VsiTe College of Information Technology

Workshop

Deep Learning - A Basic Tool of Artificial Intelligence

Cross Validation
for
Model Design
i.e., for
Machine Learning Model Hyperparameters Selection

CROSS VALIDATION (CV)

- **Few facts about machine learning , a.k.a data mining, a.k.a learning from data, a.k.a statistical learning**
- *a*) **noisy and sparse data points are the only information about the reality**
- *b*) **no idea what is the true (target) function**
- *c*) **modern modeling tools can perfectly model the existing (training) data points but**
- *d*) **the question is what will be a performance of the model on the new, i.e., previously unseen, data points,**
- *e*) **this is the question of a GENERALIZATION**

  because only
- *f*) **the MODEL THAT GENERALIZES WELL, IS A GOOD MODEL**

The most standard method for **designing the model**, i.e., for **selecting the best hyperparameters** of the model, i.e., for **choosing the right complexity of the model** which will generalize well is the

CROSS-VALIDATION (CV) a.k.a. **RESAMPLING**

a) on some test dataset, or

b) by the so-called leave-one-out (LOO) approach, or

c) by *k*-fold cross-validation,

Each of which is the widely used method today, for trading-off the BIAS - VARIANCE dilemma,

in both the classic statistics and in the novel, more or less, engineering approaches.

Note! There are other methods too!

WHAT ARE THE ALTERNATIVE METHODS?

There are several tools for model design. We mention here three the most popular ones:

1) Cross-validation (CV)

2) Akaike information criterion (AIC)

Asymptotically, AIC and the leave-one-out CV should be the same.

3) Bayesian information criterion (BIC)

Asymptotically, BIC and the correctly chosen K-fold CV should be the same.

SRM is a very conservative method. Confidence term is an order of magnitude larger than the empirical overfitting effect!

What criterion to use is primarily conformity (emotional) issue!

Different Models **Comparisons**

Three methods from previous slides are used for

- **designing the model** i.e.,

- **for a selection of hyperparameters of the model**

but

If one wants to COMPARE performances of different models (say, for example, NN, linear classifier with sum-of-error-squares cost function, RBF network, k-nearest neighbors classifier and SVM) one MUST use

DOUBLE (i.e., NESTED) CV a.k.a.
DOUBLE RESAMPLING

The CV approach is an **emergency solution**. It **compensate for deficiency** of an induction principle that **cannot resolve the Bias-Variance trade-off properly**! Very often this solution is acceptable. However, it doesn't posses a **theoretical beauty**. It is (partly) a brute force solution! It solves the problem, but **it misses the strength of the beauty!**

Assume, you get **a present from your Grandpa**! A nice sport car, but just a little too old. Say, **no air-conditioning**?! And, being ingenious, you solve it as follows!

However, we long for the strength of the beauty of a good theory!!!

(or, for the beauty of a strength)

Despite all the possibly nasty comments about CV, for example, as not being based on the strong theory, CV is very useful (actually the only reliable) tool.

Thus, let make us familiar with, still **reliable** and **useful** **cross-validation** technique

Note, that by CV
we optimize the so-called
HYPERPARAMETERS of NNs

There are several hyperparameters which define a capacity of NNs

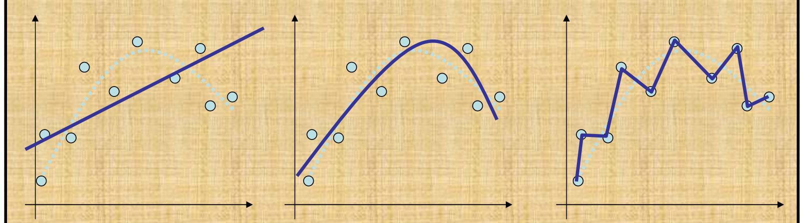
The basic ones are:

1) **a number of neurons in a hidden layer** ***J***
2) **a number of iteration steps** ***It***

Sure, there are few more, as learning rate η and a momentum term η_m but let's focus on the two most important ones CV parameters only

BASICS OF CROSS-VALIDATION

which is used, say, to choose a right model capacity for a given data.



Increase in model capacity (complexity)!

The basic modeling question is: how well will given model predict future (or, previously unseen) data drawn from the same distribution?

This is the question of GENERALIZATION.

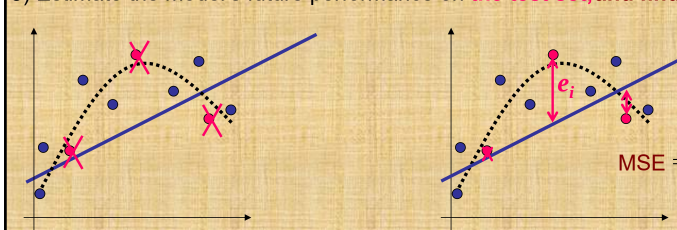
How can the good model be chosen HAVING ONLY TRAINING DATA?

