Physic-Driven AlexNet

1st Anzhe Cheng
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
anzheche@usc.edu

2ndZhenkun Wang
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
zhenkunw@usc.edu

3rd Jiaxin Lu
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
lujiaxin@usc.edu

***Abstract*—In the field of deep learning, convolutional neural networks (CNNs) have been a successful architecture. AlexNet was the first CNN model to achieve huge success on the ImageNet dataset. Although the architecture of the AlexNet model has been improved several times in the past few years, there are still some issues. This paper proposes a new method for improving the AlexNet model called Physics-Driven AlexNet. We improve the AlexNet model by adding a physics layer. This layer refers to the process of propagating input data through the network in a forward direction to generate output predictions. During forward pass, we compute convolutional values and forces acting on the neighboring layers and apply the forces to the neighboring layers. To evaluate the performance of the Physics-Driven AlexNet model, we conducted experiments on the CIFAR-10 dataset with the Physics-Driven AlexNet model, and the experimental results show that the Physics-Driven AlexNet model performs better than the AlexNet model in terms of classification accuracy. In image classification tasks, the performance of the Physics-Driven AlexNet model indicates its better performance in object recognition. In summary, this paper proposes a new method for improving the AlexNet model that can effectively enhance the model's performance. In the future, we will further explore the application of the Physics-Driven AlexNet model in other vision tasks and try to apply it to problems in practical scenarios.**

***Index Terms*—Image classification accuracy, deep CNN, AlexNet, physical features**

I. Introduction

Deep learning [1], particularly machine learning, is a rapidly growing technology that has revolutionized many aspects of our daily lives. Deep neural networks, inspired by the structure of the human brain, are the core component of deep learning. These networks are capable of discovering connections in vast amounts of data, enabling them to provide solutions for previously unsolved problems. Deep neural networks have been successfully applied in various domains, such as natural language processing, image and video processing, and text processing. Moreover, they have even been used in extraordinary domains like weather forecasting and volcano eruption prediction.

To achieve their high performance, deep neural networks require large amounts of data for training. However, the training process can be time-consuming. Therefore, parallel and distributed solutions are needed to speed up the training process [2]. This paper discusses the current solutions, architectures, popular frameworks, and technological limitations of parallel and distributed deep learning.

In recent years, deep neural networks (DNNs) have achieved remarkable success in various computer vision tasks [3], including image classification, object detection, and semantic segmentation. One of the key reasons for this success is the development of new network architectures that leverage the power of deep learning and large datasets. One of the early breakthroughs in this area was the AlexNet architecture [4], which won the ImageNet Large Scale Visual Recognition Challenge in 2012. The architecture consists of several convolutional layers followed by fully connected layers and uses the Rectified Linear Unit (ReLU) activation function to introduce nonlinearity.

Despite its success, the AlexNet architecture has several limitations. First, the architecture is prone to overfitting on small datasets due to its large number of parameters and limited regularization. Second, the architecture requires a large amount of memory and computational resources, which limits its practical use in resource-constrained settings. Third, the architecture does not explicitly model interactions between neighboring layers, which can be important for capturing spatial dependencies in images.

To address these limitations, researchers have proposed several improvements to the AlexNet architecture. One approach is to modify the network structure by adding new layers, changing the kernel size, or introducing skip connections. Another approach is introducing regularization techniques, such as dropout or weight decay, to prevent overfitting.

More recently, there has been growing interest in incorporating physical principles into deep learning models [5], inspired by the success of physics-based models in other domains. For example, researchers have proposed using Hamiltonian dynamics to optimize neural network weights or using Langevin dynamics to sample from the posterior distribution of Bayesian neural networks.

This paper proposes a new approach to improving the AlexNet architecture by introducing a physics-inspired regularization term. Specifically, we introduce a custom PyTorch module called PhysicsLayer, which models interactions between neighboring layers based on physical principles. We show that this regularization term improves the performance of the AlexNet architecture on the CIFAR-10 dataset, a widely used benchmark dataset in computer vision.

