Planning Course Advanced Learning

Week 1 and 2 - Neural Network

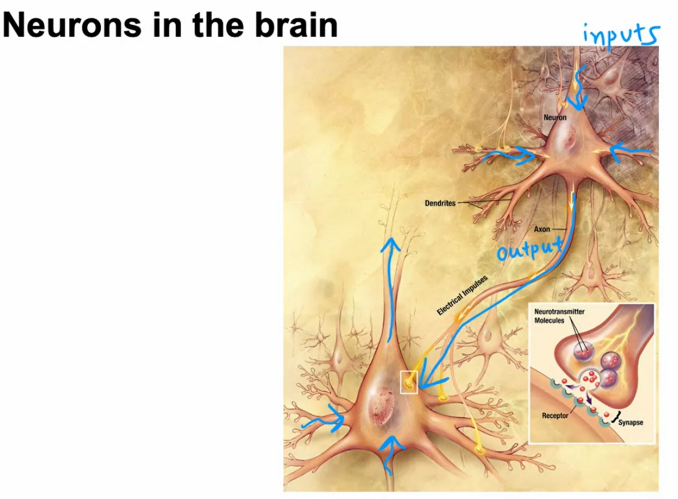
Week 3 - Practical advice for building machine learning system

Week 4 - Decision Tree

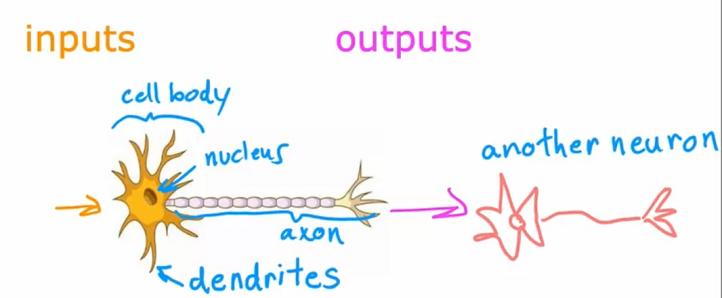
**NEURAL NETWORK**

Neural Network algoritma that try to mimic the brain.

The first we can explain how the brain works.



Neuron connect with each other. Because connect with each other neuron can have input and output.



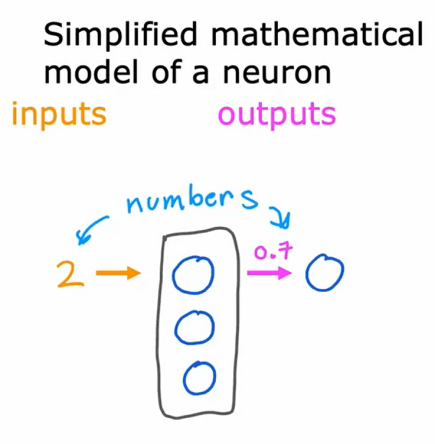
If we imagine, the biological neuron have 3 main component, dendrites, Nucleus, axon.

Dendrit -- > input

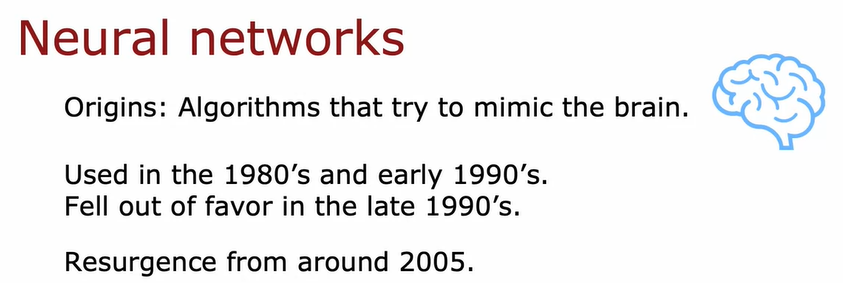
Nucleus --> Process

Axon --> Output

The schema above we duplicated to simplified mathematical model of a neuron



**History of Neural Network**



Work in neural networks had started back in the 1950s, and then it fell out of favor for a while. Then in the 1980s and early 1990s, they gained in popularity again and showed tremendous traction in some applications like handwritten digit recognition,

which were used even backed then to read postal codes for writing mail and for reading dollar figures in handwritten checks. But then it fell out of favor again in the late 1990s. It was from about 2005 that it enjoyed a resurgence and also became

re-branded little bit with deep learning.



