Assignment 2.1: Building an Artificial Neural Network

Patricia Enrique

Department of Engineering, University of San Diego

AAI-511: Neural Networks and Learning

Prof. Esnaeilli

July 10, 2023

Results

The results from experimenting with different parameters in the neural network are summarized in the tables that follow. The parameters that are explored include the addition of layers in the network, different activation functions and regularization techniques, and modifying the learning rate and batch sizes. From Table 1 it can be seen that the performance of the model worsens as layers are added. Between the base model and the model with one additional layer, the performance metrics remain similar.

Table 1

Exploration of model architecture

Performance Metric	Base Model	Addition of 1 layer	Addition of 2 layers	Addition of 3 layers
Loss	0.6366	0.8474	2.0386	6.0250
Accuracy	0.7450	0.7237	0.5025	0.4900
Precision	0.8091	0.8006	0.6562	0
Recall	0.6544	0.6103	0.0515	0
AUC	0.8089	0.8090	0.5246	0.5085

Table 2 summarizes the results from experimenting with different activation functions. The base model with the rectified linear unit (ReLu) activation outperforms the other methods with similar results as the linear activation function.

Table 2

Exploration of activation functions

Performance Metric	Base Model (ReLu)	Exponential	Sigmoid	Linear
Loss	0.6366	7.1247	1.8318	1.6205
Accuracy	0.7450	0.5425	0.5050	0.7462
Precision	0.8091	0.5517	0.5169	0.8622
Recall	0.6544	0.5490	0.4485	0.5980
AUC	0.8089	0.5409	0.5150	0.8283

Table 3 summarizes the results from experimenting with different regularization techniques. The base model contains no regularization and is only advantageous when comparing loss. The three regularizations introduced have similar performances and apply both L1 and L2 regularization penalties.

Table 3

Exploration of activation functions

Performance Metric	Base Model	Kernel_regularizer	bias_regularizer	activity_regularizer
Loss	0.6366	7.5670	2.1505	2.4998
Accuracy	0.7450	0.7837	0.8163	0.7500
Precision	0.8091	0.8529	0.9196	0.8399
Recall	0.6544	0.6961	0.7010	0.6299
AUC	0.8089	0.8726	0.9036	0.8429

Table 4 summarizes the results from experimenting with different learning rates. As the learning rate grows, the performance metrics improve until a learning rate of 1 is used which has comparable results to that of 0.001.

Table 4

Exploration of learning rates

Performance Metric	Base Model (0.001)	0.01	0.1	1
Loss	0.6366	2.6364	0.4097	1.4914
Accuracy	0.7450	0.7025	0.9038	0.7563
Precision	0.8091	0.8571	0.9636	0.8610
Recall	0.6544	0.5000	0.8431	0.6225
AUC	0.8089	0.7837	0.9586	0.8163

Table 5 summarizes the results from experimenting with different batch sizes. The performance of the model improves as the batch size grows; however, the results from batch size 5 and larger are comparable and do not provide a significant improvement.

Table 5

Exploration of batch sizes

Performance Metric	Base Model (32)	1	5	10	25	50	100
Loss	0.6366	0.7479	0.3292	0.2596	0.2022	0.2228	0.3135
Accuracy	0.7450	0.8938	0.9513	0.9600	0.9600	0.9563	0.9613
Precision	0.8091	0.9425	0.9719	0.9845	0.9772	0.9722	0.9821
Recall	0.6544	0.8431	0.9314	0.9363	0.9436	0.9412	0.9412
AUC	0.8089	0.9571	0.9849	0.9908	0.9921	0.9874	0.9897

The final model uses the trends found from the parameter exploration to optimize the performance. The model consists of three layers with 100, 50, and 1 units respectively, a rectified linear unit activation function, and both L1 and L2 regularization penalties. The learning rate and batch size with the best performance were 0.01 and 10 respectively. The training and testing performances are summarized in Table 6.

Table 6

Final model performance metrics

Performance Metric	Train	Test
Loss	2.2762	0.5900
Accuracy	0.9225	0.9650
Precision	0.9753	0.9885
Recall	0.8701	0.9348
AUC	0.9612	0.9936

Discussion

The performance metrics remain consistent between the training and the testing dataset indicating that the model is not overfitting or underfitting to the data it is trained on. The accuracy for the model is above 90% for both the training and the testing dataset indicating that the model is able to generalize the predictions to combinations of features it has not yet seen. The precision is similar between training and testing, meaning that when predicting that a user will click on an ad the model is correct about 98%

