# Project Report


Nandhakumar Vadivel  50604851

Sai Kalyan Veera 50560096

Note : In code for lambda and hidden value are in list you can place my optimal value for testing the code.

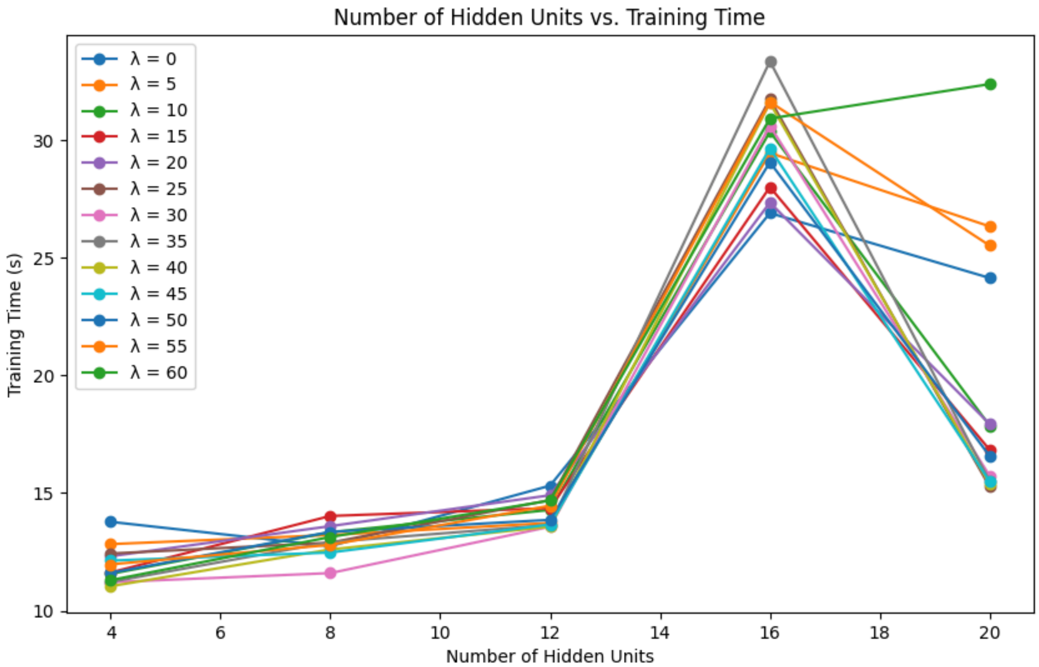
**Choosing of Hyperparemeter for the NN**

So now we can analyze the graph for choosing hyperparameter

Our goal is to achieve the model to be best fit and to be generalized .

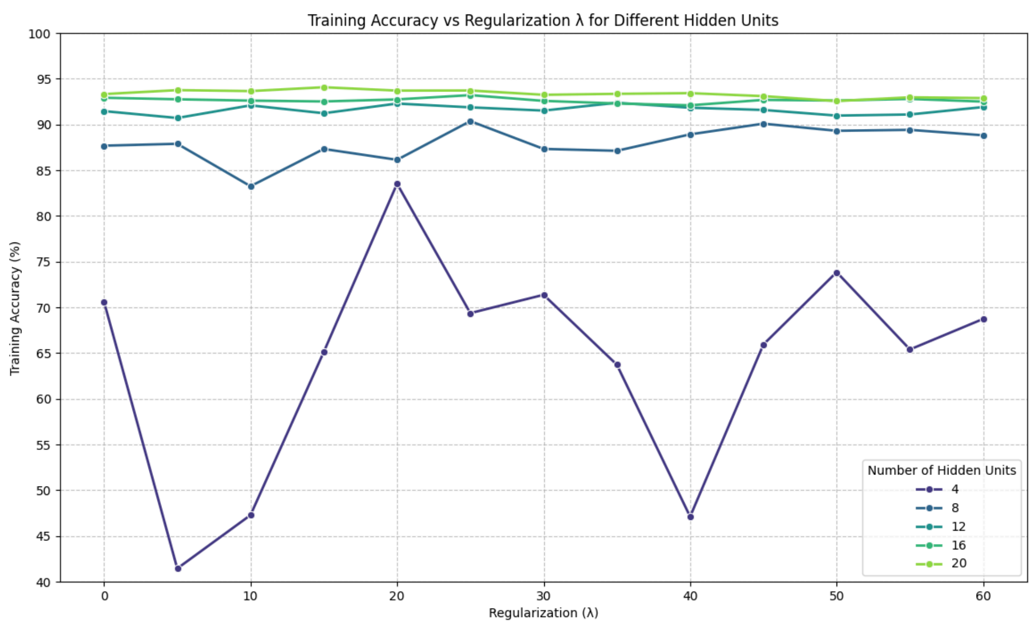
Selecting the number of hidden units .

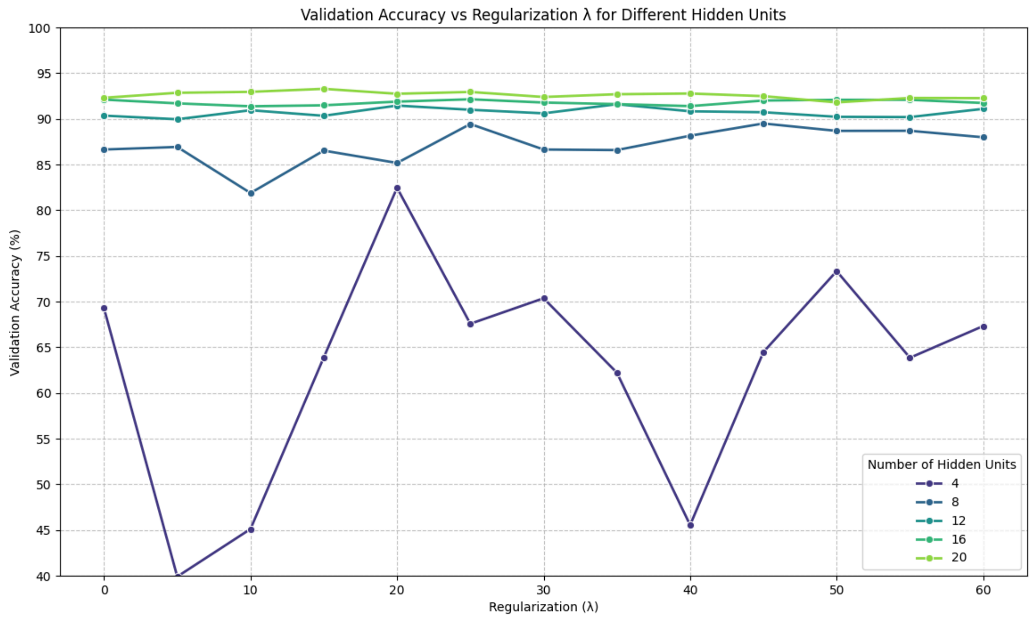
In our project as mention in the instruction we have train the model for 4,8,12,16,and 20 hidden units to analyze the performance at each value .
For the regularization we have followed the same instruction with the increment of 5 from 0 to 60 .




