New Physics Searches Using Unsupervised Learning in High Energy Physics

Seminar I - 1st year, 2nd cycle

author: Jan Gavranovič
advisor: prof. dr. Borut Paul Kerševan

University of Ljubljana
Faculty of Mathematics and Physics

March 17, 2021

Overview

1. Machine learning and anomaly detection
2. Searching for new physics
3. Introduction to neural networks
4. Classification without labels
5. Extending the bump hunt
6. Dijet resonance search
7. Conclusion

Definition of machine learning

Definition:

A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

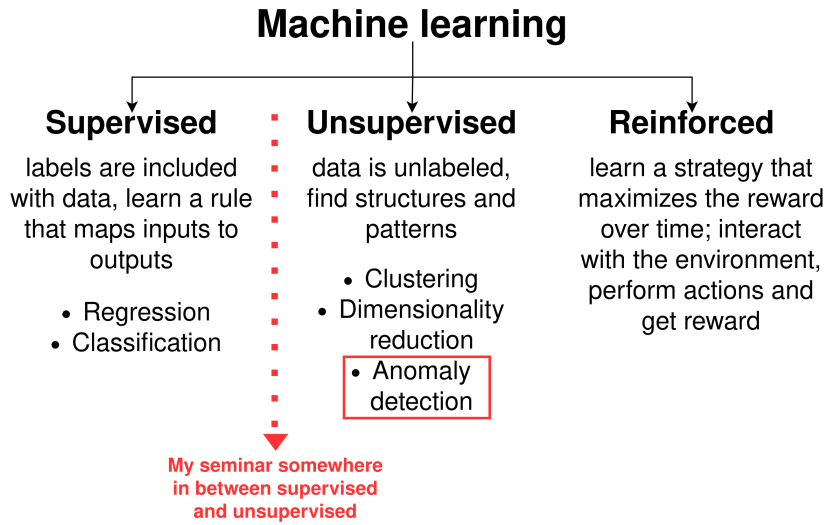
A well defined learning problem needs:

1. class of tasks (e.g predict next result),
2. measure of performance to be improved (e.g. accuracy, χ^2, entropy),
3. source of experience (e.g. measurements, events, inputs).

Example: autonomous driving

$T \Rightarrow$ driving on a highway using vision sensors
$P \Rightarrow$ average distance traveled before an error
$E \Rightarrow$ large amount of sequences of images, steering commands recorded from human drivers, satellite images of roads and other relevant information when driving a car

Types of machine learning



Anomaly detection

- Detect data that deviates from the norm.
- The algorithm finds an *optimal* metric to separate the anomaly from the bulk $\Rightarrow$ defines an anomaly.
- Objective of anomaly detection: learn what normal data looks like and then use that to detect abnormal instances when they are shown to the algorithm.

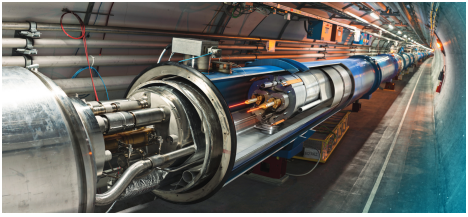
[1]

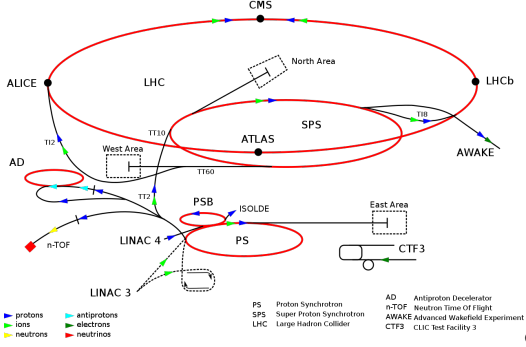
Anomaly detection use cases:

- Fraud detection (e.g. fraudulent credit card transactions)
- Medicine (e.g. unusual symptoms)
- Detecting measurement errors
- Detecting defective products in manufacturing
- Looking for outliers in datasets
- Anywhere where looking at unusual observations is relevant

Searching for new physics

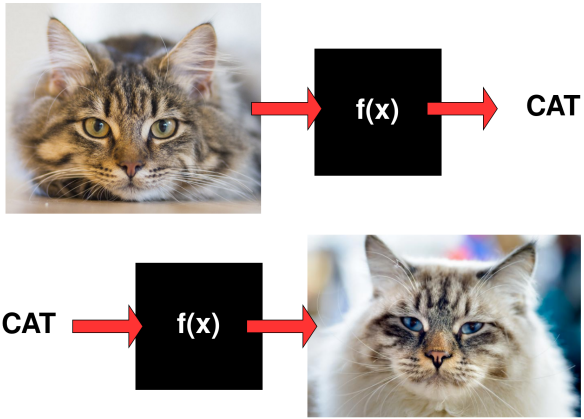
- Teams at ATLAS and CMS detectors at LHC discovered the Higgs boson in 2012.
- Last undiscovered particle of the Standard model was found.
- Standard model is not a complete theory.
- Does not describe dark matter and energy, gravity, neutrino masses.
- There are numerous theories that expend the Standard model.
- How to search for new physics with all of these competing models and ideas?
- A universal and model independent technique is needed.





A brief introduction to neural networks

- Most simple: deep feedforward neural networks $\Rightarrow$ supervised learning.
- Approximate some function $f^*(\boldsymbol{x})$.
- Training set: examples $\boldsymbol{x}$ with labels y.
- Defines a mapping $h = f(\boldsymbol{x}; \boldsymbol{\theta})$ and tries to learn the parameters $\boldsymbol{\theta}$ that give a correct result $h \approx y$.
- Can be used as a classifier that maps an input vector $\boldsymbol{x}$ to a category: $y \in \{S, B\}$.



Example: classification and generation of images.

A brief introduction to neural networks

- Composed of many neurons arranged in layers.
- A layer is given by a linear operation and a nonlinear transformation:

$$h = g(\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}) \,,$$

where g is most commonly a ReLU function

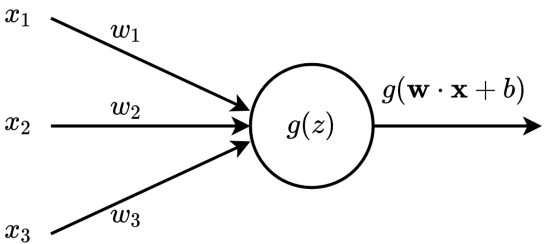
$$g(z) = \max\{0, z\} \,.$$

- Need a loss function L to calculate the model error, this is the function we want to minimize.
- Mean squared error for regression:

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{m} \left(h(\boldsymbol{x}_i) - y_i\right)^2 \,.$$

- Binary cross entropy for binary classification:

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^{m} [y_i \log h(\boldsymbol{x}_i) + (1 - y_i) \log(1 - h(\boldsymbol{x}_i))] \,.$$



Basic unit of a neural network (neuron).

Neural networks for binary classification

- How to minimize a loss function? Move in the direction of the negative gradient:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{\theta}) \,,$$

where the training set is split into mini-batches

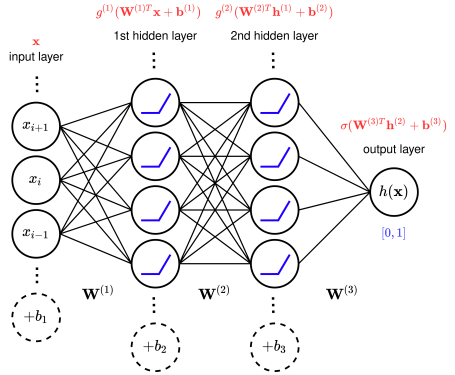
$$MB = \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), ..., (\boldsymbol{x}^{(m)}, y^{(m)})\}.$$

