A Deep Predictive Coding Network for Inferring Hierarchical Causes Underlying Sensory Inputs

Shirin Dora[1](✉), Cyriel Pennartz[1], and Sander Bohte[2]

[1] University of Amsterdam, Amsterdam, Netherlands
shirin.dora@gmail.com, c.m.a.pennartz@uva.nl
[2] Centrum Wiskunde and Informatica, Amsterdam, Netherlands
S.M.Bohte@cwi.nl

Abstract. Predictive coding has been argued as a mechanism underlying sensory processing in the brain. In computational models of predictive coding, the brain is described as a machine that constructs and continuously adapts a generative model based on the stimuli received from external environment. It uses this model to infer causes that generated the received stimuli. However, it is not clear how predictive coding can be used to construct deep neural network models of the brain while complying with the architectural constraints imposed by the brain. Here, we describe an algorithm to construct a deep generative model that can be used to infer causes behind the stimuli received from external environment. Specifically, we train a deep neural network on real-world images in an unsupervised learning paradigm. To understand the capacity of the network with regards to modeling the external environment, we studied the causes inferred using the trained model on images of objects that are not used in training. Despite the novel features of these objects the model is able to infer the causes for them. Furthermore, the reconstructions of the original images obtained from the generative model using these inferred causes preserve important details of these objects.

Keywords: Predictive coding · Deep generative models

1 Introductions

Predictive coding has been proposed as a theory of sensory information processing in which the brain infers causes that generated a sensory stimulus [1, 2]. It postulates that the top-down flow of information in the brain serve as predictions of the inferred causes of a stimulus at a lower level and the bottom-up flow of information conveys the errors in these predictions to the higher areas. Rao and Ballard [3] proposed the first neural network model of predictive coding for the processing of visual information in the brain. Their model consisted of a recurrently connected neural network with three layers.

Several studies have focused on the biological plausibility of the initial model of predictive coding that was proposed by Rao and Ballard (hereafter, referred simply as predictive coding) and its relation with other existing approaches. In [4], Spratling

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11141, pp. 457–467, 2018.
https://doi.org/10.1007/978-3-030-01424-7_45

showed that a model of biased competition [5] that uses lateral inhibition to suppress the input of other nodes is equivalent to the linear model of predictive coding. An extension to predictive coding has been proposed in [6] that relaxes the requirement of symmetric weights between two adjacent layers in the network. In a similar study, it was shown that error-backpropagation and predictive coding use similar forms of weight changes during learning [7].

From the perspective of training deep neural networks, predictive coding is an approach that is widely supported by neurophysiological data [8] and adheres to the locality (in terms of learning) constraints [3] imposed by the brain. Previous studies on predictive coding focused on small neural network models to study the development of orientation selective receptive fields in primary visual cortex [3, 6]. It is unclear how predictive coding can be used to build deep neural network models of the brain to study more complicated brain processes like perception, attention, memory, etc. An important question in this regard is how to comply with the architectural constraints applicable in the brain like the retinotopic arrangement of receptive fields that is found in the sensory cortical areas. At present, mostly neural networks with fully connected layers are used, which implies that all neurons have the same receptive field which encompasses the entire input stimulus. To overcome this, predictive coding models are often trained on patches from real world images. This approach works well when training small neural network models but it is difficult to extend it for training deep neural networks.

In this paper, we present a systematic approach for training deep neural networks using predictive coding in a biologically plausible manner. Our goal is to construct a deep neural network model to infer hierarchical (here, hierarchical refers to causes inferred at each layer in the network) causes for a given input stimulus. The architecture of these neural networks is inspired by convolutional neural network. However, to comply with the retinotopic arrangement of receptive fields observed in sensory areas, we employ neural networks in which filters are *not* applied across the entire layer. Instead, filters are applied only to a small receptive field which allows us to train the filters associated with different receptive fields independently. This approach can be easily scaled to construct deep predictive coding models for information processing along the sensory processing pathways.

