IBM Research

Efficient Neural Architecture Search

Martin Wistuba, Tejaswini Pedapati

IBM Research

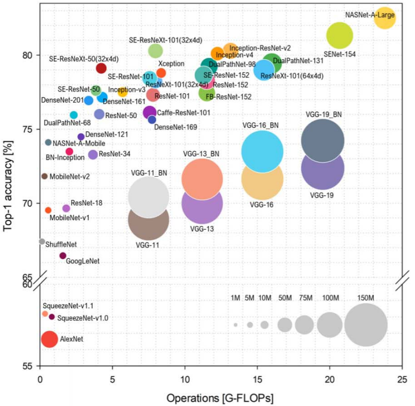
03 February 2021

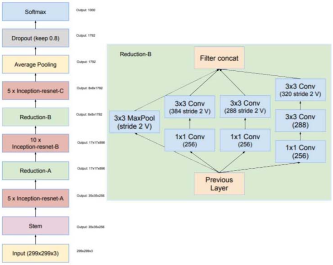
Outline

IBM Research

Architecture Design for Image Tasks

IBM Research



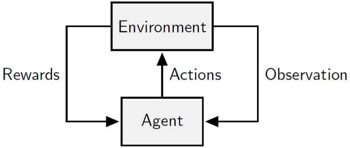


Innovations: Deeper networks, auxiliary classifiers, skip connections, bottlenecks, convolution stacking, global average pooling and many more

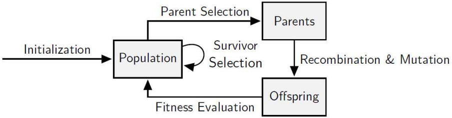
Images taken from Simone Bianco et al. “Benchmark Analysis of Representative Deep Neural Network Architectures”. In: *IEEE Access* 6 (2018), pp. 64270–64277, Christian Szegedy et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*. 2017, pp. 4278–4284

Neural Architecture Search

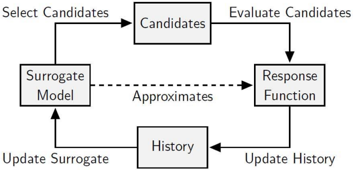
IBM Research



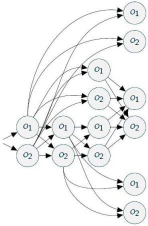
Reinforcement Learning



Evolutionary Algorithm



Surrogate Model-based Optimization



One-Shot Architecture Search

Tutorial Outline

IBM Research

Part 1

- Formal Definition of NAS and NASNet search space
- One-shot techniques in NAS
  - Overview
  - Shortcomings
  - Once-for-All Network

Part 2

- Effective NAS with transfer learning approaches based on
  - Transfer NAS optimizers
  - Few-Shot NAS optimizers
  - Learning Curve Ranking

Problem Definition

IBM Research

Machine Learning Problem

$$\Lambda\left(\boldsymbol{\alpha}, d\right) = \underset{m_{\boldsymbol{\alpha},\boldsymbol{\theta}} \in M_{\boldsymbol{\alpha}}}{\arg\min}\ \mathcal{L}\left(m_{\boldsymbol{\alpha},\boldsymbol{\theta}}, d_{\text{train}}\right) + \mathcal{R}\left(\boldsymbol{\theta}\right)\ . \tag{1}$$

- m - machine learning model
- $\boldsymbol{\theta}$ - model parameters
- $\boldsymbol{\alpha}$ - neural architecture
- d - dataset

NAS Problem

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha} \in A}{\arg\max}\ \mathcal{O}\left(\Lambda\left(\boldsymbol{\alpha}, d_{\text{train}}\right), d_{\text{valid}}\right) = \underset{\boldsymbol{\alpha} \in A}{\arg\max}\, f\left(\boldsymbol{\alpha}\right)\ . \tag{2}$$

- f - response function
- A - search space

Search Space

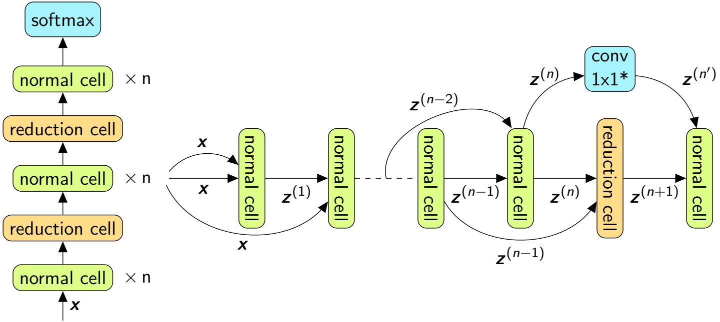
IBM Research

- Neural architecture search space: subspace of all possible neural architectures.
- The limitation to a subspace allows for considering
  - human expert knowledge,
  - specific task (e.g. mobile architectures) and
  - reduces the search time and improves the solutions.
- We distinguish two types of search spaces:
  - global search space
  - cell-based search space

NASNet Search Space

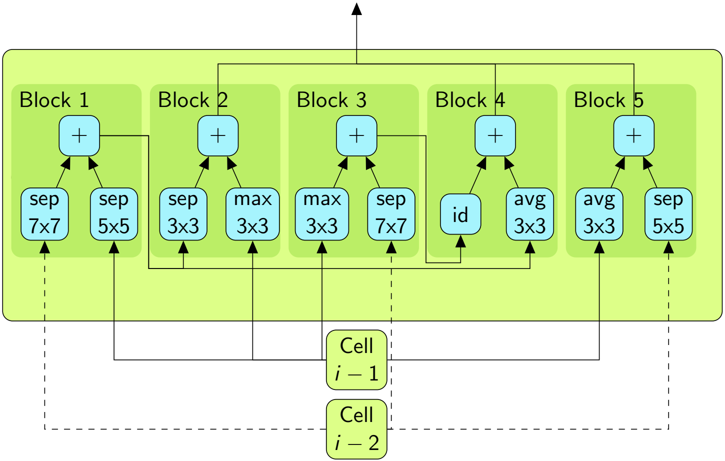
IBM Research

Architectures from a cell-based search space are build by stacking few cells with the same topology.



NASNet Search Space

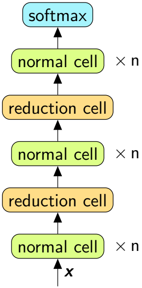
Structure of a cell.



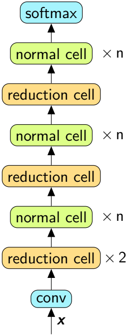
Transferring Architectures

IBM Research

Architectures from cell-based search spaces allow for easy transferability across different datasets.



(a) CIFAR-10



(b) ImageNet

Outline

Outline

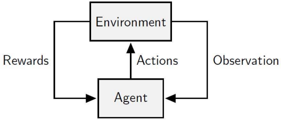
IBM Research

NAS Optimizers

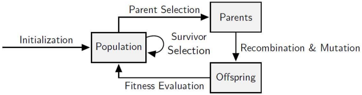
IBM Research

We distinguish several methods that maximize the response function:

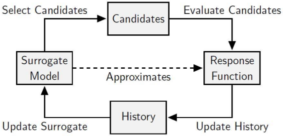
- **Reinforcement learning**: learn to sample α that maximize f.
- **Evolutionary algorithms**: evolve α that maximize f.
- **Surrogate model-based optimization**: approximate f by $\hat{f}$ and use it to maximize f.
- **One-shot architecture search**: learn one model and use it to max f.



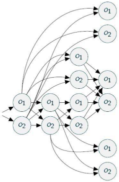
Reinforcement Learning



Evolutionary Algorithm



Surrogate Model-based Optimization



One-Shot Architecture Search

One-Shot Architecture Search

IBM Research

Until now,

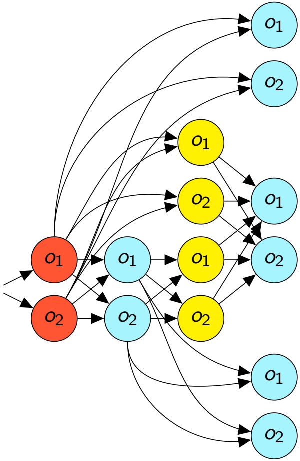
- the candidate architecture is trained from scratch to obtain validation accuracy
- Previously trained candidate architectures' weights were not reused.

