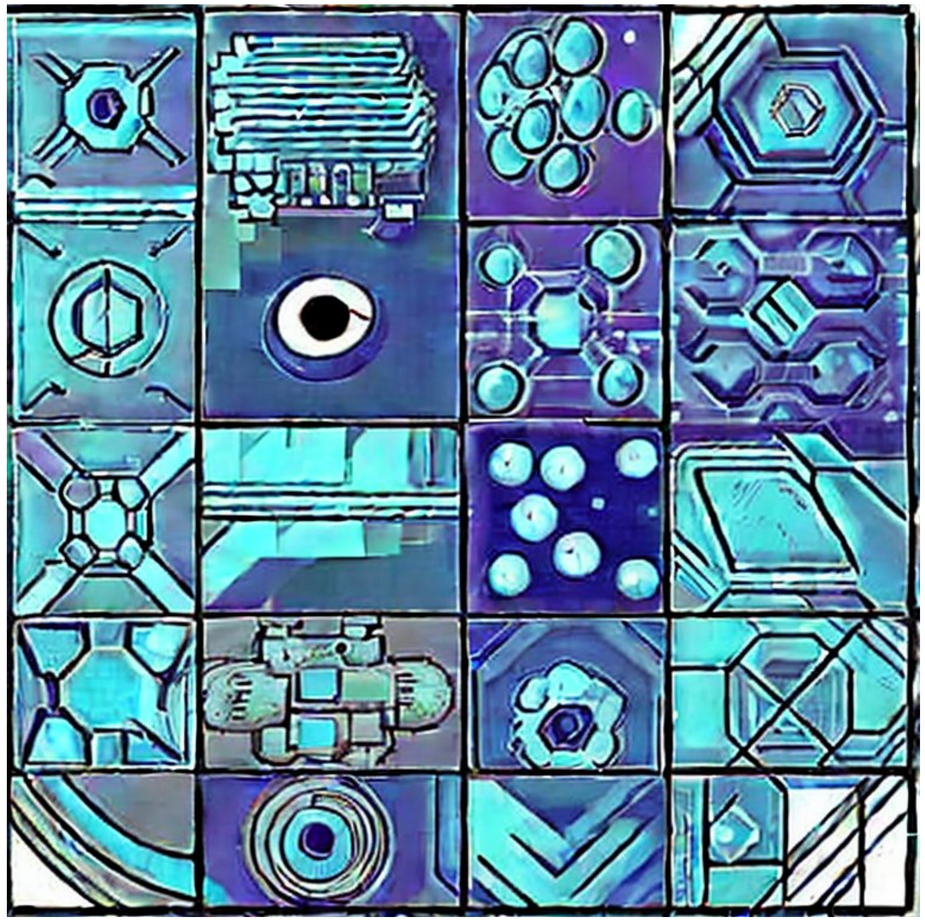


Machine learning for physicists

https://github.com/wangleiphy/ml4p

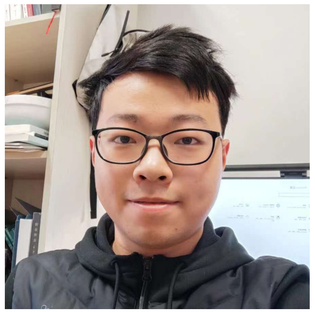
每周四上午10点

2.23	Overview
3.2	Machine learning practices
3.9	A hitchhiker’s guide to deep learning
3.16	Research projects hands-on
3.23	Symmetries in machine learning
3.30	Differentiable programming
4.6	Generative models-I
4.13	Generative models-II
4.20	Research projects presentation
4.27	AI for science: why now ?

课程微信群

助教

李子航

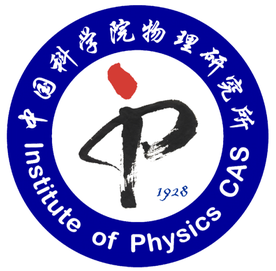


李扬帆

考核方式： project + presentation （1学分）

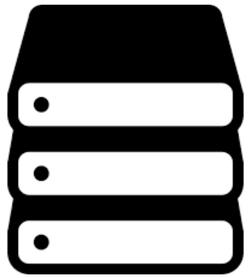
A hitchhiker's guide to deep learning

Lei Wang (王磊)
Institute of Physics, CAS
https://wangleiphy.github.io

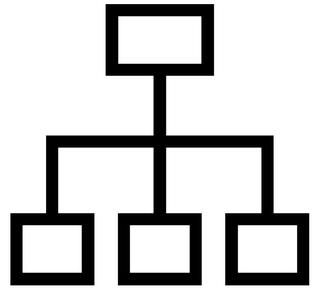


Four pillars

Data



Model



Cost function

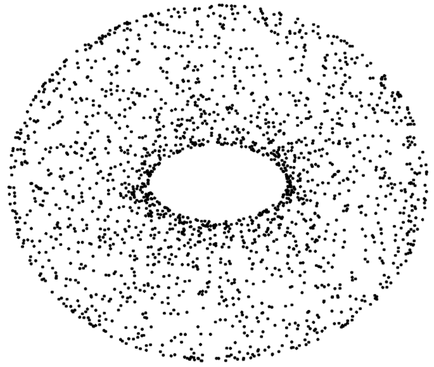
Optimization

Data modalities

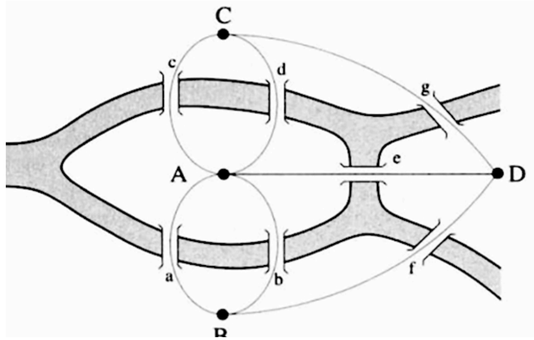
Image

Language

Point cloud

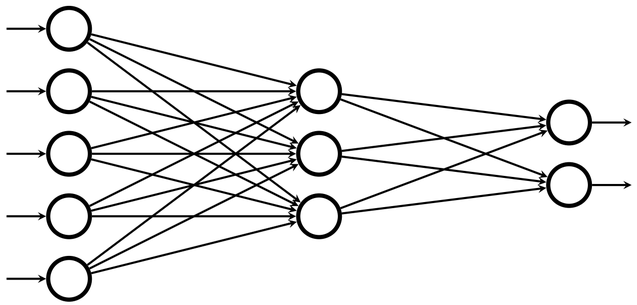


Graph

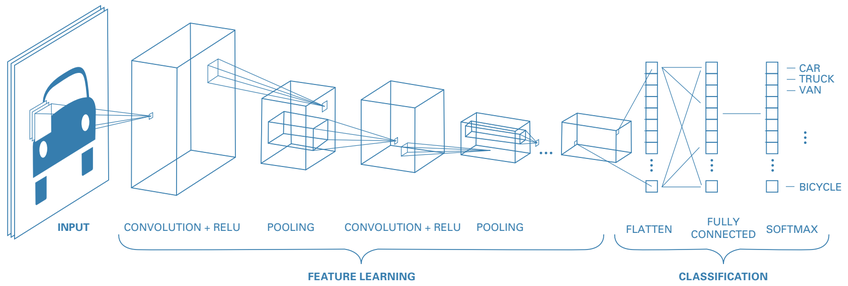


Models

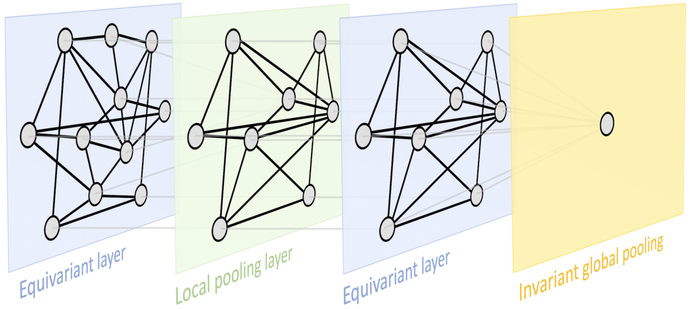
Multi-layer perceptron



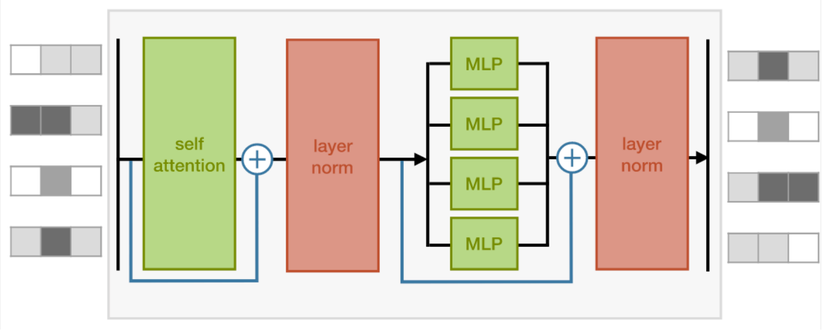
Convolutional neural network



Graph neural network



Transformer



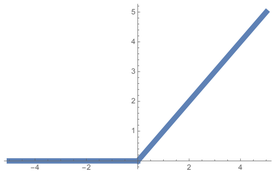
Multiple-layer perceptrons

$$\boldsymbol{y} = \sigma(W\boldsymbol{x} + b)$$

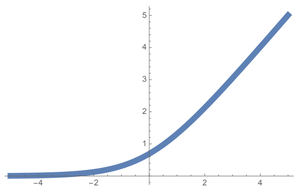
$$\boldsymbol{x} \in \mathbb{R}^{F_x} \qquad W \in \mathbb{R}^{F_y \times F_x} \qquad b \in \mathbb{R}^{F_y}$$

