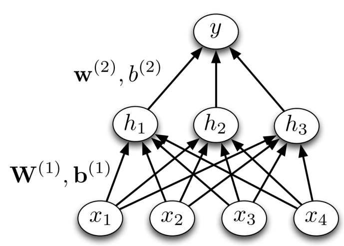
Homework 4

Due date December 6th, 2019 5:45pm

1 Problem 1

In this problem, you need to find a set of weights and biases for a multilayer perceptron which determines if a list of length 4 is in sorted order. More specifically, you receive four inputs $x_1, ..., x_4$, where $x_i \in R$ and the network must output 1 if $x_1 > x_2 > x_3 > x_4$, and 0 otherwise. You will use the following architecture:



All of the hidden units and the output unit use a hard threshold activation function:

$$\phi(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Please give a set of weights and biases for the network which correctly implements this function (including cases where some of the inputs are equal). Your answer should include:

- A weight matrix $\mathbf{W}^{(1)}$ for the hidden layer (dimension is 3×4)
- A bias vector $\mathbf{b}^{(1)}$ for hidden layer (dimension is 3)
- A 3-dimensional weight vector $\mathbf{w}^{(2)}$ for the output layer
- A scalar bias $b^{(2)}$ for the output layer

2 Problem 2

Consider a neural network with N input units, N output units, and K hidden units. The activations are computed as follows:

$$\begin{aligned}\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\ \mathbf{h} &= \sigma(\mathbf{z}) \\ \mathbf{y} &= \mathbf{x} + \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}\end{aligned}$$

where σ denotes the logistic function, applied element-wise. for given $\mathbf{r}$ and $\mathbf{s}$, the cost will involve both $\mathbf{h}$ and $\mathbf{y}$:

$$\begin{aligned}\mathcal{E} &= \mathcal{R} + \mathcal{S} \\ \mathcal{R} &= \mathbf{r}^T\mathbf{h} \\ \mathcal{S} &= \frac{1}{2}||\mathbf{y} - \mathbf{s}||^2\end{aligned}$$

- Draw the computation graph relating $\mathbf{x}$, $\mathbf{z}$, $\mathbf{h}$, $\mathbf{y}$, $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{E}$
- Derive the backprop equations for computing $\bar{\mathbf{x}} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}}$. You may use σ' to denote the derivative of the logistic function (so you don't need to write it out explicitly).
- Derive the backprop equations for computing the derivative with respect to the model parameters $\mathbf{W}$s and $\mathbf{b}$s.

3 Problem 3

You want to train the following model using gradient descent. Here, the input x and target t are both scalar-valued.

$$\begin{aligned}z &= w_0 + w_1 x + w_2 x^2 + w_3 x^3 \\ y &= 1 + e^z \\ \mathcal{L} &= \frac{1}{2}(\log y - \log t)^2\end{aligned}$$

Determine the backprop rules which will let you compute the loss derivative $\frac{\partial \mathcal{L}}{\partial w_2}$.

Your equations should refer to previously computed values (e.g. your formula for $\bar{z}$ should be a function of $\bar{y}$).

4 Problem 4

In this question, you are asked to derive the Backprop Through Time equations for the univariate version of the LSTM architecture. Note: This question is important context for understanding LSTMs, but it is just ordinary Backprop question.

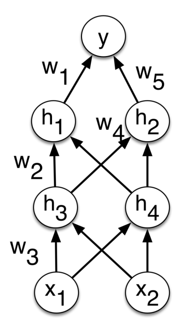
For completeness, (You can find them in lecture slides), the LSTM do the following calculations:

$$
\begin{aligned}
i^{(t)} &= \sigma(w_{ix}x^{(t)} + w_{ih}h^{(t-1)}) \\
f^{(t)} &= \sigma(w_{fx}x^{(t)} + w_{fh}h^{(t-1)}) \\
o^{(t)} &= \sigma(w_{ox}x^{(t)} + w_{oh}h^{(t-1)}) \\
g^{(t)} &= \tanh(w_{fg}x^{(t)} + w_{gh}h^{(t-1)}) \\
c^{(t)} &= f^{(t)}c^{(t-1)} + i^{(t)}g^{(t)} \\
h^{(t)} &= o^{(t)}\tanh(c^{(t)})
\end{aligned}
$$

- Derive the Backprop Through Time equations for the activations and the gates ($\overline{i^{(t)}}$,$\overline{f^{(t)}}$,$\overline{o^{(t)}}$,$\overline{g^{(t)}}$,$\overline{c^{(t)}}$,$\overline{h^{(t)}}$)
- Derive backprop for $\bar{w_{ix}}$
- Based on your answers above, explain why the gradient doesn't explode if the values of the input and output gates are very close to 0 and the values of the forget gates are very close to 1. (You answer may involve both $\overline{c^{(t)}}$ and $\overline{h^{(t)}}$)

5 Problem 5

One of the interesting features of the ReLU activation function is that it sparsifies the activations and the derivatives, i.e. sets a large fraction of the values to zero for any given input vector. Consider the following network:



Note that each w_i refers to the weight on a single connection, not the whole layer. Suppose we are trying to minimize a loss function $\mathcal{L}$ which depends only on the activation of the output unit y. (For instance, $\mathcal{L}$ could be the squared error loss $\frac{1}{2}(y-t)^2$.) Suppose the unit h_1 receives an input of -1 on a particular training case, so the ReLU evaluates to 0. Based only on this information, which of the weight derivatives

$$\frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \frac{\partial \mathcal{L}}{\partial w_3}$$

are guaranteed to be 0 for this training case? Write YES or NO for each. Justify your answers.

6 Problem 6

For this question, you need to open the attached notebook and follow the instructions.