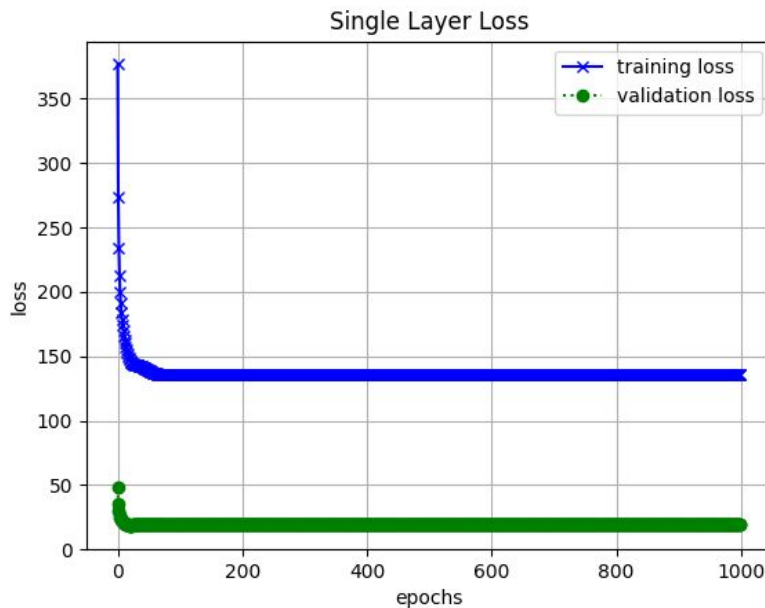


A visualization of the weights you learn for the neurons in the hidden layer of your tuned single-hidden-layer network using the framework function 'VisualizeWeights' (stride value = 2)



Tune the parameters until you have a single layer network with validation set accuracy greater than 85%. Report on the parameters you used. Include a chart with training and validation set loss on the y-axis vs epoch number on the x-axis. In 1-2 sentences indicate the approach you took to finding hyperparameters that work. Include a brief log of the hyperparameters you tried.



### Hyperparameters:

maxEpochs = 1000

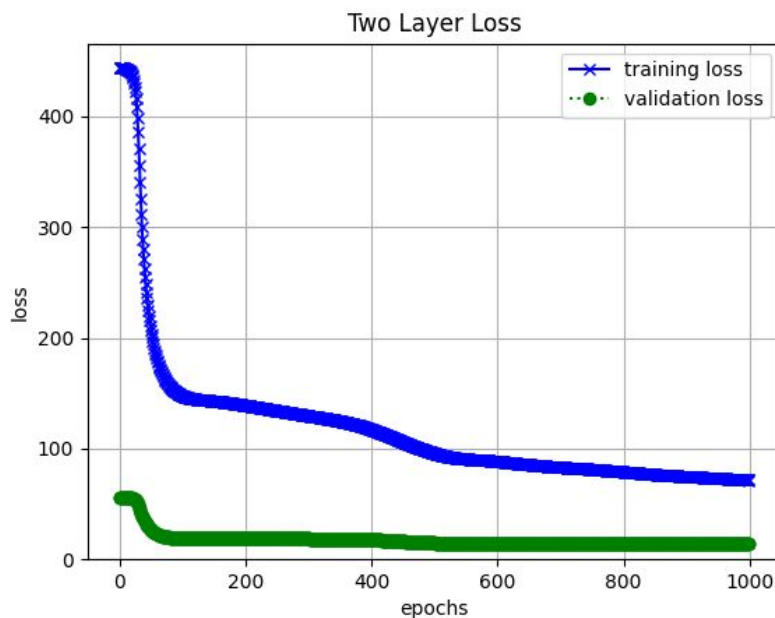
step = 1.0->0.1

convergence = 0.1

momentum = 0.0->0.1

At higher step sizes, I saw the training losses oscillate, which means it was unable to reach a minimum optimally, so I stepped back the step size. Then to make things move a bit faster, I added momentum. The accuracy I achieved was 89%.

Tune the parameters until you have a two layer network with validation set accuracy greater than 89%. Report on the parameters you used. Include a chart with training and validation set loss on the y-axis vs epoch number on the x-axis. In 1-2 sentences indicate the approach you took to finding hyperparameters that work. Include a brief log of the Hyperparameters you tried.



### Hyperparameters:

maxEpochs = 1000

step = 0.1->0.01

convergence = 0.1->0.01

momentum = 0.1

I saw the training loss oscillate again, which means it was unable to reach a minimum, so I made the step size even smaller. Then I saw that it was assuming the model converged too early, so I lowered the threshold for convergence. The accuracy I achieved was 92%.