Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 9, September 2024

Uncovering the Factors Behind 100% Accuracy in the Inner Speech Recognition Dataset

Corresponding Author: B Hrulekh Nandan
VIT-AP University
Amaravathi, India

Corresponding Author: J Ashrith Sai
VIT-AP University
Amaravathi, India

Corresponding Author: R Vyshnavi
VIT-AP University
Amaravathi, India

Corresponding Author: Nagaraju Devarakonda
VIT-AP University
Amaravathi, India

Corresponding Author: B Sai Ram
VIT-AP University
Amaravathi, India

***Abstract*—our research journey aims to improve Inner Speech Recognition but we made a surprising unexpected error that changed the journey. We want to improve the accuracy of the RNN model in the Inner Speech Recognition project. but, we got an accuracy of 100%, which is practically impossible.so, we researched to understand why our RNN (Recurrent Neural Network) model achieved 100% accuracy by using two important parameters such as "Gini" and "Entropy". we have 10 subjects, and out of them we will choose the best one, based on parameters like the Gini Index and entropy, and perform analysis using gradient descent concepts on the chosen one.**

I. INTRODUCTION

In the ever-changing field of research, our journey often takes unexpected turns, leading to discoveries and insights. Our study on Inner Speech Recognition within Brain-Computer Interface (BCI) technology is a perfect example. We started with the goal of improving existing methods but soon found ourselves facing the surprising issue of achieving 100% accuracy in our RNN model.

As we looked deeper into our dataset and model design, we found a puzzling issue: our RNN model was performing perfectly, which is often a sign of overfitting or other problems. Not discouraged by this unexpected result, we set out to understand the root causes and find new solutions. We used methods like Gini impurity, entropy analysis, and gradient descent to work through the complexities, each step bringing us closer to solving the problem. Through careful experiments and thorough investigation, we discovered several key factors that contributed to this issue. First, we examined our data splits to ensure there was no data leakage between the training and testing sets. We also scrutinized our data preprocessing steps to make sure they were applied consistently. Additionally, we explored our model architecture and hyperparameters, considering the possibility of overfitting.

These meticulous steps gave us valuable insights that reshaped our research direction. Our journey highlights the unpredictable nature of scientific research, where challenges often lead to innovation and unexpected discoveries.

In this paper, we provide a detailed account of our research journey. We start with our initial goals and motivations, describe the obstacles we encountered—such as the perfect accuracy of our RNN model—and explain the methods we used to overcome these issues. Finally, we share the conclusions we reached and the new directions our research has taken. By sharing our experiences and insights, we hope to contribute to the growing field of Inner Speech Recognition within BCI technology, highlighting the ongoing nature of scientific exploration and the power of overcoming unexpected challenges.

II. METHODOLOGY

Inspired by the research presented in 'Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition' by Nicolás Nieto, Victoria Peterson, Hugo Leonardo Rufner, Juan Esteban Kamienkowski, and Ruben Spies, we embarked on a journey to explore the potential of Recurrent Neural Networks (RNNs) in improving inner speech recognition (ISR) within the domain of Brain-Computer Interface (BCI) technology.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 9, September 2024

Acknowledging the contributions of Nieto et al. in providing an open-access EEG-based BCI dataset, we sought to build upon their work and extend the boundaries of ISR research. While previous studies leveraged Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and EEGNet versions, our endeavor introduced the utilization of RNNs as an alternative approach.
As we explored new areas, we had two main goals: to improve the accuracy of Inner Speech Recognition (ISR) models and to understand better how neural decoding works. However, our first try with RNNs gave us an unexpected result: 100% accuracy in our predictions.

III. GINI INDEX AND ENTROPY

Having identified the need to scrutinize the underlying reasons for our unexpected 100% accuracy, we have opted to examine the Gini index, Entropy, and Gradient Descent as potential explanatory variables. Let's proceed with a detailed discussion of each:

Gini Index: In the realm of decision tree algorithms, the Gini index serves as a measure of impurity within a dataset. It quantifies the probability of misclassification by assessing the distribution of labels in a given subset. A higher Gini index indicates greater impurity, suggesting a higher likelihood of misclassification.

