Assignment 1 Theory Problem Set

DO NOT TAG

Name: Xiaohu Zhang
GT Email: xzhang869@gatech.edu

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

(1)Gradient of softmax

Find the gradient of softmax with respect to the logits:

$$s_i = \frac{e^{Z_i}}{\sum_k e^{Z_k}} \tag{1}$$

We need to find:

$$\frac{\partial s_i}{\partial Z_j} = \frac{\partial \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right)}{\partial Z_j} \tag{2}$$

Let's break it down into two cases: when $i = j$ and when $i \neq j$.
For $i \neq j$: Consider the following example:

$$\frac{\partial}{\partial Z_3} \left(\frac{e^{Z_1}}{e^{Z_2} + e^{Z_3}} \right) = \frac{0 - e^{Z_1} \cdot e^{Z_3}}{\left(e^{Z_2} + e^{Z_3}\right)^2} \tag{3}$$

The general case for $i \neq j$ becomes:

$$\begin{aligned} \frac{\partial s_i}{\partial Z_j} &= \frac{\partial}{\partial Z_j} \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right) \\ &= \frac{0 \cdot \sum_k e^{Z_k} - e^{Z_i} \cdot e^{Z_j}}{\left(\sum_k e^{Z_k}\right)^2} \\ &= -s_i \cdot s_j \end{aligned}$$

For $i = j$:

$$\begin{aligned} \frac{\partial s_i}{\partial Z_i} &= \frac{\partial}{\partial Z_i} \left(\frac{e^{Z_i}}{\sum_k e^{Z_k}} \right) \\ &= \frac{e^{Z_i} \cdot \sum_k e^{Z_k} - e^{Z_i} \cdot e^{Z_i}}{\left(\sum_k e^{Z_k}\right)^2} \\ &= s_i \left(1 - s_i\right) \end{aligned}$$

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

(2)AND OR Operation

Find W_{AND} and b_{AND} satisfy the conditions,assume if b_{AND} is -1.1 and $W_{AND} = [1, 1]$

$$
\begin{aligned}
f(x_1) &= 0 + 0 - 1.1 < 0, 0 \\
f(x_2) &= 0 + 1 - 1.1 < 0, 0 \\
f(x_3) &= 0 + 0 - 1.1 < 0, 0 \\
f(x_4) &= 1 + 1 - 1.1 \geq 0, 1
\end{aligned}
$$

Find W_{OR} and b_{OR} satisfy the conditions, $W_{OR} = [1, 1]$ and $b_{OR} = 1$

$$
\begin{aligned}
f(x_1) &= 0 + 0 - 1 < 0, 0 \\
f(x_2) &= 0 + 1 - 1.1 = 0, 1 \\
f(x_3) &= 1 + 0 - 1 = 0, 1 \\
f(x_4) &= 1 + 1 - 1 \geq 0, 1
\end{aligned}
$$

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

(3) XOR Operation

If I plot the X_1 and X_2 as well as the XOR function in Figure 1, with X_1 being the class 0 and X_2 being the class 1, it would be impossible to draw a straight line to separate the two classes.

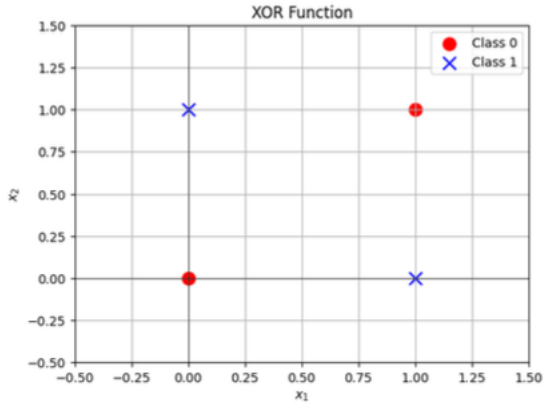


Figure 1: XOR Function with linear splitting

Assignment 1 Paper Review

DO NOT TAG

Name: Xiaohu Zhang
GT Email: xzhang869@gatech.edu

Provide a short preview of the paper of your choice.

Paper: 'Weight Agnostic Neural Networks' by Adam Gaier and David Ha at Google Brain

1. What is the main contribution of this paper? In other words, briefly summarize its key insights. What are some strengths and weaknesses of this paper?

Key Contribution

This paper fundamentally changed our understanding of neural networks. We used to believe that neural networks must tune their weights through the gradient descent algorithm. However, the paper proposes an approach that eliminates the need for weight learning by assigning shareable equal weights to each layer of the neural network and searching for optimal neural architectures. This method achieves significant performance on certain supervised learning tasks and generates high accuracy on the MNIST dataset. The method begins with shared weights across all layers and connections in the network. It then searches for the optimal network based on its complexity and performance.

Strengths

This approach is significantly different from Neural Architecture Search (NAS), which considers weight training as a key step. The Weight Agnostic Neural Network (WANN) approach minimizes the impact of weights and focuses on discovering the optimal network architecture. By not relying on weights, WANN avoids the drawbacks of traditional networks that require gradient de- scent methods, such as issues like gradient explosion or vanishing. The reduction in training requirements can further be applied to cases like few-shot learning.

