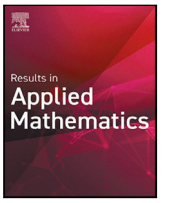
Results in Applied Mathematics 20 (2023) 100409

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics



A simple illustration of interleaved learning using Kalman filter for linear least squares

Majnu John [a,b,*], Yihren Wu [c]

[a] Departments of Mathematics and of Psychiatry, Hofstra University, Hempstead, NY, USA
[b] Feinstein Institutes of Medical Research, Northwell Health System, Manhasset, NY, USA
[c] Department of Mathematics, Hofstra University, Hempstead, NY, USA

ARTICLE INFO

Keywords:
Interleaved learning
Kalman filter
Linear least squares

ABSTRACT

Interleaved learning in machine learning algorithms is a biologically inspired training method with promising results. In this short note, we illustrate the interleaving mechanism via a simple statistical and optimization framework based on Kalman Filter for Linear Least Squares.

1. Introduction

Interleaved learning (IL) is a type of biological learning phenomenon observed in brain regions such as neocortex, and has inspired machine learning algorithms. IL is one of the mechanisms expounded by Complementary Learning Systems Theory [1,2] on how successful learners such as human beings mitigate effects of 'catastrophic interference' while learning. Recent illustrations of IL using neural networks include [3], who exhibited that if the new information is similar to a subset of old items, then deep neural networks can learn the new information rapidly and with the same level of accuracy by interleaving the old items in the subset. A similar insight was presented in [4], where it was shown that for artificial neural networks, information consistent with prior knowledge can sometimes be integrated very quickly. Another recent paper [5] formulated interleaved machine learning as a multi-level optimization problem, and developed an efficient differentiable algorithm to solve the interleaving learning problem with application to neural architecture search. A closely related biological concept is interleaved replay which also has been empirically validated in the literature [6,7]. Over the past couple of decades, ideas inspired by biological IL have been utilized in a wide array of online learning methods as well, especially to prevent catastrophic forgetting. See, for example Wang et al. [8], a comprehensive and recently updated survey on continual, lifelong learning.

All the applications and illustrations of IL to machine learning so far have used complex models such as neural networks. In this short paper, we aim to present a simple illustration of IL by adapting a framework from traditional optimization and statistics literature, namely, the Kalman-Filter (KF) approach for linear least squares (LLS). Understanding IL in this relatively straightforward framework may help in future with proving theoretical convergence properties, and then with hopefully extending to similar results for more complex models and gradient descent type algorithms. To the best of our knowledge, an illustration of IL based on Kalman filter for linear least squares, has not yet appeared in the literature (e.g., no mention of such an approach in the comprehensive survey by Wang et al. [8]).

* Correspondence to: 350 Community Drive, Manhasset, NY 11030, USA.
E-mail addresses: mjohn5@northwell.edu, majnu.john@hofstra.edu (M. John).

https://doi.org/10.1016/j.rinam.2023.100409
Received 21 September 2023; Accepted 3 November 2023
Available online 8 November 2023

2. Kalman filter for linear least squares (KF4LLS)

To better understand the concepts and notation later on, we briefly review KF4LLS by closely following the exposition provided in section 3.2 in [9]. Note that although KF is widely considered as an estimation method associated with dynamical systems, the one employed for linear least squares is a specialized case where the states of the underlying dynamical system stays constant [10].

Consider fitting a linear model to the set of input–output pairs $(\mathbf{y}_1, \mathbf{X}_1), \ldots, (\mathbf{y}_m, \mathbf{X}_m)$. Here $\mathbf{y}_i \in \mathbb{R}^{n_i}$, $\mathbf{X}_i$ is an $n_i \times q$ matrix and each $(\mathbf{y}_i, \mathbf{X}_i)$ is a given data block. Model fitting corresponds to minimizing the following quadratic cost function

$$C(\mathbf{r}) = \sum_{i=1}^{m} \|\mathbf{y}_i - \mathbf{X}_i \mathbf{r}\|^2, \quad \text{for} \quad \mathbf{r} \in \mathbb{R}^q.$$

