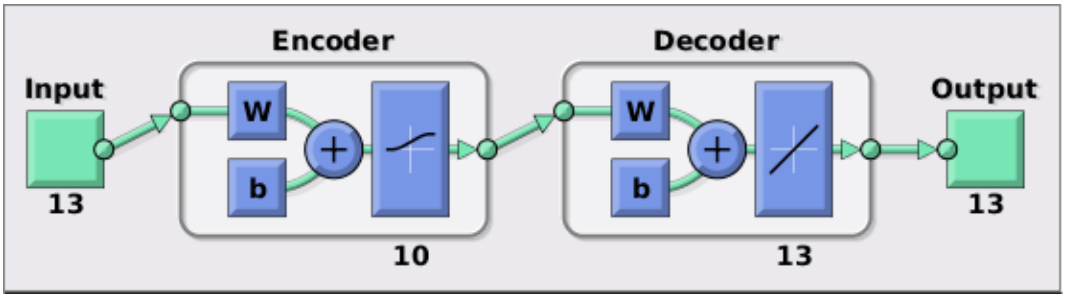
Contents

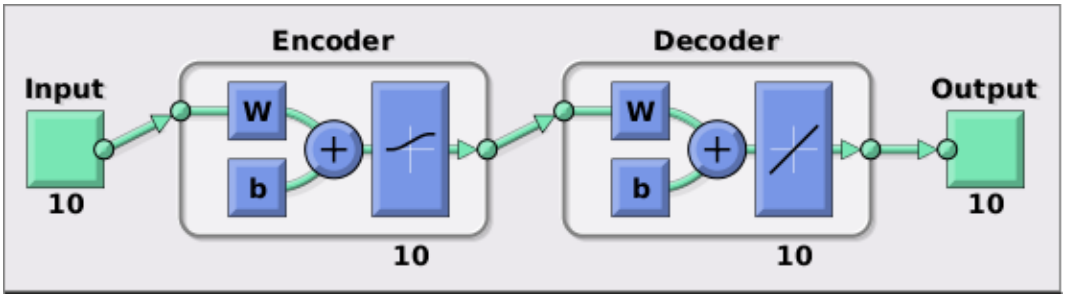
Example with an Autoecoder

```
% The data:
[X,T] = wine_dataset;

% Construction of the first autoencoder:
hiddenSize = 10;
autoenc1 = trainAutoencoder(X,hiddenSize,...
    'L2WeightRegularization',0.001,...
    'SparsityRegularization',4,...
    'SparsityProportion',0.05,...
    'DecoderTransferFunction','purelin');

% Extract the "features" in the hidden layer
features1 = encode(autoenc1,X);

view(autoenc1);
```



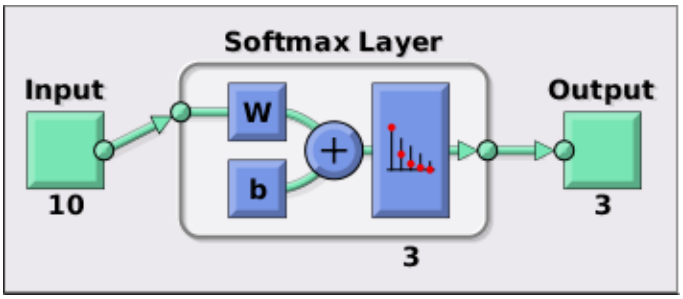
Next autoencoder

```
% Construction of a second autoencoder
hiddenSize = 10;
autoenc2 = trainAutoencoder(features1,hiddenSize,...
    'L2WeightRegularization',0.001,...
    'SparsityRegularization',4,...
    'SparsityProportion',0.05,...
    'DecoderTransferFunction','purelin',...
    'ScaleData',false);

% Extract the features on the hidden layer
features2 = encode(autoenc2,features1);

view(autoenc2);
```