Weaknesses

Based on the paper's performance, the weights still played a significant role in the study. The ensemble weights, as well as the trained and tuned weights, performed much better compared to the random weights. The weights will still play a role if we want to achieve more accurate results. Additionally, the author acknowledged that this method cannot outperform the well-known and standard CNN approach. Furthermore, the study in the paper included classification tasks but mainly focused on games with a rich amount of data available for training. It remains unclear whether the network can also work on more complex cases.

Paper specific Q1. Feel free to add extra slides if needed.

2. What is your personal takeaway from this paper?

My first impression of this paper's approach to searching for the optimal net- work structure is that it is very similar to searching for a policy in reinforcement learning.

Additionally, the weight settings remind me of the Kaiming initialization algorithm used by PyTorch, as proposed in He (2015). In Kaiming's paper, he proposes a method for assigning initial weights to ensure gradient stability. This approach contrasts with the method in this paper, which uses randomly generated weights to make the neural network less dependent on the weights.

Last but not least, the same author, Kaiming He, published another pa- per in He 2019. arguing that a randomly generated network can also achieve significant accuracy on the ImageNet benchmark, which also seems to emphasize that a randomly generated network architecture could work, as opposed to the complex network search methods performed in the WANN paper. My key takeaway is that the weights and network architecture are both very important, and the results may highly depends on the tasks applied for the algorithm.

Paper specific Q2. Feel free to add extra slides if needed.

3. **The traditional view of optimization in deep learning (and often in general) is that we are searching the space of weights to find the best ones. In other words, learning is a search problem. How would you view the above paper from the perspective of search?**

In my view, the WANN paper shift the focus of searching from searching weights to searching the optimal architecture. This way reframing the learning problem as well as the traditional gradient descent approach as the topology search in order to identify the ideal network structure, similar to the policy in the reinforcement learning space. The optimization algorithm is also different as the loss function is no longer the target, performance and complexity is the key considerations in such searching process.

4. **One of the key aspects of deep learning is that given a parameterized function, we can find weights to represent any functionif it has sufficient depth and complexity. What does this paper say about the representational power of architectures given a fixed method for determining weights? Does the method for determining the weights matter? Do you think these two have equal representational power? Why or why not?**

Fixed method for determining weights:
The paper did some study using a range of weights or a single weights, random weights etc. to search for the network architecture. It is found that the fixed value of the weight various significantly when the fixed value had marginal changes, which will failed the task.
Methods of determine the weights:
The method really matters, as mentioned in the paper, random weighs, ensemble weights, training and tuned weights will have significantly different accuracy with 10 percent difference.
Representational Power:
The representational power between the network architecture and the weight settings is not equal. This paper demonstrates that even with random weights, the accuracy can still remain around 80 percent, which indicates that the optimal network structure is more impactful, considering it balances the trade-off between complexity and performance. The paper also suggests that the architecture will have more influence when the network is implemented for a specific purpose. However, in the traditional machine learning context, we should consider the combination of both architecture and weight optimization, which is essential for achieving high performance. The architecture provides a framework for capturing complex relationships, while the weights are adjusted to learn and represent the patterns present in the data.

Assignment 1 Writeup

DO NOT TAG

Name: Xiaohu Zhang
GT Email: xzhang869@gatech.edu

Two-Layer Neural Network

DO NOT TAG

1. Learning Rates

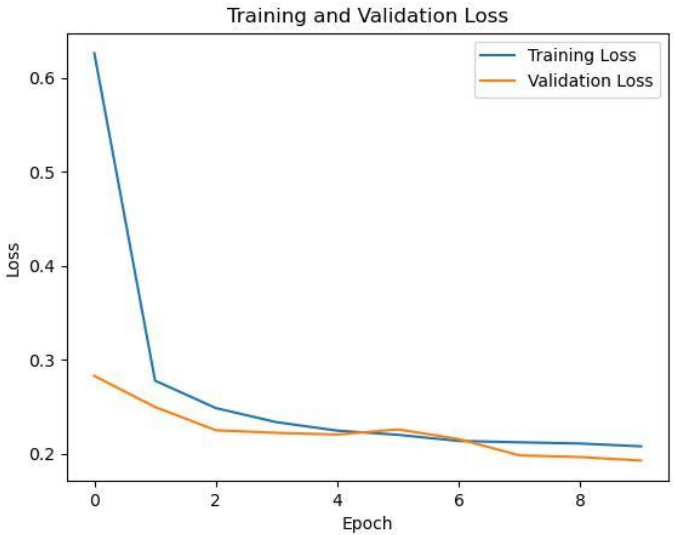
Tune the learning rate of the model with all other default hyper-parameters fixed. Fill in the table below:

	lr=1	lr=1e-1	lr=5e-2	lr=1e-2
Training Accuracy	0.9491	0.9277	0.9164	0.7725
Test Accuracy	0.9490	0.9263	0.9124	0.7604

