PHYSICS AND MACHINE LEARNING: AN OVERVIEW

Saúl Alonso-Monsalve

ETH Zurich

IPA-ML

20 March 2023

Overview

1. Introduction.
2. Foundations.
3. Applications.
4. Challenges and future directions.
5. Conclusion.

Overview

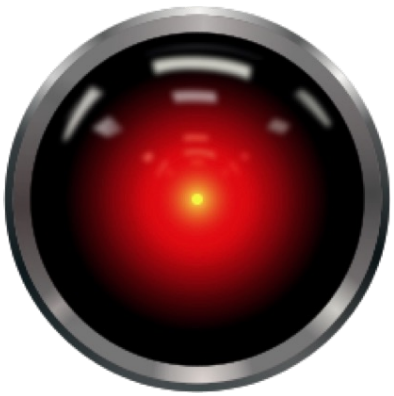
1. **Introduction.**
2. Foundations.
3. Applications.
4. Challenges and future directions.
5. Conclusion.

Credit: https://fakeyou.com


MACHINE LEARNING
MACHINE LEARNING EVERYWHERE!
memegenerator.net

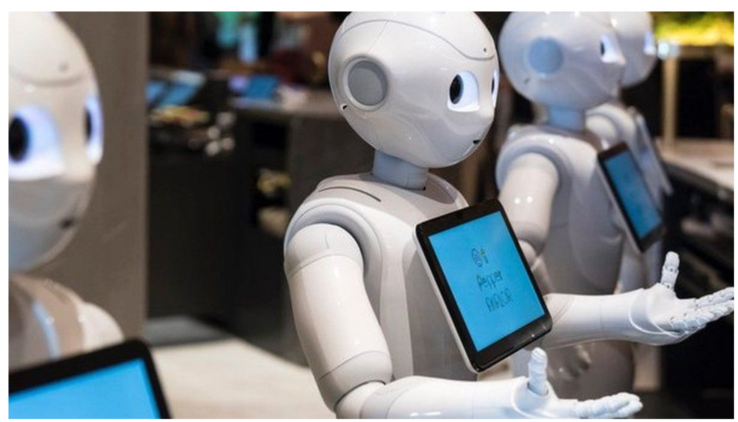
What is NOT machine learning (ML)?

- ML is not Hal-9000.



- ML is not Terminator.

- ML is not an alternative to human beings.

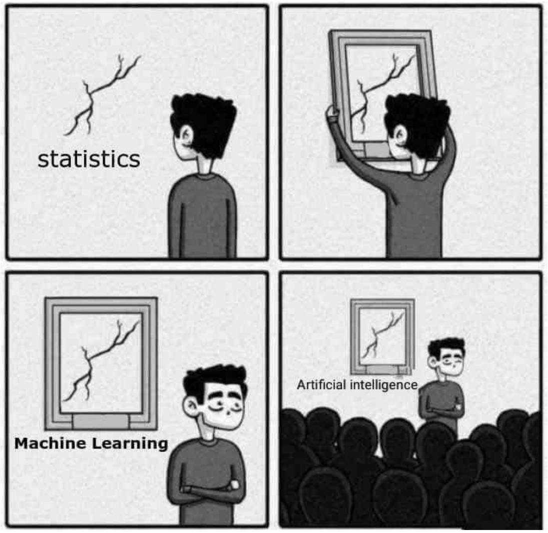


(sorry, chatGPT).

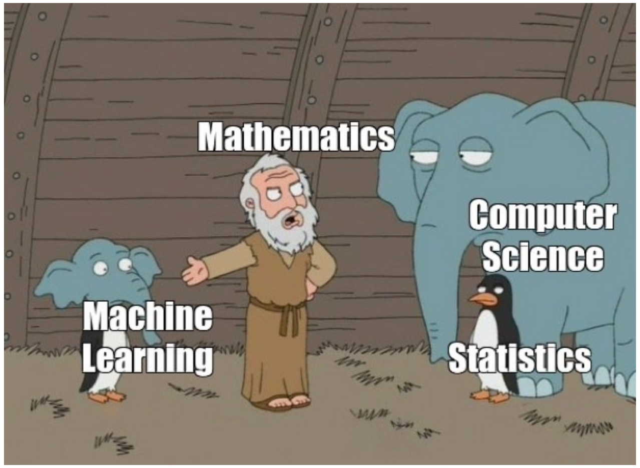
What is machine learning then?

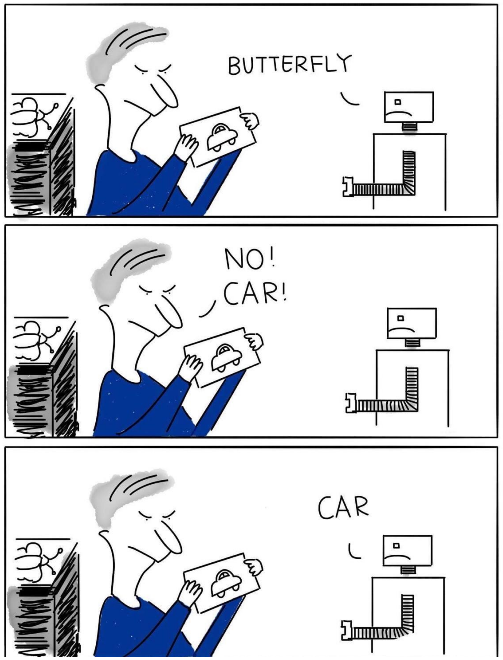
- ML is a subfield of artificial intelligence (AI).
  - AI: branch of computer science that aims to build algorithms capable of performing tasks typically (traditionally) accomplished using human intelligence.

- ML is statistics in disguise.



- ML is learning from data.
  - There is no learning without data.
  - ML algorithms only learn from the data.





What is machine learning then?

- **Arthur Samuel** defined the term machine learning in 1959 as "*the field of study that gives computers the ability to learn without being explicitly programmed*".

- **Tom Mitchell** updated Arthur's definition in 1998: "A computer program is said to learn from *experience E* with respect to some *task T* and some performance *measure P*, if its performance on *T*, as measured by *P*, improves with *experience E*".
    - Example:
        - Classifying emails as spam or not spam (T).
        - Watching a person labelling emails as spam or not spam (E).
        - The fraction of emails correctly classified as spam or not spam (P).

Machine learning applications

- More than 99% of the current machine learning applications have the form A → B (supervised learning).

→ ML algorithm that classifies images as cats or dogs → DOG!

- They learn the task that has been entrusted to them; they are not able to think for themselves.

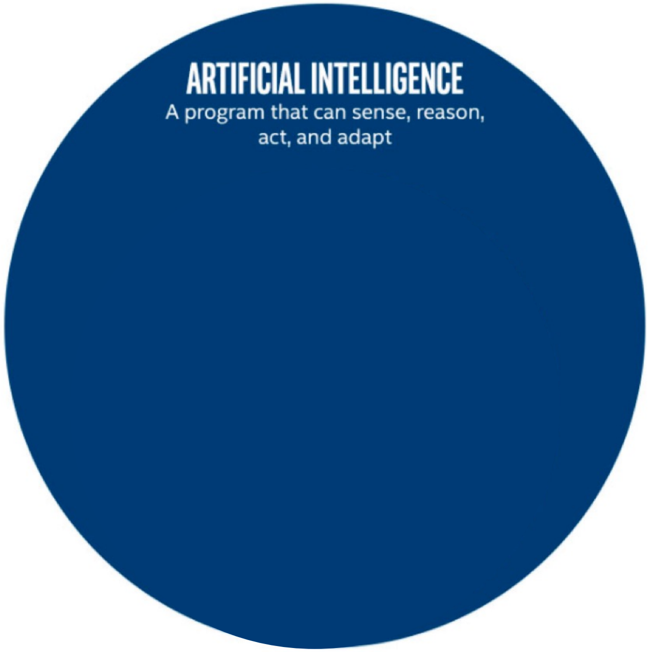
→ ML algorithm that classifies images as cats or dogs → DOG!

Overview

1. Introduction.
2. **Foundations.**
3. Applications.
4. Challenges and future directions.
5. Conclusion.

AI, machine learning, and deep learning

- **Artificial intelligence (AI):** branch of computer science that aims to build algorithms capable of performing tasks traditionally accomplished using human intelligence.
- **Machine learning (ML):** AI algorithms that learn from input data to perform "intelligent" tasks.
- **Deep learning (DL)**: subset of ML; consists of deep neural networks trained on large datasets.
- **Physics-based deep learning (PBDL)**: combinations of physical modelling and numerical simulations with methods based on artificial neural networks.



*Source: Stack Exchange

Types of machine learning

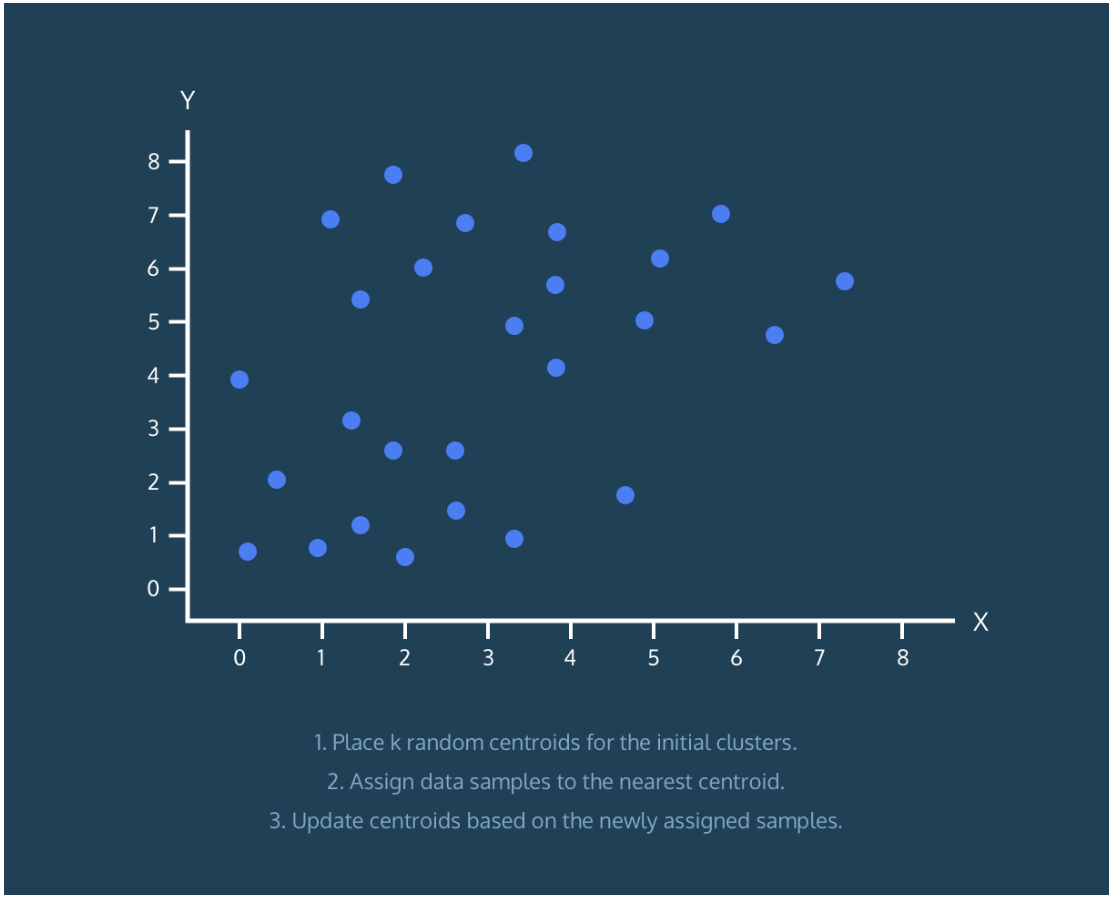
1. **Supervised learning:** we are given a dataset and already know what the correct output should look like.
   - **Regression problems**: we are trying to predict results within a continuous output.
     - Example: predicting house prices based on house size.
   - **Classification problems**: we are trying to predict results in a discrete output.
     - Tagging photos as 'cats' or 'dogs'.

