Seminars In AI

Topic 1: Continual Learning

Farooq Ahmad Wani

Matricola: 1946707

(Seminar delivered by prof. Vincenzo Lomonaco)

1. Abstract

In current times we have several algorithms and methods to make the machine learning models achieve the state of art performance, but the biggest challenge is the time, effort, and data it takes to achieve the state-of-the-art performance. The 3V's (volume, velocity, variability) of data makes it harder for us to train the new model from scratch whenever we have a requirement of executing the new task. The solution to this problem is coming from continual Learning, which gives machine learning models the ability to learn continuously whenever a new task is assigned. It is a modern technique where models are trained on small batches of data and can perform even in CPU-level embedded devices and thus making it possible to achieve remarkable levels of autonomy. The main problem associated with a family of these techniques is Catastrophic Forgetting. In this report, I will briefly discuss different techniques of continual learning and the associated methods it employs to deal with the problem of catastrophic forgetting.

2. Introduction

The process of continual learning is highly inspired by the learning process of humans where humans acquire new knowledge and built it on the previously learned experience. In the case of machine learning, till recent times the approach followed was stateless "where the model was trained every time on new data without considering the previous experience and state of the model". Recent times have seen a change in the trend of training models from stateless to stateful training. This stateful training is strongly motivated by the principle of reusing the learned knowledge to learn new tasks with lesser training and resources. Stateful training which is commonly known as Continual learning, multitask learning or lifelong learning builds cumulatively as opposed to learning from scratch.

The biggest challenge of the Continual Learning is not to only improve the current task performance but at the same time increase the performance of the model in the old tasks. Thus, in continual learning, the challenge is to get a fully accurate model under the constraints of the tasks arriving sequentially and having limited memory with limited computation to use.

Like any other technique, this family also comes with some advantages and challenges to tackle. Some of the main problems accompanying these tasks are catastrophic forgetting, memory constraints, and augmenting the model capacity while promoting the reuse of the learned parameters. Catastrophic forgetting means the model will forget the old task while learning new ones, something that is not expected since we always want it to be able to perform well on old tasks as well as on new tasks. Memory constraint assumes that we have limited space available to store the model parameters and the data from previous tasks. This technique needs to augment the model capacity because when new tasks come, they may have different semantic and syntactic requirements.

Multiple attempts have been made to overcome these issues and, in this paper, I will touch on some of them. The main techniques to deal with forgetting are the Latent Replay approach and the Continual Learning with Hypernetworks. There are some other methods that mainly grow from the architecture of the network and will be discussed in brief at the end of the report.

3. Current state of Art approaches

In this section, some of the state-of-art methods of continual learning dealing with the above-mentioned issues will be debated. This section is divided into four sub-sections. In the first sub-section, the methods of regularization will be reviewed. In the second sub-section, the methods of parameter hyper networks will be conferred. In the third sub-section, the methods of data reply will have conversed and in the last sub section, the methods which grow on network architecture will be discussed.

3.1 Regularization

The main idea of these methods to deal with the problem of forgetting is by identifying the model parameters which are necessary for the experience. These parameters are prevented from any updating or keeping the updating to a minimum. This process can be also called task priority using task augmentation in which we want to do a new task either without affecting the previous one or increase the performance of both the new one as well as of the old one. From the perspective of implementation, this method is implemented by adding the regularization term which will penalize the changes in the weights which are having more importance to previously learned tasks. In mathematics, this process is almost equal to the newly learned task in the null space of the learned tasks or to form the projection of the new task in already learned mapping. Let $\mathbf{\Omega^t_k}$ denotes the calculated importance of the parameter $\mathbf{\Theta_k}$ at task $\mathbf{T_t}$. The importance value attains a higher value for the important parameters and vice versa. While learning the task T+1 the objective function will be given as:

$$\tilde{\mathcal{L}}_{t+1} = \mathcal{L}_{t+1} + \lambda \sum_k \Omega_k^t \left(\theta_k - \theta_k^t\right)^2$$

$$\theta_k^{t+1} = \underset{\theta_k}{\operatorname{argmin}} \tilde{\mathcal{L}}_{t+1}$$

This surrogate associates an importance weighted penalty for deviation of parameters from their values and is in control to keep the experience intact.

