ARTICLES

Neural Networks for Survey Researchers

Adam Eck¹

¹ Computer Science Department, Oberlin College

Keywords: neural networks

<https://doi.org/10.29115/SP-2018-0002>

Survey Practice

Vol. 11, Issue 1, 2018

Neural networks are currently one of the most popular and fastest growing approaches to machine learning, driving advances in deep learning for difficult real-world applications ranging from image recognition to speech understanding in personal assistant agents to automatic language translation. Although not yet as commonly employed in survey research as other types of machine learning, neural networks offer natural extensions of well-known linear and logistic regression techniques in order to learn non-linear functions predicting or describing nearly any real-world process or problem (provided there are sufficient data and an appropriate set of parameters). Moreover, neural networks offer great potential towards more intelligent surveys in the future (e.g., adaptive design tailored to individual respondents' characteristics and behavior, automated digital interviewers, analysis of rich multimedia data provided by respondents). Neural networks can learn for both regression and classification tasks without requiring assumptions about the underlying relationships between predictive variables and outcomes. In this article, we describe what neural networks are and how they learn (with tips for setting up a neural network), consider their strengths and weaknesses as a machine learning approach, and illustrate how they perform on a classification task predicting survey response from respondents' (and nonrespondents') prior known demographics.

What are Neural Networks and How are They Constructed?

Neural networks (also known as artificial neural networks, ANN) are one of the most popular approaches for regression and classification modeling in the machine learning literature in terms of theoretical research and application. For example, neural networks have achieved great success in tasks such as image recognition (e.g., Krizhevsky, Sutskever, and Hinton 2012); optical character and handwriting recognition (e.g., Graves and Schmidhuber 2008); and natural language translation (Sutskever, Vinyals, and Lee 2014). Elsewhere, Google DeepMind's recent AlphaGo program (Silver et al. 2016) used neural networks in part to defeat the world expert in a game of Go, largely considered to be one of the most computationally difficult games to win due to its exceedingly large number of possible board configurations. Indeed, neural networks are behind the recent explosive growth of **deep learning** (LeCun, Bengio, and Hinton 2015), where multiple layers of learners are stacked together, each learning a more abstract representation to aid in the overall prediction. For instance, in an image recognition task where the computer needs to classify what types of objects are in an image, one layer might learn where the lines are within an image, whereas another might learn how those lines organize to represent different shapes, and then how shapes organize to represent objects (e.g., books vs. people vs. pets).

Neural networks are a form of supervised learning that are inspired by the biological structure and mechanisms of the human brain. Neural networks generate predictions using a collection of interconnected nodes, or neurons, that are organized in layers. The first layer is called the *input layer* as the neurons in this layer only accept variables from the data set as input. The final layer is called the *output layer*, since it outputs the final prediction(s). *Hidden layers* refer to those that fall between the input and final layers since their outputs are relevant only *inside* the network. Neurons create a weighted sum from the input they receive and then transform this weighted sum using some type of nonlinear function such as the logit, hyperbolic tangent, or the rectified linear function. The value computed from the function, operating on the weighted sum, is then passed to neurons in the next layer of the network. Information flows through the neural network in one direction — from the input layer through the hidden layers to the output layer. Once information reaches the output layer, it is gathered and converted to predictions.

Depending on the complexity of the data for which predictions are desired, neural networks may have several hidden layers, and the functions used within the neurons may vary. Theoretically, with enough neurons, one hidden layer of neurons plus an output layer is enough to learn any binary classification task, and two layers plus an output layer can learn a close approximation for any regression task (e.g., Alpaydin 2014, 281–83). In the increasingly popular domain of deep learning, tens of hidden layers might be used to help aid in the discovery of complex, predictive patterns in the data. The overall architecture of the neural network determines which neurons feed their output to other neurons in subsequent layers. Furthermore, the type of variable being predicted governs the number of neurons in the output layer. In particular, for regression tasks (continuous outcomes) and binary classification tasks (dichotomous outcomes), the final output layer consists of a single neuron. Alternatively, for multinomial classification tasks, where there are more than two values in the categorical outcome variable, the final output layer consists of one neuron per possible value. In this case, the predicted class corresponds to the neuron with the highest outputted value. An example neural network is displayed in Figure 1. The network displayed is a so-called “fully-connected network” because each neuron, within each layer, provides input into each neuron in the next layer. We also see from the figure that there is only one hidden layer in the network. Figure 2 provides a more technical description of how neural networks are created and Table 1 highlights a few popular R packages for constructing them.

Advantages and Disadvantages of Neural Networks

One of the most appealing aspects of neural networks is their ability to perform complex classification tasks with high levels of accuracy. Neural networks can improve the results of more traditional classification models, such as logistic regression, by combining the results of multiple models across the layers of the network. The improvements to accuracy do have a trade-off in that neural

Figure 1 An example of a fully connected neural network with a single hidden layer. The network takes various demographic variables as input (on the left) to predict a binary response (on the right).

Figure 2 Steps in constructing a neural network.

networks can take more computing time before a final prediction is made. We highlight other major advantages and disadvantages of neural networks in Table 2.

