**Summary:**

In addressing the problem, I first focused on understanding the significance of the Keras Sequential model, which functions as a layer stack for building neural networks. It was essential to import essential modules like layers, Dense, Dropout, and Regularizers from TensorFlow.keras to design the neural network.

I conducted experiments with neural networks comprising 2, 3, and 6 layers, each with varying numbers of hidden neurons (16, 64, and 64, respectively) to assess their performance. An important finding was that irrespective of the number of stacked layers, performance tended to plateau once it reached a certain threshold.

The Sequential model is initialized with model = keras.Sequential(), establishing the structure that includes input, hidden, and output layers. Incorporating a hidden layer with 64 dense units and utilizing the tanh activation function (model.add(Dense(64, activation="tanh"))) signifies the establishment of 64 neurons in the layer to process vector data.

The Dropout layer (model.add(Dropout(0.5))) is essential for mitigating overfitting by randomly excluding neurons. Setting it to 0.5 means that 50% of the neurons will be excluded.

Despite trying L1 and L2 regularizers, they didn't notably improve performance and might have even caused a decline in performance. This indicates that the model might have reached saturation, with the highest validation accuracy leveling off around 86-87%.

Replacing binary\_crossentropy with mean squared error (MSE) for loss evaluation led to enhanced performance metrics, as MSE produced a lower validation loss compared to binary\_crossentropy.

ReLU was favored over sigmoid and tanh as the activation function due to its capability to address the vanishing gradient issue. Nonetheless, in this specific scenario, tanh exhibited a performance similar to ReLU.

|  |  |  |
| --- | --- | --- |
| Combinations | Training accuracy | Validation accuracy |
| 2 dense layers  16 hidden units  Tanh activation function  Optimizers = adam | 99 | 87 |
| 3 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Regularizers  Optimizers = adam | 98 | 86 |
| 6 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Optimizers = adam  regularizers | 98 | 86 |