Looking at the graph number of hidden units vs Training time . At the hidden unit 16 the the training time peaks approximately to 30 seconds . This because the more hidden layer imply the more number of learning parameters which is high computational cost. However after 16 hidden units it slightly decrease .


Test Accuracy vs Regularization λ for Different Hidden Units
Test Accuracy (%)
Regularization (λ)
Number of Hidden Units
4
8
12
16
20
100
95
90
85
80
75
70
65
60
55
50
45
40
0
10
20
30
40
50
60


Training Accuracy vs Regularization λ for Different Hidden Units
Training Accuracy (%)
Regularization (λ)
Number of Hidden Units
4
8
12
16
20
100
95
90
85
80
75
70
65
60
55
50
45
40
0
10
20
30
40
50
60




Looking at the three graph Regularization vs Training, testing and Validation  Accuracy it shows the model with 16 and 20 hidden units shows the high accuracy with different values of lambda  .  However the model with fewer hidden units <=8 are more sensitive to the choice of lambda as it fluctuate .

## Talking about Overfitting and underfitting

## Underfitting

From the 3 graph we can see that with 4 hidden layer the training , validation are relatively low regardless of the lambda value . This indicates the model is underfit with 4 hidden units .

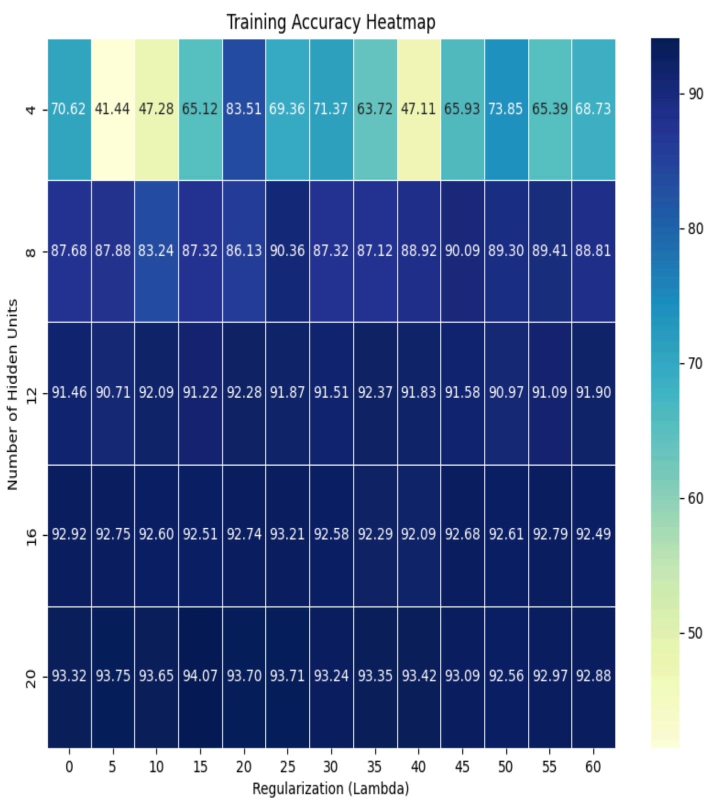
## Overfitting

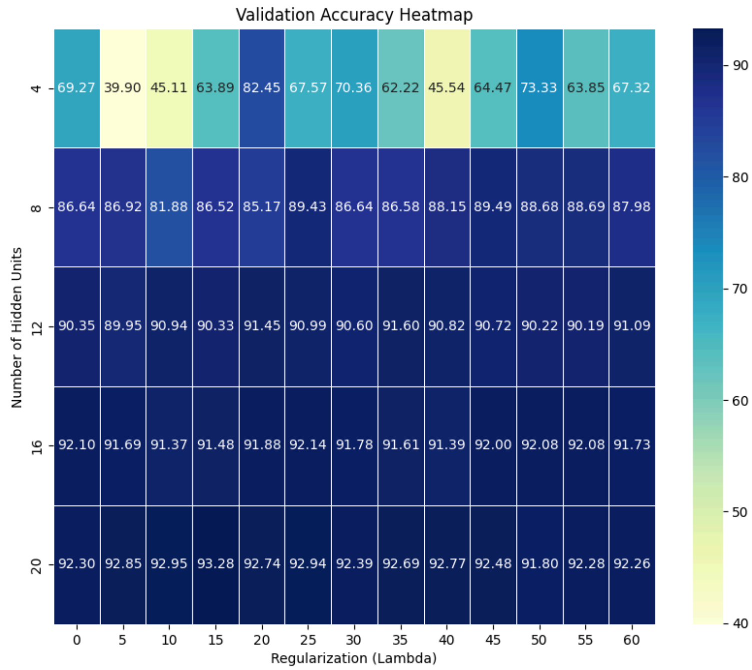
In the Training Accuracy vs Regularization grap as the number of hidden units increase in and when lambda is small it shows very high training accuracy above 90%. This show the model gets memorize the training data .
While comparing with the Validation vs Regularization the validation accuracy start diverge from the training accuracy when lambda is low. This shows the overfitting .

Specifically, for 16 or 20 hidden units with low λ values, the training accuracy remains high, while validation accuracy does not show as much improvement, indicating that the model might be overfitting to the training data.

**Now choosing optimal Hyper-Parameter**

**For choosing hyper parameter i like to include some pictures .**






**From the two heatmaps we can easily choose the hyper parameters .**

**Regularization**
**We can use this to prevent overfitting .**

Here when the regularization set to be too low the model can easily overfit if there are many hidden units . In our project where with the hidden unit of 20 and lambda =0  the training set achieved high accuracy whereas the validation set did not show similar result . This tells the  model is memorized the training data .

The graphs showed that λ values between 15 and 25 worked quite well, especially when the model had a larger number of hidden units. This balanced the need for complexity without overfitting the training data, leading to better generalization.

