Remembering Larry

(aka: A numerical comparison of seismic inversion, multilayer and basis function neural networks)

Brian Russell

Banff, Alberta, December 10, 2019

Introduction

- I first met Larry Lines in 1976 when we were both new hires (Larry with Amoco and me with Chevron) and were introduced by a mutual friend.
- When Larry got transferred to the Amoco research lab in Tulsa, we lost touch with each other for a while, but reconnected in the mid-80s through our work with SEG, especially on the Board of TLE.
- In the late 90s, Larry and I served on two SEG executive committees (Larry as Editor and I as President-elect and President).
- I was so impressed with Larry's tenacity and research and teaching abilities (and all around niceness!) that when I decided to go back to U of C to pursue my doctorate I talked him into being my supervisor.
- Tonight I would like to remember Larry through the research we did together during those years at CREWES, and show what an influence he has had on my work ever since.

First CREWES Talk

- Our first CREWES talk, in 2001 (jointly with Dan Hampson and Todor Todorov), combined multi-attribute transforms and geostatistics and lead to a publication in the Journal of Petroleum Geology.

2001 CREWES Meeting:

Combining geostatistics and multiattribute transforms – A channel sand case study

Brian H. Russell, Daniel P. Hampson[1], Laurence R. Lines, and Todor Todorov[1]

Journal of Petroleum Geology, January, 2002:

COMBINING GEOSTATISTICS AND MULTI-ATTRIBUTE TRANSFORMS: A CHANNEL SAND CASE STUDY, *BLACKFOOT* OILFIELD (ALBERTA)

B. Russell*, D. Hampson*, T. Todorov* and L. Lines**

Porosity map from multiple attributes and gestatistics

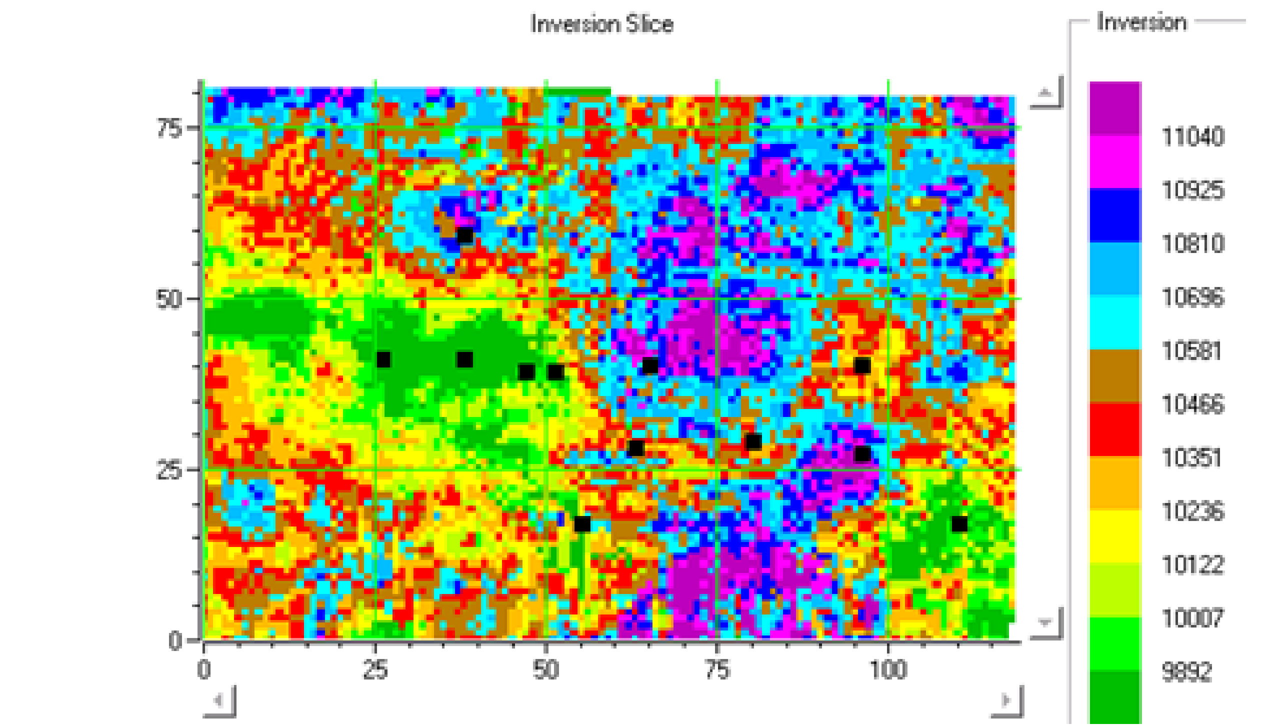


Figure 5: Map of average acoustic impedance over a 10 ms window below the picked channel top.

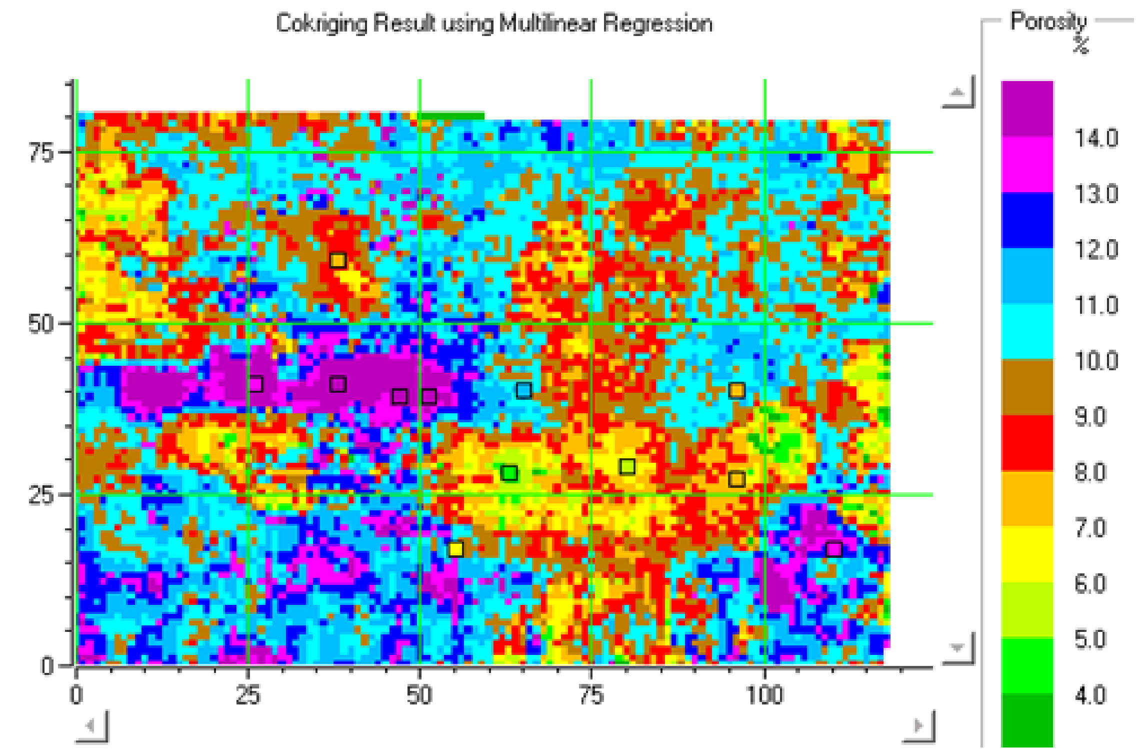


Figure 16: The result of applying cokriging to using the multi-linear regression result.

These figures are from the CREWES paper, where the one on the left shows an impedance slice though a channel sand and the one on the right shows the porosity prediction using a combination of multiple seismic attributes and geostatistics.

Second CREWES Talk

- Our second CREWES talk, in 2002 (jointly with Chris Ross), was a tutorial that explained two types of neural networks using an AVO classification problem, and lead to both an SEG expanded abstract and a TLE paper.

2002 CREWES Meeting:

AVO classification using neural networks: A comparison of two methods

Brian H. Russell, Laurence R. Lines, and Christopher P. Ross[1]

SEG Expanded Abstracts, 2002:

Neural Networks and AVO

Brian Russell, and Christopher Ross, Hampson-Russell Software Services Ltd., Larry Lines, Department of Geology and Geophysics, University of Calgary.*

The Leading Edge, 2002:

TUTORIAL

TUTORIAL

Neural networks and AVO

BRIAN RUSSELL, Hampson-Russell Software Services, Calgary, Alberta, Canada
CHRISTOPHER ROSS, Hampson-Russell Software Services (U.S.), Houston, Texas, U.S.
LARRY LINES, University of Calgary

The AVO classification problem

Here are two figures from that paper:

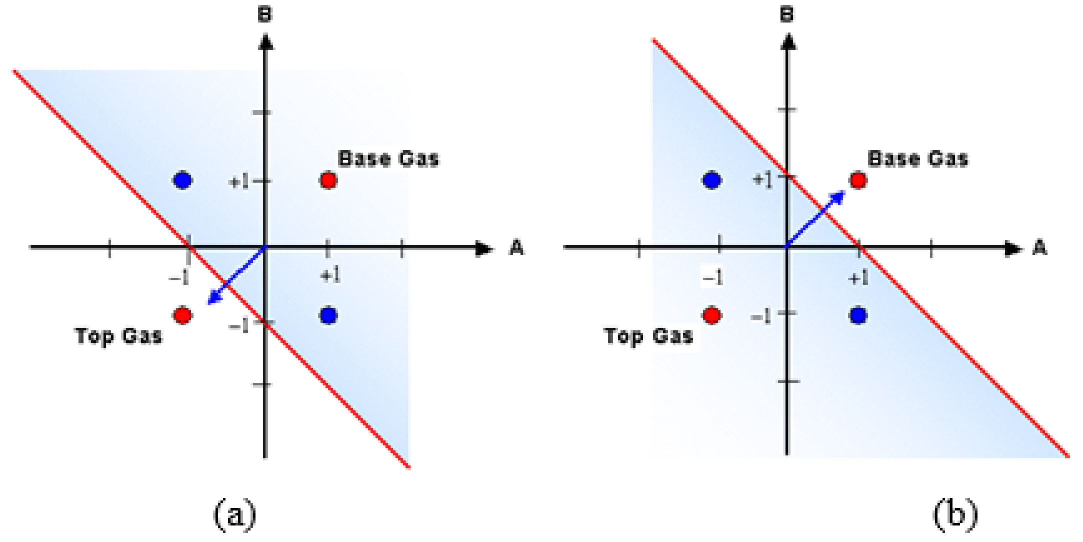


Figure 8: The AVO problem from Figure 2 with decision boundaries, where (a) shows separation of the base of the gas sand and (b) shows separation of the top of the gas sand.

