PREDICTIVE CODING INSPIRED CONVOLUTIONAL NEURAL NETWORK

Mohammad Shakil *Rong Zeng*

Department of ECE, University of Rochester

ABSTRACT

This work introduces a biologically inspired neural network, termed Predictive Coding Network (PCN), which integrates both feedforward and feedback processing inspired by predictive coding (PC). Unlike conventional feedforward convolutional neural networks, PCN includes local recurrent interactions between adjacent layers that allow representations to be iteratively refined. This dynamic interaction continues over multiple recurrent cycles, effectively enabling a shallow network to perform deep transformations. The model is evaluated on image classification benchmark CIFAR, where it achieves competitive accuracy ($\approx$92%) despite having fewer parameters ($<$0.6M) and layers than standard models for image classification tasks like ResNet.

***Index Terms*—** PCN, Local Recurrent Processing, Noise Robust Network

1. INTRODUCTION

State-of-the-art approaches in computer vision are predominantly built on feedforward convolutional neural networks (CNNs)[1], which have become the standard framework for tasks like image classification. These networks often improve accuracy by increasing depth, sometimes with shortcut connections to ease optimization. This design philosophy aligns with the idea that solving visual tasks requires learning increasingly abstract representations across multiple layers, an approach drawing parallels with the hierarchical structure of the visual cortex in the brain[2].

However, there's a key distinction: unlike deep CNNs with hundreds of layers, the brain operates with relatively shallow hierarchies but still manages highly robust and efficient perception. One critical feature of biological neural systems is the presence of abundant feedback pathways, which are largely absent in typical CNNs [3]. These feedback connections enable top-down signals to interact with bottom-up inputs over time, forming a recurrent dynamic system.

This interplay is central to the neuroscience theory of predictive coding, which proposes that higher-level brain areas send predictions downward, while lower-level areas forward the mismatch or error upward. This continuous exchange leads to refined internal states across the hierarchy. In artificial networks, such mechanisms can enable a fixed architecture to simulate deeper transformations over time.

Motivated by this concept, we implemented the Predictive Coding Network (PCN), a biologically inspired CNN that performs local recurrent processing with neighboring layers. Unlike previous models [4] using global feedback loops, PCN limits its recurrence to adjacent layers, making the network more efficient and biologically plausible. Experimental results on standard datasets show that this architecture can achieve strong performance with far fewer layers and parameters, and its behavior sheds light on its computational advantages and interpretability.

2. PREDICTIVE CODING

The Predictive-Coding Network (PCN) is a deep neural network architecture inspired by predictive coding [5] — a widely studied neuroscience theory about how the brain processes information [6]. According to this theory, the brain functions as a hierarchical inference machine that continuously predicts sensory inputs and corrects itself through feedback. Translating this principle into a computational framework, PCN models a deep neural network in which each layer attempts to predict the activity of the layer below and refines its own state through local recurrent processing.

The PCN architecture is built from a series of recurrent blocks. Each block contains a local processing loop that exchanges information between adjacent layers. The upper layer attempts to predict the activity of the lower layer using feedback connections. In response, the feedforward pathway transmits the difference between the actual and predicted values—the prediction error—back to the upper layer, which uses it to refine its own representation. This iterative update continues for a fixed number of steps. Afterward, the lower-layer activation is combined with the refined upper-layer output through a bypass connection. The result then moves to the next block, where the process repeats at a higher level. Once the entire stack has processed the input, the final representation is used to perform classification.

2.1. Theory

Predictive coding (PC) for a network of L layers can be viewed as a generative model that minimizes a global energy function (F) comprised of local errors [7].

$$F = \sum_{l=0}^{L-1} \frac{1}{2} \|\epsilon_l\|^2 = \frac{1}{2} \sum_{l=0}^{L-1} \|\mu_l - x_l\|_2^2$$

where x_l is the actual representation and μ_l is the prediction for layer l,

$$\mu_l = W_l^T x_{l+1}$$

with x_L being the generated instruction or label, and x_0 being the generated model.
Compared with back-propagation, instead of propagating the gradient back for learning, each part of the error can be minimized at the same time in PC during training stage. The PC-based training includes two stages. The first stage is inference, where the weights of the network are fixed. The input x_0 and the output label $y = x_{L+1}$ are also clamped, only the hidden representations x_l in the middle are updated to minimize $\mathcal{F}$. The second stage is called learning, where all of the hidden representations are fixed and the weights are updated using gradient descent.

2.2. Local Recurrent

If we look at each layer separately and locally, and with y being the output of the local PC recurrent, we try to minimize the local error ϵ_l with respect to y. The gradient of local error ϵ_l with respect to y is given by

$$\nabla_y \left(\frac{1}{2} \|\epsilon_{l-1}\|^2 \right) = W_{\text{pc}} (r_{l-1} - W_{\text{pc}}^T y)$$

The gradient descent is run for T cycles with respect to y to minimize the local loss defined by PC, as shown in Fig. 1. The intuition behind the local recurrence is to gain some knowledge or structure based on the input of each layer.