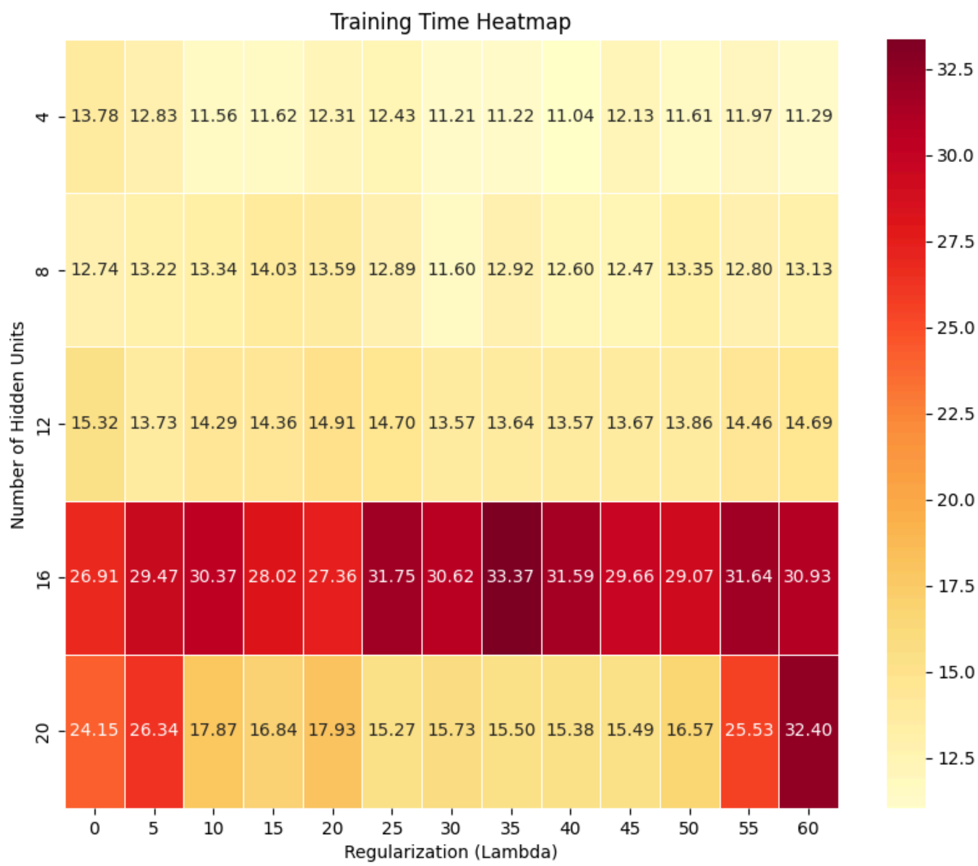
**Number of Hidden Units**

When the model had only 4 hidden units, it struggled to capture the complexity of the data, resulting in both lower training and validation accuracies across the board, which is typical of underfitting.

**Optimal Range**

Between 12 and 20 hidden units seemed to offer the best performance, providing the right balance between learning capacity and avoiding overfitting. For example, using 12 hidden units with λ around 15 or 20 produced consistently high accuracy for both training and validation, meaning the model had enough complexity to learn effectively without overfitting.

**Training Time :**




It was clear that the time taken to train with 16 hidden units spiked significantly compared to 12 hidden units, but with minimal gains in accuracy. For this reason, choosing 12 hidden units with a moderate λ seemed to be the sweet spot—delivering good performance while keeping training times under control.

So here is how i choose the optimal choice of hyper parameter

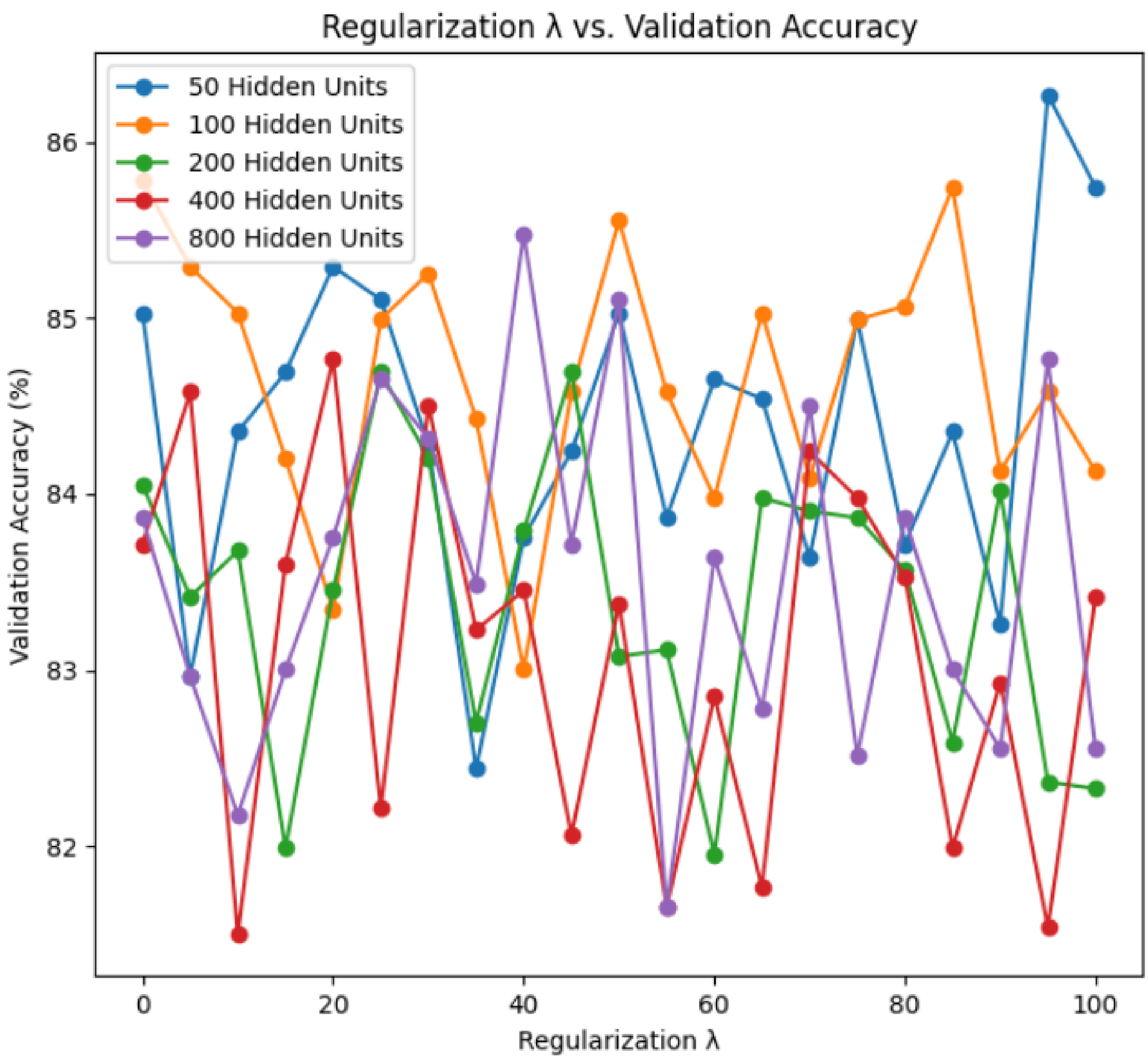
- Underfitting occurs when the model is too simple to capture the patterns in the data, which can happen with too few hidden units or excessive regularization.
- Overfitting happens when the model becomes overly complex, often due to too many hidden units and insufficient regularization.

