

NEURAL NETWORK DEEP LEARNING

ICP 3

700758238

AKHIL KASANAGOTTU

GitHub:

Repository URL for the source code: https://github.com/axk82380/NNPL/tree/main/ICP_4

Video Link:

<https://drive.google.com/file/d/1ez2OFk3UxUAEP4xkR3sLdz9WaLqC0qPe/view?usp=sharing>

1. Use the use case in the class:

- a. Add more Dense layers to the existing code and check how the accuracy changes.
- b. Change the data source to Breast Cancer dataset * available in the source code folder and make required changes. Report accuracy of the model.
- c. Normalize the data before feeding the data to the model and check how the normalization changes your accuracy (code given below).
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
- d. Breast Cancer dataset is designated to predict if a patient has Malignant (M) or Benign = B
Cancer

```
Epoch 1/100
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, pr
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
18/18 ----- 2s 4ms/step - acc: 0.3335 - loss: 8.1713
Epoch 2/100
18/18 ----- 0s 4ms/step - acc: 0.3672 - loss: 3.5076
Epoch 3/100
18/18 ----- 0s 4ms/step - acc: 0.4749 - loss: 2.2717
Epoch 4/100
18/18 ----- 0s 4ms/step - acc: 0.6272 - loss: 1.2742
Epoch 5/100
18/18 ----- 0s 5ms/step - acc: 0.6518 - loss: 0.9166
Epoch 6/100
18/18 ----- 0s 3ms/step - acc: 0.6513 - loss: 0.8067
Epoch 7/100
18/18 ----- 0s 4ms/step - acc: 0.6452 - loss: 0.7822
Epoch 8/100
18/18 ----- 0s 5ms/step - acc: 0.6825 - loss: 0.6901
Epoch 9/100
18/18 ----- 0s 6ms/step - acc: 0.6812 - loss: 0.6846
Epoch 10/100
18/18 ----- 0s 4ms/step - acc: 0.6912 - loss: 0.6786
Epoch 11/100
18/18 ----- 0s 4ms/step - acc: 0.6439 - loss: 0.6821
Epoch 12/100
18/18 ----- 0s 3ms/step - acc: 0.6353 - loss: 0.6790
Epoch 13/100
18/18 ----- 0s 4ms/step - acc: 0.6587 - loss: 0.6790
Epoch 14/100
18/18 ----- 0s 2ms/step - acc: 0.6671 - loss: 0.6745
Epoch 15/100
18/18 ----- 0s 3ms/step - acc: 0.6530 - loss: 0.6731
Epoch 16/100
```

```
Epoch 92/100
18/18 ----- 0s 3ms/step - acc: 0.6723 - loss: 0.6269
Epoch 93/100
18/18 ----- 0s 2ms/step - acc: 0.6756 - loss: 0.6297
Epoch 94/100
18/18 ----- 0s 4ms/step - acc: 0.6409 - loss: 0.6501
Epoch 95/100
18/18 ----- 0s 3ms/step - acc: 0.6352 - loss: 0.6545
Epoch 96/100
18/18 ----- 0s 4ms/step - acc: 0.6837 - loss: 0.6222
Epoch 97/100
18/18 ----- 0s 2ms/step - acc: 0.6396 - loss: 0.6520
Epoch 98/100
18/18 ----- 0s 4ms/step - acc: 0.6791 - loss: 0.6284
Epoch 99/100
18/18 ----- 0s 3ms/step - acc: 0.6661 - loss: 0.6410
Epoch 100/100
18/18 ----- 0s 2ms/step - acc: 0.6407 - loss: 0.6528
Model: "sequential_16"

```

Layer (type)	Output Shape	Param #
dense_42 (Dense)	(None, 20)	180
dense_43 (Dense)	(None, 4)	84
dense_44 (Dense)	(None, 1)	5

```

Total params: 809 (3.16 KB)
Trainable params: 269 (1.05 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 540 (2.11 KB)
None
6/6 ----- 0s 4ms/step - acc: 0.6580 - loss: 0.6528
[0.6727693676948547, 0.625]
```

