X11th Quark Confinement and the Hadron Spectrum

Section on Statistics for the 21st Century

Thessalonika, Greece

Deep Learning and Bayesian Methods

Harrison B. Prosper

Department of Physics, Florida State University

September 1, 2016

Outline

Outline

Introduction

- In 2014, machine learning enthusiasts were challenged[1] to find the best classifier of ATLAS simulated signal and background events, where the signal was $H \to \tau^+\tau^-$ [2].

[1] HiggsML, `https://higgsml.lal.in2p3.fr/`

[2] Inspired by the 2013 ATLAS result, documented in ATLAS-CONF-2013-108, JHEP 04 (2015) 117.

Introduction

- In 2014, machine learning enthusiasts were challenged[1] to find the best classifier of ATLAS simulated signal and background events, where the signal was $H \to \tau^+\tau^-$ [2].
- The competition was won by Gábor Melis (*Franz Inc., Fixnum Services, Hungary*) who created a classifier using Deep Neural Networks (DNN). Deep in this context, as in Deep Learning, refers to the use of neural network models with multiple hidden layers.

[1] HiggsML, `https://higgsml.lal.in2p3.fr/`

[2] Inspired by the 2013 ATLAS result, documented in ATLAS-CONF-2013-108, JHEP 04 (2015) 117.

Introduction

- In 2014, machine learning enthusiasts were challenged[1] to find the best classifier of ATLAS simulated signal and background events, where the signal was $H \to \tau^+\tau^-$ [2].

- The competition was won by Gábor Melis (*Franz Inc., Fixnum Services, Hungary*) who created a classifier using Deep Neural Networks (DNN). Deep in this context, as in Deep Learning, refers to the use of neural network models with multiple hidden layers.

- The winning classifier was the average of 70 DNNs, each with architecture (35,600,600,600,2), *i.e.*, 35 inputs, 3 hidden layers of 600 nodes each, and 2 outputs. This implies a classifier with more than 70 million fitted parameters!

[1] HiggsML, `https://higgsml.lal.in2p3.fr/`

[2] Inspired by the 2013 ATLAS result, documented in ATLAS-CONF-2013-108, JHEP 04 (2015) 117.

Introduction

- In the early 2000s, attempts to train deep neural networks were frustrated by the apparent failure of the back-propagation algorithm (aka, stochastic gradient descent). Interest in neural networks waned and many researchers turned their attention to models such as boosted decision trees and support vector machines.

[3] Hinton, G. E., Osindero, S. and Teh, Y., *A fast learning algorithm for deep belief nets*, Neural Computation **18**, 1527-1554.

Introduction

- In the early 2000s, attempts to train deep neural networks were frustrated by the apparent failure of the back-propagation algorithm (aka, stochastic gradient descent). Interest in neural networks waned and many researchers turned their attention to models such as boosted decision trees and support vector machines.

- Then, in 2006, Hinton, Osindero and Teh[3] succeeded in training a deep neural network by first initializing its parameters sequentially, layer by layer. Each layer was trained to produce a representation of its inputs that served as the training data for the next layer. Then the network was tweaked using gradient descent.

[3] Hinton, G. E., Osindero, S. and Teh, Y., *A fast learning algorithm for deep belief nets*, Neural Computation **18**, 1527-1554.

Introduction

- In the early 2000s, attempts to train deep neural networks were frustrated by the apparent failure of the back-propagation algorithm (aka, stochastic gradient descent). Interest in neural networks waned and many researchers turned their attention to models such as boosted decision trees and support vector machines.

