The Potential of Deep Learning with Astronomical Data

Chad M. Schafer
Department of Statistics
Carnegie Mellon University

June 2017

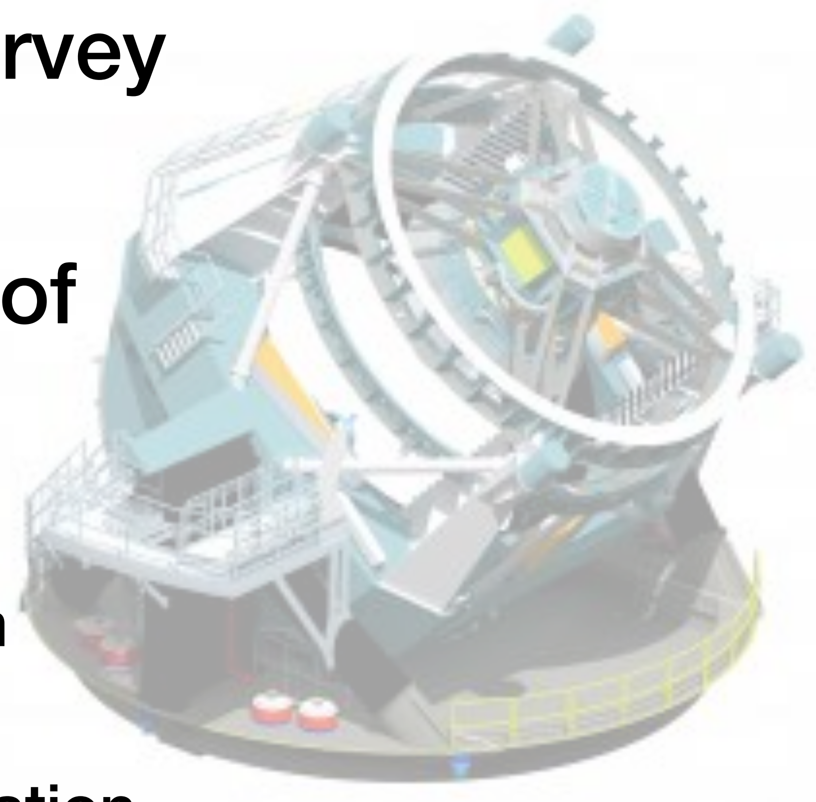
The LSST ISSC

Informatics & Statistics
Science Collaboration

- Informatics and Statistics one of eight LSST Science Collaborations
- Over 60 members and growing:
  data scientists and astronomers
- 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

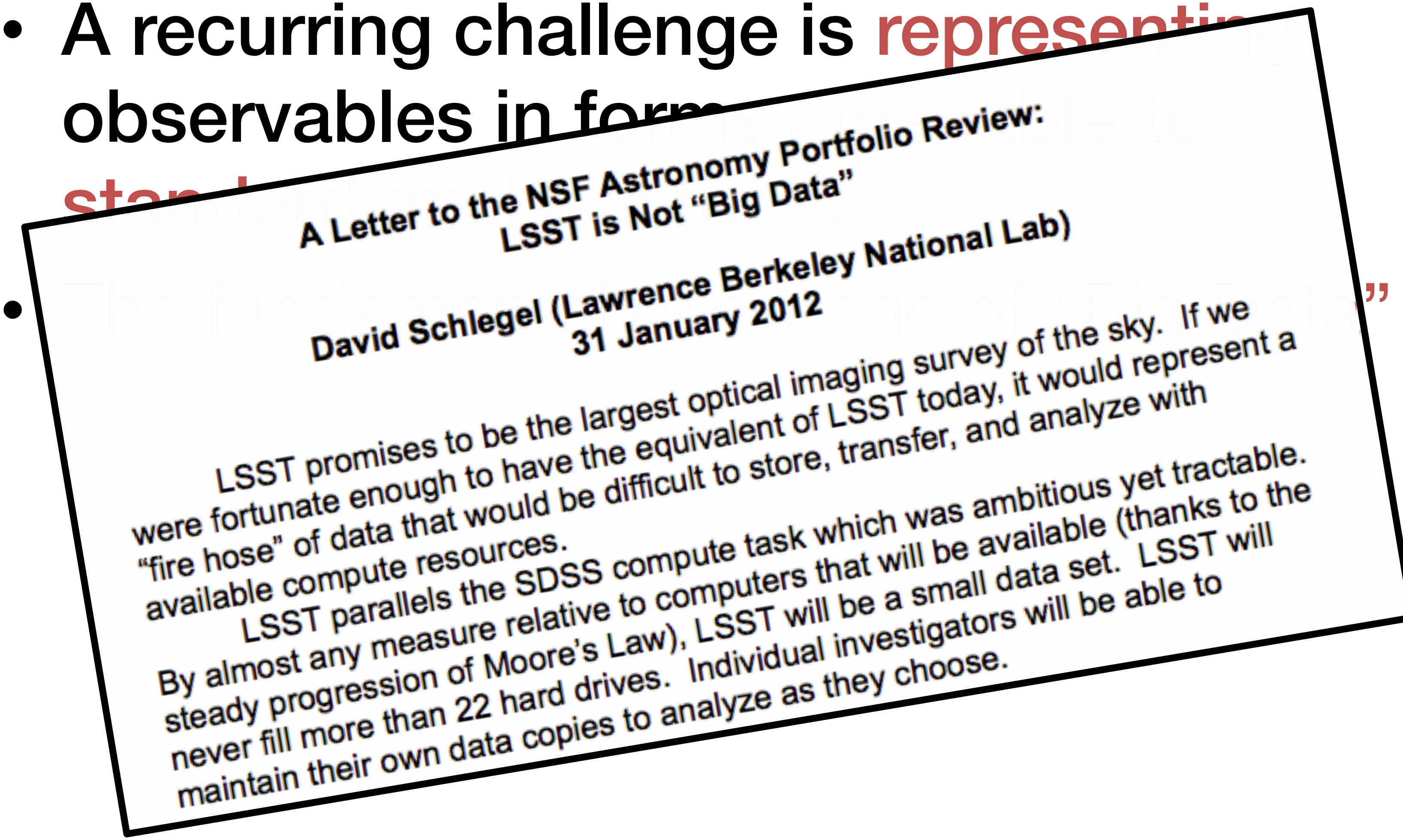
- General implementation challenges
- Existing procedures to LSST scales
- Expanding sophistication of analysis procedures in use
- Making the most of available data

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
- The fundamental challenge of "Big Data"



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?

Image: NASA/ESA

Representations

What s… e
importa… ng
parame…

Kilbinger, et al., CFHTLenS Results

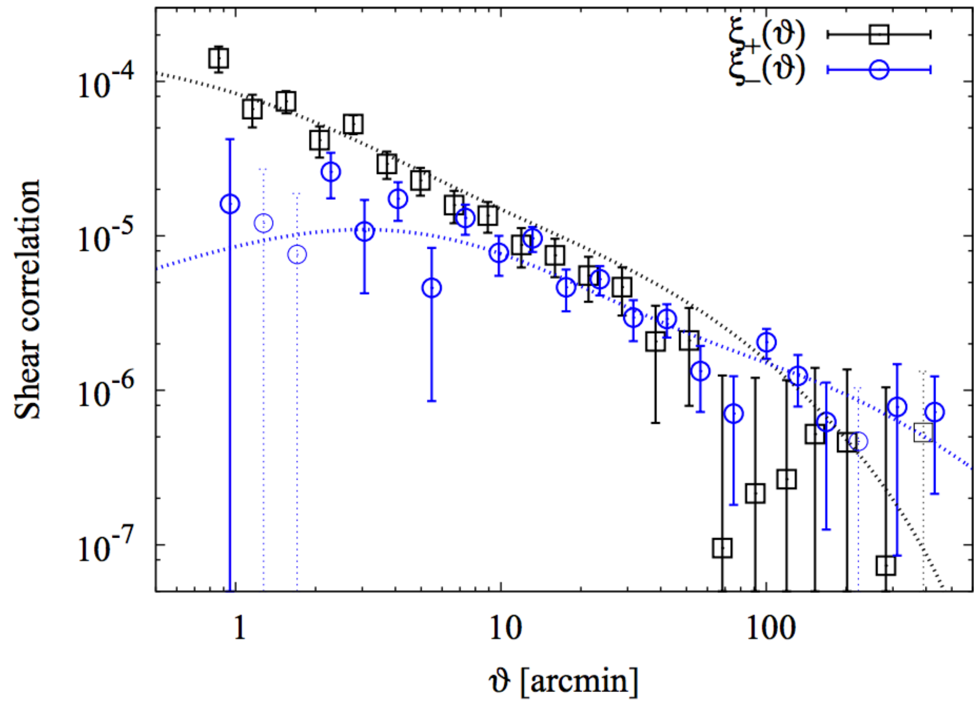


Figure 6. The measured shear correlation functions ξ_+ (black squares) and ξ_- (blue circles), combined from all four Wide patches. The error bars correspond to the total covariance diagonal. Negative values are shown as thin points with dotted error bars. The lines are the theoretical prediction using 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.

