Perception Evolution Network: Adapting to the Emergence of New Sensory Receptor*

Youlu Xing, Furao Shen, Jinxi Zhao
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210046, China
youluxing@sina.com, frshen@nju.edu.cn, jxzhao@nju.edu.cn

Abstract

The proposed *Perception Evolution Network* (PEN) is a biologically inspired neural network model for unsupervised learning and online incremental learning. It is able to automatically learn suitable prototypes from learning data in an online incremental way, and it does not require the predefined prototype number and similarity threshold. Meanwhile, being more advanced than the existing unsupervised neural network model, PEN permits the emergence of a new dimension of perception in the perception field of the network. When a new dimension of perception is introduced, PEN is able to integrate the new dimensional sensory inputs with the learned prototypes, i.e., the prototypes are mapped to a high-dimensional space, which consists of both the original dimension and the new dimension of the sensory inputs. We call it a *Cognition Deepening Process*. Artificial data and real-world data are used to test the proposed PEN, and the results show that PEN can work effectively.

1 Introduction

Biologically inspired computing has spawned many classical and powerful algorithms including Perceptron [Rosenblatt, 1958], Self-Organizing Map [Kohonen, 1982].

In 2007, using genetic engineering, Jacobs et al. [Jacobs *et al.*, 2007] inserted human L-pigment genes into female mice one X-chromosome and found that the heterozygous female mice show enhanced long-wavelength sensitivity and acquire a new capacity for chromatic discrimination. It means that the knock-in mice express the human gene in their cone cells and that the human L pigment transmits light signals with an efficiency comparable to that of the mouse's native pigments. As a consequence, the knock-in mice are able to discriminate among green, yellow, orange, and red panels that, to ordinary mice, look exactly the same. From this work, we can see that the brain of the organisms has an amazing adaptability and plasticity; it can use the new introduced sensory receptor immediately! Such adaptability and plasticity of the brain will make the knock-in mice "understand" the world much deeper than other mice.

The above experiment inspires us a very interesting and challenging problem: *can we exploit a computational model that is able to expand its cognitive dimension online freely?* If this is achieved, the agent with such model will be able to expand its sensing capability during its lifetime. This is different from the natural way that the offspring get a new cognitive dimension through genetic variation while their parents sacrifice. Here, the "parental generation" (i.e., the agent itself) is given a new cognitive dimension "online" *directly*. We think this way is more economical and suitable for the current machine type intelligence. And it will have broad applications such as in robot system, information fusion, and data stream mining. For example, if we install new sensors to an agent to expand its sensing capability, with this model, we do not need to retrain the agent offline from scratch, the information gathered from the new installed sensors is fused with the existed knowledge online automatically, in other words, the model is autonomous for the sensory dimension. However, to our knowledge, there is no such computational model. In the process of evolution from lower organisms to higher organisms, increasingly complex perceptual system is generated by the evolution of the sense organs. Therefore, more information arrives at the brain of the organisms. As a consequence, the organisms are able to understand the real world more deeply, as Figure 1 shows. We propose a *Perception Evolution Net-*

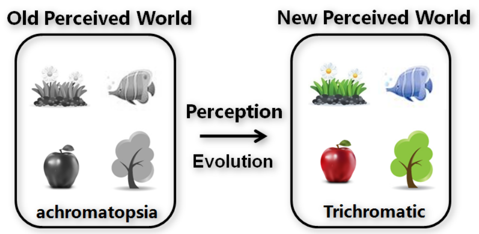

Figure 1: Perception evolution permits the organisms to understand the real world more deeply. The figure shows a world in the eyes of the achromatopsia organisms and another world in the eyes of the trichromatic organisms.

*This work is supported in part by the National Science Foundation of China under Grant Nos. (61373001, 61375064) and Jiangsu NSF grant (BK20131279). Corresponding author is Furao Shen.

work (PEN) to simulate the evolution of the perceptual system, i.e., PEN permits the emergence of a new dimensions of perception in the perception field of the network.

Figure 2 gives the mathematical modeling of the PEN. In the beginning, the agent has two types of sensory neurons to perceive the real world; correspondingly, the external world is mapped into the agent's internal world, which is a "two-dimensional" space, while each dimension is attached to its corresponding type of sensory neurons. After the agent gets a new type of sensory neurons, the agent has a new information channel to perceive the real world, the patterns (or "memories") stored in the agent should be integrated with the new dimensional sensory inputs, i.e., the patterns are mapped to a high-dimensional space which consists of both the original dimension and the new dimension of the sensory inputs. We call it a *Cognition Deepening Process*.

