Anti-Distillation: Knowledge Transfer from a Simple Model to the Complex One

Kseniia Petrushina
Moscow Institute of Physics and Technology
Moscow, Russia
petrushina.ke@phystech.edu

Oleg Bakhteev
Moscow Institute of Physics and Technology
FRC CSC of the RAS
Moscow, Russia
bakhteev@phystech.edu

Andrey Grabovoy
Moscow Institute of Physics and Technology
Antiplagiat Company
Moscow, Russia
grabovoy.av@phystech.edu

Vadim Strijov
FRC CSC of the RAS
Moscow, Russia
strijov@phystech.edu

***Abstract*—The 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.**

***Index Terms*—knowledge transfer, weight initialization, distillation**

I. 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 [1], [2], transfer learning [3], fine-tuning, low-rank model approximation [4]. Moreover, there are methods for initializing model parameters for faster convergence [5], [6]. These approaches help to decrease the time needed for training or inference and achieve high quality [7].

The distillation method is one of the model compression methods [8]. Statement of the initial problem is the transfer of knowledge from a cumbersome neural network or ensemble of models [9] to a smaller model in the classification problem. Hinton and others [1] 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 [10], 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 [11], [12].

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. Opposite to the paper [13], where the term "Anti-distillation" is considered as a decorrelation of multiple model answers, in our paper "Anti-distillation" can be considered as an information transfer from the simple model to a complex one, which is opposite to the usual distillation approach [1]. also proposes a method of 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 train a fully connected neural network as a student and teacher to analyze our anti-distillation method. The experiment compares various model initialization methods such as zero-shot and net2net with our anti-distillation. We compare methods according to differences in convergence rate, prediction variance, achieved quality and resistance to noise.

II. 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 are two datasets

$$\mathfrak{D}_1 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_1},\ \mathbf{x}_i \in \mathbb{R}^n,\ y_i \in C_1 = \{1, \ldots, c_1\},$$
$$\mathfrak{D}_2 = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_2},\ \mathbf{x}_i \in \mathbb{R}^n,\ y_i \in C_2 = \{1, \ldots, c_2\},$$

where m_1 and m_2 are numbers of objects in $\mathfrak{D}_1$ and $\mathfrak{D}_2$ respectively, n is input dimensionality. C_1 and C_2 are sets of class labels $1, \ldots, c_1, \ldots, c_2$.

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 $\mathfrak{D}_1$:

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

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

The teacher model $\mathbf{g}_{\text{tr}}$ parameters are 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

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

that minimizes cross-entropy on the validation part of the second dataset $\mathfrak{D}_2$

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

where $\mathfrak{D}_2 = \mathfrak{D}_2^{\text{train}} \sqcup \mathfrak{D}_2^{\text{val}}$ and $\hat{\mathbf{w}}$ are optimal model parameters.

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.

III. 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:

$$\boldsymbol{\varphi}(\mathbf{u}) = \arg\min_{\mathbf{w} \in \mathbb{R}^{N_{\text{st}}}} \mathcal{L}(\mathbf{w}), \tag{1}$$

where

$$\mathcal{L}(\mathbf{w}) = \lambda_1 \mathcal{L}_{\text{ce}}(\mathbf{w}, \mathfrak{D}_1) + \lambda_2 \mathcal{L}_2(\mathbf{w}, \mathbf{u}) + \lambda_3 \mathcal{L}_3^{\delta}(\mathbf{w}, \mathfrak{D}_1) + \lambda_4 \mathcal{L}_4(\mathbf{w}),$$

$$\forall i \in \overline{1,4}\ \lambda_i \geq 0$$

Here $\mathcal{L}_{\text{ce}}(\mathbf{w}, \mathfrak{D}_1)$ is the cross-entropy loss, responsible for the quality of the student model on the $\mathfrak{D}_1$.

