(e)
I use the image-based features and question-based features as the input data. To extract these features, I use Microsoft Azure API (computer vision and text analytics) to process all the VizWiz training dataset, 2000 rows of validation dataset, and 100 rows of test dataset. I use all the image tags from Azure Vision API, connect them word by word and store it as image feature. Then I use Text Analytics API to extract key phrases of each question and store it as question-based feature.
After getting all necessary data, I create the following dataset for training and testing. There are three image feature datasets of training, validation and testing. Three question feature datasets of training, validation and testing. Two output datasets of training and validation. Then I combine image and question feature datasets together and use one-hot to encode. That's how I prepare all data following the following model training.
First I use PCA to reduce the dimension of input data. Then I train the neural network, logistic regression, bagging, boost, and SVM models using the training data. During this process, I tuned the model parameters to make sure each model works well and has a relatively accuracy on the validation data. After that, I choose the one with the highest accuracy (some of them has similar accuracy, I just randomly pick one among them) and make prediction of the test dataset using that model.

(f)
I tried neural network, logistic regression, bagging, AdaBoost, SVM models. I trained the model with 20000 rows of training data, then tested the model on 2000 rows of validation data. The accuracy of different models are as follows. MLPClassifier: 0.64; Logistic Regression: 0.71; bagging: 0.64; boost: 0.71; SVM: 0.71. According to the accuracy on validation data, I choose SVM to make prediction on the 100 examples from VizWiz_test000000020100.jpg to VizWiz_test_000000020199.jpg.
Here are some hyperparameters of different models.
MLPClassifier: hidden_layer_sizes=(2048,4096,4096), max_iter=1000, random_state=42, activation='relu', solver='adam'
BaggingClassifier: max_samples=50
AdaBoostClassifier: base_estimator=DecisionTreeClassifier(max_depth=10), n_estimators=500,learning_rate=0.1
Logistic regression, boost, and SVM models have same accuracy on validation dataset, we can apply any of them to the 100 examples of the test split.

(c)
The optimal hyperparameter: number of hidden layer = 5, number of neurons per layer = 12
The number of weights is 784*12+12*12*4+12*10 = 10104

(d)
The performance of the neural network largely depends on the number of hidden layers and number of neurons per layer. We can improve the performance of the neural network by increasing the number of hidden layers and number of neurons per layer. If the number is too small, the model could be underfitting.
When the number of hidden layer and neurons per layer reach certain amount, the performance of the neural network won't improve too much if we keep enlarging that number. The appropriate number depends on the complexity of our training data. If the number is too big, it might cause the accuracy to decrease since it could be overfitting.