Artificial Intelligence: Machine Learning Applications in Physics

Hiqmet Kamberaj, PhD

Professor at Faculty of Engineering, International Balkan University, Skopje, North Macedonia

Lecture notes in

Data Science Applications in Physics, Balkan School in Tirana 2024

January 22-26, 2024 1.1

Outline

Contents

 1.2

Objectives

At the end of this lesson, you should be able to:

- know what is data science from a physics perspective;
- know what is artificial intelligence from a computer science perspective;
- learn how to apply artificial intelligence technologies in data science from a physics and chemistry perspective.

1.3

1 Data Science: A Physicist Viewpoint

What is Data Science (DS)?

Definition 1. DS is a cross-disciplinary field, which uses some principles and techniques aiming at the automation of extracting potentially useful information (labelled data), knowledge (information that is understood), getting insight (pathways insight information (source, target)), wisdom (decisions based on the knowledge of pathways), new theory (*alert: conspiracy theory may result by false explanation ignoring the facts*).

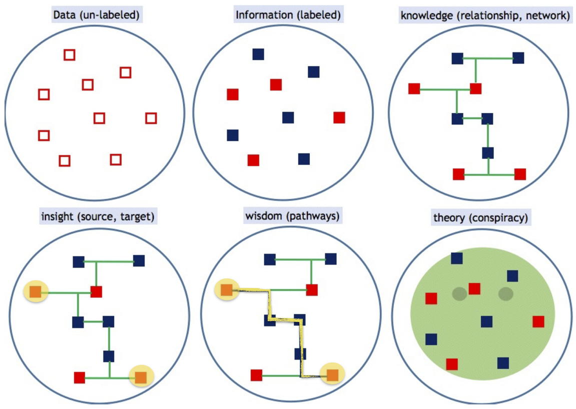

Figure 1: Data, information, knowledge, insight, wisdom, new theory

1.4

Data Science in Physics

Data science technologies are successfully applied in many different fields:

- predictive maintenance;
- healthcare;
- supply chain management;
- economics and finance;
- *physics*;
- and many others.

1.5

DS as a Cross-Disciplinary Field 1.6

2 Artificial Intelligence: A Computer Science Viewpoint

What is the Artificial Intelligence (AI)?

Definition 2. AI is a study field of computer science for building computer systems using science and engineering that exhibit intelligence that mimics human intelligence (natural intelligence).

AI research has helped in the development of many core research topics in computer science, such as:

- artificial neural networks (ANN);

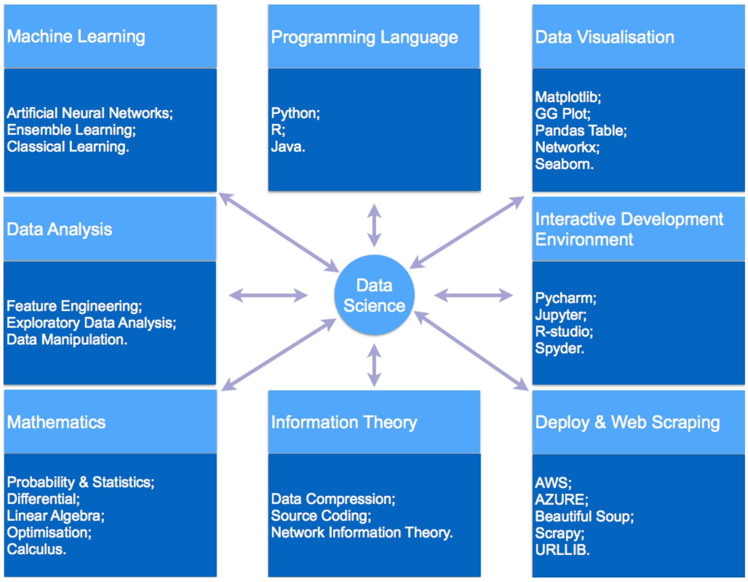

Figure 2: Data science technologies

- evolutionary computing;
- machine learning (ML);
- deep learning (DL);
- natural language processing;
- object-oriented programming. 1.7

Big Data

- AI systems have varying levels of autonomy that can make predictions, recommendations or decisions for a given set of human-defined objectives.
- AI techniques need massive amounts of alternative data sources and data analytics, known as big data.
- Importantly, in many applications, the primary focus for the research mentioned above is no longer the development of AI.
- Instead, they have become a discipline in themselves and, in some applications, are no longer thought of as being related to AI. 1.8

Artificial Neural Networks 1.9

Classical Learning 1.10

3 Artificial Neural Network

Artificial neuron

- The perceptron, or artificial neuron, takes several binary inputs (such as $x_1, x_2, x_3, \cdots$) and produces a single binary output:

$$y = \begin{cases} 0, & \sum_i w_i x_i \leq \text{threshold} \\ 1, & \sum_i w_i x_i > \text{threshold} \end{cases}$$

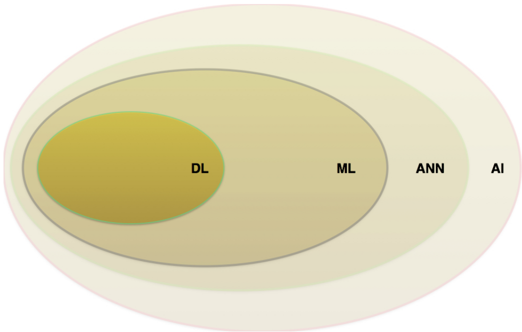

Figure 3: Relationship between AI (main field), ANN (sub-field of AI), ML (sub-field of AI and ANN), and DL (sub-field of AI, ANN, and ML)

