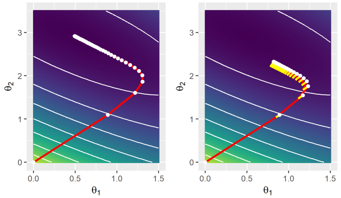
Introduction to Machine Learning

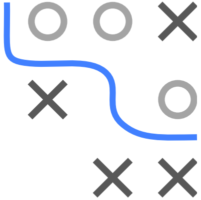
Regularization
Weight Decay and L2



Learning goals

- $L2$ regularization with GD is equivalent to weight decay
- Understand how weight decay changes the optimization trajectory

WEIGHT DECAY VS. L2 REGULARIZATION



Let's optimize $L2$-regularized risk of a model $f(\mathbf{x} \mid \boldsymbol{\theta})$

$$\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2$$

by GD. The gradient is

$$\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \boldsymbol{\theta}$$

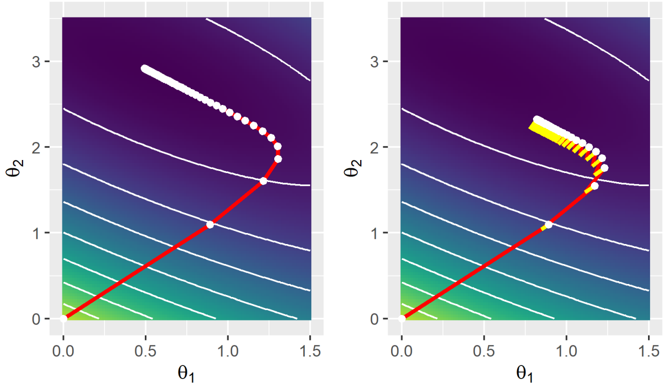
We iteratively update $\boldsymbol{\theta}$ by step size α times the negative gradient

$$\begin{aligned} \boldsymbol{\theta}^{[\text{new}]} &= \boldsymbol{\theta}^{[\text{old}]} - \alpha \left(\nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]}) + \lambda \boldsymbol{\theta}^{[\text{old}]} \right) \\ &= \boldsymbol{\theta}^{[\text{old}]}(1 - \alpha\lambda) - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]}) \end{aligned}$$

We see how $\boldsymbol{\theta}^{[old]}$ decays in magnitude – for small α and λ – before we do the gradient step. Performing the decay directly, under this name, is a very well-known technique in DL - and simply $L2$ regularization in disguise (for GD).

WEIGHT DECAY VS. L2 REGULARIZATION / 2

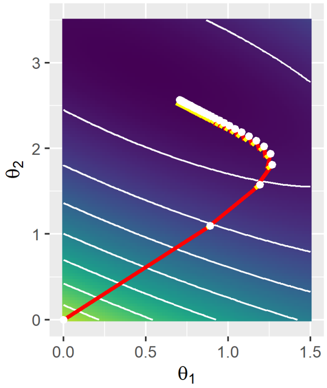
In GD With WD, we slide down neg. gradients of $\mathcal{R}_{\text{emp}}$,
but in every step, we are pulled back to origin.



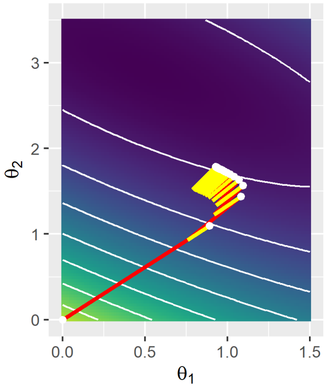
Without WD　　　　　With GD

WEIGHT DECAY VS. L2 REGULARIZATION / 3

How strongly we are pulled back (for fixed α) depends on λ:



Small λ



Large λ

CAVEAT AND OTHER OPTIMIZERS

Caveat: Equivalence of weight decay and *L*2 only holds for (S)GD!

- ▸ Hanson and Pratt 1988 originally define WD "decoupled" from gradient-updates $\alpha \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]})$ as
  $\boldsymbol{\theta}^{[\text{new}]} = \boldsymbol{\theta}^{[\text{old}]}(1 - \lambda') - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}^{[\text{old}]})$
- This is equivalent to modern WD/*L*2 (last slide) using reparameterization $\lambda' = \alpha\lambda$
- Consequence: if there is optimal λ', then optimal *L*2 penalty is tightly coupled to α as $\lambda = \lambda'/\alpha$ (and vice versa)
- ▸ Loshchilov and Hutter 2019 show no equivalence of *L*2 and WD possible for adaptive methods like Adam (Prop. 2)
- In many cases where SGD+*L*2 works well, Adam+*L*2 underperforms due to non-equivalence with WD
- They propose a variant of Adam decoupling WD from gradient updates (AdamW), increasing performance over Adam+*L*2