Neural networks

Blanca Cano Camarero

Universidad Autónoma de Madrid

December 13, 2022

Overview

The origins

The term *neural network* has its origins in attempts to find mathematical representations of information processing in biological system.
(McCulloch and Pitts, 1943; Widrow and Hoff, 1960; Rosenblatt, 1962; Rumelhart et al., 1986)

Feed-forward Network Functions

Lineal models:

$$y(x, w) = f\left(\sum_{j=1}^{M} w_j \phi_j(x)\right) \tag{1}$$

where f is a nonlinear activation function in classification and the identity in regression. Our goal is to extend the model by making ϕ depend on parameters and then to allow these parameters to be adjusted along with the coefficients $\{w_i\}$.

Activations

First we construct M linear combinations of the input variables $x_1, \ldots, x_D$ in the form

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{2}$$

where $j \in \{1, \ldots, M\}$, and the superscript indicates the layer of the network the parameters are in.
We shall refer to

- The parameters $w_{ji}^{(1)}$ as *weights*.
- The parameters $w_{j0}^{(1)}$ as the *biases*.
- The quantities a_j as *activations*.

Hidden unit and activation functions

Each of the *activations* parameters are transformed using a **differentiable**, nonlinear *activation function h*

$$z_j = h(a_j). \tag{3}$$

We shall refer to

- The quantities z_j as hidden units.

[1]

[1] Show examples of activations functions.

Discussion: What is the real importance of activation function

- The hypothesis: should't be polynomial.
- ReLu is the most used.
- Kernel method, the kernel not as relevant.
- Balance between precision and computational cost?

Output unit activations

The values (3) are again lineal combined to give *output unit activations*

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{4}$$

where $k \in \{1, \ldots, K\}$, and K is the total number of outputs. This transformation corresponds to the second layer of the network.

Network output

Finally, the output unit activations (4) are transformed using an appropriate activations function to give a set of network output.

- For standard regression problems the activation function is the identity so that $y_k = a_k$.
- For binary classification problems, each output unit activations is transformed using a logistic sigmoid function so that

$$y_k = \sigma(a_k) \tag{5}$$

where

$$\sigma(a) = \frac{1}{1 + exp(-a)}. \tag{6}$$

- For multiclass problems a softmax activation function (4.62)

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)} = \frac{exp(a_k)}{\sum exp(a_j)}. \tag{7}$$

The resulting model

$$y_k(x, w) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{j=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{8}$$

[2] [3]

[2] Comment differences with lineal regression

[3] Write as a matrix and print a graph.

Example of a neural network having a general feed-forwed topology

I have not found any article that create a math formulation.

- Add skip layer connection explicitly.
- Same treat.

Other example recurrent neural
Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network Sherstinsky (2018) (difference equations and PST).

Weight space symmetries

Change the weights and obtain the same results.
How to obtain symmetries?

- Changing the sigh of all nodes and using that *tanh* is odd.
- Permuting hidden nodes.

Bisop: Role comparing with the Bayesian model. For us: There are going to be different solutions.
Some observation:

- Homotopy of the image for me the same function.
- There are any approach that use equivalence class?
- Meng et al. (2018)

The error function

$$t = y(x, w) + \epsilon \tag{9}$$

where $\epsilon \sim \mathcal{N}(0, \beta^{-1})$ and $\beta \in \mathbb{R}$ is the precision $(\beta^{-1} = \sigma^2)$.
Thus we can write

$$p(t|x, w, \beta) = \mathcal{N}(t|y(x, w), \beta^{-1}). \tag{10}$$

For a single real-valued variable x, the Gaussian distribution is defined by

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}. \tag{11}$$

We have to minimize the following error function

$$-\log p(t|X,w,\beta) = \frac{n\beta}{2}\sum_{i=1}^{n}\{y(x_i,w)-t_i\}^2 - \frac{n}{2}\log\beta - \frac{n}{2}\log(2\pi) \tag{12}$$

which can be used to learn the parameters w and β.

The error function

Let take the error function as the

$$E(w) = \frac{1}{2}\sum_{i=1}^{n}\{y(x_i, w) - t_i\}^2 \tag{13}$$

where we have discarded additive and multiplicative constant.
Some considerations:

- The value of w found by minimizing $E(w)$ will be denoted as w_{ML} (maximum likelihood solution).
- The nonlinearity of the network function $y(x_n, w)$ causes the error $E(w)$ to be nonconvex.
- So in practice W_{ML} would be a local minimum.

About the precision

Having found w_{ML} the value of β can be found by minimizing the negative log likelihood to give

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\frac{1}{2}\sum_{i=1}^{N}\{y(x_i, w_{ML}) - t_i\}^2 \tag{14}$$

Multiple target variable

$$p(t|x, w) = \mathcal{N}(t|y(x, w), \beta^{-1} I). \tag{15}$$

The noise precision is the given by

$$\frac{1}{\beta_{ML}} = \frac{1}{NK} \sum_{i=1}^{N} \|y(x_i, w_{ML}) - t_i\|^2 \tag{16}$$

Minimizing regression case

$$\frac{\partial E}{\partial a_k} = y_k - t_k \tag{17}$$

Binary classification problem

We have a single target t such that $t = 1$ dentotes class C_1 and $t = 0$ denotes class C_2. As we see last week we consider a single output whose activation function is a logistic sigmoid:

$$y = \frac{1}{1 + exp(-a)} \tag{18}$$

so that

$$0 \leq y(x, w) \leq 1. \tag{19}$$

We can interpret $y(x, w)$ as the condicional probability $p(C_1|x)$.

The cross entropy error function

The conditional distribution of targets given inputs is the Bernoulli distribution of the form

$$p(t|x, w) = y(x, w)^t \{1 - y(x, w)\}^{1-t}. \tag{20}$$

If we consider a training set of independent observation, then the error function which is given by the negative log likelihood, is then a cross entropy error function of the form

$$E(w) = -\sum_{n=1}^{N} \{t_n \log y_n + (1 - t_n) \log(1 - y_n)\} \tag{21}$$

Using cross entropy error function instead of the sum of squares for classification problem leads to faster training as well as improved generalization.

K binary classification

$$p(t|x, w) = \prod_{k=1}^{K} y(x, w)^{t} \{1 - y(x, w)\}^{1-t} . \tag{22}$$

$$E(w) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{t_{nk} \log y_{nk} + (1 - t_{nk}) \log(1 - y_{nk})\} \tag{23}$$

1 of K coding scheme

The network outputs are interpreted as

$$y_k(x, w) = p(t_k = 1|x), \tag{24}$$

leading to the following error function

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{kn} \log y_k(x_n, w). \tag{25}$$

Parameter optimization

Backpropagation.

Next week

- More method??
- Regularization.
- Convolutional networks.
- Mixture Density Networks

References I

Qi Meng, Wei Chen, Shuxin Zheng, Qiwei Ye, and Tie-Yan Liu. Optimizing neural networks in the equivalent class space. 02 2018.

Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018. URL `http://arxiv.org/abs/1808.03314`.