2. **Unsupervised learning:** we try to approach problems with little or no idea what the results should look like.
   - Example: identifying meaningful patters in 2D data.

3. **Reinforcement learning:** an agent learns to make decisions in an environment by receiving rewards or penalties for its actions.
   - Example: in robotics, grasping objects or navigating through a space.

Unsupervised learning

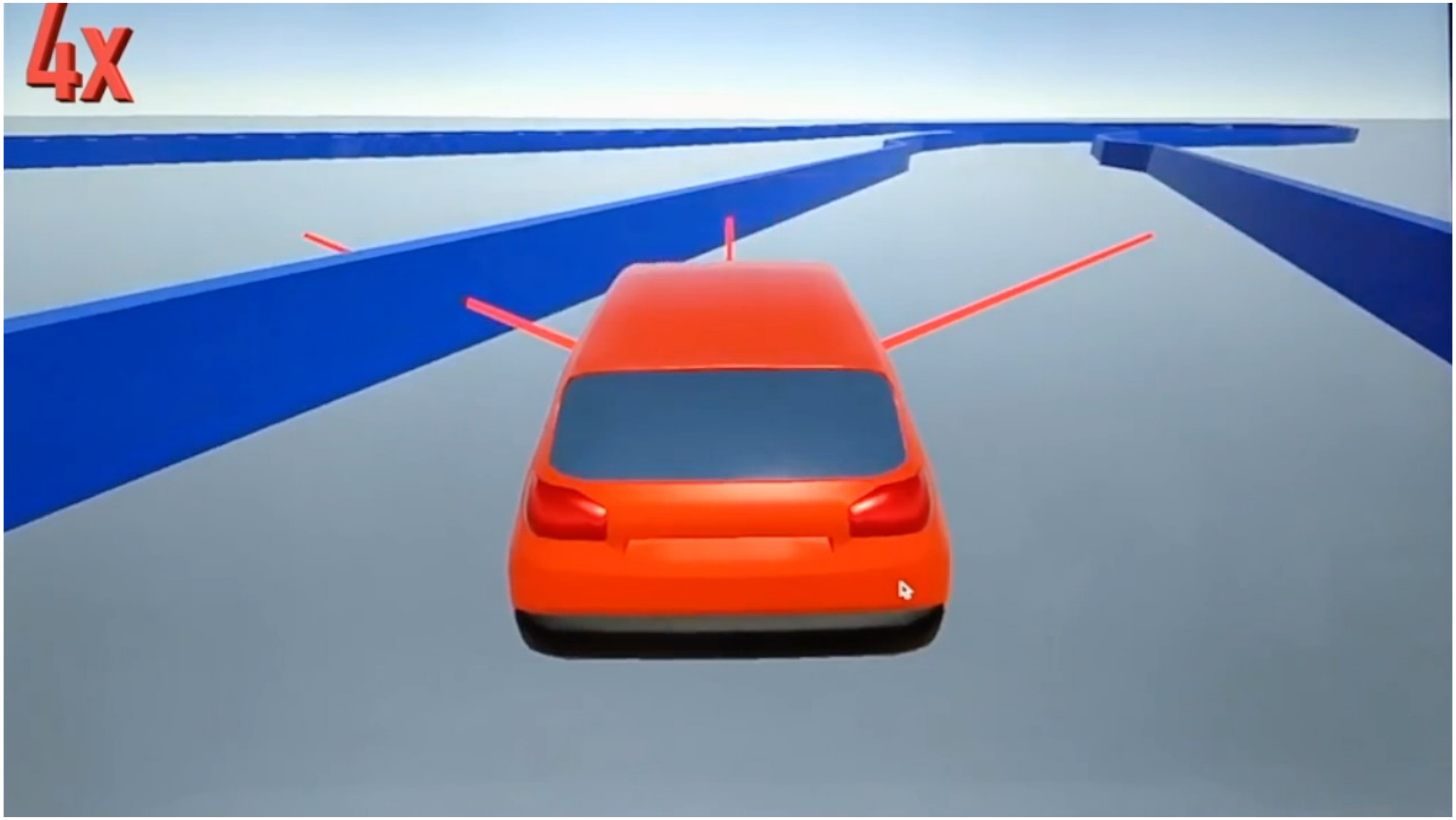
- Example: K-means clustering.



Source: [Codeacademy]

Reinforcement learning

- Example: self-driving car.



Source: Youtube

Supervised learning: regression

$$Y = X^T \beta + \varepsilon$$

$$\beta = [(X^T X)^{-1} X^T]\, Y$$

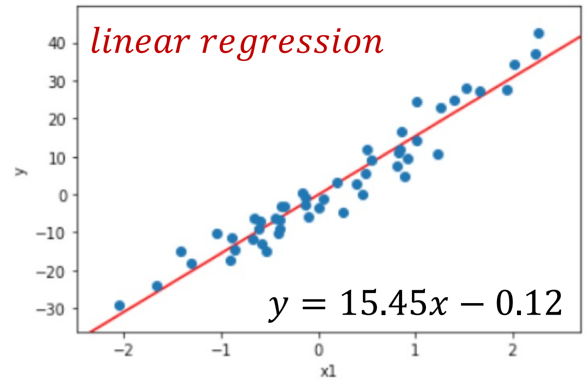
Learn $\beta \rightarrow \beta_0 = -0.12, \beta_1 = 15.45$

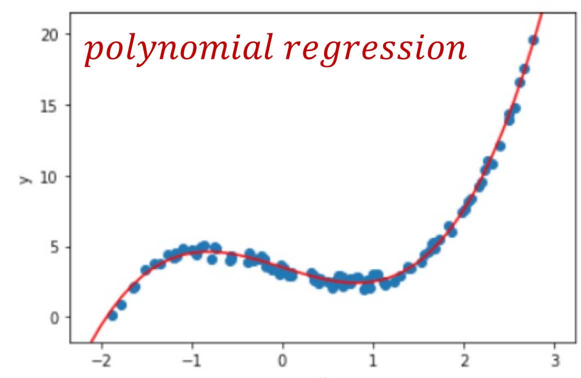
- The goal of regression is to find a function that best describes the relationship between the input (features) and the continuous output (target).
  - The function is typically represented by a straight line or a curve, and it is chosen to minimize the difference between the predicted values and the actual values.

- The most commonly used regression techniques are linear regression and polynomial regression.
  - Linear regression tries to fit a straight line to the data, while polynomial regression fits a curve of a higher degree to the data.

- To evaluate the performance of a regression model, we typically use metrics such as mean squared error (MSE).
  - MSE measures the average squared difference between the predicted and actual values.

- Examples of regression applications include predicting stock prices, weather forecasting, and medical diagnosis.





Supervised learning: classification

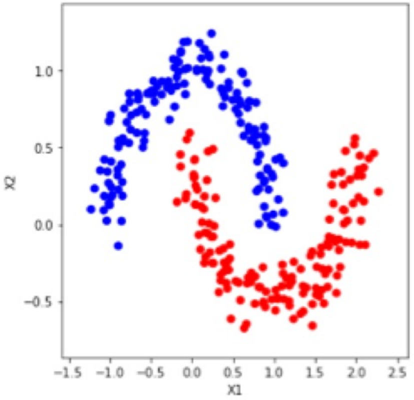
- The goal of classification is to find a function that maps the input features to a discrete output variable (class label). The function is typically represented by a decision boundary that separates the different classes in the feature space.

- The most commonly used classification techniques are logistic regression, decision trees, support vector machines (SVM), K-nearest neighbours (KNN), and neural networks.

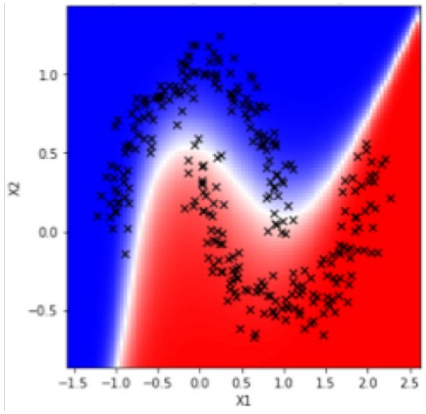
- To evaluate the performance of a classification model, we typically use metrics such as accuracy, precision, recall, and F1-score.
  - Accuracy measures the percentage of correctly classified instances.
  - Precision and recall measure the trade-off between false positives and false negatives.
  - F1-score is the harmonic mean of precision and recall.

- Examples of classification applications include image recognition, spam detection, sentiment analysis, and medical diagnosis.