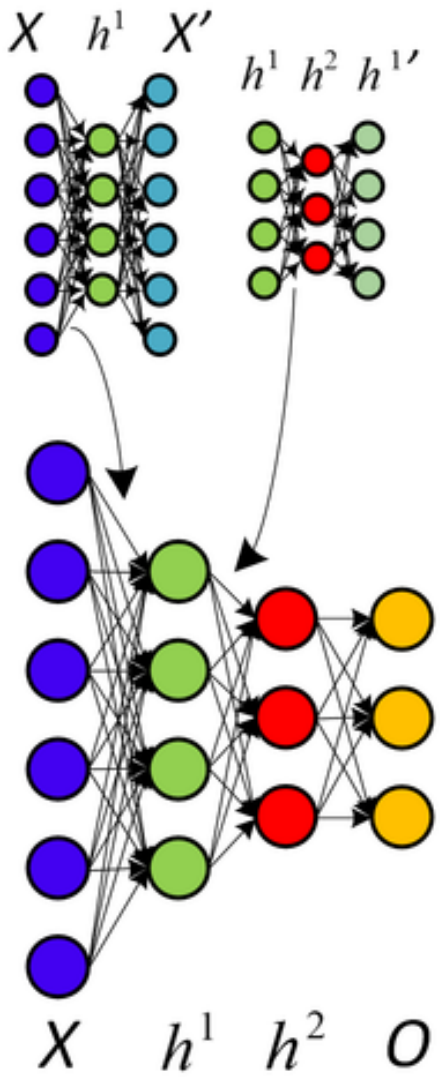
- Then, in 2006, Hinton, Osindero and Teh[3] succeeded in training a deep neural network by first initializing its parameters sequentially, layer by layer. Each layer was trained to produce a representation of its inputs that served as the training data for the next layer. Then the network was tweaked using gradient descent.

- This breakthrough seemed to provide compelling evidence that the training of deep neural networks requires careful initialization of parameters and sophisticated machine learning algorithms.

[3] Hinton, G. E., Osindero, S. and Teh, Y., *A fast learning algorithm for deep belief nets*, Neural Computation **18**, 1527-1554.

Introduction

One way to view the HOT model is as a stack of sequentially trained auto encoders. An auto encoder is a neural network that models the identity function $f : X \to X$ such that the hidden layer encodes a sparse representation of the inputs. The success of models of this kind rekindled interest in neural networks, in particular, ones with multiple hidden layers.