The second term

$$\mathcal{L}_2(\mathbf{w}, \mathbf{u}) = \|\mathbf{u} - \mathbf{Pr}[\mathbf{w}]\|_2^2$$

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 component

$$\mathcal{L}_3^{\delta}(\mathbf{w}, \mathfrak{D}_1) = \sum_{(\mathbf{x}, y) \in \mathfrak{D}_1} \mathbb{E}_{\mathbf{x}' \in U_\delta(\mathbf{x})} \mathcal{L}_{\text{ce}}(\mathbf{w}, \mathbf{x}', y)$$

accounts for the robustness of solution to noise in input data, where $U_\delta(\mathbf{x})$ represents uniform distribution in $[\delta - \mathbf{x}; \delta + \mathbf{x}]$.

The final term

$$\mathcal{L}_4(\mathbf{w}) = \text{tr}\left(\frac{\partial^2 \mathcal{L}_{\text{ce}}}{\partial \mathbf{w}^2}\right)$$

performs regularization of the Hessian, which also increases the robustness of the model.

Note, that the last term $\mathcal{L}_4$ involves Hessian computation, which naive calculation can be resource-consuming. In this paper, we use the method of stochastic approximation [14] of the trace of Hessian with the fast Hessian-vector product multiplication [15]. The resulting complexity of such a procedure is linear from the number of the model $\mathbf{f}_{\text{st}}$ parameters.

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 [16], [17].

IV. Computational experiment

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

We compare different approaches to initialization:

1) Xavier – filling all model parameters with $U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$, n is the number of input layer neurons [5], i.e. default initialization of model parameters.
2) Zero pad – filling extended parameters with zeroes.
3) Uniform pad – filling extended parameters with uniformly distributed variables $U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$, n is the number of input layer neurons.
4) Transfer learning – taking pre-trained model and changing only classification layer for a new classification task. First, the model was trained with frozen parameters on all layers except the classification layer. After 3 learning epochs, all parameters were unfrozen. Starting from the fourth epoch, all parameters of the neural network were optimized.
5) Net2Net – incremental algorithm of the extension of model parameter space [10].
6) With Data Noise – deriving initialization of student model by solving optimization problem 1 with $\lambda_1, \lambda_3 = 1$ and $\lambda_2, \lambda_4 = 0$.
7) Anti-Distillation, $\lambda_4 = 0$ – initializing using Anti-Distillation method with $\lambda_1, \lambda_2, \lambda_3$ hyperparameter optimization through Bayesian optimization ($\lambda_4 = 0$) [18].
8) Anti-Distillation – optimization of all λ_i.

Quality criteria are: accuracy on validation set, accuracy on validation set corrupted by FSGM-attack [19], accuracy on validation set, provided that the model parameters are corrupted with noise: $\mathbf{w}_\varepsilon = \mathbf{w} + \varepsilon\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

A. 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 [20].

B. Configuration of algorithm run

Carry out the experiments as follows: train the teacher model, increase its complexity as described in III, and compare different ways of initializing the parameters of the model. We consider fully connected networks. Teachers have the following sizes of hidden layers: $[128, 64, 32]$. Student models have $[256, 128, 64]$ neurons in hidden layers.

Teachers were trained for 30 epochs with an initial learning rate of 1e-2, decreasing further to 1e-3 after 10 epochs. Students were compared on training for 10 epochs at a learning rate of 1e-3. Optimization is done using Adam optimization algorithm [21]. We compare initialization methods 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

