Technical Paper

ISSN 1997-1400 Int. J. Pavement Res. Technol. 8(1):10-16

Prediction of Rigid Pavement Responses under Axle Loads Using Artificial Neural Network

Mofreh Saleh [1+]

Abstract: The analysis of stresses, strains and deflections in rigid pavement is an important step in the mechanistic pavement design of new pavements and rehabilitation design of existing pavements. However rigid pavements are complex structures to analyze utilizing closed form solutions. In order to simulate actual traffic loading cases and realistic pavement structures a finite element analysis is needed. However, in order to perform finite element analysis for different axle load configurations to analyze pavement damage using traffic spectra, it will require a great amount of time for each design option. In addition, using finite element analysis will require special expertise and software which might not be available in many highway departments of transportation. In this research, the outputs of finite element analysis for multiple axle load configurations, rigid pavement structures and material properties were used to train an artificial neural network to predict stresses and deflections in rigid pavement with great accuracy. The fully trained artificial network can be used to predict principal and normal stresses and deflections with reasonable accuracy with a mean squared error of 0.002. The artificial neural network can be used in any mechanistic empirical pavement design procedure to replace chart solutions or lengthy finite element analysis.

DOI: 10.6135/ijprt.org.tw/2015.8(1).10

***Key words**: Rigid pavements; Artificial neural network; Axle loads; Responses.*

Introduction

To analyze the complex rigid pavements structure, finite element analysis will provide the most accurate solution. The finite element methods have come to prominence in complex structures such as rigid pavements due to their ability to accurately model different axle configurations and complex boundary conditions. However, for the purpose of mechanistic empirical design, the design will require several iterations and analysis for multiple axle loads which will require long time to achieve the optimum design. One solution to this issue is to develop and train an artificial neural network that would instantly predict the rigid pavement responses to axle loads in very short time. The closed form solution provided by Westergaard is very simplistic and cannot model complex rigid pavement structure with multiple axle loads [1]. This is because the equations provide maximum values that are location independent and therefore superposition of different results will not produce accurate values. In most rigid pavements, slabs are connected with dowel bars which are embedded in both slabs for load transfer between adjoining slabs. The closed form solutions have no way of modeling how the load is transferred and therefore the corner and mid edge equations are no longer applicable.

Artificial neural networks have been successfully used in many engineering problems. Lacroix *et al.*, Coleri *et al.* and Far *et al.* developed an artificial neural network to predict the dynamic modulus of asphaltic concrete [2-4]. The neural network proved to be more accurate at predicting the dynamic modulus compared to the modified Witczak model. The authors attributed the better quality of prediction of the artificial neural network to the fact that the neural network can model the complex relationships between the resilient and the dynamic moduli whereas the modified Witczak model uses component material properties to model the relationship [2].

In all three studies, the coefficient of determination for the neural network was above 0.9 and this is compared to 0.8 for the modified Witczak model [2-4]. Both the neural network and modified Witczak model had shown an increase in the spread as the dynamic modulus increased, but this disappeared for the neural network when a log-log scale was used. The log-log scale showed the modified Witczak model had a similar spread to the neural network but it was not accurate as it overestimated the higher moduli. Kim *et al*. developed an artificial neural network for maintenance and rehabilitation project selections using pavement preservation plans and pavement condition data for the TxDOT [5]. The authors used maintenance and rehabilitation data collected over four years in addition to other relevant data extracted from the TxDOT Pavement Management Information System. In this study, 80% of data were used for training and the remaining 20% was used for testing and validation.

Mirzahosseini *et al*. investigated the applicability of using artificial neural network for the prediction of the rutting potential by predicting the flow number [6]. The authors used several asphalt mix parameters such as percentage of coarse aggregates, bitumen content, air voids content and percentage of voids in the mineral aggregates as input parameters that correlate with rutting. The authors found that the ANN accurately characterized the flow number and remarkably outperforms several existing prediction models for the flow number of asphalt mixes.

With the advance in the computing power, numerical methods are becoming popular in all areas of engineering, especially where simplistic analytical solutions are no longer acceptable. In this

[1] Associate Professor in Civil Engineering, Department of Civil and Natural Resources, University of Canterbury, Christchurch, New Zealand.

