CSC311 Assignment 1

Chen Liang

Sept 30, 2019

Question 1.1

$$
\begin{aligned}
P(t=1|x) &= \frac{P(x|t=1)P(t=1)}{P(x)} \\
&= \frac{P(x|t=1)P(t=1)}{P(x|t=1)P(t=1)+P(x|t=0)P(t=0)} \\
&= \frac{\alpha P(x|t=1)}{\alpha P(x|t=1)+(1-\alpha)P(x|t=0)} \\
&= \frac{1}{1+\frac{1-\alpha}{\alpha}\cdot\frac{P(x|t=0)}{P(x|t=1)}} \\
&= \frac{1}{1+exp(ln(\frac{1-\alpha}{\alpha}\cdot\frac{P(x|t=0)}{P(t=1)}))} \\
&= \frac{1}{1+exp(ln(\frac{1-\alpha}{\alpha}))+exp(\frac{P(x|t=0)}{P(x|t=1)})} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(ln(\frac{\prod_{i=1}^{n}P(x_i|t=0)}{\prod_{j=1}^{n}P(x_j|t=1)}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(ln((\frac{\prod_{i=1}^{n}P(x_i|t=0)}{\prod_{j=1}^{n}P(x_j|t=1)})))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(\frac{P(x_i|t=0)}{P(x_i|t=1)}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(\frac{P(x_i|t=0)}{P(x_i|t=1)}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})+exp(\sum_{i=1}^{n}ln(\frac{\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}}}{\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}})-ln(\frac{1}{\sqrt{2\pi\sigma_i^2}}e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}})} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(\frac{1}{\sqrt{2\pi\sigma_i^2}})+ln(e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}})-ln(\frac{1}{\sqrt{2\pi\sigma_i^2}})-ln(e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}})-ln(e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}))} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}ln(e^{-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}})-ln(e^{-\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}}))} \\
&= \frac{1}{1+1-\alpha\alpha)exp(\sum_{i=1}^{n}-\frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}+\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2}))} \\
&= \frac{1}{1+\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}\frac{2\mu_{i0}x_i-2\mu_{i1}x_1+\mu_{i1}^2-\mu_{i0}^2}{2\sigma^2})} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}\frac{2\mu_{i0}x_i-2\mu_{i1}x_1+\mu_{i1}^2-\mu_{i0}^2}{2\sigma^2})} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n}x_i\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}+\frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2})}
\end{aligned}
$$

In this way, we could have $P(t=1|x) = \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n} x_i \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2})}$

Since,

$$\begin{aligned}
&\frac{1}{1+(\frac{1-\alpha}{\alpha})exp(\sum_{i=1}^{n} x_i \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2})} \\
&= \frac{1}{1+(\frac{1-\alpha}{\alpha})exp(-\sum_{i=1}^{n} x_i \frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2} - \frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2})} \\
&= \frac{1}{1+exp(ln(\frac{1-\alpha}{\alpha}) - \sum_{i=1}^{n} x_i \frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2} - \frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2})}
\end{aligned}$$

, we have $w = \frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2}\cdot$ and $b = \frac{\mu_{i1}^2-\mu_{i0}^2}{2\mu_i^2}$.

Question 1.2

$$
\begin{aligned}
likelihood &= \prod_{i=1}^{n} p(t^{(i)}|x^{(i)}, w, b) \\
&= \prod_{i=0}^{n} p(t^{(i)} = 1|x^{(i)}, w, b)^{t^{(i)}} p(t^{(i)} = 0|x^{(i)}, w, b)^{(1-t^{(i)})} \\
&= \prod_{i=0}^{n} (\frac{1}{1 + exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{t^{(i)}} (1 - \frac{1}{1 + exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{(1-t^{(i)})}
\end{aligned}
$$

Now suppose $z = \sum_{i=1\ n} w_i x^{(i)} - b$, and $\sigma(z) = \frac{1}{1+e^{-z}}$. Plug in these two new parameters, we have,

$$
\begin{aligned}
L(w, b) &= -log(likelihood) \\
&= -log(\prod_{i=0}^{n} \sigma(z)^{t_i} (1 - \sigma(z))^{1-t_i}) \\
&= -\sum_{i=1}^{n} (t^{(i)} ln(\sigma(z)) + (1 - t^{(i)}) ln(1 - \sigma(z)))
\end{aligned}
$$

In this way, we could calculate,

$$
\begin{aligned}
\frac{\partial L(w, b)}{\partial w} &= \frac{\partial L(w, b)}{\partial z} \frac{\partial z}{\partial w} \\
&= -(\sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{1}{\sigma(z)} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{1 - \sigma(z)} \frac{e^{-x_j}}{(1 + e^{-x_j})^2})) x_j^{(i)} \\
&= -(\sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{1}{\frac{1}{1+e^{-z}}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{1 - \frac{1}{1+e^{-z}}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2})) x_j^{(i)} \\
&= -(\sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} ((1 + e^{-x_j}) \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1 + e^{-x_j}}{e^{-x_j}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2})) x_j^{(i)} \\
&= -(\sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{e^{-x_j}}{(1 + e^{-x_j})}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{(1 + e^{-x_j})})) x_j^{(i)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L(w, b)}{\partial b} &= \frac{\partial L(w, b)}{\partial z} \frac{\partial z}{\partial b} \\
&= \sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{1}{\sigma(z)} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{1 - \sigma(z)} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) \\
&= \sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{1}{\frac{1}{1+e^{-z}}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{1 - \frac{1}{1+e^{-z}}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) \\
&= \sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} ((1 + e^{-x_j}) \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1 + e^{-x_j}}{e^{-x_j}} \frac{e^{-x_j}}{(1 + e^{-x_j})^2}) \\
&= \sum_{i=1}^{n} t^{(i)} \sum_{j=1}^{D} (\frac{e^{-x_j}}{(1 + e^{-x_j})}) - (1 - t^{(i)}) \sum_{j=1}^{D} (\frac{1}{(1 + e^{-x_j})})
\end{aligned}
$$

Question 1.3

First we express $p(w_i)$ as $p(w_i) = \frac{1}{\sqrt{2\pi\frac{1}{\lambda}}} e^{-\frac{w_i^2}{\frac{2}{\lambda}}} = \sqrt{\frac{\lambda}{2\pi}} e^{-\frac{w_i^2\lambda}{2}}$

Because $p(w,b|D) = \frac{p(D|w,b)p(w,b)}{p(D)}$, we could rewrite it as $p(w,b|D)p(D) = p(D|w,b)p(w,b)$, and since $p(D)$ is constant, $p(w,b|D) \propto p(D|w,b)p(w)p(b)$, and we could show that $p(w,b|t^{(1)},t^{(2)},...t^{(n)}) \propto p(w)p(b)p(t^{(1)},t^{(2)},...t^{(n)}|w,b)$.
Therefore,

$$
\begin{aligned}
p(w,b|t^{(1)},t^{(2)},...t^{(n)}) &= \frac{1}{c}p(w)p(b)p(t^{(1)},t^{(2)},...t^{(n)}|w,b) \\
&= \frac{1}{c}\prod_{i=1}^{D}(w_i \cdot (\frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{t^{(i)}}(1 - \frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{(1-t^{(i)})}) \\
&= \frac{1}{c}\prod_{i=1}^{D}(\sqrt{\frac{\lambda}{2\pi}} e^{-\frac{w_i^2\lambda}{2}} \cdot (\frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{t^{(i)}}(1 - \frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{(1-t^{(i)})}) \\
&= \frac{1}{c}(\sqrt{\frac{\lambda}{2\pi}})^D \prod_{i=1}^{D}(e^{-\frac{w_i^2\lambda}{2}} \cdot (\frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{t^{(i)}}(1 - \frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{(1-t^{(i)})} \\
&= \frac{1}{c'}\prod_{i=1}^{D}(e^{-\frac{w_i^2\lambda}{2}} \cdot (\frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{t^{(i)}}(1 - \frac{1}{1+exp(-\sum_{i=1\ n} w_i x^{(i)} - b)})^{(1-t^{(i)})})
\end{aligned}
$$

In this way, we could calculate $L_{post}(w,b)$ as,

$$
\begin{aligned}
L_{post}(w,b) &= -log(\frac{1}{c'}) + \sum_{i=1}^{D}\frac{w_i^2\lambda}{2} + L(w,b) \\
&= \sum_{i=1}^{D}\frac{w_i^2\lambda}{2} + L(w,b) + C
\end{aligned}
$$

Now calculate derivatives on $L_{post}(w,b)$:

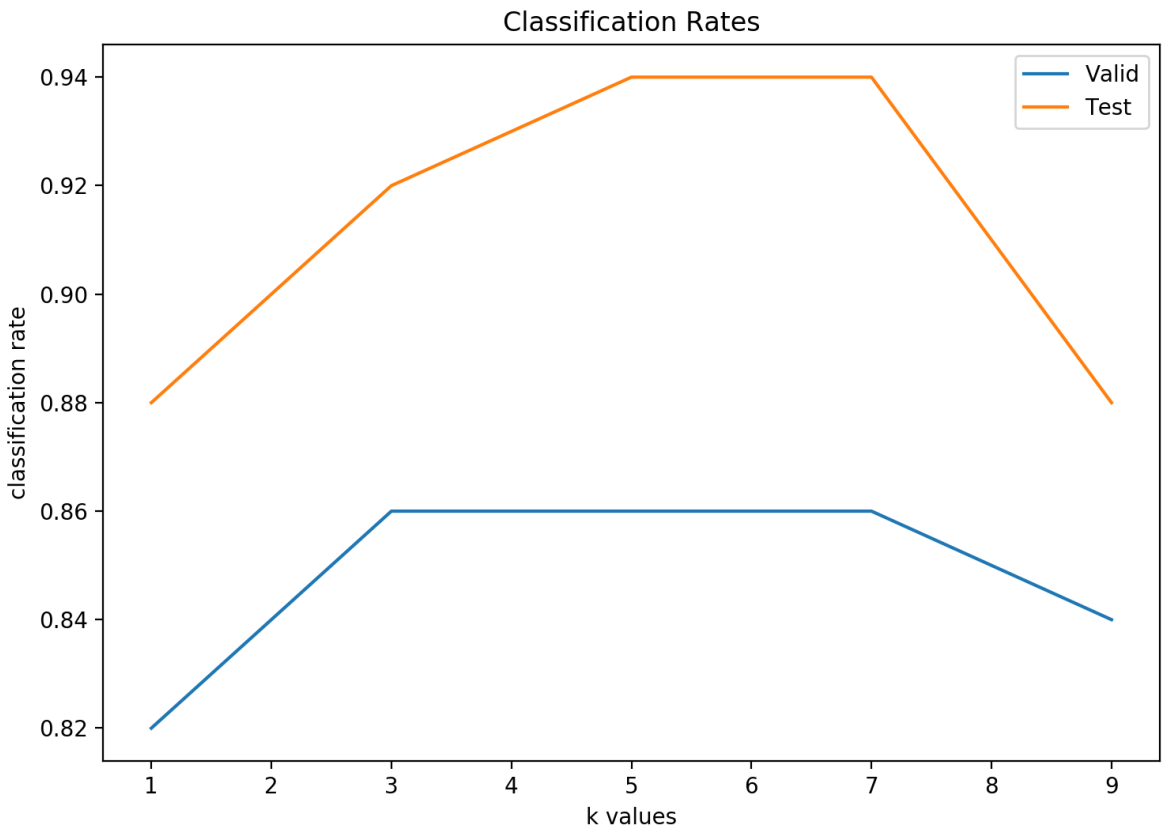
$$
\begin{aligned}
\frac{\partial L_{post}(w,b)}{\partial w} &= \lambda\sum_{i=1}^{D} w_i + \frac{\partial L(w,b)}{\partial w} \\
&= \lambda\sum_{i=1}^{D} w_i - (\sum_{i=1}^{n} t^{(i)}\sum_{j=1}^{D}(\frac{e^{-x_j}}{(1+e^{-x_j})}) - (1-t^{(i)})\sum_{j=1}^{D}(\frac{1}{(1+e^{-x_j})}))x_j^{(i)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L_{post}(w,b)}{\partial b} &= \frac{\partial L(w,b)}{\partial b} \\
&= \sum_{i=1}^{n} t^{(i)}\sum_{j=1}^{D}(\frac{e^{-x_j}}{(1+e^{-x_j})}) - (1-t^{(i)})\sum_{j=1}^{D}(\frac{1}{(1+e^{-x_j})})
\end{aligned}
$$

Question 2.1

	1	3	5	7	9
valid	0.88	0.92	0.94	0.94	0.88
test	0.82	0.86	0.86	0.86	0.84

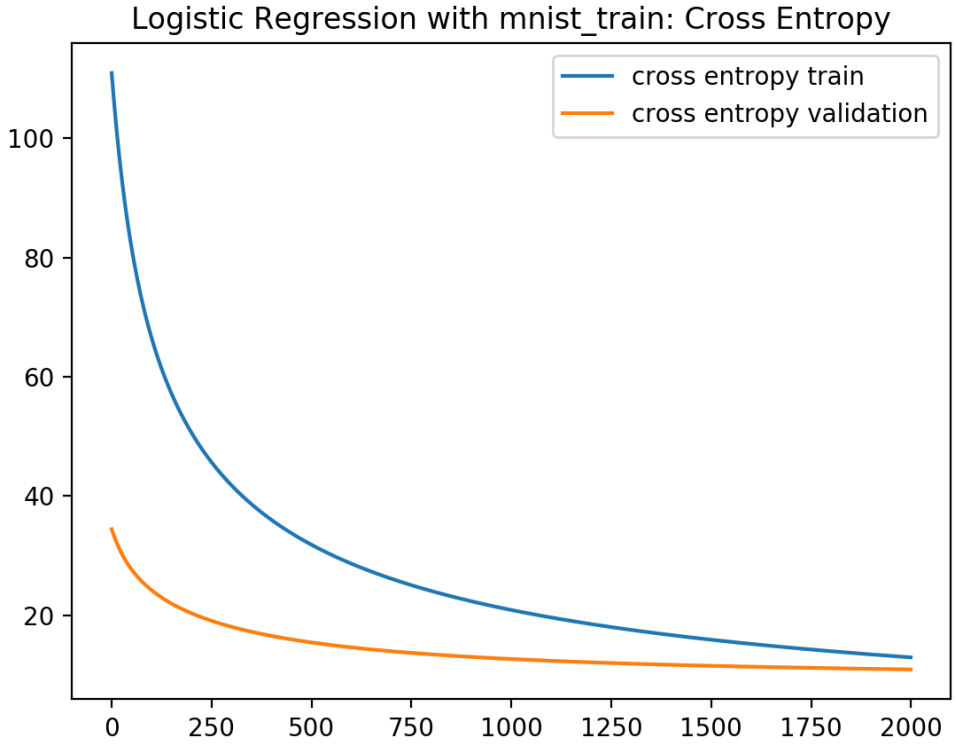
From the table above we could know that best values for k would be 3, 5, 7 for both validation set, and the test set. From my observation, $k^\star = 5$ would be the best choice, the reason is $k^\star = 5$ has a better classification rate than $k^\star = 3$, and on this line plotted below, $k^\star = 5$ is more stable than $k^\star = 7$. So $k^\star = 5$ would be the best option.


Classification Rates
Valid
Test
0.94
0.92
0.90
0.88
0.86
0.84
0.82
classification rate
1
2
3
4
5
6
7
8
9
k values

Question 2.2

In this question, there are only two changeable parameters: number of iterations and learning rate. Below I listed two plots I got by constantly changing these two factors in hyperparameters:

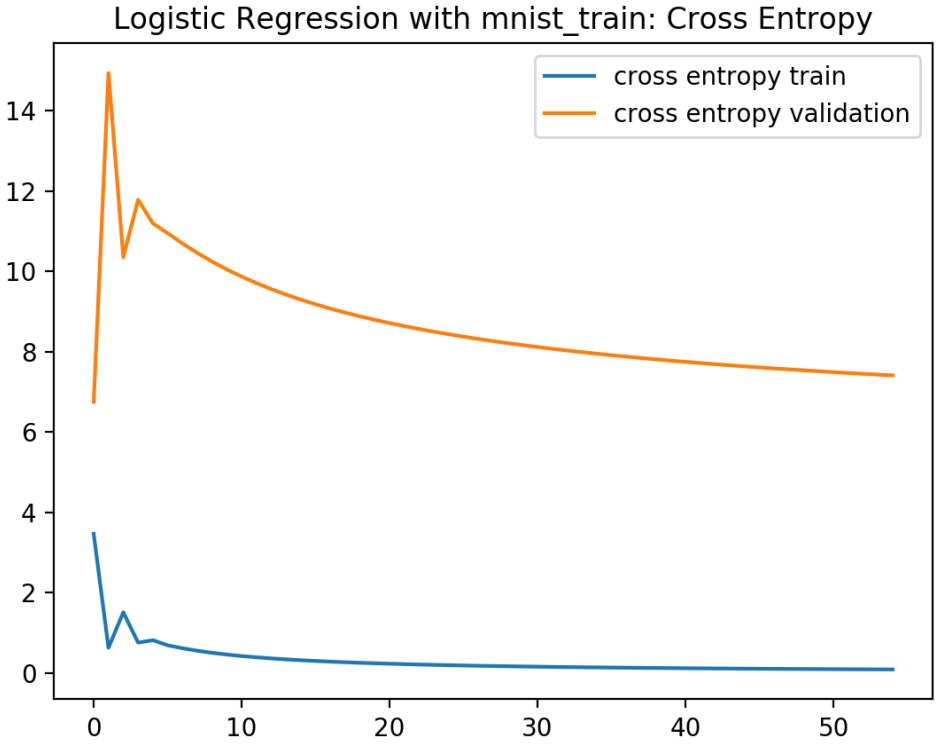
For *minist_train*, the hyperparameters I set are: **learning_rate = 0.01, num_iterations = 2000**
The resulting graph and statistcs look like below:
The model I choose is a good model.