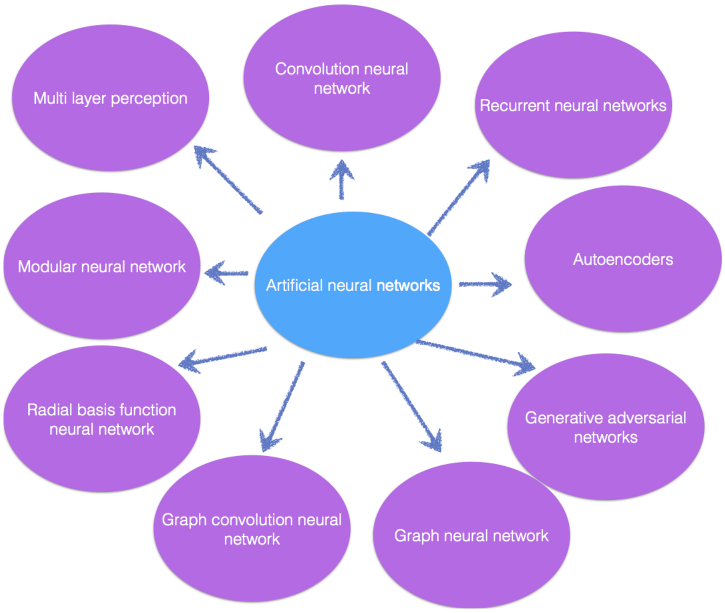

Figure 4: Artificial Neural Networks Algorithms

- The weights $(w_1, w_2, w_3, \cdots)$ and the threshold are real numbers, which are parameters of a neuron.
- A change is to shift the threshold to the other side of the inequality and to replace it with the so-known perceptron's bias, b = -threshold:

$$y = \begin{cases} 0, & \sum_i w_i x_i + b \leq 0 \\ 1, & \sum_i w_i x_i + b > 0 \end{cases}$$

1.11

To note

- The bias measures how easy it is to get the perceptron to output a value of one. In more biological terms, the bias measures how easy it is to get the perceptron to fire.
- For a perceptron with a considerable bias, it is straightforward for the perceptron to output a value of one.
- However, if the bias is very negative, it is difficult for the perceptron to output a value of one. Introducing the bias is only a tiny change in how we describe perceptrons.

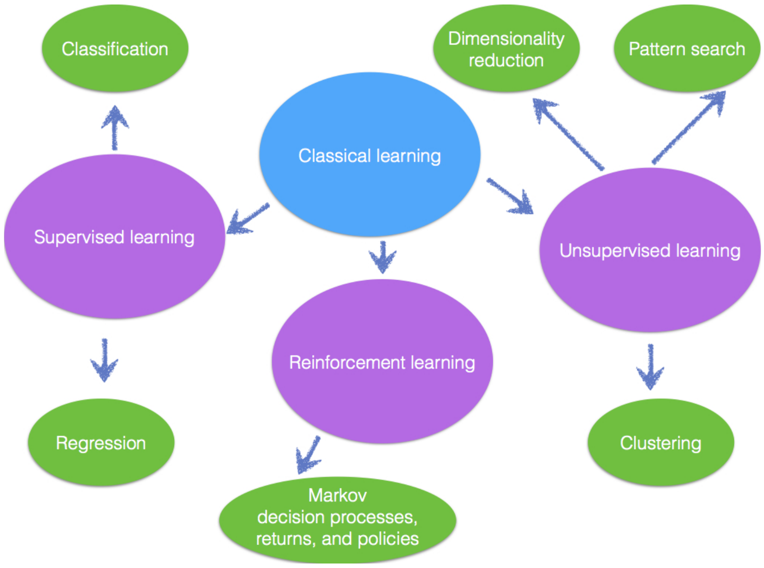

Figure 5: Classical Learning Algorithms

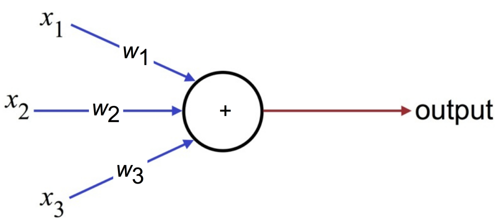

Figure 6: Artificial Neuron

- We can devise learning algorithms that automatically tune the weights and biases of a network of artificial neurons.
- That tuning happens in response to external stimuli without the direct intervention of a computer programmer.
- However, the perceptron is only a partial model of human decision-making.
- A perceptron can weigh up different kinds of evidence to make decisions; hence, it should seem plausible that a complex network of perceptrons could make quite subtle decisions. 1.12

Artificial neural network

- In this network, the first column of perceptrons is called the first layer, making simple decisions by weighing the input evidence.
- Each perceptron produces an output in the second layer by weighing up the results from the first layer of predictions.
- In this way, a perceptron in the second layer can produce a more complex and abstract output than perceptrons in the first layer.
- The perceptron can make even more complex decisions in the third layer, and so on.
- Thus, a many-layer network of perceptrons can engage in a sophisticated prediction process. 1.13

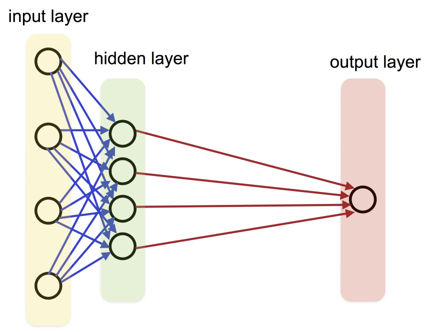

Figure 7: Artificial neural network

Remarks

- If we make a small change in some weight (or bias) in the network, we would practically expect only a tiny corresponding change in the output from the network.
- This property will make learning possible; we could modify the weights and biases to get our network to behave more like we want.
- Then, we would repeat this, changing the weights and biases many times to produce better and better output; that is, the network would be learning.
- That is not to happen if the network contains perceptrons.
- A small change in the weights or bias of any single perceptron in the network can sometimes cause the output of that perceptron to transform from 0 to 1.
- That transformation may cause the behaviour of the rest of the network to change in some highly complex way completely.
- Another type of artificial neuron (a sigmoid neuron) can be used to overcome this problem.

