Long Nguyen

1001705873

1) Training and testing on pendigits dataset, with 2 layers, 10 training rounds

Classification accuracy=0.2245

Training and testing on pendigits dataset, with 4 layers, 40 units per hidden layer, 20 training rounds, sigmoid activation for the hidden layers

Classification accuracy=0.9605

2) $z(x_n) = h(b + w^T x_n)$; where h is the step function h(a) = {0, if a < 0; 1, if a >= 0

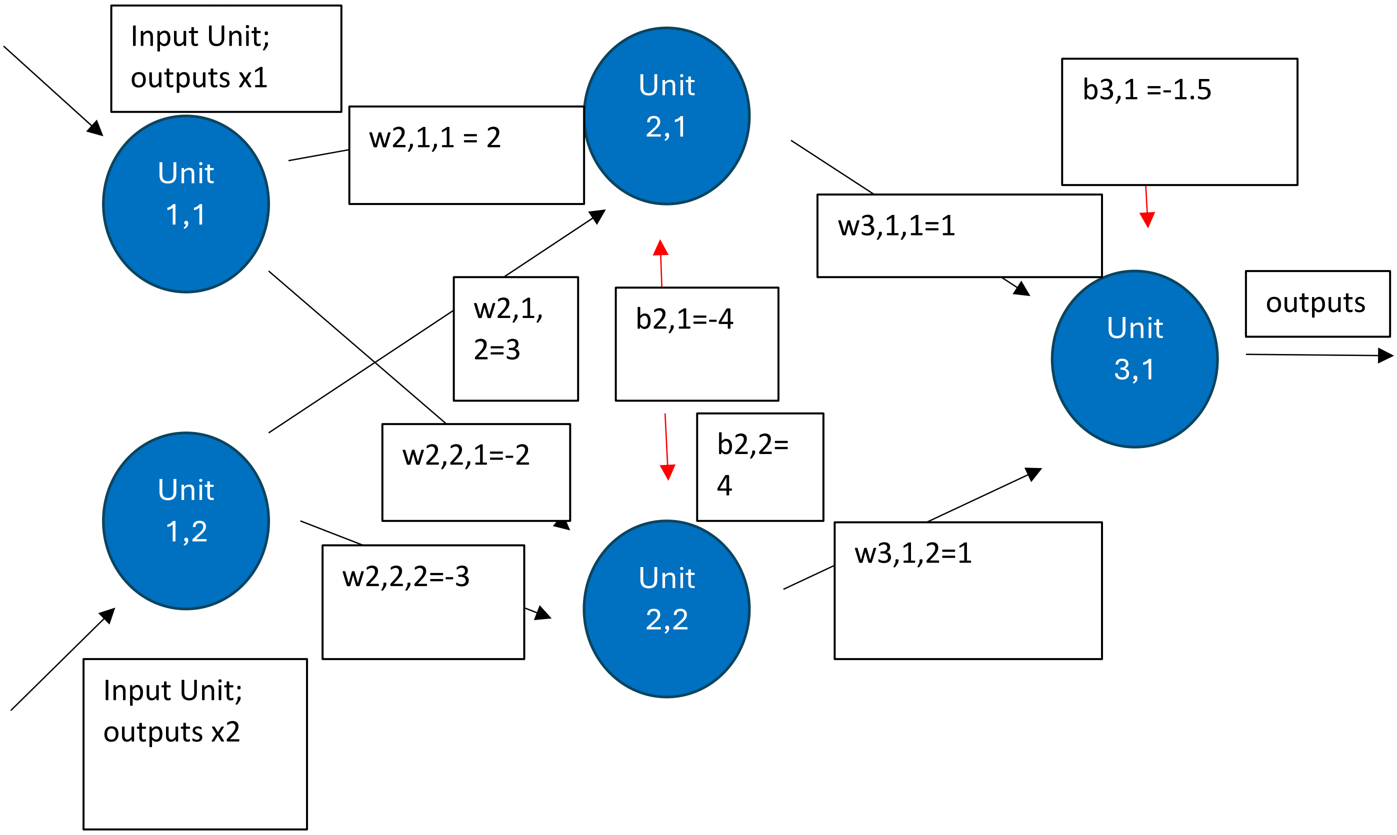
- b = -1.5 (any value less than –1 works)

