Lesson 101 of Deep Learning

ALEX

Outline

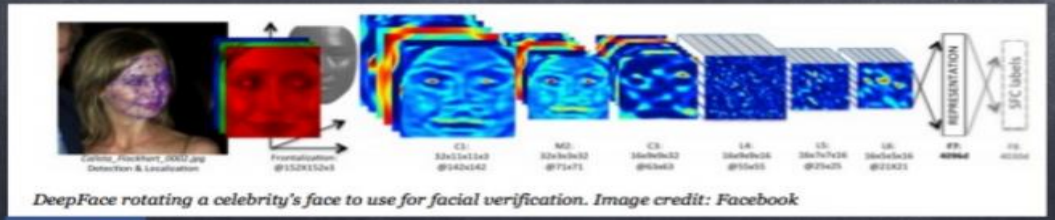
- What is Deep Learning?
- Neural Net, Support Vector Machine and Deep learning
- Convolutional Neural Net (CNN)
- Academic and Industrial applications of Deep Learning
- Available tools of Deep Learning
- Talks about Deep Learning

What is Deep Learning?

Wikipedia:

Deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions in data by using architectures composed of multiple non-linear transformations.

Example: Input data (image)



DeepFace rotating a celebrity's face to use for facial verification. Image credit: Facebook

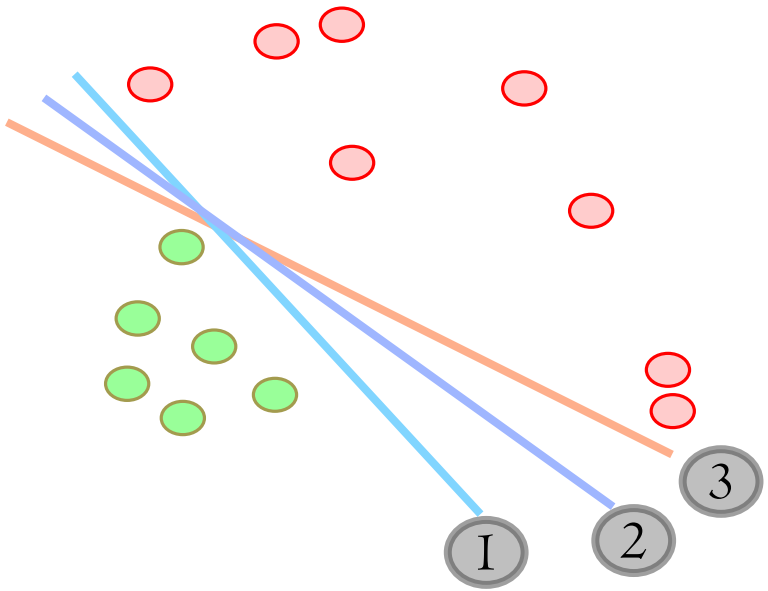
Prediction (who is it?)

Facebook's DeepFace (Yann LeCun) recognises faces as well as humans

If you make these networks deeper, it becomes easier for them to learn

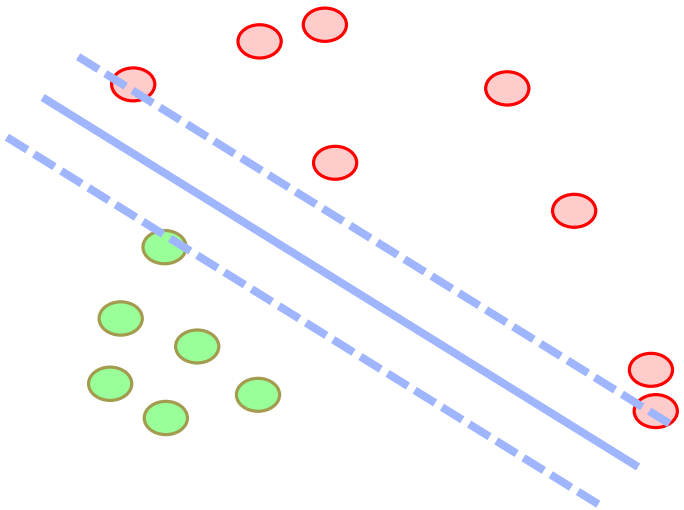
Alex Berg, researcher at the University of North Carolina

How do you classify data ?



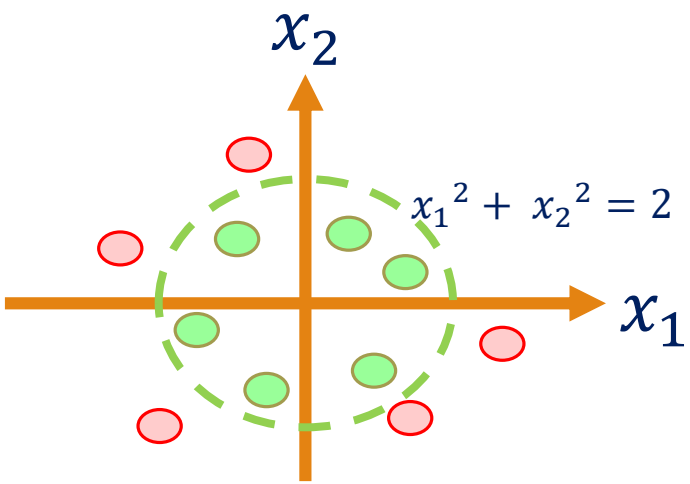
Perceptron Learning Algorithm

How do you classify data ?



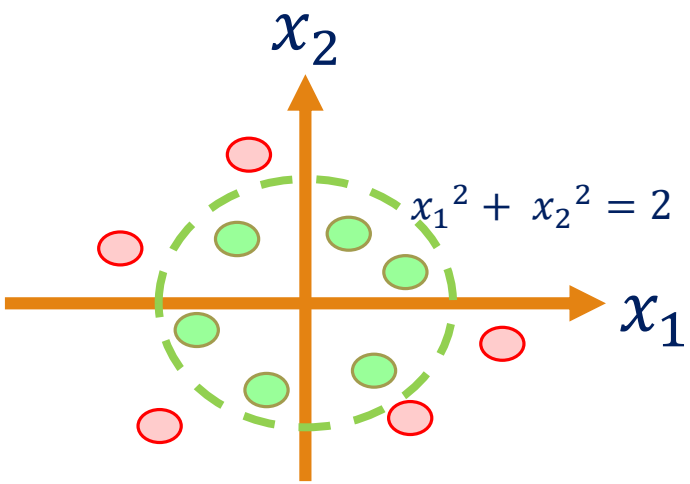
Support Vector Machine

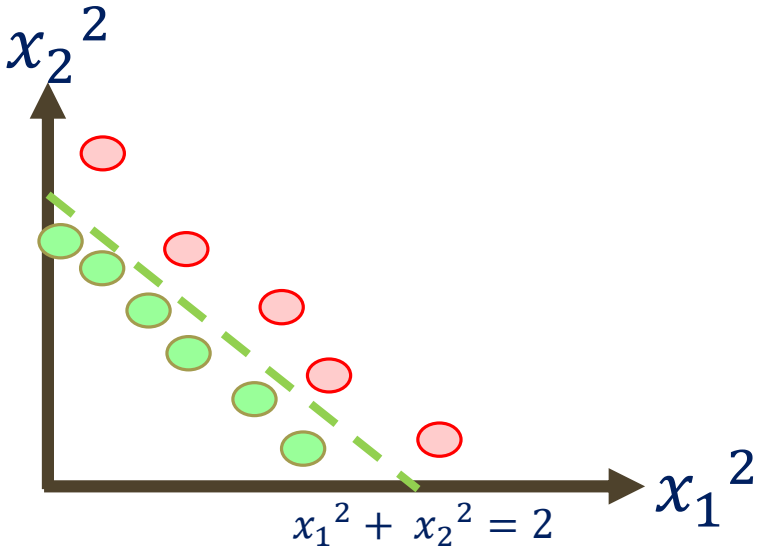
How do you classify data ?



Linearly Non-separable case

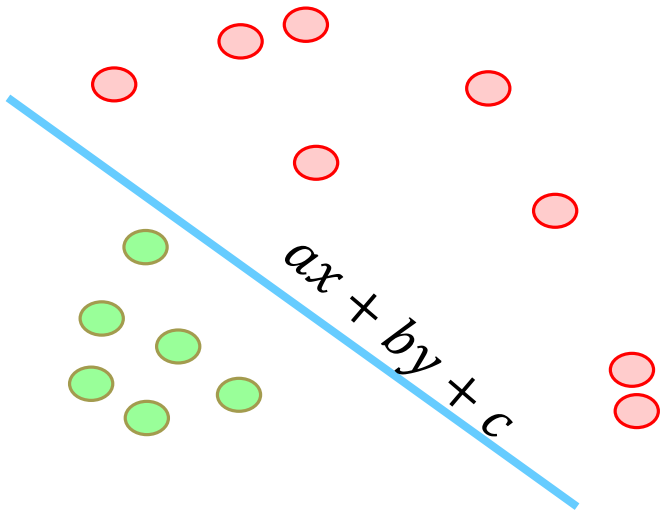
How do you classify data ? Lifting / Kernel

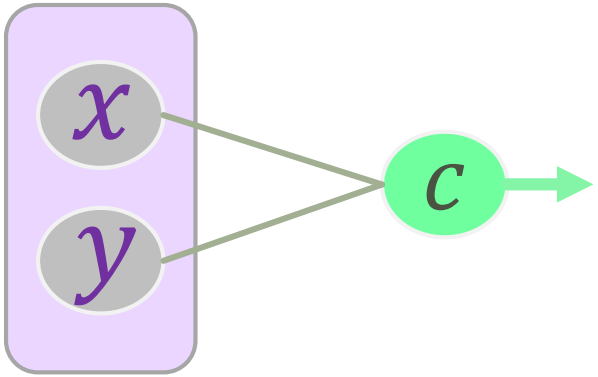




$$\varphi(x) := [x_1{}^2, x_2{}^2, x_1 x_2, x_1, x_2, 1]$$

Linearly Separable Case





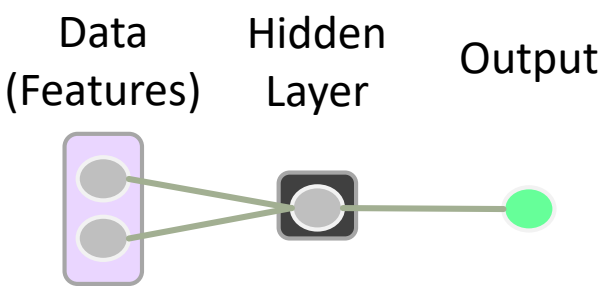
$Output = ax + by + c$

$ax + by + c$: Output

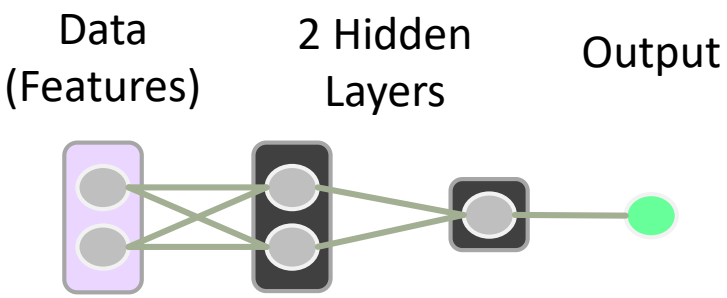
$ax + by + c > 0$: Positive Class

$ax + by + c < 0$: Negative Class

Stacked Perceptron

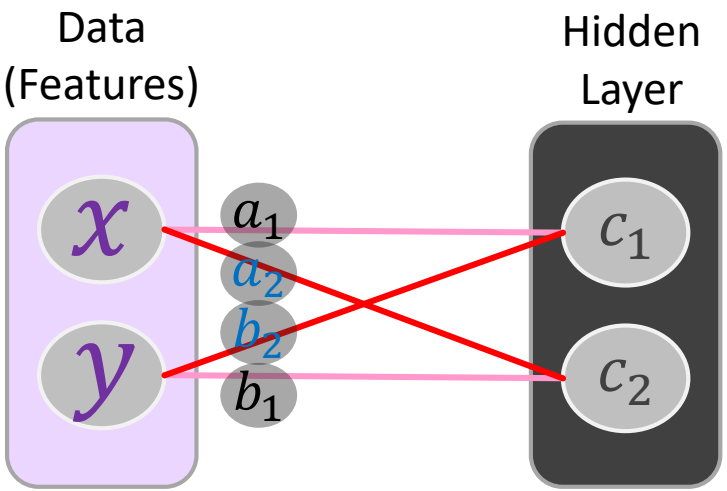


Single-Layer Perceptron



2-Layer Perceptron

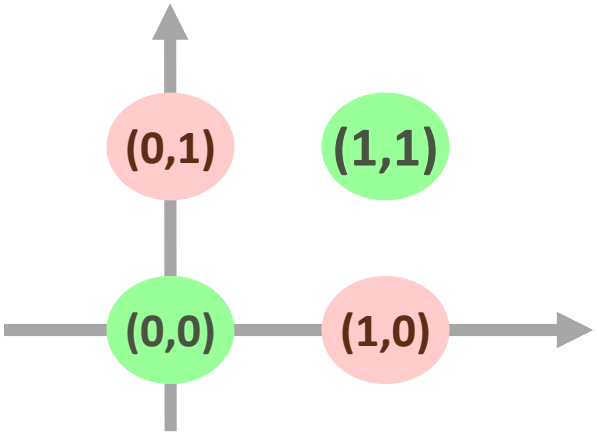
Perceptron Learning Algorithm

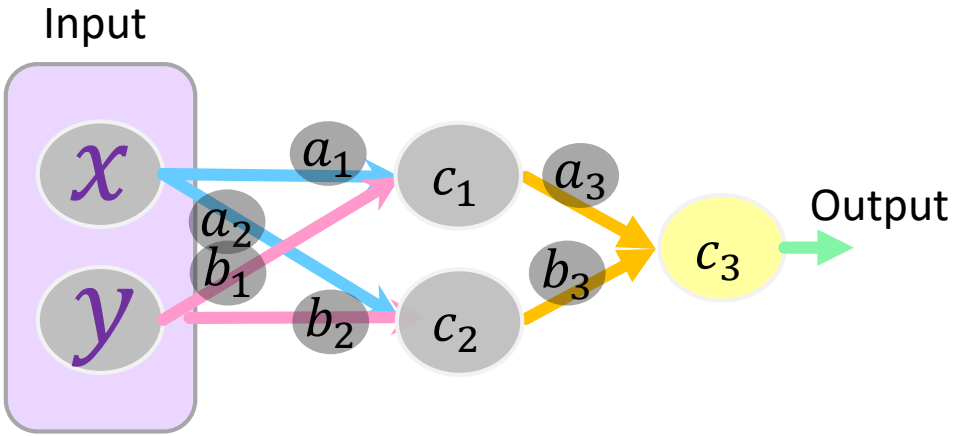


$$a_1 x + b_2 y + c_1$$

$$b_1 y + a_2 x + c_2$$

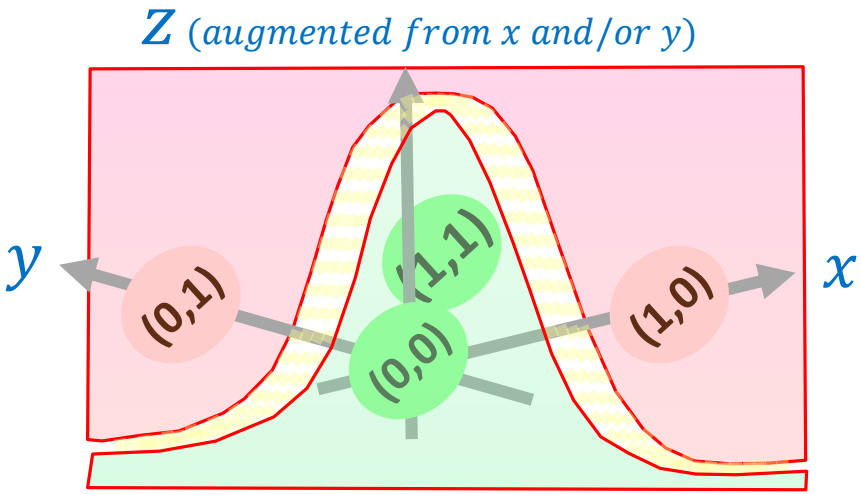
XOR Problem

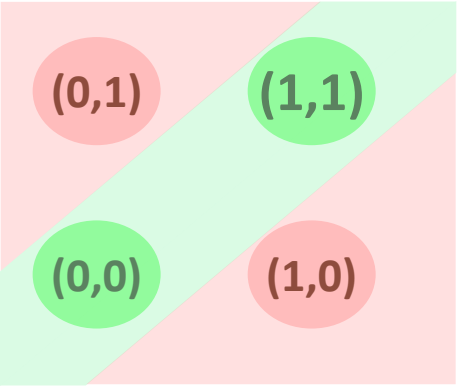




Most classification problem is not linearly separable

Kernel vs. Multi-Layer Perceptron



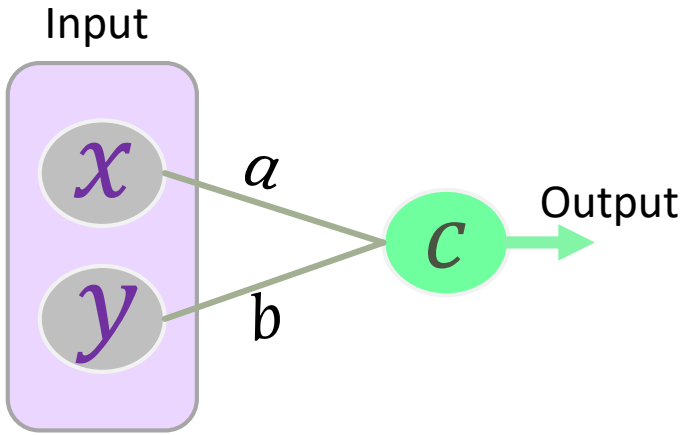


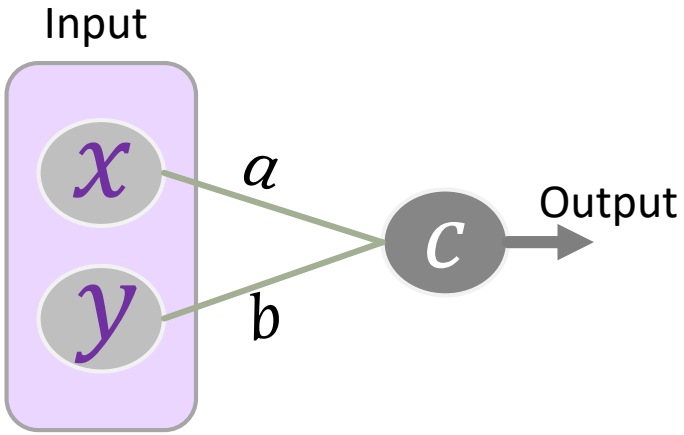
Lifting / Kernel
Another View Angle

Multi-Layer Perceptron
那你有沒有分兩次?