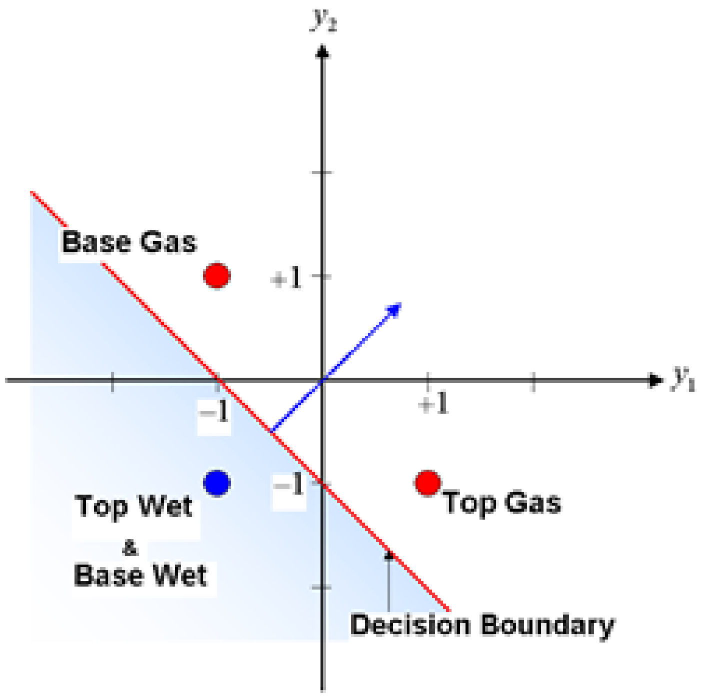


Figure 12: A crossplot of the outputs of the two perceptrons of Figure 9.

- The figure on the left shows that a linear boundary can solve for either the top or base of a gas sand on an AVO cross-plot, but not both.
- The figure on the right shows that by using a second layer of neurons, we can transform the problem in a linearly separable one.

The AVO classification problem

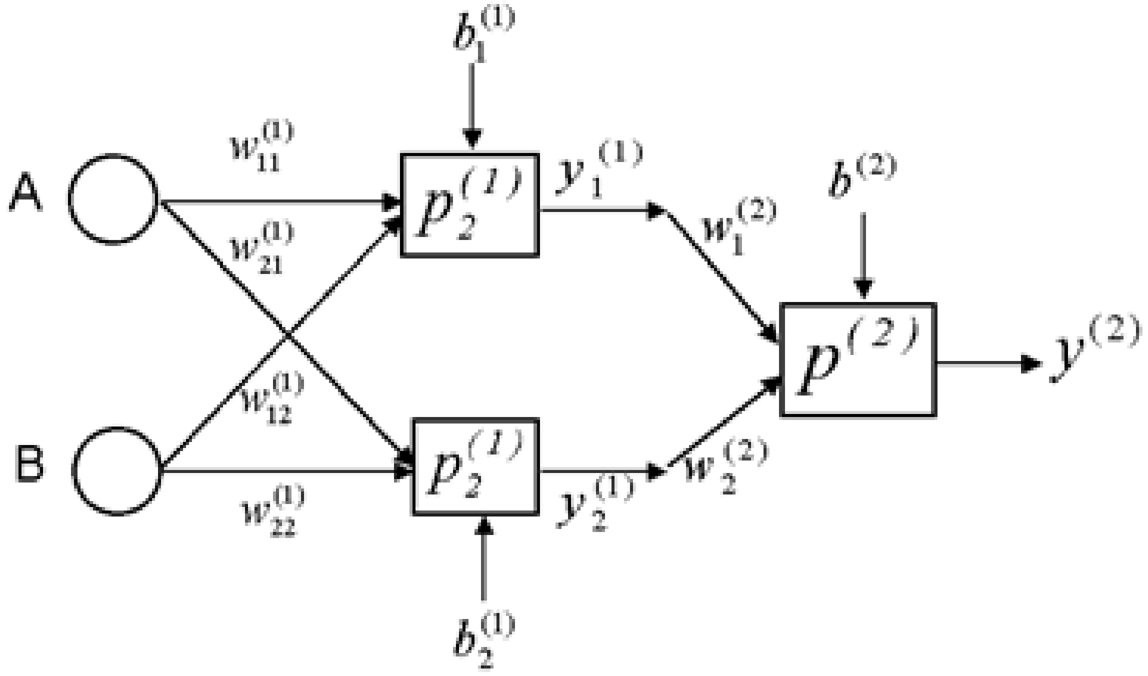


Figure 11: The multi-layer perceptron for the gas-water sand model of Figure 3.

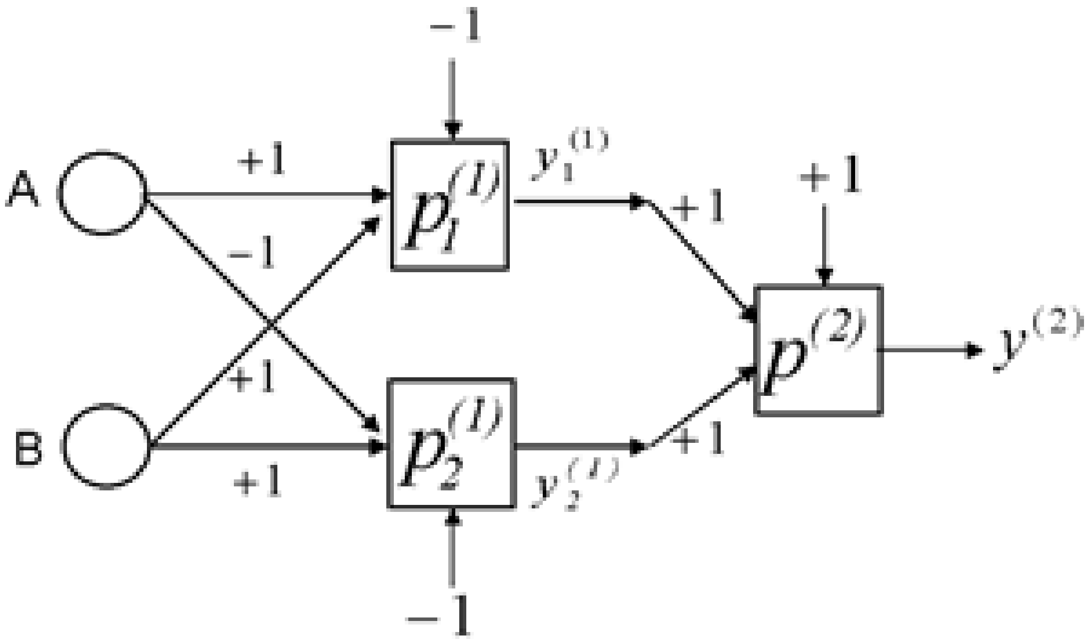
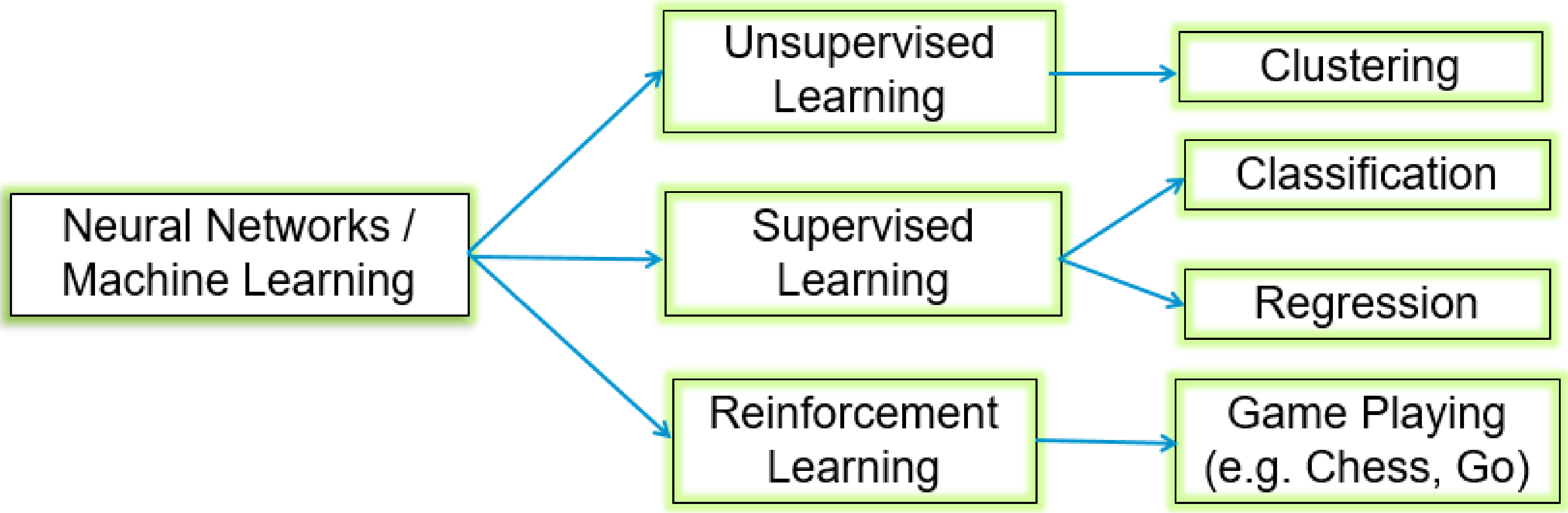


Figure 13: The final multi-layer perceptron weights.

- The figure on the left shows the generalized neural network used for solving this problem.
- The figure on the right shows the intuitive weights that were used in solving the problem, which are all + or – 1.

Types of neural networks

- But, as is shown in the following figure, neural networks and machine learning are about more than just classification, and involve clustering and nonlinear regression:



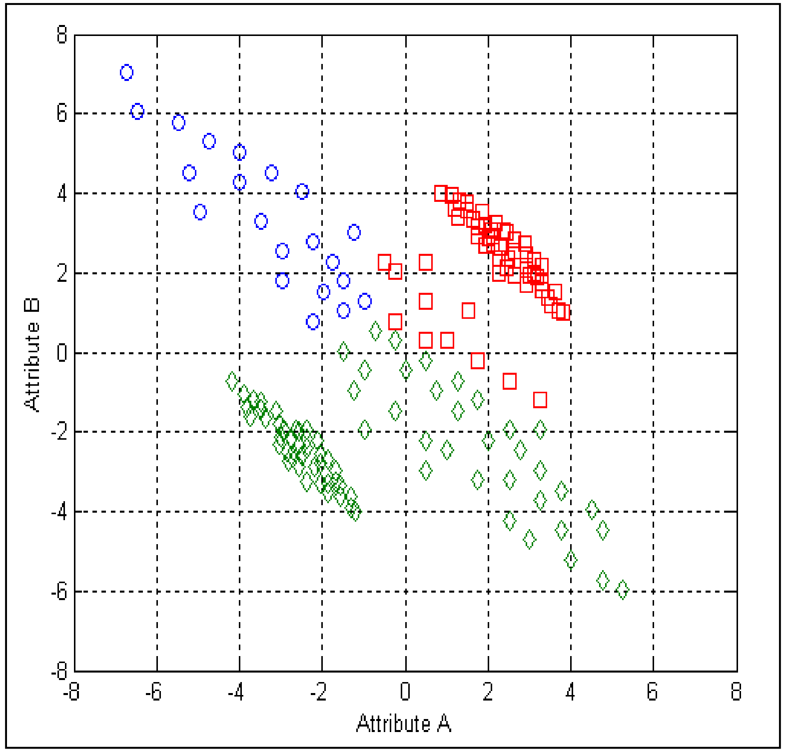
- At the 2003 meeting, Larry and I proposed a new type of clustering that extended the K-means algorithm using statistical distance (unfortunately, although this seemed like a new idea to us, it had already been published!)