Blue: class A,
Red: class B



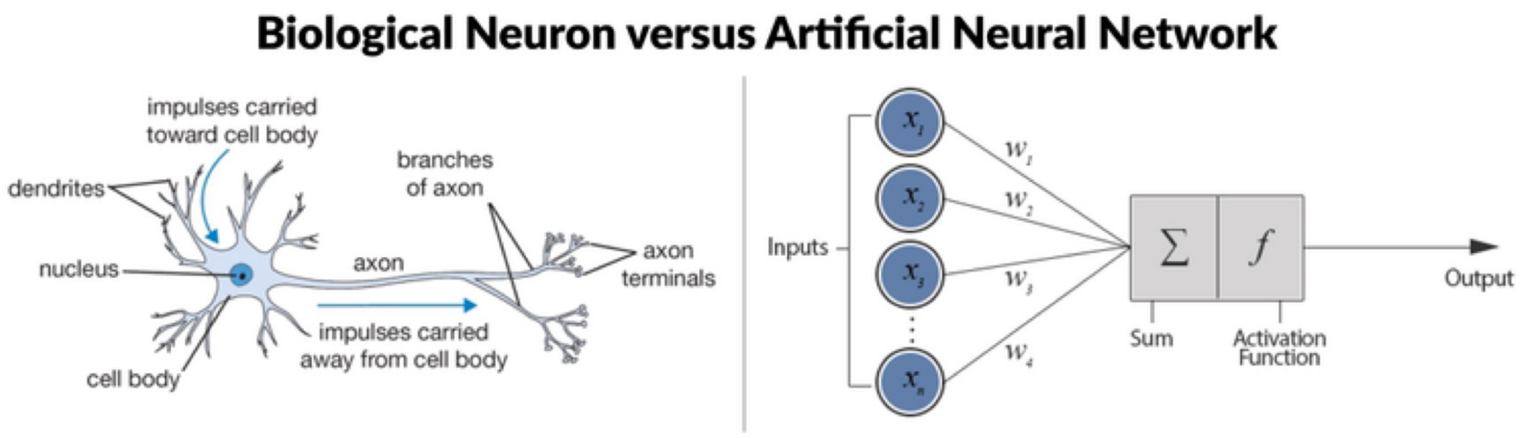
Classification example:



Neural networks

- Inspired by the nervous system (and not exactly by how the brain works).

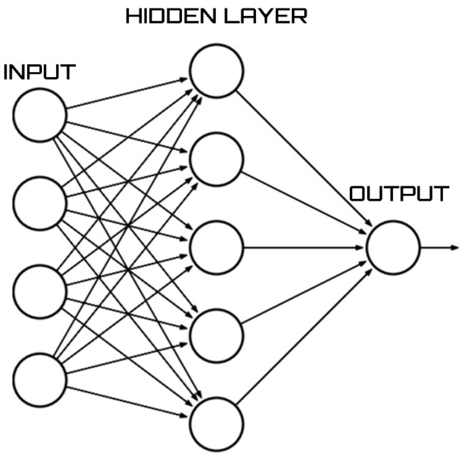
- The simplest neural network model is the perceptron (F. Rosenblatt, 1958).
  - Perceptron: mathematical model inspired by biological neurons.
  - Typically used for binary classification tasks.
  - The perceptron algorithm is a building block for more complex neural networks and deep learning models.



Source: J. Roell. 2017

Neural networks

- Neural networks refer to models consisting of multiple layers and multiple neurons per layer.
    - The layers located between the input and output layers are known as hidden layers.
    - Increasing the number of neurons and layers increases the model's capacity to solve more complex problems.



- Neural networks efficiently parameterise a multi-dimensional space. To provide a representative sample of the parameter space, it is crucial to have a sufficient number of training examples (large datasets!).

- Neural networks are trained using the forward-backward-propagation procedure.
    1. For each input example, the network calculates a prediction through forward propagation.
    2. The error is then calculated using the prediction and the actual label of the event. Finally, the network parameters (weights) are readjusted to minimize the error through backward propagation.
    3. This process is repeated iteratively until convergence, improving the model's ability to make accurate predictions.

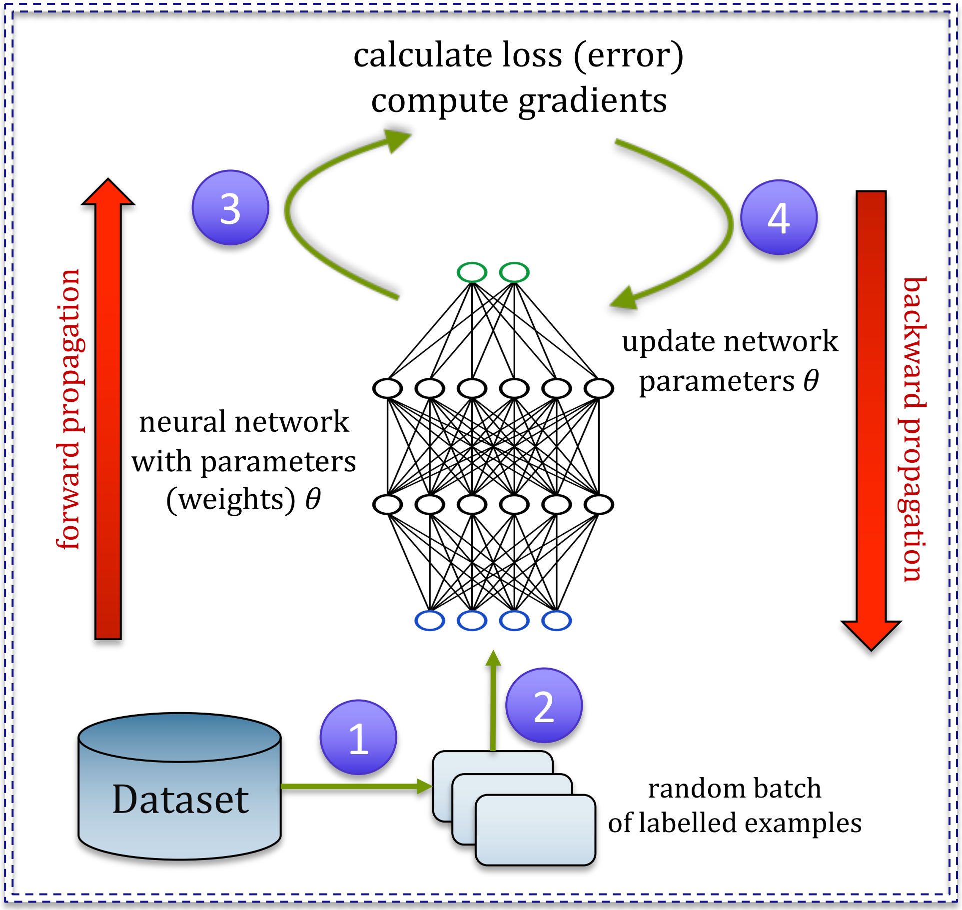
Neural networks: training

- Neural network training works by iteratively adjusting the weights of the network to minimize the difference between the predicted output and the true output. This is done through forward and backward propagation.

- **Forward propagation** involves feeding input data through the network to compute an output.

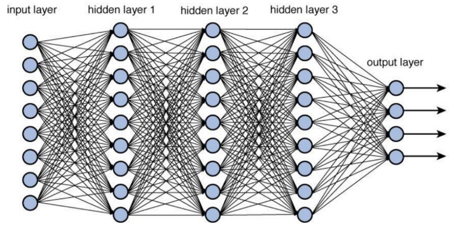
- **Backward propagation** involves computing the gradient of the error with respect to the weights of the network.
    - The weights are updated in the opposite direction of the gradient using an optimisation algorithm such as stochastic gradient descent.

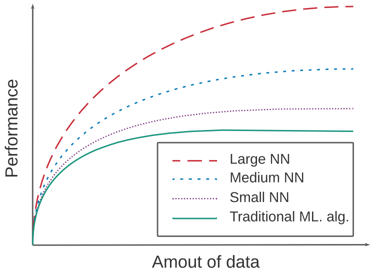
- The process of forward and backward propagation is repeated for multiple epochs until the error is minimised and the network is trained to accurately predict outputs for new inputs.



Deep learning

- Deep learning (DL) refers to neural networks with multiple layers (deep neural networks) aimed to solve complex problems.
  - Real-world applications of deep neural nets often have tens or even hundreds of layers, enabling them to capture complex patterns in data that traditional methods struggle to handle.
  - Classical machine learning techniques, particularly in the field of computer vision and natural language processing, have become less relevant due to the impressive performance of deep neural nets [O'Mahony et al.].



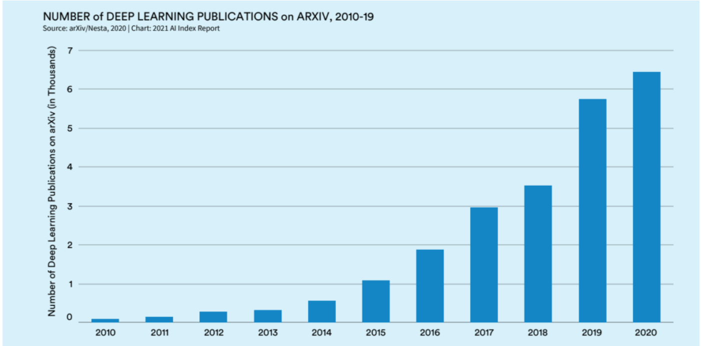


- Although have been around for decades, it wasn't until recently that they became feasible to run on large datasets using available hardware.
  - The DL revolution began in 2012, when Krizhevsky et al. achieved a breakthrough in image classification by significantly reducing the classification error of a dataset with 10,000 categories and 10 million images, using a deep neural network [DOI:10.1145/3065386].

Deep learning

- Most of the latest advances in AI are due to deep learning.
    - Powering daily services.
        - Translators, social media recommendations, maps, spam filters, fraud prevention, virtual assistants, etc.
    - And behind emerging technologies:
        - E.g., Autonomous cars, chat bots, image generation.
    - Some experts refer to deep learning as the new industrial revolution: (https://www.youtube.com/watch?v=yWa9i1ZaSes).

- Neural networks can be represented by combinations of matrix operations.
    - Input data stored as vectors or matrices.
    - Each layer extracts characteristics of the data and passes it to the next layer.
    - Enables sophisticated data transformations and feature extraction.
    - The power of GPUs (Graphics Processing Units) has revolutionized deep learning by enabling the processing of large datasets and complex neural networks with lightning-fast speed, making it possible to train and deploy deep learning models for a variety of real-world applications.