Since then, neural networks have revolutionized application area after application area. I think the first application area that modern neural networks or deep learning,

had a huge impact on was probably speech recognition, where we started to see much better speech recognition systems due to modern deep learning and authors such as [inaudible] and Geoff Hinton were instrumental to this, and then it started to make inroads into computer vision. Sometimes people still speak of

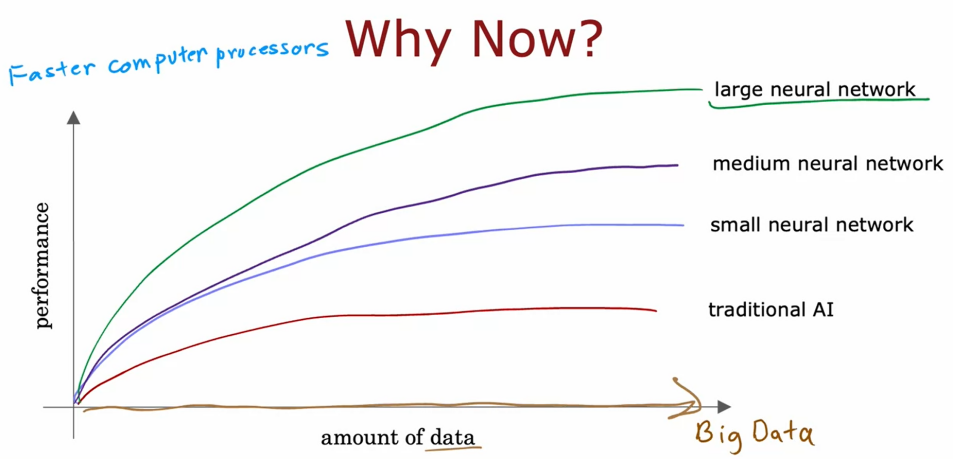
the ImageNet moments in 2012, and that was maybe a bigger splash where then [inaudible] draw their imagination and had a big impact on computer vision.

Then the next few years, it made us inroads into texts or into natural language processing, and so on and so forth. Now, neural networks are used in everything from climate change to medical imaging to online advertising to prouduct recommendations and really lots of application areas of machine learning now use neural networks

Why Neural Network Increase just now?

- the for this era, we have a lot or enough data to train with neural network

- in this era, we have device to compute a lot of data and process



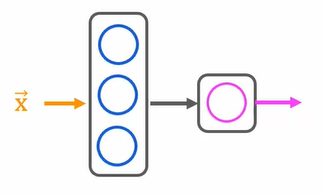
Example Case Neural Network

- Case Demand Prediction

- image recognize

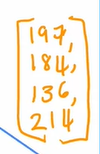
Video ini belum

NEURAL NETWORK MODEL

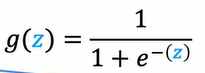


We have input data X, 3 hidden layer and, 1 ouput layer, so we can describe every layer in this model.

