Machine learning application

Objective: The purpose of this module is to demonstrate the concept of machine learning through a classical problem in quantum physics: solving the Schrödinger equation for one-dimensional potential wells using a neural network. To illustrate this concept, a pedagogical neural network is trained on a dataset of various potential energy profiles derived from polynomials for which numerical solutions for the wave functions in the ground state or other bound states can be obtained. After training, the model is then used to predict wave functions for previously unseen potentials, showcasing the ability of machine learning to generalize and approximate solutions to complex physical systems.

Activity with the interface.
https://qtechedu.centralesupelec.fr/EN/ex3.html

1. Introduction

For better understanding of the problem presented in this module, you are invited to answer the following theoretical questions in:

Quantum mechanics:

1. One-dimensional quantum mechanical systems are determined by their potential functions $V(x)$. What are some examples of these potentials? Are all of them analytically solvable, or do some of them require numerical methods?
2. What are some numerical methods for solving the Schrödinger equation?
3. When we consider potentials of polynomial type $V(x)=\sum_{i=0}^{d} \alpha_i x^i$, we require that $\alpha_d \in \mathbb{R}>0$. Why is this necessary? How does whether a system satisfies this requirement influence its behavior, particularly its energy spectrum?

Neural networks:

1. Architecture of neural networks is vaguely inspired by biological neural networks. What is an artificial neuron, and what type of mathematical function does it represent? What are neural network layers?
2. The time when one ought to write one's own implementation of neural network architecture using one's chosen programming language has long passed, as there are many off-the-shelf solutions for that. Could you mention some?
3. In the regression task, like the prediction of a wave function, the type of loss function that one chooses is the mean squared error (MSE). What is the interpretation of the mean squared error in terms of minimizing log-likelihood? What distribution of it assumes that the parameters are drawn from? Is it the same distribution from which we draw parameters α?
4. When training a neural network, one uses both training and validation datasets. What it the purpose behind it? What is the approximate shape of the training data loss as a function of the epoch, and what is the one for the validation data loss?

Problem:

This module focuses on solving the Schrödinger equation using a neural network model (specifically, a multi-layer perceptron). To predict the values of the wave function for arbitrary potential we need to train a machine learning model using a dataset of potentials for which we can obtain numerical (or analytical) solutions to the wave functions (target data).

This dataset consists of **input-output** pairs that the model will learn from. In other words, the relationship between a given potential and the resulting wave function will be defined based on these input-output pairs. For simplicity, we generate a set of potentials using a polynomial. Thus:

- **Input features**: Potentials represented by a polynomial.
- **Output labels**: The corresponding wave functions – numerical solutions to the Schrödinger equation for each potential.

2. In the section "**Training of the Multi-Layer Perceptron"**, define the parameters of the dataset and the hyperparameters of the neural network model.

In the left panel of the web interface, choose the parameters for generating the dataset and fill in the table below (please use the interface and the example provided in the table).

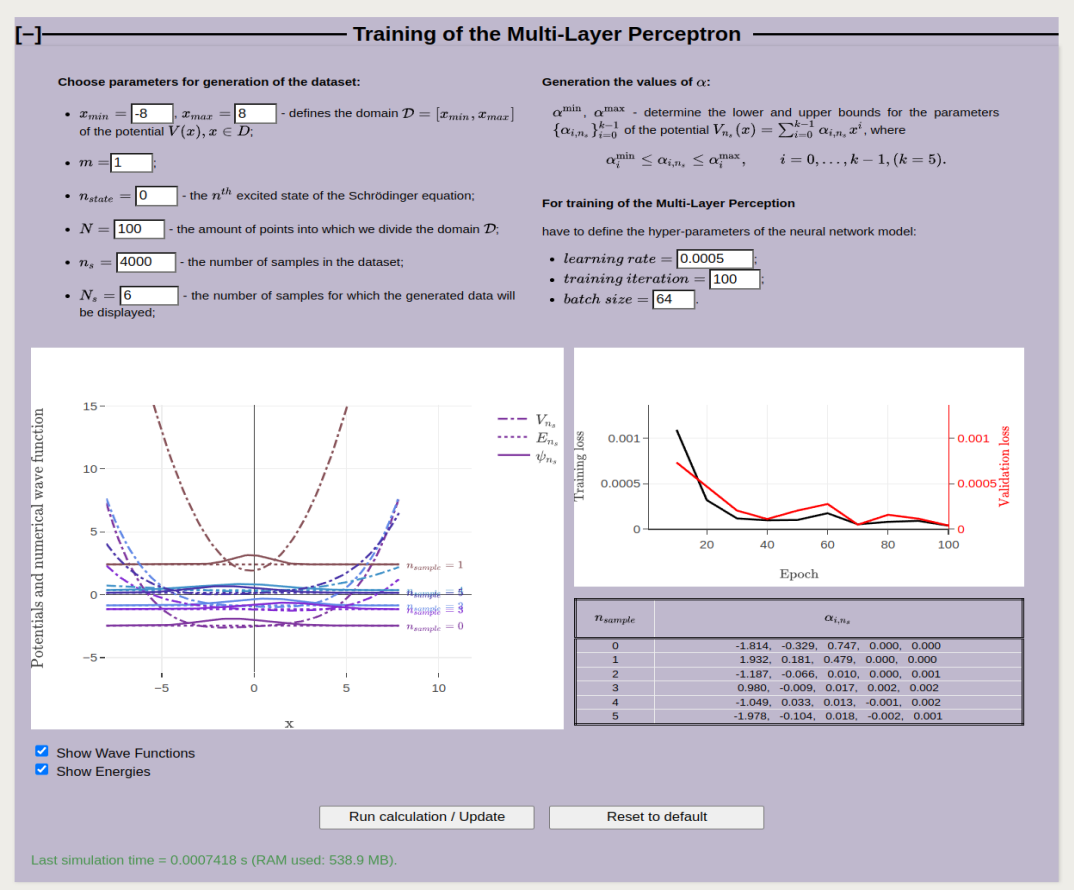
Parameters	Description of the dataset parameters
X_{min} = -8 x_{max} = 8	define the domain $\mathscr{D} = [x_{min}, x_{min}]$ of the potential $V(x)$, $V(x) \in \mathscr{D}$. Minimum and maximum x determines the potential $V(x)$, recommended intervals for x_{min} and x_{max} are [-10, 0] and [0,10] , respectively.
m = 1	
n_{state} =	
N =	
n_s =	
N_s =	

