Knowledge Distillation for Teaching Symmetry Invariances

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Knowledge distillation is used in an attempt to transfer model invariances related to specific symmetry transformations of the data. To this end, a model that exhibits such an invariance at the structural level is distilled into a simpler model that does not. The efficacy of knowledge distillation in transferring model invariances is empirically evaluated using four pairs of such networks, each pertaining to a different data invariance. Six metrics are reported; these determine how helpful the knowledge distillation is in general for the learning process and also specifically for learning the targeted invariance. It is observed that knowledge distillation fails at transferring invariances in the considered model pairs. Moreover, data augmentation shows a better performance at instilling invariances into a network.

1 Introduction

Large neural networks are able to learn data representations that generalize well. Thus, deep learning has been an essential element in overcoming many difficult tasks in a wide range of fields, from natural language processing [Vaswani et al., 2023, Devlin et al., 2019, Brown et al., 2020] to medicine [Waring et al., 2020], biology [Jumper et al., 2021], physics [Qu and Gouskos, 2020, Pata et al., 2023, Woźniak et al., 2023], and further beyond [Alzubaidi et al., 2021]. The development of recent techniques [Ioffe and Szegedy, 2015, He et al., 2019, Bronstein et al., 2021] enables the training of large models with thousands of layers on powerful GPU or TPU clusters. Nevertheless, the computational complexity and size of such models make their deployment in real-time applications an extremely difficult challenge. 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. In essence, a small network could be capable of invariance with respect to a certain symmetry in the data, although it is not able to learn this invariance by training directly on the data itself. Thus, we consider knowledge distillation.

The seminal work of Buciluǎ et al. [2006] originally showed that the knowledge acquired by a large ensemble of models can be transferred to a relatively small model through a process called *model compression*. Furthermore, the paper by Hinton et al. [2015] expands on the idea of model compression, 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.

Submitted to Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

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.

2 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 whether KD simply has a dominant regularising effect. Hence, our study fits this literature gap.

3 Models and Methods

3.1 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]. The output of neural networks is typically class probabilities, obtained by applying a softmax function to the network's output logits. Incorporating temperature in the softmax function is a technique used to make the output probability distribution of the network smoother. The student model minimizes both the conventional task-specific loss and a distillation loss; the former quantifies the difference between the softened probability distributions of the teacher and student models. The task-specific loss ensures that the student can perform the primary task accurately, while the distillation loss encourages the student to replicate the teacher's probability distribution, thus learning to generalise in the same way. The conventional distillation loss function introduced in Hinton et al. [2015] is

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

where $\mathcal{H}$ refers to the cross-entropy, $\alpha \in [0, 1]$ is a tunable parameter, $\mathbf{y}_{\mathrm{true}}$ 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:

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

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

3.2 Data, Teachers, and Students

First, the MNIST [Deng, 2012] data set is used with ResNet18 from 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, but is smaller. Thus, with this setup we evaluate whether, to some degree, the translationally 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 chosen as the teacher. Two different DGCNN teachers are trained, DGN and DGN', the first with the hyperparameters of Wang et al. [2019] and the second with only two edge convolutional layers instead of four. We use two students for each DGCNN: a permutation invariant DS, *dsinv*, and a permutation equivariant DS, *dsequiv*, identical Zaheer et al. [2018] App. H. Thus, we evaluate what degree of *translation invariance* is distilled from the DGCNN to the DS.

The last set of invariance distillation experiments is performed on physics data [Pierini et al., 2020]. For details on this data, see Moreno et al. [2020]; the 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 as 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.

3.3 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. 3.4. 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 and compare with KD.

3.4 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). The $\mathbf{NLL}$ is used alongside the accuracy comprehensively assess the model's predictions, while ECEL is used to assess alignment of predicted and observed probabilities.

Aside from using the metrics above 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, like in [Stanton et al., 2021]. Interpreting the fidelity metrics requires additional care. Consider a student that has high fidelity: it is unclear if this student agrees with the teacher on most samples because it simply generalises well or because it actually learned to generalise in the same way as the teacher. Alternatively, it is unclear if the student learned the teacher's solution or it learned just a better solution than its independently trained counterpart due to regularisation imposed through the process of knowledge distillation itself. To control for this confounding, we repeat the distillation process (t, s) as described in Sec. 3.2 with a different teacher t' but the same student architecture, called s'. If (t, s) and (t, s') have the same fidelity, then it means that s has a high fidelity because it generalises well, rather than the reverse.

Additionally, the invariance under certain symmetries is evaluated for all of the models resulting from the experiments described in Sec. 3.2 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.

4 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 true for high-fidelity students, as shown in columns 5 and 6.

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.