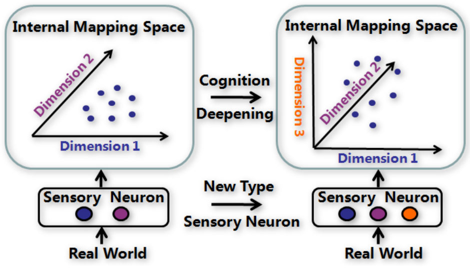

Figure 2: Mathematical modeling of the PEN. With the coming of the new type of sensory neuron, the organisms have a new information channel to perceive the real world. As a consequence, the internal mapping space has a corresponding new dimension. Thus, the points in the low-dimensional feature space will be mapped to a higher dimensional feature space. We call it a cognition deepening process.

Meanwhile, the data in the real world are usually unlabeled. Thus, PEN is designed as an unsupervised learning model to build the representations of the external data. On the other hand, the agent usually perceives the external world in an online way, and it is usually an open system, which is able to adapt to the changing or open-ended environment. In such environments, data with new knowledge is coming continuously. Therefore, keeping learning new knowledge quickly without catastrophic forgetting of already learned, and still successful, memories is a very important ability, just like humans are able to learn new "objects" or "words" without forgetting the previously learned ones throughout their lifetimes. Thus, Online Incremental Learning (OIL) is a distinct property of the brain. And OIL is also introduced to PEN.

As a summary, we give the neural network modeling of the PEN in Figure 3. The prototypes and connections between prototypes in the prototype field of the network are created online and incrementally, i.e., new prototypes (nodes with dashed edge in the prototype field in Figure 3) are created for some new patterns (or new knowledge) which are dissimilar with the existing prototypes (patterns) recorded in the network. Each prototype P_i is associated with a 3-tuple: $\{W_{P_i}, T_{P_i}, M_{P_i}\}$ which represent the weight vector, the similarity threshold and the accumulated activation times of P_i respectively. The perception field permits the emergence of new sensory neurons. After getting some new sensory neurons, PEN automatically integrates the new input signals (or information) received from these new sensory neurons with the existing knowledge (or prototypes) learned from the original sensory neurons. Thus, the perception field and the prototype field are open-ended. PEN is a kind of perception-prototype field open-ended neural network model.

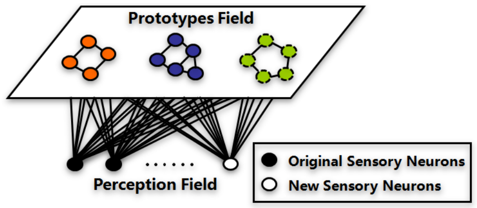

Figure 3: Neural network modeling of the PEN. The perception field permits the emergence of new sensory neurons. The green nodes with dashed edge in the prototype field are added for some new patterns during online learning. It is a kind of perception-prototype field open-ended neural network.

Briefly, the main targets of PEN are to:

(1) Permit the emergence of new sensory neurons in the perception field of the network, and integrate the new input signals with the prior learned knowledge.

(2) Learn suitable prototypes from the data in an unsupervised and incremental way, and not require predefined prototype number and similarity threshold.

2 Related Work

Many scholars study in unsupervised learning and online incremental learning. The Adaptive Resonance Theory network (ART) [Carpenter and Grossberg, 1988] and TopoART [Tscherepanow *et al.*, 2011] will create a new node when no match occurs between the current input sample and the current category set. The degree of matching is controlled by a vigilance parameter. However, the vigilance parameter should be predefined, it is a difficult job when we have little prior knowledge about the learning task, especially for the unsupervised learning task.

Growing Cell Structure (GCS) [Fritzke, 1994] and Growing Neural Gas (GNG) [Fritzke, 1995] insert new node(s) for every λ samples learned. However, during each λ period, the input sample is forced to merge with a node no matter how big the gap between them. Considering the physical meaning, it is unreasonable to merge two patterns with significant difference. Self-Organizing Incremental Neural Network (SOINN) [Shen and Hasegawa, 2006], Adjusted-SOINN (ASOINN) [Shen and Hasegawa, 2008] and LB-SOINN [Zhang *et al.*, 2014] decide whether to create a new node for the input sample according to the node distribution around the local region of the input sample.

We summarize these unsupervised incremental learning methods above as prototype field open-ended neural network (PFOENN). This type of network makes the prototype field is open-ended for new categories by adding new prototypes. In recent years, PFOENN are applied to various domains. However, the PFOENN is not able to expand the perception field which we think is a very important ability for unsupervised learning as mentioned in the introduction section. For example, if we install new sensors to a robot as new information channels, and we want the robot to use the new sensors directly and effectively. The PFOENN cannot deal with such task. The PEN is designed to solve such problem.

3 Perception Evolution Network

3.1 Problem Formulation

Assume that the original neural network $\mathcal{N}$ has n neurons in the perception field which receive n-dimensional external data $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n$. After a period of learning, some prototypes are created in the prototype field of PEN, then, new m sensory neurons emerge in the perception field of PEN, the received data becomes $\mathbf{x} = (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{n+m}) \in \mathbb{R}^{n+m}$. The learned prototypes will be mapped to a high-dimensional space which contains the dimension of these m new sensory neurons. If the new sense brings some new distinguishable categories, PEN will create prototypes for such new categories.