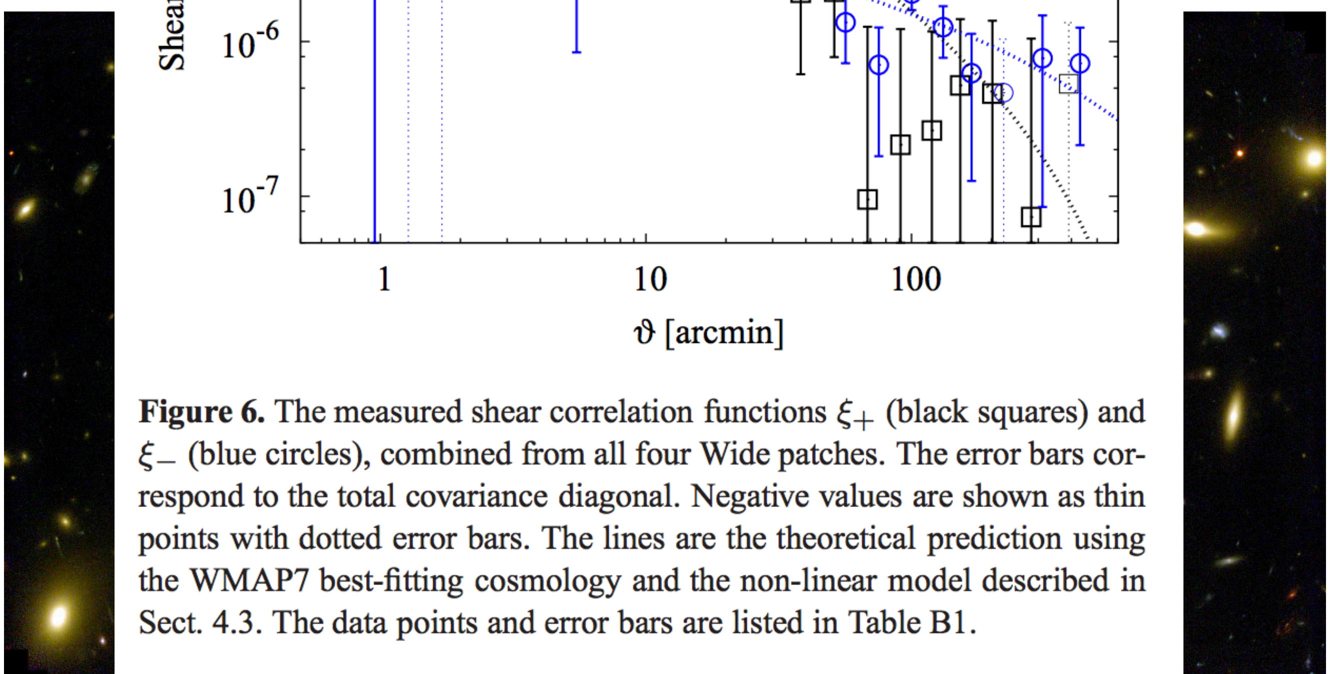
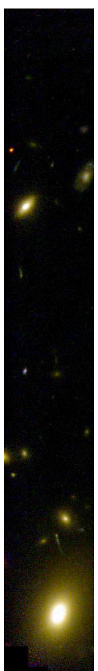


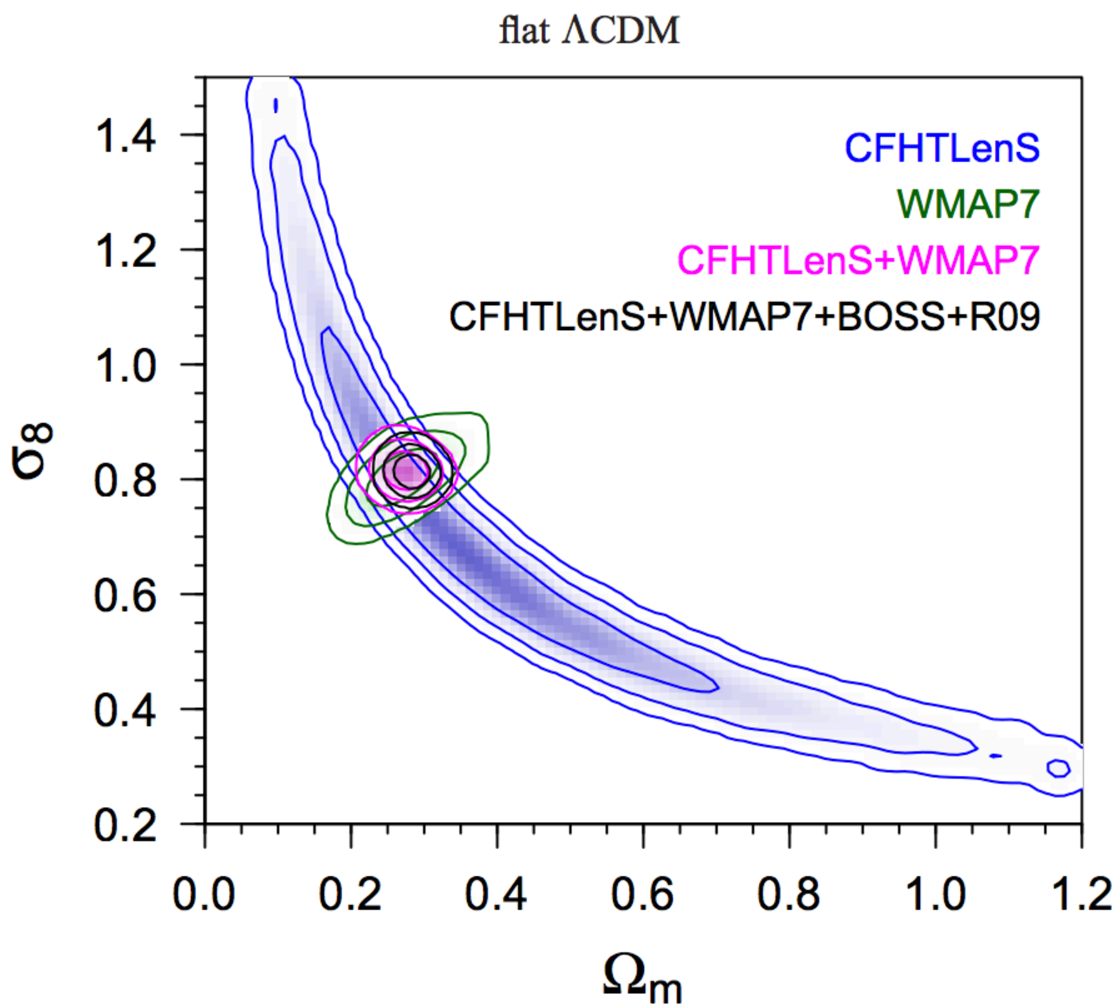
Image: NASA/ESA

Representations

Kilbinger, et al., CFHTLenS Results

What s
import
parame





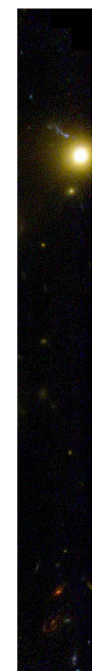
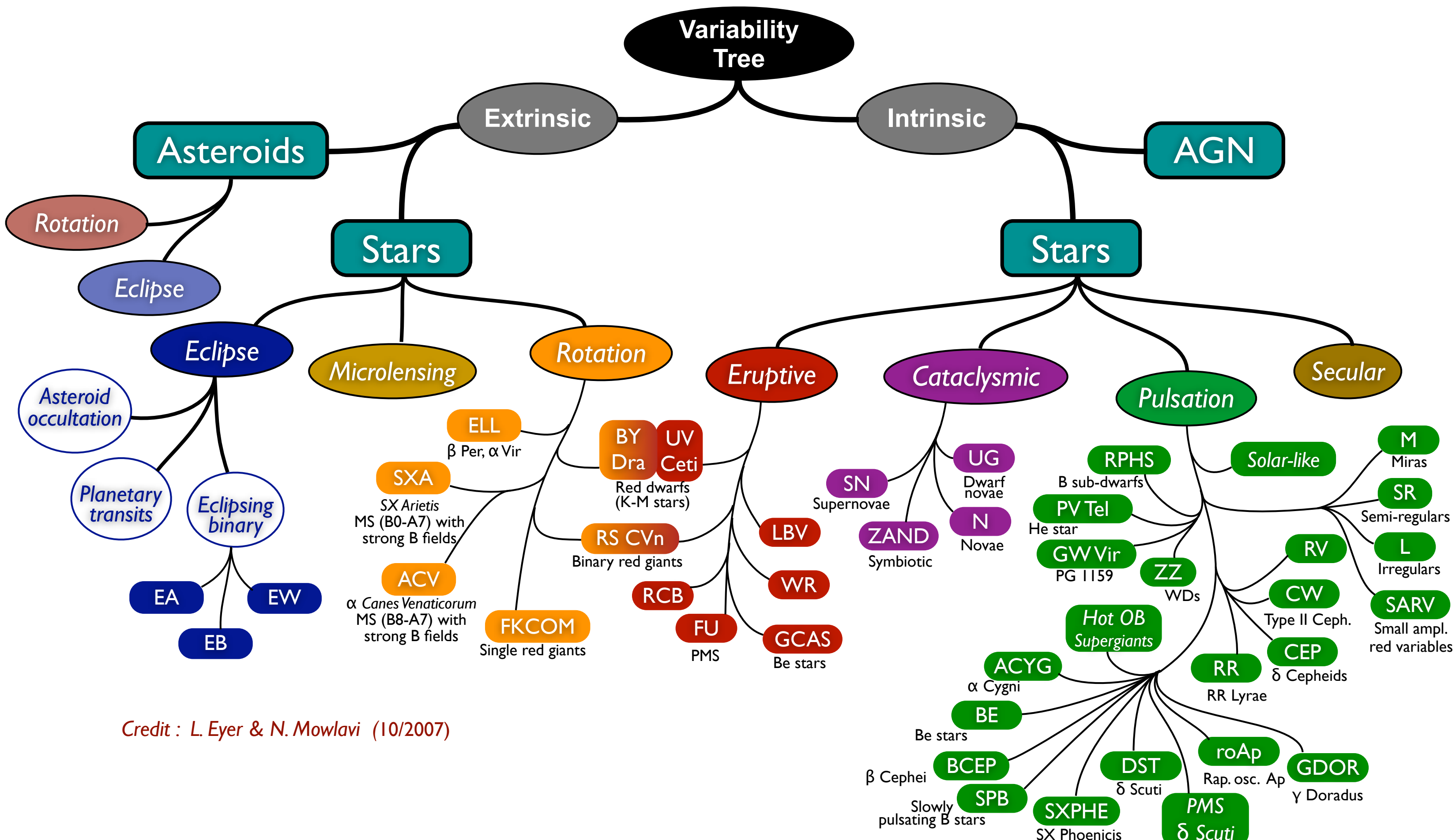


