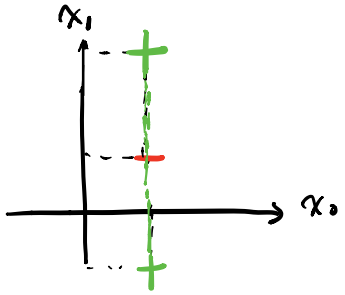
1.

(a)

x_0	x_1	t
1	-1	1
1	1	0
1	3	1

$\Rightarrow$



If the two positive points lie in the same halfspace, Then their line segment also lies in the halfspace.

However, the negative example is on the line segment. the intersection can't lie in both halfspaces.

Thus it's not linearly separable.

(b) $z = w_1\psi_1(x) + w_2\psi_2(x)$

$= w_1 x + w_2 x^2$

, plug in:

x	t
-1	1
1	0
3	1

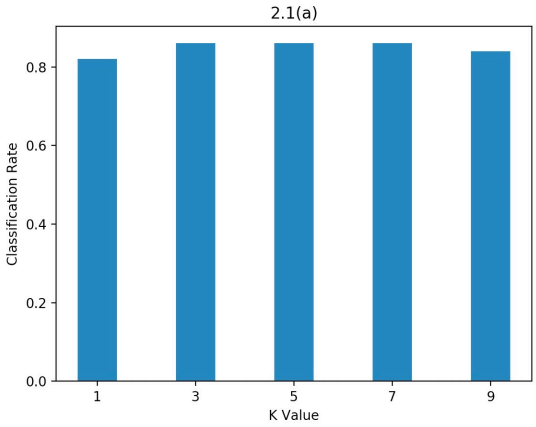
so the constraints are:

$$\begin{cases} z(-1) = -w_1 + w_2 \geq 0 \\ z(1) = w_1 + w_2 < 0 \\ z(3) = 3w_1 + 9w_2 \geq 0 \end{cases}$$

an example pair is $\begin{cases} w_1 = -2 \\ w_2 = 1 \end{cases}$

2.

2.1 (a)



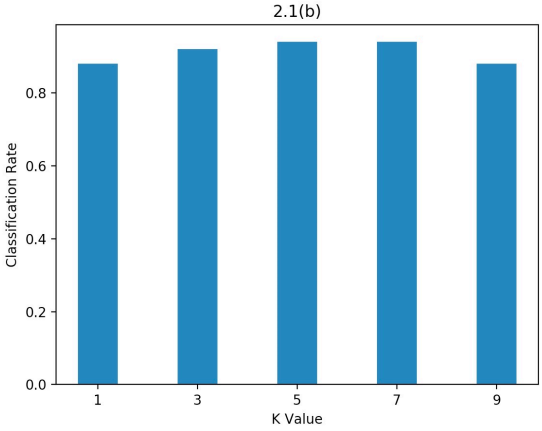
2.1(b)

K 1 3 5 7 9

Accuracy : [0.8200000000000001, 0.86, 0.86, 0.86, 0.84]

I would choose $k^* = 5$ and the classification rate is 0.86

k^*-2 and k^*+2 have the same performance on the validation set.



By the graph, the test perfomace is very similar as their corresponding validation performance.

2.2 (b)

minst_train:
Learning_rate = 0.01, num_iterations = 500

Report:

```
Training set: Cross entropy = 0.1989507519485625; Accuracy = 0.96875.
Validation set: Cross entropy = 0.30936835909693905; Accuracy = 0.9.
Test set: Cross entropy = 0.2785004434334882; Accuracy = 0.92.
```

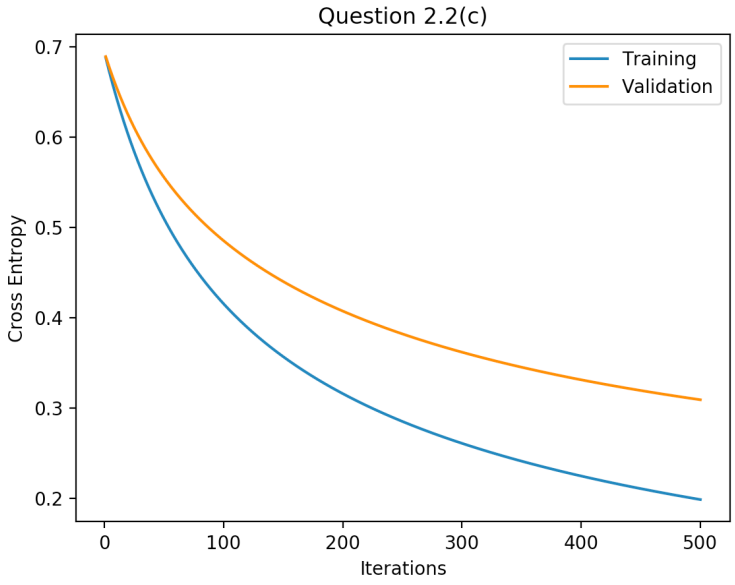
minst_train_small:
Learning_rate = 0.1, num_iterations = 1000
Report:

```
Training set: Cross entropy = 0.0023537411855112014; Accuracy = 1.0.
Validation set: Cross entropy = 0.8492591943033422; Accuracy = 0.72.
Test set: Cross entropy = 0.8202510049831307; Accuracy = 0.78.
```

