Deep Predictive Coding Network with Local Recurrent Processing for Object Recognition

Kuan Han[1], Haiguang Wen[1], Yizhen Zhang[1], Di Fu[1], Eugenio Culurciello[1,2], and Zhongming Liu[1,2,*]

[1]School of Electrical and Computer Engineering, Purdue University
[2]Weldon School of Biomedical Engineering, Purdue University

Abstract

Inspired by "predictive coding" - a theory in neuroscience, we develop a bi-directional and dynamical neural network with local recurrent processing, namely predictive coding network (PCN). Unlike any feedforward-only convolutional neural network, PCN includes both feedback connections, which carry top-down predictions, and feedforward connections, which carry bottom-up errors of prediction. Feedback and feedforward connections enable adjacent layers to interact locally and recurrently to refine representations towards minimization of layer-wise prediction errors. When unfolded over time, the recurrent processing gives rise to an increasingly deeper hierarchy of non-linear transformation, allowing a shallow network to dynamically extend itself into an arbitrarily deep network. We train and test PCN for image classification with SVHN, CIFAR and ImageNet datasets. Despite notably fewer layers and parameters, PCN achieves competitive performance compared to classical and state-of-the-art models. Further analysis shows that the internal representations in PCN converge over time and yield increasingly better accuracy in object recognition. Errors of top-down prediction also map visual saliency or bottom-up attention. This work takes us one step closer to bridging human and machine intelligence in vision.

1 Introduction

Modern computer vision is mostly based on feedforward convolutional neural networks (CNNs) [15, 27, 41]. To achieve better performance, CNN models tend to use increasingly more layers [16, 19, 41, 49], while sometimes adding "short-cuts" to bypass layers [16, 47]. What motivates such design choices is the notion that models should learn a deep hierarchy of representations to perform complex tasks in vision [41, 49]. This notion generally agrees with the brain's hierarchical organization [25, 52, 57, 14]: visual areas are connected in series to enable a cascade of neural processing [50]. If one layer in a model is analogous to one area in the visual cortex, the state-of-the-art CNNs are considerably deeper (with 50 to 1000 layers) [17, 15] than the visual cortex (with 10 to 20 areas) . As we look to the brain for more inspiration, it is noteworthy that biological neural networks support robust and efficient intelligence for a wide range of tasks without any need to grow their depth or width [31].

What distinguishes the brain from CNNs is the presence of abundant feedback connections that link a feedforward series of brain areas in reverse order [10]. Given both feedforward and feedback connections, information passes not only bottom-up but also top-down, and interacts with one another to update the internal representations over time. The interplay between feedforward and feedback connections has been thought to sub-serve the so-called "predictive coding" [38, 12, 43, 20, 11] - an influential theory in neuroscience. It says that feedback connections from a higher layer carry

[⋆] Correspondence to: Zhongming Liu <zmliu@purdue.edu>

the prediction of its lower-layer representation, while feedforward connections in turn carry the error of prediction upward to the higher layer. Repeating such interactions across layers renders the visual system a dynamical and recurrent neural network [1, 11]. As recurrent processing unfolds in time, a static network architecture is used over and over to apply increasingly more non-linear operations to the input, as if the input were computed through more and more layers stacked onto an increasingly deeper CNN [31]. In other words, running computation through a bi-directional network for a longer time may give rise to an effectively deeper network to approximate a complex and nonlinear transformation from pixels to concepts [31, 6], which is essential to invariant object recognition among many other aspects of vision.

Inspired by predictive coding, we propose a bi-directional and dynamical network, namely *Deep Predictive Coding Network* (PCN), to run a cascade of local recurrent processing for object recognition. PCN combines predictive coding and local recurrent processing into an iterative inference algorithm. When tested for image classification with benchmark datasets (CIFAR-10, CIFAR-100, SVHN and ImageNet), PCN uses notably fewer layers and parameters to achieve competitive performance relative to classical or state-of-the-art models. Further behavioral analysis of PCN sheds light on its computational mechanism and potential use for mapping visual saliency or bottom-up attention.

2 Related Work

Predictive Coding In the brain, connections between cortical areas are mostly reciprocal [10]. Rao and Ballard suggest that bi-directional connections sub-serve "predictive coding" [38]: feedback connections from a higher cortical area carry neural predictions to the lower cortical area, while the feedforward connections carry the unpredictable information (or error of prediction) to the higher area to correct the neuronal states throughout the hierarchy. With supporting evidence from empirical studies [1, 43, 23], this mechanism enables iterative inference for perception[38] and unsupervised learning[44], incorporates modern neural networks for classification [45] and video prediction [34], and likely represents a unified theory of the brain[11, 20].

