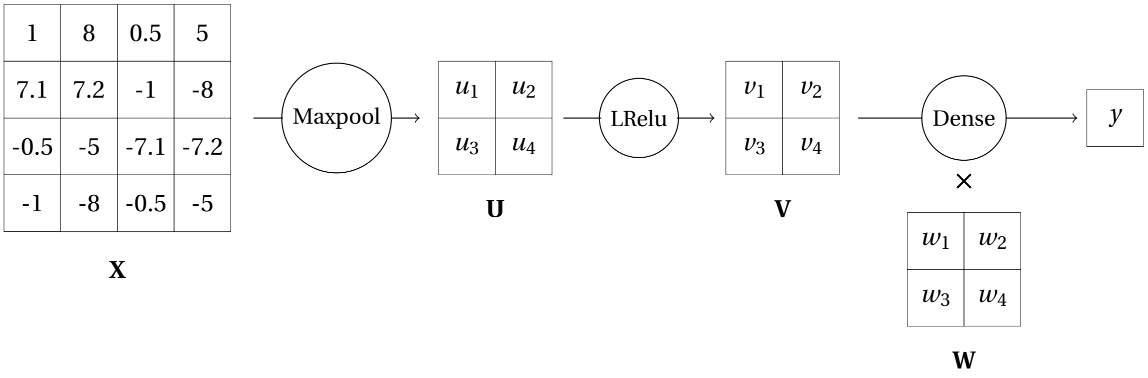
Assignment 2: EN4553 (Machine Vision)

University of Moratuwa

December 22, 2022

1. (25 points) Consider the following toy neural network that has three operations: `Maxpool`, `LRelu`, and `Dense`. The `Maxpool` operation is 2×2 max-pooling with stride 2 (i.e., non-overlapping). The `LRelu` operation is a non-linear activation function that is applied element-wise (the exact form of the function is given below). The `Dense` operation is a usual fully-connected layer with the shown parameter matrix $\mathbf{W}$. Therefore, $y = \sum_{i=1}^{4} v_i w_i$.

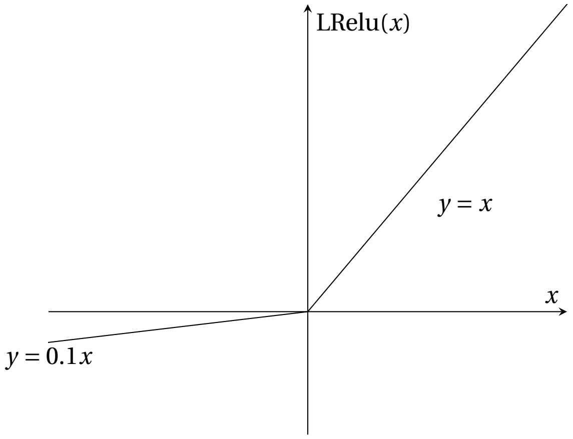


1	8	0.5	5
7.1	7.2	-1	-8
-0.5	-5	-7.1	-7.2
-1	-8	-0.5	-5

The `LRelu` (leaky rectified linear unit)operation is defined by:

$$\text{LRelu}(x) = \begin{cases} x & \text{if x} \geq 0, \\ 0.1x & \text{otherwise.} \end{cases}$$

A plot of the `LRelu`(.) function is shown below.



The parameter matrix $\mathbf{W}$ is initialized as $[w_1 = 1, w_2 = 2, w_3 = 10, w_4 = 4]$.

Your training dataset is $\{(\mathbf{X_1}, t_1), (\mathbf{X_2}, t_2), \ldots, (\mathbf{X_N}, t_\mathrm{N})\}$. For any given training datum $(\mathbf{X}_i, t_i)$, the matrix $\mathbf{X}_i \in \mathbb{R}^{4\times4}$ is the input fed to the neural network above and $t_i \in \mathbb{R}$ is the groundtruth value, which we use as the target for the neural net output y.

