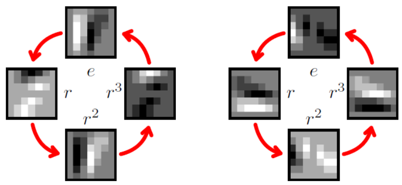
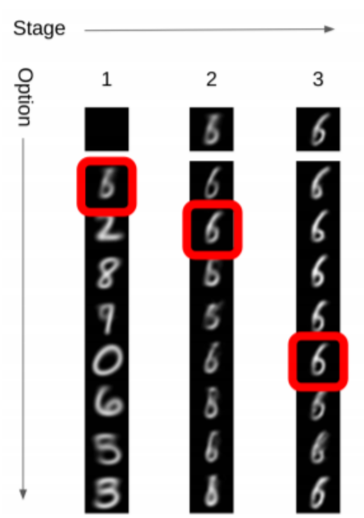
Project 01: Group-equivariant neural networks



Group-equivariant networks are a natural generalization of convolutional neural networks, where translation equivariance is replaced by equivariance with respect to the action of other groups. This gives rise to a more general notion of convolution, that in practice can be directly plugged into existing architectures without too much hassle or fine-tuning. With this project, you will study this family of learning models and: 1) report on their main properties, 2) confirm experimentally their performance, and 3) suggest further generalizations or applications that are not present in the current literature (to the best of your investigation).

Reference: https://arxiv.org/abs/1602.07576

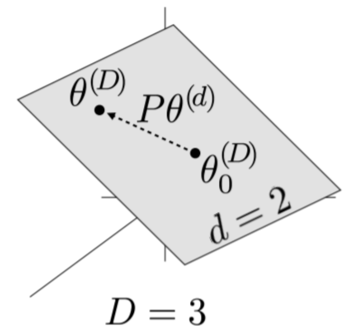
Project 02: Embedding a data distribution into a decision tree



Autoencoders aim to model the data distribution and can be used as generative models or to compress input samples. To accomplish nice generative properties, VAEs introduce a stochastic signal in the training process and make the strong assumption that data is distributed as a multidimensional unimodal gaussian. Yet other assumptions are possible, for instance in VQ-DRAW the data distribution is embedded in the leaves of a decision tree, accomplishing a very effective compression of the input.
With this project, you will study VQ-DRAW and: 1) report on its main properties, 2) test experimentally its performance on data different from images, for instance 3D models and 3) compare the performance to other existing methods using different autoencoder architectures on the same data.

Reference: https://arxiv.org/abs/2003.01599

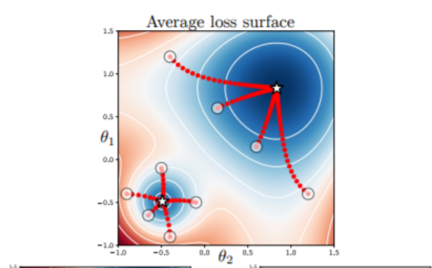
Project 03: Training on parameters subspaces



PCA reduces data dimensionality by projecting each sample to a subspace maximizing variance. In a sense, the effect of PCA pre-processing in a learning problem is equivalent to working with a larger dataset, since it augments the density of our data-space. Similarly, a desirable feat for a learning problem is to work with a smaller hypothesis space, i.e. less training parameters. We can apply the same projection idea to the training parameters K of a neural network, let K = PW, with W being lower dimensional than K, and P being a projection matrix. What is the performance of such a neural network when trained on W with a fixed P? In this project, you will discuss this family of learning models and: 1) report on their main properties compared to a standard smaller neural network, 2) test experimentally their performance for different subspace dimensions, and 3) test experimentally their performance for different families of P.

Reference: https://arxiv.org/abs/1804.08838

Project 04: Learning invariances



Searching for invariances among data collected in different environments is at the basis of modern data-based approaches yearning for out-of-distribution (o.o.d.) generalization. In this project you will explore further two ideas in this direction, Invariant Risk Minimization and the AND-mask, comparing their performance on new kinds of data, such as 3D models. With this project, you will study the problem of o.o.d. generalization and: 1) present the two approaches, highlighting their similarities and differences; 2) compare experimentally their performance on a new kind of data; and 3) suggest further generalizations or applications that are not present in the current literature (to the best of your investigation).

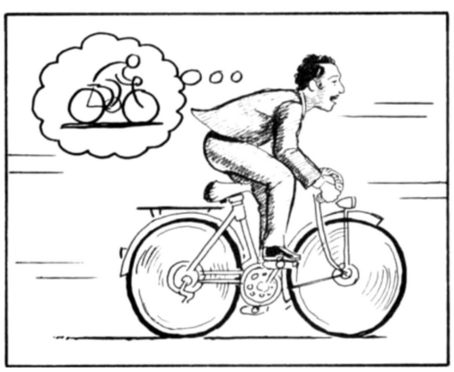
References: https://arxiv.org/abs/2009.00329 https://arxiv.org/abs/1907.02893

Project 05: Geometric interpretability

What kind of features are actually learned in a neural network? Recent investigation has shed some light on this notoriously open question, by proposing computational techniques to actually visualize the prototypical features that trigger the responses of specific neurons in a given net. However, to date, very little work has explored these ideas when dealing with structured data (as opposed to images), such as graphs, 3D shapes, or point clouds. With this exploratory project, you will 1) investigate the applicability of existing feature visualization techniques to geometric or graph data, 2) report your findings, and 3) suggest preliminaries lines of attack and elaborate on the main differences with the classical image-based setting.

