Fusing Bottom-up and Top-down Pathways in Neural Networks for Visual Object Recognition

Yuhua Zheng, Yan Meng and Yaochu Jin

***Abstract*—In this paper, an artificial neural network model is built up with two pathways: bottom-up sensory-driven pathway and top-down expectation-driven pathway, which are fused to train the neural network for visual object recognition. During the supervised learning process, the bottom-up pathway generates hypotheses as network outputs. Then target label will be applied to update the bottom-up connections. On the other hand, the hypotheses generated by the bottom-up pathway will produce expectations on the sensory input through the top-down pathway. The expectations will be constrained by the real data from the sensory input which can be used to update the top-down connections accordingly. This two-pathway based neural network can also be applied to semi-supervised learning with both labeled and unlabeled data, where the network is able to generate hypotheses and corresponding expectations. Experiments on visual object recognition suggest that the proposed neural network model is promising to recover the object for the cases with missing data in sensory inputs.**

I. Introduction

Object pattern learning and recognition remains as a key challenge in computer vision and machine learning for decades. The objective of object pattern learning and recognition is to learn the patterns (i.e., invariance features) from various training data and then to recognize the learned patterns from new unseen data. One main challenge in visual object recognition is to correctly model invariance features (or so-called latent variables). Over the last decades, varieties of machine learning models, including Bayesian networks [1, 2] and Artificial Neural Networks (ANN) [3, 4] have been applied to tackling this problem. In particular, ANNs have been well studied as associate memory and classifier [5-7]. The most popular feed-forward ANN model, multi-layer perceptrons, uses supervised learning with error back-propagation to build up the projection from data space to latent space.

Although the ANN-based approaches have demonstrated their effectiveness in various object recognition applications, only a single bottom-up pathway is used in the construction of neural networks. Recently, more evidence found in cognitive brain research and neuroscience [8-10] suggests that the nervous systems responsible for object recognition is a distributed cortical structure containing both bottom-up and top-down pathways.

Little research work has been reported on constructing artificial neural networks that consist of both the top-down and bottom-up pathways, which can be largely attributed to the difficulty in fusing the bottom-up and top-down processes systematically in one network. For example, in bidirectional associative memories [11], both pathways have been taken into account. However, no interaction between these two pathways exists. Grossberg[12] started to explore this area in 1970's and proposed the ART (Adaptive Resonance Theory), which is a general framework for representing interactions between bottom-up and top-down pathways. However, problem-specific learning algorithms and fusion technique of two pathways have to be developed for ART. Recently, biased competition theory[13, 14] has been proposed to explain the top-down attention of spatial stimulus and different feature dimensions. But generally, how to fuse the data flows of these two pathways to interpret data is still an open question in the neural network area for object recognition applications.

In this paper, we aim to propose a novel neural network model by fusing the bottom-up stimulus and top-down expectations for object pattern recognition. A learning algorithm for the neural network model has also been suggested. We focus on investigating the impact of the top-down expectations on the modulations of neuron activities in the lower-layer, and the consequential updates of neurons in the higher-layer by modulations through the bottom-up propagation iteratively. Different from using the spatial attentions to distribute attention strengths to different regions in the scene, like most work that uses the top-down attention for object recognition, we apply this new neural network model to interpret the appearance in a selected region for object recognition from various sensory data. We believe that the best interpretation of an object should contain not only the input data but also the a priori knowledge that has been learned before, which is realized through the top-down expectations.

The proposed neural network model that fuses the bottom-up and top-down pathways (FBTP-NN) is described in Section II. The learning process of the FBTP-NN is discussed in Section III, including both supervised and semi-supervised learning. Section IV presents our preliminary experimental results on visual object learning and recognition. Conclusions and future work are given in

Yuhua Zheng and Yan Meng are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA. (phone: 201-216-5496; fax: 201-216-8246; e-mail: yzheng1@stevens.edu, yan.meng@ stevens.edu).

Yaochu Jin is with Department of Computing, University of Surrey, Guildford, Surrey GU2 7TW, UK (e-mail: yaochu.jin@surrey.ac.uk).

Section V.

II. Fusing Bottom-up and Top-down Pathways in neural networks

A. The System Framework

Although the detailed mechanisms of human cortex have not been fully understood in neuroscience and cognitive science, increasing evidence has revealed that the neural system associated with learning and object recognition is a distributed cortical structure containing both bottom-up and top-down pathways. When an object is presented, the sensory input may generate ambiguous hypotheses, which could get similar scores (neuron activities) in the conventional feed-forward neural networks. However, the top-down signals that contain a priori knowledge or memory of the related objects can help to modulate the bottom-up pathway so that the ambiguousness in the stimulus can be reduced and more confident hypothesis can be generated and then selected.