Entropy: Derived from information theory, entropy is a metric for the level of uncertainty or disorder within a dataset. In machine learning contexts, it quantifies the unpredictability of data points, with higher entropy values indicating greater uncertainty and lower values indicate more predictable outcomes.

we will discuss Gradient Descent in IV
By carefully examining these parameters, we aim to uncover the basic structure and features of our dataset. This will help us understand why our predictions were so accurate and provide valuable insights into the reasons behind the remarkable accuracy we observed.
We've crafted a Recurrent Neural Network (RNN) model tailored for our Brain-Computer Interface (BCI) application. RNNs are adept at handling sequential data, making them ideal for decoding brain signals over time.
Our RNN has three main parts:

1. The first layer, a SimpleRNN, consists of 64 units and takes into account the temporal aspect of our data. This layer helps capture patterns and dependencies in the EEG signals.
2. The second layer is a Dense layer with 128 units and utilizes the ReLU activation function. This layer enhances the network's ability to learn complex relationships within the data.
3. Finally, we have another Dense layer with a single unit and a sigmoid activation function. This layer is crucial for binary classification tasks, such as distinguishing between different mental states.

A. Equations

$$\text{Gini Impurity} = 1 - \sum_{i=1}^{c} p_i^2$$

p_i is the probability in class i occurring in a dataset

$$\text{Entropy} = -\sum_{i=1}^{c} p_i \log_2 p_i$$

p_i is the probability in class i occurring in a dataset

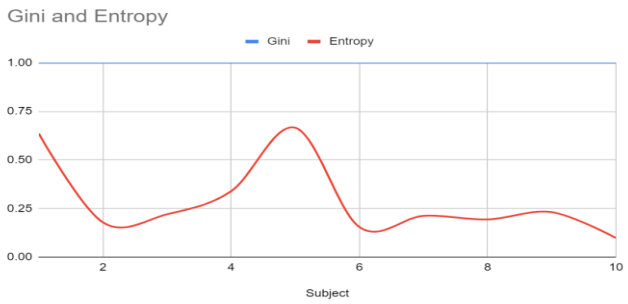
B. Gini Index and Entropy

The Gini index measures how mixed or impure a group is. It calculates the likelihood of misclassifying a randomly chosen item from the dataset if it were randomly labeled according to the distribution of labels in the subset. In the context of decision trees or classification algorithms, a lower Gini index indicates a purer node, meaning the samples within that node belong to predominantly one class. Therefore, a lower Gini index is preferred as it signifies less impurity and better separability of classes.
Entropy is like a measure of uncertainty or surprise. It quantifies the unpredictability of data points in a dataset. lower entropy values indicate less uncertainty or disorder in the dataset. It suggests that the dataset is more homogeneous, making it easier to classify. So, in general, lower entropy is desired as it implies more predictable and clearer patterns in the data

TABLE I.Measured values from the above functions

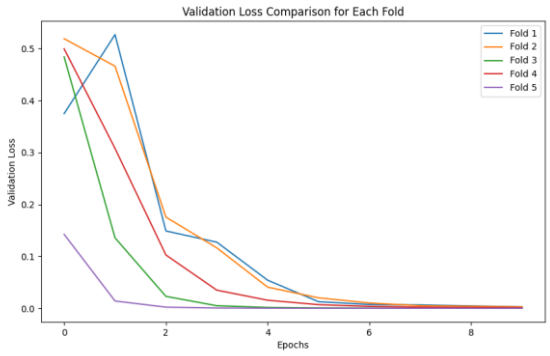
Subject	Gini	Entropy
1	0.9998800393	0.6354573114
2	0.9999950874	0.1782833283
3	0.9999884211	0.2201744629
4	0.9999784846	0.338864696
5	0.9998956833	0.6655728975
6	0.9999960587	0.1541564342
7	0.9999926958	0.2120327169
8	0.9999924974	0.193556762
9	0.9999911024	0.2310785275
10	0.9999987315	0.09821608085

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 9, September 2024



Graph: 1.01

Based on the findings in Table 1.01 and the accompanying graph 1.01, we see that the Gini index doesn't change much across the datasets, so it's not very useful for our analysis. On the other hand, the Entropy metric changes a lot, with Dataset 10 having the lowest entropy. Because of this, we choose Entropy as the main factor for evaluating the datasets, and we identify Dataset 10 as the best choice among the ten datasets analyzed.