Image: NASA/ESA

Representations

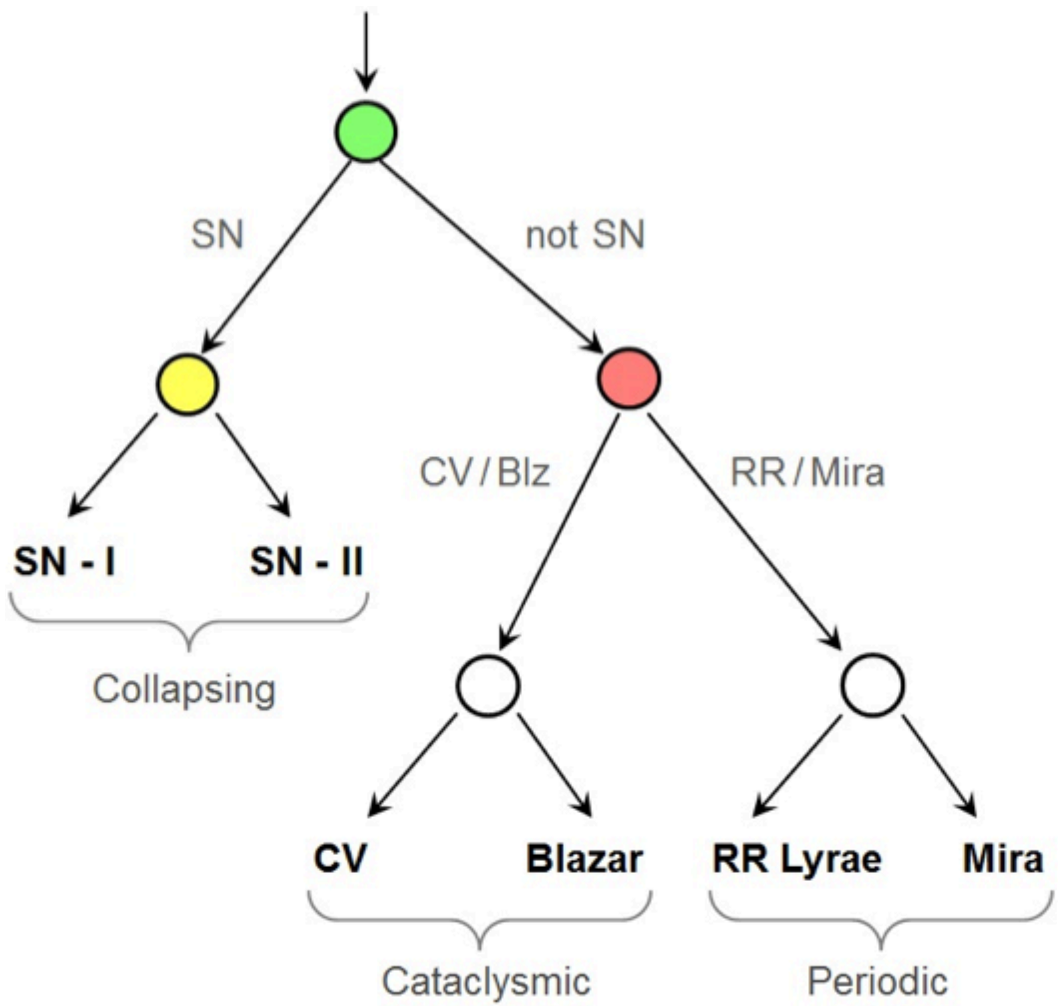
What features are most useful for classifying objects?

Classifying Variables



Eyer and Mowlavi (2008)