Reasons: 1. The cross entropy numbers for both sets are pretty low.
2. The correct classification rate for train set reaches 1, and for validation set reaches 0.88, and both numbers are high enough to support the claim that this model could make pretty precise predictions.
3. Even though the train set reaches 1, the model does not overfit. The reason is that from the curve of "cross entropy train" we spot that before 1750, the cross entropy is still relatively high, so we need at least 1750+ iterations.
4. Most importantly, both curves are converging. And the trends on both curves are pretty significant.

	train set	validation set
ce	6.417257348287248	4.858928627627072
fac rate	1.0	0.88

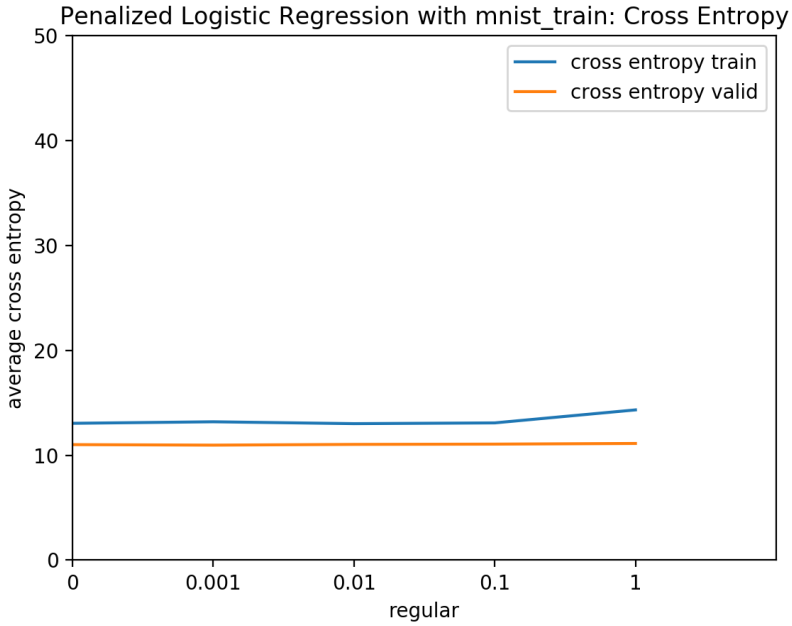
For the small training set, hyperparameters I set are: **learning_rate = 0.2, num_iterations = 55**
The reasons are quite similar to the previous analysis:
1. We could find both ce curves are conversing.
2. It might over fit a little bit, but the prediction accuracy for both sets is ensured to be over 2/3.
3. The cross entropy values are both relatively small, smaller than 10.
In this way, I state that both models I choose are good ones.



	train set	validation set
ce	0.08853605487520298	7.41131698264141
fac rate	1.0	0.68

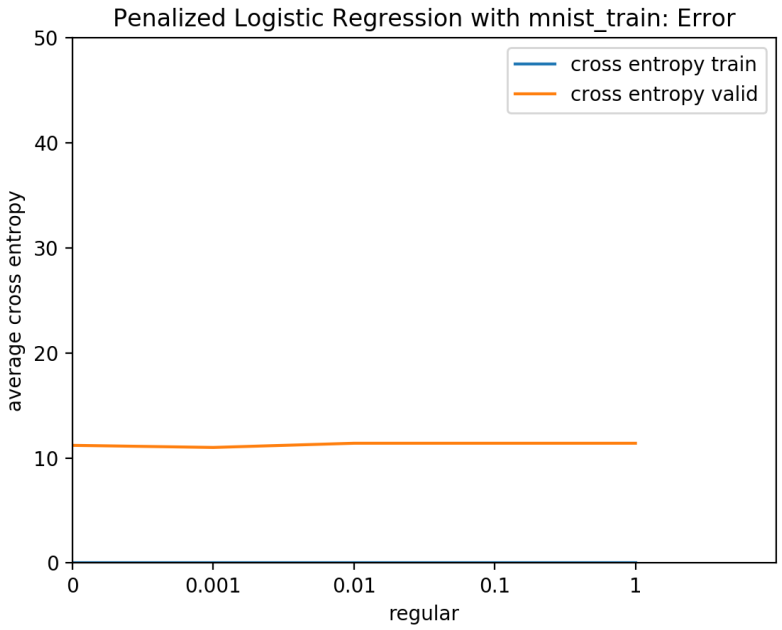
Question 2.3

To compare results of with and without penalty, we need to keep the learning rate and iteration steps same as question 2.2 when running on this part. That means for the *minist_train* set, we still keep the **learning_rate = 0.01, num_iterations = 2000**.



	0	0.001	0.01	0.1	1
train set	13.043511089962323	13.187438642174495	13.009961956081682	13.082652763145216	14.321052151817039
validation set	11.010750509340447	10.962450737131999	11.031738081506628	11.056927069148742	11.120848539528463

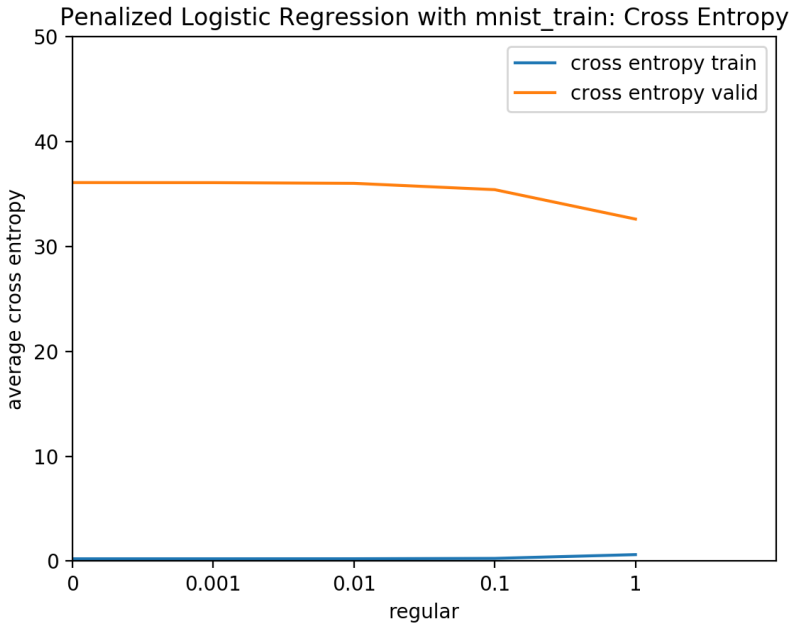
We could observe from graph that when $\lambda = 0, 0.001, 0.01, 0.1$, the cross entropy has no significant change. And when $\lambda = 1$, the cross entropy for training set increases a little bit.



	0	0.001	0.01	0.1	1
train set	0.0	0.0	0.0	0.0	0.0
validation set	11.199999999999989	10.999999999999972	11.599999999999994	11.400000000000006	11.199999999999989

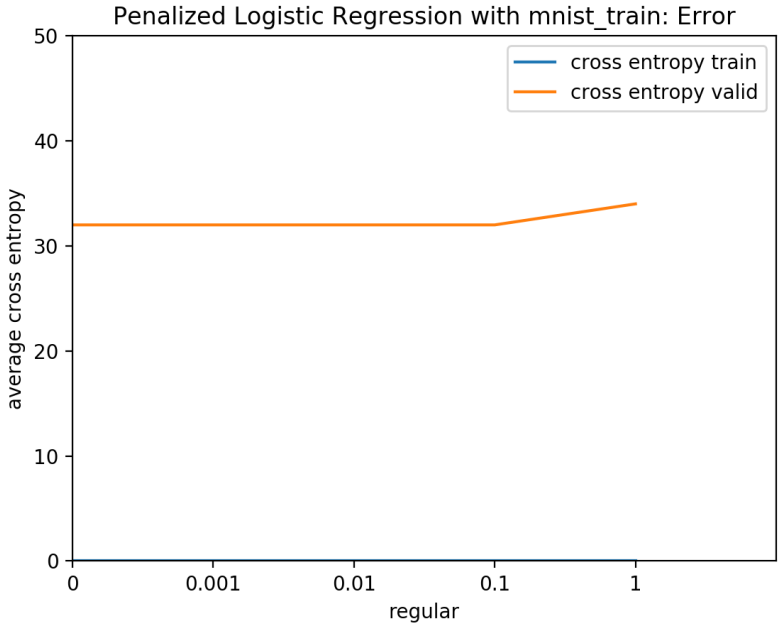
From the above graphs we observe that there are sigificant change no matter what value λ is. So for this model, $\lambda = 0.001$ might be the best option, since when $\lambda = 0.001$, the corss entropy for validation is the smallest, the cross entropy for the training set is the third smallest, and the error rate for validation set is also the smallest.

for the *minist_train_small* set, we still keep the **learning_rate = 0.2, num_iterations = 55**.