We trained a deep neural network using predictive coding on 1000 real-world images of horses and ships from the CIFAR-10 data set. The model is trained in an unsupervised learning paradigm to build a generative model for real-world images. To estimate the capacity of the network in modeling real-world images, we used the model to infer hierarchical causes for new images of horses and ships as well as objects that had never been presented before to the network during training. The causes inferred by the model can be used to reconstruct the original real-world images while retaining the important features of the objects in these images. This shows that the model is able to capture the statistical regularities generally present in the real-world images. This allows the trained network to infer causes for images with objects that have never been presented before to the network. This attribute of the network also enables it to infer causes for images that are translated versions of images of horses and ships used in training as well as images of new objects.

The paper is organized as follows: Sect. 2 describes the architecture and the predictive coding based learning algorithm used for training deep neural networks.

Section 3 describes the results of studies conducted using the trained models. Section 4 discusses the computational implications of deep predictive coding and its relationship with other approaches in machine learning. Section 5 summarizes the conclusions from our modelling work and experiments.

2 Model

Suppose, we have a set of training images $(\boldsymbol{x}_1, \ldots \boldsymbol{x}_i, \ldots)$ where $\boldsymbol{x}_i \in \boldsymbol{R}^{\boldsymbol{W} \times \boldsymbol{H} \times \boldsymbol{C}}$. The aim of the learning algorithm is to construct a deep neural network that can be used to infer causes for real-world images presented to the network.

2.1 Architecture

Consider a neural network with $(N+1)$ layers with 0 being the input layer and N being the topmost layer in the network. The input layer is used to present the training images to the network. Figure 1 shows a section of this network that depicts the recurrent connections between layer l and layers above $(l+1)$ and below $(l-1)$ it. The neurons in a given layer l are arranged in a 3-dimensional block of shape $Y_l \times X_l \times K_l$. Here, Y_l, X_l and K_l denote the height, width and the number of channels in layer l, respectively. The neurons in layers l and $(l+1)$ are connected through K_{l+1} filters of size D_l and a stride of s_l. Based on this, the height and width of layer $(l+1)$ are given as

$$Y_{l+1} = (\mathrm{Y_l} - \mathrm{D_l})/s_l + 1 \tag{1}$$

$$X_{l+1} = (X_l - D_l)/s_l + 1 \tag{2}$$

The number of channels in layer $(l+1)$ is equal to the number of filters between layers l and $(l+1)$.

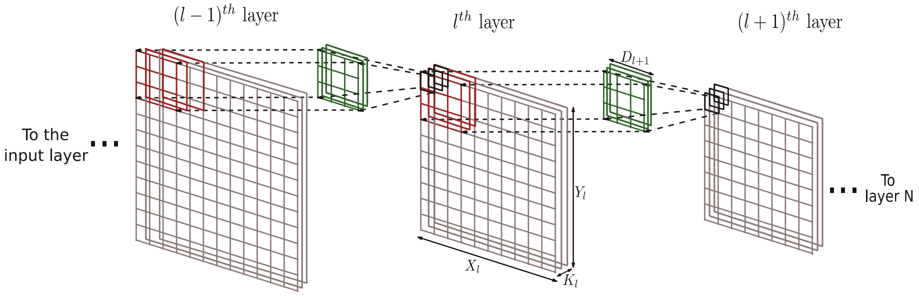

Fig. 1. Architecture of the deep predictive coding network

The architecture of the network in Fig. 1 bears some resemblance to the architecture of Convolutional Neural Networks (CNNs). However, there are two important differences between CNNs and the neural network used in this paper:

- The neurons in a given layer in the network, shown in Fig. 1, are recurrently connected to the neurons only in the corresponding receptive field. This implies that the filters for all the receptive fields in a particular layer are learnt independently.
- The most important difference with respect to CNNs lies in the direction of information propagation. In a conventional CNN, the information propagates from layer 0 to layer N and during learning the error gradients propagate from layer N to layer 0. In contrast, in our predictive coding network the predictive information (Fig. 2) propagates from layer N to layer 0 in the network and the error gradients propagate in the opposite direction. Furthermore, in a CNN both information and error gradients propagate serially (layer-by-layer) whereas in the deep predictive coding network these two processes occur in parallel across all layers in the network. Each layer in the network transmits predictions along the feedback pathway to the layer below and receives the prediction errors from the layer below along the feedforward pathway.

