

Learning a Simple Function Using an Artificial Backpropagation Neural Network ©  
Wednesday, May 14, 2025, by James Pate Williams, Jr., BA, BS, Master of Software  
Engineering, and PhD Computer Science

My original code was in C and written in the Winter Quarter of 1999 at Auburn University as part of the Machine Learning class taught by Professor Gerry V. Dozier. We used the textbook “Machine Learning” by Tom M. Mitchell © 1997, see Table 4.2 on page 92. I later created a Graphical User Interface C# program in 2011. Our function in the latest C++ application is:

$$f(u, v, x, y) = u^2 + v^2 + x^2 + y^2$$

Since we are using the sigmoid function, we must scale the outputs using the following information and equations:

Unscaled output S [0, M]     Scaled output U [0.1, 0.9]

$$S = A * s + B$$

$$0.1 = B$$

$$0.9 = A * M + 0.1$$

$$A = 0.8 / M$$

$$U = C * u + D$$

$$s = (S - B) / A$$

Here is the input table:

	u	v	x	y
0	0	0	0	0
0	1	2	3	4
0	5	6	7	8
0	0	0	1	0
0	0	0	0	1
0	2	1	0	0
0	1	3	0	5
0	2	2	0	1
0	1	1	1	1
0	3	0	4	1

The output row vector elements are: 30, 174, 1, 1, 5, 35, 9, 4, 26. I am planning on using my C++ code for a much larger single output function.