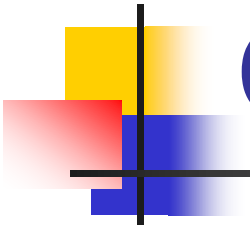
# Simultaneous Simulation and Modeling of Density, Porosity and C/O tools

**Fusheng Li, Robin P. Gardner**

**Center for Engineering Applications of Radioisotopes (CEAR)**
**Nuclear Engineering Department**

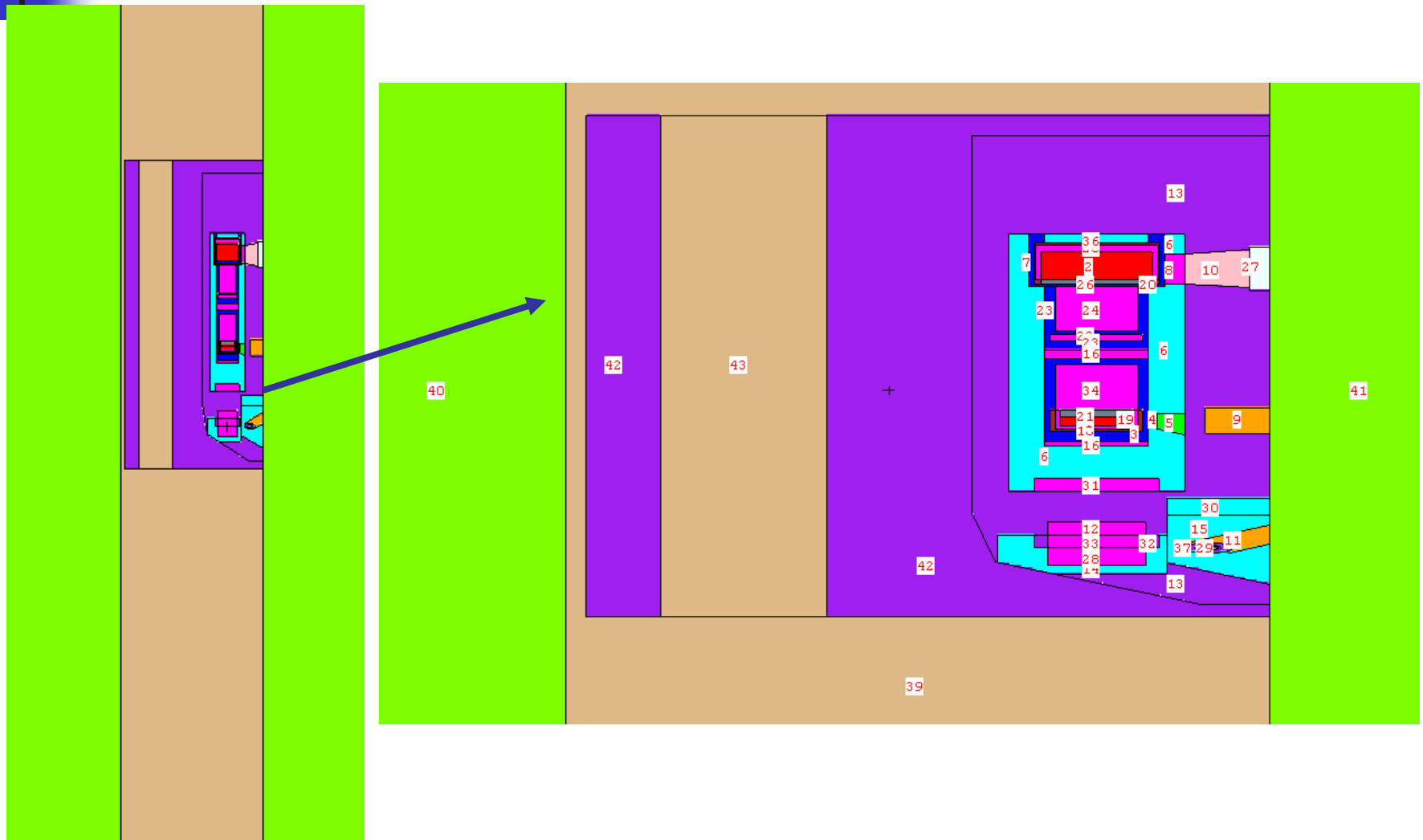
**Oct 20 2005**



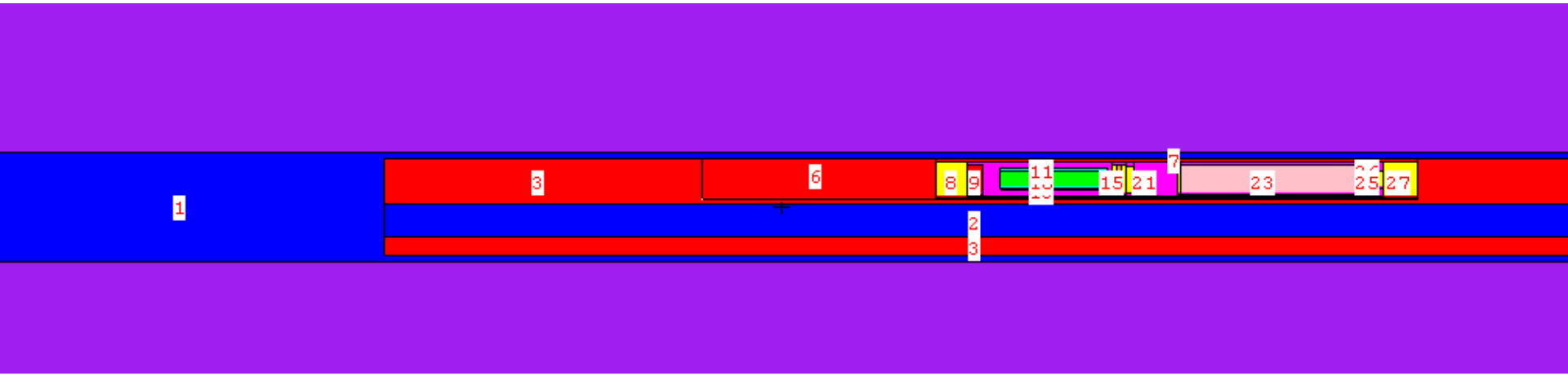
# Outline

- Simulations of density, porosity and C/O tool.
- Use neural nets to model density, porosity tools simultaneously.

