Machine Learning for Engineers

AlphaGo

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


Google


Home
Moments
Notifications
Messages
Search Twitter
Tweet
Microsoft
Tay.ai
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

landscape

10. Juli 2015





18. Juni 2014


17. Juni 2014

16. Juni 2014

IBM Watson

@fluescher

Florian Lüscher

- bei Zühlke seit 2013
- Software Architektur
- Continuous Delivery
- Machine Learning
- Robo-Challenge

@bertolami

Roman Bertolami

- bei Zühlke seit 2008
- Software Architektur
- Cloud Computing
- Pattern Recognition


YOUR
EXPECTATIONS

Goals

- You understand the basic concepts of machine learning and neural networks
- You have the tools and the knowhow to continue working on machine learning topics
- You can build a simple classifier with TensorFlow

Challenge - notMNIST

Hands-On 0: Setup Environment

Hands-On 0: Setup Docker (preferred)

Start Notebooks using Docker

Step 1

Clone Github Repo:
https://github.com/fluescher/deep-learning-presentation

Step 2

Navigate to directory and start

```
./run-docker.sh
```

Hands-On 0: Setup

Start Notebooks using Azure Notepad

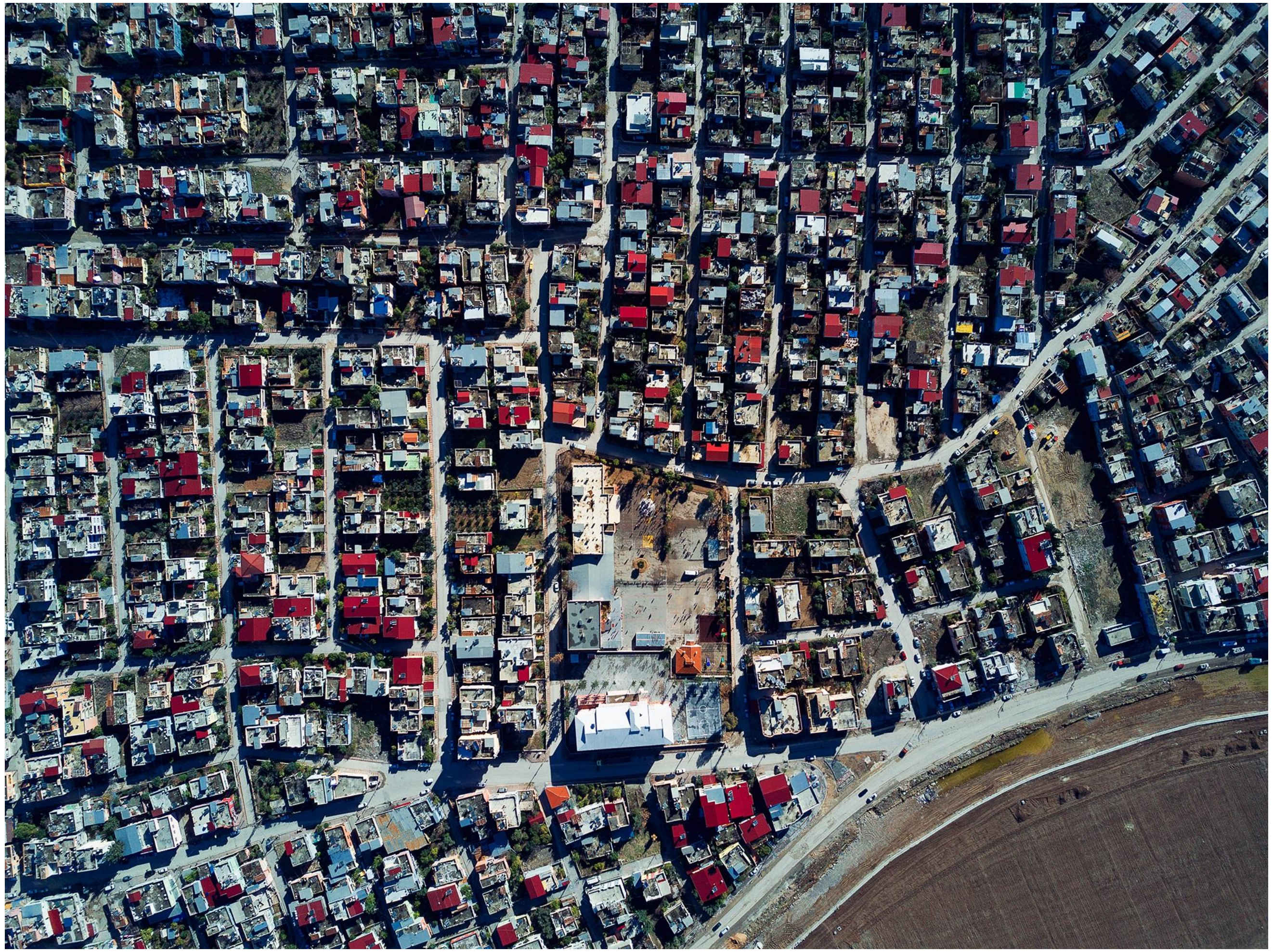
Step 1

If you don't have docker running goto:

https://notebooks.azure.com/anon-xc1gwa/libraries/machine-learning

Step 2

Clone the notebook and execute exercise 1

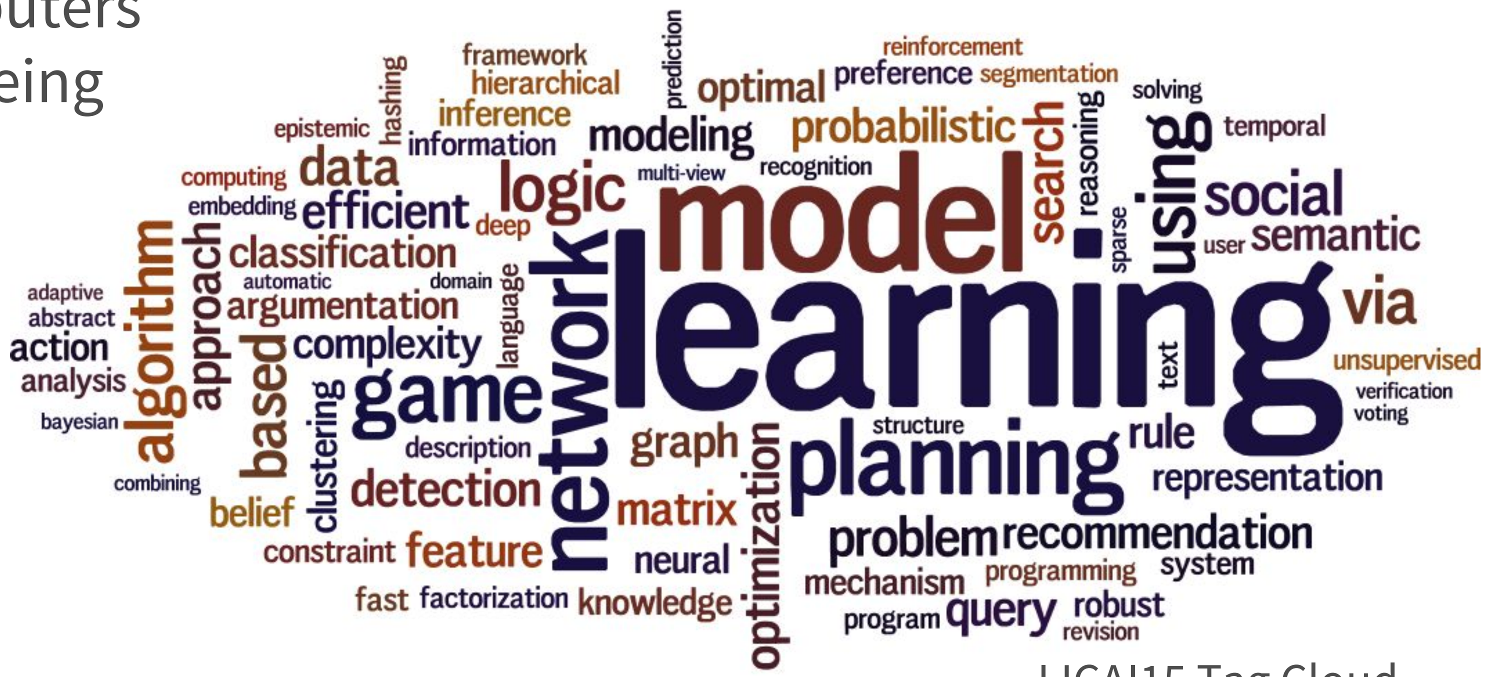


Machine Learning Overview

Machine Learning Definition

Wikipedia:

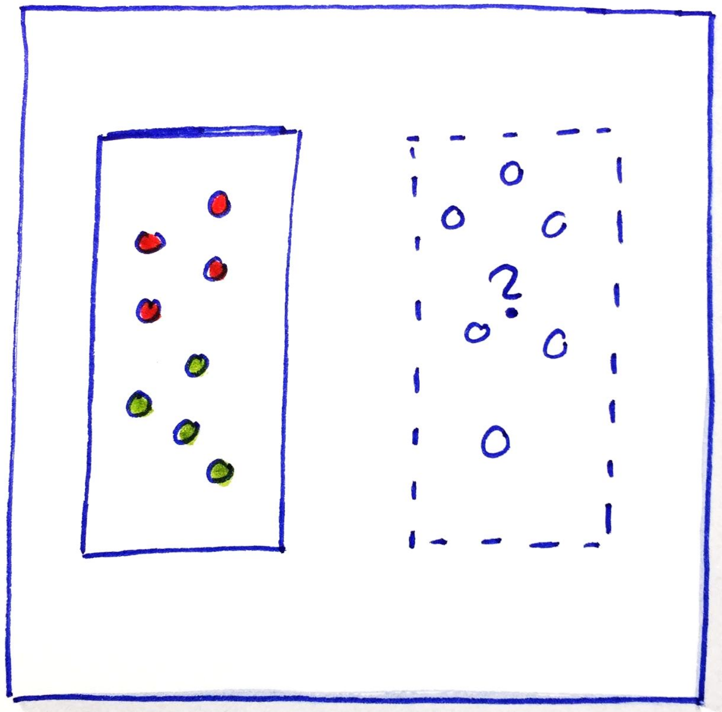
Machine learning gives computers the ability to learn without being explicitly programmed.



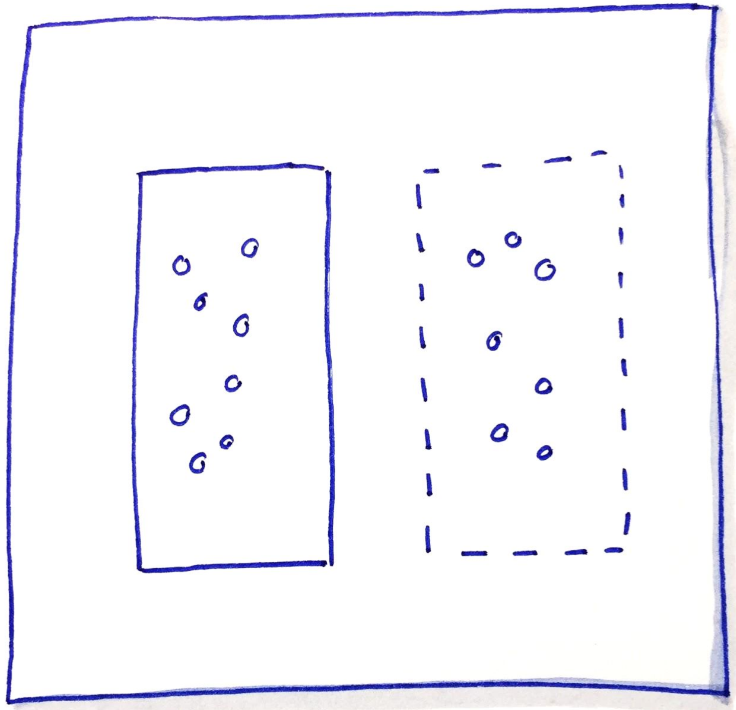
IJCAI15 Tag Cloud

Overview

Learning Methods

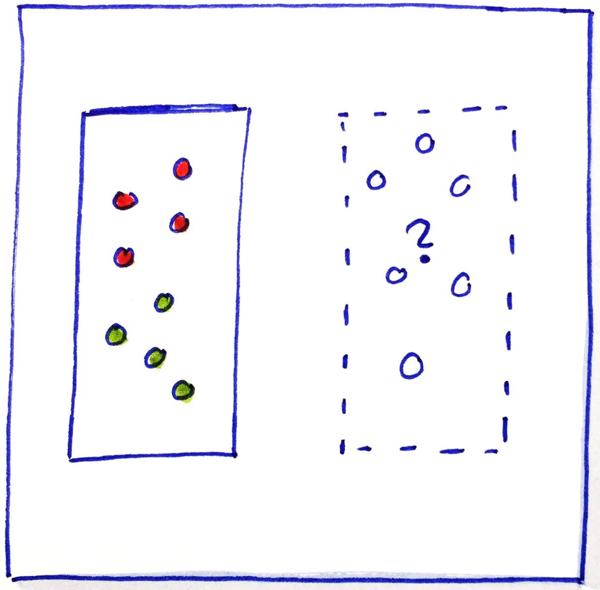


Supervised



Unsupervised

Supervised Learning



Infer a function from labeled training data

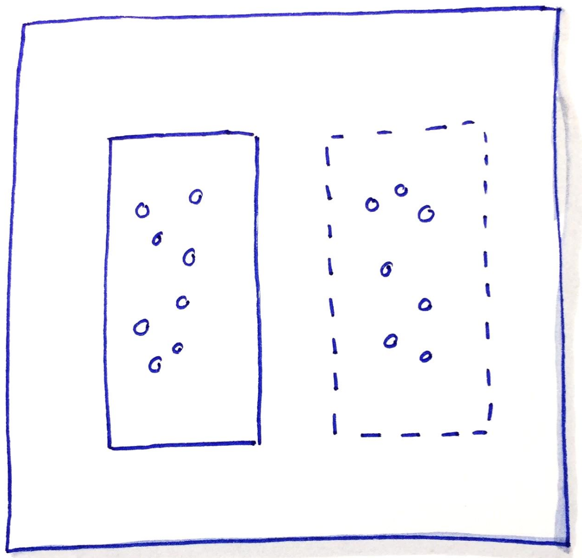
Typical problems:

- Optical Character recognition
- Handwriting recognition
- Speech recognition
- Object recognition
- ...

Algorithms:

- Naïve Bayes
- Support Vector Machine
- Nearest Neighbor Classifier
- Hidden Markov Model
- Conditional Random Fields
- Neural Networks
- Logistic Regression
- ...

Unsupervised Learning



Describe hidden structure from "unlabeled" data

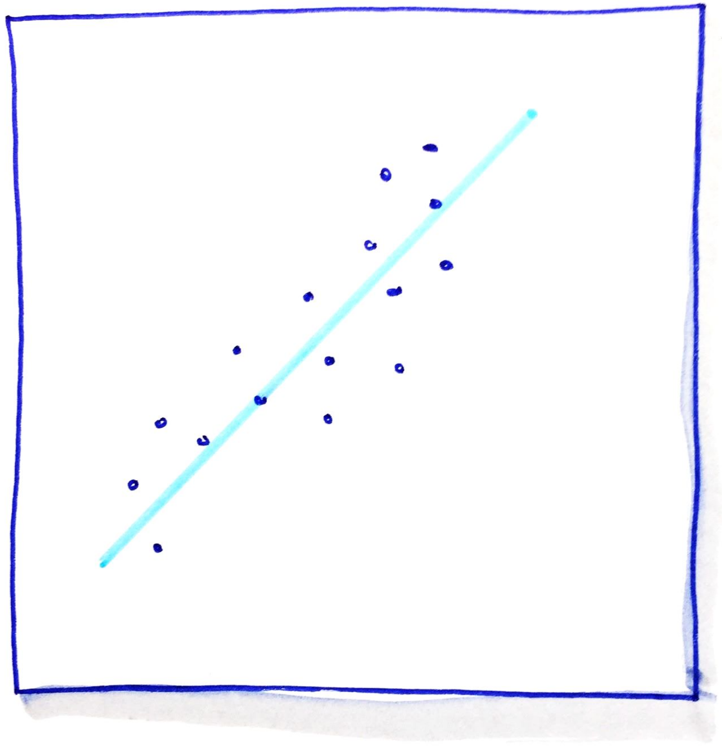
Typical problems:

- Clustering
- Product recommendation
- Outlier detection
- ...

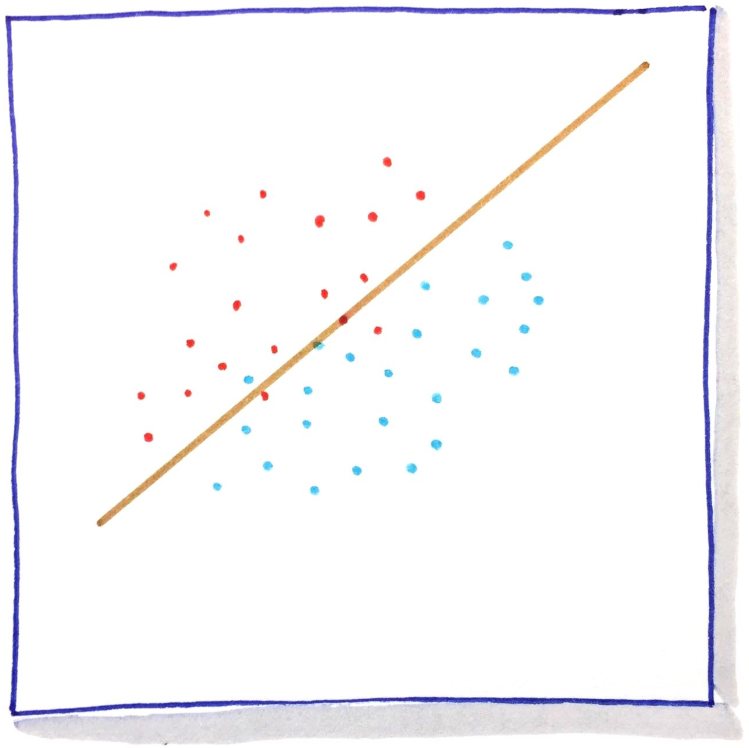
Algorithms:

- K-Means Clustering
- DBSCAN
- Neural Networks
- ...

Overview

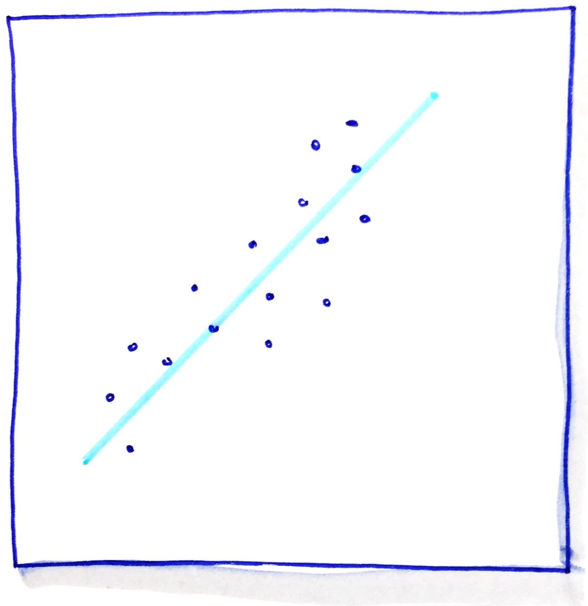


Regression



Classification

Regression



Regression analysis is a statistical process for estimating the relationships among variables.

Typical problems:

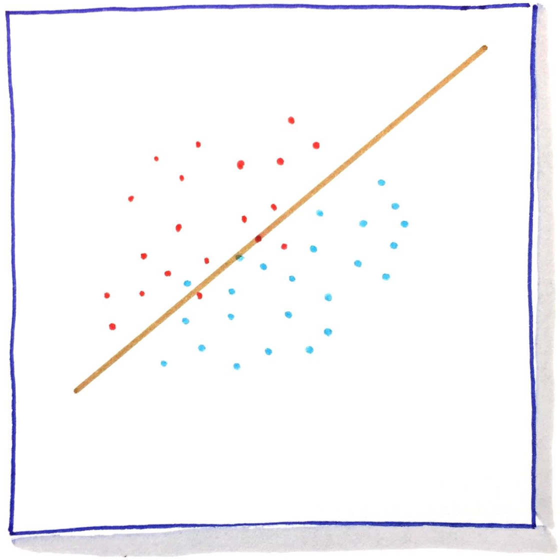
- Housing prices
- Prediction and forecasting
- Trend estimation
- …

Algorithms:

- Linear Regression
- Non-linear Regression
- Neural Networks
- ...

Important Note:
Correlation does not imply causation.

Classification



Classification is the problem of identifying to which of a set of categories a new observation belongs

Typical problems:

- Digit classification
- Fraud detection
- Fingerprint classification
- …

Algorithms:

- Naïve Bayes
- Support Vector Machine
- Nearest Neighbor Classifier
- Decision Tree
- Random Forest
- Neural Networks
- ...

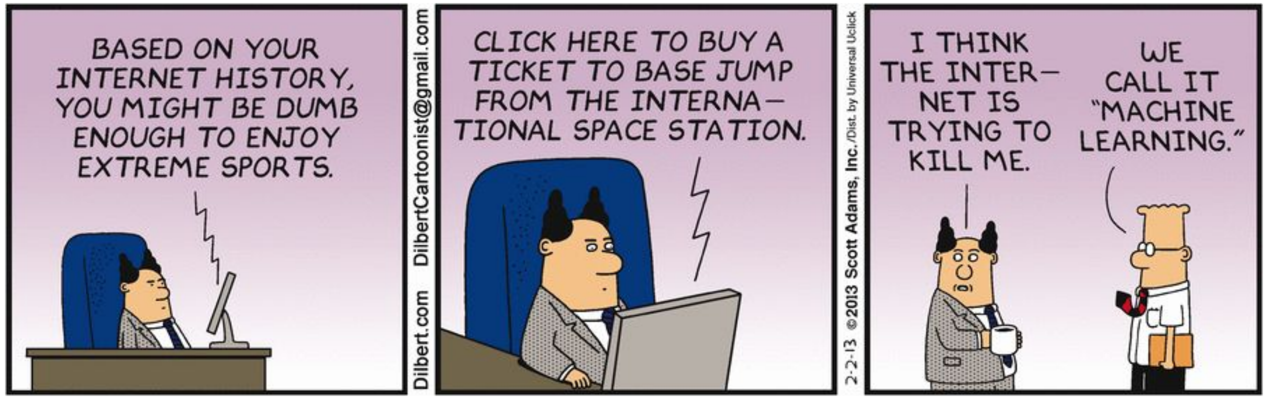
Discussion: Machine Learning in your Daily Life

1) Search concrete applications of machine learning

2) Classify the applications

- supervised / unsupervised
- regression / classification

3) How could a possible solution to the problem look like? What are the challenges?



TensorFlow

Tensorflow

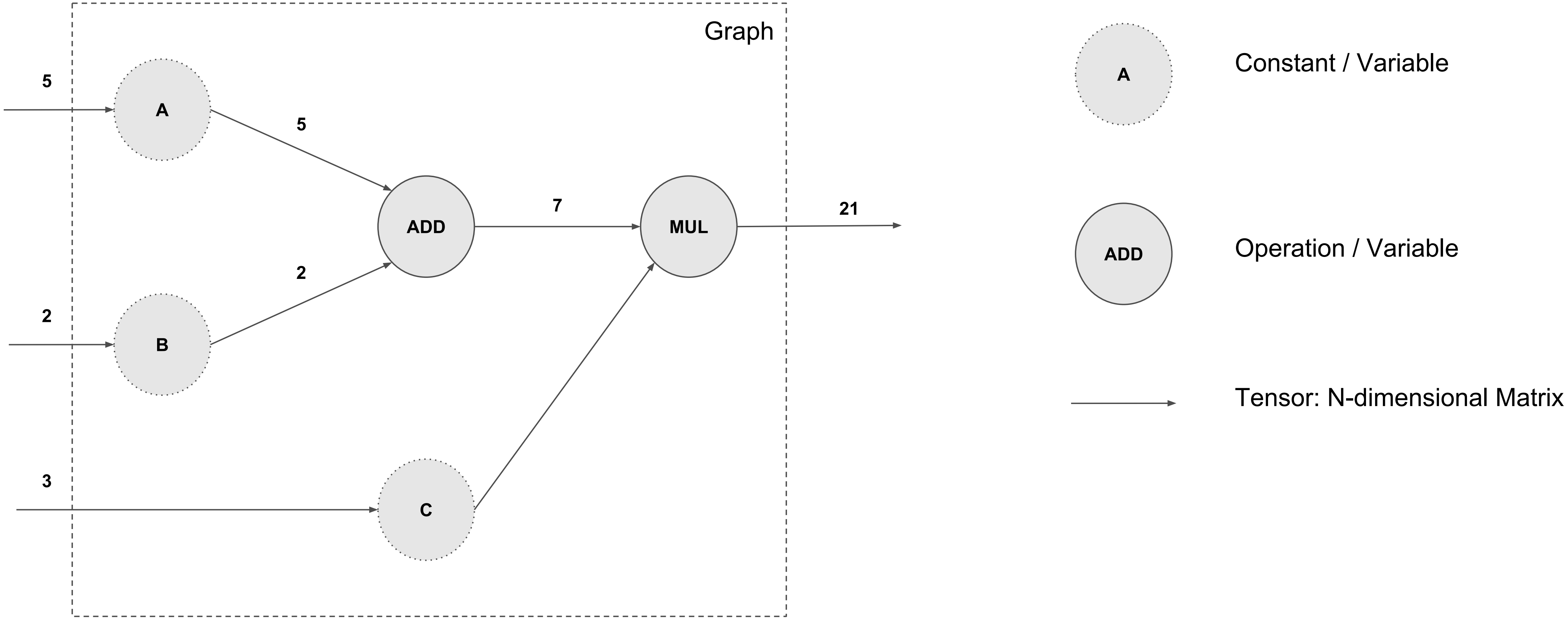
TensorFlow

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

https://www.tensorflow.org/

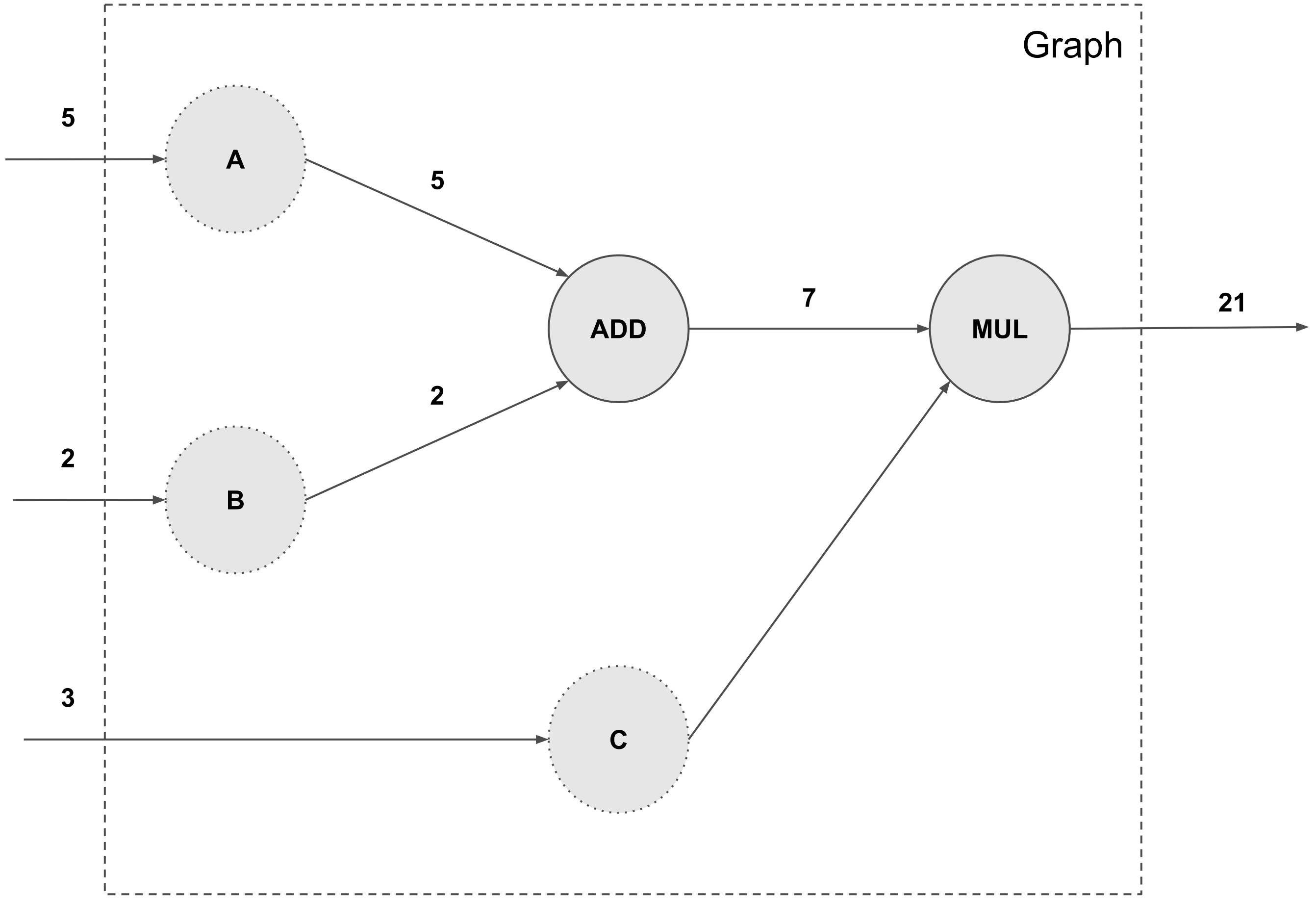
Tensorflow - Graphs

Terminology



Tensorflow - Graphs

Definition




```python
import tensorflow as tf

A = tf.constant(5)
B = tf.constant(2)
C = tf.constant(3)
I = tf.add(A, B)
R = tf.multiply(I, C)

print (R)
```

