Assignment Instructions: Assignment 1 – Mukhtar, 2022

This assignment basically aims to explore and extend the first Neural Network model.

In this assignment, I accomplished the following:

1. Modified an existing neural network model to improve performance

2. Explained how different approaches affect the performance of different models

Summary:

	Original Model	Modelo	Model x1	Modelx2	model x3	Modelx4	Modelx5
No. of hidden layes	2	3	3 & 2	2	2	2	2
Units in hidden layers	16, 16	16, 16, 16	16, 16, & 16	32, 64	16, 16	16, 16	20, 15
Acivation function	relu, sigmoid	relu, sigmoid	relu, sigmoid	relu, sigmoid	relu, sigmoid	tanh, sigmoid	relu, sigmoid
Optimizer	rmsprop	rmsprop	adam	adam	adam	adm	rmsprop
Loss function	binary_crossentropy	binary_crossentropy	binary_crossentropy	binary_crossentropy	mse	mse	binary_crossentropy
Metrics	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy
Dropout							0.2
	Yardstick						
Improved Accuracy?	0.998	Yes	Yes	Yes, but no signf. Change b/w x1	No signf. Change	No signf. Change	No.signf. Change
Improved Loss?	0.0083	Yes	Yes	Yes, but no signf. Change b/w x2	Yes, but reduced relative to x2	Yes	Yes, but reduced relative to x4
Improved Val_Accuracy?	0.8901	No	No	No	No, but reduced relative to x2	No	No.signf. Change , reduced relative to x4
Improved Val_Loss?	0.2741	Yes	No signf. Change	No signf. Change	Yes, and increased relative to x2	Yes	Yes

Conclusions:

Based on the analysis as summarized above, we can establish that:

a. Adding more hidden and increased units amount to better training accuracy and improved loss
b. Changing the optimizer from rmsprop to Adam also increases the training and loss accuracies, but reduced the validity accuracy but does not have a significant effect on the validity accuracy
c. Changing the loss function from binary _cross entropy to mse results in improved training and validity losses, but does not significantly affect on the training and validity accuracies
d. Changing the activation model from relu to tanh improves the training loss but the validity loss is not affected significantly
e. Adding a dropout function (0.2) might reduce the strength of both the training loss and the validity accuracy

Analysis:

For the IMDB example, I did the following:

1. In the original model, I used two hidden layers and tried using one or three hidden layers to see how doing so would affect validation and test accuracy.

Results:

```
model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
```

The initial results using two hidden layers provide these validation and test accuracy:

```
Epoch 1/20
30/30 [==============================] - 4s 88ms/step - loss: 0.5310 - accuracy: 0.7892 - val_loss: 0.4032 - val_accuracy: 0.8661
Epoch 2/20
30/30 [==============================] - 1s 41ms/step - loss: 0.3246 - accuracy: 0.8976 - val_loss: 0.3127 - val_accuracy: 0.8852
Epoch 3/20
30/30 [==============================] - 1s 38ms/step - loss: 0.2352 - accuracy: 0.9223 - val_loss: 0.2809 - val_accuracy: 0.8884
Epoch 4/20
30/30 [==============================] - 1s 41ms/step - loss: 0.1861 - accuracy: 0.9388 - val_loss: 0.2719 - val_accuracy: 0.8907
Epoch 5/20
30/30 [==============================] - 1s 38ms/step - loss: 0.1507 - accuracy: 0.9522 - val_loss: 0.3102 - val_accuracy: 0.8766
Epoch 6/20
30/30 [==============================] - 1s 39ms/step - loss: 0.1241 - accuracy: 0.9599 - val_loss: 0.2937 - val_accuracy: 0.8860
Epoch 7/20
30/30 [==============================] - 1s 38ms/step - loss: 0.1059 - accuracy: 0.9679 - val_loss: 0.3002 - val_accuracy: 0.8863
Epoch 8/20
30/30 [==============================] - 1s 38ms/step - loss: 0.0858 - accuracy: 0.9759 - val_loss: 0.3190 - val_accuracy: 0.8827
Epoch 9/20
30/30 [==============================] - 1s 40ms/step - loss: 0.0717 - accuracy: 0.9795 - val_loss: 0.3576 - val_accuracy: 0.8786
Epoch 10/20
30/30 [==============================] - 1s 38ms/step - loss: 0.0607 - accuracy: 0.9831 - val_loss: 0.3731 - val_accuracy: 0.8803
Epoch 11/20
30/30 [==============================] - 1s 38ms/step - loss: 0.0504 - accuracy: 0.9871 - val_loss: 0.3944 - val_accuracy: 0.8783
Epoch 12/20
30/30 [==============================] - 1s 41ms/step - loss: 0.0397 - accuracy: 0.9911 - val_loss: 0.4229 - val_accuracy: 0.8736
Epoch 13/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0323 - accuracy: 0.9942 - val_loss: 0.4520 - val_accuracy: 0.8756
Epoch 14/20
30/30 [==============================] - 1s 40ms/step - loss: 0.0252 - accuracy: 0.9957 - val_loss: 0.4815 - val_accuracy: 0.8730
Epoch 15/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0207 - accuracy: 0.9964 - val_loss: 0.5204 - val_accuracy: 0.8692
Epoch 16/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0168 - accuracy: 0.9971 - val_loss: 0.5548 - val_accuracy: 0.8677
Epoch 17/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0120 - accuracy: 0.9988 - val_loss: 0.5885 - val_accuracy: 0.8691
Epoch 18/20
30/30 [==============================] - 1s 40ms/step - loss: 0.0095 - accuracy: 0.9992 - val_loss: 0.6381 - val_accuracy: 0.8687
```

