Recent Machine Learning Techniques and Exploration of New Physics

May 12, 2022

Yuta Nakashima

大阪大学データビリティフロンティア機構
Osaka University Institute for Datability Science

About me

- **Yuta Nakashima**
  Institute for Datability Science, Osaka University
  Associate Professor

- Bio
  - -2012: Ph.D Course, Osaka University
  - 2012: Visiting Scholar, UNC Charlotte
  - 2012-2016: Assistant Prof., Nara Institute for Science and Technology
  - 2015-2016: Visiting Scholar, CMU
  - 2017- : Current Position

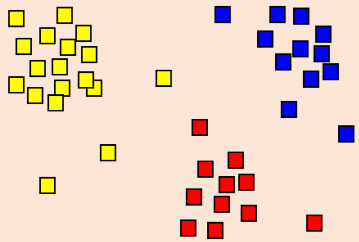
- Research interests
  - Computer Vision; CV
  - Pattern Recognition; PR
  - (Natural Language Processing; NLP)

Tasks in Machine Learning

Machine Learning

Unsupervised learning

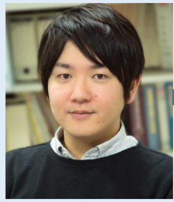
- Dimensionality reduction
- Clustering



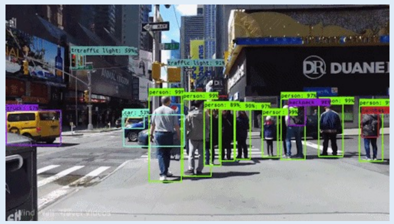
Supervised learning

- Classification
- Regression
- Detection
- Segmentation

From: https://thegradient.pub/semantic-segmentation/



$35.4 \in \mathbb{R}_+$



From: https://towardsdatascience.com/google-ais-new-object-detection-competition-6dde25cf099d

Reinforcement learning

…

Relationships among AI-related fields

Artificial Intelligence, AI

- Rule bases
- Expert systems

Machine Learning, ML

- Support vector machine
- Logistic regression
- Ridge regression
- (shallow) neural networks

Representation Learning, RL

- (shallow) autoencoder
- Dictionary learning

Deep Learning, DL

- Restricted Boltzmann machine
- Deep neural networks

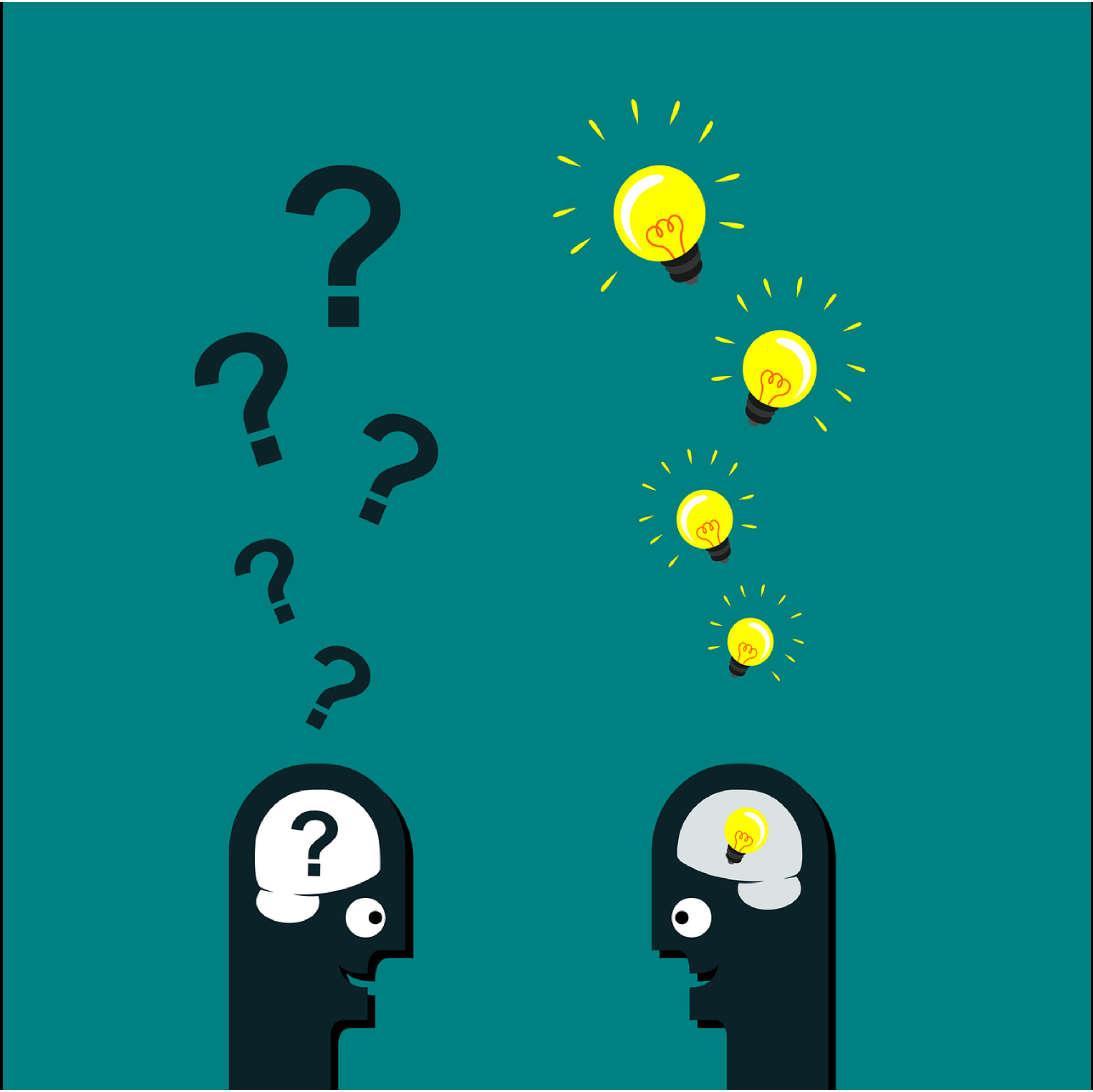
Difference between classic ML and deep learning

