Regularization

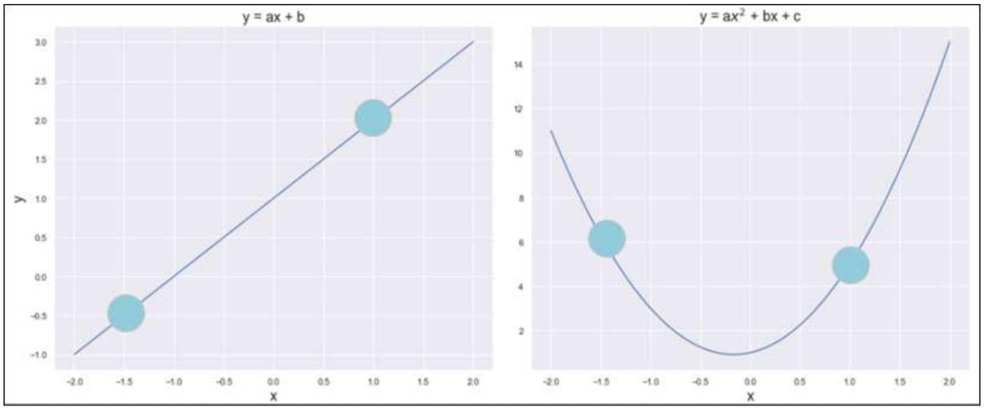
Friday, August 5, 2022 6:21 AM

A regularizer is a penalty added to the cost function to impose an extra condition

$$L_R(X,Y;\bar{\theta}) = L(X,Y;\bar{\theta}) + \lambda g(\bar{\theta})$$

A fundamental condition on g(theta) is that it must be differentiable so that the new composite cost function can still be optimized using SGD algorithms.

we normally need a function that can contrast the indefinite growth of the parameters.



Interpolation with a linear curve (left) and a parabolic one (right)

In the first diagram: the model is linear and has two parameters
In the second diagram: the model is quadratic and has three parameters.

The second option is more prone to overfitting. But if we apply regularization it's possible to avoid overfitting and transform the model into a linearized version.

with regularization we keep the same model but optimize it to reduce the variance and only slightly increase the bias.

we can say that regularization often acts as a brake that avoids perfect convergence. it keeps the model in a region where the generalization error is lower

A small bias can be acceptable when it's the consequence of a drastic variance reduction (that is, a smaller generalization error);

SUGGESTION:
Don’t employ regularization as a black-box technique, but instead check using cross-validation which value yields the optimal result.

Examples of regularization techniques:

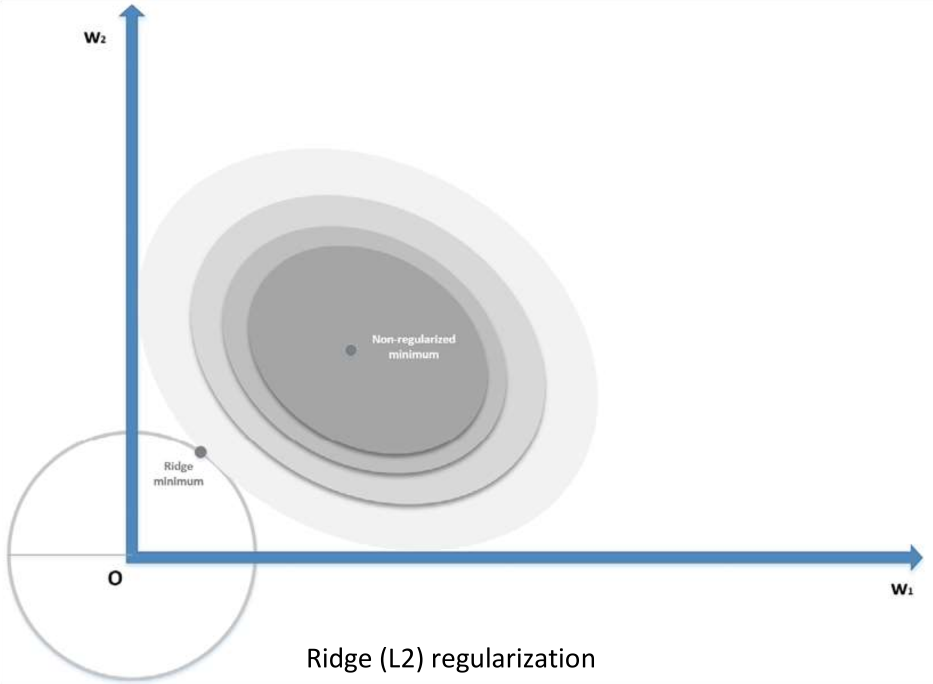
<u>L2 or Ridge regularization:</u>

L2 or ridge regularization also known as Tikhonov regularization is based on the Squared L2-norm of the parameter vector:

$$L_R(X,Y;\bar{\theta}) = L(X,Y;\bar{\theta}) + \lambda\left\|\bar{\theta}\right\|_2^2$$

This penalty avoids infinite growth of the parameters—for this reason, it's also known as weight shrinkage it's useful when the model is ill-conditioned or there is multicollinearity due to the fact that the samples are not completely independent.

we see a schematic representation of the Ridge regularization in a bidimensional scenario:



Ridge (L2) regularization

The zero-centered circle represents the Ridge boundary,

the shaded surface is the original cost function.

Without regularization, the minimum (w1, w2) has a magnitude (for example, the distance from the origin) which is about double the one obtained by applying a Ridge constraint, confirming the expected shrinkage.

When applied to regressions solved with the Ordinary Least Squares (OLS) algorithm, it's possible to prove that there always exists a Ridge Coefficient.

regularization, applied to the majority of classification algorithms, allows us to obtain rotational invariance. In other words, if the training set is rotated, a regularized model will yield the same prediction distribution as the original one.

L2-regularization

shrinks the weights independent of the scale of the data. Therefore, if the features have different scales, the result may be worse than expected.

considering a simple linear model with two variables, and L2 has a single control coefficient,
y = ax1 + bx1 + c.
effect will be the same on both a and b (excluding the intercept c).
If x1 [-1,1] and x2 [0,100]
the shrinkage will affect x1 much more than x2.

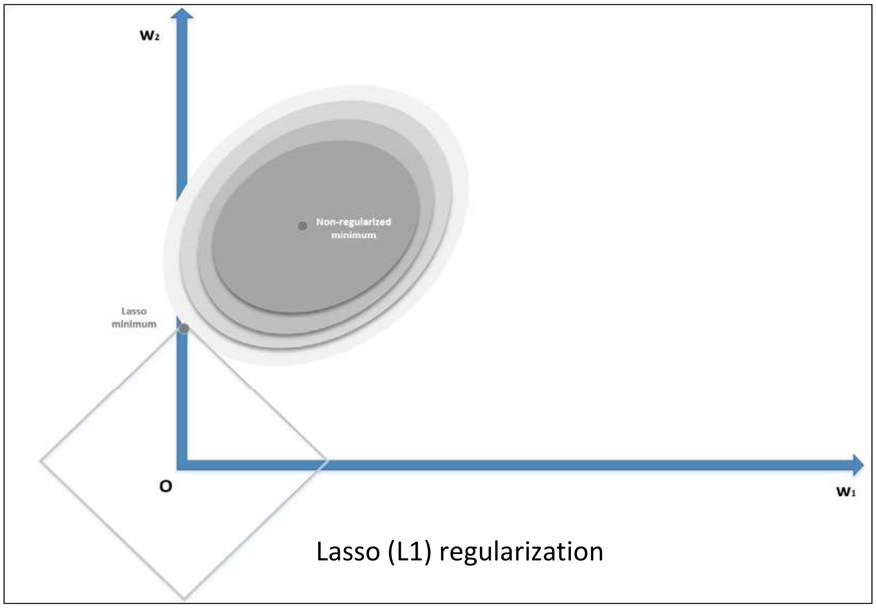
RECOMMENDATION:
Scaling the dataset before applying the dataset.

L1 or Lasso regularization

L1 or Lasso regularization is based on the L1-norm of the parameter vector:

$$L_R(X,Y;\bar{\theta}) = L(X,Y;\bar{\theta}) + \lambda\|\bar{\theta}\|_1$$

While Ridge shrinks all the weights inversely proportionally to their importance, Lasso can shift the smallest weight to zero. Creating a sparse parameter vector.