Graph: 1.02 Validation Loss vs Epochs

Graph 1.02 occurs from the RNN model of dataset 10. it shows the validation loss for each fold in a k-fold cross-validation setup for an RNN model over 9 epochs. Here's a breakdown of what it illustrates:
Validation Loss: This is a measure of how well the model is performing on the validation set. Lower values indicate better performance.
Epochs: These are the number of times the learning algorithm has processed the entire training dataset.
Folds: In k-fold cross-validation, the dataset is divided into k subsets or folds. The model is trained k times, each time using a different fold as the validation set and the remaining k-1 folds as the training set. This graph shows the validation loss for 5 different folds.

IV. GRADIENT DESCENT

Gradient Descent is another important parameter that is used to identify the reason behind the 100% accuracy in the RNN model.

Gradient Descent: It is a method used to find the best settings for a model by making small adjustments to minimize errors.

Gradient descent is like hiking down a hill to find the lowest point. Imagine you're on a hill, and it's foggy so you can't see very far. To find your way down, you take small steps in the direction that slopes downward the most. Each step you take is based on checking which way the ground is sloping. You keep repeating this until you reach the bottom of the hill

A. Algorithm

repeats until convergence{

Gradient Descent =

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for j=0 and j=1)

}

Minimize a cost function $J(\theta_0, \theta_1)$ by iteratively adjusting the parameters θ_0 *and* θ_1.

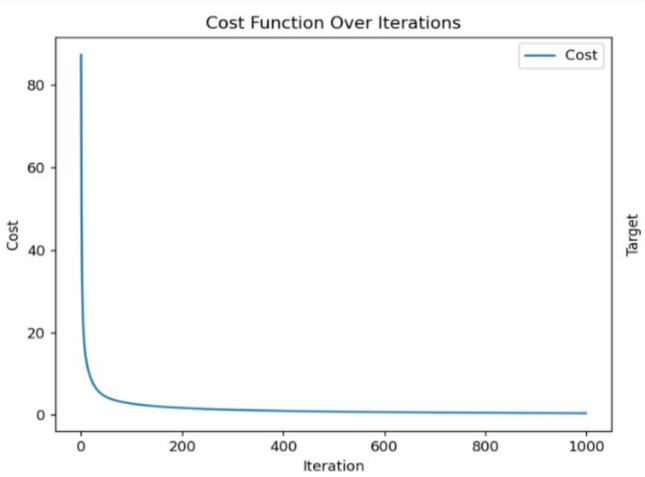
:= :- This means assignment. It implies that the current value is updated to the new value on the right-hand side of the equation.

α: The learning rate, a small positive number that controls the size of the steps taken to reach the minimum. It determines how much the parameters are adjusted in each iteration.

θ_j **:** represents the parameters (weights) of the model. Here, j can be 0 or 1, indicating different parameters.

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$: the partial derivative of the cost function J for the parameter θ_j. This is also known as the gradient. It measures how much J changes as θ_j changes. The gradient points in the direction of the steepest ascent, so moving in the opposite direction helps minimize the cost function.

B. Performance Analysis



Graph 2.1 Iteration Versus Cost

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 9, September 2024

Graph 2.1 shows a rectangular hyperbola which indicates Cost and iteration are inversely proportional.

X-axis (Iteration): This represents the number of times the gradient descent algorithm has updated the model's parameters. Each iteration involves calculating the error (cost) and adjusting the parameters to reduce that error.

Y-axis (Cost): This represents the value of the cost function, which quantifies how well the model is performing. A lower cost indicates a better fit for the data and improved model performance. The plot clearly shows that the gradient descent algorithm is working effectively.

Initial High Cost: The cost starts very high, indicating that the model is initially far from optimal in terms of fitting the BCI data.

Rapid Initial Descent: The cost drops rapidly in the first few hundred iterations. This means the algorithm is quickly finding better parameter values that significantly improve the model's performance.

Gradual Convergence: As the iterations progress, the rate of cost reduction slows down. The curve flattens out, indicating that the algorithm is approaching a minimum point where further improvements become marginal.

