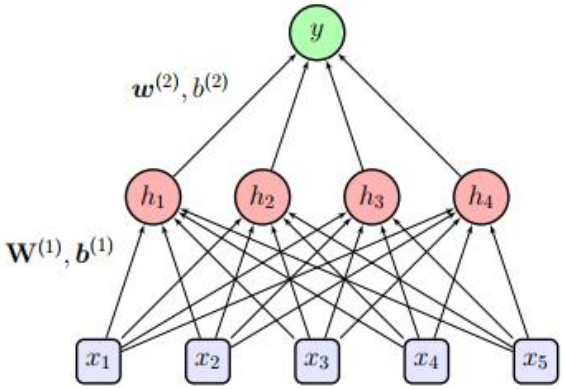
Homework Solns

Problem 1 [20 pts]: In this problem you will find a set of weights and biases for a multilayer perceptron which determine if a list of four numbers in the descending order. More specifically, you receive five inputs x_1, x_2, x_3, x_4 and x_5, where $x_i \in R$, and the network must output 1 if $x_1 > x_2 > x_3 > x_4 > x_5$ and 0 otherwise. You will use the following Multilayer Perceptron (MLP) architecture consisting of one input layer with five nodes, one hidden layer with four nodes one output layer with one node.



The activation function of the hidden units and the output unit can be assumed to be a hard threshold function.

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

Find a set of weights and biases for the network which correctly implements this function (including cases where some of the inputs are equal). Note, in some cases, if some of the inputs are equal to each other (say $x_2 = x_3$), your output should be 0. Your answer should include:

1. A 4×5 weight matrix $\mathbf{W}^{(1)}$ for the hidden layer.
2. A 4-dimensional vector of biases $\boldsymbol{b}^{(1)}$ for the hidden layer.
3. A 4-dimensional vector of weights $\boldsymbol{w}^{(2)}$ for the output layer.
4. A scalar bias $b^{(2)}$ for the output layer.

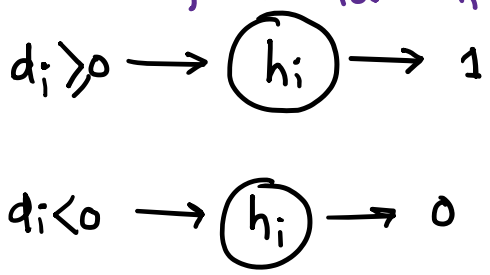
Soln: We need $x_1 > x_2 > x_3 > x_4 > x_5$, $x_i \in$ real numbers

One way of ensuring $x_i > x_{i+1}$

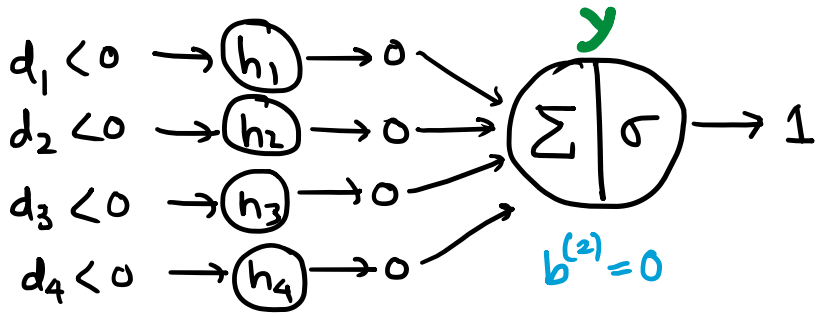
is to ensure that $d_i = x_i - x_{i+1} > 0$, $i = 1, 2, 3, 4$

So we are given 4 perceptrons in the hidden layer, and each of them could be fed with one difference d_i. We could expect the perceptron h_i to fire only if $d_i > 0$, however, given our output activation fires even when $d_i = 0$, this way will not work when $x_i = x_{i+1}$ $(d_i = 0)$.

