A Deep Learning Approach to Automated Structural Engineering of Prestressed Members

Ahmed A. Torky
The British University in Egypt, ElSherouk City, Cairo, Egypt
Email: ahmed.torky@bue.edu.eg

Anas A. Aburawwash
Canadian International College, Sheikh Zayed, Giza, Egypt
Email: anas_abdulhakim@cic-cairo.com

***Abstract*—In this paper, an implementation is presented of deep learning on the structural engineering of prestressed concrete members. Prestressed concrete beams and slabs are essential structural members supporting the floors of buildings, yet their optimum design is still a challenge for engineers as they struggle to design sections that adhere to serviceability and economical needs. Recently, the advancement of artificial neural networks has managed to propose more optimum solutions to general engineering applications with ease. Deep learning and grid search available hyperparameters can be utilized to predict optimum prestressing of members, without the need for structural engineers to produce countless analysis and design iterations. A simple prestressed beam is presented as an initial example to show the viability of neural networks against the traditional approaches. Two industrial examples of a continuous beam and a slab-beam type are added to demonstrate scalability of the design.**

***Index Terms*—deep learning, structural engineering, prestressing, artificial neural networks, economic design**

I. Introduction

The emerging breed of civil structures and systems of changing attributes need accurate, automated, and optimized methods to design structural members in more suitable periods of time without the hassle of iterative design. Recent reinforced concrete constructions have pursued smaller and more economic section designs that weigh less. Thus, the philosophy of prestressing members was implemented in design codes [1]. The advantages of prestressed concrete members are:

- Wider spans with a reduction in cross-section.
- Improved member deflection behavior.
- Reduction in crack propagation.
- More economic structures.

Design of prestressed structural members is highly nonlinear and dependent on many factors, yet all its features can be quantified and assessed. Deep learning, a branch of artificial intelligence, works well with nonlinear systems [2] to produce optimized predictions.

Deep learning has been successfully used in structural engineering, such as in predicting deformations of the well-known ten-member truss [3]. Neural networks have also been used to predict concrete compressive strength based on extensive and available lab data [4]. Other implementations of neural networks are for the analysis of structural frames, where unsupervised training concepts were used [5]. The following paper also reviews past use of neural networks in structural engineering [6] prior to improvement of computing.

This paper revises briefly concepts of deep learning and artificial neural networks in the section 2. A framework is then proposed in section 3 for optimized design of prestressed beams. Section 4 provides verification examples of the proposed framework and furthermore industrially relevant examples.

II. Neural Networks

A. Network Architecture

A deep learning neural network comprises of at least one input layer, one output layer and several hidden layers in between [7]. The input layer and its nodes represent the predictive features, whereas the output layer and its nodes are the target predictions. While inputs and outputs can be physical values that correspond to actual data, the values in the hidden layer aren't something to observe directly. Nevertheless, each node in any hidden layer represents an aggregation of information to capture interactions from input data.

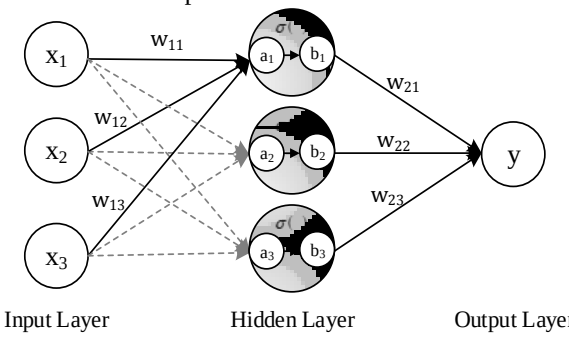

Figure 1. Neural network components.

Forward propagation is the movement of manipulated information from the input layer, to intermediate hidden layers, then to the output layer through a multiply-add process. Fig. 1 shows such a fully connected architecture. Inputs are defined with x_i, hidden nodes are a_i computed

Manuscript received December 24, 2017; revised August 8, 2018.

© 2018 Int. J. Struct. Civ. Eng. Res.
doi: 10.18178/ijscer.7.4.347-352

through the dot product of X $\{x_1, x_2, x_3\}$ and weights W_{1i}. The output of the hidden layer is b_i, calculated by multiplying a_i with an activation function σ (e.g. sigmoid, tanh, softmax, softplus, relu, and selu). Finally, the output in the final layer is calculated through the dot product of B $\{b_1, b_2, b_3\}$ and weights W_{2i}.

