COMP9444 Assignment 1

Charles Wyatt z5194905

Contents

[Part 1 2](#_Toc86157974)

[Question 1 2](#_Toc86157975)

[Confusion Matrix and Final Accuracy of NetLin 2](#_Toc86157976)

[Question 2 2](#_Toc86157977)

[Full network – 2](#_Toc86157978)

[Confusion Matrix and Accuracy of full network with 250 hidden nodes 2](#_Toc86157979)

[Question 3 3](#_Toc86157980)

[Confusion Matrix and Accuracy of convolution network 3](#_Toc86157981)

[Question 4 3](#_Toc86157982)

[Part 2 4](#_Toc86157983)

[Question 2 4](#_Toc86157984)

[Question 4 5](#_Toc86157985)

[Question 5 5](#_Toc86157986)

[Question 7 6](#_Toc86157987)

[Part 3 7](#_Toc86157988)

[Part 4 7](#_Toc86157989)

[Question 1 7](#_Toc86157990)

[Question 2 8](#_Toc86157991)

[Question 3 8](#_Toc86157992)

[Question 4 8](#_Toc86157993)

[Question 5 9](#_Toc86157994)

[Question 6 9](#_Toc86157995)

[Appendices 10](#_Toc86157996)

[Appendix A 10](#_Toc86157997)

[Appendix B 28](#_Toc86157998)

# Part 1

## Question 1

### Confusion Matrix and Final Accuracy of NetLin

Calendar

Description automatically generated

## Question 2

### Full network –

|  |  |
| --- | --- |
| **Number of Hidden nodes** | **Accuracy** |
| 20 | 73% |
| 30 | 76% |
| 40 | 79% |
| 50 | 80% |
| 100 | 83% |
| 150 | 84% |
| 200 | 84% |
| 250 | 84.97% |
| 300 | 85.15% |
| 350 | 85.4% |
| 400 | 85.51% |

250 is chosen as the best value since greater values showed no material improvement and took longer to train.

### Confusion Matrix and Accuracy of full network with 250 hidden nodes

A picture containing text, keyboard, electronics, scoreboard

Description automatically generated

## Question 3

### Confusion Matrix and Accuracy of convolution network

A picture containing text, keyboard, electronics

Description automatically generated

## Question 4

1. Clearly, the convolution network is the most accurate model at a 95% accuracy. Since the convolution network has more independent parameters it is a much more flexible, and so can fit the complexities of the image. The full network is the second most accurate network at 85% and its predictive power becomes stronger by increasing the number of hidden nodes. However, the accuracy of the full network plateaus at 250. Lastly, the worst model was the linear network with an accuracy of only 70%
2. In this case, the number of independent parameters scales with the predictive capabilities of the model.

*Note, all independent parameters were calculated through the torchsummary function.*

|  |  |
| --- | --- |
| **Model** | **Independent Parameters** |
| Linear Network | 7 850 |
| Full Network | 198 760 |
| Convolution Network | 313 108 |

1. See the following table for each model and use the examples of the characters as a reference

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image 0** | **Image 1** | **Image 2** | **Image 3** | **Image 4** | **Image 5** | **Image 6** | **Image 7** | **Image 8** | **Image 9** |
| A picture containing text, clock  Description automatically generated | Qr code  Description automatically generated | A picture containing text, clock  Description automatically generated | Qr code  Description automatically generated | A picture containing text, clipart  Description automatically generated | A picture containing text, clock  Description automatically generated | A screenshot of a computer  Description automatically generated with low confidence | A screenshot of a computer  Description automatically generated with low confidence | A screenshot of a computer  Description automatically generated with low confidence | A picture containing qr code  Description automatically generated |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Confusion Matrix** | **Most likely mistaken** | **Reasons** |
| Linear Network | Calendar  Description automatically generated | * Images 2 and 1. * Images 4 and 2. * Images 5 and 2 * Images 2 and 8 * Images 2 and 9 | * Image 2 and image 1 were the most mistaken, perhaps due to the similarities in the left-hand side of the image. Both are quite thin characters with lots of “curls” and so appear similar to a simple linear network |
| Full Network | A picture containing text, keyboard, electronics, scoreboard  Description automatically generated | * Image 2 and image 5 * Image 2 and image 9 * Image 7 and image 0 * Image 1 and image 6 | Image 2 and 5 were the most mistaken. The right side of the characters appear very similar, with a long, thin vertical line. |
| Convolution Network |  | * Image 2 and image 5 | There were not many mistakes here. Again, Image 5 and 2 were the most mistaken. Perhaps for the same reasons as above |

## Part 2

### Question 2

* The initial weights used were
* The learning rate used as 0.001.

