Permeability Prediction of 3-D Binary Segmented Images using Neural Networks

Nattavadee Srisutthiyakorn

Abstract – 3-D digital rock image have recently become an important piece of information about the hydrocarbon reservoir properties. The goal of this project is to explore and employ machine learning as a tool to better understand the 3-D digital rocks from geometric measurement and features extracted from the image.

Introduction

This project applies machine learning to 3-D binary segmented images of the Fontainebleau and Berea sandstone, with the intention of finding a robust alternative way to estimate transport properties such as permeability. The permeability is one of the key to understand the nature of hydrocarbon reservoir and estimate its production capability. Conventionally, permeability is obtained from laboratory measurement of a rock core, which can take up to months to be completed.

Recent technology in high resolution x-ray tomography have led to the increase in digital rock database. 3-D binary segmentation is applied after scanning. For single-phase fluid flow, the Lattice-Boltzmann method is the established method for solving absolute permeability. The Lattice-Boltzmann method approximates the Navier-Stokes equations at the pore scale. The calculation can be computationally expensive for large digital rock images. The calculation assumes no flow boundary conditions, and that the permeability depends only on pore geometry. In our calculations, the pressure gradient is only along the x-axis so that $\frac{dP}{dx} \neq 0$. The Lattice Boltzmann is implemented in C++ that is wrapped in MATLAB with the inputs being the 2-D or 3-D images and the size of the voxel. The Lattice Boltzmann flow simulation has been demonstrated to be a robust method for flow simulation in complex structures by comparison with laboratory measurement (Kheem, 2003).

On the other hand, geometric measurements and 2-D/3-D patterns are computationally inexpensive even at the larger scale, and it can provide insight about the structure of the pores and perhaps how the structure relates to the flow properties. In this project, machine learning is applied to understand the relationship between geometry and extracted features of rock images to permeability, in the hope to improve accuracy and reduce computational time of permeability calculation.

Data Processing/Features Extraction

Data includes 3-D Binary Segmented Images (64 images of size 50 voxels from Fontainebleau Sandstone and 1000 images of Size 100 voxels from Berea Sandstone) from the Digital Rock Physics Benchmarks Paper (Figure 1). Numerical simulation of Lattice Boltzmann permeability of each image was computed to be used as the target in supervised learning. Raw Inputs are binary segmented 3-D images where at each voxel, the value 1 represents solid and 0 represents pore. Each type of feature is extracted from 3 multi-scales: (a) original images, (b) 5x-upscaled images (Figure 2), and (c) 10x-upscaled images. The upscaling is done by averaging the value in each voxel in specified ranges. Then, if the average is higher than the specified threshold for pore, the value of new pixel is assigned to be pore. Upscaling ensures that the convolution of the 2-D and 3-D patterns are also from global scale.

Figure 1:3-D images of Fontainebleau sandstone (left) and Berea sandstone (right). The white color represents grains and black color represent pore space.

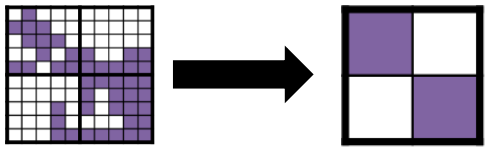

Figure 2: Example for the 5 times upscale in 2D images.

Three types of features are:

(1) Minkowski Functionals

Minkowski Functionals encompass standard geometric measurements for a binary segmentation image. For d-dimensional space, there are d+1 associated Minkowski measurements (Vogel, 2010). For example, a 2-D slice defines 3 Minkowski measurements (area, perimeter, Euler characteristic), and a 3-D solid defines 4 Minkowski measurements (M_0 - volume, M_1 - surface area, M_2 - integral of mean curvature (mean breadth), and M_3 - Euler characteristic). The units of Minkowski measurements are $L^3, L^2, L^1,$ for M_0, M_1, M_2 respectively, while M_3 is dimensionless.