http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

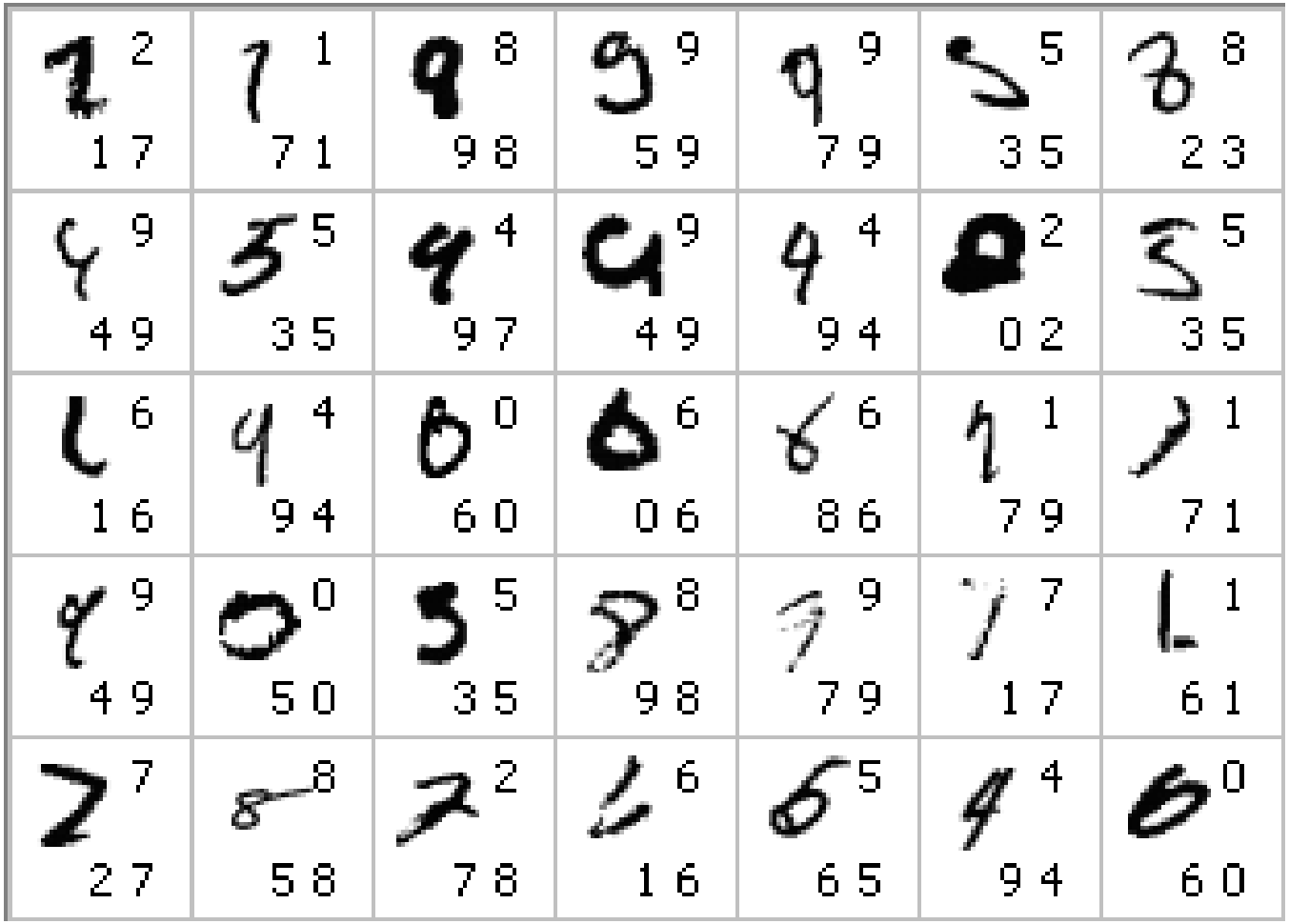
Introduction

- However, in 2010, a surprising counter example to the conventional wisdom was demonstrated by Cireşan *et al.*[4].
- The authors trained deep neural networks to classify the handwritten digits in the MNIST[5] data set, which comprises 60,000 $28 \times 28 = 784$ pixel images for training and 10,000 images for testing.
- They showed that a plain multi-layer perceptron (*i.e.*, an acyclic DNN) with architecture (784, 2500, 2000, 1500, 1000, 500, 10), trained using standard stochastic gradient descent (think `Minuit` on steroids!), outperformed all other methods that had been applied to the MNIST data set as of 2010. The error rate of this ∼12 million parameter DNN was 35 images out of 10,000. The misclassified images are shown on the next slide.

[4] Cireşan DC, Meier U, Gambardella LM, Schmidhuber J. ,*Deep, big, simple neural nets for handwritten digit recognition.* Neural Comput. 2010 Dec; 22 (12): 3207-20.

[5] `http://yann.lecun.com/exdb/mnist/`

Introduction



From Cireşan DC, Meier U, Gambardella LM, Schmidhuber J. Neural Comput. 2010 Dec; 22 (12): 3207-20.

Introduction

Question: how is it possible to fit a 12 million-parameter neural network, in a matter of hours, with a mere 60,000 images, avoid overfitting, and beat even the most sophisticated alternatives?

Introduction

Question: how is it possible to fit a 12 million-parameter neural network, in a matter of hours, with a mere 60,000 images, avoid overfitting, and beat even the most sophisticated alternatives?

Answer: brute force, which works spectacularly well. The key?

(i) Deploy lots of computing power, e.g., GPUs, and

(ii) generate a limitless sequence of training data, here images randomly, and slightly, *deformed* before every training epoch.

Introduction

Question: how is it possible to fit a 12 million-parameter neural network, in a matter of hours, with a mere 60,000 images, avoid overfitting, and beat even the most sophisticated alternatives?

Answer: brute force, which works spectacularly well. The key?