Here, we have the best “Accuracy” of 0.9992 and “Loss” of 0.0095 at Epoch 30/30.

We also have the best “Val-Accuracy” of 0.8884 and “Val-Loss” of 0.2719.

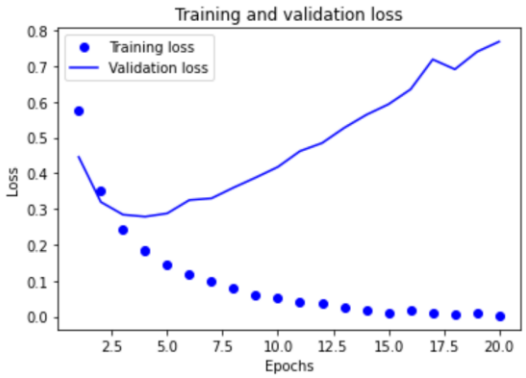
Model O:

```
model = keras.Sequential([
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid")
])
```

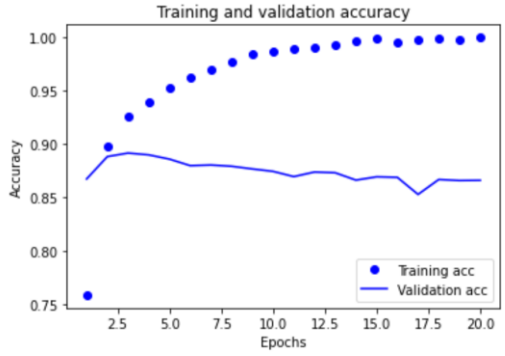
By adding a hidden layer, we now have improved “Accuracies” as highlighted:

```
30/30 [==============================] - 1s 36ms/step - loss: 0.2872 - accuracy: 0.9025 - val_loss: 0.2929 - v
Epoch 3/20
30/30 [==============================] - 1s 36ms/step - loss: 0.2073 - accuracy: 0.9285 - val_loss: 0.2999 - val_accuracy: 0.8787
Epoch 4/20
30/30 [==============================] - 1s 36ms/step - loss: 0.1617 - accuracy: 0.9441 - val_loss: 0.2801 - val_accuracy: 0.8886
Epoch 5/20
30/30 [==============================] - 1s 37ms/step - loss: 0.1229 - accuracy: 0.9605 - val_loss: 0.2938 - val_accuracy: 0.8894
Epoch 6/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0971 - accuracy: 0.9695 - val_loss: 0.3233 - val_accuracy: 0.8863
Epoch 7/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0718 - accuracy: 0.9784 - val_loss: 0.3639 - val_accuracy: 0.8742
Epoch 8/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0580 - accuracy: 0.9845 - val_loss: 0.3868 - val_accuracy: 0.8787
Epoch 9/20
30/30 [==============================] - 1s 37ms/step - loss: 0.0400 - accuracy: 0.9901 - val_loss: 0.4107 - val_accuracy: 0.8773
Epoch 10/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0297 - accuracy: 0.9932 - val_loss: 0.4466 - val_accuracy: 0.8761
Epoch 11/20
30/30 [==============================] - 1s 37ms/step - loss: 0.0226 - accuracy: 0.9946 - val_loss: 0.5139 - val_accuracy: 0.8656
Epoch 12/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0147 - accuracy: 0.9973 - val_loss: 0.5310 - val_accuracy: 0.8727
Epoch 13/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0207 - accuracy: 0.9935 - val_loss: 0.5654 - val_accuracy: 0.8727
Epoch 14/20
30/30 [==============================] - 1s 37ms/step - loss: 0.0043 - accuracy: 0.9997 - val_loss: 0.6306 - val_accuracy: 0.8681
Epoch 15/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0195 - accuracy: 0.9947 - val_loss: 0.6325 - val_accuracy: 0.8687
