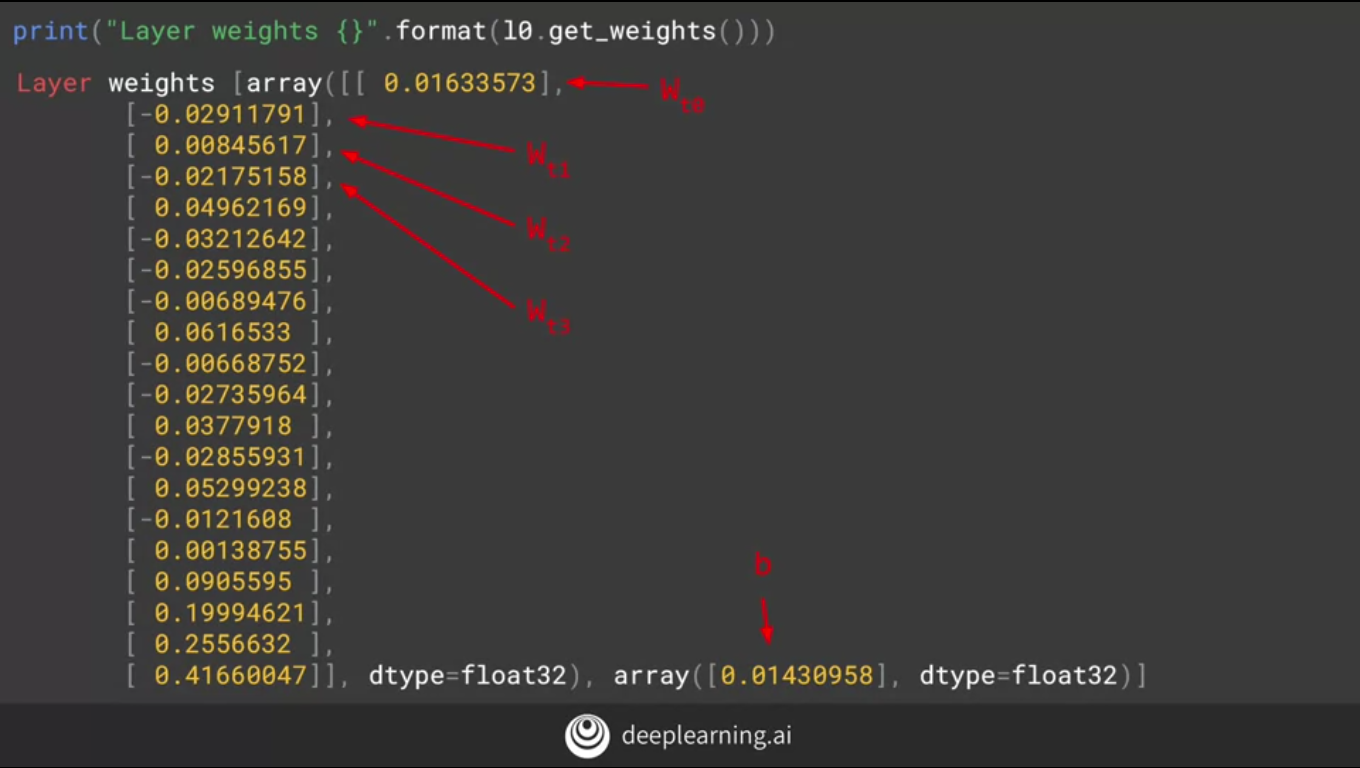
Sequence bias is when the order of things can impact the selection of things. For example, if I were to ask you your favorite TV show, and listed "Game of Thrones", "Killing Eve", "Travellers" and "Doctor Who" in that order, you're probably more likely to select 'Game of Thrones' as you are familiar with it, and it's the first thing you see. Even if it is equal to the other TV shows. So, when training data in a dataset, we don't want the sequence to impact the training in a similar way, so it's good to shuffle them up.