**Optimal Choices for Hyper-Parameters:**

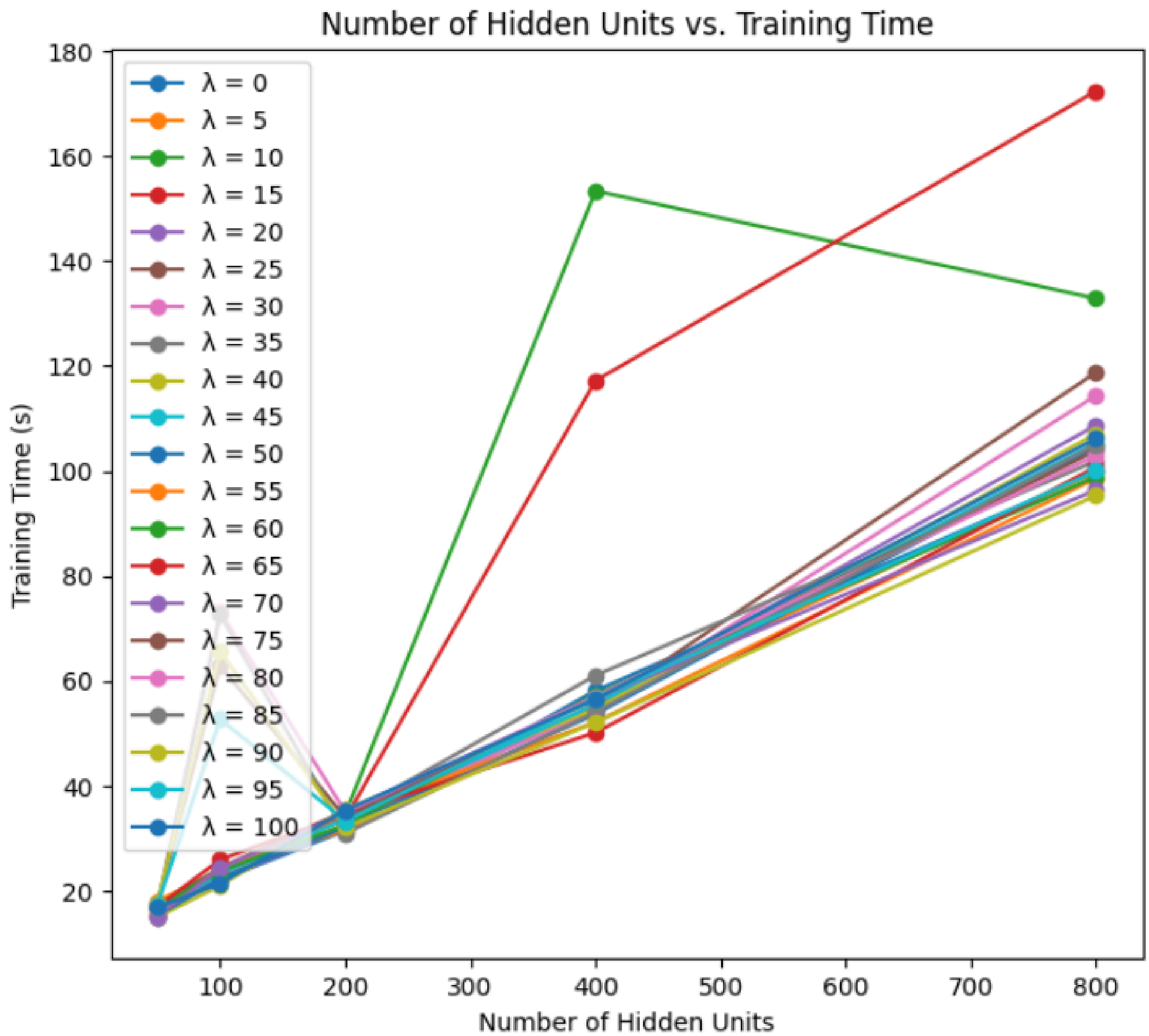
- Number of Hidden Units: Between 12 and 20 hidden units worked well, offering a good trade-off between learning capacity and generalization.
- Regularization Term : Values between 15 and 25 were optimal, providing sufficient regularization to prevent overfitting while still allowing the model to learn effectively.

This combination of hyper-parameters allowed us to achieve high accuracy without the drawbacks of either overfitting or underfitting, and did so in a reasonable amount of training time.

**For CELAB**


Regularization λ vs. Validation Accuracy
50 Hidden Units
100 Hidden Units
200 Hidden Units
400 Hidden Units
800 Hidden Units
Validation Accuracy (%)
Regularization λ
86
85
84
83
82
0
20
40
60
80
100

Training Accuracy vs. Regularization λ
Training Accuracy (%)
Regularization λ
86.5
86.0
85.5
85.0
84.5
84.0
83.5
83.0
0
20
40
60
80
100
50 Hidden Units
100 Hidden Units
200 Hidden Units
400 Hidden Units
800 Hidden Units




**1. Regularization λ vs. Validation Accuracy**

- Observation: The validation accuracy generally fluctuates significantly as λ changes.
- For lower values of hidden units (e.g., 50 and 100), the performance appears to be more stable.
- Higher hidden units (400, 800) show more fluctuations, which suggests that as model complexity increases, it becomes more sensitive to the regularization parameter.

#### 2. Number of Hidden Units vs. Training Time

- Observation: Training time increases considerably as the number of hidden units increases.
- With lower values of λ (e.g., λ = 0, 5, 10), the increase in training time is more gradual and stable.
- Higher λ values seem to lead to significant variations in training time, especially for larger numbers of hidden units.
- As the model complexity grows (more hidden units), the training time is impacted significantly, especially with higher regularization.

#### 3. Training Accuracy vs. Regularization λ

- Observation: The training accuracy fluctuates as λ changes.
- Lower hidden units seem to have more stable training accuracy across different λ values.
- The variation for higher hidden units (e.g., 800) suggests that the regularization significantly affects the model's ability to overfit or underfit the training data.

### Choosing Best hyper parameter

### Regularization:

- The best regularization parameter (λ) varies depending on the number of hidden units. It's clear that larger hidden units lead to more fluctuations with λ changes.
- For lower hidden units (50, 100), the validation accuracy is relatively stable, and λ can be chosen within a moderate range (e.g., 20 to 40).

### Hidden Units:

- Increasing the number of hidden units increases the training time significantly and makes the model more sensitive to λ.

- A moderate number of hidden units (100 to 200) seems to offer a good trade-off between training time, accuracy, and stability, while very high numbers (800) show increased training time without consistent improvements.