A neural network must be iteratively and continuously trained until it predicts viable and near accurate results. This could be achieved by modifying hyperparameters of the network, which will be discussed in section 2.2. Accuracy, however, is calculated by the mean squared error function (MSE) of the output of the network compared with the expected target.

In supervised learning, all weights used in the forward propagation can yet be optimized through the backpropagation. Backpropagation takes the error from the output layer backwards through the hidden layers towards the input layer, modifying weights sequentially. Weight functions can be updated with several readily available optimizers (e.g. gradient descent, Adagrad, Adam) to find a local minimum of that function. The overall process becomes:

- Configure a model
- Process the forward propagation
- Go back one layer at a time
- Backpropagate from output to hidden layers sequentially
- Update weights with each epoch.

B. *Grid Search with Hyperparameters*

There are several parameters to configure for a neural network, and thus these hyperparameters are the most crucial part of deep learning model optimization. These hyperparameters are:

- Epochs count
- Number of hidden layers
- Number of neurons
- Learning rate and training optimization algorithm
- Neuron activation functions
- Batch size

Tuning of deep learning models is now made easy with the use of the Python deep learning library Keras [8]. As different purpose machine learning models have different optimized hyperparameters, it is only convenient to do a full grid search and study the effects of these parameters on model performance.

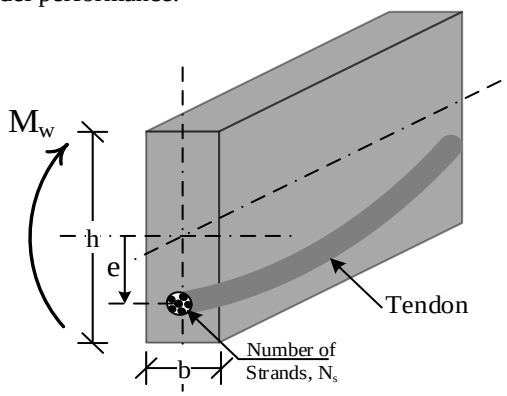

Figure 2. Strands and tendon in a prestressed concrete beam.

III. Prestressing Beam Models

A. *Prestressed Concrete Theory*

Prestressing of concrete beams greatly reduce its dimensions, as the capacity of the beam is increased. The prestressing steel strands are accumulated in separate tendons and exert compressive stresses along the cross-section. This design prevents cracks from developing along the top and bottom fibers of the section. Fig. 2 shows the tendon profile along an arbitrary concrete beam.

A structural engineer would propose dimensions of any given beam under a loading case of bending moment, along with the number of strands in the prestressing tendons and their eccentricity from the centerline. Top and bottom stresses of the member are calculated at the midspan and compared to the maximum allowable concrete stresses. The Egyptian Code of Practice (ECOP) [9] specifies the compressive stresses maximum as $0.45f_{cu}$ and the maximum tensile stresses of concrete as $0.22\sqrt{f_{cu}}$, where f_{cu} is the ultimate compressive strength of concrete. The top and bottom stresses are calculated using the equations (1) and (2).

$$f_{top} = \frac{P}{A_c} + \frac{P\,e}{Z_{top}} - \frac{M}{Z_{top}} \quad (1)$$

$$f_{bot} = \frac{P}{A_c} - \frac{P\,e}{Z_{bot}} + \frac{M}{Z_{bot}} \quad (2)$$

(Sign Convention) (+) Compression Stress, (-) Tension Stress

In which,

f_{top} , is the stress of sections' top fibers

f_{bot} , is the stress of sections' top fibers

P, is the prestressing jacking force of the tendon

A_c, is the cross-sectional area

e , is the tendon eccentricity measured form the centroid of the section

Z_{top} , is the section modulus at the top

M , is the flexural moment due to external loads at the section

Z_{bot}, is the section modulus at the bottom

It is possible to predict approximately the amount of prestressing steel from P in equations or the eccentricity of tolerable prestressing steel. This study makes use of the later concept to produce two new equations reformulated from equations (1) and (2) in separate cases where either the top or bottom stresses reach their maximum allowable stresses.

$$e_1 = \frac{Z_{top}}{A_c} + \frac{M-(f_{top}{}^{*}\,Z_{top})}{P} \quad (3)$$

$$e_2 = \frac{-Z_{top}}{A_c} + \frac{M+(f_{bot}{}^{*}\,Z_{bot})}{P} \quad (4)$$

After calculating both e_1 and e_2 from equations (3) and (4), the least of both produces stresses in both the top and bottom fibers that are acceptable. The bigger of both

© 2018 Int. J. Struct. Civ. Eng. Res.

eccentricities will result in exceeding the maximum tensile stresses in one of the extreme fibers. Infinite amounts of prestressed sections can now be calculated from the above formulation using any programming language.

