Regularization in Deep Neural Networks

Paul Stey Isabel Restrepo

March 3, 2017

Table of Contents

What is Regularization?

Method of calibrating the complexity of our model

What is Regularization?

1. Very general (i.e., applicable to variety of models)
2. Penalized regression
   1. Ridge regression
   2. Lasso
   3. Elastic net
3. XGBoost improvement on AdaBoost

L_2 Regularization Regression

1. Ridge regression, also called Tikhonov regularization (Tikhonov, 1963)
2. Penalize the L_2 norm
3. Constrains the Euclidian distance of β

Given the linear model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i$$

the ridge penalty constrains

$$\|\beta\|_2 = \sqrt{\beta_1^2 + \beta_2^2 + ... + \beta_p^2}$$

L_2 Regularization cont.

Recall the least squares solution is obtained by

$$\widehat{\beta} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T y.$$

The ridge penalty gives us

$$\widehat{\beta}^{ridge} = \left(\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}^T y$$

where λ is a penalty term, and $\boldsymbol{I}$ is the $p \times p$ identity matrix.

L_2 Regularization cont.

1. Penalty $\lambda \in [0, \infty)$
2. Shrinks β_j values towards 0 (and each other)
3. Benefits:
   1. Reduce model variance (i.e., more generalizable)
   2. Computationally efficient
   3. Gain stability in estimates involving correlated data

[example in R]