In supervised learning, the neural network is subject to minimizing a cost function that often minimizes the error between the predicted label and real one. For training the FBTP-NN proposed in this work, we treat both input data and output labels equally as the environmental constraints, meaning that the network tries to learn the environment by adapting its dynamics (including both input layer and output layer) to these constraints through the learning process. Mapping both input data and output labels naturally involves both pathways of the neural network. The bottom-up process has been well studied in most feed-forward networks, i.e. from input data to the output label. With the top-down connections, the FBTP-NN is able to generate expectations from hypothesis to input data. In other words, the network tries to learn both hypotheses and expectations at the same time. Furthermore, the bi-directional data flows are fused via modulations of neuron activities. In this way, both the information in the current input stimulus and the previously learned knowledge are presented in the network to improve the learning and recognition capability.

Base on the above ideas, the general framework for fusing bottom-up and top-down pathways in a neural network is shown in Fig.1. The network may have multiple layers but only contains one input layer and one output layer, which are the interface of the network to the environment (i.e., input data and output labels). A number of hidden layers can exist in between. The input layer receives the sensory input and generates a few hypotheses at the output layer through the bottom-up pathway layer by layer. The output layer then produces expectations on the sensory stimulus via the top-down pathway. The expectations will be fused with the sensory inputs to update neuron activities of the input layer. The input updated based on the expectations will then generate new outputs accordingly. Such iterations repeat until certain stop conditions are met. The stop conditions are usually described by a cost function constrained by labels on the output layer and sensory stimulus on the input layer. During the learning, fusion can happen at every layer so that neuron activities of each layer can contain information of both pathways.

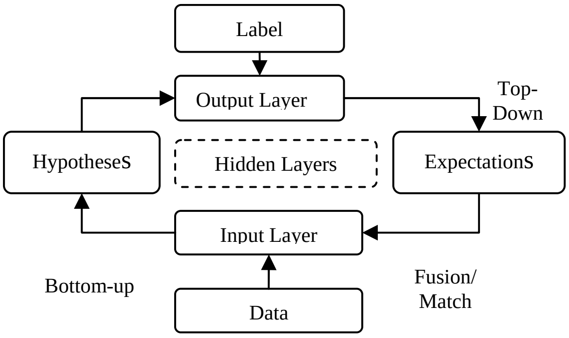

Fig. 1. The framework of the FBTP-NN model.

To train such a neural network with fused bottom-up and top-down pathways (FBTP-NN), it is essential to define a cost function that takes requirements of both pathways into account. Conventional learning algorithms for neural networks only consider minimizing the error between network outputs and the desired labels. In the FBTP-NN model with the top-down connections, the network attempts to achieve a match between the input layer of network and the sensory stimulus as well. To this end, a cost function that considers both the labeling error at the output layer and the discrepancy at the input layer has been developed. The weights in both pathways of the neural network are then updated iteratively by minimizing this cost function. Details about the cost function and the fusion technique will be discussed in following sections.

B. A Basic Two-layer FBTP-NN Model

The proposed FBTP-NN may contain multiple layers, since the neuron activities and weight updates of each neuron only depend on its adjacent layers. For the sake of simplicity, we will first discuss a basic two-layer FBTP-NN structure, as shown in Fig. 2. In this two-layer structure of a neural network, the lower layer X is called the data layer. The higher layer Y is called feature layer, which is considered as the features of the data layer.

Each layer contains a number of neurons. Neurons of different layers are fully connected by inter-layer weights, where W and P represent the inter-layer weights for bottom-up and top-down pathways, respectively. Note however that the assumption of a fully-connected structure is not plausible in the real visual cortical systems, where neurons of different layers are connected sparsely according to the receptive fields with various sizes. Since we focus on exploring the vertical data flows here, fully-connected inter-layer connections are assumed for simplicity. The lateral connections are only valid for feature neurons to reflect their dependencies. Therefore, between any two connected neurons, there are bottom-up (solid lines), top-down (dotted lines) and correlation (horizontal lines) connections. The network (inside the dotted square) is stimulated by the environment, which may contain both data information *D* and the feature information *L*.

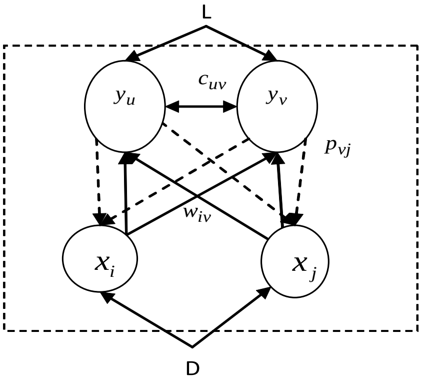

Fig. 2. A basic two-layer FBTP-NN model.