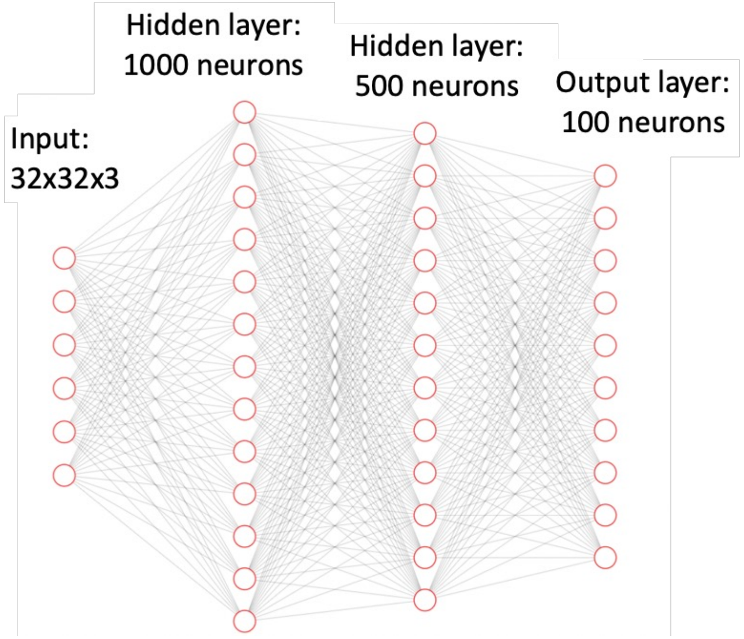


Source: AI index

Implementation

- Fortunately, deep-learning frameworks such as PyTorch and TensorFlow offer a user-friendly API to facilitate the experimentation with neural networks.
  - Most of the math (matrix operations, gradient calculations, etc) are included in a transparent way to the user.

- Example using TensorFlow:




```python
#neural network model
def create_simple_nn():
  model = Sequential()
  # Flatten function for vectorice input
  model.add(Flatten(input_shape=(32, 32, 3), name="Input_layer"))
  model.add(Dense(1000, activation='relu', name="Hidden_layer_1"))
  model.add(Dense(500, activation='relu', name="Hidden_layer_2"))
  model.add(Dense(100, activation='softmax', name="Output_layer"))

  return model
```

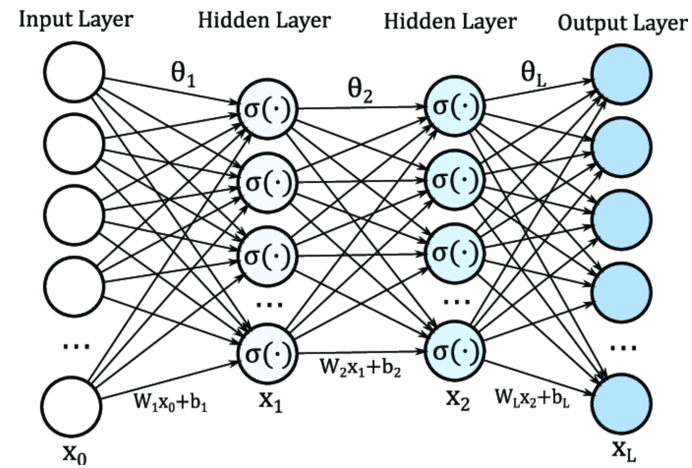
Trivial implementaiton!

Overview

1. Introduction.
2. Foundations.
3. **Applications.**
4. Challenges and future directions.
5. Conclusion.

Fully-connected neural networks (FCNN)

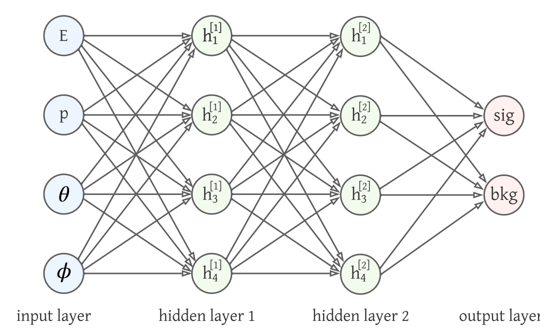
- Fully-connected neural networks (FCNNs), also known as dense neural networks or multi-layer perceptrons (MLPs), are a type of artificial neural network where each neuron in one layer is connected to every neuron in the next layer.



DOI:10.1109/ACCESS.2019.2923321

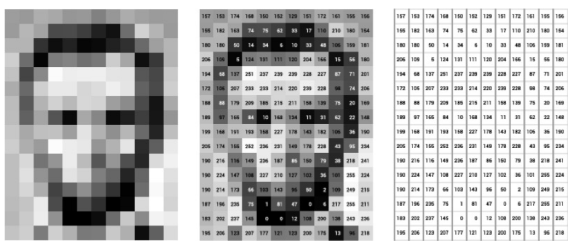
- FCNNs are commonly used for tasks such as classification and regression, and can be used in combination with other neural network architectures for more complex applications such as computer vision and natural language processing.

- Example an application of FCNNs in particle physics: “classification of particles as signal or background based on their characteristics measured in a particle detector”:
  - Inputs:
    - Energy, momentum, direction.
  - Output:
    - 1 (signal), or 0 (background).
  - Network architecture:
    - 2 hidden layers of size 4 (+input and output).

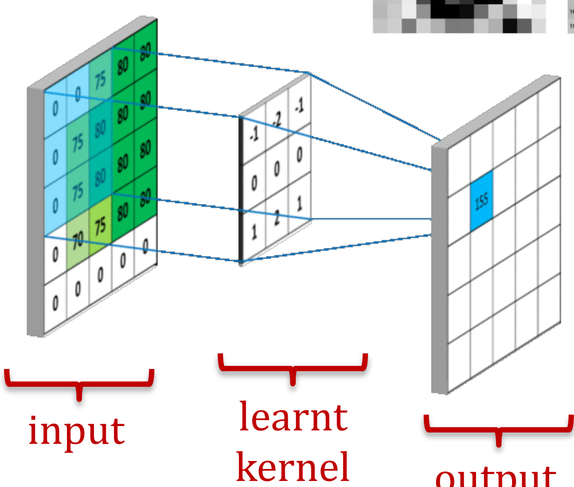


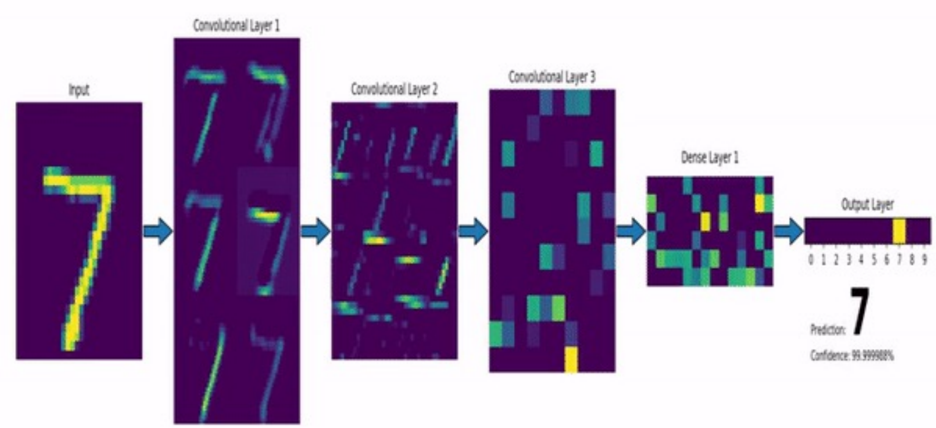
Convolutional neural networks (CNNs) in computer vision

- Computer vision is the field of computer science that tries to interpret and understand images or videos.
- Convolutional neural networks, or CNNs, are a type of neural network architecture specifically designed for image recognition tasks in computer vision.
  - CNNs use a series of convolutional layers to extract features from images, followed by pooling layers to reduce dimensionality and fully-connected layers for classification.
  - CNNs have achieved state-of-the-art performance in a variety of computer vision tasks, including object detection, image segmentation, and facial recognition.



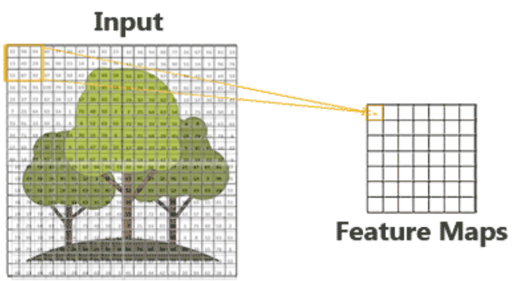
[Source: Openframeworks]





Convolutional neural networks (CNNs) in computer vision

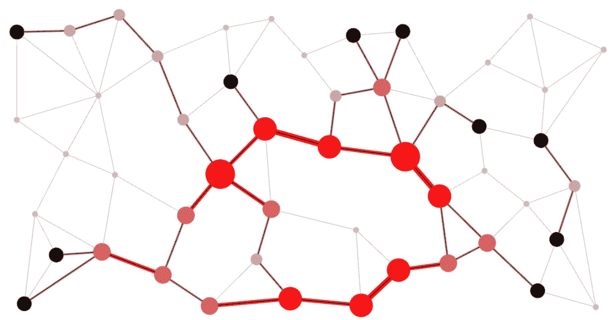
- Computer vision is the field of computer science that tries to interpret and understand images or videos.
- Convolutional neural networks, or CNNs, are a type of neural network architecture specifically designed for image recognition tasks in computer vision.

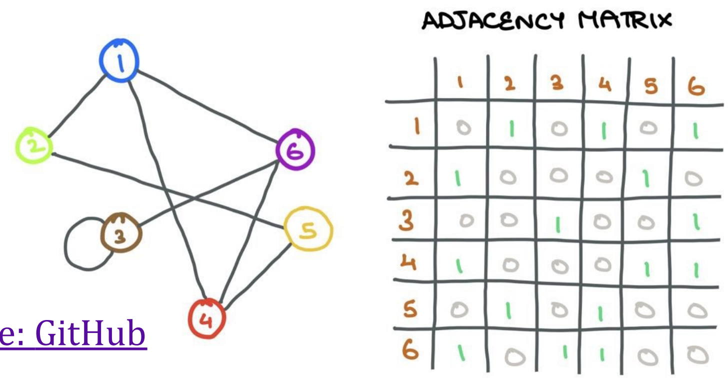


[Source: Adatis]

Graph neural networks (GNNs)

- Graph Neural Networks (GNNs) are a type of deep learning model that, Unlike traditional neural networks like Multilayer Perceptrons (MLPs) or Convolutional Neural Networks (CNNs), can learn and process information from the complex structure of graphs, which makes them suitable for tasks such as node classification, link prediction, and graph classification.





