

AI and ML for Chemists

Topic 5: Deep learning in chemistry

Pre-lecture Explore

Dr. Ganna (Anya) Gryn'ova

Deep learning

a subset of machine learning that uses multilayered **neural networks**, called deep neural networks, to simulate the complex decision-making power of the human brain.

AI:

- Systems that mimic human intelligence, including reasoning, decision-making, and problem-solving.
- Minimal human intervention.
- Robots, chatbots, autonomous vehicles, ...

ML:

- Teaching machines to find patterns in data, use them to make predictions or decisions, and improve automatically with experience.
- Requires human involvement for data preparation, model training, and optimisation.
- Recommenders, classifiers, predictors, ...

DL

Components

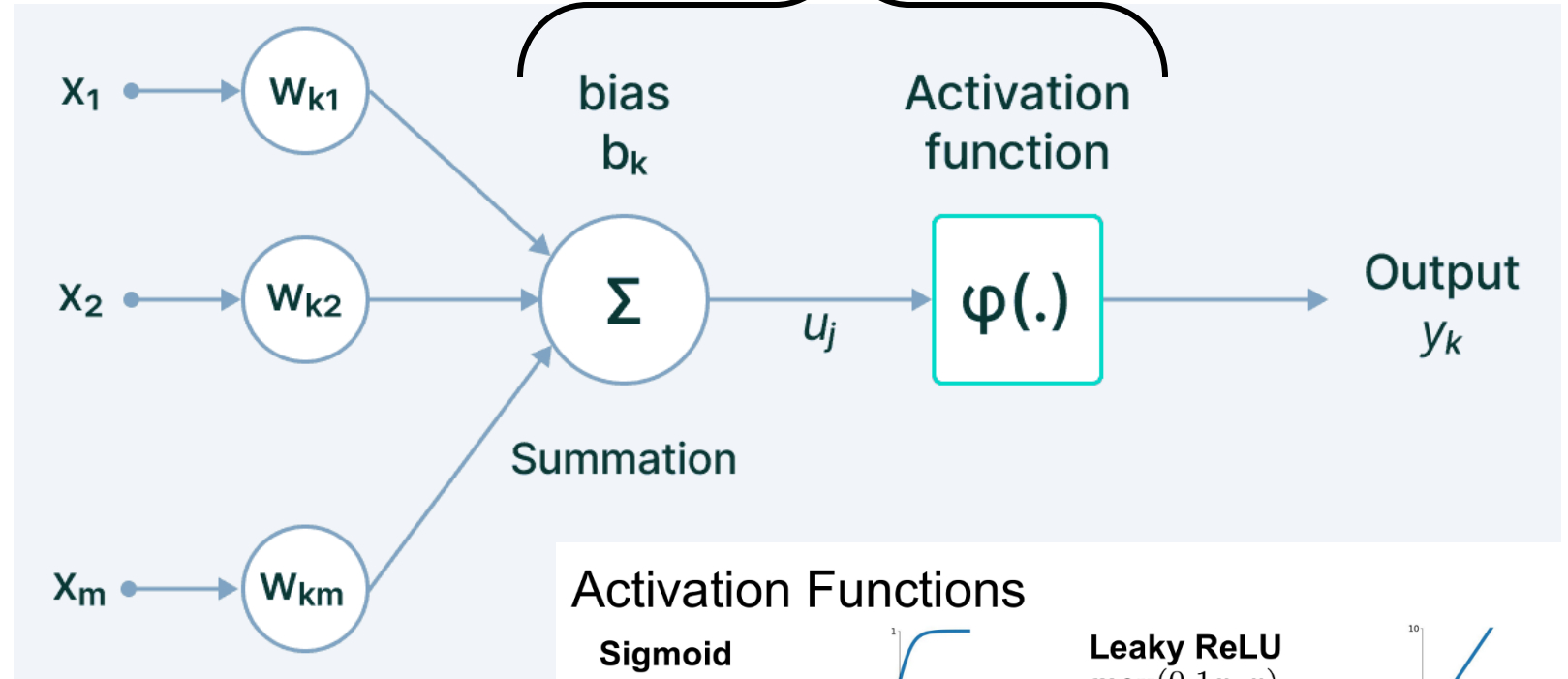
Weights (w_{ki}) give importance to those features that contribute more towards the learning. They represent the strength of the connections between neurons.

Transfer (summation) function combines multiple inputs into one output value so that the activation function can be applied.

Activation function introduces non-linearity to the network.

Biases shift the values produced by the activation function. They allow neurons to make predictions even when all inputs are zero.

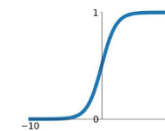
An artificial neuron – an individual building block (node) of a neural network.



Activation Functions

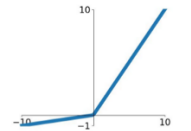
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



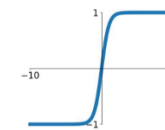
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

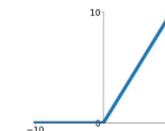


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

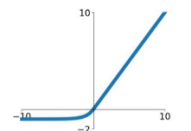
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



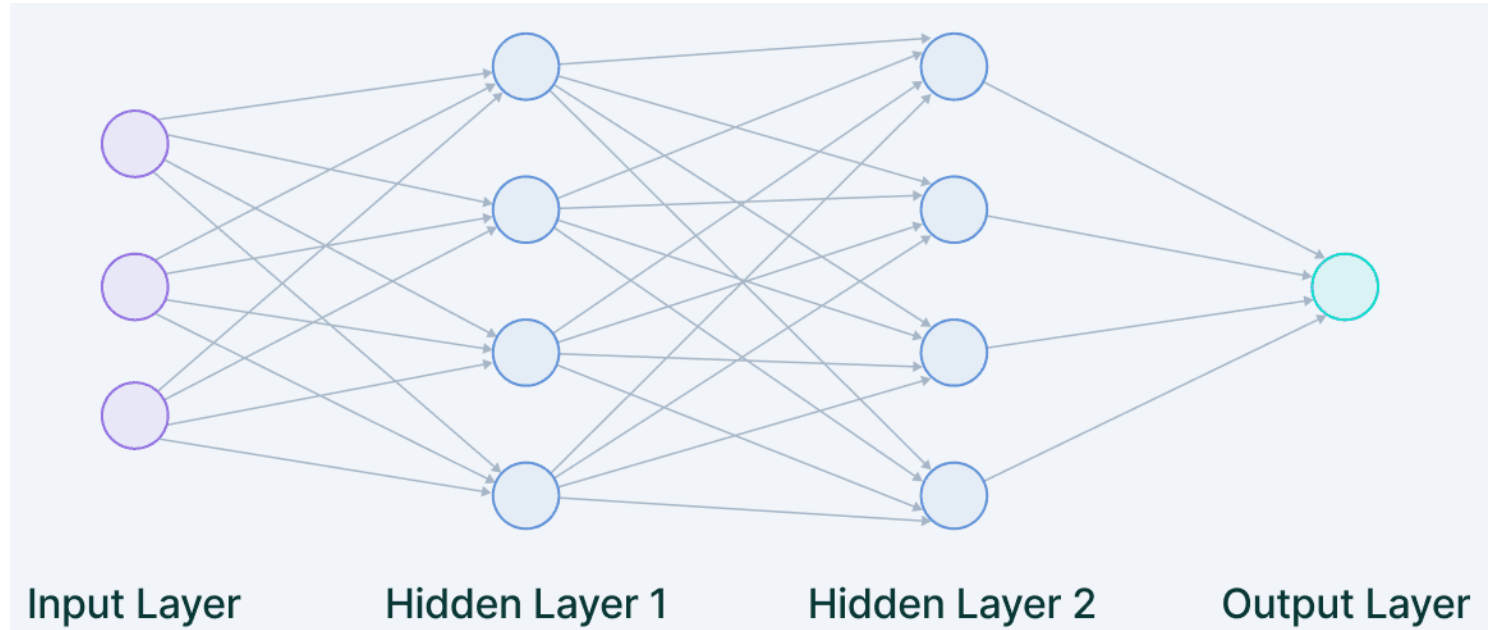
Architecture

A **layer** is a combination of multiple neurons stacked together in a row.

