

Applications: project presentation

Please don't Circulate !

Presentation anticipated to provide more time for the project

Presentazione anticipata per fornire maggior tempo per il progetto

Alessio Micheli

micheli@di.unipi.it

2018



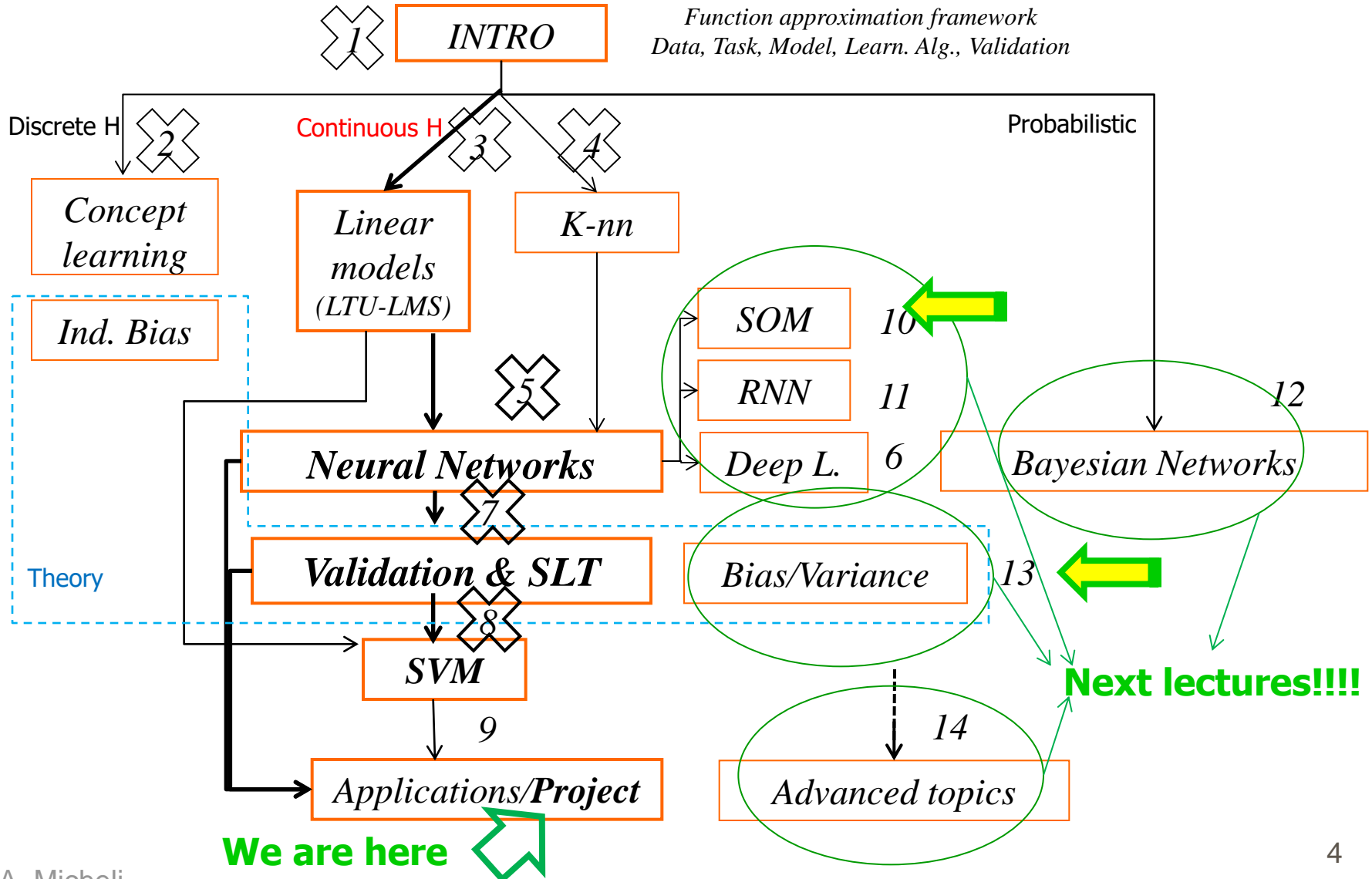
Dipartimento di Informatica
Università di Pisa - Italy

ML Course structure

Where we go



Dip. Informatica
University of Pisa



Programme

- 6 credits programme (AA1 for 2018) is ready: please check it on Moodle.
- The general programme is in (IT/EN):
<https://esami.unipi.it/esami2/programma.php?c=36545>
- The detailed programme for ML 9 credits will be presented LATER!

