The Potential of Deep Learning with Astronomical Data

Chad M. Schafer
Department of Statistics
Carnegie Mellon University
June 2017
The LSST ISSC

- Informatics and Statistics one of eight LSST Science Collaborations
- Over 60 members and growing: data scientists and astronomers
- [http://issc.science.lsst.org](http://issc.science.lsst.org)
LSST Basics

• 10-year photometric survey
• 3.2 Gigapixel camera
• 32 trillion observations of 40 billion objects

• Science Goals
  – Cataloging the Solar System
  – Exploring the Changing Sky
  – Milky Way Structure & Formation
  – Understanding Dark Matter and Dark Energy

Ivezić, et al. (2014)
Common Themes

• General implementation challenges
• Existing procedures to LSST scales
• Expanding sophistication of analysis procedures in use
• Making the most of available data
Representations

• A recurring challenge is representing observables in forms amenable to standard analysis tools

• The fundamental challenge of “Big Data”
Representations

- A recurring challenge is representing observables in forms amenable to standard analysis tools.

A Letter to the NSF Astronomy Portfolio Review:
LSST is Not “Big Data”

David Schlegel (Lawrence Berkeley National Lab)
31 January 2012

LSST promises to be the largest optical imaging survey of the sky. If we were fortunate enough to have the equivalent of LSST today, it would represent a “fire hose” of data that would be difficult to store, transfer, and analyze with available compute resources.

LSST parallels the SDSS compute task which was ambitious yet tractable. By almost any measure relative to computers that will be available (thanks to the steady progression of Moore’s Law), LSST will be a small data set. LSST will never fill more than 22 hard drives. Individual investigators will be able to maintain their own data copies to analyze as they choose.
Representations

What summary statistic retains the important information for estimating parameters of interest?
Representations

What summary statistic retains the important information for estimating parameters of interest?
Representations

What summary statistic retains the important information for estimating parameters of interest?

Figure 6. The measured shear correlation functions $\xi_+$ (black squares) and $\xi_-$ (blue circles), combined from all four Wide patches. The error bars correspond to the total covariance diagonal. Negative values are shown as thin points with dotted error bars. The lines are the theoretical prediction using the WMAP7 best-fitting cosmology and the non-linear model described in Sect. 4.3. The data points and error bars are listed in Table B1.
Representations

What summary statistic retains the important information for estimating parameters of interest?

Kilbinger, et al., CFHTLenS Results

Image: NASA/ESA Kilbinger, et al., CFHTLenS Results

flat $\Lambda$CDM

$\sigma_8$

$\Omega_m$

CFHTLenS

WMAP7

CFHTLenS+WMAP7

CFHTLenS+WMAP7+BOSS+R09

Image: NASA/ESA
Representations

What features are most useful for classifying objects?
Classifying Variables

Variability Tree

Asteroids
- Rotation
- Eclipse

Stars
- Micro-lensing
- Rotation

Extrinsic

Intrinsic

AGN

Eruptions

Planetary transits

Eclipsing binary

Asteroid occultation

Figure 1. Variability tree. A tentative organization of variable objects.

Credit: L. Eyer & N. Mowlavi (2008)

Eyer and Mowlavi (2008)
Classifying Variables

- SN
- not SN

  - SN - I
  - SN - II
  - Collapsing
  - CV / Blazar
  - RR / Mira

- CV
- Blazar
- RR Lyrae
- Mira
  - Cataclysmic
  - Periodic
Blazars versus CVs

Cataclysmic Variables (CV) – binary system in Milky Way with matter transfer from secondary (normal) star to primary white dwarf.

Blazars – Quasars with “jet” of energy pointed at Earth.

Both produce light curves with irregular variability, lacking periodic structure.
Blazars versus CVs

Light Curves from Catalina Real-Time Transient Survey (Drake 2009)
Blazars versus CVs

Comparison of Structure Functions
Summarizing the SF

Typical to fit model to structure function

- Power Law Form (Schmidt et al.)
- Damped Random Walk (Kelly et al.)

Effort to find a low-dimensional representation, avoiding the curse of dimensionality
Summarizing the SF

Figure 2 in Peters et al. Quasar light curve and SF
Summarizing the SF

Typical to fit model to structure function

• Power Law Form (Schmidt et al.)
• Damped Random Walk (Kelly et al.)

Effort to find a low-dimensional representation, avoiding the curse of dimensionality

Ideally, could utilize higher-dimensional representation
Deep Learning

“Deep learning is a particular kind of machine learning that achieves great power and flexibility by representing the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.”

--Page 8 in Deep Learning, Goodfellow, Bengio, and Courville
Deep Learning

“Deep learning is a particular kind of machine learning that achieves great power and flexibility by representing the world as a nested hierarchy, with each concept defined in terms of simpler concepts, and more abstract representations computed in terms of less abstract ones.”