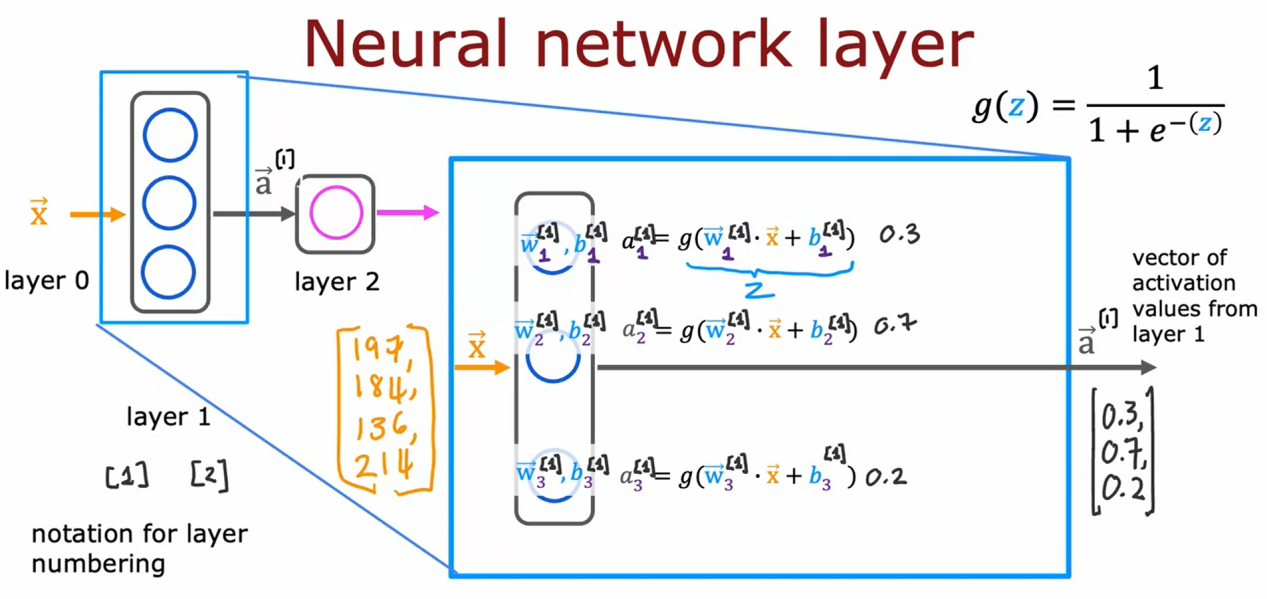
The First is Hidden Layer, we have input x =



And we can define sigmoid function for calculation neural network, same like previous course in logistic regression



So we can describe calculation in hidden layer



1. = notation for layer numbering

We have 4 data input, and then calculate with 3 node hidden layer and output can be 3 value.

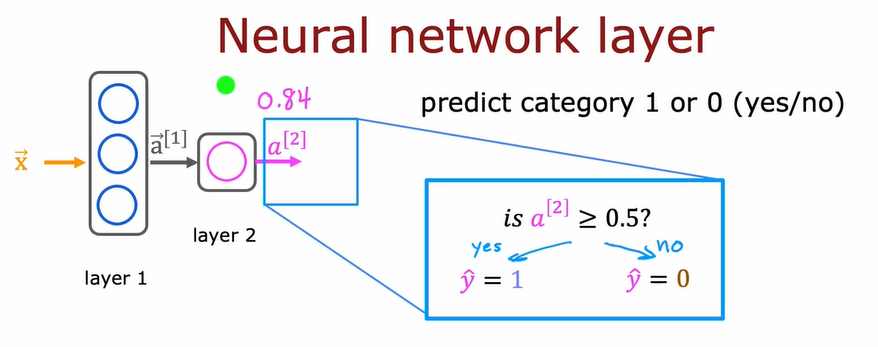
And than we can continue to the next step, it is calculate with output layer



And the output from hidden layer can be input layer for the next layer in this case is output layer.

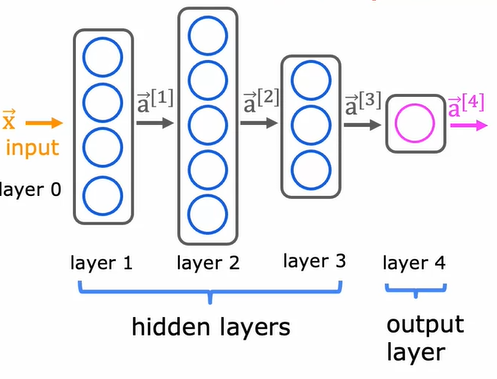
So input layer can be 3 value , and process with 1 hidden layer and the output can be 1 value it is 0.84 .