M_0 - *Volume (L^3)*

$$M_0(\mathrm{X}) = \mathrm{V}(\mathrm{X})$$

M_1 - *Surface Area (L^2)*

$$M_1(\mathrm{X}) = \int_{\delta x} \mathrm{ds} = \mathrm{S}(\mathrm{X})$$

M_2 - *Integral of Mean Curvature (L)*

$$M_2(\mathrm{X}) = \frac{1}{2}\int_{\delta x}\left[\frac{1}{r_1}+\frac{1}{r_2}\right]\mathrm{ds} = \mathrm{C}(\mathrm{X})$$

M_3 - *Euler Characteristic (Unitless)*

$M_3(\mathrm{X})$ = vertices - edges + faces - solid

(2) 2-D pattern distribution

2-D pattern distribution can be derived from the convolution between the pattern and the image. For 2-D, patterns derived from the cross shape template as in Figure 3 has been employed. There are 4 adjacent pixels to the center in order to form the template. Thus, there are 2^4 = 16 combinations of pattern (Figure 3). After convolution, the number of times that the pattern appears in the image can be obtained directly by counting the pixel that has the value equals to the number of pixels in the pattern (Figure 4). If inputs are from 3 multi-scales (original, 5x upscale, 10x upscale) then the total number of 2-D pattern is 16*3 = 48 features.

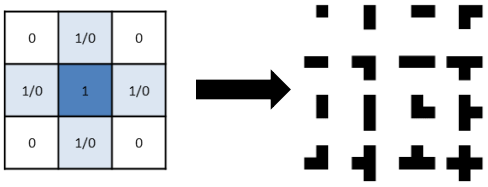

Figure 3: 16 possible 2-D patterns derived from the cross-shape template.

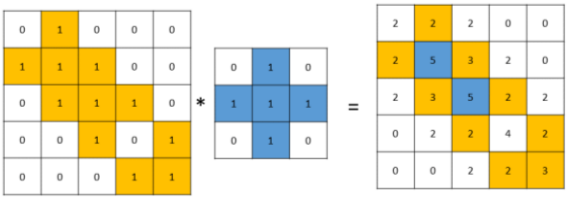

Figure 4: Convolution of a sample image and a pattern. After convolving, the pixel that has value equal to 5 (the number of total pixel in the pattern) indicates the location of pattern found in the image.

For 2-D pattern, if 3-D images have the size of 50x50x50 pixels, then the 3-D images can be sliced into 50 2-D images and the pattern distribution can be added from every 2-D images to form the pattern distribution.

(3) 3-D pattern distribution

For 3-D pattern, there are 6 adjacent pixels to the center of the 3-D the cross shape template, resulting in 2^6 = 64 combinations of pattern. If inputs are from 3 multi-scales then the total number of 3-D pattern is 64*3 = 192 Features.

Methodology

Data is divided into 3:1:1 ratio for training, testing and validation set. For 5 groups of data, if one group is selected to be the test set and the rest to be training sets, then the calculation is only 5-fold. Yet, in Multilayer Neural Network validation set is required to stop the training early, in order to prevent over-fitting. Hence, there are the total 20-fold combination of cross validation (Figure 5). After 20-fold calculation, the mean square error are calculated from the average of 20 cases. For each type of model, I have varied the number of nodes (5,10,20) and hidden layers (1 to 5).

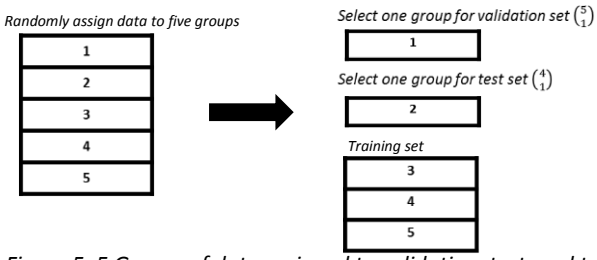

Figure 5: 5 Groups of data assigned to validation, test, and training set.

(1) Multilayer Neural Network (MNN)

Multilayer Neural Network with Feed Forward and Multilayer Neural Network with Back Propagation, Bayesian Regularization are employed in this study to estimate the permeability. There are four steps for neural network design: (1) create a network, (2) configure the network, (3) train the network, and (4) validate the network. There are a few concerns in network configuration such as how to divide data for training, testing, and validating the network, or what would be an appropriate number of nodes in a hidden layer. These questions are vital for deriving and constructing a better network.