TABLE I. SAMPLE DATA OF THE OPTIMUM PRESTRESSED BEAMS AND THEIR PARAMETERS (N = 500,000)

#	Depth (mm)	Width (mm)	Bending moment (kN.m)	Eccentricity (mm)	Number of strands
1	860	300	50	280	1
2	1500	750	1300	715	7
3	1200	600	210	650	12
…	…	…	…	…	…
n	2000	1000	1900	365	20

B. Optimum Big Data for Prestressed Beams

Using equations (3) and (4), several thousands of optimum prestressed data can be computed. As the neural network has higher affinity to learn correctly from bigger data, tremendous amounts of sections and their optimum prestressing with different cases of loading (bending moments) can be produced quickly. A Fortran code is written to process and compute this data, calculating 500,000 sections in less than a minute. The pseudo code for this program is shown is shown below. Fortran remains to be one of the fastest programming and computing languages, however, it lacks readily available machine learning libraries.

Algorithm 1: Fortran pseudo code

1) Enter Number of Strands
2) Enter Section Geometry
3) Enter Section Bending Moment
Repeat the following
4) Call Eccentricity Subroutine
5) Compute e1 and e2 and corresponding M1 and M2
6) Choose minimum of M1 and M2 as Msection
7) Call Stresses Subroutine
8) Compute top and bottom stress
9) Call PickedEcc Subroutine
10) Pick eccentricities within allowable stresses
11) Call PickBestEcc Subroutine
12) Pick best eccentricity
13) Write Features: *Depth, Width, Moment, Eccentricity, Number of Strands*
14) End

A sample data is listed in Table I. The beam sections' depths ranged from 600mm to 2000mm, whereas the widths ranged from 300mm to 1000mm. The internal forces (bending moment) in the sections started from 50 kN.m up to 1900 kN.m, whereas prestressing tendon eccentricities were at their maximum for equivalent economic choices of number of strands. The output computed was the minimum number of strands required by each configuration of input features. Note that the number of strands is based on the area of prestressing steel that is required, which is computed from the prestressing jacking force *P*. This data works for either transfer or service loading.

C. Model 1

The first proposed trained neural network model has section depth, section height, bending moment, and eccentricity as the predictors, while the number of strands as the target. The Python pseudocode is presented, Displaying all the steps to build a model, import data, train the model, make predictions with it and plotting performance of the network all with Python and Keras. An example of the trained model is shown in Fig. 3. Training of the network with different hyperparameters has shown that the best activation function is Softplus and the best optimizer is Adam. Three hidden layers with 16 nodes each was the best configuration that can be achieved. The MSE is shown in Fig. 4 to decrease rapidly with the increase of epochs, and as the data is large and strongly correlated, only a few epochs are required.

Training was compared with 3 and 4 hidden layers. From Fig. 5, four hidden layers with 16 neurons give the best performance and thus the least model training time expense.

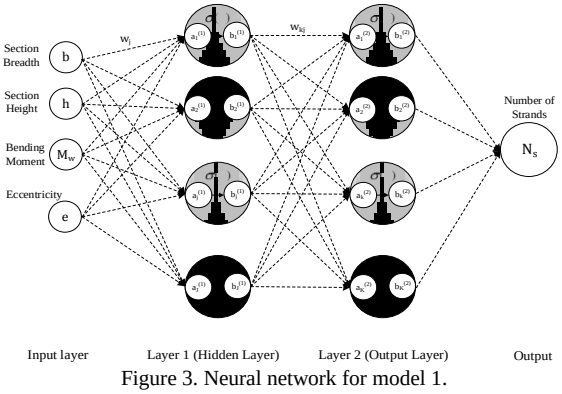

Figure 3. Neural network for model 1.