Predictive Coding Network with Global Recurrent Processing Driven by the predictive coding theory, a bi-directional and recurrent neural network has been proposed in [53]. It runs global recurrent processing by alternating a bottom-up cascade of feedforward computation and a top-down cascade of feedback computation. For each cycle of recurrent dynamics, the feedback prediction starts from the top layer and propagates layer by layer until the bottom layer; then, the feedforward error starts from the bottom layer and propagates layer by layer until the top layer. The model described herein is similar, but uses local recurrent processing, instead of global recurrent processing. Only for the convenience of notation in this paper, we refer to the proposed PCN with local recurrent processing simply as "PCN", while referring to the model in [53] explicitly as "PCN with global recurrent processing".

Local Recurrent Processing Feedforward-only processing plays a central role in rapid object recognition[40, 8]. Although less understood, feedback connections are thought to convey top-down attention [4, 3] or prediction [11, 38, 43]. Evidence also suggests that feedback signals may operate between hierarchically adjacent areas along the ventral stream [24, 5, 37] to enable local recurrent processing for object recognition [55, 2], especially given ambiguous or degraded visual input [54, 42]. Therefore, feedback processes may be an integral part of both global and local recurrent processing underlying top-down attention in a slower time scale and visual recognition in a faster time scale.

3 Predictive Coding Network

Based on predictive coding, we design a bi-directional (feedforward and feedback) neural network that runs local recurrent processing between neighboring layers. Herein, we refer to this network as Predictive Coding Network (PCN). As illustrated in Fig. 1, a PCN is a stack of recurrent blocks, each running dynamic and recurrent processing within itself through feedforward and feedback connections. Feedback connections are used to predict lower-layer representations. In turn, feedforward connections send the error of prediction to update the higher-layer representations. After repeating this processing for multiple cycles within a given block, the lower-layer representation is merged to the higher-layer representation through a bypass connection. The merged representation is further sent as the input to the next recurrent block to start another series of recurrent processing in a

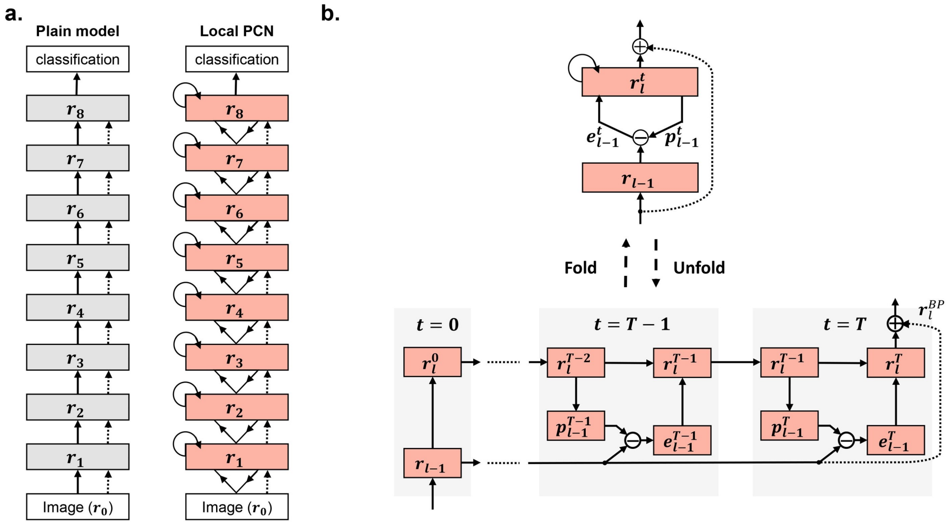

Figure 1: Architecture of CNN vs. PCN (a) The plain model (left) is a feedforward CNN with 3×3 convolutional connections (solid arrows) and 1×1 bypass connections (dashed arrows). On the basis of the plain model, the local PCN (right) uses additional feedback (solid arrows) and recurrent (circular arrows) connections. (b) The PCN consists of a stack of basic building blocks. Each block runs multiple cycles of local recurrent processing between adjacent layers, and merges its input to its output through the bypass connections. The output from one block is then sent to its next block to initiate local recurrent processing in a higher block. It further continues until reaching the top of the network.

higher level. After the local recurrent processing continues through all recurrent blocks in series, the emerging top-level representations are used for image classification.

In the following mathematical descriptions, we use italic letters as symbols for scalars, bold lowercase letters for column vectors and bold uppercase letters for matrices. We use T to denote the number of recurrent cycles, $\boldsymbol{r}_l(t)$ to denote the representation of layer l at time t, $\boldsymbol{W}_{l-1,l}$ to denote the feedforward weights from layer $l-1$ to layer l, $\boldsymbol{W}_{l,l-1}$ to denote the feedback weights from layer l to layer $l-1$ and $\boldsymbol{W}_{l-1,l}^{bp}$ to denote the weights of bypass connections.

