Energy-Aware Neural Architecture Optimization with Splitting Steepest Descent

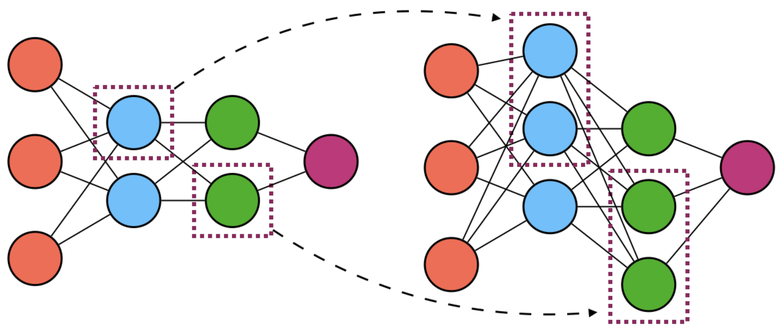
Dilin Wang[1], Lemeng Wu[1], Meng Li[2], Vikas Chandra[2], Qiang Liu[1]

[1] UT Austin [2] Facebook

Neural architecture optimization

Splitting Yields Adaptive Net Structure Optimization

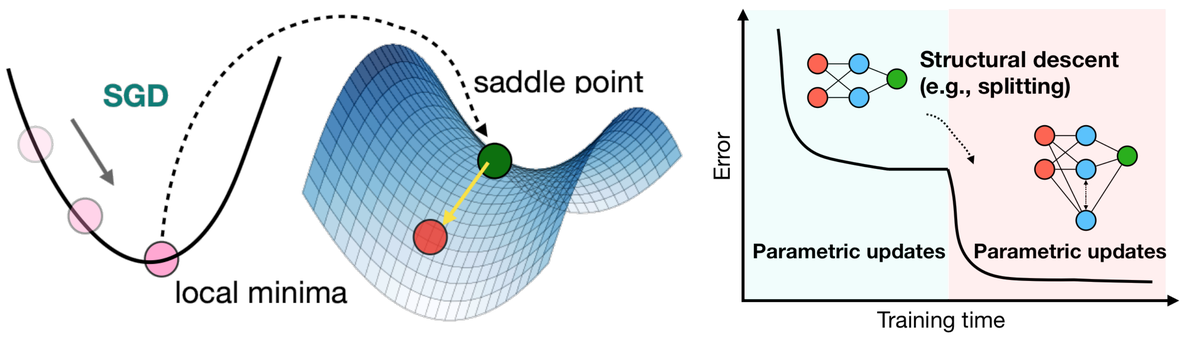
- Starting from a small net, gradually grow the net during training.
- Grow by "splitting" existing neurons into multiple off-springs.



Why, when and how?

- Why splitting? Does splitting decrease the loss? How much?
- When to split? What neurons should be split first?
- How to split a neuron optimally? How many copies to split into?

Why & when: escaping local minima



- **Optimization view**: the local optima in the low dimensional space can be turned into a saddle point in a higher dimensional of the augmented networks
- **Architecture view**: lower-dimensional space $\rightarrow$ smaller networks; higher-dimensional space $\rightarrow$ larger networks

How: splitting steepest descent

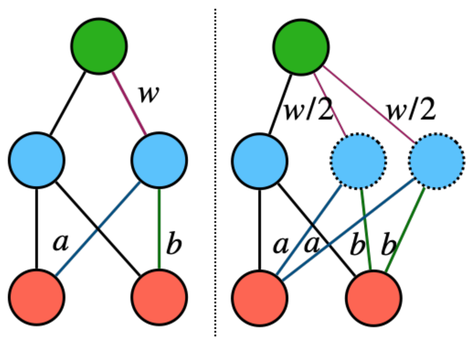
- Consider a single-neuron network

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

where $\Phi(\cdot)$ is the map from the output of the neuron to the final loss.

- Split θ into m off-springs: $\theta \rightarrow \{\theta_i, w_i\}_{i=1}^m$, we have,

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



- **Smooth loss change**:

$$\sum_{i=1} w_i = 1,\ ||\theta_i - \theta||_2 \le \epsilon, \forall i.$$

Deriving optimal splitting strategies

- Structural descent at stable local minima

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

- The optimum of Eqn. 1 is determined by

$$\min_{m,\{\theta_i\},\{w_i\}} \left\{\mathcal{L}_m(\{\theta_i, w_i\}) - \mathcal{L}(\theta)\right\} = \frac{\epsilon^2}{2}\min\{\underbrace{\lambda_{\min}(S(\theta))}_{\text{splitting index}}, 0\} + \mathcal{O}(\epsilon^3),$$

$$\text{with}\quad S(\theta) = \underbrace{\mathbb{E}_{x\sim D}\left[\nabla_\sigma\Phi(\sigma(\theta, x))\nabla^2_{\theta\theta}\sigma(\theta, x),\right]}_{\text{Splitting matrix}}$$

where $\lambda_{\min}(S(\theta))$ denotes the minimum eigenvalue of $S(\theta)$.

Optimal splitting

- When $\lambda_{\min}(S(\theta)) \ge 0$, no splitting
- When $\lambda_{\min}(S(\theta)) < 0$:

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

The corresponding maximum decrease of loss is $\epsilon^2\lambda_{\min}(S(\theta))/2$.

