Building and Training a Deep Learning Network

Deep Learning — Unit 3

Dr. Jon Krohn
jon@untapt.com

Slides available at jonkrohn.com/talks

July 28th, 2018

Outline

Outline

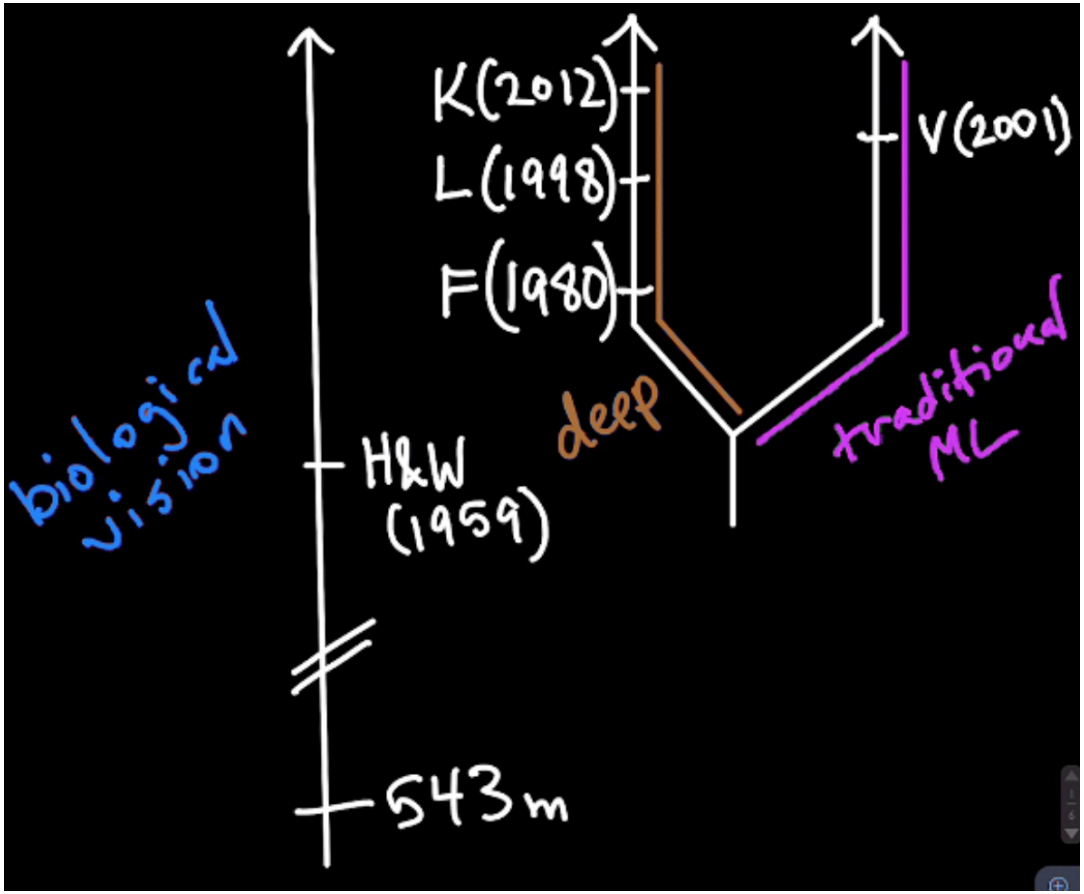
Outline

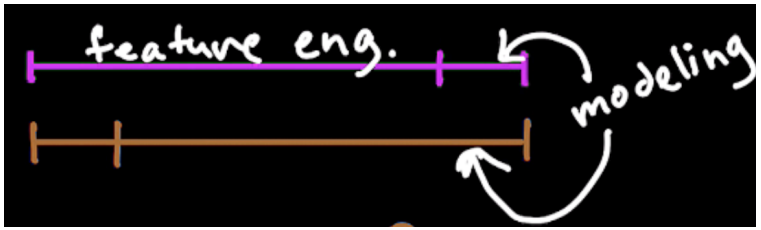
Outline

Outline

Outline

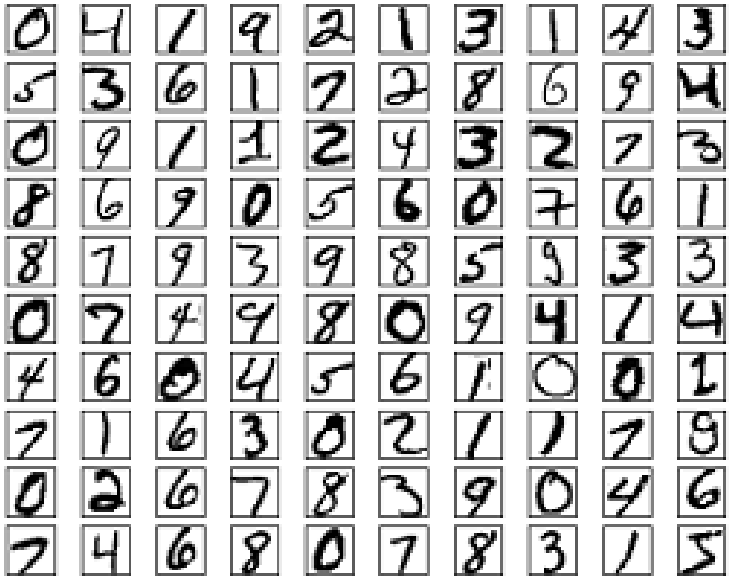
Outline


K(2012)
L(1998)
F(1980)
V(2001)
biological vision
H&W (1959)
deep
traditional ML
543m


feature eng.
modeling

MNIST Digits & LeNet-5

LeCun, Boutou, Bengio & Haffner (1998)