- If we want to predict the value of a binary variable y, we use a sigmoid output:

$$\sigma(z) = \frac{e^z}{1 + e^z} \in [0, 1] \,.$$

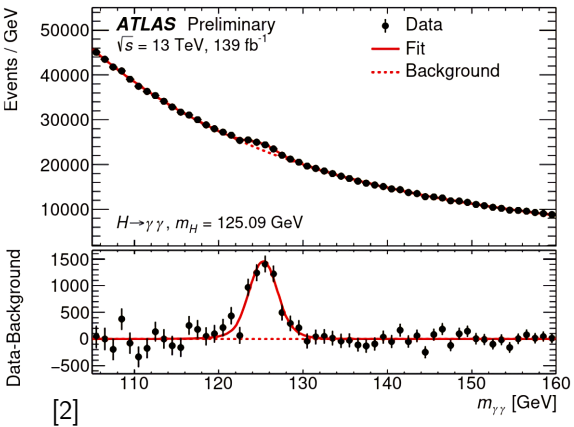
NNs have two modes of operation:

1. Forward propagation (predicting).
2. Backward propagation (learning).



Bump hunt

- One of the most traditional simulation-independent anomaly detection strategies.



[2]

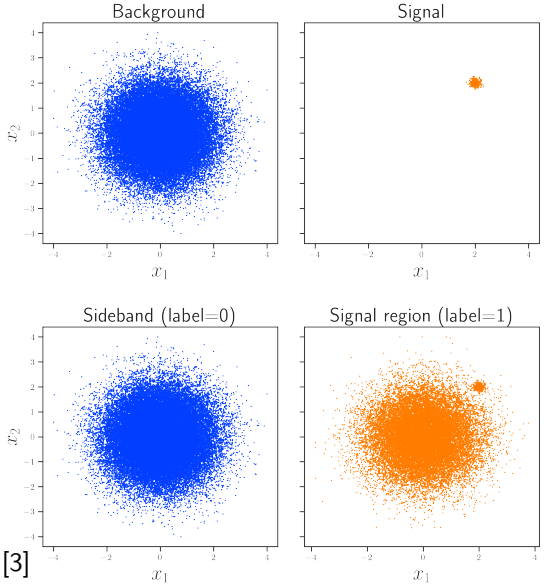
- Identifies a potential deviate (anomaly) from background.

Two regions of phase space are considered:

1. Signal region: a kinematic region where the signal might be present.
2. Sideband regions: kinematically adjacent regions to the signal region, where there is little signal contamination.

- A smooth background fit is performed in the sideband regions and extrapolated to the signal region.
- Excess over predicted background means a resonance peak is present in data.

Classification without labels



[3]

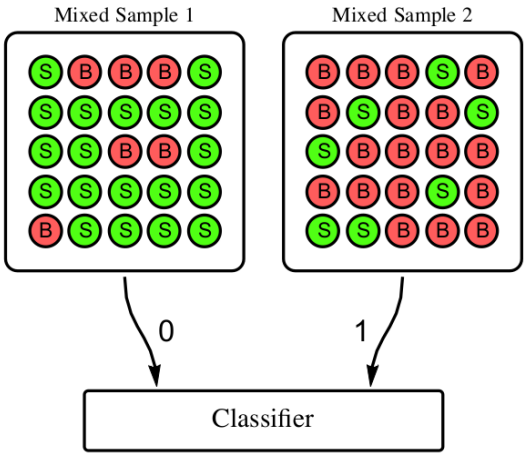
- Classification with labels (fully supervised): distinguish signal (S) from background (B) using a classifier $h : \boldsymbol{x} \to \mathbb{R}$ with observables $\boldsymbol{x}$.
- Anomaly detection: show an algorithm a sample with small signal (S) and large background (B), separate outliers from the bulk
- Classification without labels (weak supervision): use partial label information for telling the two samples (0 and 1) apart.

