Gradient Descent Explained, really

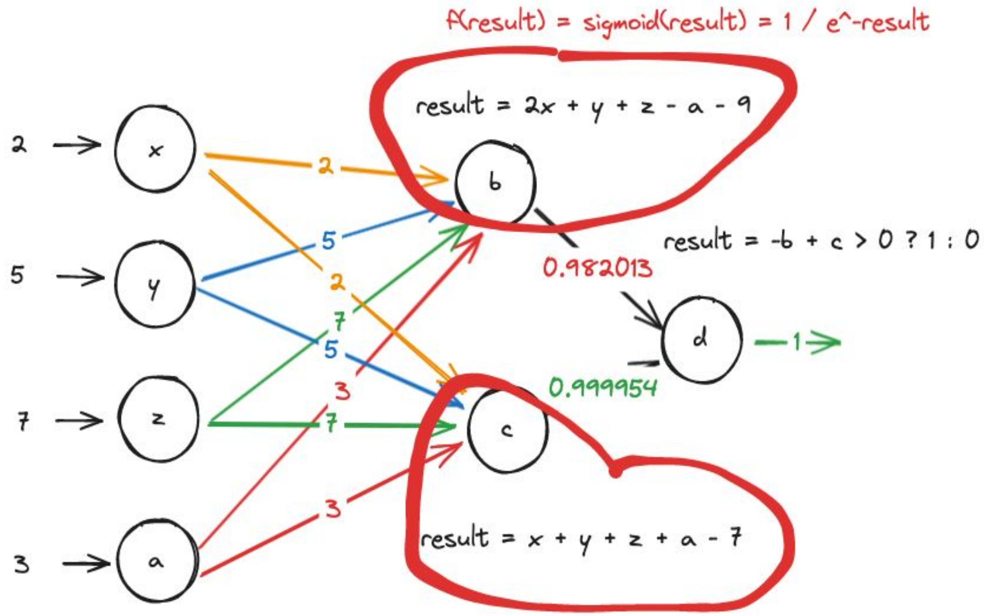
Roger Lam
lamroger.com

Neural Network

Remember our neural network?

Our collection of nodes that can be represented by equations.

And each of those equations have coefficients that make the calculations special.

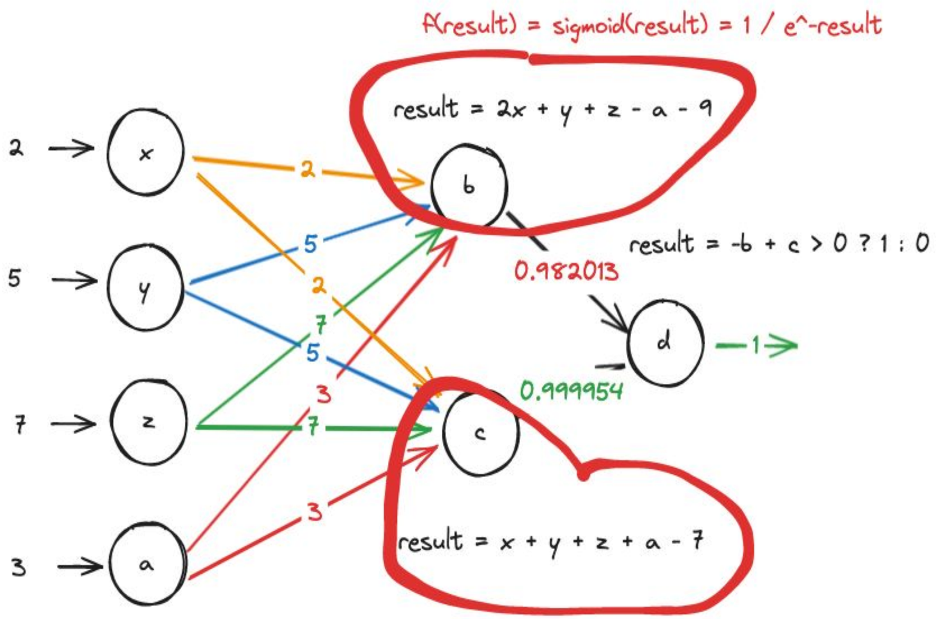


Training determines the coefficients

So how did we come up with, for example, 2x, y, z,-a and -9?

