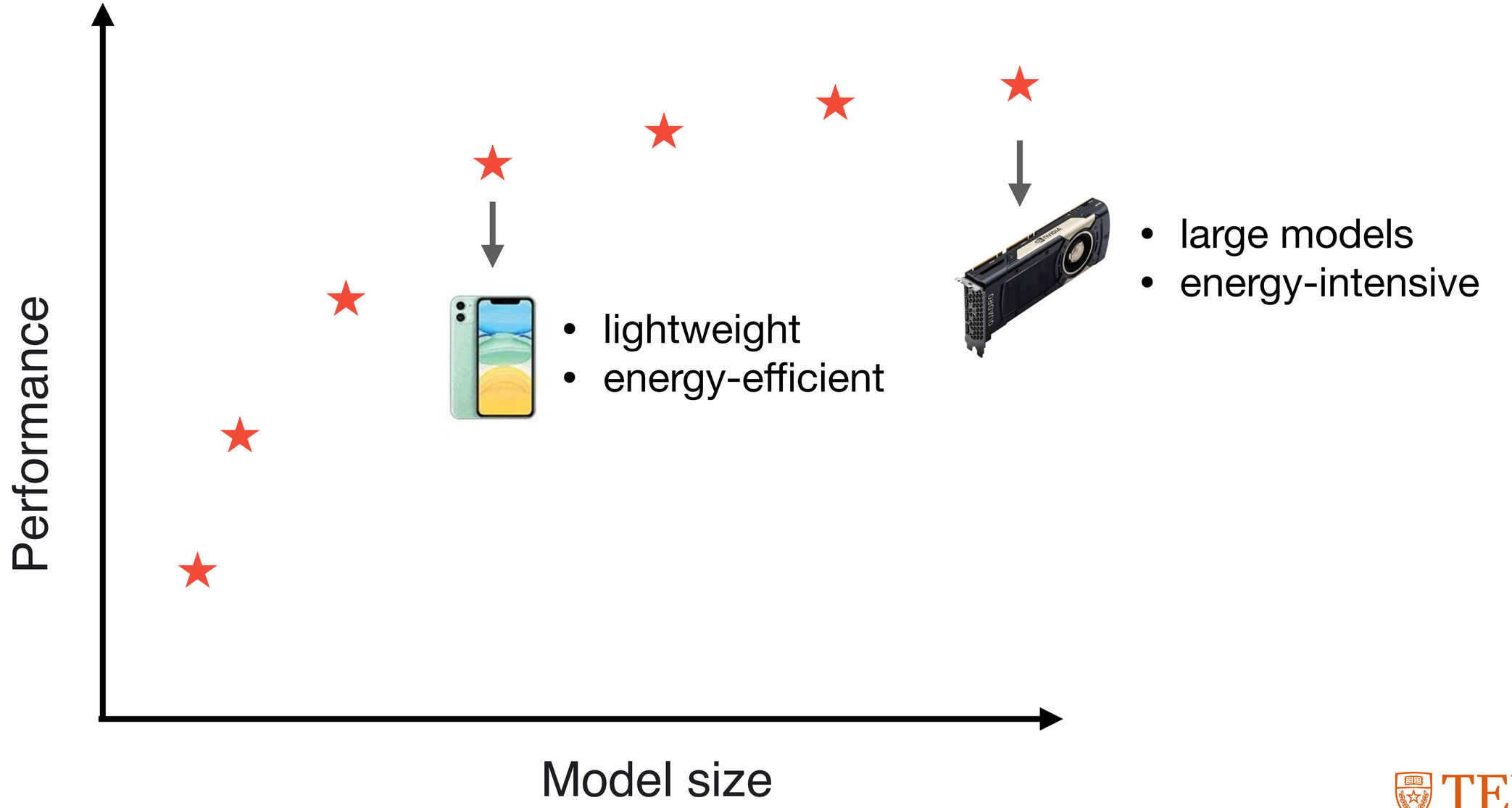
Splitting Steepest Descent for Growing Neural Architectures