4. Drawbacks:
   1. Sacrifice unbiasedness of maximum likelihood estimate

L_1 Regularization

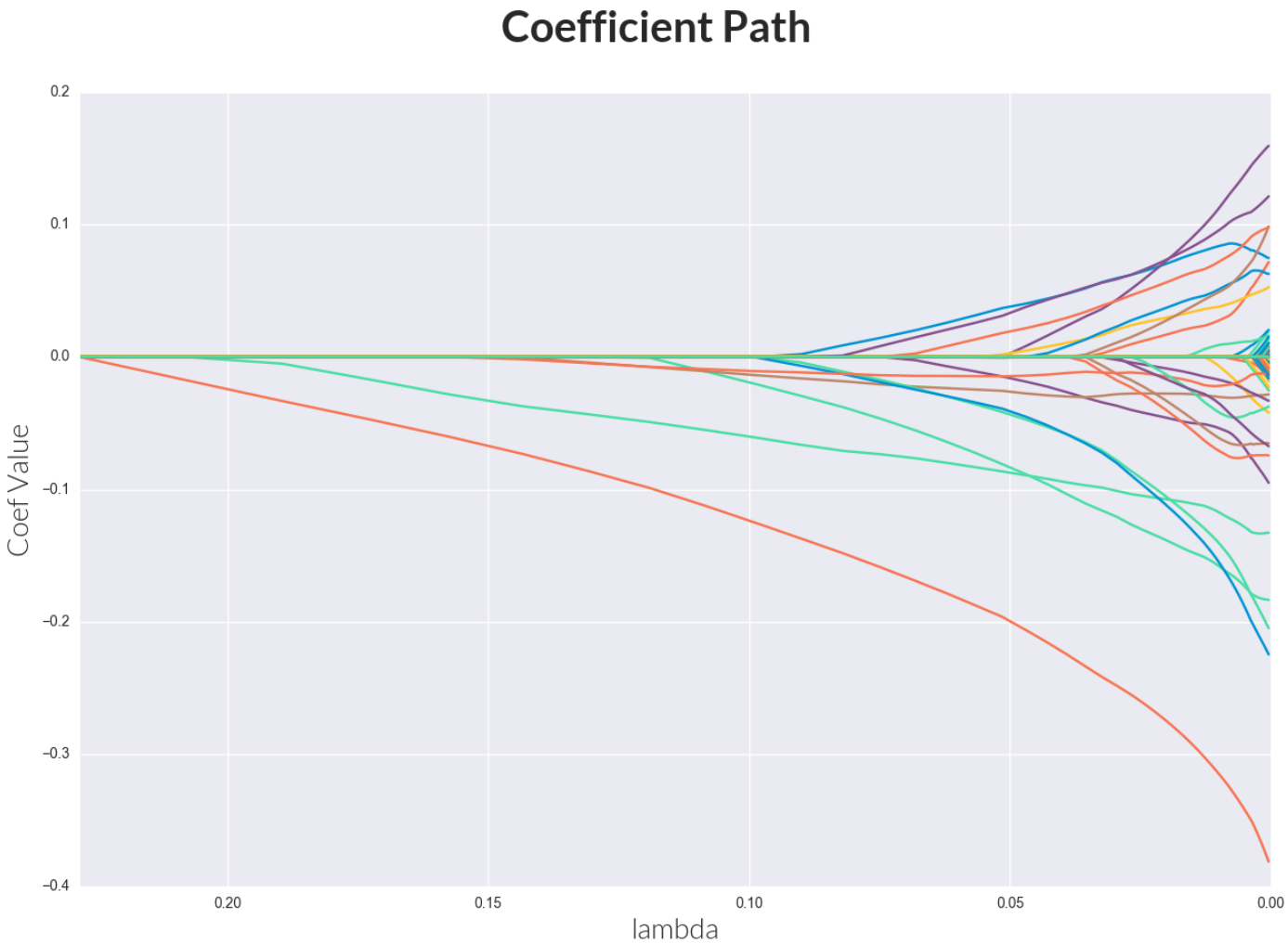
The lasso

1. Similar to ridge regression
2. Penalize L_1 norm of β
3. Constrains Manhattan distance spanned by vector of regression coefficients

$$\|\beta\|_1 = |\beta_1| + |\beta_2| + ... + |\beta_p|$$

4. As λ increases, $\beta_j \to 0$
   1. β_j for less important predictors shrink faster
   2. Can be used for "variable selection"

Drop Out cont.



Elastic Net

1. Penalize *both* the L_1 and L_2 norms of β
2. Advantage of combining strengths of both approaches

Caveat: Note that we often use ridge or lasso for very specific reason (i.e., ridge for accuracy, lasso for variable selection). One might suggest we give away this clarity of purpose when using the elastic net.

Terminology

Many papers and texts use the term "weight decay" to refer norm penalization methods in neural networks

L_2 Norm Penalty

1. Same basic idea as ridge regression
2. Cost function gets penalty term λ to constrain weights

$$C = C_0 + \frac{\lambda}{2} \sum_i w_i^2$$

where C_0 is the original cost.

3. Stabilizes weights for highly correlated features

L_1 Norm Penalty

1. Same idea as the lasso
2. Add penalty to cost function, thereby shrinking weights

$$C = C_0 + \lambda \sum_i |w_i|$$

3. Some weights will be shrunk to 0
4. Produces a more sparse model

Regularization More Broadly

Recall that regularization is merely a method of controlling model complexity.

The standard shrinkage method examples make this very transparent with the use of explicit penalty terms.

Broader View cont.

Consider less obvious examples:

1. Number of boosting iterations
2. Learning rate for gradient descent
3. Bagging proportion
4. Proportion in training *vs.* test set

