N2N Learning: Network to Network Compression via Policy Gradient Reinforcement Learning

Recent trends in Automated Machine Learning

June 9th, 2021

Maximilian Gartner

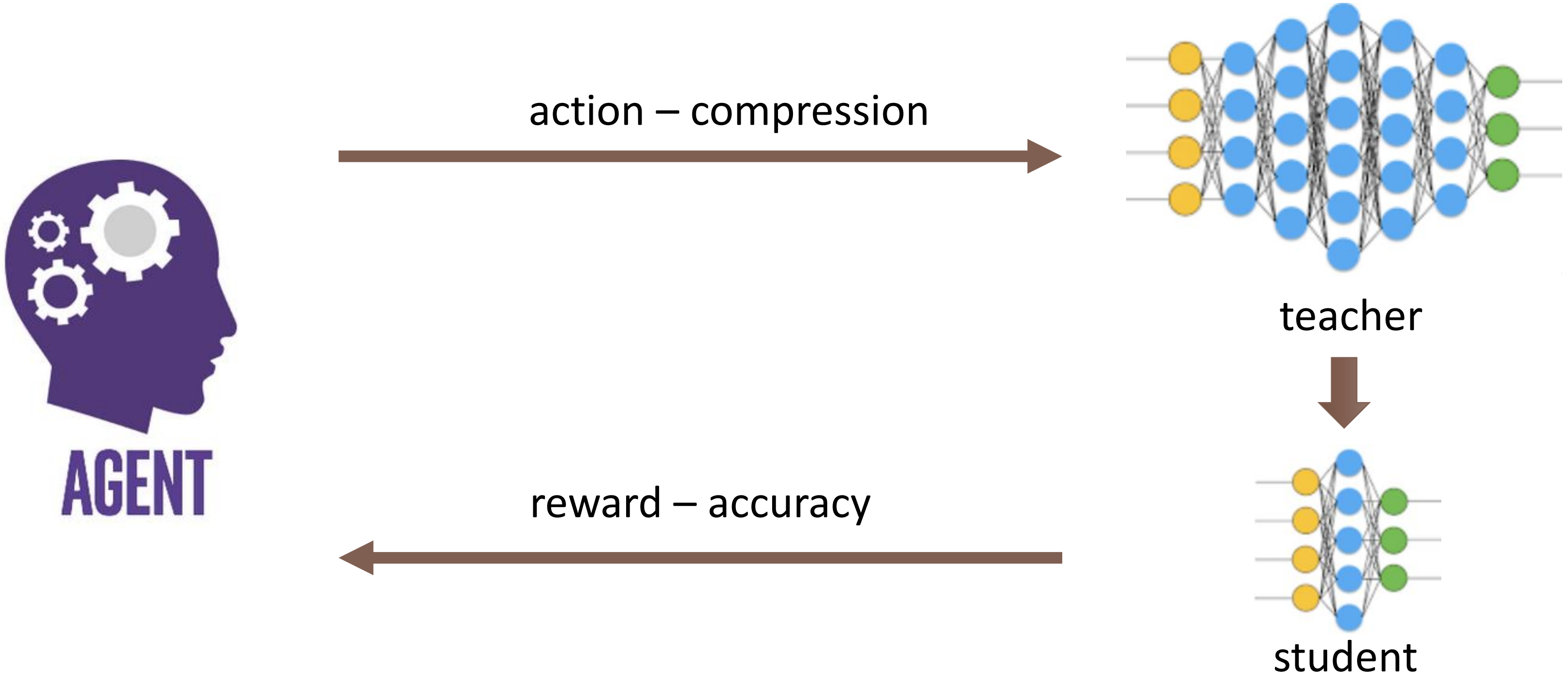
Motivation

- Neural Networks are getting bigger and bigger
  - Require high power, memory & computational resources
  - Not feasible on smaller devices (smartphones, smart-home devices, …)

