Imperial College London

May 3rd, 2019

Regularization, Bias and Variance

Olivier Dubrule and Lukas Mosser

Imperial College London

Objectives of the Day

- Underfitting and Overfitting are also called Bias and Variance
- Regularization is presented, which may help control Variance/Overfitting
- We need techniques - or Machine Learning Diagnostics - for optimizing the choice of Network Architecture and Hyperparameters
- Training set, Validation Set and Test set play a key role in Machine Learning Diagnostics
- Example of k-Fold Validation for running Machine Learning Diagnostics

Imperial College London

Regularization, Bias and Variance

1. Overfitting and Underfitting, Bias and Variance
2. Regularization
3. The Need for Machine Learning Diagnostics
4. Training Set, Validation Test and Test Set
5. K-Fold Validation

Imperial College London

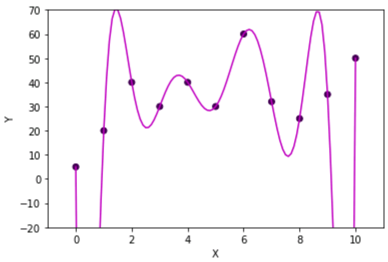
The Training Set Error is not an indicator of how well the fitted model is going to work on other data

If we have a large number of fitting parameters, the trained model may fit the training set very well:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \approx 0$$

But fail to generalize to new data! This is called *Overfitting*.

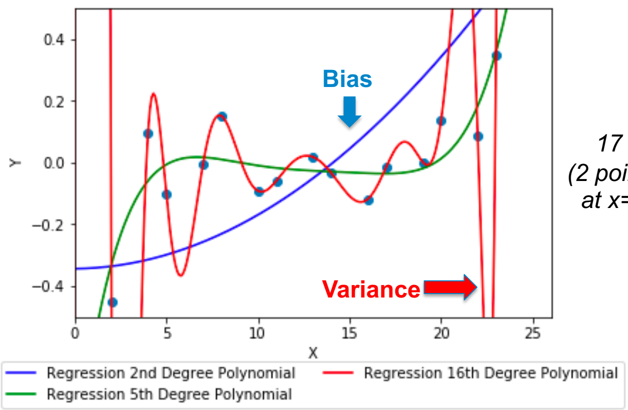
Example of Polynomial Overfitting



11 data points, 10th degree polynomial with bias

Imperial College London

Bias (or Underfitting) vs Variance (or Overfitting)



17 data points (2 points not shown at x=0 and x=26)

Imperial College London

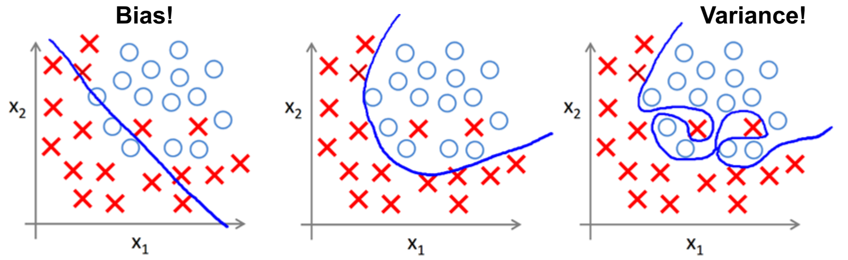
Why the terms « Bias » and « Variance »?

Bias: *the hypothesis function $h_\theta(x)$ has a « pre-conception » or an « a priori » idea of the data variations which is too simple, which is biased from the start, considering the actual variability of the data.*

Variance: *the hypothesis function $h_\theta(x)$ has too many degrees of freedom – or parameters - and, as a result, can fit too many possible functions, with too much variance, considering the actual variability of the data.*

Imperial College
London

Configurations for Non-Linear Logistic Regression



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
(g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

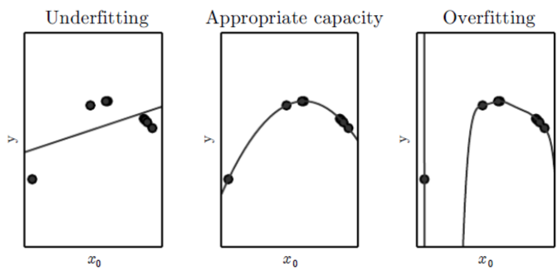
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