3.1 Local Recurrent Processing in PCN

Within each recurrent block (e.g. between layer $l-1$ and layer l), the local recurrent processing serves to reduce the error of prediction. As in Eq. (1), the higher-layer representation $\boldsymbol{r}_l(t)$ generates a prediction, $\boldsymbol{p}_{l-1}(t)$, of the lower-layer representation, $\boldsymbol{r}_{l-1}$, through feedback connections, $\boldsymbol{W}_{l,l-1}$, yielding an error of prediction $\boldsymbol{e}_{l-1}(t)$ as Eq. (2).

$$\boldsymbol{p}_{l-1}(t) = (\boldsymbol{W}_{l,l-1})^T \boldsymbol{r}_l(t) \tag{1}$$

$$\boldsymbol{e}_{l-1}(t) = \boldsymbol{r}_{l-1} - \boldsymbol{p}_{l-1}(t) \tag{2}$$

The objective of recurrent processing is to reduce the sum of the squared prediction error (Eq. (3)) by updating the higher-layer representation, $\boldsymbol{r}_l(t)$, with an gradient descent algorithm [46]. In each cycle of recurrent processing, $\boldsymbol{r}_l(t)$ is updated by taking an additive step in the direction opposite to the gradient (Eq. (4)) and proportional to an update rate, α_l. As $\boldsymbol{r}_l(t)$ is updated over time as Eq. (5), it tends to converge while the gross error of prediction tends to decrease. Note that Eq. (6) is equivalent to Eq. (5), if the feedback weights are tied to be the transpose of the feedforward weights. Eq. (6) is useful even without this assumption as shown in several prior studies [53, 32]. It is thus used in this study instead of Eq. (5).

$$L_{l-1}(t) = \frac{1}{2} \parallel \boldsymbol{r}_{l-1} - \boldsymbol{p}_{l-1}(t) \parallel_2^2 \tag{3}$$

$$\frac{\partial L_{l-1}(t)}{\partial \boldsymbol{r}_l(t)} = -\boldsymbol{W}_{l,l-1}\boldsymbol{e}_{l-1}(t) \tag{4}$$

$$\boldsymbol{r}_l(t+1) = \boldsymbol{r}_l(t) - \alpha_l \frac{\partial L_{l-1}(t)}{\partial \boldsymbol{r}_l(t)} = \boldsymbol{r}_l(t) + \alpha_l \boldsymbol{W}_{l,l-1}\boldsymbol{e}_{l-1}(t) \tag{5}$$

$$\boldsymbol{r}_l(t+1) = \boldsymbol{r}_l(t) + \alpha_l (\boldsymbol{W}_{l-1,l})^T \boldsymbol{e}_{l-1}(t) \tag{6}$$

3.2 Network Architecture

We implement PCN with some architectural features common to modern CNNs. Specifically, feedforward and feedback connections are implemented as regular convolutions and transposed convolutions [9], respectively. They both use a kernel size of 3 while bypass convolutions are implemented as 1×1 convolution. Batch normalization (BN) [21] is applied to the input to each recurrent block. During the recurrent processing between layer $l-1$ and layer l, rectified linear units (ReLU) [35] are applied to the initial output $\boldsymbol{r}_l(0)$ at $t=0$ and the prediction error at each time step $\boldsymbol{e}_{l-1}(t)$, as expressed by Eq. (7) and Eq. (8), respectively. ReLU renders the local recurrent processing an increasingly non-linear operation as it continues over time.

$$\boldsymbol{r}_l(0) = ReLU\left(\left(\boldsymbol{W}_{l-1,l}^T \boldsymbol{r}_{l-1}\right)\right) \tag{7}$$

$$\boldsymbol{e}_{l-1}(t) = ReLU\left(\boldsymbol{r}_{l-1} - \boldsymbol{p}_{l-1}(t)\right) \tag{8}$$

Besides, a 2×2 max-pooling with a stride of 2 is optionally applied to the output from every 2 (or 3) blocks. On top of the highest recurrent block is an classifier, including a global average pooling, a fully-connected layer, followed by softmax.

For comparison, we also design the feedforward-only counterpart of PCN and refer to it as the "plain" model. It includes the same feedforward and bypass connections and uses the same classification layer as in PCN, but excludes any feedback connection or recurrent processing. As a result, the architecture of the plain model is similar to that of Inception CNN [49].

