Names:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CS 455 – Homework #5

**Due start of class 4/17/2019**

Submit in class on You may work individually or in pairs. Only one submission required per pair.

1. [20 points] Perceptron

For the following three input perceptron, calculate the output, error, delta weights, and new weights for each of the inputs given. Use the updated weights from the previous row as the weights of the next row (i.e. this is executing sequentially)

α = 0.1

Initial Weights: W1 = 0.1, W2=0.1, W3=-0.3, θ = 0.1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X1** | **X2** | **X3** | **Yd** | **Y** | **E** | **ΔW1** | **ΔW2** | **ΔW3** | **W1(t+1)** | **W2(t+1)** | **W3(t+1)** |
| **1** | **0** | **0** | **1** | 0 | 1 | 0.1 | 0 | 0 | 1.1 | 0.1 | -0.3 |
| **1** | **0** | **1** | **1** | 0.7 | 0.3 | 0.03 | 0 | 0.03 | 1.13 | 0.1 | -0.27 |
| **0** | **1** | **1** | **0** | -0.27 | 0.27 | 0 | 0.027 | 0.027 | 1.13 | 0.127 | -0.243 |

W1

X1

-1

θ

W3

W2

X3

X2

Y

1. [40 points] Artificial Neural Network with Back Propagation



|  |  |  |  |
| --- | --- | --- | --- |
| **Neuron 3**  W13 = 0.20  W23 = -0.32  Θ3 = 0.15 | **Neuron 4**  W14 = 0.17  W24 = 0.22  Θ4 = -0.1 | **Neuron 5**  W35 = -0.13  W45 = 0.20  Θ5 = 0.05 | **Neuron 6**  W36 = 0.10  W46 = -0.12  Θ6 = 0.3 |
| **Inputs:** X1 = 1, X2 = 1 | | **Desired Outputs:** Y5=0, Y6=1 | |

α = 0.1

1. Calculate Y3, Y4, Y5, and Y6 (using sigmoid)
   1. Y3=0.43291
   2. Y4=0.62011
   3. Y5=0.50444
   4. Y6=0.41797
2. Calculate errors e5 and e6
   1. e5=-0.50444
   2. e6=0.58203
3. Calculate error gradients δ5, δ6
   1. δ5=-0.12610
   2. δ6=0.14159
4. Calculate error gradients δ3, δ4
   1. δ3­­5=0.00402 δ3­­6=0.00348
   2. δ45=-0.00594 δ46=-0.00400
5. [50 points] Neural Network Evaluation in TF with Keras

You will be modifying a Jupyter Notebook or Python Script to evaluate an artificial neural network implemented in TensorFlow using the Keras API. You will be using the data set: <https://www.openml.org/d/53>. Code already exists to build from including data preparation and an initial model. You must fill in the rest.

1. First, download the program. Available on the assignment page on Canvas.
2. Review the program and make sure you understand what is happening within it.
3. Modify the program to output the following metrics and plots:
   1. Metrics: Accuracy, Confusion Matrix, Precision, and Recall
   2. Plots: Precision-Recall Curve, ROC Curve

A screenshot of a cell phone

Description automatically generated

1. Now, you will modify the program to evaluate a number of model configurations:

|  |  |
| --- | --- |
| Model ID | Topology |
| 0 (default) | Input: 13  Hidden: 5 (relu)  Output: 1 (logistic) |
| 1 | Input: 13  Hidden: 10 (relu)  Hidden: 5 (relu)  Output: 1 (logistic) |
| 2 | Input: 13  Hidden: 50 (relu)  Output: 1 (logistic) |
| 3 | Input: 13  Hidden: 10 (relu)  Hidden: 10 (relu)  Output: 1 (logistic) |

For each of the configurations, you must provide the following:

* Screenshot of the graph from TensorBoard
* Screenshot of learning curve from TensorBoard
* Performance metrics (see above)
* Summary of any observations:
  + Did it take longer?
  + Did it take shorter?
  + Did it overfit?
  + Did it underfit?

1. From the configurations given, which model would you choose and why? Please note there might be more than one answer and I am looking for your overall reasoning based upon your observations.