Treatment Effect Estimation to Guide Model Optimization in Continual Learning

Anonymous submission

Abstract

Continual Learning systems are faced with potentially large numbers of tasks to be learned while the models employed have limited capacity which makes it potentially impossible to learn all required tasks with a single model. In order to detect on which point a model might break we propose to use treatment effect estimation techniques to estimate the effect of training a model on a new task w.r.t. some performance measure.

Motivation

Continually learning new concepts and solving new tasks is one key element of human intelligence which accompanies us through our entire life and seems to be more important than ever these days. For instance, the rising dynamics of the job market demands employees for continually learning, e.g. using a new software being introduced in a company, while not forgetting how to solve problems we face all the time during work, e.g. communicating properly with new customers. Continual Learning (CL) aims to transfer this ability to Machine Learning (ML) to obtain models which are capable of adapting to new tasks without losing the ability to solve tasks seen earlier and to exploit knowledge gathered by learning to solve former tasks (Delange et al. 2021) (Parisi et al. 2019). In CL the notion of a *task* is typically. A task usually corresponds to one of the widespread problem-definitions used in ML, i.e. supervised learning, unsupervised learning or combinations thereof. One of the most prominent problems in CL is catastrophic forgetting which describes the observation that ML-models (especially Neural Networks) tend to *forget* about tasks they have learned previously once they are trained to solve a different task (Delange et al. 2021; Parisi et al. 2019). For example, training a CNN to solve image classification on MNIST and then training the same CNN on Fashion-MNIST starting with the parametrization obtained from training on MNIST will lead to a model being able to solve image classification on Fashion-MNIST, but at the same time causing the MNIST performance to deteriorate. Methods aiming to overcome this issue either train expert-models for each task, replay old data while training on new tasks or fix certain parameters in the models which are considered to be important to solve tasks that can be solved already. Expert-models suffer from high resource consumption and they are not able to exploit old knowledge due to isolated parameter-sets per expert. Replay-based approaches also suffer from high memory consumption. Fixing parameters which are deemed important to solve former tasks thus seems to be the most reasonable approach to take. However, with such approaches questions like the following arise: Does our model have enough capacity to learn a new task? Given a paramerterized model, which effect will training on a new task have w.r.t. the over all model performance? Given that we have trained our model on a sequence of tasks, what would be the state and performance of our model if we had not trained on the last k tasks?

Answering such questions is crucial in order to have guarantees w.r.t. model performance and robustness. Also it increases flexibility of CL-systems since answering such questions allows to determine when model-complexity has to be increased. Estimating effects in counterfactual settings enable CL-systems to find proper trade-offs, e.g. when we *have to* learn a new task, but there is not enough capacity, i.e. we are sure that the overall model-performance will decrease. Then, with counterfactual reasoning, one could identify knowledge in the model which causes the lowest decrease in performance once this knowledge is discarded to make space for the new task to be learned. We will now show that questions of this form can be answered using the framework of *treatment effect estimation*.

Treatment Effect Estimation

Treatment effect estimation (TEE) has its grounding in Causal Inference. The goal is to estimate the effect of an intervention in a system on some variable (Becker and Ichino 2002) (Imbens 2004). The do-calculus proposed by Pearl (2009) is a strong framework which can be used to compute entities we need for TEE. The do-calculus is able to capture asymmetries rendered by causal structures (i.e. if A is the cause of B, changing A changes B but not vice versa). Following this rationale, the average treatment effect (ATE) of a variable X on a variable Y can be defined as follows:

$$\text{ATE} = \mathbb{E}[Y|do(X=1)] - \mathbb{E}[Y|do(X=0)] \qquad (1)$$

ATE is just one of many treatment effect quantities one can estimate/compute, another important quantity is the individual treatment effect (ITE) where one focuses on the outcome of an individual system configuration instead of taking an

expectation (Tabib and Larocque 2019). However, we will focus on ATE here. It is also possible to consider counterfactual scenarios: Instead of asking how the system will behave under an intervention, we ask how the system *would have* behaved if and intervention was performed.

Connecting TEE and CL

In order to perform TEE, we have to know which variable is caused by which other variable(s). In most cases a causal graph is known or can be designed by hand. For example, Figure 1 shows a causal graph of one "step" in a CL-system: t_i denotes a task we obtain at step i, τ_i is a binary decision variable indicating whether we update our model based on t_i, θ_{i-1} and θ_i are the model-parameters at step $i-1$ and i respectively, l_i is the model-performance w.r.t. all tasks at step i and T_{i-1} refers to the set of all tasks we have trained on until step i. Note that all variables except for l_i are independent of T_{i-1} since we observe θ_{i-1} which represents the accumulated knowledge over T_{i-1}, thus older tasks are not needed to estimate these variables.

