Optimizing Neural Network Structures: Faster Speed, Smaller Size, Less Tuning

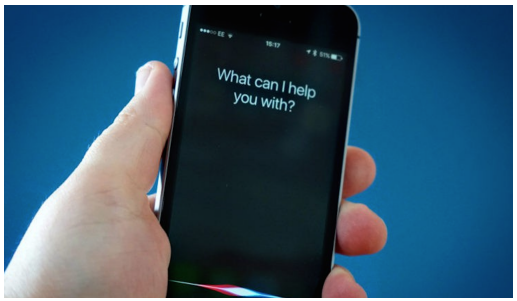
Zhe Li

Advisor: Prof. Tianbao Yang

The University of Iowa

Friday 1st June, 2018

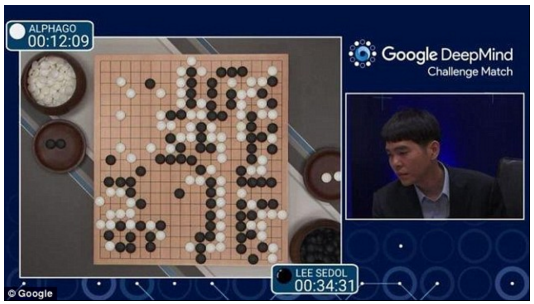
The success of deep learning



Speech Recognition

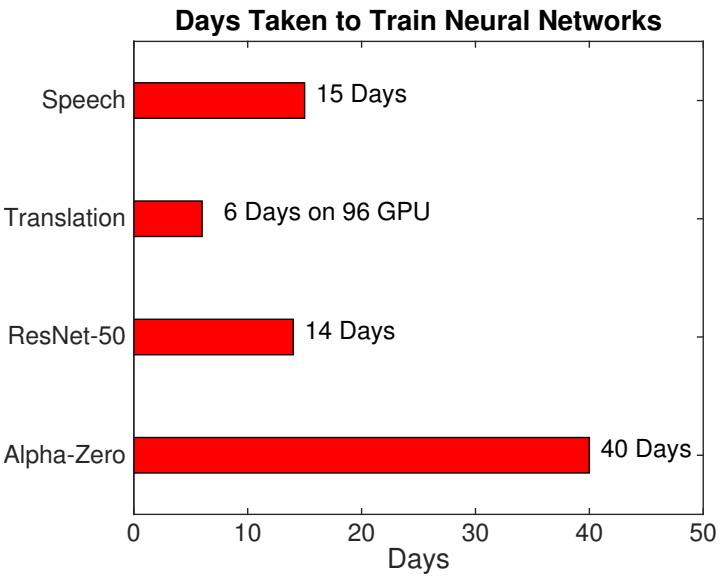
Language Translation

Face Recognition



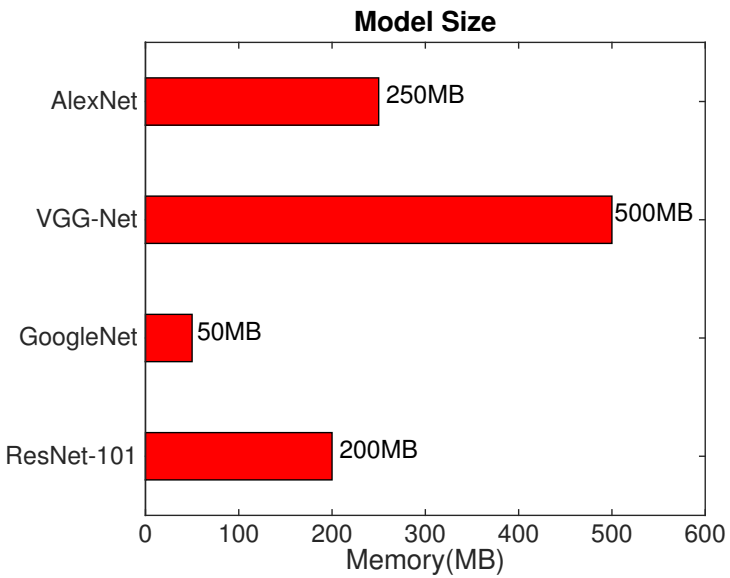
Game

The 1st challenge: slow training



- Training deep neural network takes days to weeks.
- Can we accelerate training neural networks?

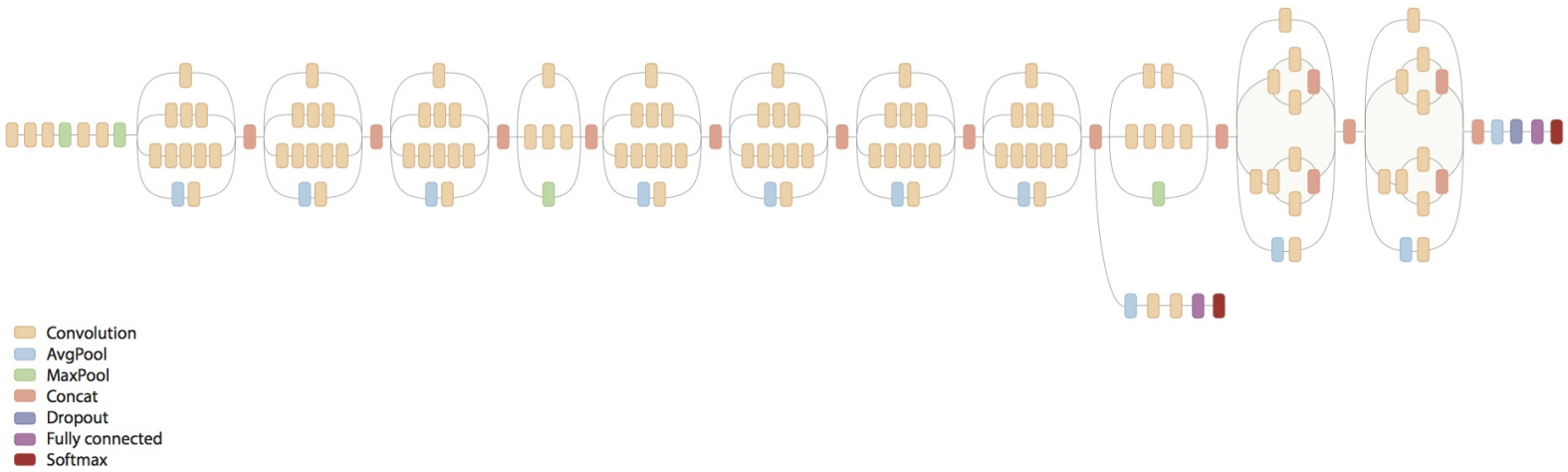
The 2nd challenge: model size too large



- Neural network models are too large to deploy to
  - Mobile device
  - Embedded device
- Can we reduce the size of neural network without large performance loss?

The 3rd challenge: how to design neural network?

- Designing network structure requires extensive human efforts.



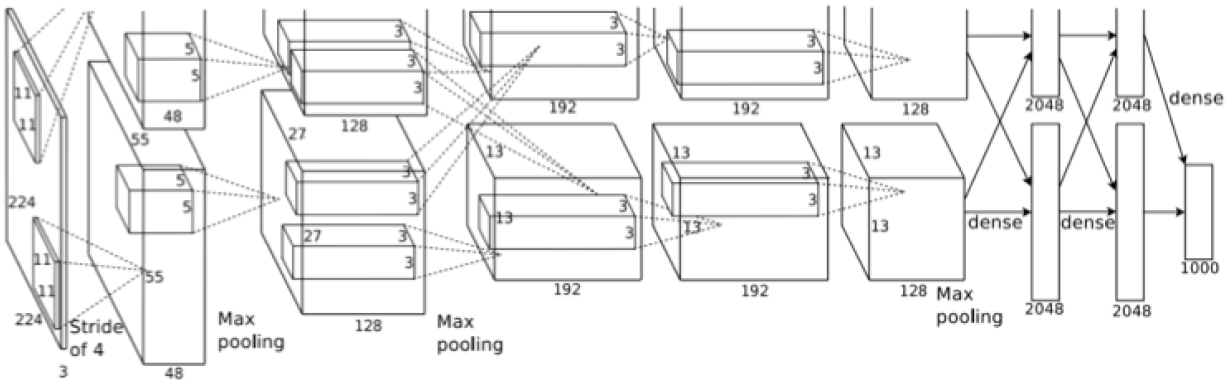
- Can we automatically design neural networks with less tuning?

1. Faster Training
   - Improved Dropout for Deep Learning
   - Experimental Results
2. Small and Effective Pattern Networks (SEP-Nets)
   - The Proposed Method
   - The Ingredients for SEP-Nets
   - Experimental Results
3. Evolution Algorithm for Searching Optimal Neural Networks
   - Motivation and Related works
   - Exploring Genetic Approach
   - Experimental Results
4. Ecologically-Inspired Approach for Searching Networks
   - Motivation
   - Ecologically-Inspired Approach
   - Experimental Results
5. Conclusion

Outline

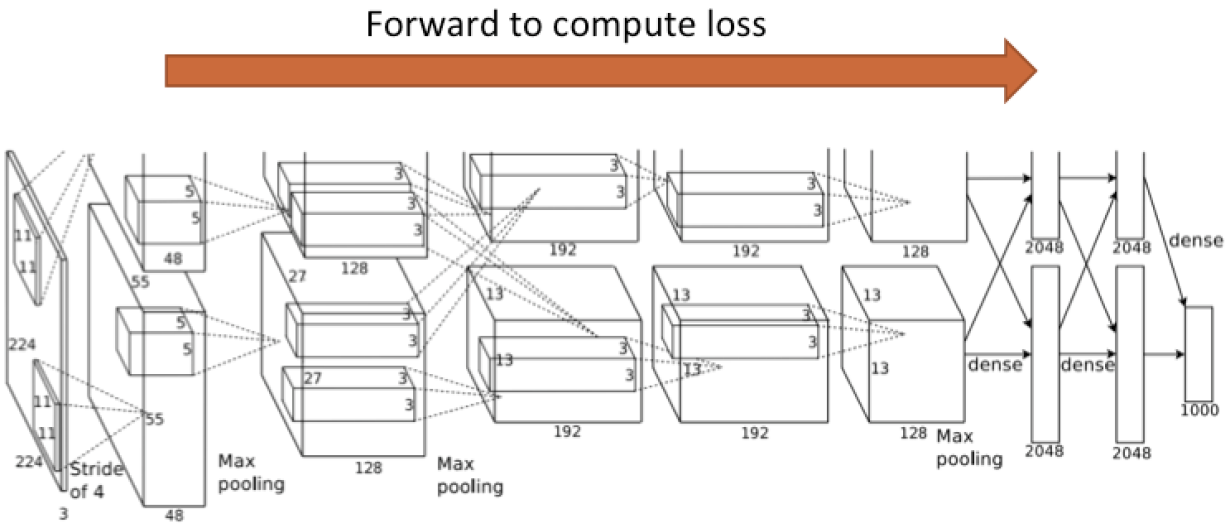
Deep neural network

- The classical example: AlexNet[5]



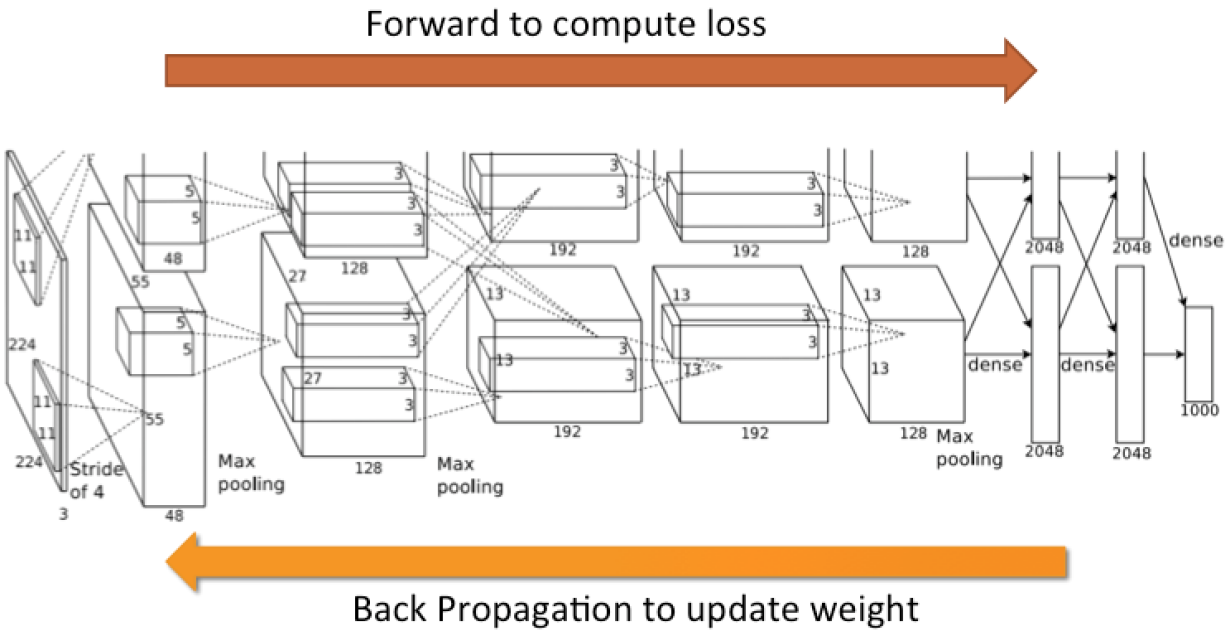
Deep neural network

- The classical example: AlexNet[5]



Deep neural network

- The classical example: AlexNet[5]

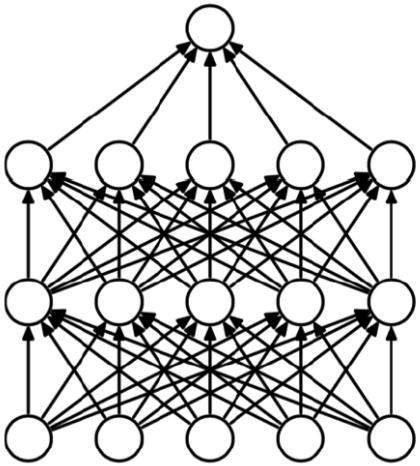


$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(\textit{prediction}, \textit{truth}) \tag{1}$$

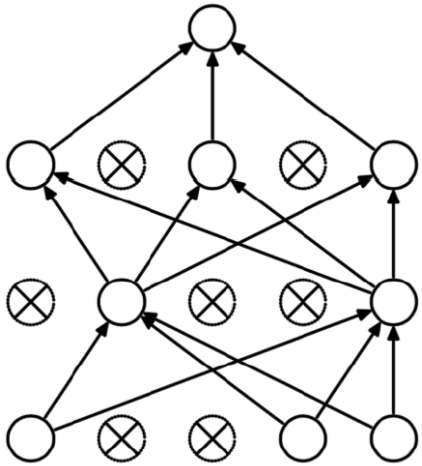
The entire neural network architeture has 60 million paraemters.

What is dropout?