σ: elementwise activation

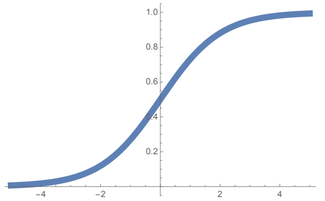
ReLU $= \max(0,x)$



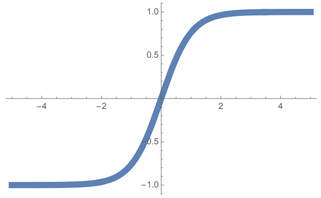
Softplus $= \ln(1 + e^x)$



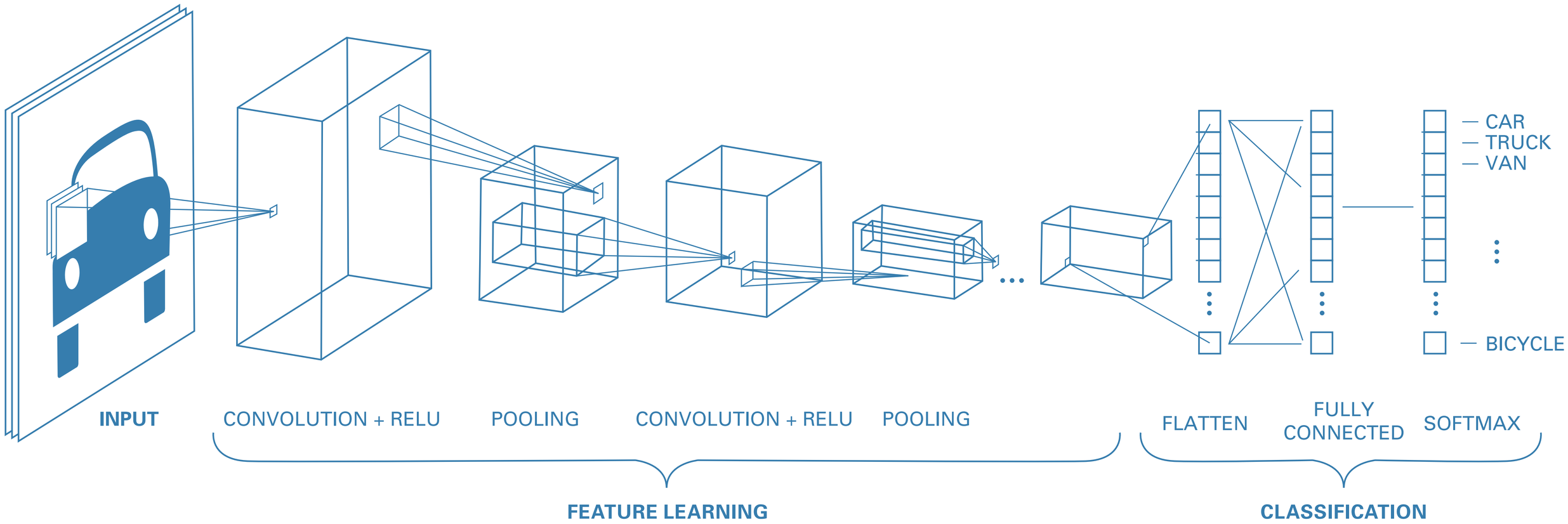
Sigmoid $= 1/(1 + e^{-x})$



tanh

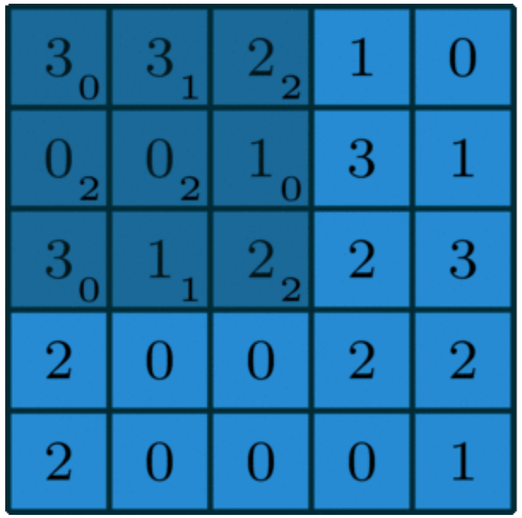


Convolutional neural networks



2d convolution (cross-correlation)

$$y_{ij} = (W \star x)_{ij} = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} W_{mn} x_{i+m,j+n}$$



Check 1603.07285 for more variations in padding, stride, dilation, ...

$$\begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} \star \begin{pmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{pmatrix}$$

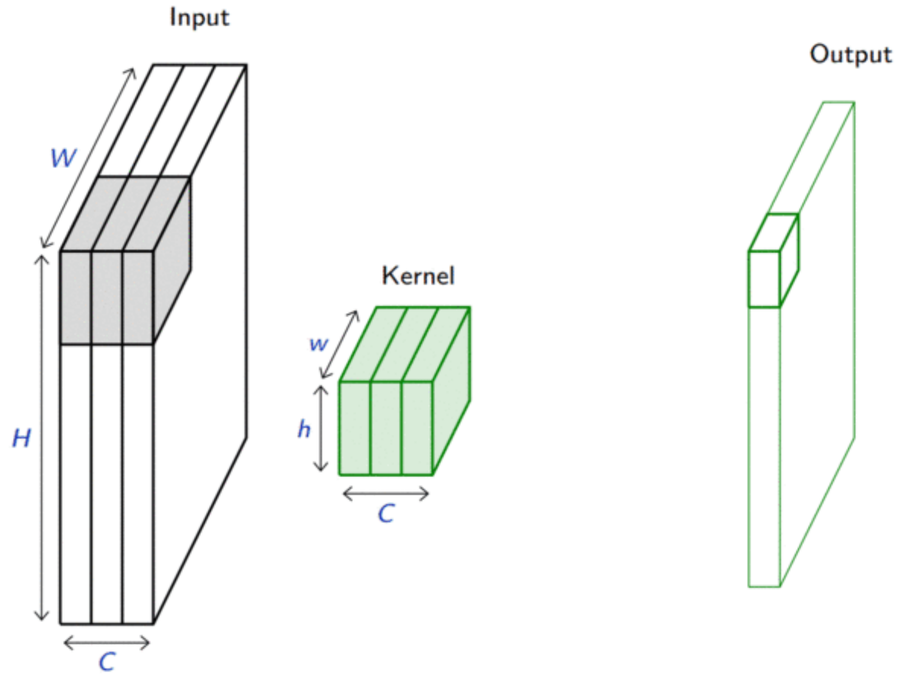
$$\equiv \begin{pmatrix} w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & 0 & w_3 & w_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix}$$

Equivalent to multiplying a sparse (doubly block circulant) weight matrix to flattened feature vector

Convolutional layer

$$y_{i,j,\mu} = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} \sum_{\nu=0}^{F_x-1} W_{m,n,\mu,\nu} x_{i+m,j+n,\nu} + b_\mu$$

$$\boldsymbol{x} \in \mathbb{R}^{H \times W \times F_x} \qquad W \in \mathbb{R}^{K \times K \times F_y \times F_x} \qquad b \in \mathbb{R}^{F_y}$$



Why is convolution useful ?

Cat

Cat

$$\text{shift} \circ \text{conv} = \text{conv} \circ \text{shift}$$

Translational equivariance

(Ignore edge effect: infinite size or on a torus)

Pooling layer

0	1	2
3	4	5
6	7	8

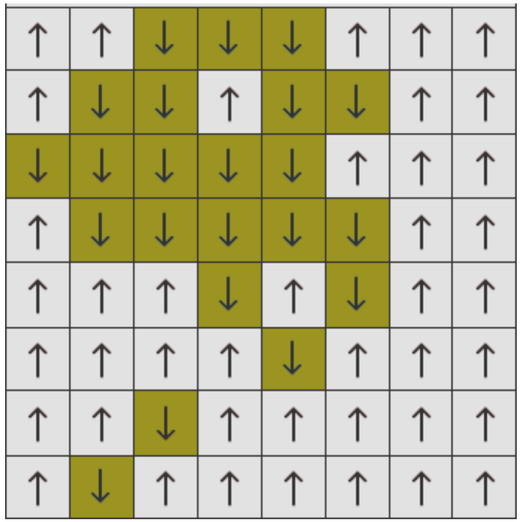
2 x 2 Max Pooling

4	5
7	8

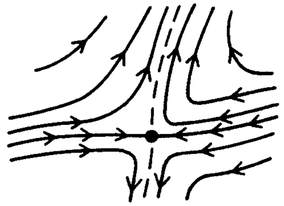
$$y_{i,j,\mu} = \max_{0<m<K} \max_{0<n<K} x_{i+m,j+n,\mu}$$

Detects a signal irrespective of of its exact location

Kadanoff's block spin transformation



Is the "connection" to renormalization group useful ?