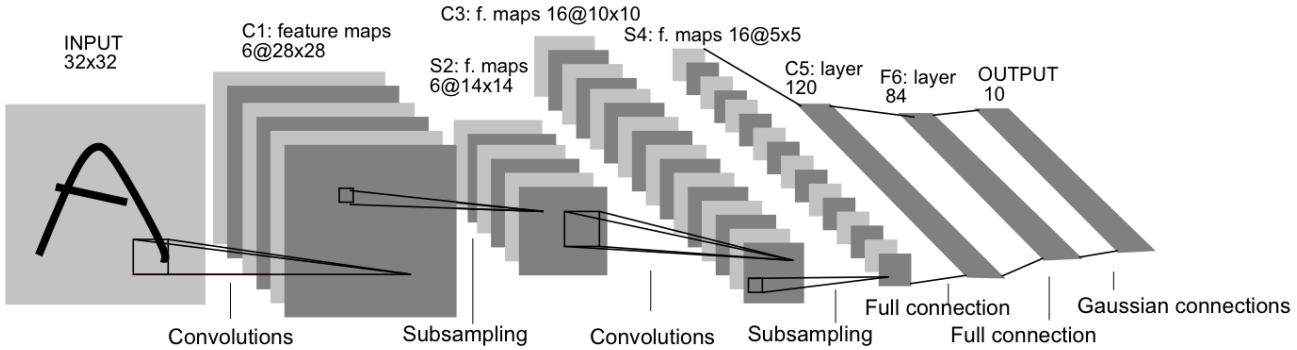
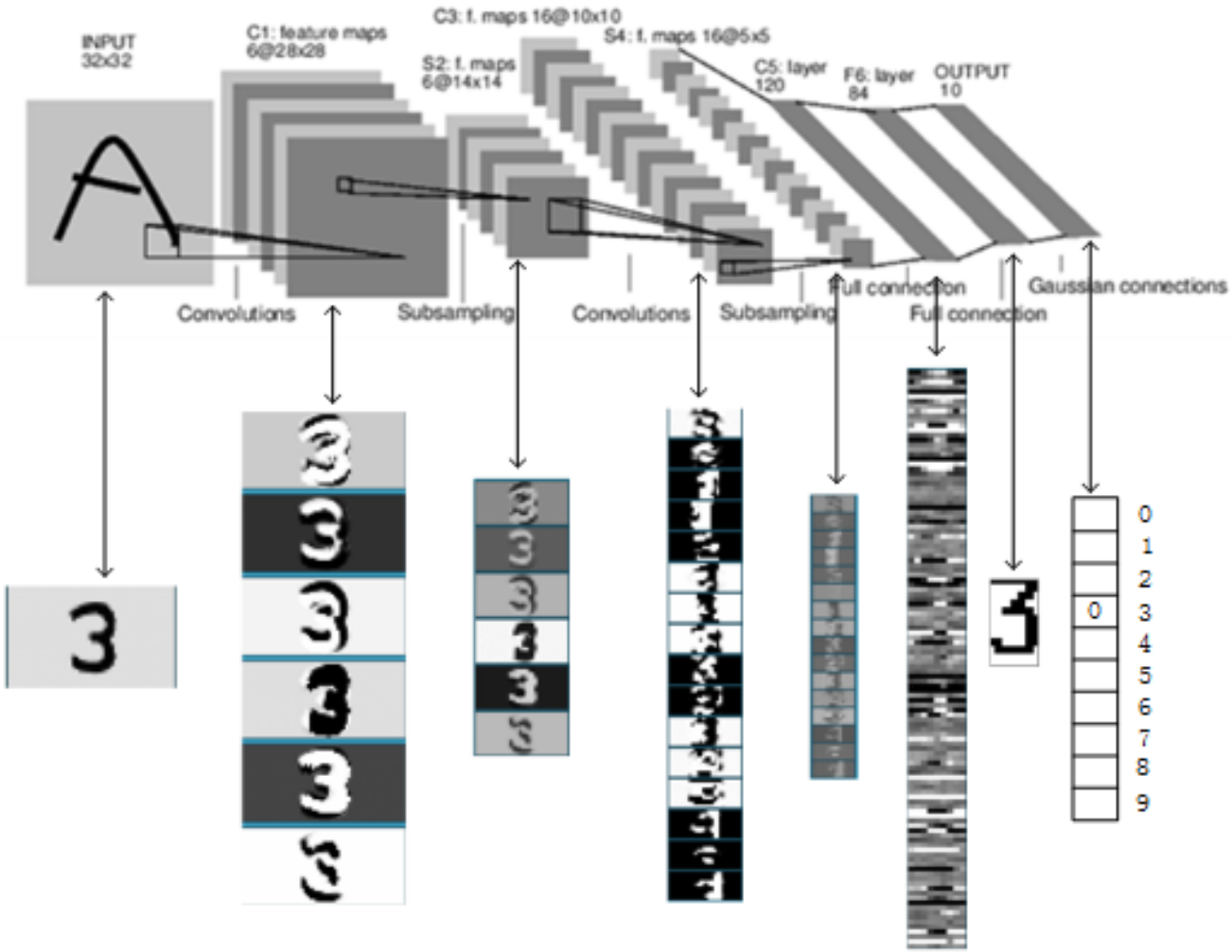


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

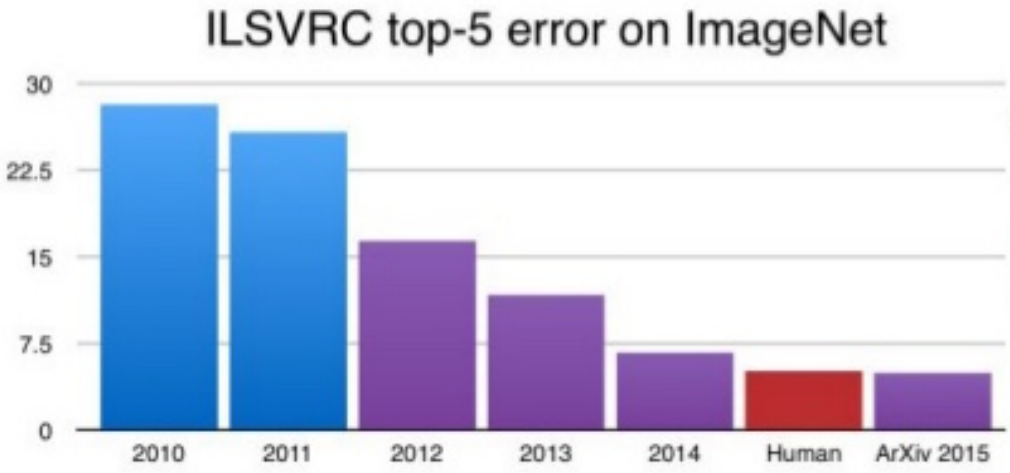
LeNet-5

LeCun, Boutou, Bengio & Haffner (1998)

