TẠP CHÍ DẦU KHÍ
Số 7 - 2019, trang 18 - 27
ISSN-0866-854X

ỨNG DỤNG MẠNG NEURON NHÂN TẠO (ANN) TRONG DỰ BÁO ĐỘ RỖNG

Tạ Quốc Dũng[1], Lê Thế Hà[2], Phạm Duy Khang[1]
[1]Trường Đại học Bách khoa - Đại học Quốc gia TP. Hồ Chí Minh
[2]Tập đoàn Dầu khí Việt Nam
Email: tqdung@hcmut.edu.vn; halt01@pvn.vn

Tóm tắt

Nghiên cứu giới thiệu phương pháp dự báo độ rỗng bằng phương pháp truyền thống và sử dụng mạng neuron nhân tạo (Artificial Neural Network - ANN). Phương pháp nội suy truyền thống Kriging sẽ được áp dụng để tìm ra mối quan hệ trong không gian của thông số độ rỗng thông qua các mô hình 2D. Nghiên cứu cũng ứng dụng công cụ "nnstart" của phần mềm Matlab thông qua các lý thuyết về ANN và áp dụng vào việc dự báo độ rỗng cho giếng nghiên cứu.

Kết quả cho thấy phương pháp sử dụng ANN đã giúp tối ưu công tác dự báo độ rỗng cho một giếng khoan từ tài liệu địa cơ học cho trước.

Từ khóa: Địa thống kê, Variogram, nội suy Kriging, mạng neuron nhân tạo.

1. Giới thiệu

Độ rỗng là thông số quan trọng trong việc mô hình hóa đặc trưng thành hệ, có ảnh hưởng lớn đến tính toán trữ lượng và quyết định sự phát triển của một mỏ dầu hoặc khí.

Mục đích của nghiên cứu này là sử dụng phương pháp truyền thống địa thống kê Kriging trong các nghiên cứu thông số độ rỗng đồng thời so sánh với các kết quả tính toán sử dụng ANN.

2. Phương pháp địa thống kê

2.1. Variogram

Variogram được định nghĩa là một nửa kỳ vọng của biến ngẫu nhiên $[Z_x - Z_{x+h}]^2$ và theo công thức thực nghiệm [1, 2]:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z_x - Z_{x+h}]^2 \quad (1)$$

Trong đó:

γ(h): Hàm variogram theo khoảng cách h;

N(h): Số lượng cặp điểm tính toán;

Z_x: Biến ngẫu nhiên x;

$Z_{(x+h)}$: Biến ngẫu nhiên cách x 1 đoạn h.

Ngày nhận bài: 19/3/2019. Ngày phản biện đánh giá và sửa chữa: 19/3 - 3/6/2019.
Ngày bài báo được duyệt đăng: 4/7/2019.

Variogram là công cụ để định lượng tính ổn định/liên tục hoặc sự tương quan không gian của đối tượng nghiên cứu bằng cách nghiên cứu các giá trị bình phương trung bình của hiệu giữa 2 giá trị cách nhau một khoảng cách "h" theo một hướng xác định.

2.2. Covariance

Nếu 2 biến ngẫu nhiên Z_x và Z_{x+h} cách nhau một đoạn "h" có phương sai, cũng có một covariance và được diễn đạt theo công thức thực nghiệm sau [1]:

$$C(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} \{[Z_x - m][Z_{x+h} - m]\} \quad (2)$$

Với m là kỳ vọng toán học của hàm.

2.3. Phương pháp nội suy Kriging

Kriging là nhóm phương pháp địa thống kê dùng để nội suy số liệu của một trường ngẫu nhiên tại một điểm chưa biết giá trị (không lấy được mẫu phân tích) từ những giá trị đã biết ở các điểm lân cận. Tính chất của Kriging là chất lượng của mẫu tốt thì giá trị xác định sẽ tốt. Kriging nội suy dựa trên quy luật BLUE - Best Linear Unbiased Estimator.

Và để nội suy được các điểm, cần giải hệ phương trình sau [3]:

$$Z_{x0} - m_0 = \sum_{i=0}^{n} \lambda_i [Z_{xi} - m_i] \quad (3)$$

$$\sum_{i=1}^{n}[\lambda_i] = 1 \quad (4)$$

Thông thường, chuyển hệ phương trình trên thành một ma trận và giải ma trận đó [3]:

$$K\lambda_i = k \quad (5)$$

Trong đó:

K: Ma trận covariance giữa các điểm dữ liệu với các thành phần $K_{ij} = C(Z_{xi} - T_{xj})$;

k: Vector covariance giữa điểm dữ liệu và điểm cần xác định với $k_i = C(Z_{xi} - Z_{x0})$;

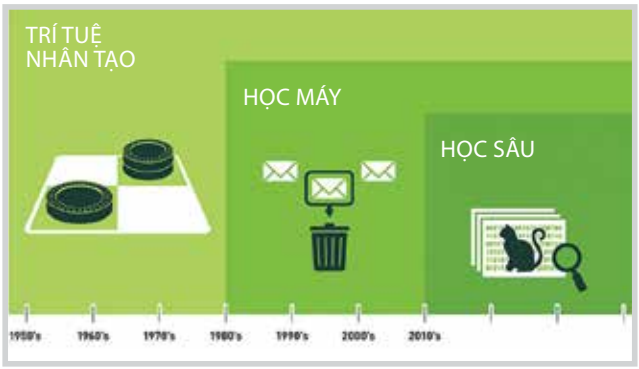
λ_i: Vector trọng số Kriging cho các dữ liệu xung quanh.

3. Trí tuệ nhân tạo và ANN

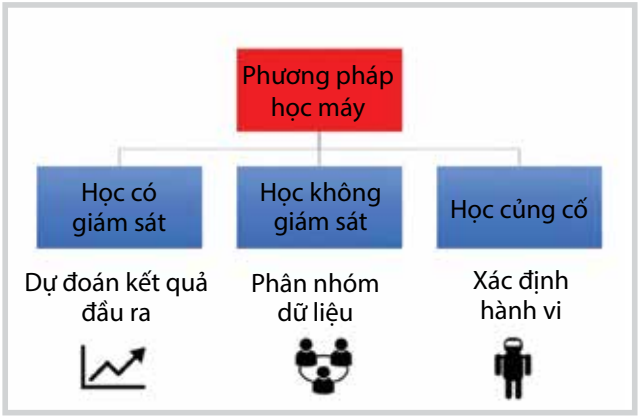
ANN ra đời xuất phát từ ý tưởng mô phỏng bộ não con người. Giống như con người, ANN được học bởi kinh nghiệm, lưu những kinh nghiệm đó và sử dụng trong tình huống phù hợp (Hình 1, 2).

