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

Kseniia Petrushina, Oleg Bakhteev, Andrey Grabovoy, Vadim Strijov
Moscow Institute of Physics and Technology
{petrushina.ke, bakhteev, grabovoy.av, 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 build a more complex model using the parameters of a simple one. We take into account not only the accuracy of the prediction on the original samples but also the adaptability to new data and the robustness of the obtained solution. The work is devoted to developing the method that allows adapting the pre-trained model to a more heterogeneous dataset. In the computational experiment, we analyse the quality of predictions and model robustness on Fashion-Mnist dataset.

Keywords Distillation · Knowledge Transfer · Weight Initialization · Machine Learning

1 Introduction

Training a model from scratch, especially large neural net, usually take a long time. To get better results faster, researchers have been developing various methods allowing to use existing trained models to solve new problems. For instance, there are knowledge distillation [4, 7], transfer learning [11], fine-tuning, low-rank model approximation [10]. Moreover, there are methods for initializing model parameters for faster convergence [3]. These approaches help to decrease the time needed for training or inference and achieve high quality.

Consider the distillation method. 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 problem. Hinton and others [4] 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, that research focuses on reducing model parameters under conditions of input data persistence. Opposite to works devoted to knowledge distillation, we want to maintain the model generalization properties under conditions of increasing sample complexity.

This work proposes a new method for increasing the complexity of the model based on a pre-trained one. An example of previous research is Net2Net technique [1], which allowed to widen or deepen existing pre-trained network using function-preserving transformations. However, our work aims not only at achieving a higher quality of performance compared to models trained from scratch, but also at the robustness of the model to input noise.

This is done by growing the dimension of the weight space, initializing part of the student neural network with teacher model parameters and solving an optimization task. So, by Anti-Distillation, we mean the method of obtaining the initial parameters of a larger student

network using a pre-trained teacher model under the conditions of increasing the number of classes. Our approach allows to speed up neural network training and obtain a more robust model. In this way, we can adapt the pre-trained model to more variable data and reuse previously learned information.

In this paper we conduct computational experiments on various ways of growing the model. We consider fully connected layers. The experiment compares uniform initialization with one based on a previously trained model and analyse differences in convergence rate, prediction variance, achieved quality and resistance to noise attacks.

2 Problem statement

In this section we describe a problem statement for the anti-distillation problem for the classification task. Note that the similar approach can be applied for arbitrary tasks.

There given two datasets

$$\mathfrak{D}_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_1},\ \mathbf{x}_i \in \mathbb{R}^{n_1},\ y_i \in C_1 = \{1, \ldots, c_1\},$$

$$\mathfrak{D}_2 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_2},\ \mathbf{x}_i \in \mathbb{R}^{n_1},\ y_i \in C_2 = \{1, \ldots, c_2\}.$$

We suppose that objects $\mathbf{x}_i$ are generated from the population shared among both datasets $\mathfrak{D}_1, \mathfrak{D}_2$ and have similar properties for these datasets. We also suppose that the dataset $\mathfrak{D}_2$ is more complex for classification and requires more complex classification model.

Given a teacher model $\mathbf{g}_{\text{tr}}$ trained on the first dataset $\mathfrak{D}_1$:

$$\mathbf{g}_{\text{tr}} : \mathbb{R}^{n_1} \to \Delta^{c_1}, \quad \mathbf{g}_{\text{tr}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}, \hat{\mathbf{u}}),$$

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

The teacher model $\mathbf{g}_{\text{tr}}$ is defined as follows:

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \mathcal{L}_{\text{ce}}(\mathbf{u}, \mathfrak{D}_1) = \arg\min_{\mathbf{u}} \sum_{i=1}^{m_1} l\left(y_i,\ g(\mathbf{x}_i, \mathbf{u})\right),$$

here, l is the cross-entropy loss

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

Our task is to construct the student model that minimizes cross-entropy on the validation part of the second dataset $\mathfrak{D}_2$

$$\mathbf{f}_{\text{st}} : \mathbb{R}^{n_1} \to \Delta^{c_2}, \quad \mathbf{f}_{\text{st}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, \hat{\mathbf{w}}),$$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \mathcal{L}_{\text{ce}}(\mathbf{w}, \mathfrak{D}_2^{\text{val}}), \tag{1}$$

where $\mathfrak{D}_2 = \mathfrak{D}_2^{\text{train}} \sqcup \mathfrak{D}_2^{\text{val}}$.

