Lab: Forward Pass

ACTL3143 & ACTL5111 Deep Learning for Actuaries

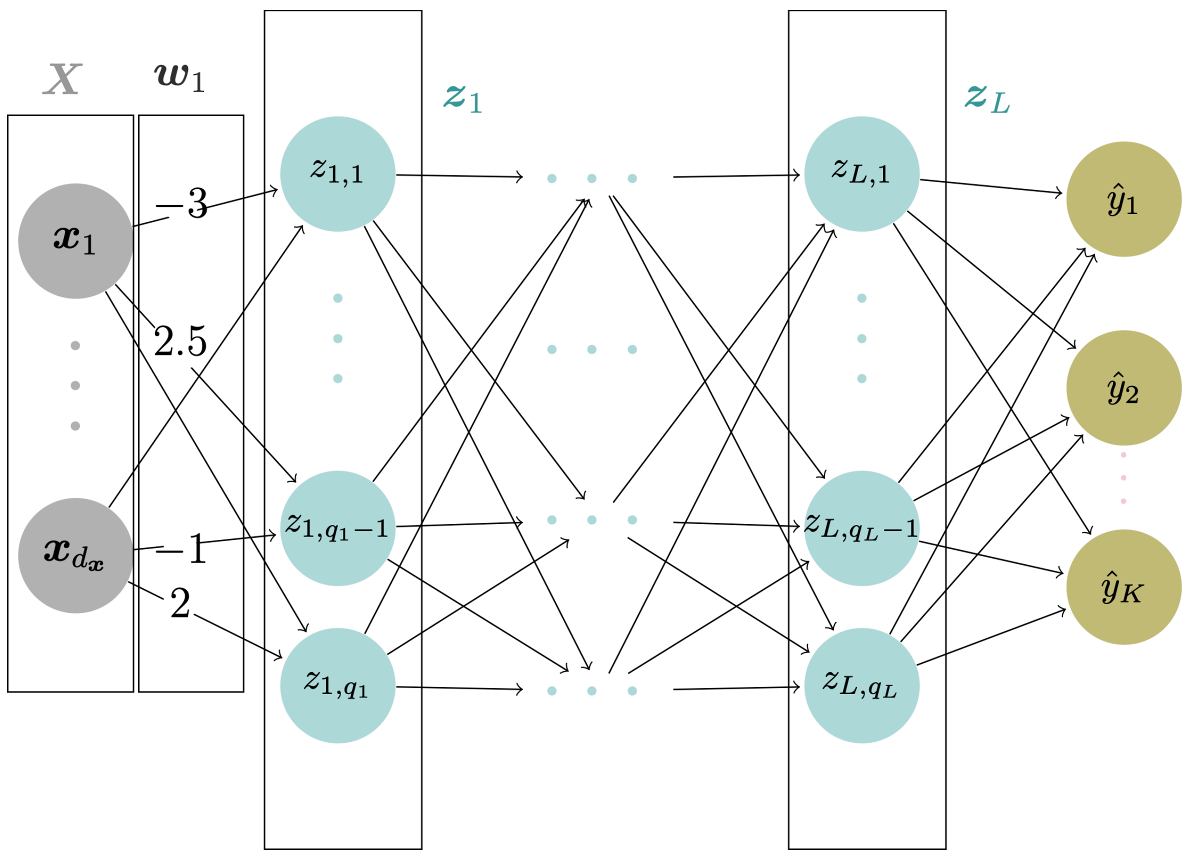


Figure 1: The structure of a neural network.

At each node in the hidden and output layers, the value z is calculated as a weighted sum of the node outputs in the previous layer, plus a bias. In other words:

$$z = Xw + b$$

where X is a $n \times d_x$ matrix representing the weights, w is an $d_x \times q$ matrix representing the weights (q representing the number of neurons in the current layer), and b is an $n \times q$ matrix representing the biases. n represents the number of observations and d_x represents the dimension of the input.

Example: Calculate the Neuron Values in the First Hidden Layer

$$X = \begin{pmatrix} 1 & 2 \\ 3 & -1 \end{pmatrix}, w = \begin{pmatrix} 2 \\ -1 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

We can calculate the neuron value as z follows:

$$\begin{aligned} z &= Xw + b \\ &= \begin{pmatrix} \quad\quad\quad\quad \end{pmatrix} \begin{pmatrix} \quad \end{pmatrix} + \begin{pmatrix} \quad \end{pmatrix} \\ &= \begin{pmatrix} \quad\quad\quad\quad \end{pmatrix} + \begin{pmatrix} \quad \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ 8 \end{pmatrix} \end{aligned}$$

Alternatively, one can use Python:

```python
import numpy as np
X = np.array([[1, 2], [3, -1]])
w = np.array([[2], [-1]])
b = np.array([[1], [1]])
print(X @ w + b)
```

```
[[1]
 [8]]
```

Exercises

1. (2×2 matrices) Calculate z, given:

    1. $X = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ $w = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

2. $X = \begin{pmatrix} 1 & -1 \\ 0 & 5 \end{pmatrix}$ $w = \begin{pmatrix} -1 \\ 8 \end{pmatrix}$ $b = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$

2. (3×3 matrices) Calculate z, given:

   1. $X = \begin{pmatrix} 4 & 4 & 0 \\ 2 & 2 & 4 \\ 2 & 4 & 1 \end{pmatrix}$ $w = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$ $b = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$
   2. $X = \begin{pmatrix} 6 & -6 & -2 \\ -3 & -1 & -5 \\ 1 & 1 & -7 \end{pmatrix}$ $w = \begin{pmatrix} 4 \\ 4 \\ -8 \end{pmatrix}$ $b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

