CSC2516: Homework #4

Due on March 14

Roger Grosse, Jimmy Ba

Thomas Hollis

Problem 1

Derive the Backprop Through Time equations for the activations and the gates.

Solution

The BackProp Through Time (BPTT) equations are as follows:

$$\begin{aligned}\overline{h^{(t)}} &= \overline{o^{(t+1)}}\frac{\partial o^{(t+1)}}{\partial h^{(t)}} + \overline{f^{(t+1)}}\frac{\partial f^{(t+1)}}{\partial h^{(t)}} + \overline{i^{(t+1)}}\frac{\partial i^{(t+1)}}{\partial h^{(t)}} + \overline{g^{(t+1)}}\frac{\partial g^{(t+1)}}{\partial h^{(t)}} \\ &= \overline{o^{(t+1)}}\sigma(w_{ox}x^{(t+1)} + w_{oh}h^{(t)})(1 - \sigma(w_{ox}x^{(t+1)} + w_{oh}h^{(t)}))w_{oh} \\ &+ \overline{f^{(t+1)}}\sigma(w_{fx}x^{(t+1)} + w_{fh}h^{(t)})(1 - \sigma(w_{fx}x^{(t+1)} + w_{fh}h^{(t)}))w_{fh} \\ &+ \overline{i^{(t+1)}}\sigma(w_{ix}x^{(t+1)} + w_{ih}h^{(t)})(1 - \sigma(w_{ix}x^{(t+1)} + w_{ih}h^{(t)}))w_{ih} \\ &+ \overline{g^{(t+1)}}(1 - \tan^2(w_{gx}x^{(t+1)} + w_{gh}h^{(t)})w_{gh} \end{aligned} \tag{1}$$

$$\overline{c^{(t)}} = \overline{h^{(t)}}\frac{\partial h^{(t)}}{\partial c^{(t)}} + \overline{c^{(t+1)}}\frac{\partial c^{(t+1)}}{\partial c^{(t)}} = \overline{h^{(t)}}o^{(t)}\left(1 - \tanh^2(c^{(t)})\right) + \overline{c^{(t+1)}}f^{(t+1)} \tag{2}$$

$$\overline{g^{(t)}} = \overline{c^{(t)}}\frac{\partial c^{(t)}}{\partial g^{(t)}} = \overline{c^{(t)}}i^{(t)} \tag{3}$$

$$\overline{o^{(t)}} = \overline{h^{(t)}}\frac{\partial h^{(t)}}{\partial o^{(t)}} = \overline{h^{(t)}}\tanh(c^{(t)}) \tag{4}$$

$$\overline{f^{(t)}} = \overline{c^{(t)}}\frac{\partial c^{(t)}}{\partial f^{(t)}} = \overline{c^{(t)}}c^{(t-1)} \tag{5}$$

$$\overline{i^{(t)}} = \overline{c^{(t)}}\frac{\partial c^{(t)}}{\partial i^{(t)}} = \overline{c^{(t)}}g^{(t)} \tag{6}$$

Since:

$$\tanh'(x) = 1 - \tan^2(x) \tag{7}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \tag{8}$$

Problem 2

Derive the BPTT equation for the weight $\overline{w_{ix}}$.

Solution

The BPTT equation is as follows:

$$\overline{w_{ix}} = \sum_{t=1}^{T}\overline{i^{(t)}}\frac{\partial i^{(t)}}{\partial w^{ix}} = \sum_{t=1}^{T}\overline{i^{(t)}}\sigma(w_{ix}x^{(t)} + w_{ih}h^{(t-1)})(1 - \sigma(w_{ix}x^{(t)} + w_{ih}h^{(t-1)}))x^{(t)} \tag{9}$$

Problem 3

How many weights does this architecture have? How many arithmetic operations are required to compute the hidden activations?

Solution

Since all the weights are shared across each input node, we have the following total number of weights:

$$\text{total weights} = \mathbf{W}_{\text{in}}^{\mathbf{T}} + \mathbf{W}_{\text{W}}^{\mathbf{T}} + \mathbf{W}_{\text{N}}^{\mathbf{T}} = D \times H + H \times H + H \times H = DH + 2H^2 \tag{10}$$

Thus there are $DH + 2H^2$ weights in total.

As for the total arithmetic operations required to compute the hidden activations, this corresponds to one full forward pass. Since we know that a matrix multiplication for matrices of dimensions A by B and B by C requires multiplications and additions thus this is of complexity $\mathcal{O}(ABC)$. Therefore in our case we have:

$$\text{total ops} \approx G^2(DH + H^2) \tag{11}$$

Thus there are $\mathcal{O}(G^2DH + G^2H^2)$ arithmetic computations in total.

Problem 4

Suppose that in each step, you can compute as many matrix-vector multiplications as you like. How many steps are required to compute the hidden activations? Explain your answer.

Solution

A naive approach would be to say that there are G^2 steps required in total. However, if we assume we can compute as many matrix vector multiplications as we like in 1 step, we know that we can compute the next step if the north and west neighbours are known. Therefore, we can traverse the grid diagonally in our computation. This means we can compute it using $2G - 1$ steps (i.e. a 3 by 3 grid takes 5 steps, a 4 by 4 takes 7...). Thus if we assume we can compute as many matrix vector multiplications as we like in 1 step, as well as the additions and activation in that same step, thus the total steps required are $(2G - 1)$ steps.

