Efficient AI: Bridging the Gap with Knowledge Distillation Techniques

snallape@buffalo.edu, 50593866

Sharanya Nallapeddi

Deep neural networks have become extremely popular to solve many complex/ different tasks, from object detection in images using object detection models to text prediction using GPT models. Deep learning models are, however, typically large and expensive, thus, they are hard to deploy on devices with limited resources like mobile phones, embedded systems, etc. Knowledge distillation is an approach that can remedy this issue by compressing a large, complex neural network into a smaller, less complex one without compromising on its performance. The use case below would improve our comprehension of the problem.

Consider working in a hospital where a cutting-edge artificial intelligence model (the "teacher") can read medical scans (MRIs, X-rays, etc.) with high accuracy. However, this model is large and requires powerful GPUs to operate. Our task is to create a smaller AI model, the "student", which can be deployed on phones in remote clinics with limited processing resources. By copying the "knowledge" from the big teacher model to a small student model, knowledge distillation makes it useful without losing (most of) the teacher's performance.

Knowledge distillation: What is it?

Hinton et al. originally presented knowledge distillation in 2015. The concept has since drawn a lot of interest from the scientific community. The concept behind knowledge distillation is that while an intricate neural network is trained to make accurate prediction, it is trained to gain meaningful and beneficial representations regarding the data. Such meaningful representations get captured by the units in the neural network and can be named "knowledge" that is accumulated by the network while training.

The process of transferring the learned knowledge from a large, well-trained model (here, we are referring to the teacher) to a smaller, more compact model (referred to as the student) is known as knowledge distillation (shortly, KD). The student essentially learns how the teacher "thinks" by attempting to replicate the teacher's outputs.

1. Teacher Model: A highly capacity large network that is generally highly accurate but slow or resource-intensive.

2. Student Model: Smaller network, decreased capacity, higher inference speed, and lower resource needs.

How Does Knowledge Distillation Operate?

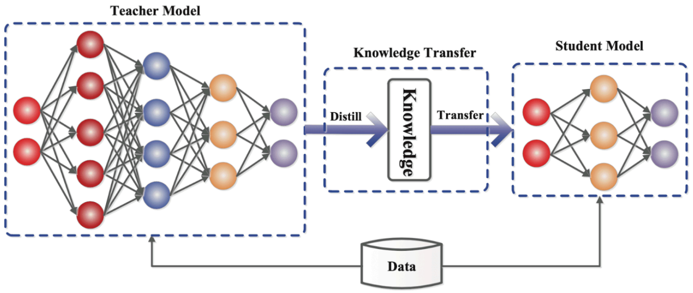
Knowledge distillation is training both student network and teacher network in two steps.

In the first stage, a teacher network—a dense and rich neural network—is developed using a traditional training procedure on a large dataset. Instead of using hard training examples, the teacher network generates "soft" training examples in the probability distribution of the courses. Soft labels show ambiguity and uncertainty in the instructor network's predictions and provide more information than hard labels.

In the second phase, student network, proportionally smaller in capacity to that of teacher network, is trained over the same dataset by leveraging the soft labels generated by the teacher network. Student network is trained to minimize difference between student outputs and the soft labels generated by the teacher network.

The intuition is that softer labels have more information regarding prediction by the student network and regarding input examples than hard labels. So, student network can recover this added information better and can generalize to novel examples. As shown in Figure 1, student modeling learns to reproduce an over-sized "teacher" model to draw upon in order to match or exceed the accuracy.

Fig 1



Efficient AI: Bridging the Gap with Knowledge Distillation Techniques

snallape@buffalo.edu, 50593866
Sharanya Nallapeddi

Suppose that you have a large neural network which is incredibly good at diagnosing medical X-ray images but too slow to implement in an busy clinic. You want to reduce that large model to a more manageable but still uniformly good-sized model with knowledge distillation. Reading about published research articles and papers presenting actual use cases will show you how others overcome comparable challenges.
Following are some suggested articles
and papers illustrating how KD has assisted in solving several challenges in neural networks, particularly when it comes to model efficiency, deployment limitations, and generalization.