[+] Corresponding Author: E-mail mofreh.saleh@canterbury.ac.nz

Note: Submitted, July 30, 2013; Revised September 5, 2014; Accepted November 6, 2014.

research paper, different loading combinations, layer moduli and pavement composition will be modeled by finite elements and the outputs are used to train and validate an artificial neural network.

Finite Elements Analysis

In this research, the slab and base layers are modeled with 20 node brick elements and are treated as linearly elastic and isotropic material. The elements are rectilinear and the same divisions are used in the base as in the slab to ensure their compatibility. Only linear strains can be accurately modeled across this element.

The interface between the slab and the uppermost base layer is modeled by a 16 node, zero thickness quadratic interface elements. This model is used to model the shear transfer at the slab-base interface and the relationship is assumed to be elastic-plastic with the ultimate stress and yield strain specified. An eight node element is used to model the dense liquid foundation and it can support tension or be tensionless. Standard quadratic shape functions are used to interpolate vertical displacements that are compatible with the 20-node brick element.

The accuracy of the finite element analyses depends on how fine the mesh used to solve the problem. The smaller the elements, the more likely the solution approaches the true result but the time to run the analysis increases as well. In order to ensure the solution is accurate several iterations has been made in which, the element size was reduced to half the elements size of the previous iteration and the analysis was compared between successive runs. When the solution has only varied by less than around 1.0% then the pervious solution is accurate enough.

Methodology

Description of Artificial Neural Network

A neural network is comprised of three main parts; inputs, an output layer and at least one hidden layer. The hidden and output layers are made up of neurons which receive an input from every neuron in the preceding layer multiplied by a weight and then summed. A bias is then added to the sum and the result is now passed on to every neuron in the following layer through a transfer function as shown in Fig. 1. The transfer functions can range from piece-wise to linear to sigmoidal.

When a neural network is being used to model multivariate nonlinear relationships, multiple hidden layers are normally needed. A network can be simplified to a single hidden layer without any loss in accuracy if linear transfer functions are used for all the neurons in a multi-layered network. This explains why the transfer functions on hidden neurons are typically sigmoid in nature. A wide range of training data is needed because a sigmoidal function has asymptotes and as such cannot extrapolate beyond that range [2]. A neural network at its most basic is a group of linear relationships being added together to model a nonlinear one.

Axle Load Configurations and Pavement Geometry

Eight different variables were chosen to be inputs into the neural network with a maximum of three levels for each variable as shown

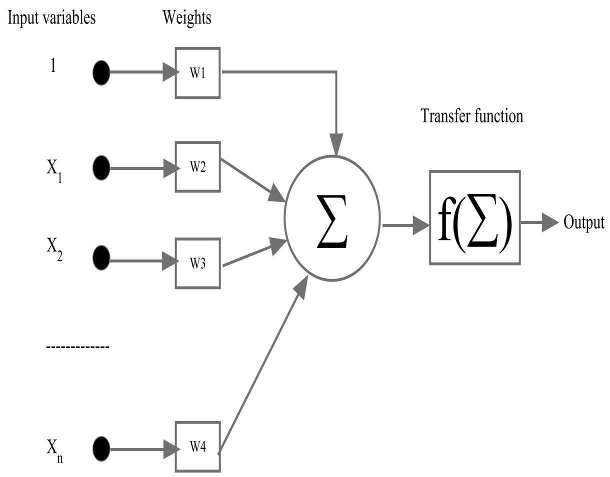

Fig. 1. Visual Representation of a Neuron in an Artificial Neural Network.

Table 1. Pavement Structure Properties and Axle Load Configurations Considered in the Finite Element Simulation.

	Axle Load (kN)		
Single Axle-Dual Axle	20	50	120
Tandem Axle-Dual Axle	40	100	190
Tridem Axle- Dual Axle	80	13	250
Load Position	Edge	Corner	
Tire Pressure (kPa)	650	750	
Modulus of Subgrade Reaction (MPa/mm)	0.027	0.054	0.081
Base Course Thickness (mm)	200	300	
Base Course Modulus (MPa)	350	1000	

in Table 1. This small number of options was used to ensure a manageable number of cases were produced. The load cases were comprised of every possible combination of the input options.

