SANAS

Stochastic Adaptive Neural Architecture Search for Keyword Spotting

ICASSP 2019

Tom Veniat
Olivier Schwander
Ludovic Denoyer

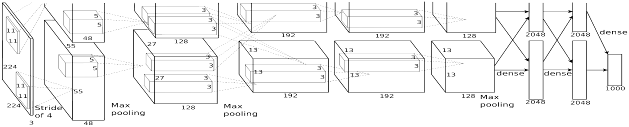
Context

Deep Learning Success

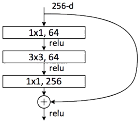
- **State of the art** in a number of domains (vision, speech, etc...)

Deep Learning Success

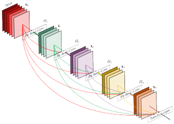
- **State of the art** in a number of domains (vision, speech, etc...)
- Example in computer vision : **plenty** of different architectures.



AlexNet [Krizhevsky, Sutskever, and Hinton 2012]



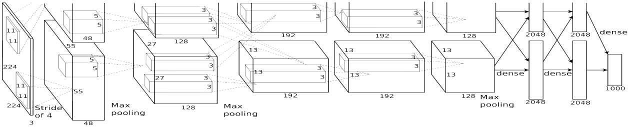
ResNet [He et al. 2015]



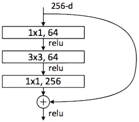
DenseNet [Huang, Liu, and Weinberger 2016]

Deep Learning Success

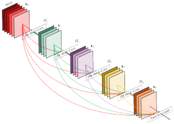
- **State of the art** in a number of domains (vision, speech, etc...)
- Example in computer vision : **plenty** of different architectures.



AlexNet [Krizhevsky, Sutskever, and Hinton 2012]



ResNet [He et al. 2015]



DenseNet [Huang, Liu, and Weinberger 2016]

- Need for an automatic way to discover architectures

Existing approaches

- Searches

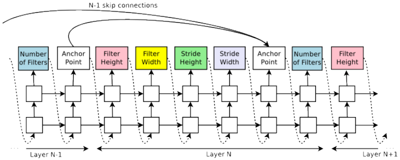
SCIENCES SORBONNE UNIVERSITÉ

Existing approaches

- Searches
- Evolutionary methods

Existing approaches

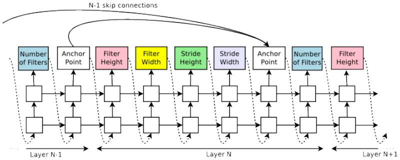
- Searches
- Evolutionary methods
- Learning
  - Example using Recurrent Neural Networks and Reinforcement Learning:



Neural architecture search with RL, [Zoph and Le 2016]

Existing approaches

- Searches
- Evolutionary methods
- Learning
  - Example using Recurrent Neural Networks and Reinforcement Learning:



Neural architecture search with RL, [Zoph and Le 2016]

Existing approaches limitations

- Mainly focused on performance
- Ignore real world constraints

Problematic

Architecture search under budget constraints

Problematic

Architecture search under budget constraints

Contributions

- New model : Budgeted Super Networks
- Joint optimization on performance and inference cost
  - **Costs :** Time, Memory, Parallelization
  - Custom costs based on production infrastructure
  - **No assumption on the cost nature**

Budgeted Super Networks

Definition

- A Super Network is a DAG of layers $(l_1, ..., l_N)$
- l_1 is the input layer and l_N is the output layer
- $E = \{e_{i,j}\} \in \{0,1\}^{N\times N}$ is the edge between l_i and l_j and is associated with function $f_{i,j}$ parametrized by $\theta_{i,j}$

Definition

- A Super Network is a DAG of layers $(l_1, ..., l_N)$
- l_1 is the input layer and l_N is the output layer
- $E = \{e_{i,j}\} \in \{0,1\}^{N \times N}$ is the edge between l_i and l_j and is associated with function $f_{i,j}$ parametrized by $\theta_{i,j}$

Inference

- **Input:** $l_1(x, E, \theta) = x$
- **Layer Computation:** $l_i(x, E, \theta) = \sum_k e_{k,i} f_{k,i}(l_k(x, E, \theta))$
- **Output:** $f(x, E, \theta) = l_N(x, E, \theta)$
- Learning can be done by back-propagation