Algorithm 1 Predictive Coding Network with local recurrent processing.

```
Input: The input image $\boldsymbol{r}_0$;
for $l = 1$ to $L$ do
    $\boldsymbol{r}_{l-1}^{BN} = BatchNorm(\boldsymbol{r}_{l-1})$;
    $\boldsymbol{r}_l(0) = ReLU\left(FFConv\left(\boldsymbol{r}_{l-1}^{BN}\right)\right)$;
    for $t = 1$ to $T$ do
        $\boldsymbol{p}_{l-1}(t) = FBConv\left(\boldsymbol{r}_l(t-1)\right)$;
        $\boldsymbol{e}_{l-1}(t) = ReLU\left(\boldsymbol{r}_{l-1} - \boldsymbol{p}_{l-1}(t)\right)$;
        $\boldsymbol{r}_l(t) = \boldsymbol{r}_l(t-1) + \alpha_l FFconv\left(\boldsymbol{e}_{l-1}(t)\right)$;
    end for
    $\boldsymbol{r}_l = \boldsymbol{r}_l(T) + BPConv\left(\boldsymbol{r}_{l-1}^{BN}\right)$;
end for
return $\boldsymbol{r}_L$ for classification;
▷ FFconv represents the feedforward convolution, FBconv represents the feedback convolution
  and BPconv represents the bypass convolution.
```

Our implementation is described in Algorithm 1. Note that the update rate α_l used for local recurrent processing is a learnable and non-negative parameter separately defined for each filter in each recurrent block. The number of cycles of recurrent processing - an essential parameter for PCN, is varied to be $T = 1, ..., 5$. For both PCN and its CNN counterpart (i.e. T=0), we design multiple different architectures (labeled as A through E) suitable for different benchmark datasets (SVHN, CIFAR and ImageNet), as summarized in Table 1. For example, PCN-A-5 stands for a PCN with architecture A and 5 cycles of local recurrent processing.

Table 1: Architectures of PCN. Each column shows a model. We use PcConv to represent a predictive coding layer, with its parameters denoted as "PcConv<kernel size>-<number of channels in feedback convolutions>-<number of channels in feedforward convolutions>". The first layer in PCN-E is a regular convolution with a kernel size of 7, a padding of 3, a stride of 2 and 64 output channels. "*" indicates the layer applying maxpooling to its output and feature maps in one grid have the same size.

PCN Configuration					
Dataset	SVHN		CIFAR		ImageNet
Architecture	A	B	C	D	E
#Layers	7		9		13
Image Size	32 × 32				224 × 224
Layers	PcConv3-3-16 PcConv3-16-16	PcConv3-3-16 PcConv3-16-32	PcConv3-3-64 PcConv3-64-64	PcConv3-3-64 PcConv3-64-64	Conv7-64 PcConv3-64-64
	PcConv3-16-32* PcConv3-32-32	PcConv3-32-64* PcConv3-64-64	PcConv3-64-128* PcConv3-128-128	PcConv3-64-128* PcConv3-128-128	PcConv3-64-128* PcConv3-128-128
	PcConv3-32-64* PcConv3-64-64	PcConv3-64-128* PcConv3-128-128	PcConv3-128-256* PcConv3-256-256	PcConv3-128-256* PcConv3-256-256	PcConv3-128-128* PcConv3-128-128
			PcConv3-256-256 PcConv3-256-256	PcConv3-256-512 PcConv3-512-512	PcConv3-128-256* PcConv3-256-256 PcConv3-256-256
					PcConv3-256-512* PcConv3-512-512 PcConv3-512-512
Calssification	global average pooling, FC-10/100/1000, softmax				
#Params	0.15M	0.61M	4.91M	9.90M	17.26M

4 Experiments

We train PCN with local recurrent processing and its corresponding plain model for object recognition with the following datasets, and compare their performance with classical or state-of-the-art models.

4.1 Datasets

CIFAR The CIFAR-10 and CIFAR-100 datasets [26] consist of 32×32 color images drawn from 10 and 100 categories, respectively. Both datasets contain 50,000 training images and 10,000 testing images. For preprocessing, all images are normalized by channel means and standard derivations. For data augmentation, we use a standard scheme (flip/translation) as suggested by previous works [16, 18, 33, 39, 19].

SVHN The Street View House Numbers (SVHN) dataset [36] consists of 32×32 color images. There are 73,257 images in the training set, 26,032 images in the test set and 531,131 images for extra training. Following the common practice [18, 33], we train the model with the training and extra sets and tested the model with the test set. No data augmentation is introduced and we use the same pre-processing scheme as in CIFAR.

ImageNet The ILSVRC-2012 dataset [7] consists of 1.28 million training images and 50k validation images, drawn from 1000 categories. Following [16, 17, 49], we use the standard data augmentation scheme for the training set: a 224×224 crop is randomly sampled from either the original image or its horizontal flip. For testing, we apply a single crop or ten crops with size 224×224 on the validation set and the top-1 or top-5 classification error is reported.

