Continuous Learning of Context-dependent Processing in Neural Networks

Guanxiong Zeng [1,2,*], **Yang Chen** [1,*], **Bo Cui** [1,2] **and Shan Yu** [1,2,3,†]

[1]*Brainnetome Center and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China.*
[2]*University of Chinese Academy of Sciences, 100049 Beijing, China.*
[3]*CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, 100190 Beijing, China.*

* These authors contributed equally to this work.

† Correspondence shan.yu@nlpr.ia.ac.cn

arXiv:1810.01256v2 [cs.LG] 5 Oct 2018

ABSTRACT

Deep artificial neural networks (DNNs) are powerful tools for recognition and classification as they learn sophisticated mapping rules between the inputs and the outputs. However, the rules that learned by the majority of current DNNs used for pattern recognition are largely fixed and do not vary with different conditions. This limits the network's ability to work in more complex and dynamical situations in which the mapping rules themselves are not fixed but constantly change according to contexts, such as different environments and goals. Inspired by the role of the prefrontal cortex (PFC) in mediating context-dependent processing in the primate brain, here we propose a novel approach, involving a learning algorithm named orthogonal weights modification (OWM) with the addition of a PFC-like module, that enables networks to continually learn different mapping rules in a context-dependent way. We demonstrate that with OWM to protect previously acquired knowledge, the networks could sequentially learn up to thousands of different mapping rules without interference, and needing as few as $\sim$10 samples to learn each, reaching a human level ability in online, continual learning. In addition, by using a PFC-like module to enable contextual information to modulate the representation of sensory features, a network could sequentially learn different, context-specific mappings for identical stimuli. Taken together, these approaches allow us to teach a single network numerous context-dependent mapping rules in an online, continual manner. This would enable highly compact systems to gradually learn myriad of regularities of the real world and eventually behave appropriately within it.

1   INTRODUCTION

One of the hallmarks of high-level intelligence is flexibility [1]. Humans can respond differentially to the same stimulus according to contexts, such as different goals, environments, and internal states [2–7]. The prefrontal cortex (PFC), which is highly elaborated in primates, is pivotal for such an ability [6–10]. The PFC can quickly learn "rules of the game" and dynamically apply them to map the sensory inputs to different actions in a context-dependent way [11–13]. This process, named cognitive control, allows primates to behave appropriately in an unlimited number of situations [10, 14]. With impaired PFC, the subjects' reaction are largely dictated by stronger sensory stimuli and they lose the ability to respond to task-related, weaker stimuli [15]. In addition, these subjects tend to stubbornly follow the established rules in behavioral tasks even when the rules no longer bring desirable outcome, i.e., they

lose the ability to dynamically adjust the mapping between the sensory inputs and the motor outputs [16]. Not only the experiments with PFC-impaired human patients show that this areas is the key for flexible, context-dependent processing, numerous electrophysiological studies in non-human primates have also demonstrated that the PFC neurons can indeed represent various contextual-related information [10]. Such ability of flexible, context-dependent processing empowered by the PFC is quite different from the current artificial deep neural networks (DNNs). DNNs are very powerful in extracting high-level features from raw sensory data and learning sophisticated mapping rules for pattern detection, recognition, and classification [17]. However, except for the recently proposed approaches of meta-learning and few-shot learning [18–20], in majority of networks the responses are largely dictated by the sensory inputs, exhibiting stereotyped input-output mappings, and these mappings are usually fixed once the training is completed. Therefore, the current DNNs lack enough flexibility to work in complex situations in which 1) the mapping rules may change according to context and 2) these rules need to be learned "on the go" from a small number of training trials. This constitutes a significant ability gap between DNNs and human brains.

2   ORTHOGONAL WEIGHTS MODIFICATION (OWM)

Here we propose an approach that enables one neural network to quickly learn various mapping rules in a context-dependent way. To this end, the first step is to have a method for efficient and scalable continual learning, i.e., to learn different mappings sequentially, one at a time. Such an ability is crucial to humans as well as neural networks for two reasons: 1) there are too many possible contexts to learn all mappings concurrently, and 2) the useful mappings cannot be pre-determined but must be learned when corresponding contexts are encountered. Therefore, in the present study, to protect previously learned mappings from being erased by subsequent training, i.e., to avoid catastrophic forgetting [21–23], we propose the method of Orthogonal Weights Modification (OWM). Specifically, when training a network consecutively for different tasks, its weights are only allowed to be modified in the direction orthogonal to the subspace spanned by all inputs on which the network has been trained (termed input space hereafter) (Fig. 1A and Fig.S1). This ensures that new learning processes will not interfere with the learned tasks, as weight changes in the network as a whole do not interact with the old inputs. Consequently, combined with a gradient descent-based search, OWM helps the network to find a weight configuration that can accomplish new tasks while keeping the performance of the learned tasks unchanged (Fig. 1B). In OWM, the projector used to find the orthogonal direction to the input space is defined as $\mathbf{P} = \mathbf{I} - \mathbf{A}\left(\mathbf{A^T A} + \alpha \mathbf{I}\right)^{-1}\mathbf{A}$, where matrix $\mathbf{A}$ consists of all previously trained input vectors as its columns $\mathbf{A} = [\mathbf{x_1}, \cdots, \mathbf{x_n}]$ and $\mathbf{I}$ is a unit matrix multiplied with a relatively small constant α. The learning-induced modification of weights is then determined by $\Delta\mathbf{W} = \kappa\mathbf{P}\Delta\mathbf{W^{BP}}$, where κ is the learning rate and $\Delta\mathbf{W^{BP}}$ is the weights adjustment calculated according to the standard backpropagation. We note that to calculate $\mathbf{P}$, an iterative method can be used (see Methods for details). The algorithm does not need to store all previous inputs $\mathbf{A}$. Instead, only the current inputs and the projector for the last task are needed. This iterative method is essentially the Recursive Least Square (RLS) algorithm [24, 25] (see Methods), which has been used to train feedforward and recurrent neural networks to achieve fast convergence [26, 27], tame chaotic activities [28] and avoid interference between consecutively loaded patterns or tasks [29, 30].