Output:

```
Tensor("Mul_2:0", shape=(), dtype=int32)
```

Tensorflow - Graphs

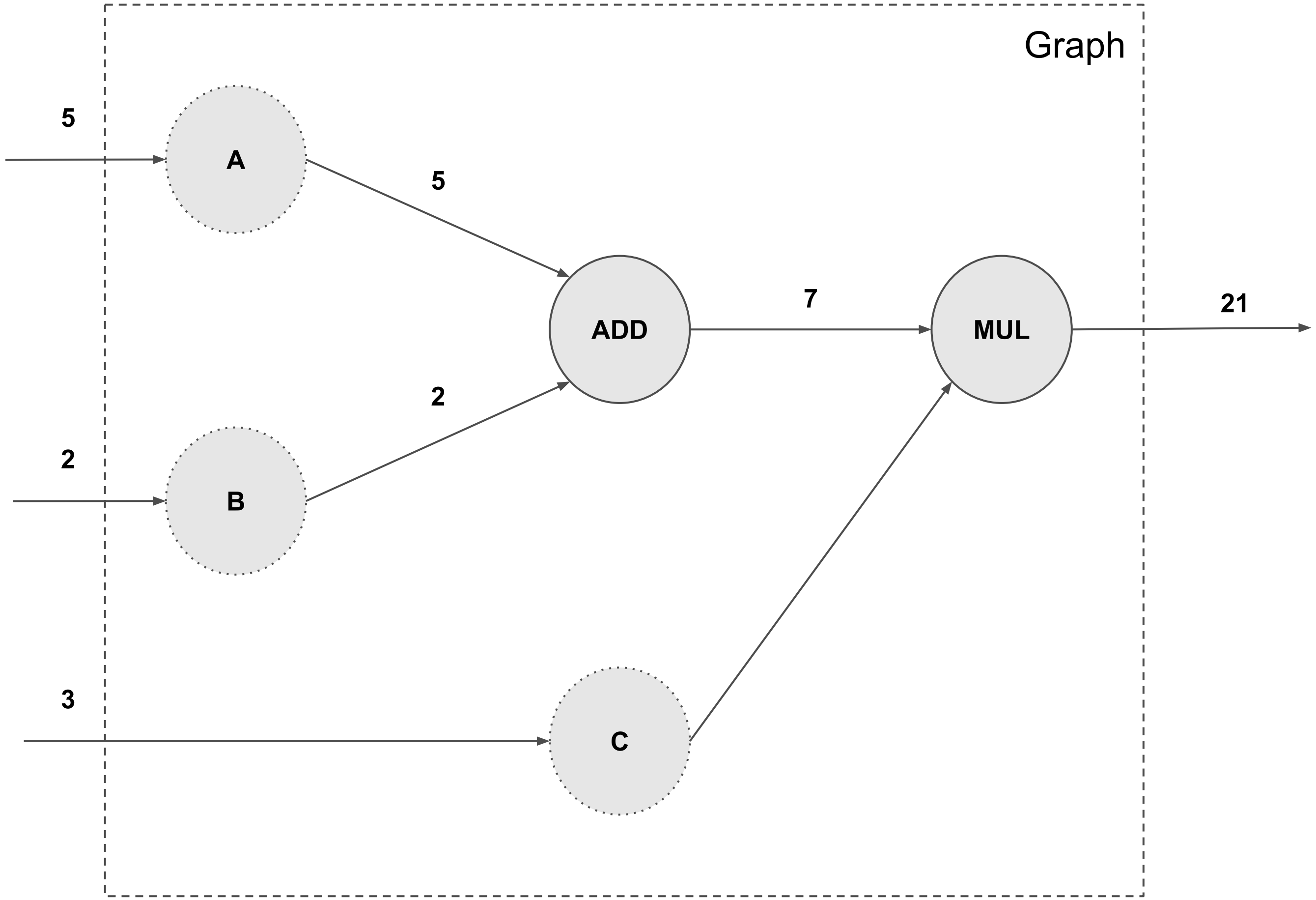
Execution

```
import tensorflow as tf

A = tf.constant(5)
B = tf.constant(2)
C = tf.constant(3)
I = tf.add(A, B)
R = tf.multiply(I, C)

with tf.Session() as session:
    tf.global_variables_initializer().run()
    res = session.run([R])
    print(res)
```

Output: `[21]`

Tensorflow - Graphs

Training

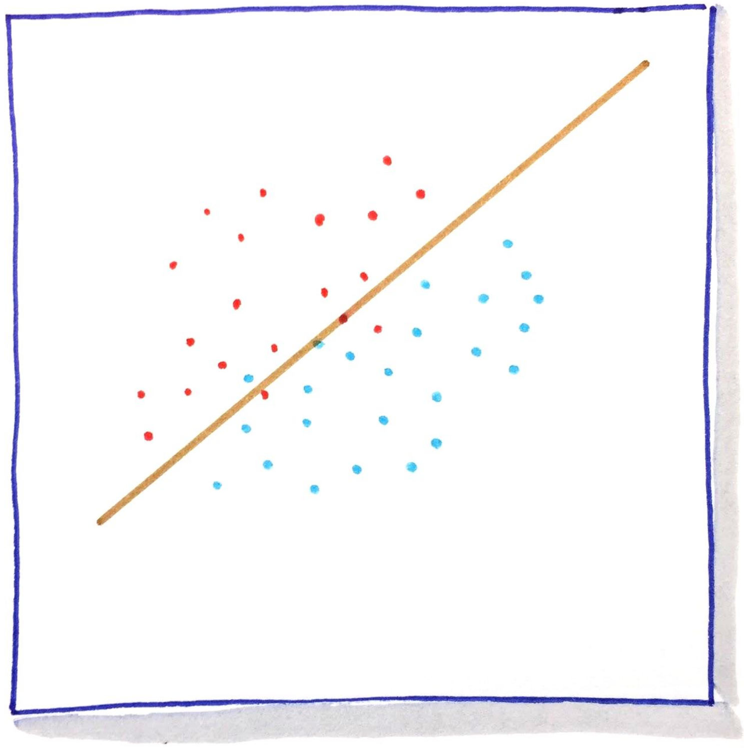


How can Tensorflow train a model?

Tensorflow analyzes the Graph. If an Optimizer like `tf.train.GradientDescentOptimizer` is used, Tensorflow starts to change the *`Variable`* values while leaving *`Constant`* unchanged.

Tensorflow

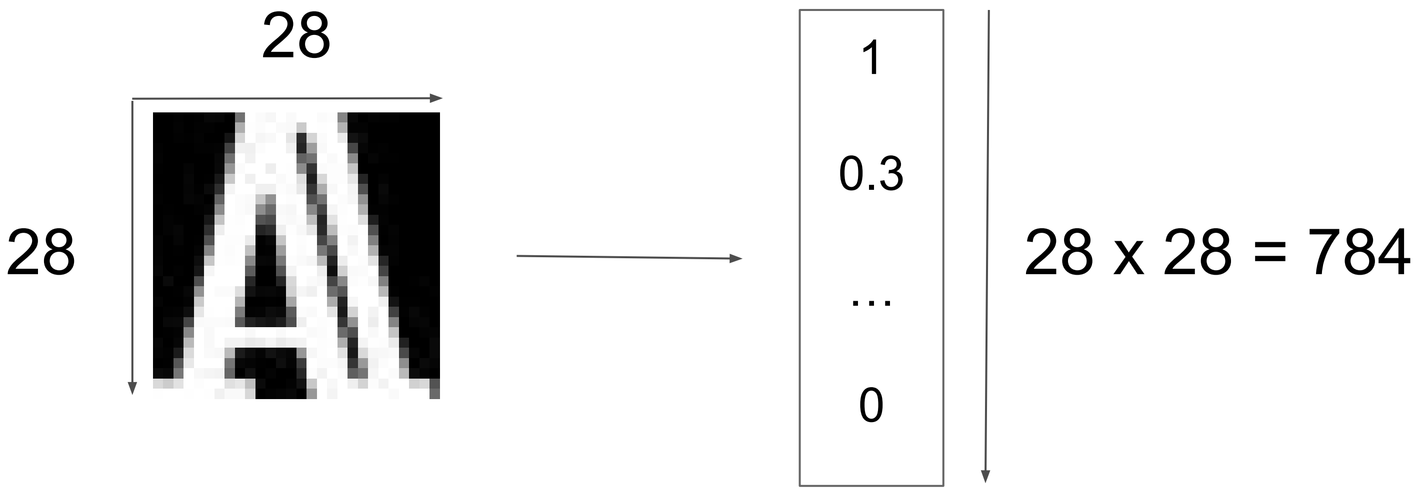
Hands-On 1: Intro into TensorFlow



Logistic Regression

Representation of Data

Images



How we represent our images

To simplify things, we transform our two dimensional images to a vector containing all pixel values.

Representation of Data

Classes

All Classes: A, B, C

One-Hot Encoding

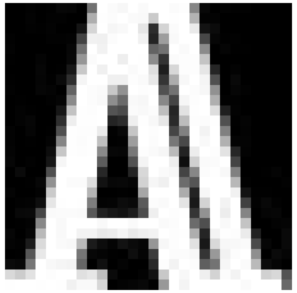
We represent a class by specifying a 1 in the corresponding column and 0 in all other columns

A	=>	1 0 0	**A** B C
B	=>	0 1 0	A **B** C
C	=>	0 0 1	A B **C**

Representation of Data

Input

X



Classificator

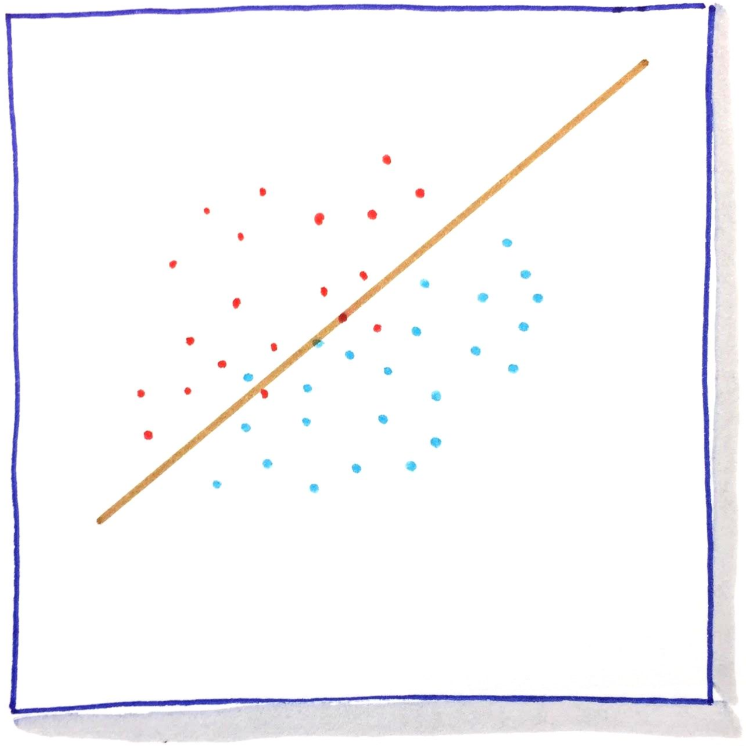
Estimated probability of class

Y

Class

0.8	A
0.1	B
0.1	C

Fitting a Linear Model



Logistic classifier

Simple model, easy to train:

$$Wx+b = y$$

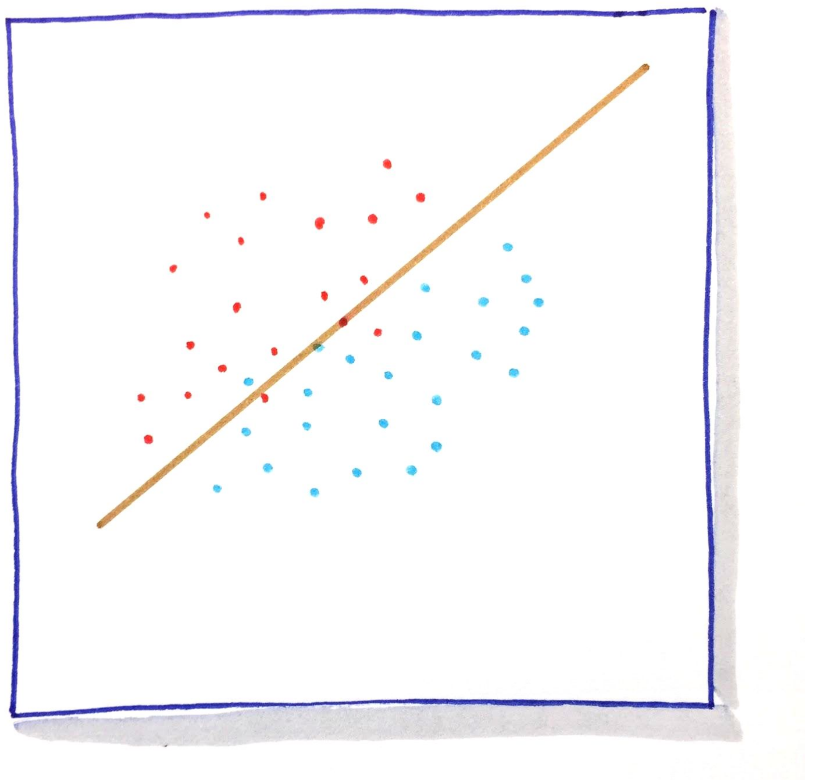
Tries to linearly separate the training data.

Logistic Regression - Linear Model

Input $\rightarrow$ x

Logits $\rightarrow$ y

$$\begin{bmatrix} 5 \\ 3 \\ 2 \\ 6 \end{bmatrix} \xrightarrow[Wx+b]{\text{Linear model}} \begin{bmatrix} 8 \\ 3 \\ 3 \end{bmatrix}$$



Logistic Regression - Softmax

Input → x

Logits → y

Prediction → $softmax(y)$

x	Linear model $Wx+b$	y	Softmax $\frac{exp(y_i)}{\sum_j exp(y_j)}$	$softmax(y)$
1		2.0		0.7
2	→	1.0	→	0.2
3		0.1		0.1
4				

Logistic Regression - Cross Entropy

Training labels

s = softmax(y)

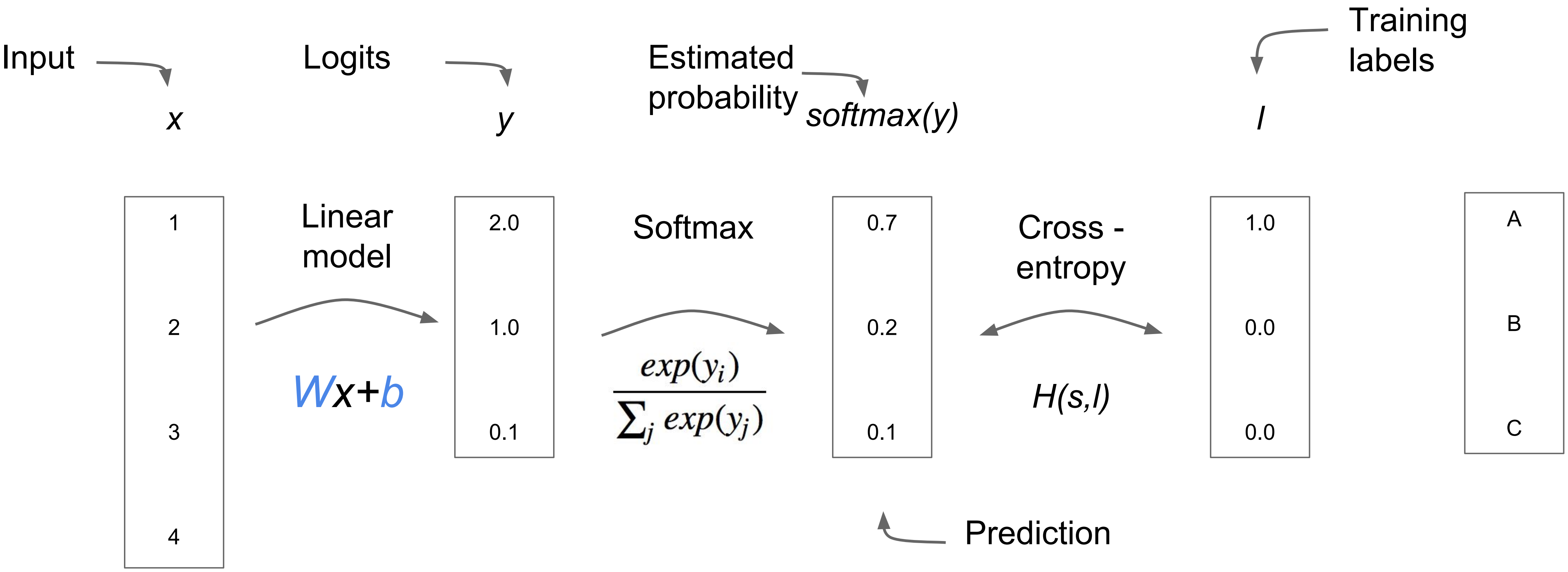
l

0.7	Cross - Entropy	1.0	A
0.2		0.0	B
0.1	*H(s,l)*	0.0	C

Prediction

$$H(s,l) = -\sum_i l_i log(s_i)$$

Logistic Regression - Full Example



Logistic Regression - Learning the Model Parameters

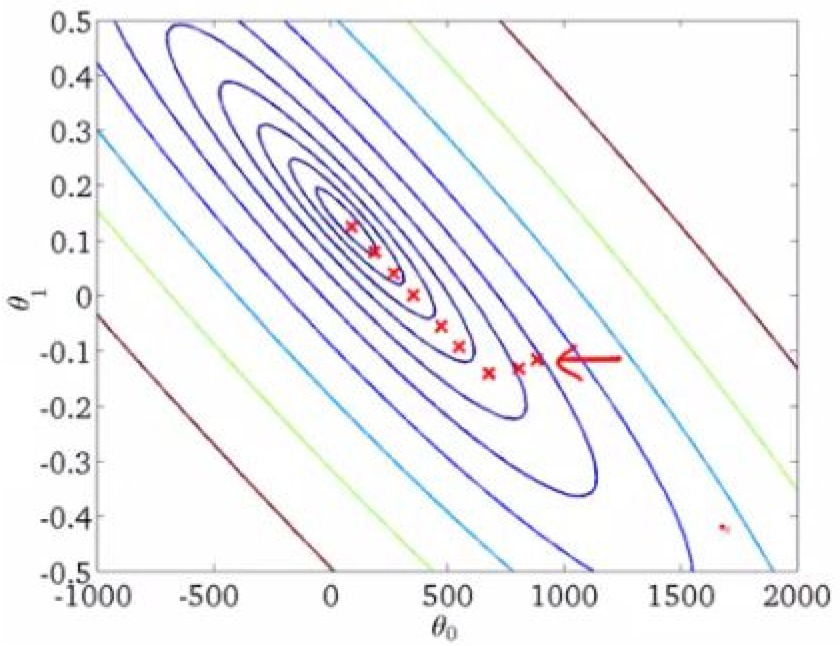
Our learning problem now is an optimization problem

Loss Function

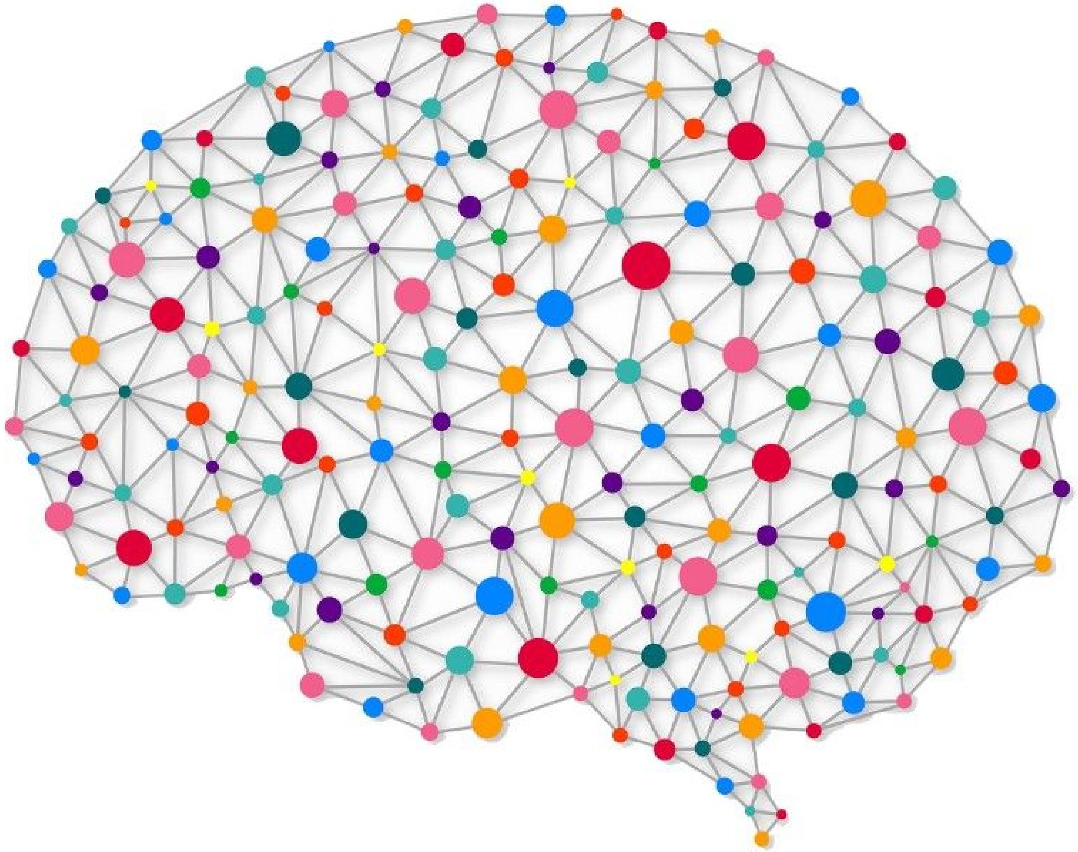
In order to find our weights we want to minimize the loss in our training set by choosing the appropriate weights and biases.

