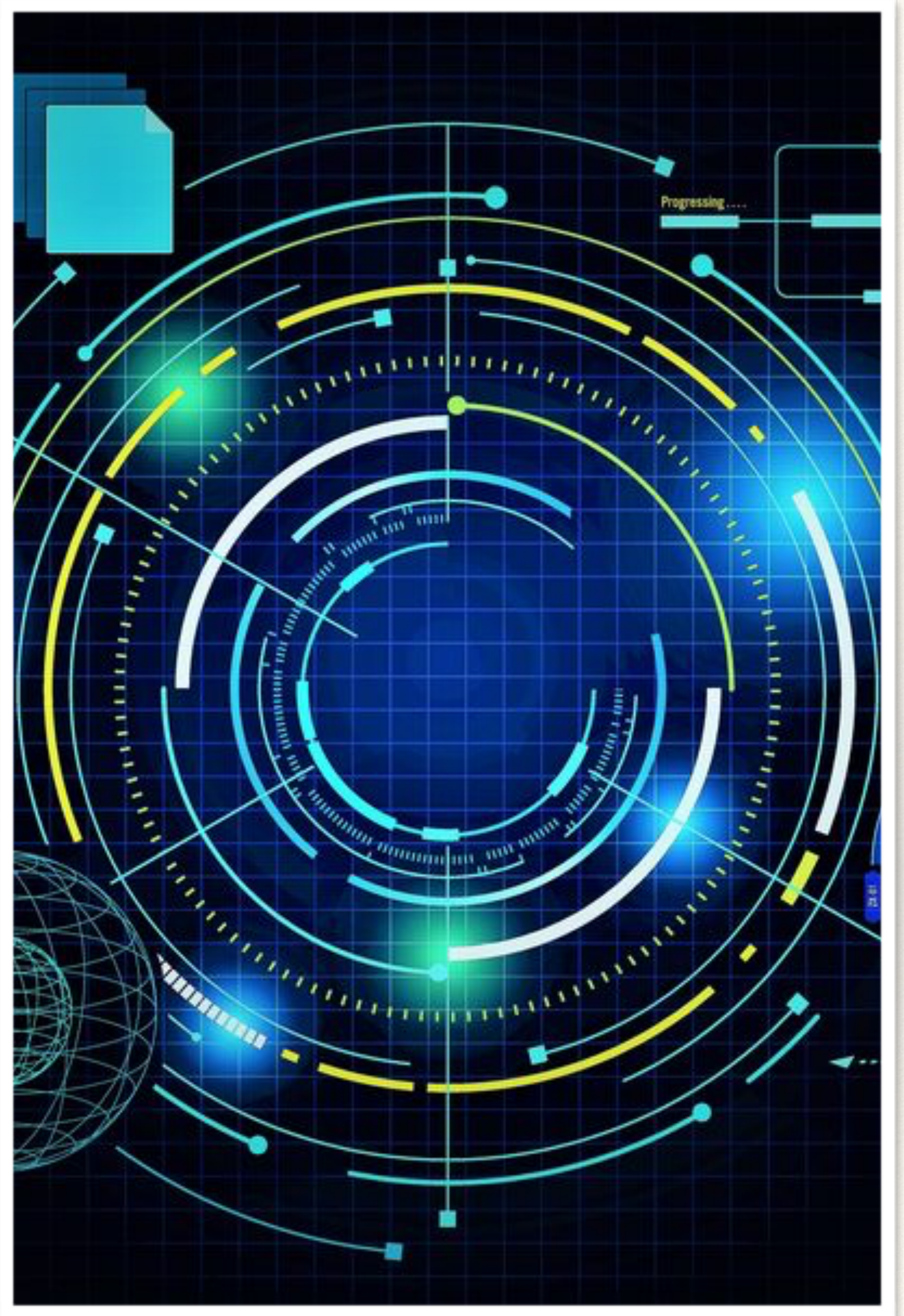
Postgraduate course
Universitat de Valencia 2020

Introduction to Machine Learning for physicists

Veronica Sanz (UV/IFIC)

LECTURE 7
TRANSFER & OTHER ADVANCED APPLs.



Transfer Learning

So far, our Machine was like a newborn baby
looking at a dataset/environment
and learning from it
with or without guidance, using rewards

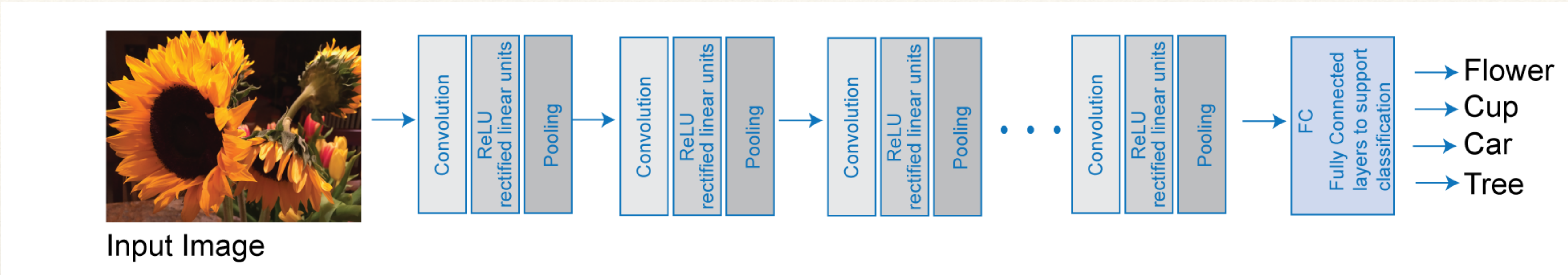
Complex datasets require Deep Learning
long time to run and optimise
and *specific* to the dataset

