1. Describe K-fold cross-validation.

i) Describe K-fold Cross-Validation: Randomly divide the data into K equal-sized parts. Leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictions for the left-out kth part. This is done in turn for each part $k = 1, 2, \ldots, K$, and then the results are combined. More specifically, let the K parts be $C_1, \ldots, C_K$, where C_k denotes the indices of the observations in part k. There are n_k observations in part k: if N is a multiple of K, then $n_k = N/K$. Then we compute:

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{N} MSE_k$$

where $MSE_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$, and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed.

ii) What it is used for: This is widely used for estimating test error. The estimates can be used to select best model, and to give an idea of the test error of the final chosen model.

iii) What are the advantages/disadvantages of using more folds (increasing K)?
Advantage: When you increase K, you will use more data in your training set, so this will help with the overestimation of test error for the model fit of the validation set. Disadvantage: The estimates from each fold are more correlated as K increases, and hence their average can have high variance. Also since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward if we increase K.

iv) When does cross-validation estimate the performance of the actual predicting function being used?
With one-fold the model is validated on a left-out test set. This is an estimate of the performance of that actual model being used for prediction. With k-fold cross-validation a different model is being estimated for each fold, none of which are being used for actual future prediction. What one estimates is the expected performance of the model building procedure over the randomization induced by the cross-validation procedure. As we increase K and choose K close to N, the training set on each fold becomes almost equal to the entire training set, and the performance is almost identical to that of the model used for prediction.

2. Single output neural networks.

i) What are the relative advantages/disadvantages of these two respective approaches.

1. train separate single output neural networks for each outcome:

Advantage: Usually take less time for training. May work better when the models for different outcomes follow quite different patterns. Disadvantage: Less efficient in capturing the relationship of the outcomes. Harder for modeling.

2. train a single network with multiple outputs: Advantage: Probably more effective if the outputs are related and dependent. Easier for modeling. Disadvantage: May probably need to train longer. May not work well if for different outcomes the data follows very different patterns.

ii) In what situations would one expect each to be better than the other.

1. train separate single output neural networks for each outcome: When the outcome models follow very different patterns.

2. train a single network with multiple outputs: When the outcomes are strongly correlated with each other. Like K-classification problem.