Introduction to Continual Learning

Gioele Migno[1]

[1]DIAG - Sapienza University of Rome

Abstract

Continual learning is a particular category of machine learning approaches with the aim of remove all the main assumptions that currently differentiate how an artificial and a biological intelligent system learn. In this article, we will introduce the main strategies adopted in recent years and provide a method as an example for each of them.

1. Introduction

Since birth, humans are able to acquire new skills and transfer the acquired knowledge from a domain to an another. In the meanwhile, artificial intelligence systems are still in their infancy regarding what is referred to as transfer learning [1]. The human brain can do that since it has evolved mechanisms of neurosynaptic plasticity, an essential feature of the brain yielding physical changes in the neural structure and allowing it to learn and remember. However, the differences between biological and artificial systems go beyond architectural differences and also include the way in which these two systems are exposed to external stimuli. During their existences, humans can interact and explore a highly dynamic world using a multi-sensory approach obtaining a potential infinite and continuous stream of information. In contrast, current artificial neural networks are typically trained to adapt to batches of task-specific uni-sensory data collected in a controlled environment, shown in isolation and random order [1].

In order to reduce the gap between biological and artificial intelligence systems, a particular machine learning paradigm called *continual learning* has recently received increasing attention especially due to its implications in autonomous learning agents and robots [1] [2]. In the literature, to refer to continual learning several synonyms are used, such as *Incremental Learning*, *Lifelong Learning* and *Never Ending Learning* [2]. All of them refer to the same scenario where the algorithm must be capable of learning from a continuous stream of information, with such information becoming progressively available overtime and where the number of tasks to be learned are not predefined [1].

In this setting, the main assumptions made in a machine learning problem are no longer admissible. Indeed, in a continual learning problem we have: *1) Data distribution is not static, 2) Data cannot be assumed i.i.d (Independent and identically distributed).* These characteristics follow from the requirement of the model to be able to deal with different tasks or data distributions, and obtain information from an online stream rather than from an offline data batch [2]. Note that, especially during training phase, we still provide information to the model using batches of data, however they are not built specifically to represents as well as possible the data distribution but simply to mimic a streaming buffer.

When we try to use a standard deep learning approach to solve a continual learning problem, we are immediately faced with *catastrophic forgetting* phenomena also called *catastrophic interference* [2]. This phenomenon corresponds to an abrupt performance decrease for a task when the network is trained to solve a new task or the data distribution changes. This happens because the new knowledge interferes with the old one [1].

2. Formal Definitions

Despite the rapidly growing interest in continual learning, we currently lack of a common formal and generic definition of a continual learning problem. Here, we use as formal definitions of all components required, the ones described in the framework proposed by [2]. That framework is supposed to be general enough to cover all the combinations of continual learning with the classical unsupervised, supervised and reinforcement learning approaches.

Since we cannot assume data being sampled from a unique distribution D, we have first to define the concept of a sequence of distributions D_i:

Definition 2.1 (*Continual Distributions*). In Continual Learning, $\mathcal{D}$ is a potentially infinite sequence of unknown distributions $\mathcal{D} = \{D_1, ..., D_N\}$ over $X \times Y$, with X and Y input and output random variables, respectively. At time i a training set T_{r_i}, containing one or more observations, is provided by D_i to the algorithm [2].

Then, a formal definition of what we mean with the term *task*:

Definition 2.2 (*Task*). A task is a learning experience characterized by a unique task label t and its target function $g^*_{\hat{t}}(x) \equiv h^*(x, t = \hat{t})$, i.e., the objective of its learning [2].

CEUR Workshop Proceedings (CEUR-WS.org)

Notice that, there is no one-to-one correspondence between a task label t and a distribution D_i. For instance, consider a continual learning scenario in which the goal is to distinguish pictures of cats from pictures of dogs. At the beginning, we may show to the model only images of white dogs and white cats (D_1), then only pictures of black dogs and black cats D_2. This variation corresponds to a distribution shift from D_1 to D_2 that if not detected by the model, leads to catastrophic forgetting. So in this case, we have a continual distribution composed by two distributions $\mathcal{D} = \{D_1, D_2\}$ but only a task t with target function $g_t^*(x)$ informally defined as:

$$g_t^*(x) = \begin{cases} \text{"cat"} & \text{if } x \text{ is a picture of a cat} \\ \text{"dog"} & \text{if } x \text{ is a picture of a dog} \end{cases}$$

The previous example, in the framework [2], is classified as a *Single-Incremental-Task (SIT)* scenario. Other possible scenarios are: *1) Multi-Task (MT)* where we have a new task at every learning session; *2) Multi-Incremental-Task (MIT)* where we have multiple tasks, but one or more tasks occur in multiple sessions. For more details and formal definitions, please refer to the original paper.

