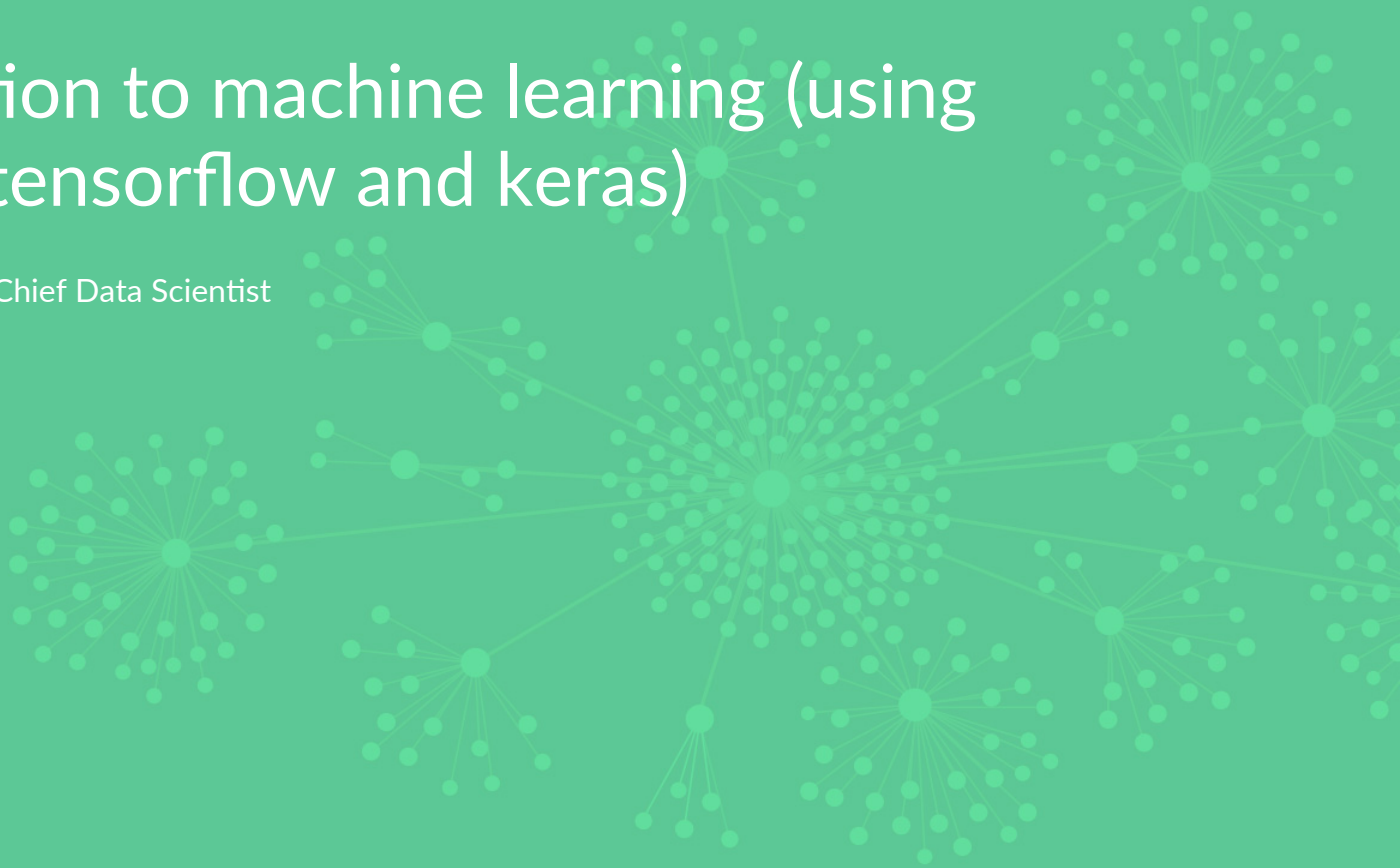
CrowdFlower

An introduction to machine learning (using scikit-learn, tensorflow and keras)

Lukas Biewald, Founder and Chief Data Scientist



Prerequisites for following along

1. Slides available at http://lukas.show
2. Install scikit-learn (and pandas and numpy and keras and tensorflow)

How to do it:

- `git clone https://github.com/lukas/ml-class`
- `cd ml-class`
- `pip install scikit-learn`
- `pip install pandas`
- `pip install tensorflow`
- `pip install keras`

Try this code to test:

```
python test-scikit.py
python test-keras.py
```

If it runs without error – you're ok!

Introductions

Who am I?

Lukas Biewald, Founder, CrowdFlower

Nick Gaylord, Data Scientist, CrowdFlower

Goals

1) Concrete Useful Understanding of Machine Learning

2) Feel What It’s Really Like to Do Machine Learning

Agenda

9:00 – 10:00 Machine Learning Theory

10:00 – 12:00 Scikit-learn on a text model

12:00 – 12:30 Overview of Machine Learning Platforms

12:30 – 1:30 Lunch

1:30 – 2:00 Deep Learning Theory

2:00 – 2:30 TensorFlow

2:30 – 4:00 Deep Learning on Handwritten Digits with Keras/TensorFlow

4:00 – 5:00 Transfer Learning and Deep Dream

5:00 – 5:30 Where to go from here.

5:30 – 7:00 Hang out on the Roof

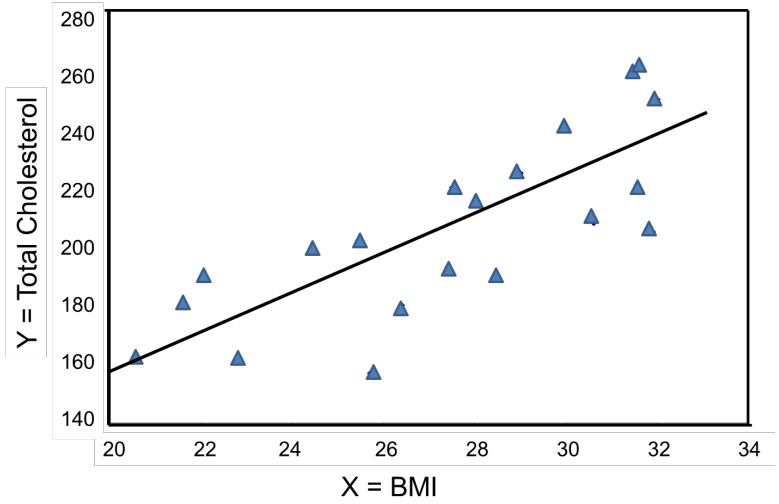
What do we mean by “machine learning”?

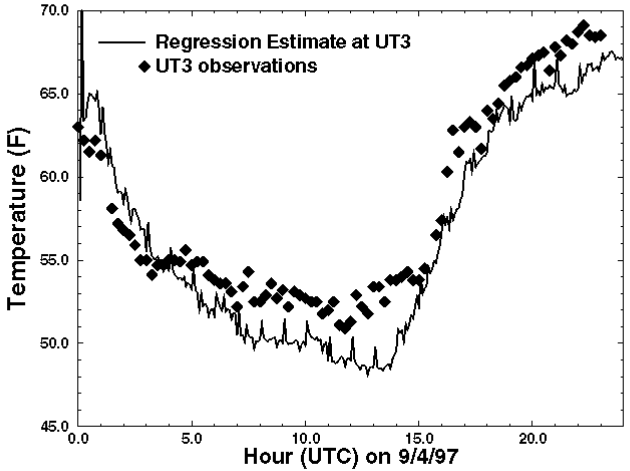
5 kinds of questions

1. How much / how many?
2. Which category?
3. Which groups?
4. Is it weird?
5. Which action?

How much / how many? (Regression)

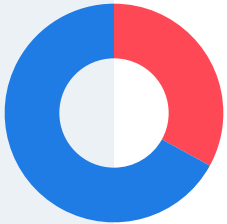
1. What temperature will it be next Tuesday?
2. How promising is this sales lead?
3. How many Twitter followers will I have by the end of the year?





Age	Zip Code	Income
58	02138	$95,824
73	94110	$20,708
59	45323	$82,152
66	34134	$25,334

Training Data

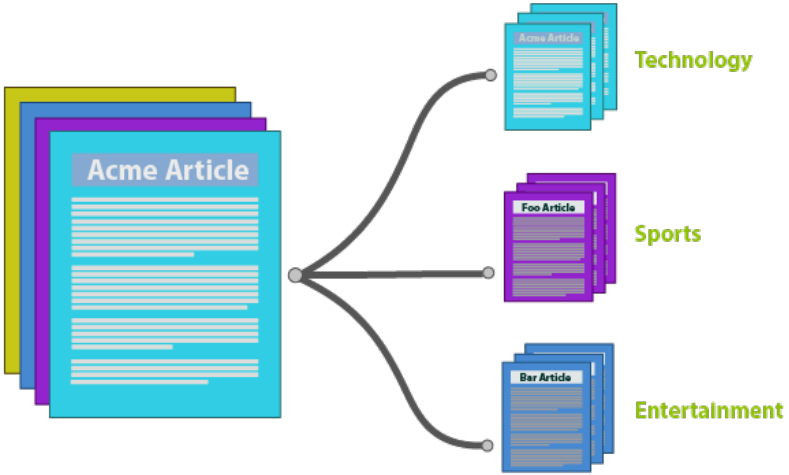


Age	Zip Code	Income
73	01233	
61	34134	
47	92349	
44	81112	

Test Data

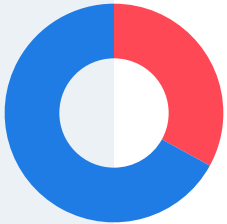
Which category? (Classification)

1. Is this a picture of a cat or a dog?
2. Is this email message spam or not?
3. What is the topic of this news article?



Age	Income	Default
58	$95,824	True
73	$20,708	False
59	$82,152	False
66	$25,334	True

Training Data

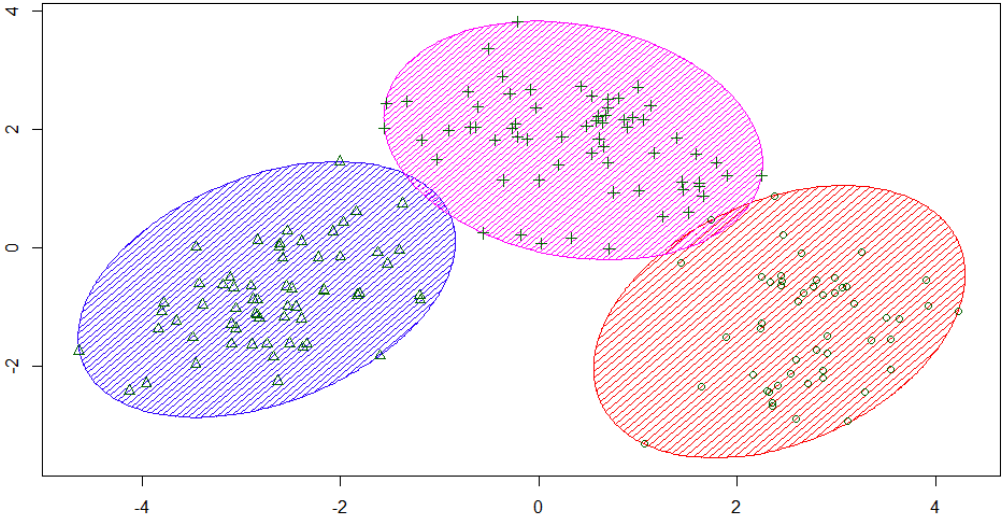


Age	Income	Default
73	$53,445	
61	$36,679	
47	$90,422	
44	$79,040	

Test Data

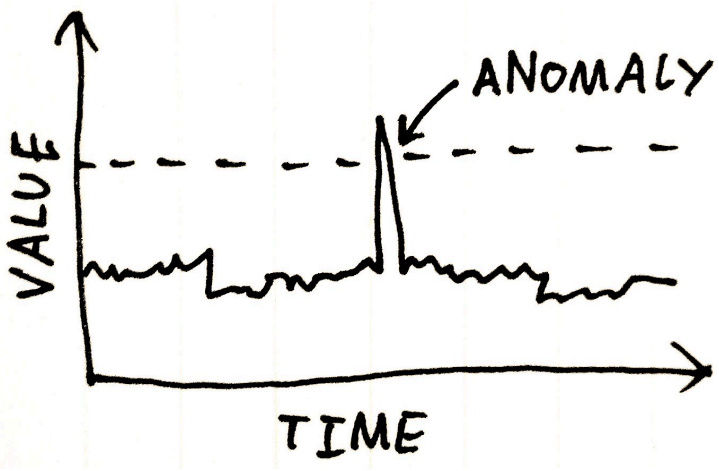
Which groups? (Clustering)

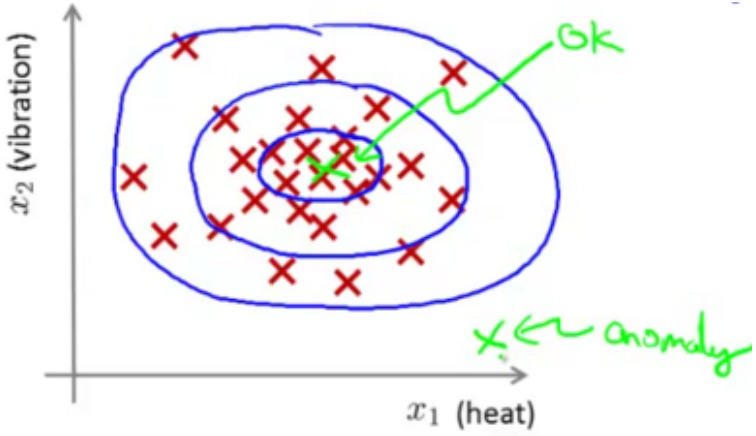
1. Which customers have similar shopping preferences?
2. Which of these images look similar?
3. How can I group these documents together by topic?



Is it weird? (Anomaly detection)

1. Is this sensor reading unusually high?
2. Is someone trying to hack into my system?
3. Does this credit card transaction appear fraudulent?



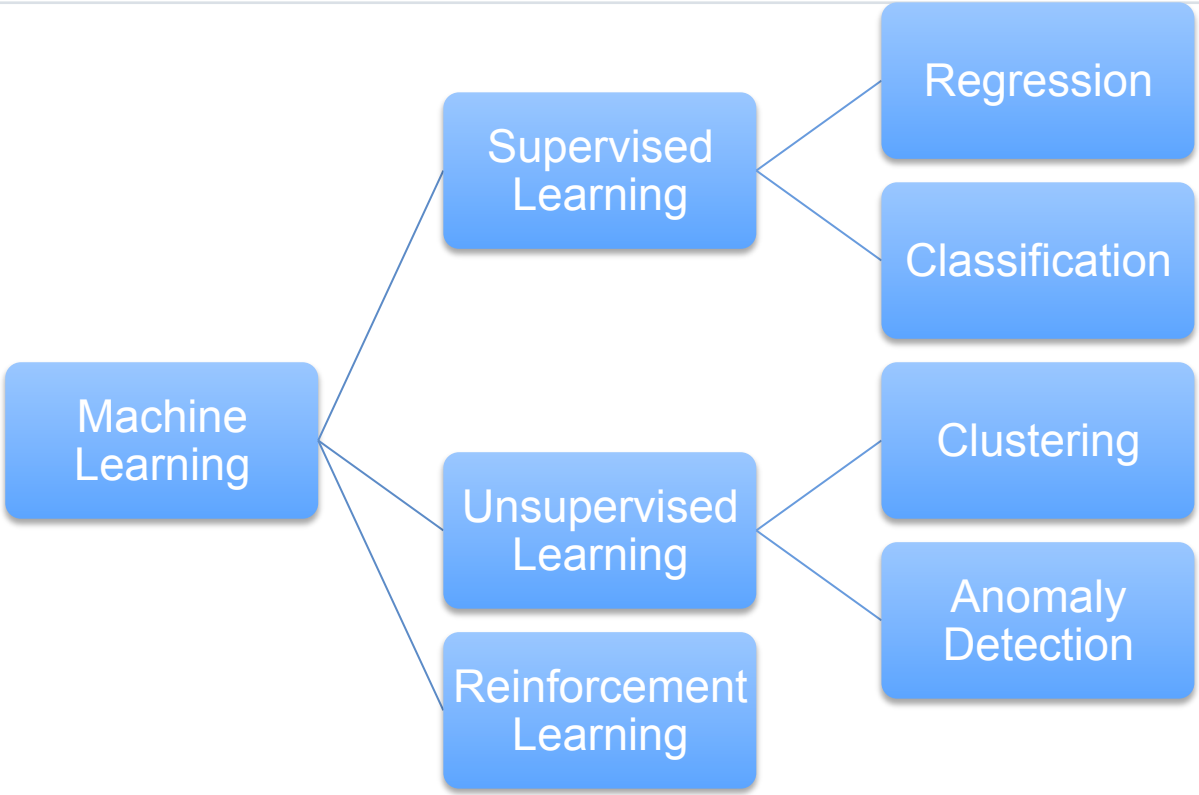


Which action? (Reinforcement Learning)

1. Should I speed up or stop at this yellow light?
2. Should I raise or lower the temperature?
3. Do I continue vacuuming or do I return to my charging station?

Supervised vs Unsupervised Learning

Machine Learning by Question Type



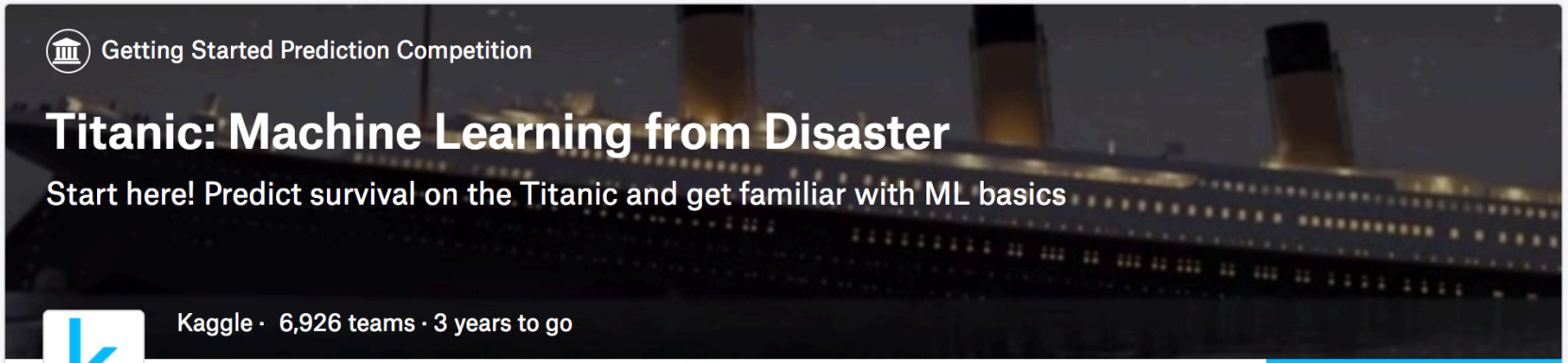
Let’s go car shopping

I want to buy a used car.

I don’t want one with more than 50K miles on it.

How much should I expect to pay?

Let’s go to the dealership and look at prices!



Overview

Start here if...

You're new to data science and machine learning, or looking for a simple intro to the Kaggle prediction competitions.

Competition Description

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912,

Intel & MobileODT Cervical Cancer Screening — $100,000
Which cancer treatment will be most effective? — 659 teams
Featured · a month to go

Google Cloud & YouTube-8M Video Understanding Challenge — $100,000
Can you produce the best video tag predictions? — 601 teams
Featured · 11 days to go

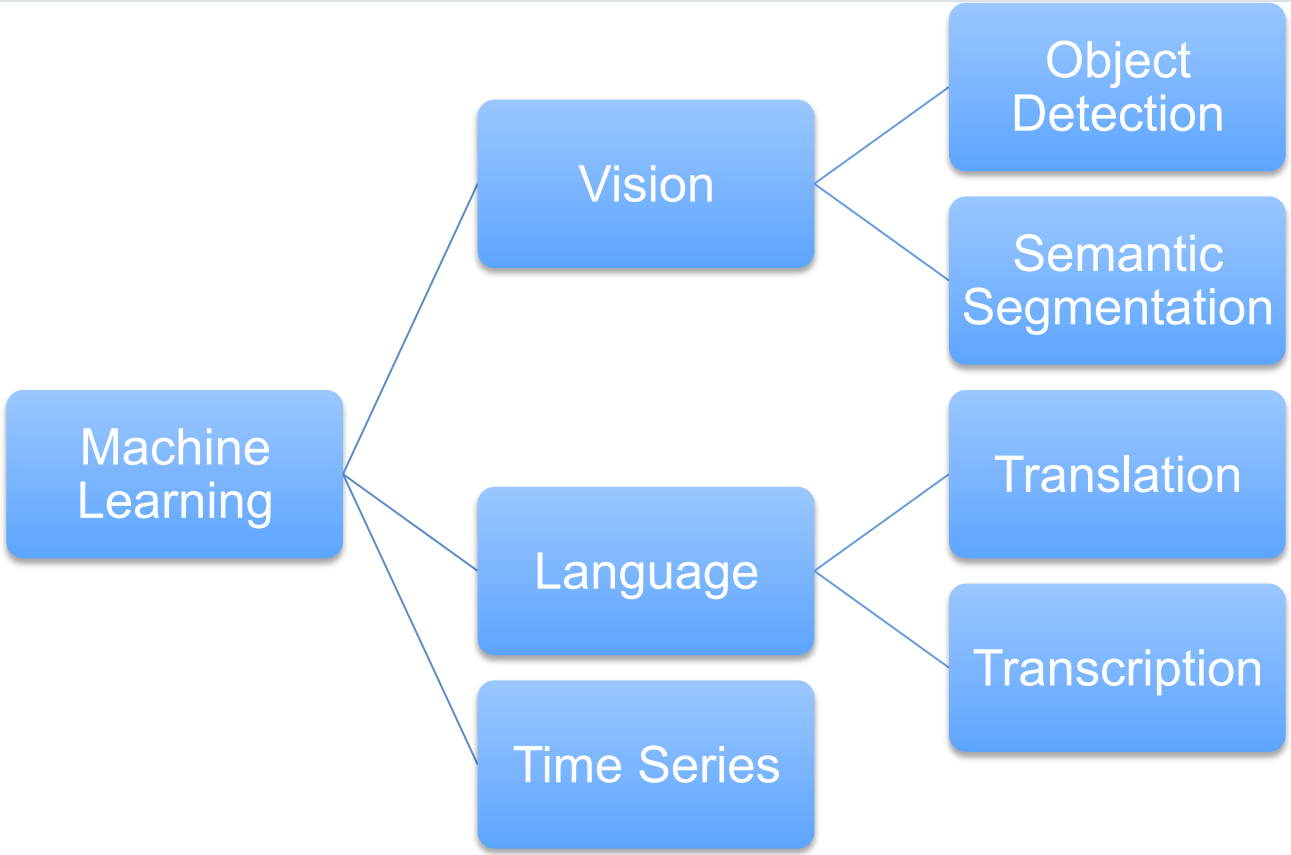
Planet: Understanding the Amazon from Space — $60,000
Use satellite data to track the human footprint in the Amazon rainforest — 253 teams
Featured · 2 months to go

Instacart Market Basket Analysis — $25,000
Which products will an Instacart consumer purchase again? — 189 teams
Featured · 3 months to go

