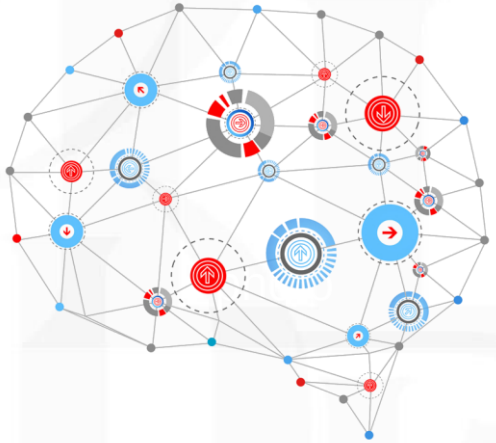
NATIONAL RESEARCH UNIVERSITY

Deep neural networks performance optimization via the Distillation the Knowledge

Rassadin A. G., Savchenko A. V.

Problem Discussion

The success of deep CNNs has started from the paper [Krizhevsky et al. 2012]. Contemporary CNN architectures are much more accurate when compared with original AlexNet. However, their runtime complexity becomes insufficient for application in several practical tasks, especially with implementation on mobile platforms. Hence, the performance optimization of deep CNNs is now considered as one of the most important studies in deep learning.



- **Pruning**

  [Han et al. 2016], [Molchanov et al. 2016]

- **Distillation The Knowledge**

  [Hinton et al. 2014], [Romero et al. 2014]

- **Weights Hashing / Quantization**

  [Chen et al. 2015], [Han et al. 2016]

- **Tensor Decompositions**

  [Lebedev et al. 2015], [Kim et al. 2015], [Novikov et al. 2015], [Garipov et al. 2016]

- **Binarization**

  [Courbariaux / Hubara et al. 2016], [Rastegari et al. 2016], [Merolla et al. 2016], [Hou et al. 2016]

- **Architectural tricks**

  [Hong et al. 2016], [Iandola et al. 2016], [Teerapittayanon et al. 2016]

Training Deep Learning models always require:

- massive labeled data and
- massive computational resources.

Meanwhile,

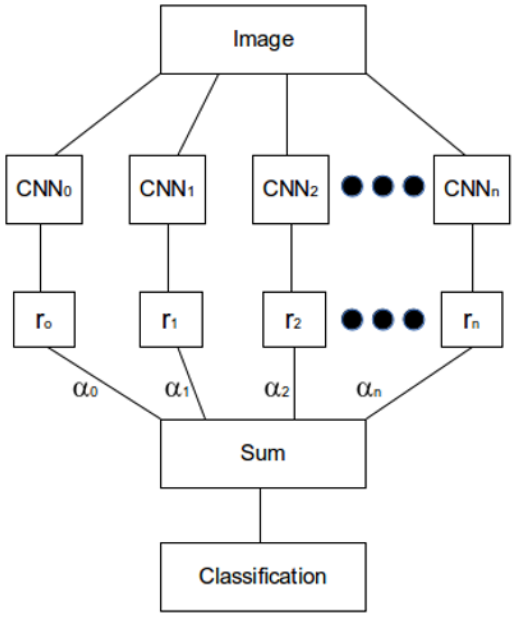
- [Hinton et al. 2014] shows the possibility to distill the knowledge from the cumbersome ensemble to the one model,
- [Tramèr et al. 2016] shows the possibility to reproduce existing model having only *labels* it predicts. Having also *scoring* for each sample (*softmax* output in our case), such reverse engineering becomes more accurate.

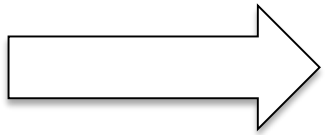
Problem Formulation

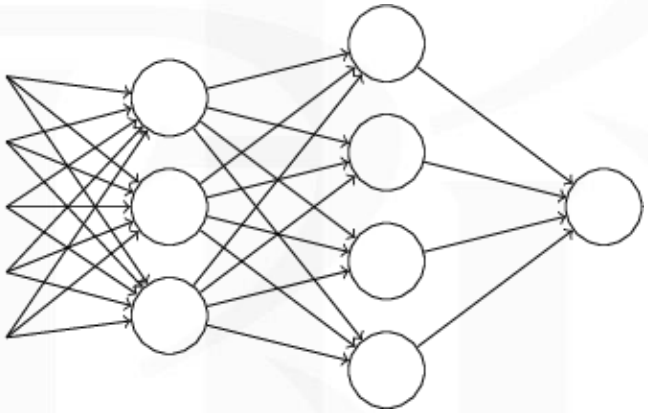
Can we train a model having

- a black-box model,
- **limited** computational resources,
- **no** labeled data?

