International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

A Novel Paradigm For Enhancing Linear Model Predictions Across Multivariate Functions Through Integration of Cnns

Narasinga Sai Satwik Tenneti [1], Sneha Edupuganti[2]

[1,2] Student, School of Computer Science and Engineering, VIT-AP University

Abstract

Linear models have been a cornerstone in statistical data analysis, yet their inherent linearity often constrains their ability to extract nuanced patterns from complex datasets. This limitation poses a challenge in real-world applications where non-linearity prevails. In this research, we introduce an innovative methodology aimed at overcoming this deficiency. Leveraging the powerful non-linear modelling capabilities of Convolutional Neural Networks(CNNs), we augment the predictive capabilities of linear models. Through a systematic approach, we extend our findings to encompass specific classes of crucial multivariate functions. Our investigation primarily focuses on neural network architectures that integrate convolution (conv), Rectified Linear Unit (ReLU) activation functions, and max pooling layers. By harnessing the synergies of these components, we unveil a novel paradigm for enhancing linear model predictions, unlocking a wealth of potential applications across diverse domains. This research lays the foundation for a more robust and accurate predictive modelling framework that transcends the boundaries of conventional linear approaches.

Keywords: Convolutional Neural Network, Max Pooling, Rectified Linear Unit

1. Introduction

Supervised learning algorithms, exemplified by neural networks, are designed to approximate the underlying function within a given dataset. This is achieved by iteratively calculating the disparity between anticipated and actual outputs, with the aim of minimizing this error throughout the training process.

The term "approximate" is crucial here, as it signifies our acknowledgment that while we hypothesize the existence of a mapping function, its specifics remain elusive. Referred to as the target function, this elusive entity serves as the focal point of the training process—a function we endeavour to emulate using the available data. Indeed, were we privy to the precise details of this target function, the need for a supervised machine learning algorithm would be obviated? Consequently, function approximation emerges as an invaluable tool in scenarios where the true mapping function is shrouded in uncertainty. Within our dataset, we possess observations that furnish us with samples of both inputs and outputs. These observations come endowed with crucial characteristics, including their quantity and quality.

Notably:

- The abundance of examples in our possession directly impacts our capacity to glean insights about the mapping function.

- Likewise, the fidelity of our observations, gauged by the presence or absence of noise, profoundly influences the fidelity of our approximation to the mapping function.

These considerations lay the groundwork for our exploration into an innovative methodology that endeavors to bolster the predictive capabilities of linear models. By integrating the formidable capacities of Convolutional Neural Networks (CNNs) with traditional linear modelling techniques, we aim to transcend the limitations of conventional linear approaches and forge a path toward more accurate and adaptable predictive modelling in diverse real-world contexts[1]. This research endeavor seeks to redefine the boundaries of predictive modelling, offering a promising avenue for tackling complex data scenarios where linearity alone may fall short.

2. Literature review:

The preference for utilizing neural networks in function approximation arises from their demonstrated universal approximation property across a diverse range of function classes. A body of research, including seminal works by Cybenko (1989), Hornik et al. (1989), Funahashi (1989), Hornik (1991), Chui and Li (1992), Barron (1993), and Poggi et al. (2015), has consistently affirmed this fundamental characteristic. At its core, this property underscores the neural network's capacity as a universal approximator, theoretically capable of approximating any function.

This universal approximation theorem, articulated by Ian Goodfellow, Yoshua Bengio, and Aaron Courville in their authoritative book, asserts that a feedforward network equipped with a linear output layer and a minimum of one hidden layer employing a "squashing" activation function—such as the logistic sigmoid activation function—holds the potential to approximate any function within a finite-dimensional space to another with any desired non-zero margin of error. Crucially, this capability is contingent upon the provision of a sufficient number of hidden units within the network.