(i) Deploy lots of computing power, e.g., GPUs, and

(ii) generate a limitless sequence of training data, here images randomly, and slightly, *deformed* before every training epoch.

Item (ii) was a veritable stroke of genius because it meant that the entire set of 60,000 undeformed images could be used as the validation set during training, since none were used as training data, and overfitting was avoided because the training algorithm was essentially being fed an unlimited quantity of data.

Introduction

Before the Cireşan paper, conventional wisdom held that some combination of clever architecture and algorithms, clever feature engineering (machine learning-speak for finding good variables), clever initialization, and clever regularization, etc., was necessary to achieve state of the art results with DNNs.

[6] https://arxiv.org/abs/1202.2745v1

Introduction

Before the Cireşan paper, conventional wisdom held that some combination of clever architecture and algorithms, clever feature engineering (machine learning-speak for finding good variables), clever initialization, and clever regularization, etc., was necessary to achieve state of the art results with DNNs.

However, this paper and others, such as a 2012 paper on the recognition of German traffic signs[6] in which super-human pattern recognition was achieved, show that none of this is strictly necessary in order to achieve spectacular results on difficult problems.

[6] https://arxiv.org/abs/1202.2745v1

Introduction

Before the Cireşan paper, conventional wisdom held that some combination of clever architecture and algorithms, clever feature engineering (machine learning-speak for finding good variables), clever initialization, and clever regularization, etc., was necessary to achieve state of the art results with DNNs.

However, this paper and others, such as a 2012 paper on the recognition of German traffic signs[6] in which super-human pattern recognition was achieved, show that none of this is strictly necessary in order to achieve spectacular results on difficult problems.

You just need to embrace your inner brute!

[6] https://arxiv.org/abs/1202.2745v1

Outline

The Automated Physicist

The proximate goal of particle physics is to find evidence of new physics, that is, new patterns in data. The current approach entails deploying human intelligence to do the feature engineering, which has yielded variables such as M_{T2}, α_{T}, Razor variables, etc.

The Automated Physicist

The proximate goal of particle physics is to find evidence of new physics, that is, new patterns in data. The current approach entails deploying human intelligence to do the feature engineering, which has yielded variables such as M_{T2}, α_{T}, Razor variables, etc.

Surely, variables such as these will shed light on the nature of the new physics once found. And, until recently, practical considerations necessitated feature engineering by humans, inspired by the physics of the problem at hand.

The Automated Physicist

The proximate goal of particle physics is to find evidence of new physics, that is, new patterns in data. The current approach entails deploying human intelligence to do the feature engineering, which has yielded variables such as M_{T2}, α_{T}, Razor variables, etc.

Surely, variables such as these will shed light on the nature of the new physics once found. And, until recently, practical considerations necessitated feature engineering by humans, inspired by the physics of the problem at hand.

But is this really still necessary to *discover* new patterns data?

The Automated Physicist

The proximate goal of particle physics is to find evidence of new physics, that is, new patterns in data. The current approach entails deploying human intelligence to do the feature engineering, which has yielded variables such as M_{T2}, α_T, Razor variables, etc.

Surely, variables such as these will shed light on the nature of the new physics once found. And, until recently, practical considerations necessitated feature engineering by humans, inspired by the physics of the problem at hand.

But is this really still necessary to *discover* new patterns data?

A variable like M_{T2} cannot add information beyond that already present in the observables on which it is based. This truism underlies the so-called matrix element method, an *ab initio* approximation to the full probability density over the space of 4-vectors and particle identity.

The Automated Physicist

The recent successes in solving extremely difficult problems, such as

- face recognition in photographs (e.g., `Facebook`),
- natural language understanding (e.g., `Google`, `Microsoft`, `Apple`),
- self-learning of the game Go (`Google`), etc.,

offer potentially important lessons for the future of analysis in particle physics.

