New ideas from physics to machine learning

Stefano Carrazza

Durham, 3-6th April 2018.

European Organization for Nuclear Research (CERN)

Acknowledgement: This project has received funding from HICCUP ERC Consolidator grant (614577) and by the European Unions Horizon 2020 research and innovation programme under grant agreement no. 740006.

N3PDF
Machine Learning • PDFs • QCD

CERN

Introduction

Motivation

Usually, machine learning methods require investigation and tuning of:

Parameterization, e.g.

- NN, Deep NN
- New architectures
- Auto-ML

Minimization, e.g.

- Gradient descent methods
- Genetic optimizers
- Reinforcement/Q-learning

In the context of NNPDF the next level of refiniments includes:

- Better minimizers based on SGD algorithms:
  $\rightarrow$ possibility to test algorithms for modern DNN training
  $\rightarrow$ improve fit convergence/speed
- Test more efficient architectures:
  $\rightarrow$ NN, DNN and new models.

The development of both points will provide hints towards new methodological ideas (single multi-flavour agent, Q-learning).

New ideas from physics to machine learning

In this talk we present new a machine learning architecture:

Riemann-Theta Boltzmann Machine

- flexible as NN but with less parameters
- allow multiple applications:
  - data regression
  - data classification
  - feature detection
  - pdf sampling

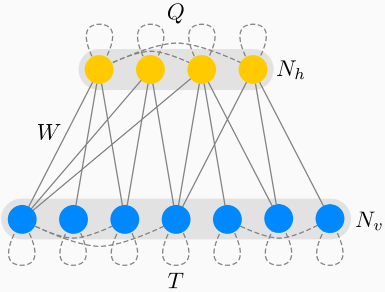
We derive a new architecture from physics:

- model based on physics $\rightarrow$ ML new architecture

Theory

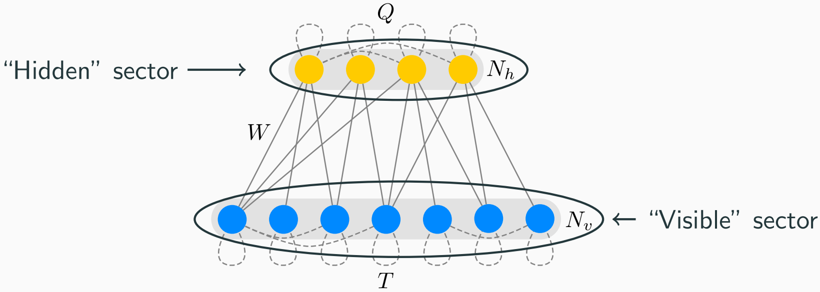
Boltzmann machine

Graphical representation:



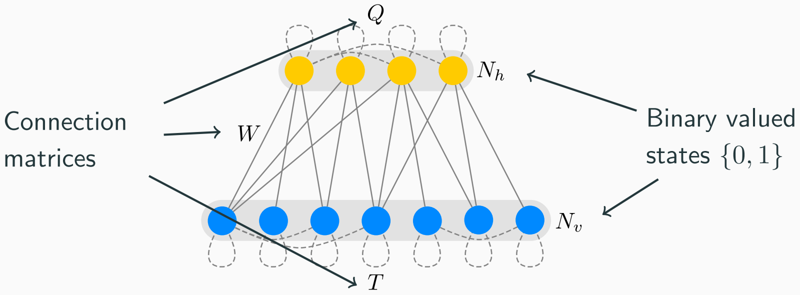
Boltzmann machine

Graphical representation: **[Hinton, Sejnowski ‘86]**



Boltzmann machine

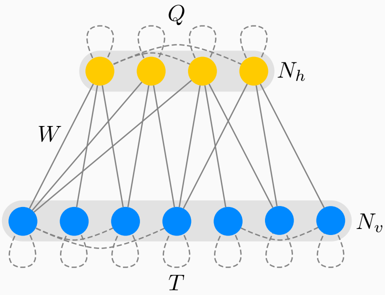
Graphical representation: **[Hinton, Sejnowski '86]**



- Boltzmann machine (BM): T and Q symmetric arbitrary.
- Restricted Boltzmann machine (RBM): $T = Q = 0$.

