Question 1

The first chart is classification model. To be more specific, it is a confusion model, which work for classification neural network, letting us learn the result of the difference between true label and predicted label. In this chart, the left-right (x) axis are the predictions, the top-bottom (y) are the expected outcomes, and there are four labels existing. For label Apple, in the top left of the chart, it is darkest blue, which means the indication truth of ‘Apple’ almost always meet the prediction ‘Apple’, the misclassification rarely happens. For the label ‘Banana’ it is the same. However, for label ‘Grape’, the misclassification happens at high rate, and it is the worst in the four labels. It is easily to be classified as cherries. And for label ‘Cherries’, the misclassification also happens a lot, because it has nearly half the probability to be classified as grapes. A perfect model should have dark blue diagonal running from top-left to bottom-right.

The second chart is regression model. To be more specific, it is lift curve, which has two lines, one for expected and one for prediction. The x-axis means 0 to 100% of the dataset, and the y-axis is ranged according to the values predicted. It tells us how different is one line from the other. The predict line is first above the expected one, then drops down to underlies the expected one, which demonstrate the difference of result when data in the dataset pours in. If it is a good prediction, the expected line should be close to predict line.

I choose accuracy to numerically measure the first since it is classification, and RMSE to numerically measure the second chart since it is regression.

Question 2

I would use the following functions for encoding values:

	Input	Output
Age	encode_numeric_zscore	No need to encode
Favorite Color	encode_text_dummy	encode_text_index

Length	encode_numeric_zscore	No need to encode
Gender	encode_text_dummy	encode_text_index

From the above, it is obvious that it is different using the value as input from as output. To be more generally, when data is used as input -- function missing_median will be used to fill missing data, and function encode_text_dummy will be utilized to encode text or category fields.

When data is used as output -- rows with missing output will be discarded, and function encode_text_index will be utilized to encode text or category values. Apart from this, numeric values of output have no need to be encoded.

The problem we may encounter is we may find filling missing zip code values with function missing_median is worthless because each region has a zip code, and we cannot just map a median value with a place.

We may relate the zip code data to a range of latitude and longitude, and try to fill the missing data with median of latitude and the median of longitude.

Question 3

We cannot always get enough test data which comes from different training methods and algorithms, so we need validation for better model and out-of ample prediction. For holdout set, it is set aside before cross validation, when we have big amount of data, and will be the final evaluation before we utilize our model for real-world use. It is valuable when the data is in considerable amount. For K-fold Cross validation, it could generate out of sample predictions on the entire dataset by utilizing a number of models equal to the folds. This is intelligent because even though simply setting more validation data will make us more sure about if our model is precise, it means less training data and make more out of sample error. K-fold uses the iteration to give us enough training data and entire data set as validation set, so it is clever.

Out of sample predictions are important because when new data is obtained, we need to predict the data with the model we construct or the model we choose from a set of models. We need out of sample predictions to compare our prediction with the existing true ones to evaluate our model and decide its accuracy. Only by doing that, we can know to what extend the out of sample prediction from the model could be trusted when new data is given.

Overfitting occurs when a neural network is trained to the point that it begins to memorize rather than generalize. It means the model we have fits the data more than is warranted, fitting the noise of the data to an extent that affects our result seriously. Learning is led astray by fitting the noise more than the signal. And as a result, although the in-sample error of one data set is reduced to zero, it will not perfectly fit other new datasets as well, and in fact it usually leads to the opposite effect.

To prevent it, we should limit the complexity of our model. Validation and regularization are two useful and usually utilized methods to prevent overfitting.

Question 4

For backpropagation training, learning rate is very important. And setting it can be complicated：

Using too high of a learning rate may either fail outright, or come to a higher error than a fitting learning rate.

Using too low of a learning rate will make the process very slow, even though it usually comes to a good solution.

Momentum is utilized in in backpropagation, adding the current weight change amount (v_t) with the scaled value of the previous weight change amount (v_{t-1}). An usually used value for momentum is 0.9.

ADAM estimates the first (mean) and second (variance) moments to determine the weight corrections. It begins with an exponentially decaying average of past gradients (m), which reaches a similar goal as classic momentum update, though its value is calculated automatically based on the current gradient (g_t). What is more, ADAM is very tolerant to initial learning rate (η) and other training parameters.

Question 5

The goal of classification is to predict the target class into -1/1 (positive or negative). Hence an example could be: when a man is applying for a credit card in a bank, the staff of the bank collect his data, like his age, work year and liability situation (testing data). Then they analyze the data of the past application persons (training data) to predict if this new application is qualified or not.

The goal of regression model is to generate numeric prediction. Hence an example could be: the bank staff use the data collected from the new applicant (test data), like his age, work year and liability situation, and analyze the data of the past applicants with the amount of credit they got if they were approved (train data) to make prediction of how much credit the new applicant could get.

For classification problems, we should compare the prediction class with the expected class, which means calculating the classification accuracy together with classification log loss. If there are high accuracy and low log loss, it should be evaluated as wanted.

For regression problems, we should compare the predicted value with the expected value, which means calculating the RMSE to evaluate the neural network. If the RMSE is small, it should be evaluated as wanted.

Apart from the above, K-Fold cross validation could be utilized for evaluating both types of the problems. We will get an out of sample error, and if the error is small, it should be evaluated as wanted.