

Obtaining Adjustable Regularization for Free via Iterate Averaging

Jingfeng Wu, Vladimir Braverman, Lin F. Yang
Johns Hopkins University & UCLA

June 2020

Searching optimal hyperparameter

ML/Opt problem

$$\min_w L(w) + \lambda R(w)$$

Main loss

Hyperparameter

Regularization

GD/SGD

$$w_{k+1} = w_k - \eta (\nabla L(w) + \lambda R(w))$$

$$w_k \rightarrow w_{\lambda}^*$$

Learning rate/step size

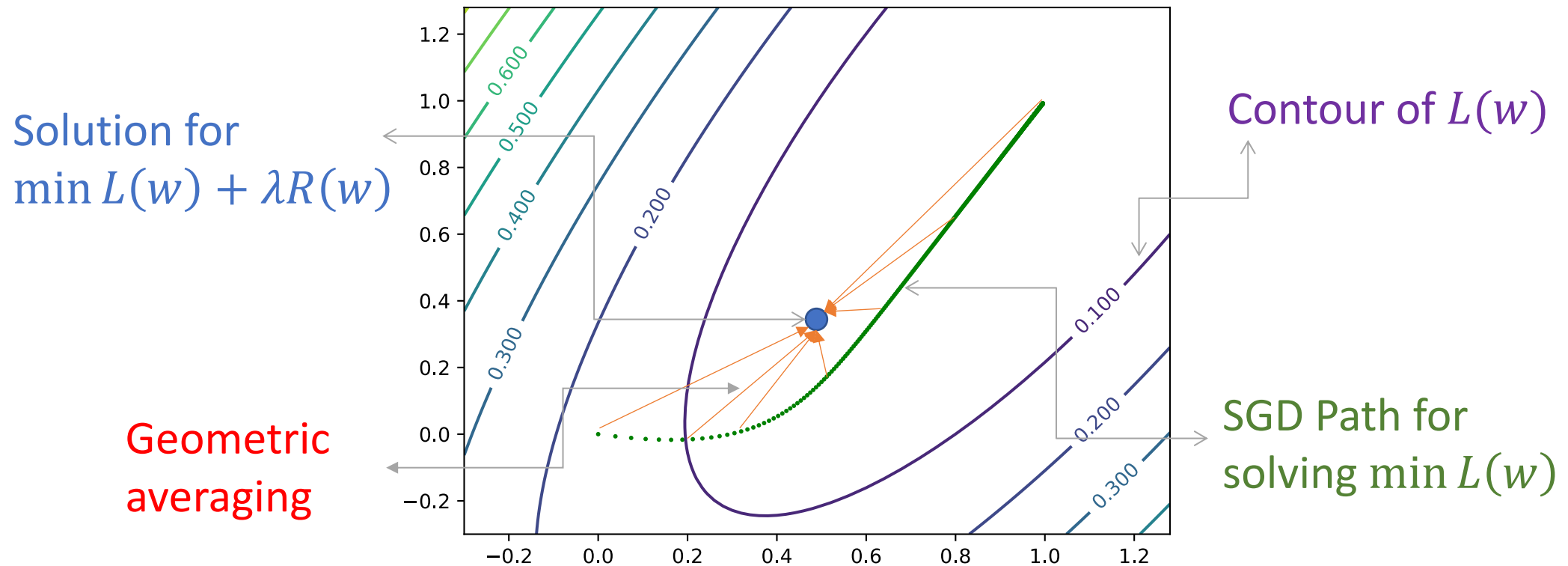
Re-running the optimizer is expensive! 😞

ResNet-50 + ImageNet + 8 GPUs

- A single round of training takes about *3 days*.
- *Almost a year* to try a hundred different hyperparameters.

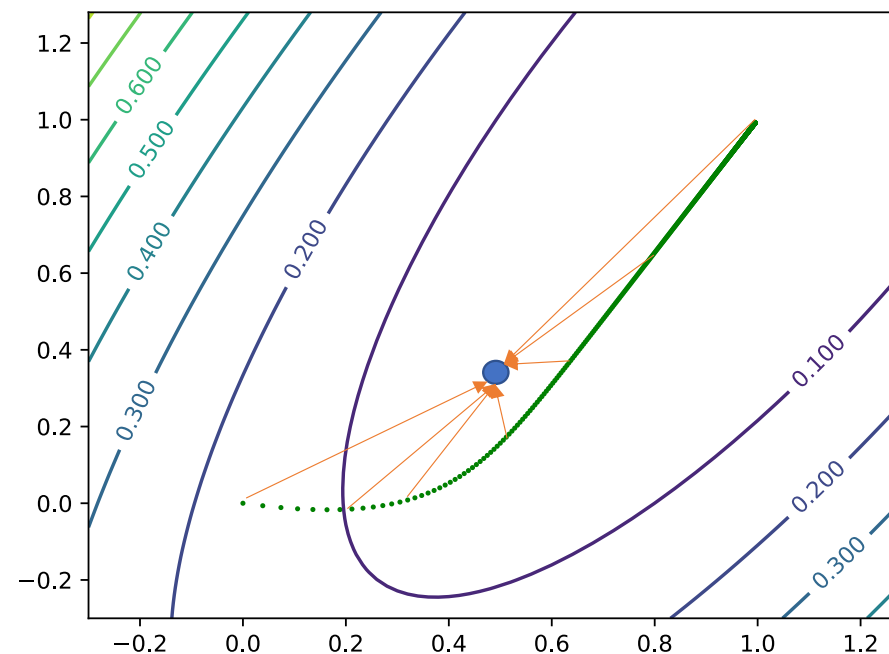
Can we obtain *adjustable* regularization for *free*?

