CS-E4850 Computer Vision

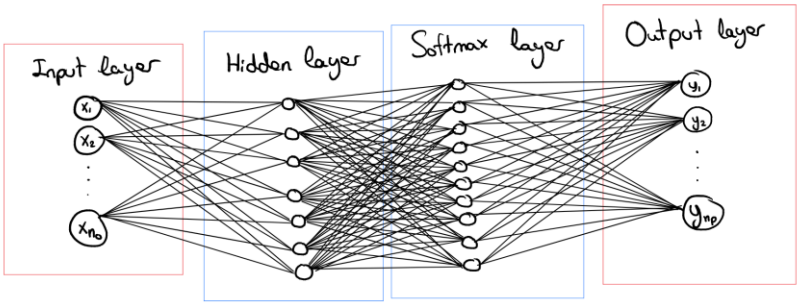
Exercise Round 9

Laura Alejandra Encinar Gonzalez

101583950

Exercise 1

$L = 2, \quad n_1 = 7, \quad n_2 = 10$



1.

Eq(1) $\quad E = \frac{1}{m} \sum_{j=1}^{m} -t_j \cdot \log(y_j)$

$m=1 \Rightarrow \quad \boxed{E = -t_1 \cdot \log(y_1)}$

2.

$$\frac{\partial E}{\partial z^{(2)}} = \left(y^{(2)} - t\right)^T$$

$$\frac{\partial E}{\partial z^{(2)}} = \frac{\partial E}{\partial y^{(2)}} \frac{\partial y^{(2)}}{\partial z^{(2)}}$$

$$y^{(2)} = S(z^{(2)}) \qquad S\left(z_i^{(2)}\right) = \frac{e^{z_i^{(2)}}}{\sum_{k=1}^{10} e^{z_k^{(2)}}} \qquad \sum_{j=1}^{m} t_j = 1$$

→ Differenciate softmax function $S(z^{(2)})$

- if $i=j$:

$$\frac{\frac{\partial e^{z_i}}{\partial z_j}}{\sum_{k=1}^{n} e^{z_k}} = \frac{e^{z_i} \Sigma - e^{z_j} e^{z_i}}{\Sigma^2} = \frac{e^{z_i}}{\Sigma} \cdot \frac{\Sigma - e^{z_j}}{\Sigma} = S_i(1 - S_j)$$

$$\frac{\partial S_i}{\partial z_j} = S_i(1 - S_j) = y_i \cdot (1 - y_j)$$

if $i \neq j$:

$$\frac{\frac{\partial e^{z_i}}{\partial z_j}}{\sum_{k=1}^{n} e^{z_k}} = \frac{0 - e^{z_j} e^{z_i}}{\Sigma^2} = \frac{-e^{z_j}}{\Sigma} \cdot \frac{e^{z_i}}{\Sigma} = -S_j \cdot S_i$$

$$\Rightarrow \frac{\partial S_i}{\partial z_j} = -S_j S_i = -y_j \cdot y_i$$

Exercise 2

```
% This is the only function that you're expected to change. Right now, it just returns a lot of zeros
[hid_input, hid_output, class_input, log_class_prob, class_prob] = forward_pass(model, data);

% Compute ∂E/∂z^(2)
dE_dz2 = (class_prob-data.targets)';

% Compute ∂E/∂y^(1)
dE_dy1 = (dE_dz2 * model.hid_to_class);

% Compute ∂E/∂z^(1)
dE_dz1 = dE_dy1 * diag(hid_output .* (1 - hid_output));

% Compute ∂E/∂W^(2)
dE_dW2 = dE_dz2' * hid_output';

% Compute ∂E/∂W^(1)
dE_dW1 = dE_dz1' * data.inputs';

% Assign the computed gradients to the return structure
ret.input_to_hid = dE_dW1;
ret.hid_to_class = dE_dW2;
```