The practical implications of this theorem are profound, establishing neural networks as versatile tools for capturing intricate relationships within data. By leveraging this universal approximation property, researchers and practitioners can endeavour to enhance the predictive capabilities of linear models in scenarios where the underlying mapping function is complex and elusive. This foundational property forms the bedrock of our current investigation, as we aim to harness the synergistic potential of Convolutional Neural Networks (CNNs) in conjunction with traditional linear modelling techniques. This combined approach holds promise in overcoming the limitations of conventional linear modelling, thereby paving the way for more accurate and adaptable predictive modelling in a diverse array of real-world contexts.

3. f(x), A highly non-linear Function

Neural networks can have deeper neutral networks and more hidden layers with the help of non-linearity. Many times, layers with linear activation functions are combined into one layer, thereby creating a neural network with a hidden layer that lacks the benefits of deep neural networks.

A normalized output can also be produced via a nonlinear activation function. The most often used type of neural network, one that is Rectified linear measure, or Relu, is just a linear model that has been "rectified" in some manner.

output = input > 0? input: 0

linear output = w0 + w1*x + w2*y

x is input variable; y is target variable.

Inside a neural network, the Relu output would be a modification of the linear output as
relu_output = linear_output > 0? linear_output: 0
This method may be used repeatedly by feeding Relu outputs to more Relu units. This strengthens the conventional neuronal specification. I'll be able to demonstrate a simple network using two Relu units below. The weights of the fundamental Relu are w0, w1, and w2. The weights for the second Relu are u0, u1, and u2. Then, to encourage the final product, we merge the two Relus using a linear model with weights v0, v1, and v2.[3]

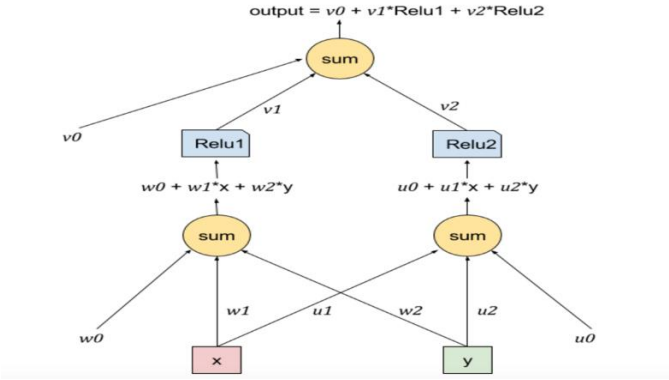

Fig. 1. Representation of the model used

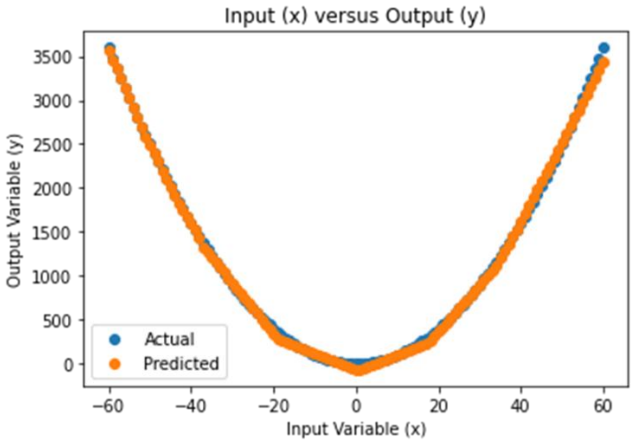

Fig. 2. The below graph depicts the non-linear data of the input, output variables of actual and predicted values graph.

4. A Deep Learning Model for (x)

The Convolutional Neural Network (CNN) stands at the forefront of signal processing techniques, particularly in the analysis of multidimensional data such as images[2]. Its adoption provides an unprecedented level of control over network behaviour. In this study, we present a novel approach wherein a CNN is constructed exclusively using the NumPy library. The architecture comprises three fundamental layers: convolution (abbreviated as conv), Rectified Linear Unit (ReLU) activation, and max pooling. By leveraging this minimalist yet potent framework, we aim to unveil the underlying principles and mechanisms governing the behaviour of CNNs. [4]
The following are the major steps:

1. Reading the input image.
2. Preparing filters.

3. Conv layer: Convolving each filter with the input image.
4. ReLU layer: Applying ReLU activation function on the feature maps (output of conv layer).
5. Max Pooling layer: Applying the pooling operation on the output of ReLU layer.
6. Stacking conv, ReLU, and max pooling layers.

