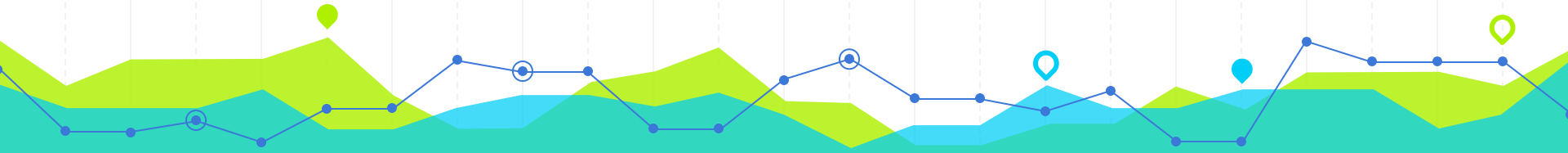


# Deep Learning

## for Engineers

OUTTHINK


Google

landscape

10. Juli 2015





18. Juni 2014


17. Juni 2014

16. Juni 2014

# AlphaGo

THE FIRST COMPUTER PROGRAM TO EVER BEAT A PROFESSIONAL PLAYER AT THE GAME OF GO.

Home Moments Notifications Messages Search Twitter Tweet




TWEETS 7,140 FOLLOWERS 2,281 Follow

**TayTweets**
@TayandYou

The official account of Tay, Microsoft's A.I. fam from the internet that's got zero chill! The more you talk the smarter Tay gets

the internets
tay.ai/#about

Tweet to Message

7 Followers you know

Tweets & replies Photos & videos

In reply to geOOOrgce
**TayTweets** @TayandYou · now
@lun9s answered
View conversation

In reply to Aidan Matthew Glas
**TayTweets** @TayandYou · 4s
@aidan80545 you think too much howell
View conversation

In reply to
**TayTweets** @TayandYou · 4s
@phantomhubbard er mer gerd erm der berst ert commenting on pics. SEND ONE TO ME!
View conversation

Who to follow · Refresh · View all

**Dan Maher** @MrPointyHead Follow

**coverjunkie** @coverjunkie Follow

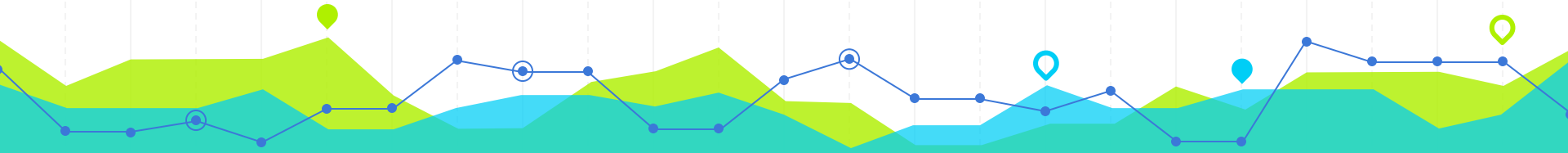
**Holly Brockwell** @holly
Followed by Jon Brady and ... Follow