- Classic machine learning

x → **Feature extraction** $h(x)$ → v → **Classification** $g(v)$ → Class label "cat" y

- Deep learning

x → **Mapping** $f(x)$ → Class label "cat" y

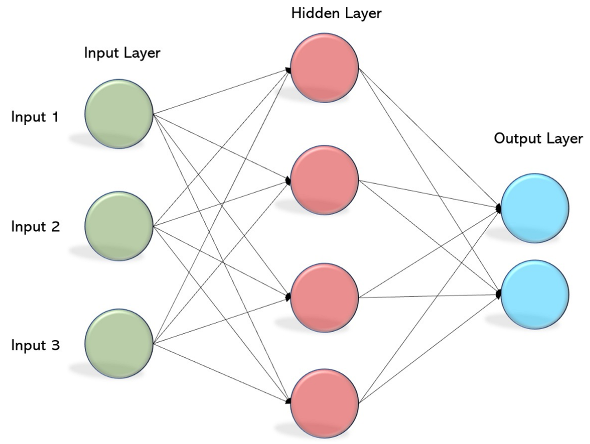


You know what you want as output

- Flavor tagging: a classification problem
- Filtering: an anomaly detection problem
- Particle tracking: a clustering problem
- Vertex finding: a classification/clustering problem
- Energy correction: a regression problem

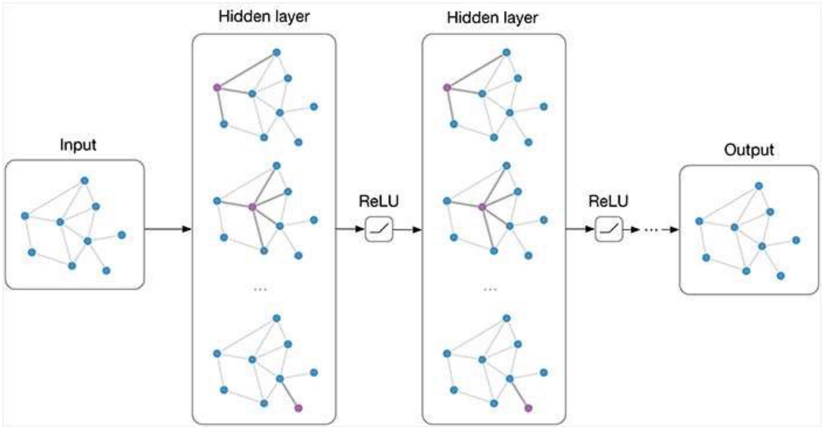
But then with which network architecture?

Multi-layer perceptron (MLP)



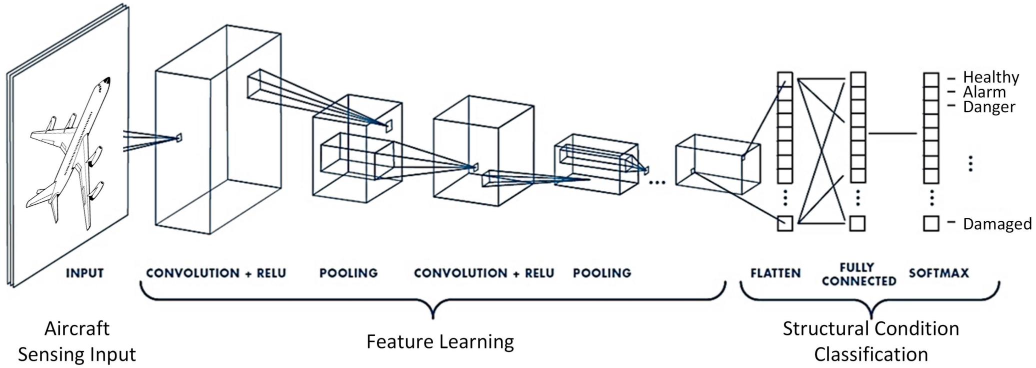
From: https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f

Graph neural networks (GNNs)



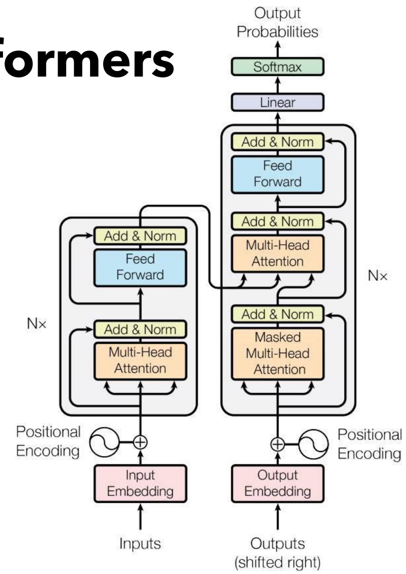
From: https://theaisummer.com/Graph_Neural_Networks/

