12/27/2013

* Initial commit of project solution and other important base files
* Researched the problem set. Researched error backpropagation and feed forward neural networks and researched comparisons among sigmoid activation functions.
* Added sigmoid activation class where each activation class is described by lambda definition of its mathematical form and the derivative thereof. Logistic, Hyperbolic Tangent, and Linear are defined in the neural network class.
* Moved the sigmoids into their respective classes
* Implemented abstract neurons
* Bias Neuron, Hidden Neuron, Input Neuron, Output Neuron implementations
* Cleaned the solution recursively and forced uniform white spacing.
* Fundamentally changing the model through which Neurons calculate values. Neurons now have void UpdateOutput(...) and void UpdateError(...) which handle those parameters internally instead of Get[Error/Output](...). The getters were then replaced by properties with protected setters and public getters. All changes occurred subsequently to subclasses.
* Added a random generator to create random numbers pertinent to the neural network
* [Added the Connection class. The Connection class contains weight, pos](https://github.com/MadcowD/ISEFNeuralNetwork/commit/0978d0808d9d7bfc00e1c65bbc425e43a7ccacbb)terior neuron, and anterior neuron properties. The update weight rule for individual connections was also implemented in double UpdateWeight()
* Added FeedForward() function to Connection.cs. It feeds the product of output from the anterior neuron and the weight of the connection forward to the anterior neuron.
* Formatted the BiasNeuron.cs
* Completed Layer-Neuron and Layer-Connection initialization
* Implemented the feed forward functionality of the network
* Cleaning the solution

12/28/13

* Implemented Error Backpropagation
* Calculates global error
* Adjusts output neuron to calculate proper error
* Implemented full learning rate control
* Implemented updating weights using learning rate
* Taught the network XOR
* Created a Gaussian distribution class, weights are set with a normal distribution between -1 and 1.
* Added net reset to Bias Neurons
* Added correct sum squared error calculation to the Network class.
* Switched difference terms in output error calculation
* Implemented Gaussian switch and changing weight sign
* Added momentum to weight training
* Added train method to Network Class
* Changed the constructor of the Network class
* Fixed a bug where neurons were not being reset
* Added None to Sigmoid for the input layer
* Finished base implementation, XNOR training and testing, with working bias neurons
* Cleaned the solution

1/12/2014

* Created Neural Network Library
* Cleaned the solution

1/16/2014

* Finished the DataPoint abstract class for training and neural analysis

1/20/2014

* Final trainer implementation with successful XOR network (2,4,1) added to the solution
* Removing debug parameter to train making it solution dependent

1/21/2014

* Added nudging functionality to the neural network

2/1/2014

* Added cancer dataset
* Fixed XOR program
* Fixed logistic curve of the trainer
* Cleaned the solution

2/5/2014

* Added dataset shuffling
* Added training error history with more accurate implementation and less processing power
* Added hyperbolic step function to sigmoids
* Moved nudging threshold down
* Added new dataset
* Fixed a fatal flaw with error backpropagation algorithm which made the convergence working
* Added RPROP connection and implemented initial RPROP, but discontinued RPROP algorithm
* Fixed the solution file
* Cleaned the solution
* Removed the readkey

2/8/2014

* Added saving and loading weights

2/9/2014

* Initial App project started
* Data calculations added

2/11/2014

* Final commit of the app
* Committing initial application for use with cancer diagnostics

2/15/2014

* Removed conflict
* Cleaned the repo to account for normalized master networks
* Removed the cancer dataset
* Constructed XOR for working dataset
* Removed Wisconsin from solution
* Reduced and then increased the network size

2/16/2014

* Cleaning the project and leaving it in its most basic form
* Initial commit of clean BPNN
* Committing basic folder structure
* Establishing further directory structure with more changes
* Created initial commit of the numerical experiment
* Added an experiment class for running multiple tests
* Added first learning rate experiment
* Cleaned the solution

2/18/2014

* Worked more on the learning rate experiment and moved the dataset into the experiment folder
* Dataset and separation into training and testing dataset for state
* Created CancerData.cs to separate datasets and practicing thereof through Linq

2/20/2014

* Finished experimental analysis
* Created a control experiment
* Finalized learning rate experiment

2/21/2014

* Created new experiments for different parameters
* Researched about the input data information of the datasets

2/22/2014

* Fixed the hidden layer code
* Implemented nudging with standard deviation
* Add project experiment files into neural network class
* Fixed problems in experimentation code
* Added automatic stopping of training when nudging should usually occur
* More additions for proper analysis of error
* Optimized the network
* Worked on experimentation
* Added imaging project
* Committing the CS project
* Added new constants into the network
* Step Experiment
* Made 10 base networks for experimentation
* Further research and addition of comments
* Updated learning rate experiment
* Cleaned the output and solution

2/23/2014

* Added low pass experimentation
* Did the smallest experiment
* Conducted momentum experiment

2/28/2014

* Added conclusion experiment

3/1/2014

* Worked on the conclusion experiment
* Removed unnecessary output data
* Added website to the solution

3/2/2014

* Added covariance experiment data
* Cleaned the history
* Finished Wavelet transform
* Researched Gabor Filters
* Finished conclusion and worked on the website

3/4/2014

* Did image Dataset preprocessing

3/7/2014

* Worked on submission forms
* Completed initial submission form for proportionality test
* Added the second FNA test (detailed)
* Worked on the Journal

3/8/2014

* Added a diagnostic choice page and renamed diagnosis page to P-FNA page
* Committed the conclusion and step experiment
* Reorganized images and fixed image preprocessing
* Finished D-FNA portion of app

3/9/2014

* Finished D-FNA testing for less accuracy
* Created image experiment and finished it
* Continued work on the journal
* Created Graphs
* Created the work log