We first tested the performance of OWM on the tasks of learning to classify handwritten digits (MNIST dataset) sequentially. In several benchmark tasks, OWM exhibited either superior or equally good performance in comparison with other methods for continual learning [30–33] (Tables S1, S2).
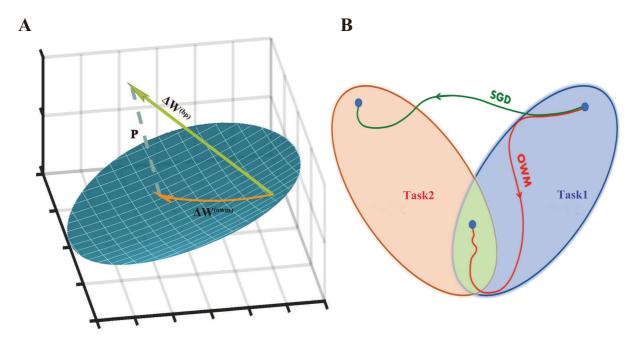
Figure 1. **Schematic diagram of OWM**. **(A)** In the training process for a new task, the original weight modification calculated by the standard backpropagation, $\Delta \mathbf{W^{BP}}$, is projected to the subspace (dark green surface) in which good performance for learned tasks has been achieved. As a result, the weight modification actually implemented is $\Delta \mathbf{W^{OWM}}$. This process ensures that the weights configuration after learning the new task is still within the same subspace. **(B)** With OWM, the training process searches for configurations that can accomplish Task 2 (the pale red area), within the subspace that enables the network to accomplish Task 1 (the blue area). A successful search necessarily stops at a position inside the overlapping subspace (the light green area). In comparison, the solution obtained by simple stochastic gradient descent search (SGD) is more likely to end outside this overlapping area.

To examine whether OWM is scalable, i.e., whether it can be applied to learn more sophisticated tasks, regarding both the complexity of the inputs and the number of different mappings, we tested the network's ability in learning to classify pictures of natural scenes (ImageNet dataset). For these tasks, pre-trained feature extractors (Table 1) were used to analyze the raw images. The feature vectors were fed into an OWM-trained classifier to learn the mapping between combinations of features and the label of individual classes. With a sequential training paradigm, this process is analogous to humans' learning to form new concepts of objects in cognition, with fully developed feature extractors-sensory cortices. We observed remarkable performance of the system in sequentially learning to classify up to 1000 individual categories, with the final accuracy closely approaching the results obtained by training the system to classify all categories concurrently (Table 1). These results suggest that, by using OWM, the performance of the system in classification approached the limit set by the front-end feature extractor, with the liability caused by sequential learning itself effectively mitigated.

Data Set	Classes	Feature Extractor	Concurrent Training by SGD (%)	Sequential Training by OWM (%)	Sequential Training by SGD (%)
ImageNet	1000	ResNet152	78.31	75.24	4.27
CASIA-HWDB1.1	3755	ResNet18	97.46	93.46	35.86

Table 1. The performance of sequential learning achieved by OWM in comparison with traditional concurrent training method in various datasets. ResNet was adopted from [34].
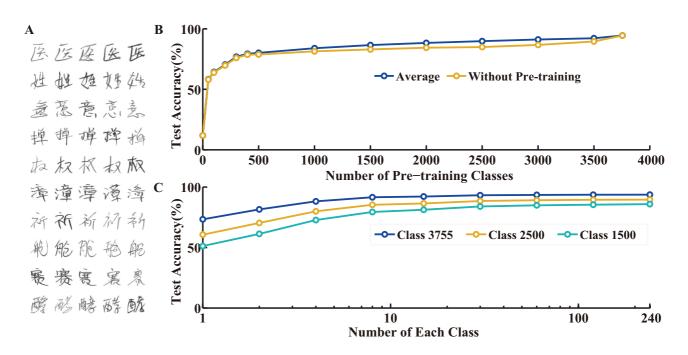
Figure 2. Online learning with small sample size achieved by OWM in recognizing Chinese characters. (A) Examples showing 10 characters with five samples for each. **(B)** Classification accuracy is plotted as a function of the number of classes used for pre-training the feature extractor. The performance was assessed based on classifying all characters (blue) or characters that were not included in the pre-training (orange). **(C)** Classification accuracy is plotted as a function of the sample size used for sequential training, obtained with feature extractors having different degrees of pre-training (color-coded).

To have a comparison to human's ability in continuously forming new conceptual categories, we tested the performance of OWM in learning handwritten Chinese characters sequentially. In total there are 3755 characters forming the level I vocabulary, which constitutes more than 99% of the usage frequency in written Chinese literature [35] (see Fig. 2A for exemplars). We found that, combined with a pre-trained feature extractor (Table 1), a classifier trained with OWM could learn to recognize all 3755 characters sequentially, resulted in the final recognition accuracy of 93% across all classes (Table 1), which was very close to human's performance in recognizing handwritten Chinese characters ($\sim 96\%$) [36]. Considering the fact that humans learn those characters over years and that their learning necessarily contains revising, these results suggest that our method endows neural networks with the human level ability in continuously learning new mappings between sensory features and class labels.

In the results mentioned above, pre-trained feature extractors were used to provide the feature vectors for the OWM-trained classifier. Next, we examined whether the classifier can learn categories that the feature extractor has never seen before. The results shown in Fig. 2B indicate that the answer was affirmative. For example, the feature extractor trained with randomly selected 500 characters (out of 3755, less than 15% of categories) could already support the classifier to sequentially learn the remaining 3255 characters with near 80% accuracy (the chance level is $1/3255$), demonstrating that the network could sequentially learn new categories it has never encountered. This would remove the usual distinction between the training and testing phases for DNN, allowing the system's capacity to keep increasing with more interactions with the environments.

Another important question is how quickly the OWM-trained classifier can learn. In Fig. 2C, we showed that it needed very small sample size to learn new mappings. For Chinese characters, a single sample for

individual classes in sequential training could already increase the performance to be well above 50% (chance level $\leq 0.05\%$), and ~ 10 samples per class were enough to approach the learning plateau. These results demonstrate an impressive speed of learning for the system, which would allow it to continuously form new categories not from seeing a large number of training samples, but from just a few encounters with the members of individual categories.