The first model that was successfully trained using these parameters was the model with 20 hidden nodes. The results from the tested models can be shown below.

|  |  |  |
| --- | --- | --- |
| **Hidden nodes** | **Accuracy** | **Epochs Needed** |
| 1 | ~ 56% | 23 100 |
| 5 | ~69% | 21 200 |
| 10 | ~ 87% | 41 900 |
| 15 | ~ 91% | 110 100 |
| 20 | 100% | 90 000 |

**Plot of output two layer 20 hidden nodes.**

Chart

Description automatically generated

**Total independent Parameters:**

501

### Question 4

* The initial weights used were
* The learning rate used as 0.001.

The first model that was successfully trained using these parameters was the model with 30 hidden nodes. The results from the tested models can be shown below.

|  |  |  |
| --- | --- | --- |
| **Hidden nodes** | **Accuracy** | **Epochs Needed** |
| 10 | ~80% | 200 000 |
| 20 | ~90% | 200 000 |
| 30 | 100% | 96 000 |

**Output Plot**

Chart

Description automatically generated

*See Appendix A for the plots of all the hidden units*

**Total Independent Parameters:**

1981

## Question 5

|  |  |  |
| --- | --- | --- |
| **Number of nodes** | **Accuracy** | **Epochs Needed** |
| 10 | 89% | 25 000 |
| 15 | 96% | 42 500 |
| 17 | 99% | 53 700 |
| 20 | 100% | 20 100 |

**20 node output plot**

Chart

Description automatically generated

*See Appendix B for the plots of all the Hidden Nodes*

**Total Independent Parameters**

563

### Question 7

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Hidden Nodes** | **Independent Parameters** | **Epochs** |
| Full2Net | 20 | 501 | 90 000 |
| Full3Net | 30 | 1981 | 96 000 |
| DenseNet | 20 | 563 | 20 100 |

Interestingly, the Full2Net required less hidden nodes than the Full3Net model, perhaps the added complexity led to some overfitting by the Full3Net model. DenseNet was clearly the best model with the least number of epochs and a similar number of independent parameters to the Full2Net.

1. The nodes in the first layer of both Full3Net and DenseNet are linear lines. For both networks, many of the first nodes are horizontal and vertical. This is because many dots from the same class lie on these vertical and horizontal lines such as the vertical line of red dots at 0 and the horizonal line of blue dots at 0. Therefore, these nodes will be able to classify these points easily.

Other lines in the first layer capture some of the diagonal elements of the fractal.

In the second layer the complexity of the boundary increases. For the dense net the complexity of the boundary increases more than the 3-layer full net since it includes the skip connections. The 3-layer full network boundary again becomes more complex at the 3rd hidden layer.

1. There were not obvious qualitative differences between the 3 output plots. However, the dense plot was able to recognize the vertical features of the red dots more effectively then the other two. The 3 layer has some diagonal areas which seem to account for the alternating blue / red pattern on the sides of the fractal, where the dense plot has instead made blue “pocket” areas as seen in the bottom right-hand corner.