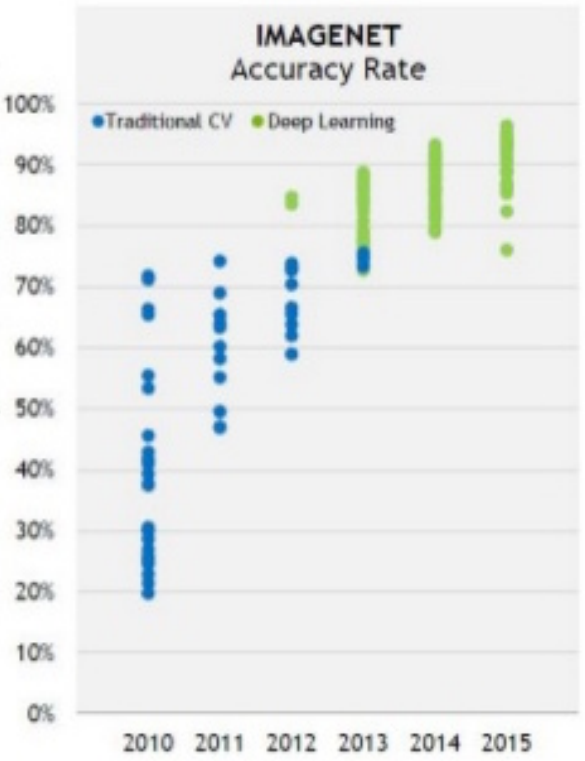


ImageNet Classification Error

ILSVRC: 1.4m, 1k object classes

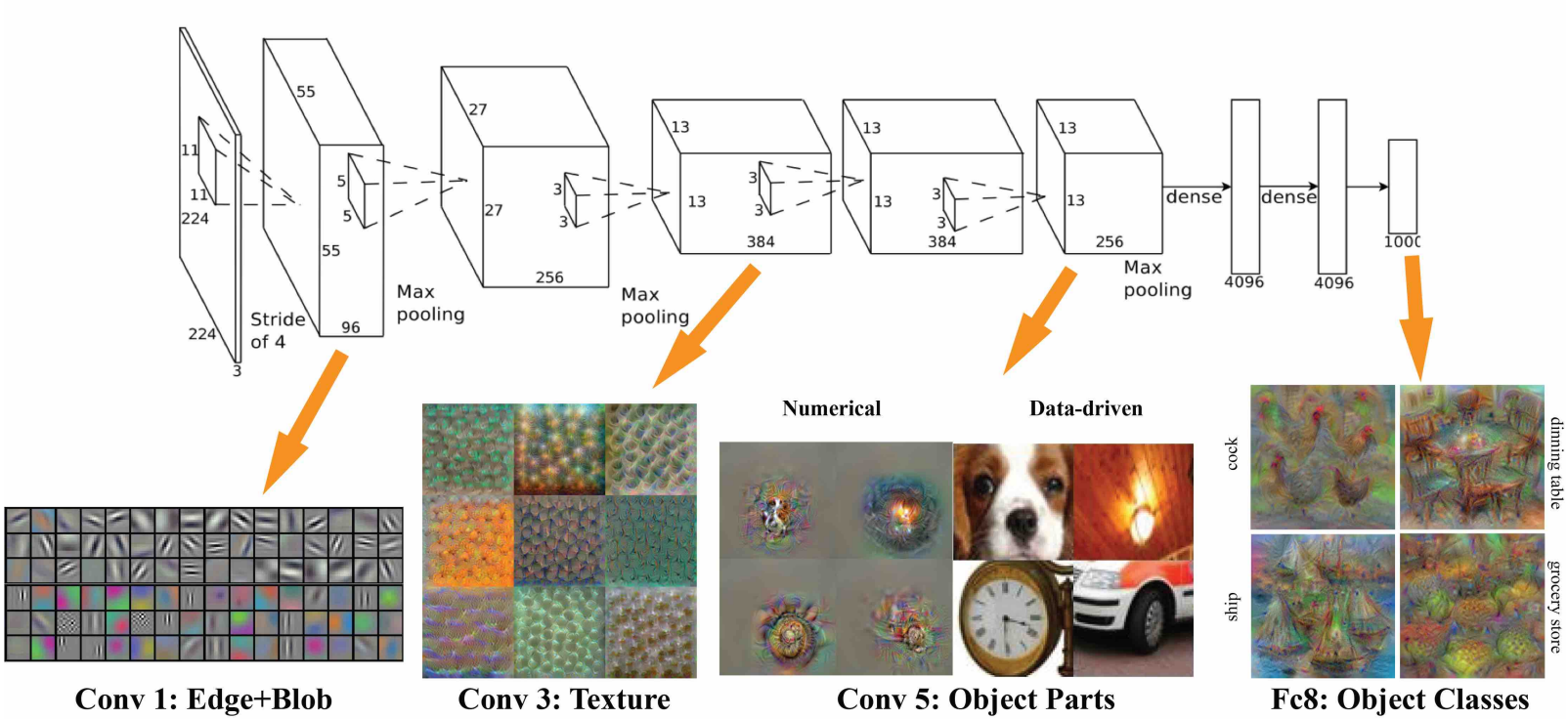


- Blue: Traditional CV
- Purple: Deep Learning
- Red: Human

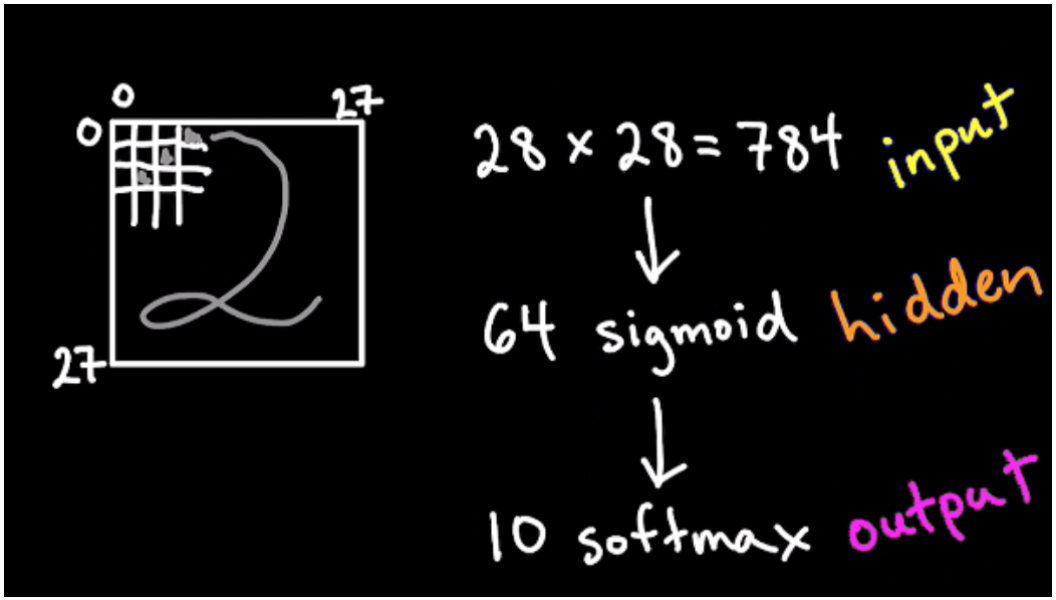


AlexNet

Krizhevsky, Sutskever & Hinton (2012)



A Shallow Neural Network



