Adjusting Weights in Backward Propagation: Simple Example

Let's go step by step to understand how weights are adjusted in backward propagation, using a very simple example.

Scenario: Predicting a Test Score

- A student studies for 5 hours.
- A neural network predicts the student will score **60%** on the test.
- The **actual score** is **80%**.

The network needs to adjust its weights to improve the prediction.

Simplified Neural Network

The network has:

1. **Input**: Study hours (5 hours).
2. **Weight (w)**: The importance of studying (initially set to 10).
3. **Output**: Prediction = $\text{Input} \times w = 5 \times 10 = 50$.
4. **Error**: The difference between the actual score and the prediction.

Steps to Adjust Weights

1. Forward Pass: Make a Prediction

The network calculates:

$$\text{Prediction} = \text{Input} \times w = 5 \times 10 = 50$$

But the actual score is 80, so the error is:

$$\text{Error} = (\text{Actual} - \text{Prediction})^2 = (80 - 50)^2 = 900$$

2. Calculate the Gradient

The gradient tells the network how much to change the weight. It's calculated as:

$$\text{Gradient} = \frac{\partial \text{Error}}{\partial w}$$

First, calculate:

$$\text{Error} = (\text{Actual} - \text{Prediction})^2$$

$$\frac{\partial \text{Error}}{\partial \text{Prediction}} = 2 \times (\text{Prediction} - \text{Actual})$$

Then:

$$\frac{\partial \text{Prediction}}{\partial w} = \text{Input}$$

Combine these:

$$\text{Gradient} = 2 \times (\text{Prediction} - \text{Actual}) \times \text{Input}$$

$$\text{Gradient} = 2 \times (50 - 80) \times 5 = -300$$

3. Update the Weight

The weight is updated using the formula:

$$w = w - \text{Learning Rate} \times \text{Gradient}$$

Let's say the **learning rate** is 0.01:

$$w = 10 - 0.01 \times (-300)$$

$$w = 10 + 3 = 13$$

4. Try Again (Next Iteration)

In the next iteration:

- The new weight ($w = 13$) gives a better prediction.
- Repeat the process until the error is small.

Summary of Adjustment Formula

1. Compute the **error**: $(\text{Actual} - \text{Prediction})^2$.
2. Compute the **gradient**: $\frac{\partial \text{Error}}{\partial w}$.
3. Update the weight: $w = w - \text{Learning Rate} \times \text{Gradient}$.

This iterative process ensures the weight gets adjusted in the right direction to reduce the error and improve predictions!