✓ 13s [10] Epoch 1/100
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Se
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
14/14 ----- 1s 2ms/step - acc: 0.6087 - loss: 63.8767
Epoch 2/100
14/14 ----- 0s 2ms/step - acc: 0.7260 - loss: 11.0967
Epoch 3/100
14/14 ----- 0s 2ms/step - acc: 0.5885 - loss: 2.8690
Epoch 4/100
14/14 ----- 0s 2ms/step - acc: 0.8702 - loss: 0.8847
Epoch 5/100
14/14 ----- 0s 2ms/step - acc: 0.8911 - loss: 1.0527
Epoch 6/100
14/14 ----- 0s 3ms/step - acc: 0.8716 - loss: 0.6283
Epoch 7/100
14/14 ----- 0s 7ms/step - acc: 0.9022 - loss: 0.6254
Epoch 8/100
14/14 ----- 0s 5ms/step - acc: 0.8462 - loss: 0.8862
Epoch 9/100
14/14 ----- 0s 6ms/step - acc: 0.8630 - loss: 0.8260
Epoch 10/100
14/14 ----- 0s 6ms/step - acc: 0.8839 - loss: 0.6950
Epoch 11/100
14/14 ----- 0s 7ms/step - acc: 0.8655 - loss: 0.8427
Epoch 12/100
14/14 ----- 0s 4ms/step - acc: 0.8900 - loss: 0.6507
Epoch 13/100
14/14 ----- 0s 4ms/step - acc: 0.9004 - loss: 0.4464
Epoch 14/100
14/14 ----- 0s 3ms/step - acc: 0.8622 - loss: 0.7535
Epoch 15/100
14/14 ----- 0s 10ms/step - acc: 0.8848 - loss: 0.6996
Epoch 16/100

✓ 13s [10] Epoch 91/100
14/14 ----- 0s 2ms/step - acc: 0.9215 - loss: 0.3542
Epoch 92/100
14/14 ----- 0s 3ms/step - acc: 0.9298 - loss: 0.2702
Epoch 93/100
14/14 ----- 0s 2ms/step - acc: 0.9334 - loss: 0.2907
Epoch 94/100
14/14 ----- 0s 3ms/step - acc: 0.9318 - loss: 0.2919
Epoch 95/100
14/14 ----- 0s 2ms/step - acc: 0.9252 - loss: 0.2938
Epoch 96/100
14/14 ----- 0s 2ms/step - acc: 0.9066 - loss: 0.3054
Epoch 97/100
14/14 ----- 0s 2ms/step - acc: 0.9178 - loss: 0.3415
Epoch 98/100
14/14 ----- 0s 2ms/step - acc: 0.9419 - loss: 0.2126
Epoch 99/100
14/14 ----- 0s 2ms/step - acc: 0.9273 - loss: 0.2905
Epoch 100/100
14/14 ----- 0s 2ms/step - acc: 0.8902 - loss: 0.3319
Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 20)	620
dense_4 (Dense)	(None, 1)	21

Total params: 1,925 (7.52 KB)
Trainable params: 641 (2.50 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 1,284 (5.02 KB)

None

5/5 ----- 0s 4ms/step - acc: 0.8366 - loss: 0.8101
[0.6811095476150513, 0.8601398468017578]

```

✓ 11s [14] Epoch 1/100
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. W
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
14/14 ----- 1s 3ms/step - acc: 0.6019 - loss: 115.2636
Epoch 2/100
14/14 ----- 0s 3ms/step - acc: 0.5998 - loss: 54.6094
Epoch 3/100
14/14 ----- 0s 4ms/step - acc: 0.1657 - loss: 15.1049
Epoch 4/100
14/14 ----- 0s 4ms/step - acc: 0.2944 - loss: 9.6360
Epoch 5/100
14/14 ----- 0s 3ms/step - acc: 0.3421 - loss: 5.3422
Epoch 6/100
14/14 ----- 0s 4ms/step - acc: 0.4081 - loss: 3.9531
Epoch 7/100
14/14 ----- 0s 6ms/step - acc: 0.4959 - loss: 3.3488
Epoch 8/100
14/14 ----- 0s 7ms/step - acc: 0.5505 - loss: 2.3070
Epoch 9/100
14/14 ----- 0s 7ms/step - acc: 0.5780 - loss: 2.3237
Epoch 10/100
14/14 ----- 0s 7ms/step - acc: 0.6458 - loss: 1.6994
Epoch 11/100
14/14 ----- 0s 7ms/step - acc: 0.5880 - loss: 1.4381
Epoch 12/100
14/14 ----- 0s 10ms/step - acc: 0.6418 - loss: 1.0489
Epoch 13/100
14/14 ----- 0s 3ms/step - acc: 0.6976 - loss: 0.7873
Epoch 14/100
14/14 ----- 0s 2ms/step - acc: 0.7269 - loss: 0.7139
Epoch 15/100
14/14 ----- 0s 2ms/step - acc: 0.8422 - loss: 0.5445
Epoch 16/100
14/14 ----- 0s 2ms/step - acc: 0.8282 - loss: 0.4505