- Goal: Find a small, but still high-performance architecture

Main Idea

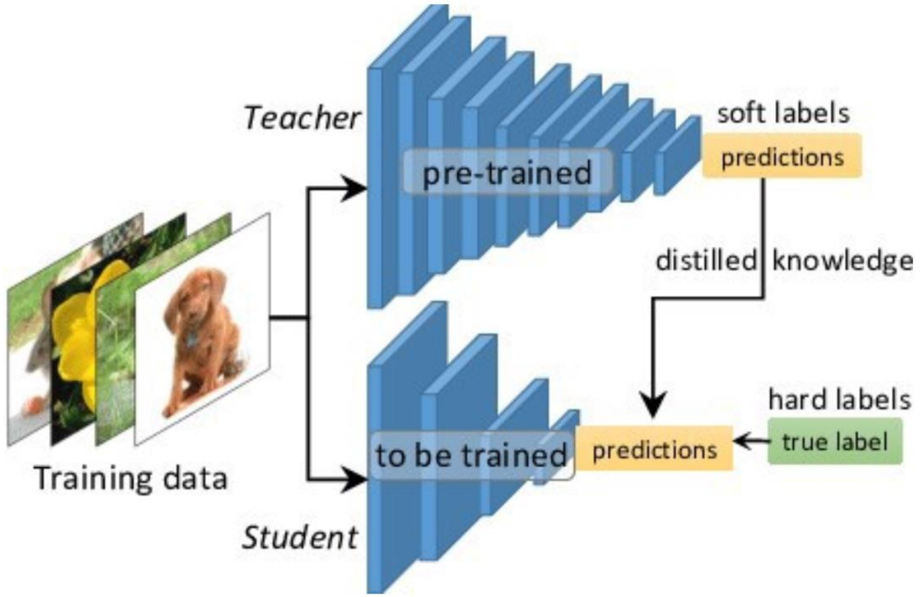
- Reinforcement Learning approach
- Take larger network (**Teacher** model) and identify compressed high-performance smaller network (**Student** model)



Underlying Concepts

(I) Knowledge Distillation

- Model compression approach
- Student is trained to learn the exact behavior of the (pre-trained) teacher by trying to replicate its output