In the right of the web-interface define the hyper-parameters of the neural network model and fill the table below:

Parameter	Description of the neural network parameters
Learning rate =	
Training iterations =	
Batch size =	

Once the dataset is complete - consisting of generated inputs (potentials) and outputs (wave functions) - it is split into two parts:

- **Training set**: Used to train the neural network. The model learns by minimizing the error between its predictions and the actual wave functions.
- **Validation set**: Used to evaluate the model after training. It helps determine whether the model has overfitted (memorized the training data) or generalizes well to unseen data.



To run the calculation, use the ‘’Run calculation” botton. To restore the default values, click the “Reset to default” button. In this module, Atomic units are used*.

Attention! Training the system may take some time (about 15 minutes), depending on the selected parameters of the dataset and the neural network model.

The **left figure** visualizes the randomly chosen polynomial potentials generated during the dataset construction, along with their corresponding wave functions.

Analyze the parameters and the coefficients α used for generating the set of polynomial potentials. Note the key ideas below:

During training, the machine learning algorithm (model) uses the wave functions values obtained by solving the Schrödinger equation for corresponding potential as the "correct answers" to guide learning.

The neural network is trained to **minimize the loss function**, which we choose as the **mean squared error (MSE)** between the actual output values and the predicted ones.

The **right figure** displays evaluation metrics, including **training loss** and **validation loss**, which characterize the results of the training process.

