Extended Abstract Track

Knowledge Distillation for Teaching Symmetry Invariances

Editors: -

Abstract

Knowledge distillation is used in an attempt to transfer model invariances related to specific symmetry transformations of the data. The efficacy of knowledge distillation in transferring permutation and translation invariance is empirically evaluated in four different settings. It is observed that knowledge distillation fails at transferring invariances in the considered model pairs; data augmentation performs just as well or better.

Keywords: Knowledge Distillation; Geometric Learning; Data Augmentation

1. Introduction

Extremely large neural networks are able to learn data representations that generalize well. Conversely, smaller networks lack the inductive biases to find the same representations as their larger counterparts from training data alone. However, the former may have the *capacity* to represent the solutions found by the latter (LeCun et al., 1989; Ba and Caruana, 2014; Frankle and Carbin, 2019; Urban et al., 2017). This work focuses on investigating this claim for the specific case of symmetry invariances.

The seminal work of Hinton et al. (2015) expands on the idea of model compression (Buciluǎ et al., 2006), establishing Knowledge Distillation (KD) as a more general paradigm through which a smaller, so called student model learns to generalise in the same way as a much larger, heavily regularised teacher model. Thus, training with KD allows for deploying a model that performs better than its conventionally trained counterpart, while simultaneously achieving faster inference times and using less computational resources than a large model. Then, it follows that if a large teacher model exhibits invariances with respect to certain symmetries in the data which help with generalisation, then they would be transferred to the student model.

Contribution Within the KD framework, we consider a teacher with an invariance embedded in its structure, e.g., the Deep Sets (DS) (Zaheer et al., 2018) architecture and permutation invariance. Further, we consider a simpler student architecture lacking the invariance exhibited by teacher, e.g., a Multi Layer Perceptron (MLP). We then attempt to teach the invariance of the teacher to the student by training the latter using KD. The students are evaluated with respect to a set of metrics that tests how well they learned to generalise and specifically how well they learned the teacher invariance. Our results give a clearer understanding of what knowledge can actually be distilled in KD.

Related Work Stanton et al. (2021) makes a first investigation into the KD paradigm by decoupling student generalisation ability from teacher-student output agreement, i.e., fidelity. Furthermore, additional attempts at understanding KD have been initiated in recent times: some general (Ojha et al., 2023) and some pertaining to a specific type of models (Liu et al., 2023). However, what knowledge is distilled in a high fidelity KD training remains esoteric even after these studies: it is not well understood whether the student learns specific teacher properties or KD simply has a dominant regularising effect. Hence, our study fits this literature gap.

2. Models and Methods

2.1. Knowledge Distillation

There are different ways to distill knowledge from a teacher to a student model. For our experiments, we employed offline output-based KD (Hinton et al., 2015). Therein, the student model minimizes both the conventional task-specific loss and a distillation loss; the former quantifies the difference between the softened output distributions of the teacher and student models. The conventional distillation loss function from Hinton et al. (2015) is

$$\mathcal{L}_{\text{KD}} = (1-\alpha)\mathcal{H}(\mathbf{y}_{\text{true}}, \mathbf{P}_s) + \alpha\mathcal{H}(\mathbf{P}_t^\tau, \mathbf{P}_s^\tau), \tag{1}$$

where $\mathcal{H}$ refers to the cross-entropy, $\alpha \in [0,1]$ is a tunable parameter, $\mathbf{y}_{\text{true}}$ are the truth labels, $\mathbf{P}_s$ is the student softmax output, and $\mathbf{P}_{t(s)}^\tau$ are the teacher (student) softmax outputs with temperature τ. Following Stanton et al. (2021), we set $\alpha = 1$ to avoid confounding from the true labels and arrive at the loss function for the distillation process:

$$\mathcal{L}_s := \tau^2 \text{KL}(\mathbf{P}_t^\tau || \mathbf{P}_s^\tau) \tag{2}$$

where KL denotes the Kullback-Leiber divergence measure. Conducting KD on a teacher-student pair with identical architectures is known as self-distillation (Furlanello et al., 2018).

2.2. Data, Teachers, and Students

First, the MNIST (Deng, 2012) data is used with ResNet18 (Chaman and Dokmanic, 2021) as the teacher, which is translation invariant. Two teachers are trained, denoted as ResNet and ResNet', for 10 and 2 epochs, respectively. The student is an MLP with 4 hidden layers, each with 2048 neurons, and ReLU activations: this configuration ensures that the MLP is likely to have the capacity to model the ResNet18 invariance. Thus, with this setup we evaluate whether the translation invariant behaviour of the ResNet18 is distilled.

Then, the ModelNet40 (Wu et al., 2015) data is used, with standard scaling and downsampled to 1000 points. A Dynamic Graph Convolutional Network (DGCNN) with a translation invariant edge function (Wang et al., 2019) is the teacher. Two different DGCNN teachers are trained, DGN and DGN', the first with exactly the same hyperparameters as in Wang et al. (2019) and the second with only two convolutional layers instead of four. We use two students for each DGCNN: a permutation invariant DS, *dsinv*, and a permutation equivariant DS, *dsequiv*, identical to Zaheer et al. (2018). Thus, we evaluate what degree of *translation invariance* is distilled from the DGCNN to the DS.

