Neural Networks

Andresito Ponce

April 11, 2020

The general idea behind neural networks is to maintain a **basis function** whose parameters can change during testing. Then the function can know how to perform "better" as the testing goes on.

Previously, we had only considered a linear combination of a fixed basis function, which usually took the form

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

However, our goal is to make some sort of these functions depend on certain parameters.

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

where a_j is the **activation** value.[1]

We then transform the activation value by another nonlinear function h, and are left with the final result $z_j = h(a_j)$.

To summarize the basic structure: supposing the result fits into K classes, we then sum the weights of the D nodes in the first layer and the M nodes of the second layer, and we get a formula in the end resembling[2]

$$y_k(x, w) = \phi(\sum_{j=1}^{M} w_{kj}^{(2)} h(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}) + w_{k0}^{(2)})$$

[1] Remember the activation is the final output of this node. In classification, it determines which class we eventually assign to an input. The (1) superscript indicates they are the first layer of the NN.

[2] This equation basically says that the final value of class k is the result of adding all the nodes from all the layers that feed into k for the two levels?