Firefly Neural Architecture Descent




*Lemeng Wu[1], ***Bo Liu[1]**, Peter Stone[1,2], Qiang Liu[1] ,

[1] The University of Texas at Austin, [2] Sony AI

2020 Conference on Neural Information Processing Systems (NeurIPS)

Motivation

Biological brains can grow new neurons (neurogenesis). Artificial neural networks are fixed in size.

The **benefits** of growing a dynamic architecture:

1. Learning capacity is enlarged on demand (adaptive, energy efficient).
2. Dynamic architecture has been shown effective to mitigate *catastrophic forgetting* in continual learning (Rusu et al., 2016, Yoon et al., 2017, Li et al., 2019).

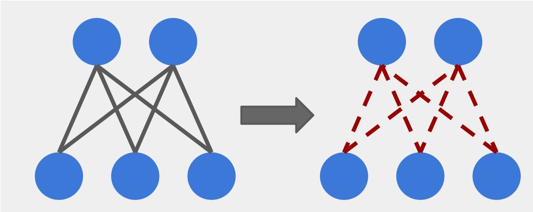
Motivation

Limitations of existing growing methods:

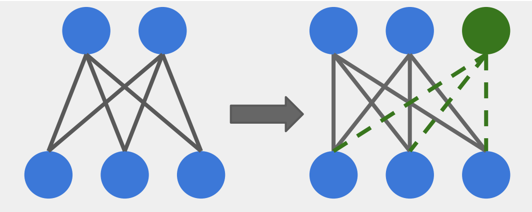
1. Previous growing methods are often based on heuristics.
2. An exception is *splitting steepest descent* (Liu et al., 2019) that progressively splits neurons greedily. But the method is *limited to* splitting (does not consider new neurons/layers) and has *high time complexity* (requires solving an eigen-problem per growth).

Joint Parametric & Architecture Descent

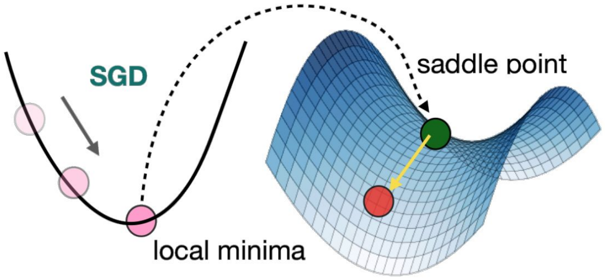
A neural network consists of both its **parameters** and its **architecture**. In this work, we propose to jointly optimize both.



Parametric Descent



Architecture Descent



(SGD refers to Stochastic Gradient Descent; Image from Wang et al., 2019)

When a network grows, the previous local minima can become a saddle point in the larger space.

A General Framework for Network Optimization

Assume the current neural network is f_t. Then we looks for

$$f_{t+1} = \arg\min_{f} \left\{ L(f) \quad s.t. \quad f \in \mathcal{B}(f_t,\ \epsilon), \qquad C(f) \leq C(f_t) + \eta_t \right\}$$