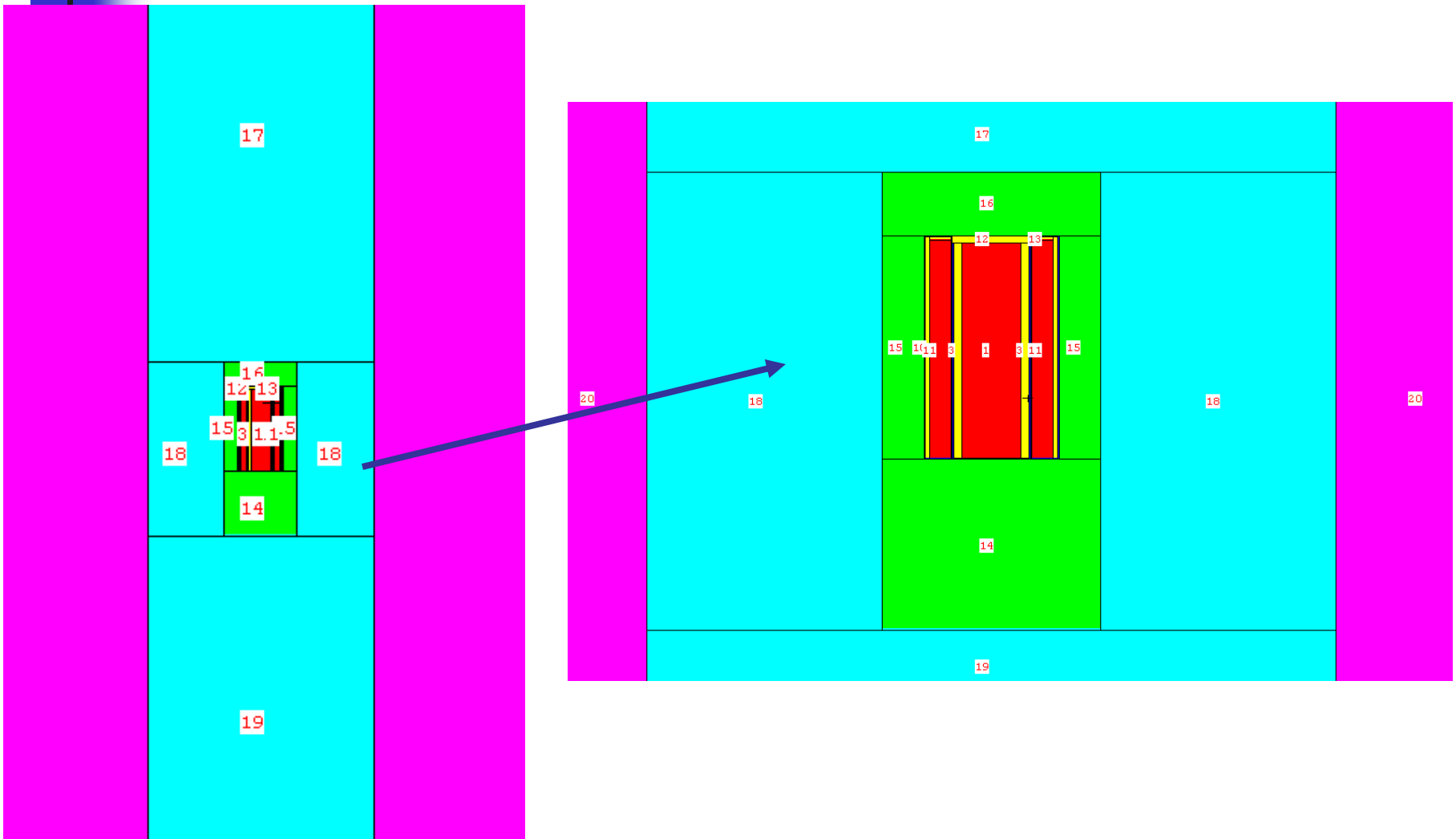
# Density Tool



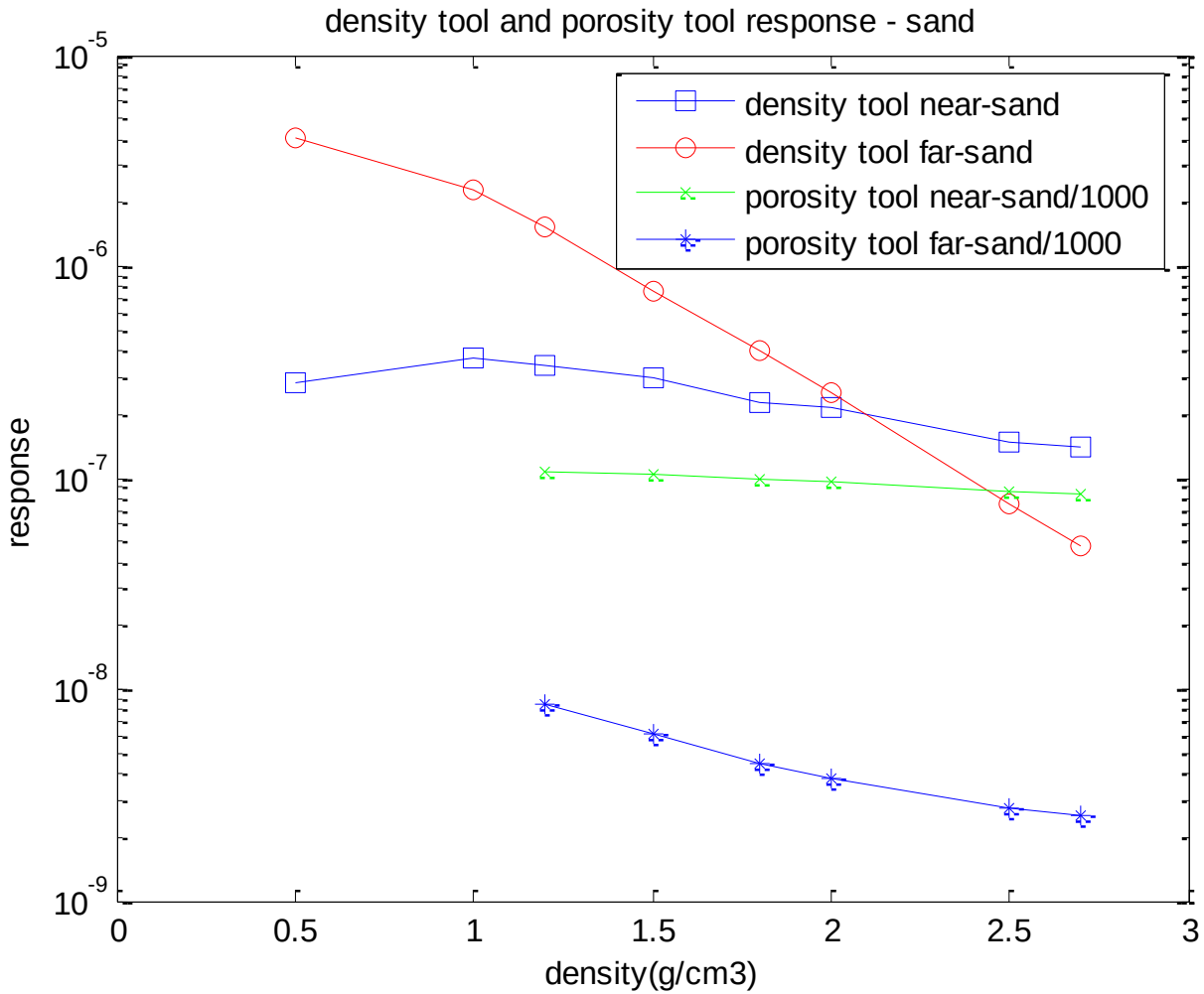
