Continual Deep Learning by Functional Regularisation of Memorable Past

Pingbo Pan,[1,*,†] Siddharth Swaroop,[2,*] Alexander Immer,[3,†] Runa Eschenhagen,[3,†] Richard E. Turner,[2] Mohammad Emtiyaz Khan[5]

[1] University of Technology Sydney, Australia; [2] University of Cambridge, UK; [3] École Polytechnique Fédérale de Lausanne, Switzerland; [4] University of Tübingen, Germany; [5] RIKEN Center for AI Project, Tokyo, Japan
[†] This work is conducted during an internship at RIKEN Center for AI project, Tokyo, Japan
Corresponding authors: ss2163@cam.ac.uk, emtiyaz.khan@riken.jp

Summary

- Intelligent systems need to adapt quickly to changing environments
- In Continual Learning, all past data can not be observed in the future
- Standard training methods lead to catastrophic forgetting of the past
- Weight-regularisation methods improve this, but are not enough
- We propose a function-regularisation method called **Functional Regularisation of Memorable Past** (FROMP)
- The key idea is to
  - convert neural networks to Gaussian Processes,
  - find a few crucial examples to avoid forgetting of the past (memorable examples),
  - train a new network while regularising over memorable examples to avoid forgetting.

Weight-space vs function-space

Weight-space methods find important weights for past tasks, and keep new weights close to them:

$$N\bar{\ell}_t(\mathbf{w}) + \delta(\mathbf{w}-\mathbf{w}_{t-1})^\top \mathbf{F}_{t-1}(\mathbf{w}-\mathbf{w}_{t-1})$$

where $\bar{\ell}_t(\mathbf{w}) := (1/N)\Sigma_{i=1}^N \ell(\mathbf{y}_i, \mathbf{f}_w(\mathbf{x}_i))$ is the loss over datapoints on current task t, δ is regularisation strength, $\mathbf{w}_{t-1}$ is previous task weights, and $\mathbf{F}_{t-1}$ is a **preconditioning matrix** that favours weights relevant to past tasks more than the rest [1, 2, 3].

Making current weights closer to the previous ones does not always ensure that the **predictions** on the past tasks also remain unchanged.

Better approach is to **directly regularise neural network outputs** $\mathbf{f}_w$. For example, we can use l_2-regularisation [4],

$$N\bar{\ell}_t(\mathbf{w}) + \delta \sum_{s=1}^{t-1} (\mathbf{f}_{t,s} - \mathbf{f}_{t-1,s})^\top (\mathbf{f}_{t,s} - \mathbf{f}_{t-1,s})$$

where $\mathbf{f}_{t,s}$ and $\mathbf{f}_{t-1,s}$ are vectors of function values using the current network and the previous task network, over **all datapoints from all previous tasks**.

This is over all past datapoints, and therefore **computationally expensive**. Previous attempts have used inducing points [5] and "working memory" [4].

FROMP: Functional Regularisation of Memorable Past

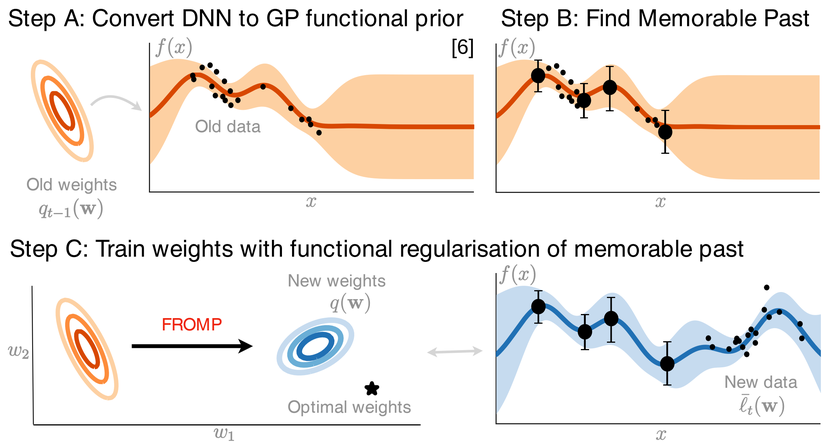


Figure 1: FROMP has three steps. In Step A, we convert the previously trained network (orange ellipses) to a Gaussian process (top left) which is then used as a "functional prior" to regularise the next task. In Step B, we choose a few memorable past examples (orange circles) that are crucial to avoid forgetting. In Step C, we train a new network over the next task (black dots in the bottom row) while making sure that the predictions over memorable past examples remain unchanged.