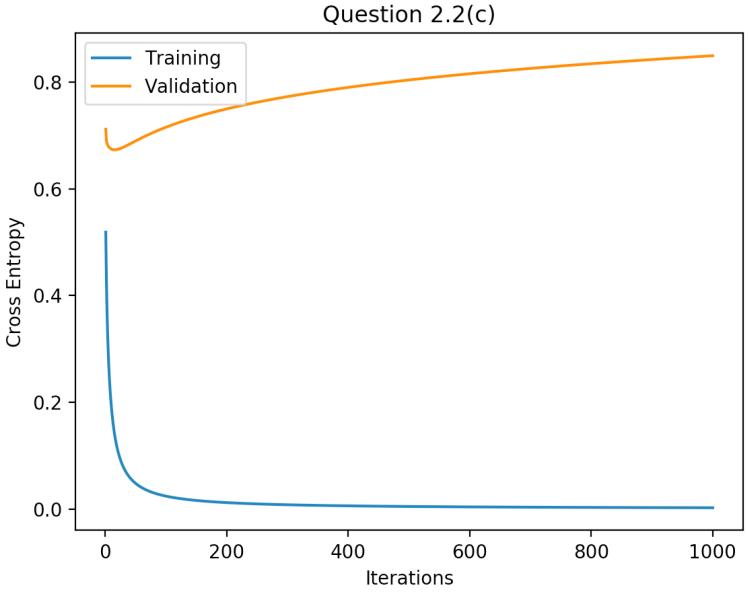
2.2(c)

minst_train

they don't change. If the change is big, change the learning rate and num-iterations until they become stable.



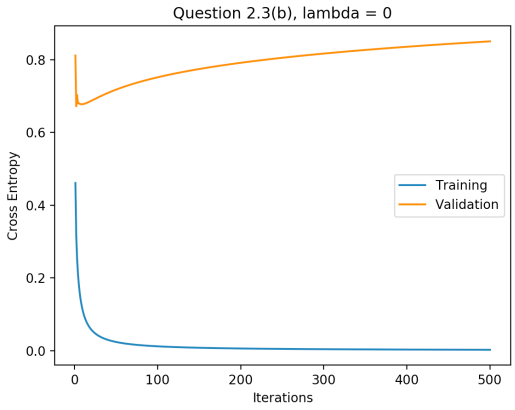
minst_train_small

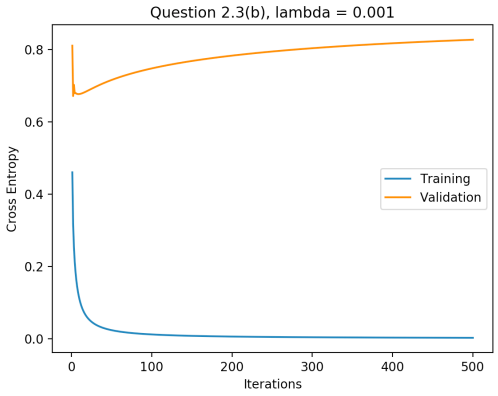


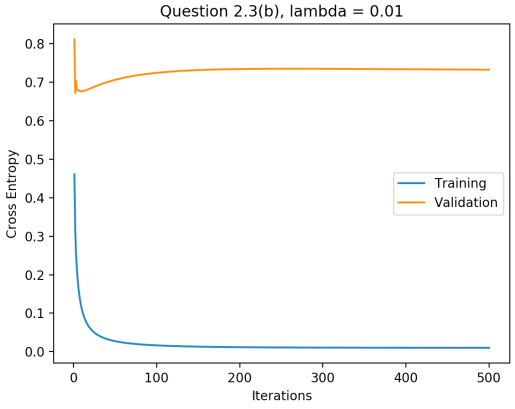
2.3(b)

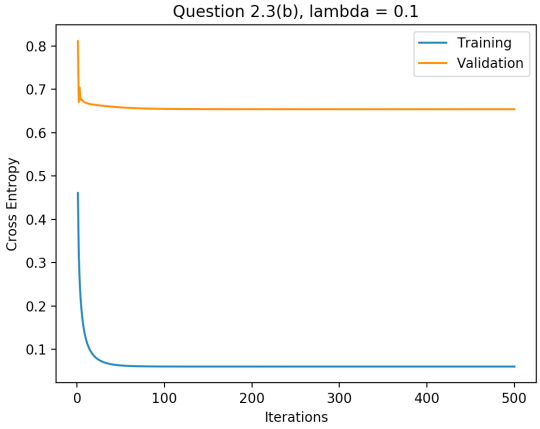
minst_train_small: learning_rate = 0.2 num_iterations = 500

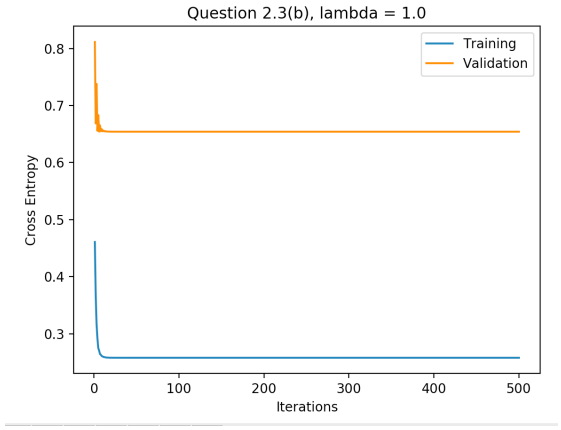
```
lambda = 0:
        Training Set: Cross Entropy = 0.0023476385882421555; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.8503265060955287; Accuracy = 0.72

lambda = 0.001:
        Training Set: Cross Entropy = 0.0029981743736484023; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.827741976717989; Accuracy = 0.72

lambda = 0.01:
        Training Set: Cross Entropy = 0.010154176907660703; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.7326792328919822; Accuracy = 0.72

lambda = 0.1:
        Training Set: Cross Entropy = 0.060312711369383744; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.6542352033575066; Accuracy = 0.6599

lambda = 1.0:
        Training Set: Cross Entropy = 0.25805743921464586; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.6541228615918029; Accuracy = 0.56
```





