ANTI-DISTILLATION: KNOWLEDGE TRANSFER FROM SIMPLE MODEL TO A COMPLEX ONE

Kseniia Petrushina, Andrey Grabovoy, Oleg Bakhteev, Vadim Strijov
Moscow Institute of Physics and Technology
{petrushina.ke, grabovoy.av, bakhteev, strijov}@phystech.edu

ABSTRACT

This paper considers the problem of adapting the model to new data with a large amount of information. We propose to construct a complex model with further knowledge transfer from a simple model to it. It is necessary to take into account not only the quality of the prediction on the original samples but also the adaptability to new data and the robustness of the obtained solution. The novelty of the work lies in the fact that our method allows adapting the pre-trained model to a more heterogeneous dataset. This study considers both probabilistic and algebraic methods for obtaining a new model. In the computational experiment, we analyse the quality of predictions on synthetic and natural samples. FashionMnist and CIFAR10 datasets are our sources of real-world data.

Keywords Distillation · Knowledge Transfer · Weight Initialization · Machine Learning

1 Introduction

In the modern world, the number of machine learning tasks grows rapidly, as is the amount of data to be processed. Typically, training a model from scratch can lead to poor results or take a long time. To get better results faster, researchers have been developing various methods allowing to use of existing trained models to solve new problems. For instance, there are knowledge distillation [2, 3], transfer learning [5], fine-tuning, low-rank model approximation [4]. Moreover, scientists have studied methods for initializing model weights for faster convergence [1]. These approaches help to decrease the time needed for training or inference and achieve higher quality.

Consider the distillation method in more detail. Statement of the initial problem is the transfer of knowledge from a cumbersome neural network or ensemble of ones to a smaller model in the classification task. Hinton and others were able to achieve this by training the student model to reproduce the probability distribution of the classes produced by the teacher model. The use of such soft targets helped to carry more information, so the student models generalization ability is comparable to the teachers. However, this research focuses on model compression under conditions of input data persistence. And we want to solve the inverse problem: in a sense to keep the model under conditions of increasing sample complexity.

Thus, the novelty of this work lies in the proposal of a method for increasing the complexity of the model based on a pre-trained one. This is done by growing the dimension of the weight space and initializing part of the student neural network with teacher model weights. We propose that our approach allows us to speed up neural network training and obtain a more stable solution. As the learning is supposed to start close to the optimum point. In this way, we can adapt the pre-trained model to more variable data and reuse previously learned information.

This paper presents computational experiments on various ways of complicating the model. We consider fully connected, convolutional layers and LSTM modules. An experiment is to compare uniform initialization with one based on a previously trained model and analyse differences in convergence rate, prediction variance, and achieved quality.

2 Setup

We focus on c-class classification, although the same ideas apply to other machine learning tasks. Consider two samples

$$S_1 = \{(x_i, y_i)\}_{i=1}^{N_1},\ x_i \in \mathbb{R}^{d_1},\ y_i \in \Delta^{c_1},$$

$$S_2 = \{(x'_j, y'_j)\}_{j=1}^{N_2},\ x'_j \in \mathbb{R}^{d_2},\ y'_j \in \Delta^{c_2},$$

where Δ^c is the set of c-dimentional probability vectors.

Having a teacher model $f_{tr} : \mathbb{R}^{d_1} \to \Delta^{c_1},\ f_{tr}(x) = f(x, w^*)$, where weights of the model are defined as follows:

$$w^* = \arg\min_w \mathcal{L}_w = \arg\min_w \sum_{i=1}^{N_1} l(y_i,\ f(x_i, w)),$$

here, l is a cross-entropy loss

$$l(y, \hat{y}) = -\sum_{k=1}^{c} y_k \log \hat{y}_k,\ y \in \Delta^c,$$

our approach proposes constructing student model $f_{st} : \mathbb{R}^{d_2} \to \Delta^{c_2},\ f_{st}(x) = g(x, u^*)$,

$$u^* = \arg\min_u \mathcal{L}_u = \arg\min_u \sum_{j=1}^{N_2} l(y'_j,\ g(x'_j, u)).$$

We find the solution to the above optimization problem using gradient optimization methods. Model weights update as

$$u_{t+1} = T(u_t | \mathcal{L}_u, S_2),$$

where $T : \mathbb{R}^{|u|} \to \mathbb{R}^{|u|}$ is the optimization operator and $t \in \mathbb{N}$ is the gradient step number.

In this case, function $\varphi : \mathbb{R}^{|w|} \to \mathbb{R}^{|u|}$ determines student model initialization $u_1 = \varphi(w^*)$.

References

[1] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[2] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[3] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information, 2016.

[4] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 67–76, 2017.

[5] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019.