**Accuracy of classification method on the handwritten digits test data**

**Using NN**

Best Model Result:

Number of Hidden Units: 20

Regularization λ: 10

Validation Accuracy: 93.66%

Test Accuracy: 93.66%

**Using CNN**

Test Accuracy after 1 iteration: 11.0%

est Accuracy after 100 iterations: 74.2%

Test Accuracy after 1000 iterations: 93.2%

Test Accuracy after 10000 iterations: 98.6%

**The NN model achieved a final test accuracy of 93.66% with 20 hidden units and λ = 10.**

**The CNN model, on the other hand, improved significantly over multiple iterations:**

- **After 10000 iterations, it achieved a test accuracy of 98.6%.**

## Accuracy of classification method on the CelebA data set:

**Using Neural Network (NN)**

- **Best Model Result:**
  - **Number of Hidden Units: 50**
  - **Regularization λ: 95**
  - **Validation Accuracy: 86.27%**
  - **Test Accuracy: 86.75%**
  - **Training Time: 17.39 seconds**

**Using Deep Neural Network (DeepNN) with 1 Hidden Layer**

- **Training Progress:**

**Epoch 1: Loss = 36.7876, Validation Accuracy = 62.55%**

**Epoch 11: Loss = 10.0918, Validation Accuracy = 74.97%**

**Epoch 21: Loss = 6.9715, Validation Accuracy = 77.90%**

**Epoch 31: Loss = 5.4652, Validation Accuracy = 79.77%**

**Epoch 41: Loss = 4.5026, Validation Accuracy = 80.49%**

**Epoch 51: Loss = 3.8709, Validation Accuracy = 81.31%**

**Epoch 61: Loss = 3.4192, Validation Accuracy = 81.31%**

**Epoch 71: Loss = 3.0578, Validation Accuracy = 81.46%**

**Epoch 81: Loss = 2.6809, Validation Accuracy = 81.73%**

**Epoch 91: Loss = 2.3734, Validation Accuracy = 82.25%**

- **Test Accuracy: 82.59%**

**Summary**

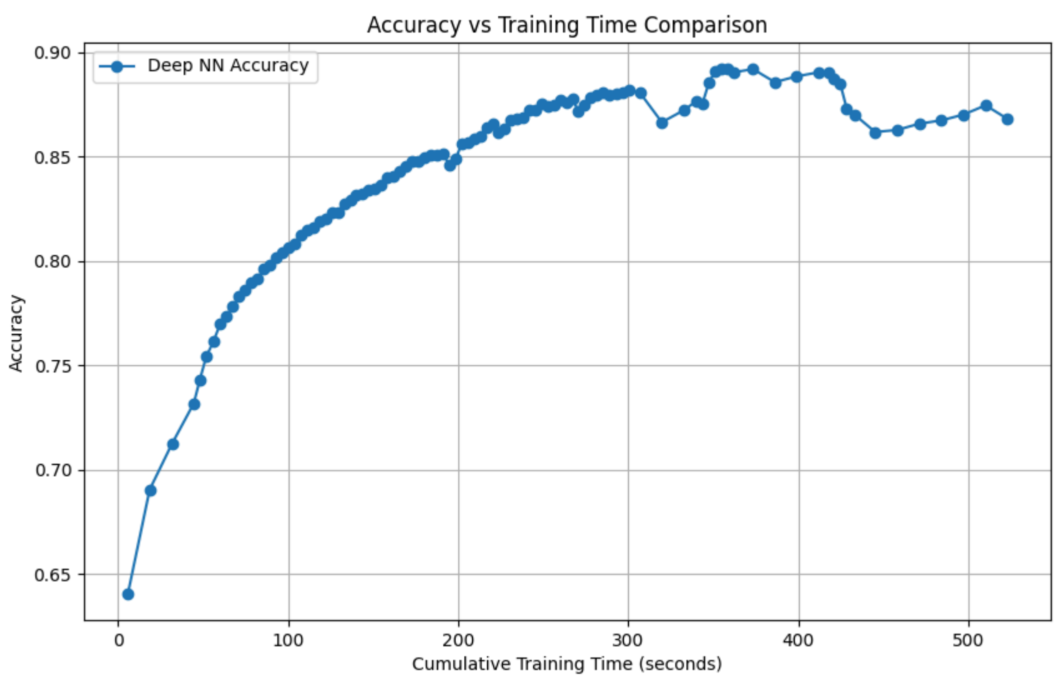
- **The Neural Network (NN) with 50 hidden units and λ = 95 performed better, achieving a Test Accuracy of 86.75%.**

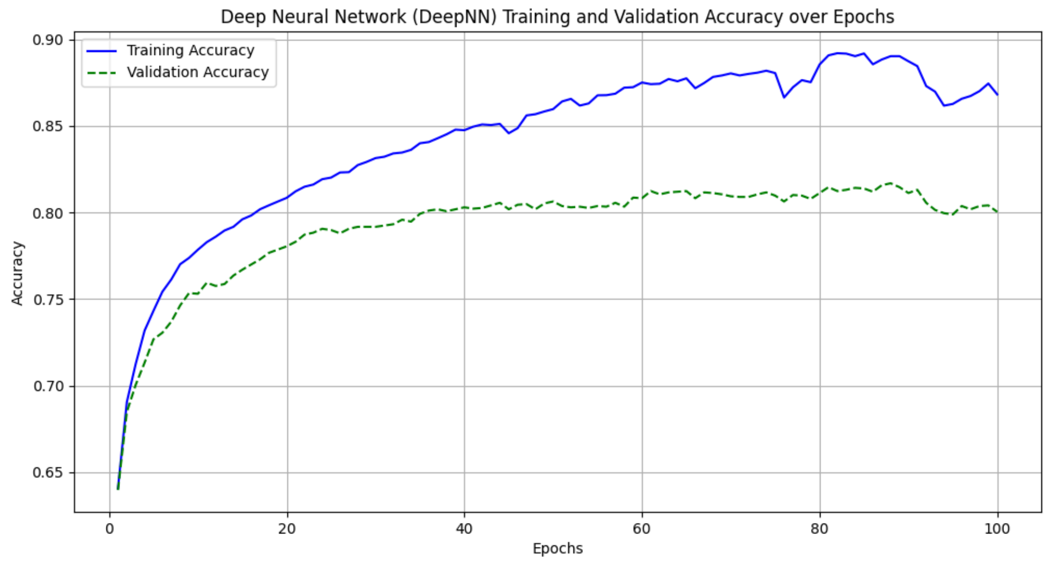
- **The Deep Neural Network (DeepNN) with 1 hidden layer achieved a Test Accuracy of 82.59% after 100 epochs.**