How to add non-linearity in perceptron learning?

Single-Layer Perceptron





$ax + by + c > 0$

Activated

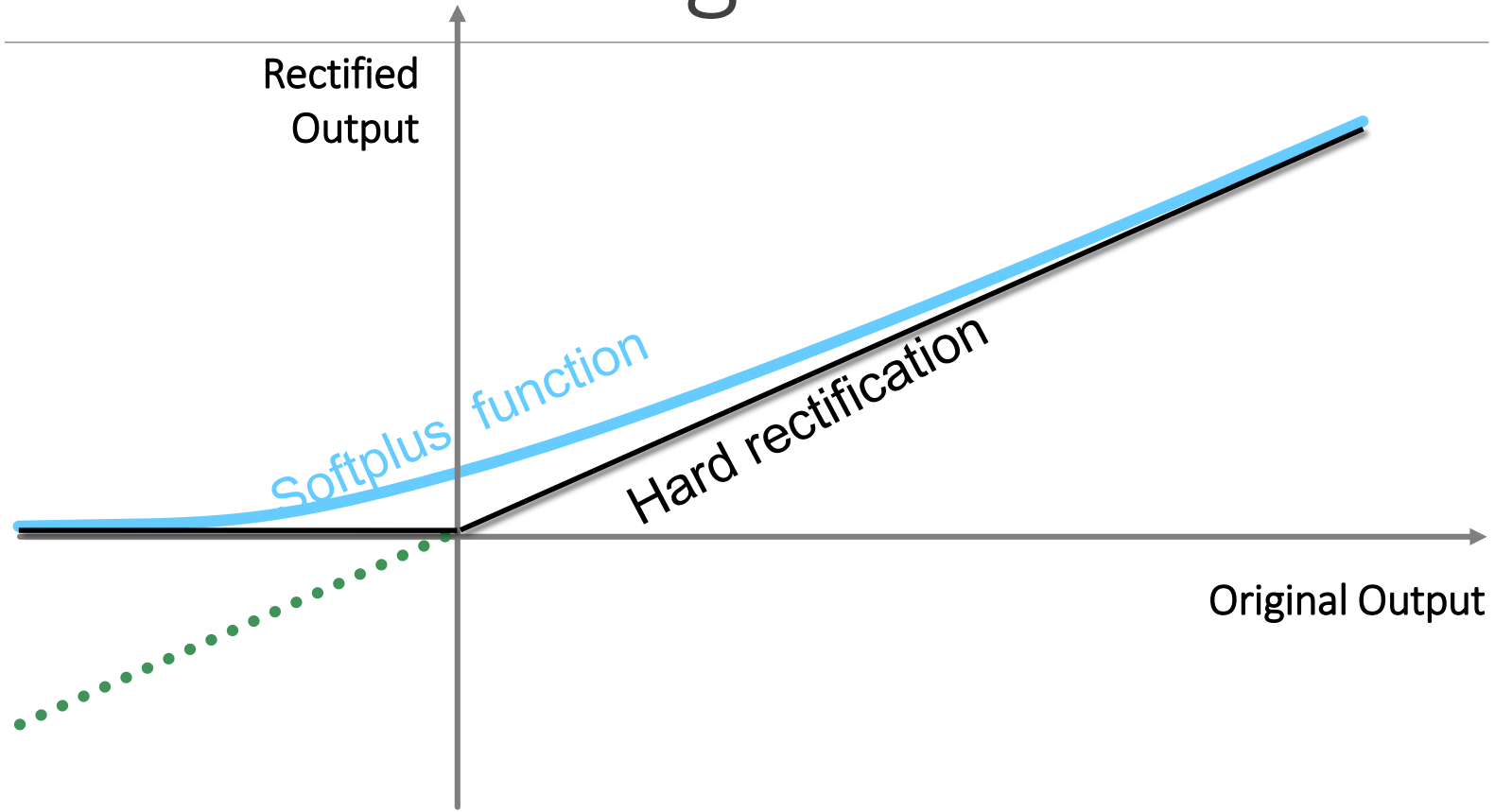
Output activation intensity

$ax + by + c < 0$

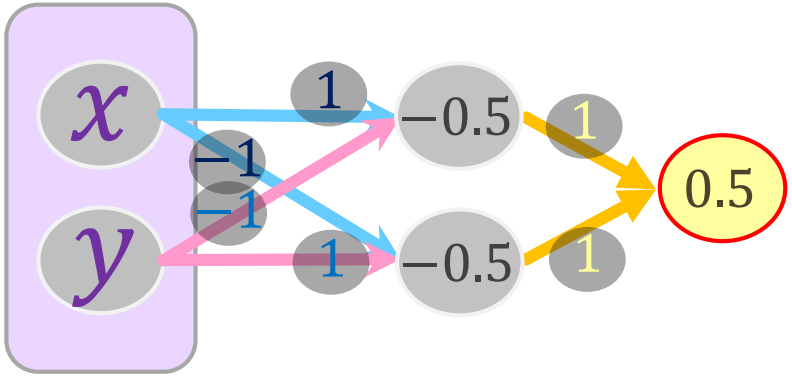
Inactivated

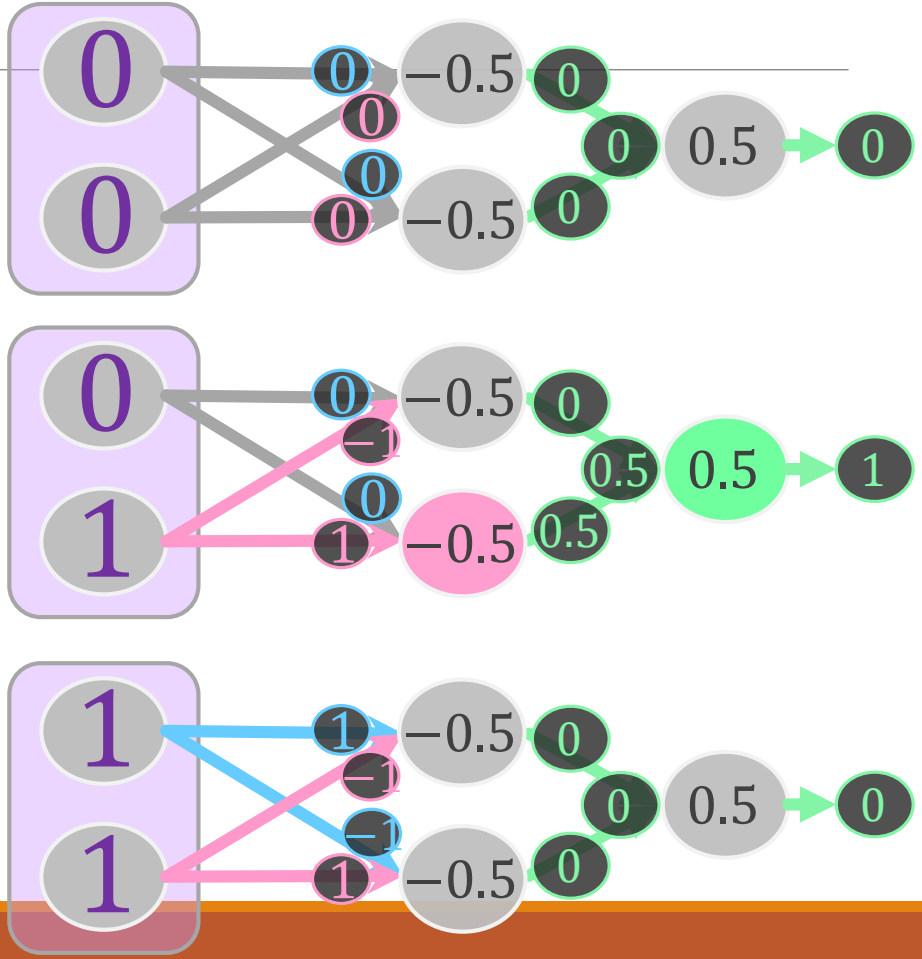
Output 0

Makes Learning Possible



XOR Problem

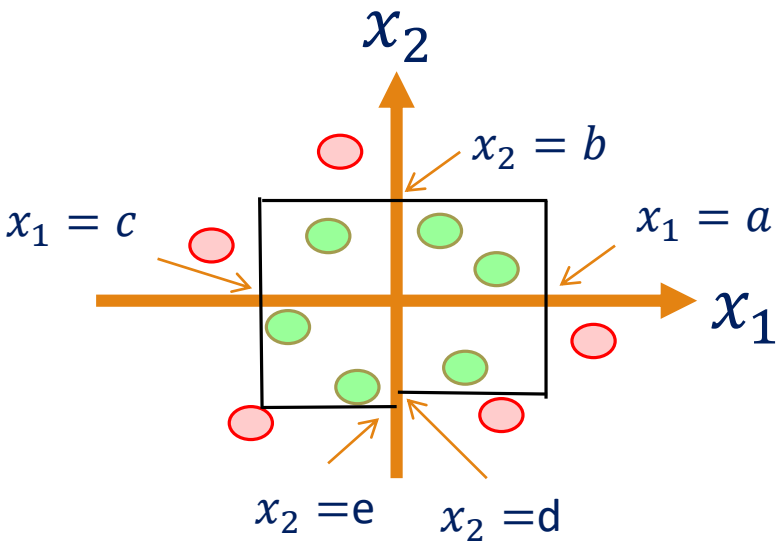

x
y
1
−1
−1
1
−0.5
−0.5
1
1
0.5

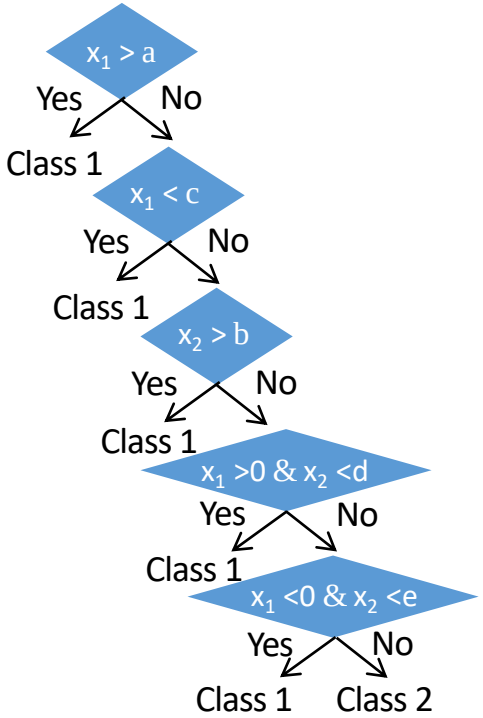

0
0
0
0
0
0
−0.5
−0.5
0
0
0
0
0.5
0
0
1
1
0
−1
0
1
−0.5
−0.5
0
0.5
0.5
0.5
1
1
1
1
−1
−1
1
−0.5
−0.5
0
0
0
0
0.5
0

Neural Net suffers from overfitting?

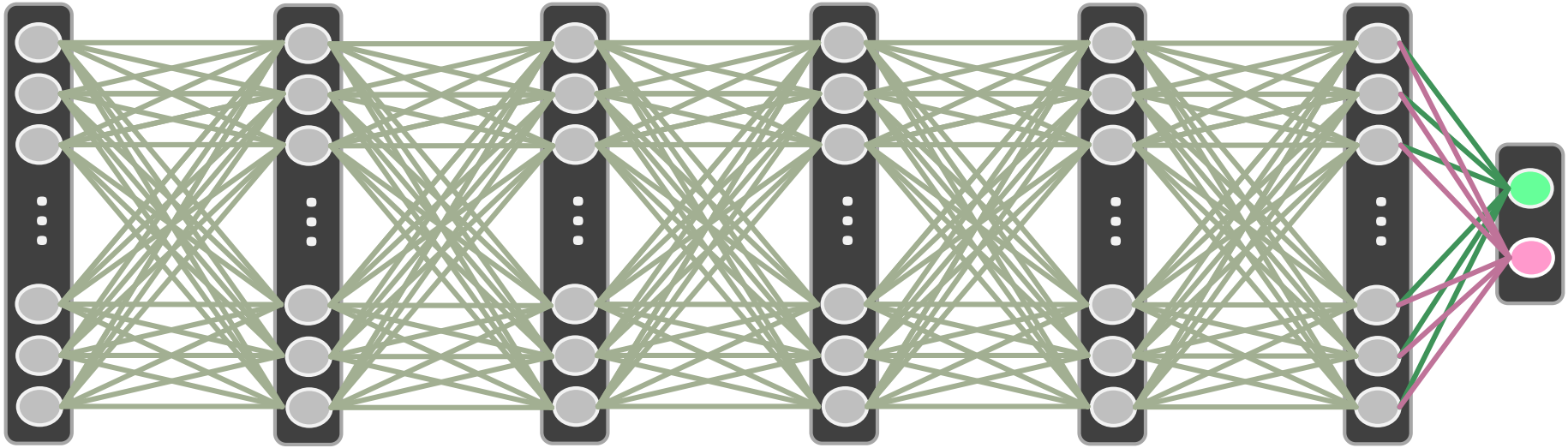
Sometimes , It is but there is another approach that could give you 0% classification error.

That approach is Decision Tree!





Restricted Boltzmann Machine



Low-Level Layers

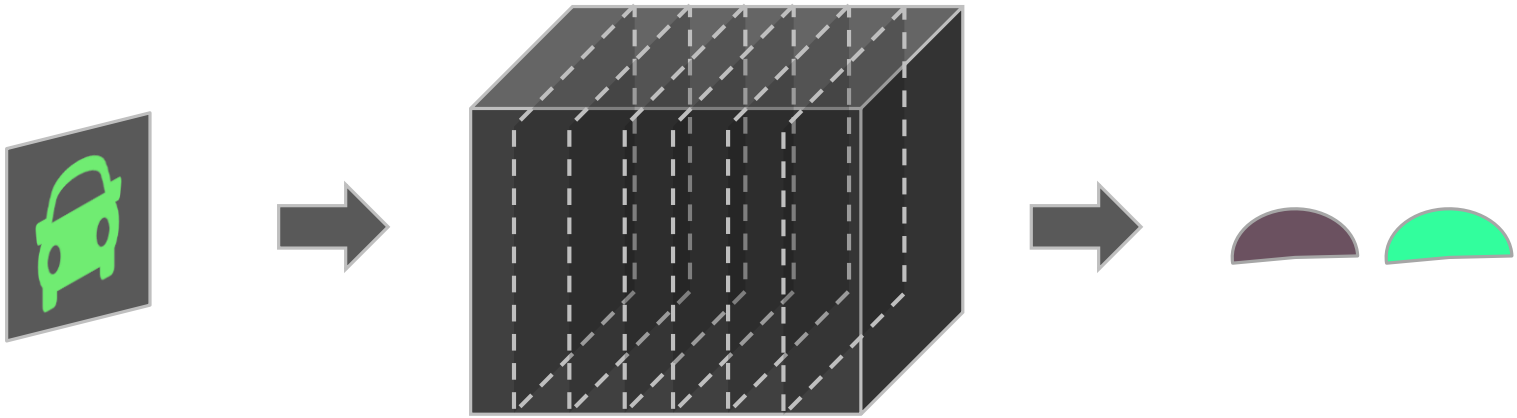
(Concrete Information)

High-Level Layers

(Abstract Concept)

Forward and Backward Computation

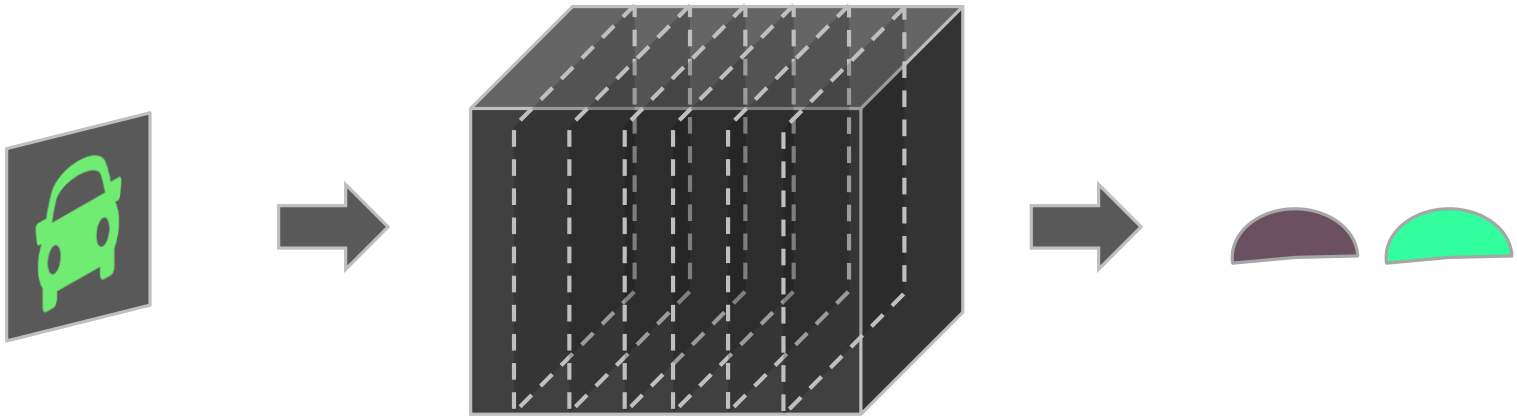
Forward : Feature Transformation to Top Layers



Backward : Parameter Adjustment for Bottom Layers

Forward and Backward Computation

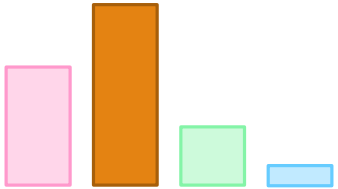
Forward : Inference, using $f_W(x)$



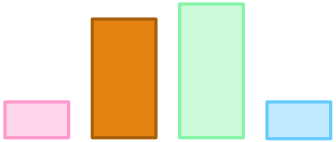
Backward : Learning, using $\nabla f_W(x)$

Loss

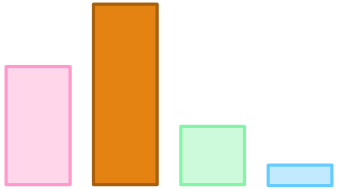
Loss is How Far the prediction is from our expectation



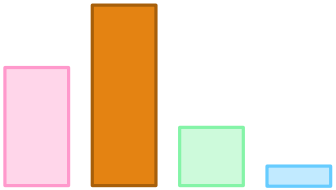
Your Expectation



Prediction



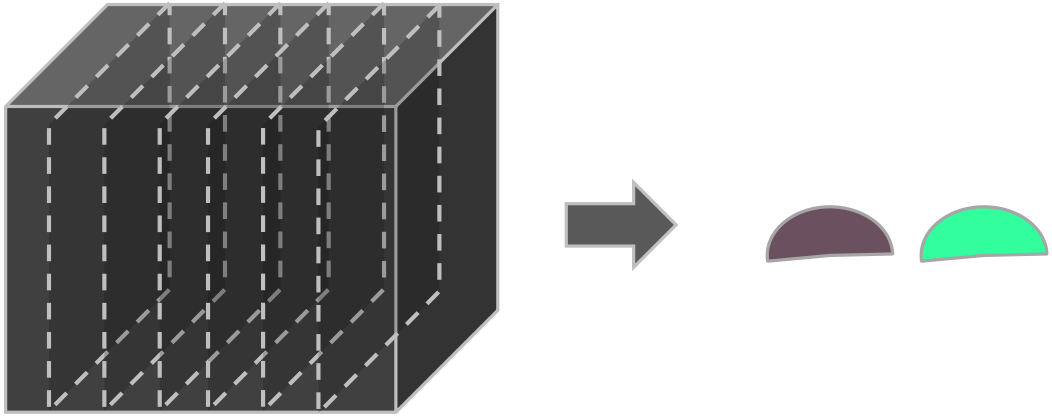
Your Expectation



Prediction

Gradient

To minimize loss, Gradient is essential



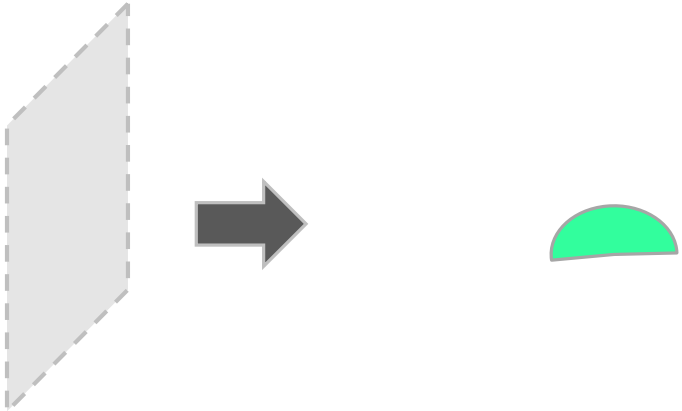
Black box

with many layers

Result

Gradient

To minimize loss, Gradient is essential



Last Hidden Layer

Partial Result

Gradient

-Gradient is the Fastest Direction to Minimize Loss (at present)

Each unit has a value

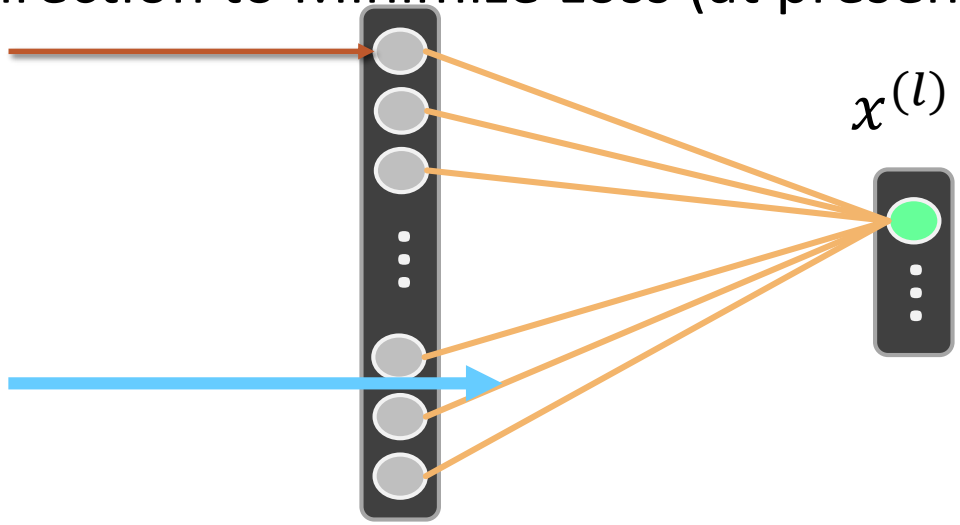
(from previous layers)

$x_i^{(l-1)}$

Each link is a weight

(to adjust)

$w_i^{(l-1,l)}$



Here, bias is neglected for simplification

Gradient

-Gradient is the Fastest Direction to Minimize Loss (at present)

Each unit has a value

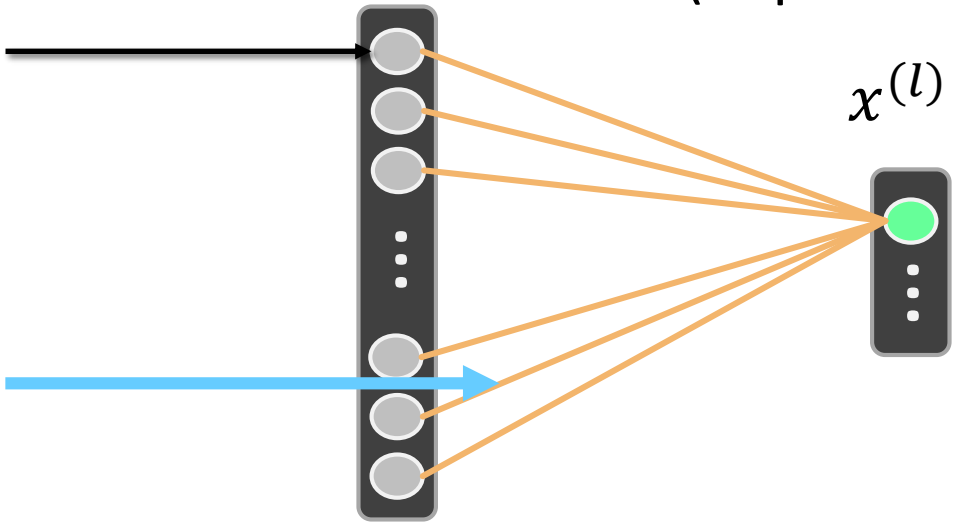
(from previous layers)

$x_i^{(l-1)}$

Each link is a weight

(to adjust)

$w_i^{(l-1,l)}$



$$x^{(l)} = x_1^{(l-1)} w_1^{(l-1,l)} + x_2^{(l-1)} w_2^{(l-1,l)} + \cdots + x_n^{(l-1)} w_n^{(l-1,l)}$$

Gradient

-Gradient is the Fastest Direction to Minimize Loss (at present)

$$x^{(l)} = x_1^{(l-1)} w_1^{(l-1,l)} + x_2^{(l-1)} w_2^{(l-1,l)} + \cdots + x_n^{(l-1)} w_n^{(l-1,l)}$$

What we can adjust

If we want to increase $x^{(l)}$

Gradient

-Gradient is the Fastest Direction to Minimize Loss (at present)

$$x^{(l)} = x_1^{(l-1)} w_1^{(l-1,l)} + x_2^{(l-1)} w_2^{(l-1,l)} + \cdots + x_n^{(l-1)} w_n^{(l-1,l)}$$

If we want to increase $x^{(l)}$

If $x_1^{(l-1)} > 0$

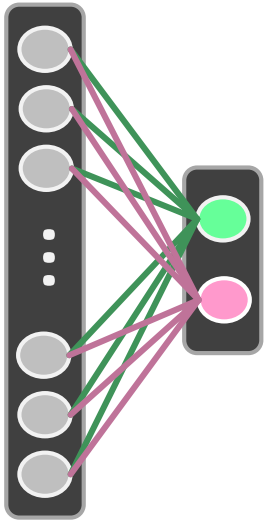
we increase $w_1^{(l-1,l)}$

If $x_1^{(l-1)} < 0$

we decrease $w_1^{(l-1,l)}$

Back Propagation

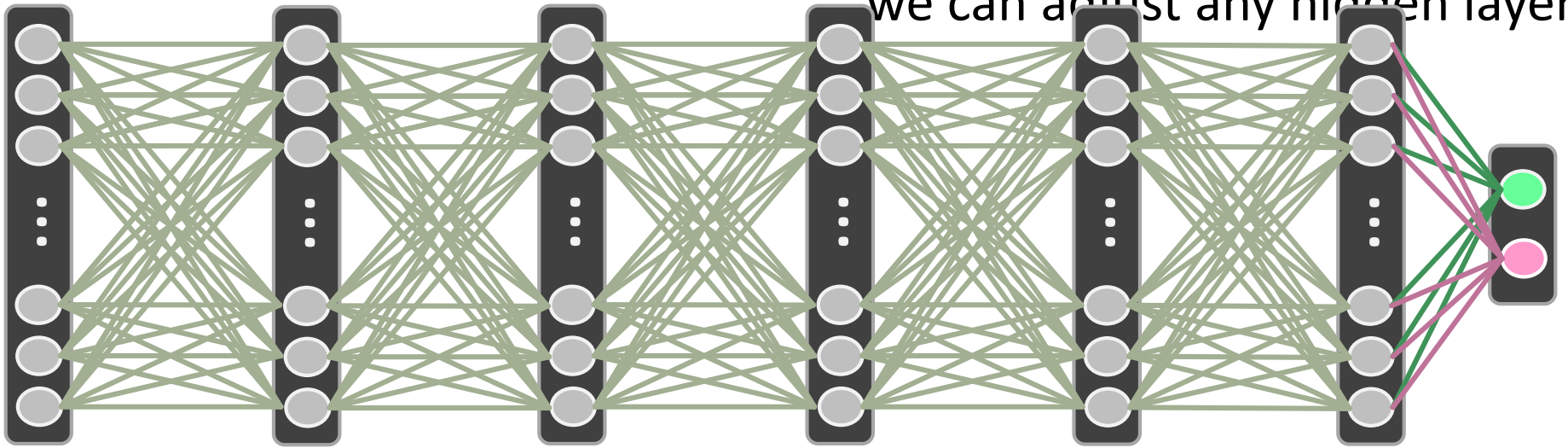
Once we can adjust the last hidden layer,



Back Propagation

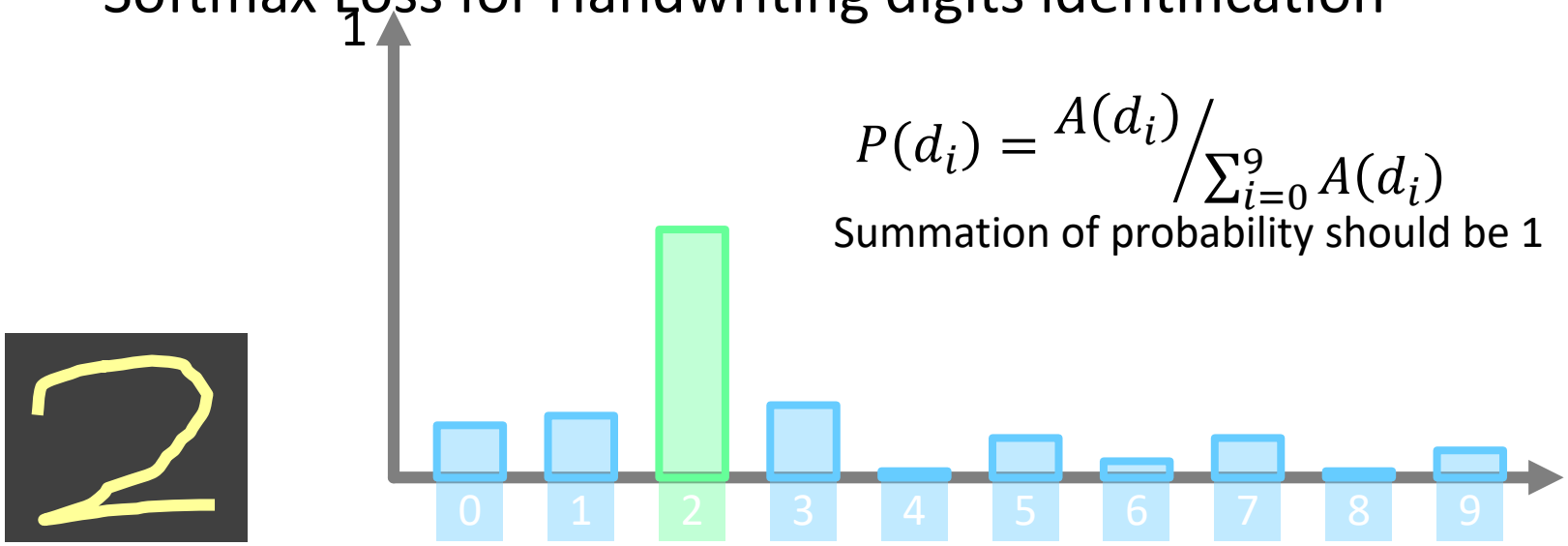
Once we can adjust the last hidden layer,

we can adjust any hidden layer



Softmax Loss

Softmax Loss for Handwriting digits identification



$$P(d_i) = A(d_i) \Big/ \sum_{i=0}^{9} A(d_i)$$

Summation of probability should be 1

Input

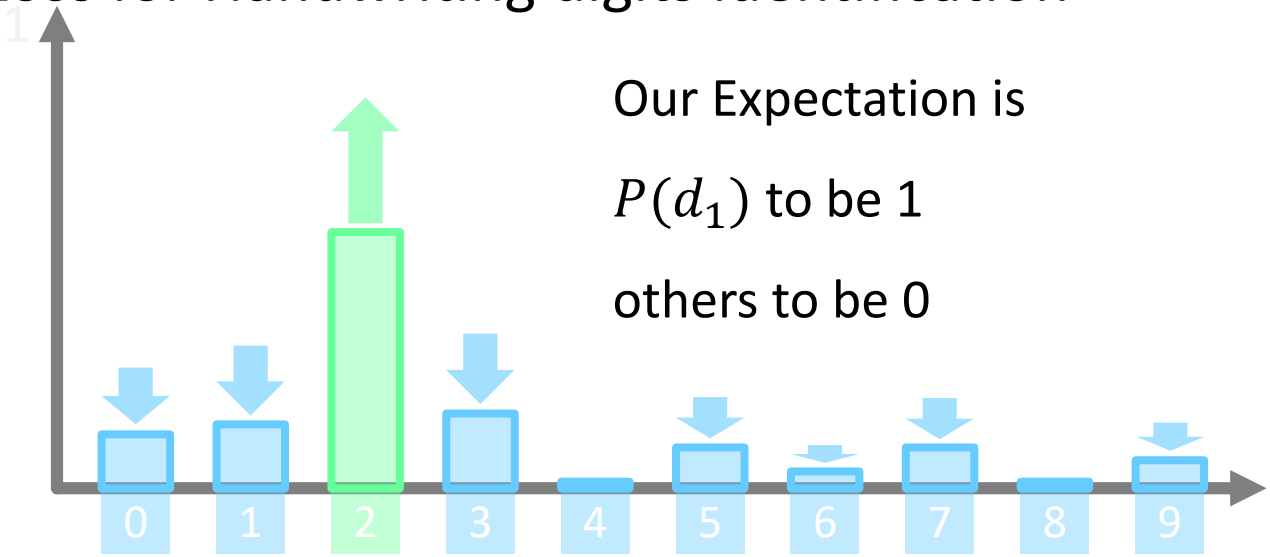
$P(d_0)P(d_1)$ … $P(d_9)$

Predicted probability of each digits

Softmax Loss

Softmax Loss for Handwriting digits identification

Input



Our Expectation is

$P(d_1)$ to be 1

others to be 0

$P(d_0)P(d_1)$ ... $P(d_9)$

Predicted probability of each digits

Winters of Neural Networks since 90's

- Non-convex
- Need a lot of tricks to play with
- Hard to do theoretical analysis

Deep learning Since 2006

materials are identical for all configurations. The blue bars in Fig. 1 summarize the measured SHG signals. For excitation of the *LC* resonance in Fig. 1A (horizontal incident polarization), we find an SHG signal that is 500 times above the noise level. As expected for SHG, this signal closely scales with the square of the incident power (Fig. 2A). The polarization of the SHG emission is nearly vertical (Fig. 2B). The small angle with respect to the vertical is due to deviations from perfect mirror symmetry of the SRRs (see electron micrographs in Fig. 1). Small detuning of the *LC* resonance toward smaller wavelength (i.e., to 1.3-μm wavelength) reduces the SHG signal strength from 100% to 20%. For excitation of the Mie resonance with vertical incident polarization in Fig. 1D, we find a small signal just above the noise level. For excitation of the Mie resonance with horizontal incident polarization in Fig. 1C, a small but significant SHG emission is found, which is again po-

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

Neural networks are coming back!