(a) (3 points) Considering that this is a regression problem, and that we use only one datum at a time during training (SGD with mini-batch size 1), write down a suitable loss function in terms of the target variable t and the neural network output y.

(b) **Forward Pass**: The first training datum you pick after initializing **W** with the given values happens to have the **X** input matrix derived from you index number as follows: starting from the top-left corner and proceeding to right and then down, write down each digit in your index number, one digit per cell. If there are repeated digits in your index number, add 0.1, 0.2, or 0.3 to each repeating digit to make the values in the first six cells unique. Replace any occurrences of 0 with 0.5, 0.6, etc. Then repeat the same sequence in the next 6 cells but each digit multiplied by -1. Then repeat the first row in the last row, but multiplied by -1. An example has been done for you assuming that your index number is 180577X.

You may copy the diagram to your answer sheet and show the answers there. Remember to replace **X** with values from your index number.

i. (3 points) By performing the `Maxpool` operation, find the value of the matrix **U**.

ii. (2 points) Work out the value of **V** by sending **U** through the LRelu(.) activation function.

iii. (2 points) Find the network output y by performing the `Dense` operation.

(c) **Backward Pass**: Assume that, after the forward pass step above, you calculate $\frac{\partial \mathrm{L}}{\partial y}$, the derivative of the loss function with respect to the output, to be 2.0 for your training datum.

i. (2 points) Show that, the loss derivative with respect to the neuron v_1 in the above diagram, is given by:

$$\frac{\partial \mathrm{L}}{\partial v_1} = \frac{\partial \mathrm{L}}{\partial y} w_1 = 2 \times 1.$$

ii. (2 points) Complete the following matrix (copy it to your answer sheet):

$$\frac{\partial \mathrm{L}}{\partial \mathbf{V}} = \begin{array}{|c|c|}\hline \frac{\partial \mathrm{L}}{\partial v_1} = 2.0 & \frac{\partial \mathrm{L}}{\partial v_2} = ? \\ \hline \frac{\partial \mathrm{L}}{\partial v_3} = ? & \frac{\partial \mathrm{L}}{\partial v_4} = ? \\ \hline \end{array}$$

iii. (3 points) Similarly, work out $\frac{\partial \mathrm{L}}{\partial \mathbf{U}}$ (a 2×2 matrix) by backpropagating $\frac{\partial \mathrm{L}}{\partial \mathbf{V}}$ through the LRelu(.) function.

iv. (3 points) Backpropagate through `Maxpool` to find $\frac{\partial \mathrm{L}}{\partial \mathbf{X}}$, the loss derivative w.r.t. the input (a 4 × 4 matrix).

(d) Instead of the LRelu(.) function, you decide to use the PRelu(.) function (parametric rectified linear unit), which is defined by:

$$\mathrm{PRelu}(x) = \begin{cases} x & \text{if x} \geq 0, \\ \alpha x & \text{otherwise,} \end{cases}$$

where the coefficient $\alpha > 0$ is learned with backpropagation and SGD. As usual, PRelu(.) non-linearity is applied element-wise. Consider a training datum where the matrix **U** takes the following value during the forward pass,

$$\underset{\mathbf{U}}{\begin{array}{|c|c|}\hline 14 & -4 \\ \hline -6 & 3 \\ \hline \end{array}} \xrightarrow{\boxed{\text{PRelu}}} \underset{\mathbf{V}}{\begin{array}{|c|c|}\hline v_1 & v_2 \\ \hline v_3 & v_4 \\ \hline \end{array}}$$

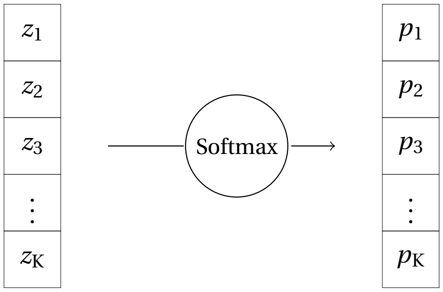
i. (3 points) Assuming that the current value of the α parameter is α_0, find out the entries of matrix **V** in terms of α_0.