Super Network Inference

Algorithm 1: SN Forward

Data: x, E, θ

$l_1 \leftarrow x$; `// Init the first layer`

for $i \in [2..N]$ **do**

  $l_i \leftarrow \sum_{k<i} e_{k,i} f_{k,i}(l_k; \theta_{k,i})$; `// Propagate through the super network`

end

Idea: Identifying a sub-network

- Keep a good accuracy
- Reduce cost

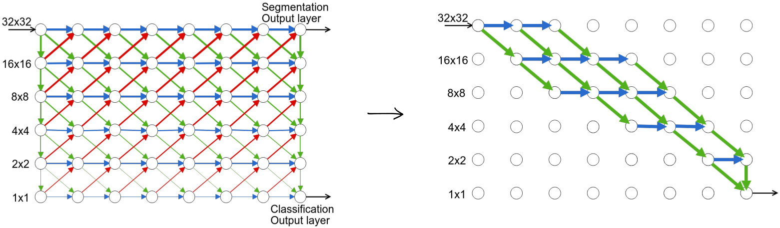


Figure 2: Convolutional Neural Fabrics [Saxena and Verbeek 2016]

Idea: Identifying a sub-network

- Keep a good accuracy
- Reduce cost

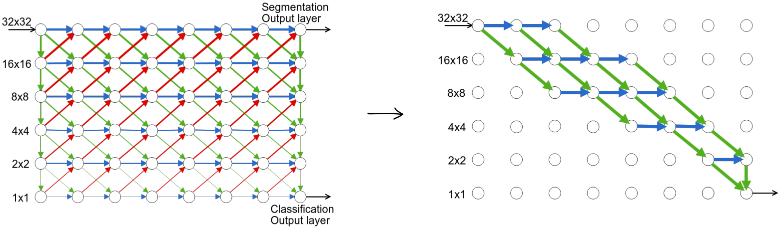


Figure 2: Convolutional Neural Fabrics [Saxena and Verbeek 2016]

Notation

- $H \in \{0,1\}^{N \times N}$ such that $H \odot E$ defines a (sub) Super Network
- $C(H \odot E)$ the cost for computing $f(x, H \odot E, \theta)$

Learning under budget constraints

$$H^*, \theta^* = \arg\min_{H,\theta} \frac{1}{\ell} \sum_i \Delta(f(x^i, H \odot E, \theta), y^i),$$
$$\text{under constraint : } C(H \odot E) \leq \mathbf{C}$$

Learning under budget constraints

$$H^*, \theta^* = \arg\min_{H,\theta} \frac{1}{\ell} \sum_i \Delta(f(x^i, H \odot E, \theta), y^i),$$
$$\text{under constraint}: \; C(H \odot E) \leq \mathbf{C}$$

Soft version

$$H^*, \theta^* = \arg\min_{H,\theta} \frac{1}{\ell} \sum_i \Delta(f(x^i, H \odot E, \theta), y^i) + \lambda \max(0, C(H \odot E) - \mathbf{C})$$

Combinatorial Problem

- How to explore the **discrete** architecture space ?
- How to handle the **unknown** cost function $C(H \odot E)$?

Combinatorial Problem

- How to explore the **discrete** architecture space ?
- How to handle the **unknown** cost function $C(H \odot E)$?

Idea

- Reformulate the learning problem as a **stochastic** problem.
- Apply Reinforcement Learning techniques to overcome the combinatorial problem.

Stochastic Super Network

- We consider a matrix of probabilities Γ
- At each inference, H is sampled following $H \sim \Gamma$

Stochastic Super Network Inference

Algorithm 2: SSN Forward

Data: x, E, Γ, θ

$H \sim \Gamma$; // Sample an architecture

$l_1 \leftarrow x$;

for $i \in [2..N]$ **do**

  $l_i \leftarrow \sum_{k<i} e_{k,i} h_{k,i} f_{k,i}(l_k; \theta_{k,i})$; // Propagate through the sampled network

end

Stochastic learning problem

$$\Gamma^*, \theta^* = \arg\min_{\Gamma,\theta} \frac{1}{\ell} \sum_i \mathbb{E}_{H\sim\Gamma} \Big[\Delta(f(x^i, H \odot E, \theta), y^i) \\ + \lambda \max(0, C(H \odot E) - \mathbf{C}) \Big]$$

- **Solving this problem is equivalent to solving the original constrained problem.**
- Can be optimized by SGD using REINFORCE.