Mathematical details

We start with a Bayesian formulation of continual learning (ELBO) [1]:

$$\mathbb{E}_{q(w)}\left[(N/\tau)\bar{\ell}_t(\mathbf{w}) + \log q(\mathbf{w})\right] - \underbrace{\mathbb{E}_{q(w)}\left[\log q_{t-1}(\mathbf{w})\right]}_{\approx \mathbb{E}_{\tilde{q}_w(\mathbf{f})}\left[\log \tilde{q}_{w_{t-1}}(\mathbf{f})\right]}$$

We replace the final weight-space regularisation term with a function-space regulariser. This is over memorable points only.

After some approximations (see paper), we arrive at FROMP loss function:

Network outputs pushed towards previous mean

$$N\bar{\ell}_t(\mathbf{w}) + \frac{1}{2}\tau \sum_{s=1}^{t-1} \left[\mathbf{m}_{t,s}(\mathbf{w}) - \mathbf{m}_{t-1,s}\right]^\top \mathbf{K}_{t-1,s}^{-1} \left[\mathbf{m}_{t,s}(\mathbf{w}) - \mathbf{m}_{t-1,s}\right]$$

Sum over past tasks' memorable points

Kernel automatically weights examples

Experiments

See paper for further results on Split MNIST and Permuted MNIST

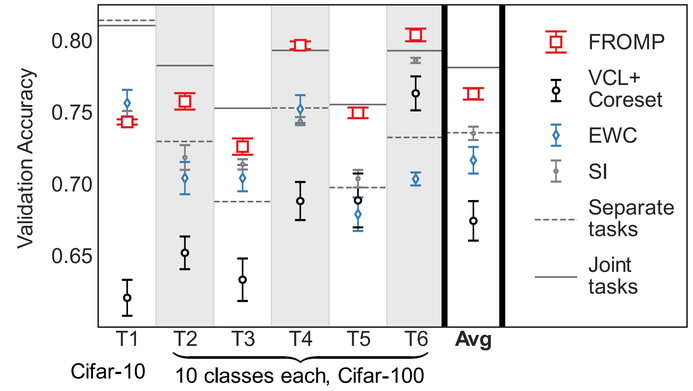
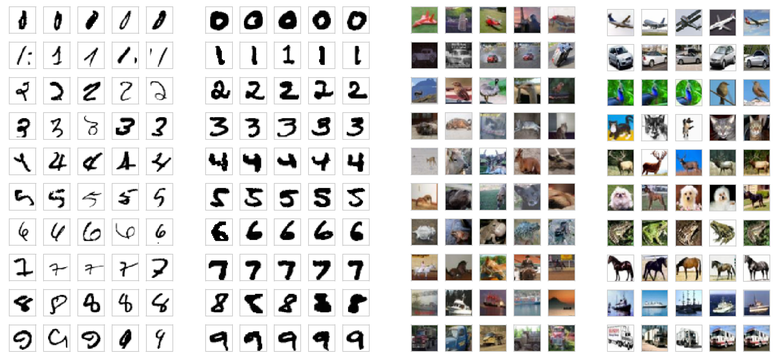


Figure 2: FROMP outperforms baselines on Split CIFAR (see 'Avg' column). 'Separate tasks': different networks are trained on each task separately. 'Joint tasks': a single network is trained jointly on all task data (upper-bound to continual learning performance).



(a) Most memorable (b) Least memorable (a) Most memorable (b) Least memorable

Figure 3: Most memorable and least memorable datapoints on MNIST (left) and CIFAR-10 (right). Memorable points are difficult to classify and lie on the decision boundary. Additionally, our method for choosing memorable points (Step B in FROMP) is computationally cheap.

References

[1] Nguyen et al., "Variational Continual Learning", ICLR 2018.
[2] Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks", PNAS 2017.
[3] Zenke et al., "Continual learning through synaptic intelligence", ICML 2017.
[4] Benjamin et al., "Measuring and regularizing networks in function space", ICLR 2019.
[5] Titsias et al., "Functional regularisation for continual learning using Gaussian processes", ICLR 2020.
[6] Khan et al., "Approximate Inference Turns Deep Networks into Gaussian Processes", NeurIPS 2019.