The cost function of this basic two-layer FBTP-NN structure is defined as:

$$E = E_p[(X,Y),(D,L)], \tag{1}$$

where E_p is the potential energy, whose cost depends on network interfaces *(X, Y)*, as well as environmental constraints *(D, L)*. The potential energy describes how the environment constrains the network. For example, given a number of sensory input and label pairs (also termed sample pairs) of the environment (D,L), which construct a potential surface, the network will try to adapt its interfacing layers (X,Y) to the surface by minimizing the distance between (X,Y) and (D,L). E_p may include both data and feature information in supervised learning, or data information only in semi-supervised learning. Under the influence of the environment, the neural network attempts to minimize its potential energy as:

$$\min(E_p[(X,Y),(D,L)]) = \min(dis(X,D) + dis(Y,L)) \tag{2}$$

The distance can be defined as the Manhattan distance as $\|D - X\|$ or p-norm as $\|D - X\|_P$. Under some circumstances, it is possible that the cost function contains only *D* or *L*. For example without minimizing the feature error, the field energy E_p will only have one term *dis(X, D)*.

C. Neuron Dynamics

In the above network model, given any environmental constraints *(D, L)*, the neuron activities will be updated iteratively through both bottom-up and top-down pathways. During the iterations, the data layer will be affected by both sensory stimulus and expectations from the feature layer. The neuron dynamics in the feature layer depends on the given feature information, data information and the correlations between the feature neurons.

Thus, the dynamics of a neuron *x* on the data layer is defined as:

$$\Delta x_i(t+1) = -\alpha_1 \cdot x_i(t) + \beta_1 \cdot g(\sum_{u=1}^{M} p_{ui}(t) \cdot y_u(t))$$
$$x_i(t+1) = x_i(t) + \Delta x_i(t+1), \tag{3}$$

where $\Delta x_i(t+1)$ is the change of activity of *i*-th input neuron at time *t+1*, which consists of two terms: self-decay and the top-down expectations from all feature neurons. $y_u(t)$ is the activity of u-th feature neuron at time *t* and *M* is the number of feature neurons. $p_{ui}(t)$ is the top-down weight at time *t*, and $\sum_{u=1}^{M} p_{ui}(t) \cdot y_u(t)$ represents the sum of top-down expectations from all feature neurons. The expectation is then fed into the activation function *g*, which is a sigmoid function defined as $g(x) = \frac{1}{1+e^{-x}}$ to represent the activation characteristic of neurons. α_1 and β_1 are decay constant and stimulus coefficient, respectively. Therefore, the current neuron activity depends on its previous activity and the change.

The dynamics of a feature neuron *y* is defined as:

$$\Delta y_u(t+1) = -\alpha_2 y_u(t) + \beta_2 \cdot g(\sum_{i=1}^{N} w_{iu}(t) x_i(t) + \sum_{v=1}^{M} c_{uv}(t) y_v(t))$$
$$y_u(t+1) = y_u(t) + \Delta y_u(t+1), \tag{4}$$

where the change $\Delta y_u(t+1)$ includes three terms. The first term is self-decay. The second one is the bottom-up stimuli from the data layer $\sum_{i=1}^{N} w_{iu}(t) x_i(t)$, where $w_{iu}(t)$ are weights in the bottom-up pathway and *N* is the number of neurons in the data layer. The third term is the correlation between the feature neurons, which can be either inhibition or excitation $\sum_{v=1}^{M} c_{uv}(t) y_v(t)$, where $c_{uv}(t)$ is the correlation strength and *M* is is the number of feature neurons. α_2 and β_2 are decay constant and stimulus coefficient of feature neurons, respectively.

A multi-layer neural network can be constructed by assembling a number of basic two-layer structures. More specifically, the first basic structure consists of the input layer (as the data layer) and the first hidden layer (as the feature layer). Then, the second basic structure includes the first hidden layer (as the data layer) and the second hidden layer (as the feature layer). This procedure continues all the way up to the output layer, which is the feature layer of the last basic structure. For a multi-layer neural network, the data information of the input layer is the sensory input, and the feature information of the output layer is the corresponding labels. Any hidden layer can be either treated as the data layer or the feature layer depending on which basic structure it is referring to at the current moment.

III. The Leaning algorithms

A. Cost Function

The proposed learning algorithm for the FBTP-NN is based on a new cost function that makes it possible to update the network parameters using information from both the bottom-up and top-down pathways. Given a number of data-label constraints (D, L), the network learns to adjust its parameters, mainly the connection weights (W, P, C), to minimize the cost function E defined in Equation (2). For the basic model defined in Fig.2, (X, Y) are neurons of data and feature layers and (D, L) are the data and feature information from the environment. By taking (X, Y) and (D, L) as vectors, and applying square error to measure their distances, the cost function can be rewritten as:

$$E = \sum_{i=1}^{N}(d_i - x_i)^2 + \sum_{u=1}^{M}(l_u - y_u)^2 \qquad (5)$$

For clarity, the time index is omitted in the following equations. The derivative of the cost function with respect to the bottom-up weight w_{iu} can be obtained as follows:

$$\frac{dE}{dw_{iu}} = -2(d_i - x_i)\frac{dx_i}{dw_{iu}} - 2(l_u - y_u)\frac{dy_u}{dw_{iu}} \qquad (6)$$