The neural network input variables were chosen because they would have the most significant effect on the pavement response and also they corresponded to the likely geometries and loading conditions that engineers would want to analyze. The outputs to be predicted were the most likely to govern the design of rigid pavements and they are:

1. Maximum principal stresses
2. Maximum slab deflection
3. Maximum Horizontal Stress in X direction (σ_{xx})
4. Maximum Horizontal Stress in Direction (σ_{yy})
5. Maximum Shear Stress in XY plane (τ_{xy})
6. Maximum Shear Stress in YZ plane (τ_{xz})

Three axle configurations (Single axle-dual tire, tandem axle-dual tire and tridem axle-dual tire) were modeled in the finite element simulations. These axle loads represent the most common axles in the traffic spectrum. The axle loads were chosen to represent the average, upper and lower bound values for each configuration as stated in the New Zealand and Australian guidelines [7]. Two tire pressures (650 and 750 kPa) were considered in the analysis. The modulus of the subgrade reaction is used to model the subgrade stiffness. In this simulations, three types of subgrades: weak, moderate and strong were considered with the modulus of subgrade values 0.271, 0.0543 and 0.0814, respectively. Three slab thicknesses were considered in the finite elements simulation 200, 250 and 300 mm to represent thin, moderate and thick slabs respectively. Concrete slab properties were considered constant during the analysis in which the concrete modulus and Poisson ratio were assumed 28 GPa and 0.2, respectively. Two base course thicknesses were also considered 200 and 300 mm. The bases course moduli ranges from 350 MPa to 1000 MPa to represent unbound granular base and cement stabilized bases courses respectively. Total number of analysis combinations used equals 3*3*2*3*2*3*2*2=1296 combinations.

Slab Geometry and Loading Positions

The slab used in the finite element analysis was assumed to be 10 m long and 4.5 m wide. These values were used because 10 m is a typical length for a rigid pavement slab and 4.5 m accounts for 3.5 m lane and one meter of integrated shoulder. Three loading scenarios were considered to attain the worst stresses, strains and deflections. The edge and corner loading positions are the most critical loading locations as they produce the worst stresses and strains at the edge of the slab with the worst deflection is at the corner. Fig. 2 shows the mid span location near the edge loading.

Finite Element Results

All of the finite element analysis has been done using EverFE2.25 software; it was developed by the Universities of Maine and Washington [8]. EverFE2.25 has a simple graphic user interface that allows all model changes to be seen as they are applied. The software also allows complex slab geometries, load configurations, multi-layered foundations dowel-slab interactions and up to tri-linear temperature gradients through the slab. In order to get the required accuracy from the finite element analyses, a test case was used to determine how small the elements needed to be. This process involved reducing the element size until there was only a very small change, less than 1%, in the stresses and deflections. This element density was then used for all of the finite element analyses performed.

Network Training

In order for a neural network to most accurately model the given situation, it must be trained. During training, the weights and bias are updated based on how big the difference is between the target data and the current neural network prediction output. All of the neural network creation and training was done using MATLAB 2010b.

Selection of Optimal Weights

A neural network can be trained well enough that it will give nearly exact answers for the provided training data but in this case it will have a poor generalization capabilities [3]. This can be prevented by having a second set of data, a validation set of data that after each training step is put through the network and its mean squared error, MSE, found. As training progresses the MSE of both the training and validation data set decrease but whereas the training MSE will continue to decrease the validation MSE reach a minimum [9]. This minimum is reached when the network is at its optimal point to predict new cases.

It should be noted that over training of the artificial neural network will result in the network being very good at predicting the training cases but it will be very poor at predicting new cases. If training was allowed to continue then the neural network would be over fitted and this would result in the network very accurately predicting the training cases but would be very poor at predicting new cases.

Network Architecture Selection

As stated above there will be many outputs needed from the neural network and because of this the network will be large. In order to combat this problem, each output could have its own small neural network or the outputs could be split into smaller related categories with each category having a network.

Having a network with two or more output variables may result in difficultly during the learning process. This is because the neurons are trying to model multiple relationships at once. This can become a problem if one or more of the relationships are dominate over the rest. If this problem does occur the best solution is to create many networks with one output variable each and later combine them into a one large unit [10]. That makes the selection of the number of neuron in the output layer easy but it still leaves the problem of how