Source: Machne Learning Course, Andrew Ng

Imperial College
London

Capacity of a Model

*The **Capacity** of a model is associated with its likely underfitting or overfitting. Informally a model's capacity is its ability to fit a wide variety of functions. Models with low capacity may tend to **underfit (Bias) because they do not have enough parameters** , models with high capacity may tend to **overfit (Variance) because they have too many parameters.***



Goodfellow et al, 2017

Imperial College London

The Generalization Error

The trained model must perform well on new, previously unseen data, not just those on which the model was trained. The ability to perform well on previously unseen data is called ***Generalization.***

Goodfellow et al, 2017

Imperial College London

Underfitting and Overfitting

A good Machine Learning algorithm must:

1. Make the Training Error small. If this is not the case, we have <u>Underfitting</u>.
2. Make the gap between Training and Test Error small. If this is any not the case we have <u>Overfitting.</u>

Goodfellow et al, 2017

Imperial College London

Regularization, Bias and Variance

1. Overfitting and Underfitting, Bias and Variance
2. Regularization
3. The Need for Machine Learning Diagnostics
4. Training Set, Validation Test and Test Set
5. K-Fold Validation

Imperial College London

One Way to Avoid Overfitting: Regularization

Regularization is any modification we make to a learning algorithm that is intended to reduce its Generalization Error but not its Training Error…

An effective regularizer is one that makes a profitable trade, reducing Variance significantly while not increasing the Bias.

Goodfellow et al, 2017

Imperial College London

(L1 or L2) Regularization

L1 and L2 Regularizations consist of adding a new term to the objective function in order to control the variations of the parameters:

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right)$$ if L2 norm is used

Regularization Parameter, controlling the "Weight Decay"

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} |\theta_i|\right)$$ if L1 norm is used

Imperial College London

Ridge and Lasso Regressions

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x_2 + \ldots + \theta_n x_n$$

Ridge Regression (preferred because math derivations simpler)

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right)$$ if L2 norm is used

Lasso Regression

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} |\theta_i|\right)$$ if L1 norm is used

Imperial College London

Logistic Regression with Regularization: Solution

Take the Case of Binary Linear Logistic Regression in 2-D.

Write $h_\theta(x)$ *as a function of* x *and* θ. $h_\theta(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \theta_2 x_2)}}$

Assume there are m data points $(x^{(i)}, y^{(i)})$ *for* $i = 1, \ldots, m$

Express $J(\theta)$ *using L2 regularization and a regularization parameter* λ.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)})\log\left(1 - h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}(\theta_0^2 + \theta_1^2 + \theta_2^2)$$

Calculate $\frac{\partial J(\theta)}{\partial \theta_j}$ *for one of the* θ_j *parameters:* $\frac{\partial J(\theta)}{\partial \theta_2} = \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x_2^{(i)} + \frac{\lambda}{m}\theta_2$

Imperial College London

Gradient Descent for Regularized Regression

Gradient Descent Approach without Regularization Term (ie Logistic Regression)

$$\theta_j := \theta_j - \alpha\frac{\partial J}{\partial \theta_j} := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\ )x_j^{(i)}\right)$$

Gradient Descent with Regularization Term (if L2 norm is used) (Logistic Regression)

$$\theta_j := \theta_j - \alpha\frac{\partial J}{\partial \theta_j} := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\ )x_j^{(i)}\right) - \alpha\frac{\lambda}{m}\theta_j$$

$$\theta_j := \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\ )x_j^{(i)}\right)$$

Imperial College London

Gradient Descent with Regularized Regression

Consider in more detail the Gradient Descent term in case of Regularization:

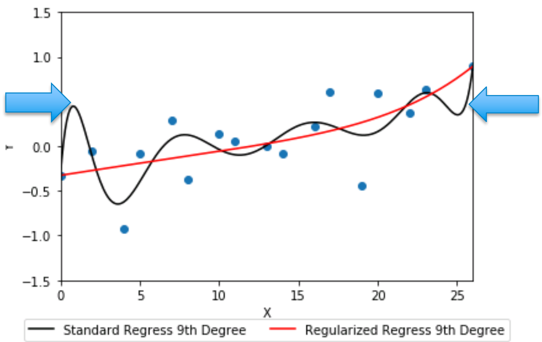
$$\theta_j := \theta_j\left(1-\alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)x_j^{(i)}$$

