Machine Learning and AI for Quantum Many-body Physics

Lavoisier Wah[1,2,3]

[1] *Max Planck Institute for the Science of Light, 91058 Erlangen, Germany*
[2] *Department of Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany*
[3] *Vice President, Diaspora Physique Cameroun, 3 Rue Auguste Rodin, 91120 Palaiseau, France*

February 11, 2026

Email: lavoisier.wahkenounouh@mpl.mpg.de

Abstract

This lecture series is intended for any machine learning (ML) or artificial intelligence (AI) enthusiast, regardless of background or prior knowledge, provided they have minimal coding experience. The goal is to familiarize students with ML tools that can be applied to physical problems. In particular, we will explore how neural networks can be used to represent quantum many-body states and how such representations allow us to extract physical properties such as ground-state energies and quantum phase transitions. We will also employ methods such as exact diagonalization, which will serve as a benchmark for our neural-network-based quantum states. While no prior knowledge of ML is required, students should be comfortable with quantum mechanics, including—but not limited to—Dirac notation, the Schrödinger equation, and the variational principle. Part of this lecture will be adapted from Florian Marquardt's course *"Machine Learning for Physicists: Neural Networks and Their Application"*. The lecture will be delivered in English, and each student is expected to have a laptop with stable internet connection.

1 Overview

The lecture will consist of **7** lecture sessions and **6** tutorials, for a total of **28**h, structured as follows:

- **Session 0:** Introduction (exceptionally in French)
- **Lecture 1:** Fundamentals of machine learning
- **Tutorial 1:** Constructing a neural network from scratch
- **Lecture 2:** Training and analyzing a neural network
- **Tutorial 2:** Training a neural network
- **Lecture 3:** Exact diagonalization: sparse methods and `NetKet`
- **Tutorial 3:** Performing exact diagonalization on an Ising chain and extracting physical observables
- **Lecture 4:** Multilayer perceptrons
- **Tutorial 4:** Building a neural quantum state — Part 1
- **Lecture 5:** Boltzmann machines
- **Tutorial 5:** Building a neural quantum state — Part 2
- **Lecture 6:** Recurrent neural networks
- **Lecture 7:** Transfer learning
- **Tutorial 6:** Building a neural quantum state — Part 3

2 Lectures & Tutorials

Session 0: Introduction

This session serves as both the setup phase and an opportunity to get to know one another. During this session, we will install all required packages, the appropriate Python version, and the Python interpreter. Students will also need to create a **GitHub** account at `https://github.com` to access all notebooks and course material.

The following should be completed prior to this session:

- A stable internet connection
- Download the Python-friendly editor **VS Code** (`https://code.visualstudio.com/download`) do not install it yet!
- Create a **GitHub** account at `https://github.com` (optional)
- Download `Python 3.11.5` but do not install it yet!
- Students with a stable internet connection (during the whole lecture series) but less powerful laptops may run all simulations remotely on `https://colab.google` and therefore do not need to install anything locally.

Note: A stable internet connection is required during this session in order to install all additional dependencies!

This lecture is also devoted to a concise introduction to basic Python packages, including `NumPy`, `SciPy`, and `Matplotlib`.

Lecture 1: Fundamentals of machine learning

Introduction to key concepts of machine learning, including supervised and unsupervised learning. Overview of common architectures such as feedforward neural networks. Basic Python packages like `NumPy`, `SciPy`, and `Matplotlib` will be demonstrated.

Tutorial 1: Constructing a neural network from scratch

Hands-on session to build a simple neural network without external libraries. Students will implement forward and backward propagation in Python. The focus is on understanding the internal structure and operations of a neural network.

Lecture 2: Training and analyzing a neural network

Covers optimization techniques, loss functions, and evaluation metrics. Students learn how to train networks effectively and interpret their performance. Practical examples with Python will illustrate model analysis and debugging.

Tutorial 2: Training a neural network

Students train a neural network on a sample dataset using Python. Techniques such as gradient descent and mini-batching are applied. Emphasis on visualizing training dynamics and understanding convergence behavior.

Lecture 3: Exact diagonalization: sparse methods and `NetKet`

Introduction to exact diagonalization for quantum systems. Sparse matrix techniques and the `NetKet` library will be presented. Applications to small spin chains and benchmarking methods are discussed.

Tutorial 3: Performing exact diagonalization on an Ising chain and extracting physical observables

Hands-on implementation of exact diagonalization on an Ising chain. Students compute eigenvalues, eigenvectors, and observables. Results are compared to theoretical predictions and analyzed.

Lecture 4: Multilayer perceptrons

Covers feedforward neural networks with multiple hidden layers. Discussion on activation functions, network depth, and representational power. Applications to approximating quantum many-body states are introduced.

Tutorial 4: Building a neural quantum state — Part 1

Students implement a neural network to represent a quantum wavefunction. Focus on constructing the network architecture and encoding spin configurations. Initial training setup is demonstrated.

Lecture 5: Boltzmann machines

Introduction to energy-based models, including Restricted Boltzmann Machines (RBMs). Theory behind probabilistic neural networks and their applications to physics. Connections to neural quantum states are discussed.

Tutorial 5: Building a neural quantum state — Part 2

Hands-on implementation of RBM-based quantum states. Students learn how to train the network to approximate the ground state. Methods to evaluate accuracy and physical properties are demonstrated.

Lecture 6: Recurrent neural networks

Covers architectures suited for sequential data, such as RNNs and LSTMs. Emphasis on autoregressive models for representing quantum states. Examples and potential advantages over other architectures are highlighted.

Lecture 7: Transfer learning

Introduction to transfer learning concepts in machine learning. Applications to neural quantum states, including reusing pre-trained networks for new problems. Benefits in efficiency and accuracy are discussed.

Tutorial 6: Building a neural quantum state — Part 3

Final hands-on session integrating previous lessons. Students refine and train neural quantum states using RNNs and transfer learning. Comparison with exact diagonalization results and performance evaluation are performed.

Mini-project: Recover the results of Ref. [2] and/or Ref. [1] (optional).

3 Additional Informations

The lecture will be held online at the University of Dschang once per week, starting in **January 2026** (exact dates to be determined), and will be supervised on-site by *Prof. Christian Sadem.*

Location: Condensed Matter Laboratory, **B1.5**, Physics Department, University of Dschang, Cameroon.

References

[1] Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, Roger G. Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Phys. Rev. Res.*, 2:023358, Jun 2020.

[2] Lavoisier Wah, Remmy Zen, and Flore K Kunst. Many-body neural network wavefunction for a non-hermitian ising chain. *arXiv preprint arXiv:2506.11222*, 2025.