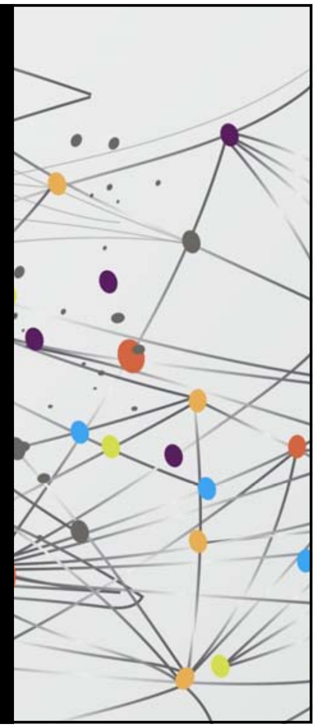
Damage Detection in Concrete Bridge T girders using 3D Finite Element Simulations Trained by Artificial Neural Network

Hayder A. Rasheed, Ph.D. P.E. F.ACI

AlaaEldin Abouelleil, Ph.D.

Eric Fletcher, M.S.



Outline

- Objectives
- Introduction
- Methodology
- Results and Discussion
- Conclusions

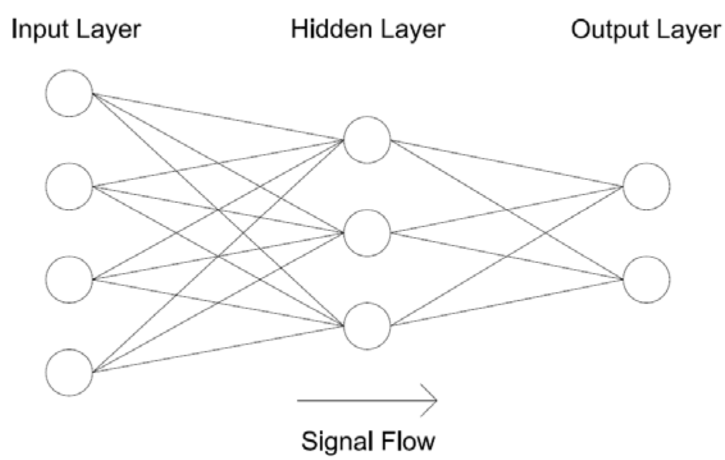
Objective

"Investigate the potential for application of the finite element (FE) method and artificial neural networks (ANN) to create a damage detection model for 3D reinforced concrete bridge T-girders that could be utilized with nodal stiffness ratios as inputs."

- Construct a T-girders Database
- Create, train, validate an Artificial Neural Network (ANN)
- Utilize the ANN for damage detection

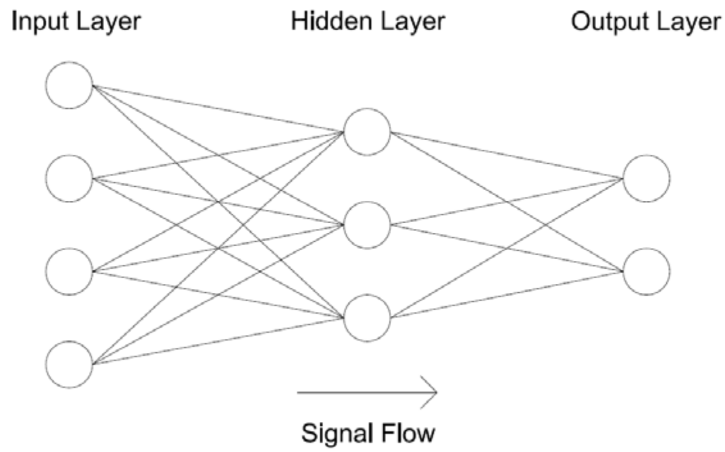
Introduction

- Most network structures consist of at least three layers of neurons, or nodes: an input layer, one or more hidden layers, and an output layer.
- Learning occurs through mathematical operations performed within the hidden layers and their connections to the input and output layers.



