Assignment 1 Theory Problem Set

DO NOT TAG

Name:
GT Email:

Theory PS Q1. Feel free to add extra slides if needed.

1.

$$\frac{\partial s_i}{\partial z_i} = e^{z_i}\left(\frac{1}{\sum_k e^{z_k}}\right)' + \frac{e^{z_i}}{\sum_k e^{z_k}}$$

$$= \frac{e^{z_i}}{\sum_k e^{z_k}} - \left(\frac{e^{z_i}}{\sum_k e^{z_k}}\right)^2 = s_i(1-s_i)$$

$$\frac{\partial s_i}{\partial z_j} = e^{z_i}\left(-\frac{e^{z_j}}{\left(\sum_k e^{z_k}\right)^2}\right) = -\frac{e^{z_i}e^{z_j}}{\left(\sum_k e^{z_k}\right)^2} = -s_i \cdot s_j$$

$$\therefore \quad \frac{\partial s_i}{\partial z_j} = \begin{cases} s_i(1-s_i), & \text{if } i=j \\ -s_i s_j, & \text{otherwise} \end{cases}$$

The Jacobian of $\frac{\partial s}{\partial z} = \begin{pmatrix} \frac{\partial s_1}{\partial z_1} & \frac{\partial s_1}{\partial z_2} & \cdots & \frac{\partial s_1}{\partial z_n} \\ \frac{\partial s_2}{\partial z_1} & & \ddots & \vdots \\ \vdots & & & \\ \frac{\partial s_n}{\partial z_1} & \cdots & & \frac{\partial s_n}{\partial z_n} \end{pmatrix}$

$$= \begin{pmatrix} s_1(1-s_1) & -s_1 \cdot s_2 & \cdots \\ -s_2 \cdot s_1 & s_2(1-s_2) & \\ \vdots & & \ddots \\ & & & s_n(1-s_n) \end{pmatrix}$$

Theory PS Q2. Feel free to add extra slides if needed.

2.

$$W_{AND} = (1, 1)$$

$$b_{AND} = -1.5$$

$$W_{OR} = (1, 1)$$

$$b_{OR} = -0.5$$

Theory PS Q3. Feel free to add extra slides if needed.

3. If XOR can be represented using a linear model

$W_{XOR} = (w_1, w_2)$

$b_{XOR} = b$

We have

$$\begin{cases} b < 0 & ① \\ w_1 x_1 + b \geq 0 & ② \\ w_2 x_2 + b \geq 0 & ③ \\ w_1 x_1 + w_2 x_2 + b < 0 & ④ \end{cases}$$

②+③−④: $b > 0$ contradicts with ①

∴ XOR can not be represented using a linear model.

Assignment 1 Writeup

DO NOT TAG

Name:
GT Email:

