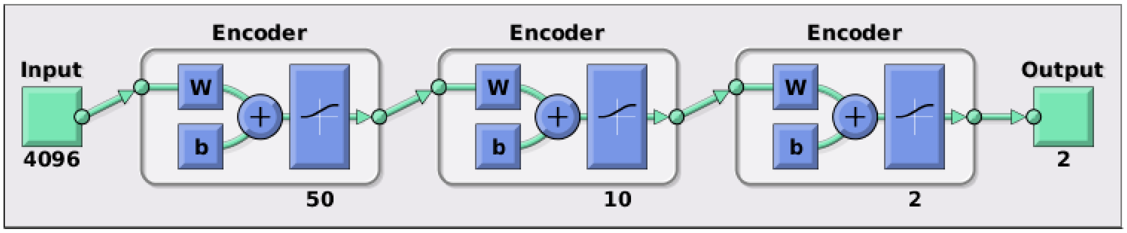
Final Exam Problem 4

Bryn Louise

```
% Load Data
X = load('CatsDogs.mat');
X = X.X;

% Define Targets
T= [ones(1,99) zeros(1,99); zeros(1,99) ones(1,99)];

% Randomize Order of Data and Targets
[m,n] = size(X);
idx = randperm(n);

Xrand = X(:, idx(1:n));
Trand = T(:, idx(1:n));
```

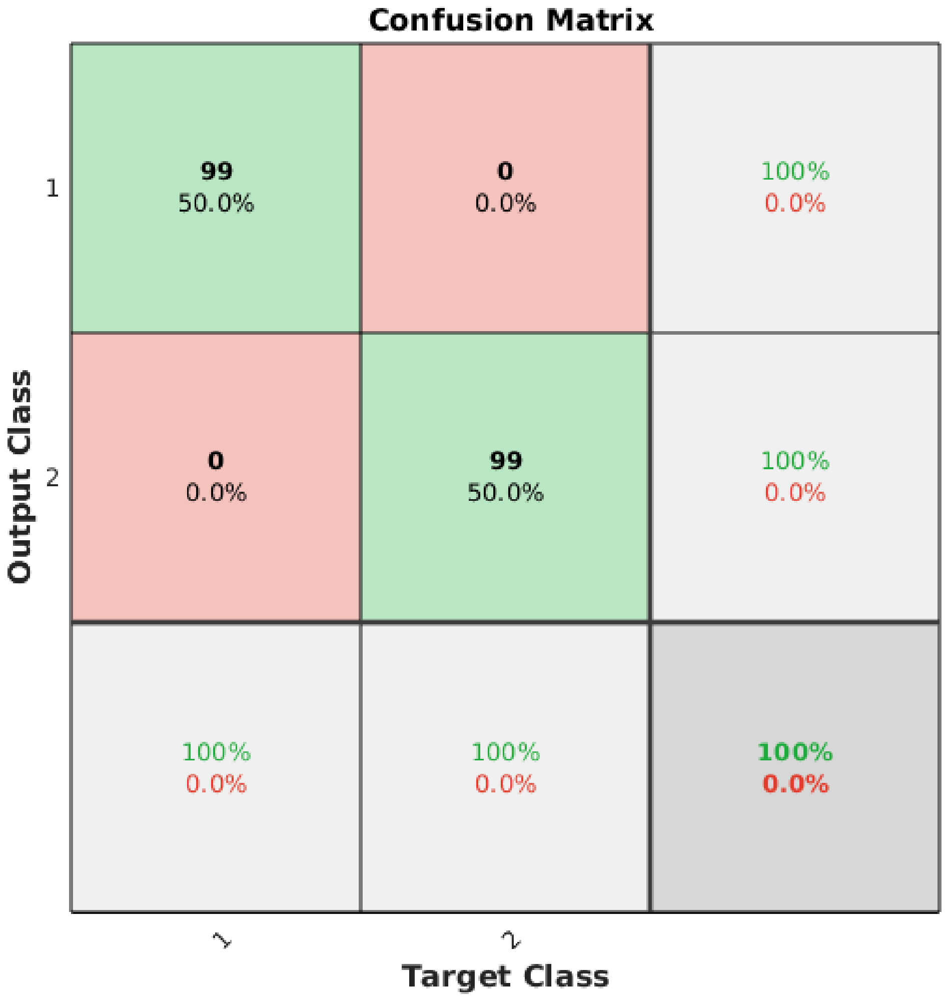
Build Network

```
hiddenSize1 = 50;
autoenc1 = trainAutoencoder(Xrand, hiddenSize1,...
    'L2WeightRegularization',0.004,...
    'SparsityRegularization', 4,...
    'SparsityProportion', 0.15);

feat1 = encode(autoenc1, Xrand);

hiddenSize2 = 10;
autoenc2 = trainAutoencoder(feat1, hiddenSize2,...
    'L2WeightRegularization',0.004,...
    'SparsityRegularization', 4,...
    'SparsityProportion', 0.15);

feat2 = encode(autoenc2, feat1);

hiddenSize3 = 2;
autoenc3 = trainAutoencoder(feat2, hiddenSize3,...
    'L2WeightRegularization',0.004,...
    'SparsityRegularization', 4,...
    'SparsityProportion', 0.15);

feat3 = encode(autoenc3, feat2);

stackednet = stack(autoenc1, autoenc2, autoenc3);

view(stackednet)
```



