Article Analysis :Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons

Ghassen Chaabane, Moez Ezzeddine, Gaspard Donada–Vidal, Wafa Bouzouita

Abstract

In recent years, fueled by progress in machine learning, we have seen amazing leaps in the use of artificial intelligence. However, these applications need a lot of data and their performance relies extremely on the quality and quantity of the training data. This issue makes the machine learning models require a lot of time and computational resources. Inspired by the human capacity to transfer knowledge, the machine learning community has turned its focus on the knowledge transfer to overcome these issues.

Knowledge transfer methods consist in transferring information from a previously learned network (teacher) to a new one (student), in order to improve the performance of the latter.

Two main applications in knowledge transfer include transfer learning between two networks of identical architecture or from a large network to a small network.

Previous works have focused on transferring knowledge via neuron responses of hidden layers. The present paper proposes a new knowledge transfer method through the distillation of activation boundaries formed by hidden neurons ; where boundary activation refers to the hyperplane that separates activated and deactivated neurons.

1. Proposed method

To transfer the responses of the hidden layer neurons of a teacher network to a student, the network initialization method is used. The network is firstly initialized using the transfer loss, and it is then trained for classification thanks to the cross-entropy loss.

1.1. Transfer loss

The proposed activation transfer loss aims to minimize the difference of neuron activations between the teacher and the student. It considers whether the neuron is activated or not. To define the transfer loss, we consider the following terms :

— $\mathcal{T}$: the portion of teacher network, from the input image to a hidden layer.
— $\mathcal{S}$: a portion of student network.

For an image $\mathbf{I}$, the neuron response vector of the hidden layer before the activation function, is $\mathcal{T}(\mathbf{I}) \in R^M$ for the teacher and $\mathcal{S}(\mathbf{I}) \in R^M$ for the student. Where M is the number of neurons in the hidden layer.

When using the ReLU function as activation function, the activation boundary position is between weak neuron responses and strong neuron responses. It is then difficult to select the activation boundary accurately and thus the decision boundary for classification of the feature space in the hidden layer. In order to accurately transfer the activation boundaries, this article proposes to amplify the negligible transfer loss at the region near the activation boundaries, by using an element-wise activation indicator function. This function expresses whether a neuron is activated or not, i.e.,

$$\rho(x) = \begin{cases} 1, & if \ \ x > 0 \\ 0, & otherwise \end{cases}$$

Then transfer loss is given by :

$$\mathcal{L}(\mathbf{I}) = ||\rho(\mathcal{T}(\mathbf{I})) - \rho(\mathcal{S}(\mathbf{I}))||_1$$

This transfer function is not differentiable. It cannot be minimized by the gradient descent method. In order to obtain a differentiable loss with the same purpose, this article proposes an alternative loss similar to the hinge loss of SVM, i.e.

$$\mathcal{L}(\mathbf{I}) = ||\rho(\mathcal{T}(\mathbf{I}))\odot\sigma(\mu\mathbf{1}-\mathcal{S}(\mathbf{I}))+(1-\rho(\mathcal{T}(\mathbf{I})))\odot\sigma(\mu\mathbf{1}+\mathcal{S}(\mathbf{I}))||_2^2$$

Minimizing this alternative loss ensures that the neuron of the teacher and student has the same activation, and puts a margin to the studentâs neuron response.

1.2. Number of neurons

Previously, we presented the case where the number of neurons in the teacher and the student are the same. However, if the teacher's and student's architectures are different, as in the case of network compression, the number of neurons could be different. In this case, this work uses a connector function defined as $r : R^N \rightarrow R^M$. Where M

and N are the number of neurons in the teacher and student networks, respectively. This function resizes a neuron's response vector of the student to make it the same size as a teacher's vector. Using the connector function, the alternative loss is modified :

$$\mathcal{L}(\mathbf{I}) = ||\rho(\mathcal{T}(\mathbf{I})) \odot \sigma(\mu\mathbf{1} - r(\mathcal{S}(\mathbf{I}))) + (1 - \rho(\mathcal{T}(\mathbf{I}))) \odot \sigma(\mu\mathbf{1} + r(\mathcal{S}(\mathbf{I})))||_2^2$$

Usually, the connector function is a completely connected layer or a combination of a fully connected layer and a batch normalization layer. During the initialization phase, the connector function r and the student network S are trained simultaneously to minimize the alternative loss. After initialization, only the student network is trained for classification.

