Tracking Kaggle Challenge

Vincenzo Innocente
for the Tracking Kaggle Group
Slides by Paolo Calafiura

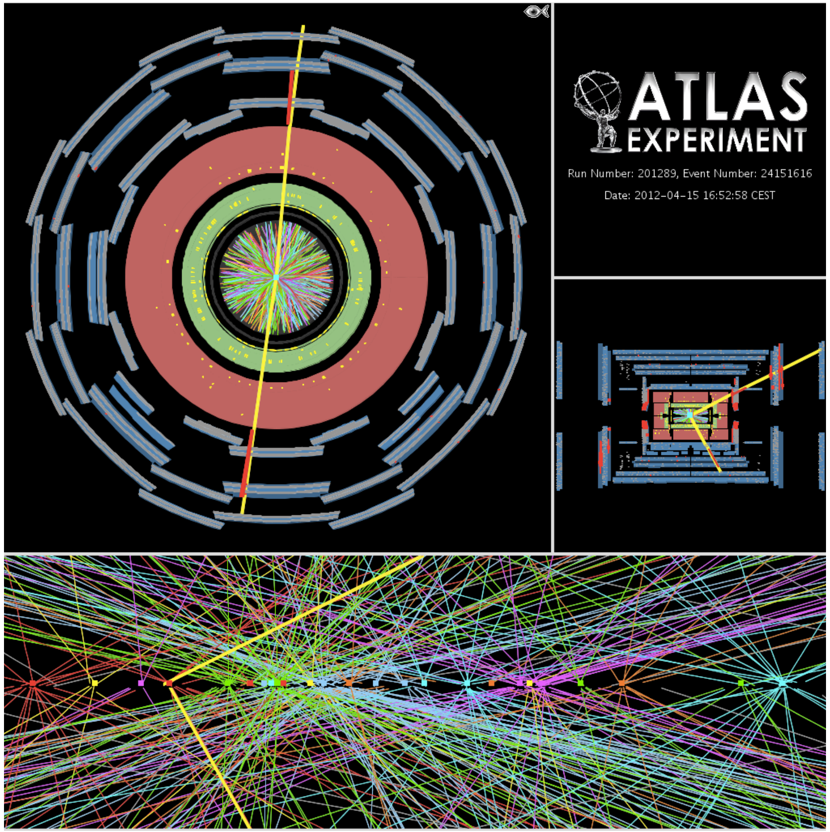
Tracking in Run 4

~60M track/s (20x Run 2)

- x2-5 CPU shortage offline (another one…)

If we can't write **~40GB/s RDOs** we'll need to gain (a lot) more than that **online**.

- Surely we can parallelize our way out of trouble?



Why is Parallel Tracking so hard?

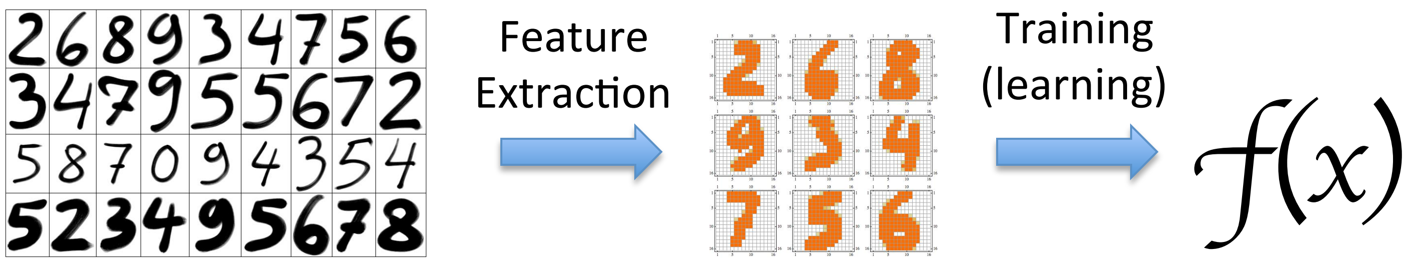
Algorithms: Iterative (propagation, fitting), irregular (combinatorial searches with lots of branch points)

