

Introduction to Deep Learning

Chapter 7: Basic Regularization

Bernd Bischl

Department of Statistics – LMU Munich

Winter term 2021



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).

L2-REGULARIZATION / WEIGHT DECAY

Weight decay can be interpreted **geometrically**. Let us make a

quadratic approximation of the unregularized objective $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ in the neighborhood of its minimizer $\hat{\boldsymbol{\theta}}$,

$$\tilde{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\boldsymbol{\theta}}) + \nabla \mathcal{R}_{\text{emp}}(\hat{\boldsymbol{\theta}}) \cdot (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \mathbf{H}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}),$$

where \mathbf{H} is the Hessian matrix of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ evaluated at $\hat{\boldsymbol{\theta}}$.
Because $\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$,

- the first order term is 0 in the expression above because the gradient is 0, and,
- \mathbf{H} is positive semidefinite.

Source: Goodfellow et al. (2016), ch. 7

L2-REGULARIZATION / WEIGHT DECAY

The minimum of $\tilde{\mathcal{R}}_{\text{emp}}(\theta)$ occurs where $\nabla_{\theta} \tilde{\mathcal{R}}_{\text{emp}}(\theta) = \mathbf{H}(\theta - \hat{\theta})$ is 0. Adding the weight decay gradient $\lambda\theta$, we get the regularized version of

$\tilde{\mathcal{R}}_{\text{emp}}(\theta)$. We solve it for the minimizer $\hat{\theta}_{\text{Ridge}}$:

$$\lambda\theta + \mathbf{H}(\theta - \hat{\theta}) = 0$$

$$(\mathbf{H} + \lambda\mathbf{I})\theta = \mathbf{H}\hat{\theta}$$

$$\hat{\theta}_{\text{Ridge}} = (\mathbf{H} + \lambda\mathbf{I})^{-1} \mathbf{H}\hat{\theta}$$

where \mathbf{I} is the identity matrix. As λ approaches 0, the regularized solution $\hat{\theta}_{\text{Ridge}}$ approaches $\hat{\theta}$. What happens as λ grows?

L2-REGULARIZATION / WEIGHT DECAY

- Because \mathbf{H} is a real symmetric matrix, it can be decomposed as $\mathbf{H} = \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^\top$ where $\mathbf{\Sigma}$ is a diagonal matrix of eigenvalues and \mathbf{Q} is an orthonormal basis of eigenvectors.
- Rewriting the equation on the previous slide using the eigendecomposition above,

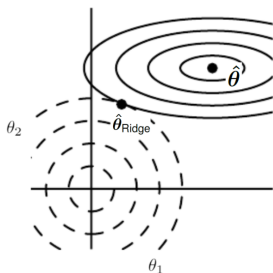
$$\begin{aligned}\hat{\theta}_{\text{Ridge}} &= \left(\mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^\top + \lambda \mathbf{I} \right)^{-1} \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^\top \hat{\theta} \\ &= \left[\mathbf{Q}(\mathbf{\Sigma} + \lambda \mathbf{I})\mathbf{Q}^\top \right]^{-1} \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^\top \hat{\theta} \\ &= \mathbf{Q}(\mathbf{\Sigma} + \lambda \mathbf{I})^{-1} \mathbf{\Sigma}\mathbf{Q}^\top \hat{\theta}\end{aligned}$$

- Therefore, the weight decay rescales $\hat{\theta}$ along the axes defined by the eigenvectors of \mathbf{H} . The component of $\hat{\theta}$ that is aligned with the i -th eigenvector of \mathbf{H} is rescaled by a factor of $\frac{\sigma_i}{\sigma_i + \lambda}$, where σ_i is the corresponding eigenvalue.

L2-REGULARIZATION / WEIGHT DECAY

- Along directions where the eigenvalues of \mathbf{H} are relatively large, for example, where $\sigma_i \gg \lambda$, the effect of regularization is quite small.
- On the other hand, components with $\sigma_i \ll \lambda$ will be shrunk to have nearly zero magnitude.
- In other words, only directions along which the parameters contribute significantly to reducing the objective function are preserved relatively intact.
- In the other directions, a small eigenvalue of the Hessian means that moving in this direction will not significantly increase the gradient. For such unimportant directions, the corresponding components of θ are decayed away.