Image Classification

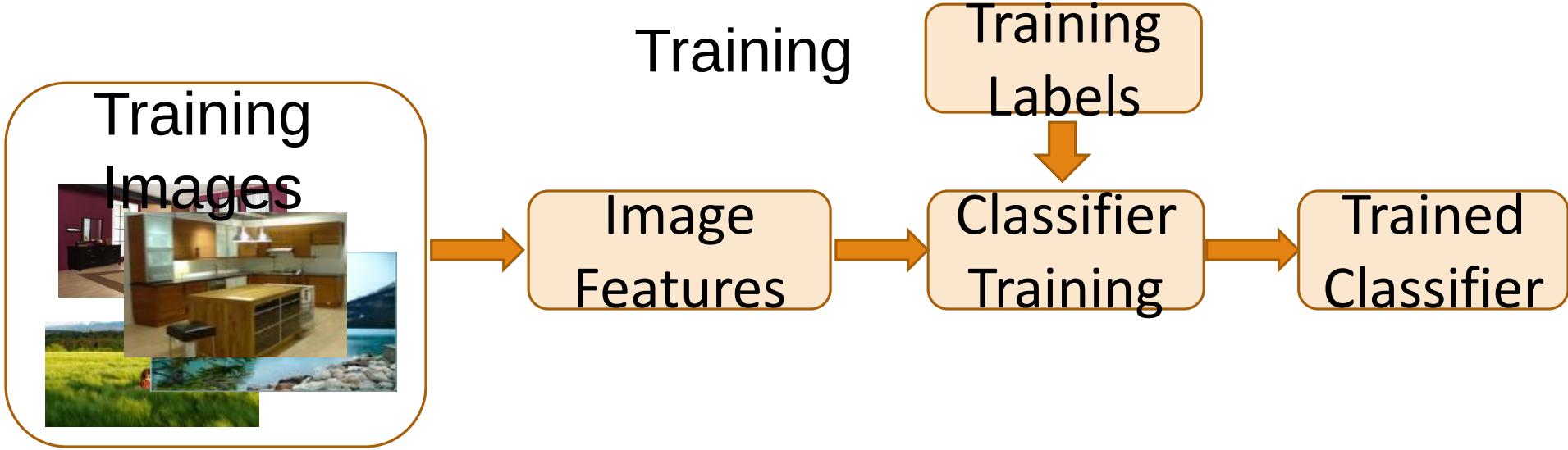


Image Classification

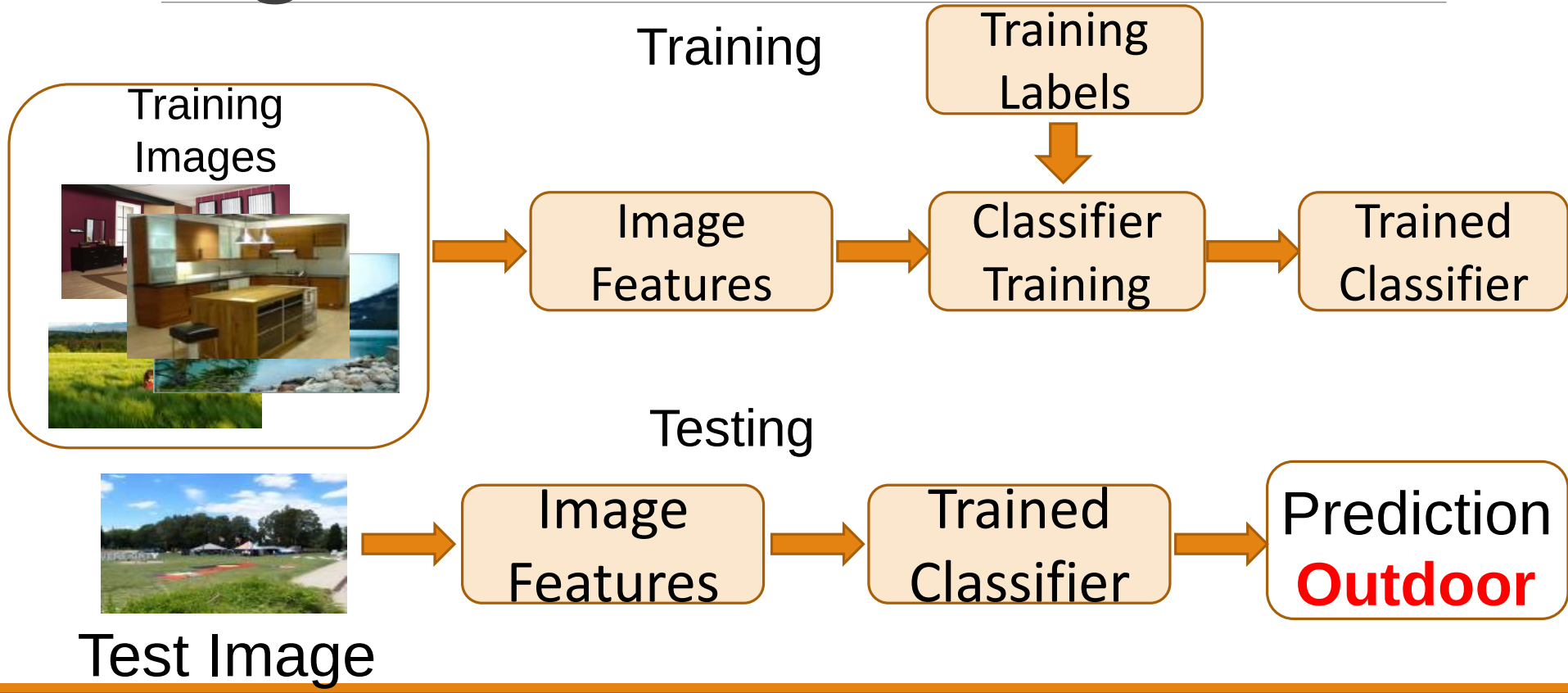


Image Classification

Training Labels

Image Features

Classifier Training

Trained Classifier

- Manually feature design (e.g., SIFT, HOG, etc.)
- Multiple separately learned layers
  (e.g., K-means, Pyramid, etc.)

Comparison

Classical CV

Hand-crafted Feature Extractor → "Simple" Trainable Classifier →

Deep Learning

Trainable Feature Extractor → **Trainable** Classifier →

End-to-End Learning

SIFT
HoG

K-means
Sparse Coding

Pooling

Classifier

Fixed

Unsupervised

supervised

2012

Low-level
Features

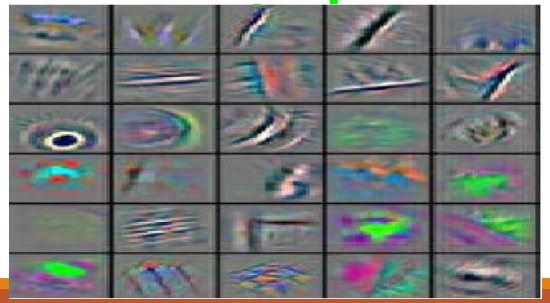
Mid-level
Features

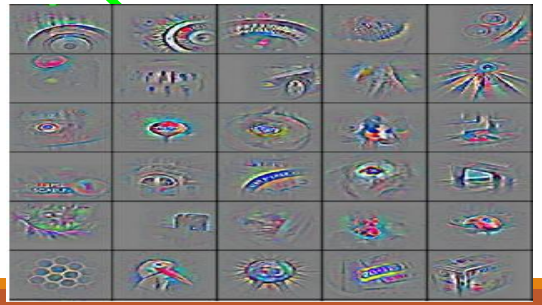
Low-Level
Feature

Mid-Level
Feature

High-Level
Feature

Trainable
Classifier





IMAGENET

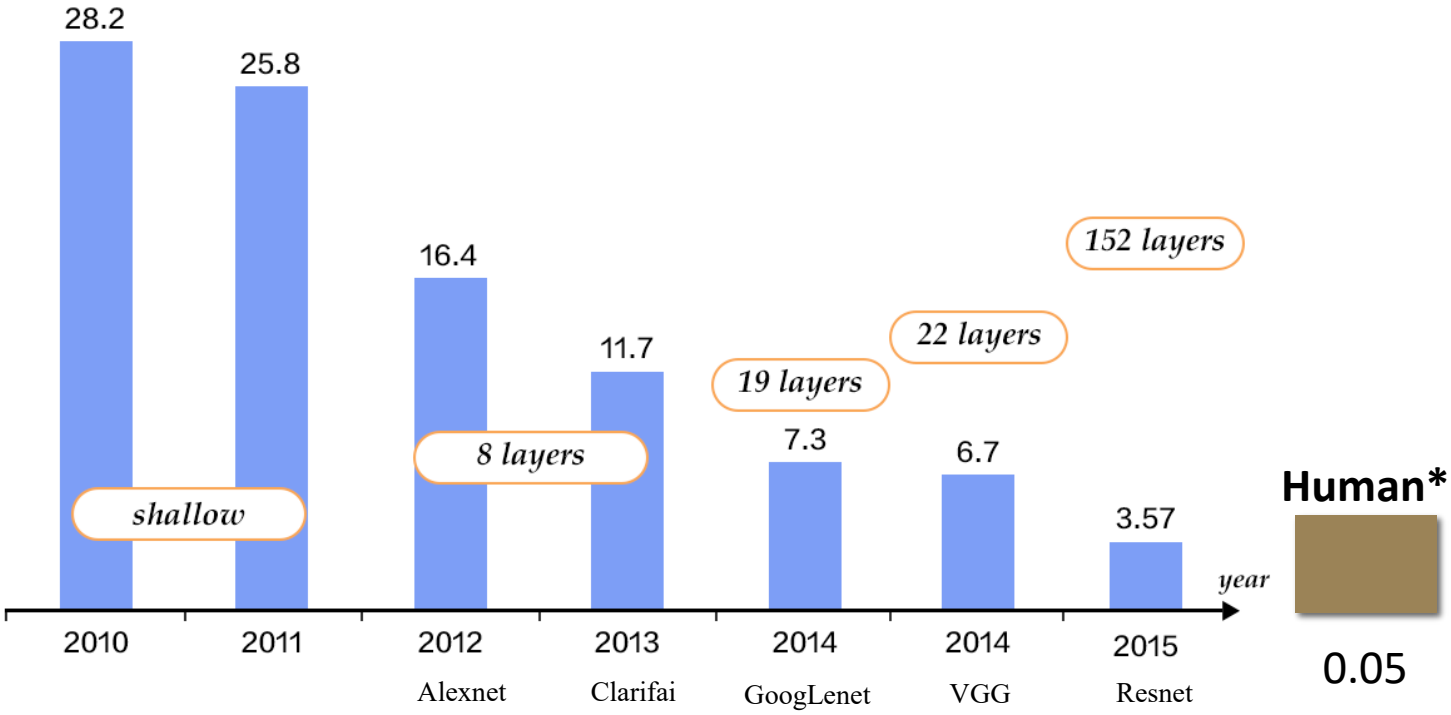
- Start in 2007 at Princeton
- Debut in CVPR'2009
- Stop in 2010
  - ➢Total classes：22K
  - ➢Total images：15M
- ILSVR Challenge: 2010 - present

Jia Deng

Fei-Fei Li

Info from http://www.image-net.org/

1K Image Classification (Top-5 Error)

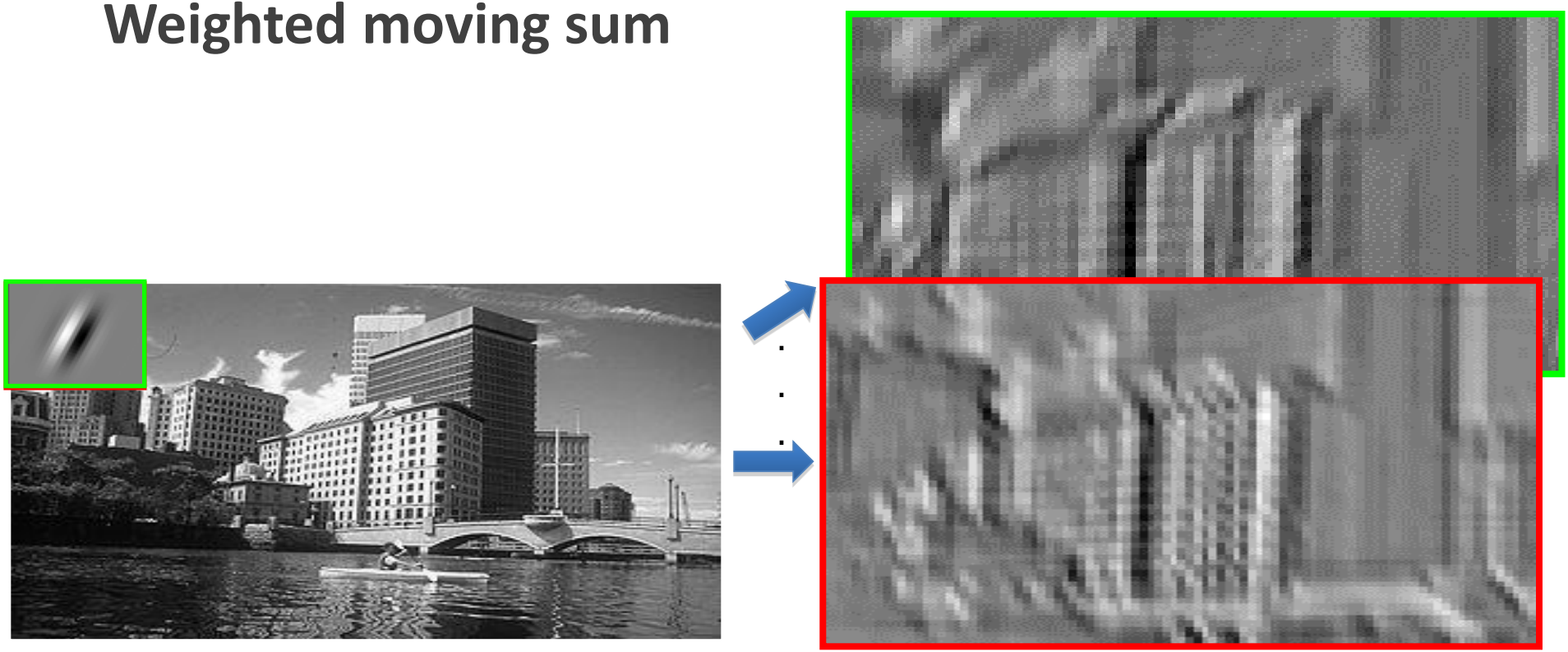


2016?
3.08%

Inception-v4 +
Inception-Resnet-v2

What is a Convolution?