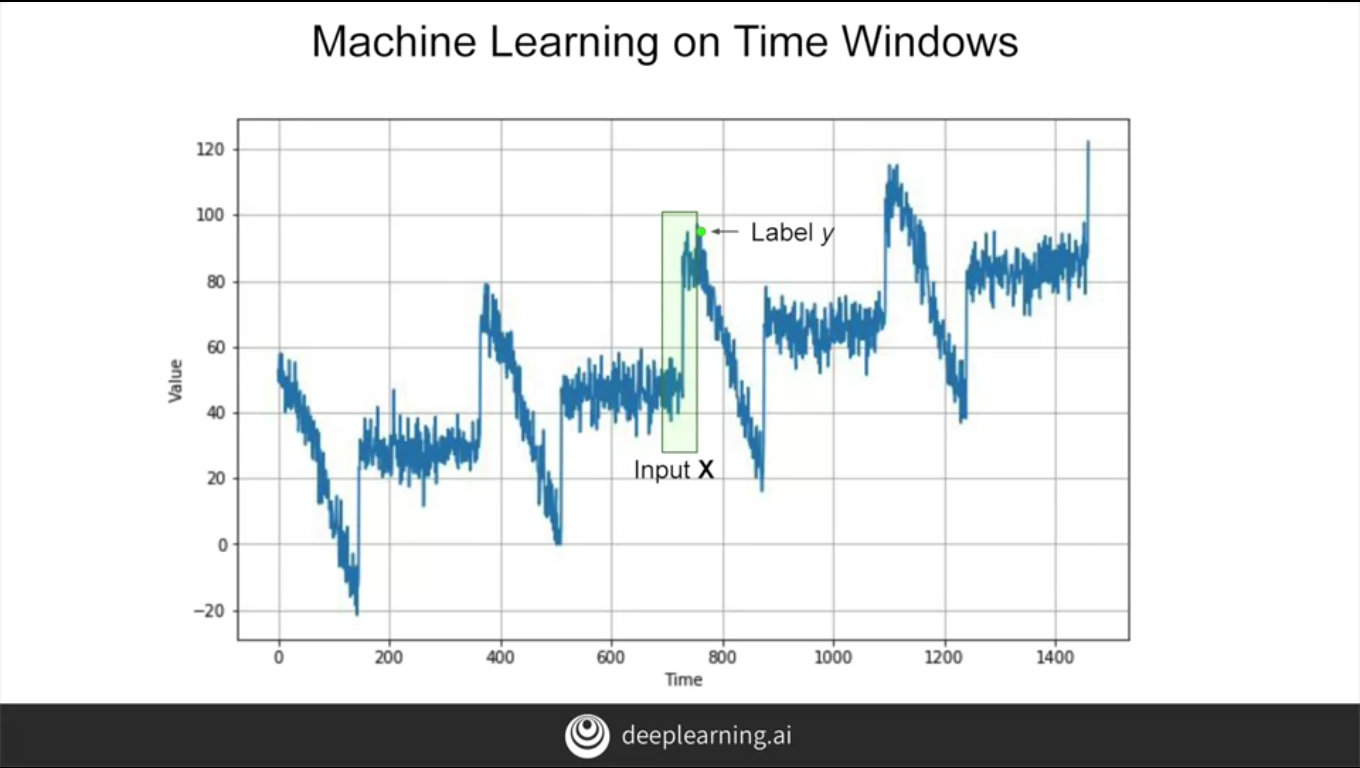


The output will look like this.If you inspect it closely, you will see that the first array has 20 values in it, and the secondary has only one value.

This is because the network has learned a linear regression to

fit the values as best as they can. So each of the values in the first array can

be seen as the weights for the 20 values in x, and the value for the second array is the b value.