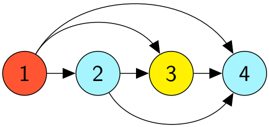
To overcome this, in one-shot architecture search the

- Entire search space is a directed acyclic graph - SuperNet
- Candidate architecture α is sampled from SuperNet
- The weights of all the operations are shared

One-Shot Architecture Search

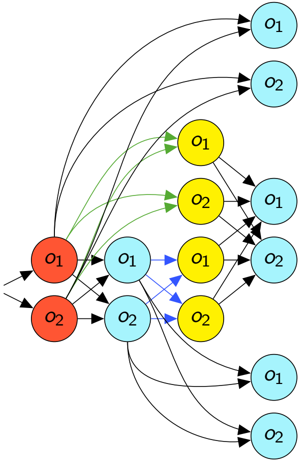
IBM Research

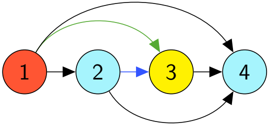




One-Shot Architecture Search

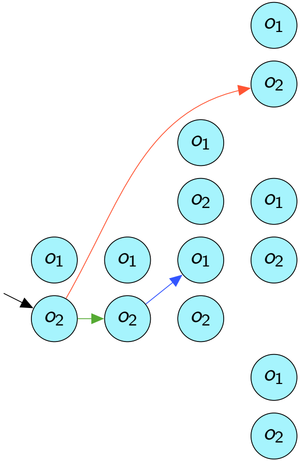
IBM Research

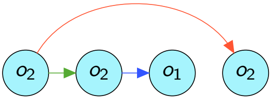




One-Shot Architecture Search

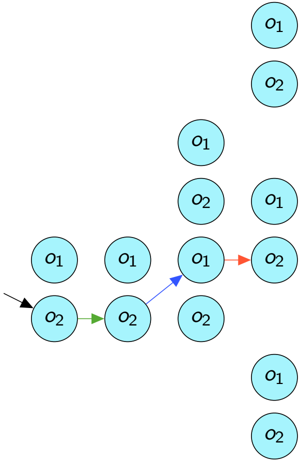
IBM Research

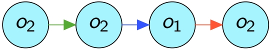




One-Shot Architecture Search

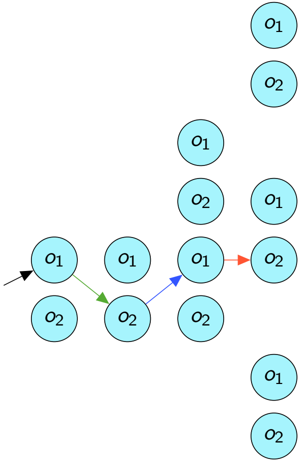
IBM Research

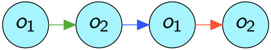




One-Shot Architecture Search

IBM Research





One-Shot Architecture Search

IBM Research

- Cross-entropy loss of $\boldsymbol{\alpha}$ is computed on a minibatch of training data
- SuperNet parameters θ are updated using the gradients from the model α.
- Accelerated the search from 360 GPU days to 0.32 GPU days.
- Best architecture obtained by NAS is again trained from scratch

One-Shot Architecture Search

IBM Research

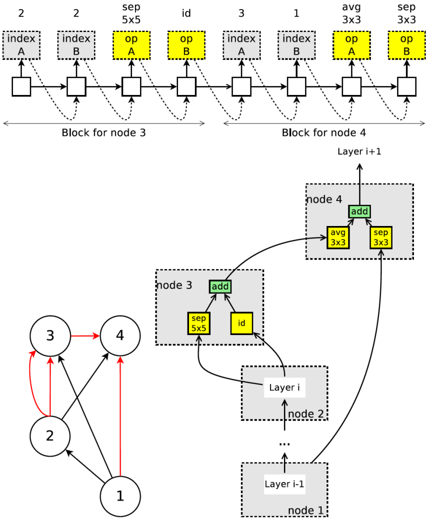
Sample Strategies

- Reinforcement learning (Pham et al.)
- Surrogate model-based optimization (Luo et al.)
- Learn a parameterized distribution (Casale et al.)
- Random sampling (Bender et al.)

Efficient Neural Architecture Search (ENAS)

IBM Research

Uses LSTM controller trained using RL to predict candidate network



Efficient Neural Architecture Search (ENAS)

IBM Research

Algorithm 1 ENAS

Input: Controller's policy parameters ω, SuperNet's parameters θ,
for every iteration **do**
Controller's policy samples candidate model α
Compute cross-entropy loss $\nabla_\theta E_\alpha$ on m for a mini-batch of training data
Fix ω and perform SGD on θ using $\nabla_\theta E_\alpha$
Fix θ and update ω to maximize expected reward on validation data.

Differentiable Architecture Search (DARTS)

IBM Research

- In ENAS, choosing operations at every edge is a discrete decision
- DARTS Makes it continuous defining mixed operation

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \tag{3}$$

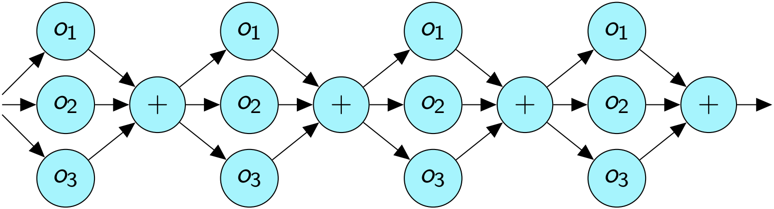
- Architecture α is parameterized by β and network weights θ
- Strength of an operation: $\frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})}$
- Derive discrete architecture by (1) $o^{(i,j)} = \mathrm{argmax}_{o \in \mathcal{O}}\ \alpha_o^{(i,j)}$ (2) choose top-k incoming edges

---

Hanxiao Liu, Karen Simonyan, and Yiming Yang. "DARTS: Differentiable Architecture Search". In: *Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA*. 2019

Differentiable Architecture Search

IBM Research

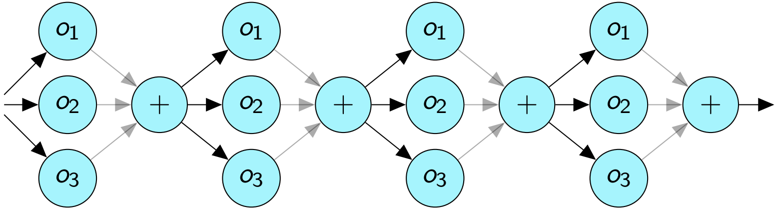


Relaxation of binary structural parameters α leads to differentiable loss:

$$\min_{\alpha(\beta)\in A} \mathcal{L}\left(\underset{m_{\alpha(\beta),\theta}\in M_{\alpha(\beta)}}{\arg\min} \mathcal{L}\left(m_{\alpha(\beta),\theta}, d_{\text{train}}\right) + \mathcal{R}\left(\theta\right), d_{\text{valid}}\right) \tag{4}$$

Differentiable Architecture Search

IBM Research

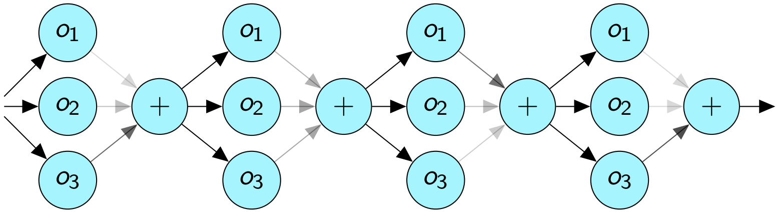


Relaxation of binary structural parameters $\boldsymbol{\alpha}$ leads to differentiable loss:

$$\min_{\boldsymbol{\alpha}(\boldsymbol{\beta})\in A} \mathcal{L}\left(\underset{m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}} \in M_{\boldsymbol{\alpha}(\boldsymbol{\beta})}}{\arg\min} \mathcal{L}\left(m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}}, d_{\text{train}}\right) + \mathcal{R}\left(\boldsymbol{\theta}\right), d_{\text{valid}} \right) \tag{4}$$

Differentiable Architecture Search

IBM Research

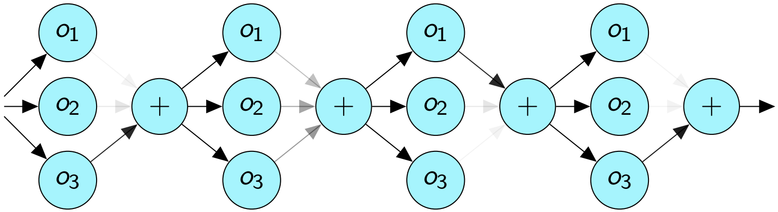


Relaxation of binary structural parameters α leads to differentiable loss:

$$\min_{\boldsymbol{\alpha}(\boldsymbol{\beta})\in A} \mathcal{L}\left(\operatorname*{arg\,min}_{m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}}\in M_{\boldsymbol{\alpha}(\boldsymbol{\beta})}} \mathcal{L}\left(m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}}, d_{\text{train}}\right) + \mathcal{R}\left(\boldsymbol{\theta}\right), d_{\text{valid}}\right) \tag{4}$$

Differentiable Architecture Search

IBM Research

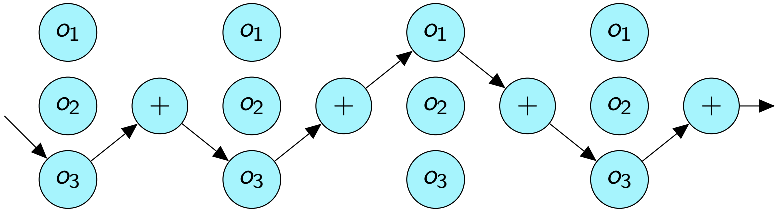


Relaxation of binary structural parameters α leads to differentiable loss:

$$\min_{\boldsymbol{\alpha}(\boldsymbol{\beta})\in A} \mathcal{L}\left(\operatorname*{arg\,min}_{m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}}\in M_{\boldsymbol{\alpha}(\boldsymbol{\beta})}} \mathcal{L}\left(m_{\boldsymbol{\alpha}(\boldsymbol{\beta}),\boldsymbol{\theta}}, d_{\text{train}}\right) + \mathcal{R}\left(\boldsymbol{\theta}\right), d_{\text{valid}}\right) \tag{4}$$

Differentiable Architecture Search

IBM Research



Relaxation of binary structural parameters α leads to differentiable loss:

$$\min_{\alpha(\beta)\in A} \mathcal{L}\left(\arg\min_{m_{\alpha(\beta),\theta}\in M_{\alpha(\beta)}} \mathcal{L}\left(m_{\alpha(\beta),\theta}, d_{\text{train}}\right) + \mathcal{R}\left(\theta\right), d_{\text{valid}}\right) \tag{4}$$

IBM Research

Bi-level optimization

$$\min_{\alpha} \quad \mathcal{L}_{val}(\theta^*(\alpha), \alpha) \tag{5}$$

$$\text{s.t.} \quad \theta^*(\alpha) = \operatorname{argmin}_\theta \; \mathcal{L}_{train}(\theta, \alpha) \tag{6}$$

Algorithm 2 DARTS – Differentiable Architecture Search

Input: A mixed operation $\bar{o}^{(i,j)}$

1: **while** not converged **do**
2: Update architecture α by descending
3: $\nabla_\alpha \mathcal{L}_{val}(\theta - \xi \nabla_\theta \mathcal{L}_{train}(\theta, \alpha), \alpha)$
4: ($\xi = 0$ if using first-order approximation)
5: Update weights θ by descending $\nabla_\theta \mathcal{L}_{train}(\theta, \alpha)$
6: Derive the final architecture based on the learned α.