Notes on the formal exam subscription



Dip. Informatica
University of Pisa

- Date for the exam: esami.unipi.it portal
- Therein you find the date for the oral (beginning)
- The prj must be delivered ~10 days in advance:
typically oral date - ~10 (But see the exact deadline date and hour in the Moodle folder for your session)
- Take care also to register your name in the official UNIFI (esami.unipi.it) portal for exams (look the deadline)

Next Sessions

See the esami.unipi platform

Unofficial news for the first 2 sessions (to be confirmed!):

- Deadline **10/1** for material delivery → orals start **21/1**
- Deadline **30/1** for material delivery → orals start **11/2**
- You CAN always deliver before the deadline!
- For the following sessions consider again a *minimum* of ~10 days in advance for the material delivery
 - But see the ***exact deadline*** in the Moodle folder for your session

Assessment methods

(REPETITA from introduction)



Exam:

- **Project** (or Written exam)
 - Students have the **opportunity** to develop a project realizing/applying a learning system simulator (typically a simple neural network) and to validate it through benchmarks. A written report will show the results.
 - Great *opportunity* to apply the concepts by yourself
 - Great *opportunity* to show your concrete understanding and effort for the exam
 - **Deadline:** ~(around)10 days before the oral exam session (see the Moodle folder of your session for the exact deadline)
 - **See details in the lecture for project presentation**
 - ***Competition*** with *blind-test*
 - that is part of the benchmark results in the prj
 - 2018: also some joint proposals with CM course
- **Oral exam** (date according to the exams sessions)





Exam (II): Some details

- **Project:** we will discussed in a specific lecture all the details but from now it useful to know that:
 - It includes code + results (also for the competition) + report
 - It is made by a couple of students (exception MUST be justified and authorized in advance)
 - It is made just one time, i.e. with 1 delivery
 - It is corrected/discussed jointly with the oral exam (at the exam session)
 - *Programming language is free* (we will discuss later)
- **Oral:** Prj discussion + questions on *all the course content*
 - I'll ask answer first by math language (e.g. in the form of *equations* due to clarity, synthesis, ...; then we can discuss on them)



The main hints

- Follow the lectures and slides as a guide, studying *progressively* during the course
 - Special interactive classes will be used to assess activities and make a ML discussion forum (Q&A classes).



- A major hint *from past students*:

1. **FIRST** study the course content
2. **THEN** apply for the project

- For the PRJ: Develop (implement) a self made NN simulator if you are self-motivated and with good programming skills, else apply existing tools (type A or B see later)

Project General Rules: Deliver



Dip. Informatica
University of Pisa

1. Use the Moodle (elearning platform) for delivery of the Project (PRJ) package: code, report, cup-files (next slide)
 - See in Moodle the Section **Prj Student Material**
2. After the last upload (until the deadline):
Send email at micheli@di.unipi.it → It is a must (!) to use this tag
Subject: **[ML-2018]** Report by <your names>
Include in the main text:
 - Your name(s) & email contact information in the main text
 - Your degree course (master programme)
 - The name of file used in Moodle (to find it)
- Don't 'forget a CC to your colleague (all the group members must be included in the communications).
- Use the Moodle to deliver all the other files, do **not** attach them by email!!!
- Projects no announced by an e-mail will not be considered !!!
- (be sure I received it: resend the email if I do not reply with an ack.)

Project General Rules

Material



Dip. Informatica
University of Pisa

- Deliver the report and material according to the deadline in the Moodle folder, including:
 - **Code** (if it is developed by you, usually well commented)
 - **The report** resuming of the implementation and the experimental aspects: **Max** 10 pages (8 for 1 student group), **font 11** at least !
 - **Files** for the ML-CUP: “blind test set” + short abstract
(see the next slides with details)
- Bring with yourself the code to the oral exam (printed or electronic version)

Project General Rules: Groups



Dip. Informatica
University of Pisa

- Autonomy: it is part of the evaluation
- **Groups:** *it is assumed groups of 2 students*
 - $1+1=2.5!$
 - Increase autonomy and never discourage
 - Best results (in the general sense) in the past editions



- Can I do the project by myself only?
No, read the FAQ and "Partners for the project" in the Moodle:
 - Note that no individual projects will be accepted without a preventive agreement with the teacher (motivated by special need).
 - To help you to find a partner use the "Partners for the project" file.
 - **NEW:** I'm going to open it so that you can add or delete your name by yourself!!!
 - You can use it also after the first sessions!

The projects

The simulator

A premise

On the empirical approach

- Experimental method (using **empirical** evidence, from experiments) is the basis of the "*scientific method*" per-se.
- In our case, the numerical simulation to test your hypothesis on the model and the quantitative parameters.
 - (you don't know the result in advance, you have to formulate your hypothesis and then to measure the empirical results/observations, to compare and reason on the results etc.)
 - If you don't have any experience with experiments, this is new for you, but it is not negative but REALLY a nice and useful* experience you take the opportunity to gain !!! Exciting or lost? Break the ice !!!