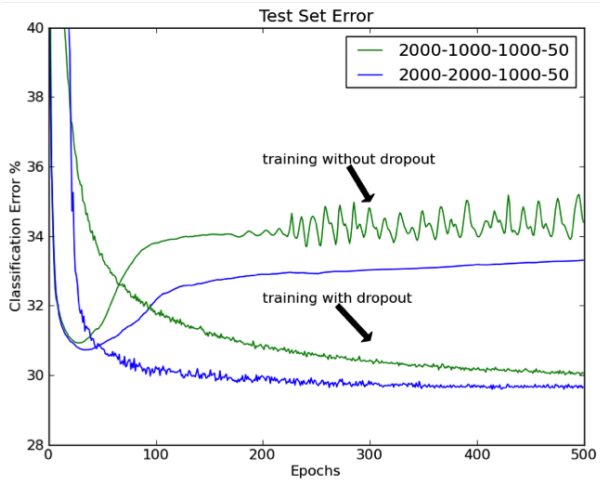
- Dropout: A simple effective way to prevent overfitting[8].
  - randomly drops (with probability 0.5) some neurons in training.



(a) Standard Neural Net



(b) After applying dropout.

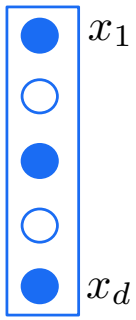


How to sample neurons for dropout?

- Uniformly random (neurons are treated equally)
  - Intuitively, this is not the optimal way.
  - some neurons may be more important (some features may be more important)
- What is the optimal distribution to drop neurons?
  - Generalization error perspective.

Analyze dropout

- Introduce multinomial distribution for dropout



$$\mathbf{p} = (p_1, p_2, \cdots, p_d) \quad \sum_{i=1}^{d} p_i = 1$$

- Analyze the dependence of generalization error on the sampling probabilities

$$\textit{generalization error} \leq e(n, \mathbf{p}, \mathcal{D}) \tag{2}$$

  - n: number of training data; $\mathbf{p}$: sampling probability; $\mathcal{D}$: distribution of data

Analyze dropout (rigorous manner)

Theorem 1:

Let $\mathcal{L}(\mathbf{w})$ be the expected risk of $\mathbf{w}$. Assume $\mathrm{E}_{\widehat{\mathcal{D}}}[\|\mathbf{x} \circ \boldsymbol{\epsilon}\|_2^2] \leq B^2$ and $\ell(z, y)$ is convex and G-Lipschitz continuous. For any $\|\mathbf{w}_*\|_2 \leq r$, by appropriately choosing η, we can have

$$\mathrm{E}[\mathcal{L}(\widehat{\mathbf{w}}_n)] \leq \mathcal{L}(\mathbf{w}_*) + R_{\mathcal{D},\mathcal{M}}(\mathbf{w}_*) + \frac{GBr}{\sqrt{n}}$$

How to prove the above theorem?

- Standard SGD analysis.
- Dropout is a data-dependent regularizer.

Improved dropout for shallow learning

- Minimizing the term $\mathrm{E}_{\widehat{\mathcal{D}}}[\|\mathbf{x} \circ \boldsymbol{\epsilon}\|_2^2]$ and the relaxed upper bound of term $R_{\mathcal{D},\mathcal{M}}(\mathbf{w}_*)$ yields the optimal sampling probabilities:

$$p_i^* = \frac{\sqrt{\mathrm{E}_{\mathcal{D}}[x_i^2]}}{\sum_{j=1}^{d} \sqrt{\mathrm{E}_{\mathcal{D}}[x_j^2]}}, i = 1, \ldots, d \tag{3}$$

- Can we compute the above probability for dropout?
  - ✗

Improved dropout for shallow learning

- Practically, we use the empirical second-order statistics to compute the probabilities:

$$p_i = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_i^2]}}{\sum_{i'=1}^{d}\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_{i'}^2]}}, i = 1, \ldots, d \qquad (4)$$

Improved dropout for deep learning

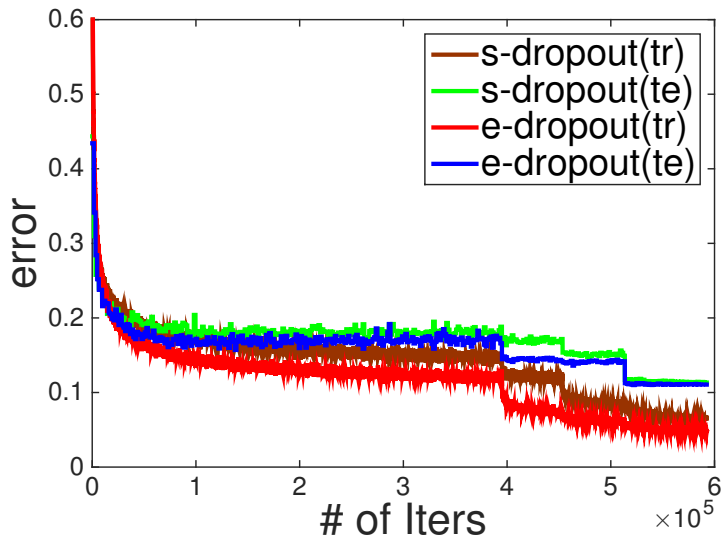
- Could we directly use the above idea to Deep Learning?
  - ✗
- Why not?
  - Too expensive to compute dropout probablity from all examples.
- How to address this issue?
  - Use a mini-batch of examples to calculate the dropout probablity.

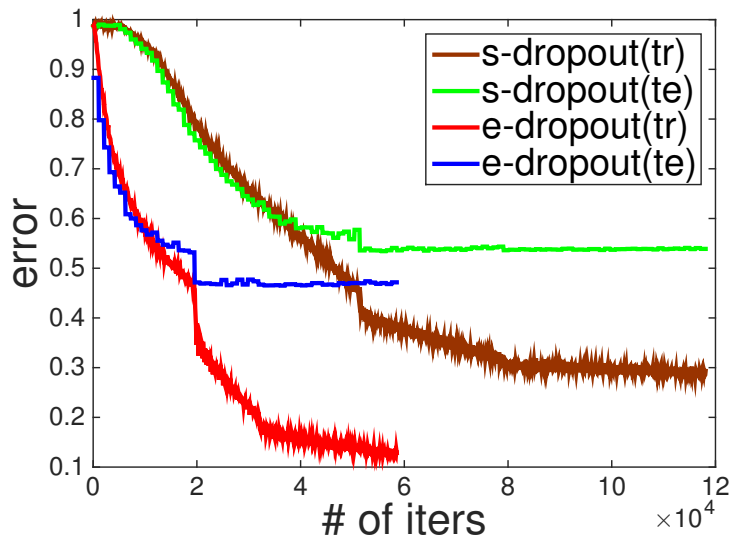
- Let $X^l = (\mathbf{x}_1^l, \ldots, \mathbf{x}_m^l)$ denote the outputs of the l^{th} layer for a mini-batch of m examples, calculate the probabilities for dropout by

$$p_i^l = \frac{\sqrt{\frac{1}{m}\sum_{j=1}^{m}[[\mathbf{x}_j^l]_i^2]}}{\sum_{i'=1}^{d}\sqrt{\frac{1}{m}\sum_{j=1}^{m}[[\mathbf{x}_j^l]_{i'}^2]}}, i = 1, \ldots, d \qquad (5)$$

Improved Dropout

- Dropping out the output of the neuron based on multinomial distribution computed from the training data.



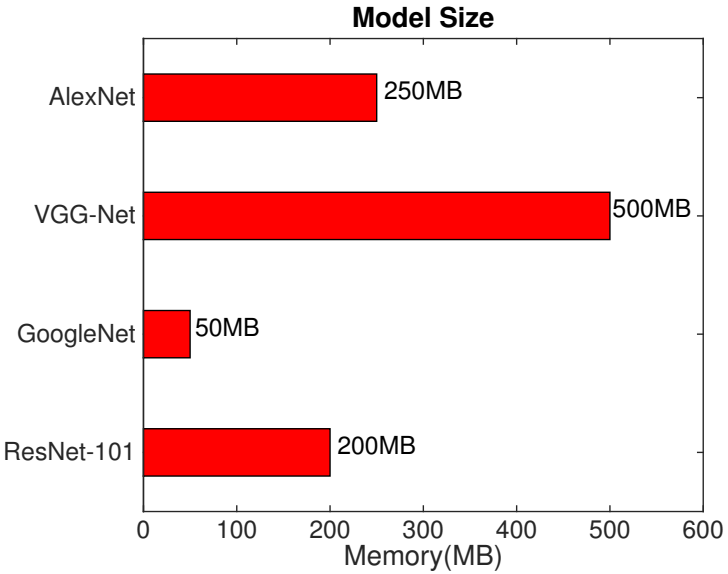


Contribution for faster training

- Proposed and theoretically analyzed a multinomial dropout for shallow learning.
- Proposed an efficient evolutional dropout for deep learning.
- Justified the proposed dropouts empirically.

Outline

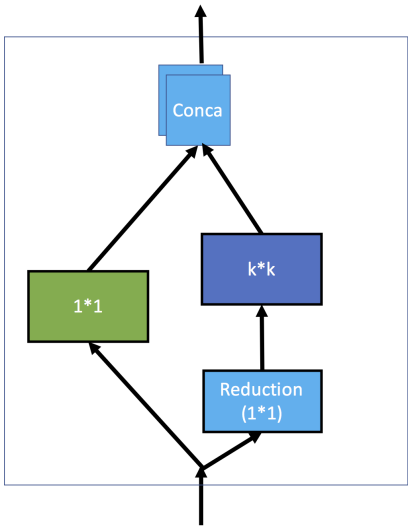
Where to focus to reduce model size?



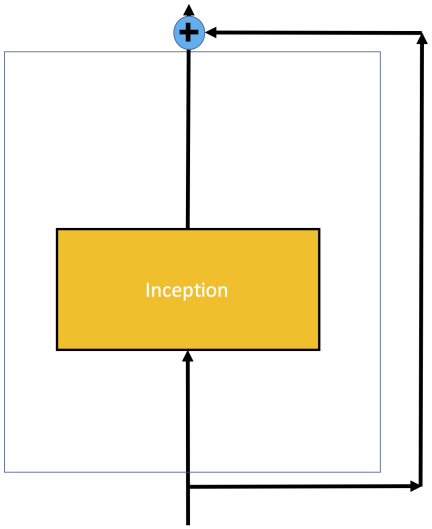
- Fully connected and convolution layers have most parameters in neural network models.
- Fully connected layers have been removed from GoogleNet, ResNet.

Focus on convolutional layers to reduce model size.

Zoom in GoogleNet and ResNet



Inception Module in GoogleNet



Residual Module in ResNet

Pattern binarization

- $k \times k (k > 1)$ filters serve as spatial pattern extraction.
- 1×1 filters serve as data transformation.
- Reduced number of parameters in model dramatically.

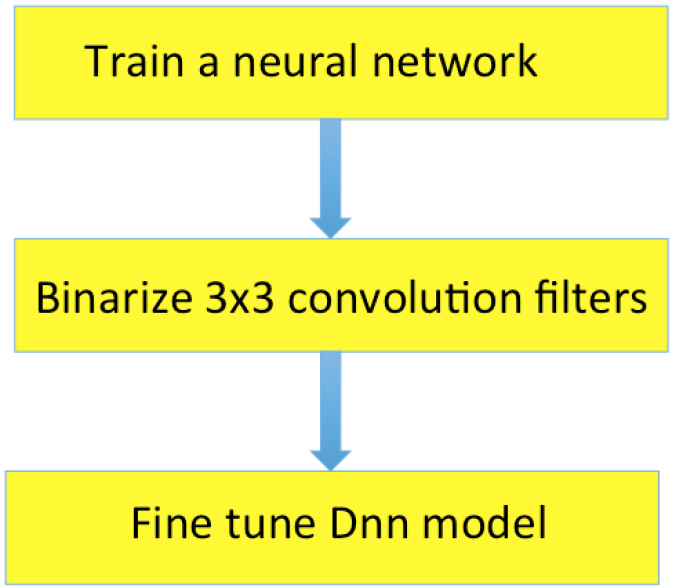
-0.0219	0.0408	-0.0547
-0.0855	0.0478	-0.0510
-0.0105	0.0924	-0.0126

-1	1	-1
-1	1	-1
-1	1	-1

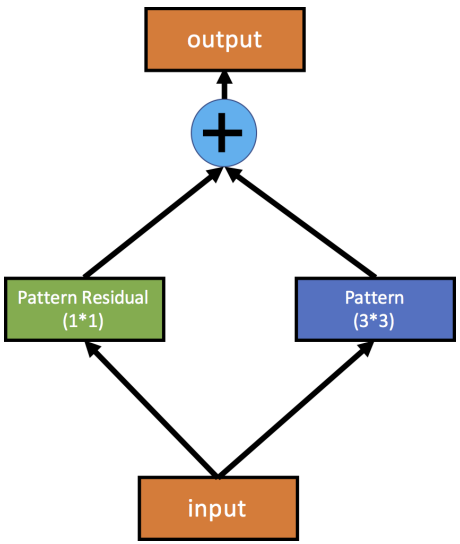
A trained 3×3 filter from GoogleNet (Left) and its binarized version (right)

How to use pattern binarization?

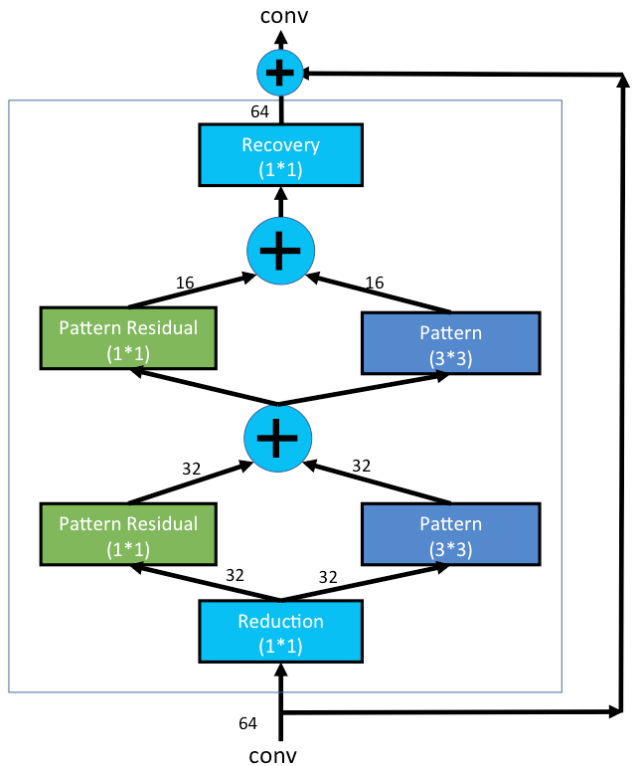
- Easily adopted to any successful networks structures such as GoogleNet, ResNet as following procedure:



Pattern Residual Block and SEP-Net module



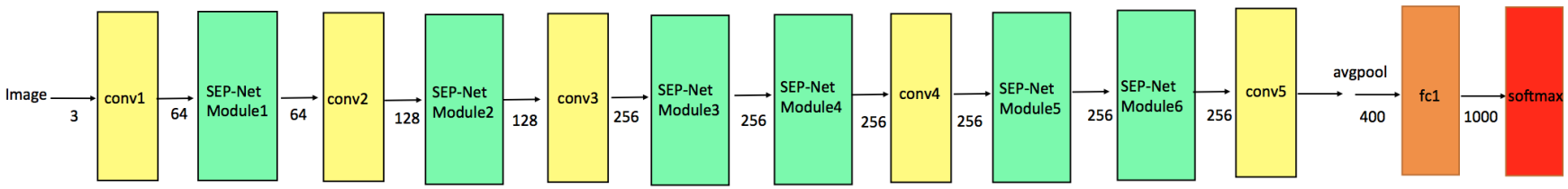
Pattern Residual Block



SEP-Net Module

The Proposed SEP-Net structures

- Proposed SEP-Net for mobile/embeded devices.
- The designed SEP-Net has 1.3M parameters (5.2MB).