Since we cannot optimize validation loss straightforwardly, the common practice is using gradient optimization methods on the training part $\mathfrak{D}_2^{\text{train}}$ of the dataset $\mathfrak{D}_2$. In order to minimize overfitting and use more information about the data we obtain information from the teacher model $\mathbf{g}_{\text{tr}}$. Here we use our proposition that the datasets $\mathfrak{D}_1$ and $\mathfrak{D}_2$ share common properties.

The function

$$\boldsymbol{\varphi} : \mathbb{R}^{N_{\text{tr}}} \to \mathbb{R}^{N_{\text{st}}}$$

maps the teacher model parameters to student initial parameters $\mathbf{w} = \boldsymbol{\varphi}(\hat{\mathbf{u}})$.

Hypothesis 1 *The student models initialized by the result of applying the function $\boldsymbol{\varphi}$ to the parameters of the pre-trained teacher model is more persistent and achieve higher accuracy than models with default parameters.*

3 Teacher model extension

The major problem of the proposed method is that teacher model $\mathbf{g}_{\text{tr}}$ trained on a simple dataset $\mathfrak{D}_1$ can be much simpler than the student model $\mathbf{f}_{\text{st}}$. In order to use more information from the teacher model parameters $\hat{\mathbf{u}}$ we need to extend teacher model parameter space N_{tr} dimension to the dimension N_{st} of the student model parameter space.

To deal with it we optimize the following composite loss function:

$$\boldsymbol{\varphi}(\mathbf{u}) = \underset{\mathbf{w}\in\mathbb{R}^N_{\text{st}}}{\arg\min}\, \mathcal{L}(\mathbf{w}), \tag{2}$$

where

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{ce}}(\mathbf{w},\mathfrak{D}_1) + \lambda_2\mathcal{L}_2(\mathbf{w},\mathbf{u}) + \lambda_3\mathcal{L}_3^{\delta}(\mathbf{w},\mathfrak{D}_1) + \lambda_4\mathcal{L}_4\left(\frac{\partial^2\mathcal{L}_{\text{ce}}}{\partial\mathbf{w}^2}\right), \lambda_i \geq 0$$

Here $\mathcal{L}_{\text{ce}}(\mathbf{w},\mathfrak{D}_1)$ is a cross-entropy loss on the $\mathfrak{D}_1$,

$\mathcal{L}_2(\mathbf{w},\mathbf{u}) = \|\mathbf{u} - \mathbf{Pr}[\mathbf{u}]\|_2^2$ provides a small difference between the parameters of the teacher model and the student model in the respective places, where **Pr** takes only first parameters common for both models (in case of multilayer perceptron models, **Pr** takes parameters of the same neurons for each layer of the model).

The loss components $\mathcal{L}_3^{\delta}(\mathbf{w},\mathfrak{D}_1) = \sum_{(\mathbf{x},y)\in\mathfrak{D}_1} \mathbb{E}_{\mathbf{x}'\in U_\delta(\mathbf{x})}\mathcal{L}_{\text{ce}}(\mathbf{w},\mathbf{x}',y)$ and $\mathcal{L}_4\left(\frac{\partial^2\mathcal{L}_{\text{ce}}}{\partial\mathbf{w}^2}\right) = \text{tr}\left(\frac{\partial^2\mathcal{L}_{\text{ce}}}{\partial\mathbf{w}^2}\right)$ account for the robustness of solution to noise in input data, where $U_\delta(\mathbf{x})$ represents uniform distribution in range $[\delta - \mathbf{x}; \delta + \mathbf{x}]$.