Table 1 Popular packages for implementing neural networks in R.

R Package Name	Brief Description
nnet	This package provides support for feed-forward networks with a single hidden layer. It can minimize either the sum of squares error or cross-entropy as its objective function (when finding a good set of weights for each neuron during training).
neuralnet	This package provides support for feedforward networks with any number of hidden layers. It contains multiple variants of the backpropagation algorithm for training, allows the user to choose different activation functions for the hidden neurons (e.g., logistic and hyperbolic tangent), and also can minimize either sum of squares error or cross-entropy as its objective function. Functions are also provided for visualizing the network after training.
mxnet	This advanced package provides access to the popular MXNet Scalable Deep Learning framework in R, which can be used to create standard or deep feed-forward networks, as well as advanced models such as recurrent (for sequential data) and convolutional (for spatially related data such as images) neural networks. Support is also provided for training networks with video cards (also known as graphics processing units, GPUs) in order to speed up training.

Table 2 Additional advantages and disadvantages of neural networks.

Major advantages of neural nets	Major disadvantages of neural nets
By combining multiple layers, instead of considering only a single logistic regression function, for example, neural networks are able to learn non-linear separations between the different categories of prediction and are capable of learning very complex concepts and patterns (e.g. image recognition) that are often too difficult for other machine learning approaches.	Neural networks are relatively opaque "black boxes". Because neural networks are created using a large number of different weights learned between the different neurons, combined with their separation across layers, it can be incredibly difficult for a human to interpret how any given prediction was made.
Neural networks are quite robust at handling noisy data.	Relatedly, because neural networks rely on subsequent learning across many layers, it is also difficult to determine what importance each input variable has on the eventual prediction. What could be a significant predictor in the input layer may be down-weighted in a subsequent layer, for example.
Neural networks are nonparametric methods and do not require distributional assumptions or model forms to be specified prior to their construction.	Depending on the complexity of the predictive task, the neural network can require extensive training. In some cases this could mean larger amounts of data are required to apply them.
Neural networks are extendable in that they can be stacked together to learn more complex abstractions to aid in prediction, as described above with respect to "deep learning".	Neural networks may require greater computational resources and time compared to other machine learning methods.

How Have Neural Networks Been Used in Survey Research?

Neural networks are emerging as a useful model for a variety of tasks in survey research literature. For instance, Gillman and Appel (1994) describe the use of neural networks for automated coding of response options (e.g., occupation coding). Nin and Torra (2006) consider the use of neural networks for record linkage (an increasingly relevant task for survey research in the era of Big Data), focusing on cases where different records contain different variables. An advanced type of neural network called recurrent neural networks (RNNs) allow for connections between neurons within a hidden layer that enable RNNs to remember information over time, making them highly useful for sequential data. Eck et al. (2015) have considered the use of RNNs to predict whether a respondent will break off from a Web survey, based on the respondents' behaviors exhibited in paradata describing their actions within the survey (e.g., navigational patterns between questions and pages, answering and reanswering questions, and scrolling vertically on a page). Recently, these models have also been extended to predict errors at the question level, including predicting whether a respondent will commit straight-lining on a battery of grid questions (Eck and Soh 2017).

Deep learning with neural networks also offers much promise in supporting and augmenting survey-based data collection. For example, sequence-to-sequence models using neural networks (e.g., Sutskever, Vinyals, and Lee 2014) could enable the automated translation between languages spoken by an interviewer and a respondent, removing barriers for data collection from underrepresented populations. Similarly, image segmentation models using convolutional neural networks (e.g., He et al. 2017) could be used to identify objects within smartphone images uploaded by respondents as answers to survey questions (e.g., in food diary surveys) and providing background context paradata to their responses.

Classification Example

Using the National Health Interview Survey (NHIS) example training dataset, we estimated both a main effects logistic regression model and a collection of neural network models for predicting survey response based on a collection of demographic variables. To illustrate how the performance of neural networks can depend on their internal parameters and structure, we constructed a number of neural networks that vary in both the number of neurons used in a single hidden layer¹ [2, 5, 10, 20, 50, 100], and the number of training iterations ranging from 10 to 200 in increments of 10.

¹ We also experimented with using two and three hidden layers with equal numbers of neurons, but the final predictive performances were similar to those reported for one layer.

Table 3 Final predictive performance of neural networks and logistic regression.

Hidden Neurons	Accuracy	Balanced Accuracy	Sensitivity	Specificity
2	0.7788	0.745	0.5785	0.9115
5	0.7872	0.7585	0.6171	0.8999
10	0.79	0.7604	0.6148	0.9061
20	0.789	0.7627	0.6329	0.8925
50	0.7874	0.7633	0.6446	0.882
100	0.7825	0.759	0.6429	0.875
Logistic Regression	0.6982	0.6642	0.4965	0.832

For both prediction tasks, the variables used as input to the neural networks include the respondent's (1) region, race, education, class of worker, telephone status, and income categorical variables, which were converted into multiple inputs using one-hot coding; (2) Hispanic ethnicity and sex dichotomous variables; and (3) age and ratio of income to the poverty threshold continuous variables that were normalized to a Z score. For evaluating the models (both neural network and logistic regression techniques), 84% of the data was randomly selected for training the models, whereas the remaining 16% was held back as an independent testing data for evaluating the accuracy of the predictions.