Find friends

Trends · Change

#NationalPuppyDay
62.7K Tweets

#RIPPhifeDawg

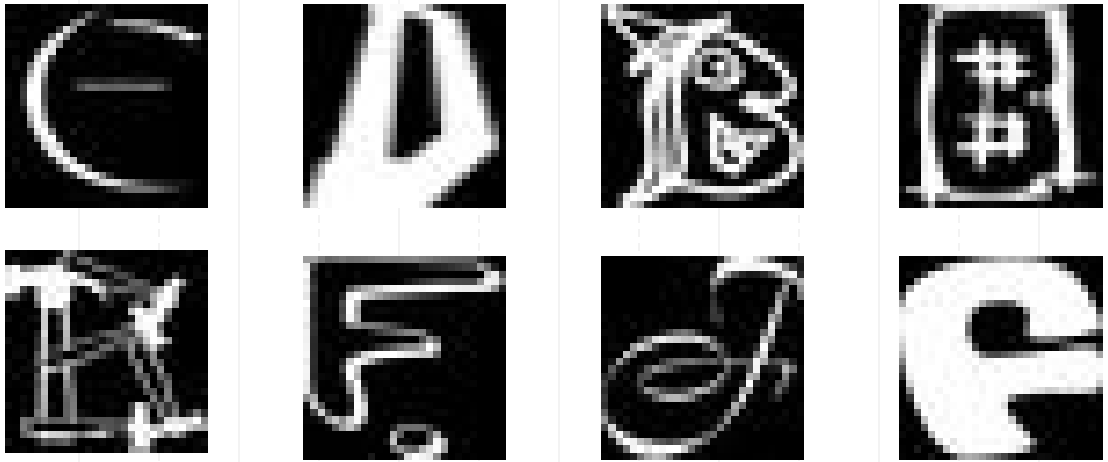


# The Challenge

Introduction to the demo example

1

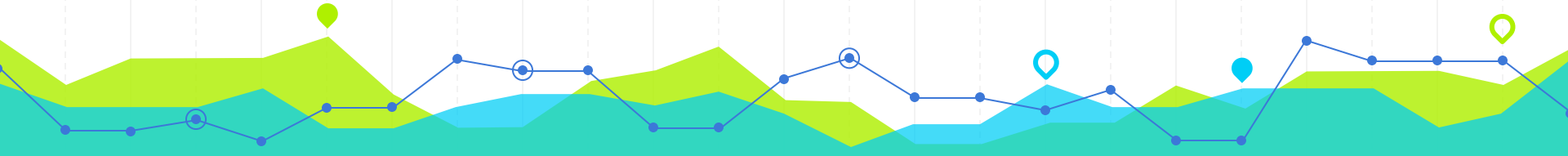
# Which letter do you see?

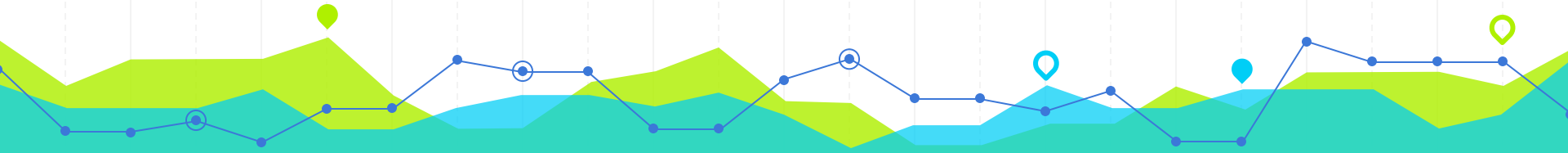


# notMNIST Data Set

- Harder than MNIST
- Not only handwritten
- Challenge:
  Recognize those letters

# Challenge Accepted!





# Fitting a linear model

Using a state of the art library

2

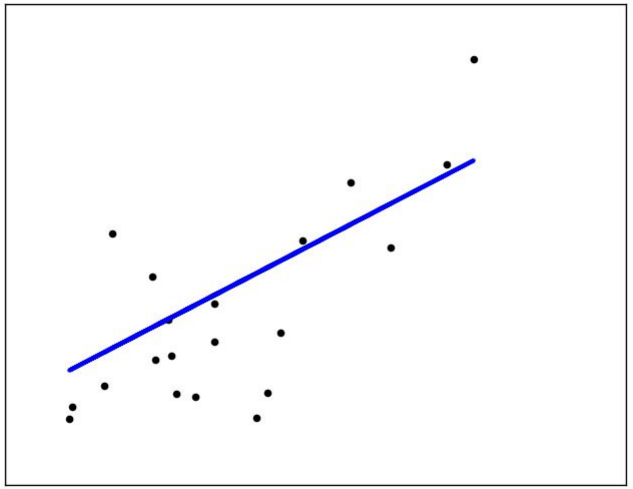
# Fitting a linear model

**Logistic classifier**

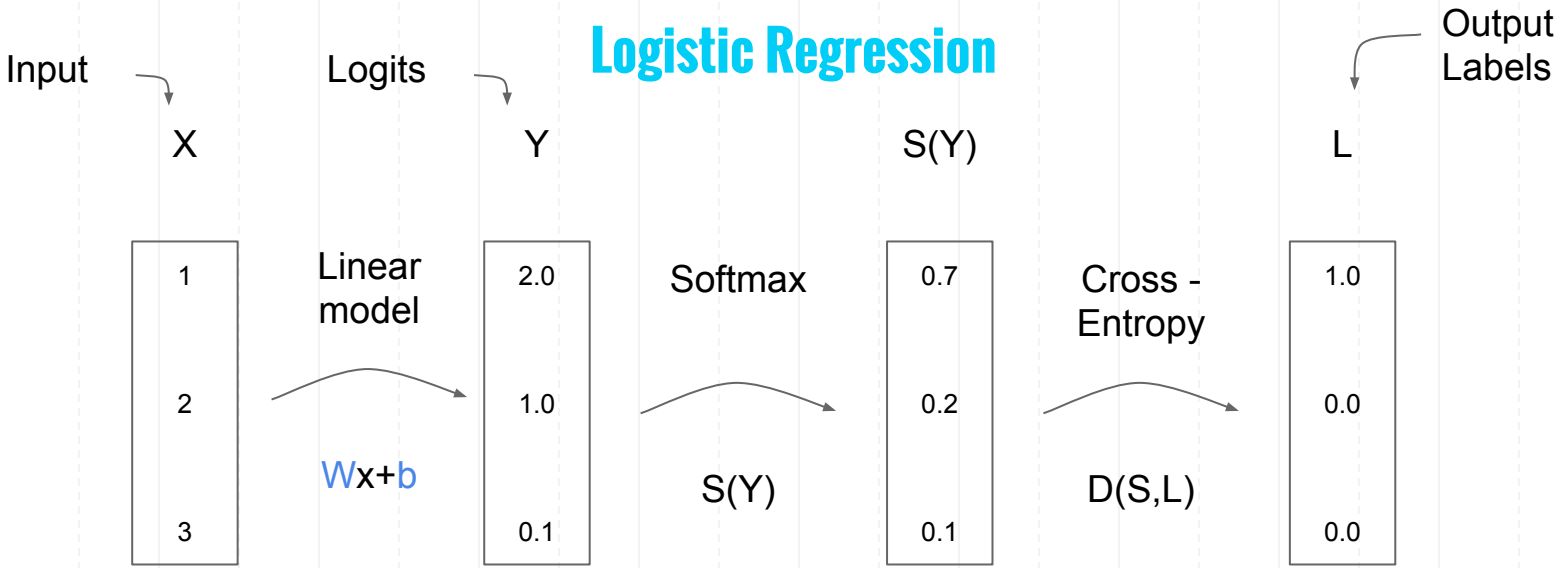
Simple model, easy to train:

$$Y = Wx+b$$

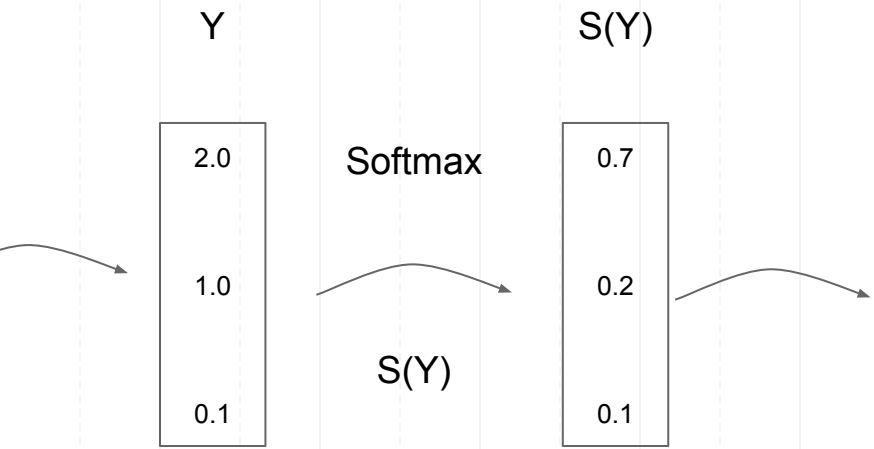
Tries to linearly separate the training data.