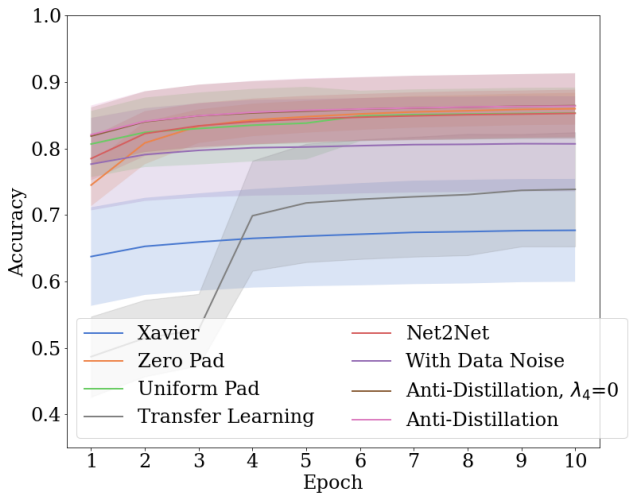

Fig. 1. Comparison of validation accuracy for different initialization methods

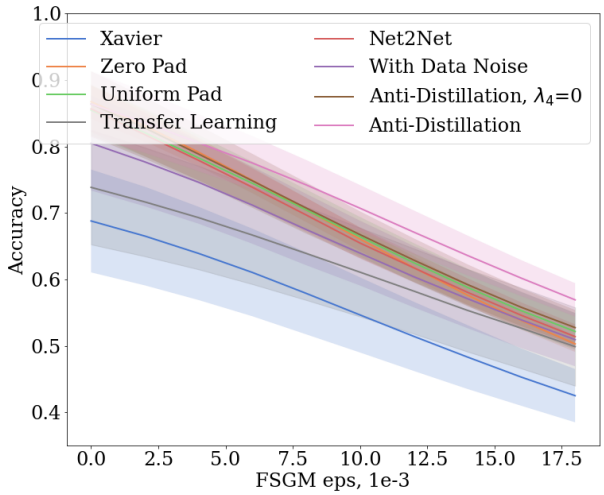

Fig. 2. Dependence of validation accuracy on adversarial data noise

results over 10 training runs and study the mean and variance of the metrics. [1]

C. Error analysis

The dataset $\mathfrak{D}_2$ 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 Figure 1 models utilizing Anti-Distillation in average have smaller variance and higher accuracy than models with different initialization of parameters. Training model from scratch turned out to be not the best solution. The proposed method gives us better results with lower number of iterations

[1] https://github.com/intsystems/anti-distillation

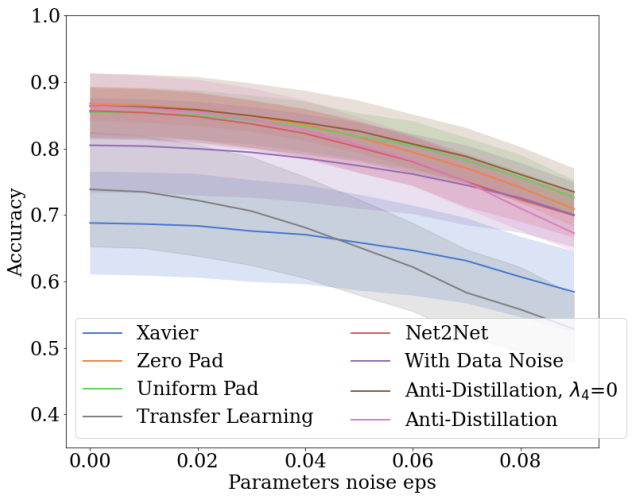

Fig. 3. Dependence of validation accuracy on noise intensity parameter ε

for convergence. Note, that we did not consider the number of iterations required for the teacher model extension which also requires an optimization procedure. We argue that in many real-life cases this time can be neglected as the proposed method allow us to extend the teacher model once using only the basic dataset $\mathfrak{D}_1$ for further use in multiple student training tasks [22].

The Figure 2 shows that Anti-Distillation is the most adversarial attack-resistant method of initializing model parameters, since it has the highest validation accuracy with a large margin at high noise levels.

In Figure 3, we can see that the Anti-Distillation method without Hessian regularization ($\lambda_4 = 0$) is the most robust to normal noise in the model parameters as it retains the highest accuracy at the maximum considered noise level.

TABLE I
ACCURACY ON VALIDATION SET.

