TTIC 31230 Fundamentals of Deep Learning, winter 2019

Quiz 2

Problem 1. 25 points. Equations defining a UGRNN are given below.

$$
\begin{aligned}
\tilde{R}_t[b,j] &= \left(\sum_i W^{h,R}[j,i]h_{t-1}[b,i]\right) + \left(\sum_k W^{x,R}[j,k]x_t[b,k]\right) - B^R[j] \\
R_t[b,j] &= \tanh(\tilde{R}_t[b,j]) \\
\tilde{G}_t[b,j] &= \left(\sum_i W^{h,G}[j,i]h_{t-1}[b,i]\right) + \left(\sum_k W^{x,G}[j,k]x_t[b,k]\right) - B^G[j] \\
G_t[b,j] &= \sigma(\tilde{G}_t[b,j]) \\
h_t[b,j] &= G_t[b,j]h_{t-1}[b,j] + (1 - G_t[b,j])R_t[b,j]
\end{aligned}
$$

(a) Rewrite the first equation defining $\tilde{R}_t$ using `+=` loops instead of summations assuming that all computed tensors are initialized to zero.

Solution:

$$
\begin{aligned}
\text{for } b,j,i\ \tilde{R}_t[b,j] &\mathrel{+}= W^{h,R}[j,i]h_{t-1}[b,i] \\
\text{for } b,j,k\ \tilde{R}_t[b,j] &\mathrel{+}= W^{X,R}[k,i]x_t[b,k] \\
\text{for } b,j\ \tilde{R}_t[b,j] &\mathrel{-}= B^R[j]
\end{aligned}
$$

(b) Give `+=` loops for the backward computation for your solution to part (a) using the convention that parameter gradients are averaged over the batch and where the batch size is B.

Solution:

$$\text{for } b,j,i\ W^{h,R}.\text{grad}[j,i] \quad \texttt{+=} \quad \frac{1}{B}\, h_{t-1}[b,i]\tilde{R}_t.\text{grad}[b,j]$$

$$\text{for } b,j,i\ h_{t-1}.\text{grad}[b,j] \quad \texttt{+=} \quad W^{h,R}[j,i]\tilde{R}_t.\text{grad}[b,j]$$

$$\text{for } b,j,k\ W^{x,R}.\text{grad}[j,k] \quad \texttt{+=} \quad \frac{1}{B}\, x[b,k]\tilde{R}_t.\text{grad}[b,j]$$

$$\text{for } b,j,k\ h_{t-1}.\text{grad}[b,j] \quad \texttt{+=} \quad W^{x,R}[j,k]\tilde{R}_t.\text{grad}[b,j]$$

$$\text{for } b,j\ B^{R}.\text{grad}[j] \quad \texttt{-=} \quad \frac{1}{B}\, \tilde{R}_t.\text{grad}[b,j]$$

Problem 2. 25 points. Images have translation invariance — a person detector must look for people at various places in the image. Translation invariance is the motivation for convolution — all places in the image are treated the same.

Images also have some degree of scale invariance — a person detector must look for people of different sizes (near the camera or far from the camera). We would like to design a deep architecture that treats all scales (sizes) the same in a manner that similar to the way CNNs treat all places the same.

Consider a batch of images $I[b,x,y,c]$ where c ranges over the three color values red, green, blue. We start by constructing an "image pyramid" $I_s[x,y,c]$. We assume that the original image $I[b,x,y,c]$ has spatial dimensions 2^k and construct images $I_s[b,x,y,c]$ with spatial dimensions 2^{k-s} for $0 \leq s \leq s_{\max} < k$. These images are defined by the following equations.

$$\begin{aligned} I_0[b,x,y,c] &= I[b,x,y,c] \\ I_{s+1}[b,x,y,c] &= \frac{1}{4}\left(\begin{array}{l} I_s[b,2x,2y,c] + I_s[b,2x+1,2y,c] \\ +I_s[b,2x,2y+1,c] + I_s[b,2x+1,2y+1,c] \end{array}\right) \end{aligned}$$

We want to compute a set of layers $L_{s,\ell}[b,x,y,i]$ where s is the scale and ℓ is the level of processing. First we set

$$L_{0,s}[b,x,y,c] = I_s[b,x,y,c].$$

The layers $L_{\ell,0}[b,x,y,i]$ can be computed using the standard CNN equations holding the scale at zero.

