

SỬ DỤNG MATLAB ĐỂ HUẤN LUYỆN MẠNG ANNs TRONG BÀI TOÁN ƯỚC LƯỢNG CHI PHÍ XÂY DỰNG CHUNG CƯ

Luu Trường Văn*

Phan Văn Khoa**

Hiện nay khi lập báo cáo đầu tư cho các dự án dân dụng, chúng ta thường dùng phương pháp suất vốn đầu tư hoặc giá thành 1m² sàn XD. Các phương pháp trên đều rất kém chính xác, đặc biệt trong điều kiện VN, khi mà cơ sở dữ liệu về chi phí còn rất sơ sài, không đồng nhất, thiếu cập nhật. Vướng mắc trên là một trong các nguyên nhân gây ra tình trạng vượt chi phí khi thực hiện dự án. Mạng nơ ron nhân tạo (Artificial Neural Networks – ANNs) cho phép ước lượng chi phí xây dựng cho các dự án chung cư một cách dễ dàng trên cơ sở các dữ liệu về chi phí của những dự án đã có trước đây. Huấn luyện ANNs là một công việc quan trọng trong bài toán ước lượng chi phí xây dựng bằng ANN. Các nhà nghiên cứu trên thế giới thường sử dụng các phần mềm phức tạp để huấn luyện mạng ANN. Matlab là một công cụ rất đơn giản để huấn luyện mạng ANNs mà hiện nay chưa được các nhà nghiên cứu Việt nam trong lĩnh vực xây dựng quan tâm. Bài báo này trình bày kết quả nghiên cứu ứng dụng Matlab để huấn luyện mạng ANN trong dự báo chi phí xây dựng chung cư tại TP.Hồ Chí Minh.

In Vietnam, building projects increasingly rise in quantity with the urbanization and population growth. This resulted in estimating construction cost for apartment building projects has become more important because the construction environment is much riskier. Artificial neural networks (ANNs) can help less experienced estimators to predict construction costs because the neural network (NN) can learn from historical data and then generalize its characteristics. The aim of this paper is how to train ANNs using MATLAB so as to obtaining the weighted matrix for estimating construction costs of apartment building projects in an early phase. The model was developed that consisting of an input layer with 6 nodes, and an output layer with one node. Inputs were identified through literature comprehensive review. The data for training NN was actual data sets of fourteen completed apartment building projects. MATLAB software has been used to train the NN. The study resulted in the weighted matrix which is an input to estimate construction cost of apartment projects. Findings may prove the reliability of using neural networks to cost models. The findings also encourage practitioners from developing countries as Vietnam to benefit from the NN technique.

* TS. Trưởng Khoa Kỹ thuật & Công Nghệ, Trường đại học Mở TP.HCM

** Th.S. Công ty tư vấn Meinhardt VN

1. Phương pháp nghiên cứu:

Mục tiêu nghiên cứu:

- Thu thập dữ liệu để chọn các biến đầu vào (input) và các xuất lượng (output)
- Thiết lập mô hình thuật toán và sử dụng Matlab để huấn luyện mạng ANN nhằm có được bộ ma trận trọng số (weights matrix) phục vụ cho bài toán ước lượng.

Phạm vi nghiên cứu: Các dự án chung cư tại TP.HCM

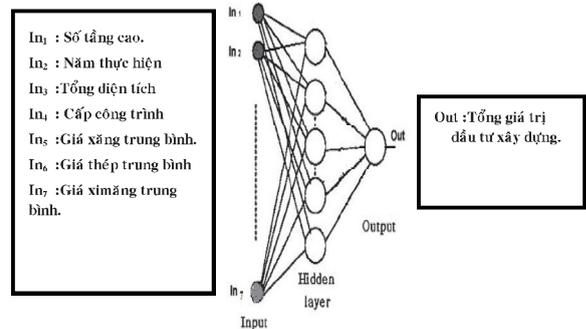
Dữ liệu thu thập: Các số liệu về thiết kế, quyết toán của các công trình đã thực hiện xong và các số liệu dự toán thiết kế của các công trình đang triển khai.

Mẫu nghiên cứu (research sample) là các chung cư được đầu tư xây dựng từ 2000 đến 2006, bao gồm 3 chung cư 5 tầng, 1 chung cư 9 tầng, 1 chung cư 11 tầng, 5 chung cư 12 tầng, 1 chung cư 14 tầng, 2 chung cư 15 tầng và 1 chung cư 25 tầng.

