# Deep Learning, Dark Knowledge, and Dark Matter

**Peter Sadowski** PETER.J.SADOWSKI@UCI.EDU *Dept. of Computer Science*
**Julian Collado** COLLADOU@UCI.EDU *Dept. of Computer Science*
**Daniel Whiteson** DANIEL@UCI.EDU *Dept. of Physics and Astronomy*
**Pierre Baldi** PFBALDI@ICS.UCI.EDU *Dept. of Computer Science*
*UC Irvine, Irvine, CA 92617*

**Editor:** Glen Cowan, Cécile Germain, Isabelle Guyon, Balàzs Kégl, David Rousseau

## Abstract

Particle colliders are the primary experimental instruments of high-energy physics. By creating conditions that have not occurred naturally since the Big Bang, collider experiments aim to probe the most fundamental properties of matter and the universe. These costly experiments generate very large amounts of noisy data, creating important challenges and opportunities for machine learning. In this work we use *deep learning* to greatly improve the statistical power on three benchmark problems involving: (1) Higgs bosons; (2) supersymmetric particles; and (3) Higgs boson decay modes. This approach increases the expected discovery significance over traditional shallow methods, by 50%, 2%, and 11% respectively. In addition, we explore the use of model compression to transfer information (*dark knowledge*) from deep networks to shallow networks.

**Keywords:** Deep learning, neural networks, high-energy physics

## 1. Introduction

Dark matter is believed to account for 85% of the matter in the universe (Planck Collaboration, 2013), but we have no information regarding the particle or particles which constitute it (Feng, 2010). To investigate the fundamental properties of matter and to develop increasingly unified theories of the physical universe, such as the *Standard Model* (Weinberg, 1967; Beringer et al., 2012), high-energy physicists have designed the most advanced experimental apparatuses ever constructed — particle accelerators — that collide protons or anti-protons at very high energies. These collisions produce exotic particles that are not otherwise observable in nature.

Producing and detecting these particles requires extraordinary resources. The largest particle collider in the world, the Large Hadron Collider (LHC), and its associated detectors cost billions of dollars to construct and approximately a million dollars per day to operate (Lefevre, 2009). Furthermore, the vast majority of collision events do not produce particles of interest. The LHC produces approximately $10^{11}$ collisions per hour (ATLAS Collaboration, 2013a), but only 300 of these collisions will result in a Higgs boson. It is therefore essential to process and analyze this experimental data efficiently. Fast trigger mechanisms are needed to curate events of interest *online* and sensitive statistical tools are needed to extract as much relevant information from the data as possible.

Identifying the collisions that contain particles of interest is a formidable challenge. These particles cannot be directly observed since they quickly decay into lighter, more stable particles. Therefore, the point of collision is surrounded by multiple layers of detectors (ATLAS Collaboration, 2008) that measure the direction and momentum of these decay products (Beringer et al., 2012). Collisions occur every 25 ns, and for each collision measurements are taken at more than 100M detector elements. The volume of raw data is therefore enormous — approximately 1 petabyte per second.

Background processes produce the same decay products, mimicking the events of interest. To distinguish between background processes and signal processes, physicists perform a statistical analysis of the measurements taken, namely the 3-dimensional momenta of the observed particles. This analysis is performed by computing the relative likelihood (Cowan et al., 2011) of the data under the two models, in the space of the momenta of the decay products. While the true analytic form of the likelihood is not available, good approximations are available using Monte Carlo methods (Sjostrand et al., 2006; Alwall et al., 2011; ATLAS Collaboration, 2010) which generate simulated collisions to populate the space of decay product momenta. However, the dimensionality of the space for measured quantities about each collision is large, approximately $3^N$, where $N$ is the number of observed decay products, so machine learning techniques are used to reduce the dimensionality of this space. Hence, the problem becomes a machine learning challenge: to efficiently use all of the information contained in the high-dimensional simulated data to discriminate between signal and background collisions.

The function to be learned is a non-linear function in a high-dimensional space. While in theory any reasonable function can be approximated to arbitrary precision by a 'shallow' neural network with a single hidden layer (Hornik et al., 1989), an intractable number of hidden units may be required, and learning still presents a challenge. A common approach is to engineer high-level features that make classification easier. These are generally non-linear functions of the input features that capture physical insights about the data. While helpful, this approach is labor-intensive and incomplete; an ideal machine learning method would not require this step.

Deep (many-layered) neural networks have the potential to compute complex functions more efficiently, with fewer hidden units than shallow neural networks. Historically, these networks were considered difficult to train, in part due to the vanishing gradient problem (Bengio et al., 1994; Hochreiter, 1998; Larochelle et al., 2009), but advances primarily in computing power and available training data have made these methods significantly more practical. Deep neural networks are now the state-of-the-art on multiple benchmark machine learning tasks (Szegedy et al., 2014; Cirean et al., 2012). Perhaps surprisingly, many of these state-of-the-art methods do not use human-engineered features (e.g. Gabor filters for computer vision) and instead *learn* multiple layers of features from the data. While this tends to require more data and computational resources, it can yield better performance with less feature engineering.

In this work, we use three benchmark datasets to demonstrate the performance gains achievable with deep learning in high-energy physics, building upon previous work by Baldi et al. (2014, 2015). Section 2 describes the datasets, including the distinction between low- and high-level features. Section 3 presents experimental results in which deep neural network architectures are optimized for each dataset using a Bayesian optimization algorithm. For each dataset, we present state-of-the-art results using deep learning. Section 4 describes model compression experiments that we use to explore why deep neural network architectures learn better than shallow ones.