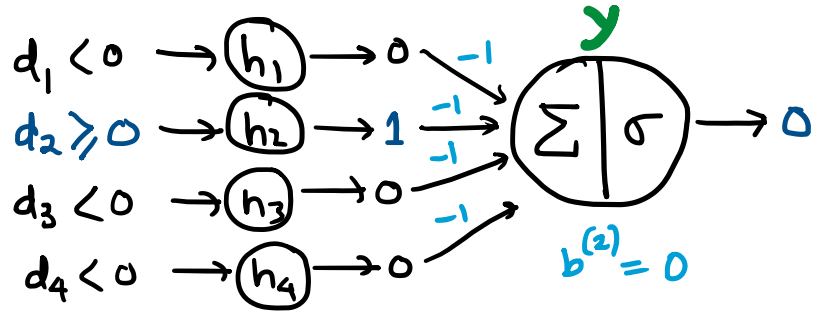
So consider $d_i = x_{i+1} - x_i$. For this, when $d_i < 0$, the output would be zero, and for $d_i \geq 0$, the output would be 1



If all $d_i < 0$, $i = 1, 2, 3, 4$, then the output of hidden activations $h_i = 0 \; \forall \; i$, and hence the output would be set to 1



However, if any $d_i \geq 0$, then the output of the corresponding hidden unit would be 1, i.e. $h_i = 1$. To ensure that the output activation y does not fire when any of the h_i's becomes 1, one could set the weights of the second layer to some negative value say $w_i^{(2)} = -1$



As for the weights of the first layer, we can define individual differences d_i as follows

(1) The weight matrix for the first layer is

$$\underset{\sim}{W}^{(1)} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

(2) The bias vector for first layer can be all zeros

$$\underset{\sim}{b}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(3) Weight vector for 2nd layer $\quad \underset{\sim}{w}^{(2)} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$

(4) Bias for 2nd layer, $b^{(2)} = 0$

Note: There can be many possible solutions of this problem!

Marking scheme: If your model
- A) Works for $x_1 \geq x_2 \geq x_3 \geq x_4 \geq x_5$ → 10/20
- B) Works for $x_1 > x_2 > x_3 > x_4 > x_5$ → 20/20

Problem 2: **[20 pts]** Consider a linear combination of M input variables $x_1, ..., x_D$ in the form

$$z_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

where $j = 1, \ldots, M$, resulting in the pre-activation z_j and the superscript (1) indicates that the corresponding parameters are in the first 'layer' of the network. Also consider

$$h_j = g(z_j)$$

where $g(\cdot)$ represents some differentiable, nonlinear activation function. The outputs of the activation functions are linearly combined to obtain the output pre-activation

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} h_j + w_{k0}^{(2)}$$

where $k = 1, \ldots, K$, and k is the total number of outputs. This transformation corresponds to the second layer of the network. Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs y_k. The choice of activation function is determined by the nature of the data and the assumed distribution of target variables. Thus, for standard regression problems, the activation function is the identity so that $y_k = a_k$. Similarly, for multiple binary classification problems, each output unit activation is transformed using a logistic sigmoid function so that

$$y_k = \sigma(a_k)$$

where,

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

Finally, for multiclass problems, a SoftMax activation function is used. We can combine these various stages to give the overall network function that, for sigmoidal (logistic) output unit activation functions, takes the form

$$y_k(x; w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} g\left(\sum_{j=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right)$$

where the set of all weight and bias parameters have been grouped together into a vector $\boldsymbol{w}$. Thus, the neural network model is simply a nonlinear function from a set of input variables x_i to a set of output variables y_k controlled by a vector $\boldsymbol{w}$ of adjustable parameters. Considering a two-layer neural network as shown above with logistic sigmoid activation function, show that there exists an equivalent network,

which computes the same function, but with hidden unit activation functions given by $\texttt{tanh}(a)$ where the $\texttt{tanh}(a)$ is defined as follows:

$$\texttt{tanh}(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

Soln: Before showing how two networks (one with sigmoid activation at hidden layer, and another with tanh activation at hidden layer) are similar, establish the relation between sigmoid & tanh

Relation betn sigmoid & tanh:

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{1}{1+e^{-\left(\frac{x}{2}+\frac{x}{2}\right)}}$$

$$= \frac{1}{1+e^{-x/2}\cdot e^{-x/2}}$$

$$= \frac{e^{x/2}}{e^{x/2}+e^{-x/2}}$$

$$= \frac{e^{x/2}-e^{-x/2}+e^{-x/2}}{e^{x/2}+e^{-x/2}}$$

$$= \frac{e^{x/2}-e^{-x/2}}{e^{x/2}+e^{-x/2}} + \frac{e^{-x/2}}{e^{x/2}+e^{-x/2}}$$

$$= \tanh\left(\frac{x}{2}\right) + \frac{1}{1+e^{x}}$$

$$= \tanh\left(\frac{x}{2}\right) + \left(\frac{1}{1+e^{x}} - 1\right) + 1$$

$$= \tanh\left(\frac{x}{2}\right) + \frac{1-1-e^{x}}{1+e^{x}} + 1$$

$$= \tanh\left(\frac{x}{2}\right) - \frac{e^{x}}{1+e^{x}} + 1$$

$$= \tanh\left(\frac{x}{2}\right) - \frac{1}{1+e^{-x}} + 1$$

$$= \tanh\left(\frac{x}{2}\right) - \sigma(x) + 1$$

$$\Rightarrow \quad \sigma(x) = \tanh\left(\frac{x}{2}\right) - \sigma(x) + 1$$

$$\Rightarrow \quad 2\sigma(x) = \tanh\left(\frac{x}{2}\right) + 1$$

$$\Rightarrow \quad \boxed{\sigma(x) = \frac{1}{2}\tanh\left(\frac{x}{2}\right) + \frac{1}{2}}$$

Now lets look at the two networks

$$y_k(\underset{\sim}{x}; \underset{\sim}{w}) = \sigma\left(\sum_j w_{kj}^{(2)}\, g\left(\underbrace{\sum_i w_{ji}^{(1)} x_i + w_{j0}^{(1)}}_{z_j}\right) + w_{k0}^{(2)}\right)$$

for sigmoid activation $g \equiv \sigma(\cdot)$

$$y_k(\underset{\sim}{x}; \underset{\sim}{w}) = \sigma\left(\sum_j w_{kj}^{(2)}\, \sigma(z_j) + w_{k0}^{(2)}\right)$$

For tanh activation, $g = \tanh(\cdot)$ [substitute the relation of $\sigma(\cdot)$ & $\tanh(\cdot)$]

$$y_k(\underset{\sim}{x}; \underset{\sim}{w}) = \sigma\left(\sum_j w_{kj}^{(2)}\left(\frac{1}{2}\tanh\left(\frac{z_j}{2}\right) + \frac{1}{2}\right) + w_{k0}^{(2)}\right)$$

$$= \sigma\left(\sum_j \frac{w_{kj}^{(2)}}{2}\tanh\left(\frac{z_j}{2}\right) + \sum_j \frac{w_{kj}^{(2)}}{2} + w_{k0}^{(2)}\right)$$

$$= \sigma\left(\sum_j \underbrace{\frac{w_{kj}^{(2)}}{2}}_{w_{kj}^{(2)*}} \tanh\left(\sum_i \underbrace{\frac{w_{ji}^{(1)}}{2}}_{w_{ji}^{(1)*}} x_i + \underbrace{\frac{w_{j0}^{(1)}}{2}}_{w_{j0}^{(1)*}}\right) + \underbrace{\sum_j \frac{w_{kj}^{(2)}}{2} + w_{k0}^{(2)}}_{w_{k0}^{(2)*}}\right)$$

Parameters of equivalent network

EQUIVALENT Network with tanh hidden activation will have

$$y_k(\underset{\sim}{x}; \underset{\sim}{w}) = \sigma\left(\sum_j w_{kj}^{(2)*} \tanh\left(\sum_i w_{ji}^{(1)*} x_i + w_{j0}^{(1)*}\right) + w_{k0}^{(2)*}\right)$$

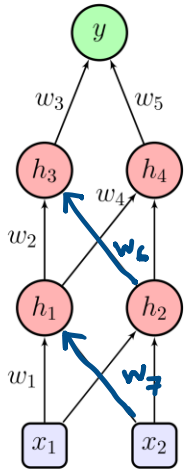
where,

$$w_{kj}^{(2)*} = \frac{w_{kj}^{(2)}}{2}, \quad w_{k0}^{(2)*} = w_{k0}^{(2)} + \sum_j \frac{w_{kj}^{(2)}}{2}, \quad w_{ji}^{(1)*} = \frac{w_{ji}^{(1)}}{2}, \quad w_{j0}^{(1)*} = \frac{w_{j0}^{(1)}}{2}$$