Two-Layer Neural Network

DO NOT TAG

1. Learning Rates

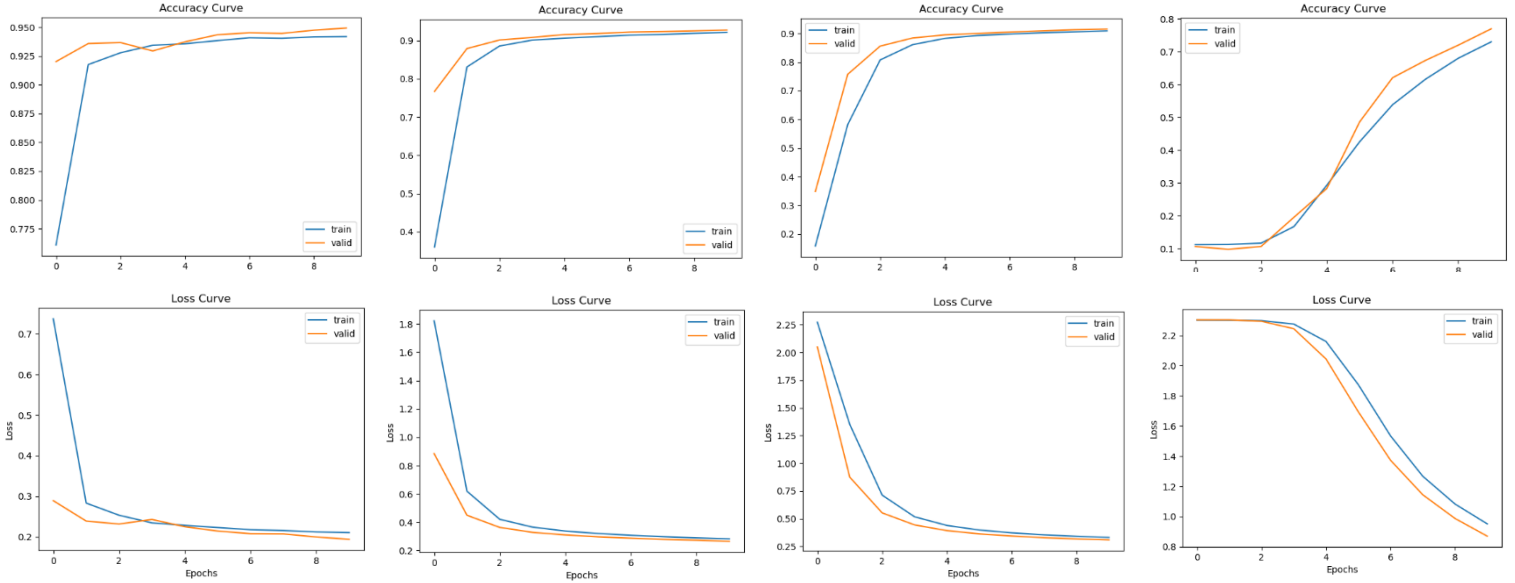
Tune the learning rate of the model with all other default hyper-parameters fixed. Fill in the table below:

	lr=1	lr=1e-1	lr=5e-2	lr=1e-2
Training Accuracy	0.9420	0.9216	0.9090	0.7301
Test Accuracy	0.9502	0.9281	0.9114	0.7581

1. Learning Curve

Plot the learning curves using the learning rates from the previous slide and put them below (you may add additional slides if needed).

lr=1	lr=1e-1	lr=5e-2	lr=1e-2



1. Learning Rates

Describe and Explain your findings: *Explanation should go into **WHY** things work the way they do in the context of Machine Learning theory/intuition, along with justification for your experimentation methodology. **DO NOT** just describe the results, for example, you should explain why the learning rate has the observed effect. Also, be cognizant of the best way to organize and show the results that best emphasizes your key observations. If you need more than one slide to answer the question, you are free to create new slides.*

Findings: For smaller learning rates, the accuracy is lower, and the loss is higher for both training and test data.

It is because as we are using smaller learning rate, the weight updates are smaller for each update step. It makes the models not fully optimized, and the model is under-fitting. Also training loss and validation loss are very close for large learning rate, which means the model is still under-fitting at large learning rate.

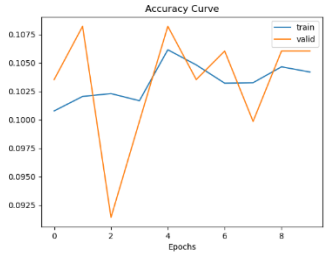
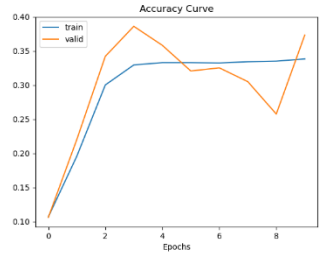
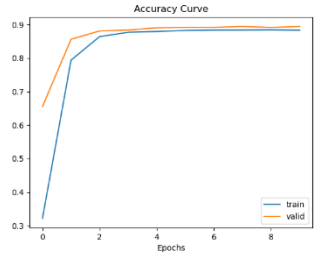
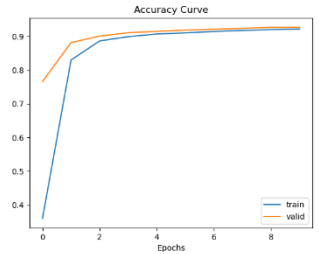
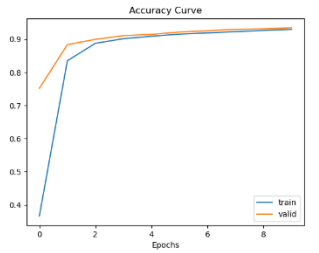
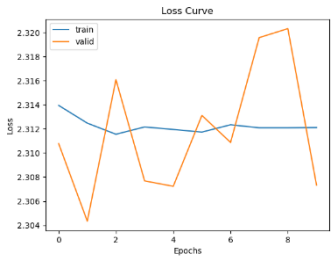
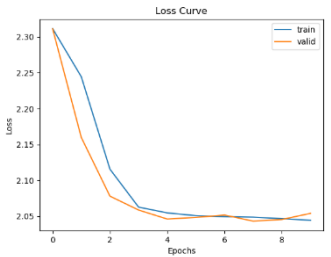
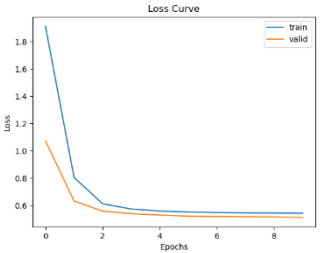
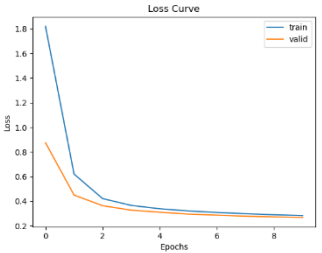
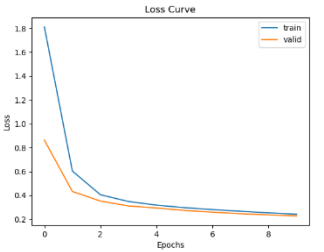
2. Regularization