1.14

Sigmoid neuron

- For a sigmoid neuron, taking binary inputs (such as $x_1, x_2, x_3, \cdots$), the output is:

$$y = f\left(\sum_i w_i x_i + b\right) = f\left(\mathbf{wx} + b\right)$$

- Here, $\mathbf{w} = (w_1, w_2, w_3, \cdots)$, $\mathbf{x} = (x_1, x_2, x_3, \cdots)^T$, and $f(\cdots)$ is a sigmoid function.
- Sigmoid neurons are similar to perceptrons but modified so that small changes in their weights and biases cause only a tiny change in their output.
- That is essential, allowing a network of sigmoid neurons to learn.

1.15

Sigmoid functions

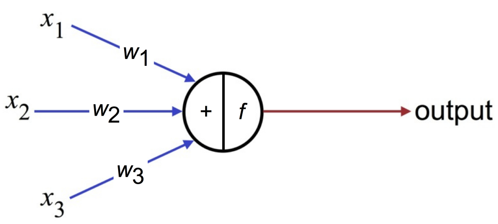

Figure 8: Sigmoid Neuron

- The main difference between perceptrons and sigmoid neurons is that they do not just output 0 or 1 but can have any real number between 0 and 1 as output.
- Sometimes, neurons with different activation functions $f(\cdots)$ for the output can be considered as long as the shape of the function f remains roughly the same.
- $f(z)$ must enable analytical expression for the derivative of $f(z)$ for z to reduce the computational costs.

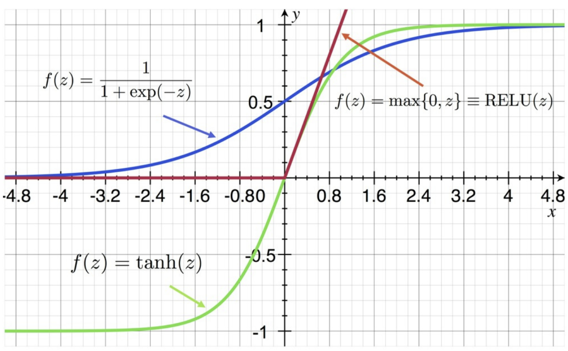

Figure 9: Some sigmoid functions

1.16

4 Machine learning

Machine learning: with $\mathcal{K}$ hidden layers and output Y_α

$$Y_\alpha = f^{\text{linear}}_{\mathbf{P}} \underbrace{\left(\sum_{\beta=1}^{L_o} f^{\text{ann}}_{\mathbf{P}} \underbrace{\left(\sum_{l_\mathcal{K}=1}^{L_\mathcal{K}} f^{\text{ann}}_{\mathbf{P}} \underbrace{\left(\cdots f^{\text{ann}}_{\mathbf{P}} \underbrace{\left(\sum_{l_2=1}^{L_2} f^{\text{ann}}_{\mathbf{P}} \underbrace{\left(\sum_{l_1=1}^{L_1} \underbrace{f^{\text{ann}}_{\mathbf{P}} \left(\sum_{j=1}^{L_i} X_j W_{jl_1} + b_{l_1} \right)}_{\text{input layer}} W_{l_1 l_2} + b_{l_2} \right)}_{\text{1st hidden layer}} \cdots \right)}_{\text{2nd hidden layer}} \cdots \right)}_{\cdots} W_{l_\mathcal{K}\beta} + b_\beta \right)}_{(\mathcal{K})\text{th hidden layer}} W_{\beta\alpha} + b_\alpha \right)}_{\text{output layer}}$$

1.17

Machine learning

- $f^{\text{linear}}_{\mathbf{P}}$ is a linear function and $f^{\text{ann}}_{\mathbf{P}}$ are non-linear function with existing analytical first derivative.
- It determines a machine learning transformation of the input feature descriptor vector according to:

$$M_{\mathbf{P}} : \ \mathbf{X} \to \mathbf{Y}$$

- In general, the ML is characterised by an input vector $\mathbf{X}$ of dimension L_i, $\mathcal{K}$ hidden layers of $l^{(1)}_{L_1}, l^{(2)}_{L_2}, \cdots, l^{(\mathcal{K})}_{L_\mathcal{K}}$ neurons each, and an output vector $\mathbf{Y}$ of dimension m.
- $\mathbf{W}$ and $\mathbf{b}$ are considered as free parameters, which need to be optimised for a given training data used as inputs and given outputs, which are known.
- The so-called loss function is minimised to optimise these parameters:

$$L(\mathbf{W}, \mathbf{b}) = \mathcal{F}\left(\mathbf{Y}^0, \mathbf{Y}, \lambda_1, \lambda_2, \cdots\right)$$

- $\mathbf{Y}^0$ represents the true output vector, and $\mathcal{F}$ is a convex function (such as mean square error or mean absolute error) and λ_i regularisations (learning rate, regulation strength, etc.).

1.18

5 Deep Learning

What is a DL?

- DL is a general-purpose machine learning technology, which typically refers to a set of machine learning algorithms based on learning data representations.
- Each hidden layer neuron represents a task, and each layer can be broken down through a multilayer to sub-networks.
- DL network is a many-layer structure: two or more hidden layers.

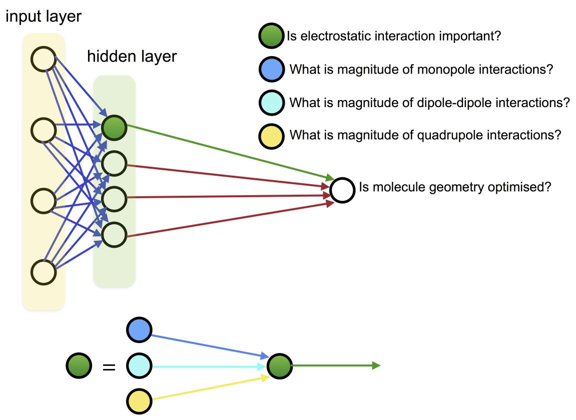

Figure 10: Deep learning

1.19

What is Learning?