**The NN model outperformed the DeepNN model in this case, both in validation and test accuracy.**

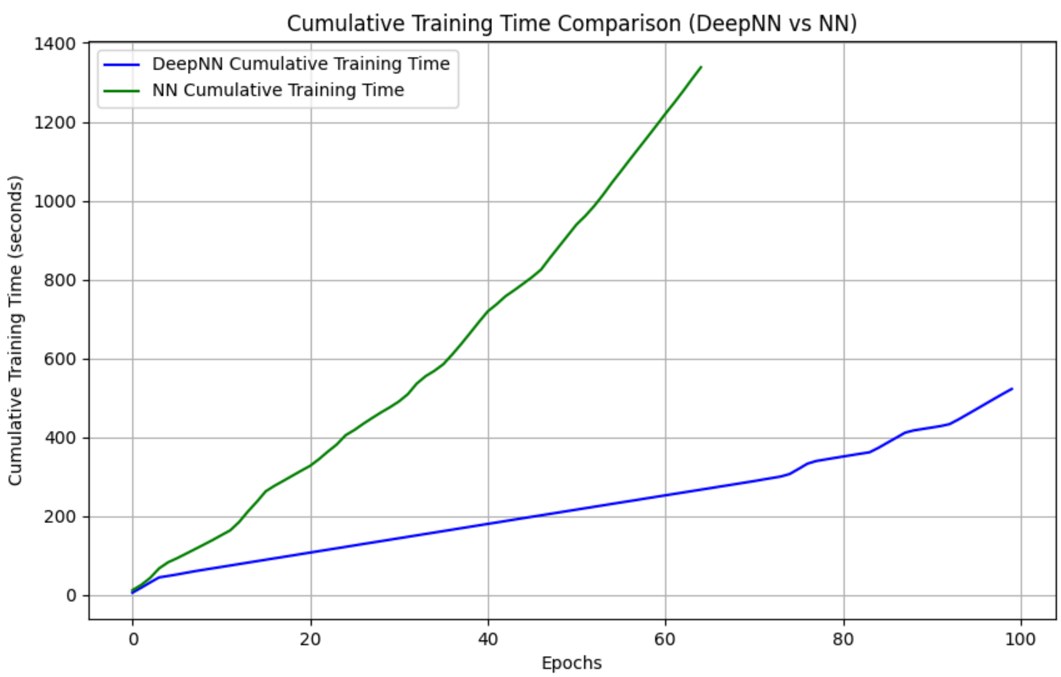
**—--------------------------------------------------------------------------------**

## Comparison of your neural network with a deep neural network (using TensorFlow) in terms of accuracy and training time: 8 points

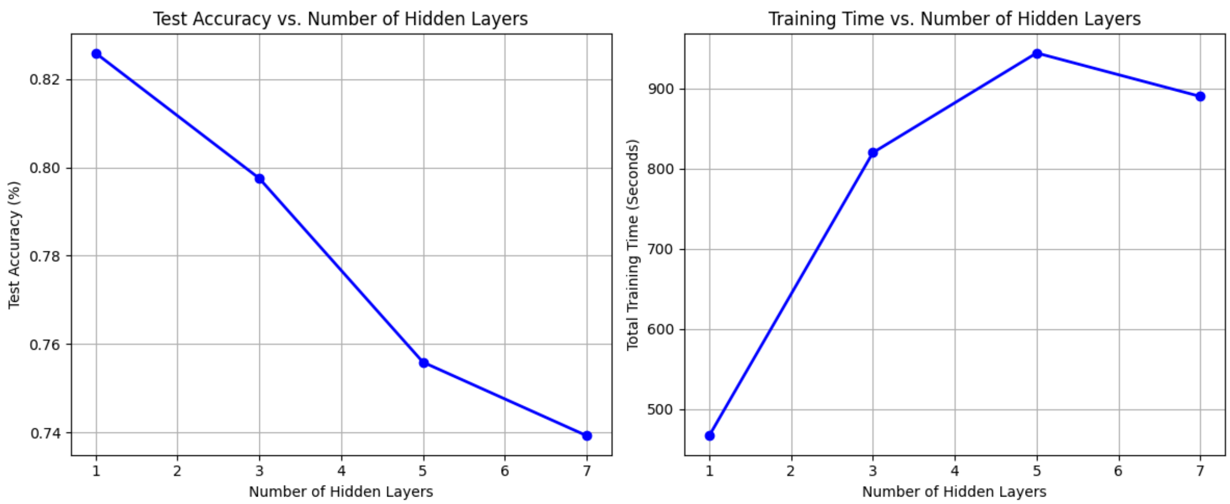




Deep Neural Network (DeepNN) Training and Validation Accuracy over Epochs
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.90
0.85
0.80
0.75
0.70
0.65
0
20
40
60
80
100

Cumulative Training Time Comparison (DeepNN vs NN)
DeepNN Cumulative Training Time
NN Cumulative Training Time
Cumulative Training Time (seconds)
Epochs
1400
1200
1000
800
600
400
200
0
0
20
40
60
80
100




**This graph show that a simpler model with fewer layers worked better in terms of both accuracy and efficiency.**

**Neural Network (NN):**

- **Optimal Hyperparameters:**
    - **Hidden Units: 50**
    - **Regularization (λ): 95**
- **Accuracy:**
    - **Number of Hidden Units: 50**
    - **Regularization λ: 95**
    - **Validation Accuracy: 86.27%**
    - **Test Accuracy: 86.75%**
    - **Training Time: 17.39 seconds**

**Deep Neural Network (DeepNN):**

- **The DeepNN model was trained with multiple hidden layers (1, 3, 5, and 7).**
- **Accuracy for Different Hidden Layers:**

- **1 Hidden Layer:**
    - **Test Accuracy: 82.59%**
    - **Training Time (per epoch): 4.62 seconds**
- **3 Hidden Layers:**
    - **Test Accuracy: 79.75%**
    - **Total Training Time: 100.52 seconds**
- **5 Hidden Layers:**
    - **Test Accuracy: 75.59%**
    - **Total Training Time: 136.13 seconds**
- **7 Hidden Layers:**
    - **Test Accuracy: 73.21%**
    - **Total Training Time: 250.32 seconds**

**Accuracy:**

The NN model with 50 hidden units outperformed the DeepNN models across different configurations.

The best test accuracy for the NN was 85.05%, while the DeepNN achieved a highest test accuracy of 82.59% for the configuration with 1 hidden layer.

As the number of hidden layers increased for the DeepNN, the accuracy tended to decrease, potentially due to overfitting or insufficient training time for more complex architectures.

**Training Time:**

The NN model showed better training efficiency, requiring 114.62 seconds to achieve optimal accuracy.

