Neural Networks (NNs) in Remote sensing
Introduction to Multi-Layer Perceptrons (MLP)
2007.4.19

The purpose of this session is to introduce neural networks for classification tasks with remote sensing data. *Feedforward multilayer perceptrons* are mainly discussed. An example is given using *multisource* data including both *spectral* and *ancillary* data (topographic).

1. The data available is in c:\PODEPRO_student\NN\
   - `landsatc1.tif`
   - `landsatc2.tif`
   - `landsatc3.tif`
   - `landsatc4.tif`
   - `landsatc5.tif`
   - `landsatc6.tif`
   - `landsatc7.tif`
   - `dem.tif`
   - `slope.tif`
   - `aspect.tif`
   - `clc.tif (CORINE land cover data)`

2. make C:\PODEPRO_student\NN\matlab your current directory in matlab.
3. in order to be able to form input and output *vectors* that the data has to:
   - be in the same coordinate system (overlap)
   - have the sample spatial resolution

4. The first set is to sample the data and form *train/test and validation sample sets.* Open mail.m in matlab.

5. Stratified sampling is used since it is suitable when each class cover different areas on the ground (that is the general case). If the representation is not equal some of the classes are favored and some not. Also the classification result might be overall compromised (rf. K-hat).

6. open stratified_sampling.m The data in each band has to be normalized. A common choice is [0,1] and [-1,1]. See function mapminmax in matlab's help.

7. what is the percentage of samples to pick? The percentage is bounded by computing resources. Too few samples -> inefficient representation of classes. Too many classes -> inefficient use of computational resources.

8. open divide.m This script is used to divide the samples selected into three groups.
   - training (to be actually used during training in order to fit the model to the data)
   - testing (to assess the accuracy during training by an independent set. Optimal *termination criterion*.
   - Validation. Independent of any training process. Presented only after weights have frozen. To assess the accuracy of the overall process. Major handicap if not used (i.e. the two sets approach).
9. How the samples are split? Equally one option. Less in training more in validation and testing provided that results permit.
10. Notice that for each set we define a *structure* that consists of two subsets: the *input vectors* and the *output vectors* (targets). Inputs are normalized and output is coded.
11. The main NN is defined in PODEPRO__NN.m. Open it in matlab
12. *newff* is the statement to create a new feed forward neural network (MLP)
13. one of the main decisions has to do with the number of nodes in the hidden layer.
14. notice the declaration of the domain of each input dimension ([-1,1]).
15. notice the declaration of transfer functions between each layer
    - here tan-sigmoid function is used
    - see the graph of this function in matlab's help page
    - what is the practical meaning of such a funtion
16. notice the (indirect) way the number of hidden layers is declared.
17. Notice the training algorithm
    a. *A fast implementation version of backpropagation* algorithm *(Trainlm*) is used.

b. *The method as well as standard BP is described in matlab's help.*

18. notice the declaration of the number of output nodes.
19. Mean square error is used as the performance criterion (much like in linear regression)
20. The learning rate is an important parameter that has to be set usually after trial and error
21. Also the momentum which allows the network to respond also to recent trends in the error surface. Acting like a lowpass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum
22. termination criterion
    - minimum training goal
    - max # training epochs.
    - Training Vs overfitting is best judged by using a test set (draw diagram)
23. when a network is initialized its weights are set to small random values (hence the noisy fitness evaluation problem)
24. train is the command to train the NN based on the training algorithm and parameters set
25. when training is over the weitghs are frozen and the validation set is presented to the network for accuracy assessment (sim)
26. in remote sensing one of the best known accuracy assessment tools is the accuracy matrix (Congalton paper).
27. run main.m
28. observe the training process and the results
29. view classified image
30. run a second time with no modifications. Are the results consistent/the same?
31. experiment with changing of parameters
32. is it possible to see the parameters one by one? (i.e. first find one and fix it, then find another one and fix it and so on)
33. is it possible after we have found the best set of parameters to apply them in an other classification problem? Why?