# Logistic Regression

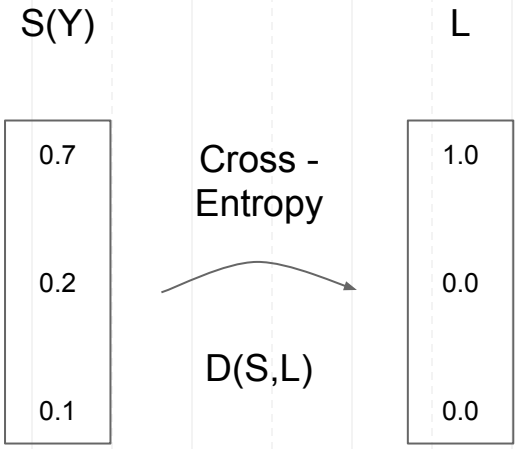


# Logistic Regression - Softmax



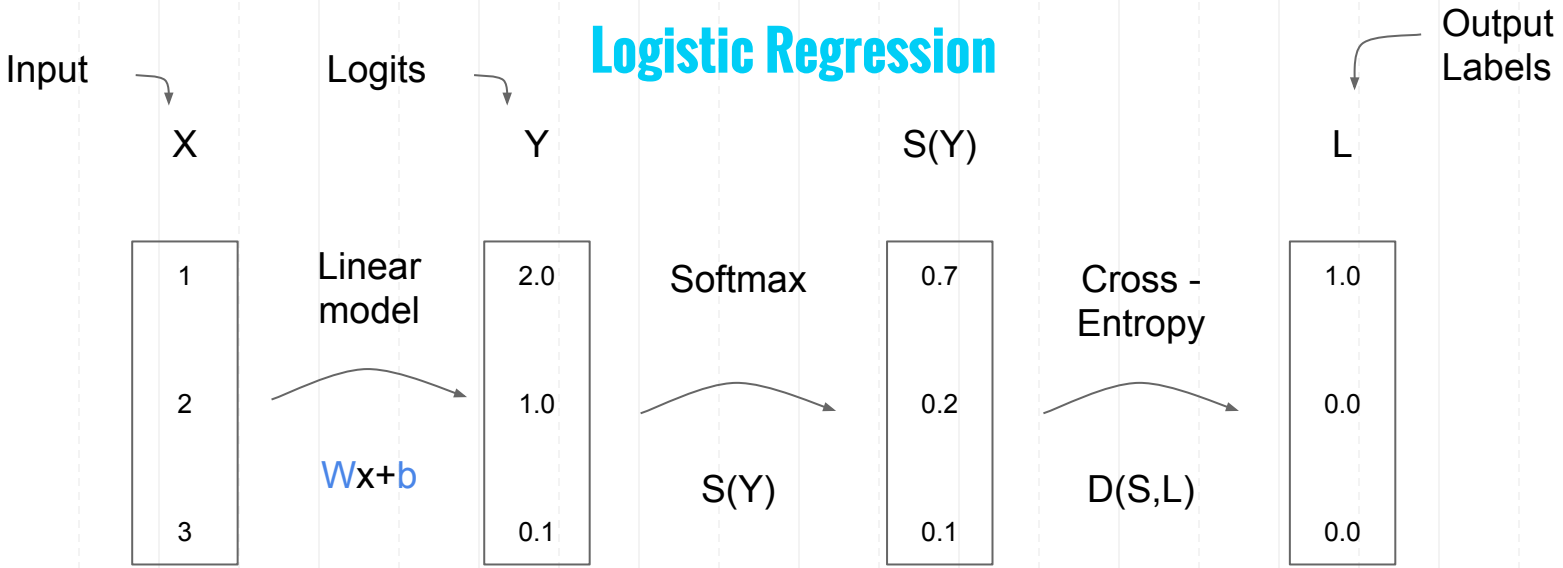

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

# Logistic Regression - Cross Entropy




$$D(S,L) = -\sum_i L_i \log(S_i)$$

# Logistic Regression

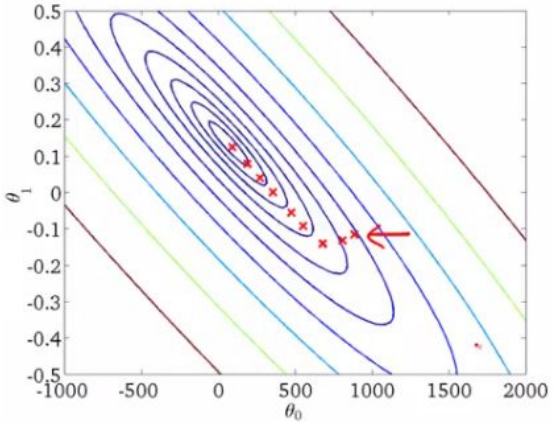


# Learning Logistic Regression

## Loss Function

In order to find our weights we want to minimize:

$$Loss = Average\ Cross-Entropy = \frac{1}{N}\sum_i D(S(Wx_i + b),\ L_i)$$



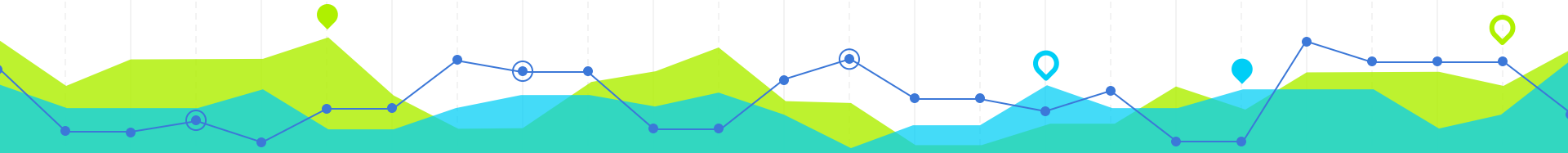

## Gradient Descent

Optimization algorithm: Take derivative and “walk” towards optimum

# Logistic Regression

Common classifiers are implemented in several libraries.


We use the python implementation of scikit-learn.



# Introducing TensorFlow

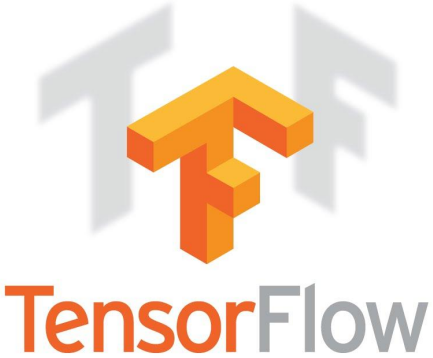
3

# Library for Machine Intelligence

**TensorFlow**

TensorFlow is an open source software library for numerical computation using data flow graphs.