BASICS OF CROSS-VALIDATION with A TEST SET

1) Randomly choose, say, 10% or 20% or ??% of the data to be in **a test set.**

2) Train your model (function from the hypothesis space) on the remainder data, that comprise **a training set.**

3) Estimate the model's future performance on **the test set, and find the MSE!**



MSE = $(\sum e_i^2)/l_{tst}$

If you repeat these steps for all models, you'll get something like below:

Linear model	Mean Square Error on the test set = 2.3
Quadratic model	Mean Square Error on the test set = **1.1**
Cubic model	Mean Square Error on the test set = 1.8

Well, this was done only one time

- It might have just happened that for this particular 10% of the data selected as the test data, the second order polynomial (MSE = 1.1) was the best, and so,
- repeat the experiments 100 times (or whatever may be, timewise, suitable)
- find the mean errors over 100 experiments for each model and you'll get the winner
- in the case of ties, pick up SIMPLER model

ADVANTAGES:

Extremely simple.

DISADVANTAGES:

Waste of data. The model is obtained by using 10% or 20% less data.

It can be detrimental in most of the modern age applications, where one works with scarce data.

Comment: If there are not enough data, our test set can be just

lucky, more or less lucky or unlucky,

and we may never know the true story until application,

when it may be too late!!!

For scarce data it may be useful to use the

LEAVE-ONE-OUT (LOO) cross-validation

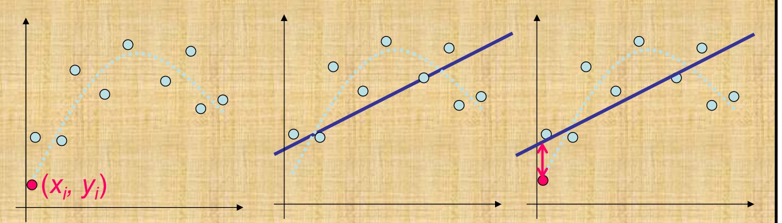
see next slide

LOO CV or Leave-One-Out Cross-Validation

For $i = 1, L$, *where L is the number of training data*

1) Temporarily remove the (x_i, y_i) data pair from the training set,
2) Train your model on remaining $L - 1$ datapoints,
3) Find and store the error on the removed data point (x_i, y_i).
4) After going through all points, calculate the mean square error of the model used, and choose another (more complex) model.

Select the model with the smallest MSE!



While LOO CV saves data,

it is computationally much more involved than CV with test set.

(You calculate the parameters of the model *L* times)!

Natural idea is; use a *K*-fold cross-validation, which instead of **ONE**-left-out, leaves ***K* data points** out (say, **5%, 10%, or 20%,** or so). It works same as LOO CV, but with *K* data points used for validation, and the above examples would be dubbed as **20-fold-CV**, **10-fold-CV** and **5-fold-CV**, respectively

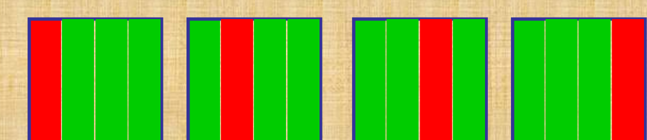
In the case of **classification** problems instead of computing the sum of squared errors **on a test set**, one computes the **total number of misclassifications on a test set**!

A lot of heuristics???!!!, Well, yeah,

but, very often all the mentioned, or presented, methods work very well.

Selection of a Model Hyper-Parameters

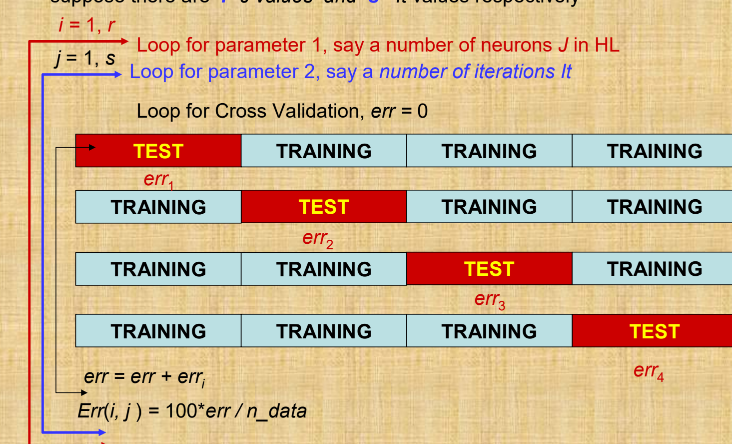
- We learn classification model on **training set**
- We evaluate model on dataset known as **test set**
- ***K-fold*-cross validation**
  - Split available data into *K* disjoint subsets
  - Select smaller subsets for test, a bigger for training
  - Repeat *K* times, such that each subset has chance to be test set i.e. that each data appears once in the test set
  - Average results from these *K* experiments



