**Summary:**

My approach to addressing the problem involved initially understanding the significance of the Keras Sequential model, which serves as a stack of layers for building neural networks. Importing crucial modules like layers, Dense, Dropout, and Regularizers from TensorFlow.keras was essential for designing the neural network.

I experimented with implementing neural networks containing 2, 3, and 6 layers, each with different numbers of hidden neurons (16, 64, and 64 respectively) to evaluate their performance. A noteworthy observation was that regardless of the number of layers stacked, the performance tended to plateau once it reached a certain threshold.

The initialization of the Sequential model with model = keras.Sequential() sets up the structure comprising input, hidden, and output layers. Adding a hidden layer with 64 dense units and employing the tanh activation function (model.add(Dense(64, activation="tanh")) indicates the creation of 64 neurons in the layer to process vector data.

The Dropout layer (model.add(Dropout(0.5))) plays a crucial role in addressing overfitting by randomly dropping out neurons. Specifying 0.5 implies dropping out 50% of the neurons.

Despite experimenting with L1 and L2 regularizers, they didn't significantly enhance performance and may have even led to performance deterioration. This suggests that the model might have become saturated, with the best validation accuracy plateauing around 86-87%.

Substituting binary\_crossentropy with mean squared error (MSE) for loss evaluation resulted in improved performance metrics, with MSE yielding a lower validation loss compared to binary\_crossentropy.

ReLU emerged as the preferred activation function over sigmoid and tanh due to its ability to mitigate the vanishing gradient problem. However, in this particular context, tanh performed similarly to ReLU.

|  |  |  |
| --- | --- | --- |
| Combinations | Training accuracy | Validation accuracy |
| 2 dense layers  16 hidden units  Tanh activation function  Optimizers = adam | 99.63 | 87.01 |
| 3 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Regularizers  Optimizers = adam | 98 | 86.73 |
| 6 dense layers  64 hidden units  Tanh activation function  Dropouts (0.5)  Optimizers = adam  regularizers | 98.75 | 86.12 |