The last invariance distillation experiment is done on physics data (Pierini et al., 2020). For details on the data, see Moreno et al. (2020); data is processed as in Odagiu et al. (2024) and downsampled to the 16 most energetic particles. The teacher in this case is an invariant DS, *dsinv*, and the student is an MLP with hyperparameters like in Odagiu et al. (2024). A second teacher dsinv' is also trained, with one less layer in the first MLP compared to the original dsinv model. The efficacy of transferring permutation invariance is evaluated by distilling the dsinv to the MLP.

2.3. General Experiment Design

We perform a set of four experiments for each data set. First, the student model is trained independently on the data using the loss pertaining to the given task, without KD. Then, a new instantiation of the same architecture is trained through self-distillation using the loss shown earlier in Eq. 2. Second, the student is reset and trained on data that is transformed with respect to a symmetry exhibited by the teacher; self-distillation is performed again on a new student model instantiation. Third, the teacher is trained independently on the data and distilled into a new student using Eq. 2. Fourth, a different teacher model, denoted as teacher', is trained independently on the data and distilled into a new student. This last experiment is performed to control for confounding in the fidelity measure, as initially established by Stanton et al. (2021) and detailed in Sec. 2.4. The trainings wherein Eq. 2 is used are repeated for $T \in \{1, 4, 8, 16\}$. Finally, we also attempt to teach the chosen invariances to the respective students via training on an augmented data set.

2.4. Evaluation

For consistency, the generalisation ability of our models is measured by using the same metrics as Stanton et al. (2021): the top-1 accuracy, the negative log-likelihood (NLL), and the expected calibration error (ECEL). Aside from using these metrics to evaluate the generalisation ability of the student, the distillation process is validated by employing two additional metrics: the top-1 student-teacher agreement and the KL divergence between their softmaxed output distributions.

Furthermore, the invariance under certain symmetries is evaluated for all of the models resulting from the experiments described in Sec. 2.2 using **IM** of network n as

$$\mathbf{IM}(\mathbf{D}, n) := \frac{1}{|\mathbf{D}|} \sum_{\mathbf{D}} |\mathbf{P}_n(x_i) - \mathbf{P}_n(x_i')|, \tag{3}$$

where $(\mathbf{x'}_i, \mathbf{y}_i)$ is created from $(\mathbf{x}_i, \mathbf{y}_i)$ by a symmetry transformation of $\mathbf{x}_i$. $\mathbf{D}$ is the set containing all pairs $\{(\mathbf{x}_i, \mathbf{y}_i),(\mathbf{x'}_i, \mathbf{y}_i)\}$. $\mathbf{IM}(\mathbf{D}, s)$ is 0 if n is exactly invariant for the considered transformation.

3. Results and Conclusions

The results are presented in Fig. 1. Notice that for each distillation experiment (row), the respective students fail at learning the invariance of the teacher. As shown in column 4 of Fig. 1, distillation from teacher to student leads to comparable invariance as obtained by performing self-distillation. This is especially true for high-fidelity students.

Although generalisation ability of students improves, the invariance of the teacher is not transferred to the student to any significant degree. Moreover, the student models that perform the best in the invariance metric are the ones that are trained on transformed data. Thus, for learning invariances, we observe that KD does not provide anything beyond what can be achieved by training on augmented data, while the latter is also simpler and less computationally expensive than the former.