Therefore, under these assumptions and definitions, this computation is possible in $\mathcal{O}(G)$ steps.

Problem 5

Give one advantage and one disadvantage of an MDRNN compared to a conv net.

Solution

As indicated in the original paper, an advantage of the MDRNN is that it is more robust to input warping than CNNs. This is because MDRNNs generalise standard RNNs by providing recurrent connections along all spatio-temporal dimensions present in the data. Indeed, these connections make MDRNNs robust to local distortions along any combination of input dimensions (e.g. image rotations, shears, vertical and horizontal displacements) and allow MDRNNs to model multidimensional context in a flexible way.

On the other hand, a disadvantage of MDRNNs is that they are more vulnerable to vanishing gradients than CNNs. Fortunately, it is possible to implement MDRNNs with LSTM cells to help alleviate this issue of vanishing gradients.

Problem 6

Show how to compute the inverse, $\mathbf{s}^{(k)} = f^{-}1(\mathbf{s}^{(k+1)})$.

Solution

Since the architecture is reversible and since we know that:

$$\begin{aligned} \mathbf{s}^{(k+1)} &= f(\mathbf{s}^{(k)}) \\ \therefore f^{-1}(\mathbf{s}^{(k+1)}) &= f^{-1}(f(\mathbf{s}^{(k)})) \\ \therefore \mathbf{s}^{(k)} &= f^{-}1(\mathbf{s}^{(k+1)}) \end{aligned} \tag{12}$$

Thus we can compute the inverse $\mathbf{s}^{(k)} = f^{-}1(\mathbf{s}^{(k+1)})$ by rewriting the updates as follows:

$$\mathbf{p}^{(k)} = \frac{\mathbf{p}^{(k+1)} + \alpha \nabla J(\boldsymbol{\theta}^{(k)})}{\beta} \tag{13}$$

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k+1)} - \mathbf{p}^{(k+1)} \tag{14}$$

Problem 7

Find the determinant of the Jacobian, i.e. $\det \partial \mathbf{s}^{(k+1)} / \partial \mathbf{s}^{(k)}$.

Solution

We can begin by rewriting the update rules in terms of each other as follows:

$$\mathbf{p}^{(k+1)} = \beta \mathbf{p}^{(k)} - \alpha \nabla J(\boldsymbol{\theta}^{(k)}) \tag{15}$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \beta \mathbf{p}^{(k)} - \alpha \nabla J(\boldsymbol{\theta}^{(k)}) \tag{16}$$

To simplify the subsequent equations and to be consistent with Prof. Grosse's notation in lectures, let:

$$x_1 = \mathbf{p}^{(k)} \tag{17}$$

$$x_2 = \boldsymbol{\theta}^{(k)} \tag{18}$$

$$\mathbf{x} = \mathbf{s}^{(k)} \tag{19}$$

$$y_1 = \mathbf{p}^{(k+1)} \tag{20}$$

$$y_2 = \boldsymbol{\theta}^{(k)} \tag{21}$$

$$\mathbf{z} = \mathbf{s}^{(k+1)} \tag{22}$$

$$z_1 = \mathbf{p}^{(k+1)} \tag{23}$$

$$z_2 = \boldsymbol{\theta}^{(k+1)} \tag{24}$$

Following the provided hint, the first step is to split the Jacobian into two functions, by using the Jacobian chain rule as follows:

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{z}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \tag{25}$$

Problem 7 continued on next page...

Hence, using the notation in (17)-(24), we can express the model as:

$$\text{Input Block} = \begin{cases} y_1 = \beta x_1 + F(x_2) \\ y_2 = x_2 \end{cases} \tag{26}$$

$$\text{Output Block} = \begin{cases} z_1 = y_1 \\ z_2 = y_2 + G(y_1) \end{cases} \tag{27}$$

Plugging (26)-(27) into (25), we can now state:

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \frac{\partial \mathbf{z}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial z_1}{\partial y_1} & \frac{\partial z_1}{\partial y_2} \\ \frac{\partial z_2}{\partial y_1} & \frac{\partial z_2}{\partial y_2} \end{bmatrix} \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \frac{\partial G}{\partial x_1} & I \end{bmatrix} \begin{bmatrix} \beta & \frac{\partial F}{\partial x_1} \\ 0 & I \end{bmatrix} \tag{28}$$

$$\therefore \frac{\partial \mathbf{s}^{(k+1)}}{\partial \mathbf{s}^{(k)}} = \begin{bmatrix} I & 0 \\ \frac{\partial G}{\partial x_1} & I \end{bmatrix} \begin{bmatrix} \beta & \frac{\partial F}{\partial x_1} \\ 0 & I \end{bmatrix} \tag{29}$$

Thus, using the rule that $\det(AB) = \det(A)\det(B)$, we have:

$$\det \frac{\partial \mathbf{s}^{(k+1)}}{\partial \mathbf{s}^{(k)}} = \det \begin{bmatrix} I & 0 \\ \frac{\partial G}{\partial x_1} & I \end{bmatrix} \det \begin{bmatrix} \beta & \frac{\partial F}{\partial x_1} \\ 0 & I \end{bmatrix} = I \cdot \beta^D = \beta^D \tag{30}$$