Higgs Machine Learning Challenge with Neural Network Adversarial Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Higgs particle was theorized in 1964 and finally discovered in 2012 from the ATLAS and CMS experiments at CERN. This particle is the keystone of the Standard Model of particle physics, and provides the mechanism by which other particles acquire mass. Such discoveries are statistical in nature, and require vast amounts of information from the debris of particle collisions. Sifting and categorizing this data is a continual challenge, and in 2014 CERN and Kaggle invited the public to try their hand at such work, via the Higgs Machine Learning Challenge. A number of top entries, including the winner, used neural networks. An extended approach using the so-called ”adversarial training” technique is presented herein.

1 Introduction

Particle physics arguably began with the discovery of the electron by J.J. Thomson in 18?? [?]. This began a century of subsequent rapid discoveries of new particles, next being the identification of the proton in 19?? [?] and then the neutron in 19?? [?], both by Ernest Rutherford. These three particles - electron, proton and neutron - are all that’s needed to build the everyday chemical elements that comprise the apparent material universe. However, it was with early studies of cosmic rays that our understanding of fundamental particles began to change. One after the other more and more particles were discovered, with a broad spectrum of characteristics. Patterns slowly emerged, leading to the so-called Standard Model of particle physics, which encompasses not only particles of a material nature but also particles that transmit the very forces between them, thereby enabling their interactions.

As part of this framework, an altogether different particle was theorized to exist in order to explain why certain particles have mass. This came to be known as the Higgs particle, named after the author of the groundbreaking 1964 paper by one of the pioneers of this field [?]. So began a search that spanned nearly four decades, culminating with the discovery of the Higgs particle in 2012 from the Large Hadron Collider (LHC) at the CERN lab in Geneva, as measured by the ATLAS and CMS experiments [?]. The tell-tale signals were given by the physics properties of the Higgs particle’s decay products, as the particle itself cannot be directly observed due to decaying so rapidly that it cannot travel out of its creation region and into the detector.

There are two main abilities of the experimental apparatus that allowed for the Higgs particle to be discovered, that were not had with earlier experiments. First and foremost is the particle collision energy, and second is the ability of the detector equipment to record and analyze data. Every particle has a characteristic mass energy, and this energy must be met by a device such as the LHC in order for the particle to be created. Energy is the raw ingredient of all particles, in other words, and different particles require different amounts of this ingredient. The Higgs particle happens to be a relatively massive particle compared to a number of others, and therefore requires a relatively large amount of energy. A technological evolution over many years and many generations of device was necessary to develop the LHC, which is powerful enough to provide the Higgs creation energy.

Once created, a high mass particle such as the Higgs rapidly decays into other, lighter particles. It is these decay products that are detected by the machine surrounding the creation point, which are evaluated for such properties as momentum and electric charge, based on their trajectories through the machine, thereby leading to their classification. Originally, detectors were comprised of some continuous medium that would reveal a particle track that could be photographed and measured by hand. Things have since come a long way, with current detectors being formed of multiple layers of various electronics, each sampling the outgoing decay products in different ways. Despite the LHC providing an appropriate energy range for the creation of Higgs particles, their production is still exceedingly rare compared with the production of a number of other particles. An exceedingly large number of creation events is therefore required to compensate for the rarity, and the particle tracks simply cannot be assessed individually as used to be possible in photographic detectors from previous generations. The vast stream of electronic readout must instead be dealt with computationally and statistically, and a great deal of time and effort is spent by scientists determining how best to seek the signals of interest.

Machine learning has inevitably become an invaluable tool in this task, which, following the reconstruction of particle properties from the raw signal, is simply one of classification, namely it is a question of identifying signal versus background. In the case of the Higgs particle, its various decay channels are not unique, and are in fact shared by the decays of other heavy particles. Seeing a certain set of decay products does not then give indisputable proof of one particular parent particle or the other. But there are nonetheless subtle signs within the data when viewed *en masse*, which machine learning algorithms can be trained to recognize.

In 2014, CERN scientists collaborated with the Kaggle machine learning organization, developing an open competition for the public to try and develop a machine learning classifier to distinguish the Higgs particle from other heavy particles in the tau-tau decay channel (explained further in 2) [**?**]. This was the Higgs Machine Learning Challenge (HMLC), which at the time of its release was the most popular Kaggle challenge up to that point, with a total of 35,772 uploaded entries submitted by 1,785 teams (1,942 people) comprised of participants from around the world, some working within the physics community, others simply interested in computing [**?**]. The winning entry came from a sole programmer with no advanced physics training, who developed a neural network model that was clearly superior to anything else [**?**].