- Học có giám sát (supervised learning)

Học có giám sát là nhóm thuật toán dự đoán đầu ra (output) của dữ liệu mới (new input) dựa trên các cặp dữ liệu đã biết trước. Cặp dữ liệu này còn được gọi là dữ liệu - nhãn (data - label). Đây là nhóm phổ biến nhất trong các thuật toán học máy.



Hình 1. *Lịch sử phát triển của trí tuệ nhân tạo [4, 5]*



Hình 2. *Các phương pháp học của mạng neuron [4]*

Theo toán học, học có giám sát là khi có một tập hợp "n" biến đầu vào $X = \{x_1, x_2, \ldots, x_n\}$ và một tập hợp "n" nhãn tương ứng $Y = \{y_1, y_2, \ldots, y_n\}$. Các cặp dữ liệu biết trước (x_i, y_i) được gọi là tập dữ liệu huấn luyện (training data). Từ tập huấn luyện này cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y.

$$\hat{y} \approx f(x_i) \quad (6)$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu x mới có thể suy ra được nhãn y tương ứng từ hàm số $\hat{y}$.

Nhóm thuật toán học có giám sát gồm các bài toán chính sau:

Phân loại (classification): Các nhãn của dữ liệu đầu vào được chia thành các nhóm hữu hạn.

Hồi quy (regression): Nhãn là một giá trị thực cụ thể. Ở nghiên cứu này, nhóm tác giả đã áp dụng bài toán hồi quy để dự báo phân bố độ rỗng của vỉa.

- Học không giám sát (unsupervised learning)

Trong thuật toán này không biết trước được đầu ra hay nhãn của tập dữ liệu đầu vào, chỉ dựa vào cấu trúc của dữ liệu để thực hiện công việc như: phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Học không giám sát là khi chỉ có dữ liệu đầu vào X mà không biết nhãn Y tương ứng.

- Học bán giám sát (semi - supervised learning)

Các bài toán khi có một lượng lớn dữ liệu X nhưng chỉ có một phần được gán nhãn được gọi là học bán giám sát. Những bài toán thuộc nhóm này nằm giữa 2 nhóm trên.

- Học củng cố (reinforcement learning)

Học củng cố giúp hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximising the performance).

3.1. Thuật toán Gradient descent và tốc độ học

Gradient descent (GD) là một thuật toán tối ưu dùng để tìm cực tiểu của hàm số. Thuật toán sẽ khởi tạo một điểm ngẫu nhiên trên hàm số và sau đó điểm này sẽ được di chuyển theo chiều giảm của đạo hàm cho đến khi đạt

đến điểm cực tiểu. Thông thường, GD sẽ được dùng để cập nhật các trọng số (weights) và hệ số bias cho từng lớp thông qua phương trình:

$$w_{new} = w_{old} - \alpha \frac{\partial J}{\partial w_{old}} \quad (7)$$

$$b_{new} = b_{old} - \alpha \frac{\partial J}{\partial b_{old}} \quad (8)$$

Trong đó:

w_{new}: Trọng số đã được cập nhật;

w_{old}: Trọng số chưa được cập nhật;

b_{new}: Hệ số bias đã được cập nhật;

b_{old}: Hệ số bias chưa được cập nhật;

α: Tốc độ học tập;

$\frac{\partial J}{\partial w_{old}}$: Đạo hàm của hàm mất mát theo trọng số cũ;

$\frac{\partial J}{\partial b_{old}}$: Đạo hàm của hàm mất mát theo hệ số bias cũ.

Tốc độ học tập là tham số quan trọng (hyper parameter), được dùng để kiểm soát số lượng vòng lặp trong quá trình Gradient descent. Khi tham số này nhỏ, thuật toán sẽ cần nhiều bước lặp để hàm số có thể đạt tới điểm cực tiểu. Ngược lại, nếu tham số này lớn, thuật toán sẽ cần ít vòng lặp hơn, tuy nhiên khi đó, có thể hàm số sẽ bỏ qua điểm cực tiểu và không thể hội tụ được (Hình 3).

3.2. Hàm truyền

Hàm kích hoạt hay còn gọi là hàm truyền có chức năng chuyển đổi thông số đầu vào sang một khoảng giá trị khác. Mạng cần có các hàm truyền để quyết định có nên truyền tiếp dữ liệu hay không và truyền với cường độ bao nhiêu. Hàm truyền bao gồm hàm tuyến tính và hàm phi tuyến. Các hàm phi tuyến giúp mô hình dễ dàng khái quát hóa và thích hợp với nhiều loại dữ liệu.

- Hàm tuyến tính (Purelin)

Công thức của hàm:

$$f(x) = x \quad (9)$$

Đạo hàm của hàm tuyến tính:

$$\frac{df}{dx} = 1 \quad (10)$$

Nhận xét: Giá trị đầu ra không bị giới hạn trong một khoảng cụ thể mà chỉ phụ thuộc vào miền giá trị của thông số đầu vào khi đi qua hàm.

Do bài toán giải quyết của nghiên cứu là xây dựng ANN để tính toán và dự báo giá trị độ rỗng nên hàm hoạt động ở lớp đầu ra chính là hàm tuyến tính.

- Hàm sigmoid

Đây là hàm thông dụng thường được dùng trong ANN đa lớp. Hàm có công thức như sau:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

Đạo hàm của hàm sigmoid:

$$\frac{df}{dx} = f(x)[1 - f(x)] \quad (12)$$

Nhận xét: Đồ thị hàm sigmoid cho giá trị đầu ra từ 0 đến 1 khi giá trị neuron đầu vào đi từ -∞ đến +∞ (Hình 4).

- Hàm tanh

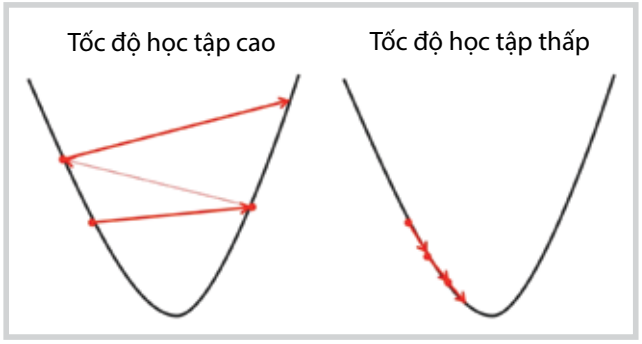
Giống với hàm sigmoid, hàm tanh cũng được sử dụng nhiều trong mạng neuron đa lớp. Công thức của hàm có dạng:

$$f(x) = tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (13)$$

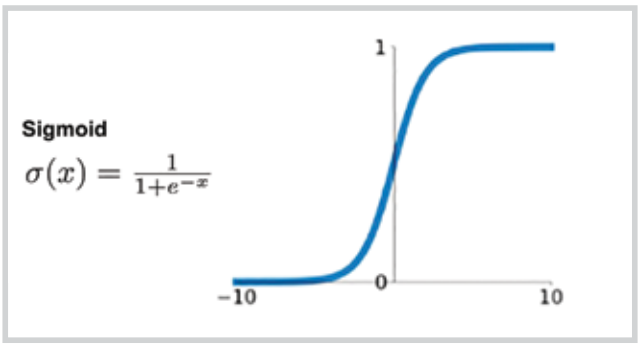
Đạo hàm của hàm tanh:

$$\frac{df}{dx} = 1 - f^{2}(x) \quad (14)$$

Nhận xét: Dựa vào đồ thị, giá trị của hàm đi từ -1 đến 1 ứng với giá trị đầu vào đi từ -∞ đến +∞ (Hình 5).