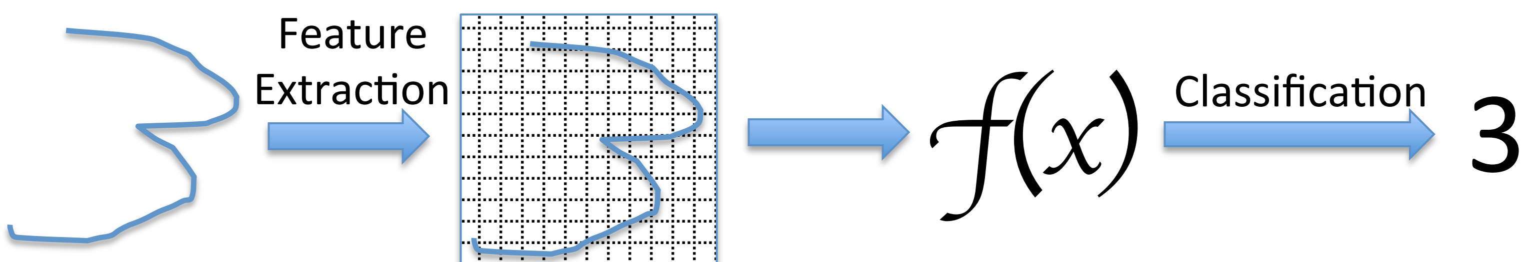
Data: sparse (hits), non-local (B-field integration)

*Can **Machine Learning (ML)** provide a solution that uses regular, simple algorithms, and is naturally data parallel?*

Machine Learning

Data-driven adaptive modeling of a system



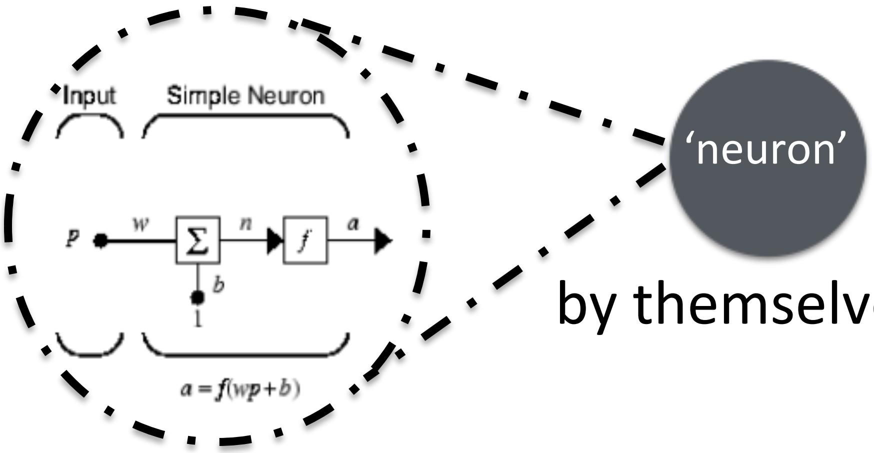


Why Now?

Current hardware (HPC, GPUs) make possible to train very complex NNs with millions of inputs and thousands of outputs.

Exploiting GPUs, FPGAs, upcoming Neuromorphic hardware pushes us towards **Computing with many simple elements**

Computing with simple elements

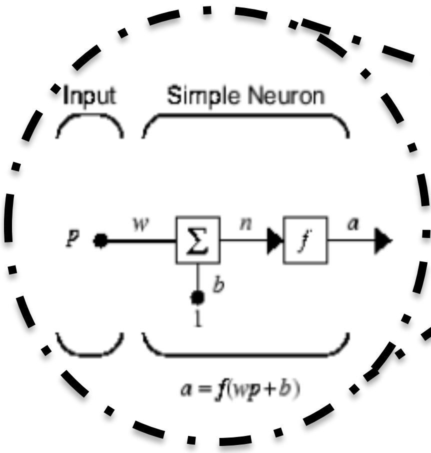


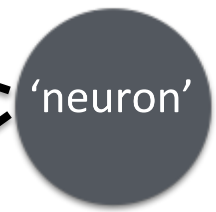
Simple computing elements…

by themselves, limited functional repertoire.