CREWES Talks

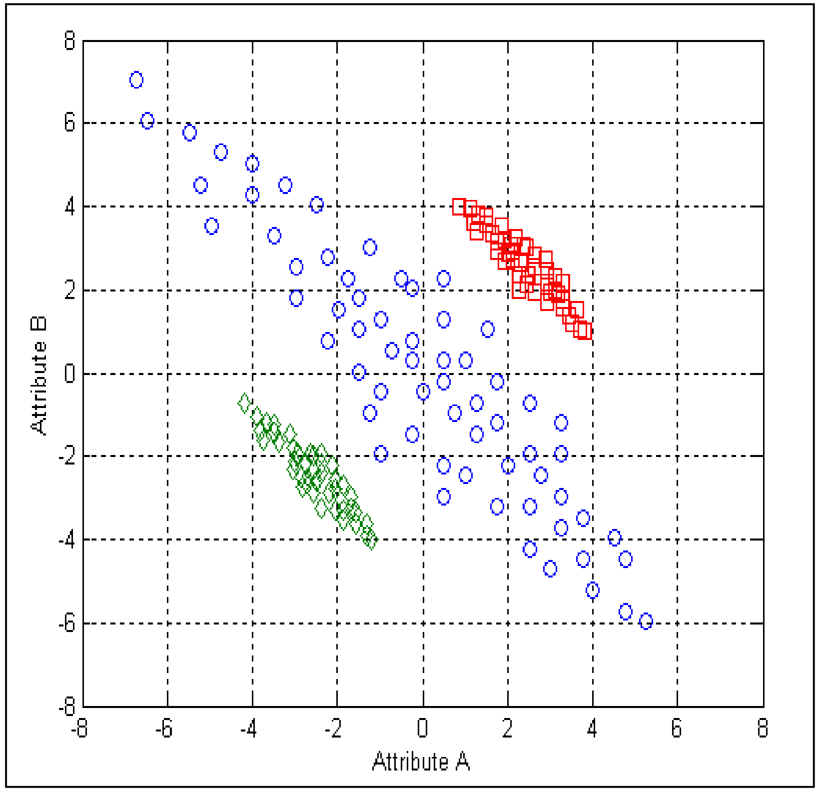
2003 CREWES Meeting:

Mahalanobis clustering, with applications to AVO classification and seismic reservoir parameter estimation

Brian H. Russell and Laurence R. Lines



The figure on the left, from this talk, shows how traditional K-means will miss-identify the wet trends and anomalies in a synthetic AVO crossplot, whereas the figure on the right shows the correctly classified clusters using Mahalanobis K-means.



Nonlinear regression neural networks

- We were also working on neural networks that performed nonlinear regression with seismic attributes, which included:
  - The Multi-layer Feedforward Network (MLFN)
  - The Generalized Regression Network (GRNN)
  - The Radial Basis Function Network (RBFN)
- Our CREWES presentation at the 2002 meeting (and at the 2003 SEG meeting) compared the RBFN network to the other two networks:

2002 CREWES Meeting:

Application of the radial basis function neural network to the prediction of log properties from seismic attributes

Brian H. Russell, Laurence R. Lines, and Daniel P. Hampson[1]

2003 SEG Expanded Abstract:

Application of the radial basis function neural network to the prediction of log properties from seismic attributes – A channel sand case study

Brian H. Russell, and Daniel P. Hampson, Hampson-Russell Software Services Ltd., Laurence R. Lines, Department of Geology and Geophysics, University of Calgary.*

Neural network comparison

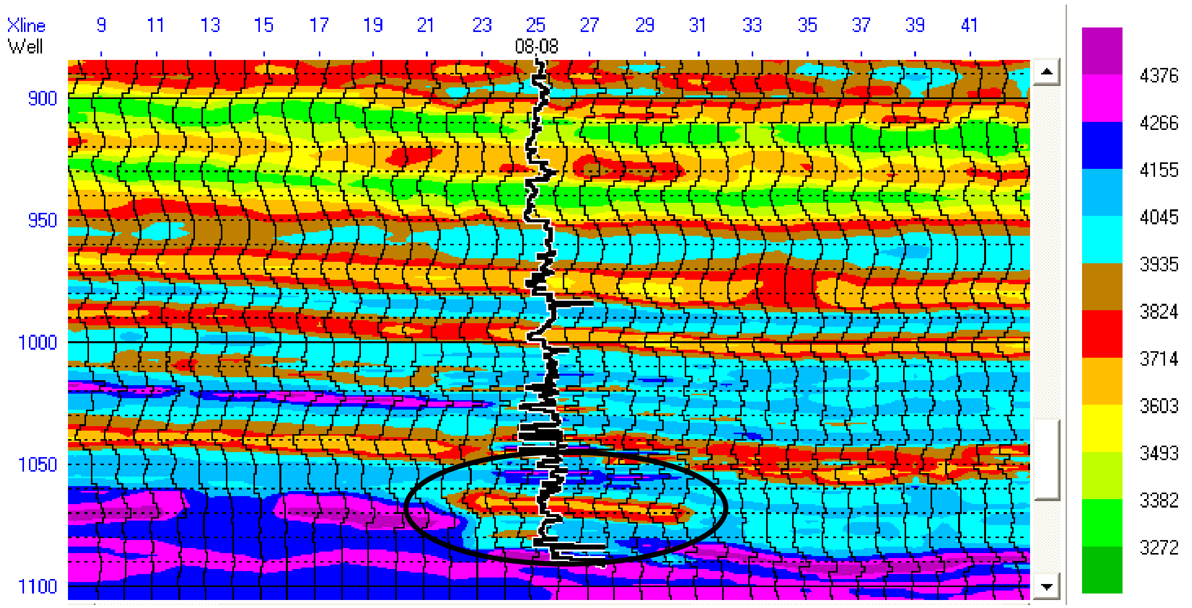


Figure 10: Application of the GRNN algorithm to line 95 of the 3D volume, after training using all the wells.

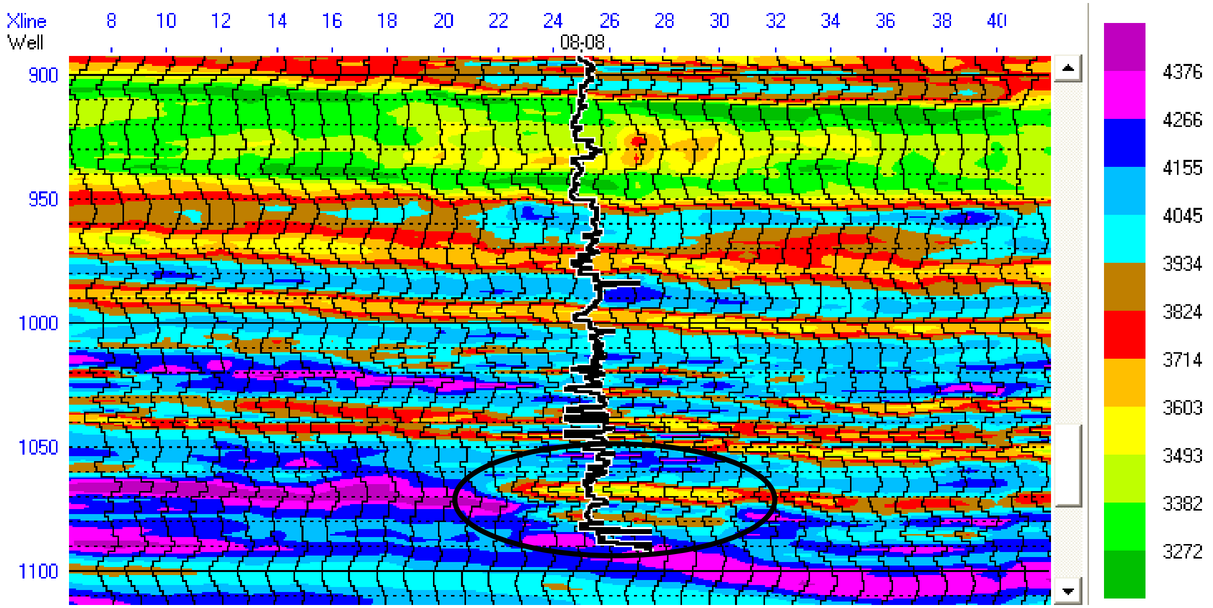


Figure 11: Application of the RBFN algorithm to line 95 of the 3D volume, after training with all the wells.

- Here are two figures from that paper that give a comparison between the Radial Basis Function and Generalized Regression Neural Networks.
- Notice the definition of the low-velocity channel sand in the deeper part of the section.

Other Talks and Papers

- Here are several other papers that Larry and I published during my time as his Ph.D. student (but I won't have time to discuss them tonight):

Geophysics, 2003:

Tutorial

Fluid-property discrimination with AVO: A Biot-Gassmann perspective

Brian H. Russell*, Ken Hedlin‡, Fred J. Hilterman**, and Lawrence R. Lines§

2004 CREWES Meeting:

A case study in the local estimation of shear-wave logs

Brian H. Russell, Laurence R. Lines and Daniel P. Hampson

2005 SEG Extended Abstract:

A case study in the local estimation of shear-wave logs

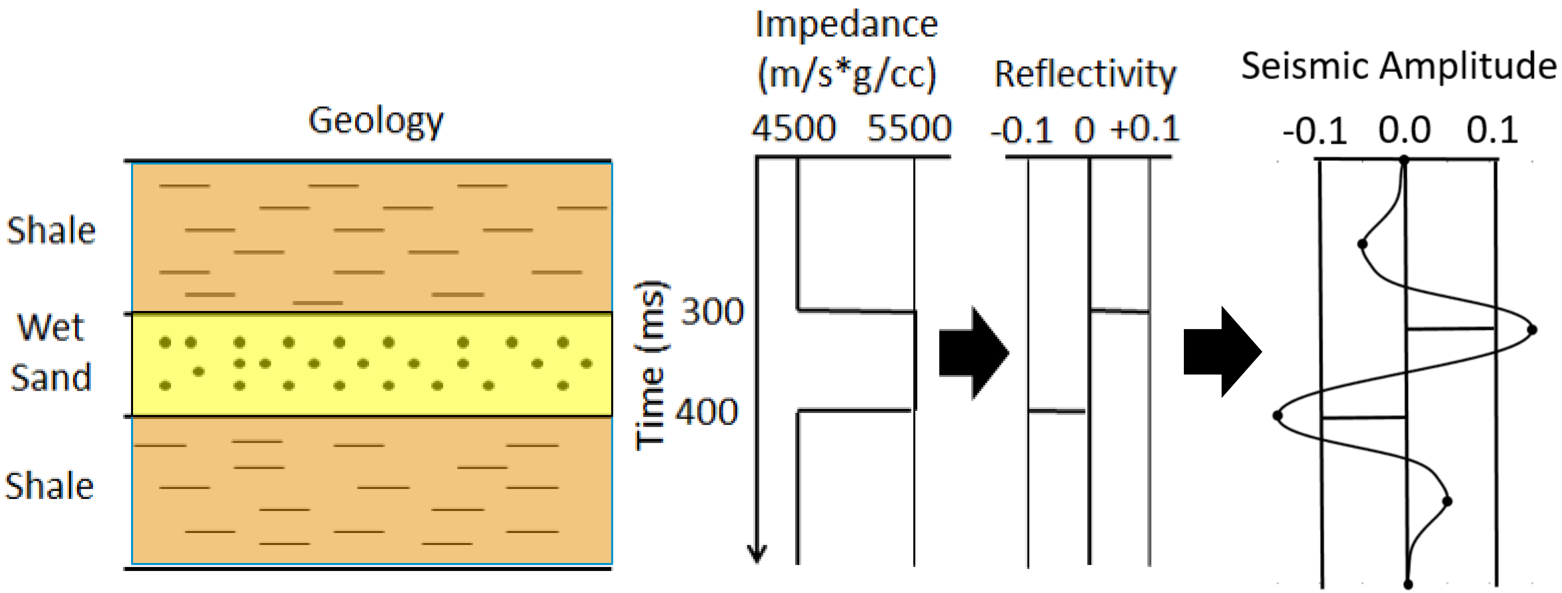
Brian H. Russell, and Daniel P. Hampson, Hampson-Russell Software Services Ltd., Laurence R. Lines, Department of Geology and Geophysics, University of Calgary.*