	0	0.001	0.01	0.1	1
train set	0.21984053451346824	0.2201979258672318	0.223423872746195 52	0.2565450781755554 4	0.6163671227702391
validation set	36.093418757057364	36.08604043746397	36.020226578037516	35.41729836489627	32.61265003970112

Observe from above graphs we will find that the with λ increasing, with cross entropy for the validation set decreases, and conversely the cross entropy for the training set increases a little bit.



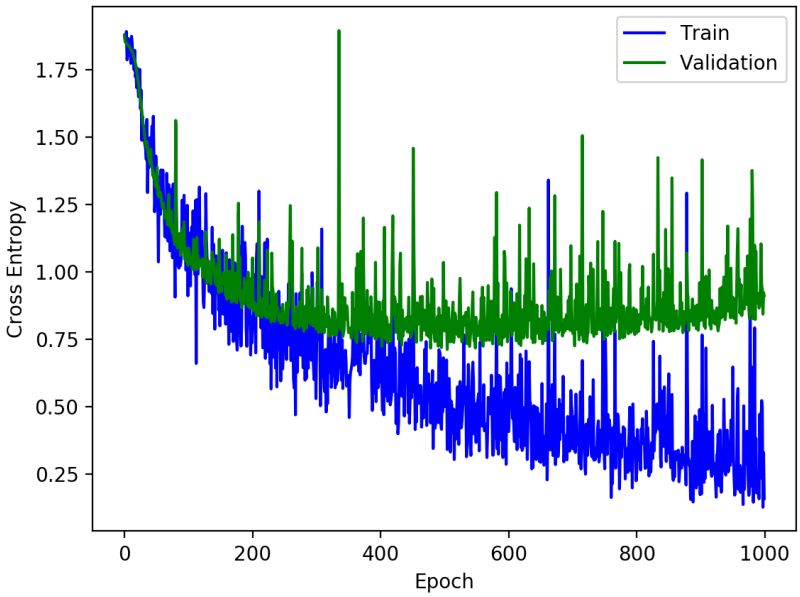
	0	0.001	0.01	0.1	1
train set	0.0	0.0	0.0	0.0	0.0
validation set	32.0	32.0	32.0	32.0	34.0

Observe from the above graphs we will find that when λ to 1,the average error also increases slightly. And there's no error for the training set. In this way, we state for this model, $\lambda = 0.01$ might be the best option since the cross entropy for validation set is the smallest, and for the training set is the third smallest (really close to the top two results), and at $\lambda = 0.01$ the error rate for the validation set is the smallest. In this way, we prefer $\lambda = 0.01$.

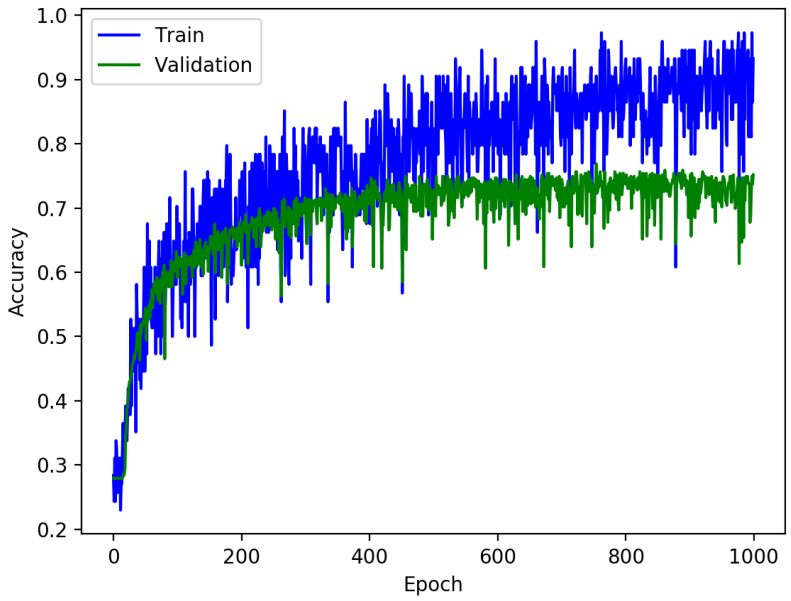
Comparing the results without regularization, we could notice that in our cases, there's no significant improvements in both cross entropy and error rates. The reason might be the models I chose for the situation without regularization might not fit the case with regularization. Hypothetically, adding regularization to model could make the model perform better due to the reason that regularization could reduce the weights parameters and consequently present the over fitting of the model. Since this might not occur on all cases, if I choose another model for question 2.2 the results might be different for this question.

Question 3.1

Observation 1: Oberseve from graphs below we could find both cross entropy curves experiences a sharp decrease between epoch 0 to 300, indicating that the models are updating fast at the first 300 epoches. After 300 epoches, the decreasing rate slows down while epoch number increasing. The reson is that the model are learning faster at the first few epoches and it could gain more information than later epoches.
Observation 2: From the graph below we could observe that the cross entropy of the validation set starts to increase after point 600. The reason might be the model might learn noise terms from the training set, and might make the prediction for validation set inaccurate than before.
Observation 3: From the entropy graph we could notice that training set's cross entropy is significantly lower than validation set's cross entropy. The reason might be the size of the training set is 3314 and the size of the validation set is 419. Because the size of the training set is way larger than the size of the validation set, so the impact of an outlier on the entropy of the validation set would be more significant compare to the training set. In this way, due to the size difference, the cross entropy of the training set is smaller than the validation set.



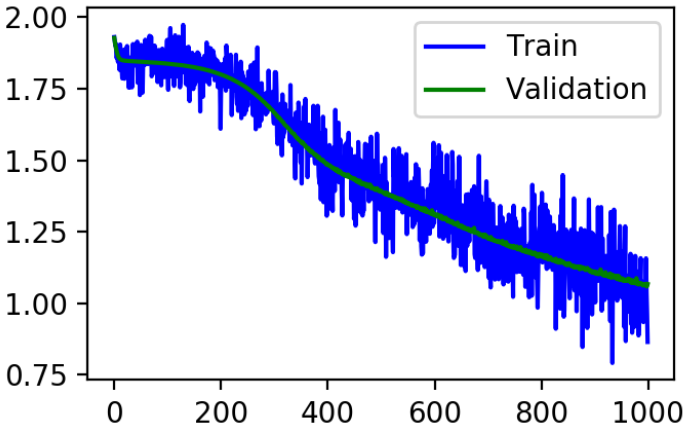
Entropy Graph.png

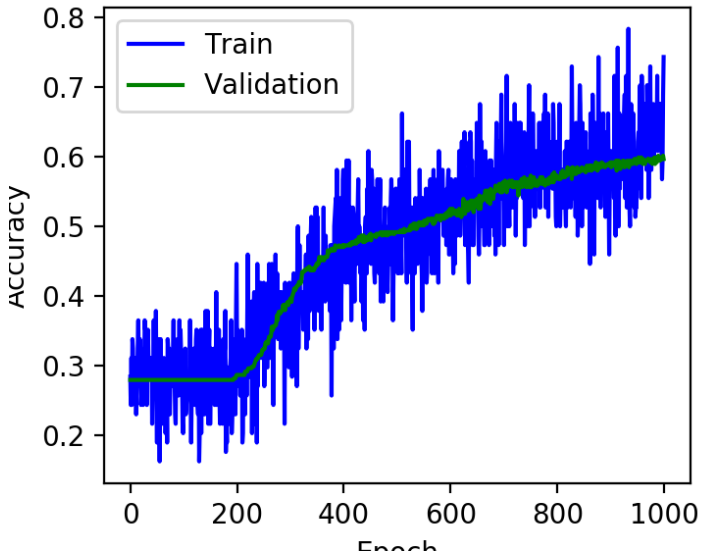


Graph.png

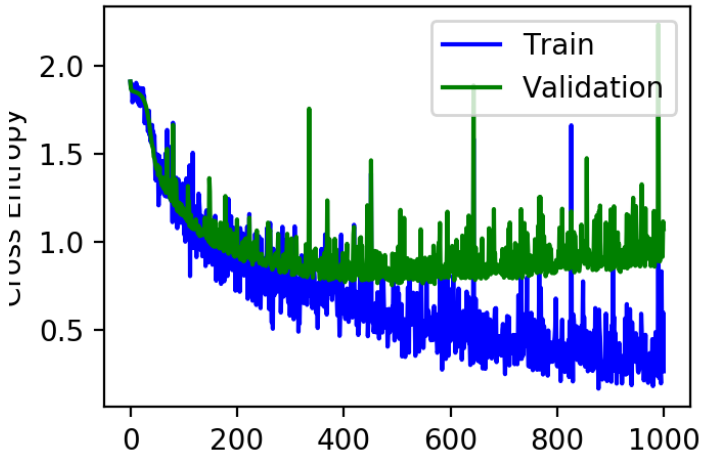
Question 3.2

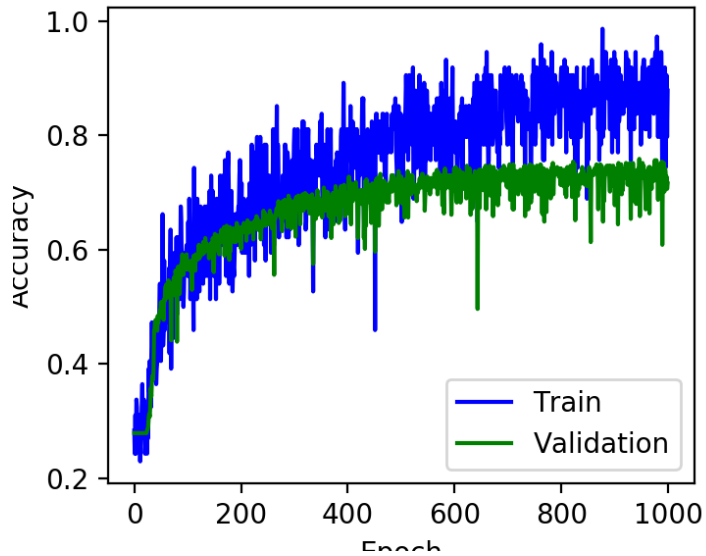
Five different values for learning rate:
I choose $\epsilon = 0.001, 0.01, 0.1, 0.5, 1$.
Below are graphs for $\epsilon = 0.001$.