[shallow notebook]

Outline

TensorFlow Playground

Interactive ANN Visualization

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

TensorFlow Playground

Interactive ANN Visualization

- sigmoid neuron
- tanh neuron
- ReLU

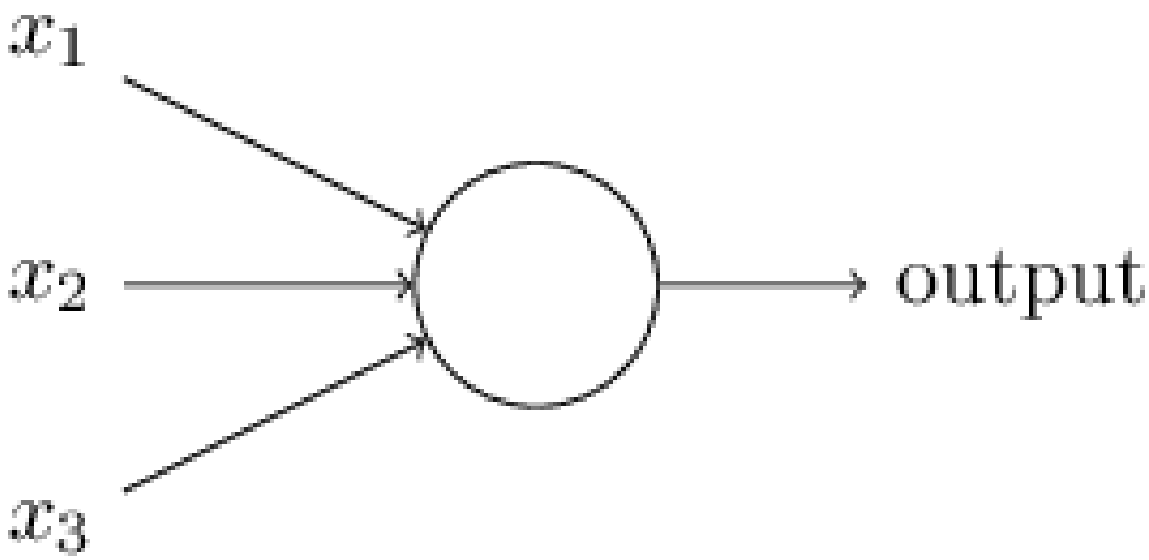
- weight
- bias
- cost function
- gradient descent