Outline

IBM Research

Pitfalls of DARTS

IBM Research

- All parameters need to be stored in memory.
- DARTS Collapse: Final architectures comprise of too many skip connections
- Discretization step results in architectures with higher validation loss

Outline

IBM Research

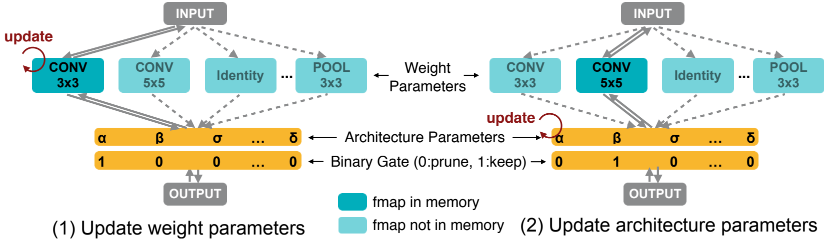
Memory consumption of DARTS

IBM Research

- In DARTS output of an edge is a weighted sum of all the operations $\sum_{i=1}^{N} \frac{\exp(\alpha_i)}{\sum_j \exp(\alpha_j)} o_i(x)$
- It requires all possible combinations of the operations to be stored in memory
- The batch size used to train the SuperNet is small
- Searching on Imagenet takes several days.

ProxylessNAS

IBM Research



- Each edge of the SuperNet has N different operations.
- It is equivalent to storing N models in memory
- In ProxylessNAS only one operation is active at a time.

Han Cai, Ligeng Zhu, and Song Han. "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware". In: *Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA*. 2019

ProxylessNAS (cont.)

IBM Research

Use binary gates for each edge:

$$g = \text{binarize}(p_1, \cdots, p_N) = \begin{cases} [1, 0, \cdots, 0] & \text{with probability } p_1, \\ \cdots & \\ [0, 0, \cdots, 1] & \text{with probability } p_N. \end{cases} \quad (7)$$

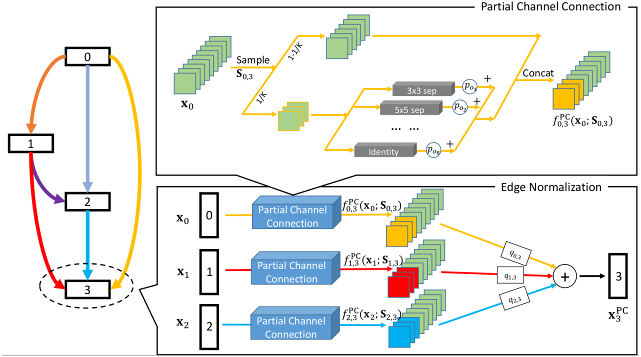
$$m_{\mathcal{O}}^{\text{Binary}}(x) = \sum_{i=1}^{N} g_i o_i(x) = \begin{cases} o_1(x) & \text{with probability } p_1 \\ \cdots & \\ o_N(x) & \text{with probability } p_N. \end{cases} \quad (8)$$

- Devise it as multiple binary selection tasks
- Requires only 2 paths in memory at any point.
- Able to search on ImageNet in 8.3 days

PC-DARTS

IBM Research

- Use channel mask $S_{i,j}$ to sample a 1/K channels each time
- K will determine accuracy vs search cost trade-off



Yuhui Xu et al. "PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020

PC-DARTS (cont.)

IBM Research

$$f_{i,j}^{\mathrm{PC}}(\mathsf{x}_i; \mathsf{S}_{i,j}) = \sum_{o \in \mathcal{O}} \frac{\exp\left\{\alpha_{i,j}^{o}\right\}}{\sum_{o' \in \mathcal{O}} \exp\left\{\alpha_{i,j}^{o'}\right\}} \cdot o(\mathsf{S}_{i,j} * \mathsf{x}_i) + (1 - \mathsf{S}_{i,j}) * \mathsf{x}_i. \quad (9)$$

- To account for changing sampled channels, edge normalization is introduced
- All the edges contributing to the output of node j are assigned weights

$$\mathsf{x}_j^{\mathrm{PC}} = \sum_{i<j} \frac{\exp\{\gamma_{i,j}\}}{\sum_{i'<j} \exp\{\gamma_{i',j}\}} \cdot f_{i,j}(\mathsf{x}_i). \quad (10)$$

- First train SuperNet for 15 epochs
- Increased batch size stabilizes the training
- The bias of choosing parameter-free operations is less pronounced

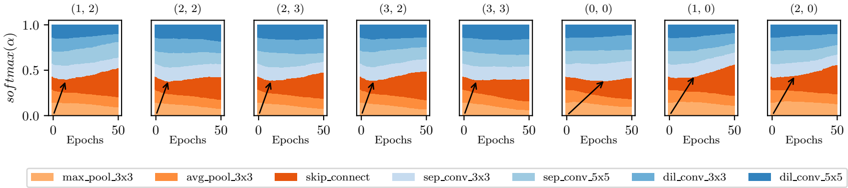
Outline

IBM Research

DARTS Collapse

IBM Research

- Skip connections increase as search progresses
- Skip connections make it easier for the SuperNet to train although they do not boost the accuracy of the final discretized architecture



The softmax evolution where skip connections gradually become dominant. Image taken from Chu et al.

DARTS Collapse

IBM Research

S1: This search space uses a different set of only two operators per edge.

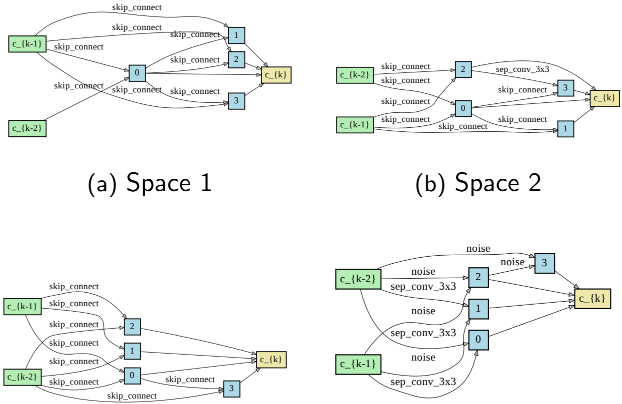
S2: Operated used are $\{3 \times 3$ *SepConv*, *SkipConnect*$\}$.

S3: The set of candidate operations per edge is $\{3 \times 3$ *SepConv*, *SkipConnect*, *Zero*$\}$.

S4: The set of candidate operations per edge is $\{3 \times 3$ *SepConv*, *Noise*$\}$.

DARTS Collapse

IBM Research



(a) Space 1 (b) Space 2

(c) Space 3 (d) Space 4

The normal cells standard DARTS finds on spaces S1-S4. Image taken from Zela et al.

DARTS Collapse

IBM Research

- Dropout after skip connection as suggested by P-DARTS
- Explicitly limit the number of skip connections: DARTS+
- Experiment done by FairDARTS

Methods	Cifar10-Acc
Random (M=2)	97.01 ± 0.24
Random (M=2, MultAdds $\geq$ 500M)	97.14 ± 0.28
DARTS without skip-connection	96.88 ± 0.18
DARTS (First Order) + Gaussian (cosine decay)	97.12 ± 0.23
DARTS (First Order)	97.00 ± 0.14

Operation choice no longer mutually exclusive

IBM Research

Apply a *sigmoid activation* (σ) for each $\alpha_{o_{i,j}}$, so that each operation can be switched on or off independently without being suppressed.

$$\bar{o}_{i,j}(x) = \sum_{o \in \mathcal{O}} \sigma(\alpha_{o_{i,j}}) o(x). \tag{11}$$

For the sigmoid of architectural weights to tend towards 0 or 1, additional loss is used

$$L_{0-1} = -\frac{1}{N} \sum_{i}^{N} (\sigma(\alpha_i) - 0.5)^2 \tag{12}$$

$$L_{total} = \mathcal{L}_{val}(w^*(\alpha), \alpha) + w_{0-1} L_{0-1}. \tag{13}$$

The final architecture is discretized by using a threshold ($\sigma_{threshold}$) instead of argmax

Xiangxiang Chu et al. "Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*. 2020, pp. 465–480

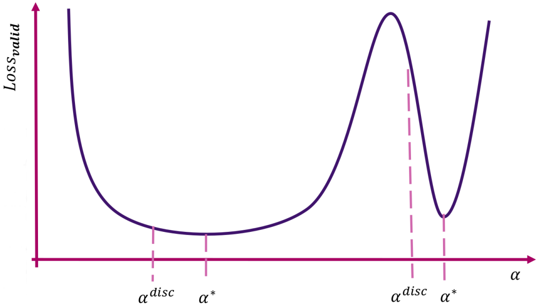
Outline

IBM Research

DARTS Discretization

IBM Research

- DARTS found a sharp local minima
- Validation loss increased on discretization
- Need to find smoother local minima

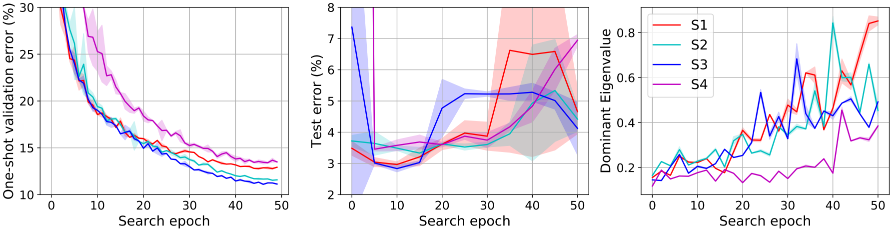


Arber Zela et al. "Understanding and Robustifying Differentiable Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020

DARTS Discretization

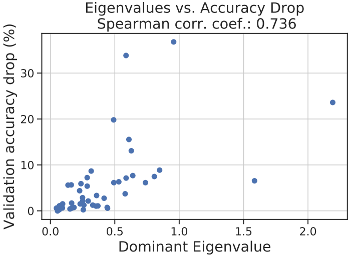
IBM Research

- RobustDARTS studied the relationship between the eigenvalues of the Hessian matrix of validation loss $\nabla^2_{\alpha(\beta)} L_{valid}$ and the generalization error.