Các input của mô hình ANNs:

- Năm thực hiện xây dựng
- Cấp công trình
- Tổng diện tích sàn xây dựng
- Số tầng cao
- Giá xăng trung bình tại thời điểm xây dựng
- Giá sắt thép trung bình tại thời điểm xây dựng
- Giá xi măng trung bình tại thời điểm xây dựng

Output của mô hình ANNs: Tổng chi phí xây dựng



Hình 1: Mô hình mạng ANNs để dự báo chi phí xây dựng chung cư

2. Thu thập dữ liệu để huấn luyện (training) mạng ANN

Dữ liệu về 14 chung cư đã hoàn thành tại TP.HCM được trình bày trong bảng bên dưới.

Bảng 1. Dữ liệu của 14 chung cư hiện hữu tại TP.HCM

Công trình	Đơn vị	1	2	3	4	5	6
Chiều cao tầng	m	12	12	25	12	5	12
Tổng diện tích sàn xây dựng	m ²	9259	21363	41289	18680	3819	11990
Cấp công trình		2	2	1	2	3	2
Năm bắt đầu xây dựng		2000	2002	2002	2003	2003	2003
Giá xăng trung bình	Lit	4800	5400	5400	5500	5500	5500
Giá cốt thép trung bình (đường kính >=10mm)	Kg	4375	5025	5025	5653	5653	5653
Giá xi măng trung bình	Kg	853	892	892	900	900	900
Tổng chi phí	Đồng	24274596203	54377505193	165423049019	65379591922	7614996967	39579000000

7	8	9	10	11	12	13	14	Max	Min
12	5	11	14	9	5	15	15	25	5
10313	1508	16838	6384	5482	2413	13350	11857	41289	1508
2	3	2	2	3	3	2	2	3	1
2003	2003	2004	2005	2005	2005	2005	2006	2006	2000
5500	5500	6458	8692	8692	8692	8692	10250	10250	4800
5653	5653	7367	7400	7400	7400	7400	7505	7505	4375
900	900	902	884	884	884	884	869	902	853
42781413000	3007261000	37569769727	28510000000	17951921743	8447875363	46724764835	45057739755	165423049019	3007261000

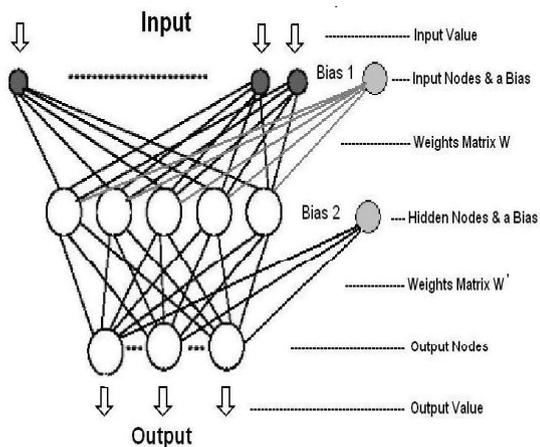
2. Xây dựng thuật toán:

Trình tự thực hiện một bài toán bằng ANN được mô tả trong hình 2. Quá trình này bao gồm 6 bước:

Bước 1: Tổ chức dữ liệu

Trong bước này, các biến đầu vào (N) được đặt vào nút của nó và biến đầu ra hoặc biến cuối cùng (O) được xác định bởi nút (node) tại lớp xuất – đầu ra.

Mỗi một biến đầu vào và đầu ra được định nghĩa bởi dữ liệu phù hợp của chúng đã được chọn từ (P) trường hợp huấn luyện. Những dữ liệu này trở thành giá trị cho bước huấn luyện của Neural Network (NN).



Hình 2: Quy trình thực hiện của mạng ANNs

Những dữ liệu đầu tiên được chuyển đổi thành giá trị số và được chứa trong danh

sách dữ liệu – đó là một ma trận của (N+O) cột và (P) dòng

Bước 2: Scale dữ liệu

Dữ liệu của ma trận đầu tiên được tỷ lệ theo cấp độ từ [-1 đến 1] để cấu trúc thành ma trận thứ 2 với 1 công thức tuyến tính cho tỷ lệ giá trị của ma trận đầu tiên

$$Scale\ Value = \frac{2 * (Unscaled_Value - Column_Min)}{(Column_Max - Column_Min)} - 1 \quad (1)$$

Bước 3: Xây dựng và xác định những giá trị ban đầu của ma trận trọng số (W).

Tất cả các giá trị trong ma trận trọng số (W) được hiểu như những biến số để xác định mô hình Neural Network. Mọi giá trị đầu vào (1 đến N) và một nút định hướng được kết nối đến nút ẩn (hidden nodes). Số nút ẩn (L) là tập hợp 1 nửa của tổng nút nhập và nút xuất.

