Applications: project presentation

Please don’t Circulate !

Anticipated presentation to provide more time for the project

Presentazione anticipata per fornire maggior tempo per il progetto

Alessio Micheli

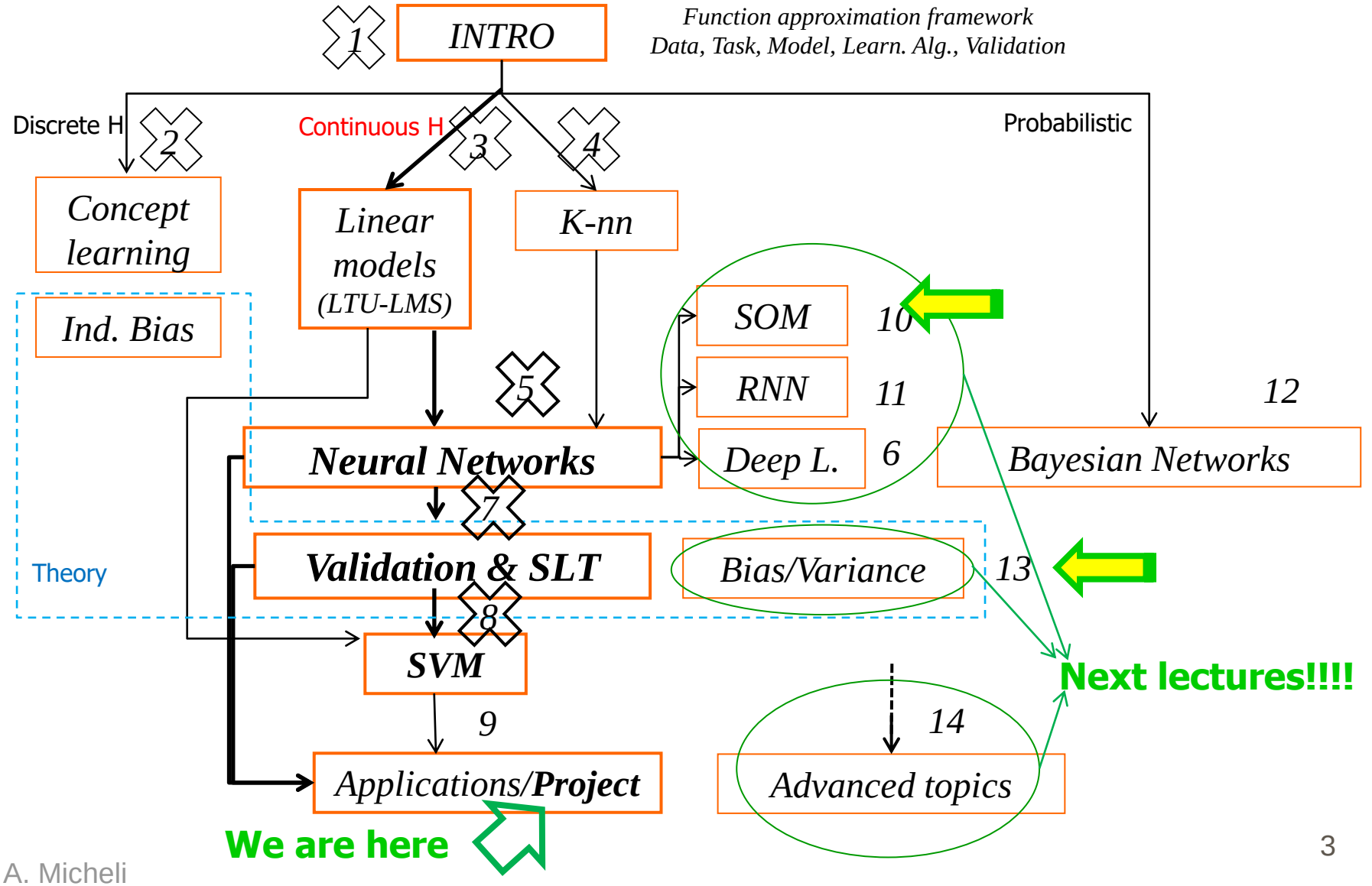
micheli@di.unipi.it

2023

Dipartimento di Informatica
Università di Pisa - Italy

ML Course structure
Where we go

Dip. Informatica
University of Pisa



Assessment methods
(REPETITA from introduction)

Dip. Informatica
University of Pisa

Exam:

- **Project** (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 (slides) 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/or **more than**) 10 working 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 → TODAY**
  - We include a *competition* with *blind-test*
    - which is part of the benchmark results in the prj
    - also some joint proposals with CM course
- **Oral exam** (dates according to the exams sessions)

Exam (II): Some details
(REPETITA from introduction)

Dip. Informatica
University of Pisa

- **Project**: we will discuss in a *specific lecture* all the details but from now it useful to know that:
  - It is made by a team of 2 or 3 students (exceptions are very rare, subject to restrictions, MUST be justified by serious impairments and authorized in advance : see later)
    - A shared document *is open* to find partners (see Moodle)
  - It is made just one time, i.e. with 1 delivery
  - It is corrected/discussed jointly with the oral exam (at the exam session)
    - So all the team members need to do the exam in the session
  - It is an implementation of a NN/ML model or an application of ML SW (type A or B)

… (see the Introduction slides)

The main hint
(REPETITA from introduction)

Dip. Informatica
University of Pisa

A major hint *from past students*:

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

Exam: schematic synthesis (REPETITA from introduction)

Dip. Informatica
University of Pisa

In the following order:

1. **Course lectures** (stay on-line with them!)
   - And perform the **Intermediate tests** during the course
2. **Project work** (you can start around in the middle of the semester)
3. **Project package delivery:** results, slides report etc. at the date specified in the Moodle for each session
4. **Oral:** at the date of the official exam session (for the questions in written form) (it is in the same session of the prj delivery).
5. **Project presentation and discussion, and oral finalization,** according to a specific calendar for each group, i.e. as soon as possible in one of the following days (*ALL the students of the group must be present for the exam, i.e. making the exam in the same session*).

I also updated the FAQ section in the Moodle with the "pipeline for the project/exam"

The project at a glance

Dip. Informatica
University of Pisa

- In the following: A top-down presentation (first general info/rules than details)

In short (the **phases**):

I. Form a **group**

II. Prepare a **simulator**, made by you (prj type A) or from software libraries (prj type B)

III. Apply (**experimental part**) to all the benchmarks (including the CUP)

IV. Write your **slides** report (with all the info indicated in the demo & info files) and the **results for the cup** (following the provide file format)

V. Deliver the **package** till the deadline

VI. You will **present the slides** at the beginning of your oral

- *It is easy*, all the rest are details (although important)

Good news

Dip. Informatica
University of Pisa

- This year we have a ML course assistant (Lorenzo Simone <lorenzo.simone@di.unipi.it>)
  - to help for the project development (but **not** for questions on the course!)

