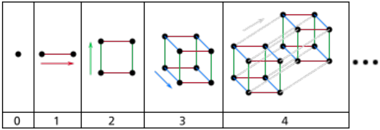
Complicated parameter spaces and machine learning

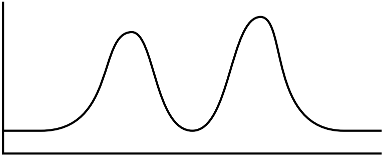
Raymundo Ramos
Seoultech
(based on: A. Hammad, M. Park, R.R., P. Saha, arXiv:2207.09959)

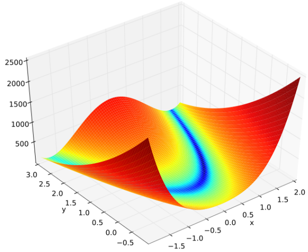
AI and Quantum Information Applications in Fundamental Physics
February 15, 2023

What makes a parameter space complicated?

- Several dimensions
- Multimodality
- Curved degeneracy
- ...







What makes our work more complicated?

- Some high energy physics calculations (HEPC) take **a very long time/too much computational power**
  - Simulations
  - Matrix diagonalization
  - Amplitudes with many terms and corrections
- More parameters:

exponential increase in required points $\times$ **time required per point**

How we want to approach this problem

- Neural networks (NN) as generic function approximators
- Useful when training a NN could be more efficient than passing every single point through the HEPC
- Design a process where the accuracy of the NN becomes proportional to our interest in sampled regions:
  - spend, relatively, **more time sampling regions of interest**,
  - *just enough time* for low importance regions

Follow an iterative process similar to others proposed in:

Ren, Wu, Yang and Zhao [arXiv:1708.06615];
Caron, Heskes, Otten and Stienen [arXiv:1905.08628];
Goodsell and Joury [arXiv:2204.13950]

An iterative process

0. L_0: sizable but not too large: $(L_0, Y_0) \rightarrow$ NN-training

---

1. L: large set of points:

$$L \rightarrow \text{NN-prediction} \rightarrow \hat{Y}(L)$$

2. Select an smaller set

$$(L, \hat{Y}(L)) \rightarrow \textbf{selection criteria} \rightarrow (K, \hat{Y}(K))$$

3. Get the correct results from the HEPC

$$K \rightarrow \text{HEPC} \rightarrow Y(K)$$

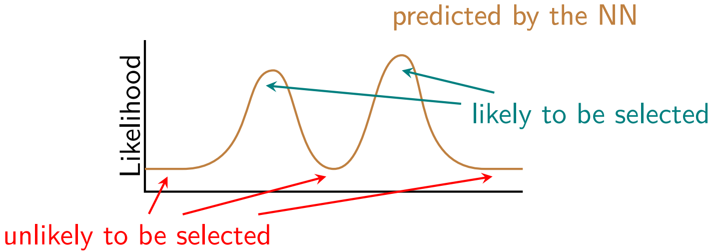
4. Train with set K and true results $Y(K)$

$$(K, Y(K)) \rightarrow \text{NN-training}$$

Selection of points for HEPC – Regression

We want to pass a set of meaningful points to the HEPC.

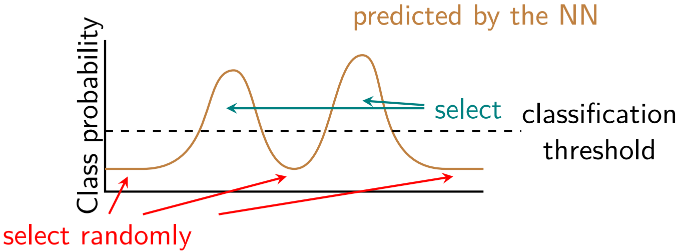
- **Highest** predicted likelihood/**lowest** predicted χ^2
  - But keep **diversity** of observables/likelihood
- Points predicted with low likelihood/high χ^2 may be included as part of some **rectifying strategy**.
- Fraction of random points to find new regions



Selection of points for HEPC – Classification

We want to pass a set of meaningful points to the HEPC.