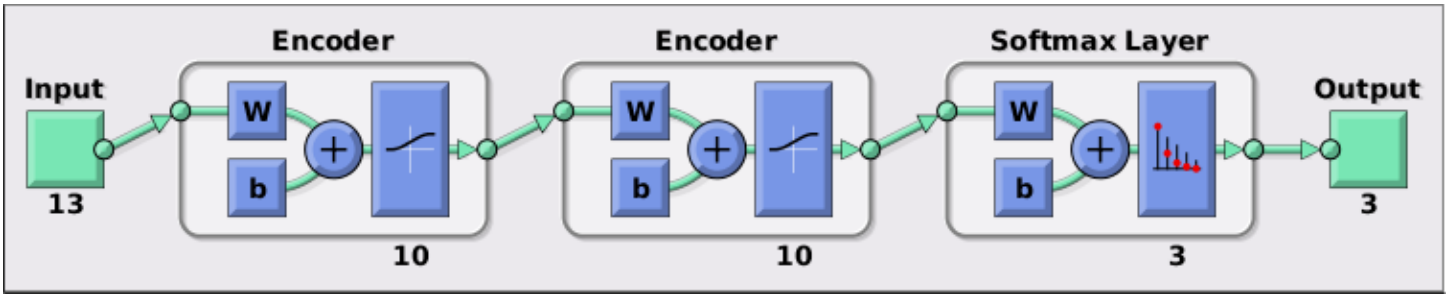
The softmax layer comes from the softmax algorithm we used earlier

```
%Train a softmax layer:
softnet = trainSoftmaxLayer(features2,T,'LossFunction','crossentropy');

view(softnet);
```



Construct the deep network

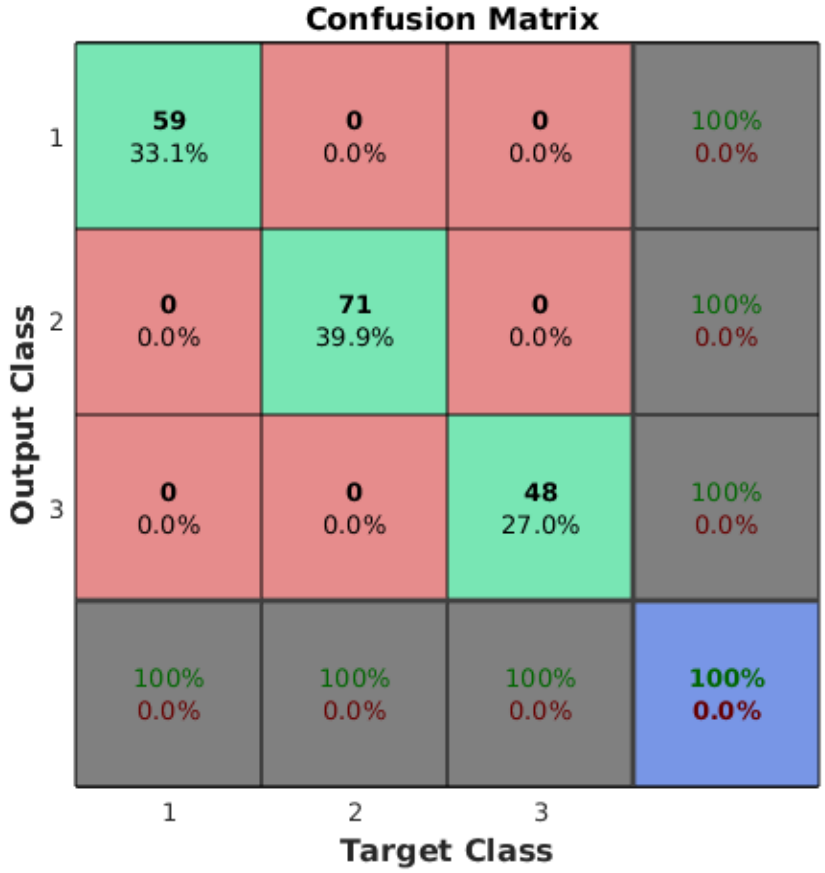
```
% Stack the encoders together to get the deep network:
deepnet = stack(autoenc1,autoenc2,softnet);

view(deepnet);
```



Train the deep net (This is "fine tuning" the net)

```
[deepnet,tr]=train(deepnet,X,T);

wine_type=deepnet(X);
```

```
plotconfusion(T,wine_type)
```



Published with MA TLAB® R2016a