--Page 8 in Deep Learning, Goodfellow, Bengio, and Courville
Deep Learning

Input nodes
Connections
Hidden nodes
Output nodes
Deep Learning

What makes it “deep?”
Deep Learning

What makes it “deep?”

The number of hidden layers is typically large, allowing for the modeling of complex relationships.
Deep Learning

What makes it “deep?”

The number of hidden layers is typically large, allowing for the modeling of complex relationships.

Isn’t this just a neural network?
Deep Learning

What makes it “deep?”
The number of hidden layers is typically large, allowing for the modeling of complex relationships.

Isn’t this just a neural network?
Yes, basically.
Resurgence of ANN

Multiple factors contributed to growth of interest in Deep Learning:

• Increase in training set sizes
• Improved algorithms for training deeper networks (e.g., Hinton, et al. in 2006)
• Growth in computational resources
• Successes
Flexibility

A primary appeal of the approach is the flexibility in constructing the layers

– How many units are there in each layer?
– What is the mapping from one layer to the next?
– How is the output constructed from the final hidden layer?
Flexibility

A primary appeal of the approach is the flexibility in constructing the layers

– How many units are there in each layer?
– What is the mapping from one layer to the next?
– How is the output constructed from the final hidden layer?
Fully Connected Layer

A standard mapping is a fully connected layer, simply a linear combination of the input (either the data or the output of the preceding layer)!
Input Layer

\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ b + \sum_{i=1}^{d} w_i x_i \]
$\mathbf{x} = (x_1, x_2, \ldots, x_d)$

$b + \mathbf{w}^T \mathbf{x}$
Input Layer

\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ b_1 + \mathbf{w}_1^T \mathbf{x} \quad b_2 + \mathbf{w}_2^T \mathbf{x} \]
Input Layer

\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ b_1 + \mathbf{w}_1^T \mathbf{x} \quad b_2 + \mathbf{w}_2^T \mathbf{x} \quad \ldots \quad b_m + \mathbf{w}_m^T \mathbf{x} \]
\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ \mathbf{u} = (u_1, u_2, \ldots, u_{m_1}) \]
\[ x = (x_1, x_2, \ldots, x_d) \]

\[ u = (u_1, u_2, \ldots, u_{m_1}) \]

\[ b_1 + w_1^T u \quad b_2 + w_2^T u \quad \cdots \quad b_{m_2} + w_{m_2}^T u \]
\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ \phi(b_1 + \mathbf{w}_1^T \mathbf{x}) \quad \phi(b_2 + \mathbf{w}_2^T \mathbf{x}) \quad \ldots \quad \phi(b_m + \mathbf{w}_m^T \mathbf{x}) \]

\( \phi(\cdot) \) is the activation function, a simple nonlinear mapping
rectified linear

\[ \phi(u) = \max(0, u) \]

hyperbolic tangent

\[ \phi(u) = \tanh(u) \]

logistic sigmoid

\[ \phi(u) = \frac{1}{1 + \exp(-u)} \]

softplus

\[ \phi(u) = \log(1 + \exp(u)) \]
\[
\mathbf{x} = (x_1, x_2, \ldots, x_d)
\]

\[
\phi(b_1 + \mathbf{w}_1^T \mathbf{x}) \quad \phi(b_2 + \mathbf{w}_2^T \mathbf{x}) \quad \cdots \quad \phi(b_m + \mathbf{w}_m^T \mathbf{x})
\]

Input Layer
Input Layer

\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

First Layer

\[ \mathbf{u} = (u_1, u_2, \ldots, u_{m_1}) \]
\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ \mathbf{u} = (u_1, u_2, \ldots, u_{m_1}) \]

\[ \phi(b_1 + \mathbf{w}_1^T \mathbf{u}) \quad \phi(b_2 + \mathbf{w}_2^T \mathbf{u}) \quad \cdots \quad \phi(b_{m_2} + \mathbf{w}_{m_2}^T \mathbf{u}) \]
\[ \mathbf{x} = (x_1, x_2, \ldots, x_d) \]

\[ \mathbf{u} = (u_1, u_2, \ldots, u_{m_1}) \]

\[ \phi(b_1 + \mathbf{w}_1^T \mathbf{u}) \quad \phi(b_2 + \mathbf{w}_2^T \mathbf{u}) \quad \cdots \quad \phi(b_{m_2} + \mathbf{w}_{m_2}^T \mathbf{u}) \]

\[ \vdots \]
\[ x = (x_1, x_2, \ldots, x_d) \]

\[ u = (u_1, u_2, \ldots, u_{m_1}) \]

\[ \phi(b_1 + w_{1}^T u) \quad \phi(b_2 + w_{2}^T u) \quad \ldots \quad \phi(b_{m_2} + w_{m_2}^T u) \]

...
Output Layer

There are standard choices for generating the output from the final hidden layer.
Output Layer

There are standard choices for generating the **output** from the final hidden layer.