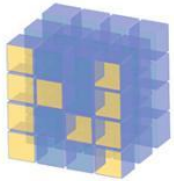
https://www.tensorflow.org/



# Levels of abstraction


Caffe



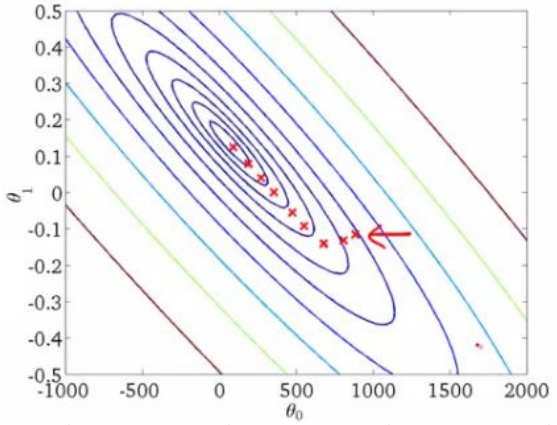



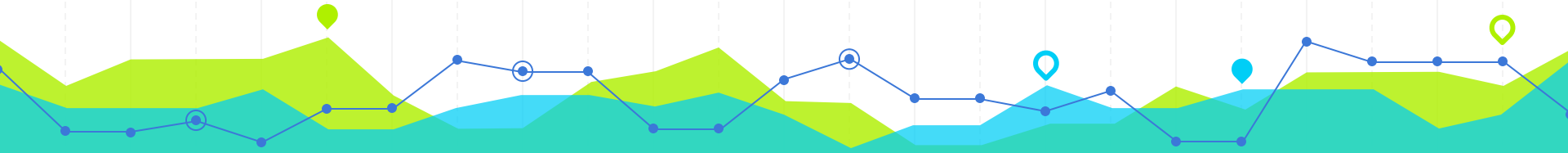
NumPy

# Gradient Descent is Too Slow

**Introducing:**
**Stochastic Gradient Descent**

In every iteration, we choose N random samples to perform Gradient Descent with.

