University *of Ljubljana*
Faculty *of Mathematics and Physics*

Department of Physics

Seminar I - 1st year, 2nd cycle

New Physics Searches Using Unsupervised Learning in High Energy Physics

Author: Jan Gavranovič

Advisor: prof. dr. Borut Paul Kerševan

Ljubljana, March 2021

Abstract

The seminar presents unsupervised machine learning, specifically its use for anomaly detection in high energy physics. First, a description of machine learning and anomaly detection is given. A method of classification without labels is presented in the context of new physics searches. Deep neural networks and their use for binary classification are then discussed. What follows is an illustrative example that is meant to showcase the full method, which combines classification without labels and neural networks with bump hunt and is used for distinguishing between a signal region and sideband regions. At the end, a dijet resonance search using $\sqrt{s} = 13$ TeV pp collisions in the ATLAS detector is presented.

Contents

1 Introduction

1.1 Definition of machine learning

Machine learning is the science of programming computers so they can learn from data. A more rigorous and widely used definition of learning is as follows [1]:

> 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.

To have a well defined learning problem we must specify: the class of tasks (e.g. identify appropriate signals, predict next result), the measure of performance to be improved (e.g. accuracy, χ^2, entropy) and the source of experience (e.g measurements, events, inputs). The examples that the system uses to learn are called the training set and the process of learning from this set is called training. The definition is very broad and can be used to include any computer program that improves its performance at some task through learning from data. An example of this is autonomous driving. Task T is driving on a highway using vision sensors. Performance measure P can be average distance traveled before an error. Training experience E is in this case a large amount of sequences of images, steering commands recorded from human drivers, satellite images of roads and other relevant information when driving a car.

1.2 Types of machine learning

There are three major categories of machine learning systems. The distinction is made whether or not they are trained with human supervision. These three categories are:

- Supervised learning: the training set includes the desired solutions, called labels. The algorithm tries to learn a general rule that maps inputs to outputs.
- Unsupervised learning: the training data is unlabeled, leaving the learning algorithm to find structures and patterns in data by itself.
- Reinforcement learning: the learning algorithm interacts with the environment, selects and performs actions and gets rewards in return. It must learn by itself what is the best strategy that maximizes the reward over time (e.g robots learning how to walk or how to play video games).

1.3 Anomaly detection

One of the important unsupervised tasks is anomaly detection or outlier detection. The task of such algorithms is to detect data that deviates from the norm. In unsupervised learning, the algorithm is supposed to find the optimal metrics to separate the anomaly from the bulk of data, thereby defining what is an anomaly. While training the system is shown mostly ”normal” examples and learns to recognize them. When a new instance deviating from the normal examples observed is shown, the algorithm can tell it apart from others and tag it as an anomaly (Figure 1). Anomaly detection is used in fraud detection (e.g. fraudulent credit card transactions), medicine (e.g. unusual symptoms), detecting measurement errors from sensors, detecting defective products in manufacturing, looking for outliers in datasets or any application where looking at unusual observations is relevant.

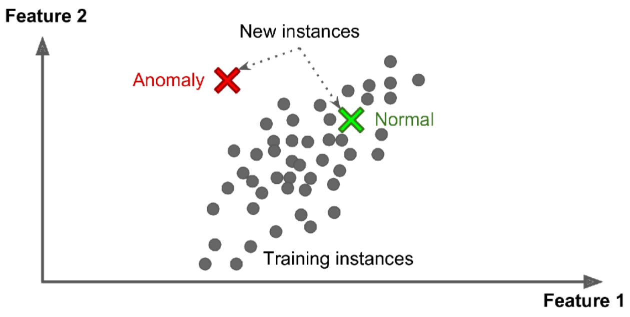


Figure 1: The objective of anomaly detection is to learn what normal data looks like and then use that to detect abnormal instances when they are shown to the algorithm [2].

2 Searching for new physics

The goal of high energy physics is to determine the fundamental building blocks of matter and to characterize the interactions between these particles. In order to achieve this goal experiments collide particles with high momentum to produce new elementary particles and study their interactions and decays. Currently the Large Hadron Collider (LHC) at CERN is the world’s largest and most powerful particle accelerator [3]. The teams at ATLAS [4] and CMS [5] detectors at LHC discovered the Higgs boson in 2012, which was the last undiscovered particle of the Standard model.