Source: GitHub

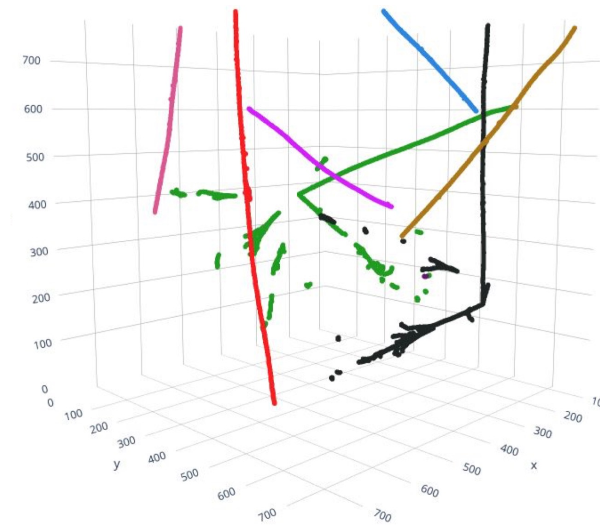
- Compared to MLPs and CNNs, GNNs can handle graph data with variable size and structure, which makes them more suitable for applications involving relational data. GNNs can also capture the local and global structure of graphs and can learn to aggregate information from neighbouring nodes and edges.

- Some applications of GNNs include social network analysis, recommendation systems, or bioinformatics. GNNs can also be used to model and reason about physical and biological systems, such as predicting the behaviour of proteins or designing new molecules.

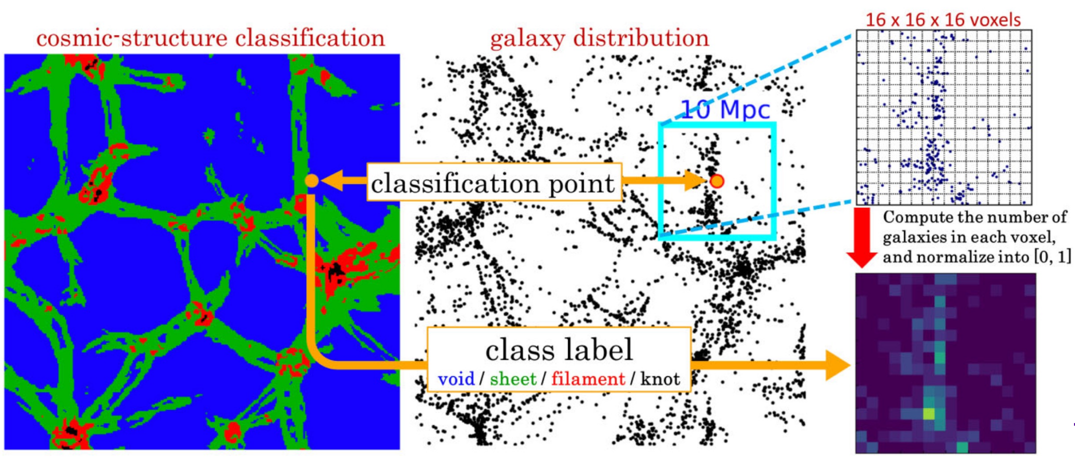
CNNs and GNNs: applications in physics

- In physics, CNNs and GNNs have been used for a variety of applications, including:
    - Anomaly detection.
    - Signal vs background discrimination.
    - Galaxy identification and classification.
    - Neutrino interaction classification.
    - Pileup mitigation.
    - Event energy reconstruction.
    - Track vs shower separation.
    - Particle tracking.
    - Etc.

- Some of the above applications will be shown at this workshop!



K. Terao, 2020



S. Inoue et al., 2022

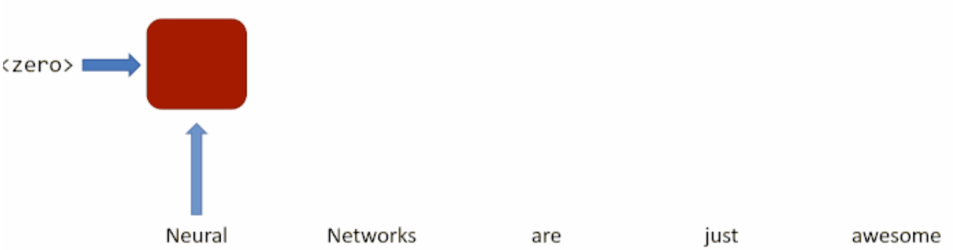
Recurrent neural networks (RNNs) in natural language processing

- Recurrent neural networks, or RNNs, are a type of neural network architecture that are designed to process sequential data.

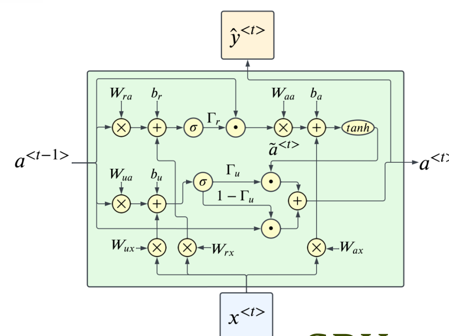
- Unlike FCNNs or CNNs, RNNs have a "memory" that allows them to maintain information about previous inputs and use it to influence the processing of current inputs.

- There are several types of RNNs, including Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRUs), which vary in their memory mechanisms.

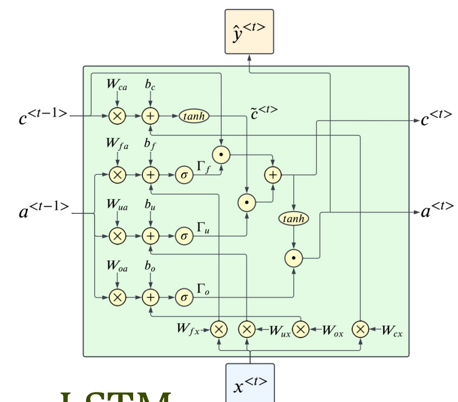
- RNNs have achieved great performance in a variety of natural language processing tasks, including language translation, speech recognition, and sentiment analysis.



Source: Reflect



GRU unit



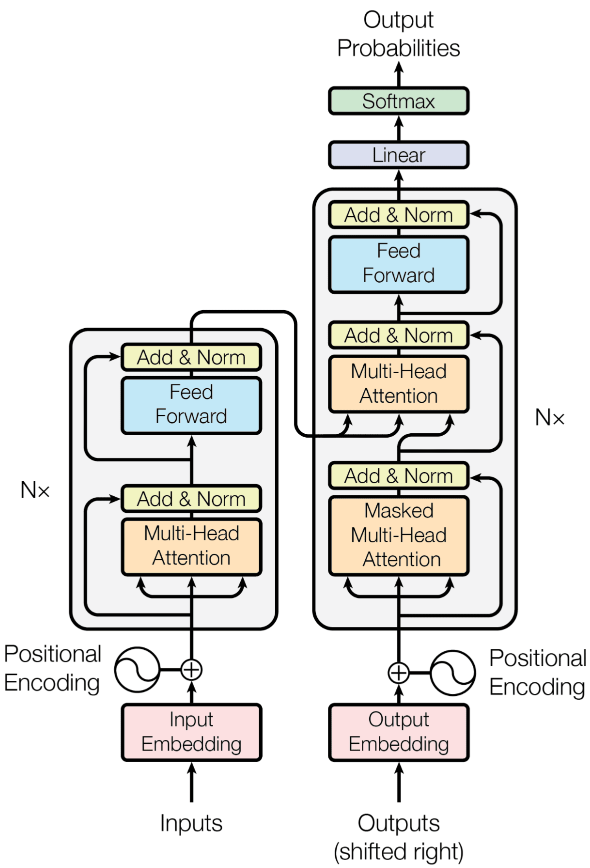
LSTM unit

Transformers

- Transformers are a type of deep neural network architecture that have revolutionised natural language processing (NLP) and other sequence modeling tasks.

- They were first introduced in the 2017 paper "Attention is All You Need" by Vaswani et al.(arXiv:1706.03762) and have since become one of the most popular deep learning models.

- Transformers have been successfully applied to a wide range of NLP tasks, including machine translation, text summarization, sentiment analysis, and named entity recognition.
  – ChatGPT is in a Transformer.



Transformers: input embedding

- The first step in the Transformer model is to convert the input sequence of tokens (words, characters, etc.) into a sequence of dense vectors called embeddings.
  - These embeddings capture the meaning of the tokens and their relationships to each other.
  - The input embeddings are fed into the self-attention mechanism, which is the core of the Transformer model.

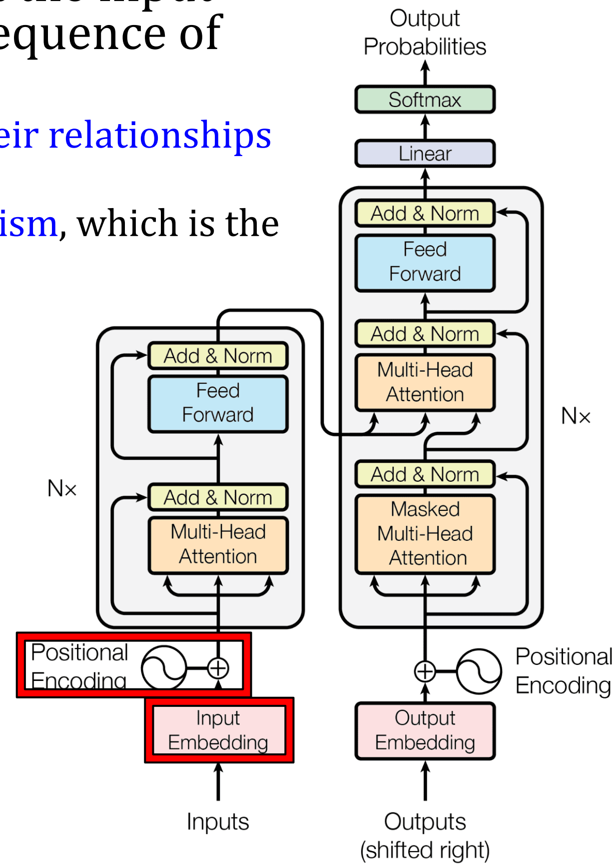
Token	Embedding
...	...
a	[0.40, 0.15, 0.99]
...	...
am	[0.11, 0.28, 0.11]
...	...
I	[0.59, 0.23, 0.02]
...	...
student	[0.12, 0.35, 0.61]
...	...