Finnaly we have probabilty of prediction, the final step is we can define treshold can make the prediction

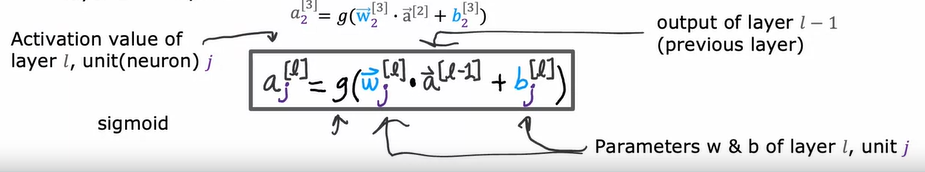


We have treshold if probability more than or equal 0.5 the result y = 1, and if less than 0,5 the result y = 0 .

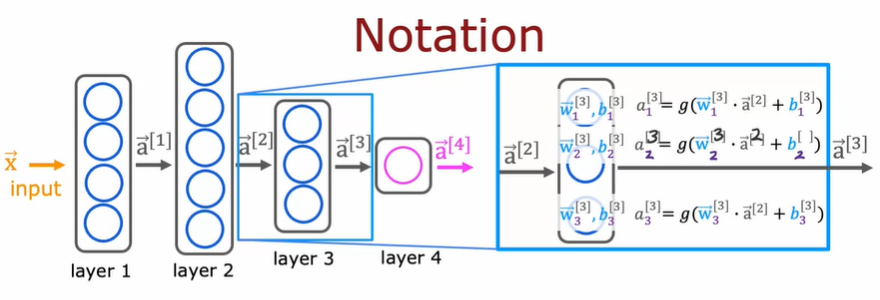
MORE COMPLEX NEURAL NETWORK



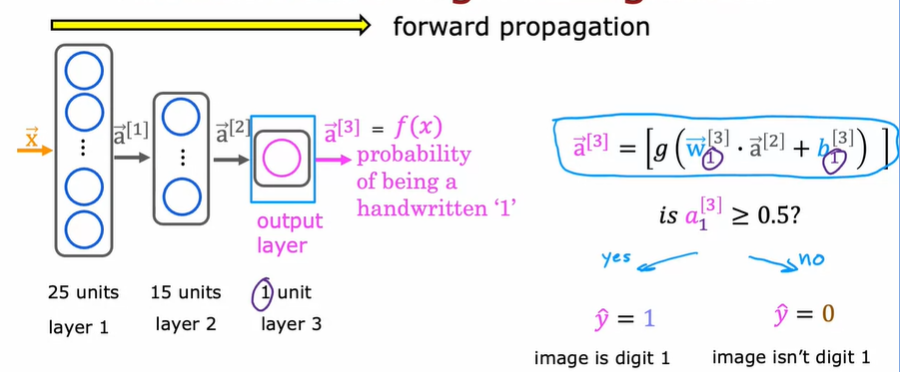
This is more complex neural network, to calculate this one with manualy can be tricky, the specially for define every parameter, because that , the first we can define the activation function (a) like that.