- *Distilling the Knowledge in a Neural Network* - https://arxiv.org/abs/1503.02531
- *Do Deep Nets Really Need to be Deep?* - https://arxiv.org/abs/1312.6184
- *Patient Knowledge Distillation for BERT Model Compression (BERT-PKD)* - https://arxiv.org/abs/1908.09355
- *Born-Again Neural Networks* - https://arxiv.org/abs/1805.04770
- *DistilBERT: A Distilled Version of BERT* - https://arxiv.org/abs/1910.01108

Knowledge Distillation Types

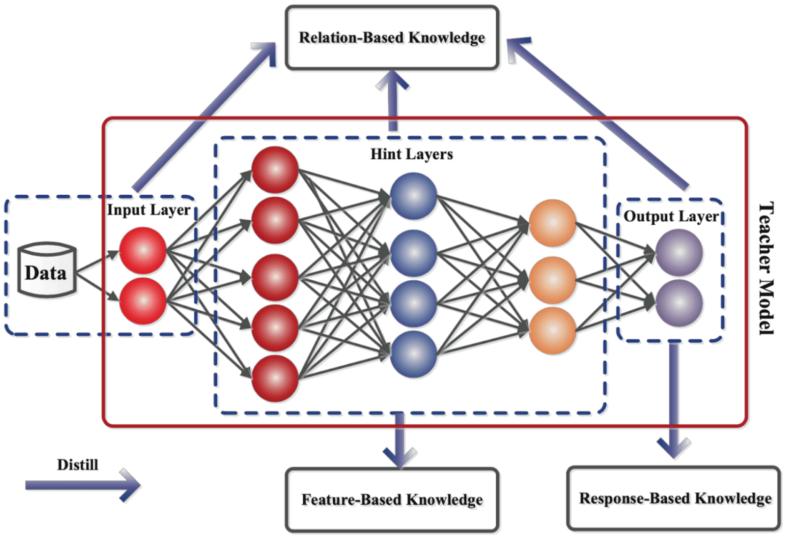
As shown in Fig 2, depending on the process of obtaining the information from the instructor model, there are various types of knowledge distillation.

In general, there have been three categories of knowledge distillation, and each one of them conveys the knowledge from the instructor model to the student model in another way. There are three forms of distillation: response based, feature based, and relation-based.

1. Response based

In response-based knowledge distillation, the student model is trained to mimic the teacher model's predictions by minimizing the difference between predicted outputs. The teacher model generates soft labels, i.e., probability distributions over the classes, for every input example during distillation. The student model is then trained to predict the identical soft labels as the teacher model by minimizing a loss function measuring the difference between their predicted outputs.

Fig 2



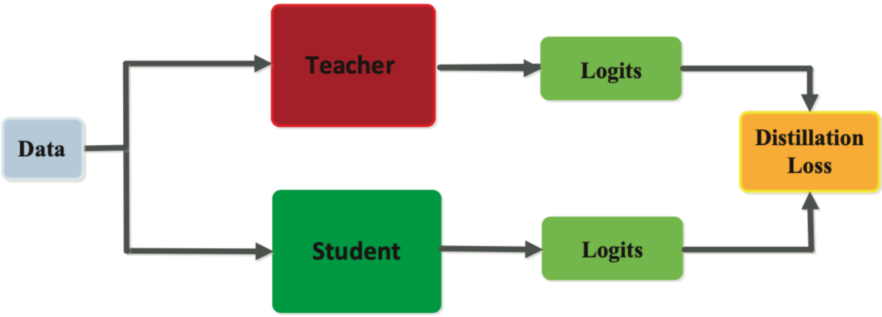
Response-based distillation is being widely applied in a variety of machine learning domains, including image classification, natural language processing, and speech recognition.

Response-based knowledge distillation is essentially useful in cases when the teacher model has an large number of output classes, where it would be computationally infeasible to train a student model from the beginning. Using response-based knowledge distillation, the student model has the ability to learn to mimic the behavior of the teacher model without having to learn any complex decision boundaries between all of the output classes that are available.sOne of the main advantages of response-based knowledge distillation is that it is easy to apply. Since the approach only requires the teacher model's predictions and soft labels, it can be applied to any models and datasets.

In addition, response-based distillation can be utilized to reduce a model's computational expenses of running by compressing it into a smaller, simpler model. Despite all of these, response based knowledge distillation still has its limitations. This approach can only transfer the knowledge related to the predicted outputs of the teacher model and has no ability to capture the internal representations which were learned by the teacher model. As it is, it may not be suitable for tasks requiring more complex decision-making or feature extraction.

Fig 3 gives a detailed overview on response based distillation type.

