Demystifying Machine Learning

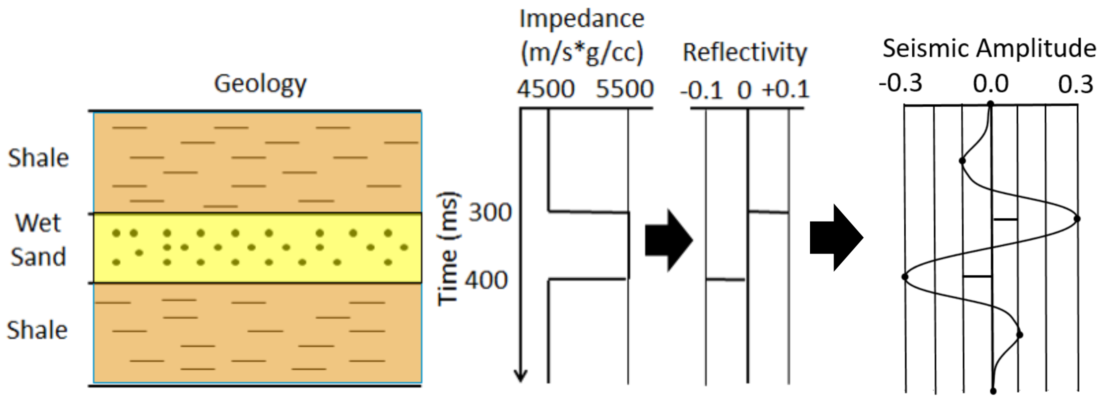
Brian H. Russell, CGG GeoSoftware

Summary

In this presentation, a numerical example is used to illustrate the difference between geophysical inversion and several machine learning approaches to inversion. The results will show that, like inverse geophysical solutions, machine learning algorithms have a definite mathematical structure that can be written down and analyzed. The example used in this study is the extraction of the reflection coefficients from a synthetic seismogram created by convolving a dipole reflectivity with a symmetric three point wavelet. This simple example leads to the topics of deconvolution, recursive inversion, linear regression and nonlinear regression using several machine learning techniques. The first machine learning method discussed is the Multi-Layer Feedforward neural Network (MLFN) with a single hidden layer consisting of two neurons. The other two methods which are discussed are the Radial Basis Function neural Network (RBFN) and the Generalized Regression Neural Network (GRNN).

The forward and inverse seismic model

The forward model used in this study is shown in Figure 1 (Russell, 2020), where the geology in Figure 1a consists of a thin porous wet sand of P-impedance 5500 m/s*g/cc between two shale layers of P-impedance 4500 m/s*g/cc (shown in Figure 1(b)), which results in reflection coefficients of + and –0.1 (Figure 1(c)). The reflectivity is then convolved with a three point symmetric wavelet given by (-1,2,-1) to produce the synthetic seismic trace as shown in Figure 1(d).



(a) (b) (c) (d)

Figure 1. The forward seismic model, where (a) is the geology, (b) is the P-impedance, (c) is the reflection coefficients, and (d) is the synthetic seismic trace.

Mathematically, we can write

$$\boldsymbol{s} = G\boldsymbol{r} = \begin{bmatrix} -1 & 0 \\ 2 & -1 \\ -1 & 2 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} +0.1 \\ -0.1 \end{bmatrix} = \begin{bmatrix} -0.1 \\ +0.3 \\ -0.3 \\ +0.1 \end{bmatrix}, \tag{1}$$

where $\boldsymbol{s}$ is the seismic trace vector, G is the geophysical wavelet matrix and $\boldsymbol{r}$ is the reflectivity. If we know the wavelet matrix we can invert equation (1) using least-squares deconvolution, as follows:

$$\boldsymbol{r} = (G^T G)^{-1} G^T \boldsymbol{s} = \begin{bmatrix} -0.3 & 0.4 & 0.1 & -0.2 \\ -0.2 & 0.1 & 0.4 & -0.3 \end{bmatrix} \begin{bmatrix} -0.1 \\ +0.3 \\ -0.3 \\ +0.1 \end{bmatrix} = \begin{bmatrix} +0.1 \\ -0.1 \end{bmatrix}, \tag{2}$$

which recovers the correct reflection coefficients. Next, linear regression and three different machine learning techniques will be discussed and compared to the inversion result.