4.2 Training

Both PCN and (plain) CNN models are trained with stochastic gradient descent (SGD). On CIFAR and SVHN datasets, we use 128 batch-size and 0.01 initial learning rate for 300 and 40 epochs, respectively. The learning rate is divided by 10 at 50%, 75% and 87.5% of the total number of training epochs. Besides, we use a Nesterov momentum of 0.9 and a weight decay of 1e-3, which is determined by a 45k/5k split on the CIFAR-100 training set. On ImageNet, we follow the most common practice [16, 56] and use a initial learning rate of 0.01, a momentum 0f 0.9, a weight decay of 1e-4, 100 epochs with the learning rate dropped by 0.1 at epochs 30, 60 & 90. The batch size is 256 for PCN-E-0/1/2, 128 for PCN-E-3/4, 115 for PCN-E-5 due to limited computational resource.

Table 2: Error rates (%) on CIFAR datasets. #L and #P are the number of layers and parameters, respectively.

Type	Model	#L	#P	C-10	C-100
Feed-Forward Models	HighwayNet [47]	19	2.3M	7.72	32.39
	FractalNet [28]	21	38.6M	5.22	23.30
	ResNet [16]	110	1.7M	6.41	27.22
	ResNet [17] (Pre-act)	164	1.7M	5.23	24.58
		1001	10.2M	4.62	22.71
	WRN [59]	28	36.5M	4.00	19.25
	DenseNet BC [19]	100	0.8M	4.51	22.27
		190	25.6M	**3.46**	**17.18**
Recurrent Models	RCNN [30]	160	1.86M	7.09	31.75
	DasNet [48]	-	-	9.22	33.78
	FeedbackNet [60]	12	-	-	28.88
	CliqueNet [58]	18	10.14M	5.06	23.14
		30	10.02M	**5.06**	**21.83**
	PCN [53] with Global Recurrent Processing	7	0.57M	7.60	31.69
		9	1.16M	7.20	30.66
		9	4.65M	6.17	27.42
PCN	PCN-C-5	9	4.91M	5.10	22.43
	PCN-D-5	9	9.90M	**4.89**	**21.77**
Plain Models	Plain-C	9	2.59M	5.68	25.65
	Plain-D	9	5.21M	5.61	25.31

Table 3: Error rates (%) on SVHN. #L and #P are the number of layers and parameters, respectively.

Model	#L	#P	SVHN
MaxOut [13]	-	-	2.47
NIN [33]	-	-	2.35
DropConnect [51]	-	-	1.94
DSN [29]	-	-	1.92
RCNN [30]	6	2.67M	1.77
FitNet [39]	13	1.5M	2.42
WRN [59]	16	11M	1.54
PCN (Global) [53]	7	0.14M	2.42
	7	0.57M	2.42
PCN-A-5	7	0.15M	2.07
PCN-B-5	7	0.61M	1.96
Plain-A	7	0.08M	2.85
Plain-B	7	0.32M	2.43

4.3 Evaluating the Behavior of PCN

To understand how the model works, we further examine how local recurrent processing changes the internal representations of PCN. For this purpose, we focus on testing PCN-D-5 with CIFAR-100.

Converging representation? Since local recurrent processing is governed by predictive coding, it is anticipated that the error of prediction tends to decrease over time. To confirm this expectation, the L2 norm of the prediction error is calculated for each layer and each cycle of recurrent processing and averaged across all testing examples in CIFAR-100. This analysis reveals the temporal behavior of recurrently refined internal representations.

What does the prediction error mean? Since the error of prediction drives the recurrent update of internal representations, we exam the spatial distribution of the error signal first for each layer and then average the error distributions across all layers after rescaling them to the same size. The resulting error distribution is used as a spatial mask applied to the input image to reveal its computational role.

Does predictive coding help image classification? The goal of predictive coding is to reduce the error of top-down prediction, seemingly independent of the objective of categorization that the PCN model is trained for. As recurrent processing progressively updates layer-wise representation, does each update also sub-serve the purpose of categorization? As in [22], the loss of categorization, as a nonlinear function of layer-wise representation $\mathcal{L}(\boldsymbol{r}_l(t))$, shows its Taylor's expansion as Eq. (9), where $\Delta \boldsymbol{r}_l(t) = \boldsymbol{r}_l(t+1) - \boldsymbol{r}_l(t)$ is the incremental update of $\boldsymbol{r}_l$ at time t.

$$\mathcal{L}(\boldsymbol{r}_l(t+1)) - \mathcal{L}(\boldsymbol{r}_l(t)) = \Delta \boldsymbol{r}_l(t) \cdot \frac{\partial \mathcal{L}(\boldsymbol{r}_l(t))}{\partial \boldsymbol{r}_l(t)} + \mathcal{O}(\Delta \boldsymbol{r}_l(t)^2) \tag{9}$$