Epoch 16/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0021 - accuracy: 0.9999 - val_loss: 0.6681 - val_accuracy: 0.8685
Epoch 17/20
30/30 [==============================] - 1s 37ms/step - loss: 0.0015 - accuracy: 0.9999 - val_loss: 0.7140 - val_accuracy: 0.8685
Epoch 18/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0178 - accuracy: 0.9948 - val_loss: 0.7463 - val_accuracy: 0.8683
Epoch 19/20
30/30 [==============================] - 1s 37ms/step - loss: 7.2786e-04 - accuracy: 0.9999 - val_loss: 0.7758 - val_accuracy: 0.8688
Epoch 20/20
30/30 [==============================] - 1s 37ms/step - loss: 5.4256e-04 - accuracy: 1.0000 - val_loss: 0.8137 - val_accuracy: 0.8679
```

I now generated a plot of the training and validation, loss and accuracy



The result indicates that training loss decreases with every epoch, and the training accuracy increases with every epoch. However, here we have overfitting, and to remedy that, we could stop training following fourth epoch.



Changing the optimizer

Modelx1: I now created modelsx1(3-layers) and x1_1(1-layer), changed the optimizer to "adam", applied the model_fit function, and compared the models.

Results:

```
Epoch 1/20
30/30 [==============================] - 2s 43ms/step - loss: 0.6032 - accuracy: 0.7295 - val_loss: 0.4508 - val_accuracy: 0.8578
Epoch 2/20
30/30 [==============================] - 1s 32ms/step - loss: 0.3292 - accuracy: 0.8967 - val_loss: 0.3078 - val_accuracy: 0.8809
Epoch 3/20
30/30 [==============================] - 1s 32ms/step - loss: 0.2097 - accuracy: 0.9295 - val_loss: 0.2748 - val_accuracy: 0.8902
Epoch 4/20
30/30 [==============================] - 1s 32ms/step - loss: 0.1481 - accuracy: 0.9519 - val_loss: 0.2975 - val_accuracy: 0.8821
Epoch 5/20
30/30 [==============================] - 1s 31ms/step - loss: 0.1085 - accuracy: 0.9683 - val_loss: 0.3078 - val_accuracy: 0.8843
Epoch 6/20
30/30 [==============================] - 1s 31ms/step - loss: 0.0762 - accuracy: 0.9817 - val_loss: 0.3367 - val_accuracy: 0.8803
Epoch 7/20
30/30 [==============================] - 1s 31ms/step - loss: 0.0523 - accuracy: 0.9894 - val_loss: 0.3714 - val_accuracy: 0.8793
Epoch 8/20
30/30 [==============================] - 1s 31ms/step - loss: 0.0342 - accuracy: 0.9950 - val_loss: 0.4122 - val_accuracy: 0.8763
Epoch 9/20
```