* to build a skill for any field of science.



Possible Aims

A) Realize a ML/NN model simulator and apply it (**implementation**)

- Programming language is a free choice (C++, Python are popular for ML, or even environments as Matlab or R can be considered)
- A) with CM : a coordinate prj with CM

B) Extensive experimental applications of existing ML/NN simulators (**comparison**)

- Simulator is a free choice (a list will be discussed in the next slides)

C) Contact me. Only in case there are reasons/impediment not to apply to A and B: we can for instance think to a report developing a ML topic or other written exam or e.g. (special case PhD student) a seminar during the course !

A) or B): participate to the “**ML-2018 cup**” competition.

Let us see details for A) and B) and then the cup details.

A) Model implementation

Realize 1 ML model simulator : typically a **MLP Neural Networks (with regularization techniques)**

This is the typical project case, allowing you to realize a simple models by your original code, and to experiments all the variants that you like, and have fun;-)

Examples:

- **[typical case] Implement a MLP with backpropagation, momentum, and regularization L2:**
try the regularization and other hyper-parameters effects
- Backprop with variants for the weights upgrade : Quick-prop, Rprop, ... or other gradient based techniques.

Other models (but not for the CUP):

- MLP for classification: LMS versus Cross-entropy*
- Bayesian models: contact me.
- **SVM**: see A) with CM

General note: you can exploit numerical libraries,
e.g. NumPy, Armadillo (C++), .. 24

Special case <A) with CM>



Dip. Informatica
University of Pisa

A) with CM: A coordinate prj between ML and CM

See a detailed list of proposals within the CM course

- You will provide 1 report for ML with the basic A) results + the new result with the CM technique (comparison)
- AND 1 report for CM (according to CM teachers rules)

Categories (examples) [through CM approaches] :

- (improved) NN training by new descent methods algorithms
 - see many examples in the lectures NN2 Heuristics
 - [comparing wrt the basic gradient descent with momentum and L2 regularization]
- Non-differentiable optimization for Piece-Wise linear functions
 - PWL or ReLU activation function or *L1 regularization*
 - [comparing wrt the basic gradient descent with momentum and L2 regularization]
- SVM/SVR implementation (through different approaches), applied with kernels...
 - [for this case NN comparison is not need]

The proposal must be agreed by ML AND CM teachers!!!

You can still compete for the ML-CUP with your results.

<A) with CM> only for group of 2 students.

These are challenging prjs

25

More (double?) effort!, volunteer choice!

Notes on <A> with CM:

Further details on <A> with CM:

- More effort? It is because you like to try and show something more and not something less! This fully exercises your full understanding of the singular parts (by adapting them to the combined construction)
- For many cases it is useful to refine the literature basis (provided by us) to see previous studies of the impact of the used methods for the ML area.
- L2 regularization is the Tikhonov with norm-2 penalty
- L1 uses norm-1 penalty (we will discuss it later)
- (see lecture on linear models and NN-part2)
- Also ask for PWL if you are interested, ReLU will be discussed later
- Note that the aim is not to improve the performance/results obtained with the basic approach, but to critically exercise the use of CM approaches for ML*
- Unforeseen issues? Discussed by a case-based approach*

B) Compare models

- **Extensive experimental applications of existing ML/NN simulators**
- Compare different models/ existing tools e.g.:
 - SVM or NN by standard library versus basic models (linear, k-nn, naïve bayes, ...) implemented by yourself or by standard software tools
 - Compare NN vs SVM vs other models (even not included in the ML program) within the same software tool.
 - Compare 3 or more models even from different sw tools.
 - Compare different software tools for the same model (e.g. SVM or NN).
 - Compare different software tools.
- (also for fairness wrt A) the B case implies a larger effort on the comparison among models (including accurate validation) and to the experimental part
- The report can include also evaluation on the sensitivity to the hyperparameters values (for different models), efficacy, efficiency, predictive performance (of course!) but also issues of tool usage, usability, richness of the set of hyperparameters etc. (for different tools)
- Repetita: you can still apply to the CUP competition just selecting the best model to apply.

For both A) and B) cases

- For **NN heuristics** see the lecture on “Neural Networks: part 2”
- Try the effects of different configurations/ hyperparameters values according to your experimental schema (and explain the schema in the report)
 - In any case include the momentum and a regularization approach (weight decay, early stopping,). See all the (!) indications
- If you use a library, please not limit yourself to default values!
- You are free to choice the **model selection/assessment-evaluation** strategy:
whose fitness for the problem at hand is evaluated:
 - Directly from the description in the report
 - Through the results on the blind test set

FAQ:

When is a project A or B?



