American Journal of Epidemiology, 2025, **194**, 1473–1475
https://doi.org/10.1093/aje/kwae380
Advance access publication date October 2, 2024
The *AJE* Classroom

A primer on neural networks

Paul N. Zivich*[,1] and Ashley I. Naimi[2]

[1]Department of Epidemiology, UNC Gillings School of Global Public Health, Chapel Hill, NC, United States
[2]Department of Epidemiology, Rollins School of Public Health, Emory University, Atlanta, GA, United States
*Corresponding author: Paul N. Zivich, Department of Epidemiology, Gillings School of Global Public Health, UNC Campus Box 7435, Chapel Hill, NC 27599-7435 (zivich.5@gmail.com)

Introduction

Neural networks are a class of machine-learning algorithms inspired by the structure of the nervous system.[1] An important result underlying the utility of neural networks is the universal approximation theorem, which states that there exists a sequence of neural network that can closely approximate any smooth function. Practically, this result means neural networks provide a way to more flexibility model data relative to tools more familiar to epidemiologists (eg, generalized linear models [GLMs]). Despite their popularity, how neural networks operate can be mysterious. Here, we review the basics of neural networks.

Motivating problem

Consider predicting high-density lipoprotein cholesterol (ie, Y) given the following covariates: age (years; X_1), gender (male, female; X_2), body weight (kilograms; X_3), and height (centimeters; X_4). Data from the National Health and Nutrition Examination Survey 2017-2018 are used and restricted to complete cases 18 years or older ($n = 5092$).

Neural networks

Three components define a neural network: structure, activation functions, and loss function.

Structure

A neural network is a graph consisting of nodes (or neurons) connected by directed edges. Nodes are nested within layers, with layers being divided into an input layer, hidden layer(s), and an output layer. The input layer consists of the predictors (ie, X_1, X_2, X_3, X_4). Hidden layers are intermediates between the input and output. The structure of the hidden layers (eg, number of nodes, number of hidden layers) is specified by the analyst. Finally, the output layer returns the predicted outcomes. Here, the output layer is a single node. Neural networks are often referred to by their number of hidden layers (eg, a 3-layer neural network has 2 hidden layers, with the output being the third layer). Each edge in a network is associated with a "weight."

Inputs flow through the layers (Figure 1). Inputs from nodes that directly point to a node are combined using a weighted sum, using the edge's associated weight. To this weighted sum, another value called "bias" is added. This value is then transformed via an activation function (discussed next). For intuition, each node can be thought as a GLM, where the inputs from the previous layer are the independent variables, bias is the intercept, weights are the coefficients, and the activation function is the link function. The transformed value becomes the input to the next layer. Therefore, one can view a neural network as a multistage regression model, where regression models feed into other regression models. Because weight and bias have a multitude of meanings in epidemiology, these are referred to as *parameters* hereafter.

Activation functions

Activation functions are akin to the link function in a GLM. Some activation functions familiar to epidemiologists include the identity and logistic (ie, sigmoid) functions. However, there is a variety of activation functions without a GLM equivalent, such as the rectified linear unit (Appendix S1). Selection of an activation function for the output layer shares a similar motivation to the GLM link functions (ie, using a logistic function for a binary outcome), but the motivation for hidden layers differs. Hidden layers use nonlinear activation functions. By using nonlinear activation functions, a neural network can be used to model nonlinear relationships between predictors and the outcome without the analyst having to explicitly specify those relationships.

Loss function

The loss function is used to judge the quality of predictions. Specifically, the discrepancy between predicted and observed outcomes is summarized as a single value (Appendix S2). In general, the smaller the value, the better the predictions. Two common loss functions are mean squared error (MSE), often used for continuous outcomes, and cross-entropy, often used for categorical outcomes.

Fitting a neural network

Once the 3 components are determined, the parameters that minimize the loss function can be searched for (referred to as "training"). This process is like maximum likelihood estimation (MLE), where the parameters of a model are estimated by maximizing the likelihood function (equivalent to minimizing the negative likelihood function). Like MLE, a variety of algorithms