2. Experiments

Extensive experiments were performed to verify the effectiveness of the proposed method. The first experiment is about various aspects of knowledge transfer. This work shows that the proposed method outperforms the current state-of-the-art in most aspects. The second experiment is about evaluating the proposed method in transfer learning in order to substantiate its performance in more general configurations. Eventually, numerous empirical experiments have been performed to help understand the proposed method.

2.1. Knowledge transfer for the same task

In this part, experiments were assessed on the CIFAR10 dataset. The base network architecture is the wide residual networks (WRN) which can change depth and the number of channels. To specify the type of **WRN**, we refer to it, by WRNx-y, where x is the number of layers in the network, ie. depth, and y is the multiplication factor of the channels. For instance, WRN16-4 is composed of 16 layers and is using 4 times the number of base channels. The Knowledge Distillation (KD) loss is used as baseline. The result of the training without any transfer of knowledge has been also presented.

First, experiment was conducted to asses the learning speed of the different algorithms. It is expected that knowledge transfer can accelerate student network training because it uses information of a pretrained network. Experiments were performed using WRN22-4 as a teacher network and WRN16-2 as a student network. In this case, the error rate of the teacher is 4.51%. The results are shown in table 1. Where the upper row of Table 1 shows the number of epochs used for network initialization and classification training. From the table 1, we notice that most knowledge transfer algorithms speed up learning of the student network. We remark too that the proposed method is better to the current state-of-the-art in terms of learning speed. Second, the current algorithms were evaluated in terms of generalization performance. Like the previous assessment, same teacher and student network were used. In this case, training data for the student network is limited to a certain percentage. For the teacher network, all the training is used and an error rate of 4.51% was obtained. Table 2 shows the results. Most knowledge transfer approaches allow the student classifier to generalize well. also, in this case, the approach suggested provides the best performance for all scenarios. Third, experiment was conducted to verify knowledge transfer for networks of various sizes. In this case, performance of knowledge transfer was assessed for various configurations of network sizes. The results are shown in table 3. All current knowledge transfer algorithms are more susceptible to depth compression than to channel compression. The proposed method still gives the best result. These three experiments indicates that the proposed method outperforms the current state-of-the-art in terms of learning speed up, generalization performance and transfer learning performance.

Table 1: Learning speed experiment using various training epochs. Table shows error rate(%) on test set.

Training epochs (initialize + training)	1+1	1+5	3+12	5+25	10+50	20+100
Cross-Entropy training	43.37%	17.72%	11.76%	8.63%	7.41%	6.47%
Knowledge Distillation (KD) training	48.42%	19.80%	12.09%	8.66%	6.80%	6.19%
FITNET (Romero et al. 2015) + KD	48.16%	19.82%	11.10%	8.38%	7.02%	6.28%
FSP (Yim et al. 2017) + KD	43.51%	19.29%	11.15%	8.48%	6.87%	6.22%
AT (Zagoruyko and Komodakis 2017) + KD	37.66%	14.14%	8.35%	6.68%	5.94%	5.80%
Jacobian (Srinivas and Fleuret 2018) + KD	38.46%	14.29%	8.37%	6.98%	5.98%	5.77%
Proposed + KD	**16.39%**	**11.16%**	**6.95%**	**6.08%**	**5.72%**	**5.58%**

Table 2: Performance for small size training data. Table shows error rate (%) on test set.

Percentage of training data	0.1%	1%	5%	10%	20%	100%
Cross-Entropy training	73.83%	48.41%	28.12%	21.76%	15.26%	6.47%
Knowledge Distillation (KD) training	74.62%	48.34%	28.13%	21.21%	14.62%	6.19%
FITNET (Romero et al. 2015) + KD	74.81%	49.91%	27.28%	21.42%	14.52%	6.28%
FSP (Yim et al. 2017) + KD	73.96%	47.98%	26.90%	20.80%	13.85%	6.22%
AT (Zagoruyko and Komodakis 2017) + KD	67.54%	37.11%	18.86%	15.61%	9.94%	5.80%
Jacobian (Srinivas and Fleuret 2018) + KD	68.65%	36.99%	18.34%	15.03%	9.83%	5.77%
Proposed + KD	**50.32%**	**21.54%**	**14.99%**	**13.09%**	**9.16%**	**5.58%**

Table 3: Performance for various sizes of networks. Table shows error rate(%) on test set.