Fig 3

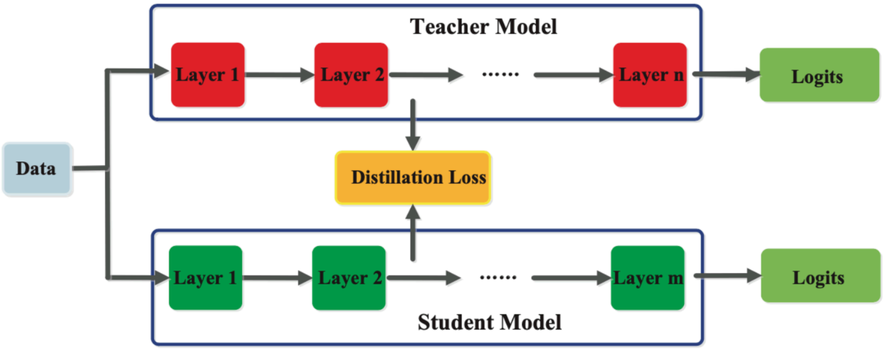


2. Feature-based

In feature-based knowledge distillation, Fig 4, the student model is usually trained to mimic the internal representations or the features learned by the teacher model, which is the primary source. The internal representations of the teacher model are extracted from one or many intermediate layers of the model, which are then used as targets for the student model(child of the teacher model). In distillation, the teacher model is first trained on the training data to learn the features related to the specific tasks that are relevant to the task being performed. Then the student model is trained to learn the exact same features by reducing the distance between the features learned by the student model as well as those learned by the teacher model. This is typically done with a loss function that measures the distance between the teacher and student model learned representations, e.g., the mean squared error or the Kullback-Leibler divergence.

One of the main advantages of feature-based knowledge distillation is that it has the ability to help the student model learn more informative and robust representations than it would be able to learn from the start. This is because the teacher model has already discovered the most relevant and informative features from the dataset, which can be transferred to the student model through the known distillation process. Additionally, feature-based knowledge distillation can be applied to a wide range of tasks and models. Therefor, it can be classified as a general model. However, feature-based knowledge distillation has its own disadvantages. This technique is computationally more expensive than other types of knowledge distillation techniques, as it requires extracting the internal representations of the teacher model at each iteration. Additionally, a feature-based approach may be unsuitable for tasks where the internal representations of the teacher model are not transferable or meaningful to the student model.

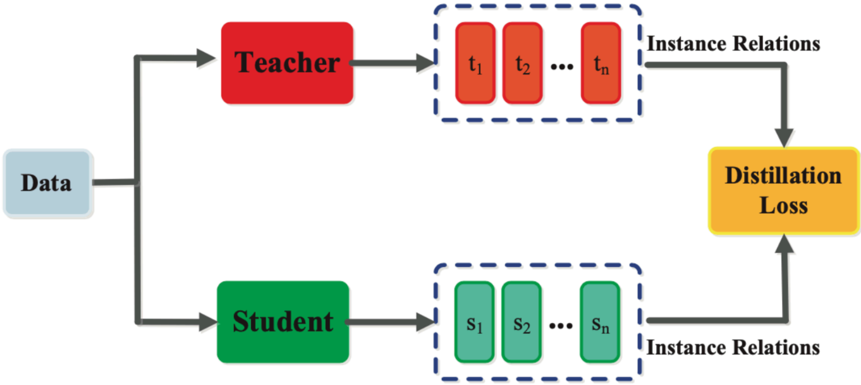
Fig 4



3. Relation-based

The relation-based distillation attempts to learn a student model to discover the relationship between output labels and the input samples of relation-based distillation. Feature-based distillation attempts to move the intermediate features that the teacher model has acquired to the student model, but relation-based distillation attempts to move the internal relationships between inputs and outputs to the student model.