Experimental results on Pattern Binarization

1,000 classes image recognition

model	Original Model		After Binarization	
	#Model size	Top1-Top5	#Model size	Top1-Top5
GoolgeNet	27.96MB	0.6865 0.8891	17.62MB	0.6797 0.8827
C-Inception	20.40MB	0.6480 0.8630	9.62MB	0.6400 0.8550

Experimental results for the designed SEP-Nets

- Small and Effective on the designed SEP-Nets

Model	Parameter number	Size (bytes)	Top-1 Acc
MobileNet[2]	1.3M	5.2MB	0.637
	2.6M	10.4MB	0.684
SEP-Net-R	1.3M (small)	5.2MB	0.658
	1.7M (large)	6.7MB	0.667
SqueezeNet[4]	1.2M	4.8MB	0.604
MobileNet	1.3M	5.2MB	0.637
SEP-Net-R (Small)	**1.3M**	**5.2MB**	**0.658**
SEP-Net-B (Small)	1.1M	4.2MB	0.637

SEP-Net-R: SEP-Net with raw valued weights

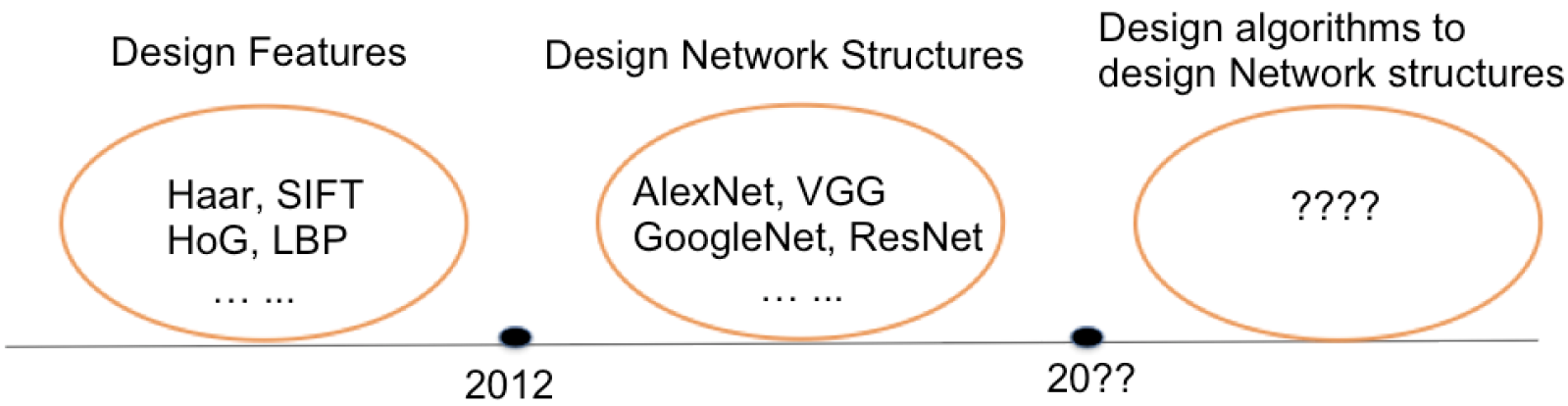
SEP-Net-B: SEP-Net with pattern binarization

Contribution to smaller size

- Proposed pattern binarization method.
- Designed a novel pattern residual block and SEP-Net Module.
- Proposed Small and Effective Pattern Networks.
- Achieved the-state-of-art performance.

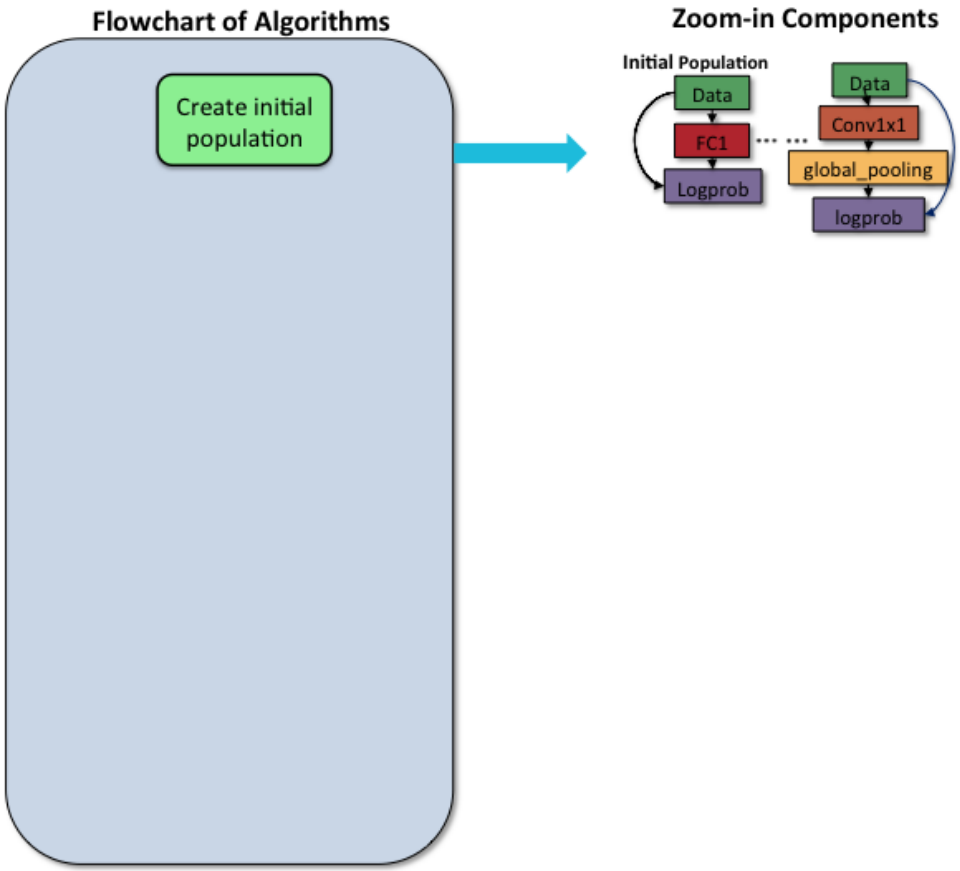
Outline

Motivation

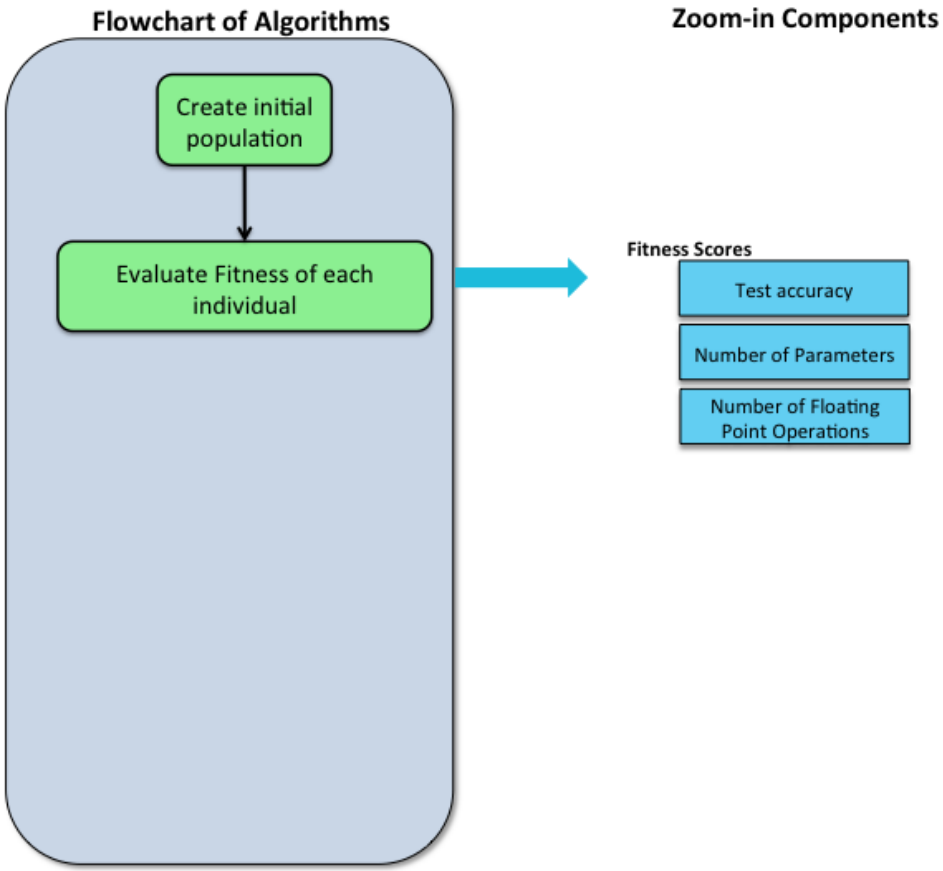


- We focus on developing efficient algorithms to design network structures.

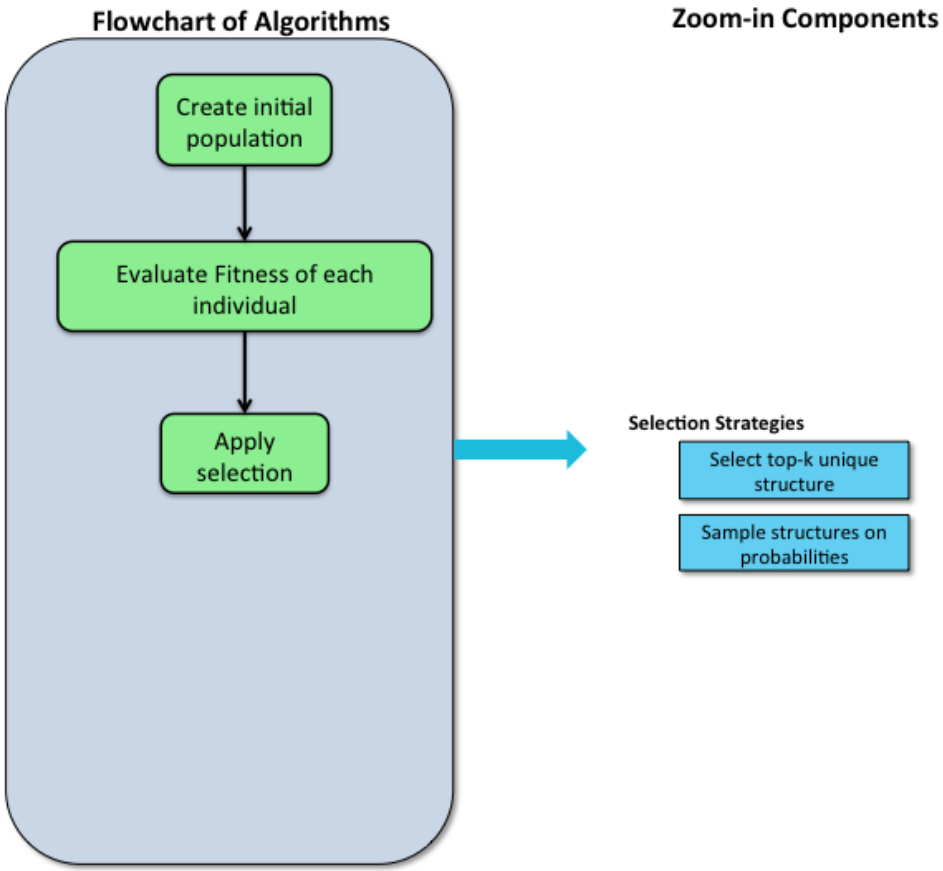
How does genetic approach work?



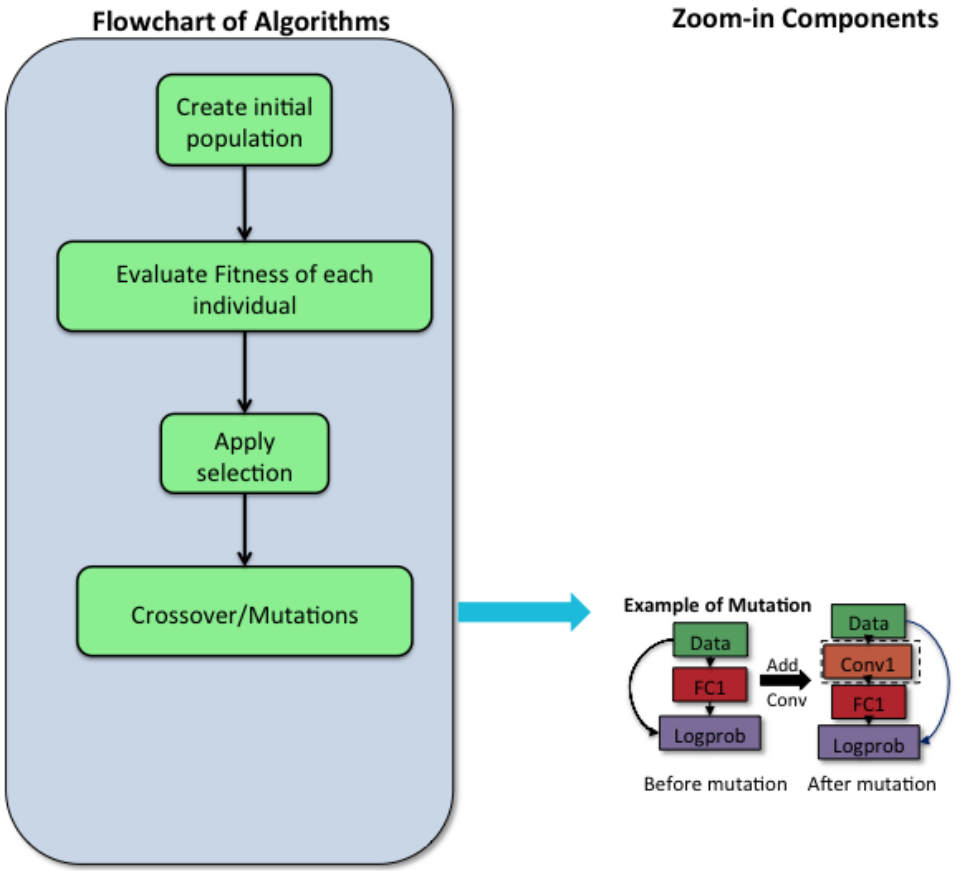
How does genetic approach work?