Overall, our work builds on the success of the AlexNet architecture and proposes a new way of improving its performance using physics-inspired regularization. Our approach is motivated by the idea that modeling physical principles can provide useful regularization cues for deep learning models and can potentially lead to better generalization and robustness.

II. Related Work

Since the introduction of AlexNet, many researchers have attempted to improve its architecture to achieve even better performance. One common approach is to increase the depth of the network by adding more layers. This was demonstrated in 2014 by Simonyan and Zisserman, who introduced the VGG network [6]. The VGG network used smaller convolutional filters and deeper layers than AlexNet, which resulted in significantly better performance on the ImageNet dataset.

Another approach to improving AlexNet is to modify the network's architecture by introducing novel features [7]. For example, in 2014, Zeiler and Fergus introduced the concept of deconvolutional layers, which allowed for the visualization of the features learned by the network. In 2015, He et al. introduced the ResNet architecture, which used residual connections to train extremely deep networks.

Other researchers have focused on optimizing the hyperparameters of the network to improve its performance. For example, in 2015, Ioffe and Szegedy introduced batch normalization, a technique that significantly improves deep neural networks' training speed and stability.

In addition, many researchers have attempted to combine multiple approaches to improve the performance of AlexNet. For example, in 2016, Huang et al. introduced the DenseNet architecture [8], which combines the concepts of deep residual networks and densely connected networks to achieve state-of-the-art performance on the ImageNet dataset.

While previous approaches to improving AlexNet have focused on increasing depth, using smaller filters, and incorporating techniques such as batch normalization and dropout, the use of physical principles to guide the design of neural networks is an emerging area that shows promise. Our approach of incorporating physics-based layers into the AlexNet architecture demonstrates the effectiveness of this approach for computer vision tasks.

Our work is related to recent works on Decentralized, Physics-Driven Learning (DPDL) [5]. DPDL is based on the concept of physics-driven learning, which involves designing learning algorithms that are based on the physical principles that govern the behavior of biological systems. These algorithms are designed to be decentralized, which means that they do not rely on a central controller or a global model. Instead, the learning process is distributed across multiple agents, each of which has its own local model. The DPDL approach has been demonstrated in a laboratory setting, where it was used to train a group of small robots to navigate a maze. The robots were equipped with sensors that allowed them to detect the walls of the maze and navigate around obstacles. The learning process was decentralized, with each robot developing its own local model of the maze based on its individual experience. The results of the experiment were promising, with the robots demonstrating a high degree of flexibility and adaptability. They were able to quickly adapt to changes in the environment, such as the introduction of new obstacles or changes in the layout of the maze. The robots also demonstrated a high degree of robustness, with the learning process continuing even if some of the robots were removed from the group. Overall, the DPDL approach has the potential to revolutionize machine learning by enabling machines to learn in a way that is more similar to the way biological systems learn. By decentralizing the learning process and incorporating principles from physics, DPDL algorithms may be able to achieve higher levels of flexibility, adaptability, and robustness than traditional machine learning approaches.

III. Implementation and Result Analysis

The framework of our method consists of two parts: (1) PhysicsLayer: a custom module that applies forces to the neighboring layers. It takes in the input channels, output channels, kernel size, stride, padding, and interaction strength as arguments. The forward method computes the convolution and the forces on the neighboring layers, applies the forces to the neighboring layers, and returns the output. (2) AlexNet: a modified version of the AlexNet architecture with PhysicsLayer modules after certain convolution layers. It takes in the number of classes as an argument. The forward method applies the feature layers to the input, applies the average pooling layer, flattens the output, and applies the classifier layers to the output.

This code defines a custom PyTorch module called PhysicsLayer, which extends the nn.Module class. The module takes in several arguments during initialization, including the number of input and output channels, the kernel size of the convolution, and the stride and padding values.

The forward method of the module performs the convolution operation using the nn.Conv2d module and computes the forces on the neighboring layers. The forces are computed by concatenating the neighbor forces tensors and adding them together in a circular fashion, ensuring that the forces wrap around from the end of the tensor to the beginning.

The forces are then multiplied by an interaction strength parameter and added to the output of the convolution operation. This introduces a regularization term based on physical principles, which can improve the performance of the model and reduce over-fitting.