```

```

✓ 11s [14] 14/14 ----- 0s 2ms/step - acc: 0.9093 - loss: 0.2814
Epoch 92/100
14/14 ----- 0s 2ms/step - acc: 0.9160 - loss: 0.2774
Epoch 93/100
14/14 ----- 0s 2ms/step - acc: 0.9151 - loss: 0.2480
Epoch 94/100
14/14 ----- 0s 2ms/step - acc: 0.9237 - loss: 0.2546
Epoch 95/100
14/14 ----- 0s 2ms/step - acc: 0.9007 - loss: 0.2967
Epoch 96/100
14/14 ----- 0s 2ms/step - acc: 0.9069 - loss: 0.2538
Epoch 97/100
14/14 ----- 0s 2ms/step - acc: 0.9142 - loss: 0.2183
Epoch 98/100
14/14 ----- 0s 2ms/step - acc: 0.9325 - loss: 0.1860
Epoch 99/100
14/14 ----- 0s 3ms/step - acc: 0.9276 - loss: 0.2312
Epoch 100/100
14/14 ----- 0s 2ms/step - acc: 0.8597 - loss: 0.4081
Model: "sequential_2"

```

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 20)	620
dense_6 (Dense)	(None, 1)	21

```

Total params: 1,925 (7.52 KB)
Trainable params: 641 (2.50 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 1,284 (5.02 KB)
None
5/5 ----- 0s 4ms/step - acc: 0.8284 - loss: 0.4921
[0.4397737383842468, 0.8601398468017578]

```

Use Image Classification on the hand written digits data set (mnist)

1. Plot the loss and accuracy for both training data and validation data using the history object in the source code.
2. Plot one of the images in the test data, and then do inferencing to

check what is the prediction of the model on that single image.

3. We had used 2 hidden layers and Relu activation. Try to change the number of hidden layer and the activation to tanh or sigmoid and see what happens.