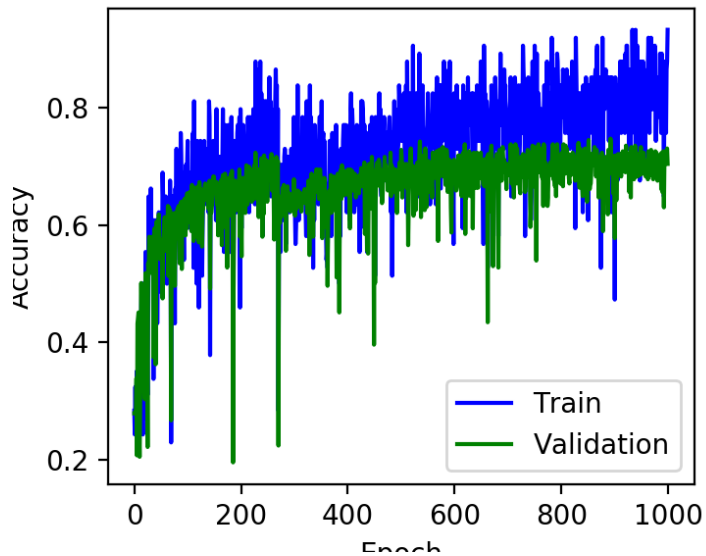


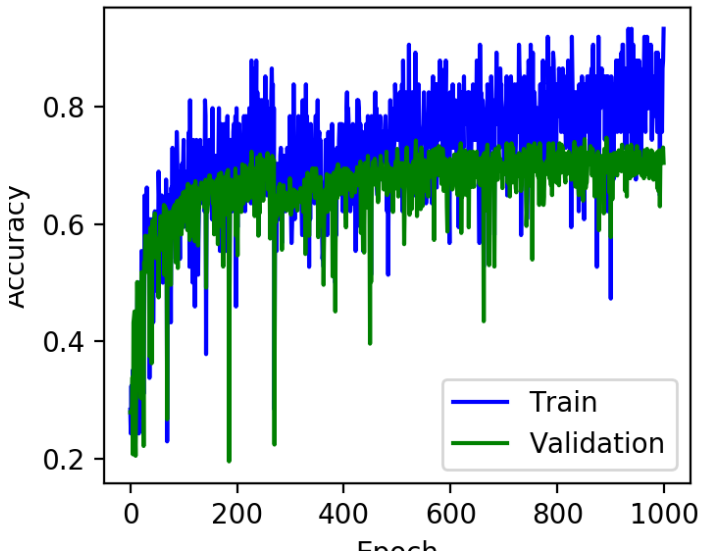
Below are graphs for $\epsilon = 0.01$.



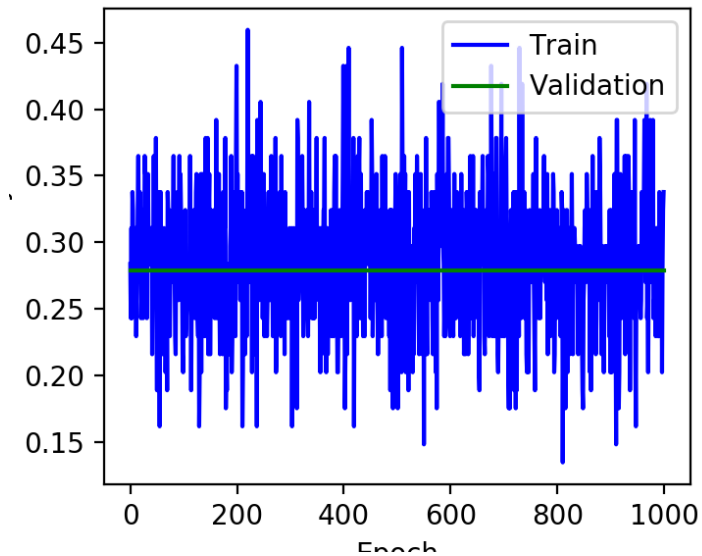


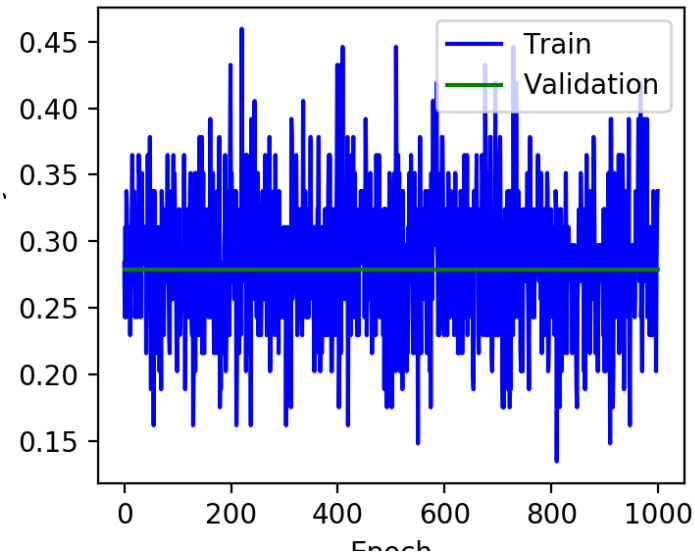
Below are graphs for $\epsilon = 0.1$.



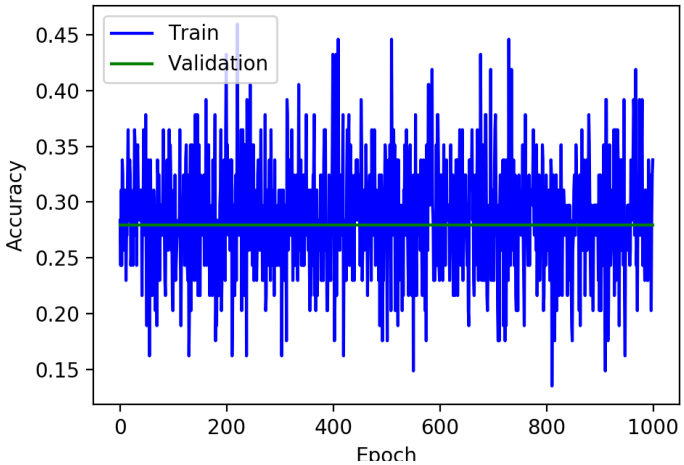


Below are graphs for $\epsilon = 0.5$.

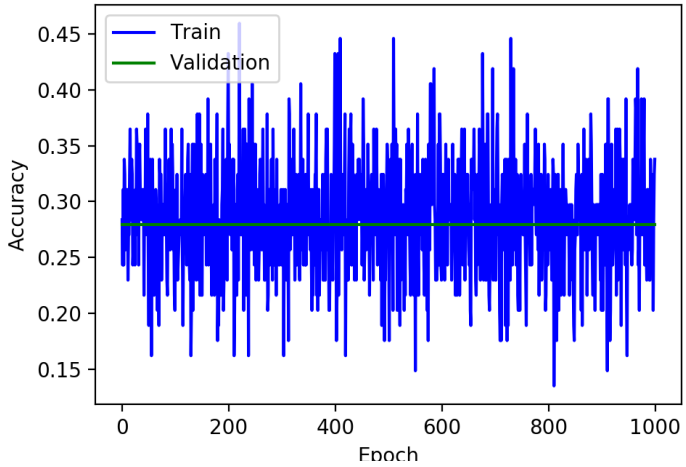




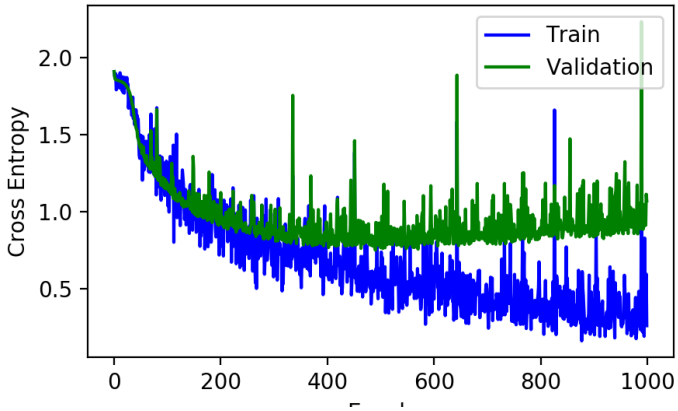
Below are graphs for $\epsilon = 1$.