BUT babies do not learn every task from scratch

example: **language**
learn generic concepts
actions->verbs
objects/people->names
characteristics->adjectives

Babies are able to
TRANSFER LEARNING
ENGLISH -> SPANISH, CHINESE etc
and our Machine should do too

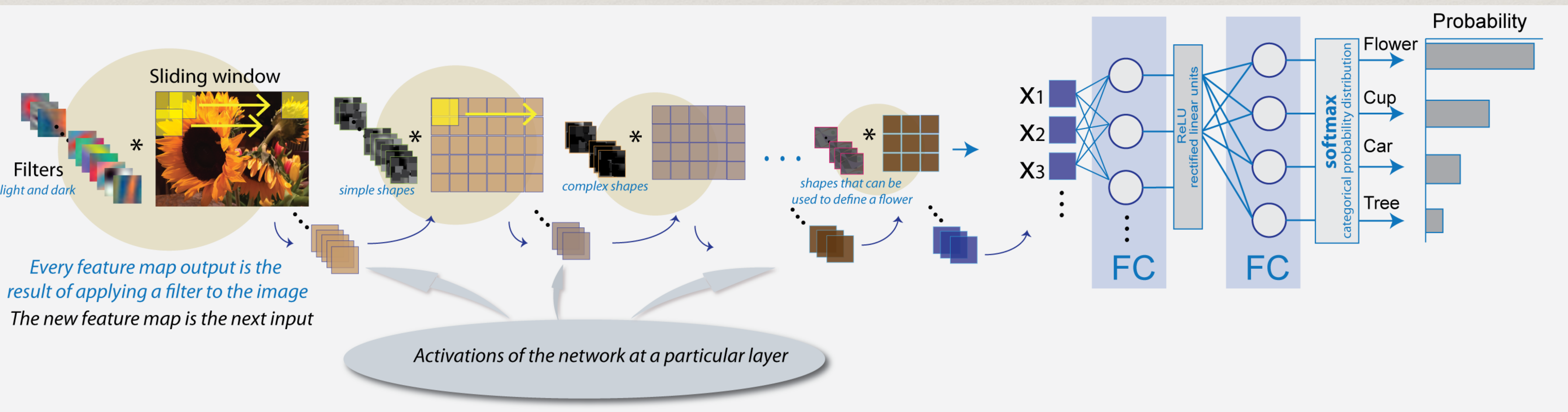
Supervised learning Image classification problem



Hidden layers are transforming the initial image into something more abstract (simple/complex shapes->shapes specific to a flower) and at the end this is transformed into a set of vectors which are then fed to a set of FC NNs

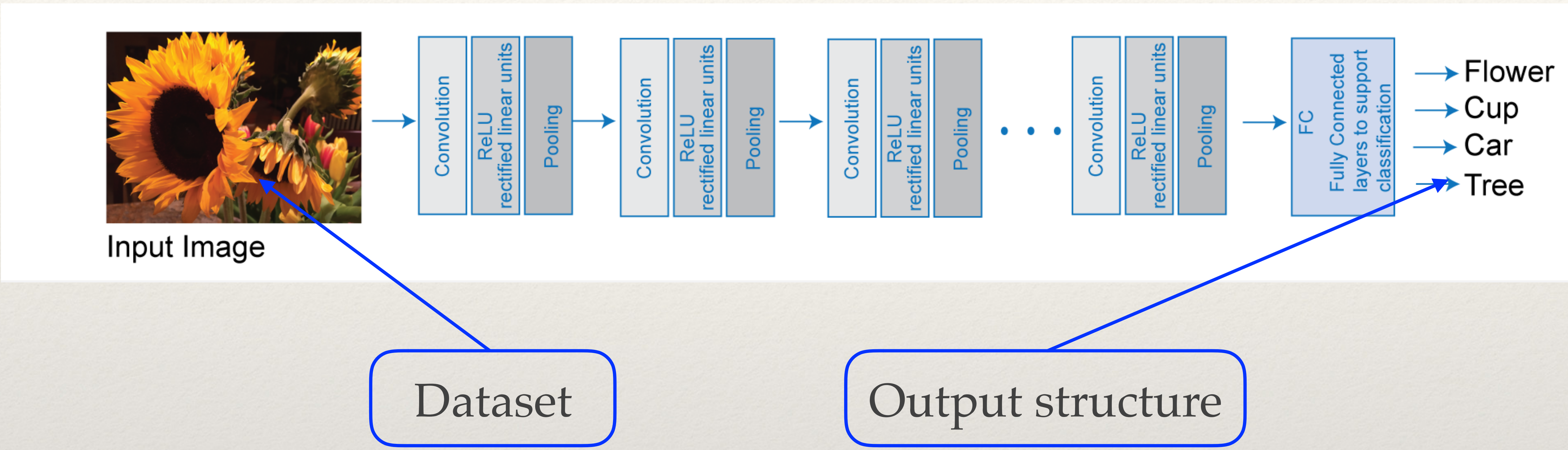
initial layers are more problem-independent

later layers are more specific to the problem at hand



Changing the target

What if now I want to classify types of dogs or leaves or cars?
I would start by changing…