Gradient Descent

Optimization algorithm: Take derivative and "walk" towards optimum

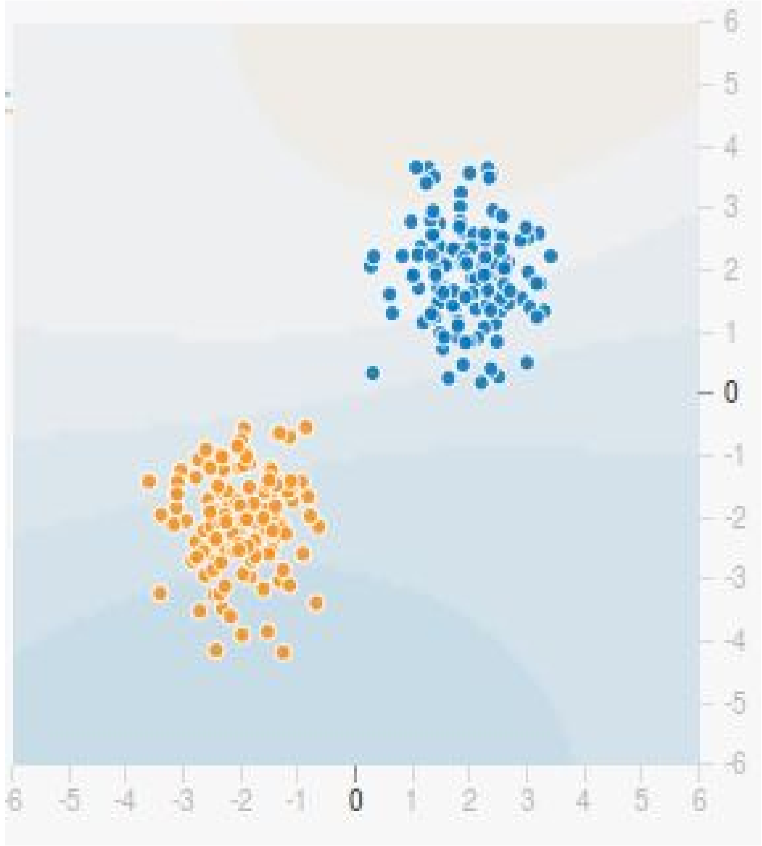


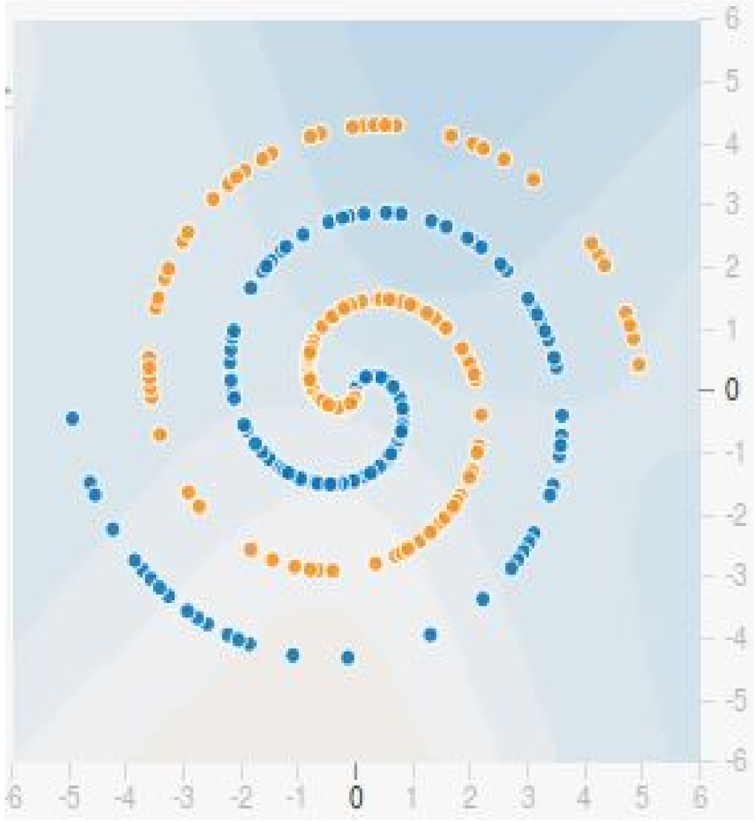
Hands-On 2: Our First Classifier



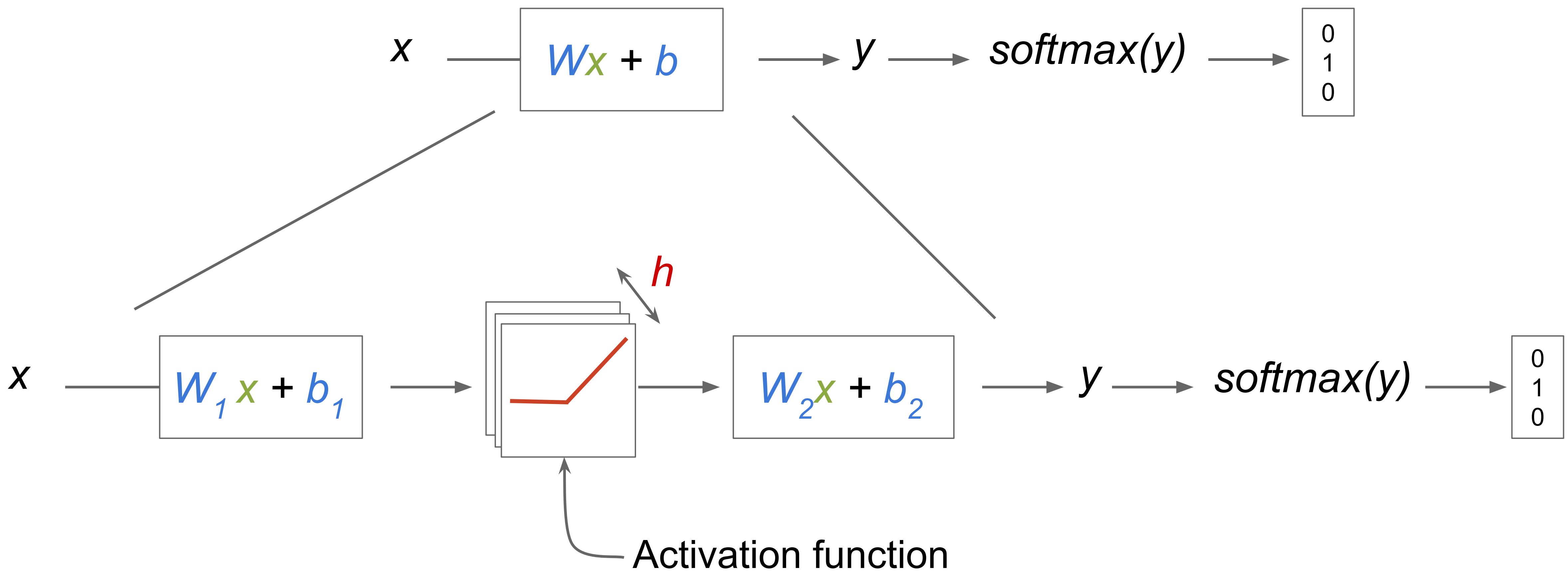
First Neural Network

Handling Non-Linear Problems

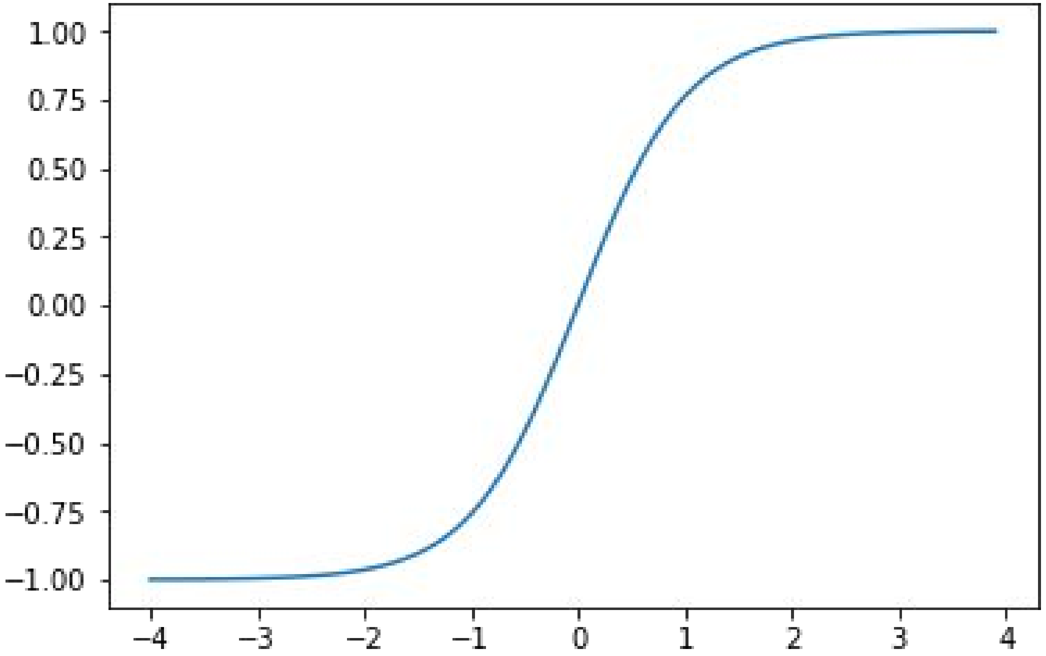




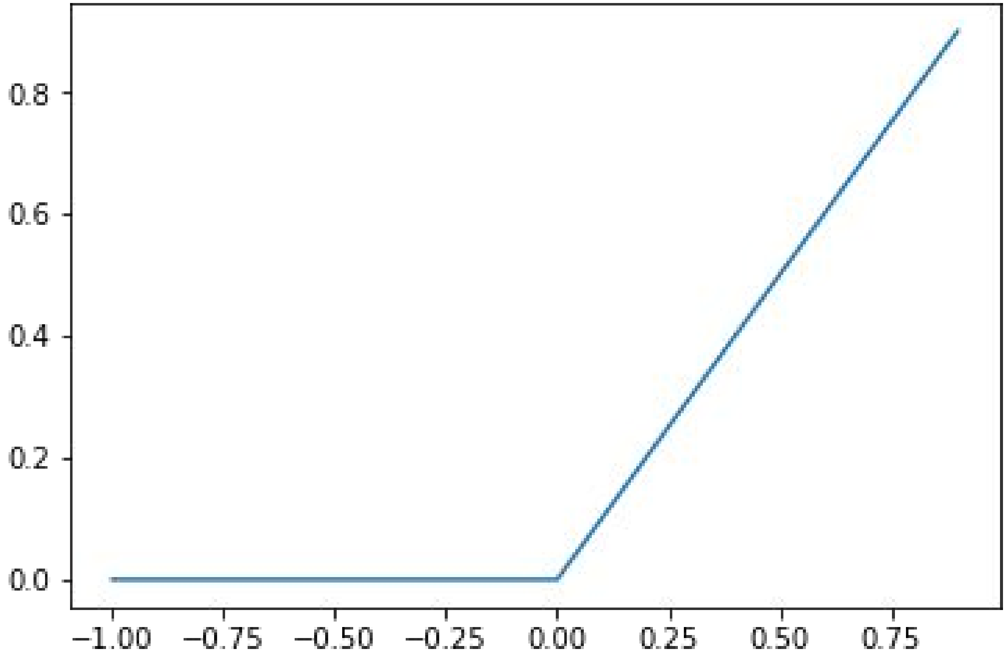
Handling Non-Linear Problems



Activation Functions