Many technical fields are developing incredibly fast at present, and machine learning is no exception. Even just more than one year on from the Higgs machine learning challenge, new ideas and techniques have become available. This project seeks to explore one such technique that has started to generate a lot of interest of late, namely that of adversarial training, and to apply it to the HMLC.

2 Higgs Machine Learning Challenge

Despite machine learning being a key component to many physics analyses at CERN, there is relatively little development of these techniques within the organisation itself [**?**]. When one technique is found to work and is well understood, and, crucially, it is well understood how to interpret the resulting physics results, then that technique gets entrenched and remains in play for a long time. This is not a bad thing in many ways, as by having staple workhorse codes it is possible for physicists to spend a greater fraction of their time analysing the data in search of physics.

However, machine learning is a field in its own right, and has its own experts and development trends quite separate from any of its applications, in science or otherwise. The last decade in particular has seen an enormous rise in both the ability of machine learning and the applications [**?**]. An ever increasing fraction of the developed world's activities has some computational aspect, and with that comes more data and more need to analyse that data for patterns. Coupled with this is the increase in the ability of hardware to run ever more complicated algorithms, making theoretical work from previous generations of computer science suddenly more relevant and useful.

The Higgs Machine Learning Challenge (HMLC) was develped in order to tap into the knowledge of this field straight from the enthusiasts, in order to port that knowledge into the particle physics domain. This is not only beneficial for CERN, but also the machine learning community at large, which demands as many real world problems as possible in order to develop the most relevant algorithms [?]. Sourcing good data is a job in itself, and if there's one thing that CERN is good at it's generating data. The HMLC was therefore a mutually beneficial arrangement for the data generation experts and the data processing experts.

The HMLC has two datasets - one fully labelled dataset for training, and another unlabelled dataset for testing. The training dataset has information about 250,000 labelled events, and the testing dataset corresponds to 550,000 unlabelled events. There are only two labels involved, namely ”signal” and ”background”. In a particle physics collider, two input particles (protons in the case of the LHC for ATLAS and CMS experiments) are accelerated and collided at high speed, resulting in a high energy state that can form any particle which ”costs” that amount of energy or less. Call this created particle the *initial state*. This initial state often has a number of possible decay channels to some set of *final state* particles, and some initial states share similar final states. In the case of the Higgs particle, one decay channel is that to a pair of lighter tau particles, with these then decaying to yet lighter particles still. These events are those labelled as ”signal”. Such final states are possible from the decay of other initial states such as W and Z particles (which are the carriers of the weak force) and top particles (which are the heaviest of the quark family) [?]. These events are those labelled as ”background”.

It is the subtle differences in the ensemble physics variables that we seek to distinguish through machine learning. Both training and testing datasets are defined by a feature set comprised of 30 different physics variables, representing quantities such as the mass, energy and momentum of decay products that entered the detector or were otherwise inferred [?, ?]. These entities are either (1) electrons, (2) heavier variants of electrons (the muon and tau particles), (3) jets of particles comprised of quarks (which are particles within the family of the constituents of protons and neutrons), and (4) so-called missing energy, representing very light and weakly interacting particles called neutrinos, which are not directly detected but inferred via a deficit in the energy accounting [?].

Physics details aside, this format of problem is ubiquitous and well known to machine learning, and as such requires no knowledge or understanding of what any of the features actually are. The HMLC dataset is actually a vastly simplified version of the data actually encountered by CERN physicists, who have to carefully reconstruct such high level output from a very low level input. The HMLC dataset was carefully tailored to be simply enough for public use, but still complicated and realistic enough to promote machine learning techniques that could be of actual use [?]. Furthermore, the dataset is in fact the product of a very fine Monte Carlo simulation, of the sort that is actually used for understanding the background in real data. The signal to background ratio in the HMLC dataset is much larger than that in real data, which requires many millions of events in order to provide statistically significant findings.