- Setting the weights manually is difficult, even for minimal networks.
- In biological organisms, synapses have a plasticity that enables learning.
- The power of artificial neural networks comes from their ability to learn, unlike ordinary algorithms that have to be specified to the last detail.
- There are three main categories of learning algorithms:
  1. Supervised learning;
  2. Reinforcement learning;
  3. Unsupervised learning.

1.20

What is Supervised Learning?

1) A supervised Neural Network represents a transformation from an input vector onto a (known) output vector. The error between the desired and the actual outputs is used to correct the weights to reduce the error. It is necessary to train a network if we already know how it should behave because the network is required to provide outputs in situations for which it was not trained. If the weights are adjusted so the network behaves correctly on known inputs, it is reasonable to

assume that its behaviour will be more or less correct on other inputs. Furthermore, training a network simplifies the learning process: rather than directly relating the outputs to specific input values, placing the network in several different situations and telling it which outputs are expected in each situation is more effortless. 1.21

What is Reinforcement Learning?

2) In reinforcement learning, the exact output value is not given in each situation; we specify whether the computed output is good for the network. Reinforcement is appropriate when it is easy to distinguish correct behaviour from incorrect behaviour, but we need to know the exact output for each situation. Reinforcement learning is a sequential decision-making framework in which agents learn to perform actions in an environment to maximise rewards. 1.22

What is Unsupervised Learning?

3) Unsupervised learning is learning without external feedback, where the network adapts to many inputs. Unsupervised learning is not appropriate for achieving specified goals; instead, it is used in classification problems where the network is presented with raw data and attempts to find trends within the data. 1.23

Remarks

- The ANN architectures, such as the multilayer perceptron and the radial basis function, are often used.
- A sequence of ANN layers in cascade is considered to build a deep learning framework.
- The current success of deep learning methods is related to the advances in algorithms and high-performance computing technology, which allow analysis of the large data sets that have now become available.
- For example, the robot-advisor tools use deep learning technologies to improve accuracy.
- Another approach includes recurrent connections connecting the neural networks' hidden units back to themselves with a time delay, which is the principle of Recurrent Neural Networks and Long Short-Term Memory Networks (LSTMs).
- These ANNs are designed to work with sequential data that arise in applications such as time series, natural language processing, and speech recognition. 1.24

Back-propagation

- The back-propagation is an essential and powerful technique used in neural network training approaches.
- For that, the gradients of $L(\mathbf{W}, \mathbf{b})$ for $\mathbf{W}$ and $\mathbf{b}$ are calculated:

$$\Delta\mathbf{W} = -\left(\frac{\partial L(\mathbf{W}, \mathbf{b})}{\partial \mathbf{W}}\right)_{\mathbf{b}}$$

$$\Delta\mathbf{b} = -\left(\frac{\partial L(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}}\right)_{\mathbf{W}}$$

- A common problem of the machine learning approaches is over-fitting, which is avoided by the following regularisation terms:

$$\mathbf{W}^{(\text{new})} = \mathbf{W}^{(\text{old})} + \gamma_w (\Delta\mathbf{W} + \gamma_1 \mathbf{W})$$

$$\mathbf{b}^{(\text{new})} = \mathbf{b}^{(\text{old})} + \gamma_w (\Delta\mathbf{b} + \gamma_1 \mathbf{b})$$

- γ_w is called learning rate for the gradient and γ_1 is called the regulation strength.

1.25

L_2 norm regularisation

- L_2 norm or regulation strength (λ_1) penalises the sum of squares of the parameter values set.
- The net effect of L_2 regularisation is to produce smaller parameter values (and so it is often used for weights). Thus, the loss function is smoother (less local minima and faster convergence to the desired global minimum).
- The minima in the $L'(\mathbf{W}, \mathbf{b})$ and $L(\mathbf{W}, \mathbf{b})$ differ, and thus the procedure converges to different parameter values set.
- If λ_1 is small, it has little effect.
- If λ_1 is large, the function becomes smooth and fit to the data becomes less accurate.

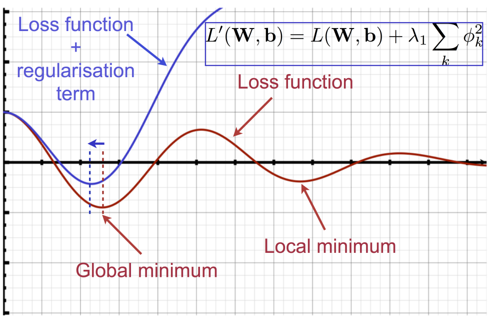

Figure 11: L_2 norm regularisation

1.26

Test performance improvement due to L_2 norm regularisation

However, λ_1 being large may improve the test performance because:

1. If the network is overfitting, the regularisation term implies a trade-off between the given data (+ noise) and the desire to be smooth (− noise). Since the model does not need to go through every data point, this model applies to fitting only smooth functions.
2. If the network has too many parameters to optimise, then the model can describe regions of the space where the training data are missing. Therefore, the regularisation term produces functions that smoothly interpolate between the neighbouring points, which is useful in cases when the knowledge about the true functions is missing.

1.27

Implicit regularisation

- In the case of gradient descent method (GDM) ($\phi \equiv (\mathbf{W}, \mathbf{b})$):

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{rate change}} = - \underbrace{\frac{\partial L}{\partial \phi}}_{\text{gradient of } L}$$

- Along the discrete path: $\phi_{t+1} = \phi_t + \gamma_w \cdot \frac{\partial \phi}{\partial t} = \phi_t - \gamma_w \cdot \frac{\partial L(\phi_t)}{\partial \phi}$.
- γ_w is a step size.

