

Neural Network Representation

(SPEECH)

You

(DESCRIPTION)

Text, deep learning dot AI. One hidden layer neural network. Neural network representation

(SPEECH)

see me draw a few pictures of neural networks.

In this video, we'll talk about exactly what those pictures means.

In other words, exactly what those neural networks that we've been drawing represent.

And we'll start with focusing on the case of neural networks with what was called a single hidden layer.

Here's

(DESCRIPTION)

Diagram of a network. At left, a column of three input features, x_1 , x_2 , and x_3 . Next column to the right, four vertically stacked circles representing nodes. Arrows show that each node takes all three of the input features as its own input. To the right of the column of four nodes, a single node, which takes as its input the four outputs of the four stacked nodes. The output of that single node is the label \hat{y}

(SPEECH)

a picture of a neural network.

Let's give different parts of these pictures some names.

We have the input features, x_1 , x_2 , x_3 stacked up vertically.

And this is called the input layer of the neural network.

So maybe not surprisingly, this contains the inputs to the neural network.

Then there's another layer of circles.

And

(DESCRIPTION)

Indicating column of four nodes

(SPEECH)

this is called a hidden layer of the neural network.

I'll come back in a second to say what the word hidden means.

But the final layer here is formed by, in this case, just one node.

And this single-node layer is called the output layer, and is responsible for generating the predicted value \hat{y} .

In a neural network that you train with supervised learning, the training set contains values of the inputs x as well as the target outputs y .

So the term hidden layer refers to the fact that in the training set, the true values for these nodes in the middle are not observed.

That is, you don't see what they should be in the training set.

You see what the inputs are.

You see what the output should be.

But the things in the hidden layer are not seen in the training set.

So that kind of explains the name hidden layer; just because you don't see it in the training set.

Let's introduce a bit more notation.

Whereas previously, we were using the vector X to denote the input features and alternative notation for the values of the input features will be A^0 .

And the term A also stands for activations, and it refers to the values that different layers of the neural network are passing on to the subsequent layers.

So the input layer passes on the value x to the hidden layer, so we're going to call that activations of the input layer A^0 .

The next layer, the hidden layer, will in turn generate some set of activations, which I'm going to write as A^1 .

So

(DESCRIPTION)

Filling in the activation values for the four individual nodes of the hidden layer

(SPEECH)

in particular, this first unit or this first node, we generate a value A^1_1 .

This second node we generate a value.

Now we have a subscript 2 and so on.

And so, A^1 , this is a four dimensional vector you want in Python because the 4×1 matrix, or a 4 column vector, which looks like this.

And

(DESCRIPTION)

Rectangular matrix notation

(SPEECH)

it's four dimensional, because in this case we have four nodes, or four units, or four hidden units in this hidden layer.

And then finally, the open layer regenerates some value A^2 , which is just a real number.

And so \hat{y} is going to take on the value of A^2 .

So this is analogous to how in logistic regression we have y hat equals a and in logistic regression which we only had that one output layer, so we don't use the superscript square brackets.

But with our neural network, we now going to use the superscript square bracket to explicitly indicate which layer it came from.

One funny thing about notational conventions in neural networks is that this network that you've seen here is called a two layer neural network.

And the reason is that when we count layers in neural networks, we don't count the input layer.

So the hidden layer is layer one and the output layer is layer two.

In our notational convention, we're calling the input layer layer zero, so technically maybe there are three layers in this neural network.

Because there's the input layer, the hidden layer, and the output layer.

But in conventional usage, if you read research papers and elsewhere in the course, you see people refer to this particular neural network as a two layer neural network, because we don't count the input layer as an official layer.

Finally, something that we'll get to later is that the hidden layer and the output layers will have parameters associated with them.

So the hidden layer will have associated with it parameters w and b .

And I'm going to write superscripts square bracket 1 to indicate that these are parameters associated with layer one with the hidden layer.

We'll see later that w will be a 4 by 3 matrix and b will be a 4 by 1 vector in this example.

Where the first coordinate four comes from the fact that we have four nodes of our hidden units and a layer, and three comes from the fact that we have three input features.

We'll talk later about the dimensions of these matrices.

And it might make more sense at that time.

But in some of the output layers has associated with it also, parameters w superscript square bracket 2 and b superscript square bracket 2.

And it turns out the dimensions of these are 1 by 4 and 1 by 1.

And

(DESCRIPTION)

W is 1 by 4, B is 1 by 1

(SPEECH)

these 1 by 4 is because the hidden layer has four hidden units, the output layer has just one unit.

But we will go over the dimension of these matrices and vectors in a later video.

So you've just seen what a two layered neural network looks like.

That is a neural network with one hidden layer.

In the next video, let's go deeper into exactly what this neural network is computing.

That is how this neural network inputs x and goes all the way to computing its output \hat{y} .