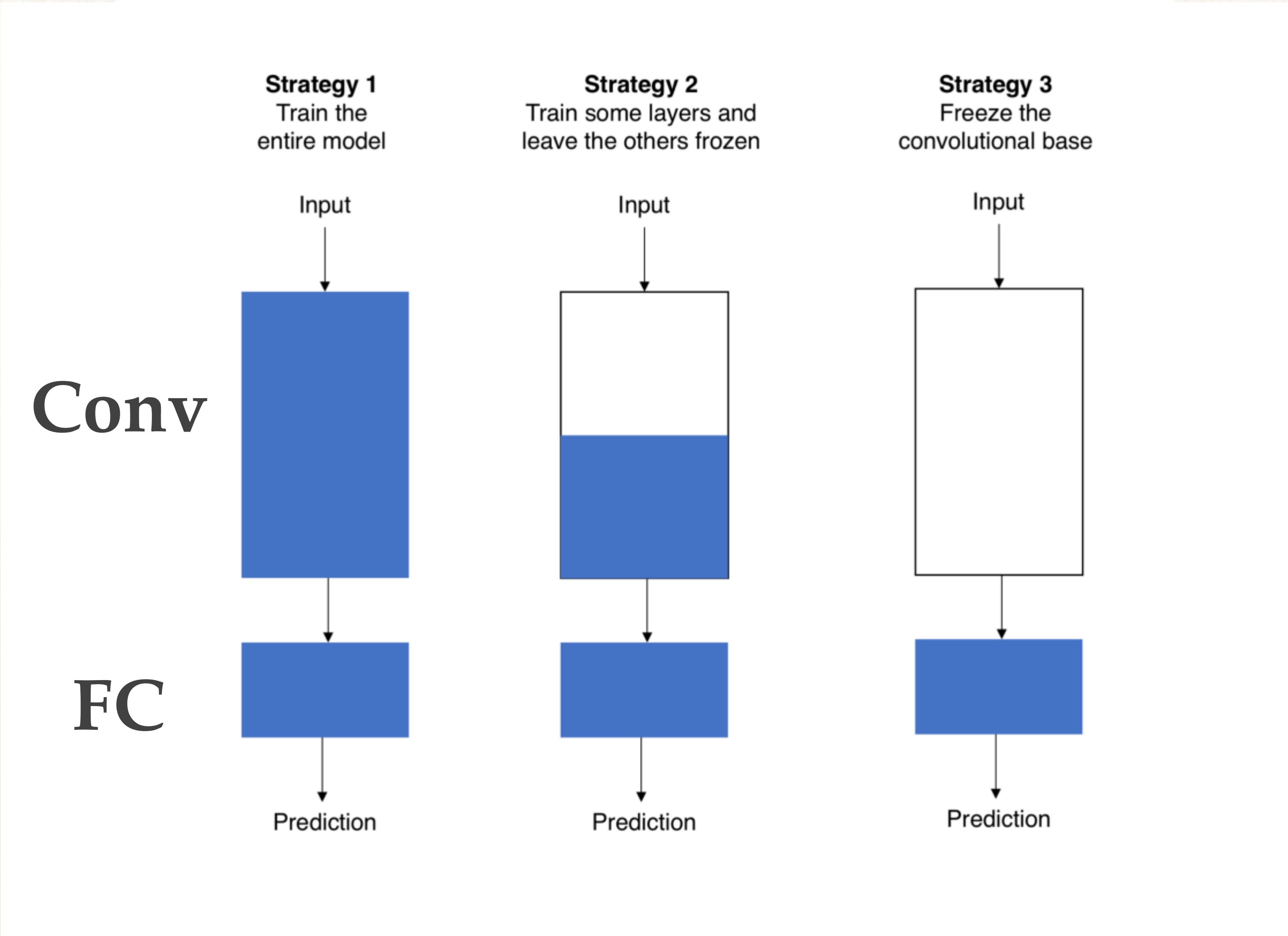


But probably build a very similar NN architecture

Transfer learning: allows me to keep some of the architecture and initial computations (some weights in the hidden layers) and just re-run part of the network to specialise on this new problem