Hình 3. *Các trường hợp của tốc độ học tập [6]*



Hình 4. *Đồ thị biểu diễn hàm sigmoid [7]*

- Hàm ReLU (Rectified Linear Unit)

Hàm ReLU là hàm kích hoạt phổ biến nhất vì tính đa dụng trong mạng neuron tích chập (convolutional neural network) và học sâu.

Công thức hàm có dạng:

$$f(x) = max(0, z) \quad (15)$$

Đạo hàm của hàm ReLU:

$$\frac{df}{dx} = \begin{cases} 0, nếu\ x < 0 \\ 1, nếu\ x \geq 0 \end{cases} \quad (16)$$

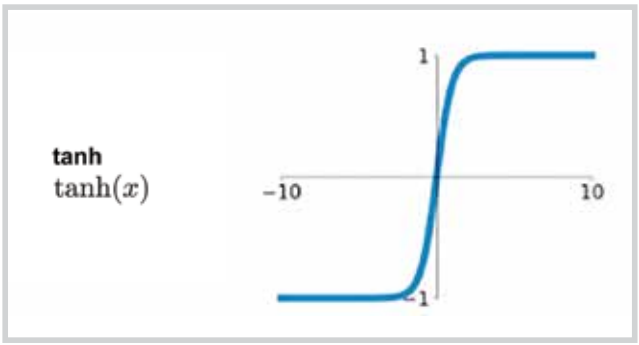
Dựa vào Hình 6a, giá trị của hàm đi từ 0 đến $+\infty$ khi giá trị đầu vào lớn hơn hoặc bằng 0.

Khi đầu vào là giá trị âm, hàm sẽ bằng 0, điều này sẽ làm giảm khả năng phù hợp dữ liệu của mô hình và ảnh hưởng đến quá trình huấn luyện dữ liệu. Do đó, hàm Leaky ReLU (Hình 6b) được hình thành để giải quyết vấn đề trên.

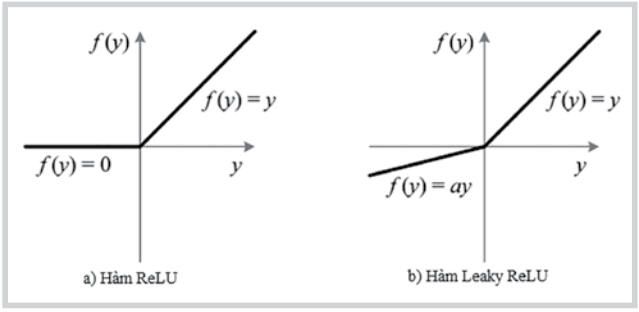
3.3. Cấu tạo và nguyên lý hoạt động của mạng neuron

Nhóm tác giả giới thiệu và sử dụng mạng neuron thông thường (Regular Neural Network) để thực hiện vì phương thức tính toán đơn giản, dễ tiếp cận.

Một ANN thường tổ chức các neuron thành từng lớp và mỗi lớp chịu trách nhiệm cho một công việc cụ thể. ANN thường có 3 lớp: lớp nhập hay lớp đầu vào, lớp ẩn và lớp xuất.



Hình 5. *Đồ thị biểu diễn hàm tanh [7]*



Hình 6. *Đồ thị biểu diễn hàm ReLU, Leaky ReLU [8]*

- Lớp nhập (input layer) cung cấp cho mạng các số liệu cần thiết. Số lượng neuron trong lớp nhập tương ứng với số lượng thông số đầu vào được cung cấp cho mạng và các thông số đầu vào này được giả thiết ở dạng vector.

- Lớp ẩn (hidden layer) chứa các neuron ẩn giúp kết nối giá trị đầu vào đến giá trị đầu ra. Một mạng neuron có thể có một hoặc nhiều lớp ẩn chịu trách nhiệm chính cho việc xử lý các neuron của lớp nhập và đưa các thông tin đến neuron của lớp xuất. Các neuron này thích ứng với việc phân loại và nhận diện mối liên hệ giữa thông số đầu vào và thông số đầu ra.

- Lớp xuất (output layer) chứa các neuron đầu ra nhằm chuyển thông tin đầu ra của các tính toán từ ANN đến người dùng. Một ANN có thể được xây dựng để có nhiều thông số đầu ra.

Số neuron của lớp nhập và lớp xuất sẽ do bài toán quyết định, số neuron lớp ẩn và số lớp ẩn sẽ do người nhập quyết định. Tuy nhiên, việc chọn loại và số lượng của thông số đầu vào có ảnh hưởng lớp đến chất lượng của mạng.

Mô hình toán học của ANN lan truyền thẳng được trình bày như sau:

$$y(x) = f\left(\sum_{i=1}^{n} w_i x_i\right) \quad (17)$$

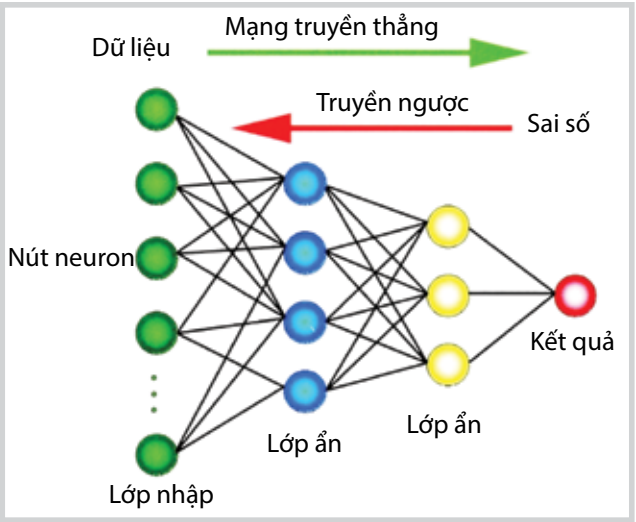
Trong đó:

y(x): Giá trị đầu ra theo biến x;

f: Hàm kích hoạt hay hàm truyền;

w_i: Trọng số liên kết của neuron x_i;

x_i: Các giá trị đầu vào.



