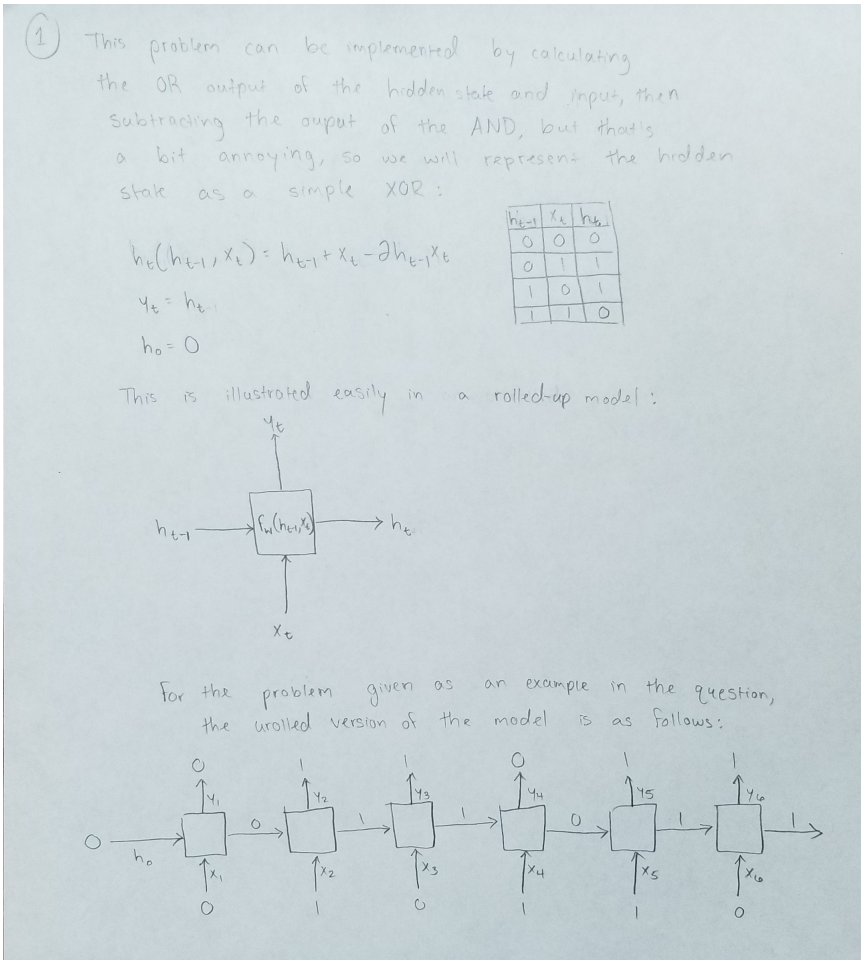
CS7643: Deep Learning
Fall 2019
HW3 Solutions