## 2. Datasets

In this section we describe three datasets that we use to benchmark classification performance. Each dataset includes millions of simulated collision events, produced using software packages that are standard in the high-energy physics community: collision events are generated with the MADGRAPH5 (Alwall et al., 2011) event generator, showering and hadronization is performed by PYTHIA (Sjostrand et al., 2006), and the detector response is simulated by DELPHES (Ovyn et al., 2009). Additional details regarding the generation of the datasets can be found in Baldi et al. (2014) and Baldi et al. (2015).

Each collision event is characterized by a small number of essential measurements, mainly the 3D momenta of the observed particles, along with a number of additional, *high-level* features that have been engineered by physicists to aid in each classification task. These high-level features are derived from the low-level features — they are simple functions of the essential measurements, with no additional information about the collision. For each dataset, we describe the problem and the high-level features that have been proposed.

### 2.1. Benchmark 1: Higgs Boson

The first benchmark dataset concerns a theoretical process investigated by experiments at the LHC and the Tevatron colliders (ATLAS Collaboration, 2013b; CDF Collaboration, 2013). It consists of a signal process in which a Higgs boson is produced, and a background process with identical decay products but distinct kinematic features.

In the signal process, two gluons fuse into a heavy electrically-neutral Higgs boson ($gg \to H^0$), which subsequently decays to a $W$ boson and a heavy electrically-charged Higgs boson ($H^{\pm}$). The electrically-charged Higgs then decays to a second $W$ boson and the light Higgs boson, $h^0$, which predominantly decays to a pair of bottom quarks ($b\bar{b}$).

$$gg \to H^0 \to W^{\mp}H^{\pm} \to W^{\mp}W^{\pm}h^0 \to W^{\mp}W^{\pm}b\bar{b} \tag{1}$$

The background process is similar but does not produce the intermediate Higgs boson state, producing instead a pair of top quarks that each decay to $Wb$, also giving $W^{\mp}W^{\pm}b\bar{b}$ (Figure 1).

We focus on the semi-leptonic decay mode, in which one $W$ boson decays to a lepton and neutrino ($\ell\nu$) and the other decays to a pair of narrow cones of stable particles called jets ($jj$), giving decay products $\ell\nu b\ jjb$. We consider events which satisfy the requirements:

- Exactly one electron or muon, with $p_T > 20$ GeV and $|\eta| < 2.5$;
- at least four jets, each with $p_T > 20$ GeV and $|\eta| < 2.5$;
- $b$-tags on at least two of the jets, indicating that they are likely due to $b$-quarks rather than gluons or lighter quarks.

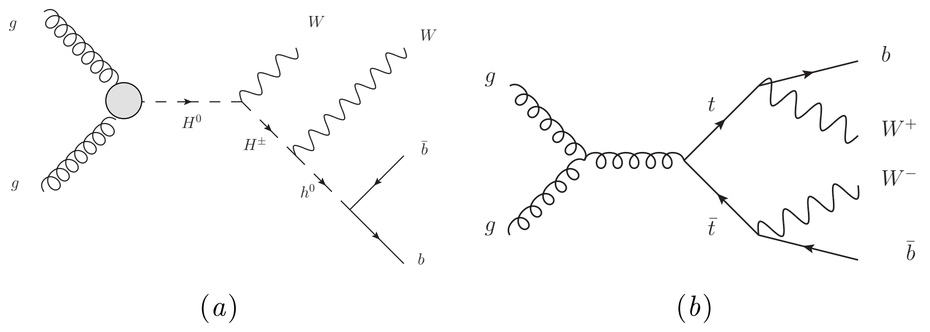

Figure 1: Feynman diagrams for Benchmark 1. (Left) Signal process involving new exotic Higgs bosons $H^0$ and $H^\pm$. (Right) Background process involving top-quarks ($t$).

The basic measurements made by the particle detector are the three-dimensional momenta of the five observable particles (the neutrino is not detected). In addition we reconstruct the missing transverse momentum in the event and have $b$-tagging information for each jet. Together, these twenty-one features comprise our *low-level feature set.*

To distinguish between the two processes, it is helpful to reconstruct the characteristic invariant mass of the intermediate particles. If a particle A decays into particles B and C, the invariant mass of particle A ($m_A$) can be reconstructed as:

$$m_A^2 = m_{B+C}^2 = (E_B + E_C)^2 - |(\mathbf{p}_B + \mathbf{p}_C)|^2 \tag{2}$$

where $E$ is the energy and $\mathbf{p}$ is the three-dimensional momentum of the particle. Similarly, the invariant mass of more than two particles can be defined as the modulus of the particles' Lorentz four-vector sum. In the signal hypothesis we expect that:

- $W \to \ell\nu$ gives a peak in the $m_{\ell\nu}$ distribution at the known $W$ boson mass, $m_W$,
- $W \to jj$ gives a peak in the $m_{jj}$ distribution at $m_W$,
- $h^0 \to b\bar{b}$ gives a peak in the $m_{b\bar{b}}$ distribution at the known Higgs boson mass, $m_{h^0}$,
- $H^\pm \to Wh^0$ gives a peak in the $m_{Wb\bar{b}}$ distribution at the assumed $H^\pm$ mass, $m_{H^\pm}$,
- $H^0 \to WH^\pm$ gives a peak in the $m_{WWb\bar{b}}$ distribution at the assumed $H^0$ mass, at $m_{H^0}$,

while in the background case we expect that:

- $W \to \ell\nu$ gives a peak in $m_{\ell\nu}$ at $m_W$,
- $W \to jj$ gives a peak in $m_{jj}$ at $m_W$,
- $t \to Wb$ gives a peak in $m_{jbb}$ and $m_{j\ell\nu}$ at $m_t$.