So we can give some example in layer 3, the output of later 2 is a[2]. and the output can be input of hidden layer and calculate this one

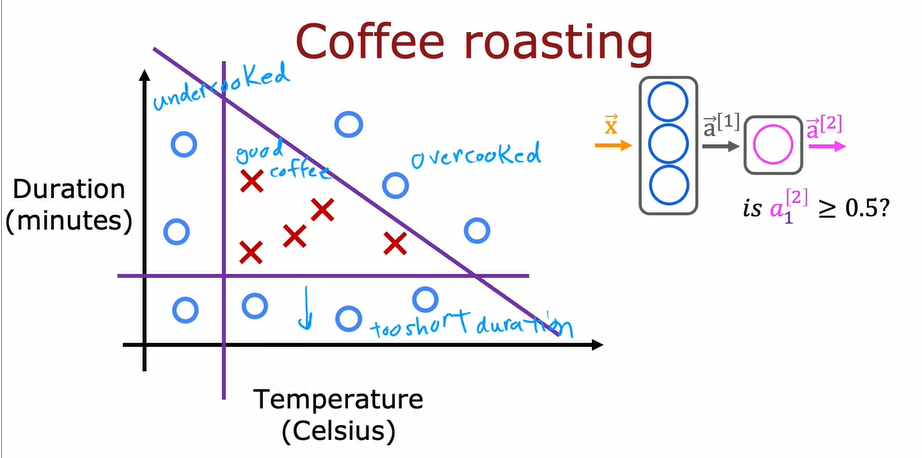


In neural network process calculating from left to right so the calculating call as **Forward Propagation.**



TENSORFLOW IMPLEMENTATION

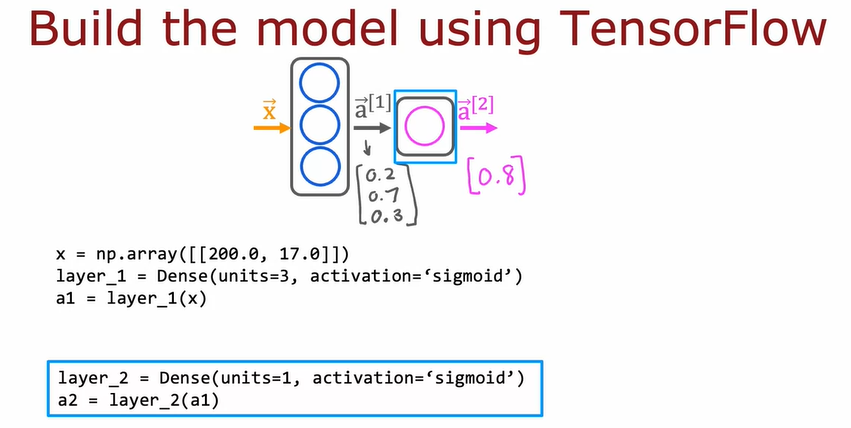
Case Coffe Roasting



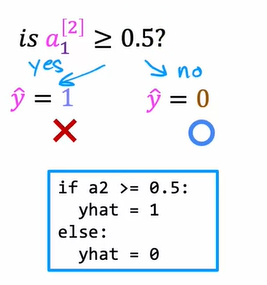
We make model to create good roasting coffe.

In this case we have 2 input (duration and Temperature)

We can create tensoflow model

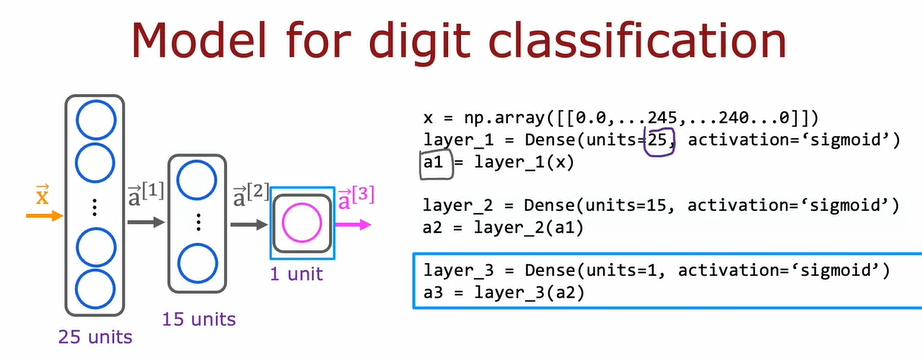


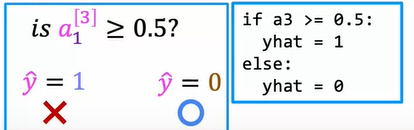
The result can be probabilty , and than we define treshold