Boltzmann machine

Energy based model: **[Hinton, Sejnowski '86]**



View as statistical mechanical system.

The system energy for given state vectors (v, h):

$$E(v,h) = \frac{1}{2}v^t T v + \frac{1}{2}h^t Q h + v^t W h + B_h h + B_v v$$

State vectors — Connection matrices — Biases

Boltzmann machine

Energy based model: **[Hinton, Sejnowski '86]**

Starting from the system energy for given state vectors (v, h):

$$E(v,h) = \frac{1}{2}v^tTv + \frac{1}{2}h^tQh + v^tWh + B_hh + B_vv$$

The canonical partition function is defined as:

$$Z = \sum_{h,v} e^{-E(v,h)}$$

Probability the system is in specific state given by Boltzmann distribution:

$$P(v,h) = \frac{e^{-E(v,h)}}{Z}$$

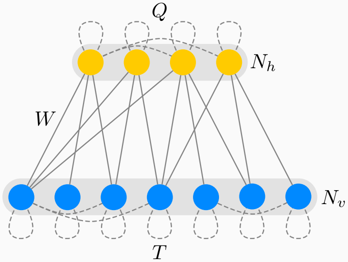
with marginalization:

$$P(v) = \frac{e^{-F(v)}}{Z}$$

← Free energy

Boltzmann machine

Learning: **[Hinton, Sejnowski '86]**



Theoretically, general compute medium.
Via adjusting W, T, Q, B_h, B_v able to learn the underlying probability distribution of a given dataset.

However: practically not feasible

For applications only RBMs have been considered.

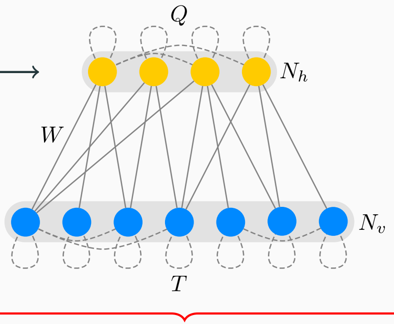
Riemann-Theta Boltzmann machine

How to change the status quo? **[Krefl, S.C., Haghighat, Kahlen '17]**

Keep the inner sector couplings non-trivial, but the machine solvable?

$\rightarrow$ Create the domain of state values.

Continuous valued $\in \mathbb{R}$ $\longrightarrow$



$\longleftarrow$ Continuous valued $\in \mathbb{R}$

$P(v) \equiv$ multi-variate gaussian (*too trivial*)

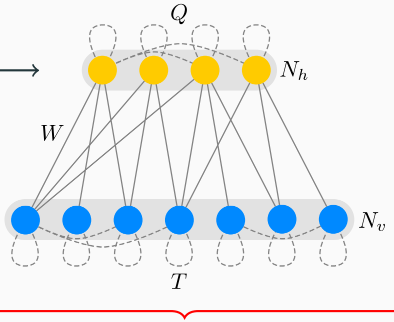
Riemann-Theta Boltzmann machine

How to change the status quo? **[Krefl, S.C., Haghighat, Kahlen '17]**

Keep the inner sector couplings non-trivial, but the machine solvable?

$\rightarrow$ Create the domain of state values.

"Quantized" $\in \mathbb{Z}$ $\longrightarrow$



$\longleftarrow$ Continuous valued $\in \mathbb{R}$

Something interesting happens

Under mild constraints on connection matrices (positive definiteness,...)

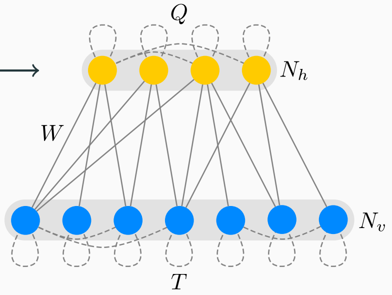
Riemann-Theta Boltzmann machine

How to change the status quo? **[Krefl, S.C., Haghighat, Kahlen '17]**

Keep the inner sector couplings non-trivial, but the machine solvable?

$\rightarrow$ Create the domain of state values.

