Introduction to Deep Learning

Chapter 3: Basic Regularization

Bernd Bischl

Department of Statistics – LMU Munich

WS 2021/2022

REGULARIZATION

- Any technique that is designed to reduce the test error possibly at the expense of increased training error can be considered a form of regularization.
- Regularization is important in DL because NNs can have extremely high capacity (millions of parameters) and are thus prone to overfitting.

REVISION: REGULARIZED RISK MINIMIZATION

- The goal of regularized risk minimization is to penalize the complexity of the model to minimize the chances of overfitting.
- By adding a parameter norm penalty term $J(\boldsymbol{\theta})$ to the empirical risk $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ we obtain a regularized cost function:

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

  with hyperparamater $\lambda \in [0, \infty)$, that weights the penalty term, relative to the unconstrained objective function $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$.
- Therefore, instead of pure **empirical risk minimization**, we add a penalty for complex (read: large) parameters $\boldsymbol{\theta}$.
- Declaring $\lambda = 0$ obviously results in no penalization.
- We can choose between different parameter norm penalties $J(\boldsymbol{\theta})$.
- In general, we do not penalize the bias.

L2-REGULARIZATION / WEIGHT DECAY

Let us optimize the 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 gradient descent. The gradient is

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

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

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

$\rightarrow$ The term $\lambda \boldsymbol{\theta}^{[old]}$ causes the parameter (**weight**) to **decay** in proportion to its size (which gives rise to the name).

EQUIVALENCE TO CONSTRAINED OPTIMIZATION

Norm penalties can be interpreted as imposing a constraint on the weights. One can show that

$$\arg\min_{\theta} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda J(\boldsymbol{\theta})$$

is equvilalent to

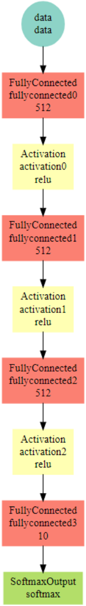
$$\arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$$
$$\text{subject to} \quad J(\boldsymbol{\theta}) \leq k$$

for some value k that depends on λ the nature of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$.

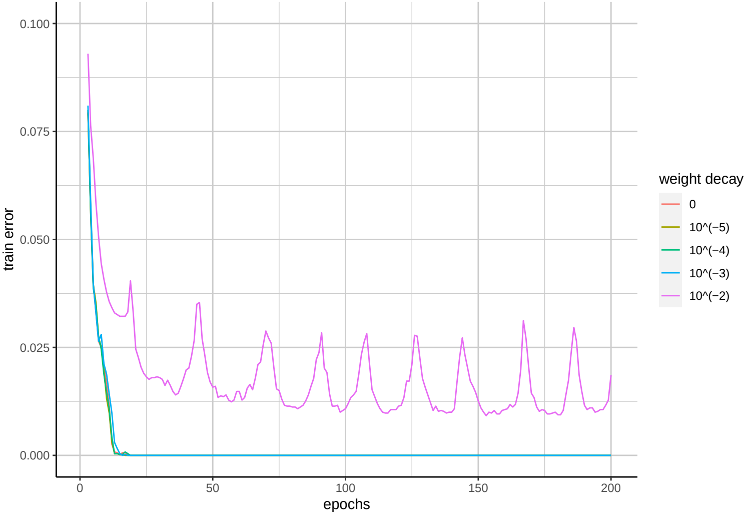
(Goodfellow et al. (2016), ch. 7.2)

EXAMPLE: WEIGHT DECAY

- We fit the huge neural network on the right side on a smaller fraction of MNIST (5000 train and 1000 test observations)
- Weight decay: $\lambda \in (10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0)$

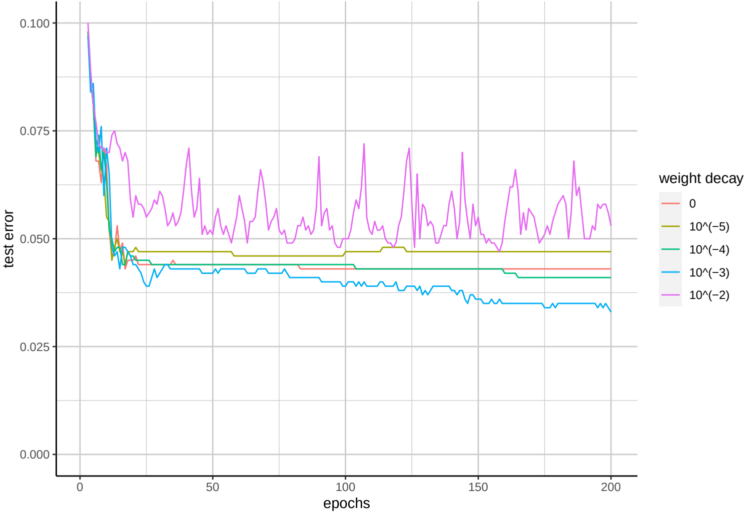


EXAMPLE: WEIGHT DECAY



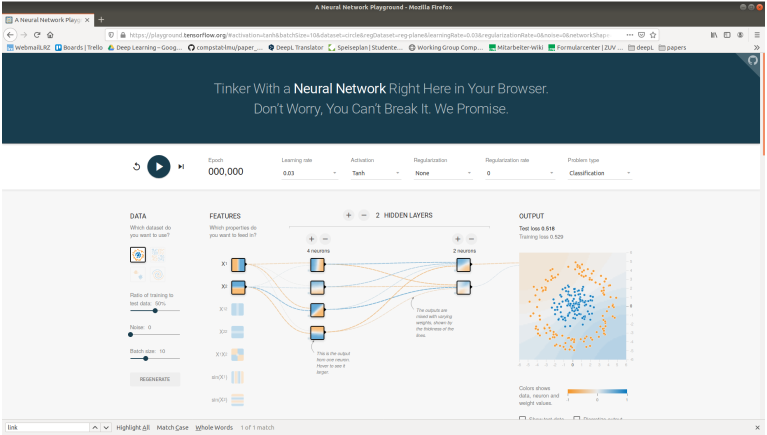
A high weight decay of 10^{-2} leads to a high error on the training data.

EXAMPLE: WEIGHT DECAY



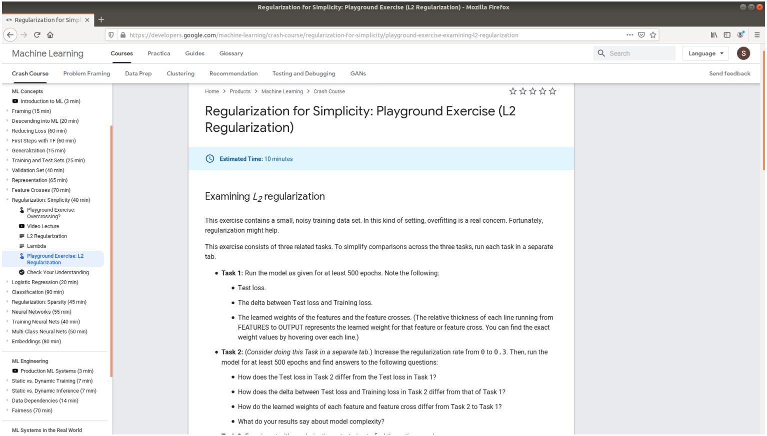
Second strongest weight decay leads to the best result on the test data.

TENSORFLOW PLAYGROUND



https://playground.tensorflow.org/

TENSORFLOW PLAYGROUND - EXERCISE



https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/playground-exercise-examining-l2-regularization