These quantities along with the invariant mass of the 4-vector sum of the jets, $m_{jjj}$, comprise a set of seven high-level features that incorporate domain knowledge about the classification task. We expect them to be useful for training a machine learning classifier. However, they are all functions of the low-level features, so a powerful classification model should not require them given enough training data.

The dataset generated for this benchmark consists of 11 million examples, with 53% signal and 47% background. Each input feature was standardized with mean zero and standard deviation one, except for those features with values strictly greater than zero – these were scaled so that the mean value was one. A random selection of 500,000 events was set aside for validation and hyperparameter selection, and another 500,000 for testing. The complete dataset can be found at the UCI Machine Learning Repository at archive.ics.uci.edu/ml/datasets/HIGGS.

### 2.2. Benchmark 2: Supersymmetry

The second benchmark task is to detect the production of supersymmetric particles. The signal process leads to a final state in which some particles are detectable and others are invisible to the experimental apparatus, while the primary background process produces the same detectable particles but fewer invisible particles and distinct kinematic features.

The signal process produces electrically-charged supersymmetric particles ($\chi^{\pm}$), which decay to $W$ bosons and an invisible, electrically-neutral supersymmetric particle $\chi^0$ (Figure 2). The final state in the detector is two charged leptons ($\ell\ell$) and missing momentum carried off by the invisible particles ($\chi^0\chi^0\nu\nu$). The background process is the production of pairs of $W$ bosons, which decay to charged leptons $\ell$ and invisible neutrinos $\nu$. The visible portion of the signal and background final states both contain two leptons ($\ell\ell$) and large amounts of missing momentum due to the invisible particles.

The eight low-level measurements relevant to this task are the 3D momenta of the two charged leptons, the particle jets induced by radiative processes, and the missing transverse momentum, $\not{E}_T$. High-level features are more difficult to construct for this dataset than for Benchmark 1; reconstructing the invariant mass of the intermediate state is infeasible as too much information is lost due to the two escaping neutrinos. There has been a vigorous effort in the literature to build high-level features which can aid in this classification task (Cheng and Han, 2008; Barr et al., 2003; Rogan, 2010; Buckley et al., 2013), including the ten used for this benchmark:

- Axial $\not{E}_T$: missing transverse energy along the vector defined by the charged leptons;
- Transverse mass $M_{T2}$: estimating the mass of particles produced in pairs and decaying semi-invisibly (Barr et al., 2003; Cheng and Han, 2008);
- ${\not{E}_T}^{Rel}$: $\not{E}_T$ if $\Delta\phi \geq \pi/2$, $\not{E}_T \sin(\Delta\phi)$ if $\Delta\phi < \pi/2$, where $\Delta\phi$ is the minimum angle between $\not{E}_T$ and a jet or lepton;
- Razor quantities $\beta$,$R$, and $M_R$ (Rogan, 2010);
- Super-razor quantities $\beta_{R+1}$, $\cos(\theta_{R+1})$, $\Delta\phi_R^{\beta}$, $M_{\Delta}^{R}$, $M_R^T$, and $\sqrt{\hat{s}_R}$ (Buckley et al., 2013).

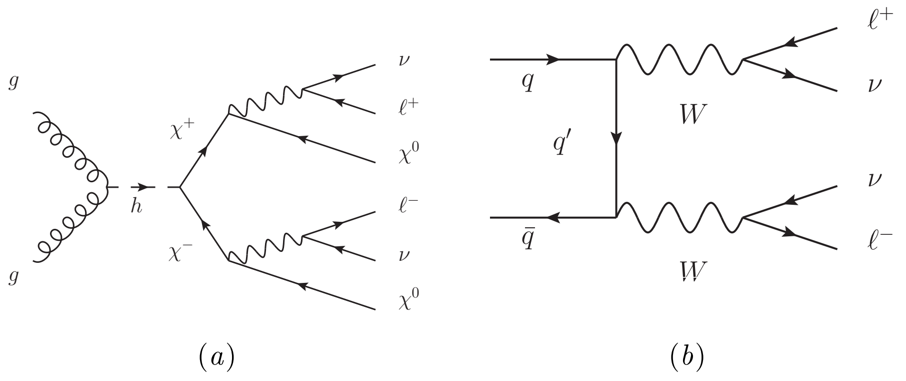

Figure 2: Feynman diagrams for Benchmark 2. (Left) Signal process involving hypothetical supersymmetric particles $\chi^{\pm}$ and $\chi^0$ along with charged leptons $\ell^{\pm}$ and neutrinos $\nu$. (Right) Background process involving $W$ bosons.

The supersymmetry dataset has 6 million examples, with 46% signal and 54% background. Each input feature was standardized with mean zero and standard deviation one, except for those features with values strictly greater than zero – these we scaled so that the mean value was one. A random selection of 500,000 events was set aside for validation and hyperparameter selection, and another 500,000 for testing. The complete dataset can be found at the UCI Machine Learning Repository at archive.ics.uci.edu/ml/datasets/SUSY.