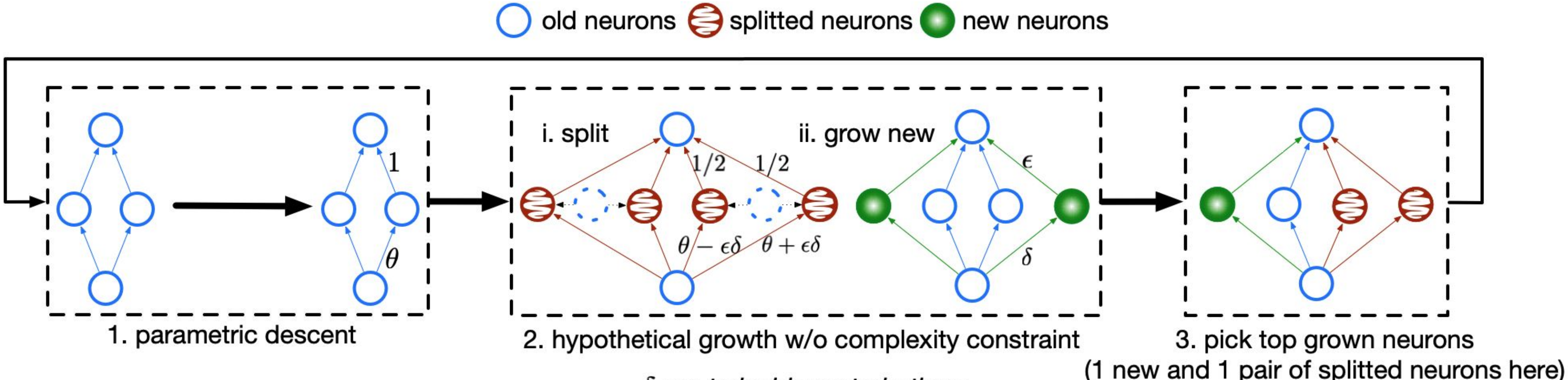
- $L(\cdot)$ denotes the loss function;
- $\mathcal{B}(f_t,\ \epsilon)$ represents a ball of radius ϵ centered at f.
- $C(\cdot)$ measures the complexity of the network, i.e. the FLOPs.

Firefly Neural Architecture Descent

We introduce *firefly neural architecture descent* to solve

$$f_{t+1} = \arg\min_f \left\{ L(f) \quad s.t. \quad f \in \mathcal{B}(f_t,\ \epsilon), \qquad C(f) \le C(f_t) + \eta_t \right\}$$

Specifically, we propose parametric descent + 2-step growing:



δ are trainable perturbations

Firefly Neural Architecture Descent

In practice, to solve

$$f_{t+1} = \arg\min_{f} \left\{ L(f) \quad s.t. \quad f \in \mathcal{B}(f_t,\, \epsilon), \qquad C(f) \leq C(f_t) + \eta_t \right\}$$

Between epochs of parametric descent, we first grow the network without the complexity constraint, then greedily pick the top grown neurons that contribute the most to loss decrease.

Algorithm 1 Firefly Neural Architecture Descent

Input: Loss function $L(f)$; initial small network f_0; search neighborhood $\mathcal{B}(f, \epsilon)$; maximum increase of size $\{\eta_t\}$.

Repeat: At the t-th growing phase:

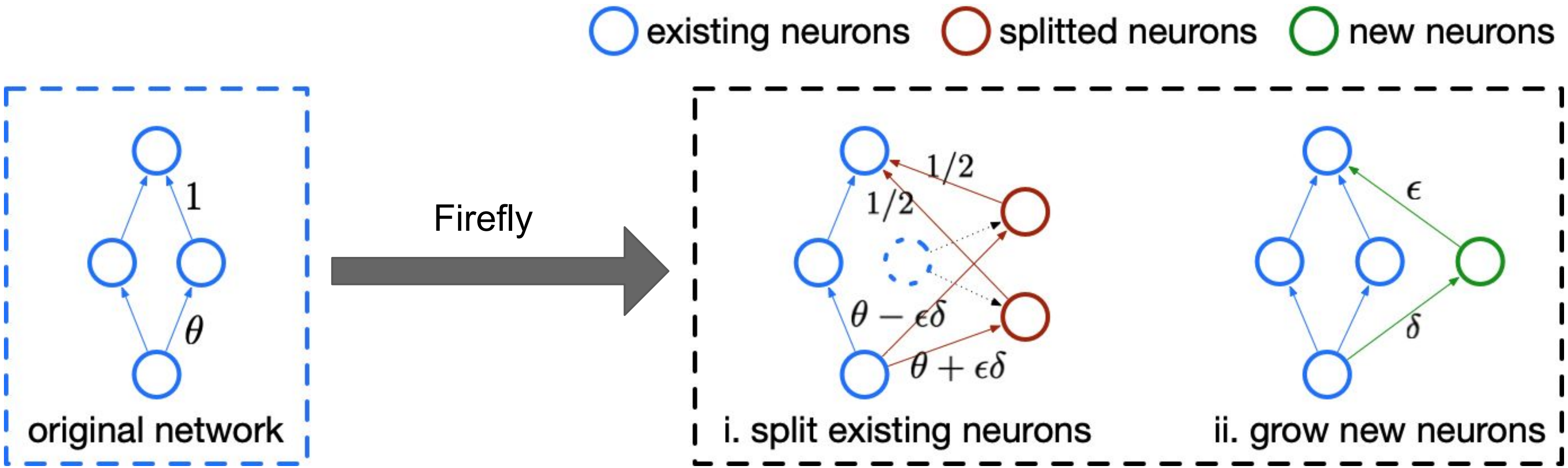
1. Optimize the parameter of f_t with fixed structure using a typical optimizer for several epochs. — **parametric descent**