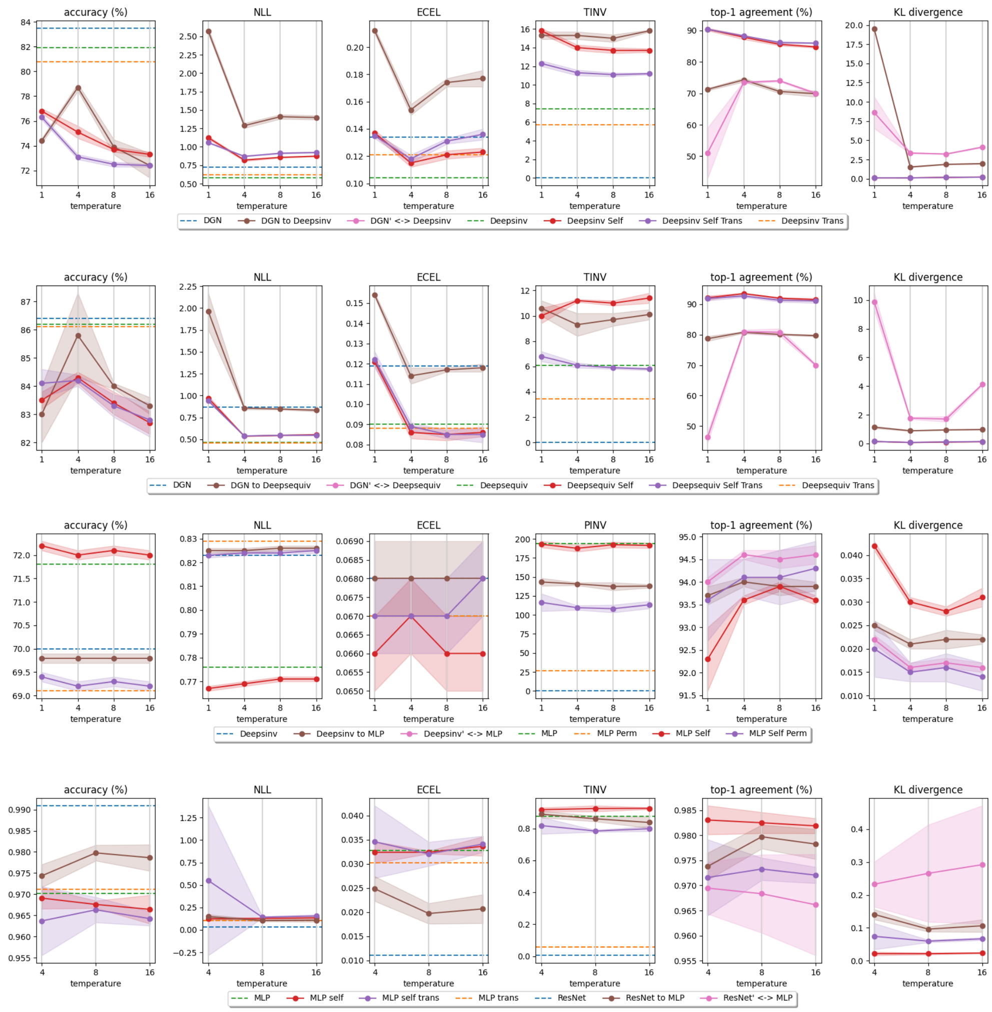

Figure 1: Summary of the attempts to transfer invariances using knowledge distillation. Each row corresponds to a distillation experiment, from top to bottom: distilling a DGCNN to an invariant DeepSets on ModelNet data, distilling a DGCNN to an equivariant DeepSets on ModelNet data, distilling an invariant DeepSets to an MLP, and distilling a ResNet to an MLP. The temperature axis refers to the temperature used in the distillation process described in Sec. 3.1. For the ResNet distillation, setting the temperature to 1 resulted in the MLP not learning at all and hence, we omit this point from the plots. Columns represent the validation metrics introduced within Sec. 3.4. The dashed lines represent the performance on the independently trained student or teacher models; adding "trans" or "perm" to these labels means the student was trained on a data set augmented with respect to its symmetry. Solid lines show the KD performance and represent the results of the experiments described in Sec. 3.3. The legend labels with "self" after the model name refer to self-distillation; if "trans" or "perm" is appended, the self-distillation teacher model is trained on augumented data. Furthermore, the legend entries with an apostrophe on the teacher and double arrow, for example ResNet' $<->$ MLP, pertain to the (t', s) fidelity assessment from Sec. 3.4. The uncertainties on the results are computed by k-folding the data with $k = 5$.

References

Ashish Vaswani et al. Attention is all you need, 2023.

Jacob Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Tom B. Brown et al. Language models are few-shot learners, 2020.

Jonathan Waring et al. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial intelligence in medicine*, 104:101822, 2020.

John Jumper et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873): 583–589, 2021.

Huilin Qu and Loukas Gouskos. Jet tagging via particle clouds. *Physical Review D*, 101(5), March 2020. ISSN 2470-0029. doi: 10.1103/physrevd.101.056019. URL `http://dx.doi.org/10.1103/PhysRevD.101.056019`.

Joosep Pata et al. Machine learning for particle flow reconstruction at cms. *Journal of Physics: Conference Series*, 2438(1):012100, February 2023. ISSN 1742-6596. doi: 10.1088/1742-6596/2438/1/012100. URL `http://dx.doi.org/10.1088/1742-6596/2438/1/012100`.

Kinga Anna Woźniak et al. Quantum anomaly detection in the latent space of proton collision events at the lhc, 2023.

Laith Alzubaidi et al. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

Fengxiang He et al. Why resnet works? residuals generalize, 2019.

Michael M. Bronstein et al. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

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

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

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

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

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.

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

Manzil Zaheer et al. Deep sets, 2018.

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

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

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

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

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

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.

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.

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

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

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`.