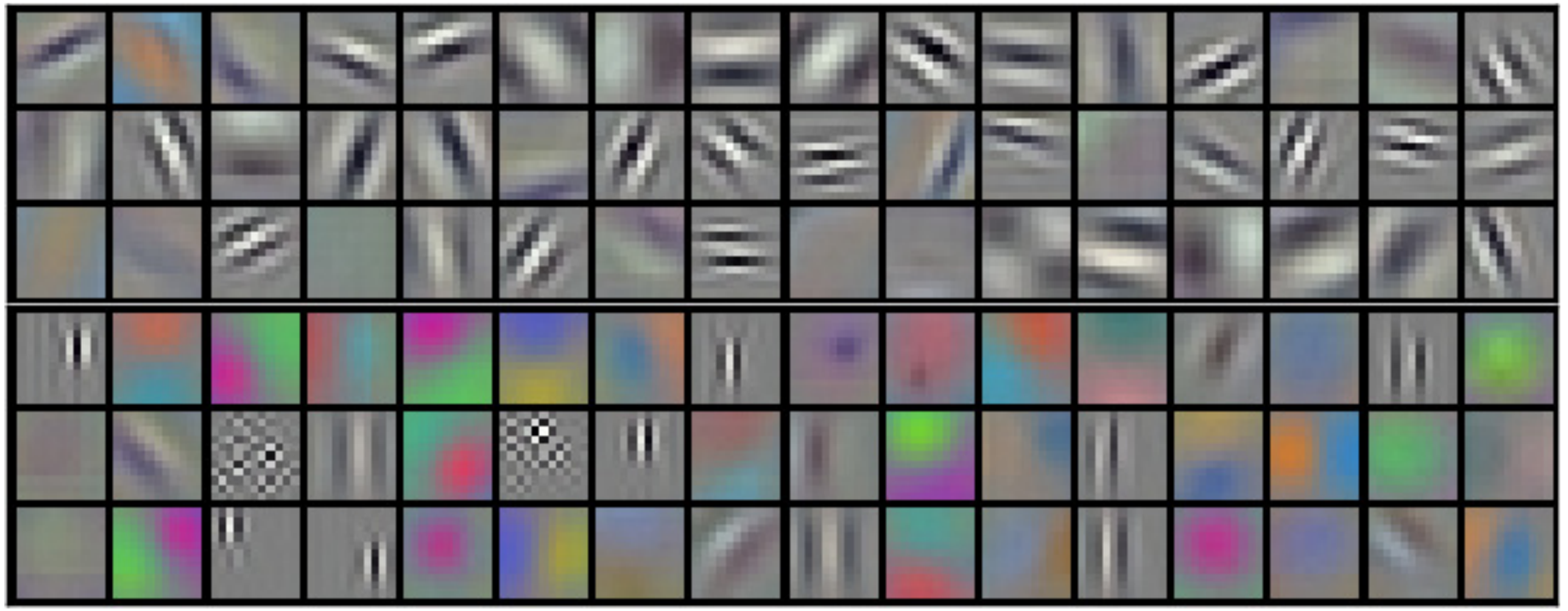


Many discussions, not conclusive yet.

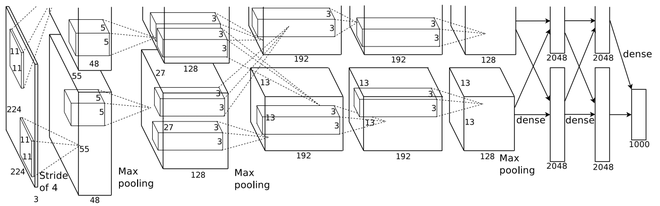
Bény, 1301.3124, Mehta, Schwab, 1410.3831
Lin, Tegmark, 1608.08225, Kenway, 1803.06111, ...

Learned feature detectors

$$W \in \mathbb{R}^{11 \times 11 \times 96 \times 3}$$



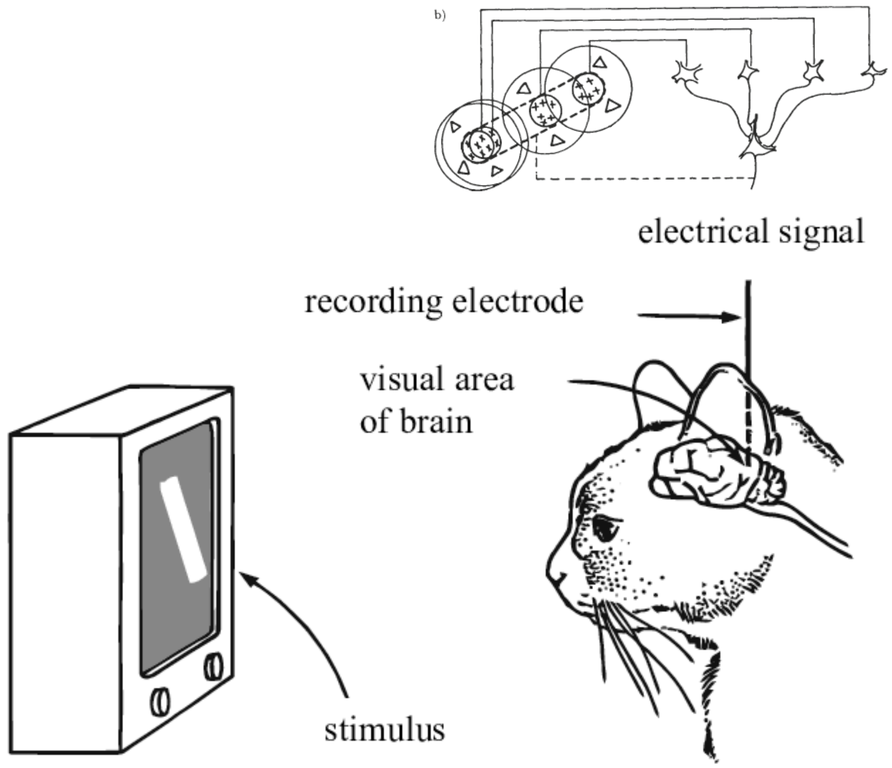
The kernels of the first convolutional layer of AlexNet



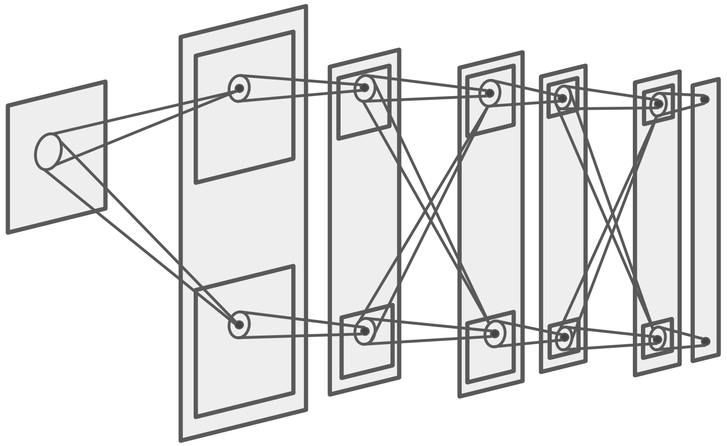
Trained on 2 GTX 580 GPU
with 3GB memory each

Krizhevsky, Sutskever, Hinton, 2012

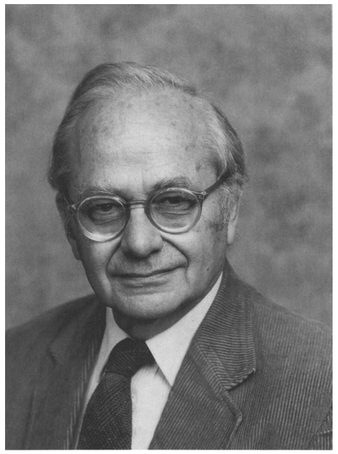
CNN and visual system



Hubel & Wiesel, 1959
Nobel prize 1981



“Neocognitron” Fukushima 1979



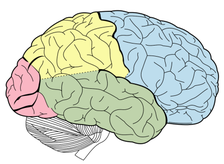
David H. Huber

This event has sometimes been held up as an example of the importance of "accident" in science. We have never felt that it was an accident. If there is something there to discover one has to take the time to find it, and one has to be relaxed enough about the way one works so as not to foreclose the unexpected. Two other groups failed to discover orientation selectivity because they were too scientific, in a simplistic sense of that word: one group built a device to generate horizontal bright bars, the other group, vertical, in both cases so that they could explore the retina more efficiently than with a roving spot. In a certain early phase of science a degree of sloppiness can be a huge advantage.

https://www.sfn.org/-/media/SfN/Documents/TheHistoryofNeuroscience/Volume-1/c9.pdf

Frequently asked questions

Biologically-inspired learning?

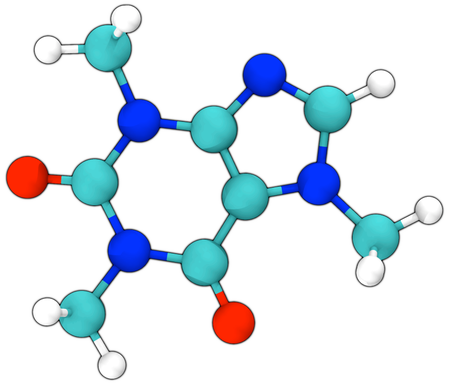
		
Power	10 W	300 W
Frequency	< kHz	GHz
Learning	Spikes	Gradient

Human intelligence is an existence proof of **artificial** intelligence

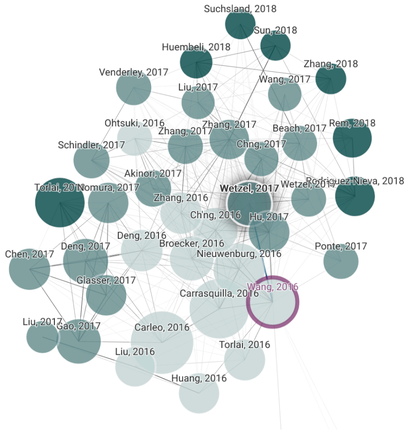
"Can a plane fly, can a submarine swim ?"

Learning on graphs

Molecule graph



Citation graph



Social network

Traffic map

tasks

Graph level task

How does the molecule smell ?