2. Minimize $L(f)$ in $f \in \mathcal{B}(f, \epsilon)$ without the complexity constraint (see e.g., (4)) to get a large "over-grown" network $\tilde{f}_{t+1}$ by performing gradient descent.

3. Select the top η_t neurons in $\tilde{f}_{t+1}$ with the highest importance measures to get f_{t+1} (see (5)).

} **architecture descent** (steps 2–3)

An Example (Growing Wider)



A single hidden layer network:

split existing neurons **grow new**

$$f_t(x) = \sum_{i=1}^{m} \sigma(x, \theta_i) \qquad f_{\varepsilon,\delta}(x) = \sum_{i=1}^{m} \boxed{\frac{1}{2}\Big(\sigma(x, \theta_i + \varepsilon_i\delta_i) + \sigma(x, \theta_i - \varepsilon_i\delta_i)\Big)} + \sum_{i=m+1}^{m+m'} \boxed{\varepsilon_i \sigma(x, \delta_i)}$$

An Example (Growing Wider)

Under this setting, the optimization can be formulated as:

split existing neurons **grow new**

$$f_t(x) = \sum_{i=1}^{m} \sigma(x, \theta_i) \longrightarrow f_{\varepsilon,\delta}(x) = \sum_{i=1}^{m} \boxed{\frac{1}{2}\Big(\sigma(x, \theta_i + \varepsilon_i \delta_i) + \sigma(x, \theta_i - \varepsilon_i \delta_i)\Big)} + \sum_{i=m+1}^{m+m'} \boxed{\varepsilon_i \sigma(x, \delta_i)}$$

$$\min_{\varepsilon,\delta} \Big\{ L(f_{\varepsilon,\delta}) \quad s.t. \quad \|\varepsilon\|_0 \leq \eta_t, \quad \|\varepsilon\|_\infty \leq \epsilon, \quad \|\delta\|_{2,\infty} \leq 1 \Big\}$$

To solve above, we adopt a two-step optimization scheme:

Step One. Optimizing δ and ε without the sparsity constraint $\|\varepsilon\|_0 \leq \eta_t$, that is, **(grow without constraint)**

$$[\tilde{\varepsilon}, \tilde{\delta}] = \arg\min_{\varepsilon,\delta} \Big\{ L(f_{\varepsilon,\delta}) \quad s.t. \quad \|\varepsilon\|_\infty \leq \epsilon, \quad \|\delta\|_{2,\infty} \leq 1 \Big\}.$$

Step Two. Re-optimizing ε with Taylor approximation on the loss. To do so, note that when ϵ is small, we have by Taylor expansion: **(select neurons that contribute most to loss decrease)**

$$L(f_{\varepsilon,\tilde{\delta}}) = L(f) + \sum_{i=1}^{m+m'} \varepsilon_i s_i + O(\epsilon^2), \qquad s_i = \frac{1}{\tilde{\varepsilon}_i} \int_0^{\tilde{\varepsilon}_i} \nabla_{\zeta_i} L(f_{[\tilde{\varepsilon}_{\neg i}, \zeta_i], \tilde{\delta}}) d\zeta_i,$$

Application (Continual Learning)

Problem: Continual learning (CL) studies online multitask learning. But the agent loses its access to prior data when learning on new tasks.

One typical approach in CL is dynamic architecture, where we freeze the model parameters learned from prior tasks, and use the fixed parameters and newly grown neurons to learn a new task.