Note : dense is another name for layer in tensorflow

MODEL FOR DIGIT CLASSIFICATION

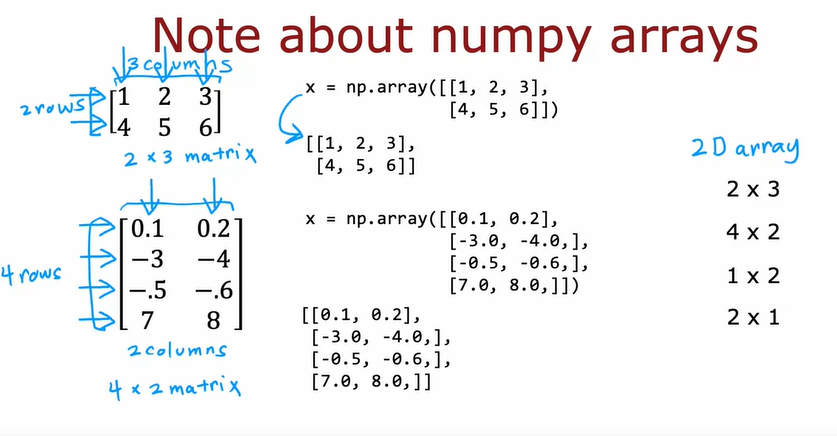


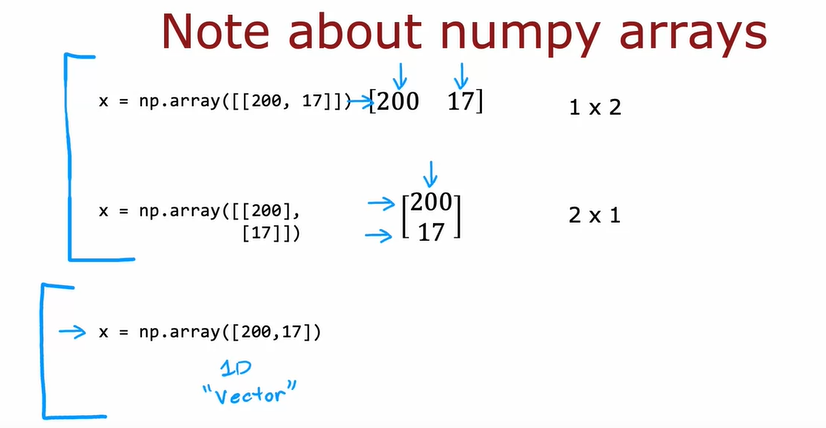


DATA IN TENSORFLOW

I want to step through with you how data is represented in NumPy and in TensorFlow. So that as you're implementing new neural networks, you can have a consistent framework to think about how to represent your data. One of the unfortunate things about the way things are done in code today is that many, many years ago NumPy was first created and became a standard library for linear algebra and Python. And then much later the Google brain team, the team that I had started and once led created TensorFlow.And so unfortunately there are some inconsistencies between how data is represented in NumPy and in TensorFlow. So it's good to be aware of these conventions so that you can implement correct code and hopefully get things running in your neural networks. Let's start by taking a look at how TensorFlow represents data

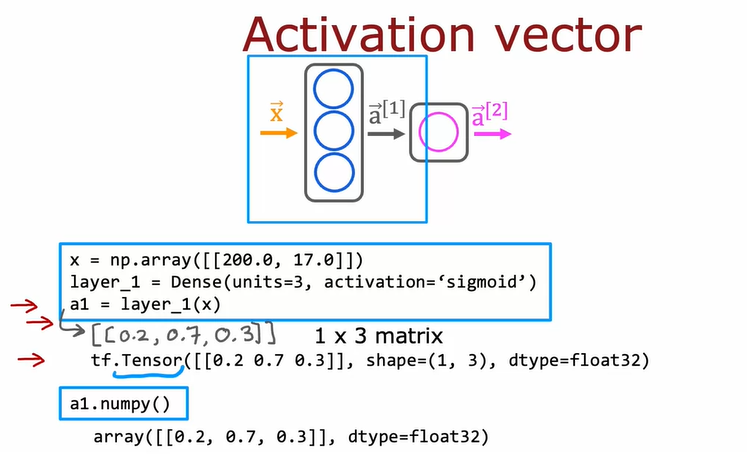
Before we explain about data in tensorflow wa can explain again how numpy work.