3 PFC-LIKE MODULE

Although a system that can learn many different mapping rules in an online, sequential manner and needs only small sample size is highly desirable, such a system cannot accomplish context-dependent learning by itself. To achieve that, the contextual information need to interact with the sensory information properly to 1) change the representation of sensory information to allow different processing across contexts, but 2) not to distort the content of sensory information. To this end, here we adopted a solution inspired by primate PFC. The PFC receives sensory inputs as well as the contextual information, which enables it to choose the sensory features that are most relevant to the present task to guide action. To mimic such an architecture, we added a module before the OWM-trained classifier, which was fed with both sensory feature vectors and contextual information (Fig. 3A). Mathematically, this module serves the role of rotating the sensory input space according to the contextual information (Fig. 3B, see Methods for details), thereby changing the representation of sensory information without interfering with its content. The rotation of the input space also makes it possible for OWM to be applied for identical sensory inputs in different contexts. To demonstrate the effectiveness of this PFC-like module, we trained the system to classify a set of faces according to 40 different attributes [37], i.e., to learn 40 different mappings sequentially with the same sensory inputs. The contextual information was chosen randomly for individual tasks to demonstrate that the system can work with arbitrary coding schemes for context. Fig. 3C shows that the system sequentially learned all 40 different, yet context-specific mapping rules with a single classifier, with the accuracy very close to that achieved by multi-task training, in which the network was trained to classify all 40 attributes by using 40 separate classifiers (Fig. 3D). In addition, similar to the results obtained in learning Chinese characters, the network exhibited an ability to learn context-dependent processing quickly. With the simple task of classifying males from females, ~ 20 faces were enough to reach the learning plateau. Even for more difficult tasks such as classifying whether a face is attractive, ~ 100 samples were enough to reach the plateau (Fig. 3E), indicating the ability to adapt quickly in highly dynamic environments with regularities changing with the contexts.

4 DISCUSSIONS

If we view traditional DNNs as powerful sensory processing modules, the current approach could be understood as adding an efficient cognitive module to the system. This architecture is inspired by the primate brain. For example, the primate visual pathway is dedicated to analyzing raw visual images and eventually to represent it with ~ 100 features in higher visual areas such as the inferotemporal cortex [38]. The outputs of this "feature extractor" are then sent to the PFC for object identification and categorization [39–41]. The training of the feature extractor is difficult and time-consuming. In humans, it takes years or even decades for higher visual cortices to be fully developed and to reach peak performance [42]. However, with sufficiently developed visual cortices, humans can quickly learn new category of visual object, often by seeing just a few positive examples [43]. By adding a cognitive module to DNN-based feature extractors, here we found a qualitatively similar behavioral trend in neural networks, suggesting that part of the mechanisms underlying fast concept forming in humans may be
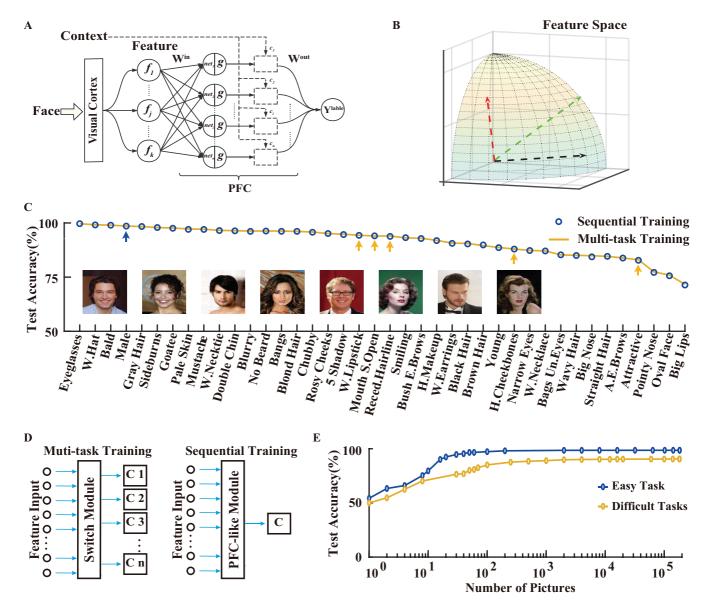
Figure 3. Achieving context-dependent sequential learning by OWM and a PFC-like module. (**A**) Schematic diagram of the network architecture. In comparison to the primate brain, the feature extractor plays the role similar to sensory cortices. It sends processed sensory information as inputs to the "cognitive" module similar to the PFC. Besides the sensory inputs, the PFC also receives the contextual information, which changes the representation of the sensory inputs. The weights transmitting the context-modulated sensory information to the classifier, W^{out}, are trained by OWM. (**B**)Schematic diagram showing the role of the PFC-like module as rotating the inputs in the feature space (see Methods for details). (**C**) Performance of sequentially learning to classify faces by 40 different attributes, each associated with a unique contextual signal, compared with the results obtained by multi-task training. Tasks are sorted by the test accuracy. Insets: examples of input faces. (**D**) Schematic diagrams showing the network architecture for multi-task (left) and sequential (right) training. C, classifier. To achieve context-dependent processing, in multi-task training a switch module and n classifiers are needed, where n is the number of different attributes. (**E**) Classification accuracies for a relatively easy task (gender, blue curve) and five more difficult, sequentially learned tasks (Attractiveness, etc.; orange curve; mean results across all five tasks are shown) are plotted as a function of the training sample size. The tasks and corresponding performance obtained by training on the full dataset are marked with arrows in (**C**).

understood from a connectionist perspective. In addition to the role of supporting fast concept learning, another function of the PFC is to represent the contextual information in the form of working memory [10], which guides the selection of the sensory features that are most relevant for the current task [6]. Such an architecture gives rise to the flexibility exhibited in primates' behavior and we demonstrated here that it can do the same for artificial neural networks. Interestingly, we found that in the PFC-like module in the network, the neuronal responses showed mixed selectivity to sensory features, contexts, as well as their combinations (Fig. S2), similar to what has been found for real PFC neurons [44]. It would be informative to see whether the rotation of input space adopted in our PFC-like module captures the operation carried out in the real PFC. For tasks similar to the face classification tested above, one possible solution to achieve context-dependent processing is to adding additional classifier outputs for each new task/context. However, this approach only works if there is no hidden layer between the feature extractor and the final output layer. Otherwise the shared weights between different classifier outputs will suffer from catastrophic forgetting in continuous learning, especially if the inputs are the same for all contexts. More importantly, adding additional classifier outputs (and all related weights) for each new task/context would lead to increasingly complex and bulky systems. Due to the fact that the total number of possible context could be arbitrarily large, such a solution is clearly not scalable. Finally, for artificial intelligence systems, the importance of the PFC-module would depend on applications. In a scenario that a compact system need to sequentially learn numerous contexts, similar to a human individual needing to do in his/her lifetime, the ability enabled by the PFC-module to reuse the feature representation and the classifier would be of paramount importance.