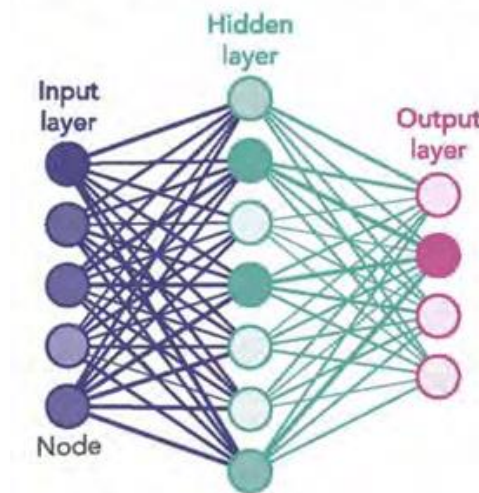
In the **input layer**, each neuron corresponds to a feature in the input dataset.

The **hidden layers** are intermediate layers that do all the computations and extract the features from the data. There can be multiple interconnected hidden layers that account for searching different hidden features in the data.

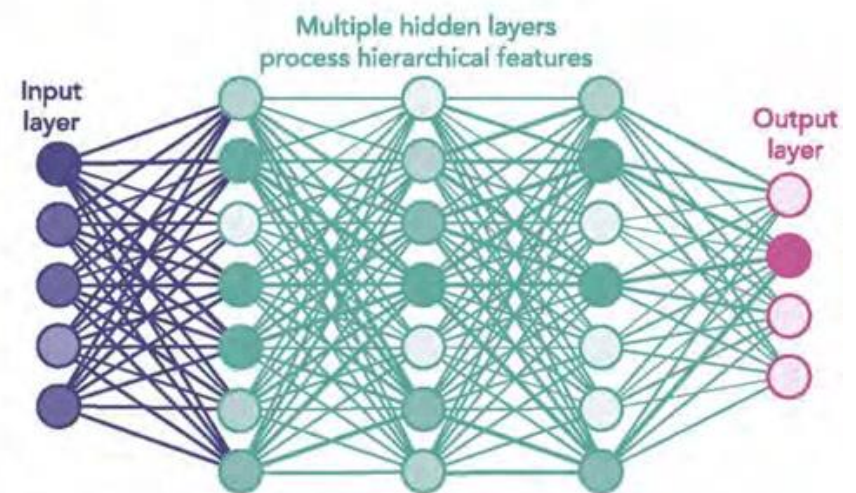
The **output layer** maps the learned features, such as class labels in classification tasks, from the hidden layers to the final output.



SHALLOW NEURAL NETWORK



DEEP NEURAL NETWORK



Algorithms

Supervised (discriminative) learning:

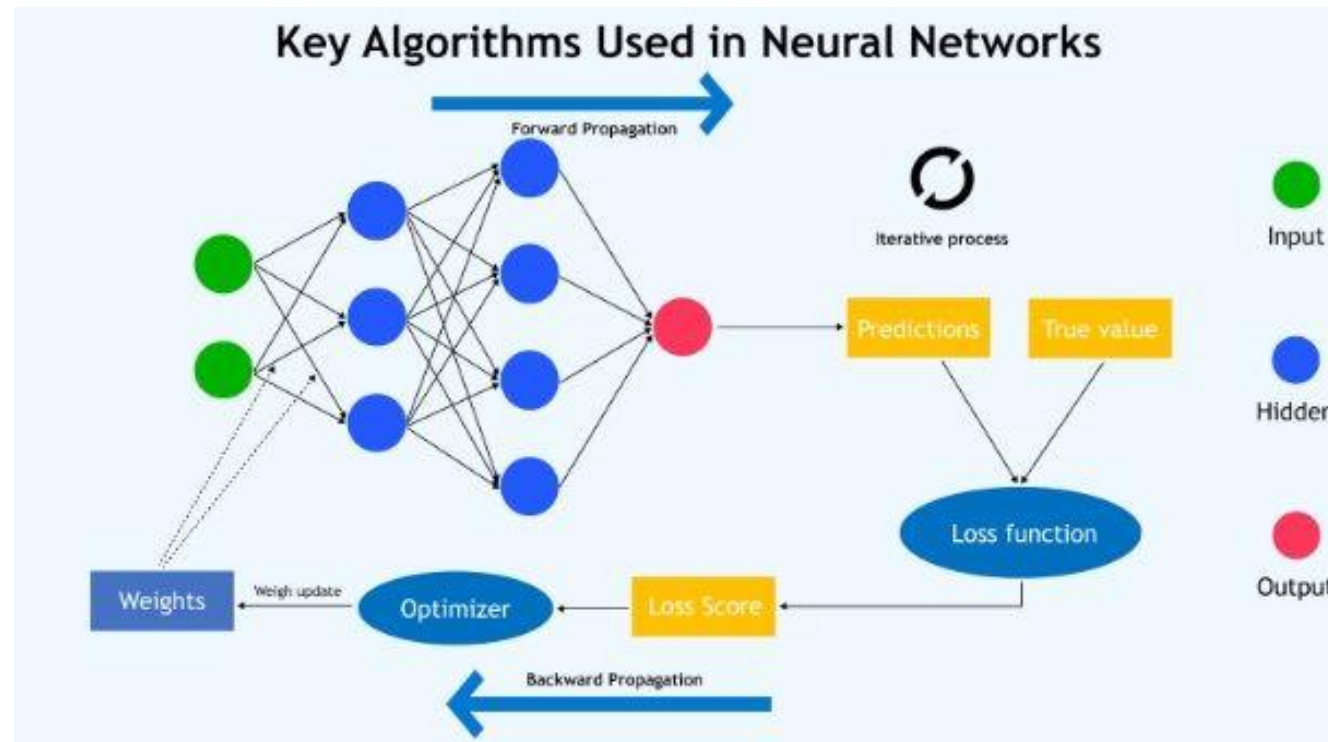
- Backpropagation
- Gradient descent
- Stochastic gradient descent

Unsupervised (generative) learning:

- Autoencoders
- Generative adversarial networks

Reinforcement learning:

- Q-learning
- Policy gradient networks



Classification

Neural networks are highly task-specific, and *no single architecture works for all types of problems*. Choosing the right architecture is critical to achieving high performance, increasing the model's ability to learn from data and make accurate predictions.