KF4LLS is an incremental version of Gauss–Newton method, which cycles through the data blocks. Specifically, the solution given by KF4LLS to the above minimization program is $\boldsymbol{\psi}_m$ which can be obtained recursively by the algorithm

$$\boldsymbol{\psi}_i = \boldsymbol{\psi}_{i-1} + \mathbf{H}_i^{-1} \mathbf{X}_i^t (\mathbf{y}_i - \mathbf{X}_i \boldsymbol{\psi}_{i-1}), \quad \mathbf{H}_i = \mathbf{H}_{i-1} + \mathbf{X}_i^t \mathbf{X}_i, \quad i = 1, \ldots, m,$$

where $\boldsymbol{\psi}_0$ is an arbitrary vector and $\mathbf{H}_0 = \mathbf{0}$. We assume that $\mathbf{X}_1^t \mathbf{X}_1$ is positive definite and that makes all $\mathbf{H}_i$'s (except $\mathbf{H}_0$) positive definite as well. KF4LSS has been well-studied in the literature (see, for example, papers citing [10]). A derivation of the algorithm is given in section 3.2 in [9] and convergence analysis is presented in [10].

3. Interleaving KF4LLS

Consider data blocks from two different 'populations'

$$(\mathbf{b}_1, \mathbf{U}_1), \ldots, (\mathbf{b}_m, \mathbf{U}_m) \quad \text{and} \quad (\mathbf{f}_1, \mathbf{V}_1), \ldots, (\mathbf{f}_m, \mathbf{V}_m).$$

To fix ideas, it may help to think in terms of an example provided in McClelland, McNaughton and O'Reilly, 1995 [1], where the two populations are birds and fish. In our notation, we may think of columns of $\mathbf{U}_i$'s and $\mathbf{V}_i$'s as features related to birds and fish, respectively, and similarly $\mathbf{b}_i$'s and $\mathbf{f}_i$'s the corresponding target variables. In previous papers that mentioned this example, the target variables were class variables but in this paper, for convenience and simplicity, we focus on continuously distributed target variables.

We also consider a third population which is a 'mixture' of the first two populations in terms of data characteristics. In our running example, the third population will be penguins. In terms of features, we think of penguins as an admixture of birds and fish — they have wings like a bird and they can swim like a fish! We denote the data blocks from the penguin population as

$$(\mathbf{p}_1, \mathbf{Z}_1), \ldots, (\mathbf{p}_m, \mathbf{Z}_m).$$

For all populations, we assume the relationship between the corresponding target variables and features data to be linear models:

$$\mathbf{b}_i = \mathbf{U}_i \mathbf{r}_b + \varepsilon_b, \quad \mathbf{f}_i = \mathbf{V}_i \mathbf{r}_f + \varepsilon_f, \quad \mathbf{p}_i = \mathbf{Z}_i \mathbf{r}_p + \varepsilon_p, \quad i = 1, \ldots, m,$$

where $\varepsilon_b, \varepsilon_f$ and ε_p are *i.i.d.* mean zero error variables, and for convenience, we assume $m = 2k$ (*i.e.*, an even number). We are interested in knowing whether we can train a model by interleaving data blocks from birds and fish, and then use the trained model to predict target variables related to penguins (*i.e.* the $\mathbf{p}_i$'s) using features data from penguins (*i.e.* the $\mathbf{Z}_i$'s). That is, the goal is to train a model via interleaving, using only data from birds and fish, but then test the model using only features and target variables from penguins.

In this short note, we assume

$$\mathbf{Z}_i = \alpha \mathbf{U}_i + (1 - \alpha) \mathbf{V}_i \tag{1}$$

and

$$\mathbf{r}_p = \alpha \mathbf{r}_b + (1 - \alpha) \mathbf{r}_f, \text{ for some } \alpha \in [0, 1]. \tag{2}$$

In words, each feature matrix $\mathbf{Z}_i$ of penguins is a weighted average of $\mathbf{U}_i$ and $\mathbf{V}_i$, the corresponding feature matrices of birds and fish. Similarly, the weight-parameters in the model for penguins, $\mathbf{r}_p$, connecting the features to the target variable is a weighted average of the weight-parameters in the models for birds and fish. If instead of (1), we assumed the distribution of $\mathbf{Z}_i$'s to be a mixture of distributions of $\mathbf{U}_i$'s and $\mathbf{V}_i$'s we observed similar results as the ones presented later in this short note, but to focus the presentation we will just work with the assumption made in (1). In this case, our interleaved algorithm is as follows.