Accuracy Drop During Discretization Step

IBM Research



- Early stop if $\lambda_{max}^{-\alpha}$(i-k) / $\lambda_{max}^{-\alpha}$(i) < 0.75
- Increase l2 regularization of the network weights
- Apply cutout augmentation along with scheduled drop path

Anneal and Prune

IBM Research

- Avoid discretization by gradually removing operations from the mixed operation
- Anneal each operation to make its strength:

$$\Phi_o(\alpha^{(i,j)}; T) = \frac{\exp(\frac{\alpha_o^{(i,j)}}{T})}{\sum_{o' \in \mathcal{O}} \exp(\frac{\alpha_{o'}^{(i,j)}}{T})} \quad (14)$$

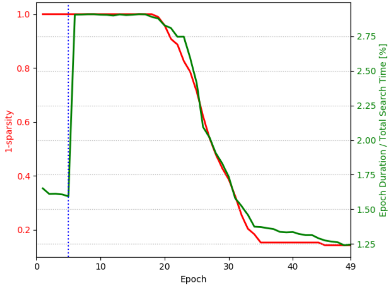
- Train SuperNet for some grace cycles τ
- For every iteration during bi-level optimization:
  - Prune operation if $\Phi_o(\alpha^{(i,j)}; T) <$ threshold
  - Update threshold and T

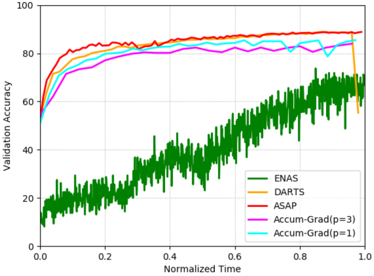
Asaf Noy et al. "ASAP: Architecture Search, Anneal and Prune". In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*. 2020, pp. 493–503

Anneal and Prune

IBM Research

- As SuperNet size reduces, search speed increases
- Searches on CIFAR-10 in 4.8 hours





Outline

IBM Research

Ranking Discrepancy of Weight-Sharing

IBM Research

Experiments are performed on reduced search space of NASBench-101 with 3 operations and 7 nodes.
Accuracy of NAS algorithms on 10 different searches.

NAS algo	Mean Acc.	Best Acc.	Best Rank	p($>$ random)
DARTS	92.21 ± 0.61	93.02	57079	0.24
NAO	92.59 ± 0.59	93.33	19552	0.62
ENAS	91.83 ± 0.42	92.54	96939	0.07
NAO w/o WS	93.08 ± 0.71	94.22	3543	0.92
ENAS w/o WS	93.54 ± 0.45	94.22	4610	0.90
Best Arch	90.93 ± 5.84	95.06		

Kaicheng Yu et al. "Evaluating The Search Phase of Neural Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020

Ranking Discrepancy of Weight-Sharing

IBM Research

- Correlation between the architecture rankings found with and without weight-sharing for 200 architectures.

#Nodes	**Kendall's** τ
4	0.441
5	0.314
6	0.214
7	0.195

Effective Training of One-Shot Architectures

IBM Research

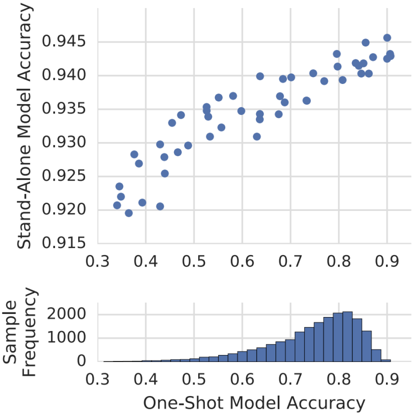
- Operations in one-shot model are subjected to co-adaptation
- Removing operations deteriorates performance
- Add dropout for every operation
- Use a variant of batch-normalization
- Apply L2 normalization only for the selected paths

Gabriel Bender et al. "Understanding and Simplifying One-Shot Architecture Search". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2018, pp. 549–558

Effective Training of One-Shot Architectures

IBM Research

- Sample 2000 architectures
- Train from scratch for 28 epochs



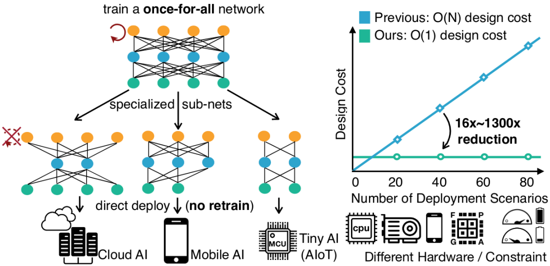
Outline

IBM Research

Once-for-All

IBM Research

- A single network is trained to support versatile architectural configurations including depth, width, kernel size, and resolution.
- Training is difficult since weights will interfere with each other.
- Progressive shrinking: train the largest network and then fine-tune the network to support smaller sub-networks.

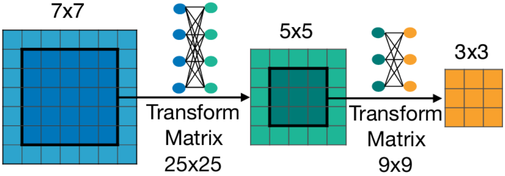


Han Cai et al. "Once-for-All: Train One Network and Specialize it for Efficient Deployment". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020

Elastic Kernel Size

IBM Research

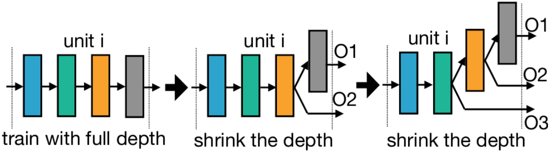
- Large kernels also serve as kernel for smaller sizes.
- However, forcing the weights to be the same degrades performance.
- Hence, a kernel transformation is used which is shared across the filter.



Elastic Depth

IBM Research

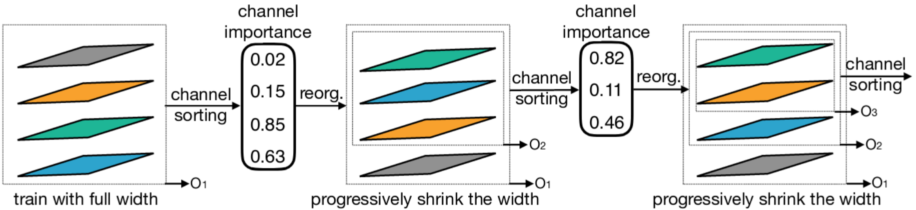
- Keep first layers and skip last ones.



Elastic Width

IBM Research

- Keep the channels with highest L1 norm.



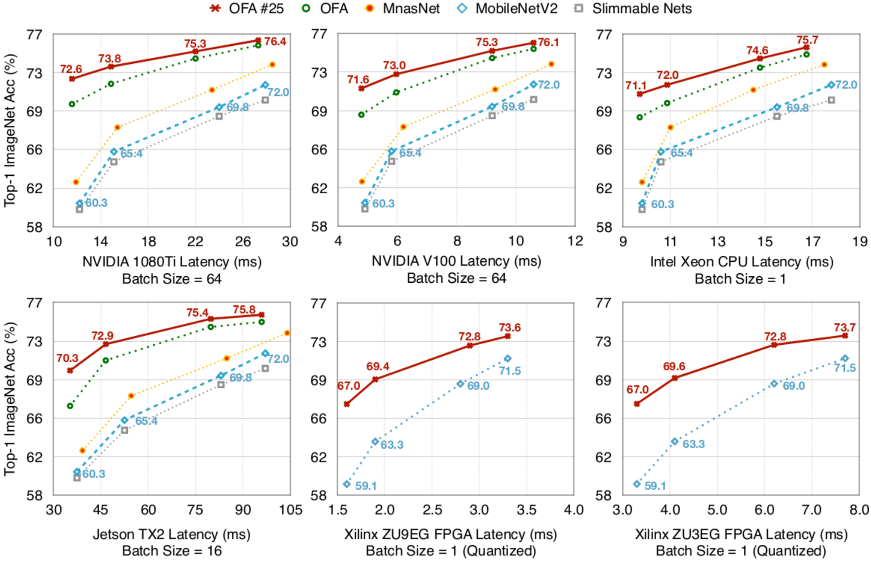
Search For Model With Constraints

IBM Research

- Learn a surrogate that predicts for an architecture its hardware requirements and accuracy.
- Training data for surrogate model is obtained by sampling different architectures from OFA.
- As sampled architectures are already trained, directly evaluate to obtain validation accuracy and constraint value.
- Surrogate model eliminates evaluation cost
- Use an evolutionary algorithm (Real et al.) to find an architecture that maximizes accuracy under the given efficiency constraints.