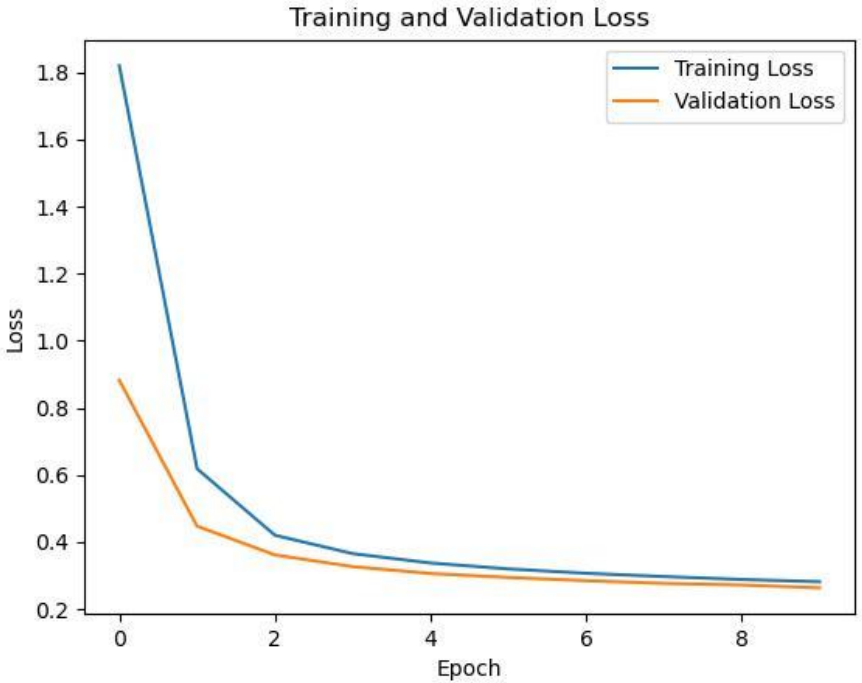
1. Learning Curve

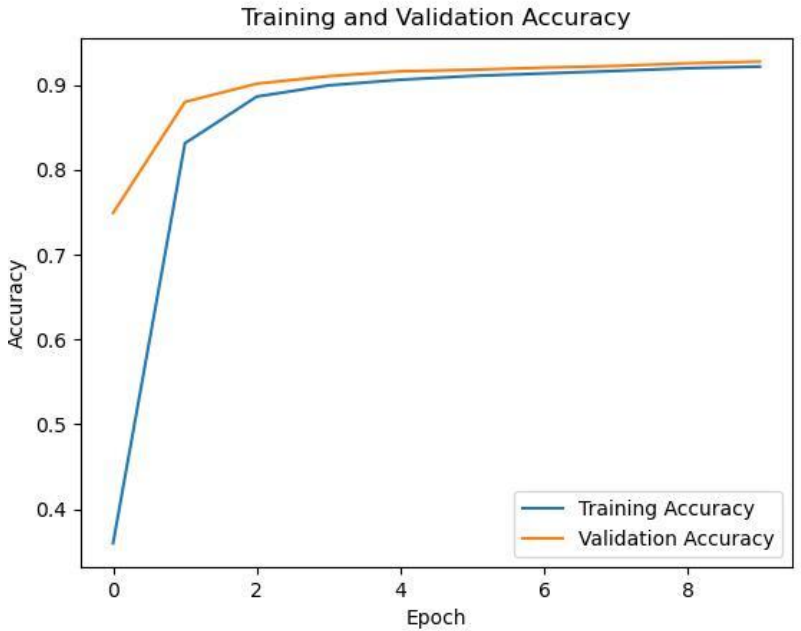
Plot the learning curves using the learning rates from the previous slide and put them below (you may add additional slides if needed).