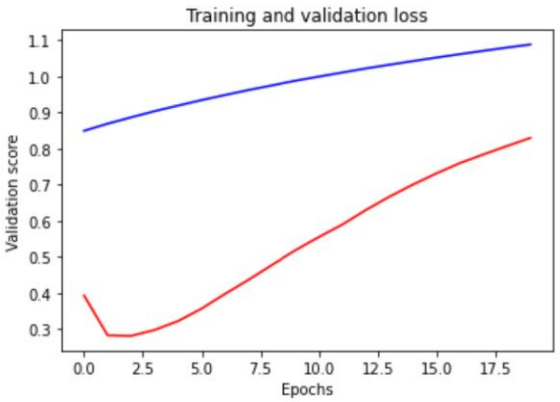
```
Epoch 9/20
30/30 [==============================] - 1s 32ms/step - loss: 3.8265e-04 - accuracy: 0.9999 - val_loss: 0.8553 - val_accuracy: 0.8697
Epoch 10/20
30/30 [==============================] - 1s 31ms/step - loss: 3.5219e-04 - accuracy: 0.9999 - val_loss: 0.8651 - val_accuracy: 0.8695
Epoch 11/20
30/30 [==============================] - 1s 30ms/step - loss: 3.2549e-04 - accuracy: 0.9999 - val_loss: 0.8741 - val_accuracy: 0.8692
Epoch 12/20
30/30 [==============================] - 1s 31ms/step - loss: 3.0144e-04 - accuracy: 0.9999 - val_loss: 0.8834 - val_accuracy: 0.8699
Epoch 13/20
30/30 [==============================] - 1s 31ms/step - loss: 2.8078e-04 - accuracy: 0.9999 - val_loss: 0.8924 - val_accuracy: 0.8702
Epoch 14/20
30/30 [==============================] - 1s 31ms/step - loss: 2.6027e-04 - accuracy: 1.0000 - val_loss: 0.9007 - val_accuracy: 0.8698
Epoch 15/20
30/30 [==============================] - 1s 32ms/step - loss: 2.4384e-04 - accuracy: 1.0000 - val_loss: 0.9086 - val_accuracy: 0.8697
Epoch 16/20
30/30 [==============================] - 1s 31ms/step - loss: 2.2680e-04 - accuracy: 1.0000 - val_loss: 0.9165 - val_accuracy: 0.8696
Epoch 17/20
30/30 [==============================] - 1s 31ms/step - loss: 2.1185e-04 - accuracy: 1.0000 - val_loss: 0.9239 - val_accuracy: 0.8701
Epoch 18/20
30/30 [==============================] - 1s 31ms/step - loss: 1.9678e-04 - accuracy: 1.0000 - val_loss: 0.9315 - val_accuracy: 0.8699
Epoch 19/20
30/30 [==============================] - 1s 31ms/step - loss: 1.8309e-04 - accuracy: 1.0000 - val_loss: 0.9381 - val_accuracy: 0.8691
Epoch 20/20
30/30 [==============================] - 1s 31ms/step - loss: 1.6967e-04 - accuracy: 1.0000 - val_loss: 0.9450 - val_accuracy: 0.8697
```

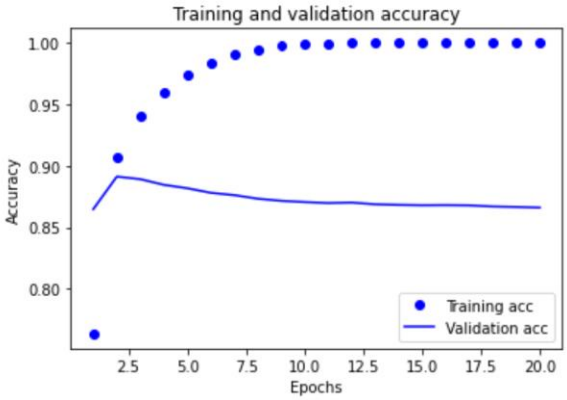
Here, we now have a changed best “Accuracy” of 1.0000 and “Loss” 1.6967e-04 beginning from epoch 15/20 through 20/20.

We now have the best “Val-Accuracy” of 0.8843 and “Val-Loss” of 0.2748.

This indicates that changing the optimizer to “Adam” gives us a much more accurate outcome. The plots also signify that specifically, the validation accuracy is relatively steady compared to the previous model where the accuracy significantly declined at epoch 17.5 iterations.

