

What is a neural network?

(DESCRIPTION)

Text, Introduction to Deep Learning. What is a Neural Network? Website, deep learning, dot, A.I.

(SPEECH)

The term, Deep Learning, refers to training Neural Networks, sometimes very large Neural Networks.

So what exactly is a Neural Network?

In this video, let's try to give you some of the basic intuitions.

(DESCRIPTION)

New slide, Housing Price Prediction. A right angle graph is displayed. The X plane is labeled, size of house. The Y plane is labeled, price.

(SPEECH)

Let's start to the Housing Price Prediction example.

Let's say you have a data sets with six houses, so you know the size of the houses in square feet or square meters and you know the price of the house and you want to fit a function to predict the price of the houses, the function of the size.

So if you are familiar with linear regression you might say, well let's put a straight line to these data so and we get a straight line like that.

But to be Pathans you might say well we know that prices can never be negative, right.

So instead of the straight line fit which eventually will become negative, let's bend the curve here.

So it just ends up zero here.

So this thick blue line ends up being your function for predicting the price of the house as a function of this size.

Whereas zero here and then there's a straight line fit to the right.

So you can think of this function that you've just fit the housing prices as a very simple neural network.

It's almost as simple as possible neural network.

Let me draw it here.

We have as the input to the neural network the size of a house which one we call x .

It goes into this node, this little circle and then it outputs the price which we call y .

So this little circle, which is a single neuron in a neural network, implements this function that we drew on the left.

And all the neuron does is it inputs the size, computes this linear function, takes a max of zero, and then outputs the estimated price.

And by the way in the neural network literature, you see this function a lot.

This function which goes to zero sometimes and then it'll take off as a straight line.

This function is called a ReLU function which stands for rectified linear units.

So R-E-L-U. And rectify just means taking a max of 0 which is why you get a function shape like this.

You don't need to worry about ReLU units for now but it's just something you see again later in this course.

So if this is a single neuron, neural network, really a tiny little neural network, a larger neural network is then formed by taking many of the single neurons and stacking them together.

So, if you think of this neuron that's being like a single Lego brick, you then get a bigger neural network by stacking together many of these Lego bricks.

Let's see an example.

Let's say that instead of predicting the price of a house just from the size, you now have other features.

You know other things about the house, such as the number of bedrooms, I should have wrote [INAUDIBLE] bedrooms, and you might think that one of the things that really affects the price of a house is family size, right?

So can this house fit your family of three, or family of four, or family of five?

And it's really based on the size in square feet or square meters, and the number of bedrooms that determines whether or not a house can fit your family's family size.

And then maybe you know the zip codes, in different countries it's called a postal code of a house.

And the zip code maybe as a feature to tell you, walkability?

So is this neighborhood highly walkable?

Thing just walks the grocery store?

Walk the school?

Do you need to drive?

And some people prefer highly walkable neighborhoods.

And then the zip code as well as the wealth maybe tells you, right.

Certainly in the United States but some other countries as well.

Tells you how good is the school quality.

So each of these little circles I'm drawing, can be one of those ReLU, rectified linear units or some other slightly non linear function.

So that based on the size and number of bedrooms, you can estimate the family size, their zip code, based on walkability, based on zip code and wealth can estimate the school quality.

And then finally you might think that well the way people decide how much they're will to pay for a house, is they look at the things that really matter to them.

In this case family size, walkability, and school quality and that helps you predict the price.

So in the example x is all of these four inputs.

And y is the price you're trying to predict.

And so by stacking together a few of the single neurons or the simple predictors we have from the previous slide, we now have a slightly larger neural network.

How you manage neural network is that when you implement it, you need to give it just the input x and the output y for a number of examples in your training set and all this things in the middle, they will figure out by itself.

So what you actually implement is this.

Where, here, you have a neural network with four inputs.

So the input features might be the size, number of bedrooms, the zip code or postal code, and the wealth of the neighborhood.

And so given these input features, the job of the neural network will be to predict the price y .

And notice also that each of these circle, these are called hidden units in the neural network, that each of them takes its inputs all four input features.

So for example, rather than saying these first nodes represent family size and family size depends only on the features X_1 and X_2 .

Instead, we're going to say, well neural network, you decide whatever you want this known to be.

And we'll give you all four of the features to complete whatever you want.

So we say that layers that this is input layer and this layer in the middle of the neural network are density connected.

Because every input feature is connected to every one of these circles in the middle.

And the remarkable thing about neural networks is that, given enough data about x and y , given enough training examples with both x and y , neural networks are remarkably good at figuring out functions that accurately map from x to y .

So, that's a basic neural network.

It turns out that as you build out your own neural networks, you probably find them to be most useful, most powerful in supervised learning incentives, meaning that you're trying to take an input x and map it to some output y , like we just saw in the housing price prediction example.

In the next video let's go over some more examples of supervised learning and some examples of where you might find your networks to be incredibly helpful for your applications as well.