

Musical Instrument Identification Using Wavelets and Neural Networks

Jeffrey Livingston
Nathan Shepard

EE371D Intro. To Neural Networks
Electrical and Computer Engineering Department The
University of Texas at Austin

Fall 2005

1.0 Introduction

In many instrumental musical genres, especially those featuring improvisation, a natural way in which listeners divide up a song into logical subsections is to segment the song according to which instrument is soloing during a given section. Improvisational Jazz is a prime example of such a genre, where performances usually consist of a sequence of individual solos. A system which could analyze an audio recording, locate solo sections and identify the soloing instrument could allow users of digital audio playback devices to easily navigate their way to sections of interest in a song. Such a system could also be used to do content based searches of online music, based on the presence or absence of solos by particular instruments. In this paper we present a method that performs the fundamental classification task of such a system. What follows is a description a method for analyzing a short clip of music, and identifying the dominant, i.e. soloing instrument using a combination of wavelet transform analysis for preprocessing and a neural network for classification.

2.0 Wavelets and Audio Signal Content Analysis

2.1 STFT Analysis and Its Drawbacks

The majority of techniques currently used for audio content description are based on the short time fourier transform (STFT) for time-frequency analysis [1]. The STFT is a very powerful tool for analyzing the frequency spectrum of a time varying signal, and is capable of producing very detailed information about the frequency content of a signal. Despite its considerable usefulness for time-frequency analysis, there are a number of drawbacks in using the STFT for analyzing audio signals [2]. Firstly, the analysis produces equally spaced frequency bands, which does not correspond to humans' logarithmically spaced perception of frequencies. To get acceptable frequency resolution in low frequencies, with respect to human perception, long analysis windows must be used, which results in loss of temporal resolution and excessive resolution in high frequencies. Therefore, the STFT is quite inefficient for this purpose. Secondly, when window filtering is applied to the data to reduce errors, the STFT is even more inefficient, as techniques such as overlap-and-add need to be used. Lastly, the compromise of time and

frequency resolution, that must be accepted due to the uncertainty principle, is the same over all frequencies, and cannot be adjusted for different frequency ranges.

2.2 Wavelet Transform Analysis

The wavelet transform deals with the time-frequency resolution limitations of fixed-size windowing that burden the STFT by using variable sized windows for different frequency bands[3]. The wavelet transform uses analysis windows (wavelets) that dilate according to the frequency being analyzed, with long time windows where more precise low frequency information is desired, and shorter windows where the high frequency information is desired.

Mathematically, the Fourier Transform represents the process of the Fourier analysis [4]:

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Therefore, the mathematical bases of the Fourier Transform are sine waves (the complex exponential can be broken down into real and imaginary sinusoidal components) of infinite duration. Similarly, the Continuous Wavelet Transform (CWT) is defined as:

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

where $*$ is the complex conjugate and ψ is the mother wavelet basis function, scaled by a factor a and dilated by a factor b . Just as the Fourier Transform breaks up a signal into sine waves of different frequencies, the wavelet transform breaks up a signal into scaled and shifted versions of the wavelet basis function.

If we define

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

the CWT can be viewed as the inner product of the original signal $f(t)$, with the basis ψ :

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

or, equivalently, the wavelet transform is the convolution of the function f with the filter whose impulse response is ψ

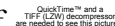

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

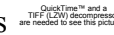
where

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

The filter frequency response of the dilated mother wavelet is:

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Taking the Fourier Transform on both sides of  above, the wavelet transform can be viewed as the filtering of the signal with a bandpass filter whose frequency response is .

The CWT as given above is *over-complete*. The coefficients  are redundant. The discrete version of the wavelet transform, the DWT, changes scale by powers of two, producing an octave band decomposition of a signal. Figure 1 shows side-by-side plots of the STFT and CWT with power of two spaced (dyadic) frequency scales of an audio signal. The figure illustrates how the wavelet transform's time and frequency resolution scale with frequency in contrast to the uniform time-frequency resolution of the STFT.

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Figure 1. STFT of an audio signal(left plot) and corresponding CWT with dyadic frequency scaling.

For the analysis carried out on audio signals in this paper, the CWT will be used, with scales that change in powers of two, similar to the DWT.

3.0 Wavelet Dispersion Measure for Classification

3.1 Background

In more practical terms, what the previous discussion tells us is that wavelet transform analysis overcomes the specific shortcomings of the STFT for time-frequency analysis of audio signal listed earlier. The information contained in the wavelet transform of an audio signal is concentrated in a way that closely matches the characteristics of human auditory perception. Motivated by this fact, we would like to obtain a compact representation of the information in the wavelet transform coefficients that is useful for instrument classification using neural networks. Such a wavelet based measure for audio content description was proposed by Stephan Rein and Martin Reisslein for identification of similar musical works[5],[6]. The measure they proposed, dubbed the *wavelet rank dispersion vector*, captures the patterns formed by the relationships between the magnitudes of the wavelet coefficients across the frequency scale dimension and across the time dimension. Such constitute just the types of signal features necessary for our classification goal.

3.2 Wavelet Rank Dispersion Vector (WRDV)

To construct the *WRDV*, we start with the coefficients of the CWT of a signal. The coefficients form an N column by M row matrix, where N is the length of the signal and M is the number of wavelet scales (figure 2). Next we assign ranks, from highest magnitude to lowest over each column, i.e. we rank the scale magnitudes at each sample time (figure 3). Once ranks are assigned, we discard the coefficients (figure 4) and accumulate the rank histogram data for each scale (row) to generate a M by M square matrix, where the number in the first column represents the number of times that each scale was assigned a rank of one, and likewise, column two reports the number of times each scale was ranked 2, and so on for all remaining columns. This is the *wavelet coefficient dispersion matrix* (figure 5).

$$\mathbf{C} = \begin{array}{ccccc|c} & 1 & 2 & 3 & 4 & 5 & \\ \hline & 0.43 & 0.22 & 0.14 & 0.76 & 0.33 & 1 \\ & 0.10 & 0.32 & 0.11 & 0.28 & 0.90 & 2 \\ & 0.54 & 0.49 & 0.34 & 0.18 & 0.91 & 3 \end{array}$$

Figure 2. CWT coefficients for a length 5 signal, over 3 wavelet scales

1	2	3	4	5	
0.43 (2)	0.22 (3)	0.14 (2)	0.76 (1)	0.33 (3)	1
0.10 (3)	0.32 (2)	0.11 (3)	0.28 (2)	0.90 (2)	2
0.54 (1)	0.49 (1)	0.34 (1)	0.18 (3)	0.91 (1)	3

Figure 3. CWT coefficients with magnitude rankings (in parentheses) over scales (columns)

1	2	3	4	5	
2	3	2	1	3	1
3	2	3	2	2	2
1	1	1	3	1	3

Figure 4. CWT coefficients' magnitude rankings only (coefficients discarded)

$$C_{\text{disp}} = \begin{array}{ccc|c} 1 & 2 & 3 & \\ \hline 1 & 2 & 2 & 1 \\ 0 & 3 & 2 & 2 \\ 4 & 0 & 1 & 3 \end{array}$$