"Quantized" $\in \mathbb{Z}$ $\longrightarrow$



$\longleftarrow$ Continuous valued $\in \mathbb{R}$

$$P(v) \equiv \sqrt{\frac{\det T}{(2\pi)^{N_v}}} e^{-\frac{1}{2}v^tTv - B_v^t v - \frac{1}{2}B_v^t T^{-1} B_v} \frac{\tilde{\theta}(B_h^t + v^t W | Q)}{\tilde{\theta}(B_h^t - B_v^t T^{-1} W | Q - W^t T^{-1} W)}$$

Closed form analytic solution still available!

Riemann-Theta Boltzmann machine

RTBM **[Krefl, S.C., Haghighat, Kahlen '17]**

Novel very generic probability density:

$$P(v) \equiv \sqrt{\frac{\det T}{(2\pi)^{N_v}}} e^{-\frac{1}{2}v^t T v - B_v^t v - \frac{1}{2}B_v^t T^{-1} B_v} \frac{\tilde{\theta}(B_h^t + v^t W | Q)}{\tilde{\theta}(B_h^t - B_v^t T^{-1} W | Q - W^t T^{-1} W)}$$

Damping factor

Riemann-Theta function

Mathematically striking:

$$\theta(z, \Omega) := \sum_{n \in \mathbb{Z}^{N_h}} e^{2\pi i \left(\frac{1}{2} n^t \Omega n + n^t z\right)}$$

Key properties: Periodicity, modular invariance, solution to heat equation, etc.

Note: Gradients can be calculated analytically as well so gradient descent can be used for optimization.

Applications

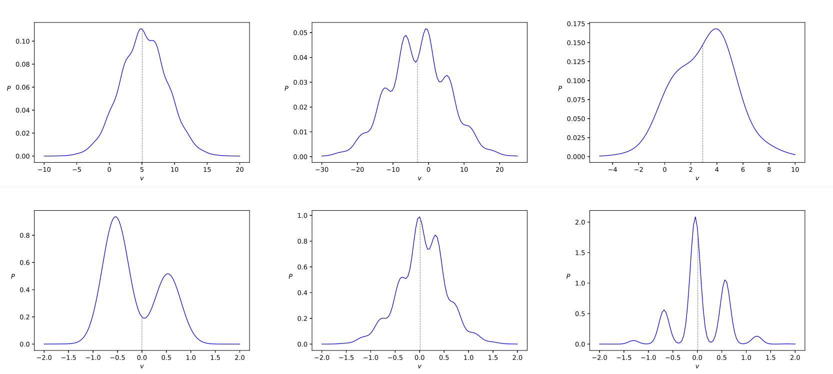
Riemann-Theta Boltzmann machine

In the next we show examples of RTBMs for

- Probability determination
- Data classification
- Data regression

Riemann-Theta Boltzmann machine

RTBM $P(v)$ examples: **[Krefl, S.C., Haghighat, Kahlen '17]**



For different choices of parameters (with hidden sector in 1D or 2D)

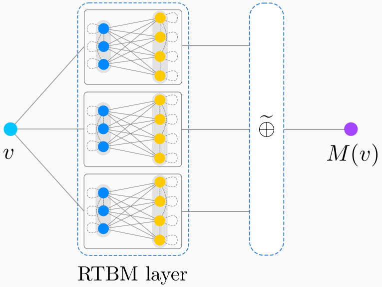
Riemann-Theta Boltzmann machine

Mixture model:

[Krefl, S.C., Haghighat, Kahlen '17]

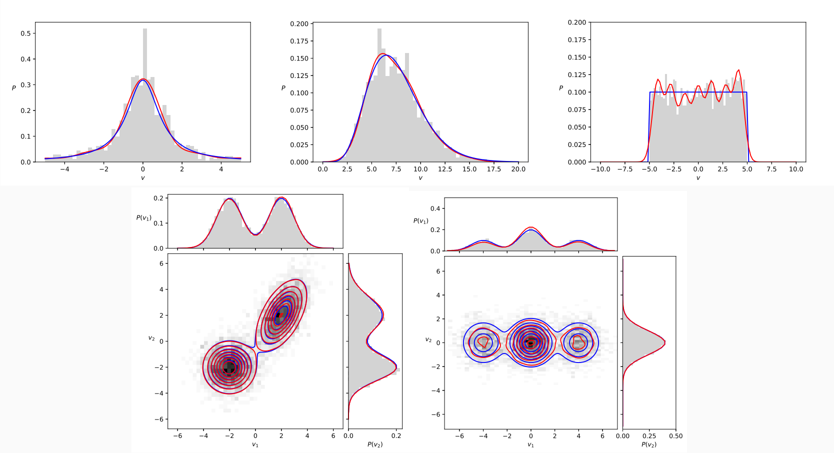
Expectation:

As long as the density is well enough behaved at the boundaries it can be learned by an RTBM mixture model.