Deriving the stochastic learning problem

Let us define:

$$\mathcal{D}(x, y, \theta, E, H) = \Delta(f(x, H \odot E, \theta), y) + \lambda \max(0, C(H \odot E) - \mathbf{C})$$

$$\mathcal{L}(x, y, E, \Gamma, \theta) = \mathbb{E}_{H \sim \Gamma}\left[\mathcal{D}(x, y, \theta, E, H)\right]$$

We have:

$$\begin{aligned}\nabla_{\theta,\Gamma}\mathcal{L}(x, y, E, \Gamma, \theta) = &\sum_H P(H|\Gamma)\left[(\nabla_{\theta,\Gamma} \log P(H|\Gamma))\mathcal{D}(x, y, \theta, E, H)\right] \\ &+ \sum_H P(H|\Gamma)\left[\nabla_{\theta,\Gamma}\Delta(f(x, H \odot E, \theta), y)\right]\end{aligned}$$

SANAS

From static

$$\mathcal{L}(x, y, E, \Gamma, \theta) = \mathbb{E}_{H \sim \Gamma} \left[\Delta(f(x, H \odot E, \theta), y) + \lambda \max(0, C(H \odot E) - \mathbf{C}) \right]$$

From static

$$\mathcal{L}(x, y, E, \Gamma, \theta) = \mathbb{E}_{H \sim \Gamma}\left[\Delta(f(x, H \odot E, \theta), y) + \lambda \max(0, C(H \odot E) - \mathbf{C})\right]$$

To dynamic

$$\mathcal{L}(x, y, \theta) = \mathbb{E}_{\mathcal{A}}\Big[\sum_{t=1}^{\#x}[\Delta(f(z_t, x_t, \theta, \mathcal{A}_t), y_t) + \lambda C(\mathcal{A}_t)]\Big]$$

General Model

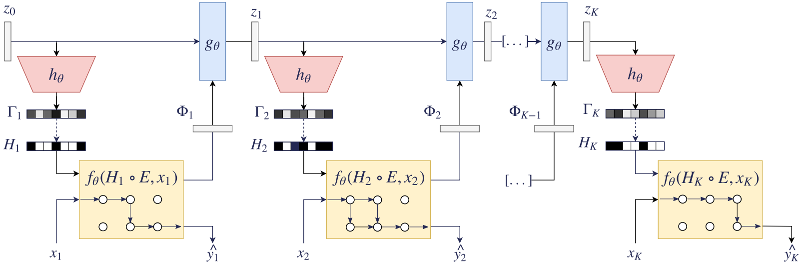


Figure 3: SANAS Architecture unrolled on sequence of length K.

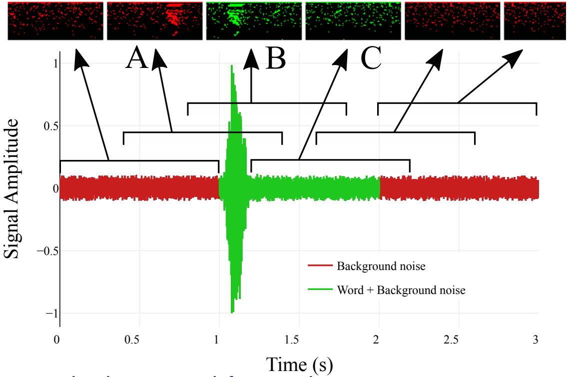
Keyword Spotting - Speech Commands Dataset [Warden 2018]

- 65000 short audio clips
- 30 common words
- 12 classes

Keyword Spotting - Speech Commands Dataset [Warden 2018]