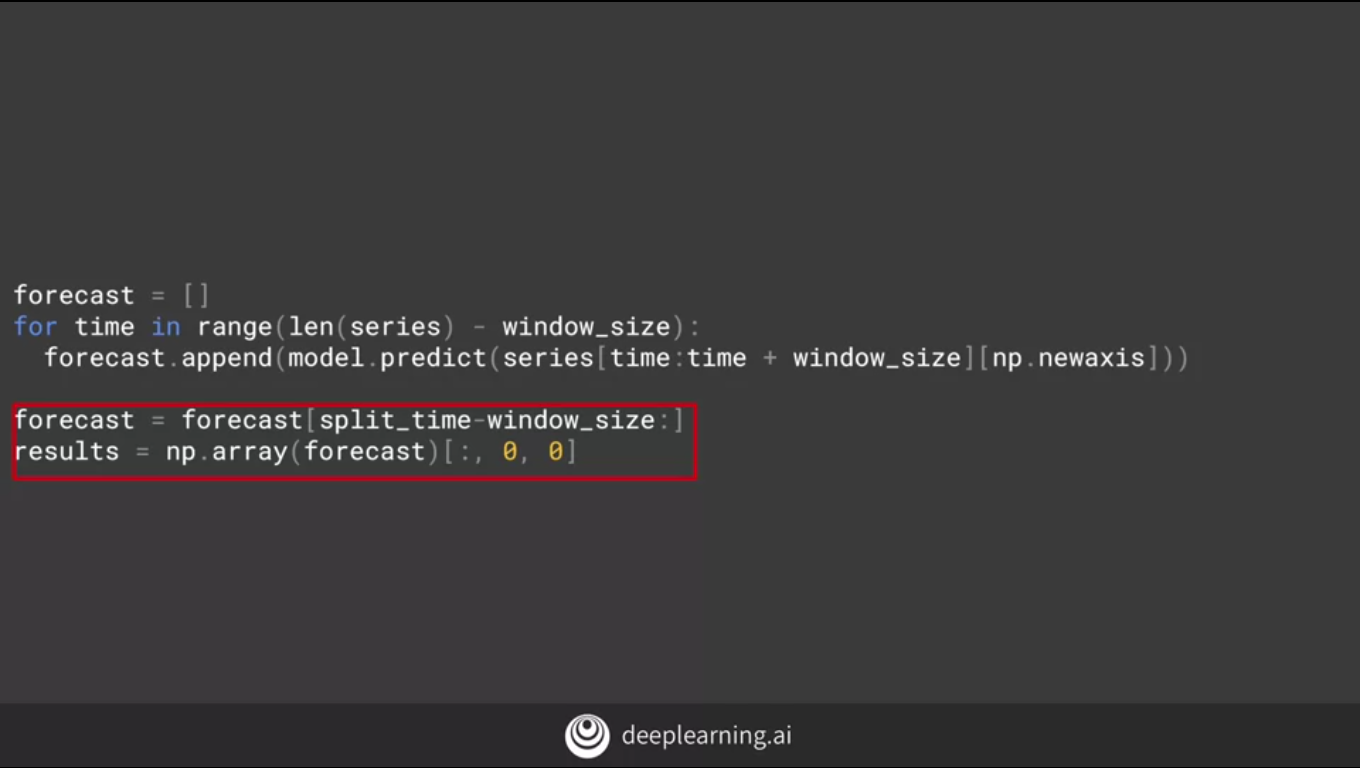
So if you think back to this diagram and you consider the input window to be 20 values wide,

then let's call them x0, x1, x2, etc, all the way up to x19.

But let's be clear. That's not the value on the horizontal axis which is commonly called the x-axis, it's the value of the time series at that point on the horizontal axis.