Results in Different Deployment Scenarios

IBM Research



Conclusion

IBM Research

- NASNet Search Space
- One-shot model flavour of optimizers
  - ENAS and DARTS
  - Drawbacks of DARTS
  - Problem ranking weight-shared models
  - Once for all network

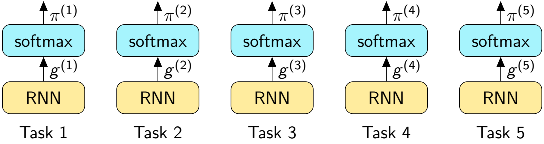
Outline

IBM Research

Motivation for Transfer Learning

IBM Research

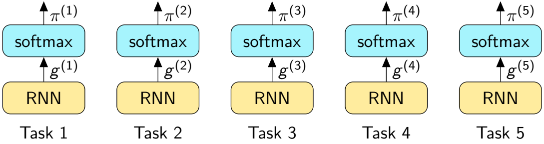
- Standard NAS methods solve every problem independently.
- No knowledge is shared between different optimization problem.
- Every search starts from scratch again.



Motivation for Transfer Learning

IBM Research

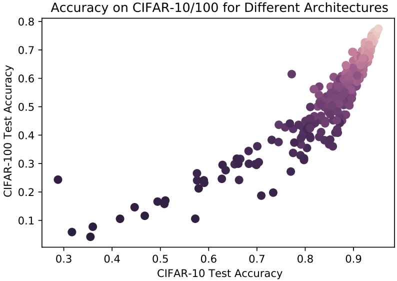
- Standard NAS methods solve every problem independently.
- No knowledge is shared between different optimization problem.
- Every search starts from scratch again.

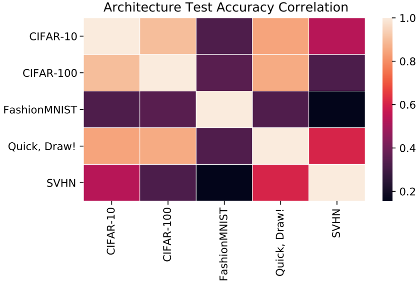


- **Can you reuse the knowledge of source tasks 1 to n for a new target task $n+1$?**

Metadata for Architecture Selection

IBM Research



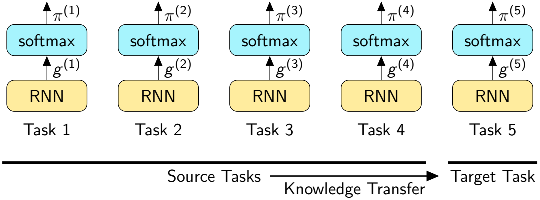


- Strong architecture test accuracy correlation across tasks
- A good architecture on one task is very likely a good candidate for another

Motivation for Transfer Learning

IBM Research

How can we use metadata to improve Neural Architecture Search?



Agenda

IBM Research

- We cover various ways of using the *metadata* (knowledge of source tasks) to improve NAS on the target task.

Agenda

IBM Research

- We cover various ways of using the *metadata* (knowledge of source tasks) to improve NAS on the target task.
- Transfer Neural Architecture Search
  - Methods that incorporate transfer learning methods directly into NAS methods.

Agenda

IBM Research

- We cover various ways of using the *metadata* (knowledge of source tasks) to improve NAS on the target task.
- Transfer Neural Architecture Search
  - Methods that incorporate transfer learning methods directly into NAS methods.
- Few-Shot Learning
  - Methods that combine NAS with meta-learning.

Agenda

IBM Research

- We cover various ways of using the *metadata* (knowledge of source tasks) to improve NAS on the target task.
- Transfer Neural Architecture Search
  - Methods that incorporate transfer learning methods directly into NAS methods.
- Few-Shot Learning
  - Methods that combine NAS with meta-learning.
- Learning Curve Prediction
  - Methods that accelerate NAS methods by using early stopping methods that use transfer learning.

Outline

IBM Research

Trainless Accuracy Predictor Architecture Search

IBM Research

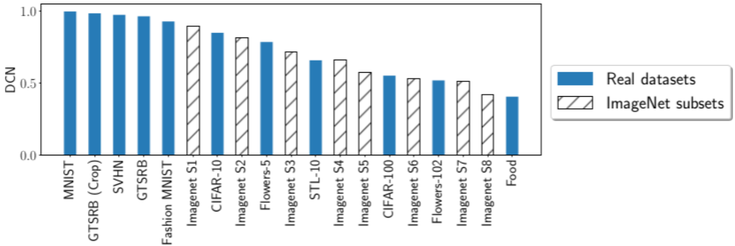
- TAPAS is a zero-shot Neural Architecture Search algorithm
- The best architecture is searched using an evolutionary algorithm
- Instead of training and evaluating each architecture, a surrogate model is used
- This surrogate model is trained on metadata from similar datasets

Roxana Istrate et al. "TAPAS: Train-less Accuracy Predictor for Architecture Search". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA*. 2019

Dataset Similarity

IBM Research

- A dataset is defined by its difficulty (DCN)
- The DCN is defined by the validation accuracy obtained by a fixed architecture (landmarker)
- Assumption: datasets are similar iff their DCN is similar



Metadata (LDE)

IBM Research

- 11 publicly available datasets and 8 datasets generated from ImageNet.
- 800 architectures are trained per datasets.
- Architectures are trained incrementally, adding one layer at a time.

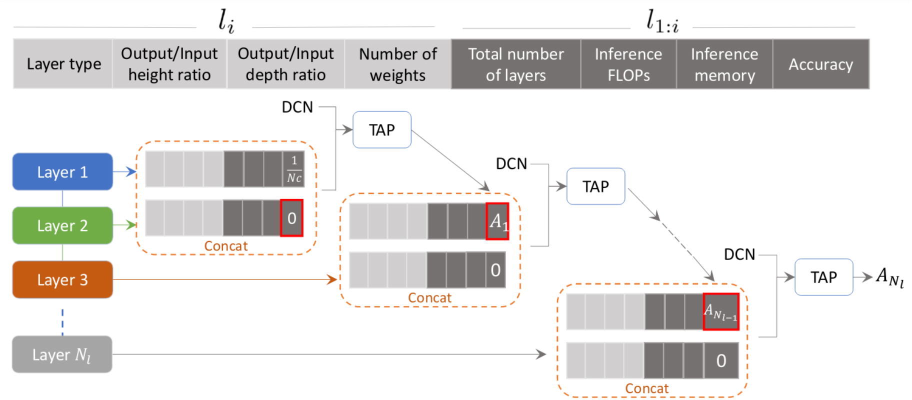
Surrogate Model

IBM Research

- Surrogate model (TAP) is designed using two stacked LSTMs
- Datasets with DCN similar to given dataset are selected.
- TAP is trained on the corresponding metadata.
- Architecture Encoding and DCN are inputs

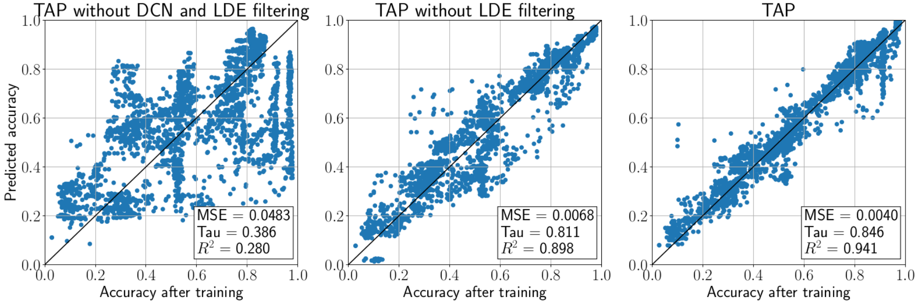
Encoding and Architecture Search

IBM Research



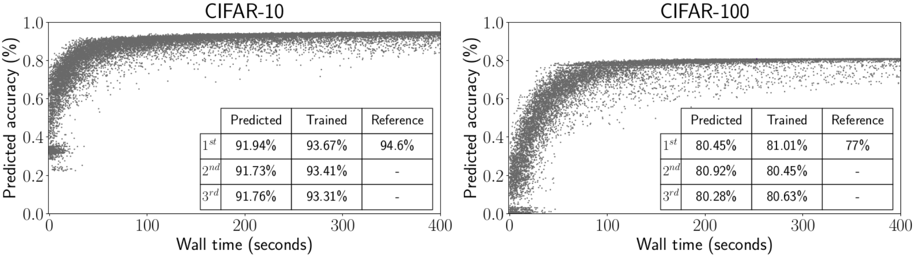
TAP Predictions

IBM Research



TAPAS Simulated Search

IBM Research



CIFAR-10

	Predicted	Trained	Reference
1^{st}	91.94%	93.67%	94.6%
2^{nd}	91.73%	93.41%	-
3^{rd}	91.76%	93.31%	-

CIFAR-100

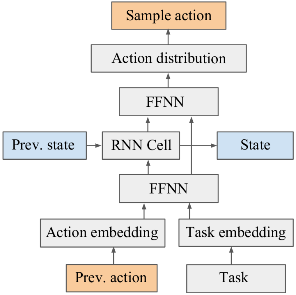
	Predicted	Trained	Reference
1^{st}	80.45%	81.01%	77%
2^{nd}	80.92%	80.45%	-
3^{rd}	80.28%	80.63%	-

- TAPAS simulates large-scale evolution of image classifiers algorithm[1]
- The algorithm took 250 hours

[1] Esteban Real et al. "Large-Scale Evolution of Image Classifiers". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 2902–2911

Transfer Learning with Neural AutoML

IBM Research



- RL controller is pretrained in a multi-task setting to optimize architectures for several tasks.
- The task embeddings helps learn across tasks

Catherine Wong et al. "Transfer Learning with Neural AutoML". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 2018, pp. 8366–8375

Transfer Learning with Neural AutoML

IBM Research

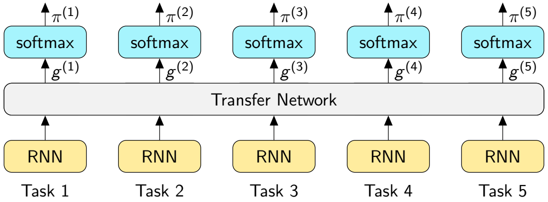
- Search space is FFNN models with embedding layer.
- Some of the Architecture choices:
  - input embedding
  - to fine tune or not
  - number of hidden layers
  - hidden layer sizes
  - further hyperparameters
- Controller pretrained on 8 tasks.
- Each optimizer has 500 trials.