Systematic decrease of absolute value θ_j at each iteration

Same term as for optimization without regularization

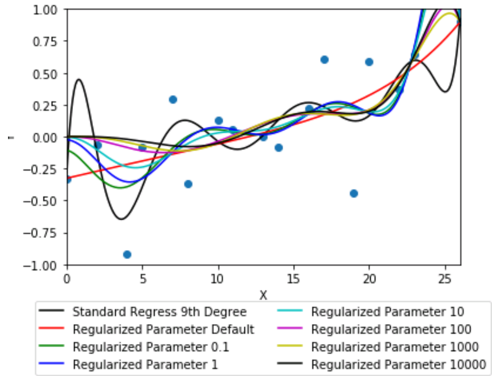
Imperial College London

The Impact of Applying Regularization to Regression



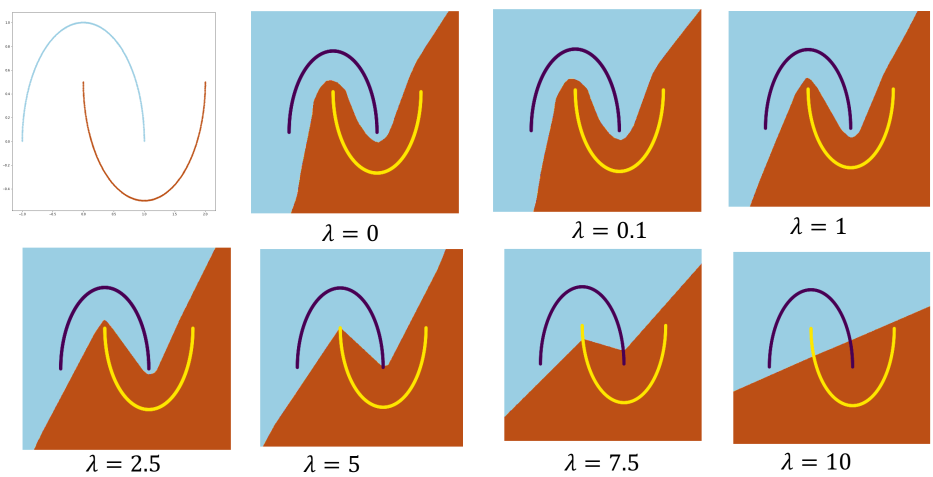
Imperial College
London

Regression: Changing the Regularization Parameter



Imperial College
London

A Simple Neural Network Regularization Example (1)

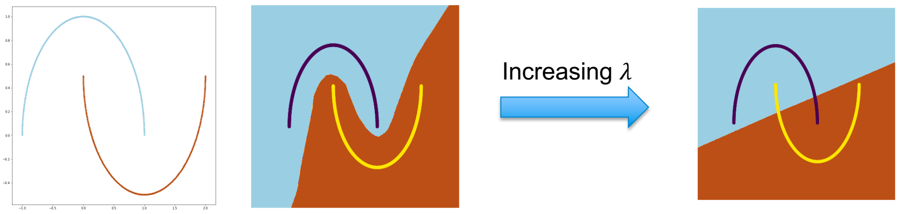


$\lambda = 0$ $\lambda = 0.1$ $\lambda = 1$

$\lambda = 2.5$ $\lambda = 5$ $\lambda = 7.5$ $\lambda = 10$

Imperial College London

A Simple Neural Network Regularization Example (2)



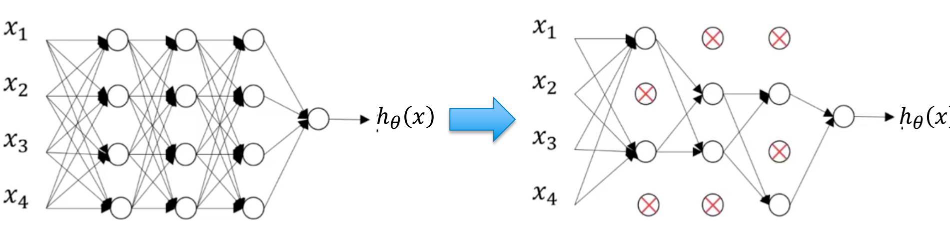
$\lambda = 0$ $\lambda = 10$

As λ increases, the values of the weights tend to zero, and the hypothesis function is close to a linear approximation.