James Hahn

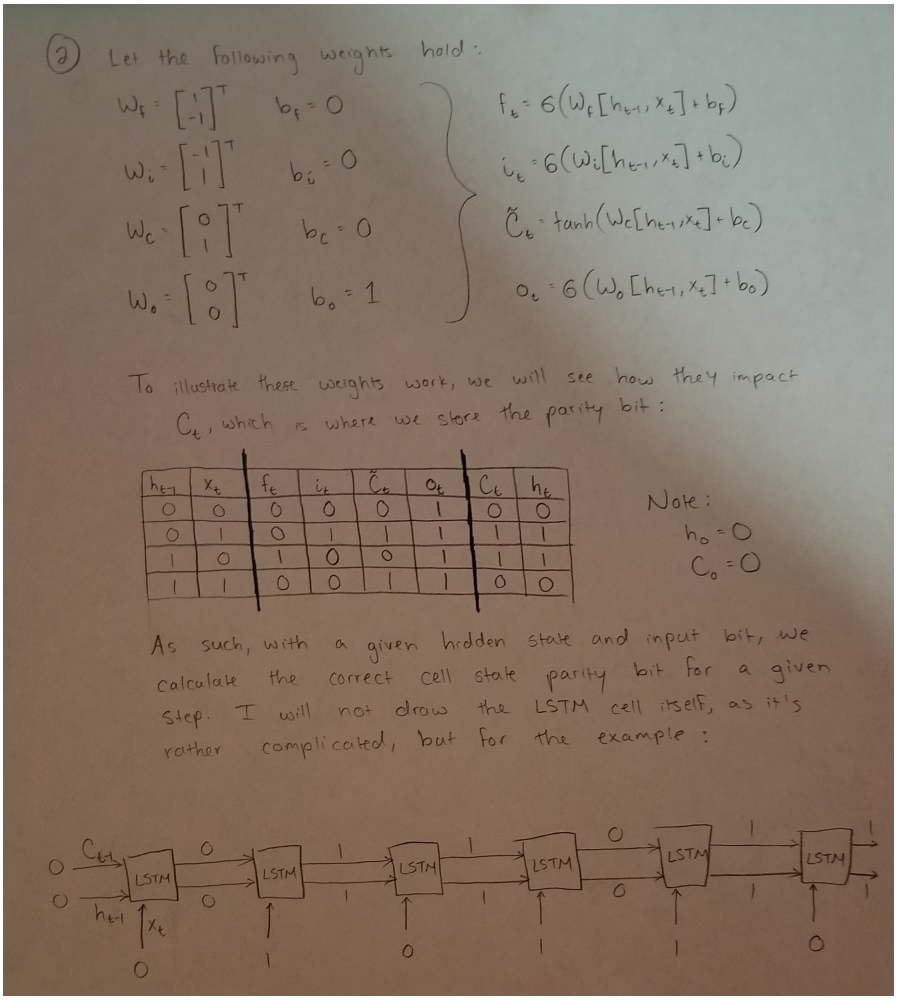
October 18, 2019

1 Recurrent Neural Network

1. The answer can be found below:



2. The answer can be found below:



(2) Let the following weights hold:

$W_f = \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T \quad b_f = 0$

$W_i = \begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \quad b_i = 0$

$W_C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T \quad b_C = 0$

$W_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}^T \quad b_o = 1$

$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$

$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$

$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

To illustrate these weights work, we will see how they impact C_t, which is where we store the parity bit:

h_{t-1}	x_t	f_t	i_t	$\tilde{C}_t$	o_t	C_t	h_t
0	0	0	0	0	1	0	0
0	1	0	1	1	1	1	1
1	0	1	0	0	1	1	1
1	1	0	0	1	1	0	0

Note:
$h_0 = 0$
$C_0 = 0$

As such, with a given hidden state and input bit, we calculate the correct cell state parity bit for a given step. I will not draw the LSTM cell itself, as it's rather complicated, but for the example:

3. We want to show at a given step i of the beam search, if $best_{\leq i}$ exists, then all other beam scores B_i at iteration i will be worse than $best_{\leq i}$, so it is not worth exploring the search.

   First, we know for a given B_i list of length i, its probably $p(y_i|x, y_{<i}) = s$. Second, we know $\sum_x p(y_{i+1}|x, y_{<i+1}) = 1$. As such, for a given x, $p(y_{i+1}|x, y_{<i+1}) = p(x)p(y_{i+1}|y_{<i+1})$. This means if $p(x) \in [0, 1]$, then the current score s at B_i will have the value $s_{i+1} = p(x) \cdot s \leq s$ for the list B_{i+1}. As such, if B_i has probability s that's lower than $best_{\leq i}$, then B_{i+1} will have score s_{i+1} that is at best as high as s, so it is useless exploring that set of prefixes. To conclude, $best_{\leq i}$ will still be the optimal solution for all iterations i and beyond.

   $\therefore$ $best_{\leq i}$ is the overall highest-probability completed hypothesis and future steps will be no better than it

4. We want to show in a vanilla RNN, with a basic recurrence relation (i.e. $h_t = W^T h_{t-1}$), there exists both the vanishing and exploding gradient problem.

   First, we notice $h_t = W^T h_{t-1} = (W^T)^2 h_{t-2} = \cdots = (W^T)^t h_0$.

   So, $\frac{dh_t}{dh_0} = (W^T)^t$. Let W be size $n \times n$. We know W can be decomposed into its eigenvalues and eigenvectors such that $W = Q\Lambda Q^{-1}$, where Q is a square $n \times n$ matrix holding the eigenvectors in its columns and Λ is a diagonal matrix where each element along the diagonal corresponds to its respective eigenvector in Q.

   Now, we get $\frac{dh_t}{dh_0} = (W^T)^t$
   $= ([Q\Lambda Q^{-1}]^T)^t$
   $= ([Q^{-1}]^T \Lambda^T Q^T)^t$ (NOTE: $(ABC)^T = C^T B^T A^T$)
   $= ([Q^{-1}]^T \Lambda Q^T)^t$ (NOTE: If A is diagonal, then $A = A^T$)
   $= [Q^{-1}]^T \Lambda^t Q^T$

   So, we see the gradient we are trying to calculate heavily relies on the eigenvalues in Λ. We are given that $\rho(W) = max\{|\lambda_1|, \cdots, |\lambda_n|\}$. Assume t itself is relatively large (i.e. we are doing deep learning, otherwise the whole vanishing/exploding gradients thing isn't a concern in the first place). If $\rho(W)$ is greater than 1, then at least one value of Λ^t will exponentially grow toward $\pm\infty$, and the gradient itself will eventually be multiplied by this $\pm\infty$ value, hence the term "exploding gradients". On the other hand, if $\rho(W)$ is less than 1, then all eigenvalues will be somewhat close to 0, and the elements of Λ^t will converge to 0 (due to repeated multiplications), leading to the gradient becoming almost completely 0, hence the term "vanishing gradient".

   $\therefore$ We have shown in a vanilla RNN, with a simple recurrence relation, the vanishing/exploding gradient problem exists purely due to the eigenvalues of W $\square$