Convolutional neural networks (CNNs)



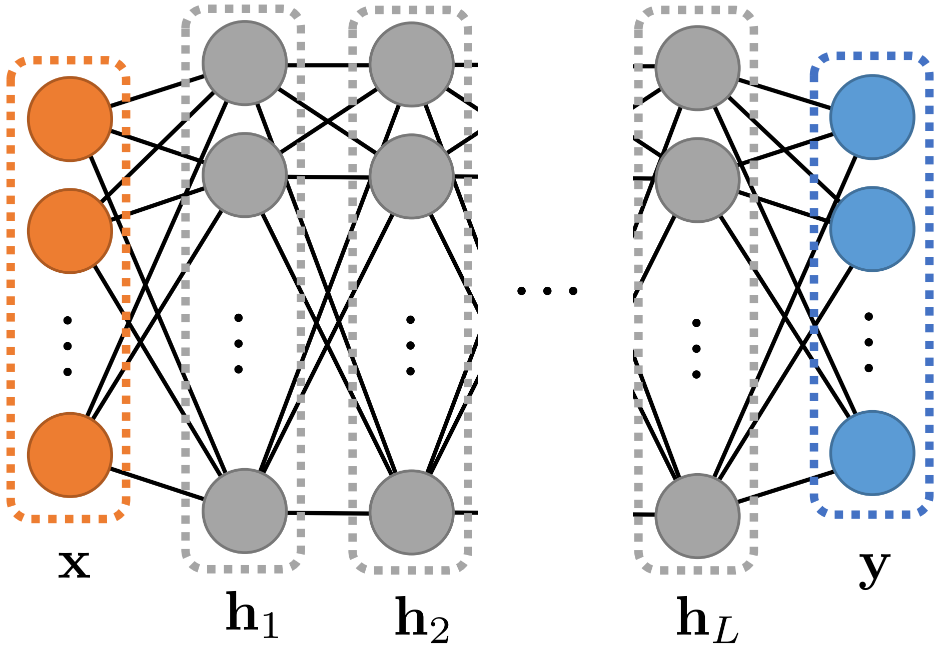
From: [Tabian et al., "A Convolutional Neural Network for Impact Detection and Characterization of Complex Composite Structures," Sensors 19(22), 2019]

Transformers



From: https://pytorch.org/tutorials/beginner/transformer_tutorial.html

Multilayer Perceptron



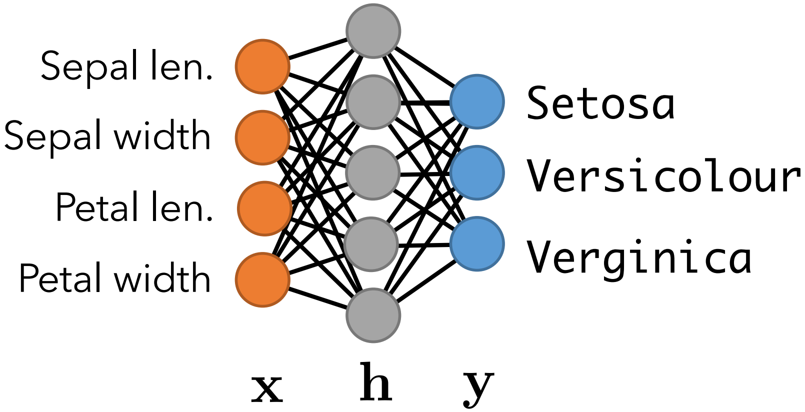
Basic relationship between $\mathbf{h}_l$ and $\mathbf{h}_{l+1}$:

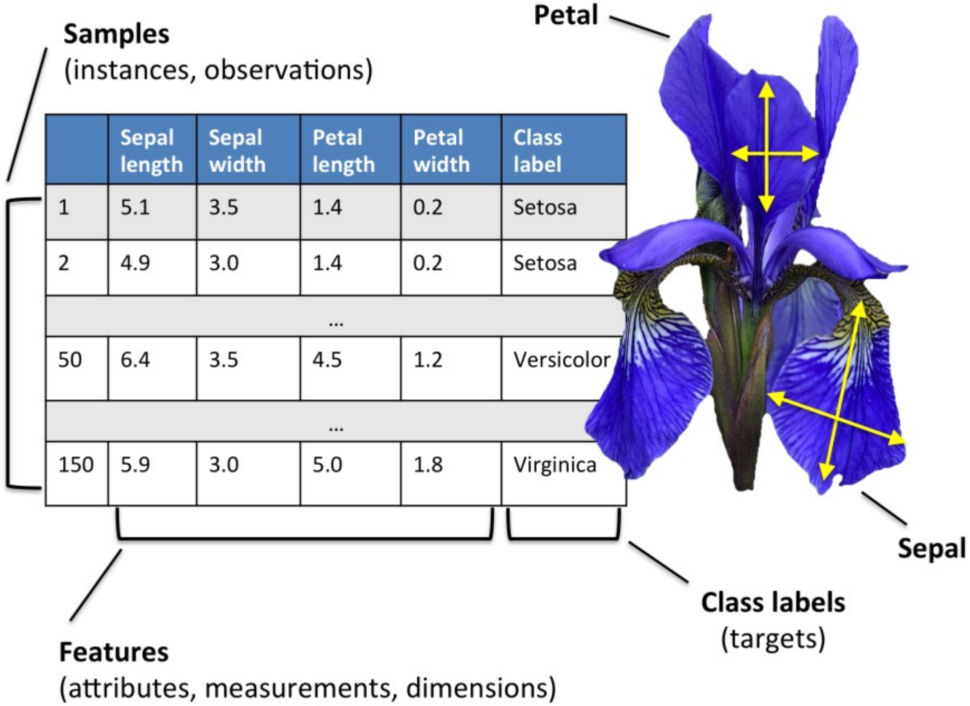
$$\mathbf{h}'_{l+1} = \mathbf{W}_l \mathbf{h}_l + \mathbf{b}_l$$