For the rest of this article, with the term *knowledge* we refer to the pair of a task t and a static distribution D.

3. Catastrophic Forgetting

In DNNs, catastrophic forgetting occurs when the new instances to be learned differ significantly from previous ones due to distribution shift or even task switch. The performance degradation is caused by the overwriting of new knowledge over the old one. In standard learning offline scenarios, the loss of knowledge can be recovered by showing all the samples shuffled over and over [1]. This approach cannot be used when the model gets information through an infinite stream of data, and even assuming an infinite resources capability, both mnemonic and computational, a model that stores all the samples cannot be considered as continual learning [2].

Catastrophic forgetting occurs due to the *stability-plasticity dilemma.* If the network is too plastic, older memories will quickly be overwritten; however, if the network is too stable, it is unable to learn new data [3]. More specifically, the stability-plasticity dilemma is a trade-off between the precision of the information saved and the acceptable forgetting. Indeed, the model should be able to save only important information and efficaciously transfers knowledge and skills to future tasks. Since it is impossible to know what will be important for the future, the network has to deal with this trade-off [2].

To conclude, catastrophic forgetting is the main challenge of continual learning. In the next section we summarize the most common approaches used to deal with it.

4. Most Popular CL Strategies

We can characterize the most popular continual learning strategies to deal with catastrophic forgetting into four classes: *1) Regularization*, *2) Dynamic architecture*, *3) Rehearsal* and *4) Pseudo-rehearsal.*

4.1. Regularization

Regularization is a technique used to prevent overfitting in standard deep learning models. In the context of continual learning, it could be used to deal with catastrophic forgetting. Basic regularization methods such as dropout, weight norm and early stopping reduce the chance of weights being updated, and thus decrease knowledge forgetting. More complex approaches instead of reduce plasticity of all weights, search for important weights for the old knowledge and protect them by reducing their plasticity [2].

EWC

Elastic Weight Consolidation (EWC) [4] is an example of continual learning method based on an advanced regularization technique. Assume a neural network with parameters θ trained to solve a task A using data sampled from a distribution D_{A1}. Currently, the parameters are adjusted properly to the best possible values for that porpuse i.e., $\theta = \theta^*_{A,D_{A1}}$. To reduce effect of catastrophic forgetting when we learn a new task B from a distribution D_{B1}, EWC uses the following loss function:

$$\mathcal{L}(\theta) = \mathcal{L}_{B,D_{B1}} + \sum_i F_i \frac{\lambda}{2}(\theta_i - \theta^*_{A,D_{A1},i})^2$$

where $\mathcal{L}_{B,D_{B1}}$ is a generic loss function computed for the current training step, λ sets the importance of the old knowledge compared to the new one, F_i is the i-th element on the diagonal of the Fisher information matrix F and indicates the importance of the weight $\theta^*_{A,D_{A1},i}$ for the old task. In the next learning session, with a new task or a new distribution, EWC will include two penalties to preserve both of the two old knowledge.

4.2. Dynamic architecture

Catastrophic forgetting happens due to the need of change weights of the network to acquire new knowledge, a possible approach to eliminate completely this issue is to allocate a new network for each new knowledge. This method is not easy applicable in the real world due to the lack of any assumption on the number of knowledge the continual learning algorithm will face with, however catastrophic forgetting can be alleviated with limited architectural changes. As done in [2] [1], we refer to methods that use this kind of approach as *dynamic architecture techniques*. In [2], the authors divide them in two subcategories: *1) Explicit dynamics architecture* where new parameters are added, cloned or saved; *2) Implicit dynamics architecture* where we perform model adaptation without modifying its architecture. This can be done for instance, by inactivating some learning units (freeze) or by changing the forward pass path. Freezing technique is used also by biological brains, indeed there is a limited time window in development, called *critical period*, in which infants are particularly sensitive to the effects of their experiences and after this period, some neurological changes are irreversible [1].

PathNet

PathNet [5] falls into implicit dynamics architecture category. It uses a genetic algorithm to select the best weights path (*pathway*) in the network among several candidates. Once chosen the best, it froze the weights belonging to it and associate the pathway to the just learned knowledge in order to be used in the future forward steps. In this method, catastrophic forgetting is prevented by design since it virtually uses a separate network for each knowledge.