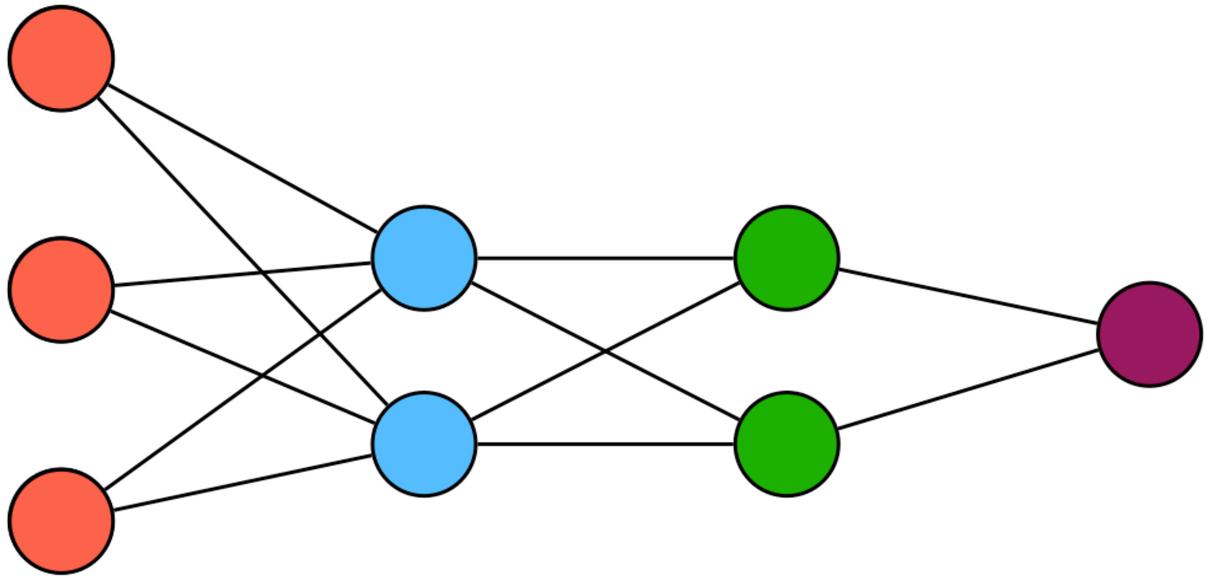
Qiang Liu, Lemeng Wu* and Dilin Wang*

Department of Computer Science
UT Austin

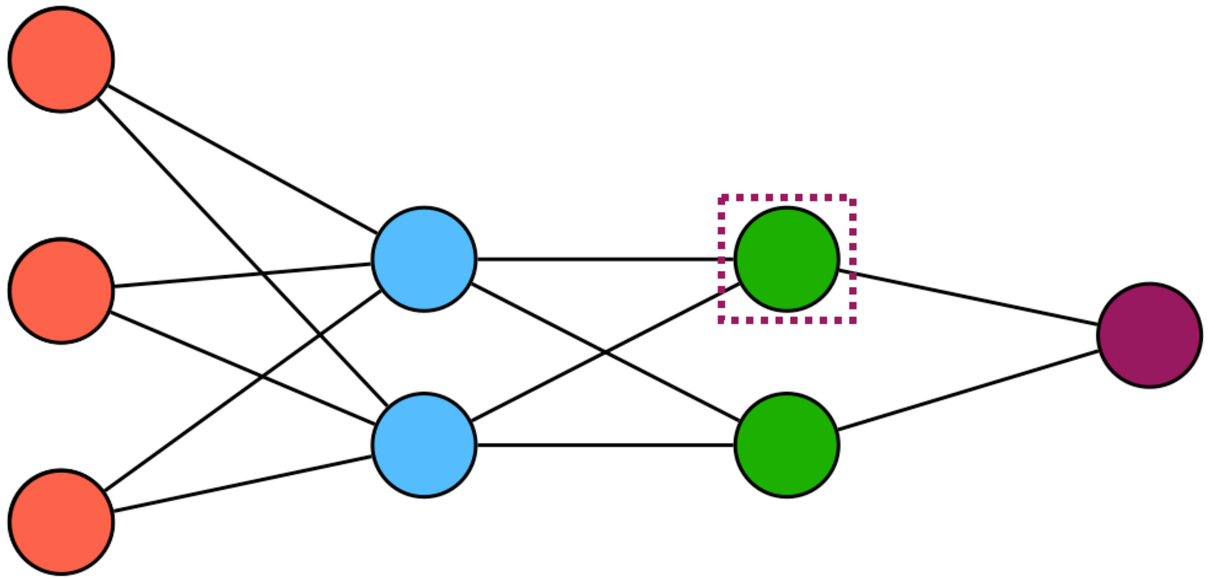
Our goal: finding small & accurate neural networks



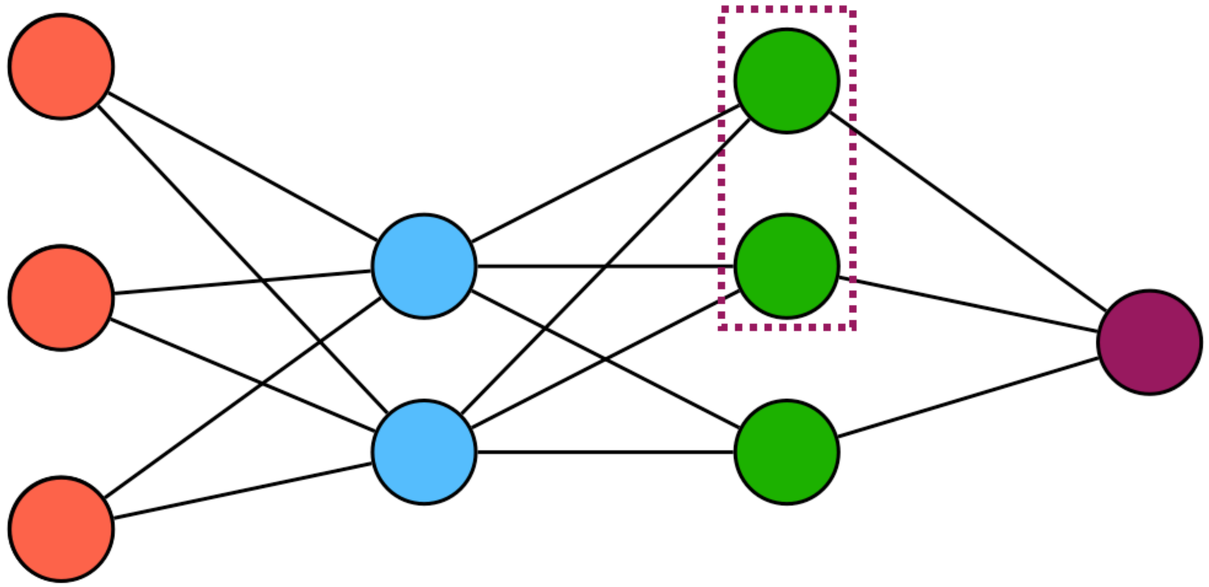
Splitting yields adaptive net structure optimization



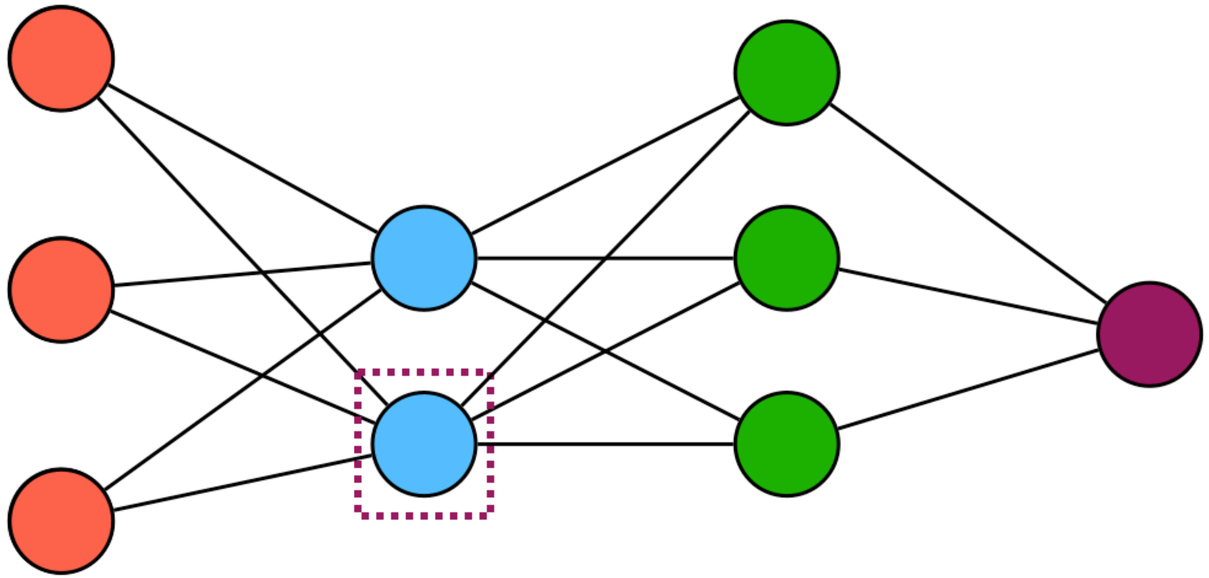
Splitting yields adaptive net structure optimization



