Solved by: Islam Nabi

Date: 11/06/2025

Problem 01: Computational Graphs
Part 1: Forward Pass
Given:

$c = 2$

$d = 5$

$a = 3d^3 + 2c + b^2$

$b = 2c^2$

$z = 3a + 2b^2$

Calculations:

Compute b:
$b = 2c^2 = 2*(2)^2 = 2*4 = 8$

Compute a:
$a = 3d^3 + 2c + b^2 = 3*(5)^3 + 2*2 + (8)^2 = 3*125 + 4 + 64 = 375 + 4 + 64 = 443$

Compute z:
$z = 3a + 2b^2 = 3*443 + 2*(8)^2 = 1329 + 2*64 = 1329 + 128 = 1457$

Final Values:

$a = 443$

$b = 8$

$z = 1457$

Part 2: Chain Rule for dz/dc
We need to compute dz/dc using the chain rule:

Given:
$z = 3a + 2b^2$
$a = 3d^3 + 2c + b^2$
$b = 2c^2$

Chain rule path: $z \to a \to c$ and $z \to b \to c$

$dz/da = 3$

dz/db = 4b (since dz/db = d/db($2b^2$) = 4b)

da/dc = 2 + 2b(db/dc) (since a has both direct c term and b term which depends on c)

db/dc = 4c

Now compute da/dc:
da/dc = 2 + 2b*(db/dc) = 2 + 2*8*4*2 = 2 + 128 = 130

Compute dz/dc:
dz/dc = (dz/da)(da/dc) + (dz/db)(db/dc)
= 3*130 + 4*8*4*2
= 390 + 256
= 646

Final Value:
dz/dc = 646

Problem 02: Back Propagation
Part A: Forward Pass
Given:

Inputs: $x_1 = 0.5$, $x_2 = 0.3$

Activation functions: tanh for hidden layers, sigmoid for output layer

Biases = 0

1. First Hidden Layer (a_{11}, a_{12}, a_{13}, a_{14})
For each node $a_{1\square}$:
$a_{1\square} = \tanh(w^1_{1\square} x_1 + w^1_{2\square} x_2)$

Calculations:

a_{11} = tanh(0.9*0.5 + (-0.79)*0.3) = tanh(0.45 - 0.237) = tanh(0.213) ≈ 0.210

a_{12} = tanh(0.75*0.5 + (-0.03)*0.3) = tanh(0.375 - 0.009) = tanh(0.366) ≈ 0.350

a_{13} = tanh(-0.15*0.5 + 0.1*0.3) = tanh(-0.075 + 0.03) = tanh(-0.045) ≈ -0.045

a_{14} = tanh(0.33*0.5 + 0.02*0.3) = tanh(0.165 + 0.006) = tanh(0.171) ≈ 0.169

2. Second Hidden Layer (a_{21}, a_{22}, a_{23})
For each node $a_{2\square}$:
$a_{2\square} = \tanh(\sum(w^2_{i\square} * a_{1i})$ for i=1 to 4)

Calculations:

a_{21} = tanh(-0.58*0.210 + 0.23*0.350 + 0.25*(-0.045) + (-0.57)*0.169)
= tanh(-0.1218 + 0.0805 - 0.01125 - 0.09633) ≈ tanh(-0.14888) ≈ -0.148

a_{22} = tanh(1.0*0.210 + 0.45*0.350 + (-0.65)*(-0.045) + 0.51*0.169)
= tanh(0.210 + 0.1575 + 0.02925 + 0.08619) ≈ tanh(0.48294) ≈ 0.449

a_{23} = tanh(-0.8*0.210 + 0.05*0.350 + (-0.09)*(-0.045) + (-0.33)*0.169)
= tanh(-0.168 + 0.0175 + 0.00405 - 0.05577) ≈ tanh(-0.20222) ≈ -0.199

3. Output Layer (o_1, o_2)
For each output node $o_{\square}$:
$o_{\square} = \sigma(\sum(w^3_{i\square} * a_{2i})$ for i=1 to 3), where σ is sigmoid function

Calculations:

o_1 = σ(0.97*(-0.148) + (-0.01)*0.449 + 0.85*(-0.199))
= σ(-0.14356 - 0.00449 - 0.16915) ≈ σ(-0.3172) ≈ 0.421

o_2 = σ(0.3*(-0.148) + (-0.2)*0.449 + 0.76*(-0.199))
= σ(-0.0444 - 0.0898 - 0.15124) ≈ σ(-0.28544) ≈ 0.429

Part B: Backward Propagation and Weight Update
1. Compute $\partial E/\partial w^3$ (Output layer weights)
Binary cross-entropy error for one sample:
$E = -[y_1 \ln(o_1) + y_2 \ln(o_2) + (1-y_1)*\ln(1-o_1) + (1-y_2)*\ln(1-o_2)]$

For output layer with sigmoid activation:
$\partial E/\partial w^3_{i\square} = (o_{\square} - y_{\square})*a_{2i}$