Compression type	Teacher	Student	Size ratio	KD	FITNET	FSP	AT	Jacobian	Proposed
Depth	WRN 22-4	WRN 10-4	27.9%	22.98%	23.34%	22.99%	18.06%	18.28%	**14.05%**
Channel	WRN 16-4	WRN 16-2	25.2%	20.48%	19.98%	19.78%	14.81%	14.41%	**11.62%**
Depth & Channel	WRN 22-4	WRN 16-2	16.1%	21.21%	21.42%	20.80%	15.61%	15.03%	**13.09%**
Tiny network	WRN 22-4	WRN 10-1	1.8%	29.57%	29.18%	28.70%	29.44%	28.70%	**23.27%**
Same network	WRN 16-4	WRN 16-4	100%	18.29%	17.91%	17.81%	12.03%	11.28%	**6.63%**

2.2. Transfer learning

Here, authors focus on verifying knowledge transfer in more general datasets via transfer learning. Where transfer learning consists in achieving generalization performance without a separate pre-training on a large dataset. In this expirimental part , ResNet50 was used as a teacher network and Mobilenet was used as a student network. This two networks

have a quite different structure. As baseline, distillation-in-transfer-learning (DTL) was used. The MIT scenes dataset and the CUB 2011 dataset, were used as target datasets of transfer learning. Using these two datasets, we experimented on how knowledge transfer contributes to the generalization of the performance of classifiers. Table 4 show the results for the MIT scenes dataset and Table 5 shows the performance for the CUB 2011 dataset. The classifier performance was assessed in terms of accuracy. Here, we remark that the proposed solution gives far better performance than all the other methods in all cases, and the performance improvement increases when the number of training samples becomes smaller. We also notice that the proposed method achieves better performance than the pre-trained ImageNet network in most cases. So, the proposed solution achieves better generalization than the ImageNet pre-training.

Table 4: Performance of transfer learning on MIT scenes. Table shows accuracy (%) on test set.

Number of training data per class	5	10	20	40	80
ImageNet pre-trained network	32.39%	42.46%	52.99%	62.54%	69.78%
randomly initialized network	11.12%	17.99%	30.37%	43.88%	56.94%
DTL on randomly initialized network	20.52%	34.55%	49.78%	59.48%	64.93%
FITNET (Romero et al. 2015) + DTL	29.55%	43.81%	53.51%	63.36%	66.57%
FSP (Yim et al. 2017) + DTL	25.60%	39.63%	51.42%	60.08%	63.73%
AT (Zagoruyko and Komodakis 2017) + DTL	22.61%	35.37%	49.10%	60.15%	65.15%
Jacobian (Srinivas and Fleuret 2018) + DTL	26.64%	38.28%	51.34%	61.19%	63.96%
Proposed + DTL	**43.36%**	**56.72%**	**66.34%**	**70.75%**	**74.10%**

Table 5: Performance of transfer learning on CUB 2011 dataset. Table shows accuracy (%) on test set.

Number of training data per class	1	5	10	20	30
ImageNet pre-trained network	8.01%	29.84%	48.83%	69.02%	74.70%
randomly initialized network	2.40%	10.68%	25.54%	44.43%	56.63%
DTL on randomly initialized network	5.35%	26.29%	42.15%	48.24%	62.05%
FITNET (Romero et al. 2015) + DTL	10.70%	36.75%	50.74%	52.85%	63.93%
FSP (Yim et al. 2017) + DTL	7.06%	29.43%	42.44%	43.96%	56.52%
AT (Zagoruyko and Komodakis 2017) + DTL	6.65%	28.56%	44.43%	52.80%	63.70%
Jacobian (Srinivas and Fleuret 2018) + DTL	7.85%	31.02%	45.39%	49.41%	60.11%
Proposed + DTL	**13.38%**	**45.39%**	**57.11%**	**62.93%**	**70.54%**

2.3. Analysis

Here, authors perform other experiments to better understand the proposed method. They use WRN as the neural network model and CIFAR-10 as the dataset. In WRN, residual layers are gathered to form one layer group, and three layer groups are gathered to form a network. The last layers of each group are identified by layer 1, layer 2 and layer 3, with spatial sizes of 32×32, 16×16 and 8×8, respectively. Here, the characteristics of each layer were analysed in knowledge transfer through three analytical experiments.