Fig 5

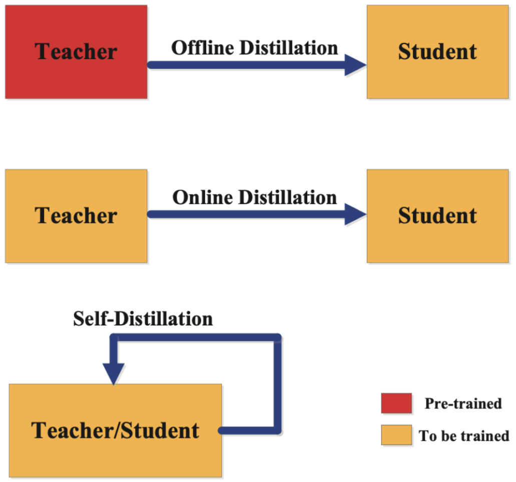


In order to represent the relationships between the input instances and the output labels, the teacher model first generates a series of relationship matrices, also known as tensors. By minimizing a loss function that quantifies the discrepancy between the connection matrices or tensors produced by the teacher model and those anticipated by the student model, the student model is then trained to learn the same relationship matrices or tensors. One of the primary benefits of a relation based knowledge distillation is its ability to enable the student model to learn a stronger and more generalizable relationship between the input instances and output labels than it could on its own. Its because the student model can benefit from the distillation process because the teacher model has already learned the most useful relationships between the inputs and outputs from the dataset thats available to perform analysis on.

Even though it might be computationally costly to build the relation matrices or tensors, especially for the large datasets. Also, problems where relationships between the input examples and the output labels are uncertain or hard to encode into a matrix or set of matrices or tensors may not be good candidates for a relation-based method.

Techniques for Distillation Knowledge Training

Fig 6



From Fig 6, assume that we have an extensive pre-trained language model (such as BERT) that is computationally infeasible to apply in mobile phones. We would rather have an equivalent-sized model slightly inferior in performance but faster in memory usage and speed. This can be done in various ways by various methods of distillation. Consider following methods:

Offline Distillation

- Firstly, we thoroughly train your giant teacher model, such as your big BERT.
- Next, one would use the teacher's outputs or hidden representations as "guidance" to train the smaller student model in a different procedure (offline).
- Here, we have to understand that the teacher model is fixed and already taught.

Online Distillation

- Usually, it begins with training both the teacher and the pupil simultaneously, or partially simultaneously.
- Students use the outputs and presentations that the teacher creates as they learn to get better right away.
- We need to note that the students and teachers can participate in a cooperative training loop and update jointly.

Self Distillation

- A single model that functions as both teacher and student (sometimes via separating layers or copies of itself) is used in place of two distinct models (teacher vs. student).
- The later (or current) portions of the network are taught by the earlier layers or checkpoints as the model trains.
- It's important to remember that it's basically "distilling knowledge from itself," which can result in regularization or better performance without requiring a sizable, independent teaching model.

Knowledge distillation algorithms

Below are the algorithms for training students models to gain knowledge from the teacher models.

Adversarial distillation

A generator is trained to produce samples most similar to true data distribution by undergoing an adversarial training process by both an adversary generator model that learns to distinguish true samples from fake samples and an adversary discriminator model that learns to detect fake samples from true samples. This has been used in an attempt to expose both student and teacher models to better true data distribution.

To meet this goal to get to know the original distribution of the data, an adversary can be employed to train a generator model to get fake training data to be used directly or to be used to supplement the original training dataset. Adversarial learning is used in an alternate technique where attention is put to a discriminator model to distinguish between student and teacher model-generated samples by logit or by feature map. This technique is used to approximate the student in an efficient manner. Adversarial learning-based is used in an alternate technique where attention is put to online distillation where student and teacher models get optimizes in an interactive manner.

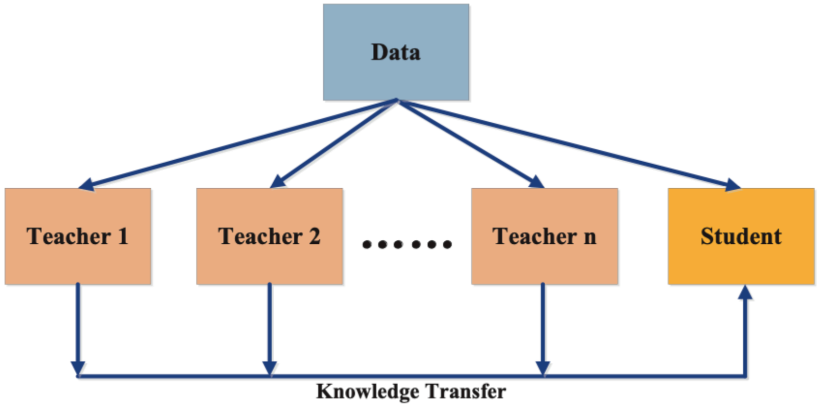
Multi-Teacher distillation

Multi-teacher distillation, Fig refers to when more than one teacher model is used in training a single student model. The motivation for multi-teacher distillation being that from multiple sources of information, the student model learns a more complete set of features and thus performance improves.