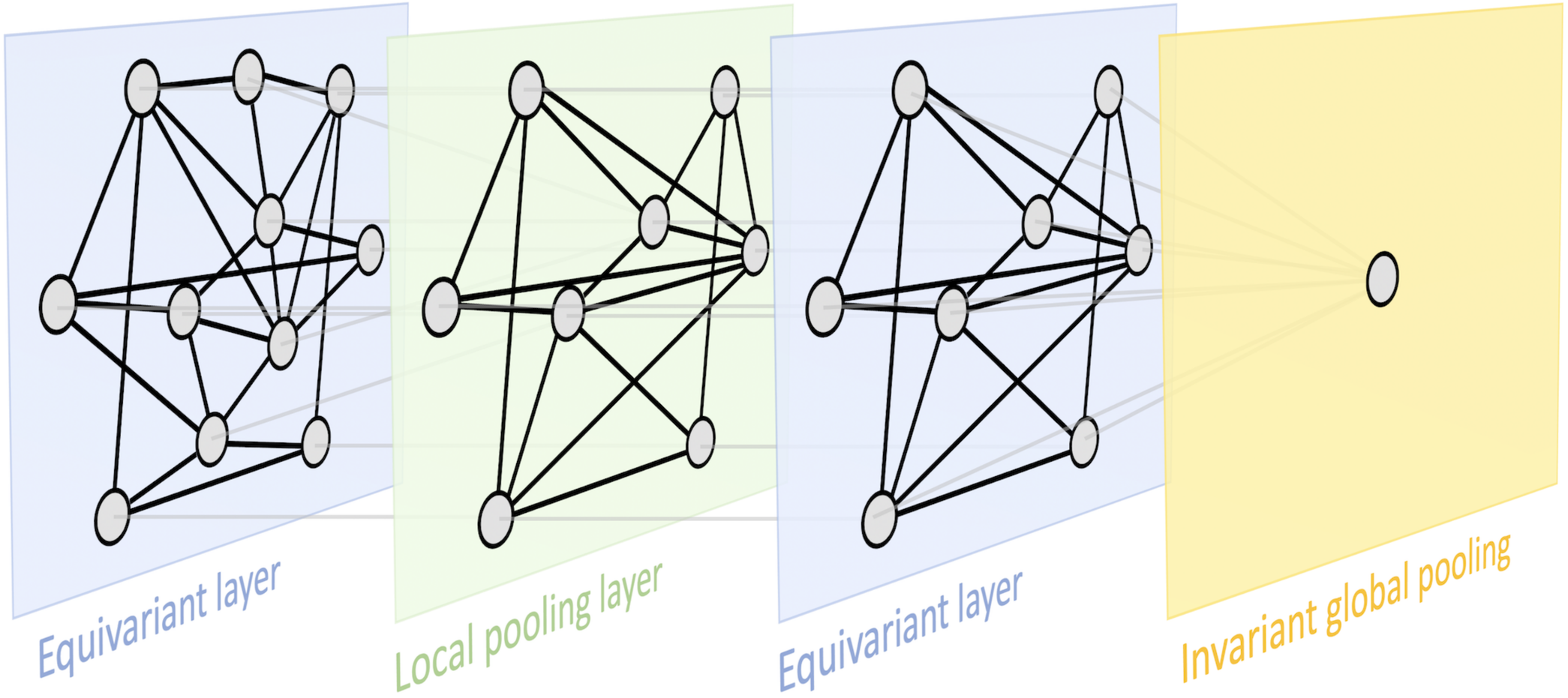
Node level task

Force on each atom ?

Edge level task

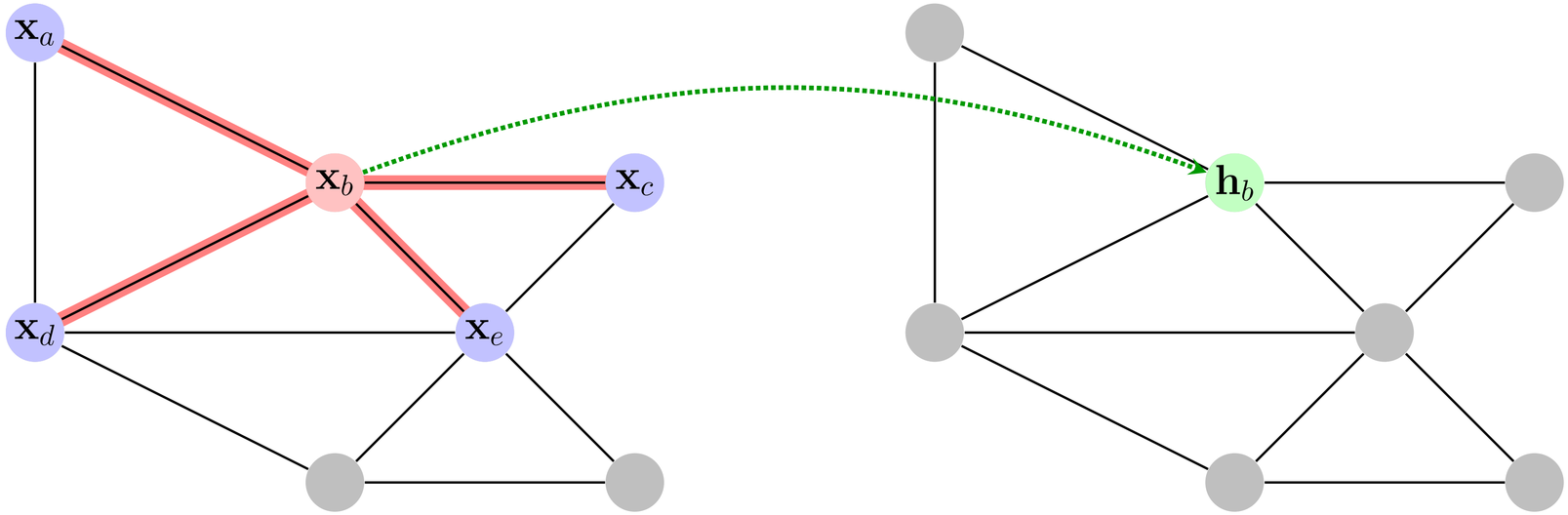
Hamiltonian of the molecule?

Graph neural network



Neural message passing

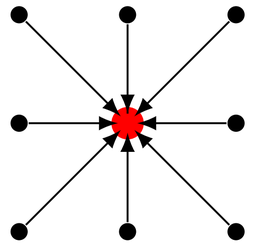
Gilmer et al, 1704.01212



$$\boldsymbol{h}_b = \phi\left(\boldsymbol{x}_b, \bigoplus_{v \in \mathcal{N}(b)} \psi(\boldsymbol{x}_b, \boldsymbol{x}_v)\right)$$

aggregation over neighbors

CNN passes message on a square grid

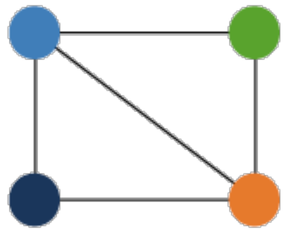


Graph convolution

Kipf, Welling, 1609.02907

$$\boldsymbol{h} = \phi(A\boldsymbol{x}W)$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



Adjacency matrix + I

$$A \in \mathbb{R}^{N \times N}$$

Weight matrix

$$W \in \mathbb{R}^{F_x \times F_h}$$

Input feature

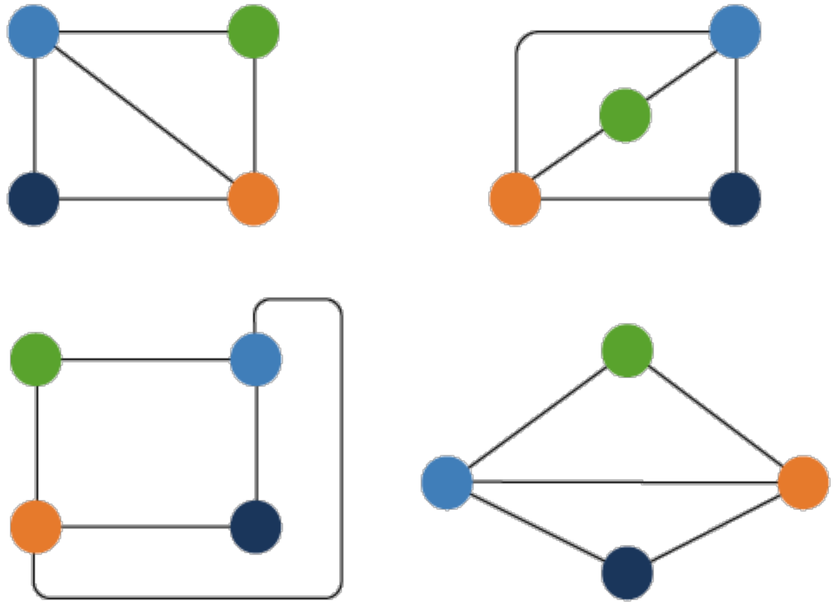
$$\boldsymbol{x} \in \mathbb{R}^{N \times F_x}$$