Imperial College London

Another Popular Regularization Approach: Drop-Out

At Training time:



At each optimization iteration… Drop hidden layers units with 0.5 probability…

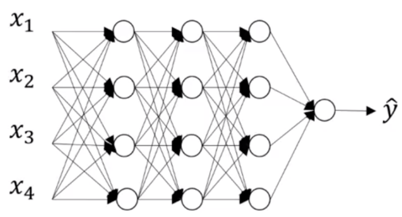
We are sampling over a set of network configurations!

From Andrew Ng, CL2W1L06

Imperial College London

Another Popular Regularization Approach: Drop-Out

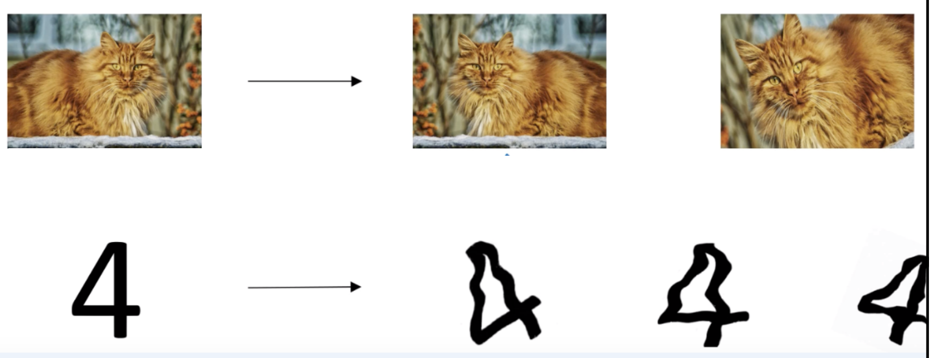
At Test time: do not use drop-out



From Andrew Ng, CL2W1L06

Imperial College London

Another Regularization Technique: Data Augmentation



From Andrew Ng, CL2W1L08

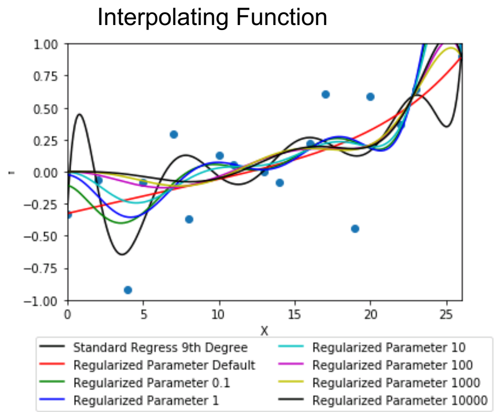
Imperial College London

Regularization, Bias and Variance

1. Overfitting and Underfitting, Bias and Variance
2. Regularization
3. The Need for Machine Learning Diagnostics
4. Training Set, Validation Test and Test Set
5. K-Fold Validation

Imperial College London

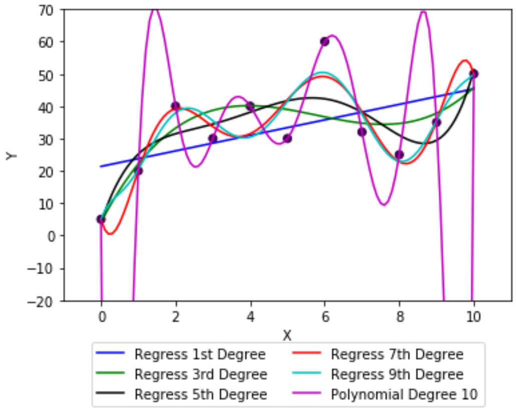
Changing the Regularization Parameter



How can we choose the "Best" Regularization Parameter?

Imperial College London

Change of Interpolating Function as Degree Increases

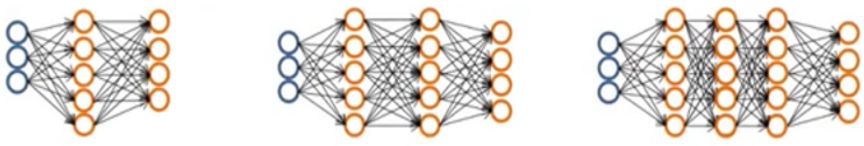


How can we choose the “Best” Polynomial Degree?

Imperial College London

Change of Network Architecture

From one to three Hidden Layers



How can we choose the “Best” Network Architecture?