Figure 5. Wavelet coefficient dispersion matrix

Each row of the *wavelet coefficient dispersion matrix* contains the rank histogram data for each frequency scale for the analyzed signal (figure 6). The set of histogram data of all the rows collectively represents the characteristic temporal and frequency scale patterns mentioned earlier for the analyzed audio signal. As a final step, we concatenate each of the rows into a single vector, which is the *wavelet rank dispersion vector (WRDV)*. The *WRDV* will be used as the input for the neural network that will perform the desired classification task. *We note that the CWT is used rather than the DWT (which is more common for processing digital signals) because, for a length N signal, the CWT produces N coefficients for each scale, which is required to construct the NxN wavelet coefficient dispersion matrix.*

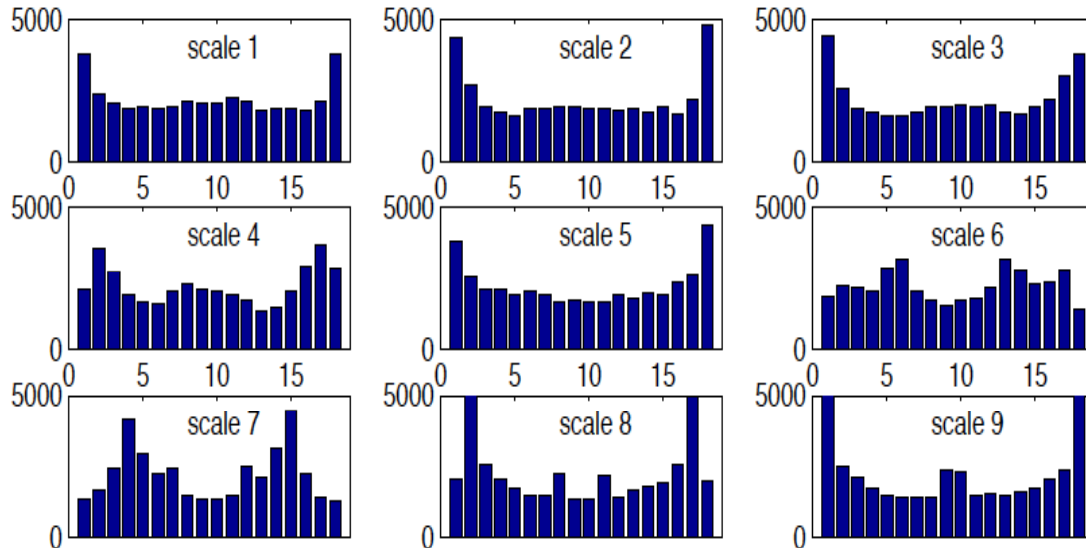


Figure 6. Rank histograms of each scale, contained in the rows of the *wavelet coefficient dispersion matrix*

3.3 Analysis of *WRDV* for Instrument Classification Performance

Thus far, we only have an intuitive rationale for using the *WRDV* as a signal content descriptor for classification. Here we present a more quantitative justification, along with more detailed arguments behind the intuition for using the *WRDV*.

As we noted earlier, the *WRDV* captures information about the relative magnitudes of the wavelet transform coefficients across scales. The *WRDV* also aggregates the wavelet coefficient relationships across the time dimension, so the intrinsic temporal characteristics of the signal are also represented[5]. Such temporal characteristics can help differentiate different musical instruments, for example, a piano note has a percussive, wide spectrum attack and exponentially decaying loudness, whereas notes played on a saxophone have more stationary spectral and loudness characteristics.

As a general quantitative indicator of the similarity of *WRDV*'s of audio signals with the same dominant instrument, and dissimilarity of *WRDV*'s with different dominant instruments, we can use correlation measures. The correlation plot in Figure 7 shows which vector pairs in our data set have maximum correlation coefficients with each other (excluding self correlation). If the *WRDV*'s for the same instruments correlate strongly with each other and weakly with different instruments, it is a good indicator that *WRDV*'s have the potential to function as a good

descriptor for classifying the dominant instrument. In the correlation plot, we see that *WRDV*'s of like instruments do, in fact, correlate well (73% have maximum correlation coefficient with like instruments).

4.0 Input Data

As our data set, we used audio excerpts from small ensemble 'modern jazz' recordings (e.g. late 1950's Miles Davis) since they typically have consistent characteristics (instrumentation, playing style) and follow a pattern of alternating individual solos in their performances, and thus provide a rich unified source of examples for our classification experiments.

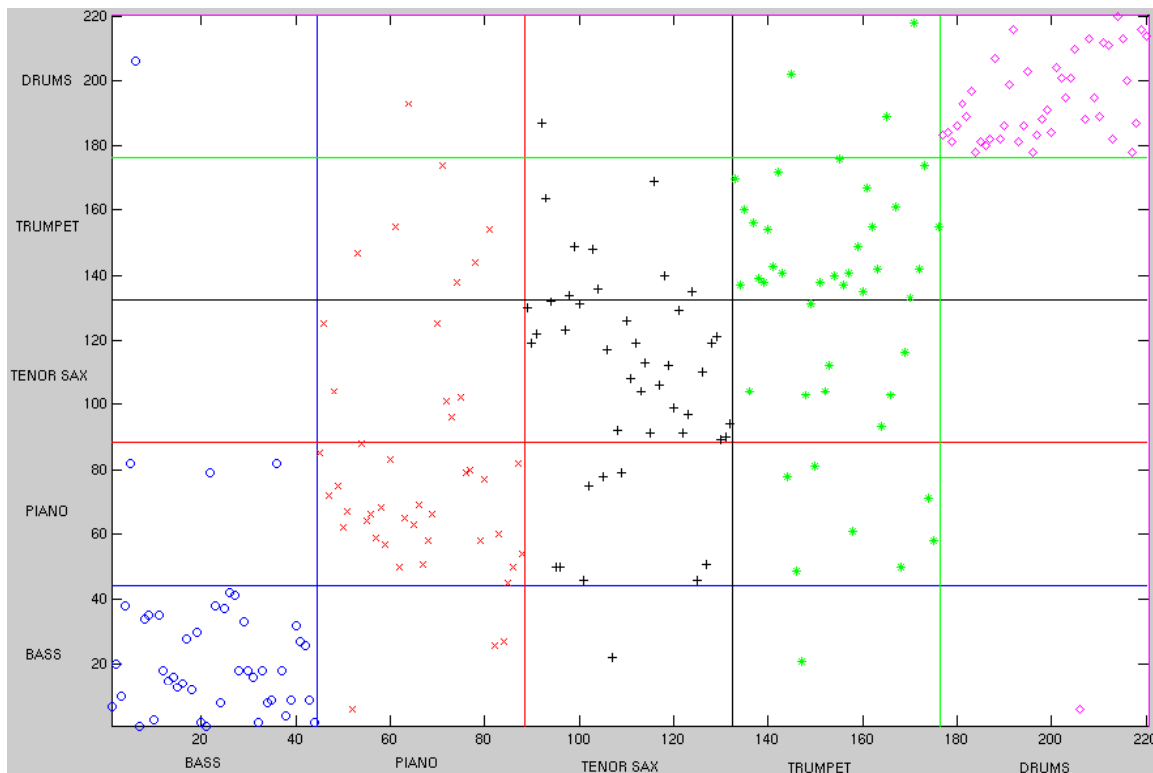


Figure 7. Maximum correlation coefficients of *WRDV*'s. 73% matching.