As the present results demonstrated, an efficient and scalable algorithm of continual learning is crucial to make the added cognitive module versatile and, at the same time, compact. In continual learning, to preserve previously acquired knowledge while leaving enough space for subsequent learning is obviously the key [45]. In the brain, it has been reported that separation of synapses utilized for different tasks are essential for sequential learning [46], which inspired the algorithms to protect the important weights involved in previously learned tasks while training the network for new ones [31, 33]. However, these "frozen" weights necessarily reduce the degree of freedom of the system, i.e., decreasing the volume of parameter space to search for a configuration that can satisfy both the old and new tasks. We showed here that the OWM is a promising solution for such a problem. By allowing those "frozen" weights to be adjustable again without erasing acquired knowledge, OWM exhibited clear advantages in performance. It awaits further studies to investigate whether algorithms similar to OWM is implemented in the brain. It was recently suggested that a variant of the back-propagation algorithm named "conceptor-aided back-prop" (CAB) can be used for continual learning by shielding gradients against degradation of previously learned tasks [30]. By providing more effective shield of gradients through constructing an orthogonal projector, OWM achieved much better protection to previously acquired knowledge, yielding highly competitive results in empirical tests compared to CAB (see supplementary text and Figs. S3, S4, S5 for details). OWM and other methods for continual learning mentioned above belong to the category of regularization approach [45]. Similar to other methods within this category, there is a tradeoff between the performance of the old and new tasks for OWM, due to limited sources to consolidate the knowledge of previous tasks. In contrast to the regularization approach, the other type of methods for continual learning involves dynamically introducing extra neurons or layers along the learning process [47], which would be helpful to mitigate the tradeoff described above [45]. However, the regularization approach needs no extra resources to accommodate newly acquired knowledge during the training and, therefore, is capable of producing compact yet versatile systems.

Another class of biologically inspired approach for continual learning is based on the complementary learning systems (CLS) theory [48, 49]. Such systems involve the interplay between two sub-systems similar to the mammalian hippocampus and neocortex, respectively, i.e., the task-solving network (neocortex) is accompanied by a generative network (hippocampus) to keep the memories of previous tasks [50]. Often with the aid of Learning without Forgetting (LwF) method [51], data for the old tasks sampled by the generative module are interleaved with the ones for the current task to train the neural network in order to avoid the catastrophic forgetting problem. However, here we used a completely different approach for continual learning, i.e., separating the training of different tasks by OWM. As a result, the PFC-like module is mainly introduced to achieve context dependent processing, and is not critical for the continual learning in our approach, except that it introduces larger capacity for the network as a whole to learn different tasks. However, the framework of CLS might also be instrumental for further development of our approach. Currently the rotation of the feature space occurring in the PFC-like module is carried out in a fixed and arbitrary manner. It is conceivable that an encoder network can be introduced to map the contextual cues, e.g., different environments, to corresponding rotation signals. This way, the encoder can be taught to recognize and classify more complex contexts. Actually, we think such a flexible module for processing contextual signals would be analogous to the hippocampus in the brain, as the real hippocampus is indeed related to the classification of different environmental cues through the processes of pattern separation and pattern completion [49]. Thus, it awaits future studies to investigate if the framework similar to CLS can be used for achieving flexible and more sophisticated context dependent processing.

Taken together, we demonstrated that it is possible to teach a highly compact network many context-dependent mappings sequentially. Although we demonstrated its effectiveness here with the supervised learning paradigm, it has the potential to be applied to other training frameworks. Another method for overcoming catastrophic forgetting that belongs to the regularization approach, i.e., the EWC, has been successfully implemented in reinforcement learning [31]. As the EWC can be viewed as a special case of OWM in some circumstances (see supplementary text for details), it suggests that similar procedure could be extended for using OWM and PFC-like module in unsupervised conditions, thereby enabling networks to learn different mapping rules for different contexts. We expect such an approach, combined with effective methods of knowledge transfer, e.g., [52–55], may eventually lead to systems that have more flexibility and can learn to work in complex and dynamically changing situations.

METHODS

Orthogonal Weights Modification (OWM). Consider a feed-forward network of $L + 1$ layers, indexed by $l = 0, 1, \cdots, L$ with $l = 0$ and $l = L$ being the input and output layer, respectively. All hidden layers share the same activation function $g(\bullet)$. $\mathbf{W}_l$ represents the connections between the $(l - 1)$th and the lth layer with $\mathbf{W}_l \in \mathbb{R}^{s \times m}$. $\mathbf{x}_l$ and $\mathbf{y}_l$ denote the output and input of the lth layer, respectively, where $\mathbf{x}_l = g(\mathbf{y}_l)$ and $\mathbf{y}_l = \mathbf{W}_l^T \mathbf{x}_{l-1}$. $\mathbf{x}_{l-1} \in \mathbb{R}^s$ and $\mathbf{y}_l \in \mathbb{R}^m$.

In OWM, the orthogonal projector $\mathbf{P}_l$ defined on the input space of layer l for learned tasks is the key to overcome catastrophic interference in sequential learning. In general, the projector can be defined as $\mathbf{P}_l = \mathbf{I}_l - \mathbf{A}_l(\mathbf{A}_l^T\mathbf{A}_l + \alpha\mathbf{I}_l)^{-1}\mathbf{A}_l^T$ [24, 25]. Matrix $\mathbf{A}_l$ consists of all trained input vectors spanning the input space of previous tasks for the lth layer as its columns, i.e., $\mathbf{A}_l = [\mathbf{x}_l(1), \cdots, \mathbf{x}_l(n)]$. $\alpha\mathbf{I}_l$ is a unit matrix multiplying with a relatively small constant α for avoiding the ill-conditioning problem in the matrix-inverse operation. In practice, $\mathbf{P}_l$ can be recursively updated for each task by using the method equivalent to calculate the correlation-inverse matrix $\mathbf{P}^{(RLS)} = (\sum_{i=1}^n \mathbf{x}(i)\mathbf{x}^T(i) + \alpha\mathbf{I})^{-1}$ in the recursive least square (RLS) algorithm [24, 26, 27]. This method allows $\mathbf{P}_l$ to be determined based on the current inputs and the $\mathbf{P}_l$ for the last task. It also avoids the matrix-inverse operation in the original definition of $\mathbf{P}_l$.

Below we provide the detailed procedure for the implementation of OWM method.