Hình 7. *Mô hình cơ chế hoạt động của ANN [7]*

Bản chất nguyên lý hoạt động của ANN truyền thẳng chính là quá trình huấn luyện mạng (training). Cụ thể, quá trình huấn luyện thường sử dụng giải thuật lan truyền ngược để tìm đạo hàm cho từng tham số trong mạng [1, 5, 9].

Giai đoạn lan truyền thẳng [9]:

Bước 1: Vector thông số đầu vào được nhập vào các neuron ở lớp nhập.

$$a^{(0)} = x \tag{18}$$

Bước 2: Tại neuron lớp ẩn thứ j, giá trị tín hiệu nhận từ lớp nhập sẽ được tính tổng trọng số hóa của tất cả các dữ liệu được nhập bằng cách cộng tất cả tích của mỗi dữ liệu đầu vào và trọng số liên kết giữa lớp ẩn và lớp nhập.

$$z_{inj} = b_{oj} + \sum_{i=1}^{n} x_i v_{ij} \tag{19}$$

Bước 3: Sau đó, hàm kích hoạt (hàm truyền) sẽ được sử dụng để chuyển giá trị được nhận thành giá trị đầu ra.

$$z_j = f(z_{inj}) \tag{20}$$

Tiếp theo, giá trị đầu ra tại neuron lớp ẩn j tiếp tục được truyền đến neuron lớp xuất k giống với phương thức từ lớp nhập đến lớp ẩn.

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk} \tag{21}$$

Sau đó, hàm truyền lại được sử dụng để tính giá trị đầu ra của neuron tại lớp xuất.

$$y_k = f(y_{ink}) \tag{22}$$

Lúc này, giai đoạn lan truyền thẳng đến đây kết thúc, mạng sẽ chuyển đến giai đoạn lan truyền ngược.

Bước 4: Trong giai đoạn nhập, số liệu nhập gồm cả số liệu đầu vào và giá trị thực tế. Từ đó, với mỗi bộ số liệu tính được từng sai số đầu ra tương ứng, giá trị này được gọi là hàm mất mát (Cost Function - J).

$$J = t_k - y_k \tag{23}$$

Bước 5: Từ hàm cost function vừa tìm được, tính đạo hàm của hàm này theo trọng số giữa lớp ẩn - lớp ra và trọng số giữa lớp nhập - lớp ẩn.

$$\Delta w_{jk} = \frac{\partial J}{\partial w_{jk}} \tag{24}$$

$$\Delta v_{ij} = \frac{\partial J}{\partial v_{ij}} \tag{25}$$

Bước 6: Kế tiếp, giá trị trọng số liên kết giữa lớp ẩn và lớp xuất cũng như giá trị trọng số liên kết giữa lớp nhập và lớp ẩn được hiệu chỉnh lại đồng thời.

$$w_{jk}(new) = w_{jk}(old) + \alpha \Delta w_{jk} \tag{26}$$

$$v_{ij}(new) = v_{ij}(old) + \alpha \Delta v_{ij} \tag{27}$$

3.4. Hiện tượng Overfitting

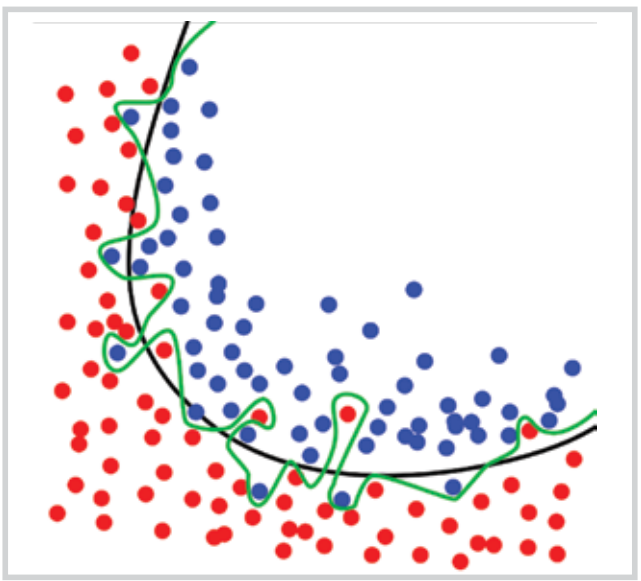
Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu huấn luyện. Việc này sẽ gây ra hậu quả lớn nếu trong tập dữ liệu huấn luyện có nhiễu. Khi đó, mô hình không thực sự mô tả tốt dữ liệu ngoài tập huấn luyện. Overfitting đặc biệt xảy ra khi lượng dữ liệu huấn luyện quá nhỏ hoặc độ phức tạp của mô hình quá cao.

Một mô hình được coi là tốt (fit) nếu cả training error và test error đều thấp. Nếu training error thấp nhưng test error cao, mô hình bị overfitting. Nếu training error cao và test error cao, mô hình bị underfitting, còn đối với việc training error cao - test error thấp thì xác suất xảy ra rất nhỏ. Để có được mô hình tốt, cần tránh hiện tượng overfitting thông qua kỹ thuật sau:

- Validation là kỹ thuật lấy từ tập huấn luyện (training data set) ra một tập con nhỏ và thực hiện việc đánh giá mô hình trên tập con này. Tập con này được gọi là tập validation và tập huấn luyện mới của mô hình là phần còn lại của tập huấn luyện ban đầu.

- Regularisation là kỹ thuật làm thay đổi mô hình một ít, giảm độ phức tạp của mô hình, từ đó tránh được hiện tượng overfitting.

Ở bài báo này, nhóm tác giả sẽ kiểm tra hiện tượng overfitting qua kỹ thuật validation.



Hình 8. *Tập dữ liệu thể hiện hiện tượng overfitting (đường màu xanh lá) [10]*

4. Kết quả tính toán độ rỗng

4.1. Kết quả tính toán từ phần mềm Gs+

Các số liệu độ rỗng theo quỹ đạo thu thập từ 3 giếng X1, X5, X9. Trong nghiên cứu này số liệu của 3 giếng offset chọn cùng trong một đối tượng để đảm bảo bộ số liệu ổn định dừng bậc 2 phục vụ cho tính toán địa thống kê. Các dữ liệu khác như bản đồ tướng và thuộc tính địa chấn không nằm trong phạm vi của nghiên cứu này. Các số liệu lựa chọn đã được kiểm tra từ các nghiên cứu tài liệu địa vật lý.