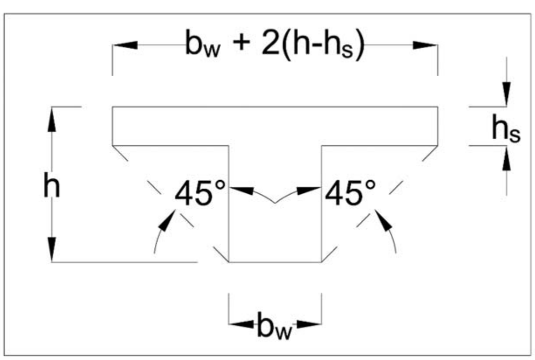
Introduction

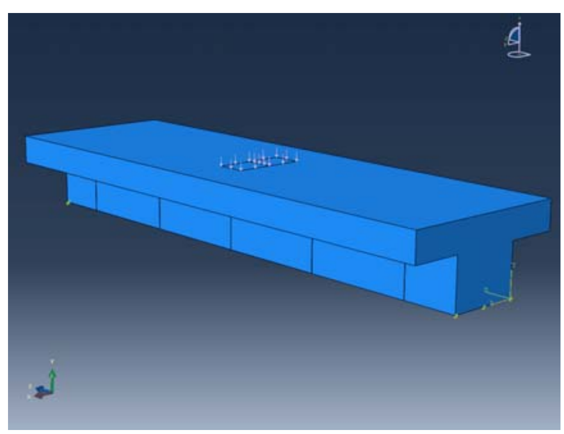
- In feedforward networks, signal flow is unidirectional from the input layer to the output layer, and nodes within layers are not interconnected.
- In supervised learning, outputs predicted by the ANN are judged against the actual or provided outputs.



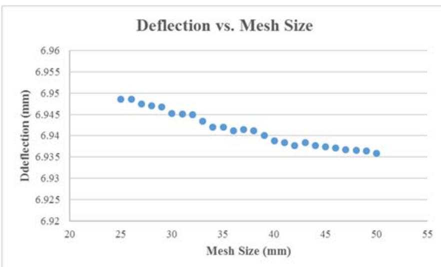
Methodology

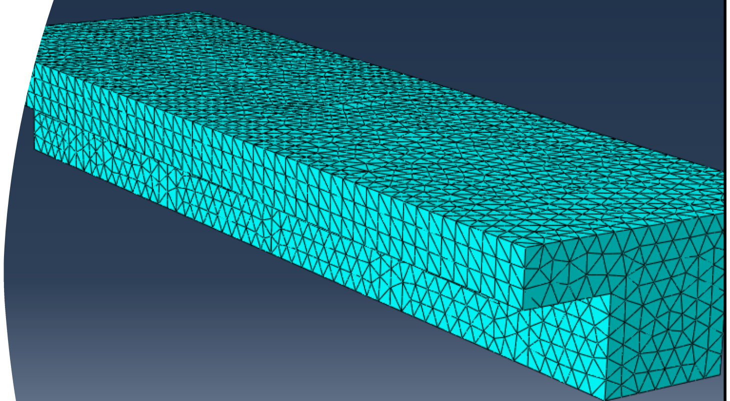
- **3D solid parts .**
- **10-node quadratic tetrahedral (C3D10) elements utilizing an auto meshing technique.**
- **Reinforcement was modeled using 3D wire parts meshed with 3-node truss elements (T3D3), and three equally spaced reinforcing bars were embedded in all concrete beam models.**





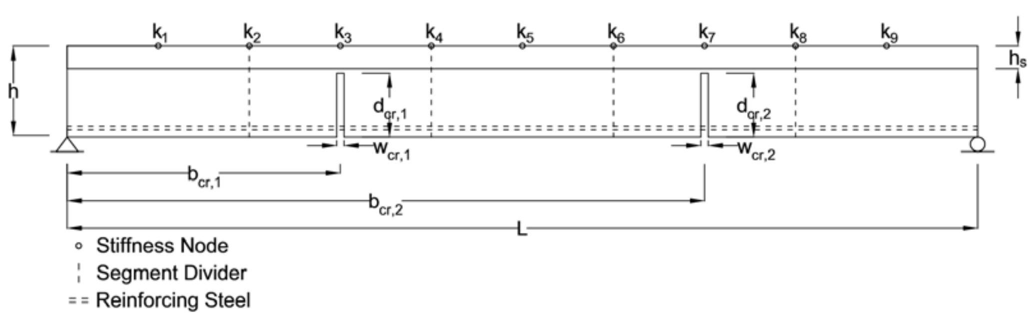
Methodology (Mesh and Convergence)