If the output is **continuous**, then simply taking a linear combination is typical:

\[ y = b + w^T u \]
Output Layer

There are standard choices for generating the output from the final hidden layer!

If the output is continuous, then simply taking a linear combination is typical:

\[ y = b + w^T u \]
Output Layer

If the output is **binary**, then transformation to a probability is done via the **logistic sigmoid function**:

\[ y = \frac{1}{1 + \exp(-(b + w^T u))} \]
If the output is \textit{multinomial}, then transformation to a probability is done via the \textit{softmax} function:

\[
\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}
\]

where

\[
z = W^T u + b
\]
Some Code

R using package **mxnet:**

```r
fc1 = mx.symbol.FullyConnected(data, name="fc1", num_hidden=128)
act1 = mx.symbol.Activation(fc1, name="relu1", act_type="relu")
fc2 = mx.symbol.FullyConnected(act1, name="fc2", num_hidden=128)
act2 = mx.symbol.Activation(fc2, name="relu2", act_type="relu")
fc3 = mx.symbol.FullyConnected(act2, name="fc3", num_hidden=2)
fullnetwork = mx.symbol.SoftmaxOutput(fc3, name="sm")
```
Flexibility

A primary appeal of the approach is the **flexibility** in constructing the layers

- How many units are there in each layer?
- What is the **mapping** from one layer to the next?
- How is the output constructed from the final hidden layer?
Flexibility

A primary appeal of the approach is the flexibility in constructing the layers

– How many units are there in each layer?
– What is the mapping from one layer to the next?
– How is the output constructed from the final hidden layer?

There are alternatives to fully connected layers, e.g. convolutional networks and recurrent networks
How Does it Work?

Instead of carefully constructing a model to relate the input to the output, deep learning exploits a large collection of simple components to make a prediction.

What is the role of expert knowledge?
How Does it Work?

Universal Approximation Theorem (Hornik, et al.): With enough units, a single hidden layer can approximate to arbitrary precision any “nice” function.

But: Deeper networks use units more efficiently, are easier to fit, and generalize better.
How Does it Work?

But: Deeper networks use units more efficiently, are easier to fit, and generalize better.

Montufar, et al.: “[f]or deep models, the maximal number of linear regions grows exponentially fast with the number of parameters, whereas, for shallow models, it grows polynomially fast with the number of parameters.”
Fitting the Model

A cost function is optimized to estimate the parameters (weights)

Choose cost function to maximize appropriate likelihood

Stochastic gradient descent with back propagation to estimate gradient
Regularization

Overfitting is a huge concern

Approaches to regularization (smoothing) manage the bias/variance tradeoff

The model is parametric, so $L^2$ (ridge) or $L^1$ (lasso) penalties on the cost function are commonly used
Regularization

**Dropout** is a novel approach to regularization.

Units are randomly included/excluded during training, approximating averaging over all possible submodels.

Variant of bagging.

Reduces potential influence of any individual unit.
Blazars versus CVs

Comparison of Structure Functions
Blazars versus CVs

Quantile regression fits

Log Magnitude Difference

Absolute Time Difference
Blazar versus CV

Fit model with three hidden layers, using Dropout

128 nodes per layer

Rectified linear units as the activation functions

958 CVs, 318 Blazars from Catalina Real-Time Transient Survey
Blazar versus CV

Performance on test set:

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Truth</th>
<th>Blazar</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blazar</td>
<td></td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>CV</td>
<td></td>
<td>8</td>
<td>91</td>
</tr>
</tbody>
</table>
## Blazar versus CV

Performance on test set:

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Deep Learning</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Blazar</td>
<td>CV</td>
</tr>
<tr>
<td>Blazar</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>CV</td>
<td>8</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Random Forest</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Blazar</td>
<td>CV</td>
</tr>
<tr>
<td>Blazar</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>CV</td>
<td>14</td>
<td>93</td>
</tr>
</tbody>
</table>
Potential of Deep Learning

Best suited to situations where high-dimensional input is required

Avoid the curse of dimensionality

Seems particularly relevant for classification challenges
Quasar Classification
Quasar Classification

Log
Magnitude
Difference

Log
Time
Difference

Log Magnitude Difference

Log Time Difference
References

Goodfellow, Bengio, and Courville. Deep Learning
Hornik, et al. Neural Networks (3): 551
Ivezic, et al. arXiv 0805.2366
Kilbinger, et al. MNRAS (430):2200
Montufar, et al. NIPS 2014
Schlegel. arXiv 1203.0591