Mark Gameng

CS 422 – HW 7

**Part 1.1 – Questions, Chapter 4**

1. **Boolean functions**
   1. **Linearly separable**
   2. **Linearly separable**
   3. **Linearly separable**
   4. **Not linearly separable – has XOR**
   5. **A and B and C is the same as: Y = 1 if A \* B \* C > 0 else 0**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **A and B and C** |
| **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **1** |

**A or B is equivalent to Y = 1 if A + B > 0 else 0**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **A or B** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **1** |

* 1. **Disadvantage of using linear functions as activation functions for multi-layer neural networks is that having a multi-layer is useless. Multi-layer has no meaning because since its linear, the result is just a linear combination of all the linear functions which is just a linear function. Thus, it’s the same as having 1 layer.**

**Part 1.2**

1. **W = [1,1,-1,0.5,1,2]  
   z1 = max(0, w1 \* x) = max(0, 1 \* 4) = 4  
   z2 = max(0, w2 \* x) = max(0, 1 \* 4) = 4  
   z3 = max(0, w3 \* x) = max(0, -1 \* 4) = 0  
   y^ = z1 \* w4 + z2 \* w5 + z3 \* w6 = 4 \* 0.5 + 4 \* 1 + 0 \* 2 = sigmoid(6) = 0.998  
   Predicted output = 0.998**
2. **Error = (y – y^)^2 = (0 – 0.998)^2 = 0.996**
3. **=[dE/dw1, dE/dw2, dE/dw3, dE/dw4, dE/dw5, dE/dw6]  
   y = z1\*w4 + z2 \* w5 + z3 \* w6  
   dE/dw6 = dE/dsigmoid(y) \* dsigmoid(y)/dy \* dy/dw6   
   = -2(y - sigmoid(y^)) \* (sigmoid(y^)(1-sigmoid(y^)) \* relu(z3)   
   =-2(0 - 0.998) \* (0.998 \* (1 – 0.998)) \* 0 = 0  
   dE/dw5 dE/dsigmoid(y) \* dsigmoid(y)/dy \* dy/dw5 = -2(0 - 0.998) \* (0.998 \* (1 – 0.998)) \* 4 = 0.016  
   dE/dw4 = dE/dsigmoid(y) \* dsigmoid(y)/dy \* dy/dw4 = -2(0 - 0.998) \* (0.998 \* (1 – 0.998)) \* 4 = 0.016  
   z1 = w1 \* x = 4w1, z2 = w2 \* x = 4w2, z3 = w3 \* x = 4w3  
   dE/dw3 = dE/relu(z3) \* drelu(z3)/dz3 \* dz3/dw3  
   = (dE/dsigmoid(y) \* dsigmoid(y)/dy \* dy/drelu(z3)) \* (1 if z3 > 0 else 0) \* 4  
   = (-2(0 - 0.998) \* (0.998(1-0.998) \* 2)) \* (0) \* 4 = 0  
   dE/dw2 = dE/relu(z2) \* drelu(z1)/dz2 \* dz2/dw2  
   = (-2(0 - 0.998) \* (0.998(1-0.998) \* 1)) \* (1) \* 4 = 0.016  
   dE/dw1 = dE/relu(z1) \* drelu(z1)/dz1 \* dz1/dw1  
   = (-2(0 - 0.998) \* (0.998(1-0.998) \* 0.5)) \* (1) \* 4 = 0.008  
   Thus, =[0.008, 0.016, 0, 0.016, 0.016, 0]**
4. **Wn = w – 1\*  
    = [1,1,-1,0.5,1,2] – [0.008, 0.016, 0, 0.016, 0.016, 0]  
   = [0.992, 0.984, -1, 0.484, 0.984, 2]  
   z1 = max(0, w1 \* x) = max(0, 0.992 \* 4) = 3.968  
   z2 = max(0, w2 \* x) = max(0, 0.984 \* 4) = 3.936  
   z3 = max(0, w3 \* x) = max(0, -1 \* 4) = 0  
   y^ = z1 \* w4 + z2 \* w5 + z3 \* w6 = 3.968 \* 0.484 + 3.936 \* 0.984 + 0 \* 2 = 5.793   
   sigmoid(5.793) = 0.997  
   Predicted output = 0.997  
   Error = (y – y^)^2 = (0 – 0.997)^2 = 0.994**
5. **The difference between the loss values is 0.002, which isn’t that much but it is a decrease in error which makes sense because the backpropagation should result in being closer to the actual output.**