# Part 3

Chart, scatter chart

Description automatically generated

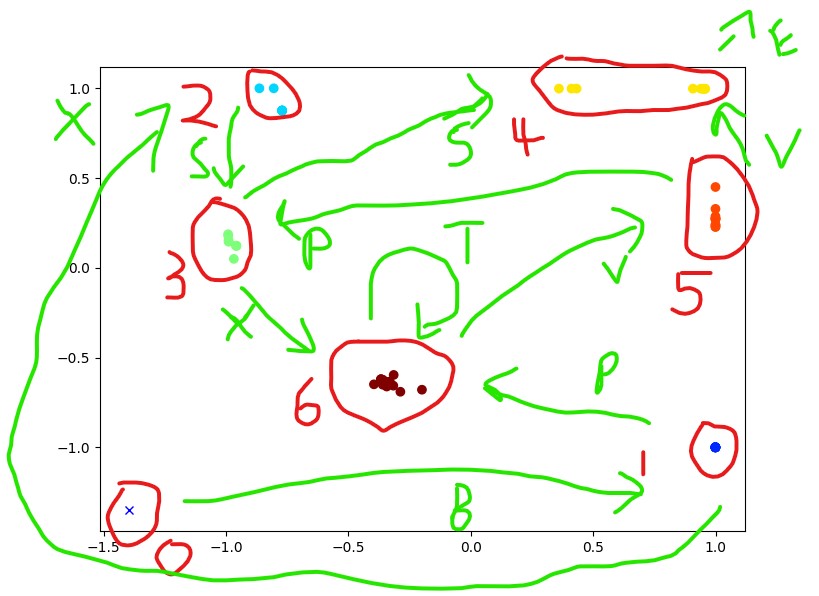
# Part 4

### Question 1

Key for the following diagram –

Red numbers = state

Green Letters = Symbol.



### Question 2

Diagram

Description automatically generated

### Question 3

There are some letters that the network can predict with certainty since there is a predictable pattern. These are represented by the “redder” dots in Q2. Therefore, the first A of each sequence are “redder” since they are deterministic and have a higher y-axis activation, as shown by the group in the top left-hand corner. All the B’s that occur after the first B are deterministic and are in the bottom right-hand corner since they most activate the hidden unit represented by the x axis.

The darker, blue dots represent the “uncertain” guesses. For example, consecutive A’s cannot continue indefinitely, but there is no deterministic way to determine how many A’s there will be. So, after each “A” guess in a row, the network reduces the probability that the next guess is an A and increases the probability that the next guess is a B

### Question 4

Chart, diagram

Description automatically generated

|  |  |
| --- | --- |
| Chart  Description automatically generated | Diagram  Description automatically generated |
| Chart, diagram  Description automatically generated with medium confidence | |

### Question 5

The 3 hidden units activations correspond to each letter as shown by the graphs above. By isolating a 2d perspective of the 3d image the following is clear –

There is not much x variation in A

There is not much z variation in C

There is not much y variation in B

Of these 3 letters, c has the least variation. C falls almost exactly on the z = 1 plane whereas A and B have slight deviations. This is an expected result C is the only class which is purely deterministic. There is some probability involved for A and B since it is uncertain when consecutive As will stop and start a new sequence of Bs. Therefore, the network can predict all the C’s, all of the B’s except for the first of a sequence, and the first A of the sequence.

Therefore, the hidden node activation of the x axis represents the likelihood of being an A

The hidden node activation of the y axis represents the likelihood of being a B

The hidden node activation of the z axis represents the likelihood of being a C.

### Question 6

How the model works –

1. Initalise model and weights
2. For each number in the sequence:
   1. First make gates based on the weights
   2. Use the sigmoid function to “forget” some context if it is below some threshold
   3. Transform input with sigmoid and tanh
   4. Use a sigmoid function to output result
   5. Update hidden units / gates
   6. Go back to step a.
3. Output result

By plotting the context nodes after every 1000 epochs we can see how the model is “learning”. Early in the training, the context nodes shift wildly since the network is exposed to new information and so may “forget” the required long-term context information. However, after about 20 000 epochs they begin to stabilize.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Epoch** | **Context Node 1** | **Context Node 2** | **Context Node 3** | **Context Node 4** |
| 0 | -0.5159 | 0.2825 | -0.3055 | -0.5741 |
| 5 000 | -1.3446 | -0.9050 | -3.0737 | -3.8454 |
| 10 000 | -0.6793 | -0.1669 | -1.2845 | -3.2066 |
| 15 000 | -1.1163 | -0.2603 | -2.4591 | -5.7861 |
| 20 000 | 1.1615 | -3.2945 | 0.3195 | 0.6405 |
| 25 000 | 0.9829 | -2.6177 | 0.3974 | 0.3096 |
| 30 000 | 0.9870 | -2.2228 | 0.9371 | -0.6356 |
| 35 000 | 0.7659 | -2.2290 | 0.8086 | -0.3102 |
| 40 000 | 0.9626 | -1.3299 | 0.8989 | -0.4284 |
| 45 000 | 0.9744 | -2.2622 | 1.0170 | -1.1428 |
| 50 000 | 0.9464 | -1.9169 | 0.9432 | -0.8253 |

Therefore, over time the LSTM can effectively have “long term memory” through these context units as they approach a value approximately [1, -2, 1, -1] which will lead to the highest accuracy in its predictions.