We collected solo's from eight different recordings by eight different groups for five different instruments: bass, piano, drums, tenor sax, and trumpet.

The audio excerpts were then divided up into two second blocks (22050Hz sampling rate) for analysis. The two second block length was empirically, somewhat arbitrarily, chosen to the smallest sized block that was big enough to insure with high confidence that the soloist is playing in the excerpt. We produced a large data set consisting of 44 audio blocks for each instrument, and a smaller data set consisting of 4 audio blocks per instrument.

We used an equal number of training pairs for each instrument. We note that in doing so we imposed equal prior probabilities for each of the instruments, which does not reflect the actual likelihood of different instruments' solos. For the purposes of our experiments, we chose to ignore these prior probabilities since the basic performance of the classification network will not be fundamentally dependent on them.

For the wavelet analysis, we used the CWT (from the Matlab Wavelet Toolbox) with 24 wavelet scales, which resulted in 24x24 wavelet coefficient dispersion matrices. We discarded the top two highest and lowest columns of the dispersion matrices, which represented the highest and lowest magnitude outliers, which reduced the dimensionality of the *WRDV*'s to 480 (24x20). We used the Meyer, the Morlet and the bior3.9 wavelet basis functions to do three separate analyses on the small data set for performance comparison of the different wavelet bases, then, based on the results, we chose the Meyer wavelet basis function to analyze the large data set for our final results.

5.0 Instrument Classification Neural Network

5.1 Neural Network Requirements

In order to choose an effective neural network, the nature of the task must first be considered. From a 480-dimensional wavelet dispersion input vector, a musical instrument must be clearly classified. At first glance, a single linear output seems to be ideal, and it certainly is the easiest output to implement in a neural network. However, early attempts to employ this single output resulted in unclear classification, with some outputs far removed from established classification criteria and others equidistant from two classification points. In order to increase the clarity of the network output, a 5-dimensional output was implemented with a binary classification method. Ideally, only one dimension will see a large response, and thus the output vector will be

dominated by the dimension corresponding to the correct instrument. In practice, the networks showed improved input/output correlation and clear classification when this strategy was implemented.

By choosing the component of the output vector with the highest magnitude, these networks can accurately classify the dominant musical instrument in the sample. Because the networks we trained so rarely misclassified instruments (2 validation samples out of 300 for the 20-training-vector networks, 0 out of 3300 for the 220-training-vector networks), for the purposes of this discussion, the error used to determine the relative effectiveness of each network is the mean squared error of each output deviation from the ideal of 0 magnitude in 4 dimensions and a response of 1 in the correct dimension (See figure 8).

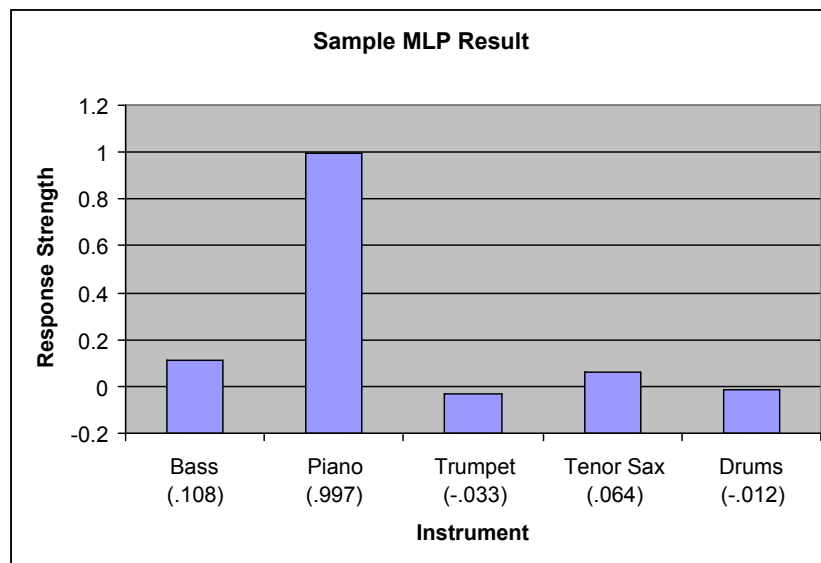


Figure 8. Sample output from a multi-layer perceptron network, typical mean square error

5.2 Network Selection

When evaluating an appropriate network choice for the classification of instruments from a wavelet dispersion vector input, we chose to examine two of the most common networks. We implemented similar evaluation techniques to measure the accuracy and complexity of both a multi-layer perceptron (MLP) network and a radial basis function (RBF) network.

Both networks share a common set of features. The input to each is a 480 dimensional wavelet dispersion vector, and the output of each is a 5 dimensional classification vector.

Each network has several parameters that determine the efficiency and the complexity of the network. Although the number of inputs and outputs are defined, the number of hidden units in the multi-layer perceptron network is variable, as well as the number of Gaussian centers in the radial basis function network. The spread on each Gaussian center in the radial basis function network is a variable parameter in our experiments. Similarly, the number of epochs in the multi-layer perceptron needs to be determined to most effectively train the network without overtraining.

5.3 Network Performance Measure

Because the input space is 480 dimensions, the number of known training samples necessary to reasonably populate the input space is quite large. Because the number of training input vectors is limited, we chose to implement the leave-one-out training method in order to maximize the number of known training points. In order to maintain the integrity of our validation error calculation, we used all but one known input vector for training and then tested the remaining input. To get a dependable error value for each network, we tested the validation error leaving out each sample one at a time and then averaged those error values. From the validation and training error data, we were able to compare different network types in order to choose the best set of network parameters.

5.4 Experimental Improvements

In order to decrease the validation error, we used two approaches. First, we evaluated three different input wavelet dispersion vectors generated by three different algorithms and compared their results through identical multi-layer perceptron networks. We also evaluated the beneficial effects of normalizing the input vector. The second component of our analysis was to vary two parameters for each network type and observe the relative performance of each in search of an ideal classification network.

5.5 Input data manipulation and normalization

Three different algorithms for generating wavelet dispersion vectors were tested in identical multi-layer perceptron networks to determine which algorithm was superior. We trained the networks using the leave-one-out method and 20 input vectors (4 for each instrument) generated by each algorithm. The multi-layer perceptron networks were utilized because the limited number of input vectors did not sufficiently populate the 480-dimension input space to implement a radial basis function network.

For each series of input vectors, we tested several different combinations of multi-layer perceptron network parameters. The networks had combinations of 1, 5, 15, 25, and 35 hidden units and 5-2000 epochs (See figure 1). In addition, we tested the error of each original dispersion vector before and after normalization to observe the effects of this preprocessing (See figure 10).