This however does not mean that the theory is complete and that the searching for new discoveries has stopped since then. While the Standard model has been very successful, it is not a complete theory. It does not describe dark matter, gravity and neutrino masses in addition to various other more specific problems. There are numerous suggestions [6] on how to expend the theory beyond the Standard model (BSM). A question arises how would we even search for new physics with all of these competing models and ideas. It is clear that a universal and model independent technique is needed.

A commonly used technique to search for new particles is the ”bump hunt” method. The method identifies a potential deviate (anomaly) from background using histogram comparisons. In the bump hunt a kinematic region where the signal could be present is first identified. This region of phase space is called a signal region. After that the kinematically adjacent regions to the signal region, where there is little signal contamination expected, are defined as sideband regions. 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. It works very well because sharp structures in invariant mass spectra are not common in background processes, which typically have a smooth and slowly varying distribution. With this method the background can be estimated directly from data by fitting a function away from the resonance (sideband) and then extrapolating it to the signal region to find a potentially new resonance. This was the method used to find the Higgs boson. This seminar will present a new model-independent anomaly detection technique that uses unsupervised machine learning [7]. It extends the bump hunt and is being used to search for physics beyond the Standard model.

3 Classification without labels (CWoLa)

The goal of classification is to distinguish two processes from each other, specifically signal S and background B. Let $\boldsymbol{x}$ be a list of observables that can be used to separate signal from background. The probability distributions of signal and background are $p_S(\boldsymbol{x})$ and $p_B(\boldsymbol{x})$. A classifier is defined as $h : \boldsymbol{x} \to \mathbb{R}$ and returns higher values of h for signal and lower values for background. It can be shown from the Neyman-Pearson lemma [8] that an optimal classifier is the likelihood ratio $h_{\text{optimal}} = p_S(\boldsymbol{x})/p_B(\boldsymbol{x})$. Classification algorithms try to learn h_{optimal} or, in practice, its approximate from training data. The goal is thus to discriminate signal region events $(S + B)$ from sideband region events (B) and learn $p_{S+B}(\boldsymbol{x})/p_B(\boldsymbol{x})$ [1]. When the dimensionality of a problem is large, we use machine learning algorithms (e.g deep neural networks) that minimize a loss function using data and learn how to distinguish signal from background.

Consider a case of mixed samples and define mixtures $M_{1,2}$ of the original signal and background processes:

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

where $0 \leq f_2 \leq f_1 \leq 1$ are the signal fractions. In this case we are only given samples from distributions p_{M_1} and p_{M_2} with labels M_1 and M_2 and no longer have S or B labels. If we know fractions f_1 and f_2, a specific loss function can be constructed using these label proportions (we have some additional information about labels) and efficient machine learning can be done. An alternative strategy is classification without labels where f_1 and f_2 are unknown. In this approach a model is trained to discriminate two mixed samples M_1 and M_2 from one another (Figure 2). The authors in Ref. [11] show that an optimal classifier, trained to distinguish M_1 from M_2, is also optimal for distinguishing S from B.

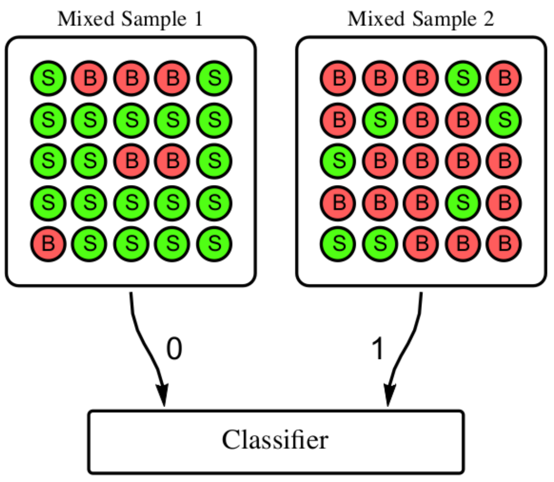


Figure 2: Mixed sample classification. The classifier is trained to distinguish data coming from the first (0) or the second (1) mixed sample. No information about S, B labels or their proportions is given to the classifier while training [11].