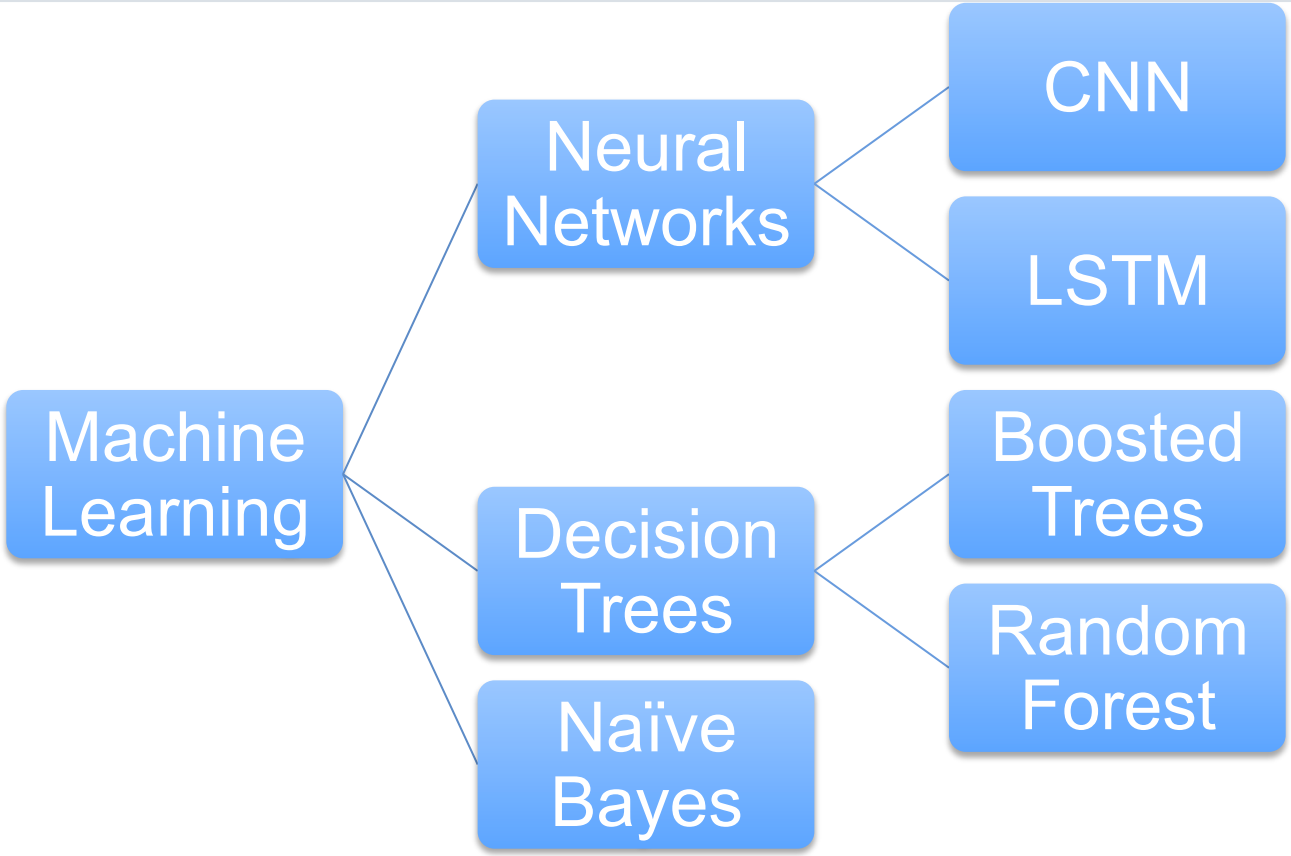
Sberbank Russian Housing Market — $25,000
Can you predict realty price fluctuations in Russia's volatile economy? — 1,920 teams
Featured · a month to go

NOAA Fisheries Steller Sea Lion Population Count — $25,000
How many sea lions do you see? — 178 teams
Featured · a month to go

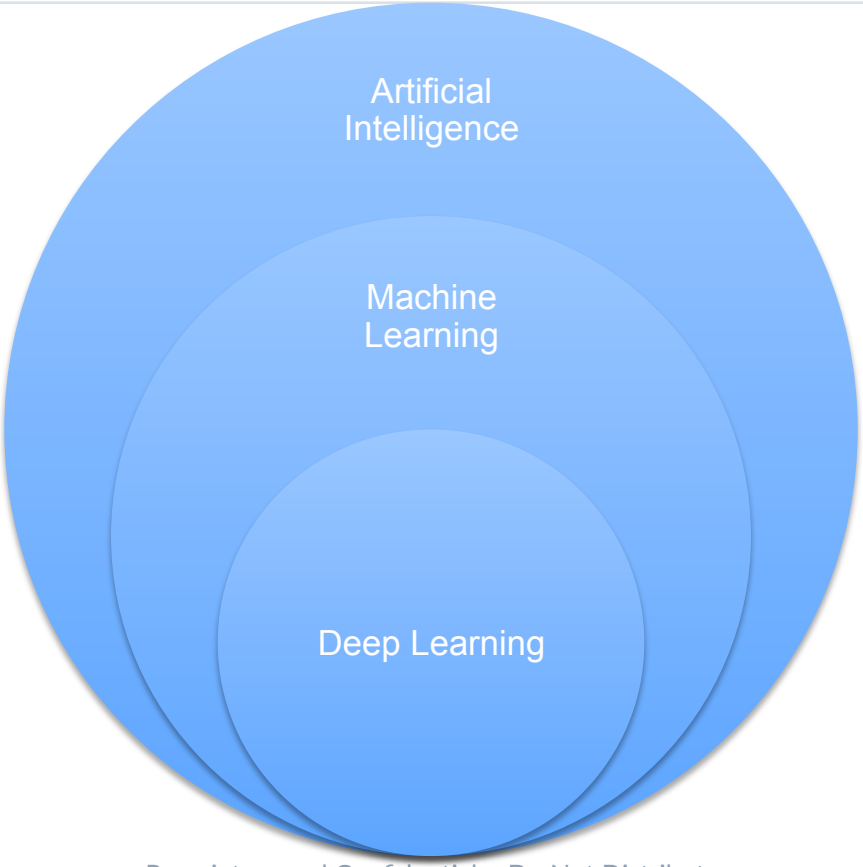
Machine Learning by Application



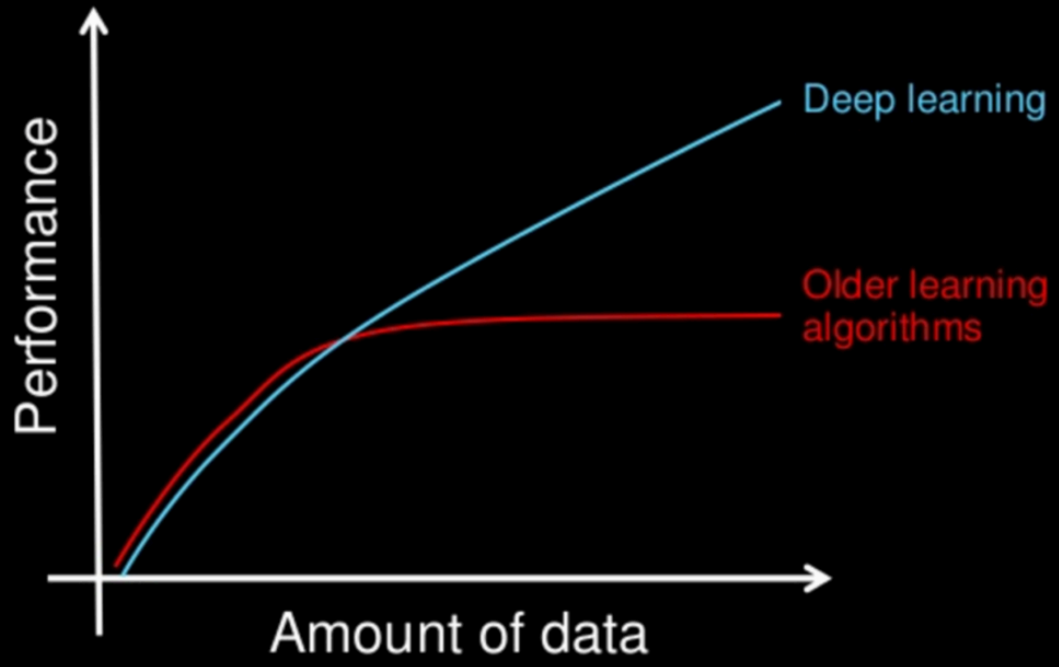
Machine Learning by Technique



What's the difference between Deep Learning, Machine Learning and Artificial Intelligence???

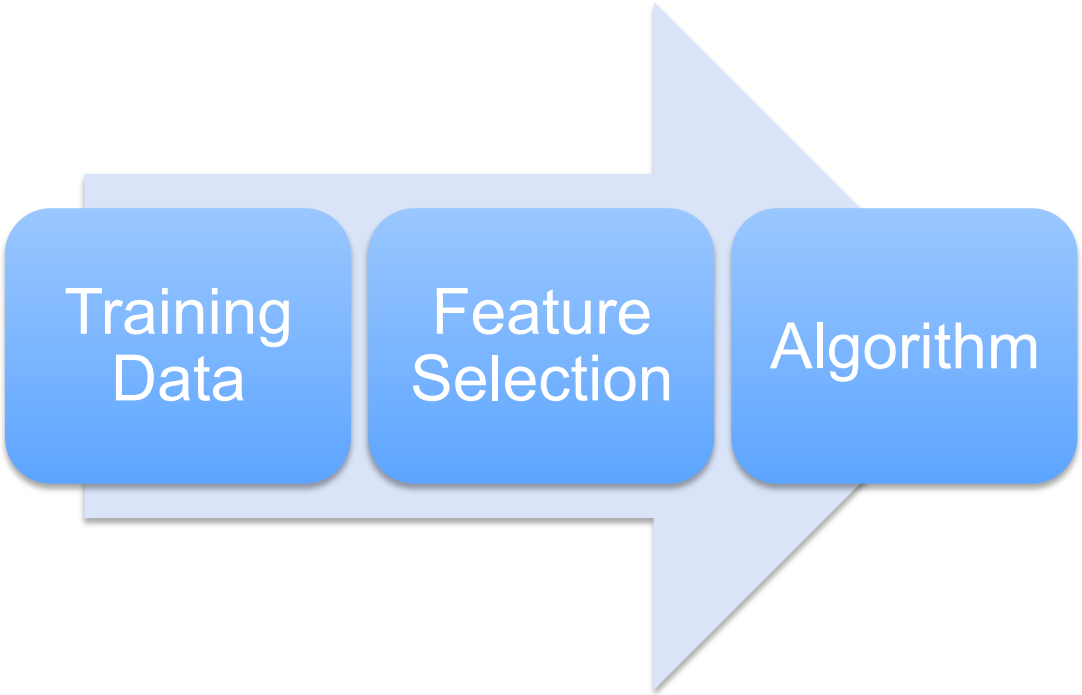


Why deep learning



How do data science techniques scale with amount of data?

Three Steps in all Supervised Machine Learning

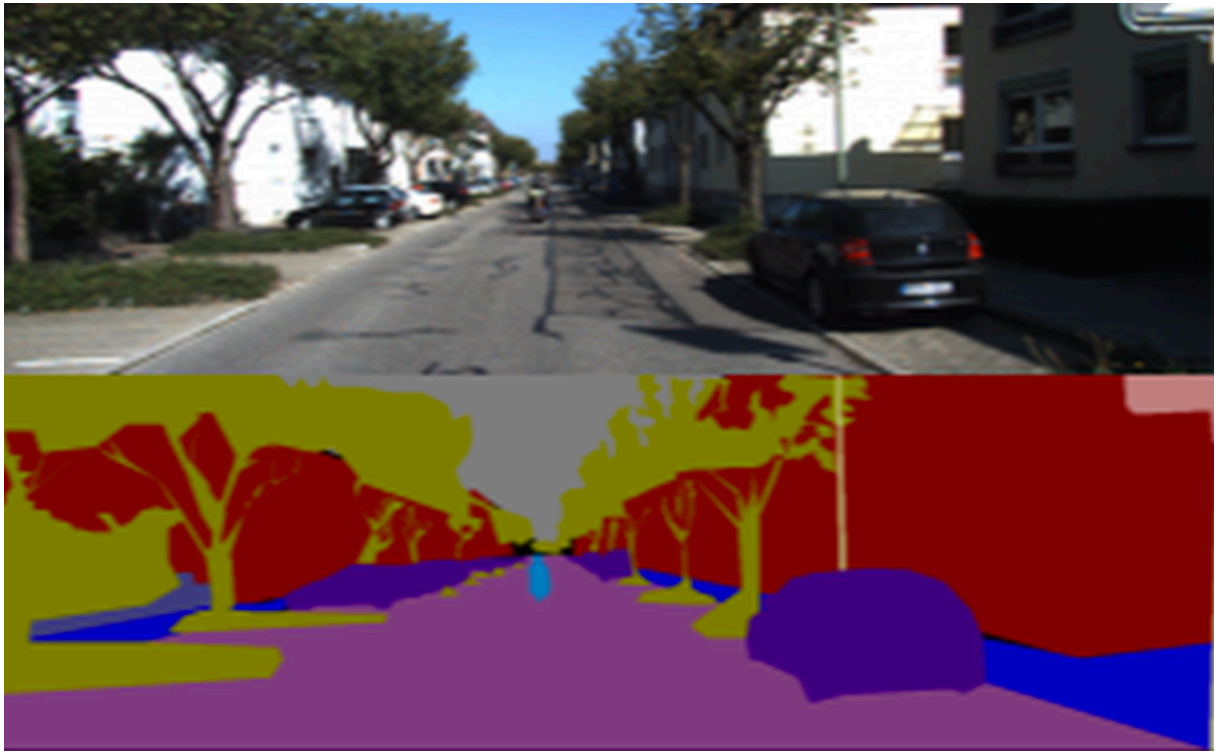


Framing Problems: Object Recognition

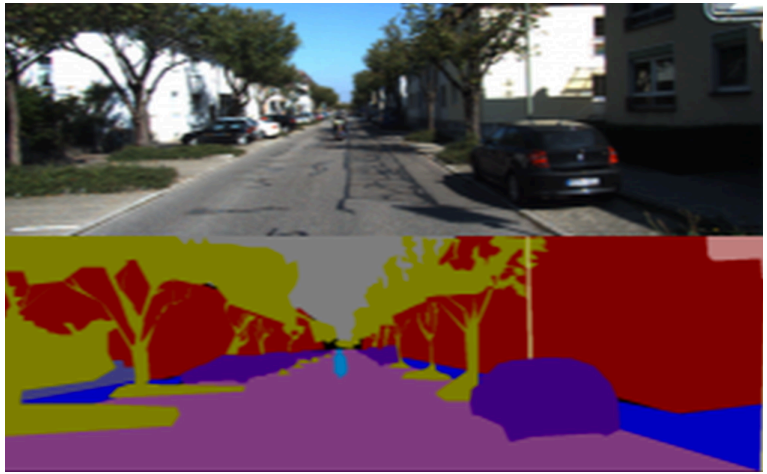
Framing Problems: Object Recognition

(0,0)	(0,1)	(0,2)	....	(1,0)	(1,1)	(1,2)	....	(2,0)	(2,1)	(2,2)	Label
23	15	3		56	23	12		56	23	12	Cat

Framing Problems: Vision Applications



Semantic Segmentation



(0,0)	(0,1)	(0,2)
23	15	3

....

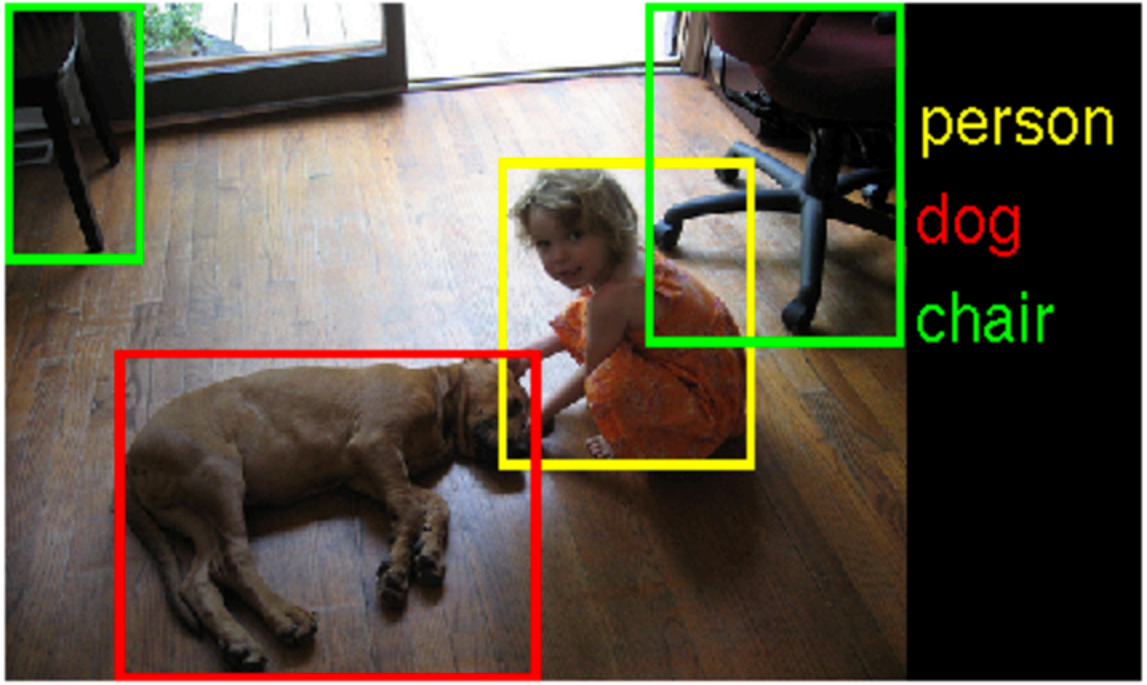
(1,0)	(1,1)	(1,2)
56	23	12

....

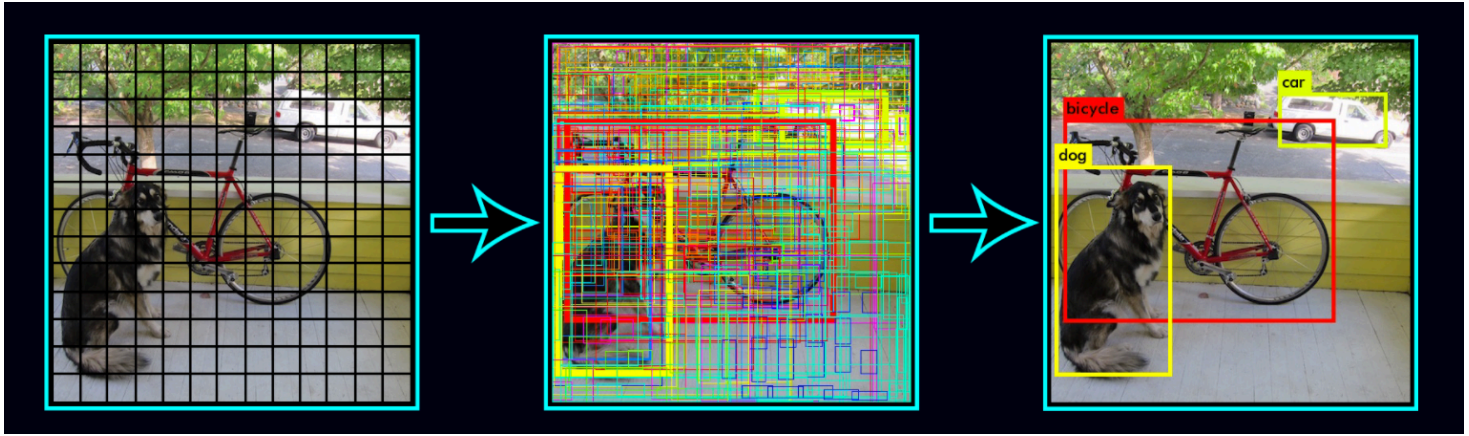
Label (0,0)	Label (0,1)
Tree	Tree

Label (1,0)	Label (1,1)
Car	Car

Bounding Box

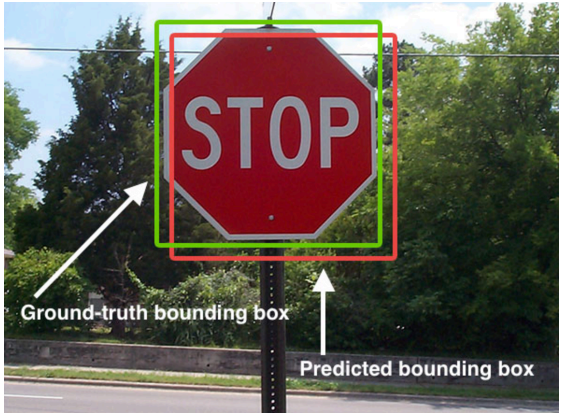


Bounding Box as Classification



(0,0)	(0,1)	(0,2)		(1,0)	(1,1)	(1,2)	Box Upper Left X	Box Upper Left Y	Good Box	Object
23	15	3	....	123	89	56	15	90	True	Bicycle
23	15	3		123	89	56	23	23	False	Bicycle
23	15	3		123	89	56	56	23	False	Dog

Bounding Box as Regression



(0,0)	(0,1)	(0,2)		(1,0)	(1,1)	(1,2)	Box Upper Left X	Box Upper Left Y
23	15	3	....	123	89	56		
123	143	23		78	54	1		
12	17	6		90	9	30		

Project for Today

Judge Emotion About Brands & Products

Instructions ▲

In this job you will see tweets about several brands and products. Does the tweet include emotion directed at a brand, product, or user experience? If so, is it negative or positive?

Then, please select the brand, product, or user experience that applies. In most cases, there is a single best answer. If the tweet is about an iPad app, there's no need to also check the iPad box.

Brands considered are Apple and Google. **Products** considered are iPad, iPhone, and Android (phones or tablets). Use the "other" category for other products/services, like an emotion directed toward a Google Calendar. **User experiences** are mentions of using an application (App) on either iPad/iPhone or Android.

Note that in some cases links have been replaced with {link} and mentions have been replaced with @mention.

I just noticed DST is coming this weekend. How many iPhone users will be an hour late at SXSW come Sunday morning? #SXSW #iPhone

Is there an emotion directed at a brand or product?