LR=1

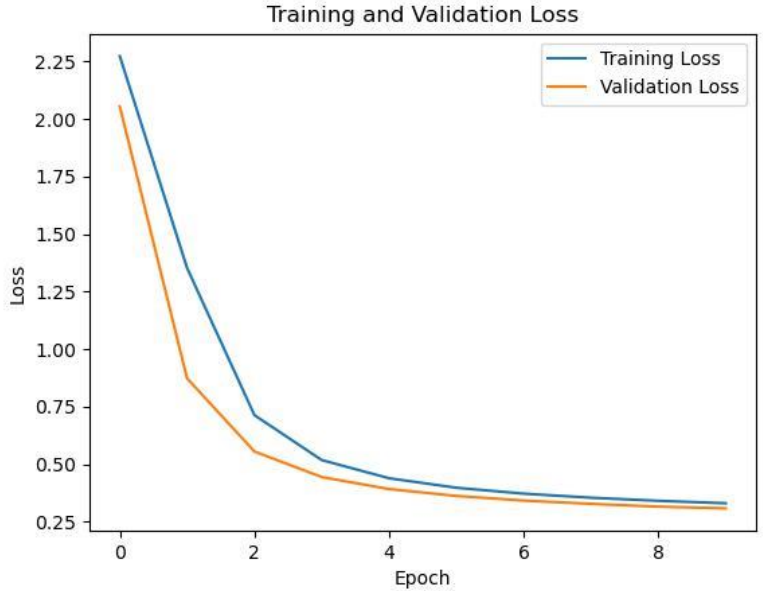


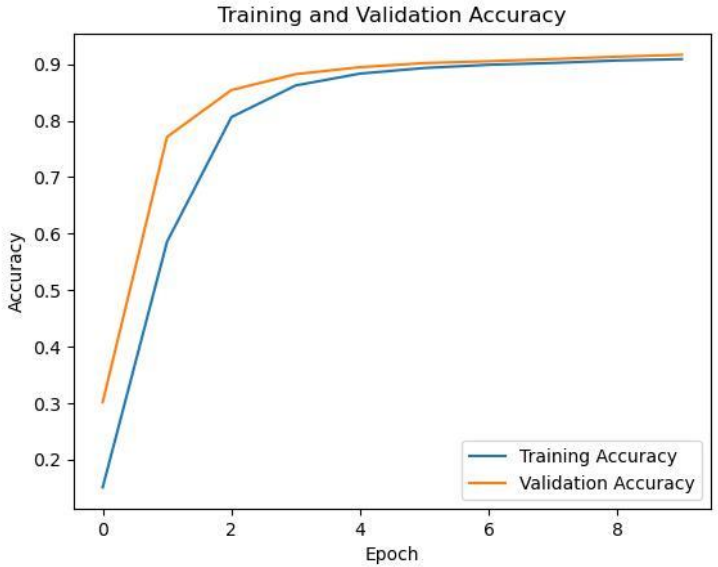
LR=1e-1


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8

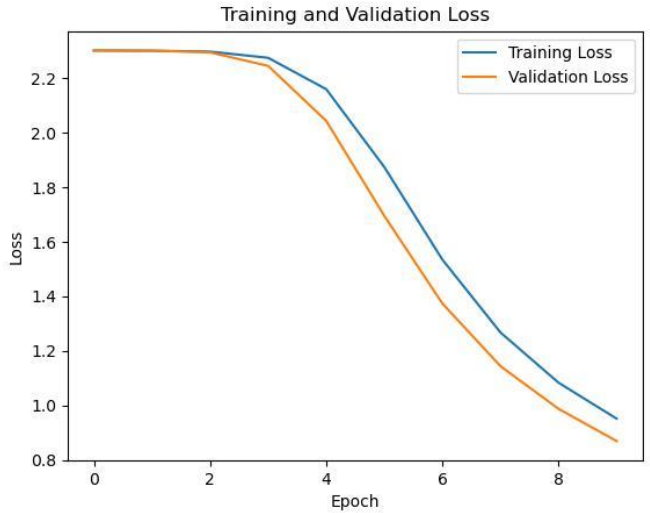

Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.9
0.8
0.7
0.6
0.5
0.4
0
2
4
6
8

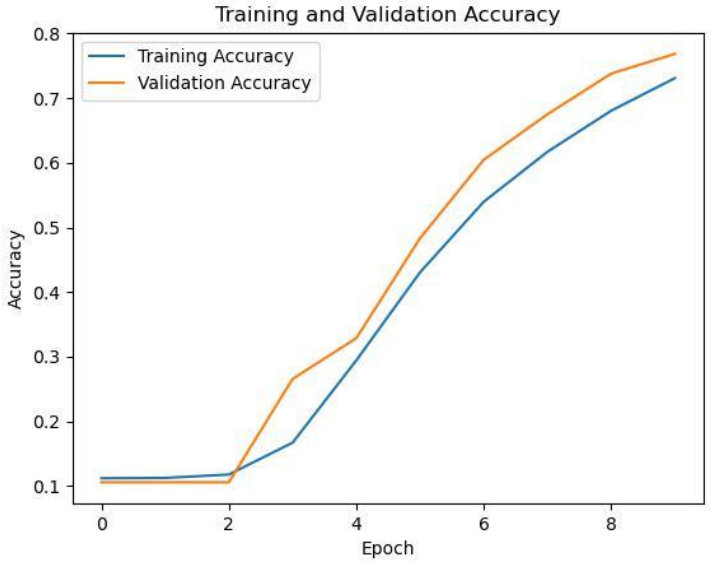
LR=5e-2


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
2
4
6
8


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
2
4
6
8

LR=1e-2


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
0
2
4
6
8


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
2
4
6
8

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.*

Explain: A learning rate of 1 resulted in the best accuracy for both the training and testing datasets. However, the graph shows that the loss experienced some volatility as the number of epochs increased, which may lead to stability issues in the model's performance. In theory, a learning rate of 1 is quite high, causing the weights to change significantly at each step.