- input layer
- hidden layer
- dense/FC layer
- softmax layer
- output layer

Outline

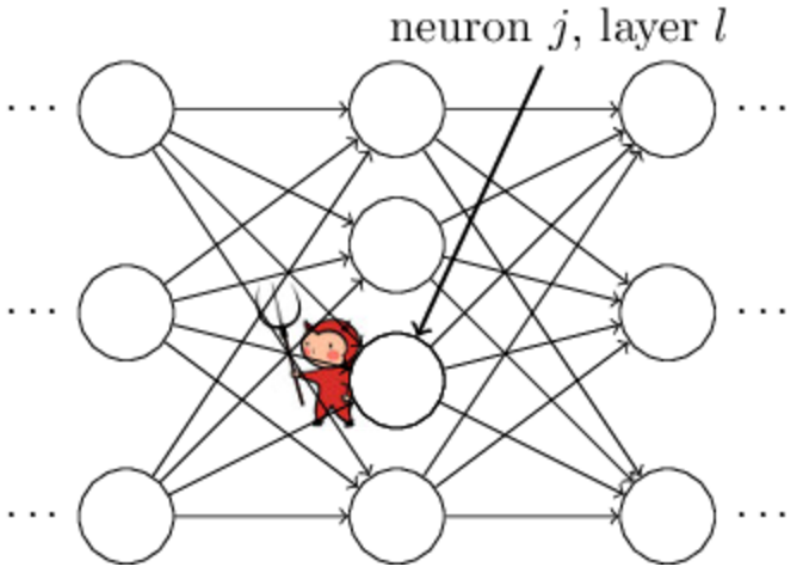
Essential Theory I

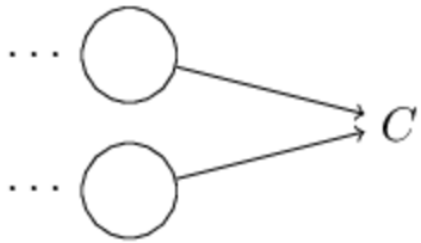
Neural Units



Essential Theory II

Cost Functions, Gradient Descent, and Backpropagation

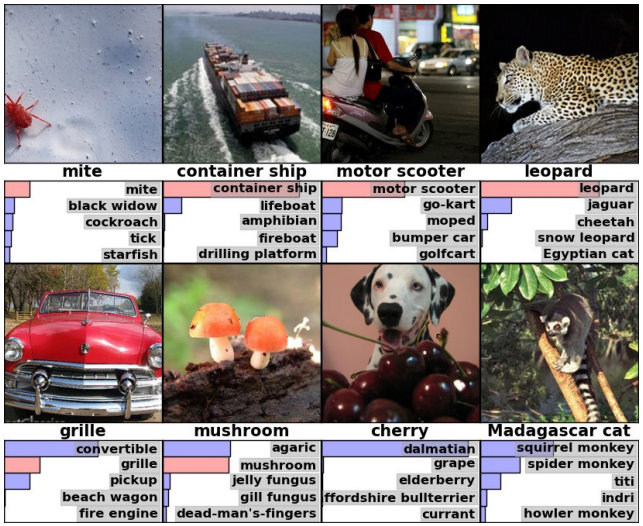


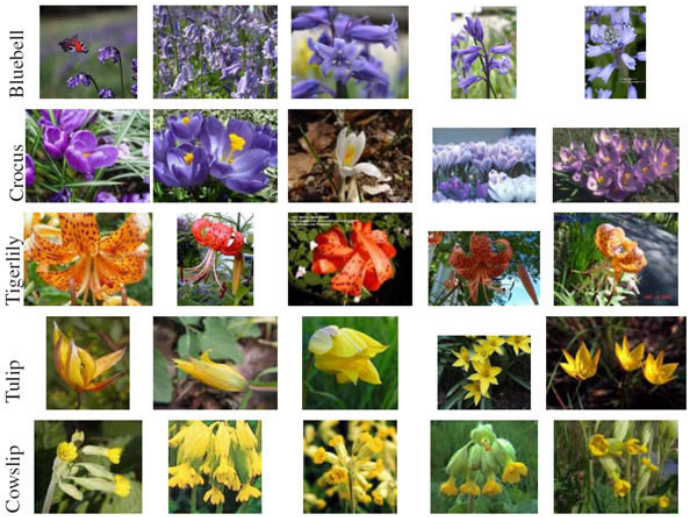


An Intermediate Neural Network

[intermediate notebook]

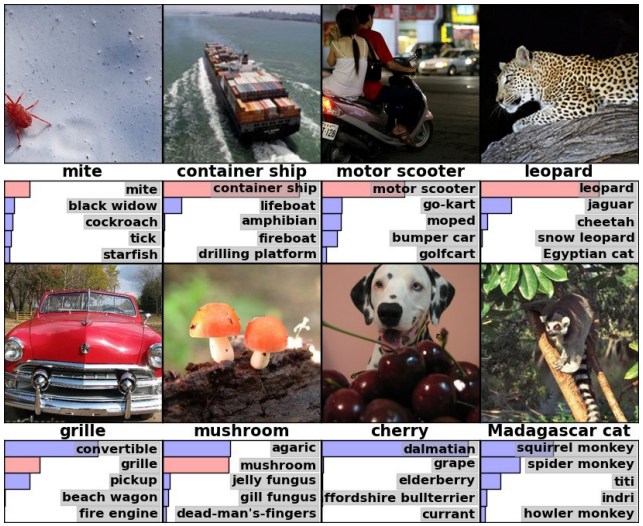
Data Sets for Deep Learning

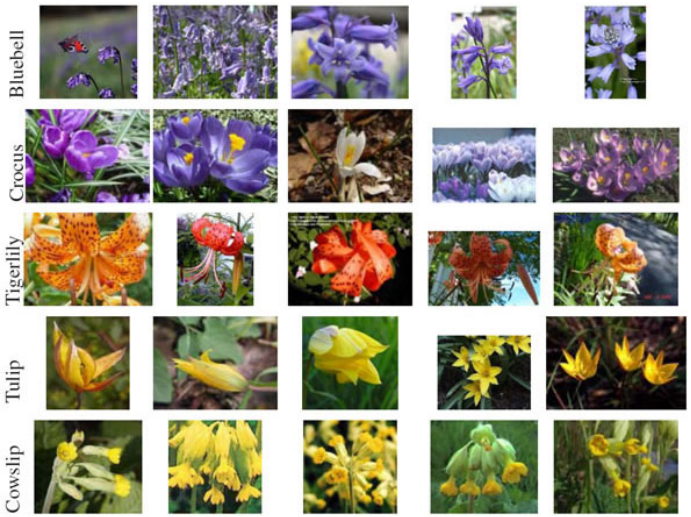




Dataset	Classes	Train Samples
AG's News	4	120,000
Sogou News	5	450,000
DBPedia	14	560,000
Yelp Review Polarity	2	560,000
Yelp Review Full	5	650,000
Yahoo! Answers	10	1,400,000
Amazon Review Full	5	3,000,000
Amazon Review Polarity	2	3,600,000