Parameter	Types	Description
Connection pattern	FeedForward, Recurrent	Feedforward - In which graphs have no loops. Recurrent - Loops occur because of feedback.
Number of hidden layers	Single-layer, Multi-Layer	Single Layer - Having one hidden layer. E.g., single Perceptron Multilayer - Having multiple hidden layers. E.g., multilayer Perceptron
Nature of weights	Fixed, Adaptive	Fixed - Weights are a fixed priority and not changed at all. Adaptive - Updates the weights and changes during training.
Memory unit	Static, Dynamic	Static - Memoryless unit. The current output depends on the current input. E.g., Feedforward network. Dynamic - Memory unit - The output depends upon the current input as well as the current output. E.g., Recurrent Neural Network

Classification

A mostly complete chart of Neural Networks

©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

- Input Cell
- Backfed Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Gated Memory Cell
- Kernel
- Convolution or Pool

Perceptron (P)



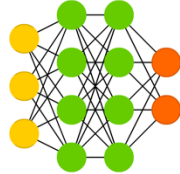
Feed Forward (FF)



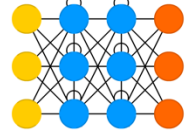
Radial Basis Network (RBF)



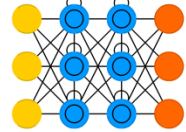
Deep Feed Forward (DFF)



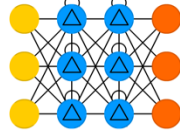
Recurrent Neural Network (RNN)



Long / Short Term Memory (LSTM)



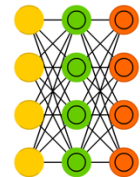
Gated Recurrent Unit (GRU)



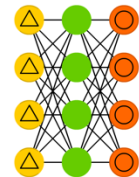
Auto Encoder (AE)



Variational AE (VAE)



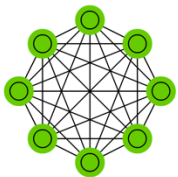
Denoising AE (DAE)



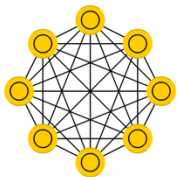
Sparse AE (SAE)



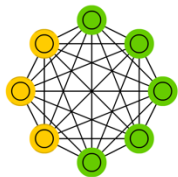
Markov Chain (MC)



Hopfield Network (HN)



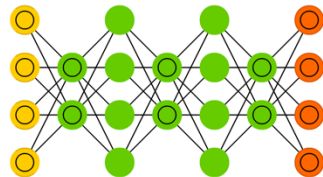
Boltzmann Machine (BM)



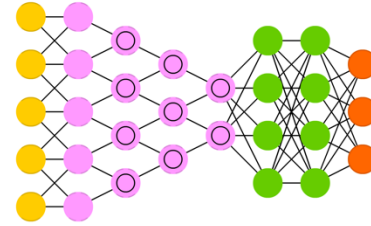
Restricted BM (RBM)



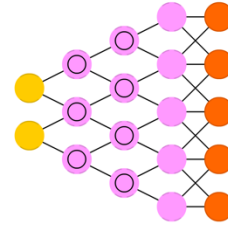
Deep Belief Network (DBN)



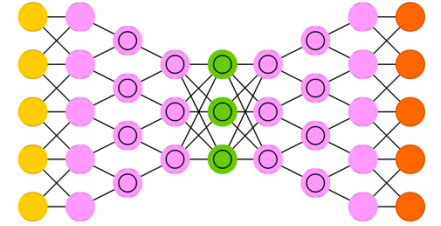
Deep Convolutional Network (DCN)



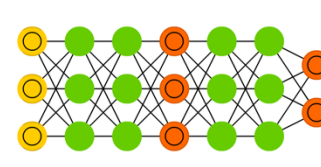
Deconvolutional Network (DN)



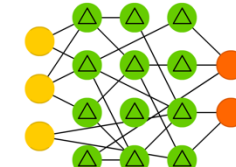
Deep Convolutional Inverse Graphics Network (DCIGN)



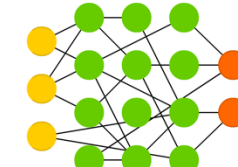
Generative Adversarial Network (GAN)



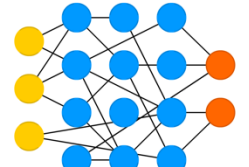
Liquid State Machine (LSM)



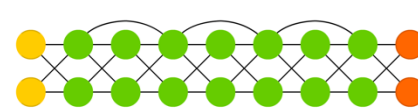
Extreme Learning Machine (ELM)



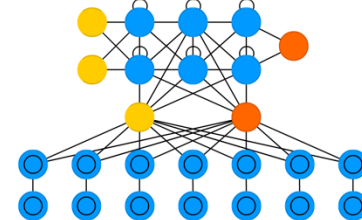
Echo State Network (ESN)



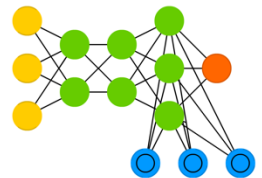
Deep Residual Network (DRN)



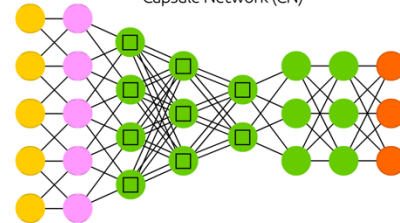
Differentiable Neural Computer (DNC)



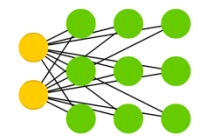
Neural Turing Machine (NTM)



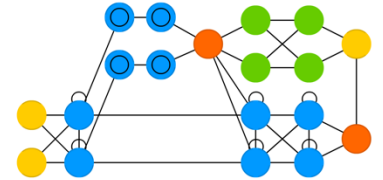
Capsule Network (CN)



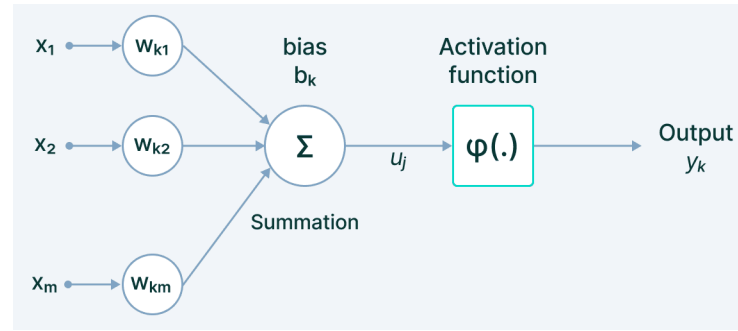
Kohonen Network (KN)



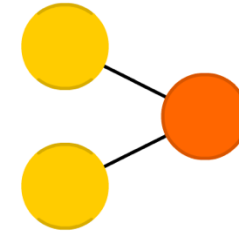
Attention Network (AN)



Perceptron



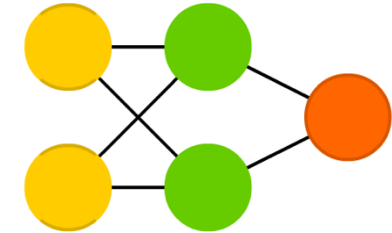
=



Feed-forward neural networks (FNNs)

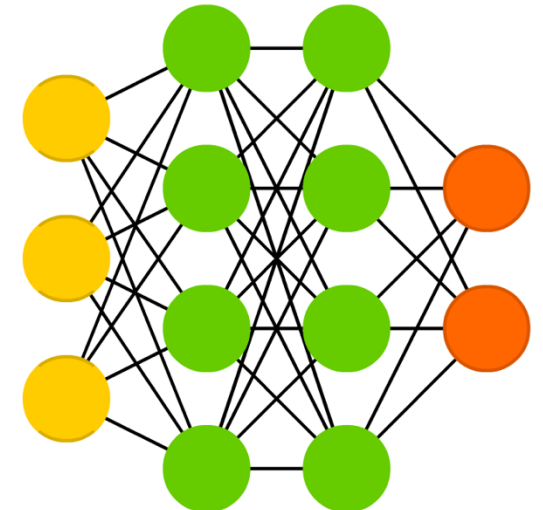
In FNNs, data flows from the input layer to the output layer *without any cycles or loops*. In this architecture, the interconnected neurons are arranged in layers, with each layer fully connected to the next. As the name suggests, the information is passed in the *forward direction* – from left to right.