Dip. Informatica
University of Pisa

- If you program the NN model and back-propagation (or any training algorithm), and the CV approach, from scratch by your-self (independently from the software environment) is **type A**. E.g.:
 - A NN made by you within the standard Matlab is type A
 - A NN using Matlab NN tool box is type B
 - The same for TensorFlow and other frameworks
- Using high level libraries (SciKit learn, Keras etc.) is **type B**
- Using numerical or graphical libraries in your code maintain type A.

Simulators/Software Tools

- Software / Tools to be used for **B case**
- Or you can use a tool as an “oracle” to compare with your simulator for the **A case** (helping in assessing its correctness)
- If you use a library you must specify in the report the complete link to the source !!! (for both A and B cases)
- In the following some examples: an exhaustive list is out of our scope and it is even impossible to keep it updated!!!
- The best one is the one that is more useful FOR YOU.
 - Check also the documentation and if it still has maintenance/developments (new releases, support, ...)

Simulators/Software

General ML: examples (older)



Dip. Informatica
University of Pisa

- **Torch** (NN, SVM, AdaBoost, K-nn, Bayesian, ...) started as C++ library, now Lua/GPU emphasis !!!
- **Scikit-learn** (Python open source, many tools for preprocessing, model selection, linear models, regularization, SVM, DT, ... NN only from version 0.18 – even if still in a basic form, but wrappers for other sw exists) [JMLR 2011]
- **Shark** *(evolutionary and gradient-based algorithms, NN, kernel-based learning methods, SVM, ...[JMLR 2008]): C++ library
- **Dlib-ml** (Bayesian networks and kernel-based methods, clustering, anomaly detection, and feature ranking, ... [JMLR Jul 2009]): C++ library
- **Shogun** (SVM, HMM, K-NN, LDA,...[JMLR 2010]): C++/Python
- **Mlpack** (C++, armadillo matrix library, basic models, no NN/SVM, [JMLR 2013])
- **MLlib** Sparke scalable ML library (NumPy, R, Hadoop), NO NN yet [JMLR 2016]
- For Python: e.g. PyMVPA, MLPY , PyML, Plearn, **PyBrain**, ...

Environments:

- For **R** (Statistical Computing language)
- For **Matlab*** and **Octave**

Other examples (NN & Deep) (newer)



Dip. Informatica
University of Pisa

- **PyBrain** [JMLR 2010]: various NN architectures
- **Torch** (renewed for deep learning/GPU, coll. with Facebook, Google, Twitter ...)
- **Theano** (and Pylearn2): also for deep learning, GPU etc. wrappers with scikit-learn, ... (stop dev. on 9/17)
See http://deeplearning.net/software_links/
- **TensorFlow** (Google), open source, C++/Python like (API, interface). Include deep neural networks. Since November 2015!
- **Keras** is a high-level neural networks library (Python) capable of running on top of either TensorFlow, Theano, CNTK, MXNet, Deeplearning4j (+Wrappers versus Scikit-Learn API)
- **Caffe** is a deep learning framework, originally developed at UC Berkeley. It is open source, C++, with a Python and MATLAB interfaces
- **OpenNN** (Open Neural Networks Library) C++

Historical:

- **Stuttgart Neural Network Simulator: C, since 1995!**
- **SOM_PAK: C++ / SOM toolbox (free)**

Popular Supporting Libraries

- **Panda**: software library written for the Python programming language for data manipulation and analysis
- **NumPy (SciPy), Armadillo** (C++): numerical/linear algebra libraries
- https://en.wikipedia.org/wiki/List_of_numerical_libraries
- https://en.wikipedia.org/wiki/Comparison_of_linear_algebra_libraries

General note: you can exploit such numerical libraries for your code!

Simulators/Software

General ML: examples (2)



Dip. Informatica
University of Pisa

And many other...(started as DM tools).