Imperial College London

Machine Learning Diagnostic

Diagnostic: A test to gain insight about the performance of a learning algorithm, and find clues about how to improve this performance.

Imperial College London

Regularization, Bias and Variance

1. Overfitting and Underfitting, Bias and Variance
2. Regularization
3. The Need for Machine Learning Diagnostics
4. Training Set, Validation Test and Test Set
5. K-Fold Validation

Imperial College
London

What is a Hyperparameter?

A **hyperparameter** is a neural network parameter whose value is set before the Training process begins. It does not change during Training. By contrast, the values of parameters θ are derived via Training.

Exercise: Give Examples of Hyper-Parameters of a Neural Network

Imperial College
London

Training Set vs Test Set

In a Supervised Learning context, the Machine Learning algorithm is trained on the Training Set, but tested on the Test Set. The Test Set constitutes the unseen data, it should not be used to calculate the parameters of the Neural Network, or to choose the Hyperparameters.

Example of MNIST: *the Training Set contains 60,000 images, the « official » Test Set contains 10,000 images. This ratio is usually appropriate for this size of datasets. For very large datasets (say 100.000's to millions), split between Training and Test Set sizes can be smaller and be as low as 90%-10%.*

Imperial College London

The Need for a Validation Set

We use the Training Set to optimize the Neural Network Parameters or Weights.

How to choose the best set of Hyperparameters?

We cannot use the Training Set, as our goal is the minimization of the Generalization Error.

We cannot use the Test Set, which should be used only at the end to test performance on unseen data.

We need a third Set, the Validation Set!

Imperial College London

Create Validation Set to Optimize Hyperparameters

Split Data Into Three Sets:
Training Set $(\sim 70\%)$, *Validation Set* $(\sim 15\%)$, *Test Set* $(\sim 15\%)$

Training Set	***Validation Set***	***Test Set***

Imperial College London

Use Validation Set to Optimize Hyperparameters

For each Possible Neural Network Architecture or Hyperparameters:

1. Train Neural Network on Training Set
2. Test its Performance on the Validation Set
3. Pick the Architecture and Hyperparameters that gives the best performance on the Validation Set.

The Test Set is then used as the final measure of performance but is never used in the Training. It is often unknown to the Neural Network developer.

Imperial College London

Warning

Too many people still use the Test Set to optimize the Hyper-Parameters.

This means that the ultimate Test Set Error is not representative of the Generalization Error!

Imperial College London

Sampling the Validation Set and possibly Test Set

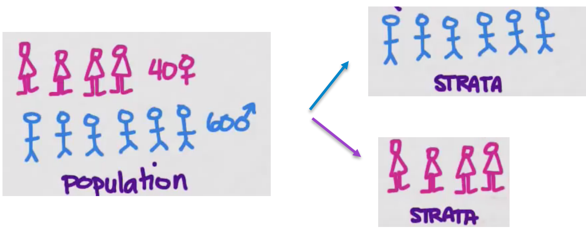
In Classification problems, make sure the proportions of each class are the same for Training, Validation and possibly Test Sets.

For smaller data sets, use Stratified Random Sampling (class by class) instead of global sampling over the whole set.

Imperial College London

Stratified Random Sampling: *Create a Validation Set of 30*

Sort Data into two "Strata"



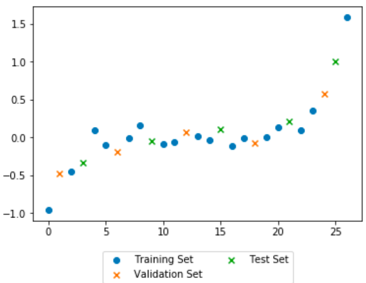
Randomly extract 30%

Randomly extract 30%

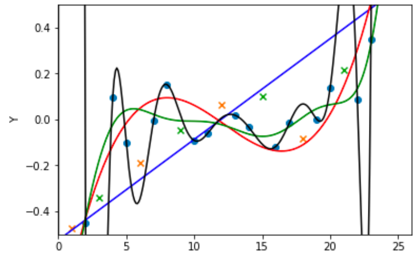
First sort the population into strata, then sample!