Machine Learning

In the machine learning approach, we present both the input seismic trace and desired reflectivity to the machine learning algorithm and let the algorithm determine the relationship, as shown in Figure 2. That is, machine learning does not understand the physics of the problem, but develops a mathematical transform to convert the seismic trace into reflectivity.

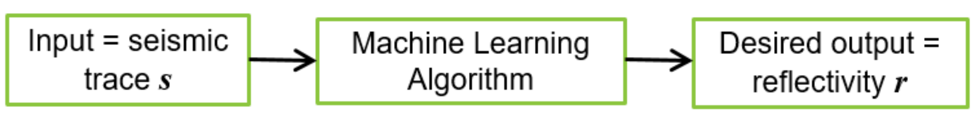


Figure 2. The basic concept behind Machine Learning

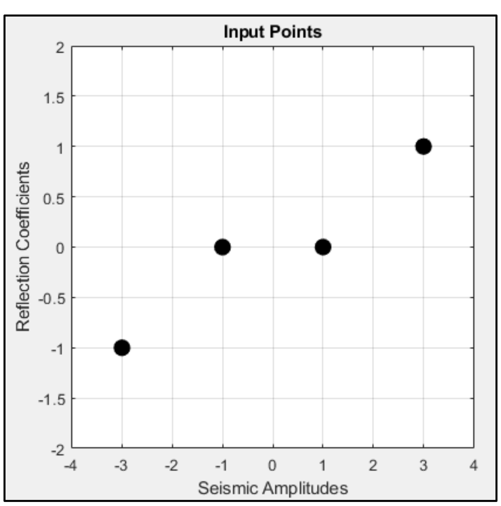
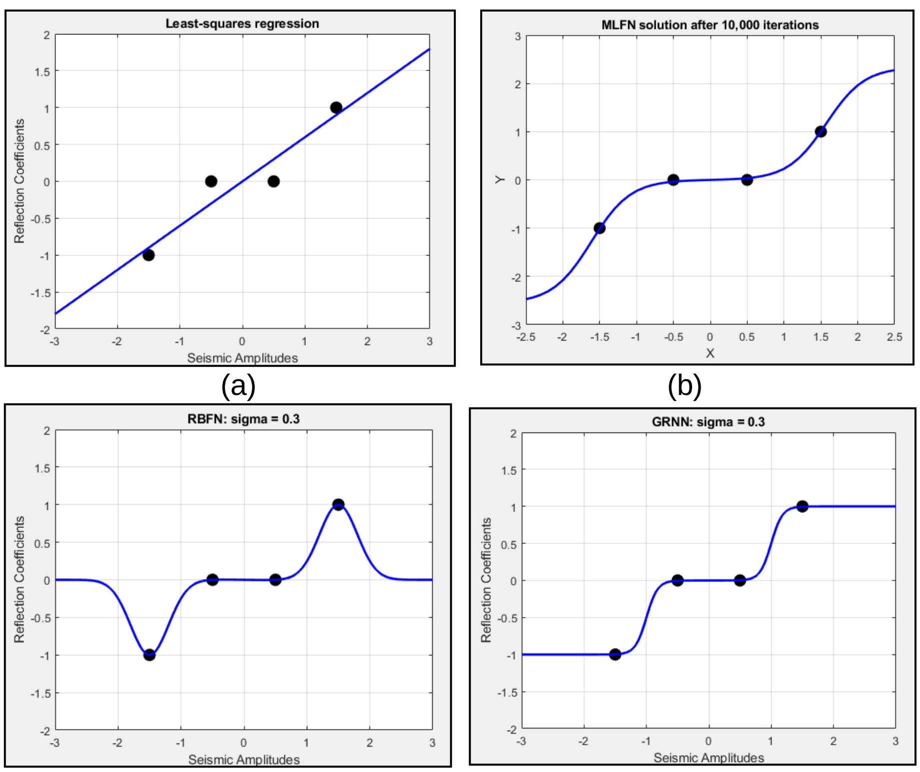


Figure 3. Machine learning input.

