Assignment 1: Neural Network

I have used binary\_crossentropy and the MSE loss function to learn how different methodologies affect the performance of the models, but the MSE loss function is not the best option for binary classification. Although the mean square error loss function has a relatively low loss function, I do not use it because it is intended for regression problems and may not adequately capture the classification aims.

* Loss and Accuracy values for different combinations of hidden layers, activation function, dense units, regularization, and dropout values.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Using: binary\_crossentropy | | | | | | |
| **S.No** | **Hidden Layers** | **Activation function** | **Dense Units** | **Regularization** | **Dropouts** | **Loss and Accuracy** |
| 1 | 2 | relu | 16 | None | None | 0.3109, 0.8775 |
| 2 | 3 | relu | 16 | None | None | 0.2999, 0.8824 |
| 3 | 1 | relu | 16 | None | None | 0.2830, 0.8870 |
| 4 | 2 | relu | 32 | None | None | 0.3060, 0.8804 |
| 5 | 3 | relu | 32 | None | None | 0.3216, 0.8769 |
| 6 | 1 | relu | 32 | None | None | 0.2916, 0.8838 |
| 7 | 2 | relu | 64 | None | None | 0.3323, 0.8744 |
| 8 | 3 | relu | 64 | None | None | 0.3379, 0.8701 |
| 9 | 1 | relu | 64 | None | None | 0.2986, 0.8828 |
| 10 | 2 | tanh | 16 | None | None | 0.3425, 0.8709 |
| 11 | 3 | tanh | 16 | None | None | 0.3431, 0.8718 |
| 12 | 1 | tanh | 16 | None | None | 0.2791, 0.8882 |
| Using: mse loss function | | | | | | |
| 13 | 2 | relu | 16 | None | None | 0.0867, 0.8831 |
| 14 | 2 | tanh | 16 | None | None | 0.1354, 0.8291 |
| 15 | 3 | tanh | 16 | None | None | 0.1055, 0.8631 |
| 16 | 1 | tanh | 16 | None | None | 0.0843, 0.8855 |
| Using: binary\_crossentropy | | | | | | |
| 17 | 2 | relu | 16 | L1= 0.005, L2= 0.01 | None | 0.3015, 0.8794 |
| 18 | 2 | relu | 32 | L1= 0.005, L2= 0.01 | None | 0.3271, 0.8752 |
| 19 | 2 | relu | 16 | L1= 0.001, L2= 0.02 | None | 0.2935, 0.8838 |
| 20 | 2 | relu | 16 | L1= 0.001, L2= 0.01 | None | 0.2999, 0.8813 |
| 21 | 2 | relu | 16 | L1= 0.001, L2= 0.02 | 0.5 | 0.2958, 0.8817 |
| 22 | 2 | relu | 16 | L1= 0.001, L2= 0.02 | 0.1 | 0.2931, 0.8842 |
| 23 | 2 | relu | 16 | L1= 0.001, L2= 0.02 | 0.8 | 0.3030, 0.8812 |
| 24 | 2 | relu | 16 | None | 0.5 | 0.2866, 0.8869 |

**Summary:**

* In general, models with one hidden layer outperform those with two or three hidden layers. The greatest accuracy values are 0.8870 and 0.8882, respectively, for Configuration 3 (1 hidden layer with 16 units and relu activation) and Configuration 12 (1 hidden layer with 16 units and tanh activation).
* When both the tanh and relu functions are utilized, the relu function's loss function is significantly lower than the tanh functions.
* Increasing the number of dense units in the hidden layers doesn't consistently lead to better performance. In some cases, models with fewer units achieve better accuracy.
* Configuration 24 attains highest accuracy 0.8869 but there is not much difference between all the different combinations.
* Regularization methods like L1 and L2 are useful tools for managing model complexity, avoiding overfitting, and enhancing a model's capacity for generalization. Below are the graphs for L1= 0.001, L2= 0.02

A graph with blue dots

Description automatically generatedA graph with blue lines and dots

Description automatically generated

* Below are the graphs for L1= 0.001, L2= 0.01

A graph with blue dots

Description automatically generated A graph with blue dots and white text

Description automatically generated

* Dropout is a useful tool that introduces randomness during training and builds an ensemble of subnetworks inside the same model to reduce overfitting, enhance generalization, and strengthen neural networks. Below is the graph for dropout 0.5

A graph with blue dots

Description automatically generated A graph with blue dots

Description automatically generated

* Below is the graph for the combination of L1= 0.001, L2= 0.02 and dropout 0.5

A graph with blue dots

Description automatically generated A graph of a training curve

Description automatically generated with medium confidence

By comparing all the different configurations, we can conclude that dropout 0.5 has the highest accuracy of 0.8869 and lowest loss function of 0.2866. Hence, we can use the dropout function as hyperparameter. Finally, we have used the following parameters.

* Dropout function: 0.5
* Hidden Layers: 0.2
* Dense Unit: 16
* Activation function: relu
* Loss Function: binary\_crossentropy
* Optimizer: rmsprop
* Metrics: accuracy