If each update tends to reduce the categorization loss, it should satisfy $\Delta \boldsymbol{r}_l(t) \cdot \frac{\partial \mathcal{L}(\boldsymbol{r}_l(t))}{\partial \boldsymbol{r}_l(t)} < 0$, or the "cosine distance" between $\Delta \boldsymbol{r}_l(t)$ and $\frac{\partial \mathcal{L}(\boldsymbol{r}_l(t))}{\partial \boldsymbol{r}_l(t)}$ should be negative. Here $\mathcal{O}(\cdot)$ is ignored as suggested by [22], since the incremental part becomes minor because of the convergent representation in PCN. We test this by calculating the cosine distance for each cycle of recurrent processing in each layer, and then averaging the results across all testing images in CIFAR-100.

4.4 Experimental results

Classification performance On CIFAR and SVHN datasets, PCN always outperforms its corresponding CNN counterpart (Table 2 and 3). On CIFAR-100, PCN-D-5 reduces the error rate by 3.54% relative to Plain-D. The performance of PCN is also better than those of the ResNet family [16, 17], although PCN is nominally much shallower with only 9 layers whereas ResNet may use as many as 1001 layers. Although PCN under-performs WRN [59] and DenseNet-190 [19] by 2-4%, it uses a much shallower architecture with many fewer parameters. On SVHN, PCN also shows competitive performance despite fewer layers and parameters. On ImageNet, PCN also performs better

Table 4: Error rates (%) on ImageNet.

Model	#Layers	#Params	Single-Crop		10-crop	
			top-1	top-5	top-1	top-5
ResNet-18	18	11.69M	30.24	10.92	28.15	9.40
ResNet-34	34	21.80M	26.69	8.58	24.73	7.46
ResNet-50	50	25.56M	23.87	7.14	22.57	6.24
PCN-E-5	13	17.26M	25.31	7.79	23.52	6.64
PCN-E-3			25.36	7.78	23.62	6.69
Plain-E	13	9.34M	31.15	11.27	28.82	9.79

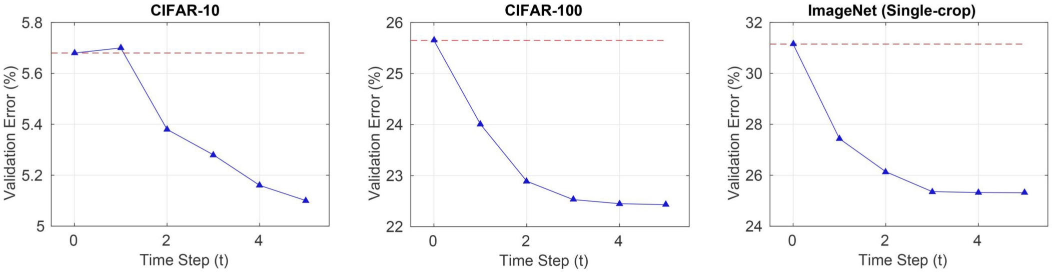

Figure 2: PCN shows better categorization performance given more cycles of recurrent processing, for CIFAR-10, CIFAR-100 and ImageNet. The red dash line represents the accuracy of the plain model.

than its plain counterpart. With 5 cycles of local recurrent processing (i.e. PCN-E-5), PCN slightly under-performs ResNet-50 but outperforms ResNet-34, while using fewer layers and parameters than both of them. Therefore, the classification performance of PCN compares favorably with other state-of-the-art models especially in terms of the performance-layer ratio.

Recurrent Cycles The classification performance generally improves as the number of cycles of local recurrent processing increases for both CIFAR and ImageNet datasets, and especially for ImageNet (Fig. 2). It is worth noting that this gain in performance is achieved without increasing the number of layers or parameters, but by simply running computation for longer time on the same network.

Local vs. Global Recurrent Processing The performance comparison between PCN with local and global recurrent processing suggests that the former (proposed in this paper) performs better than the latter (proposed elsewhere[53]. Table 2 shows that PCN with local recurrent processing reduces the error rate by 5% on CIFAR-100 compared to PCN with the same architecture but global recurrent processing. In addition, local recurrent processing also requires less computational resource for both training and testing the model .

4.5 Behavioral Analysis

To understand how/why PCN works, we further examine how recurrent processing changes its internal representations and whether the change helps categorization. Behavioral analysis reveals some intriguing findings.

