A Wizard's Journey to Master Logic Gates

by Antonio Bernardini

Introduction

I left the section on the math used to build the neural network at the end because most people just hear the word "math" get the chills.

Precisely for this reason I will try to explain myself as simply as possible.

> In this section we refer to the code in `src/nn.py`.

Training Neural Networks Like a Magic Spell

The `NeuralNetwork` class is like a little math whiz learning to solve logic gate problems.

Wearing his magical hat, the wizard uses calculating magic to train.

First, he starts with random weights, like numbers picked at random from a magic cylinder.

Training Neural Networks Like a Magic Spell

Then, when shown an example of a logic gate, the magician does some calculations with his secret formulas.

Use the *sigmoidal magic function* to transform the numbers and calculate the final output.

Training Neural Networks Like a Magic Spell

If the output is incorrect, the magician adjusts his weights based on the *error* made, using the *descending gradient formula*.

He keeps repeating these math spells for a while, until he can guess the output correctly for all the training examples.

Training Neural Networks Like a Magic Spell

And voila!

The wizard has mastered the logic gates!

Now he can help you solve logic problems with his math magic!

Magic Revealed Through Formulas

The `NeuralNetwork` class uses the *sigmoid activation function*, defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

to transform the values of neurons in the network.

Magic Revealed Through Formulas

During training, the *descending gradient technique* is used to update the network weights.

The *error* between the expected output and the actual output is calculated and *back-propagated* through the network.

Backpropagation

The weights are then updated using the formula:

$$w_+ = w + \eta \cdot \text{input} \cdot \text{error} \cdot \sigma'(\text{output})$$

where η is the learning rate and $\sigma'(x)$ is the derivative of the sigmoid function, defined as:

$$\sigma'(x) = x \cdot (1 - x)$$

Conclusion

Thus, the math whiz trains his neural network using these magic formulas, trying to minimize the *error* and correctly guess the output of the logic gates.

> I won't dwell too much on mathematics because I'm preparing a pdf that starts from the basics in order to fully understand the algorithms behind neural networks.