Bước 4: Xuất nút ẩn

Bước này cho phép ẩn (hide) các nút để xử lý dữ liệu nhập và xuất giá trị để được các lớp (layer) tiếp theo.

Bước 5: Xác định ma trận trọng số W'

Một ma trận thứ 2 được cấu trúc để nối (L) nút ẩn và nút bias (bias node) đến nút

xuất đơn.

Bước 6: Xuất final NN

Xuất lượng (O) được tính toán bởi tổng hợp các kết quả các kết quả của mỗi út ẩn bằng cách liên kết các trọng số của nó và sau đó đưa những giá trị này vào hàm tanh (hàm được chọn trong bài toán này)

Bước 7: Quay ngược lại Neural Network xuất và tính toán lỗi (error)

Để tính toán một đo lường của sự thực hiện Neural Network (NN), một cột sẽ được cấu trúc cho việc xác định lỗi giữa giá trị thực xuất ra và giá trị NN xuất ra như sau :

$$\text{Estimating Error (\%)} = \frac{(\text{Neural_Network_output} - \text{Actual_Output}) * 100}{\text{Actual_output}} \quad (7)$$

Giá trị lỗi trung bình của mỗi nhóm trường hợp có thể được tính toán theo :

$$\text{Weighted Error(\%)} = 0.5(\text{test set average Error}) + 0.5(\text{Training set Average Error})$$

Ở đây những trọng số 0,5 và 0,5 là được giả định để minh họa. Bước này là khá quan trọng để kiểm tra nhằm đảm bảo sự

thực hiện tổng quát và sau đó cho phép huấn luyện mạng.

3. Huấn luyện mạng ANN cho bài toán ước lượng chi phí xây dựng chung cư bằng Matlab :

Phần mềm Matlab – một software có khả năng quản lý tổ về mảng (array), trường (field) và ma trận (matrix) - với hàm các hỗ trợ cho việc huấn luyện mạng ANN được chọn để thực hiện bài toán. Thuật toán Levenberg-Marquardt Back-propagation được chọn bởi vì có tốc độ hội tụ nhanh nhất và sai số ít nhất. Hai hàm học thích nghi sau đây được sử dụng :

- Learngd: Gradient descent weight/bias learning function.

(Hàm học giảm gradient sử dụng trọng số nút bias (hình höông))

- Learnghm: Grad. descent w/momentum weight/bias learning function.

(Hàm học giảm gradient sử dụng trọng số nút bias (hình höông) tăng cường)

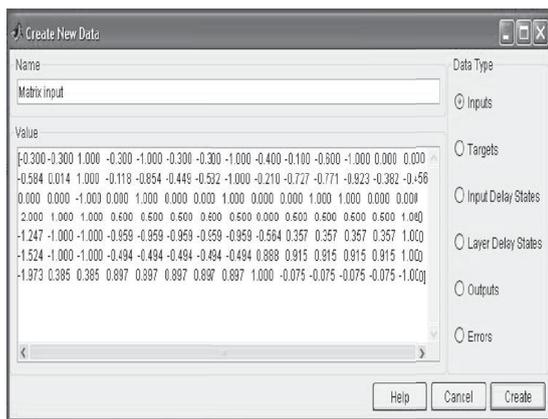
Quá trình huấn luyện mạng NN tuân tự theo các bước sau:

Bảng 2. Bảng ma trận [-1 ; 1] (sử dụng cho mục đích scale dữ liệu)

Công trình thứ	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Chiều cao tầng	-0.300	-0.300	1.000	-0.300	-1.000	-0.300	-0.300	-1.000	-0.400	-0.100	-0.600	-1.000	0.000	0.000
Tổng diện tích sàn XD	-0.610	-0.002	1.000	-0.137	-0.884	-0.473	-0.557	-1.000	-0.229	-0.755	-0.800	-0.954	-0.405	-0.480
Cấp Công trình	0.000	0.000	-1.000	0.000	1.000	0.000	0.000	1.000	0.000	0.000	1.000	1.000	0.000	0.000
Năm bắt đầu XD	-1.000	-0.333	-0.333	0.000	0.000	0.000	0.000	0.000	0.333	0.667	0.667	0.667	0.667	1.000
Giá xăng tb	-1.000	-0.780	-0.780	-0.743	-0.743	-0.743	-0.743	-0.743	-0.391	0.428	0.428	0.428	0.428	1.000
Giá Sắt thép tb	-1.000	-0.585	-0.585	-0.184	-0.184	-0.184	-0.184	-0.184	0.912	0.933	0.933	0.933	0.933	1.000
Giá xi măng tb	-1.000	0.586	0.586	0.931	0.931	0.931	0.931	0.931	1.000	0.277	0.277	0.277	0.277	-0.346
Tổng Chi Phí	-0.738	-0.367	1.000	-0.232	-0.943	-0.550	-0.510	-1.000	-0.574	-0.686	-0.816	-0.933	-0.462	-0.482