### 2.3. Benchmark 3: Higgs to $\tau\tau$ decay

A major prediction of the Standard Model is that the Higgs boson couples to fermions through direct decay modes. In Benchmark 3, the task is to detect the decay of the Higgs to a pair of tau leptons ($\tau^{\pm}$), which subsequently decay to lighter leptons ($e$ and $\mu$) and two pairs of neutrinos ($\nu$). The complete process is given by:

$$gg \to H \to \tau^+\tau^- \to \ell^-\nu\nu\ell^+\nu\nu \tag{3}$$

The primary background process is $q\bar{q} \to Z \to \tau^+\tau^- \to \ell^-\nu\nu\ell^+\nu\nu$, which yields the identical set of particles. Given perfect measurement resolution and a complete description of all final particles, the invariant mass of the of the short-lived intermediate state could be computed using Equation 2, but finite measurement resolution and multiple escaping neutrinos make this impossible. Instead, we have the following 10 essential measurements to study:

- The three-dimensional momenta, $p$, of the charged leptons;
- The imbalance of momentum ($\not{p}_T$) in the final state transverse to the beam direction, due to unobserved or mismeasured particles;
- The number and momenta of particle 'jets' due to radiation of gluons or quarks.

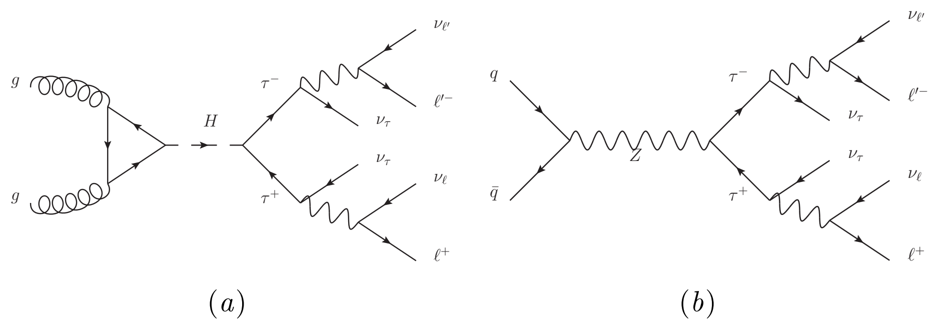

Figure 3: Feynman diagrams for Benchmark 3. (Left) Signal process in which the Higgs $H$ decays to leptons $\tau^+\tau^-$. (Right) The dominant background process.

The following 25 quantities can be derived from these essential measurements, and constitute the high-level features.

- Axial missing momentum, $\not{p}_T \cdot p_{\ell^+\ell^-}$;
- Scalar sum of the observed momenta, $|p_{\ell^+}| + |p_{\ell^-}| + |\not{p}_T| + \sum_i |p_{\text{jet}_i}|$;
- Relative missing momentum, $\not{p}_T$ if $\Delta\phi(p, \not{p}_T) \geq \pi/2$, and $\not{p}_T \times \sin(\Delta\phi(p, \not{p}_T)$ if $\Delta\phi(p, \not{p}_T) < \pi/2$, where $p$ is the momentum of any charged lepton or jet;
- Difference in lepton azimuthal angles, $\Delta\phi(\ell^+, \ell^-)$;
- Difference in lepton polar angles, $\Delta\eta(\ell^+, \ell^-)$;
- Angular distance between leptons, $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$;
- Invariant mass of the two leptons, $m_{\ell^+\ell^-}$;
- Missing mass, $m_{\text{MMC}}$ (Elagin et al., 2011);
- Sphericity and transverse sphericity;
- Invariant mass of all visible objects (leptons and jets).

This dataset contains 80 million examples, with 50% signal and 50% background. All 25 features were normalized with mean zero and standard deviation one. For those features with a skewness greater than 1.0, a small value of $10^{-8}$ was added, the logarithm was taken, then the values were normalized.

## 3. Results

In this section, we evaluate the classification performance of deep and shallow neural networks on the three benchmark datasets. For each benchmark, the neural network architecture and a number of other hyperparameters are optimized using the Spearmint Bayesian optimization algorithm (Snoek et al., 2012). Spearmint intelligently explores the high-dimensional hyperparameter space using a Gaussian process to model the validation error as a function of the hyperparameters.

The performance metrics used are the area under the signal efficiency vs. background rejection curve (AUC) and *expected discovery significance*, a null-hypothesis likelihood calculation measured in units of Gaussian $\sigma$ that is the standard metric in high-energy physics. Note that the discovery significance is calculated for a pre-specified signal/background ratio and is dataset-specific; the values reported here are solely for the purpose of comparing classifiers.

## 3.1. Benchmark 1

For the first benchmark, the Spearmint algorithm was used to optimize the neural network architecture and hyperparameters over the support in Table 1. The optimization included a total of 389 experiments, with 15 experiments running in parallel at any given time. In each experiment, a neural network was trained on 9 million examples and the complete set of features, using stochastic gradient descent with mini-batches of 100 examples, until the validation error did not decrease by more than 0.00001 for 10 epochs. These networks consisted of fully-connected layers of rectified linear (ReLU) units (Jarrett et al., 2009; Glorot et al., 2011). A momentum term increased linearly from an initial value to some final value over a specified number of epochs. Dropout with $p = 0.5$ was used in the top layers for regularization. The initial weights were drawn from a normal distribution with zero mean and a standard deviation of 0.1 for the first layer, 0.001 for the last layer, and 1.0 divided by the square root of the number of inputs for the other layers. The best network was selected based on the validation set error.