- (•) Positive emotion
- ( ) Negative emotion
- ( ) I can't tell

What the Data Looks Like

tweet_text	emotion_in_tweet_is_directed_at	is_there_an_emotion_directed_at_a_brand_or_product
.@wesley83 I have a 3G iPhone. After 3 hrs tweeting at #RISE_Austin, it was dead! I need to	iPhone	Negative emotion
@jessedee Know about @fludapp ? Awesome iPad/iPhone app that you'll likely appreciate f	iPad or iPhone App	Positive emotion
@swonderlin Can not wait for #iPad 2 also. They should sale them down at #SXSW.	iPad	Positive emotion
@sxsw I hope this year's festival isn't as crashy as this year's iPhone app. #sxsw	iPad or iPhone App	Negative emotion
@sxtxstate great stuff on Fri #SXSW: Marissa Mayer (Google), Tim O'Reilly (tech books/conf	Google	Positive emotion
@teachntech00 New iPad Apps For #SpeechTherapy And Communication Are Showcased At	The #SXSW Conference http://ht.ly/49n4M #iear #edchat	No emotion toward brand or product
		No emotion toward brand or product
#SXSW is just starting, #CTIA is around the corner and #googleio is only a hop skip and a jum	Android	Positive emotion
Beautifully smart and simple idea RT @madebymany @thenextweb wrote about our #hollei	iPad or iPhone App	Positive emotion
Counting down the days to #sxsw plus strong Canadian dollar means stock up on Apple gear	Apple	Positive emotion
Excited to meet the @samsungmobileus at #sxsw so I can show them my Sprint Galaxy S stil	Android	Positive emotion
Find & Start Impromptu Parties at #SXSW With @HurricaneParty http://bit.ly/gVLrIn I (	Android App	Positive emotion
Foursquare ups the game, just in time for #SXSW http://j.mp/grN7pK) - Still prefer @Gowall	Android App	Positive emotion
Gotta love this #SXSW Google Calendar featuring top parties/ show cases to check out. RT @	Other Google product or service	Positive emotion
Great #sxsw ipad app from @madebymany: http://tinyurl.com/4nqv92l	iPad or iPhone App	Positive emotion
haha, awesomely rad iPad app by @madebymany http://bit.ly/hTdFim #hollergram #sxsw	iPad or iPhone App	Positive emotion
Holler Gram for iPad on the iTunes App Store - http://t.co/kfN3f5Q (via @marc_is_ken) #sxsw		No emotion toward brand or product
I just noticed DST is coming this weekend. How many iPhone users will be an hour late at SX	iPhone	Negative emotion
Just added my #SXSW flights to @planely. Matching people on planes/airports. Also downlo	iPad or iPhone App	Positive emotion
Must have #SXSW app! RT @malbonster: Lovely review from Forbes for our SXSW iPad app	iPad or iPhone App	Positive emotion
Need to buy an iPad2 while I'm in Austin at #sxsw. Not sure if I'll need to Q up at an Austin A	iPad	Positive emotion
Oh. My. God. The #SXSW app for iPad is pure, unadulterated awesome. It's easier to browse	iPad or iPhone App	Positive emotion
Okay, this is really it: yay new @Foursquare for #Android app!!!!11 kthxbai. #sxsw	Android App	Positive emotion
Photo: Just installed the #SXSW iPhone app, which is really nice! http://tumblr.com/x6t1pi6	iPad or iPhone App	Positive emotion
Really enjoying the changes in Gowalla 3.0 for Android! Looking forward to seeing what else	Android App	Positive emotion
RT @LaurieShook: I'm looking forward to the #SMCDallas pre #SXSW party Wed., and hopin	iPad	Positive emotion
RT haha, awesomely rad iPad app by @madebymany http://bit.ly/hTdFim #hollergram #sxsv	iPad or iPhone App	Positive emotion
someone started an #austin @PartnerHub group in google groups, pre-#sxsw. great idea	Other Google product or service	Positive emotion
The new #4sq3 looks like it is going to rock. Update for iPhone and Android should push ton	iPad or iPhone App	Positive emotion
They were right, the @gowalla 3 app on #android is sweeeeet! Nice job by the team there. #	Android App	Positive emotion
Very smart from @madebymany #hollergram iPad app for #sxsw! http://t.co/A3xvWc6 (ma	iPad or iPhone App	Positive emotion
You must have this app for your iPad if you are going to #SXSW http://itunes.apple.com/us/	iPad or iPhone App	Positive emotion
Attn: All #SXSW frineds, @mention Register for #GDGTLive and see Cobra iRadar for Android. {link}		No emotion toward brand or product
Anyone at #sxsw want to sell their old iPad?		No emotion toward brand or product
Anyone at #SXSW who bought the new iPad want to sell their older iPad to me?		No emotion toward brand or product
At #sxsw, Oooh, RT @mention Google to Launch Major New Social Network Called Circles, Possibly Today {link}		No emotion toward brand or product

Load the Data (load_data.py)

```
1  import pandas as pd
2  import numpy as np
3
4
5  df = pd.read_csv('tweets.csv')
6  target = df['is_there_an_emotion_directed_at_a_brand_or_product']
7  text = df['tweet_text']
8
9  print target[0:5]
10 print text[0:5]
```


How do we turn the text into numbers?

	A	aardvark	…	hate	I	iphone	love	my	…	Zyzzyva
I love my iphone	0	0	…	0	1	1	1	1	…	0
I hate my iphone	0	0	…	1	1	1	0	1	…	0

http://www.python-course.eu/text_classification_python.php

How do we turn the text into numbers? (feature_extraction.py)

load_data.py | feature_extraction.py | feature_extraction_... | feature_extraction_... | tweets.csv

```
1  import pandas as pd
2  import numpy as np
3
4
5  df = pd.read_csv('tweets.csv')
6  target = df['is_there_an_emotion_directed_at_a_brand_or_product']
7  text = df['tweet_text']
8
9  text = text[pd.notnull(text)]
10 target = target[pd.notnull(text)]
11
12 from sklearn.feature_extraction.text import CountVectorizer
13 count_vect = CountVectorizer()
14 count_vect.fit(text)
15
16 print count_vect.vocabulary_.get(u'3g')
```


Handling weird input data (feature_extraction_2.py)

load_data.py | feature_extraction.py | feature_extraction_2.py | feature_extraction_... | tweets.csv

```
1  import pandas as pd
2  import numpy as np
3
4
5  df = pd.read_csv('tweets.csv')
6  target = df['is_there_an_emotion_directed_at_a_brand_or_product']
7  text = df['tweet_text']
8
9  fixed_text = text[pd.notnull(text)]
10 fixed_target = target[pd.notnull(text)]
11
12 from sklearn.feature_extraction.text import CountVectorizer
13 count_vect = CountVectorizer()
14 count_vect.fit(fixed_text)
15
16 print count_vect.vocabulary_.get(u'3g')
```


Run the feature extraction (feature_extraction_3.py)

load_data.py | feature_extraction.py | feature_extraction_... | feature_extraction_3.py | tweets.csv

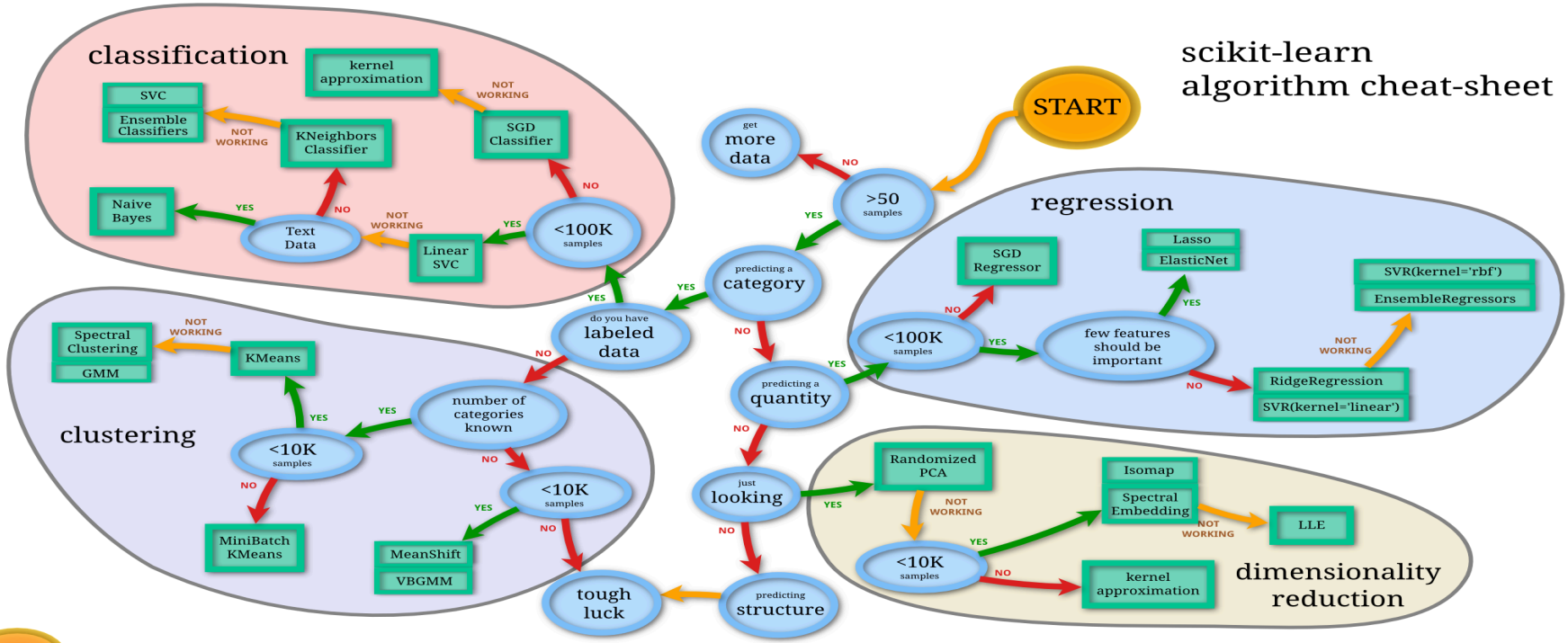
```
1  import pandas as pd
2  import numpy as np
3
4
5  df = pd.read_csv('tweets.csv')
6  target = df['is_there_an_emotion_directed_at_a_brand_or_product']
7  text = df['tweet_text']
8
9  fixed_text = text[pd.notnull(text)]
10 fixed_target = target[pd.notnull(text)]
11
12 from sklearn.feature_extraction.text import CountVectorizer
13 count_vect = CountVectorizer()
14 count_vect.fit(fixed_text)
15
16 counts = count_vect.transform(fixed_text)
17
18 print counts
19
```

Lots of important choices already!

- `class sklearn.feature_extraction.text.CountVectorizer(input=u'content', encoding=u'utf-8', decode_error=u'strict', strip_accents=None, lowercase=True, preprocessor=None, tokenizer=None, stop_words=None, token_pattern=u'(?u)\b\w\w+\b', ngram_range=(1, 1), analyzer=u'word', max_df=1.0, min_df=1, max_features=None, vocabulary=None, binary=False, dtype=<type 'numpy.int64'>)`

- Should we remove really rare words?
- Should we remove really common words?
- Should we remove “stop words”?
- Should we lower case all the words?
- What is a word?
  - For those at #SXSW: Apple sets up 5,000-square-foot temporary store at SXSW to sell new iPads, test potential traffic
  - If ur not at the #google #aclu 80's party....u should be! #sxsw
  - My iPhone battery at 100%. #winning at #SXSW

Choose an algorithm



http://scikit-learn.org/stable/tutorial/machine_learning_map/

Run the Algorithm (classifier.py)

```
13  count_vect = CountVectorizer()
14  count_vect.fit(fixed_text)
15
16  counts = count_vect.transform(fixed_text)
17
18  from sklearn.naive_bayes import MultinomialNB
19  nb = MultinomialNB()
20  nb.fit(counts, fixed_target)
21
22  print nb.predict(count_vect.transform(["I love my iphone!!!"]))
```


Lots of scary choices!

- MultinomialNB vs GaussianNB vs BernoulliNB

Attributes:	**class_prior_** : array, shape (n_classes,) probability of each class. **class_count_** : array, shape (n_classes,) number of training samples observed in each class. **theta_** : array, shape (n_classes, n_features) mean of each feature per class **sigma_** : array, shape (n_classes, n_features) variance of each feature per class

- Do I want to weight one class more than the other?
- Do I want to monkey around with the algorithm?

How well is the algorithm working? (test_algorithm_1.py)

```
21
22  predictions = nb.predict(counts)
23  sum(predictions == fixed_target)
24
```


Test/Train Split (test_algorithm_2.py)

```
21  nb.fit(counts[0:6000], fixed_target[0:6000])
22
23  predictions = nb.predict(counts[6000:9092])
24  print sum(predictions == fixed_target[6000:9092])
```

Label Distributions (test_algorithm_3.py)

```
46 from sklearn.metrics import confusion_matrix
47 ## We're ignoring "I can't tell" here for simplicity
48 label_list = ['Positive emotion', 'No emotion toward brand or product', 'Neg
   ative emotion']
49 cm = confusion_matrix(target[6000:9092], predictions, labels=label_list)
50 print("Labels in data:")
51 print(label_list)
52 print("Rows: actual labels, Columns: Predicted labels")
53 print(cm)
```

```
[[ 527  454    7]
 [ 362 1504   24]
 [  65   75   22]]
```


Label Distributions (test_algorithm_3.py)

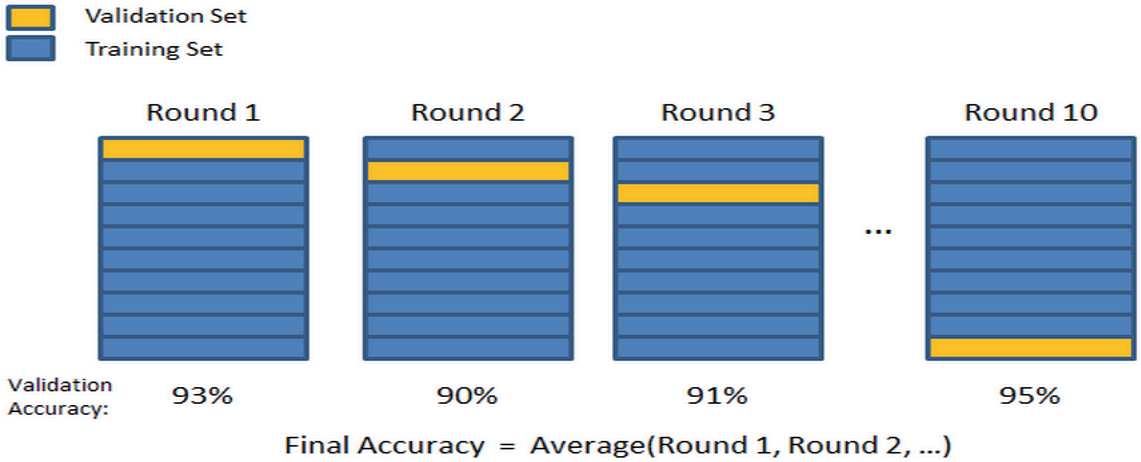
		Predicted		
		Positive	Neutral	Negative
Actual	Positive	527	454	7
	Neutral	362	1504	24
	Negative	65	75	22

What patterns do you see in the confusion matrix?
Are all the labels equally represented?
Is the classifier equally accurate on all the labels?
Do you see a relationship between label representation and classifier predictions?

Baselines (test_algorithm_dummy.py)

```
19  from sklearn.dummy import DummyClassifier
20
21  nb = DummyClassifier(strategy='most_frequent')
22
23  nb.fit(counts[0:6000], fixed_target[0:6000])
24
25  predictions = nb.predict(counts[6000:9092])
26  print sum(predictions == fixed_target[6000:9092])
27
```


Cross Validation



https://chrisjmccormick.wordpress.com/2013/07/31/k-fold-cross-validation-with-matlab-code/

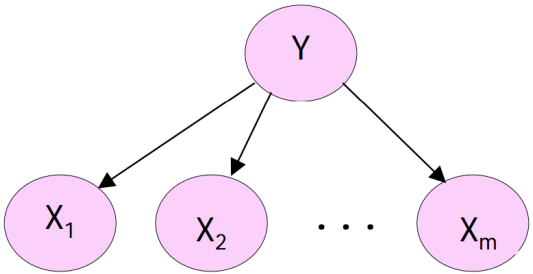
Cross Validation (test_algorithm_cross_validation.py)

```
20
21  from sklearn import cross_validation
22
23  scores = cross_validation.cross_val_score(nb, counts, fixed_target, cv=10)
24  print scores
25  print scores.mean()
26
```


Cross Validation With Dummy (test_algorithm_cross_validation_dummy.py)

```
21  nb = DummyClassifier(strategy='most_frequent')
22
23  from sklearn import cross_validation
24
25  scores = cross_validation.cross_val_score(nb, counts, fixed_target, cv=10)
26  print scores
27  print scores.mean()
```

What is Naïve Bayes?



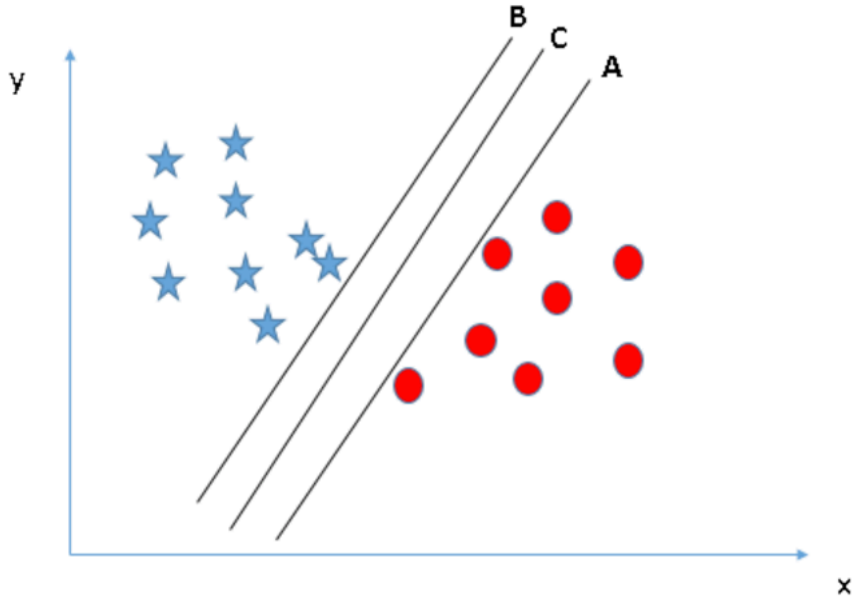
1. Estimate P(Y=v) as fraction of records with Y=v
2. Estimate $P(X_i=u \mid Y=v)$ as fraction of "Y=v" records that also have X=u.
3. To predict the Y value given observations of all the X_i values, compute

$$Y^{\text{predict}} = \text{argmax}\, P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

Naïve Bayes for Hackers

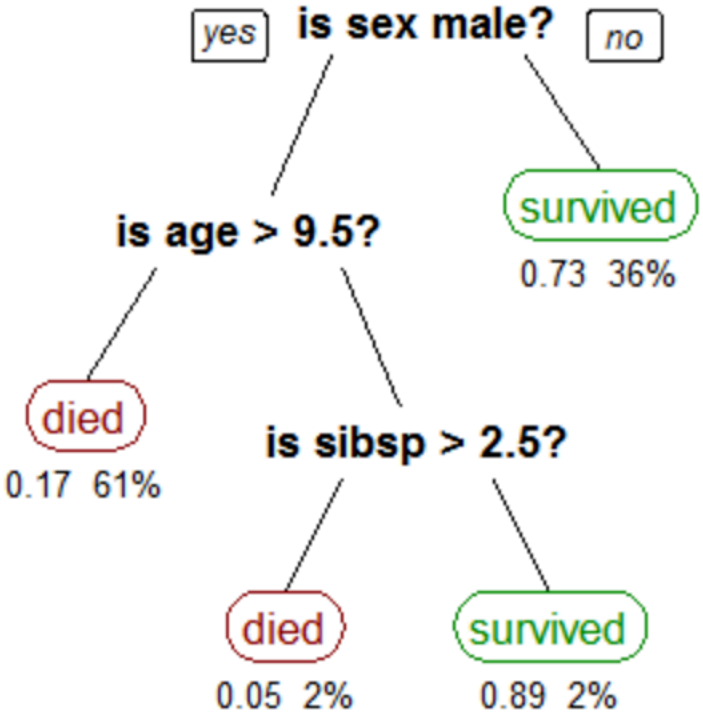
- http://norvig.com/spell-correct.html

Other Algorithms: SVM

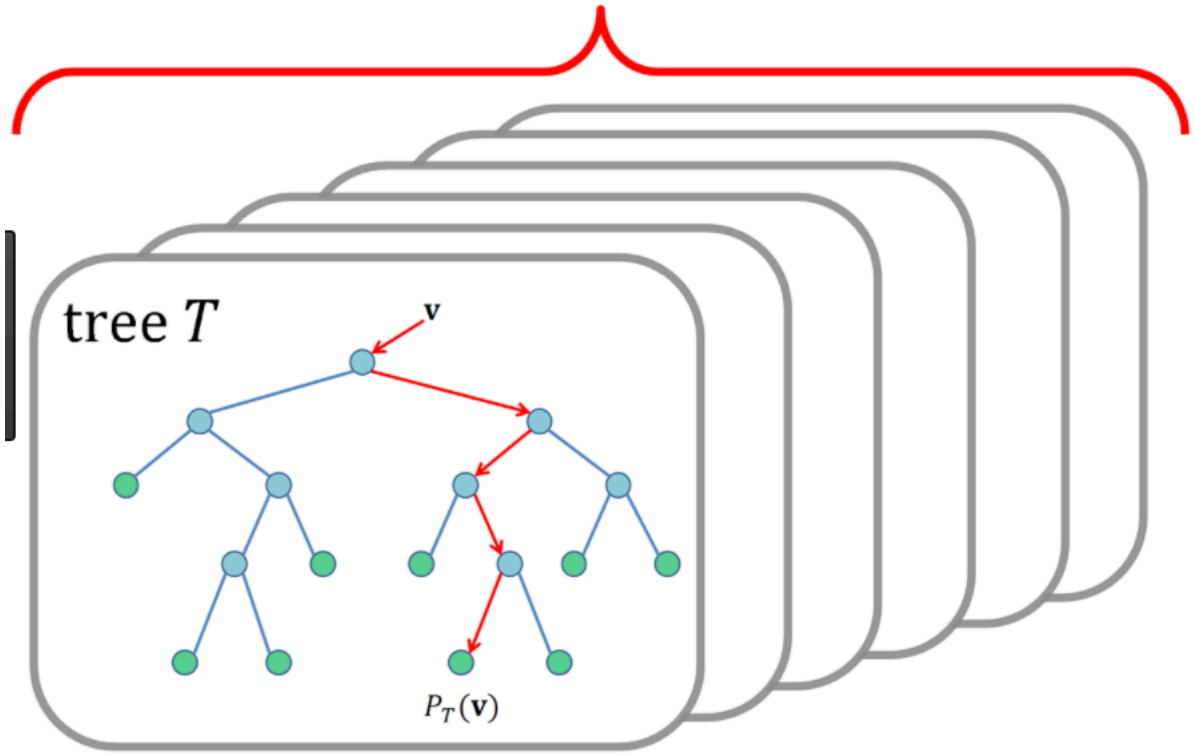


Cares more about correct classification than understanding probabilities.