Riemann-Theta Boltzmann machine

Examples: [Krefl, S.C., Haghighat, Kahlen '17]



Riemann-Theta Boltzmann machine

Feature detector:

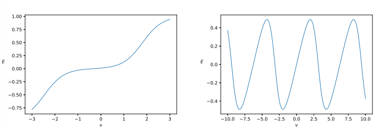
[Krefl, S.C., Haghighat, Kahlen '17]
Similar to [Krizhevsky '09]

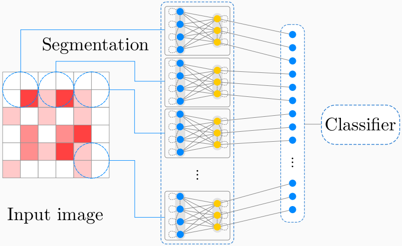
New:

Conditional expectations of hidden states after training

$$E(h_i|v) = -\frac{1}{2\pi i}\frac{\nabla_i\tilde{\theta}(v^tW + B_h^t|Q)}{\tilde{\theta}(v^tW + B_h^t|Q)}$$

The detector is trained in probability mode and generates a feature vector.





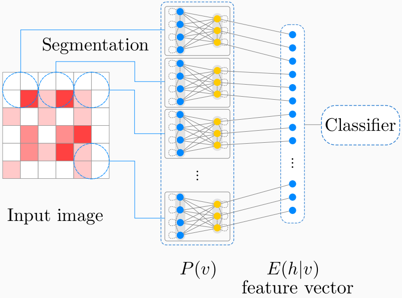
Feature detector example - jet classification

Jet classification:

[Krefl, S.C., Haghighat, Kahlen '17]

Data from [Baldi et al. '16, 1603.09349]

Descriminating jets from single hadronic particles and overlapping jets from pairs of collimated hadronic particles.



Data (images of 32x32 pixels)

- 5000 images for training
- 2500 images for testing

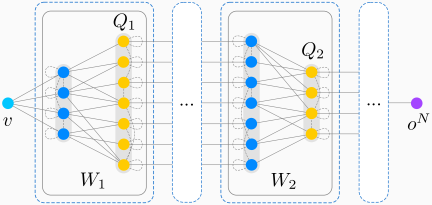
Classifier	Test dataset precision
Logistic regression (LR)	77%
RTBM feature detector + LR	83%

Riemann-Theta Boltzmann machine

Theta Neural Network: **[Krefl, S.C., Haghighat, Kahlen '17]**

Idea:

Use as activation function in a standard NN. The particular form of non-linearity is learned from data.

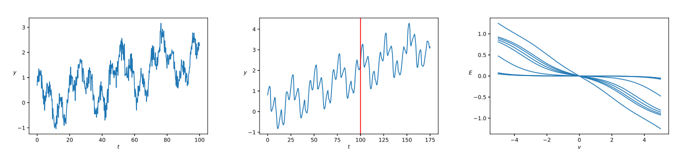


Key point:

smaller networks needed but Riemann-Theta evalution is expensive.

Example:

$$y(t) = 0.02t + 0.5\sin(t + 0.1) + 0.75\cos(0.25t - 0.3) + \mathcal{N}(0, 1)$$



$y(t)$ TNN fit TNN activations

Implementation

Implementation

Theta: Python machine learning framework for RTBMs and TNNs (with heavy lifting done by numpy, cython and C)

[riemann.ai/theta]

- Easy interface: Keras like definition of model.
- SGD and genetic optimizer out of the box.
  Easy integration of custom optimizers.
- Easy to extend functionality (object oriented)
- CPU based
  GPU, ,FPGA support in work
  Better math backend in work

Expected speedup will bring large scale applications in reach.

Thank you!