The best ReLU architecture had 3 hidden layers of 490 units each and dropout in the last layer. Table 2 compares the test set performance against the best of the shallow architectures from the Bayesian optimization, which contained the maximum of 500 hidden units. We also compare these results against the deep and shallow neural networks of *tanh* units from Baldi et al. (2014), which were optimized using a grid search. The best deep network from that work has 4 hidden layers, with 300 hidden units in each hidden layer, and the best shallow network has 1000 hidden units.

The best classifier overall is a deep *tanh* network with 8 hidden layers of 300 hidden units each. This network was trained for 600 epochs, with learning rate decreasing from 0.1 to 0.000001 by a factor of 1.0000002 after each minibatch update of 100 examples, momentum increasing from 0.9 to 0.99 linearly over the first 200 epochs, and a L2 weight-decay coefficient of 0.00001. This network achieves state-of-the-art performance on this dataset with an additional 1% AUC.

All classifiers were trained on the low- and high-level feature subsets in addition to the complete feature set, using the same hyperparameters optimized for the complete set of features. DNNs trained on the low-level features alone perform nearly as well as the DNNs trained on the full set, suggesting that the deep networks are able to learn the useful information contained in the reconstructed characteristic invariant mass of the intermediate particles.

Table 1: Hyperparameter support for Bayesian optimizations on benchmarks 1 and 2.

| Hyperparameter | Min | Max | Best benchmark 1 | Best benchmark 2 |
|---|---|---|---|---|
| Hidden layers | 1 | 9 | 3 | 9 |
| Hidden units/layer | 100 | 500 | 490 | 100 |
| Dropout layers | 0 | 3 | 1 | 0 |
| Initial learning rate | 0.1 | 0.1 | 0.1 | 0.1 |
| Learning rate decay $(1 + 10^x)$ | -6 | -7 | $-6.2$ | $-6.0$ |
| Initial momentum | 0.0 | 0.5 | 0.19 | 0.0 |
| Final momentum (log-scaled) | 0.500 | 0.999 | 0.998 | 0.500 |
| Epochs until mom. saturation | 10 | 50 | 50 | 10 |

Table 2: **Performance comparison on benchmark 1.** Performance of shallow neural networks (NN), deep neural networks (DNN) of tanh units tuned with a grid search, and DNNs of rectifier units tuned with Spearmint for three sets of input features: low-level features, high-level features and the complete set of features. Neural network was trained five times with different random initializations; the table displays the mean AUC, with standard deviations in parentheses, as well as the expected significance of a discovery for 100 signal events and 5000 background events.

| | | AUC | |
|---|---|---|---|
| Technique | Low-level | High-level | Complete |
| NN (ReLU) | 0.790 ($<$ 0.001) | 0.794 ($<$ 0.001) | 0.844 ($<$ 0.001) |
| DNN (ReLU) | 0.874 ($<$ 0.001) | 0.801 ($<$ 0.001) | 0.880 (0.001) |
| NN (tanh) | 0.785 (0.001) | 0.788 (0.001) | 0.841 ($<$ 0.001) |
| DNN (tanh) | 0.880 (0.001) | 0.800 ($<$ 0.001) | 0.885 (0.002) |
| DNN (best) | 0.891 | 0.801 | 0.896 |
| | | Discovery significance | |
| Technique | Low-level | High-level | Complete |
| NN (ReLU) | 2.16$\sigma$ (0.03) | 2.64$\sigma$ (0.01) | 3.28$\sigma$ (0.02) |
| DNN (ReLU) | 3.96$\sigma$ (0.02) | 2.89$\sigma$ (0.01) | 4.34$\sigma$ (0.07) |
| NN (tanh) | 2.08$\sigma$ (0.03) | 2.36$\sigma$ (0.02) | 3.23$\sigma$ (0.03) |
| DNN (tanh) | 4.29$\sigma$ (0.05) | 2.82$\sigma$ (0.01) | 4.66$\sigma$ (0.06) |
| DNN (best) | 4.57$\sigma$ | 2.87$\sigma$ | 4.82$\sigma$ |

Table 3: **Performance comparison on benchmark 2.** The best model from each method is compared in terms of AUC. The expected significance of discovery is shown for the neural network models, calculated for 100 signal events and 5000 background events.

| | AUC | | |
|---|---|---|---|
| Technique | Low-level | High-level | Complete |
| NN | 0.8745 ($< 0.0001$) | 0.8679 ($< 0.0001$) | 0.8788 (0.0001) |
| DNN | 0.8760 ($< 0.0001$) | 0.8695 ($< 0.0001$) | 0.8790 ($< 0.0001$) |
| | Discovery significance | | |
| Technique | Low-level | High-level | Complete |
| NN | $7.86\sigma$ (0.06) | $7.22\sigma$ (0.02) | $7.81\sigma$ (0.05) |
| DNN | $7.73\sigma$ (0.07) | $7.58\sigma$ (0.08) | $7.94\sigma$ (0.08) |

## 3.2. Benchmark 2

The same Bayesian optimization procedure from Benchmark 1 was used to optimize neural network architectures for the supersymmetry task. This optimization was run for 111 experiments. The best architecture had 9 hidden layers, and outperforms the results in Baldi et al. (2014). Table 3 compares the performance of this architecture against the single best shallow network out of the 32 networks with a single hidden layer that were trained by Spearmint. On this benchmark, we do not see a large performance boost with deep learning, but the human-engineered features do not help much either.