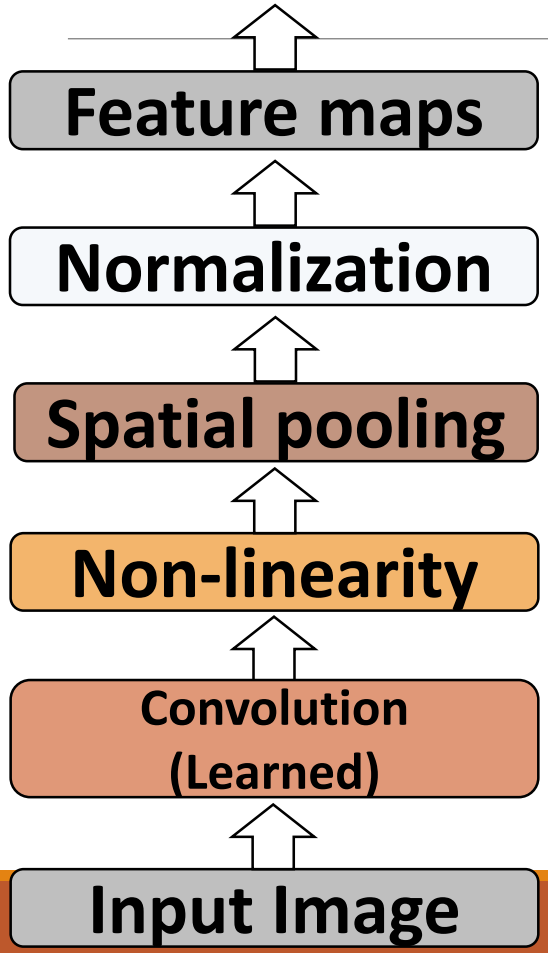
Weighted moving sum



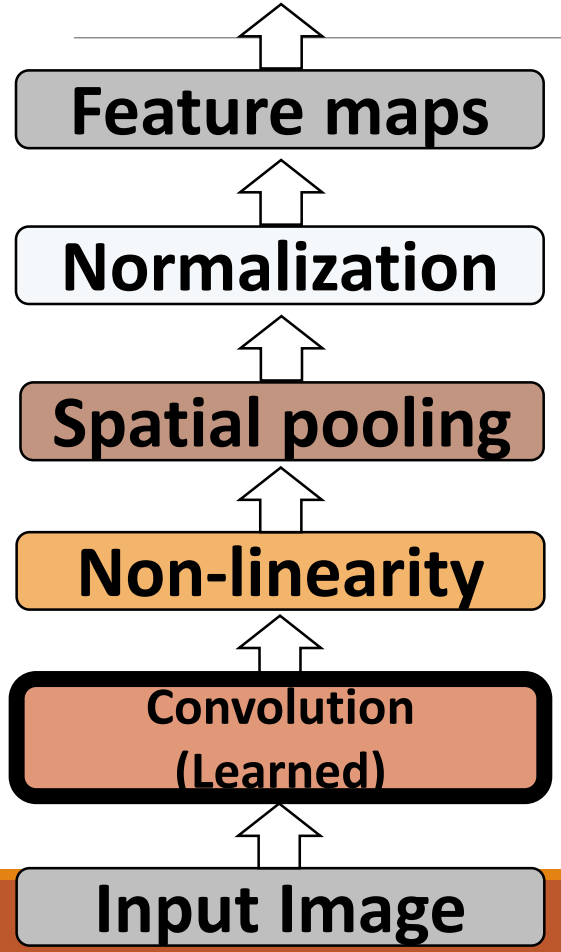
Input

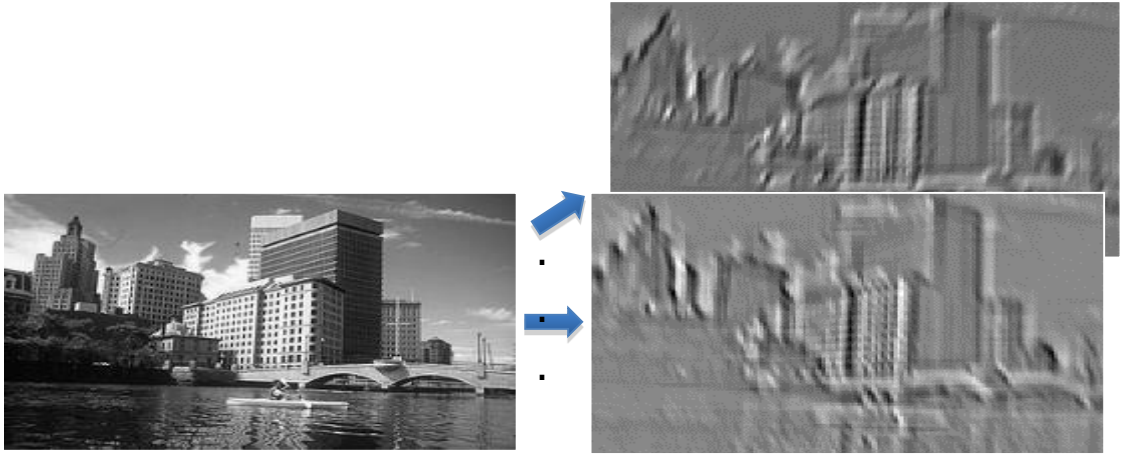
Feature Activation Map

Convolutional Neural Networks



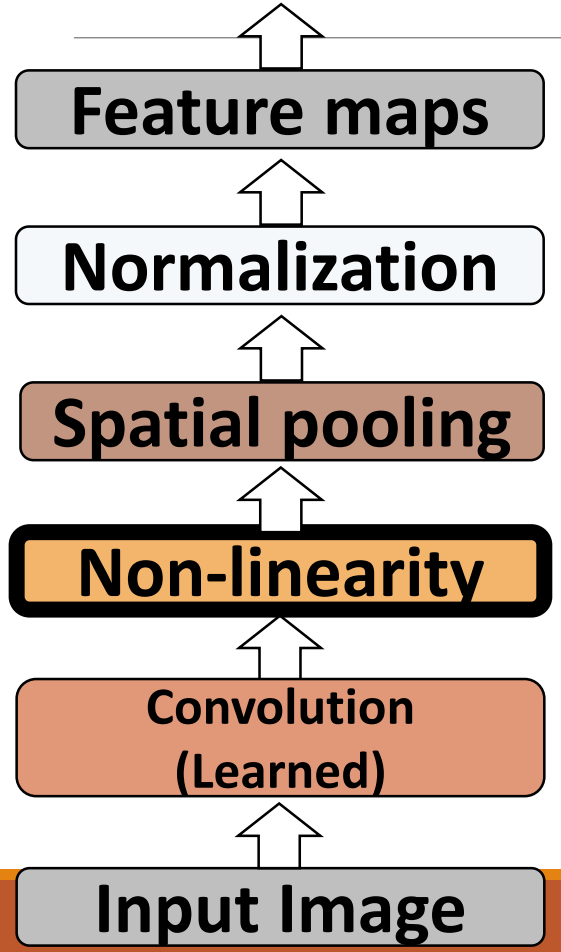
Convolutional Neural Networks



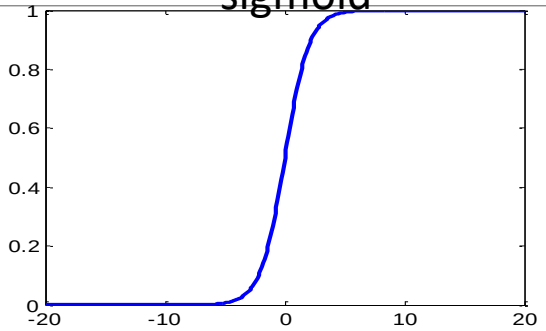


Input Feature Map

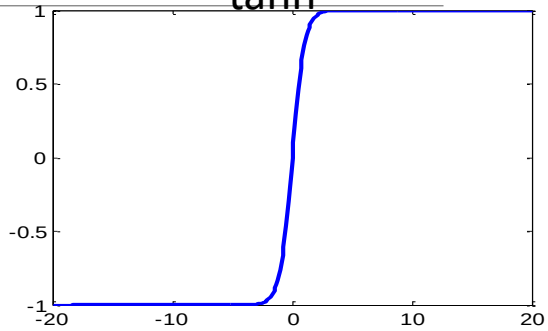
Convolutional Neural Networks



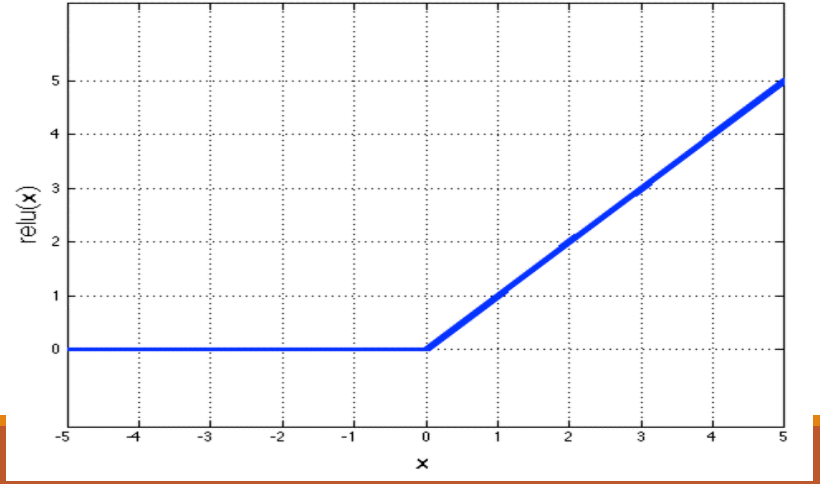
sigmoid



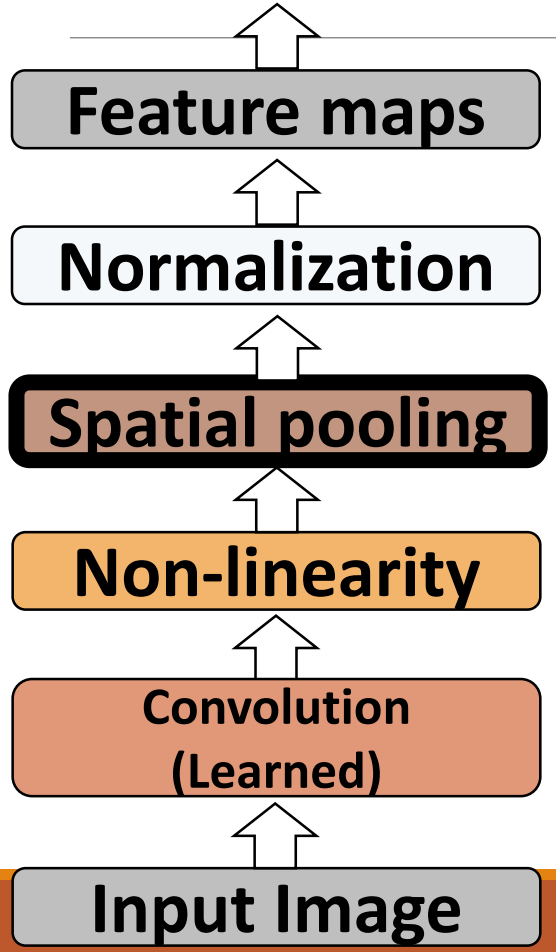
tanh

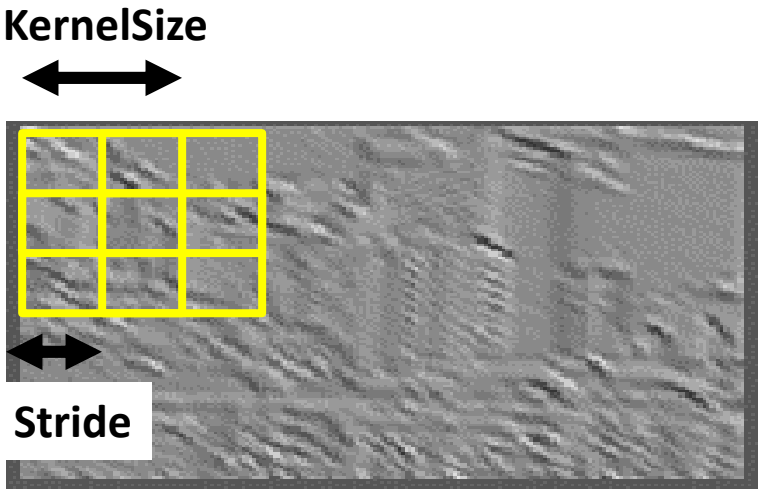


Rectified Linear Unit (ReLU)

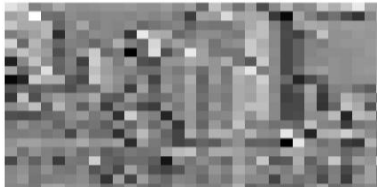


Convolutional Neural Networks





Max pooling

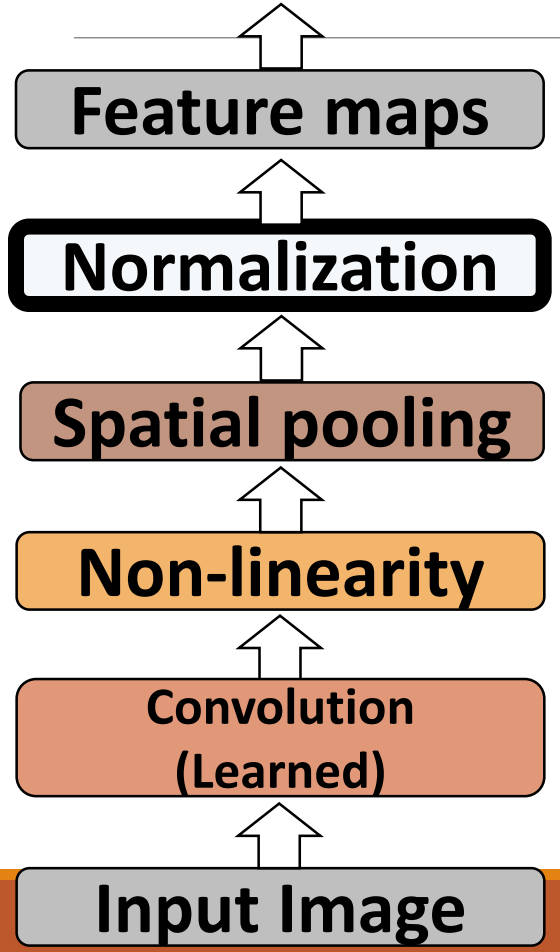


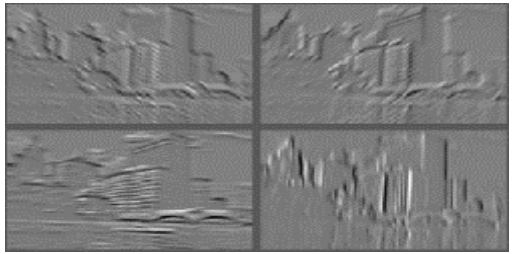
Max-pooling:

a non-linear down-sampling

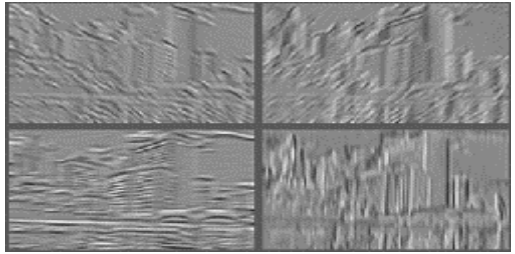
Provide *translation invariance*

Convolutional Neural Networks





Feature Maps

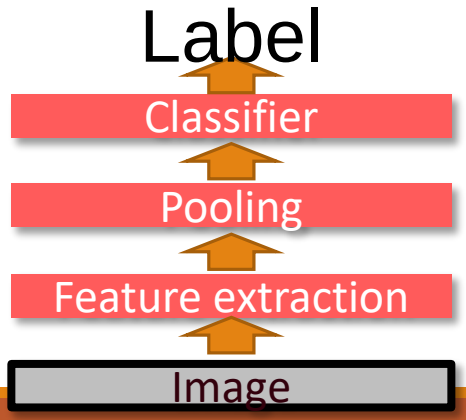


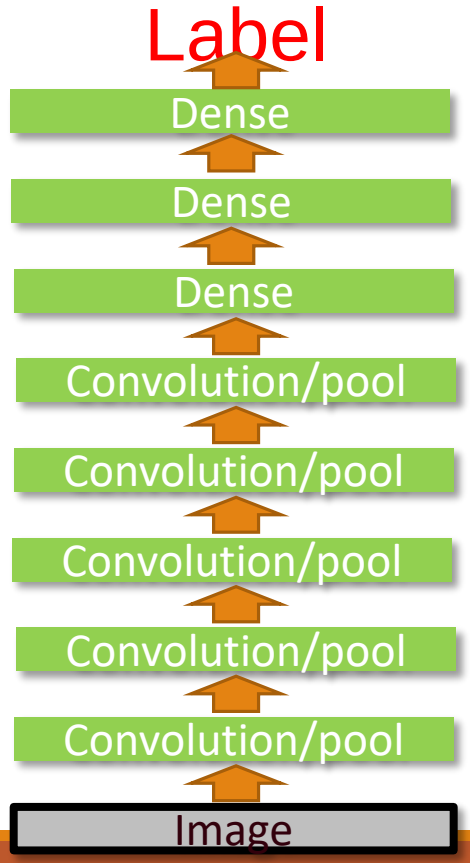
Feature Maps After Contrast Normalization

Equalizes the feature maps locally: weak signals are boosted, whereas strong signals are suppressed.

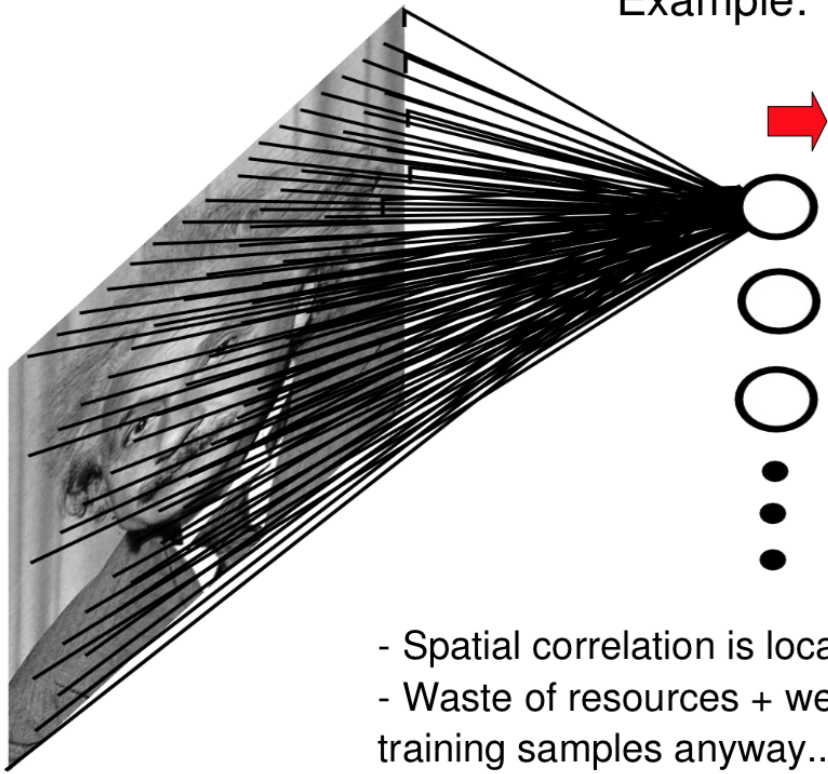
Engineered vs. learned features

Convolutional filters are trained in a supervised manner by back-propagating classification error





Fully Connected Layer



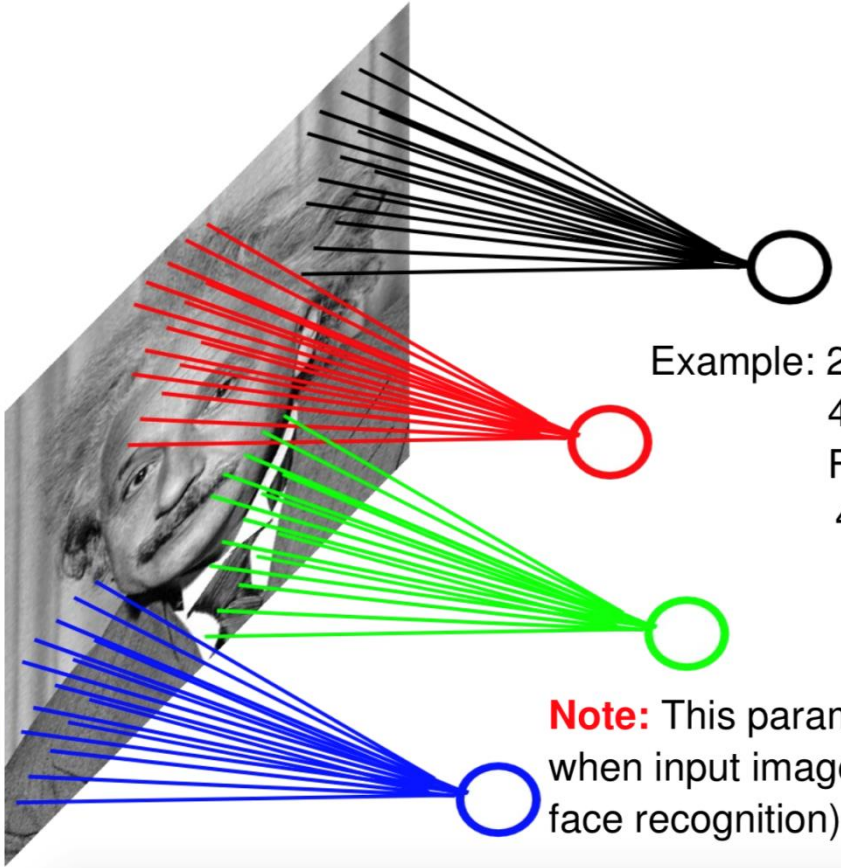
Example: 200x200 image
40K hidden units
~2B parameters!!!

- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

33

Ranzato

Locally Connected Layer



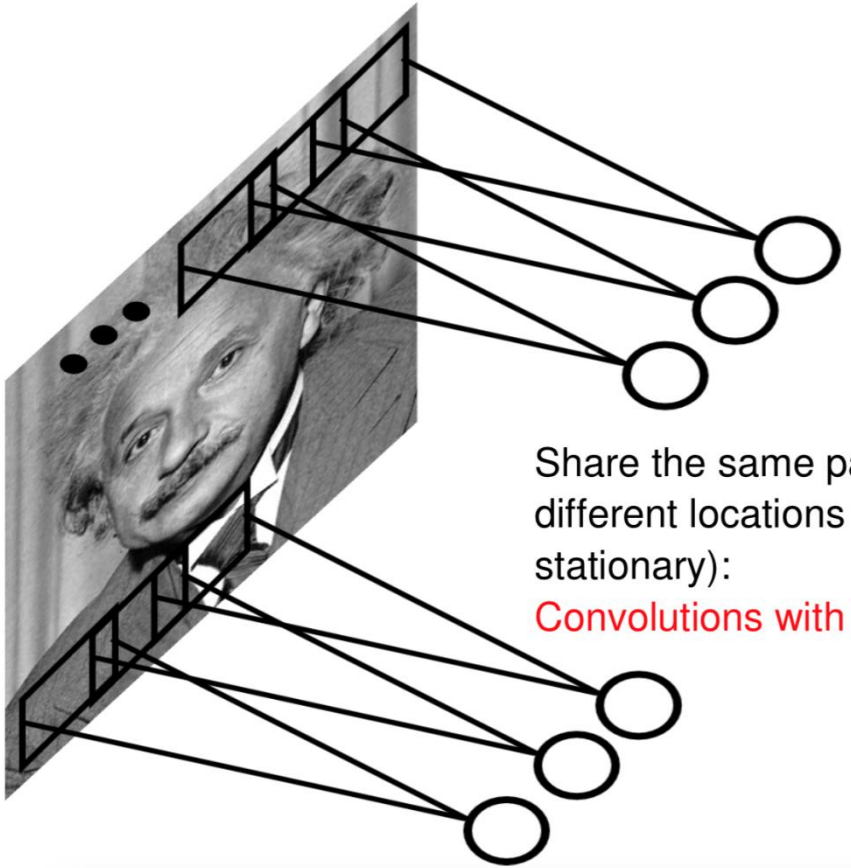
Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

34

Ranzato

Convolutional Layer



Assuming 10 stride without overlapping
(200/10)x(200/10)=400 hidden units for a 10*10 Filter.
This feature map share the same parameters
Parameters for 1 feature map = 100

Assuming 100 filters,
parameters for 100 feature map
=> 100*100=10k

Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

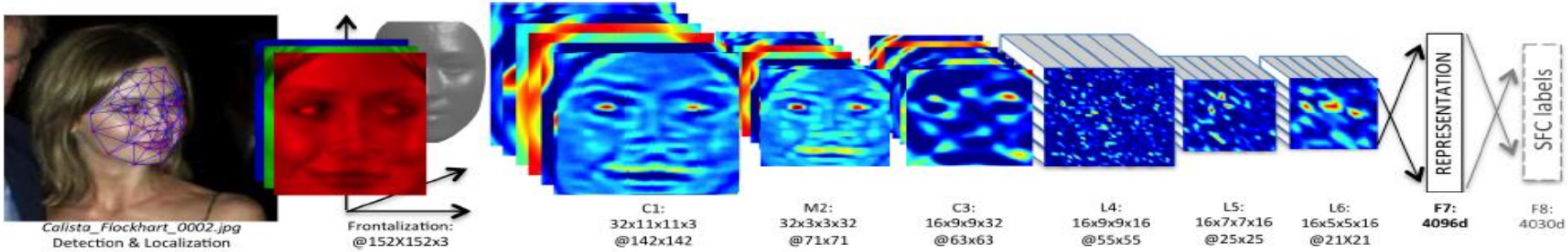
2B->4M->10K

36

Ranzato

The importance of the “size” of dataset

- Deepface achieves 0.975 accuracy in the Labeled Faces in the Wild database (LFW) which contains more than 13,000 images of faces collected from the web.
- However, Social Face Classification (SFC) dataset which is used for training is not released.
- The SFC dataset includes 4.4 million labeled faces from 4,030 people each with 800 to 1200 faces.

