ECEN 649 Pattern Recognition – Spring 2019

Final Computer Project

Due on: May 4

Assignment 1: In this part we will apply classifier design, error estimation, and feature selection techniques to a gene expression data set from the following cancer classification study:

van de Vijver,M.J., He,Y.D., van't Veer, L.J., et al. (2002), "A gene-expression signature as a predictor of survival in breast cancer." *New Eng. J. Med.*, 347, 1999-2009.

This paper analyzes gene expression in breast tumor biopsies from 295 patients. The authors performed feature selection to obtain 70 genes; hence, the full data matrix is 70×295. This is a **retrospective study**, meaning that the patients were tracked over the years and their outcomes recorded. Using this clinical information, the authors labeled the patients into two classes: the "good' prognosis" group were disease-free for at least five years after first treatment, whereas the "bad prognosis" group developed distant methatasis within the first five years. Of the 295 patients, 216 belong to the "good-prognosis" class, whereas the remaining 79 belong to the "poor-prognosis" class.

The gene expression data was randomly divided into a training data set (containing 80 points, with 40 points from each class) and testing data set (containing the remaining 215 points). The latter will be used for test-set error estimation of the true classification error. The tab-delimited files are available on the TAMU Google Drive at `http://bit.ly/2jaGCkg`. The first row contains the gene symbol names, whereas the first column contains the patient ID. The last column contains the label (1 = good prognosis, 0 = poor prognosis).

We are going to search for gene feature sets and design classifiers that best discriminate the two prognosis classes on the training data, and use the testing data to determine their accuracy.

We will consider the following classification rules:

- Linear SVM, $C = 1$.
- Nonlinear SVM with RBF kernel, $C = 10$.
- Neural network with 2 hidden layers of 5 neurons each and logistic nonlinearities.

(Hint: Use the modules `svm` from `sklearn` and `MLPClassifier` from `sklearn.neural_network` with the solver 'lbfgs.') The criterion for the search will be simply the resubstitution error estimate of the designed classifier for the current feature set (wrapper feature selection).

We will consider the following feature selection methods.

- Top 2 genes (exhaustive search).
- Top 3–5 genes (sequential forward search).
- All genes (no feature selection).

Therefore, you will use 5 different gene sets with each of the 3 classification rules, for a total of 15 classifiers. Submit a table with the results, where each row corresponds to one of the 15 classifiers and should contain the gene set found (or "all genes"), the corresponding resubstitution error estimate on class 0, class 1 and total, as well the test-set estimate on class 0, class 1, and total. Explain what you see. For example, how do you compare the different classifiers based on the dimensionality and the estimates of the true error? What can you say about the resubstitution error estimator based on these results? How do you compare the error rates on class 0 and 1?

Assignment 2: For this assignment, we will use the Carnegie Mellon University Ultrahigh Carbon Steel (CMU-UHCS) dataset in We will use the Carnegie Mellon University Ultrahigh Carbon Steel (CMU-UHCS) dataset in

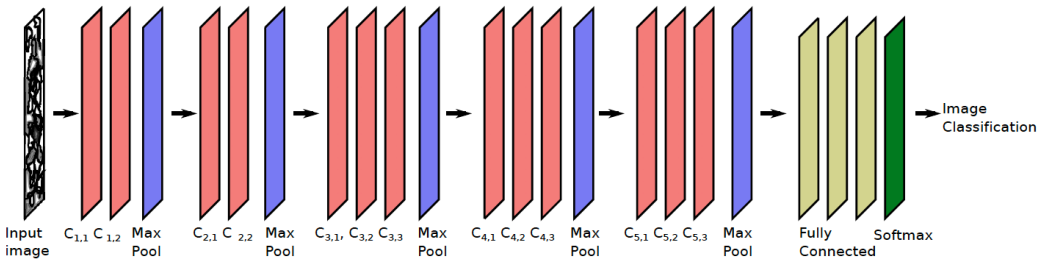
> B. DeCost, T. Francis and E. Holm (2017), "Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures." arXiv:1702.01117v2.

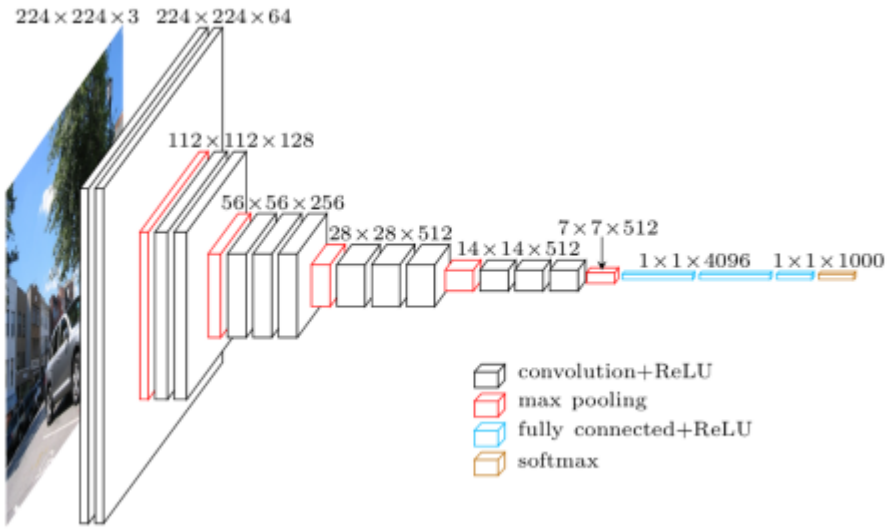
The data set is available on the TAMU Google Drive at `http://bit.ly/2jaGCkg`. There are three files: a ZIP file containing the raw images and two excel files containing the labels and sample preparation information. Please read DeCost's paper to learn more about the data set.

We will classify the micrographs according to **primary microconstituent**. There are a total of seven different labels, corresponding to different phases of steel resulting from different thermal processing. In this assignment, we will use only the spheroidite (374 micrographs), network (212 micrographs), and pearlite (124) micrographs. The training data will be **the first 100 data points** in the spherodite, network, and pearlite categories. The remaining data points from these categories will compose the test sets (more below).

The classification rule to be used is a Radial Basis Function (RBF) nonlinear SVM classification rule. We will use a *one-vs-one* approach to deal with the multiple labels, where each of the 3 classification problems for each pair of labels are carried out. Given a new image, each of the 3 classifiers is applied and then a vote is taken to achieve a consensus for the most often predicted label. If there is a 3-way tie, then no label is output.

To featurize the images, we will use the pre-trained VGG16 deep convolutional neural network (CNN), which has the following architecture:





We will ignore the fully connected layers, and take the features from the **max pool layers** only (following the the intermediate layers **C1,2 C2,2 C3,3 C4,3 C5,3**), using the "channels" mean value as the feature vector (each channel is a 2D image corresponding to the output of a different filter). This results in feature vectors of length 64, 128, 256, 512, 512, respectively (these lengths correspond to the number of filters in each layer and are fixed, having nothing to do with the image size). In each pairwise classification experiment, we will select one of the five layers according to the best 10-fold cross-validation error estimate.

You are asked to obtain:

(a) The convolution layer used and the cross-validated error estimate for each of the three two-label classifiers.

(b) The test error rates on the unused micrographs of three categories, for the three pairwise two-label classifiers and the multilabel one-vs-one voting classifier described previously. For the pairwise classifiers use only the test micrographs with one of the two labels used to train the classifier. For the multilabel classifier, use the test micrographs with one of the three labels in the training data.

In each case above, interpret your results. Implementation should use the Scikit-Learn and Keras python libraries.