Analysis of Multi-Activation Layers in Neural Network Architectures

Jaskirat Sohal

Braden Stonehill
Kennesaw State University

Analysis of Multi-Activation Layers in Neural Network Architectures

Braden Stonehill and Jaskirat Sohal

Department of Computer Science, Kennesaw State University

Marietta, Georgia

ABSTRACT

A common practice for developing a Neural Network architecture is to build models in which each layer has a single activation function that is applied to all nodes uniformly. This paper explores the effects of using different activation functions to analyze the effects of architectures' ability to generalize datasets under new conditions. The approach will be tested on a fully connected neural network and compared to traditional models and an identical network with uniform activations.

INTRODUCTION

Currently, most Neural Networks developed utilize uniform activation functions partially due to the efficiency in implementation. The data transformations in a Neural Network can be accomplished through a series of matrix multiplication operations followed by an element-wise transformation using the specified activation function. This allows for concise and efficient computations, which can be essential depending on the task and training time available.

Activation functions apply non-linear transformations to the data, allowing the model to map to complex datasets. Even though typical models have uniform activation functions, they can extract different features through the different weight parameters per node.

This study analyzes the different aspects and effects of applying different activation functions, including generalization, training time, underfitting or overfitting, and performance. As this approach applies nonregularity to the layers, the model complexity should increase overall performance and generalization at the expense of training time and a higher risk of overfitting.

BACKGROUND AND RELATED WORK

Activation functions are vital for developing Neural Networks, allowing the model to map itself to complex data. Without activation functions, the network would only apply linear transformations, often insufficient for adequately separating or predicting data. The study of activation functions is broad, resulting in many novel functions and improvements to existing functions, such as Parametrized ReLU, which prevents the gradient from being zero for negative values resulting in better performance over ReLU [2].

In a typical design, neural networks often apply uniform activation functions throughout the entire network except for the output layer which is tailored to the specific task. One study explored the effects

of using different activation functions for each hidden layer of a network and compared the performance to uniform networks. The results indicate that the multiple activation function network outperformed the uniform networks and traditional machine learning models [3]. Our study expands upon this approach and analyzes the performance of numerous activation functions per layer.

METHODOLOGY

The proposed approach will be tested through a fully connected Neural Network containing three and six layers containing the following node counts: 15, 10, 5 for three layers and 15, 15, 10, 10, 5, 5 for six layers. The network will start with three nodes for Sigmoid, ReLU, Tanh, Swish, and ELU activation functions and decrease to one node per activation function before reaching the output layer. The two variations will be created to adequately test the network by preventing overfitting if the data complexity of the datasets is too low.

The architecture will be tested on several medical classification datasets, with the performance metrics, execution time, and loss being measured and analyzed. As most medical datasets have a small number of data samples, an autoencoder network will generate synthetic data to provide multiple training samples. The performance of each model will be evaluated with and without synthetic data to determine if synthetic data would have detrimental effects on the models.

EXPERIMENTAL EVALUATIONS

The initial dataset used for this paper contained categorical, numerical, and categorical attributes represented as numerical. ID and age were represented as numerical data. Gender, marital status, work type, residence type, and smoking status were represented as categorical data, while Hypertension and Heart Disease, which were categorical data, were represented as numerical values. The dataset was unbalanced and contained missing values. The dataset contained results and evaluations of fifty-nine percent of females and forty-one percent of males.

The ages ranged from 0.08 to 82, with a median of 43.2. Upon further investigation, it was found that every attribute available had some form of imbalance. Hypertension showcased that Ninety percent of the patients did not have hypertension. Ninety-four percent of the patients did not have any heart diseases. Sixty-six percent of the patients had never been married. Fifty-seven percent of the patients worked a private job, sixteen percent were self-employed, and others never worked, cared for children, or had government jobs. The residency type was nearly evenly split between fifty-one percent urban and forty-nine percent rural. Lastly, ninety-five percent of the data showed that none of the patients had a stroke. To clean and normalize the data, all the categorical values were assigned a numerical value to convert the entire dataset to numerical attributes to simplify the process for normalization, which in turn would make it easier to train for all machine learning methods, as they would only have to deal with a single data type. Any four percent of

the BMI dataset that had the value of NaN or N/A were removed during cleaning. The categorical values that were converted to numerical are shown below:

Attributes	Categorical	Numerical
Gender	Female	0
	Male	1
Ever_Married	No	0
	Yes	1
Work_type	Never_Worked	0
	Private	1
	Govt_Job	2
	Children	3
	Self-Employed	4
Residence_type	Rural	0
	Urban	1
Smoking_Status	Never smoked	0
	Formerly smoked	1
	Smokes	2
	Unknown	3

Figure 1: Categorical Attributes to Numerical Conversion

The ID, which served as a unique ID in the dataset, represented different patients. As the patient IDs should not affect the final classification of stroke based on the symptoms available, it was removed from the final dataset used for training and testing.

During the preliminary testing of the dataset, multiple machine learning methods were used, such as Decision Trees, Gaussian Naïve Bayes, K Nearest Neighbor, Linear Support Vector Machine, Logistic Regression, Random Forest, and SVM. The models ranged from simple to complex to test the dataset's complexity. As Gaussian Naïve Bayes relies on the dataset having properly distributed information to produce an accurate result, it is surprising that even though it had the lowest accuracy, it could detect most cases where the patient did have a stroke. While the remaining methods could predict if the patient did not have a stroke accurately, they had multiple cases where the patient was incorrectly diagnosed as false negative ranging from fifty-five to sixty cases. It is also surprising that although other complex methods, such as logistic regression, linear SVM, and SVM, could accurately detect a True Negative at a higher rate, they also had the highest number of False Negatives. As this is a medically related issue, the models should aim for the lowest number of false negatives. In our case, a True Negative would be if the model predicted that the patient did not have a stroke and has no record of a stroke. While a false positive would be if the patient has had a stroke, the model has predicted that the patient has not.

Model	Accuracy	TP	TN	FP	FN
Decision Tree	91.039%	7	1337	74	55
Gaussian Naïve Bayes	87.169%	22	1262	149	40
KNN	95.723%	2	1408	3	60
Linear SVM	95.791%	0	1411	0	62
Logistic Regression	95.859%	1	1411	0	61
Random Forest	95.655%	1	1408	3	61
SVM	95.791%	0	1411	0	62

Figure 2: Models, Accuracy, and Confusion Matrix

Our future testing and data cleaning entails creating new data based on readily available data to balance the current dataset, testing neural networks based on different parameters, and seeing how specific symptoms affect the patient's health. Future testing will also include creating a new neural network architecture with varying activation functions per node

per layer to test how different activation functions may affect different parameters and applying the new model to other possible medical conditions that may share similar symptoms.

REFERENCES

[1] Ding, Bin, Huimin Qian, and Jun Zhou. "Activation functions and their characteristics in deep neural networks." *2018 Chinese control and decision conference (CCDC)*. IEEE, 2018.

[2] Sharma, Sagar, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." *Towards Data Sci* 6.12 (2017): 310-316.

[3] Vijayakumar, K., Vinod J. Kadam, and Sudhir Kumar Sharma. "Breast cancer diagnosis using multiple activation deep neural network." *Concurrent Engineering* 29.3 (2021): 275-284.