Methodology (Parameters)

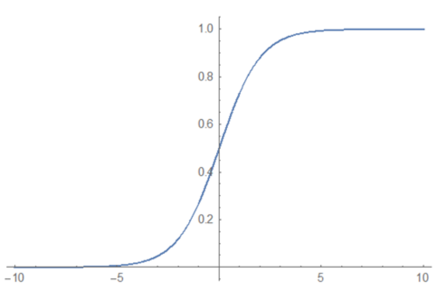
ID	b_w/h	h_s/h	L/h	ρ	f'_c (MPa)	$k\%_n$	$b_{cr,m}/L$	$d_{cr,m}/h$	$w_{cr,m}$ (mm)	*Set*
#	I	I	I	I	I	I	O	O	O	1, 2, or 3



Methodology (Parameters)

<table>
<tr><td>Parameters</td><td colspan="9">Values</td></tr>
<tr><td>b_w/h</td><td colspan="9">0.5 0.7 0.9</td></tr>
<tr><td>L/h</td><td colspan="9">7 10 13</td></tr>
<tr><td>ρ</td><td colspan="9">0.005 0.01</td></tr>
<tr><td>f'_c (MPa)</td><td colspan="9">20 30 40 50</td></tr>
<tr><td>$K\%_n$</td><td>$K\%_1$</td><td>$K\%_2$</td><td>$K\%_3$</td><td>$K\%_4$</td><td>$K\%_5$</td><td>$K\%_6$</td><td>$K\%_7$</td><td>$K\%_8$</td><td>$K\%_9$</td></tr>
</table>

Methodology (Parameters after activation)



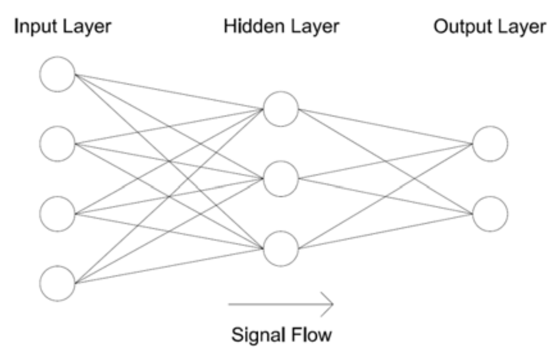
Inputs			Outputs		
Parameter	Minimum	Maximum	Parameter	Minimum	Maximum
b_w/h	0.45	0.95	b_{cr1}/L	0.013333	0.24667
h_s/h	0.18	0.38	d_{cr1}/h	0	0.852
L/h	6.25	13.75	w_{cr1}	0	6.6667
ρ	0.004375	0.010625	b_{cr2}/L	0.13333	0.46667
f'_c	16.25	53.75	d_{cr2}/h	0	0.852
$k\%_1$	0.63539	1	w_{cr2}	0	6.6667
$k\%_2$	0.50907	1	b_{cr3}/L	0.33333	0.66667
$k\%_3$	0.47686	1	d_{cr3}/h	0	0.852
$k\%_4$	0.43376	1	w_{cr3}	0	6.6667
$k\%_5$	0.41189	1	b_{cr4}/L	0.53333	0.86667
$k\%_6$	0.43176	1	d_{cr4}/h	0	0.852
$k\%_7$	0.48287	1	w_{cr4}	0	6.6667
$k\%_8$	0.51158	1	b_{cr5}/L	0.75333	0.98667
$k\%_9$	0.63937	1	d_{cr5}/h	0	0.852
			w_{cr5}	0	6.6667