(Kristofer Bouchard, LBNL)

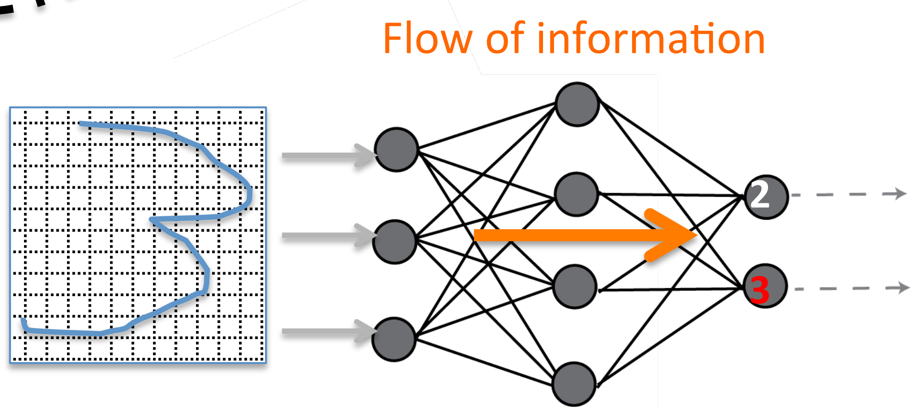
Feed-forward NN: Classification





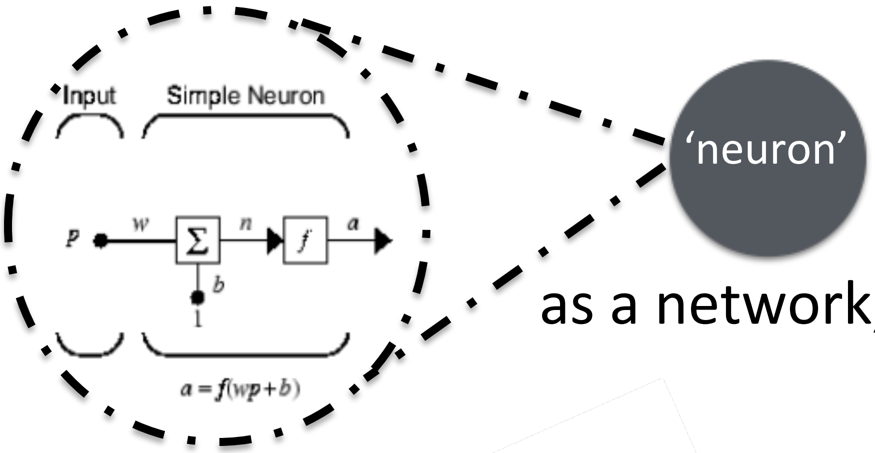
Simple computing elements…

as a network, learn to perform diverse functions



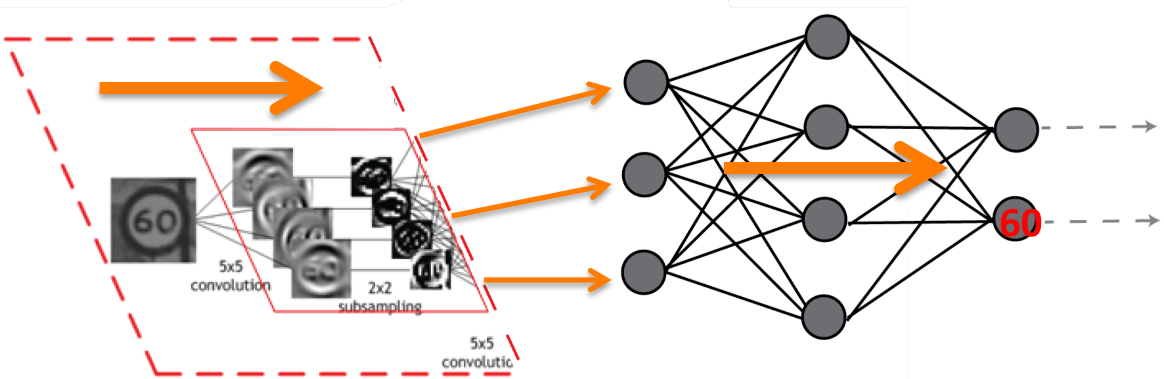
Classification

Convolutional NN: Feature Extraction



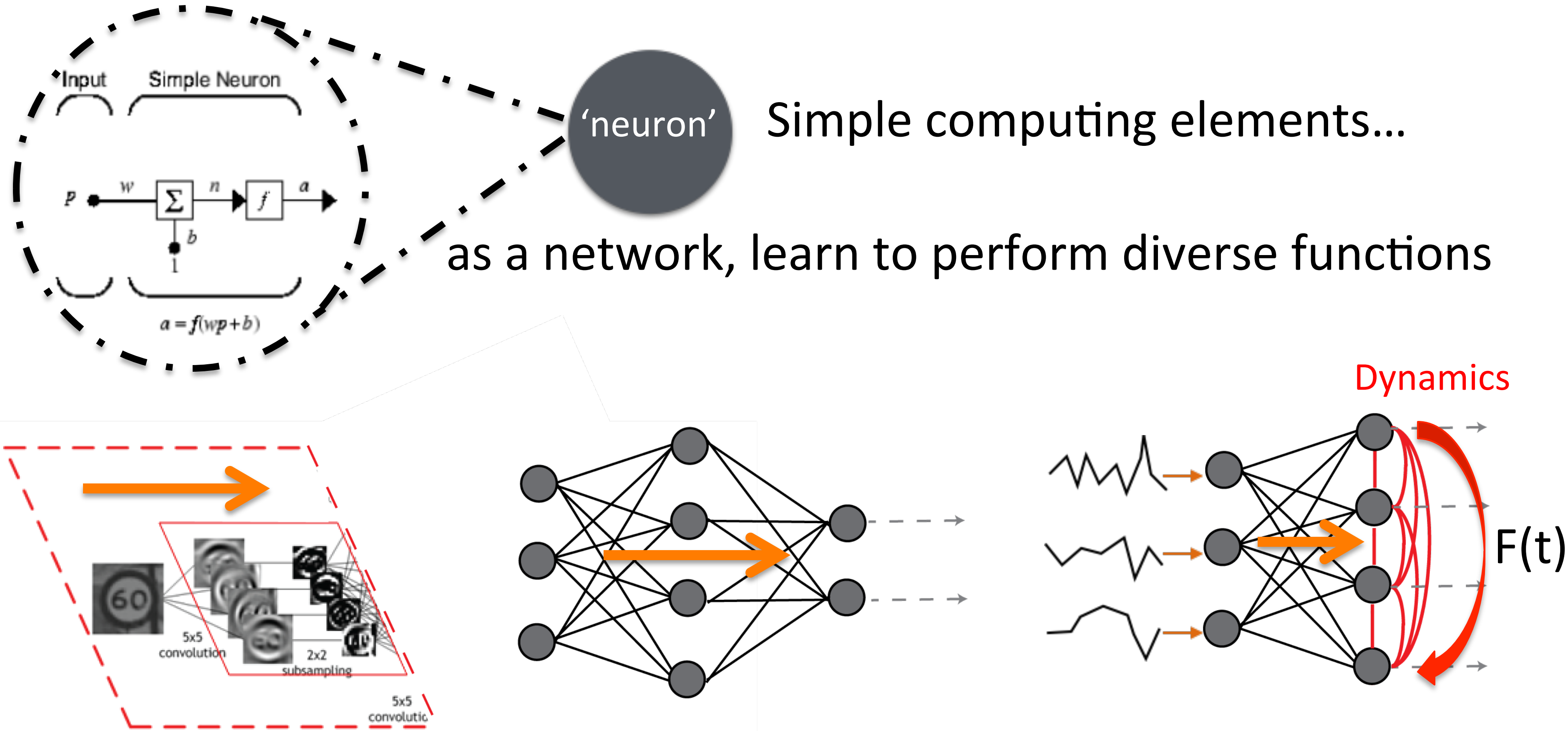
Simple computing elements…

as a network, learn to perform diverse functions



Feature Extraction **Classification**

Recurrent NN: Time-varying Functions



Feature Extraction **Classification** **Time-varying Functions**

Higgs ML Challenge

Huge success

ML already in use for Higgs analysis

Still a big effort to setup (~2 years)