In above this is how numpy representation data .

So we can work with Tensor



If we are print a1, the result it is tf.Tensor(….)

A tensor here is a data type that the TensorFlow team had created in order to

store and carry out computations on matrices efficiently.

So whenever you see tensor just think of that matrix on these few slides.

Technically a tensor is a little bit more general than the matrix but for

the purposes of this course,

think of tensor as just a way of representing matrices.

So remember I said at the start of this video that there's the TensorFlow way of

representing the matrix and the NumPy way of representing matrix.

If you wanna convert a2 to numpy array you can do with code a2.numpy()

I'm used to loading data and manipulating data in NumPy, but when you pass a NumPy

array into TensorFlow, TensorFlow likes to convert it to its own internal format.

The tensor and then operate efficiently using tensors.

And when you read the data back out you can keep it as a tensor or

convert it back to a NumPy array.

I think it's a bit unfortunate that the history of how these library evolved has

let us have to do this extra conversion work when

actually the two libraries can work quite well together.

But when you convert back and forth, whether you're using a NumPy array or

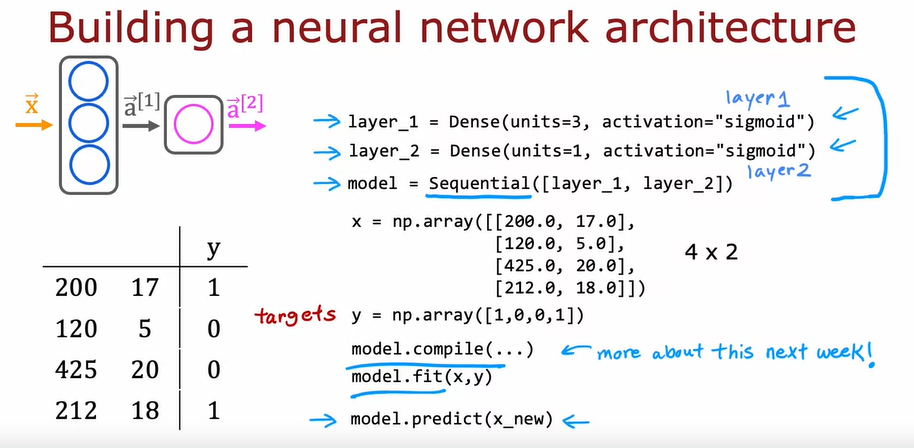
a tensor, it's just something to be aware of when you're writing code.

Next let's take what we've learned and

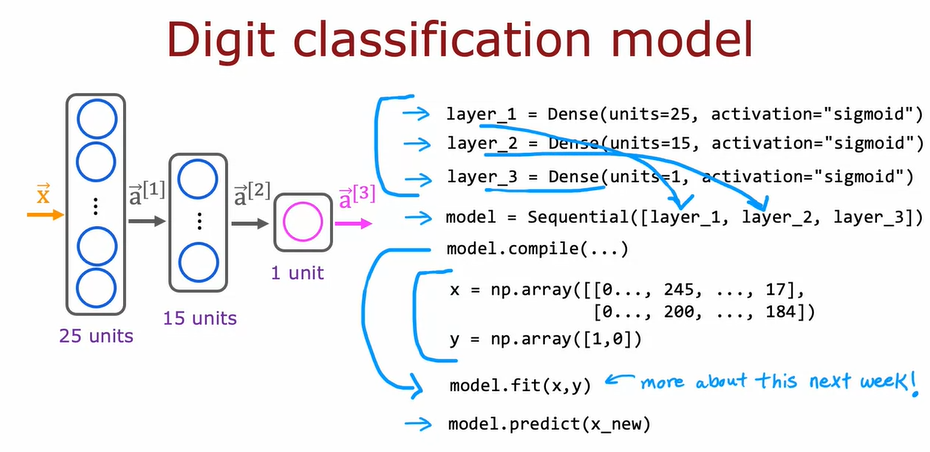
put it together to actually build a neural network.