Given:
$y_1 = 0.4$, $y_2 = 0.45$
$o_1 \approx 0.421$, $o_2 \approx 0.429$

Calculations:
For w^3_{11}:
$\partial E/\partial w^3_{11} = (o_1 - y_1)*a_{21}$ = (0.421 - 0.4)*(-0.148) ≈ 0.021*(-0.148) ≈ -0.003108

For w^3_{21}:
$\partial E/\partial w^3_{21} = (o_1 - y_1)a_{22}$ ≈ 0.0210.449 ≈ 0.009429

For w^3_{31}:
$\partial E/\partial w^3_{31} = (o_1 - y_1)a_{23}$ ≈ 0.021(-0.199) ≈ -0.004179

Similarly for o_2 weights:
For w^3_{12}:

$\partial E/\partial w^3_{12} = (o_2 - y_2)*a_{21} \approx (0.429 - 0.45)*(-0.148) \approx -0.021*(-0.148) \approx 0.003108$

For w^3_{22}:
$\partial E/\partial w^3_{22} = (o_2 - y_2)*a_{22} \approx -0.021*0.449 \approx -0.009429$

For w^3_{32}:
$\partial E/\partial w^3_{32} = (o_2 - y_2)*a_{23} \approx -0.021*(-0.199) \approx 0.004179$

2. Compute $\partial E/\partial w^2$ (Second hidden layer weights)
Using chain rule through output layer:
$\delta_{2\square} = \sum(w^3_{\square\square}*(o_\square - y_\square)) * (1 - \tanh^2(a_{2\square}))$

Calculations:
For node a_{21}:
$\delta_{21} = [w^3_{11}(o_1-y_1) + w^3_{12}(o_2-y_2)] * (1 - \tanh^2(a_{21}))$
$\approx [0.97*0.021 + 0.3*(-0.021)] * (1 - (-0.148)^2)$
$\approx [0.02037 - 0.0063] * (1 - 0.0219)$
$\approx 0.01407 * 0.9781 \approx 0.01376$

For node a_{22}:
$\delta_{22} \approx [(-0.01)*0.021 + (-0.2)*(-0.021)] * (1 - 0.449^2)$
$\approx [-0.00021 + 0.0042] * (1 - 0.2016)$
$\approx 0.00399 * 0.7984 \approx 0.003186$

For node a_{23}:
$\delta_{23} \approx [0.85*0.021 + 0.76*(-0.021)] * (1 - (-0.199)^2)$
$\approx [0.01785 - 0.01596] * (1 - 0.0396)$
$\approx 0.00189 * 0.9604 \approx 0.001815$

Now compute $\partial E/\partial w^2_{i\square} = \delta_{2\square} * a_{1i}$

Example calculations:
For w^2_{11}:
$\partial E/\partial w^2_{11} = \delta_{21} * a_{11} \approx 0.01376 * 0.210 \approx 0.00289$

For w^2_{22}:
$\partial E/\partial w^2_{22} = \delta_{22} * a_{12} \approx 0.003186 * 0.350 \approx 0.001115$

3. Compute $\partial E/\partial w^1$ (First hidden layer weights)
First compute $\delta_{1\square}$ for first hidden layer:
$\delta_{1\square} = \sum(w^2_{\square\square} * \delta_{2\square}) * (1 - \tanh^2(a_{1\square}))$

Calculations:
For node a_{11}:
$\delta_{11} = [w^2_{11}\delta_{21} + w^2_{12}\delta_{22} + w^2_{13}*\delta_{23}] * (1 - \tanh^2(a_{11}))$
$\approx [-0.58*0.01376 + 1.0*0.003186 + -0.8*0.001815] * (1 - 0.210^2)$
$\approx [-0.007981 + 0.003186 - 0.001452] * (1 - 0.0441)$

$\approx -0.006247 * 0.9559 \approx -0.005971$

Now compute $\partial E/\partial w^1_{i\square} = \delta_{1\square} * x_i$

Example calculations:
For w^1_{11}:
$\partial E/\partial w^1_{11} = \delta_{11} * x_1 \approx -0.005971 * 0.5 \approx -0.002986$

For w^1_{22}:
$\partial E/\partial w^1_{22} = \delta_{12} * x_2 \approx$ (similar calculation for δ_{12}) * 0.3

4. Weight Updates
Update rule: $w = w - \eta*(\partial E/\partial w)$, where $\eta = 0.01$

Example updates:
For w^3_{11}:
$w^3_{11}(new) = 0.97 - 0.01*(-0.003108) \approx 0.97 + 0.000031 \approx 0.970031$

For w^2_{11}:
$w^2_{11}(new) = -0.58 - 0.01*0.00289 \approx -0.58 - 0.0000289 \approx -0.5800289$

For w^1_{11}:
$w^1_{11}(new) = 0.9 - 0.01*(-0.002986) \approx 0.9 + 0.00002986 \approx 0.90002986$