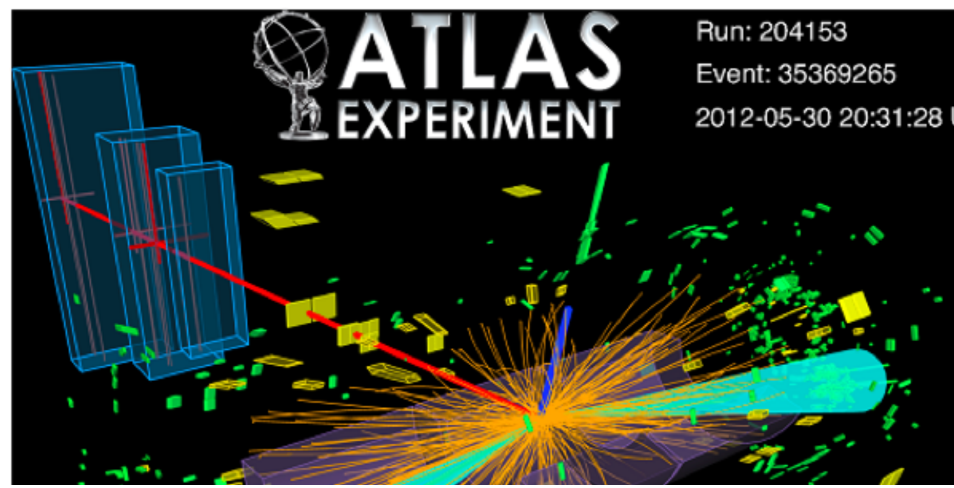
Completed • $13,000 • 1,785 teams

Higgs Boson Machine Learning Challenge

Mon 12 May 2014 – Mon 15 Sep 2014 (9 months ago)

Competition Details » Get the Data » Make a submission

Use the ATLAS experiment to identify th
Higgs boson



Connecting The Dots 2015

A Workshop on Pattern Recognition in Sparsely Sampled Data

The Berkeley Experimental Particle Physics Center Workshop Series

Connecting The Dots 2015
LBNL, Feb. 9-11, 2015

David R, Markus E, and PC volunteered to organize a Tracking Challenge

Tracking Kaggle Challenge

- One question (follow-up questions possible)
- One evaluation metric (training function)
- Two data samples: Training (labelled), Test
- One “starting kit” (reference solution)

Our Question

<u>Immediate Goal</u>:
Build a fast, **scalable,** pattern recognition engine

<u>Real Goal</u>:
Learn if-how-where to apply ML to reconstruction

<u>Question</u>:
Given a list of space-points, identify those belonging together (to a track).

May add fitting at a later time

Evaluation Metric

- Non-trivial, still under discussion
  - Positive weight for each hit correctly assigned to track
  - Negative weight for fake hits
- Issues
  - Balance efficiency, fake rate, and complexity
  - Need to fold into metric instructions needed to run.

Data Sample

json format, may move to HDF5 (on afs ~calaf/public/kaggle)

```
{
        "Identifier" : 147838943612633088,
        "GlobalX" : 33.180407,
        "GlobalY" : -3.219767,
        "GlobalZ" : -112.9625,
        "WidthPhiR" : 0.05,
        "WidthZ" : 0.5,
        "energyLoss" : -28,
        "splitProbability1" : 0,
        "splitProbability2" : 0,
        "Deposits" : [
            {
                "Charge" : 13051.327148,
                "TruthEventIndex" : 0,
                "BarCode" : 200001
            }
        ]
    },
```

- Per-job, geometry info in separate file

- Not there for noise hits.
- Training file only

Currently Run 2 Pixel with single mu events. Will try tau soon.
Next we will use ITK (Run 4) configuration. Later will add pileup.
No idea how many events will be needed, expect $O(10^9)$ tracks.