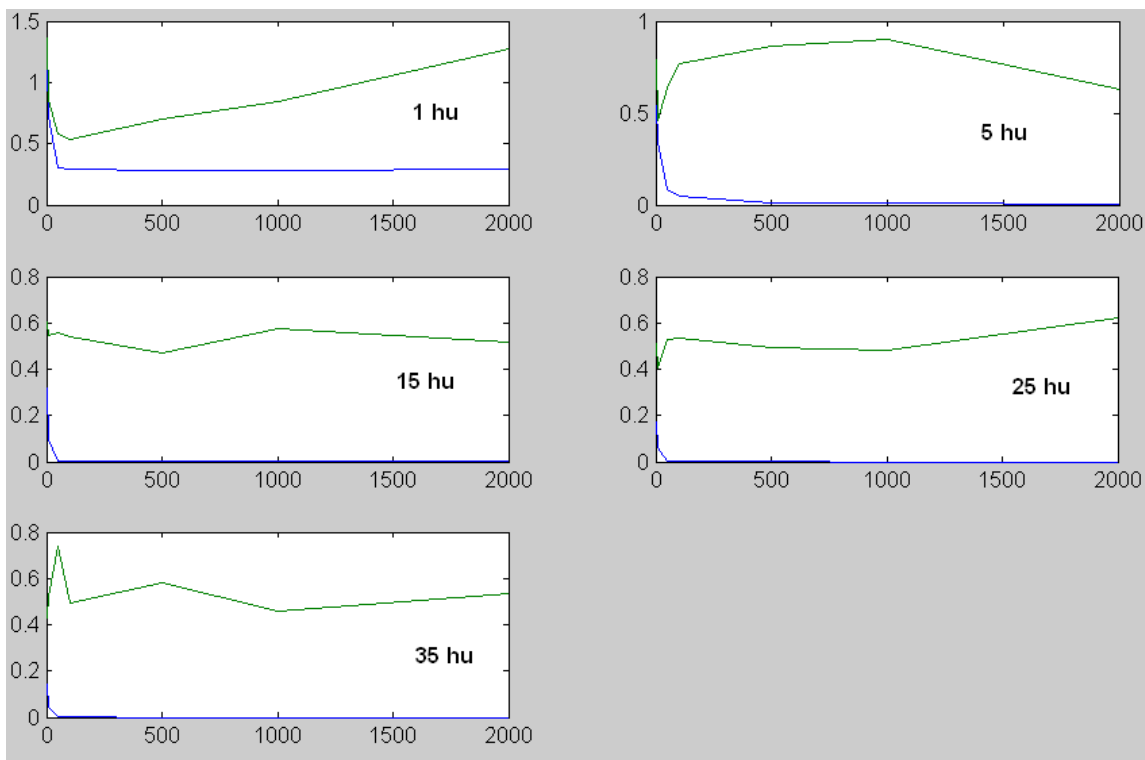


Figure 9.1 Meyer wavelet basis, mean squared error (blue = training, green = validation)

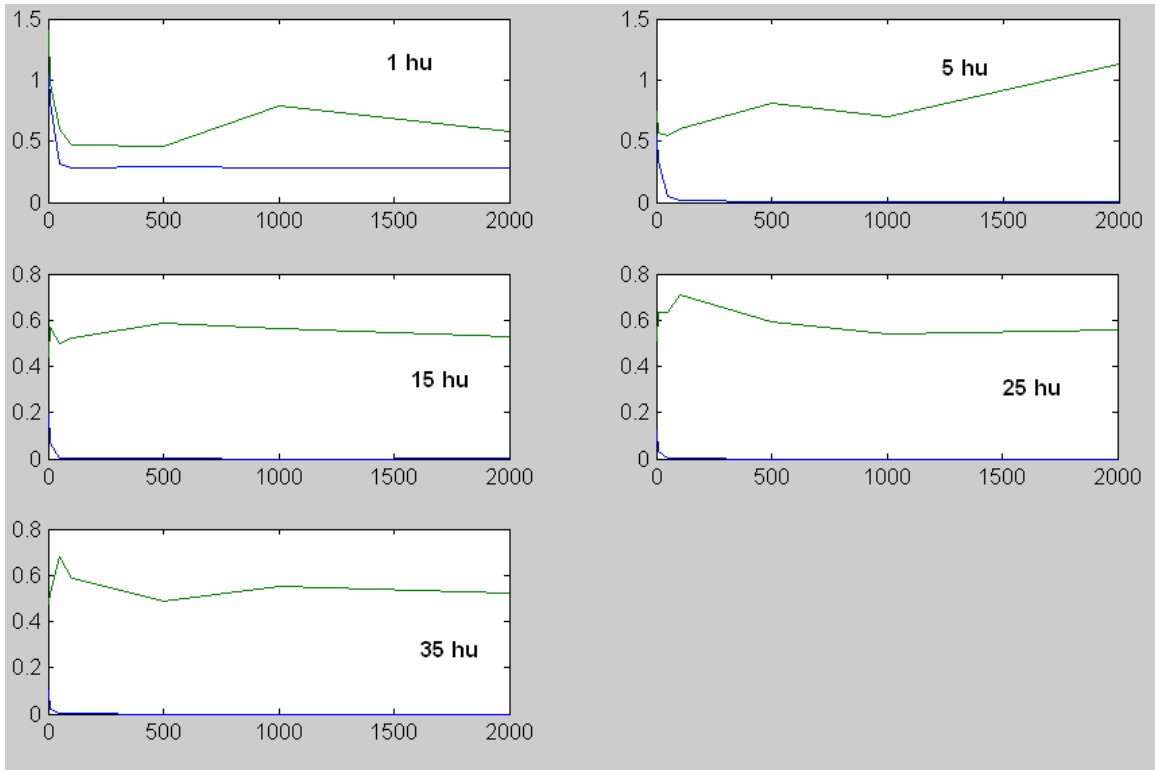


Figure 9.2 Bior3.9 wavelet basis, mean squared error (blue = training, green = validation)