- **Highest** probability of being allowed
  - But keep **diversity** of points in and out of region of interest
- Points predicted with low probability of being allowed may be included as part of some **rectifying strategy**.
- Fraction of random points to find new regions



Selection of points for training

Training is also time consuming

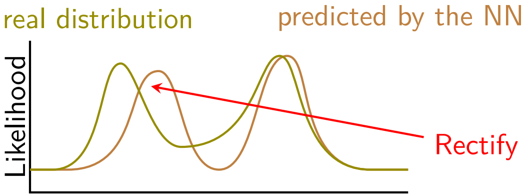
Required time depends on:

- **epochs**
- number of **hidden layers**
- number of **nodes**
- number of **points used for training**

We have to be smart about the points used for training

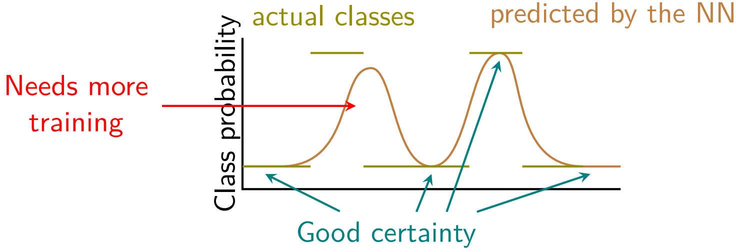
Selection of points for training – Regression

- wrongly predicted as high likelihood: **rectify inaccurate predictions**
- What about points wrongly predicted with low likelihood/high χ^2.
  - This needs a well thought strategy
  - There is a chance that will be corrected by additional random points.

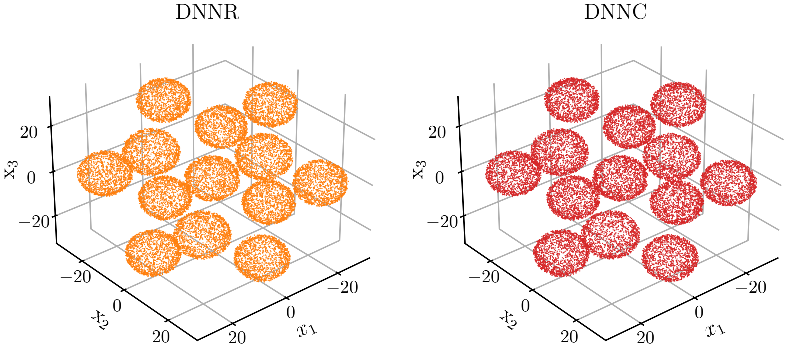


Selection of points for training – Classification

- **True allowed**: Good certainty. These we are interested in
- **False allowed**: Confusing. These we want to correct
- **False excluded**: Confusing. These we want to correct
- *True excluded*: Good certainty. The region we care the least



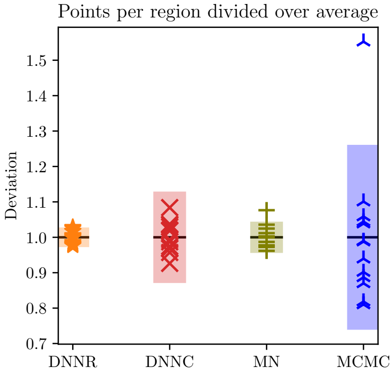
Applied to toy model, region coverage (20k points)



$$O_{\mathrm{3d}} = \left[2 + \cos\left(\frac{x_1}{7}\right)\cos\left(\frac{x_2}{7}\right)\cos\left(\frac{x_3}{7}\right)\right]^5 = 100 \pm 20$$

4 hidden layers (ReLU), 1000 epochs, Adam, loss: (MAE, Binary cross-entropy), output layer activation: (linear, sigmoid)

Applied to toy model, deviations



Average deviation in 10 runs. Markers show deviation for best result.

