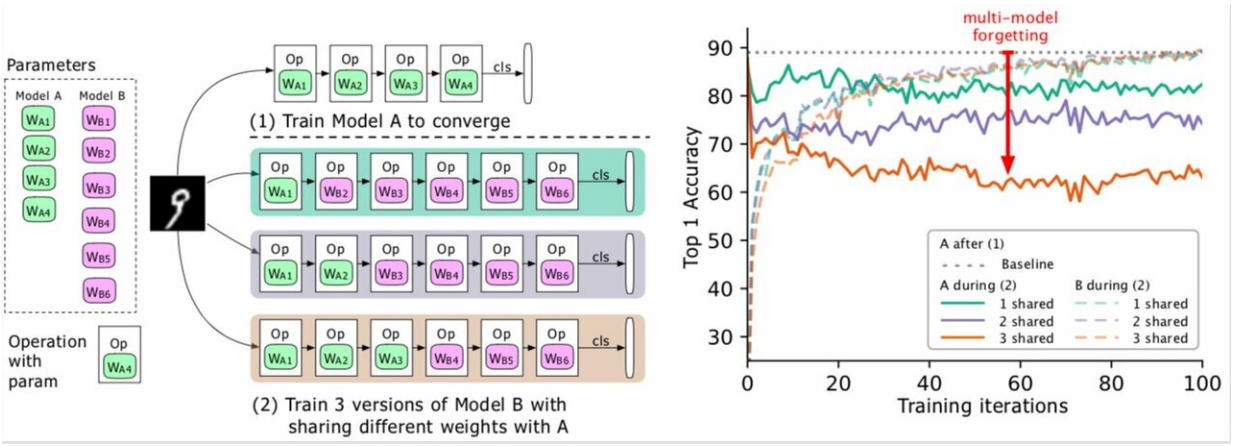
A new approach to overcome multi-model forgetting in deep neural networks