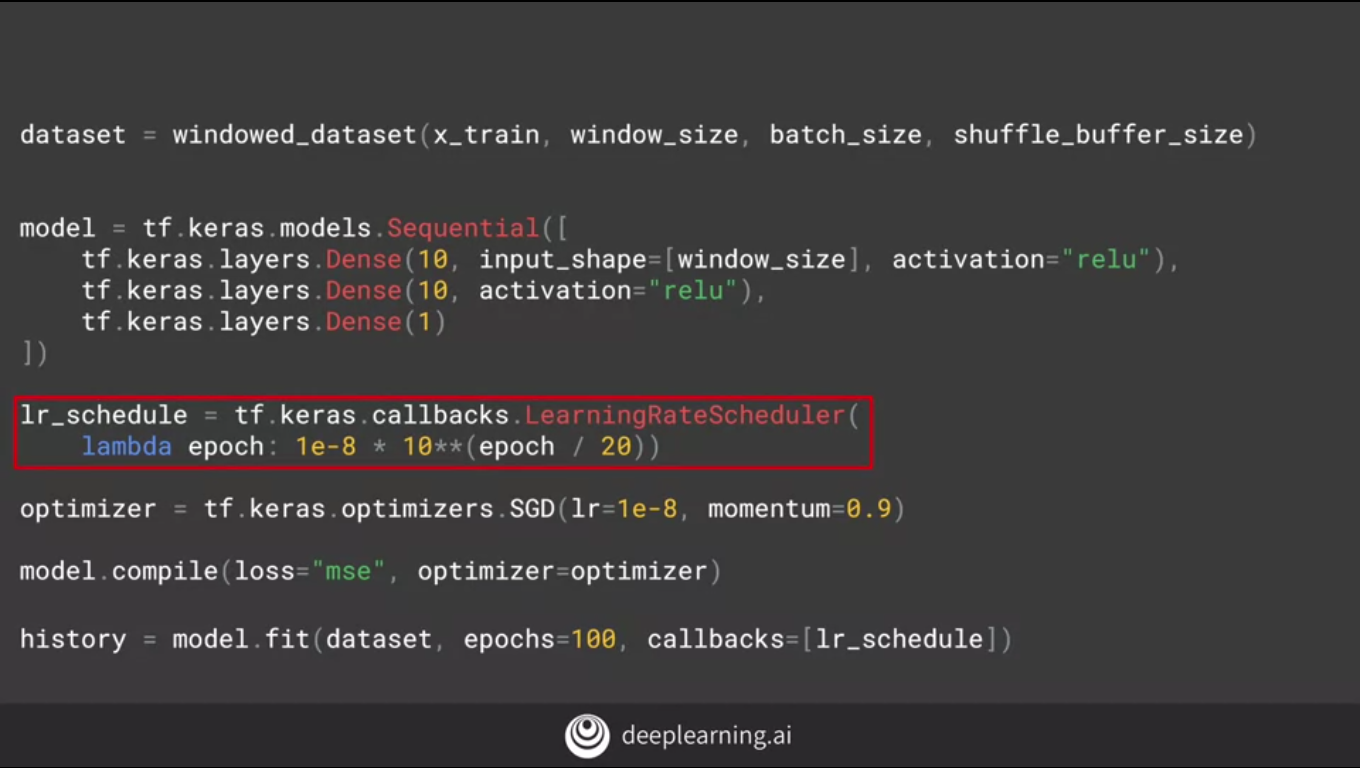
So the value at time t0, which is 20 steps before the current value is called x0,

and t1 is called x1, etc. Similarly, for the output, which we would then consider to be the value at the current time to be the y.



We create an empty list of forecasts and then iterate over the series taking slices and window size, predicting them, and adding the results to the forecast list. We had split our time series

into training and testing sense taking everything before a certain time is training and the rest is validation. So we'll just take the forecasts after the split time and load them into a NuimPy array for charting.



You can see that code here. This will be called at the callback at the end of each epoch.

What it will do is change the learning rates to a value based on the epoch number.

So in epoch 1, it is 1 times 10 to the -8 times 10 to the power of 1 over 20. And by the time we reach the 100 epoch, it'll be 1 times 10 to the -8 times 10 to the power of 5, and that's 100 over 20. This will happen on each callback because we set it in the callbacks parameter of modeled outfit.

# **Exercise 2 - Predict with a DNN**

**In class you saw how to split a dataset, and how to start training a DNN using it. For this exercise you’ll create your own synthetic dataset -- I’ve plotted a chart for what it should look like, see if you can figure out the parameters that get this series.**

**Once you have your series, you’ll create a DNN to predict values for that series!**