Classification without labels (CWoLa)

- Consider two mixed samples:

$$p_{M_1}(\boldsymbol{x}) = f_1 p_S(\boldsymbol{x}) + (1 - f_1) p_B(\boldsymbol{x}) \,,$$
$$p_{M_2}(\boldsymbol{x}) = f_2 p_S(\boldsymbol{x}) + (1 - f_2) p_B(\boldsymbol{x}) \,,$$

  where $0 \leq f_2 \leq f_1 \leq 1$ are the signal fractions.
- Only have samples from distributions p_{M_1} and p_{M_2} with labels M_1 and M_2 and no longer have S and B labels or class proportions f_1 and f_2.
- Train a classifier to discriminate mixed sample M_1 from mixed sample M_2.
- An optimal classifier, trained to distinguish M_1 from M_2, is also optimal for distinguishing S from B.

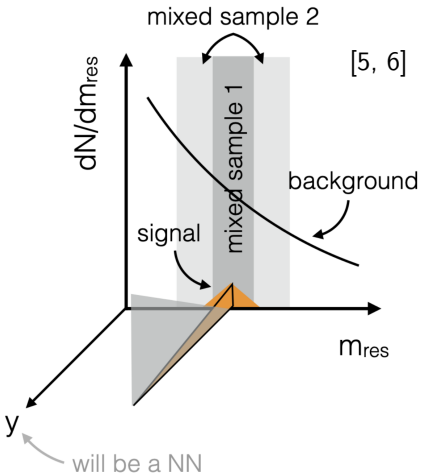


[4]

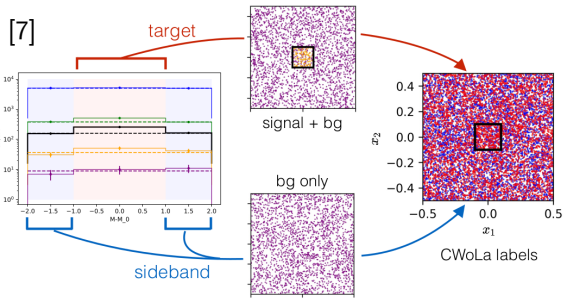
Extending the bump hunt

Searching for new heavy particle (resonance) decays in a model-independent way.

1. For a mass hypothesis m_0 identify an observable m_{res} in which signal is resonant.
2. Select auxiliary variables **x** used for signal selection.
3. A background model $f(m_{res})$ is needed for m_{res}.
4. Define a signal region in a window around $m_0 \Rightarrow M_1$.
5. Define sideband regions $\Rightarrow M_2$.
6. Train a classifier (NN) to discriminate sideband regions from a signal region using features **x** to calculate the prediction y.
7. Select a fraction ϵ of the most signal like test events as given by the classifier.
8. Search for an excess in the signal region of m_{res} distribution. Use sidebands for background determination using $f(m_{res})$.
9. Repeat for a series of m_0 resonance mass hypotheses.



An illustrative toy example



- Let auxiliary features be a two dimensional vector $\boldsymbol{x} = (x_1, x_2)$ where $x_1, x_2 \sim U(-\frac{1}{2}, \frac{1}{2})$.
- Define signal and background regions as

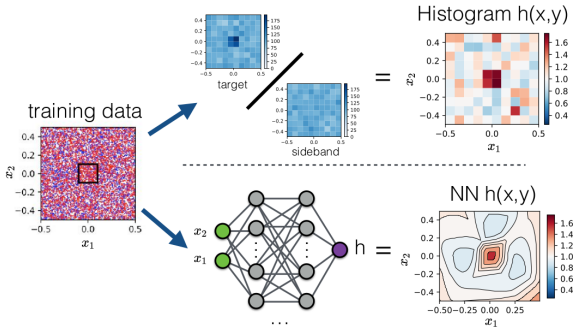
$$SR = (-w/2 < x < w/2, -w/2 < y < w/2)\,,$$
$$BR = (-1/2 < x < 1/2, -1/2 < y < 1/2)\,.$$

- Let there be three mass bins that corespond to a signal region and two sidebands.
- The signal is only present in the middle bin.
- A neural network is trained on (x_1, x_2) values using mixed labels.

