Activation-Based Saliency Maps: Two-Neuron Example

We extend the single-neuron example to a network with two neurons in the hidden layer.

Parameters:

- **Hidden Neuron 1 (h1):** Weights: w11 = 2, w12 = -1, w13 = 0.5, Bias: b1 = 0.5
- **Hidden Neuron 2 (h2):** Weights: w21 = -1, w22 = 1.5, w23 = -0.5, Bias: b2 = -0.5
- **Output Neuron:** Weights: wo1 = 1, wo2 = -2, Bias: bo = 0.5
- **Input Values:** x1 = 1, x2 = 2, x3 = -1

Step 1: Compute Activations of Hidden Neurons

h1 = sigmoid(2*1 + (-1)*2 + 0.5*(-1) + 0.5) = sigmoid(0) = 0.5

h2 = sigmoid(-1*1 + 1.5*2 + (-0.5)*(-1) - 0.5) = sigmoid(2) approximately 0.88

Step 2: Compute Output

y = sigmoid(1*h1 + (-2)*h2 + 0.5) = sigmoid(0.5 - 1.76 + 0.5) = sigmoid(-0.76) approximately 0.32

Step 3: Compute Gradients (Saliency Map)

dy/dz_o = 0.32 * (1 - 0.32) = 0.217

Gradients w.r.t. hidden neurons:

- dy/dh1 = 0.217 * 1 = 0.217
- dy/dh2 = 0.217 * (-2) = -0.434

Gradients w.r.t. inputs:

- dy/dx1 = (0.217 * 0.25 * 2) + (-0.434 * 0.1056 * -1) = 0.1544
- dy/dx2 = (0.217 * 0.25 * -1) + (-0.434 * 0.1056 * 1.5) = -0.1229
- dy/dx3 = (0.217 * 0.25 * 0.5) + (-0.434 * 0.1056 * -0.5) = 0.05

Step 4: Interpretation

- x1 (0.1544) has a strong positive effect on y.
- x2 (-0.1229) has a negative effect on y.
- x3 (0.05) has a small positive effect.

This saliency map helps interpret how each input affects the model's decision.