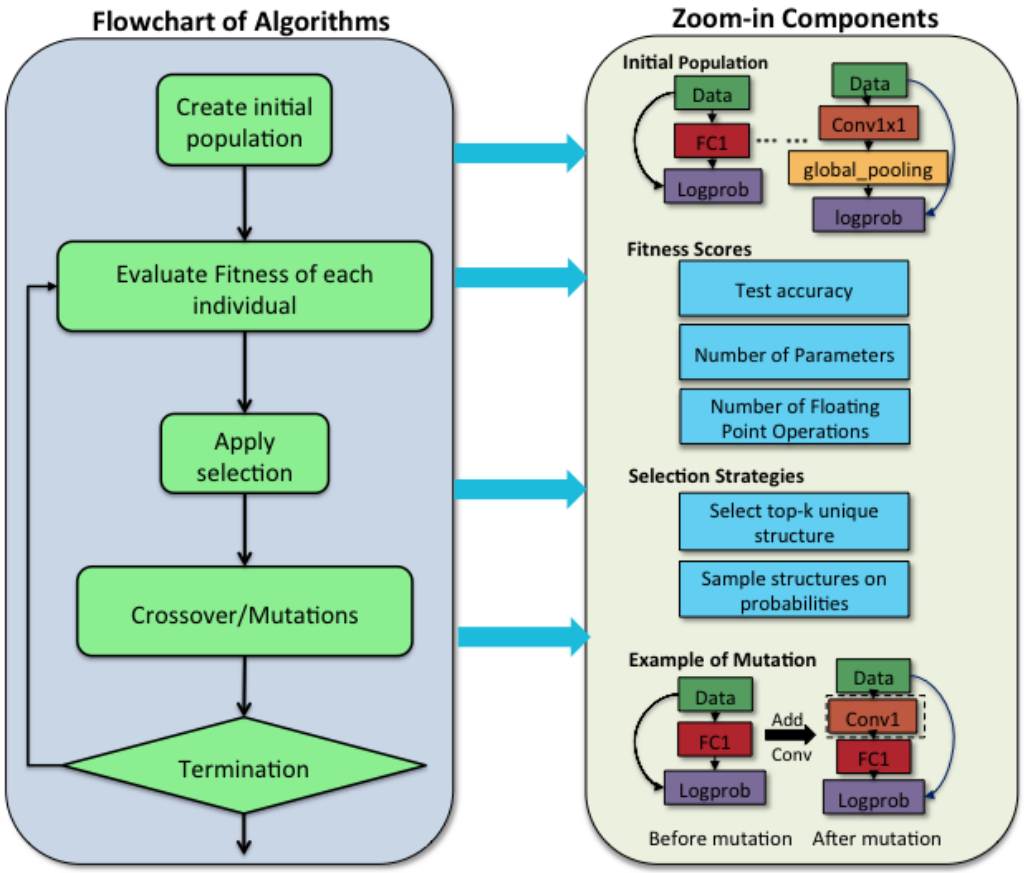
How does genetic approach work?



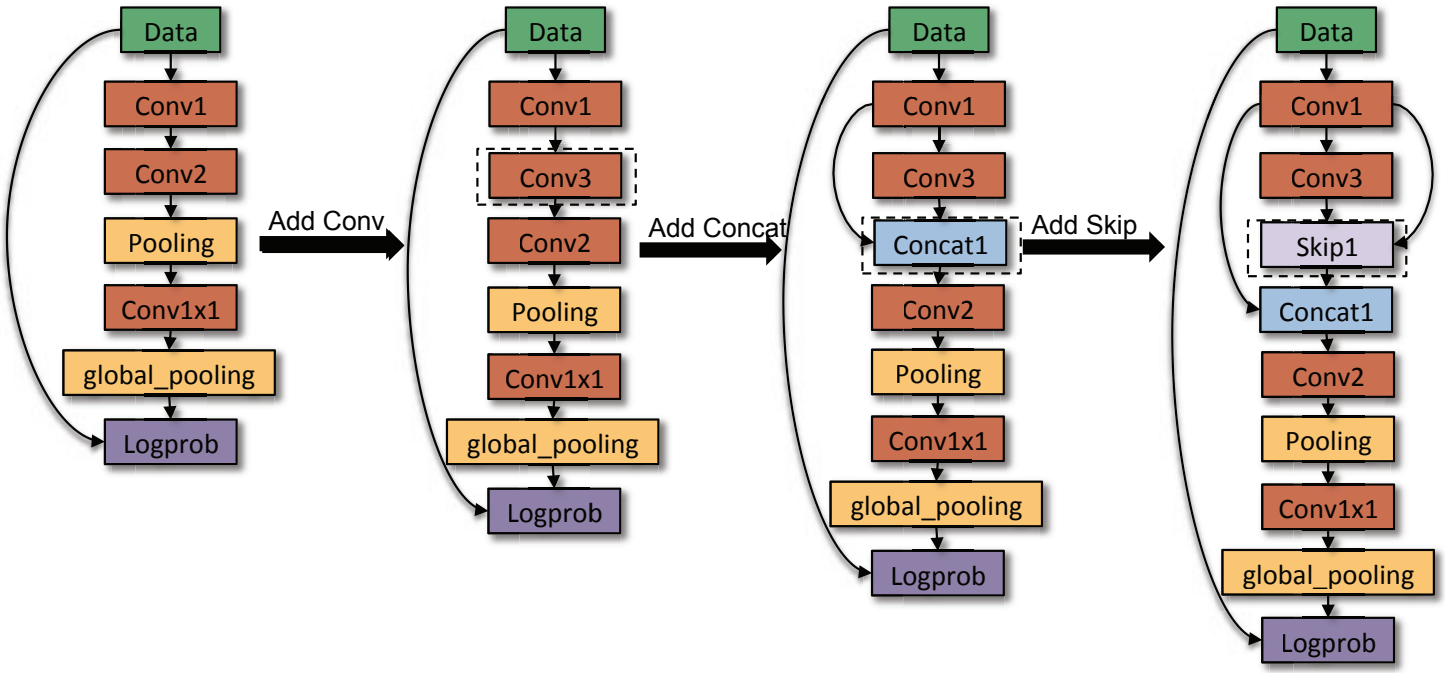
How does genetic approach work?



How does genetic approach work?



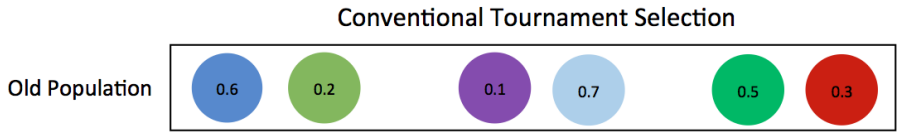
Example of mutation operations



Example `add_convolution`, `add_concatenate` and `add_skip`

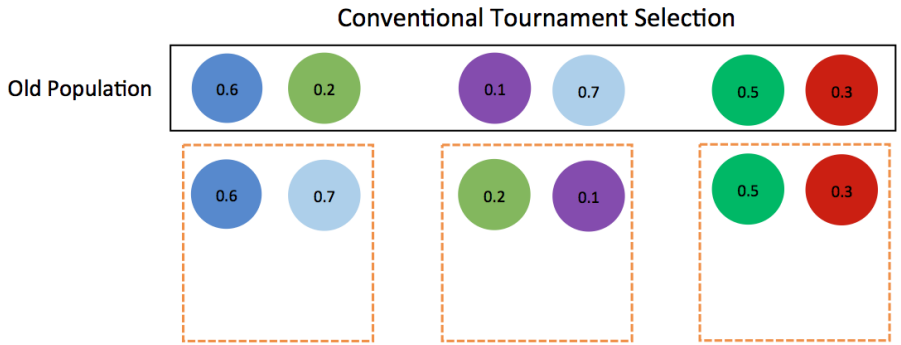
How to reduce computational cost?

- Traditional selection strategies:
  - Samping uniformly
  - Sampling by fitness score
  - Tournament selection



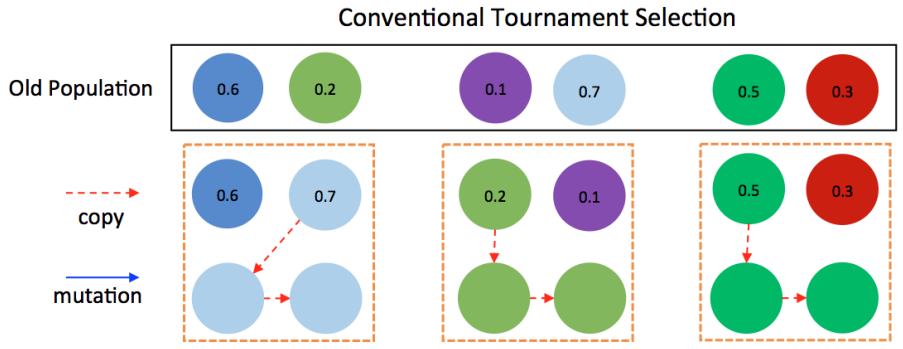
How to reduce computational cost?

- Traditional selection strategies:
  - Samping uniformly
  - Sampling by fitness score
  - Tournament selection



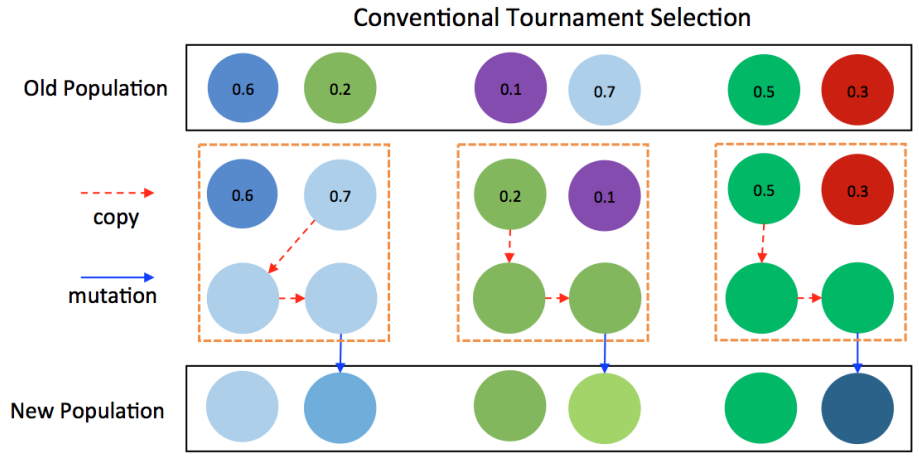
How to reduce computational cost?

- Traditional selection strategies:
  - Samping uniformly
  - Sampling by fitness score
  - Tournament selection



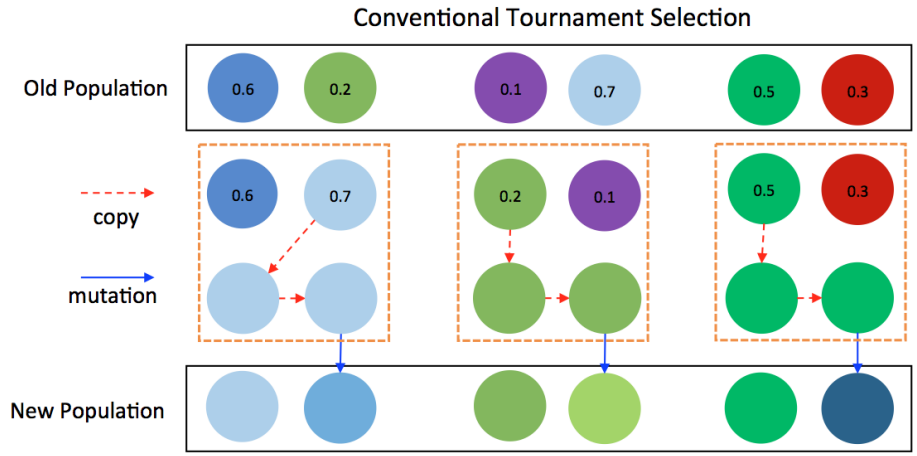
How to reduce computational cost?

- Traditional selection strategies:
  - Samping uniformly
  - Sampling by fitness score
  - Tournament selection



How to reduce computational cost?

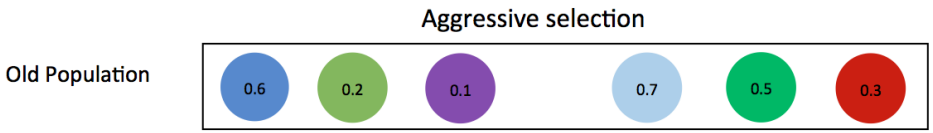
- Traditional selection strategies:
  - Samping uniformly
  - Sampling by fitness score
  - Tournament selection



- Potential Issue: Weak individuals might survive for a long period and it wastes a lot of computation to train those weak individuals that will be eventually killed.

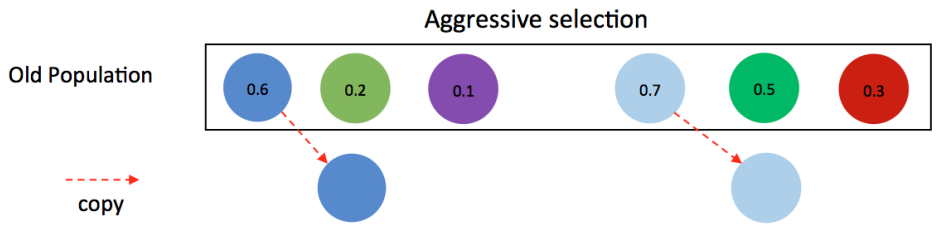
How to reduce computational cost?

- Propose aggressive selection stragegy to select top k survival and eliminate other weak individuals at early stages.



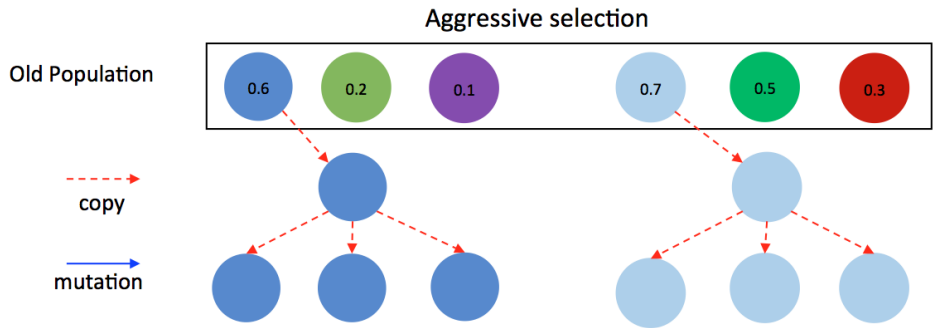
How to reduce computational cost?

- Propose aggressive selection stragegy to select top k survival and eliminate other weak individuals at early stages.



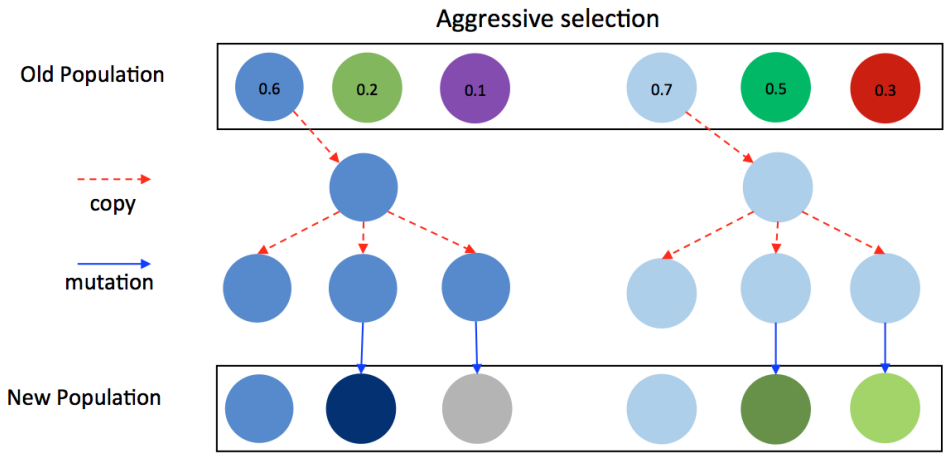
How to reduce computational cost?