Starting Kit

Provide reference solution to kaggle competitors

- Typically a simple ML solution
- Could provide simplified HEP Tracking, issue is how to package to run standalone.

David Clark (UCB) started working on single track classifier using Caffe.

  - Outputs probability N hits belong to track
  - Trained using chisquare(hit, propagated GenParticle)

See https://github.com/davidclark1/TrackNet

Tracking Kaggle Group

- tracking-kaggle@googlegroups.com (open, currently 20 members)
- Members from
  - ATLAS, CMS, LHCb (since last week)
  - ML experts from HiggsML
  - Tracking experts from Connecting The Dots
- Meets every other Mon at 17:10 CET on vidyo
- Time-frame next CTD workshop (Feb 16?)
  - Settle on question, data samples, starting kit strategy by Sep 15

Final Thoughts

- In 2025 ML will be more mainstream than C++
  - Ideally suited for highly parallel architectures with simple computing nodes (GPUs, FPGAs, …)
  - Likely **part of** any Run 4 tracking solution
- ML is not a trivial subject, will take years to build HEP expertise.
- The Kaggle Tracking Challenge is a great opportunity to enter this brave new world.

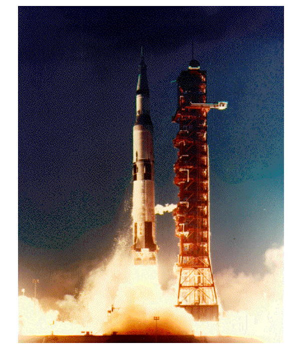
Thanks

- Paolo Calafiura and Collaborators
- Kristofer Bouchard
- Maurice Garcia-Sciveres
- Beate Heinemann
- Peter Nugent
- Peter Sadowski
- …

Backup

Kalman Filters and Recurrent NNs

Classic Data Assimilation algorithm (1960, NASA)
Iteratively **track** evolution of a **dynamic** system



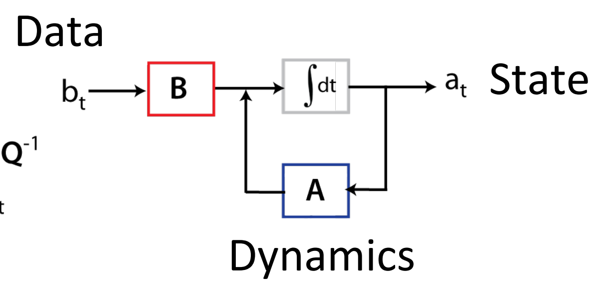
Kalman Filter

(1) $a_t = \mathbf{A}a_{t-1} + N(0,\mathbf{W})$

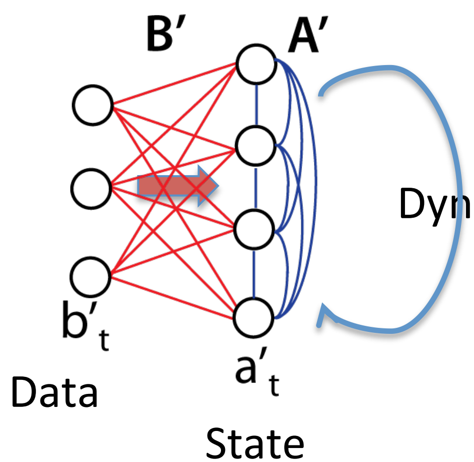
(2) $b_t = \mathbf{B}a_t + N(0,\mathbf{Q})$

(3) $\mathbf{K} = (\mathbf{I}\text{-}\mathbf{WBQ}^{-1}\mathbf{C})^{-1}\mathbf{WBQ}^{-1}$