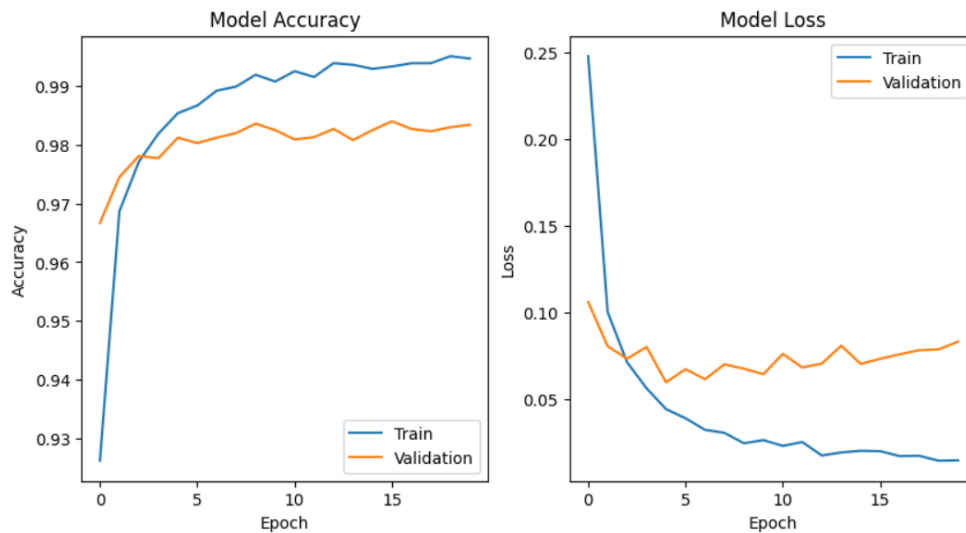
4. Run the same code without scaling the images and check the performance?

```
✓ [15] Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz  
4m 11490434/11490434 0s 0us/step  
Epoch 1/20  
469/469 14s 23ms/step - accuracy: 0.8685 - loss: 0.4438 - val_accuracy: 0.9667 - val_loss: 0.1061  
Epoch 2/20  
469/469 20s 23ms/step - accuracy: 0.9682 - loss: 0.1046 - val_accuracy: 0.9745 - val_loss: 0.0806  
Epoch 3/20  
469/469 9s 19ms/step - accuracy: 0.9781 - loss: 0.0678 - val_accuracy: 0.9781 - val_loss: 0.0734  
Epoch 4/20  
469/469 12s 22ms/step - accuracy: 0.9822 - loss: 0.0544 - val_accuracy: 0.9777 - val_loss: 0.0802  
Epoch 5/20  
469/469 10s 22ms/step - accuracy: 0.9839 - loss: 0.0461 - val_accuracy: 0.9812 - val_loss: 0.0599  
Epoch 6/20  
469/469 20s 21ms/step - accuracy: 0.9873 - loss: 0.0378 - val_accuracy: 0.9803 - val_loss: 0.0674  
Epoch 7/20  
469/469 10s 20ms/step - accuracy: 0.9895 - loss: 0.0306 - val_accuracy: 0.9812 - val_loss: 0.0616  
Epoch 8/20  
469/469 13s 27ms/step - accuracy: 0.9904 - loss: 0.0289 - val_accuracy: 0.9820 - val_loss: 0.0702  
Epoch 9/20  
469/469 10s 22ms/step - accuracy: 0.9925 - loss: 0.0227 - val_accuracy: 0.9836 - val_loss: 0.0677  
Epoch 10/20  
469/469 10s 22ms/step - accuracy: 0.9909 - loss: 0.0257 - val_accuracy: 0.9825 - val_loss: 0.0645  
Epoch 11/20  
469/469 8s 18ms/step - accuracy: 0.9932 - loss: 0.0215 - val_accuracy: 0.9809 - val_loss: 0.0763  
Epoch 12/20  
469/469 10s 22ms/step - accuracy: 0.9926 - loss: 0.0226 - val_accuracy: 0.9813 - val_loss: 0.0684  
Epoch 13/20  
469/469 10s 21ms/step - accuracy: 0.9942 - loss: 0.0174 - val_accuracy: 0.9827 - val_loss: 0.0707  
Epoch 14/20  
469/469 9s 18ms/step - accuracy: 0.9945 - loss: 0.0171 - val_accuracy: 0.9808 - val_loss: 0.0810  
Epoch 15/20  
469/469 11s 23ms/step - accuracy: 0.9936 - loss: 0.0187 - val_accuracy: 0.9825 - val_loss: 0.0705  
Epoch 16/20  
469/469 20s 22ms/step - accuracy: 0.9930 - loss: 0.0215 - val_accuracy: 0.9840 - val_loss: 0.0734
```

```

Epoch 18/20
469/469 11s 21ms/step - accuracy: 0.9940 - loss: 0.0170 - val_accuracy: 0.9823 - val_loss: 0.0784
Epoch 19/20
469/469 12s 25ms/step - accuracy: 0.9955 - loss: 0.0122 - val_accuracy: 0.9830 - val_loss: 0.0788
Epoch 20/20
469/469 10s 21ms/step - accuracy: 0.9953 - loss: 0.0135 - val_accuracy: 0.9834 - val_loss: 0.0833