# Porosity Tool



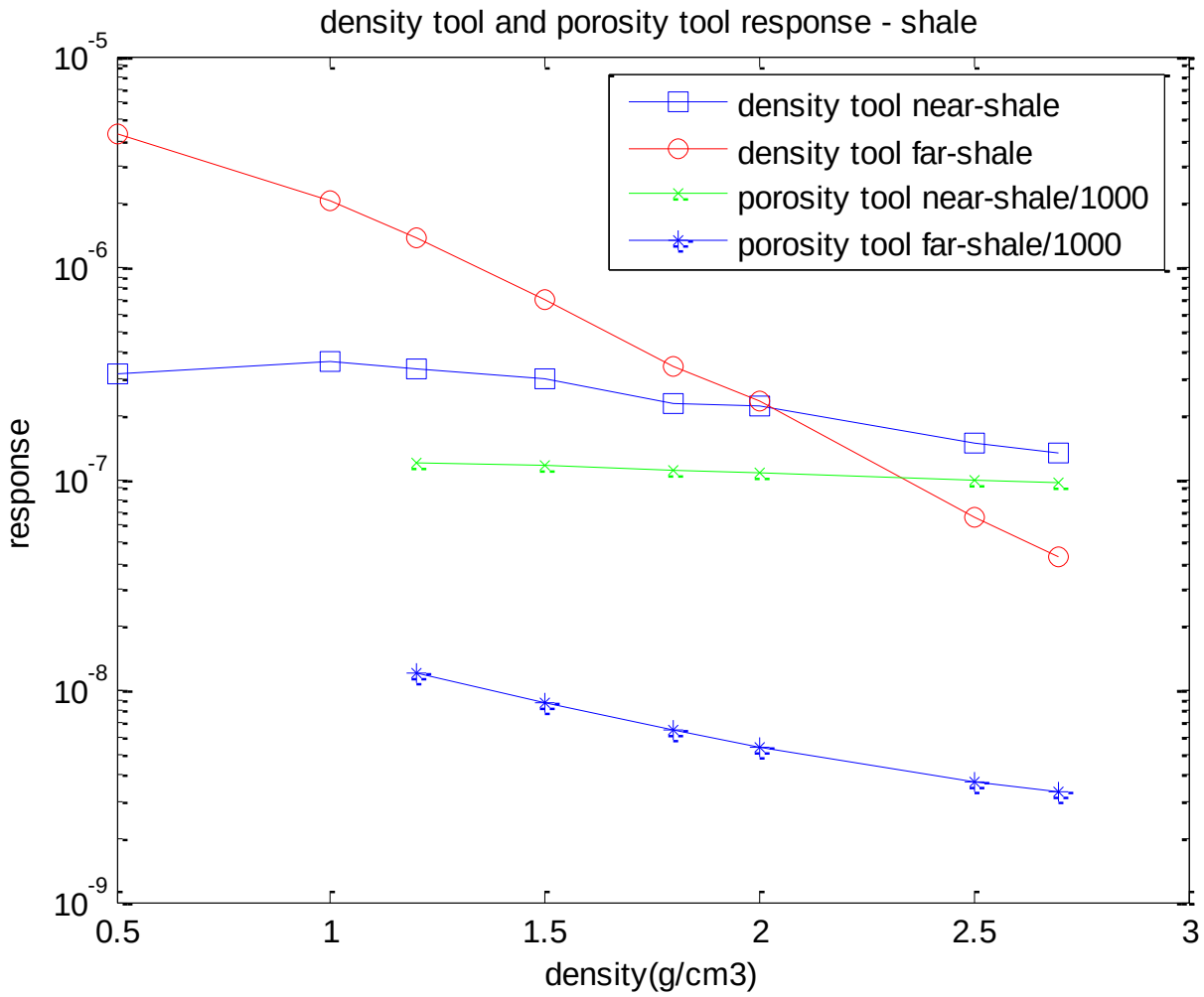
# C/O Tool


17
16
12
13
15
3
1
1.1
5
18
18
14
19
17
16
12
13
15
10
11
3
1
3
11
15
20
18
18
20
14
19