Increase AI's speed, reusability and generalisation

Repurposing pre-trained networks



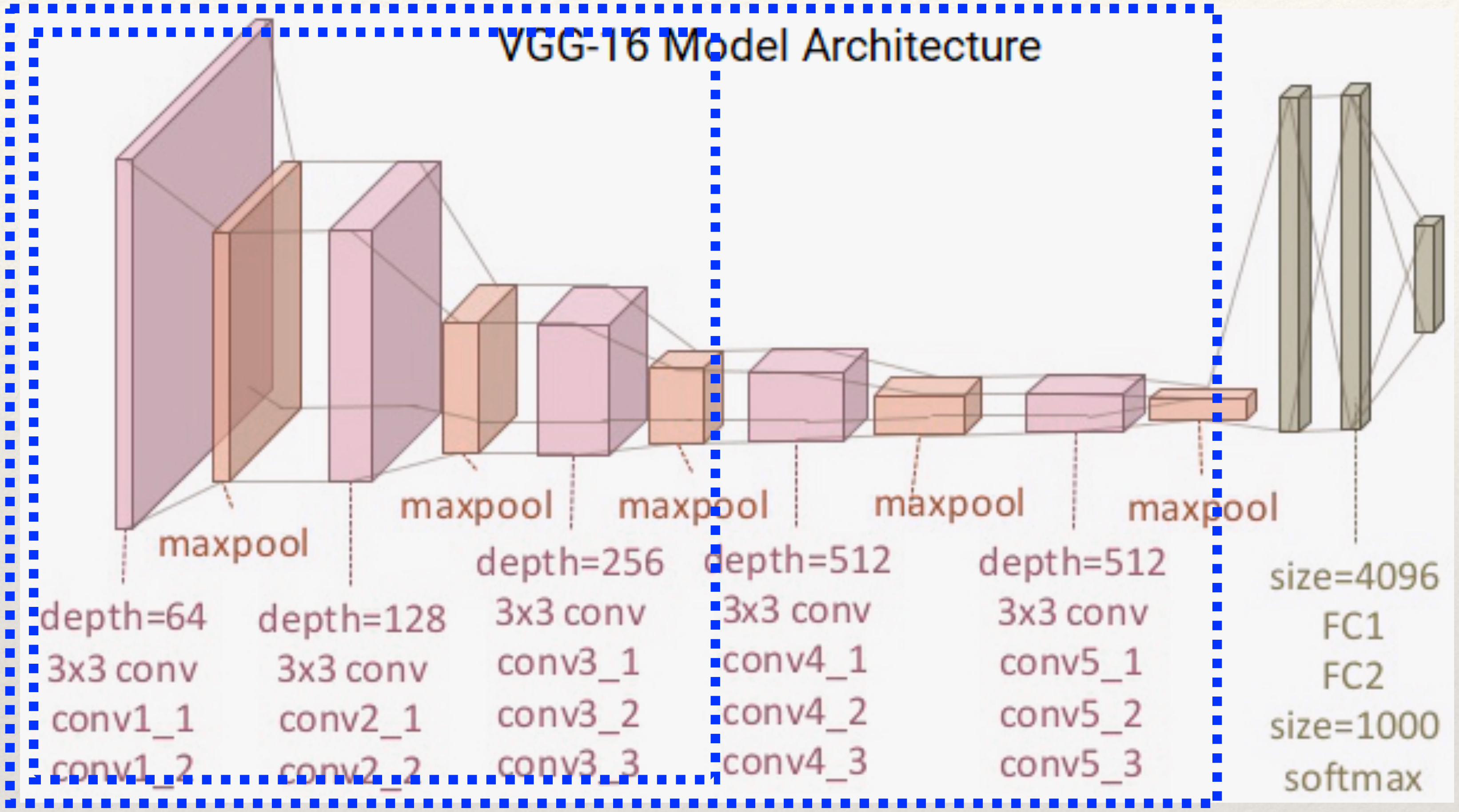
as we run the network, we update the weights to improve the accuracy
this weight updating is costly
we can **freeze** some of the weights that are generic for problem = image
and just adjust the later layers

All ML frameworks contained pre-trained models

Images==> VGG/ResNet/DenseNet/Inception/Xception…
Language (NLP) ==> Word2Vec, GloVe, FastText…

An example of pre-trained model: VGG16

Freeze initial layers and tune the rest



This strategy reduces a lot the computing time and helps generalising tasks

Data augmentation is typically used to make the procedure more robust

Wrapping up

The practical use of transfer learning is not difficult implemented and easy to use
(need to adapt input shape and output layer)
Today's notebook contains examples of this technique template to start exploring

Now

We are going to discuss some examples of non-standard applications of AI directly related to Physics and Math
I'm cherry-picking but there are plenty more out there!

ML & tradicional numerical problems

ML offers powerful techniques to sample in parameter space and speed up numerical computations

Remember NNs have a huge expressivity, can quickly explore relations among inputs and outputs

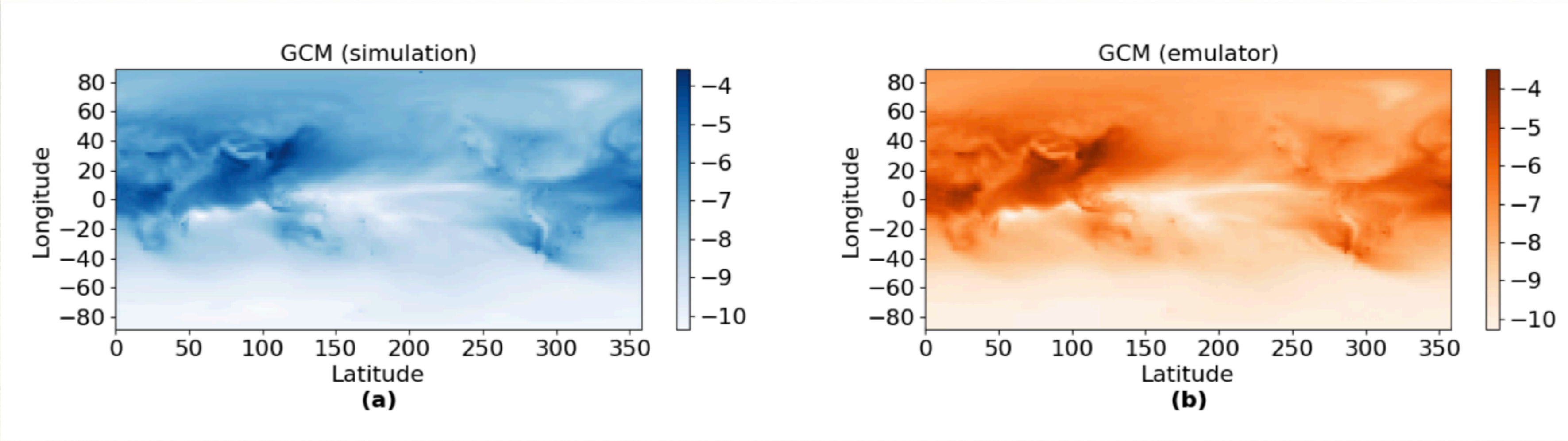
Yesterday we saw an example: fluid dynamics and RL