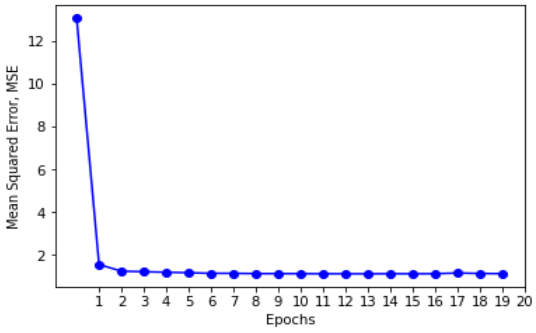

Figure 4. Mean square error for a training Model 1 with (4−16−16−16−16−1).

Algorithm2: Python pseudo code

```
1) Import necessary modules
2) import numpy
3) import matplotlib
4) import keras: layers, models
5) np.loadtxt: predictors,target
6) n_cols = predictors.shape[1]
7) model = Sequential()
8) model.add(Dense(16, activation='softplus',input_shape =
(n_cols,)))
9) model.add(Dense(16, activation='softplus'))
10) model.add(Dense(16, activation='softplus'))
11) model.add(Dense(16, activation='softplus'))
12) model.add(Dense(1))
13) model.compile(adam, loss, accuracy])
14) model_1 = model.fit(predictors,target,epochs=20,
batch_size=20 ,validation_split=0.2)
15) np.loadtxt:data_to_predict_with
16) predictions = model.predict(data_to_predict_with)
17) model.summary()
18) Plot the points using matplotlib
19) plt.plot(model_1.history['loss'],'r')
20) plt.xlabel('Epochs')
21) plt.ylabel('Mean Squared Error, MSE')
22) plt.show()
```

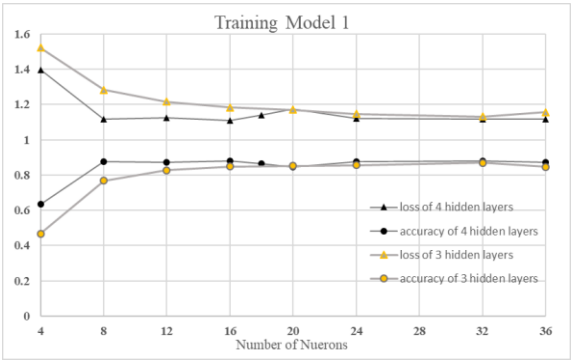

Figure 5. Training of model 1 using 3 and 4 hidden layers.

D. Model 2

The second proposed trained neural network model is achieved to predict eccentricities of continuous beams where the number of strands is already known and enforced from the peak value of one section of the beam. The model has section depth, section height, bending moment, and number of strands as the predictors, while the eccentricity as the target. Training of the network was also completed with a similar configuration and achieved similar performance.

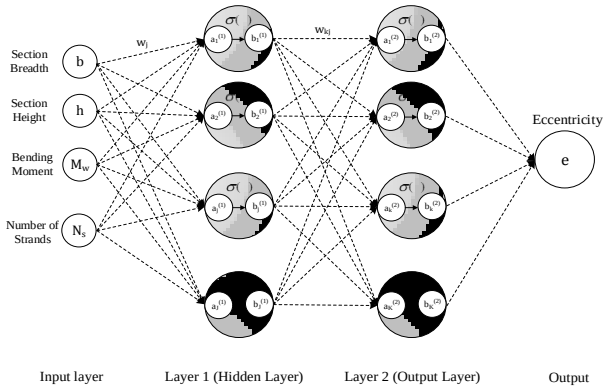

Figure 6. Neural network for Model 2.

E. Proposed Design Framework

An automated framework to designing simple beams to a multitude of continuous beams is displayed in Fig. 6. Material properties of concrete and prestressing steel is entered according to design codes and industrial availability. Massive data of optimized designs of prestressed beams are produced with the Fortran code, and data is exported for training of Model 1 and Model 2. If a beam is simple then only Model 1 is deployed, whereas continuous beams utilize the full capacity of the flow chart displayed in Fig.7.

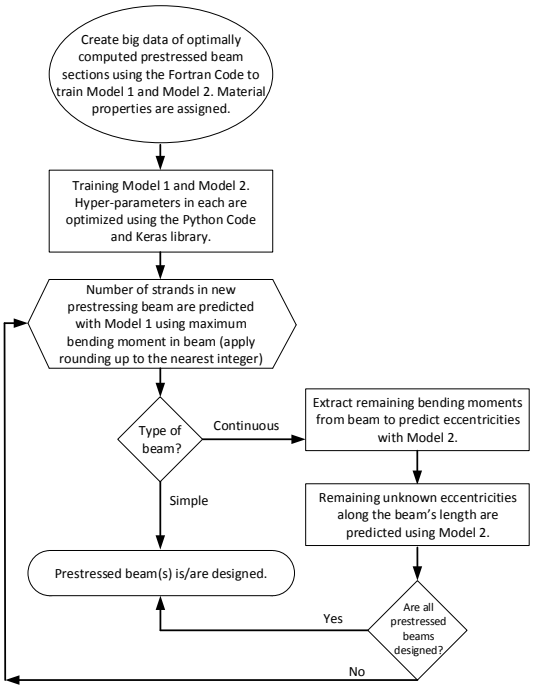

Figure 7. Proposed framework to optimize design.