Initialization method	Accuracy	FSGM-attack	Noise in parameters
Xavier	0.68 ± 0.08	0.42 ± 0.04	0.58 ± 0.06
Zero Pad	**0.86 ± 0.02**	0.50 ± 0.01	0.71 ± 0.03
Uniform Pad	0.85 ± 0.04	0.52 ± 0.03	**0.73 ± 0.03**
Transfer Learning	0.74 ± 0.09	0.50 ± 0.06	0.53 ± 0.05
Net2Net	0.85 ± 0.04	0.51 ± 0.02	0.70 ± 0.03
With Data Noise	0.81 ± 0.07	0.51 ± 0.03	0.70 ± 0.05
Anti-Distillation, λ_4=0	**0.86 ± 0.05**	0.53 ± 0.03	**0.73 ± 0.04**
Anti-Distillation	**0.86 ± 0.05**	**0.57 ± 0.03**	0.67 ± 0.03

The results are also presented in the tabular form. The Table I contains data on validation accuracy for different initialization methods after the last training epoch, the validation accuracy values at the highest level of image noise by the adversarial attack and information about the accuracy values for the highest noise level in the model parameters.

V. CONCLUSION

The paper considered the problem of the model extension to a new dataset. We considered a case when the new dataset is more complex than the original one. To deal with it we proposed a new method for transfer knowledge from a simple model to a more complex model, which helps to achieve higher accuracy on *more complex* dataset and makes model more persistent to adversarial noise in input data and normal noise in model parameters. To demonstrate the efficiency of the proposed method we conducted an experiment on Fashion-MNIST dataset. As a next step, we plan to consider other datasets, for example CIFAR-10 [23], and apply a similar approach to other neural network architectures: CNN and RNN, conduct experiments with different default parameters initialization.

REFERENCES

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.
[2] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, "Unifying distillation and privileged information," 2016.
[3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *CoRR*, vol. abs/1911.02685, 2019. [Online]. Available: http://arxiv.org/abs/1911.02685
[4] X. Yu, T. Liu, X. Wang, and D. Tao, "On compressing deep models by low rank and sparse decomposition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 67–76.
[5] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html
[6] S. Koturwar and S. H. I. Merchant, "Weight initialization of deep neural networks(dnns) using data statistics," *ArXiv*, vol. abs/1710.10570, 2017.
[7] M. Skorski, A. Temperoni, and M. Theobald, "Revisiting weight initialization of deep neural networks," in *Proceedings of The 13th Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, V. N. Balasubramanian and I. Tsang, Eds., vol. 157. PMLR, 17–19 Nov 2021, pp. 1192–1207. [Online]. Available: https://proceedings.mlr.press/v157/skorski21a.html
[8] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression." in *KDD*, T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, Eds. ACM, 2006, pp. 535–541.
[9] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.
[10] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," 2015. [Online]. Available: https://arxiv.org/abs/1511.05641
[11] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.
[12] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.
[13] G. I. Shamir and L. Coviello, "Anti-distillation: Improving reproducibility of deep networks," *arXiv preprint arXiv:2010.09923*, 2020.
[14] Z. Bai, G. Fahey, and G. Golub, "Some large-scale matrix computation problems," *Journal of Computational and Applied Mathematics*, vol. 74, no. 1-2, pp. 71–89, 1996.
[15] B. A. Pearlmutter, "Fast exact multiplication by the hessian," *Neural computation*, vol. 6, no. 1, pp. 147–160, 1994.
[16] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, "Pyhessian: Neural networks through the lens of the hessian," in *2020 IEEE international conference on big data (Big data)*. IEEE, 2020, pp. 581–590.

[17] X. Chen and C.-J. Hsieh, "Stabilizing differentiable architecture search via perturbation-based regularization," in *International conference on machine learning*. PMLR, 2020, pp. 1554–1565.
[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.
[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
[20] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980
[22] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.
[23] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html