**Sequential Data:**

Slide 2:

Okay, so we’re going to first look at a different type of data, called sequential data.

Slide 3:

In the neural networks workshop, our data was a set of observations. Where each observation had a set of features and a target variable.

Slide 4:

Now, our data is a set of values that are organised in a specific order. In this case, our data is a set of stock prices that are sorted by their date. This is actually called a time series, which is a type of sequential data that is sorted by time.

Slide 5:

So how do we feed our sequential data into a machine learning model? Well, we take every x consecutive values and turn them into observation. Where these values become the observation’s features and the next value in the sequence becomes the observation’s target variable. In other words, we use consecutive values in a sequence to try and predict the next value in that sequence. In this case, we took every 4 consecutive values. Any questions so far?

**Recurrent Neural Networks:**

Slide 6:

Okay, so the model that we’re going to be building today is called a recurrent neural network.

Slide 7:

In the neural networks workshop, we looked at a feed-forward neural network. Where the output of each neuron is passed forwards.

Slide 8:

With a recurrent neural network, the output of a neuron can now be passed back into itself. This type of neuron is called a recurrent neuron. We’re going to look at a different way of visualising this, if this doesn’t make much sense to you.

Slide 9:

Here, the 2nd row represents the recurrent neuron feeding its output back into itself. Which is why each layer’s weights are the same. This is called unrolling an RNN.

Slide 10:

The recurrent neuron can actually keep on feeding its output back into itself. The more a recurrent neuron does this, the more rows the unrolled RNN has. Each time a recurrent neuron feeds its output back into itself, it also needs a new input. However, as you can see, its output can still be passed forwards. This makes RNNs useful for working with sequential data, as they process data sequentially which allows them to use the previous processed data to process the current data.

Slide 11:

So how does a recurrent neuron work? Well, it works just like a normal neuron, however, it only receives 2 inputs. It receives its previous output and the output of the neuron in the previous layer, as its inputs. Any questions?

**Types Of RNNs:**

Slide 12:

Okay, there are actually 4 types of RNNs.

Slide 13:

There is a one to one RNN. It takes in only 1 input and outputs only 1 output. This is known as a vanilla neural network, it’s not really an RNN. It’s, basically, just a really simple feed-forward neural network.

Slide 14:

There is a one to many RNN. It takes in only 1 input and outputs many outputs. Notice how the inputs start from the top.

Slide 15:

There is a many to one RNN. It takes in many inputs and outputs only 1 output. Notice how the outputs start from the bottom.

Slide 16:

There is a many to many RNN. This is the one that we were looking at before. It takes in many inputs and outputs many outputs.

Slide 17:

Here is another many to many RNN. As we can see, a many to many RNN doesn’t actually need to have the maximum amount of inputs and outputs. It also doesn’t need to have as many outputs as it has inputs either.

Slide 18:

Okay, so we’ve covered the model that we’re going to be building later. We’re now going to take a quick break and play a Kahoot.

**[Play Kahoot]**.

Kahoot Link:

<https://create.kahoot.it/share/rnn-kahoot/bd73121e-bb69-405c-a4b7-6c5e95f11217>

**Deep RNN:**

Slide 19:

Okay, so let’s look at a deep RNN. A deep RNN is just an RNN with multiple hidden layers.

Slide 20:

As we can see, a deep RNN can have multiple recurrent neurons. If this doesn’t make much sense to you, then let’s unroll this RNN.

Slide 21:

Here, we’ve unrolled our deep RNN into a many to one RNN. It works just like our previous many to one RNN, just on a larger scale.

Slide 22:

Here’s a real life example. Here, we’re using an RNN to predict tomorrow’s stock price, using the stock prices from today and the last 4 days.

Slide 23:

RNNs can be used for several things. They can be used for speech recognition, as speech are sequences of sounds. They can also be used for video tagging, as videos are sequences of frames. They’re also used in Google Translate and name entity recognition, as sentences are sequences of words. Any questions?

**Recap:**

Slide 24:

So, to recap what we’ve learnt.

Slide 25:

We’ve learnt what sequential data is. We’ve learn how a recurrent neural network works. We’ve learnt about the different types of RNNs. And we’ve learnt about the possible usages for a RNN.

**Workshop Info:**

Slide 26:

Now onto the fun part, let’s build an RNN.

Slide 27:

Okay, so we’re going to try and train an RNN to predict tomorrow’s highest Telsa stock price. We’re using the Tesla Stock Price dataset, that I used earlier as an example, to do this.

Datset Link:

<https://www.kaggle.com/datasets/jillanisofttech/tesla-stock-price?select=Tasla_Stock_Updated_V2.csv>