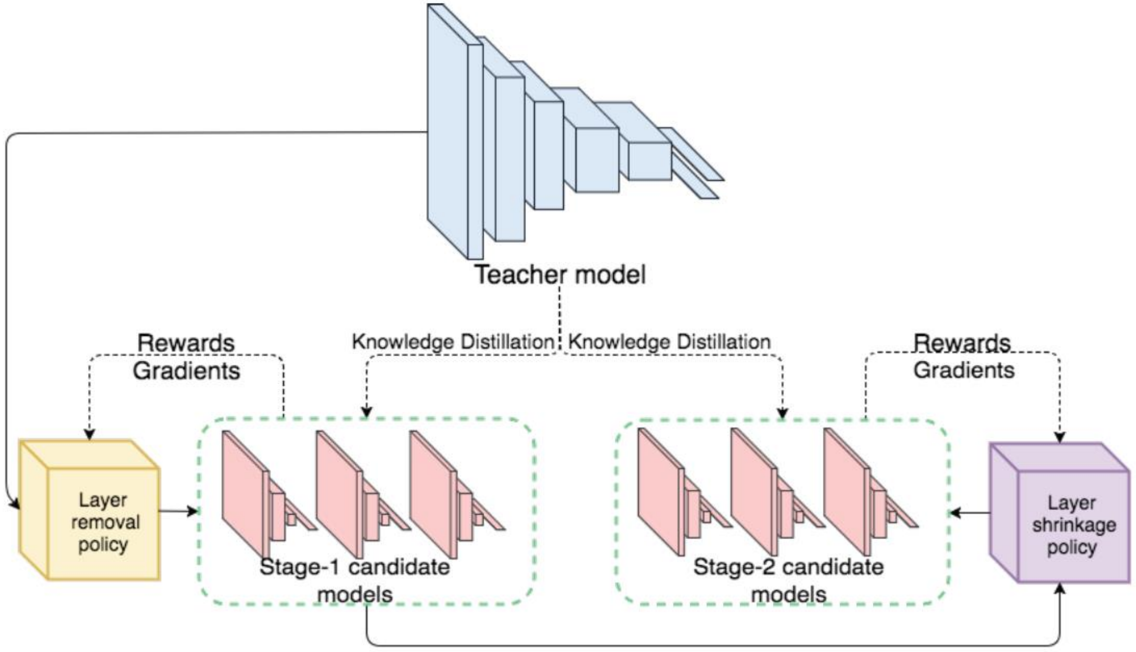
Underlying Concepts

(II) Neural Network Pruning

- Given the teacher model, find a much smaller subset that can provide same accuracy
- Preserve what matters most, remove redundance

N2N Process

- Two step reinforcement learning procedure
- First step: **Layer removal**
- Second step: **Layer shrinkage**



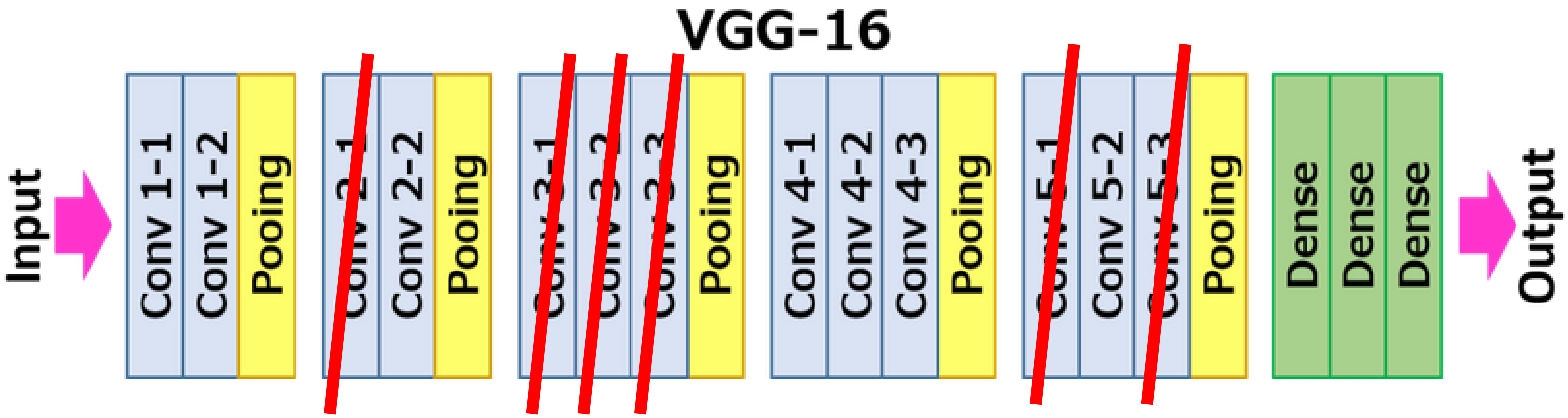
- Therefore: Encode layers of teacher
- E.g., Convolutional layer:

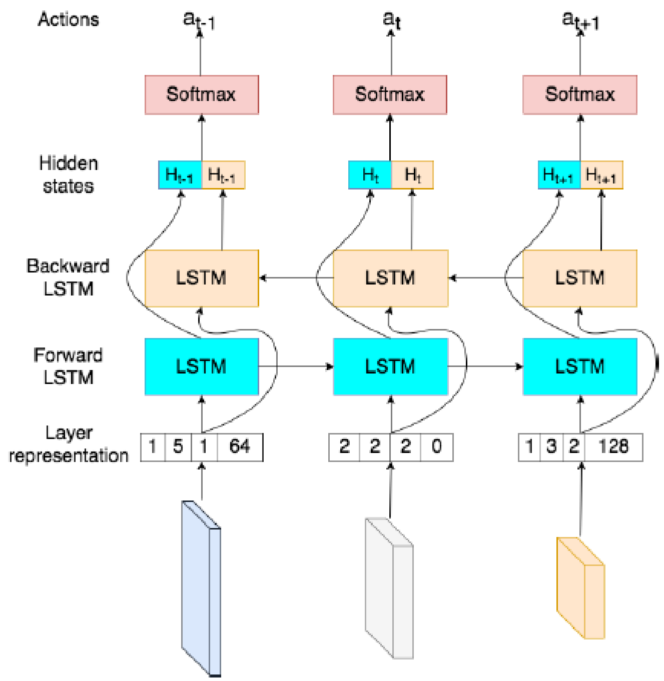
$$x_t = (l, k, s, p, n)$$

where l layer type, k filter size, s stride, p padding, n number of filters

Layer Removal (Step 1)

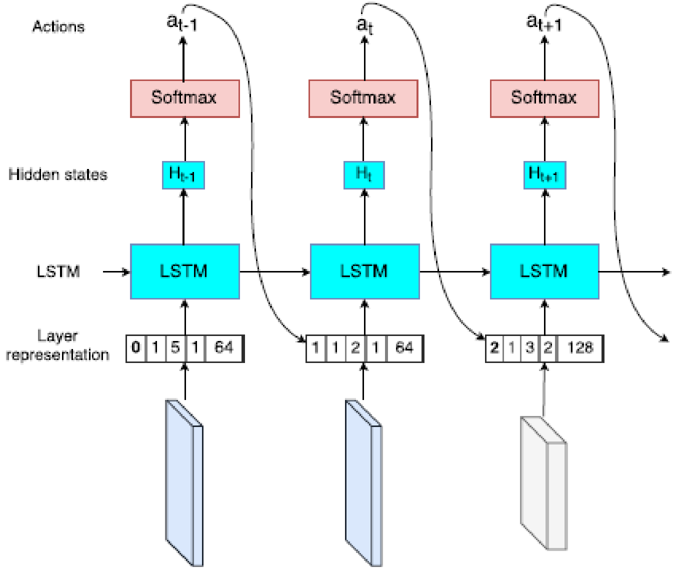
- Input: Teacher model
- Outputs for each layer an action $a_t \in \{0, 1\}$, whether to keep or remove the corresponding layer
- $\pi_{remove} = (a_t \mid x_t, h_{t-1}, h_{t+1})$
- Underlying policy network: bi-directional LSTM





Layer Shrinkage (Step 2)

- Input: output model of Step 1
- Outputs for each configuration variable of each layer an action $a_t \in [0.1, 0.2, \ldots, 1]$, deciding about how much to shrink the parameter
- $\pi_{shrink} = (a_t \mid x_t, a_{t-1}, h_{t-1},)$
- Underlying policy network: LSTM



$$
a_t = 0.6 \quad a_{t+1} = 1 \quad a_{t+2} = 1 \quad a_{t+3} = 0.4
$$

$$
x_t = (5, 1, 1, 64) \longrightarrow x'_t = (3, 1\,, 1, 26)
$$

Training of the Student model

- Goal: Knowledge Distillation of the teacher model
  - ➡ Not only train with hard labels (groundtruth) y_{true}, but also with logits z of the teacher

$$\mathcal{L}(W) = \mathcal{L}_{hard}(f(x;W), y_{true}) + \lambda \cdot \mathcal{L}_{KD}(f(x;W), z)$$

e.g., cross-entropy loss

e.g., mean squared error MSE

Reward Function - Requirements

- Calculation at the end with resulting trained student model
- Reward should depend on
  - How much is the student compressed?
  - How good is the student?