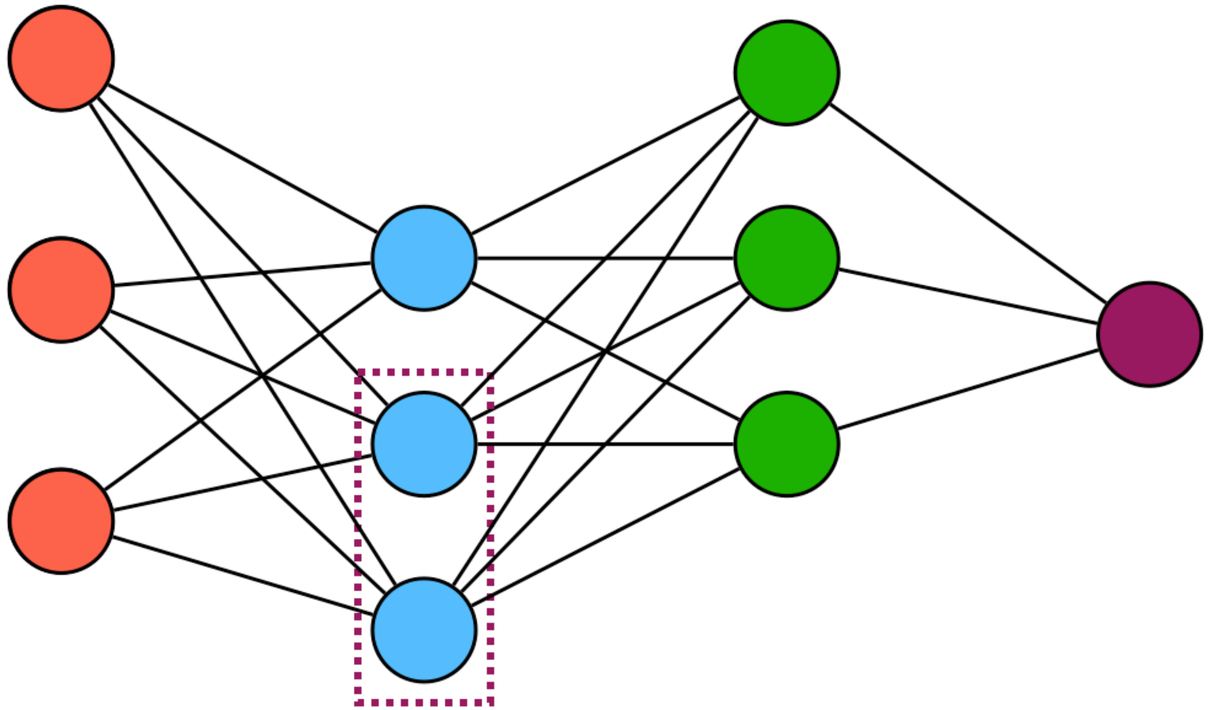
Splitting yields adaptive net structure optimization



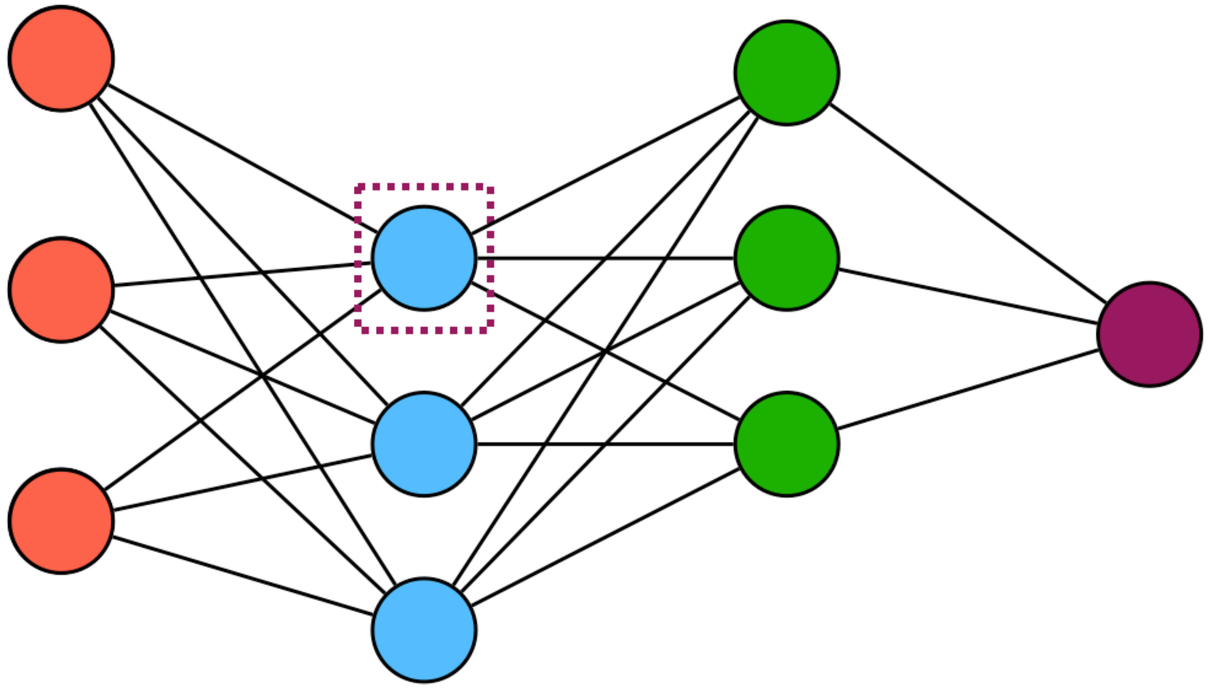
Splitting yields adaptive net structure optimization