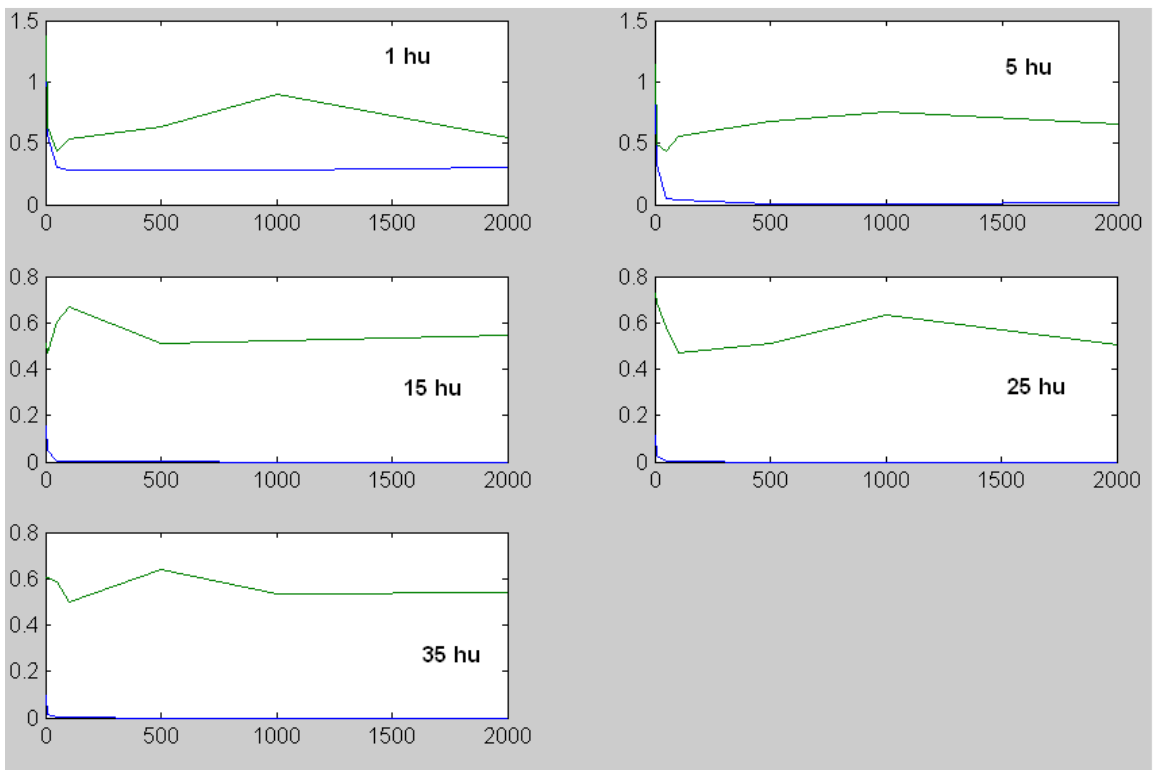


Figure 9.3 Morlet wavelet basis, mean squared error (blue = training, green = validation)

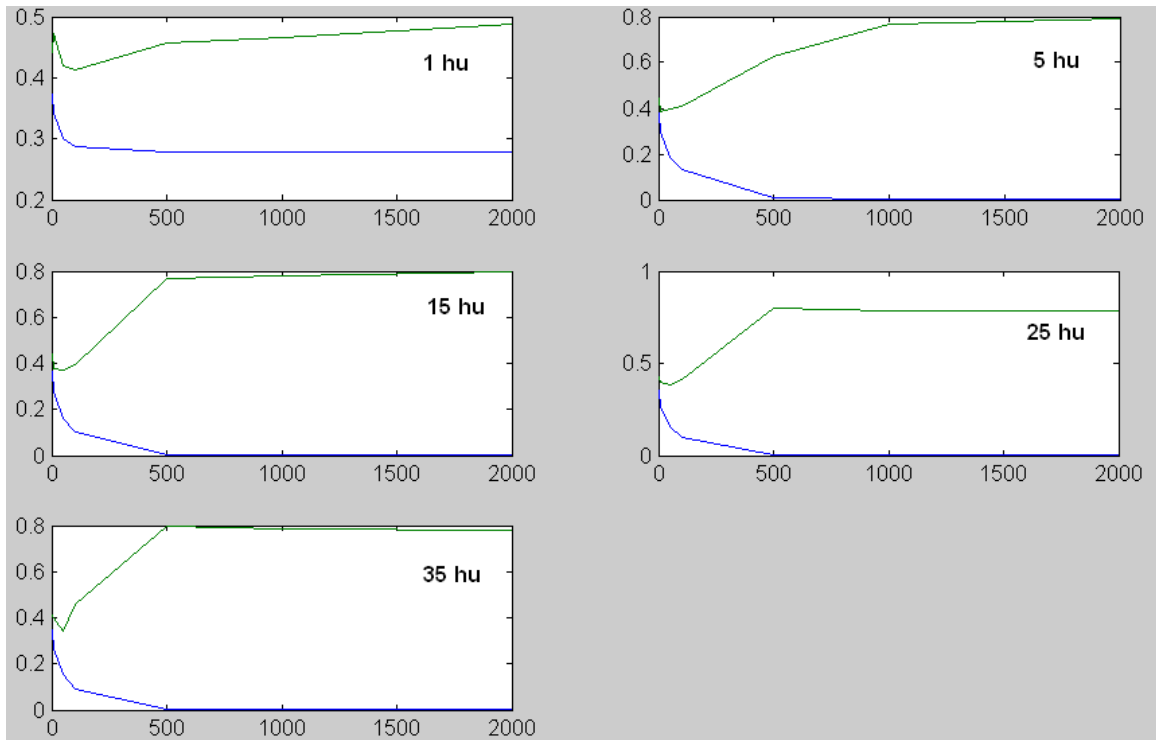


Figure 10.1 Normalized Meyer wavelet basis, mean squared error (blue = training, green = validation)

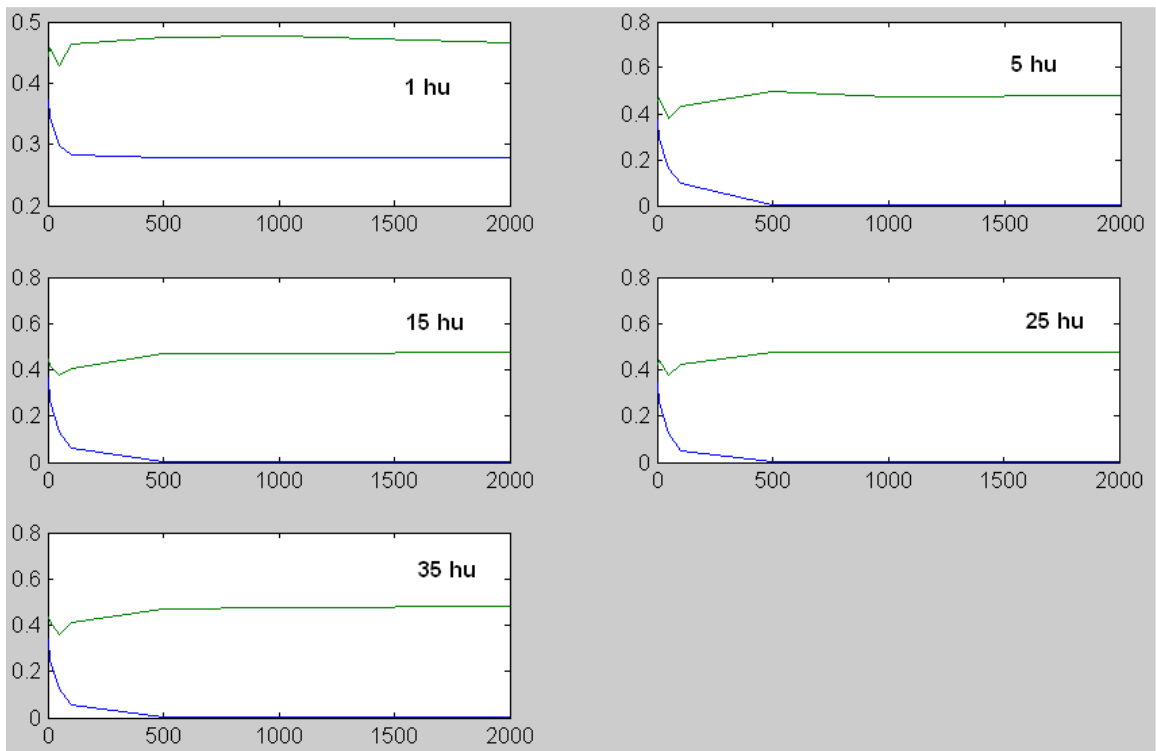


Figure 10.2 Normalized Bior3.9 wavelet basis, mean squared error (blue = training, green = validation)

6.0 Results

The multi-layer perceptron networks showed a considerably smaller error when the input vectors were normalized. This is consistent with our expectations because normalizing the input vector contains the training samples inside a relatively small portion of the input space. Based on this result, we normalized the input vectors for the final multi-layer perceptron networks as well as all of the radial basis function networks.

