Automatically inferring task context for continual learning

Jasmine Collins (jazzie@berkeley.edu)
Redwood Center for Theoretical Neuroscience, BAIR
University of California, Berkeley

Kelvin Xu (kelvinxu@berkeley.edu)
BAIR
University of California, Berkeley

Bruno Olshausen (baolshausen@berkeley.edu)
Redwood Center for Theoretical Neuroscience, BAIR
University of California, Berkeley

Brian Cheung (bcheung@berkeley.edu)
Redwood Center for Theoretical Neuroscience, BAIR
University of California, Berkeley

Abstract

While neural network research typically focuses on models that learn to perform well within the context of a single task, models that operate in the real world are often required to learn multiple tasks or tasks that change under different contexts. Furthermore, in the real world the learning signal for each of these tasks usually arrives in sequence, rather than simultaneously in a batch, as in the deep learning setting. We propose a method to infer when the task context has changed when learning from a continual datastream, and to adjust the model's learning accordingly to prevent interference between learned tasks. We show how to automatically infer the context of a previously learned task for use in the future (e.g. during model evaluation). These preliminary results show that learning autonomously in a continually changing environment is possible in neural network models. This learning is better suited to how data naturally arrives in a real world environment.

Keywords: continual learning; neural networks

Introduction

Data naturally arrives as an ordered stream. Events in the future depend on events in the past. Furthermore, this dependence can exist at multiple timescales simultaneously. For example, a camera can provide a persistent video stream which reveals changes both from day to night and from summer to winter. This perspective of data as a stream (i.e datastream) is in stark contrast to how learning currently operates in neural network models. During training, data is artificially shuffled and arrives as an unordered set (i.e dataset). This destructive operation is done to satisfy the statistical assumption that the data is independent and identically distributed (i.i.d.). However, this lack of ordering also assumes the information of interest only exists within the timescale of individual elements in the dataset. Consequently, this dramatically increases the difficulty of learning larger timescale dependencies, and in the continual learning setting can lead to a problem termed *catastrophic forgetting*, which refers to the tendency of neural networks to forget an old task suddenly when presented with data from a new task.

In order to relax the constraints imposed by the i.i.d. assumption, we propose a modification to neural network learning that accounts for the distribution shift that occurs in a datastream. By treating model parameters more like memories rather than static variables after training, we can significantly reduce the problem of catastrophic forgetting.

Instead of distilling the information of a datastream into a single set of model parameters, we propose storing the non-stationary knowledge as distinct memories which can potentially be consolidated in the future. Our method avoids prematurely consolidating temporal information as it arrives which can hinder learning information in the future (Schaul, Borsa, Modayil, & Pascanu, 2019).

The importance of algorithms that can learn from data arriving in their natural order is especially important in the problem of continual learning. Much of the previous work in continual learning relies on the assumption that datastreams are broken into discrete 'tasks', and furthermore, that the points of transition between tasks are known. Datastreams in the real world often consist of more gradual, continuous changes, and the transitions between tasks, or contexts, are not given, but must be inferred. In neurobiology, the basal ganglia is believed to modulate task-switching (A. G. Collins & Frank, 2013). Inspired by this ability of humans to effortlessly infer task changes without explicit supervision, we relax the assumption that an oracle provides information about context changes and instead infer them automatically from the data.

Related work

Inferring tasks automatically

While most work in continual learning assumes that the task transitions are known, there have been a few approaches to inferring them instead. Kirkpatrick et al. (2017) use a genera-

Figure 1: Permuted MNIST task. The first task is the original MNIST classification dataset (left) and the second task (right) is generated by applying a random permuation to the pixels in each image of the dataset. Subsequent tasks are generated similarly, by applying a different random permutation.

tive model of the input which estimates the context probabilities conditioned on the current observation. This requires the learning of a new generative model for each task as it arrives. While our proposed approach achieves the same outcome, it is much simpler and the implementation is built into the model rather than requiring a separate system for task inference.