Zoom to the present

- Thanks to Larry, I graduated from the U of C with my Ph.D. in geophysics in 2004, and my thesis reflected much of what I have just touched on.
- But in the fifteen years since, Larry and I have never stopped talking and collaborating, through my continued involvement in CREWES, the SEG (Larry was President from 2008-2009), and the fact we live about fifteen minutes apart in Varsity.
- In fact, our best technical discussions have happened while walking with one of Larry's three Malamutes (most recently, Pearl, but I think this is Denali) at Varsity Ravine Park.
- So the last part of this talk was inspired by man's best friend!

A seismic inversion example

- In the July, 2019 issue of The Leading Edge, I published an article entitled "Machine learning and geophysical inversion – A numerical study".
- The article was based on discussions with Larry.
- Here is the forward model that I used.
- The mathematics is shown below:



$$\boldsymbol{s} = G\boldsymbol{r} = \begin{bmatrix} -0.5 & 0 \\ 1 & -0.5 \\ -0.5 & 1 \\ 0 & -0.5 \end{bmatrix} \begin{bmatrix} +0.1 \\ -0.1 \end{bmatrix} = \begin{bmatrix} -0.05 \\ +0.15 \\ -0.15 \\ +0.05 \end{bmatrix}$$

- As shown below, this problem has an exact geophysical inverse:

$$\hat{\boldsymbol{r}} = \boldsymbol{r} = \left[G^T G\right]^{-1} G^T \boldsymbol{s} = \begin{bmatrix} +0.1 \\ -0.1 \end{bmatrix}$$

The machine learning approach

- I then looked at the machine learning approach, which I implemented as a supervised neural network, where we know both the input and output:

Input = seismic trace ***s*** → Machine Learning Algorithm → Desired output = reflectivity ***r***

- That is, the machine learning algorithm learns the weights that will transform the seismic trace into the reflectivity.
- This is actually a type of nonlinear regression, so I first discussed the linear regression approach, given by the following mathematics:

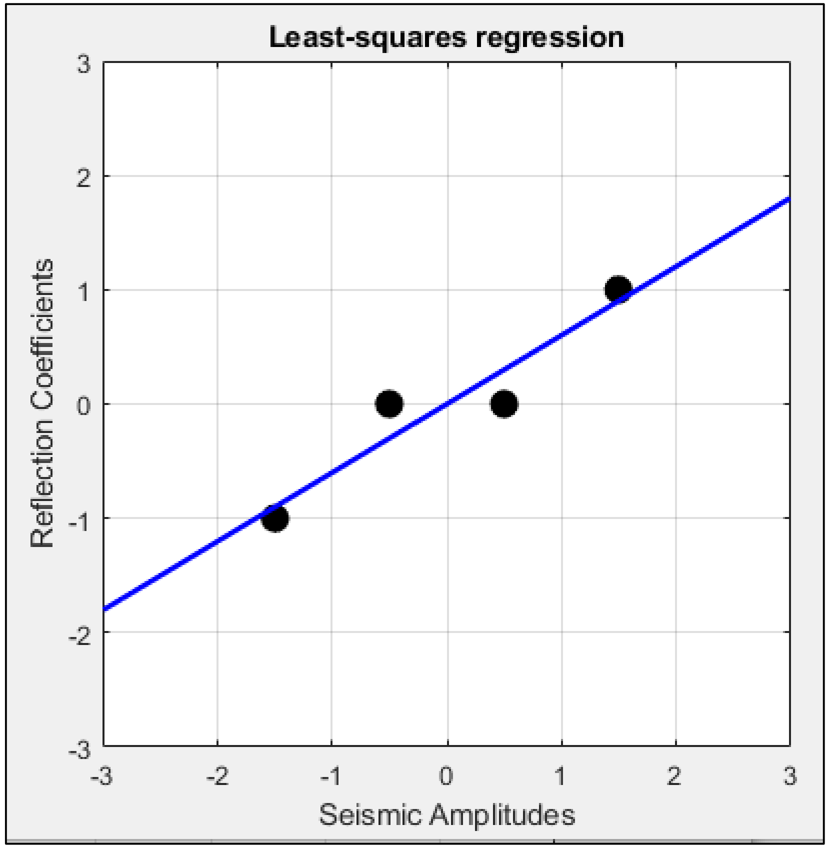
$$\hat{\boldsymbol{r}} = w_0 + w_1\boldsymbol{s} \quad \Rightarrow \boldsymbol{r} = S\boldsymbol{w} = \begin{bmatrix} 1 & -0.05 \\ 1 & +0.15 \\ 1 & -0.15 \\ 1 & +0.05 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 0 \\ +0.1 \\ -0.1 \\ 0 \end{bmatrix} \Rightarrow \boldsymbol{w} = (S^T S)^{-1} S^T \boldsymbol{s} = \begin{bmatrix} 0 \\ 0.6 \end{bmatrix} \Rightarrow \hat{\boldsymbol{r}} = \begin{bmatrix} -0.03 \\ +0.9 \\ -0.9 \\ +0.03 \end{bmatrix}$$

Linear regression

- A way to visualize the weights is to show the straight line fit to the reflection coefficients versus seismic amplitudes, as shown here.
- The true values are shown by the black points and the line represents the equation:

$$\hat{\boldsymbol{r}} = 0 + 0.6\boldsymbol{s}$$

- In deconvolution we got a perfect fit because our model assumptions were correct.

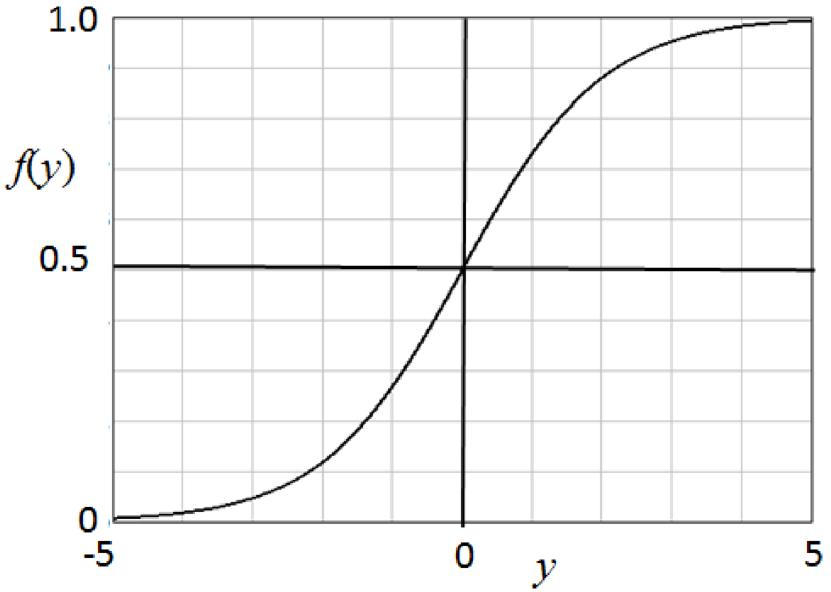


- In least-squares regression, the points are fit in a “best” least-squares sense.
- Let us now move to the neural network approach.

The feedforward neural network

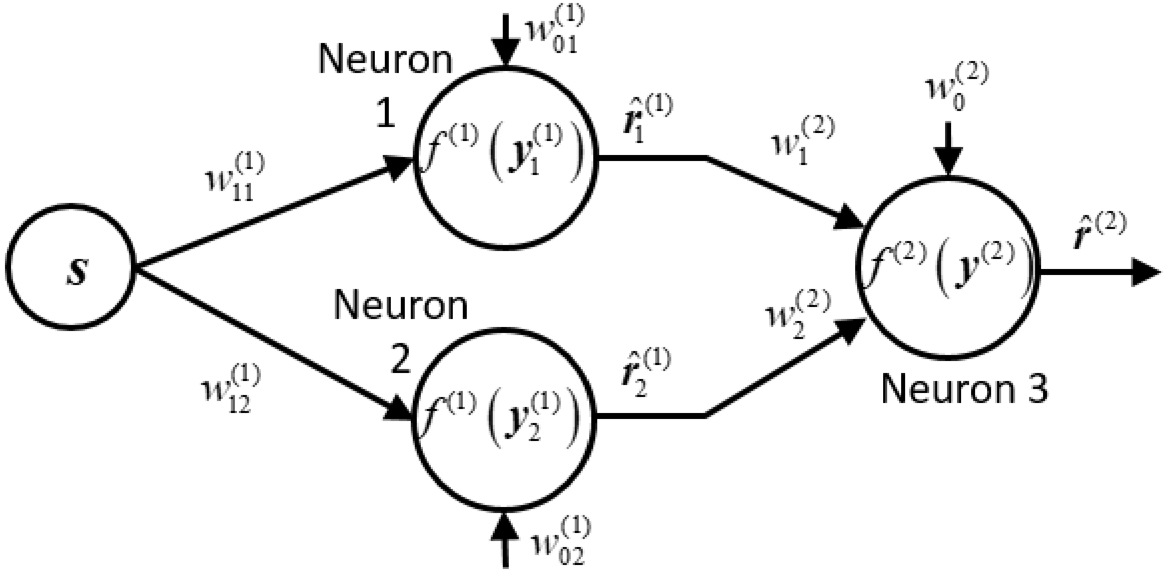
- We saw that the straight-line solution given by linear regression did not give a perfect fit between the true seismic and reflectivity values.
- Neural networks allow us to extend linear regression to nonlinear regression.
- In the feedforward neural network the term feedforward refers to how the output is computed from the input if the weights have already been determined,where the training of the weights is performed using what is called error backpropagation.
- The key innovation in the feedforward neural network is the used of multiple neurons with nonlinear functions, which most commonly are logistic functions, as shown on the right, with the formula:

$$f(y) = \frac{1}{1 + \exp(-y)}$$



The feedforward neural network

- In the first part of the process we apply two sets of bias and gradient weights to the seismic.
- This can be written in vector format as follows, where superscript (1) is the first layer:



$$\boldsymbol{y}_1^{(1)} = w_{01}^{(1)} + w_{11}^{(1)}\boldsymbol{s},$$
$$\boldsymbol{y}_2^{(1)} = w_{02}^{(1)} + w_{12}^{(1)}\boldsymbol{s},$$

where:

$$\boldsymbol{s}^T = \begin{bmatrix} -.05 & .15 & -.15 & .05 \end{bmatrix}$$

- We then apply the logistic function in each neuron:

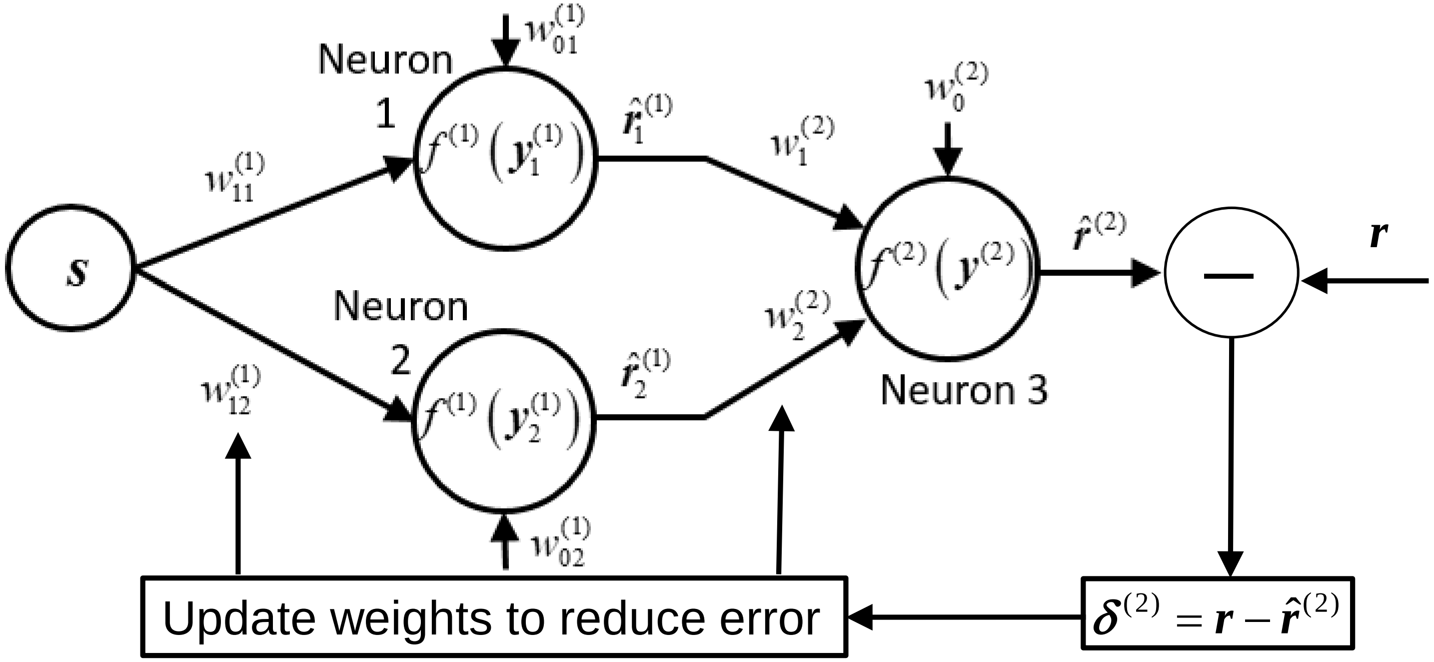
$$\hat{\boldsymbol{r}}_1^{(1)} = \frac{1}{1+\exp\left(-\boldsymbol{y}_1^{(1)}\right)},$$
$$\hat{\boldsymbol{r}}_2^{(1)} = \frac{1}{1+\exp\left(-\boldsymbol{y}_2^{(1)}\right)}.$$

- Finally, we apply linear weights in the last neuron where superscript (2) is the second layer:

$$\hat{\boldsymbol{r}}^{(2)} = w_0^{(2)} + w_1^{(2)}\hat{\boldsymbol{r}}_1^{(1)} + w_2^{(2)}\hat{\boldsymbol{r}}_2^{(1)}$$

Error backpropagation

- As shown here, error backpropagation involves updating the weights in a backwards manner to reduce the error.
- This is an iterative process.

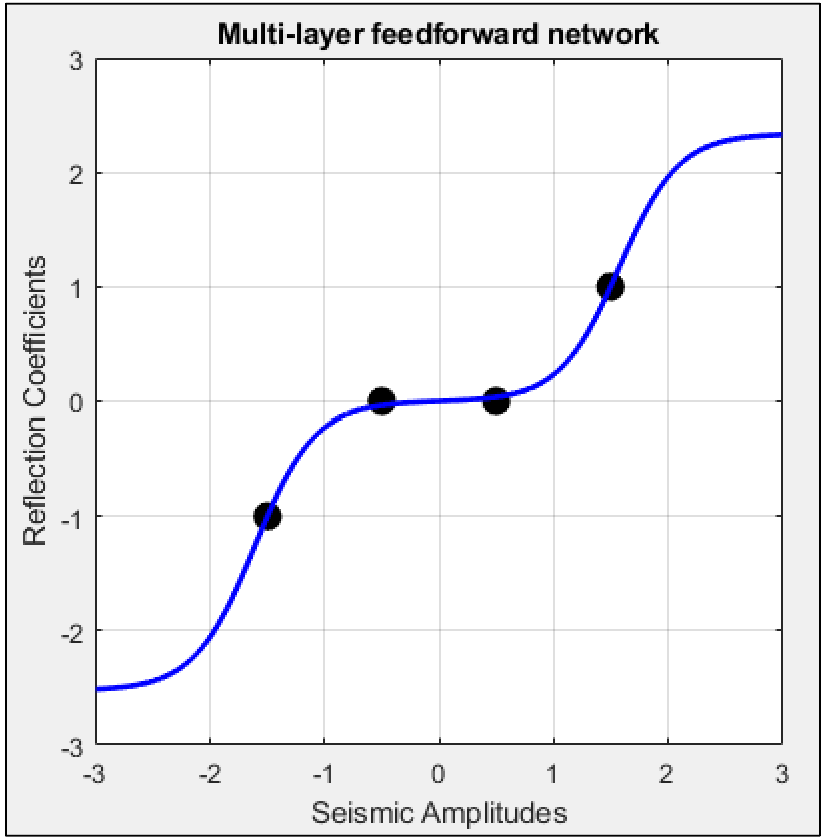


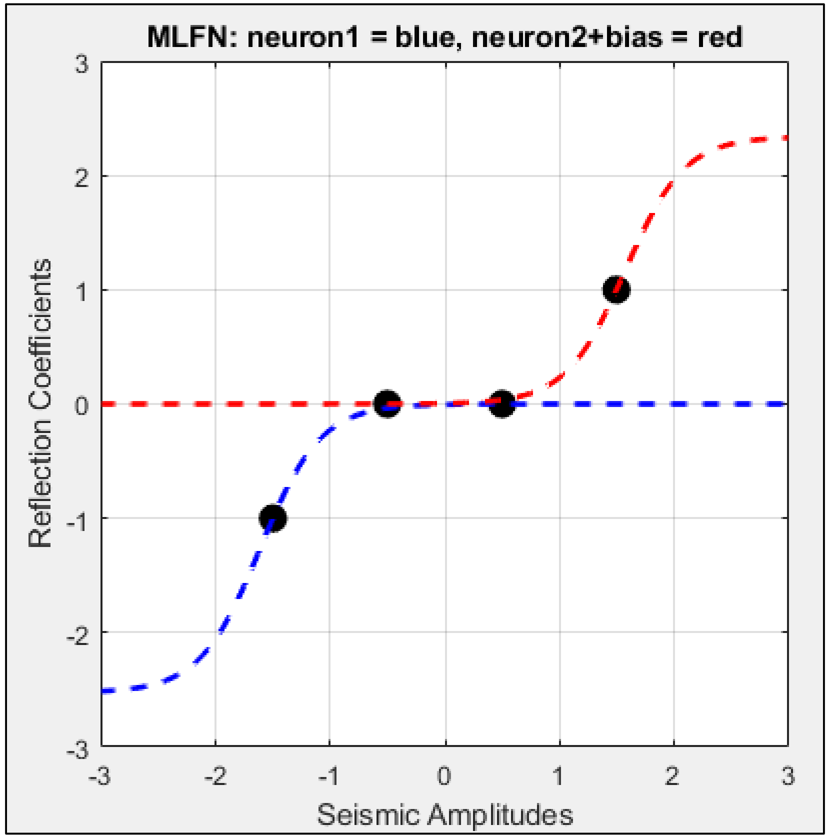
- The computation of the scaled final reflectivity after 10,000 iterations is:

$$\hat{r}_i^{(2)} = 2.34 - \frac{2.56}{1+\exp\left(6.10+3.78s_i\right)} - \frac{2.34}{1+\exp\left(-6.06+3.84s_i\right)}$$

Feed-forward network results

- The cross-plot of the output reflectivity against the input seismic is shown by the blue line.
- Note the good fit to the training points.





