# Workshop 3

In this workshop you will be creating a Convolutional Neural Network using Keras. We are then going to train this to recognise hand-written digits.

Before you start, you should complete workshop 2. It will also help if you have a basic understanding of Keras (<https://keras.io/>) and NumPy (<http://www.numpy.org/>), scikit-Learn (<http://scikit-learn.org>), and **matplotlib** (<https://matplotlib.org/users/tutorials.html>). Just working through the quick-start guides will be enough.

## Getting Started

1. Start working through the following tutorial: <http://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/>   
   But stop when you have the MLP working. i.e. before you get to the section “Simple Convolutional Neural Network for MNIST”.

It takes a while to train on this dataset, so we’re not going to run k-folds validation yet, but you should be sceptical of all results until you do.

At this point, it’s useful to get some idea of the MLP performance beyond just the numbers. Visualisation is a very important part of machine learning research.

The first thing you might like to know is: “how bad is it?” Perhaps the data is just really poor and we can’t expect the network to perform better.

1. Write some code to show the examples that were incorrectly classified
   * Use model.predict to get the predictions on the test set
   * Loop through the each prediction to check if it’s correct using numpy.argmax
   * If it’s not correct,
     + Print the incorrect answer
     + Plot the image using plt.imshow and plt.show (you will need an original version of the image for this - not the reshaped version - so make a copy before you reshape)

You should see that some of the errors are indeed reasonable mistakes, but the network could definitely do better.

1. Next start the “Simple Convolutional Neural Network for MNIST” section of the tutorial

This can take a while to run, so depending on how far you get, you might want to skip the next section.

1. Either: repeat for the “Larger Convolutional Neural Network” described in the tutorial; or, examine the effect of one of the following:
   * Changing dropout rate
   * Try some other architectures/layers
   * Change the size of the convolution kernel
   * Different pool size
   * Different pool type
   * Different batch sizes

That is all you will have time for in this workshop, but if you have time on the weekend, you should try a more interesting dataset by working through this tutorial: <http://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/>