I	0.59	0.23	0.02
am	0.11	0.28	0.11
a	0.40	0.15	0.99
student	0.12	0.35	0.61

4x3

Input Embedding

"I am a student"



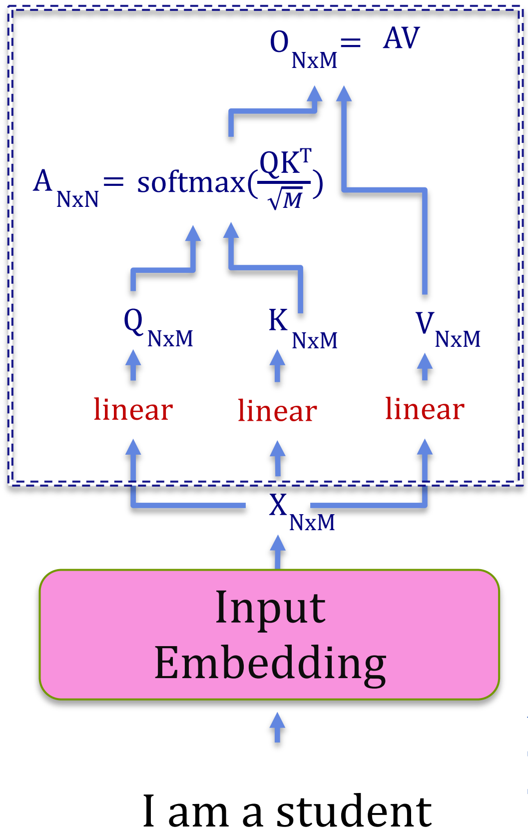
Transformers use **positional encoding**, adding a set of sinusoidal functions to the input embeddings to provide information about the relative positions of tokens, as they lack a built-in notion of sequence order.

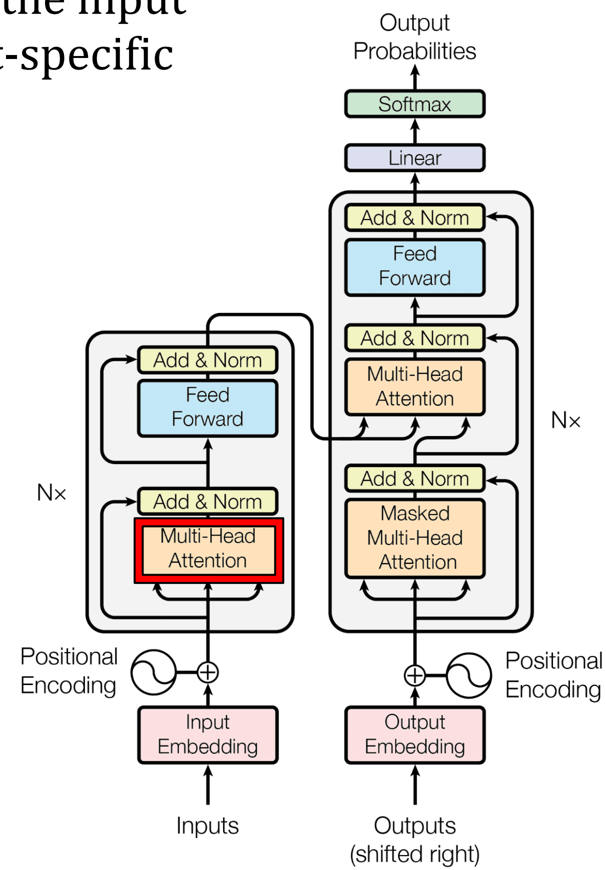
Transformers: self-attention

- Self-attention is a mechanism that allows each token in the input sequence to attend to all other tokens and learn context-specific representations.

	I	am	a	student
I	0.4	0.1	0.2	0.3
am	0.3	0.6	0.0	0.1
a	0.2	0.1	0.5	0.2
student	0.4	0.1	0.1	0.4

A_{NxN}





- The self-attention mechanism computes a weighted sum of the input embeddings, where the weights are learned based on the similarity between the tokens.
- Unlike memory mechanisms in RNNs, self-attention enables the Transformer model to capture long-range dependencies and handle variable-length input sequences.

I	0.59	0.23	0.02
am	0.11	0.28	0.11
a	0.40	0.15	0.99
student	0.12	0.35	0.61

X_{NxM}

While self-attention is a key component of the Transformer architecture, it is important to note that Transformers use **multi-head attention**, which allows the model to attend to information from different representation subspaces.

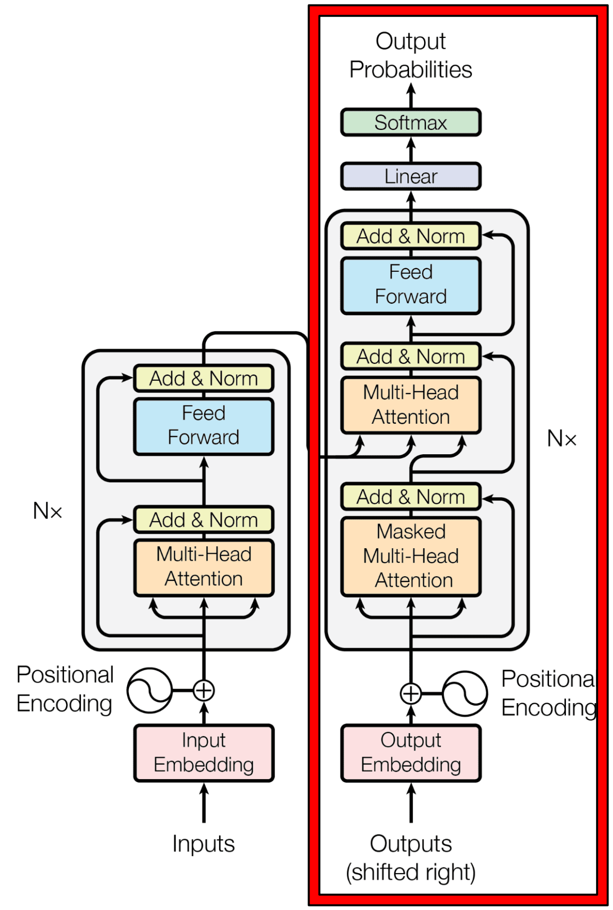
Transformers: decoder

- As seen, the encoder component processes the input sequence and produces a set of encoded representations that capture the contextual information of each token in the sequence.

- Transformers can be used in both encoder and decoder configurations for sequence-to-sequence tasks (e.g., text generation).

- The decoder is a variant of the Transformer-encoder model that is used to generate the output sequence from the encoded input sequence.

- The decoder uses masked self-attention to attend only to the previously generated tokens in the output sequence, ensuring that the model does not cheat by looking ahead in the sequence.
  - K and V are the encoder representations in the second multi-head attention block.



RNNs and Transformers: applications in physics

- In physics, RNNs and Transformers have been used for a variety of applications, including:
  - Particle decay prediction.
  - Particle track fitting.
  - Vertex finding.
  - Jet identification.
  - Analysis of unordered set of particles.
  - Etc.

- Although Transformers were initially developed for natural language processing (NLP) tasks, they have found applications in a wide range of domains beyond NLP as well.
  - Transformers have been applied to computer vision tasks such as image classification, object detection, and segmentation. Vision Transformers (ViT) is one such example that can achieve state-of-the-art results on several benchmark datasets.

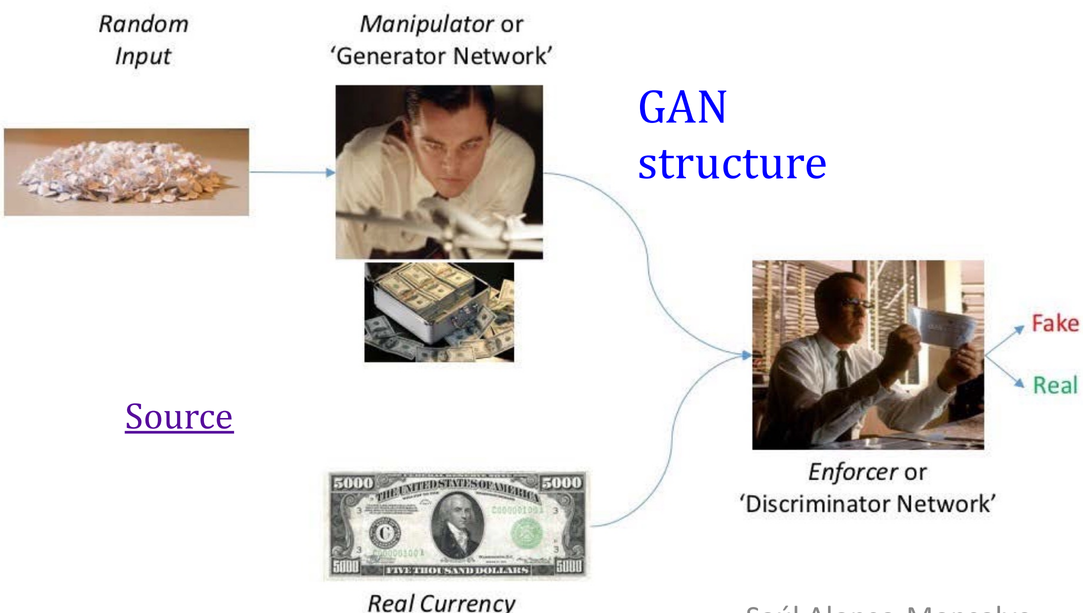
Choosing the right architecture

- When choosing a neural network architecture, consider the following factors:
  - Data type and task complexity: different architectures are designed to handle different types of data and tasks. For example, CNNs are best for image and video recognition, while RNNs and Transformers are best for natural language processing.
  - Amount of training data: some architectures require large amounts of data to train effectively, while others can achieve good results with smaller amounts of data.
  - Network capacity and computing resources: having more model parameters can potentially improve a model's performance, as it allows the model to learn more complex representations of the data. However:
    - Larger models require more computational resources to train and inference, which can be a practical limitation in some applications.
    - As the number of parameters increases, so does the risk of overfitting the training data, which can lead to poor performance on new, unseen data.
    - Optimisation algorithms can also struggle with larger models due to increased computation time and the possibility of getting stuck in local minima.