Classifying Variables



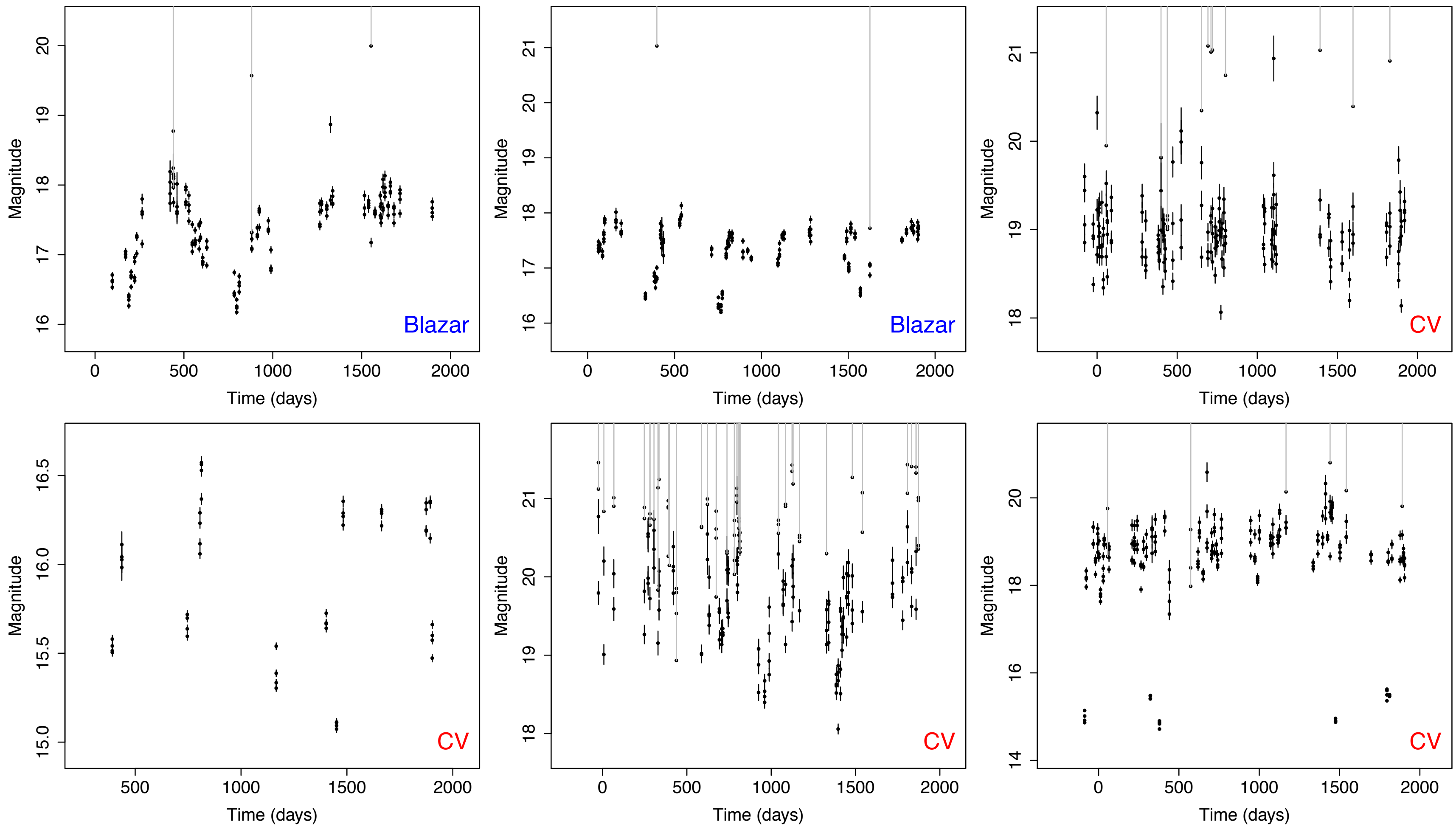
Blazars versus CVs

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

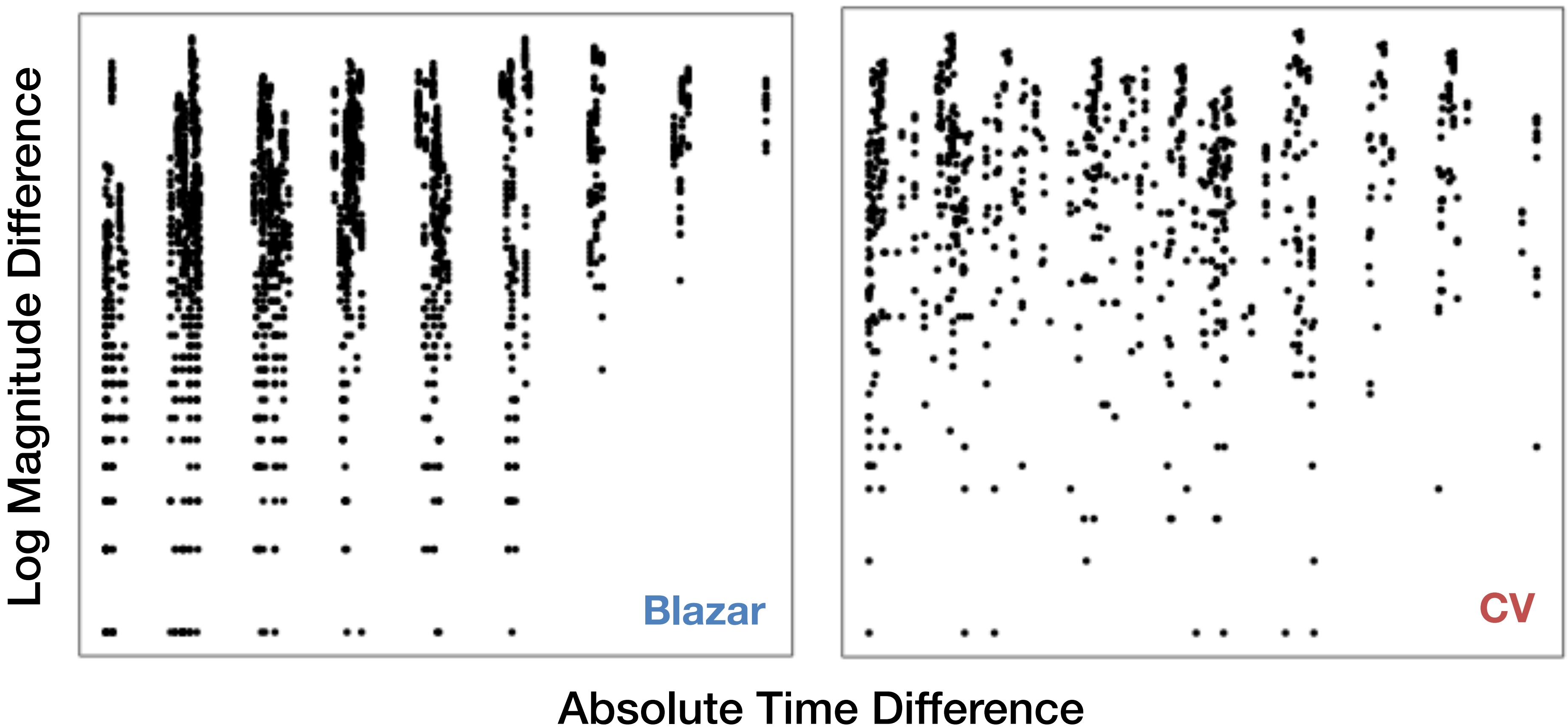
Both produce light curves with irregular variability, lacking periodic structure

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

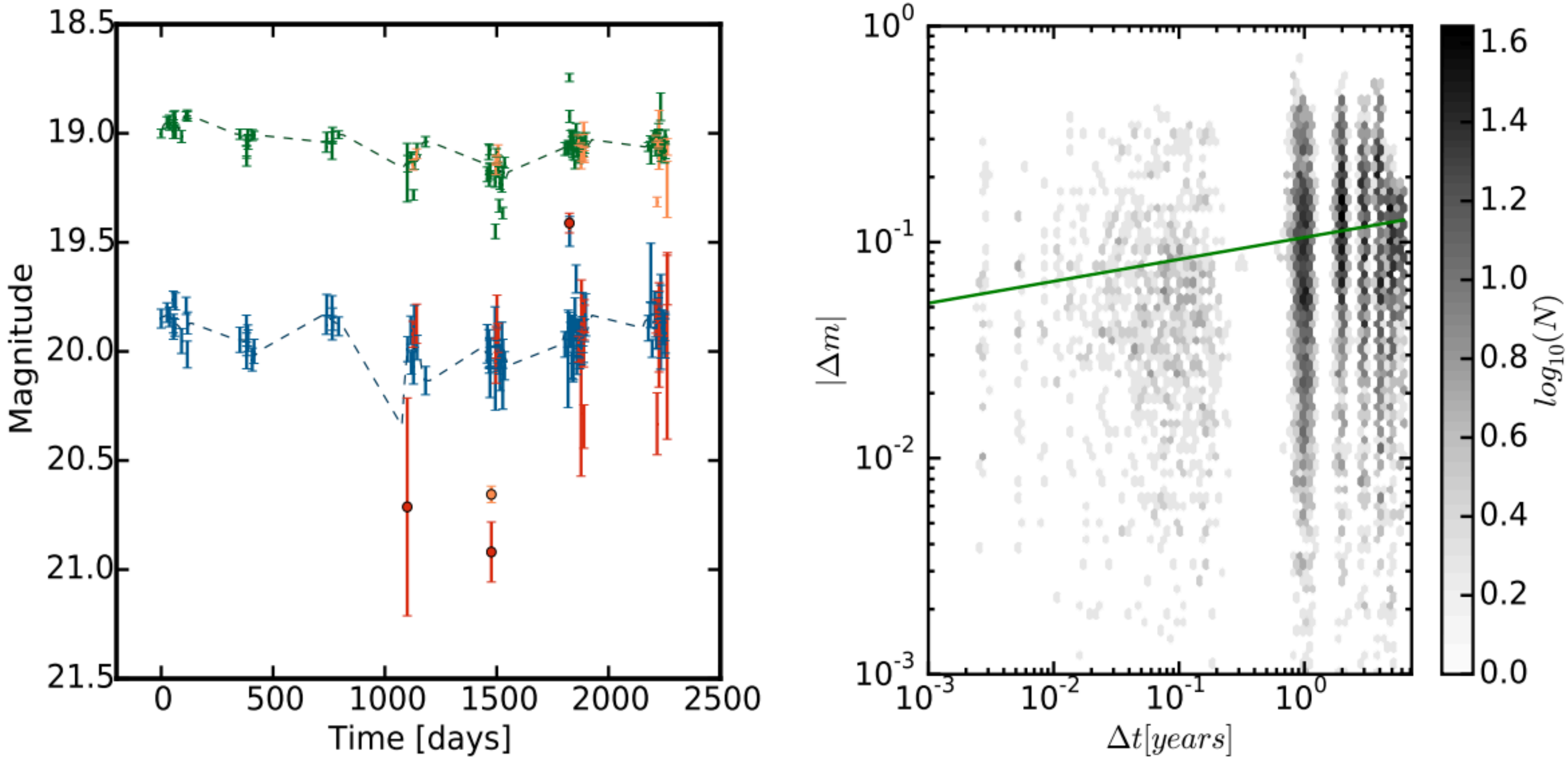


Figure 2 in Peters et al. Quasar light curve and SF

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

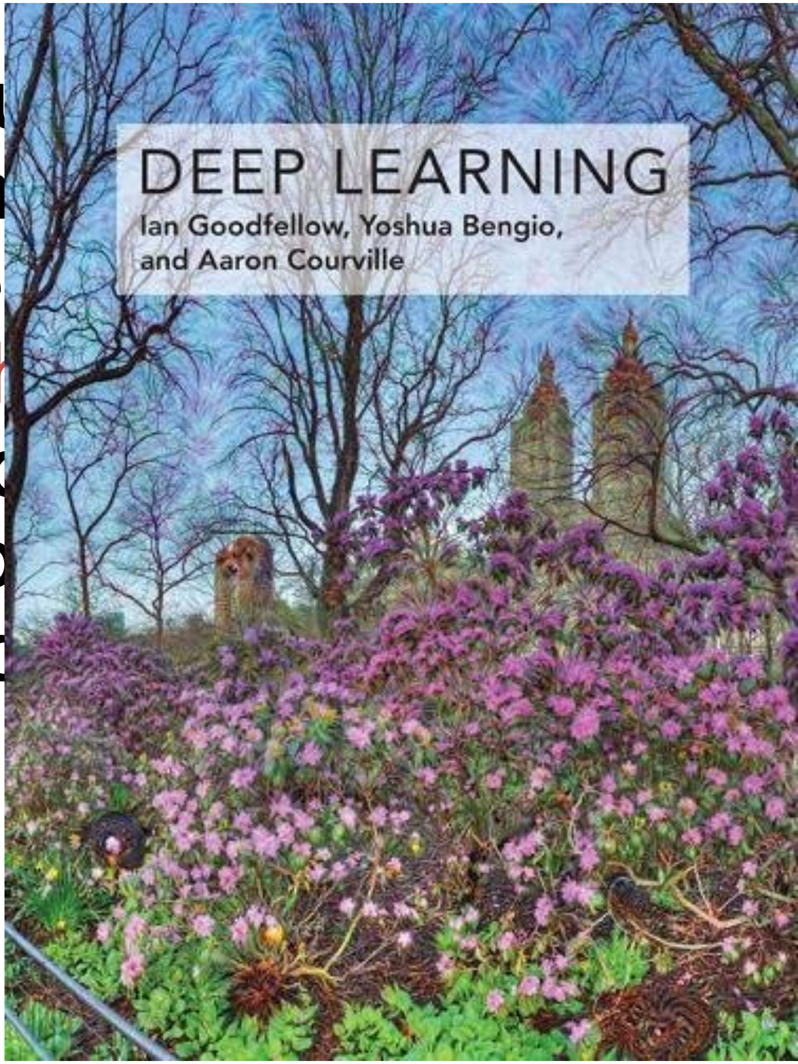
"Deep learning is a particular kind of machine learning that achieves great power and flexibility by representing the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones."