Marking scheme: 8 points for relation between $\sigma(\cdot)$ & $\tanh(\cdot)$

3 points each for $w_{ji}^{(1)*}$, $w_{j0}^{(1)*}$, $w_{kj}^{(2)*}$, $w_{k0}^{(2)*}$

Problem 3: **[10 pts]** 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

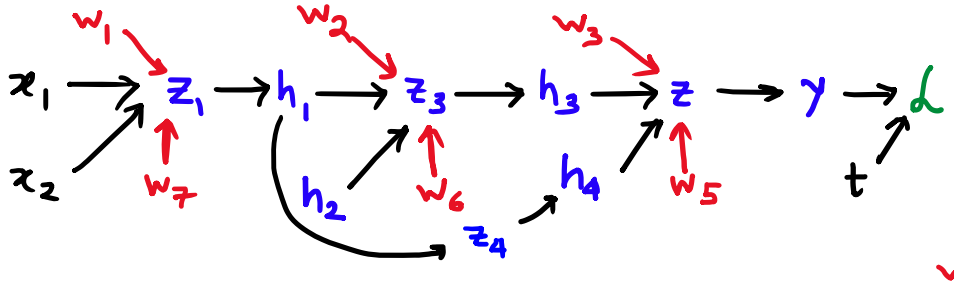
$$\frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \frac{\partial \mathcal{L}}{\partial w_3}$$

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

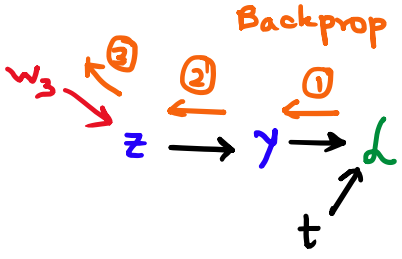
Soln: You may wish to define weights for two additional connections for convenience, as shown in figure (above).

Note: Each neuron is associated with a ReLU activation

So we could draw a partial computation graph (not necessary)



(a) $\frac{\partial \mathcal{L}}{\partial w_3} = \underbrace{\frac{\partial \mathcal{L}}{\partial y}}_{①} \times \underbrace{\frac{\partial y}{\partial z}}_{②} \times \underbrace{\frac{\partial z}{\partial w_3}}_{③}$

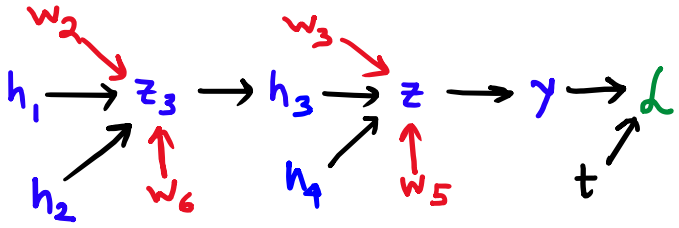


During forward computation, the value of $z_1 = -1$ and $h_1 = 0$
All other variables need not be zero.

- $\frac{\partial \mathcal{L}}{\partial y}$ need not be 0
- $\frac{\partial y}{\partial z} = \frac{\partial}{\partial z} \text{ReLU}(z) = \frac{\partial}{\partial z} \max(0, z)$ need not be 0
- $\frac{\partial z}{\partial w_3} = \frac{\partial}{\partial w_3}(w_3 h_3 + w_5 h_4) = h_3$ ← need not be 0

$\therefore$ $\frac{\partial \mathcal{L}}{\partial w_3}$ is not guaranteed to be zero

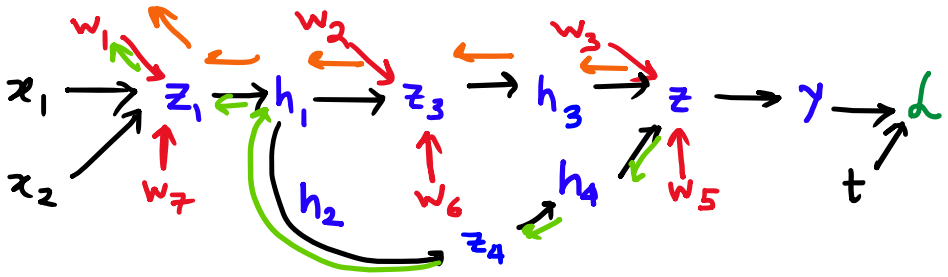
(b) $\frac{\partial \mathcal{L}}{\partial w_2} = \frac{\partial \mathcal{L}}{\partial z} \times \frac{\partial z}{\partial h_3} \times \frac{\partial h_3}{\partial z_3} \times \frac{\partial z_3}{\partial w_2}$