- a. Initialization of parameters: randomly initialize $\mathbf{W}_l(0)$ and set $\mathbf{P}_l(0) = \mathbf{I}_l/\alpha$ for $l = 1, \cdots, L$.
- b. Forward propagate the inputs of the ith batch in the jth task, then back propagate the errors and calculate weight modifications $\Delta\mathbf{W}_l^{BP}(i, j)$ for $\mathbf{W}_l(i - 1, j)$ by the standard BP method.
- c. Update the weight matrix in each layer by

$$
\begin{aligned}
\mathbf{W}_l(i, j) &= \mathbf{W}_l(i - 1, j) + \kappa(i, j)\Delta\mathbf{W}_l^{BP}(i, j) \quad \text{if} \quad j = 1 \\
\mathbf{W}_l(i, j) &= \mathbf{W}_l(i - 1, j) + \kappa(i, j)\mathbf{P}_l(j - 1)\Delta\mathbf{W}_l^{BP}(i, j) \quad \text{if} \quad j = 2, 3, \cdots
\end{aligned}
\tag{1}
$$

  where $\kappa(i, j)$ is the predefined learning rate.
- d. Repeat steps from (b) to (c) for the next batch.
- e. If the jth task is accomplished, forward propagate the mean of the inputs for each batch ($i = 1, \cdots, n_j$) in the jth task successively. In the end, update $\mathbf{P}_l$ for $\mathbf{W}_l$ as $\mathbf{P}_l(j) = \mathbf{P}_l(n_j, j)$, where $\mathbf{P}_l(j) = \mathbf{P}_l(n_j, j)$ can be calculated iteratively according to:

$$
\begin{aligned}
\mathbf{P}_l(i, j) &= \mathbf{P}_l(i - 1, j) - \mathbf{k}_l(i, j)\bar{\mathbf{x}}_{l-1}(i, j)^T\mathbf{P}_l(i - 1, j) \\
\mathbf{k}_l(i, j) &= \mathbf{P}_l(i - 1, j)\bar{\mathbf{x}}_{l-1}(i, j)/[1 + \bar{\mathbf{x}}_{l-1}(i, j)^T\mathbf{P}_l(i - 1, j)\bar{\mathbf{x}}_{l-1}(i, j)]
\end{aligned}
\tag{2}
$$

  in which $\bar{\mathbf{x}}_{l-1}(i)$ is the output of the $l - 1^{th}$ layer in response to the mean of the inputs in the ith batch of the jth task, and $\mathbf{P}_l(0, j) = \mathbf{P}_l(j - 1)$.
- f. Repeat steps from (b) to (e) for the next task.

We note that the algorithm achieved the same performance if the orthogonal projector $\mathbf{P}_l$ was updated for each batch according to Eq.2. This method can be understood as treating each batch as a different task. It avoids the extra storage space as well as data-reloading in (d) and, therefore, significantly accelerates the processing. In this case, if the learning rate is set to $\kappa(i) = 1/[1 + \bar{\mathbf{x}}_{l-1}(i)^T\mathbf{P}_l(i-1)\bar{\mathbf{x}}_{l-1}(i)]$, the procedure

is equivalent to use RLS to train neural networks under the name of Enhanced Back Propagation (EBP), which was proposed to increase the speed of convergence in training [27]. Therefore, our algorithm has the same computational complexity as EBP- $O(N_n N_w^2)$, where N_n is the total number of neurons and N_w is the number of input weights per neuron [27].

For interested readers, below we illustrate how the projector we constructed in OWM is equivalent to the $\mathbf{P}^{(RLS)} = (\sum_{i=1}^{n} \mathbf{x}(i)\mathbf{x}^T(i) + \alpha \mathbf{I})^{-1}$ used in RLS, in the case that $\sum_{i=1}^{n} \mathbf{x}(i)\mathbf{x}^T(i) + \alpha \mathbf{I}$ is invertible. $\mathbf{P}^{(RLS)} = (\sum_{i=1}^{n} \mathbf{x}(i)\mathbf{x}^T(i) + \alpha \mathbf{I})^{-1}$ is the inversion of correlation matrix $\mathbf{\Phi}$ of the input signals, i.e., $\mathbf{P}^{RLS}(n) = \mathbf{\Phi}^{-1}(n)$, where

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \lambda^{-1} \mathbf{x}(i)\mathbf{x}(i)^T + \alpha \lambda^n \mathbf{I} \tag{3}$$

Assume $\lambda = 1$ and let $\mathbf{A}(n) = [\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(n)]$, where x(i) is a vector recording the ith input, $\mathbf{\Phi}$ can also be written as

$$\mathbf{\Phi}(n) = \mathbf{A}(n)\mathbf{A}^T(n) + \alpha \mathbf{I} \tag{4}$$

According to Woodbury matrix identity

$$\begin{aligned}
\mathbf{P}^{RLS}(n) &= \alpha^{-1}\mathbf{I} - \alpha^{-1}\mathbf{A}(\mathbf{I} + \alpha^{-1}\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\alpha^{-1} \\
&= \alpha^{-1}[\mathbf{I} - \mathbf{A}(\alpha\mathbf{I} + \mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T]
\end{aligned} \tag{5}$$

Clearly, the projector we constructed in OWM, $\mathbf{P}_l$, is equivalent to $\mathbf{P}^{RLS}$ if $\mathbf{P}_l$ is defined on the input space.

In addition, we provide an analysis regarding the capacity of OWM, i.e., how many different tasks can be learned by using the method. The capacity of one layer of network can be measured by the rank of $\mathbf{P}^{(owm)}$. We define that $\mathbf{P}_i^{(owm)}$ is the orthogonal projector calculated after the task i, and $\Delta\mathbf{P}_{i+1}^{(owm)}$ is the update in the next task satisfying $\mathbf{P}_{i+1}^{(owm)} = \mathbf{P}_i^{(owm)} - \Delta\mathbf{P}_{i+1}^{(owm)}$. Since $range(\mathbf{P}_i^{(owm)}) \cap range(\Delta\mathbf{P}_{i+1}^{(owm)}) = \varnothing$, $rank(\mathbf{P}_{i+1}^{(owm)}) = rank(\mathbf{P}_i^{(owm)}) - rank(\Delta\mathbf{P}_{i+1}^{(owm)})$. When $\alpha = 0$, the capacity is directly related to the rank of the matrix $\mathbf{A}_l$, which is consisted of the input vector $\mathbf{x}$ of all learned tasks as its columns. As the continual learning process goes on, $rank(\mathbf{A}_l)$ will reach its limits—the row rank of $\mathbf{A}_l$, indicating that this particular layer runs out the capacity to learn new tasks. The capacity of the whole networks can be roughly approximated by the summation of capacity of each layer $\sum_{l=1}^{L} rank(\mathbf{A}_l)$. If the capacity limit of the entire network is finally approached, two solutions can be considered: 1) to introduce a larger α or the forgetting factor as used in RLS [24] and online EWC [55]; 2) to add more layer(s), e.g., the PFC-like module (see below for details), to provide more space to preserve previously learned knowledge.

