EECS/EEAP 484 Computational Intelligence, Fall 2018
Problem Set 2: Feedforward Neural Networks with Error Backpropagation
Assigned: 9/24/18
Due: 10/1/18

Neural networks with nonlinear activation functions and hidden layer(s) can behave as universal function approximators. In this problem set, you will set up a neural network with two inputs (plus a bias), a number of "hidden" nodes in an intermediate layers (plus bias) and two output nodes. The objective is to fit data points with a smooth function by adjusting the weights of the connections between layers.

The training data for this assignment is in arm_xy.dat, which contains data corresponding to 2-D inputs with 2-D outputs. This data is based on forward kinematics for a 2-DOF robot arm with closed-chain, 4-bar-linkages with lead-screw actuators. The kinematics relate actuator displacements (analogous to muscle stretch and stretch receptors) to hand position (x-y coordinates in Cartesian space).

Your neural-net code should fit the training data with smooth surfaces. There are two main functions provided: ps2_fdfwd_net.m and ps2_fdfwd_net_presets.m. These functions contain debug lines of code that can be enabled or commented out to help you debug your computations of dE/dW.

The key function to edit is: compute_dW_from_sensitivities.m. You need to fill in this function with the appropriate algorithms for recursive computation of bias-sensitivity (delta) vectors and corresponding synaptic sensitivities. (The same function will work with both versions of the main program). You should confirm that your computations are consistent with the numerical estimates before attempting training or experiments.

The difference between the two main functions is that ps2_fdfwd_net uses a single hidden layer, whereas ps2_fdfwd_net_presets has 3 hidden layers. However, the weights and biases for the first 2 layers are set manually and frozen. Learning only takes place with weights into and from the last hidden layer. Also, ps2_fdfwd_net_presets uses a *linear* activation function for the last layer. (At present, ps2_fdfwd_net_presets only fits one component of the output. You should add the second component of the output).

Note that you can change the activation function for any given layer by specifying an activation code (currently, only 1=logsig and 2=linear are options).

Once your code is working, perform experiments on the data. Report on:

- Influence of learning factor
- Accuracy achieved in training (e.g., vs number of interneurons and number of iterations)
- Differences between 1 vs. 3 hidden layers.
- Alternative values for the fixed weights
- Show plots of your results
- Report on any other observations or variations