$$\mathbf{h}_{l+1} = \sigma(\mathbf{h}'_{l+1})$$

MLP examples

- Classification of iris species
  - 4D data
  - 3 class classification

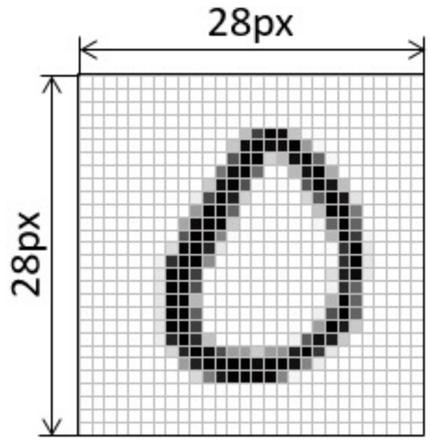




	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

From https://bishwamittra.github.io/imli.html

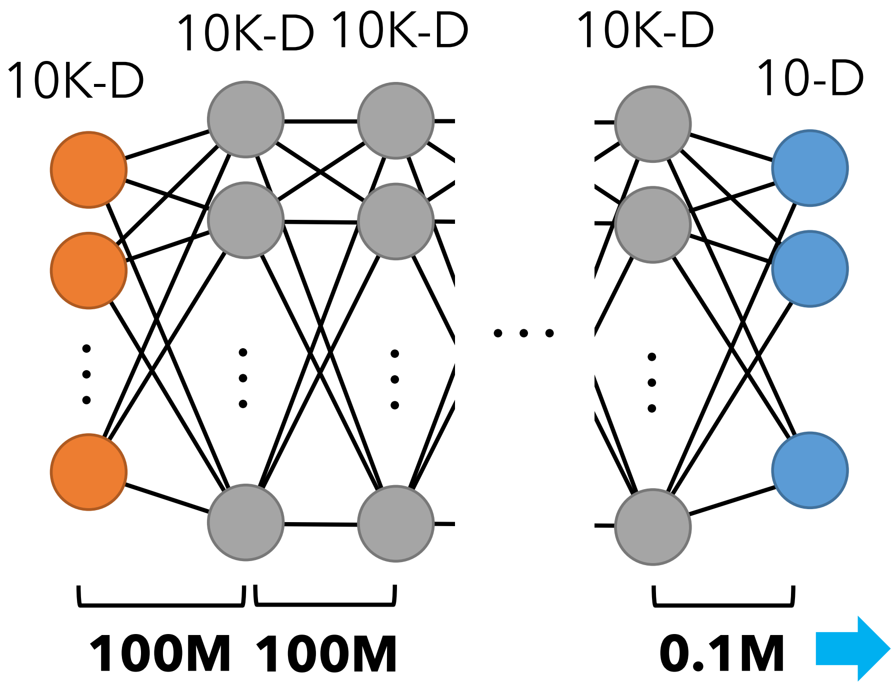
- Handwritten digit recognition (MNIST)
  - 28x28=784D data
  - 10 class classification



From https://liclog.net/excel-vba-deeplearning-mnist-download/

Is it scale?

- MLPs model dense dependency between layers
  - A single element in an output layer depends on all elements in the input layer
  - The most flexible and capable
- The number of learnable parameters is enormous



$100\mathrm{M} \times L + 0.1\mathrm{M} + \alpha$

Structures and dependency of input matters

- Data often has a structure
  - An image: 3D (red, green blue) vectors aligned on 2D grid points
  - A sentence: a sequence of symbols (words)
  - A molecule: a graph of symbols (atoms)