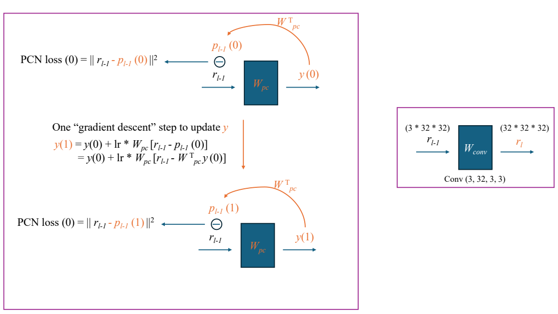


Fig. 1: Gradient step to update y.

3. PREDICTIVE CODING INSPIRED NEURAL NETWORK

Inspired by the predictive coding scheme described above, a convolutional neural network with PC recurrent added is implemented. On the high-level, this modifies each convolutional layer with extra recurrent structure. Fig. 2 shows the high-level comparison with normal convolution.

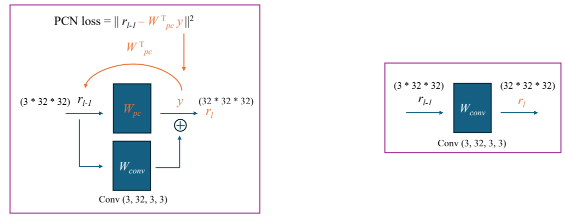


Fig. 2: PC recurrent added versus normal convolution layer.

When minimizing the local PC error, each gradient descent step can be viewed as one feedforward convolution layer plus a feedback transposed convolution layer. Unlike the naive way of optimizing the local PC loss, we use different sets of parameters for feedback and feedforward convolution. To increase the complexity of the model, we also added ReLU as nonlinearity between feedforward and feedback convolutions. We also added a bypass convolution layer to directly propagate the input information to the end of the recurrent layer to help back-propagate the gradient.

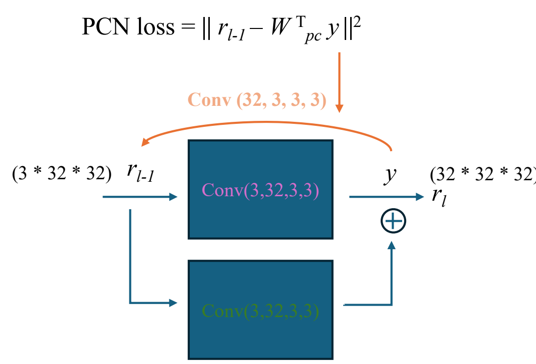


Fig. 3

The updated convolutional layer takes r_{l-1} as input and outputs r_l, with y being the intermediate variable as the output of PC recurrent structure.

$$y_{t+1} = y + lr * W_{FF} (r_{l-1} - W_{FB}\, y_t)$$

after T steps,

$$r_l = y_T + W_{BP}\, r_{l-1}$$

4. EXPERIMENTS

Based on the convolutional layer structured above, we build multiple models with different configurations and tested their performance on standard image classification tasks.

4.1. Baseline model

To evaluate the PC-inspired CNN, we built a CNN with 5 convolutional layers with recurrent structure, followed by an average pooling layer and a final linear layer for 10 class classification task. We trained and evaluated the model on the CiFar-10 data. The model is trained for 150 epochs using stochastic gradient descent with initial learning rate of 0.01 and momentum equaling 0.9. The learning rate is decreased by a factor of 10 at epochs 80 and 122. The model is compared with ResNet of different sizes.

PCN (∼0.59M)	**ResNet-18** (∼11M)	**ResNet-34** (∼21M)
91.11%	93.07%	93.34%

The model is able to achieve more than 90% of accuracy using with much smaller number of parameters compared with other deeper convolutional neural networks.
The recurrence in each convolutional layer is inspired by PC, which aims to minimize the local error $||\mu_l - x_l||^2$. However, in the implementation, the weights for feedforward and feedback convolutions are decoupled. To verify if the local PC loss has decreased under such setting, we plot the number of recurrent iterations versus local PC loss in Fig. 4.
For the last two layers, the layer is actually not optimizing the local PC loss, which is intuitive because the last layers are more closed to classifiers and thus, the weights of the last two layers are trained more to reducing the train loss.
We then take the trained model and tested it using different schemes, including tying the weights of feedforward and feedback layer, removing the relu between feedforward and feed layer, treating PC recurrent as solving an optimization problem (direct) and so on.