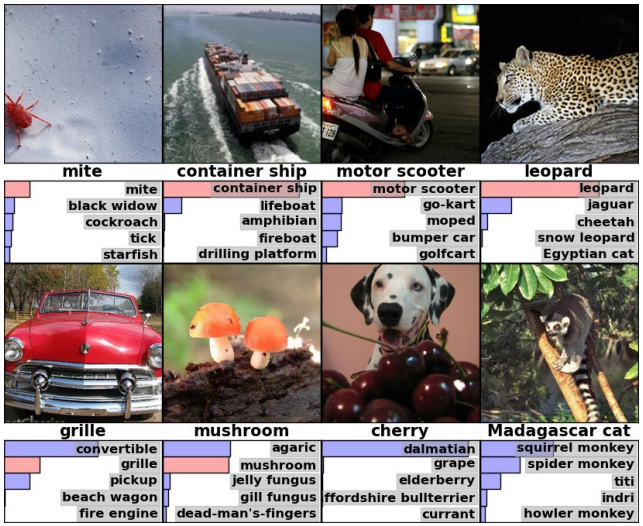
Data Sets for Deep Learning

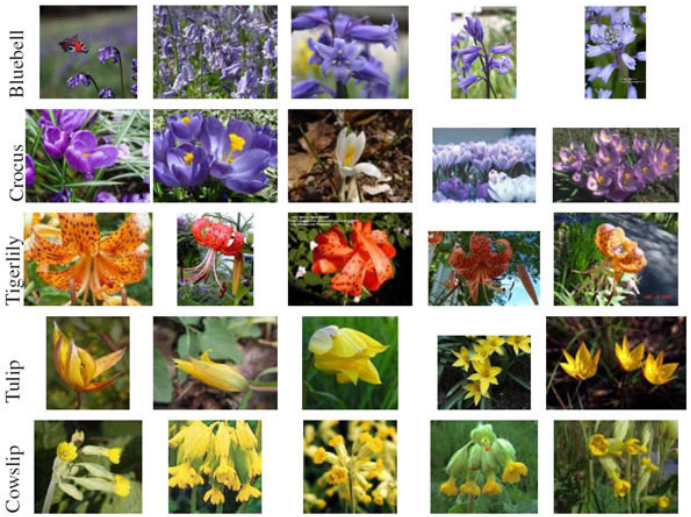




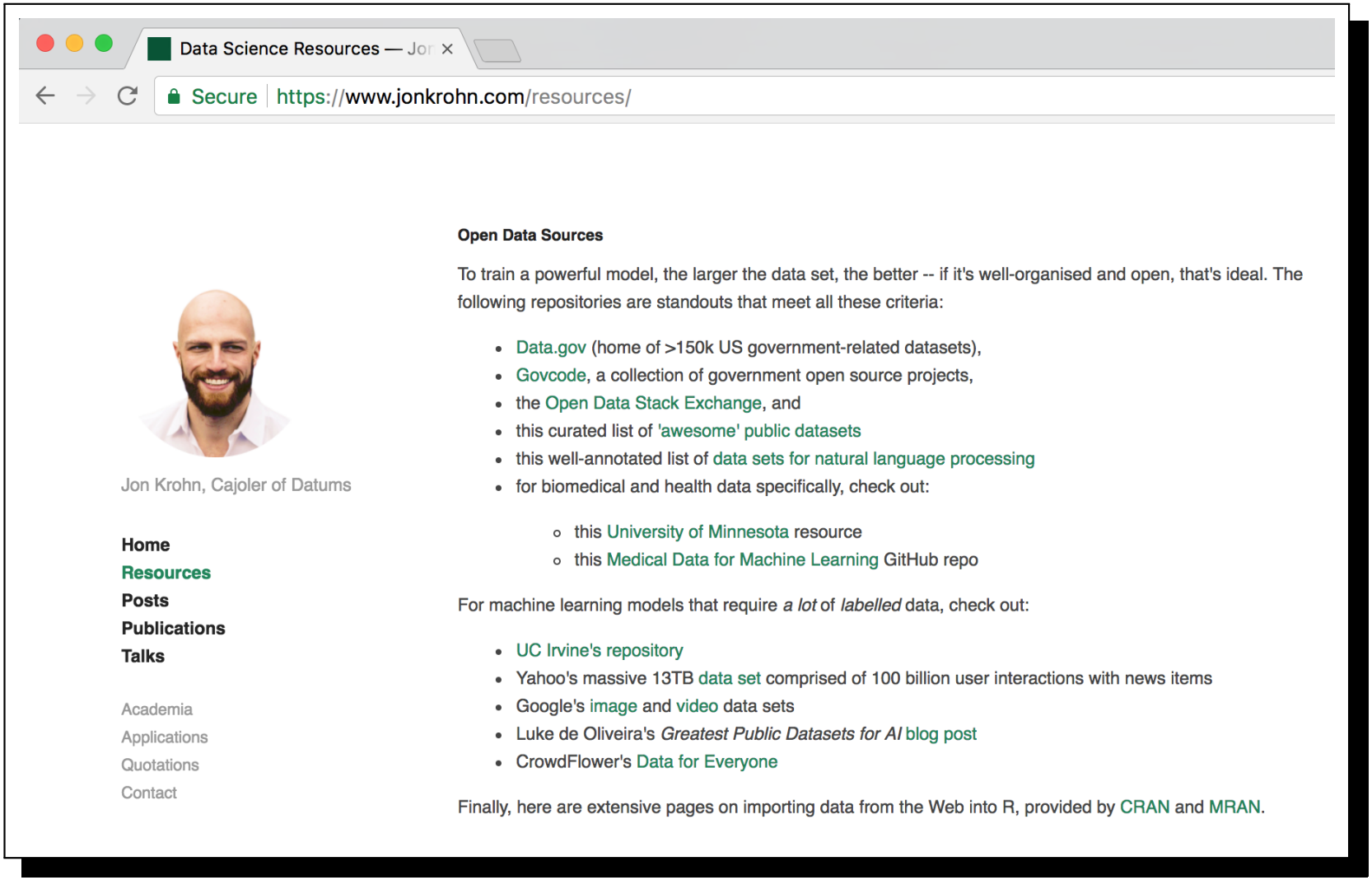
Dataset	Classes	Train Samples
AG's News	4	120,000
Sogou News	5	450,000
DBPedia	14	560,000
Yelp Review Polarity	2	560,000
Yelp Review Full	5	650,000
Yahoo! Answers	10	1,400,000
Amazon Review Full	5	3,000,000
Amazon Review Polarity	2	3,600,000