NEW

- Take the opportunity to interact with him/them during the course

- Also available via email or de-visu (room 382) or video-call each Wednesday evening,
  better by fixing an appointment by email in advance with your group

Let's start with phases I and V

- **Form a group (phase I)**
- **Delivery and deadlines (phase V)**
  - Note (the phase V include rules for all the students)

Project General Rules: I. form a group (I)

Dip. Informatica
University of Pisa

- **Groups**: *it is assumed groups of 2 or 3 students*
  - For the prj with CM see later

- Advantages of a team work:
  - The result is greater than the sum
  - Increase autonomy and never discourage
  - Mutual enrichment of experience
  - Best results (in the general sense) in the past editions
  - Of course, the evaluation (final mark) can be different for the students (the orals are different for each student)

  - [Repetita] Please, remember that the project is corrected/discussed jointly with all the students within the final oral exam and hence the oral is given at the same session (in the same day) for all the group candidates.

Project General Rules: I. form a group (II)

Dip. Informatica
University of Pisa

FAQ: Can you help me to find partners to form a team?

- To find partners (a team or the third partner) see the shared document with a link in the Moodle [**Prj partners searching page**].

- You can fill it with your preferences and contacts so that other students can contact you (and you the others) with a good matching.
- Connect with the **UNIPI account** to edit it.
- You can use it also after the first sessions!

- I invite also students in a team of two members to see the **Prj partners searching page** to form a team with 3 students

Project General Rules: I. form a group (III)

Dip. Informatica
University of Pisa

Can we do a group of **1 student (by myself)**?

- **No prj can be done alone**, the project is a team work
- You lose the *opportunity* to do a prj, you will have instead an *additional exam part* *
- Moreover, the qualification in this "*individual student*" category needs to satisfy 2 conditions:
  1. a preventive trial through the "Partners searching page" if you are searching for a partner
  2. a preventive (I mean largey before the prj delivery deadline) agreement with the teacher (by email), motivated by a very special (serious and rare) impairments

V. Deadlines and important notes: <formal exam> subscription and prj

Dip. Informatica
University of Pisa

- Date for the exams: esami.unipi.it portal
- Therein you will find the "session date" = date for the oral (period beginning) (this holds for all the students)

1. Take care to register your name in the official UNIPI (esami.unipi.it) portal for exams (check the **deadline** in large advance): *It is a must*, only registered student can be admitted to the exam (all students rule)

2. The prj delivery has a different **deadline** (automatic in the Moodle): the deadline is *typically* =session date **–** (minus) ~10-14 working days (but see the exact deadline date and hour in the Moodle folder for your session, check it in large advance!):
   - especially at the first sessions a *greater time* is need to correct all the projects (or sometimes due to other, often formal, commitments)

- Of course, you CAN always deliver before the deadline!

- The **two deadlines** are different (one is for the prj, one is for you)!

V. Notes on timing

Dip. Informatica
University of Pisa

- You will have all the sessions of the academic year to deliver your project and make the exam
- For the first session (on January), I will take in account the shorter time you have to complete the project,
  - with a larger tolerance on the "quantity" (extension of your investigation)

V. Project General Rules: Material

Dip. Informatica
University of Pisa

- Deliver the report and material according to the **deadline** in the Moodle folder, with a package including (see also FAQ II):
  - **Code** (the code developed by you, usually well commented)
  - **The slides** (report) describing the implementation and the experimental aspects:
    - Details later for presentation time, slides, font, demo etc.
      (a file describing in detail the report, and a demo, is in the folder for the project)
  - **Files** for the ML-CUP: "blind test set" + a short abstract
    (see the next slides with details)

- Bring with yourself the code to the oral exam (printed or electronic version)

V. Project General Rules: How to deliver it

Dip. Informatica
University of Pisa

1. Use the Moodle (elearning platform) to deliver the Project (PRJ) a zip package: code, report, cup-files, short-abstract
   - See in Moodle the Section **Prj Student Material**
2. After the last upload (until the deadline):
   Send email at both micheli@di.unipi.it, lorenzo.simone@di.unipi.it
   Subject: **[ML-2023]** 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)

It is a must **(!)** to use this **tag**

- Don't forget a CC to your colleague (*all the group members* must be included in *all* 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.)

The project

- **Premise**
- **Type A, B, etc. (phase II)**
  - The simulators
- **The application (phase III)**
- **The slides (phase IV)**
- **And slides presentation (Phase VI)**

First premise
On the empirical approach (I)

Dip. Informatica
University of Pisa

- Experimental method (using ***empirical*** evidence, from experiments) is the basis of the "*scientific method*" per-se
- In our case, the experiments are the numerical simulations to test your hypothesis on the models and its 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 to reason on the results etc.
  - Don't worry if this is a new experience for you (for many Computer scientists)
  - In any case, this experience will be useful to you as a ground for *your thesis*

First premise
On the empirical approach (II)

Dip. Informatica
University of Pisa

- You don't know the result in advance, …
- So, *"I feel lost", what have I to expect?*
  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 !!!
- Nothing explodes: explore different cases, it is not dangerous! It is a numerical experiments 😉

Students: "*The challenging of implement different solvers and hyperparameters selection figured out to be interesting and one thing that makes us learn a lot.*
Initially the experience was a little disorienting for the huge amount of point to test, but going ahead the funny part was the same that at the start scare us."

Second premise: The real aim of the project

Dip. Informatica
University of Pisa

The real aim of the project is not to achieve a state –of–the art result (least of all frustrating you with your first ML experience)

- The NN or ML models implementation or usage is a mean, your understanding is my/ the objective
- Learning with NN or ML models is a mean, your learning on how such models work is my/the aim

YOU are the objective!

The project

- **Premise**
- **Type A, B, etc. (phase II)**
  - The simulators (ML tools)
- **The application (phase III)**
- **The slides (phase IV)**
- **And slides presentation (Phase VI)**

II. Possible Aims: Type of projects

Dip. Informatica
University of Pisa

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

- Especially useful for students that will go to develop new models *
- 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)

- Especially useful for students that will go to apply ML in other fields
- Simulator is a free choice (a list will be discussed in the next slides)

C) Contact me. Only in case there are serious reasons/impediment not to apply to A and B. Or special cases, e.g. for PhD students

Both **A)** or **B)**: participate to the "**ML-2023 cup**" competition.