March 11 2019, by Ingrid Fadelli

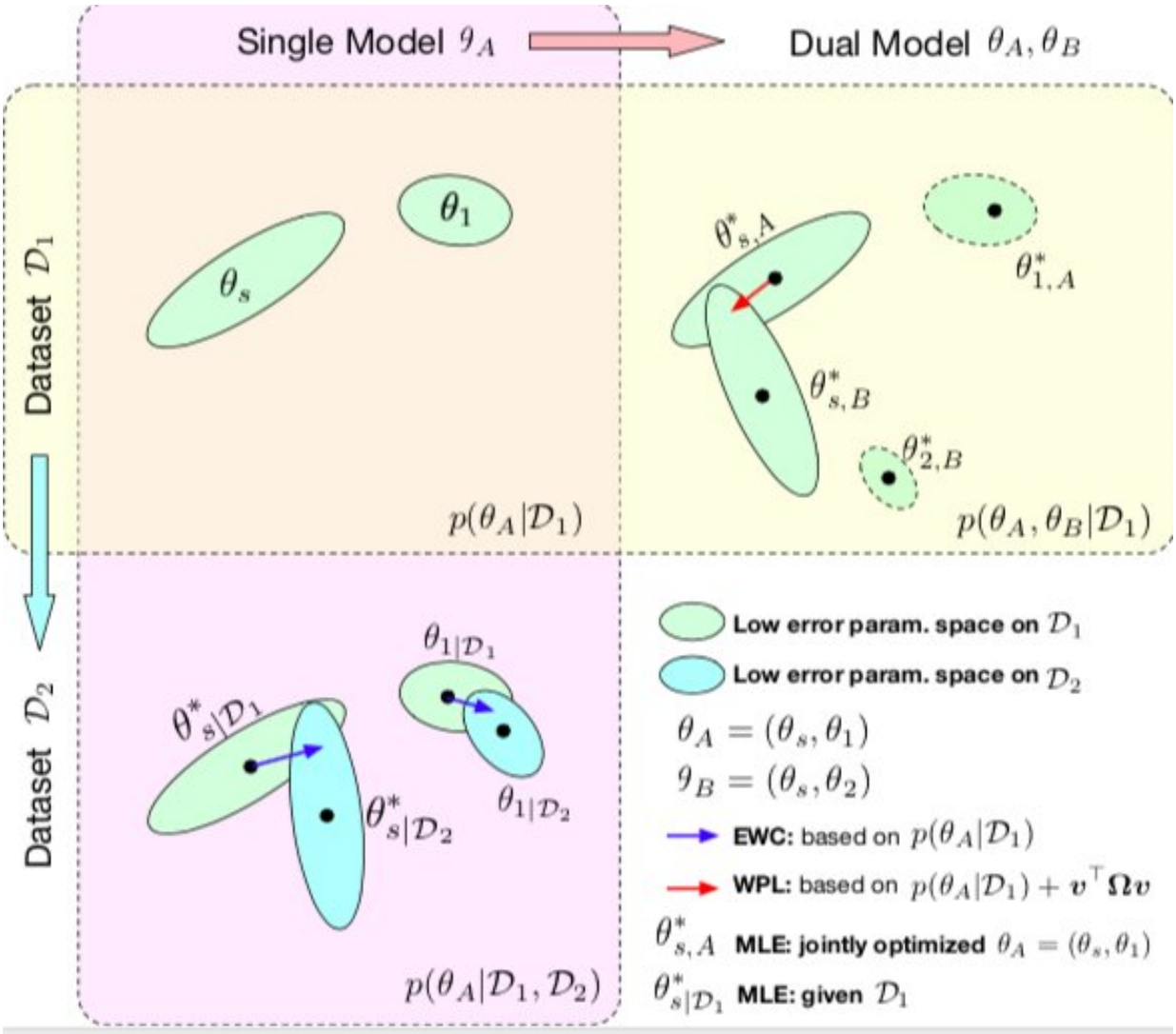


(Left) Two models to be trained (A, B), where A's parameters are in green and B's in purple, and B shares some parameters with A (indicated in green during phase 2). The researchers first train A to convergence and then train B. (Right) Accuracy of model A as the training of B progresses. The different colors correspond to different numbers of shared layers. The accuracy of A decreases dramatically, especially when more layers are shared, and the researchers refer to the drop (the red arrow) as multi-model forgetting. Credit: Benyahia, Yu et al.

In recent years, researchers have developed deep neural networks that can perform a variety of tasks, including visual recognition and natural language processing (NLP) tasks. Although many of these models achieved remarkable results, they typically only perform well on one particular task due to what is referred to as "catastrophic forgetting."

Essentially, catastrophic forgetting means that when a model that was initially trained on task A is later trained on task B, its performance on task A will significantly decline. In a paper pre-published on arXiv, researchers at Swisscom and EPFL identified a new kind of forgetting and proposed a new approach that could help to overcome it via a statistically justified weight plasticity loss.

"When we first started working on our project, designing neural architectures automatically was computationally expensive and unfeasible for most companies," Yassine Benyahia and Kaicheng Yu, the study's primary investigators, told TechXplore via e-mail. "The original aim of our study was to identify new methods to reduce this expense. When the project started, a paper by Google claimed to have drastically reduced the time and resources required to build neural architectures using a new method called weight-sharing. This made autoML feasible for researchers without huge GPU clusters, encouraging us to study this topic more in depth."

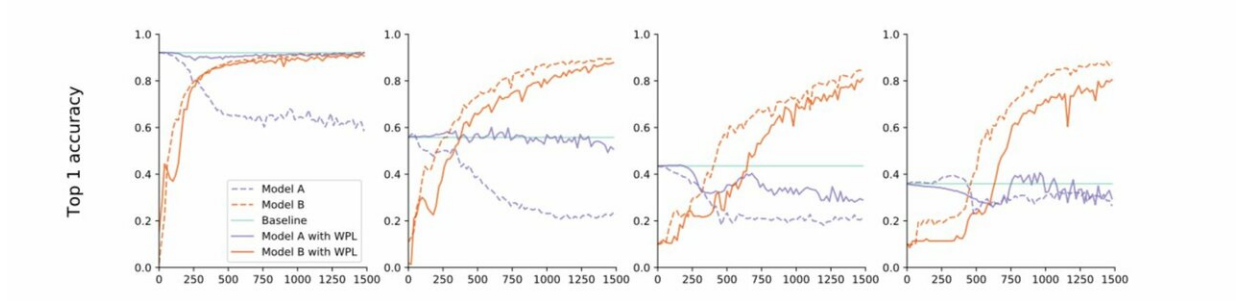


Comparison between EWC and WPL. The ellipses in each subplot represent parameter regions corresponding to low error. (Top left) Both methods start with a single model, with parameters θA = {θs, θ1}, trained on a single dataset D1. (Bottom left) EWC regularizes all parameters based on p(θA|D1) to train the same initial model on a new dataset D2. (Top right) By contrast, WPL makes use of the initial dataset D1 and regularizes only the shared parameters θs based on both p(θA|D1) and v >Ωv, while the parameters θ2 can move freely. Credit: Benyahia, Yu et al.

During their research into neural network-based models, Benyahia, Yu and their colleagues noticed a problem with weight sharing. When they trained two models (e.g. A and B) sequentially, model A's performance declined, while model B's performance increased, or vice versa. They showed that this phenomenon, which they called "multi-model forgetting," can hinder the performance of several auto-mL approaches, including Google's efficient neural architecture search (ENAS).

"We realized that weight-sharing was causing models to impact each other negatively, which was causing the architecture search process to be closer to random," Benyahia and Yu explained. "We also had our reserves on architecture search, where only the final results are shed to light and where there is no good framework to evaluate the quality of the architecture search in a fair way. Our approach could help to fix this forgetting problem, as it is related to a core method that nearly all recent autoML papers rely on, and we consider such impact to be huge to the community."