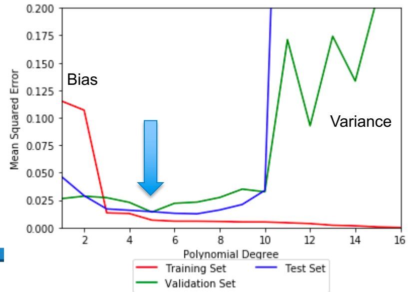
Imperial College London

Example of Different Sets for Polynomial Fitting



17 Training data
5 Validation data
5 Test Data





Imperial College London

$J_{train}(\theta)$ and $J_{val}(\theta)$ for picking the Regularization Parameter

The Training Set has m data points. The optimized Loss Function is (if L2 norm):

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right)$$

For each tested value of λ, train the network on the Training Set, and evaluate:

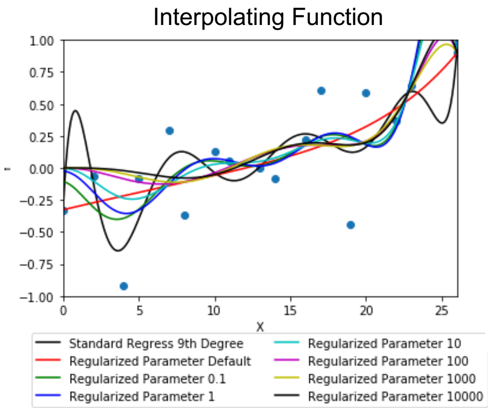
$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

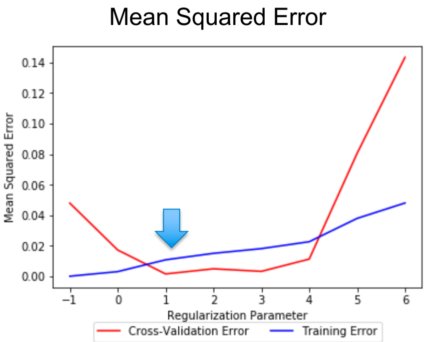
And on the Validation Set: $$J_{val}(\theta) = \frac{1}{2m_{val}}\sum_{i=1}^{m_{val}}\left(h_\theta\left(x_{val}^{(i)}\right) - y_{val}^{(i)}\right)^2$$

On the Test Set: $$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h_\theta\left(x_{test}^{(i)}\right) - y_{test}^{(i)}\right)^2$$

Imperial College London

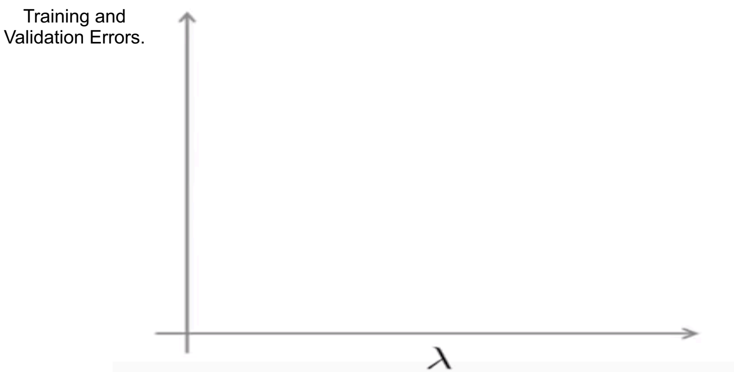
Changing the Regularization Parameter





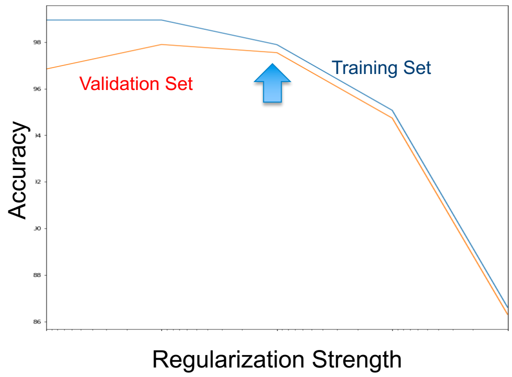
Imperial College London

Typical Behaviour Associated with Regularization



Imperial College London

From Tuesday: Regularization Parameter Optimization



Imperial College London

The Role of the Validation Set: Back to MNIST

The “official” MNIST Training Set is 60000 data. The “official” Test Set is 10000. But the Test Set is not known.

So the user needs to split the official Training Set into a new Training and Validation sets, for example with 50000 and 10000 data points in each.

Then for a number of possible Network Architectures or Hyperparameters:

1. Train Neural Network on Training Set
2. Test its Performance on Validation Set

Then pick the Architecture or Hyperparameters that gives the optimal performance on the Validation Set.