For network construction, tan-sigmoid function (a = tanh(x)) and positive linear function (or rectified linear) (a = max(0,x)) are tested in the hidden layers, and a linear function is used in the output layers. The number of nodes and the number of hidden layers are tested to obtain the optimal network structure using the feed-forward neural net. Training is performed through feed-forward and through back-propagation with Bayesian Regularization using gradient descent algorithm. Bayesian Regularization can be used to help achieve the goal of improved generalization. This can be accomplished by modifying the previous performance function by adding another term that includes the mean of the sum of squares of the network weights and biases. Levenberg-Marquardt algorithm is the combination of Gauss-Newton and Steepest Descent algorithm. The combination prevents the case that the Hessian matrix is not invertible. If μ = 0, Levenberg-Marquardt is equivalent to Gauss-Newton and if μ -> ∞, the equation approaches Steepest Descent.

$$Xx_{k+1} = x_k - [H(x_k) + \mu I]^{-1} J^T(x_k) e(x_k) \ , where\ e = error\ vector$$

The performance can be regularized to prevent over fitting as followed

$$msereg = (1-\delta) \cdot \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 + \ \delta \cdot \frac{1}{M}\sum_{i=1}^{M}(w_i)^2$$

With this performance function, it is possible to minimize both mean square errors and mean square weights. The function therefore forces the network to have a smaller weight and bias, leading to smoother output.

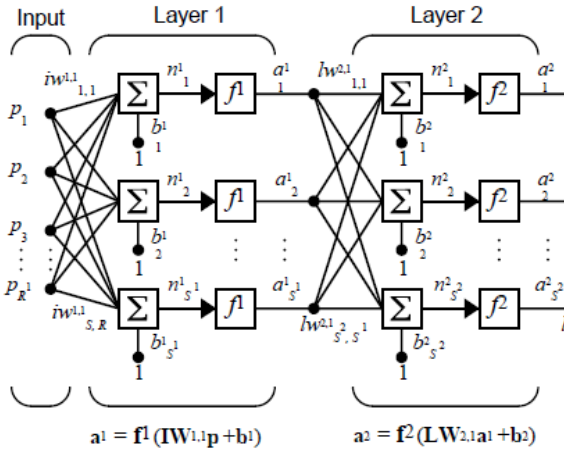

Figure 7: Example of network architecture (Demuth and Beale, 2000).

(2) Convolutional Neural Network (CNN)

The convolutional neural network contains one extra convolution layer, which followed by general multilayer neural network as described in the previous section. For the convolutional layer, the features are 2-D and 3-D pattern distribution extracted from original images and upscaled images (Figure 4).

Result

Model from Feed Forward Network	# Features	Train MSE	Test MSE	Iterations
MNN with Minkowski Functionals	4	0.2953e4	3.4147e5	10
MNN with Minkowski Functionals (multi-scale)	504	1.4159e5	3.3678e5	10
CNN with 2-D Convolution	16	1.6324e5	3.6123e5	16
CNN with 2-D Convolution (multi-scale)	48	0.8750e5	2.4307e5	13
CNN with 3-D Convolution	64	1.6410e5	3.3144e5	11
CNN with 3-D Convolution (multi-scale)	192	0.6352e5	2.7400e5	10

Table 1: Results from Feed-Forward MNN using tan-sigmoid function

Model from Bayesian Regularization Network	# Features	Train MSE	Test MSE	Iterations
MNN with Minkowski Functionals	4	2.1379e5	1.1631e5	53
MNN with Minkowski Functionals (multi-scale)	504	4.6362e5	2.3999e5	416
CNN with 2-D Convolution	16	1.6802e5	5.2754e4	50
CNN with 2-D Convolution (multi-scale)	48	1.1989e5	1.1049e5	77
CNN with 3-D Convolution	64	2.1789e5	8.4937e4	57
CNN with 3-D Convolution (multi-scale)	192	4.6362e5	2.3999e5	283

Table 2: Results from Beyesian Regularization MNN using rectified linear or positive linear function

The table 1 and 2 summarize the case with the best test performance (mean square error for feed-forward and regularized mean square error for Bayesian Regularization) for each model.

Figure 8 shows both training and test MSE for each model and for different network architecture from the feed-forward neural net, where x axis represents the number of nodes (5,10,20) and y axis represents the number of hidden layer. There is no unique network architecture for digital rock images as different type of feature has different optimum neurons and hidden layers. Although there is not a unique answer. There are two observable trends from the feed-forward test MSE: (1) CNNs with multi-scale favor larger network with more number node nodes in hidden layers and (2) MNN and 2-D CNN with only original images favor small and simple network. This may be due to the number of features supplied to the network is small.