The Automated Physicist

The recent successes in solving extremely difficult problems, such as

- face recognition in photographs (e.g., `Facebook`),
- natural language understanding (e.g., `Google`, `Microsoft`, `Apple`),
- self-learning of the game Go (`Google`), etc.,

offer potentially important lessons for the future of analysis in particle physics.

One such lesson is this. In an era in which models with tens of millions of parameters can be trained in a matter of hours, and in which the goal is to discover scientifically significant discrepancies between observations and the predictions of the Standard Model, perhaps

> *there is less need to devote as much time as we do inventing clever variables.*

The Automated Physicist

Indeed, a few particle physicists, e.g., Daniel Whiteson (UC Irvine), have begun to explore the potential of deep learning in particle physics.

In a recent paper[7], deep neural networks, trained with low-level observables, were used to effect a 25% reduction in the amount of data needed to achieve the same result as an optimized shallow neural network (NN) trained with the same inputs and with the same number of free parameters ($\sim 56,000$).

While better results were obtained when high-level observables were added to the low-level ones, the authors noted that one expects better results would ensue by using deeper networks and more training data.

[7] P. Baldi, P. Sadowski, D. Whiteson, *Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning*, PRL **114**, 111801 (2015)

The Automated Physicist

Indeed, a few particle physicists, e.g., Daniel Whiteson (UC Irvine), have begun to explore the potential of deep learning in particle physics.

In a recent paper[7], deep neural networks, trained with low-level observables, were used to effect a 25% reduction in the amount of data needed to achieve the same result as an optimized shallow neural network (NN) trained with the same inputs and with the same number of free parameters ($\sim 56,000$).

While better results were obtained when high-level observables were added to the low-level ones, the authors noted that one expects better results would ensue by using deeper networks and more training data.

Again, you just need to embrace your inner brute!

[7] P. Baldi, P. Sadowski, D. Whiteson, *Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning*, PRL **114**, 111801 (2015)

The Automated Physicist

Could the era of the automated physicist be at hand? A few years from now, what might our tireless, friendly, automaton do on our behalf?

The Automated Physicist

Could the era of the automated physicist be at hand? A few years from now, what might our tireless, friendly, automaton do on our behalf?

- Automatically determine the set of characteristics that distinguish particles from the primary vertex from those from other vertices and automatically classify particles based on this information.

The Automated Physicist

Could the era of the automated physicist be at hand? A few years from now, what might our tireless, friendly, automaton do on our behalf?

- Automatically determine the set of characteristics that distinguish particles from the primary vertex from those from other vertices and automatically classify particles based on this information.
- Automatically reduce particle event data, e.g., $(p_{\mathrm{T}}, \eta, \phi)$ and particle identity, into a smaller fixed set of numbers, say $N \sim 500$ – which may be thought of as "pixelized images" – that can be the basis of further analysis.

The Automated Physicist

Could the era of the automated physicist be at hand? A few years from now, what might our tireless, friendly, automaton do on our behalf?

- Automatically determine the set of characteristics that distinguish particles from the primary vertex from those from other vertices and automatically classify particles based on this information.
- Automatically reduce particle event data, e.g., $(p_{\mathrm{T}}, \eta, \phi)$ and particle identity, into a smaller fixed set of numbers, say $N \sim 500$ – which may be thought of as "pixelized images" – that can be the basis of further analysis.
- Automatically classify these "images" into two sets: those that look like simulated events and those that don't.

The Automated Physicist

Could the era of the automated physicist be at hand? A few years from now, what might our tireless, friendly, automaton do on our behalf?

