

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).

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 $J(\theta)$ to the empirical risk $\mathcal{R}_{\text{emp}}(\theta)$ we obtain a regularized cost function:

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

with hyperparameter $\lambda \in [0, \infty)$, that weights the penalty term, relative to the unconstrained objective function $\mathcal{R}_{\text{emp}}(\theta)$.

- Therefore, instead of pure **empirical risk minimization**, we add a penalty for complex (read: large) parameters θ .
- Declaring $\lambda = 0$ obviously results in no penalization.
- We can choose between different parameter norm penalties $J(\theta)$.
- 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

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

→ The term $\lambda \boldsymbol{\theta}^{[\text{old}]}$ 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

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

is equivalent to

$$\arg \min_{\theta} \mathcal{R}_{\text{emp}}(\theta)$$

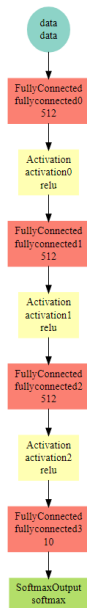
$$\text{subject to } J(\theta) \leq k$$

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

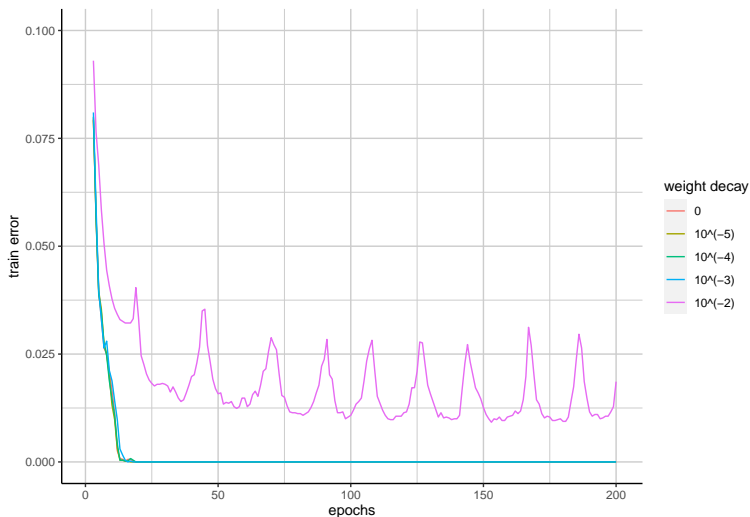
(Goodfellow et al. (2016), ch. 7.2)

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: $\lambda \in (10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0)$