The role of the PFC-like module in context-dependent processing. In context-dependent learning, in order to change the representation of the sensory inputs without distorting the information contents in different contexts, we added one layer of neurons after the input layer (cf. Fig. 3A), which was inspired by the function and structure of primate PFC. Below we describe, from a mathematic point of view, how this PFC-like layer works, using the face classification task as an example.

In this task, the PFC-like layer was fed with feature vectors for different faces, $\mathbf{F} = [f_1 f_2 \cdots, f_k]^T \in \mathbb{R}^k$, modulated by contextual signals, $\mathbf{C} = [c_1, c_2, ..., c_m]^T \in \mathbb{R}^m$, and then generated the outputs $\mathbf{Y}^{out} = [y_1, y_2, ..., y_m]^T \in \mathbb{R}^m$ for further processing. The input weight matrix $\mathbf{W}^{in} = [\mathbf{w}_1^{in}, \mathbf{w}_2^{in}, .., \mathbf{w}_m^{in}] \in \mathbb{R}^{k \times m}$

for the PFC-like layer was randomly initialized and fixed across all contexts. Each column of $\mathbf{W}^{in}$ was normalized with $\left\|\mathbf{w}_i^{\text{in}}\right\| = 1, i = 1, 2, ..., \text{m}$. The output weight matrix $\mathbf{W}^{\text{out}} = [w_1, w_2, ..., w_m]^{\text{T}} \in \mathbb{R}^m$ was trained by the OWM method. Let net_i (cf. Fig.3A) represents the input of the i th neuron in this layer, i.e., $net_i = \left(\mathbf{w}_i^{\text{in}}\right)^{\text{T}}\mathbf{F} = \sum_{j=1}^k f_j w_{ji}^{\text{in}}$, and y_i indicates the corresponding output, then $y_i = c_i\,\text{g}(net_i)$ with $\text{g} = \max(0, x)$. Different vectors $\mathbf{C}$, representing contextual information, were generated randomly from the uniform distribution within (0,1) for each task. The function of this PFC-analogous layer can then be summarized as

$$
\begin{aligned}
\mathbf{Y}^{\text{out}} &= \text{g}\left(\left(\mathbf{W}^{\text{in}}\right)^{\text{T}}\mathbf{F}\right) \odot \mathbf{C} \\
&= \text{g}\left(\begin{bmatrix}\mathbf{w}_1^{\text{in}} & \mathbf{w}_2^{\text{in}} & \cdots & \mathbf{w}_m^{\text{in}}\end{bmatrix}^{\text{T}}\mathbf{F}\right) \odot \mathbf{C} \\
&= \text{g}\left(\begin{bmatrix}c_1\left\|\mathbf{F}\right\|\left\|\mathbf{w}_1^{\text{in}}\right\|cos\theta_1, & c_2\left\|\mathbf{F}\right\|\left\|\mathbf{w}_2^{\text{in}}\right\|cos\theta_2, & \cdots, & c_m\left\|\mathbf{F}\right\|\left\|\mathbf{w}_m^{\text{in}}\right\|cos\theta_m\end{bmatrix}^{\text{T}}\right) \quad (6)\\
&= \text{g}\left(\begin{bmatrix}c_1\left\|\mathbf{w}_1^{\text{in}}\right\|cos\theta_1, & c_2\left\|\mathbf{w}_2^{\text{in}}\right\|cos\theta_2, & \cdots, & c_m\left\|\mathbf{w}_m^{\text{in}}\right\|cos\theta_m\end{bmatrix}^{\text{T}}\right)\left\|\mathbf{F}\right\| \\
&= \text{g}\left(\begin{bmatrix}c_1 cos\theta_1, & c_2 cos\theta_2, & \cdots, & c_m cos\theta_m\end{bmatrix}^{\text{T}}\right)\left\|\mathbf{F}\right\|
\end{aligned}
$$

where $\odot$ represents element-wise multiplication and θ_i is the angle between $\mathbf{w}_i^{\text{in}}$ and $\mathbf{F}$. Note that for any $\upsilon \geq 0$, $\text{g}(\upsilon x) = \max(0, \upsilon x) = \upsilon\max(0, x) = \upsilon\,\text{g}(x)$.

For individual faces, given the same feature vector $\mathbf{F}$ and fixed $\mathbf{W}^{in}$, $\cos\theta_i$ and $\mathbf{F}$ are constants. Thus, the output $\mathbf{Y}^{\text{out}}$ is affected by the contextual input $\mathbf{C}$, which is different across tasks. If we normalize $\mathbf{C}$ by $\sqrt{\sum_{i=1}^m \left(c_i\,\text{g}\left(cos\theta_i\right)\right)^2}$, it is apparent from Eq.6 that the PFC-like layer "rotates" the input vector in the feature space, as illustrated in Fig. 3B. That explains why this added layer can change the representation of sensory inputs while keeping the information contents unchanged. Importantly, it also enables the system to sequentially learn different tasks with OWM for identical inputs.

Datasets. The MNIST [56] database contains handwritten digits from 0 to 9 collected by the National Institute of Standards and Technology (NIST). MNIST has a training set of 60,000 samples and a test set of 10000 samples. Each sample is a grey scale picture, with the size of 28×28.

The ILSVR2012 [57] is a subset of the ImageNet, which is the world's largest image recognition database [58]. There are in total 1,000 categories of images to be classified. The training dataset contains 1.2 million images. The validation dataset contains 50,000 images belonging to the same 1000 categories. The classification accuracies for this task was calculated based on the validation set.

The offline Chinese handwriting database CASIA-HWDB [59] were collected by the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. The dataset consists of isolated handwritten Chinese characters. Here we used one subset of CASIA-HWDB1.1, which has more than one million samples written by 300 writers. It contains 3755 commonly used Chinese characters. Each class has 240 training images and 60 testing images.

Large-scale CelebFaces Attributes (CelebA) [37] contains 202599 celebrity face images of 10177 identities, covering a wide range of attitude and background clutter. Each of the images has 40 binary attributes annotated (see Fig.3C or Table S3 for all attributes).