4-fold cross-validation

4-fold CV goes then as follows

Define parameter 1, say a number of neurons *J* in HL
Define parameter 2, say a *number of iterations It*
suppose there are ***r*** *J values and* ***s*** *It* values respectively

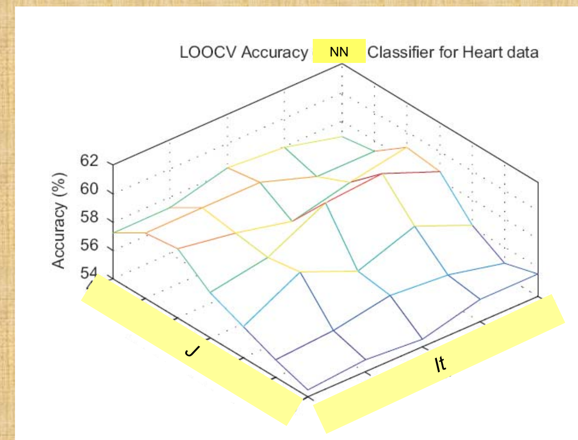


Find min(*Err*), and corresponding parameters *J* and *It*

What is a computational price for a *K*-fold-CV

- Well, take for example designing **NL SVMs** by using **Gaussian** kernels.
- We want to do **10**-fold-CV
- Number of HL neurons let be [5 10 25 50 100 250], **6** values, and
- A number of iterations *It* are [100 500 1000 2500 5000 7500 10000 15000], **8** values, and thus
- There will be 10*6*8 = **480** training runs resulting in an
- Error function *E(J, It) being a two dimensional function which can be shown as the surface over the J – It space, as shown on the next slide!!!*

The accuracy surface obtained by a CROSS-VALIDATION



Note that before training

- *i*) Usually, we shuffle all data set
- *ii*) Then, we scale data (or, do *ii* first and then *i*)
- *iii*) Test chunks must be of approximately same size.
- *iv*) In each training data set all classes must be present.
- *v*) Each data point must appear only once in all test chunks, meaning if data point $\mathbf{x}_i$ is in a test chunk *j*, it must not appear in all the other test chunks $k \neq j$.

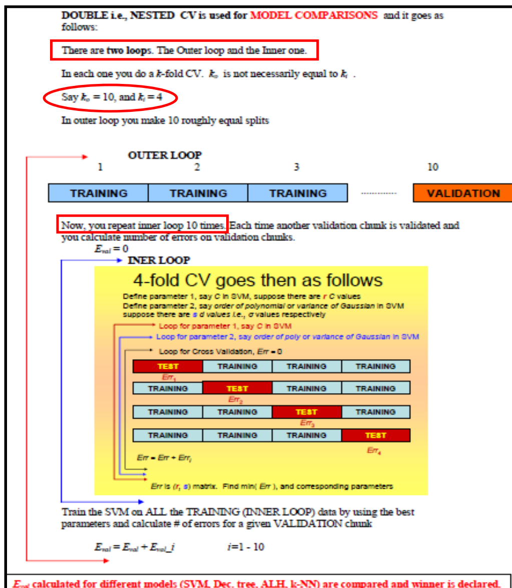
DOUBLE (NESTED) CROSSVALIDATION, i.e., DOUBLE RESAMPLING, is The Tool for Different Model COMPARISONS

and, **NOT FOR A MODEL DESIGN !!!**

What has been shown is the CV used for finding the best model (the one which makes the least number of errors), i.e., to find the hyperparameters that define the model

- **However, to compare performances of different models (say NN, SVMs, decision trees, KNN, etc.) one has to use the so-called double (i.e. nested) crossvalidation a.k.a. double resampling !!!**

- Given the dataset, this DOUBLE experimental procedure is **the only** fair approach for **MODEL COMPARISONS**



Environment for the double resampling experiments can be as follows for a NNs

Double-10x4-CV, 8x8 hyperparameters (*J*, *It*) which amounts to 2560 runs for the dataset.
This is a very time consuming process.
I was running it for weeks !!!

That would be all on the **CROSSVALIDATION**,

the good practical tool for reliable, but sometimes, slow, meaning, with a very long CPU time, procedure