TEE in Factual Settings Sticking with the example in Figure 1, a natural question to be answered is: Obtaining a new task t_i, will the average model performance l_i significantly decrease when updating the current parameters θ_{i-1} on t_i? Formally this question corresponds to estimating the *conditional average treatment effect (CATE)* $\mathbb{E}[l_i|do(\tau_i = 1), t_i, \theta_{i-1}] - \mathbb{E}[l_i|do(\tau_i = 0), t_i, \theta_{i-1}]$. Estimating this quantity requires us to estimate the case where $do(\tau_i = 1)$ only since $do(\tau_i = 0)$ can be approximated by evaluating the current model on all tasks and average the performance. Estimation of $do(\tau_i = 1)$ case can be done with a 2-step-procedure: First, estimate a distribution over θ_i s.t. the parameters that would result from training on t_i have high probability, denoted by $p(\theta_i|t_i, \theta_{i-1})$. Once this distribution is estimated, the expectation of l_i can be computed by:

$$\int_{l_i} \int_{\theta_i} l_i \cdot p(l_i|\theta_i) \cdot p(\theta_i|t_i, \theta_{i-1}) \tag{2}$$

One possible approach to estimate this quantity is to use density estimators for $p(l_i|\theta_i)$ and $p(\theta_i|t_i, \theta_{i-1})$ and approximate the integral (e.g. using Monte Carlo approaches). Such a quantity allows to determine when the model should be equipped with additional capacity, e.g. by adding more parameters. Additionally, the estimated distribution over parameters can be used to warm-start the next training-stage.

TEE in Counterfactual Settings Another issue we are confronted with in CL-settings is the following: Assume we have a fixed resource-constraint (i.e. our model has a maximum possible capacity) and we obtain a new task which will decrease the overall model performance. Then we have to identify those parts of knowledge represented by our model which will cause the lowest decrease in performance. This can be considered as identifying the task that contributes the lowest amount of knowledge to our model, which in turn can be formulated as a counterfactual question: *What would the model performance be if we had not trained on t_{i-k} but on t_i?* This question can be answered by estimating a series of ATEs in counterfactual settings s.t.

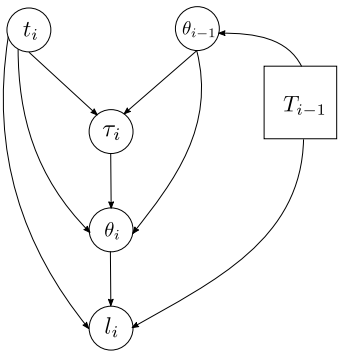

Figure 1: Continual Learning represented as a causal graph. The decision τ_{i-1} if the current model parameters θ_{i-1} are updated using task t_i depend on θ_{i-1} (which is obtained) and t_i only. The model-parameters θ_i at timestep i influence the overall model-performance l_i across all i tasks.

$$\mathbb{E}[\overline{l_i}|do(\tau_i = 1), t_i, \overline{\theta_{i-1}}] - \mathbb{E}[l_i|do(\tau_i = 1), t_i, \theta_{i-1}]$$

is maximized where $\overline{l_i}$ and $\overline{\theta_{i-1}}$ corresponds to the value of l_i and θ_{i-1} respectively if τ_{i-k} had been 0, i.e. if we had not trained on t_{i-k}.

Having such a quantity would not only allow for assessing which knowledge does not contribute much to the overall model-performance, it also can be used to prune models s.t. we minimize the model size while we retain the ability to solve all tasks in an acceptable manner. Also, we can use the estimated parameter-distribution to warm-start the model once the knowledge causing the lowest performance-drop if discarded was identified. Estimating the counterfactual case can be done similar to the factual case.

Conclusion & Further Work

This vision paper looked at the benefits of using the TEE-framework to increase the robustness and flexibility of CL-systems. We propose a starting point that can be used to answer factual and counterfactual questions about CL-systems to guide the optimization behavior. Answering such questions is crucial in productive systems in order to give guarantees w.r.t. model-performance and to minimize computational costs (e.g. by using parameter-estimations as a warm-start). Additionally viewing at CL-systems from a causal lens allows us to make models more transparent, e.g. by identifying knowledge that does not contribute much to the overall model-performance.

Further work should start with solving the factual case, followed by solving the counterfactual case. This requires representing tasks and model-parameters properly which can be achieved using learned representations thereof. Then, the NCM framework proposed by Xia et al. (2021) could be employed to estimate CATE to answer the questions mentioned above. Also, instead of ATE other quantities such as ITE can be considered, e.g. to answer questions about specific tasks.

References

Becker, S. O.; and Ichino, A. 2002. Estimation of Average Treatment Effects Based on Propensity Scores. *The Stata Journal*, 2(4): 358–377.

Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Imbens, G. W. 2004. Nonparametric Estimation of Average Treatment Effects Under Exogeneity: A Review. *The Review of Economics and Statistics*, 86(1): 4–29.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113: 54–71.

Pearl, J. 2009. *Causality*. Cambridge, UK: Cambridge University Press, 2 edition. ISBN 978-0-521-89560-6.

Tabib, S.; and Larocque, D. 2019. Non-parametric individual treatment effect estimation for survival data with random forests. *Bioinformatics*, 36(2): 629–636.

Xia, K.; Lee, K.; Bengio, Y.; and Bareinboim, E. 2021. The Causal-Neural Connection: Expressiveness, Learnability, and Inference. *CoRR*, abs/2107.00793.