There is **no** difference in the grade and in the global effort (≠aim)

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

Type A) Model implementation

Dip. Informatica
University of Pisa

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** (multilayers)**:** try the regularization and other hyper-parameters effects
- Backprop with variants for the weights upgrade : Quick-prop, Rprop, … or other gradient based techniques. Or Cas-Cor, etc.

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++), ..

A) Model implementation (II)

Dip. Informatica
University of Pisa

- With a group of 3 students take the opportunity to try something more than the basic setting (I expected to see these), examples (see the NN-part2 lectures with many variants beyond the topic with the **(!) symbol**, e.g. with the  symbol), examples:
  - Comparison among different strategies for learning rate
  - Consider also the Nesterov momentum and compare
  - Comparison among different regularization approaches
  - Different architectures (the minimum is 1 and 2 hidden vs multilayers)
  - Different activation functions
  - Any variants to the training algorithm (beyond SGD, e.g. R-prop, Quick-prop, but see also CM projects)
  - Type A for NN + 1 other model from libraries (as for type B) to compare
  - Type <A with CM> (also implementing 2 models, e.g. NN and SVM)
  - …
- Not necessarly all them, e.g. just two, and then … *it is up to you*!
- Of course, a group of 2 can anyway consider some of them as well!

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 results with the CM technique (comparison)
- AND **1 report for CM** (according to CM teachers' rules, and a different aim)

Categories (examples) [through CM approaches] :

- (improved) NN training by other gradient descent methods algorithms
  - see many examples in the lectures NN2 Heuristics (Conjugate gradients, …)

  [comparing w.r.t. 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 w.r.t. the basic gradient descent with momentum and L2 regularization]
- SVM/SVR implementation (through different approaches), applied with kernels...
  - [for this case see the FAQ for the NN comparison with NNs]

- **If** it is not in the CM list, or it includes a ML part not considered as basic prj A (see the FAQ), the proposal must be agreed by ML **AND** CM teachers!!!
- You can still compete for the ML-CUP with your results.

Notes on <A) with CM>

Dip. Informatica
University of Pisa

Further details on<A) with CM>:

- More effort? It is because you like to try and show something more (a volunteer challenging choice) and not something less! This fully exercises your full understanding of the singular parts (by adapting them to the combined construction)
- For may cases it is useful to refine the literature basis (provide by us) to see previous studies of the impact of the used methods for the ML area.
- So, we consider this **a plus** (if it is well done)

Notes:

- 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 (re)discussed also 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 (the aim is a critical analysis the results)*
- Unforeseen issues? You can go back: discussed by a case-based approach*

Notes on <A) with CM>: groups

Dip. Informatica
University of Pisa

You can do <A) with CM> either:

- In group of **3** (you have to agree this with the CM teachers with the modality specified by them)
- In a group of **3**, but only 2 are making the prj with CM, the third one is making only the ML prj (for my side)
- A group of 2

- Note: these are **two different** projects → you are NOT forced to sustain the exam of ML and CM at the same session, provided that, of course, the results will be already present for ML in the report provided to me, you can do CM in any other time.

Type B) Compare models

Dip. Informatica
University of Pisa

- **Extensive experimental applications of existing ML/NN simulators**
- Compare different models/ existing tools, e.g.:
  - SVM and/or NN by a 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.
- The code written by yourself (to use the libraries) must be included in the package

Further notes

Dip. Informatica
University of Pisa

See the FAQ section (end of these slides) for further aspects:

- The boundary between A and B according to the usage of different libraries
- The applications of different models for PRJ B to the provided benchmarks

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 and 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* (but see in the FAQ on Matlab)
  - Instead, a NN using Matlab NN/DL *toolbox* is *type B*
  - The same for TensorFlow, PyTorch, Keras and other frameworks (see later): it is *type B*.

- Using high level libraries (SciKit learn, Keras etc.) is *type B*
  - However, if for example you just exploit part of the PyTorch features (e.g. the "automatic differentiation") but you implement the main training loop, the CV etc. you can indicate it in the report.
- Using numerical or graphical libraries in your code maintains it of *type A* prj.

For both A) and B) cases

Dip. Informatica
University of Pisa

- For **NN heuristics** see the lectures 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, for NN, include the momentum and a regularization approach (weight decay, early stopping, ….). See all the **(!)** indications
  - For type A: consider examples in the slide "A) Model implementation (II)"
  - For type B (*ça va sans dire*): if you use a library, please not limit yourself to default values!

- You are free to choose 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

The project

- **Premise**
- **Type A, B, etc. (phase II)**
  - **The simulators (ML tools)**
- **The application (phase III)**
- **The slides (phase IV)**
- **And slides presentation (Phase VI)**

Simulators/Software Tools

Dip. Informatica
University of Pisa

- Software / ML Tools to be used for the **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, use a READ-ME file in the code package)

- In the following some examples: an exhaustive list is out of our scope and it is even impossible to keep it updated!!!
  - Developers make changes everyday!!! (big companies have a major role)

- The best one is the one that is more useful FOR YOU.
  - Check also the documentation and if it still have maintenance/developments (new releases, support, …)
  - Some of them have a reference in [JMLR]= http://jmlr.csail.mit.edu/mloss/

Simulators/Software Main (ML & NN & Deep)

Dip. Informatica
University of Pisa

- **Keras was** a high-level Neural Networks/Deep Learning library (Python) capable of running on top of either TensorFlow , Microsoft Cognitive Toolokit (formerly CNTK), Theano and others, (+Wrappers versus Scikit-Learn API).
  Nowadays it acts as an interface for TensorFlow library (see below) !
- **PyTorch** is an open-source machine learning library for Python, based on Torch (was in C++). Neural Networks, Deep Learning. It exploits "automatic differentiation"*. Developed by *Facebook AI* research group (and Uber)!
  - **Torch** (NN, SVM, AdaBoost, K-nn, Bayesian, …renewed for deep learning/GPU, coll. with Facebook, Google, Twitter …development stopped in 2018)
- **Theano** (and Pylearn2): also for deep learning, GPU etc. wrappers with scikit-learn, … (stop dev. on 9/17)
- **Caffe** is a deep learning framework, originally developed at UC Berkeley. It is open source, C++, with a Python and MATLAB interfaces
- **TensorFlow** *(Google)*, open source, C++/Python like (API, interface). Include deep neural networks. Released as open since November 2015. Now: *TensorFlow 2.0*.
  - Used by Google for all their products, Photos, Gmail, speech recognition, …
- Others:https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

Simulators/Software Main (ML & NN & Deep)

Dip. Informatica
University of Pisa

- **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]:
  - very good documentation with graphical examples: HAVE A LOOK!