- **Overall, the best architecture for a neural network depends on a variety of factors and requires experimentation and iteration to find the optimal solution.**

Extra: Generative models

- Generative models can create new data samples that resemble the input data distribution.
- Two main types of generative models are Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs).
  - GANs consist of a generator network and a discriminator network that are trained together to generate realistic samples.
  - VAEs encode input data into a latent space and generate new samples by sampling from this latent space and decoding the samples back into the original input space.



GAN structure

Source

Source

Generative models

- Particle Flows and Stable Diffusion are two newer types of generative models that have shown promising results.
    - Particle Flows transform an initial distribution of particles to a target distribution through a series of continuous transformations.
    - Stable Diffusion uses a multi-step diffusion process with controlled noise levels, allowing the algorithm to produce high-quality and diverse images.

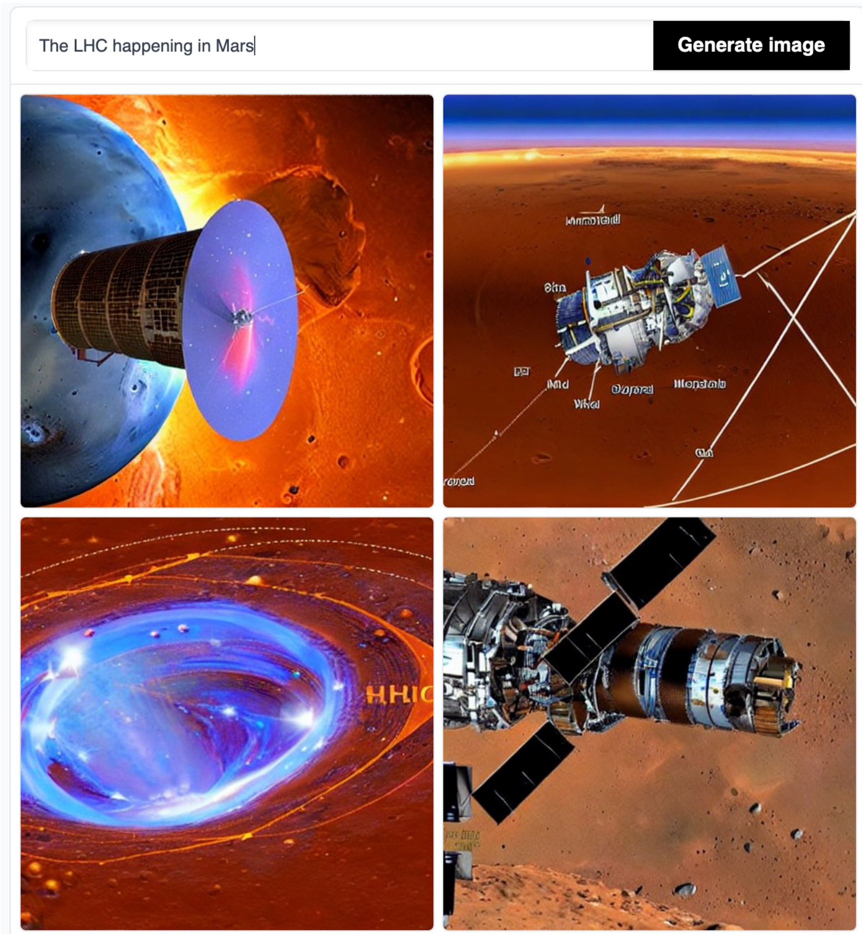
- Generative models have applications in various areas such as data augmentation, super resolution, or style transfer.

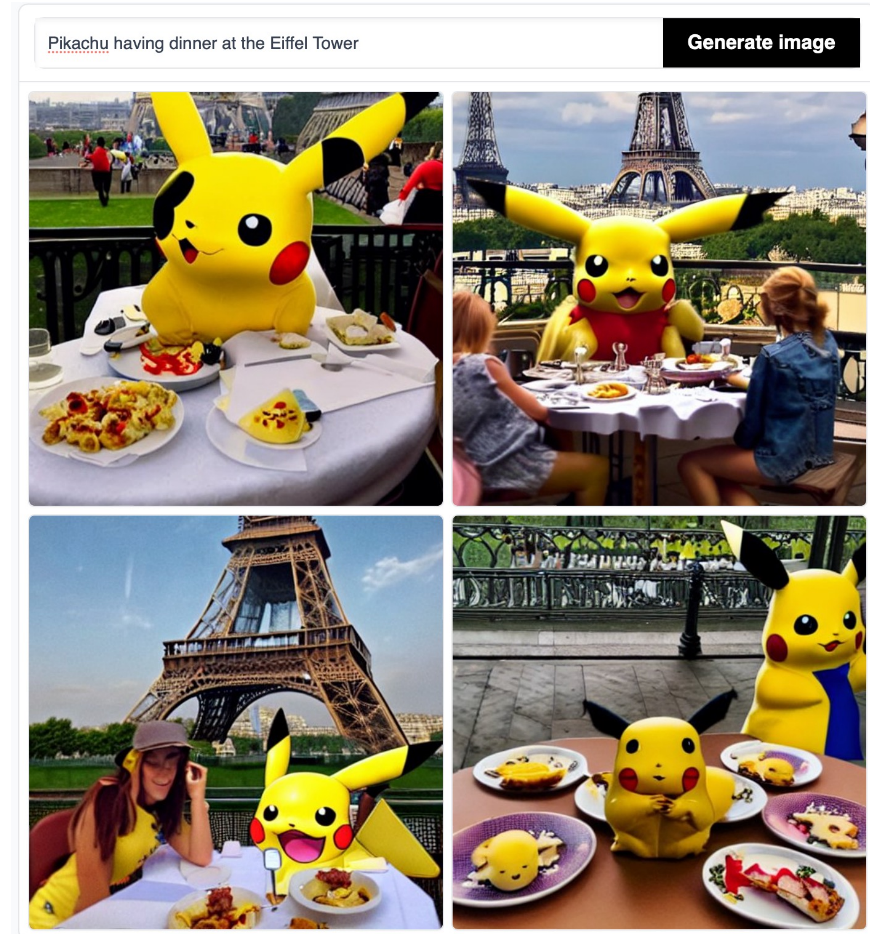
- In particle physics, generative models can be used to simulate particle interactions and generate new data samples for analysis.
    - Generative models are in general much faster than Montecarlo simulations.

- In astrophysics, generative models can be used to generate simulations of the universe and the distribution of dark matter.

Example of Stable Diffusion: source

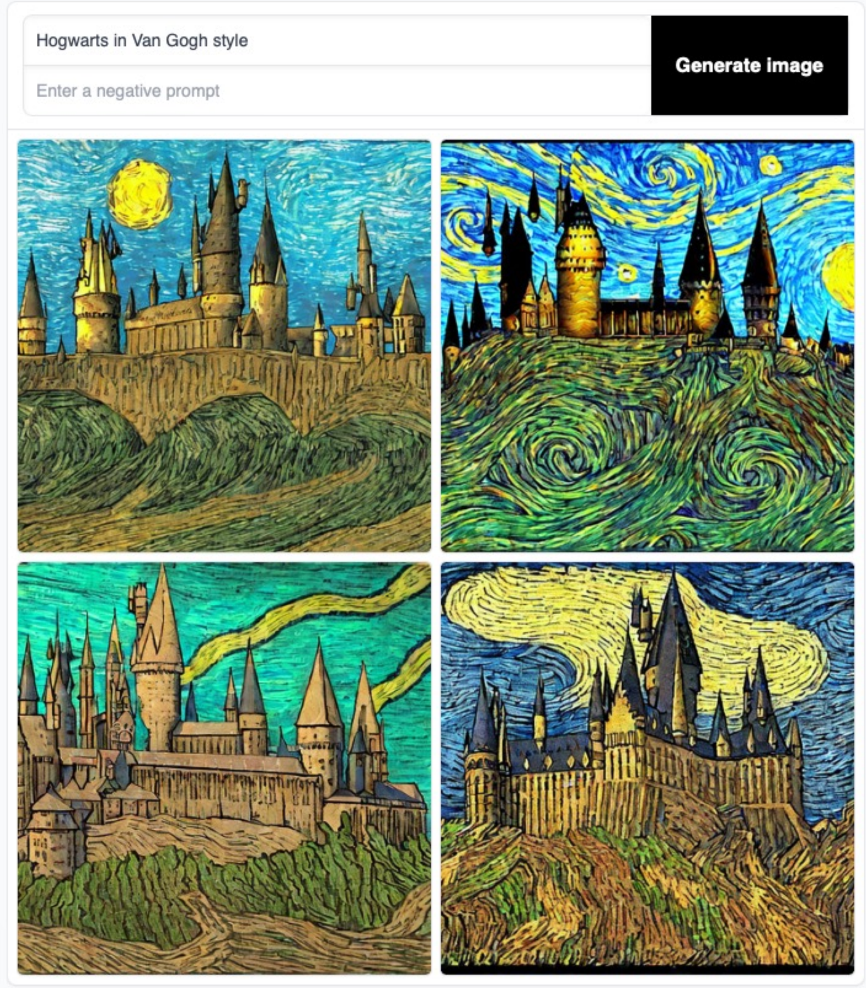
Examples of Stable Diffusion





Credit: https://stablediffusionweb.com

Examples of Stable Diffusion



Credit: https://stablediffusionweb.com

Overview

1. Introduction.
2. Foundations.
3. Applications.
4. **Challenges and future directions.**
5. Conclusion.

Sparse data

- In particle physics and astrophysics, data is often sparse, due to the nature of the objects being studied or the particles detected.

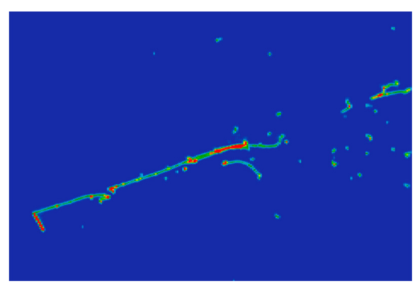
- This poses a challenge for machine learning, as traditional machine learning algorithms are designed to work with dense data. To address this, researchers are developing new algorithms and techniques specifically tailored to sparse data.
  - For example, one approach is to use Submanifold Sparse Convolutional Networks (SSCN), where the convolution operation is performed only on the non-zero elements of the sparse data, resulting in an efficient and accurate representation of the data.
  - Another approach is to use graph-based methods, which can effectively capture the relationships between entities in sparse data.