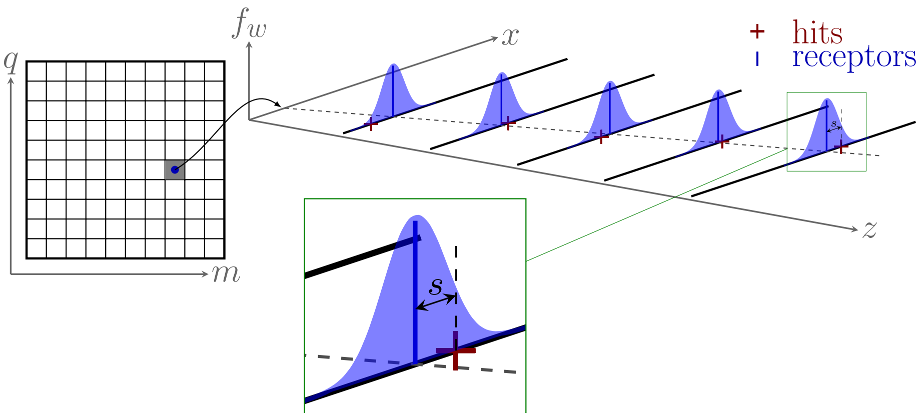
(4) $\hat{a}_t = (\mathbf{I}\text{-}\mathbf{KB})\mathbf{A}\hat{a}_{t-1} + \mathbf{K}b_t$



Recurrent Neural Network

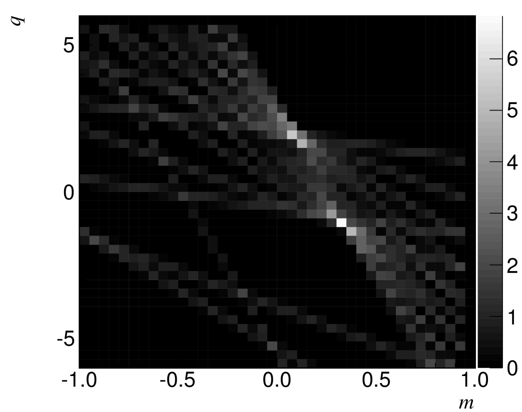


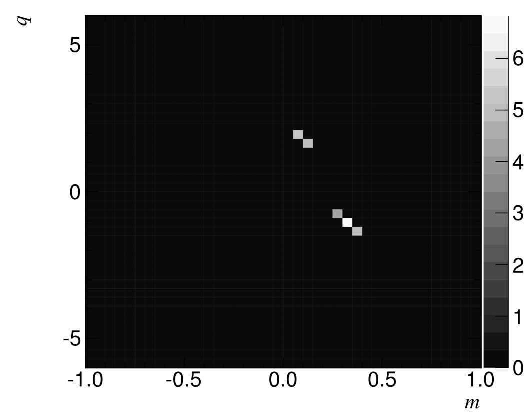
LHCb Trigger Retina Processor



Transform to track parameter space
22K bins, one "receptor" per bin

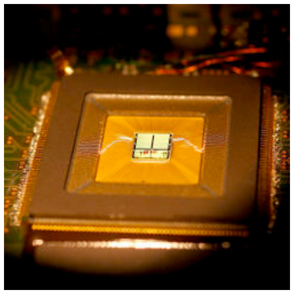
FPGA implementation
1mus tracking
Offline-quality performance
Certainly good enough for seeding





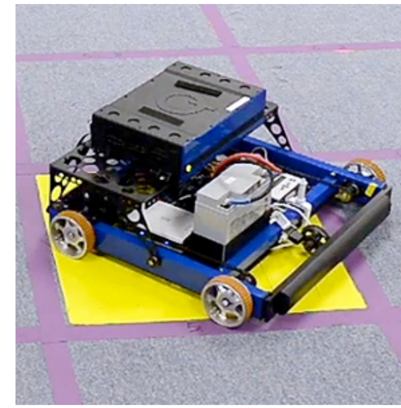
(Simone Stracka, Pisa)