***Interleaved KF4LLS algorithm*:**

***Step 0(a)*:** Center all data blocks, including the target variables, individually by subtracting the corresponding column means. Thus, in the following step, all $\mathbf{b}_i$'s, $\mathbf{f}_i$'s, and all columns of $\mathbf{U}_i$'s and $\mathbf{V}_i$'s are mean-zero vectors.

***Step 0(b)*:** Set $\mathbf{H}_0^{(\alpha)} = \mathbf{0}$ and

$$\mathbf{U}_i^{(\alpha)} = \sqrt{\alpha}\, \mathbf{U}_i, \ \mathbf{b}_i^{(\alpha)} = \sqrt{\alpha}\, \mathbf{b}_i; \quad \mathbf{V}_i^{(\alpha)} = \sqrt{1-\alpha}\, \mathbf{V}_i, \ \mathbf{f}_i^{(\alpha)} = \sqrt{1-\alpha}\, \mathbf{f}_i, \ i = 1, \ldots, m.$$

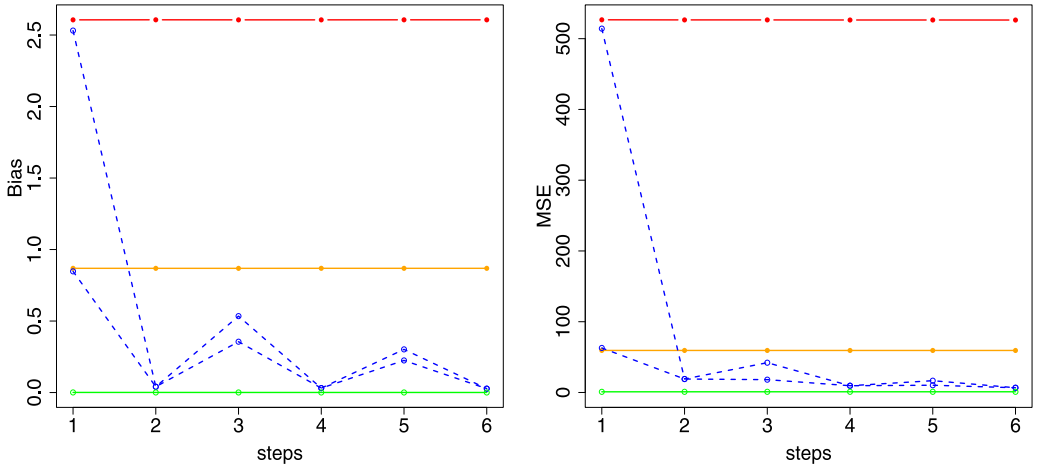


Fig. 1. Bias and MSE, averaged across 5000 simulation-iterations, of interleaved KF4LLS algorithm applied on synthetic data. In all scenarios, MSE was calculated as the prediction error when the trained models where applied on 'penguin' test data. Red, orange and green lines correspond to training based on birds, fish and penguin data blocks, respectively, without interleaving. The blue lines correspond to training based on interleaving algorithm, either starting with a bird data block or with a fish data block. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Step 1: $\mathbf{H}_1^{(\alpha)} = \mathbf{H}_0^{(\alpha)} + (\mathbf{U}_1^{(\alpha)})^t(\mathbf{U}_1^{(\alpha)});\ \boldsymbol{\psi}_1 = \boldsymbol{\psi}_0 + (\mathbf{H}_1^{(\alpha)})^{-1}(\mathbf{U}_1^{(\alpha)})^t(\mathbf{b}_1^{(\alpha)} - \mathbf{U}_1^{(\alpha)}\boldsymbol{\psi}_0)$.

Step 2: $\mathbf{H}_2^{(\alpha)} = \mathbf{H}_1^{(\alpha)} + (\mathbf{V}_1^{(\alpha)})^t(\mathbf{V}_1^{(\alpha)});\ \boldsymbol{\psi}_2 = \boldsymbol{\psi}_1 + (\mathbf{H}_2^{(\alpha)})^{-1}(\mathbf{V}_1^{(\alpha)})^t(\mathbf{f}_1^{(\alpha)} - \mathbf{V}_1^{(\alpha)}\boldsymbol{\psi}_1)$.

Step 3: $\mathbf{H}_3^{(\alpha)} = \mathbf{H}_2^{(\alpha)} + (\mathbf{U}_2^{(\alpha)})^t(\mathbf{U}_2^{(\alpha)});\ \boldsymbol{\psi}_3 = \boldsymbol{\psi}_2 + (\mathbf{H}_3^{(\alpha)})^{-1}(\mathbf{U}_2^{(\alpha)})^t(\mathbf{b}_2^{(\alpha)} - \mathbf{U}_2^{(\alpha)}\boldsymbol{\psi}_2)$.