Environments:

- **R** (Statistical Computing language): programming language and free software environment for statistical computing
- **Matlab**: multi-paradigm numerical computing environment and proprietary programming language (available at Unipi),
  - or as a simulator for NN ("Deep Learning Toolbox")
- and **GNU Octave** (free, Matlab compatible)
- **Wolfram Mathematica** (mathematical computation integrated tolls)

Others … (often old) (not read in class)

Dip. Informatica
University of Pisa

- **MXNet:** by Apache, is open-source deep learning software framework. Deep neural networks. Multiple programming languages.
  Related to Gluon : Amazon + Microsoft (2017)
- **MLlib** Spark scalable ML library (NumPy, R, Hadoop), NO NN yet [JMLR 2016]
- **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])
- **OpenNN** (Open Neural Networks Library) C++ (bugs!)
- Eclipse **Deeplearning4j** is a deep learning programming library written for Java
- Others for Python: e.g. PyMVPA, MLPY , PyML, Plearn, **PyBrain** [JMLR 2010]: **, …** check if still supported)

See also: https://en.wikipedia.org/wiki/Comparison_of_deep-learning_software

Historical:

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

Other examples (cnt)

Dip. Informatica
University of Pisa

And many other…(started as DM tools).

- Weka (DM and ML Software in Java)
  - (not well suited for us in terms of hyperparameters control)
- RapidMiner
- Orange (JMLR 2013, C++/Python)
- ….
- Many commercial software ! …→ "*predictive analytics*"

Visual workbench approaches are popular for commercial tools:

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

Just to witness the interest of the big companies

Dip. Informatica
University of Pisa

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

- Google: google prediction (service) / **TensorFlow** *
- IBM: Watson
- Microsoft : Azure (platform and services)/ The Microsoft Cognitive Toolkit (formerly **CNTK)**, is a deep learning framework developed by Microsoft Research.
- Amazon Machine Learning (service) / (old) DSSTNE: Deep Scalable Sparse Tensor Network Engine (open, Deep ML)
- Facebook (AI research lab): ~~FBLearner Flow,~~ → **PyTorch**…
  - (used by FB for computer vision and NLP, and also by Tesla, Uber, …)
- ……

Simulators for SVM

Dip. Informatica
University of Pisa

- **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

Popular Supporting Libraries

Dip. Informatica
University of Pisa

- **Pandas**: software library written for the Python programming language for data manipulation and analysis
- **NumPy** (*popular for NN!*) (**SciPy**), **Armadillo** (C++): numerical/linear algebra libraries: suggested also for prj of type A!!!

- 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!

The project

- **Premise**
- **Type A, B, etc. (phase II)**
  - The simulators
- **The applications (phase III)**
  1. MONK
  2. CUP
- **The slides (phase IV)**
- **And slides presentation (Phase VI)**

III. Applications for the PRJ

Dip. Informatica
University of Pisa

Your results will include (for **both** prj A and B type):

1) MONK benchmarks

2) The CUP data set (competition)

In the report: results for 1) and 2)

III. 1) MONK benchmark [repetita]

Dip. Informatica
University of Pisa

- 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

Solve it!

  - Input encoding: 6 variables with 2, 3 or 4 symbolic values each: 1-of-k → correspond to introduce 17 input units (see also sec 1.1 and check it!)
  - 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 (hundres of data sets !)

Monk data set results

Dip. Informatica
University of Pisa

- 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)

  - 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 your revision of the code/setting

- Also: the results on Monk should be achieved using a **small network** (2-4 units): The test works if you achieve the state of the art results with few units (you have to not to improve the accuracy).
- For the CUP, on the other hand, there are no constraints, because instead you have to do the best performance ...

III. 2) ML CUP !

Dip. Informatica
University of Pisa

ML CUP 2023

Dip. Informatica
University of Pisa

- I provide to you a **data set for the CUP**, with a training set and a blind test set (examples without target values)
- Apply the models that you are investigating selecting the final one that you think it is more accurate through the training set (used for training and validation, and your *internal test **) (*see the FAQ).
- Report in the "report (slides)" the TR/VALIDATION and your internal TEST errors, in the original scale i.e. MEE for the 2023 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 and 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 3 target variables, i.e. x,y,z coordinates, in a 3D space
  - **3** target variables: 1 NN with 3 *linear* output units, or 3 NNs, 3 SVR, ….

- 1000 **training examples**
  - Column 1: pattern name (id)
  - 10 central columns: **10** variables with continuous values (from a noise source)
  - Last 3 columns: target of 3 continuous value variables : x,y,z

- **Blind test set**: 900 patterns, with the same input format (of course without the 3 target columns)

Task & Errors 2023

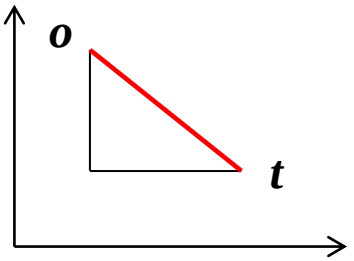
Dip. Informatica
University of Pisa

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 l=number of data, p=pattern, o=output, t=target, x, y and z the coordinates in the 3D space*

$$E_{MEE} = \frac{1}{l}\sum_{p=1}^{l} \|\boldsymbol{o}_p - \boldsymbol{t}_p\|_2 =$$

$$= \frac{1}{l}\sum_{p=1}^{l} \sqrt{(o_{p,x} - t_{p,x})^2 + (o_{p,y} - t_{p,y})^2 + (o_{p,z} - t_{p,z})^2}$$



E.g. 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)

$$E_{MSE} = \frac{1}{l}\sum_{p=1}^{l}(\boldsymbol{o}_p - \boldsymbol{t}_p)^2 = \frac{1}{l}\sum_{p=1}^{l}((o_{p,x} - t_{p,x})^2 + (o_{p,y} - t_{p,y})^2 + (o_{p,z} - t_{p,z})^2)$$

Note that MEE≠RootMSE since sum(root(a), root(b)) ≠ root (sum(a,b))

What is provided 2023

Dip. Informatica
University of Pisa

- Training set
  - ML-CUP23-TR.csv
- Blind test set (without target)
  - ML-CUP23-TS.csv

!!!!!