- Nhập ma trận các biến đầu vào:

Sau khi khởi động Matlab, rồi nhập “>> nntool” ta sẽ nhận được giao diện cho phần huấn luyện mạng NN của Matlab. Tiếp tục bấm vào nút “New data” để nhập ma trận đầu vào. Kết quả sau khi nhập ma trận đầu vào thể hiện trên hình 3.

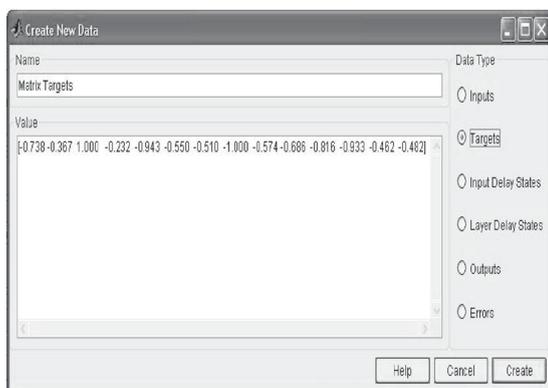


Hình 3: Ma trận các biến đầu vào (input)

- Nhập ma trận đầu ra (các giá trị đầu ra mà chúng ta mong muốn).

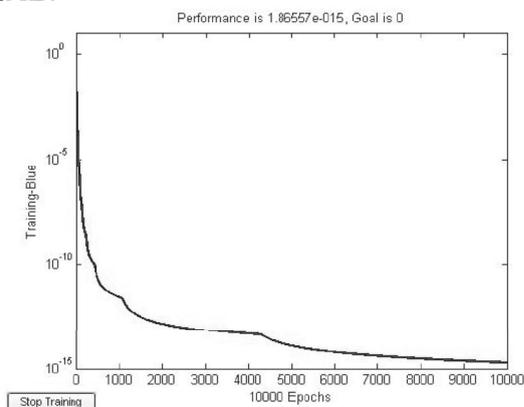
Kết quả nhập ma trận các biến đầu ra phục vụ cho việc huấn luyện (training) mạng. Kết quả nhập ma trận các biến đầu ra được thể hiện trong hình 4.

Hình 4: Ma trận các biến đầu ra (output)



- Hiệu lực (validation) việc huấn luyện.

Việc huấn luyện mạng ANN được kiểm tra bằng các kỹ thuật kiểm định thống kê (statistical testing) theo MSE (Mean Square Error) hoặc MAPE (Mean Absolute Percent Error). Trong nghiên cứu này, các tác giả chọn MSE để kiểm định hiệu lực của việc huấn luyện mạng. Kết quả kiểm tra giá trị MSE sau khi huấn luyện mạng bằng MATLAB.



Hình 5: Biểu đồ giá trị MSE (Mean Square Error) sau khi kết thúc huấn luyện với 10.000 epoch (Sai số của quá trình huấn luyện : $1.86557 \cdot 10^{-15}$)

Tổng giá trị sai số (sau khi Scale back):
 $= 0.5 * (0.04\% + 0.06\% + 0.00\% + 0.01\% + 0.28\% + 0.07\% + 0.04\%) + 0.5 * (0.14\% + 0.09\% + 0.01\% + 0.01\% + 0.00\% + 0.06\% + 0.03\%)$
 $= 0.335\%$

Sai số trên là chấp nhận được đối với các công trình xây dựng.

Bảng 3 trình bày so sánh giữa giá trị thực chi phí xây dựng và giá trị chi phí mà mạng ANN học được từ sự huấn luyện của MATLAB. Kết quả huấn luyện mạng ANN do MATLAB thực hiện được trình bày trong bảng 4 và bảng 5.

