Orthogonal Polynomials using Jacobi Polynomials in Artificial Neural Network

Phi Duong
Florida Institute of technology
Melbourne Florida United Stated
pduong2018@my.fit.edu

Introduction

Simple Artificial Neural Networks (ANN) consists of limitations that block the capabilities for the network to improve upon itself. Orthogonal polynomials provide a solution to one of these problems. A simple ANN uses a sigmoid function that bounds the output between 0 and 1. This limits the possible solution that a single layer can solve. For example, a simple ANN will only need 1 layer of units to represent AND, OR, NAND, NOR, but cannot represent XOR functions.

Orthogonal polynomials provide two main benefits. First, the function uses non-linearity, which allows the unit values to be more flexible in its representation. The second is the use of a set of polynomials that are orthogonal to each other. This allows each layer to act independent of one another, and thus, "helps to avoid column degeneracy of regression matrix" [2]. With these advantages, we expect to see an increase in the network's ability and speed to learn more complicated networks with less layers and units needed. The downside of using these orthogonal polynomials is computational expenses. This paper will try to compare the differences in learning capabilities and computational expenses between a simple ANN and one with orthogonal polynomials

This paper uses one of the orthogonal polynomials, specifically the Jacobi polynomial set. A Jacobi polynomial can be characterized by the orthogonality rule

$$\int_{-1}^{1} (1-x)^{\alpha}(1+x)^{\beta} P_m^{(\alpha,\beta)}(x) P_n^{(\alpha,\beta)}(x)\, dx = 0,\ m \neq n\ \ (1)$$

Where P(x) is the polynomial function with α and β as non-negative hyper parameters that control the shape of the function, and the subscript n and m are the degree of the polynomial.

Related work

In other works, Chakraborty [1], describes the theory and design uses for the Jacobi polynomials. They will use more complicated algorithms that can give insights into these polynomials can be used and applied. This can be inferred to understand better approached to improving the algorithm. Chu Kwong Chak and Chi Ming Chen's paper on "Orthogonal polynomials neural networks for function approximation" [1] and H. Zhang and R, Yu's paper on "Risk Assessment of substation based on Hermite orthogonal basis Feedforward Neural Network data fusion algorithm" [3] uses different orthogonal polynomials to research into, specifically the Legendre and Chebyshev polynomials. Their research can be checked with the results of this paper to find parallels in efficiency, and can be compared with each other to see which set of polynomials is best used for ANN.

Approach

The approach here, we will be using sets of Jacobi polynomial functions using different parameter sets and comparing the average accuracies and computational time against each other. In addition, the base ANN will also be tested to act as a control group for the experiment. We will use the solution to the Jacobi polynomial set as the activation function for each unit. The polynomial function can be expressed as

$$P_n^{(\alpha,\beta)}(x) = \sum_{s=0}^{n} \binom{n+\alpha}{n-s}\binom{n+\beta}{s}\left(\frac{x-1}{2}\right)^{s}\left(\frac{x+1}{2}\right)^{n-s}\ \ (2)$$

In formula (2). Larger polynomials tend to have more flexibility on how the formula will be shaped but at the expense of longer computation time as well as longer training time needed to learn the function's complexity. Since the range of the Jacobi polynomials are between -1 and 1, we will need to limit the dataset so it will output in those ranges. The network used in this paper will be a Feedforward Neural Network (FNN) and use the stochastic gradient decent algorithm to search through the hypothesis space. To combat overfitting, there will be a portion of the test data used as verification that will score the network. Then, the sets of weights of the highest accuracy are then stored and used for final analysis.

Evaluation

The quality of network will be determined based on 3 factors: the computation time, the number of learning iterations needed, and the accuracy of the network to the output. The different hyper parameters will then be selected for use in testing. The same n-order polynomials with then be compared to each other. Computational and accuracy will then be assessed when finding the most optimal n value. The paper will use at least two datasets: one for continuous attributes, using database on "Breast Cancer Wisconsin (Diagnostic)" [5], and the other using discrete attributes, from "Default of Credit Card Clients" [4].

REFERENCES

[1] H. S. Chakraborty, T. Miezaki, M. Oura, and Y. Tanaka, 'Jacobi polynomials and design theory I', *Discret. Math.*, vol. 346, p. 113339, 2022.
DOI: 10.48550/arXiv.2207.10911

[2] Chu Kwong Chak, G. Feng and Chi Ming Cheng, "Orthogonal polynomials neural network for function approximation and system modeling," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 594-599 vol.1,
DOI: 10.1109/ICNN.1995.488246.

[3] H. Zhang, R. Yu, J. Yu, W. Liu, S. Wu, and D. Liu, 'Risk Assessment of Substation based on Hermite orthogonal basis Feedforward Neural Network data Fusion algorithm', in *Proceedings of the 2024 International Conference on Power Electronics and Artificial Intelligence*, Xiamen, China, 2024, pp. 527–531.

DOI: https://doi.org/10.1145/3674225.3674320

[4] "Default of Credit Card Clients," UC Irvine Machine Learning Repository, This research aimed at the case of customers' default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods, 4/2/2025 [Online]. Available: https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

[5] "Breast Cancer Wisconsin (Diagnostic)," UC Irvine Machine Learning Repository, Diagnostic Wisconsin Breast Cancer Database. 4/2/2025 [Online]. Available: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic