Feedforward Neural Network - Forward + backward propagation

Goal: Learn parameters that minimize prediction error by composing multiple linear transformations with non-linear activations

- Each hidden layer computes:

$$z^{(1)} = XW^{(1)} + b^{(1)}, \quad a^{(1)} = \text{ReLU}(z^{(1)})$$

- The output layer computes:

$$z^{(2)} = a^{(1)}W^{(2)} + b^{(2)}, \quad \hat{y} = z^{(2)}$$

- Loss function (MSE):

$$L = \frac{1}{n}\sum(y-\hat{y})^2$$

- Gradients via chain rule:

$$\frac{\partial L}{\partial W^{(2)}} = a^{(1)T}\frac{\partial L}{\partial \hat{y}}$$

$$\frac{\partial L}{\partial W^{(1)}} = X^T\left(\frac{\partial L}{\partial \hat{y}} W^{(2)T} \odot \text{ReLU}'(z^{(1)})\right)$$

- Parameter **update**:

$$W \leftarrow W - \eta\frac{\partial L}{\partial W}$$

Conclusion: Forward propagation computes predictions, and backward Propagation adjusts weights so the network gradually learns better representations.