Từ các thông số trong Bảng 1, nhập vào Gs+ và thu được quỹ đạo 2D của 3 giếng theo từng phương riêng biệt Tây - Đông (Hình 9).

Biểu đồ Variogram theo độ rỗng trên được xây dựng theo mô hình hàm mũ (Exponential) và có hệ số tương quan là 0,772.

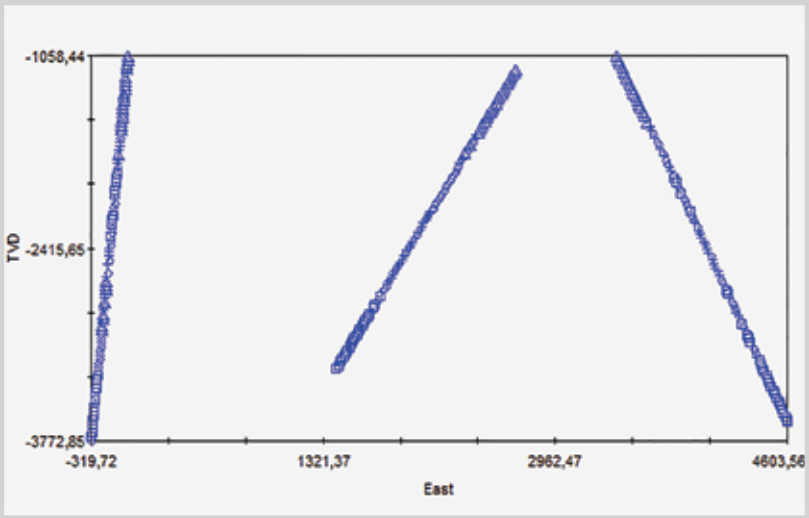
Sau khi tìm được mô hình Variogram cho thông số độ rỗng, nhóm tác giả tiến hành bước kiểm tra chéo (Cross-validate) để đánh giá độ chính xác trước khi đưa vào tính toán. Để có được mô hình Variogram tốt nhất, có thể loại bỏ hoặc chỉnh sửa các giá trị ngoại lai (do sai số trong quá trình đo đạc). Với mô hình Variogram trên, nhóm tác giả đã tìm được hệ số hồi quy cho mô hình (Hình 10). Kết quả cho ra tốt (0,968), do đó mô hình trên sẽ được dùng để nội suy độ rỗng.

Tiến hành nội suy độ rỗng theo quỹ đạo giếng X11 từ thông số độ rỗng của các giếng lân cận trên thông qua mô hình xây dựng được.

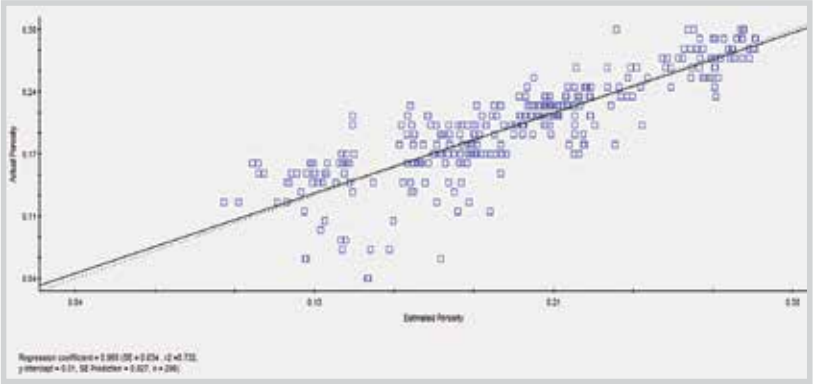
Hình 12 cho thấy độ chính xác của thuật toán Kriging tương đối chính xác, với hệ số hồi quy của dữ liệu dự báo là trên 60%. Nguyên nhân dẫn đến hệ số hồi quy này chỉ đạt 60% là do chỉ áp dụng thuật toán Kriging 1 thông số. Nếu muốn tăng độ chính xác của thuật toán lên có thể áp dụng Cokriging, từ đó có thông số phụ hỗ trợ cho việc dự đoán thông số chính.

Bảng 1. *Thông số đầu vào của 3 giếng lân cận*

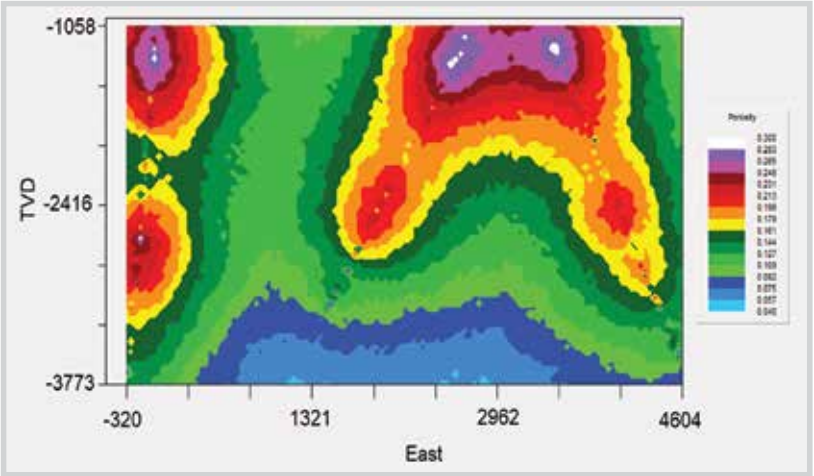
Thông số	Ký hiệu (đơn vị)
Độ sâu theo phương thẳng đứng	TVD (m)
Tọa độ theo phương Đông	East (m)
Độ rỗng	Porosity



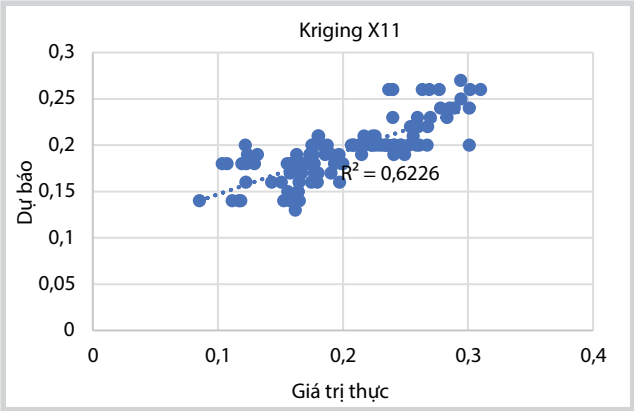
Hình 9. *Mặt cắt giếng khoan X1, X5, X9 theo trục TVD và hướng Đông - Tây*



Hình 10. *Đồ thị cross validation của bộ số liệu*



Hình 11. *Kết quả nội suy độ rỗng khu vực từ 3 giếng X1, X5, X9 theo độ sâu thẳng đứng (từ 1.000 - 3.800m) và theo hướng Tây - Đông*



Hình 12. *Đồ thị biểu diễn độ chính xác giữa số liệu dự báo và số liệu thực từ phương pháp Kriging*

4.2. Kết quả từ ANN

4.2.1. Bước 1: Thu thập và xử lý số liệu

Từ các thông số đầu vào như độ bền nén đơn trục (UCS), áp suất lỗ rỗng (Pore Pressure) và độ sâu theo phương thẳng đứng (TVD), mong muốn dự báo được số liệu đầu ra là độ rỗng tương ứng với các thành phần trên. Các thông số đầu vào này phải có tính ổn định, tính phổ biến và có liên quan mật thiết đến thông số đầu ra.

