Firefly Neural Architecture Descent: a General Approach for Growing Neural Networks

Model selection problem

$$\arg\min_{f}\{L(f) \quad \text{s.t.} \quad f \in \Omega, \quad C(f) \leq \eta\}$$

Ω – family of models

$C(f)$ – complexity of model

Model selection problem

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

$$\partial\left(f_t, \epsilon\right) - \text{neighbourhood of } f_t$$

Algorithm 1 Firefly Neural Architecture Descent

Input: Loss function $L(f)$; initial small network f_0; search neighborhood $\partial(f, \epsilon)$; 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.

2. Minimize $L(f)$ in $f \in \partial(f, \epsilon)$ 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)).

Neighbourhood

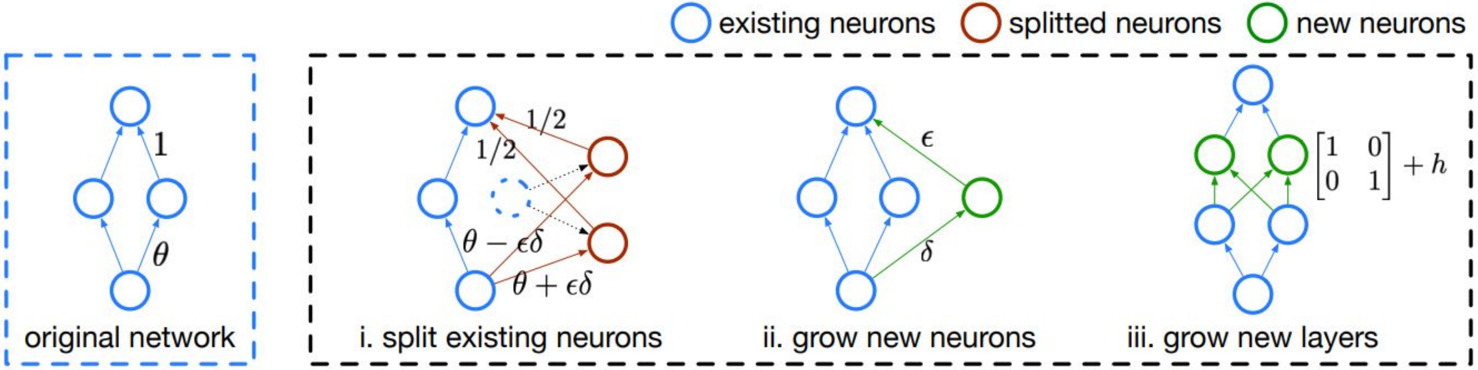
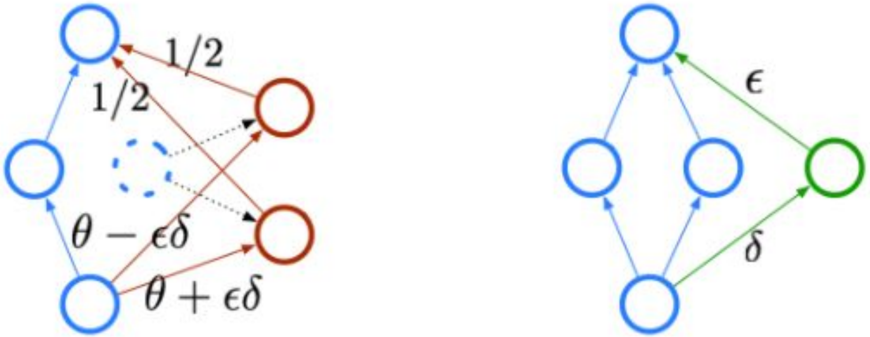


Figure 1: An illustration of three different growing methods within firefly neural architecture descent. Both δ and h are trainable perturbations.