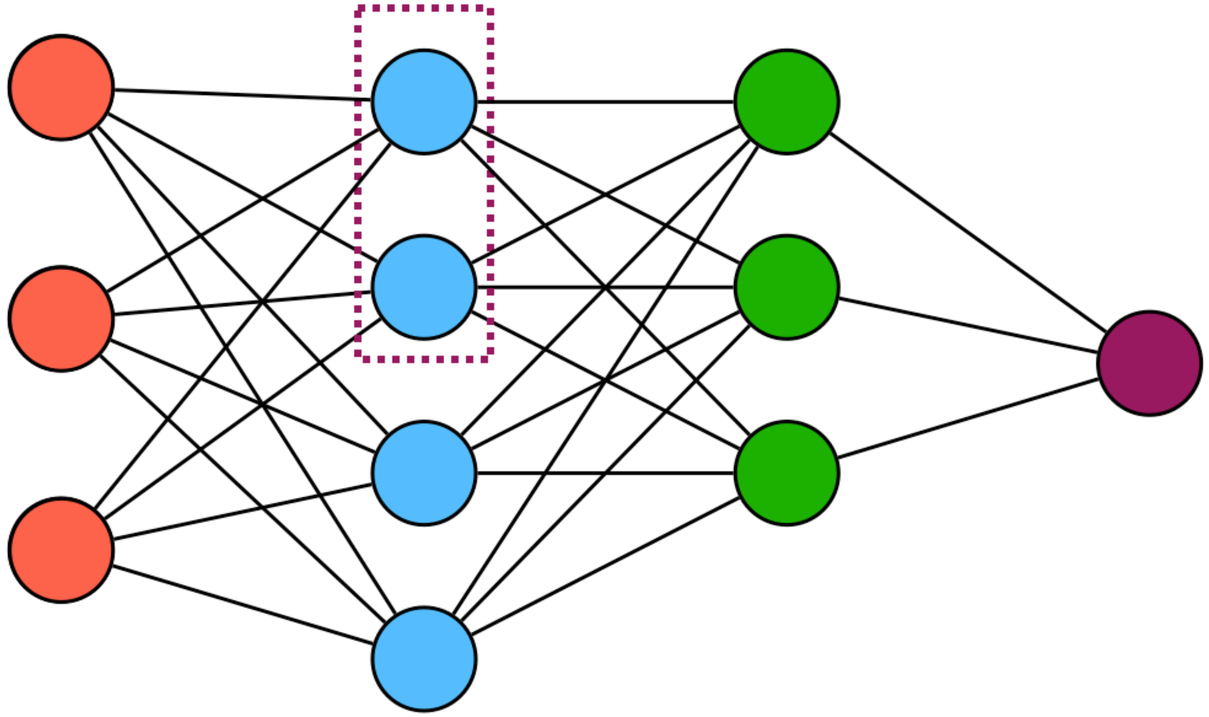
Splitting yields adaptive net structure optimization



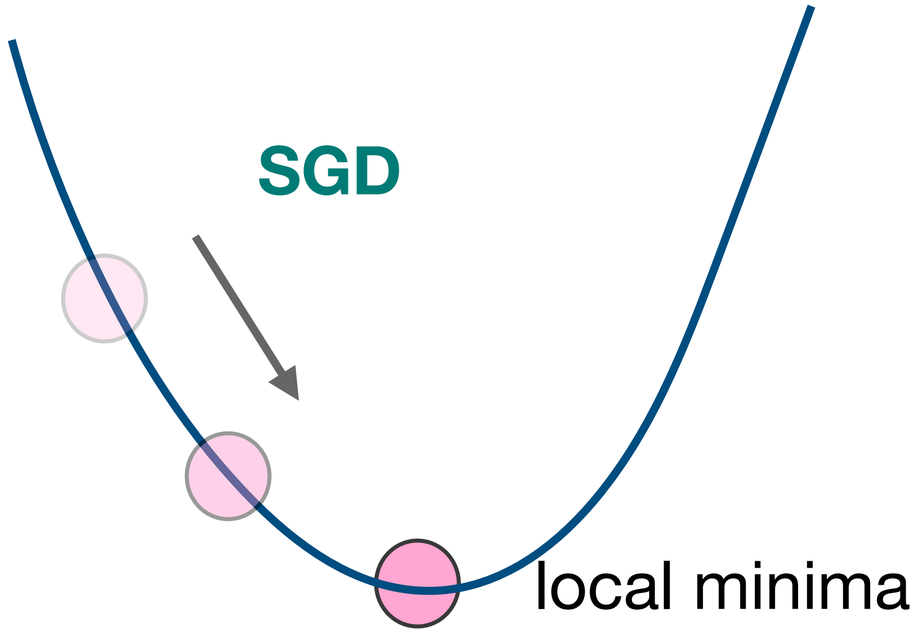
Splitting yields adaptive net structure optimization