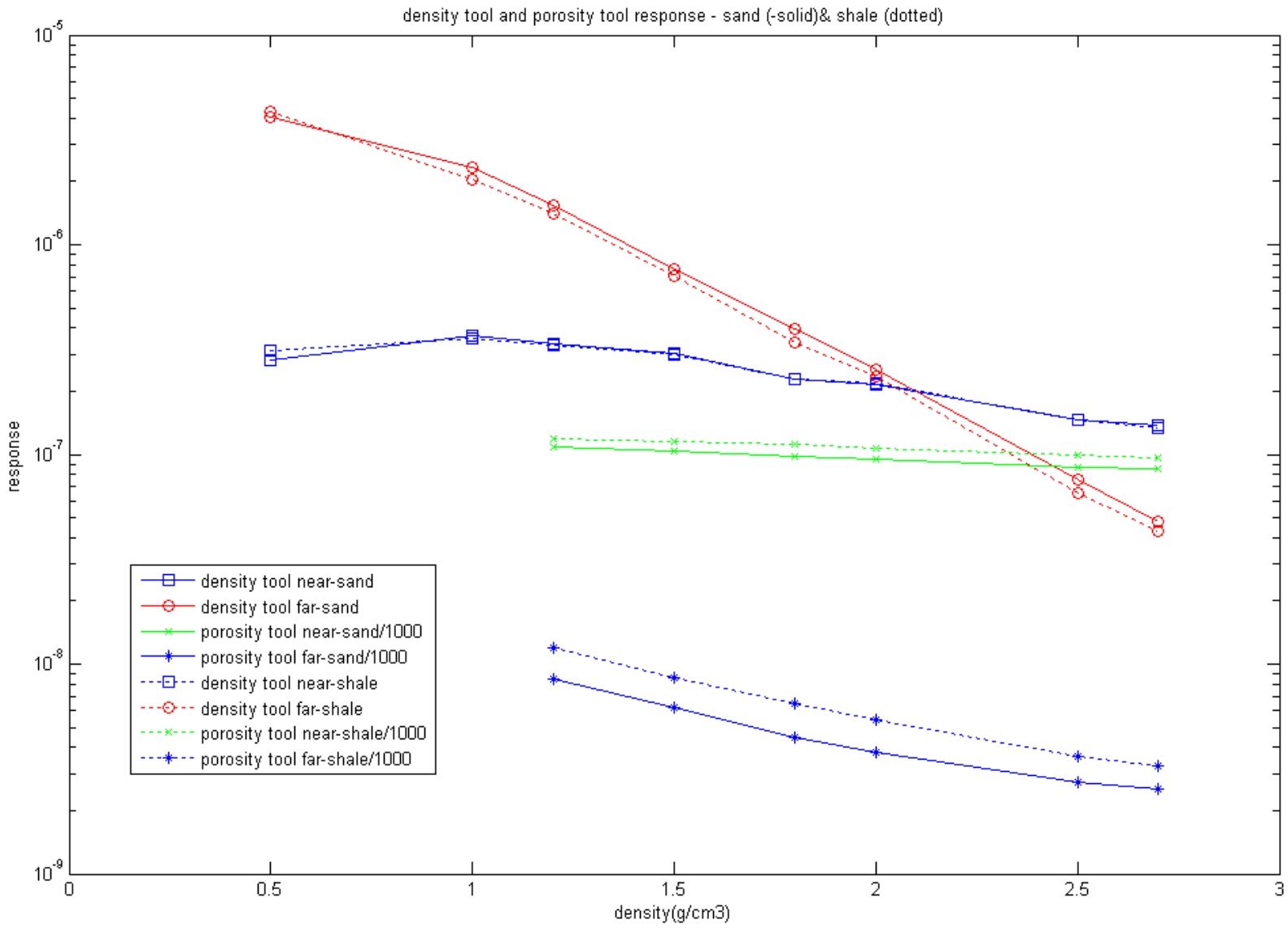
# Density & Porosity Tool Results - Sand



# Density & Porosity Tool Results - Shale



# Density & Porosity Tool



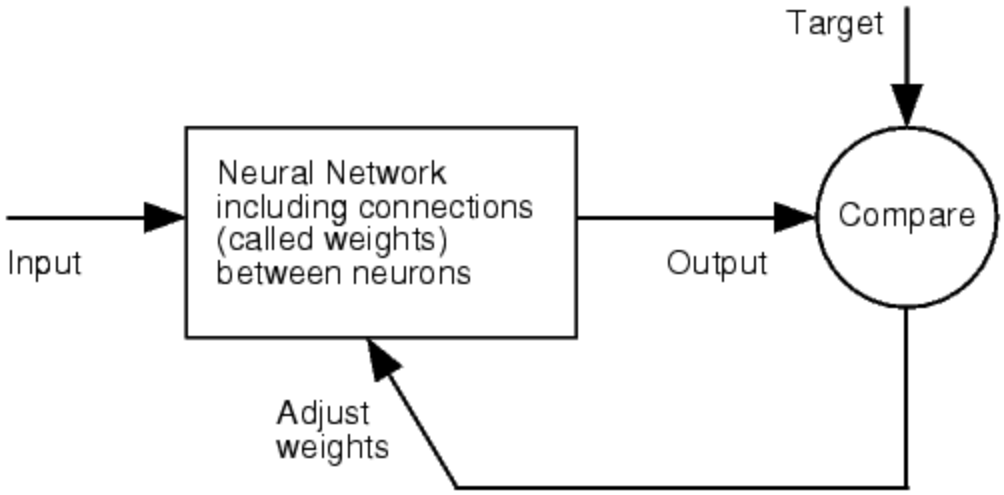
# Neural Network Approach

- Simulation
  - density tool, porosity tool, and C/O tool simulation (maybe) based on one same sample (3 simulations for one sample), and many simulations are needed for neural nets approach.
- Data Compilation
- Neural Network Training
- Neural Network Testing
- Use Neural Network to Predict Density and Porosity Based on Detector Responses