- Propose aggressive selection stragegy to select top k survival and eliminate other weak individuals at early stages.



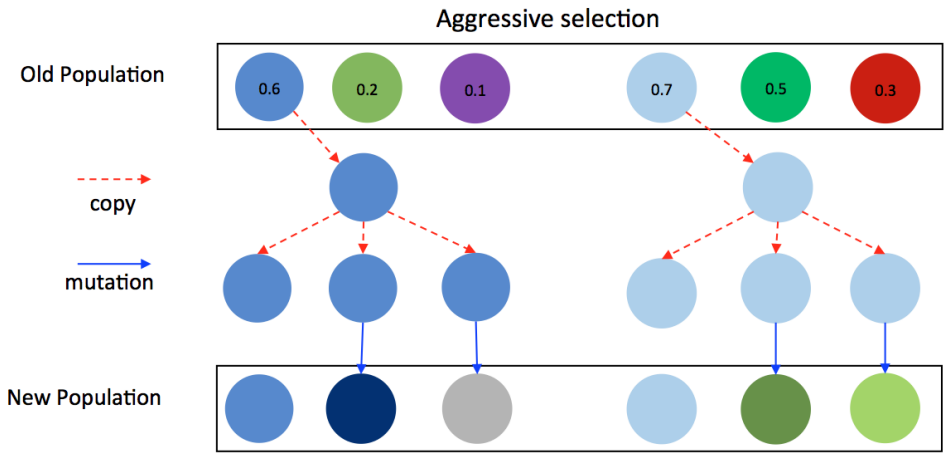
How to reduce computational cost?

- Propose aggressive selection stragegy to select top k survival and eliminate other weak individuals at early stages.



How to reduce computational cost?

- Propose aggressive selection stragegy to select top k survival and eliminate other weak individuals at early stages.



- However, this strategy decreases the diversity of the population.

More mutations for diversity

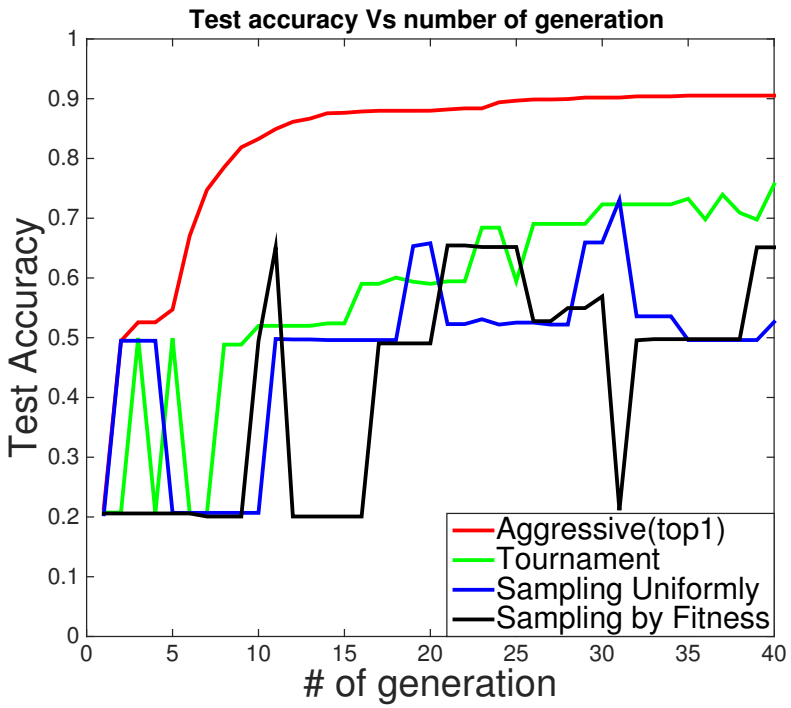
- More mutation operations defined

Mutations	[6]	Ours
add_convolution	✓	✓
remove_convolution	✓	✓
alter_channel_number	✓	✓
alter_filter_size	✓	✓
alter_stride	✓	✓
add_dropout	-	✓
remove_dropout	-	✓
add_pooling	-	✓
remove_pooling	-	✓
add_skip	✓	✓
remove_skip	✓	✓
add_concatenate	-	✓
remove_concatenate	-	✓
add_fully_connected	-	✓
remove_fully_connected	-	✓

The allowed mutation operations in our work and in [1]; ✓represents defined while - represents NA

Experimental results

- Justify that the aggressive selection strategy works.



Experimental results

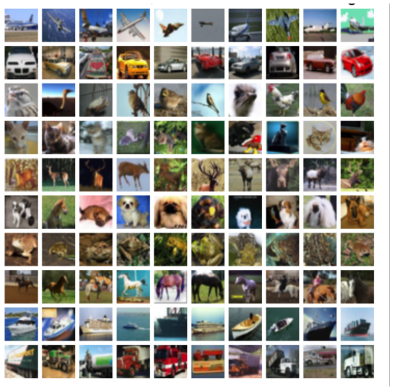
Handwriten digits recognition

Approach	**Test Acc**	**Comp Cost**
SOTA[9]	0.9979	–
Genetic-CNN[10]	0.9966	48 GPUH
EDEN[1]	0.9840	–
Ours	0.9969	35 GPUH

Comparison of test accuracy and computational cost on MNIST dataset.

Experimental results

10 classes image recognition

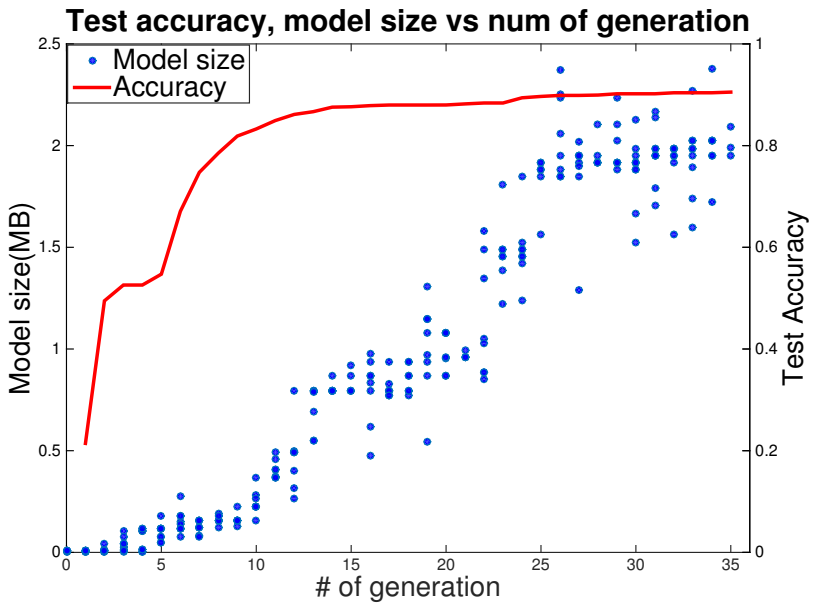


Approach	**Test Acc**	**Comp Cost**
SOTA[3]	0.9654	-
LS-Evolution[6]	0.9460	65,536 GPUH
Genetic-CNN[10]	0.7706	408 GPUH
EDEN[1]	0.7450	–
Ours	0.9052	72 GPUH

Comparison of test accuracy and computational cost on CIFAR-10 dataset.

Experimental results

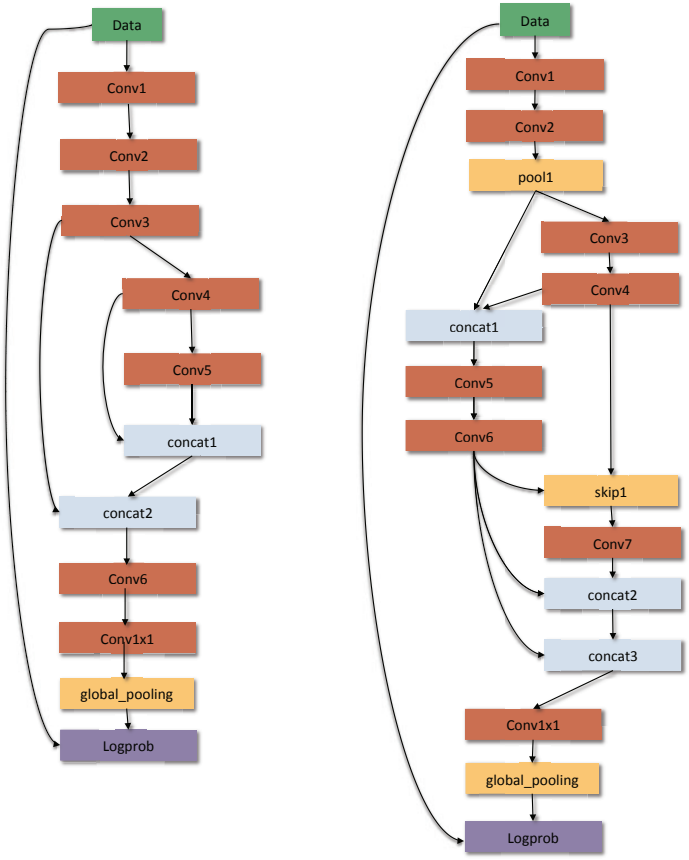
- Show how model size and test accuracy of neural network change along evolution.



The evolution of model size and test accuracy of the best individual in our algorithm on CIFAR-10.

Experimental results

- Show the finally learned neural network structures by our approach.



Discovered neural network structures for CIFAR-10 and CIFAR-100 dataset.

Contribution for less tuning

- Explored genetic approach for searching neural networks.
- Implemented a variety of mutation operations.
- Proposed different strategies to reduce computational cost.
- Achieved competitive performance with dramatically reduced computational cost.

Outline

Motivation

Approach	PARAMS.	CIFAR-10	CIFAR-100	Comp Cost
EDEN [1]	0.2 M	74.5%	-	-
Genetic CNN [10]	-	92.9%	71.0%	408 GPUH
LS-Evolution [6]	5.4 M	94.6%	-	65,536 GPUH
LS-Evolution [6]	40.4 M	-	77.0%	65,536 GPUH
AG-Evolution	-	90.5%	-	72 GPUH
AG-Evolution	-	-	66.9%	136 GPUH

Motivation

Approach	PARAMS.	CIFAR-10	CIFAR-100	Comp Cost
EDEN [1]	0.2 M	74.5%	-	-
Genetic CNN [10]	-	92.9%	71.0%	408 GPUH
LS-Evolution [6]	5.4 M	94.6%	-	65,536 GPUH
LS-Evolution [6]	40.4 M	-	77.0%	65,536 GPUH
AG-Evolution	-	90.5%	-	72 GPUH
AG-Evolution	-	-	66.9%	136 GPUH

Can we do better?

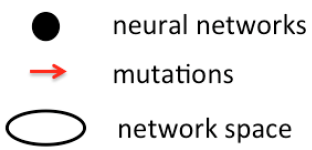
Ecologically-Inspired Approach

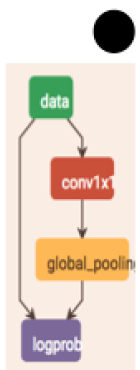
Inspired by ecological system, we propose to utilize the following three concepts:

- Rapid Succession[7]
  - The community is dominated by diversified fast-growing individuals during the primary succession, while in the secondary succession, the community is dominated by more competitive individuals.
- Gene Duplication[6]
  - Obtaining new genes and leading to evolutionary innovation [11].
- Acclerated Extinction

Ecologically-Inspired Approach–Rapid Succession

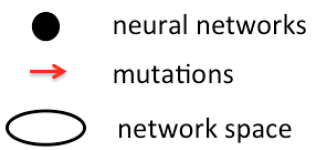
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)

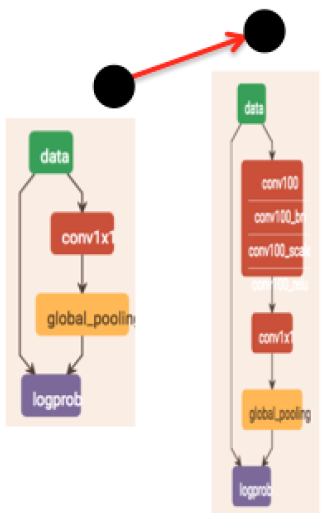




Ecologically-Inspired Approach–Rapid Succession

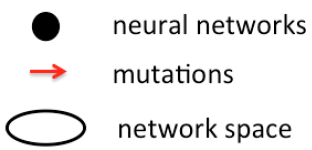
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)

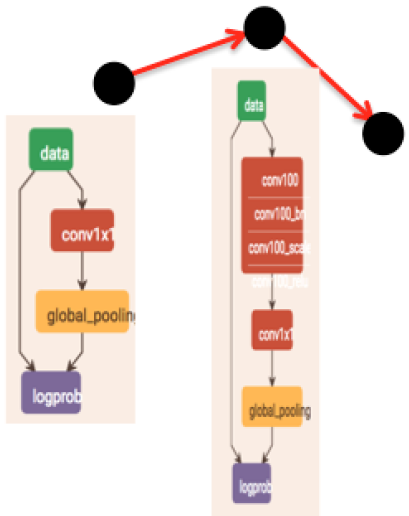




Ecologically-Inspired Approach–Rapid Succession

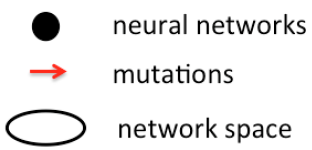
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)

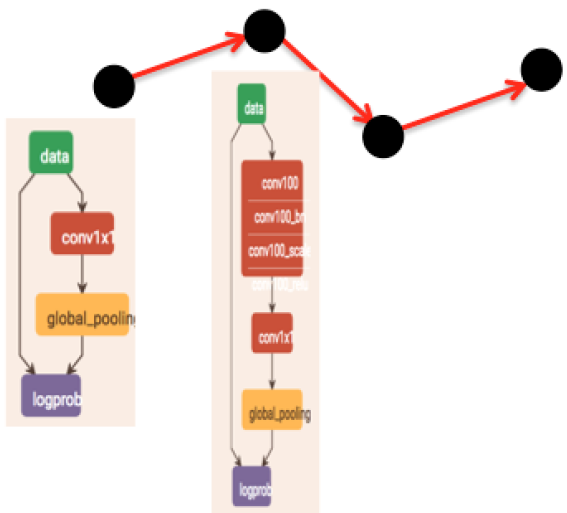




Ecologically-Inspired Approach–Rapid Succession

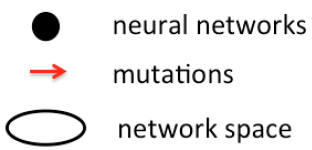
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)