## 3.3. Benchmark 3

For the $H \rightarrow \tau^+\tau^-$ decay task, we perform two separate optimizations using Spearmint, one for deep networks and one for shallow networks, with 100 experiments each. In each experiment, a neural network was trained on a random set of 40 million examples and the complete set of features for 100 epochs. The neural network classifiers used a simple architecture, with fully-connected layers of rectifier units of the same size in each layer. Stochastic gradient descent was used with mini-batches of 100 examples. A momentum term increased linearly from an initial value to some final value over a specified number of epochs.

The hyperparameter supports for the deep and shallow network optimizations are shown in Table 4. Both optimizations are carried out in six dimensions — for the deep networks, the number of layers is allowed to vary but the initial learning rate is fixed. The initial weights of both the shallow and deep networks were drawn from a normal distribution with zero mean and a standard deviation of 0.1 for the first layer, 0.001 for the last layer, and 1.0 divided by the square root of the number of inputs for the other layers. The best shallow and deep neural network hyperparameters were then used to train classifiers on a larger set of 60 million training examples, with five different random weight initializations and train/test splits. The mean and standard deviation of the five networks is shown in Table 5. Figure 4 shows how deep networks trained on the low-level features perform almost as well as those trained on the complete set.

Table 4: Hyperparameter support and Spearmint selection for shallow and deep networks on benchmark 3.

| | Deep Networks | | | Shallow Networks | | |
|---|---|---|---|---|---|---|
| Hyperparameter | Min | Max | Best | Min | Max | Best |
| Hidden layers | 1 | 7 | 7 | 1 | 1 | 1 |
| Hidden units/layer | 100 | 500 | 274 | 100 | 56000 | 693 |
| Initial learning rate | 0.1 | 0.1 | 0.1 | $10^{-2}$ | $10^{-7}$ | 0.006 |
| Learning rate decay $(1 + 10^x)$ | -9 | -3 | $-6.29$ | $-9$ | $-3$ | $-8.05$ |
| Initial momentum | 0.0 | 0.5 | 0.0 | 0.0 | 0.5 | 0.5 |
| Final momentum | 0.500 | 0.999 | 0.996 | 0.500 | 0.999 | 0.500 |
| Epochs until mom. saturation | 20 | 100 | 20 | 20 | 100 | 99 |

Table 5: Comparison of the performance of shallow neural networks (NN), and deep neural networks (DNN) for three sets of input features: low-level variables, high-level variables, and the complete set of variables. Each neural network was trained five times with different random weight initializations and different train/test splits. Performance of an ensemble classifier using all five instances is also shown. The table displays the mean AUC of the signal-rejection curve calculated from 10 million test points; standard deviations are in parentheses. The mean expected significance of a discovery is given for 100 signal events and 5000 background events.

| | AUC | | |
|---|---|---|---|
| Technique | Low-level | High-level | Complete |
| NN | 0.789 (0.0010) | 0.792 (0.0002) | 0.797 (0.0004) |
| NN ensemble | 0.791 | 0.793 | 0.798 |
| DNN | 0.798 (0.0001) | 0.798 (0.0001) | 0.802 (0.0001) |
| DNN ensemble | 0.798 | 0.798 | 0.803 |
| DNN (best) | - | - | 0.803 |
| | Discovery significance | | |
| Technique | Low-level | High-level | Complete |
| NN | $2.57\sigma$ (0.006) | $2.92\sigma$ (0.006) | $3.02\sigma$ (0.008) |
| NN ensemble | $2.61\sigma$ | $2.96\sigma$ | $3.06\sigma$ |
| DNN | $3.16\sigma$ (0.003) | $3.24\sigma$ (0.003) | $3.37\sigma$ (0.003) |
| DNN ensemble | $3.18\sigma$ | $3.26\sigma$ | $3.39\sigma$ |
| DNN (best) | - | - | $3.39\sigma$ |

To analyze the effect of ensembles, we took the arithmetic mean of the predictions from the five classifiers on another 10 million examples of test data. This ensemble gives us an additional small boost in performance.

However, the same performance boost is achieved by adding layers to the deep network (with some additional tuning). The best single classifier we were able to train had 10 layers, 300 hidden ReLU units in each layer, a learning rate decay factor of 1.00000027, and momentum increasing from 0.0 to 0.997 over 50 epochs.

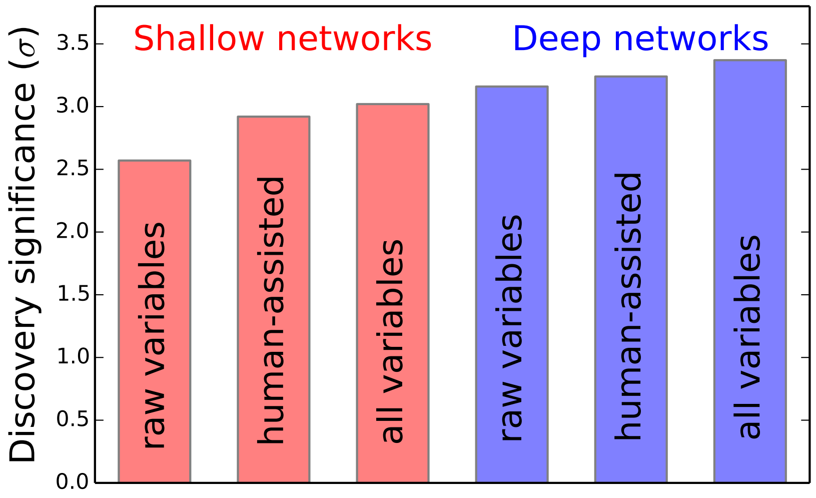

Figure 4: Relative performance of best shallow and deep networks trained on each feature set of benchmark 3. Hyperparameters were selected separately for shallow and deep networks on the complete feature set using the Spearmint algorithm.