As expected for predictive coding, local recurrent processing progressively reduces the error of top-down prediction for all layers, except the top layer on which image classification is based (Fig. 3a). This finding implies that the internal representations converge to a stable state by local recurrent processing. Interestingly, the spatial distribution of the prediction error (averaged across all layers) highlights the apparently most salient part of the input image, and/or the most discriminative visual information for object recognition (Fig. 3b). The recurrent update of layer-wise representation tends to align along the negative gradient of the categorization loss with respective to the corresponding representation (Fig. 3c). From a different perspective, this finding lends support to an intriguing implication that predictive coding facilitates object recognition, which is somewhat surprising because predictive coding, as a computational principle, herein only explicitly reduces the error of top-down prediction without having any explicit role in computation or learning towards object categorization.

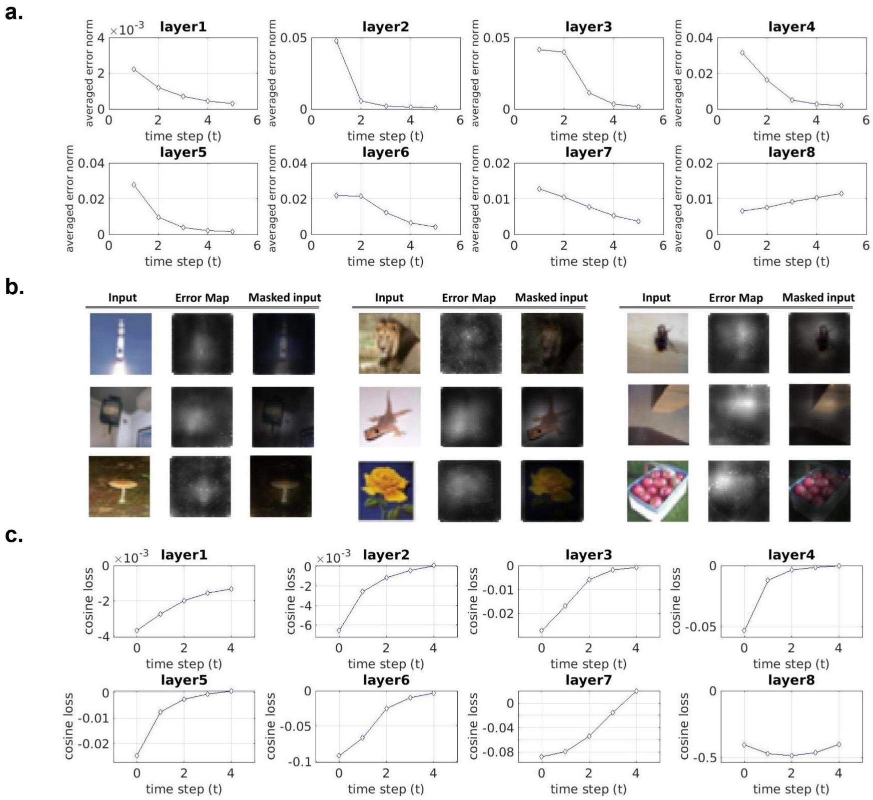

Figure 3: Behavioral analysis of PCN. (a) Errors of prediction tend to converge over repeated cycles of recurrent processing. The norm of the prediction error is shown for each layer and each cycle of local recurrent processing. (b) Errors of prediction reveal visual saliency. Given an input image (left), the spatial distribution of the error averaged across layers (middle) shows visual saliency or bottom-up attention, highlighting the part of the input with defining features (right). (3) The averaged cosine loss between $\Delta \boldsymbol{r}_l(t)$ and $\frac{\partial \mathcal{L}(\boldsymbol{r}_l(t))}{\partial \boldsymbol{r}_l(t)}$.

5 Discussion and Conclusion

We advocate further synergy between neuroscience and artificial intelligence. There are more to learn from the brain in order to build better AI. Complementary to engineering innovation in computing and optimization, the brain must possess hidden tricks to enable generalizable, continuous, and efficient learning and inference. In this work, we highlight the fact that the brain is a dynamical system. It runs recurrent processing, instead of serial processing. Using this as an inspiration, we should try to reuse building blocks in a network model, as opposed to adding blocks to it. The former is arguably a better strategy [31] because the model does not need to use more parameters in order to implement a wide range of complex transformation for many different intelligent tasks.

As one example of this strategy, we present a predictive coding network with local recurrent processing, as a variation of a similar model with global recurrent processing proposed earlier [53]. For object recognition, PCN has several advantages. 1) It can achieve competitive performance in image classification with a shallow network and fewer parameters. 2) Its internal representations converge as recurrent processing proceeds over time, suggesting a self-organized mechanism towards stability [11]. 3) It reveals visual saliency or bottom-up attention while performing object recognition. However, its disadvantages are 1) the longer time of computation than plain networks and 2) the sequentially executed recurrent processing, both of which should be improved or addressed in future studies.

Acknowledgement

The research was supported by NIH R01MH104402 and Purdue University.