The work flow of PEN is as follows: the prototype field is empty in the beginning, learning samples are fed into the network sequentially. PEN will create two prototypes using the first two input samples. For the later input sample, PEN first conduct prototype competition, then prototype learning and self-organizing are conducted according to the result of the competition step, meanwhile, the similarity threshold of the activated prototypes will be updated. When all the steps above are done, PEN will process the next input sample. Prototype pruning is conducted after every λ samples learned.

When some new sensory neurons are introduced, PEN will find some low-dimensional prototype to map to high-dimensional space. Self-organizing and similarity threshold updating are conducted similarly as the procedure before new sensory neurons appear. We remove the prototypes that cannot be mapped to the high space after a long period of learning because they are potential distortion prototypes. Postprocessing is executed when all prototypes are mapped to the high-dimensional space.

Below, we denote the original low-dimensional space as $S_l = \mathbb{R}^n$, the new high-dimensional space as $S_h = \mathbb{R}^{n+m}$, the learned prototypes by PEN are stored in set P.

3.2 Prototype Competition

When an input sample $\mathbf{x} = (x_1, x_2, ..., x_n, ..., x_{n+m}) \in S_h$ comes, we first calculate the Euclidean distance between $\mathbf{x}$ and all prototypes in set P. The dimension of each P_i may be different with $\mathbf{x}$ (because during learning, some P_i may be already mapped to space S_h and some may still stay in space S_l). For the prototype $P_i \in S_l$, we calculate the Euclidean distance between $\mathbf{x}$ and P_i using the first n-dimensional attributes, i.e., the attributes of space S_l. For the prototype $P_i \in S_h$, we use all (n+m)-dimensional attributes to calculate the Euclidean distance. Then we find the winner prototype P_w^l in space S_l as:

$$P_w^l = \underset{P_i \in P}{\operatorname{argmin}} \|\mathbf{x} - W_{P_i}\|_{S_l} \tag{1}$$

where W_{P_i} is the weight vector of P_i and $\|\cdot\|_{S_l}$ represents the Euclidean distance in space S_l. For the prototypes belonging to S_h, we use their first n-dimensional weights (i.e., the part of their weight vector in space S_l) to compete with the prototypes belonging to S_l. Therefore, P_w^l may belong to space S_l or S_h. Then we find the winner and runner-up prototypes P_w^h, P_r^h in space S_h as:

$$P_w^h = \underset{P_i \in P^h}{\operatorname{argmin}} \|\mathbf{x} - W_{P_i}\|_{S_h} \tag{2}$$

$$P_r^h = \underset{P_i \in P^h \setminus P_w^h}{\operatorname{argmin}} \|\mathbf{x} - W_{P_i}\|_{S_h} \tag{3}$$

where P^h represents the set of prototypes belonging to space S_h, and $\|\cdot\|_{S_h}$ represents the Euclidean distance in space S_h. The prototypes that belonging to S_l cannot be applied to $\|\cdot\|_{S_h}$. Therefore, the prototypes which belong to S_l do not take part in this competition. For convenience in notation, we rewrite P_w^l as P_c, P_w^h as P_a, and P_r^h as P_b.

Note: If all prototypes in P have been mapped to space S_h, we do not need to conduct the competition in space S_l.

3.3 Prototype Learning

The winner prototype P_c is the candidate prototype to be mapped to the high-dimensional space S_h. If

$$\mathbf{dim}(P_c) = n \tag{4}$$

i.e., $P_c \in S_l$, it is indeed a low-dimensional prototype, then we check the following condition:

$$\|\mathbf{x} - W_{P_c}\|_{S_l} \leq T_{P_c}^l \tag{5}$$

where $T_{P_c}^l$ represents the similarity threshold of P_c in space S_l. Formula (5) means that the distance between P_c and $\mathbf{x}$ is less than the similarity threshold $T_{P_c}^l$, i.e., sample $\mathbf{x}$ is very similar with P_c in space S_l. Then P_c is activated, we add the accumulated times of activation of P_c in space S_l by 1, i.e., $M_{P_c}^l = M_{P_c}^l + 1$. Then we map P_c to space S_h as:

$$W^l = W_{P_c} + (1/M_{P_c}^l)(\mathbf{x}^l - W_{P_c}) \tag{6}$$

$$W^{h-l} = \mathbf{x}^{h-l} \tag{7}$$

$$W_{P_c} = (W^l, W^{h-l}) \tag{8}$$

where $\mathbf{x}^l$ represents the attributes of space S_l, i.e., $\mathbf{x}^l = (x_1, x_2, ..., x_n)$. $\mathbf{x}^{h-l}$ represents the attributes of the new perception neurons, i.e., $\mathbf{x}^{h-l} = (x_{n+1}, x_{n+2}, ..., x_{n+m})$. According to formulas (6)–(8), the weights of P_c learned from original perception neurons are moved toward $\mathbf{x}$, the weights from new perception neurons are added in the tail of W_{P_c}. Meanwhile, the threshold $T_{P_c}^h$ (the similarity threshold of P_c in space S_h) is initialized to $+\infty$, and $M_{P_c}^h$ (the accumulated times of activation of P_c in space S_h) is initialized to 1.