- **Weka** (DM and ML Software in Java)
- RapidMiner
- Orange (JMLR 2013, C++/Python)
-
- Many commercial software ! ...→ "*predictive analytics*"

Visual workbench approaches are popular for commercial tools:

- E.g. (by students) **Knime**: data analytics platform: DM but also ML/NN (open source/commercial!)
- Major sw developers have now ML branch: azure (Microsoft), google, facebook, IBM, Amazon

Others & Others (from companies/commercial)



Dip. Informatica
University of Pisa

Just to witness what happens (not for your prj expect*):

- [Google](#): google prediction (service) / **TensorFlow** *
- [IBM](#): Watson
- [Microsoft](#) : Azure (platform and services)/ The Microsoft Cognitive Toolkit, **CNTK***, is a deep learning framework developed by Microsoft Research.
- [Amazon](#) Machine Learning (service) / **DSSTNE***: Deep Scalable Sparse Tensor Network Engine (open, Deep ML)
- [Facebook](#): FBLearner Flow, ...
-

Simulators for SVM

- **LIBSVM** (C++)
- SVM light (C)
- Torch (C++)
- JKernelMachines (Java library for learning with kernels [JMLR 2013]).
- Weka (Java)
- mySVM
- SVM in R
-
- http://www.support-vector-machines.org/SVM_soft.html

The projects

The application

Applications for the PRJ

Your results will include:

1) MONK benchmarks

2) The CUP data set (competition)

In the report: results for 1) and 2)

1) MONK benchmark [repetita]

- Difficulty of assessing implementation correctness:
- A first test ("*collaudo*") **Monk data set**
 - The results *must* be reported in the prj report: **performance and learning curve plots for the 3 monk tasks**

<http://archive.ics.uci.edu/ml/datasets/MONK's+Problems>

- 3 tasks of binary classification, small artificial data set, "not difficult" (a small NN with few units achieve a very high accuracy, up to 100%, with small time of convergence)

- There is a report with previous results using MLP (and others ML models)

- chapter 9 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.2363>

- Input encoding: 1-of-k → correspond to introduce 17 input units (see also sec 1.1)

- TS includes TR, which is a bad practice, but does not change here (since 100% accuracy is 100% accuracy also on the test set !!)

- Other sources of data sets for software tests: <http://archive.ics.uci.edu/ml/>
UCI Machine Learning Repository (hundreds of data sets !)

Solve it!

Monk data set results

- Please see examples of plots (learning curves for MSE and Accuracy) in previous lecture on NN:
the "Neural Networks: part 2" lecture (file: ML-**-NN-part2-...pdf)

Note:

- See "Other suggestions for your first trials" in the same lecture
- and examples of Hyperparameters were specified in the slides
- ❖ Good results on the MONK benchmark does not guarantee the simulator correctness
- ❖ Bad results on the MONK benchmark for sure require revision of the code/setting

2) ML CUP !



ML CUP 2018



I provide to you 1 data set for the cup

For the data set:

- I provide a training set and a blind test set (examples without target values)
- Apply 1 or more models selecting the final one that you think it is more accurate through the training set (used for training and validation, and your internal test).
- Report in the report document the TR/VALIDATION (and test) errors, in the original scale i.e. MEE for the 2018 cup (see next slides).
- Apply your final model to the blind test producing an output for each example of the blind test set and record them into an output file
- Provide with the report the output file containing the blind test results
- Test result: the accuracy on the blind test set will be automatically computed
- The final results will be summarized on the web site using your nicknames
- Glory to the winner! ;-)

Tasks and Data



Dip. Informatica
University of Pisa

- **Regression** on 2 target variables, i.e. x, y coordinates in a 2D space
 - FAQ: regression = linear output units
 - 2 target variables: 1 NN with 2 output units, or 2 NNs, 2 SVR,
- 1016 training examples
- Column 1: pattern name (id)
- Central columns: 10 variables with continuous values (from a noise source, real sensor data)
- Last 2 column: target of two continuous value variables : x and y .
- Blind Test set: 315 patterns, with the same input format (of course without the 2 target columns)

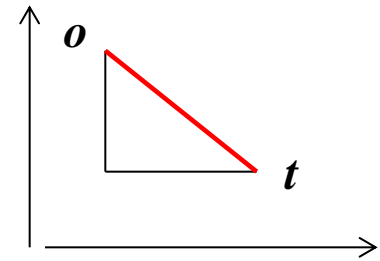
Task & Errors 2018

Report in the original scale the following error measure (**Euclidian distance**) :

- **Mean Euclidian Error** (it is the error used for the competition performance evaluation), where N =number of data, p =pattern, o =output, t =target

$$E_{MEE} = \frac{1}{N} \sum_{p=1}^N \|o_p - t_p\|_2 = \frac{1}{N} \sum_{p=1}^N \sqrt{(o_{p,x} - t_{p,x})^2 + (o_{p,y} - t_{p,y})^2}$$

Distance between 2 points in 2 dim space



- Typically you will also observe the (root) mean squared error (that is the typical loss used for the LMS training approach)

$$E_{MSE} = \frac{1}{N} \sum_{p=1}^N (o_p - t_p)^2 = \frac{1}{N} \sum_{p=1}^N ((o_{p,x} - t_{p,x})^2 + (o_{p,y} - t_{p,y})^2)$$

MEE \nleftrightarrow RMSE since $\text{sum}(\text{root}(a), \text{root}(b)) \nleftrightarrow \text{root}(\text{sum}(a,b))$ 72

What is provided 2018



Dip. Informatica
University of Pisa

- Training sets
 - ML-CUP18-TR.csv
- Blind test sets (without target)
 - ML-CUP18-TS.csv



!!!!!!