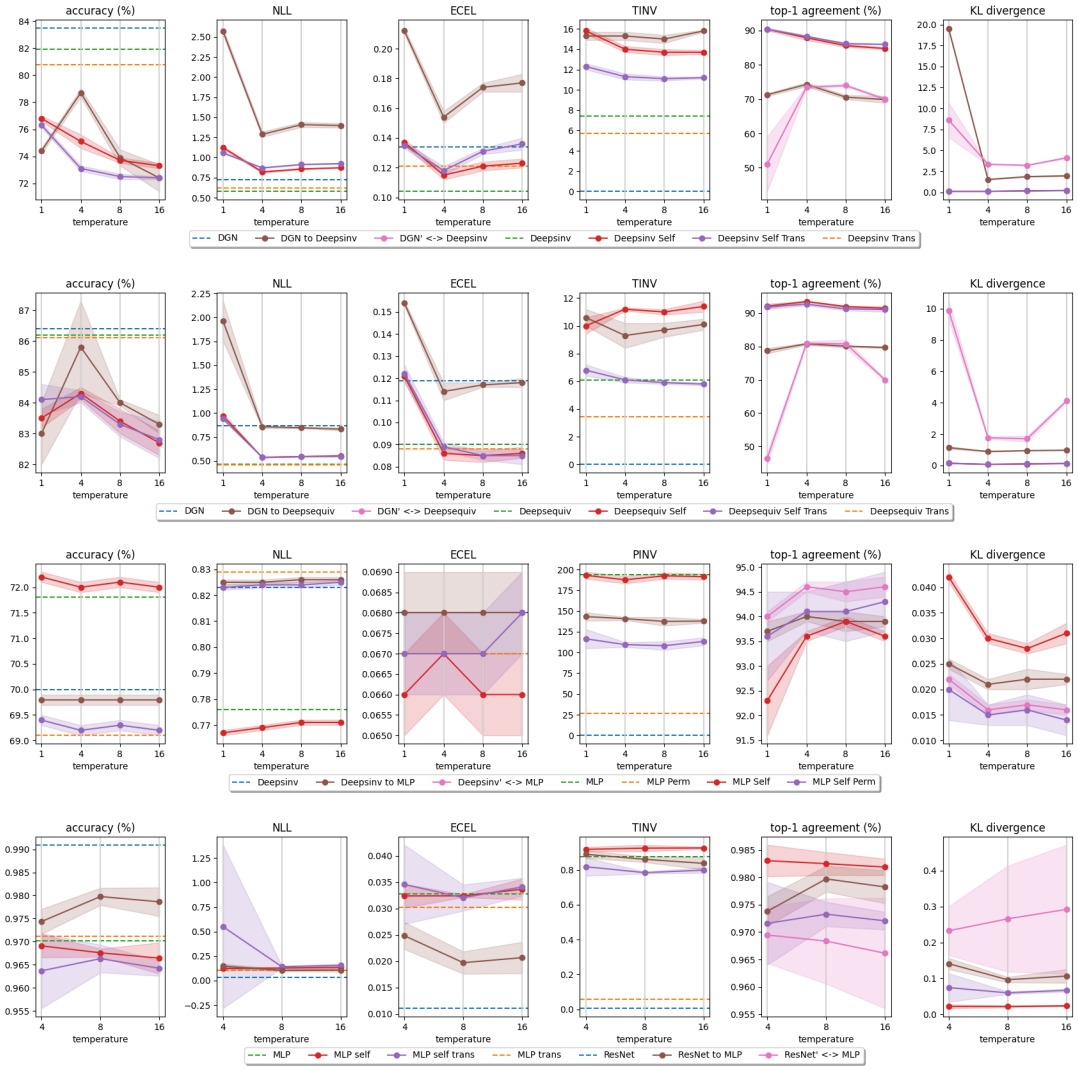

Figure 1: Summary of the attempts to transfer invariances using knowledge distillation. Each row corresponds to a distillation experiment, from top to bottom: distilling a DGCNN to an invariant DeepSets on ModelNet data, distilling a DGCNN to an equivariant DeepSets on ModelNet data, distilling an invariant DeepSets to an MLP, and distilling a ResNet to an MLP. For the ResNet distillation, setting the temperature to 1 resulted in the MLP not learning at all. The dashed lines represent the performance on the independently trained models; adding "trans" or "perm" to these labels means the student is trained on an augmented data set. Solid lines represent the results of each experiment that is described in Sec. 2.3. The labels with "self" after the model name refer to self-distillation; if "trans" or "perm" is appended, the teacher model is trained on an augumented data set. Furthermore, the legend entries with a double arrow, e.g., ResNet' $<->$ MLP, pertain to the (t', s) fidelity assessment from Sec. 2.4. The uncertainties are computed by k-folding the data with $k = 5$.

References

Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep?, 2014.

Cristian Buciluǎ et al. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

Anadi Chaman and Ivan Dokmanic. Truly shift-invariant convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3773–3783, June 2021.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019.

Tommaso Furlanello et al. Born again neural networks, 2018.

Geoffrey Hinton et al. Distilling the knowledge in a neural network, 2015.

Yann LeCun et al. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

Jing Liu et al. Graph-based knowledge distillation: A survey and experimental evaluation, 2023.

Eric A. Moreno et al. Jedi-net: a jet identification algorithm based on interaction networks. *The European Physical Journal C*, 80(1), January 2020. ISSN 1434-6052. doi: 10.1140/epjc/s10052-020-7608-4. URL http://dx.doi.org/10.1140/epjc/s10052-020-7608-4.

Patrick Odagiu et al. Ultrafast jet classification at the hl-lhc. *Machine Learning: Science and Technology*, 5(3):035017, July 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad5f10. URL http://dx.doi.org/10.1088/2632-2153/ad5f10.

Utkarsh Ojha et al. What knowledge gets distilled in knowledge distillation?, 2023.

Maurizio Pierini et al. Hls4ml lhc jet dataset (150 particles), January 2020. URL https://doi.org/10.5281/zenodo.3602260.

Samuel Stanton et al. Does knowledge distillation really work?, 2021.

Gregor Urban et al. Do deep convolutional nets really need to be deep and convolutional?, 2017.

Yue Wang et al. Dynamic graph cnn for learning on point clouds, 2019.

Zhirong Wu et al. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

Manzil Zaheer et al. Deep sets, 2018.