In contrast, the DeepNN model's training time increased significantly with more hidden layers:

3 hidden layers: 100.52 seconds

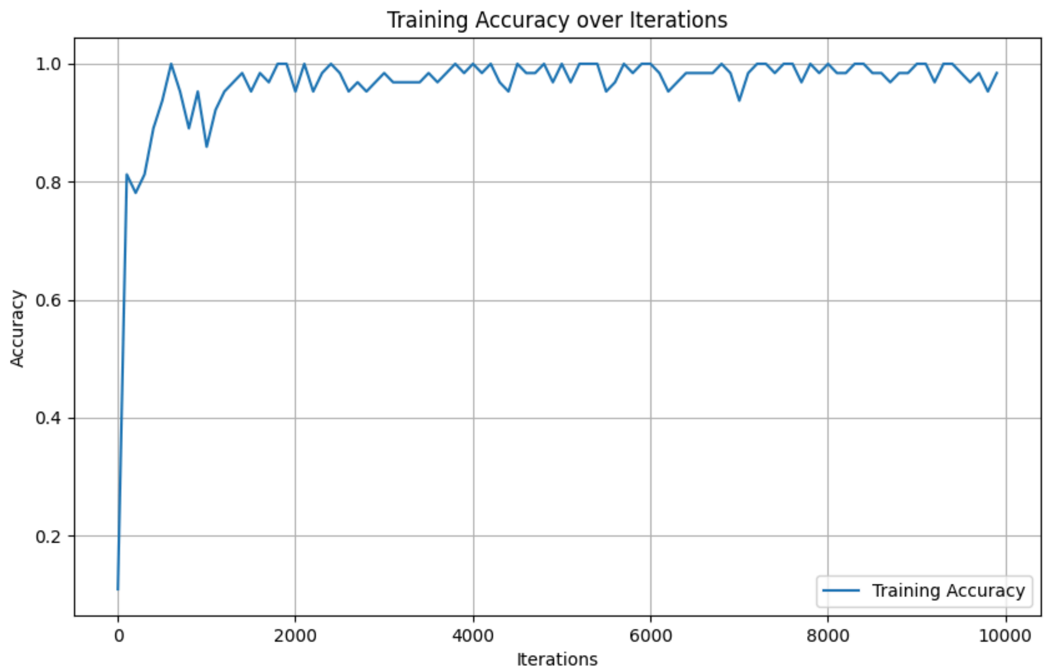
5 hidden layers: 136.13 seconds

7 hidden layers: 250.32 seconds

The training time for DeepNN increased linearly with the number of layers, whereas the accuracy did not improve proportionally.

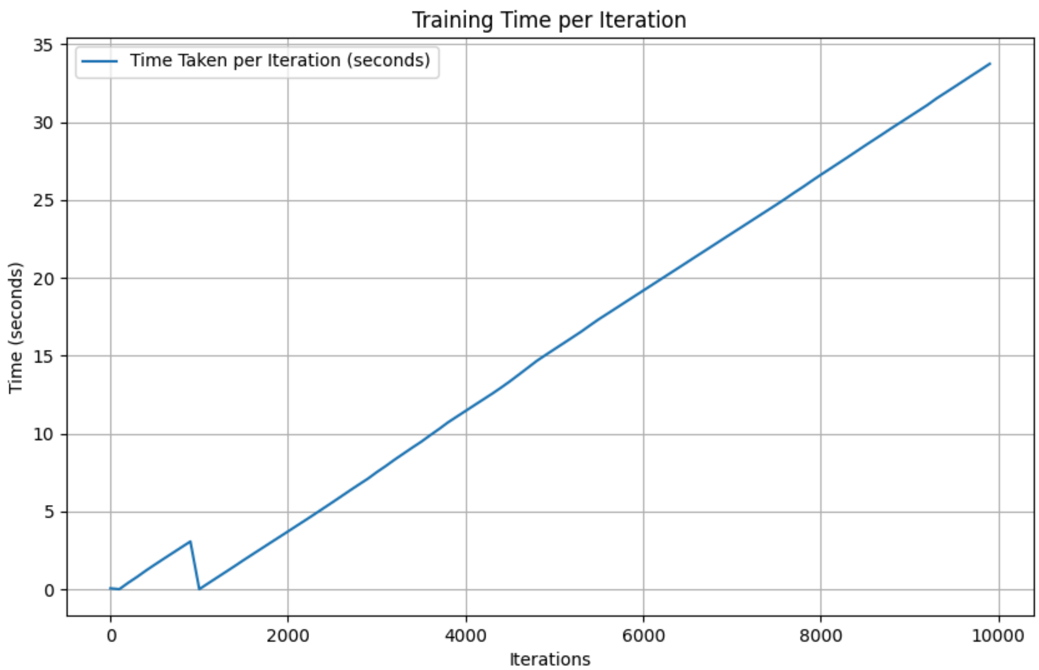
**The (NN) performed better in terms of both accuracy and computational efficiency compared to the deep neural network (DeepNN) with multiple layers .**