- The structure (may) defines possible dependency among the smaller units

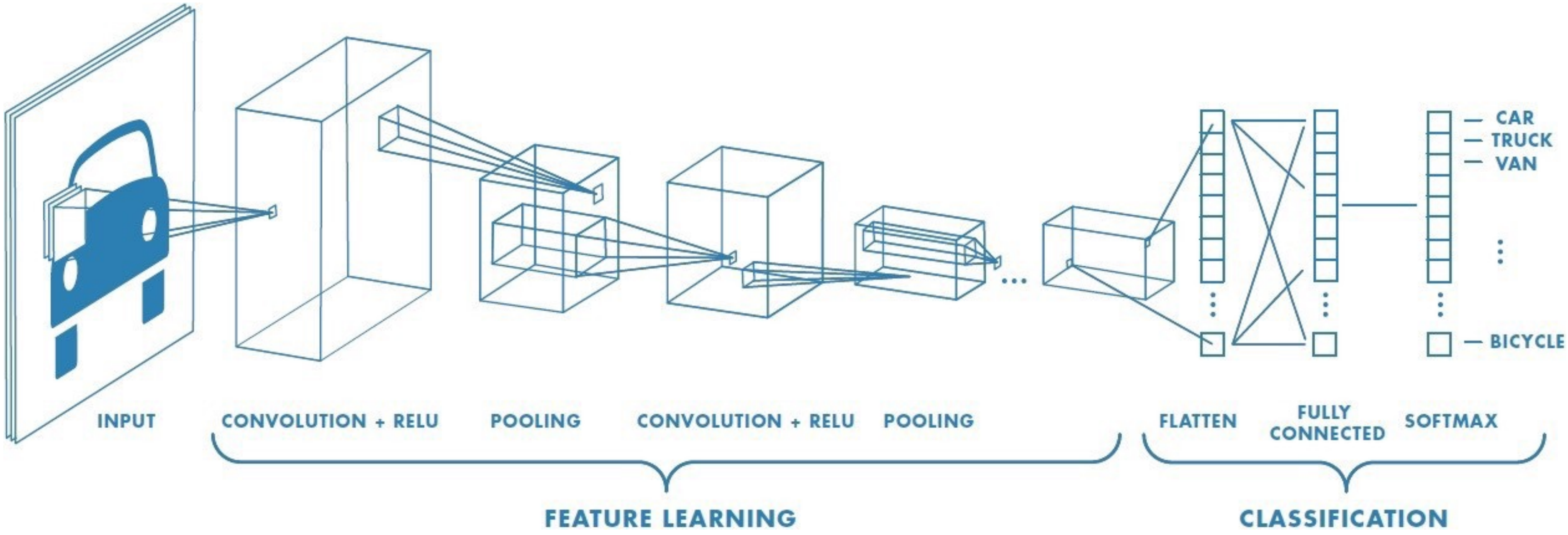
Make the dependency sparse using the structure

Convolutional Neural Networks

- CNNs are a special case of MLPs
  - Assumption:
    - Locality of dependency
    - Spatial invariance

➡ **Use of convolution kernels**

From https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53



A CNN

An MLP

Reduction of parameters over MLPs

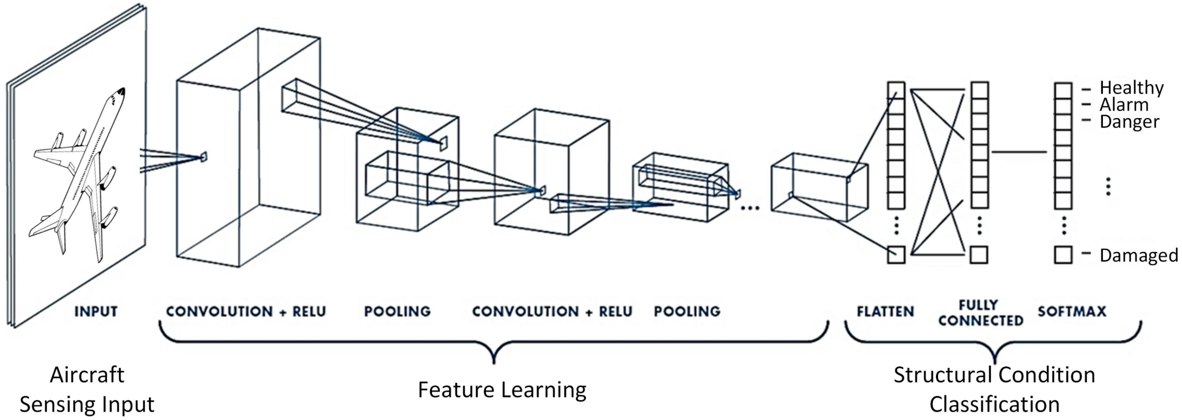
- For example: An image of 100 px by 100 px in RGB
  - Input data: 100x100x3 = 30K-D
  - The first hidden layer: 50x50x20 = 50K-D

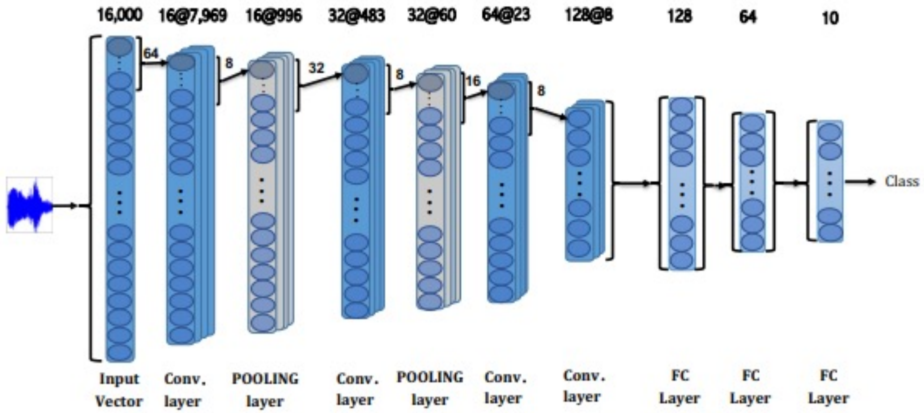
- Parameters to be learned for input to first hidden layer:
  - MLP: 30K x 50K + α = 1500M + α
  - 5 x 5 Convolution kernel: 5 x 5 x 3 x 20 + α = 1500 + α