Step 4: $\mathbf{H}_4^{(\alpha)} = \mathbf{H}_3^{(\alpha)} + (\mathbf{V}_2^{(\alpha)})^t(\mathbf{V}_2^{(\alpha)});\ \boldsymbol{\psi}_4 = \boldsymbol{\psi}_3 + (\mathbf{H}_4^{(\alpha)})^{-1}(\mathbf{V}_2^{(\alpha)})^t(\mathbf{f}_2^{(\alpha)} - \mathbf{V}_2^{(\alpha)}\boldsymbol{\psi}_3)$.

... etc. ...

Step (m-1): $\mathbf{H}_{m-1}^{(\alpha)} = \mathbf{H}_{m-2}^{(\alpha)} + (\mathbf{U}_k^{(\alpha)})^t(\mathbf{U}_k^{(\alpha)});\ \boldsymbol{\psi}_{m-1} = \boldsymbol{\psi}_{m-2} + (\mathbf{H}_{m-1}^{(\alpha)})^{-1}(\mathbf{U}_k^{(\alpha)})^t(\mathbf{b}_k^{(\alpha)} - \mathbf{U}_k^{(\alpha)}\boldsymbol{\psi}_{m-2})$, where $k = m/2$.

Step m: $\mathbf{H}_m^{(\alpha)} = \mathbf{H}_{m-1}^{(\alpha)} + (\mathbf{V}_k^{(\alpha)})^t(\mathbf{V}_k^{(\alpha)});\ \boldsymbol{\psi}_m = \boldsymbol{\psi}_{m-1} + (\mathbf{H}_m^{(\alpha)})^{-1}(\mathbf{V}_k^{(\alpha)})^t(\mathbf{f}_k^{(\alpha)} - \mathbf{V}_k^{(\alpha)}\boldsymbol{\psi}_{m-1})$.

The algorithm alternates between using $(\mathbf{b}_i, \mathbf{U}_i)$'s (i.e. birds data blocks) in odd-numbered steps and $(\mathbf{f}_i, \mathbf{V}_i)$'s (i.e. fish data blocks) in even-numbered steps making it a proper interleaved training approach. Note that the algorithm is an oracle algorithm because it can be implemented only if the mixing coefficient α is known. Typically this is possible only for simulated synthetic data. Thus, the above algorithm in its current form serves only for illustrating IL and not for any practical applications. For real data, if assumption (1) truly holds then α can be estimated. It is more likely that for real data there will be a separate mixing coefficient for each column of $\mathbf{Z}_i$; such separate coefficients can also be estimated, for example, using a grid search on the unit interval.

Illustration with synthetic data

We illustrate the algorithm on synthetic data generated as follows. We set $\alpha = 0.25$, $n_i = n = 100$, $q = 6$ and $m = 6$ (i.e. $k = 3$). Performance of the algorithm was assessed by calculating the bias and the mean-squared error (MSE) based on the estimates $\boldsymbol{\psi}_i$'s after each step of the algorithm. Average bias and MSE over 5000 simulation-iterations were plotted (see Fig. 1). Elements in $\mathbf{r}_b$ and $\mathbf{r}_f$ were generated separately from $Uniform(-5, 5)$ distribution, and then fixed for all simulation iterations. For each simulation-iteration, each row of $\mathbf{U}_i$ was generated independently from a multivariate normal - $N(\boldsymbol{\mu}_1, I_{6\times6})$, and similarly each row of $\mathbf{V}_i$ was generated from $N(\boldsymbol{\mu}_2, I_{6\times6})$ where $\boldsymbol{\mu}_1 = [\mu_1, \ldots, \mu_1]^t$ and $\boldsymbol{\mu}_2 = [\mu_2, \ldots, \mu_2]^t$. Here μ_1, μ_2 were generated separately from a $Uniform(-10, 10)$ distribution, and then fixed for all the simulation-iterations. Bias plotted below was averaged across all simulation-iterations, but within each simulation-iteration it was also averaged across elements of the parameter vector. Codes used for this example with detailed comments are posted in the following GitHub page (https://github.com/mjohn5/InterleavedKF4LLS/)