Shuffled MNIST Experiment. Shuffled MNIST experiment [23, 30–33] usually consists a number of sequential tasks. All tasks are classifying handwritten digit from 0 to 9. However, for each new task, the

pixels in the image are randomly shuffled, with the same randomization across all digits in the same task and different randomization across tasks. For this experiment, we trained 3- or 4- layer, feed-forward networks with [784,800,10] (3-layer) or [784-800/2000-800/2000-10] (4-layer) neurons (see Table S1 for details) to minimize cross entropy loss by OWM method. Rectified Linear Unit (ReLU) activation function [60] was used in the hidden layer.

Table S1 shows the performance of OWM method for the shuffled MNIST tasks in comparison with other continual learning algorithms. The accuracy of OWM method was measured by repeating the experiments for 10 times. The results for other algorithms are adopted from corresponding publications. The size of the network, regarding the number of layers and number of neurons in each layer, was chosen to be the same as in previous publications for a fair comparison.

Two sided t test was used to compare the performance between OWM and other continual learning methods for both the Shuffled and Disjoint (see below) MNIST Experiments. t values were calculated according to the mean and standard deviations across ten experiments. The mean and standard deviations for method other than OWM were adopted from corresponding publications. The significant level was chosen as $p < 0.01$. The results are shown in Table S1.

Disjoint MNIST Experiment. In the disjoint MNIST experiment [61], the original MNIST data set was divided into two parts: The first part contained the digits from 0 to 4 and the second part consisted of the remaining digits from 5 to 9. Correspondingly, the first task of the network was to recognize digits among 0, 1, 2, 3 and 4 and the second task was to recognize digits among 5, 6, 7, 8 and 9. Again, to facilitate comparison, the network size and architecture was chosen to be the same as in previous work [61]. The performance was calculated based on ten repeated experiments and was shown in Table S2.

Sequential learning of classification tasks with ImageNet and Chinese characters. The classification tasks with ImageNet and Chinese handwritten characters are more challenging due to the complex structure in each image and more classes to "memorize" in a sequential learning task. For these tasks, we first trained a DNN as the feature extractor on the whole or partial data set to extract features of each image. Then, the extracted feature vectors were fed into a 2-layer classifier with [Dimension of Feature, Number of Classes] neurons. The classifier was trained to recognize each of the classes sequentially by OWM method. The results are shown in Table 1 in the main text. We note that in these experiments, as well as in other tests mentioned in the above sections, no negative samples were used for training the network to recognize a new class. In other words, only the positive samples of a particular class were presented to the network during the training.

Context-dependent Face Recognition with CelebA. In this experiment, we first trained a feature extractor with the architecture of ResNet50 [34] on the whole training data set, using conventional multi-task training procedure. Then, the outputs of the feature extractor were fed into the PFC-analogous layer, which also received the contextual information C (cf. Fig. 3A in the main text). The size of the PFC-analogous layer was [2048-5000-1]. As explained above, the function of this PFC-like layer is to rotate the feature space. In principle, these rotated feature vectors can be further processed by downstream networks. For the face classification task examined in the present study, they were directly fed into the classifier by weights $\mathbf{W}^{\text{out}}$, which were subjected to OWM during sequential learning. Specifically, the output of networks in this task was determined as:

$$\mathbf{Y}^{\text{Lable}} = \left(\mathbf{W}^{\text{out}}\right)^{\text{T}}\mathbf{Y}^{\text{out}} = \|\mathbf{W}^{\text{out}}\| \|\mathbf{F}\| \cos\phi \,, \phi = \arccos\left(\frac{\sum_{j=0}^{n} w_j c_j \, \mathrm{g}(cos\theta_j)}{\|\mathbf{W}^{\text{out}}\|}\right) \tag{7}$$

(see the 2nd section in Methods for the definition of symbols). When the training for all contexts was completed, $\mathbf{W}^{out}$ was fixed. Therefore, for the same face, the changes in $\mathbf{Y}^{Lable}$ was due to different $cos\phi$, which in turn was determined by the contextual input $\mathbf{C}$. Before training, all weights and bias were randomly initialized. Weights in the output layer were modified by the OWM method. The detailed results of classifying individual attributes are listed in Table S3.

Network parameters. Weights in layers of the classification module were initialized by the method suggested previously [62] except that the output layer were all initialized to be zero. The bias of each layer were randomly initialized according to a uniform distribution within (0, 0.1). Rectified Linear Unit (ReLU) neurons were applied to every hidden layer in all experiments. The momentum in all optimization algorithms was chosen to be 0.9. The details of hyperparameters used for feature extractors are shown in Table S4. Early stopping was used for training both the feature extractors and classifiers. The hyperparameters for OWM method are shown in Table S5. For tasks with MNIST and CelebA, the classifier was trained to minimize cross entropy loss, while for tasks with ImageNet and Chinese characters, the classifier was trained by to minimize mean squared loss.

Mixed Selectivity analysis. In the experiments of classifying different facial attributes, responses of neurons in the PFC-analogous layer were analyzed to examine if they exhibited mixed selectivity similar to that of real PFC neurons. To this end, we chose two attributes-Attractiveness (Task 1) and Smile (Task 2). Both of them has about 50% positive and negative samples in the whole data set, and the correlation between these two attributes was low. The responses of each neuron in the PFC-analogous layer to different inputs as well as context signals were analyzed. There were in total 19962 test pictures, of which about 90% are correctly classified after training for both tasks. The threshold of excitation for each neuron was chosen as the average activity level across all neurons during the processing of all correctly-classified pictures. In Fig.S2 we show the selectivity of three exemplar neurons. According to the criteria usually used in electrophysiological experiments, these three neurons belonged to different categories, such as task-sensitive (Neuron 1), attribute-sensitive (Neuron 2). Importantly, Neuron 3 exhibited complex selectivity towards combinations of task and sensory attributes, as well as combinations between different attributes. Such mixed selectivity was commonly reported for real PFC neurons [63].

REFERENCES

[1] Allen Newell. *Unified theories of cognition*. Harvard University Press, 1994.

[2] G. A. Miller, G. A. Heise, and W. Lichten. The intelligibility of speech as a function of the context of the test materials. *Journal of Experimental Psychology*, 41(5):329–335, 1951.

[3] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception .1. an account of basic findings. *Psychological Review*, 88(5):375–407, 1981.

[4] R. Desimone and J. Duncan. Neural mechanisms of selective visual-attention. *Annual Review of Neuroscience*, 18:193–222, 1995.

[5] Pascal Fries. Neuronal gamma-band synchronization as a fundamental process in cortical computation. *Annual Review of Neuroscience*, 32:209–224, 2009.