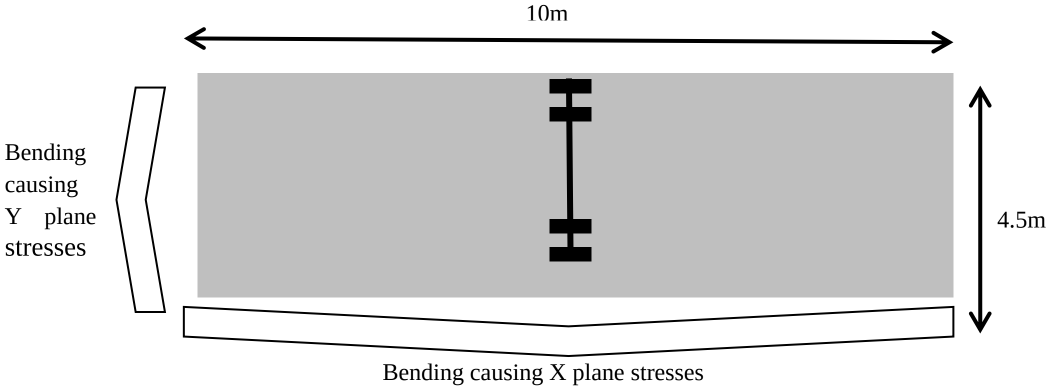

Fig. 2. The Slab Geometry and Direction of Normal Stresses.

Table 2. The Mean Squared Error for the Top Ten Architectures and Their Respective Performance Criteria.

ANN Architecture	Overall MSE	Training MSE	Validation MSE	Testing MSE	Training Plus Testing	Training Less Testing	Sum of Ranks
12:07	6.08×10^{-3}	5.55×10^{-3}	4.05×10^{-3}	8.88×10^{-3}	1.44×10^{-2}	3.33×10^{-3}	5849
08:14:07	6.21×10^{-3}	5.74×10^{-3}	4.10×10^{-3}	8.92×10^{-3}	1.47×10^{-2}	3.17×10^{-3}	5851
08:16:13	6.35×10^{-3}	5.90×10^{-3}	4.31×10^{-3}	8.95×10^{-3}	1.49×10^{-2}	3.05×10^{-3}	5932
09:13	6.56×10^{-3}	6.17×10^{-3}	4.49×10^{-3}	8.98×10^{-3}	1.51×10^{-2}	2.80×10^{-3}	5937
11:09:13	6.46×10^{-3}	6.04×10^{-3}	4.44×10^{-3}	8.98×10^{-3}	1.50×10^{-2}	2.94×10^{-3}	5957
08:15:14	6.25×10^{-3}	5.78×10^{-3}	4.09×10^{-3}	8.96×10^{-3}	1.47×10^{-2}	3.18×10^{-3}	5962
07:20:08	6.75×10^{-3}	6.38×10^{-3}	4.89×10^{-3}	8.99×10^{-3}	1.54×10^{-2}	2.61×10^{-3}	5992
12:13:14	6.02×10^{-3}	5.45×10^{-3}	3.99×10^{-3}	8.93×10^{-3}	1.44×10^{-2}	3.48×10^{-3}	6005
10:11:15	6.29×10^{-3}	5.81×10^{-3}	4.19×10^{-3}	8.99×10^{-3}	1.48×10^{-2}	3.17×10^{-3}	6013
14:10:10	6.67×10^{-3}	6.25×10^{-3}	4.85×10^{-3}	9.02×10^{-3}	1.53×10^{-2}	2.77×10^{-3}	6054

many hidden layers are needed and the number of neuron in each layer. The method used in this paper was outlined by Manica *et al*. [11] and is stated below.

This is done by selecting a range of network architectures and training each one multiple times. To ensure consistency of the training, the same training and validation data is used for all tests. Once each architecture has been optimized, the average mean squared error (MSE) for each training run is found for the training and validation data as well as the standard deviation. Once that has been done, a third data set, the testing set, is used to find the two performance criteria of the network, conforming and generalization capability. The conforming capability is given by the sum of the MSE for the training and testing data sets. Each architecture is given a rank of one, the best, to k, number of different architectures tried, based on increasing conforming capability. The generalization capability of a network is found by subtracting the training MSE from the testing MSE. The same ranking system is used to rank the generalization capability of the networks and the two ranks are then summed to get the total architecture ranking. The network architecture with the lowest rank is then chosen.

Having a large number of neurons in a network can have the same effect as overtraining [3]. This is why all of the work cited here that relate to neural network use have no more than three hidden layers and twenty neurons per layer. This can help with the choosing which network architectures are worth trying and which can be discarded straight away. This method leads to a network that is good at providing accurate solutions to previously unseen problems and also one that accurately predicts problems that was presented during training.