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

Deep Learning

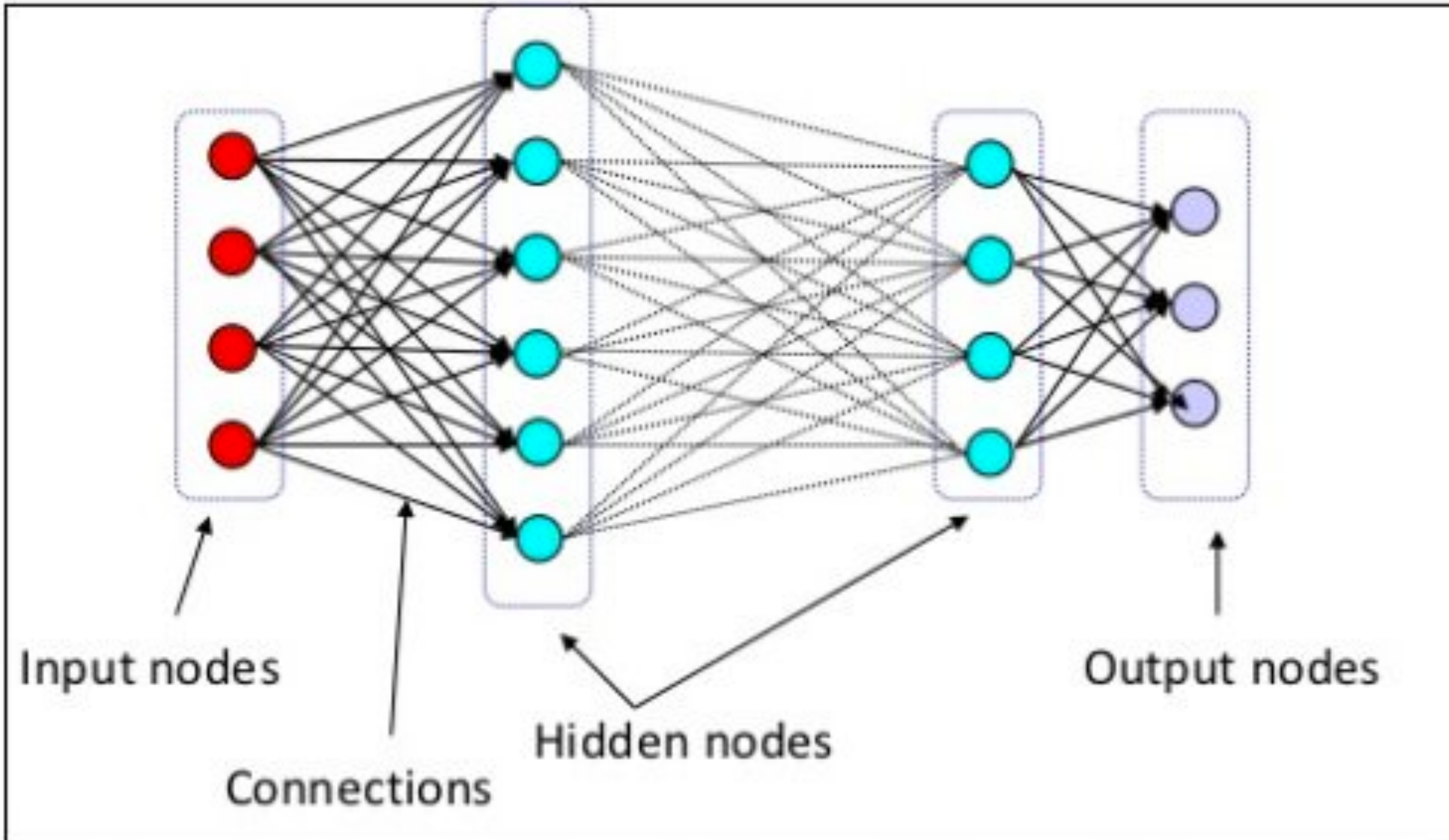
"Deep learning is a partic
machine learning that ach
power and flexibility by re
world as a nested hierarch
with each concept definec
simpler concepts, and mo
representations computec
less abstract ones."

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



www.deeplearningbook.org

Deep Learning



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

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

$$b + \sum_{i=1}^{d} w_i x_i$$

Input
Layer

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

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

Input Layer

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

$$b_1 + \mathbf{w}_1^T \mathbf{x} \qquad b_2 + \mathbf{w}_2^T \mathbf{x}$$

Input Layer

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

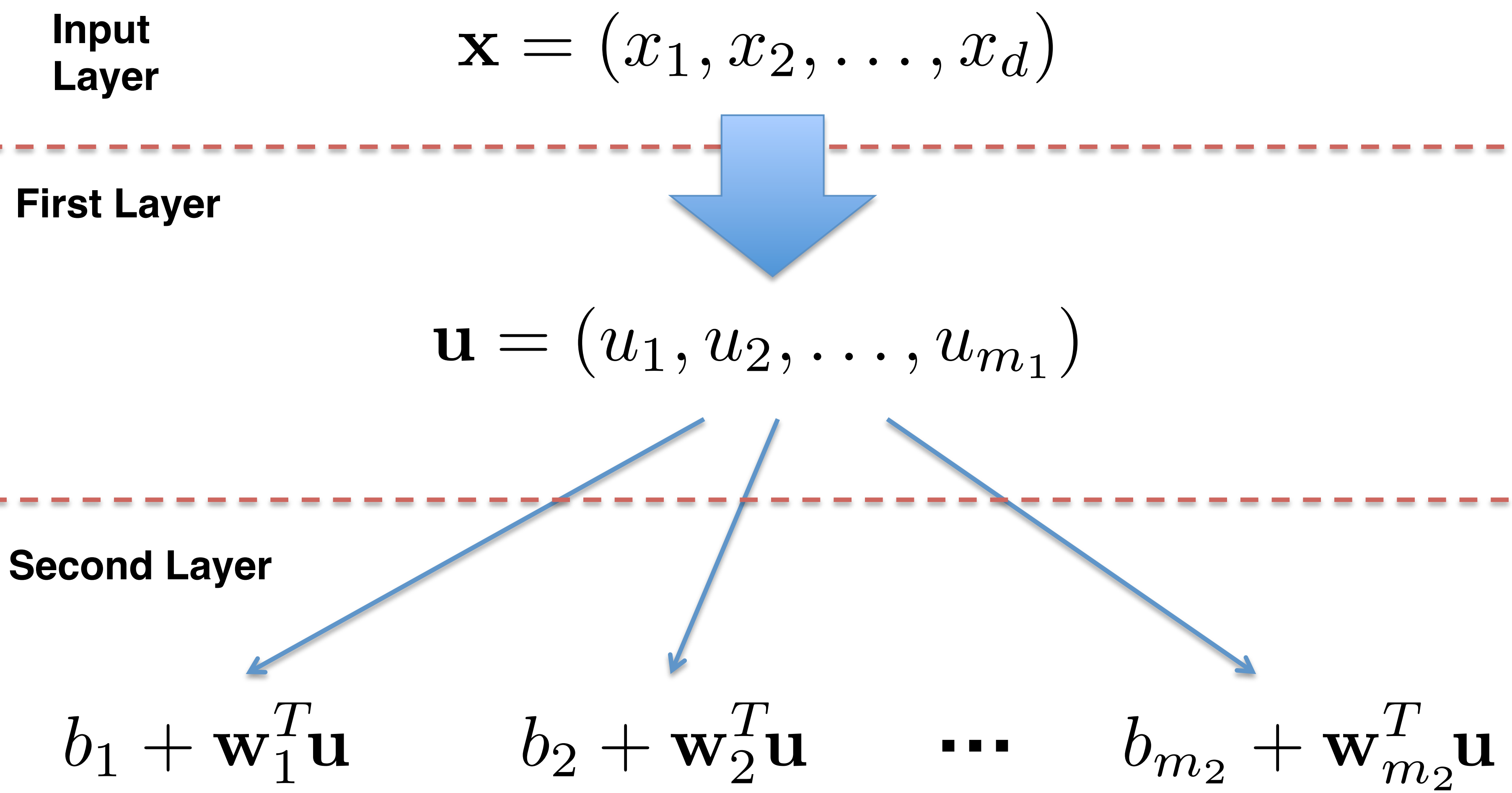
$$b_1 + \mathbf{w}_1^T\mathbf{x} \qquad b_2 + \mathbf{w}_2^T\mathbf{x} \qquad \cdots \qquad b_m + \mathbf{w}_m^T\mathbf{x}$$