Upon plotting the outcome, we still obtained better training and validation accuracy. The validation accuracy is smooth across the epoch, but it is at its best at the 2.5 epoch.





2. ModelX2: I tried using layers with more hidden units or fewer hidden units: 32 units, 64 units, and so on.

Using the modelx2, I now created a model and changed the hidden units to 32, and 64 units with the same “adam” optimizer.

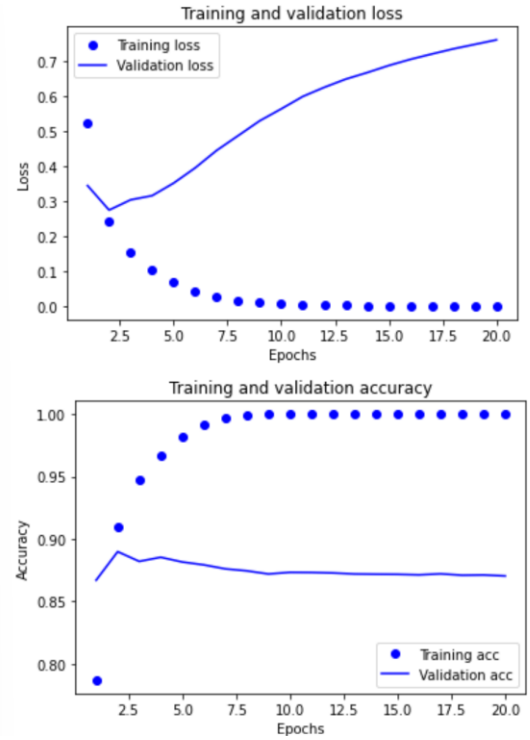
Results:

Although there is no significant change, we now have the best “Accuracy” of 1.0000 and “Loss” 5.3937e-04 beginning at epoch 20/20 but beginning earlier at 13/20 epoch.

The “Val-Accuracy” is virtually unchanged.

```
Epoch 2/20
30/30 [==============================] - 1s 40ms/step - loss: 0.2435 - accuracy: 0.9093 - val_loss: 0.2750 - val_accuracy: 0.8898
Epoch 3/20
30/30 [==============================] - 1s 39ms/step - loss: 0.1529 - accuracy: 0.9475 - val_loss: 0.3042 - val_accuracy: 0.8820
Epoch 4/20
30/30 [==============================] - 1s 41ms/step - loss: 0.1050 - accuracy: 0.9673 - val_loss: 0.3161 - val_accuracy: 0.8853
Epoch 5/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0696 - accuracy: 0.9822 - val_loss: 0.3520 - val_accuracy: 0.8815
Epoch 6/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0435 - accuracy: 0.9915 - val_loss: 0.3955 - val_accuracy: 0.8792
Epoch 7/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0268 - accuracy: 0.9969 - val_loss: 0.4455 - val_accuracy: 0.8760
Epoch 8/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0163 - accuracy: 0.9990 - val_loss: 0.4877 - val_accuracy: 0.8744
Epoch 9/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0099 - accuracy: 0.9997 - val_loss: 0.5301 - val_accuracy: 0.8719
Epoch 10/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0065 - accuracy: 0.9999 - val_loss: 0.5642 - val_accuracy: 0.8732
Epoch 11/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0042 - accuracy: 0.9999 - val_loss: 0.5996 - val_accuracy: 0.8731
Epoch 12/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0030 - accuracy: 0.9999 - val_loss: 0.6258 - val_accuracy: 0.8727
Epoch 13/20
30/30 [==============================] - 1s 40ms/step - loss: 0.0022 - accuracy: 1.0000 - val_loss: 0.6493 - val_accuracy: 0.8719
Epoch 14/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0017 - accuracy: 1.0000 - val_loss: 0.6680 - val_accuracy: 0.8717
Epoch 15/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0013 - accuracy: 1.0000 - val_loss: 0.6882 - val_accuracy: 0.8716
Epoch 16/20
30/30 [==============================] - 1s 39ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.7058 - val_accuracy: 0.8712
Epoch 17/20
30/30 [==============================] - 1s 40ms/step - loss: 8.7251e-04 - accuracy: 1.0000 - val_loss: 0.7211 - val_accuracy: 0.8720
Epoch 18/20
30/30 [==============================] - 1s 39ms/step - loss: 7.3257e-04 - accuracy: 1.0000 - val_loss: 0.7357 - val_accuracy: 0.8709
Epoch 19/20
30/30 [==============================] - 1s 39ms/step - loss: 6.2372e-04 - accuracy: 1.0000 - val_loss: 0.7486 - val_accuracy: 0.8711
Epoch 20/20
30/30 [==============================] - 1s 39ms/step - loss: 5.3937e-04 - accuracy: 1.0000 - val_loss: 0.7614 - val_accuracy: 0.8703
```