There are many different implementations outside the general one. Li and Hoiem (2016) proposed learning "the forgetting" model by using CNN in which they used knowledge distillation i.e. transferring the experience from the highly regularized model to smaller models. This method in general classified the weights into three categories, the weights associated with the old task, the new task, and the shared weights. The learning of this algorithm puts the constraint on the updating of the shared weights to remember the previous experience. However, this has many drawbacks and the major one is the linear increase in training time.

Over the time, many other approaches have been given in this field which are out of scope for this report. Mentioning a few of them in brief:

- Kirkpatrick, et al. (2017) proposed the elastic weight consolidation (EWC) model in supervised and reinforcement learning scenarios.
- Zenke, Poole, et al. (2017) proposed to alleviate catastrophic forgetting by allowing individual synapses to estimate their importance for solving a learned task.
- Maltoni and Lomonaco (2018) proposed the AR1 model for single-incremental-task scenarios which combines architectural and regularization strategies.

In my view, during the recent times, the ensemble methods are performing very well in this part of research though the initial attempts showed their disadvantage linked to intense use of storage memory. But in newer approaches, the number of methods has been employed to restrict the size of the memory and made ensembles very good contender in this field.

In summary, the regularization methods have done quite well in dealing with the problem of the catastrophic forgetting but the use of different constraints on parameter learning may lead to trade off in the performance of old and new tasks.

3.2 Continual Learning with hypernetworks

In this method, the problem of Forgetting is taken to the meta-level, where the outputs of a metamodel called "task-conditioned hyper network", maps the task embeddings to weights, are fixed. This approach made it possible to save every task as a single point. In other words, in this method, we have constraints on the updating of the neural weights. From the computational perspective, this is generally modeled via additional constraints of mapping that restore any changes in the mapping function of the neural networks.

The central approach of this model is to learn the parameters of the metamodel rather than directly learning the parameters of the target model. Thus, a hyper network can be treated as a weight generator and was originally introduced to dynamically compress the model parameters.

One approach to avoid the catastrophic forgetting in this method is to store data from previous tasks and corresponding model outputs and then fix such outputs. In this way, we can somehow avoid the issue of forgetting by mixing the data from the past tasks with the new ones, this approach can be seen as a multitask learning because we are learning all the tasks simultaneously. Although this task seems to be the easy option, aligning to this approach is potentially memory expensive and not strictly online learning.

To tackle this issue, the hyper network fixes the weight space of the target network to a single point per task. This constraint can be employed by using two-step optimizations:

- **Compute the current loss:** Firstly, the candidate is changed to minimize the current task loss concerning the current network, then candidate parameter of the meta-model is obtained by minimizing the loss with the target parameter.
- **Learned task embeddings:** The task embeddings are differentiable deterministic that can be then learned from the candidate meta-model parameters.

This model is very well suited for the neural network models, but we need to keep past tasks in memory which are then used by hypernetwork models to penalize the changes in the previous output.

3.3 Latent Replay

Typically, in deep learning models, the layers which are close to input usually perform low-level feature extraction and the weights associated with these layers tend to be stable, more often these weights give good results if being reused. On the other hand, the layers close to the output layers are mainly responsible for developing the discriminatory features for the classification and thus need to be retrained for every new task. This method stores the activation volumes of the layers till some intermediate layer and then focuses the model not to change the weights or slow down the changes up until this intermediate layer. The layers which are above the intermediate

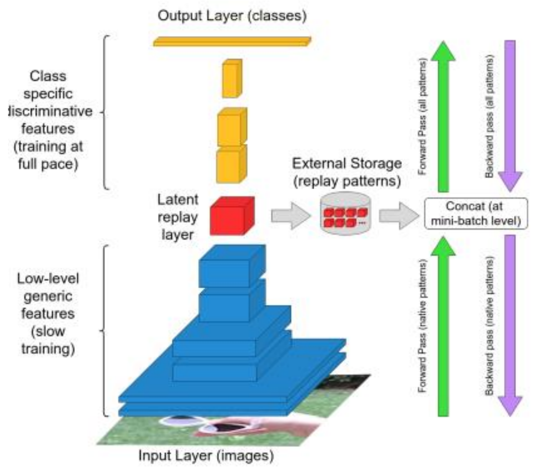
Output Layer (classes)
Class specific iscriminative features (training at full pace)
Latent replay layer
External Storage (replay patterns)
Concat (at mini-batch level)
Forward Pass (all patterns)
Backward pass (all patterns)
Low-level generic features (slow training)
Forward Pass (native patterns)
Backward pass (native patterns)
Input Layer (images)

Figure 1: Architectural diagram of Latent Replay.

layer model are allowed to change and develop the discriminatory characteristics for the new task.