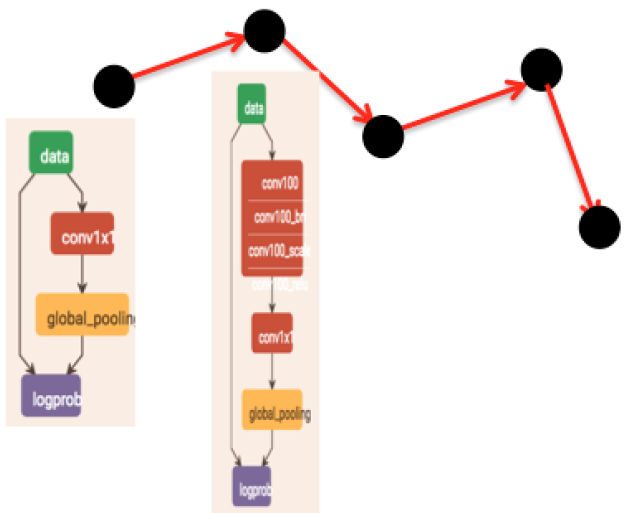




Ecologically-Inspired Approach–Rapid Succession

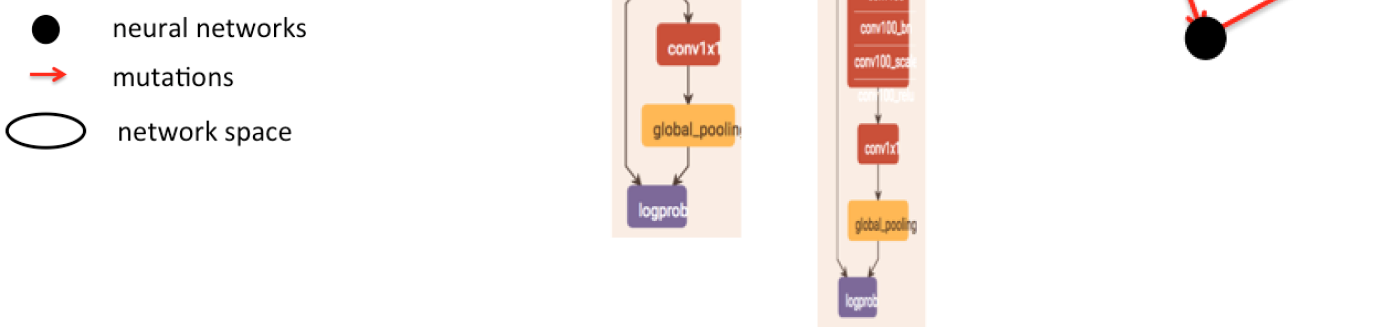
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)





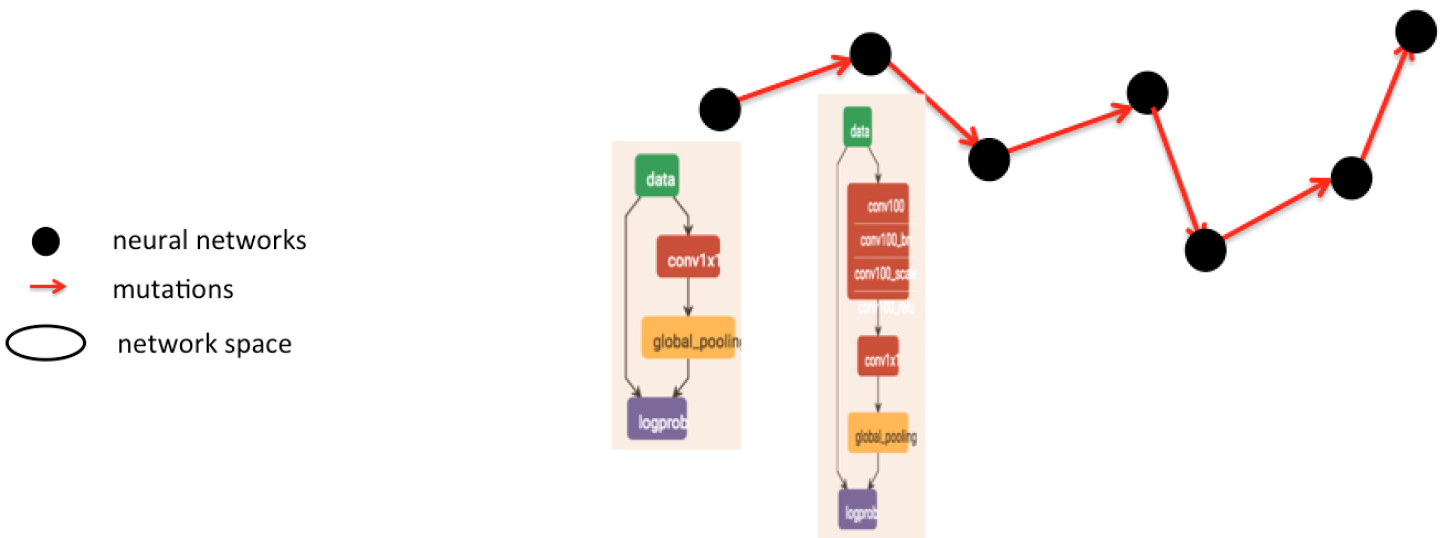
Ecologically-Inspired Approach–Rapid Succession

- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)



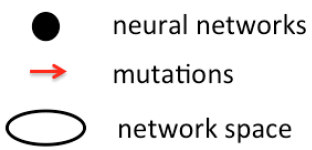
Ecologically-Inspired Approach–Rapid Succession

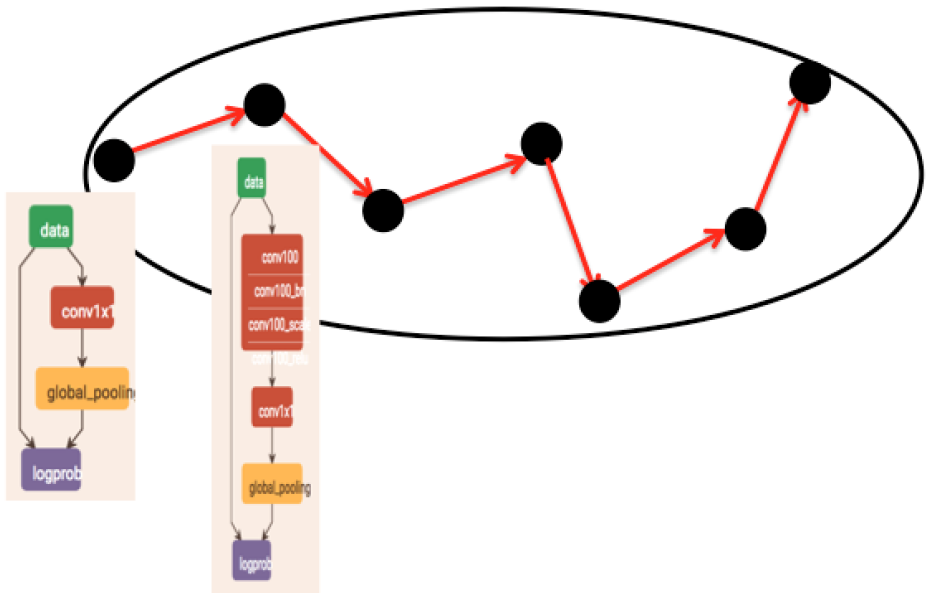
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)



Ecologically-Inspired Approach–Rapid Succession

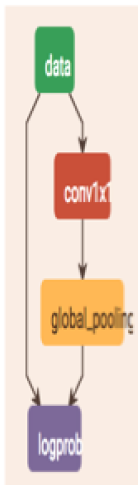
- Rapid Succession[7]
  - Primary Succession: rapidly evolve a community of poor initialized neural network structures into a more diverse community.
  - Secondary Succession: fine-grained searching based on the networks from the primary succession.
- Standard evolution (explored in previous section)





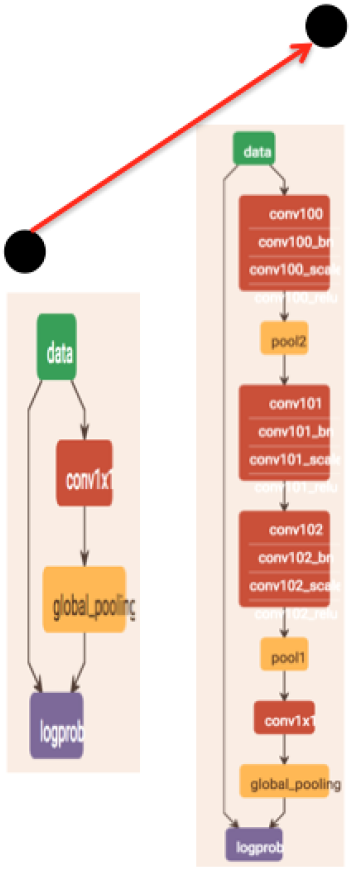
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



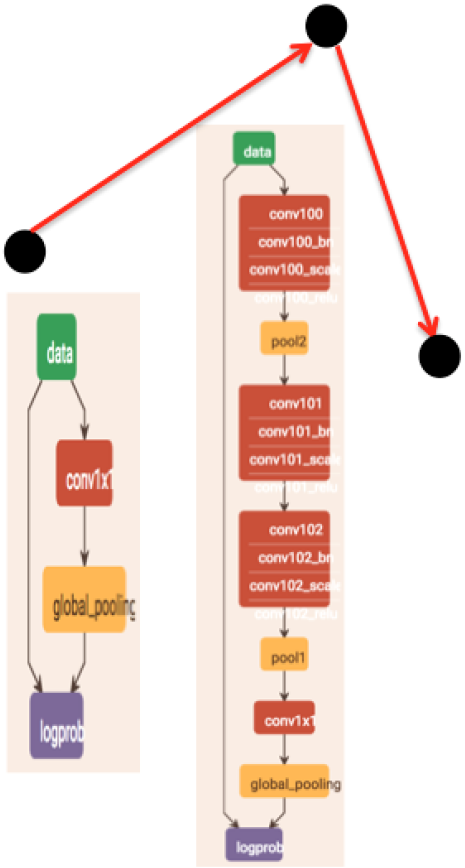
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



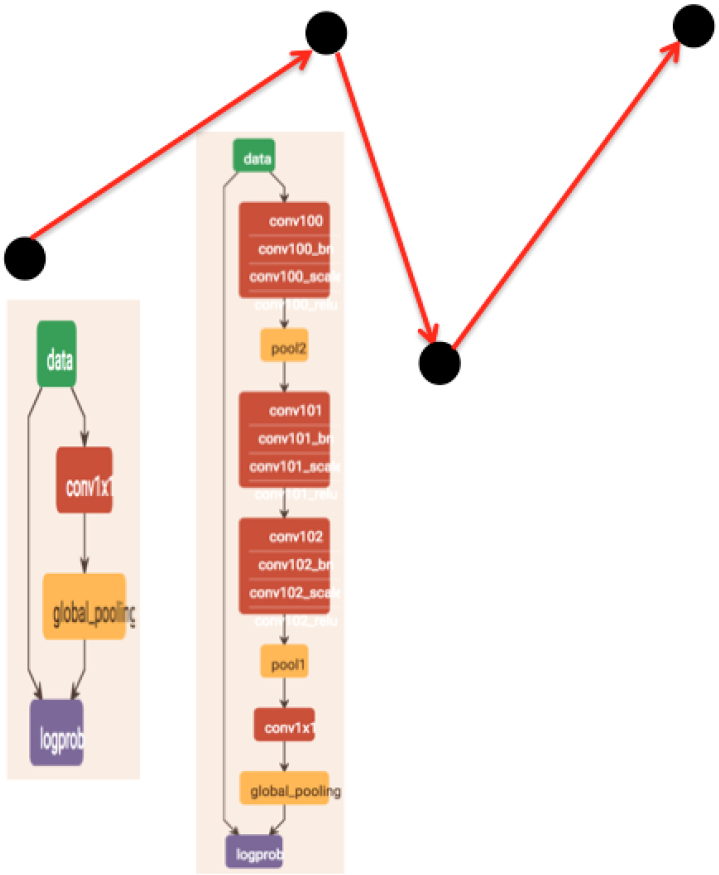
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



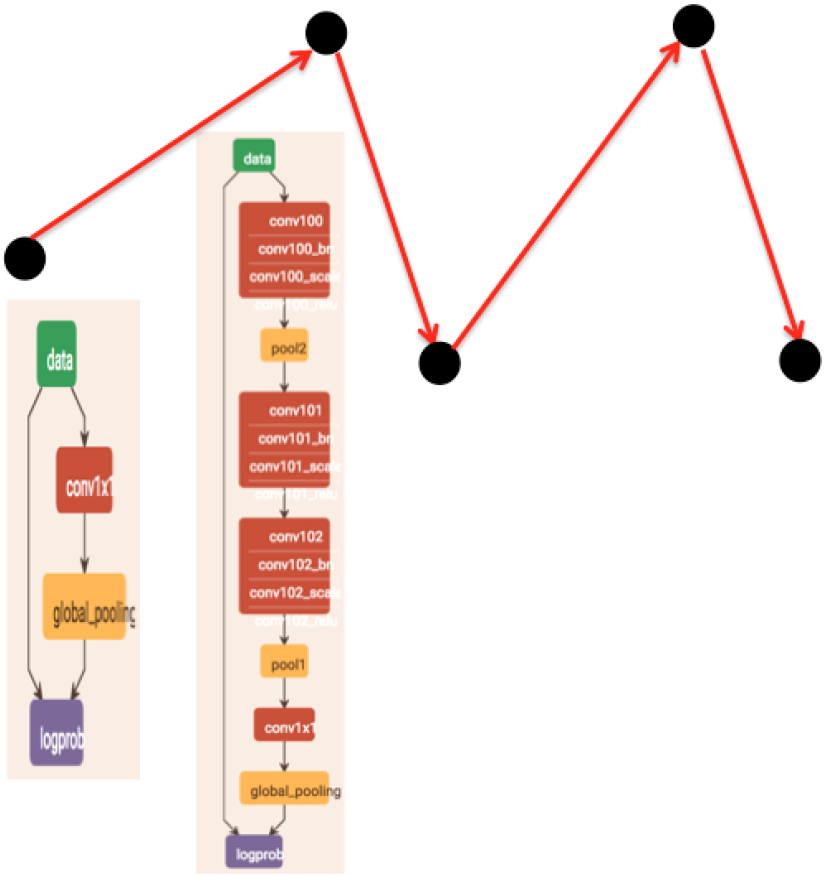
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