Dataset	**RS**	**NAML**	**T-NAML**
20 Newsgroups	87.5	87.4	**88.1±0.4**
Brown Corpus	37.0	38.2	**53.4±3.3**
SMS Spam	97.9	97.8	**98.1±0.1**
Corp Messaging	**90.0**	**90.2**	**90.2±0.3**
Disasters	81.7	81.5	**82.1±0.3**
Emotion	33.9	33.7	**35.3±0.3**
Global Warming	82.4	**82.8**	**82.9±0.3**
Prog Opinion	68.9	66.3	**70.3±0.9**
Customer Reviews	77.8	79.0	**81.4±0.5**
MPQA Opinion	87.9	87.9	**88.6±0.3**
Sentiment Cine	73.2	**76.3**	75.4±0.4
Sentiment IMDB	85.8	87.3	**88.1±0.1**
Subj Movie	92.6	93.2	**93.4±0.2**

XferNAS

IBM Research

- Warmstart:
  - Learn an initial policy which does better than random.
- Minimally invasive:
  - Easy integration.
  - Converge to the original NAS optimizer's behavior.
- Solution: share weights across tasks.



Martin Wistuba. "XferNAS: Transfer Neural Architecture Search". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020*

Transfer Network

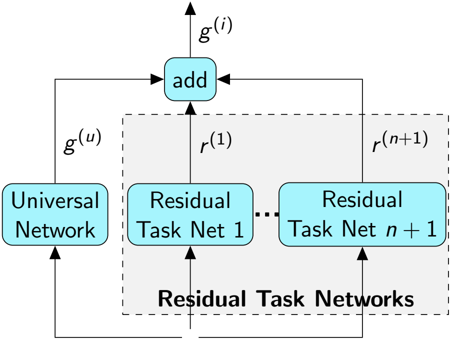
IBM Research

Core idea is to separate the task-dependent function $g^{(i)}$ into

- a universal function $g^{(u)}$ (warmstart initialization) and
- a task-dependent residual $r^{(i)}$.

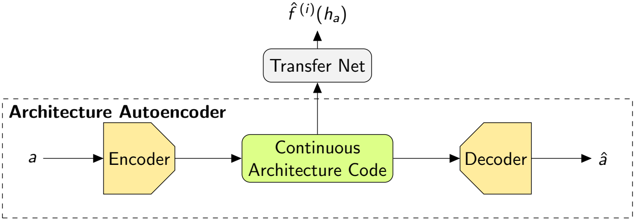
Thus,

$$g^{(i)} = g^{(u)} + r^{(i)} . \tag{15}$$



XferNAS

IBM Research



Example: Integration into NAO.

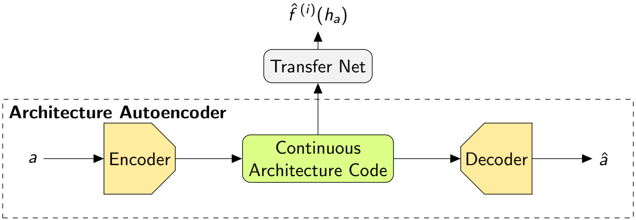
- Auto-encoder with surrogate model $\hat{f}$ that predicts the accuracy of an architecture based on its code h.

$$L = \alpha L_{\text{pred}} + (1 - \alpha) L_{\text{rec}} \tag{16}$$

  - L_{pred}: error when predicting accuracy.
  - L_{rec}: auto-encoder reconstruction loss.

XferNAS - Search

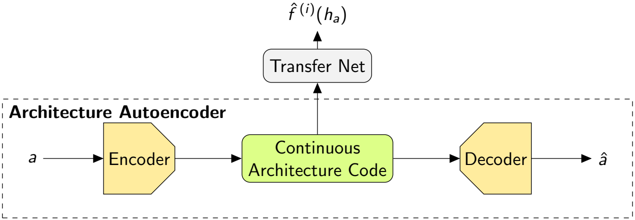
IBM Research



1. Solve $h_a^\star = \arg\max_{h_a} \hat{f}^{(i)}(h_a)$.
2. Estimate $a^\star = \text{Decoder}(h_a^\star)$.
3. Evaluate $f^{(i)}(a^\star)$.
4. Update the prediction model.
5. Go to 1.

XferNAS vs. NAO

IBM Research



Advantages of XferNAS over NAO:

- Auto-encoder is trained at the beginning of the search.
- Knowledge is leverage to warmstart the search.

Results on CIFAR-10

IBM Research

Model	F	#op	Err	#pms	M	GPU Days
NASNet-A	32	13	3.41	3.3M	20000	2000
AmoebaNet-B	36	19	3.37	2.8M	27000	3150
AmoebaNet-B (c/o)	128	19	2.13	34.9M	27000	3150
PNAS	48	8	3.41	3.2M	1280	225
NAONet	36	11	3.18	10.6M	1000	200
NAONet (c/o)	128	11	2.11	128M	1000	200
TAPAS	/	/	6.33	2.7M	1	0
T-NAML	/	/	3.5	N/A	150	N/A
Best on CIFAR-100	32	19	4.14	6.1M	200	/
XferNASNet	32	19	3.37	4.5M	33	6
XferNASNet (c/o)	32	19	2.70	4.5M	33	6
XferNASNet	64	19	3.11	17.5M	33	6
XferNASNet (c/o)	64	19	2.19	17.5M	33	6
XferNASNet (c/o)	128	19	1.99	69.5M	33	6

Results on CIFAR-10

IBM Research

Model	F	#op	Err	#pms	M	GPU Days
NASNet-A	32	13	3.41	3.3M	20000	2000
AmoebaNet-B	36	19	3.37	2.8M	27000	3150
AmoebaNet-B (c/o)	128	19	2.13	34.9M	27000	3150
PNAS	48	8	3.41	3.2M	1280	225
NAONet	36	11	3.18	10.6M	1000	200
NAONet (c/o)	128	11	2.11	128M	1000	200
TAPAS	/	/	6.33	2.7M	1	0
T-NAML	/	/	3.5	N/A	150	N/A
Best on CIFAR-100	32	19	4.14	6.1M	200	/
XferNASNet	32	19	3.37	4.5M	33	6
XferNASNet (c/o)	32	19	2.70	4.5M	33	6
XferNASNet	64	19	3.11	17.5M	33	6
XferNASNet (c/o)	64	19	2.19	17.5M	33	6
XferNASNet (c/o)	128	19	1.99	69.5M	33	6

Results on CIFAR-10

IBM Research

Model	F	#op	Err	#pms	M	GPU Days
NASNet-A	32	13	3.41	3.3M	20000	2000
AmoebaNet-B	36	19	3.37	2.8M	27000	3150
AmoebaNet-B (c/o)	128	19	2.13	34.9M	27000	3150
PNAS	48	8	3.41	3.2M	1280	225
NAONet	36	11	3.18	10.6M	1000	200
NAONet (c/o)	128	11	2.11	128M	1000	200
TAPAS	/	/	6.33	2.7M	1	0
T-NAML	/	/	3.5	N/A	150	N/A
Best on CIFAR-100	32	19	4.14	6.1M	200	/
XferNASNet	32	19	3.37	4.5M	33	6
XferNASNet (c/o)	32	19	2.70	4.5M	33	6
XferNASNet	64	19	3.11	17.5M	33	6
XferNASNet (c/o)	64	19	2.19	17.5M	33	6
XferNASNet (c/o)	128	19	1.99	69.5M	33	6

Results on CIFAR-10

IBM Research

Model	F	#op	Err	#pms	M	GPU Days
NASNet-A	32	13	3.41	3.3M	20000	2000
AmoebaNet-B	36	19	3.37	2.8M	27000	3150
AmoebaNet-B (c/o)	128	19	2.13	34.9M	27000	3150
PNAS	48	8	3.41	3.2M	1280	225
NAONet	36	11	3.18	10.6M	1000	200
NAONet (c/o)	128	11	2.11	128M	1000	200
TAPAS	/	/	6.33	2.7M	1	0
T-NAML	/	/	3.5	N/A	150	N/A
Best on CIFAR-100	32	19	4.14	6.1M	200	/
XferNASNet	32	19	3.37	4.5M	33	6
XferNASNet (c/o)	32	19	2.70	4.5M	33	6
XferNASNet	64	19	3.11	17.5M	33	6
XferNASNet (c/o)	64	19	2.19	17.5M	33	6
XferNASNet (c/o)	128	19	1.99	69.5M	33	6

