CSCI3230 Fundamentals of Artificial Intelligence

# Written Assignment 1

Jonathan Lam – 1155093597

## Definition

For eligibility, a symbol is denoted by

$$N_{n^{th}node}^{layer}$$

In other words, comparing to defined symbols in the assignment,

$$w_{i,j,k} = w_{j,k}^{i}$$
$$N_{i,j} = N_{j}^{i}$$
$$h_{i,j} = h_{j}^{i}$$

## Question 1a

$$O = f(\sum_{j=0}^{n} W_j I_j)$$

## Question 1b

$$h_k^i = f(\sum_{j=1}^{H_{i-1}} w_{j,k}^{i-1} h_j^{i-1} + 1) \tag{1}$$

$$O_m = h_k^{K+1} = f(\sum_{j=1}^{H_K} w_{j,k}^{K} h_j^{K} + 1) \tag{2}$$

## Question 1c

$$
\begin{aligned}
f'(z) &= \frac{d(1+e^{-z})^{-1}}{dz} \\
&= -(1+e^{-z})^{-2} \cdot \frac{d(1+e^{-z})}{dz} \\
&= -(1+e^{-z})^{-2} \cdot \frac{de^{-z}}{dz} \\
&= -(1+e^{-z})^{-2}(-e^{-z}) \\
&= \frac{1}{1+e^{-z}} \cdot \frac{(1+e^{-z})-1}{1+e^{-z}} \\
&= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}}\right)
\end{aligned}
$$

$$
f'(z) = f(z)(1-f(z)) \tag{3}
$$

## Question 1d

The learning rate determines how "big" a step (i.e. how much to change the model) when doing gradient decent. A smaller $\alpha$ means each step is smaller and more accurate towards the local minima, but it requires more time to compute/converge; whereas a larger $\alpha$ means faster processing, but prone to overshooting or even fail to reach the local minima.

# Question 1e

Let $in_k^{i+1}$ be the weighted sum of inputs to unit $k$ in layer $L_{i+1}$, where

$$in_k^{i+1} = \sum_{j=1}^{H_{i-1}} w_{j,k}^i h_j^i + 1 \tag{4}$$

In the layer $L_{i+1}$, since $E \leftarrow h_k^{i+1} \leftarrow in_k^{i+1} \leftarrow w_{j,k}^i$ & $h_j^i$

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial h_k^{i+1}} \cdot \frac{\partial h_k^{i+1}}{\partial in_k^{i+1}} \cdot \frac{\partial in_k^{i+1}}{\partial X} \tag{5}$$

where $X$ is $w_{j,k}^i$ or $h_j^i$ (or bias $b_{j,k}^i$).

## (i)

From (5), substituting $X = w_{j,k}^i$ and $i \leftarrow K$,

$$\frac{\partial E}{\partial w_{j,k}^K} = \frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial in_k^{K+1}} \cdot \frac{\partial in_k^{K+1}}{\partial w_{j,k}^K} \tag{6}$$

From the error term $E = \frac{1}{2}\sum_{m=1}^{H_{K+1}}(O_m - T_m)^2$,

$$\frac{\partial E}{\partial O_k} = \frac{\partial}{\partial O_k}\left(\frac{1}{2}\sum_{k=1}^{H_{K+1}}(O_k - T_k)^2\right) = O_k - T_k$$

From (2) & (3),

$$\begin{aligned}\frac{\partial O_k}{\partial in_k^{K+1}} &= \frac{\partial}{\partial in_k^{K+1}} f(in_k^{K+1}) = f'(in_k^{K+1}) \\ &= f(in_k^{K+1})(1 - f(in_k^{K+1})) \\ &= O_k(1 - O_k)\end{aligned}$$

From (4),

$$\frac{\partial in_k^{K+1}}{\partial w_{j,k}^K} = \frac{\partial}{\partial w_{j,k}^K}\left(\sum_{j=1}^{H_K} w_{j,k}^K h_j^K + 1\right) = h_j^K \tag{7}$$

Thus, substituting into (6),

$$\begin{aligned}\frac{\partial E}{\partial w_{j,k}^K} &= \frac{\partial E}{\partial O_k} \cdot \frac{\partial O_k}{\partial in_k^{K+1}} \cdot \frac{\partial in_k^{K+1}}{\partial w_{j,k}^K} \\ &= (O_k - T_k) \cdot O_k(1 - O_k) \cdot h_j^K\end{aligned}$$

## (ii)

From (5), substituting $X = h^i_{j,k}$ and $i \leftarrow i+1$, and since the input is sum of all the errors from $L_{i+2}$,

$$\frac{\partial E}{\partial h^{i+1}_k} = \sum_{m=1}^{H_{i+2}} \frac{\partial E}{\partial h^{i+2}_m} \cdot \frac{\partial h^{i+2}_m}{\partial in^{i+2}_m} \cdot \frac{\partial in^{i+2}_m}{\partial h^{i+1}_k} \tag{8}$$