Let's go see that in the next video.

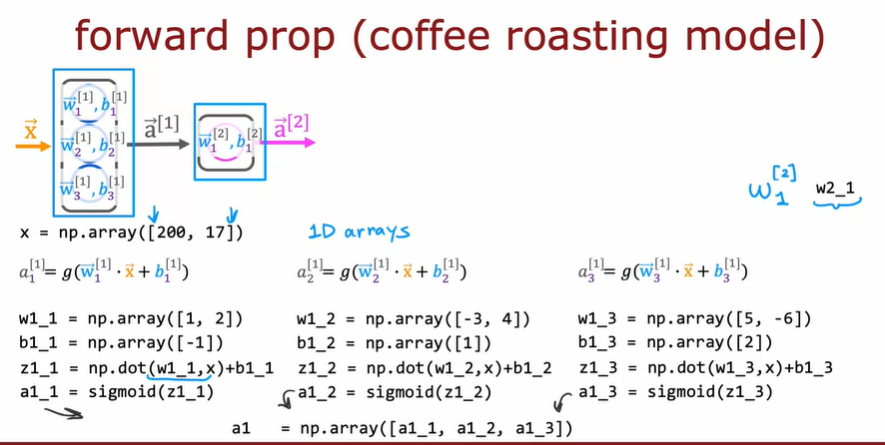
BUILDING A NEURAL NETWORK



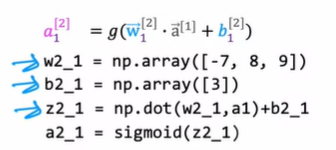
And for the Digit Classification



NEURAL NETWORK IMPLEMENTATION IN PYTHON

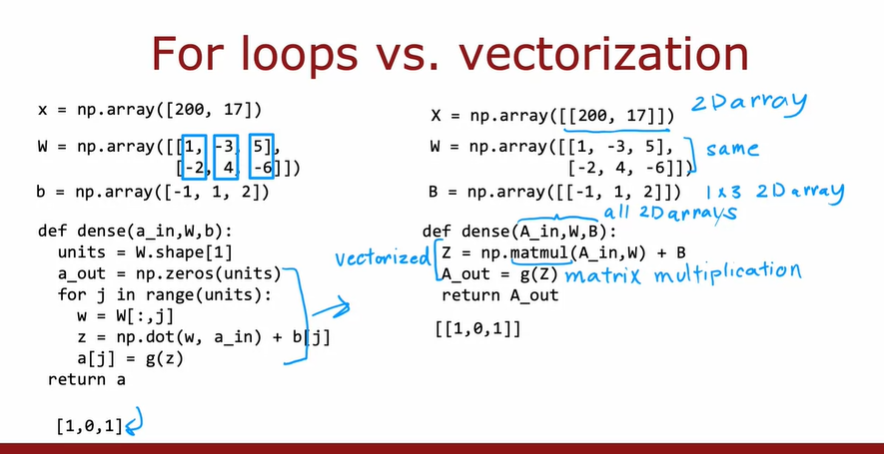


And then we can calculate a2



GENERAL IMPLEMENTATOIN OF FORWARD PROPAGATION

HOW NEURAL NETWORK ARE IMPLEMENTATION effeciently



MATRIX MULTIPLICATOIN