IV. Numerical Examples

This section provides numerical validation of a simple prestressed beam and two industrial prestressing applications. The proposed framework in section 3.5 is implemented to automate the design of all beam(s) section(s) with maximum positive and negative moments using deep learning.

A. Simple Beam

The example presented here is a benchmark example presented in reference [10], which has been used by structural engineers in Egypt as it gives the best practices in design with the ECOP. The beam material properties and allowable stresses are in Table 2. Section dimensions Fig. 8 are 300mm (width) x 1300mm (depth), and a bending moment of 878.64 kN.m is applied to it. Eccentricity is set at its maximum (100mm from the bottom fibers) to allow for a concrete cover. The prestressing force computed translates to 13 strands (area of 140mm^2 each) in the tendon, while the proposed neural network predicts 8 strands are only necessary.

The method traditionally uses equations (1) and (2) to attain the maximum prestressing force, however, it

cannot predict the optimum state of stresses along the section due to the equating the equations with maximum allowable stresses. Fig. 9 shows the different stress diagrams in both methods. The less the number of strands in the tendon, the less will be the weight of the section and the price of prestressing materials.

TABLE II. PARAMETERS FOR THE SIMPLE BEAM EXAMPLE

Material Properties & Allowable Stresses	
Normal Strength Concrete, f_{cu}	30 MPa
Transfer Concrete Strength, f_{cui}	22.5 MPa
Prestressing Steel Yield stress, f_{py}	1700 MPa
Ultimate stress, f_{pu}	2000 MPa
Compression Stress, f_{ci}	10.125 MPa
Tension Stress, f_{ti}	−1.04 MPa

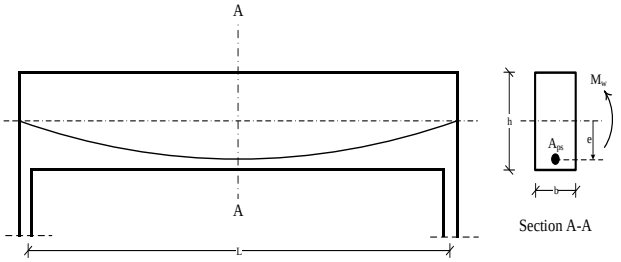

Figure 8. Geometry of the simple beam example.

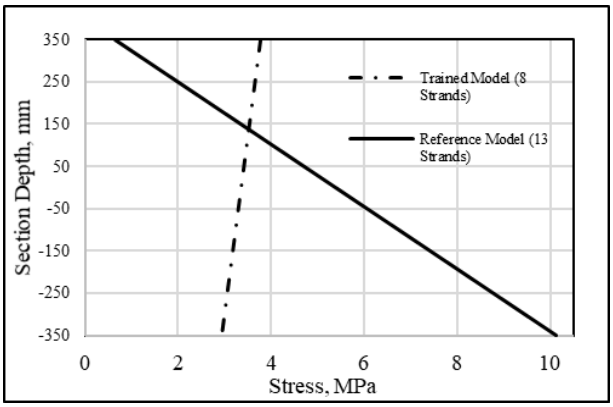

Figure 9. Comparison of section stresses.

B. Continuous Beam

Industrially in construction, reinforced concrete beams are usually continuous over several supports. The framework is tested on the continuous beam presented in Fig. 10. The same material properties are assumed as that of the previous example. Mainly three sections would produce the maximum positive and negative moments, defined by Sec-1, Sec-2, and Sec-3. The absolute maximum of the three is used for inference with Model 1 to predict the number of strands for the overall beam, as only one tendon with a constant number of strands is more economical. Then the other two sections' features are used with Model 2 to predict the eccentricities of the already enforced number of strands. Table 3 presents the final design of the prestressed continuous beam.