- $\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial y} \times \frac{\partial y}{\partial z}$ already proved to not necessarily zero
- $\frac{\partial z}{\partial h_3} = \frac{\partial}{\partial h_3}(w_3 h_3 + w_5 h_4) = w_3$ ← need not be 0
- $\frac{\partial h_3}{\partial z_3} = \frac{\partial}{\partial z_3} \text{ReLU}(z_3) = \frac{\partial}{\partial z_3} \max(0, z_3)$ ← need not be 0
- $\frac{\partial z_3}{\partial w_2} = \frac{\partial}{\partial w_2}(w_2 h_1 + w_6 h_2) = h_1 = 0$ (given)

$\therefore$ The product $\frac{\partial \mathcal{L}}{\partial w_2} = 0$ is guaranteed to be zero

(c)



$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial h_3} \times \frac{\partial h_3}{\partial z_3} \times \frac{\partial z_3}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$+ \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial h_4} \times \frac{\partial h_4}{\partial z_4} \times \frac{\partial z_4}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$= \underbrace{\frac{\partial L}{\partial z}}_{A} \times \left(\underbrace{\frac{\partial z}{\partial h_3} \times \frac{\partial h_3}{\partial z_3} \times \frac{\partial z_3}{\partial h_1}}_{B_1} + \underbrace{\frac{\partial z}{\partial h_4} \times \frac{\partial h_4}{\partial z_4} \times \frac{\partial z_4}{\partial h_1}}_{B_2} \right) \times \underbrace{\frac{\partial h_1}{\partial z_1}}_{C} \times \underbrace{\frac{\partial z_1}{\partial w_1}}_{D}$$

(A) $\frac{\partial L}{\partial z}$ ← need not be zero (proved previously)

(B_1)
- $\frac{\partial z}{\partial h_3} = \frac{\partial}{\partial h_3}(w_3 h_3 + w_5 h_4) = w_3$ ← need not be 0
- $\frac{\partial h_3}{\partial z_3} = \frac{\partial}{\partial z_3} \max(0, z_3)$ ← need not be 0
- $\frac{\partial z_3}{\partial h_1} = \frac{\partial}{\partial h_1}(w_2 h_1 + w_3 h_2) = w_2$ ← need not be 0

(B_2) Similarly you can check that $\frac{\partial z}{\partial h_4}$, $\frac{\partial h_4}{\partial z_4}$, $\frac{\partial z_4}{\partial h_1}$ need not be 0

(C) $\frac{\partial h_1}{\partial z_1} = \frac{\partial}{\partial z_1} \max(0, z_1) = \frac{\partial}{\partial z_1} \max(0, -1) = \frac{\partial}{\partial z_1} 0 = 0$

So the product turns out to be 0 because of $\frac{\partial h_1}{\partial z_1}$ being 0

$\therefore \frac{\partial L}{\partial w_1}$ is guaranteed to be 0

Problem 4: **[20 pts]** Consider a MLP consisting of N input units, K hidden units and N output units. The activations are computed as follows:

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

where $\sigma(\cdot)$ is the sigmoid function, applied element wise. The loss $\mathcal{L}$ will involve both $\boldsymbol{h}$ and $\boldsymbol{y}$:

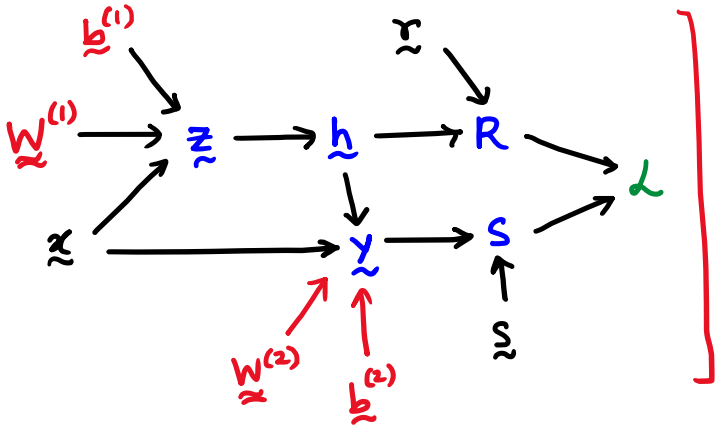
$$\begin{aligned}
\mathcal{L} &= \mathcal{S} + \mathcal{R} \\
\mathcal{S} &= \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{s}\|_2^2 = \frac{1}{2}(\underset{\sim}{y}-\underset{\sim}{s})^T(\underset{\sim}{y}-\underset{\sim}{s}) \\
\mathcal{R} &= \boldsymbol{r}^T\boldsymbol{h}
\end{aligned}$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are given vectors.