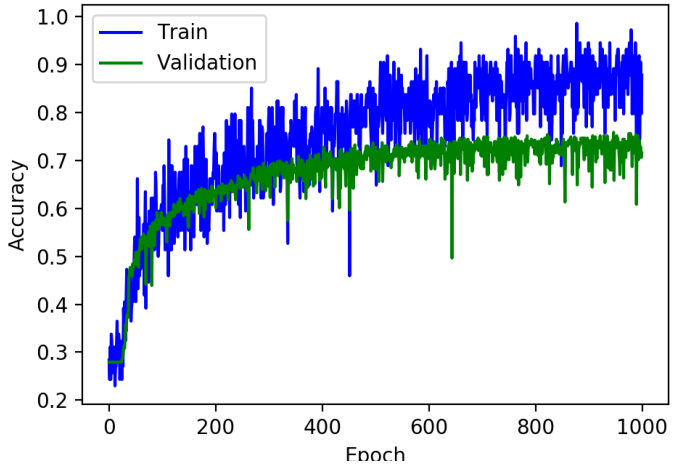


Observe graphs above we could conclude that with ϵ increasing, the convergence will be lost.

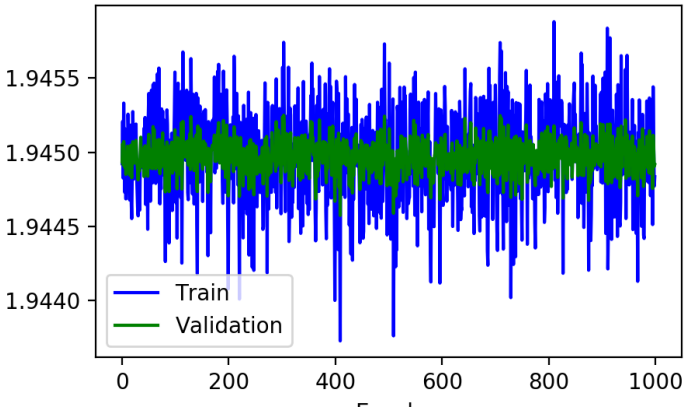


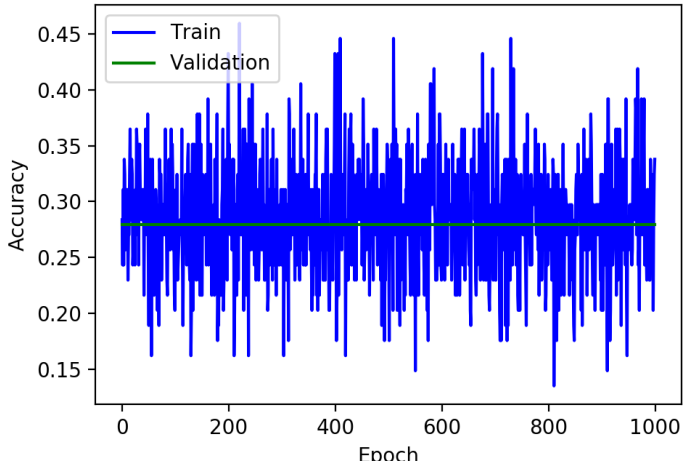
I choose momentum as $0, 0.3, 0.9$
Below are graphs for $momentum = 0$.



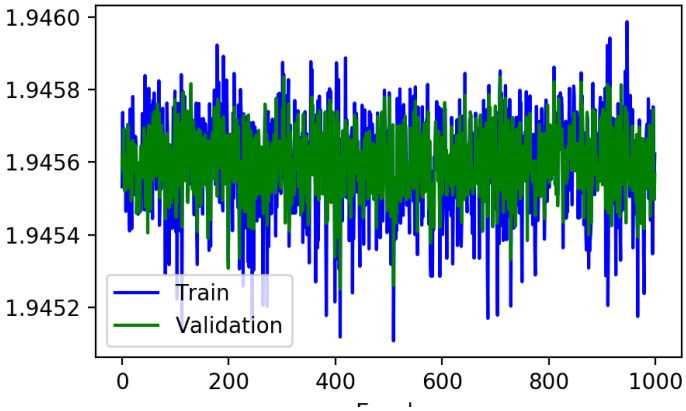


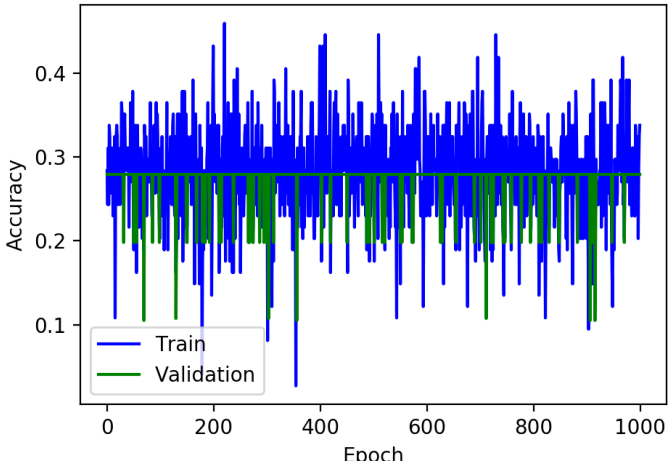
Below are graphs for $momentum = 0.3$.



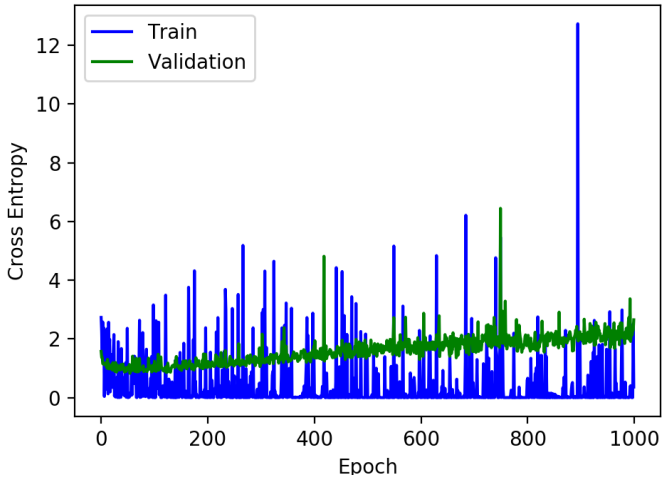


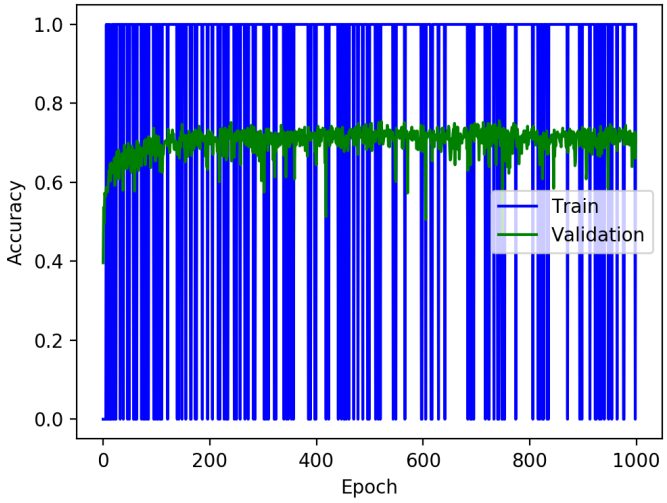
Below are graphs for $momentum = 0.3$.
With momentum increasing, the model converges fast and becomes more inaccurate since it detects more noise.