3 HMLC Winner and Initial Impulse

As mentioned in §1, there were many hundreds of entrants to the original Kaggle competition, with the overall winner being a single programmer with no professional physics training, specializing instead in computational development and consulting [?]. In fact, the physicist with the best score came in eighth place overall, highlighting the disparity in the computational skill sets between the professions, as should be somewhat expected. It was, after all, with such an outcome in mind that the challenge was run in the first place.

The winning entry was based on the neural network approach. The final iteration of this entry utilized 3 layers of 600 nodes each, finished off with 2 softmax output units (one for signal, one for background) [?]. Some feature preprocessing was performed on the input data, normalizing each feature to have zero mean and unit standard deviation, with some also being log-transformed. Further to the given features, four more were derived from some of the given angular features, and five more were derived from some of the given mass features, so giving nine extra hybrid quantities [?]. The algorithm was trained by minimising cross-entropy, with a correctness probability then assigned to each classification, and with only those above a certain threshold retaining their determined class labels at any one iteration [?]. Further to this, a range of model variants were employed simultaneously via *bagging*, with each variant given a different random subset of the training data and with their outputs then being averaged.

All these elements of the winning entry were included for good reasons. The effects of each were carefully examined over a four month development period, along with many others that were found to be of no benefit and discarded. The speed with which computer science is evolving means that even now, little more than a year after the close of the original Kaggle competition, there are new things to try that were not part of the winning formula. Soon after the start of this current project, the Kaggle winner was contacted for advice based on their past experience and their understanding of the machine learning field in general. They proposed that a so-called generative approach could be applicable to this task, and so initial research was then undertaken in this direction, and the seed of this project's central concept.

4 Adversarial Training

4.1 Style

Papers to be submitted to NIPS 2015 must be prepared according to the instructions presented here. Papers may be only up to eight pages long, including figures. Since 2009 an additional ninth page *containing only cited references* is allowed. Papers that exceed nine pages will not be reviewed, or in any other way considered for presentation at the conference.

Please note that this year we have introduced automatic line number generation into the style file (for $\LaTeX\ 2_\varepsilon$and Word versions). This is to help reviewers refer to specific lines of the paper when they make their comments. Please do NOT refer to these line numbers in your paper as they will be removed from the style file for the final version of accepted papers.

The margins in 2015 are the same as since 2007, which allow for $\approx 15\%$ more words in the paper compared to earlier years. We are also again using double-blind reviewing. Both of these require the use of new style files.

Authors are required to use the NIPS LaTeX style files obtainable at the NIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

4.2 Retrieval of style files

The style files for NIPS and other conference information are available on the World Wide Web at

`http://www.nips.cc/`

The file `nips2015.pdf` contains these instructions and illustrates the various formatting requirements your NIPS paper must satisfy. LaTeX users can choose between two style files: `nips15submit_09.sty` (to be used with LaTeX version 2.09) and `nips15submit_e.sty` (to be used with LaTeX2e). The file `nips2015.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own. The file `nips2015.rtf` is provided as a shell for MS Word users.

The formatting instructions contained in these style files are summarized in sections 5, 6, and 7 below.

5 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing of 11 points. Times New Roman is the preferred typeface throughout. Paragraphs are separated by 1/2 line space, with no indentation.

Paper title is 17 point, initial caps/lower case, bold, centered between 2 horizontal rules. Top rule is 4 points thick and bottom rule is 1 point thick. Allow 1/4 inch space above and below title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in section 7 regarding figures, tables, acknowledgments, and references.

6 Headings: first level

First level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 12. One line space before the first level heading and 1/2 line space after the first level heading.

6.1 Headings: second level

Second level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the second level heading and 1/2 line space after the second level heading.

6.1.1 Headings: third level

Third level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the third level heading and 1/2 line space after the third level heading.

7 Citations, figures, tables, references

These instructions apply to everyone, regardless of the formatter being used.

7.1 Citations within the text

Citations within the text should be numbered consecutively. The corresponding number is to appear enclosed in square brackets, such as [1] or [2]-[5]. The corresponding references are to be listed in the same order at the end of the paper, in the **References** section. (Note: the standard BIBTEX style `unsrt` produces this.) As to the format of the references themselves, any style is acceptable as long as it is used consistently.

As submission is double blind, refer to your own published work in the third person. That is, use “In the previous work of Jones et al. [4]”, not “In our previous work [4]”. If you cite your other papers that are not widely available (e.g. a journal paper under review), use anonymous author names in the citation, e.g. an author of the form “A. Anonymous”.

