CNN Tutorial Questions

* What type of data are they using?
  + They are using images.
* What conversions (if any) had to be done to the data before it could be put into the neural network?
  + For the labels, OHE was used. For the regular neural network, the image data was converted to float32, reshaped into “individual vectors”, and then 0-1 normalized. For the CNN, the image data was converted to float32 and then 0-1 normalized (they weren’t reshaped).
* What is the output of the neural network, both in terms of what it looks like to the computer (e.g. integers in the range [0-2]) and how humans should interpret it (e.g. the type of iris)?
  + The output of the neural network would a probability (floats from 0-1) for each of the classes the image may be classified as.
  + A human interprets the highest probability as the class the image has been classified as.
* How many hidden layers does the network have, and what type are they (e.g. fully connected, convolutional, recurrent, LSTM, sparse, etc.)?
  + The layer has four hidden layers, the first two are convolutional, and the final two are dense (fully connected).
* What activation function(s) does it use?
  + Activation functions used: ReLU and softmax.
* What loss or cost function is it using?
  + The loss (cost) function used was categorical cross-entropy.
* What kind of validation (if any) are they using?
  + The type of validation used was accuracy.
* What other validation methods might work for this type of problem?
  + Other validation methods that may work for this type of problem include precision, recall, and f-measure.
* Why do you think the authors may have chosen this architecture for their network?
  + They first created a simple neural network to illustrate the concept, then created a basic CNN so that the model would be able to perform feature-based recognition, then created a CNN with more layers to improve performance, and finally added dropout to reduce overfitting.
* Are there any changes you might try, if you were going to improve on their model?
  + I would add more hidden layers as well as dropout because the model is still significantly overfitting.