Construct Deep Network Using Autoencoders

Load the sample data.

[X,T] = wine\_dataset;

Train an autoencoder with a hidden layer of size 10 and a linear transfer function for the decoder. Set the L2 weight regularizer to 0.001, sparsity regularizer to 4 and sparsity proportion to 0.05.

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);

Train a second autoencoder using the features from the first autoencoder. Do not scale the data.

hiddenSize = 10;

autoenc2 = trainAutoencoder(features1,hiddenSize,...

'L2WeightRegularization',0.001,...

'SparsityRegularization',4,...

'SparsityProportion',0.05,...

'DecoderTransferFunction','purelin',...

'ScaleData',false);

Extract the features in the hidden layer.

features2 = encode(autoenc2,features1);

Train a softmax layer for classification using the features, features2, from the second autoencoder, autoenc2.

softnet = trainSoftmaxLayer(features2,T,'LossFunction','crossentropy');

Stack the encoders and the softmax layer to form a deep network.

deepnet = stack(autoenc1,autoenc2,softnet);

Train the deep network on the wine data.

deepnet = train(deepnet,X,T);

Estimate the wine types using the deep network, deepnet.

wine\_type = deepnet(X);

Plot the confusion matrix.

plotconfusion(T,wine\_type);