One particularly impressive <u>example</u>

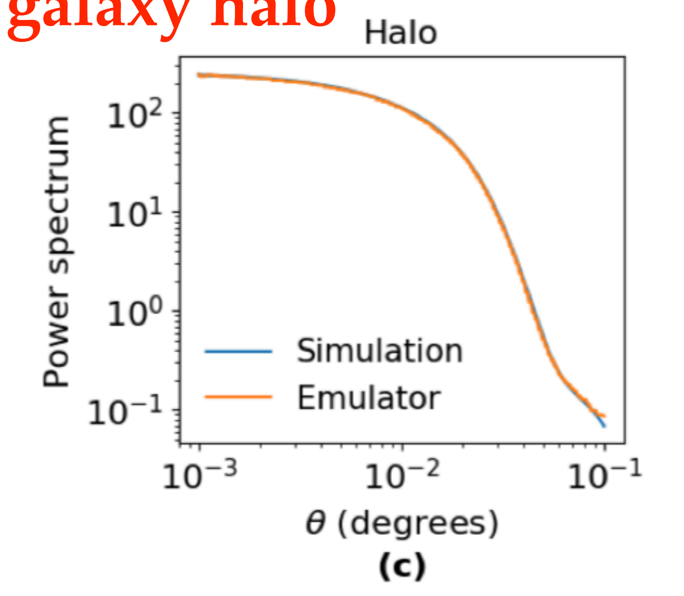
"Up to two billion times acceleration of scientific simulations with deep neural architecture search"

The method successfully accelerates simulations by up to 2 billion times in 10 scientific cases including astrophysics, climate science, biogeochemistry, high energy density physics, fusion energy, and seismology, using the same super-architecture, algorithm, and hyperparameters.

aerosol circulation



galaxy halo



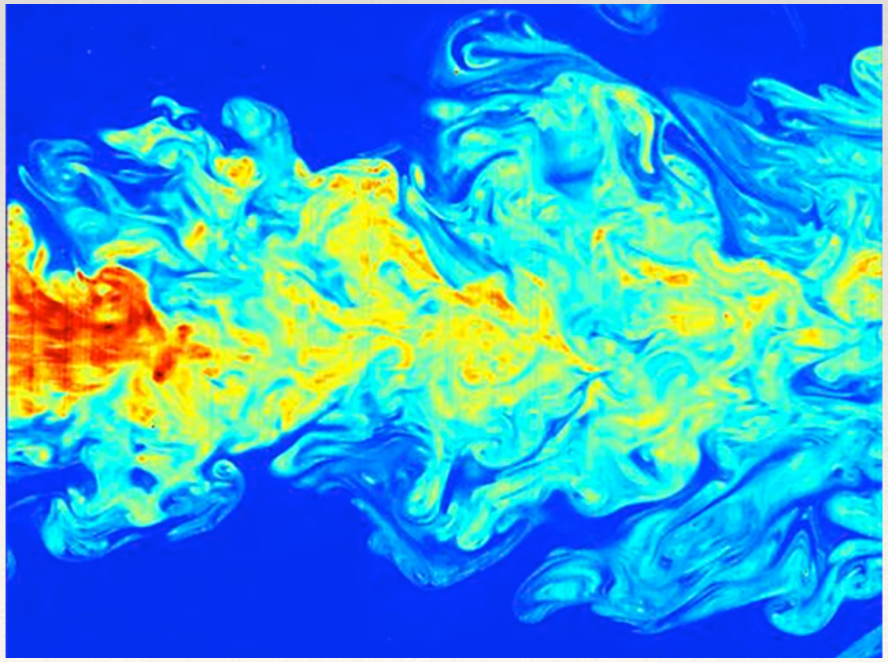
Numerical acceleration is great but what is a highly-nontrivial set of problems?

Inverse problems

data-> find laws

a fit to data is a simple example of trying to solve the inverse problem (see also this example)

if I showed you many examples of fluid behaviour



would you learn the Navier-Stokes equation?

Example 1 PDE-Nets

Navier Stokes, Maxwell, Schrodinger…

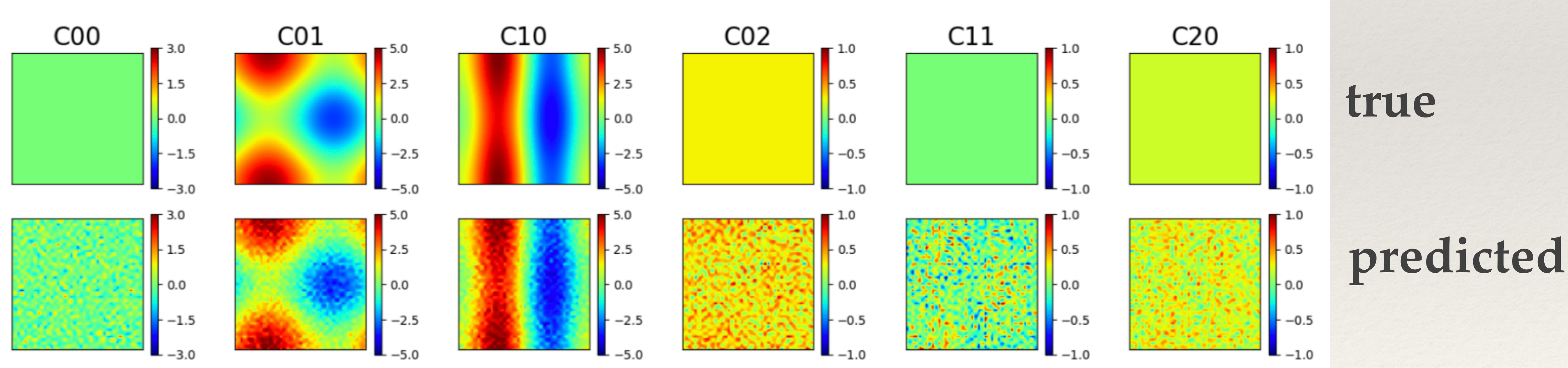
Laws of Nature are PDEs

But there are other areas in physics where equations are not known and even when we know them, they include assumptions

Pose an inverse problem: given an observed temporal distribution, can a NN learn the non-linear response?