  } with respect to the teacher
- Also:

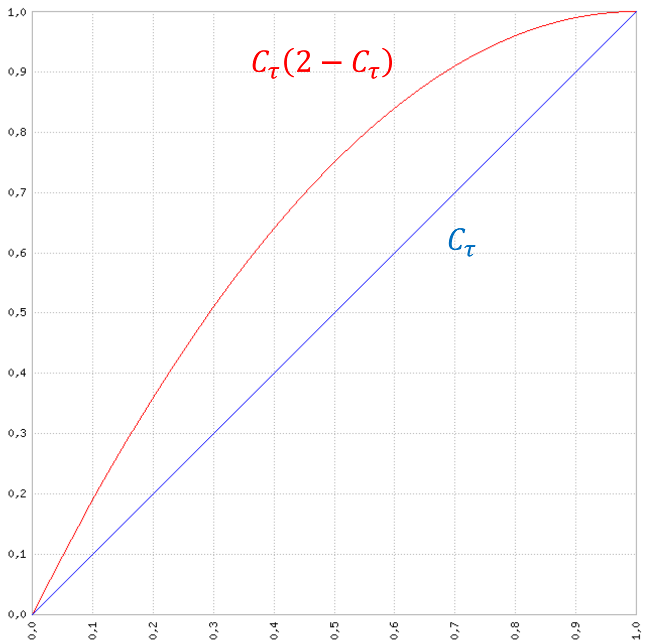
Model with low compression + high accuracy	>	Model with high compression + low accuracy

- Set $\mathrm{R} \leftarrow -1$ for degenerated student models

Reward Function

The relative compression $C_\tau \in [0, 1)$ is defined as $C_\tau = 1 - \frac{\#params(student)}{\#params(teacher)}$

Given C_τ, the **compression reward $\boldsymbol{R_{c,\tau}}$** is defined as $R_{c,\tau} = C_\tau(2 - C_\tau)$



Reward Function

The relative compression $C_\tau \in [0, 1)$ is defined as $C_\tau = 1 - \frac{\#params(student)}{\#params(teacher)}$

Given C_τ, the **compression reward $\boldsymbol{R_{c,\tau}}$** is defined as $R_{c,\tau} = C_\tau(2 - C_\tau)$

Given the validation accuracy of the with trajectory τ produced student model $A_{student}$ and the (fixed) accuracy of the teacher model $A_{teacher}$, the **accuracy reward $\boldsymbol{R_{a,\tau}}$** is defined as

$$R_{a,\tau} = \frac{A_{student}}{A_{teacher}}$$

$$R_\tau = R_{c,\tau} \cdot R_{a,\tau} = C_\tau(2 - C_\tau) \cdot \frac{A_{student}}{A_{teacher}}$$

Reward Function

The relative compression $C_\tau \in [0, 1)$ is defined as $C_\tau = 1 - \frac{\#params(student)}{\#params(teacher)}$

$$R_\tau = R_{c,\tau} \cdot R_{a,\tau} = C_\tau(2 - C_\tau) \cdot \frac{A_{student}}{A_{teacher}}$$

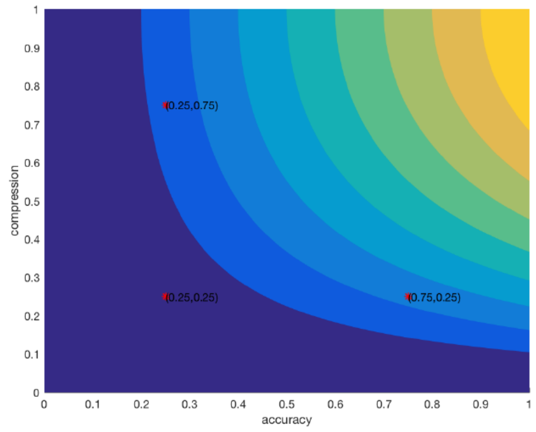
Example:

$R_{a,\tau} = 0.75, C_\tau = 0.25$ ➡ $R_\tau \approx 0.328$

$R_{a,\tau} = 0.25, C_\tau = 0.75$ ➡ $R_\tau \approx 0.234$

Model with low compression + high accuracy > Model with high compression + low accuracy

Reward Function



$$R_\tau = R_{c,\tau} \cdot R_{a,\tau} = C_\tau(2 - C_\tau) \cdot \frac{A_{student}}{A_{teacher}}$$