- As shown on the upper right, this is the sum of two “basis functions” given by:

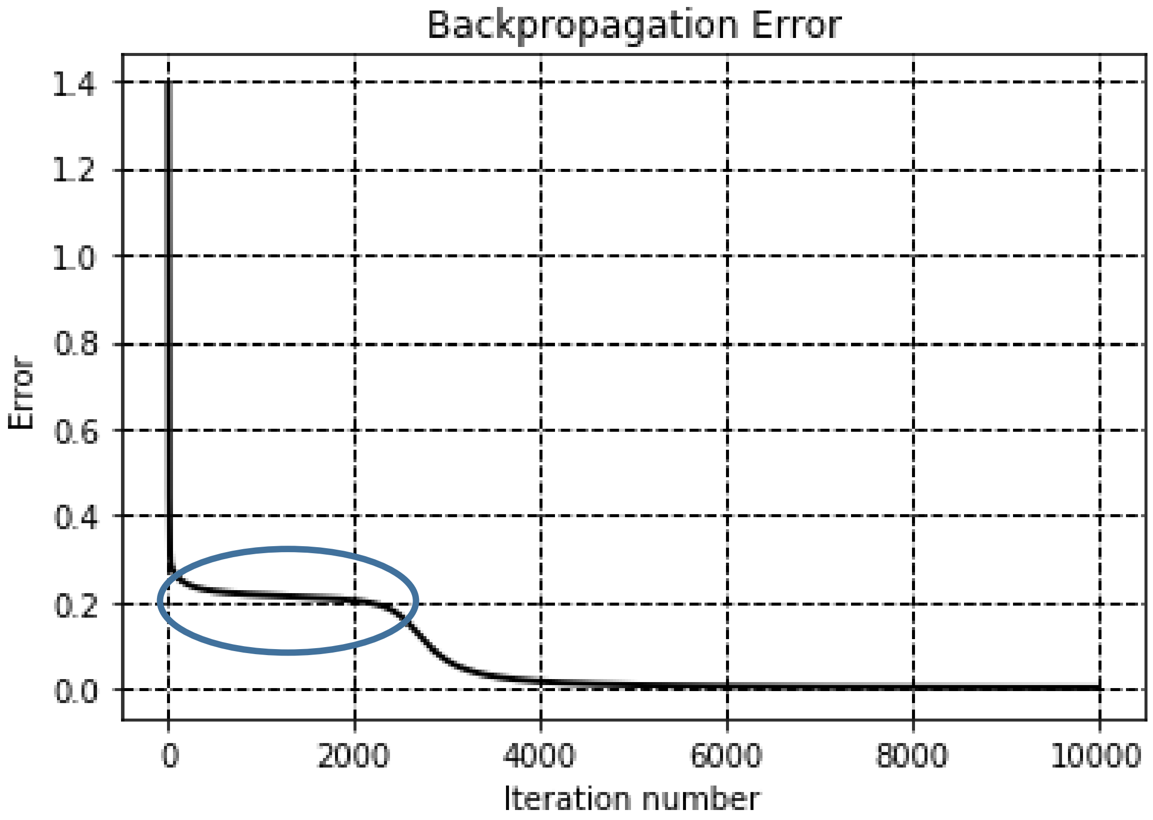
$$neuron\,1 = \frac{2.56}{1+\exp\left(6.10+3.78s_i\right)}, \quad bias + neuron\,2 = 2.34 - \frac{2.34}{1+\exp\left(-6.06+3.84s_i\right)}$$

The least-squared error

- The back propagation network approaches its answer in an iterative way.
- To understand the computation, we can compute the least-squared error after each iteration, using the formula:

$$E_{(k)} = \frac{1}{2}\sum_{i=1}^{4}(r_i - \hat{r}_{i(k)}^{(2)})^2$$

- The backpropagation error is shown in the graph on the right.
- Note that between iterations 100 and 2000 the network has become trapped in a local minimum.



- This can be a problem for back propagation networks, so let's now look at several different approaches.

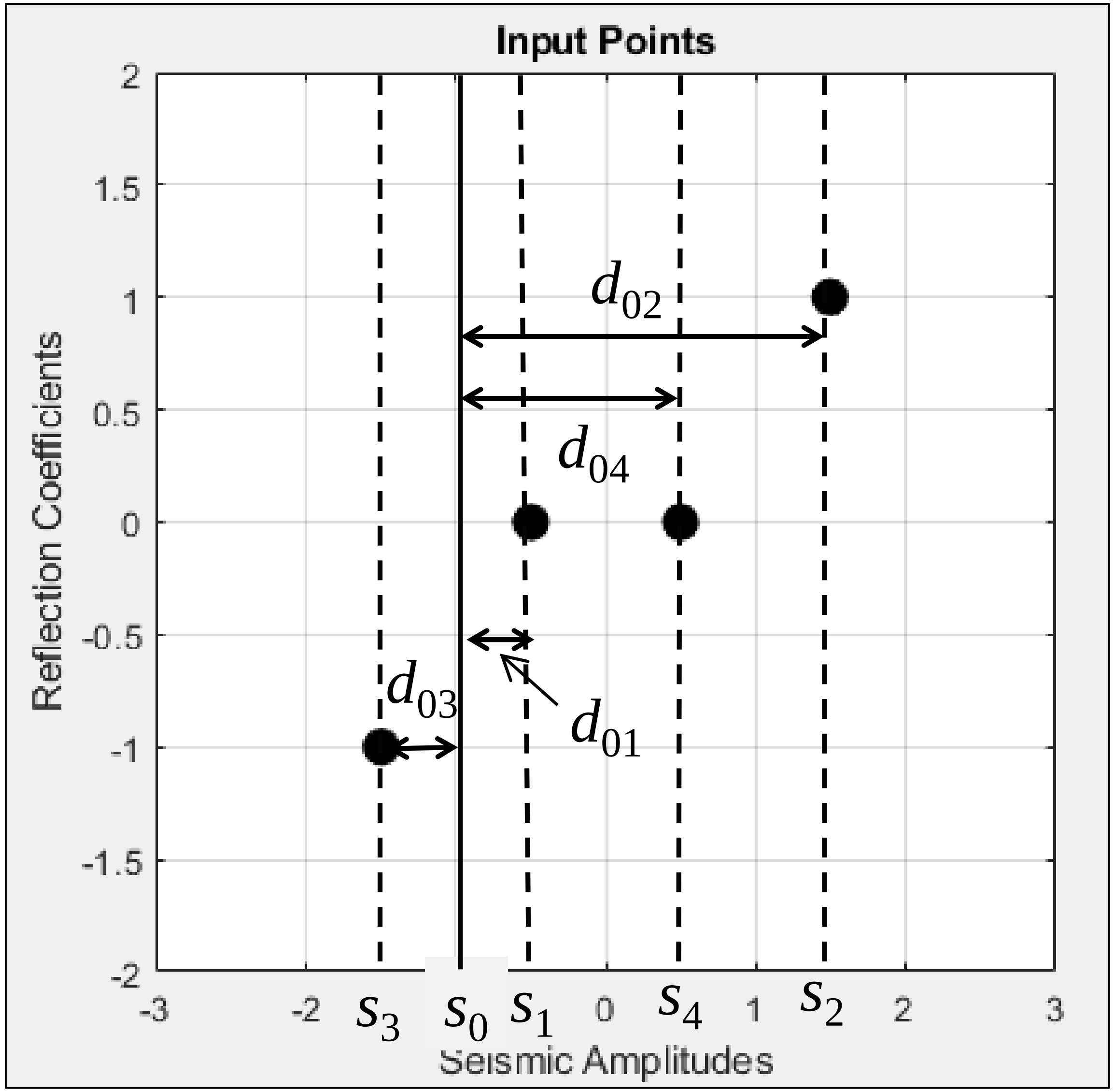
Gaussian basis function networks

- So let's now apply the Radial Basis Function Network (RBFN) and Generalized Regression Neural Network (GRNN).
- This takes me full circle to what Larry and I were working on almost 20 years ago.
- Both are based on the a weighted sum of the Gaussian of the distance squared between an output point and each input, divided by a variance:

$$\hat{r}(s_0) = w_1 e_{01} + w_2 e_{02} + w_3 e_{03} + w_4 e_{04}, \text{ where}$$

$$e_{0i} = \exp\left(-\frac{(s_0 - s_i)^2}{2\sigma^2} \right) = \exp\left(\frac{-d_{0i}^2}{2\sigma^2} \right).$$

- The figure shows an example with $s_0 = -1$.



RBFN weights

- For our deconvolution problem, the RBFN weights are computed by inverting the inter-point basis vector matrix and multiplying by the desired output:

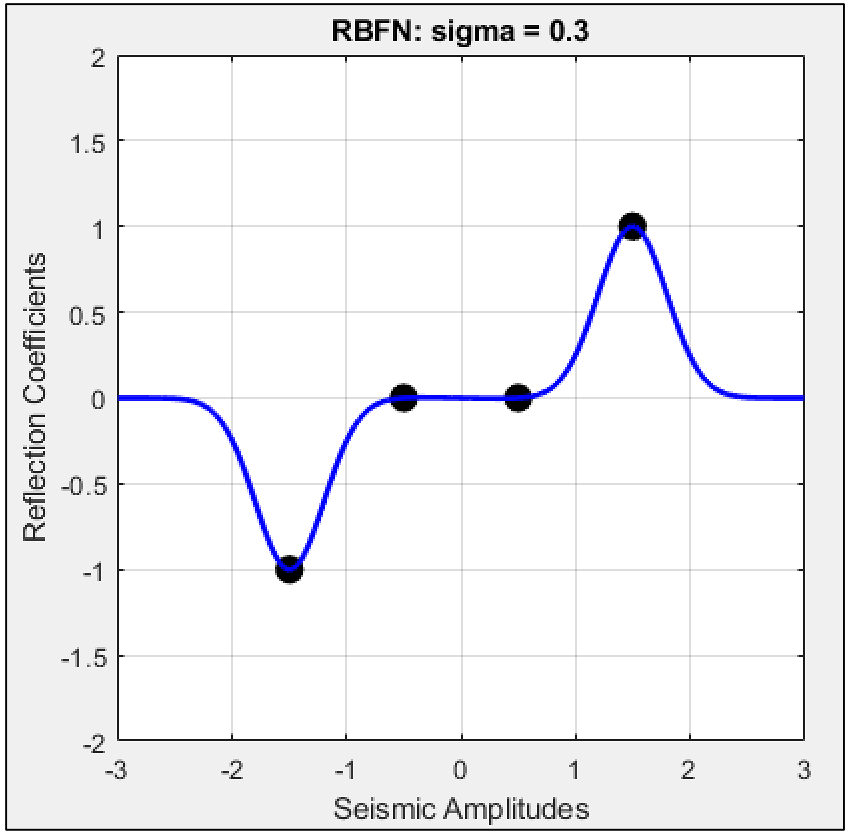
$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \Phi^{-1}\boldsymbol{r} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ +1 \\ -1 \\ 0 \end{bmatrix},\ \phi_{ij} = \exp\left(\frac{-\left(s_i - s_j\right)^2}{2\sigma^2} \right) = \exp\left(\frac{-d_{ij}^2}{2\sigma^2} \right).$$

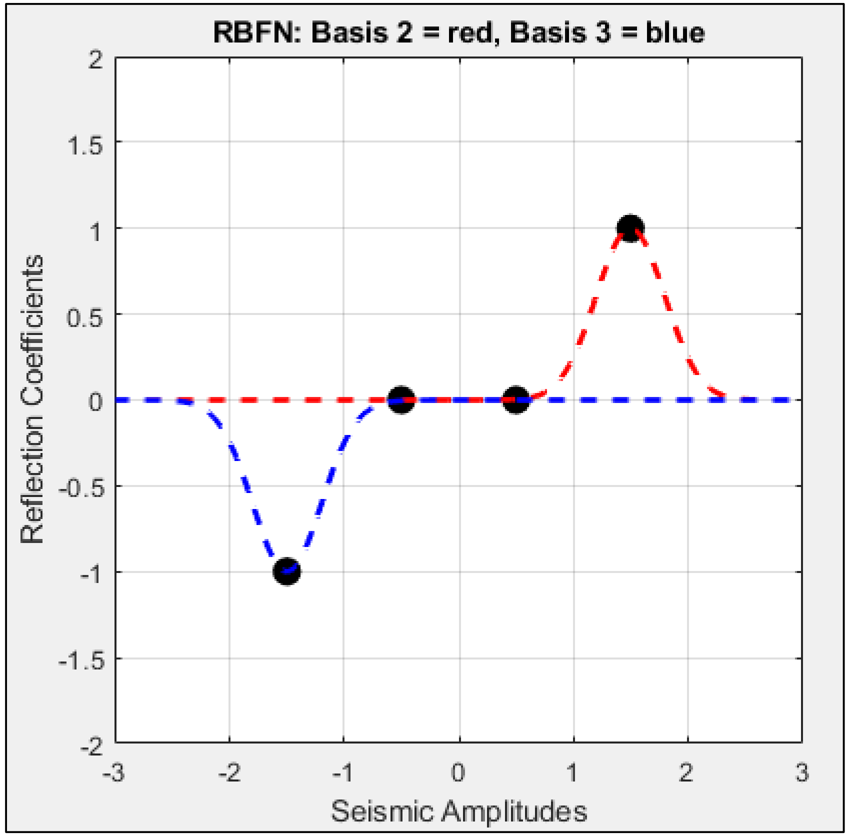
- For a small value of sigma, this means that the values on the main diagonal are equal to 1 and on the off diagonals are equal to 0, giving weights that are equal to the reflection coefficients:

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ +1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ +1 \\ -1 \\ 0 \end{bmatrix}$$

- This is shown on the next slide for $\sigma = 0.3$.