Table 6: Regression models of different architectures were trained to predict the high-level features from the low-level features. The average mean squared error (MSE) is computed on the test sets.

| | MSE | | |
|---|---|---|---|
| Architecture | Benchmark 1 | Benchmark 2 | Benchmark 3 |
| Linear Regression | 0.1471 | 0.2790 | 0.7153 |
| NN | 0.0814 | 0.0100 | 0.0372 |
| DNN | 0.0785 | 0.0088 | 0.0269 |

## 4. Learning with Dark Knowledge

Deep neural networks appear to have an advantage over shallow networks in learning complex functions, but where this advantage comes from is not entirely understood. One possibility is that the extra representational power provided by the architecture is necessary to represent the complex features relevant to classification. Indeed, the high-level engineered features are clearly useful for classification, and our attempts to learn the high-level features from the low-level features suggest that it is easier for a deep neural network to learn these features than a shallow neural network (Table 6).

However, another possibility is that deep neural networks *learn* better than shallow methods due to architectural constraints. The architecture may encourage the lower layers to learn re-usable features that generalize well to lots of examples. This hypothesis is

Table 7: Performance of shallow networks trained with dark knowledge.

| | AUC | | |
|---|---|---|---|
| Architecture | Benchmark 1 | Benchmark 2 | Benchmark 3 |
| NN | 0.842 | 0.8786 | 0.797 |
| NN w/ dark knowledge | 0.850 | 0.8788 | 0.799 |
| DNN | 0.885 | 0.8790 | 0.802 |

supported by the fact that the best architectures for Benchmarks 2 and 3 are very deep architectures (the maximum allowed) with relatively few units in each hidden layer.

Recent work by Ba and Caruana (2014) and Hinton et al. (2014) suggests a way to further explore these hypotheses. They argue that shallow classifiers can often perform as well as deep neural networks or large ensembles through a model compression process. If this were true, then the performance advantages of deep models are not simply a consequence of their ability to represent certain functions.

In the simplest form of the model-compression algorithm, the output predictions of the complex model on the training data are used as labels for training the shallow model. Thus the shallow model is trained on the soft targets produced by the deep model, rather than the *true* binary labels. Thus, for instance, it gets to see which examples the large model is uncertain about. Likewise, in multiclass problems, the shallow model gets to see for each training example a distribution over the set of classes. The extra information contained in this modified training set is referred to as *dark knowledge* (Hinton et al., 2014).

We investigated this idea by training shallow neural network architectures to mimic the best deep networks on each benchmark, using the simple algorithm described above. For a single logistic output, this algorithm is equivalent to the algorithm described by Ba and Caruana (2014) and also the algorithm described by Hinton et al. (2014), adapted to a logistic output, with a temperature parameter of $T = 1$. Variations on these ideas include increasing the temperature of the normalized exponential output layer, or using the pre-output layer values (logits) as the targets. While increasing the temperature can further smooth the target probabilities, it seemed unnecessary here, given the substantial uncertainty in the class predictions in the three benchmark datasets.

Our results confirm the previous findings: the dark knowledge improves performance of the shallow classifier on each of the three benchmarks. This performance increase is only a fraction of the total increase one achieves from using a deep network, so this experiment does not completely rule out the representation hypothesis, but it is likely that the performance of the shallow networks can be further improved with more training and larger shallow networks. These questions are currently under investigation.

## 5. Discussion

The high-level variables derived by physicists clearly capture features of the data that are relevant to classification. But while the shallow neural networks perform poorly on the low-level variables alone, deep networks trained on the low-level variables tend to perform nearly as well as the networks trained on the complete feature set. This is *despite* the fact that all neural network hyperparameters were tuned using the full feature set.

This work strongly suggests that multiple layers of hidden units are useful in learning discriminative information of the type contained in the high-level features. This parallels the progress of deep learning in computer vision, where convolution neural networks learn filters that look like Gabor filters and other features that have been engineered for computer vision tasks (Krizhevsky et al., 2012).

In Benchmarks 1 and 3, the deep networks trained on the low-level features outperform the shallow networks that have been trained on the full set, indicating that they have learned patterns in the low-level features that are *more* useful for the classification task than the human-engineered features. This suggests that deep learning can be more effective than feature-engineering for these applications, the only cost being the additional computation needed to train and tune deep neural network architectures.

A competition hosted through Kaggle recently challenged participants to build classifiers for the $H \to \tau^+\tau^-$ decay task in Benchmark 3. The training dataset for this competition used slightly different features and only contained 250,000 training examples, which made overfitting a much greater challenge than on the datasets here.