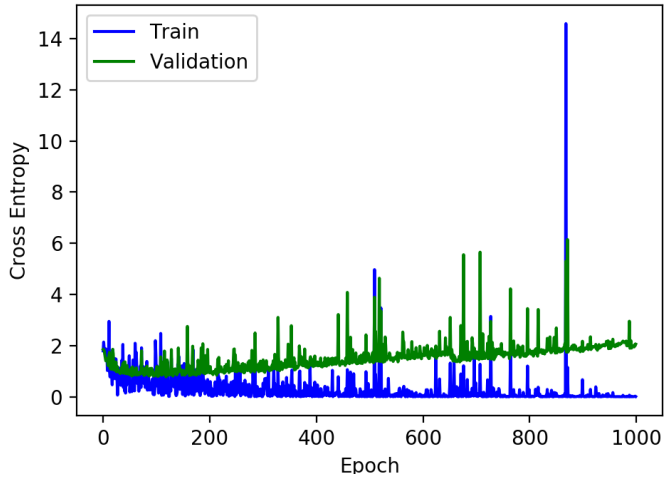


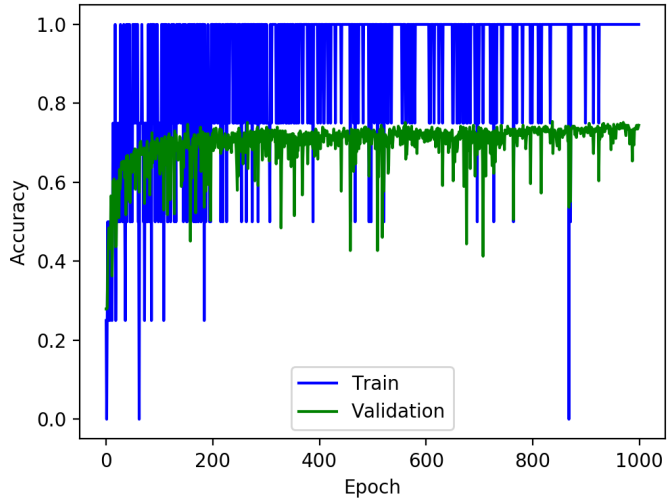
I choose $batch = 1, 10, 100, 500, 1000$
Below is the graph with $batch = 1$