If condition (5) is not satisfied, i.e., sample $\mathbf{x}$ is not similar with P_c in space S_l. We will create a new prototype P_{new} for

$\mathbf{x}$ as:

$$\begin{aligned} &W_{new} = \mathbf{x} \\ &M^l_{new} = 1, \quad M^h_{new} = 1 \\ &T^l_{new} = +\infty, \quad T^h_{new} = +\infty \end{aligned} \tag{9}$$

If condition (4) is not satisfied, i.e., P_c already belongs to S_h, we will deal with $\mathbf{x}$ using P_a and P_b. If

$$\|\mathbf{x} - W_{P_a}\|_{S_h} = 0 \tag{10}$$

i.e., sample $\mathbf{x}$ is equal to P_a, then P_a is activated and the time of activation $M^l_{P_a}$ and $M^h_{P_a}$ will be added by 1. Else, we will check the following condition:

$$\|\mathbf{x} - W_{P_a}\|_{S_h} \leq T^h_{P_a} \ \textbf{and} \ \|\mathbf{x} - W_{P_b}\|_{S_h} \leq T^h_{P_b} \tag{11}$$

If it is satisfied, i.e., P_a and P_b are activated simultaneously, we first add $M^l_{P_a}$ and $M^h_{P_a}$ by 1. Then we move prototype P_a toward $\mathbf{x}$ as:

$$\begin{aligned} W^l_{P_a} &= W^l_{P_a} + (1/M^l_{P_a})(\mathbf{x}^l - W^l_{P_a}) \\ W^{h-l}_{P_a} &= W^{h-l}_{P_a} + (1/M^h_{P_a})(\mathbf{x}^{h-l} - W^{h-l}_{P_a}) \end{aligned} \tag{12}$$

where W^l_* represents the attributes of space S_l, i.e., W^l_*=$(w_1, w_2, ..., w_n)$. W^{h-l}_* represents the attributes of new perception neurons, i.e., W^{h-l}_*=$(w_{n+1}, w_{n+2}, ..., w_{n+m})$. If formula (11) is not satisfied, we will create a new prototype for $\mathbf{x}$ using formula (9).

3.4 Self-Organizing

Self-Organizing process is conducted through connection establishing, strengthening, weakening and removing.

Connection establishing and strengthening are conducted according to the Hebbian learning rule. If P_a and P_b are activated simultaneously, i.e., formula (11) is satisfied, we will establish a connection between P_a and P_b; if there is no connection between them, then we set the outdate degree $Age_{(a,b)}$ to 0 to represent the connection is the most recent. If there is already a connection between P_a and P_b, we will set $Age_{(a,b)}$ to 0 to strengthen the connection.

For the dynamically changing environment, the prototypes change their locations slowly during learning. Prototypes that are connected to each other at an early stage might not be near at advanced stage, and we remove such connections. If prototype P_i is mapped to space S_h or moved toward input sample $\mathbf{x}$, we weaken the connections emanating from P_i by increasing the outdate degree of these connections as:

$$Age_{(i,j)} = Age_{(i,j)} + 1, \quad P_j \in N_{P_i} \tag{13}$$

where $Age_{(i,j)}$ is the outdate degree of the connection between prototype P_i and P_j. N_{P_i} is the neighbor prototype set of P_i, i.e., the set of the prototypes connected to P_i. Connections whose Age is larger than a predefined threshold Age_{max} will be removed.

3.5 Adaptation of the Similarity Threshold

Similarity threshold T is used to make the decision whether to add a new prototype or move an existing prototype when an input sample comes. In PEN, we do not use global predefined threshold, we give each prototype a similarity threshold which is adaptively changing with the environment. The threshold of P_i is decided by the prototype-distribution around P_i: If P_i does not have neighbor prototype, i.e., no other prototype is directly connected to P_i, the similarity threshold of P_i is calculated using the minimum distance between P_i and all the other prototypes:

$$T^k_{P_i} = \min_{P_j \in P \setminus P_i} \|W_{P_i} - W_{P_j}\|_{S_k}, \ \ k = l, h \tag{14}$$

If P_i has neighbor prototypes, i.e., some prototype(s) is directly connected to P_i, the similarity threshold is calculated using the maximum distance between P_i and its neighbor prototypes:

$$T^k_{P_i} = \max_{P_j \in N_{P_i}} \|W_{P_i} - W_{P_j}\|_{S_k}, \ \ k = l, h \tag{15}$$

For the prototype P_i belongs to space S_l, we only calculate threshold $T^l_{P_i}$. For the prototype P_i belongs to space S_h, we calculate threshold $T^l_{P_i}$ and $T^h_{P_i}$. To improve the computational efficiency, in practice, we only update the thresholds of P_a, P_b and P_c when an input sample comes.