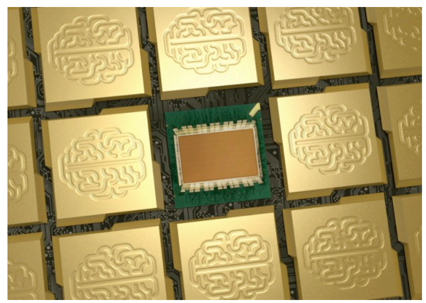
Neuromorphic Computing



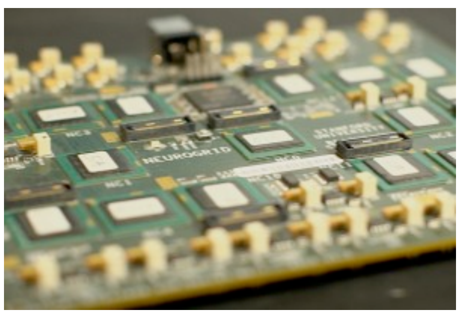
"Spikey" from Electronic Visions group in Heidelberg

Qualcomm's NPU's for robots.



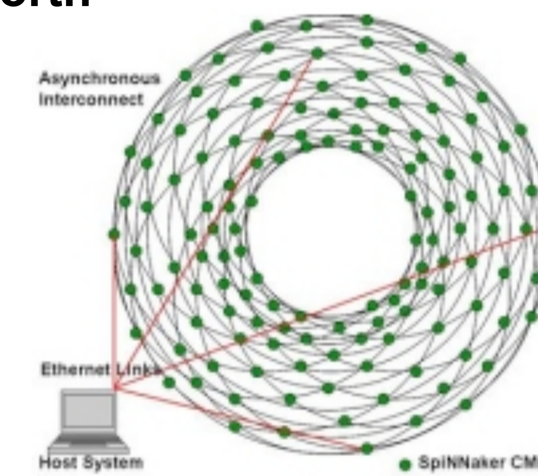


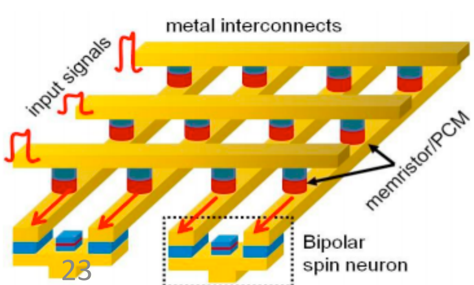
IBM's TrueNorth



Stanford's Neurogrid

SpiNNaker's 1B neuron machine

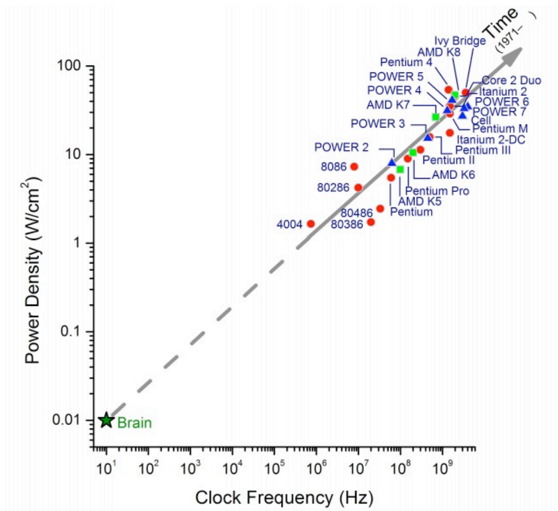




Intel's concept design...

(Peter Nugent, LBNL)

IBM TrueNorth



Merolla+ *Science* (2014)

- **1 million programmable neurons**
- **256 million synapses**
- **4096 neurosynaptic cores**
- **Uses 70mW per chip**
- **5.4 billion transistors**
- **Spiking rate >1000Hz**

A single chip can process color video in real-time while consuming 176,000 times less energy than a current Intel chip performing the exact same analysis. Note the Intel chip can *not* do this analysis in real-time and is in fact 300 times slower!