✓ 41s completed at 12:56 AM



Unlike the previous model, we have a smooth training and validity loss and accuracy.

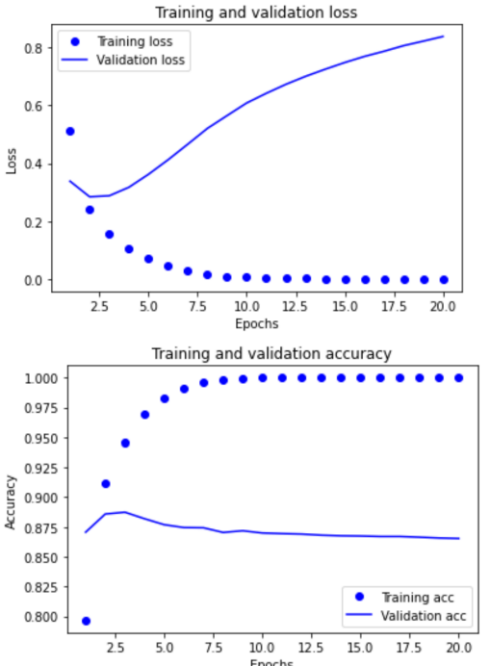
3. Modelx3: I tried using the mse loss function instead of binary_crossentropy.

Using the modelx3, I now created a model and changed the hidden units back to 16, and 16 units with the same “adam” optimizer but used the “mse” as the loss function.

Results:

There is a significant change in the outcome, we now have a less improved best “Accuracy” relative to the previous model where we used "binary_crossentropy" as our loss function. We now have “Accuracy” of 0.9969 and “Loss” 0.0037 at epoch 20/20 and the best “Va-accuracy" and “Val_loss” at 4/20 epoch.

```
Epoch 2/20
30/30 [==============================] - 1s 39ms/step - loss: 0.1026 - accuracy: 0.8965 - val_loss: 0.0991 - val_accuracy: 0.8805
Epoch 3/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0673 - accuracy: 0.9296 - val_loss: 0.0869 - val_accuracy: 0.8884
Epoch 4/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0496 - accuracy: 0.9500 - val_loss: 0.0833 - val_accuracy: 0.8913
Epoch 5/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0379 - accuracy: 0.9651 - val_loss: 0.0836 - val_accuracy: 0.8872
Epoch 6/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0297 - accuracy: 0.9743 - val_loss: 0.0846 - val_accuracy: 0.8851
Epoch 7/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0230 - accuracy: 0.9822 - val_loss: 0.0870 - val_accuracy: 0.8812
Epoch 8/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0181 - accuracy: 0.9871 - val_loss: 0.0895 - val_accuracy: 0.8775
Epoch 9/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0143 - accuracy: 0.9902 - val_loss: 0.0918 - val_accuracy: 0.8756
Epoch 10/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0114 - accuracy: 0.9927 - val_loss: 0.0940 - val_accuracy: 0.8750
Epoch 11/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0092 - accuracy: 0.9946 - val_loss: 0.0972 - val_accuracy: 0.8722
Epoch 12/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0077 - accuracy: 0.9951 - val_loss: 0.0978 - val_accuracy: 0.8742
Epoch 13/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0065 - accuracy: 0.9957 - val_loss: 0.0993 - val_accuracy: 0.8730
Epoch 14/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0056 - accuracy: 0.9963 - val_loss: 0.1008 - val_accuracy: 0.8714
Epoch 15/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0050 - accuracy: 0.9967 - val_loss: 0.1018 - val_accuracy: 0.8705
Epoch 16/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0046 - accuracy: 0.9967 - val_loss: 0.1026 - val_accuracy: 0.8705
Epoch 17/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0043 - accuracy: 0.9967 - val_loss: 0.1035 - val_accuracy: 0.8704
Epoch 18/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0040 - accuracy: 0.9968 - val_loss: 0.1043 - val_accuracy: 0.8698
Epoch 19/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0039 - accuracy: 0.9969 - val_loss: 0.1052 - val_accuracy: 0.8702
Epoch 20/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0037 - accuracy: 0.9969 - val_loss: 0.1056 - val_accuracy: 0.8693
```