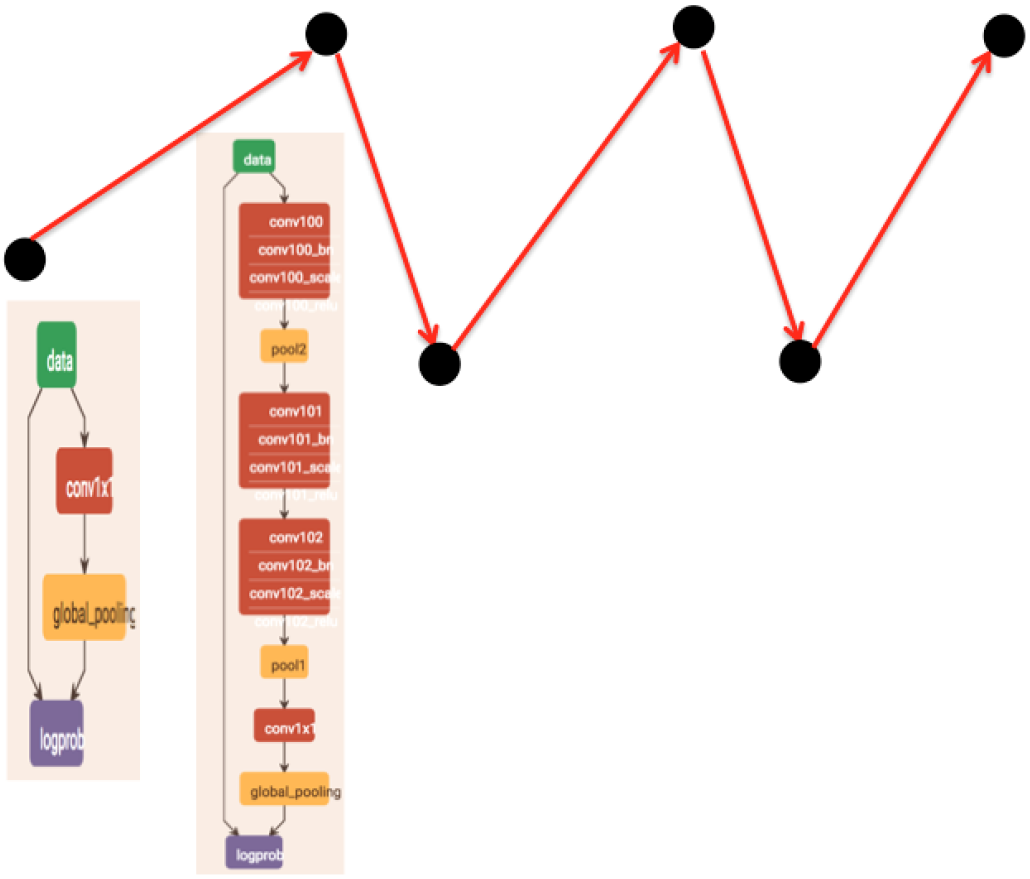
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



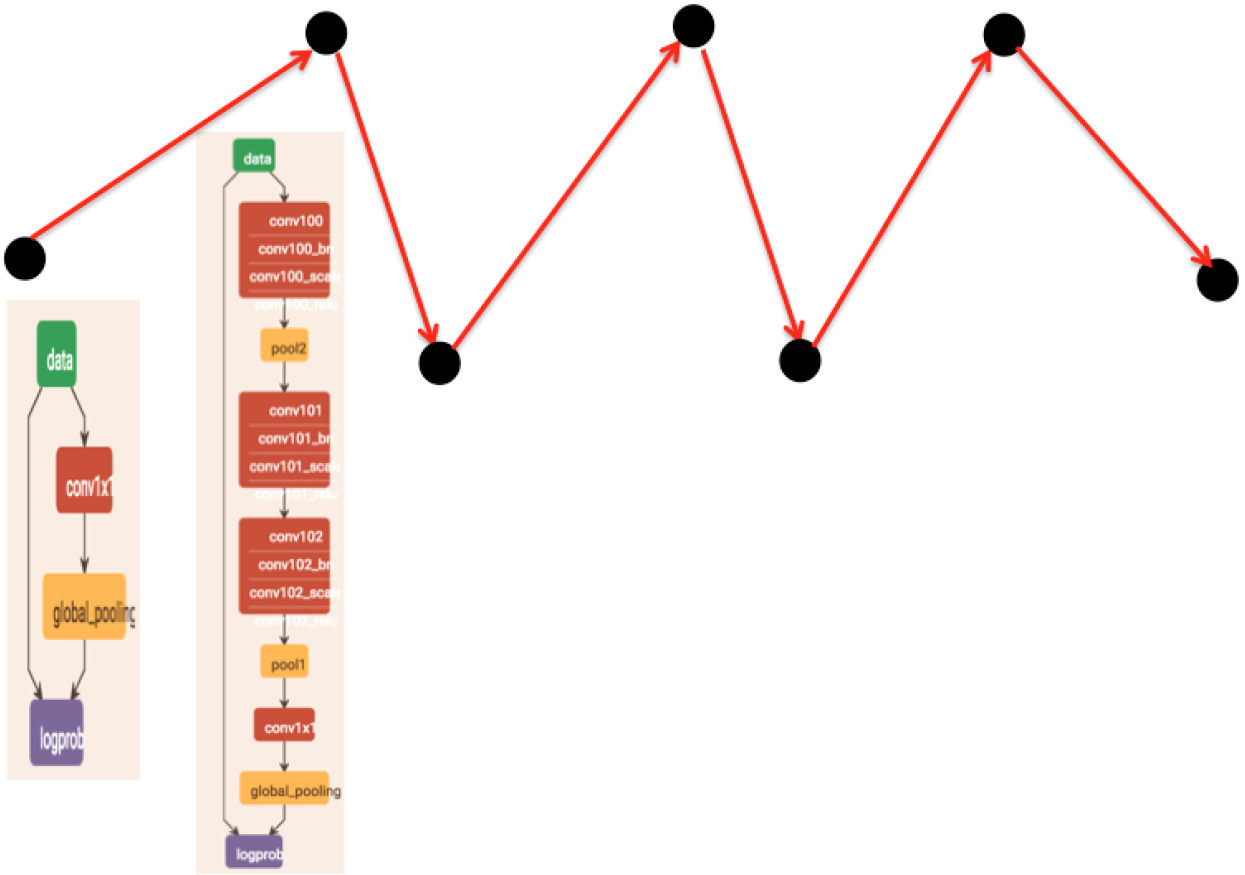
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



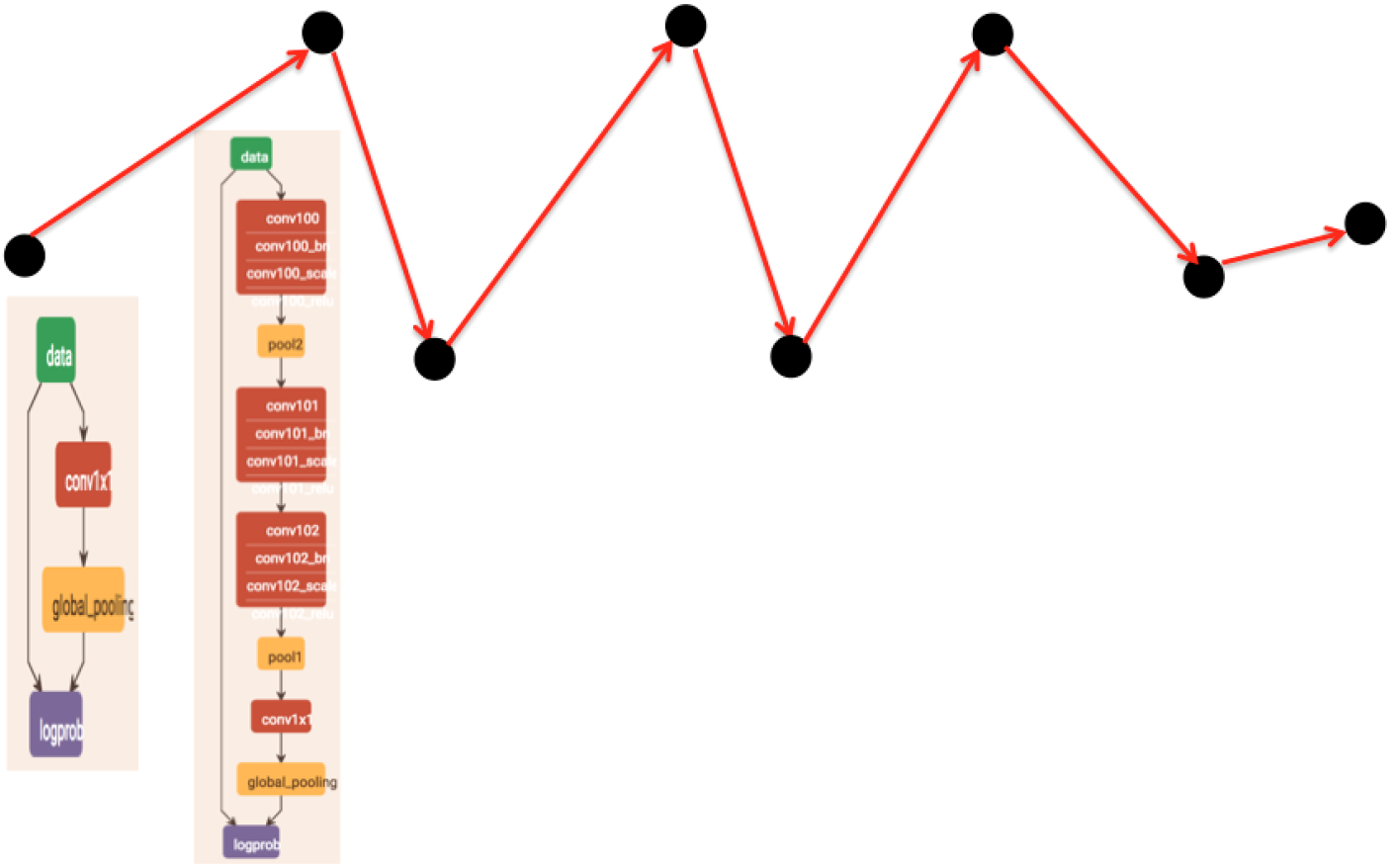
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



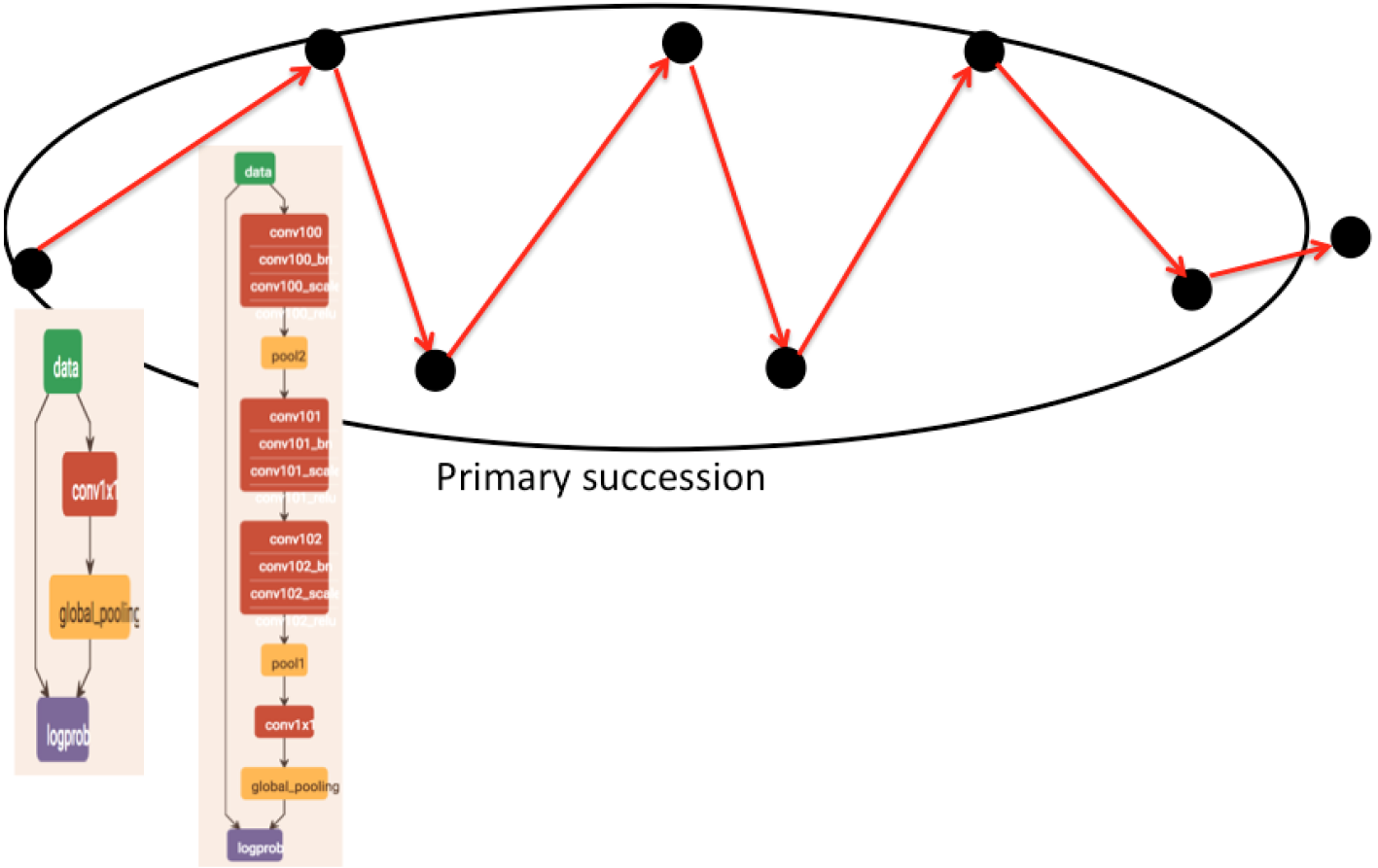
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



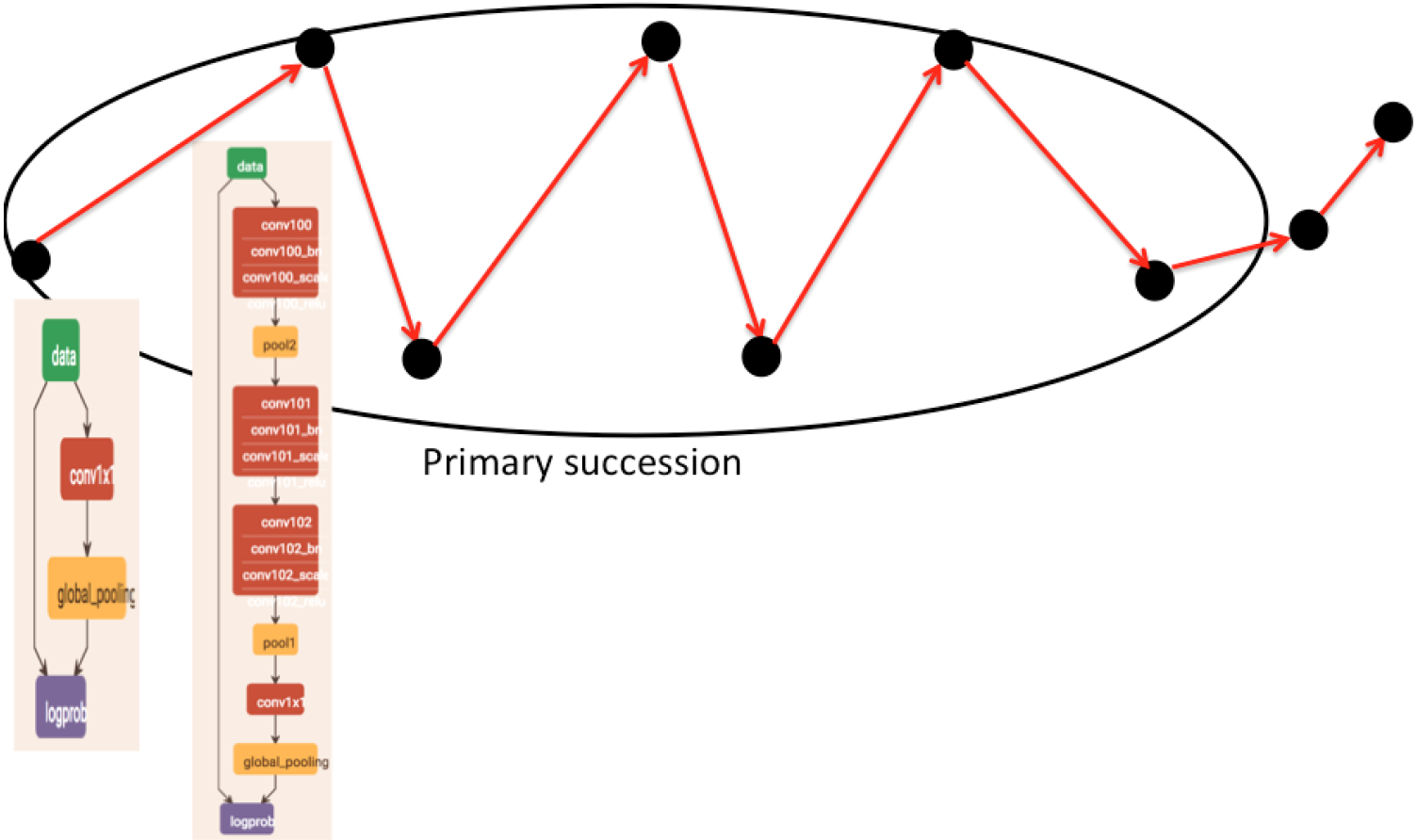
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