Compared to the Rehearsal technique, this approach reaches better results in terms of storage and computational resources. The implication of the constraint reduces the trainable parameters which result in decreased number of computations for the new training.

In general, in cases where the representation layers are not frozen, the activation stored in external layers may be from the aging effect. However, if the training of these layers is sufficiently slow the aging effect is not disruptive. Apart from the less computation required this method has one more advantage, the ability to train in small batches.

3.4 Dynamic Architectures

These are a group of techniques that change architectural properties in response to the new tasks by dynamically reconfiguring the novel neural resources like network layers etc.

The first model in this family of techniques was given by Rusu who proposed that the learned task on the network should be blocked and any new task should be expanded by adding the sub-networks with the fixed capacity. Given "n" existing tasks if a new task comes a new sub-network is added and the lateral connections with existing tasks are learned. This approach prevents the forgetting problem but brings in lots of architectural problems.

The incremental training model for the denoising of the auto encoders has been introduced by Zhou, Sohn, and Lee. This model adds the neuron for the samples which have a higher loss and once the accuracy of the model is calculated these neurons are added to the existing structure. This adding of the new features minimizes the residual of the object function. This method merges similar features into a single representation to get the compact feature representation. This model out-performed the non-incremental models of the auto encoders but increased the model complexity and risk maximization.

Part and Lemon proposed the combination of CNN with the self-reorganizing incremental neural network. This model takes the advantages of CNN for its dexterous representation and at the same time uses the neural network structures to get adjusted for the new tasks.

In recent times many other models have been given to self-organize the architecture to accommodate the new changes but the main problem with these networks is that they are highly dependent on the order in which training data has been processed.

4. Conclusions

To state in brief, the process of lifelong learning is very important for the growth of AI and machine learning but at the same time, it is very challenging to produce global techniques and models. Although, most of the models and methods discussed above take inspiration from the human brain from a mathematical perspective, it is very challenging. These models are still far away to produce flexibility, robustness, and scalability. Additional research efforts are required to merge different methodologies and come up with some global solutions to the above-mentioned issues. Fundamentally, we should not ignore the previous work done and try to accommodate the already existing methods with proper improvements.

5. References

5.1 Tyler L. Hayes, Nathan D. Cahill, and Christopher Kanan. Memory Efficient Experience Replay for Streaming Learning. In 2019 International Conference on Robotics and Automation (ICRA), pages 9769–9776. IEEE, may 2019.

5.2 James Kirkpatrick, 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. In Proceedings of the National Academy of Sciences, volume 114, pages 3521–3526, 2017.

5.3 Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual Learning for Robotics. arXiv preprint arXiv:1907.00182, pages 1–34, 2019.

5.4 Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. Connection Science, 7(2):123–146, 1995.

5.5 Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, Davide Maltoni. Latent Replay for Real-Time Continual Learning. https://arxiv.org/abs/1912.01100v2

5.6 Johannes von Oswald, Christian Henning, Benjamin F. Grewe, João Sacramento. Continual learning with hypernetworks. https://arxiv.org/abs/1906.00695v4

5.7 Continual lifelong learning with neural networks: A review German I. Parisi , Ronald Kemker, Jose L. Part, Christopher Kanan, Stefan Wermter .

5.8 Continual Learning with Neural Networks: A Review Abhijeet Awasthi IIT Bombay awasthi@cse.iitb.ac.in Sunita Sarawagi IIT Bombay sunita@iitb.ac.in

5.9 Towards datascience, You Do not Need Neural Networks to Do Continual Learning, Samuele Mazzanti