Still, the result indicates that the training loss decreases with every epoch, and the training accuracy increases with every epoch. However, here we have overfitting. For the validation loss and accuracy, they seem to peak at the third epoch. In general, we could say that this is a typical model that performs well on the training data but would not do well on unseen data. To remedy that, we could stop the training following the third epoch

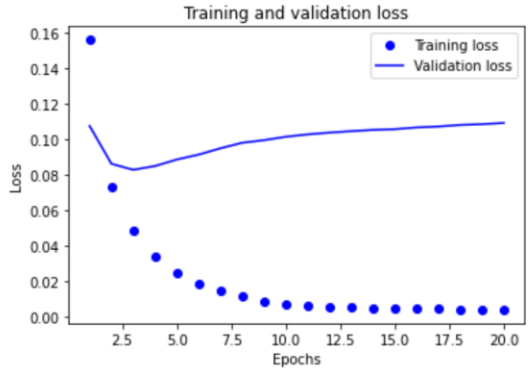
4. Modelx4: I tried using the tanh activation (an activation popular in the early days of neural networks) instead of relu.

I now created a modelx4 and changed the activation to “tanh” and also maintain the “sigmoid”, “me”, and “adam” functions.

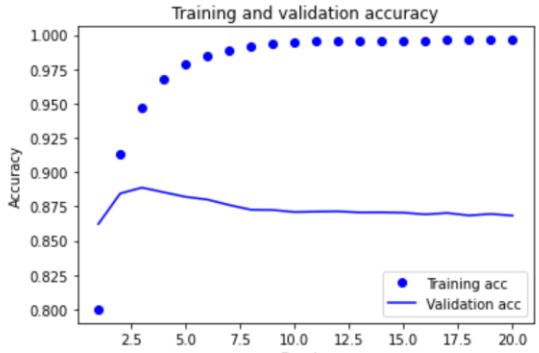
Results:

```
30/30 [==============================] - 1s 36ms/step - loss: 0.0482 - accuracy: 0.9466 - val_loss: 0.0828 - val_accuracy: 0.8887
Epoch 4/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0338 - accuracy: 0.9675 - val_loss: 0.0849 - val_accuracy: 0.8853
Epoch 5/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0245 - accuracy: 0.9789 - val_loss: 0.0886 - val_accuracy: 0.8820
Epoch 6/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0188 - accuracy: 0.9845 - val_loss: 0.0913 - val_accuracy: 0.8800
Epoch 7/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0148 - accuracy: 0.9891 - val_loss: 0.0949 - val_accuracy: 0.8760
Epoch 8/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0112 - accuracy: 0.9916 - val_loss: 0.0980 - val_accuracy: 0.8726
Epoch 9/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0087 - accuracy: 0.9937 - val_loss: 0.0995 - val_accuracy: 0.8724
Epoch 10/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0072 - accuracy: 0.9945 - val_loss: 0.1014 - val_accuracy: 0.8709
Epoch 11/20
30/30 [==============================] - 1s 38ms/step - loss: 0.0062 - accuracy: 0.9952 - val_loss: 0.1027 - val_accuracy: 0.8712
Epoch 12/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0056 - accuracy: 0.9955 - val_loss: 0.1037 - val_accuracy: 0.8714
Epoch 13/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0050 - accuracy: 0.9958 - val_loss: 0.1046 - val_accuracy: 0.8706
Epoch 14/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0048 - accuracy: 0.9960 - val_loss: 0.1053 - val_accuracy: 0.8707
Epoch 15/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0046 - accuracy: 0.9960 - val_loss: 0.1057 - val_accuracy: 0.8704
Epoch 16/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0044 - accuracy: 0.9961 - val_loss: 0.1066 - val_accuracy: 0.8692
Epoch 17/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0043 - accuracy: 0.9961 - val_loss: 0.1072 - val_accuracy: 0.8702
Epoch 18/20
30/30 [==============================] - 1s 37ms/step - loss: 0.0042 - accuracy: 0.9961 - val_loss: 0.1081 - val_accuracy: 0.8684
Epoch 19/20
30/30 [==============================] - 1s 36ms/step - loss: 0.0041 - accuracy: 0.9961 - val_loss: 0.1085 - val_accuracy: 0.8695
Epoch 20/20
30/30 [==============================] - 1s 35ms/step - loss: 0.0039 - accuracy: 0.9963 - val_loss: 0.1092 - val_accuracy: 0.8683
```