1.28

Graphical illustration of implicit regularisation

- A discrete map produces a deviation from the continuous path.
- In GDM, a modified loss function L' for continuous the path is constructed that arrives at the exact location as a discrete path along L:

$$L'(\phi) = L(\phi) + \frac{\gamma_w}{4} \mid\mid \frac{\partial L}{\partial \phi} \mid\mid^2$$

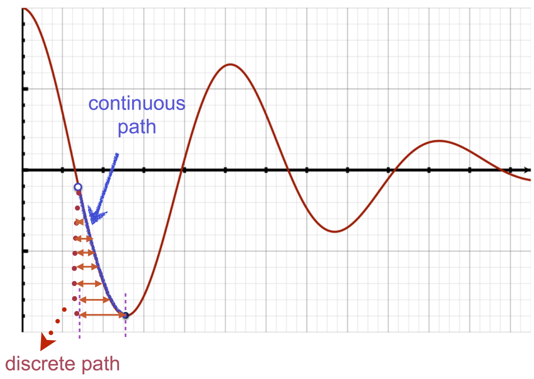

Figure 12: Implicit regularisation

1.29

Explanation of implicit regularisation

- The discrete path is *forced* away from the continuous path at the positions where the gradient becomes large (where the line/surface is steep).
- The additional term does not change the location of (local) minima because $\frac{\partial L}{\partial \phi} = 0$.
- Nevertheless, it changes the effective loss function $L'(\phi)$ elsewhere ($\frac{\partial L}{\partial \phi} \neq 0$), which most likely converges to a different minimum.

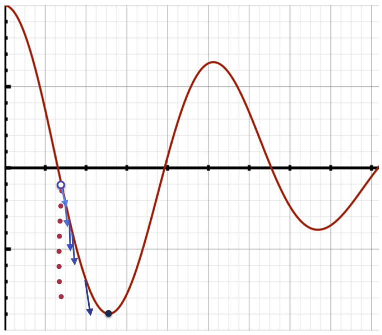

Figure 13: Explanation of implicit regularisation

1.30

A note on implicit regularisation

- Full batch gradient descent converges better for large step sizes due to gradient descent implicit regularisation.
- Implicit regularisation is also used in stochastic gradient descent, which generalises better than gradient descent.
- In stochastic gradient descent, small batch sizes, in general, perform better than large ones (perhaps) due to randomness allowing reaching different parts of a loss function.

1.31

Heuristic methods

- Adding explicit regularisation terms enables the training algorithm to improve in finding an optimal solution.
- In particular, this is done by modifying the loss function by additional penalty terms.
- In addition, heuristic methods can be used to improve the generalisation.
- For example, early stopping, which indicates stopping training the data before the convergence is reached, can reduce overfitting if the model has already captured the coarse-grained shape of the loss function.
- Ensembling, which will be discussed later.
- Dropout can randomly choose a subset of hidden neurons to zero at each iteration.

- Applying noise to the network activation (such as Bernoulli noise).
- Bayesian inference, transfer learning, multi-task learning, self-supervised learning, and augmentation.

1.32

Ensembling

- An ensemble is obtained by training using different random seeds, hyper-parameters, or entirely different model families.
- The models can be combined by averaging their predictions, weighting the predictions, or stacking (in which the results are combined using another ML model).
- The average of the outputs generated independently (or combined) trained neural network models improves accuracy, calibration, and robustness.
- An efficient way is to swap the hyper-parameters or training data between neural network models of the ensemble at regular time intervals.
- In contrast, the averaging together the hyper-parameters (weights and biases) to make one ML model out of the ensemble that network fails!
- However, choosing the hyper-parameters of the ML model of the ensemble that performs the best (global best output prediction) can result in an improvement.

1.33

Bootstrapping Swarm Artificial Neural Network Method

- An efficient approach is to generate several different datasets by resampling the training data with replacement and training a different model from each.
- It is called bootstrap aggregating or bagging.
- The net effect is smoothing out the data; furthermore, if a data point is not included in one training dataset, the model will either interpolate from neighbouring data points or perform regular exchanges of training datasets between models.

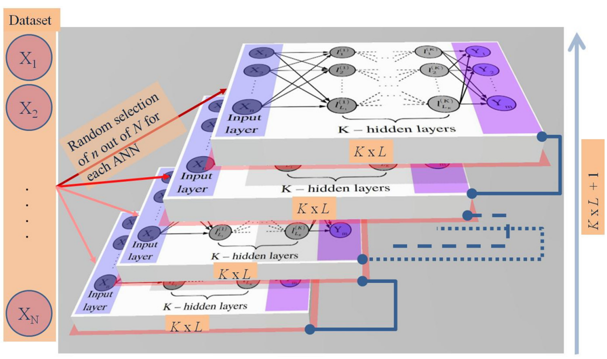

Figure 14: Ensemble ML model

1.34

Swam-particle intelligence

Assuming that the search space is d-dimensional, and the position of the i particle is denoted by $X_i = (x_{i1}, x_{i2}, \cdots, x_{1d})$

$$V_{ij}(t+1) = w\left(c_1 V_{ij}(t) + c_2 u_1 (X_{ij}^{LBest}(t) - X_{ij}(t)) + c_3 u_2 (X^{GBest}(t) - X_j(t))\right)$$
$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t), \quad i = 1, \cdots N_p,\ j = 1, \cdots d$$

- The first term is inertia;
- The second term is cognitive term;
- The third term is the social knowledge term.

