Introduction to Deep Learning

Examples from High Energy Physics

Sofia Vallecorsa

26/07/2021

Intelligence .. or the hability to:

- Learn from experience
- Extract semantics
- Model
- Generalize
- Abstraction
- Meta-learning

Outline

- Motivation: Deep Learning in High Energy Physics
- Introduction & Basic Concepts
- Example architectures and applications in HEP
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Graph Neural Networks
  - Generative Models

Big Data at the LHC

Experiments (detectors & physics data)

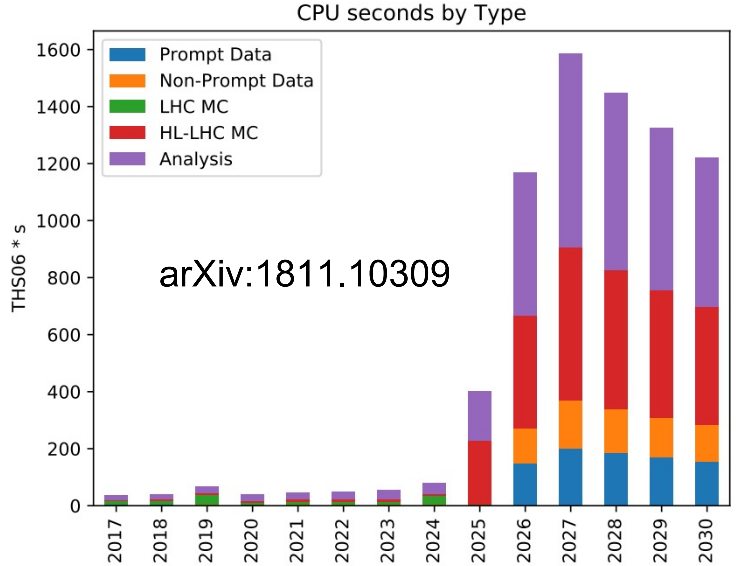
- **330 PB of collisions data** stored by end 2018

Accelerators infrastructure

- 9600 magnets for beam control
- 1232 superconducting dipoles for bending

Computing infrastructure

LHC data is **multi-structured, hybrid**



Next generation colliders will require **larger, highly granular detectors** that will generate **huge particle data rates O(100 TB/s)**

CERN openlab

Deep Learning in HEP

DL can **recognize patterns** in large complicated data sets

- Better performances if applied directly to **raw** data

Re-cast physics problems as “DL problems”

- Interpret detector output as **images** and apply techniques borrowed from **computer vision**
- Interpret physics events as **sentences** and apply **NLP techniques**

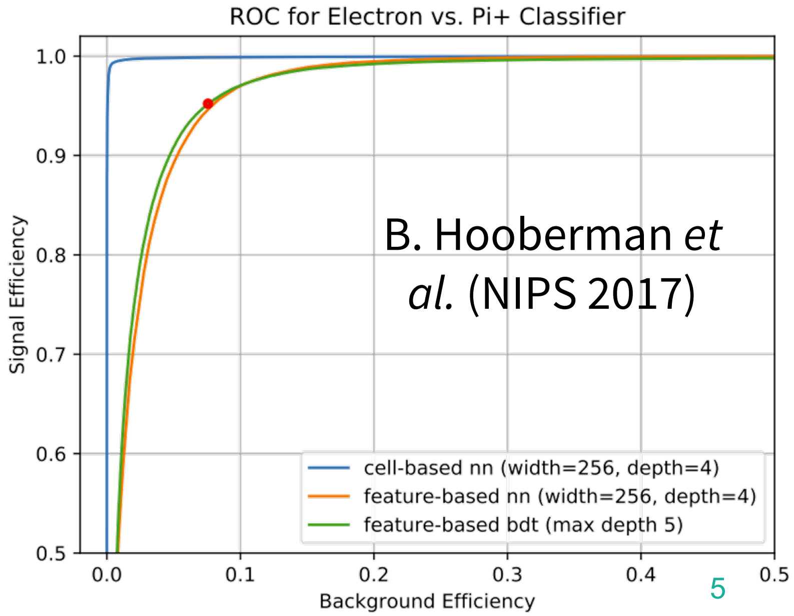
Intense R&D activity

Adapt DL to HEP requirements

- In terms of model **interpretability**
- Results **validation** against classical methods
- Detailed **systematics**

Adopting ”new” computing models

- **Accelerators** and dedicated hardware
- **HPC** integration
- **Cloud** resources
- **Big Data** platforms

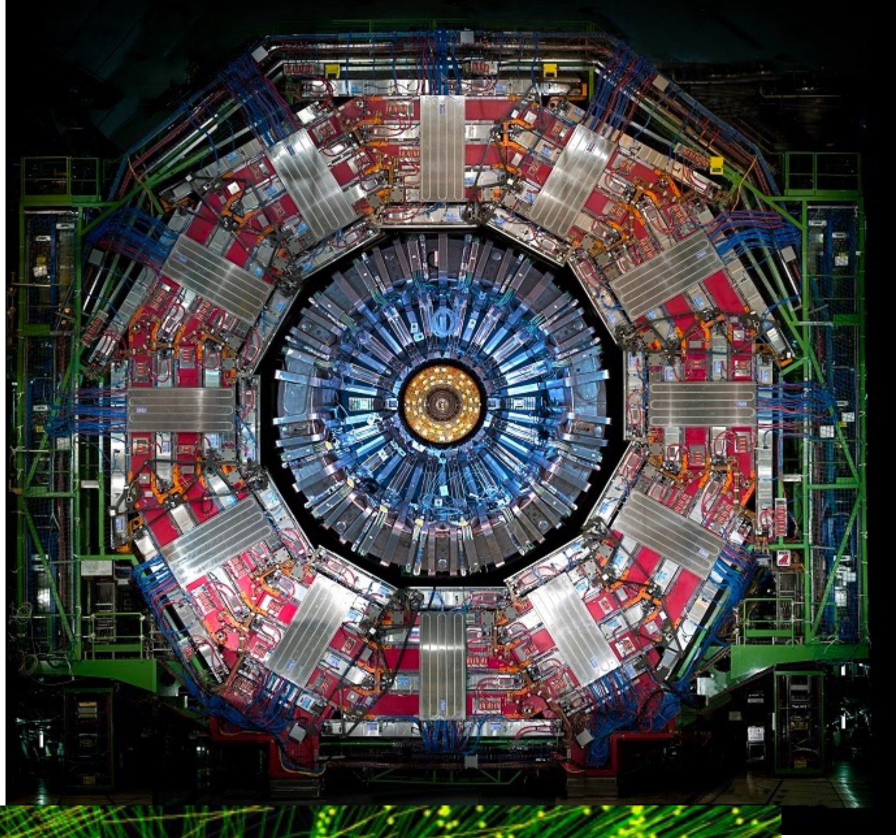


B. Hooberman *et al.* (NIPS 2017)

Applications in HEP (II)

Classical Machine Learning has been used for many years, mostly during the final steps of data analysis for signal /background separation

Deep Learning is studied for many different applications



Real-time filtering

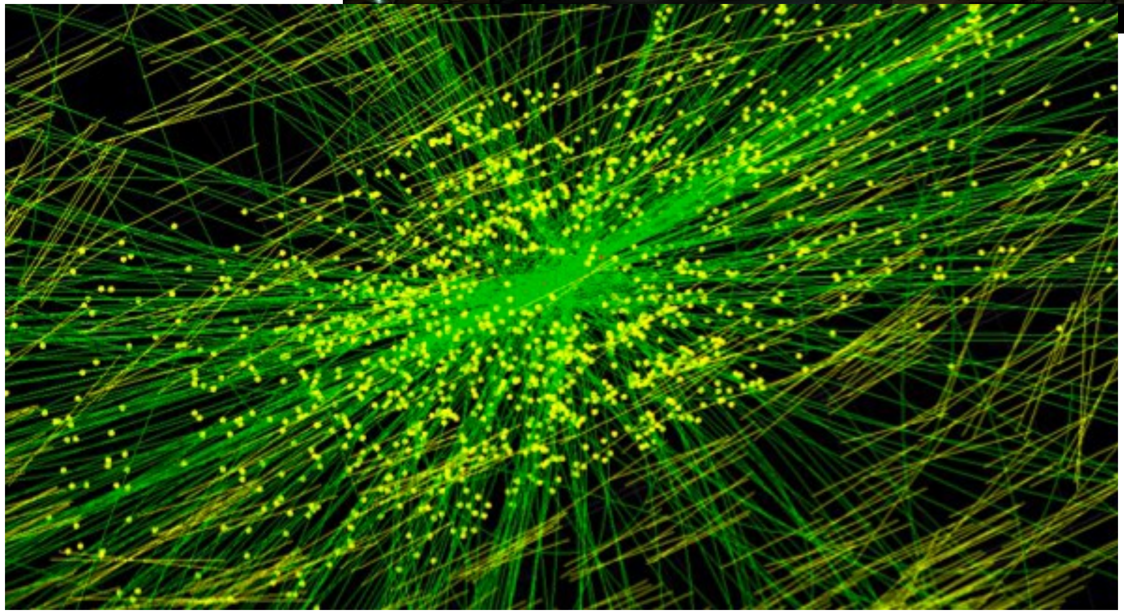
Raw data processing

Monitoring and Control Systems

Analysis

Optimisation

Simulation



CERN openlab

Universal approximator

NN with a single hidden layer containing a finite number of non-linear neurons approximate continuous functions to any desired degree of accuracy.

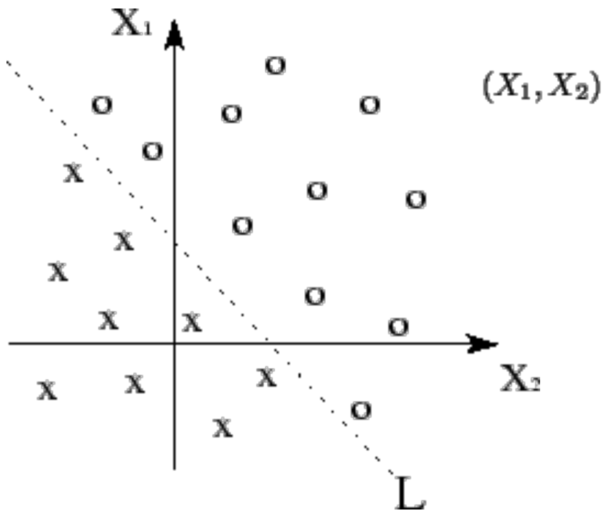
Hornik, Kurt; Tinchcombe, Maxwell; White, Halbert (1989). *Neural Networks*. **2**. Pergamon Press. pp. 359–366.

The need for depth

A single layer perceptron can categorize "linearly separable" patterns

Two classes classification:

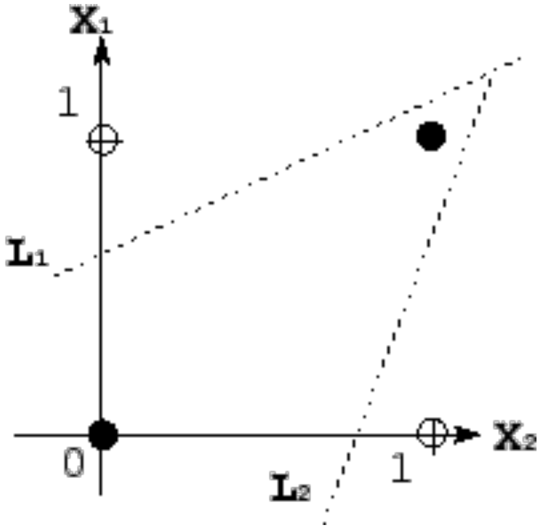
(OR function) (linearly separable)



Exclusive OR is an example of a non linearly separable pattern:

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

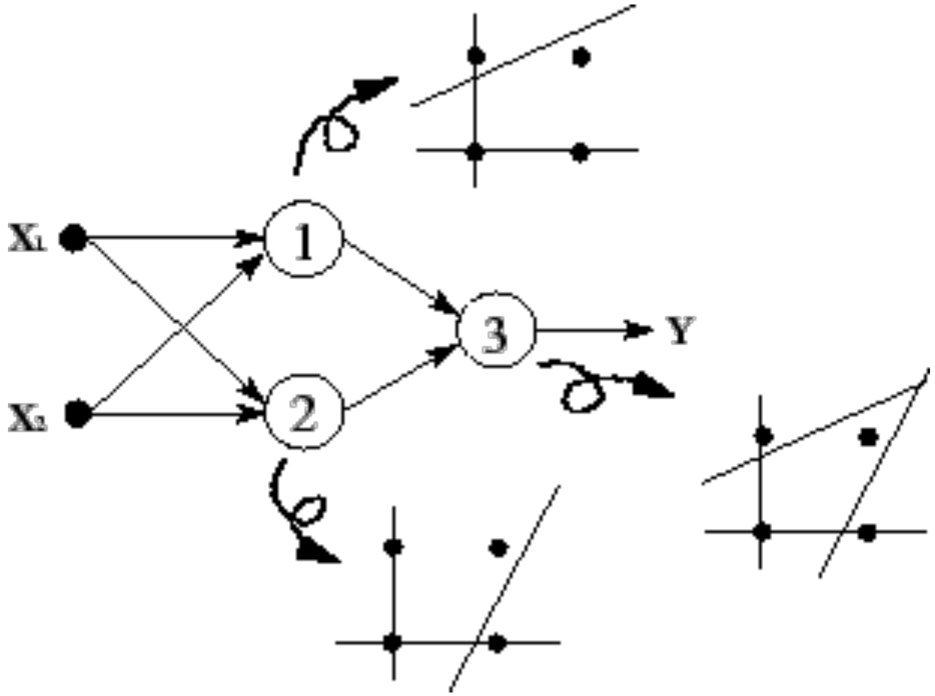
$$Y = X_1 \oplus X_2$$



The need for depth (II)

Need a Multi-Layer architecture to solve the exclusive OR problem:

Two-stages approach



Deep Neural Networks

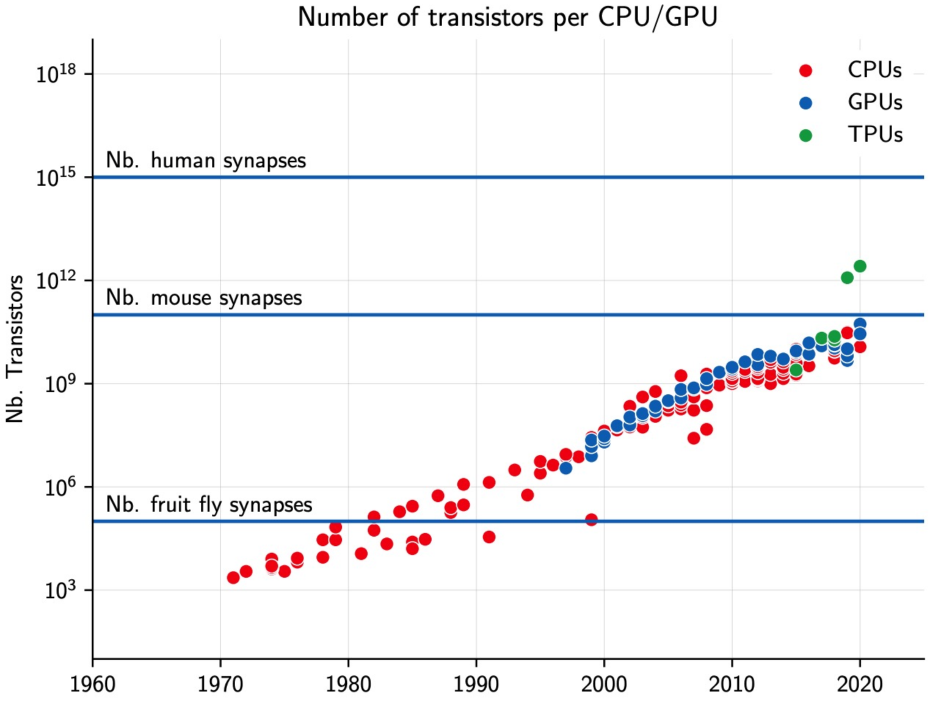
"Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

…

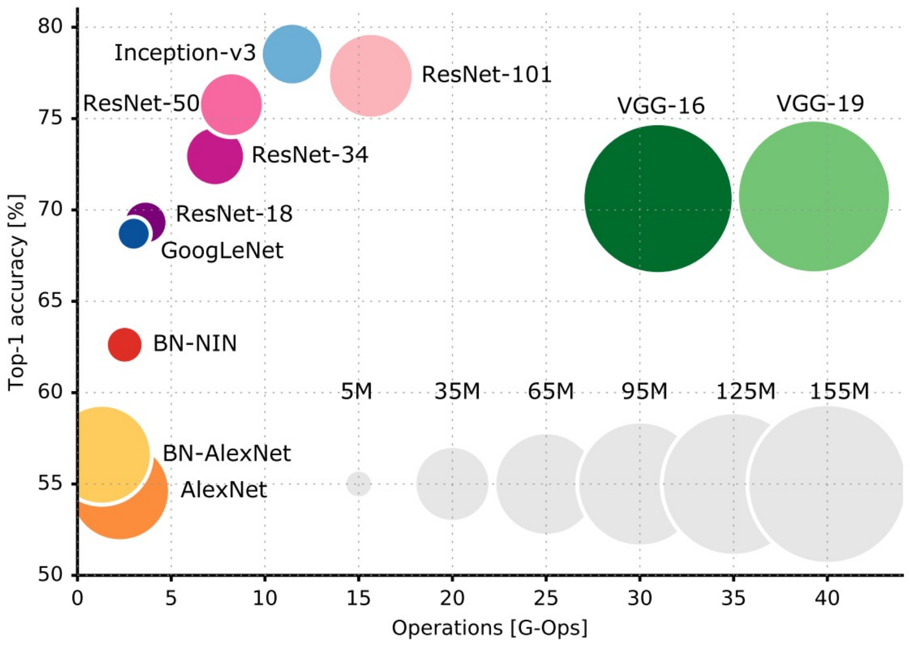
Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer…"

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).

