July 20

I. Responses
   A. Write a Problem Statement. Introduce your topic, quantify its significance, and describe the problem as a process. Identify and quantify significant obstacles to solving your problem. Demonstrate why your topic is important, and why the obstacles associated with your topic are significant both globally as well as within the context of your selected application. Describe and analyze the complex nature of the process you are investigating, including the system, the environment, agents and networks. Describe and analyze scope, scale and hierarchy of processes and sub-processes. Describe and analyze factors that contribute to quantified obstacles. Describe and analyze process oriented causes-effect relationships.
   B. Up next for your final project: *data* (then after that: *methods*)
   C. Go to the website http://cs229.stanford.edu/proj2017/ and review some of the posters for the University of Stanford's Fall 2017 machine learning course. Consider how you will design and present your final project as a poster.
   D. Describe your implementation of the cats & dogs exercise. How did you setup the data?
      1. Which optimizer have you selected, and how might it compare to other possible choices? (have a look at this site - https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a)
      2. Describe your selected loss function and it's implementation. How is it effectively penalizing bad predictions? (have a look at this site - https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a)
      3. What is the purpose of the metric= argument in your model.compile() function? (look here - https://keras.io/api/metrics/)
      4. Plot the accuracy and loss results for both the training and test datasets. Include these in your response. Assess the model and describe how good you think it performed.
      5. Use the model to predict 3 dog images and 3 cat images. Upload you images and the prediction. How did your model perform in practice? Do you have any ideas of how to improve the model's performance?