Received: April 17, 2024. **Accepted:** September 19, 2024

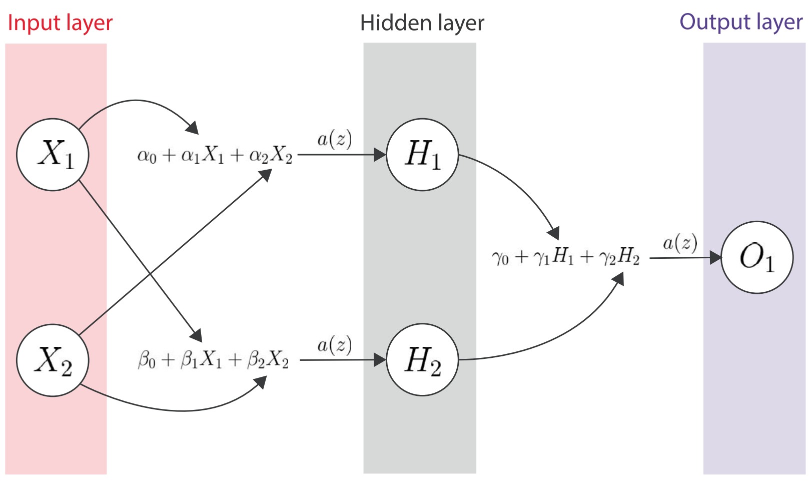
Input layer
Hidden layer
Output layer
X_1
X_2
$\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2$
$\beta_0 + \beta_1 X_1 + \beta_2 X_2$
$a(z)$
H_1
H_2
$\gamma_0 + \gamma_1 H_1 + \gamma_2 H_2$
O_1

Figure 1. Schematic diagram of information flow through a neural network. The variable α, β, and γ are parameters of the neural network; and $a(z)$ represents the activation function. Note that the intercept or bias term is not traditionally depicted as a node or with arrows in neural networks. Therefore, the number of parameters in a neural network is the $(N_I + 1) + \sum_{h \in H} (N_h + 1)$, where N_I is the number of nodes in the input layer, H is the set of different hidden layers, and N_h is the number of nodes in the h hidden layer.

can be used. However, neural networks face challenges that are less prominent in MLE.

First, there are often multiple minima for a given neural network (referred to as a nonconvex loss function). To see why, consider a set of parameters that minimized the loss function for the neural network in Figure 1, denoted as $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}$. If we swap $\hat{\boldsymbol{\alpha}}$ with $\hat{\boldsymbol{\beta}}$ (where $\hat{\boldsymbol{\alpha}} \neq \hat{\boldsymbol{\beta}}$), and $\hat{\gamma}_1$ with $\hat{\gamma}_2$, the reversed parameters result in the same loss. Therefore, at least 2 minima exist. Worse, some minima can be local (ie, smaller than other nearby parameters) instead of global (ie, smaller than all other feasible parameter values). Second, fitting neural networks with many parameters is computationally complex. Third, neural networks tend to overfit to the training data, particularly those with complex architectures. To overcome these issues, standard practice is to use algorithms that are less liable to become stuck in local minima and to randomly drop parameters or use out-of-sample predictions to reduce overfitting.[2] These techniques are provided in software such as Torch and TensorFlow.

Application

We now apply different neural networks to the motivating problem. Neural networks were built from scratch in Python, version 3.9.4, and R, version 4.1.0. All neural networks used the MSE loss function. For didactic purposes, we used the Nelder-Mead algorithm to estimate the parameters (not recommended in practice).

One-layer neural network

Consider how a linear regression model can be expressed as a 1-layer neural network, also referred to as a perceptron. This neural network consists of only input and output layers with the identity activation function (Figure 2A). After fitting this neural network (loss: 9.94×10^5), the intercept was 3.330 and the coefficients were 0.087, 11.899, −0.225, and 0.348 for age, gender, body weight, and height, respectively. When comparing with a GLM, the parameter estimates were equivalent.

Two-layer neural network

To highlight how neural networks are more flexible, consider the neural network in Figure 2B, which consists of 1 hidden layer and is fully connected (ie, all possible directed edges between layers are present). To allow for nonlinear relationships, nodes in the hidden layer used the leaky rectified linear unit activation function. The activation function for the output node was the

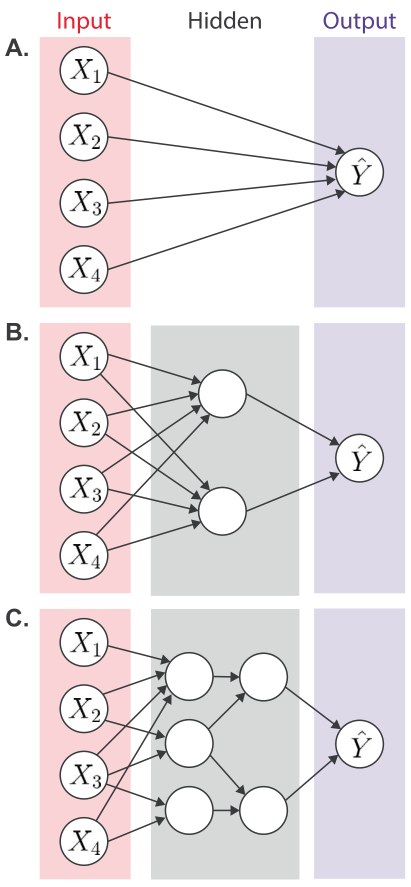

Figure 2. Neural networks used to compute predicted values for high-density lipoprotein cholesterol based on age, gender, height, and weight. A) A 1-layer neural network, also known as a perceptron, with 5 parameters. B) A 2-layer neural network with 13 parameters. C) A 3-layer neural network, also sometimes referred to as a deep neural network, with 20 parameters. Note that the intercept or bias term is implicit in these diagrams. In general, the intercept is not depicted by a node or with arrows.