Reference: https://distill.pub/2017/feature-visualization/ and references therein.

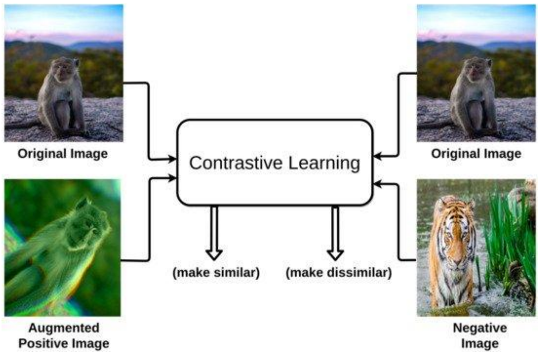
Project 06: World Models



The idea behind model-based reinforcement learning is that modeling the entire environment can be a shortcut to learn faster good policies, where with faster we mean with less actual real world simulations. This idea is extensively and nicely presented in the paper World Models. With this project, you will study World Models and: 1) report on the main ideas behind model based RL with respect to standard RL; 2) test experimentally the performance of model based RL on a new environment, for instance one of the OpenAI PROCGEN games; and 3) discuss the pros and cons of model-based RL and possible evolutions of this paradigm (to the best of your investigation).

References: https://worldmodels.github.io/

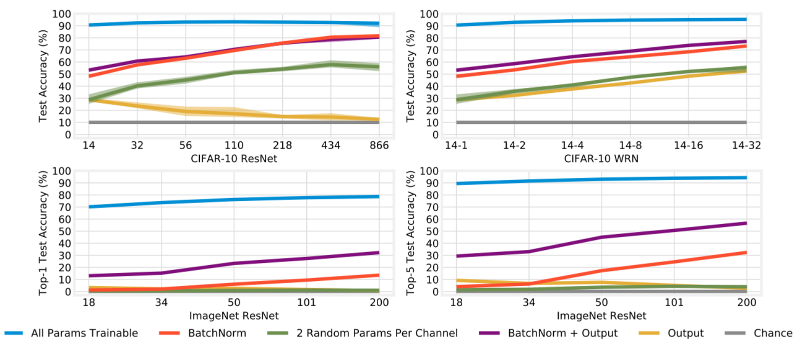
Project 07: Contrastive Learning



Contrastive learning is a machine learning technique used to learn representations without supervision. This technique can be used as a pre-training step to exploit large amounts of unlabeled data. In this project you will study this technique and: 1) report on the main ideas and state of the art of contrastive techniques, 2) implement a contrastive learning regimen for a non-Euclidean data type of your choice (e.g. 3D shapes) and 3) test experimentally the performance on a task of your choice.

References: https://www.mdpi.com/2227-7080/9/1/2, https://arxiv.org/abs/2006.10029

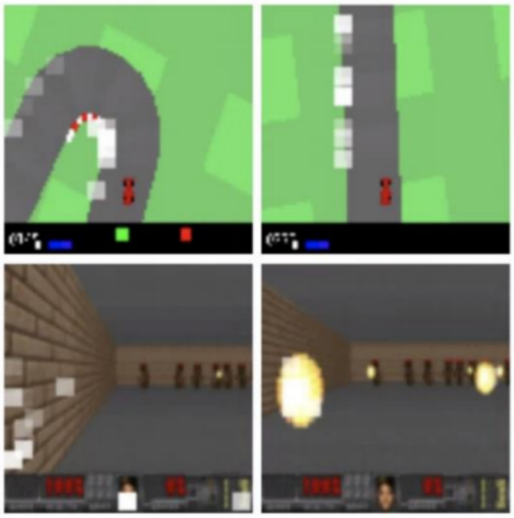
Project 08: Training BatchNorm and only BatchNorm



A CNN with weights frozen at their random original value and trained only on the beta and gamma parameters of the batch-normalization can achieve surprisingly good results on image classification problems, much better than training an equivalent number of weights chosen at random. With this project, you will investigate further this idea and: 1) discuss why this happens, 2) test experimentally this result on other architectures and other kinds of data, like MLPs or GNN, and 3) suggest further generalizations or applications that are not present in the current literature (to the best of your investigation).

