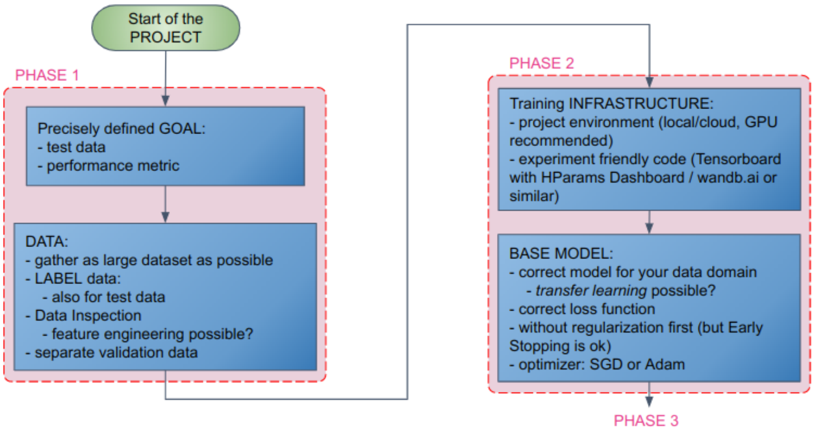
Deep neural networks for data analysis – Project (ed. 2024/2025)

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 Master'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

3. How & when?
   a) Project can be made **in groups of three (preferred) or two** students, if you really insist
   b) A group on e-nauczanie must be created/joined before sending reports
   c) There are 3 phases of this project, details described below.

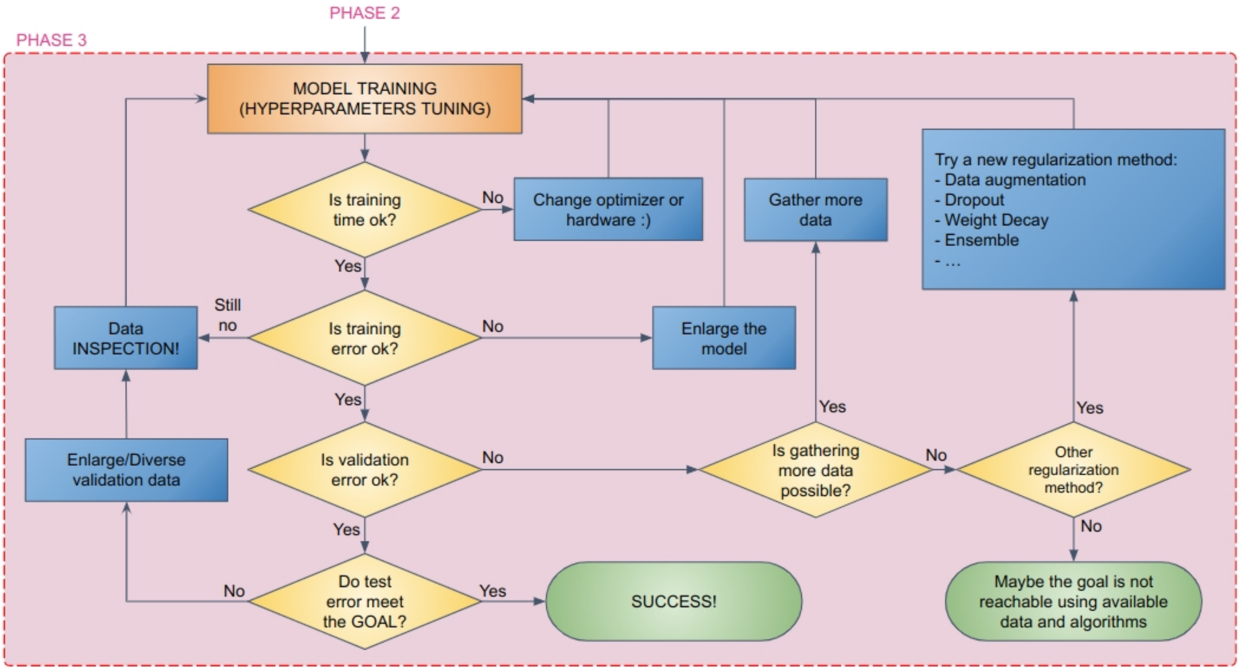

Start of the PROJECT
PHASE 1
Precisely defined GOAL:
- test data
- performance metric
DATA:
- gather as large dataset as possible
- LABEL data:
- also for test data
- Data Inspection
- feature engineering possible?
- separate validation data
PHASE 2
Training INFRASTRUCTURE:
- project environment (local/cloud, GPU recommended)
- experiment friendly code (Tensorboard with HParams Dashboard / wandb.ai or similar)
BASE MODEL:
- correct model for your data domain
- *transfer learning* possible?
- correct loss function
- without regularization first (but Early Stopping is ok)
- optimizer: SGD or Adam
PHASE 3

- **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)
    - ✔ first run training (try to overfit to one train data batch – sanity check that training works), log progress & result
  - How to get points (max. 15): upload **project code** 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:**


PHASE 2
PHASE 3
MODEL TRAINING (HYPERPARAMETERS TUNING)
Is training time ok?
No
Change optimizer or hardware :)
Gather more data
Try a new regularization method:
- Data augmentation
- Dropout
- Weight Decay
- Ensemble
- …
Yes
Data INSPECTION!
Still no
Is training error ok?
No
Enlarge the model
Yes
Enlarge/Diverse validation data
Is validation error ok?
No
Is gathering more data possible?
Yes
No
Other regularization method?
Yes
No
Yes
No
Do test error meet the GOAL?
Yes
SUCCESS!
Maybe the goal is not reachable using available data and algorithms

  - 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)
- ✔ 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 (max. 10 min)** to eNauczenie site.
  - ✔ Final code with working demo script/app *[7 pts]*
  - ✔ Logs from the best run *[3 pts]*
  - ✔ Final presentation: *[10 pts]*
    - Live demo (if possible)
    - Show **the evidence of effort** put into the project (results, logs, descriptions of experiments you tried, conclusions) [grading here depends on character and difficulty of the tackled problem]
- DEADLINE: LOOK MOODLE [end of the semester]