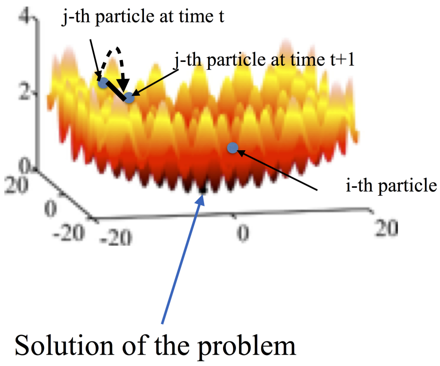

Figure 15: Swarm-particle intelligence

1.35

Swam-particle intelligence regularisation terms

The following new regularisation terms are introduced:

$$\begin{aligned}
\mathbf{W}^{(\text{new})} &= \mathbf{W}^{(\text{old})} + \gamma_w \left[\Delta \mathbf{W} + \gamma_1 \mathbf{W}\right. \\
&\quad \left. - \gamma_2 U(0,1)\left(\mathbf{W} - \mathbf{W}^{\text{Lbest}}\right) - \gamma_3 U(0,1)\left(\mathbf{W} - \mathbf{W}^{\text{Gbest}}\right)\right] \\
\mathbf{b}^{(\text{new})} &= \mathbf{b}^{(\text{old})} + \gamma_w \left[\Delta \mathbf{b} + \gamma_1 \mathbf{b}\right. \\
&\quad \left. - \gamma_2 U(0,1)\left(\mathbf{b} - \mathbf{b}^{\text{Lbest}}\right) - \gamma_3 U(0,1)\left(\mathbf{b} - \mathbf{b}^{\text{Gbest}}\right)\right]
\end{aligned}$$

γ_w is again learning rate for the gradient and γ_1 is the regulation strength. This results from modifying the loss function:

$$\begin{aligned}
L'(\phi) = L(\phi) &+ \frac{\gamma_2}{2} U(0,1)\left(\mathbf{W} - \mathbf{W}^{\text{Lbest}}\right)^2 + \frac{\gamma_3}{2} U(0,1)\left(\mathbf{W} - \mathbf{W}^{\text{Gbest}}\right)^2 \\
&+ \frac{\gamma_2}{2} U(0,1)\left(\mathbf{b} - \mathbf{b}^{\text{Lbest}}\right)^2 + \frac{\gamma_3}{2} U(0,1)\left(\mathbf{b} - \mathbf{b}^{\text{Gbest}}\right)^2
\end{aligned}$$

$U(0,1)$ is a random number between zero and one, and γ_2 and γ_3 represent the strength of biases toward the local and global best optimal parameters, respectively. 1.36

6 Applications

Databases

1. The database of 642 small organic molecules, for which we know the experimental hydration-free energies.
2. A database contains 1475 experimental pKa values of ionisable groups in 192 wild-type and mutated proteins (153 proteins) and mutated (39 proteins).
3. Another database comprises 2693 experimental values of the Gibbs free energy changes in 14 mutant proteins.
4. The database contains 7101 quantum mechanics heat of formation calculations QM7, a subset of the GDB13 molecules optimised at the quantum mechanics level with the Perdew-Burke-Ernzerhof hybrid functional (PBE0).
5. A database of around 200 liquid crystal dopant molecules helical twisting power in different liquid crystal solvent matrices. The three-dimensional structures are optimised using molecular mechanics and quantum mechanics methods.

1.37

Microservice architecture

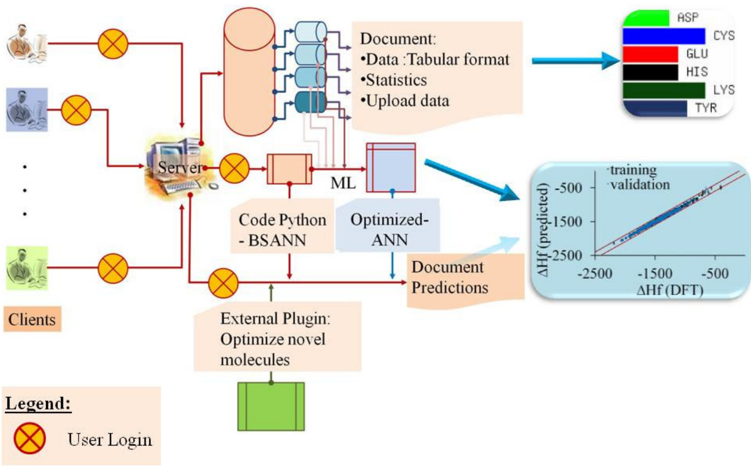

Figure 16: Microservice architecture

1.38

7 Feature Extraction from Input Data

Feature extraction

Definition 3 (feature extraction)**.** The process of selecting and transforming input data into relevant and informative features that can be used to train machine learning models is termed feature extraction.

- Feature extraction includes the identification of the critical variables and patterns from a vast input vector of data.
- By filtering this information into a maintainable set of features, AI can successfully analyse and interpret the data to uncover hidden information and make more effective and accurate predictions.

- One of the main advantages of using AI for feature extraction is its potential to process and analyse large volumes of data much faster than human analysts can.
- Furthermore, AI algorithms uncover delicate patterns and relationships between variables that may not be immediately apparent to data analysts, which can lead to more accurate and reliable predictions.
- Another benefit of the AI-powered feature extraction technique is its ability to adapt and evolve; AI algorithms can continuously remove impurities and improve their feature input datasets to maintain their predictive accuracy, enabling a dynamic feature extraction approach.

1.39

Some remarks

- Often, the complexity and diversity of input data make it challenging to identify the most relevant and informative features.
- Furthermore, data may be influenced by a wide range of factors, introducing a high degree of uncertainty and noise into the data, making it challenging for AI algorithms to distinguish between data conveying information and noises.
- Several promising approaches to AI-driven feature extraction exist without being affected by these difficulties.
- For instance, deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown great potential in processing and analysing data.
- These improved machine learning models can automatically learn and extract relevant features from input datasets.
- Other relevant approaches in AI-powered feature extraction include natural language processing (NLP) techniques, enabling AI algorithms to analyse and interpret unstructured text data.
- Integration of the NLP into the feature extraction process of AI-driven models can benefit by gaining a more comprehensive understanding of the factors driving data changes and making more accurate predictions.