Substituting Eqns. (3) and (4) into Eqn. (6), we have:

$$\frac{dE}{dw_{iu}} = 0 - 2\beta_2 \cdot g' \cdot x_i \cdot (l_u - y_u)\ , \qquad (7)$$

where β_2 is the stimulus coefficient of feature neuron. g' is the derivative of the activation function. For the sigmoid function $g(x) = \frac{1}{1+e^{-x}}$, $g'(x) = g(x)(1-g(x))$ is a constant for a given input. x_i is the activity of the related data neuron. l_u and y_u are the desired feature and real activity of the output neurons, respectively. Therefore, to minimize the cost function *E*, the change of weight Δw_{iu} should be:

$$\Delta w_{iu} = r_1 \cdot x_i \cdot (l_u - y_u)\ , \qquad (8)$$

where r_1 is the learning rate of the bottom-up weights. Eqn. (8) is a Hebbian-like error-driven learning method.

Similarly, we can get the derivative of cost function with respect to the top-down weights *P* and the update rule for a specific p_{ui} can be derived as:

$$\frac{dE}{dp_{ui}} = 0 - 2\beta_1 \cdot g' \cdot y_u \cdot (d_i - x_i)\ ,$$

$$\Delta p_{ui} = r_2 \cdot y_u \cdot (d_i - x_i)\ , \qquad (9)$$

where r_2 is the learning rate of the top-down weights.

On the other hand, the correlation connections *C* reflect the dependencies among the feature neurons and can be updated as:

$$\Delta c_{uv} = r_3 \cdot \text{sgn}(y_u) \cdot \text{sgn}(y_v)\ , \qquad (10)$$

where r_3 is the learning rate of the correlation weights. $\text{sgn}(y)$ is the sign function defined as:

$$\text{sgn}(y) = \begin{matrix} 1\ if\ y > 0 \\ -1\ otherwise \end{matrix}\ .$$

Therefore, if the two neurons are activated simultaneously, their correlation weight will be strengthened; weaken otherwise. For tasks such as object recognition or classification, we usually assume that feature neurons in the output layer of a multi-layer network are mutually suppressive, which means that the output neurons are competitive.

B. The Supervised Learning Process

If we assume that all input data are labeled, then a supervised learning algorithm can be applied. For a multi-layer network, each layer tries to converge to its desired value under the environmental constraints. The desired value of the neurons on the input layer is the input data, the desired value for the output layer is the true label, and the desired value for the hidden layers is the fusion of the bottom-up stimulus and top-down expectation which is defined as:

$$v_i^l = f \cdot y_i^l + (1-f) \cdot x_i^l \qquad (11)$$

where v_i^l represents the stable state of *i*-th neuron of *l*-th layer. For the same neuron, x_i^l and y_i^l are often different, since y_i^l is produced by the bottom-up stimulus, while x_i^l is the top-down expectation. v_i^l influences the neuron activity and updates the related weights. *f* is the fusion ratio in combining the bottom-up stimulus and top-down expectation.

By defining the desired values for all layers, the supervised learning can be conducted, through a number of bottom-up, top-down and fusion iterations. Driven by the input data, the network generates hypotheses layer by layer through the bottom-up process. Then expectations are built up based on these hypotheses along the top-down pathway. For hidden layers, the stimulus and expectation are fused to generate the desired values. The supervised learning procedure of the FBTP-NN can be summarized as followings.

Generally, a FBTP-NN is a multi-layer neuron network consisting of a number of basic two-layer structures described in Section II.B. Given a FBTP-NN with randomly initialized weights (W, P, C) and a number of data-label pairs (D, L), the bottom-up process is started from the input layer *X* with input data *D*. Here, the input layer of the network is the data layer and the first hidden layer is the feature layer in a basic two-layer structure.

- Step 1. Calculate the activity of the neurons on the feature layer *Y* via Eqn. (4).
- Step 2. Update the bottom-up weights *W* via Eqn. (8). *L* is the feature information for the current feature layer, which can be defined in two cases. If the current feature layer is the output layer of the neural network, *L* is the true label. If the current feature layer is a hidden layer, *L* is the desired value *V* defined in Eqn.

(11).

- Step 3. Update the correlation weights C via Eqn. (10).
- Step 4. Move up one layer to build a new basic structure. The current feature layer becomes the data layer and the adjacent top layer becomes the feature layer in the new basic structure. Repeat steps 1-3 for the learning of the new basic structure.
- Step 5. Repeat steps 1-4 until the output layer of the whole neural network.

Now perform the top-down and fusion process from the output layer. Here, we start with a basic structure using the output layer as the feature layer and the last hidden layer as the data layer.