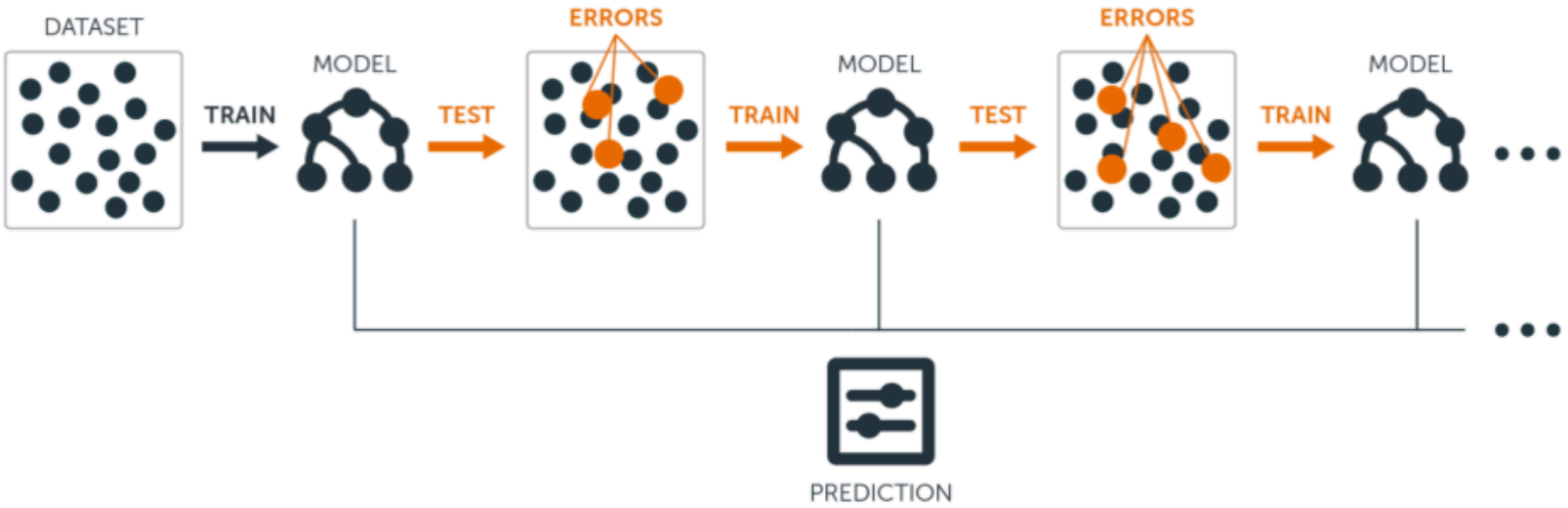
Decision Trees



Decision Forest



Boosted Trees



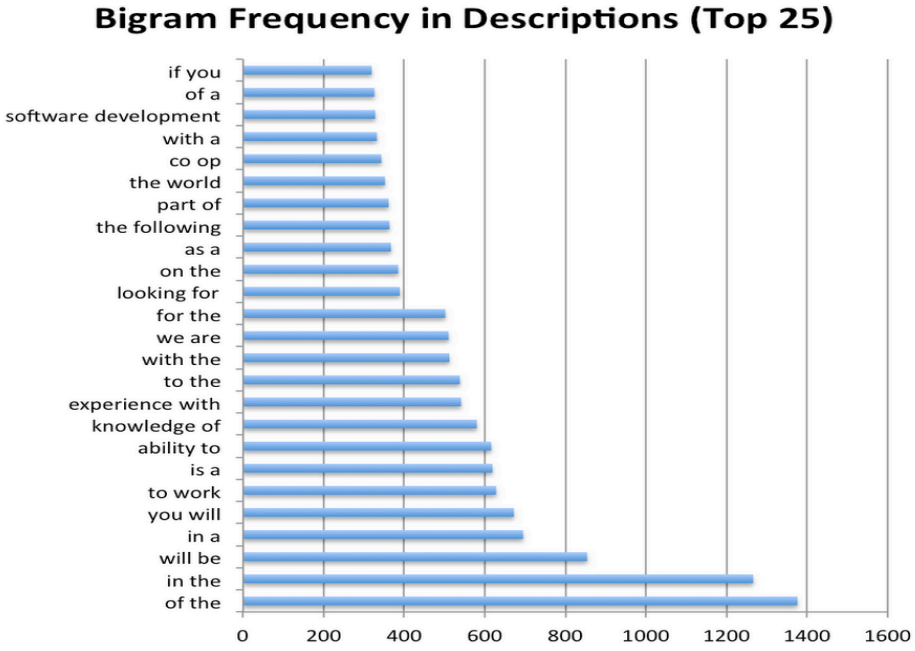
XGBoost

Pipelines (pipeline.py)

```
12  from sklearn.feature_extraction.text import CountVectorizer
13  from sklearn.naive_bayes import MultinomialNB
14  from sklearn.pipeline import Pipeline
15
16  p = Pipeline(steps=[('counts', CountVectorizer()),
17                      ('multinomialnb', MultinomialNB())])
18
19  p.fit(fixed_text, fixed_target)
20  print p.predict(["I love my iphone!"])
21
```


N-Grams

- Great vs. “Oh, great”



Bigrams (pipeline_bigrams.py)

```
p = Pipeline(steps=[('counts', CountVectorizer(ngram_range=(1, 2))),
                    ('multinomialnb', MultinomialNB())])

p.fit(fixed_text, fixed_target)
print p.named_steps['counts'].vocabulary_.get(u'garage sale')
print len(p.named_steps['counts'].vocabulary_)
```


Bigrams Accuracy (pipeline_bigrams_cross_validation.py)

```
p = Pipeline(steps=[('counts', CountVectorizer(ngram_range=(1, 2))),
                    ('multinomialnb', MultinomialNB())])

p.fit(fixed_text, fixed_target)

from sklearn import cross_validation

scores = cross_validation.cross_val_score(p, fixed_text, fixed_target, cv=
print scores
print scores.mean()
```


Feature Selection (feature_selection.py)

```
p = Pipeline(steps=[('counts', CountVectorizer(ngram_range=(1, 2))),
                    ('feature_selection', SelectKBest(chi2, k=10000)),
                    ('multinomialnb', MultinomialNB())])

p.fit(fixed_text, fixed_target)

from sklearn import cross_validation

scores = cross_validation.cross_val_score(p, fixed_text, fixed_target, cv
print scores
print scores.mean()
```


Grid Search (grid_search.py)

```
parameters = {
    'counts__max_df': (0.5, 0.75, 1.0),
    'counts__min_df': (1, 2, 3),
    'counts__ngram_range': ((1,1), (1,2)),
#     'feature_selection__k': (1000, 10000, 100000)
    }

grid_search = GridSearchCV(p, parameters, n_jobs=1, verbose=1, cv=10)

grid_search.fit(fixed_text, fixed_target)

print("Best score: %0.3f" % grid_search.best_score_)
print("Best parameters set:")
best_parameters = grid_search.best_estimator_.get_params()
for param_name in sorted(parameters.keys()):
    print("\t%s: %r" % (param_name, best_parameters[param_name]))
```


Make Your Own Features!

- Overwrite “fit” (optional) and “transform”
- Some Ideas
  - # of exclamation points
  - Emoji
  - Length of tweet
  - Language of tweet

Get More Data

crowdflower.com/data-for-everyone

Machine Learning Platforms

Watson APIs

Conversation

Learn more | Documents

Document Conversion

Learn more | Documents

Language Translator

Learn more | Documents

Natural Language Classifier

Learn more | Documents

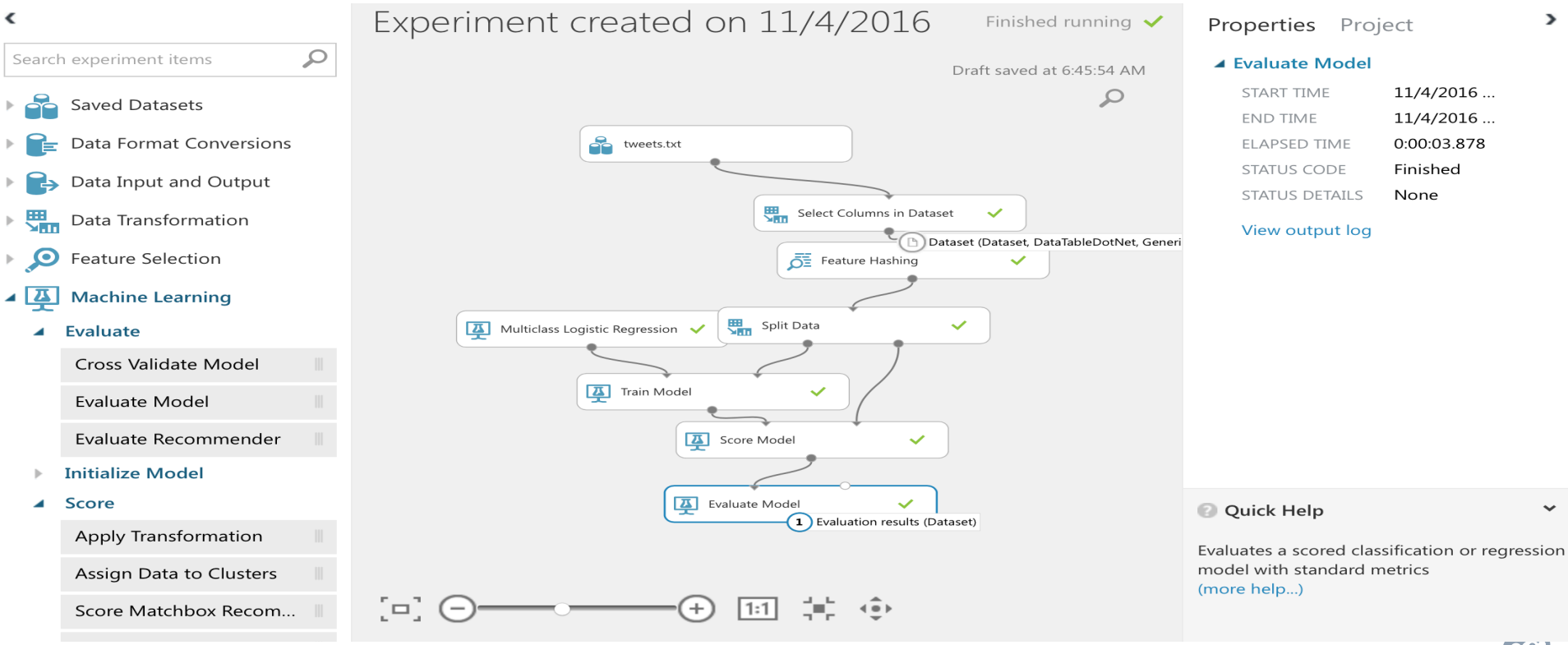
Natural Language Understanding

Learn more | Documents

Personality Insights

Learn more | Documents

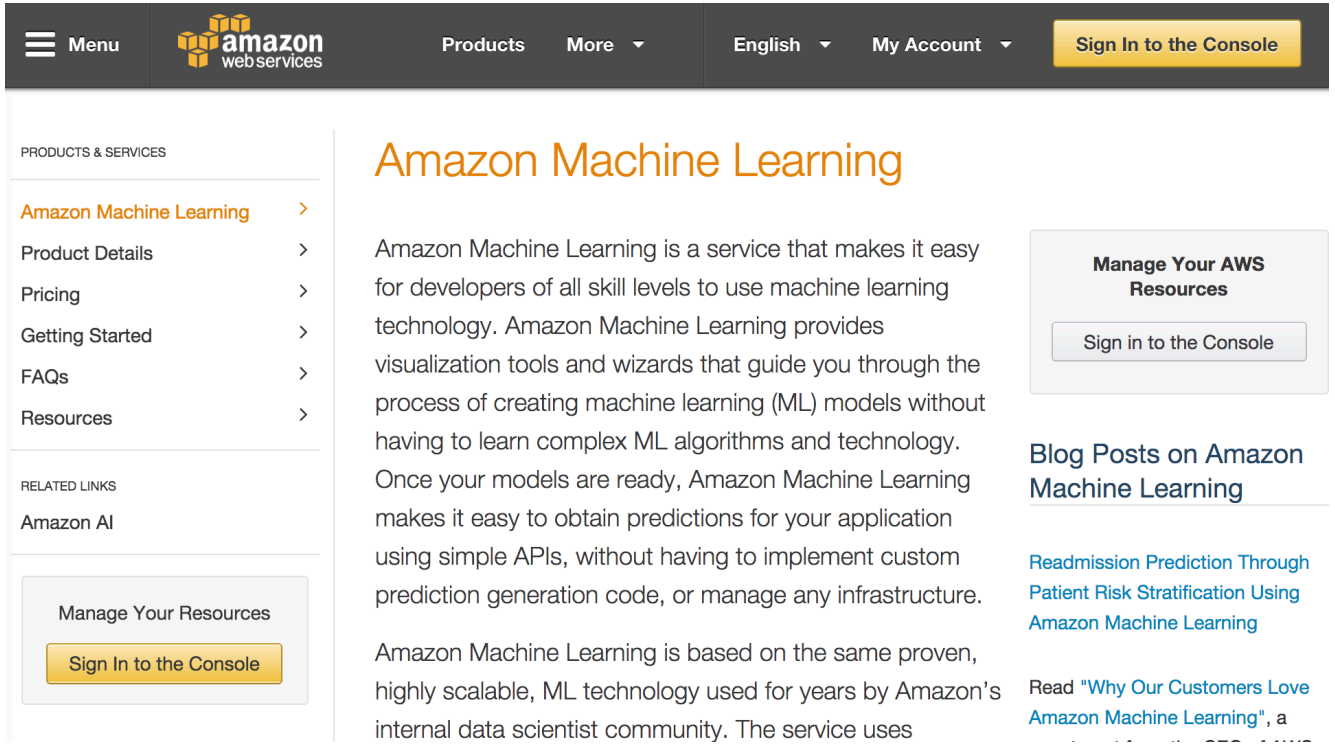
Azure-ML


Search experiment items
Saved Datasets
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
Evaluate
Cross Validate Model
Evaluate Model
Evaluate Recommender
Initialize Model
Score
Apply Transformation
Assign Data to Clusters
Score Matchbox Recom...
Experiment created on 11/4/2016
Finished running
Draft saved at 6:45:54 AM
tweets.txt
Select Columns in Dataset
Dataset (Dataset, DataTableDotNet, Generi
Feature Hashing
Multiclass Logistic Regression
Split Data
Train Model
Score Model
Evaluate Model
1 Evaluation results (Dataset)
1:1
Properties
Project
Evaluate Model
START TIME 11/4/2016 ...
END TIME 11/4/2016 ...
ELAPSED TIME 0:00:03.878
STATUS CODE Finished
STATUS DETAILS None
View output log
Quick Help
Evaluates a scored classification or regression model with standard metrics
(more help...)

Azure APIs

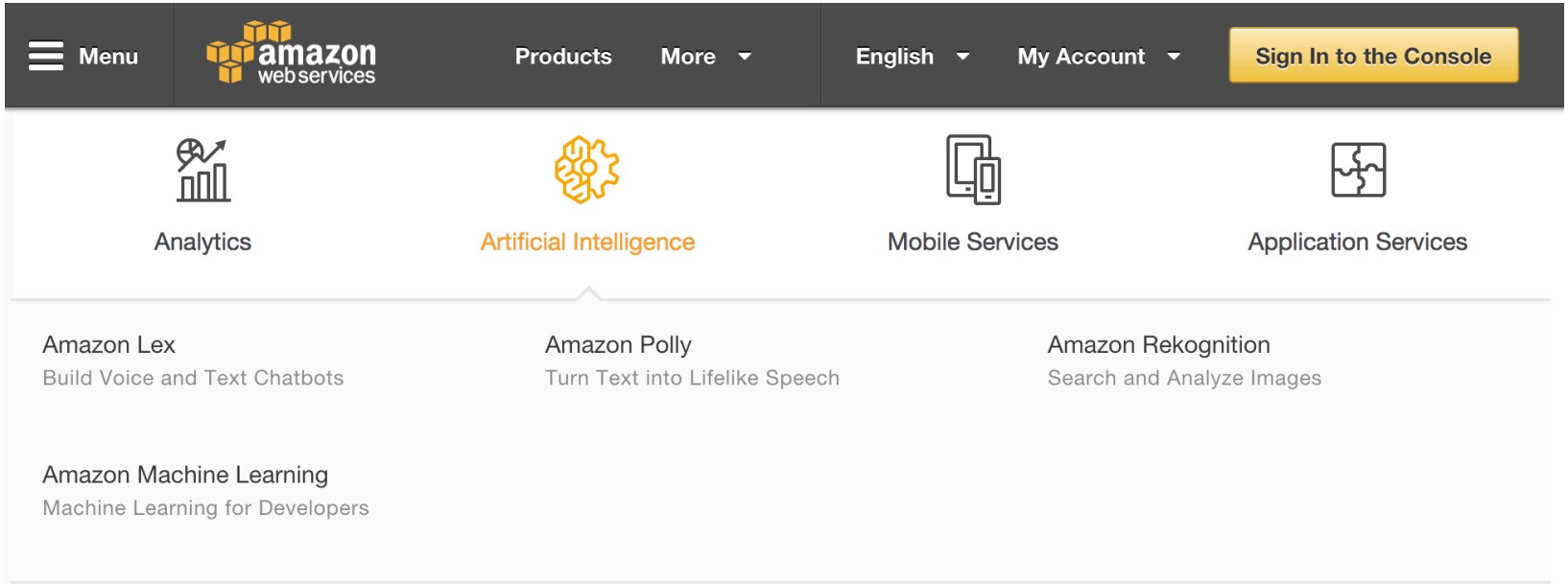
- https://www.oreilly.com/ideas/how-to-build-an-autonomous-voice-controlled-face-recognizing-drone-for-200

Amazon ML

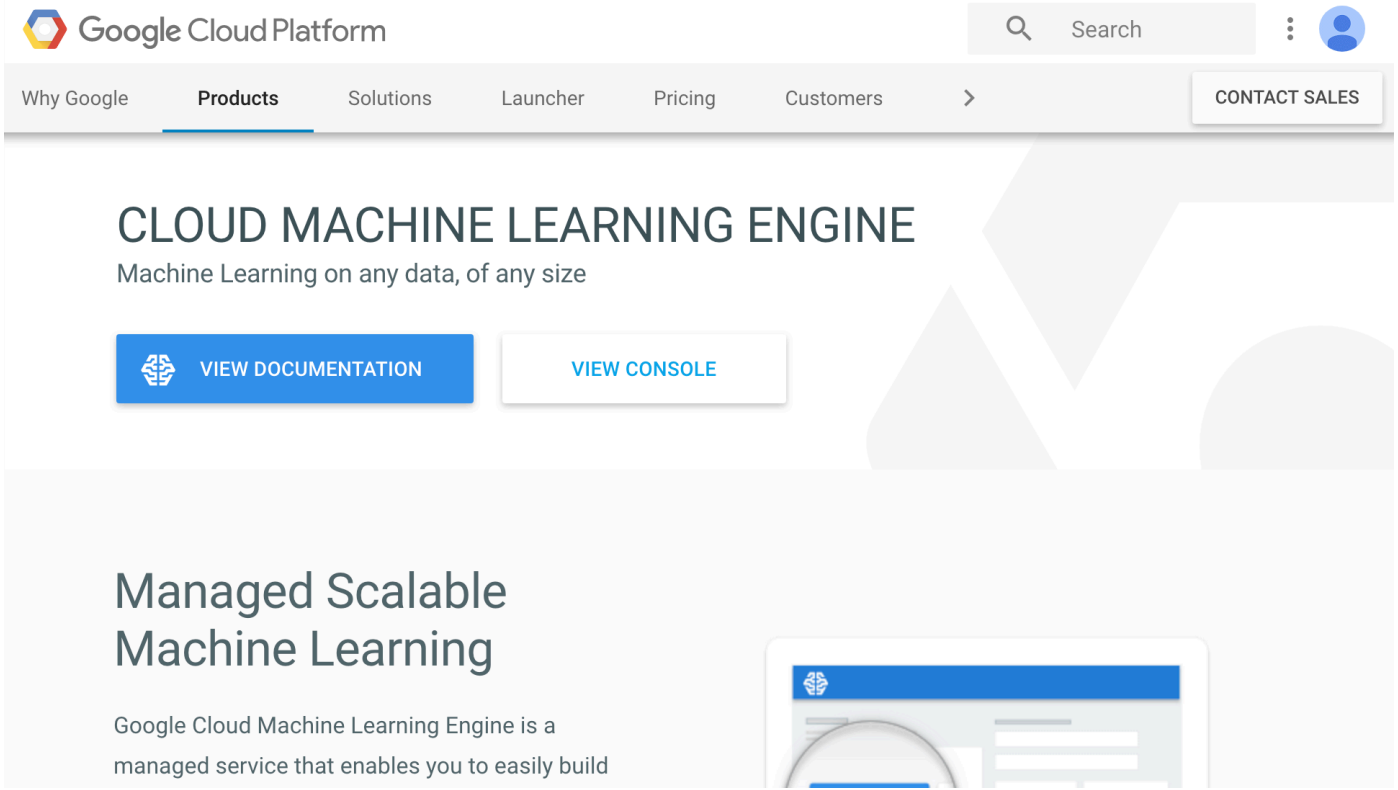


Amazon APIs

- https://www.oreilly.com/ideas/build-a-talking-face-recognizing-doorbell-for-about-100



Google Cloud ML



Tensorflow Robot

- https://www.youtube.com/watch?v=8t-wcs_JoVs

One last thing

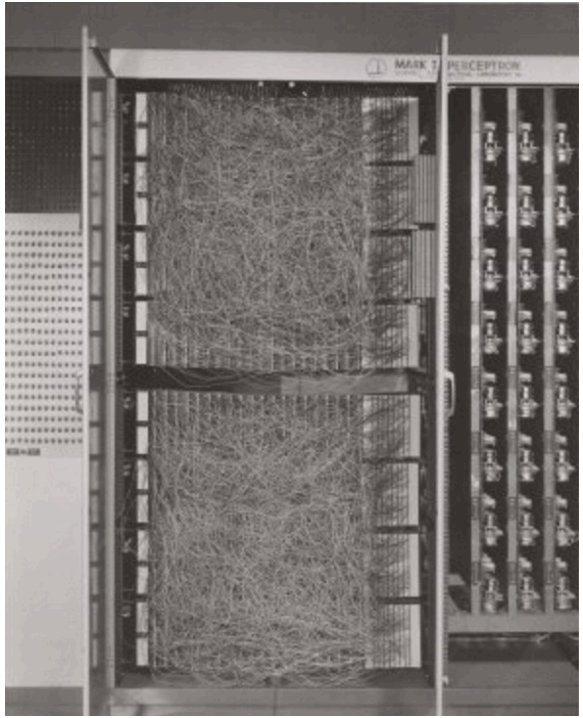
- Download http://bit.ly/model-lukas

Deep Learning!

