Deep Learning HW2

**Question 1 - Generalization in A Teacher-Student Setup**

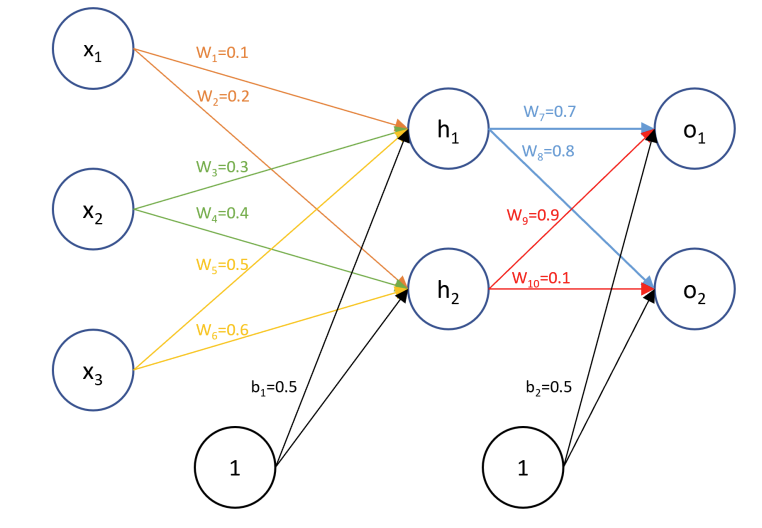
**Prove:**

**Solution:**

=

\*\*\*

**Question 2 – Backpropagation By Hand**



1. **Perform forward pass, and calculate the MSE**
2. **Perform backward pass, and calculate the gradients**
3. Calculate the new weights with SGD with learning rate of 0.01

Foreach parameter x we apply the following equation: and get:

**Question 3 – Deep Double Decent**

1. **Where is the critical point?**
2. **What type of double descent is shown?**
3. **Part A**

A picture containing text, line, diagram, plot

Description automatically generated

1. The critical point is when the first plot reaches 200, and for the second graph when reaching 300.
2. Model double decent.
3. **Part B**

A picture containing text, diagram, plot, line

Description automatically generated

1. The critical point is located approximately between 10 to 14 for both graphs.
2. Model double decent
3. **Part C**

A picture containing text, screenshot, line, plot

Description automatically generated

1. Small models don’t suffer from this phenomenon, hence no critical point in the yellow model, same goes for the intermediate model where it reaches a stable maximum at higher epochs, however larger models suffer quite a bit, the red has a critical point in between 70 and 90 epochs.
2. Epoch-wise double descent. Reverse overfitting on the intermediate model.

**Question 4 – Initialization**

From the lecture we concluded that when for large widths and the following holds

Hence

The width of the previous layer is we get the following:

**Question 5 – MLP and Invariance**

1. The activation function (Leaky RELU) is the only invariance in this network when

\*\*\* the \*= is possible since the arguments of the max function are two linear functions.

1. For this case we need to force the parameters for each row in the input matrix to be symmetric meaning the first and last element parameter should be the same along the d-axis.

Foreach row in input matrix:

Foreach col in range(d/2) :

w[row,col] == w[row, d-col]

1. We need to force the same parameters across equivalent rows, meaning all parameters in row 1 are the same as the parameters in row 4, same goes for rows 2 and 3.
2. (a) The network is already trained so the positive neurons will still fire a positive result and the negative neurons will still fire a negative result, since the network is only a single layer no propagated change takes effect, hence no change at all in the classification.

(b) the learning rate is multiplied by c hence we expect the algorithm to never converge if the multiplication c\*a is greater than 1.

(c) one way to eliminate the effect of this constant is by making each column a one hot encoding only one row is turned on.

**Question 6 – VGG Architecture**

1. **Complete the network architecture**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Output Dimension** | **Number of parameters** |
| INPUT | 224x224x3 | 0 |
| CONV3-64 | 224x224x64 | 1792 |
| RELU | 224x224x64 | 0 |
| POOL2 | 112x112x64 | 0 |
| CONV3-128 | 112x112x128 | 73856 |
| RELU | 112x112x128 | 0 |
| POOL2 | 56x56x128 | 0 |
| CONV3-256 | 56x56x256 | 295168 |
| RELU | 56x56x256 | 0 |
| CONV3-256 | 56x56x256 | 590080 |
| RELU | 56x56x256 | 0 |
| POOL2 | 28x28x256 | 0 |
| CONV3-512 | 28x28x512 | 1180160 |
| RELU | 28x28x512 | 0 |
| CONV3-512 | 28x28x512 | 2359808 |
| RELU | 28x28x512 | 0 |
| POOL2 | 14x14x512 | 0 |
| CONV3-512 | 14x14x512 | 2359808 |
| RELU | 14x14x512 | 0 |
| CONV3-512 | 14x14x512 | 2359808 |
| RELU | 14x14x512 | 0 |
| POOL | 7x7x512 | 0 |
| FC-4096 | 4096 | 102764544 |
| FC-4096 | 4096 | 16781312 |
| FC-1000 | 1000 | 4097000 |
| SOFTMAX | 1 | 0 |

1. **Total number of params**

For conv layers the number of parameters is:

For FC layers the number of parameters is:

Sum of parameters = 132,863,336

1. **Percentage of fully connected layer params from overall params**