Ở bộ số liệu thu thập được từ 3 giếng trong khu vực, nhóm tác giả đã chia thành 2 bộ con chính:

- Bộ 1 (Dữ liệu huấn luyện - Training Data) bao gồm 266 giá trị bộ mẫu đầu vào (UCS, Pore Pressure, độ sâu) và 266 giá trị độ rỗng tương ứng. Từ đây, công cụ xây dựng ANN của Matlab sẽ tiếp tục chia thành 3 nhóm nhỏ:

Luyện mạng (training) chiếm 70% bộ số liệu ứng với 186 bộ mẫu đầu vào. Các giá trị này được sử dụng liên tục trong quá trình luyện mạng và mạng neuron sẽ được tinh chỉnh dựa trên sai số mạng.

Kiểm tra chéo (validation) chiếm 15% bộ số liệu ứng với 40 bộ mẫu đầu vào. Chúng dùng để kiểm tra mạng có xảy ra hiện tượng quá khớp hay là không.

Kiểm tra mạng (testing) chiếm 15% bộ số liệu ứng với 40 bộ mẫu đầu vào, kiểm tra mức độ hiệu quả của mạng trong và sau khi luyện mạng.

- Bộ 2 (Dữ liệu kiểm tra mạng - Testing data) gồm 30 giá trị bộ mẫu đầu vào và 30 giá trị độ rỗng tương ứng, xác định độ tin cậy của mạng vừa mới huấn luyện để từ đó có thể dự đoán cho giếng lân cận.

Ngoài ra, để dự báo độ rỗng cho giếng X11, nhóm tác giả đã chuẩn bị 1 bộ số liệu của giếng X11 gồm độ sâu theo phương thẳng đứng, độ bền nén đơn trục, áp suất lỗ rỗng. Thông qua mạng neuron vừa được huấn luyện, nhóm tác giả sẽ suy được độ rỗng tương ứng.

4.2.2. Bước 2: Xây dựng mạng

Với yêu cầu bài toán là dự đoán độ rỗng từ tài liệu địa cơ học, vì thế lớp đầu vào chứa các giá trị địa cơ học thu thập được và lớp đầu ra chứa giá trị độ rỗng từ mạng. Sau đó, tiến hành quá trình xây dựng mạng hay chọn cấu trúc mạng thông qua việc chọn số lớp và số neuron ẩn trong từng lớp cho mạng.

Thông thường, việc thiết kế ANN sẽ bắt đầu với một lớp ẩn. Số neuron trong lớp ẩn đó sẽ được điều chỉnh tăng dần cho đến khi đạt được kết quả sai số đầu ra của mạng và giá trị đầu ra mong muốn là chấp nhận được. Nếu số neuron quá lớn (hơn 50) mà sai số vẫn chưa chấp nhận được thì tăng lớp ẩn thành 2. Quá trình này được lặp đi lặp lại cho đến khi đạt được sai số và đầu ra mong muốn.

Trong bài báo này, nhóm tác giả sẽ xây dựng mạng với số lớp ẩn là 1 và số neuron của lớp ẩn này lần lượt là 10, 20.

4.2.3. Bước 3: Luyện mạng

Với cấu trúc mạng được chọn ở bước 2, tiếp tục tiến hành bước huấn luyện mạng. Thực chất, quá trình huấn luyện mạng chính là quá trình điều chỉnh trọng số liên kết (weights). Các giá trị trọng số liên kết này sẽ được mặc định ngẫu nhiên khi bắt đầu xây dựng mạng, sau đó, trong suốt quá trình luyện mạng, các thuật toán của mạng sẽ điều chỉnh các giá trị trên.

Kết quả quá trình luyện mạng sẽ hiển thị sai số toàn phương trung bình (MSE) và hệ số tương quan (R) của 3 bộ số liệu nhỏ được chia từ bộ 1, đó là sai số bộ số liệu luyện mạng, sai số kiểm tra chéo và sai số kiểm tra mạng. Công thức của MSE được trình bày như sau:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^*)^2}{n} \quad (28)$$

Trong đó:

y_i: Giá trị dự báo;

y_i^*: Giá trị thực đo từ mẫu;

n: Số mẫu.

4.2.4. Bước 4: Kiểm tra độ chính xác của mạng

Để tránh hiện tượng quá khớp và đánh giá mạng được luyện ở bước 3 phải kiểm tra độ chính xác của mạng.

Thực tế, mạng sau khi được huấn luyện sẽ sử dụng phần số liệu kiểm tra mạng để kiểm tra mức độ hiệu quả

của mạng. Tiếp đến, mạng sẽ sử dụng dữ liệu trung gian hay số liệu kiểm tra chéo để tính toán sai số nhằm đảm bảo hiện tượng quá khớp không xảy ra.

Việc quan sát đồ thị thể hiện (Performance) qua xây dựng mạng được dùng để đánh giá mạng và xem xét hiện tượng quá khớp. Các giá trị MSE và R ở bước 3 sẽ hiển thị tại vị trí vòng lặp (Epoch) cho hiệu quả tốt nhất của quá trình xây dựng mạng.

Nếu độ tin cậy của mạng sau khi kiểm tra không đạt kết quả mong muốn, sẽ thực hiện một trong 2 cách sau:

- Tiếp tục luyện lại mạng để có được kết quả tốt hơn.

- Quay lại bước 2, tiến hành điều chỉnh số neuron ở lớp ẩn hoặc cấu trúc mạng, sau đó luyện mạng lại.

Một ANN hoạt động tốt sẽ cho ra các kết quả sau:

- Sai số luyện mạng, kiểm tra chéo, kiểm tra mạng thấp;

- Sai số luyện mạng ở những vòng lặp cuối ổn định;

- Mức độ quá khớp không đáng kể;

- Kiểm tra mạng bằng bộ số liệu khác (bộ 2) cho kết quả tốt.