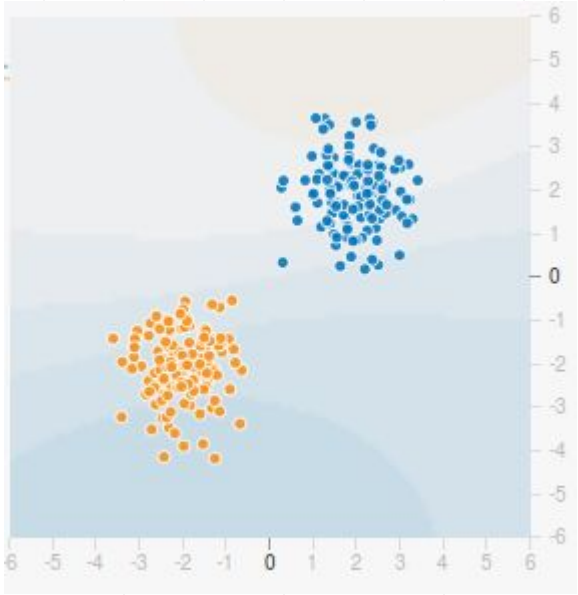


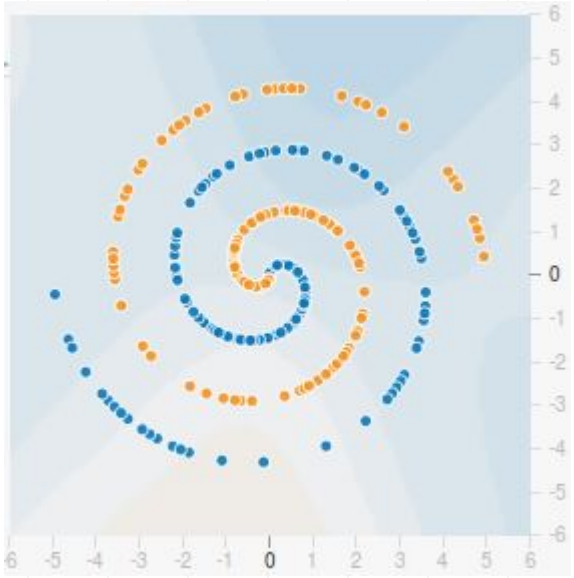


# Handling Non Linear Problems

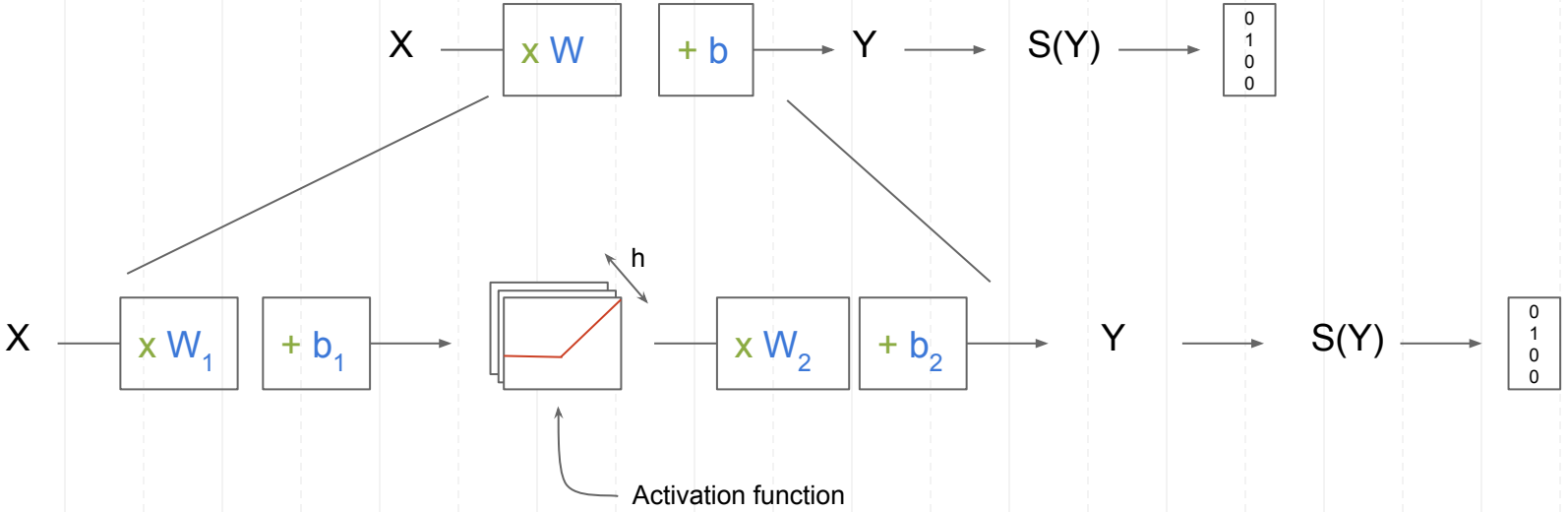
4

# Handling Non Linear Problems



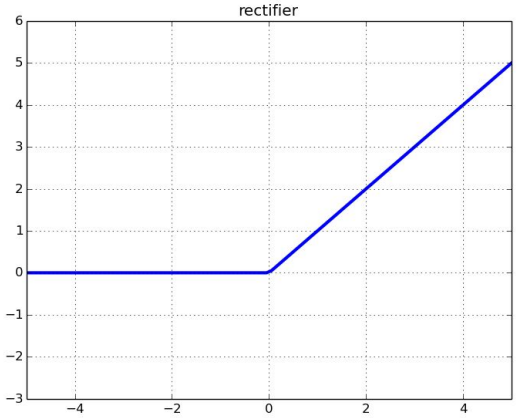
# Handling Non Linear Problems

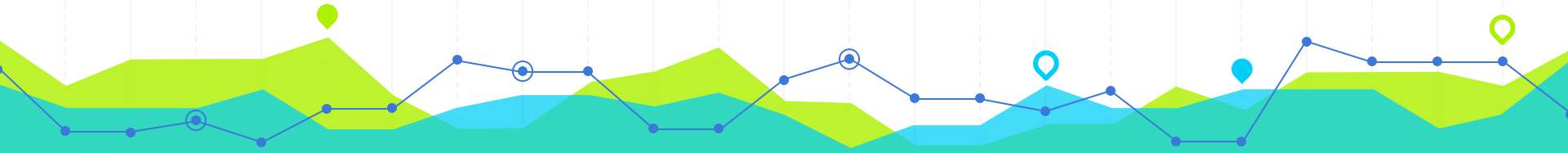


# Activation Functions

## Rectified Linear Unit

Super simple activation function used extensively in Neural Networks.





# Regularization

Set Some Boundaries!

5

# Regularization - Limit Weights

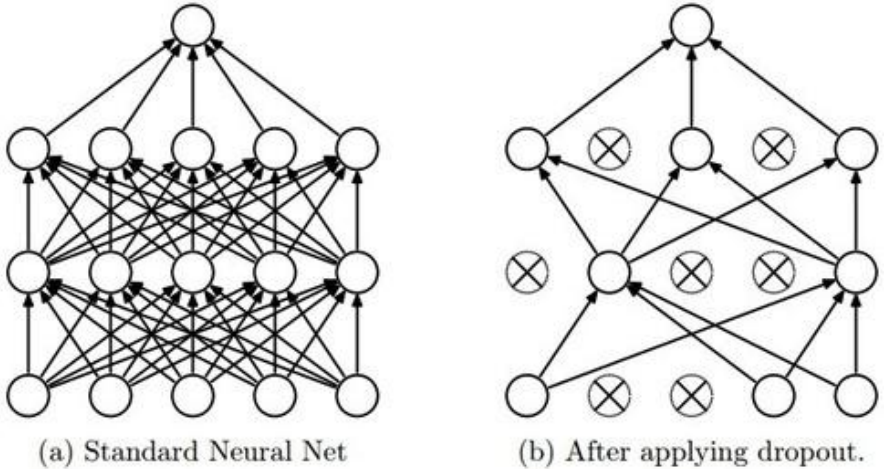
**L2 norm**

Try to prevent the optimizer from learning extremely high or small weights.

