**Images:**

Slide 2:

**[Skip Slide]**.

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, each observation has an image and a target variable. In this case, an observation’s image is an animal and its target variable is the type of animal. Where 0 represents a dog and 1 represents a cat. With this type of data, our machine learning model uses an observation’s image to try and predict its target variable.

Slide 5:

So how do we feed an image into a machine learning model? Well, an image is actually stored as an array of pixel values. The greater the value of a pixel, the lighter the shade of the pixel. If an image has colour, then it’s usually stored as an RGB image. This is an image made up of 3 colour channels. A red channel, a green channel and a blue channel. As red, green and blue can be used to create any colour. An RGB image is stored as an array of pixel values, with 3 channels. This array of pixel values is what we feed into our machine learning model. Any questions so far?

**Convolutional Neural Networks:**

Slide 6:

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

Slide 7:

In the neural networks workshop, we looked at a feed-forward neural network. This network was made up of an input layer, hidden layers and an output layer.

Slide 8:

With a convolutional neural network, the network also contains a convolutional layer and a pooling layer. The convolutional layer extracts a feature, from the image, into a feature map. A feature can be an edge, a texture or part of an object. The pooling layer then reduces the dimensions of the feature map, while retaining essential information. The feature map is then flattened and fed into a fully-connected layer. In this case, the feature map is flattened into 6, 9, 5, 9, 8, 6, 9, 8 and 8. A fully-connected layer is just a normal feed-forward neural network. This layer classifies the image based on the extracted feature.

Slide 9:

Here’s a real life example. Here, we’re using a CNN to classify an image of an animal based on the features within the image. There are 3 final outputs. Each is a different type of animal. The prediction 0 means that it’s not that animal and the prediction 1 means that it is that animal.

**Convolutional Layer:**

Slide 10:

**[Skip Slide]**.

Slide 11:

Okay, so a convolutional layer extracts a feature, from the image, into a feature map, but how does it work?

Slide 12:

Well, it first performs a convolution operation. A convolution operation sweeps a weighted 3x3 filter across the output of the previous layer. This produces a feature map. This filter has to have as many channels as the output of the previous layer. In this case, the filter has to have 3 channels. This filter is also known as a feature extractor or the Sobel kernel.

Slide 13:

Okay, so how does the filter actually work? Well, we first place the filter on the top left of our input. We then multiply each matching value together and add the results all up. We then add a bias to this. This gives us the top left value of the resulting feature map. We then move the filter 1 to the right and repeat this whole process again. We keep doing this until we’ve filled up the resulting feature map.

Slide 14:

Once the convolutional layer has performed a convolution operation, it applies an activation function to the resulting feature map. In this case, we’ve applied the ReLU activation function as all the negative values have been mapped to 0. And that’s how the convolutional layer works.

**Pooling Layer:**

Slide 15:

**[Skip Slide]**.

Slide 16:

Okay, so a pooling layer reduces the dimensions of a feature map, while retaining essential information, but how does it work?

Slide 17:

Well, it performs a pooling operation. A pooling operation sweeps an unweighted 2x2 filter across the output of the previous layer. This produces another feature map. This filter has to have as many channels as the output of the previous layer. In this case, the filter has to have 1 channel.

Slide 18:

Okay, so how does the filter actually work? Well, just like the convolution operation, we place the filter on the top left of our input. However, this time, we choose the highest value within the filter. We repeat this until we’ve filled up the resulting feature map.

Slide 19:

And that’s how the pooling layer works.

Slide 20:

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/cnn-kahoot/b7e5fca3-7eb8-4c60-8df8-ffd75f2eca60>

**Convolutional Layer With Multiple Filters:**

Slide 21:

**[Skip Slide]**.

Slide 22:

The convolutional layer that we’ve looked at so far only applied 1 filter to its input.

Slide 23:

However, a convolution layer can actually apply multiple filters to its input. This causes the convolutional layer to produce multiple feature maps. One for each filter.

Slide 24:

If we take a step back, this is how our CNN would look like if its convolutional layer used multiple filters.

Slide 25:

So the more filters a convolutional layer uses, the more feature maps it produces. In other words, the more features it extracts from the image. Unlike before, where it only extracted 1 feature from the image. This improves the performance of our CNN.

**Convolutional Neural Network With Multiple Layers:**

Slide 26:

**[Skip Slide]**.

Slide 27:

The CNN that we’ve looked at so far only had 1 convolutional layer and 1 pooling layer.

Slide 28:

However, a convolutional layer can actually be followed by more convolutional layers. In this case, it’s followed by 1 more.

Slide 29:

A CNN can also contain several of these blocks. Where a block is 1 or more convolutional layers followed by a pooling layer.

Slide 30:

Each convolutional layer actually extracts features from the features extracted by previous convolutional layers. This creates a hierarchy of features.

**Recap:**

Slide 31:

So, to recap what we’ve learnt.

Slide 32:

We’ve learnt how an image is stored. We’ve learn what a convolutional neural network is used for. We’ve learnt how the convolutional and pooling layers work. And we’ve learn how a convolutional layer can extract multiple features from an image.

**Workshop Info:**

Slide 33:

Now onto the fun part, let’s build a CNN.

Slide 34:

Okay, so we’re going to try and train a CNN to classify an image based on the Simpsons character in it. We’re using dataset with images on 47 Simpsons characters.

Datset Link:

<https://www.kaggle.com/code/paultimothymooney/simpsons-characters-dataset-with-fastai-v1/input?select=simpsons_dataset>

Note:

Use Google Colab for this workshop. Go on “Change runtime type” and select the “T4 GPU” as your hardware accelerator. This GPU is free and reduces the model’s training time from 10 hours to 10 minutes.