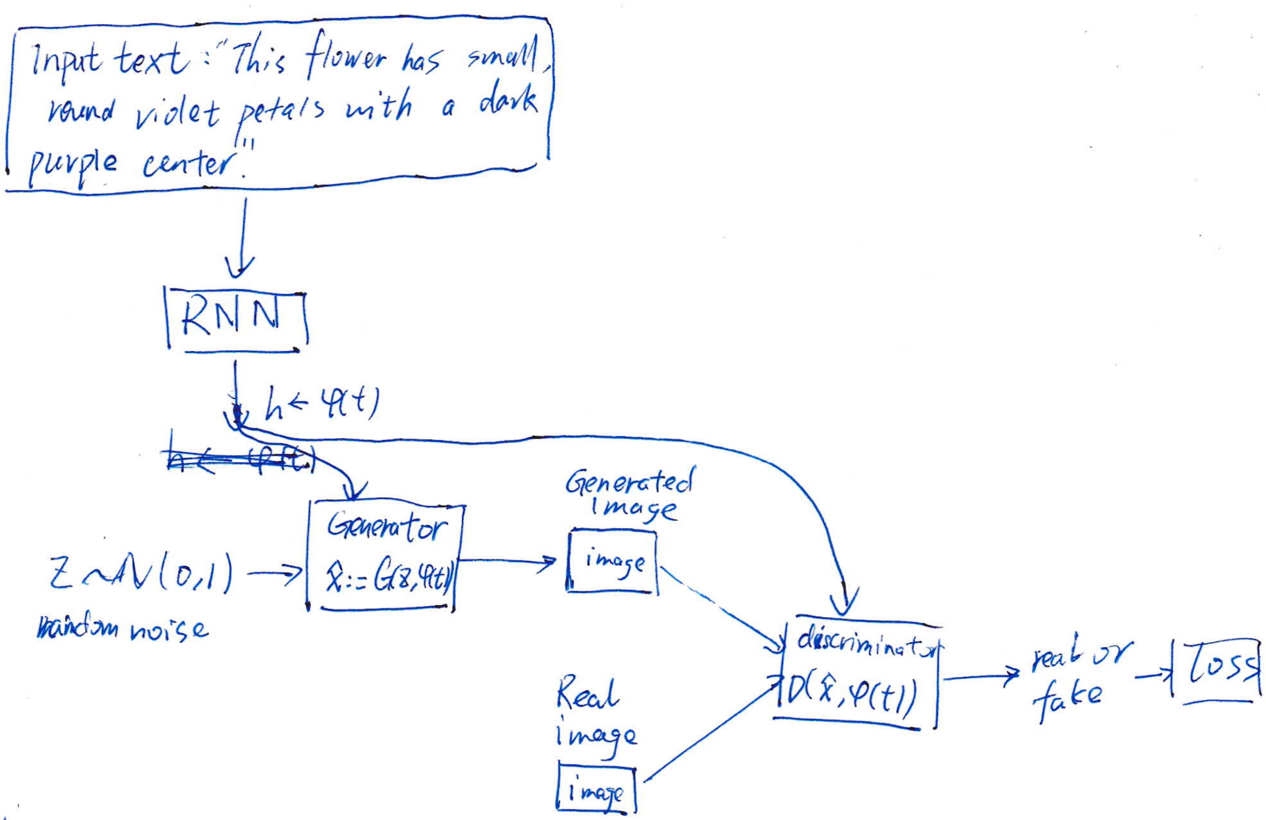
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Key difference: A GRU does not have a cell state, so it has 2 gates.
A GRU is more computationally efficient than an LSTM
Due to the reduction of gates, a GRU comes second to LSTM network in terms of performance. So GRU is more viable when computation power is limited or faster training time is preferred.