WHERE:

- **ML section on Moodle: https://elearning.di.unipi.it/**
- **Folder** ML-23-PRJ lecture & package *
- With files for cup data, info (slides and txt), slides-demo & report-info, cup results template

- Note: past editions **are NOT valid for this edition/ year**

What to produce 2023 (I)

Dip. Informatica
University of Pisa

- **Output file** in a simple text/*txt* format. The name must be:
  - *team-name*_ML-CUP23-TS.csv
- Using the following format
  - First 4 rows are for comments :
    - # your names
    - # team (nickname max 8-10 char) for the web results
    - # data set name (ML-CUP23 v1)
    - # date (e.g. 2 Jan 2023)
  - Table with 900 rows and 3 columns (comma separated values):
    - id, output_x, output_y, output_z
    - id ordered from 1 to 900 (exactly as for the file ML-CUP23-TS.csv)

    See the demo file: ***template-example-with-random-outputs_ML-CUP23-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 bottom of the rank !

Repetita: you can send only 1 *team-name*_ML-CUP-23-TS.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 final model** (e.g. NN/SVR/…) and validation technique.

- HENCE, you have to send (for the cup):
  - *team-name*_ML-CUP23-TS.csv
  - *team-name*_abstract.txt

- Along with (see the "Project general rules" slide at the beginning of this lecture)
  - Your code
  - Your written slides report (with results on 3 MONKs and the CUP)

Results

Dip. Informatica
University of Pisa

- 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 award *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 ;-))

IV. The report

- **Premise**
- **Type A, B, etc. (phase II)**
  - The simulators
- **The application (phase III)**
- **The slides (phase IV)**
  - And template demo & info files
- **And slides presentation (Phase VI)**

The report: slides

Dip. Informatica
University of Pisa

- To help you: see the *demo*/template of the slides file: **ML-2023-PRJ-Slides-Demo** **https://docs.google.com/presentation/d/1Gyb3t9CjxvZY1ffvAmmjU2BW1dDNZq5rVg7mlInS6vQ**
  - Check the last version (Vx.y)
  - Don't' change the demo **(make your copy!)**
  - Use your UniPi account to connect
  - A copy will be part of the package
- The demo file includes descriptions of the information to be provided:
  - these are mandatory to accept the report as valid for the exam
  - **double check in the demo files if you reported all the needed information** (see also the next slide)
- Please follow such basic organization for your slides
- English language

An other auxiliary file

Dip. Informatica
University of Pisa

- Use also this document with useful info for your project and slides content, file:
  ML-23-Report-info-v*.pdf

- It helps you, it is in the form of a report since in the past we used a report style (instead of slides), …

- But, this year don't write a report in that form, ***you have to prepare only slides and present them at your oral session***

Slide presentation: mandatory rules

Dip. Informatica
University of Pisa

- Time: you will have sharp time to present
  - **Max. 20 minutes** for a group of 2 students
  - **Max. 25 minutes** for a group of 3 students
- I'll interrupt the presentation after this time, due to conciseness and fairness reasons
- I expect that all the members will present for the same time (so 10 min. each for a group of two, or 8 min. each for a group of three)

- This typically correspond to around
  - **Around 20** slides for a group of 2 students
  - **Around 20-24** slides for a group of 3 students
- Font: **font** ≥ 14 for the text (exception for some rare cases to 12 for secondary things), typically a **PDF** file (include a pdf copy in any case)
  - Small font slides will be rejected (we will control also with a tool)
  - Too full slides are not useful for our aims (see later)

Slide presentation: suggestions!

Dip. Informatica
University of Pisa

- Synthesis issue:
  - **Focus on what is relevant to show**!
  - Given that you provide all the necessary information in the slides (see the demo & info files), the choices regarding the synthesis of the all the other results depends on *your decisions/preferences*.
  - The *time and space limit* must not be forced, also because it is the constraint that imposes the synthesis in order to *exercise/test your choices.*
- Don't forget to ***empathize the novelties*** that you introduced in your model and/or any significant/critical analyses and ***any interesting finding/insight***.
  1. [model/approach/techniques/implementation **novelties**] It is important to stress what you have been investigated beyond ordinary techniques (if any)
  2. [**Discussion**] Evaluate the effects of the novelties on the results.
     Also, don't forget a "Discussion/Analysis" answering "what did you learn?" on the Models/Hyperparameters/Results/Efficacy/Efficiency, selecting/highlighting what is *more significant in your opinion* (again, time and space constrains helps!)

Look at …

Dip. Informatica
University of Pisa

I'll in particular evaluate (not in this order):

- Adherence to requests (in particular see the [*] sign in the ML-23-Report-info file for mandatory info)
- Correctness with respect to the ML concepts
- Clarity and correctness of the validation/assessment procedure and choice of the final model
- Effort to achieve the best CUP result (not necessarily the result in itself)
- Depth, introduced novelties, and quality of the analysis and discussion
- Quality/clarity of the oral presentation
- Autonomy of the group

(further details later)