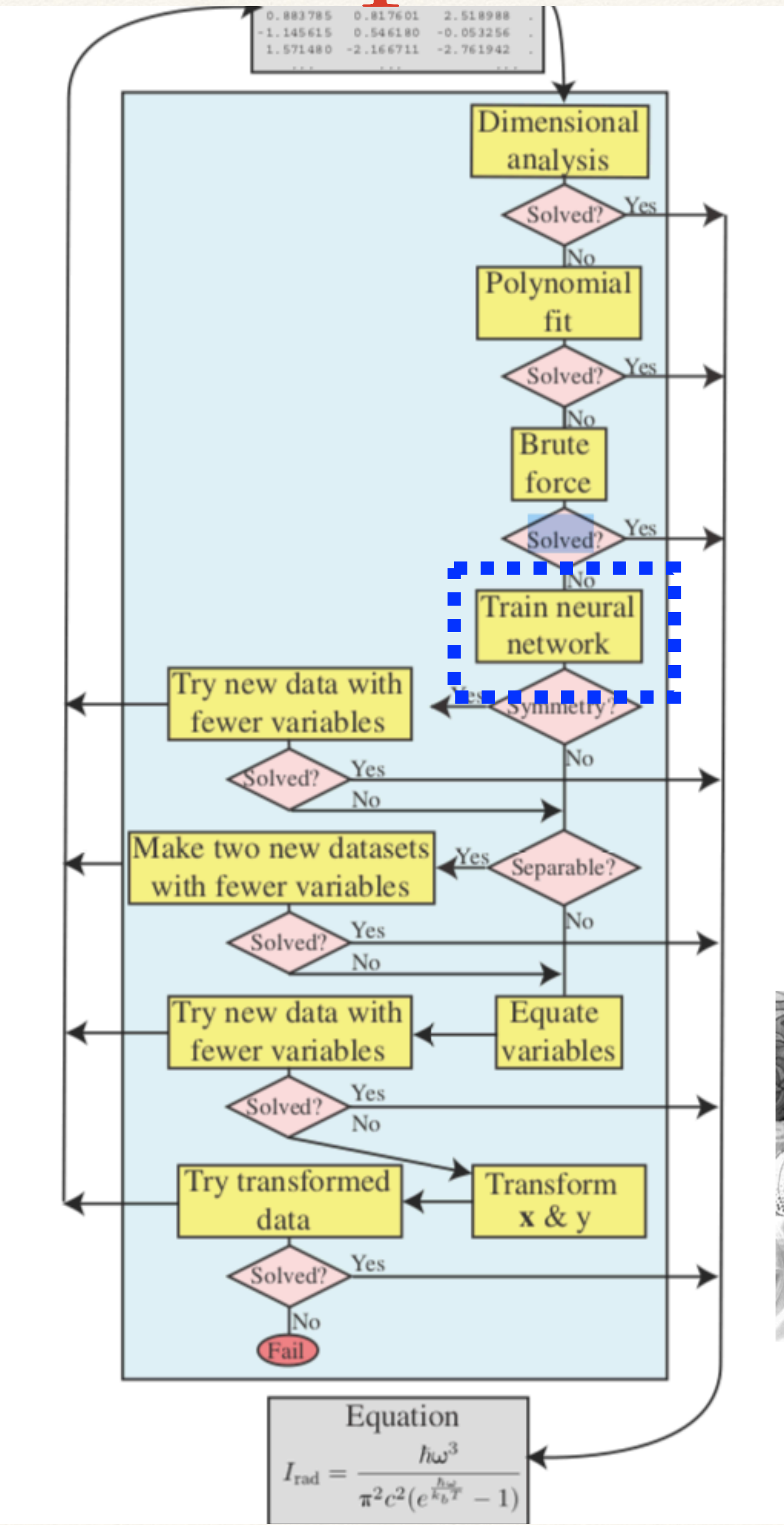
$$u_t = F(x, u, \nabla u, \nabla^2 u, \ldots), \quad x \in \Omega \subset \mathbb{R}^2, \quad t \in [0, T].$$

$$u_t(t, x, y) = F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, \ldots), \quad (x, y) \in \Omega \subset \mathbb{R}^2, t \in [0, T].$$



paper here, v2.0 here and code here, and a simpler example Neural Diff Eqs

Example 2: AI Feynman



Given a dataset (X, y) find the function y=f(X)

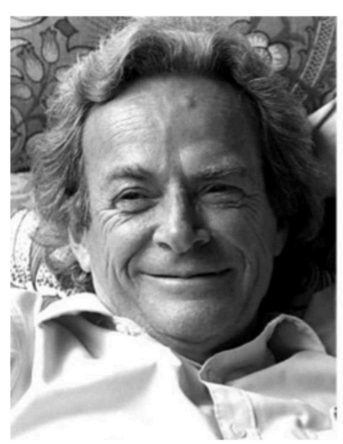
Clearly a task for a NN

The authors (physicists) use a set of tricks to speed up things (symmetries/dimensional analysis etc)