# Appendices

## Appendix A

Hidden unit plots for the 3-layer full network–

|  |  |  |
| --- | --- | --- |
| Node number | Hidden layer | Plot |
| 0 | 1 | Chart  Description automatically generated |
| 1 | 1 | A picture containing chart  Description automatically generated |
| 2 | 1 | Chart, scatter chart  Description automatically generated |
| 3 | 1 |  |
| 4 | 1 |  |
| 5 | 1 |  |
| 6 | 1 |  |
| 7 | 1 |  |
| 8 | 1 |  |
| 9 | 1 |  |
| 10 | 1 |  |
| 11 | 1 |  |
| 12 | 1 |  |
| 13 | 1 |  |
| 14 | 1 |  |
| 15 | 1 |  |
| 16 | 1 |  |
| 17 | 1 |  |
| 18 | 1 |  |
| 19 | 1 |  |
| 20 | 1 |  |
| 21 | 1 |  |
| 22 | 1 |  |
| 23 | 1 |  |
| 24 | 1 |  |
| 25 | 1 |  |
| 26 | 1 |  |
| 27 | 1 |  |
| 28 | 1 |  |
| 29 | 1 |  |
| 0 | 2 |  |
| 1 | 2 |  |
| 2 | 2 |  |
| 3 | 2 |  |
| 4 | 2 |  |
| 5 | 2 |  |
| 6 | 2 |  |
| 7 | 2 |  |
| 8 | 2 |  |
| 9 | 2 |  |
| 10 | 2 |  |
| 11 | 2 |  |
| 12 | 2 |  |
| 13 | 2 |  |
| 14 | 2 |  |
| 15 | 2 |  |
| 16 | 2 |  |
| 17 | 2 |  |
| 18 | 2 |  |
| 19 | 2 |  |
| 20 | 2 |  |
| 21 | 2 |  |
| 22 | 2 |  |
| 23 | 2 |  |
| 24 | 2 |  |
| 25 | 2 |  |
| 26 | 2 |  |
| 27 | 2 |  |
| 28 | 2 |  |
| 29 | 2 |  |
| 0 | 3 |  |
| 1 | 3 |  |
| 2 | 3 |  |
| 3 | 3 |  |
| 4 | 3 | 5 |
| 5 | 3 |  |
| 6 | 3 |  |
| 7 | 3 |  |
| 8 | 3 |  |
| 9 | 3 |  |
| 10 | 3 |  |
| 11 | 3 |  |
| 12 | 3 |  |
| 13 | 3 |  |
| 14 | 3 |  |
| 15 | 3 |  |
| 16 | 3 |  |
| 17 | 3 |  |
| 18 | 3 |  |
| 19 | 3 |  |
| 20 | 3 |  |
| 21 | 3 |  |
| 22 | 3 |  |
| 23 | 3 |  |
| 24 | 3 |  |
| 25 | 3 |  |
| 26 | 3 |  |
| 27 | 3 |  |
| 28 | 3 |  |
| 29 | 3 |  |

## Appendix B

Hidden unit plots for the dense network

|  |  |  |
| --- | --- | --- |
| Node number | Hidden layer | Plot |
| 0 | 1 |  |
| 1 | 1 |  |
| 2 | 1 |  |
| 3 | 1 |  |
| 4 | 1 |  |
| 5 | 1 |  |
| 6 | 1 |  |
| 7 | 1 |  |
| 8 | 1 |  |
| 9 | 1 |  |
| 10 | 1 |  |
| 11 | 1 |  |
| 12 | 1 |  |
| 13 | 1 |  |
| 14 | 1 |  |
| 15 | 1 |  |
| 16 | 1 |  |
| 17 | 1 |  |
| 18 | 1 |  |
| 19 | 1 |  |
| 0 | 2 |  |
| 1 | 2 |  |
| 2 | 2 |  |
| 3 | 2 |  |
| 4 | 2 |  |
| 5 | 2 |  |
| 6 | 2 |  |
| 7 | 2 |  |
| 8 | 2 |  |
| 9 | 2 |  |
| 10 | 2 |  |
| 11 | 2 |  |
| 12 | 2 |  |
| 13 | 2 |  |
| 14 | 2 |  |
| 15 | 2 |  |
| 16 | 2 |  |
| 17 | 2 |  |
| 18 | 2 |  |
| 19 | 2 |  |