3. (non-square matrices) Calculate z, given:

   1. $X = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \end{pmatrix}$ $w = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$ $b = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$
   2. $X = \begin{pmatrix} 1 & -1 \\ 0 & 5 \\ 2 & -2 \end{pmatrix}$ $w = \begin{pmatrix} 5 \\ -7 \end{pmatrix}$ $b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

4. If X is a 2×3 matrix, what does this say about the neural network's architecture? What about a 3×2 matrix?

Activation Functions

The result of $z = Xw + b$ will be in the range $(-\infty, \infty)$. However, sometimes we might want to constrain the values of z. We apply an **activation function** to z to do this. Activation functions include:

- Sigmoid: $S(z_i) = \frac{1}{1+\mathrm{e}^{-z_i}}$, constrains each value in z to $(0, 1)$
- Tanh: $\tanh(z_i) = \frac{\mathrm{e}^{2z_i}-1}{\mathrm{e}^{2z_i}+1}$, constrains each value in z to $(-1, 1)$.
- ReLU: $\mathrm{ReLU}(z_i) = \max(0, z_i)$, only activates for a value of z if it is positive.
- Softmax: $\sigma(z_i) = \frac{\mathrm{e}^{z_i}}{\Sigma_{j=1}^{K}\mathrm{e}^{z_j}}$. This maps the values in z so that each value is in $[0, 1]$ and the sum is equal to 1. This is useful for representing probabilities and is often used for the output layer.

Example: Applying Activation Functions

Given $z = \begin{pmatrix} 1 \\ 8 \end{pmatrix}$, calculate the resulting vector $a = \mathrm{activation}(z)$ using the four activation functions above.

- Sigmoid:

$$S(z) =$$

- Tanh:

$$\tanh(z) =$$

- ReLU

$$\mathrm{ReLU}(z) =$$

- Softmax

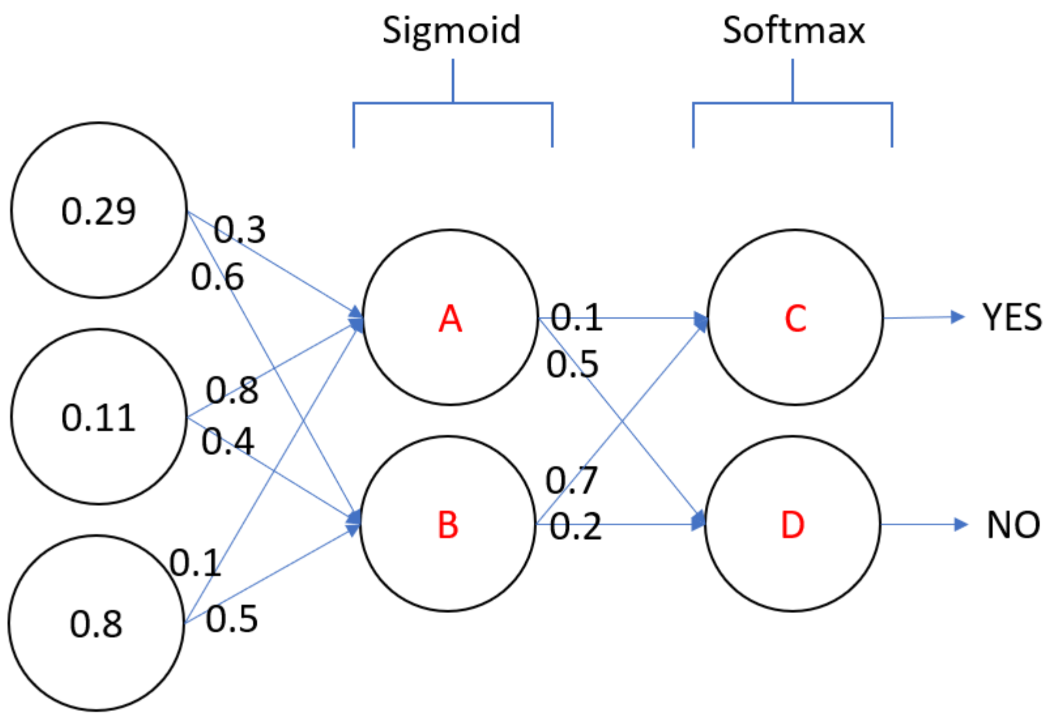
$$\sigma(z) =$$

Exercises

1. Given $z = \begin{pmatrix} 8 \\ 6 \end{pmatrix}$, calculate the resulting vector $a = \mathrm{activation}(z)$ using the four activation functions above.
2. Given $z = \begin{pmatrix} -8 \\ 9 \\ -3 \end{pmatrix}$, calculate the resulting vector $a = \mathrm{activation}(z)$ using the four activation functions above.
3. For extra practice, try calculating the vector a, using the results of the exercises in section 1.

Final Output

Example: Calculate the Final Output



1. With the activations, weights, and activation functions given in the above figure and a constant bias of 1 for each node, calculate the values of **A**, **B**, **C**, and **D**.
2. If the **C** node represents "YES" and the **D** node represents "NO", what final value is predicted by the neural network?

Hint: Write out

1. The input matrix X (should be 1×3):

$$X = (\qquad\qquad\qquad)\,.$$

2. The weight matrix w_1 between the input layer and the first hidden layer (should be 3×2):

$$w_1 = \left(\qquad\qquad\right), b_1 = (\qquad\qquad\qquad)\,.$$

3. The weight matrix w_2 between the first hidden layer and the output layer (should be 2×2):

$$w_2 = \left(\quad\quad \right), b_2 = (\quad\quad\quad) .$$

See more details in `Week_2_Tutorial_Notebook`