Yuan Hong (Bill) Sun
1003039838

MIE324 Assignment 1

Part 3: Understand and Visualize the data

1.

example_cat.png

example_dog.png

2. The variable length of "relevant_train_indices" is 8000, meaning there are 8000 training examples. There are 8000 training, 2000 validation, and 2000 testing examples each for combined cat and dog classes.

3. The validation set is needed to evaluate the results of the model after training on the training set. It is an intermediate step before finally evaluating the model using the testing set.

If the testing set was used to calculate error instead of the validation set, there would not be a fresh set of data left to evaluate the overall performance of the model. There would be less data to train to help correct potential errors.

4. It's important to have an equal number of training examples for both classes because the machine needs an equal amount of learning for each class in order to have an equal chance of classifying each type of input correctly.

If there are significantly more samples of one type of training data, the model would be more proficient at recognizing that one type of input; however, will have trouble recognizing the other type due to less exposure. Also, in a binary classification problem, less training on one type of input would also affect the accuracy of classifying the other type correctly, as the model could confuse between the two input types. It will increase the overall error rate of the model.

4. Training the Network

1.
Using default parameters:
Total time elapsed: 857.24 seconds
Average 17.145 seconds per epoch

2. a).

Learning Rate	**0.1**	**0.01**	**0.001**
Total time taken	923.15 s	906.24 s	802.28 s
Min training error	0.114625	0.0045	0.319375
Min validation err	0.279	0.2805	0.357

b).

Batch Size	**32**	**64**	**512**
Total time taken	872.79 s	857.24 s	759.85 s
Min training error	0.00	0.0045	0.300625
Min validation err	0.2835	0.2805	0.316

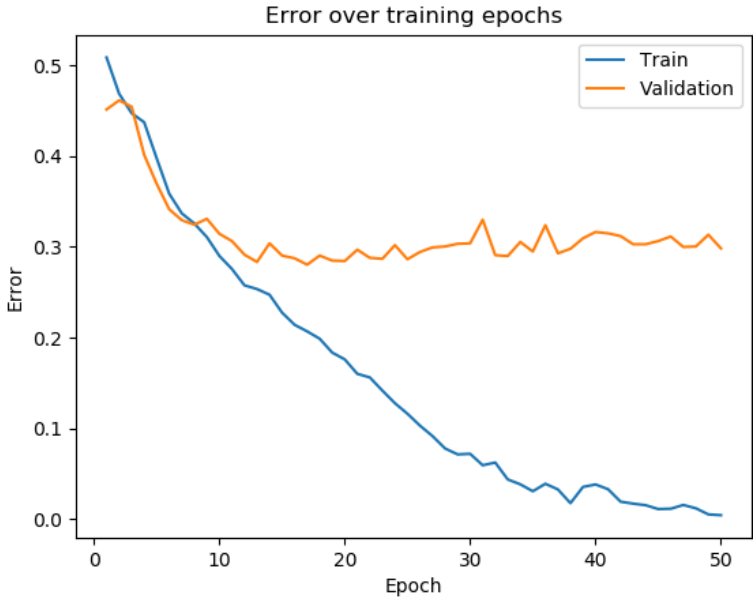
5. Evaluate the performance of the networks

5.1. Default Hyperparameter Settings

1.

This graph showed an increasing cross entropy loss for validation data (after about epoch 20) while showing a decreasing trend on the training data. This shows that the model does no longer performs any better on the validation data after about epoch 20, while it keeps performing better on the training data, which is a sign of overfitting. The model is simply memorizing training data without learning the patterns.

2.



