Neural Network Forward and Backpropagation for Loan Risk Classification

Tran Quoc Thai
Student ID: 2370759

April 9, 2025

1 Forward Pass Calculation

Given input values:

$$\boldsymbol{x} = \begin{bmatrix} 0.375 \\ 0.583 \end{bmatrix}$$

Weights and biases for the hidden layer:

$$\boldsymbol{W_1} = \begin{bmatrix} 0.3 & 0.5 \\ 0.4 & -0.2 \end{bmatrix}, \quad \boldsymbol{b_1} = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}$$

Hidden layer pre-activation:

$$\begin{aligned} \boldsymbol{z_1} &= \boldsymbol{W_1}\boldsymbol{x} + \boldsymbol{b_1} \\ &= \begin{bmatrix} 0.3 & 0.5 \\ 0.4 & -0.2 \end{bmatrix} \begin{bmatrix} 0.375 \\ 0.583 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} \\ &= \begin{bmatrix} (0.3 \times 0.375 + 0.5 \times 0.583) + 0.1 \\ (0.4 \times 0.375 + (-0.2) \times 0.583) - 0.1 \end{bmatrix} \\ &= \begin{bmatrix} 0.425 \\ 0.023 \end{bmatrix} \end{aligned}$$

Applying sigmoid activation:

$$\begin{aligned} \sigma(z) &= \frac{1}{1 + e^{-z}} \\ \boldsymbol{a_1} &= \begin{bmatrix} \sigma(0.425) \\ \sigma(0.023) \end{bmatrix} = \begin{bmatrix} 0.6047 \\ 0.5057 \end{bmatrix} \end{aligned}$$

Weights and bias for the output layer:

$$\boldsymbol{W_2} = \begin{bmatrix} 0.6 & -0.4 \end{bmatrix}, \quad b_2 = 0.2$$

Output pre-activation:

$$\begin{aligned} z_2 &= \boldsymbol{W_2 a_1} + b_2 \\ &= (0.6 \times 0.6047) + (-0.4 \times 0.5057) + 0.2 \\ &= 0.3628 \end{aligned}$$

Applying sigmoid activation:

$$\hat{y} = \sigma(0.3628) = 0.5897$$

2 Backpropagation and Gradient Calculation

Error with respect to target output $y = 1$:

$$\delta_2 = \hat{y} - y = 0.5897 - 1 = -0.4103$$

Gradient for output weights:

$$\frac{\partial L}{\partial W_2} = \delta_2 \boldsymbol{a_1} = -0.4103 \begin{bmatrix} 0.6047 \\ 0.5057 \end{bmatrix} = \begin{bmatrix} -0.2481 \\ -0.2076 \end{bmatrix}$$

Gradient for output bias:

$$\frac{\partial L}{\partial b_2} = \delta_2 = -0.4103$$

Backpropagating to hidden layer:

$$\begin{aligned} \delta_1 &= \delta_2 W_2 \sigma'(z_1) \\ &= (-0.4103) \begin{bmatrix} 0.6 \\ -0.4 \end{bmatrix} \odot \begin{bmatrix} 0.6047(1 - 0.6047) \\ 0.5057(1 - 0.5057) \end{bmatrix} \\ &= \begin{bmatrix} (-0.4103 \times 0.6) \times (0.2390) \\ (-0.4103 \times -0.4) \times (0.2499) \end{bmatrix} \\ &= \begin{bmatrix} -0.0588 \\ 0.0410 \end{bmatrix} \end{aligned}$$

Gradient for hidden weights:

$$\begin{aligned} \frac{\partial L}{\partial W_1} &= \delta_1 \boldsymbol{x}^T = \begin{bmatrix} -0.0588 \\ 0.0410 \end{bmatrix} \begin{bmatrix} 0.375 & 0.583 \end{bmatrix} \\ &= \begin{bmatrix} -0.0221 & -0.0343 \\ 0.0154 & 0.0239 \end{bmatrix} \end{aligned}$$

Gradient for hidden biases:

$$\frac{\partial L}{\partial b_1} = \delta_1 = \begin{bmatrix} -0.0588 \\ 0.0410 \end{bmatrix}$$

3 Weight Updates

Using learning rate $\alpha = 0.1$:

$$
\begin{aligned}
W_2 &\leftarrow W_2 - \alpha \frac{\partial L}{\partial W_2} \\
&= \begin{bmatrix} 0.6 & -0.4 \end{bmatrix} - 0.1 \begin{bmatrix} -0.2481 & -0.2076 \end{bmatrix} \\
&= \begin{bmatrix} 0.6248 & -0.3792 \end{bmatrix}
\end{aligned}
$$

Similarly updating W_1 and biases.

4 Naive Bayes Classification for T1

4.1 Probability Calculations

Applying Bayes' theorem:

$$
P(\text{High Risk}|\text{T1}) = \frac{P(\text{T1}|\text{High Risk})P(\text{High Risk})}{P(\text{T1})}
$$

4.2 Comparison with Non-Naive Bayesian Approach

Naive Bayes assumes conditional independence between features, whereas a full Bayesian approach models dependencies explicitly.