|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Hidden Layers & Activation Used** | **Dense Units** | **Regularizations** | **Dropouts** | **Loss and Accuracy on Validation** |
| 1. | 2 & “relu” (From original code) | 16 | None | None | loss: 0.2774 - accuracy: 0.8880 |
| 2. | 1 & “relu” | 16 | None | None | loss: 0.2884 - accuracy: 0.8862 |
| 3. | 3 & “relu” | 16 | None | None | loss: 0.2763 - accuracy: 0.8916 |
| 4. | 1 & “relu” | 32 | None | None | loss: 0.2902 - accuracy: 0.8833 |
| 5. | 2 & “relu” | 32 | None | None | loss: 0.2793 - accuracy: 0.8892 |
| 6. | 3 & “relu” | 32 | None | None | loss: 0.2775 - accuracy: 0.8898 |
| 7. | 1 & “relu” | 64 | None | None | loss: 0.2909 - accuracy: 0.8833 |
| 8. | 2 & “relu” | 64 | None | None | loss: 0.2842 - accuracy: 0.8865 |
| 9. | 3 & “relu” | 64 | None | None | loss: 0.3108 - accuracy: 0.8803 |
| 10. | 2 & “tanh” | 64 | None | None | loss: 0.3181 - accuracy: 0.8776 |

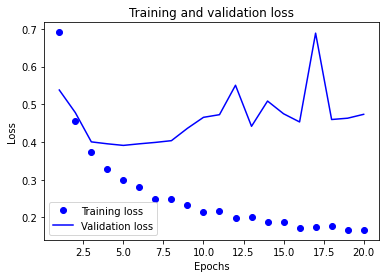
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Hidden Layers & Activation Used** | **Dense Units** | **Regularizations** | **Dropouts** | **Loss and Accuracy on Validation** |
| 11. | 2 & “relu” | 64 | None | None | loss: 0.0854 - Accuracy: 0.8858  (Mse loss function used. |
| 12. | 2 & “relu” | 16 | L2 = 0.001 | None | loss: 0.3245 - Accuracy: 0.8876 |
| 13. | 2 & “relu” | 64 | L2 = 0.001 | None | loss: 0.4183 - Accuracy: 0.8686 |
| 14. | 2 & “relu” | 32 | L1\_L2(L1=0.001, L2=0.01) | None | loss: 0.6535 - accuracy: 0.8624 |
| 15. | 2 & “relu” | 64 | L1\_L2(L1=0.001, L2=0.01) | None | loss: 0.7425 - accuracy: 0.8754 |
| 16. | 2 & “relu” | 16 | None | 0.5 | loss: 0.2708 - accuracy: 0.8923 |
| 17. | 2 & “relu” | 32 | None | 0.5 | loss: 0.2775 - accuracy: 0.8888 |
| 18. | 2 & “relu” | 16 | L2 = 0.001 | 0.5 | loss: 0.3283 - accuracy: 0.8869 |
| 19. | 2 & “relu” | 64 | L2 = 0.001 | 0.5 | loss: 0.3985 - accuracy: 0.8741 |

**Summary:**

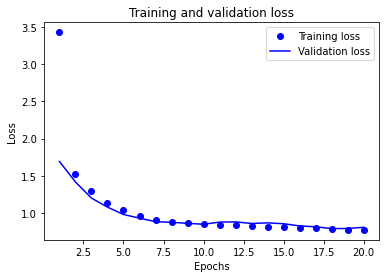
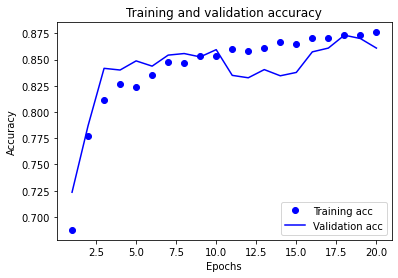
* All the above models have been run with the following parameters to compile the model:

optimizer**=**'rmsprop', loss**=**'binary\_crossentropy', metrics**=**['accuracy']).

* Except for model no.11, which is run on the loss function of “mse” or mean\_squared\_error. Even though the loss function “mse” shows very little validation loss for each epoch, it cannot be applied to binary classification, where the objective is to predict the correct label for each sample, whereas “mse” is used for regression problems to predict continuous variables.
* Using the “tanh” function instead of “relu” shows a steep increase in validation loss as high as 0.98 with no impact on the validation accuracy.
* On the current dataset, changing hidden from 2 to 1 & 3 doesn’t significantly impact, with loss varying between 0.27 and accuracy at 0.88.
* Changing dense units from 16 to 32 and 64 shows a slight increase in validation loss for each epoch and no significant change in the accuracy.
* The concept of regularization helps control the data's overfitting by adding a penalty to the loss function with large weights. In the current scenario, L2 = 0.001 regularizations were applied, where the cost added is proportional to the square of the value of  the weight coefficients.



* The effect of regularization L2 is clearly shown as the validation loss is at the peak of 0.6887. However, the validation accuracy is still at the constant of 0.88
* The concept of regularization can be applied in combination with L1 and L2 to mitigate overfitting. The current regularization was set as L1\_L2(L1=0.001, L2=0.01).



* The comparison above shows how the regularization model affects validation loss and accuracy. The effects weights are high for both training and validation sets. Both training and validation sets have proportionated loss and accuracy.
* Chart

  Description automatically generatedDrop-out is another method of regularization to control the overfitting of data.Chart

  Description automatically generated
* From the above, drop-out is effective concerning overfitting. The validation accuracy shows an increase of 0.8923.
* Since earlier models showed improvement with L2 and dropout regularization, we applied L2 and dropout together to check the improvement of the model. The application of both regularizations doesn’t show any significant performance improvement, as the model's accuracy is constant at 0.88.

**Conclusion:**

Upon comparison of the model using different parameters, the regularization method using drop-out showed improvement in accuracy over other models at 0.8923, with a loss on the validation set being 0.2708. We can use the drop-out method as a hyperparameter to tune the model to increase its accuracy of the model.