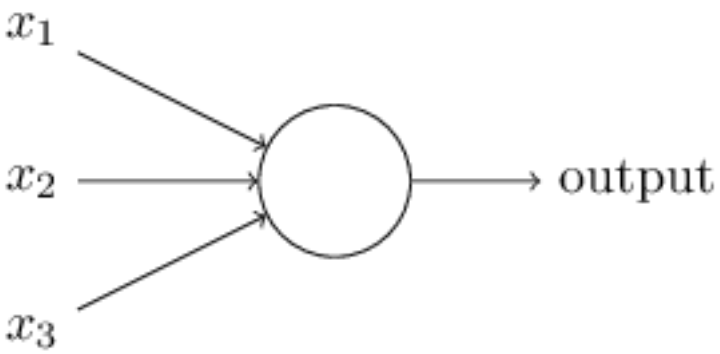
Get the downloads out of the way

```
1 pip install keras
2 python -c "from keras.datasets import mnist"
```

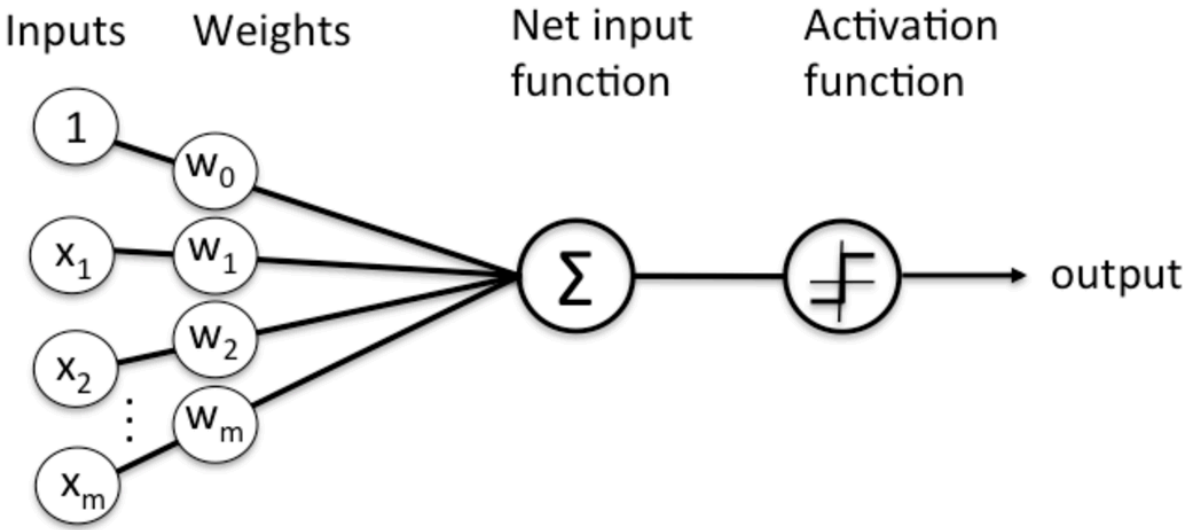
The First Perceptron



Perceptron



Perceptron



Schematic of Rosenblatt's perceptron.

Perceptron on Tweets

```
import pandas as pd
import numpy as np


df = pd.read_csv('tweets.csv')
target = df['is_there_an_emotion_directed_at_a_brand_or_product']
text = df['tweet_text']

fixed_text = text[pd.notnull(text)]
fixed_target = target[pd.notnull(text)]

from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
count_vect.fit(fixed_text)

counts = count_vect.transform(fixed_text)

from sklearn.linear_model import Perceptron

perceptron = Perceptron()

from sklearn import cross_validation

scores = cross_validation.cross_val_score(perceptron, counts, fixed_target, cv=10)
print scores
print scores.mean()
```


Digits Dataset (keras-digits.py)

```
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()

digit = X_train[0]
print(digit.shape)
str = ""
for i in range(digit.shape[0]):
    for j in range(digit.shape[1]):
        if digit[i][j] == 0:
            str += " "
        elif digit[i][j] < 128:
            str += "."
        else:
            str += "X"
    str += "\n"

print(str)
print("Label: ", y_train[0])
```


Perceptron on Digits (keras-scikit-learn.py)

```
from sklearn import datasets

digits = datasets.load_digits()

from sklearn import cross_validation
from sklearn.linear_model import Perceptron

perceptron = Perceptron()

scores = cross_validation.cross_val_score(perceptron, digits.data, digits.target, cv=10)
print scores
print scores.mean()
```

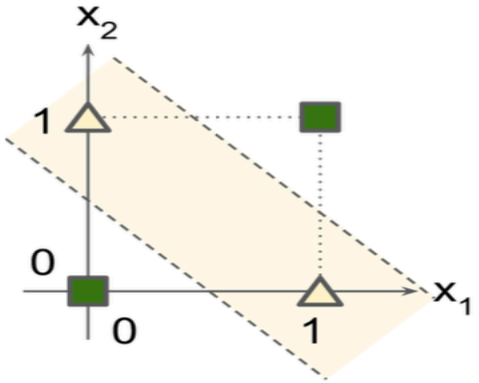

One hot encoding Keras-one-hot.py

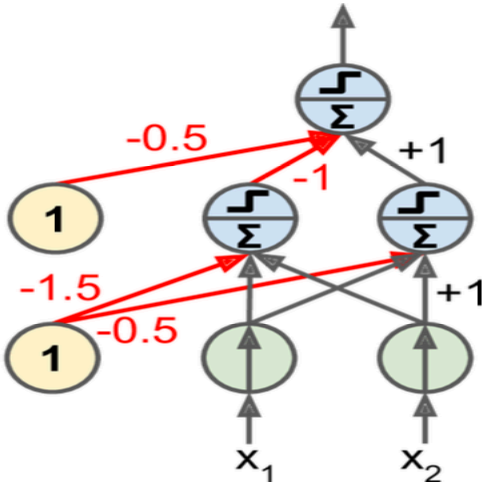
Label	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1

Keras Perceptron (keras-perceptorn)

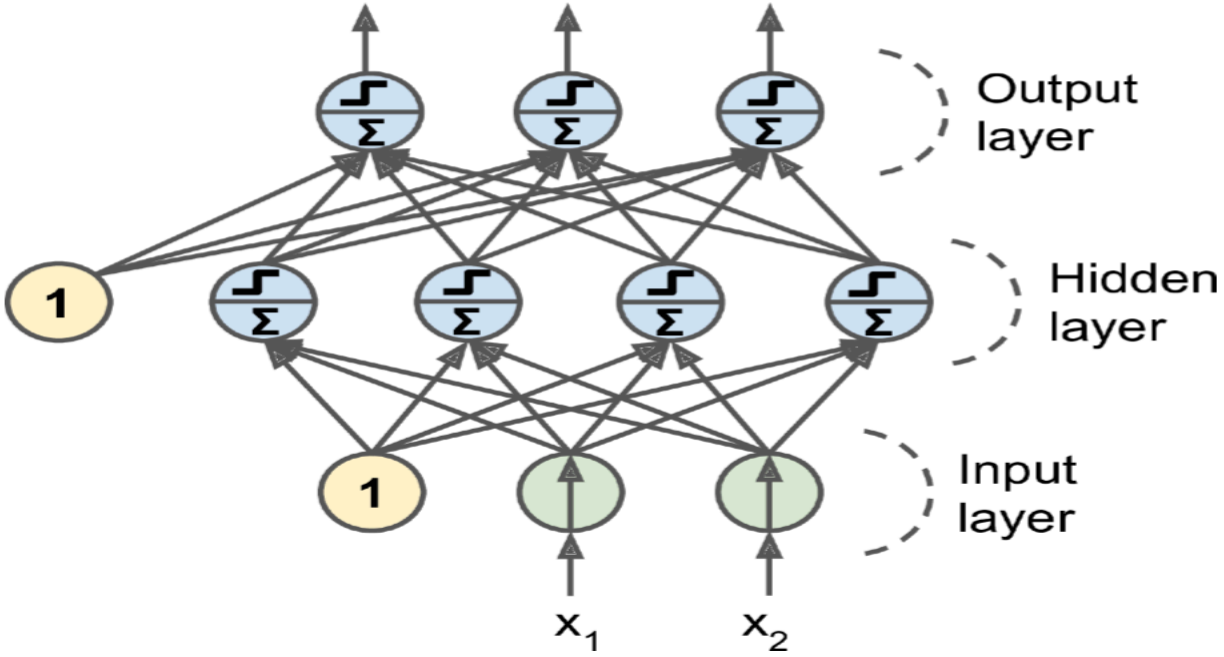
```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Flatten
from keras.layers import Dense
from keras.utils import np_utils

(X_train, y_train), (X_test, y_test) = mnist.load_data()

img_width=28
img_height=28

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# build model
model = Sequential()
model.add(Flatten(input_shape=(img_width,img_height)))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train)
```


Solving the “XOR” Problem

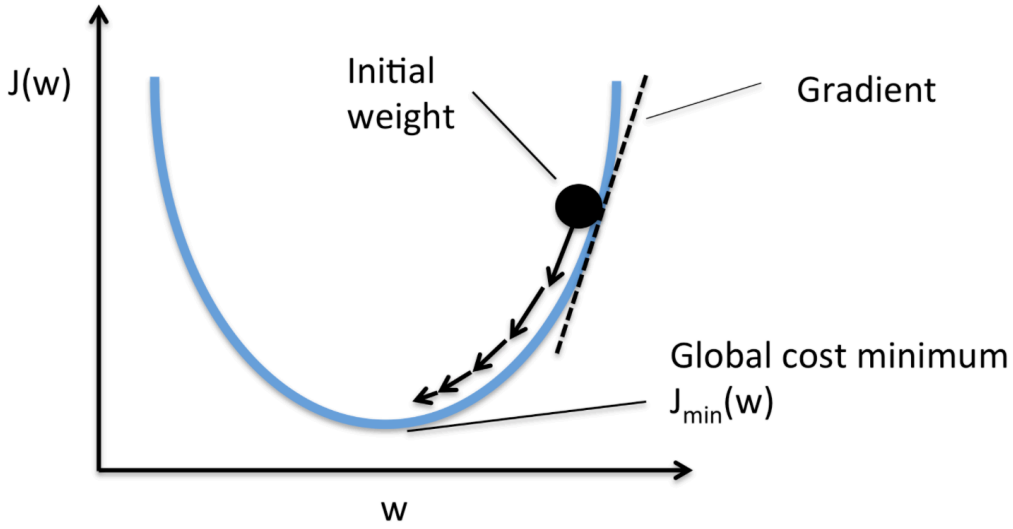




Two Layers of Perceptrons

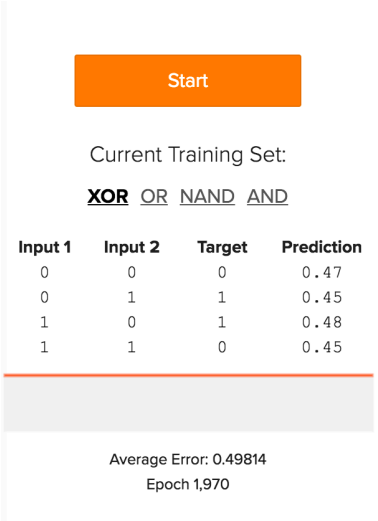


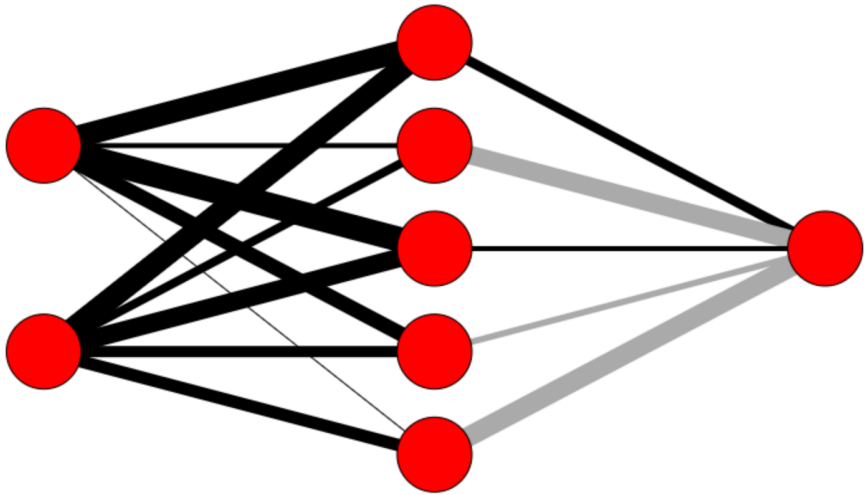
Gradient Descent



Backpropagation (1985)

http://www.emergentmind.com/neural-network





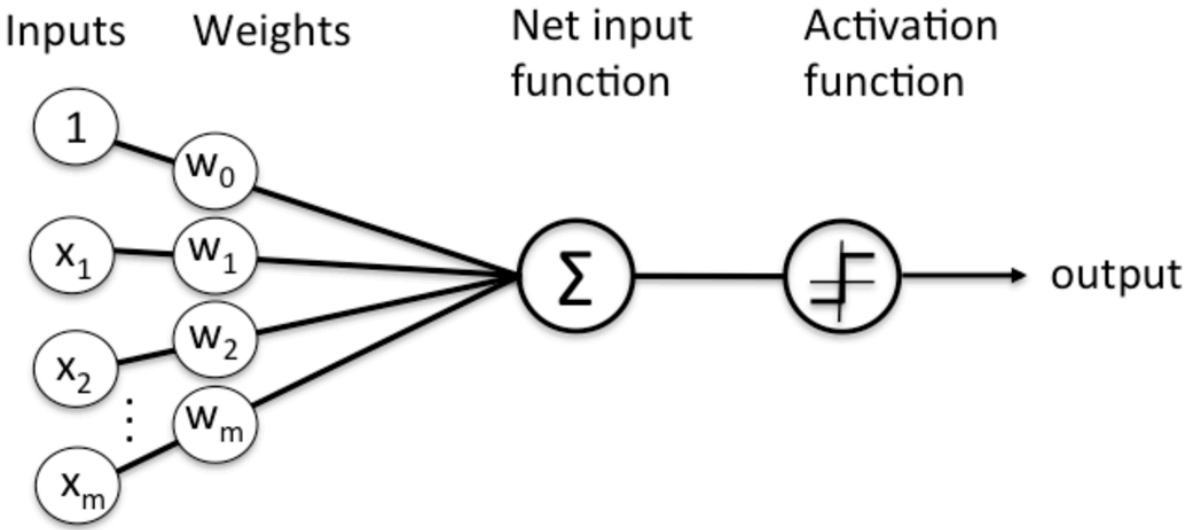
Neural Network Visualization

- http://playground.tensorflow.org/

keras two layer perceptron

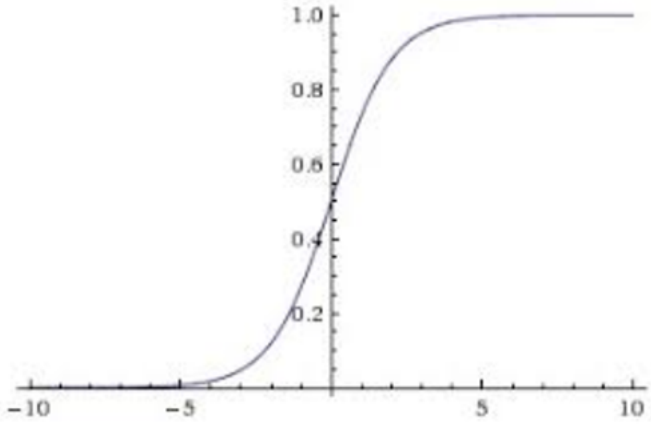
```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Flatten
from keras.layers import Dense
from keras.utils import np_utils

(X_train, y_train), (X_test, y_test) = mnist.load_data()

img_width=28
img_height=28

X_train /= 255
X_test /= 255

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# build model
model = Sequential()
model.add(Flatten(input_shape=(img_width,img_height)))
model.add(Dense(30, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train)
```


Activation Functions Revisited

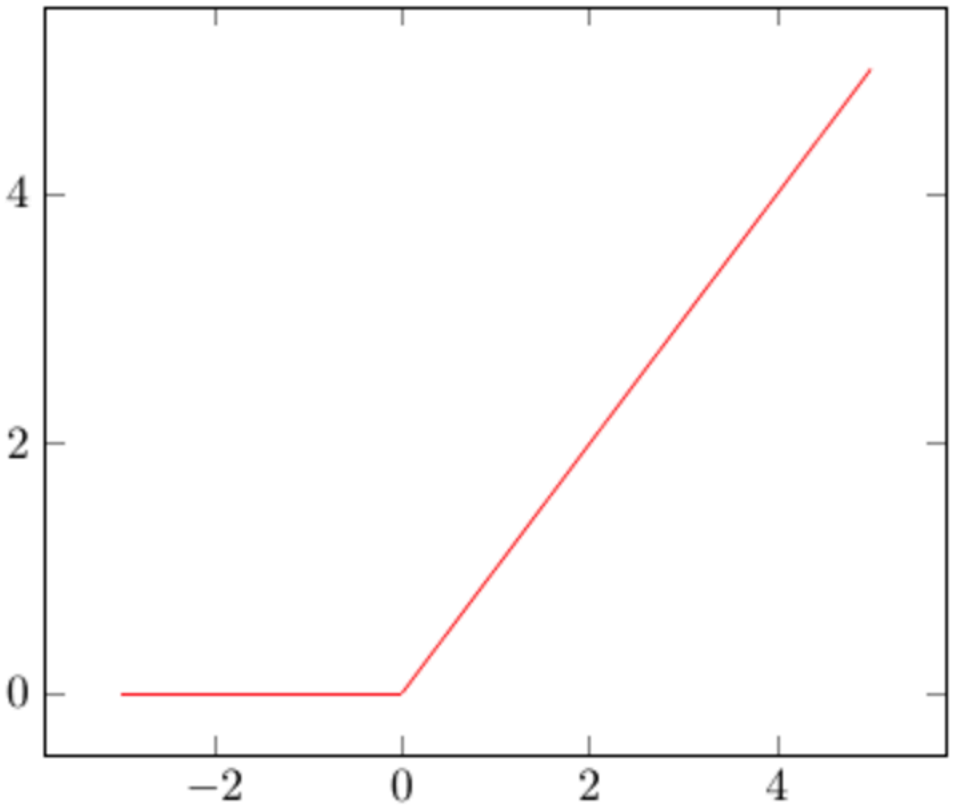


Schematic of Rosenblatt's perceptron.