[6] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–+, 2013.

[7] Markus Siegel, Timothy J. Buschman, and Earl K. Miller. Cortical information flow during flexible sensorimotor decisions. *Science*, 348(6241):1352–1355, 2015.

[8] Joaquin Fuster. *The prefrontal cortex*. Academic Press, 2015.

[9] Richard E Passingham and Steven P Wise. *The neurobiology of the prefrontal cortex: anatomy, evolution, and the origin of insight*. Oxford University Press, 2012.

[10] E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24:167–202, 2001.

[11] E. K. Miller. The prefrontal cortex: Complex neural properties for complex behavior. *Neuron*, 22(1):15–17, 1999.

[12] S. P. Wise, E. A. Murray, and C. R. Gerfen. The frontal cortex basal ganglia system in primates. *Critical Reviews in Neurobiology*, 10(3-4):317–356, 1996.

[13] RE Passingham. The frontal lobes and voluntary action. oxford psychology series. 1993.

[14] Earl K The prefontral cortex and cognitive control. 1(1):59, 2000.

[15] C. M. Macleod. Half a century of research on the stroop effect - an integrative review. *Psychological Bulletin*, 109(2):163–203, 1991.

[16] R. Dias, T. W. Robbins, and A. C. Roberts. Primate analogue of the wisconsin card sorting test: Effects of excitotoxic lesions of the prefrontal cortex in the marmoset. *Behavioral Neuroscience*, 110(5):872–886, 1996.

[17] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[18] Marcus Rohrbach, Michael Stark, Bernt Schiele, and Ieee. *Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting*, pages 1641–1648. IEEE Conference on Computer Vision and Pattern Recognition. 2011.

[19] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, Bernt Schiele, and Ieee. *Evaluation of Output Embeddings for Fine-Grained Image Classification*, pages 2927–2936. IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[20] Eleni Triantafillou, Hugo Larochelle, Jake Snell, Josh Tenenbaum, Kevin Jordan Swersky, Mengye Ren, Richard Zemel, and Sachin Ravi. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[21] Michael McCloskey and Neal J Cohen. *Catastrophic interference in connectionist networks: The sequential learning problem*, volume 24, pages 109–165. Elsevier, 1989.

[22] R. Ratcliff. Connectionist models of recognition memory - constraints imposed by learning and forgetting functions. *Psychological Review*, 97(2):285–308, 1990.

[23] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[24] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2008.

[25] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[26] Sharad Singhal and Lance Wu. Training feed-forward networks with the extended kalman algorithm. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 1187–1190. IEEE, 1989.

[27] S. Shah, F. Palmieri, and M. Datum. Optimal filtering algorithms for fast learning in feedforward neural networks. *Neural Networks*, 5(5):779–787, 1992.

[28] David Sussillo and L. F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[29] Herbert Controlling recurrent neural networks by conceptors. 2014.

[30] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018.

[31] James Kirkpatricka, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 2017.

[32] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017.

[33] Friedemann Zenke, Ben Poole, and Surya Continual learning through synaptic intelligence. 2017.

[34] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, and Ieee. *Deep Residual Learning for Image Recognition*, pages 770–778. IEEE Conference on Computer Vision and Pattern Recognition. Ieee, New York, 2016.

[35] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Chinese handwriting recognition contest 2010. In *Pattern Recognition (CCPR), 2010 Chinese Conference on*, pages 1–5. IEEE, 2010.

[36] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1464–1470. IEEE, 2013.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, Xiaoou Tang, and Ieee. *Deep Learning Face Attributes in the Wild*, pages 3730–3738. IEEE International Conference on Computer Vision. 2015.

[38] Sidney R. Lehky, Roozbeh Kiani, Hossein Esteky, and Keiji Tanaka. Dimensionality of object representations in monkey inferotemporal cortex. *Neural Computation*, 26(10):2135–2162, 2014.

[39] D. J. Freedman, M. Riesenhuber, T. Poggio, and E. K. Miller. Categorical representation of visual stimuli in the primate prefrontal cortex. *Science*, 291(5502):312–316, 2001.

[40] C. P. Hung, G. Kreiman, T. Poggio, and J. J. DiCarlo. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310(5749):863–866, 2005.

[41] Dwight J. Kravitz, Kadharbatcha S. Saleem, Chris I. Baker, Leslie G. Ungerleider, and Mortimer Mishkin. The ventral visual pathway: an expanded neural framework for the processing of object quality. *Trends in Cognitive Sciences*, 17(1):26–49, 2013.

[42] Jesse Gomez, Michael A. Barnett, Vaidehi Natu, Aviv Mezer, Nicola Palomero-Gallagher, Kevin S. Weiner, Katrin Amunts, Karl Zilles, and Kalanit Grill-Spector. Microstructural proliferation in human cortex is coupled with the development of face processing. *Science*, 355(6320):68–+, 2017.

[43] F. Xu and J. B. Tenenbaum. Word learning as bayesian inference. *Psychological Review*, 114(2):245–272, 2007.

[44] Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

[45] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan arXiv preprint arXiv:.07569 Wermter. Continual lifelong learning with neural networks: A review. 2018.

[46] Joseph Cichon and Wen-Biao Gan. Branch-specific dendritic ca2+ spikes cause persistent synaptic plasticity. *Nature*, 520(7546):180–U80, 2015.

[47] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia preprint arXiv:.04671 Hadsell. Progressive neural networks. 2016.

[48] J. L. McClelland, B. L. McNaughton, and R. C. Oreilly. Why there are complementary learning-systems in the hippocampus and neocortex - insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

[49] Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. 20(7):512–534, 2016.

[50] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[51] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

[52] Marcus Rohrbach, Michael Stark, Gyorgy Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where -and why? semantic relatedness for knowledge transfer. 2010.

[53] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[54] Geoffrey Hinton, Oriol Vinyals, and Jeff Distilling the knowledge in a neural network. 2015.

[55] Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabskabarwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370*, 2018.

[56] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[58] Deng J, Berg A, Satheesh S, Khosla A Su H, and Fei-Fei L. Ilsvrc-2012, 2012.

[59] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 37–41. IEEE, 2011.

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

[61] Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. Compete to compute. In *Advances in neural information processing systems*, pages 2310–2318, 2013.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[63] Araceli Ramirez-Cardenas and Pooja Viswanathan. The role of prefrontal mixed selectivity in cognitive control. *Journal of Neuroscience*, 36(35):9013–9015, 2016.