An illustrative toy example

- A deep feedforward neural network with three hidden layers was trained.
- The data was split into three equal sets: training set, validation set and test set.
- Efficiency ϵ: a fraction of events with a given neural network value or higher.
- Signal tagging $\Rightarrow$ low efficiency parameter values.
- Background was estimated by fitting a line to the mass sidebands.
- Significance estimation as $S \approx (n_0 - \hat{n}_b)/\sqrt{\hat{n}_b}$, where n_0 is the number of events in the signal bin and $\hat{n}_b$ is the background estimation.
- Result: significance increase from 3σ to above 10σ.

[7]



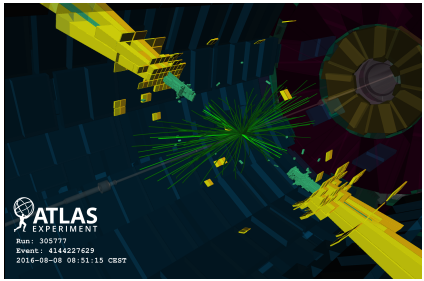
Dijet resonance search

- A dijet resonance decay:

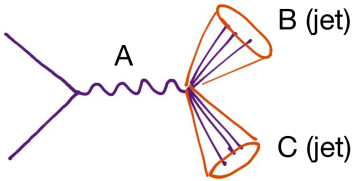
$$pp \to A \to BC \to JJ\,.$$

- $m_A \sim \mathcal{O}(1\text{ TeV})$ and $m_B, m_C \sim \mathcal{O}(100\text{ GeV})$.
- $\sqrt{s} = 13$ TeV collision data set of 139 fb^{-1} recorded by the ATLAS detector at LHC in Run 2.
- Fully data-driven machine-learning-enhanced anomaly detection search.
- Used masses of the two jets $\boldsymbol{x} = (m_1, m_2)$ as features for classification, where $m_1 \geq m_2$.
- Bump hunt done on dijet invariant mass distribution m_{JJ} of the two leading jets.

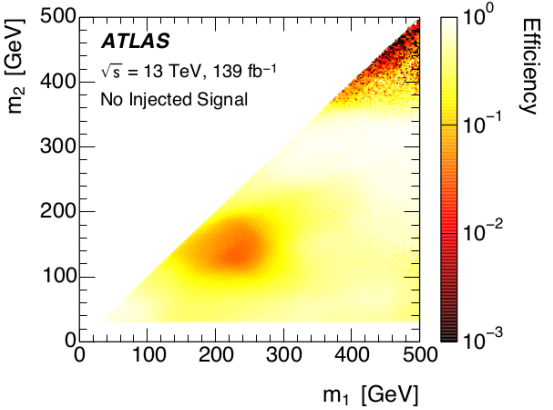
[8]

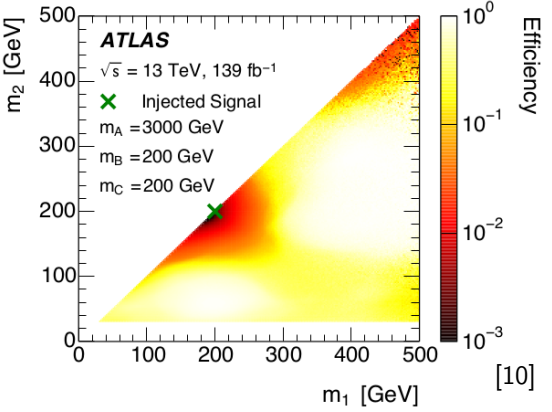


[9]