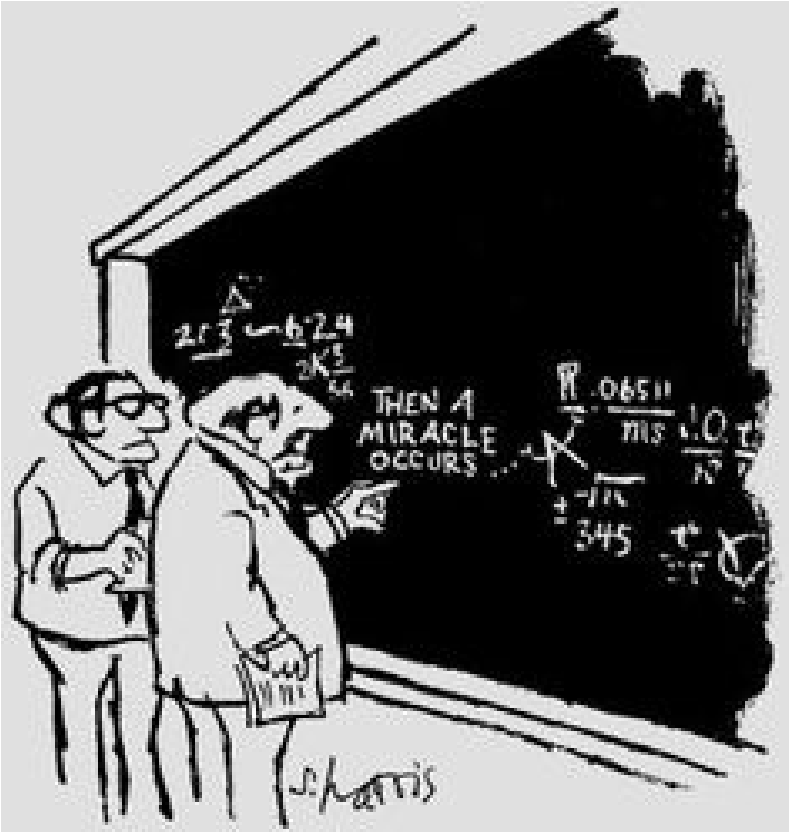
- Automatically determine the set of characteristics that distinguish particles from the primary vertex from those from other vertices and automatically classify particles based on this information.
- Automatically reduce particle event data, e.g., $(p_{\mathrm{T}}, \eta, \phi)$ and particle identity, into a smaller fixed set of numbers, say $N \sim 500$ – which may be thought of as "pixelized images" – that can be the basis of further analysis.
- Automatically classify these "images" into two sets: those that look like simulated events and those that don't.
- Automatically construct a hyper-fast simulator by auto encoding the mapping from parton level events to reconstruction level events using all available fully simulated events.

But, if all the fun stuff is automated, won't that precipitate existential angst in the clever variable industry?

Maybe!

But, after a spot of therapy, our colleagues would realize that there are still lots of hard problems to solve, such as understanding the nature of the physics that (we hope!) we shall have discovered.

"I think you should be more explicit here in step two."



"I think you should be more explicit here in step two."

Outline

The Bayesian Connection

Deep neural networks are now the method of choice to solve really, really, hard problems[8]. But, alas, they seem even more inscrutable than, for example, boosted decision trees, or even shallow neural networks.

Moreover, like most machine learning methods, a DNN provides, in effect, a point estimate with no uncertainty!

[8] D. Silver *et al.*, *Mastering the game of Go with deep neural networks and tree search*, Nature **529**, 484489 (28 January 2016)

[9] Y. Gal and Z. Ghahramani, *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, arXiv:1506.02142v5, 25 May 2016

The Bayesian Connection

Deep neural networks are now the method of choice to solve really, really, hard problems[8]. But, alas, they seem even more inscrutable than, for example, boosted decision trees, or even shallow neural networks.

Moreover, like most machine learning methods, a DNN provides, in effect, a point estimate with no uncertainty!

Recently, however, Gal and Ghahramani (U. of Cambridge)[9] demonstrated that a DNN, in which nodes are randomly dropped during training (a procedure referred to as dropout), approximates variational inference in Bayesian neural networks.