4.2.5. Bước 5: Sử dụng mạng để dự báo

Sử dụng mạng để dự báo các giá trị độ rỗng cần tìm với bộ số liệu là các thông số cơ học đá đã nhập để xây dựng mạng. Các giá trị được dự báo sẽ được so sánh với giá trị độ rỗng thực của giếng X11. Từ đó, tiến hành tính toán MSE để đưa ra những kết luận về phương pháp ANN.

Với bài toán trên, nhóm tác giả sẽ xây dựng các mạng sau để dự báo độ rỗng:

- Mạng 2-10-1 (mạng A) với các thông số đầu vào gồm áp suất lỗ rỗng và UCS, 10 neuron trong lớp ẩn.

- Mạng 2-20-1 (mạng B) với các thông số đầu vào gồm áp suất lỗ rỗng và UCS, 20 neuron trong lớp ẩn.

- Mạng 3-10-1 (mạng C) với các thông số đầu vào gồm áp suất lỗ rỗng, UCS và độ sâu theo phương thẳng đứng, 10 neuron trong lớp ẩn.

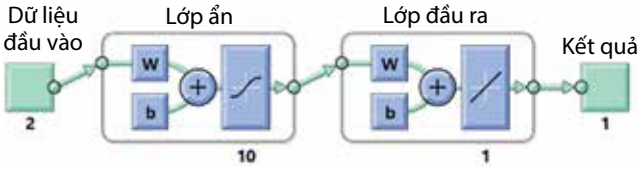
- Mạng 3-20-1 (mạng D) với các thông số đầu vào gồm áp suất lỗ rỗng, UCS và độ sâu theo phương thẳng đứng, 20 neuron trong lớp ẩn.

Đối với mạng A:

Với 2 thông số đầu vào là áp suất lỗ rỗng và độ bền nén đơn trục, nhóm tác giả đã xây dựng được mạng A như Hình 13.

Bảng 2 thể hiện kết quả huấn luyện mạng, kiểm tra chéo và kiểm tra mạng A ở vòng lặp thứ 7 và đồ thị thể hiện quá trình huấn luyện trong 13 vòng lặp.

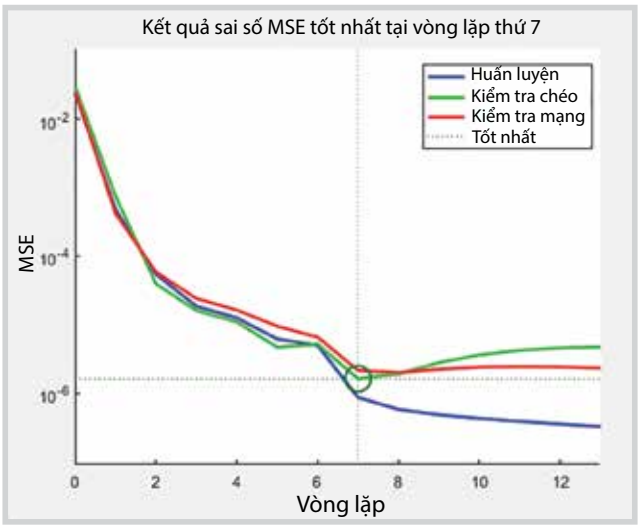
Ngoài ra, nhóm tác giả tiếp tục sử dụng số liệu gồm 30 mẫu ở bộ 2 để đánh giá sự chính xác của mạng vừa được huấn luyện. Kết quả Bảng 3 thể hiện làm tăng độ tin cậy của mạng A.



Hình 13. *Mô hình huấn luyện của mạng A dùng để dự báo độ rỗng*

Bảng 2. *Kết quả huấn luyện ở epoch thứ 7 của mạng A dùng để dự báo độ rỗng*

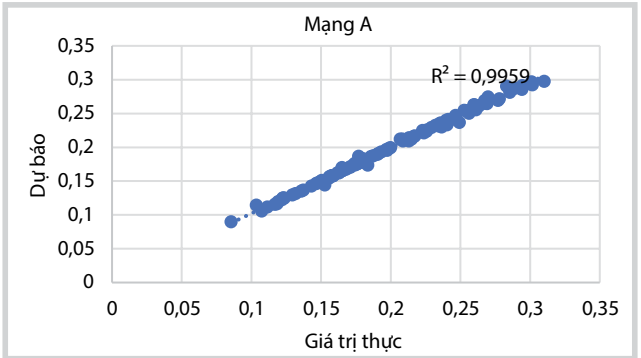
	Mẫu	MSE × 10^{-7}	R
Tập huấn luyện	186	8,60788	0,999836
Tập kiểm tra chéo	40	16,1608	0,999751
Tập kiểm tra mạng	40	21,4578	0,999709



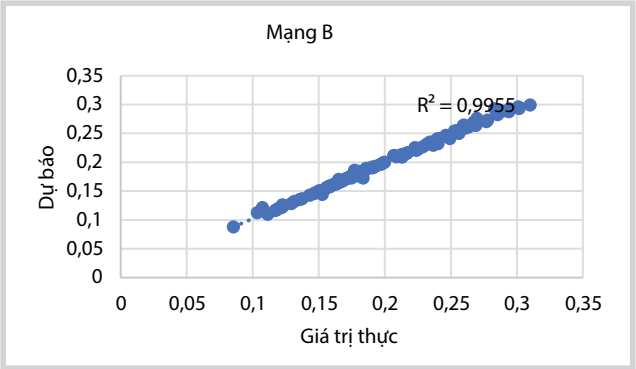
Hình 14. *Đồ thị thể hiện MSE của 3 tập số liệu trong quá trình huấn luyện mạng A*

Bảng 3. *Kết quả đánh giá độ chính xác của mạng A khi dùng bộ số liệu 2*

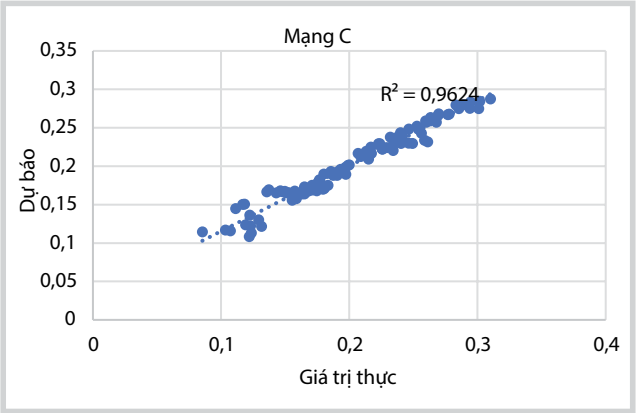
MSE × 10^{-7}	171,899
R	0,998413



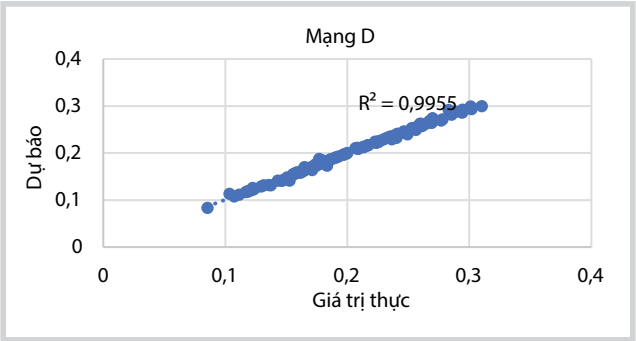
Hình 15. *Kết quả dự báo độ rỗng từ mạng A*