[1] This is the ratio of likelihoods for the two hypotheses of interest. The null hypothesis is the background hypothesis B (describes all known processes). The typical search is not free of background, so the alternative is the signal + background $(S + B)$ hypothesis [9, 10].

4 A brief introduction to neural networks

This section will quickly recap basic ideas of deep learning, see Ref. [12] for theory and Ref. [2] for a software implementation. Deep learning is part of a broader family of machine learning methods based on artificial neural networks. When we talk about neural networks we usually think of deep feedforward neural networks, meaning there are no feedback connections that fed the output of a network back into itself. The goal of such an algorithm is to approximate some function f^*. An example would be a classifier $h = f^*(\boldsymbol{x})$ that maps an input vector $\boldsymbol{x}$ to a category y (signal or background in our case). Each training example $\boldsymbol{x}$ comes with a label $y \approx f^*(\boldsymbol{x})$ that tells the network what to do; it must produce a value close to y. A feedforward network defines a mapping $y = f(\boldsymbol{x};\boldsymbol{\theta})$ and tries to learn the parameters $\boldsymbol{\theta}$ (weights and biases). The result is a function $f(\boldsymbol{x})$ that approximates $f^*(\boldsymbol{x})$ by trying to find an optimal set of values of $\boldsymbol{\theta}$, based on the available data in training. Neural networks are composed of many units (neurons) arranged in layers, with each layer being a function of the layer that preceded it (Figure 3).

Training a neural network requires: deciding on an architecture of a network and on activation functions, choosing the form of the output units and a corresponding cost function, specifying an optimizer and computing the gradients using the back propagation algorithm.

A layer can be described with a matrix of weights $\boldsymbol{W}$ connecting inputs and outputs of layers (individual neurons) and with a bias vector $\boldsymbol{b}$ as:

$$\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b} \,, \tag{2}$$

on which we apply an element-wise nonlinear function $g(\boldsymbol{z})$ called an activation function. Most commonly used activation function is a rectified linear unit (ReLU) [12] given by

$$g(z) = \max\{0, z\} \,. \tag{3}$$

The role of the output layer is to provide some additional transformation that characterizes the networks task. Choosing the loss function depends on the choice of this output unit. The simplest output unit is the linear output, where we do not apply any other function to the output of the network. Linear output layers are often used to produce the mean of a Gaussian distribution. If we want to predict the value of a binary variable y, we use a sigmoid output layer given by a transformation:

$$\sigma = \frac{e^x}{1 + e^x} \in [0, 1] \,, \tag{4}$$

which gives us a Bernoulli output distribution. Typical loss functions are mean squared error

$$L_{\mathrm{MSE}} = \frac{1}{N} \sum_{i=1}^{m} \left(h(\boldsymbol{x}_i) - y_i\right)^2 \,, \tag{5}$$

used for regression and binary cross entropy

$$L_{\mathrm{BCE}} = -\frac{1}{N} \sum_{i=1}^{m} \left[y_i \log h(\boldsymbol{x}_i) + (1 - y_i) \log(1 - h(\boldsymbol{x}_i))\right] \,, \tag{6}$$

used for binary classification tasks where m is the size of the subset of the available training data with values $\{x_i, y_i\}$.

We want to minimize these loss functions. We do this by moving in the direction of the negative gradient. The algorithm used is known as stochastic gradient descent (SGD). The basic idea is that the gradient can be approximately estimated using a small set of samples: $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ called mini batches. We may fit a training set with billions of examples using updates computed on only a handful of examples. Generally written the parameter update step is then

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{x}_i, y_i, \boldsymbol{\theta}) \,, \tag{7}$$

where η is the learning rate.

The input $\boldsymbol{x}$ provides the initial information that then propagates through the network to produce an output, this is called forward propagation. During training a scalar loss L is calculated at the end of forward propagation. The back propagation allows the information from the loss to then flow backward through the network in order to compute the gradients of parameters $\boldsymbol{\theta}$. Back propagation is an algorithm that computes the chain rule, that is highly efficient. The topic that is concerned with how to compute derivatives algorithmically is called automatic differentiation (see Ref. [13] for example).