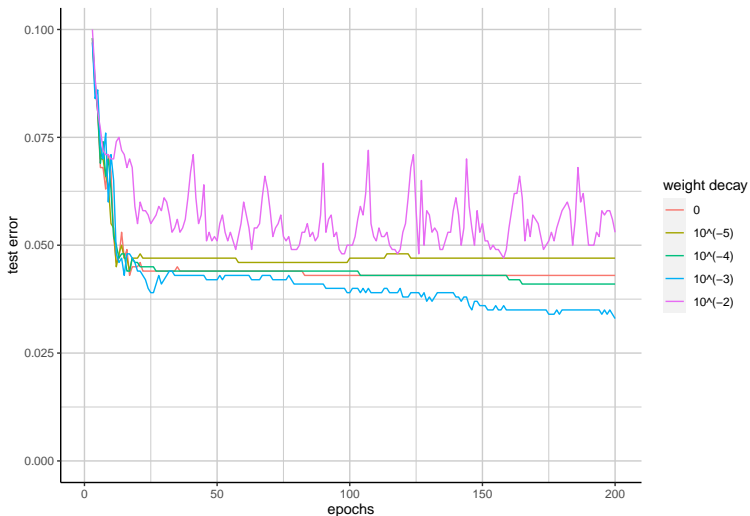


EXAMPLE: WEIGHT DECAY



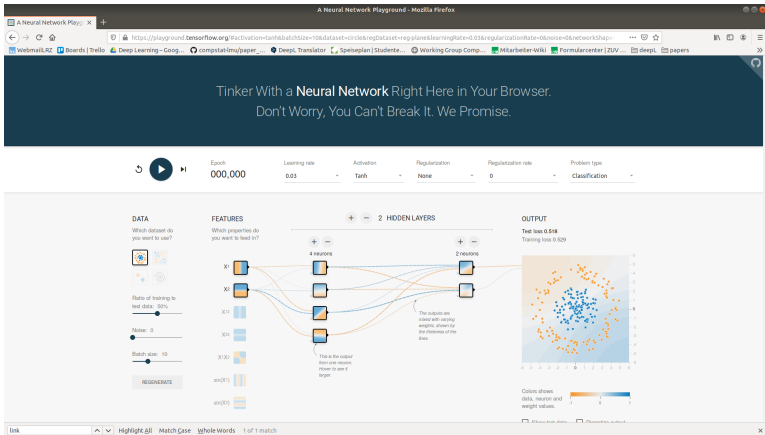
A high weight decay of 10^{-2} leads to a high error on the training data.

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

The screenshot shows a web browser window displaying the 'Machine Learning' course page on developers.google.com. The page title is 'Regularization for Simplicity: Playground Exercise (L2 Regularization) - Mozilla Firefox'. The URL is 'https://developers.google.com/machine-learning/crash-course/regularization-for-simplicity/playground-exercise-examining-l2-regularization'. The page has a navigation bar with 'Machine Learning' and 'Courses' tabs. Below the navigation bar, there's a 'Crash Course' section with a list of topics: 'ML Concepts', 'Problem Framing', 'Data Prep', 'Clustering', 'Recommendation', 'Testing and Debugging', and 'GANs'. The 'ML Concepts' section is expanded, showing a list of topics with durations: 'Introduction to ML (3 min)', 'Framing (15 min)', 'Densifying with ML (20 min)', 'Reducing Loss (30 min)', 'First Steps with TF (30 min)', 'Generalization (15 min)', 'Training and Test Sets (25 min)', 'Validation Set (40 min)', 'Representation (35 min)', 'Feature Crosses (70 min)', 'Regularization: Simplicity (40 min)', 'Playground Exercise: Overcrossing?', 'Video Lecture', 'L2 Regularization', 'Lambdas', 'Playground Exercise: L2 Regularization' (highlighted), and 'Check Your Understanding'. Below this, there's a 'ML Engineering' section with topics: 'Production ML Systems (3 min)', 'Static vs. Dynamic Training (7 min)', 'Static vs. Dynamic Inference (7 min)', 'Data Dependencies (14 min)', and 'Fairness (20 min)'. The main content area is titled 'Regularization for Simplicity: Playground Exercise (L2 Regularization)' and has a '☆ ☆ ☆ ☆ ☆' rating. Below the title, there's a blue bar indicating 'Estimated Time: 10 minutes'. The main text describes the exercise: 'This exercise contains a small, noisy training data set. In this kind of setting, overfitting is a real concern. Fortunately, regularization might help. This exercise consists of three related tasks. To simplify comparisons across the three tasks, run each task in a separate tab.' Below this, there are two tasks: 'Task 1: Run the model as given for at least 500 epochs. Note the following: • Test loss. • The delta between Test loss and Training loss. • The learned weights of the features and the feature crosses. (The relative thickness of each line running from FEATURES to OUTPUT represents the learned weight for that feature or feature cross. You can find the exact weight values by hovering over each line.)' and 'Task 2: (Consider doing this Task in a separate tab.) Increase the regularization rate from 0 to 0.3. Then, run the model for at least 500 epochs and find answers to the following questions: • How does the Test loss in Task 2 differ from the Test loss in Task 1? • How does the delta between Test loss and Training loss in Task 2 differ from that of Task 1? • How do the learned weights of each feature and feature cross differ from Task 2 to Task 1? • What do your results say about model complexity?'

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