Of the different wavelet dispersion vector generating algorithms, the combination with the least error was the Meyer transform vector input to a multi-layer perceptron network with 5-15 hidden units and less than 200 training iterations.

6.1 Network parameters

In order to find network parameters to generate the lowest error, we used 220 training vectors generated by the Meyer transform (44 for each instrument), and trained multi-layer perceptron and radial basis function networks with various parameters. Based on the 20-training-vector data, we tested multi-layer perceptron networks with 4-12 hidden units and 200 epochs (See figure 11.1). We tested radial basis function networks with 2-70 centers and values of sigma (Gaussian spread on each center) between 0.05 and 10.0 (See figure 11.2).

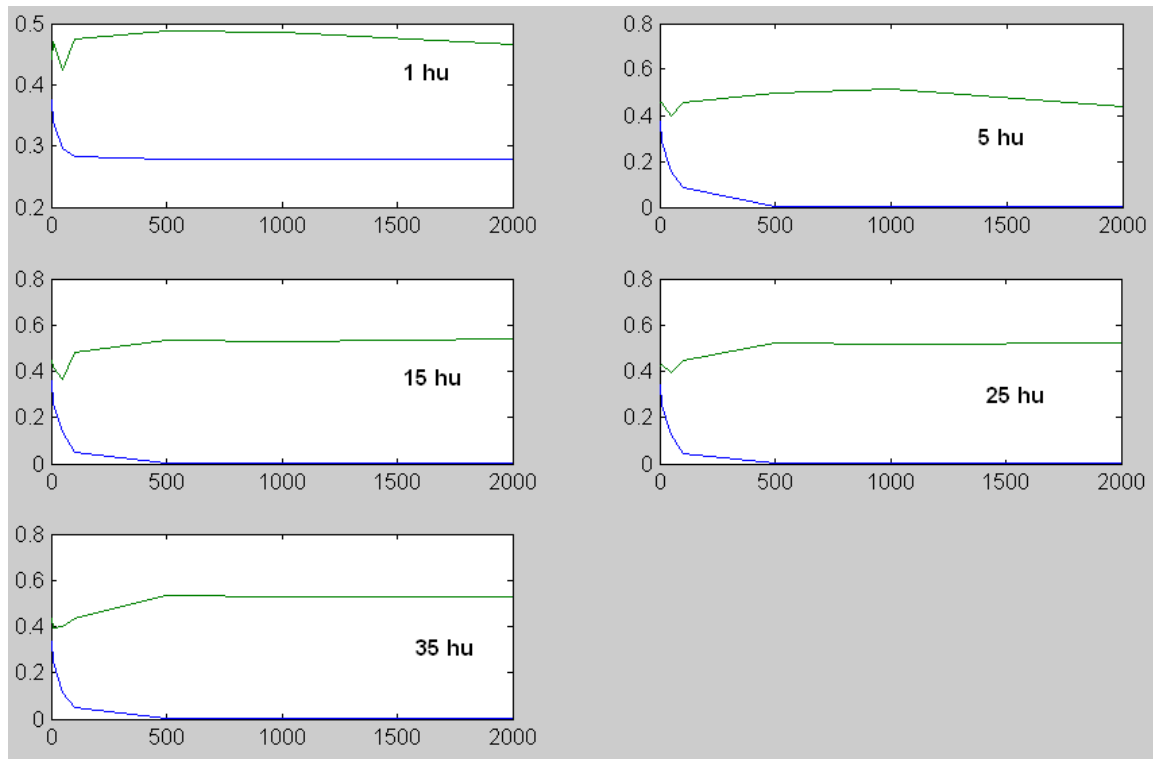


Figure 10.3 Morlet basis (normalized output), mean squared error (blue = training, green = validation)

6.3 Network performance comparison

The multi-layer perceptron network that performed the best had 5 hidden units and 1000 training epochs (See figures 11.3 and 12.2). The resulting validation error was less than 19%. The most effective radial basis function network had a 35 centers and a Gaussian spread on each center of 0.1 (See figures 11.2 and 12.1). The validation error of this network was less than 40%.

Therefore, of the networks we trained, the network with the least error was the multi-layer perceptron network (See figure 12.2).