Then they test this procedure against hundreds of physics equations in Feynman lectures and compare with a commercial tool

It's quite fun, and they pose this as a mystery hunt

they do not handle (yet) PDEs

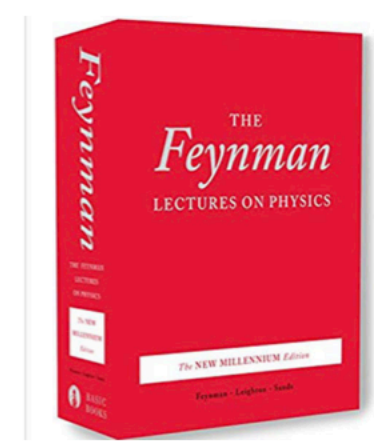


$$L = \frac{\hbar\omega^3}{\pi^2c^2(e^{\hbar\omega/k_bT}-1)}$$

$$F = \frac{Gm_1m_2}{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$$

$$r = \frac{a(1-e^2)}{1+e\cos(\theta_1-\theta_2)}$$

$$\frac{d\sigma}{d\cos\theta} = \frac{\pi\alpha^2\hbar^2}{m^2c^2}\left(\frac{\omega'}{\omega}\right)^2\left(\frac{\omega'}{\omega}+\frac{\omega}{\omega'}-\sin^2\theta\right)$$



This is the article and here the code and datasets

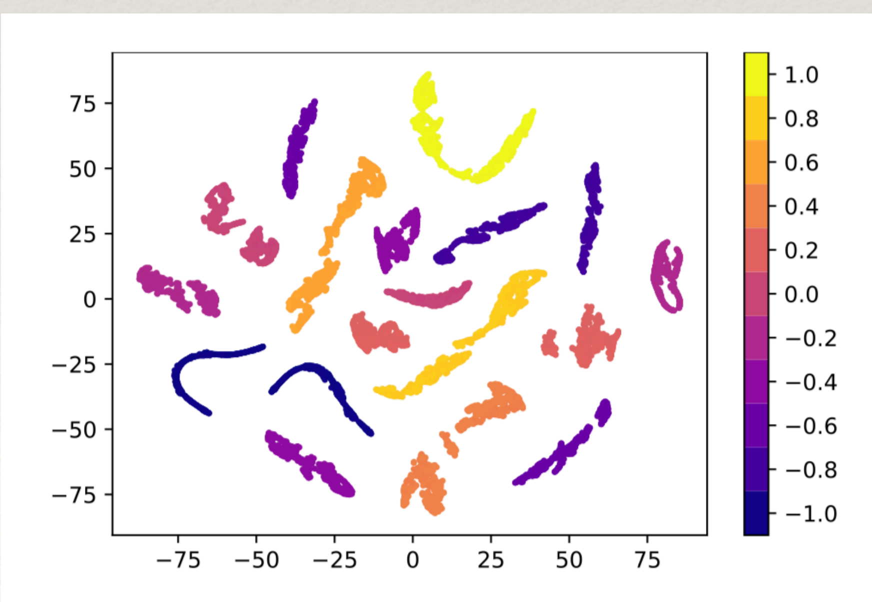
Example 3: Detecting symmetries

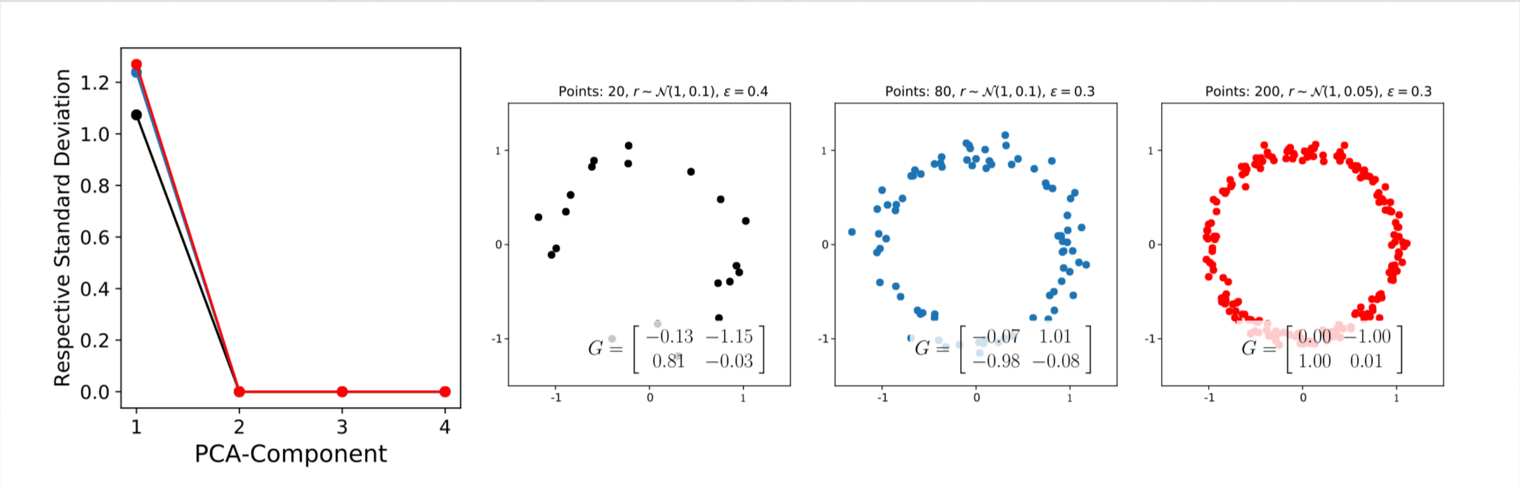
By looking at a dataset, can be identify there is some symmetry?