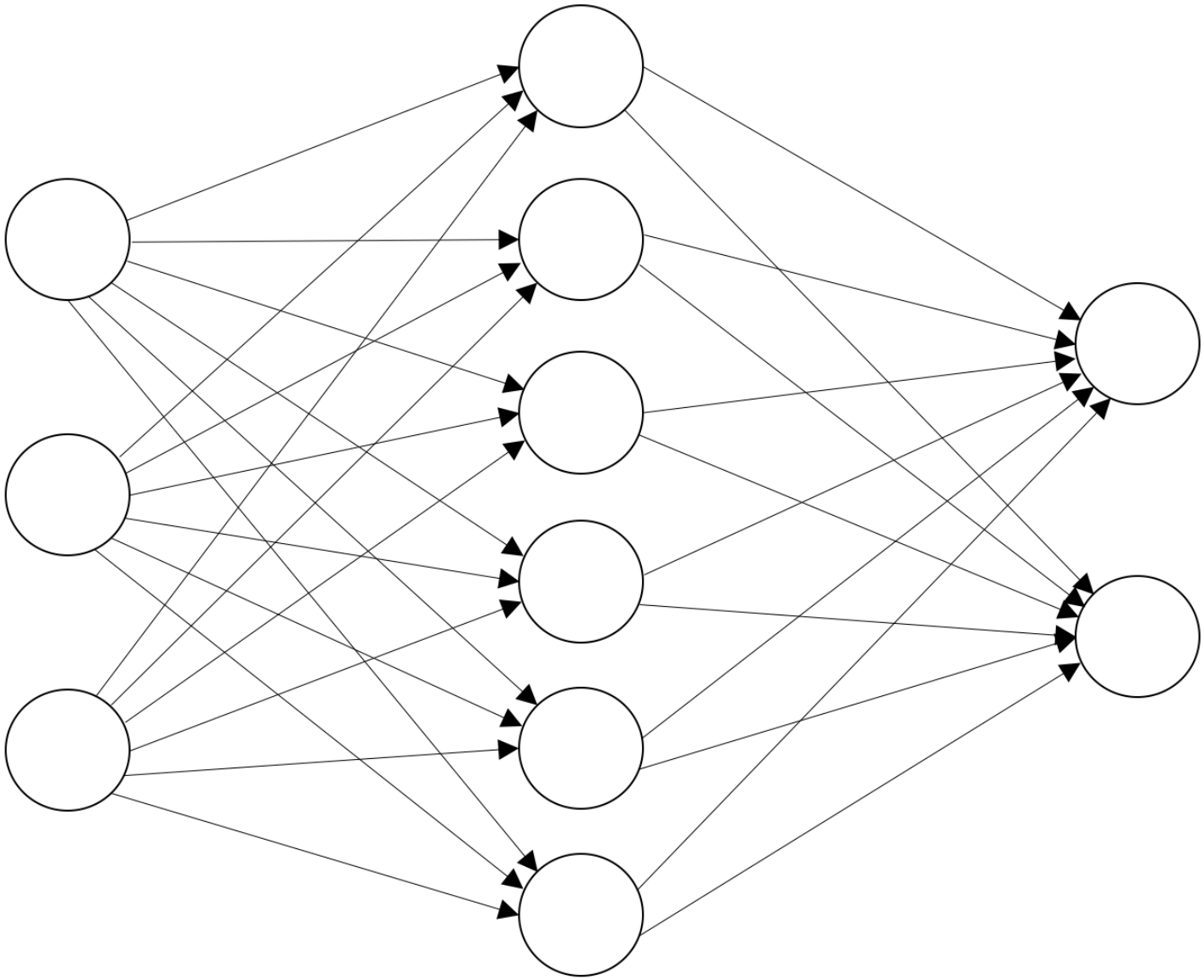
Broader View cont.

The essential feature of regularization is that it involves using some mechanism to control the complexity of our model. This makes our model more generalizable.