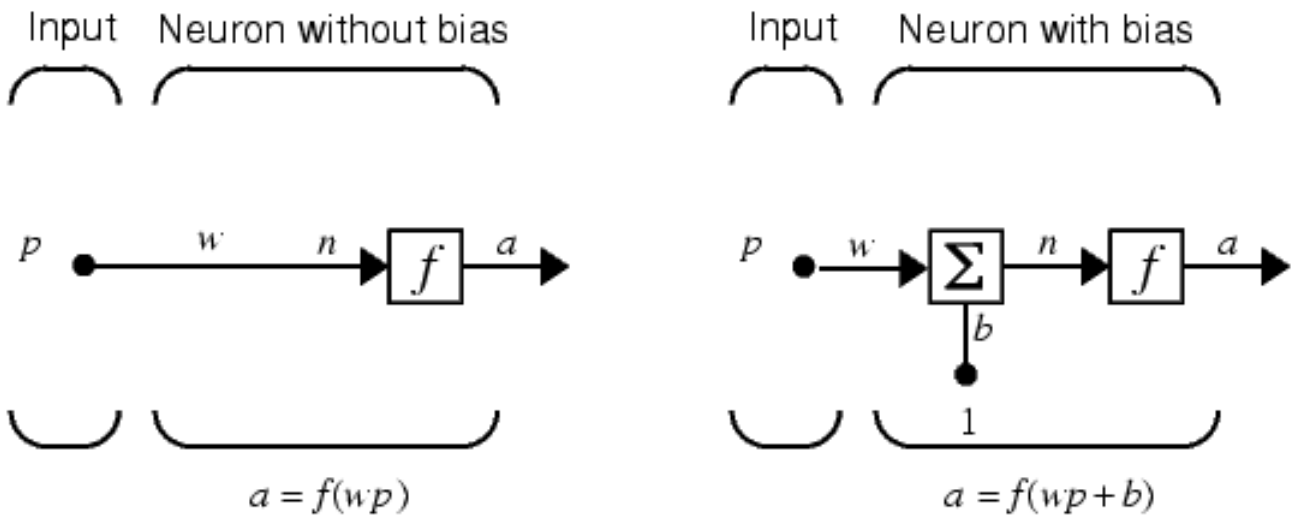
# Introduction of Neural Network

- Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.
- Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown as right. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network.

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems.

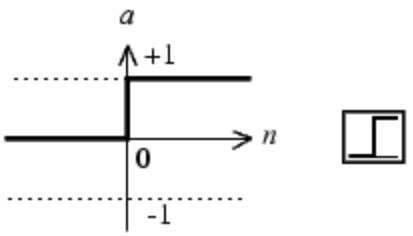


# Neuron Model-Simple Neuron




- w-weight scalar factor, b-bias factor, **f-Transfer Function**
- The central idea of neural networks is that such parameters (w, b) can be adjusted so that the network exhibits some desired or interesting behavior. Thus, we can train the network to do a particular job by adjusting the weight or bias parameters

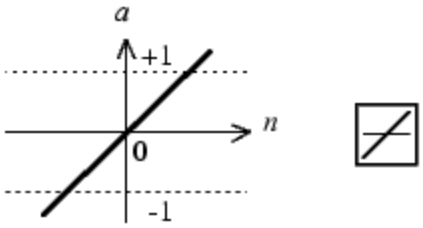
# Transfer Functions




$a = hardlim(n)$

Hard-Limit Transfer Function

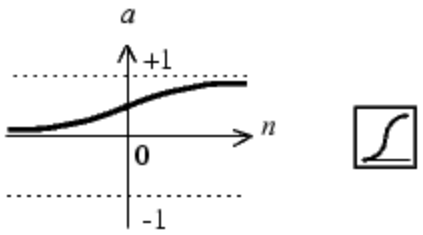
The hard-limit transfer function shown above limits the output of the neuron to either 0, if the net input argument n is less than 0; or 1, if n is greater than or equal to 0.




$a = purelin(n)$

Linear Transfer Function

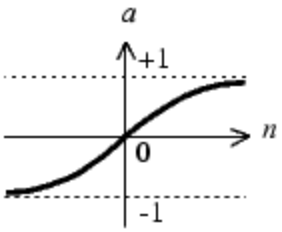
Neurons of this type are used as linear approximators




$a = logsig(n)$

Log-Sigmoid Transfer Function

This transfer function is commonly used in backpropagation networks. **This is important for us.**

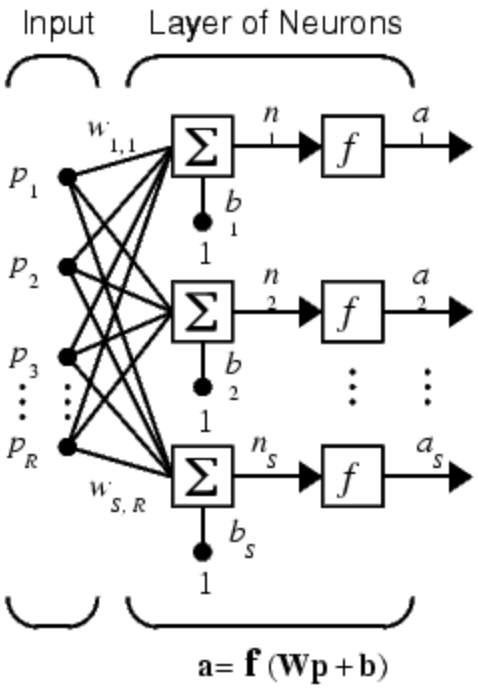



$a = tansig(n)$

Tan-Sigmoid Transfer Function

The function logsig generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity. Alternatively, multilayer networks may use the tan-sigmoid transfer function tansig.

