SUMMARY:

1. Two hidden layers were used. Examine how applying one or three hidden layers impacts test accuracy and validation.

Certainly! The depth of the network, which represents the number of hidden layers in a neural network, has a great influence on the performance of the learning process and the network's ability to generalize on new data. The model is less flexible and likely to fail at identifying fine-grained features of input data, which leads to the growth of error rate on the test sample and the decrease of the model's validation performance at the same time. Raising it to two hidden layers provides a better balance where the network can learn more details about the particular features used in the representation learning process which in many cases leads to better performance compared to the case of a single hidden layer network. Three discreet layers on the other hand, offer even higher ability to learn the interrelationships but at the same time come with the potential to overfit in case of small datasets. Thus, the number of nodes in the hidden layers depends on the complexity and volume of the data as well as the desired trade-off between the model's complexity and its ability to generalize on the training and validation stages of model creation.

2. Consider utilizing layers with 32, 64, and so on hidden units, or layers with fewer or more hidden units.

The amount of hidden units in each layer within architecting structures for neural networks is what define their ability to learn and generalize. Reducing the number of hidden units mitigates issues with overfitting, and may also lead to reduced computational load, especially with a small number of samples, or with hardware constraints. On the other hand, having more hidden units enhances the local complexity of the model and would possibly enhance the training error but degrades the generalization ability of the model. But it also poses the danger of overfitting in case the model is not well constrained by regulatory measures such as dropout or early stopping. Estimating the number of hidden nodes is one of the most crucial steps, which depends on the features of the given data set, the capacity of the computer, and the selected ratio between the complexity of constructed NN model and its ability to generalize in the context of training, validation, and test data sets.

3. Attempt to substitute the MSE loss function for Binary_crossentropy.

Replacing the Mean Squared Error (MSE) for the Binary Cross-Entropy in binary classification problems affects the process and results of neural network training. MSE is mostly applied where the task is to make predictions of numerical values: continuous variables are the target value in

regression analysis whereas Binary Cross-Entropy is developed to work with binary variables: probability in binary classification tasks. The reason is MSE might slow down the convergence rate when applied to binary classification and could possibly provide worse model evaluations than Binary Cross-Entropy since the latter is better suited to penalize deviations in predicted probabilities. To conclude, it is crucial to emphasize that the purpose of comparing the proposed MSE with Binary Cross-Entropy comes from the necessary empirical investigation of which loss function best complements the model's performance and convergence when applied to different datasets and binary classification problems.

4. Instead of relu, consider utilizing the tanh activation, which was well-liked in the early days of neural networks.

By replacing ReLU (Rectified Linear Unit) activation function, that has dominated the current networks architectures, by a Hyperbolic Tangent function that has been used in the early days of Artificial Neural Networks, different characteristics are introduced to the model. Instead, Tanh outputs values ranging from -1 to 1, thus preserving the 0 centered data distribution, and can possibly favor learning intricate patterns by means of the negative values. However, tanh can be problematic at times because it decreases gradient during backpropagation for very large or small input data. Further, it would be important to compare the performance of the newly proposed activation function against ReLU in current architectures of neural networks and analyse the aspects like the speed of convergence and accuracy, to establish whether the use of the new activation function would prove beneficial and whether it would solve the issues of the vanishing gradients or not depending on datasets and tasks at hand.

5. Make use of any technique we reviewed in class to improve the model's validation performance, such as dropout and regularization.

To optimize the validation accuracy of the model other tools such as regularization for the dropout can be applied proficiently. Dropout randomly drops out a few neurons in the training process, which helps avoid fitting neurons together and thus, overfitting. Thus, by adding dropout layers between the dense layers in the neural network architecture, it is possible to increase the generalization capacity on the validation data. Besides, other techniques of weight space limitation, for example, L1 or L2 regularization can be used to fine tune weights in order to reduce their value and thus the capacity of the model, which in turns enhance performance on viability data sets. The use of all these techniques and a close watch on the validation metrics assists in optimizing the model's generalization and overall performance on classification.