The red line in Fig. 1 corresponds to the scenario where only the $(\mathbf{b}_i, \mathbf{U}_i)$ blocks were used for training, and the orange line corresponds to the scenario where only the $(\mathbf{f}_i, \mathbf{V}_i)$ blocks were used for training. Since all the testing was done on $(\mathbf{p}_i, \mathbf{Z}_i)$ blocks, it is not surprising to see that the scenarios corresponding to the red and orange lines show substantial bias and MSE for all steps. The green line corresponds to the scenario at the other extreme where training and testing were both done on the penguin data (i.e. $(\mathbf{p}_i, \mathbf{Z}_i)$). Again, it is not surprising to see that the bias and MSE for this scenario is very close to zero. Blue lines correspond to the scenario with Interleaved KF4LLS algorithm used for training, and as in all other cases, testing done on 'penguin' blocks. There are two blue lines in each panel, one starting with $(\mathbf{b}_1, \mathbf{U}_1)$ and the other starting with $(\mathbf{f}_1, \mathbf{V}_1)$; in both cases the algorithm alternates between the 'birds' and 'fish' data blocks. It is easy to see from the figure that, similar to the biological interleaved learning phenomenon, interleaving the training in this simple least squares setting leads to almost nil bias and MSE. The reduction in bias

and MSE achieved by the Interleaved KF4LLS algorithm in a few steps is almost the same as that achieved by the algorithm that is trained exclusively with 'penguin' data. With this synthetic data example, it is also observed that the Interleaved KF4LLS algorithm achieves almost zero bias in just two steps, a phenomenon that has some theoretical justification (see below).

Some theoretical justification

Let $\mathcal{F}_{2j}$ denote the 'history of the algorithm' up to and including the $2j^{th}$ step, $j = 1, \ldots, k$. That is, $\mathcal{F}_{2j}$ is the sigma-field generated by $\mathbf{b}_1, \mathbf{U}_1, \ldots, \mathbf{b}_j, \mathbf{U}_j$; $\mathbf{f}_1, \mathbf{V}_1, \ldots, \mathbf{f}_j, \mathbf{V}_j$. Then the following lemmas show that even with two steps the estimator obtained by the algorithm (i.e. $\boldsymbol{\psi}_2$) is a good approximation to the unknown parameter-vector that we are trying to estimate, namely, $\mathbf{r}_p$. Thus, the following theory closely mirrors the result that we saw with synthetic data above.

Lemma 1.

$$\boldsymbol{\psi}_2 = (\mathbf{H}_2^{(\alpha)})^{-1}[(\mathbf{U}_1^{(\alpha)})\mathbf{b}_1 + (\mathbf{V}_1^{(\alpha)})\mathbf{f}_1]. \tag{3}$$

Hence,

$$\mathbb{E}(\boldsymbol{\psi}_2/\mathcal{F}_2) = \left[\alpha(\mathbf{U}_1^t\mathbf{U}_1) + (1-\alpha)(\mathbf{V}_1^t\mathbf{V}_1)\right]^{-1}\left[\alpha(\mathbf{U}_1^t\mathbf{U}_1)\mathbf{r}_b + (1-\alpha)(\mathbf{V}_1^t\mathbf{V}_1)\mathbf{r}_f\right]. \tag{4}$$

Proof of Lemma 1. Adding

$$\mathbf{H}_1^{(\alpha)}\boldsymbol{\psi}_1 = \mathbf{H}_1^{(\alpha)}\boldsymbol{\psi}_0 + (\mathbf{U}_1^{(\alpha)})^t\mathbf{b}_1^{(\alpha)} - (\mathbf{U}_1^{(\alpha)})^t\mathbf{U}_1^{(\alpha)}\boldsymbol{\psi}_0$$

and

$$\mathbf{H}_2^{(\alpha)}\boldsymbol{\psi}_2 = \mathbf{H}_2^{(\alpha)}\boldsymbol{\psi}_1 + (\mathbf{V}_1^{(\alpha)})^t\mathbf{f}_1^{(\alpha)} - (\mathbf{V}_1^{(\alpha)})^t\mathbf{V}_1^{(\alpha)}\boldsymbol{\psi}_1$$

and cancelling terms, we get Eq. (3). Eq. (4) follows from Eq. (3) since $\mathbb{E}(\mathbf{b}_1/\mathcal{F}_2) = \mathbf{U}_1\mathbf{r}_b$ and $\mathbb{E}(\mathbf{f}_1/\mathcal{F}_2) = \mathbf{V}_1\mathbf{r}_f$.