Iterate averaging => regularization (Neu et al.)



Iterate averaging protocol

- Require: A stored opt. path
- Input: a hyperparameter λ
 - Compute a weighting scheme
 - Average the path
- Output: the regularized solution



Iterate averaging is cheap 😊

But Neu et al.'s result is limited 😞

Formally, Neu et al. shows

- Linear regression

$$L(w) = \frac{1}{n} \sum_{i=1}^n \|w^T x - y\|_2^2$$

- ℓ_2 -regularization

$$R(w) = \frac{1}{2} \|w\|_2^2$$

- GD/SGD path

$$w_{k+1} = w_k - \eta \nabla L(w)$$

- Geometric averaging

$$p_k = (1 - p)p^k, \quad p = \frac{1}{1 + \lambda \eta}$$

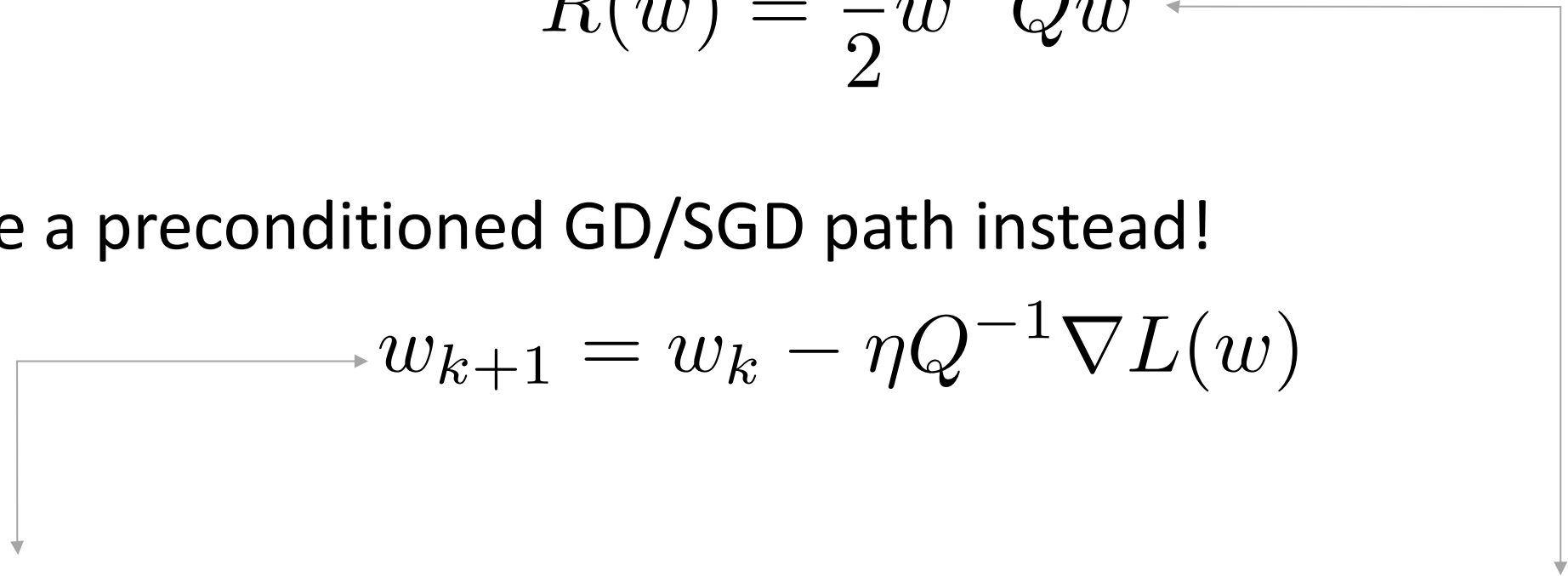
$$p_1 w_1 + p_2 w_2 + \cdots + p_k w_k \quad \text{ solves } \min_w L(w) + \lambda R(w)$$

Our contributions: 😊 😊 😊 😊

Iterate averaging works for more general

1. regularizers \leq generalized ℓ_2 -regularizer
2. optimizers \leq Nesterov's acceleration
3. objectives \leq strongly convex and smooth losses
4. deep neural networks! (Empirically)