4.3. Rehearsal

One of the earliest methods used for reducing catastrophic forgetting is called *rehearsal*. Since we cannot keep all samples in memory, rehearsal approach selects and saves only few samples representative for each knowledge acquired. Catastrophic forgetting occurs when a system is trained on non-i.i.d data, mixing old samples with new samples of a new knowledge, approximates i.i.d conditions and reduce performance degradation [3].

Maintain raw and unprocessed samples ensure that the memories are not degraded through time, however this is also an important disadvantage of rehearsal methods since does not respect data privacy and could violate laws like *the right to be forgotten* of EU GDPR, or other laws referring for instance to medical records [2].

iCaRL

The method iCaRL (incremental classifier and representation learning) [6] is an example of continual learning approach based on rehearsal. It is designed to perform image classification tasks without any prior information about the number of classes to distinguish. For each class, s exemplars, properly chosen, are stored in a limited memory of size K, the storage is equally distributed among all classes and re-equilibrated when a new class is discovered in order to avoid memory saturation. To perform classification, iCaRL uses the exemplars stored, it computes the mean for each class and then compares them to the new sample, the class with the nearest mean is chosen. To do this process, the method does not use the raw exemplars rather, a more meaningful representation (embedding) given by a DNN used as features extractor. When a new class is discovered, the DNN is trained using a regularization technique to avoid catastrophic forgetting. In particular, the loss function is composed by two parts: *1) classification loss*, encourages the network to output the correct class indicator for new classes; *2) distillation loss*, it is used to avoid a drop in performance on the old raw exemplars stored in memory.

Since iCaRL uses also a regularization technique to train the embedding, in addition to rehearsal, it also falls in regularization category [2].

4.4. Pseudo-rehearsal

Instead of preserving old knowledge using raw samples, pseudo-rehearsal methods rely on a generative model trained to generate useful artificial samples to use as exemplars for each knowledge learned. This approach removes the need of store raw samples in memory reducing privacy-related issues. Pseudo-rehearsal category is also called *generative replay* or *intrinsic replay* [2].

Due to its similarity with a mechanism used in biological brain, we devote the next separate section to delving into pseudo-rehearsal methods.

5. Complementary Learning Systems

A biological brain is able to learn and memorizes new knowledge concurrently. This is done thanks to the interplay of several brain structures that work at different timescales and learning rates [1]. Neocortex is a set of neurons layers of the mammalian cerebral cortex, hippocampus instead, is a brain structure located deeper in the brain. According to complementary learning systems (CLS) theory, neocortex and hippocampus are two components at the base of learning and memory consolidation mechanisms.

The hippocampus uses a high learning rate and allows for the rapid learning of novel information (short-term memory), neocortex instead, is characterized by a slow learning rate necessary to generalize the new knowledge (long-term memory). So memories are initially stored within the hippocampus and over-time are slowly consolidated within the neocortex for permanent storage [1] [7]. This memory consolidation happens thanks to the novel information, acquired by the hippocampus, being played back over time to the neocortex [1].

CLS inspired a lot of continual learning approaches that rely on a dual models architecture. One model is easily adaptable and acts like the hippocampus learning the current new knowledge. The other one instead, is more stable imitating the neocortex to preserve old knowledge [2].

Dual models architecture are especially used to implement pseudo-rehearsal approaches in which instead of storing old raw exemplars in memory, a generative model is trained and used to generate exemplars of old knowledge to combine with samples of the new knowledge presented in the current learning session.

Depending on the type of the generative model exploited, pseudo-rehearsal methods can be categorized into two classes: *1) Marginal Replay*, where a standard generative method is used; *2) Conditional Replay* where a conditional model is used. This last allows to generate data from a specific condition like a task or a distribution, this is very useful to generate balanced dataset [2].

It is needed to highlight an important difference between a biological CLS and a pseudo-rehearsal approach. In the first, the artificial experiences (i.e. samples) refer to the new knowledge and are generated by the short-term memory system (hippocampus) and used to perform fine-tuning on the long-term memory system (neocortex) during sleep phase. In a standard pseudo-rehearsal approach instead, artificial samples refer to old knowledge and are generated by the long-term system, then raw new knowledge samples are combined with the generated ones and all of them are used to fine-tune the long-term system [1] [3].