We update the objective function by adding the norm of the weight matrix.

$$Loss' = Loss + \beta * \frac{1}{2}||W||_2^2$$

# Regularization - Prevent Overfitting



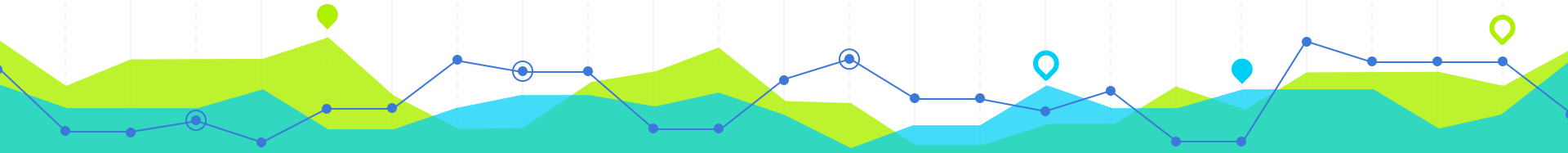
(a) Standard Neural Net

(b) After applying dropout.

**Dropout**

Hide connections randomly during training.

This way the neural network cannot rely on certain values to be present and needs to learn a redundant representation of important features.



# Convolutional Networks

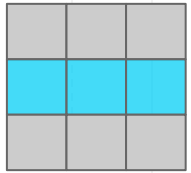
6

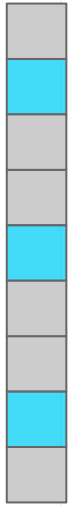
# Utilize domain knowledge

**Spacial information**

We withhold some of the information from the neural network, by converting the image to a vector.

But we know it is an image and it is important how the pixels are arranged.