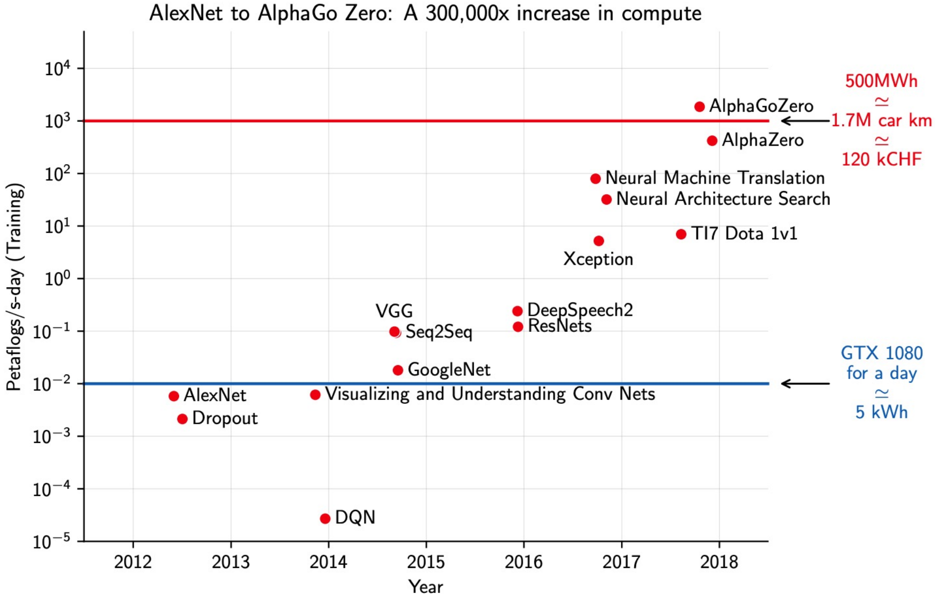
Increasing sizes



(Wikipedia "Transistor count")



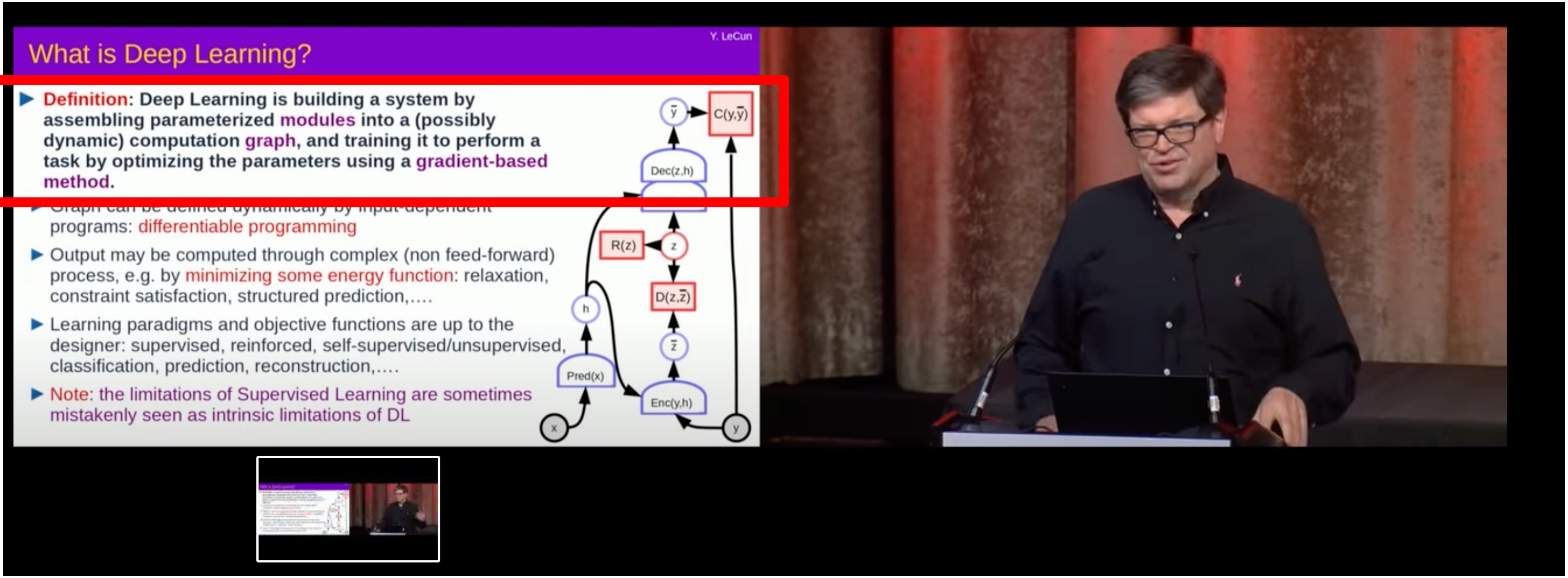
(Canziani et al., 2016)



(Radford, 2018)

Fleuret, Deep Learning Course: https://fleuret.org/dlc

More than just a deeper NN



AAAI 20 keynotes Turing Award Winners (Geoff Hinton Yann Le Cunn, Yoshua Bengio):
https://www.youtube.com/watch?v=UX8OubxsY8w

openAI GTP-3

Generative Pretrained Transformer-style autoregressive model

- **175 billion parameters**
- Previously largest model was **Microsoft's Turing NLG**, with 17 billion parameters (Feb. 2020)

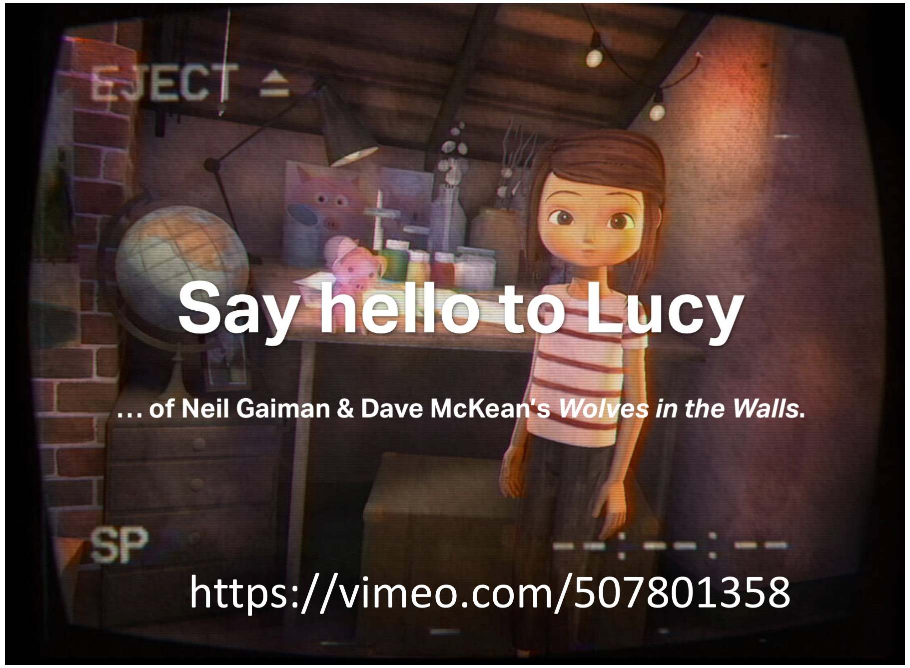
A **generative model**: learns a probabilty distribution from a data set and generate a new set belonging to the same distribution

- Create **realistic texts**
- Can do other tasks (translation, question-answering, etc..)

Trained with large Internet data sets (bias?)

https://arxiv.org/pdf/2005.14165.pdf

Text fragments:
https://arr.am/2020/07/09/gpt-3-an-ai-thats-eerily-good-at-writing-almost-anything/

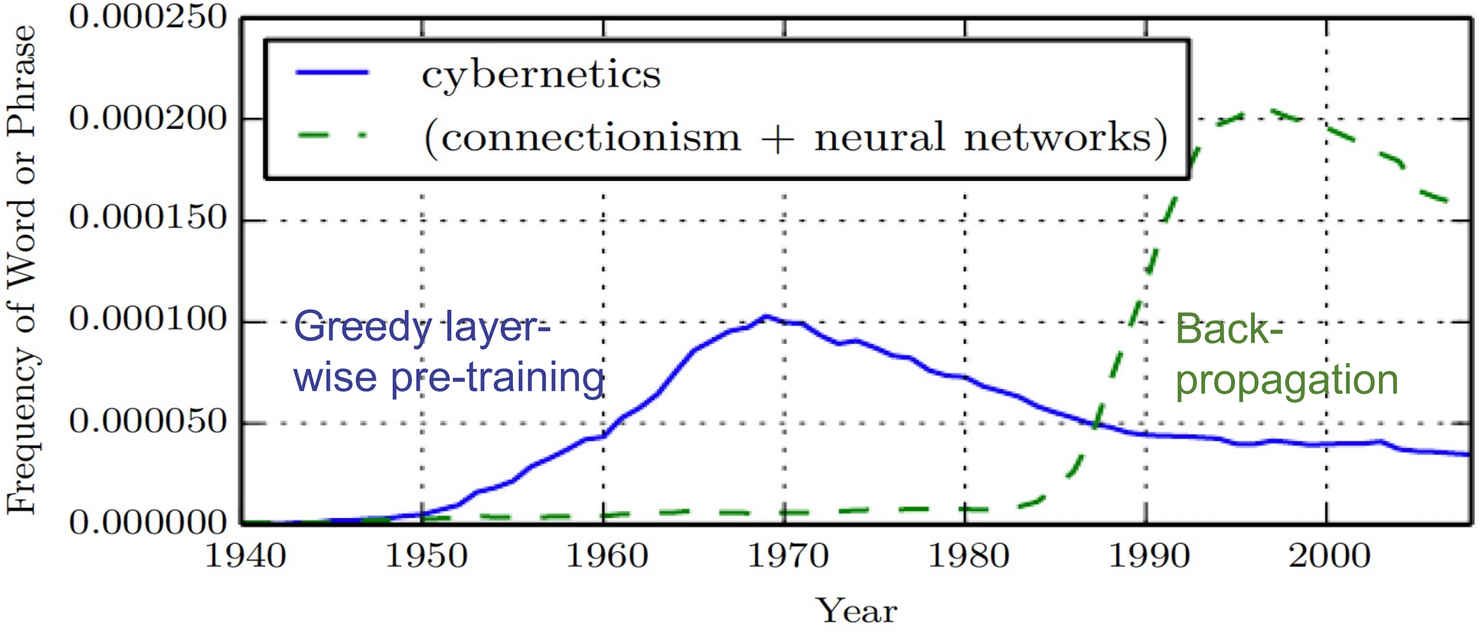


CERN openlab

How do we train DL

- Algorithms improvements
  - Back-propagation, Auto Differentiation
- Large amount of data (labelled data for supervised learning)
- Computing power
  - Highly parallel hardware
  - Dedicated accelerators (GPUs, Google TPUs, AWS INF1, Graphcore..)
  - Cloud and HPC resources

Image from "Deep Learning", I. GoodFellow, Y. Bengio, A. Courville , MIT press book



Different approaches to training:

Unsupervised pre-training

Transfer learning and fine tuning

Few-shot learning

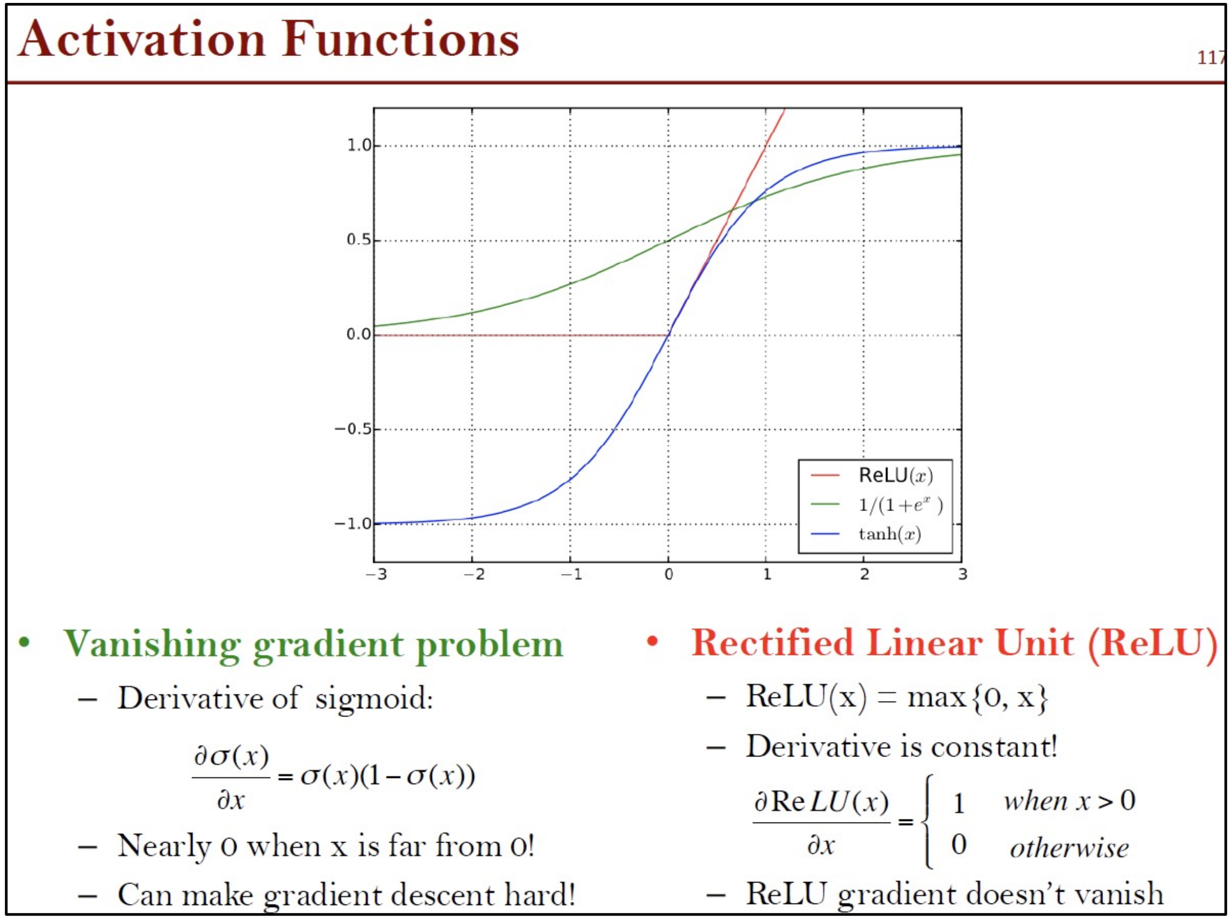
Meta-learning

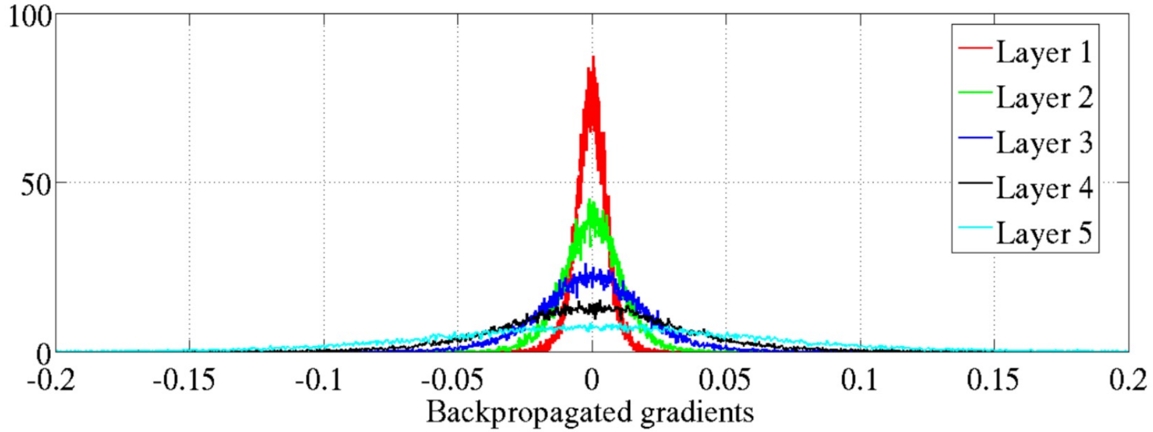
Vanishing gradients

Small gradients slow down stochastic gradient descent.