The results for learning rates of 0.1 and 0.05 are quite similar. The curve indicates a relatively slow but more stable convergence, suggesting a more consistent model performance compared to when the learning rate is 1.

A learning rate of 0.01 makes the training process very slow, and given that the batch size is not large enough, the model is likely not well-fitted.

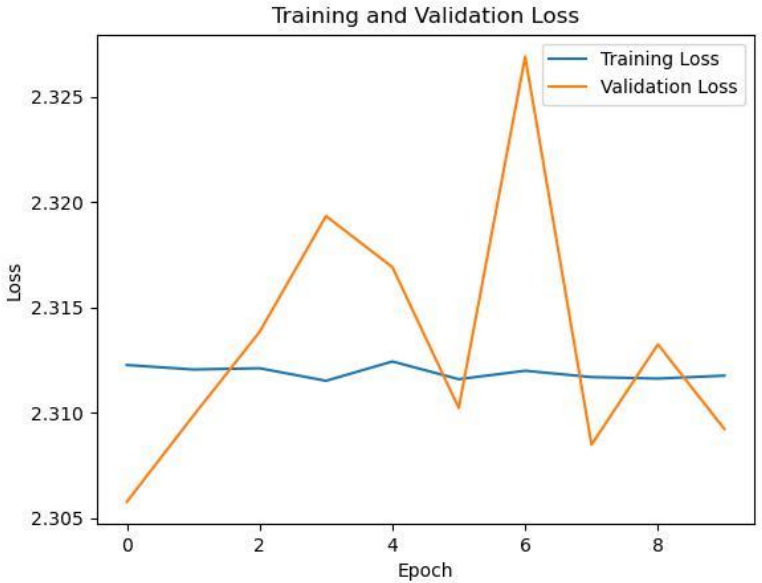
2. Regularization

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	0.1036	0.3354	0.8845	0.9216	0.9295
Validation Accuracy	0.1060	0.3652	0.8953	0.9262	0.9354
Test Accuracy	0.1135	0.3845	0.8913	0.9262	0.9338

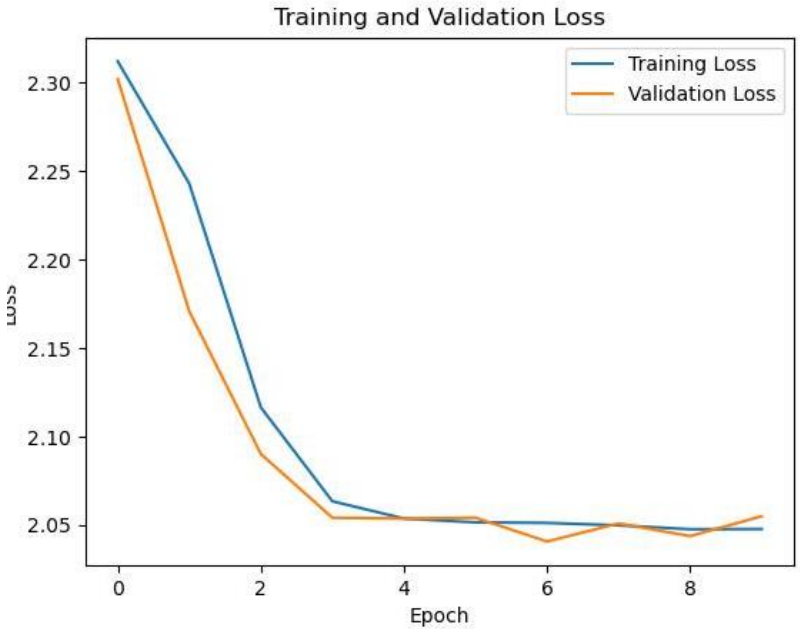
2. Regularization

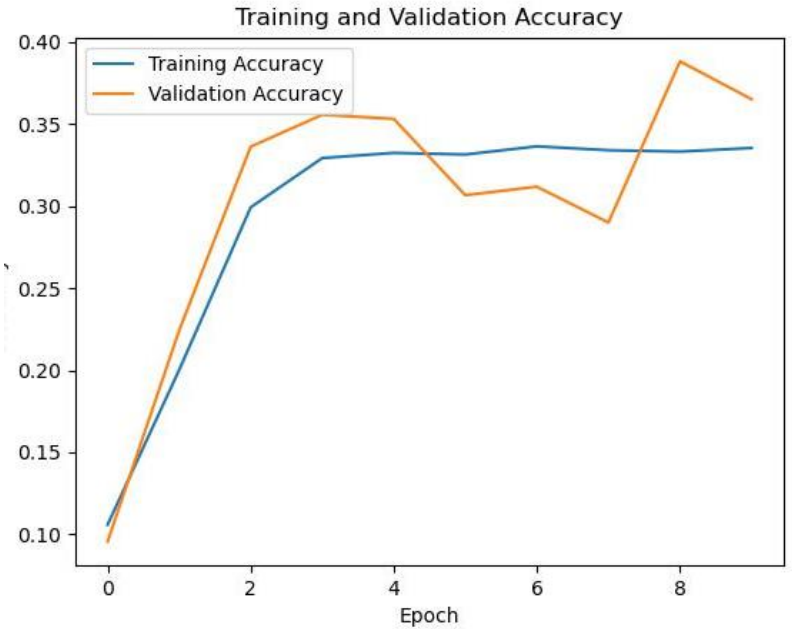
Plot the learning curves using the regularization coefficients from the previous slide and put them below (you may add additional slides if needed).