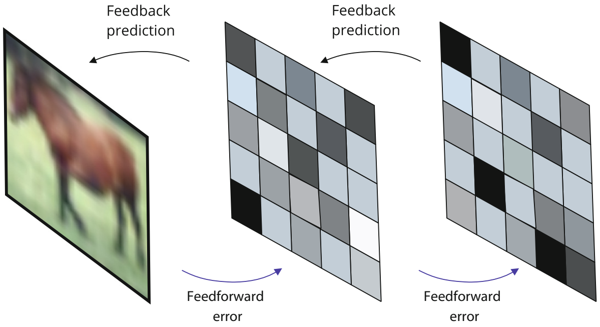

Fig. 2. Direction of information propagation and error gradients in the deep predictive coding network

To better understand the structure of recurrent connections between layer l and layer $(l-1)$, let us denote the activities of the neurons in the m^{th} row and the n^{th} column (here, referred to as (m, n)) of layer l as $y^{(l)}_{m,n}$ which is a vector with K_l elements. Here, the activities of neurons in layer l represent the causes behind the activities of neurons in layer $(l-1)$. Based on this, the feedback predictions generated by the neurons in layer l for the activities of neurons in layer $(l-1)$ are given as

$$\hat{y}^{(l-1)}_{(s_{l-1}m+i),(\mathrm{s}_{\mathrm{l-1}}\mathrm{n}+\mathrm{j})} = \phi\left(\mathrm{w}^{(\mathrm{l})}_{m,n,i,j}\mathrm{y}^{(\mathrm{l})}_{\mathrm{m,n}}\right), \quad \begin{matrix} \mathrm{i,j} \in \left\{1,\ldots,\mathrm{D}_{(\mathrm{l-1})}\right\}, \\ \mathrm{m} \in \left\{1,\ldots,\mathrm{Y}_{\mathrm{l}}\right\}, \mathrm{n} \in \left\{1,\ldots,\mathrm{X}_{\mathrm{l}}\right\} \end{matrix} \tag{3}$$

where $\boldsymbol{w}^{(l)}_{m,n,i,j}$ denotes the filters through which the neurons at position (m, n) in layer l project to the position $(s_{l-1}m + i, s_{l-1}n + j)$ in layer $(l - 1)$. The filter $w^{(l)}_{m,n,i,j}$ will be a matrix with dimensions $K_{l-1} \times K_l$. ϕ represents a non-linear vector-valued activation function with K_{l-1} elements.

It may be noted that when the stride is less than the filter size, this results in an architecture with overlapping receptive fields. As a result, neurons in layer l generate predictions for overlapping receptive fields in layer $(l - 1)$. Therefore, the predicted activity of neurons in layer $(l - 1)$ is computed by taking the mean of the predictions across overlapping receptive fields.

2.2 Learning Algorithm

We use the classical methodology of predictive coding [3] to train a deep neural network model that can be used to infer the hierarchical causes of a given input image. For a given input image $(\boldsymbol{x}_i)$, the activities of the neurons in layer l of the network are inferred such that they can predict (using Eq. 3) the activities of the neurons in layer $(l - 1)$. The activities inferred in layer l of the network serve as target for inferring the activities in layer $(l + 1)$ of the network.

Suppose y_l and $\hat{y}_l$ represent the actual and predicted activities of the neurons in layer l of the network, then the total error (E) for all layers in the network is given as

$$E = \sum_{l=0}^{N} \left(\ell_p\left(\boldsymbol{y}^{(l)} - \hat{\boldsymbol{y}}^{(l)}\right) + \ell_p\left(\boldsymbol{y}^{(l)}\right) + \sum_{m,n,i,j} \ell_p\left(w^{(l)}_{m,n,i,j}\right) \right) \tag{4}$$

where $\ell_p(.)$ denotes the error computed in accordance with p-norm. The total error in Eq. 4 includes both errors, the prediction error and the regularization error.