[8] D. Silver *et al.*, *Mastering the game of Go with deep neural networks and tree search*, Nature **529**, 484489 (28 January 2016)

[9] Y. Gal and Z. Ghahramani, *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, arXiv:1506.02142v5, 25 May 2016

Bayesian Neural Networks

Many machine learning methods, including DNN, amount to finding an optimal function $f(x, \omega^*)$ from a parameterized function class $\mathbb{F}_\omega$, by minimizing a cost function $C(T, \omega, \alpha)$ using some variation of stochastic gradient descent. Here, $T = (t, x)$ denotes the training data, ω the parameters to be found by the minimizer to yield a best-fit value ω^*, and α denotes the training parameters. The quantities t are the known targets for inputs x.

Bayesian Neural Networks

Many machine learning methods, including DNN, amount to finding an optimal function $f(x, \omega^*)$ from a parameterized function class $\mathbb{F}_\omega$, by minimizing a cost function $C(T, \omega, \alpha)$ using some variation of stochastic gradient descent. Here, $T = (t, x)$ denotes the training data, ω the parameters to be found by the minimizer to yield a best-fit value ω^*, and α denotes the training parameters. The quantities t are the known targets for inputs x.

In the Bayesian approach to neural networks, a posterior density

$$\begin{aligned} p(\omega|T) &= \frac{p(T|\omega)\,\pi(\omega)}{p(T)}, \\ &= \frac{p(t|x,\omega)\,\pi(\omega)}{p(t|x)}, \end{aligned}$$

is calculated for the network parameters ω, where $p(t|x, \omega)$ is the likelihood for the training data and $\pi(\omega)$ is the prior density that encodes what we know or wish to impose on the function class $\mathbb{F}_\omega$.

Bayesian Neural Networks

Given the posterior density, $p(\omega|T)$, we can compute, for example,

$$f(x) = \int f(x,\omega)\, p(\omega|T)\, d\omega,$$

$$\delta^2[f(x)] = \int [f(x,\omega) - f(x)]^2\, p(\omega|T)\, d\omega.$$

The first equation can be taken as an estimate of the desired function (under quadratic loss), while the second is a point-by-point estimate of the associated variance. In particle physics, we refer to the expression for $f(x)$ as a Bayesian neural network (BNN).

Bayesian Neural Networks

Given the posterior density, $p(\omega|T)$, we can compute, for example,

$$f(x) = \int f(x,\omega)\, p(\omega|T)\, d\omega,$$

$$\delta^2[f(x)] = \int [f(x,\omega) - f(x)]^2\, p(\omega|T)\, d\omega.$$

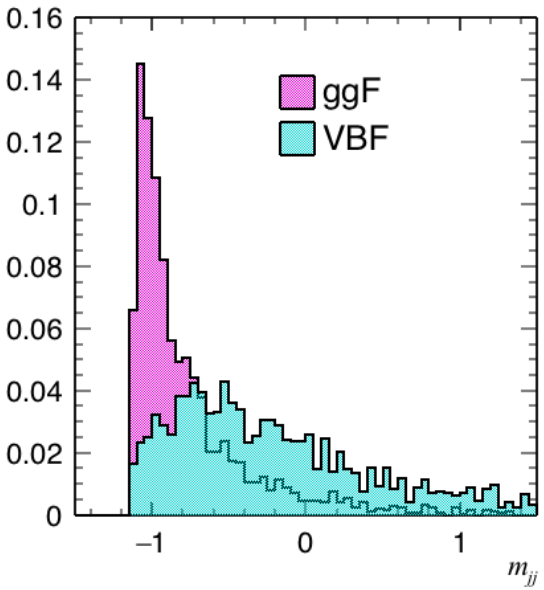
The first equation can be taken as an estimate of the desired function (under quadratic loss), while the second is a point-by-point estimate of the associated variance. In particle physics, we refer to the expression for $f(x)$ as a Bayesian neural network (BNN).