1. Draw the computation graph relating $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{h}$, $\boldsymbol{y}$, $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{L}$.
2. Derive the backpropagation equations for computing $\bar{\boldsymbol{x}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}}$. You may use σ' to denote the derivation of the sigmoid function, so you do not have to write it out explicitly.

Solu:

1) Computation graph →



6

2) $\bar{\mathcal{L}} = \frac{\partial \mathcal{L}}{\partial \mathcal{L}} = 1$, $\bar{\mathcal{R}} = \frac{\partial \mathcal{L}}{\partial \mathcal{R}} = 1$, $\bar{\mathcal{S}} = \frac{\partial \mathcal{L}}{\partial \mathcal{S}} = 1$ ③

$$\bar{\underset{\sim}{y}} = \frac{\partial \mathcal{L}}{\partial \underset{\sim}{y}} = \bar{\mathcal{S}}\frac{\partial \mathcal{S}}{\partial \underset{\sim}{y}} = \bar{\mathcal{S}}\begin{bmatrix} \partial \mathcal{S}/\partial y_1 \\ \vdots \\ \partial \mathcal{S}/\partial y_n \end{bmatrix} = \bar{\mathcal{S}}\begin{bmatrix} y_1 - s_1 \\ \vdots \\ y_n - s_n \end{bmatrix} = \bar{\mathcal{S}}\cdot(\underset{\sim}{y}-\underset{\sim}{s})$$ ②

$$\bar{h} = \frac{\partial \mathcal{L}}{\partial h} = \frac{\partial R}{\partial h}\bar{R} + \left(\frac{\partial y}{\partial h}\right)^T \bar{y}$$

$$= \begin{bmatrix} \partial R/\partial h_1 \\ \vdots \\ \partial R/\partial h_K \end{bmatrix} \bar{R} + \begin{bmatrix} w_{11}^{(2)} & \cdots & w_{1K}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{n1}^{(2)} & \cdots & w_{nK}^{(2)} \end{bmatrix}^T \bar{y}$$

$$= \begin{bmatrix} r_1 \\ \vdots \\ r_K \end{bmatrix} \bar{R} + \begin{bmatrix} w_{11}^{(2)} & \cdots & w_{1K}^{(2)} \\ \vdots & \ddots & \vdots \\ w_{n1}^{(2)} & \cdots & w_{nK}^{(2)} \end{bmatrix}^T \bar{y}$$

$$= r\bar{R} + W^{(2)T}\bar{y} \quad \text{④}$$

$$\bar{z} = \frac{\partial \mathcal{L}}{\partial z} = \left(\frac{\partial h}{\partial z}\right)^T \bar{h} = \begin{bmatrix} \frac{\partial h_1}{\partial z_1} & \cdots & \frac{\partial h_K}{\partial z_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_K}{\partial z_1} & \cdots & \frac{\partial h_K}{\partial z_K} \end{bmatrix}^T \bar{h} = \begin{bmatrix} \sigma'(z_1) & & 0 \\ & \ddots & \\ 0 & & \sigma'(z_K) \end{bmatrix}^T \bar{h}$$

$$= \bar{h} \circ \sigma'(z) \quad \text{②}$$

elementwise product

$$\bar{x} = \frac{\partial \mathcal{L}}{\partial x} = \left(\frac{\partial z}{\partial x}\right)^T \bar{z} + \left(\frac{\partial y}{\partial x}\right)^T \bar{y}$$

identity matrix

$$= W^{(1)T}\bar{z} + I\bar{y}$$

$$= W^{(1)T}\bar{z} + \bar{y} \quad \text{③}$$