Achille et al. (2018) introduce an approach that is similar to ours, by developing an 'atypicality score' measured from a generative model of the environment to infer the context. They measure the divergence of the behavior of stochastic latent units from their prior. Our approach is different in that it is able to infer task context for classification without doing any generative modelling of the data.

Neural network superposition and standardization loss

Our own previous work (Cheung, Terekhov, Chen, Agrawal, & Olshausen, 2019) controlled when memories (parameters) are stored and retrieved by encoding knowledge of the distribution shift. Here, we propose using the *standardization loss* described by J. Collins, Ballé, and Shlens (2019) to automatically detect this distribution shift.

Method

Task

We consider the permuted MNIST task from Goodfellow, Mirza, Xiao, Courville, and Bengio (2013). We note that this is not a very realistic model of natural distribution shift (tasks transitions are very discrete and there is no overlap in input features) but nonetheless it is a commonly studied toy task in the continual learning literature (Kirkpatrick et al., 2017; Zenke, Poole, & Ganguli, 2017).

Neural network superposition

Traditionally, models have been viewed as a parameterized function fit to data. Much of the neural network literature focuses on training these parameters in a stationary setting where all data is drawn from a single distribution. In contrast, Cheung et al. (2019) treat the parameters of neural network models more akin to a memory. To handle multiple data distributions, they propose treating the network parameters much like an associative memory. Context information is used to recall distribution specific parameters from this memory. Much like the memories stored in a Hopfield network, these task specific parameters exist superposed in the memory. To implement superposition, the standard linear transformation in networks $y = Wx$ is augmented with a context C:

$$y = WCx \tag{1}$$

For computational efficiency, the context is parameterized as $C = diag(c)$ where c is a context vector.

Standardization loss

J. Collins et al. (2019) proposed a method to accelerate neural network training by using a loss-based normalization instead of explicit normalization techniques such as batch normalization (Ioffe & Szegedy, 2015).

For a given layer of activations $\boldsymbol{x}$ ($\boldsymbol{x} \in \mathbb{R}^{b \times n}$ for a fully connected network, where b refers to the batch dimension and n to the number of hidden units in that layer) before the nonlinearity, batch normalization ensures that the activations are zero mean and unit variance by normalizing activations across the batch dimension.

Rather than explicitly normalizing activations like batch normalization, J. Collins et al. (2019) propose an auxiliary loss that encourages the activations at each layer to be close to a standard Normal distribution. This is achieved minimizing the following KL-divergence:

$$L_s = D_{KL}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(\mathbf{0}, \mathbf{I})) = \frac{\mathbf{1}}{\mathbf{2}} \sum_{\mathbf{i=0}}^{\mathbf{n}} (\mu_\mathbf{i}^\mathbf{2} + \sigma_\mathbf{i}^\mathbf{2} - \mathbf{log}(\mathbf{o}_\mathbf{i}^\mathbf{2}) - \mathbf{1})$$

where μ_i and σ_i^2 are calculated the same as in batch normalization.

Batch normalization will force this KL-divergence measure to be zero for any batch of inputs. Standardization loss in comparison converts this divergence into a useful measure of how far from standard normal the current activation distribution is, which can be useful in the continual learning setting. The divergence measure between the standard normal distribution and the current activation distribution can be thought of as a measure of information gain. From this perspective, we can interpret the standard normal distribution as a prior and the current activation distribution as a posterior. The standardization loss measures how 'surprising' the current input distribution is for the model. This surprise corresponds to information that can be learned from the current data that is not already present in the model parameters. If the current distribution of data provides new information for the model, this serves as an indicator to allocate space in the model parameters to learn that information.

Training details

We train a 2-layer fully connected ReLU network with 2000 units in each layer. The network is trained with a learning rate of 0.0001 using the RMSProp optimizer. Every 2000 steps, the training task changes. For the context vector c we use binary superposition vectors as in Cheung et al. (2019).