- Step 6. Calculate the neuron activities in data layer X via Eqn. (3).
- Step7. Update the top-down weights P using Eqn. (9). Here, the data information D can be defined in two cases. If the current data layer is the input layer, D is the true sensory data. If the current data layer is a hidden layer, D is the desired value V defined in Eqn. (11).
- Step 8. Fuse the bottom-up stimulus and top-down expectation to get the desired values for the current data layer using Eqn. (11).
- Step 9. Move down one layer to build a new basic structure. The current data layer becomes the feature layer and the adjacent bottom layer becomes the data layer in the new basic structure. Repeat steps 6-8 for the learning of this new basic structure.
- Step 10. Repeat steps 6-9 until the input layer of the whole neural network.
- Step 11. Repeat steps 1 to 10 until the stop condition is met.

By repeating the above steps, the network will learn the labels in the output layer as well as the corresponding stimulus in the input layer by updating the weights in the bottom-up and top-down pathways and the correlation weights among the neurons in the same layer (except for the input layer).

C. Extension to Semi-supervised Learning

For many learning tasks, labeling data is extremely time consuming and needs expert skills. It is found that the learning performance can be improved significantly for a neural network to learn from a mixture of small amount of labeled data and large amount of unlabeled data. However ambiguous data may trigger confusing hypotheses that have similar quality values. To make use of the top-down expectations as well as bottom-up stimulus to evaluate hypotheses, the neural network is trained by a small number of labeled data initially. When unlabeled data come in, the network performs the bottom-up, top-down and fusion process to generate potential hypotheses. If more than one hypothesis is produced, they will be evaluated by combining the bottom-up and top-down scores. The winner will be adopted to train the neural network.

It is assumed that only one output is desired for object recognition. The network generates K hypotheses as $\{(\overline{X}_k, \overline{Y}_k),\ k = 1,2,...,K\}$. Each hypothesis has one output vector $\overline{Y}_k$ with only one output neuron activated, whose value usually is between 0 and 1. Different hypotheses have different output neurons activated. Each hypothesis also has one expectation on the input layer as $\overline{X}_k$. Therefore the hypothesis score is defined as:

$$h_k = (1 - \overline{Y}_k) + \phi \cdot Dis(D, \overline{X}_k), \qquad (12)$$

where the first part is the score from the bottom-up pathway and the second part is the score from the top-down pathway. The larger $\overline{Y}_k$ is, the better the hypothesis is. The smaller the distance between the input data D and expectation $\overline{X}_k$ is, the better the hypothesis is. ϕ is a hyper parameter for combining scores.

This semi-supervised learning method can be described as followings.

Again, the FBTP-NN is a multi-layer neural network. Given a FBTP-NN with the learned weights (W, P, C) from some labeled data initially and a number of unlabeled data D.

Stage 1: Free runs to generate hypotheses

Steps 1, 4, and 5 in Section III.B are applied to the bottom-up process and Steps 6, 8-11 are applied to the top-down process to generate hypotheses. Steps 2, 3, and 7 are skipped in the free runs to keep the weights unchanged.

Stage 2: Hypotheses evaluation

Evaluate hypotheses one by one.

- Step 1: With only output vector $\overline{Y}_k$ activated, calculate the expectation $\overline{X}_k$ on the input layer.
- Step 2: Compute the hypothesis score via Eqn. (12).
- Step 3: Repeat Steps 1-2 for all hypotheses and choose the winner with the minimal score.

Stage 3: Learning the winner hypothesis $(D, \overline{Y}_k)$

Choose the winner hypothesis as the label for the input data, and apply the supervised learning algorithm to train the neural network.

IV. The Experimental Results

A. Experimental Setup

To demonstrate the effectiveness of the proposed learning algorithms, a few experiments on visual object recognition have been performed using the FBTP-NN. The training datasets are taken from Caltech 256 [15], as shown in Fig. 3. Firstly, the original images are transformed into gray images, where objects are presented as white pixels and the background as black pixels. Due to the cluttered background, some images are very noisy and it is hard to tell the object from the background. We choose images of four object categories, air planes, bicycles, shotguns and motorcycles.

For each category, objects with different appearances, sizes, orientations, backgrounds and lightening conditions are presented. The pixel values will be fed into the neural network directly as input, although extracted salient features as inputs may help to improve the recognition performance.

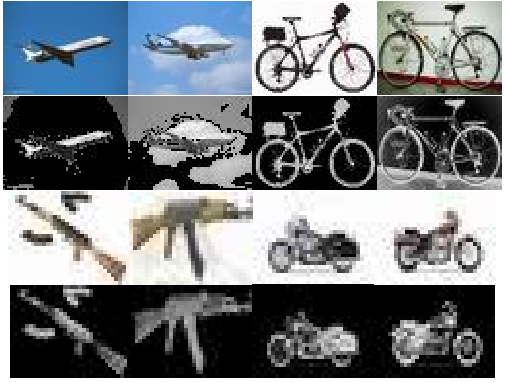

Fig. 3. Experimental data sets taken from Caltech 256.