Distillation The Knowledge







$$P_T = softmax(a_T); P_S = softmax(a_S)$$

where T – "teacher" network indicator, S – "student" network indicator

Distillation the knowledge is equal to optimization $\mathcal{L}$,

$$\mathcal{L} = \mathcal{H}(y_{true}, P_S) + \lambda\mathcal{H}(P_T, S_T)$$

where $\mathcal{H}$ is a cross-entropy.

Problem reFormulation

Can we train a model having

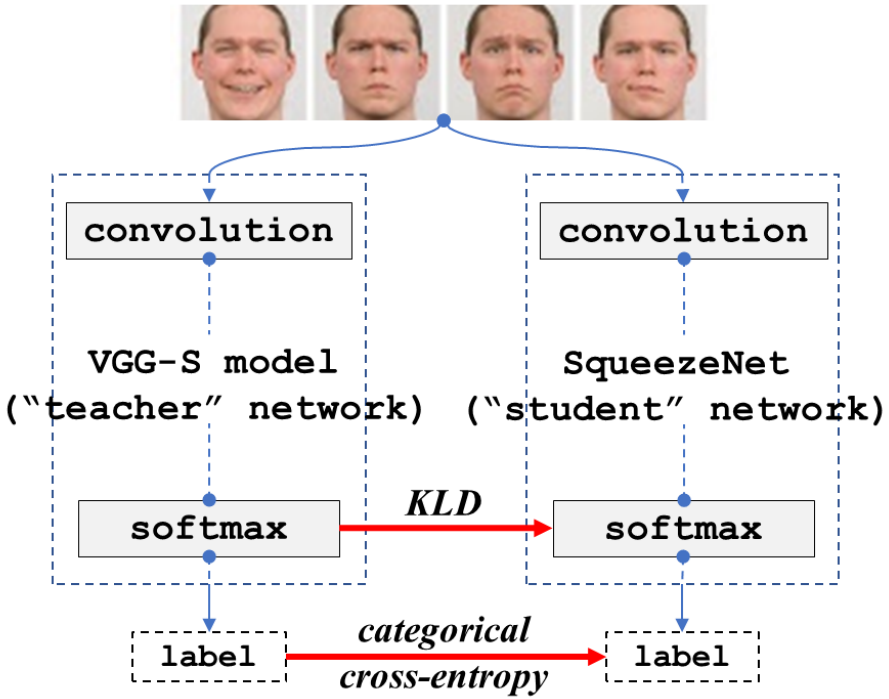
- a black-box model
- **limited** computational resources,
- **no** labeled data?

Distill the knowledge:

- from one single model to another,
- without architecture constraints,
- without true labels.

Experimental Setup

1. Taking a pre-trained on **EmotiW 2015** dataset model [Levi et al. 2015] for labeling **PubFig83** dataset
2. Training a lightweight **SqueezeNet 1.1** model on soft and hard outputs jointly



Experimental Results

	labels only	***labels + softmax***
baseline accuracy	41.45%	
distilled accuracy	66.5%	73%
new data accuracy	12.3%	18.6 (23.8)%

Experimental Setup

~~1. Taking a pre-trained on **EmotiW 2015** dataset model [Levi et al. 2015] for labeling **PubFig83** dataset~~

1. Training a powerful model using **Radboud Faces Database**
2. Labeling **PubFig83** dataset
3. Training a lightweight **SqueezeNet 1.1** model on soft and hard outputs jointly

Experimental Results

	labels only	***labels + softmax***
baseline accuracy	81%	
distilled accuracy	75.5%	77%
new data accuracy	40.9%	46.9%

model size reduced from **372** MB to **2.8** MB (<1% of original size)

Conclusions

What can we underline from this?

- **yes, it is possible!**
- training on hard and soft outputs jointly are better than separately
- weaker initial model tends to weaker result on new data
- straightforward implementation still suffer in accuracy

NATIONAL RESEARCH UNIVERSITY

Thank you for your attention!