CNN examples

Image recognition by CNN

From: https://theaisummer.com/Graph_Neural_Networks/



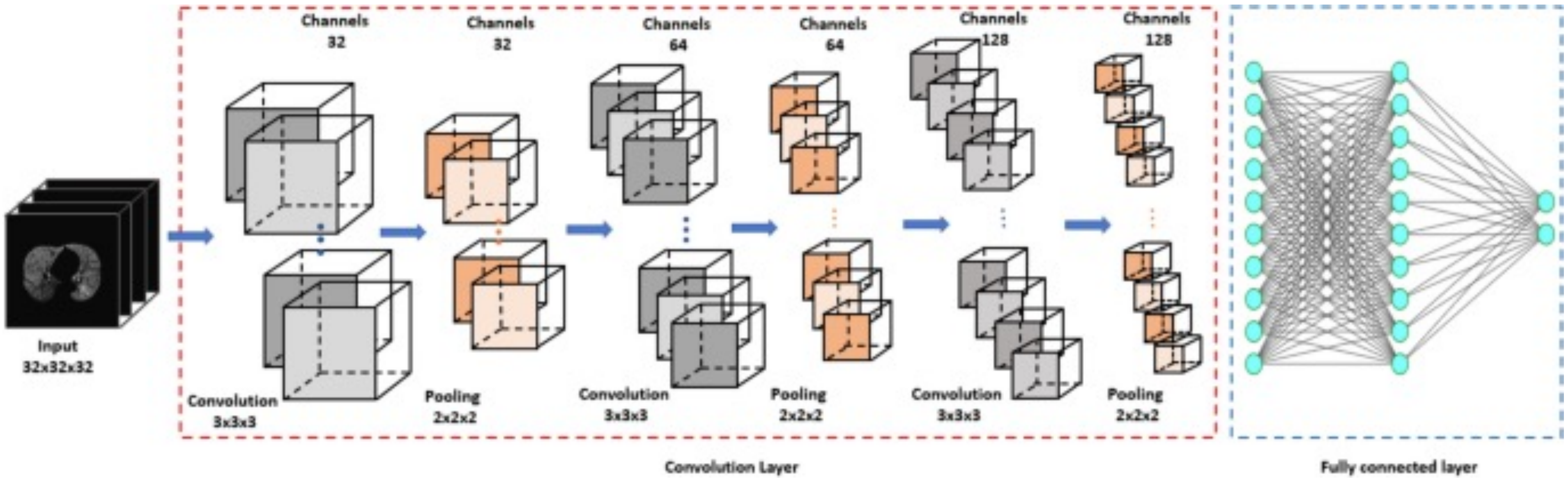


Sound classification by 1D CNN

From: https://medium.com/ai%C2%B3-theory-practice-business/enhanced-environmental-sound-classification-with-a-cnn-1ca388748bc9

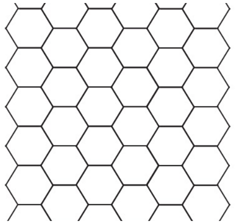
CT image classification by 3D CNN

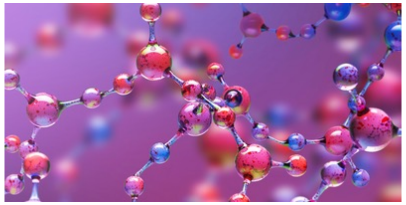
From https://www.nature.com/articles/s41598-020-79336-5



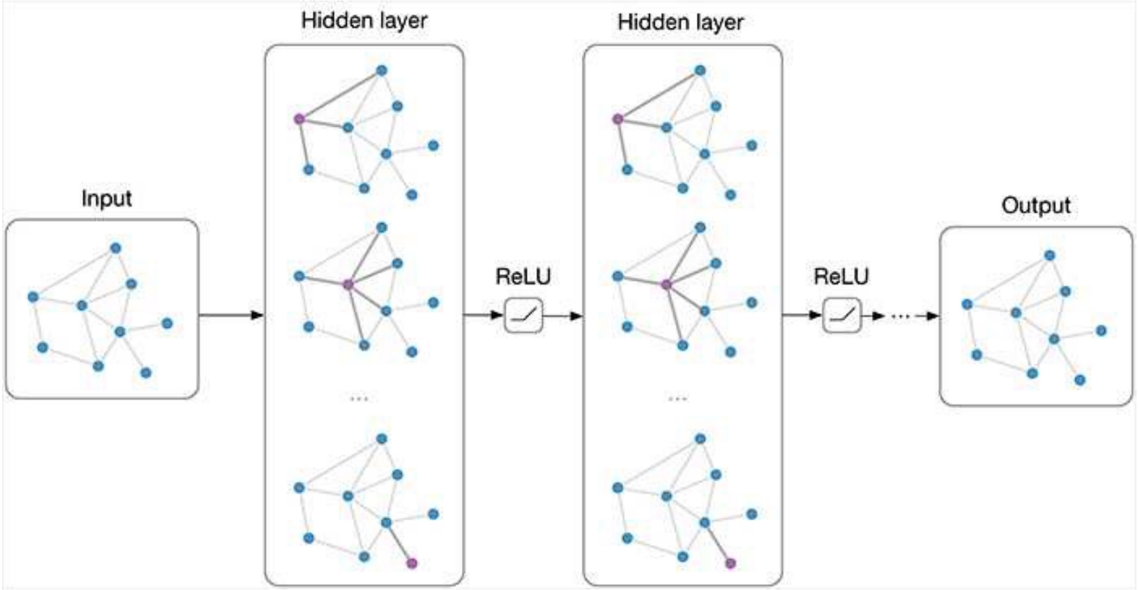
Graph neural networks

- CNNs assumes regular grid points but some types of data may have different (arbitrary) structures
  - Regular structure but hexagonal adjacency
  - Arbitrary connections as in molecules