5. Results and Discussion

The first step is reading the picture because the next ones rely on the input size. The picture after it has been made grey is given below.

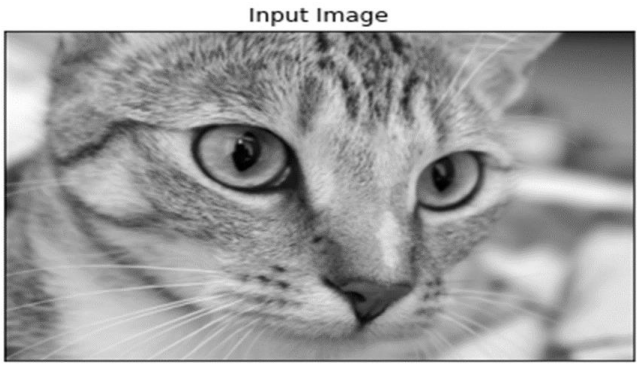

Fig. 3. Chelsea Gray Image accessed via skimage data

- In l1_filter: A zero array is created according to the number of filters and the size of each filter. 2 filters of size 3x3 are created that is why the zero array is of size (2=num_filters, 3=num_rows_filter, 3=num_columns_filter)
- The l1_feature_map convolves the image with the filters bank using a function called conv
- In l1_feature_map_relu: The ReLU layer applies the ReLU activation function over each feature map returned by the conv layer.
- In l1_feature_map_relu_pool: The max pooling layer accepts the output of the ReLU layer and applies the max pooling operation[5].

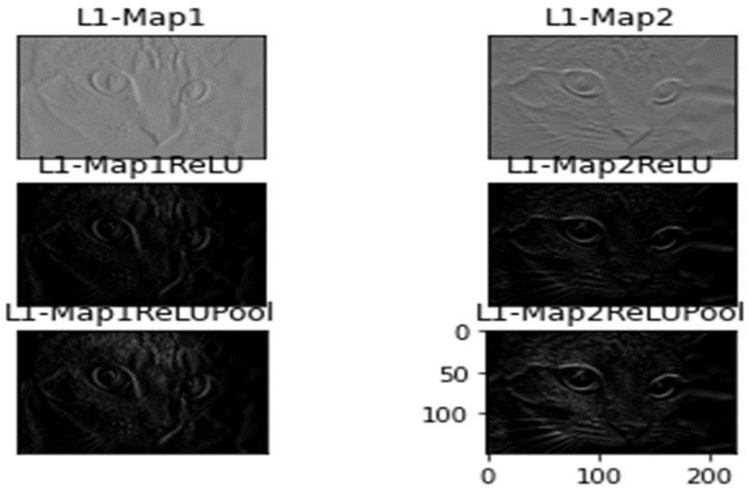

Fig. 4. Output feature maps, Relu layer, Max pooling layer of the first conv layer.

This is also the same for the successive ReLU and pooling layers. Outputs of such layers are shown below.

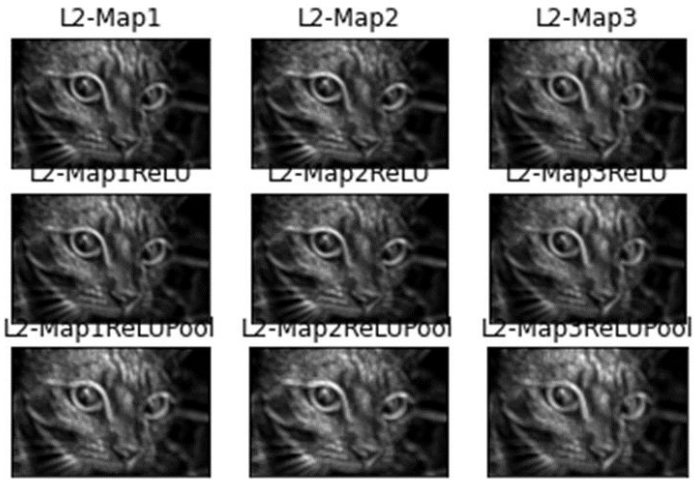

Fig. 5. Output feature maps, Relu layer, Max pooling layer of the second conv layer.