Neural network performance



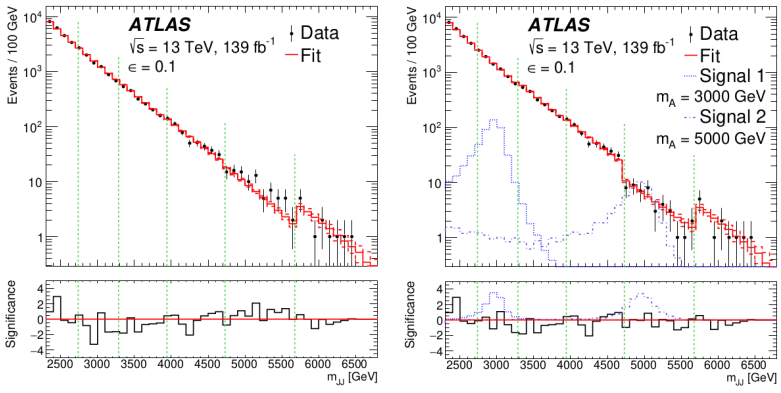


[10]

- Neural network used: three hidden layers with 64, 32 and 8 neurons.
- The method was validated using injected simulated events: $W' \to WZ$.
- With no signal injected, no evidence for excess was found.
- Learns to tag the signal and enhances a bump in the invariant mass spectrum.

Fits in signal regions

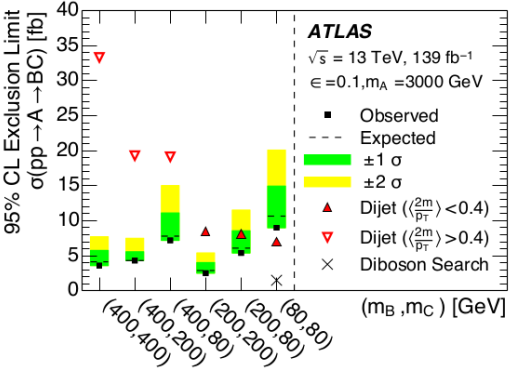
[10]

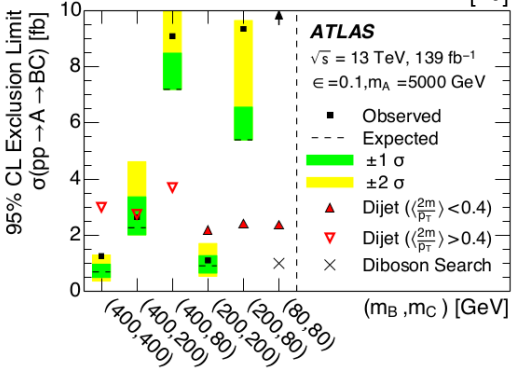


- Mixed samples constructed using eight regions for different m_0 hypotheses.
- Parametric function fit after applying an event selection based on neural network efficiency in the given signal region.
- Fitting was done using an iterative procedure using three fit functions until a good enough χ^2 is achieved.

Limits

[10]





- Injected signal model $W' \to WZ$ can be used to set limits on the production cross section.
- Illustrates the sensitivity of the search to the (m_A, m_B, m_C) parameter space.
- 95% confidence level upper limits on the cross section for $m_{B,C}$ signal models were calculated.

Conclusion

Summary:

- Anomaly detection combined with supervised machine learning provides a way to search for new physics that is model-independent and data-driven.
- Neural network classifier that distinguishes signal from background without having S, B labels.
- Enhances a bump in the invariant mass spectrum $\Rightarrow$ bump hunt more successful.
- Need a resonant variable m_{res} (e.g. invariant mass m_{JJ}) and additional features $\boldsymbol{x}$ (e.g. observables describing the jet substructure).

Future:

- Addition of other jet features.
- Deeper neural network architectures.
- Other anomaly detection algorithms.