- DNNR: regressor
- DNNR: classifier
- MN: MultiNest (`pyMultiNest`)
- MCMC: Markov Chain Monte Carlo (`emcee`)

Boosting initial convergence

During the initial steps, predictions should be expected to be **mostly wrong**

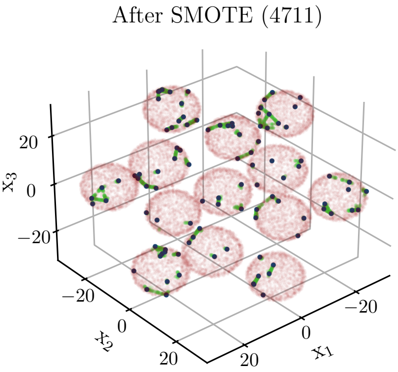
Many options to improve initial convergence:

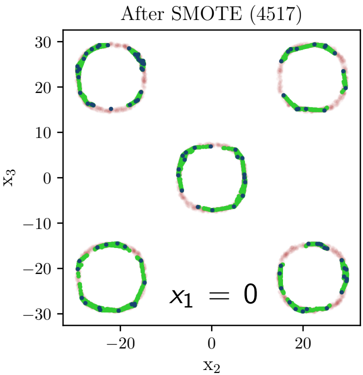
- **Naive/Brute force**: run more points to collect usable points
- Sample more points **around** known points in the target region
- Sample points **between** known points (**S**ynthetic **M**inority **O**versampling **Te**chnique, SMOTE) [Chawla et al, arXiv:1106.1813]

If these techniques work they should be needed **only in the first few iterations**.

Boosting initial convergence

Suggest new points using 3 nearest neighbors

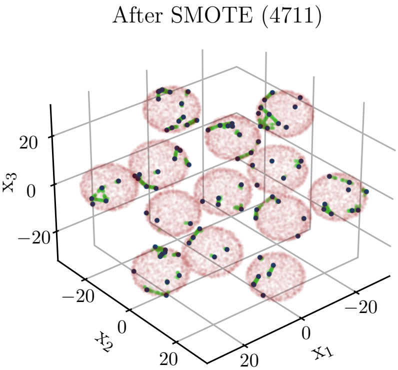


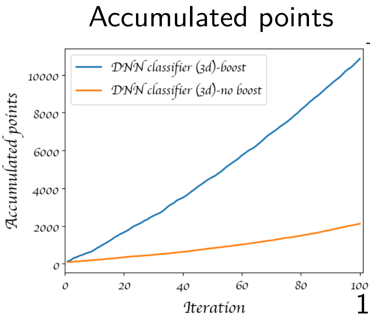


efficiency: 50% 90%

Boosting initial convergence

Suggest new points using 3 nearest neighbors





+10k

∼2k

100 iterations

Learning the Higgs signal strength in the 2HDM

The two Higgs doublet models (2HDM) [Lee, PRD **8**, 1226] are extensions of the standard model scalar sector

$$\phi_1 = \begin{pmatrix} \eta_1^+ \\ (v_1 + h_1 + i h_3)/\sqrt{2} \end{pmatrix}, \qquad \phi_2 = \begin{pmatrix} \eta_2^+ \\ (v_2 + h_2 + i h_4)/\sqrt{2} \end{pmatrix}.$$

Z_2 symmetry: $(\phi_1, \phi_2) \to (\phi_1, -\phi_2) \to$ No FCNC.
Softly broken by m_{12}^2 [Glashow, Weinberg, PRD **15**, 1958 (1977)]

$$\begin{aligned} V_\phi =& m_{11}^2(\phi_1^\dagger\phi_1) + m_{22}^2(\phi_2^\dagger\phi_2) - \left[m_{12}^2(\phi_1^\dagger\phi_2) + \text{h.c.}\right] + \lambda_1(\phi_1^\dagger\phi_1)^2 \\ &+ \lambda_2(\phi_2^\dagger\phi_2)^2 + \lambda_3(\phi_1^\dagger\phi_1)(\phi_2^\dagger\phi_2) + \lambda_4(\phi_1^\dagger\phi_2)(\phi_2^\dagger\phi_1) \\ &+ \frac{1}{2}\left[\lambda_5(\phi_1^\dagger\phi_2)^2 + \text{H.c.}\right] \end{aligned}$$

$$\tan\beta \equiv \frac{v_2}{v_1} \quad \text{with} \quad v = \sqrt{v_1^2 + v_2^2} \sim 246\text{GeV}$$