FearNet

FearNet [3] is a conditional replay pseudo-rehearsal method. It is inspired by CLS theory but in addition to the two models for long and short-memory, it integrates also a third model used to choose which of the two models use at prediction time. The third model, called BLA, is inspired by basolateral amygdala which is responsible for regulating the brain's fear response. Neuroscientists found that this brain part shifts where to retrieve a memory from (long or short-term memory) as that memory is consolidated over time [3].

The authors designed FearNet to solve incremental class learning problems that involve sequentially learning classes in bursts of examples from the same class. More specifically, in the first learning session the model receives a batch containing more than one class in order to build its base-knowledge, in the next sessions instead, the batches contain only one class [3].

As short-term memory, FearNet uses a variant of a probabilistic neural network. To compute the class conditional probabilities it uses temporarily stored training examples collected during different learning sessions. These exemplars are deleted once performed memory consolidation, namely transferred the new knowledge from the short-term model to the long-term model. The long-term model is instead based on an autoencoder architecture that makes possible to generate artificial samples, the hidden representation is used to make predictions. In order to generate samples from a specific class, during memory consolidation phase, all the exemplars in the short-term memory are given in input to the encoder to extract a dense feature representation, and then the mean feature vector μ_c and the covariance matrix Σ_c for each class c are computed. As BLA model, a standard feedforward neural network takes as input the predictions of both long and short-term models and provides a value between 0 and 1. Instead of using solely BLA to determine which network to use, authors found that combining its output with those of the two other models improved results [3].

Memory consolidation takes place at regular interval imitating brain's sleep-wake cycle. During this phase, the long-term system generates a balanced batch of samples for the old knowledge that is then mixed with exemplars in the short-term memory and used to fine-tune the long-term model. At the end of this process, all exemplars in the short-term memory are deleted [3].

6. Challenges of CL

Despite the rapidly growing interest in continual learning with many methods proposed in the recent years, we currently lack of a common precise continual learning problem definition that specifies important points like: *1) Data Availability 2) Prior Knowledge* and *3) Memory and Computational Constraints.* Moreover, there are no benchmark datasets and metrics for proper evaluations and comparisons. The availability of common ground is essential to better advance research, reducing ambiguities and fostering fair comparisons [1] [2].

A challenge bigger than reaching a theoretical agreement, is finding stable algorithms that can learn in real world by acquiring multisensorial signals and be able to actively interact with the environment similar to how biological intelligent systems learn. Assuming a such algorithm in an embodied platform, other practical hardware challenges must be faced in the robotics field.

7. Conclusion

In this article, we introduced a machine learning paradigm called continual learning that has as goal to imitate how biological agents learn in the real world. The main challenge to address is dealing with catastrophic forgetting that affects any classical machine learning approach when it is applied to a continual learning scenarios. We then described the main four categories of proposed methods by highlighting pseudo-rehearsal type due to its similarity with memory consolidation mechanism of the brain.

References

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual Lifelong Learning with Neural Networks: A Review, Neural Networks 113 (2019) 54–71. URL: http://arxiv.org/abs/1802.07569. doi:10.1016/j.neunet.2019.01.012., arXiv:1802.07569 [cs, q-bio, stat].

[2] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, N. Díaz-Rodríguez, Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges, 2019. URL: http://arxiv.org/abs/1907.00182, arXiv:1907.00182 [cs].

[3] R. Kemker, C. Kanan, FearNet: Brain-Inspired Model for Incremental Learning, 2018. URL: http://arxiv.org/abs/1711.10563, arXiv:1711.10563 [cs].

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks 114 (????) 3521–3526. URL: http://arxiv.org/abs/1612.00796. doi:10.1073/pnas.1611835114. arXiv:1612.00796.

[5] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, D. Wierstra, PathNet: Evolution Channels Gradient Descent in Super Neural Networks, ???? URL: http://arxiv.org/abs/1701.08734. arXiv:1701.08734.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, iCaRL: Incremental Classifier and Representation Learning, ???? URL: http://arxiv.org/abs/1611.07725. arXiv:1611.07725.

[7] T. Kitamura, S. K. Ogawa, D. S. Roy, T. Okuyama, M. D. Morrissey, L. M. Smith, R. L. Redondo, S. Tonegawa, Engrams and circuits crucial for systems consolidation of a memory 356 (????) 73–78. doi:10.1126/science.aam6808. arXiv:28386011.