

# TTIC 31230 Fundamentals of Deep Learning, winter 2020

## Quiz 2

In these problems capital letter indeces are used to indicate subtensors (slices) so that, for example,  $M[I, J]$  denotes a matrix while  $M[i, j]$  denotes one element of the matrix,  $M[i, J]$  denotes the  $i$ th row, and  $M[I, j]$  denotes the  $j$ th column.

Throughout these problems we assume a word embedding matrix  $e[W, I]$  where  $e[w, I]$  is the word vector for word  $w$ . We then have that  $e[w, I]^\top h[t, I]$  is the inner product of the word vector  $w[w, I]$  and the hidden state vector  $h[t, I]$ .

We will adopt the convention, similar to true Einstein notation, that repeated capital indeces in a product of tensors are implicitly summed. We can then write the inner product  $e[w, I]^\top h[t, I]$  simply as  $e[w, I]h[t, I]$  without the need for the (meaningless) transpose operation.

The batch index is omitted in all equations in this quiz.

**Problem 1. Image captioning as translation with attention.** We consider a simple version of machine translation with attention. We first run a right-to-left (backward) RNN on the input sentence to get a sequence  $\tilde{h}[T_{\text{in}}, J]$  of hidden vectors  $\tilde{h}[t, J]$  for  $1 \leq t \leq T_{\text{in}}$  where  $T_{\text{in}}$  is the length of the input sentence. We then define an autoregressive conditional language model

$$P_\Phi(w_1, \dots, w_{T_{\text{out}}} \mid \tilde{h}[T_{\text{in}}, J])$$

as follows.

$$\vec{h}[0, J] = \vec{h}[1, J]$$

for  $t$  from 1 to  $T_{\text{out}}$

$$P(w_t \mid w_0, \dots, w_{t-1}) = \underset{w_t}{\text{softmax}} e[w_t, I] W^{\text{auto}}[I, J] \vec{h}[t-1, J]$$

$$\alpha[t_{\text{in}}] = \underset{t_{\text{in}}}{\text{softmax}} h[t-1, J_1] W^{\text{key}}[J_1, J_2] \tilde{h}[t_{\text{in}}, J_2]$$

$$V[J] = \sum_{t_{\text{in}}} \alpha[t_{\text{in}}] \tilde{h}[t_{\text{in}}, J]$$

$$\vec{h}[t, J] = \text{CELL}_\Phi(\vec{h}[t-1, J], V[J], e[w_t, I])$$

Here CELL is some function taking (objects for) two vectors of dimensions  $J$  and one vector of dimension  $I$  and returning (an object for) a vector of dimension  $J$ .

Rewrite these equations for image captioning where instead of  $\tilde{h}[t_{\text{in}}, J]$  we are given an image feature tensor  $L[x, y, J]$

**Solution:**

$$P_{\Phi}(w_1, \dots, w_{T_{\text{out}}} \mid L[X, Y, J])$$

$$\vec{h}[0, J] = \frac{1}{XY} \sum_{x,y} L[x, y, J] \text{ Any function of the image gets full credit}$$

for  $t$  from 1 to  $T_{\text{out}}$

$$P(w_t \mid w_0, \dots, w_{t-1}) = \underset{w_t}{\text{softmax}} e[w_t, I] W^{\text{auto}}[I, J] \vec{h}[t-1, J]$$

$$\alpha[x, y] = \underset{x,y}{\text{softmax}} h[t-1, J_1] W^{\text{key}}[J_1, J_2] L[x, y, J_2]$$

$$V[J] = \sum_{x,y} \alpha[x, y] L[x, y, J]$$

$$\vec{h}[t, J] = \text{CELL}_{\Phi}(\vec{h}[t-1, J], V[J], e[w_t, I])$$

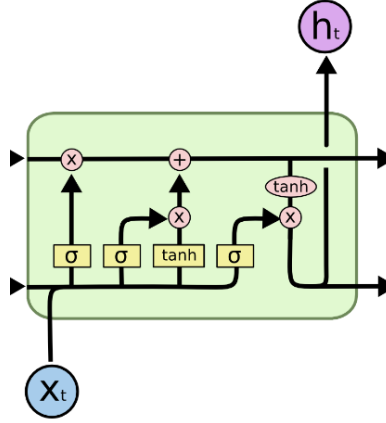
**Problem 2. Translating diagrams into equations.** A UGRNN cell for computing  $h[t, J]$  from  $h[t-1, J]$  and  $x[t, J]$  can be written as

$$G[t, j] = \sigma \left( W^{h,G}[j, \tilde{J}] h[t-1, \tilde{J}] + W^{x,G}[j, K] x[t, K] - B^G[j] \right)$$

$$R[t, j] = \tanh \left( W^{h,R}[j, \tilde{J}] h[t-1, \tilde{J}] + W^{x,R}[j, K] x[t, K] - B^R[j] \right)$$

$$h[t, j] = G[t, j] h[t-1, j] + (1 - G[t, j]) R[t, j]$$

Modify the above equations so that they correspond to the following diagram for an LSTM.



The top line in the diagram is the “carry vector”  $c_t$ . The equations for an LSTM should define  $h[t, j]$  and  $c[t, j]$  in terms of  $h[t-1, J]$ ,  $c[t-1, J]$  and  $x[t, K]$ .