Data Sets for Deep Learning





Dataset	Classes	Train Samples
AG's News	4	120,000
Sogou News	5	450,000
DBPedia	14	560,000
Yelp Review Polarity	2	560,000
Yelp Review Full	5	650,000
Yahoo! Answers	10	1,400,000
Amazon Review Full	5	3,000,000
Amazon Review Polarity	2	3,600,000


Data Science Resources — Jon
Secure https://www.jonkrohn.com/resources/
Jon Krohn, Cajoler of Datums
Home
Resources
Posts
Publications
Talks
Academia
Applications
Quotations
Contact
Open Data Sources
To train a powerful model, the larger the data set, the better -- if it's well-organised and open, that's ideal. The following repositories are standouts that meet all these criteria:
Data.gov (home of >150k US government-related datasets),
Govcode, a collection of government open source projects,
the Open Data Stack Exchange, and
this curated list of 'awesome' public datasets
this well-annotated list of data sets for natural language processing
for biomedical and health data specifically, check out:
this University of Minnesota resource
this Medical Data for Machine Learning GitHub repo
For machine learning models that require a lot of labelled data, check out:
UC Irvine's repository
Yahoo's massive 13TB data set comprised of 100 billion user interactions with news items
Google's image and video data sets
Luke de Oliveira's Greatest Public Datasets for AI blog post
CrowdFlower's Data for Everyone
Finally, here are extensive pages on importing data from the Web into R, provided by CRAN and MRAN.

Outline

Your Deep Learning Project I

Ideating

Your Deep Learning Project I

Ideating

Perspectives to approach ideating from:

- Identify a data set $\Rightarrow$ use it to solve a problem
- Identify a problem that you'd like to solve $\Rightarrow$ find an appropriate data set

Your Deep Learning Project I

Ideating

Perspectives to approach ideating from:

- Identify a data set $\Rightarrow$ use it to solve a problem
- Identify a problem that you'd like to solve $\Rightarrow$ find an appropriate data set

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Recommended starting points:

1. a machine-vision architecture to classify images, e.g.:
   - [Fashion MNIST]
   - one of the dozens of data sets with the keyword *image* in the title from [CrowdFlower]
   - one of the *Computer Vision* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
2. an NLP architecture to classify text, e.g.:
   - the Yelp or Amazon sentiment [data sets] detailed in Section 4 of [Xiang Zhang et al.'s paper]
   - the Yahoo! Answers categories data set also detailed in Xiang Zhang et al.'s paper
   - one of the dozens of data sets with the keywords *sentiment* or *text* in the title from [CrowdFlower]
   - one of the *Natural Language* data sets (other than the MNIST data set) in Luke de Oliveira's [blog post]
3. a regression model

Outline

Weight Initialization

- uniform
- normal
- Xavier Glorot

[Jupyter demo]

Weight Initialization

- uniform
- normal
- Xavier Glorot

[Jupyter demo]

Weight Initialization

- uniform
- normal
- Xavier Glorot

[Jupyter demo]

Stochastic Gradient Descent

- learning rate
- batch size
- second-order gradient learning
  - momentum
  - Adam

Stochastic Gradient Descent

- learning rate
- batch size
- second-order gradient learning
  - momentum
  - Adam

Stochastic Gradient Descent

- learning rate
- batch size
- second-order gradient learning
  - momentum
  - Adam

Stochastic Gradient Descent

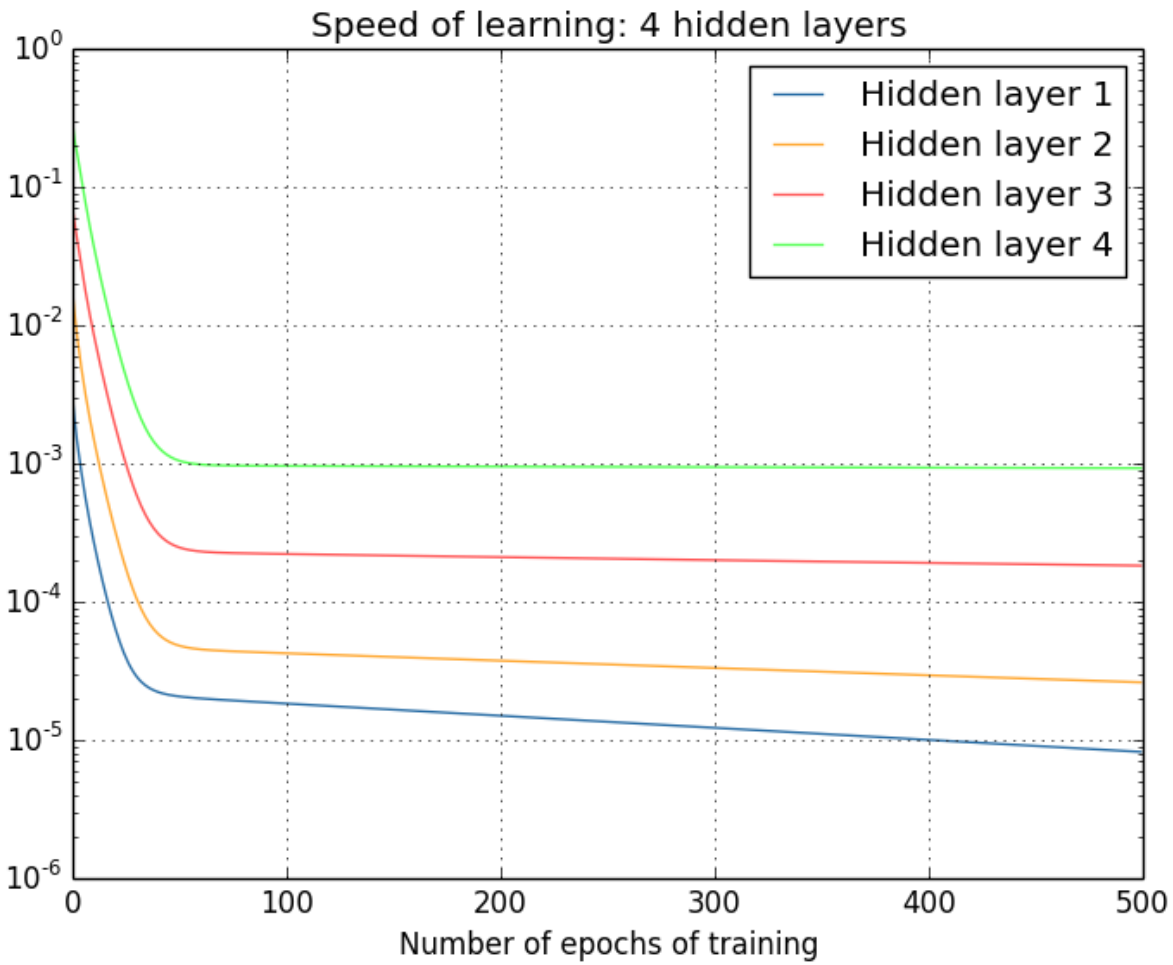
- learning rate
- batch size
- second-order gradient learning
  - momentum
  - Adam