The total error in Eq. 4 is minimized in order to simultaneously infer the activities and learn the synaptic weights in the network. This implies that the neuronal activities inferred at a particular layer in the network represent the causes behind activities of neurons in the layer below. This allows us to infer hierarchical causes for a given image presented to the network. To explicitly include the aspect of retinotopic arrangement of receptive fields, the total error in Eq. 4 is expanded as

$$E = \sum_{l=0}^{N} \left(\sum_{\mathrm{m,n}}^{\mathrm{Y_l,X_l}} \ell_p\left(y^{(l)}_{\mathrm{m,n}} - \hat{y}^{(l)}_{m,n}\right) + \sum_{\mathrm{m,n}}^{\mathrm{Y_l,X_l}} \ell_p\left(y^{(l)}_{m,n}\right) + \sum_{\mathrm{m,n,i,j}} \ell_p\left(w^{(l)}_{m,n,i,j}\right) \right) \tag{5}$$

Using gradient descent on the error function in Eq. 5, the activities of neurons at a given position (m, n) in layer l are adapted as

$$\begin{aligned} \Delta y^{(l)}_{m,n} = \epsilon_{bu} \left(\sum_{i=1,j=1}^{D_{(l-1)}} \ell'_p\left(y^{(l-1)}_{(m+i),(n+j)} - \hat{y}^{(l-1)}_{(m+i),(n+j)}\right) \phi'\left(\boldsymbol{w}^{(l)}_{m,n,i,j} y^{(l)}_{m,n}\right) \left(\boldsymbol{w}^{(l)}_{m,n,i,j}\right)^T \right) \\ - \epsilon_{td}\left(y^{(l)}_{m,n} - \hat{y}^{(l)}_{m,n}\right) - \epsilon_p \ell'_p\left(y^{(l)}_{m,n}\right) \end{aligned} \tag{6}$$

where $\ell_p'(.)$ denotes partial differentiation of p-norm. ϵ_{bu} is the bottom-up learning rate, ϵ_{td} is the top-down learning rate and ϵ_p is the learning rate for regularization. For a given layer l, the bottom-up learning rate helps in inferring activities that can make better predictions about the activities of the neurons in layer $(l-1)$ and the top-down learning rate helps in ensuring that the inferred activities can be easily predicted by layer $(l+1)$. Together with regularization, these update terms help in inferring causes with sparsely active neurons and provide numerical stability to the learning algorithm.

The filters in the network are also learnt by performing gradient descent along the error function in Eq. 5. The filters are adapted using the learning rule below

$$\Delta \boldsymbol{w}^{(l)}_{m,n,i,j} = \epsilon_w \left(\ell^{\varepsilon}_{\mathrm{p}} \left(y^{(l-1)}_{(m+i),(n+j)} - \hat{y}^{(l-1)}_{(m+i),(n+j)} \right) \phi' \left(\boldsymbol{w}^{(l)}_{m,n,i,j} y^{(l)}_{m,n} \right) \left(y^{(l)}_{m,n} \right)^T \right) \\ - \epsilon_p \left(\boldsymbol{w}^{(l)}_{m,n,i,j} \right) \tag{7}$$

where ϵ_w is the learning rate.

For adapting the neuronal activities and the filters simultaneously, we employ the approach described in [3]. At first the filters are held constant and the neuronal activities are adapted using κ update steps in accordance with Eq. 6 and then we update filters once using the update rule in Eq. 7.