There is no significant change in the outcomes. The plots of loss and accuracy of the training and validation confirm that.




```
<matplotlib.legend.Legend at 0x7fe89287da50>
```



5. Finally, I also used a few techniques we studied in class, including regularization, dropout, etc., to get my model to perform better on validation.

   I now used a “dropout” function in modelx5 by setting the parameter to “0.2” while maintaining other functions.
   Results:

```
30/30 [==============================] - 3s 66ms/step - loss: 0.5618 - accuracy: 0.7593 - val_loss: 0.3875 - val_accuracy: 0.8632
Epoch 2/20
30/30 [==============================] - 1s 43ms/step - loss: 0.3189 - accuracy: 0.8850 - val_loss: 0.2905 - val_accuracy: 0.8874
Epoch 3/20
30/30 [==============================] - 1s 42ms/step - loss: 0.2226 - accuracy: 0.9237 - val_loss: 0.2738 - val_accuracy: 0.8903
Epoch 4/20
30/30 [==============================] - 1s 44ms/step - loss: 0.1679 - accuracy: 0.9451 - val_loss: 0.2832 - val_accuracy: 0.8868
Epoch 5/20
30/30 [==============================] - 1s 42ms/step - loss: 0.1301 - accuracy: 0.9581 - val_loss: 0.2918 - val_accuracy: 0.8847
Epoch 6/20
30/30 [==============================] - 1s 43ms/step - loss: 0.1012 - accuracy: 0.9713 - val_loss: 0.3128 - val_accuracy: 0.8822
Epoch 7/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0768 - accuracy: 0.9795 - val_loss: 0.3356 - val_accuracy: 0.8831
Epoch 8/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0584 - accuracy: 0.9863 - val_loss: 0.3709 - val_accuracy: 0.8800
Epoch 9/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0440 - accuracy: 0.9905 - val_loss: 0.4062 - val_accuracy: 0.8769
Epoch 10/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0327 - accuracy: 0.9944 - val_loss: 0.4346 - val_accuracy: 0.8778
Epoch 11/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0259 - accuracy: 0.9953 - val_loss: 0.4713 - val_accuracy: 0.8756
Epoch 12/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0201 - accuracy: 0.9975 - val_loss: 0.4991 - val_accuracy: 0.8750
Epoch 13/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0154 - accuracy: 0.9986 - val_loss: 0.5280 - val_accuracy: 0.8762
Epoch 14/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0130 - accuracy: 0.9985 - val_loss: 0.5745 - val_accuracy: 0.8723
Epoch 15/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0108 - accuracy: 0.9988 - val_loss: 0.5944 - val_accuracy: 0.8729
Epoch 16/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0089 - accuracy: 0.9994 - val_loss: 0.6146 - val_accuracy: 0.8736
Epoch 17/20
30/30 [==============================] - 1s 43ms/step - loss: 0.0073 - accuracy: 0.9993 - val_loss: 0.6280 - val_accuracy: 0.8736
Epoch 18/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0070 - accuracy: 0.9993 - val_loss: 0.6459 - val_accuracy: 0.8728
Epoch 19/20
30/30 [==============================] - 1s 42ms/step - loss: 0.0050 - accuracy: 0.9998 - val_loss: 0.6760 - val_accuracy: 0.8726
Epoch 20/20
```

✓ 42s completed at 12:45 PM

There is no significant change in the results