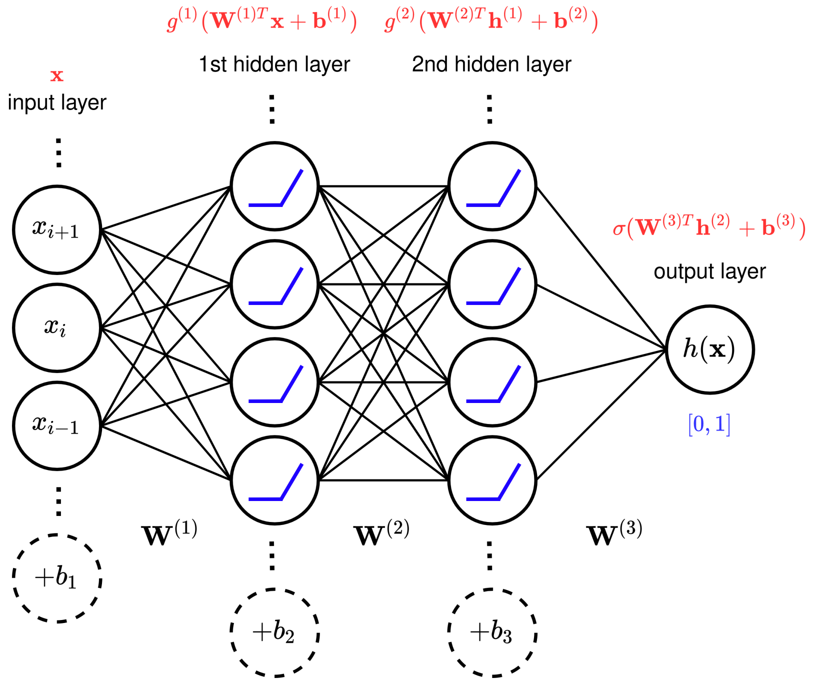


Figure 3: A neural network with two hidden layers that can be used for classification.

5 Extending the bump hunt

A resonance search looks for products of a heavy particle (resonance) decay. Using four-momentum conservation, events containing a resonance decay have to have at least two objects whose properties can be used to reconstruct an invariant mass spectrum. The structure of these objects can be used to distinguish signal from background. Let m_{res} be an observable in which a signal is resonant. The distribution of m_{res} is smooth given a background distribution and peaks around m_0 given a signal distribution. The new algorithm trains a classifier to distinguish a signal region M_1 from a sideband region M_2, using observables $\boldsymbol{x}$ (auxiliary features) that are not correlated to the resonance variable (target feature). In other words, machine learning is used to identify regions of phase space that are signal-like. Bump hunt is then performed on the identified regions of interest with respect to m_{res}, using a background model $f(m_{\text{res}})$. The procedure is then repeated for all mass hypotheses m_0. This is classification without labels, as described in Section 3, where the two mixed samples are the signal region and the sideband region.

5.1 An illustrative example

This section will present a simple example on how to use a neural network for classification without labels as described in Ref. [7]. Let auxiliary features be a two dimensional vector $\boldsymbol{x} = (x_1, x_2)$ where the observables x_1 and x_2 are uniformly distributed on a square of side length 1 centered at the origin. Let us define a signal region and a background region as:

$$\begin{aligned} SR &= (-w/2 < x < w/2, -w/2 < y < w/2)\,, \\ BR &= (-1/2 < x < 1/2, -1/2 < y < 1/2)\,. \end{aligned} \tag{8}$$

Let there be three mass bins that corespond to a signal region and two sidebands. The signal is present only in the middle bin. A neural network is then trained on (x_1, x_2) values do distinguish events in signal region from events belonging to both sidebands. In this two dimensional case it is possible to construct an accurate approximation of an optimal classifier without neural networks. The problem is that this approximation does not scale well to higher dimensions.

Figure 4 shows this procedure. Left plot shows a resonant mass distribution m_{res}. The middle two plots show both mixed samples and the right plot shows full training data. A neural network should reject all events outside of the black square centered at 0 of side length w.