FNNs are suitable for straightforward classification or regression problems.






Multi-Layer Perceptron (MLP)

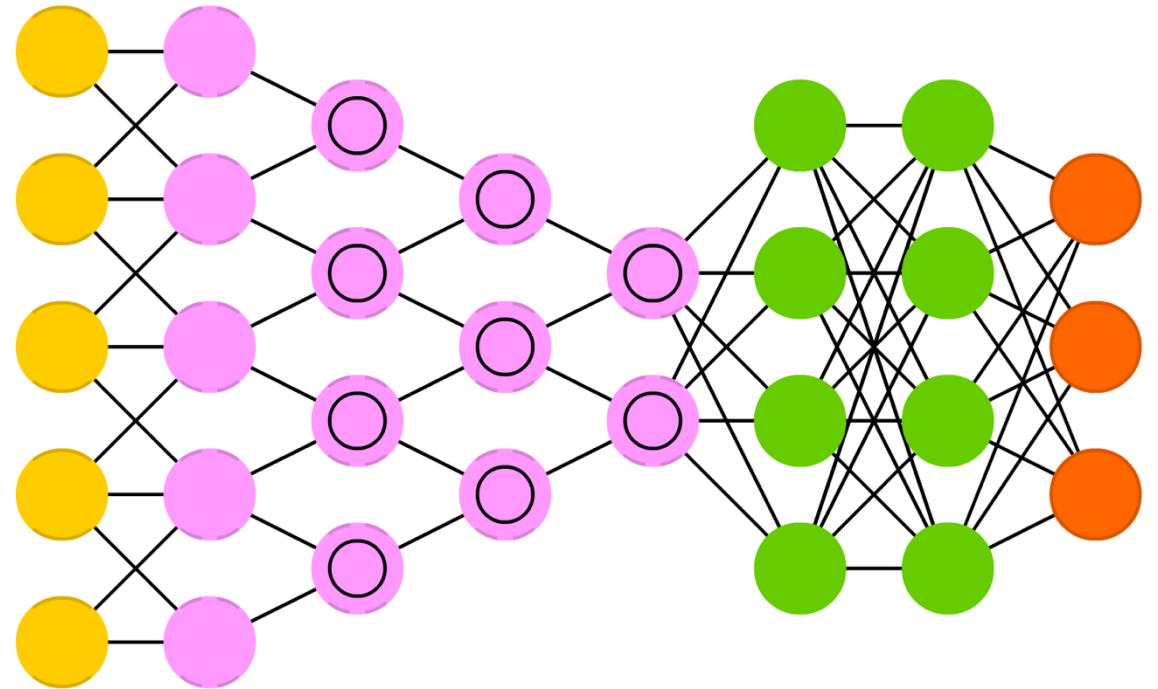
MLP is a modern feedforward neural network consisting of fully connected neurons with nonlinear activation functions, organised in layers, notable for being able to distinguish data that is not linearly separable.



Convolutional neural networks (CNNs)

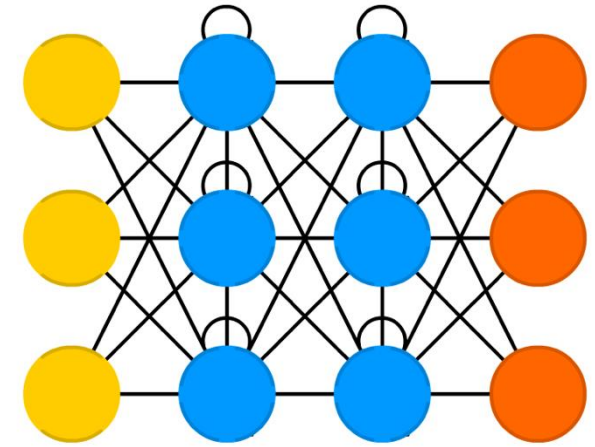
CNNs are designed to process grid-like data, especially images. The layers used in CNNs perform convolutions to automatically extract features from images, such as edges, textures, and shapes. These features are then used to recognise objects, patterns, or classify images.

-  **Convolutional layers** detect local features using convolution operations.
-  **Pooling layers** minimise computational load and avoid overfitting by reducing spatial dimensions.
-  **Fully-connected layers** at the end of the network integrate the features learned in previous layers to make final predictions.



Recurrent neural networks (RNNs)

RNNs can process *sequential data* where the order of the inputs matters. RNNs contain *loops* that allow information to be passed from one step to the next, making them suitable for tasks that involve time-series data or sequences of information.



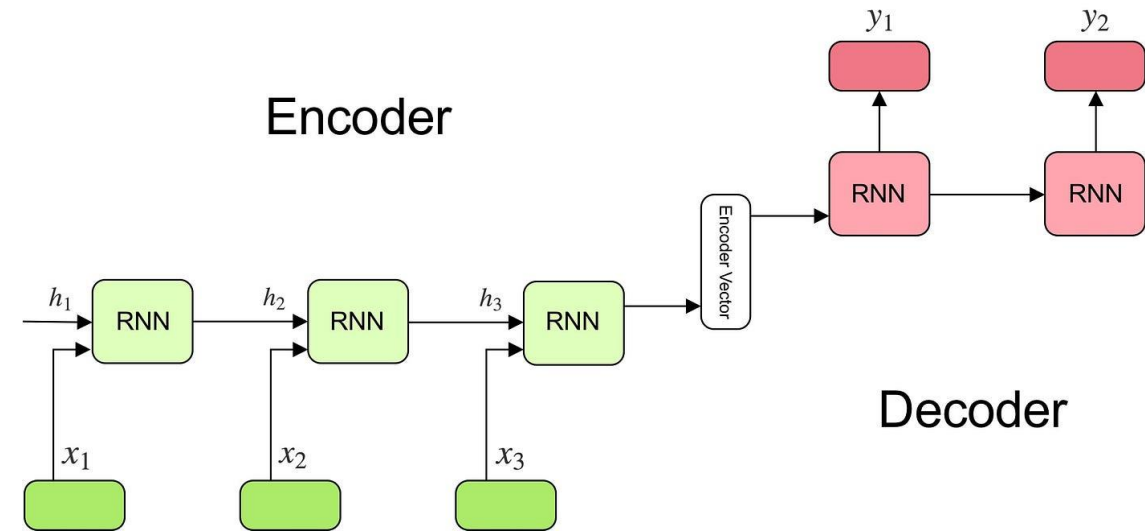
Encoder-decoder

The encoder-decoder model is a way of organising RNNs for *sequence-to-sequence* prediction problems or challenging sequence-based inputs.