"Dense" image

- **All pixels might be helpful for the classification.**
- **Ideal for standard CNNs.**

Source: https://www.britannica.com/

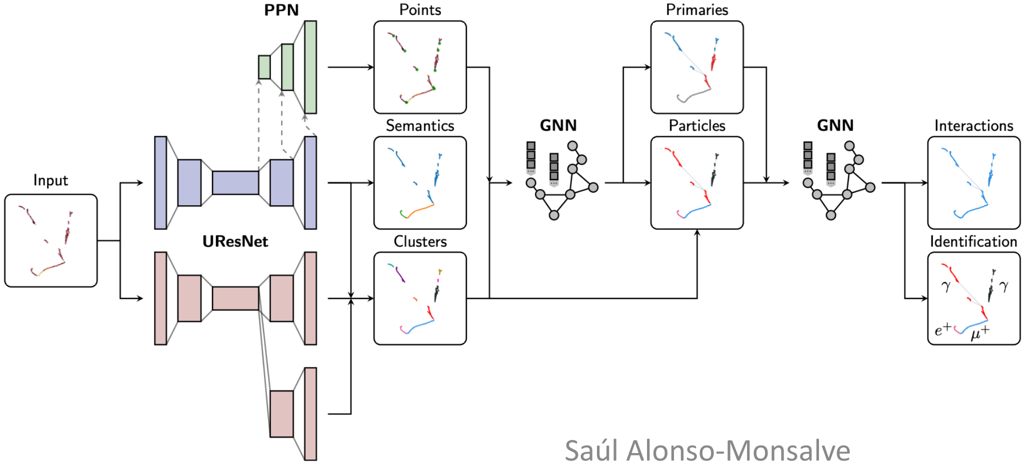
"Sparse" images



- **Most pixels are background.**
- **A standard CNN would perform loads of useless computations.**

Automated physics analyses

- Machine learning can be used to automate certain aspects of physics analyses, such as the data preprocessing, event selection, reconstruction, etc (or event for calibration/modelling in particle physics detectors).
- This can save significant time and resources, and can also help ensure that analyses are reproducible and consistent.
  - For example, machine learning can be used to automatically detect and remove background events in particle physics experiments, or to identify and classify different types of galaxies in astrophysics.
  - It can also help reduce human bias in the analysis process.
- There are many remarkable advances in this regard.
  - Despite promising advances in this area, integrating machine learning techniques into the analysis flow of physics experiments can be challenging due to technical, logistical, and sometimes skeptical barriers.

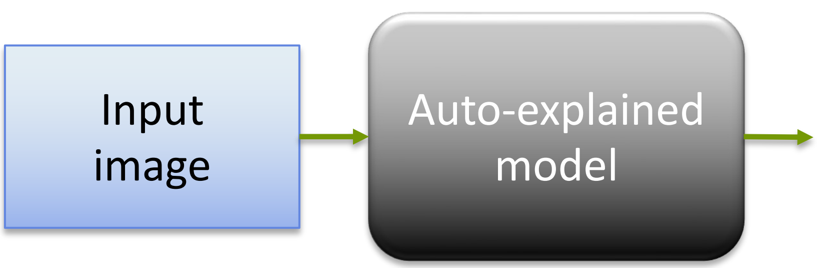


"Scalable, End-to-End, Deep-Learning-Based Data Reconstruction Chain for Particle Imaging Detectors" - F. Drielsma et al. 2021

Addressing the interpretability and explainability of machine learning models

- Addressing the interpretability and explainability of machine learning models in particle physics and astrophysics is a significant challenge.
  - It is not enough to have a model that can accurately predict outcomes; scientists need to know how and why the model is making these decisions.
  - Developing methods for understanding and interpreting machine learning models is an area of active research.

- Auto-explained models are models that can explain their decisions in a way that is understandable to humans.
  - This is important for applications where it is critical to know why the model is making a certain decision, such as in medical diagnosis.
  - In particle physics and astrophysics, auto-explained models can help scientists understand, for instance, why a certain object was classified in a certain way.



"*This is an electron because it has a sudden high energy deposit, and because it produces a characteristic electromagnetic shower near the end of the track*."

Robustness against systematic uncertainties and simulation mismodellings

- In particle physics and astrophysics, there are often systematic uncertainties related to the measurements, as well as mismodellings in the simulations.
  - These uncertainties can arise from a variety of sources and can affect the accuracy and precision of the measurements and simulations in these fields.

- Machine learning models can be biased or inaccurate as a result.
  - To address this, researchers are developing methods to make machine learning models more robust against these uncertainties and mismodellings.

- One approach is to use adversarial training, where the model is trained to be robust against adversarial examples that are specifically designed to trick the model.
  - Another approach is to incorporate physics-based constraints or priors into the model (e.g. penalty terms in the loss function), to help ensure that the model is consistent with known physics.
  - Adversarial trainings can also be used with detector data to refine the ML models in an unsupervised way.

Generative models to replace simulations

- Generative models are machine learning models that can generate new data that is similar to the training data.

- In particle physics and astrophysics, generative models can be used to generate new simulated data, which can be used to supplement or eventually replace existing simulations.
  - This can save significant time and resources, and can also help address uncertainties and mismodellings in the simulations.
  - Current work cannot fully-replace current simulations yet, but are more suited for fast prototyping.

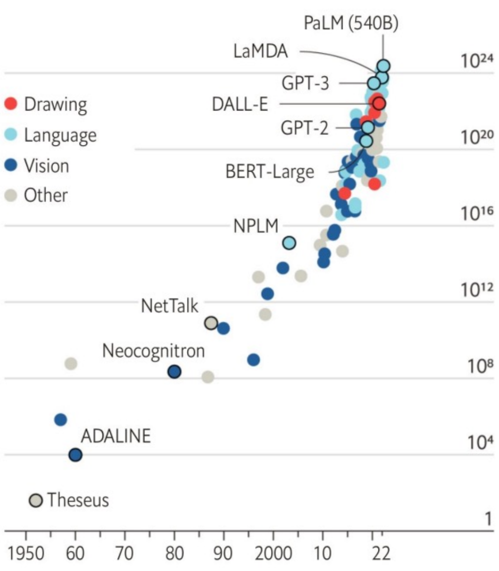
- Despite the limitations, generative models are a promising area of research in HEP, and have the potential to revolutionize the way simulations are performed in the field.
  - Although Stable Diffusion shows promise for replacing simulations in HEP experiments, its current computational cost remains a challenge.

Large models and infrastructure

- Particle physics and astrophysics generate vast amounts of data, and machine learning models trained on this data can be very large and complex.
  - This requires significant computational resources and infrastructure to train and deploy these models.
  - Investing in large-scale infrastructure and end-to-end systems for machine learning in particle physics and astrophysics is an important future direction.
- We are very far away to state-of-the-art applications:
  - A typical deep learning model in physics usually has never more than a few million parameters.
  - GPT-3.5 (the model behind ChatGPT) was trained for ~12-18 months on a supercomputer with ~10,000 GPUs and ~285,000 CPU cores (~1 billion dollars to rent) and has 175 billion parameters. Source.
- Beware of the significant environmental impact caused by the large carbon footprint of deep learning models.

AI training runs, estimated computing resources used

Floating-point operations, selected systems, by type, log scale



Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

Real-time models

- Finally, another important future direction is developing machine learning methods that can work in real-time or near-real-time.
- This is especially important for particle physics experiments, where researchers need to preselect (trigger) data as it is collected.
- Developing machine learning algorithms that can operate in real-time is an important present and future challenge.
- Many current applications (ATLAS, IceCube, LIGO, Dark Energy Survey, etc).
    - They use specialized hardware, such as Field-Programmable Gate Arrays (FPGAs) or Graphics Processing Units (GPUs), to achieve the required computational performance and low-latency response times.
    - They employ various techniques to optimize performance, such as reducing the precision of the model's parameters (e.g., using 16-bit floating-point arithmetic instead of 32-bit) or using model compression techniques to reduce the model's size and memory footprint.

Overview

1. Introduction.
2. Foundations.
3. Applications.
4. Challenges and future directions.
5. **Conclusion.**

Summary and conclusion

- Machine learning is an essential tool in particle physics and astrophysics research.
- We have discussed the foundations of machine learning, including neural networks.
- We have explored the different types of machine learning and their applications in particle physics and astrophysics.
- Challenges for future research include dealing with sparse data, ensuring interpretability and explainability of models, addressing uncertainties, and creating generative models.
- Development of large models, infrastructure, and real-time models are also crucial for future research.
- Overall, machine learning has opened up new avenues of research, and addressing its challenges can lead to a deeper understanding of the universe.

Interesting links

- A visual introduction to machine learning: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/.
- Natural language processing course: https://www.youtube.com/playlist?list=PLo2EIpI_JMQvWfQndUesu0nPBAtZ9gP1o.
- “Catalog” of Transformers: https://arxiv.org/abs/2302.07730.
- Computer vision tool for anyone to use! : https://landing.ai.
- Consensus AI (evidence answers, useful for research): https://consensus.app.
- Ted Chiang's critique of the threat of superintelligence: https://www.buzzfeednews.com/article/tedchiang/the-real-danger-to-civilization-isnt-ai-its-runaway.

Recommended literature

- "*Understanding Machine Learning*", Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press.
- "*Deep Learning*", I. Goodfellow et al., MIT Press (2016): https://www.deeplearningbook.org/.
- "*Deep learning specialization*" (Coursera). DeepLearning.AI (2021): https://www.coursera.org/specializations/deep-learning.
- "*Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*", Aurelien Geron, O'Reilly Media (2017).
- "*Machine learning at the energy and intensity frontiers of particle physics*", A. Radovic et al., Nature (2018): https://doi.org/10.1038/s41586-018-0361-2.
- "*A Living Review of Machine Learning for Particle and Nuclear Physics*" (2021): https://iml-wg.github.io/HEPML-LivingReview/review/hepml-review.pdf.
- "*Physics-based Deep Learning Book*", N. Thuerey et al. (2021): https://physicsbaseddeeplearning.org.

PHYSICS AND MACHINE LEARNING: AN OVERVIEW

Saúl Alonso-Monsalve

ETH Zurich

IPA-ML

20 March 2023