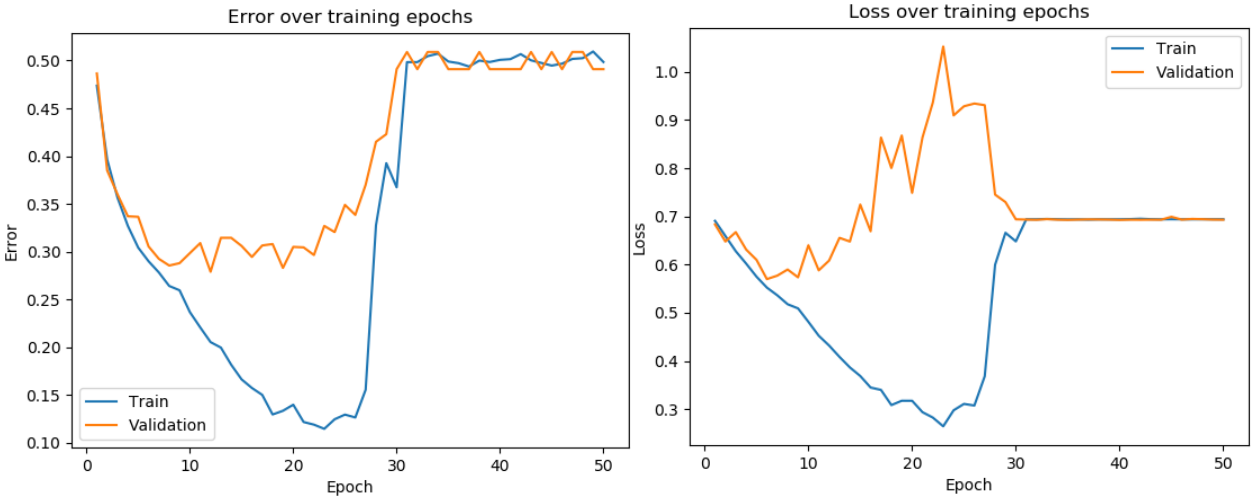
In this graph, the validation error curve stalled (after around epoch 15) while the training error kept decreasing. This differs from the graph above (loss over training epochs) in which the validation error no longer decreases but approaches a horizontal line. This shows that the model is overfitting the training data after about 15 epochs.

3. Final values:
Final training error: 0.0045
Final validation error: 0.2985
Final training loss: 0.01509
Final validation loss: 2.06526

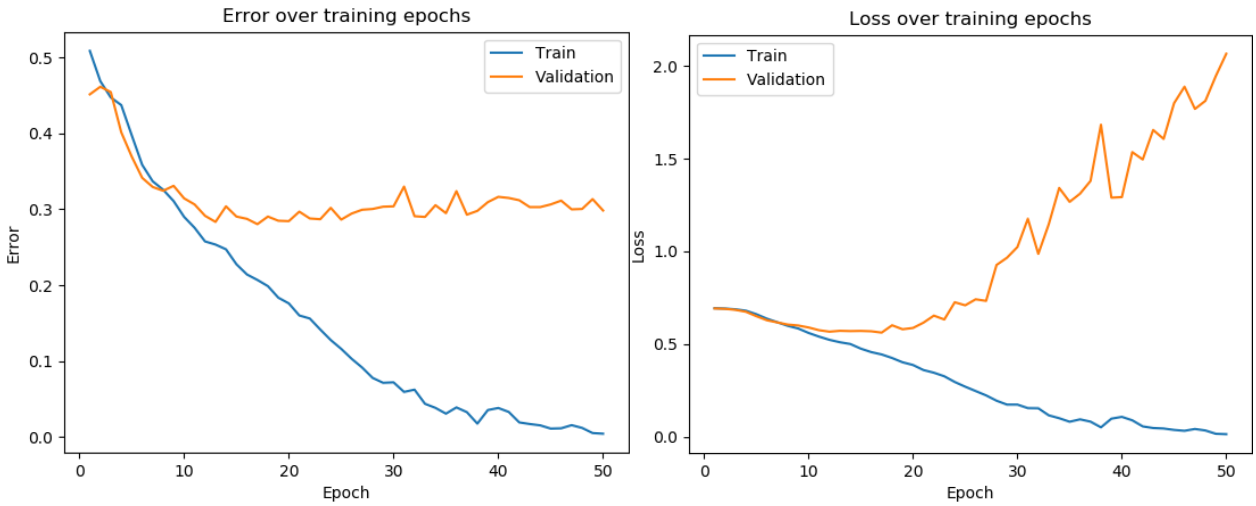
5.2. Evaluate the Hyperparameter Effects