- Limits ability to learn

Gradients for layers far from the output vanish to zero.



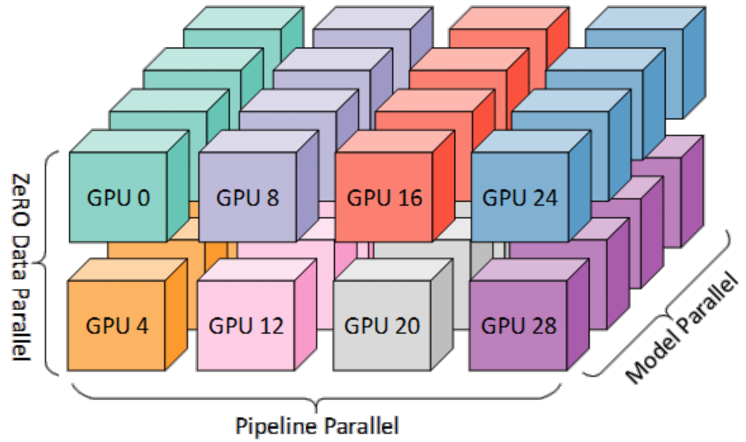


Backpropagated gradients normalized histograms (Glorot and Bengio, 2010).

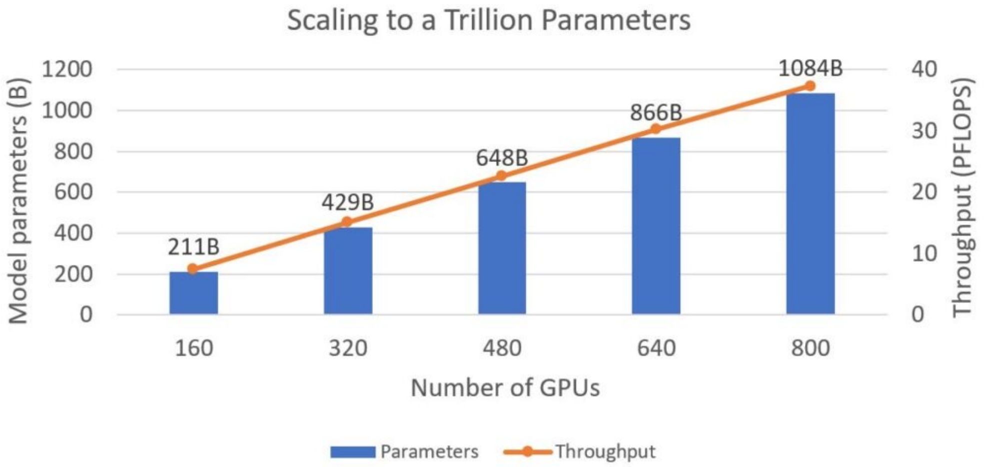
Accelerating the training process

Introducing techniques to **parallelise** training

- **Data parallelism**
  - Compute gradients on several batches independently
  - Update the model synchronously or asynchronously
- **Model Parallelism**, **Hybrid techniques**
- Use **reduced precision** representation (INT6, BF16, …)
- Extreme parallelism using **combined strategies** and SGD algorithm optimisation
  - DeepSpeed and ZeRO-2 on Microsoft Azure



https://www.microsoft.com/en-us/research/blog/deepspeed-extreme-scale-model-training-for-everyone/



Fugako System @RIKEN, Japan

158 000 nodes:

48-core Arm 8.2-A

Computing resources

Summit @Oak Ridge, USA

4,356 nodes:

2x 22-core IBM Power9 CPUs

6 NVIDIA Tesla V100 GPUs.

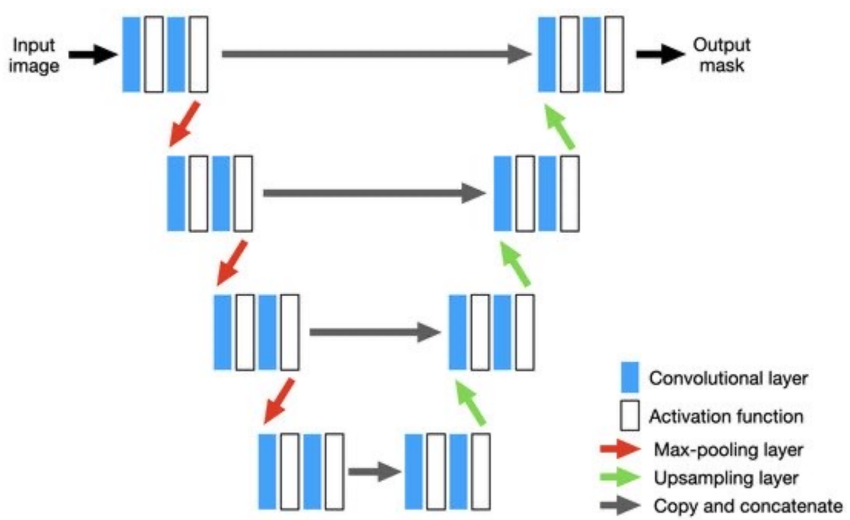
Transfer learning, pre-training, fine-tuning

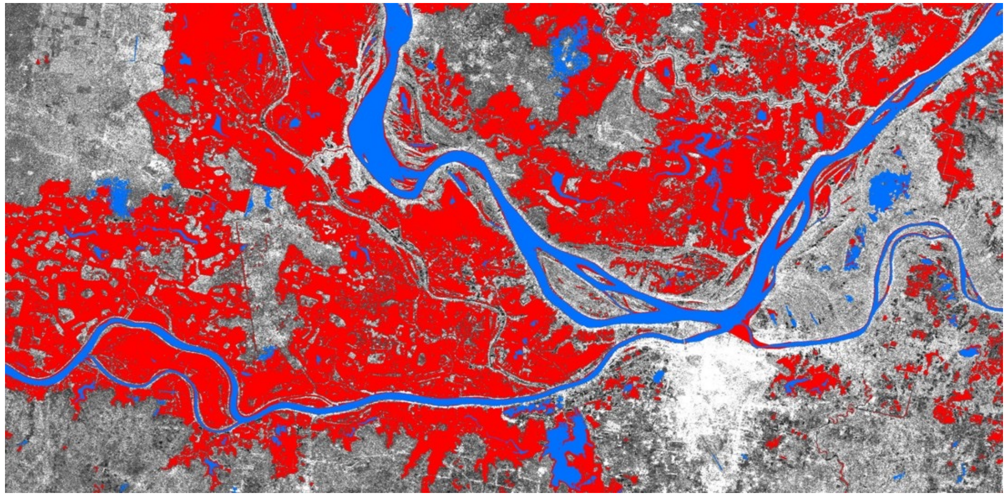
*“**Transfer learning** and domain adaptation refer to the situation where what has been learned in one setting … is exploited to improve generalization in another setting”*

Deep Learning, 2016.

- Transferring learned knowledge to similar task
- How much of the pre-trained model to use in new one?
  - CNN features are more generic in early layers and more dataset-specific in later layers
- Can be used to train large models
- Ex. **Extraction of flood water extent from satellite images using U-Net**

U-Net





Nemni, Edoardo, et al., *Remote Sensing* 12.16 (2020): 2532.

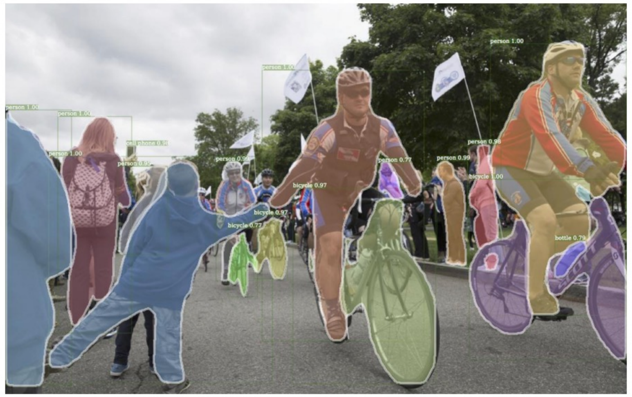
CERN openlab

Counting shelters in refugee camps

2019 CERN openlab Summer Students Program

CERN openlab and UNOSAT collaboration

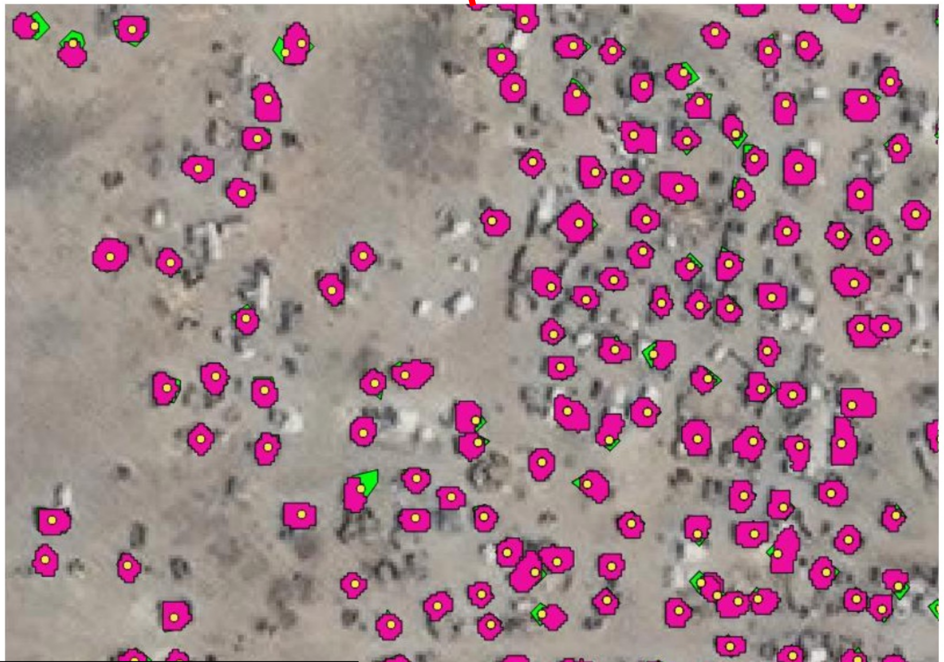
(UN Operational Satellite Applications Centre)

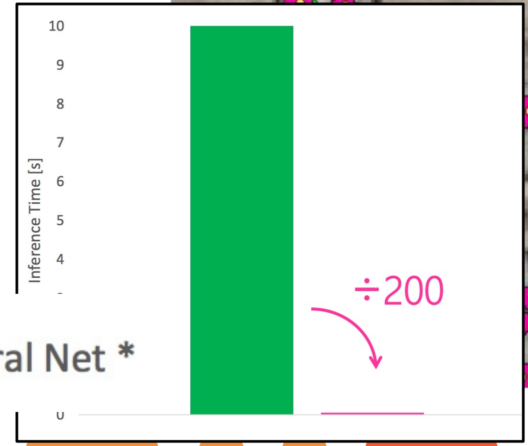


Detectron Framework (FacebookAI)

Retrain & encode point data cleverly

Unosat Adapted model





Human Neural Net *

Transfer learning from RCNN model
Average precision is 82%
Speedup is x200 wrt (human) expert processing

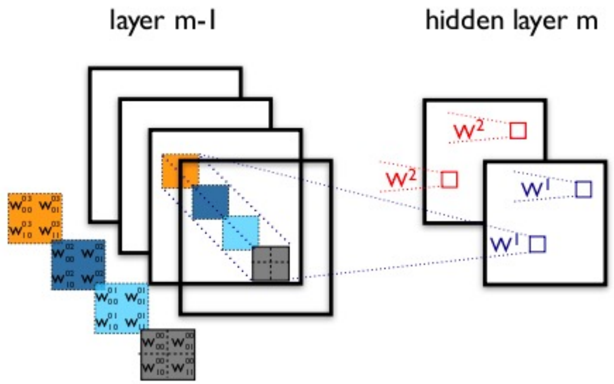
CERN openlab

https://indico.cern.ch/event/727274/contributions/3100369/

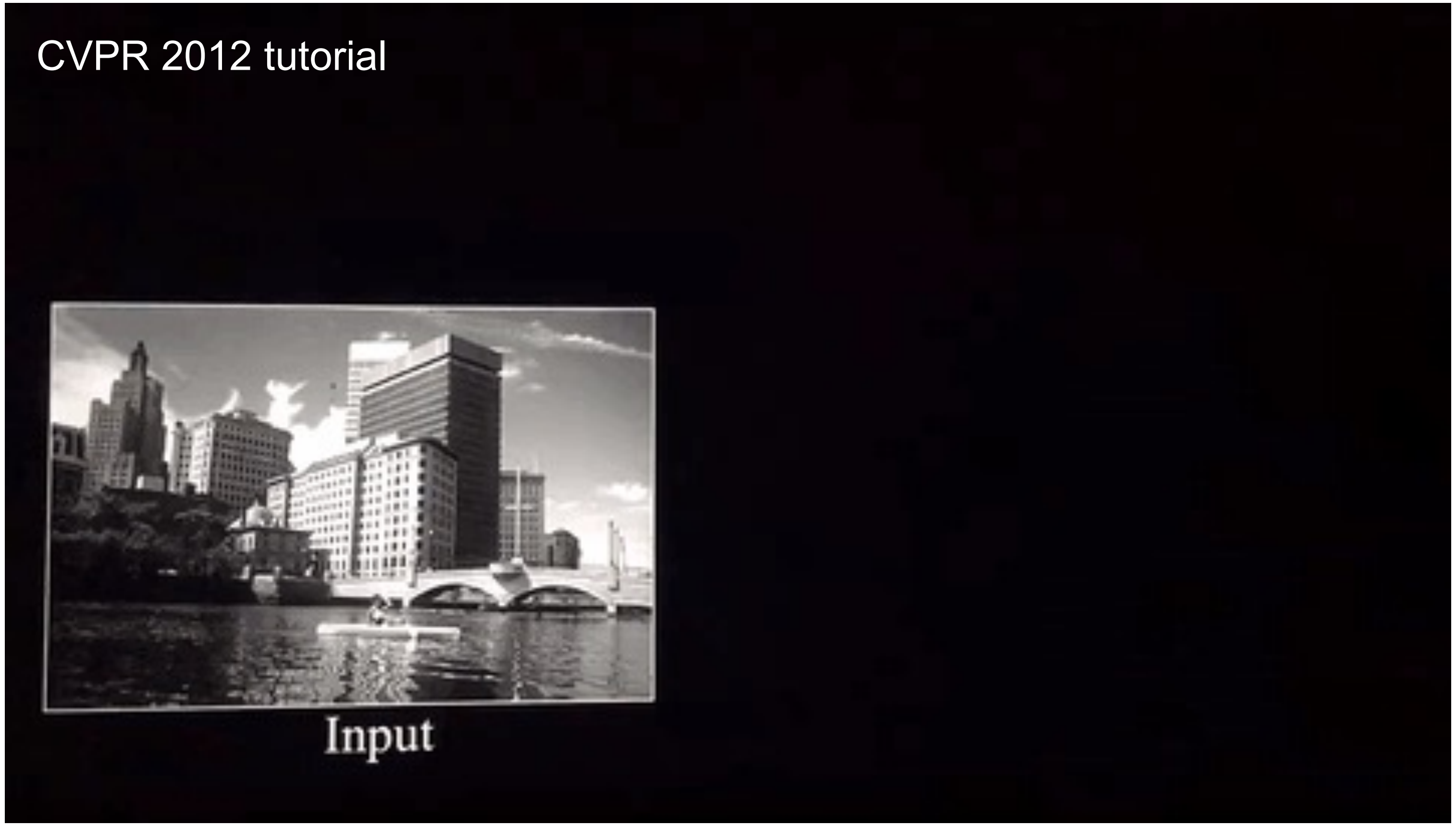
Example Architectures

CERN openlab

Convolutional Neural Networks



- Exploit **spatially-local correlation**
  - Enforce local connectivity pattern between neurons of adjacent layers
- **Increasing level of abstraction**
  - Initial layers learn simple features (edges and color gradients)
- Output dense layers combine **high level features** and produce predictions.