The case of our interest, Anti-Distillation, implies $\lambda_2 > 0$, i.e. the optimization that make model parameters close for the teacher and student close enough. We also are interested in getting a model that is robust to input data corruption. For such property we use terms $\mathcal{L}_3$ and $\mathcal{L}_4$. Both of these terms regularize Hessian of the cross-entropy loss function [9, 2].

4 Computational experiment

We compere different approaches to initialization:

1. Scratch – filling all model parameters with $U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$.
2. Zero initialization – filling extended parameters with zeroes.
3. Uniform initialization $U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$ – filling extended parameters with uniformly distributed variables, n is the number of layer input neurons.
4. Transfer learning – taking pre-trained model and changing only classification layer.
5. Net2net – extending model parameters according to the paper [1].
6. L1 + L3 – deriving initialization of student model by solving optimization problem 2 with $\lambda_1, \lambda_3 = 1$ and $\lambda_2, \lambda_4 = 0$.
7. Antidistillation – initializing using Anti-Distillation method with $\lambda_1, \lambda_2, \lambda_3$ hyperparameter optimization through Gaussian processes ($\lambda_4 = 0$).

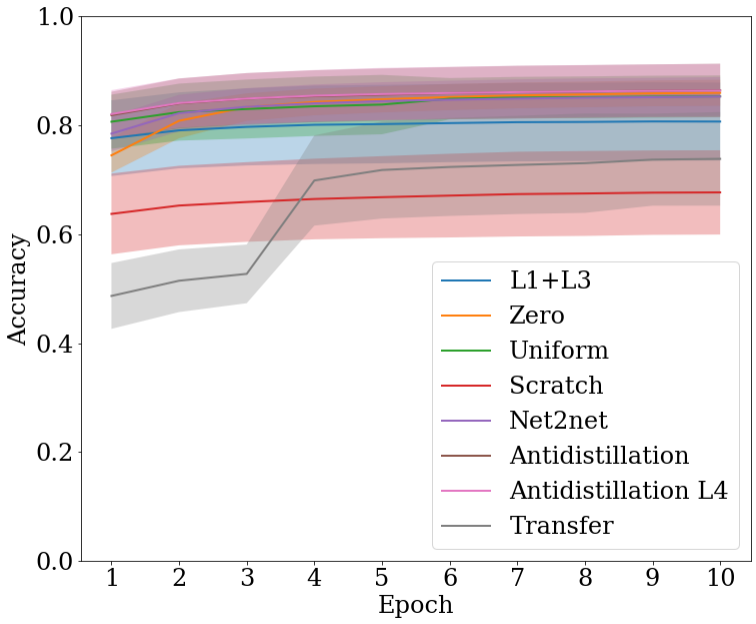

Figure 1: Comparison of validation accuracy for different initialization methods

8. Antidistillation L4 – optimization of all λ_i.

The goal of computational experiment is to compare the performance of models depending on the initialization of parameters.

4.1 Data

Fashion-MNIST is a dataset of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes [8].

4.2 Configuration of algorithm run

Carry out the experiments as follows: train the teacher model, increase its complexity as described in 3, and compare different ways of initializing the parameters of the model. Optimization is done using Adam optimization algorithm [5]. We compare them by measuring accuracy of predictions, cross-entropy loss value on validation sample, and prediction variance. Also we investigate the case of noisy input data, considering above quality measures depending on the percentage of image corrupted. We average results over 10 training runs and study the mean and variance of the metrics.

4.3 Error analysis

$\mathfrak{D}_2$ set consists of Fasion-MNIST and $\mathfrak{D}_1 = \{(\mathbf{x}, y) \mid (\mathbf{x}, y) \in \mathfrak{D}_2, y \in C_1\}$, $C_1 \subset C_2$, $C_1 = \{0, \dots 4\}, C_2 = \{0, \dots 9\}$.

As seen in 1 models utilizing Anti-Distillation with on average have smaller variance and higher accuracy than models with different initialization of parameters.