Input Layer

$$\mathbf{x} = (x_1, x_2, \dots, x_d)$$

First Layer

$$\mathbf{u} = (u_1, u_2, \dots, u_{m_1})$$

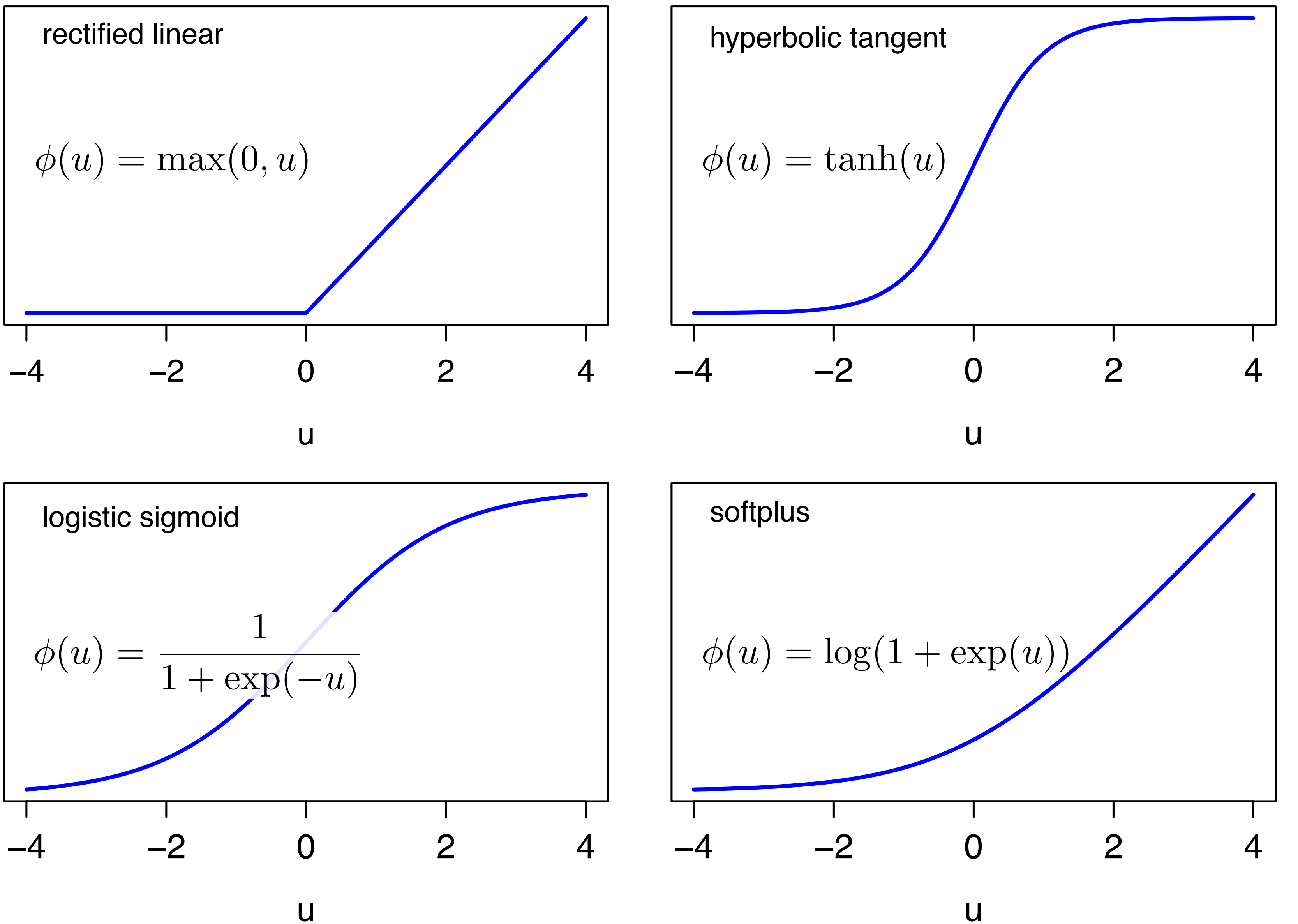

Input Layer
$\mathbf{x} = (x_1, x_2, \ldots, x_d)$
First Layer
$\mathbf{u} = (u_1, u_2, \ldots, u_{m_1})$
Second Layer
$b_1 + \mathbf{w}_1^T \mathbf{u}$
$b_2 + \mathbf{w}_2^T \mathbf{u}$
$\cdots$
$b_{m_2} + \mathbf{w}_{m_2}^T \mathbf{u}$

Input Layer

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

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

$\phi(\cdot)$ 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
$\phi(u) = \log(1 + \exp(u))$
–4
–2
0
2
4
u

Input Layer

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

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

Input Layer

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

First Layer

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

Input Layer

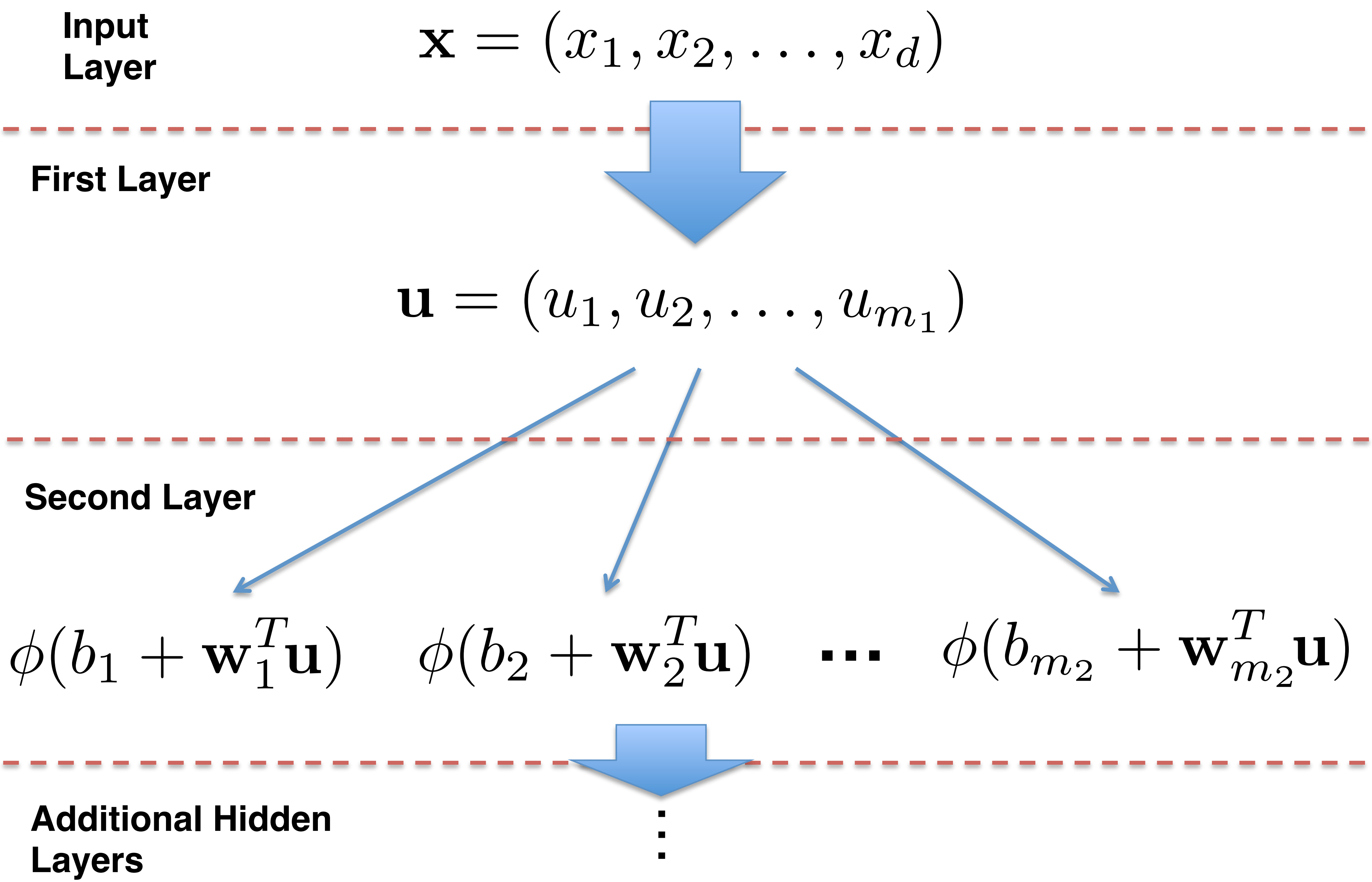
$$\mathbf{x} = (x_1, x_2, \ldots, x_d)$$

First Layer

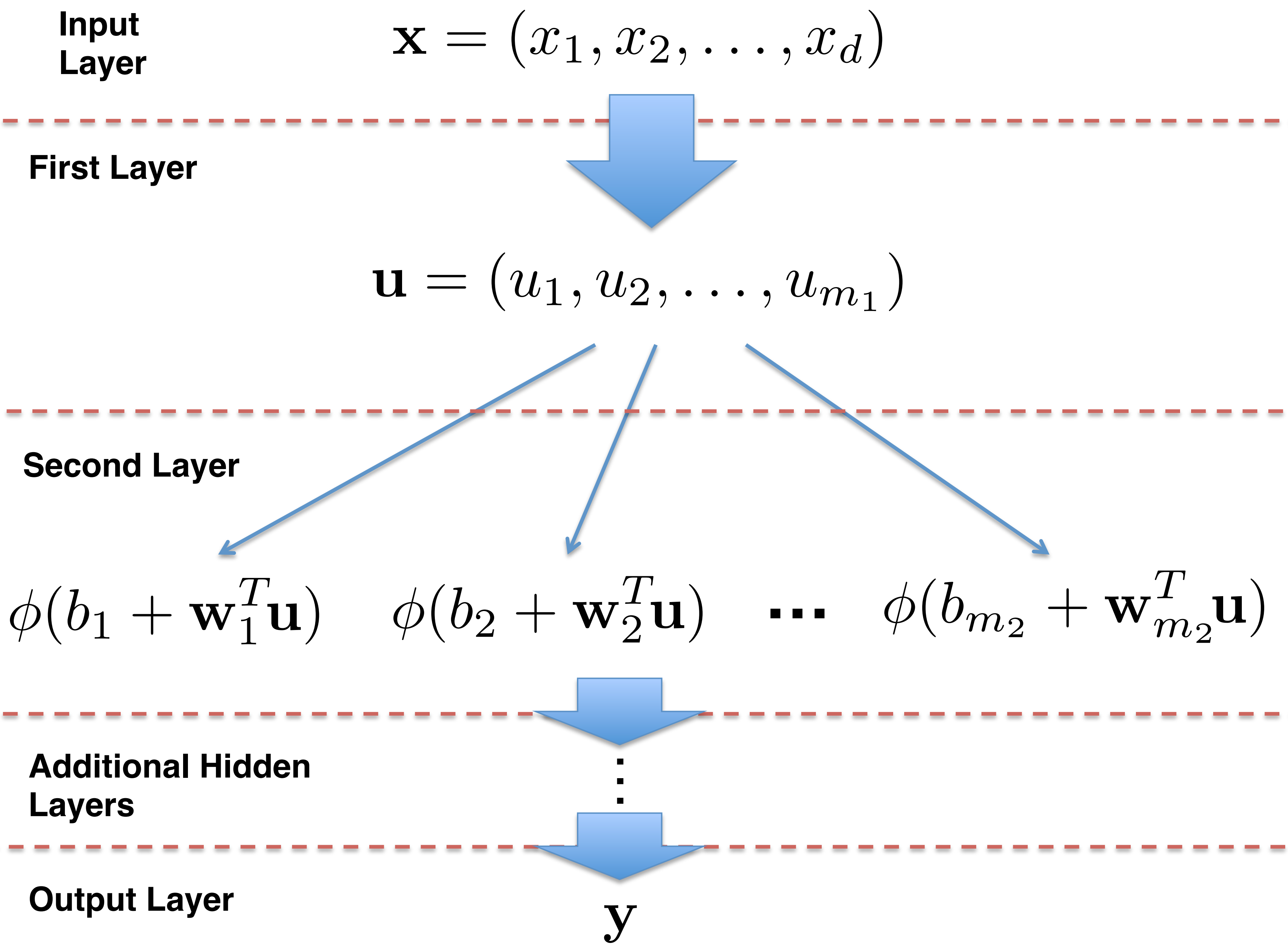
$$\mathbf{u} = (u_1, u_2, \ldots, u_{m_1})$$