The first experiment concerns the effect of layer-wise transfer. In this experiment, knowledge transfer methods are tested for a single layer to show the differences of performance in terms of hidden layer location. The number of training epochs was fixed to 120 epochs 10% of the training data. Table 6 shows the results. The table 6 shows that best results for most knowledge transfer algorithms were when transferring multiple layers, except FITNET, which shows the best performance when only layer 3 is transferred. It also shows that the proposed method gives better results for all configurations. The second experiment is to see the influence of the proposed loss norm on the activation similarity between student and teacher network. The experiment used all training data in CIFAR-10, and WRN16-4 was used for both the teacher and student networks. The experimental results are shown in Table 7.

Table 6: Layer-wise analysis of knowledge transfer. Table shows error rate (%) of classification.

	Layer1	Layer2	Layer3	Multi-layer
KD		21.21%		
FITNET + KD	21.90%	22.00%	21.42%	21.83%
FSP + KD	21.55%	21.28%	21.27%	20.80%
AT + KD	20.18%	17.64%	17.48%	15.61%
Jacobian + KD	19.52%	15.45%	23.61%	15.03%
Proposed + KD	**19.06%**	**15.37%**	**17.23%**	**13.09%**

Table 7: Comparison of activation similarity with l_p norm. Table shows percentage of same activation (%) and classification error rate (%).

	l_2 loss	l_1 loss	$l_{0.5}$ loss	Proposed
Layer 1	56.1%	66.9%	68.8%	**96.3%**
Layer 2	67.3%	72.5%	73.0%	**96.4%**
Layer 3	56.0%	56.1%	56.4%	**92.8%**
Error rate	10.1%	9.8%	9.6%	**5.1%**

The table 7 shows that l_p (p <2) losses make the activation more similar than the l_2 loss and also has better performance. It also indicates that the proposed loss gives much higher similarity and much better performance, which confirms that the proposed method is indeed a reasonable approximation of the ideal activation transfer loss.

The last experiment is a study on the margin value μ, a parameter of the proposed method. Here, the dependency of the proposed solution on the margin is assessed. Experiments were realized on CIFAR-10 with 10% training data. The experimental results are shown in Table 8. This table shows that the performance of the proposed method is not very affected but the change of the margin value. This means that the proposed method is not sensitive to the parameter μ.

Table 8: Ablation study for margin value (μ).

Margin (μ)	0.75	1	1.5	2	4
Error rate	13.9%	13.1%	12.1%	12.1%	11.7%

3. Discussion

In order to evaluate the rigor and the faithfulness of the mentioned results, and using the author's code, same experiments were made. The following sections detail the made tests and contain a brief discussion analysing and comparing the obtained results with the shared ones.

3.1. Testing

The proposed solution in this paper outperforms the-state-of-the-art in most aspects. Testing the same experiments as the author would help to better understand the limits of such an approach and ensuring its stability.

The proposed experiments aim to compare various algorithms in term of :

— Learning speed
— Dependency to dataset size
— Dependency to network size

So, and according to what was mentioned, the first three explained experiments were considered, as they are the most related to these criteria.