1. Generalized ℓ_2 -regularization

$$R(w) = \frac{1}{2} w^\top Q w$$


Use a preconditioned GD/SGD path instead!

$$w_{k+1} = w_k - \eta Q^{-1} \nabla L(w)$$

$$p_1 w_1 + p_2 w_2 + \cdots + p_k w_k \quad \text{ solves } \quad \min_w L(w) + \lambda R(w)$$

2. Nesterov's acceleration

Weighting scheme

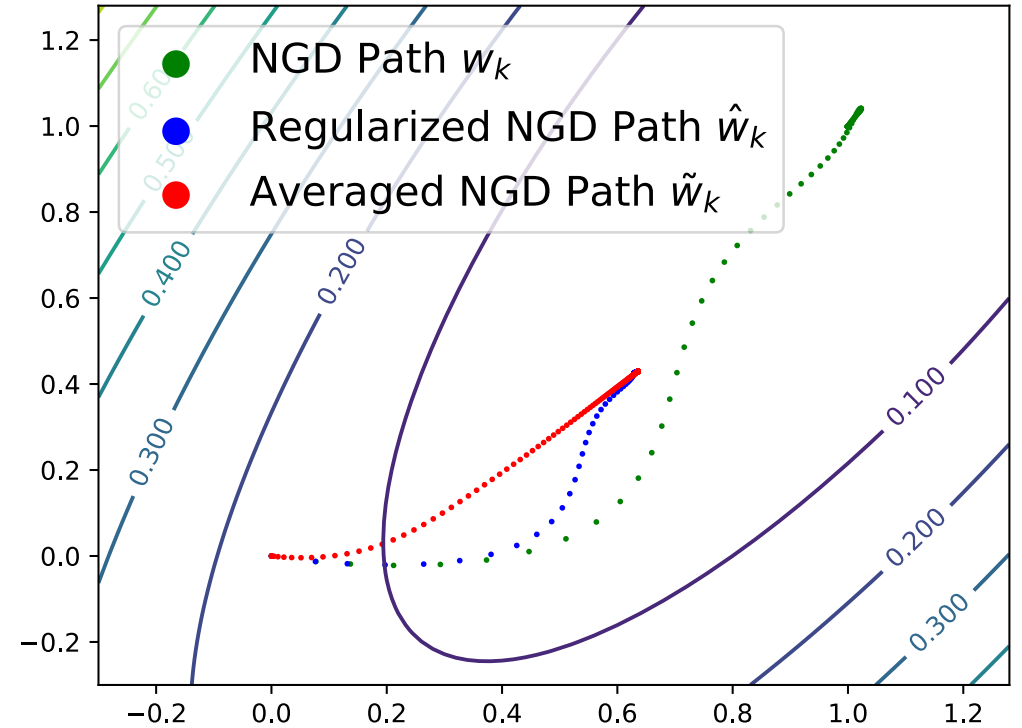
$$p_k = \frac{\gamma}{\eta} \left(\frac{\sqrt{\gamma(\alpha + \lambda)} - \sqrt{\eta\alpha}}{1 - \sqrt{\eta\alpha}} \right) \left(\frac{1 - \sqrt{\gamma(\alpha + \lambda)}}{1 - \sqrt{\eta\alpha}} \right)^{k-2}$$

where $\gamma = \frac{\eta}{1 + \lambda\eta}$

$$p_1 w_1 + p_2 w_2 + \cdots + p_k w_k$$

solves $\min_w L(w) + \lambda R(w)$

ℓ_2 -regularizer



3. Strongly convex and smooth objectives

Yes! But only approximately...

Geometric weighting scheme $p_k = (1 - p)p^k$



$$\hat{w}_{\lambda_1} \lesssim \sum_{k=1}^{\infty} p_k w_k \lesssim \hat{w}_{\lambda_2} \quad \hat{w}_{\lambda} = \arg \min L(w) + \lambda R(w)$$

ℓ_2 -regularizer

4. Deep neural networks ☺

Dataset	CIFAR-10		CIFAR-100
Model	VGG-16	ResNet-18	ResNet-18
Accuracy after training (%)	92.54 ± 0.22	94.54 ± 0.04	75.62 ± 0.16
Accuracy after averaging (%)	93.18 ± 0.06	94.72 ± 0.04	76.24 ± 0.05
Time of training	$\sim 4.5\text{h}$	$\sim 8.3\text{h}$	$\sim 8.3\text{h}$
Time of averaging	$\sim 47\text{s}$	$\sim 56\text{s}$	$\sim 58\text{s}$

A single GPU K80 →

Iterate averaging is effective and efficient!

Take Home

Iterate averaging => **adjustable** regularization for **free**

- For ℓ_2 -type regularization
- For SGD/NSGD optimizers
- For quadratic/strongly convex and smooth objectives
- Regularizing deep neural networks

Join our poster session for more details!