Encoder reads the input sequence and summarises the information in a single fixed-length internal state (hidden, context) vectors, which are passed along as inputs for decoder.

Decoder interprets the context vector obtained from the encoder and generates the output sequence.

Autoencoders are encoder-decoder models in which the input and output domains are the same.

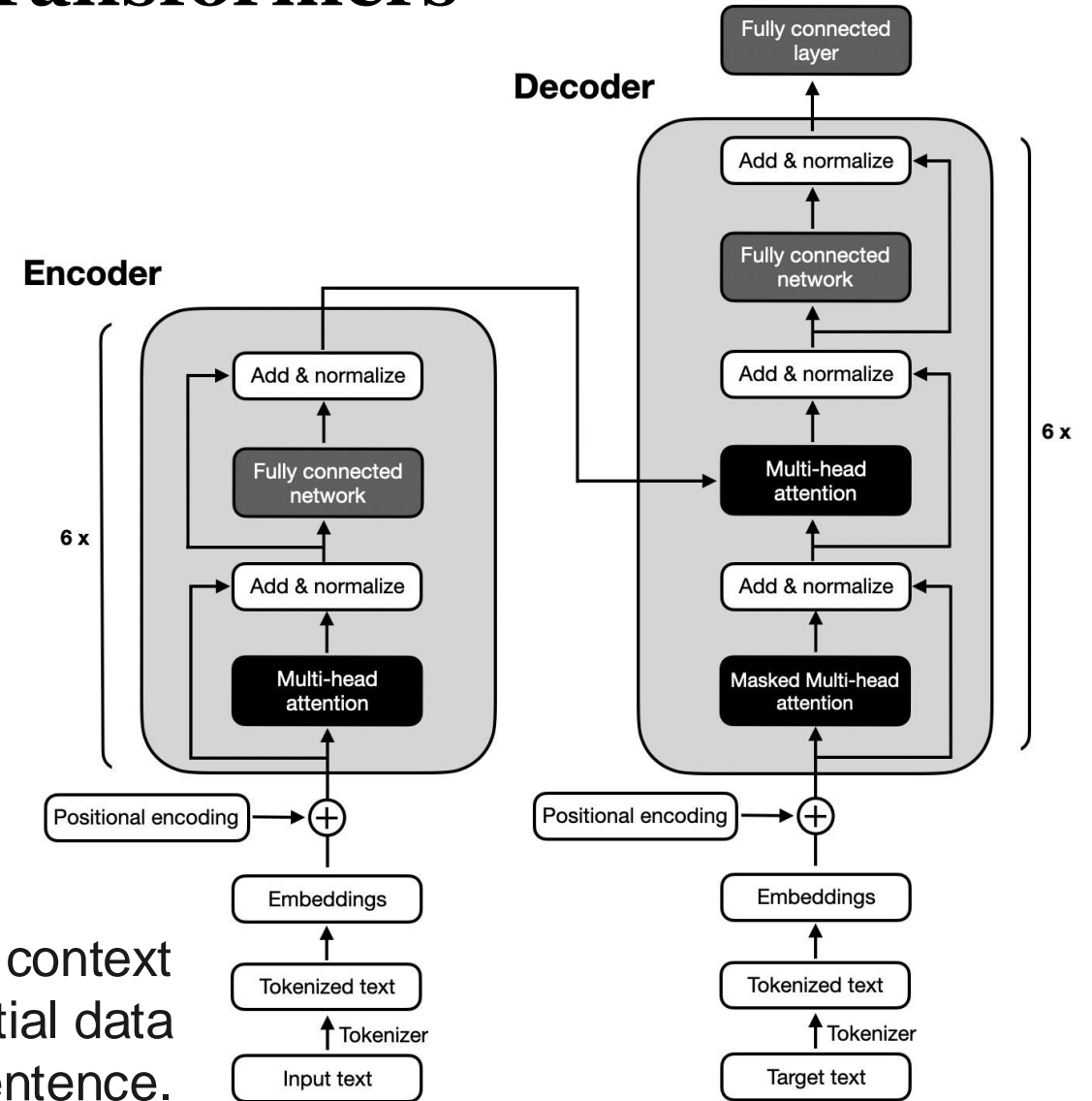


Attention

Introducing an **attention mechanism** into sequence-to-sequence (encoder-decoder) removes the limitation of a fixed-length context vector. Attention model tries to develop a context vector that is selectively filtered specifically for each output time step. This is achieved by keeping the intermediate outputs from the encoder network which correspond to a certain level of significance, from each step of the input sequence and at the same time training the model to learn and give selective attention to these intermediate elements and then relate them to elements in the output sequence.

A **transformer model** is a neural network that learns context and thus meaning by tracking relationships in sequential data like the words in this sentence.