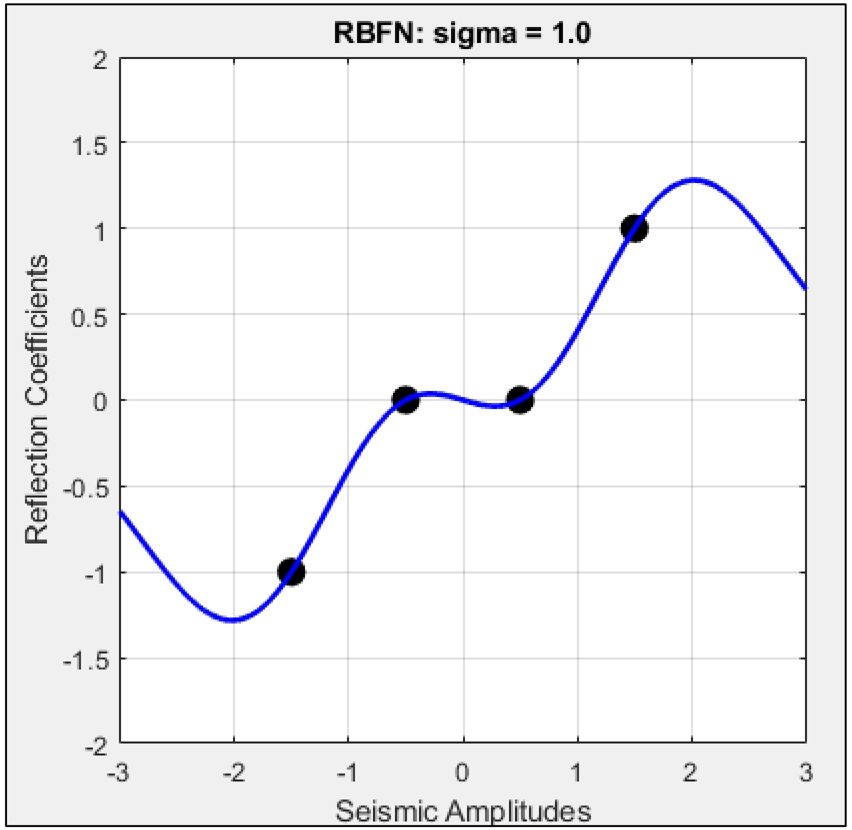
Radial basis function network results, sigma = 0.3

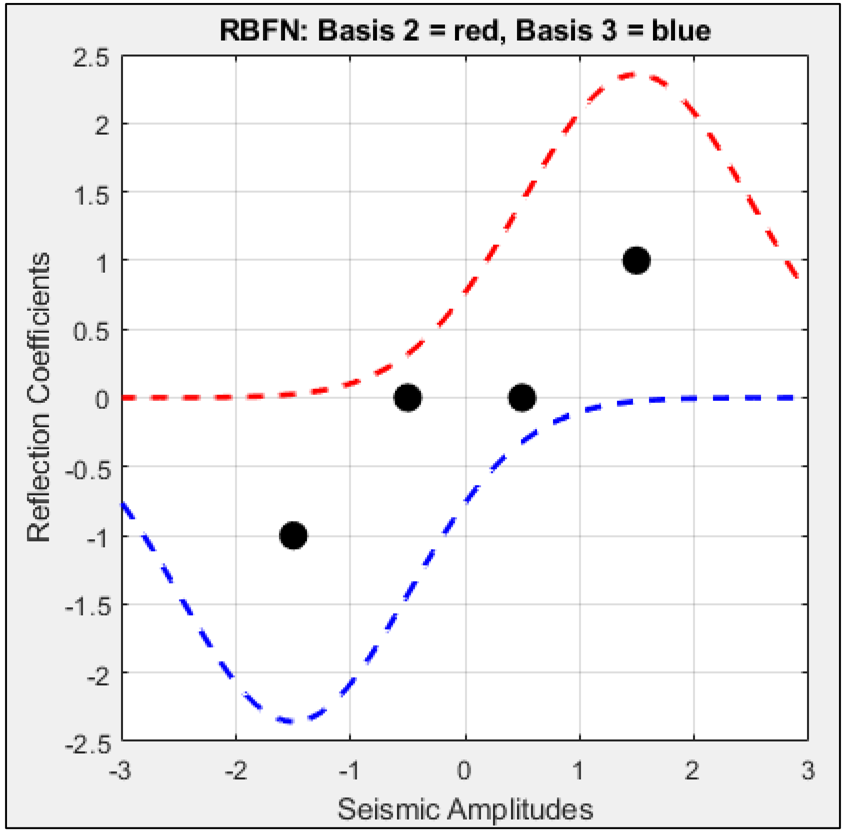




- As with the regression result, we can crossplot the output reflectivity against the input seismic, shown above left, where we get a perfect fit to the input points.
- The figure on the left shows the weighted RBFN Gaussian basis functions (note that there are four basis functions but two are equal to zero).

Radial basis function network results, sigma = 1.0





- Here, the sigma factor is equal to 1.0, which makes the Gaussian basis functions much wider.
- However, note that we still get a perfect fit to our points.

GRNN weights

- Unlike the RBFN weights, the GRNN weights are not pre-computed by inversion but are computed "on-the-fly" using the basis functions and desired outputs:

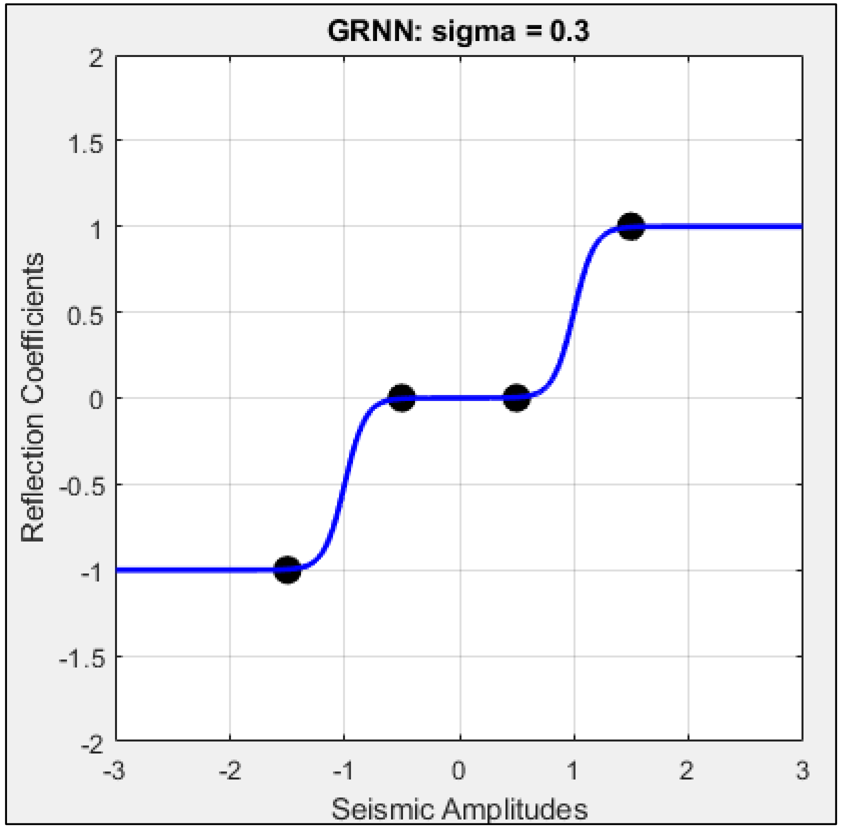
$$r(s_0)=\frac{r_1e_{01}+r_2e_{02}+r_3e_{03}+r_4e_{04}}{e_{01}+e_{02}+e_{03}+e_{03}}, \text{ where } e_{0i}=\exp\left(-\frac{(s_0-s_i)^2}{2\sigma^2}\right).$$

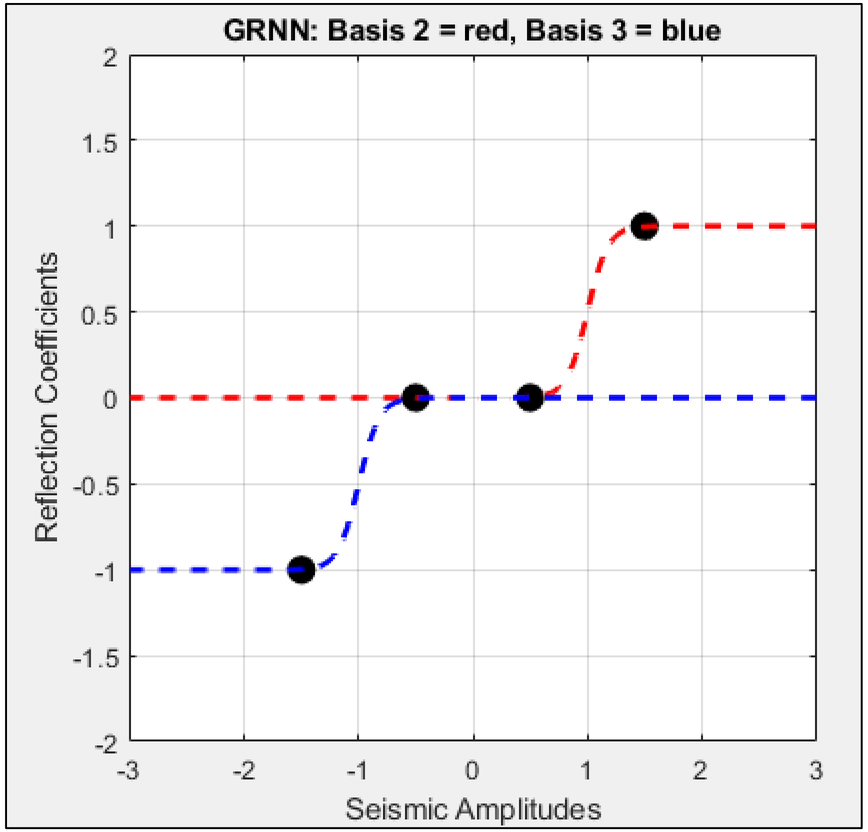
- For small sigma values, this again ensures a perfect fit at the output points.
- For example, at $s_0 = s_3 = -1.5$, we would get the exact result (for small sigma):

$$\hat{r}(-1.5)=\frac{0(0)+1(0)-1(1)+0(0)}{0+0+1+0}=-1$$

- This is shown on the next slide for all output points, using $\sigma = 0.3$.
- However, for $\sigma = 1.0$, on the following slide, the fit is no longer perfect.