- *The ultimate aim is to evaluate the **maturity** of your approach*

Finally

- **Other hints/request for the prj**
- **FAQs**

Screening phase

Dip. Informatica
University of Pisa

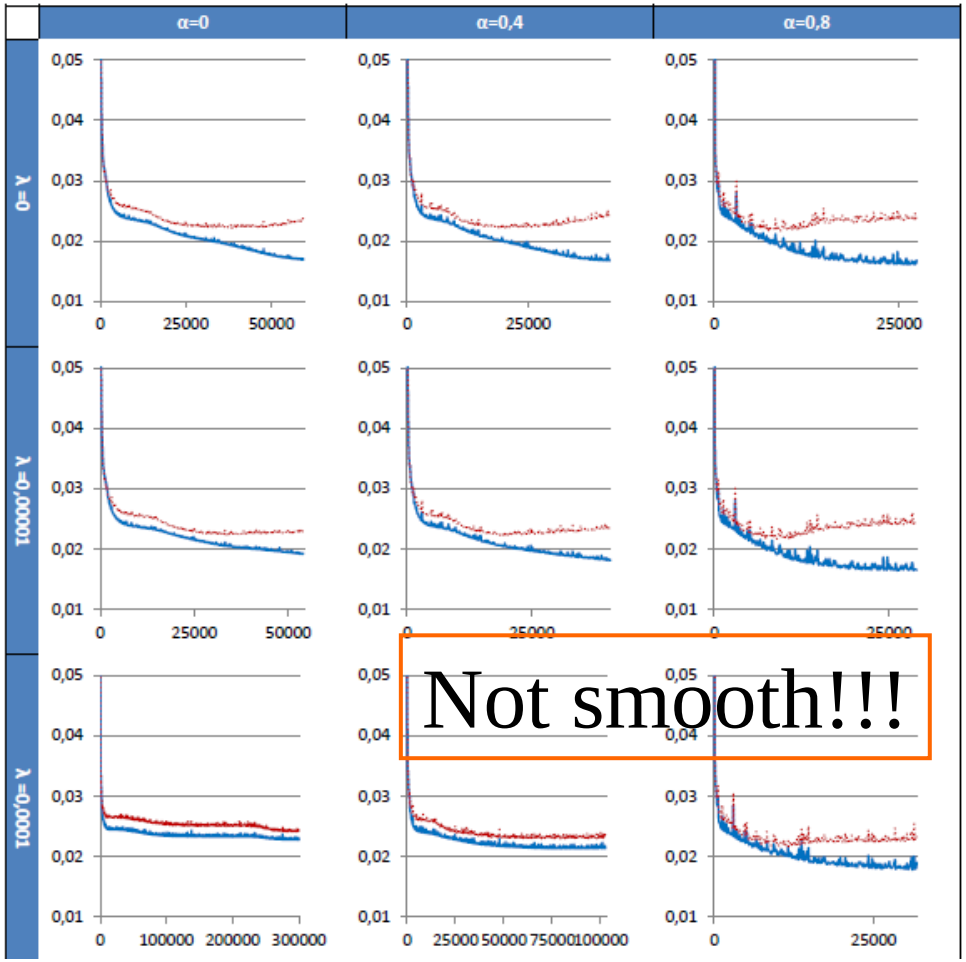
- Initially you will try different values of hyper-parameters
- (if you think interesting) significant cases of learning curves can be reported in the report
- **but for yourself** plot them for various combination*
  - →the screening phase is essential for yourself to learn from this experience [see next slide demo]

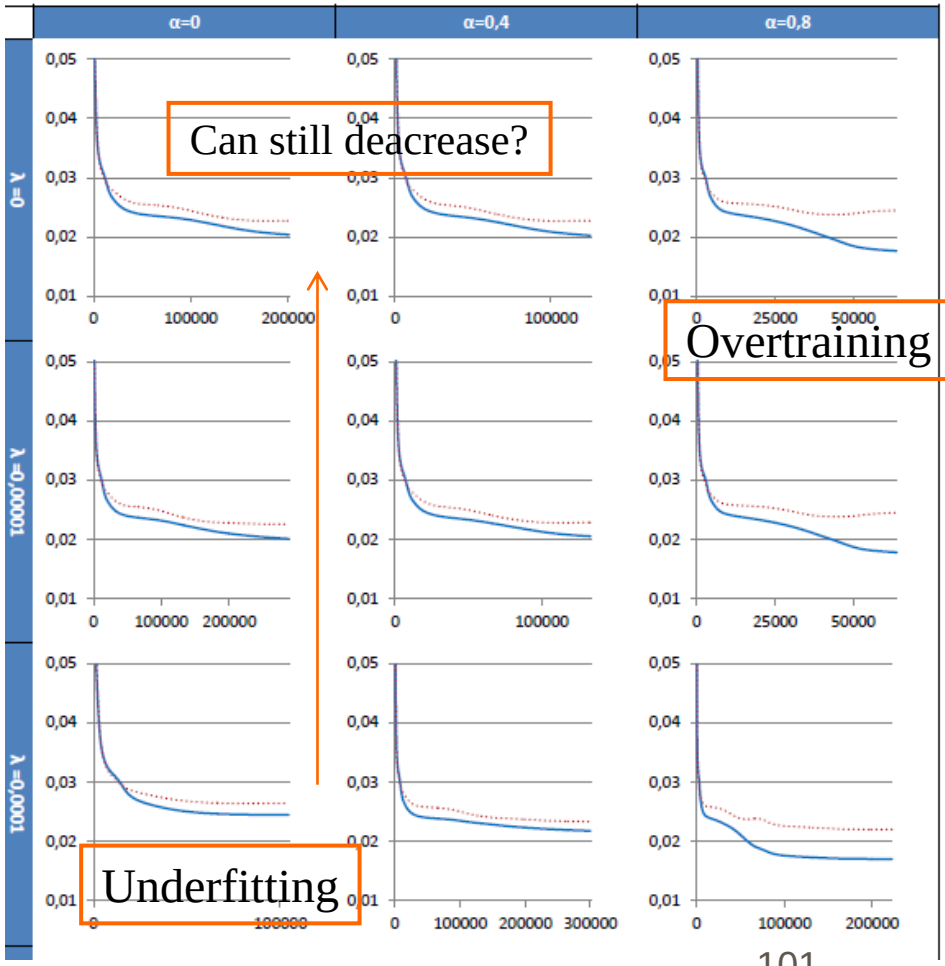
- Results of this kind (also from the grid search) could be part of the appendix

Screening demo

Dip. Informatica
University of Pisa

- 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: further info (I)

Dip. Informatica
University of Pisa

- **Autonomy** (is part of the evaluation)
- Soundness and quality of the (code) simulator (including any characteristics of modularity, efficiency, ...)
  - Note that more efficiency allows you to have more time for experimental part that is typically time demanding
- Proper operation and behavior:
  - e.g. see the regularity of learning behavior on the plots (see the FAQ), …
- **Pertinence of the choice** for the model selection and evaluation
- Extension and/or depth of your investigation
  - Model and/or algorithms variants
  - Hyperparameters range
  - Validation and assessment …(**accuracy/rigorousness of the validation**)

How to evaluate your work: further info (II)

Dip. Informatica
University of Pisa

- **Quality of the report**:

1. completeness (adherence to requests, see the demo & info files), 2. correctness, 3. synthesis, 4. depth.
And in general: organization, rigor, soundness and synthesis (no repetitions); motivations suitability; choices made; breadth and depth of the experimental investigation (although expressed in a concise way); **clarity of the validation schema**.

- In case: difficulty of the challenge (if it is well done) can be a plus (e.g. <A) with CM>)

- CUP: The **blind test** result is just one of the possible parameters (not the most important one): however, very very often it strongly depends on the quality of the model and of the validation approach.

- *The ultimate aim is to evaluate the **maturity** of your approach*.

How many rules?