Numerical scan

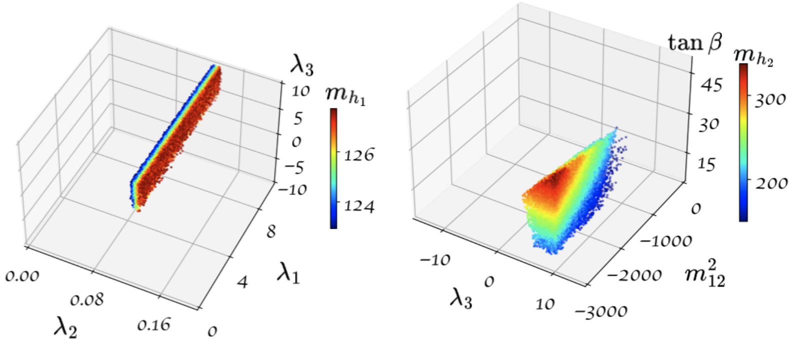
Scanned parameters and ranges

$$0 \leq \lambda_1 \leq 10, \quad 0 \leq \lambda_2 \leq 0.2, \quad -10 \leq \lambda_3 \leq 10, \quad -10 \leq \lambda_4 \leq 10,$$
$$-10 \leq \lambda_5 \leq 10, \quad 5 \leq \tan\beta \leq 45, \quad -3000 \leq \frac{m_{12}^2}{\mathrm{GeV}^2} \leq 0\,,$$

Tools used

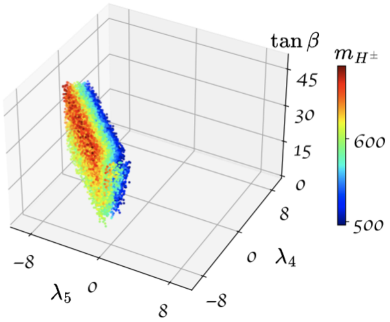
- `SPheno` to obtain the mass spectrum
- `HiggsBounds` to obtain limits on the Higgses [Bechtle et al, arXiv:1507.06706]
- `HiggsSignals` to obtain a χ^2 for the signals and mass of the Higgs [Bechtle et al, arXiv:1403.1582]

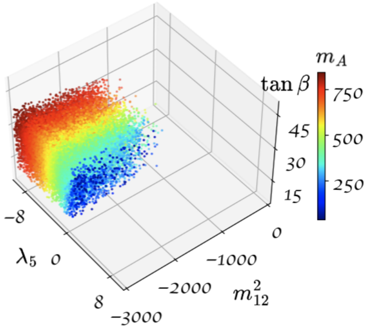
Numerical scan results



4 hidden layers, 100 nodes, ReLU, 1000 epochs per iteration
optimizer: Adam, loss: binary cross-entropy

Numerical scan results





What to do with this?

This process could be good for

- Adjusting complicated allowed regions
- Reduce the amount of calls to a time consuming calculation
- Compare against an ever increasing amount of experimental tests

This process could be *great* for

- A study where we already have a sense of the parameter space
  - Update limits to new data
  - Test future expectations of a model
- Anything where a precise and fast estimation of observables/likelihood could be employed (*after the model has been trained enough*)

The code

Implementation using `tensorflow`

- https://github.com/AHamamd150/MLscanner

Where do we want to go next?

Well known example of complicated space: phase space integration

$$p_a + p_b \to p_1 + p_2 + \ldots + p_n$$

$$\int \prod_{i=1}^{n} d^4 p_i \delta(p_i^2 - m_i^2) \delta^4(p_a + p_b - p_1 - p_2 - \ldots - p_n)$$

$3n - 4$ integration variables

Add the complexity of the squared amplitud $|M_{a+b\to 1+2+\ldots+n}|^2$

We usually look for: accurate estimation of integral, (unweighted) event generation, accurate simulation of background/signal

Inspiration from previous works

- ANN as event generator.
  (Klimek, Perelstein [arXiv:1810.11509]; Chen, Klimek, Perelstein [arXiv:2009.07819])
- ML training with amplitude values.
  (Bishara, Montull [arXiv:1912.11055]; Maître, Truong [arXiv:2107.06625])
- Normalizing flows for phase space integration.
  (Gao, Höche, Isaacson, Krause, Schulz [arXiv:2001.10028])
- Normalizing flows (INN) for multichannel integration.
  (Bothmann, Janßen, Knobbe, Schmale, Schumann [arXiv:2001.05478];Heimel, Winterhalder, Butter, Isaacson, Krause, Maltoni, Mattelaer, Plehn [arXiv:2212.06172])
- . . . (and references found in the works above)

Thanks for listening!