Then finally evaluate the performance of the Neural Network associated with these optimal Hyperparameters or Architectures using the Test Set.

Imperial College London

Split between Test and Validation Set on MNIST

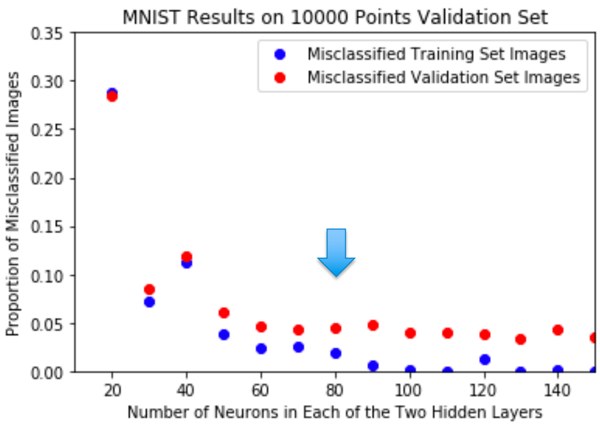
The number of labelled images in the Data Set is the same for each label.

There are 60,000 images.

Because of this large number, Random Shuffling is enough. No need for Stratified Random Sampling.

Imperial College London

MNIST: Optimizing Number of Neurons in Layers

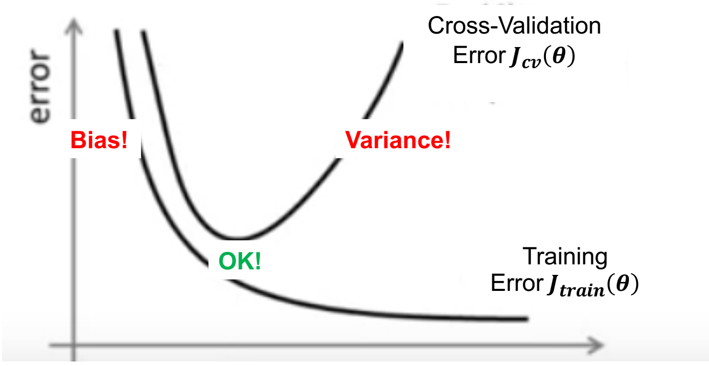


Optimal value seems to be 70 neurons in each of the two hidden layers.

This hyperparameter value gives a proportion of 0.04 misclassified images in the Test Set (same as Validation Set!)

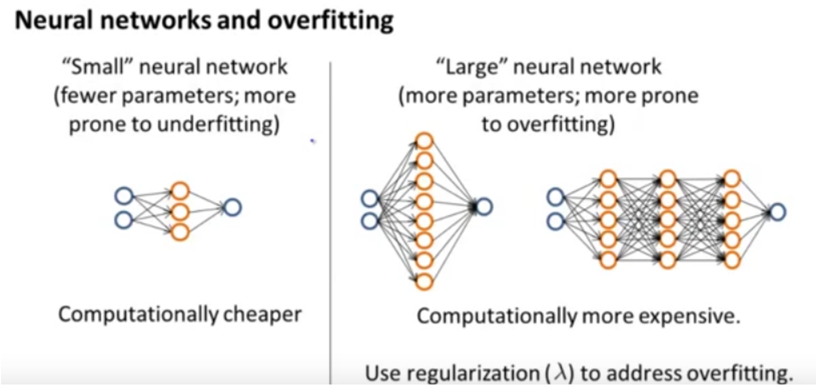
Imperial College London

How to Identify Bias vs Variance Problems



Imperial College London

Bias/Underfitting and Variance/Overfitting in NNs



Andrew NG 10.7

Imperial College
London

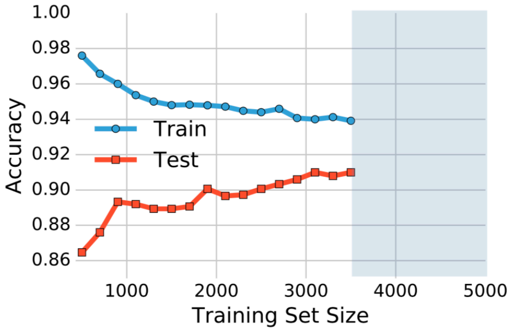
General Guidelines for Improving Training