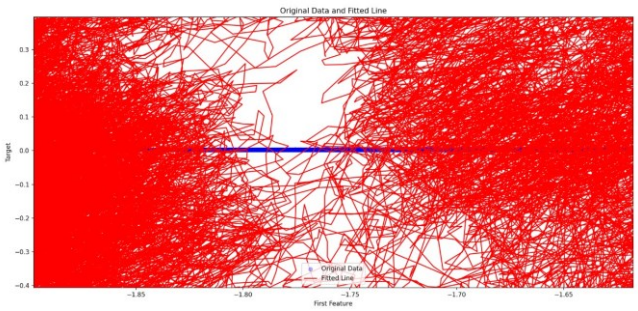
Potential Convergence: Around iteration 1000, the cost appears to have almost leveled off, suggesting that the algorithm may have converged to a solution. This means that further iterations are unlikely to lead to significant improvements in model performance.

Convergence: The flattening curve suggests that it likely found a good set of model parameters. Further iterations might not be necessary.

Local Minimum: While the plot shows convergence, it's important to be aware that gradient descent can sometimes get stuck in local minima (good solutions, but not the absolute best).

Data Overfitting: The cost continues to decrease very slowly over many iterations, which might indicate that the model is overfitting the training data.

C. Linear Regression



Graph 2.2 First Feature Versus Target

Data Points (Blue): The blue dots represent individual data points from the dataset. Each point shows a specific value of the "First Feature" and its corresponding "Target" value. the widespread and seemingly random distribution of the blue dots indicates that the "First Feature" doesn't have a strong linear relationship with the "Target." There's a lot of variability in the "Target" values for a given value of the "First Feature."

Fitted Line (Red): The red line results from fitting a linear regression model to the data. This model aims to find the best-fitting straight line that describes the relationship between the "First Feature" and the "Target".The nearly horizontal nature of the red line suggests that the linear regression model doesn't find a significant linear relationship between the two variables. In other words, changes in the "First Feature" don't lead to predictable changes in the "Target."

Weak Linear Relationship: The graph demonstrates that there's no strong linear relationship between the "First Feature" and the "Target" variable. The linear regression model struggles to explain the variability in the "Target" based on the "First Feature".

RESULTS AND DISCCUSSION

Our analysis revealed that dataset 10, with the lowest entropy, was the most suitable for further investigation. Gradient descent analysis indicated a rapid initial decrease in the cost function, followed by gradual convergence, suggesting potential overfitting. Linear regression highlighted a weak linear relationship between the analyzed feature and the target, implying a complex underlying relationship that might require more advanced modeling techniques.

CONCLUSION

Encountering perfect accuracy in our RNN model prompted a thorough investigation of our dataset and methodology. Entropy analysis and gradient descent proved valuable in understanding the data's characteristics and the model's training behavior. Our journey uncovers the importance of evaluation and highlights the potential of alternative approaches, such as ensemble methods or more complex RNN architectures, to reduce overfitting and improve RNN model generalization.

REFERENCES

[1] N. Nieto, V. Peterson, H. L. Rufner, J. E. Kamienkowski, and R. Spies, "Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition," *PLOS ONE*, vol. 17, no. 8, p. e0272498, 2022.

[2] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Muller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41-56, 2008.

[3] D. M. Hawkins, "The problem of overfitting," Journal of Chemical Information and Computer Sciences, vol. 44, no. 1, pp. 1-12, 2004.

[4] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An introduction to statistical learning," Springer, 2013.

[5] C. M. Bishop, "Pattern recognition and machine learning," Springer, 2006.

[6] C. E. Shannon, "A mathematical theory of communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 9, September 2024

[7] T. M. Cover and J. A. Thomas, "Elements of information theory," Wiley, 2006

[8] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," Clinical Neurophysiology, vol. 113, no. 6, pp. 767-791, 2002.

[9] B. Blankertz et al., "The Berlin brain-computer interface: non-medical uses of BCI technology," Frontiers in Neuroscience, vol. 4, p. 198, 2010.

[10] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2013, pp. 6645-6649.

[11] M. Angrick, C. Herff, T. Schultz, and M. Krauledat, "Inner speech decoding from single-trial EEG with recurrent neural networks," Journal of Neural Engineering, vol. 19, no. 5, 2022.

[12] M. Martin et al., "Decoding inner speech using electrocorticography in a tetraplegic patient," Nature Communications, vol. 13, no. 1, 2022.