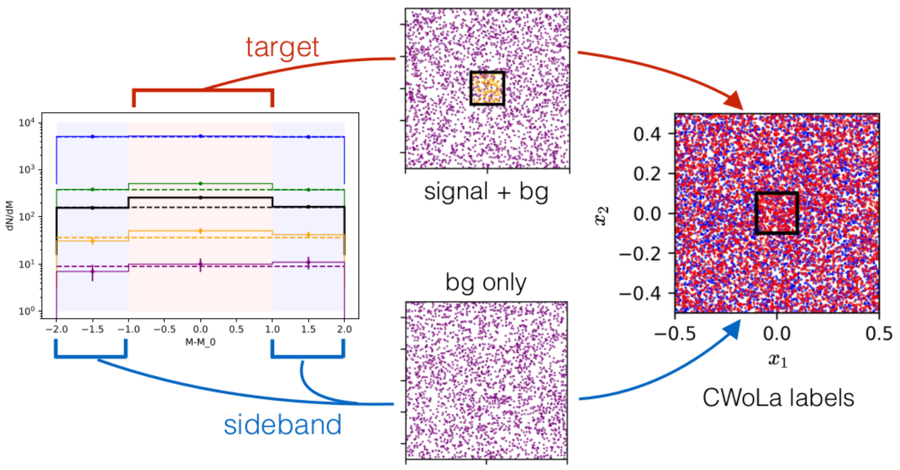


Figure 4: Illustration of CWoLa data preparation. Left plot shows mass distribution for three bins (signal region and two sidebands). Middle two plots show mixed samples (labels 0 and 1). The true signal is present in the black square with side lengths of $w < l$. The right plot shows combined distribution in the (x_1, x_2) plane that is used to train a classifier [7].

A deep feedforward neural network with three hidden layers was trained on data as presented in Figure 4. Construction and training of this neural network was done as described in Section 4. The data was split into three equal sets: training set, validation set and test set. The validation set was used for loss monitoring and for early stopping (stop training if loss does not improve after a given number of iterations). The test set was used for actual signal region searching. A more sophisticated scheme could be used that would not waste 2/3 of all data. An illustration of this procedure is showcased in Figure 5.

When a neural network has been trained it can be used to find signal regions in test data. Different colored histograms in Figure 4 indicate different thresholds on the neural network output (blue histogram is before any thresholding). This threshold is defined as a fraction of events with a given neural network value or higher, also called efficiency and denoted ϵ. Low ϵ value indicates high signal content. For each threshold the background expectation $\hat{n}_b$ is estimated by fitting a straight line to the mass sidebands (as shown in left plot in Figure 4). The significance is then estimated as $S \approx (n_0 - \hat{n}_b)/\sqrt{\hat{n}_b}$ for each threshold, where n_0 is the number of events in the signal region.

The numerical example presented in Figures 4 and 5 uses $N_b = 10000$ background events in each bin, $N_s = 300$ signal events in signal region and $w = 0.2$. Without any auxiliary variables $\boldsymbol{x}$ the significance corresponds to $N_s/\sqrt{N_b} = 3\sigma$. Using neural network classifier with efficiency $\epsilon = 0.01$ this increases to above 10σ.