Lasso (L1) regularization

The zero-centered square represents the Lasso boundaries in a bidimensional scenario

If we consider a generic line, the probability of that line being tangential to the square is higher at the corners, where at least one parameter is null—exactly one parameter in a bidimensional scenario. In general, if we have a vectorial convex function f(x).

$$g(\bar{x}) = f(\bar{x}) + \|\bar{x}\|_p$$

As any Lp-norm is convex, as well as the sum of convex functions g(x), is
also convex.

The regularization term is always non-negative, and therefore the minimum
corresponds to the norm of the null vector.

When minimizing g(x) we also need to the contribution of the gradient of
The norm in the ball centered in the origin where the partial derivatives don't
Exist

Increasing the value of p, the norm becomes
smoothed around the origin, and the partial derivatives approach zero for
Lim(x) --> 0

excluding the L0-norm and all the norms with p[0,1] allows an even stronger
Sparsity but is non-convex (even if L0 is employed in the quantum algorithm QBoost.)
with p = 1 the partial derivatives are always +1 or -1, according to the sign of Xi (Xi <>0)
it's easier for the L1-norm to push the smallest components to zero, because the contribution
to the minimization (for example, with gradient descent) is independent of xi,
while an L2-norm decreases its speed when approaching the origin.

Lasso regularization. when a sparse representation of a dataset is needed.
we could be interested in finding the feature vectors corresponding to a group of images.
subset of those features present in each image, applying
the Lasso regularization allows us to force all the smallest coefficients to
become null, which helps us by suppressing the presence of the secondary
features.

potential application is latent semantic analysis,
our goal is to describe the documents belonging to a corpus in terms of a limited
number of topics those techniques can be called sparse coding.

where the objective is to reduce the dimensionality of dataset by extracting the
most representative atoms, using different approaches to achieve sparsity.

One of the important features of L1 is the ability to perform an implicit feature
Selection induced by sparsity.

a dataset contains n features, the minimum number
of samples required to increase the accuracy over a predefined threshold is
affected by the logarithm of the number of redundant or irrelevant features.

How to use:
it's automatic and no preprocessing steps are required. This is extremely
useful in deep learning.

if dataset X contains 1000 points and the optimal accuracy is achieved with this
sample size when all the features are informative when k < p features are irrelevant,
features are irrelevant, we need approximately 1000 + O(log k) samples.
This is a simplification of the original result; for example, if p = 5000 and 500 features

are irrelevant, assuming the simplest case, we need about 1000 +log(500)=1007 data points.

A result is very important as it's difficult and expensive to obtain a large number
Of new samples

let's consider a synthetic dataset
containing 500 points xi belongs to R(10) with only five informative features:
See L1 versus L2 notebook

RECOMMENDATION:
when working with linear models, perform a feature selection to remove all
non-determinant factors; L1 regularization is an excellent choice that avoids
an additional preprocessing step.

ELASTICNET:

In some time it's useful to apply both Ridge and Lasso regularization to force wight shrinkage
And global sparsity it's possible by employing ElasticNet regularization defined as:

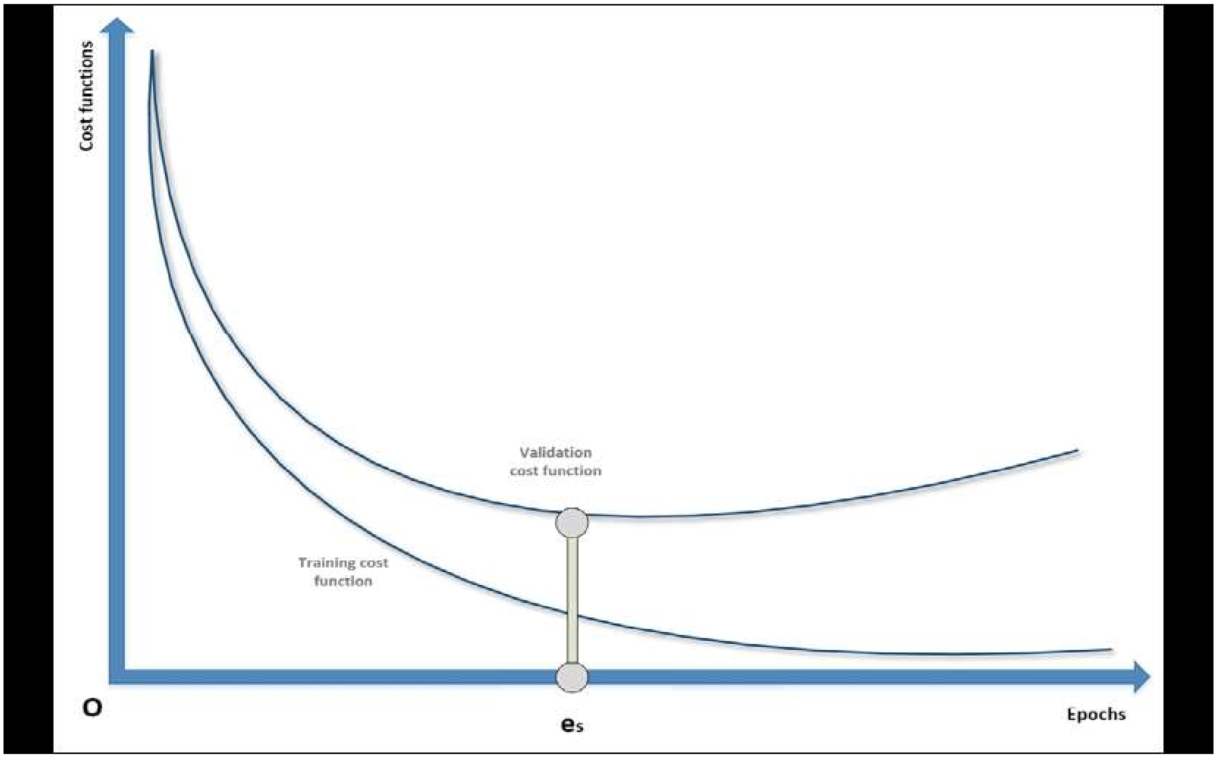
$$L_R(X,Y;\bar{\theta}) = L(X,Y;\bar{\theta}) + \lambda_1\|\bar{\theta}\|_2^2 + \lambda_2\|\bar{\theta}\|_1$$

The strength of each regularization is controlled by the parameters and
. ElasticNet can yield excellent results whenever it's necessary to mitigate
overfitting effects while encouraging sparsity

EARLY STOPPING:

Early stopping is often considered as a last resort when all approaches to prevent overfitting,
And maximum validation accuracy fail.

It happens in many deep learning scenarios and SVMs and other simpler classifiers
It's possible to observe a typical behavior of the training process, considering both
training and the validation cost functions:



Example of early stopping before the beginning of the ascending phase of a U-curve

During the first epochs, both costs decrease, but it can happen that after a threshold epoch es, the validation cost starts increasing. If we continue with the training process, this results in overfitting the training set and increasing the variance.

there are no other options, it's possible to prematurely stop the training process. In order to do so, it's necessary to store the last parameter vector before the beginning of a new iteration and, in the case of no improvements or the accuracy worsening, to stop the process and recover the last set of parameters.

This option should never be considered as the best choice, because a better model or an improved Dataset could yield higher performances
It should be adopted at the last stage of the process and never at the beginning

Many deep learning frameworks such as keras include helpers to implement early stopping callback. However its important to check the minimum validation cost when it has been achieved.

Summary
In this chapter, we introduced the loss and cost functions, first as proxies of the expected risk, and then we detailed some common situations that can be experienced during an optimization problem. We also exposed some common cost functions, together with their main features and specific applications.
In the last part, we discussed regularization, explaining how it can mitigate

the effects of overfitting and induce sparsity. In particular, the employment of Lasso can help the data scientist to perform automatic feature selection by forcing all secondary coefficients to become equal to 0.

In the next chapter, Chapter 3, Introduction to Semi-Supervised Learning, we're going to introduce semi-supervised learning, focusing our attention on the concepts of transductive and inductive learning.