```

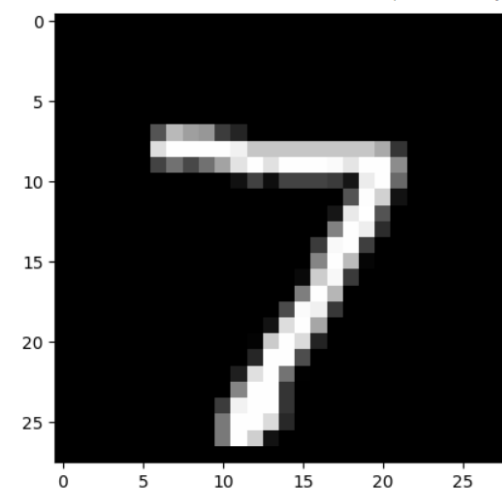


```

print('Model prediction: ', np.argmax(prediction))
Epoch 1/20
469/469 11s 20ms/step - accuracy: 0.8627 - loss: 0.4575 - val_accuracy: 0.9686 - val_loss: 0.1000
Epoch 2/20
469/469 11s 21ms/step - accuracy: 0.9670 - loss: 0.1036 - val_accuracy: 0.9733 - val_loss: 0.0819
Epoch 3/20
469/469 10s 22ms/step - accuracy: 0.9782 - loss: 0.0702 - val_accuracy: 0.9784 - val_loss: 0.0726
Epoch 4/20
469/469 9s 18ms/step - accuracy: 0.9820 - loss: 0.0547 - val_accuracy: 0.9793 - val_loss: 0.0694
Epoch 5/20
469/469 10s 22ms/step - accuracy: 0.9857 - loss: 0.0451 - val_accuracy: 0.9800 - val_loss: 0.0682
Epoch 6/20
469/469 10s 22ms/step - accuracy: 0.9884 - loss: 0.0366 - val_accuracy: 0.9820 - val_loss: 0.0612
Epoch 7/20
469/469 9s 18ms/step - accuracy: 0.9884 - loss: 0.0329 - val_accuracy: 0.9799 - val_loss: 0.0685
Epoch 8/20
469/469 10s 18ms/step - accuracy: 0.9888 - loss: 0.0337 - val_accuracy: 0.9842 - val_loss: 0.0595
Epoch 9/20
469/469 13s 24ms/step - accuracy: 0.9917 - loss: 0.0239 - val_accuracy: 0.9825 - val_loss: 0.0631
Epoch 10/20
469/469 10s 22ms/step - accuracy: 0.9921 - loss: 0.0225 - val_accuracy: 0.9828 - val_loss: 0.0655
Epoch 11/20
469/469 10s 22ms/step - accuracy: 0.9938 - loss: 0.0191 - val_accuracy: 0.9823 - val_loss: 0.0723
Epoch 12/20
469/469 9s 18ms/step - accuracy: 0.9928 - loss: 0.0215 - val_accuracy: 0.9830 - val_loss: 0.0702
Epoch 13/20
469/469 10s 22ms/step - accuracy: 0.9931 - loss: 0.0212 - val_accuracy: 0.9812 - val_loss: 0.0717
Epoch 14/20
469/469 10s 21ms/step - accuracy: 0.9942 - loss: 0.0174 - val_accuracy: 0.9825 - val_loss: 0.0720
Epoch 15/20
469/469 9s 18ms/step - accuracy: 0.9941 - loss: 0.0168 - val_accuracy: 0.9847 - val_loss: 0.0657
Epoch 16/20
469/469 11s 20ms/step - accuracy: 0.9946 - loss: 0.0167 - val_accuracy: 0.9855 - val_loss: 0.0679
Epoch 17/20

```

✓ [16] 1m 469/469 10s 22ms/step - accuracy: 0.9950 - loss: 0.0136 - val_accuracy: 0.9839 - val_loss: 0.0766
Epoch 18/20
469/469 11s 23ms/step - accuracy: 0.9950 - loss: 0.0143 - val_accuracy: 0.9828 - val_loss: 0.0765
Epoch 19/20
469/469 20s 22ms/step - accuracy: 0.9946 - loss: 0.0161 - val_accuracy: 0.9825 - val_loss: 0.0757
Epoch 20/20
469/469 12s 25ms/step - accuracy: 0.9952 - loss: 0.0137 - val_accuracy: 0.9836 - val_loss: 0.0820

