ITMO University

Methods of Machine Learning

Course Workload		Assessment form (examination/ graded test/ ungraded test)
ECTS	Hours	
6	216	Exam

This course aims to provide an introduction to classical machine learning (ML) and the state-of-the-art knowledge of new approaches based on new physical principles. The goal is to become familiar with applications of ML in physics and how physics can improve hardware for ML. First objective is to form a strong mathematical foundation for ML, get skills in coding ML algorithms, data acquisition and analysis of the results. The next objective is to learn problems that can be solved with ML and are faced by physicists, and get experience in application of underlying methods. Finally, new approaches to ML and new hardware for ML will be reviewed, the students will get experience in simulation of some of the hardware including quantum computing and quantum ML. Expected learning outcomes: After the course you will have a base in ML, get skills in the solution of real-world problems using ML, be familiar with principles and be used to simulation of artificial neural networks (ANN) and unconventional ANN including quantum networks. You will understand the modern techniques in ML and directions of further development both in theory of alternative computing and in implementation of ML using new physical principles. Relevance, novelty, significance, and uniqueness: The course targets understanding of state-of-the-art approaches to machine learning both doing simulation in classical computing and using specialized hardware utilizing new physical principles. The course combines modern results in physics with recent advances in artificial intelligence. It has a unique and specialized curriculum that is based on contemporary research, giving access to knowledge available only in the best research centers. The course teaches the way computing will be performed tomorrow.

Course structure:

1. Classical machine learning

1.1. Linear algebra and matrix methods
1.2. Probability theory. Introduction to Information theory
1.3. Tasks of machine learning. Supervised learning. Maximum likelihood estimation
1.4. Artificial neural networks. Automatic differentiation. Gradient based learning
1.5. Optimization methods. Stochastic optimizers
1.6. Regression. Capacity, over- and underfitting. Hyperparameters and validation sets
1.7. Support vector machines. Random decision forest. Principal Component Analysis
1.8. Bayesian statistics
1.9. Regularization. Parameter norm penalties. Dataset augmentation. Early stopping. Sparse representation. Bagging
1.10. Unsupervised learning. Clustering
1.11. Generative models. Generative adversarial network
1.12. Convolutional networks. Recurrent neural networks. Autoencoders
1.13. Reinforcement learning. Monte Carlo Methods

2. Quantum Computing and Quantum Machine Learning

2.1. Bits and qubits. Dirac notation. Entanglement. Quantum measurement. Classical computing and reversible computing. Quantum gates
2.2. Physical platforms for quantum computing. Quantum software infrastructure and simulation tools. QuASM. Quantum packages from IBM (qiskit)
2.3. Implementing quantum gates, measurements and circuits using one of the existing quantum packages
2.4. Quantum teleportation and Bells states. Deutsch algorithms and Deutsch–Jozsa algorithm
2.5. Quantum complexity. Simon's algorithm. Quantum phase estimation
2.6. Factoring and Shor's algorithm. Quantum search and Grover's algorithm
2.7. Current hardware for quantum computing, NISQ vs. FTQC protocols. Hybrid quantum-classical workflow. Variational quantum eigensolver (VQE)
2.8. Quantum annealing
2.9. Classical optimization and gradient descent. Variational ansatz
2.10. Variational quantum algorithms for linear algebra. Automatic differentiation and analytical gradients
2.11. Quantum approximate optimization algorithm and QAOA-type quantum ansatz. Embedding of NP-hard problems and financial modeling. MaxCut and unsupervised machine learning (clustering)
2.12. Open problems of Quantum Machine Learning (Input, Output, Costing, and Benchmarking problems). Perspectives
2.13. Coding one of the QML approaches as a final project. Presenting code and benchmarking