Exploring Loss Functions and Optimizers

To this point, you've been largely guided through different loss functions and optimizers that you can use when training a network.

It's good to explore them for yourself, as well as understanding how to declare them in TensorFlow, particularly those that can accept parameters!

Note that there are generally 2 ways that you can declare these functions -- by name, in a string literal, or by object, by defining the class name of the function you want to use.

Here's an example of doing it by name:

```
optimizer = 'adam'
```

And one of doing it using the functional syntax

```
from tensorflow.keras.optimizers import Adam
opt = Adam(learning_rate=0.001)
optimizer = opt
```

Using the former method is obviously quicker and easier, and you don't need any imports, which can be easy to forget, in particular if you're copying and pasting code from elsewhere! Using the latter has the distinct advantage of letting you set internal hyperparameters, such as the learning rate, giving you more fine-grained control over how your network learns.

You can learn more about the suite of **optimizers** in TensorFlow at https://www.tensorflow.org/api_docs/python/tf/keras/optimizers -- to this point you've seen SGD, RMSProp and Adam, and I'd recommend you read up on what they do. After that, consider reading into some of the others, in particular the enhancements to the Adam algorithm that are available.

Similarly you can learn about the **loss functions** in TensorFlow at https://www.tensorflow.org/api_docs/python/tf/keras/losses, and to this point you've seen Mean Squared Error, Binary CrossEntropy and Categorical CrossEntropy. Read into them to see how they work, and also look into some of the others that are enhancements to these.

When you're done, go back to some of the exercises or other colabs that you've done, and experiment with different loss functions or optimizers. In particular tweak things like the learning rate or other parameters to see if you can improve your models!