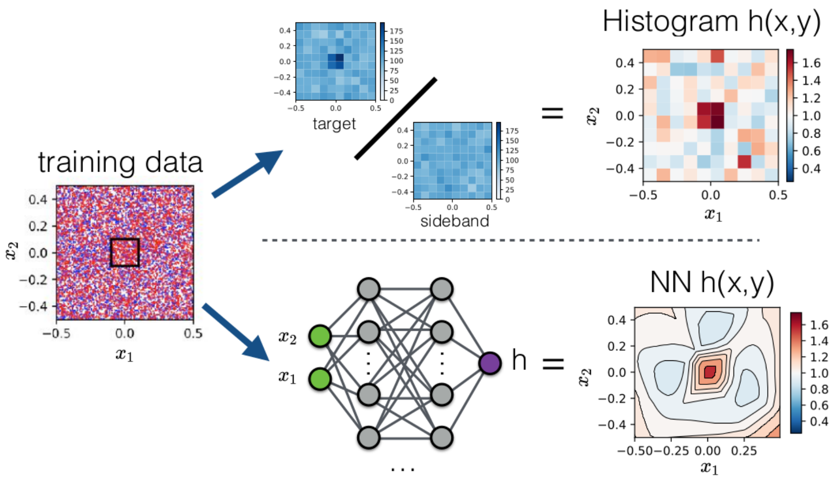


Figure 5: Estimation of an optimal classifier can be constructed by histogramming training events (upper path). The lower path shows a neural network classifier and its results in the (x_1, x_2) plane. Both methods give similar results but a neural network classifier can be generalized to higher dimensional data [7].

6 Dijet resonance search

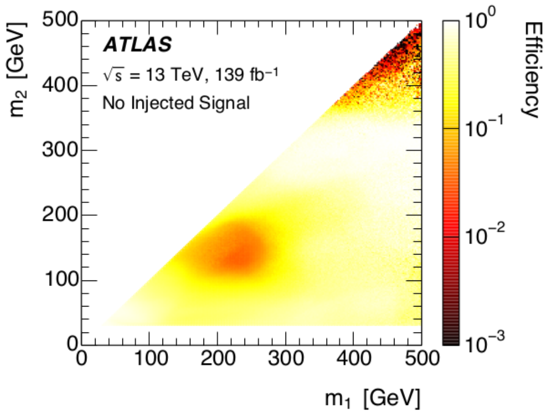
A dijet resonance decay can be described as

$$pp \to A \to BC \to JJ \tag{9}$$

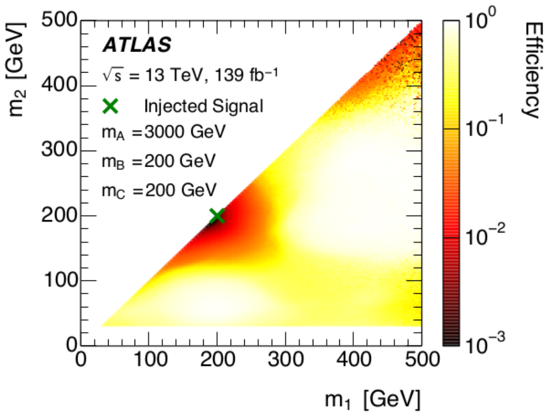
where pp are colliding protons, A is a resonance and B, C can be τ-leptons, b-quarks, top quarks, vector bosons, Higgs bosons and more, or even BSM particles. Decay products of B and C are measured as hadronic jets in the detector. The search described in Ref. [14] is a three-dimensional search for $m_A \sim \mathcal{O}(1\text{ TeV})$ and $m_B, m_C \sim \mathcal{O}(100\text{ GeV})$. It used the $\sqrt{s} = 13$ TeV collision data set of 139 fb^{-1} recorded by the ATLAS detector at LHC in Run 2 (2014 - 2018). The search is fully data driven and uses the machine learning anomaly detection technique described in Section 5.

Events with at least two jets were considered and the bump hunt was performed for dijet invariant mass distribution of the two leading jets (largest transverse momentum) in the range of 2.28 TeV $< m_{JJ} <$ 6.81 TeV. Some further jet preprocessing was also done. The masses of the two jets $\boldsymbol{x} = (m_1, m_2)$ were used as features for classification, where $m_1 \geq m_2$. The two mixed samples were constructed using eight regions with boundaries of m_{JJ}: [1.90, 2.28, 2.74, 3.28, 3.94, 4.73, 5.68, 6.81, 8.17] TeV. The signal region is defined as one of the six central regions and the sidebands are defined as the two adjacent regions. The entire search procedure was then repeated for every signal region. The neural network used had three hidden layers with 64, 32 and 8 neurons in each layer. The network learns to tag the signal and enhances a bump in the invariant mass spectrum, or, if there is no signal, the tagging will be random.

The method was validated using injected simulated events. With no signal injected, no evidence for excess was found in data. Figure 6 shows the network output in the absence of any signal (left) and with the injected simulated signal (right). The simulated signal was $W' \to WZ$, where W' is a new vector boson. We can see that the low-efficiency (signal-like) region of the neural network output is localized near the injected signal.