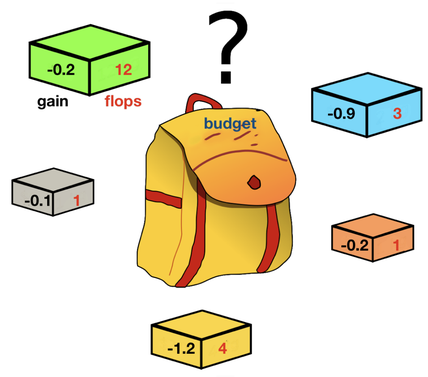
Energy-aware splitting

- Our formulation

$$\min_{\beta} \quad \sum_{\ell=1}^n \beta_\ell \underbrace{\lambda_{\min}(S(\theta^\ell))}_{gain}$$

$$s.t. \quad \beta_\ell \in \{0, 1\}$$

$$\sum_{\ell=1}^n \underbrace{e_\ell}_{flops}\beta_\ell \le \text{budget}$$

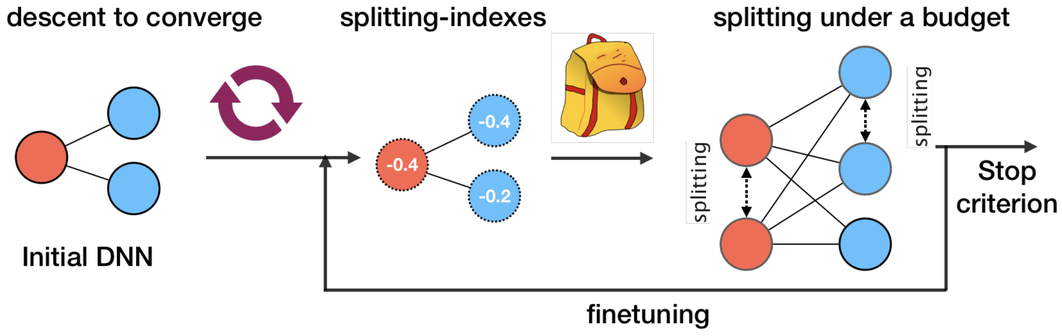


Main algorithm

Step 1: Run gradient descent to converge

Step 2: calculate the splitting-indexes

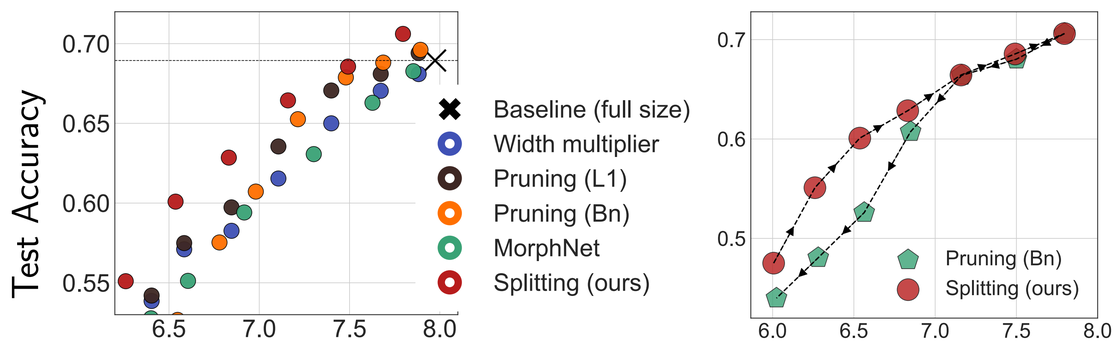
Step 3: grow the network via splitting under a budget



Experiments

Results on CIFAR100

- We apply splitting on a small version of MobileNetV1 Howard et al., 2017



(a) Log10(flops) (b) Log10(flops)

Results on ImageNet

- MobileNetV1

Model	MACs (G)	Top-1 Accuracy	Top-5 Accuracy
MobileNetV1 (1.0x)	0.569	72.93	91.14
Splitting-4	**0.561**	**73.96**	**91.49**
MobileNetV1 (0.75x)	0.317	70.25	89.49
AMC He el al., 2018	0.301	70.50	89.30
Splitting-3	**0.292**	**71.47**	**89.67**
MobileNetV1 (0.5x)	0.150	65.20	86.34
Splitting-2	**0.140**	**68.26**	**87.93**
Splitting-1	0.082	64.06	85.30
Splitting-0 (seed)	0.059	59.20	81.82

- MobileNetV2

Model	MACs (G)	Top-1 Accuracy	Top-5 Accuracy
MobileNetV2 (1.0x)	0.300	72.04	90.57
Splitting-3	**0.298**	**72.84**	**90.83**
MobileNetV2 (0.75x)	0.209	69.80	89.60
AMC He et al., 2018	0.210	70.85	89.91
Splitting-2	**0.208**	**71.76**	**90.07**
MobileNetV2 (0.5x)	0.097	65.40	86.40
Splitting-1	**0.095**	**66.53**	**87.00**
Splitting-0 (seed)	0.039	55.61	79.55

Conclusion

- Incremental training with splitting gradient.
- Simple and fast, promising in practice.
- Opens a new dimension for energy-efficient NAS.