From (1) & (3),

$$\begin{aligned} \frac{\partial h^{i+2}_m}{\partial in^{i+2}_m} &= \frac{\partial}{\partial in^{i+2}_m} \left( f(\sum_{k=1}^{H_{i+1}} w^{i+1}_{k,m} h^{i+1}_k + 1) \right) \\ &= \frac{\partial}{\partial in^{i+2}_m} f(in^{i+2}_m) \\ &= h^{i+2}_m (1 - h^{i+2}_m) \end{aligned} \tag{9}$$

From (4),

$$\frac{\partial in^{i+2}_m}{\partial h^{i+1}_k} = \frac{\partial}{\partial h^{i+1}_k} \left( \sum_{k=1}^{H_{i+1}} w^{i+1}_{k,m} h^{i+1}_k + 1) \right) = w^{i+1}_{k,m}$$

Thus, substituting into (8),

$$\begin{aligned} \frac{\partial E}{\partial h^{i+1}_k} &= \sum_{m=1}^{H_{i+2}} \frac{\partial E}{\partial h^{i+2}_m} \cdot \frac{\partial h^{i+2}_m}{\partial in^{i+2}_m} \cdot \frac{\partial in^{i+2}_m}{\partial h^{i+1}_k} \\ &= \sum_{m=1}^{H_{i+2}} \frac{\partial E}{\partial h^{i+2}_m} \cdot h^{i+2}_m (1 - h^{i+2}_m) \cdot w^{i+1}_{k,m} \\ &= \sum_{m=1}^{H_{i+2}} \Delta^{i+2}_m \cdot w^{i+1}_{k,m} \end{aligned}$$

## (iii)

From (5), substituting $X = w^i_{j,k}$,

$$\frac{\partial E}{\partial w^i_{j,k}} = \frac{\partial E}{\partial h^{i+1}_k} \cdot \frac{\partial h^{i+1}_k}{\partial in^{i+1}_k} \cdot \frac{\partial in^{i+1}_k}{\partial w^i_{j,k}} \tag{10}$$

Similar to (9) where $i+2 \leftarrow i+1$,

$$\frac{\partial h^{i+1}_k}{\partial in^{i+1}_k} = h^{i+1}_k(1 - h^{i+1}_k)$$

Similar to (7) where $K \leftarrow i$,

$$\frac{\partial in^{i+1}_k}{\partial w^i_{j,k}} = h^i_j$$

Thus, with the result in e(ii), substituting into (10),

$$\begin{aligned}\frac{\partial E}{\partial w^i_{j,k}} &= \frac{\partial E}{\partial h^{i+1}_k} \cdot \frac{\partial h^{i+1}_k}{\partial in^{i+1}_k} \cdot \frac{\partial in^{i+1}_k}{\partial w^i_{j,k}} \\ &= \left( \sum_{m=1}^{H_{i+2}} \Delta^{i+2}_m \cdot w^{i+1}_{k,m} \right) \cdot h^{i+1}_k(1 - h^{i+1}_k) \cdot h^i_j\end{aligned}$$

## (iv)

**repeat**
  **for** $e$ in *examples* **do**
    $O \leftarrow RunNetwork(network, I^e)$
    $w^K_{j,k} \leftarrow w^K_{j,k} + \alpha \cdot -\frac{\partial E^e}{\partial w^K_{j,k}}$
    **for** $L_i$ in $[L_K, L_0]$ **do**
      $x_t \leftarrow w^i_{j,k}$
      $w^i_{j,k} \leftarrow w^i_{j,k} + \alpha \cdot -\frac{\partial E^e}{\partial w^i_{j,k}}$
    **end for**
  **end for**
**until** $|w^i_{j,k} - x_t| < \varepsilon$

When expanded with result in e(i) and e(iii),

**4:** $w^K_{j,k} \leftarrow w^K_{j,k} - \alpha \cdot (O_k - T_k) \cdot O_k(1 - O_k) \cdot h^K_j$

**7:** $w^i_{j,k} \leftarrow w^i_{j,k} - \alpha \cdot \left( \sum_{m=1}^{H_{i+2}} \Delta^{i+2}_m \cdot w^{i+1}_{k,m} \right) \cdot h^{i+1}_k(1 - h^{i+1}_k) \cdot h^i_j$

## Question 1f

- **The vanishing gradient problem:** For activation function such as the sigmoid function, as the number of layer increases, the derivatives (gradient) towards the initial layers from back-propagation gets smaller and smaller exponentially. This causes minimal changes to the weight on the initial layers and makes the network not as cost-effective than lesser layers.
- **Time:** As the number of layer increases, the time needed to process forward- and back-propagation increases.

## Question 1g

A neural network attempts to generalize from the dataset and makes prediction when encounter with new data. **Overfitting** occurs when the dataset is too large that the network attempts to "memorize" the data instead of generalizing them. This results in poor performance when new data are run through the neural network, as the data and noises are out of those the network has memorized.

A simple solution to the said problem is to figure out a suitable amount of data that gets feed into the network. A limited amount of subset of the data, say 20% (this amount should not be too large so as to cause underfitting), are reserved to retrain the network in the case of inadequate data, or to validate the network's performance.