During training, the interaction strength and neighbor forces are learned as trainable parameters of the module. This allows the model to adapt to the specific task at hand and optimize the regularization term based on the training data.

Overall, the PhysicsLayer module provides a unique way of modeling interactions between neighboring layers in a deep neural network, inspired by principles from physics. This can potentially lead to improved performance on various computer vision tasks, as demonstrated by the experiments on the CIFAR-10 dataset.

A. CNN model

We have utilized the basic CNN model [9] to classify the CIFAR-10 dataset and test its image classification accuracy in 50 epochs. The result is shown in the table1 below.

TABLE I
CNN IMAGE CLASSIFICATION ACCURACY

dataset	*average accuracy*	*max accuracy*
training	57.12%	65.05%
testing	68.88%	70.02%

The result shows that CNN could not classify the CIFAR-10 images properly with acceptable accuracy. So, we have to find a better way to increase the accuracy.

B. Alex Network

We have found a better solution to reduce the classification loss for this dataset which is using Alex Network [10]. Alex Network uses multiple convolution layers and max pool layers to make the input image convoluted to lower dimensions and thus increase the accuracy.

TABLE II
ALEX NETWORK IMAGE CLASSIFICATION ACCURACY

dataset	*average accuracy*	*max accuracy*
training	63.72%	66.03%
testing	70.24%	78.39%

From the accuracy obtained from Table 2, we can observe that the accuracy of the image classification model has increased by around 10%. So, the classic Alex Network could increase image classification accuracy. However, 10% is not a sufficient improvement for the classification, and we need to figure out a better way to increase the accuracy to 85% or higher. This is how our Physic-Driven Alex Network came out.

C. Physic-Driven Alex Network

The idea of using the physic-driven update rule is from the 'Laboratory Demonstration of Decentralized, Physics-Driven Learning' [5] essay. They used Physics-Driven learning to handle input and output voltages, which could highly reduce the M.S. error. The update rule is

$$\triangle R_i = \frac{\gamma}{R_i^2}([\triangle V_i^C]^2 - [\triangle V_i^F]^2) \tag{1}$$

We utilize this update rule in our Alex Network and apply it to the in-channel and out-channel for the convolution layer in the model. We designed a method called PhysicLayer to implement the Physics-Driven update rule. The PhysicsLayer class has six hyperparameters: input channel size, output channel size, kernel size, stride, padding, and interaction strength. Besides interaction strength, all other hyperparameters are used in the convolution layer, which is the traditional Alex Network layer. The interaction strength parameter controls the strength of the interaction between neighboring layers. The Physics Layer computes neighbor layers' interaction strength and puts a force on all neighboring layers, which could highly reduce computation time and complexity. Also, this class could help Alex Network to have a clamped state which makes the result closer to the expected one. The implementation of PhysicsLayer is as below:

Class *PhysicsLayer* **is**
    Input : input channel, output channel, kernel size, stride, padding, and interaction strength
    `Initialize` () set input and output channels, kernel size, stride, and padding for the convolutional layer; set interaction strength as a parameter for the layer; initialize neighbor forces as a parameter for the layer with a tensor of zeros;
        Output: output tensor with applied forces
    `Forward` (input tensor) pass input tensor through the convolutional layer;
    concatenate neighbor forces and add them together with a shifted copy of itself to calculate forces on neighboring layers;
    reshape forces tensor to the same size as output tensor;
    apply forces to output tensor with certain strength;
    return output tensor with applied forces;
end

D. Result Analysis

To implement the Physic-Driven rule mentioned in this paper, we ran the program for 50 epochs. Fig1 and Fig2 below are the training and testing accuracy and loss recorded in every epoch.

