**Accuracy and Loss**

**Sigmoid Cross Entropy**

While developing models we use many functions to check the model’s accuracy and loss. For a model to be in good condition, loss calculation is a must since loss acts like a penalty. The lower the loss better will be the working of the model.

There are many kinds of loss functions. One such function is the Sigmoid cross entropy function of TensorFlow. The [sigmoid function](https://www.geeksforgeeks.org/python-tensorflow-nn-sigmoid/) or [logistic function](https://www.geeksforgeeks.org/how-to-compute-the-logistic-sigmoid-function-of-tensor-elements-in-pytorch/) is the function that generates an S-shaped curve. This function is used to predict probabilities therefore, the range of this function lies between 0 and 1.

[Cross Entropy loss](https://www.geeksforgeeks.org/cross-entropy-cost-functions-used-in-classification/) is the difference between the actual and the expected outputs. This is also known as the log loss function and is one of the most valuable techniques in the field of Machine Learning.

**CNN Architecture**

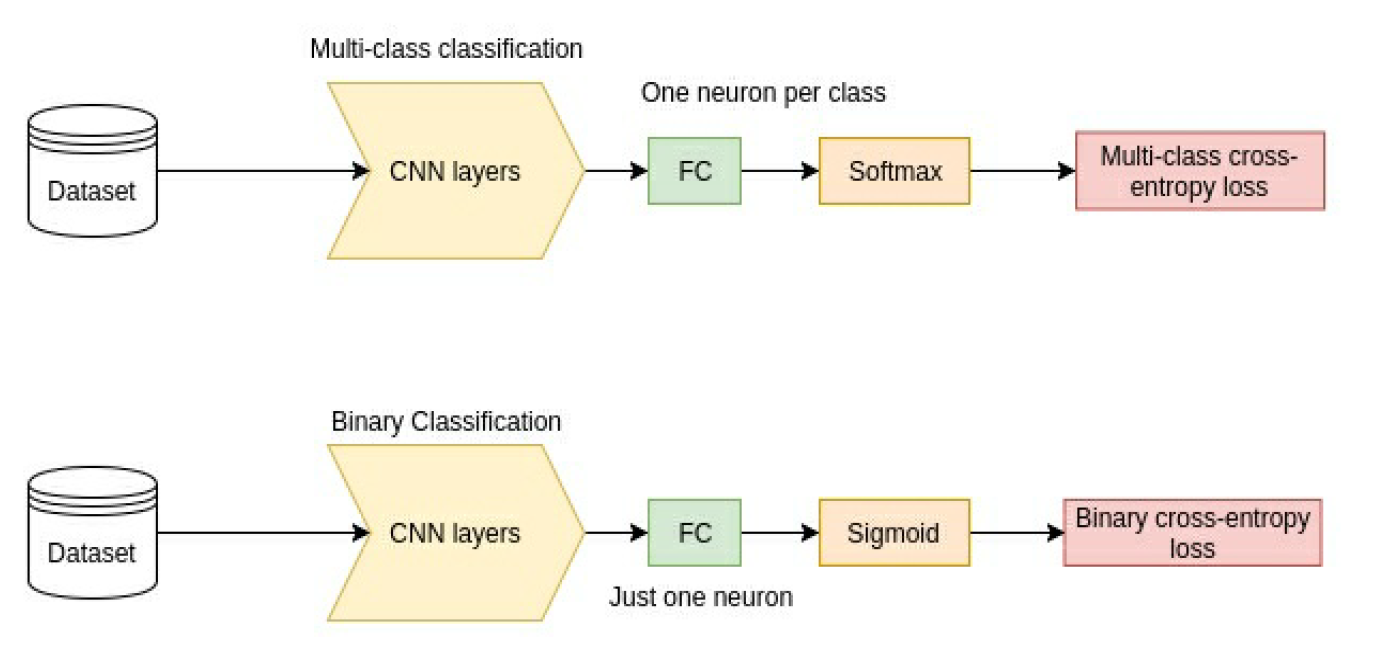
The crucial part of an image classification model is its CNN layers. These layers will be responsible for

extracting features from image data. The output of these CNN layers will be a feature vector, which like before, we can use as input for the classifier of our choice. For many CNN models, the classifier will be just a fully connected layer attached to the output of our CNN.

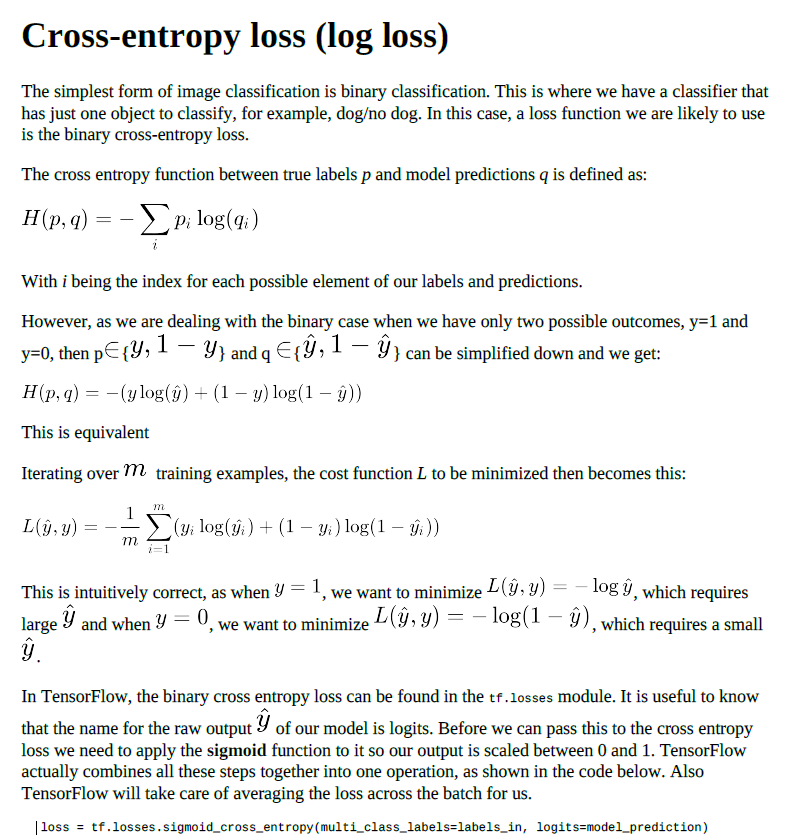
It is important to note that at its core, the CNN architecture used in classification or a regression problem such as localization would be the same.

The only real difference will be what happens after the CNN layers have done their feature extraction.

For example, one difference could be the loss function used for different tasks, as it is shown below:



Different problems that CNNs can be used to solve. It will become apparent that lots of tasks involving images can be solved using a CNN to extract some meaningful feature vector from the input data, which will then be manipulated in some way and fed to different loss functions, depending on the task. For now, let’s crack on and focus firstly on the task of image classification by looking at the loss functions commonly used for it.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p | q | log(q) | p.log(q) | -(1-p)\*LOG(1-q) |
| 1 | 0.8 | -0.09691 | 0.09691 |  |
| 0 | 0.2 | -0.69897 |  | 0.096910013 |
| 0 | 0.4 | -0.39794 |  | 0.22184875 |