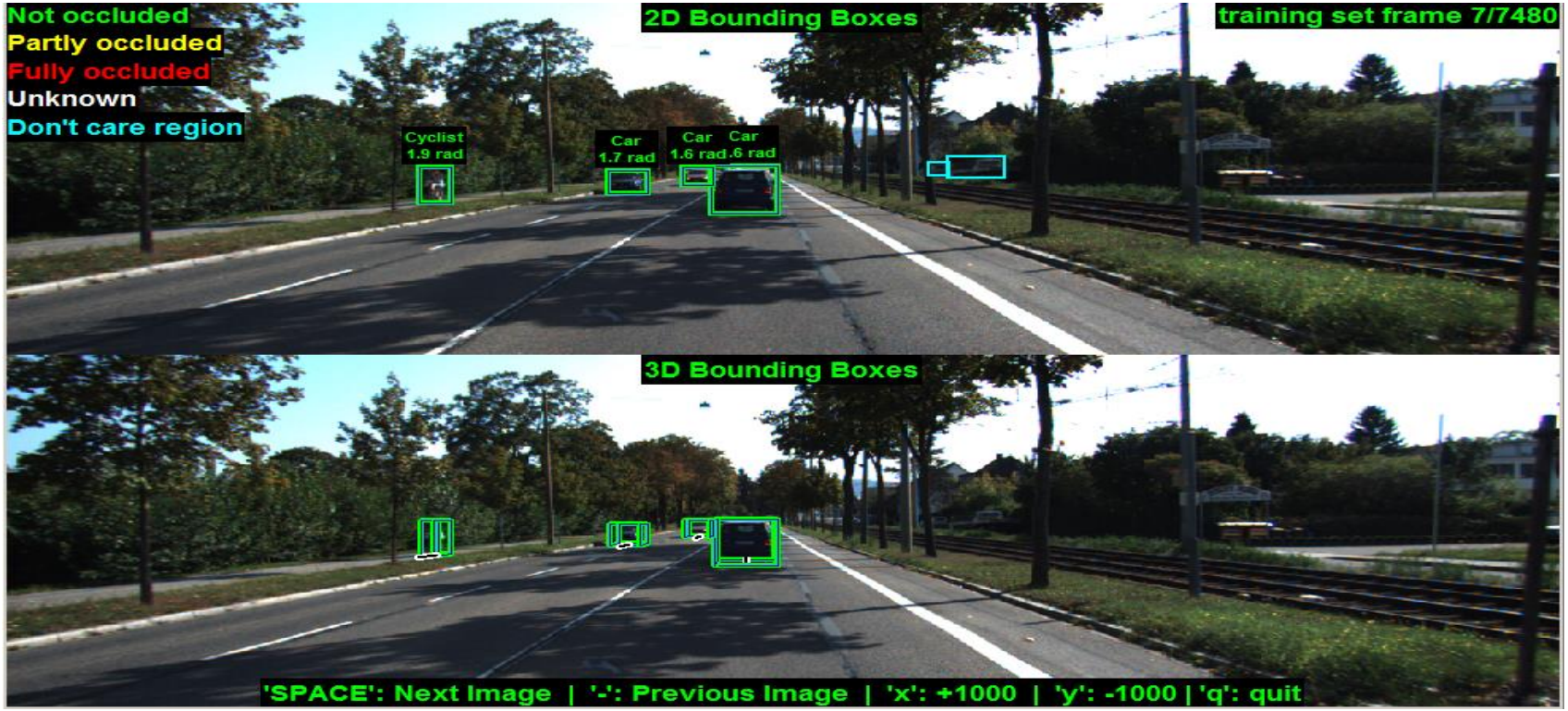


Public datasets for ADAS

KITTI:

- 7481 training images and 7518 test images
- 80256 labeled objects
- Including vehicle viewed from different perspective

KITTI example 1

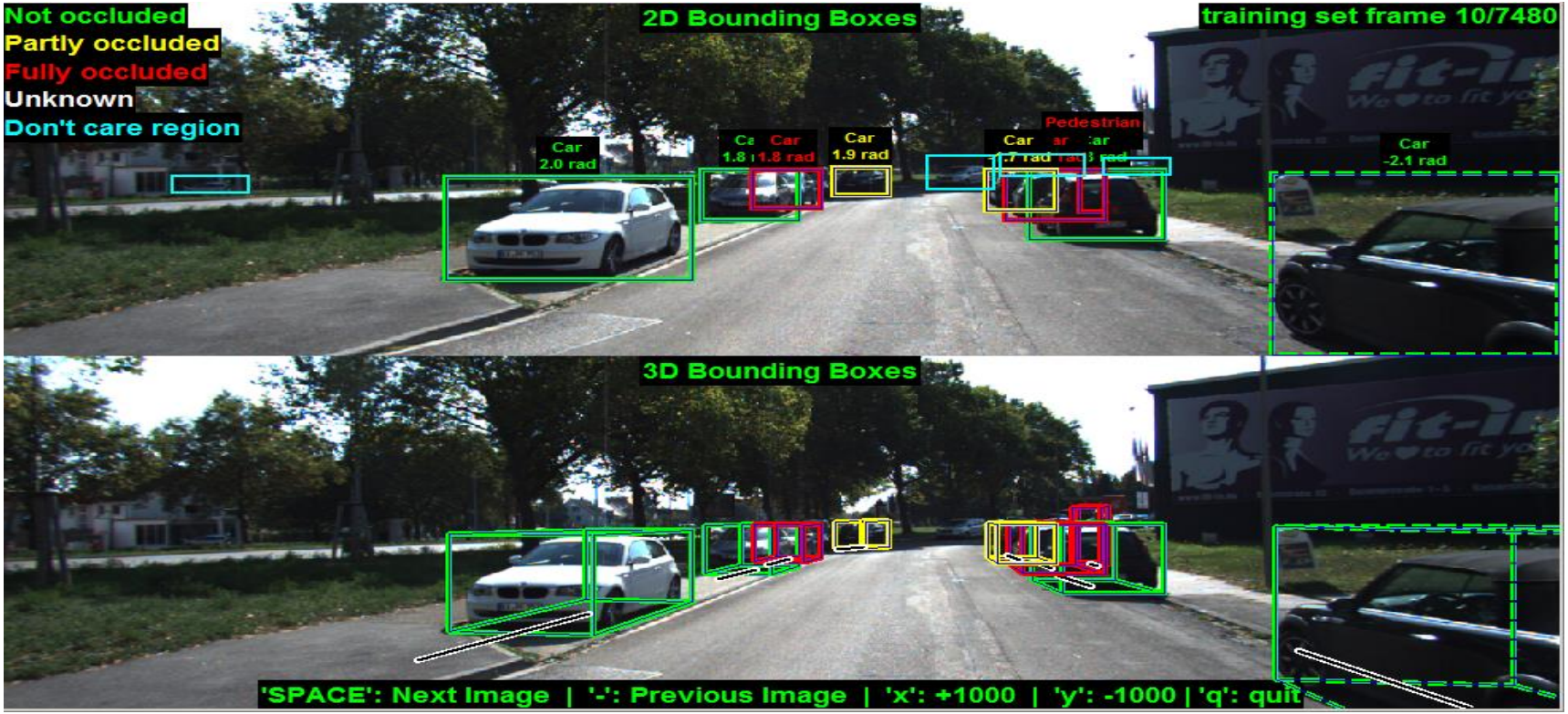


53

KITTI example 2

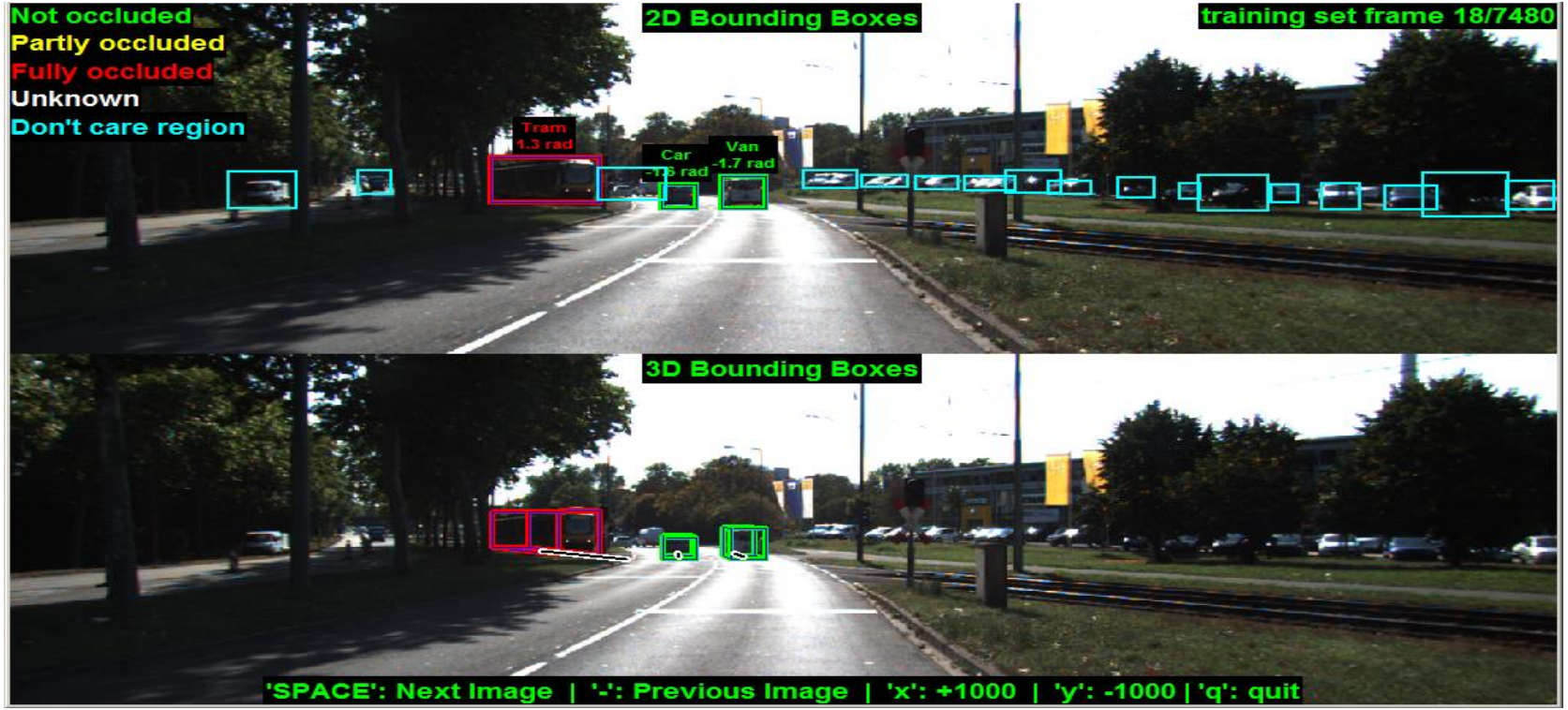
54

KITTI example 3



55

KITTI example 4



Public datasets for ADAS

Cityscapes:

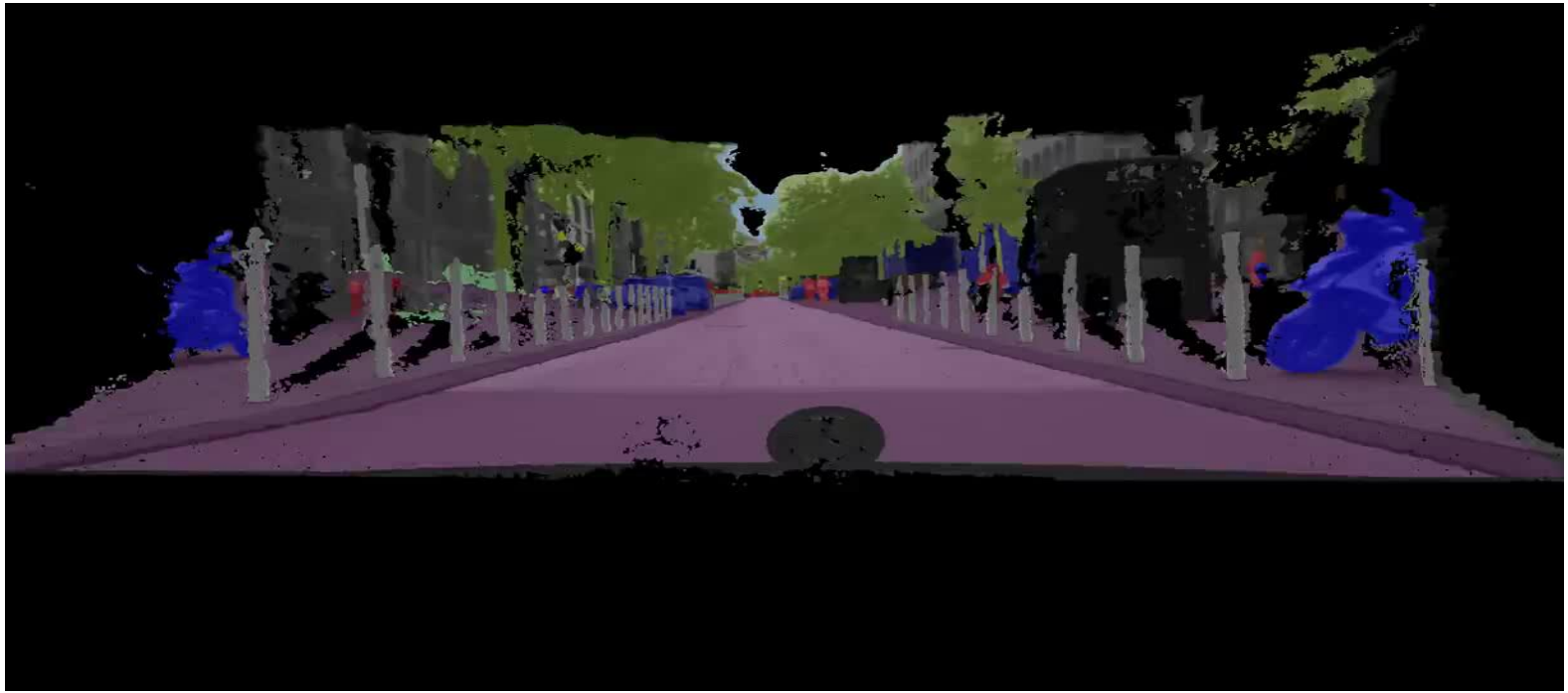
- 30 classes
- 50 cities
- 5000 annotated images with fine annotations
- 20 000 annotated images with coarse annotations

Group	Classes
flat	road · sidewalk · parking+ · rail track+
human	person* · rider*
vehicle	car* · truck* · bus* · on rails* · motorcycle* · bicycle* · caravan*+ · trailer*+
construction	building · wall · fence · guard rail+ · bridge+ · tunnel+
object	pole · pole group+ · traffic sign · traffic light
nature	vegetation · terrain
sky	sky

Coarse annotations

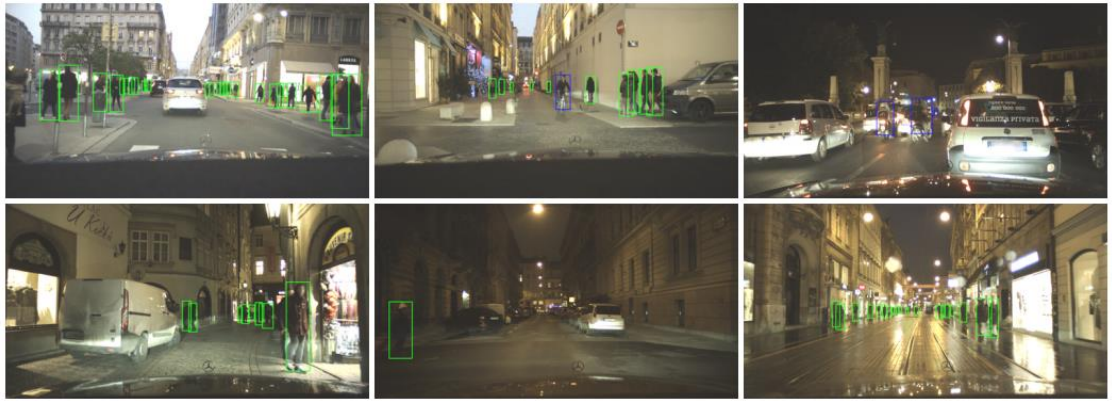
Fine annotations

Depth map with annotations



The EuroCity Persons Dataset

- 31 cities of 12 European countries.
- over 238200 person instances manually labeled in over 47300 images,
- nearly one order of magnitude larger than person datasets used previously for benchmarking.
- contains a large number of person orientation annotations (over 211200) viewed from different perspective



Braun M, Krebs S, Flohr F, Gavrila DM. The EuroCity Persons Dataset: A Novel Benchmark for Object Detection. arXiv preprint arXiv:1805.07193. 2018 May 18.

The EuroCity Persons Dataset

Insight 1: Separate Detection mode (Day/Night)

Insight 2: Appearance bias across regions

Insight 3:Transfer learning

TABLE 10
Effect of day- vs. night-time condition on detection performance ($LAMR$) for the reasonable scenario. Datasets compiled to provide otherwise similar conditions.

	Test Set	
Training Set	Night	Day
Night	**18.6**	21.3
Day	33.4	**14.5**
Day and Night	23.1	14.6

TABLE 8
Log average miss-rate ($LAMR$) on the CityPersons (CP) validation set for different training settings of Faster R-CNN. $A \rightarrow B$ denotes pre-training on A and finetuning on B.

	CityPersons Validation Set		
Training Data	reasonable	small	occluded
CP	17.2	38.9	52.0
ECP	22.5	41.2	53.7
ECP→CP	**14.9**	**30.0**	**44.8**
KITTI	57.7	81.4	87.2
KITTI→CP	17.2	37.7	49.5

TABLE 7
Average Precision on the KITTI validation set for different training settings of Faster R-CNN. $A \rightarrow B$ denotes pre-training on A and finetuning on B.

	KITTI Validation Set		
Training Data	easy	moderate	hard
KITTI	80.8	72.3	62.6
ECP	73.9	68.7	61.4
ECP→KITTI	**85.6**	**80.8**	**73.9**
CP	69.8	65.2	58.6
CP→KITTI	83.6	77.5	68.5

TABLE 9
Effect of geographical bias on detection performance ($LAMR$) for the reasonable scenario: central West Europe (WE) vs. central East Europe (EE). Datasets compiled to provide otherwise similar conditions. Results involve averages over different dataset splits.