Results and Discussion

Each Architecture was trained and tested ten times and the average of the mean squared errors was found and the performance criteria were calculated as outlined above.

Architecture Selection

A range of architectures were selected from a single neuron up to three layers of twenty neurons each and this resulted in 8420 different network architectures being tried. Each architecture was trained ten times and the mean of the MSEs found and Table 2 shows the results for the top ten ranked network layouts. The top ten network architectures all performed very well and this is shown by the fact that the top ten have a difference in rank of 205 whereas the ranks range from 5849 up to 16840. From this analysis, it was found that a network with a 12:7 architecture performed the best. A 12:7 network architecture means that there will be two hidden layers with twelve neuron in the first layer and seven in the second.

Network Performance

The final network to be used was found by training the 12:7 architecture 1000 times and the same performance criteria used in architecture selection was used here to pick the best trained network. The performance of the final network was calculated with respect to each output separately and the results were compared with the finite element simulation results. Fig. 3 shows the relationship between the artificial neural networked predicted principal stresses and the finite elements simulation results. It is clear from Fig. 3 that the predicated and calculated results lined up on the equality line with the MSE of 0.0012.

Similarly, Figs. 4 to 6, compare the predicted maximum slab surface deflection, maximum horizontal stresses σ_{yy} and σ_{xx} with the finite elements simulated values. The predicted and calculated values matches each other very closely with the maximum mean squared of error (MSE) of 0.0021 MPa in the stresses in x direction as shown in Fig. 6.

In general, the neural network performed well for the max principal stresses, normal stresses in X and Y planes, and maximum slab deflections with the largest MSE of the four outputs being 0.0021 as shown in Figs. 3 to 6 and Table 3. This accuracy is also shown in the coefficient of determination between the data and the line of equality being very close to unity.

The neural network did not perform as well when predicting the shear stress τ_{xy} and τ_{yz} with the mean squared error for the shear stress in XY plane and YZ plane being 0.028 and 0.0072, respectively. Figs. 7 and 8 show relationship between the predicted shear stresses τ_{xy} and τ_{yz} and the calculated shear stresses using the finite element simulations. The large errors for the shear stresses are part of the network and cannot be removed by further or repeated training. This is a result of the neural network trying to learn to

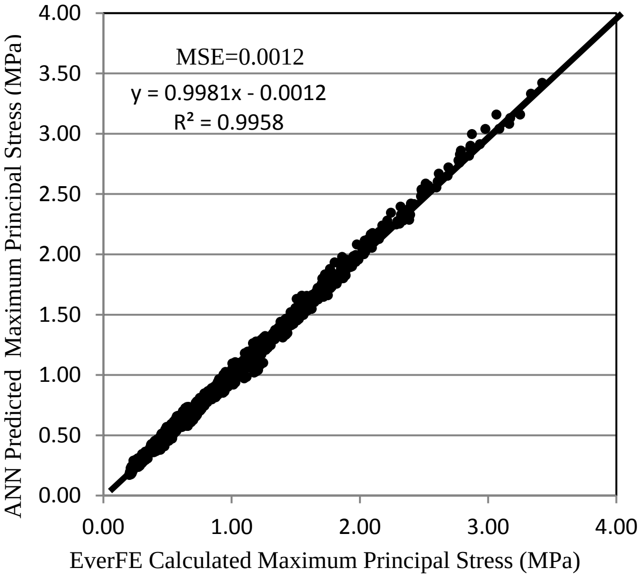

Fig. 3. ANN Predicted Versus Calculated Maximum Principal Stress.