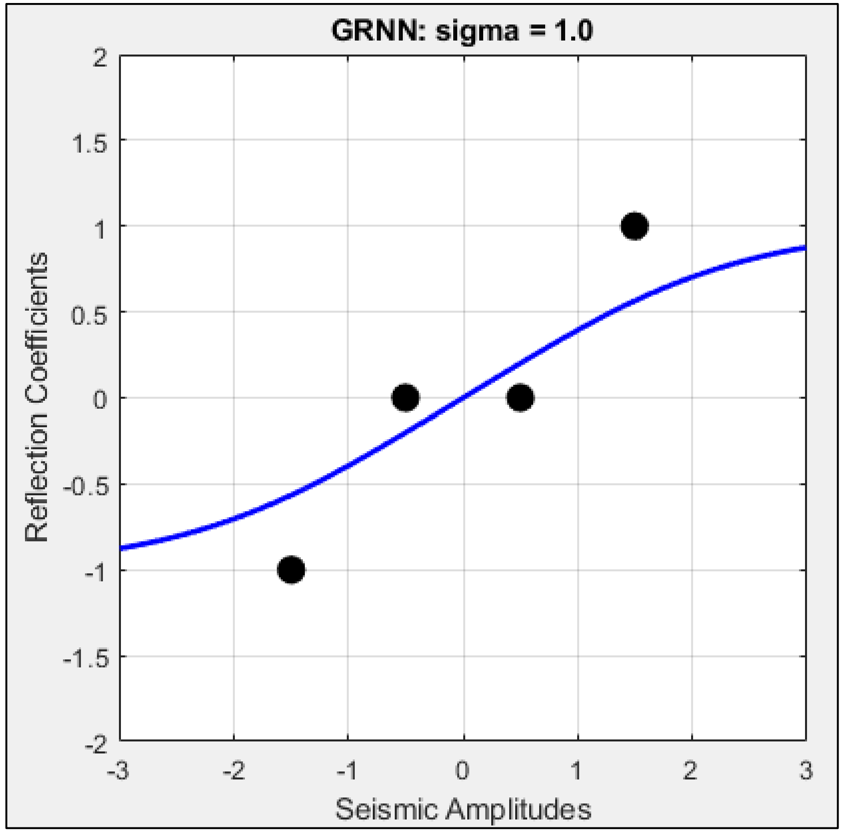
Generalized regression network results, sigma = 0.3

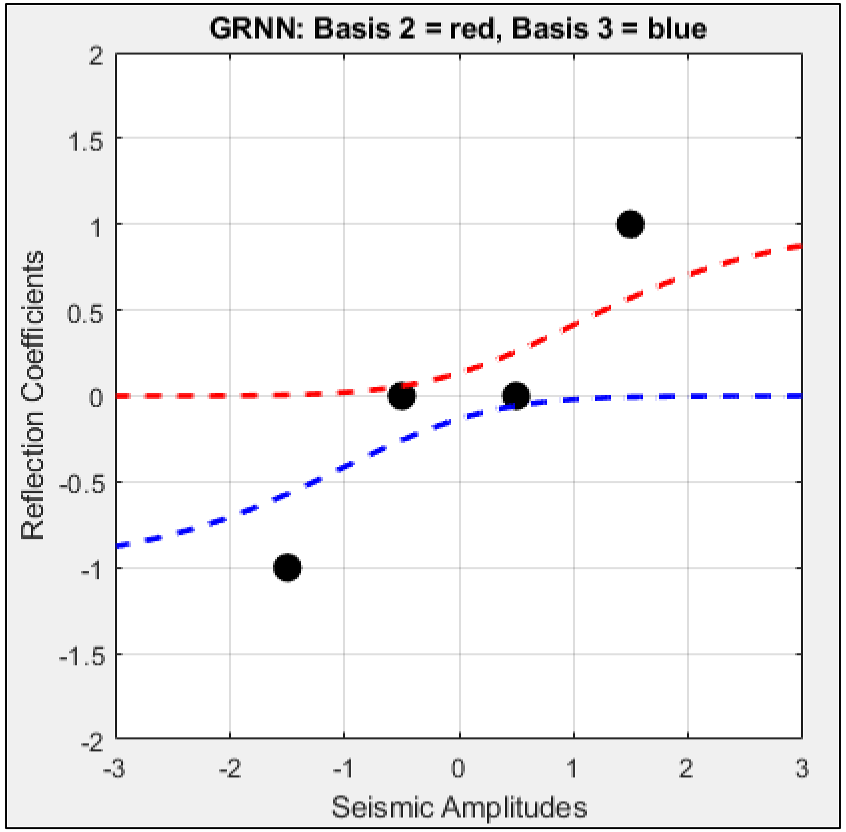




- Again, we can crossplot the output reflectivity against the input seismic, shown above left, where we get a perfect fit to the input points.
- The figure on the left shows the weighted GRNN Gaussian basis functions (note that there are four basis functions but two equal zero).

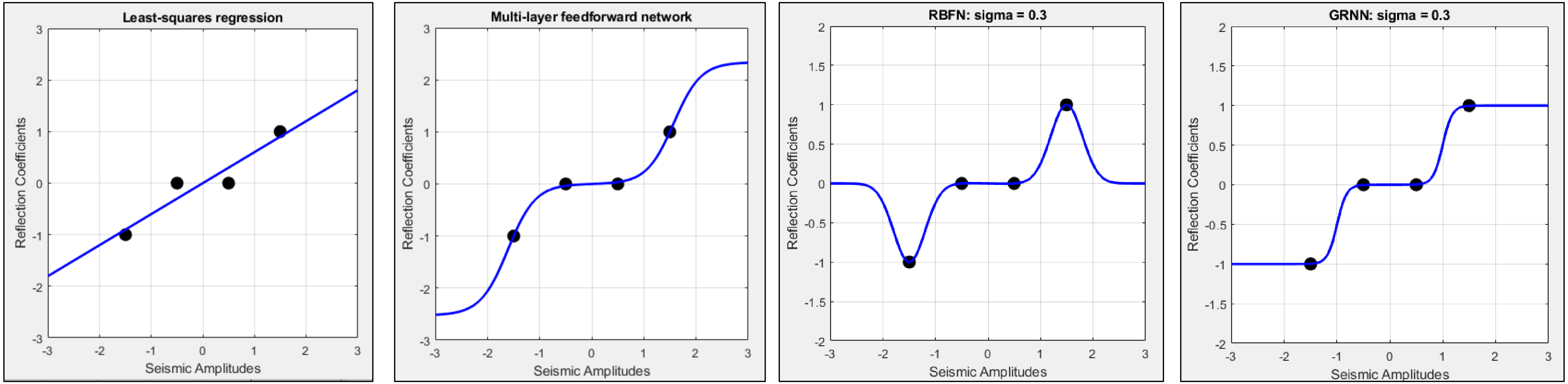
Generalized regression network results, sigma = 1.0





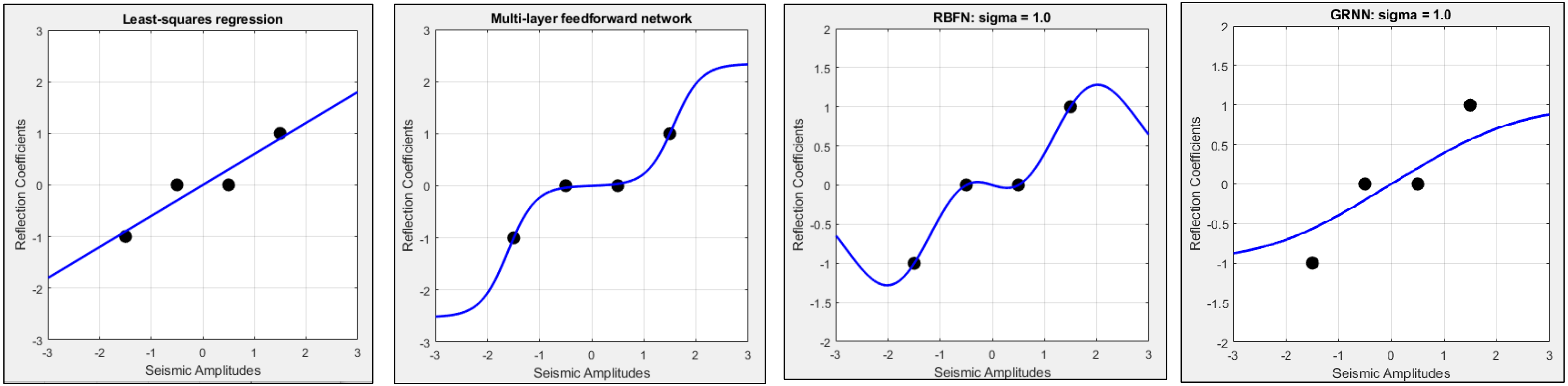
- Here, the sigma factor is equal to 1.0, which makes the basis functions much wider.
- Unlike the RBFN method, the fit to the points is not perfect, and is similar to a least-squares fit as sigma increases, except that the outside values of -1 and +1 are never exceeded (that is, the line does not go to + and – infinity).

Comparison of the regression results



- The linear regression on the left is quite simple to understand.
- The multi-layer feedforward network (second to left) is the sum of two nonlinear logistic basis functions (since we used two neurons).
- For small sigma values, the RBFN and GRNN networks on the right both give a perfect fit to the points by summing four Gaussian basis functions.

Comparison of the regression results



- The linear regression on the left is quite simple to understand.
- The multi-layer feedforward network (second to left) is the sum of two nonlinear logistic basis functions (since we used two neurons).
- For small sigma values, the RBFN and GRNN networks on the right both give a perfect fit to the points by summing four Gaussian basis functions.
- For large values of sigma, RBFN still fits the points, but GRNN does not.
- However, in practice, we have found that GRNN is less prone to over-fitting.

Conclusions

- In this talk I have attempted to honour Larry’s memory by talking about the work we have done together over the years and the influence that Larry has had on my own research.
- In the process, I have also tried to demystify machine learning algorithms and show that they have a definite mathematical structure that can be understood.
- Larry was a gentle soul but also had a formidable intellect and was a wonderful and beloved teacher.
- The best part of Larry was how interested he was in the science we were discussing and how his penetrating questions always improved my understanding of the topic at hand.
- I will really miss our walks in the park with Pearl!

Larry’s Last Lecture (courtesy Ahmed Elsabban)

- The one thing I did not mention in this talk was that for many years Larry and I co-taught a course on inversion through the SEG and CSEG.
- My favourite part was when I got to sit and watch Larry lecture.
- Like Ahmed, I attended Larry’s last lecture, and it is a memory I will always cherish.