Transformers

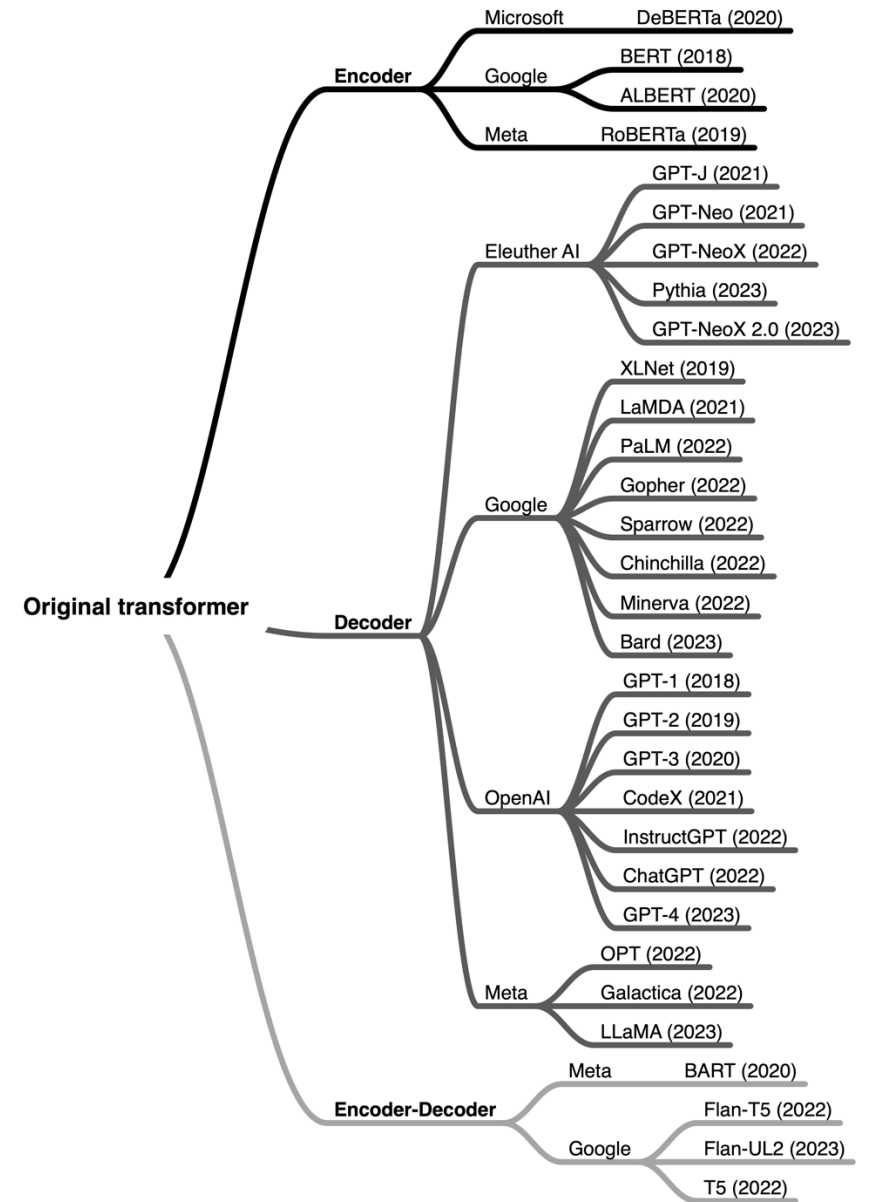


Bigger picture

Large language models (LLMs) are designed for natural language processing tasks, e.g., language generation. Many modern LLMs are based on a transformer architecture and are trained with self-supervised learning on a vast amount of text.

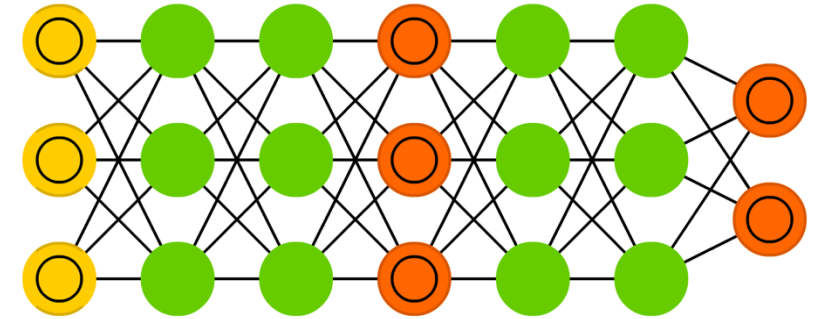
Foundation models are ML models trained on broad/vast data, generally use self-supervision, contain at least tens of billions of parameters, and are generalisable = applicable across a wide range of contexts.

Generative AI (GenAI) is a subset of artificial intelligence able to create content, e.g., text (LLMs), images (Midjourney, StableDiffusion), code (Copilot), audio (VALL-E), or other forms of data. Generative models learn the underlying patterns and structures of their training data and use them to produce new data based on the input.



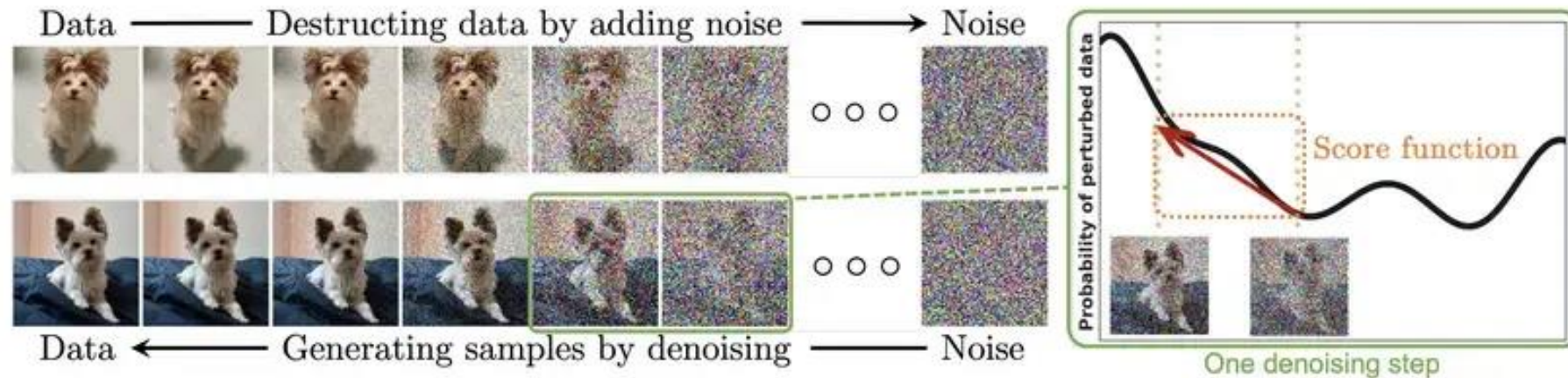
Generative adversarial networks (GANs)

GANs consist of two competing networks, a *generator* and a *discriminator*, which are trained together. The generator tries to create realistic data, and the discriminator attempts to distinguish real data from fake data.