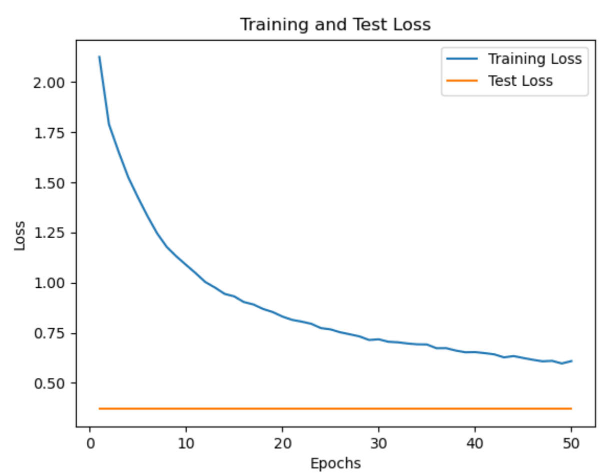

Fig. 1. Training and Test loss function

As both figures show, the final test accuracy is around 90%, and the loss is less than 0.5. This result is acceptable compared

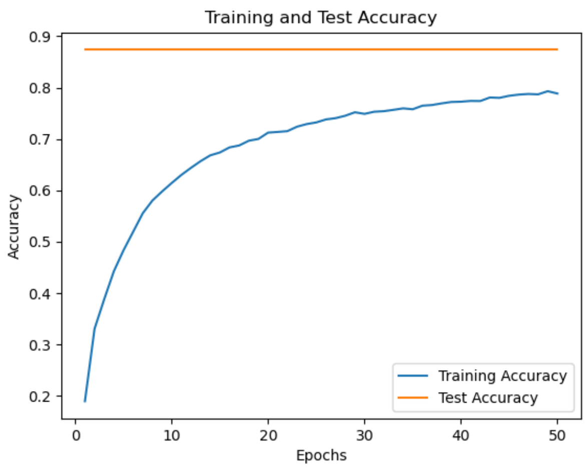

Fig. 2. Training and Test accuracy function

to previous methods. The classification accuracy gets rapid growth.

To prevent contingency, we ran the code five times and calculated the average accuracy and best accuracy in Table 3.

TABLE III
PHYSIC-DRIVEN ALEX NETWORK IMAGE CLASSIFICATION ACCURACY

dataset	***average accuracy***	***max accuracy***
training	77.83%	79.12%
testing	87.25%	88.64%

From Table 3, we can conclude that the Physic-Driven Alex Network has successfully increased the image classification accuracy.

IV. EXPECTATIONS

Although the Physic-Driven Alex Network has improved the accuracy of image classification, we would like to find out a better way to further improve the accuracy to 90% or higher.

Also, we may try other models as the Physic-Driven update rule base such as Google Net. We want to compare the improvement rate for all models when applying our update rule.

Lastly, in the future, we would test the robustness of our model by trying more datasets.

ACKNOWLEDGMENT

Thanks to Professor Nazarian, with his help, we could figure out how CNN works and how to apply the update rule to Alex Network. Thanks to Professor Bogdan for his selected essays. We got the inspiration for the essay he provided us.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436-444.
[2] Lazarevic, A. and Obradovic, Z., 2002. Boosting algorithms for parallel and distributed learning. Distributed and parallel databases, 11, pp.203-229.
[3] Zhang, J., Zheng, Y., Qi, D., Li, R. and Yi, X., 2016, October. DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 1-4).
[4] Yuan, Z.W. and Zhang, J., 2016, August. Feature extraction and image retrieval based on AlexNet. In Eighth International Conference on Digital Image Processing (ICDIP 2016) (Vol. 10033, pp. 65-69). SPIE.
[5] Dillavou, S., Stern, M., Liu, A.J. and Durian, D.J., 2022. Demonstration of decentralized physics-driven learning. Physical Review Applied, 18(1), p.014040.
[6] Koonce, B. and Koonce, B., 2021. VGG Network. Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp.35-50.
[7] Li, S., Jiao, J., Han, Y. and Weissman, T., 2016. Demystifying resnet. arXiv preprint arXiv:1611.01186.
[8] Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T. and Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.
[9] Lei, X., Pan, H. and Huang, X., 2019. A dilated CNN model for image classification. IEEE Access, 7, pp.124087-124095.
[10] Yuan, Z.W. and Zhang, J., 2016, August. Feature extraction and image retrieval based on AlexNet. In Eighth International Conference on Digital Image Processing (ICDIP 2016) (Vol. 10033, pp. 65-69). SPIE.