Splitting yields adaptive net structure optimization



Intuition: escaping local minima



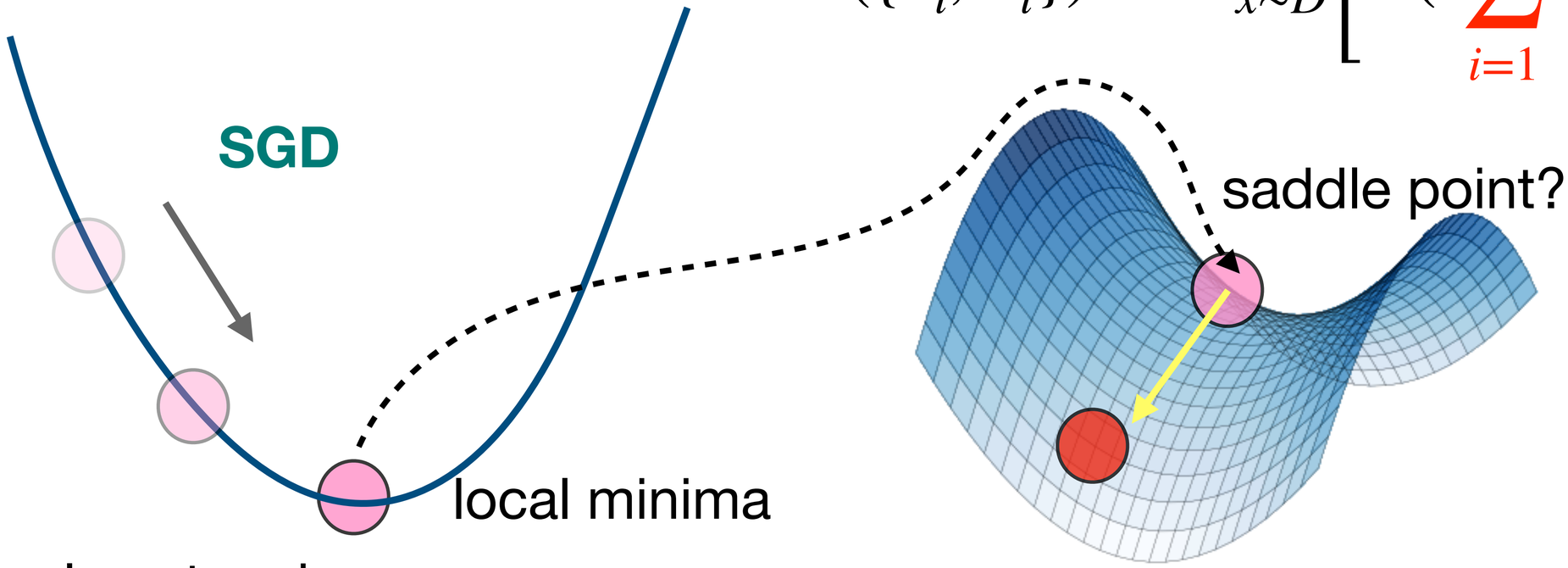
- A simple network:

$$\mathscr{L}(\theta) := \mathbb{E}_{x \sim D}\Big[\Phi(\ \sigma(\theta, x)\ )\Big] .$$

Intuition: escaping local minima

- Splitting θ into m copies $\{w_i, \theta_i\}_{i=1}^m$:

$$\mathscr{L}(\{\theta_i, w_i\}) := \mathbb{E}_{x\sim D}\left[\Phi\left(\sum_{i=1}^{m} w_i\ \sigma(\theta_i, x)\ \right)\right]$$



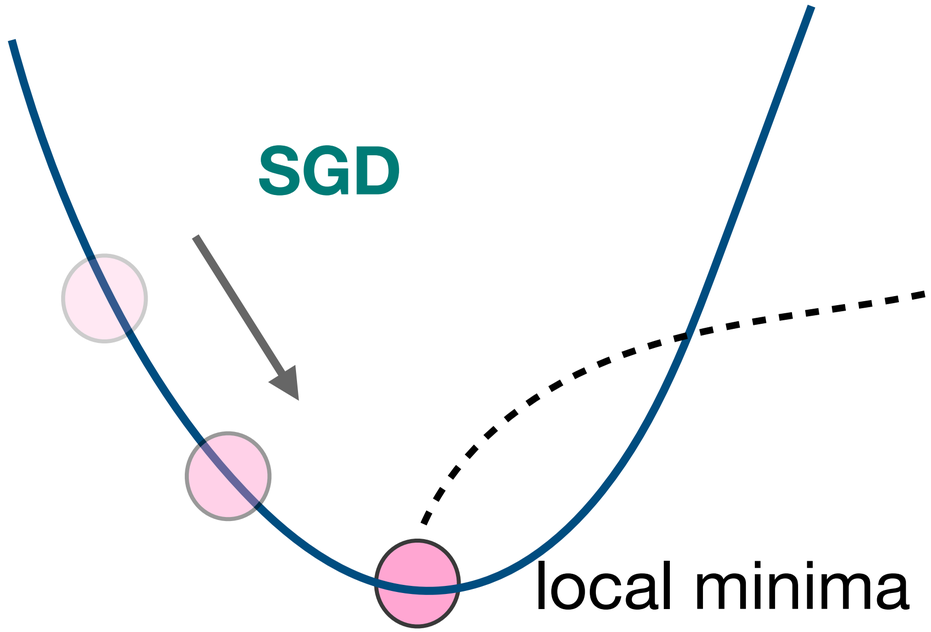
- A simple network:

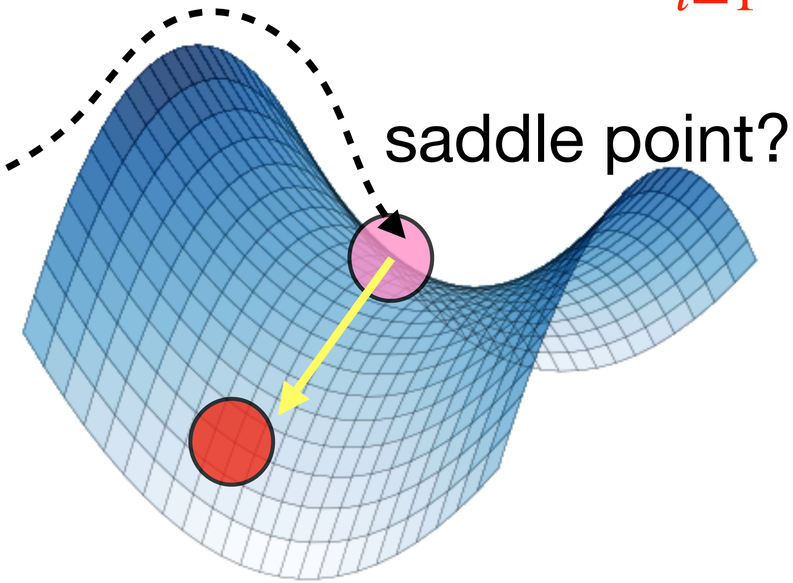
$$\mathscr{L}(\theta) := \mathbb{E}_{x\sim D}\left[\Phi\left(\ \sigma(\theta, x)\ \right)\right].$$

Intuition: escaping local minima

- Splitting θ into m copies $\{w_i, \theta_i\}_{i=1}^m$:

$$\mathscr{L}(\{\theta_i, w_i\}) := \mathbb{E}_{x\sim D}\left[\Phi\left(\sum_{i=1}^{m} w_i\ \sigma(\theta_i, x)\ \right)\right]$$