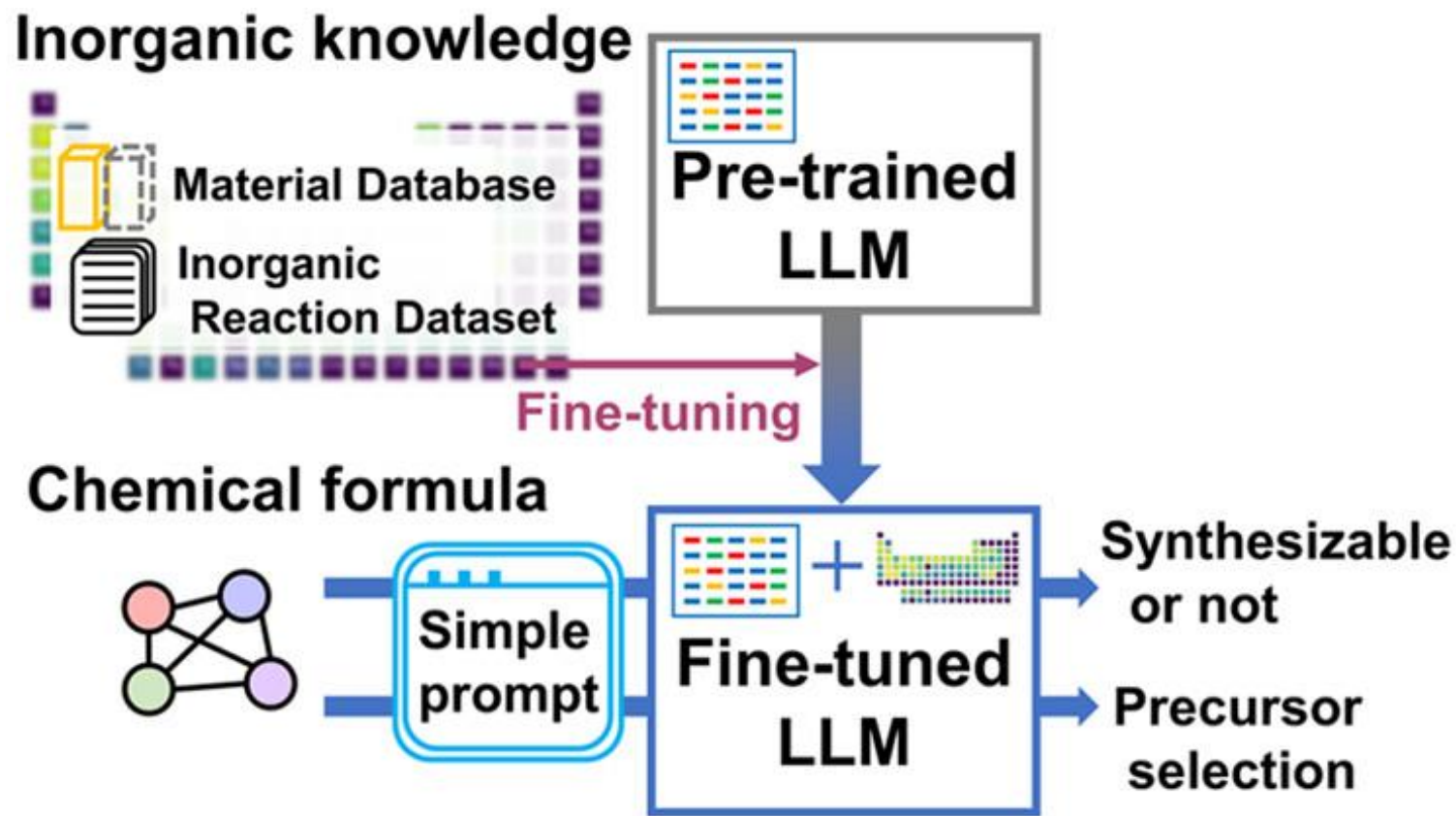
Diffusion models

Diffusion models work by adding *noise* to the available training data and then learning how to reverse the process. The reverse operation may then be applied to new random data to generate new outputs.



Fine-tuning

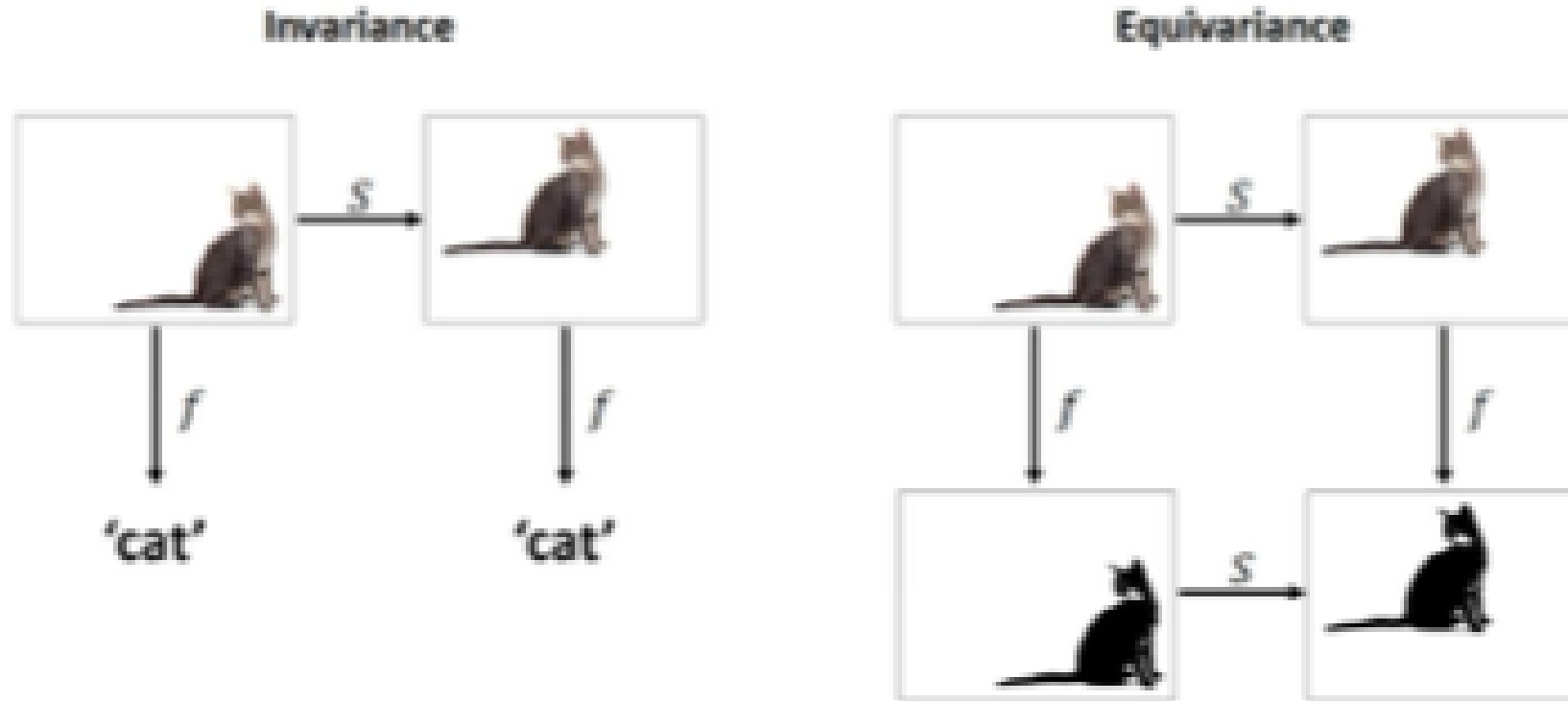
– adapting pre-trained (foundation) models to specific tasks by training them on smaller, targeted datasets.



We evaluate the effectiveness of pretrained and fine-tuned large language models (LLMs) for predicting the synthesizability of inorganic compounds and the selection of precursors needed to perform inorganic synthesis. The predictions of fine-tuned LLMs are comparable to – and sometimes better than – recent bespoke machine learning models for these tasks but require only minimal user expertise, cost, and time to develop.

Geometric deep learning

is based on neural network architectures that incorporate and process symmetry information.

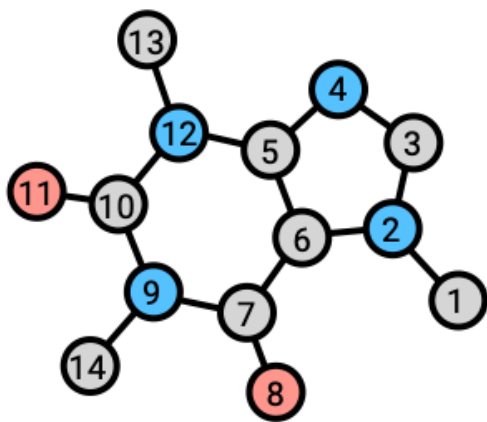
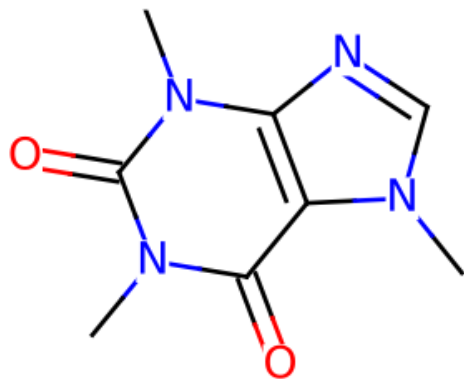


Invariance: the output stays constant no matter the input is symmetry-transformed.

Equivariance: the output undergoes exactly the same symmetry transformations as applied to the input.

Geometric deep learning

is well-adapted to learning on molecules due to their inherent graph nature.