It is also interesting that the Bayesian optimization for the supersymmetry and $H \to \tau^+\tau^-$ decay task chose deep networks with relatively few hidden units (100 and 274 units in each layer, respectively). This lends credence to the idea that deep neural networks have an advantage over shallow methods due to architectural constraints; these constraints might encourage the lower network layers to learn *general* features that can be re-used for many different data samples. Our experiments in Section 4 show that at least some of the *dark knowledge* contained in the deep networks can be used to train and improve the performance of shallow networks.

## 6. Methods

Computations were performed using machines with 16 Intel Xeon cores, NVIDIA Titan graphics processors, and 64 GB memory. All neural networks were trained using the GPU-accelerated Theano and Pylearn2 software libraries (Bergstra et al., 2010; Goodfellow et al., 2013).

## Acknowledgments

We are grateful to David Rousseau and Kyle Cranmer for their insightful comments, Michael Gelbart for his help with the Bayesian optimization, and Yuzo Kanomata for computing support. We also wish to acknowledge a hardware grant from NVIDIA, NSF grant IIS-0513376, and a Google Faculty Research award to PB.

## References

Johan Alwall et al. MadGraph 5 : Going Beyond. *JHEP*, 1106:128, 2011. doi: 10.1007/JHEP06(2011)128.

ATLAS Collaboration. ATLAS experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

ATLAS Collaboration. The ATLAS Simulation Infrastructure. *Eur.Phys.J.*, C70:823–874, 2010. doi: 10.1140/epjc/s10052-010-1429-9.

ATLAS Collaboration. Luminosity determination in pp collisions at $\sqrt{s} = 7$ TeV using the ATLAS Detector at the LHC. *Eur.Phys.J*, C73:2518, 2013a.

ATLAS Collaboration. Search for a Multi-Higgs Boson Cascade in $W^+Wb\bar{b}$ events with the ATLAS detector in pp collisions at s = 8 TeV. 2013b.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2654–2662. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf.

P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5, July 2014. doi: 10.1038/ncomms5308. URL http://www.nature.com/ncomms/2014/140702/ncomms5308/full/ncomms5308.html.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Enhanced higgs to $\tau^+\tau^-$ searches with deep learning. *Physics Review Letters*, 2015. In press.

Alan Barr, Christopher Lester, and P. Stephens. m(T2): The Truth behind the glamour. *J.Phys.*, G29:2343–2363, 2003. doi: 10.1088/0954-3899/29/10/304.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

James Bergstra, Olivier Breuleux, Frdric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Austin, TX, June 2010. Oral Presentation.

J. Beringer et al. Review of Particle Physics (RPP). *Phys.Rev.*, D86:010001, 2012. doi: 10.1103/PhysRevD.86.010001.

Matthew R. Buckley, Joseph D. Lykken, Christopher Rogan, and Maria Spiropulu. Super-Razor and Searches for Sleptons and Charginos at the LHC. 2013.

CDF Collaboration. Search for a two-Higgs-boson doublet using a simplified model in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV. *Phys.Rev.Lett.*, 110(12):121801, 2013. doi: 10.1103/PhysRevLett.110.121801.

Hsin-Chia Cheng and Zhenyu Han. Minimal Kinematic Constraints and m(T2). *JHEP*, 0812:063, 2008. doi: 10.1088/1126-6708/2008/12/063.

Dan Cirean, Ueli Meier, Jonathan Masci, and Jrgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32(0):333 – 338, 2012. ISSN 0893-6080. doi: http://dx.doi.org/10.1016/j.neunet.2012.02.023. URL http://www.sciencedirect.com/science/article/pii/S0893608012000524. Selected Papers from {IJCNN} 2011.

Glen Cowan et al. Asymptotic formulae for likelihood-based tests of new physics. *Eur. Phys. J*, C71(2):1–19, 2011. doi: 10.1140/epjc/s10052-011-1554-0.

A. Elagin, P. Murat, A. Pranko, and A. Safonov. A New Mass Reconstruction Technique for Resonances Decaying to di-tau. *Nucl.Instrum.Meth.*, A654:481–489, 2011. doi: 10.1016/j.nima.2011.07.009.

Jonathan L. Feng. Dark Matter Candidates from Particle Physics and Methods of Detection. *Ann.Rev.Astron.Astrophys.*, 48:495–545, 2010. doi: 10.1146/annurev-astro-082708-101659.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume JMLR W&CP 15, Fort Lauderdale, FL, USA, 2011.

Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.

Geoffrey E Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*, 2014.

Sepp Hochreiter. *Recurrent Neural Net Learning and Vanishing Gradient.* 1998.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, July 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL http://dx.doi.org/10.1016/0893-6080(89)90020-8.

K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, September 2009. doi: 10.1109/ICCV.2009.5459469.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, 10:1–40, 2009.

C. Lefevre. Lhc: The guide. 2009. URL http://cds.cern.ch/record/1165534.

S. Ovyn, X. Rouby, and V. Lemaitre. DELPHES, a framework for fast simulation of a generic collider experiment. 2009.

Planck Collaboration. Planck 2013 results. XVI. Cosmological parameters. 2013.

Christopher Rogan. Kinematical variables towards new dynamics at the LHC. 2010.

T. Sjostrand et al. PYTHIA 6.4 physics and manual. *JHEP*, 05:026, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

Steven Weinberg. A model of leptons. *Phys. Rev. Lett.*, 19:1264–1266, Nov 1967. doi: 10.1103/PhysRevLett.19.1264. URL http://link.aps.org/doi/10.1103/PhysRevLett.19.1264.