- $w_1 = w_2 = w_3 = 1$

3) $z(x_n) = h(b + w^T x_n)$; where h is the step function h(a) = {0, if a < 0; 1, if a >= 0

- $b_{2,1} = -4;\ b_{2,2} = 4;\ b_{3,1} = -1.5$

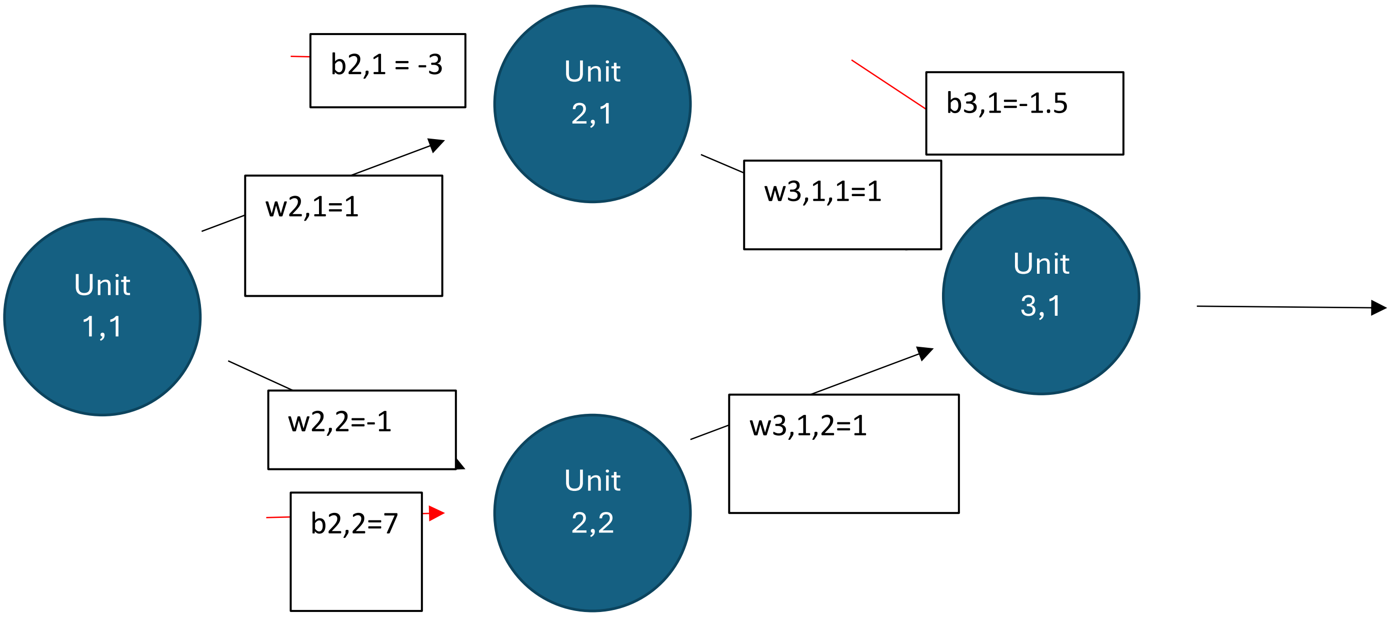
- $w_{2,1,1} = 2,\ w_{2,1,2} = 3,\ w_{2,2,1} = -2,\ w_{2,2,2} = -3,\ w_{3,1,1} = 1, w_{3,1,2} = 1$



4) Yes, it is possible to design a neural network for that problem.

- $b_{2,1} = -3,\ b_{2,2} = 7,\ b_{3,1} = -1.5$

- $w_{2,1} = 1,\ w_{2,2} = 7,\ w_{3,1,1} = 1,\ w_{3,1,2} = 1$



5) If b and w_d were initialized to 0, then computing z(x_n) = h(0) for all x_n since z(x_n) = h(b+w_t*x_n). Depending on what activation function is used depends on what z(x_n) equal, but in general the output will be the same for all inputs. This would then affect the other steps since all the inputs would have no differentiation or importance amongst each other, resulting in a classification accuracy to be a lot lower than if you were to randomly initialize the weights.