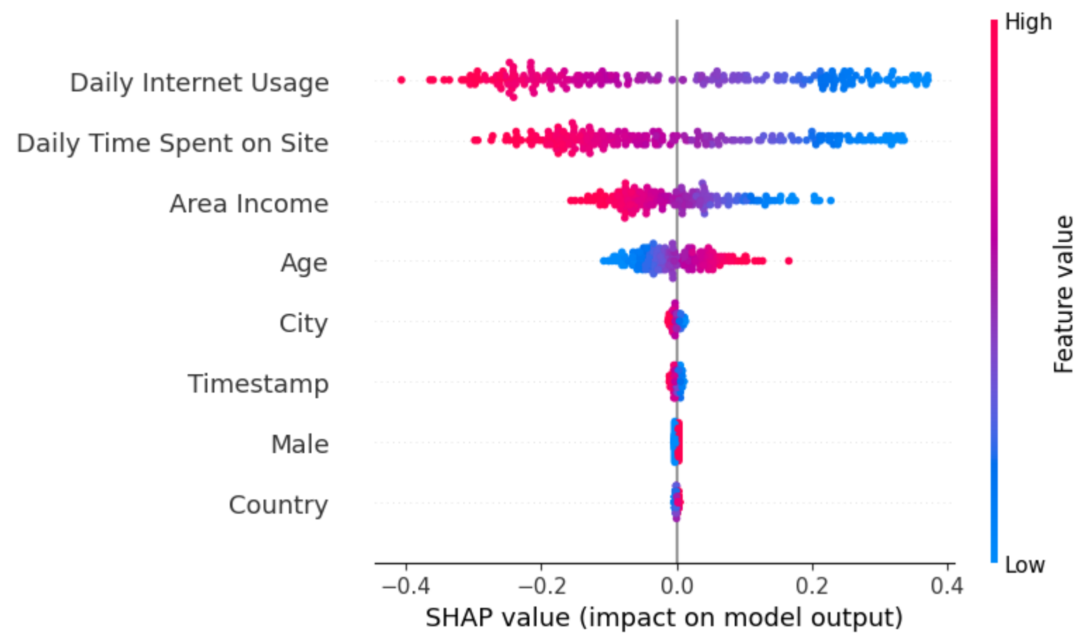
of the time. The recall metric has the biggest discrepancy and worst performance between the training and testing dataset as the model has a difficult time correctly identifying actual positives. On the other hand, the model is able to differentiate between an instance that will result in an ad click versus one that will not with a high probability as shown by the AUC score.

Insights From Analysis

SHAP is used to visualize the importance of the features on the model predictions. Figure 1 illustrates that the daily internet usage and daily time spent on the site are the two most important factors when predicting whether a user will click on an online ad (SHAP, 2023).

Figure 1

SHAP visualization of the feature importance



The neural network achieved good results after parameter tuning. The performance of the final model is compared to a basic gradient boosting classifier with minimal parameter tuning in Table 7. The classifier is able to achieve similar results with less modifications. From the classifier, Figure 2 is generated illustrating that both models identify the same top four features of most importance.

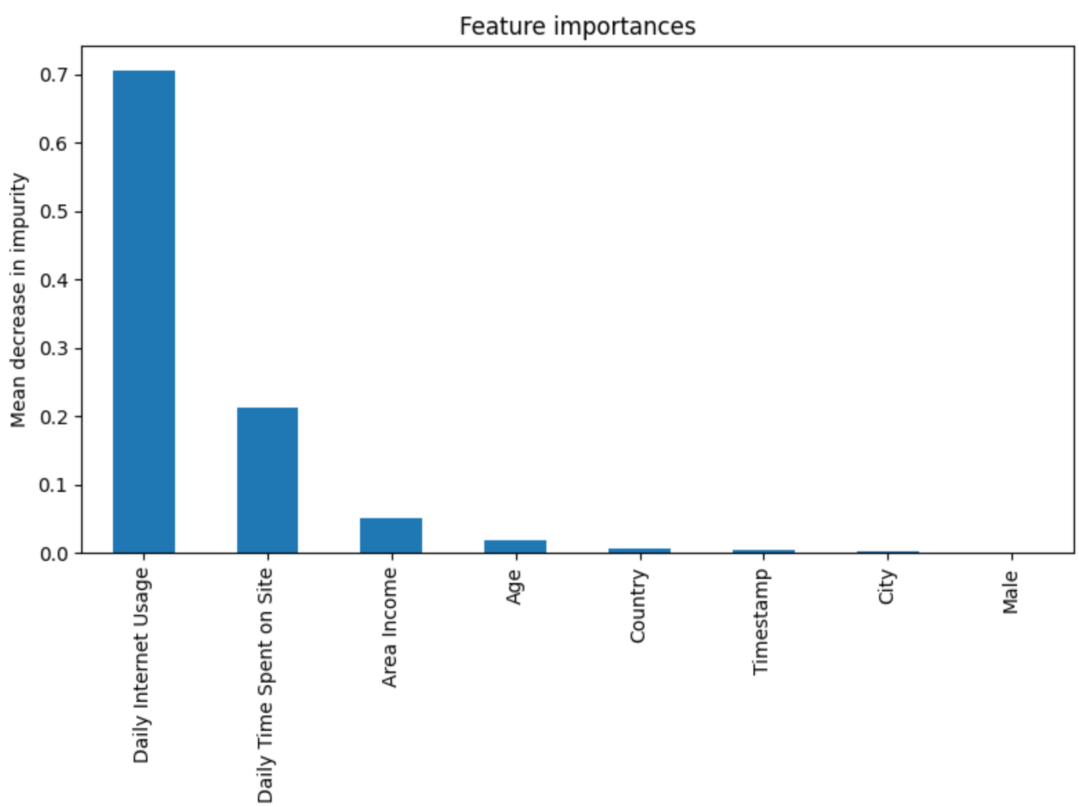
Table 7

Neural network and gradient boosting classifier comparison

Performance Metric	Neural Network	Gradient Boosting Classifier
Accuracy	0.97	0.96
Precision	0.99	0.96
Recall	0.93	0.96
AUC	0.99	0.96

Figure 1

Visualization of the feature importance from the gradient boosting classifier



Future Suggestions

The parameters explored in this analysis are a few of the ones available. The performance of the model is acceptable and does not require in depth parameter tuning to obtain. Other areas that can be further explored are the effect of the units used. In this model the first layer contains 100 units which is chosen arbitrarily, and each layer afterwards has half the number of units. A different starting number as well as the amount each layer decreases by can be further analyzed to obtain better results. Additionally, the optimizer Adam is used for all the model exploration. Other functions such as AdamW, Adafactor, or Lion have the possibility of achieving improved results.

References

SHAP. (2023, July 8). Welcome to the SHAP documentation. https://shap.readthedocs.io/en/latest/index.html