Node features

1. Atomic number
2. # of directly-bonded neighbors
3. # of hydrogens
4. Formal charge
5. Atomic mass
6. Is in a ring?

		1.	2.	3.	4.	5.	6.
1	C	6	4	3	0	12.011	0
2	N	7	3	0	0	14.007	1
3	C	6	3	1	0	12.011	1
4	N	7	2	0	0	14.007	1
5	C	6	3	0	0	12.011	1
6	C	6	3	0	0	12.011	1
7	C	6	3	0	0	12.011	1
8	O	8	1	0	0	15.999	0
9	N	7	3	0	0	14.007	1
10	C	6	3	0	0	12.011	1
11	O	8	1	0	0	15.999	0
12	N	7	3	0	0	14.007	1
13	C	6	4	3	0	12.011	0
14	C	6	4	3	0	12.011	0

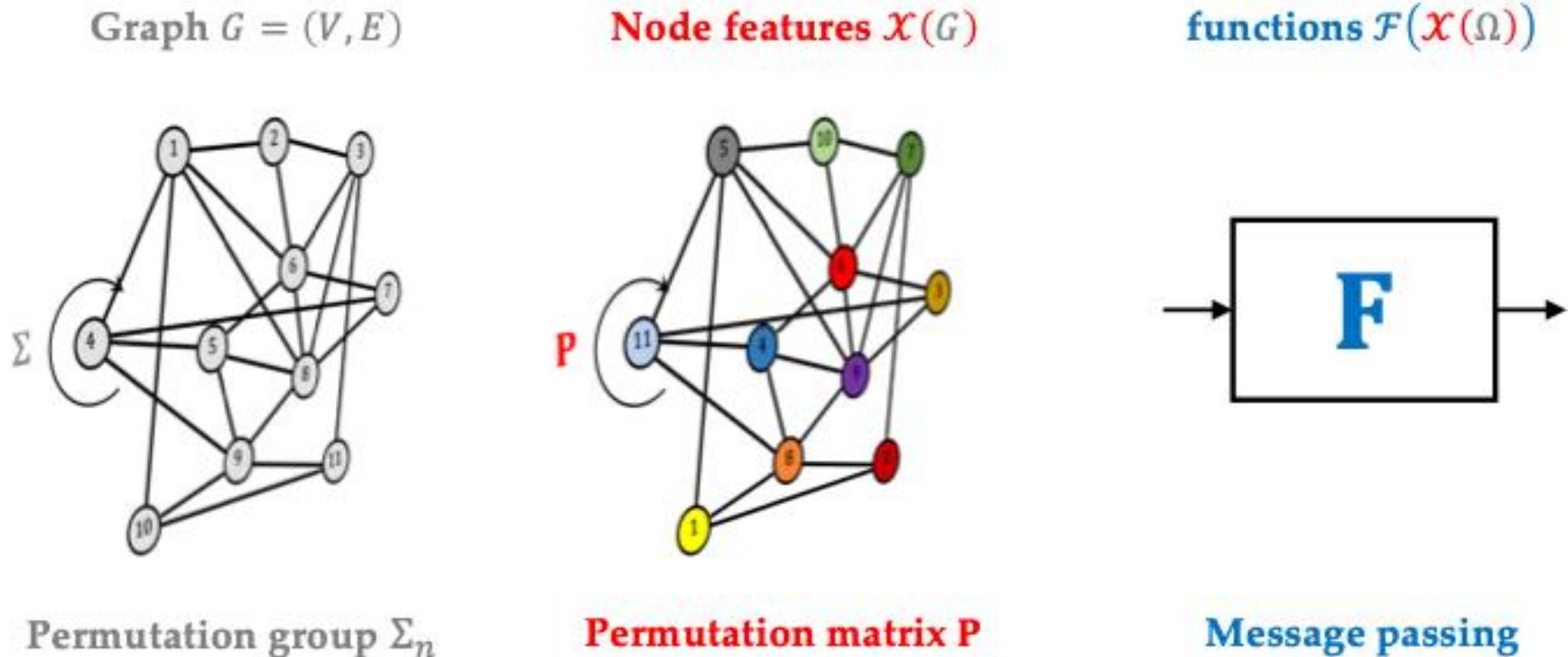
Edge features

1. Bond type
2. Stereochemistry

		1.	2.
1	2	1	0
2	3	12	0
3	4	12	0
4	5	12	0
5	6	12	0
6	7	12	0
7	8	2	0
7	9	12	0
9	10	12	0
10	11	2	0
10	12	12	0
12	13	1	0
9	14	1	0
6	2	12	0
12	5	12	0

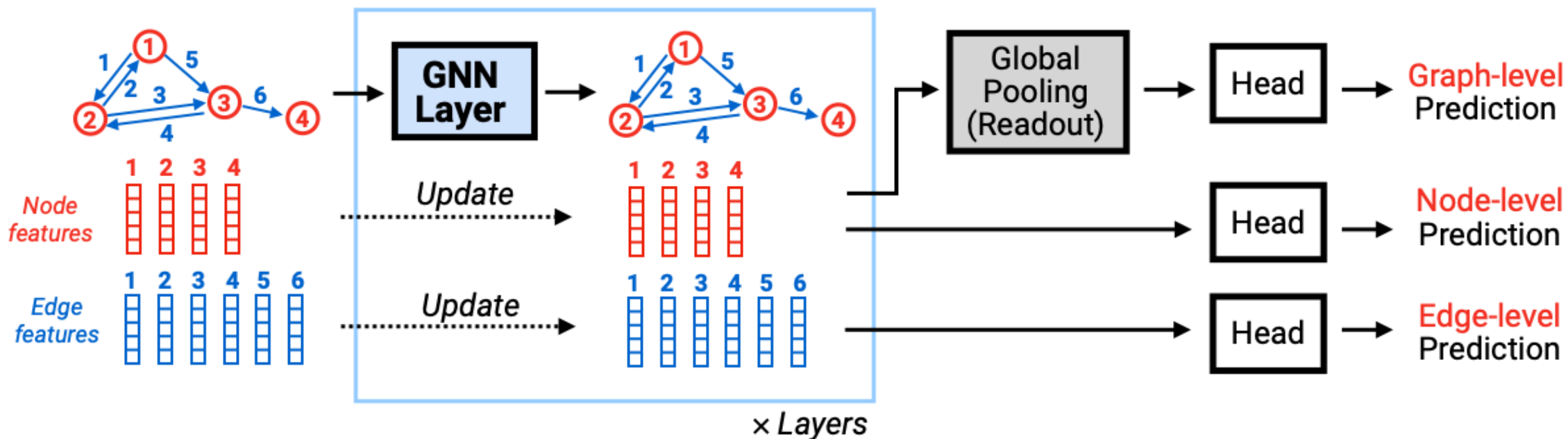
Geometric deep learning

Message passing – sharing information between nodes in a graph along the edges that connect them. Through message passing, graph nodes iteratively update their representations by exchanging information with their neighbors. A node can only send a message to its neighbors when it has finished getting all the messages from its other neighbors.



Geometric deep learning

Pooling is a process that aggregates node representations to generate a single graph-level representation, allowing GNNs to learn and predict on the graph as a whole.



Machine learning vs. Deep learning

	(Classical) ML	DL
Datapoints	Hundreds	Millions
Training time	Short (seconds to hours)	Long (weeks)
Architecture	Algorithms (e.g., regression)	Artificial neurons
Data	Structured	Unstructured
Human intervention	Significant (e.g., featurisation, continuous learning)	Minimal
Interpretability	High	Low
Hardware	CPU	GPU
Accuracy	Limited	Unlimited