|  |  |
| --- | --- |
| ***CS2NN16 MLP Lab 1 Report Sheet 2018/19*** | |
| **Student Number: 26007252** | **Date : 31/10/18** |

|  |  |
| --- | --- |
| **Introduction** | **Mark / 3** |
| << Write intro here >> | |

|  |  |
| --- | --- |
| **Output of Untrained LinearLayerNetwork network** | **Mark / 1** |
| Inputs Targets Raw Ops Outputs  x1 x2 AND OR XOR AND OR XOR AND OR XOR  0 0 0 0 0 0.200 0.300 0.400 0 0 0  0 1 0 1 1 0.500 0.400 0.600 1 0 1  1 0 0 1 1 0.700 0.800 0.500 1 1 1  1 1 1 1 0 1.000 0.900 0.700 1 1 1  Over Set : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Weights  Epoch 1 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 2 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 3 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 4 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 5 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 6 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Epoch 7 : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Inputs Targets Raw Ops Outputs  x1 x2 AND OR XOR AND OR XOR AND OR XOR  0 0 0 0 0 0.200 0.300 0.400 0 0 0  0 1 0 1 1 0.500 0.400 0.600 1 0 1  1 0 0 1 1 0.700 0.800 0.500 1 1 1  1 1 1 1 0 1.000 0.900 0.700 1 1 1  Over Set : SSE 0.1950 0.1250 0.2650 : %Correct 50 75 75  Weights | |

|  |  |  |
| --- | --- | --- |
| **LinearLayerNetwork** | | |
| **getWeights** | **Code Mark / 2** | **Comment / 2** |
| **protected** **void** findDeltas(ArrayList<Double> errors) {  **for** (**int** cnt = 0; cnt < errors.size();cnt++){ // for each error in errors  deltas.set(cnt, errors.get(cnt)); // set the delta (of the same index as the error's index) as the error  }  } | | |

|  |  |
| --- | --- |
| **Weights of Untrained LinearLayerNetwork network** | **Mark / 1** |
| << Having written getWeights, put here string from console listing all the weights >> | |

|  |  |  |
| --- | --- | --- |
| **findDeltas** | **Code / 2** | **Comments / 2** |
| **protected** **void** findDeltas(ArrayList<Double> errors) {  **for** (**int** cnt = 0; cnt < errors.size();cnt++){ // for each error in errors  deltas.set(cnt, errors.get(cnt)); // set the delta (of the same index as the error's index) as the error  }  } | | |

|  |  |  |
| --- | --- | --- |
| **changeAllWeights** | **Code / 2** | **Comments / 2** |
| **protected** **void** changeAllWeights(ArrayList<Double> ins, **double** learnRate, **double** momentum) {  **for**(**int** neuron = 0; neuron < numNeurons; neuron++){ // for every neuron  changeOneWeight(neuron, -1, 1, learnRate, momentum); // calculate the change in weight for the bias weight of each neuron  **for** (**int** weight = 0; weight < ins.size(); weight++){ // for each weight between each input and the neuron (excluding the bias weight because that has already been accounted for)  changeOneWeight(neuron, weight, ins.get(weight), learnRate, momentum); // change a particular weight of the neuron using the input that the weight is connected to  }  }  } | | |

|  |  |  |
| --- | --- | --- |
| **changeOneWeight** | **Code / 3** | **Comments / 2** |
| **private** **void** changeOneWeight(**int** wNeuron, **int** wWeight, **double** theIn, **double** learnRate, **double** momentum) {  /\*  \* the new wWeight is the index of that particular weight for its respective neuron  \*/  wWeight = weightIndex(wNeuron, wWeight);    /\*  \* the change in weight (wWeight) is the learning rate  \* multiplied by the delta of that particular neuron  \* multiplied by the input of that particular neuron  \* plus the momentum multiplied by the previous change in weight  \*/  changeInWeights.set(wWeight, (learnRate \* deltas.get(wNeuron) \* theIn + momentum\*changeInWeights.get(wWeight)));    /\*  \* After calculating the new change in weight the current weight  \* is equal to itself plus the new change in weight  \*/  weights.set(wWeight, weights.get(wWeight) + changeInWeights.get(wWeight));    } | | |

|  |  |  |
| --- | --- | --- |
| **weightIndex** | **Code / 1** | **Comments / 1** |
| **private** **int** weightIndex (**int** wNeuron, **int** wWeight) {  /\*  \* the index of a weight for a particular neuron is that neuron's index  \* multiplied by the number of weights per neuron  \* plus the weight itself  \* plus 1 for the bias weight  \*/    **return** wNeuron \* (numWeights / numNeurons) + wWeight + 1;  } | | |