WHERE:

- **ML section on Moodle:** <https://elearning.di.unipi.it/>
- **Folder** [ML-18-PRJ lecture & package](#)
- With files for data, info (slides and txt), report-demo, results-demo
- Note: editions 2010-2017 **are NOT used this edition/ year**

What to produce 2018 (I)



Dip. Informatica
University of Pisa

- Output file in a simple text/*txt* format. The name is:
 - *team-name_ML-CUP18-TS.csv*
 - Using the following format
 - First 4 rows are for comments :
 - # your name/names
 - # team (nickname max 8-10 char) for the web results
 - # data set name (ML-CUP18 v1)
 - # date (e.g. 20 Dec 2018)
 - Table with 315 rows and 3 columns (comma separated values):
 - id, output_x, output_y
 - id ordered from 1 to 315 (exactly as for the file ML-CUP18-TS.csv)
- See the demo file: ***output_template_example-with-random-output-ML-CUP18-TS.csv*** (which is filled with random-output values for demo)

PLEASE, double check the output file format!!!!

If it is not OK we cannot evaluate it automatically → jump to the the rank bottom !

Repetita: you can send only 1 *team-name_ML-CUP18TS.csv* file
(assuming it is your best result)

What to produce (II)



Dip. Informatica
University of Pisa

- File: `team-name_abstract.txt`, in a simple text/*txt* format
- With a very short (5 rows) description of the used model and validation technique.
- HENCE, you have to send (for the cup):
 - `team-name_ML-CUP18-TS.csv`
 - `team-name_abstract.txt`
- Along with (see the “Project general rules” slide at the beginning of this lecture)
 - Your code
 - Your written report (with results on 3 MONKs and the CUP)

Results



- Initially I will personally communicate the result to the single participant during each oral (for fairness with participants following in time)
 - At the end of next year it is possible to publish the final ranking
 - And the winner ;-)
-
- Criteria for the winner on the task: accuracy (MEE) and possibly the quality of final plot of the results (at the discretion of the jury ;-))

Finally

- The report
- Other hints/request for the prj
- FAQs



The report

- See the DEMO/Template file: **ML-18-Report-template-v*.doc**
- Check the last version (vx.y)
- The demo file includes descriptions of the information to be provided by the report:
 - these are mandatory to accept the report as valid for the exam
 - double check you reported all the needed information
- Please follow such basic organization for your document