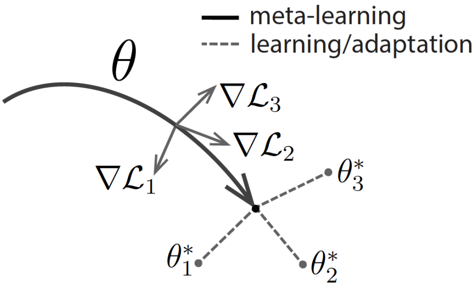
Outline

IBM Research

Model-Agnostic Meta-Learning

IBM Research

- Model learns from all the tasks
- Learn a representation that requires only few steps to the optimal representation for each task
- Performs well for few-shot learning problems



Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 1126–1135

MAML Algorithm

IBM Research

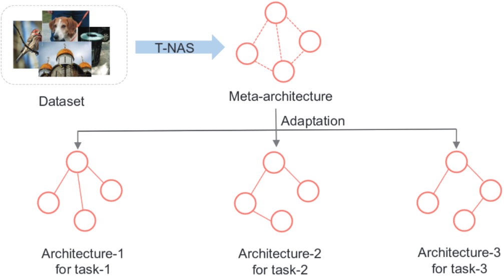
Algorithm 3 Model-Agnostic Meta-Learning

Input: $p(\mathcal{T})$: distribution over tasks
Input: β, γ: step size hyperparameters
1: randomly initialize θ
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: $\theta'_i = \theta - \beta \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
6: $\theta \leftarrow \theta - \gamma \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$

T-NAS

IBM Research

- Objective: Given multiple tasks, learn a meta-architecture
- Using bilevel optimization combined with MAML to estimate α and network weights w
- Finetune both parameters for each new task



Dongze Lian et al. "Towards Fast Adaptation of Neural Architectures with Meta Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020

T-NAS

IBM Research

Algorithm 4 T-NAS

Input: $p(\mathcal{T})$: distribution over tasks

randomly initialize α and w
while not done **do**
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ **do**
        Alternately update α' and w'
    Update α and w

Architecture Evaluation

IBM Research

Methods	Arch.	#Param.	1-shot	5-shot
Matching nets (Vinyals et al., 2016)	4CONV	32.9K	$43.44 \pm 0.77\%$	$55.31 \pm 0.73\%$
ProtoNets (Snell et al., 2017)	4CONV	32.9K	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
Meta-LSTM (Ravi & Larochelle, 2017)	4CONV	32.9K	$43.56 \pm 0.84\%$	$60.60 \pm 0.71\%$
Bilevel (Franceschi et al., 2018)	4CONV	32.9K	$50.54 \pm 0.85\%$	$64.53 \pm 0.68\%$
CompareNets (Sung et al., 2018)	4CONV	32.9K	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
LLAMA (Grant et al., 2018)	4CONV	32.9K	$49.40 \pm 1.83\%$	-
MAML (Finn et al., 2017)	4CONV	32.9K	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
MAML (first-order) (Finn et al., 2017)	4CONV	32.9K	$48.07 \pm 1.75\%$	$63.15 \pm 0.91\%$
MAML++ (Antoniou et al., 2019)	4CONV	32.9K	$52.15 \pm 0.26\%$	$68.32 \pm 0.44\%$
Auto-Meta (small) (Kim et al., 2018)	Cell	28/28 K	$49.58 \pm 0.20\%$	$65.09 \pm 0.24\%$
Auto-Meta (large) (Kim et al., 2018)	Cell	98.7/94.0 K	$51.16 \pm 0.17\%$	$69.18 \pm 0.14\%$
BASE (Softmax) (Shaw et al., 2018)	Cell	1200K	-	$65.40 \pm 0.74\%$
BASE (Gumbel-Softmax) (Shaw et al., 2018)	Cell	1200K	-	$66.20 \pm 0.70\%$
Auto-MAML (ours)	Cell	23.2/26.1 K	$51.23 \pm 1.76\%$	$64.10 \pm 1.12\%$
T-NAS (ours)	Cell	24.3/26.5 K*	$52.84 \pm 1.41\%$	$67.88 \pm 0.92\%$
T-NAS++ (ours)	Cell	24.3/26.5 K*	$\mathbf{54.11 \pm 1.35\%}$	$\mathbf{69.59 \pm 0.85\%}$

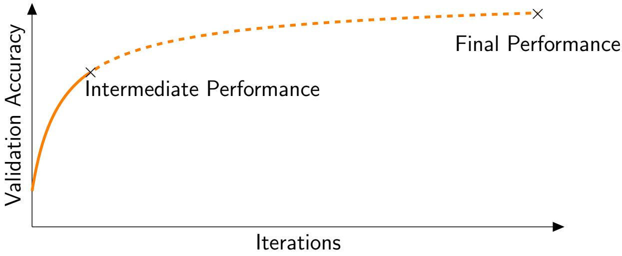
Outline

IBM Research

Motivation

IBM Research

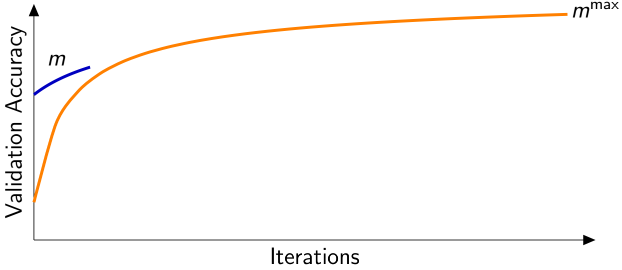
- Hyperparameter and neural architecture optimization are computationally expensive.
- Human experts decrease this effort by monitoring the model's learning curve and terminate options early that are unlikely to improve over the currently best solutions.
- With the rise of AutoML, a system that is able to perform this automatically is desired.



Use Simple Statistics

IBM Research

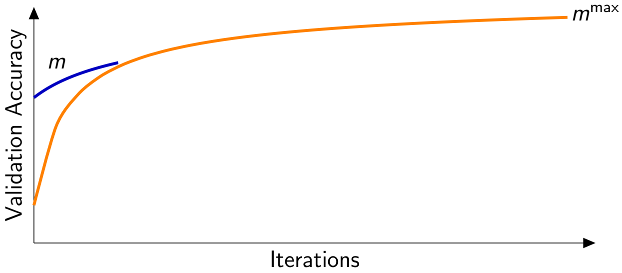
1. Use median/mean or last value in learning curve to make decision.



Use Simple Statistics

IBM Research

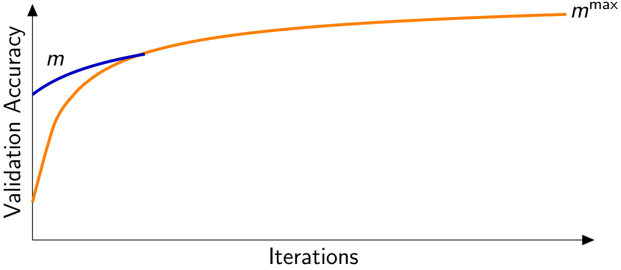
1. Use median/mean or last value in learning curve to make decision.



Use Simple Statistics

IBM Research

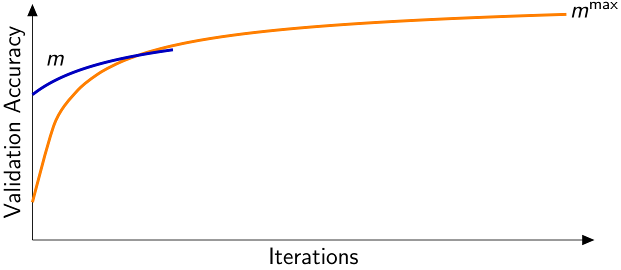
1. Use median/mean or last value in learning curve to make decision.



Use Simple Statistics

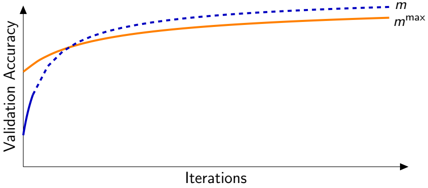
IBM Research

1. Use median/mean or last value in learning curve to make decision.



Simple Statistics - Problems

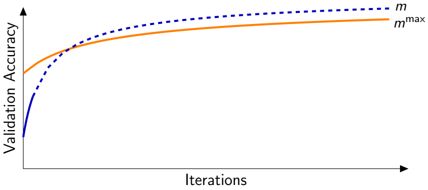
- Late bloomers will not be considered.



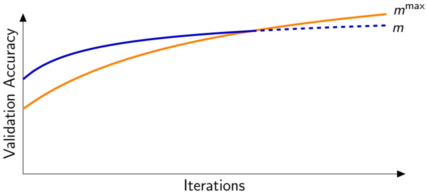
Simple Statistics - Problems

IBM Research

- Late bloomers will not be considered.



- Quick learners will be considered unnecessarily long.



Learning Curves Prediction

IBM Research

- Given a partial learning curve, predict the final performance.
- Use this prediction to estimate $p\,(m > m^{\text{max}})$.
- Terminate all runs with $p\,(m > m^{\text{max}}) \leq \delta$

Learning Curves Ranking

IBM Research

- Proposing to predict $p\left(m > m^{\max}\right)$ directly.
- Defining the probability that m_i is better than m_j as

$$p(m_i > m_j) = \hat{p}_{i,j} = \frac{e^{f(\mathrm{x}_i)-f(\mathrm{x}_j)}}{1+e^{f(\mathrm{x}_i)-f(\mathrm{x}_j)}}. \tag{17}$$

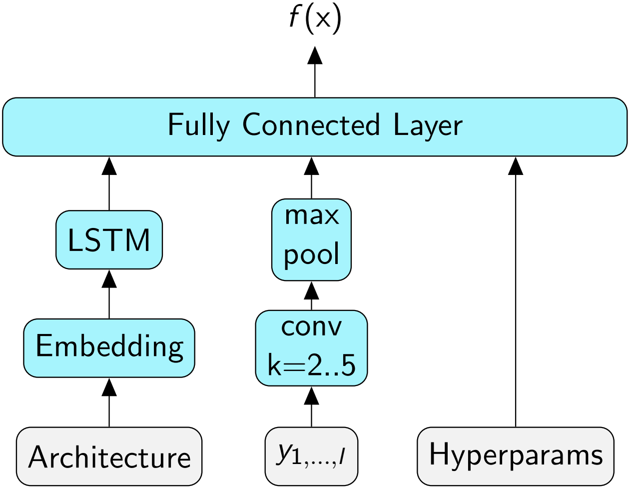
- Minimize the cross-entropy loss

$$\sum_{i,j} -p_{i,j} \log \hat{p}_{i,j} - (1-p_{i,j}) \log(1-\hat{p}_{i,j}) \tag{18}$$

Martin Wistuba and Tejaswini Pedapati. "Learning to Rank Learning Curves". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 12-18 June 2020, Vienna, Austria*. 2020

Modelling f

IBM Research



Learning Curves Ranking with Transfer Learning

IBM Research

- Learning requires data which is not available.