$$x_{\text{input}} \to S(x_{\text{input}}) \qquad f(S(x_{\text{input}})) = f(x_{\text{input}}).$$

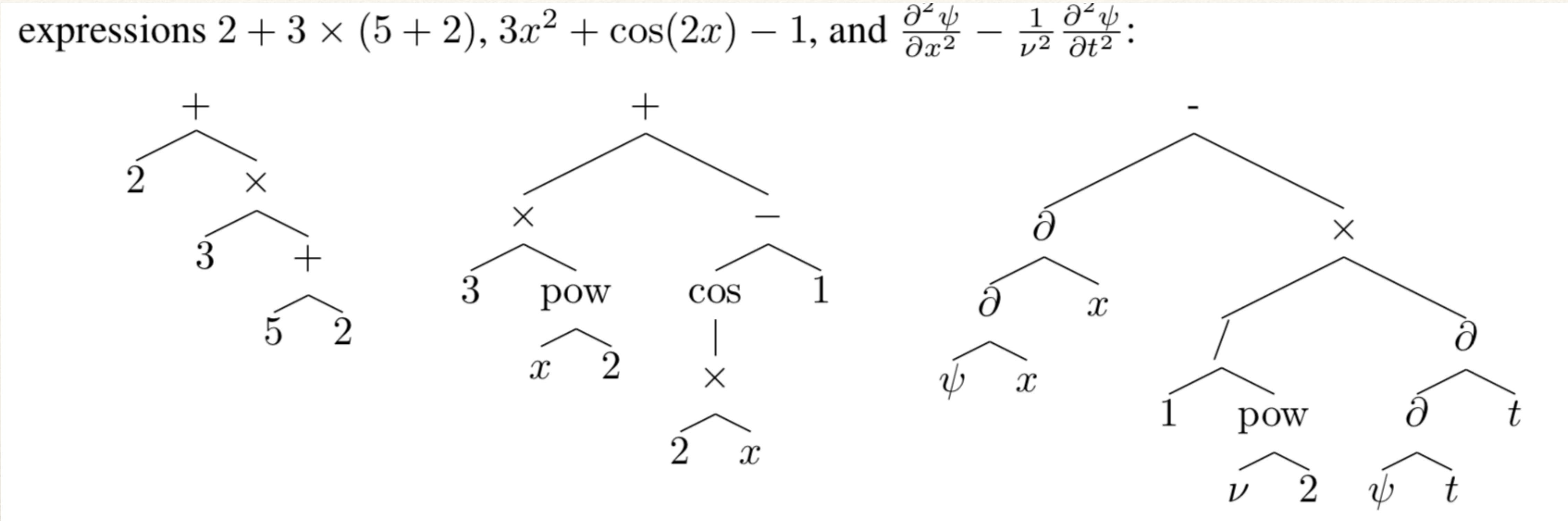
If there's a symmetry, the NN may 'realise' it
when doing transformations in the hidden layers
maybe by doing a PCA and clustering analysis in one of the last hidden layers
we see some symmetry structure

In this paper, the authors run a NN and look at the content
one the last hidden layers
do a TSNE analysis to learn to identify discrete and continuous symmetries





Example 4: Symbolic mathematics



$x^2\left(\tan^2(x)+1\right)+2x\tan(x)+1$	$x^2\tan(x)+x$
$1+\frac{2\cos(2x)}{\sqrt{\sin^2(2x)+1}}$	$x+\operatorname{asinh}(\sin(2x))$
$\frac{x\tan(x)+\log(x\cos(x))-1}{\log(x\cos(x))^2}$	$\frac{x}{\log(x\cos(x))}$
$-\frac{2x\cos(\operatorname{asin}^2(x))\operatorname{asin}(x)}{\sqrt{1-x^2}\sin^2(\operatorname{asin}^2(x))}+\frac{1}{\sin(\operatorname{asin}^2(x))}$	$\frac{x}{\sin(\operatorname{asin}^2(x))}$
$\sqrt{x}+x\left(\frac{2x}{\sqrt{x^4+1}}+1+\frac{1}{2\sqrt{x}}\right)+x+\operatorname{asinh}(x^2)$	$x\left(\sqrt{x}+x+\operatorname{asinh}(x^2)\right)$
$\frac{-3-\frac{3\left(-3x^2\sin(x^3)+\frac{1}{2\sqrt{x}}\right)}{\sqrt{x}+\cos(x^3)}}{\left(x+\log(\sqrt{x}+\cos(x^3))\right)^2}$	$\frac{3}{x+\log(\sqrt{x}+\cos(x^3))}$
$\frac{-2\tan^2(\log(\log(x)))-2}{\log(x)\tan^2(\log(\log(x)))}+\frac{2}{\tan(\log(\log(x)))}$	$\frac{2x}{\tan(\log(\log(x)))}$

transform mathematical expressions into trees of relations
Run a NLP algorithm to learn to symbolically solve integrals/ differential equations etc
Benchmark against Mathematica/Matlab etc and obtain similar or better results

paper here and code here

Today

In this notebook you can find some brief examples of transfer learning for MNIST and for a PP example
Use PYTORCH and FASTAI, syntax is compact
PP example: LHC Olympics 2020
images of boosted jets produced by SM and by a new heavy particle
small dataset, use augmentation

If you got time, check-out some of the newest applications, like *Feynman AI*
or Transfer Learning for music

Tomorrow **XAI**