To train our model, we minimize the following loss:

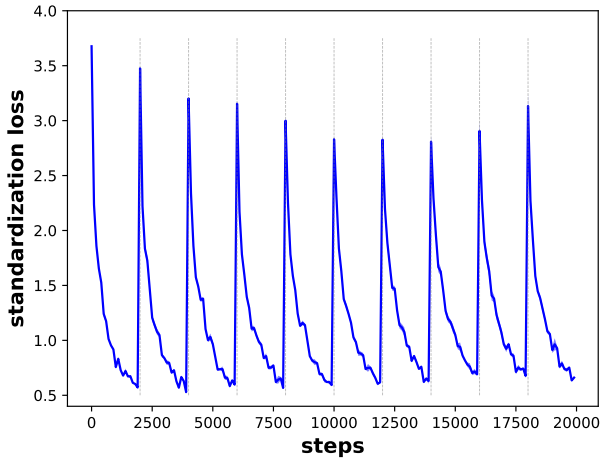

Figure 2: Networks trained with standardization loss incur a spike in the loss every time the task changes, denoted by the gray dotted line. This spike can be used as a cue for distribution shift.

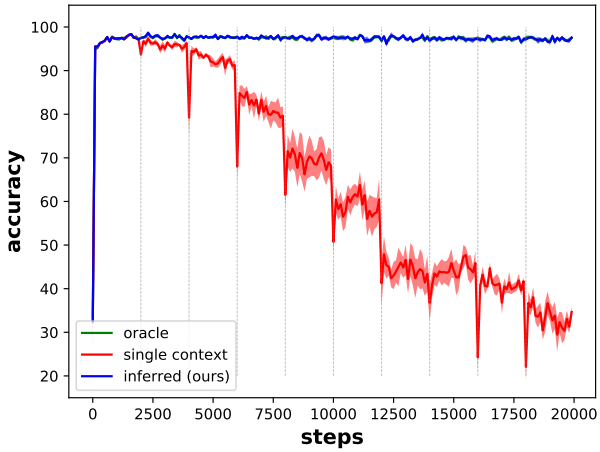

Figure 3: Accuracy on first permuted MNIST task as a function of number of training steps. Every 2000 steps, the training task changes (denoted by gray dotted line). Networks trained with ground truth knowledge about task, and therefore context changes (green, overlapped with by blue) and our method for inferring context changes (blue) are able to retain performance on the first task even as more tasks are added. Networks trained without knowledge of task information that use the same context vector across all tasks (red) quickly forget the first task. Error bars calculated across 5 runs with different random seeds.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_s$$

where $\mathcal{L}_{cls}$ is cross entropy classification loss for the permuted MNIST task and $\mathcal{L}_s$ is the standardization loss (λ = 0.001).

Standardization loss encourages activations to be zero mean unit variance over a given training dataset - thus when the data changes, standardization loss will drastically increase (Figure 2). For a given context vector, we keep track of an *upper bound* standardization loss value, that is, a scalar value that if the standardization loss exceeds, we assume that unfamiliar data is being presented. We find that setting the upper bound to be $2\times$ the minimum standardization loss value seen so far (for a given context) works well. Each context vector has a corresponding upper bound.

When selecting a new context, we calculate the standardization loss value of the current minibatch for all of the existing contexts so far, and choose the context with the minimum standardization loss that doesn't exceed that context's given upper bound. This allows us to re-use old contexts if we see old data again some time in the future. If no context vector is suitable, we use a new context vector entirely.

Results

We train on 10 permuted MNIST tasks and track the accuracy on the first training task as a way to measure catastrophic forgetting. Ideally, for a network that doesn't forget, the accuracy on the first training task should remain unaffected as more tasks are introduced. For a network without knowledge of task changes, we expect the accuracy to decrease as additional tasks are added, due to forgetting. To test this, we compare our method to a network with ground truth context information from an oracle, and a network that only uses a single context across the training of 10 different tasks. As seen in Figure 3, our method is able to correctly infer task changes just as well as if it was given oracle information, and the model that uses a single context suffers from catastrophic forgetting.