CERN openlab

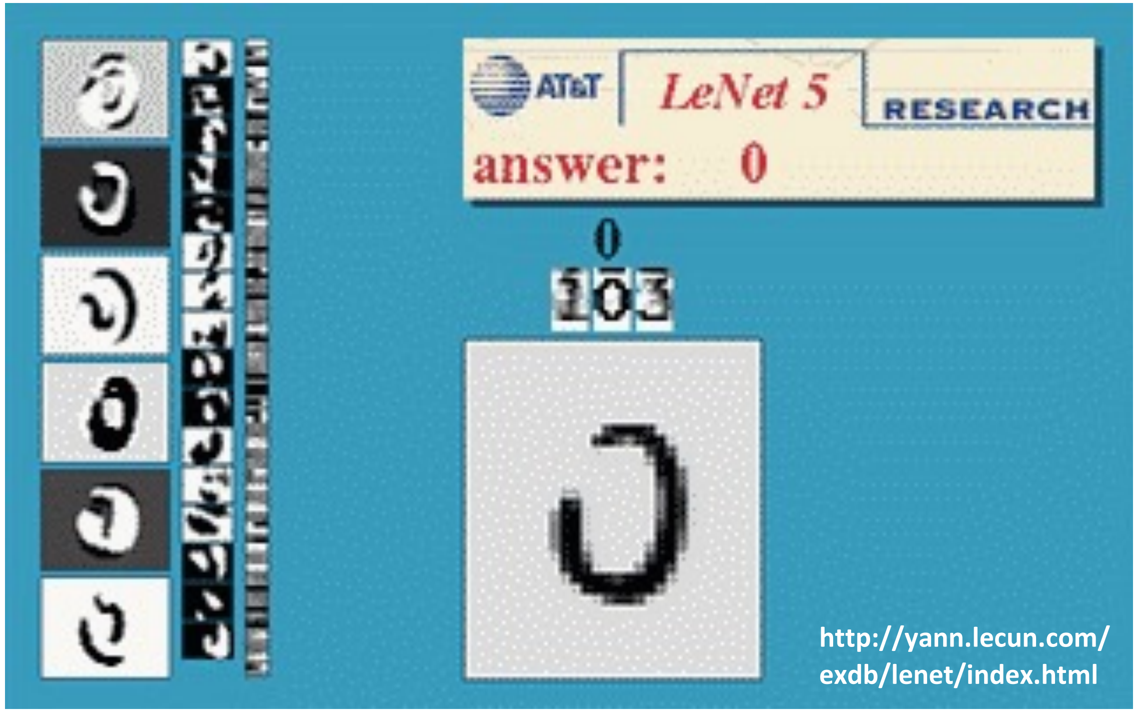
LeNet

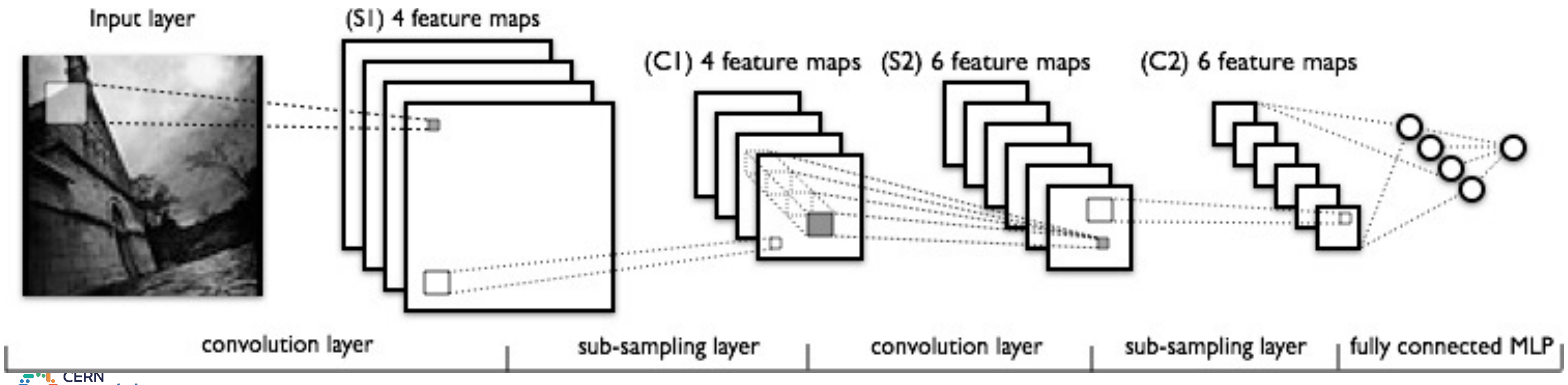
7-layers CNN to recognise hand-written numbers on checks

- digitized in 32x32 pixel greyscale input images.

to process higher resolution images need larger and more convolutional layers

- availability of computing resources!



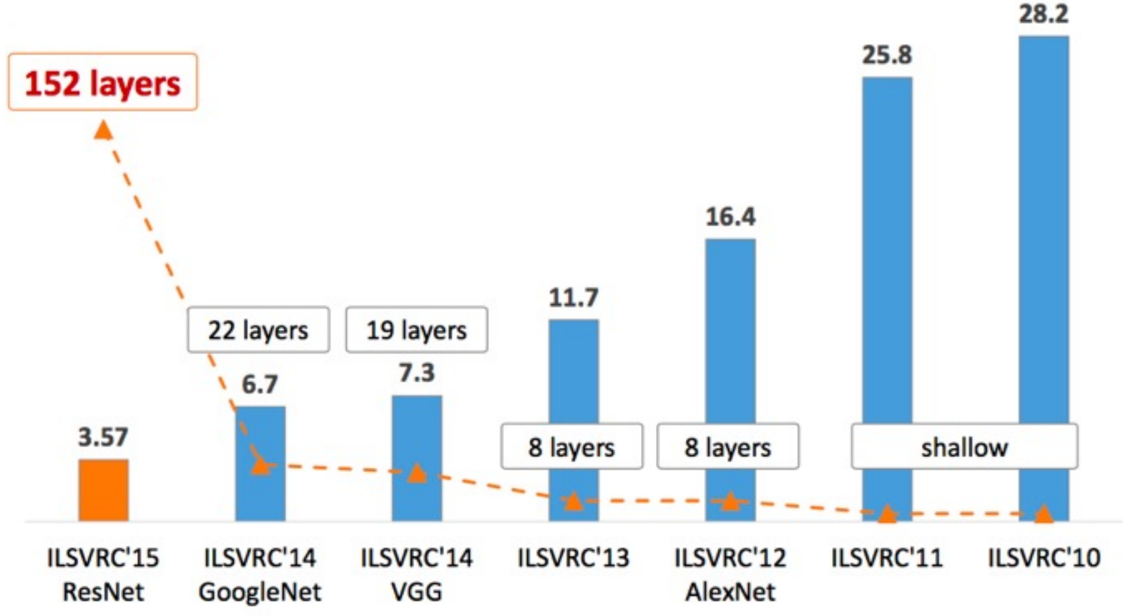


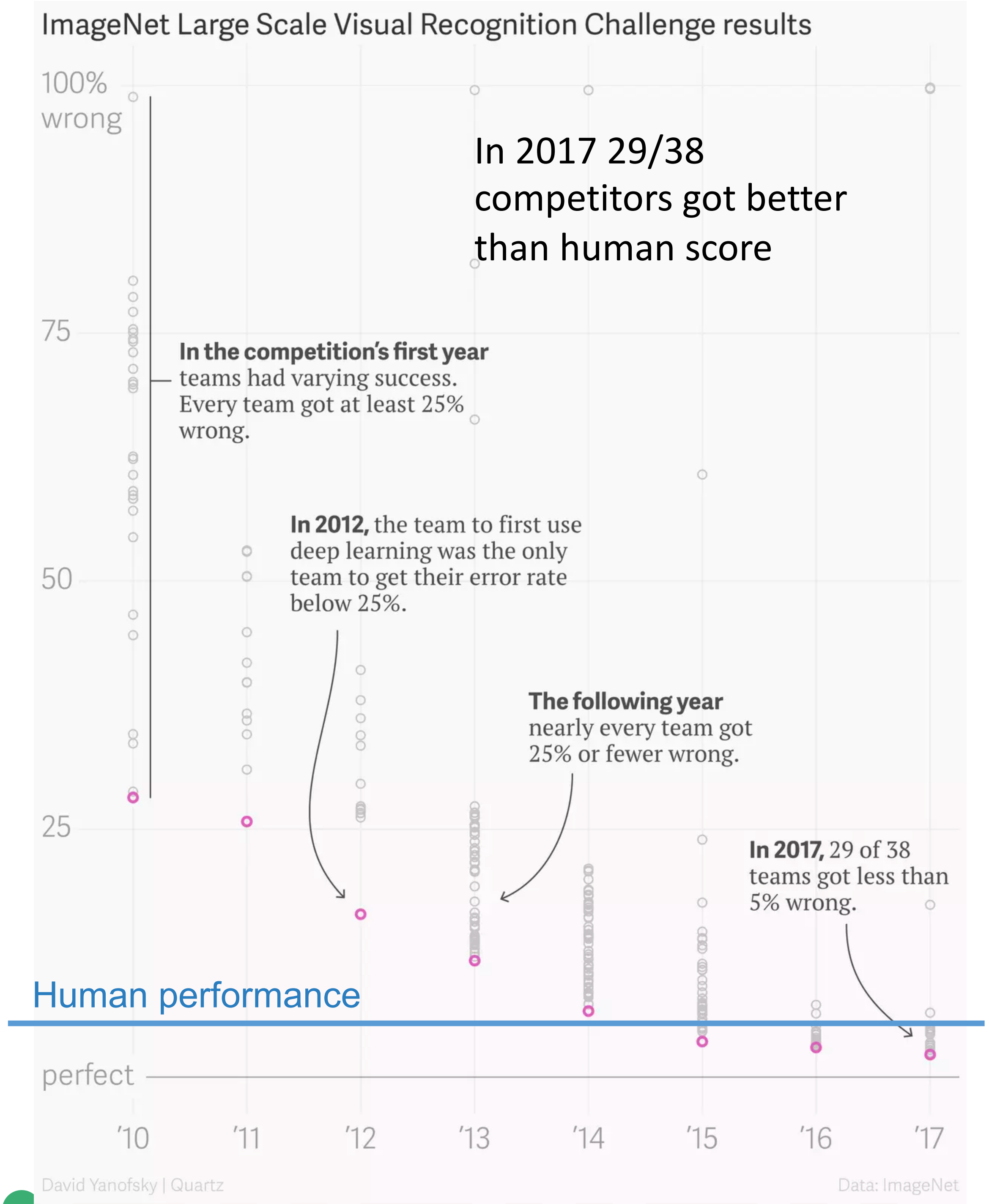
LeCun et al., 1998

ILVRC challenge

Imagenet: >14 M images with 20k classes

ImageNet Large Scale Visual Recognition Challenge started in 2010 with 100 classes (1000 classes in 2017)





CNN applications

arxiv:1712.04837

Multiple tasks:

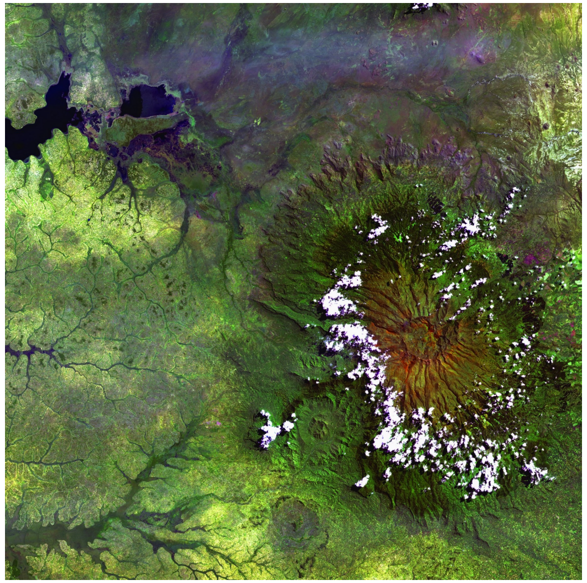
Image analysis, segmentation

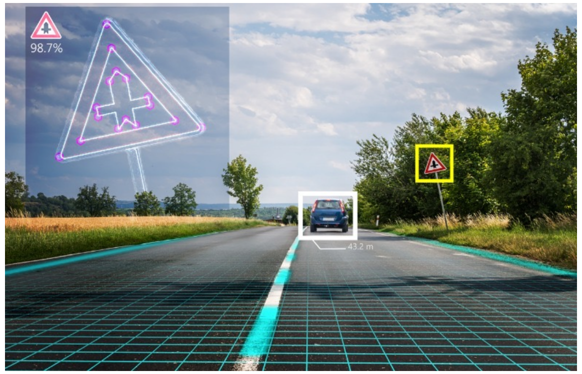
Object detection and pattern recognition

..