Give an equation for a linear threshold unit to compute $L_{\ell+1,s+1}[b,x,y,j]$ from $L_{\ell,s+1}[b,x,y,j]$ **and** $L_{\ell+1,s}[b,x,y,j]$. Assume that the spatial dimension of $L_{\ell,s}$ is 2^{k-s} and use an appropriate stride between $L_{\ell+1,s+1}[b,x,y,j]$ and $L_{\ell+1,s}[b,x,y,j]$. Use parameters $W_{\ell+1,\rightarrow}[\Delta x,\Delta y,i,j]$ for the dependence of $L_{\ell+1,s}$ on $L_{\ell,s}$ and parameters $W_{\ell+1,\uparrow}[\Delta x, \Delta y, i, j]$ for the dependence of $L_{\ell+1,s+1}$ on $L_{\ell+1,s}$. Use $B_{\ell+1}[j]$ for the threshold. Note that these parameters do not depend on s — they are scale invariant.

Solution:

$$L_{\ell+1,s+1}[b,x,y,j] = \sigma\left(\begin{array}{ll} \sum_{\Delta x,\Delta y,i} & W_{\ell+1,\rightarrow}[\Delta x, \Delta y, i, j] L_{\ell,s+1}[b, x+\Delta x,\ y+\Delta y,\ i] \\ \sum_{\Delta x,\Delta y,i} & W_{\ell+1,\uparrow}[\Delta x, \Delta y, i, j] L_{\ell+1,s}[b, 2x+\Delta x,\ 2y+\Delta y,\ i] \\ -B_{\ell+1}[j] & \end{array}\right)$$

Problem 3. 25 points.
Modify the equations for a UGRNN from problem 1 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" $L_{\ell+1}[b,x,y,j]$ from $L_\ell[b,x,y,i]$ (stride 1) using a computed "gate box" $G_{\ell+1}[b,x,y,j]$ and an "update box" $R_{\ell+1}[b,x,y,j]$.

$$\Phi = (W^{L,R}_{\ell+1}[\Delta x, \Delta y, j, j'], B^R_{\ell+1}[j], W^{L,G}_{\ell+1}[\Delta x, \Delta y, j, j'],\ B^G_{\ell+1}[j])$$

Solution:

$$\begin{aligned}
R_{\ell+1}[b,x,y,j] &= \tanh\left(\left(\sum_{\Delta x,\Delta y,j'} W^{L,R}_{\ell+1}[\Delta x, \Delta y, j', j] L_\ell[b, x+\Delta x,\ y+\Delta y,\ j']\right) - B^R_{\ell+1}[j]\right) \\
G_{\ell+1}[b,x,y,j] &= \sigma\left(\left(\sum_{\Delta x,\Delta y,i} W^{L,G}_{\ell+1}[\Delta x, \Delta y, i, j'] L_\ell[b, x+\Delta x,\ y+\Delta y,\ j']\right) - B^G_{\ell+1}[j]\right) \\
L_{\ell+1}[b,x,y,j] &= G_{\ell+1}[b,x,y,j] L_\ell[b,x,y,j] + (1 - G_t[b,x,y,j]) R_t[b,x,y,j]
\end{aligned}$$

Problem 4. 25 points. This problem is on CNNs for sentences. We consider a model with parameters

$$\Phi = (e[w,i], W_1[\Delta t, i, i'], B_1[i], \ldots, W_L[\Delta t, i, i'], B_L[i])$$

The matrix e is the word embedding matrix where $e[w, I]$ is the vector embedding of word w.

(a) Give an equation for the convolution layer $L_0[b,t,i]$ as a function of the word embeddings and the input sentence $w_1, \ldots, w_T$.

Solution:

$$L_0[b,t,i] = e[w[t], i]$$

(b) Give an equation for $L_{\ell+1}[b,t,i]$ as a function of $L_\ell[b,t,i]$ and the parameters $W_{\ell+1}[\Delta t, i', i]$ and $B_{\ell+1}[i]$ and where $L_{\ell+1}$ is computed stride 2.

Solution:

$$L_{\ell+1}[b,t,i] = \sigma\left(\left(\sum_{\Delta t, i'} W_{\ell+1}[\Delta t, i', i] L_\ell[2t + \Delta t, i']\right) - B_{\ell+1}[i]\right)$$

(c) Assuming all computations can be done in parallel as soon the inputs have been computed, what is the **parallel** order of run time for this convolutional model as a function of the input length T and the number of layers L (assume all parameter tensors of size $O(1)$). Compare this with the parallel run time of an RNN.

Solution: The CNN has $O(L)$ parallel run time while the RNN is $O(T)$ or $O(T + L)$ with L layers of RNN.