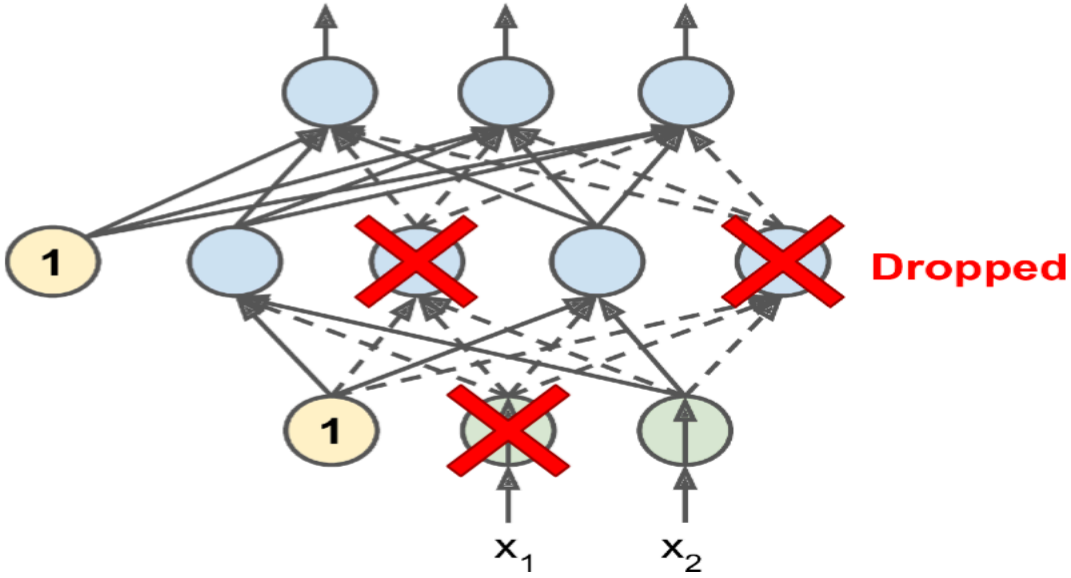
Sigmoid



ReLU

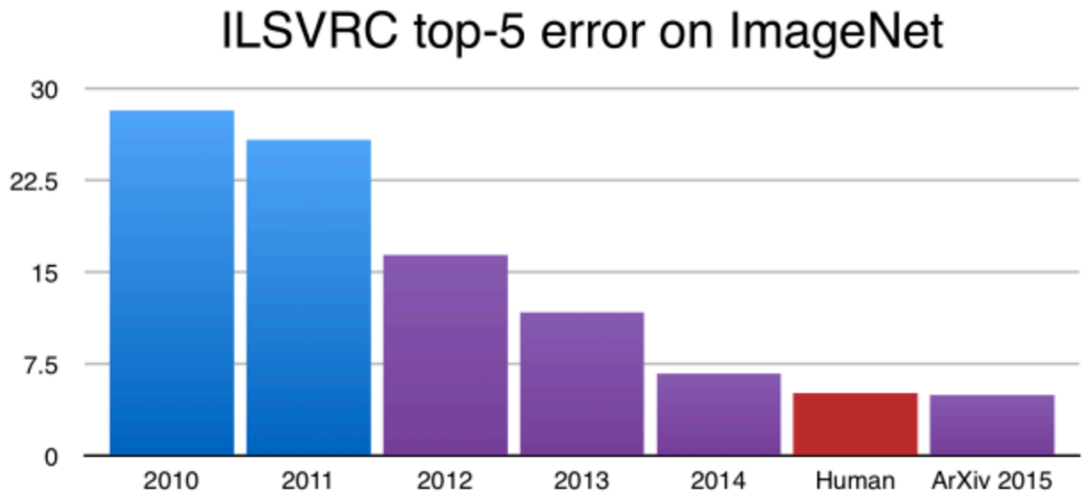


Dropout

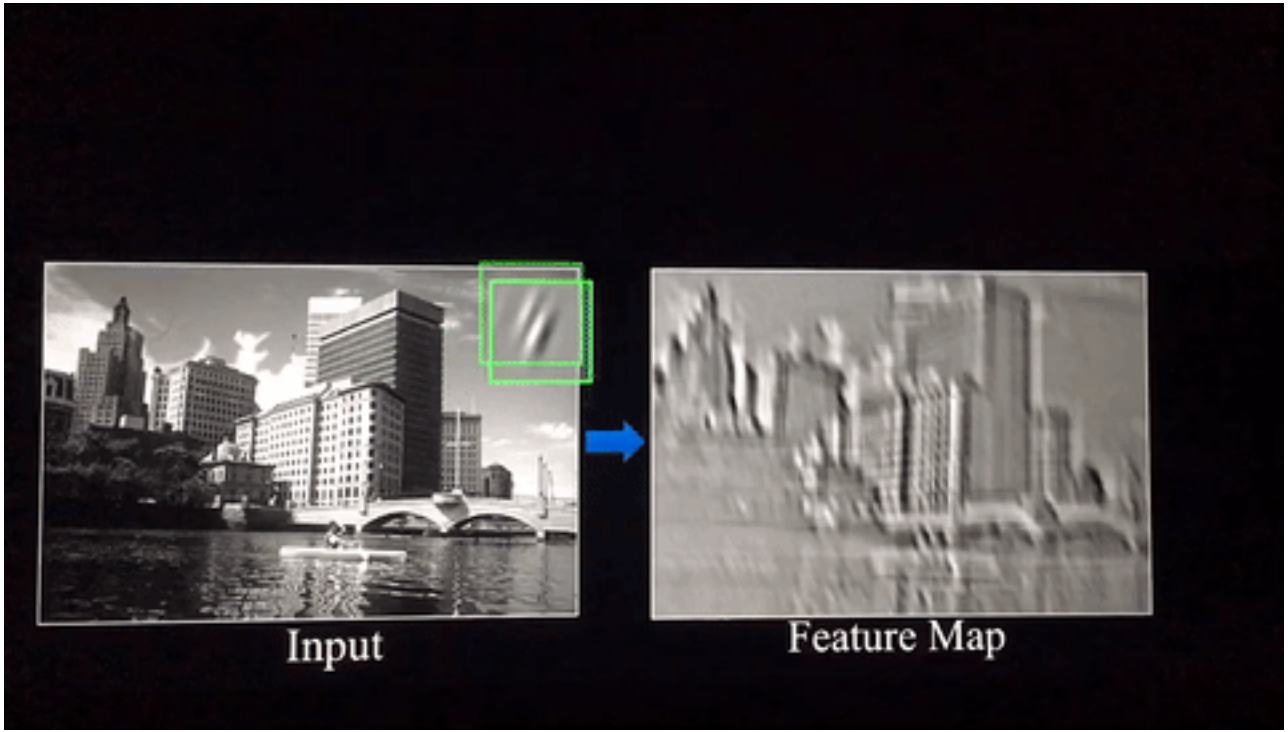


Convolutional Neural Networks

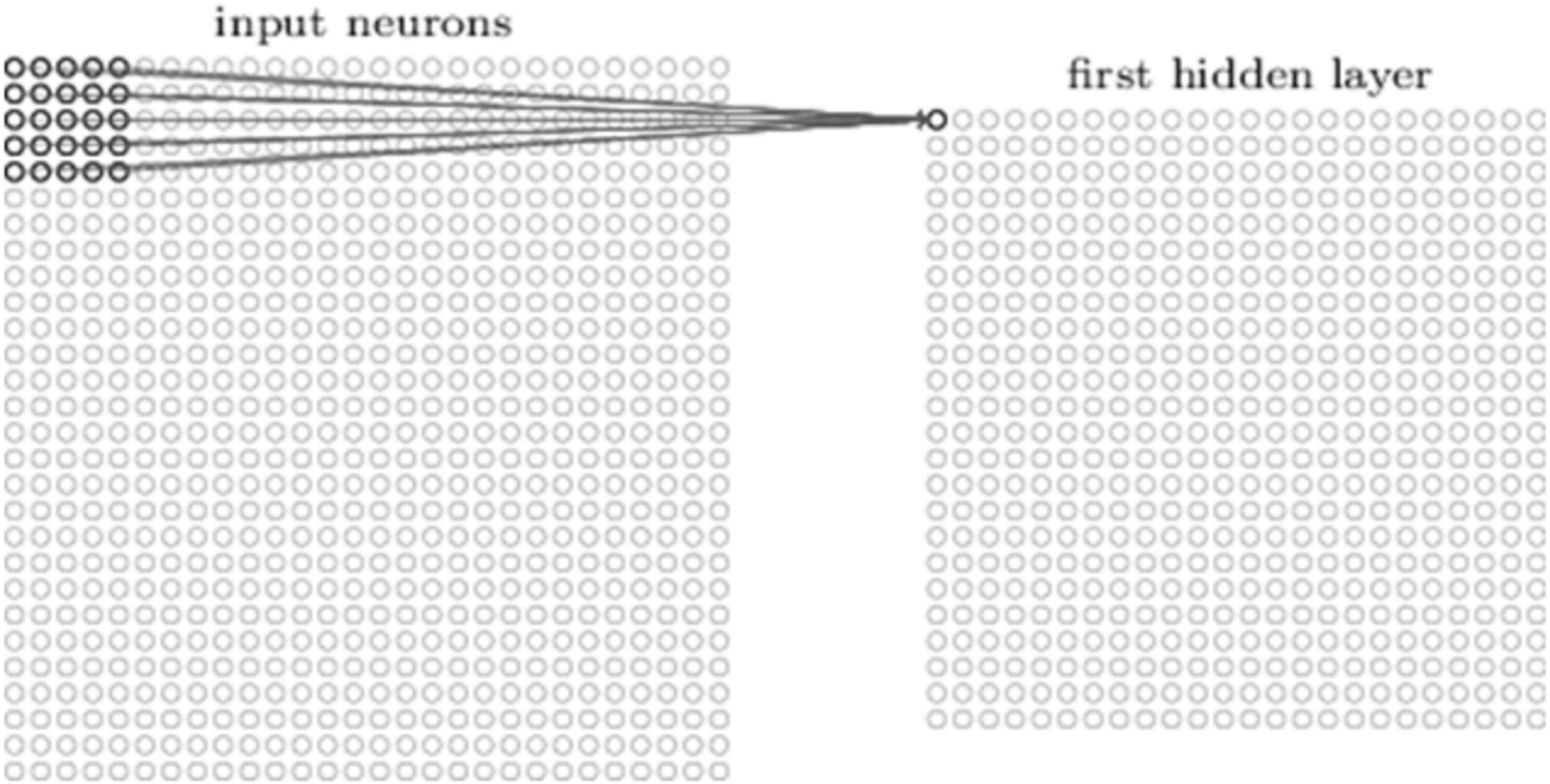
- Explanations
  https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/



Convolution

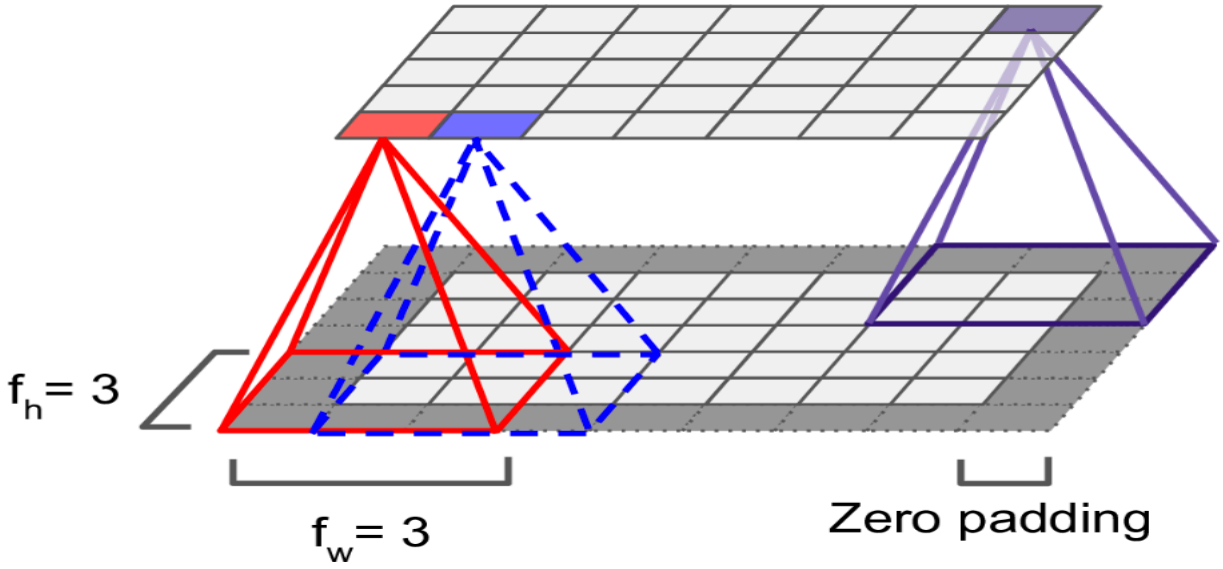


https://ujwlkarn.files.wordpress.com/2016/08/giphy.gif?w=748

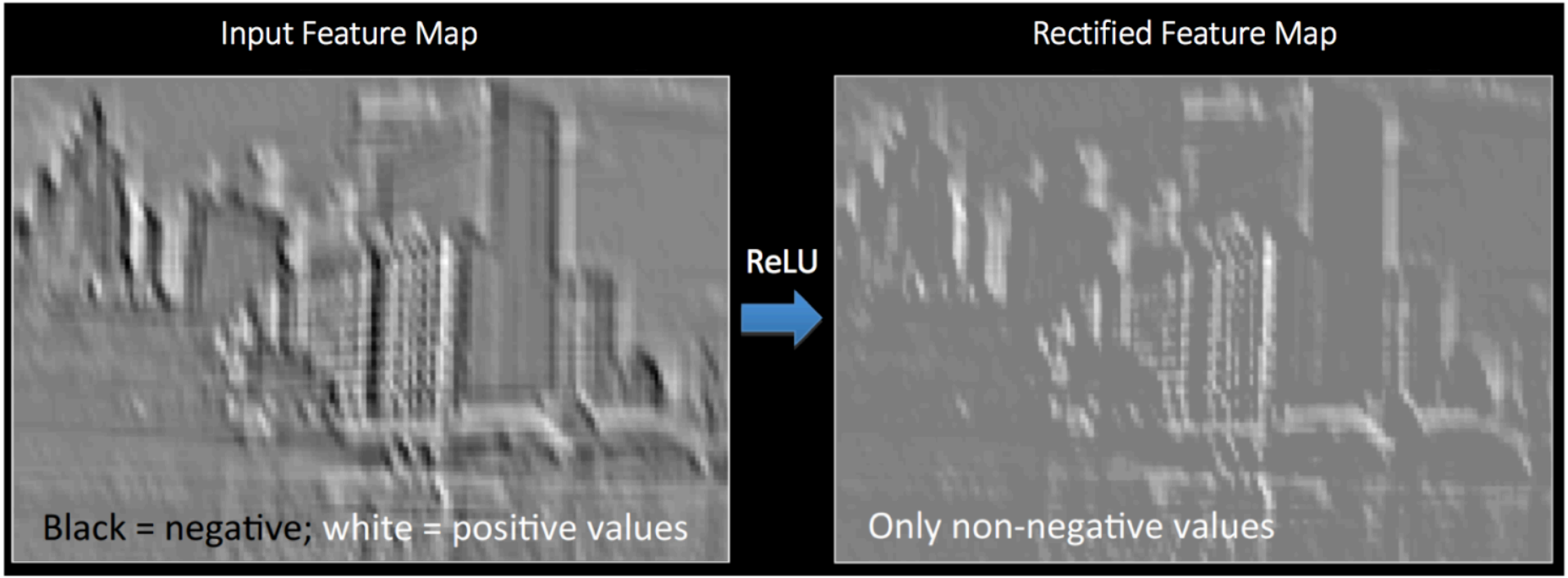


Visualization of 5 x 5 filter convolving around an input volume and producing an activation map

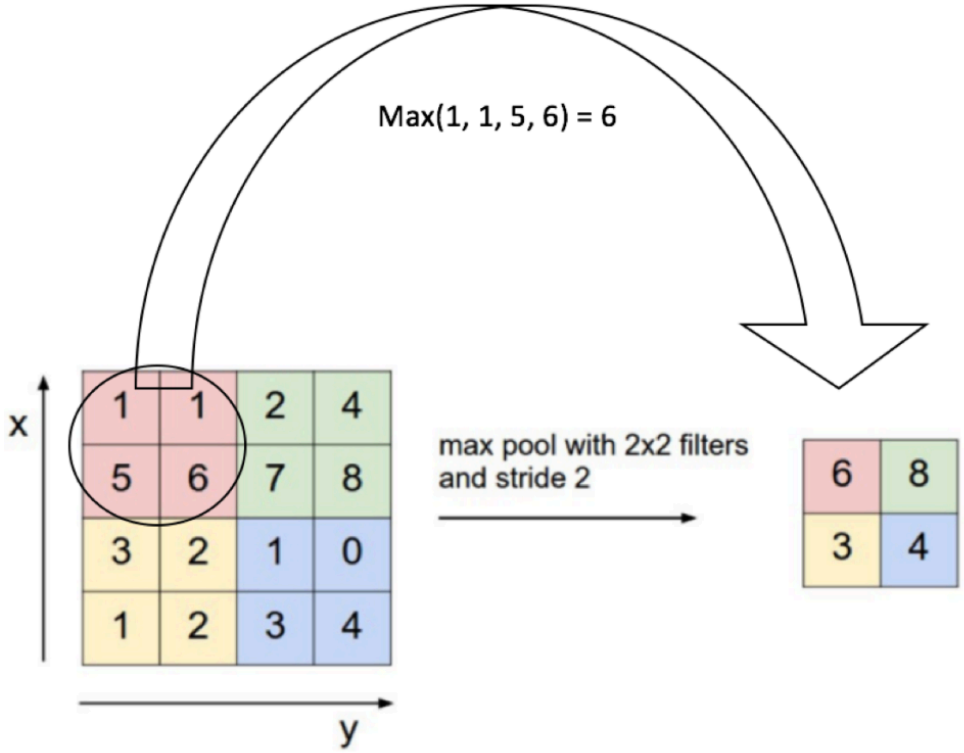
Convolutions



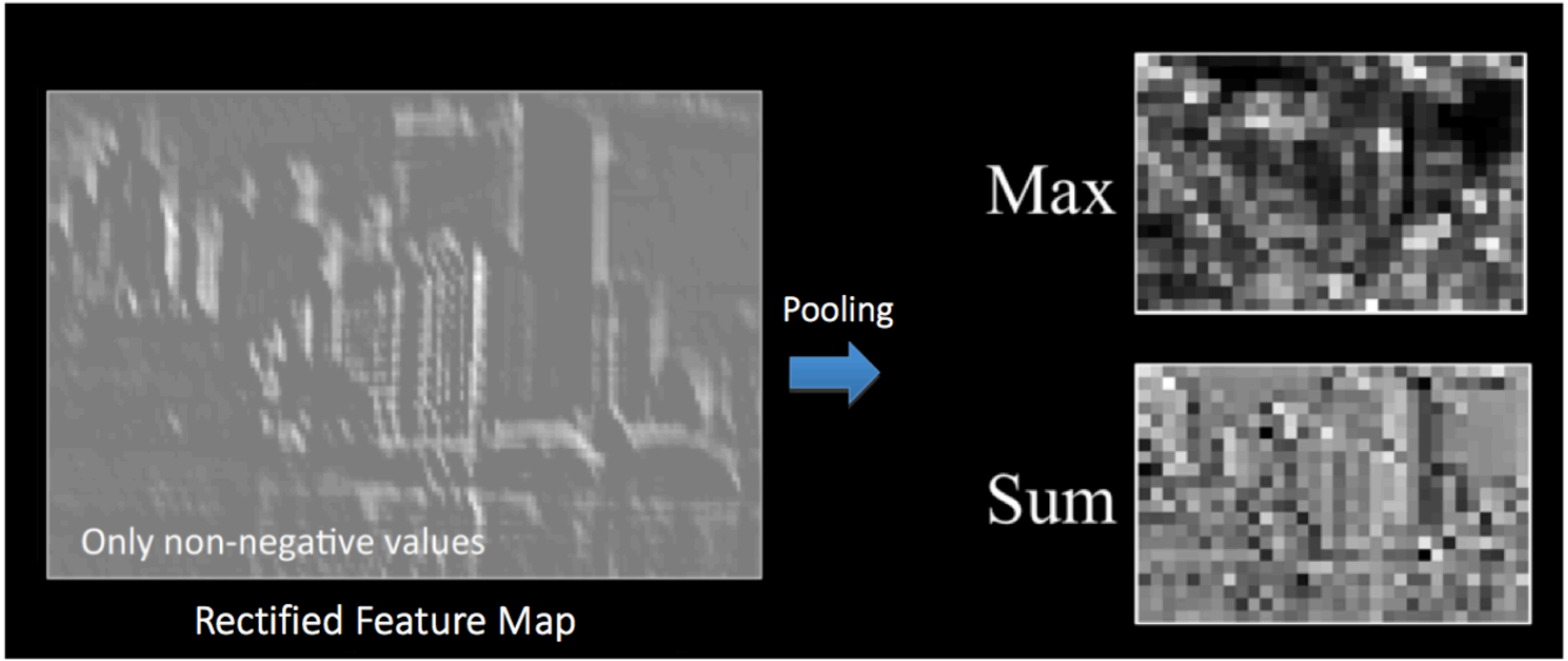
Activation Function (ReLU)



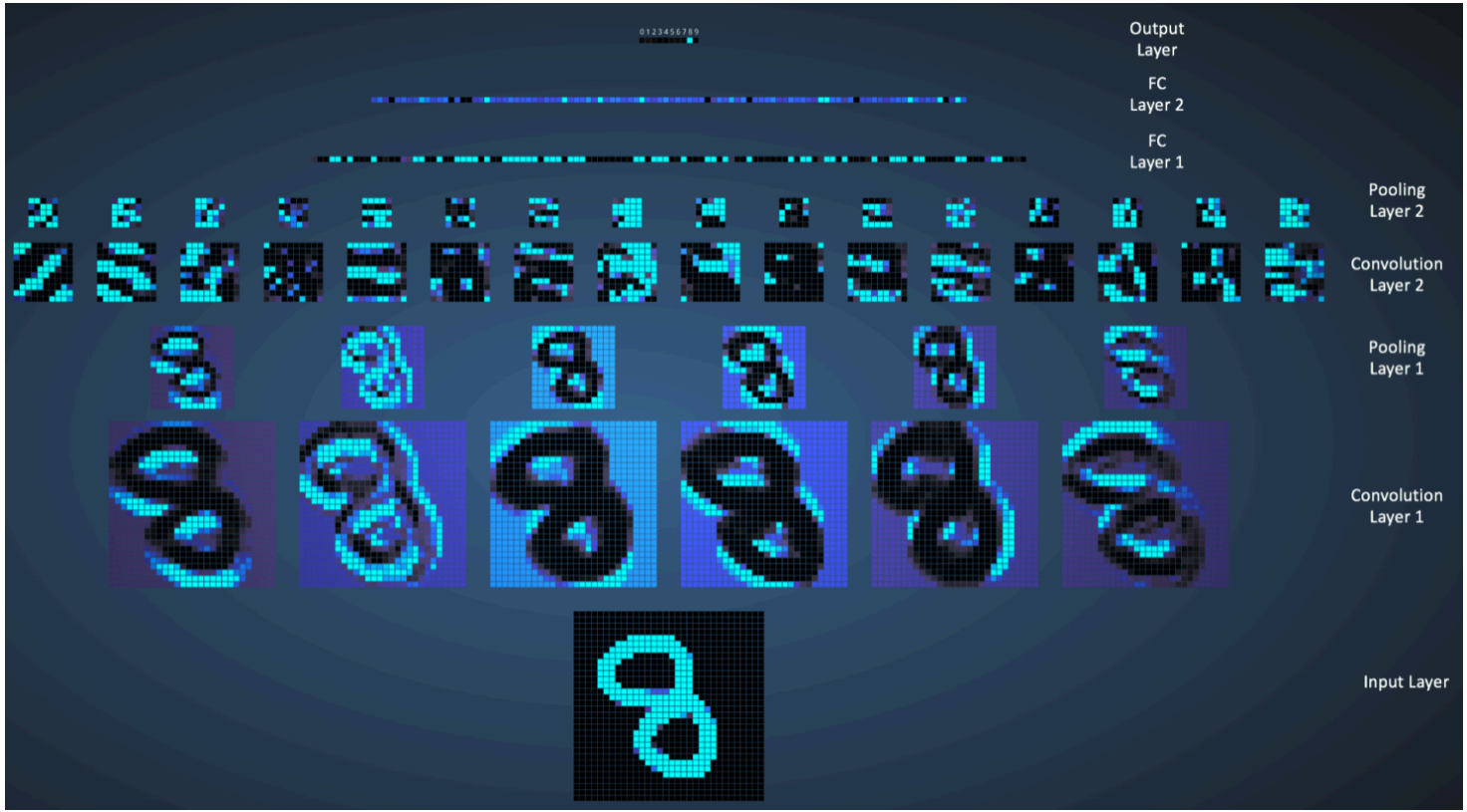
Pooling



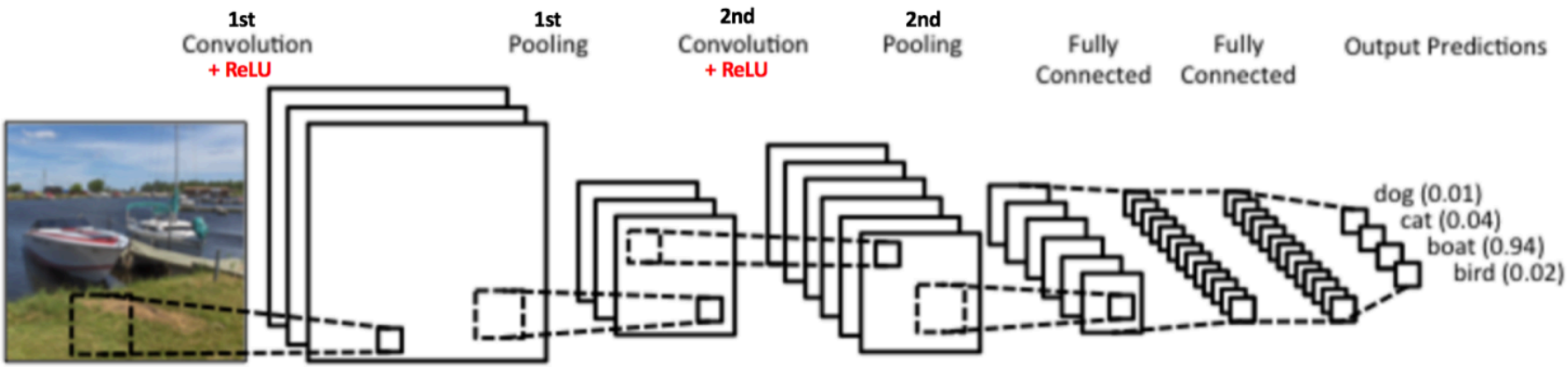
Pooling



CNN



Architecture



TensorFlow Detour! (tensorflow-mult.py)

```
from sklearn import datasets
import numpy as np
import tensorflow as tf

digits = datasets.load_digits()


x = tf.placeholder(tf.float32, [None, 64])

W = tf.Variable(tf.zeros([64, 10]))
b = tf.Variable(tf.zeros([10]))

y = tf.nn.softmax(tf.matmul(x, W) + b)

y_ = tf.placeholder(tf.float32, [None, 10])
#cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
cross_entropy = -tf.reduce_sum(y_*tf.log(tf.clip_by_value(y,1e-10,1.0)))

train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

loss = tf.reduce_mean(cross_entropy, name='xentropy_mean')

init = tf.initialize_all_variables()

sess = tf.Session()
sess.run(init)


dense_target = np.zeros([digits.target.size, 10])
for i in range(digits.target.size):
    dense_target[i, digits.target[i]] = 1

for step in range(10):
    batch_xs = digits.data
```


TensorFlow Perceptron Implementation (tensorflow-perceptron.py)

```
from sklearn import datasets
import numpy as np
import tensorflow as tf

digits = datasets.load_digits()


x = tf.placeholder(tf.float32, [None, 64])

W = tf.Variable(tf.zeros([64, 10]))
b = tf.Variable(tf.zeros([10]))

y = tf.nn.softmax(tf.matmul(x, W) + b)

y_ = tf.placeholder(tf.float32, [None, 10])
#cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_indices=[1]))
cross_entropy = -tf.reduce_sum(y_*tf.log(tf.clip_by_value(y,1e-10,1.0)))

train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

loss = tf.reduce_mean(cross_entropy, name='xentropy_mean')

init = tf.initialize_all_variables()

sess = tf.Session()
sess.run(init)


dense_target = np.zeros([digits.target.size, 10])
for i in range(digits.target.size):
    dense_target[i, digits.target[i]] = 1

for step in range(10):
    batch_xs = digits.data
```


Full CNN Tensorflow (tensorflow-cnn.py)

```
from sklearn import datasets
import numpy as np
import tensorflow as tf

sess = tf.InteractiveSession()

digits = datasets.load_digits()

def weight_variable(shape):
  initial = tf.truncated_normal(shape, stddev=0.1)
  return tf.Variable(initial)

def bias_variable(shape):
  initial = tf.constant(0.1, shape=shape)
  return tf.Variable(initial)

def conv2d(x, W):
  return tf.nn.conv2d(x, W, strides=[1, 1, 1, 1], padding='SAME')

def max_pool_2x2(x):
  return tf.nn.max_pool(x, ksize=[1, 2, 2, 1],
                        strides=[1, 2, 2, 1], padding='SAME')

W_conv1 = weight_variable([5, 5, 1, 32])  # 5x5 grid
b_conv1 = bias_variable([32])

x = tf.placeholder(tf.float32, shape=[None, 64])
y_ = tf.placeholder(tf.float32, shape=[None, 10])

x_image = tf.reshape(x, [-1,8,8,1]) # fit data into a tensor
```