3.6 Prototype Pruning

Prototype pruning is conducted after every λ samples learned. The prototypes existed before the new sensory neurons appear are not the targets to be pruned, because these prototypes are gained by a period of learning and they are already pruned by PEN during that period, they are a "*knowledge base*" of PEN now. The prototypes which are created after new sensory neurons appear will be pruned, we use set Q to represent these prototypes. The isolated prototype with small M value in Q will be removed. First, we calculate the mean value of $M^h_{P_i}$ as:

$$M^h_{mean} = \sum_{P_i \in P^h} M^h_{P_i} / |P^h| \tag{16}$$

where P^h represents the prototypes belong to space S_h, $|P^h|$ represents the element number of set P^h. If

$$|N_{Q_i}| = 0 \ \textbf{and} \ M^h_{Q_i} < M^h_{mean} \tag{17}$$

or if

$$|N_{Q_i}| = 1 \ \textbf{and} \ M^h_{Q_i} < c \times M^h_{mean} \tag{18}$$

is satisfied, where $Q_i \in Q$, $0 \leq c \leq 1$, large c means much noise, vice versa. $|N_{Q_i}|$ represents the neighbor number of prototype Q_i, we will remove prototype Q_i, connections from Q_i are also removed simultaneously.

3.7 Postprocessing

Postprocessing is enabled after all prototypes are mapped to space S_h. This procedure simplifies the learning procedure.

Because all prototypes are mapped to space S_h, there is no prototype needs to be mapped to space S_h. We do not find winner prototype P_c when an input sample $\mathbf{x} \in S_h$ comes, we only find prototypes P_a and P_b. Then the procedure of mapping a prototype to space S_h does not need to be executed anymore (procedure from formulas (4)–(9)), we directly deal with $\mathbf{x}$ using P_a and P_b, i.e., the procedure of moving P_a

toward $\mathbf{x}$ or creating a new prototype using $\mathbf{x}$. We replace parameters $M^l_{P_i}$ and $M^h_{P_i}$ with one parameter M_{P_i}:

$$M_{P_i} = (M^h_{P_i} + M^l_{P_i})/2, \quad P_i \in P \tag{19}$$

weight vector updating formula (12) can be simplified into:

$$W_{P_a} = W_{P_a} + (1/M_{P_a})(\mathbf{x} - W_{P_a}) \tag{20}$$

For formula (9) which is used to create new prototype, the parameters M^l_{new} and T^l_{new} are no longer needed. Self-Organizing process remains unchanged. The parameter $T^l_{P_i}$ of each prototype P_i can be removed, then we only update threshold $T^h_{P_i}$ of each prototype P_i. The prototype pruning is conducted on all prototypes, i.e., set P, instead of set Q.

Note: For the prototypes which can not be mapped to space S_h after a long period of learning (in the experiment, after all samples $\mathbf{x} \in S_h$ are learned), we remove them because they are potential "distortion" prototypes (see Figure 7).

If some new sensory neurons emerge later, we take the current space S_h as S_l, the space with dimension of these new sensory neurons as new S_h, then we get a recursive process.

Meanwhile, the initial learning stage before the m sensory neurons emerge is conducted similarly as the postprocessing. When an input sample $\mathbf{x} \in \mathbb{R}^n$ comes, we find the winner and runner-up prototypes P_a and P_b in space $\mathbb{R}^n$ using formulas (2) and (3), where h is replaced by l. Then we decide whether to create a new prototype or move the winner prototype according to formulas (10) and (11), where h is replaced by l. For formula (9), the parameters M^h_{new} and T^h_{new} do not exist. The weight vector updating formula is the same as (20), where M_* is the activation time in space $\mathbb{R}^n$. Self-Organizing process remains unchanged, we only need replace h with l in condition formula (11). The threshold is updated by (14) and (15), where $k = l$. The prototype pruning process will be conducted on all prototypes instead of set Q, similarly, h is replaced by l in formulas (16)–(18).