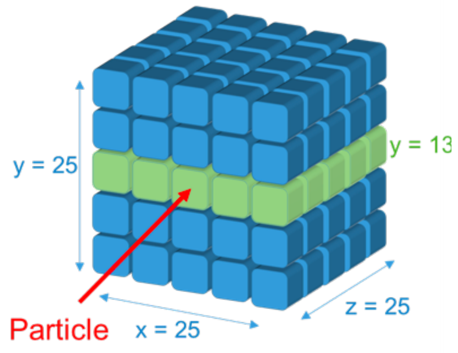
Different fields:

Science, medicine, Earth Observation





GANs for detector simulation



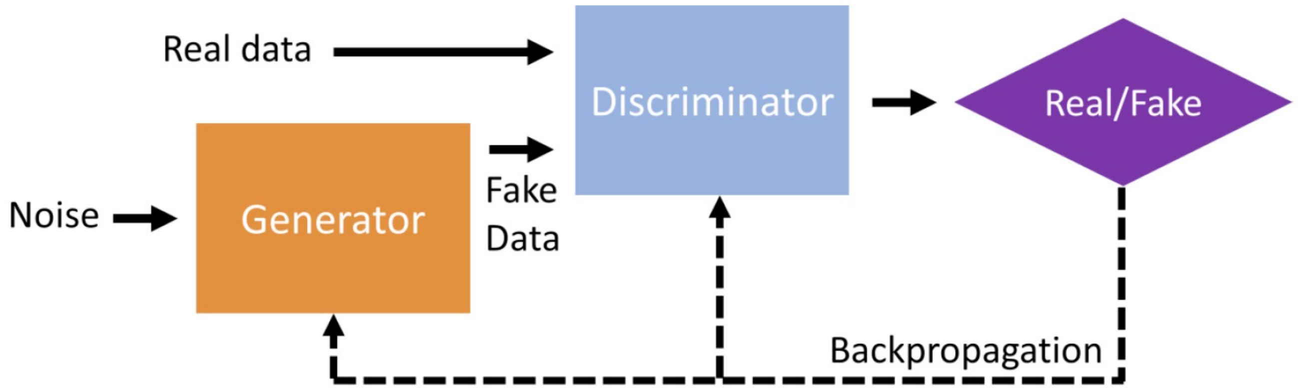
Monte Carlo simulation is extremely **demanding in terms of computing resources**

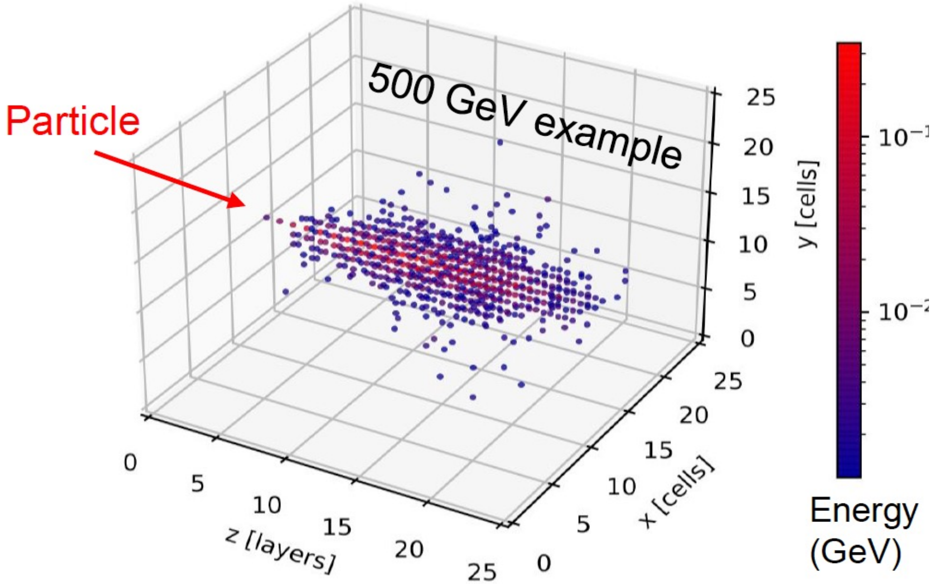
Train a **Deep Generative Model** instead

Detector output as images: read-out channels become pixels

Train a Generative Adversarial Network: a **pair of networks** in a **min-max** game

Pixelized 3D image





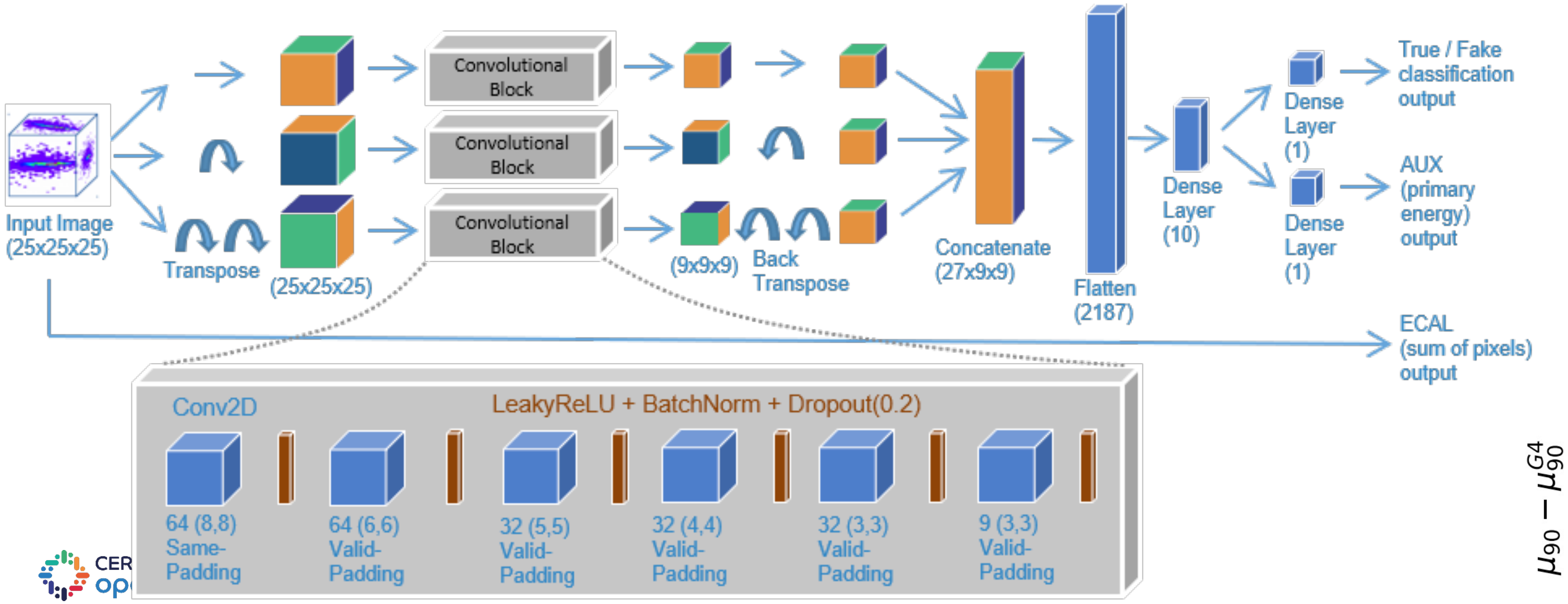
The 3DGAN prototype

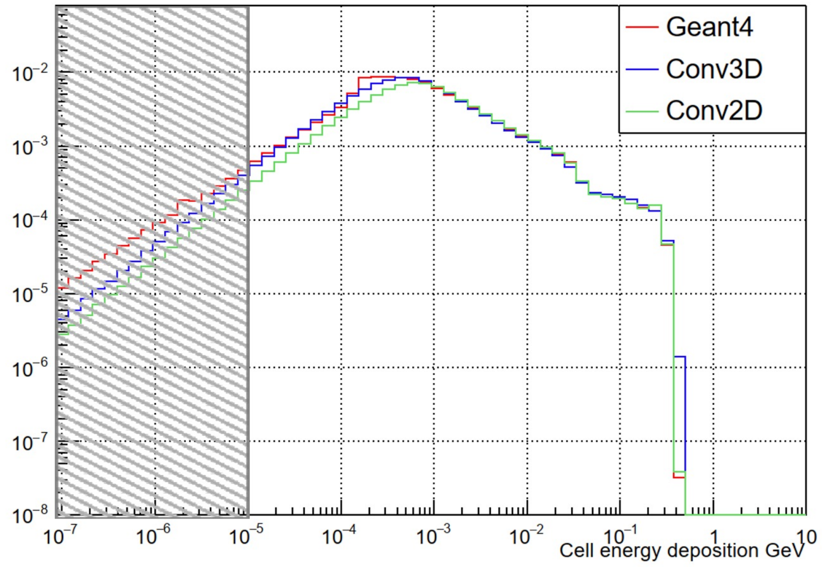
3D convolutional layers are computationally expensive

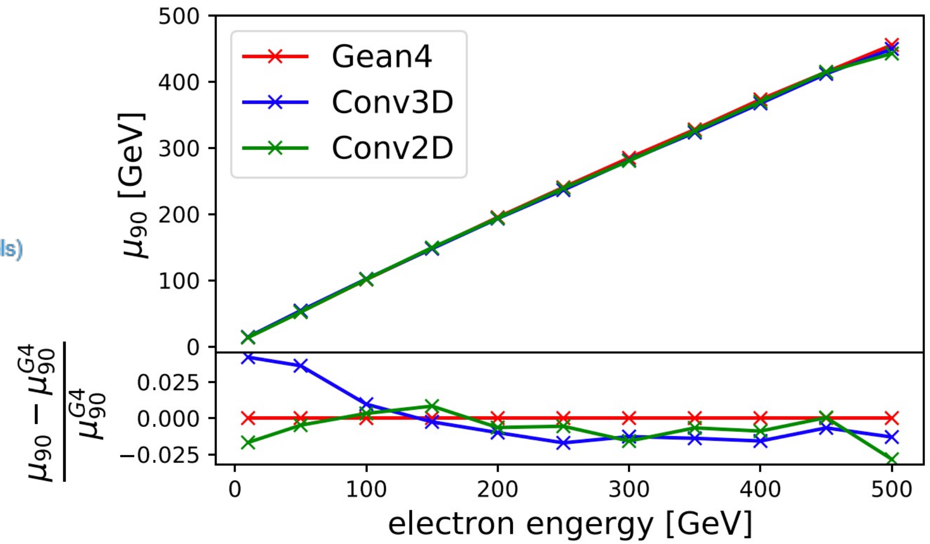
Reproduce a 3D volume using 2D convolutions

2.1x speed-up while maintaining accuracy

Generator: 2.15M parameters







arxiv: 1312.6199

CNN shortcomings

Spatial features:

Humans recognize objects under different view angles, scales or lighting conditions

CNNs can handle translations but none of the above

Adversarial examples:

Minimal changes in the image can cause entirely different outcome

A proposed solution: construct a hierarchical representation based on **instantiations of specific types of entities**

match it with already learned patterns and relationships stored in the brain

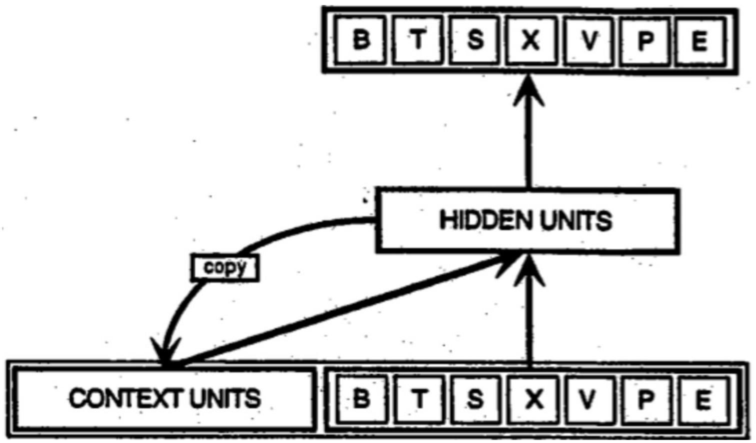
(**capsule** networks)

CERN openlab

Hinton, ICLR 2018

Recurrent Neural Networks and LSTM

Elman, 1990



Recognize patterns in **sequences of data**

Preserve sequential information in **hidden state** across multiple time steps

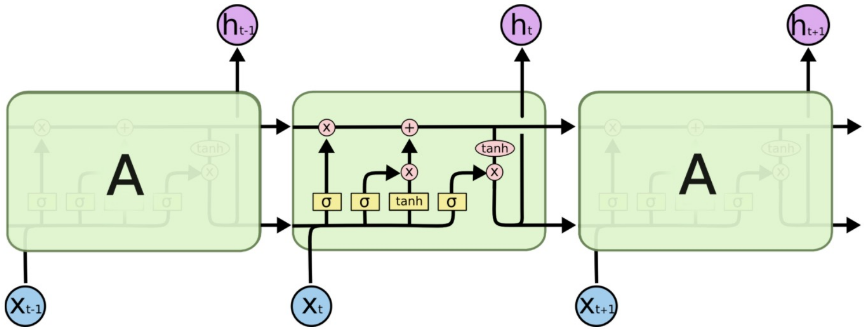
- Input previously analised example together with the current one

Long Short-Term Memory units, by Hochreiter and Schmidhuber in mid 90s

Use **analog gated cells** to allow for data store, reads writes operations.

- element-wise multiplication by sigmoids, (differentiable)

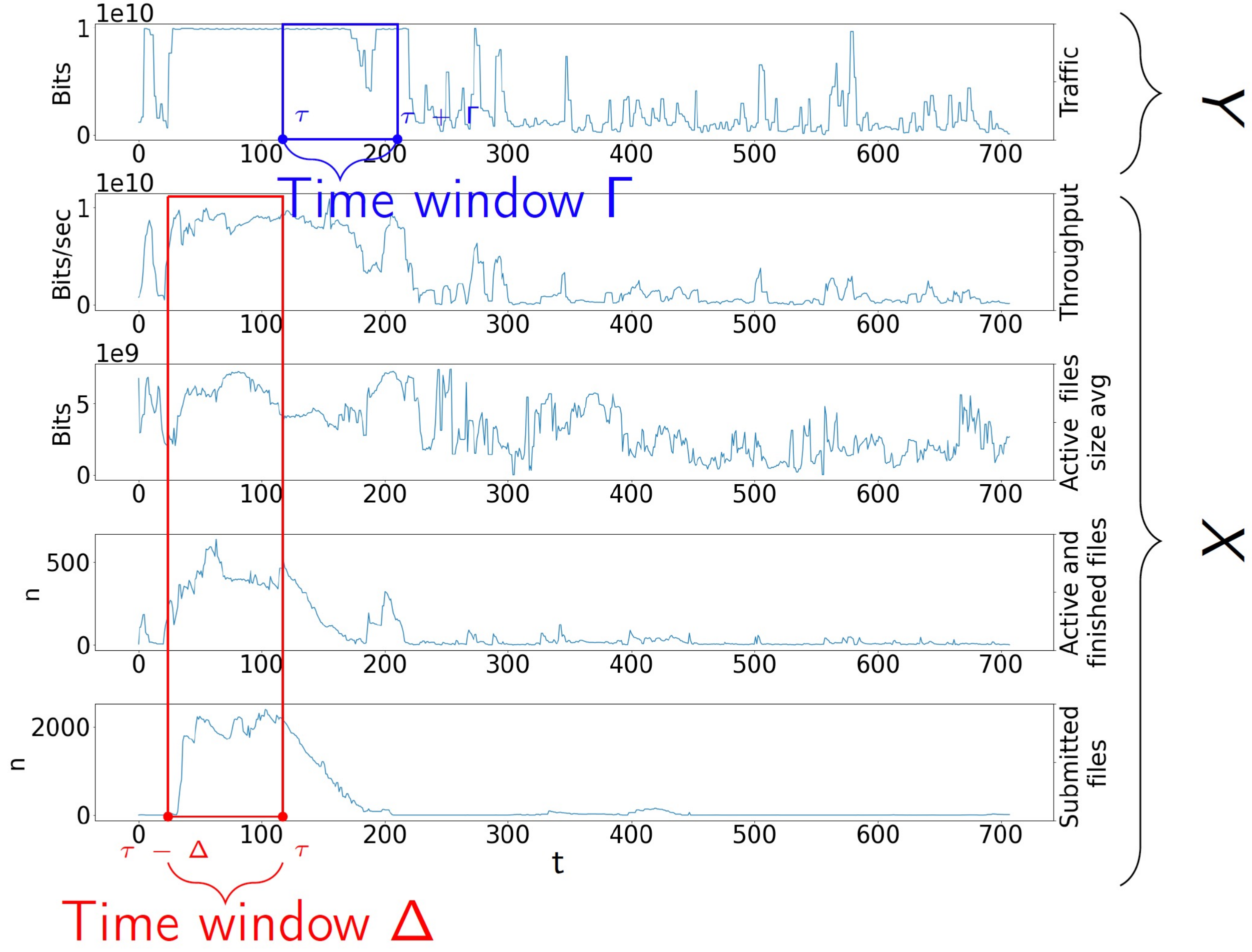
https://people.idsia.ch//~juergen/rnn.html