Reference: https://arxiv.org/abs/2003.00152

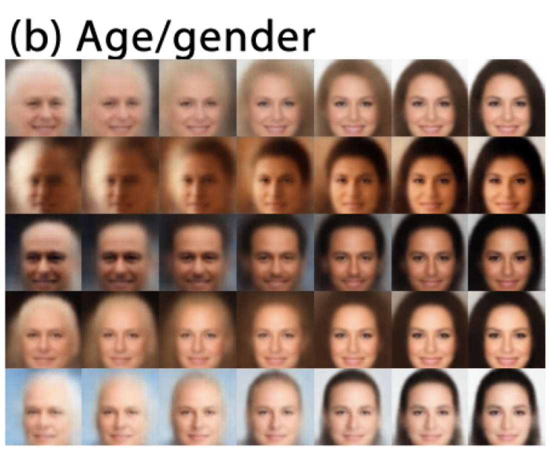
Project 09: Neuroevolution of Self-Interpretable Agents



Sometimes less is more, even in Reinforcement Learning. Agents capable of seeing only a portion of the pixel inputs from the environment not only can solve the tasks with simpler models, but also achieve better generalization. In *Neuroevolution of Self-Interpretable Agents* the portion to look is learned through an attention mechanism.
With this project, you will investigate this idea and: 1) discuss the benefits of attention in RL, 2) confirm experimentally this idea on a new environment, for instance one of the [OpenAI PROCGEN games](); and 3) test further ideas in this direction, for instance what if we choose a feature different from the patch position?

Reference: https://attentionagent.github.io/

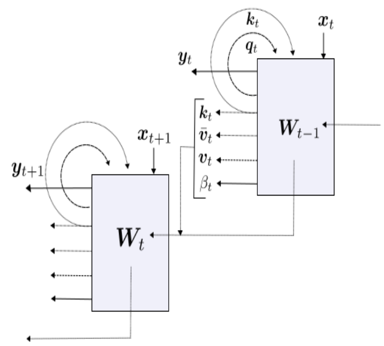
Project 10: Geometric disentanglement



Discovering factors of variation in the data is an outstanding problem in machine learning, and is typically approached as a so-called "disentanglement" problem. For example, imagine you have trained your VAE on a collection of images, and want to find out what specific dimensions of the latent code are responsible for the background color. This is already a challenging problem per se -- how about doing it without supervision? And what about geometric data, like graphs or point clouds? With this project, you will explore disentanglement techniques on non-Euclidean data, possibly by extending an existing technique or making up your own.

Reference: https://openreview.net/forum?id=Sy2fzU9gl

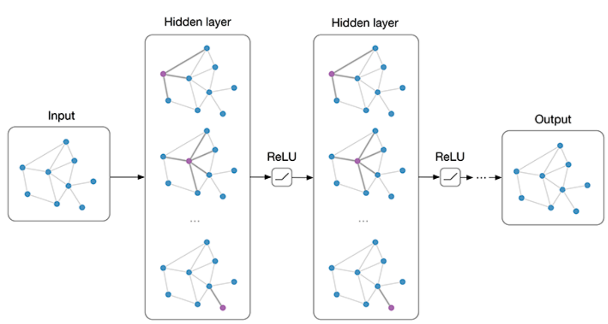
Project 11: Self-Modifying Weight Matrices



The weight matrix (WM) of a neural network (NN) can be considered its program. Since this is usually learned via gradient descent and then remains fixed, NNs also stop learning after training. The WM of a self-referential NN, however, can keep rapidly modifying all of itself during runtime. In principle, such NNs can meta-learn to learn, and meta-meta-learn to meta-learn to learn, and so on, in the sense of recursive self-improvement.
With this project, you will investigate this idea and: 1) discuss its potential and limitations, 2) employ this idea for non-euclidean data, e.g. graphs.

Reference: https://arxiv.org/abs/2202.05780

Project 12: Data Augmentation for Few-Shot Learning on Graphs



Data augmentation has often been used to improve the robustness and accuracy of machine learning models. In Few-Shot Learning, one is usually given a set of base classes for which enough data are available and a set of novel classes for which only a handful of samples are given. Data augmentation in this case can be used to enrich the small datasets for the novel classes, or in a less explored way, to enforce generalization during the training over the base classes. Given the rise of graph machine learning, many graph-tailored data augmentation techniques have recently been proposed. However, these have mostly been considered for data-abundant cases, leaving the question of which data augmentation technique makes the most sense when only a few samples are available. In this exploratory project, you will 1) investigate the current data augmentation techniques for graph-structured data, 2) analyze their benefits and limitations, and 3) implement and compare the techniques that you find most promising for the task of few-shot graph classification or molecular property prediction.

References: https://arxiv.org/abs/2202.08871, https://dl.acm.org/doi/abs/10.1145/3340531.3411951, https://arxiv.org/pdf/2102.07916.pdf

---

Project X: Choose your own

If none of the above whets your appetite, you can propose your own idea for a project. However, your proposal must be approved by us.

In order to propose a new project, you are required to write a well motivated 1-page description (references can be put on a second page). This should include references, describe its feasibility and data availability, describe the expected outcome, the possible risks, what is your intended line of attack, and state in what way it tackles an interesting and new problem in the field of deep learning.

The choose-your-own project must contain aspects of originality. Simply reproducing the results of existing work is not sufficient and will not be approved. Submitting projects from previous exams is also not allowed.

If you choose this path, please send your proposal to all these addresses:
rodola@di.uniroma1.it, moschella@di.uniroma1.it, crisostomi@di.uniroma1.it .

Please use the subject "[DLAI22 projects] Project name".