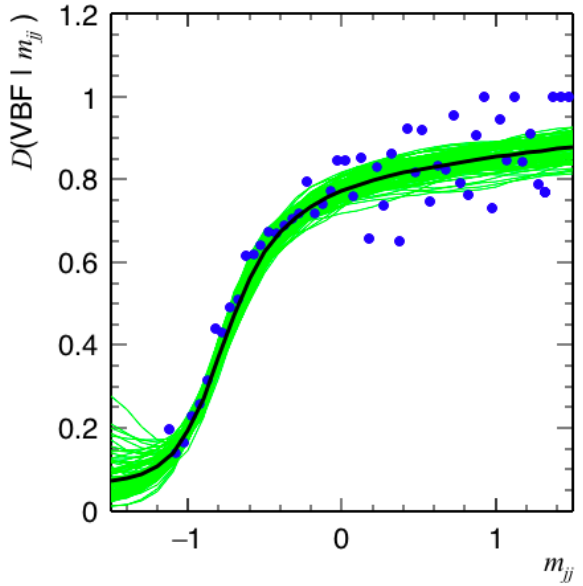
Unfortunately, computing $p(\omega|T)$ is intractable; so we must resort to approximations, for example, either

- sample $p(\omega|T)$ using Markov Chain Monte Carlo, or
- approximate $p(\omega|T)$ with the closest tractable approximation $q(\omega|\phi)$, where ϕ are variational parameters.

Bayesian Neural Networks

Example Create a BNN model of $D(x) = s(x)/[s(x) + b(x)]$, where s and b are the dijet mass ($x = m_{jj}$) densities for Higgs vector boson fusion (VBF) and gluon gluon fusion (ggF) events, respectively, where VBF is taken to be the signal. The dots in the right plot are computed using the histograms of s and b, while the green curves are the ensemble of (1, 10, 1) NNs whose average is the black curve. $p(\omega|T)$ is sampled using MCMC[10].





[10] http://www.cs.toronto.edu/~radford/fbm.software.html

Optimization

Pro: It furthers understanding of the training of DNN that it can be cast as a problem of Bayesian inference.

Con: There is the prior $\pi(\omega)$ to deal with. In practice, we do so using a prior $\pi(\omega, \alpha)$ parameterized with hyper-parameters α, which have to be chosen in some way of which there are at least three:

- Trial and error.
- Bayesian optimization[11]
- Empirical Bayes[12]

[11] See, for example, Gilles Loupe, `https://indico.cern.ch/event/516435`

[12] *An Introduction to Empirical Bayes Data Analysis*, George Casella, The American Statistician Vol. 39, No. 2 (May, 1985), pp. 83-87

Optimization

The basic idea of empirical Bayes is to replace the hyper-parameters with estimates thereof. Here is a proposed procedure:

1. Approximate using MCMC,

$$p(\omega|T,\alpha_0) \;=\; p(T|\omega)\,\pi(\omega,\alpha_0)/p(T,\alpha_0).$$

2. Then, given training data T', optimize the integral

$$\begin{aligned} p(T'|T,\alpha) &= \int p(T'|\omega)\,p(\omega|T,\alpha_0)w(\omega,\alpha,\alpha_0)\,d\omega, \\ &\approx \frac{1}{M}\sum_{i=1}^{M} p(T'|\omega_i)\,w(\omega_i,\alpha,\alpha_0), \end{aligned}$$

with respect to α. M is the number of points (NN functions) sampled and $w = \pi(\omega,\alpha)/\pi(\omega,\alpha_0)$ is a *known* weighting function.

Outline

Summary

- There were ample reasons to be skeptical of the claim by artificial intelligence researchers that machines would soon perform complex tasks at near-human or super-human levels, tasks such face recognition which most humans perform without difficulty.
- Today, however, skepticism is no longer warranted.
- Indeed, Whiteson and collaborators have demonstrated promising results in particle physics. I indulged in speculation about where DNN may be helpful and evoked the notion of an automated physicist.
- That the training of DNNs, using on-the-fly modifications of the network architecture, can be cast as an approximation to inference with Bayesian neural networks is, I think, a significant conceptual advance.
- This suggests that a breakthrough in approximating posterior densities over enormously large parameter spaces would constitute another watershed event in machine learning and, therefore, our own field.