1.40

Ideal versus real datasets

- The diversity of the feature descriptors of the compound database is essential to increase the range of the test data that can be predicted since the machine learning methodology works very well in interpolating the new data points but suffers on extrapolating new data outside the range covered by the training dataset.
- Unfortunately, ideal datasets do not exist!
- Build the dataset dynamically such that the topology of the dataset constructs the ideal dataset's coarse-grained topology.

1.41

Topological data analysis (TDA)

- Therefore, one of the critical future developments of the ML methodologies is the choice of the training dataset and the feature descriptors.

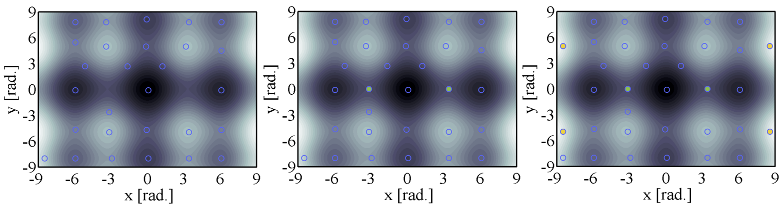

Figure 17: Dynamically building datasets

- A TDA tool helps to analyse the feature descriptors.
- The TDA is a field dealing with the topology of the data to understand and analyse large and complex datasets.
- Knowing about the topology of the data (e.g., the sparsity of the data points) is of great interest.

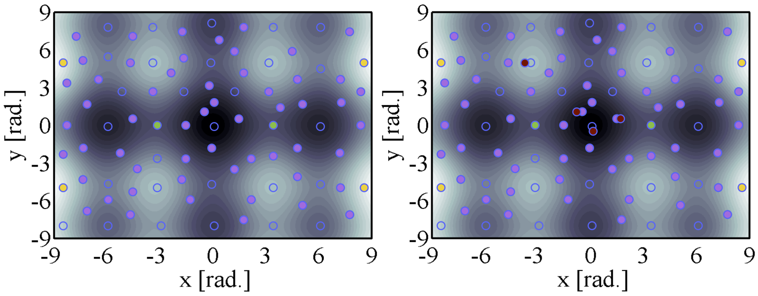

Figure 18: Topological data analysis

1.42

Training, validation, and test datasets

- The dataset is split into a training dataset $\mathcal{D}_{\text{train}}$ used for learning (or gaining experience), a validation dataset $\mathcal{D}_{\text{valid}}$ used for testing the knowledge, and a test dataset $\mathcal{D}_{\text{test}}$ used for learning:

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}} \cup \mathcal{D}_{\text{test}}; \mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{test}} = \emptyset$$

- Often, the size of the training dataset is about 80 % of the total dataset size.

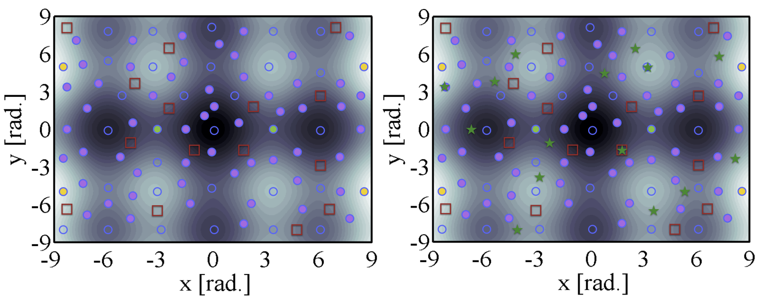

Figure 19: Training, validation, and test datasets

1.43

Technical aspects of AI for physics: **Symmetry**

- In physics problems, objects or systems of interest usually contain geometric structures, which may imply certain symmetries that the underlying physics obeys.
- For example, in molecular dynamics, molecules are represented as graphs in 3D space, and translating or rotating a molecule may not change its properties.
- Then, the symmetry here is named translational or rotational invariance.
- Here, the symmetry defines a transformation applied to a system that keeps certain properties of that system invariant or changed in a deterministic way.

1.44

Technical aspects of AI for physics: **Interpretability**

- The laws and principles of physics and science govern the physical world.
- Therefore, the AI in physics should be such that: (i) design models that can model the physics phenomena accurately; (ii) interpret models to verify and discover physics phenomena behind them.

1.45

Technical aspects of AI for physics: **Out-of-distribution and causality**

- Standard ML models assume that the training and test datasets follow the same distribution.
- However, in real situations, there may exist different distribution shifts between the two datasets; therefore, there is a need to determine the causal factors that cause the OOD.
- In particular, this is important in computer simulations to avoid the need to create training datasets for each different setting.

1.46

Technical aspects of AI for physics: **Large language models**

- When labelled training datasets do not exist, unsupervised ML models become interesting.
- Here, the natural language processing tasks have demonstrated a promising performance based on self-supervised models.
- In particular, developing large language models, such as *Generative Pre-trained Transformer (GPT)*, has provided a paradigm for discoveries in AI for physics.

1.47

Technical aspects of AI for physics: **Quantification of uncertainty**

- High prediction of decision-making under the employed data and model uncertainty is crucial in AI for physics.
- The uncertainty quantification in ML models is as important as in other fields (such as statistical modelling, applied mathematics, and information sciences) − *Bootstrapping Artificial Neural Network*.

1.48

Technical aspects of AI for physics: **Education**

- AI for physics is an emerging and fast-developing area in the era of (post)-modern physics.
- Therefore, developing valuable physical or online resources has been the main aim of many international research projects.
- A primary goal is facilitating learning and education to compile valuable resources.
- Furthermore, providing a perspective on how we, as a scientific community, can improve and facilitate the integration of AI technology with science (such as physics) and education could be of particular interest.