C. Slab-beam Type

More accurately, a beam usually support loads from an overlying structural floor/slab. This example uses the boundary element software developed by the first author [11](PLPAK) to solve shear deformable plates under bending to attain bending moments on the beams. Fig. 11 demonstrates the 3D geometry of a practical example. The slab has dimensions of 32mx32m, its thickness is 0.3m, and a uniform gravitational load of 5kN.m is applied on it. All beams have a dimension of 0.3mx0.8m, whereas columns are 1.0mx1.0m.

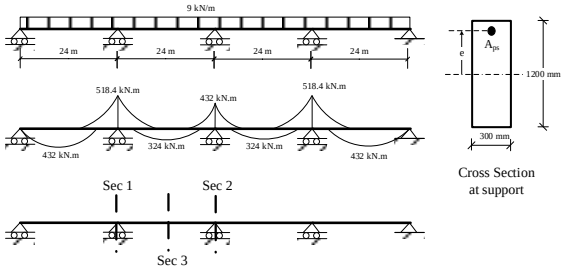

Figure 10. Geometry and loading of the continuous beam.

TABLE II. OPTIMUM DESIGN OF THE CONTINUOUS BEAM.

Cross Section	Bending Moment (kN.m)	Number of Strands	e (mm)	Stresses at fibers, MPa		Status
				Top	Bottom	
Sec-1	-518.4	5 Model 1	500↑	2.05	3.23	Safe
Sec-2	-432.0	5 Model 2	435↑	1.89	3.01	Safe
Sec-3	324.0	5 Model 2	295↓	2.85	2.03	Safe

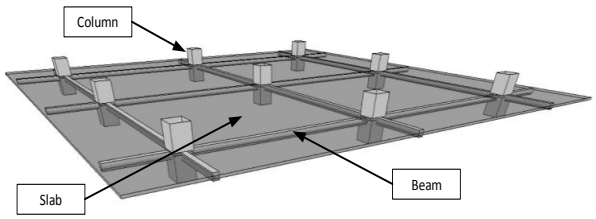

Figure 11. Geometry of the slab-beam type example.

Bending moment results on beams are displayed in Fig. 12 and listed in Table 4. For simplicity, on the two sections on the inner most beam are used for calculating prestressing need.

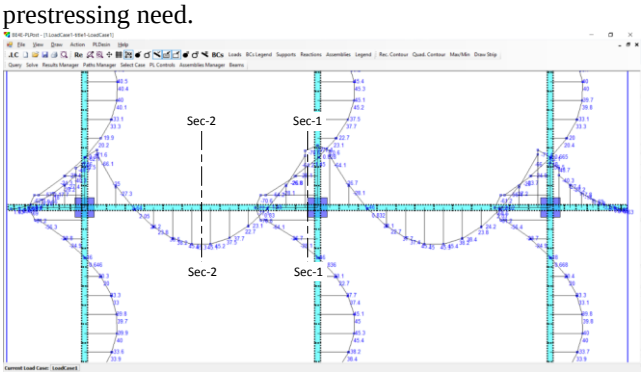

Figure 12. Bending moment results of beams.

© 2018 Int. J. Struct. Civ. Eng. Res.

The proposed framework was implemented again. A tendon of 10 strands is required to prestress this continuous beam.

TABLE III. OPTIMUM DESIGN OF THE CONTINUOUS BEAM IN THE SLAB.

Cross Section	Bending Moment (kN.m)	Number of Strands	e (mm)	Stresses at fibers, MPa		Status
				Top	Bottom	
Sec-1	-716	10 Model 1	350 ↑	5.35	7.34	Safe
Sec-2	454	10 Model 2	280 ↓	6.24	4.41	Safe

V. CONCLUSIONS

This work proposed a deep learning approach to optimize structural engineering of prestressed beams. Big data of optimized prestressed beams was computed using a Fortran code of 500,000 samples, five features each. Two neural networks were trained on the data. Model 1 aimed at predicting the number of strands in the whole beam, whereas Model 2 aimed at predicting eccentricities of the tendon along the beam. A verification example showed that the framework proposed produces a much more economical design than traditional methods, in accordance with the ECP203. Two more industrial examples were solved with the proposed system.

The approach to making use of deep learning in structural engineering will be extended to the design of post-tensioned slabs in the future. Machine learning can be applied to optimizing structural design in general. Further studies beyond this work can include a proposed framework for the optimized design structural elements in tall buildings under gravitational and lateral loads.

