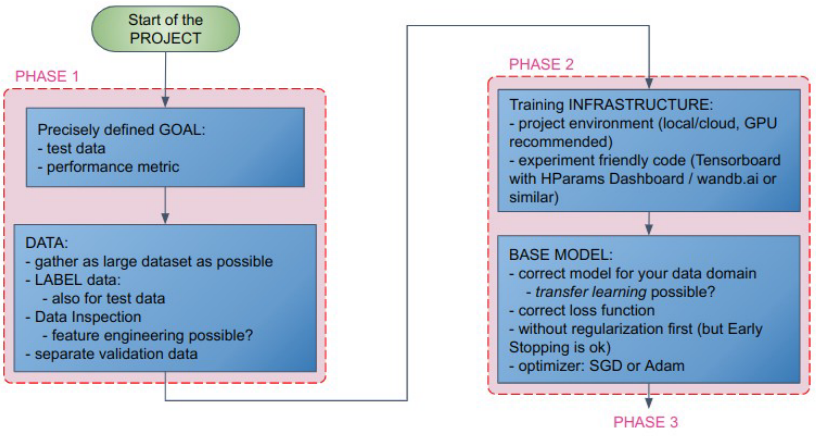
Deep Neural Networks for Data Analysis – Project (ed. 2025/2026)

1. Why?
   a) Learn to use deep learning methods in practice.
   b) In order to pass the course:
      - To pass the course, **both lecture and project** parts have to be passed independently.
      - Project is worth **50 points** in total (50% of the course points).
      - You have to get at least 25 point from project to pass this it.

2. What?
   a) You have to pick a problem/topic, for which deep learning is a suitable approach.
   b) Do not hesitate to choose a topic linked with your Bachelor's Thesis, if applicable. Your motivation for doing a good project will be even greater then.
   c) If you want ideas about project topics, these resources are good starting points:
      - https://paperswithcode.com/datasets
      - https://huggingface.co/datasets
   d) **Consulting your topic proposal** with me during our project hours or via e-mail **is mandatory** (let as call this "PHASE 0" of the project → should be done in first 2 (max. 3) weeks of the semester
   e) When the project is approved, the project group on eNauczanie will be created, which is necessary to see next phases on eNauczanie.

3. How & when?
   a) Project can be made **in groups of two or three** students or **individually**, if you really insist
   b) There are 3 phases of this project, details described below.