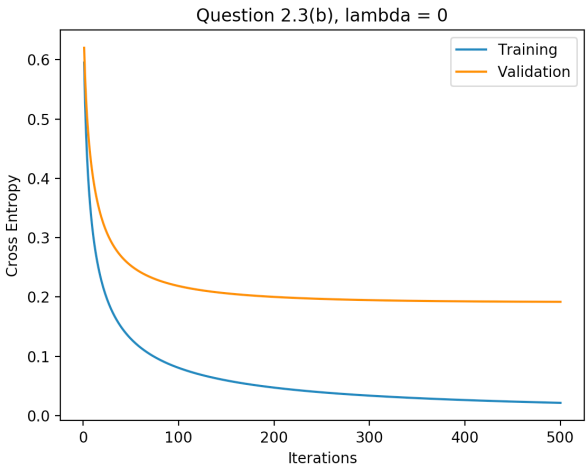


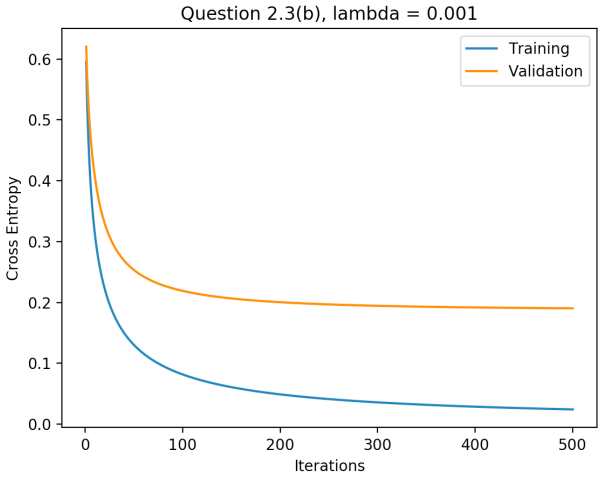


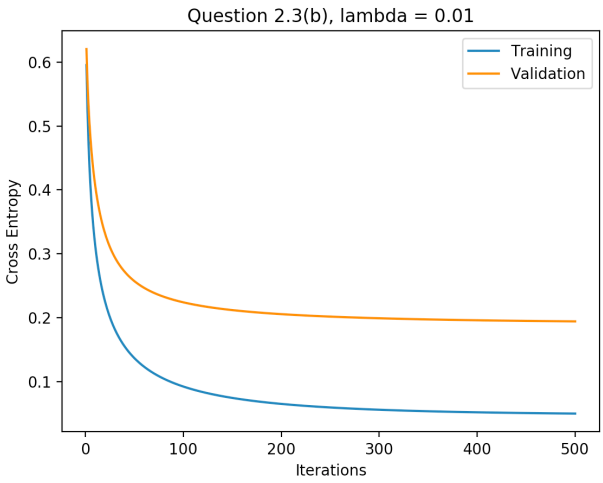


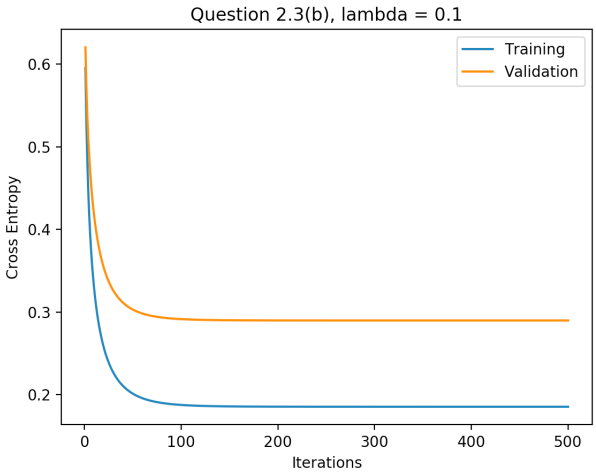
minst_train: learning_rate = 0.2 num_iterations = 500

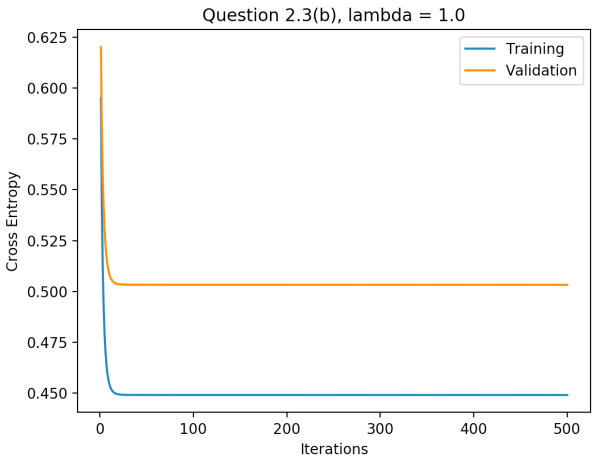
```
lambda = 0:
        Training Set: Cross Entropy = 0.02149365389873113; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.19186071414097075; Accuracy = 0.8799999999999999

lambda = 0.001:
        Training Set: Cross Entropy = 0.0242590031133181; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.19032434481318533; Accuracy = 0.8799999999999999

lambda = 0.01:
        Training Set: Cross Entropy = 0.05039118509561262; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.19434138438670204; Accuracy = 0.8799999999999999

lambda = 0.1:
        Training Set: Cross Entropy = 0.1852337669540374; Accuracy = 1.0
        Validation Set: Cross Entropy = 0.2888937127651922; Accuracy = 0.8999999999999999

lambda = 1.0:
        Training Set: Cross Entropy = 0.4485480819130602; Accuracy = 0.95
        Validation Set: Cross Entropy = 0.5017620166012411; Accuracy = 0.8399999999999999
```











2.3(c)