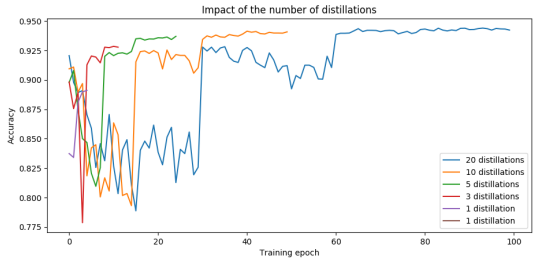
3.2. Results

The results were summarized in the following tables. Table 6 shows higher error rates than what was mentioned by the author. However, the difference average does not exceed the 0.5%. it's quite interesting to focus on the accuracy evolution during each epoch. In the following figure, each curve has two major peaks proportional to the epochs' number during distillation. It's remarkable to mention that for the same number of training epochs a system with lower epochs' number may perform better. As example, after 20 epochs of training a system with 10 distillation's epochs reaches more than 0.9% of accuracy, in return a system with 20 distillation's epochs does not exceed 0.87%. This proves that although distillation and transfer learning may increase the algorithm learning speed, the training phase of a learning network is essential and closely related to the number of epochs done in distillation phase.

Table 6. Learning speed test

D.EPOCHS*	T.EPOCHS*	FINAL ACCURACY	TEST ERROR
1	1	80.83%	19.17%
1	5	89.11%	10.89%
3	12	92.78%	07.22%
5	25	93.70%	06.30%
10	50	94.08%	05.92%
20	100	94.23%	05.77%

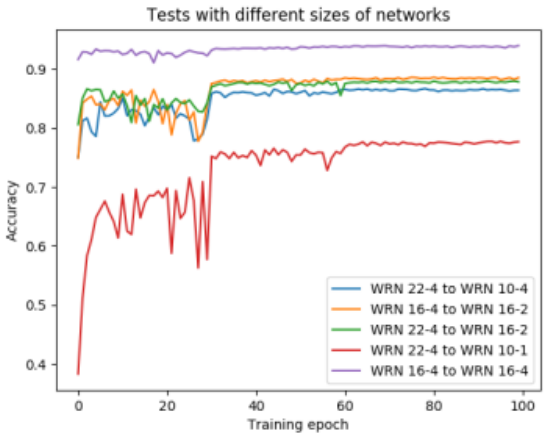
*D.epochs : Distillation epochs
*T.epochs : Training epochs



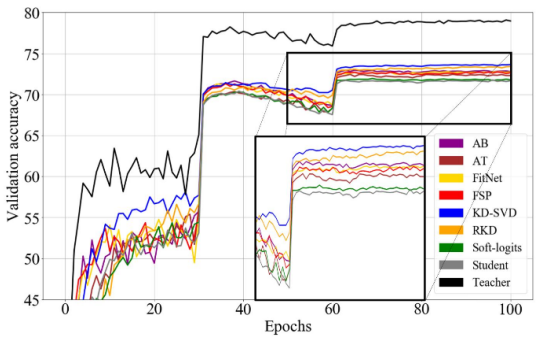
In both experiments related to performance for small dataset size and various network size, higher accuracy was obtained using the full dataset with no compression network. All curves shows a peak in the 30th training epoch which is explained by the use of the 20/100 distillation/training epochs number.

3.3. Work's limits

In this article the author proposed and defended a new approach by comparing his algorithm to the state-of-the-art existing algorithms. The main idea of the author's approach were clearly explained and detailed in his article. Besides, the experiments results were verified and they were a solid proof of the rigor and the authenticity of the study. However, other non mentioned algorithms may have better performance than what was proposed. This was proved in



"Seunghyun Lee and al." work "knowledge distillation in deep neural network"(Seunghyun Lee, 2019). As illustrated in the following figure, KD-SVD scores higher accuracy than AB algorithm.



The author believes that "We must transfer whether neurons are activated or not instead of the magnitude of their response." but how extensible is this rule ? Actually, the main focus must be on considering the knowledge to distillate and transfer.

This article was clear, interesting and instructive to read. It presents a new approach and proves its efficiency.

4. Conclusion

In the present paper, firstly, we have presented the proposed method of the article « Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons ». This work has shown through experiments, that the activation boundaries can greatly enhance performance of knowledge transfer. The experiments was about comparing the proposed method to various aspects of knowledge transfer and evaluating it in more general configurations. Secondly, we have repeated the same experiments by using author's code. We have obtained approximately the same results that confirm the authenticity of the mentioned results.

Références

Seunghyun Lee, B. C. S. Graph-based knowledge distillation by multi-head attention network. In *British Machine Vision Conference (BMVC)*, 2019.