L2-REGULARIZATION / WEIGHT DECAY

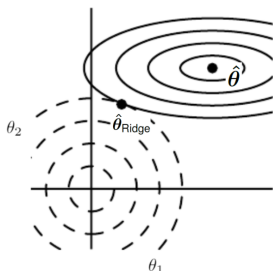


Credit: Goodfellow et al. (2016), ch. 7

Figure: The solid ellipses represent the contours of the unregularized objective and the dashed circles represent the contours of the L2 penalty. At $\hat{\theta}_{\text{Ridge}}$, the competing objectives reach an equilibrium.

In the first dimension, the eigenvalue of the Hessian of $\mathcal{R}_{\text{emp}}(\theta)$ is small. The objective function does not increase much when moving horizontally away from $\hat{\theta}$. Therefore, the regularizer has a strong effect on this axis and θ_1 is pulled close to zero.

L2-REGULARIZATION / WEIGHT DECAY



Credit: Goodfellow et al. (2016), ch. 7

Figure: The solid ellipses represent the contours of the unregularized objective and the dashed circles represent the contours of the L2 penalty. At $\hat{\theta}_{\text{Ridge}}$, the competing objectives reach an equilibrium.

In the second dimension, the corresponding eigenvalue is large indicating high curvature. The objective function is very sensitive to movement along this axis and, as a result, the position of θ_2 is less affected by the regularization.

L1-REGULARIZATION

- The L1-regularized risk of a model $f(\mathbf{x} \mid \boldsymbol{\theta})$ is

$$\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

and the (sub-)gradient is:

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

- Note that, unlike in the case of L2, the contribution of the L1 penalty to the gradient doesn't scale linearly with each θ_i . Instead, it is a constant factor with a sign equal to $\text{sign}(\theta_i)$.
- Let us now make a quadratic approximation of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$. To get a clean algebraic expression, we assume the Hessian of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ is diagonal, i.e. $\mathbf{H} = \text{diag}([H_{1,1}, \dots, H_{n,n}])$, where each $H_{i,i} > 0$.
- This assumption holds, for example, if the input features for a linear regression task have been decorrelated using PCA.

L1-REGULARIZATION

- The quadratic approximation of $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ decomposes into a sum over the parameters:

$$\tilde{\mathcal{R}}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{R}_{\text{emp}}(\hat{\boldsymbol{\theta}}) + \sum_i \left[\frac{1}{2} H_{i,i} (\theta_i - \hat{\theta}_i)^2 \right] + \sum_i \lambda |\theta_i|$$

where $\hat{\boldsymbol{\theta}}$ is the minimizer of the unregularized risk $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$.

- The problem of minimizing this approximate cost function has an analytical solution (for each dimension i), with the following form:

$$\hat{\theta}_{\text{Lasso},i} = \text{sign}(\hat{\theta}_i) \max \left\{ |\hat{\theta}_i| - \frac{\lambda}{H_{i,i}}, 0 \right\}$$

- If $0 < \hat{\theta}_i \leq \frac{\lambda}{H_{i,i}}$, the optimal value of θ_i (for the regularized risk) is 0 because the contribution of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$ to $\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$ is overwhelmed by the L1 penalty, which forces it to be 0.

L1-REGULARIZATION

- If $0 < \frac{\lambda}{H_{i,i}} < \hat{\theta}_i$, the L1 penalty shifts the optimal value of θ_i toward 0 by the amount $\frac{\lambda}{H_{i,i}}$.
- A similar argument applies when $\hat{\theta}_i < 0$.
- Therefore, the L1 penalty induces sparsity in the parameter vector.