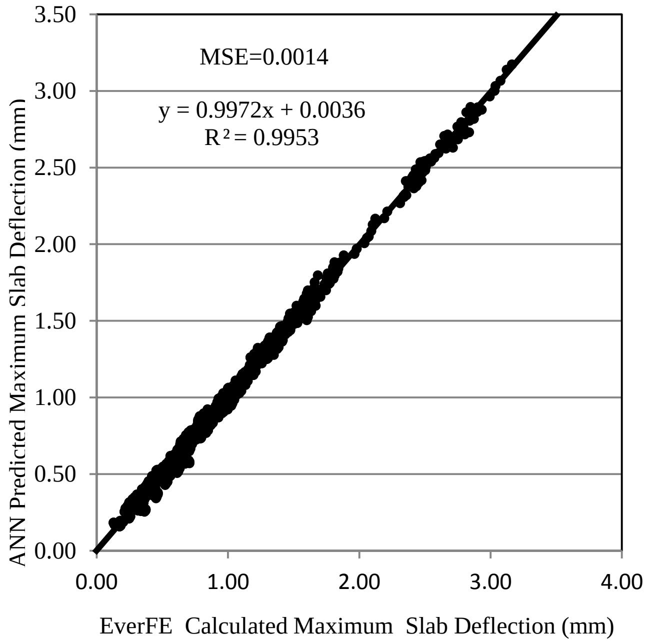

Fig. 4. ANN Predicted Versus Calculated Maximum Slab Deflection.

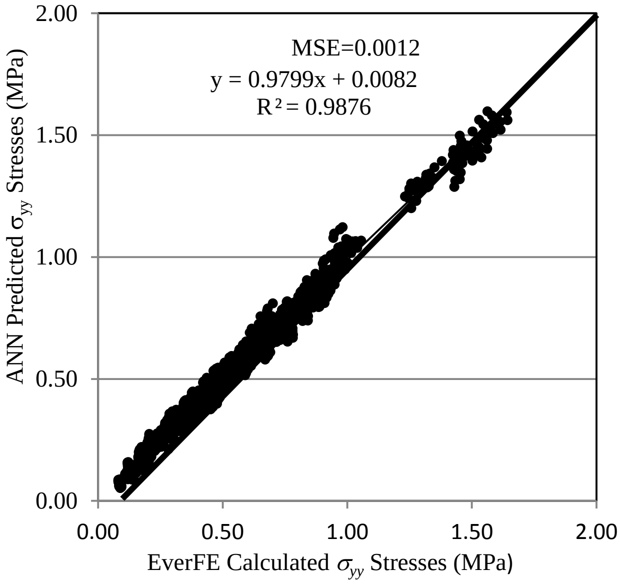

Fig. 5. ANN Predicted Versus Calculated Normal Stresses σ_{yy}

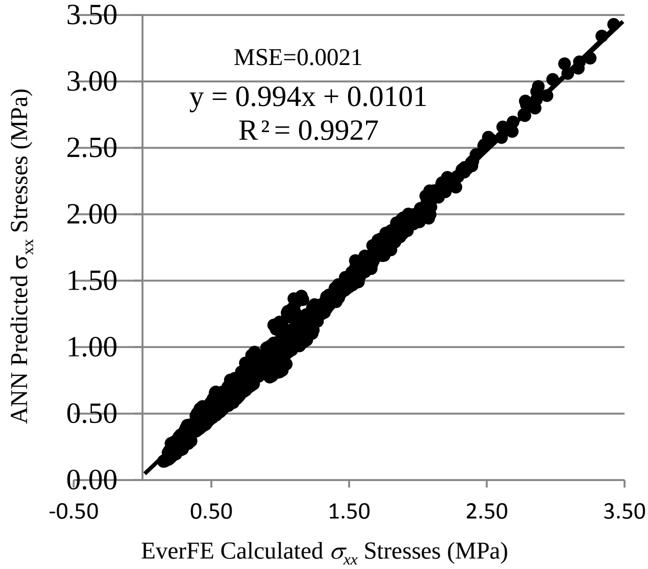

Fig. 6. ANN Predicted Versus Calculated Normal Stresses σ_{xx}.

Table 3. The Mean Squared Error and Coefficient of Determination for Each output of the Artificial Neural Network.

	Max Principal Stress	Max Deflection	X Plane Stress	Y Plane Stress	XY Plane Shear Stress	YZ Plane Shear Stress
Mean Squared Error	0.0012	0.0014	0.0021	0.0012	0.0228	0.0072
Coefficient of Determination	0.9958	0.9953	0.9926	0.9875	0.8564	0.8668

predict a large number of different patterns with different levels of dominance. During training the dominant patterns are learnt the fastest and other patterns that end up being modeled accurately are due to the patterns being similar in nature to the dominant ones. In this situation the patterns required to model the shear stresses are minor and as such would require a far more training than would produce a neural network that is both accurate and good at predicting unseen cases. In order to correct this, the shear stresses would need to have their own neural network where the patterns required to model them would be dominate. This would also have an added side effect of increasing the performance of the first neural network due to the removal of the minor patterns that had to be learnt. However, the shear stresses are not part of any current rigid pavement design and therefore limiting the artificial neural network for predicting deflections, normal and principal stresses will provide all the necessary responses for rigid pavement design and analysis.