Tune the regularization coefficient of the model with all other default hyper-parameters fixed. Fill in the table below:

	alpha=1	alpha=1e-1	alpha=1e-2	alpha=1e-3	alpha=1e-4
Training Accuracy	0.1042	0.3382	0.8831	0.9214	0.9299
Validation Accuracy	0.1060	0.3732	0.8943	0.9266	0.9348
Test Accuracy	0.1029	0.3896	0.8928	0.9232	0.9336

2. Regularization

Plot the learning curves using the regularization coefficients from the previous slide and put them below (you may add additional slides if needed).

alpha=1	alpha=1e-1	alpha=1e-2	alpha=1e-3	alpha=1e-4
				
				

2. Regularization

Describe and Explain your findings: *Explanation should go into **WHY** things work the way they do in the context of Machine Learning theory/intuition, along with justification for your experimentation methodology. **DO NOT** just describe the results, for example, you should explain why the regularization value affects performance as well as model weights. Also, be mindful of the best way to organize and show the results that best emphasizes your key observations. If you need more than one slide to answer the question, you are free to create new slides.*

Findings: For smaller regularization value alpha, the accuracy is lower, and the loss is higher for both training and test data.

It is because as we are using larger regularization, the weight are regularized and very similar to an average model, model complexity is sacrificed for generalization ability. It makes the models not fully optimized, and the model is under-fitting. Also training loss and validation loss are very close for large learning rate, which means the model is still under-fitting at large learning rate.

3. Hyper-parameter Tuning

You are now free to tune any hyper-parameters for better accuracy. Create a table below and put the configuration of your best model and accuracy into the table:

batch_size	learning_rate	reg	epochs	momentum	hidden_size	train accuracy	valid accuracy	test accuracy
64	1	1E-04	40	0.9	256	0.9907	0.9757	0.9789

Explain why your choice works: *Explanation should go into **WHY** things work the way they do in the context of Machine Learning theory/intuition, along with justification for your experimentation methodology. **DO NOT** just describe the results, you should explain the reasoning behind your choices and what behavior you expected. Also, be cognizant of the best way to mindful and show the results that best emphasizes your key observations. If you need more than one slide to answer the question, you are free to create new slides.*

For learning rate: I choose lr=1, because we can see from the result (1) that the model is under-fitting with lower learning rate. So we should use large learning rate to adjust the update step to a higher one.

For regularization, I choose alpha=1e-4. We can see from the result (2) that the model performs better with lower regularization. Model complexity is sacrificed for generalization ability. But the model is still of good generalization ability, so we can use lower alpha to increase the model complexity.

For hidden_size and epochs, we have found that model is under-fitting with train/val loss still decreasing. We should increase the model complexity with higher hidden size and use higher epochs to optimized the models to get a minimum.