Keras cnn (keras-cnn.py)

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Flatten
from keras.layers import Dense
from keras.utils import np_utils

(X_train, y_train), (X_test, y_test) = mnist.load_data()

img_width=28
img_height=28

X_train /= 255
X_test /= 255

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# build model
model = Sequential()
model.add(Flatten(input_shape=(img_width,img_height)))
model.add(Dense(30, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train)
```

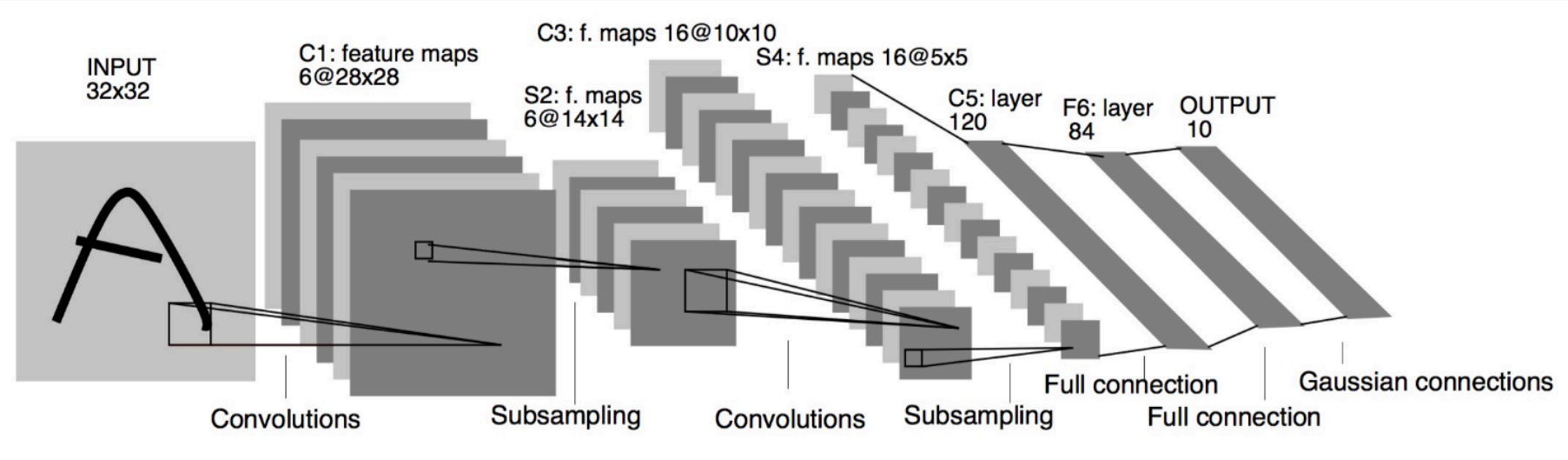

Inspect our model (keras-cnn-inspect.py)

```
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Flatten
from keras.layers import Dense
from keras.utils import np_utils

(X_train, y_train), (X_test, y_test) = mnist.load_data()