7.2 Footnotes

Indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).[2]

7.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction; art work should not be hand-drawn. The figure number and caption always appear after the figure. Place one line space before the figure caption, and one line space after the figure. The figure caption is lower case (except for first word and proper nouns); figures are numbered consecutively.

Make sure the figure caption does not get separated from the figure. Leave sufficient space to avoid splitting the figure and figure caption.

[1] Sample of the first footnote

[2] Sample of the second footnote

Table 1: Sample table title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

You may use color figures. However, it is best for the figure captions and the paper body to make sense if the paper is printed either in black/white or in color.

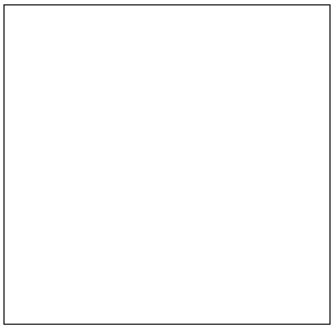

Figure 1: Sample figure caption.

7.4 Tables

All tables must be centered, neat, clean and legible. Do not use hand-drawn tables. The table number and title always appear before the table. See Table 1.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

8 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

9 Preparing PostScript or PDF files

Please prepare PostScript or PDF files with paper size “US Letter”, and not, for example, “A4”. The -t letter option on dvips will produce US Letter files.

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`

- LaTeX users:
  - Consider directly generating PDF files using `pdflatex` (especially if you are a MiKTeX user). PDF figures must be substituted for EPS figures, however.
  - Otherwise, please generate your PostScript and PDF files with the following commands:

    ```
    dvips mypaper.dvi -t letter -Ppdf -G0 -o mypaper.ps
    ps2pdf mypaper.ps mypaper.pdf
    ```

    Check that the PDF files only contains Type 1 fonts.
  - xfig ”patterned” shapes are implemented with bitmap fonts. Use ”solid” shapes instead.
  - The `\bbold` package almost always uses bitmap fonts. You can try the equivalent AMS Fonts with command

    ```
    \usepackage[psamsfonts]{amssymb}
    ```

    or use the following workaround for reals, natural and complex:

    ```
    \newcommand{\RR}{I\!\!R} %real numbers
    \newcommand{\Nat}{I\!\!N} %natural numbers
    \newcommand{\CC}{I\!\!\!\!C} %complex numbers
    ```
  - Sometimes the problematic fonts are used in figures included in LaTeX files. The ghostscript program `eps2eps` is the simplest way to clean such figures. For black and white figures, slightly better results can be achieved with program `potrace`.
- MSWord and Windows users (via PDF file):
  - Install the Microsoft Save as PDF Office 2007 Add-in from `http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=4d951911-3e7e-4ae6-b059-a2e79ed87041`
  - Select “Save or Publish to PDF” from the Office or File menu
- MSWord and Mac OS X users (via PDF file):
  - From the print menu, click the PDF drop-down box, and select “Save as PDF...”
- MSWord and Windows users (via PS file):
  - To create a new printer on your computer, install the AdobePS printer driver and the Adobe Distiller PPD file from `http://www.adobe.com/support/downloads/detail.jsp?ftpID=204` *Note:* You must reboot your PC after installing the AdobePS driver for it to take effect.
  - To produce the ps file, select “Print” from the MS app, choose the installed AdobePS printer, click on “Properties”, click on “Advanced.”
  - Set “TrueType Font” to be “Download as Softfont”
  - Open the “PostScript Options” folder
  - Select “PostScript Output Option” to be “Optimize for Portability”
  - Select “TrueType Font Download Option” to be “Outline”
  - Select “Send PostScript Error Handler” to be “No”
  - Click “OK” three times, print your file.
  - Now, use Adobe Acrobat Distiller or ps2pdf to create a PDF file from the PS file. In Acrobat, check the option “Embed all fonts” if applicable.

If your file contains Type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

9.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the graphicx package. Always specify the figure width as a multiple of the line width as in the example below using .eps graphics

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.eps}
```

or

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

for .pdf graphics. See section 4.4 in the graphics bundle documentation (`http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

References follow the acknowledgments. Use unnumbered third level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to 'small' (9-point) when listing the references. **Remember that this year you can use a ninth page as long as it contains *only* cited references.**

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D. S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609-616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer-Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.