1. Varying learning rates:

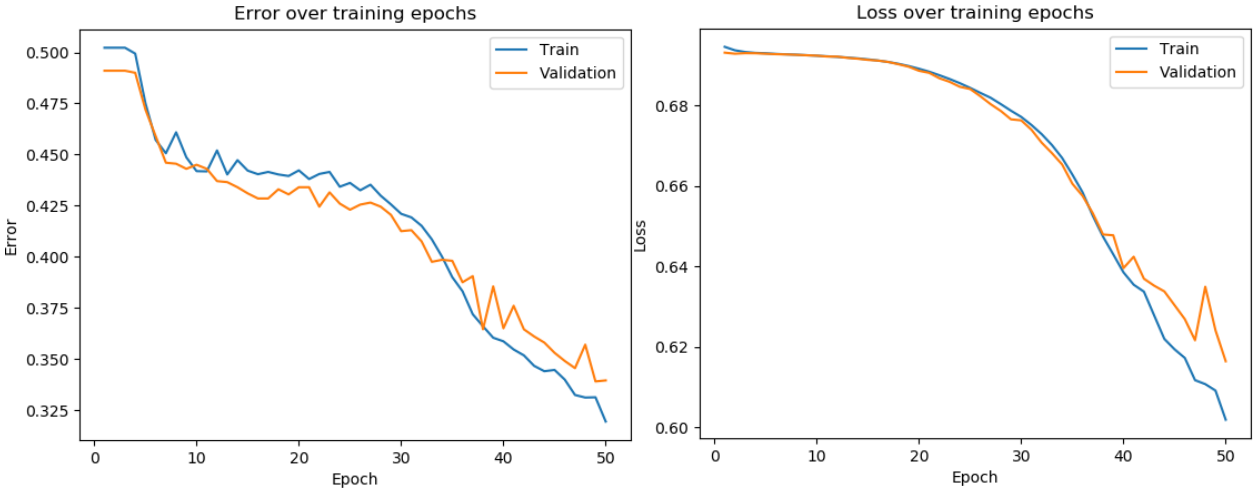
Learning rate = 0.1



Learning rate = 0.01



Learning rate = 0.001



At a high learning rate, the model seems to learn the patterns first (best at around epoch 20), but then starts to diverge and overcompensate, resulting in worsening values that reached the original untrained state (50% error). It performs a lot better temporarily on the training data.

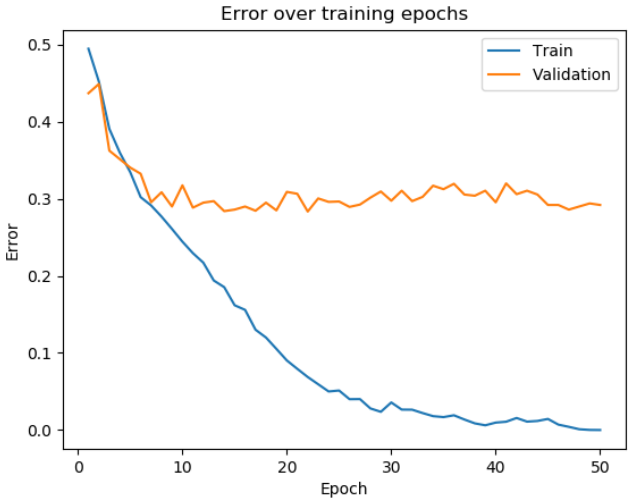
At the default learning rate, the model overfits on the training data, therefore resulting in no more improvement on the validation data after roughly epoch 15.

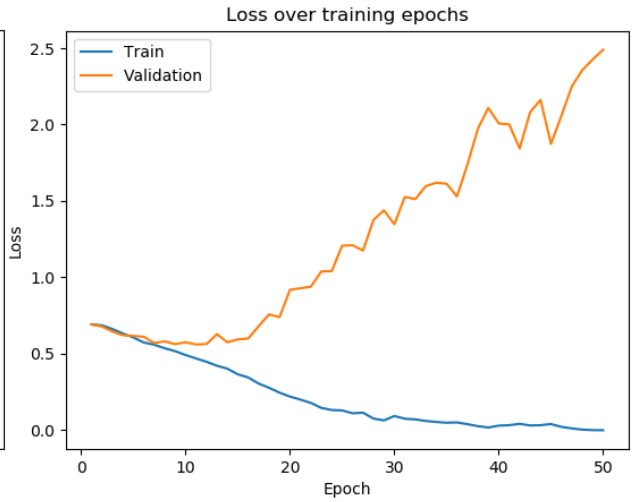
At a low learning rate, the model keeps improving until the last epoch, with both the training and validation errors improving at about the same rate. The model, however, did not reach anything close to optimal results after even 50 epochs (with training and validation error rate of well over 30%).