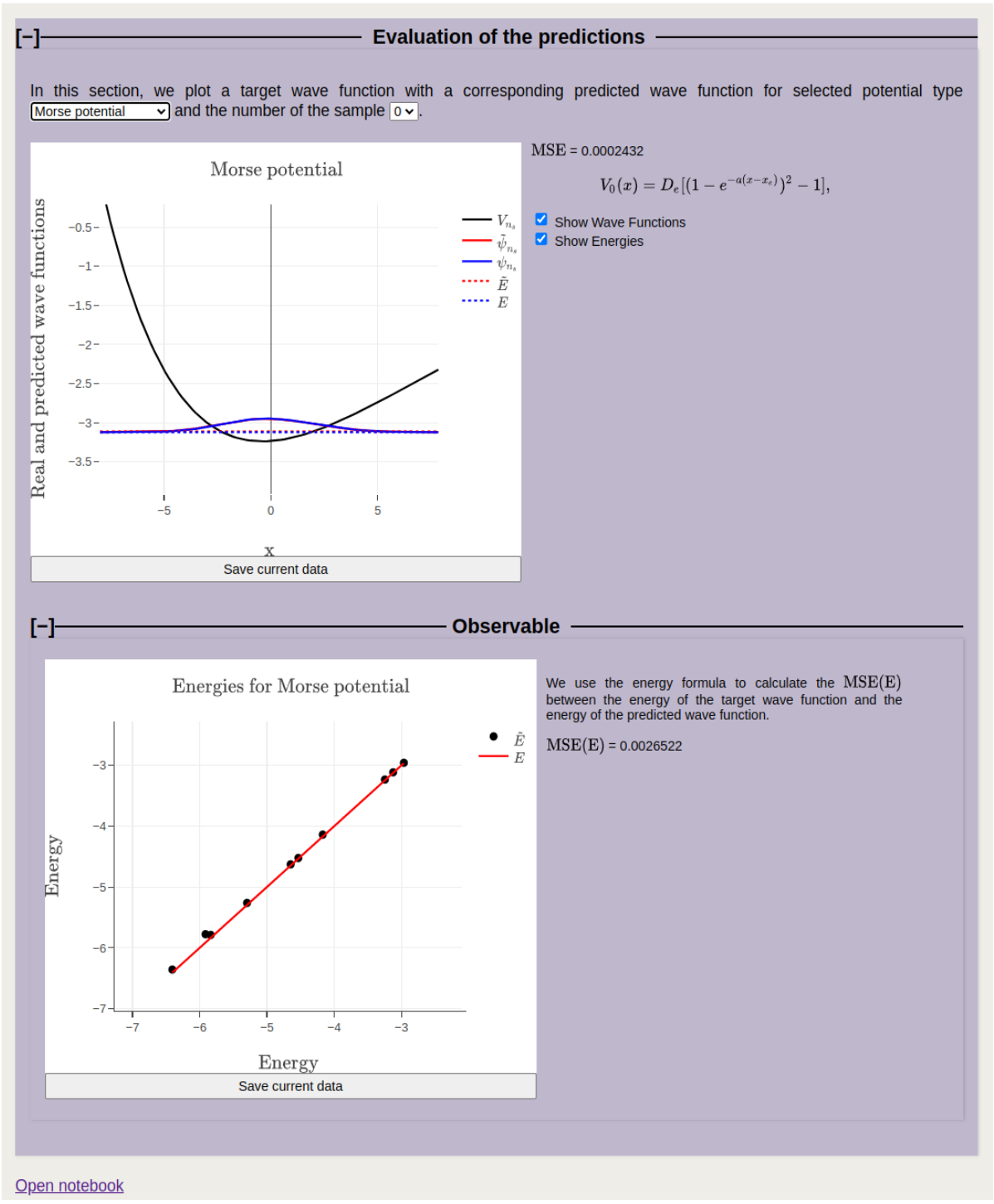
Give definition (as you understand them) for Loss Functions and analyze obtained Loss Functions as a function of the Epoch:	
Training loss	
Validation loss	
Analysis	
What does it mean, if the **validation loss** is much higher than the **training loss?**	

3. In the section **“Evaluation of the Predictions”**, you need to compare the results obtained through machine learning with those obtained via numerical calculations for different types of potentials. **Would the performance of the machine learning model be effective for a harmonic oscillator or Morse potential?**

To evaluate this, you are asked to evaluate the graphics in the section “Evaluation of the Predictions” (one example is shown in the figure below) and answer the questions at the end of the paragraph:

1. **The target wave function** and **the predicted wave function** are plotted in the top figure to visually assess how closely they match.
2. T**he predicted energies** are plotted against **the target energies** for all wave functions in the validation dataset in the bottom figure. Ideally, all points should lie on the diagonal, indicating accurate predictions.

You can select type of potential and the number of sample for which the figures is plotted. In the figure below you can see initial and predicted wave functions for Morse potential.



1. The parameter N determines the amount of points that are used to determine the quantum mechanical wave function. Inspect how (for a small number of samples) the wave functions change as a function of N. What is the effect on the accuracy of the numerical solution to the Schrödinger equation?
2. During the training, the class used to perform the training of the model uses the parameter *training_iters* (and *early_stopping* in the notebook) which determines the duration of the training. Inspect how the graph of training and validation losses changes with those parameters.

3. Compare the MSE for the polynomial potentials, harmonic oscillator and Morse potential. Explain the differences. Why the Morse potential loss is significantly larger, while the harmonic oscillator loss is much closer to the one for polynomial potentials?
4. What does representativeness mean in machine learning, and why is it important when creating a training dataset?

---

* Atomic units https://en.wikipedia.org/wiki/Hartree_atomic_units are used:

the unit of charge is e = 1 (1.602176634×10^{-19}C);
the unit of mass m_e = 1 is defined as the mass of the electron (9.1093837139(28)×10^{-31} kg);
the unit of action $\hbar$ = 1 - reduced Planck constant (1.054571817...×10^{-34} J·s);
the unit of length a_0 = 1 - Bohr radius ($a_0 = 4\pi\varepsilon\hbar^2/m_e e^2$ = 0.529×10^{-10}m);
the unit of energy E_H = 1 - Hartree energy ($E_H = \hbar^2/m_e a_0^2$ = 4.359×10^{-18} J = 27.211 eV).