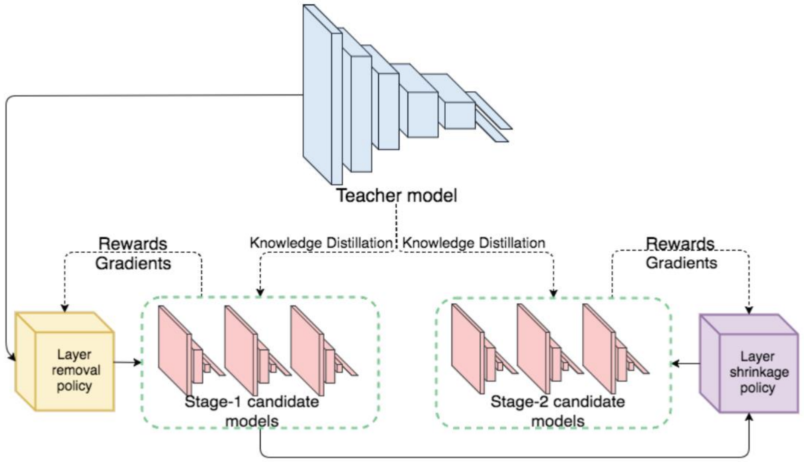
Example:

$$R_{a,\tau} = 0.75, C_\tau = 0.25 \quad \Rightarrow \quad R_\tau \approx 0.328$$

$$R_{a,\tau} = 0.25, C_\tau = 0.75 \quad \Rightarrow \quad R_\tau \approx 0.234$$

Model with low compression + high accuracy > Model with high compression + low accuracy

Optimization



- Optimize two policies
  - Bi-directional LSTM with parameters θ_{remove}
  - LSTM with parameters θ_{shrink}
- Maximize expected reward over all sequences of actions:

$$\max_{\theta} J(\theta), \qquad J(\theta) = E_{a_{1:T} \sim P_\theta}(R)$$

- Optimization: with REINFORCE

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta E_{a_{1:T} \sim P_\theta}(R) \\ &= \sum_{t=1}^{T} E_{a_{1:T} \sim P_\theta}[\nabla_\theta \log P_\theta(a_t | a_{1:(t-1)}) R] \\ &\approx \frac{1}{m} \sum_{k=1}^{m} \sum_{t=1}^{T} [\nabla_\theta \log P_\theta(a_t | h_t) R_k] \end{aligned}$$

Experiments

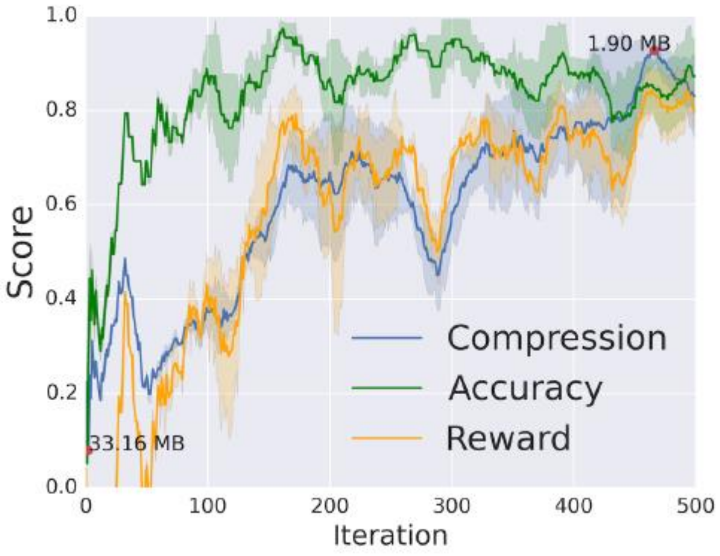
- Teacher: VGG-13 (9.4 Mio Params)
- Dataset: MNIST
- Epochs Training Student: 5

MNIST					
Architecture		Acc.	#Params	Δ Acc.	Compr.
VGG-13	Teacher	99.54%	9.4M	—	—
	Student	99.55%	73K	+0.01%	127x