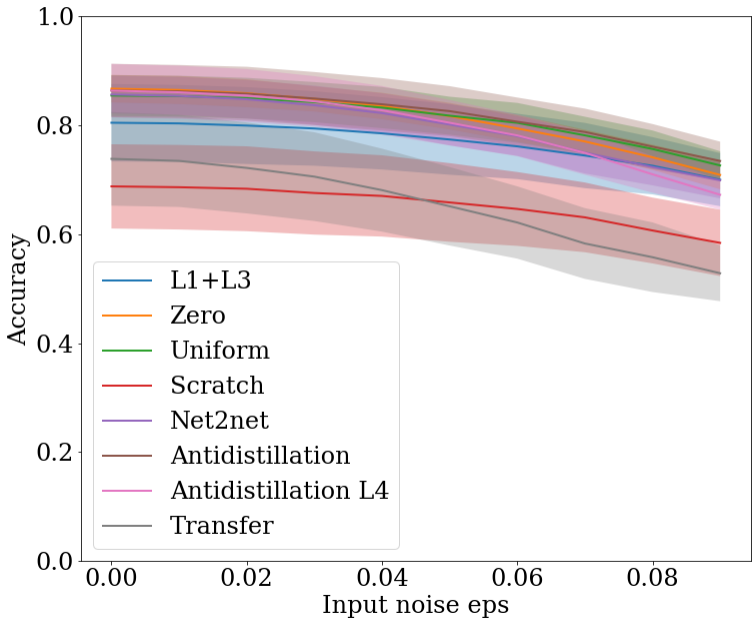

Figure 2: Dependence of validation accuracy on uniform data noise

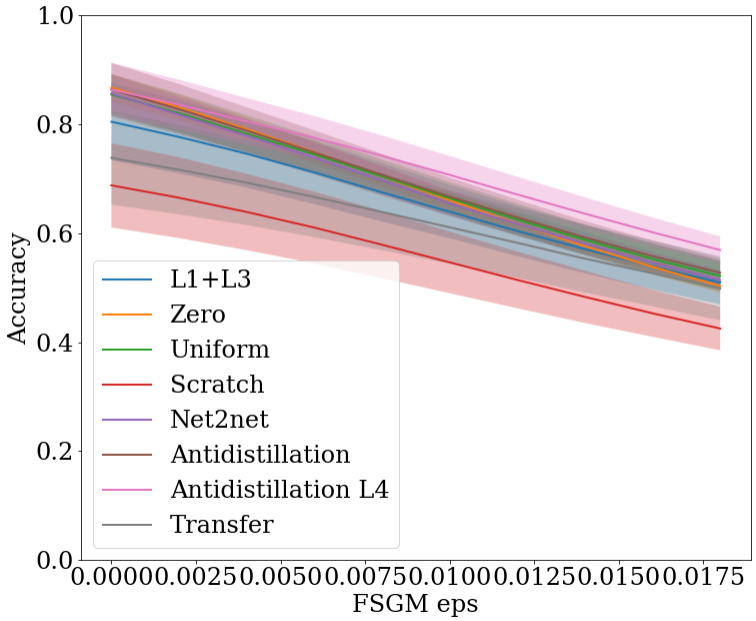

Figure 3: Dependence of validation accuracy on adversarial data noise

In 2, we can see that the Anti-Distillation method without $\mathcal{L}_4$ loss retains the best quality with a lot of uniform noise in the data.

The figure 3 shows that Anti-Distillation is the most adversarial attack-resistant method of initializing model parameters.

5 Conclusion

We proposed a new method for parameters initialization of fully-connected neural newtorks, which helps to achieve higher accuracy on *more complex* dataset and makes model more persistent to noise in input data. As a next step, we plan to consider other datasets, for example CIFAR-10 [6], and apply a similar approach to other neural network architectures: CNN and RNN.

References

[1] Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer, 2015.

[2] Xiangning Chen and Cho-Jui Hsieh. Stabilizing differentiable architecture search via perturbation-based regularization. In *International conference on machine learning*, pages 1554–1565. PMLR, 2020.

[3] 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.

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

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[6] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

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

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[9] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020.

[10] 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.

[11] 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.