Splitting and Growing Neuron



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

Optimization

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

Step 1

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

Optimization

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

Step 2

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

$$\hat{\varepsilon} = \arg\min_{\varepsilon} \left\{ \sum_{i=1}^{m+m'} \varepsilon_i s_i \quad \text{s.t.} \quad \|\varepsilon\|_0 \leq \eta_t, \quad \|\varepsilon\|_\infty \leq \epsilon \right\}$$

Growing new layer

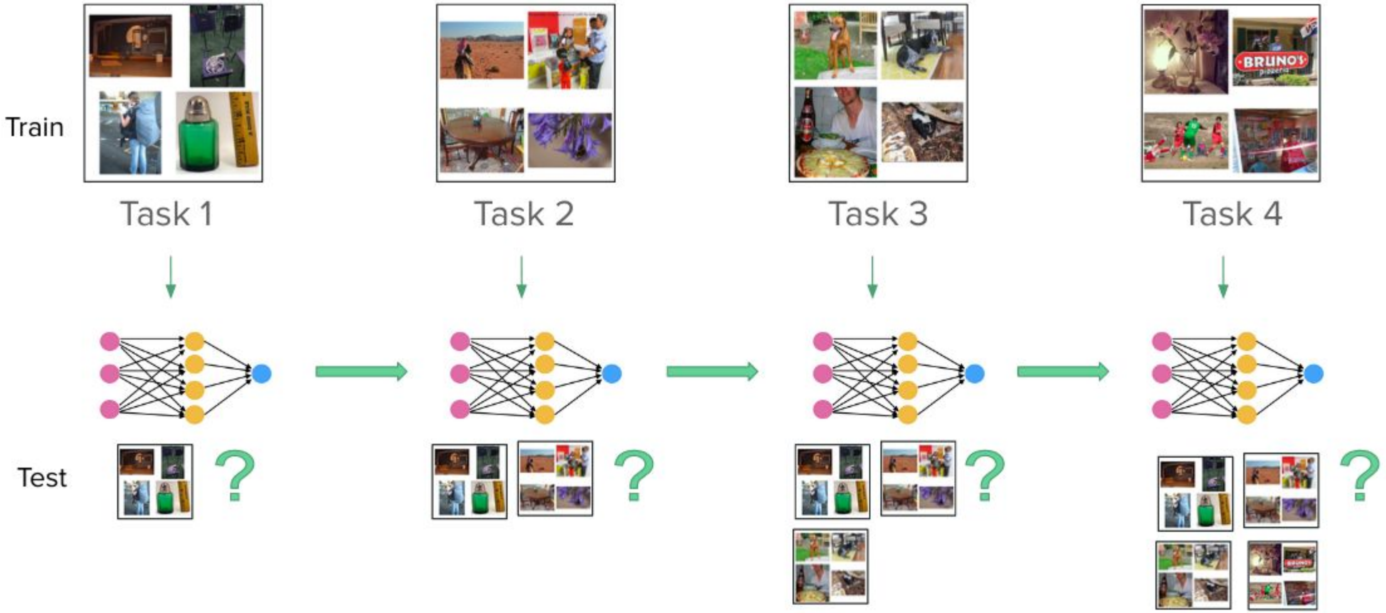
New layer representation

$$f_{\varepsilon,\delta} = g_d \circ (I + h_{d-1}) \cdots (I + h_2) \circ g_2 \circ (I + h_1) \circ g_1, \qquad \text{with} \qquad h_\ell(\cdot) = \sum^{m'} \varepsilon_{\ell i} \sigma\left(\cdot, \delta_{\ell i}\right)$$

Optimization problem to solve

$$\min_{\varepsilon, \boldsymbol{\delta}} \left\{ L\left(f_{\varepsilon,\delta}\right) \text{ s.t. } \|\varepsilon\|_0 \leq \eta_{t,0}, \quad \underline{\|\varepsilon\|_{\infty,0} \leq \eta_{t,1}}, \quad \|\varepsilon\|_\infty \leq \epsilon, \quad \|\boldsymbol{\delta}\|_{2,\infty} \leq 1 \right\},$$