Output feature

$$\boldsymbol{h} \in \mathbb{R}^{N \times F_h}$$

ϕ Node wise activation

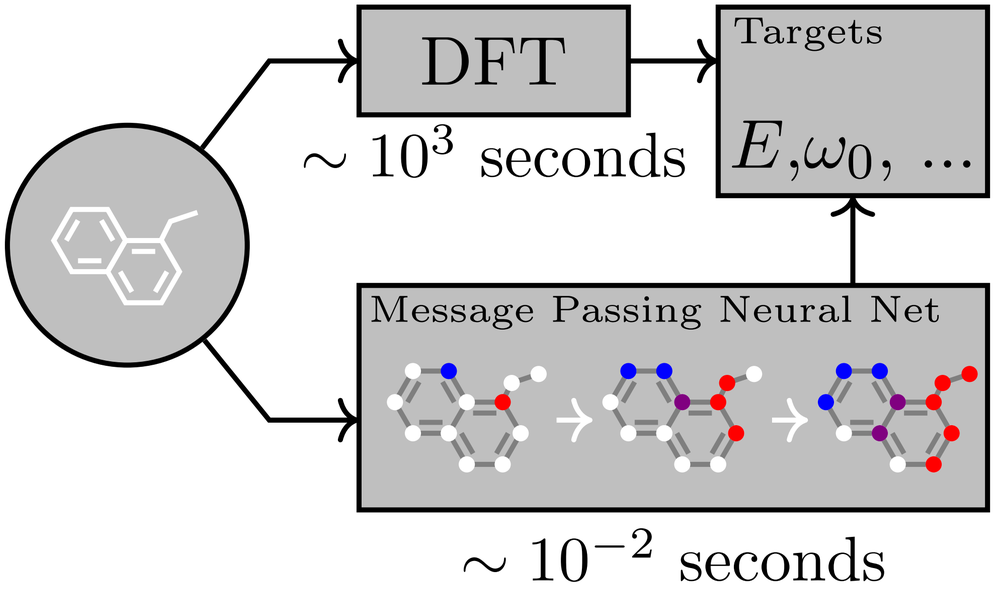
Graph Isomorphism



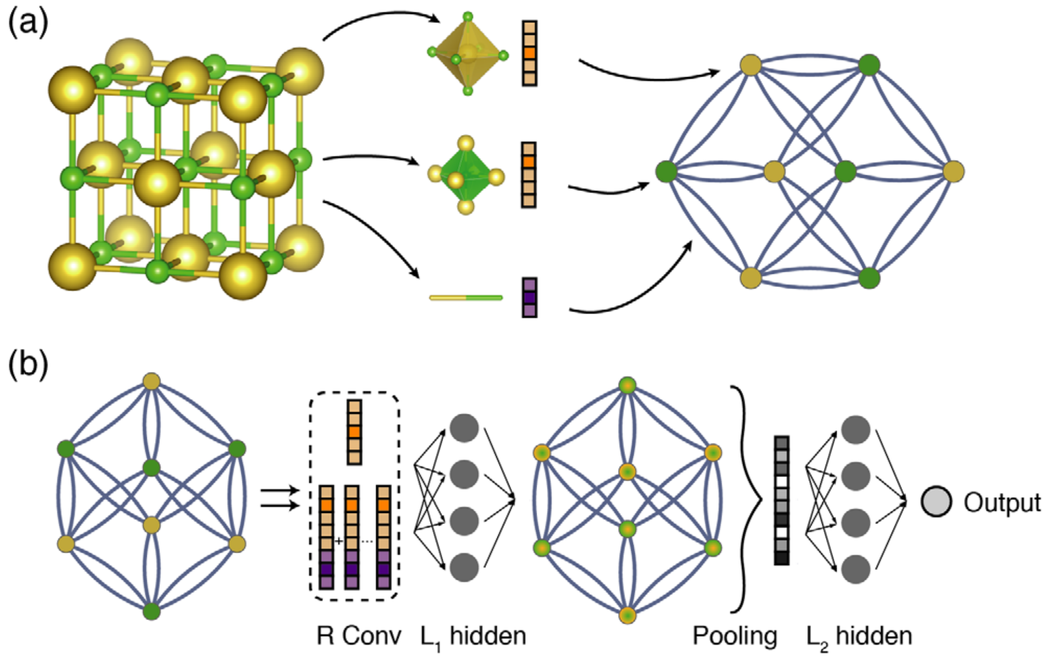
Graph convolution is equivariant to node permutation

$$\boldsymbol{Ph} = P\phi(A\boldsymbol{x}W) = \phi(PAP^T P\boldsymbol{x}W)$$

GNN for chemistry and material



Gilmer et al, 1704.01212

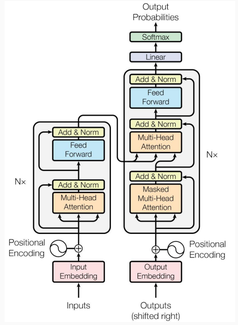


Xie et al, PRL '18

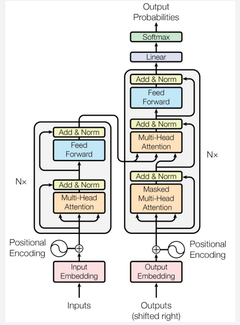
Attention is all you need

Vaswani et al, 1706.03762

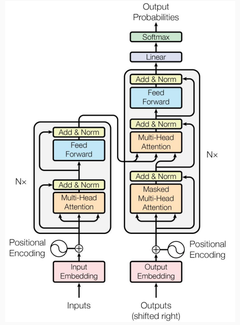
Computer Vision



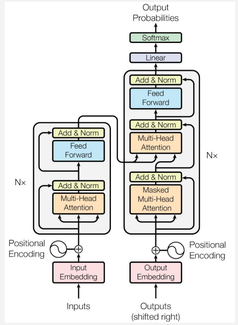
Natural Lang. Proc.



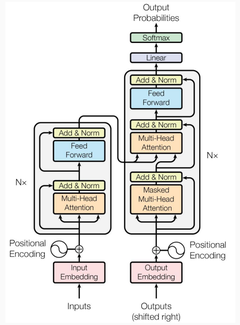
Reinf. Learning



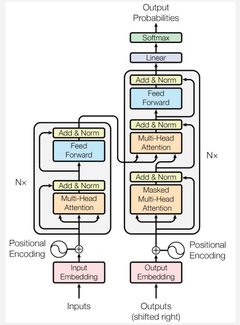
Speech



Translation



Graphs/Science

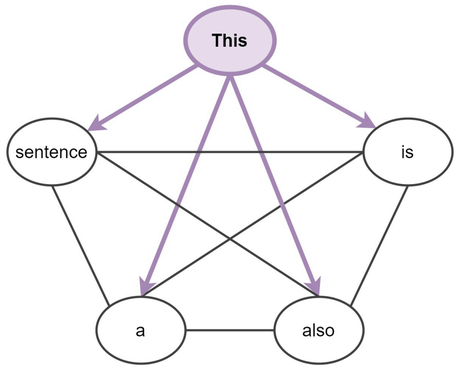


Transformer image source: "Attention Is All You Need" paper

Attention mechanism

Up to now, we've seen models (MLP, CNN, GNN) process information depending on its relative locations.

Attention mechanism process information depending on its actual content.



Attention is "soft" dictionary

$$\sum_j \alpha(q_i, k_j) v_j$$

$\alpha(q_i, k_j)$ → attention weight

```
d = {'a' : 1, 'b' : 2, 'c' : 3}

d['b'] = 3
```