Learning Curves Ranking with Transfer Learning

IBM Research

- Learning requires data which is not available.
- Solution 1: Do not learn, only consider given partial learning curve.

Learning Curves Ranking with Transfer Learning

IBM Research

- Learning requires data which is not available.
- Solution 1: Do not learn, only consider given partial learning curve.
- Solution 2: First collect sufficient learning curves and then train your model.

Learning Curves Ranking with Transfer Learning

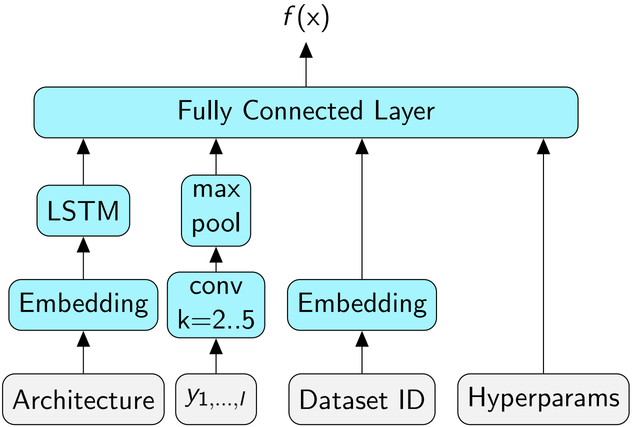
IBM Research

- Learning requires data which is not available.
- Solution 1: Do not learn, only consider given partial learning curve.
- Solution 2: First collect sufficient learning curves and then train your model.
- Proposal: Use transfer learning to reduce this problem.

Considering Transfer Learning in our Modelling

IBM Research

To account for transfer learning, an embedding per dataset is added.



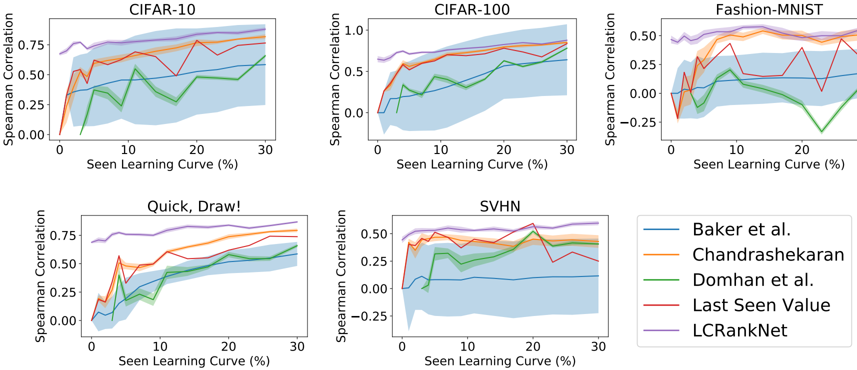
Setup

IBM Research

- Experiments are conducted on five different datasets: CIFAR-10, CIFAR-100, Fashion-MNIST, Quickdraw, and SVHN.
- To create the meta-knowledge, 200 architectures per dataset are choosen at random from the NASNet search space (i.e. 1000 unique architectures) and train it for 100 epochs.
- Experiments are conducted in a leave-one-dataset-out cross-validation.

Ranking Performance

IBM Research



Random Search

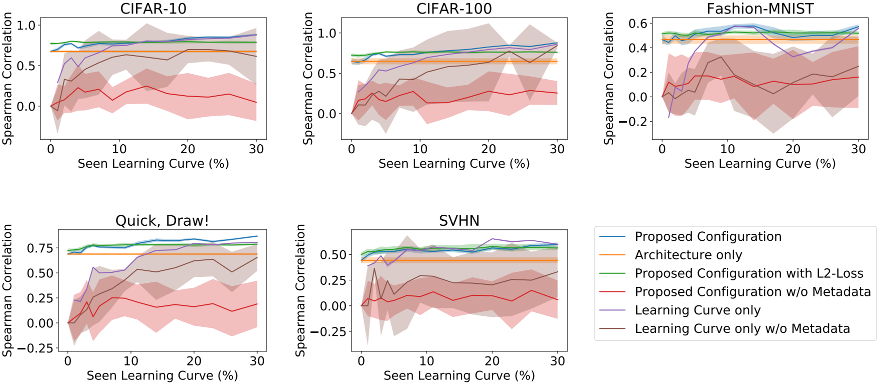
IBM Research

Random Neural Architecture Search with Early Stopping.

- Regret: Difference of best solution and best solution without early stopping.
- Time: GPU time in hours.

Method	CIFAR-10		CIFAR-100		Fashion		Quickdraw		SVHN	
	Regr.	Time	Regr.	Time	Regr.	Time	Regr.	Time	Regr.	Time
No Early Termination	0.00	1023	0.00	1021	0.00	1218	0.00	1045	0.00	1485
Domhan et al.	0.56	346	0.82	326	0.00	460	0.44	331	0.28	471
Hyperband	0.22	106	0.78	102	0.32	132	0.54	109	0.00	156
Baker et al.	0.00	89	0.00	77	0.00	129	0.00	107	0.00	241
Successive Halving	0.62	62	0.00	54	0.18	70	0.40	60	0.28	88
Chandrashekaran	0.62	30	0.00	35	0.28	41	0.30	82	0.06	164
LCRankNet	0.22	20	0.00	11	0.10	19	0.00	28	0.10	74

Component Analysis



Conclusions

IBM Research

- Transfer learning for Neural Architecture Search has been explored in various ways.
  - By means of special neural architecture search methods,
  - meta-learning,
  - and early termination techniques for incremental model training.
- All imply that it can be used to significantly decrease the computational effort for NAS.
- Yet, it is a relatively unexplored research topic.

Outline

1. Introduction

2. One-Shot Architecture Search

3. Transfer Learning for NAS

4. Conclusions

Final Conclusions

IBM Research

- A common search space for NAS.
- Various efficient optimizers based on parameter sharing and differentiable architecture search.
- Discussion of several problems being faced with these very methods.
- A deep dive on transfer learning for NAS.

IBM Research

Thank you for your attention.

Survey Paper: https://arxiv.org/abs/1905.01392

References I

IBM Research

Gabriel Bender et al. "Understanding and Simplifying One-Shot Architecture Search". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2018, pp. 549–558.

Simone Bianco et al. "Benchmark Analysis of Representative Deep Neural Network Architectures". In: *IEEE Access* 6 (2018), pp. 64270–64277.

Han Cai, Ligeng Zhu, and Song Han. "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware". In: *Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA*. 2019.

References II

IBM Research

Han Cai et al. "Once-for-All: Train One Network and Specialize it for Efficient Deployment". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.

Francesco Paolo Casale, Jonathan Gordon, and Nicolo Fusi. "Probabilistic Neural Architecture Search". In: *CoRR* abs/1902.05116 (2019).

Xiangxiang Chu et al. "Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search". In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*. 2020, pp. 465–480.

References III

IBM Research

Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 1126–1135.

Roxana Istrate et al. "TAPAS: Train-less Accuracy Predictor for Architecture Search". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA*. 2019.

Dongze Lian et al. "Towards Fast Adaptation of Neural Architectures with Meta Learning". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.

References IV

IBM Research

Hanxiao Liu, Karen Simonyan, and Yiming Yang. "DARTS: Differentiable Architecture Search". In: *Proceedings of the International Conference on Learning Representations, ICLR 2019, New Orleans, Louisiana, USA*. 2019.

Renqian Luo et al. "Neural Architecture Optimization". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 2018, pp. 7827–7838.

Asaf Noy et al. "ASAP: Architecture Search, Anneal and Prune". In: *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*. 2020, pp. 493–503.

References V

IBM Research

- Hieu Pham et al. "Efficient Neural Architecture Search via Parameter Sharing". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 2018, pp. 4092–4101.
- Esteban Real et al. "Aging Evolution for Image Classifier Architecture Search". In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA*. 2019.
- Esteban Real et al. "Large-Scale Evolution of Image Classifiers". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 2902–2911.

References VI

IBM Research

Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* 2017, pp. 4278–4284.

Martin Wistuba. "XferNAS: Transfer Neural Architecture Search". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020*.

Martin Wistuba and Tejaswini Pedapati. "Learning to Rank Learning Curves". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 12-18 June 2020, Vienna, Austria*. 2020.

References VII

IBM Research

- Catherine Wong et al. "Transfer Learning with Neural AutoML". In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 2018, pp. 8366–8375.
- Yuhui Xu et al. "PC-DARTS: Partial Channel Connections for Memory-Efficient Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.
- Kaicheng Yu et al. "Evaluating The Search Phase of Neural Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.

References VIII

Arber Zela et al. "Understanding and Robustifying Differentiable Architecture Search". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.