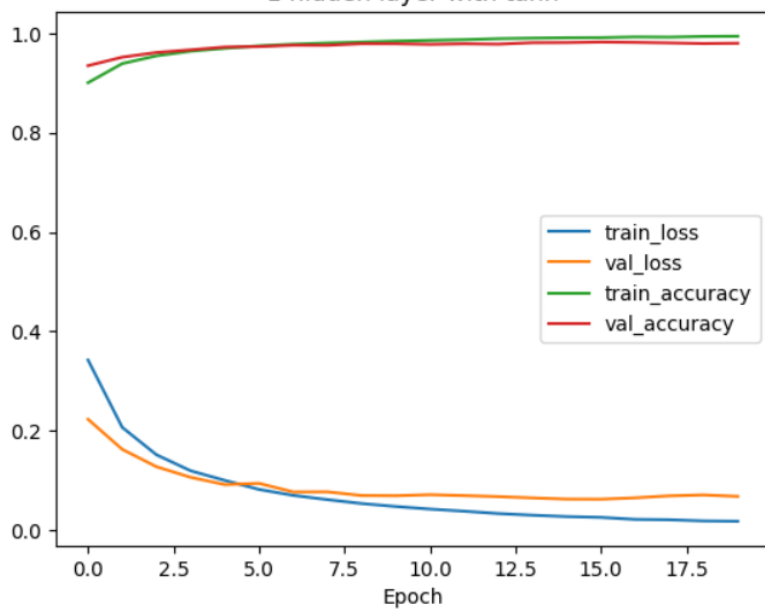


1/1 0s 69ms/step
Model prediction: 7

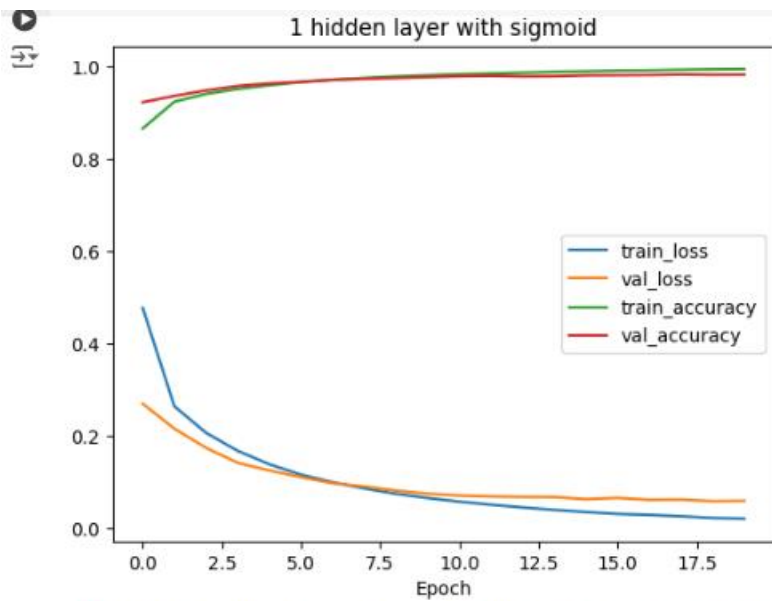
✓ [17] 1m



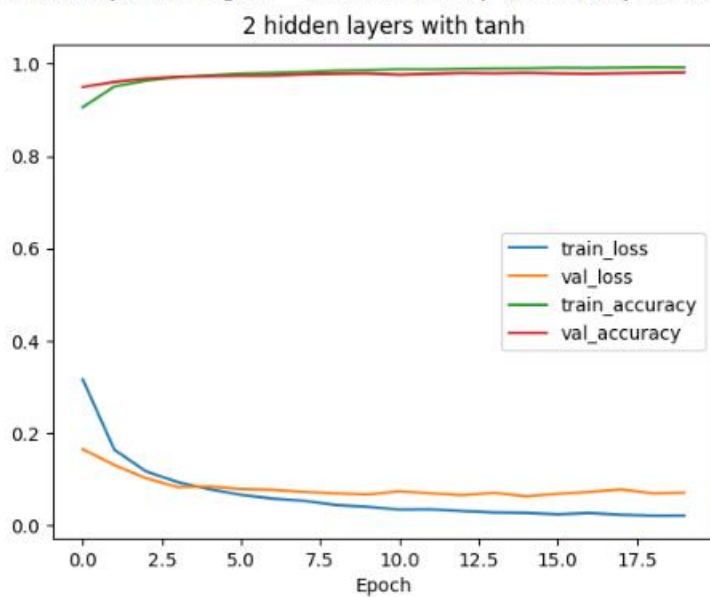
1 hidden layer with tanh



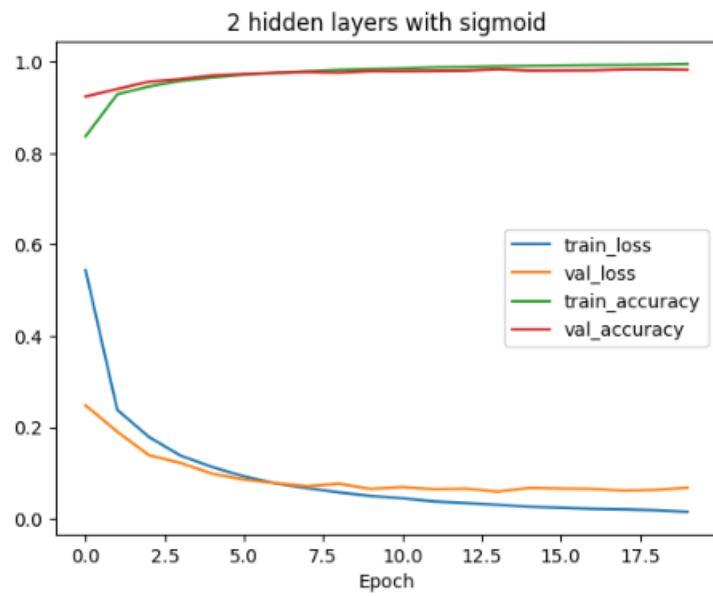