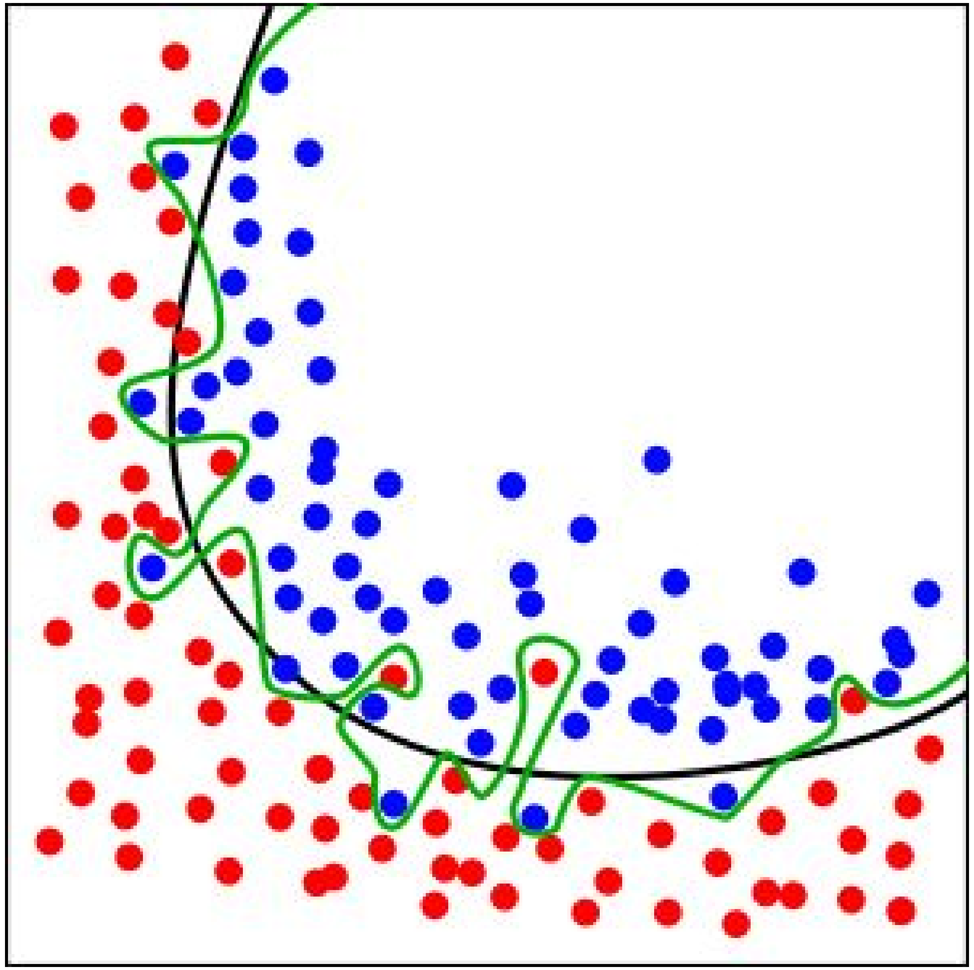


Tanh



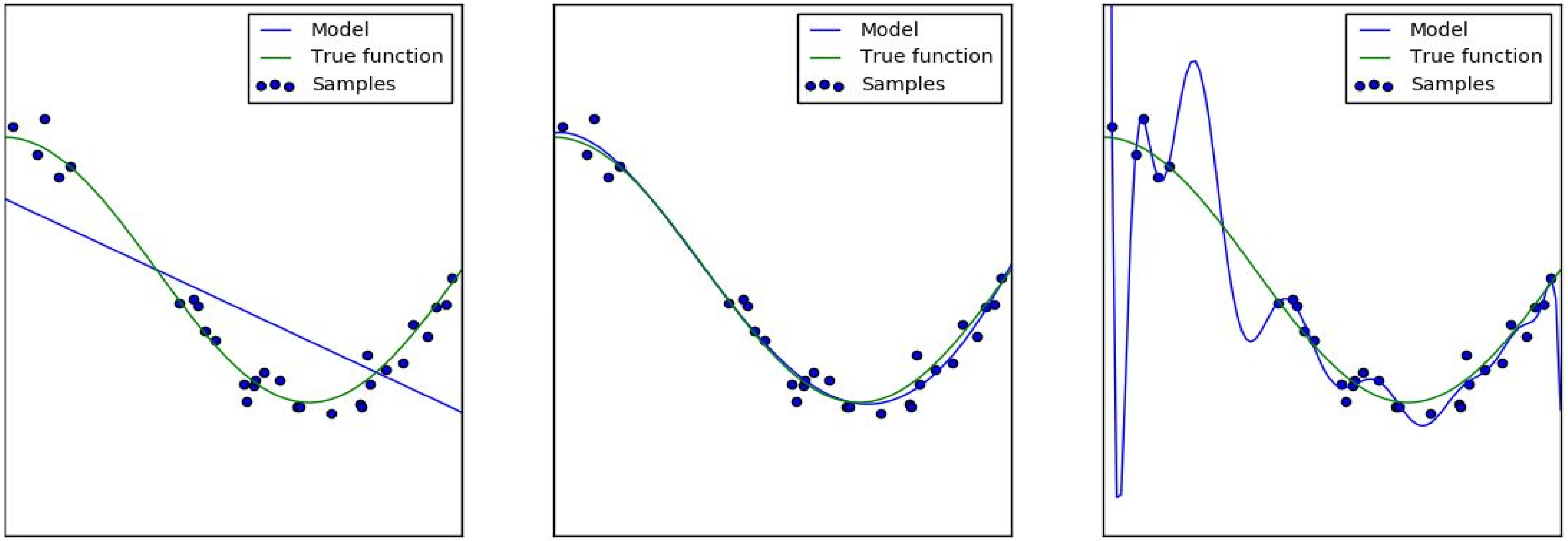
RELU
(rectified linear unit)

Hands-On 3: Our First Neural Network



Regularization

Regularization - Issues in Supervised Learning



Underfitting

Overfitting

Regularization - Limit Weights

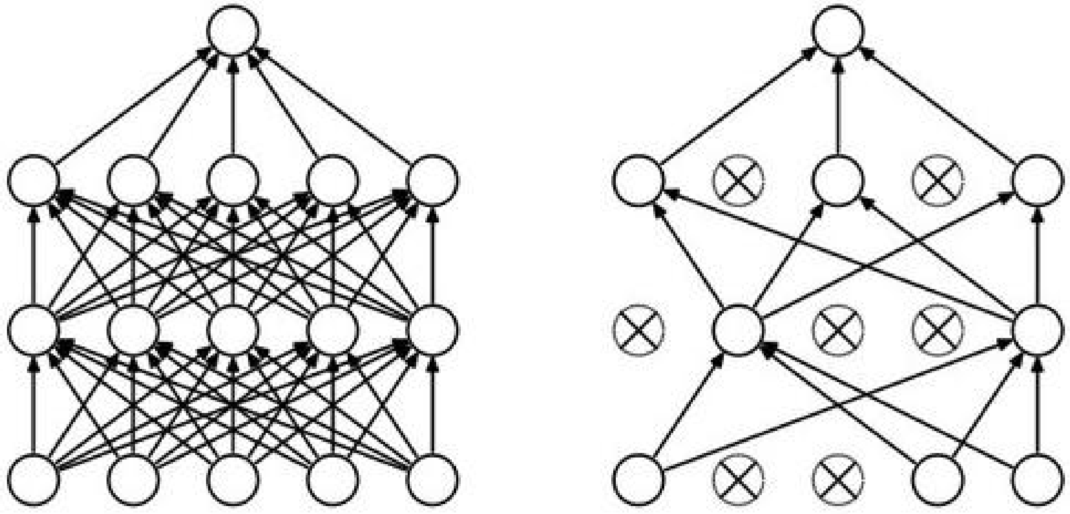
L2 norm

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 - Dropout



(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.