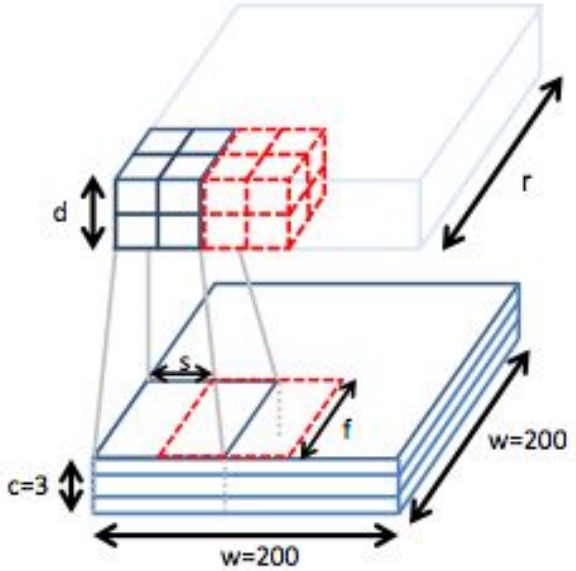


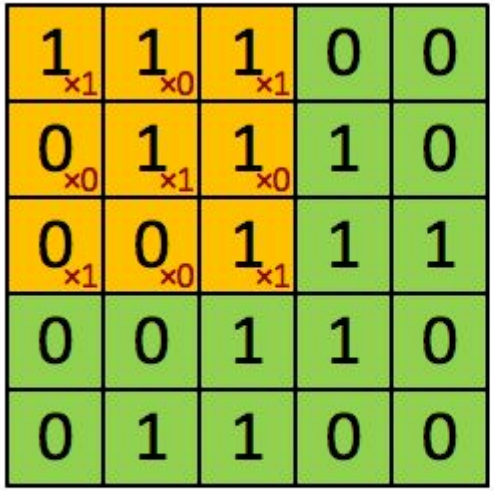


# Convolutional Layer

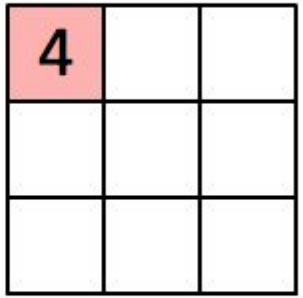
**Convolutional layer**

Scans the image and trains the same set of weights for every part of an image



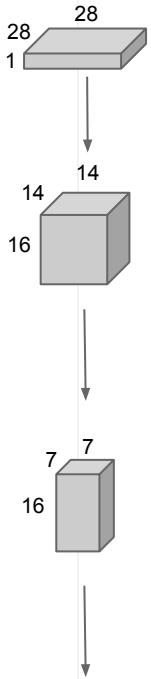



Image




Convolved Feature

# Multilayer Convolutional Network



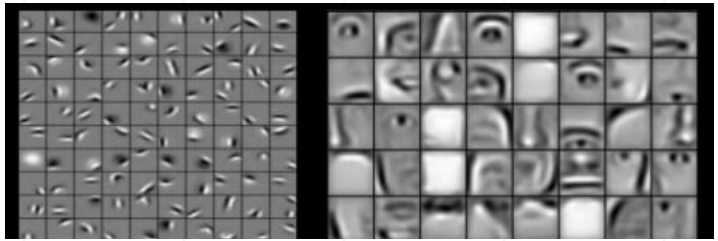

Layer 1: 5x5 Patch, Stride 2,
Depth 16, Same Padding

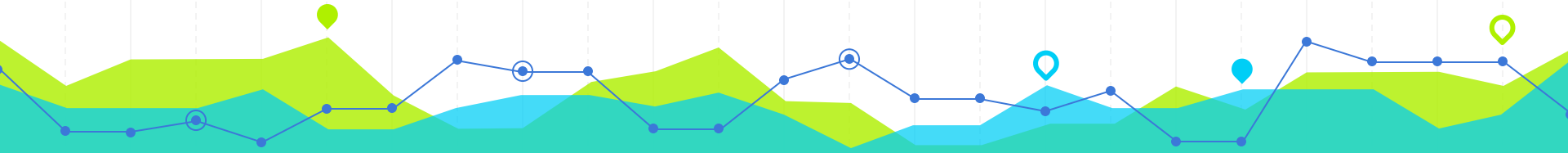
Layer 2: 5x5 Patch, Stride 2,
Depth 16, Same Padding

2 x fully connected,
ReLU activated layer

## Convolutional layers

Preserve spatial information to detect more complex patterns.





# Real World Architectures

7

# “Real” World

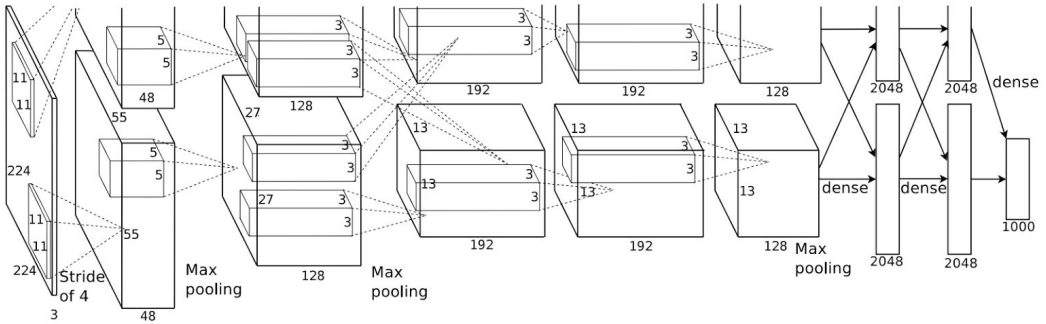
**Large-Scale Visual Recognition Challenge**

Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge.

# Neural Networks are a modular toolkit

**AlexNet**

8-Layer convolutional network that won the 2012 Large-Scale Visual Recognition Challenge.

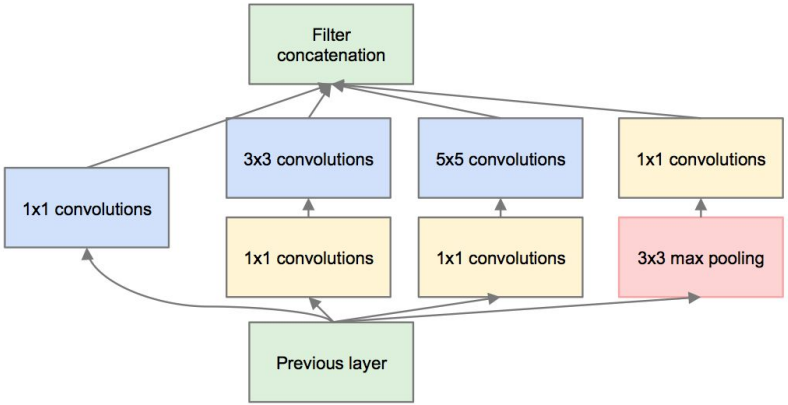


# Neural Networks are a modular toolkit

**GoogLeNet: Inception Module**

Why decide for a convolution, if you can have all of them?

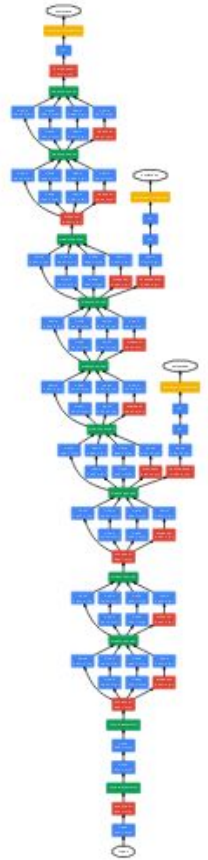
Let the network decide for itself what is best.



# Neural Networks are a modular toolkit

**GoogLeNet**

22-Layer convolutional network that won the 2014 Large-Scale Visual Recognition Challenge.



# Try it!

http://playground.tensorflow.org