3 Experiments

In this section, we study the capabilities of the network in inferring the hierarchical causes for a given input image. First, we will study the capabilities of the generative model in reconstructing the original images from the inferred causes. Second, we analyze the model's abilities in inferring the causes for a new image that was not used in training. Finally, we study the capability of the model to infer causes for an image that is a translated version of the original image. For this purpose, we trained a 6-layered

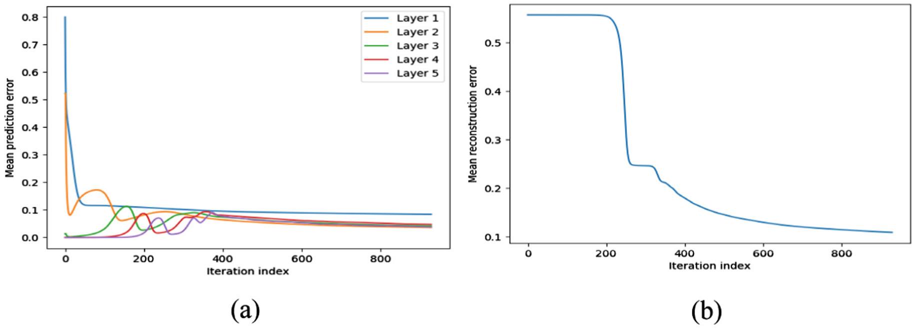

Fig. 3. (a) Mean prediction error at each layer in the network during training. (b) Mean reconstruction error during training. The reconstruction error is based on the images reconstructed by the model using the causes inferred at the topmost layer (as described in Sect. 3.1).

(including the input layer which is referred as the 0^{th} layer in the following sections) neural network on 1000 images of horses and ships from the CIFAR-10 data set. Figure 3 shows the mean prediction error at each layer in the network as well as the mean reconstruction error during training.

3.1 Generative Model

In this section, we study whether the causes inferred at different layers in the network are able to capture the information present in the input image. For a given layer l, we set the activities of the neurons in that layer to the inferred causes. Then, the neurons in layer l predict the activities of neurons in layer $(l-1)$ through the feedback pathways (see Fig. 2). The predicted activities of neurons in layer $(l-1)$ are used to compute the activities of neurons in layer $(l-2)$. This process is repeated across all layers below layer l to compute the activities of neurons in layer 0. If the inferred activations are able to capture the information in the input images then the activities of neurons in layer 0 will provide a closer reconstruction of the original image.

Figure 4 presents some examples of the images reconstructed using the inferred causes at each layer in the network. It can be observed that the images reconstructed by the model are blurry. This is a known problem with the mean square error for computing the error [9]. It may be possible to obtain visually better images using l1-norm, as suggested in [10]. This will be a future direction of research.

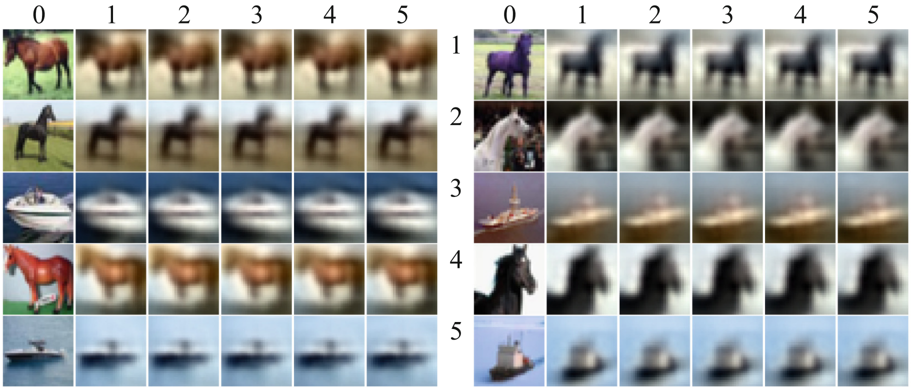

Fig. 4. Examples of images reconstructed by the network when the activities of neurons in different layers of the network have been set to the inferred causes. Each panel (left and right) contains 5 rows. Each row contains six images. The first image in each row is the original image and the following 5 images are reconstructed using the causes inferred in 5 layers of the network. The layer in which the activities of the neurons were set to the inferred causes is shown at the top. The numbers in the center denote the index of the example in the left and right panels.

3.2 Capacity to Represent Novel Input Patterns

To understand, whether the trained model can truly capture the statistical regularities of the real-world images, we used the trained network to infer causes of images from the CIFAR-10 data set that were not used in training. The set of images used included images of objects like airplanes, dogs, birds, etc. which were never presented to the network during training. Note that we used the trained network only for inferring the causes. The filters are no longer adapted in this network.