Test Network

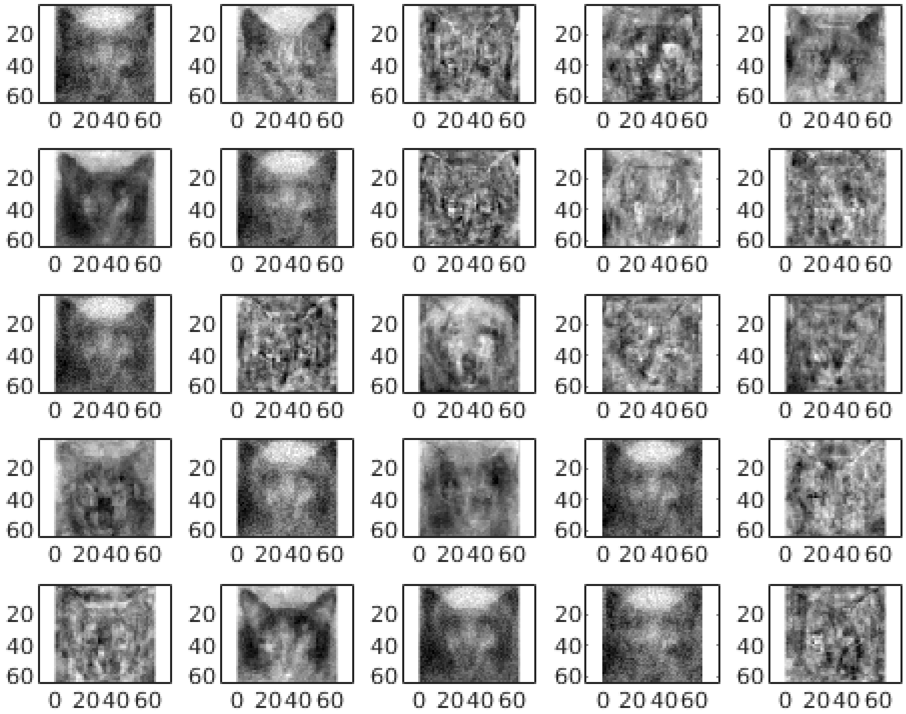
```
% Randomize data and targets again for fine tuning
[m,n] = size(Xrand);
idx = randperm(n);

Xrand2 = Xrand(:, idx(1:n));
Trand2 = Trand(:, idx(1:n));

% Perform Fine Tuning
stackednet = train(stackednet, Xrand2, Trand2);

% Ranomize data and targets a third time for testing
[m,n] = size(Xrand2);
idx = randperm(n);

Xrand3 = Xrand2(:, idx(1:n));
Trand3 = Trand2(:, idx(1:n));

% Test the network
y = stackednet(Xrand3);
plotconfusion(Trand3, y);
```



Output Class \ Target Class	1	2	
1	99 50.0%	0 0.0%	100% 0.0%
2	0 0.0%	99 50.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%

Visualize Weights

```
W=autoenc1.EncoderWeights;

% Visualize rows of Weights
figure(4)
for i = 1:25
    subplot(5,5,i)
    imagesc(reshape(W(i, :), 64, 64));
    axis equal; colormap(gray);
end
```



Published with MATLAB® R2018a