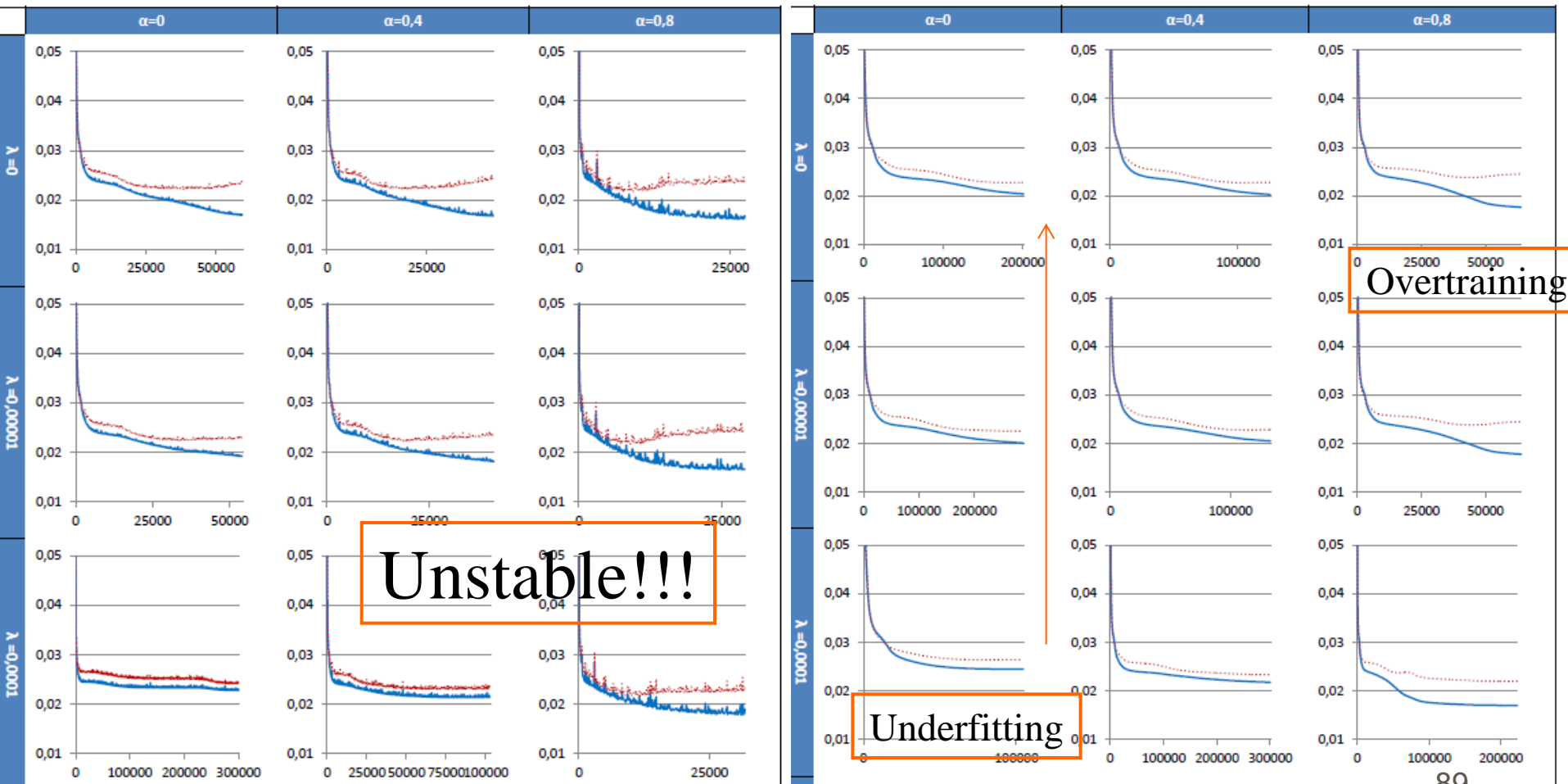


A zoom for Screening phase:

- Initially you will try different values of hyper-parameters
- (if you think interesting) significant cases of leaning curves for can be reported
- **but for yourself** plot them for various combination*
→the screening phase is essential for yourself to learn from this experience [see next slide demo]
- Results of this kind (also from the grid search could be part of the appendix)

Screening demo

- Some instances of screening phase (don't mind of the specific demo hyper. values, look to behavior changes...and make the same with your values)



How to evaluate your work

- Autonomy
- Soundness and quality of the (code) simulator (including any characteristics of modularity, efficiency, ...)
- Proper operation and behavior (e.g. see the regularity of learning behavior on the plots, ...)
- Pertinence of the choice for the model selection and evaluation
- **Quality of the report:** e.g. organization, rigor, soundness and synthesis of written text; motivations suitability; choices made; breadth and depth of the experimental investigation (although expressed in a concise way); **accuracy of the validation**.
- In case: difficulty of the challenge (if it is well done) can be a plus (e.g. <A> with CM>)
- The **blind test** result is just one of the possible parameters (not the most important one): very very often it strongly depends on the quality of the model and of the validation approach.

FAQ I



Dip. Informatica
University of Pisa

- **Deadline:** ~ (around)10 days in advance to the oral session is the deadline, before is better (e.g. sometimes additional info are required), and later is indeed in advance for the next session.
 - See the **exact deadline** for the project material delivery in the Moodle folder for your session, i.e. in Prj Student Material --> Session xx (your session of interest)
- **Groups** of 2 people is needed (see previous slide)
 - The evaluation (final mark) can be different for the two students (oral is different)
- What change using **MATLAB/OCTAVE/R?**
 - If you code by yourself → prj type A, but exploit your advantage (less time for the coding phase) to use more time in exploration/usage of advanced Matlab numerical computing functions/ possibly a comparison with other available models in the environment/extensive cross-validation/ impressive graphical results/....
 - If you use the NN toolbox → prj type B



FAQ II



Dip. Informatica
University of Pisa

- Is there any **check list** for material delivery?
 - Previous slide on <Project general rules> and <What to produce> (2 slides)
 - For the **report**: see the template file with [*] for mandatory info on the results
 - Template file: ML-18-Report-template-v*.doc
 - Check that cup **result file** has the proper name and format
 - Provide the following files:
 - Your code (if original, else use a link to libraries in the report)
 - Your written report (with results on 3 MONKS and the CUP)
 - *team-name_ML-CUP18-TS.csv*
 - *team-name_abstract.txt*
- **CSV** format for the results: see the files, it is a comma-separated values without spaces after the comma. Each pattern is a row. The header of input file has some rows of description beginning with #.
 - **Example:** output_template_example-with-random-output-ML-CUP18-TS.csv (which is filled with random-output values for demo)