query, key, value

key	value
a	1
b	2
c	3

https://dlvu.github.io/slides/dlvu.lecture12.pdf

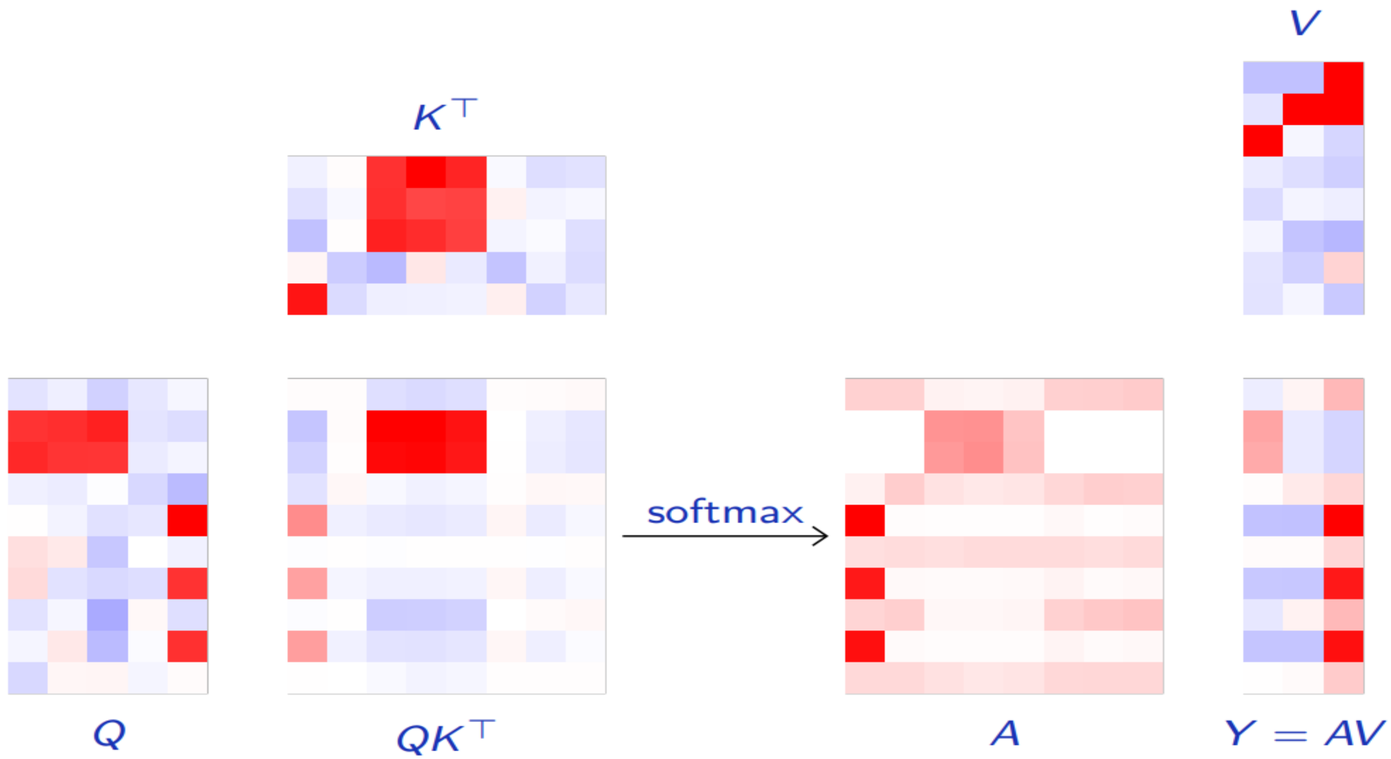
Scaled dot product attention

$$Q \in \mathbb{R}^{N \times d}$$

$$K \in \mathbb{R}^{N \times d}$$

$$V \in \mathbb{R}^{N \times d_v}$$

$$\text{attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

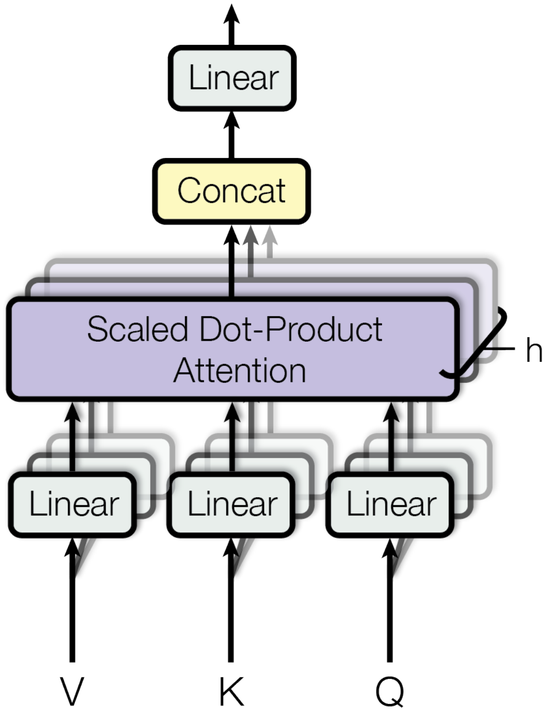


Multihead attention

Project Q, K, V to h heads,

Run multiple attention layers in parallel

Concatenate the output together

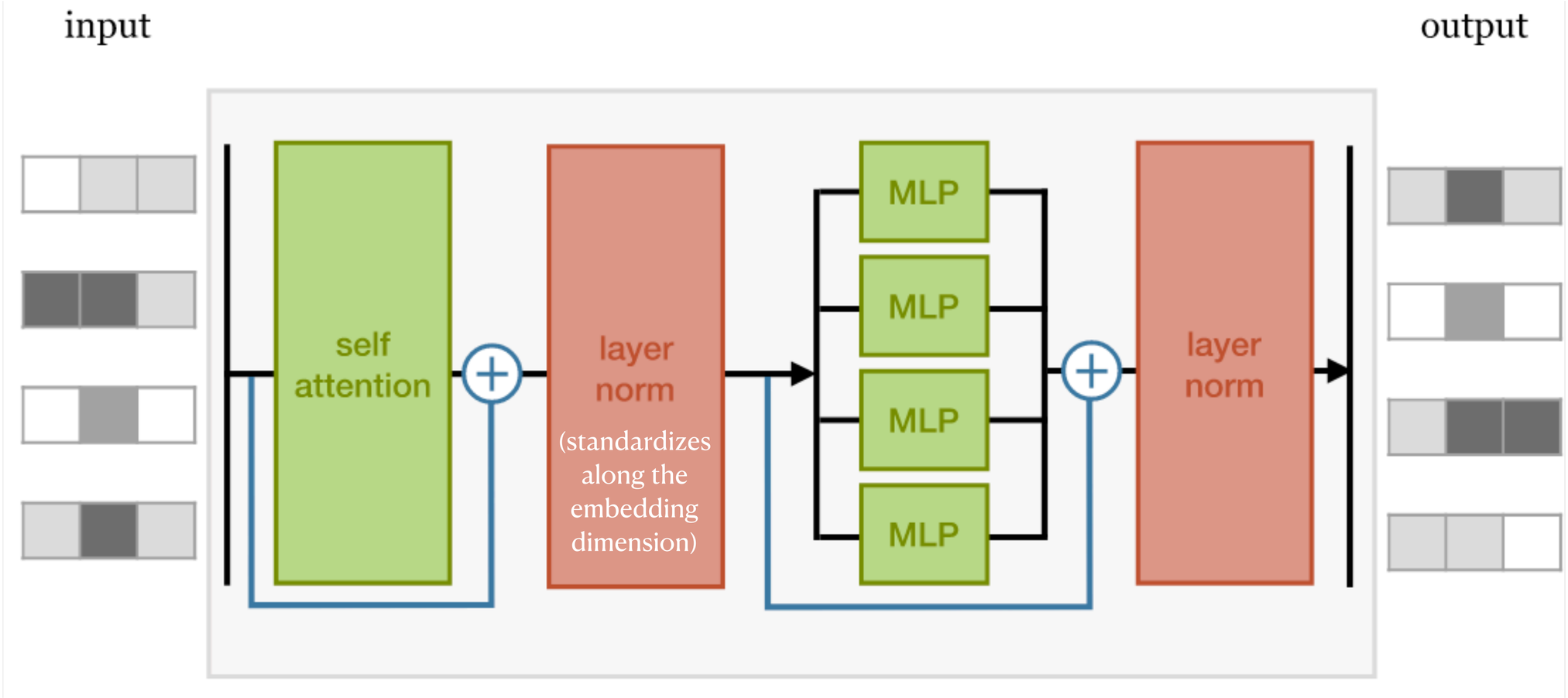


$$\text{mha}(Q, K, V) = [\text{head}_1, \ldots, \text{head}_h]W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$W_{1\ldots h}^Q \in \mathbb{R}^{F\times d} \qquad W_{1\ldots h}^K \in \mathbb{R}^{F\times d} \qquad W_{1\ldots h}^V \in \mathbb{R}^{F\times d_v} \qquad W^O \in \mathbb{R}^{h\cdot d_v\times d_{out}}$$

The transformer block



permutation equivariant $P \circ T = T \circ P$

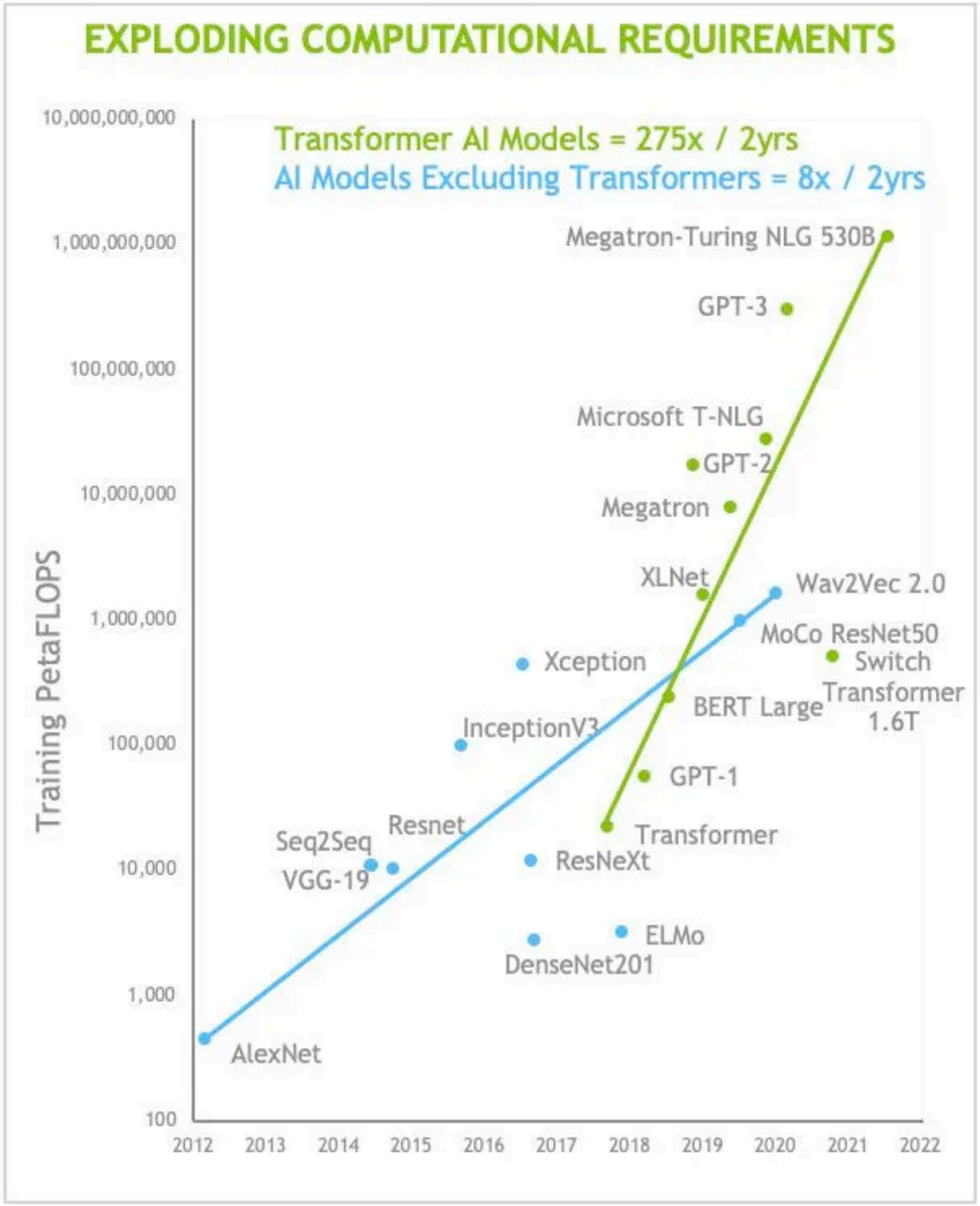
add position embedding if position is needed

Why transformer ?