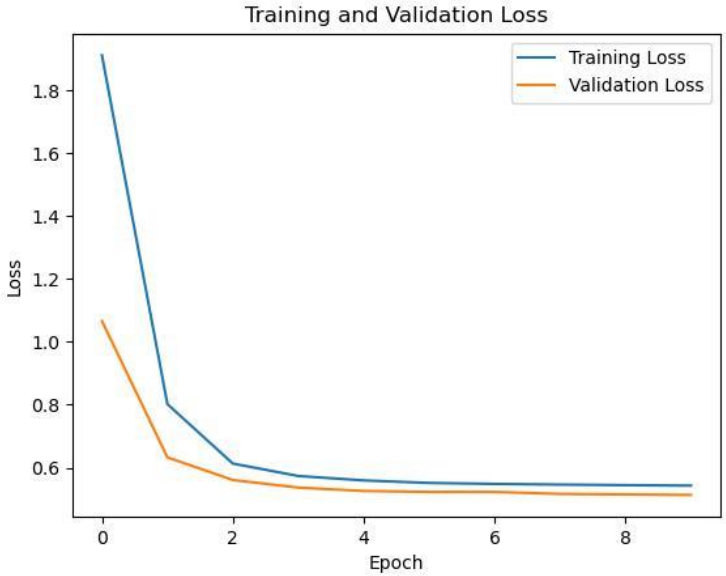


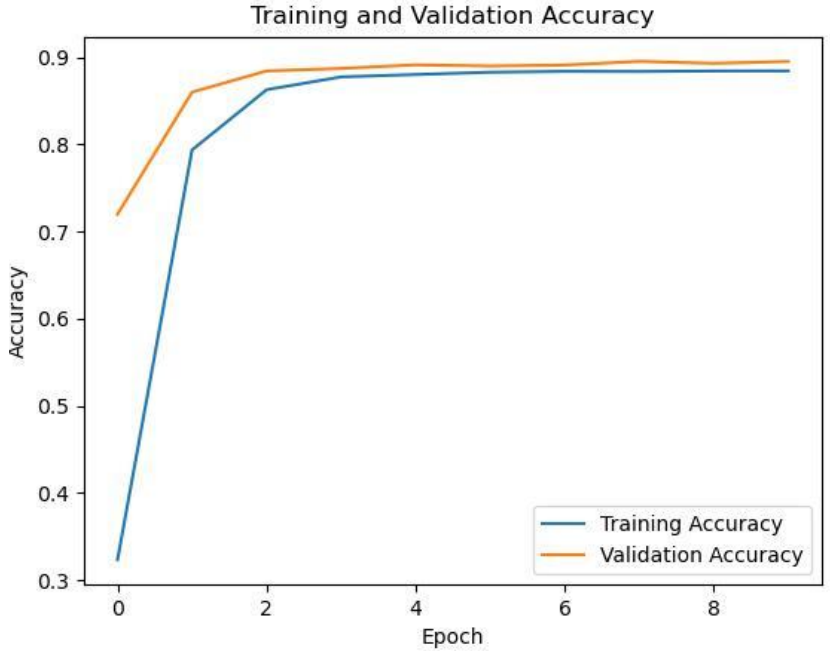
Alpha=1

alpha=1e-1

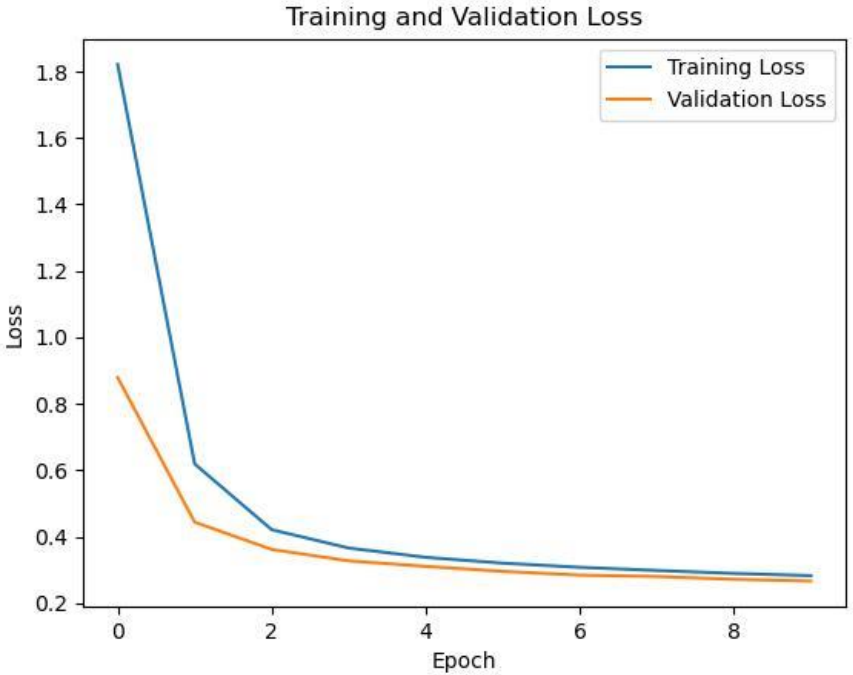

Training and Validation Loss
Training Loss
Validation Loss
2.30
2.25
2.20
2.15
2.10
2.05
0
2
4
6
8
Epoch

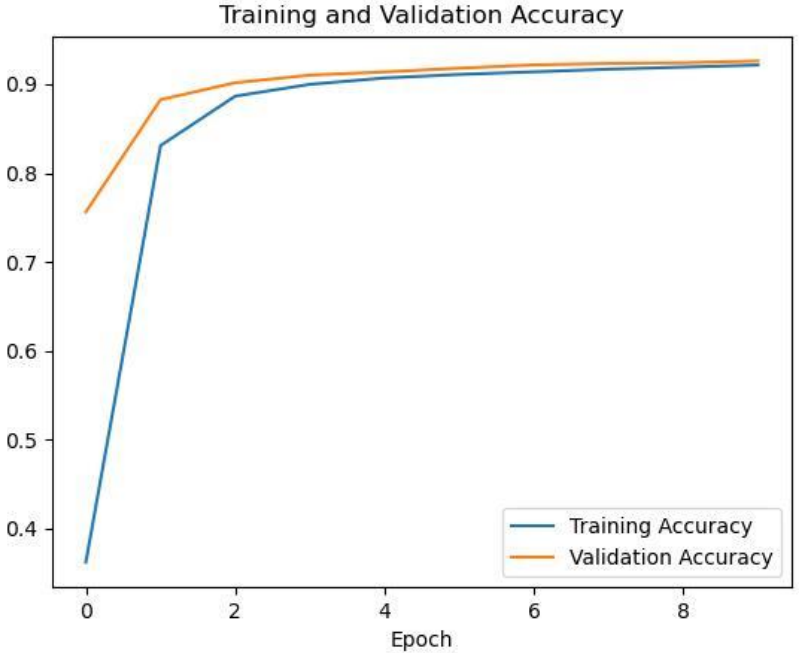

Training and Validation Accuracy
Training Accuracy
Validation Accuracy