1.49

European Project

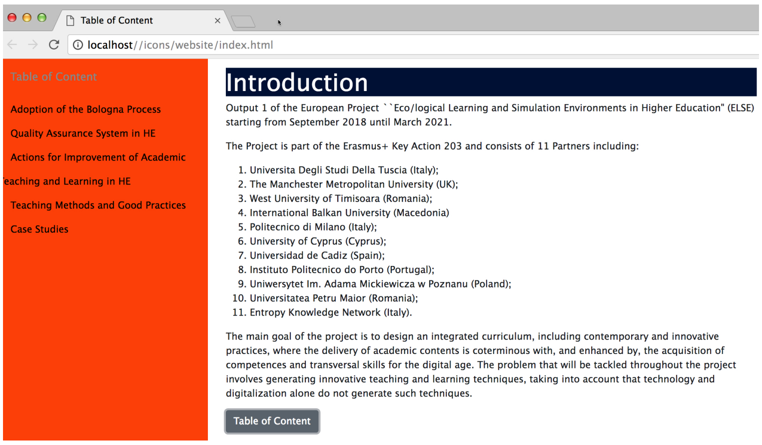

Figure 20: Eco/Logical Learning and Simulation Environments in Higher Education

1.50

In-class demonstration

Figure 21: Quantum machine learning

1.51

Future Directions in Computer Software

1. Heterogeneous Architectures - competing architectures can be coupled to handle a diverse range of tasks for complex workflows efficiently.
2. Synthesis of Machine Learning and Hardware Acceleration for Molecular Dynamics - Integration of Machine Learning and Molecular Dynamics (MD).
   - JAX (a Python library built upon GPU and Tensor Processing Units (TPUs) - AI ASIC designed by Google), a framework for allowing integration of ML and MD codes;
   - TorchMD - Python Library - an interface built upon PyTorch - using GPUs and TPUs;
   - The Graph-core Intelligence Processing Unit (IPU) is yet another AI accelerator.
3. Improvement of Accuracy and Algorithms: Application of a heterogeneous computation system to scale deep-learning-based Potential Energy Surface models.

1.52

8 Summary

Summary

1. AI and its sub-fields (ANN, ML, and DL) can be used for various physics and chemistry problems in industry and academic research.
2. The focus should be on developing education curricula that prepare students for cutting-edge science and technology careers.
3. These curricula should address the strategic goals of education, research, and technology transfer.
4. The primary aim of this lecture is to encourage more widespread use of artificial intelligence technologies in physics.

1.53

9 Acknowledgements

Acknowledgements

1. International Balkan University, Skopje, North Macedonia;
2. National Institute of Physics, Tirana, Albania;
3. PhD and Master's students;
4. Artificial Intelligence, Machine Learning, Deep Learning, and Data Science Social Media Discussion Forum;
5. Eco/Logical Learning and Simulation Environments in Higher Education (ELSE), Erasmus+, under the grant IT02-KA203-048006;
6. Transversal Skills in Applied Artificial Intelligence (TSAAI) - Curriculum Development, Erasmus+, under the grant ES01-KA220-HED-000030125.

1.54

10 References

1. H. Kamberaj, Springer Nature, Switzerland, Scientific Computation Series, 2020.
2. Rabie A. Ramadan, Ahmed Y. Khedr, Kusum Yadav, Eissa Jaber Alreshidi, Md Haidar Sharif, Ahmad Taher Azar & Hiqmet Kamberaj, Convolution neural network based automatic localisation of landmarks on lateral x-ray images. Multimedia Tools Applications, 81, 37403, 2022.
3. Ramadan, Rabie A. and Sharif, Md. Haidar and Alreshidi, Eissa Jaber and Sharif, Md. Haris Uddin and Ince, Ibrahim Furkan and Kamberaj, Hiqmet and Khedr, Ahmed Y., Energy Coherent Fog Networks using Multi-Sink Wireless Sensor Networks, IEEE Access, 9, 167715, 2021.
4. M. Hoxha and H. Kamberaj, Automation of Some Macromolecular Properties Using a Machine Learning Approach, Machine Learning: Science and Technology, 2, 035016, 2021.
5. M. Hoxha, B. Rahmani, H. Kamberaj, A database used for automation of molecular properties using machine deep-learning approach, Bulletin of Natural Sciences, 28, 62, 2019.
6. H. Kamberaj, Heat Flow Random Walks in Biomolecular Systems Using Symbolic Transfer Entropy and Graph Theory, Journal of Molecular Graphics and Modelling, 104, 107838, 2021.
7. D. Rahmani and H. Kamberaj, The Capsule − An Artificial Intelligence Powered Chatbot in Human Resources Management Systems, IBU International Journal of Technical and Natural Sciences, 2, 53-69, 2021.
8. R. Izairi and H. Kamberaj, Comparison Study of Polar and Nonpolar Contributions to Solvation Free Energy, Journal of Chemical Information and Modeling, 57 (10), 2539-2553, 2017.
9. Simon J.D. Prince, Understanding Deep Learning, MIT Press, 2023.
10. Somdatta Goswami et. al, Physics-Informed Deep Neural Operator Networks, 2023.
11. Shiqiang Zhu et. al, Intelligent Computing: The Latest Advances, Challenges, and Future, Intelligent Computing, 2023.
12. Frank Noé et. al, Machine Learning for Molecular Simulation, Annual Review of Physical Chemistry, 2020.
13. Gabrielle Black et. al, Exploring chemical space in non-targeted analysis: a proposed ChemSpace tool, Anal Bioanal Chem., 415, 35, 2023.
14. Puck van Gerwen et. al, Physics-based representations for machine learning properties of chemical reactions, Mach. Learn.: Sci. Technol. 3, 045005, 2022.
15. William J. Teahan, Artificial Intelligence - Agent Behaviour I, ISBN 978-87-7681-559-2, 2010.
16. William J. Teahan, Artificial Intelligence - Agent and Environments, ISBN 978-87-7681-528-8, 2010.
17. Micheal Nielsen, Neural networks and deep learning, 2017. (Accessed 2017, http://neuralnetworksanddeeplearning.com)