(a) No signal was injected. The signal region neural network tagging is random.



(b) Injected signal (marked with green cross) of $m_A = 3$ TeV and $m_B = m_C = 400$ GeV.

Figure 6: The efficiency of the neural network classifier in the (m_1, m_2) feature space [14].

After applying an event selection based on neural network efficiency in the given signal region, fitting of a parametric function was performed on the full m_{JJ} spectrum between 1.8 and 8.2 TeV with a binning of 100 GeV. A fit signal region and a fit sideband region were defined for evaluating the quality of the fit. The fit signal region was the same as the one used for neural network training. Each sideband was then split in half and the part closest to the signal region was used for the fit sideband region. Fitting was done using an iterative procedure using three fit functions:

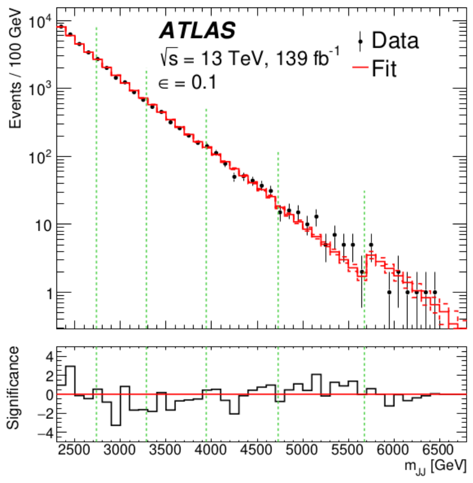
$$\frac{dn}{dx} = p_1(1-x)^{p_2-\xi_1 p_3} x^{-p_3} \,, \tag{10}$$

$$\frac{dn}{dx} = p_1(1-x)^{p_2-\xi_1 p_3} x^{-p_3+(p_4-\xi_2 p_3-\xi_3 p_2)\log(x)} \,, \tag{11}$$

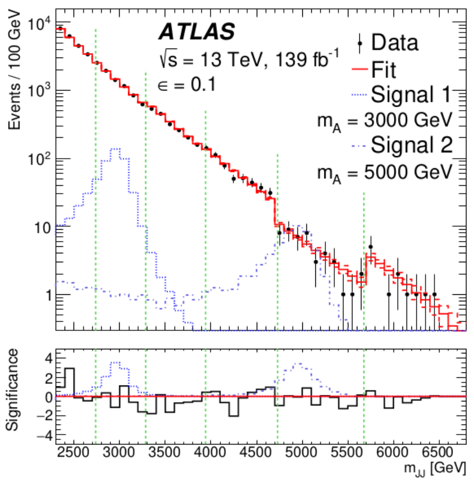
$$\frac{dn}{dx} = p_1 x^{p_2-\xi_3} e^{-p_3 x+(p_4-\xi_2 p_3-\xi_3 p_2)x^2} \,, \tag{12}$$

where $x = m_{JJ}/\sqrt{s}$, p_i are fit parameters and ξ_i are chosen such that p_i are not correlated. An iterative procedure of fitting is applied until the p-value from the sideband only χ^2 is greater then 0.05. First the data is used to fit the shape for Eq. 10, if the fit quality is insufficient Eq. 11 is used and then finally Eq. 12. If the fit quality is still insufficient the fit sidebands are reduced by 400 GeV and the three functions are tried again in the same order. This procedure is repeated until the fit is successful. The results are presented in Figure 7 for no injected signal (left) and for two different injected signals (right) at neural network efficiency of $\epsilon = 0.1$.

The injected signal model $W' \to WZ$ can be used to set limits on the production cross section of the two new particles. This is meant to illustrate the sensitivity of the search to the (m_A, m_B, m_C) parameter space. A 95% confidence level (upper limit) for the excluded signal cross section was calculated. The limits are presented in Figure 8 for different selected values of (m_B, m_C) and fixed m_A values of 3 TeV and 5 TeV. The limits on cross section vary with different masses and are poor when the neural network fails to find the signal. It is difficult for the neural network to determine the signal when m_B and m_C are low, because the Standard model background is larger in those regions. This results in weaker limits than those from other searches that did not use machine learning. For $m_B = m_C = 400$ GeV the upper limit cross section shows significant improvement over existing limits.