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
2
4
6
8
Epoch


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
2
4
6
8

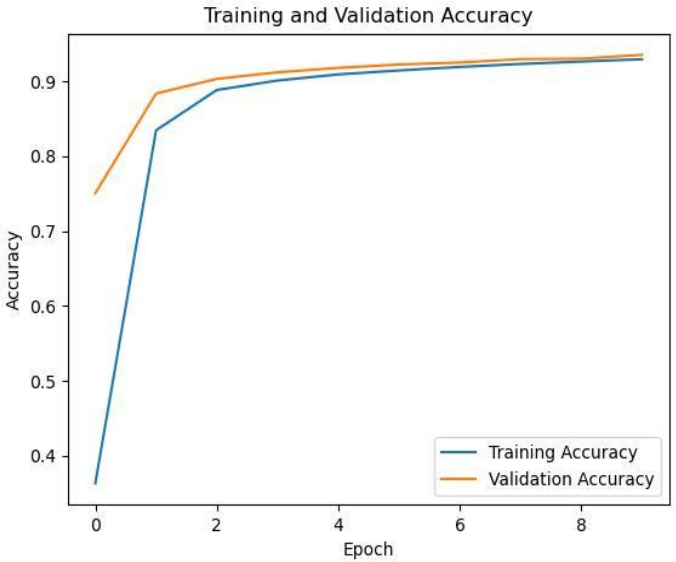

Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8

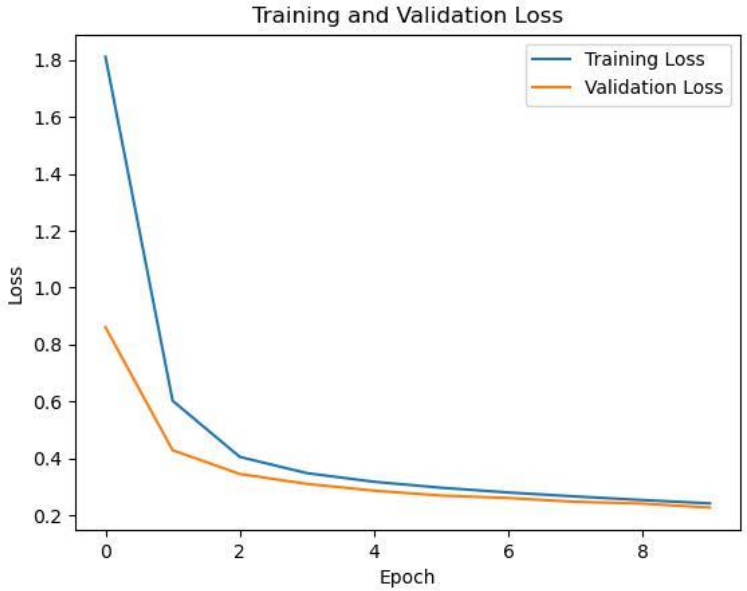
Alpha=1e-2


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Epoch
0.9
0.8
0.7
0.6
0.5
0.4
0
2
4
6
8