img_width=28
img_height=28

X_train /= 255
X_test /= 255

# one hot encode outputs
y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

# build model
model = Sequential()
model.add(Flatten(input_shape=(img_width,img_height)))
model.add(Dense(30, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train)
```

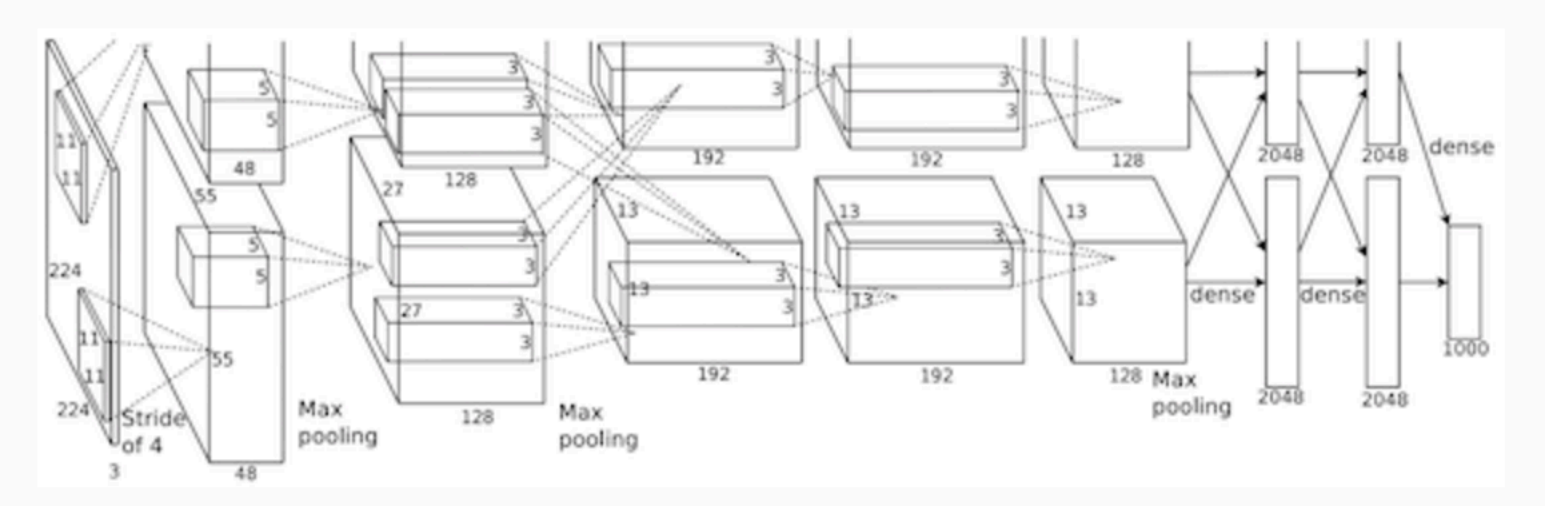

Image Models – Quick History

- https://culurciello.github.io/tech/2016/06/04/nets.html

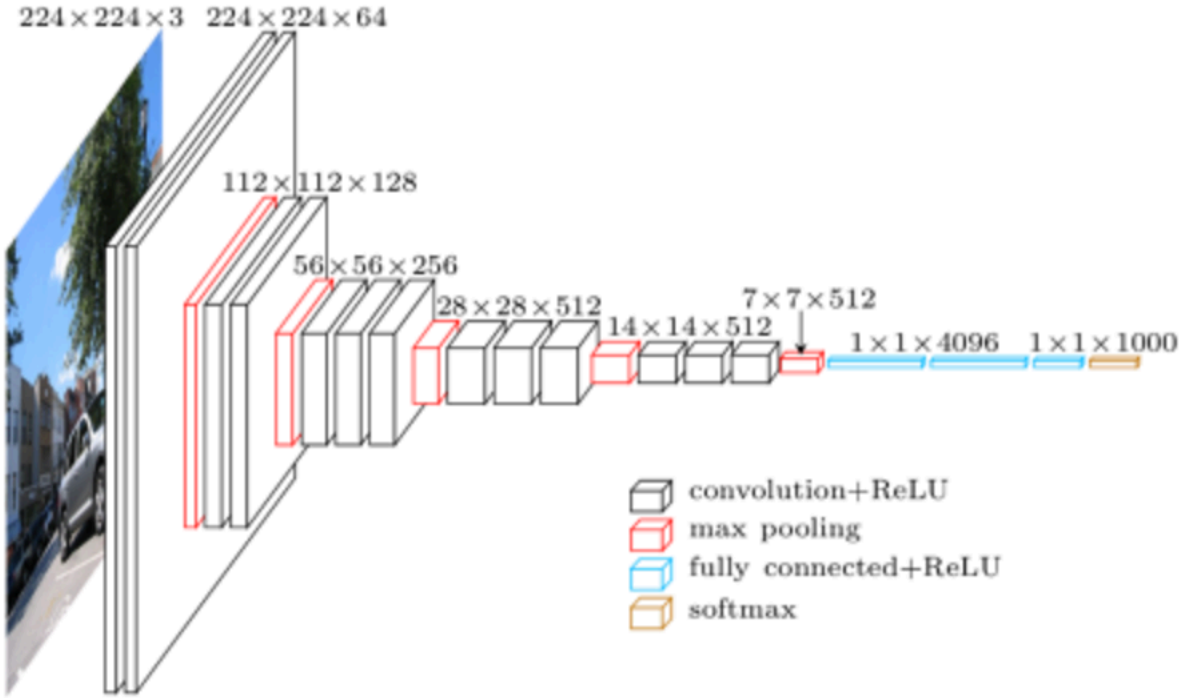
1994 LeNet



AlexNet

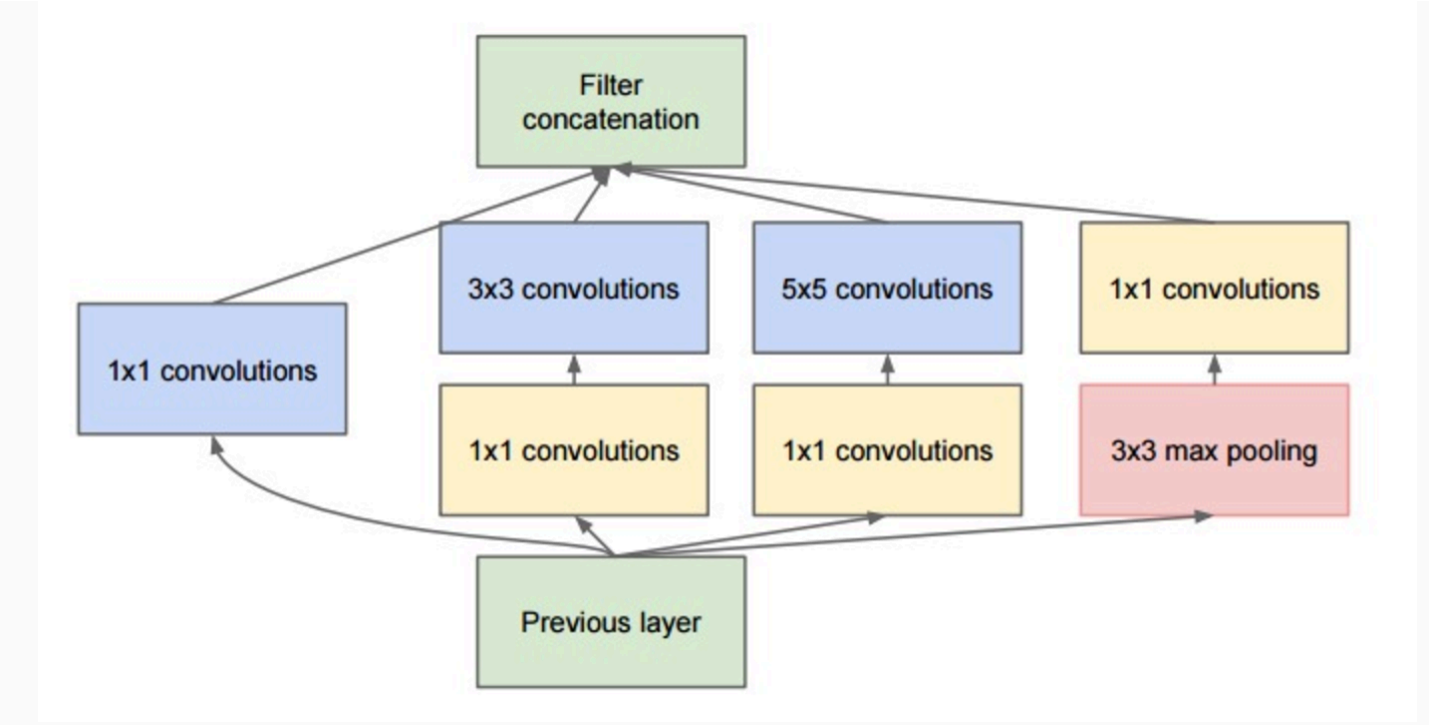


VGG architecture (2014)

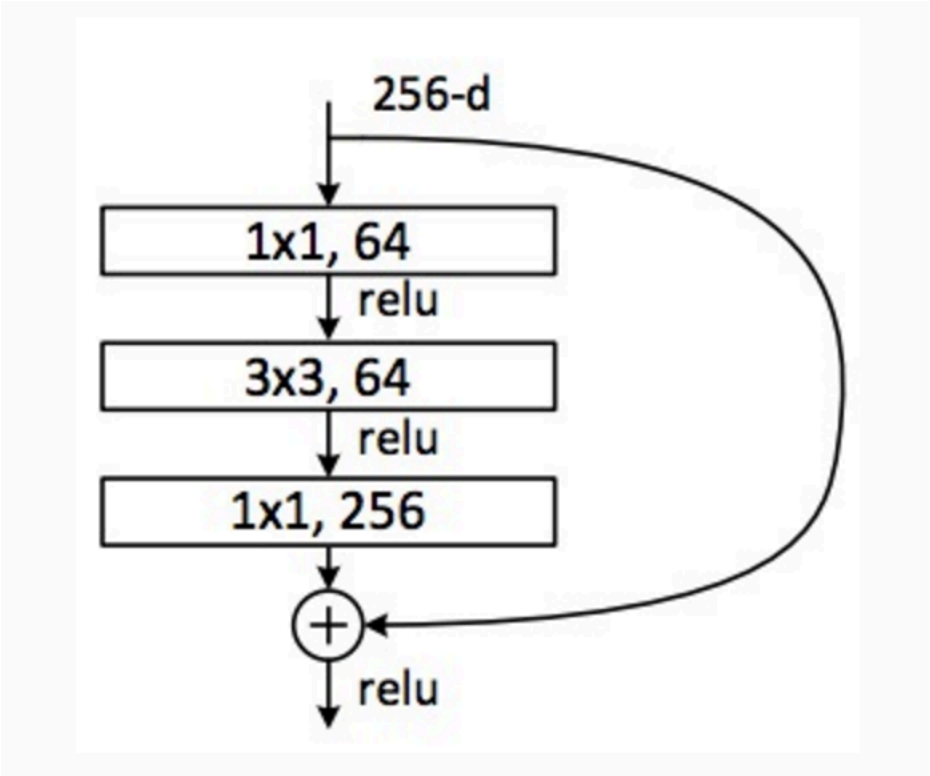


http://blog.christianperone.com/2016/01/convolutional-hypercolumns-in-python/

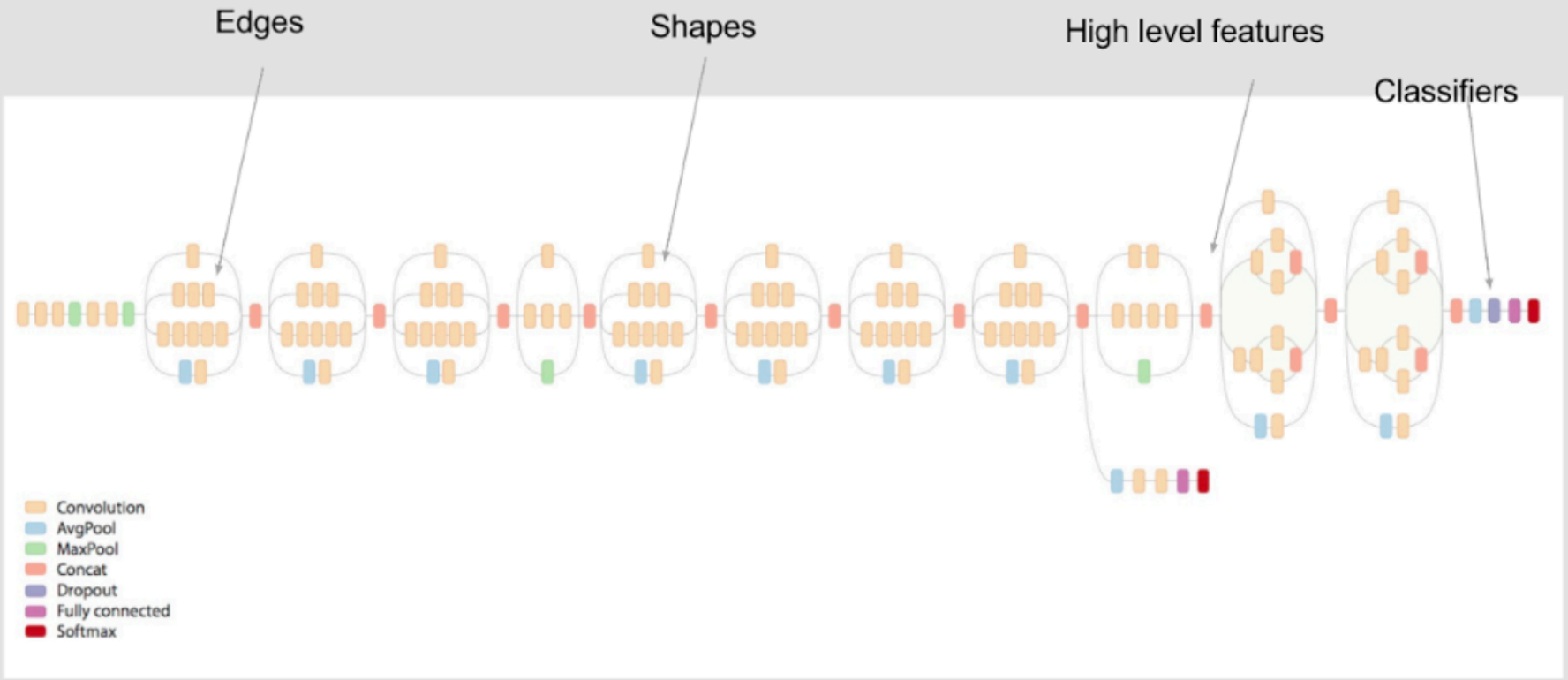
Google Inception (2014,15,16,17)



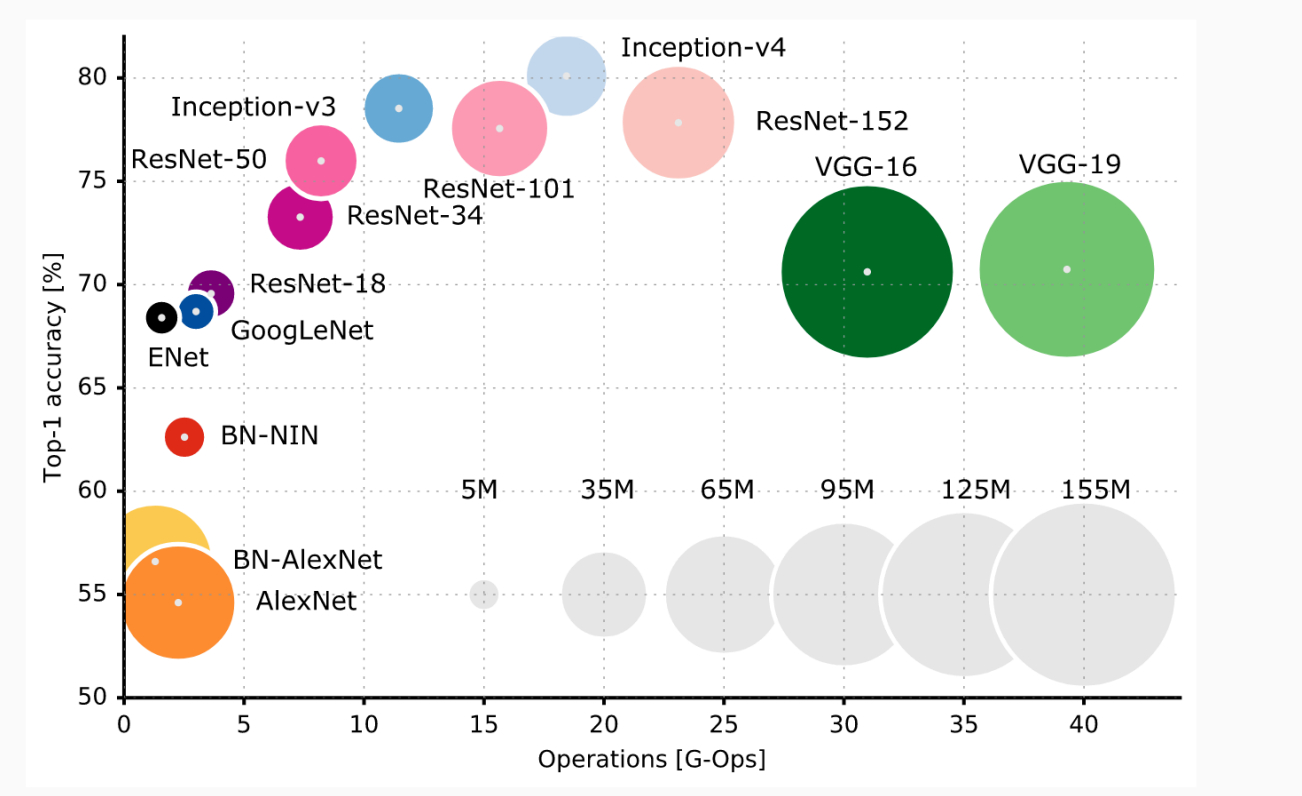
Resnet



What does the layers learn?



Overview of Neural Networks



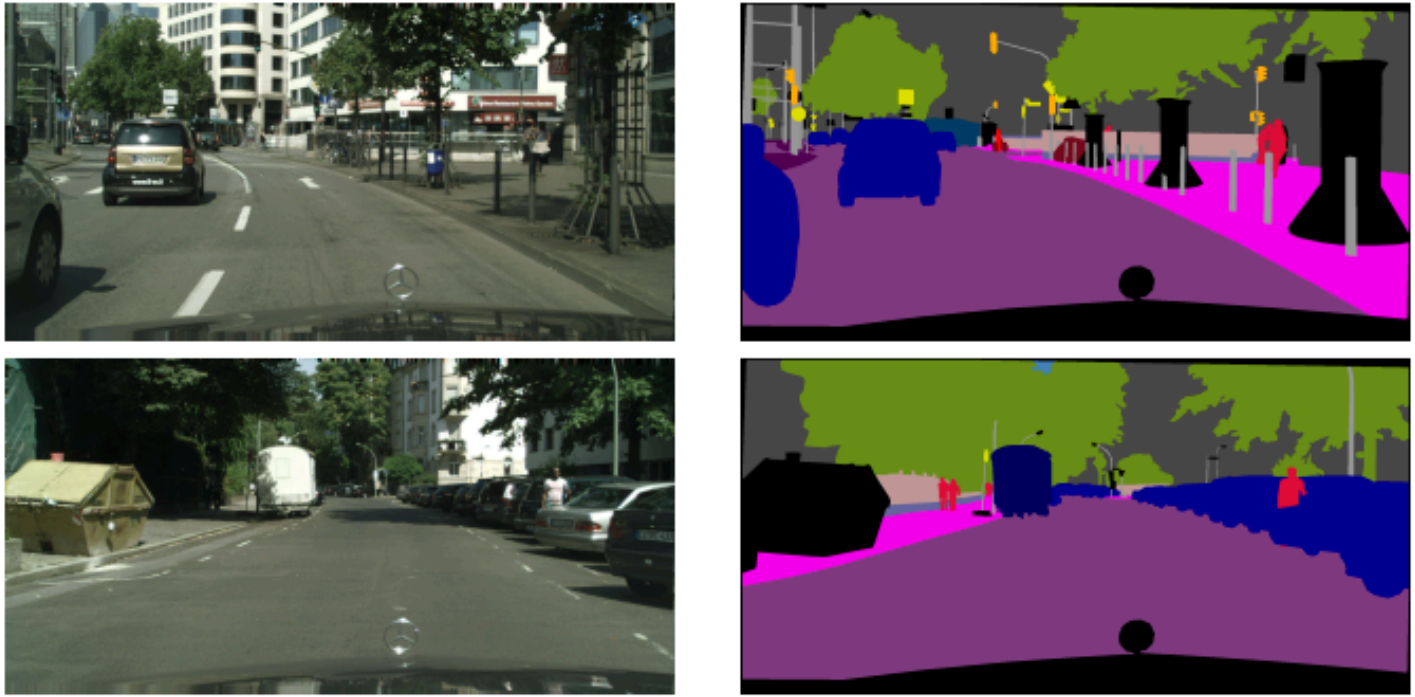
Transfer Learning

Code https://github.com/fchollet/deep-learning-models

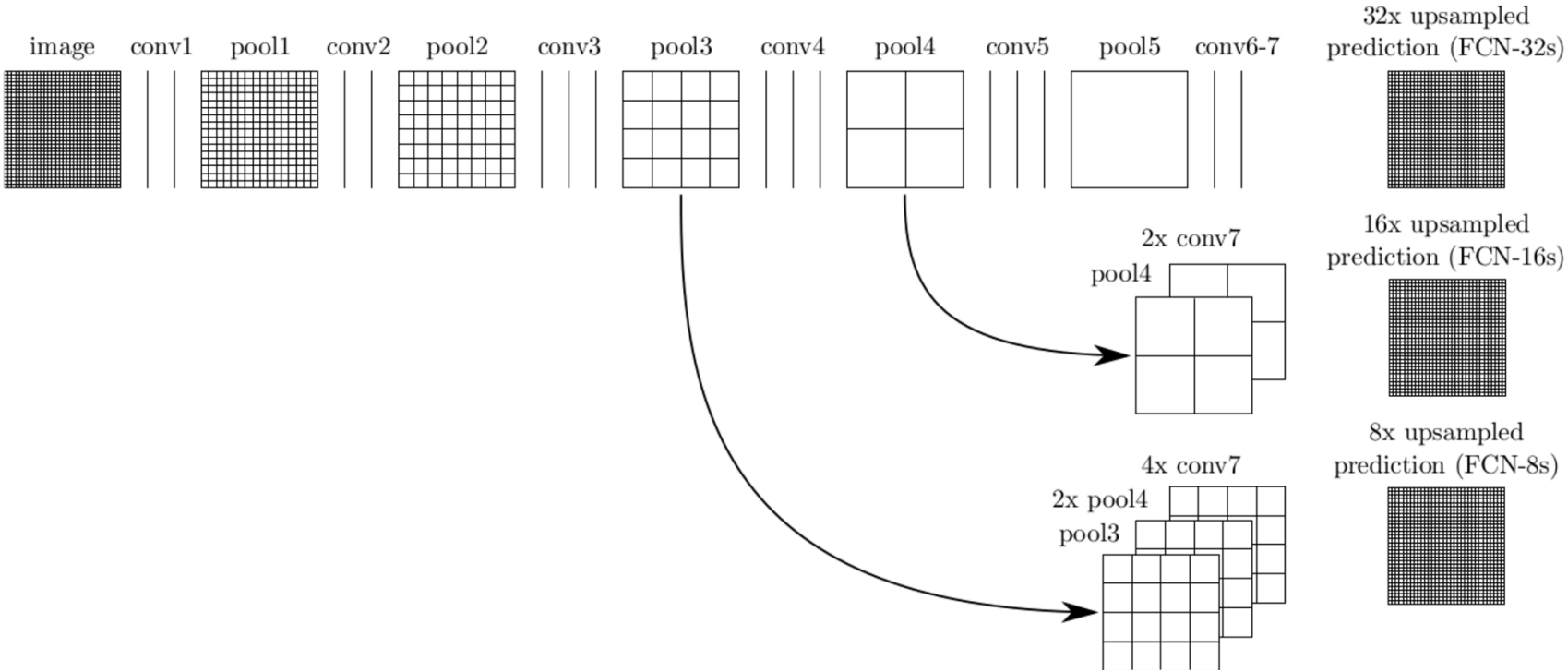
Overview http://sebastianruder.com/transfer-learning/

Architecture of Google's inception network

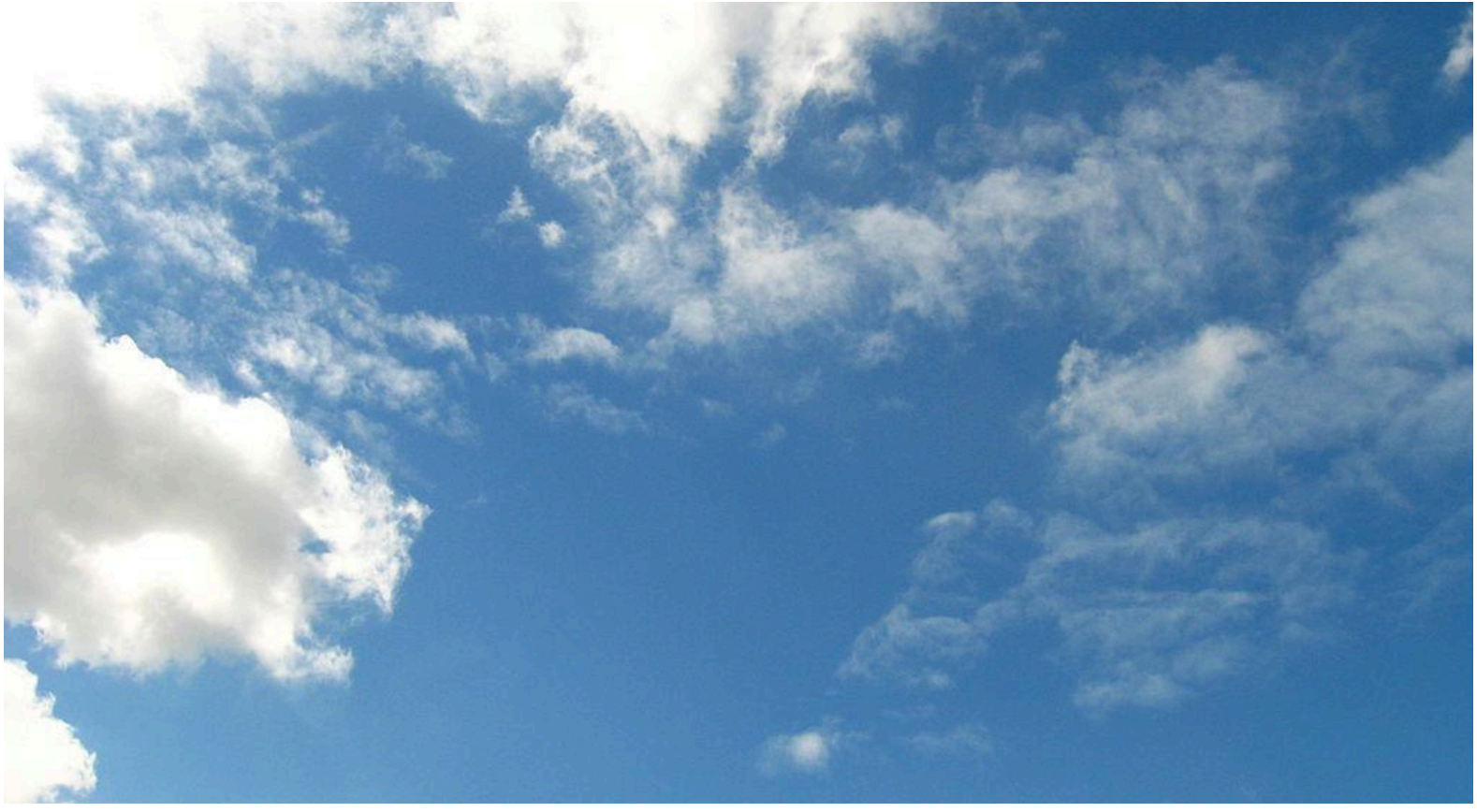
Using Models for Other Applications



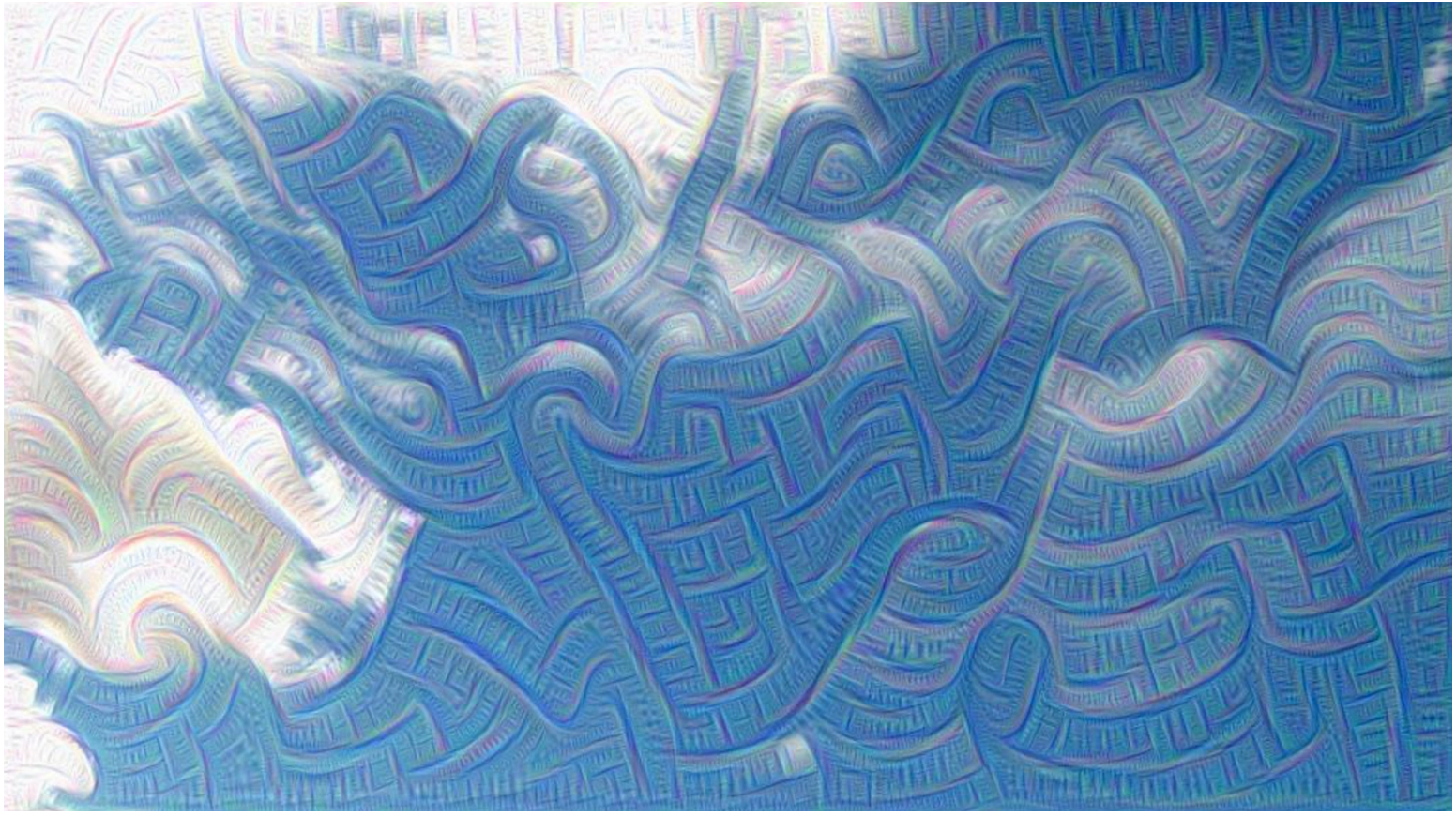
FCN Architecture



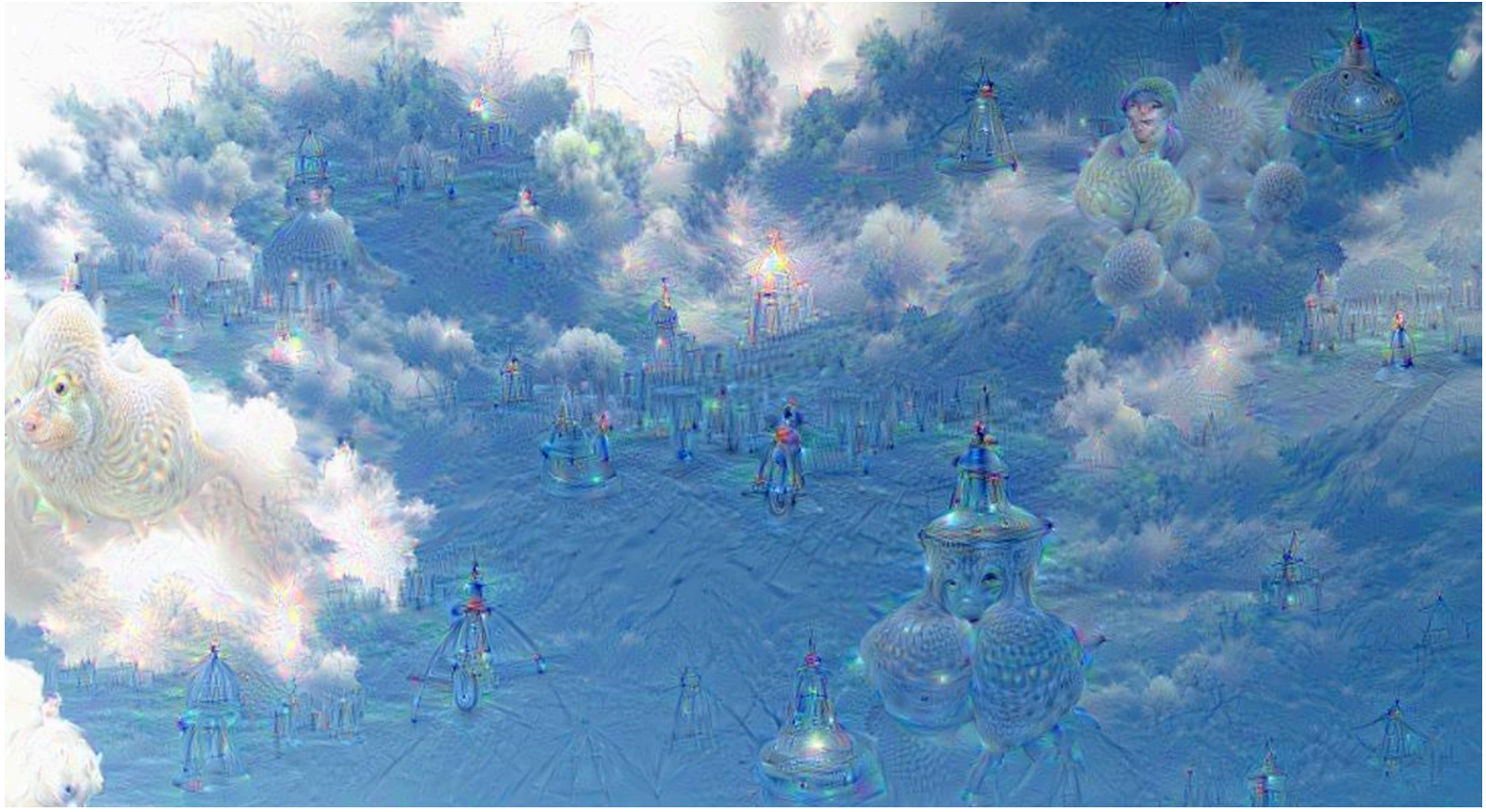
Deep Dream



Maximize Layer 3

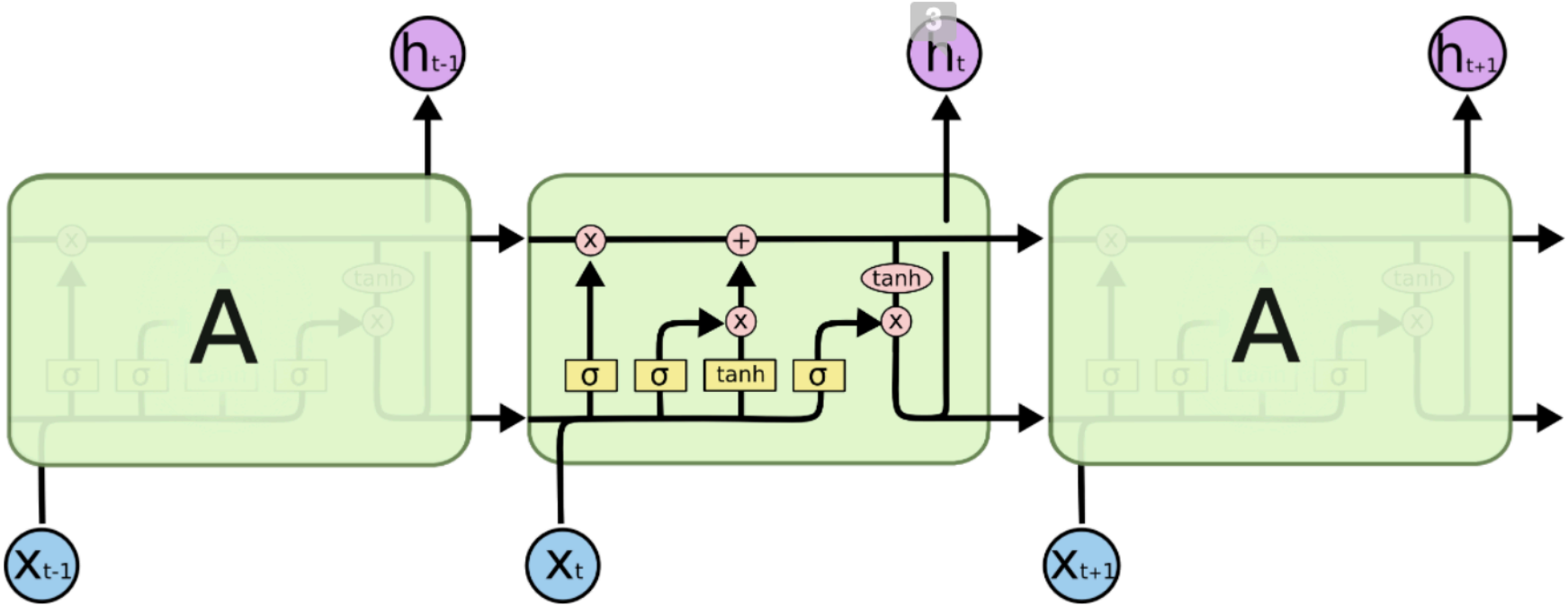


Maximize L2 Norm of Layer 4



Text, Video, other applications

- http://colah.github.io/posts/2015-08-Understanding-LSTMs/



Machine Learning

Enroll Now
Starts May 01

Financial Aid is available for learners who cannot afford the fee. Learn more and apply.

Machine Learning

About this course: Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many

More

Created by: Stanford University

Stanford University

Taught by: Andrew Ng, Associate Professor, Stanford University; Chief Scientist, Baidu; Chairman and Co-founder, Coursera

Your Home for Data Science

Kaggle helps you learn, work, and play

Create an account or Host a competition

OpenAI Gym

Deep Learning and Text

http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Data Science beta

Top Questions

active hot week month

votes	answer	views			
2	1	226	List of NLP challenges	nlp, reference-request	modified 43 mins ago rajesh 21
7	2	4k	RNN vs CNN at a high level	machine-learning, neural-network, beginner	answered 1 hour ago Biranchi 101
26	8	5k	Tools and protocol for reproducible data science using Python	python, tools, version-control	answered 2 hours ago asampat3090 16
0	2	14	Geocoding > 20 million addresses in Python	python, nlp, pandas, geospatial	answered 2 hours ago K3---rnc 685
0	1	131	Assign words to various topics	machine-learning, nlp, word-embeddings, word2vec	modified 2 hours ago Community ♦ 1

BLOG

Podcast #106: Data Team Assemble!

FEATURED ON META

Brief outage planned for Wed, May 3, 2017 at 0:00 UTC, 8pm US/Eastern (like a…

Favorite Tags edit

Add a favorite tag

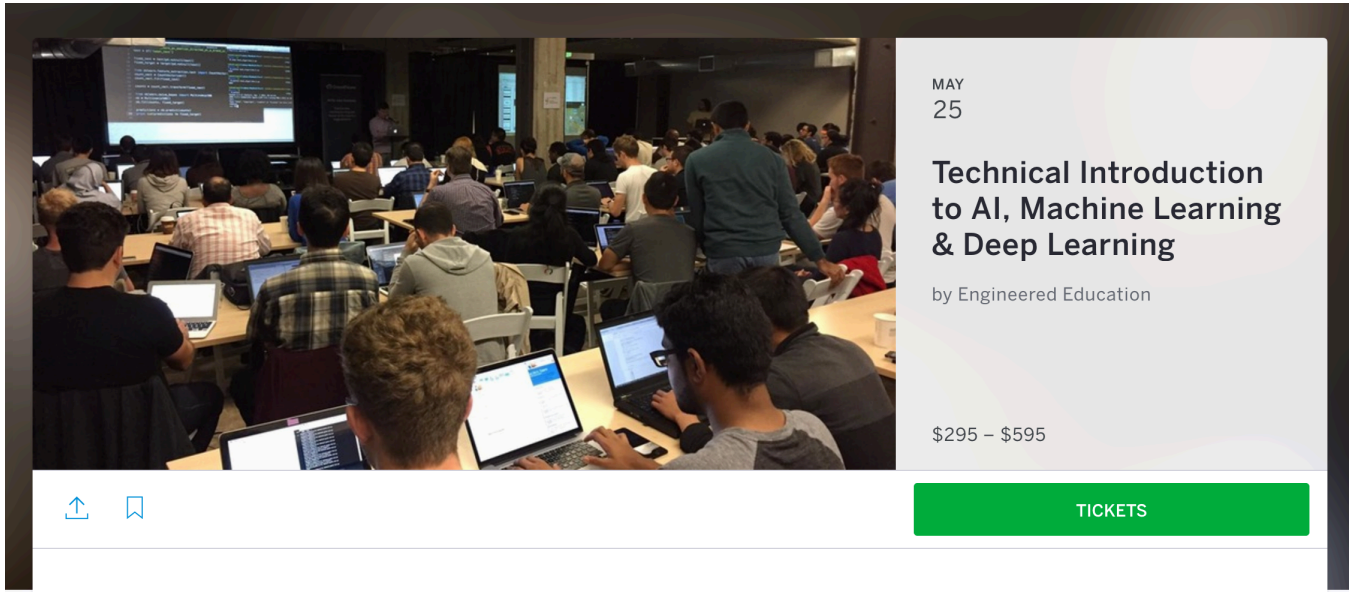
Site Stats

5,382	questions
6,649	answers
71%	answered
23,703	users
6,177	visitors/day

lukas.show/feedback

Technical Introduction to AI

July 28th



Go Deeper With Deep Learning

Jul 29-30