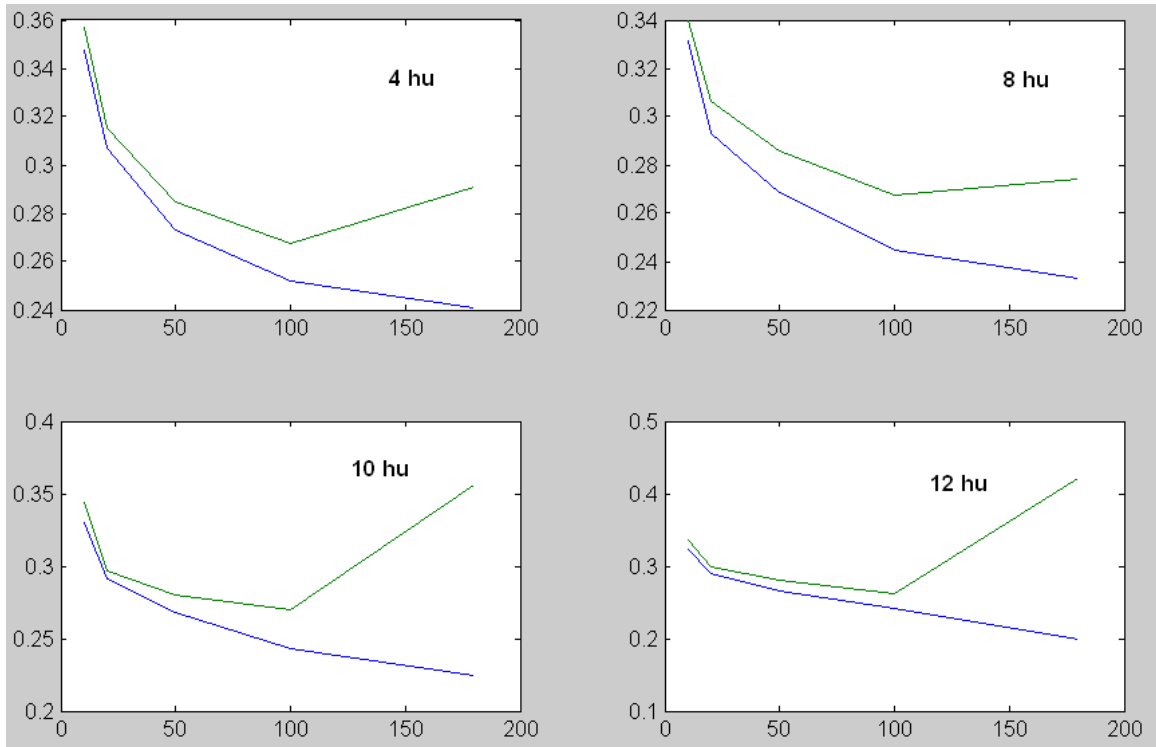
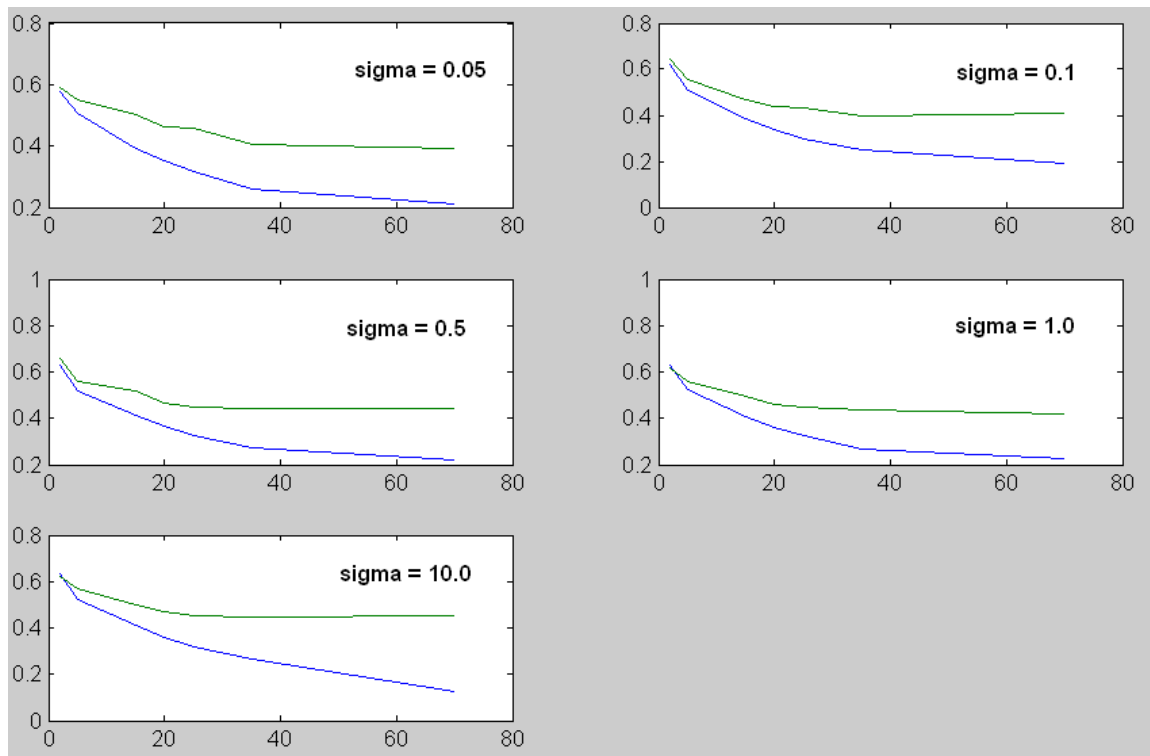


Figure 11.1 Meyer basis (normalized output) mean squared error (blue = training, green = validation)



11.2 Radial basis function networks with Meyer basis, normalized inputs, mean squared error in blue = training, green = validation)

Figure

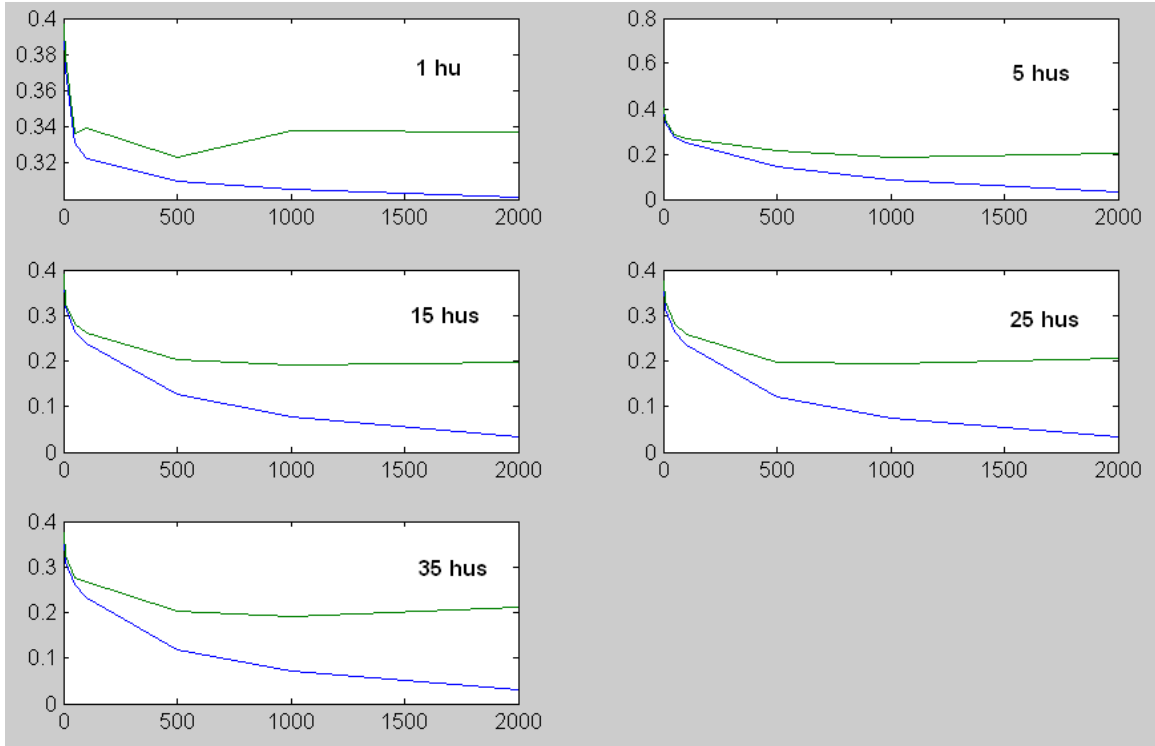


Figure 11.3 Multi-layer perceptron networks with Meyer basis, normalized inputs, mean squared error in blue = training, green = validation)