The multi-teacher distillation algorithm relies on two phases of training. Initially, the teacher models are individually trained on the training set to receive their outputs. First, all teacher models are trained individually on the training set. Second, the student model is trained on the same training set, with all teacher models' outputs as targets. The student model learns from all teacher models' outputs during training, with each teacher model giving a different view of the training set. The student model is able to learn a more complete set of features in this way, resulting in better performance.

Multi-teacher distillation has various advantages over common knowledge distillation strategies. It decreases the biasness that a solitary teacher model is likely to give as there exist numerous teachers and each presents distinctive perspectives. Multi-teacher distillation improves student model stability since it acquires knowledge from differing sources of understanding.

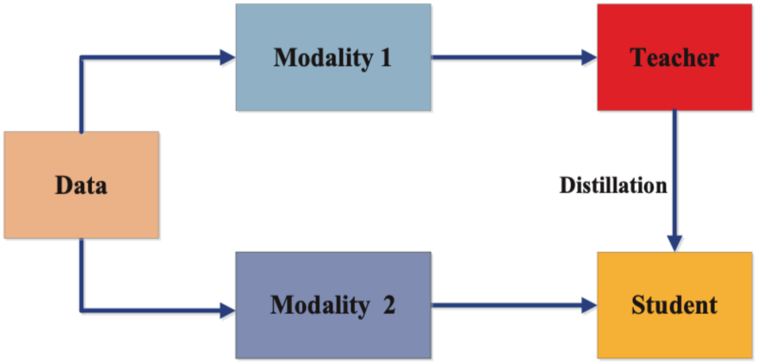
Fig 7



Cross-Modal distillation

Cross-modal distillation training paradigm is presented in Figure 8. In this case, the student needing knowledge from another modality learns the teacher's knowledge based on their own training in the other modality. This is a scenario when certain modalities have no data or labels to train or test and therefore need transfer of knowledge across modalities. The visual modality is the most prevalent area where cross-modal distillation is applied. Lets take an example to explain this in a better way, a student model with an unlabeled input domain, such as optical flow, text, or speech, can be aided by distillation based on the expertise of a teacher that is trained on labeled image data. The student model is supervised-trained with features that the teacher model learned from the pictures in such a scenario.

Fig 8



Apart from the above discussed distilled algorithms, there are several others which can be applied to knowledge distillation, please see the below listed.

- Graph-based distillation
- Attention based distillation
- Data free distillation
- Quantized distillation
- Lifelong distillation
- Neural architecture search based distillation
- Residual/Layer-wise distillation
- Progressive/Iterative distillation
- Noisy/Stochastic Distillation

- Self-Ensembling/Co-Distillation
- Variational Distillation
- Early-Exit (or Multi-Exit) Distillation

Knowledge distillation is an approach which is helpful to improve the performance of miniature models by knowledge transfer from huge and complex models. Knowledge distillation is beneficial in a wide range of applications, this includes, computer vision, natural language processing, and speech recognition.

There are three main categories of distillation knowledge technique: offline, online, and self-distillation. These are utilized depending on whether the teacher model is being fine-tuned or not during training. There are various knowledge distillation algorithms, including adversarial distillation, multi-teacher distillation, and cross-modal distillation, with each having a particular method of knowledge transfer from student to teacher.

Knowledge distillation generally provides a useful mechanism for improving both the performance as well as efficiency of machine learning models. There is significant current research and development in this direction, with significant potential applications in the field of Artificial Intelligence.

We have created a code that includes the information below for a more thorough understanding of knowledge distillation approaches.

The code implementation's objective

- Our goal is to use Knowledge Distillation to train a large (teacher) neural network and a smaller (student) network on the CIFAR-10 dataset. The popular CIFAR-10 image classification dataset consists of 10 classifications (such as cars, airplanes, and birds) with 32x32 color images in each class.
- Demonstrate how a student, who is a smaller and more efficient model, may learn to perform almost as well as the teacher, who is a larger and more accurate model.
- Examine the differences in student performance between knowledge distillation and not.