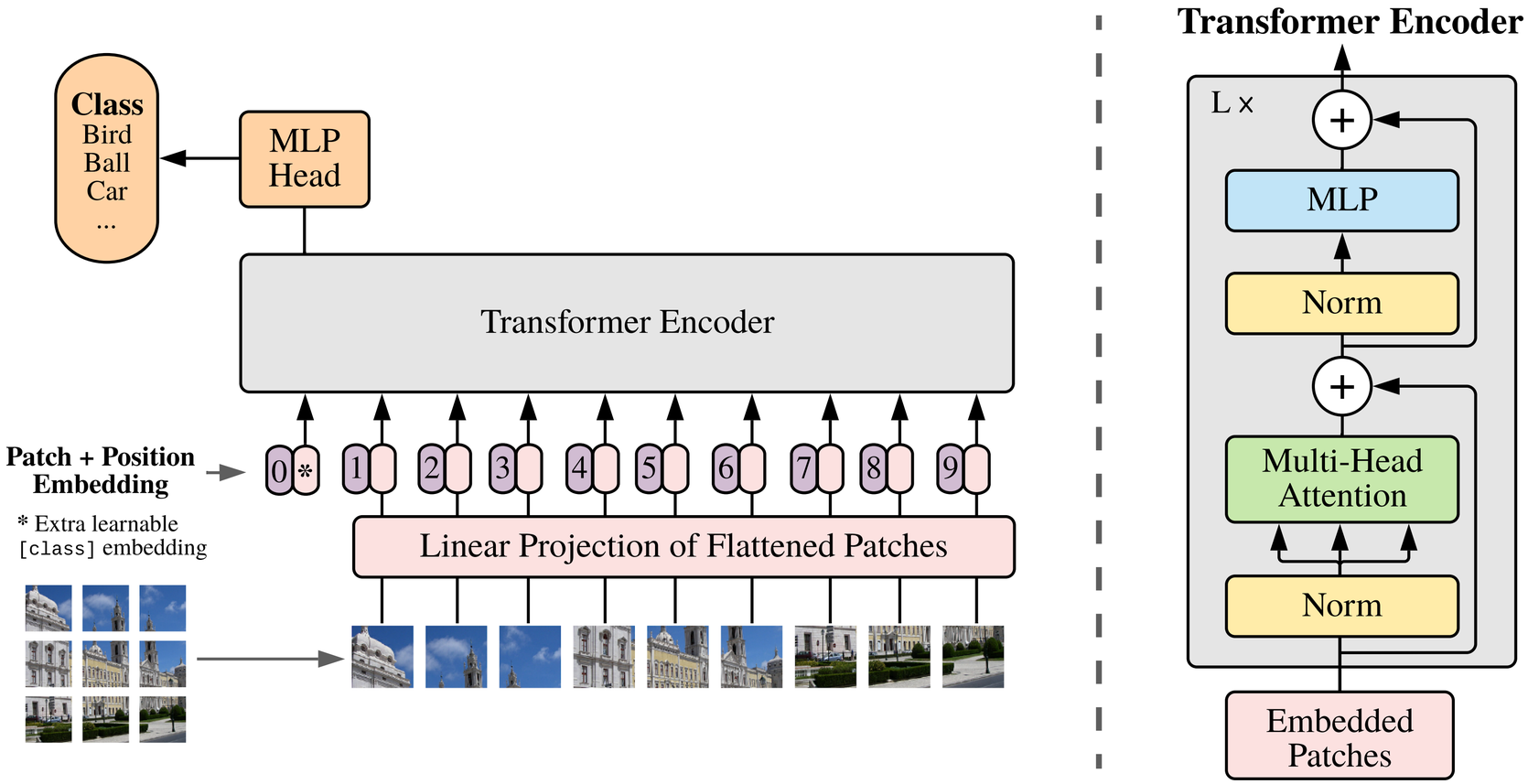
Layer type	Complexity	Sequential ops.	Max. path length
Self-attention	$O(n^2 d)$	$O(1)$	$O(1)$
Recurrent	$O(nd^2)$	$O(n)$	$O(n)$
Convolutional	$O(knd^2)$	$O(1)$	$O(\log_k n)$

n: sequence length *d*: feature dimension *k*: kernel size

- Directly captures *dynamical* long-range dependence (+transformer circuit math https://transformer-circuits.pub/2021/framework/index.html)
- Friendly to backpropagation
- Great usage of hardware accelerators



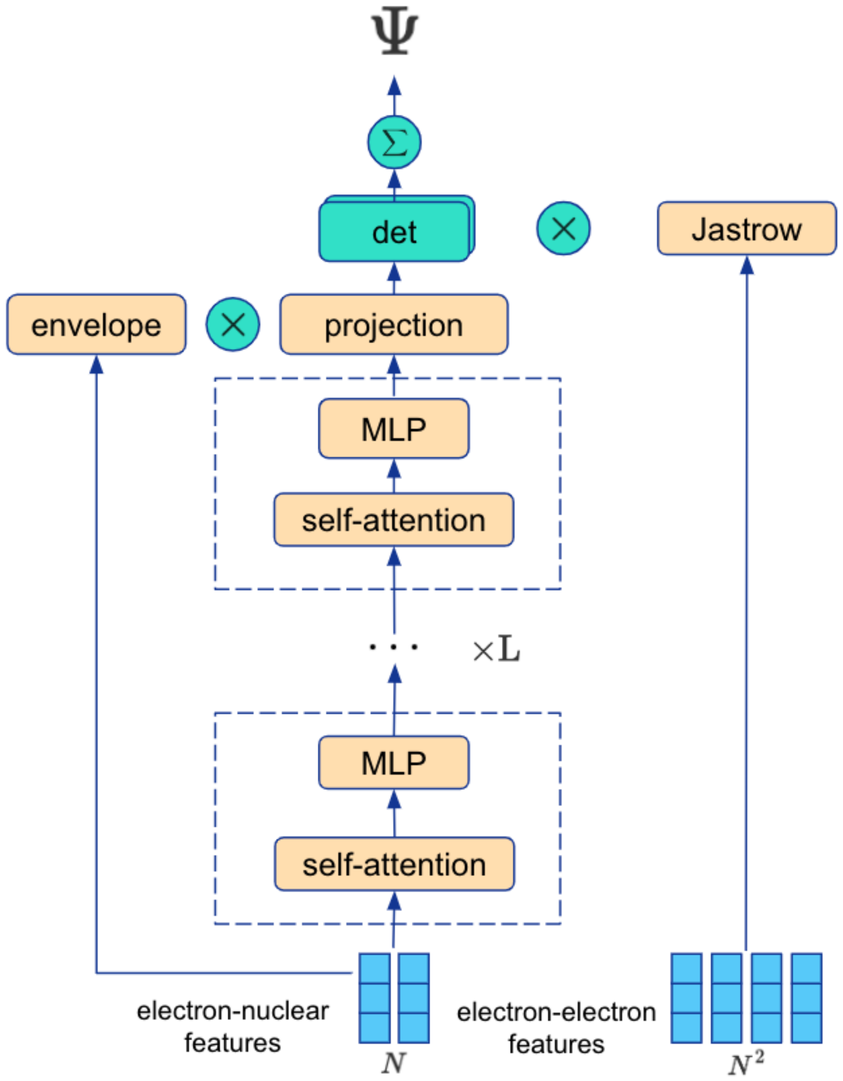
Vision transformer



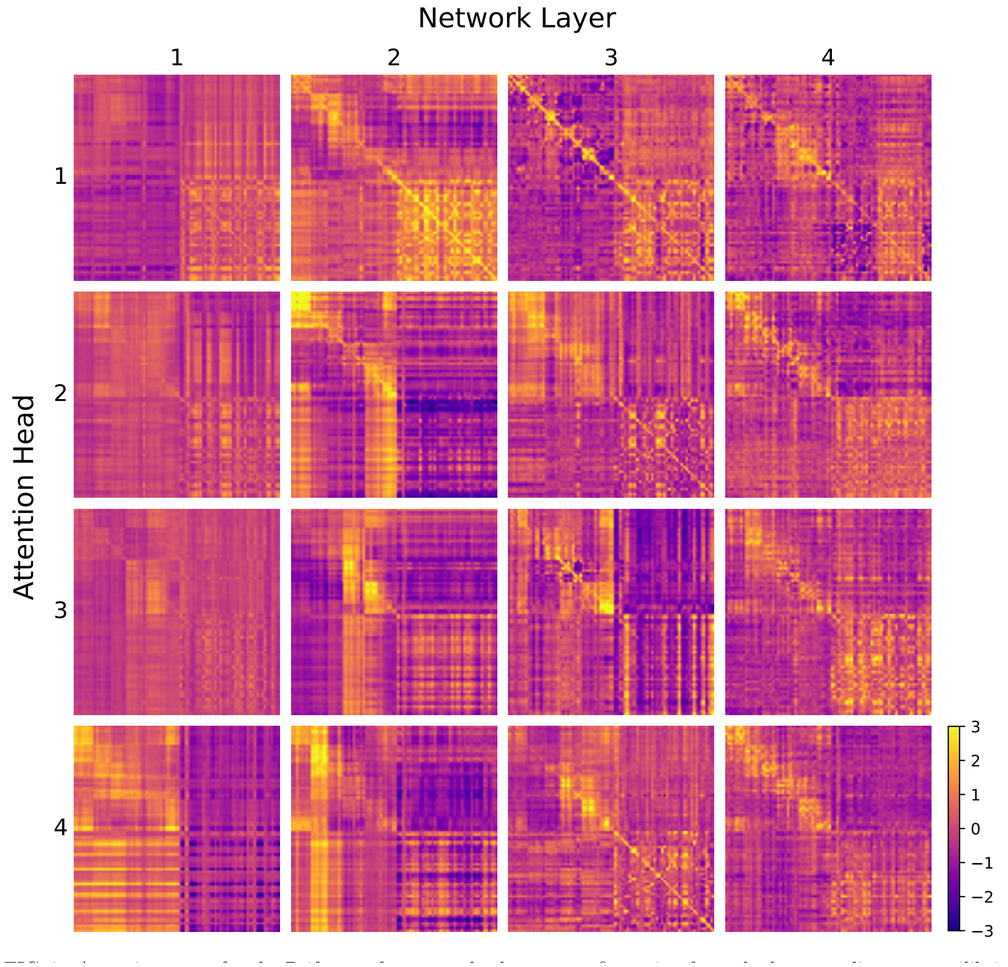
"Vision Transformer attains excellent results compared to state-of-the-art convolutional networks "