- A simple network:

$$\mathscr{L}(\theta) := \mathbb{E}_{x\sim D}\left[\Phi\left(\ \sigma(\theta, x)\ \right)\right].$$

- ***Smooth loss change:***

$$\sum_{i=1}^{m} w_i = 1,\ ||\theta_i - \theta||_2 \leq \epsilon.$$

Splitting Steepest Descent

- How to choose m and $\{\theta_i, w_i\}$ optimally?

$$\min_{m,\{\theta_i,w_i\}_{i=1}^m} \left\{ \mathscr{L}(\{\theta_i, w_i\}) - \mathscr{L}(\theta) \quad \textbf{s.t.} \quad ||\theta_i - \theta||_2 \leq \epsilon, \ \sum_{i=1}^m w_i = 1, \ \ w_i > 0, \forall\ i \right\}.$$

Splitting Steepest Descent

- How to choose m and $\{\theta_i, w_i\}$ optimally?

$$\min_{m,\{\theta_i,w_i\}_{i=1}^m} \left\{ \mathscr{L}(\{\theta_i, w_i\}) - \mathscr{L}(\theta) \quad \textbf{s.t.} \quad ||\theta_i - \theta||_2 \leq \epsilon, \ \sum_{i=1}^m w_i = 1, \ \ w_i > 0, \forall\ i \right\}.$$

Splitting-index, minimum eigenvalue

$$= \frac{\epsilon^2}{2} \underbrace{\min \left\{ \lambda_{\min}(S(\theta)), \ \ 0 \right\}}_{\textbf{CLOSED-FORM}} + \mathcal{O}(\epsilon^3) \qquad \textbf{with} \quad S(\theta) = \mathbb{E}_{x\sim D}\left[\nabla_\sigma \Phi(\sigma(\theta, x)) \ \nabla^2_{\theta\theta}\sigma(\theta, x) \right],$$

Splitting-matrix

Splitting Steepest Descent

- How to choose m and $\{\theta_i, w_i\}$ optimally?

$$\min_{m,\{\theta_i,w_i\}_{i=1}^m} \left\{ \mathscr{L}(\{\theta_i, w_i\}) - \mathscr{L}(\theta) \quad \textbf{s.t.} \quad ||\theta_i - \theta||_2 \le \epsilon, \ \sum_{i=1}^m w_i = 1, \ \ w_i > 0, \ \forall\ i \right\}.$$

Splitting-index, minimum eigenvalue

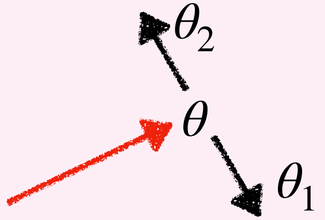
$$= \frac{\epsilon^2}{2} \underbrace{\min \left\{ \lambda_{\min}(S(\theta)), \ \ 0 \right\}}_{\textbf{CLOSED-FORM}} + \mathcal{O}(\epsilon^3) \qquad \textbf{with} \quad S(\theta) = \mathbb{E}_{x\sim D}\left[\nabla_\sigma \Phi(\sigma(\theta, x)) \ \nabla^2_{\theta\theta} \sigma(\theta, x) \right],$$