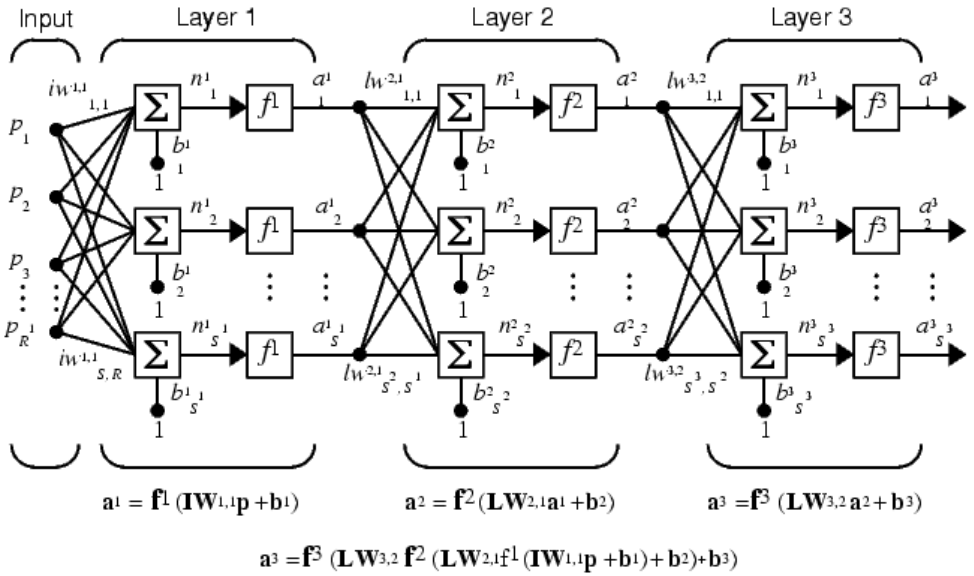
# Network Architectures




Where…

$R$ = number of elements in input vector

$S$ = number of neurons in layer




Two or more of the neurons shown earlier can be combined in a layer, and a particular network could contain one or more such layers.

# Back Propagation Network

- Many kinds of networks are applied in different areas. Back Propagation Network may be the mostly suitable for modeling.
- The term **back propagation** refers to the manner in which the gradient is computed for nonlinear multilayer networks. It is A gradient descent algorithm using the Widrow-Hoff learning rule.
- Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