Alpha=1e-3


Training and Validation Accuracy
Accuracy
Epoch
Training Accuracy
Validation Accuracy


Training and Validation Loss
Loss
Epoch
Training Loss
Validation Loss

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.*

Answer

When the regularization term is 1, a larger value, the penalization term is just the weights, meaning the overall loss will depend heavily on the weights. When the weights are larger, the model will be penalized more due to the larger weights. As a result, the weights may become smaller, and the linear combination of those features may not accurately reflect the dataset, potentially causing the model to underfit.

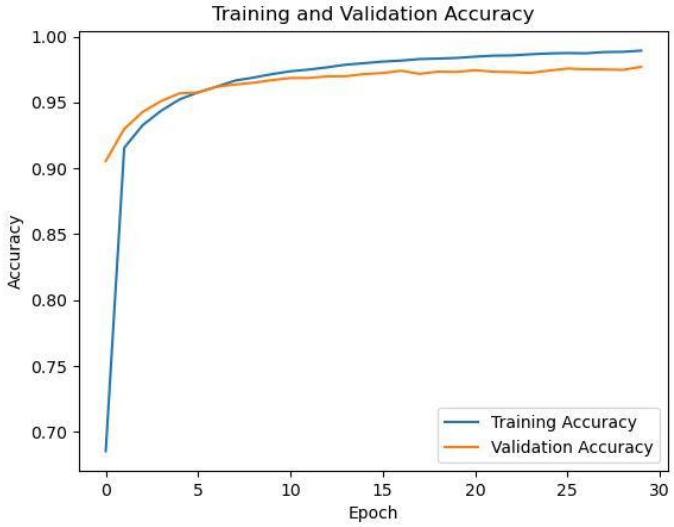
From the table, the lower regularization term with less penalty seems to improve the model's performance. In addition, the small differences between training, testing, and validation accuracy indicate that the lower penalty term is effective in preventing overfitting of the model. The model benefits from these relatively smaller regularization terms by balancing the trade-off between variance and bias

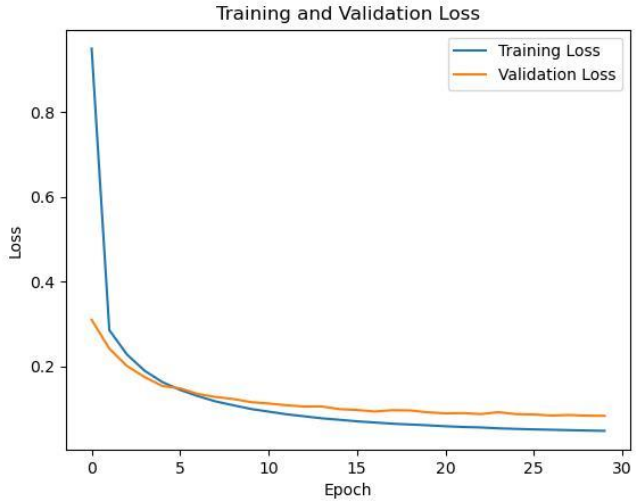
3. Hyper-parameter Tuning

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	epoch	Training Accuracy	Validation Accuracy	Testing Accuracy
128	0.9	0.0001	30	0.9893	0.9771	0.9772

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*





Explain why your choice works

- I first started by tuning the batch size because I think the population is relatively large, and a small sample may not capture the full picture of the model, probably will underfit the model and creating some noise.
- The second step was to increase the number of hidden nodes, as the default model settings might not be adequate for handling complex problems. Increasing the number of hidden nodes will improve model's capacity and performance on capturing the complex non-linear relationship of he features. At the same time, I adjusted the regularization coefficient to help prevent overfitting.
- Although is not required, I adjust the number of epochs aiming to improve the SGD optimization performance.
- I also tried smaller learning rates, starting from 0.01 to 0.05, with the goal of allowing the gradient descent algorithm to gradually identify the optimal weights with stable performance. However, this led to a slowdown in training time and did not significantly enhance model performance. Inspired by part 1 of the assignment, where a relatively larger learning rate still achieved good performance on this dataset, I switched to a relatively large value for the learning rate, aiming to improve training speed while maintaining similar performance. The results show that this approach is effective, as the performance experienced some minor fluctuations but stabilized over the last 10 epochs.