**Report the results from convolutional neural network in terms of accuracy and training time. 20 extra points**

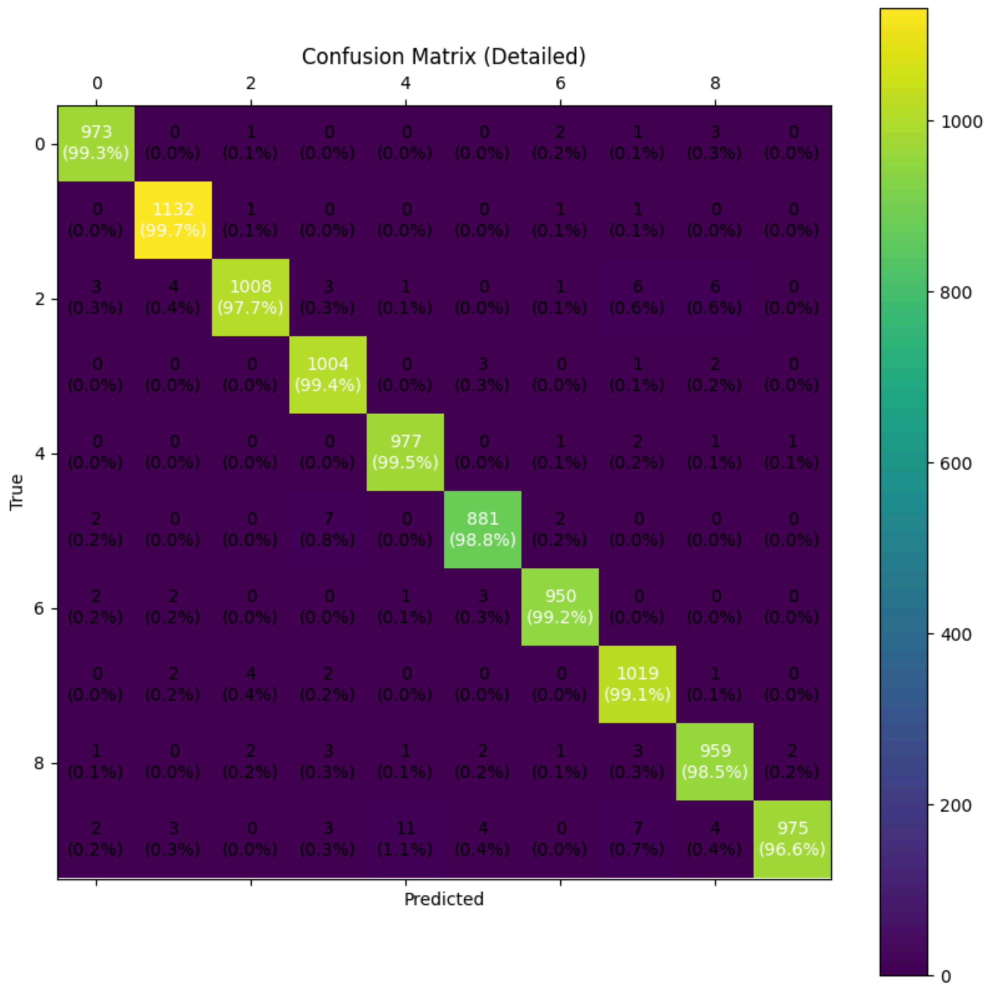


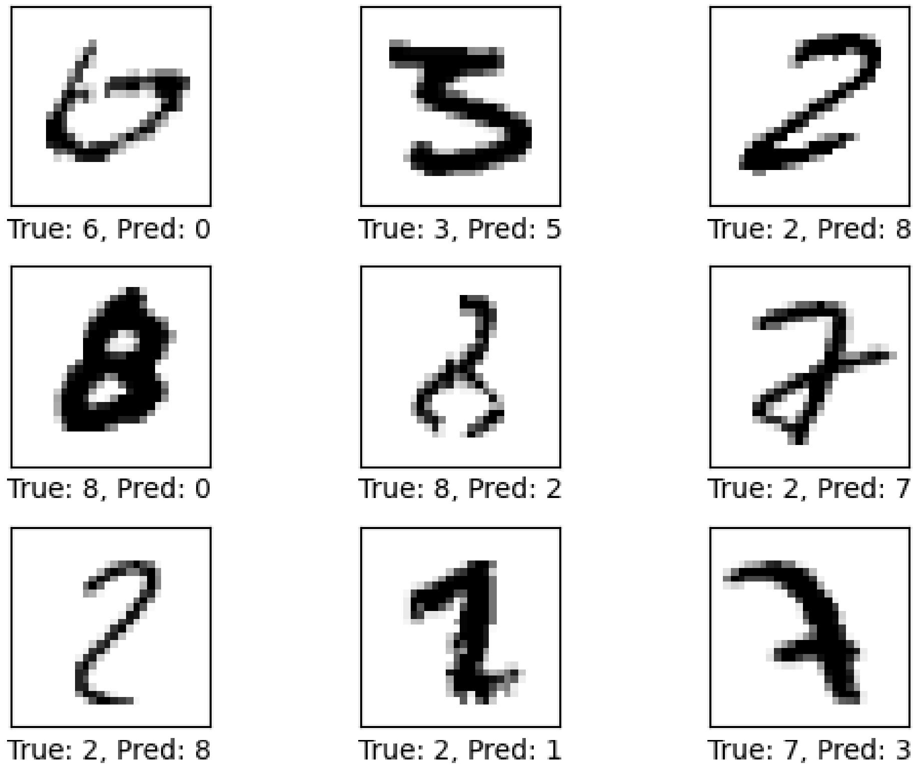

**The first graph shows the Training Accuracy over the course of the training iterations.**

**The accuracy improves rapidly at the beginning and then gradually stabilizes around 99-100%, indicating that the model has learned effectively from the training data.**




**The second graph illustrates the Training Time taken for each iteration.**
**The graph appears to have a linear upward trend, showing the cumulative increase in training time over iterations.**


Confusion Matrix (Detailed)
Predicted
True
0 2 4 6 8
0 2 4 6 8
0 200 400 600 800 1000
973 (99.3%) 0 (0.0%) 1 (0.1%) 0 (0.0%) 0 (0.0%) 0 (0.0%) 2 (0.2%) 1 (0.1%) 3 (0.3%) 0 (0.0%)
0 (0.0%) 1132 (99.7%) 1 (0.1%) 0 (0.0%) 0 (0.0%) 0 (0.0%) 1 (0.1%) 1 (0.1%) 0 (0.0%) 0 (0.0%)
3 (0.3%) 4 (0.4%) 1008 (97.7%) 3 (0.3%) 1 (0.1%) 0 (0.0%) 1 (0.1%) 6 (0.6%) 6 (0.6%) 0 (0.0%)
0 (0.0%) 0 (0.0%) 0 (0.0%) 1004 (99.4%) 0 (0.0%) 3 (0.3%) 0 (0.0%) 1 (0.1%) 2 (0.2%) 0 (0.0%)
0 (0.0%) 0 (0.0%) 0 (0.0%) 0 (0.0%) 977 (99.5%) 0 (0.0%) 1 (0.1%) 2 (0.2%) 1 (0.1%) 1 (0.1%)
2 (0.2%) 0 (0.0%) 0 (0.0%) 7 (0.8%) 0 (0.0%) 881 (98.8%) 2 (0.2%) 0 (0.0%) 0 (0.0%) 0 (0.0%)
2 (0.2%) 2 (0.2%) 0 (0.0%) 0 (0.0%) 1 (0.1%) 3 (0.3%) 950 (99.2%) 0 (0.0%) 0 (0.0%) 0 (0.0%)
0 (0.0%) 2 (0.2%) 4 (0.4%) 2 (0.2%) 0 (0.0%) 0 (0.0%) 0 (0.0%) 1019 (99.1%) 1 (0.1%) 0 (0.0%)
1 (0.1%) 0 (0.0%) 2 (0.2%) 3 (0.3%) 1 (0.1%) 2 (0.2%) 1 (0.1%) 3 (0.3%) 959 (98.5%) 2 (0.2%)
2 (0.2%) 3 (0.3%) 0 (0.0%) 3 (0.3%) 11 (1.1%) 4 (0.4%) 0 (0.0%) 7 (0.7%) 4 (0.4%) 975 (96.6%)


True: 6, Pred: 0
True: 3, Pred: 5
True: 2, Pred: 8
True: 8, Pred: 0
True: 8, Pred: 2
True: 2, Pred: 7
True: 2, Pred: 8
True: 2, Pred: 1
True: 7, Pred: 3