Hình 16. *Kết quả dự báo độ rỗng từ mạng B*



Hình 17. *Kết quả dự báo độ rỗng từ mạng C*



Hình 18. *Kết quả dự báo độ rỗng từ mạng D*

Sau khi dùng mạng A để dự báo độ rỗng từ bộ số liệu X11, nhóm tác giả đã được một bộ số liệu độ rỗng 113 giá trị. Từ đó, nhóm tác giả đã đem so sánh với bộ số liệu độ rỗng thực tế của X11 cũng gồm 113 giá trị.

Với 2 bộ số liệu độ rỗng thực tế và dự báo có được, nhóm tác giả tiến hành đưa 2 thông số này vào Excel và tính toán MSE = 142,533 × 10^{-7} và hệ số hồi quy R = 0,9959.

Tương tự đối với 3 mạng còn lại:

Ở mạng B, nhóm tác giả cũng tiến hành tương tự các bước ở mạng A và thu được kết quả cuối cùng sau khi nhập vào Excel, MSE = 143,171 × 10^{-7} và hệ số hồi quy R = 0,9955.

Kết quả ở mạng C, MSE = 1428,875 × 10^{-7} và hệ số hồi quy R = 0,9624 (Hình 17). Với kết quả vẫn ra hệ số hồi quy lớn (trên 90%) cho thấy mạng neuron đã huấn luyện thành công và cho ra một hàm xấp xỉ gần với quy luật trong thực tế.

Kết quả ở mạng D, MSE = 138,968 × 10^{-7} và hệ số hồi quy R = 0,9955.

5. Kết luận

Mạng A với 2 thông số đầu vào (áp suất lỗ rỗng và độ bền nén đơn trục) được khởi tạo cùng với 10 neuron ở lớp ẩn, sau đó thu được kết quả dự báo rất khả quan với độ chính xác lên đến 99,59% so với số liệu thực tế. Tuy nhiên, ở mạng B, mạng chỉ thay đổi số neuron của lớp ẩn từ 10 lên 20 neuron thì độ chính xác của mạng đạt 99,55. Như vậy, từ mạng A và B cho thấy với 10 neuron lớp ẩn thì chỉ cần 2 thông số đầu vào, mạng đã đạt giá trị dự báo ở mức tối đa.

Đối với mạng C, mạng được thêm thông số độ sâu TVD, kết quả cho thấy khi thêm thông số đầu vào (input) mạng sẽ giảm độ chính xác (đạt 96,24% so với thực tế). Muốn khắc phục hiện tượng trên chỉ cần tăng số lượng neuron ở lớp ẩn lên. Trong nghiên cứu này, nhóm tác giả đã tăng lên 20 để mạng C chuyển đổi thành mạng D.

Phương pháp mạng neuron cho kết quả tốt hơn khi so sánh với phương pháp dự báo độ rỗng dựa trên thuật toán Kriging (độ chính xác chỉ đạt 62,26% so với thực tế). Tuy nhiên, thuật toán Kriging cho biết bán kính ảnh hưởng và mối quan hệ không gian giữa các thông số dữ liệu, từ đó có cái nhìn tổng quan hơn về sự phân bố độ rỗng trong vỉa dầu khí.

Tài liệu tham khảo

1. Vũ Hữu Tiệp. *Machine learning cơ bản*. Nhà xuất bản Khoa học và Kỹ thuật. 2018.

2. Trương Xuân Luận. *Lý thuyết địa thống kê*. Đại học Mỏ - Địa chất.

3. Ridha B.C.Gharbi. *An expert system for selecting and designing EOR processes*. Journal of Petroleum Science and Engineering. 2000; 27(1 - 2): p. 33 - 47.

4. Elradi Abass, Cheng Lin Song. *Artificial Intelligence selection with capability of editing a new parameter for EOR screening criteria*. Journal of Engineering Science and Technology. 2011; 6(5): p. 628 - 638.

5. VietAI. *Bài giảng mô hình Neural Network*. 2018.

6. Geraldo A.R.Ramos, Lateef Akanji. *Application of artificial intelligence for technical screening of enhanced oil recovery methods*. Journal of Oil, Gas and Petrochemical Sciences. 2017.

7. Mohamed Sidahmed, Atish Roy, Anjum Sayed. *Steamline rock facies classification with deep learning cognitive process*. SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA. 9 - 11 October, 2017.

8. Nguyễn Hoàng Thiên. *Ứng dụng mô hình địa thống kê dự đoán phân bố đặc tính vỉa*. Đại học Bách khoa Tp. Hồ Chí Minh. 2012.

9. Phan Đăng Võ. *Xác định độ rỗng và độ thấm thành hệ từ tài liệu địa vật lý giếng khoan sử dụng mạng nơ ron nhân tạo*. 2018.

10. Tạ Quốc Dũng, Nguyễn Văn Thuận. *Địa thống kê và ứng dụng trong dự báo các thông số địa cơ học*. 2016.

11. Kelkar Mohan, Godofredo Perez, Anil Chopra. *Applied geostatistics for reservoir characterization*. Society of Petroleum Engineers. 2002.

12. Edward H.Isaaks, R.Mohan Srivastava. *An introduction to applied geostatistics (1st edition)*. Oxford University Press. 1990.

USING ARTIFICIAL NEURAL NETWORK TO PREDICT POROSITY

Ta Quoc Dung[1], Le The Ha[2], Pham Duy Khang[1]
[1]Ho Chi Minh City University of Technology - VNU-HCMC
[2]Vietnam Oil and Gas Group
Email: tqdung@hcmut.edu.vn; halt01@pvn.vn

Summary

The study presents the traditional geostatistic method and the new method using artificial neural network (ANN) to predict porosity. In the traditional method, Kriging algorithm is applied to find the spatial relationship of porosity in the reservoir through 2D models. In the new method, the "nnstart" tool of the Matlab software is applied to build the artificial neural network which will then be used to predict the porosity of the well being studied.

The results are compared with each other and prove that ANN has optimised the porosity prediction for the studied well.

Key words: Geostatistic, Variogram, Kriging, artificial neural network.