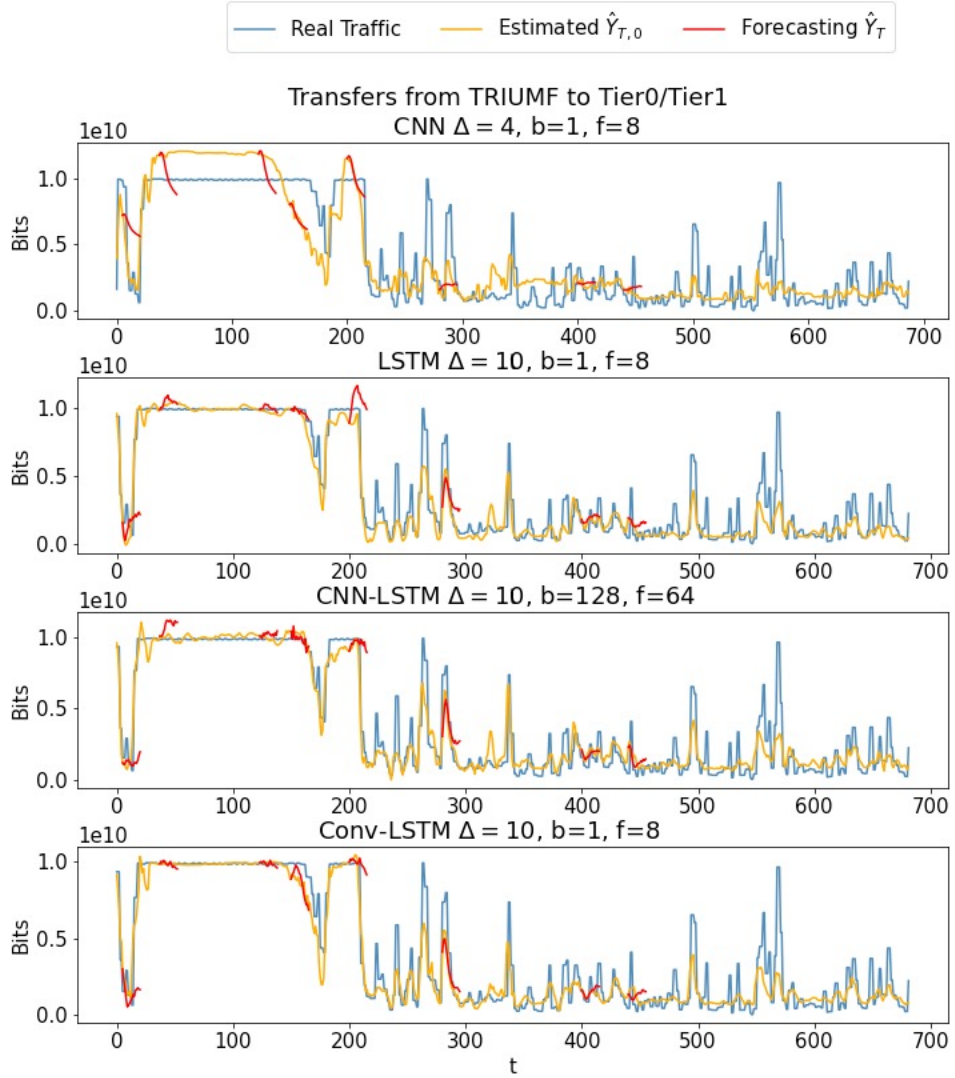


Joanna Waczynska, vCHEP2021, Grid21
arxiv: 2107.02496

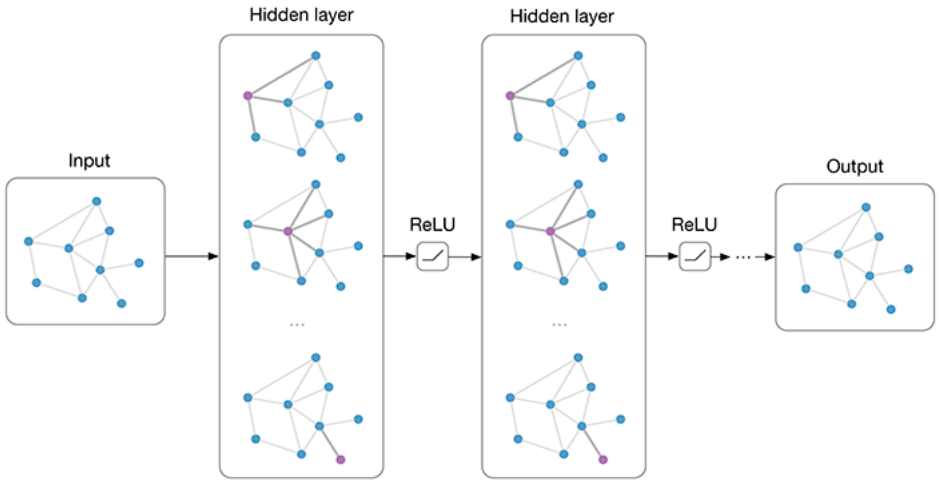
Network traffic prediction @CERN

Compare CNN, LSTM and hybrid architectures

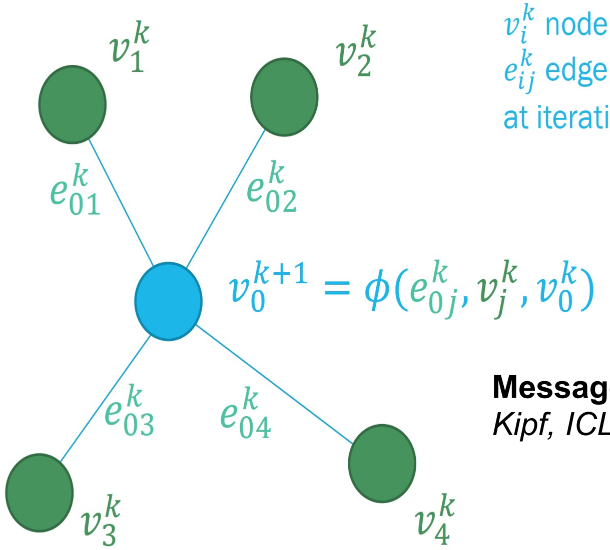




Graph Neural Networks



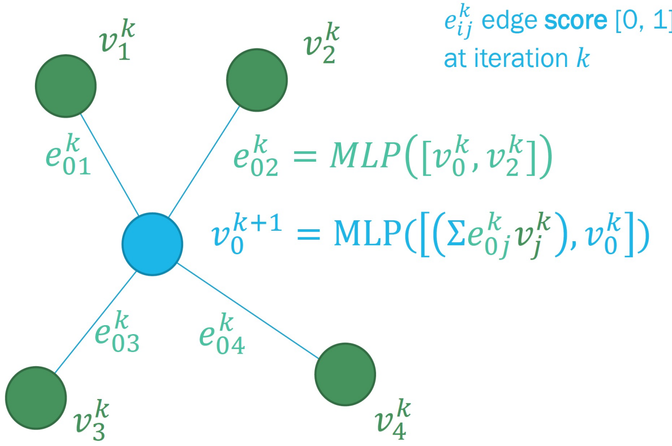
- Structure data as a (directed) graph of connected hits
    - Connect plausibly-related hits using geometric constraints
- Full event embedding requires **large graphs** (~10^5 nodes)
    - **Sparse matrix** implementation
    - Identify disjoint **sub-graphs and distributed learning** of large graphs



v_i^k node features
e_{ij}^k edge features
at iteration k

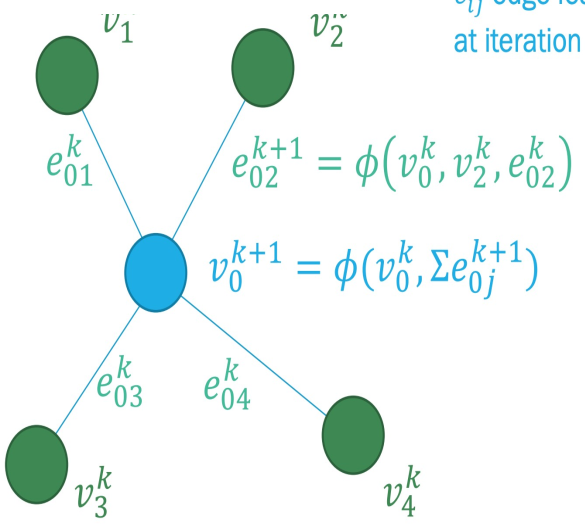
Message Passing
Kipf, ICLR 2017

Attention GNN
arXiv:1710.10903



v_i^k node features
e_{ij}^k edge **score** [0, 1]
at iteration k

Interaction GNN,
Battaglia, Advances in neural information processing systems. 2016.



v_i^k node features
e_{ij}^k edge features
at iteration k

Murnane, Xiangyang, https://indico.cern.ch/event/852553/contributions/4062229/

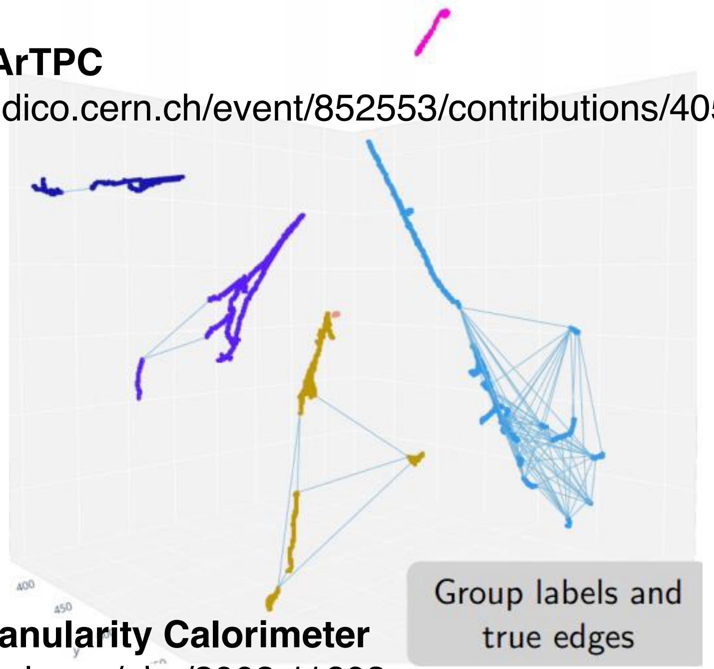
Graph Neural Networks

Next generation colliders will present challenges to **image-based methods**

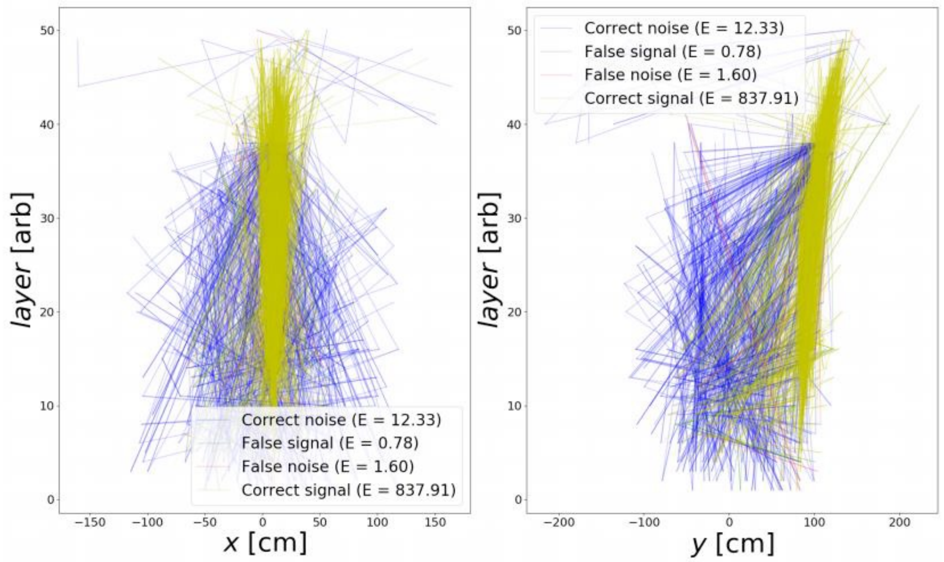
Graphs can capture inherent **sparsity** and **relational** structure

- Can **approximate geometry** of the physics problem
- Are a **generalization** of many other machine learning techniques
- E.g. Message passing convolution generalises CNN from flat to arbitrary geometry