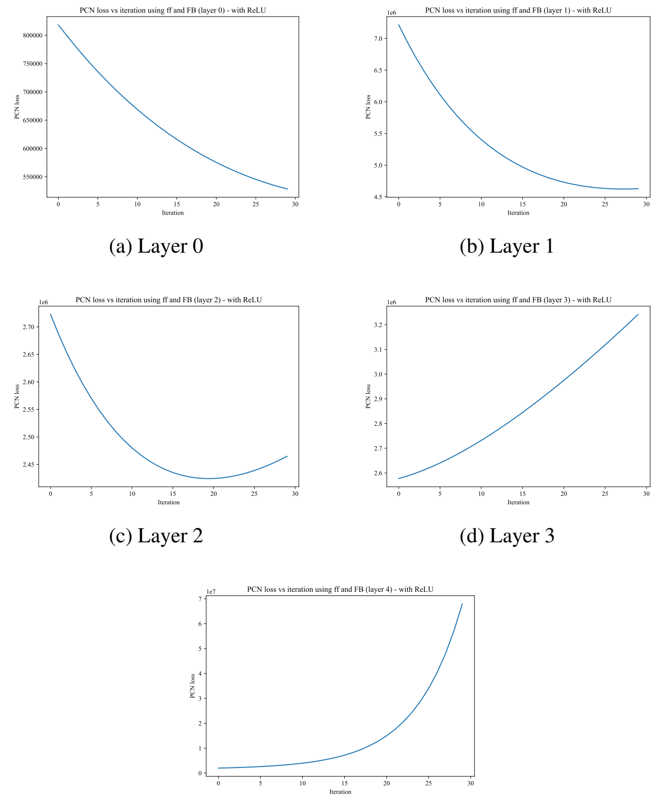
Table 1: Classification accuracy under different training setups.

Method	**Conv layers**	**With ReLU**	**Acc.**
MVM	FB, FF, BP	No	50.08 %
		Yes	90.02 %
	FB, BP	Yes	13.64 %
	FB, BP	No	14.10 %
Direct	FB, BP	No	13.08 %

The result shows that if we test the model with a different scheme than it is trained with, the accuracy will drop dramatically, even if the test scheme matches intuition of the PC local loss.

4.2. Model Configuration Exploration

Motivated by the loss plot above, we then explored the possibilities to reduce the local PC loss and retain image classification accuracy at the same time. Thus there are a couple



(a) Layer 0

(b) Layer 1

(c) Layer 2

(d) Layer 3

(e) Layer 4

Fig. 4: Training loss for each PCN layer.

of configurations we can select. Multiple models are trained under different configurations. The accuracies for different configurations are demonstrated in Table.2
From Table.2, if we make some adjustments to the recurrent layer in PCN and train the model with updated configurations, the model accuracy will not be degraded by more than 3%. The degradation caused by removing ReLU or tying the weights of feedforward and feedback convolutions can be partly resolved by adding more layers such that the model size is still small ($<$ 1M). The results imply that we can design the recurrent convolutional layer so that it matches the original idea of PC and with comparable performance on image classification task.

5. CONCLUSION

In this paper, we started from the basics of PC algorithm and it's biological inspiration. Based on the original algorithm, we implemented a PC-inspired convolutional neural network and evaluated its performance on image classification task. The biological inspired CNN is able to achieve classification accuracy with less than 0.6M parameters. We then finished

Table 2: Accuracy for different tying and ReLU settings.

Tie FF/FB	Tie BP	With ReLU	Param (M)	Acc.
Untied	Untied	Yes	0.59	91.11 %
		No	0.92	89.96 %
			0.59	89.58 %
	Tied	Yes	0.87	92.24 %
		No	0.87	89.61 %
Tied	Untied	Yes	0.86	91.21 %
		No	0.86	88.31 %
	Tied	Yes	0.77	91.62 %
		No	0.77	88.48 %

our experiments by showing that it is possible to match the PC inspiration while maintaining accuracy. Currently this model has a PC-like structure but is still trained with back-propagation. Our future goal is to train this PC-inspired CNN with PC algorithm in a energy-based hardware [8].

6. REFERENCES

[1] Abolfazl Younesi, Mohsen Ansari, MohammadAmin Fazli, Alireza Ejlali, Muhammad Shafique, and Jörg Henkel, "A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends," *arXiv preprint arXiv:2402.15490*, 2024.

[2] James CR Whittington and Rafal Bogacz, "An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity," *Neural computation*, vol. 29, no. 5, pp. 1229–1262, 2017.

[3] Sebastian Herzog, Christian Tetzlaff, and Florentin Wörgötter, "Evolving artificial neural networks with feedback," *Neural Networks*, vol. 123, pp. 153–162, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] Kuan Han, Haiguang Wen, Yizhen Zhang, Di Fu, Eugenio Culurciello, and Zhongming Liu, "Deep predictive coding network with local recurrent processing for object recognition," *Advances in neural information processing systems*, vol. 31, 2018.

[6] Yanping Huang and Rajesh PN Rao, "Predictive coding," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 5, pp. 580–593, 2011.

[7] Karl Friston, James Kilner, and Lee Harrison, "A free energy principle for the brain," *Journal of physiology-Paris*, vol. 100, no. 1-3, pp. 70–87, 2006.

[8] Richard Afoakwa, Yiqiao Zhang, Uday Kumar Reddy Vengalam, Zeljko Ignjatovic, and Michael Huang, "Brim: Bistable resistively-coupled ising machine," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 749–760.