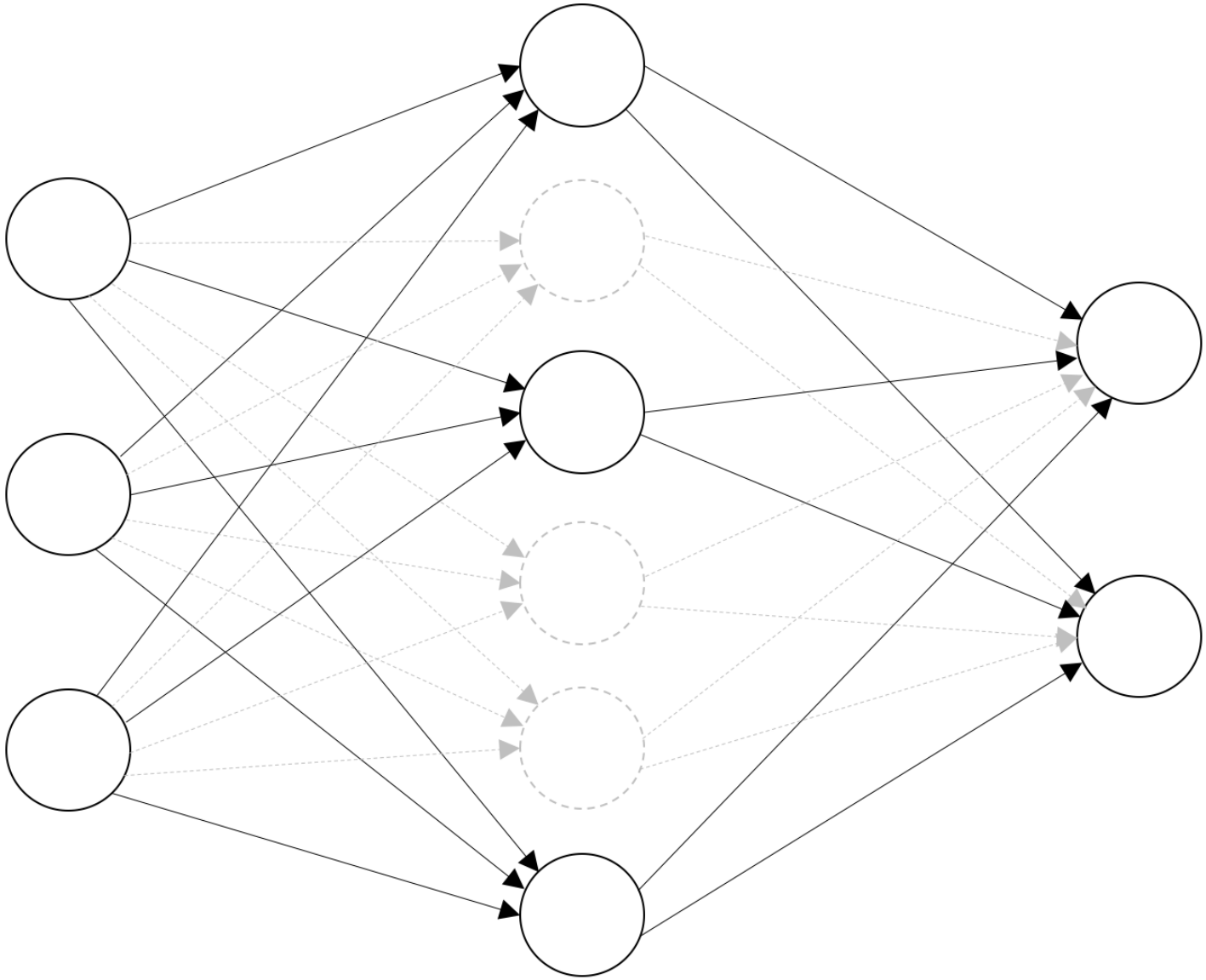
Drop Out

1. Hinton *et al.* 2012
2. Subset of hidden and input units are randomly omitted
3. Reduces number of weights and biases being estimated
4. Thereby reducing complexity of model

