Symbolic and Evolutionary Artificial Intelligence

List of Projects (1/2)

Marco Cococcioni's proposals

a.y. 2024-2025

Projects mainly supervised by Marco Cococcioni [2025]

C01: Implement the training of RBF networks in PyTorch. Train them using backpropagation, by exploiting PyTorch automatic differentiation facility. Then replace the MSE loss with the minimization of the MAE (Maximum Absolute Error). Use the log of the sum of exponents trick, using a large α.

C02: Study and test the Flow Matching algorithm. Then make comparison with 2024 Grillea&Canzoneri SEAI project (they studied a way to train an auto-encoder in the presence of latent variables having heavy-tails).

C03: Implement the inference in CNNs in python, using numpy, and avoid inner loops by using the im2col approach. Then extend the technique to the training process.

C04: Implement the inference of a Vanilla RNN in Python, using numpy, and by unrolling the network. Then vectorize the code. Then extend it to perform the training.

C05: Perform the inference of a trained DNN directly over encrypted data, without using decryption. The idea is to exploit some properties of encryption schemes, such as the CKKs one. [Privacy-Preserving Neural Networks]

C06: Verify properties of ReLU-based MLPs, by using the Reluplex algorithm. Reluplex is a variant of the simplex algorithm, that works with tropical algebra.

Projects mainly supervised by Marco Cococcioni [2025]

C07: Exploit the Knowledge Distillation technique to approximate a complex MLP (*teacher network*), that uses LeakyReLU/ELU, with a simpler MLP (fewer hidden layers), called *student network*, and based on ReLU.

C08: Implement the Priority Levels version of PAES, to solve Mixed Pareto-Lexicographic problems.

C09: Use PINNs (Physics Informed Neural Networks) for surface reconstruction from unorganized point clouds.

C10: Synthesize in SystemVerilog the inference phase of an MLP. Then do the same for the training phase.

C11: Translate from Matlab to Python (using numpy only, no PyTorch) all the script seen during the CIDL course (Cococcioni's part only).

C12: Investigate Multi-Task Learning to solve a multi-task problem using a single DNN.

C13: Investigate Continual Learning to solve a problem where the number of classes changes over time.

C14: Investigate the various techniques used for **zero shot** and **few shot** learning

C15: Test the effectiveness of Simulated Bifurcation [Toshiba] for solving Ising problems

Symbolic and Evolutionary Artificial Intelligence

List of Projects (2/2)

Lorenzo Fiaschi's proposals

a.y. 2024-2025

Projects mainly supervised by Lorenzo Fiaschi [2025]

F1: Non-standard RL for prioritized multi-objective problems: lunar lander

F2: Non-standard RL for prioritized multi-objective problems: autonomous driving

F3: Non-standard RL for prioritized multi-objective problems: pong

F4: Non-standard policy gradient for prioritized multi-objective RL: theory & algorithms

F5: Advanced RL: didactic implementation of PPO and soft-AC

F6: Advanced RL: didactic implementation of UCT applied to mini chess-engine

F7: Non-standard deep learning for prioritized learning: abstention

Projects mainly supervised by Lorenzo Fiaschi [2025]

F8: Non-standard deep learning for prioritized learning: hierarchical classification

F9: Non-standard deep learning for prioritized learning: semi-supervised problems

F10: Non-standard lexicographic transformers: theory & algorithms

F11: Improved non-standard projection operator: theory & algorithms

F12: Non-standard SVMs for one-class classification: a stable NS-IPM

F13: B&B MILP for prioritized constraints scheduling of basketball training

F14: CP MILP for prioritized constraints scheduling of basketball training

F15: Equality constrained QP: theory & algorithm of the closed-form solution