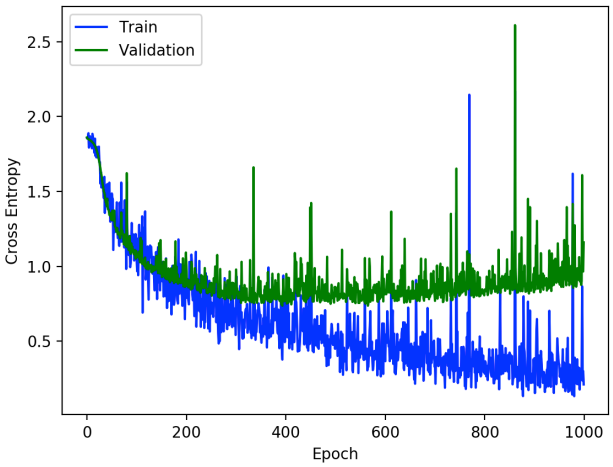
For minst_train, the final cross entropy increases as lambda increases. It also takes fewer iterations to converge when lambda becomes large.
For minst_train_small, the cross entropy of training set goes up when we increase lambda. However, the cross entropy of the validation set goes down.
The reason that CE of the training set goes up it is a price of minimizing the sum of CE+regularization. We can't pick some 'best' weights because that will give a large penalization.
We pick the best lambda to be 0.01. The result on the test set is:

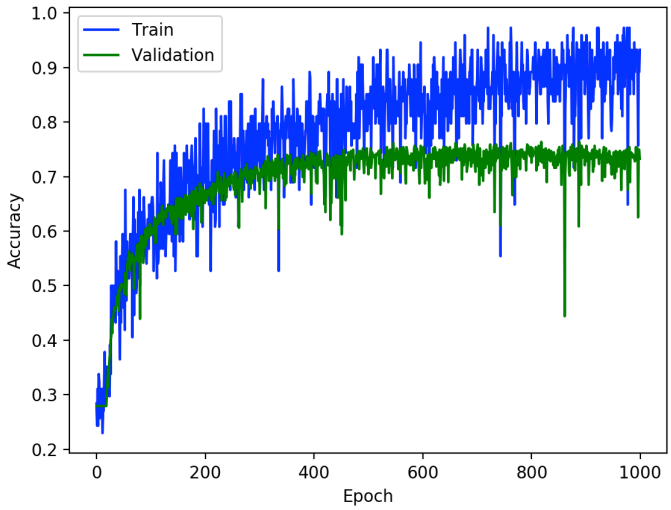
```
Test set: Cross entropy = 0.20308275573162557; Accuracy = 0.92.
```

```
Test set: Cross entropy = 0.6762196896634659; Accuracy = 0.78.
```

3(b)

```
CE: Train 0.44267 Validation 1.15963 Test 1.13064
Acc: Train 0.83640 Validation 0.73270 Test 0.71169
```

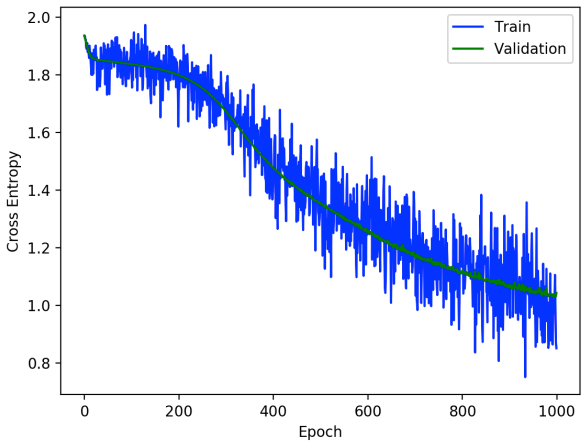


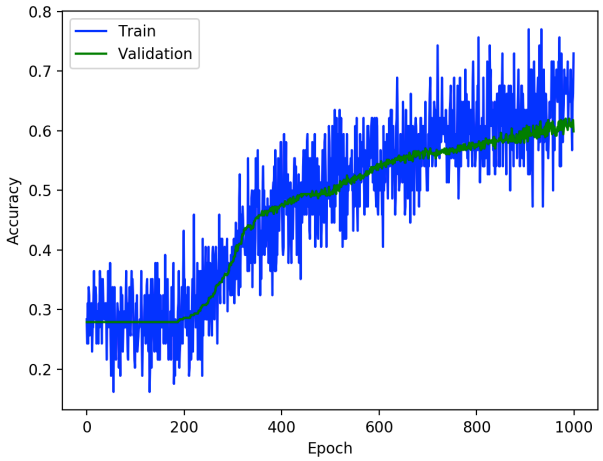


The error (Cross Entropy & Classification error) of the training set are both smaller than those of the validation set. i.e. the network performs better on the training set.

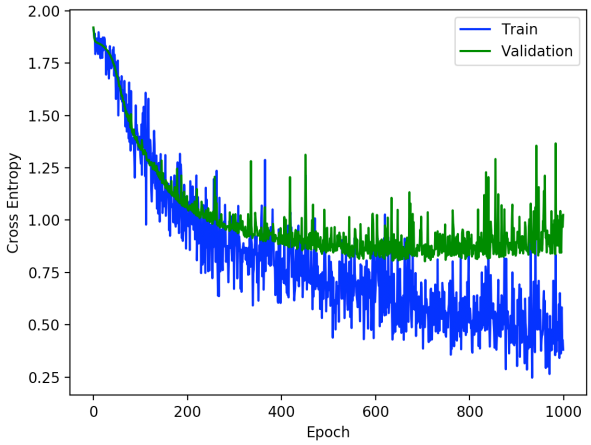
3(c). Test alpha, fix batch_size = 100

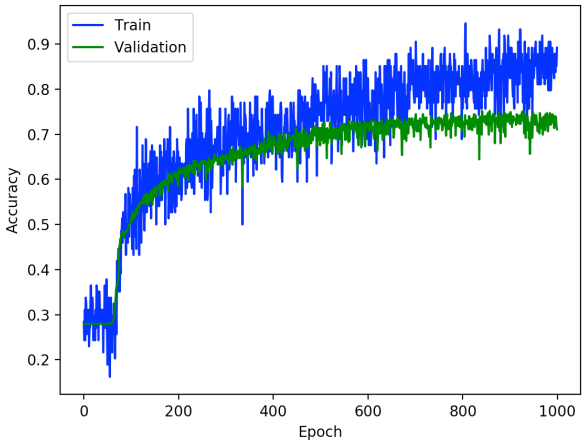
alpha = 0.001:



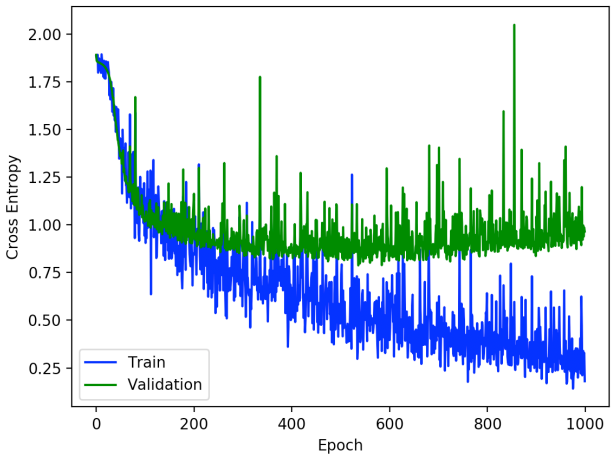


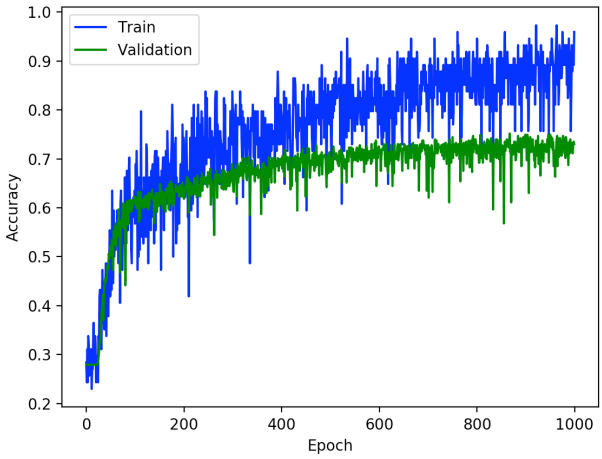
alpha = 0.005:



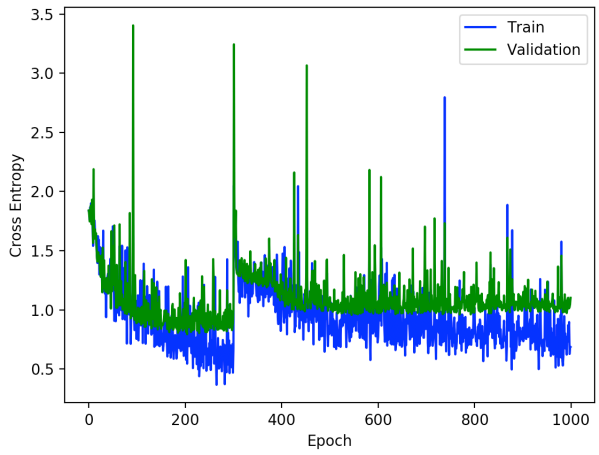


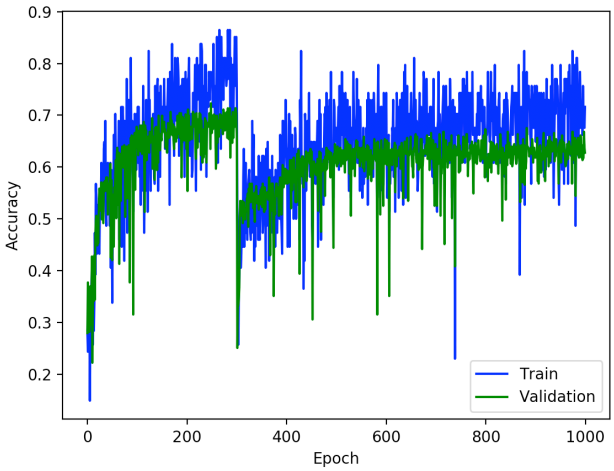
alpha = 0.01:



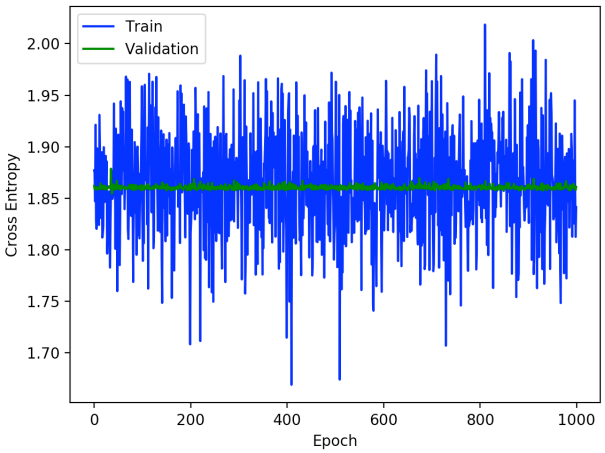


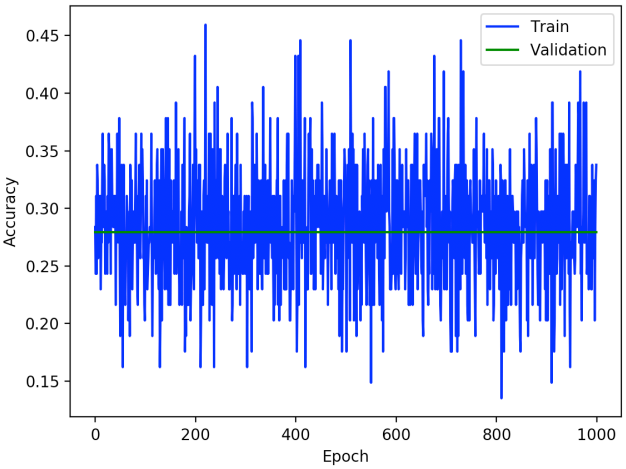
alpha = 0.1:



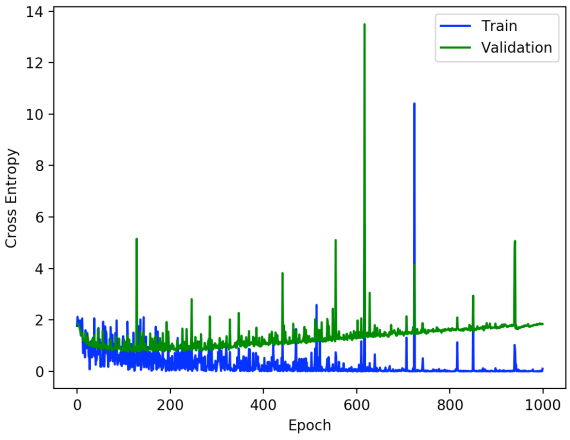


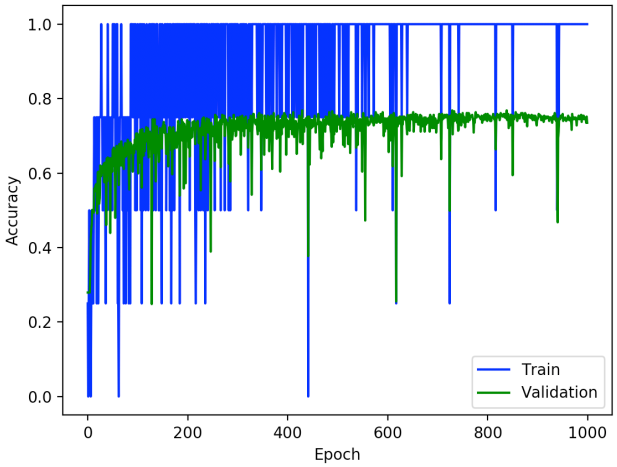
alpha = 1.0:



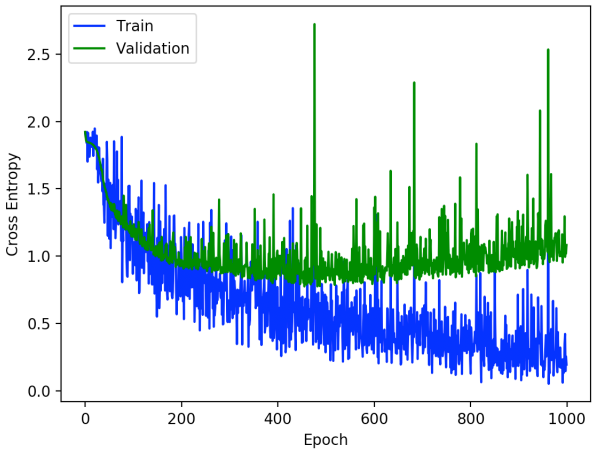


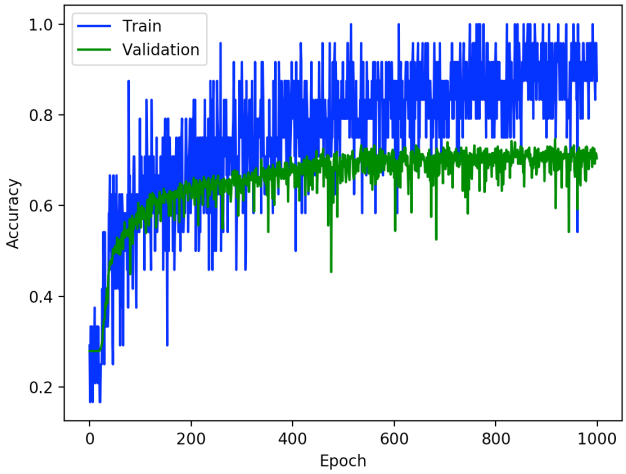
Test batch_size, fix alpha = 0.005
batch_size = 10:



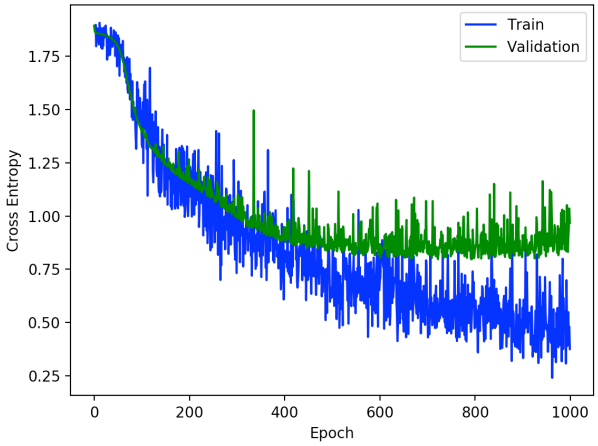


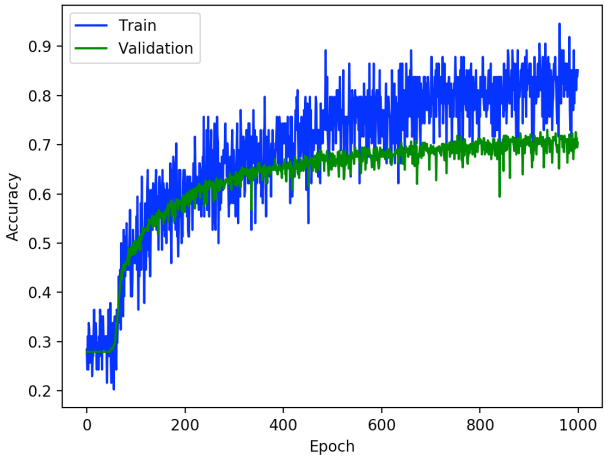
batch_size = 50:



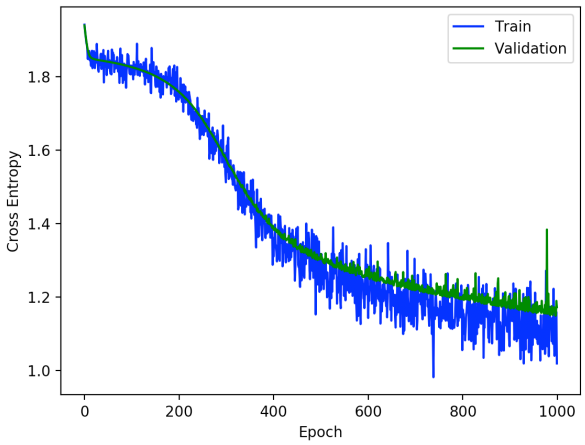


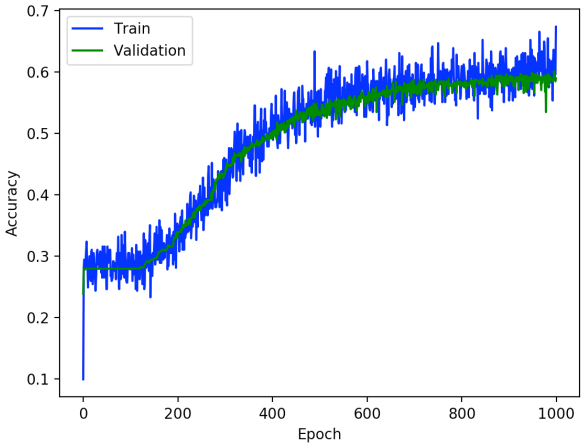
batch_size = 100:



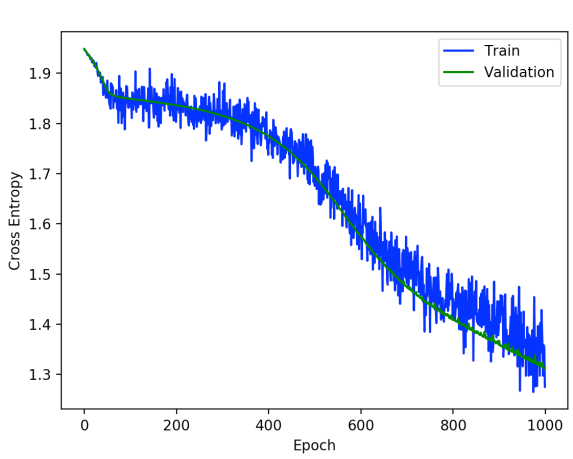


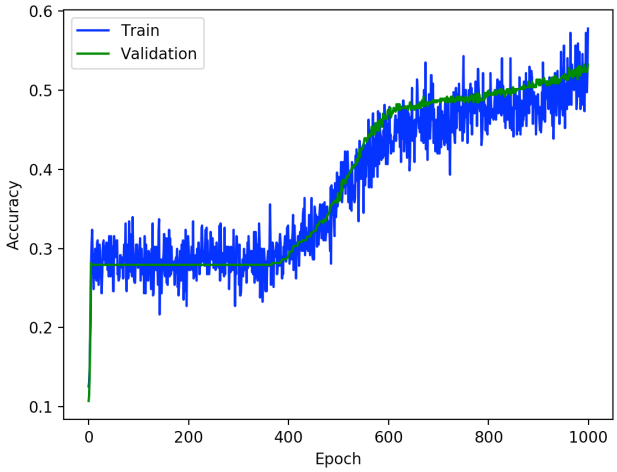
batch_size = 500:





batch_size = 1000:

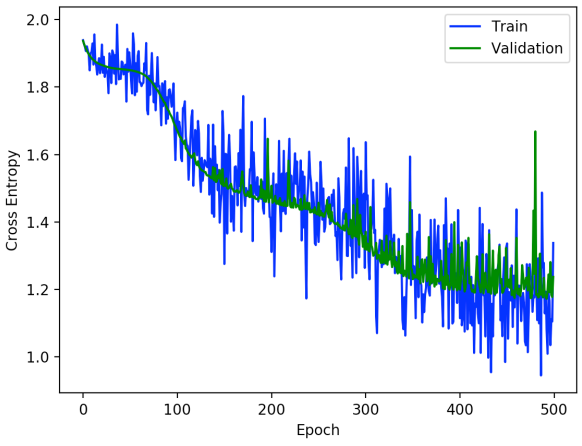


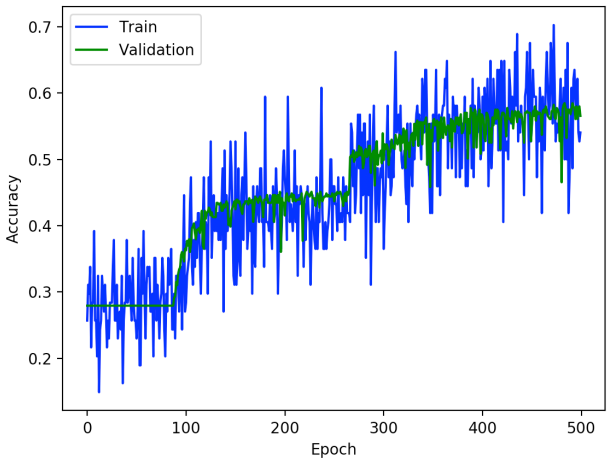


If we fix the batch_size, notice either the learning rate is too small (0.001) or too large (0.1, 1.0) will lead to the oscillation of the training error & accuracy. What we expect is both the error & accuracy can converge stably to some value, from this perspective, I would choose alpha = 0.005 to be the best learning rate.
Similarly, fix the learning rate. When the batch_size is too small (10, 50), the curve does NOT converge at all. Generally speaking, a larger batch_size leads to better convergence. Consider both stability and the accuracy, I would choose batch_size = 100 be the best among them.