If the learning rate is too high, the model would overcompensate after each epoch and perform poorly after the first several epochs. If the learning rate is too low, the performance is inconclusive for now due to limited data, but it would require much more training epochs to train and optimize the model.

2. Varying batch sizes:

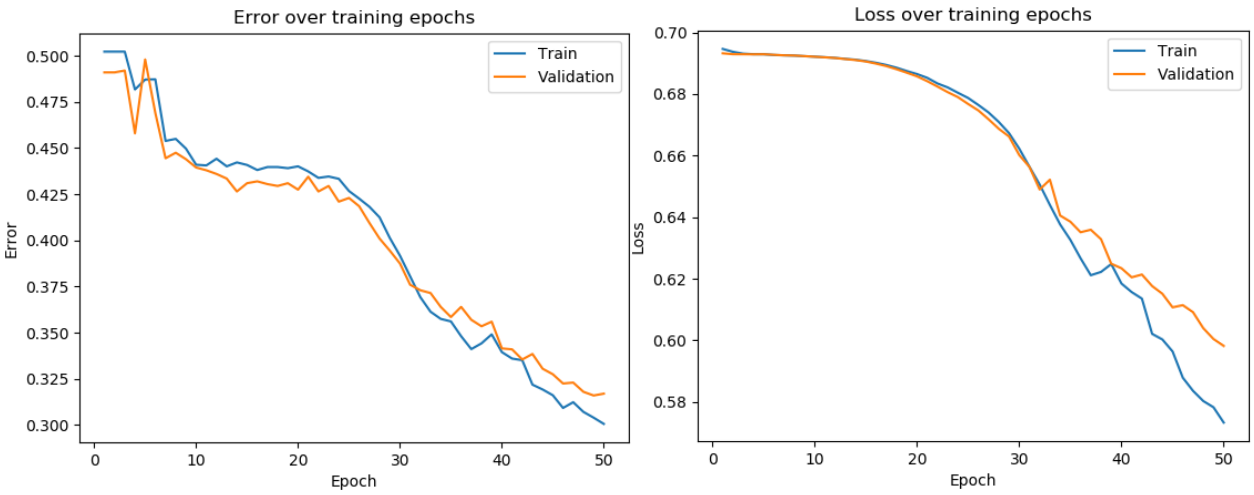
Batch size = 32 (The model stopped after 6 epochs running on batch size = 16, so 32 was used instead)





Batch size = 64

Batch size = 512



At a smaller batch size, the effects are similar to the default batch size, in which the model begins to overfit on the training data after about epoch 10, shown by the flatlining of the validation error rate and increase in validation loss. The magnitude of total loss is actually slightly more compared to the default batch size. Therefore, a smaller training batch size would make the model prone to overfitting (because less training samples per epoch).

At a large batch size, the effects closely resemble having a low learning rate. The model shows continuous improvement, however, it still had more than 30% error (both training and validation) after 50 epochs and the error plots have not converged to near horizontal. As a result, the effectiveness of using a large batch size is inconclusive during this exercise. This also shows that having a large batch size would make the learning process too slow.