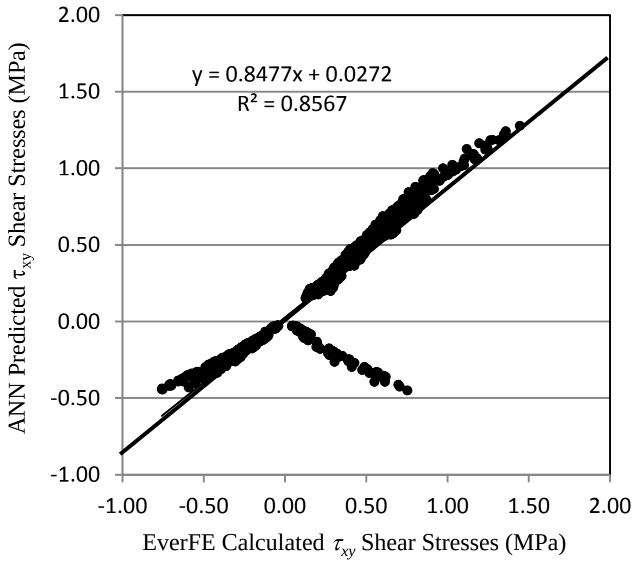

Fig. 7. ANN Predicted Versus Calculated Shear Stresses τ_{xz}.

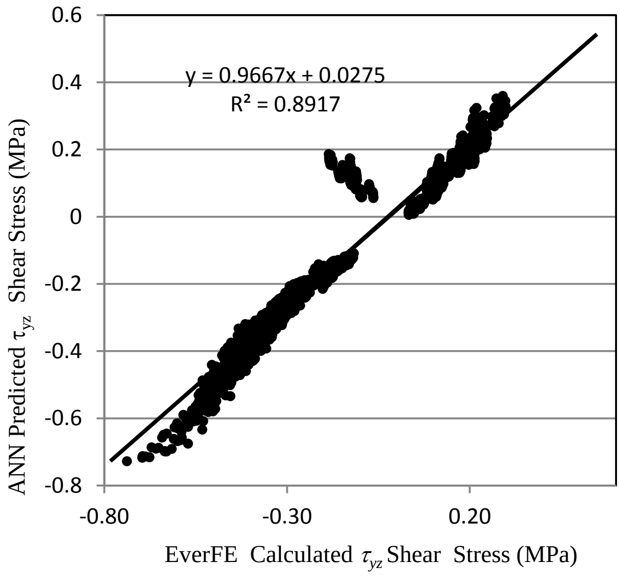

Fig. 8. ANN Predicted Versus Calculated Shear Stresses τ_{yz}.

Time Requirements

Using the artificial neural network results in a very large decrease in the time taken to perform the analysis when compared to finite element analysis. This is due to the artificial neural network being able to predict a large number of cases in a very short time whereas with a finite element analysis will take a minimum of a minute for each case needing analysis. This will result in the designer being able to analyze a larger number of cases compared to finite element analysis and as such it is less likely that the critical case will be missed due to time constraints.

It is also important to note that obtain accurate results all inputs must be within the range of the input data used in the training. On the other hand, using the neural network used outside the training data range will result in inaccurate results and therefore the extrapolation errors are made worse by the transfer function used to pass data between layers in the neural network. The tan sigmoid function, as shown in Fig. 9, has asymptotes at -1 and 1 which can result in very similar outputs for vastly different inputs. If the input values were to produce a large summation within the neuron then when the transfer function is applied a number very close to that outputted from vastly different inputs. The inaccuracies that result are inherent in the neural network and this is the reason why inputs outside the range of the training data can result in very inaccurate predictions. Before the data is passed to the neural network it is scaled to be between negative one and positive one based on the maximums from the training data. This helps to reduce the chances that a large summation within the neuron will occur. But it can still happen if inputs outside the range of the training data are used