Hands-On 4: Regularization

Summary

Linear regression model as simple classifier

Neural network classifier to handle nonlinear problems

Regularization to avoid overfitting

Outlook

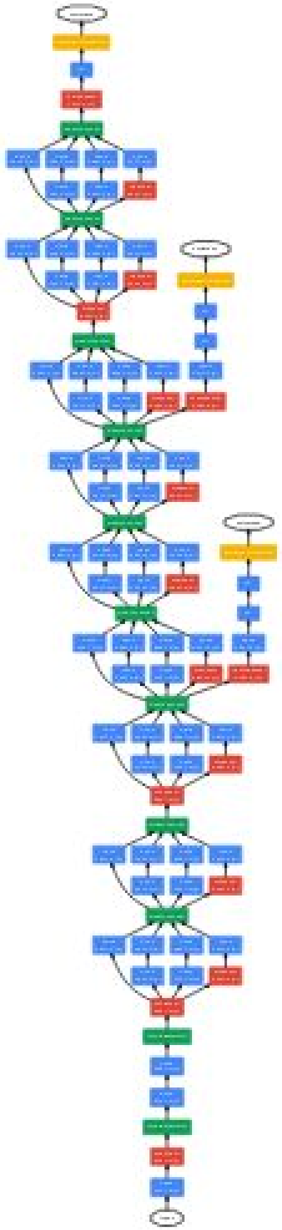
Deep Networks:

- Deep Feedforward Networks
- Convolutional Neural Networks
- Recurrent Neural Networks
- Long-Short Term Memory Nets

Representation Learning
Autoencoders

Outlook

Real World Models

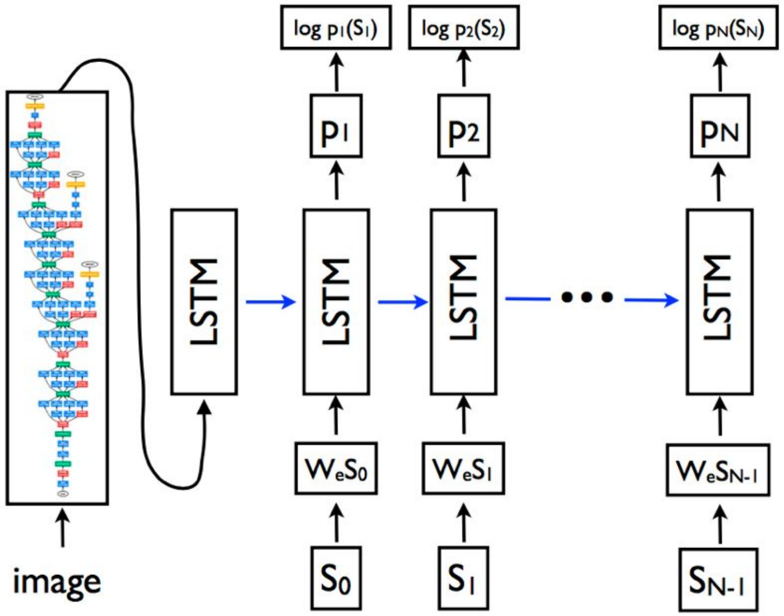


GoogLeNet

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

Model Combination

2015 MSCOCO Image Captioning Challenge



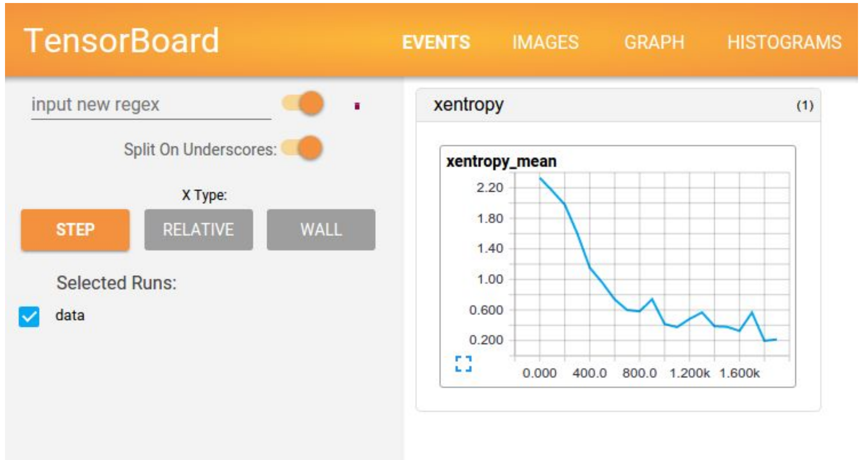
Outlook - Tools

TensorFlow Serving

Run your models in production:
https://tensorflow.github.io/serving/

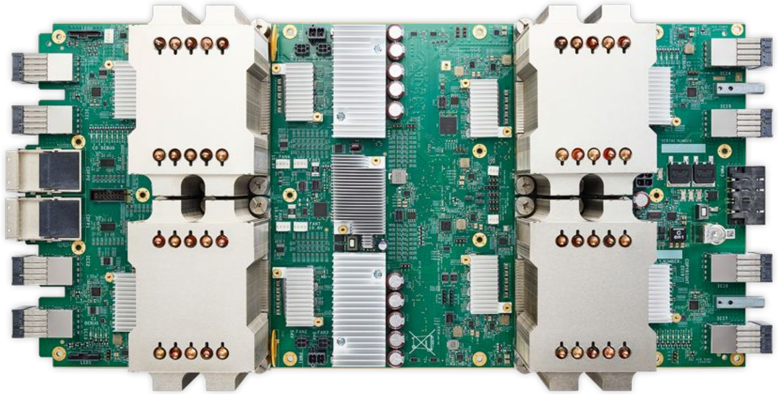
TensorBoard

Visualize Learning:
https://www.tensorflow.org/get_started/summaries_and_tensorboard



Outlook - Tools

TensorFlow on Google Cloud Platform
https://cloud.google.com/tpu/



TensorFlow Mobile

Run your models on Mobile Devices:
https://www.tensorflow.org/mobile/

Try It!

TensorFlow

https://www.tensorflow.org/

http://playground.tensorflow.org/

Examples & Presentation

https://github.com/fluescher/deep-learning-presentation


YOUR
FEEDBACK

1 Linear Model

$$Wx + b = y$$

$$x = \begin{pmatrix} 5 \\ 3 \\ 2 \\ 6 \end{pmatrix} \qquad b = \begin{pmatrix} -27 \\ -27 \\ -27 \end{pmatrix}$$

$$W = \begin{pmatrix} 3 & 2 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 2 & 3 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 3 & 2 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 3 \\ 2 \\ 6 \end{pmatrix} + \begin{pmatrix} -27 \\ -27 \\ -27 \end{pmatrix} = y$$

$$= \begin{pmatrix} 3 \times 5 + 2 * 3 + 2 * 2 + 1 * 6 \\ 1 * 5 + 3 \times 3 + 1 \times 2 + 2 \times 6 \\ 2 * 5 + 2 * 3 + 3 \times 2 + 1 * 6 \end{pmatrix} + \begin{pmatrix} -27 \\ -27 \\ -27 \end{pmatrix} = y = \begin{pmatrix} 31 \\ 28 \\ 28 \end{pmatrix} + \begin{pmatrix} -27 \\ -27 \\ -27 \end{pmatrix}$$

$$= \underline{\underline{\begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix}}}$$

② Softmax

$$\text{softmax}(y) = \frac{\exp(y_i)}{\sum_j \exp(y_j)}$$

$$\text{softmax}\left(\begin{pmatrix}4\\1\\1\end{pmatrix}\right) = \begin{pmatrix}\frac{s_1}{s}\\ \frac{s_2}{s}\\ \frac{s_3}{s}\end{pmatrix}$$

$$s_i = \frac{\exp(y_i)}{\cancel{\sum_j \exp(y_j)}}$$

$$S = s_1 + s_2 + s_3$$

$$s_1 = \exp(4) = \underline{54.5}$$

$$s_2 = \exp(1) = \underline{2.7} \qquad S = s_1 + s_2 + s_3 = 54.5 + 2.7 + 2.7$$

$$s_3 = \exp(1) = \underline{2.7} \qquad \underline{S = 60.03}$$

$$\text{softmax}\left(\begin{pmatrix}4\\1\\1\end{pmatrix}\right) = \begin{pmatrix}\frac{s_1}{s}\\ \frac{s_2}{s}\\ \frac{s_3}{s}\end{pmatrix} = \begin{pmatrix}\frac{54.5}{60}\\ \frac{2.7}{60}\\ \frac{2.7}{60}\end{pmatrix}$$

$$= \underline{\underline{\begin{pmatrix}0.91\\ 0.045\\ 0.045\end{pmatrix}}}$$

③ Cross Entropy

$$H(s, l) = -\sum_i l_i \log(s_i)$$

$s = \text{softmax}(y)$

l = One-Hot training label

$$H\left(\begin{pmatrix} 0.91 \\ 0.045 \\ 0.045 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right) = -(1 \cdot \log(0.91) + 0 \cdot \log(0.045) + 0 \cdot \log(0.045))$$

$$= -(-0.041..) = \underline{\underline{0.041}}$$

Fehler für diesen einen Datensatz

Für Optimierung muss der Fehler über das gesamte Trainingsset berücksichtigt werden ⇒ Summe oder Durchschnitt = Loss

④ Regularization

$$Loss' = Loss + \beta * \frac{1}{2} \|W\|_2^2 \qquad \Big| \qquad \|W\|_2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2}$$

$$a)\ a = \|W\|_2^2 = \left\| \begin{pmatrix} 3 & 2 & 2 & 1 \\ 1 & 3 & 1 & 2 \\ 2 & 2 & 3 & 1 \end{pmatrix} \right\|_2^2 = \sqrt{\sum_i \sum_j |a_{ij}|^2}^{\,2} \qquad \Big| \quad \cancel{\sqrt{\ }^{2}}$$

$$= \sum_i \sum_j |a_{ij}|^2 \xrightarrow[a_{ij}^2]{} \begin{pmatrix} 9 & 4 & 4 & 1 \\ 1 & 9 & 1 & 4 \\ 4 & 4 & 9 & 1 \end{pmatrix} \xrightarrow[sum]{} \underline{\underline{51}}$$

$$b)\ b = \|W_b\|_2^2 = \left\| \begin{pmatrix} 5 & 4 & 4 & 3 \\ 3 & 5 & 3 & 4 \\ 4 & 4 & 5 & 3 \end{pmatrix} \right\|_2^2 \xrightarrow[b_{ij}^2]{} \begin{pmatrix} 25 & 16 & 16 & 9 \\ 9 & 25 & 9 & 16 \\ 16 & 16 & 25 & 9 \end{pmatrix}$$

$$\xrightarrow[sum]{} \underline{\underline{191}}$$

→ Feststellung: $b > a$ da Elemente grösser