We “trained” the neural network.

What does training mean?



Start randomly

Like training in a sport - everyone starts somewhere and tries to get better.

So we start randomly.

I'm representing the coefficients as the boldness of the arrows.

For example, 5a would be bolder than 2b.

The higher the "weight", the darker the arrow.

So maybe x, y, 2z, 3a and -1?

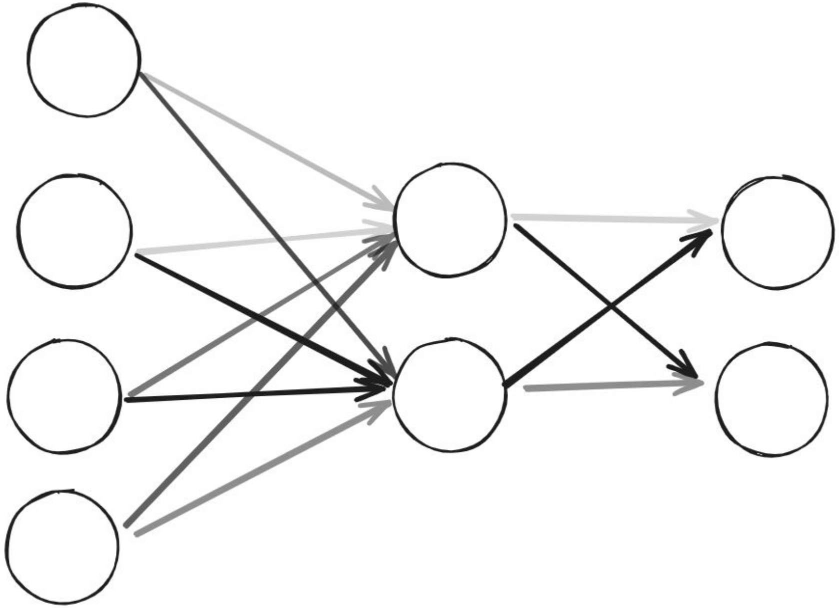
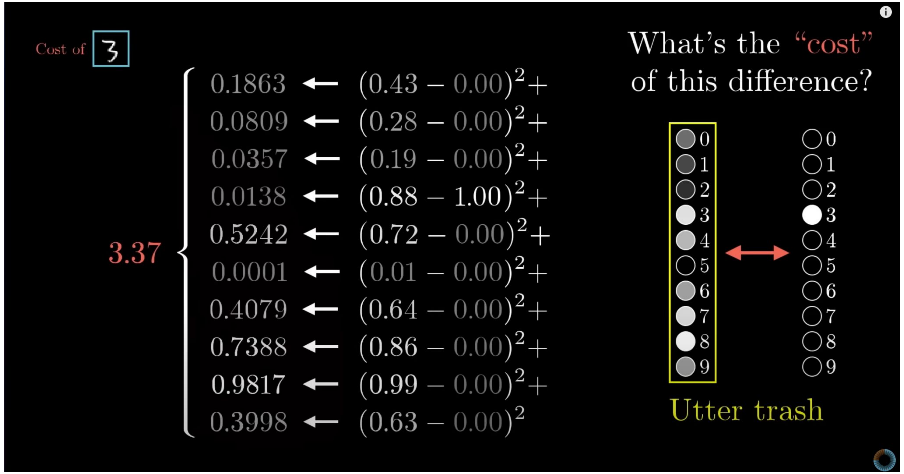


Figure out how wrong we are

To get better, we need to know what to change.

How wrong are we?

We define a “cost function” as an equation that tells us how good of an answer we have based on the parameters we chose and examples we have.



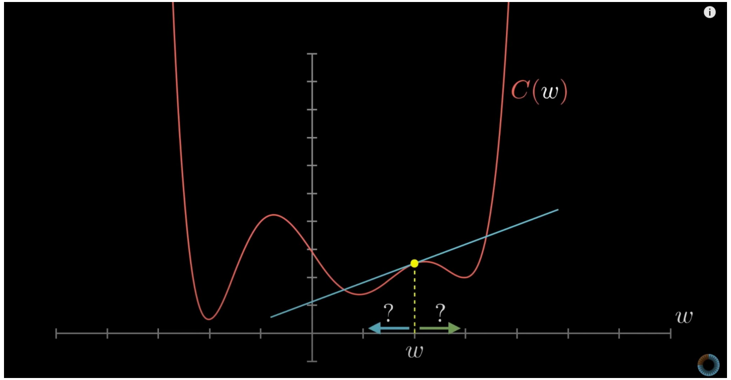
https://www.youtube.com/watch?v=IHZwWFHWa-w

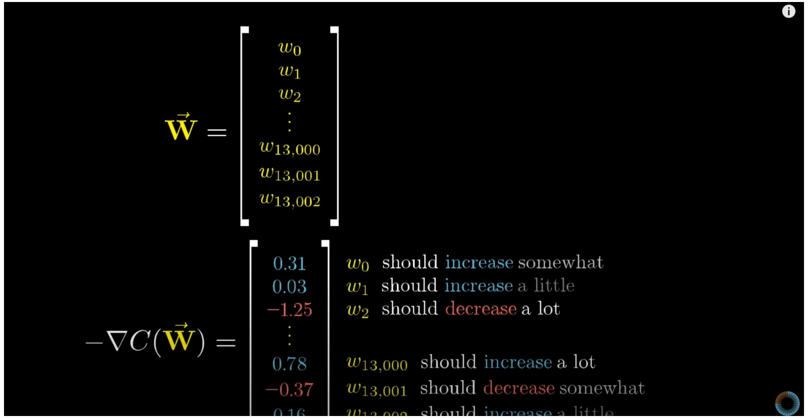
Figure how to change to improve

Since we want to minimize our cost function (how wrong we are), we can take the derivative of the function (the gradient) and go in the negative direction.

We do this with respect to all of the weight. Nudge a little.

And calculate and nudge again and again.



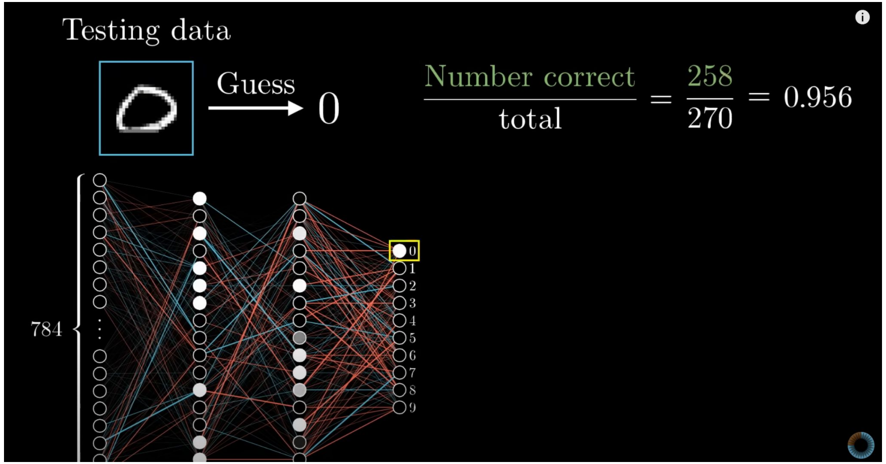


And we're good!

Well, pretty good.

We can keep training but we don't want to overfit on our examples.

This is one example of a neural network architecture.



Check out 3Blue1Brown's videos:
https://www.youtube.com/watch?v=IHZwWFHWa-w

Kaggle competition:
https://www.kaggle.com/c/digit-recognizer/overview