Methodology (Artificial Neural Network Models)

		Model 1	Model 2	Model 3
Training	R^2	0.40529	0.36465	0.42176
	ASE	0.045959	0.050298	0.04065
Testing	R^2	0.37743	0.2887	0.42179
	ASE	0.042688	0.04356	0.031652
Validation	R^2	0.36535	0.29846	0.40444
	ASE	0.046458	0.048582	0.035467
All Data	R^2	0.39081	0.35084	0.42564
	ASE	0.045105	0.049255	0.040025

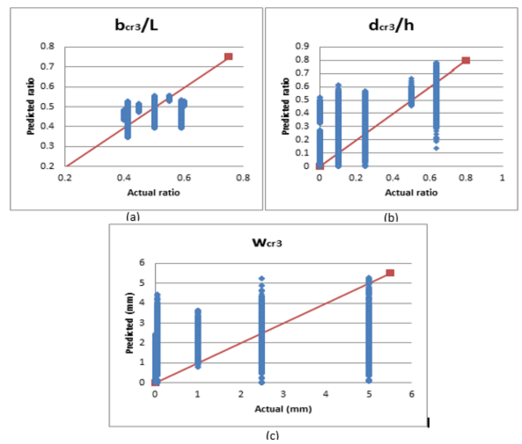
Methodology (Artificial Neural Network)

- 50% of the database for training
- 25% of the data base for testing
- 25% of the database for validation
- 14 Input Nodes, 16 Hidden Nodes, 15 Output Nodes.
- Feedforward, supervised ANN.
- Learning rate 0.01

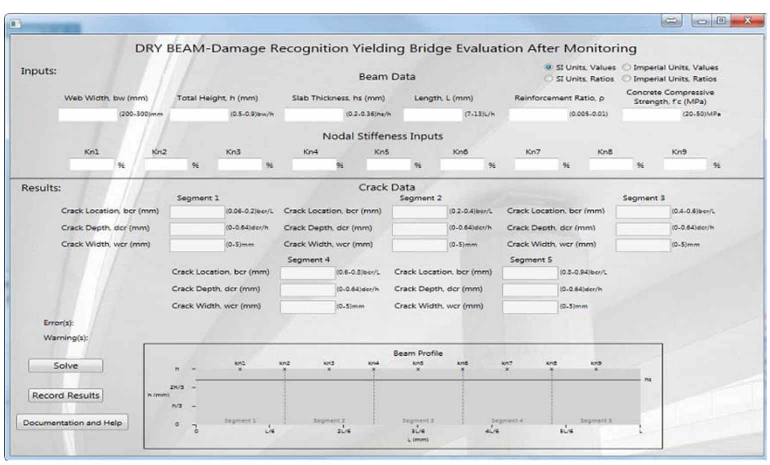


Results

- ANN models are capable of locating the cracks with a very good accuracy.
- The presently collected and trained database was not successful in predicting the depth and width of the cracks.
- A modest agreement against the actual crack configurations with an R^2 equal to 0.42.



Software



Conclusion

- One ANN was employed to predict the damage configurations of beams up to five cracks.
- The predicted damage parameters define the location, the depth, and the width of each crack.
- The ANN accepted the geometric, material, and the nodal stiffness ratios as inputs and predicted the crack parameters as outputs.
- Upon determining the optimum model for crack configuration prediction, the statistical results for this optimum model were modest with R^2 value equal to 0.42 in the training, testing, and validation phases due to the non-uniqueness of these cracking predictions and missing cases from the two, three and four cracks in the database.
- The low R^2 for the overall damage detection ANN is expected to significantly increase if the location of the cracks is considered as the only output parameter to be judged by the network.

Questions

Anafi¡ v¡ t Pa ‡n

- In machine learning and statistics, the **learning rate** is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.[1] Since it influences to what extent newly acquired information overrides old information, it metaphorically represents the speed at which a machine learning model "learns". In the adaptive control literature, the learning rate is commonly referred to as **gain**.[2]