1 hidden layer with tanh - Test loss: 0.0672, Test accuracy: 0.9803



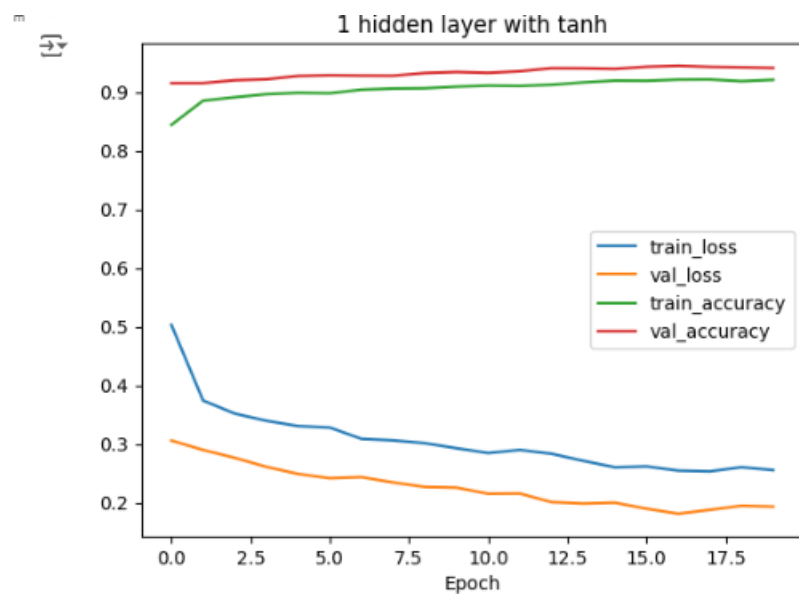
1 hidden layer with sigmoid - Test loss: 0.0605, Test accuracy: 0.9819



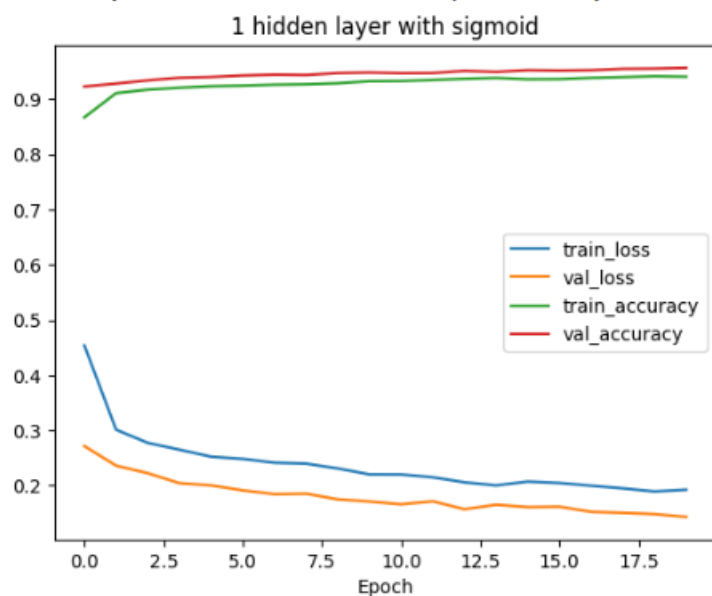
2 hidden layers with tanh - Test loss: 0.0703, Test accuracy: 0.9816