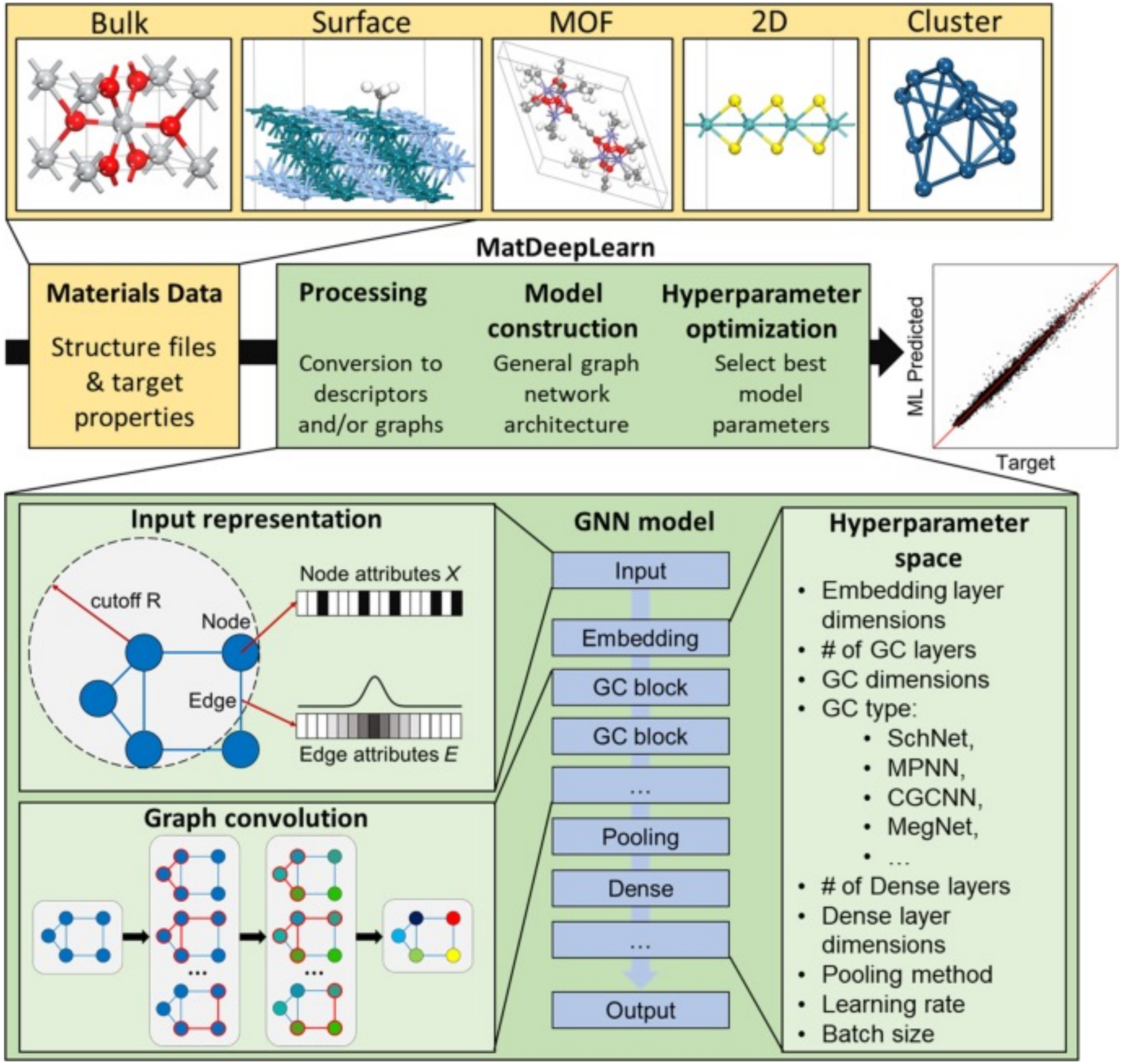
➡ Generalize convolutions with graphs



From: https://theaisummer.com/Graph_Neural_Networks/

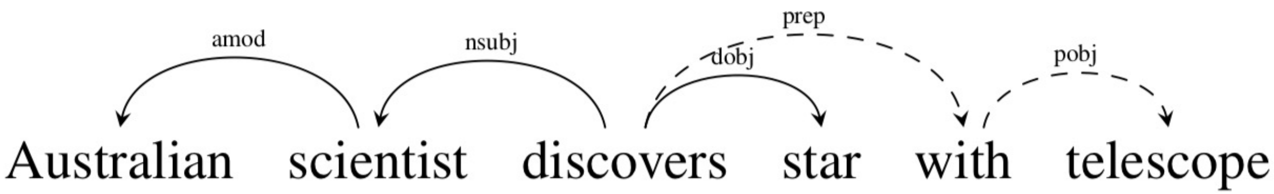
GNN examples

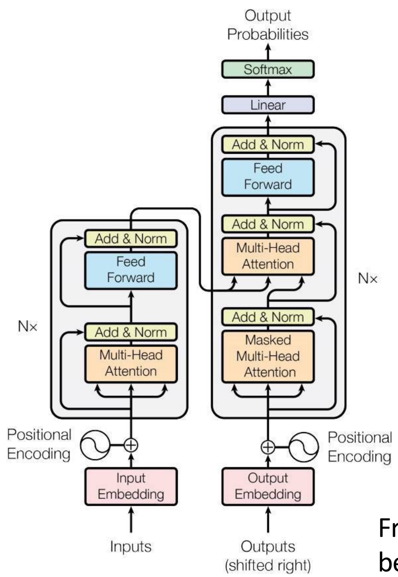
- Material prediction [Fung et al., “Benchmarking graph neural networks for materials chemistry, npj Computational Materials 7, no. 84, 2021]



Transformers [Vaswani et al., “Attention is all you need,” NeurIPS 2017]

- GNNs often assume static and predefined connections, but connections may not be always obvious
  - We often don’t know what depends on what
  - Eg., dependency in a sentence

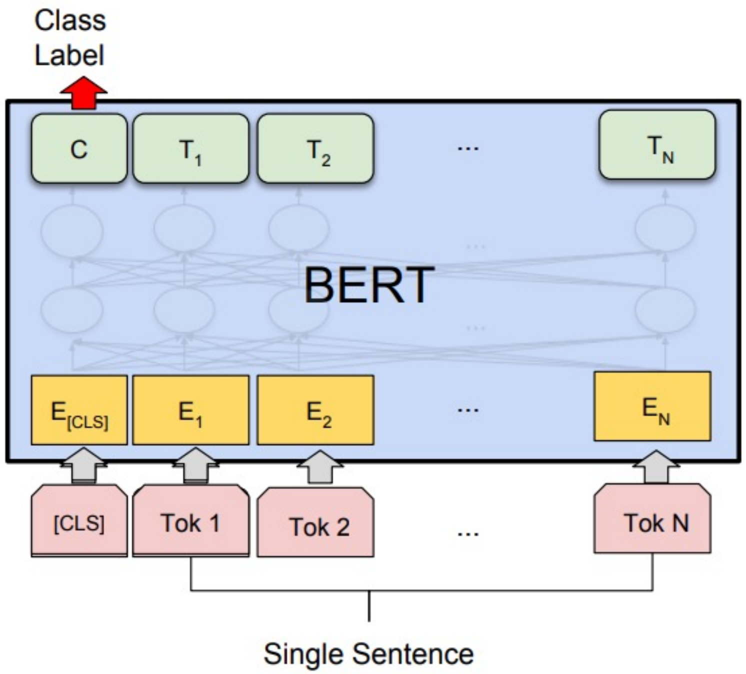




From: https://pytorch.org/tutorials/beginner/transformer_tutorial.html

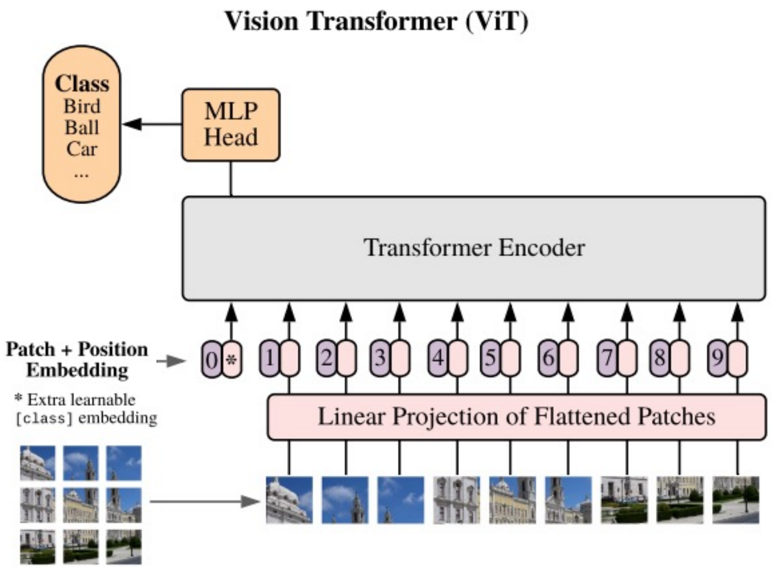
Transformer examples

- Sentiment analysis from text
  - Positive or negative
  - Analyzing reviews



From: https://www.geeksforgeeks.org/sentiment-classification-using-bert/

- Image classification
  - The locality assumption in CNNs may be too strong
  - Transformers allows more flexible dependencies



From: https://paperswithcode.com/method/vision-transformer

So which one should be use?

- If your data is small and you don't know its within-data dependency, an MLP is the first option
  - MLPs is a good choice if # training data is enough and you have abundant computational resource
  - E.g., individual particle (rep. by momentum and energy)

- If you know your data has a regular grid structure (like images), CNNs should be used
  - E.g., particle trajectories represented with an image

- If your data has explicit connections (dependency), GNNs can be used
  - E.g., particles before and after decay

- If you have a large amount of data but within-data dependency is not obvious, Transformers help

Take-away

- Review different architectures of neural networks
    - Dependency matters
    - Network structures determines flexibility of the model, but highly flexible models requires more data and computational resources

- Hope this helps someone who wants to start using neural networks