References

[1] Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

[2] CN Boehler, MA Schoenfeld, H-J Heinze, and J-M Hopf. Rapid recurrent processing gates awareness in primary visual cortex. *Proceedings of the National Academy of Sciences*, 105(25): 8742–8747, 2008.

[3] Steven L Bressler, Wei Tang, Chad M Sylvester, Gordon L Shulman, and Maurizio Corbetta. Top-down control of human visual cortex by frontal and parietal cortex in anticipatory visual spatial attention. *Journal of Neuroscience*, 28(40):10056–10061, 2008.

[4] Elizabeth A Buffalo, Pascal Fries, Rogier Landman, Hualou Liang, and Robert Desimone. A backward progression of attentional effects in the ventral stream. *Proceedings of the National Academy of Sciences*, 107(1):361–365, 2010.

[5] Joan A Camprodon, Ehud Zohary, Verena Brodbeck, and Alvaro Pascual-Leone. Two phases of v1 activity for visual recognition of natural images. *Journal of cognitive neuroscience*, 22(6): 1262–1269, 2010.

[6] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4470–4478, 2017.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[8] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

[9] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[10] Daniel J Felleman and DC Essen Van. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[11] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127, 2010.

[12] Dileep George and Jeff Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS computational biology*, 5(10):e1000532, 2009.

[13] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[14] Umut Güçlü and Marcel AJ van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35 (27):10005–10014, 2015.

[15] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5353–5360. IEEE, 2015.

[16] 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*, pages 770–778, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[18] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.

[19] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[20] Yanping Huang and Rajesh PN Rao. Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5):580–593, 2011.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[22] Stanisław Jastrzebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. *arXiv preprint arXiv:1710.04773*, 2017.

[23] Janneke FM Jehee, Constantin Rothkopf, Jeffrey M Beck, and Dana H Ballard. Learning receptive fields using predictive feedback. *Journal of Physiology-Paris*, 100(1-3):125–132, 2006.

[24] Mika Koivisto, Henry Railo, Antti Revonsuo, Simo Vanni, and Niina Salminen-Vaparanta. Recurrent processing in v1/v2 contributes to categorization of natural scenes. *Journal of Neuroscience*, 31(7):2488–2492, 2011.

[25] Nikolaus Kriegeskorte. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1:417–446, 2015.

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[29] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

[30] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.

[31] Qianli Liao and Tomaso Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016.

[32] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7:13276, 2016.

[33] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[34] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

[35] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[37] Randall C O'Reilly, Dean Wyatte, Seth Herd, Brian Mingus, and David J Jilk. Recurrent processing during object recognition. *Frontiers in psychology*, 4:124, 2013.

[38] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79, 1999.

[39] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[40] Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. A quantitative theory of immediate visual recognition. *Progress in brain research*, 165: 33–56, 2007.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Courtney J Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8: 1551, 2017.

[43] Michael W Spratling. Reconciling predictive coding and biased competition models of cortical function. *Frontiers in computational neuroscience*, 2:4, 2008.

[44] Michael W Spratling. Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural computation*, 24(1):60–103, 2012.

[45] Michael W Spratling. A hierarchical predictive coding model of object recognition in natural images. *Cognitive computation*, 9(2):151–167, 2017.

[46] Michael W Spratling. A review of predictive coding algorithms. *Brain and cognition*, 112: 92–97, 2017.

[47] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[48] Marijn F Stollenga, Jonathan Masci, Faustino Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *Advances in neural information processing systems*, pages 3545–3553, 2014.

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[50] David C Van Essen and John HR Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in neurosciences*, 6:370–375, 1983.

[51] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.

[52] Haiguang Wen, Junxing Shi, Yizhen Zhang, Kun-Han Lu, Jiayue Cao, and Zhongming Liu. Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral Cortex*, pages 1–25, 2017.

[53] Haiguang Wen, Kuan Han, Junxing Shi, Yizhen Zhang, Eugenio Culurciello, and Zhongming Liu. Deep predictive coding network for object recognition. *arXiv preprint arXiv:1802.04762*, 2018.

[54] Dean Wyatte, Tim Curran, and Randall O'Reilly. The limits of feedforward vision: recurrent processing promotes robust object recognition when objects are degraded. *Journal of Cognitive Neuroscience*, 24(11):2248–2261, 2012.

[55] Dean Wyatte, David J Jilk, and Randall C O'Reilly. Early recurrent feedback facilitates visual object recognition under challenging conditions. *Frontiers in psychology*, 5:674, 2014.

[56] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[57] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.

[58] Yibo Yang, Zhisheng Zhong, Tiancheng Shen, and Zhouchen Lin. Convolutional neural networks with alternately updated clique. *arXiv preprint arXiv:1802.10419*, 2018.

[59] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[60] Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817. IEEE, 2017.