To address Bias problems
Try using more input features
Try decreasing the regularization parameter

To address Variance problems
Try decreasing the number of features
Try increasing the regularization parameter

Imperial College
London

MNIST : Impact of Number of Training Data



Imperial College London

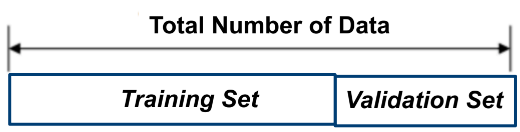
Regularization, Bias and Variance

1. Overfitting and Underfitting, Bias and Variance
2. Regularization
3. The Need for Machine Learning Diagnostics
4. Training Set, Validation Test and Test Set
5. K-Fold Validation

Imperial College London

Limitation of using just one Validation Set (Hold-Out Method)

If Data Set is not very big, the tole played by Training and Validation sets is not symmetrical. The approach used for the split may affect the results.

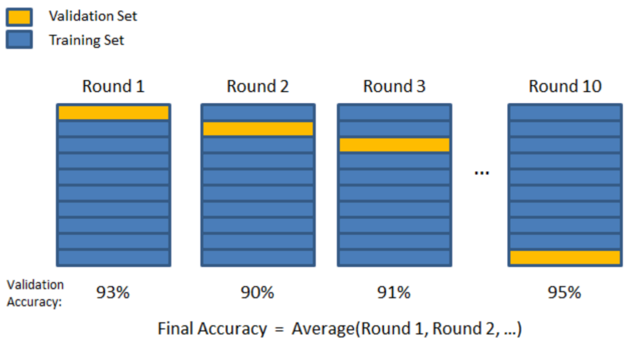


Possible approach: permutate between Validation and Training Sets and recalculate. Can do it if size of both is the same.

https://towardsdatascience.com/cross-validation-70289113a072

Imperial College London

What is k-Fold Validation?



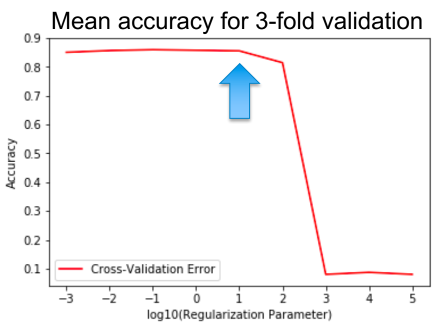
https://towardsdatascience.com/cross-validation-70289113a072

Imperial College London

3-Fold Validation on MNIST Example

1	2	3
Validation	Train	Train

For each value of the Regularization Parameter, a 3-fold validation is run: three validation sets are created in turn and predicted using the rest of the data as training set.



Imperial College London

Afternoon Exercise

Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning

Sebastian Raschka
University of Wisconsin–Madison
Department of Statistics
November 2018
sraschka@wisc.edu

Abstract

The correct use of model evaluation, model selection, and algorithm selection techniques is vital in academic machine learning research as well as in many industrial settings. This article reviews different techniques that can be used for each of these three subtasks and discusses the main advantages and disadvantages of each technique with references to theoretical and empirical studies. Further, recommendations are given to encourage best yet feasible practices in research and applications of machine learning. Common methods such as the holdout method for model evaluation and selection are covered, which are not recommended when working with small datasets. Different flavors of the bootstrap technique are introduced for estimating the uncertainty of performance estimates, as an alternative to confidence intervals via normal approximation if bootstrapping is computationally feasible. Common cross-validation techniques such as leave-one-out cross-validation and k-fold cross-validation are reviewed, the bias-variance trade-off for choosing k is discussed, and practical tips for the optimal choice of k are given based on empirical evidence. Different statistical tests for algorithm comparisons are presented, and strategies for dealing with multiple comparisons such as omnibus tests and multiple-comparison corrections are discussed. Finally, alternative methods for algorithm selection, such as the combined F-test 5x2 cross-validation and nested cross-validation, are recommended for comparing machine learning algorithms when datasets are small.

Imperial College London

Summary for Regularization, Bias and Variance

- **Bias and Variance to characterize Underfitting versus Overfitting.**
- **L1 or L2 Regularizations as a cure against Overfitting.**
- **Drop-out and Data Augmentation are other techniques for Regularization.**
- **Importance of Training, Validation and Test Set for Diagnostics.**
- **K-Fold Validation for Hyper-Parameters Optimization.**