Accuracy of test set:

1. Without dropout or regularization:

relu	sigmoid	tanh
0.758	0.765	0.769

2. With dropout (0.4 applied on hidden layer) only:

relu	sigmoid	tanh
0.788	0.792	0.787

3. With L2-Norm (0.001) regularization only:

relu	sigmoid	tanh
0.5	0.767	0.751

4. With both dropout and regularization:

relu	sigmoid	tanh
0.5	0.763	0.720

Relu vs Sigmoid vs tanh:

In three out of four conditions above, we see that sigmoid function performs better than relu and tanh.

Relu: A nonlinear function ranging from 0 to infinity.

F(x) = 0, if x<=0

F(x) = x, if x>0

Sigmoid: A non-linear function ranging from 0 to 1. Output is predicted based on the probability (higher or lower than the threshold)

Tanh: A non-linear function ranging from -1 to 1. The negative values are mapped to -1 rather than clustering around zero in sigmoid

Intuition behind better accuracy for sigmoid:

In binary classification with only two outputs, Relu may not the most appropriate function to use as it ranges from 0 to infinity and does not take into consideration the negative values. The model is not too complex to use Relu.

With tanh, we do not see a huge difference from sigmoid, however given the fact that this is a binary classification, predicting the class based on their probability suits the best and tends to give a better accuracy in the holdout set.

Dropout:

Dropout is a technique of removing random neurons during the training to overcome co-adaptation (One neuron being too dependent on other neurons and become specialized for the training set causing overfitting of model).

The neural network gives an accuracy of 76%(sigmoid) without the dropout and 79%(sigmoid) with dropout, on the holdout data set. There is a significance increase of 3% after 40% of randomly selected neurons were removed.

Intuition: The intuition behind this could be that once few neurons were dropped, the neurons dependent on the dropped neurons became independent causing a more generalized model for the unseen holdout set, resulting in an improved accuracy in the test set.

L2-Norm Regularization:

Regularization, like dropout is a technique to minimize the generalization error and prevent the model from becoming overfitting. A model with a larger capacity and complexity tends to learn training data way too well to create a more generalized model for the unseen dataset.

One way to encounter this problem is to constrain the weights of the model that prevent the capacity of the model to learn more and more about the training dataset.

However, in the neural network above, there is no significant increase in the accuracy after adding L2 Norm regularization. The performance has gone worse for relu.

Intuition: One reason for this could be because, our feed forward neural network is already very simple with just one hidden layer. There is not much of capacity or complexity to be removed. So, by doing a regularization we might have made the model to under-learn, resulting in a bad accuracy.

Similarly, by combining both dropout and regularization (similar techniques to prevent overfitting), we might have caused the model to under-fit resulting in a poorer accuracy.

Dropouts on Validation set:

dropout	relu	sigmoid	tanh
0.3	0.788	0.794	0.791
0.2	0.782	0.788	0.79
0.4	0.791	0.796	0.789
0.5	0.794	0.795	0.786
0.6	0.796	0.795	0.785
0.7	0.792	0.792	0.782
0.8	0.787	0.788	0.781
0	0.763	0.768	0.772

References:

https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/