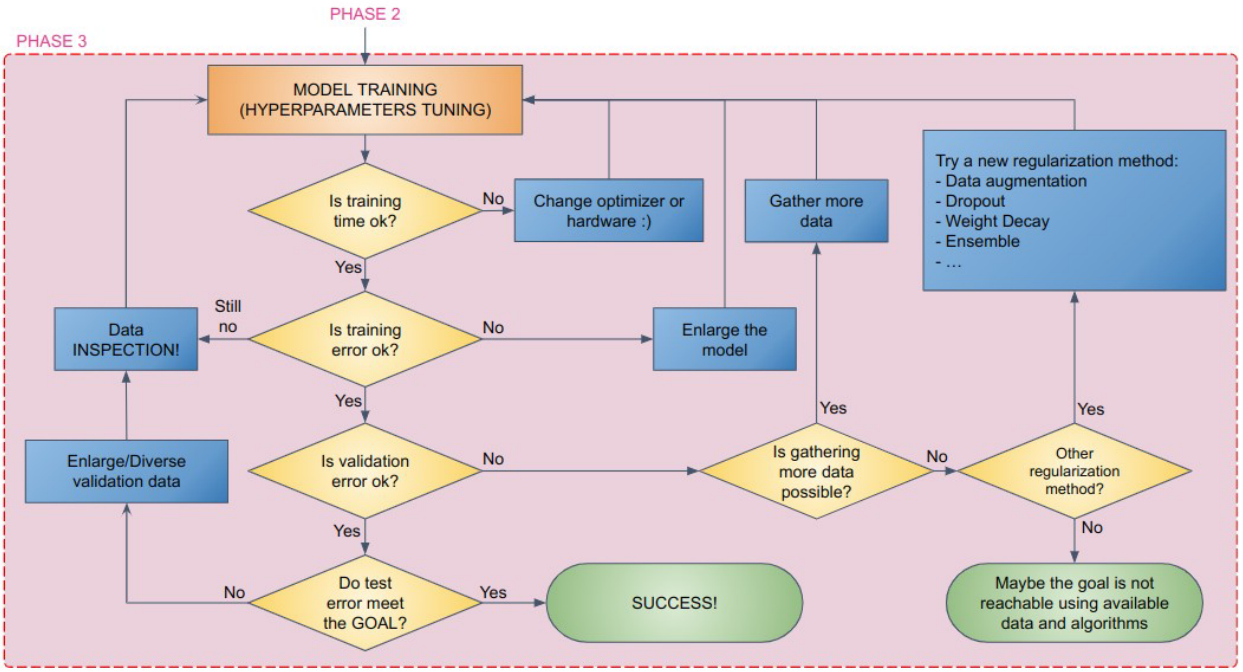


- **Phase 1:**
  - What's there:
    - ✔ Formation of a project group
    - ✔ Specification of a topic/problem
    - ✔ Train/test data collection
  - How to get points (max. 15): do ~5min presentation (on project hours) and upload the slides to eNauczenie site. The short presentation should contain:
    - ✔ Info about students in the group
    - ✔ Precise problem formulation: *[3 pts]*
      - What is the **problem**
      - Evidence of **self collected** real-world **test data** (or, at least (in the worst case): precise idea of the third-party test data source)
      - Objective: what **performance metric** on this test data will be optimized (e.g. categorical accuracy, F1-score, Word Error Rate, etc..)
    - ✔ **Quick review** of existing solutions (if any) to similar problems *[4 pts]*
      - Provide names and short descriptions of algorithms/methods (say, up to 3 sentences per solution) used in solutions to similar problems; do not focus on products/marketing names; focus on technical solutions

        - Provide references to scientific papers describing these solutions
            - you want to read these paper carefully (but maybe done later, after phase 1)
    - ✔ Info about and exploration of external downloaded **training data** *[3 pts]*
        - data statistics (num. of classes, samples, data distribution, characteristics, features etc.)
        - info about data collection and labeling process (if available) and data partition process (train/val or train/val/test splits)
    - ✔ Info about and exploration of manually collected **private test data** *[5 pts]*
        - sometimes, understandably, it is not possible to collect data (e.g. not easy to personally collect a couple of brain scans)
        - data statistics (num. of classes, samples – collect at least a few per class, data distribution, characteristics, features etc.)
        - info about data collection and labeling process (if available)
    - DEADLINE: LOOK MOODLE [-3 points per each week after deadline]

- **Phase 2:**
    - What's there:
        - ✔ Setup of a project environment (hopefully with GPU support) – local or remote (e.g. Google Colab, Kaggle etc.)
        - ✔ Coding the base model and training code – you typically do not need to start from scratch, but clone an existing open-source project
        - ✔ Coding useful training infrastructure (logging results with e.g. Tensorboard, selecting and logging hyperparameters with e.g. Hparams (hyperparameter_tuning_with_hparams) or wandb.ai site or similar)
            - logging should be persistent (to file, not only to output of a notebook cell!)
            - not only raw text logs, plots of training curves (i.e. how e.g. train/val loss changes during training) is important – inspecting it can say a lot to a trained eye
        - ✔ first run training (try to overfit to one train data batch – sanity check that training works)
            - log progress, config setup & results to appropriate files
    - How to get points (max. 15): upload **project code** (but not files with data or model weights - these can be heavy) and **logs from the sanity run** to eNauczenie site. Meeting with the teacher is not needed here.
        - ✔ Results logging support *[2 pts]*
        - ✔ Saving/Checkpointing support *[3 pts]*
        - ✔ Hyperparameters selection / configuration logging *[2 pts]*
        - ✔ Building the model (correct architecture and loss) *[5 pts]*
        - ✔ Logs from a sanity check *[3 pts]*
    - DEADLINE: LOOK MOODLE

- **Phase 3:**



- What's there:
  - ✔ Experimentation that aims to reach the goal of the project (e.g. steps similar to the ones presented in the above chart)
  - ✔ random-search of optimal hyperparameters of training (written from scratch or using a dedicated library, like https://optuna.org/ (or equivalent), visualize the impact of hparams on metrics in Tensorboard/wandb
  - ✔ developing true DL projects requires lots of experiments, so also lots of compute power and compute time; as students you typically don't them. So, as a rule of a thumb, let's say it is sufficient for getting lots of points to do less experiments (than needed to achieve success), if the total compute time for your experiments exceeded 2 GPU-days.
- How to get points (max. 20): upload **final project code**, **logs from the best run** and **final presentation (slides)** to eNauczenie site.
  - ✔ Final code with working demo script/app *[7 pts]*
  - ✔ Logs from the best run *[3 pts]*
  - ✔ Final presentation in front of other students and teacher (~ 10 min): *[10 pts]*
    - Show **the evidence of effort** put into the project (results, logs, descriptions of experiments you tried, how hparams affect performance, conclusions)
    - Live demo (if possible)

  [grading here depends on character and difficulty of the tackled problem]

- DEADLINE: LOOK MOODLE [end of the semester]