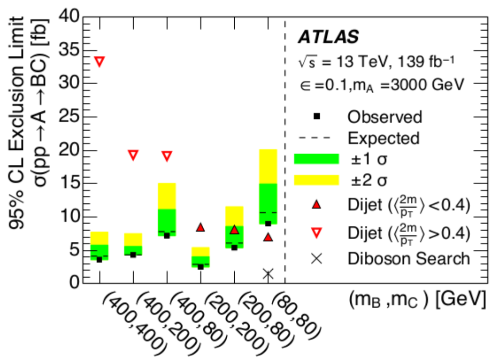


(a) No injected signal.

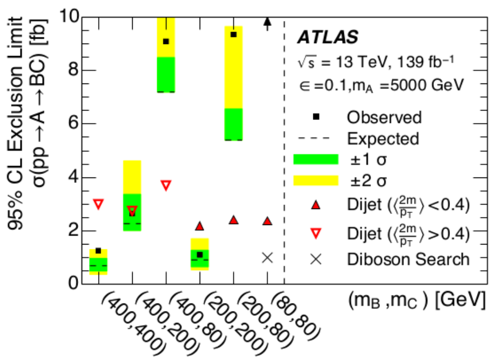


(b) Two injected signals.

Figure 7: Fitted background to data in all six signal regions (green lines). Signals are presented as blue histograms. The lower panel shows the significance, or the deviation between the fit and data. Signals injected were $m_A = 3$ TeV and $m_B = 5$ TeV with $m_B = m_C$=200 GeV [14].



(a) $m_A = 3$ TeV.



(b) $m_A = 5$ TeV.

Figure 8: 95% confidence level upper limits on the cross section for different $m_{B,C}$ signal models. The figures also shows results from two other searches for comparison [14].

7 Conclusion

Anomaly detection combined with unsupervised machine learning provides a way to search for new physics that is model-independent and data-driven. A common choice for a classifier is a neural network. The network in this approach aims to distinguish signal from background without having explicit signal and background labeled examples for training. This is achieved by the classification without labels method by defining two mixed samples, the first being a signal region and the second a sideband region. This procedure can then be used to enhances a bump in the invariant mass spectrum. For the procedure to work, a resonant variable m_{res} (e.g. invariant mass m_{JJ}) is needed in which background is smooth and signal is localized. Additional features $\boldsymbol{x}$ in the events that provide discriminating power (used for classifier training) are also needed (e.g. observables describing the jet substructure).

The method has been proven to work successfully in a dijet resonance search. The analysis considered a small neural network with only a two-dimensional feature space. Further studies are still needed for deeper architectures and larger feature spaces that could potentially give more striking results.

References

[1] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. URL: http://www.cs.cmu.edu/~tom/mlbook.html.

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

[3] *The Large Hadron Collider*. URL: https://home.cern/science/accelerators/large-hadron-collider.

[4] The ATLAS Collaboration. *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*. 2012. arXiv: 1207.7214 [hep-ex].

[5] The CMS Collaboration. *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*. 2012. arXiv: 1207.7235 [hep-ex].

[6] Wikipedia contributors. *Physics beyond the Standard Model — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Physics_beyond_the_Standard_Model&oldid=1004925780. [Online; accessed 18-February-2021].

[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] Jerzy Neyman and Egon Sharpe Pearson. "IX. On the problem of the most efficient tests of statistical hypotheses". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 231.694-706 (1933), pp. 289–337. URL: https://doi.org/10.1098/rsta.1933.0009.

[9] Alexander L Read. "Presentation of search results: the CLs technique". In: *Journal of Physics G: Nuclear and Particle Physics* 28.10 (2002), p. 2693.

[10] Glen Cowan. *Statistics for Searches at the LHC*. 2013. arXiv: 1307.2487 [hep-ex].

[11] 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].

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[13] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.2.9. 2018. URL: http://github.com/google/jax.

[14] 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].