**Solution:**

$$G_1[t, j] = \sigma \left( W^{h, G_1}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x, G_1}[j, K]x[t, K] - B^{G_1}[j] \right)$$

$$G_2[t, j] = \sigma \left( W^{h, G_2}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x, G_2}[j, K]x[t, K] - B^{G_2}[j] \right)$$

$$G_3[t, j] = \sigma \left( W^{h, G_3}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x, G_3}[j, K]x[t, K] - B^{G_3}[j] \right)$$

$$R_c[t, j] = \tanh \left( W^{h, c}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x, c}[j, K]x[t, K] - B^c[j] \right)$$

$$c[t, j] = G_1[t, j]c[t-1, j] + G_2[t, j]R_c[t, j]$$

$$R_h[t, j] = \tanh \left( W^{c, h}[j, \tilde{J}]c[t, \tilde{J}] - B^h[j] \right)$$

$$h[t, j] = G_3[t, j]R_h[t, j]$$

### Problem 3. Gated CNNs

Again, A UGRNN is defined by the following equations.

$$G[t, j] = \sigma(W^{h,G}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x,G}[j, k]x_t[t, k] - B^G[j])$$

$$R[t, j] = \tanh(W^{h,R}[j, \tilde{J}]h[t-1, \tilde{J}] + W^{x,R}[j, K]x[t, K] - B^R[j])$$

$$h[t, j] = G[t, j]h[t-1, j] + (1 - G[t, j])R[t, j]$$

Modify these to form a data-dependent data-flow CNN for vision — an Update-Gate CNN (UGCNN). More specifically, give equations analogous to those for UGRNN for computing a CNN “box”  $L_{\ell+1}[x, y, j]$  from  $L_\ell[x, y, i]$  (stride 1) using a computed “gate box”  $G_{\ell+1}[x, y, j]$  and an “update box”  $R_{\ell+1}[x, y, j]$ . In the CNN case there is no  $x[t, I]$ , just the previous layer  $L_\ell[x, y, J]$ .

**Solution:**

$$R_{\ell+1}[x, y, j] = \tanh(W_{\ell+1}^{L,R}[\Delta X, \Delta Y, I, j] L_\ell[x + \Delta X, y + \Delta Y, I] - B_{\ell+1}^R[j])$$

$$G_{\ell+1}[x, y, j] = \sigma(W_{\ell+1}^{L,G}[\Delta X, \Delta Y, I, j] L_\ell[x + \Delta X, y + \Delta Y, I] - B_{\ell+1}^G[j])$$

$$L_{\ell+1}[x, y, j] = G_{\ell+1}[x, y, j]L_\ell[x, y, j] + (1 - G_{\ell+1}[x, y, j])R_{\ell+1}[x, y, j]$$

**Problem 4. Variance of running averages.** For two independent random variables  $x$  and  $y$  and a weighted sum  $s = ax + by$  we have

$$\sigma_s^2 = a^2\sigma_x^2 + b^2\sigma_y^2$$

Now consider a running average for computing  $\hat{\mu}_1, \dots, \hat{\mu}_t$  from  $x_1, \dots, x_t$

$$\hat{\mu}_0 = 0$$

$$\hat{\mu}_t = \left(1 - \frac{1}{N}\right)\hat{\mu}_{t-1} + \frac{1}{N}x_t$$

(a) Assume that the values of  $x_t$  are independent and identically distributed with variance  $\sigma_x^2$ . We now have that  $\hat{\mu}_t$  is a random variable depending on the draws of  $x_t$ . The random variable  $\hat{\mu}_t$  has a variance  $\sigma_{\hat{\mu},t}^2$ . Assume that as  $t \rightarrow \infty$  we have that  $\sigma_{\hat{\mu},t}^2$  converges to a limit (it does). Solve for this limit  $\sigma_{\hat{\mu},\infty}^2$ . Your solution should yield that for  $N = 1$  we have  $\sigma_{\hat{\mu},\infty}^2 = \sigma_x^2$  (a sanity check).

**Solution:** The limit must satisfy

$$\sigma_{\hat{\mu},\infty}^2 = \left(1 - \frac{1}{N}\right)^2 \sigma_{\hat{\mu},\infty}^2 + \frac{1}{N^2} \sigma_x^2$$

We can then solve for  $\sigma_{\hat{\mu},\infty}^2$

$$\begin{aligned}\sigma_{\hat{\mu},\infty}^2 &= \left(1 - \frac{2}{N} + \frac{1}{N^2}\right) \sigma_{\hat{\mu},\infty}^2 + \frac{1}{N^2} \sigma_x^2 \\ 0 &= \left(\frac{-2}{N} + \frac{1}{N^2}\right) \sigma_{\hat{\mu},\infty}^2 + \frac{1}{N^2} \sigma_x^2 \\ &= \left((-2) + \frac{1}{N}\right) \sigma_{\hat{\mu},\infty}^2 + \frac{1}{N} \sigma_x^2 \\ \sigma_{\hat{\mu},\infty}^2 &= \frac{2 - \frac{1}{N}}{N} \sigma_x^2\end{aligned}$$

(b) Compare your answer to (a) with the variance of an average of  $N$  values of  $x_t$  defined by

$$\hat{\mu} = \frac{1}{N} \sum_{t=1}^N x_t$$

**Solution:** For an average of  $N$  we have  $\sigma_{\hat{\mu}}^2 = \sigma_x^2/N$ . For  $N$  large we have that the answer to part (a) is about twice as large.