In this work, a three-layer FBTP-NN is built for object recognition, which contains two basic two-layer structures. First basic structure consists of the input layer and the hidden layer. The other consists of the hidden layer and the output layer. The number of neuron in the input layer equals the size of training images, i.e. 32x24. The hidden layer has about 100 neurons and the output layer has 4 neurons. Neurons of adjacent layers are fully connected. Correlation connections are applied in the hidden and output layers only if they act as the feature layer in the basic structures.

TABLE I. PARAMETER SETUP

Coefficient	Value
Self-decay rate α_1, α_2	0.2
Stimulus rate β_1, β_2	0.5
Learning rate r_1, r_2, r_3	0.01
Fusion rate f	0.95
Score combining rate ϕ	0.5
Max loop for super training	100
Max loop for free-run	15
Max loop for semi training	30

Table I provides the parameter setup used in the experiments. The decay constants for data neurons and feature neurons are supposed to be the same for simplicity. The same assumption is made to the stimulus rates. The learning rates are set also to the same. Usually, a smaller learning rate requires more learning iterations to achieve the same performance. A fusion rate of 0.95 suggests that a neuron is less influenced by the top-down pathway each time so as not to change too fast. A score-combining rate of 0.5 means that the bottom-up and the top-down scores are treated equally. For supervised learning, one data is learned for 100 iterations. For semi-supervised learning, the free-run is executed for 15 iterations and the winner hypothesis is learned for 30 iterations.

The FBTP-NN can be trained by using either the online learning mode or the batch learning mode. For the online learning mode, the training data are mixed randomly and presented to the network sequentially to reduce the forgetting influence.

B. *Analysis of Learning Procedure*

To demonstrate the learning performance of the proposed FBTP-NN, the dynamics of the input layer and output layer are presented and discussed here. Fig.4 shows how the input neuron activities change over the learning procedure for a labeled data of airplane #1. Every image of Fig.4 is recovered from the input layer of the neural network. In Fig.4, the top left panel shows the original image, the top right is the corresponding gray image, which is also the initial neuron activity level of the input layer. At the beginning, there is no top-down expectation fused into the neuron activity. As learning goes on, the neuron activity of the input layer is updated by fusing the top-down expectations. Gradually, the network is able to learn better expectations about the input data, which means that it can produce a better image of the given object, which are shown in the bottom-left and bottom-right panels in Fig.4, as of iteration 70 and 95, respectively.

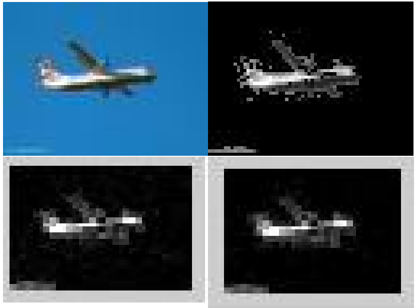

Fig. 4. The recovered images of the input layer changing over learning procedure for airplane #1.

We define the label error as the Hamming distance between the true label and activation level of the output layer neurons, as $\|L - Y(t)\|$. $Y(t)$ represents neuron activities of the output layer at time t. The smaller the distance is, the better the output. Fig. 5(a) shows how the label error for dataset airplane #1 changes over the learning iterations. In the first 20 iterations, the label error decreases. As learning continues, the input layer is modulated by the top-down expectations, which may cause a disturbance to the bottom-up stimulus, as shown in Fig. 5(a), roughly from iterations 40 to 60. After a number of bottom-up and top-down interactions, the network is able to reduce the label error.

Then we define the expectation error as the Hamming distance between the original input data and neuron activities of the fused input layer, as $\|D - X(t)\|$. Fig. 5(b) shows the error changes over the learning iterations. In the beginning, the influence from the top-down pathway is too little to activate neurons so that the distance starts from zero. As the learning process goes on, stronger perturbations are added from the top-down expectation to the input layer. When top-down weights are further trained, the distance decreases, meaning the network is able to recover the true data better.

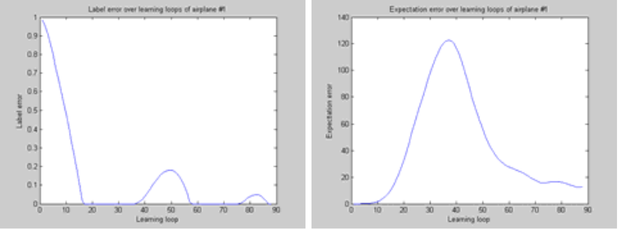

Fig. 5. (a) Change of the label error over learning iterations for dataset airplane #1; (b) Change of the expectation error over learning iterations for dataset airplane #1.

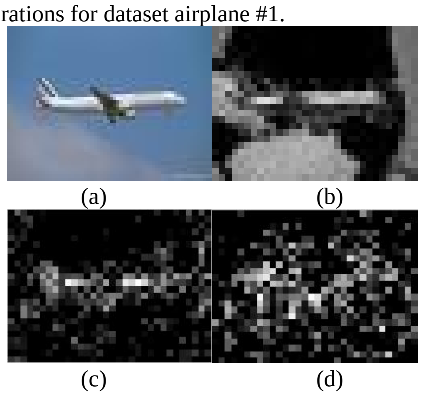

Fig. 6. The hypotheses evaluation.