identity function. When fitting this neural network with starting values for the minimization process drawn from **Uniform** $(-5, 5)$, the evaluated loss was 9.97×10^5 in Python and 1.05×10^6 in R. When using a different random draw, a smaller loss was observed (Python: 9.94×10^5; R: 1.00×10^6). This difference by starting values highlights the importance of using an optimization algorithm that can effectively search the parameter space.

Three-layer neural network

Finally, consider a neural network with 2 hidden layers that are not fully connected (Figure 2C). Neural networks consisting of many layers are also referred to as deep neural networks. Here, the first hidden layer prioritizes variations between body weight and gender, and body weight and height, because these might be thought to have important interactions. This example highlights that substantive knowledge can be encoded into the structure of a neural network. The evaluated loss was 9.73×10^5 in Python (0.98×10^6 in R), which was the best-performing neural network considered. However, the estimated parameters of either multilayer neural network may correspond to a local minimum, due to the fitting procedure used (ie, better predictions may be possible).

To summarize, the multilayer neural networks performed better than GLM in terms of MSE. Although specifying a more flexible GLM (eg, adding interaction terms, splines) might improve performance, the neural network does not require adding splines or interactions. Instead, a neural network with hidden layers and nonlinear activation functions can approximate many of these transformations without direct specification. Furthermore, transformed terms for the GLM could also be provided as inputs to the neural network, allowing for variations beyond our specifications. This flexibility may bolster one's confidence in model specification but should still be tempered in light of statistical tradeoffs.[3]

Conclusions

Neural networks are a flexible tool for prediction. The neural networks presented here were basic in structure. In practice, neural networks can consist of hundreds of nodes or dozens of layers. Their structure can also be adapted in numerous ways (eg, recurrent neural networks, variational autoencoders).[1] A list of additional readings is provided in Appendix S3. Despite this diversity, the previous ideas carry across extensions. Neural networks can also be used for causal effect estimation, but there are additional statistical concerns. These issues are reviewed in-depth elsewhere.[3] For the practicing epidemiologist, neural networks offer a flexible alternative for modeling. However, one should recognize that neural networks are not uniformly better than parametric models. To capitalize on their flexibility, many observations are needed for fitting. So, improvements over parametric models are less likely for small sample sizes. Also, neural networks can easily overfit. Finally, constraints on computational resources pose a practical limitation. Regardless of the tradeoffs, one does not need to choose between applying regression or neural networks. Instead, ensemble methods can combine multiple predictive algorithms.[4] Nevertheless, none of these tools obviates the need for background information (eg, variables considered), consideration of systematic errors (eg, missing data, measurement error), or best practices for predictive functions.[5]

Supplementary material

Supplementary material is available at the *American Journal of Epidemiology* online.

Funding

This work was supported in part by the National Institutes of Health (grant K01AI177102 to P.N.Z., and grant R01HD102313 to A.I.N.).

Conflict of interest

The authors declare no conflicts of interest.

Disclaimer

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Data availability

Data and code used for this analysis are available at https://github.com/pzivich/publications-code.

References

1. Abiodun OI, Jantan A, Omolara AE, et al. State-of-the-art in artificial neural network applications: a survey. *Heliyon*. 2018; 4(11):e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
2. Ruder S. An overview of gradient descent optimization algorithms. arXiv. https://doi.org/10.48550/arXiv.1609.04747, June 15, 2017, preprint: not peer reviewed.
3. Zivich PN, Breskin A, Kennedy EH. Machine learning and causal inference. In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons, Ltd; 2022:1-8.
4. Naimi AI, Balzer LB. Stacked generalization: an introduction to super learning. *Eur J Epidemiol*. 2018;33(5):459-464. https://doi.org/10.1007/s10654-018-0390-z
5. Leisman DE, Harhay MO, Lederer DJ, et al. Development and reporting of prediction models: guidance for authors from editors of respiratory, sleep, and critical care journals. *Crit Care Med*. 2020; 48(5):623-633. https://doi.org/10.1097/CCM.0000000000004246