|  |  |
| --- | --- |
| **Modify main so learning rate is 0.1 momentum is 0.3. Paste below how the SSE changes during training, and the data set and weights after 10 epochs of training** |  |
| **Mark /3** |
| << Paste here how SSE and %Correct changes at each epoch during training, and the training set after training and the weights >> | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Code for SigmoidLayerNetwork – mark scheme** | | | | |
| **SigmoidLayerNetwork CalcOutputs** | | **Code / 3** | **Comments / 2** | |
| **protected** **void** calcOutputs(ArrayList<Double> nInputs) {  **super**.calcOutputs(nInputs); // call the parent class calcOutputs method to get the outputs  **for** (**int** neuron = 0; neuron < numNeurons; neuron++){ // for each neuron  outputs.set(neuron, (1.0/(1.0 + Math.*exp*(-(outputs.get(neuron)))))); // set the output of the neuron as the sigmoid version using the output of the neuron  }  } | | | | |
| **SigmoidLayerNetwork findDeltas** | **Code / 2** | | | **Comments / 2** |
| **protected** **void** findDeltas(ArrayList<Double> errors) {  // write code to set delta as error \* deriv activation  **for** (**int** neuron = 0; neuron < numNeurons; neuron++){ // for each neuron    /\*  \* assign the delta as the error multiplied by the output multiplied by (1.0 - the output)  \*/  deltas.set(neuron, errors.get(neuron) \* outputs.get(neuron) \* (1.0 - outputs.get(neuron)));  }  } | | | | |

|  |  |
| --- | --- |
| **Program output with default weights, a learning rate of 0.15 and momentum of 0.4 : show state *before*, *during* and then *after* 1000 epochs of training** | **Mark /3** |
| << paste results here >> | |

|  |  |
| --- | --- |
| **SigmoidLayerNetwork Other data Set**  **In GUI on Other data, set learning rate to 0.3 and momentum to 0.2**  **Initialise, Present, Learn, Present and show result below:** | **Mark / 2** |
| Sigmoid Layer : Other Learn Rate 0.30 Momentum 0.20 Seed 100  Inputs Targets Raw Ops Outputs  x y O1 O2 O1 O2 O1 O2  0.1 1.2 1 0 0.687 0.706 1 1  0.7 1.8 1 0 0.650 0.762 1 1  0.8 1.6 1 0 0.620 0.758 1 1  1.0 0.8 0 0 0.525 0.729 1 1  0.3 0.5 1 1 0.605 0.679 1 1  0.0 0.2 1 1 0.625 0.647 1 1  -0.3 0.8 1 1 0.710 0.664 1 1  -0.5 -1.5 0 1 0.562 0.517 1 1  -1.5 -1.3 0 1 0.716 0.470 1 0  Over Set : SSE 0.2055 0.3377 : %Correct 66 44  Epoch 50 : SSE 0.0902 0.0529 : %Correct 88 100  Epoch 100 : SSE 0.0782 0.0357 : %Correct 88 100  Epoch 150 : SSE 0.0705 0.0272 : %Correct 88 100  Epoch 200 : SSE 0.0645 0.0219 : %Correct 100 100  Epoch 250 : SSE 0.0595 0.0182 : %Correct 100 100  Epoch 300 : SSE 0.0551 0.0156 : %Correct 100 100  Epoch 350 : SSE 0.0512 0.0135 : %Correct 100 100  Epoch 400 : SSE 0.0477 0.0119 : %Correct 100 100  Epoch 450 : SSE 0.0445 0.0106 : %Correct 100 100  Epoch 500 : SSE 0.0417 0.0096 : %Correct 100 100  Inputs Targets Raw Ops Outputs  x y O1 O2 O1 O2 O1 O2  0.1 1.2 1 0 0.999 0.176 1 0  0.7 1.8 1 0 0.999 0.001 1 0  0.8 1.6 1 0 0.996 0.002 1 0  1.0 0.8 0 0 0.377 0.080 0 0  0.3 0.5 1 1 0.749 0.822 1 1  0.0 0.2 1 1 0.667 0.984 1 1  -0.3 0.8 1 1 0.997 0.875 1 1  -0.5 -1.5 0 1 0.001 1.000 0 1  -1.5 -1.3 0 1 0.213 1.000 0 1  Over Set : SSE 0.0402 0.0094 : %Correct 100 100 | |

|  |  |
| --- | --- |
| **SigmoidLayerNetwork Other data Set -**  **Tadpole plots** | **Mark / 1** |
|  | |

|  |  |
| --- | --- |
| **Discussion (on code and results)** | **Mark / 4** |
| << Write discussion here >> | |
| **Conclusion** | **Mark / 4** |
| << Write conclusion here >> | |

|  |  |  |
| --- | --- | --- |
| **Self Evaluation (answer yes/no/maybe)** | **Your View** | **Markers View** |
| My code works fully |  |  |
| My code is clear and concise |  |  |
| Each function has good comments explaining what it does and its arguments |  |  |
| The code implementing the functions are well explained |  |  |
| I understand the code in these classes |  |  |

|  |
| --- |
| **Write below any issues you have or any questions you would like answered** |
|  |

|  |  |
| --- | --- |
| **Markers Comments** | **Total Mark / 50** |
|  | |