The inferred causes for the new images are used to reconstruct the original images as described in Sect. 3.1. Figure 5 presents some examples of the images reconstructed from the causes inferred using predictive coding. It can be seen that the network can infer causes even for images that contain objects which were never presented before to the network. This clearly shows the model captures the statistical regularities present in real-world images.

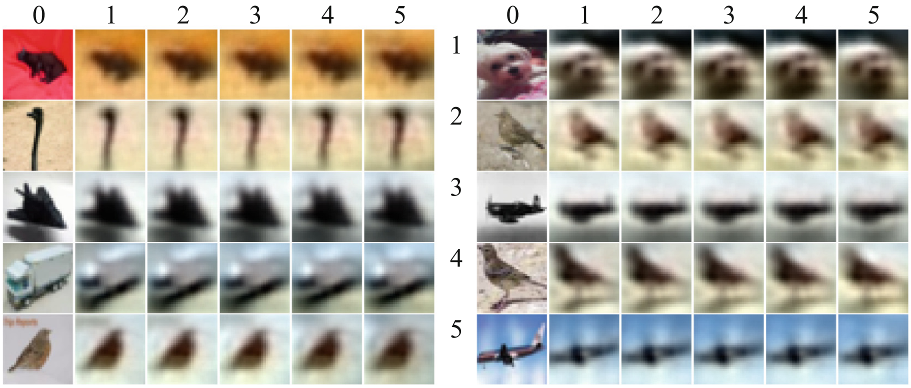

Fig. 5. Non-training images reconstructed by the network using the causes inferred for these images. These images are also arranged in 2 panels, each containing 5 examples. We have used the same layout as in Fig. 4.

3.3 Robustness Towards Translated Images

In this section, we study the quality of the causes inferred by the trained model when translated versions of the original images in the CIFAR-10 data set are presented to the network. This problem is important because the network was trained on only 1000 images of horses and ships without any data augmentation. Convolutional Neural Networks rely on data augmentation to train models that are invariant towards various transformations like translations, rotations, etc. [11]. Here, we study the effect of a specific transformation i.e. translation on the robustness of the causes inferred by the trained network. Note that, again, we do not adapt the filters of the trained network.

The translated versions of the original images are obtained by shifted the content in the images to right and down by 4 pixels. The boundary pixels on the left and top of the original images are used in place of the pixels introduced as a result of shifting the image. For this study, we used images of horses and ships that are used for training as well as images of other objects that are never used in training. These translated images are then presented to the trained network and the inferred causes are used to reconstruct the translated versions of the original images as described in Sect. 3.1.

Figure 6 shows some examples of images reconstructed by the network using the inferred causes for the translated images. It can be observed that, even after presenting translated versions of the image, the information in the input images is well represented in the inferred causes. This may be attributed towards the retinotopic arrangement of receptive fields in the network but further analysis is needed to identify the reason behind this behavior of the network.

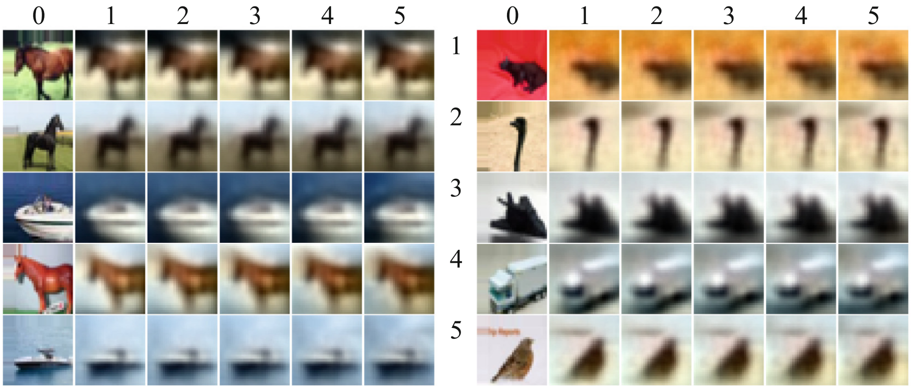

Fig. 6. Translated images reconstructed by the network using the inferred causes. As before, the images are arranged in 2 panels, each containing five rows. Note that the left panel contains translated versions of the images in the left panel of Fig. 4 and the right panel contains translated versions of the images in the left panel of Fig. 5.

