Machine Learning with PyTorch: Projects

Beniamin Bogosel

May 4, 2026

List will be updated weekly.

Contact: `beniamin.bogosel@uav.ro`

⋆ **The project grade will be your Lab Assignment.** The final grade of the course will be an average between the project grade and the exam grade. The project grade will not necessarily be the maximal one if the **live explanation and presentation** are not convincing.

⋆ **If LLMs are used:** you need to describe the process, enumerate the prompts, underline your contribution.

⋆ **Groups of 2 or 3 students/project**: the project will be presented during the course. All members of the groups should intervene. The project grade is shared by all team members.

⋆ **Final Grade formula:** `(Exam + Project)/2 - floor(nb. absences/3)`

1. Consider a data set of 1000 points generated randomly in the square $[-1, 1] \times [1, 1]$ in the plane. Data set should be taken from the notebook: `PointData2D`.

   Additional, non-trivial examples can be constructed (ask before). If more students want to take this kind of exercise, the separation into two classes of points should be done differently for each one. Contact the teacher.

   For each of the following tasks use the protocol discussed in class: divide data into TRAIN DATA/TEST DATA, propose model architecture, train model, illustrate results.

   a) Propose a linear regression model classifying the points into the two classes. What is the expected outcome?

   b) Consider more complex neural network architectures involving nonlinear activation functions. Discuss phenomena of over fitting and under fitting.

   c) Change the optimization algorithm to `Adam`. Do you observe an improvement in the learning speed?

2. Choose a data set from `torchvision` (https://docs.pytorch.org/vision/main/datasets.html).

   a) Propose and test an image classifier neural network based on the work done in the course.

   Contact the course teacher to validate your data set: students should work on different data sets.

   b) Find 10 images not in the data set, convert them into the appropriate format and test your model on them.

3. Using the framework learned in the course on creating Datasets do the following:

   - think of three categories of images: download 100 of each, split them in a ratio 75/25 into train and test sets.
   - put the images in a folder structure like in the example in the course
   - load your images into Data Loaders as indicated in the course

   a) Using a TinyVGG model (same structure as in the course) train the model for 5, 10, 20 epochs and notice in which situation we are: underfitting/overfitting/good performance.

   b) Comment on what strategies can be applied in case the model does not behave as we expected.

4. **Transfer learning**. Redo the framework of the notebook on the topic of transfer learning for a different model found at

   https://docs.pytorch.org/vision/main/models.html

   replacing the classifier with a 3 class classifier (use same data set or a create a different one). Example of models: AlexNet, ResNet18, etc.

   a) Run the training loop for 5, 10, 20 epochs and evaluate the performance of the model.

   b) Compare with the results in the notebooks discussed in class.

   c) Evaluate the model on an image you choose and interpret the results.