4 Experiments

4.1 Artificial Feature Data

We conduct this experiment on the artificial feature data as shown in Figure 4. The data set is separated into five parts, and each data set contains 2000 samples, and we add 5% random noise. Set A satisfies 3-D sinusoidal distribution. Sets B and C satisfy 3-D ring distribution. Sets D and E satisfy same 2-D Gaussian distribution in the X-Y space but different 3-D Gaussian distribution in the X-Y-Z space. Sets D and E show an actual phenomenon: in the early period, we have few perception neurons, the information we gained is only enough to make a rough cognition. However, with the evolution of perception, we get more information through some new perception neurons, then we can make a further cognition. It is a *Cognition Deepening Process*. The parameters of PEN are: $\lambda = 200$, $Age_{max} = 200$, $c = 1.0$.

In the experiment, we first give PEN two sensory neurons, i.e. one sensory neuron receives the X-dimension data, the other receives the Y-dimension data. After all samples have been learned, we give PEN another sensory neuron to receive the Z-dimension data, therefore, PEN is able to perceive the X-Y-Z space. Then we feed all samples to PEN again. 10000 samples are randomly fed to the network.

The learning results are shown in Figure 5. As shown in Figure 5(a), PEN gets a satisfactory learning result, the learned prototypes fit the learning data very well. Figure 5(b) shows the learning result after PEN gets a new sensory neuron, which receives the input signal from Z-dimension. All prototypes are mapped to the X-Y-Z space, the prototypes fit the learning data very well. Meanwhile, sets B and C are no longer concentric rings in the X-Y-Z space, sets D and E which cannot be distinguished in the X-Y space are separated from each other in the X-Y-Z space. PEN comes to a "new world" through the new sensory neuron Z.

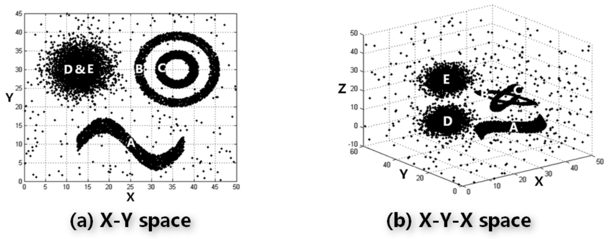

Figure 4: Artificial feature data. (a) Representation of the data in the X-Y space. (b) Representation of the data in the X-Y-Z space.

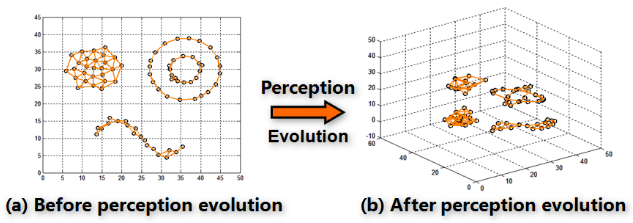

Figure 5: (a) learning result in the X-Y space (80 prototypes). (b) learning result after new sensory neurons appear (117 prototypes). Black circles denote the learned prototypes and yellow lines are the connections between prototypes.

4.2 Real-World Data

The RGB-D data set [Lai *et al.*, 2011] is used in this experiment. We use 20 objects, and the first 25 images (the object in each image is rotated by an angle) of each object are for learning. In the testing phase, we use 200 more images, the testing images are taken in different angles from the training images, and each object includes 10 testing images.

As mentioned in [Jacobs *et al.*, 2007], the dichromatic color vision comes from just two kinds of visual pigments: one absorbs blue light and the other one is sensitive to green light. The trichromatic color vision has one more visual pigment which absorbs red light. Therefore, in the experiment, we first give PEN two perception channels to receive the G and B parts of the RGB images (Period I), after all samples have been learned, we give another perception channel to PEN to receive the R part of the RGB images, i.e., PEN is used to perceive RGB color synchronously (Period II).

In recent years, depth cameras are widely used. We assume a scene: there is a robot using a RGB camera as its "eyes", now, we install a depth cameras to the robot's "eyes", then the robot has one more way to perceive the real-world. Thus, after the learning has finished in the RGB space, we give another perception channel to PEN to receive the depth information, i.e., PEN is used to perceive RGB color and spatial information synchronously (Period III). Overall, the perception evolution is as follows:

$$\textbf{GB} \rightsquigarrow \textbf{RGB} \rightsquigarrow \textbf{RGB-D} \tag{21}$$

We set $\lambda = 200$, $Age_{max} = 200$, $c = 0.5$ for PEN. We conduct this experiment in (1) Closed environment; and (2) Open-ended environment (for the Stability-Plasticity Dilemma [Carpenter and Grossberg, 1988]). We do 10000 learning iterations for each Period. In the closed environment, training images are randomly chosen from the training set in each iteration. In the open-ended environment, we first give the images of the first 10 objects in the first 5000 iterations, training images are randomly chosen during learning. After that, we give the images of the remaining 10 "new" objects in the second 5000 iterations, training images are also randomly chosen. We conduct the experiment 100 times in different random sequences of samples.