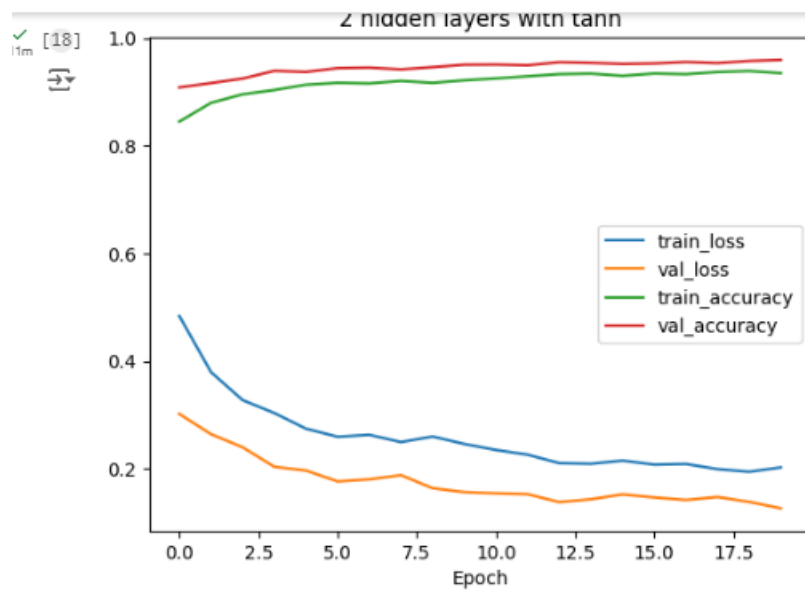
2 hidden layers with sigmoid - Test loss: 0.0685, Test accuracy: 0.9822



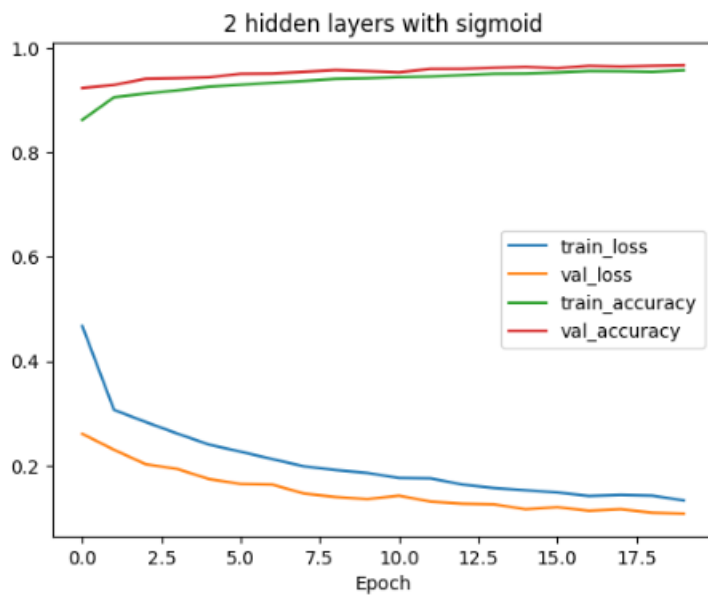
1 hidden layer with tanh - Test loss: 0.1931, Test accuracy: 0.9411



1 hidden layer with sigmoid - Test loss: 0.1430, Test accuracy: 0.9569



2 hidden layers with tanh - Test loss: 0.1265, Test accuracy: 0.9597



2 hidden layers with sigmoid - Test loss: 0.1079, Test accuracy: 0.9669