4 Discussion

In this section, we discuss the computational implications of the algorithm presented in this paper and the similarities it has with existing approaches in machine learning.

Error-backpropagation is an important algorithm for training deep neural networks. It requires systematic propagation of information through the network in forward direction and during learning, backward propagation of error gradients. This makes it difficult to update all the network parameters in parallel. In this respect, predictive coding can be easily parallelized. It may be seen from Eqs. 6 and 7 that causes and filters can be adapted for all positions in a given layer parallelly due to the retinotopic arrangement of receptive fields. Furthermore, it is also possible to adapt causes and filters across all layers parallelly due to formulation of the error function (Eq. 5).

Another interesting aspect of predictive coding is its proximity to Deconvolutional Neural Networks (DNNs) [12]. DNNs are used to infer hierarchical neuronal activities for a given image. This problem is inherently ill-posed as there is no unique solution. To handle this issue DNNs optimize auxiliary variables and the neuronal activities alternately. A continuation parameter (β) is continuously increased during learning until the inferred neuronal activities are clamped to the auxiliary variables. This requires carefully controlling the learning process and higher computational power due to an extra optimization step on auxiliary variables. Alternatively, in predictive coding the update term associated with ϵ_{td} constrains the algorithm to infer activities that can be easily predicted by successive layers in the network (Eq. 6). This allows predictive coding to infer neuronal activities without using an extra optimization step.

5 Conclusion

In this paper, we describe a method to train deep neural networks using predictive coding. The approach uses network in which neurons the feedforward pathways obey the retinotopic arrangement of receptive fields observed in the brain. More empirical research is needed to determine whether feedback pathways have a similar organization.

We trained the network on a set of real-world images and then used the trained network to infer hierarchical causes for a different set of images as well as their translated versions. Even though the network is trained on a small data set of 1000 images of horses and ships, it can infer representative causes for translated versions of original images and those of other objects like sparrows, dogs, cars, etc. This shows that the network captures statistical regularities that are characteristic of real-world images.

Acknowledgement. The research work reported in this paper is carried out under European Union Horizon 2020 Program under Grant Agreement 720270-Human Brain Project SGA1 to C. M.A. Pennartz.

References

1. Mumford, D.: On the computational architecture of the neocortex - II the role of cortico-cortical loops. Biol. Cybern. **66**(3), 241–251 (1992)
2. Pennartz, C.M.A.: The Brain's Representational Power: On Consciousness and the Integration of Modalities. MIT Press, Cambridge (2015)
3. Rao, R.P.N., Ballard, D.H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nat. Neurosci. **2**(1), 79–87 (1999)
4. Spratling, M.W.: Reconciling predictive coding and biased competition models of cortical function. Front. Comput. Neurosci. **2**, 4 (2008)
5. Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annu. Rev. Neurosci. **18**(1), 193–222 (1995)
6. Spratling, M.W.: Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. Neural Comput. **24**(1), 60–103 (2012)
7. Whittington, J.C.R., Bogacz, R.: An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. Neural Comput. **29**(5), 1229–1262 (2017)

8. Jehee, J.F.M., Ballard, D.H.: Predictive feedback can account for biphasic responses in the lateral geniculate nucleus. PLoS Comput. Biol. **5**(5), e1000373 (2009)
9. Ledig, C., et al.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, pp. 1–14. arXiv (2016)
10. Michael Mathieu, Y.L., Couprie, C.: Deep multi-scale video prediction beyond mean square error. arXiv (2015)
11. Kauderer-Abrams, E.: Quantifying Translation-Invariance in Convolutional Neural Networks. arXiv (2017)
12. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2528–2535 (2010)