In the past, such methods often randomly grow new neurons per layer.

Application (Continual Learning)

Here, the network is a union of subnetworks selected by masks:

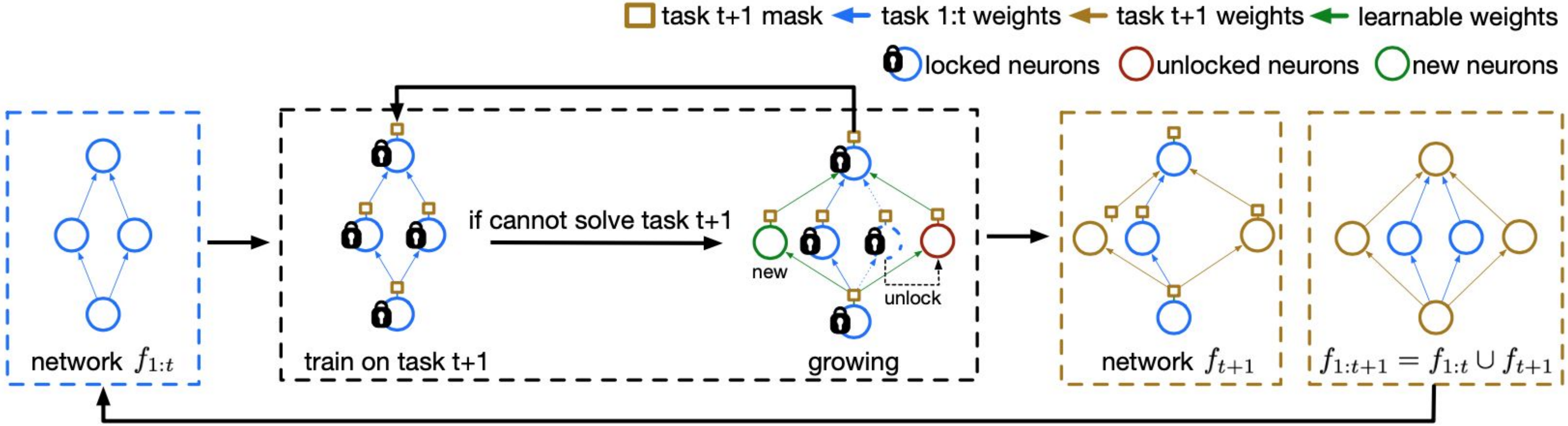


Figure 2: Illustration of how Firefly grows networks in continual learning.

Experiments (Neural Architecture Search)

We compare against some previous growing methods.

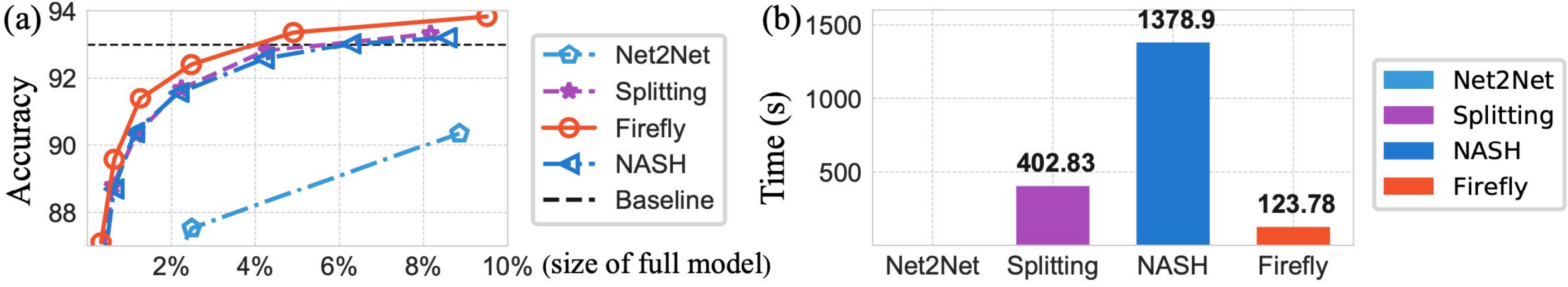


Figure 4: (a) Results of growing increasingly wider networks on CIFAR-10; VGG-19 is used as the backbone. (b) Computation time spent on growing for different methods.