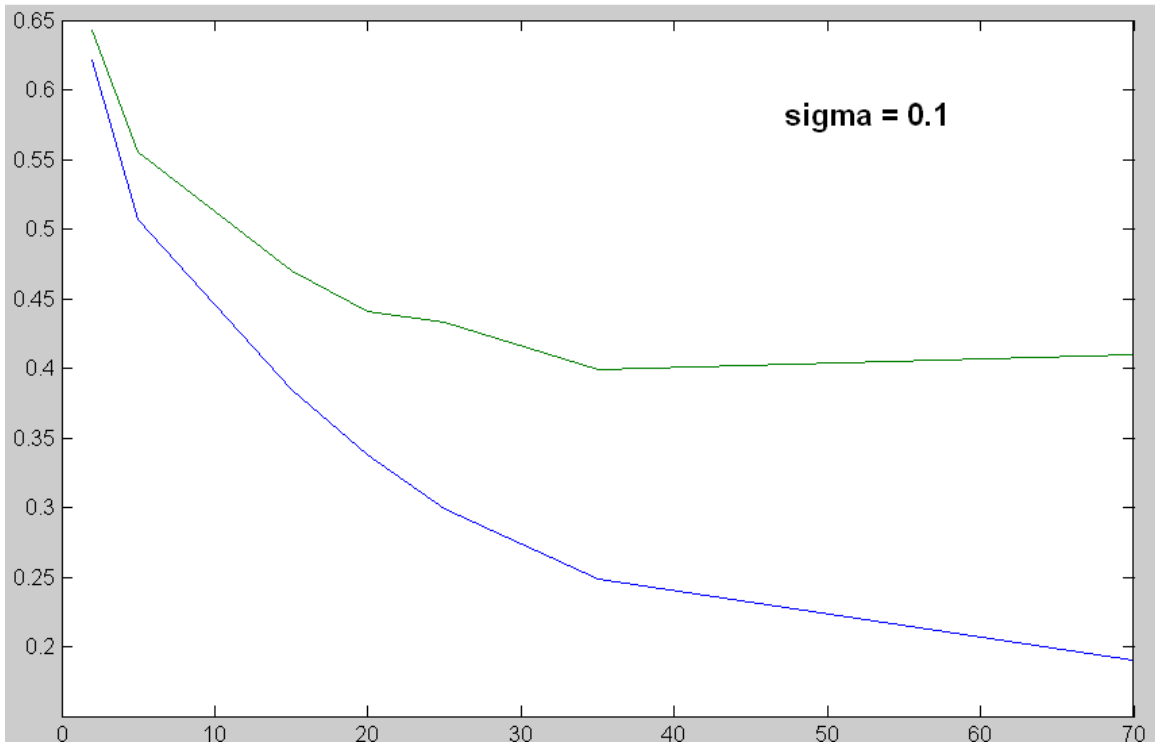


Figure 12.1 Final radial basis function network with Meyer basis, normalized inputs, mean squared error (blue = training, green = validation)

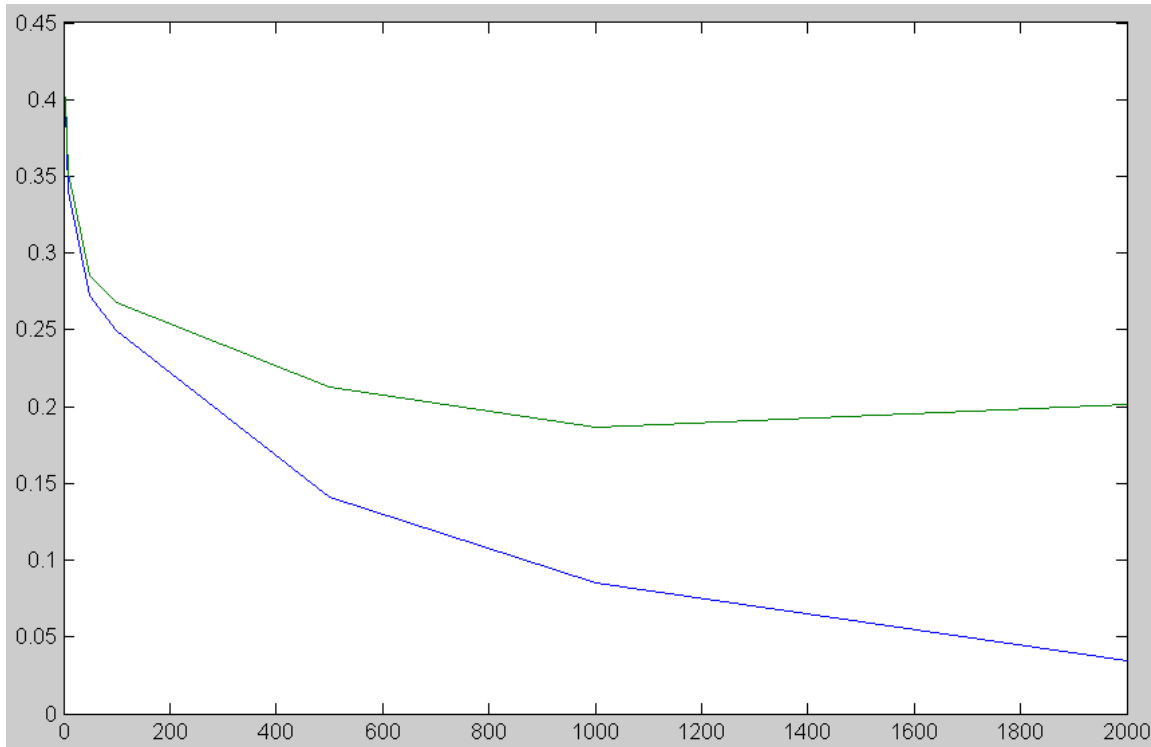


Figure 12.2 Final multi-layer perceptron network (10 hidden units and 100 iterations) with Meyer basis, normalized inputs, mean squared error

7.0 Conclusion

Our findings show that the scheme proposed for identifying the dominant musical instrument in a musical excerpt using the *wavelet rank dispersion vector (WRDV)* measure and a neural network for classification performed exceptionally well for a data set of small ensemble jazz recordings. The best performance was obtained by using the Meyer basis function for the wavelet analysis, and a multilayer perceptron network for classification. The final network exhibited a nearly perfect success rate in identifying the dominant instruments (after thresholding) for our data set examples. The classification method described in this paper has good potential for broad application to audio classification tasks beyond the one demonstrated in our experiments. The identification of similar classical musical pieces using a similar processing and neural network scheme has already been demonstrated by Rein and Reisslein [5]. Due to the ability to capture both frequency and temporal pattern characteristics of audio signals, the *WRDV* measure might be used, for example, to identify specific performers by their playing style, or composers by their compositional style.

8.0 References

- [1] Gerhard, David. *Audio Signal Classification: History and Current Techniques*. Department of Computer Science, University of Regina. Regina, Saskatchewan, Canada, 2003.
- [2] Bömers, Florian. *Wavelets In Real-Time Audio Signal Processing: Analysis and Simple Implementations*. Department of Computer Science, University of Mannheim. Mannheim, Germany, 2000.
- [3] Sanjaume, Jordi Bonada. *Audio Time-Scale Modification in the Context of Professional Audio Post-production*. Informàtica i Comunicació digital, Universitat Pompeu Fabra, Barcelona. Barcelona, Spain, 2002.
- [4] Beltrán, José R. and Beltrán, Fernando. *Additive Synthesis based on the Continuous Wavelet Transform: A Sinusoidal Plus Transient Model*. Dept of Electronic Engineering and Communications, University of Zaragoza, Spain. Zaragoza, Spain, 2003.
- [5] Rein, Stephen and Reisslein, Martin. *Identifying the Classical Music Composition of an Unkown Perormance with Wavelet Dispersion Vector and Neural Nets (Extended Version)*. Arizona State University, Tempe, AZ, 2004.
- [6] Rein, Stephen and Reisslein, Martin. *Proceedings of the IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages iv-341-iv-344, Montreal, Canada, 2004.