Bảng 3. Bảng ma trận so sánh giá trị thực và giá trị đầu ra do MATLAB huấn luyện mạng ANN

Công trình thứ	1	2	3	4	5	6
Network - Output	-0.738	-0.367	0.99999	-0.232	-0.943	-0.55
Giá trị chi phí mà mạng ANN đã học được	24,283,729,230	54,411,857,908	165,422,236,940	65,374,923,599	7,636,110,959	39,550,813,304
Giá trị thực của chi phí xây dựng	24,274,596,203	54,377,505,193	165,423,049,019	65,379,591,922	7,614,996,967	39,579,000,000
Sai số (%)	0.04%	0.06%	0.00%	-0.01%	0.28%	-0.07%

7	8	9	10	11	12	13	14
-0.51	-0.99995	-0.574	-0.686	-0.816	-0.933	-0.462	-0.482
42,799,129,065	3,011,321,395	37,601,823,848	28,506,539,719	17,949,513,498	8,448,189,899	46,697,107,977	45,072,950,097
42,781,413,000	3,007,261,000	37,569,769,727	28,510,000,000	17,951,921,743	8,447,875,363	46,724,764,835	45,057,739,755
0.04%	0.14%	0.09%	-0.01%	-0.01%	0.00%	-0.06%	0.03%

Bảng 4. Bảng ma trận trọng số đầu vào (input)

Ma trận trọng số đầu vào (input)							Bias 01
1.1124	0.3224	-1.1094	-1.1711	0.4989	-0.8225	-0.0518	-1.7051
-0.6106	0.5793	-0.3817	0.9256	-0.8105	0.7816	1.0187	1.5917
-1.0148	3.6641	2.9649	0.3725	0.7141	-0.0281	1.2533	-1.23
-0.505	1.9019	-1.3054	0.962	-0.8469	-0.1138	0.3069	1.6584
1.5712	-1.79	-2.0883	-0.1351	0.5188	-0.5963	0.4269	-0.19127
1.3568	-1.1148	-1.3311	-0.2283	-0.1227	0.6551	1.1406	0.75045
0.1875	1.9339	-0.9328	0.9793	0.5886	-0.2375	-0.3799	-0.67033
-0.9627	0.59	-0.2009	0.1693	-0.1085	1.1706	-0.4881	-1.0575
0.379	1.335	-0.1611	-0.4111	-0.3658	1.6904	-0.3848	2.1423
-0.145	1.2507	-0.9378	0.5089	-0.9667	0.1631	0.0438	-1.985

Bảng 5. Ma trận trọng số đầu ra (Weight to layer, output)

Ma trận trọng số đầu ra (output)									Bias02	
0.82897	0.077449	3.8453	2.3903	2.2038	-1.2196	0.84679	-0.28416	2.2507	0.14607	1.0659

Bộ trọng số trong bảng 4 và bảng 5 là dữ liệu cơ sở cho việc dự báo chi phí xây dựng chung cư. Các tác giả đã xây dựng một phần mềm sử dụng bộ trọng số nói trên để dự báo chi phí cho một số chung cư tại TP.HCM. Kết quả dự báo khi so sánh với chi phí thực của một số chung cư tại TP.HCM là nhỏ hơn 5%. Mã nguồn của phần mềm nói

trên độc giả có thể liên hệ TS. Lưu Trường Văn: ltvan@hcmut.edu.vn hoặc Th.S. Phan Văn Khoa: pvankhoa79@meinhardt.com.vn

Tài liệu tham khảo

[1]. Neural Networks Toolbox for use with Matlab- Howard Demuth, Mark Beale

[2] Ứng dụng Neural Network để dự báo

chi phí chung cư tại TP.HCM của Phan văn Khoa, Đại học Bách Khoa TP.HCM.

[3]. Estimating software development effort with connectionist models by Gerhard Wittig, Gavin Finnie, 1997.

[4]. A Learning Vector Quantization Neural Network Model for the Classification of Industrial Construction Projects by VK GUPTA, JG CHEN, MB MURTAZA, 1997.

[5]. Neural Network Model for Estimating Construction Productivity by Jason Portas and Simaan Abourizk. ASCE, Journal of Construction Engineering and Management, 1997.

[6]. A framework for developing an expert analysis and forecasting system for construction projects – Hashem Al-Tabatabai, 1998

[7]. Regularization Neural Network For Construction Cost Estimation by Hojjat

Adeli and Mingyang Wu. ASCE, Journal of Construction Engineering and Management, 1998.

[8]. Neural Network Model for Parametric Cost Estimating of Highway Project by Tarek Hegazy and Amr Ayed. ASCE, Journal of Construction Engineering and Management, 1998.

[9]. Neural Network Model to Support international Market Entry Decisions by Irem Dikmen and M. Talat Birgonul. ASCE, Journal of Construction Engineering and Management, 2004.

[10]. Using neural network to predict performance of design-build projects in Singapore- Florence Yean Yng Ling, Min Liu. Building and Environment, 2004.

[11]. Improving the COCOMO model using a neuro-fuzzy approach by Xishi Huang, Danny Ho, Jing Ren, Luiz F. Capretz, 2005.