Dip. Informatica
University of Pisa

Again, everything is **easy**

1. You have a task and dataset
2. You have to apply using your model/models and to report the results
3. You have to respect the deadlines

But… many questions have been raised, so to help you I report the answer to many FAQ

- Please, consider them; the first rule of the FAQ: **read the FAQ**!
  - Please, *don't ask for questions already answered in the FAQ* in these slides or in the Moodle FAQ section (most of the questions that we typically receive can be answered with "Please, read the FAQ").
  - Also avoid to miss the described needs for the project
- Please, to avoid further rules "cluttering" *do not ask other rules* ;-)

FAQ 0 [repetita]

Dip. Informatica
University of Pisa

Typical serious matters to make possible the correction:

- Respect the presentation *time limits* (hence, take care of the number of slides):
  - hence, take care of the number of slides
- Respect the *minimum font*
  - Too small font report will be rejected
- Respect the *format of the CUP results* (and range values): see the file "template-example-with-random-outputs_ML-CUP20-TS.csv"

FAQ I: Deadline [Repetita]

Dip. Informatica
University of Pisa

- **Deadline**: see the beginning slides, **Phase V slides**
  - Check the **exact deadline** for the project material delivery **in large advance** in the Moodle folder for each session, i.e. in Prj Student Material → Session xx (your session of interest)
  - Can be extended**?** No, **later is indeed in advance for the next session** (using the proper folder).
  - It is a strict automatic deadline: please, check it in large advance!
  - I'm sorry, I *cannot* (*) reply to emails asking to move the date (or to extend it for personal issues) of the exam/session or the deadline of the prj delivery.
  - (*) Please, immage the case: If I do it for you, I have to do it for all the candidates (it is also a fairness matter). If you (the group together) cannot come, you simply can come in the next session ("appello"). Please, in case advise me in advance.
  - Thank you in advance for your understanding.

FAQ II (Practical)

Dip. Informatica
University of Pisa

- Is there any **CHECK LIST** for material delivery**?**
  - Previous slides on <Project general rules> and < What to produce> (2 slides)
  - For the **report**: see the demo & info files with [*] for mandatory info on the results
    - ML-2023-PRJ-Slides-Demo* **&** ML-23-Report-info*
  - 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, add it to READ-ME file in the code package)
    - Your written slides report (with results on 3 MONKs and the CUP)
    - *team-name*_ML-CUP23-TS.csv
    - *team-name*_abstract.txt
- Deliver the package only as **zip file**!!!
- Only **1 delivery** for each group, by 1 member of the group
  - Use the **surnames** of the group components for the package file name
  - And for the pdf slide report file

FAQ III (Practical)

Dip. Informatica
University of Pisa

- **How the group communicate with teacher and assistants and for the final project delivery?**
  All the group member **must** be included in cc of the email, no answer will be sent to a singular partner of a group (also to avoid partial info among the group components)

- **Do we need a supercomputer/server/GPU to run the prj?**
  Not at all, this is a small task: you can use any machine or also external services (e.g. Colab) , **but typically your PC/laptops are sufficient** to conclude the prj in a reasonable time (also the constraints on the resources force you to focalize your effort, see the lect.s on validation)

FAQ III (Practical)

Dip. Informatica
University of Pisa

- **CSV** format for the CUP 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-CUP20-TS.csv (which is filled with random-output values for demo)

- Executable programs, libraries, … (**large files**): until the package is small size try to include everything in the package.
  If you have problems include (in the READ-ME file of the code package) a link to download the *** accessories files *** of great dimension
  - Check the zip archive size before submitting it. Please try to stay **within 5 MB**. If it is above 5MB, make sure you've cleaned up notebook outputs or that you haven't included hidden files by mistake. In any case, the limit for the package is 20 MB

- Plots , graphics etc.: it is mandatory to distinguish lines in the plots also using different lines symbols/style (to see them also in Black&White)

FAQ IV

Dip. Informatica
University of Pisa

- How many hidden layers do you suggest for the CUP?
  I strongly suggest you to try also two (or more) and compare (even if a single hidden level can result sufficient)
- How many trials (grid search dimension for the hyperparameters values) ?
  How many you feel are useful in order to do a good model selection. Such choices are part of the quality of your work.
- Which trials should I report? All the significant cases with details (experimental evidences), the other can be mentioned/synthetized 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
- Is "**internal test" result** needed? **YES**, it is, so we can compare your results with the blind test results (for our didactics aims)
  See also the ML-23-Report-info file.
- Does the use of the **(internal) test data for model selection** aims influence our prj evaluation? YES!, using "test" data in any form for the model selection is considered a serious issue

FAQ V

Dip. Informatica
University of Pisa

- What 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
  - Take advantage of the Q&A class (with also the assistant)
  - Help of the assistant
  - But…: Autonomy is part of the evaluation

FAQ VI

Dip. Informatica
University of Pisa

Even more QUESTIONS?

- Please, read the following slides (other FAQ) before the next Q&A lecture!!!!
- They are related to frequent past errors!
- Please, read other FAQ in the Moodle page (I can update them)

FAQ (NOT read in class I)

Dip. Informatica
University of Pisa

Please, read them by yourself :

- [Report] It is important to report the final Training, Validation and the internal Test error by the MEE measure?
  Yes, it is a must. I cannot accept a report without these values.
- [Report] 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 errors like Monk?
  No, zero errors is ONLY for some of the Monks tasks, an exception. For reasons of equity, no indications can be given on the reference performance for the CUP.
- 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 for each point in the plot.
- What if my learning curve is unstable? Try to solve the issue! A smooth learning curve can be obtained for these tasks by a correct implementation and a suitable setting ! Else, I have to consider it as a *not good* result.

FAQ (NOT read in class II)

Dip. Informatica
University of Pisa

Please, read them by yourself :

- Can we use other methods beyond grid search?
  Yes, but first grid search approach is mandatory. It is needed to teach to you the meaning of the hyperparameters! Others? Yes, e.g. Random search : see Validation 1 (Valid1) lecture. But if you like, and only after the result with grid search
- 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.
- 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

FAQ (NOT read in class III)

Dip. Informatica
University of Pisa

Please, read them by yourself:

- Does a NN for **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. Use a very small range! 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? Very few. Follow the manual as baseline: see a slide in this lecture!
- How many units for the CUP? You have to search by yourself, but it is completed unrelated with respect to the Monk task!!!
- What if a use a scaling/normalization of target values for the CUP ? It is not need to do it, if you do it, remember to report the MEE in the **original scale** (for the test result file and the tables on the report). It is a must in order to make possible the evaluation, i.e. the comparison!

FAQ (NOT read in class IV)

Dip. Informatica
University of Pisa

Please, read them by yourself:

- Can we collaborate with other groups for specific issues? We encourage to exchange ideas and to solve some common issues with other students (e.g. is the W update rule correct in my code)? However, of course, the project will be completely independent among different groups, and you must show your autonomy in developing it. We suggest to follow your original path and exploit in autonomy this experience!

