Sustainable Data Science and Artificial Intelligence

Tutorial #5

**Instructions**

The following C code is for a neural network that learns the XOR logical operation. The two key hyperparameters (LEARNING\_RATE and EPOCHS) are defined near the top of the code as follows:

|  |
| --- |
| **C Programming Language** |
| #include <stdio.h>#include <stdlib.h>#include <math.h>// Parameters#define LEARNING\_RATE 0.5#define EPOCHS 10000// Sigmoid activation functiondouble sigmoid(double x) { return 1 / (1 + exp(-x));}// Derivative of sigmoiddouble sigmoid\_derivative(double x) { return x \* (1 - x);}int main() { // Training data (XOR problem) double input[4][2] = { {0, 0}, {0, 1}, {1, 0}, {1, 1} }; double expected\_output[4] = {0, 1, 1, 0}; // Initialize weights and biases with random values double hidden\_weights[2][3] = {{0.5, -0.2, 0.3}, {-0.3, 0.8, -0.5}}; double hidden\_bias[3] = {0.1, -0.1, 0.2}; double output\_weights[3] = {0.6, -0.4, 0.7}; double output\_bias = 0.2; // Training loop for (int epoch = 0; epoch < EPOCHS; epoch++) { double total\_error = 0; for (int i = 0; i < 4; i++) { // Forward pass - Hidden layer double hidden\_layer[3]; for (int j = 0; j < 3; j++) { hidden\_layer[j] = sigmoid(input[i][0] \* hidden\_weights[0][j] + input[i][1] \* hidden\_weights[1][j] + hidden\_bias[j]); } // Forward pass - Output layer double output = sigmoid(hidden\_layer[0] \* output\_weights[0] + hidden\_layer[1] \* output\_weights[1] + hidden\_layer[2] \* output\_weights[2] + output\_bias); // Compute error double error = expected\_output[i] - output; total\_error += error \* error; // Backpropagation - Output layer double output\_delta = error \* sigmoid\_derivative(output); // Backpropagation - Hidden layer double hidden\_delta[3]; for (int j = 0; j < 3; j++) { hidden\_delta[j] = output\_delta \* output\_weights[j] \* sigmoid\_derivative(hidden\_layer[j]); } // Update weights and biases for (int j = 0; j < 3; j++) { output\_weights[j] += LEARNING\_RATE \* output\_delta \* hidden\_layer[j]; hidden\_weights[0][j] += LEARNING\_RATE \* hidden\_delta[j] \* input[i][0]; hidden\_weights[1][j] += LEARNING\_RATE \* hidden\_delta[j] \* input[i][1]; hidden\_bias[j] += LEARNING\_RATE \* hidden\_delta[j]; } output\_bias += LEARNING\_RATE \* output\_delta; } } // Testing the trained network for (int i = 0; i < 4; i++) { double hidden\_layer[3]; // Forward pass - Hidden layer for (int j = 0; j < 3; j++) { hidden\_layer[j] = sigmoid(input[i][0] \* hidden\_weights[0][j] + input[i][1] \* hidden\_weights[1][j] + hidden\_bias[j]); } // Forward pass - Output layer double output = sigmoid(hidden\_layer[0] \* output\_weights[0] + hidden\_layer[1] \* output\_weights[1] + hidden\_layer[2] \* output\_weights[2] + output\_bias); printf("Input: (%.0f, %.0f) -> Output: %.4f\n", input[i][0], input[i][1], output); } return 0;} |

With those settings you should get the following:

|  |  |
| --- | --- |
| **Hyperparameters** | **Outputs** |
| #define LEARNING\_RATE 0.5#define EPOCHS 10000 | Input: (0, 0) -> Output: 0.0164Input: (0, 1) -> Output: 0.9790Input: (1, 0) -> Output: 0.9869Input: (1, 1) -> Output: 0.0146 |

For this tutorial change the LEARNING\_RATE, and record the outputs, as follows:

Learning Rate = 1, Epochs = 10000

Learning Rate = 5, Epochs = 10000

Learning Rate = 10, Epochs = 10000

Learning Rate = 15, Epochs = 10000

Learning Rate = 20, Epochs = 10000

Learning Rate = 25, Epochs = 10000

Learning Rate = 30, Epochs = 10000

Learning Rate = 35, Epochs = 10000

Learning Rate = 40, Epochs = 10000

Learning Rate = 45, Epochs = 10000

1. Identify which settings give you the closest to 0,1,1,0.
2. If the best setting is N, then test if any of the following gives a better answer:

N - 3, N - 2, N - 1, N + 1, N + 2, or N + 3

**Submission Instructions**

Please submit in the following two ways

|  |
| --- |
| Please e-mail me your completed solution, and with filename:*Familyname\_FirstName\_TU850\_1\_SDSaAI\_Tutorial5.docx*e.g.*Gordon\_Damian\_TU850\_1\_SDSaAI\_ Tutorial5.docx*e-mail to Damian.X.Gordon@TUDublin.ie with subject heading as follows: [DT850/1] SDSaAI Tutorial #5 |
| Please submit into Brightspace in:* Assessment
	+ Assignments
		- Tutorial #5
 |

For the email, please include the following message:

|  |
| --- |
| **Damian,****I am a student in your Sustainable DS & AI (CMPU1040) class in the BSc in DS & AI (DT850/1).****Please find attached Tutorial #5****Regards,****Your Name****Student Number.****DT580, BSc in DS & AI** |