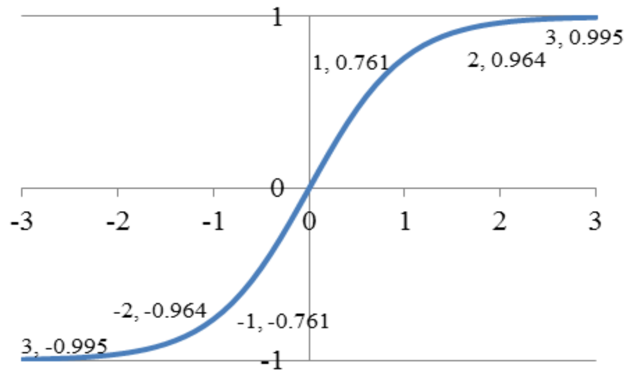

Fig. 9. A Tan Sigmoid Transfer Function with the Data Point Labelled.

Conclusions

The prediction of rigid pavement responses can accurately be predicted by finite element analysis. However, to perform an analysis or design using traffic spectra and carry out damage analysis for each axle load for each pavement trial, it will take significant computational effort and time. In addition, practitioners in the pavement industry will not favor any analysis that will require specialized expertise such as finite elements analysis. An artificial neural network is a quick and easy alterative that does not require much expertise to achieve accurate results. The neural network proved to be very accurate at predicting the required outputs but it did fall short in predicting shear stresses. However, this will not be a problem as typically shear stresses are not required input for the current rigid design methods.

Further research can be done on this topic to improve the accuracy of the artificial neural networks predictions and expand its capabilities so as to make it more attractive to practitioners. In order to do this, the finite element results would need to be verified experimentally so as to ensure the neural network has realistic data to predict. In addition, to improve the accuracy of the predictions, large number of training cases would be required and maybe several neural networks with a smaller number of outputs would be produced. The number of inputs could be expanded to include the slab size and modulus to make the neural network more versatile and as such more likely to be adopted by departments of transportations.

Acknowledgment

The author would like to acknowledge Mr. Michael Cullum for his dedicated work in producing the results of the finite elements simulations and ANN work analysis.

References

1. Huang, Y.H. (1993). *Pavement Analysis and Design*, Prentice-Hall, Englewood Cliffs, NJ, USA.
2. Lacroix, A., Kim, Y.R., and Ranjithan, S.R. (2008). Backcalculation of Demonic Modulus from Resilient Modulus of Asphalt Concrete with an Artificial Neural Networks, *Transportation Research Record*, No. 2057, pp. 107-113.
3. Coleri, E., Guler, M., Gungor, A.G., and Harvey, J.T. (2010). Prediction of Subgrade Resilient Modulus Using Genetic Algorithm and Curve-Shifting Methodology, *Transportation Research Record*, No. 2170, pp. 64-73.
4. Far, M.S., Kim, Y.R., Ranjithan, S.R., and Underwood, B.S. (2009). Application of Artificial Neural Networks for Estimating Dynamic Modulus of Asphalt Concrete, *Transportation Research Record*, No. 2127, pp. 173-186.
5. Kim, M., Burton, M., Prozzi, J., and Murphy M. (2014). Maintenance and Rehabilitation Project Selection Using Artificial Neural Networks, *93rd Annual Meeting*, Transportation Research Board, Washington, DC, USA.
6. Mirzahosseini, M., Najjar, Y., Hossein G., and Hossein A. (2013). ANN-Based prediction model for rutting propensity of Asphalt Mixtures, *92nd Annual Meeting*, Transportation Research Board, Washington, DC, USA.
7. NZTA. (2007). New Zealand Supplement to the Document: Pavement Design – A Guide to the Structural Design of Road Pavements (Austroads, 2004), Wellington, New Zealand.
8. Davids, W. (2008). *EverFE 2.23*, Department of Civil and Environmental Engineering, University of Maine, Orono, Maine, USA.
9. Dandy, G.C. and Maier, H.R. (1998). Understanding the Behaviour and Optimising the Performance of Back-Propagation Neural Networks: An Empirical Study, *Environmental Modelling and Software*, 13, pp. 179-191.
10. Statsoft, Inc. (2011). *Electronic Statistics Textbook*, Tulsa, Oklahoma, USA.
11. Manica, M., Sabharwallb, P., Utgikarc, V., and Wijayasekaraa, D. (2011). Optimal Artificial Neural Network Architecture Selection for Performance Prediction of Compact Heat Exchanger with the EBaLM-OTR Technique, *Nuclear Engineering and Design*, 241(7), pp. 2549–2557.