References

[1] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[2] *Review of Particle Physics*. URL: https://pdg.lbl.gov/2020/html/computer_read.html.

[3] CWoLa Hunting: Extending the Bump Hunt with Machine Learning. URL: https://indico.cern.ch/event/558411/contributions/3420889/attachments/1845259/3027242/NPKI_CWoLa_Hunting.pdf.

[4] Eric M. Metodiev, Benjamin Nachman, and Jesse Thaler. *Classification without labels: Learning from mixed samples in high energy physics*. 2017. arXiv: 1708.02949 [hep-ph].

[5] *Machine learning qualitatively changes the search for new particles*. URL: https://atlas.cern/updates/briefing/search-new-particles-machine-learning.

[6] Jack Collins, Kiel Howe, and Benjamin Nachman. "Anomaly Detection for Resonant New Physics with Machine Learning". In: *Physical Review Letters* 121 (2018). ISSN: 1079-7114. URL: http://dx.doi.org/10.1103/PhysRevLett.121.241803.

[7] Jack Collins, Kiel Howe, and Benjamin Nachman. "Extending the search for new resonances with machine learning". In: *Physical Review D* 99.1 (Jan. 2019). ISSN: 2470-0029. URL: http://dx.doi.org/10.1103/PhysRevD.99.014038.

[8] *Particle-hunting at the energy frontier*. URL: https://atlas.cern/updates/briefing/particle-hunting-energy-frontier.

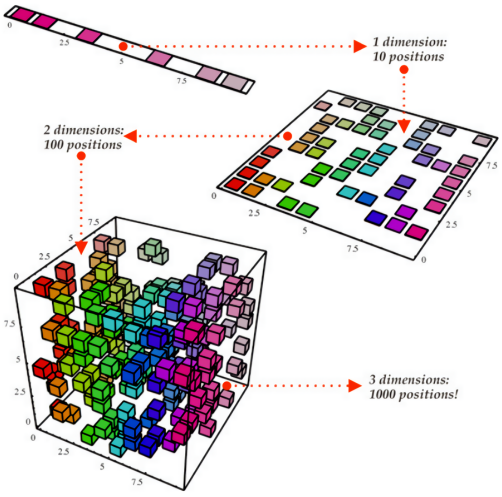
[9] ATL-PHYS-SLIDE-2020-235. URL: http://cds.cern.ch/record/2724057.

[10] ATLAS Collaboration. *Dijet resonance search with weak supervision using $\sqrt{s} = 13$ TeV pp collisions in the ATLAS detector*. 2020. arXiv: 2005.02983 [hep-ex].

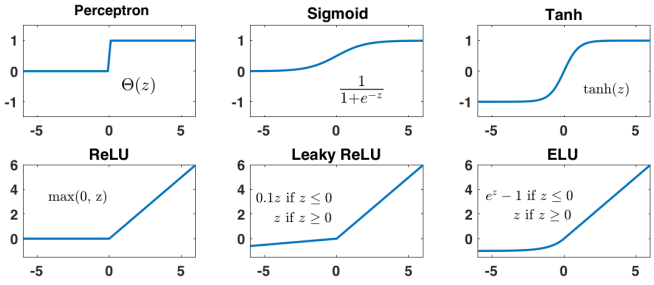
Thank you!