Also, as a side remark, the symmetry in the result above explains why it is irrelevant whether we start with $(\mathbf{b}_1, \mathbf{U}_1)$ or with $(\mathbf{f}_1, \mathbf{V}_1)$ as seen in the synthetic data example. The following lemma states that up to a first order approximation based on Taylor series expansion, $\boldsymbol{\psi}_2$ calculated in step-2 of the Interleaving KF4LLS algorithm is an unbiased estimator of $\mathbf{r}_p$, if the columns of $\mathbf{U}_1$ (and similarly columns of $\mathbf{V}_1$) are (respectively) pairwise uncorrelated and with constant standard deviation.

Lemma 2. *If*

$$n^{-1}\mathbb{E}(\mathbf{U}_1^t\mathbf{U}_1) = n^{-1}\mathbb{E}(\mathbf{V}_1^t\mathbf{V}_1) = \sigma I_{n\times n}, \tag{5}$$

then up to a first order approximation

$$\mathbb{E}(\boldsymbol{\psi}_2) \approx \alpha\mathbf{r}_b + (1-\alpha)\mathbf{r}_f = \mathbf{r}_p. \tag{6}$$

Proof of Lemma 2. It is well-known that using a first-order Taylor series approximation, the expectation of a ratio is approximately the ratio of the expectation. Taking expectations in Eq. (4), applying the above-mentioned fact and using Eq. (5) we get Eq. (6).

4. Conclusions and future directions

Interleaved learning is a learning technique observed in human brain areas such as neocortex which helps with long-term retention and in general better learning. Inspired by this biological phenomenon, machine learning algorithms have tried to incorporate interleaving while training models, especially complex neural network models. In this short note, we presented a simple statistical framework based on linear least squares to better understand computational interleaving learning. Our simple framework based on linear least squares can probably be extended to logistic regression models or any generalized linear models and support vector machines as well, which we intend to pursue as future work. A framework like the one presented in this short note will also help with better understanding the convergence properties of interleaving algorithms. Future work will include stating and proving such theoretical properties as well.

Declaration of competing interest

None.

Data availability

No data was used for the research described in the article.

Acknowledgments

We thank Rajat Saxena and Prof. Bruce McNaughton for sharing with us an early version of their review paper on a biological model for forward transfer in continual learning. The following grant funding for MJ is acknowledged: Wellcome Trust grant number 226716/Z/22/Z (PIs: Malhotra, Voineskos), NIMH grant numbers R01MH117646 (PI: Lencz), R01MH120594 (PIs: Kane, Robinson), R01MH120313 (PI: Deligiannidis), R01MH117172 (PI: DeChoudhury), RF1MH122886 (PI: Barber) and R61/R33MH123574 (PIs: Cornblatt, Carrion).

References

[1] McClelland JL, McNaughton BL, O'Reilly RC. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Psychol Rev 1995;102:419–57.
[2] Marr D. Simple memory: A theory for archicortex. Philos Trans R Soc Lond B Biol Sci 1971;262:23–81.
[3] Saxena R, Shobe JL, McNaughton BL. Learning in deep neural networks and brains with similarity-weighted interleaved learning. Proc Natl Acad Sci U S A 2022;119(27):e2115229119. http://dx.doi.org/10.1073/pnas.2115229119.
[4] McClelland JL, McNaughton BL, Lampinen AK. Integration of new information in memory: New insights from a complementary learning systems perspective. Philos Trans R Soc Lond B Biol Sci 2020;375:20190637.
[5] Ban H, Xie P. Interleaving learning, with application to neural architecture search. 2021, arXiv preprint arXiv:2103.07018.
[6] Gepperth A, Karaoguz C. A bio-inspired incremental learning architecture for applied perceptual problems. Cognit Comput 2016;8:924–34.
[7] Kemker R, Kanan C. FearNet: Brain-inspired model for incremental learning. ICLR poster 2018. 2017, arXiv:1711.10563.
[8] Wang L, Zhang X, Su H, Zhu J. A comprehensive survey of continual learning: Theory, method and application. 2023, arXiv preprint. arXiv:2302.00487.
[9] Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Bellmont, MA: Athena Scientific; 1996, 4th printing.
[10] Bertsekas DP. Incremental least squares methods and the extended Kalman filter. SIAM J Optim 1996;6(3):807–22.