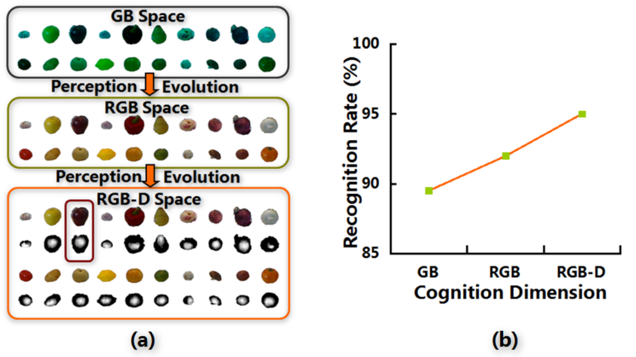

Figure 6: (a) Prototypes are mapped to high-dimensional space perfectly when new dimension of perception emerges. (b) Trend of the recognition rate according to the cognition dimension.

The number of the prototypes learned by PEN in the closed environment ranges from 143 to 171 for 100 times experiments, the mean value of the prototype number is 155.6. In the open-ended environment, the number of the prototypes learned by PEN ranges from 225 to 236, the mean value is 228.4. Figure 6(a) shows the learning result of PEN from GB space to RGB-D space of 20 prototypes (one prototype for each object). Due to space constraint, we do not show all of the prototypes, learning results of the other prototypes are similar to these 20 prototypes. In the RGB-D space of Figure 6(a), the RGB image and the depth image of a prototype are shown separately, actually, the two images are integrated in one prototype by PEN. In Figure 6(a), the prototypes are mapped to high-dimensional space perfectly when PEN gets new dimension of perception. In the GB space, there are few kinds of colors. After dimension R comes, much more distinguishable colors appear. At last, after dimension D comes, PEN acquires both the image information and the spatial information of the objects. The robot comes to a new world with the concept of the "space."

Figure 6(b) shows the trend of the recognition rate according to the cognition dimension. In the GB space, PEN can only "see" green and blue color, and it classifies the GB part of the testing images by nearest neighbor method. RGB and RGB-D space are treated similarly. In Figure 6(b), the recognition rate is increasing with the deepening of the cognition dimension.

Figure 7: "Distortion" prototypes learned by other methods.

We also compare PEN with some classical and state-of-the-art methods including SOM, GNG, ASOINN, TopoART, and LB-SOINN. We run PEN in two conditions: *(condition 1)* To ensure the consistent experimental environment with other methods, we directly use RGB-D images to train PEN, i.e., give PEN RGB-G four channels directly without "perception evolution"; *(condition 2)* To measure the learning results of PEN with "perception evolution", we run PEN as formula (21). We conduct the experiment in both closed and open-ended environments. To make SOM and GNG generate similar scale of prototypes with PEN, we set the grid size of SOM as 12×13 in the closed environment and 15×15 in the open-ended environment. λ of GNG is set to 65 in the closed environment and 45 in the open-ended environment. The parameters of ASOINN, TopoART and LB-SOINN are set following suggestions of their papers.

As shown in Figure 7, there are many "distortion" prototypes generated by the other methods, these prototypes merge many different types of objects. They are meaningless and useless. To quantify the comparison, we give each prototype an entropy to measure the "purity" of the prototype. Assume there are k objects $\{o_1, o_2, ..., o_k\}$, prototype P_i merges m samples coming from k objects, the entropy of P_i can be calculated as: $Entr(P_i){=}-\sum_{i=1}^{k} p(o_i)\log p(o_i)$.

For W_{P_i} is weighted summed by the weight vectors of the m samples, $p(o_i)$ represents the proportion of the weight vector of o_i in W_{P_i}. We calculate the entropy of each prototype and get the mean value ME. As shown in Table 1, the ME of PEN is much smaller than the other methods, it means that PEN seldom produces "distortion" prototypes. We also calculate the mean quantization errors MQE. Table 1 shows that PEN gets a much smaller MQE than the other methods.

At last, nearest neighbor is used to classify the testing images. As shown in Table 1, the correct recognition CR ratio of PEN is much higher (the baseline CR of supervised method LibSVM [Chang and Lin, 2011] is 93.0%). The CR of other methods is very unstable, significant decrease of the CR in one environment is marked by ↓ in the table. They suf-

Table 1: Statistical results of 100 experiments (mean+std). PEN$^{\circ}$ and PEN$^{\bullet}$ represent the results of *condition 1* and *condition 2* respectively. "—" represents the measurement is Not Applicable for the methods. The best and second best results are in bold.