- What is the time to be reported when you say «Provide an estimation of the (training) computing time (and of your HW resources)" in the info doc for the report?
  A measure of the time like the mean time (e.g. in second/minutes/hours) for an epoch of training, or better for a full training of the final selected model, and/or a cycle of validation (and the description of the computer that you are using to make this training).

FAQ (NOT read in class V)

Dip. Informatica
University of Pisa

Please, read them by yourself:

- Making a prj with SVM, which plot can I do?
  See E.g. Slide 6 <ML-**-SVM-other inform> lecture (e.g. results w.r.t. hyper-parameters values, including epsilon-tube parameter)
- Project B (or <A with CM>): If the winner is not a NN (or a methods with iterative training) which plot can I do?
  See the answer on SVM above (you can show effects of the hyper-parameters). Moreover, if you are considering a NN in the set of models, even if it is not the winner, you can report the learning curve for the selected NN. Explain the case in the report.
- Prj <A with CM>: comparison also on Monks?
  Using a NN the results on MONK are still needed (at least summarized if the not interesting for the basic or the advanced algorithms), and of course you have to apply to the CUP dataset. For the SVR PRJs (with CM) applying to MONK is *not* required. But see the Q&A on do we need to implement also the NN? (next page) or you can also always use your SVM also in the form of a classifier.

FAQ (NOT read in class VI)

Dip. Informatica
University of Pisa

Please, read them by yourself:

- Making <A with CM> implementing a NN, we implement the basic and advanced NN by ourselves? Yes. It is a type A prj.
  - In the CM report you will discuss the properties of the algorithms from the CM view point.
  - In the ML report you will discuss the "basic" NN and the comparison with the "advanced" algorithm developed with CM.

- Making <A with CM> implementing a SVM, do we need to implement also the NN? Up to you, in this case
  - you can develop the NN by yourself (type A prj) or
  - you can take the NN from a library (like for a type B prj, in this case it is a mix),

  but, in any case, you have to compare the results (for the ML report)
  General note: the slide for <A with CM> said: "More effort? It is because you like to try and show something more and not something less!", hence I strongly suggest to compare also with a NN

FAQ (NOT read in class VII)

Dip. Informatica
University of Pisa

Please, read them by yourself:

Type B Prj:

- Project B: Do we have to compare the models on both MONK and the CUP data sets? **Yes**, you have to compare ALL the models on both the Monk datasets & on the CUP dataset. You can synthetize using only the most interesting results and you can use the appendix for details

- Project B: Do we have to include the code?
  Yes, the code written by yourself, in case, e.g. to use the libraries/notebooks etc., must be included in the package

- Package size:
  "Check the zip archive size before submitting it. Please try to stay **within 5 MB**. If it is above 5MB, make sure you've cleaned up notebook outputs or that you haven't included hidden files by mistake. In any case, the **limit for the package is 20 MB**"

FAQ (NOT read in class VIII)

Dip. Informatica
University of Pisa

- What changes using **MATLAB/OCTAVE/R?** **[repetita]**
  - 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/Deep Learning Toolbox → prj type B

FAQ (NOT read in class IX)

Dip. Informatica
University of Pisa

- Project delivery: For the type B project, because its analysis is more focused on the comparison of different models, for the software part, can we deliver a Jupyter Notebook (at least for the models used with Python library) or do we have to send "pure" Python script?
  - Since this is a project that lends itself to using different libraries and frameworks, I would very well expect a Jupyter or Colab Notebook (if you are using them) rather than pure python code with various requirements to install in order to test it.
  - It's different for project A, where you will use only numerical libraries.
  - But it is good to remember to remove the output of the cells before delivery, to avoid that you also save the images or plots of each run. Just run "Cell -> Output cells -> Clear" before saving the notebook for delivery (to avoid cluttering and huge delivery in terms of space).
- Package size (any kind of project):
  Check the zip archive size before submitting it. **Please try to stay within 5 MB**. If it is above 5MB, make sure you've cleaned up notebook outputs or that you haven't included hidden files by mistake. In any case, the **limit for the package is 20 MB**

Motivational hints by past students

Dip. Informatica
University of Pisa

Selection of general nice hints/motivations by past students reports conclusions (when it was a report):

- "The powerful libraries available nowadays provide a high level of abstraction and while this is useful for quickly getting things done, a naive use of a user with no solid knowledge of **the theoretical ground** may lead to a disastrous outcome. […] In conclusion, by implementing our own library, we feel that we are now much more aware of whats happening behind the scenes."

- "The limited time and computing resources **has pushed us to think** more about reasonable choices or shortcuts to speed up the searching for a fine model selection for the Cup dataset, showing us how necessary is the integration between artificial and human intelligence."

- "Last consideration is on how important is to **visualize results in the best way**. Not only it is fundamental for presentation itself, often it is the fastest way to understand how the model behaves in the hyperparameter space."

- "Lo sviluppo di questo progetto è stato sicuramente un'ottima motivazione per mettere in pratica quanto affrontato nel corso, sia in termini di implementazione e comprensione degli algoritmi alla base delle tecniche più moderne di Machine Learning, sia in termini di applicazione di una rigorosa metodologia per la valutazione e soprattutto la validazione di un modello, indispensabile per raggiungere risultati plausibili e confrontabili con quanto studiato in teoria. **La coordinazione che un lavoro condiviso come questo richiede è inoltre un ottimo 'training test' per lo sviluppo in gruppo**, ed il confronto continuo durante le diverse fasi dell'implementazione e validazione ha benificiato entrambi nella comprensione del lavoro svolto."

- "I **concetti teorici trattati a lezione** di cui fanno parte ad esempio le tecniche di regularization […] e le nozioni di underfitting/overfitting di un modello sui dati sono stati visti in un ambito più pratico e è stato molto utile ai fini didattici. Le richieste ci hanno permesso di scoprire più a fondo il mondo del Machine Learning e in particolare quello delle Neural Network, infatti abbiamo avuto l'occasione di mettere le mani su dinamiche e funzionamenti che dall'esterno risultano trasparenti."

Enjoy !

Dip. Informatica
University of Pisa

- … Now you can **enjoy** with ML !
- To stimulate your initiative, listen "rock music":
  - "La Follia" (Corelli)
  - Il "cimento dell'armonia e dell'invenzione", "L'estro armonico", "La stravaganza" (Vivaldi)

Future – just ahead

Dip. Informatica
University of Pisa

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

2023 THIS PART WILL BE PRESENTED LATER, Just wait

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