This means that the machine learning algorithm "sees" the problem quite differently than we see the problem as a geoscientist. For the simple problem shown in Figure 1, there are only four points input to the algorithm, the seismic amplitudes (-0.1, 0.3, -0.3, 0.1) and the desired output consists of the reflection coefficients (0, 0.1, -0.1, 0), where the two actual reflection coefficients are padded with zeros indicating that the wavelet side lobes are not real geology. This is illustrated by the cross-plot in Figure 3, where the x axis represents seismic amplitude and the y axis represents the reflection coefficients. The machine learning algorithms will find a "best fit" to these points.

The three machine learning methods discussed in this presentation are the multi-layer feedforward neural network (MLFN), with a single hidden layer consisting of two neurons, the radial basis function neural network (RBFN) and the generalized regression neural network (GRNN). All three of these networks involve different basis functions. The application of the feedforward neural network, which is the most common machine learning algorithm, was discussed by Russell (2019) in a TLE article and Russell (2000). In the MLFN the basis function is the sigmoidal logistic function (see Russell, 2020) and in both RBFN and GRNN the basis function is the Gaussian. However, for RBFN the weights are computed using a least-squares algorithm and in GRNN the weights are computed “on-the-fly” using the observed data. The MLFN algorithm is iterative and the key parameters are the initial random weights, the learning rate and the number of iterations. The RBFN and GRNN are not iterative and the key parameter in both methods is the width of the Gaussian, or sigma factor. Figures 4 (a) to (d) show a comparison of the results of linear regression and the three machine learning algorithms.



(a) (b) (c) (d)

Figure 4: The above figures show predicted reflection coefficient versus seismic amplitude for (a) linear regression, (b) the feed-forward neural network with 10,000 iterations and a learning rate of 0.2, (c) the radial-basis function neural network with sigma = 0.5 and (d) the generalized regression neural network with sigma = 0.5, where the black circles are the training points and the solid line is the fitting function (note that both input and output have been scaled for better neural network performance).

In all four figures, the horizontal axis represents the computed seismic amplitudes and the vertical axis represents the desired reflection coefficients. The black circles show the four training values, and the solid line shows the fitting function from the linear or nonlinear regressions. As mentioned in Russell (2019, 2020) neural networks can be thought of as an extension of linear regression, so the linear regression solution has been shown in Figure 4a as a comparison to the three machine learning methods. Note that the linear regression solution does not fit the points exactly. However, in the other three methods the fit to the points is perfect, although each method uses a different fitting function. For the MLFN result, 10,000 iterations and a learning rate of 0.2 were used, and for both the RBFN and GRNN results a sigma factor of 0.5 was used. The key observation about the differences in the three methods is how the extrapolation away from the points is done. In the MLFN, the extrapolation shows a slow absolute increase, but nonlinear rather than linear in the case of linear regression. In the RBFN, the value regresses to the mean of zero. In the GRNN the first and last values are simply extrapolated.

Conclusions

An obvious conclusion to this study is that applying physics to a problem is better than applying a machine learning algorithm like the MLFN, RBFN or GRNN, since then the solution has a real physical meaning and is not just a mathematical transform. But real geophysical problems are not that simple. When the geophysics is fully understood and applicable it is always the better option. However, our geophysical solution is usually overly simplistic (for example, in the real earth we have to take into account dispersion of the wavelet, inhomogeneity and anisotropy in the earth layers, etc.). Therefore, a neural network might find nonlinearities in the solution that we were unaware of in our theory. Second, our example consisted of very few points. In real geophysical studies we have large amounts of data, so a neural network might find hidden regularities in the data that we have overlooked. Furthermore, large amounts of data allow us to cross-validate our results by leaving parts of the data out of the initial training and using our trained weights to predict those parts of the data that were unknown to the neural network algorithm.

Acknowledgments

This talk is dedicated to the memory of my good friend and colleague Larry Lines. Many of the ideas in this talk were developed initially when Larry was my Ph.D. supervisor from 2000-2004 at the University of Calgary and then fleshed out in our frequent Sunday walks at Varsity Ravine Park with Pearl, Larry's faithful Malamute.

References

Russell, B.H., 2019, Machine learning and geophysical inversion — A numerical study: The Leading Edge, 38**,** no. 7, 512-519.

Russell, B.H., 2020, Machine learning and geophysical inversion: Extended Abstract, geo**convention** 2020, Calgary, Alberta.