Splitting-matrix

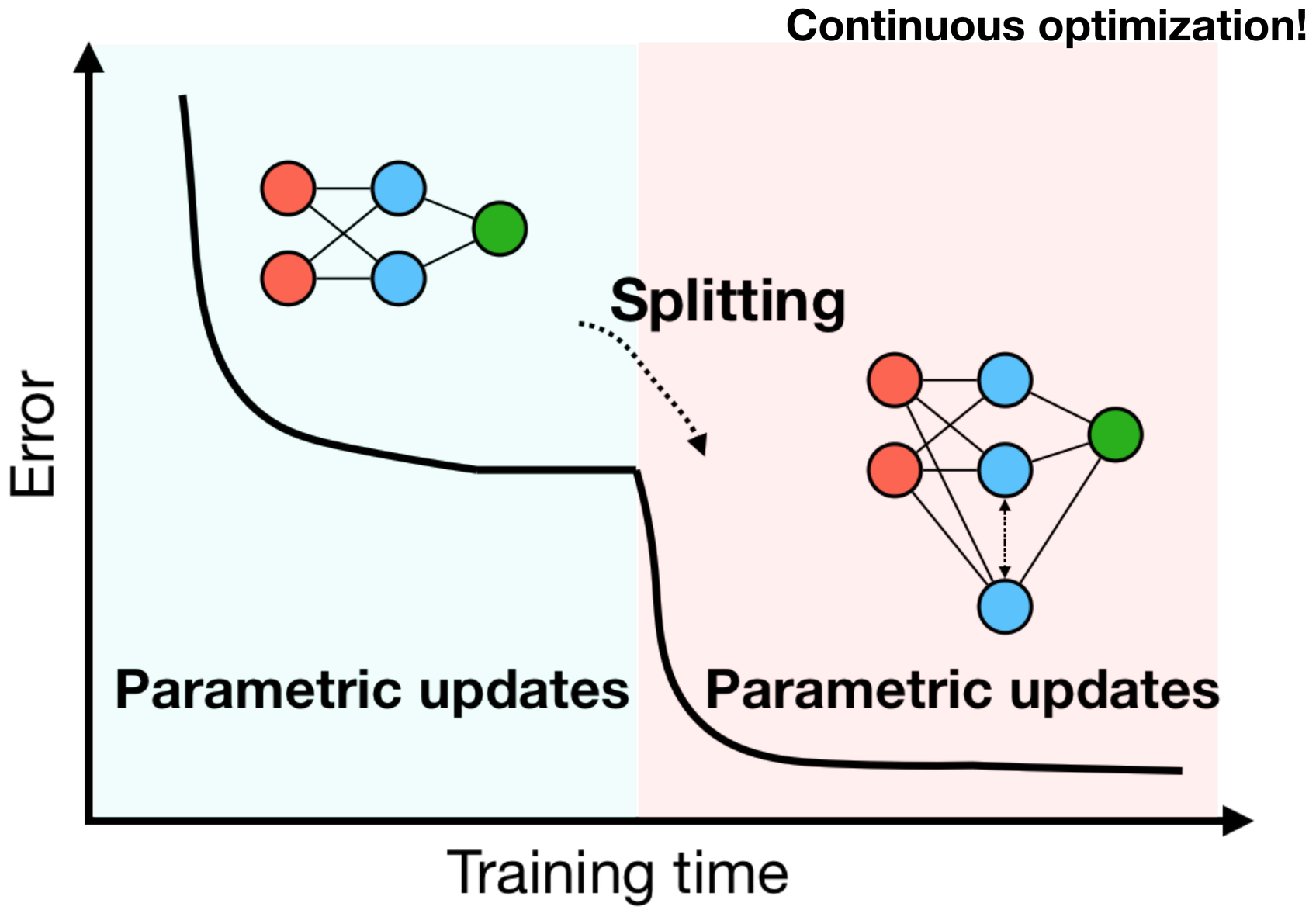
- Optimal splitting strategy

$$\lambda_{\min} S(\theta) \ge 0, \quad \text{no splitting}$$

$$\lambda_{\min} S(\theta) < 0, \quad m = 2, \ \theta_1 = \theta + \epsilon v_{\min}(S(\theta)), \ \ \theta_2 = \theta - \epsilon v_{\min}(S(\theta)), \ \ w_1 = w_2 = 1/2.$$