Dune LArTPC
https://indico.cern.ch/event/852553/contributions/4059542/

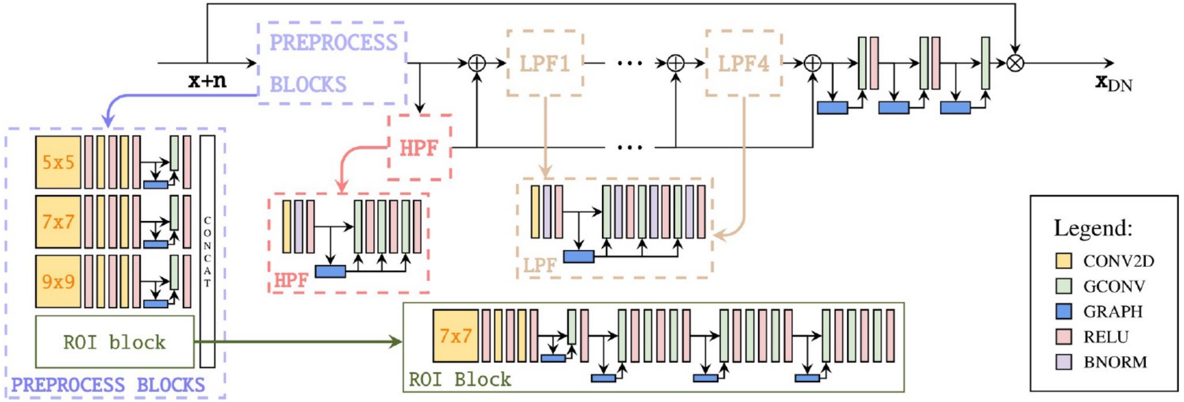


High Granularity Calorimeter
https://arxiv.org/abs/2003.11603

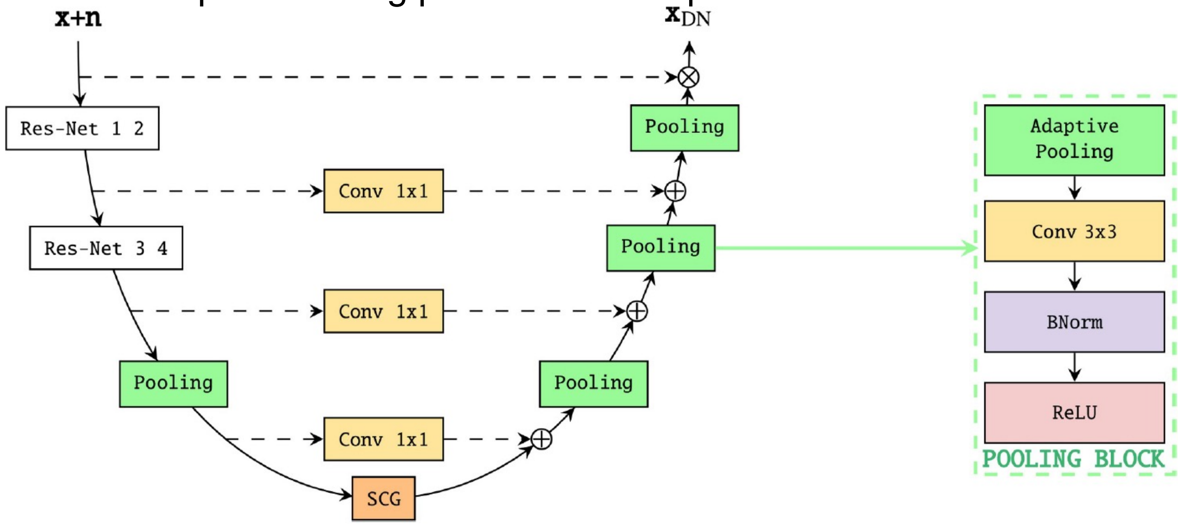


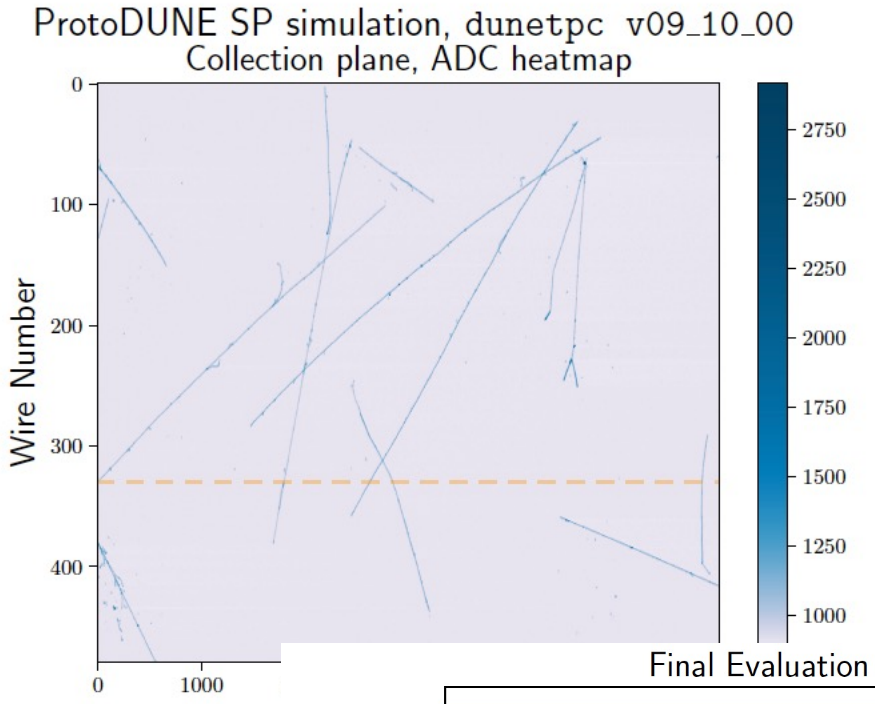
Raw data denoising with hybrid models

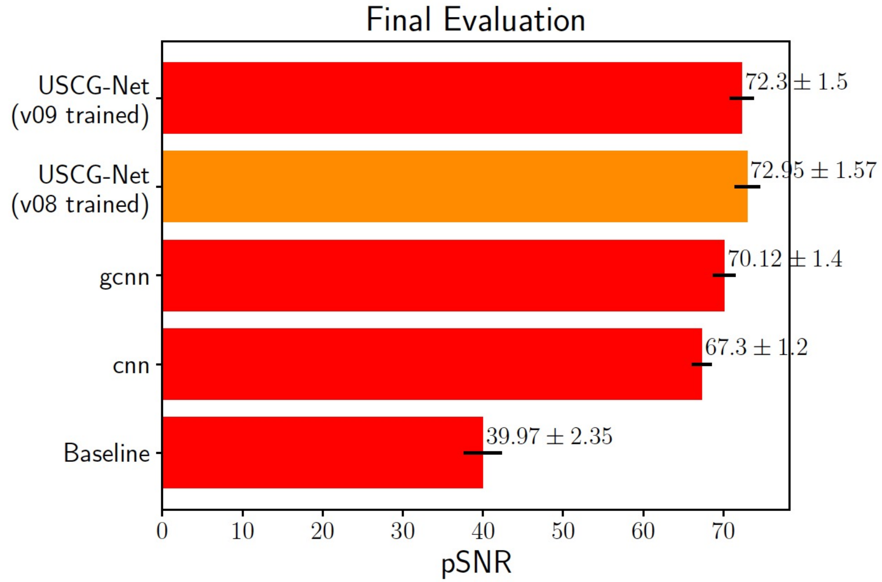
GConV: https://arxiv.org/abs/1907.08448



USCG Net https://arxiv.org/pdf/2009.01599.pdf







Generative models

The problem:

Assume data sample follows p_{data} distribution

Can we draw samples from distribution p_{model} such that $p_{model} \approx p_{data}$?

A well known solution:

Assume some form for p_{model} (using prior knowledge, parameterized by θ)

Find the maximum likelihood estimator

$$\theta^* = \arg\max_{\theta} \sum_{\mathbf{x} \in \mathcal{D}} \log(p_{\text{model}}(\mathbf{x}; \theta))$$

draw samples from $p_{\theta *}$

Generative models don't assume any prior form for p_{models}

Use Neural Networks instead

Deep Generative Models

Deep models allow higher levels of **abstractions** and improve **generalization** wrt to **shallow models**

Multiple applications in Simulation, Anomaly Detection, Data manipulation

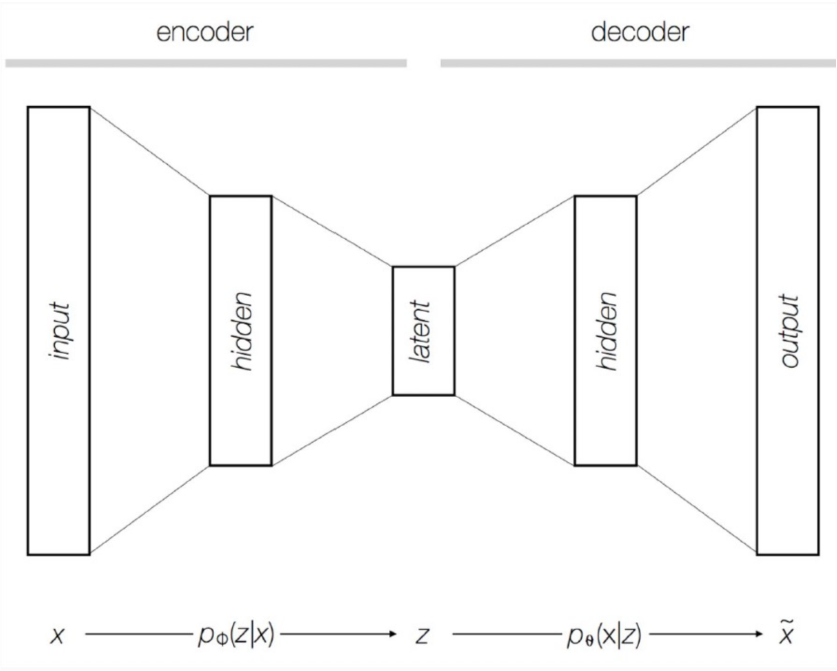
A variety of models:

Generative Adversarial Networks

Auto Encoders

- Compression and decompression are **data-specific, lossy**, **learned automatically from examples**
- Used for data compression, dimensionality reduction (PCA) and de-noising

Variational AEs learn the **latent variable model**



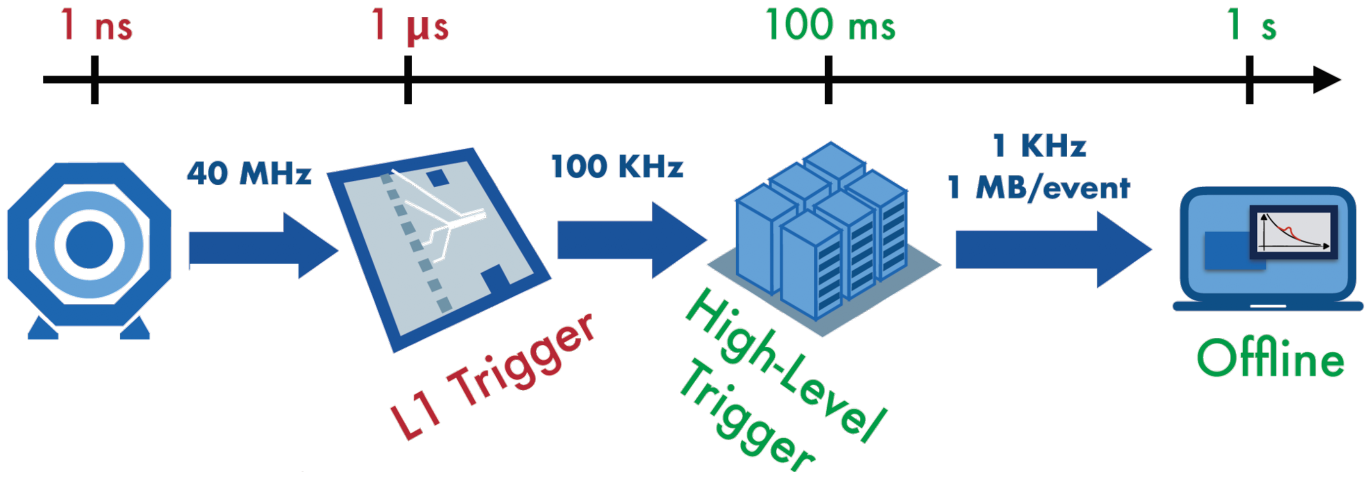
See Danilo Rezende tutorial on Deep Generative Models

Real-time event selection

Only a **minimal fraction** of collider data can be stored and processed

Keep only the **interesting** events

Sophisticated studies to **optimise selection** for specific physics processes



We don't know what **unknown physics** looks like!

arXiv:1811.10276

Physics Mining as anomaly detection

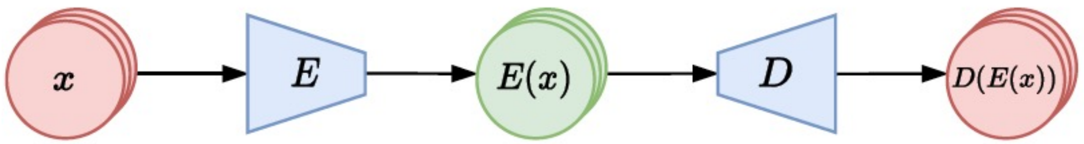
Classical strategy uses very **loose selection**

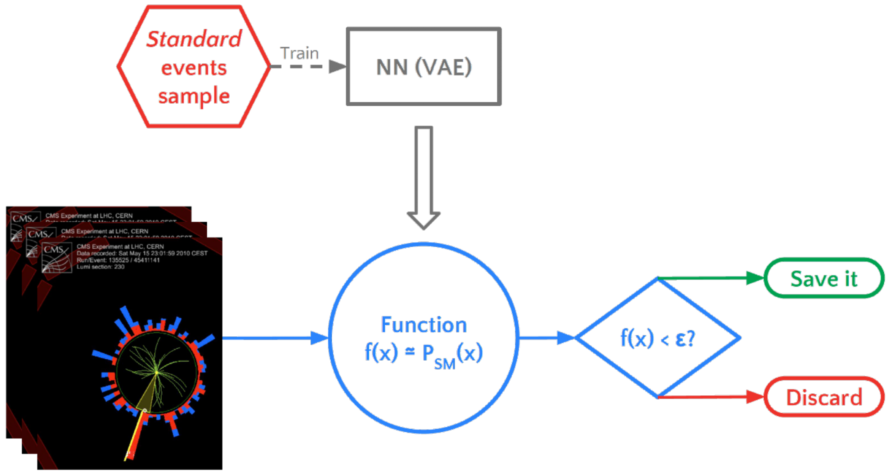
- 1M Standard Model ("known physics") events per day

Train **Variational Auto Encoders** on known physics

- Monte Carlo data
- Real detector data

Run it in real time and store only "**anomalies**"

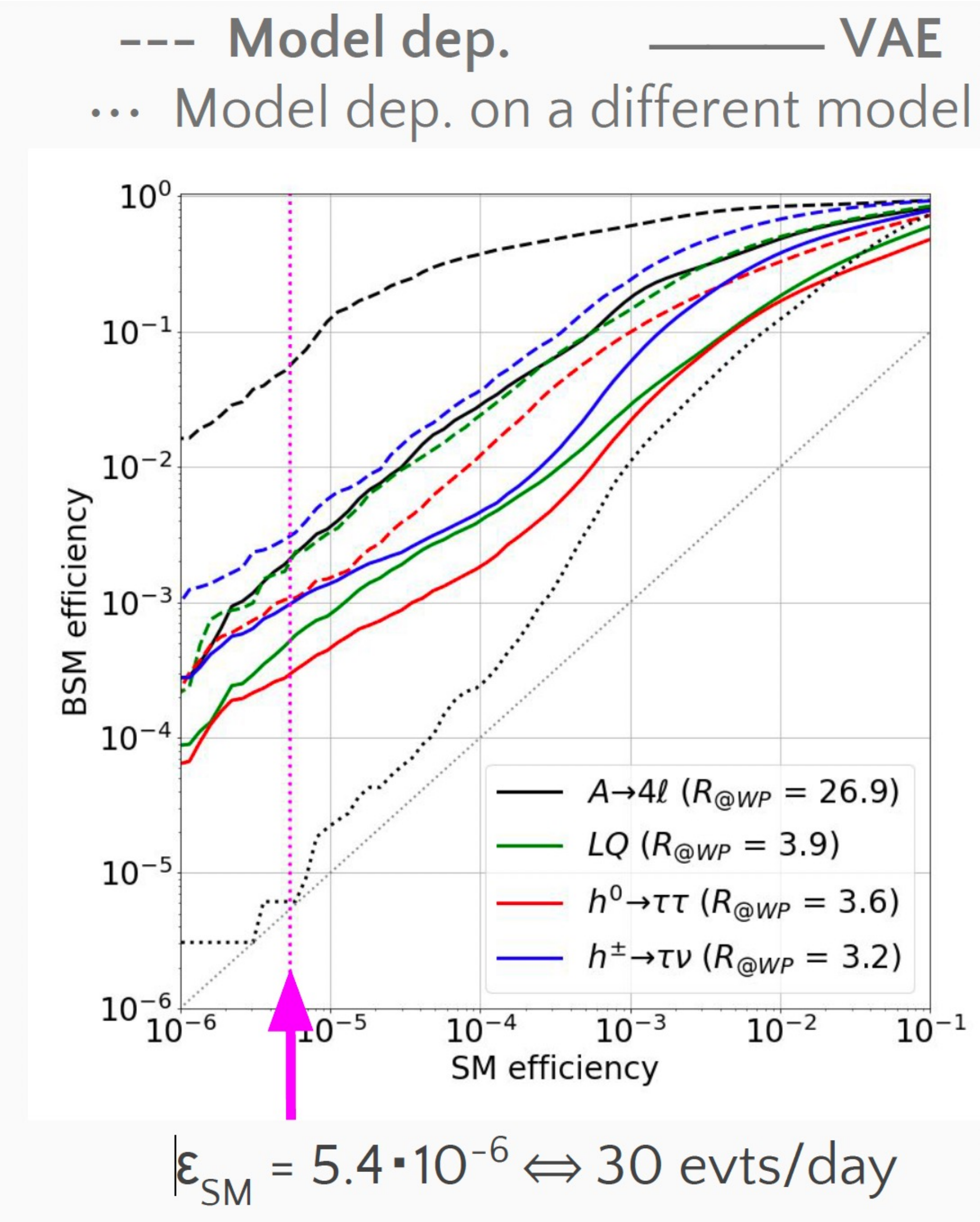




Arxiv:1811.10276

Model-independent selection

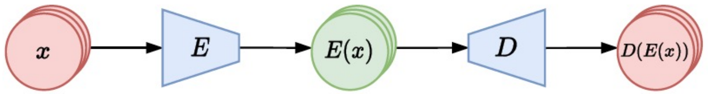
- VAE as **model-independent** new physics selection tool
  - **Robust** alternative solution
- Create a dataset of **anomalous events**
  - Probe **large range of processes**
- Might open **new physics** directions



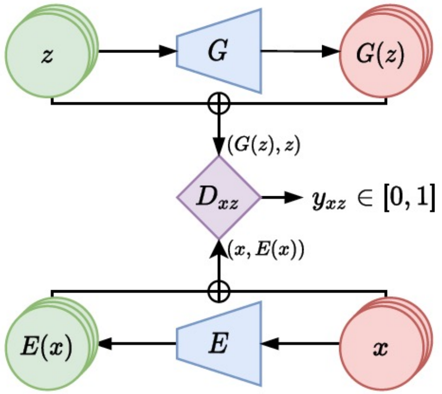
CERN openlab

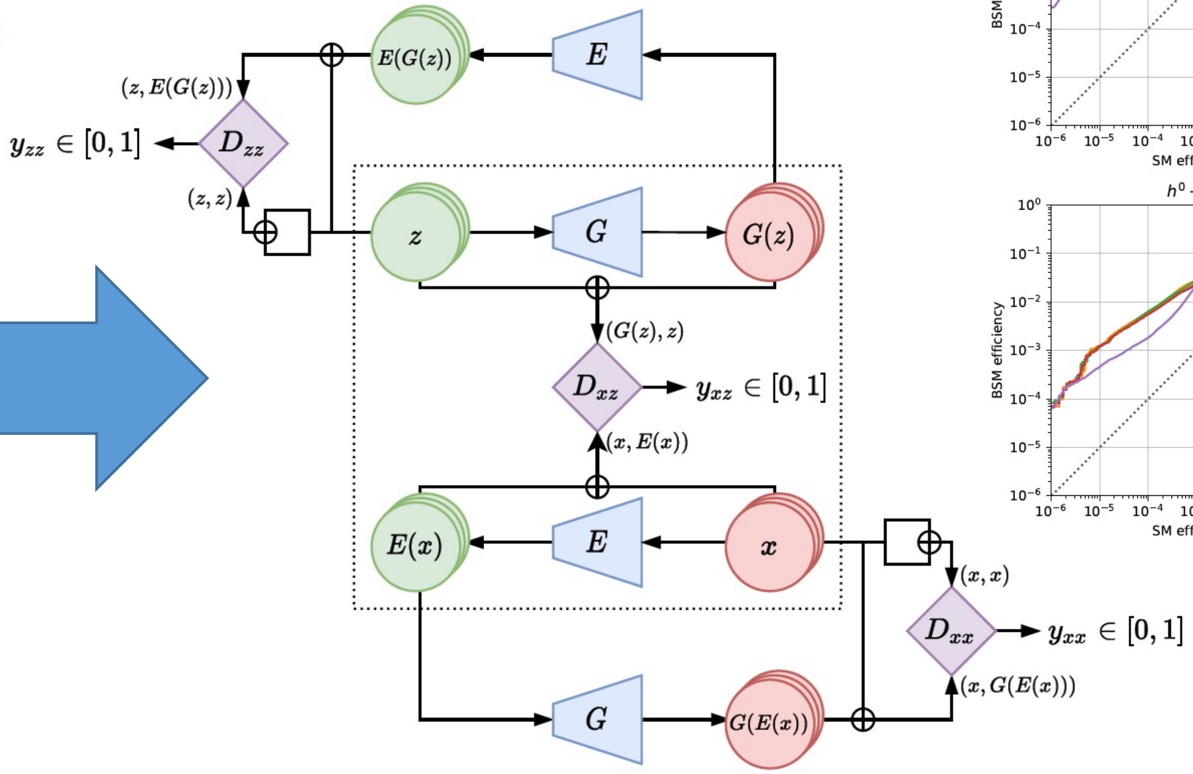
Adversarial training for Anomaly Detection