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

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

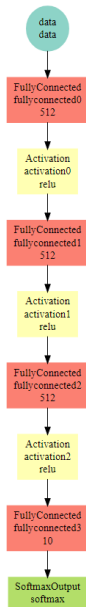
$$\text{subject to } J(\theta) \leq k$$

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

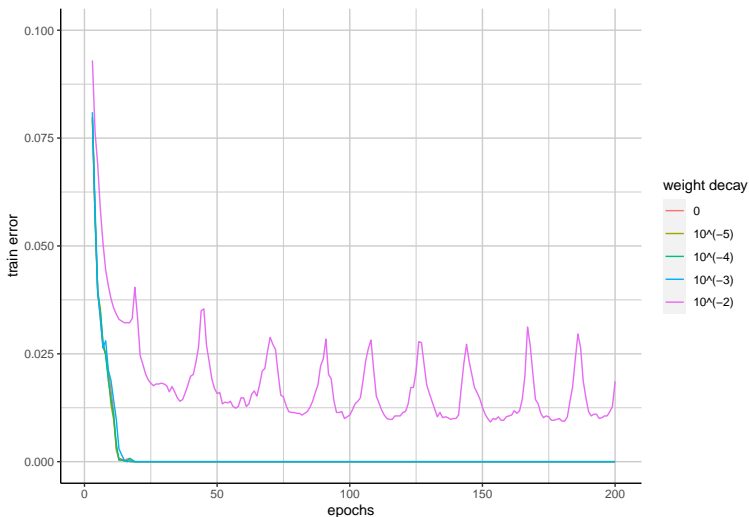
(Goodfellow et al. (2016), ch. 7.2)

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

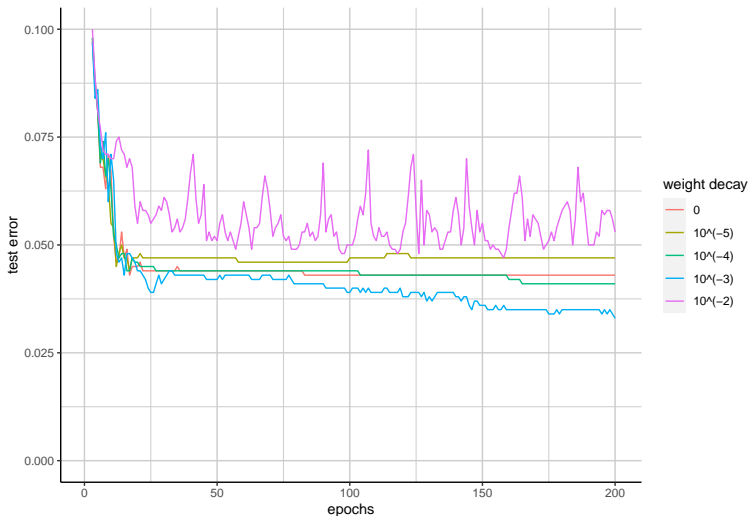


EXAMPLE: WEIGHT DECAY



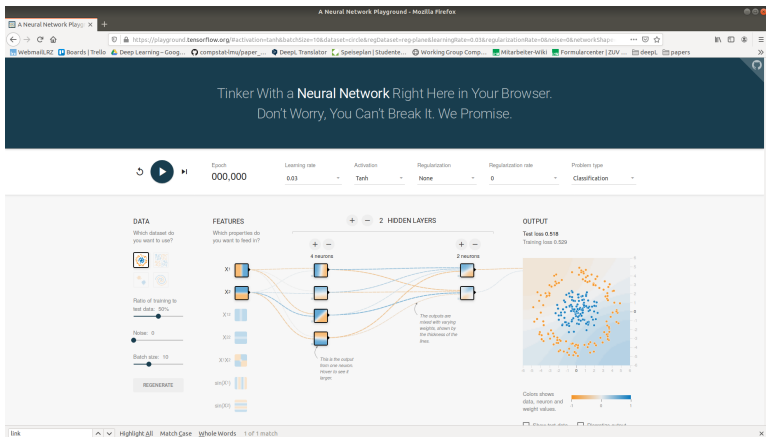
A high weight decay of 10^{-2} leads to a high error on the training data.

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 displaying the 'Regularization for Simplicity: Playground Exercise (L2 Regularization)' page. The page is part of the 'Machine Learning' crash course. The left sidebar contains a navigation menu with categories like 'ML Concepts', 'ML Engineering', and 'ML Systems in the Real World'. The main content area is titled 'Regularization for Simplicity: Playground Exercise (L2 Regularization)' and includes an 'Estimated Time: 10 minutes' badge. The text explains that the exercise contains a small, noisy training data set and that regularization might help. It outlines three tasks: Task 1 (Run the model for at least 500 epochs, note test loss and delta between test and training loss), Task 2 (Increase regularization rate from 0 to 0.3, then run the model for at least 500 epochs and answer questions about test loss, delta, and learned weights), and Task 3 (What do your results say about model complexity?).

Regularization for Simplicity: Playground Exercise (L2 Regularization)

Estimated Time: 10 minutes

Examining L_2 regularization

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.

- **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.)
- **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>