REFERENCES

[1] C. R. Bischof. 2002. ACI Manual of Concrete Practice, 2010. Practice 552: 2002–2002. [Online]. Available: https://doi.org/ISSN 0065-7875

[2] A. Goodfellow, Ian, Bengio, Yoshua, Courville. 2016. Deep Learning. MIT Press. [Online]. Available: http://www.deeplearningbook.org/

[3] S. Lee, J. Ha, M. Zokhirova, H. Moon, and J. Lee. 2017. "Background information of deep learning for structural engineering. Archives of Computational Methods in Engineering 0, 1–9. [Online]. Available: https://doi.org/10.1007/s11831-017-9237-0

[4] H. G. Ni and J. Z. Wang. 2000. Prediction of compressive strength of concrete by neural networks. Cement and Concrete Research 30, 8: 1245–1250. [Online]. Available: https://doi.org/10.1016/S0008-8846(00)00345-8

[5] L. R Pinto and A. R. Zambrano. 2014. Unsupervised Neural Network Approach to Frame Analysis of Conventional Buildings. July, pp. 203–211.

[6] P. Hajela and L. Berke. 1992. Neural networks in structural analysis and design: An overview. Computing Systems in Engineering 3, 1–4: 525–538. [Online]. Available: https://doi.org/10.1016/0956-0521(92)90138-9

[7] L. Deng. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing 3. [Online]. Available: https://doi.org/10.1017/atsip.2013.9

[8] Fran ̧ois Chollet. 2015. Keras: Deep Learning library for Theano and TensorFlow. GitHub Repository: 1–21.

[9] ECP Committee 203, "The Egyptian Code for design and Construction of Concrete Structures," Housing and Building Research Center, Giza, Egypt, 2007.

[10] M. Ghoneim and M. El-Mihilmy, "Design of reinforced concrete structures," vol. 3, 3rd edition, 2015.

[11] A. A. Torky and Y. F. Rashed. 2017. GPU acceleration of the boundary element method for shear-deformable bending of plates. Engineering Analysis with Boundary Elements 74: 34–48. [Online]. Available: https://doi.org/10.1016/j.enganabound.2016.10.006

Ahmed A. Torky was born on 15th September 1988 in Cairo, Egypt. He obtained his B.Sc. degree in civil engineering from The British University in Egypt in 2010. He later obtained his M.Sc. in structural engineering from Cairo University in 2015 under the supervision of Youssef F. Rashed. His thesis investigated the parallel computing enrichment of the boundary element method for analysis of shear deformable plates.

After graduation, engineer Torky joined the Civil Engineering Department at The British University in Egypt as a demonstrator and later as an assistant lecturer. Since then, he has been assisting in teaching various Civil Engineering Modules for undergraduate students with enthusiasm. Modules include Prestressed Concrete, Earthquake Resistant Design, Stiffness Analysis, and Geoinformatics. He is also an active member in the boundary element research group at Cairo University, CUFE-BE.

Mr. Torky's research interests are focused on the development of novel computational and computing methods for engineering applications. He has worked mainly on hybrid boundary element and finite element methods for applications including tall buildings analysis including soil structure interactions. His current goal is incorporating deep learning algorithms in structural analysis and design of structures.

Anas A. Aburawwash was born on 17th February 1990 in Cairo, Egypt. Anas obtained his B.Sc. degree in civil engineering from The Higher Institute of Engineering at El-Shorouk City in 2013, and then went to earn his M.Sc. degree in structural engineering in August 2017 from Cairo University in Giza, Egypt under supervision of Prof. Youssef F. Rashed. His thesis developed an interactive integrated environment for analysis and design of post-tensioned floors based on boundary element method.

He is currently Lecturer Assistant at Civil Engineering Department at Canadian International College, Giza, Egypt. He is responsible of teaching structural engineering courses. Mainly, structural analysis courses (e.g., stiffness analysis of structures, stresses), high rise buildings and numerical methods. Furthermore, he is assisting in following structural analysis graduation project. He was researcher at computational mechanics research group at Cairo University, CUFE-BE. Moreover, he was Structural Designer at pioneer firms in Egypt.

Mr. Aburawwash research interests in application of computational mechanics in structural engineering and software development. He did his research, previously, in developing computerized design tool for post-tensioning slabs. He is currently interested in fracture mechanics and simulating cracking problems.

© 2018 Int. J. Struct. Civ. Eng. Res.