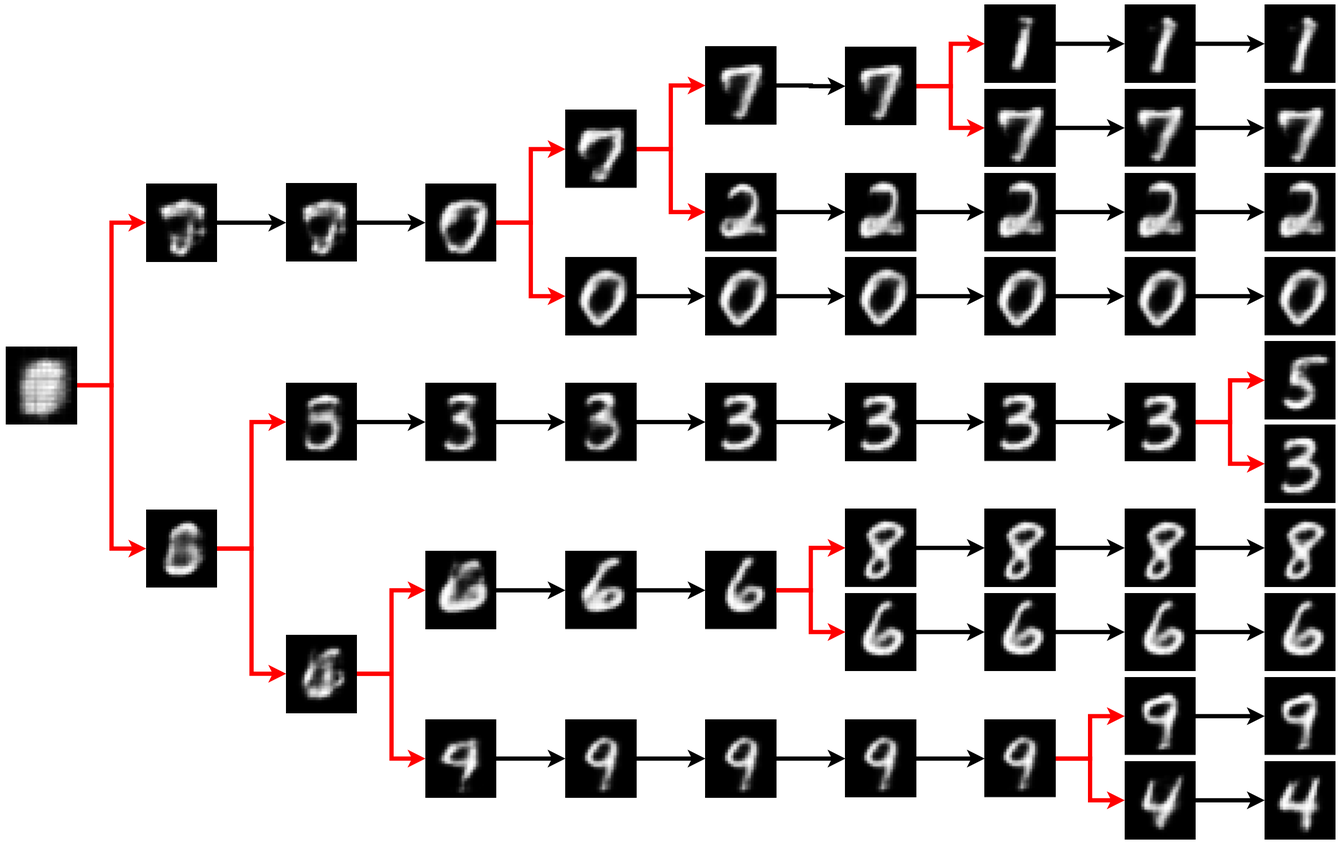


Our Algorithm



Growing Interpretable Networks

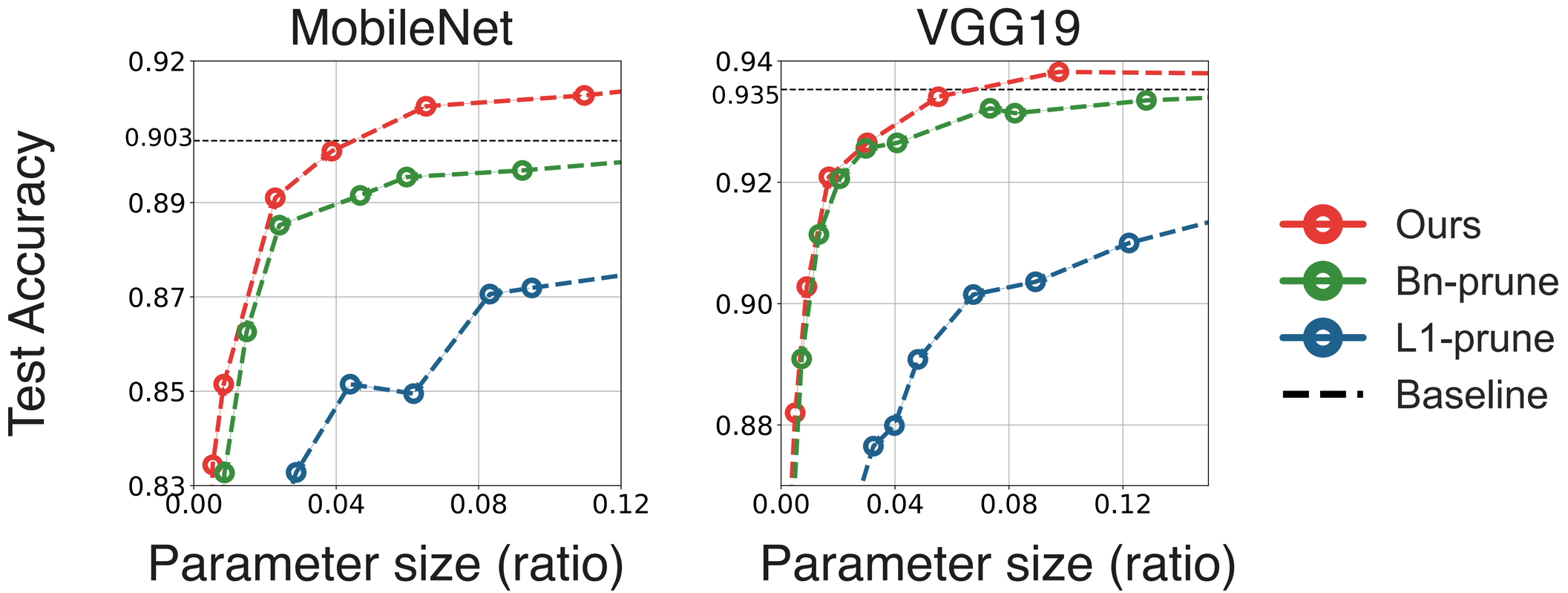
- Training the interpretable neural network by Li et al., 2018[1].



1. Li et al., Deep learning for case-based reasoning through prototypes: A neural network that explains its prediction. AAAI. 2018.

Results on CIFAR10

- Compare with pruning methods: batch-normalization-based pruning (Bn-prune) (Liu et al., 2017[1]) and L1-based pruning (L1-prune) (Li et al., 2017[2])

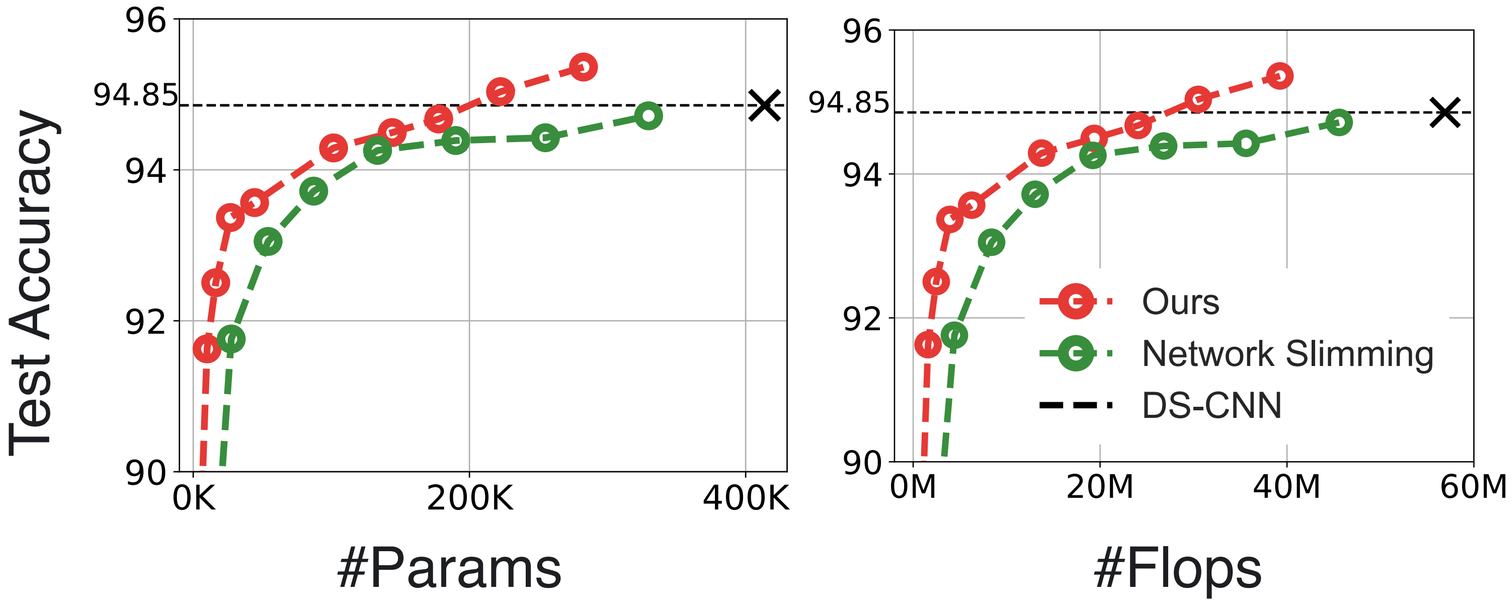


1. Liu et al., Learning efficient convolutional networks through network slimming. ICCV. 2017.
2. Li et al., Pruning filters for efficient convnets. ICLR. 2017

Keyword Spotting on Microcontrollers

- Identifying a set of keywords from speech signal, e.g. “hey siri"
- use benchmark from Zhang et al., 2017[1].

Method	Acc	Params (K)	Ops (M)
DNN	86.94	495.7	1.0
CNN	92.64	476.7	25.3
BasicLSTM	93.62	492.6	47.9
LSTM	94.11	495.8	48.4
GRU	94.72	498.0	48.4
CRNN	94.21	485.0	19.3
DS-CNN	94.85	413.7	56.9
Ours	**95.36**	**282.6**	**39.2**



1. Zhang et al., Hello edge: Keyword spotting on microcontrollers. arXiv preprint arXiv: 1711.07128. 2017

Conclusion

- Incremental neural structure optimization with splitting gradient
- Simple and fast, promising in practice

Thank you!

Poster #35, Today 10:45am – 12:45am @East Exhibition Hall B+C

TEXAS
The University of Texas at Austin