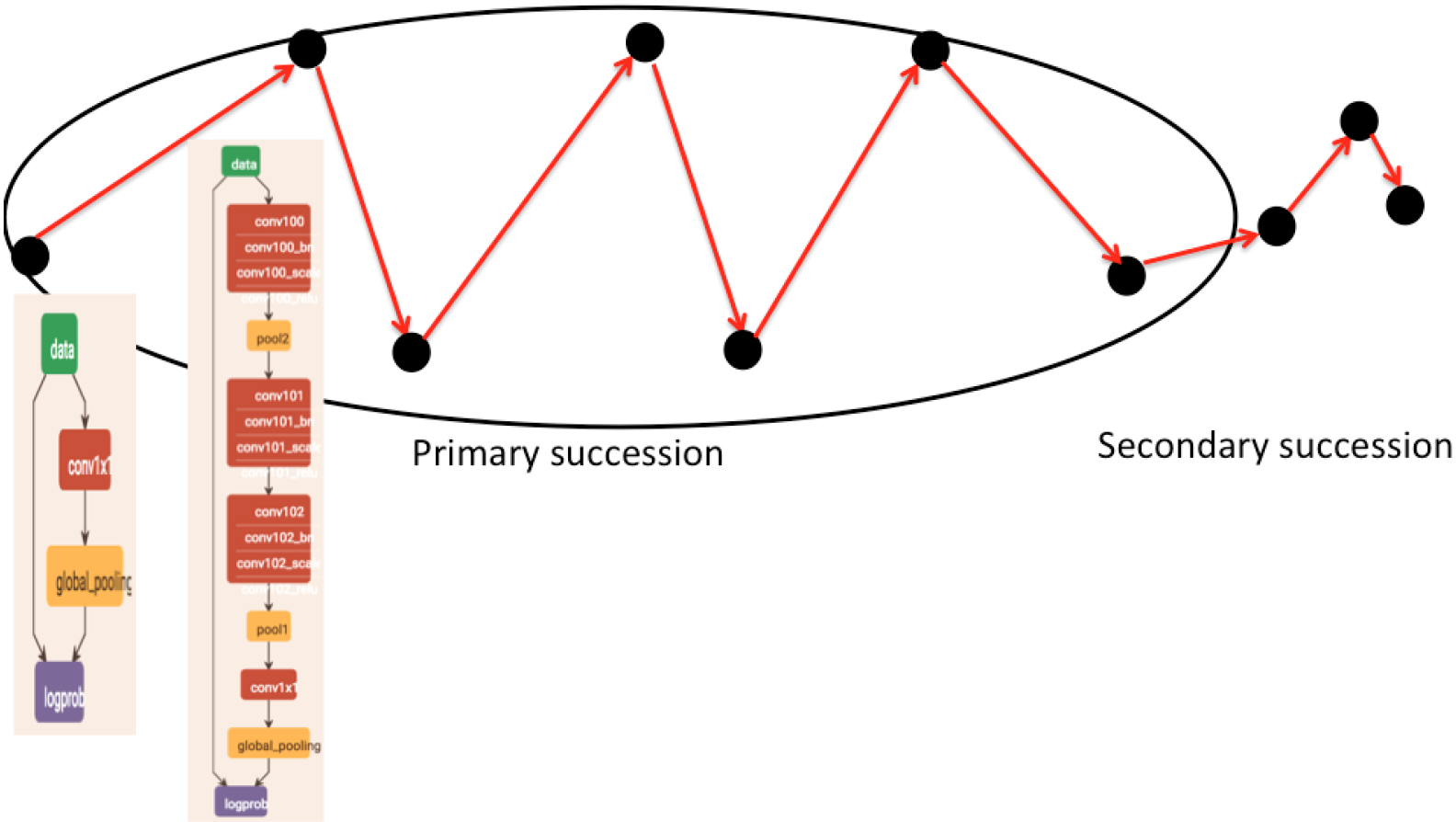
Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



Ecologically-Inspired Approach–Rapid Succession

- Primary and secondary succession



- Primary succession: more diverse and large searching space
- Secondary succession: fine-grained searching

These two types of succession are anology to reducing learning rate in gradient descent algorithm.

Ecologically-Inspired Approach–Accelerated Extinction

- Accelerated Extinction: extinguish the individuals that may possibly fail at early iterations.
  - Choose lankmark points $T1, T2$ during training each network.

Ecologically-Inspired Approach–Accelerated Extinction

- Accelerated Extinction: extinguish the individuals that may possibly fail at early iterations.
  - Choose lankmark points $T1, T2$ during training each network.

  - Update threshold $S1, S2$ in those landmark points $T1, T2$ based on previous generation.

Ecologically-Inspired Approach–Accelerated Extinction

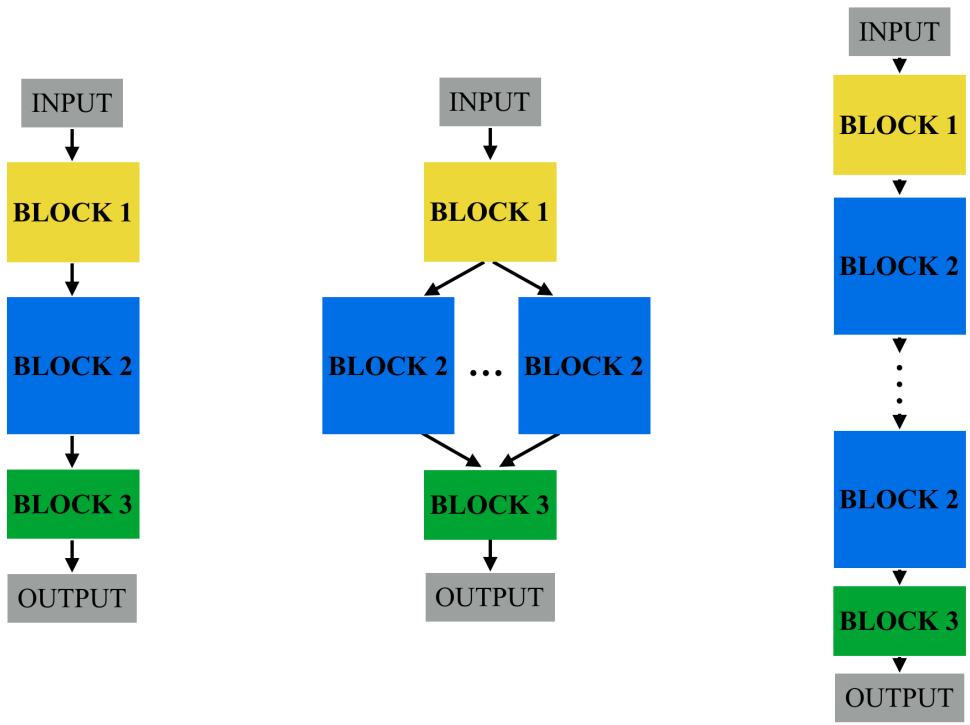
- Accelerated Extinction: extinguish the individuals that may possibly fail at early iterations.
  - Choose lankmark points $T1, T2$ during training each network.

  - Update threshold $S1, S2$ in those landmark points $T1, T2$ based on previous generation.

  - Extinguish individuals that do not exceed those thresholds at landmark points.

Ecologically-Inspired Approach–Gene Duplication

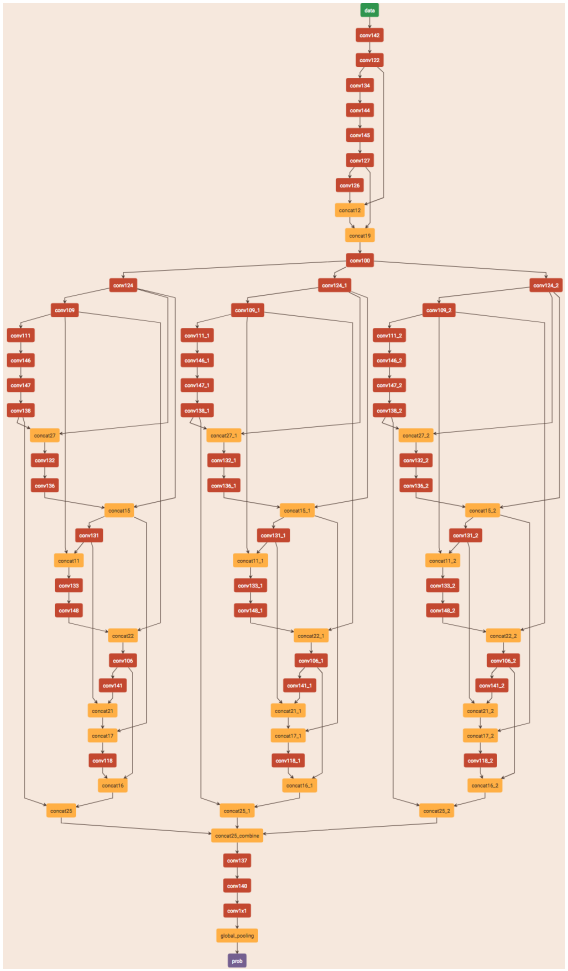
- Gene Duplication[6]: utilize the novel beneficial structures discovered.

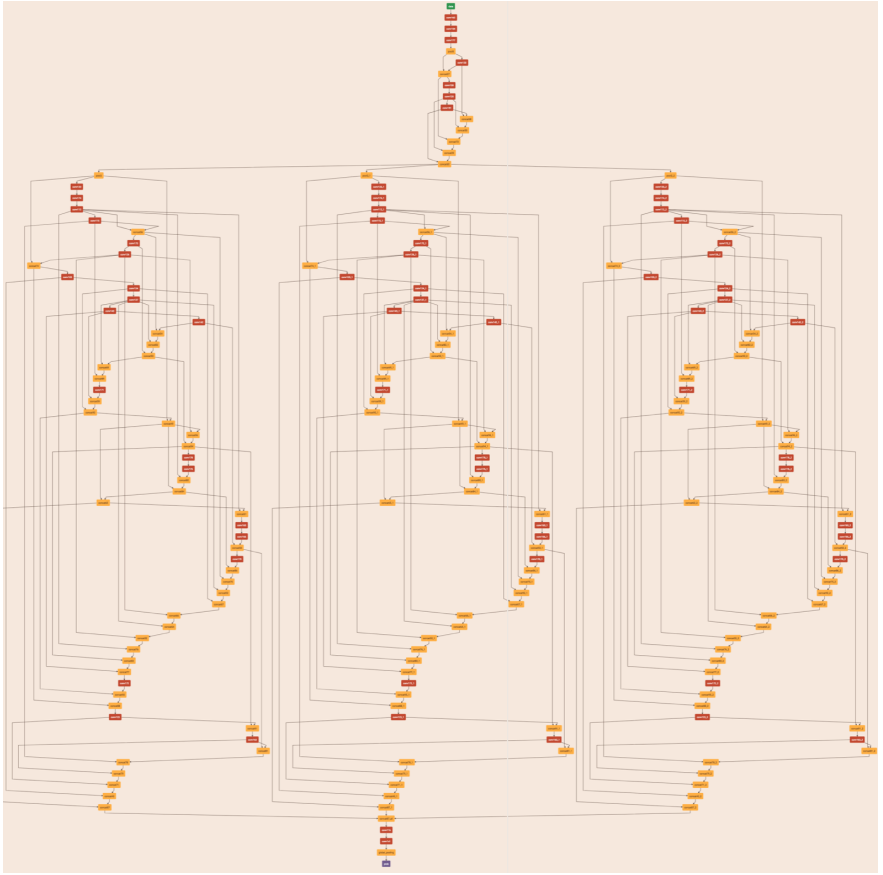


Gene Duplication

The Learned Structures

The finally obtained structures on CIFAR-10 and CIFAR-100





Experimental results

- Analyze Gene Duplication.

Method	CIFAR-10 (PARAMS.)	CIFAR-100 (PARAMS.)
EIGEN	93.7% (1.2 M)	76.9% (6.1 M)
EIGEN-D	94.6% (2.6 M)	78.1% (11.8 M)

Performance of the Gene Duplication on CIFAR-10 and CIFAR-100.

Experimental results

Approach	PARAMS.	CIFAR-10	CIFAR-100	Comp Cost
EDEN [1]	0.2 M	74.5%	-	-
Genetic CNN [10]	-	92.9%	71.0%	408 GPUH
LS-Evolution [6]	5.4 M	94.6%	-	65,536 GPUH
LS-Evolution [6]	40.4 M	-	77.0%	65,536 GPUH
AG-Evolution†	-	90.5%	-	72 GPUH
AG-Evolution†	-	-	66.9%	136 GPUH
EIGEN	1.2 M	**93.7%**	-	48 GPUH
EIGEN-D	2.6 M	**94.6%**	-	48 GPUH
EIGEN	6.1 M	-	**76.9%**	120 GPUH
EIGEN-D	11.6 M	-	**78.1%**	120 GPUH

Performance Comparison. †: approach in the previous section

Outline

Conclusion

- Proposed the improved dropout to speed up training neural networks.
- Proposed pattern binarization technique and designed SEP-Nets to achieve the-state-of-art performance.
- Explored genetic approach to design neural network structure and proposed aggressive-selection strategy to reduce computation cost.
- Proposed Ecologically-Inspired approach based on genetic framework to further improve performance of searching network structures.

Thank You!

References I

[1] E. Dufourq and B. A. Bassett. Eden: Evolutionary deep networks for efficient machine learning. *arXiv preprint arXiv:1709.09161*, 2017.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[3] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[4] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

References II

[5] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[6] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. Le, and A. Kurakin. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*, 2017.

[7] S. Sahney and M. J. Benton. Recovery from the most profound mass extinction of all time. *Proceedings of the Royal Society of London B: Biological Sciences*, 275(1636):759–765, 2008.

[8] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

References III

[9] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013.

[10] L. Xie and A. Yuille. Genetic cnn. arxiv preprint. *arXiv preprint arXiv:1703.01513*, 2017.

[11] J. Zhang. Evolution by gene duplication: an update. *Trends in ecology & evolution*, 18(6):292–298, 2003.