Fig.6 shows how the combined hypotheses score improves the semi-supervised learning. Fig. 6(a) is the original image, whose gray image is noisy due to cluttered background, as shown in Fig. 6(b). In the experiment, as the first part of Eqn. (12), the bottom-up score of airplane is 0. 42 and is 0.4 for bicycle. So if the network output is based only on the bottom-up pathway, this image will be recognized as a bicycle. Fig.6(c) shows the expectation image of the network by selecting an airplane as the hypothesis, which captures the most pixels along the plane body in (b). The expectation image when a bicycle is selected as the hypotheses is shown in Fig. 6(d). We can see that the recovered image contains some wheel-like structures, which are similar to the clouds in Fig. 6(b) and cause the incorrect output of the bottom-up pathway. When the second term in Eqn. (12) also plays a role in the neuron dynamics, the top-down score will balance the hypotheses of bottom-up pathway. After combination of the information of both pathways, the airplane hypothesis has a score of 0.32 and the bicycle hypothesis has a score of 0.35, which leads to the correct recognition of the object as an airplane.

C. Robustness Test

One important advantage of having the top-down pathway is its ability of recovering missing data in the sensory input. Fig. 7(a) presents the original image whose right part is lost. Fig. 7(b) is the initial neuron activities of the input layer at the beginning of learning. During the learning, the top-down expectation is able to recover the lost information in the stimulus, which is fused into the input layer, as shown in Fig. 7(c). Fig. 7(d) shows the neuron activities of the input layer at the end of learning, where the lost information has been recovered.

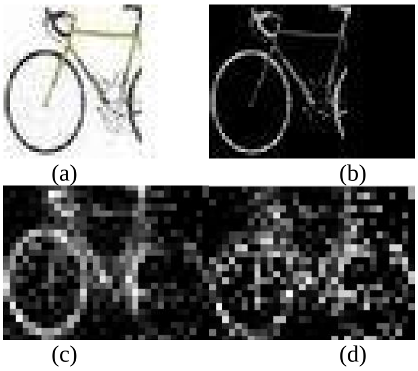

Fig. 7. The recovery ability of FBTP-NN.

D. Comparative Study on Learning and Recognition

Firstly, we evaluate the recognition ability of the proposed FBTP-NN on the airplane and bicycle datasets, as a two-category case. The FBTP-NN has been trained using training datasets of different sizes, such as 10, 15, and 20. Firstly, the batch learning is adopted for training the neural network. The learning iteration number is set to 1000. As shown in Table II, the FBTP-NN achieves better performance up to 98% with more training samples. Then the same training datasets are used to train a three-layer feed forward neural network (FF-NN). The FF-NN has 100 hidden neurons and its weights are updated by error back-propagation. The number of learning iterations for the FF-NN is also 1000. From Table II, we can tell that FBTP-NN has comparable performance with FF-NN. This is reasonable since FBTP-NN also adopts the gradient-decent learning strategy as described in Eqn. (7) and (9), which is similar to back-propagation in principle. Therefore under the same training and testing conditions, FBTP-NN and FF-NN have similar performances.

TABLE II. RECOGNITION RATE OF BATCH LEARNING OVER DIFFERENT TRAINING DATA SIZES FOR TWO-CATEGORY EXPERIMENTS

Training Data	10 Data		15 Data		20 Data	
Network	FBTP	FF	FBTP	FF	FBTP	FF
Airplane	78%	90%	95%	95%	98%	98%
Bicycle	90%	90%	95%	95%	98%	95%

In addition to batch learning, we also compare the performance of FBTP-NN and FF-NN using the on online learning mode, where the order in which the training data are presented and the number of iterations become important in preventing the neural network from over-fitting any individual data. The iteration number of 35 is chosen through trial-and-errors and 40 test samples are applied then. Similar to batch learning, FBTP-NN has comparable performance with FF-NN, as shown in Table III. On the other hand, in the online learning mode, the performance of learning is affected by the order of the training samples and the number of learning iterations, which may be a reason why in some cases the networks can even achieve better recognition with fewer training data.

TABLE III. RECOGNITION RATE OF ONLINE LEARNING OVER DIFFERENT TRAINING SIZES FOR TWO-CATEGORY EXPERIMENTS

Training Data	10 Data		15 Data		20 Data	
Network	FBTP	FF	FBTP	FF	FBTP	FF
Airplane	75%	77%	93%	95%	90%	90%
Bicycle	95%	80%	90%	90%	95%	90%

Due to the top-down pathway, FBTP-NN is able to represent the learned knowledge about the sensory input, which is expected to help improving recognition when the new sensory input is incomplete. Both FBTP-NN and FF-NN are trained by same training samples under the batch learning mode. Then 40 testing samples with 25% and 50% data missing are made by replacing the image pixels of 25% or 50% of the images with all zeros. These samples with incomplete information are applied as the testing data to evaluate both models, respectively. As shown in Table IV, FBTP-NN outperforms FF-NN in both experiments. This can be attributed to the top-down pathway of FBTP-NN that can recall the learned feature of sensory input given a hypothesis in the output. The recalled sensory input can help to recover the missing data in the presented stimuli as shown in Fig. 7, thereby improving the recognition performance. On the contrary, FF-NN can only make judgments based on partial data, leading to a lower recognition rate.