CNN with 2-D convolution with multi-scale (original, 5x upscale, 10x upscale) shows the best testing result overall from both feed-forward and Bayesian regularization. The highest test MSE is from Minkowski Functional at original scale is as expected since it only contains 4 features.

The regression plots between the predicted data and target of features are shown in figure 9. The prediction is generally in agreement with the target permeability.

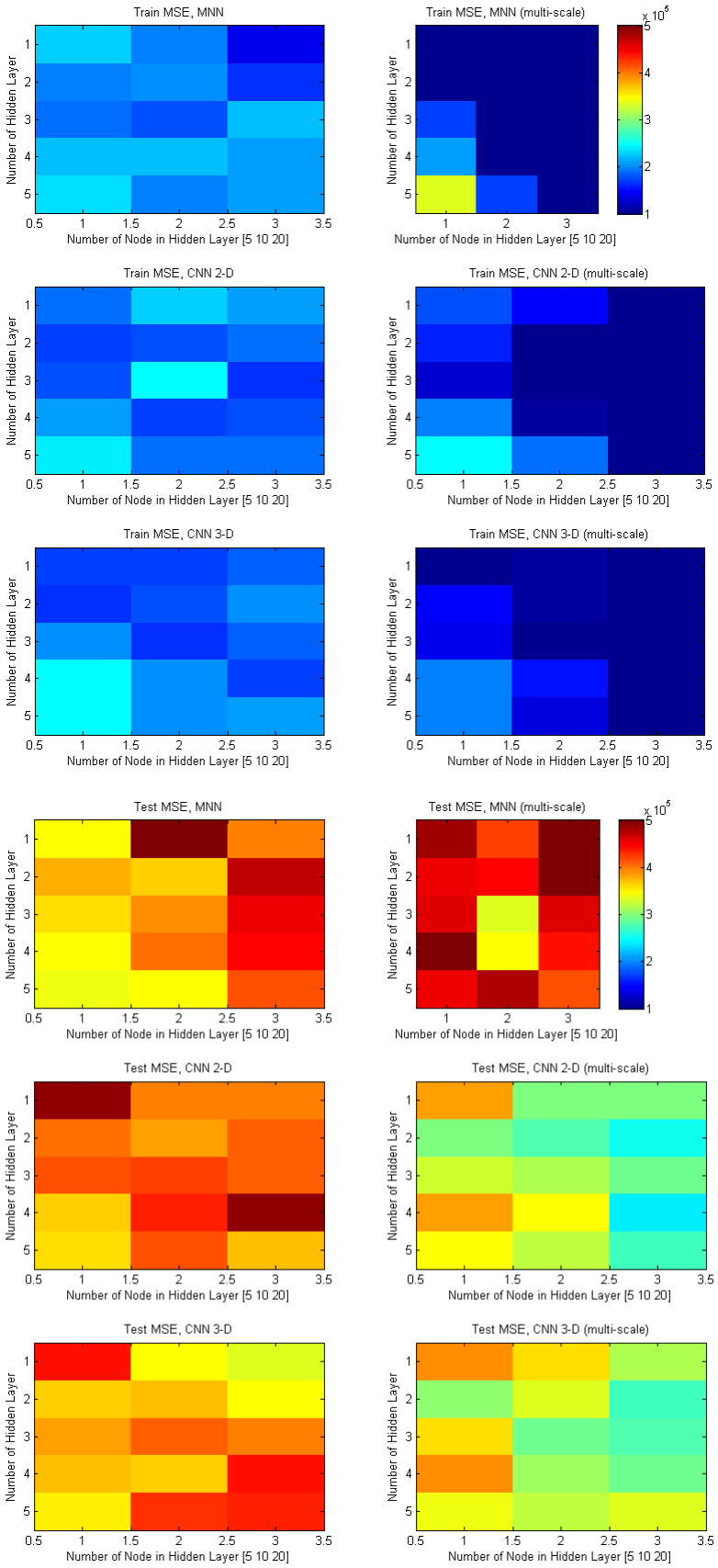

Figure 8: Mean square error of each network architecture from (a) Training set and (b) Test set. The y axis is the number of hidden layer from 1 to 5 and x axis is the number of nodes where 1 2 3 correspond to 5, 10, 20 number of nodes consequently.