FAQ III



Dip. Informatica
University of Pisa

- If my model **does not work well**?
 - Check the learning curve to find clues
 - HINT: Try to compare with respect to a known tool in the same condition
 - Autonomy is part of the evaluation
- **How many trials** ? How many do you feel useful in order to do a good model selection
- **Which trials** should I report? All the significant cases with details (experimental evidences), the other can be mentioned/synthesized in the text. Such choices are part of the quality of your work.
- **MSE** or **MEE**? You can use MSE (LMS) for training and MEE to evaluate



FAQ IV



Dip. Informatica
University of Pisa

- Plots , graphics etc.: the print is **Black&White**: it is mandatory to distinguish lines in the plots also using different lines symbols/style (to see them also in B/W)
- Report Format: free style (**but font ≥ 11**), typically **PDF** (include a pdf copy in any case).
- Italian or English? The language more easy for you ;-)
- Executable programs, libraries, ... (**large files**): until the package is small size (that can be sent as an attachment via email) try to include everything. If you have problems send/include (in the report and in the READ-ME file) a link to download the *** accessories files *** of great dimension



FAQ V



Dip. Informatica
University of Pisa

Even other QUESTIONS?



FAQ (NOT read in class I)

You can read them later:

- It is important to report the final Training, Validation and possibly Test error by the MEE measure?
Yes, it is a must. I cannot accept a report without these values.
- It is important to report how we obtained the final model (how selected)?
Yes, it is a must. I cannot accept a report without this information.
- Can you provide to me an idea of the basic result for the CUP, should I have near 0 like Monk?
No, zero errors is ONLY for Monks, an exception. **For reasons of equity, no indications can be given on the reference performance for the CUP.**
- How many layers do you suggest for the CUP?
A single hidden level is sufficient, but you can try two and compare (why not?). This holds also for student with 6 credits exam without following the final part of the course.
- The error to be reported in the tables is without the penalty term? Yes, without it
- General note on plots in the report: to be comparable, use a point for each epoch (defined as the total number of training data): if you are using on-line or mini-batch provide the mean over an epoch.

FAQ (NOT read in class II)

You can read them later:

- [Can we use other methods beyond grid search?](#)

Yes, but first grid search is mandatory. E.g. Random search : see Validation 1 (Valid1) lecture.

- [Is it needed an advanced validation approach for Monks?](#)

No, Monk dataset is a benchmark to assess your code and system. You can also assess the validation process with Monk, but it is not needed to address this task by a complex CV approach.

- [Making a prj with SVM, which plot can I do?](#)

See E.g. Slide 6 <ML-17-SVM-other inform> lecture (results w.r.t. hyper-parameters values, including epsilon-tube parameter)

- [Plots with MEE or MSE?](#)

MEE for the tables (and this is mandatory!), for plots up to you (typically the quality of the learning behavior does not change)

- [Usage of Early Stopping:](#)

See Validation 3 (Valid3) lecture

- [Prj A with CM: comparison also on Monks?](#)

Not needed! It is needed for the CUP.

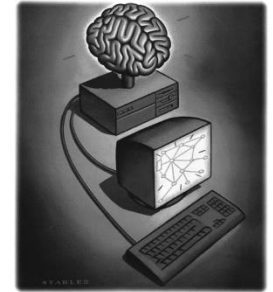
FAQ (NOT read in class III)

You can read them later:

- Does Monk converge always with few hundred of epochs"?
The model converge easily in "most of the case" with the suggested setting (and also with other settings). But not "always" due to the sensitivity to the random initialization using a large range of the initial weights. Please see in the lecture NN2.
- So what accuracy have to be reported for the Monk, just the best over different trials/initializations?
No, the mean accuracy over that trials. See the discussion above.
- How many units for Monk? Follow the manual as baseline and then you can try also different values
- How many units for the CUP? You have to search by yourself, but it is completed unrelated wit respect to the Monk task!!!

Enjoy !

- ... Now you can **enjoy** with ML !



Remember to have
FUN !!!

Future – just ahead

- CIML@Pisa
- Didactics: other courses (ISPR, CNS,...)
- General topics for ML applications
- CIML@Pisa research



**2018 Note: THIS PART WILL BE PRESENTED LATER,
If you are a AA1 student (6 credits) but
you like to see them, please contact me
to know the date of the lecture**

DRAFT, please do not circulate!

For information

Alessio Micheli

micheli@di.unipi.it

www.di.unipi.it/groups/ciml



Dipartimento di Informatica
Università di Pisa - Italy



**Computational Intelligence &
Machine Learning Group**