Technical Concepts, Steps, and Results

- **Data Loading**

Here, we have used torchvision.datasets to load CIFAR-10. CIFAR10, using common transformations (such as normalization and ToTensor()).
Once that is done, this dataset is divided into training and testing sets and processed in batches using DataLoaders.

- **Defining Models**

a. Teacher Model (DeepNN) is created, which is a larger CNN with more channels and layers.
b. Student Model (LightNN) is created, which is a smaller CNN with fewer parameters.

- **Training the teacher**

a. train(teacher, trainLoader,...): Here, we have applied selective number of epochs for training the instructor.
b. Training is further guided by a CrossEntropyLoss, and the training loss for each epoch is eventually printed out.

- **Testing the teacher**

a. We would be assessing the teacher's accuracy using the CIFAR-10 test set using test(teacher, testLoader).
b. For above, we estimate the outputs, which is, accuracy of teachers (e.g., approximately 74% to 75% depending on your epochs and hyperparameters that are used).

- **Training the student**

a. The student is initially trained on CIFAR-10 using normal supervised learning without implementing any techniques of knowledge distillation.

b. Later, we would be determining the test accuracy after step a is completed, result is less than that of the teacher as it has fewer parameters.

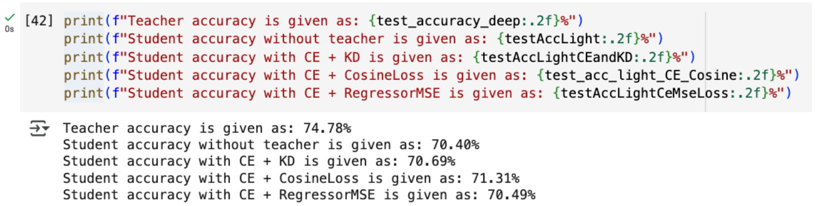
- **Knowledge Distillation**

1. Distillation Setup: Firstly, we would build a function (trainKD) that combines two kinds of losses to train the student, please see below:
a. Standard cross-entropy against ground truth labels, or label loss.
b. KL-divergence-like term distillation loss which will measure the difference between soft forecasts of a teacher and students normalized by temperature T.
2. Weights (ce_loss_weight against soft_target_loss_weight): Adjust the student's reliance on the teacher's soft outputs in relation to the real facts.
3. When you evaluate the student's test accuracy following distillation training, you see that they have improved in comparison to the baseline student (it frequently approaches the teacher's accuracy).

- **Additional Models (ModifiedDeepNNCosine / ModifiedLightNNCosine)**

1. Variants of the teacher & student can be made that also perform certain operations (e.g., avg_pool1d) or yield concealed representations.
2. This can be used for more investigation, like making intermediate feature maps or comparing cosine distances.

- **Outputs and Observations**



```
[42] print(f"Teacher accuracy is given as: {test_accuracy_deep:.2f}%")
     print(f"Student accuracy without teacher is given as: {testAccLight:.2f}%")
     print(f"Student accuracy with CE + KD is given as: {testAccLightCEandKD:.2f}%")
     print(f"Student accuracy with CE + CosineLoss is given as: {test_acc_light_CE_Cosine:.2f}%")
     print(f"Student accuracy with CE + RegressorMSE is given as: {testAccLightCeMseLoss:.2f}%")
```

```
Teacher accuracy is given as: 74.78%
Student accuracy without teacher is given as: 70.40%
Student accuracy with CE + KD is given as: 70.69%
Student accuracy with CE + CosineLoss is given as: 71.31%
Student accuracy with CE + RegressorMSE is given as: 70.49%
```

Since none of the previously described methods increase the amount of network parameters or the inference time, the performance gain comes at the slight cost of computing gradients during training. In machine learning applications, inference time is of the utmost importance since training is completed before model deployment. In the event that our light model is still too large for deployment, we can also use post-training quantization. Apart from adjusting variables such as temperature, number of neurons, or coefficients, you can also experiment with adding more losses to certain jobs.

References

- Chao, X., Xu, Z., Ba, J., & Caruana, R. (2020). *A Comprehensive Study on Model Compression and Acceleration*.
arXiv:2003.11316
- https://www.ibm.com/think/topics/knowledge-distillation
- Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: Proceedings of the International Conference on Learning Representations (2015)
- https://www.v7labs.com/blog/knowledge-distillation-guide
- https://snorkel.ai/blog/llm-distillation-demystified-a-complete-guide/