The previous conv layer accepts just a single filter. That is why there is only one feature map as output.

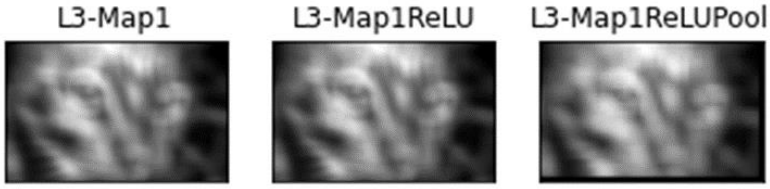

Fig. 6. Output feature maps, Relu layer, Max pooling layer of the third conv layer.

The visualization of the outputs from each layer using the **Matplotlib** library.

6. Conclusion

In this research paper, we embarked on a journey into the realm of deep learning, focusing on a Convolutional Neural Network (CNN) model tailored to approximate our selected function. The architecture, characterized by three pivotal layers—convolution, Rectified Linear Unit (ReLU), and max pooling—formed the foundation of our investigation[6]. Through meticulous examination, we delved into the intricacies of feature maps across these layers, illuminating the evolving patterns and highlighting the discernible shifts from dark to light regions. Our findings underscore the profound impact of architectural choices in deep learning models. The incorporation of convolution, ReLU activation, and max pooling layers facilitated a nuanced understanding of the chosen function, enabling us to glean insights into its underlying structure. The dynamic alterations observed in the feature maps as the network processed the data exemplified the network's capacity to extract intricate details and hierarchies within the data[7]. This research not only reinforces the significance of a minimalist yet powerful CNN architecture but also emphasizes the value of hands-on implementation using fundamental libraries like NumPy. By doing so, we empowered ourselves with a deeper comprehension of the intricacies at play within the network, affording us unparalleled control and insight.

7. Future scope

This research has opened up several promising avenues for future exploration and development in the field of deep learning and multidimensional signal analysis. Here are some key areas:

- Deployment in Real-world Applications: Extending the research to practical, real-world applications is a crucial step. Implementing the CNN model in scenarios where accurate signal analysis is essential, such as medical diagnostics, autonomous systems, or quality control in manufacturing, can have a significant societal and industrial impact.
- Architectural Refinement: The current study focused on a minimalist CNN architecture comprising convolution, ReLU activation, and max pooling layers. Future research could delve into exploring more complex architectures, including variations in layer types, depths, and connections. Investigating the impact of additional layers and alternative activation functions may yield further enhancements in performance and understanding[8].
- Diverse Data Science Domains: While this study focused on multidimensional signal analysis, the techniques and insights gained can be applied to a wide array of data domains beyond images. Exploring applications in fields such as natural language processing, speech recognition, and medical imaging could lead to significant advancements in various domains.

8. References

1. Kuo, C. C. J. (2016). Understanding convolutional neural networks with a mathematical model. Journal of Visual Communication and Image Representation, 41, 406-413.
2. Koushik, J. (2016). Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081*.
3. Y. LeCun, Y. Bengio, G. E. Hinton, Deep learning, Nature 521 (2015) 436–444.
4. J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1872–1886.
5. K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034.
6. J. Dai, Y. Lu, Y.-N. Wu, Generative modeling of convolutional neural networks, arXiv preprint arXiv:1412.6296
7. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366
8. L. Zhang, B. Zhang, A geometrical representation of mcculloch-pitts neural model and its applications, IEEE Transactions on Neural Networks 10 (4) (1999) 925–929