ii. (2 points) For this training datum, during backpropagation, $\frac{\partial \mathrm{L}}{\partial \mathbf{V}}$ takes the following value:

$$\frac{\partial \mathrm{L}}{\partial \mathbf{V}} = \begin{array}{|c|c|}\hline \frac{\partial \mathrm{L}}{\partial v_1} = 2.0 & \frac{\partial \mathrm{L}}{\partial v_2} = -2.5 \\ \hline \frac{\partial \mathrm{L}}{\partial v_3} = 1.5 & \frac{\partial \mathrm{L}}{\partial v_4} = 3.0 \\ \hline \end{array}$$

Calculate $\left(\frac{\partial \mathrm{L}}{\partial \alpha}\right)_{\alpha_0}$.

2. (15 points) We analyze the softmax function in this question. Let $\mathbf{z} = [z_1, z_2, \ldots, z_\mathrm{K}]^\mathrm{T}$ be the input logit vector and $\mathbf{p} = [p_1, p_2, \ldots, p_\mathrm{K}]^\mathrm{T}$ be the output probabilities vector, where K is the number of classes. Therefore, $\mathbf{p} = \mathrm{softmax}(\mathbf{z})$. We use the notation v_i to denote the i-th element of a vector $\mathbf{v}$.



Recall that, from the definition of the softmax function,

$$p_i = \frac{\exp(z_i)}{\sum_{j=1}^{\mathrm{K}} \exp(z_j)}, \tag{1}$$

for all $i \in \{1, 2, \ldots, \mathrm{K}\}$.

(a) Let $\tilde{\mathbf{z}} = \mathbf{z} - \mathbf{1_K} \times \max\{z_1, z_2, \ldots, z_\mathrm{K}\}$, where $\mathbf{1_K}$ is a K-dimensional column vector of ones. That is, $\tilde{\mathbf{z}}$ the vector obtained by element-wise subtraction of the maximum element of $\mathbf{z}$ from $\mathbf{z}$.

i. (2 points) Show that: $\mathrm{softmax}(\mathbf{z}) = \mathrm{softmax}(\tilde{\mathbf{z}})$.

ii. (2 points) Assume that you are implementing a brand new deep learning library from scratch. Explain why it is a good idea to do the above transformation before using Eq.(1) when implementing the softmax function for your library. (Note: most existing library implementations of the softmax function already includes this transformation.)

(b) The *cross-entropy* distance measure between two discrete probability distributions with probability mass functions u and v with the same support $\mathscr{X}$ is given by:

$$\mathrm{H}(u, v) = -\sum_{x \in \mathscr{X}} u(x) \log v(x). \tag{2}$$

i. (3 points) The groundtruth label for the above data point says it belongs to class y, where $1 \leq y \leq \mathrm{K}$. If we use the cross-entropy "distance" between the groundtruth probability distribution $\mathbf{g}$ and the predicted probability distribution $\mathbf{p}$ as our loss function L(.,.), show that,

$$\mathrm{L}(\mathbf{g}, \mathbf{p}) = -\log p_y. \tag{3}$$

Hint: Write down the groundtruth probability distribution as a one-hot vector.

ii. (2 points) Explain why minimizing the above loss function during training intuitively makes sense.

(c) We now calculate the derivative of the above loss function w. r. t. the logit vector $\mathbf{z}$.

i. (3 points) Show that $d\mathrm{L}/dz_y = -1 + p_y$.

ii. (3 points) Show that $d\mathrm{L}/dz_{y'} = p_{y'}$, for all $y' \in \{1, 2, \ldots, \mathrm{K}\} \setminus \{y\}$.

Hint: Combine Eq. (3) and Eq. (1), and work out the derivatives.