TABLE IV. RECOGNITION RATE COMPARISONS ON DATA WITH PARTIAL INFORMATION MISSING FOR TWO-CATEGORY EXPERIMENTS

Training Data	25% Missing		50% Missing	
Network	FBTP	FF	FBTP	FF
Airplane	92%	87%	88%	86%
Bicycle	90%	90%	90%	83%

Table V shows the performance of FBTP-NN on four-category recognition using the online learning mode. Each data is trained with 35 learning iterations and 40 testing samples are applied. In this case, the average probability of each category is 25%. FBTP-NN can achieve very good recognition rate up to about 80%, which reflects the learning capacity of the proposed neural network. The proposed FBTP-NN can be applied for multi-category object classification using one integrated structure. On the other hand, many algorithms in computer vision field adopt One-VS-All strategy and have to construct individual classifier for each individual class separately[16].

TABLE V. RECOGNITION RATE OF ONLINE FBTP-NN OVER DIFFERENT TRAINING SIZES FOR FOUR-CATEGORY EXPERIMENTS

Training Data	10 Data	15 Data	20 Data
Airplane	62%	78%	79%
Bicycle	75%	80%	80%
Motor	80%	85%	85%
Shotgun	78%	78%	80%

V. THE CONCLUSION AND FUTURE WORK

In this paper, a novel neural network model with both bottom-up and top-down pathways is proposed. The network learns the maps from data to label as well as the expectation on sensory input given a hypothesis on the label. Neuron dynamics of the network is regulated by both the bottom-up stimuli and top-down expectations. A new cost function has also been suggested to train the weights of the network. Extensive experimental results have demonstrated that the proposed model is efficient in object recognition problems.

For future work, we intend to analyze the fusion dynamics of the proposed neural network, especially on exploring new fusion mechanisms. Currently, the bottom-up stimulus and top–down expectations are fused linearly, which can be extended to more advanced and bio-inspired nonlinear mechanisms. Additional empirical study is also needed to verify the performance of the proposed model in comparison to the state-of-the-art methods on more benchmark problems.

REFERENCES

[1] B. Ommer and J. M. Buhmann, "Learning the Compositional Nature of Visual Objects," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1-8.

[2] R. P. N. Rao, "An optimal estimation approach to visual perception and learning," *Vision Research,* vol. 39, pp. 1963-1989, 1999.

[3] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, pp. 994-1000 vol. 2.

[4] S. Kirstein, H. Wersing, and E. Körner, "A biologically motivated visual memory architecture for online learning of objects," *Neural Networks,* vol. 21, pp. 65-77, 2008.

[5] J. J. Hopfield, "Neural networks and phsical systems with emergent collective computational abilities," *Biophysics,* vol. 79, pp. 2554-2558, 1982.

[6] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks,* vol. 1, pp. 17-61, 1988.

[7] K. Fukushima, "Neural network model restoring partly occluded patterns," *International Journal of Knowledge-Based and Intelligent Engineering Systems,* vol. 8, pp. 59-67, 2004.

[8] A. K. Engel, P. Fries, and W. Singer, "Dynamic predictions: Oscillations and synchrony in top-down processing," *Nat Rev Neurosci,* vol. 2, pp. 704-716, 2001.

[9] S. Treue, "Visual attention: the where, what, how and why of saliency," *Current Opinion in Neurobiology,* pp. 428-432, 2003.

[10] N. Kanwisher and E. Wojciulik, "Visual attention: Insights from brain imaging," *Nat Rev Neurosci,* vol. 1, pp. 91-100, 2000.

[11] B. Kosko, "Adaptive bidirectional associative memories," *Appl. Opt.,* vol. 26, pp. 4947-4960, 1987.

[12] S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," in *Connectionist models and their implications: readings from cognitive science,* ed: Ablex Publishing Corp., 1988, pp. 243-283.

[13] J. Duncan, G. Humphreys, and R. Ward, "Competitive brain activity in visual attention," *Current Opinion in Neurobiology,* vol. 7, pp. 255-261, 1997.

[14] D. M. Beck and S. Kastner, "Top-down and bottom-up mechanisms in biasing competition in the human brain," *Vision Research,* vol. 49, pp. 1154-1165, 2009.

[15] G. H. Griffin, AD. Perona, P, "The Caltech-256," Caltech Technical Report.

[16] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 29, pp. 411-426, 2007.