Continual learning



Continual learning

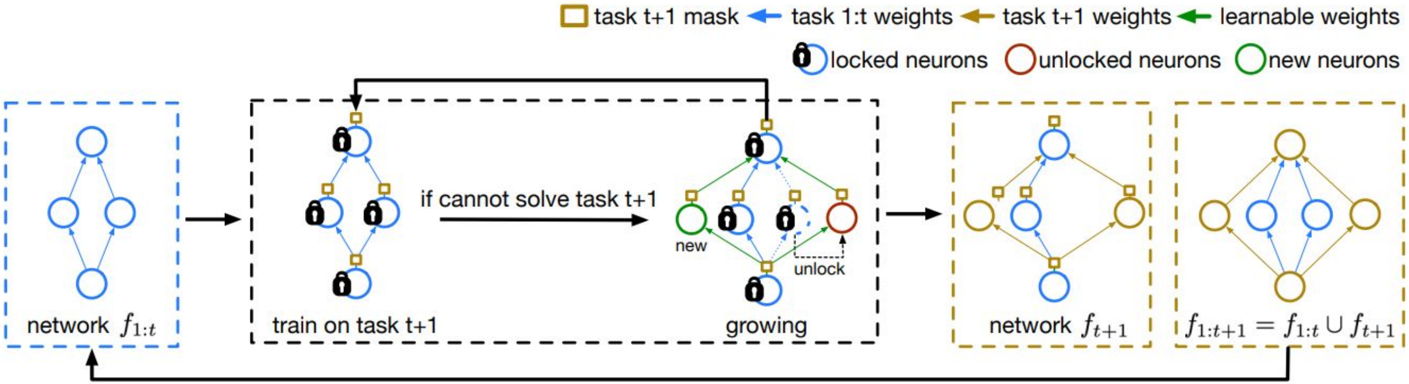


Figure 2: Illustration of how Firefly grows networks in continual learning.

Firefly vs Random

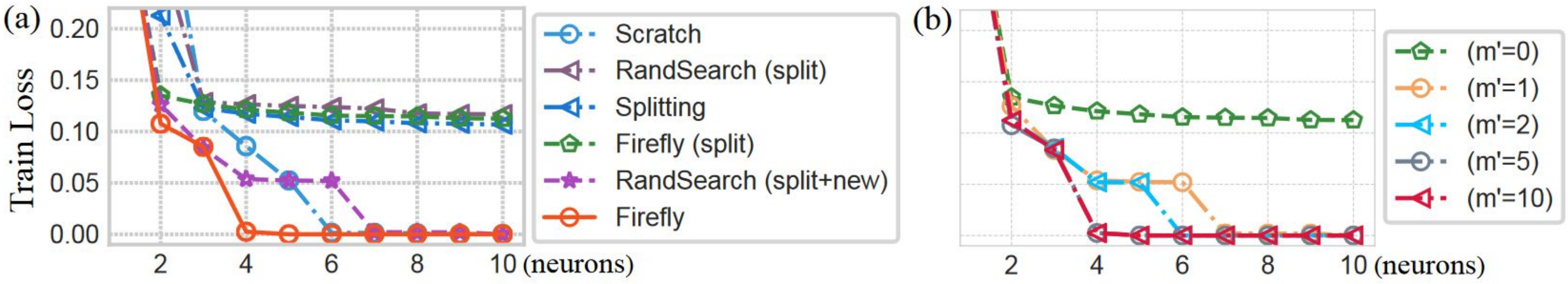


Figure 3: (a) Average training loss of different growing methods versus the number of grown neurons. (b) Firefly descent with different numbers of new neuron candidates.

Comparison with other methods

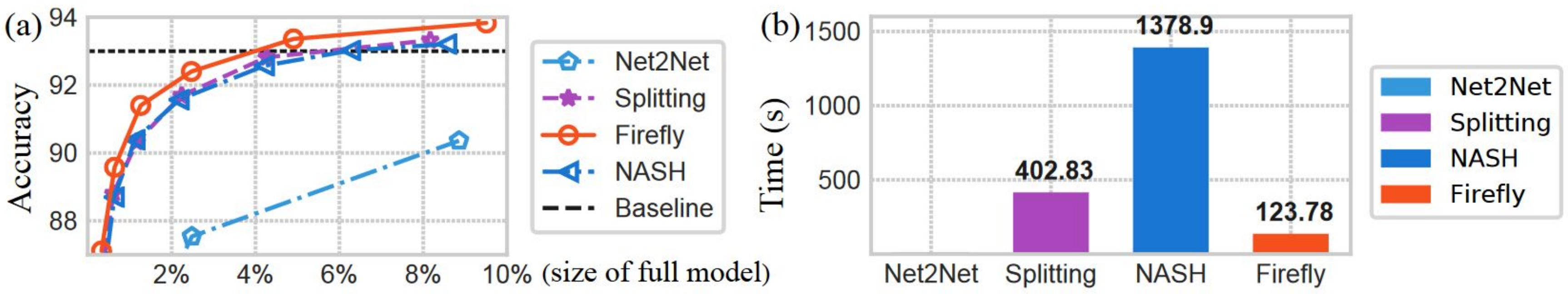


Figure 4: (a) Results of growing increasingly wider networks on CIFAR-10; VGG-19 is used as the backbone. (b) Computation time spent on growing for different methods.