Our method is not only able to pick a new context when an unseen tasks is presented, it is also able to correctly recall old contexts used for previous training tasks that are revisited. To demonstrate this, we train on the same 10 tasks and then present the learned model with the 10 tasks again. We find that the model is perfectly able to recall the correct context vector for all tasks (Figure 4).

Discussion and future work

In this work we demonstrated that augmenting a neural network model with a simple standardization loss can allow for automatic inference of context changes. This approach requires less overhead than existing approaches which often involve entirely separate systems to infer the context. Additionally, the standardization loss was introduced as a method to accelerate training of neural networks without using explicit normalization techniques like batch normalization (J. Collins et al., 2019), so it may serve a dual purpose in addition to being useful for continual learning.

While we only considered 10 training tasks in this work, it is important to experiment more with a larger number of training tasks or tasks that may have overlap in structure, where inter-

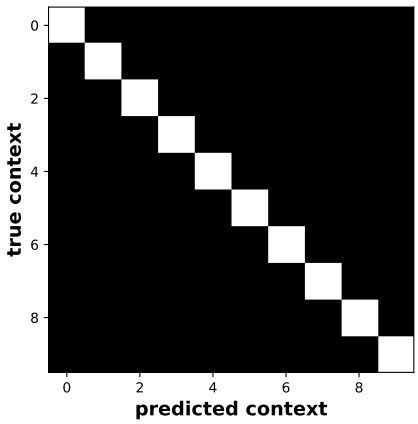

Figure 4: Our model is able to correctly infer the context for tasks it was previously trained on when they are seen again. Confusion matrix showing that for the 10 permuted MNIST tasks, our model is able to perfectly the previous contexts used for each task.

ference between tasks may play more of a role and inferring the correct context may be more difficult.

In this work, contexts are fixed to their initialized values and simply selected using standardization loss. Given that the contexts are differentiable and trainable like any other parameter in a neural network, learning to generate new contexts from previous ones is a promising future direction. New contexts which are functions of previous ones make it possible to performance knowledge transfer from the past to the future. A method to decide what knowledge should be transferred is an open research direction. One possibility is to use variational inference and optimize the standardization loss to generate new contexts.

Incorporating an episodic memory would make it easier in learning how to combine contexts in a retrospective manner. For example, if two tasks require overlapping knowledge, an episodic memory makes it possible to compare these two examples which arrived at different points in time during learning.

Here, context selection is based on the input image. But a broader notion of contexts is possible. For example, contexts can potentially denote multiple tasks which are applied to the same input image (e.g. classification and segmentation). We hope to expand how contexts are used and allow the selection process to be dependent on other sources of information beyond the input. In future work, we believe this will enable what we propose here to apply to more realistic continual learning settings like navigation in an open world.

Acknowledgments

We would like to thank Sasha Sax and members of the Redwood Center for insightful comments and discussions. This work was supported by the National Science Foundation Graduate Research Fellowship and hardware donations from NVIDIA.

References

Achille, A., Eccles, T., Matthey, L., Burgess, C., Watters, N., Lerchner, A., & Higgins, I. (2018). Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in neural information processing systems* (pp. 9873–9883).

Cheung, B., Terekhov, A., Chen, Y., Agrawal, P., & Olshausen, B. (2019). Superposition of many models into one. *arXiv preprint arXiv:1902.05522*.

Collins, A. G., & Frank, M. J. (2013). Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological review*, *120*(1), 190.

Collins, J., Ballé, J., & Shlens, J. (2019). Accelerating training of deep neural networks with a standardization loss. *CoRR*, *abs/1903.00925*. Retrieved from http://arxiv.org/abs/1903.00925

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning (icml)* (p. 448-456).

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526. Retrieved from https://www.pnas.org/content/114/13/3521 doi: 10.1073/pnas.1611835114

Schaul, T., Borsa, D., Modayil, J., & Pascanu, R. (2019). Ray interference: a source of plateaus in deep reinforcement learning. *arXiv preprint arXiv:1904.11455*.

Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 3987–3995).