Problem 2

vii)

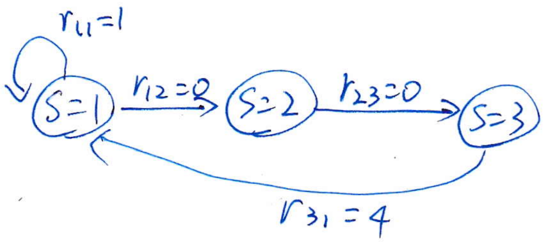


The approach is to train a ~~RNN~~ GAN conditioned on text features encoded by a RNN. Both the generator network and the discriminator network perform feed-forward inference conditioned on the text feature.

Intro to Deep Learning, HW 5+6 Yifan Li yl506

Problem 3.

(a)



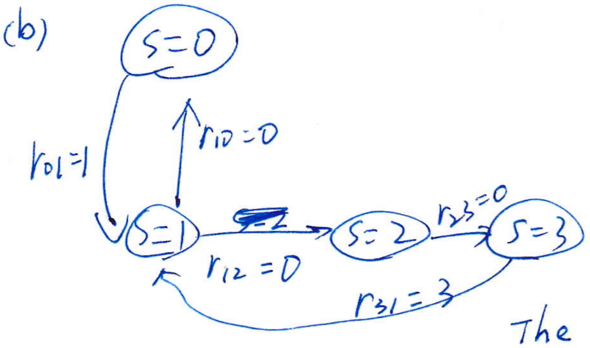
$$Q^{new}(s,a) = (1-\alpha)\, Q^{old}(s,a) + \cancel{\alpha [r+]} \cdot \alpha \cdot [r(s,a,s') + \gamma \cdot \max_{a'} Q^{old}(s',a')]$$

i) when $\gamma = 1$, the policy will aim for the long term rewards $Q(s',a')$. Thus, the optimal policy when $\gamma=1$ is to ~~move~~ explore (move) to state 2 for better future rewards.

ii) When $\gamma = 0$, the future rewards will be ignored and the short-term reward is maximized. So the policy is to stay at state 1.

iii) The total rewards of staying at $s_1 = 1$
The total rewards of exploring $= 4\gamma^2$. let $1 = 4\gamma^2 \Rightarrow \gamma = \frac{1}{2}$

(b)

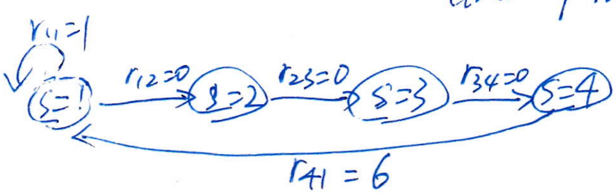


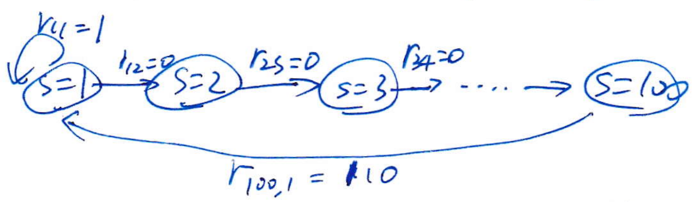
from state 1, reward of going back to $s_0 = 0 + \gamma$
and the reward of going to s_2 and keeping exploring
$= 0 + 0\gamma + 3\gamma^2$

So $3\gamma^2 = \gamma$, $\gamma = \frac{1}{3}$ is the threshold γ.

The policy prioritizes short-term rewards if $\gamma < \frac{1}{3}$ and prioritizes long-term rewards if $\gamma > \frac{1}{3}$

(c)





$1 = 0 + 0\gamma + 0\gamma^2 + 6\gamma^3 \Rightarrow \gamma = \sqrt[3]{\frac{1}{6}}$

$1 = 0 + 0\gamma + 0\gamma^2 + \cdots 110\gamma^{99} \Rightarrow \gamma = \sqrt[99]{\frac{1}{110}}$

d) Our Learning process finds better strategy with higher rewards. We would prefer constant reward less frequently because exploration is necessary to learn a better policy, while more exploitation delays the learning process.