The results of our models are presented in Figure 3 and Table 3. From these results, we can make several key observations. First, each of the neural networks with different numbers of neurons in the hidden layer were able to achieve significantly higher accuracy, sensitivity, and specificity after a sufficient number of training iterations than a logistic regression model that trained on the same data. Remarkably, even a neural network with only two neurons in the hidden layer achieved much better performance than the logistic regression, which is notable given that logistic regression is equivalent to a neural network with one neuron. Thus, even a small increase in the complexity of the model can greatly improve predictive performance.

Next, we consider the effects of increasing the complexity of the model, as measured by the number of neurons. We observe that with respect to balanced accuracy (Figure 3b, Table 3), which best² measures combined performance on both positive (response) and negative (nonresponse) data points, that

increasing the number of neurons generally led to higher predictive performance. Thus, the additional neurons improved the ability of the neural network to learn more nuanced patterns within the data, increasing the ability of the model to differentiate respondents who would ultimately respond versus those who would not. In particular, this result was caused by the networks with

² This is important since there is an imbalance between the two classes, with nonresponse data points making up 60% of the data set.

Figure 3 Predictive performance across training iterations.

more neurons achieving higher sensitivity (Figure 3c, Table 3). This is notable since this implies that the additional neurons were valuable for improving predictions of the less common response outcome, which is indeed more difficult to predict given that there were fewer data points with this outcome from which to learn. However, we also note that the neural network with 50 neurons in the hidden layer slightly outperformed the one with 100 neurons. This could indicate that the model started to become too complex and was beginning to overfit the training data, reducing the generalizability of the patterns it learned.

Finally, considering the number of training iterations, we make two key observations. First, for all of the neural networks, a small number of iterations were needed to outperform logistic regression. Thus, training a neural network model does not necessarily require significantly more computational work than a logistic regression model in order to achieve significant improvements in predictive performance. Second, as the number of neurons in the network increased, a larger number of iterations were required for the model to converge to its best performance. This highlights one of the key trade-offs in neural networks: performance vs. time. That is, the more complex models achieved greater performance at the expense of requiring more time to learn the final, stable model. For smaller problems, such as the data set presented here, the added time expense of increased complexity is relatively small, but for more difficult problems (e.g., with millions of data points or more and with a larger number of possible predicted outcomes), care is often needed to optimally balance this trade-off.

REFERENCES

- Alpaydin, E. 2014. *Introduction to Machine Learning*. 3rd ed. Cambridge, MA: MIT Press.
- Eck, A., and L.K. Soh. 2017. “Sequential Prediction of Respondent Behaviors Leading to Error in Web-Based Surveys.” In *The 72nd Annual Conference of the American Association for Public Opinion Research*. New Orleans, LA, May 18–21, 2017.
- Eck, A., L.K. Soh, A.L. McCutcheon, and R.F. Belli. 2015. “Predicting Survey Outcomes Using Sequential Machine Learning Methods.” In *The Annual Conference of the Midwest Association for Public Opinion Research*. Chicago, IL, November 20–21, 2015.
- Gillman, D.W., and M.V. Appel. 1994. “Automated Coding Research at the U.S Census Bureau.” *U.S. Census Bureau Research Report*, no. 4.
- Graves, A., and J. Schmidhuber. 2008. “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks.” In *Advances in Neural Information Processing Systems 21*, edited by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, 545–52. Curran Associates, Inc.
- He, K., G. Gkioxari, P. Dollar, and R. Girshick. 2017. “Mask R-CNN.” 2017. <https://arxiv.org/abs/1703.06870v2>.
- Krizhevsky, A., I. Sutskever, and G.E. Hinton. 2012. “ImageNet Classification with Deep Convolutional Neural Networks.” In *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, 1097–1105. Curran Associates, Inc.
- LeCun, Y., Y. Bengio, and G.E. Hinton. 2015. “Deep Learning.” *Nature* 521 (7553): 436–44.
- Nin, J., and V. Torra. 2006. “New Approach to the Re-Identification Problem Using Neural Networks.” In *Modeling Decisions for Artificial Intelligence. MDAI 2006. Lecture Notes in Computer Science 3885*, edited by V. Torra, Y. Narukawa, A. Valls, and J. Domingo-Ferrer, 251–61. Berlin, Heidelberg: Springer.
- Silver, D., A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Den Driessche, J. Schrittwieser, et al. 2016. “Mastering the Game of Go with Deep Neural Networks and Tree Search.” *Nature* 529 (7587): 484–89.
- Sutskever, I., O. Vinyals, and Q.V. Lee. 2014. “Sequence to Sequence Learning with Neural Networks.” In *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, N. Cortes, D. Lawrence, and K.Q. Weinberger, 3104–3112. Curran Associates, Inc.

SUPPLEMENTARY MATERIALS

R-Code Example

Download: <https://www.surveypractice.org/article/2714-neural-networks-for-survey-researchers/attachment/9429.zip>