In their study, the researchers modeled multi-model forgetting mathematically and derived a novel loss, called weight plasticity loss. This loss could reduce multi-model forgetting substantially by regularizing the learning of a model's shared parameters according to their importance for previous models.

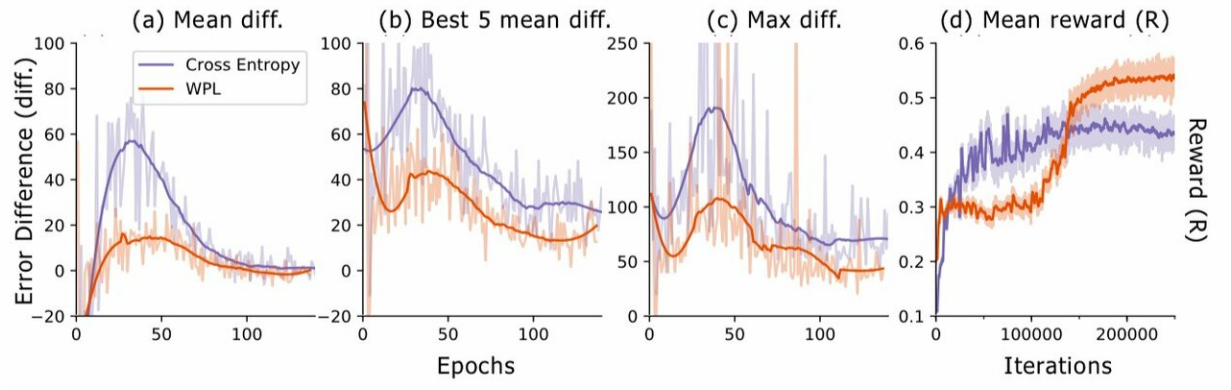


From strict to loose convergence. The researchers conduct experiments on MNIST with models A and B with shared parameters and report the accuracy of Model A before training Model B (baseline, green) and the accuracy of Models A and B while training Model B with (orange) or without (blue) WPL. In (a) they show the results for strict convergence: A is initially trained to convergence. They then relax this assumption and train A to around 55% (b), 43% (c), and 38% (d) of its optimal accuracy. WPL is highly effective when A is trained to at least 40% of optimality; below, the Fisher information becomes too inaccurate to provide reliable importance weights. Thus WPL helps to reduce multi-model forgetting, even when the weights are not optimal. WPL reduced forgetting by up to 99.99% for (a) and (b), and by up to 2% for (c). Credit: Benyahia, Yu et al.

"Basically, due to the over-parameterization of neural networks, our loss decreases parameters that are 'less important' to the final loss first, and keeps the more important ones unchanged," Benyahia and Yu said. "Model A's performance is thus unaffected, while model B's performance keeps increasing. On small datasets, our model can reduce forgetting up to 99 percent, and on autoML methods, up to 80 percent in the middle of training."

In a series of tests, the researchers demonstrated the effectiveness of their approach for decreasing multi-model forgetting, both in instances where two models are trained sequentially and for neural architecture search. Their findings suggest that adding weight plasticity in neural architecture search can significantly improve the performance of multiple models on both NLP and computer vision tasks.

The study carried out by Benyahia, Yu and their colleagues sheds light on the issue of catastrophic forgetting, particularly that which occurs when multiple models are trained sequentially. After modeling this problem mathematically, the researchers introduced a solution that could overcome it, or at least drastically reduce its impact.



Error difference during neural architecture search. For each architecture, the researchers compute the RNN error differences err2−err1, where err1 is the error right after training this architecture and err2 the one after all architectures are trained in the current epoch. They plot (a) the mean difference over all sampled models, (b) the mean difference over the 5 models with lowest err1, and (c) the max difference over all models. In (d), they plot the average reward of the sampled architectures as a function of training iterations. Although WPL initially leads to lower rewards, due to a large weight α in equation (8), by reducing the forgetting it later allows the controller to sample better architectures, as indicated by the higher reward in the second half. Credit: Benyahia, Yu et al.

"In multi-model forgetting, our guiding principle was to think in formulas and not just by simple intuition or heuristics," Benyahia and Yu said. "We strongly believe that this 'thinking in formulas' can lead researchers to great discoveries. That is why for further research, we aim to apply this approach to other fields of machine learning. In addition, we plan to adapt our loss to recent state-of-the-art autoML methods to demonstrate its effectiveness in solving the weight-sharing problem observed by us."

More information: Overcoming multi-model forgetting.

arXiv:1902.08232 [cs.LG]. arxiv.org/abs/1902.08232

Efficient neural architecture search via parameter sharing. arXiv:1802.03268 [cs.LG]. arxiv.org/abs/1802.03268