Model	Acc.	#Params	Compr.
Teacher (MNIST/VGG-13)	**99.54%**	9.4M	1x
Student (Stage 1 & 2)	**98.91%**	**17K**	**553x**

Stage 1 (removal):

Stage 2 (removal & shrinkage):

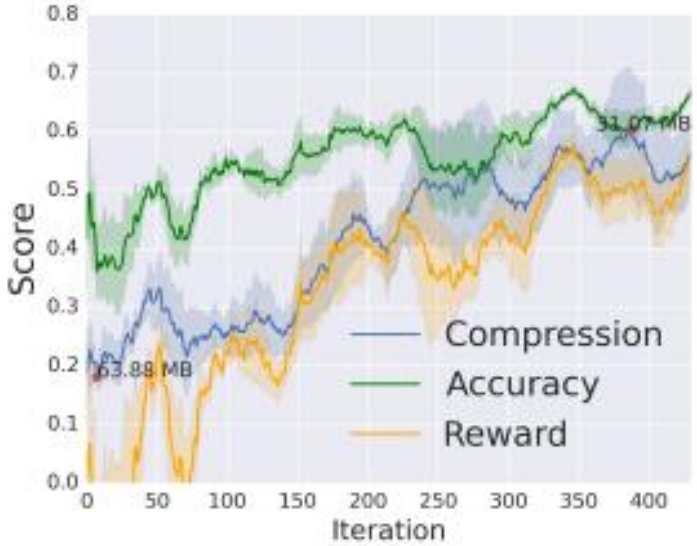


Experiments

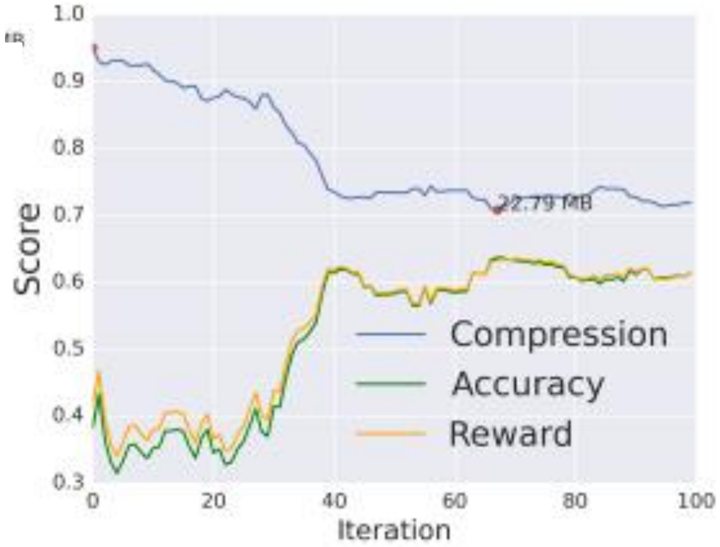
- Teacher: VGG-19 (20.2 Mio Params)
- Dataset: CIFAR-10
- Epochs Training Student: 5

CIFAR-10					
VGG-19	Teacher	91.97%	20.2M	—	—
	Student (Stage1)	92.05%	1.7M	+0.08%	11.8x
	Student (Stage1+Stage2)	91.64%	984K	-0.33%	20.53x

Stage 1 (removal):



Stage 2 (removal & shrinkage):



Evaluation

NN become more applicable, not only for real world application, but also a wider group of AI researchers

Accuracy-Performance Tradeoff

Low student training cost

High policy training costs
Also: Policy update after 5 epochs of training

Pruning and knowledge distillation better than with hand-crafted techniques

Search space limitations:
Student will always be similar to the teacher

References

- N2N Learning: Network to Network Compression via Policy Gradient Reinforcement Learning, Ashok et al.
- Neural Architecture Search with Reinforcement Learning, Zoph et al.
- https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764
- https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-1-a-brief-introduction-a53a849771cf
- https://bdtechtalks.com/2020/10/12/deep-learning-neural-network-pruning/
- https://neurohive.io/en/popular-networks/vgg16/

Thank you!