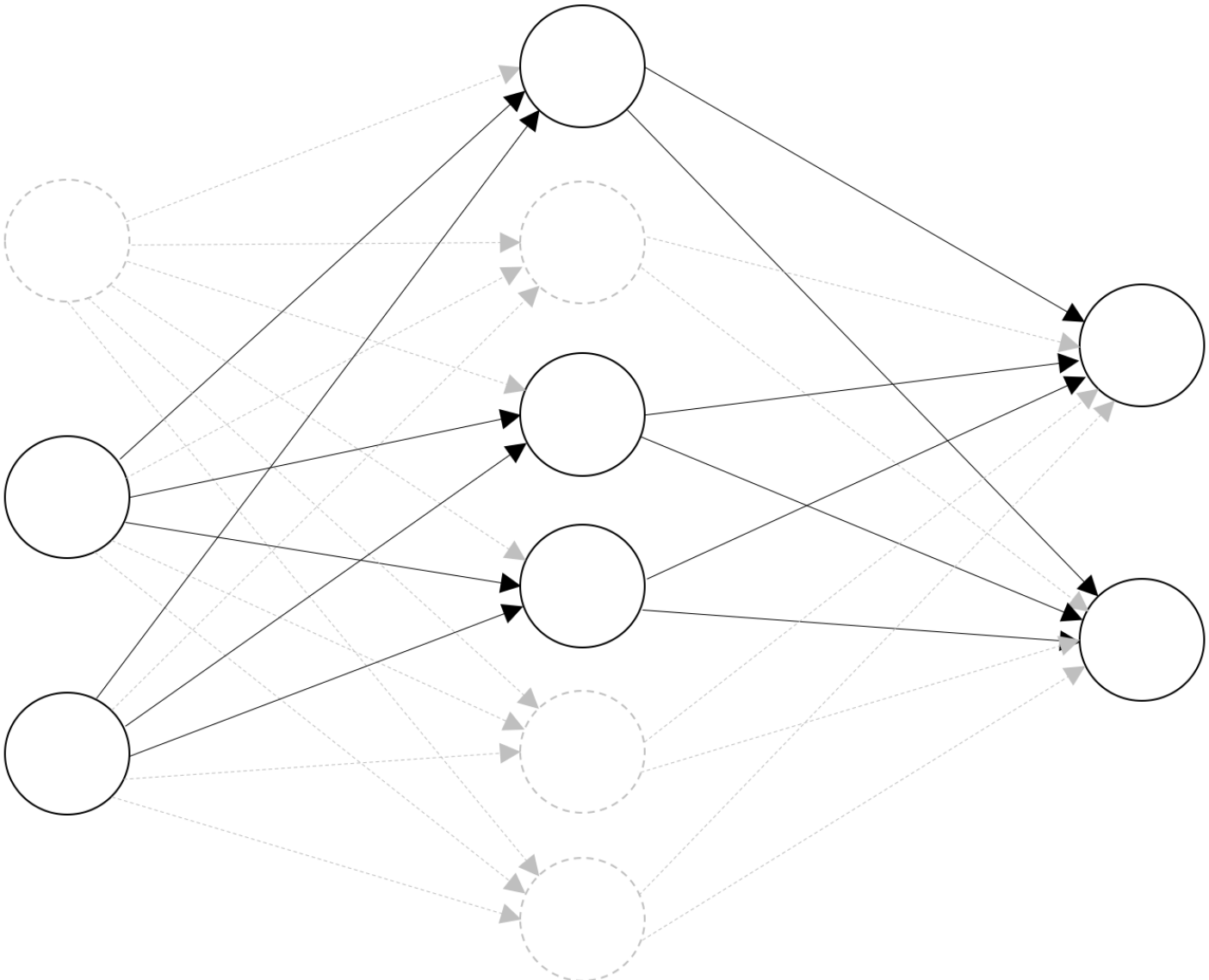
Drop Out cont.



Drop Out cont.



Drop Out cont.



Drop Out in Practice

1. Units omitted according to meta-parameters we specify
   1. Input unit probability of inclusion frequently 0.8
   2. Hidden unit 0.5
2. In most cases, omission means multiplying activation by 0

Compare with Bagging and Random Forests

Consider similarities with bagging and feature subset selection in random forests.

These methods have a regularizing effect on model by using randomization to artificially restrict the parameters.

Data Augmentation

1. Adding [synthetic] data
2. Or perturbing existing data
3. Or both!

Data Augmentation Examples

1. Translating images by a few pixels
2. Injecting noise in speech recognition data

Adversarial Training

1. Szegedy *et al.* 2014
2. Similar in a sense to data augmentation
3. Generate new observations in such a way as to maximize the chance they are misclassified
4. These in-domain examples that have been altered are referred to adversarial examples
5. Related to "rubbish class" examples (LeCun *et al.*, 1998)

Adversarial Training Example

$+ .007 \times$

$=$

$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon \, \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
$y =$"panda"	"nematode"	"gibbon"
w/ 57.7%	w/ 8.2%	w/ 99.3 %
confidence	confidence	confidence

Goodfellow, Bengio, & Courville, 2016

Methods we Did Not Discuss

1. Early stopping
2. Bagging (Breiman, 1994)
3. Parameter sharing (Laserre *et al.*, 2006)

TensorFlow examples