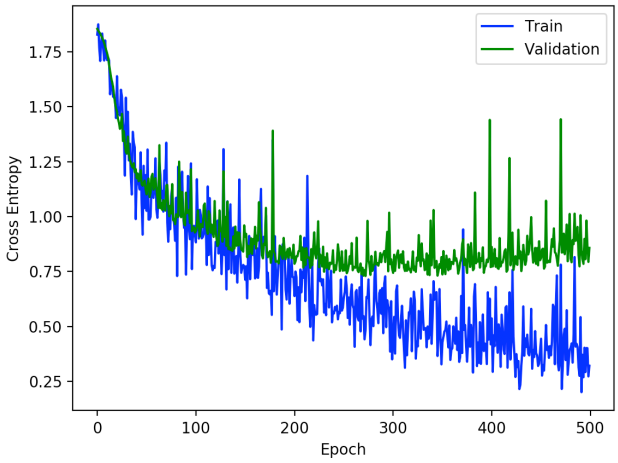
3(d).

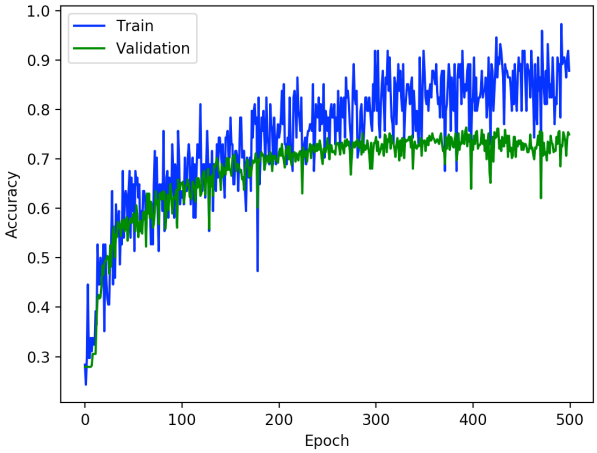
num_hiddens = [2, 4], alpha = 0.01:



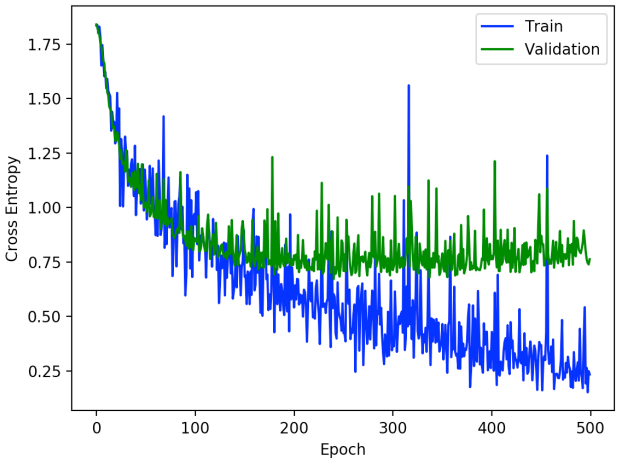


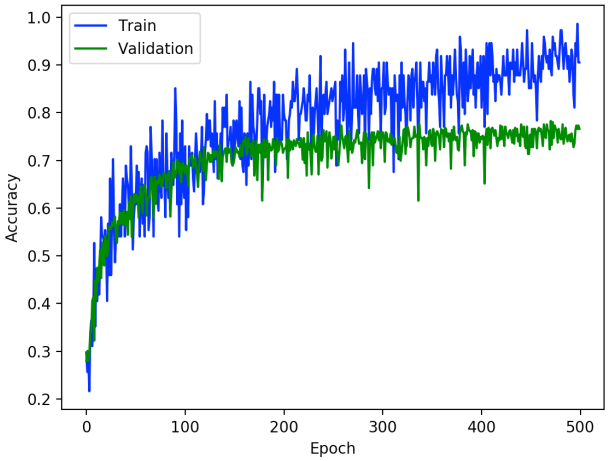
num_hiddens = [40, 60], alpha = 0.01:





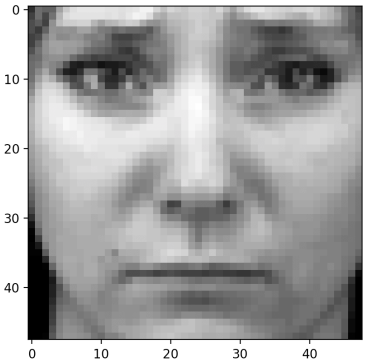
num_hiddens = [90, 100], alpha = 0.01:



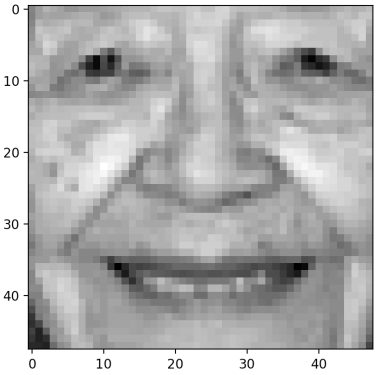


When the number of hidden units is too small (2,4), the error & accuracy won't converge and so that the training result of the network is bad.
As the number of hidden units increases, the error & accuracy converges better and better in our given range. So as the generalization of the network.

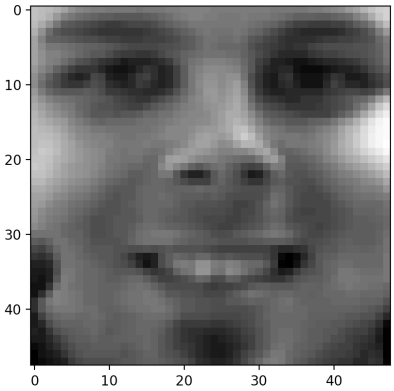
3(e). No, there is some samples that predicted to be wrong. Some of the images is hard to classify even by people.



```
The predicted is Neutral.
The target is Sad.
```



```
The predicted is Disgust.
The target is Happy.
```



```
The predicted is Happy.
The target is Happy.
```