Dosovitskiy et al, 2010.11929

Psiformer: a self-attention quantum state



von Glehn et al, 2211.13672



Attention map of electron of a benzene dimer
"Electrons attend more to other electrons around the same molecule"

Cost function

Maximum likelihood estimation

In general, we can think of the neural network as representing a function $f(\boldsymbol{x}; \boldsymbol{\theta})$. The outputs of this function are not direct predictions of the value $\boldsymbol{y}$. Instead, $f(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\omega}$ provides the parameters for a distribution over y. Our loss function can then be interpreted as $-\log p(\mathbf{y}; \boldsymbol{\omega}(\boldsymbol{x}))$.

Goodfellow, Bengio, Courville, Deep Learning

Discriminative learning

$$\mathscr{L} = - \mathop{\mathbb{E}}_{(\boldsymbol{x},y)} \Big[\ln p(y \,|\, \boldsymbol{x})\Big]$$

Generative learning

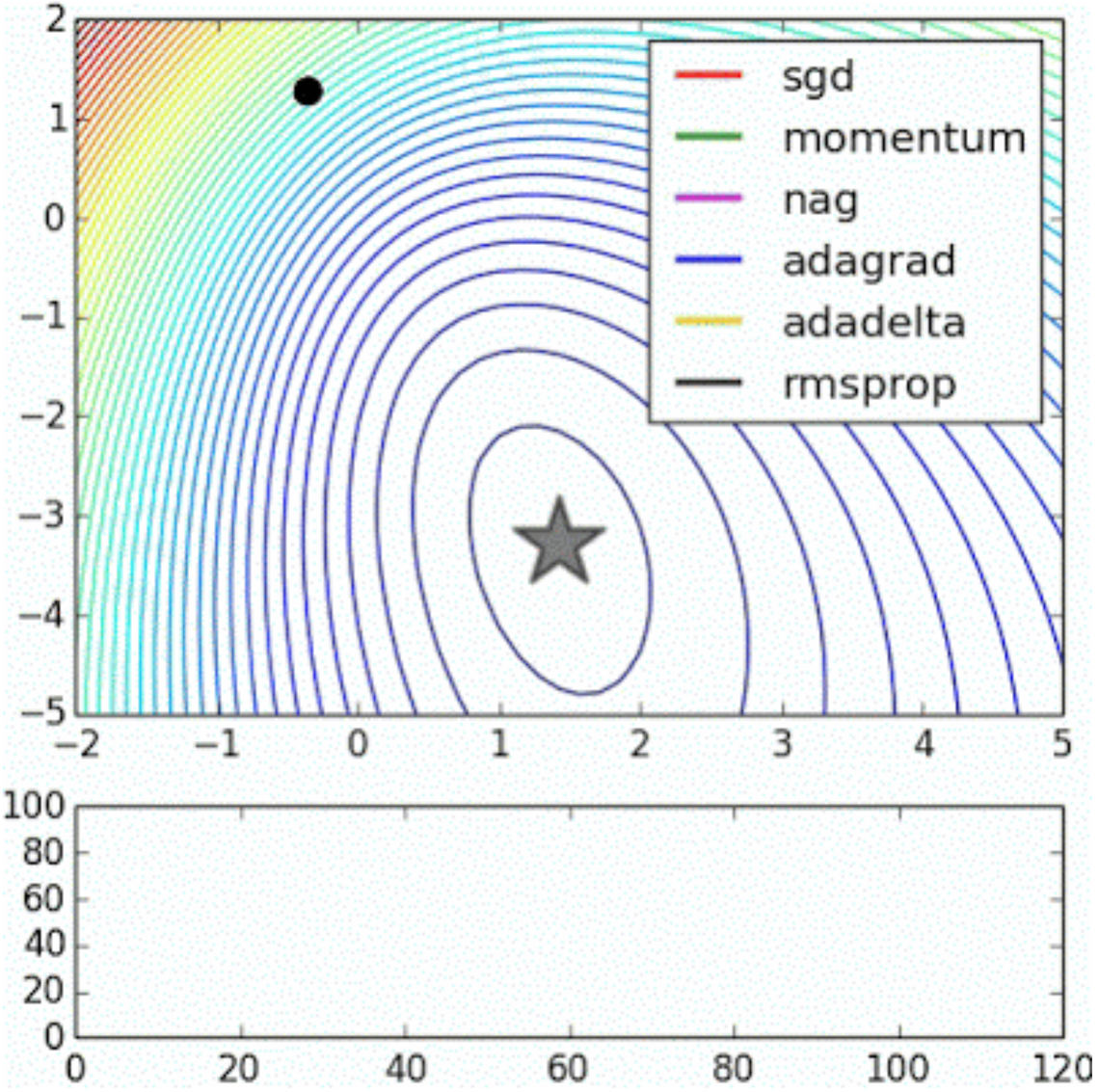
$$\mathscr{L} = - \mathop{\mathbb{E}}_{\boldsymbol{x}} \Big[\ln p(\boldsymbol{x})\Big]$$

Optimization

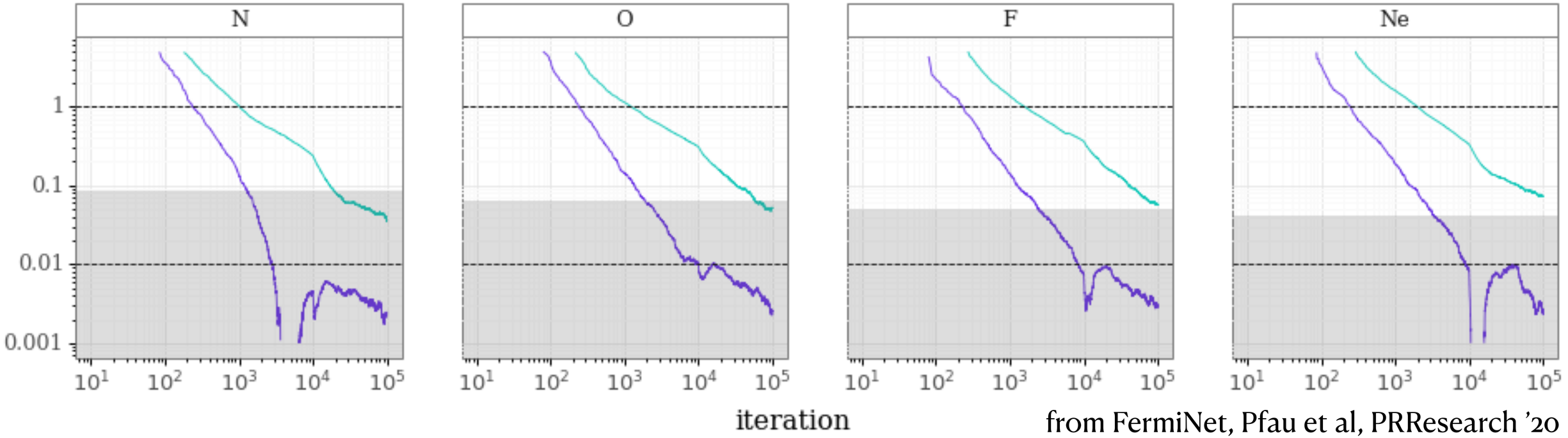
Stochastic gradient descent

$$\theta \leftarrow \theta - \eta \frac{\partial \mathscr{L}}{\partial \theta}$$

$$\mathscr{L} = \mathop{\mathbb{E}}_{x} \left[\ell(x) \right]$$



Higher-order optimizations

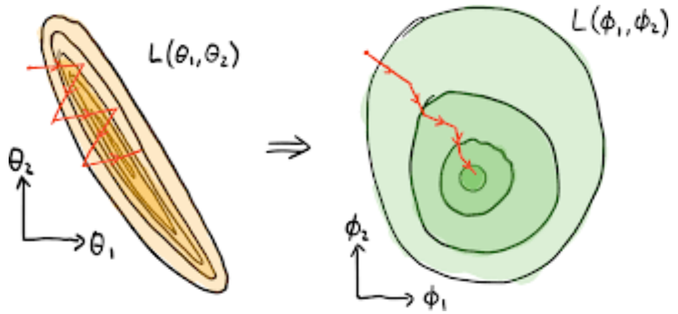


from FermiNet, Pfau et al, PRResearch '20

ADAM Kingma & Ba, 2014

KFAC Martens & Grosse 2015

$$\theta \leftarrow \theta - \eta G^{-1} \frac{\partial \mathscr{L}}{\partial \theta}$$



Natural gradient, Amari 1998

Stochastic reconfiguration, Sorella 2005

Hessian free optimization, Martens 2015

https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2022/