# Neural Network Training

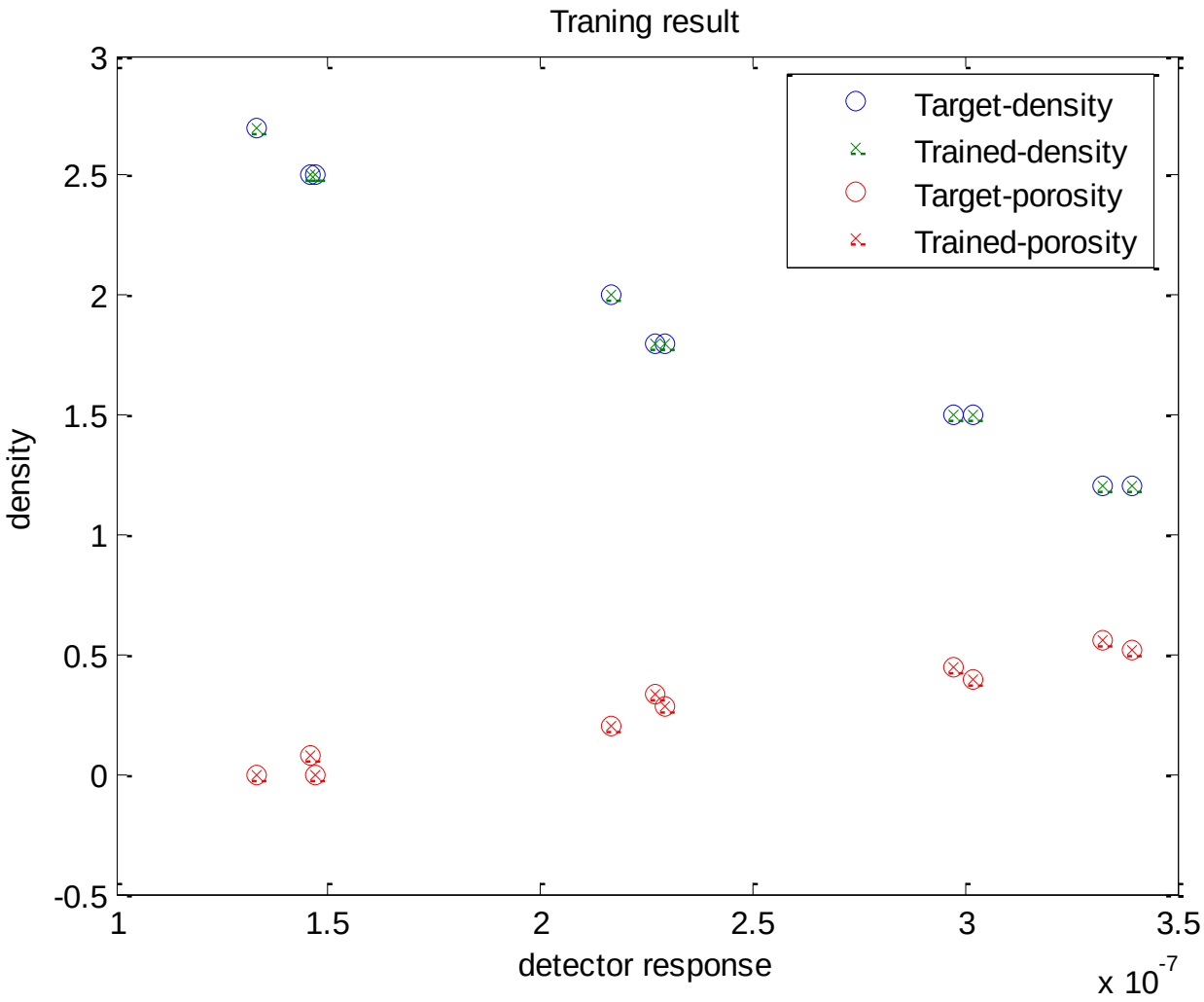
To get neural network's parameters with original data (including inputs and targets). Thus we get a trained network (bias vectors, weight matrix) based on these data.

- Inputs
  - Far detector response from density tool
  - Near detector response from density tool
  - Far detector response from Porosity tool
  - Far detector response from Porosity tool
  - Detector response from C/O tool (maybe)
- Target
  - Density
  - Porosity

# Input Data

| Porosity | Density | Near-Den | Far-Den | Near-Poro | Far-Poro |
|---|---|---|---|---|---|
| 5.2000000e-001 | 1.2000000e+000 | 3.3906163e-007 | 1.5495447e-006 | 1.0839535e-004 | 8.5471604e-006 |
| 4.0000000e-001 | 1.5000000e+000 | 3.0164148e-007 | 7.6693801e-007 | 1.0324025e-004 | 6.1618368e-006 |
| 2.8000000e-001 | 1.8000000e+000 | 2.2939156e-007 | 3.9537953e-007 | 9.7858044e-005 | 4.4470369e-006 |
| 2.0000000e-001 | 2.0000000e+000 | 2.1636299e-007 | 2.5471973e-007 | 9.5131967e-005 | 3.7919546e-006 |
| 0.0000000e+000 | 2.5000000e+000 | 1.4693425e-007 | 7.5410886e-008 | 8.7207477e-005 | 2.7356368e-006 |
| 5.5555556e-001 | 1.2000000e+000 | 3.3221054e-007 | 1.4001839e-006 | 1.1931586e-004 | 1.2015867e-005 |
| 4.4444444e-001 | 1.5000000e+000 | 2.9750385e-007 | 7.0829545e-007 | 1.1603292e-004 | 8.6565714e-006 |
| 3.3333333e-001 | 1.8000000e+000 | 2.2714425e-007 | 3.4329207e-007 | 1.1117228e-004 | 6.4575278e-006 |
| 7.4074074e-002 | 2.5000000e+000 | 1.4603313e-007 | 6.5376099e-008 | 9.8815093e-005 | 3.6533281e-006 |
| 0.0000000e+00 | 2.7000000e+00 | 1.3299174e- | 4.3033387e- | 9.5774024e- | |

# Training Results



# Neural Nets Testing

Original Data Not Participating in the Training:

| Porosity | Density | Near-Den | Far-Den | Near-Poro | Far-Poro |
|---|---|---|---|---|---|
| 0.26 | 2.00 | 2.19E-07 | 2.35E-07 | 1.08E-04 | 5.42E-06 |

Fit results with the neural nets:

| | |
|---|---|
| 0.27787070839307 | 0.26 |
| 1.94964447764871 | 2.00 |

# Conclusion and Future Work

- The preliminary results shows that neural network can work for simultaneous modeling of density and porosity tools
- More data needed for neural nets approach
- Benchmark by experimental data