Stochastic Gradient Descent

- learning rate
- batch size
- second-order gradient learning
  - momentum
  - Adam

Outline

Unstable Gradients



Unstable Gradients

- vanishing
- exploding

Unstable Gradients

- vanishing
- exploding

Avoiding Overfitting

(Or, Model Generalization)

- L1/L2 regularization
- dropout
- artificial data set expansion

Avoiding Overfitting

(Or, Model Generalization)

- L1/L2 regularization
- dropout
- artificial data set expansion

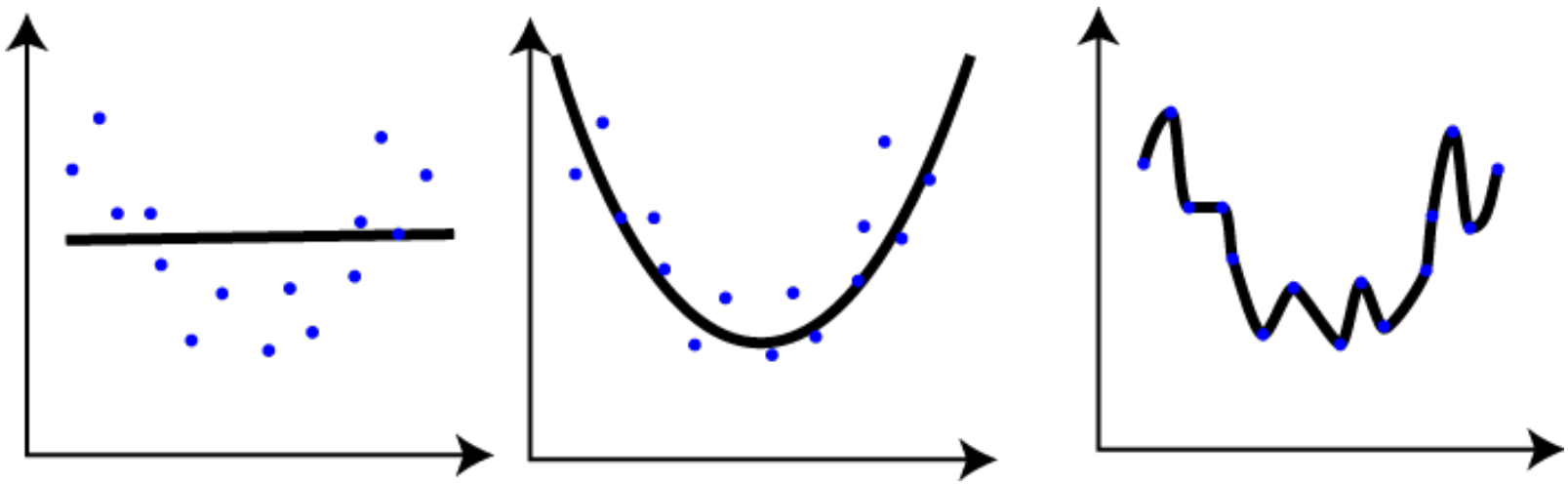
Avoiding Overfitting

(Or, Model Generalization)

- L1/L2 regularization
- dropout
- artificial data set expansion

Avoiding Overfitting

(Or, Model Generalization)



Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Last, But Not Least

- more layers
  - max-pooling
  - flatten
- *batch normalization* avoids *covariate shift*; advantages:
  1. initialization parameters
  2. avoid neuron saturation
  3. regularizing effect

Outline

Let's make [intermediate net] *deep*!

Outline

TensorBoard

1. add callback as in [Deep Net in Keras Jupyter notebook]
2. use Terminal to navigate to your `logs` directory
3. run `tensorboard --logdir=. --port 6006`
4. navigate to `http://localhost:6006/` in a web browser

TensorBoard

1. add callback as in [Deep Net in Keras Jupyter notebook]
2. use Terminal to navigate to your `logs` directory
3. run `tensorboard --logdir=. --port 6006`
4. navigate to `http://localhost:6006/` in a web browser

TensorBoard

1. add callback as in [Deep Net in Keras Jupyter notebook]
2. use Terminal to navigate to your `logs` directory
3. run `tensorboard --logdir=.  --port 6006`
4. navigate to `http://localhost:6006/` in a web browser

TensorBoard

1. add callback as in [Deep Net in Keras Jupyter notebook]
2. use Terminal to navigate to your `logs` directory
3. run `tensorboard --logdir=. --port 6006`
4. navigate to `http://localhost:6006/` in a web browser

TensorBoard

The Interpretation of Model Outputs