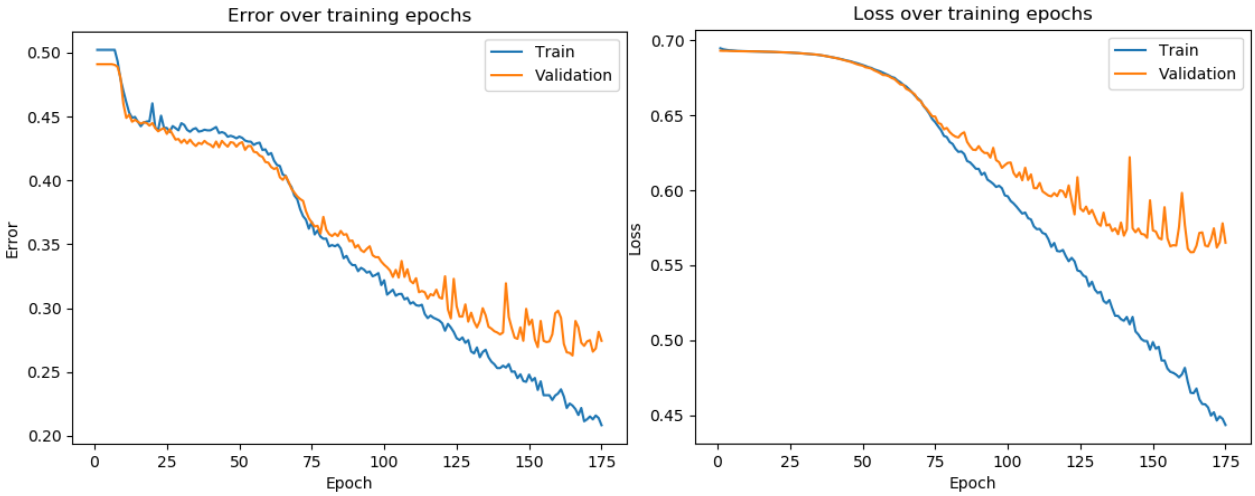
5.3. Find the Best Hyperparameter Setting

1. During initial trials, it was found that for most hyperparameter settings, the training error gets close to zero, but the validation error only approaches 30%. It was just a matter of the training speed.

In this case, I picked a moderately large batch size (128) to counter overfitting (but not too large such that it would slow down the training) and a small learning rate (0.001). Even though it was unproven (see above) that a small learning rate was better, hypothetically it would be the better choice as it would be unlikely for the model to overcompensate or make other bizarre choices.

The number of training epochs was chosen at 175 given that the learning rate is a lot slower and 50 would not nearly be enough. During experimental trials using the above hyperparameters (batch size = 128, learning rate = 0.001), it was noted that at roughly between 160 and 190 epochs, the validation error and validation loss reaches a minimum before rebounding to worse values. Therefore, an arbitrary value of 175 was chosen to best approximate the optimal performance point of the model.

2. Data plots:

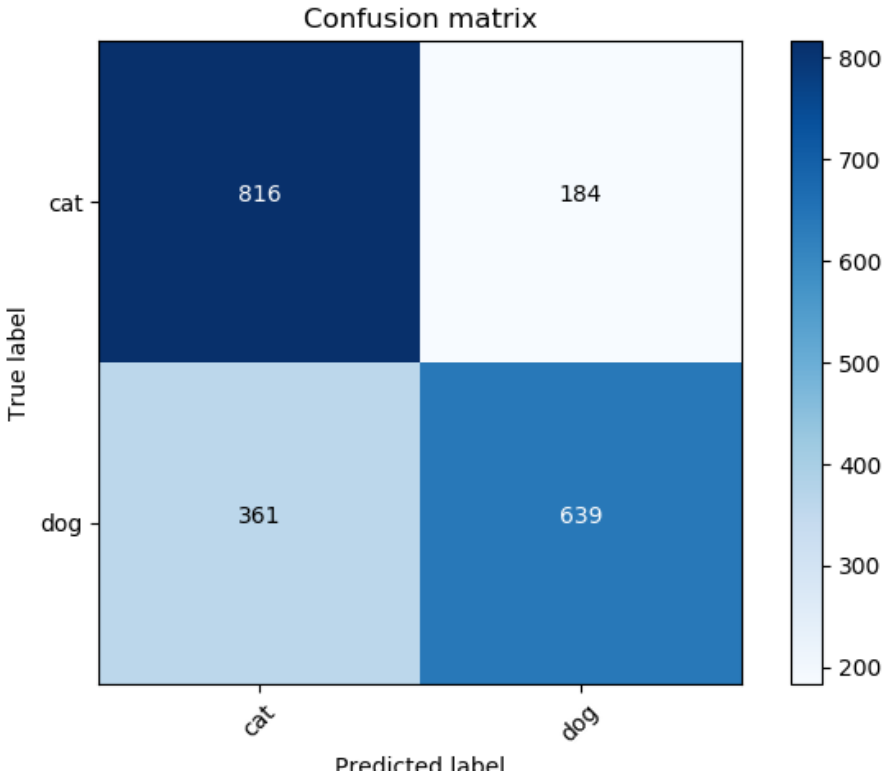


3.

