**Summary: training a machine learning algorithm with functional movement data, solved as a pattern-recognition problem using a two-layer feed-forward network.**

Intro

In pattern recognition problems, you want a neural network to classify inputs into a set of target categories. Neural pattern recognition creates and trains a network of data, and evaluate its performance using cross-entropy and confusion matrices. Minimizing Cross-Entropy results in good classification. A confusion matrix indicates the fraction of samples which are misclassified.

A two-layer feed-forward network, with sigmoid hidden and softmax output neurons, can classify vectors arbitrarily well, given enough neurons in its hidden layer. The network is trained with scaled conjugate gradient backpropagation.

The measurements for each subject included maximum values, minimum values, range values, imbalance values, time (%) of max value, time (%) of min value. For network training, the only features that were used were the imbalance values\* of each of the three specified segments. \**(For explanation of how imbalance values were obtained, refer to section 4.1.4 of Applying Machine Learning Techniques to Prediction of Low Back Pain Using A Multi-Segment Spine Model by Zirui Qiu).* The segments that were specified were:

* LLPelX (Lower Lumber to Pelvis, frontal plane)
* LTLLX (Lower Thoracic to Lower Lumbar, frontal plane)
* UTLLX (Upper Thoracic to Lower Lumbar, frontal plane)

These segments were specified due to the fact that they displayed a significant difference between LBP and Non-LBP subjects during statistical analysis.

The pool contained 118 samples. Prior to network training, 10 samples were removed from the pool in order to conduct further manual testing of independent subjects. For network training, 75% of the samples were used for training: they are presented to the network during training, and the network is adjusted according to its error. 20% were used as validation samples: used to measure network generalization, and to halt training when generalization stops improving. 5% of the samples have no effect on training and were used to provide an independent measure of network performance after training. Different numbers of hidden neurons were experimented with, but typically 90-100 hidden neurons provided increased accuracy.

A typical test for data: Input is a 3x108 matrix, representing static data: 108 samples of 3 features. Target is a 2x108 matrix, representing static data: 108 samples of 2 targets (LBP vs Non-LBP).

All of the trials are in the Excel document attached.

Manual Testing of Passe Data (Appendix 1)

Manual testing of the 10 samples removed from the pool using the trained algorithm was conducted.

Test 1 pulled out samples 22 to 31. The three different trained algorithms are described as follows:

Function 1:

A trained network using 90 hidden neurons. 81 samples trained with 4.3% error, 22 samples validated with zero error, 5 samples independently tested with zero error.

Function 2:

A trained network using 92 hidden neurons. 81 samples trained with 1.9% error, 22 samples validated with zero error, 5 samples independently tested with zero error.

Function 3:

A trained network using 105 hidden neurons. 81 samples trained with 3.1% error, 22 samples validated with zero error, 5 samples independently tested with zero error.

Function 4:

A trained network using 95 hidden neurons. 81 samples trained with zero error, 22 samples validated with zero error, 5 samples independently tested with zero error.

Manual Testing of Arabesque Data (Appendix 2)

Manual testing of the 10 samples removed from the pool using the trained algorithm was conducted.

Test 1 pulled out samples 22 to 31. The three different trained algorithms are described as follows:

Function 1:

A trained network using 100 hidden neurons. 81 samples trained with 13.6% error, 22 samples validated with 13.6% error, 5 samples independently tested with 20% error.

Function 2:

A trained network using 91 hidden neurons. 81 samples trained with 8.6% error, 22 samples validated with 13.6% error, 5 samples independently tested with 20% error.

Function 3:

A trained network using 96 hidden neurons. 81 samples trained with 7.4% error, 22 samples validated with 4.5% error, 5 samples independently tested with 20% error.

Function 4:

A trained network using 105 hidden neurons. 81 samples trained with 14.8% error, 22 samples validated with 9.1% error, 5 samples independently tested with zero error.

Function 5:

A trained network using 75 hidden neurons. 81 samples trained with 13.6% error, 22 samples validated with 22.7% error, 5 samples independently tested with zero error.

Manual Testing of FMS Data (Appendix 3)

Manual testing of the 10 samples removed from the pool using the trained algorithm was conducted.

Test 1 pulled out samples 22 to 31. The three different trained algorithms are described as follows:

Function 1:

A trained network using 90 hidden neurons. 81 samples trained with 19.8% error, 22 samples validated with 9.1% error, 5 samples independently tested with zero error.

Function 2:

A trained network using 90 hidden neurons. 81 samples trained with 12.3% error, 22 samples validated with 9.1% error, 5 samples independently tested with 20% error.

Function 3:

A trained network using 89 hidden neurons. 81 samples trained with 14.8% error, 22 samples validated with 18.2% error, 5 samples independently tested with zero error.

**Appendix 1: Results of Manual Testing of Passe Data**



**Appendix 2: Results of Manual Testing of Arabesque Data**



**Appendix 3: Results of Manual Testing of FMS Data**