Second Layer

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



Input Layer
$\mathbf{x} = (x_1, x_2, \ldots, x_d)$
First Layer
$\mathbf{u} = (u_1, u_2, \ldots, u_{m_1})$
Second Layer
$\phi(b_1 + \mathbf{w}_1^T \mathbf{u})$
$\phi(b_2 + \mathbf{w}_2^T \mathbf{u})$
$\cdots$
$\phi(b_{m_2} + \mathbf{w}_{m_2}^T \mathbf{u})$
Additional Hidden Layers
$\vdots$


Input Layer
$\mathbf{x} = (x_1, x_2, \ldots, x_d)$
First Layer
$\mathbf{u} = (u_1, u_2, \ldots, u_{m_1})$
Second Layer
$\phi(b_1 + \mathbf{w}_1^T\mathbf{u})$
$\phi(b_2 + \mathbf{w}_2^T\mathbf{u})$
$\cdots$
$\phi(b_{m_2} + \mathbf{w}_{m_2}^T\mathbf{u})$
Additional Hidden Layers
Output Layer
$\mathbf{y}$

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:

$$\mathbf{y} = b + \mathbf{w}^T \mathbf{u}$$

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:

$$\mathbf{y} = b + \mathbf{w}^T \underbrace{\mathbf{u}}_{\text{Result of final hidden layer}}$$

Output Layer

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

$$\mathbf{y} = \frac{1}{1 + \exp(-(b + \mathbf{w}^T \mathbf{u}))}$$

Output Layer

If the output is multinomial, then transformation to a probability is done via the softmax function:

$$\mathrm{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

where

$$\mathbf{z} = \mathbf{W}^T\mathbf{u} + \mathbf{b}$$

Some Code

R using package mxnet:

```
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

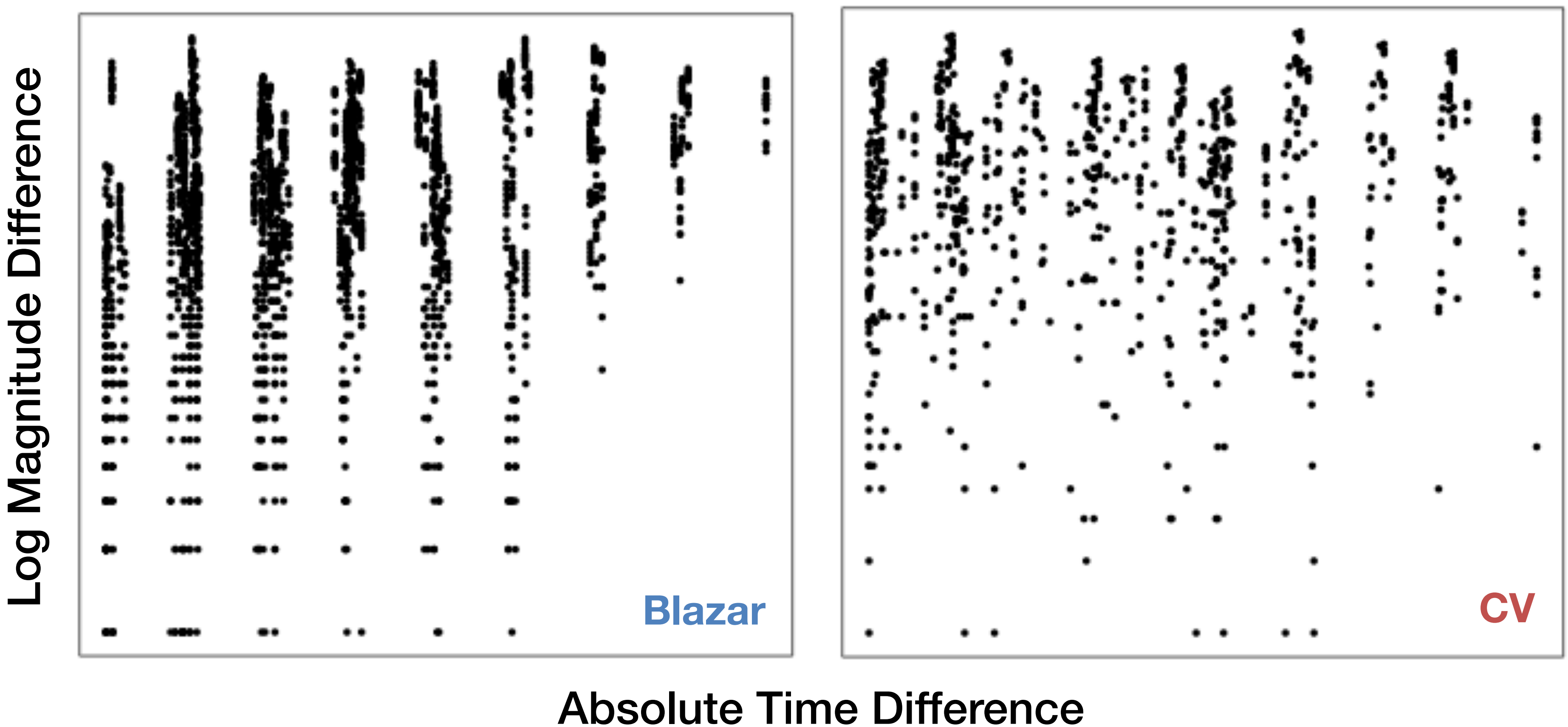
Dropout is a novel approach to regularization

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

Variant of bagging

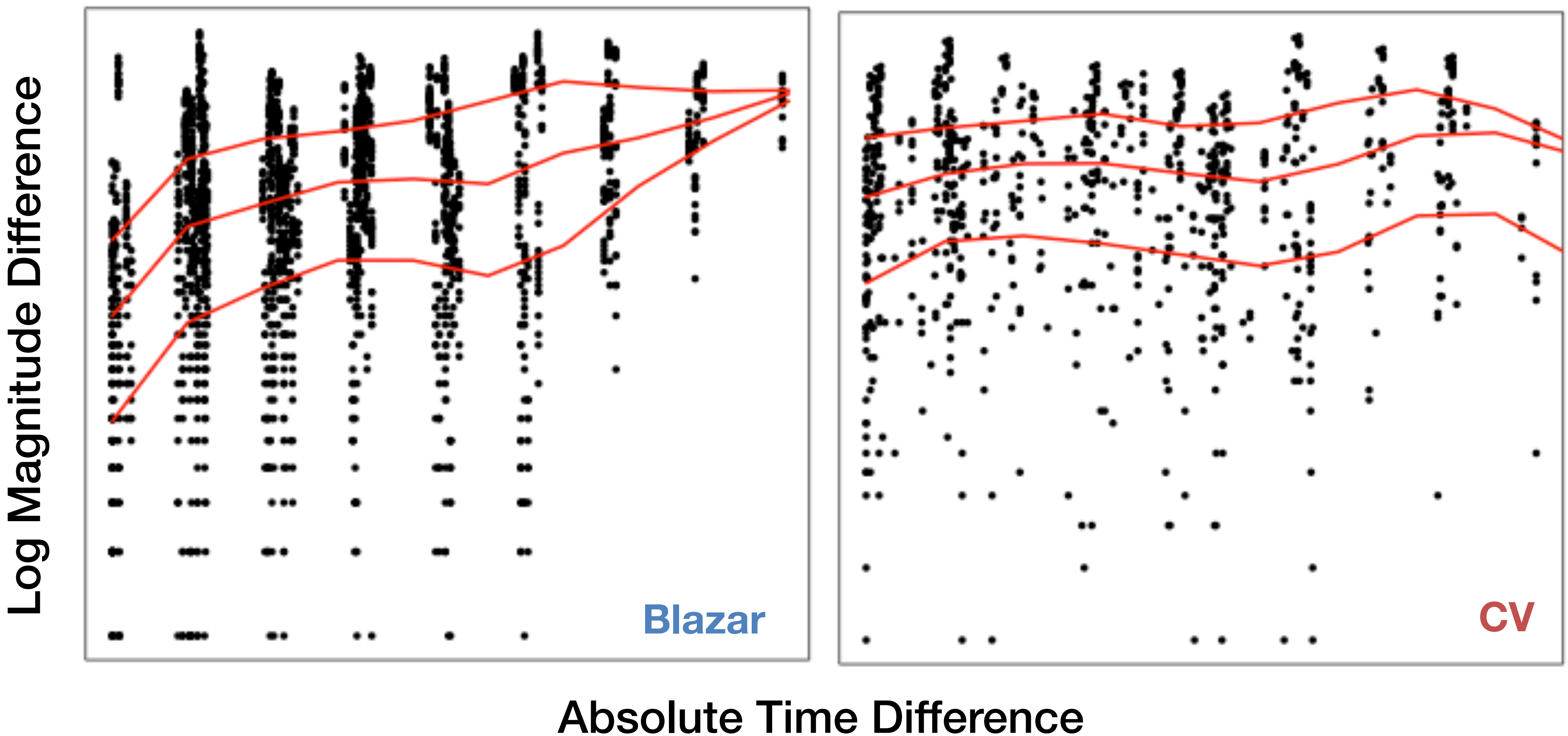
Reduces potential influence of any individual unit

Blazars versus CVs



Comparison of Structure Functions

Blazars versus CVs



Quantile regression fits

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:

Prediction \ Truth	Blazar	CV
Blazar	18	10
CV	8	91

Blazar versus CV

Performance on test set:

Deep Learning

Prediction \ Truth	Blazar	CV
Blazar	18	10
CV	8	91

Random Forest

Prediction \ Truth	Blazar	CV
Blazar	12	8
CV	14	93

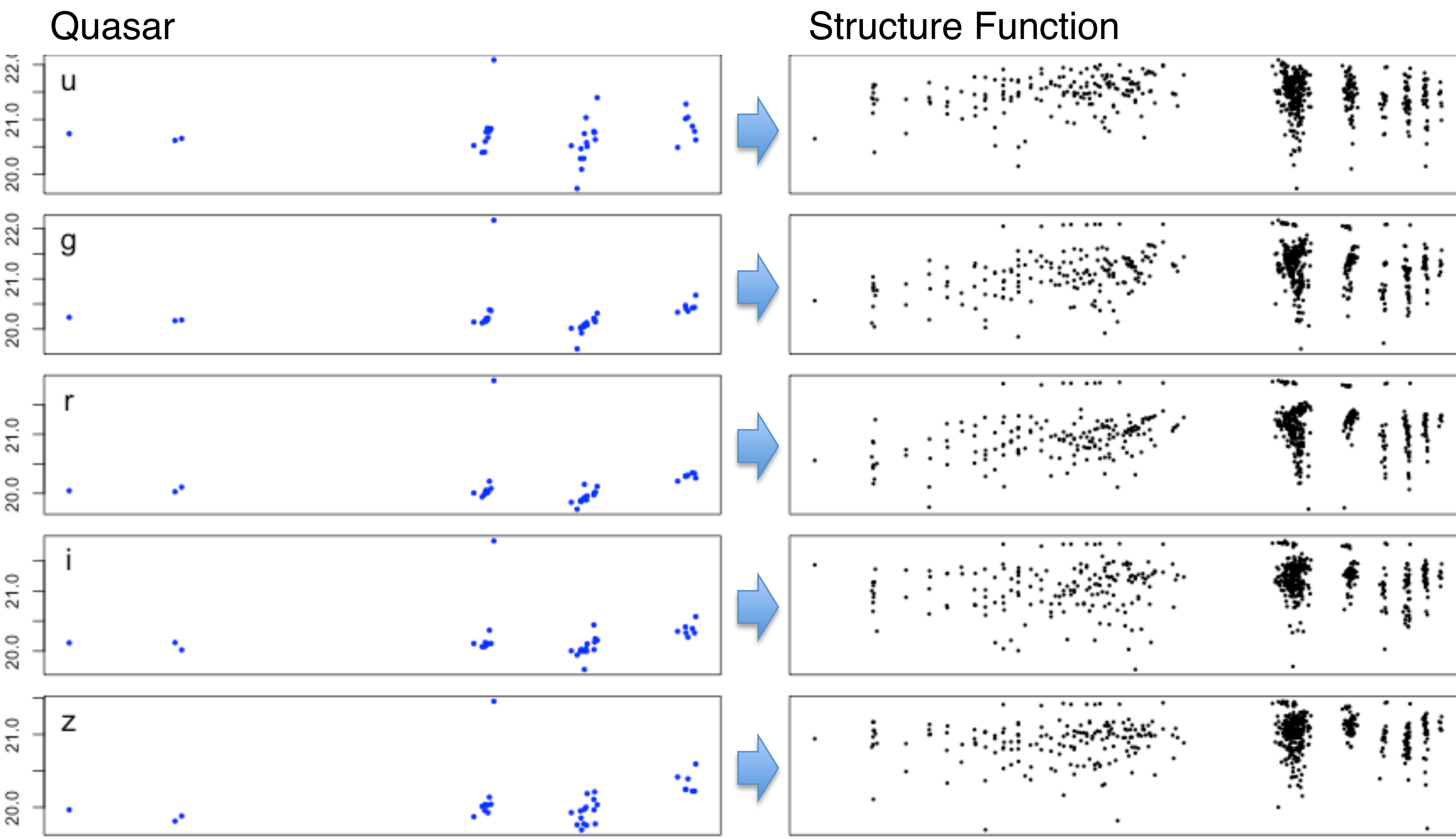
Potential of Deep Learning

Best suited to situations where high-dimensional input is required

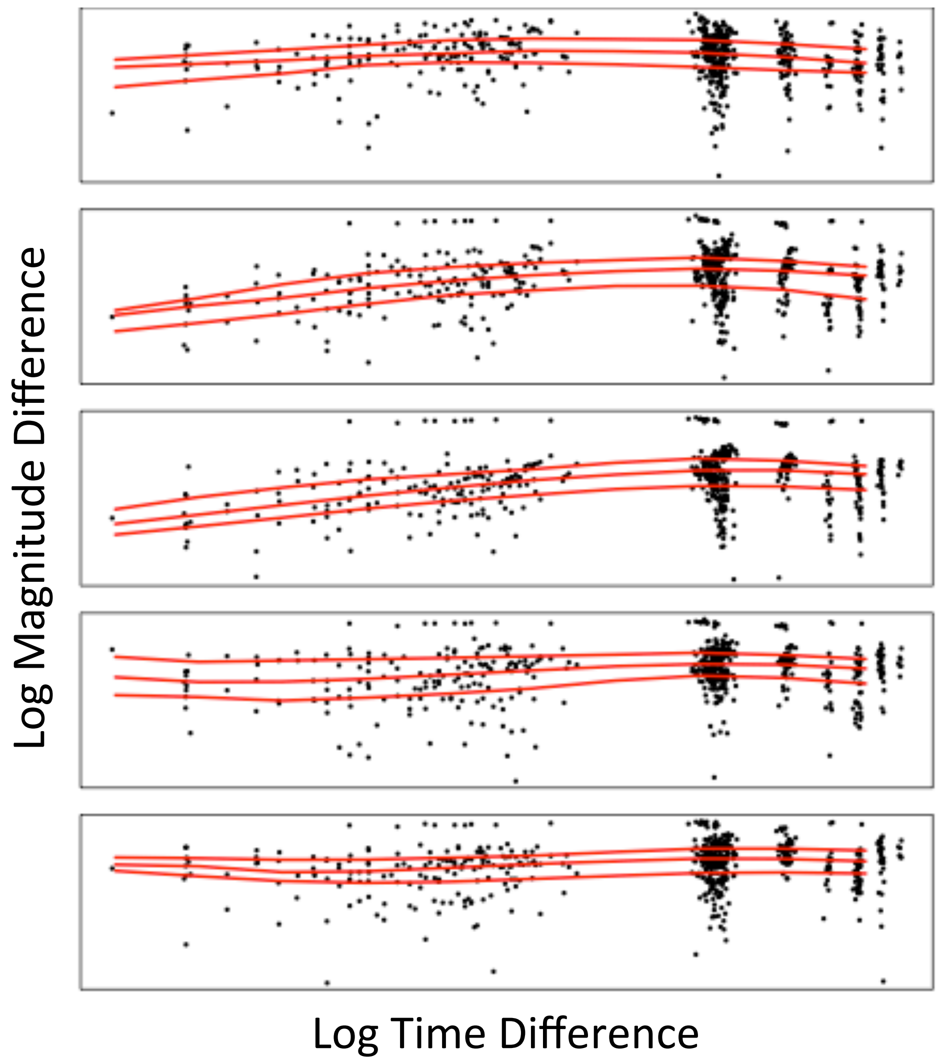
Avoid the curse of dimensionality

Seems particularly relevant for classification challenges

Quasar Classification



Quasar Classification



References

Drake, AJ, et al. ApJ (696): 870

Eyer L, and Mowlavi, N. 2008. Variable Stars across the Observational HR Diagram. Journal of Physics Conference Series

Goodfellow, Bengio, and Courville. *Deep Learning*

Hornik, et al. Neural Networks (3): 551

Ivezic, et al. arXiv 0805.2366

Kelly, et al. ApJ (698): 895

Kilbinger, et al. MNRAS (430):2200

Mahabal, et al. 2011. Discovery, classification, and scientific exploration of transient events from the Catalina Real-time Transient Survey

Montufar, et al. *NIPS 2014*

Peters, et al. ApJ (811): 95

Schlegel. arXiv 1203.0591

Schmidt et al. ApJ (714): 1194