Environment		SOM	GNG	ASOINN	TopoART	LB-SOINN	PEN$^{\circ}$	PEN$^{\bullet}$
Closed	ME	2.51±0.03	2.49±0.03	0.92±0.01	1.03±0.01	1.04±0.02	**0.025±0.003**	**0.021±0.004**
	MQE	7.47±0.31	5.89±0.28	3.05±0.21	—	3.01±0.20	**1.49±0.15**	**1.43±0.18**
	CR	73.5±0.80%	76.6±0.80%	88.2±0.71%	83.5±0.55% ↓	90.1±0.67%	**92.1±0.50%**	**94.2±0.58%**
Open-ended	ME	2.53±0.06	2.51±0.06	1.13±0.04	0.90±0.01	1.02±0.03	**0.024±0.003**	**0.021±0.003**
	MQE	8.50±0.40	6.17±0.38	3.28±0.27	—	2.98±0.18	**1.45±0.15**	**1.41±0.15**
	CR	55.5±0.86% ↓	57.9±0.79% ↓	83.4±0.64% ↓	89.2±0.57%	85.2±0.61% ↓	**92.3±0.53%**	**94.5±0.51%**

fer the Stability-Plasticity Dilemma; The CR of PEN is very stable in both environments.

4.3 Experiment Summary

In the experiments, we see that PEN is able to create new prototypes for new categories of objects effectively; this implies that PEN has a degree of freedom on the *breadth of cognition*. PEN also permits the emergence of new dimension of perception; it means that PEN has a degree of freedom on the *depth of cognition*. Meanwhile, the new dimension of perception will promote the breadth of cognition, i.e., PEN will find some new categories of objects in the "new perceived world".

5 Conclusion

The proposed Perception Evolution Network is a biologically inspired computing model which permits the emergence of a new dimension of perception. Zhou and Chen [Zhou and Chen, 2002], based on the supervised learning view, gave three types of incremental learning including example-incremental learning, class-incremental learning [Da *et al.*, 2014], and attribute-incremental learning. Based on the unsupervised learning view, we think the incremental learning should be classified into: (1) the extension of the connotation of the existing knowledge system; and (2) the discovery and fusion of new knowledge system. The two points are another explanation of the breadth and depth of the cognition. They are an intrinsic whole that complement each other.

From the experiment, we see that PEN can deal with many potential practical problems. For example, if we install new sensors to a robot to expand its sensing capability, with PEN, we do not need to retrain the robot offline from scratch, the information gathered from the new installed sensors is fused with the existing knowledge online by PEN automatically. This means PEN will have broad applications such as robot system, information fusion, video game [Stanley *et al.*, 2005].

The influence of the order of sensor addition will be discussed in the future. This is an extremely interesting topic which aims to see the outcome of different *evolution orders*.

Acknowledgments

The authors would like to thank the anonymous reviewers for their time and effort and their valuable suggestions.

References

[Carpenter and Grossberg, 1988] G. A. Carpenter and S. Grossberg. The art of adaptive pattern recognition by self-organizing neural network. *IEEE Computer*, 21(3):77–88, 1988.

[Chang and Lin, 2011] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

[Da *et al.*, 2014] Q. Da, Y. Yu, and Z.-H. Zhou. Learning with augmented class by exploiting unlabeled data. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1760–1766, Quebec, Canada, July 2014.

[Fritzke, 1994] B. Fritzke. Growing cell structures self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9):1449–1460, 1994.

[Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 625–632, Vail, Colorado, Novemeber 1995.

[Jacobs *et al.*, 2007] G. H. Jacobs, G. A. Williams, H. Cahill, and J. Nathans. Emergence of novel color vision in mice engineered to express a human cone photopigment. *Science*, 315(5819):1723–1725, 2007.

[Kohonen, 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[Lai *et al.*, 2011] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Proceeding of the IEEE International Conference on Robotics and Automation*, pages 1817–1824, Beijing, China, May 2011.

[Rosenblatt, 1958] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[Shen and Hasegawa, 2006] F. Shen and O. Hasegawa. An incremental network for on-line unsupervised classification and topology learning. *Neural Networks*, 19(1):90–106, 2006.

[Shen and Hasegawa, 2008] F. Shen and O. Hasegawa. A fast nearest neighbor classifier based on self-organizing incremental neural network. *Neural Networks*, 21(10):1537–1547, 2008.

[Stanley *et al.*, 2005] K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Real-time neuroevolution in the nero video game. *IEEE Transactions on Evolutionary Computation*, 9(6):653–668, 2005.

[Tscherepanow *et al.*, 2011] M. Tscherepanow, M. Kortkamp, and M. Kammer. A hierarchical art network for the stable incremental learning of topological structures and associations from noisy data. *Neural Networks*, 24(8):906–916, 2011.

[Zhang *et al.*, 2014] H. Zhang, X. Xiao, and O. Hasegawa. A load-balancing self-organizing incremental neural network. *IEEE Transactions on Neural Networks and Learning System*, 25(6), 2014.

[Zhou and Chen, 2002] Z.-H. Zhou and Z.-Q. Chen. Hybrid decision tree. *Knowledge-Based Systems*, 15(8):515–528, 2002.