CS5613: Neural Networks

Assignment 01

T.M.B.S.B Thennakoon (219412A)

Question 01

- There is no “universal method” to determine the number of hidden layers or the number of neurons per layer in a neural network. The optimum values for these two parameters depend on the nature of the problem. Complex problems like face recognitions and object detection may require more hidden layers [1].

- The selection of the hidden layers and neural networks will ultimately come down to trial and error [2].

- Neural network with a single hidden layer can be considered as universal approximator. This means that a feed-forward network, with a single hidden layer, containing a finite number of neurons, can approximate continuous functions with mild assumptions on the activation function [2].

- [3] contains several rules of thumbs for deciding the number of neurons to use in hidden layers.
  - The number of hidden neurons should be between the size of the input layer and the size of the output layer
  - The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.
  - The number of hidden neurons should be less than twice the size of the input layer.
- [4] suggests designing automated search with different network configurations.
  Random: Try random configurations of layers and nodes per layer.
  Grid: Try a systematic search across the number of layers and nodes per layer
  Heuristic: Try a directed search across configurations such as a genetic algorithm or Bayesian optimization
  Exhaustive: Try all combinations of layers and the number of nodes; it might be feasible for small networks and datasets.

- Using hyperparameter optimization techniques such as Bayesian optimization, Tree-structured Parzen estimator (TPE), Hyperband is good way to find the best fit model [5].

Question 02

Network depth: 02 layers

Activation Functions (from I/P to O/P): [relu, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,1]	Binary Cross Entropy	0.7	0.397
	Mean Square	0.707	0.393
[30,1]	Binary Cross Entropy	0.708	0.39
	Mean Square	0.702	0.396

Network depth: 02 layers

Activation Functions (from I/P to O/P): [tanh, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,1]	Binary Cross Entropy	0.703	0.391
	Mean Square	0.697	0.398
[30,1]	Binary Cross Entropy	0.698	0.393
	Mean Square	0.698	0.393

Network depth: 03 layers

Activation Functions (from I/P to O/P): [relu, relu, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,10,1]	Binary Cross Entropy	0.722	0.405
	Mean Square	0.715	0.415
[15,25,1]	Binary Cross Entropy	0.705	0.43
	Mean Square Error	0.619	0.417
[30,15,1]	Binary Cross Entropy	0.612	0.412
	Mean Square Error	0.625	0.408

Network depth: 03 layers

Activation Functions (from I/P to O/P): [tanh, tanh, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,10,1]	Binary Cross Entropy	0.709	0.388
	Mean Square	0.71	0.39
[15,25,1]	Binary Cross Entropy	0.694	0.4
	Mean Square Error	0.689	0.402
[30,15,1]	Binary Cross Entropy	0.695	0.392
	Mean Square Error	0.701	0.398

Network depth: 04 layers

Activation Functions (from I/P to O/P): [relu, relu, relu, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,20,10,1]	Binary Cross Entropy	0.654	0.46
	Mean Square	0.632	0.409
[15,20,30,1]	Binary Cross Entropy	0.738	0.46
	Mean Square Error	0.606	0.435
[15,10,5,1]	Binary Cross Entropy	0.722	0.387
	Mean Square Error	0.716	0.433
[15,15,15,1]	Binary Cross Entropy	0.627	0.436
	Mean Square Error	0.726	0.401

Network depth: 04 layers

Activation Functions (from I/P to O/P): [tanh, tanh, tanh, sigmoid]

Optimizer: adam

Neurons in Each layer (I/P to O/P)	Loss function	F1 -Score (average over 5 folds)	Loss (average over 5 folds)
[15,20,10,1]	Binary Cross Entropy	0.723	0.408
	Mean Square	0.717	0.406
[15,20,30,1]	Binary Cross Entropy	0.71	0.383
	Mean Square Error	0.686	0.402
[15,10,5,1]	Binary Cross Entropy	0.703	0.402
	Mean Square Error	0.722	0.419
[15,15,15,1]	Binary Cross Entropy	0.722	0.398
	Mean Square Error	0.712	0.423

Question 03

The best model was obtained with following specs:

No. of hidden layers: 2

No of neurons in each layer: [15,20,30,1]
(Input to Output)

Activation Function in each layer: [relu, relu, relu, sigmoid]

Optimizer: adam

Loss Function: Binary Cross Entropy

F1_Score: 0.738

The results were in line with findings from literature. There were no straightforward rules to determine the best model, other than experimenting with various combinations.

The value for number of neurons and hidden layers will depend on the application. This dataset is not complex hence a smaller number of hidden layers and number of neurons was sufficient to achieve an acceptable level of performance in terms of F1 Score.

Sigmoid function was used as the output activation function as the task was a binary classification.

Question 04

Regularizer Term	Value	F1 -Score (average over 5 folds)
L2	0.001	0.629
L2	0.01	0.694
L2	0.1	0.286
L2	1	0.143
L1	0.001	0.709
L1	0.01	0.673
L1	0.1	0.143
L1	1	0.143

Adding of the L2, or L1 regularizations terms has not improved the F1 score, instead it has caused the F1 Score to decrease. Since the model is less complex (lower number of neurons and hidden layers), it is less likely to overfit. Therefore, this result can be expected.

Larger regularizers values to lead to drastic drop in F1 - Score. This is due to addition of more penalties make the model less complex. This hinders the model's capability to generalize over the data well.

The best lambda value we can select is 0.001 for L1.
When this lambda value used to train model with 70% percent of the whole dataset as training data F1_Score of 0.841 was obtained with best fit model.

References

[1] A. Gad, "Beginners Ask “How Many Hidden Layers/Neurons to Use in Artificial Neural Networks?”," 27 June 2018. [Online]. Available: https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e. [Accessed 31 May 2022].

[2] J. Heaton, "The Number of Hidden Layers," 01 June 2017. [Online]. Available: https://www.heatonresearch.com/2017/06/01/hidden-layers.html. [Accessed 31 May 2022].

[3] S. Krishnan, "Medium," 9 September 2021. [Online]. Available: https://medium.com/geekculture/introduction-to-neural-network-2f8b8221fbd3. [Accessed 31 May 2022].

[4] J. Brownlee, "How to Configure the Number of Layers and Nodes in a Neural Network," 27 July 2018. [Online]. Available: https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/. [Accessed 31 May 2022].

[5] A. Bissue, "Hyper-parameter optimization algorithms: a short review," 16 April 2019. [Online]. Available: https://medium.com/criteo-engineering/hyper-parameter-optimization-algorithms-2fe447525903 . [Accessed 01 June 2022].