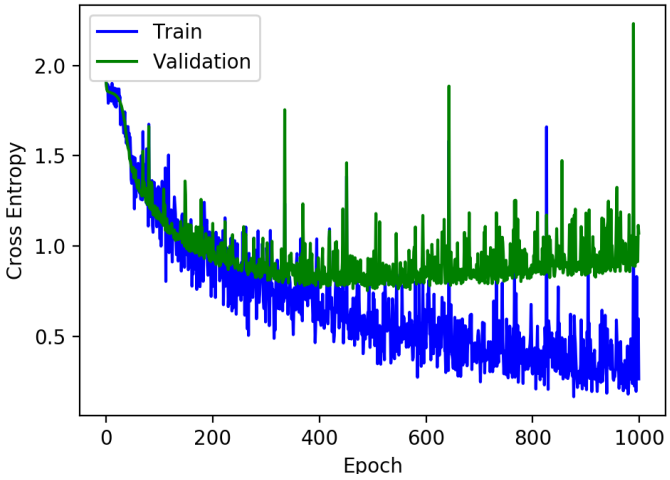


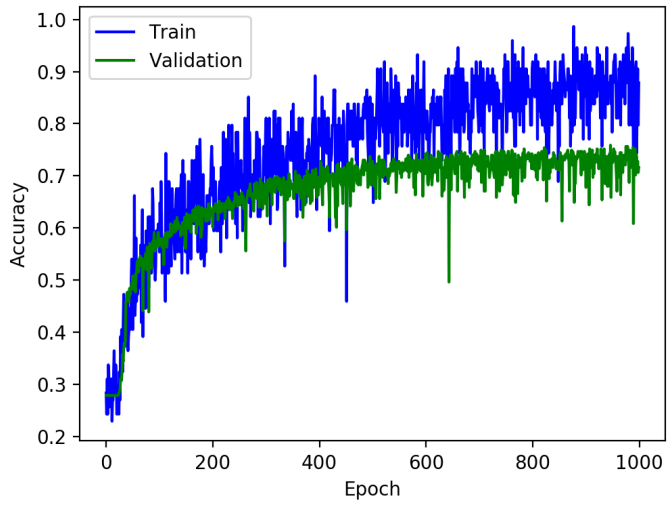
Below is the graph with $batch = 10$



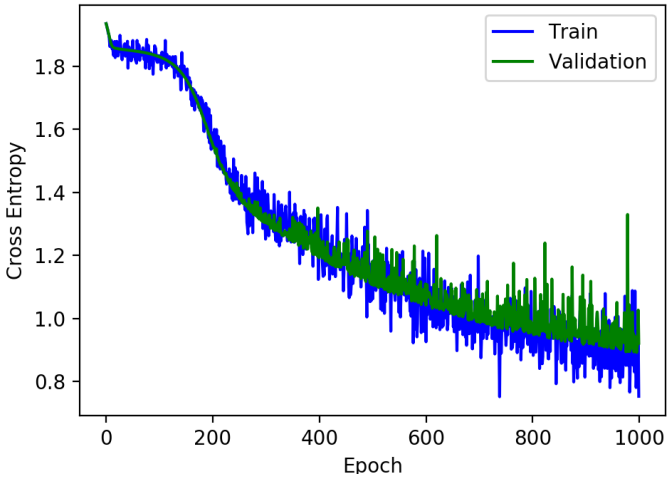


Below is the graph with $batch = 100$

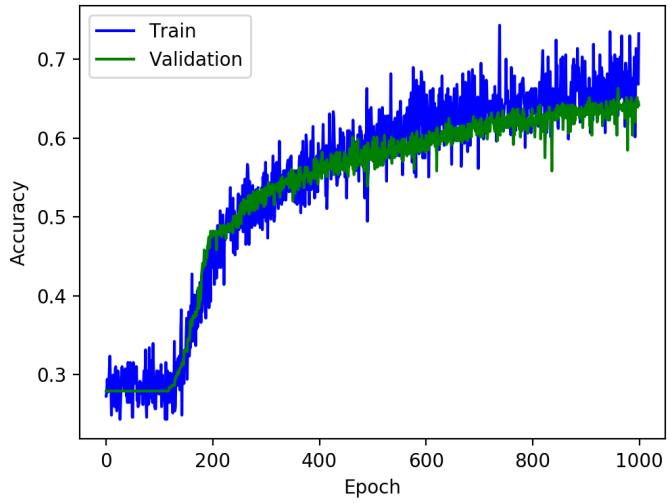


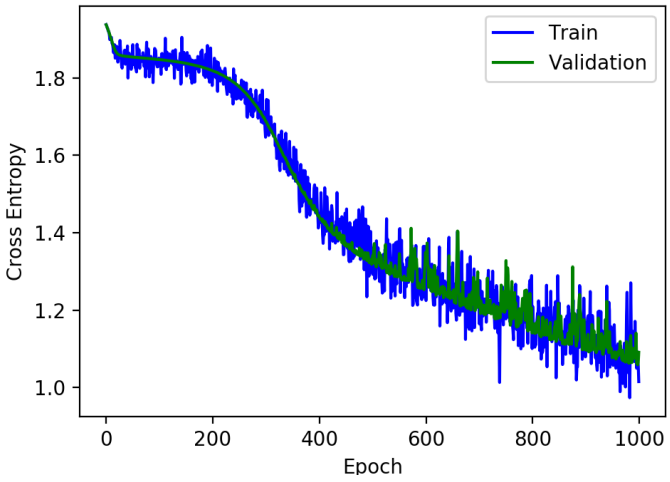


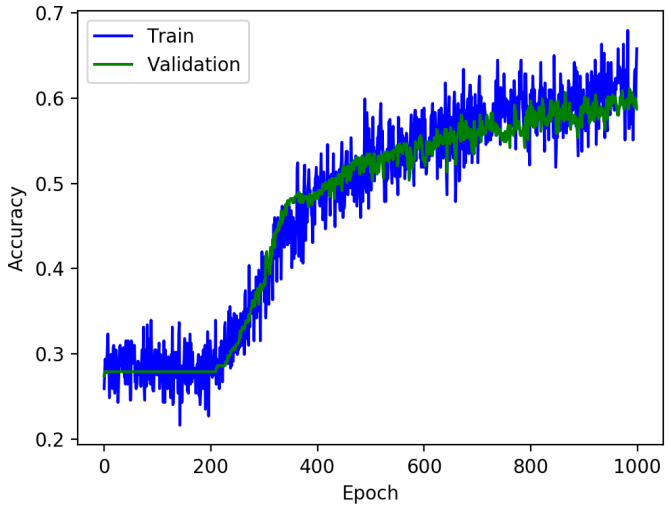
Below is the graph with $batch = 500$



Below is the graph with $batch = 1000$


Train
Validation
Accuracy
0.3
0.4
0.5
0.6
0.7
0
200
400
600
800
1000
Epoch

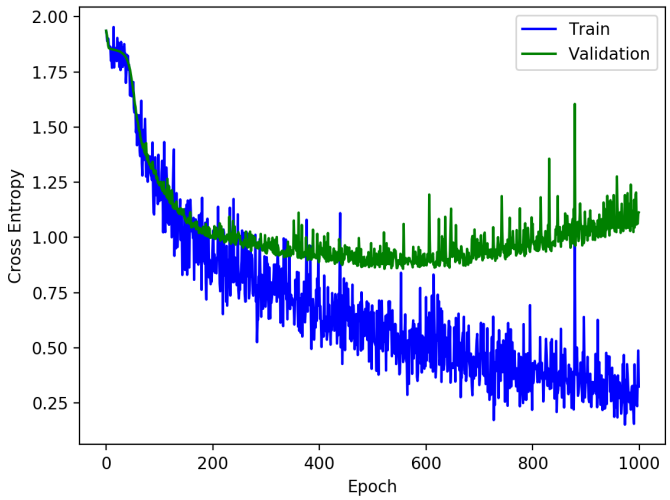

Train
Validation
Cross Entropy
1.0
1.2
1.4
1.6
1.8
0
200
400
600
800
1000
Epoch

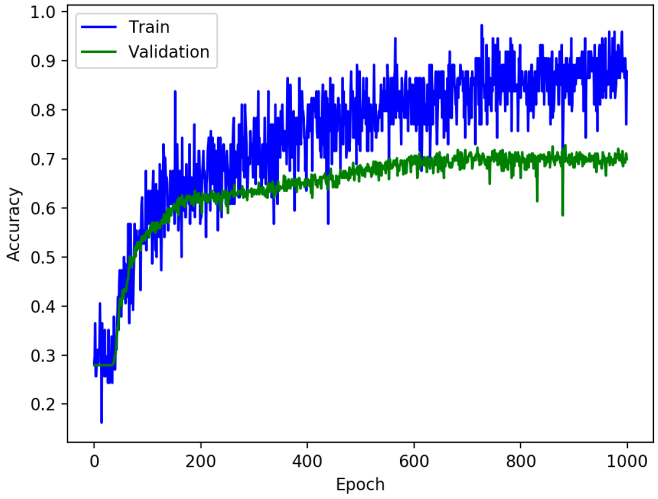


Conclude graphs above, with batch number increasing, the model converges slower.
So the best model I choose would have learning rate at 0.001, momentum at 0.1, batch at 100. In this way the model I created would have strong convergence property and is not over fitted.

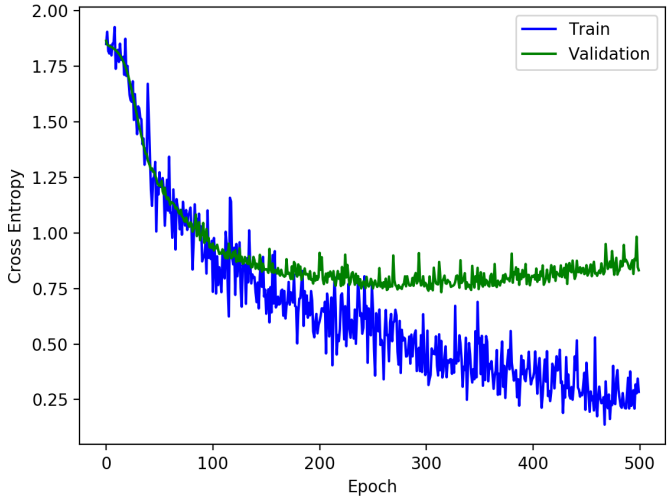
Question 3.3

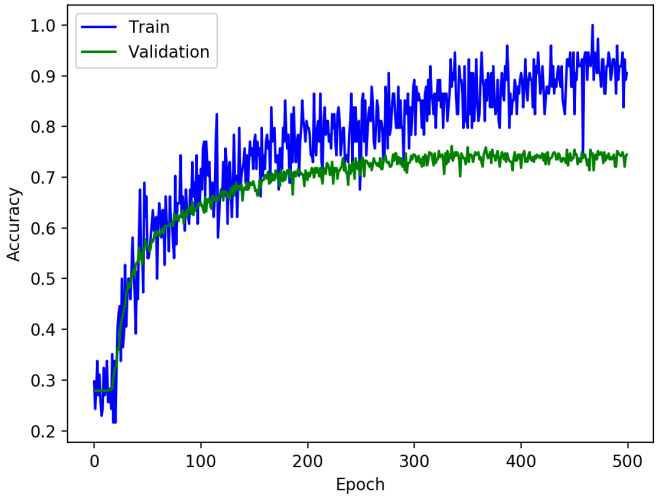
Below I choose hidden layer as [8, 16], and 1000 epoches.



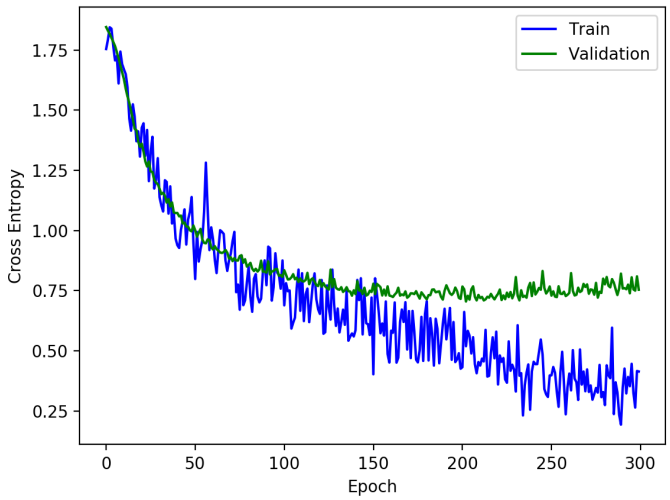


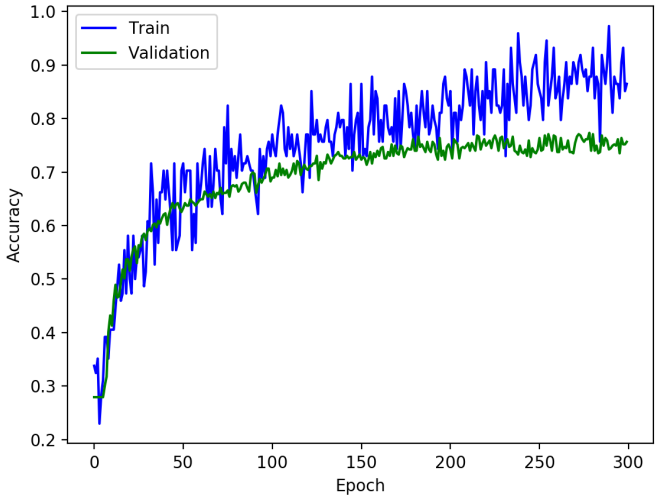
Below I choose hidden layer as [32, 64], and 500 epoches.





Below I choose hidden layer as [64, 128], and 300 epoches.

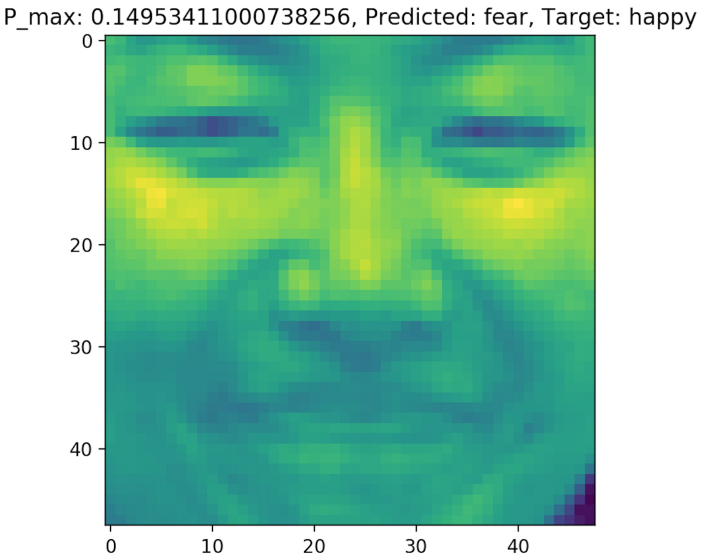


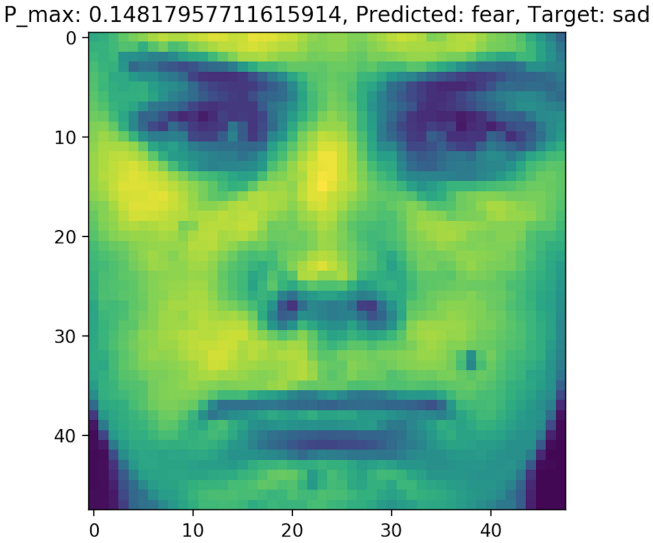


With hidden layer size increasing, it takes shorther time to converge, and the cross entropy of the validation set is also decreasing with the hidden layer size. (eg ce for validation set in [8, 16] layer is over 1, while ce for validation set in [64, 128] layer decreases down to around 0.75). Also, the accuracy for validation set increases slightly when the size of hidden layer increase. Here I conclude that with hidden layer increasing from a relatively small number to a properly large number, the model would perform better and the speed of convergence process also increases.

Question 3.4

Here I provided two examples.





The above two examples are misclassified my the model. The reason is softmax would converted the calculated value to a number with range [0, 1]. In this case, if a feature is not that obvious could easily be misclassified, leading to the predicted category is not the target one.