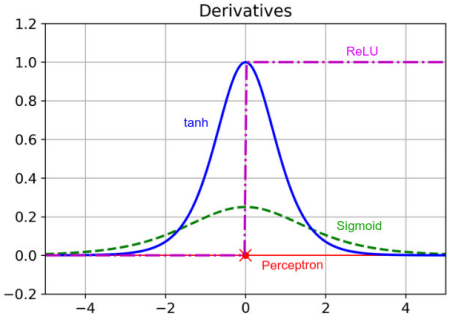
Backup slides

Curse of dimensionality



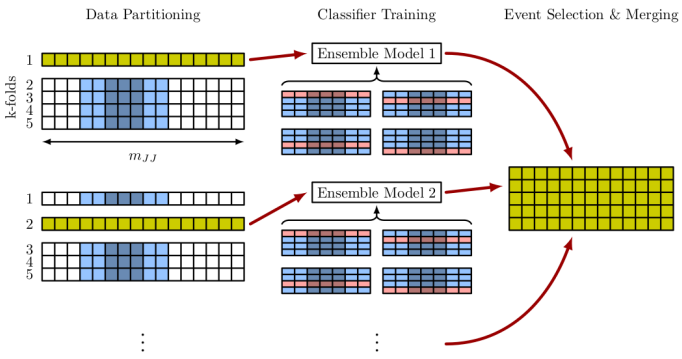
Activation functions





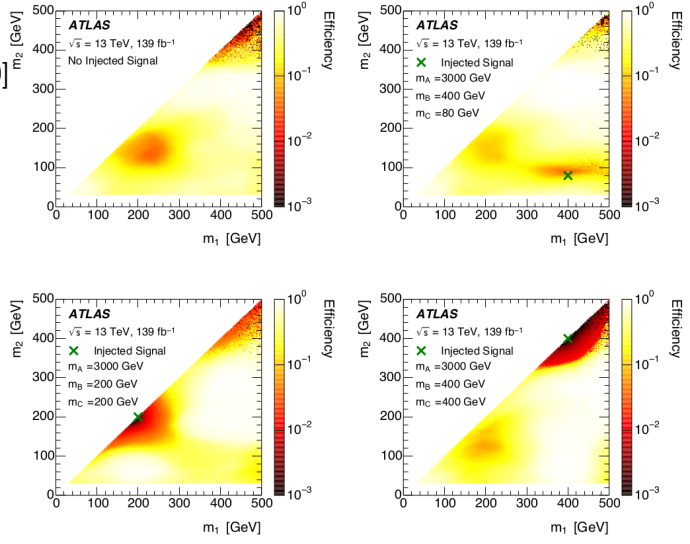
Cross validation

[7]



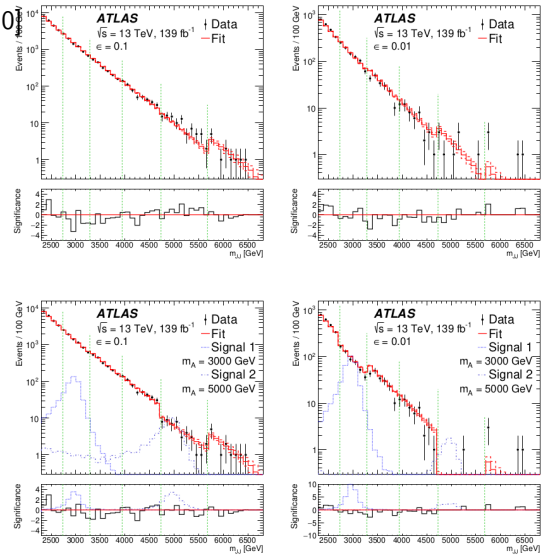
Different injected signals

[10]



Efficiencies $\epsilon = 0.1$ and $\epsilon = 0.01$

[10]



Fitting was done using an iterative procedure using three fit functions:

$$\frac{dn}{dx} = p_1(1-x)^{p_2-\xi_1 p_3} x^{-p_3} ,$$

$$\frac{dn}{dx} = p_1(1-x)^{p_2-\xi_1 p_3} x^{-p_3+(p_4-\xi_2 p_3-\xi_3 p_2)\log(x)} ,$$

$$\frac{dn}{dx} = p_1 x^{p_2-\xi_3} e^{-p_3 x+(p_4-\xi_2 p_3-\xi_3 p_2)x^2} ,$$

where $x = m_{JJ}/\sqrt{s}$, p_i are fit parameters and ξ_i are chosen such that p_i are not correlated