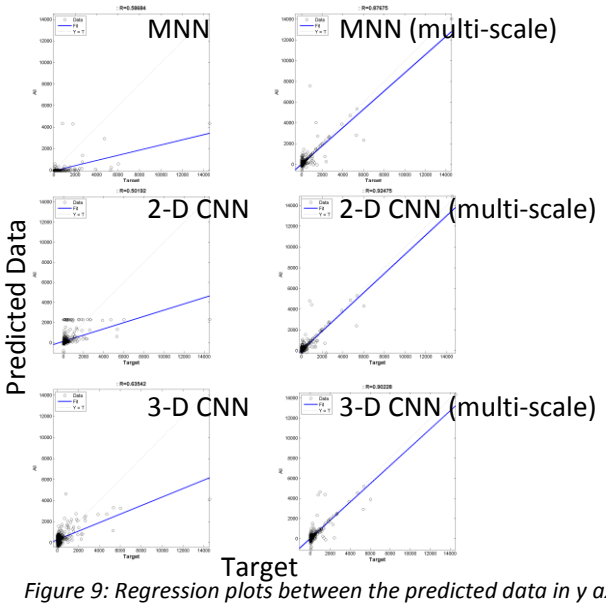

Figure 9: Regression plots between the predicted data in y axis and target in x axis for each model. This is from the feed-forward network. The multi-scale models have better regression since they are near 45 degree line.

Discussion

Obtaining features in multi-scale helps improve the fit tremendously. This may be due to features from larger scale help the network to capture global structure of the 2-D/3-D images. The convolution in larger scale gives better understanding of how the pore space connects in larger scale.

This study shows that 2-D CNN performs better than geometric measurement (Minkowski Functionals). As the cost of acquiring 2-D digital images is lower than that of 3-D images, 2-D CNN then offers a good alternative when it is not possible to obtain the 3-D images for predicting the permeability.

The test MSE is as good as the train MSE in Bayesian Regularization because of the generalization. In some cases, the test MSE is lower and this may indicate that the network is overly smooth or underfit.

As seen from the number of iterations, Multilayer neural network with Bayesian Regularization takes longer time to train than the feed-forward network so this may not be practical to use when there is a need to quickly approximate the permeability. In the future, I plan to test the network architecture on Bayesian Regularization network to see whether the optimum network architecture (number of nodes and number of hidden layers) matches that of feed-forward network or not.

Using positive linear function (or rectified linear) in transfer function helps improving the fit in Bayesian Regularization compared to the tan-sigmoid.

Conclusion

The result of using machine learning to predict permeability is promising, especially for the case of 2-D CNN in multi-scale. Pixel-based convolution helps linking between 2-D structures of rock to 3-D properties of permeability. This property will be useful for when only 2-D data of the rock can be obtained. In the future, I aim to expand the work to cover other lithologies such as shale and carbonate. There is also the need to optimize the network further to get faster and reliable computation.

References

Andra, H., et al., 2013, Digital rock physics benchmarks - Part I: Imaging and segmentation: Computers & Geosciences, 50, 25-32.

Demuth H., and M. Beale, 2000, Neural Network Toolbox For Use with MATLAB, User's Guide: The MathWorks, Inc.

Egmont-Petersen, M., D. de Ridder, and H. Handels, 2001, Image Processing with neural networks – a review. Pattern Recognition, Vol. 35.

Keehm, Y., 2003. Computational rock physics: Transport properties in porous media and applications. Ph.D. Dissertation. Stanford University, 135pp.

Legland, D., K. Kieu, and M. Devaux, 2007, Computation of Minkowski measures on 2D and 3D binary images. Image Analysis and Stereology, Vol 26.

Ng, A. et al. "Unsupervised Feature Learning and Deep Learning Tutorial." Internet: ufldl.stanford.edu, 2013 [Nov. 1, 2014].

Remy N., A. Boucher, J. Wi, 2009, Applied Geostatistics with SGeMS: A Users' Guide, Cambridge University Press.

Vogel H.-J., U. Weller, S. Schluter, 2010, Quantification of soil structure based on Minkowski functions: Computers & Geosciences, 36, 1236-1245.