Experiments (Continual Learning)

We apply Firefly to continual image classification task on the CIFAR dataset. Firefly outperforms state-of-the-art dynamic architecture approaches.

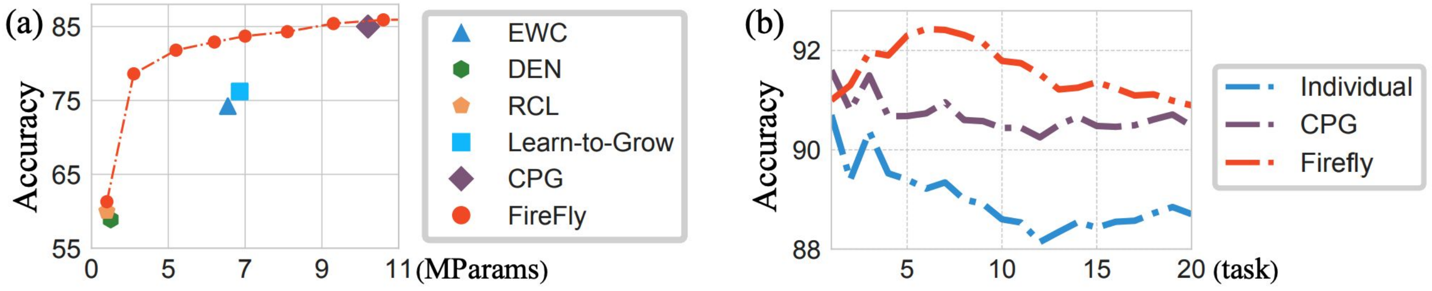


Figure 5: (a) Average accuracy on 10-way split of CIFAR-100 under different model size. We compare against Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), Dynamic Expandable Network (DEN) (Yoon et al., 2017), Reinforced Continual Learning (RCL) (Xu & Zhu, 2018) and Compact-Pick-Grow (CPG) (Hung et al., 2019a). (b) Average accuracy on 20-way split of CIFAR-100 dataset over 3 runs. Individual means train each task from scratch using the Full VGG-16.

References

[1] Liu, Qiang, Wu, Lemeng, and Wang, Dilin. Splitting steepest descent for growing neural architectures. Neural Information Processing Systems (NeurIPS), 2019.

[2] Wang, Dilin, Li, Meng, Wu, Lemeng, Chandra, Vikas, and Liu, Qiang. Energy-aware neural architecture optimization with fast splitting steepest descent. arXiv preprint arXiv:1910.03103, 2019.

[3] Rusu, Andrei A, Rabinowitz, Neil C, Desjardins, Guillaume, Soyer, Hubert, Kirkpatrick, James, Kavukcuoglu, Koray, Pascanu, Razvan, and Hadsell, Raia. Progressive neural networks. arXiv preprint arXiv:1606.04671, 2016.

[4] Yoon, Jaehong, Yang, Eunho, Lee, Jeongtae, and Hwang, Sung Ju. Lifelong learning with dynamically expandable networks. International Conference on Learning Representation (ICLR), 2018.

[5] Li, Xilai, Zhou, Yingbo, Wu, Tianfu, Socher, Richard, and Xiong, Caiming. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. International Conference on Machine Learning (ICML), 2019.

[6] Hung, Ching-Yi, Tu, Cheng-Hao, Wu, Cheng-En, Chen, Chien-Hung, Chan, Yi-Ming, and Chen, Chu-Song. Compacting, picking and growing for unforgetting continual learning. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

[7] Xu, Ju and Zhu, Zhanxing. Reinforced continual learning. In Advances in Neural Information Processing Systems (NeurIPS), 2018.

[8] Kirkpatrick, James, Pascanu, Razvan, Rabinowitz, Neil, Veness, Joel, Desjardins, Guillaume, Rusu, Andrei A, Milan, Kieran, Quan, John, Ramalho, Tiago, Grabska-Barwinska, Agnieszka, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017.