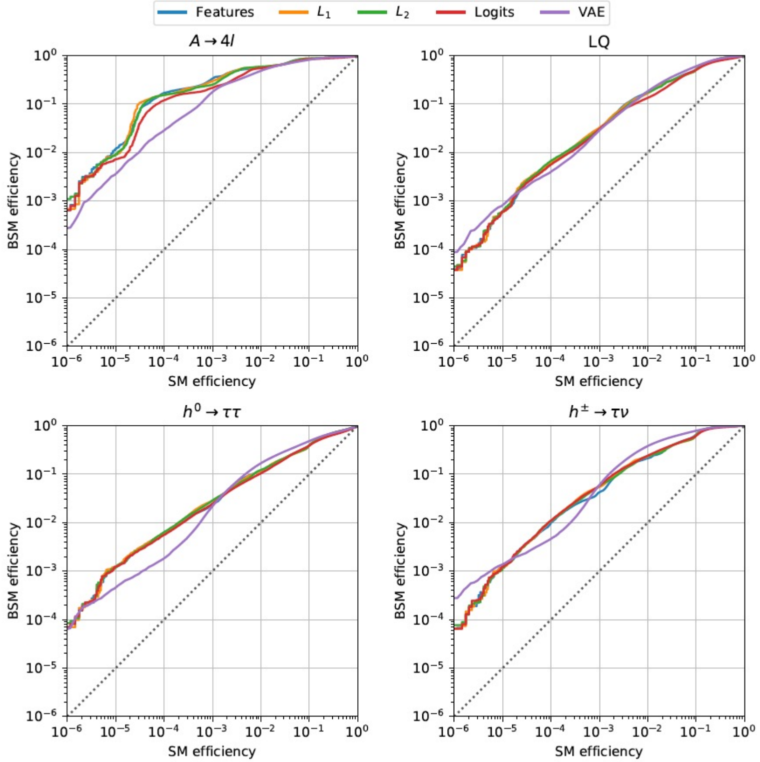
Variational AutoEncoder



BiGAN





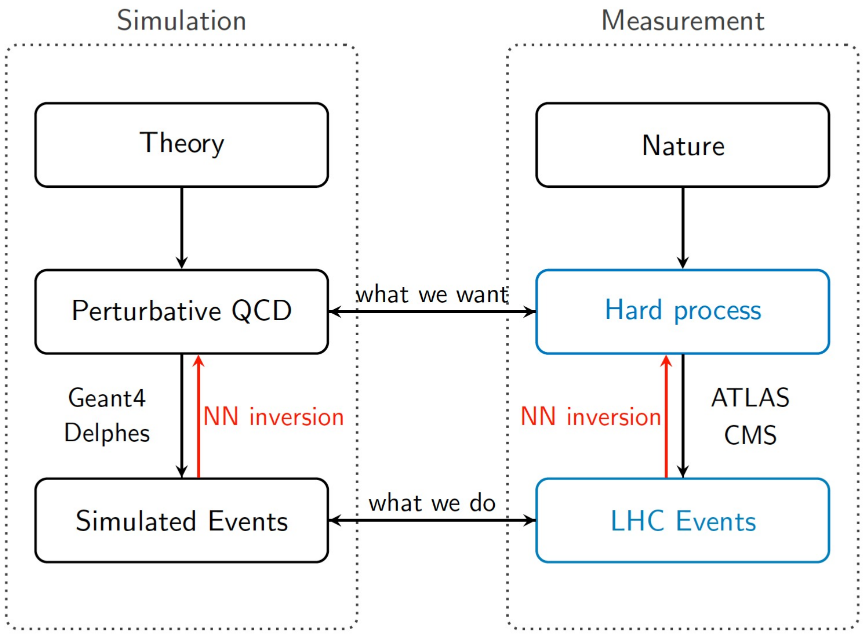


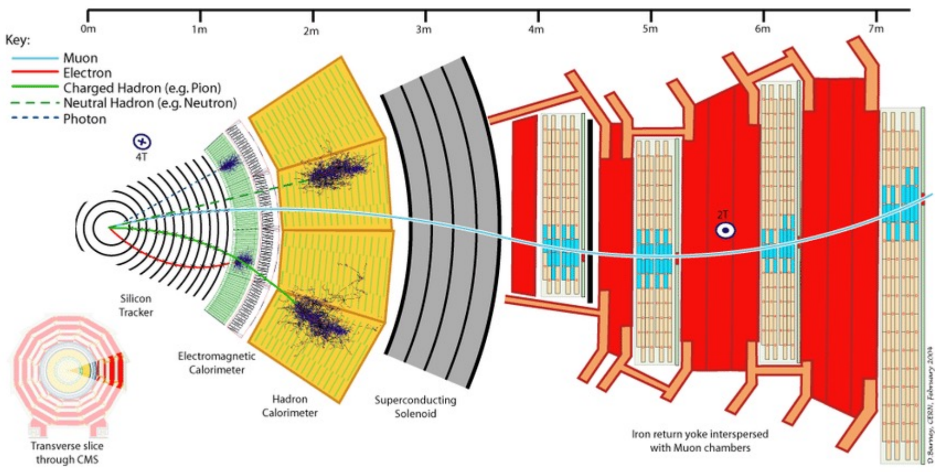
arxiv: 2005.01598

Comparing experimental data to theory

- Detectors measure the results of **particle interactions** with matter
- But we are interested in the **particle production processes**
- Go **back from experiments to theory**:
  - **Disentangle** production process from the experimental setup
  - Bayesian problem

$$p(x|\,y) = \frac{p(y\,|\,x)\,p(x)}{p(y)}$$





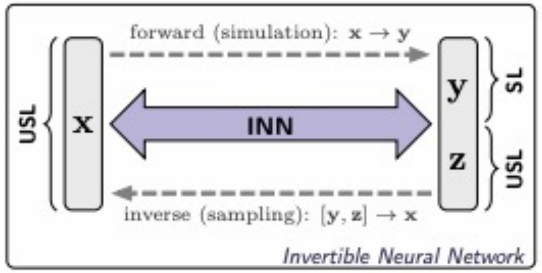
Inverting the experiment

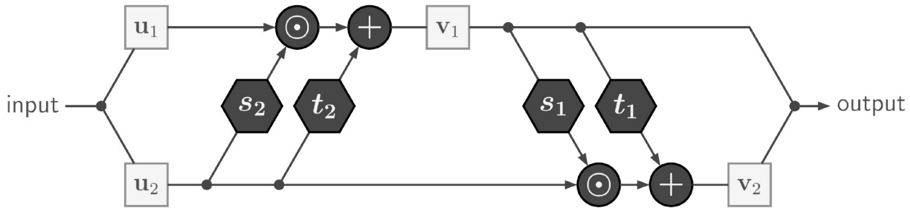
arxiv:1808.04730
arxiv:2006.06685

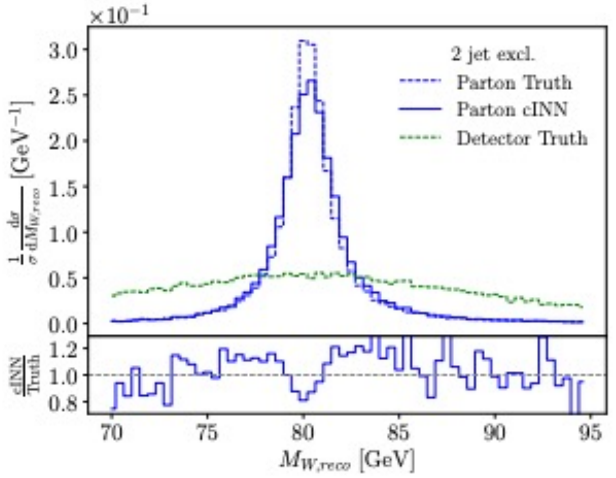
- **Inverse problem**: given observations **y** determine underlying hidden parameters **x**
- **Use invertible networks**
  - Train on the forward process **x** → **y**
  - Run backward **y** → **x** to get prediction
  - Add latent variable z to compensate information loss during forward process

$$[\mathbf{y},\mathbf{z}] = f(\mathbf{x})$$
$$p(\mathbf{z})$$

$$\mathbf{x} = f^{-1}(\mathbf{y},\mathbf{z}) = g(\mathbf{y},\mathbf{z})$$

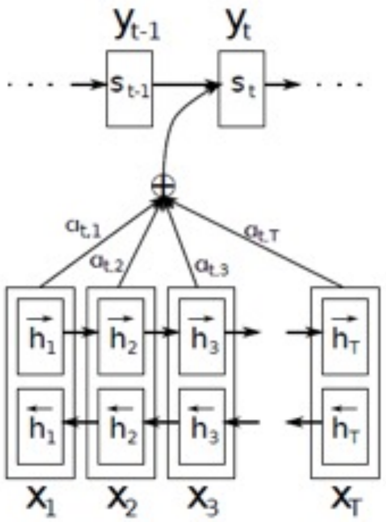






Attention mechanism

Attention mechanism as originally formulated in a bi-directional LSTM Auto-Encoder https://arxiv.org/abs/1409.0473



Focus on special region of input phase space

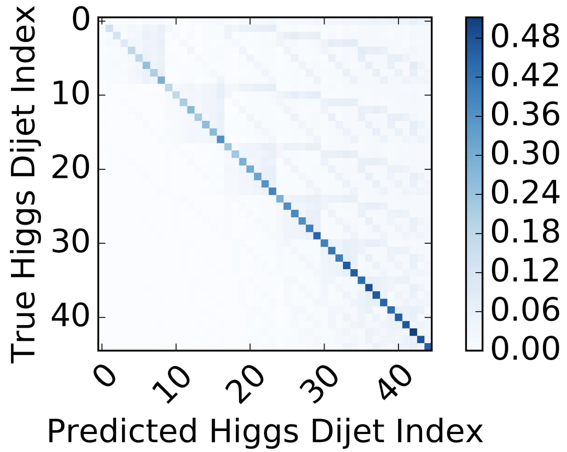
- interpretation as a vector of importance weights
- Ex. soft attention as modules in a layer to dynamically select vectors from the previous layer

Output is independent of the order of input examples (set instead of sequences)

Use **relationships** between different inputs (as graph representation).

Stacked self-attention layers at the base of **transformers** (Vaswani et al., *Advances in Neural Information Processing Systems*, 2017, 5998–6008)

Attention mechanism applied to Higgs classification: C. Reissel, ML4Jets 2021



Example **transformers** application in HEP: https://iopscience.iop.org/article/10.1088/2632-2153/ac07f6/meta

Kendal, Gal, NIPS 2017, https://papers.nips.cc/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf

Uncertainties

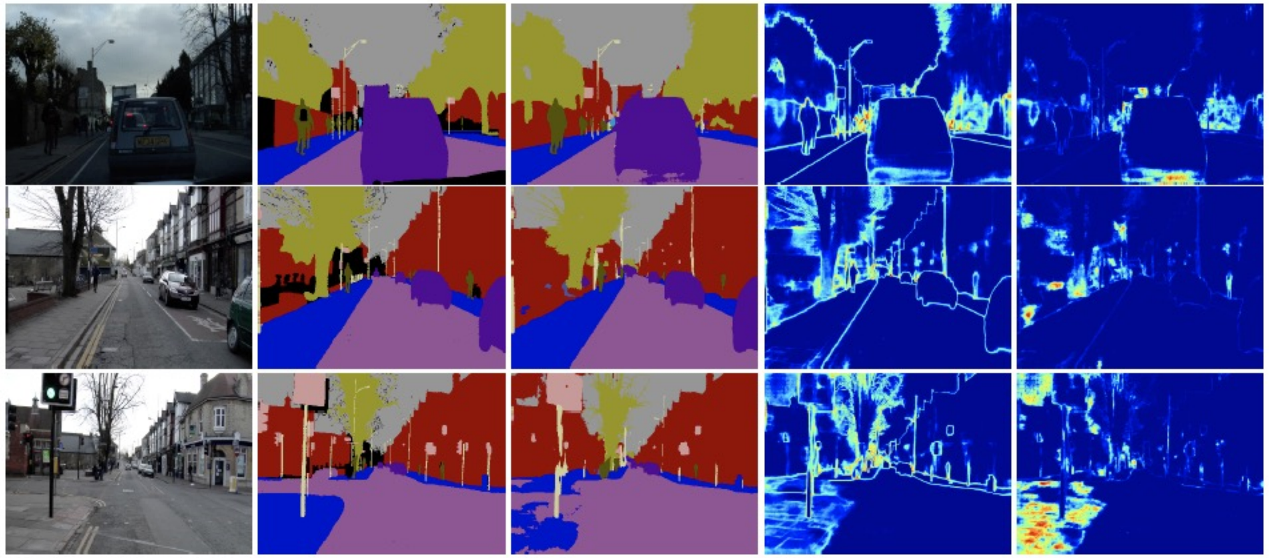
Aleatoric uncertainty captures noise inherent in the observations.

- Higher on object boundaries and for objects far from the camera.
- Cannot be reduced using more data, needs better measurements

Epistemic uncertainty accounts for ignorance about which model generated the data.

- Higher for semantically and visually challenging pixels.
- It can be explained away given enough data.
- Introduce a prior distribution (Bayes statistics)

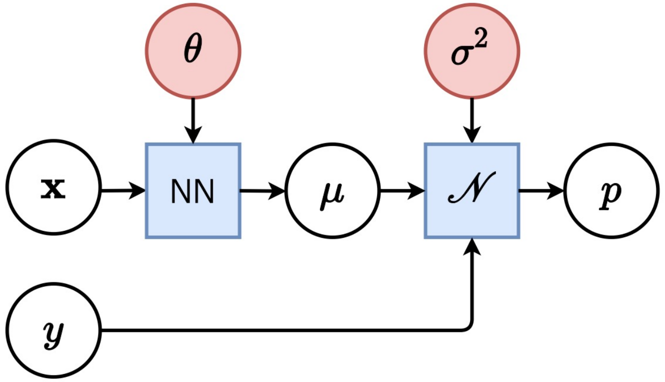
Learn uncertainty within the task.



(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

the model fails to segment the footpath due to increased epistemic uncertainty, but not aleatoric uncertainty

Ex. Regression: model aleatoric uncertainty in the output by modelling the conditional distribution as a Normal distribution



Find more details in: G.Louppe, Introduction to Deep Learning, https://glouppe.github.io/info8010-deep-learning/pdf/lec11.pdf

Development directions

ML/DL have their origins in the studies on the human brain, but today DL doesn't learn like humans do.

Current research in DL tries to improve on this aspects

G. Hinton, Y. Le Cunn, Y. Bengio , AAAI 2020 keynotes, Turing Award Winners Event
https://www.youtube.com/watch?v=UX8OubxsY8w

New improvements will not be achieved by simply making models **larger and larger**

Alternative architectures and **approaches to learning** :

- **Attention mechanism**
- **Self Supervised Learning**: systems learn from raw data to label it.

Generalisation: capability to generalize to different data distributions (out-of-distribution generalisation)

Thanks!

Sofia.Vallecorsa@cern.ch

https://openlab.cern/