- 65000 short audio clips
- 30 common words
- 12 classes

Streaming dataset



Keyword Spotting - Model

- based on *cnn-trad-fpool3* [Sainath and Parada 2015]

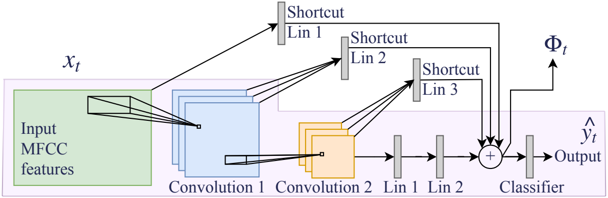


Figure 4: SANAS architecture for Keyword Spotting

Quantitative results

Match	Correct	Wrong	FA	FLOPs per frame
cnn-trad-fpool3				
81.7%	72.8%	8.9%	0.0%	124.6M
***cnn-trad-fpool3* + shortcut connections**				
82.9%	77.9%	5.0%	0.3%	137.3M
SANAS				
61.2%	53.8%	7.3%	0.7%	519.2K
62.0%	57.3%	4.8%	0.1%	22.9M
86.5%	80.7%	5.8%	0.3%	37.7M
86.3%	80.6%	5.7%	0.2%	48.8M
81.7%	76.4%	5.3%	0.1%	94.0M
81.4%	76.3%	5.2%	0.2%	105.4M

Table 1: Evaluation of models on 1h of audio.

Quantitative results

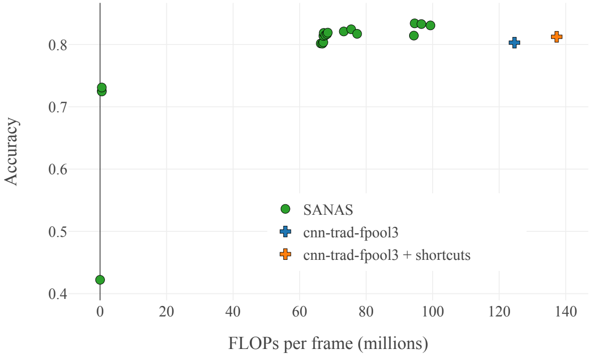


Figure 5: Cost/accuracy curves on test set.

Training dynamics

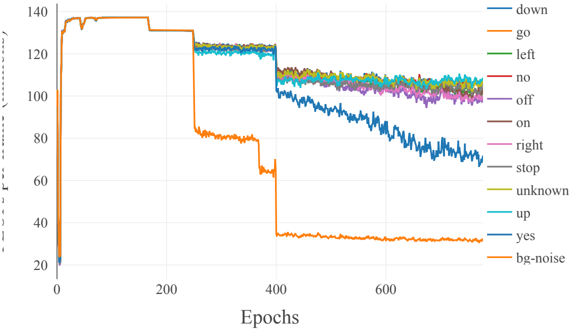


Figure 6: Cost per word during training

Perspectives

- Use new models (Currently training on Resnets)
- Test other sound datasets
- Evaluate over different tasks (Video, Event detection, RL ...)

References

References I

He, Kaiming et al. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. url: http://arxiv.org/abs/1512.03385.

Huang, Gao, Zhuang Liu, and Kilian Q. Weinberger (2016). "Densely Connected Convolutional Networks". In: *CoRR* abs/1608.06993. url: http://arxiv.org/abs/1608.06993.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* Pp. 1106–1114. url: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

Sainath, Tara N. and Carolina Parada (2015). "Convolutional neural networks for small-footprint keyword spotting". In: *INTERSPEECH*. ISCA, pp. 1478–1482.

Saxena, Shreyas and Jakob Verbeek (2016). "Convolutional Neural Fabrics". In: *CoRR* abs/1606.02492. url: http://arxiv.org/abs/1606.02492.

Warden, Pete (2018). "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". In: *CoRR* abs/1804.03209. arXiv: 1804.03209. url: http://arxiv.org/abs/1804.03209.

References II

Zoph, Barret and Quoc V. Le (2016). "Neural Architecture Search with Reinforcement Learning". In: *CoRR* abs/1611.01578. url: http://arxiv.org/abs/1611.01578.