Final training error: 20.838%
Final validation error: 27.450%
Final training loss: 0.44340
Final validation loss: 0.56505

5.4. Evaluate the Best Model on the Test Set

1.

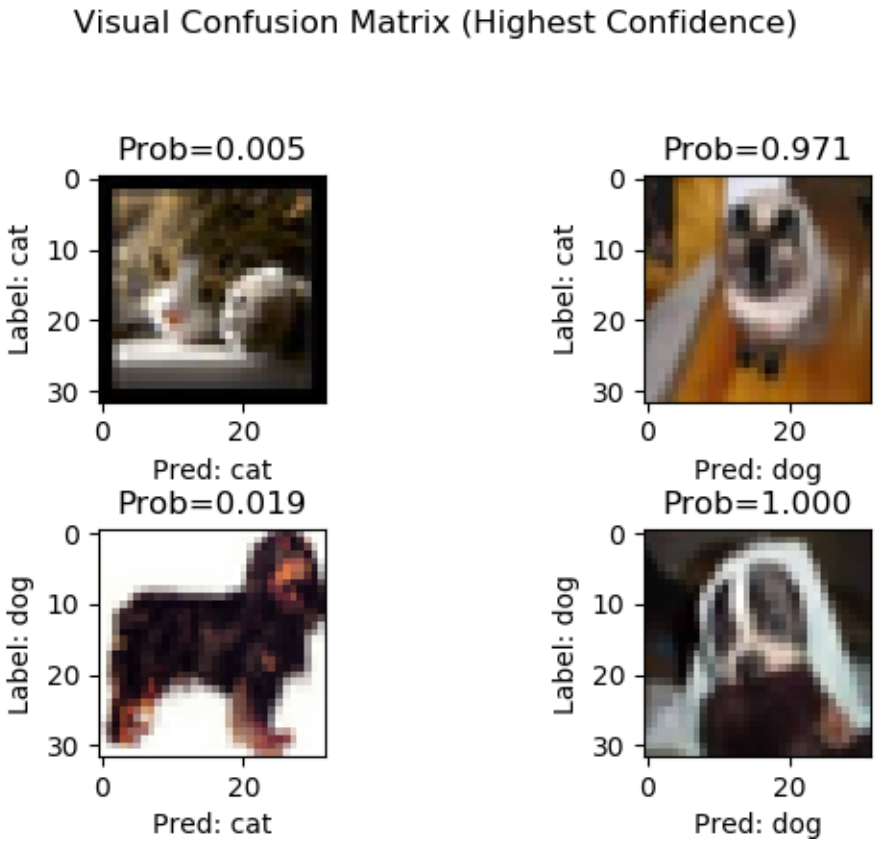


Confusion matrix, without normalization:
[816 184
 361 639]

The confusion matrix is somewhat unbalanced as it is 27.7% more likely for the model to recognize cats than dogs. Specifically, it is much more likely for the model to mistake dogs for cats than the other way around (almost twice as likely).

2. Total number of test items = 816 + 184 + 361 + 639 = 2000
Total number of images classified correctly = 816 + 639 = 1455
Overall test classification accuracy rate = 72.75%
Overall test classification error rate = 1 - 72.72% = **27.28%**

3.



Since all the images are quite blurry and are small in pixel size, there may have been challenges in the model to recognize certain features in cats and dogs.

The top left and bottom right images are obvious choices since it zooms into the faces of the cat and dog, making their facial patterns easily distinguishable (such as the cat's paws, the more detailed fur colour patterns, red nose; the black nose of a dog, and more plain fur patterns).

On the other hand, in the top right and bottom left images where the errors are the greatest, the faces of the animals only took up a small part of the pictures, making the facial patterns hardly distinguishable. Instead, it shows the body more. However, it can be noted that in both images, it is somewhat obvious to classify the animal by looking at the body features (specifically the longer legs of the dog below, and the pointy ears of the cat above).

Therefore, it can be hypothesized that the model may have put too much emphasis on facial features (or the closeup features) of cats and dogs, instead of looking at their body features. This may be bias resulted from training data that is more heavily weighted/numbered towards facial images or close-ups instead of side/front views of the body. As a result, it was difficult to improve the validation and testing error beyond 25%.