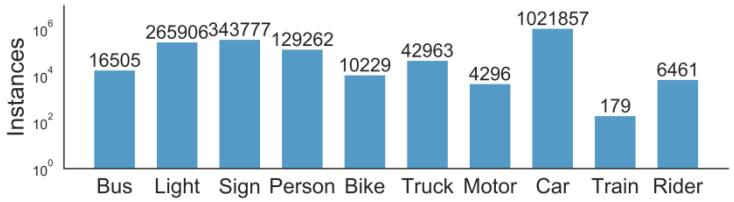
	Test Set			
Training Set	WE (mean)	WE (std)	EE (mean)	EE (std)
WE	12.8	1.3	11.2	0.7
EE	14.5	2.4	**9.2**	0.5
WE&EE	**12.4**	1.4	9.8	0.8

BDD 100k: A Large-scale Diverse Driving Video Database

- consists of 100,000 videos
- Each video is about 40 seconds long, 720p, and 30 fps
- with GPS/IMU information recorded by cell-phones to show rough driving trajectories.
- diverse locations in the United States under different weather conditions, including sunny, overcast, and rainy, at daytime or nighttime.
- labeled at several levels: image tagging, road object bounding boxes, drivable areas, lane markings, and full-frame instance segmentation

	KITTI	Cityscapes	ApolloScape	Mapillary	BDD100K
# Sequences	22	~50	4	N/A	100,000
# Images	14,999	5000 (+2000)	143,906	25,000	120,000,000
Multiple Cities	No	Yes	No	Yes	Yes
Multiple Weathers	No	No	No	Yes	Yes
Multiple Times of Day	No	No	No	Yes	Yes
Multiple Scene types	Yes	No	No	Yes	Yes

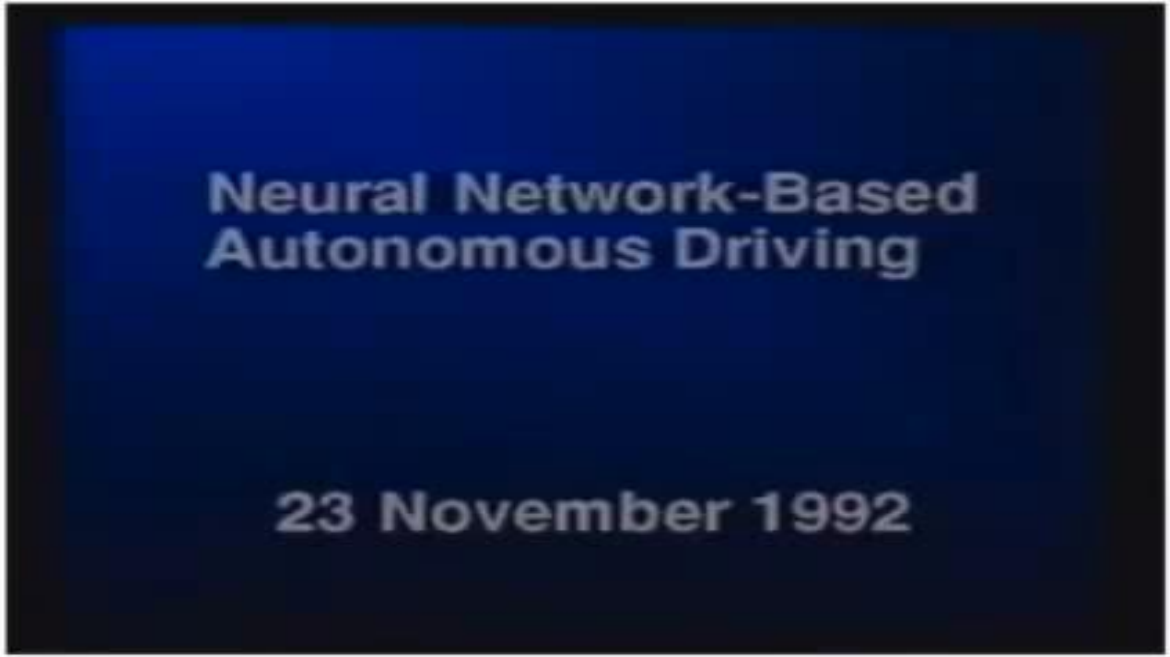
	Caltech	KITTI	CityPerson	BDD100K
# persons	1,273	6,336	19,654	86,047
# per image	1.4	0.8	7.0	1.2



Yu, Fisher, et al. "BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling." *arXiv preprint arXiv:1805.04687* (2018).

BDD100k: overview of their annotations

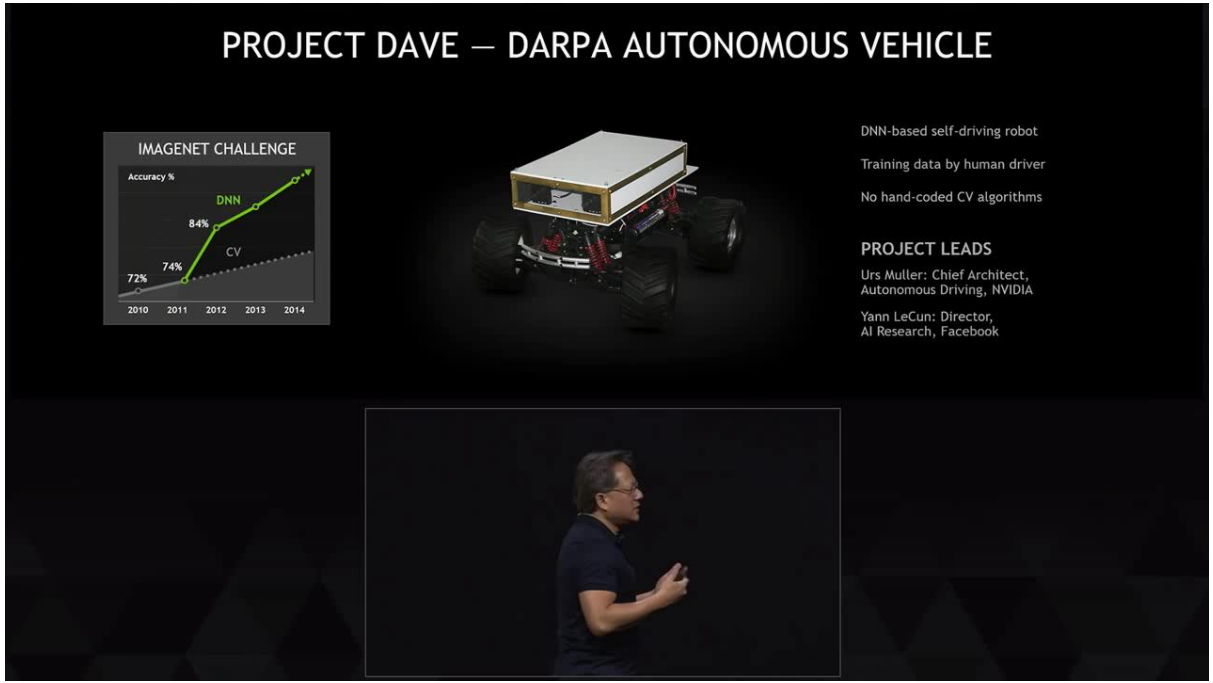
How directly can neural net achieve autonomous driving?



[Courtesy of Dean Pomerleau]

Capability of Deep Learning

DAVE Demo


PROJECT DAVE — DARPA AUTONOMOUS VEHICLE
IMAGENET CHALLENGE
Accuracy %
DNN
84%
CV
74%
72%
2010
2011
2012
2013
2014
DNN-based self-driving robot
Training data by human driver
No hand-coded CV algorithms
PROJECT LEADS
Urs Muller: Chief Architect, Autonomous Driving, NVIDIA
Yann LeCun: Director, AI Research, Facebook

DeepDrive: self-driving car AI

DeepDrive

self-driving car AI

Motivation

Self-driving cars are poised to become the most impactful technology of the 21st century. Yet, to work on them, you must be in one of the few companies who have taken on the task. Even then, you would be working in isolation from other groups doing the same thing. This does not have to be the case. Linux, PostGres, and ROS are just a few examples of a different approach -- where companies, researchers, and individuals work together on foundational technology to provide a common platform for bringing better products to market. DeepDrive aims to be such a platform for self-driving cars powered by deep learning. By combining a highly realistic driving simulation with hooks for perception and control, DeepDrive gives anyone with a computer the opportunity to build better self-driving cars.

The current generation of self-driving cars struggles in an urban setting at speeds above 25mph, requiring people to be ever-ready at the wheel. This problem is aggravated by the high risk of getting into an accident and the resultant limitation of testing new approaches on real roads. Modern video-games like GTAV, however, present a world where you can test self-driving car AI's in large complex urban areas replete with realistic roads, weather, pedestrians, cyclists, and vehicles with zero risk or cost. Another important advantage of this type of simulation is that you can quickly run the car through a barrage of safety critical situations, some of which may only occur every several million miles in reality. This dramatically decreases the time involved in properly vetting cars while giving transparency to the decisions different types of AI's will make.

The high fidelity and vast open world provided by GTAV also presents the most complex virtual RL environment to date for testing sensorimotor AI. This allows a new level of testing for safety in AI as existing environments don't offer the same opportunities for reward hacking, distributional shift, and negative side effects. Finally, developing self-driving cars out in the open provides a level of transparency not usually seen in AI applications and in an area where it is crucial to have visibility into both the safety and correctness of the system.

Baseline model

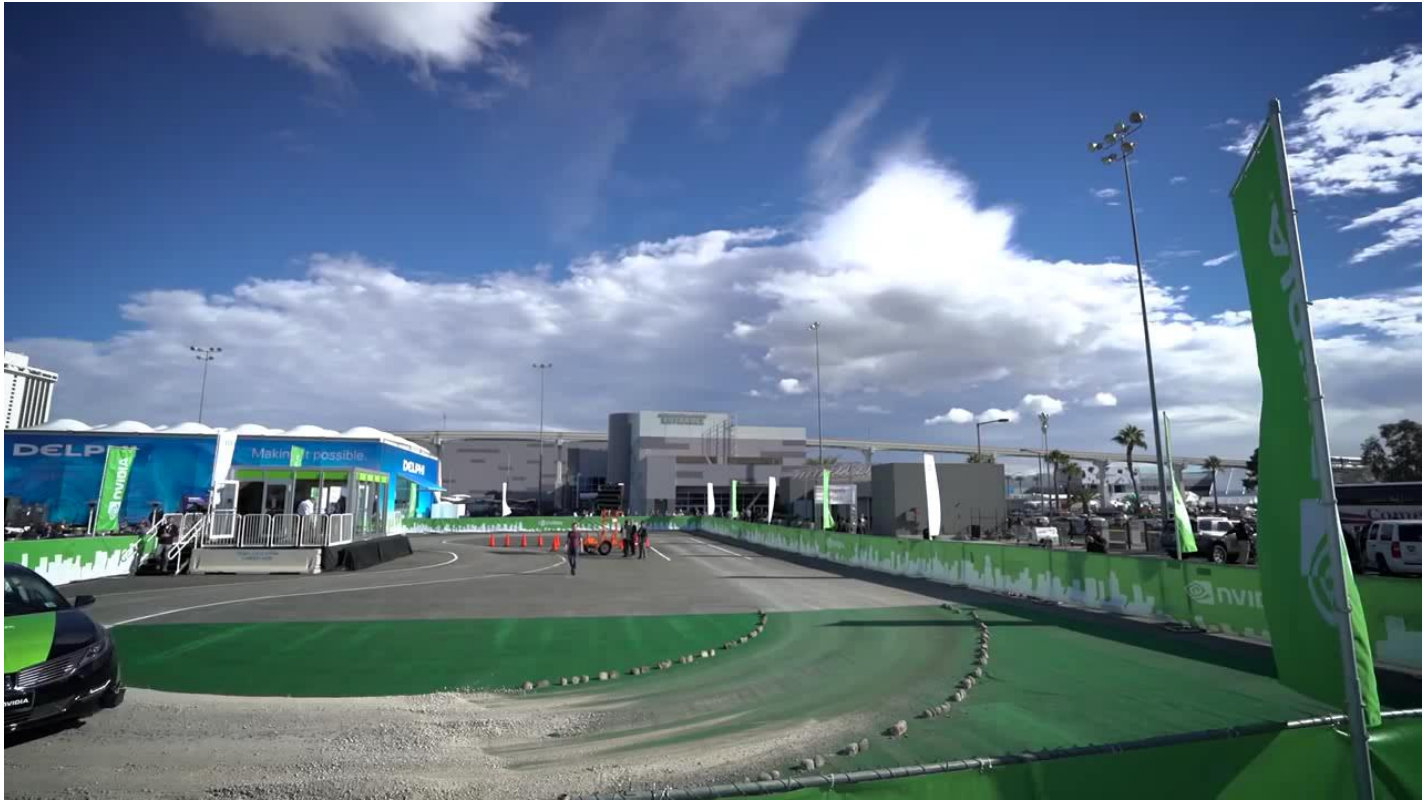
An initial 8-layer neural net with the AlexNet architecture is being made available as well as the dataset it was trained on. Training was done on raw image input from a forward mounted camera regressed against steering, throttle, yaw, and forward-speed control values produced by an in-game AI. This model is able to steer the car to stay in the lane, stop for other cars, and works well in a variety of weather and lighting conditions as shown below:

DeepDrive Demo1

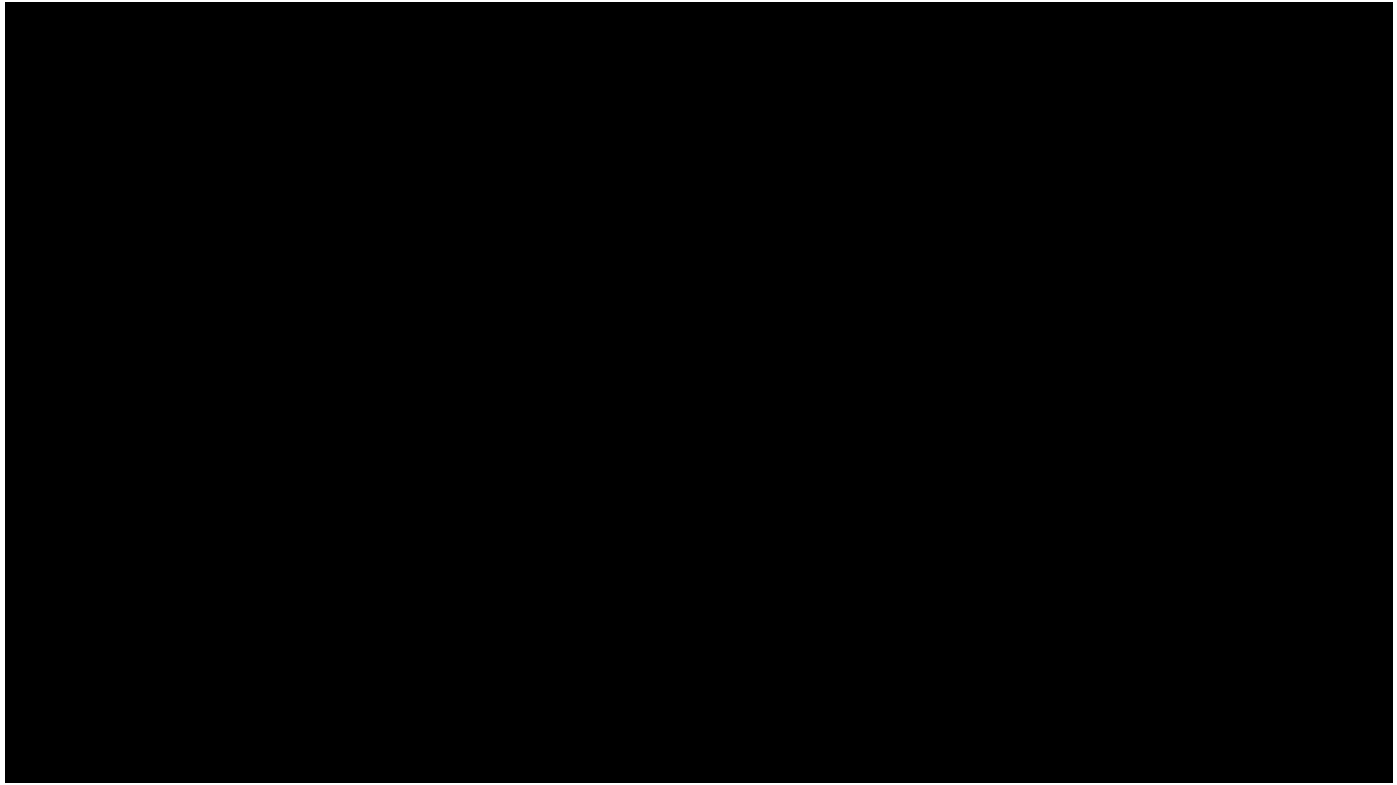
DeepDrive Demo2

DeepDrive Demo3

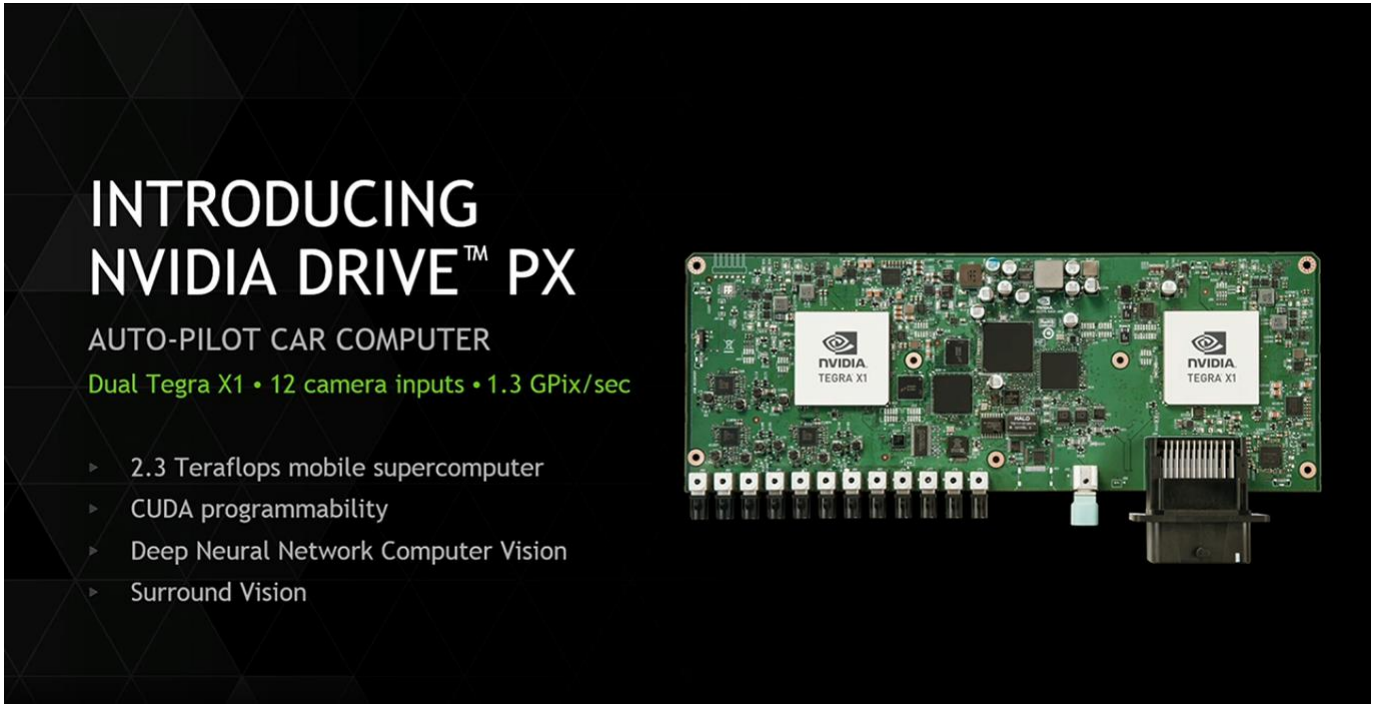
NVIDIA’s BB8



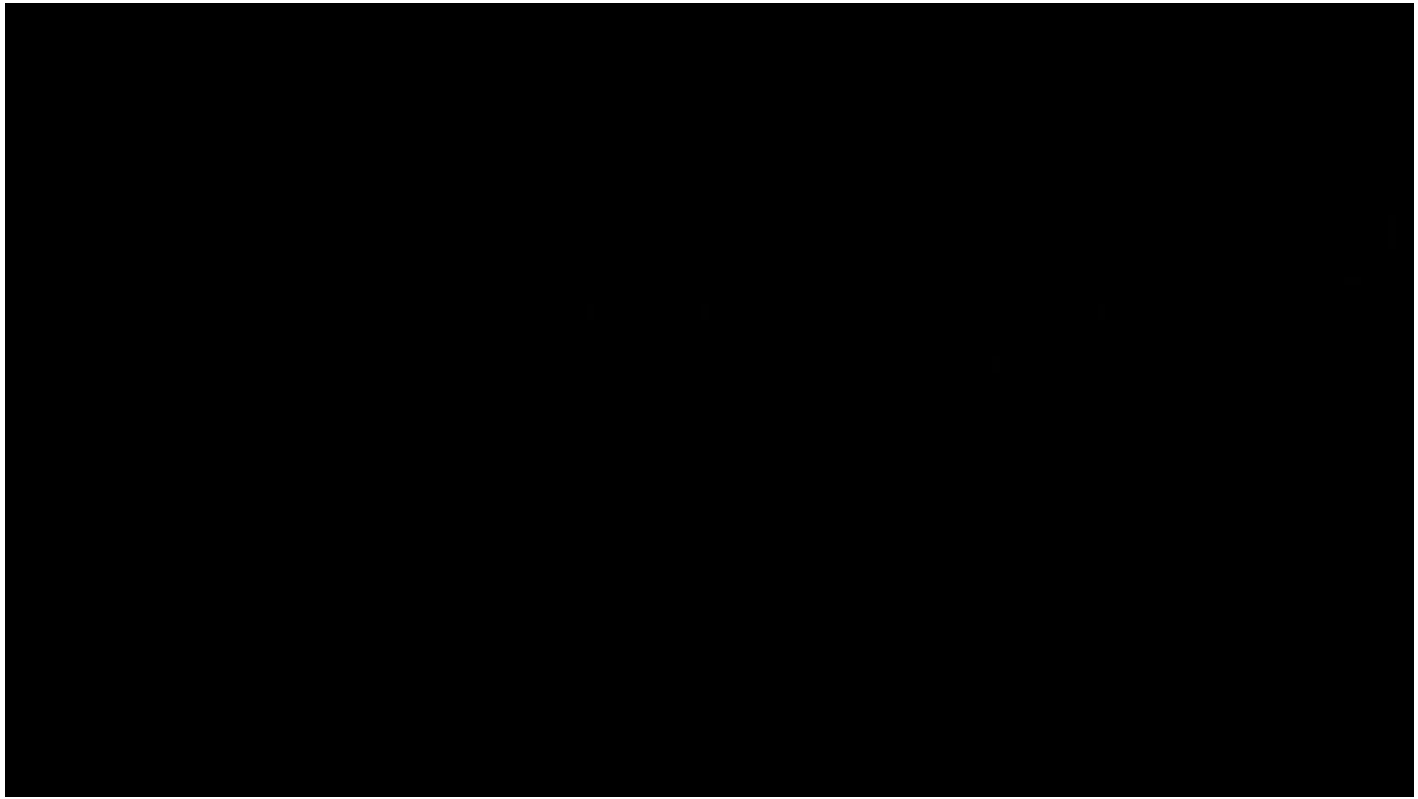
NVIDIA’s BB8 on Lombard Street



NVIDIA DRIVE™ PX



2016 CES NVIDIA Demo



Introduction: Semantic Segmentation

Label every pixel!

Don’t differentiate instances (cows)

Classic computer vision problem

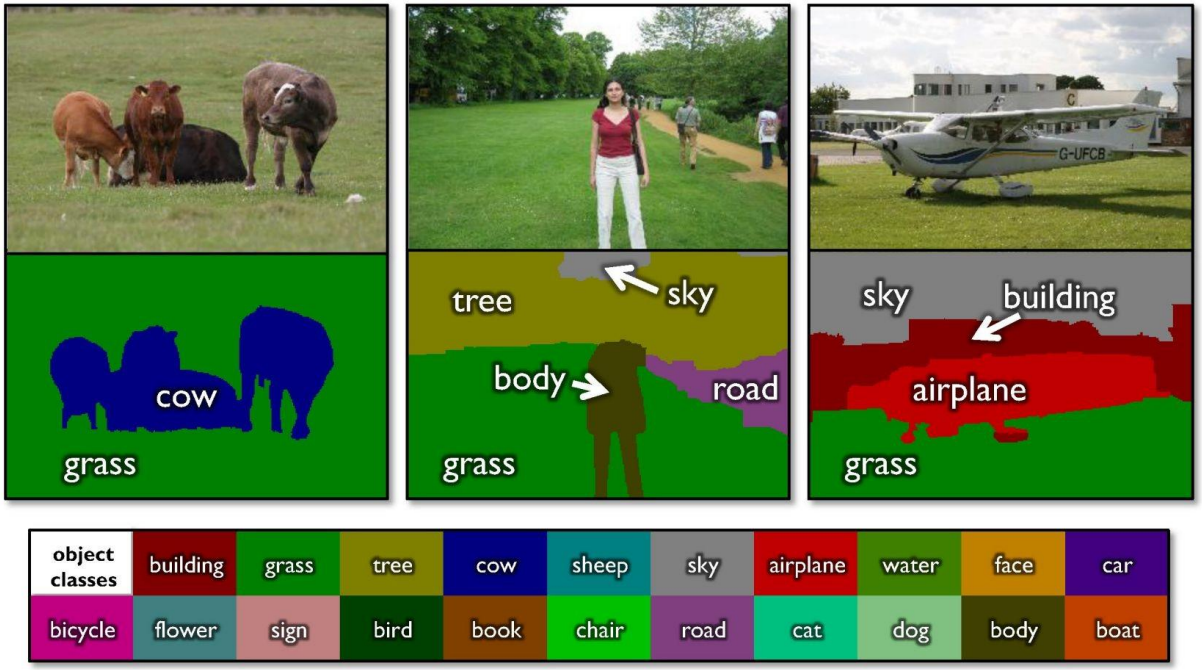


Figure credit: Shotton et al, “TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context”, IJCV 2007

Introduction: Instance Segmentation

Detect instances, give category, label pixels

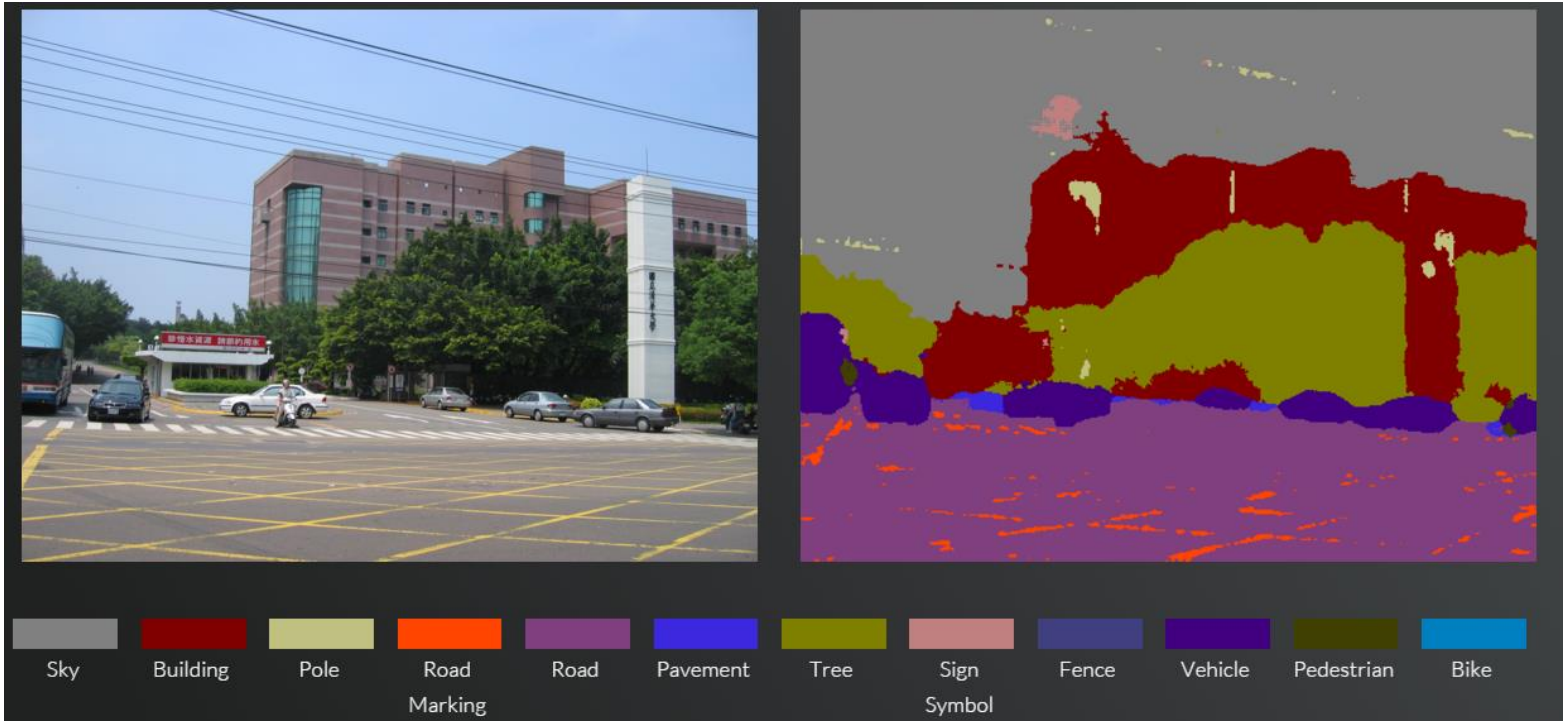
“simultaneous detection and segmentation” (SDS)

Lots of recent work (MS-COCO)

Predictions **Ground truth**

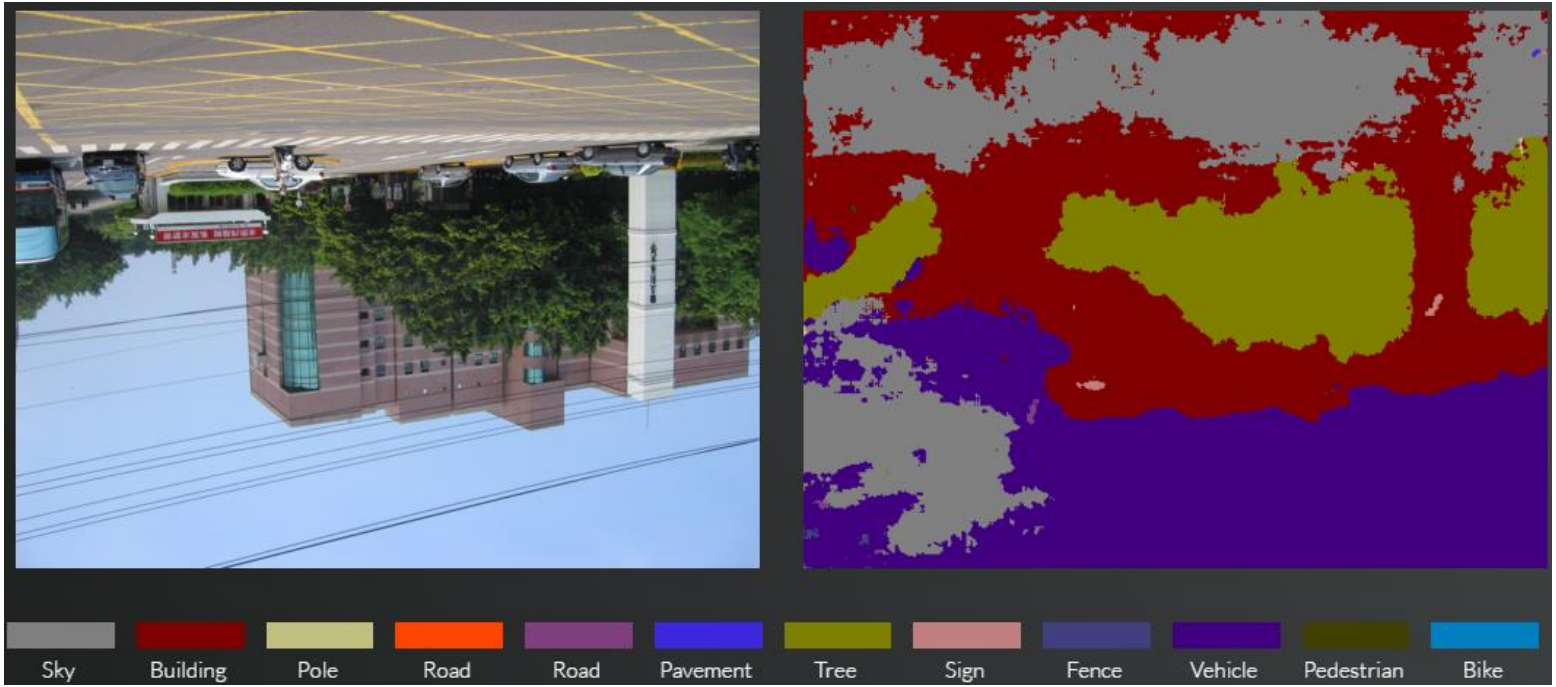
Figure credit: Dai et al, “Instance-aware Semantic Segmentation via Multi-task Network Cascades”, arXiv 2015

Segnet: semantic segmentation



78

Segnet: extreme case



Segnet video demo

80

Semantic & instance segmentation

PoseNet

PoseNet was trained with the Cambridge Landmarks Dataset.

This is a large urban relocalization dataset with 6 scenes from around Cambridge University containing over 12,000 images labelled with their full 6-DOF camera pose.

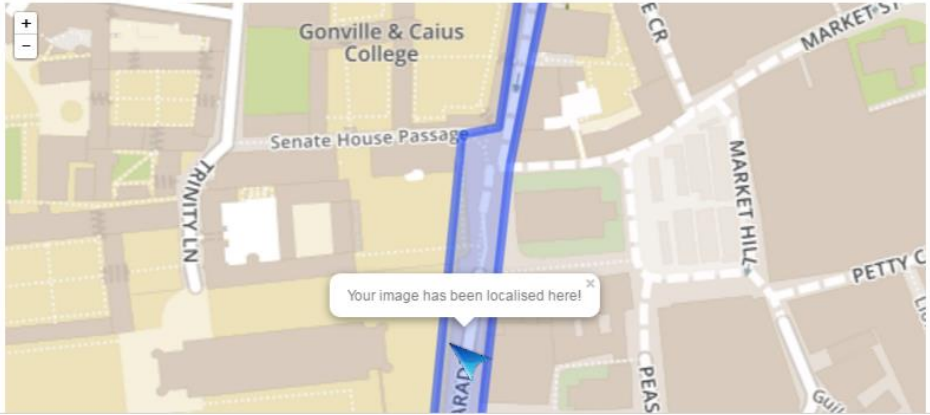
Six scenes include:

- King's College:5.63Gb
- Street:10.6Gb
- Old Hospital:4.7Gb
- Shop Façade:1.23Gb
- St Mary's Church:6.87Gb
- Trinity Great Court:6.22Gb

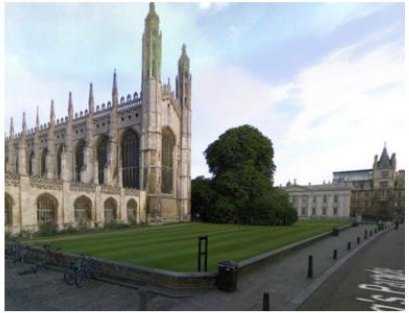
PoseNet: example 1

We have localised your input image on the map below! The **blue arrow** shows where we think you are. The image must have been taken within the blue highlighted region on the map.

This is the closest view in Google Maps Street View to the blue arrow.

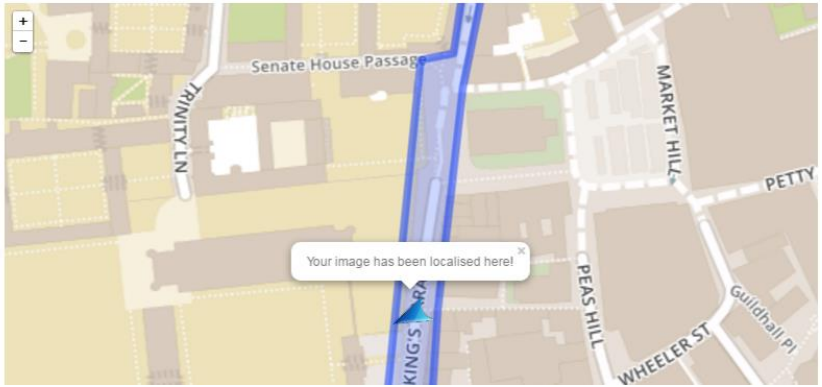


PoseNet: example 2



This is the closest view in Google Maps Street View to the blue arrow.

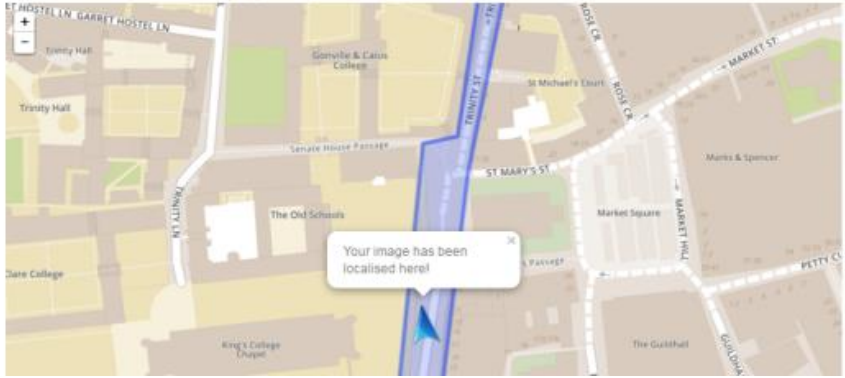
We have localised your input image on the map below! The **blue arrow** shows where we think you are. The image must have been taken within the blue highlighted region on the map.



PoseNet: example 3

This is the closest view in Google Maps Street View to the blue arrow.

We have localised your input image on the map below! The **blue arrow** shows where we think you are. The image must have been taken within the blue highlighted region on the map



Strong AI vs Weak AI

- Strong AI:

Strong is about building a machine that is capable of emulating the human mind to such an extent that it matches or exceeds it.

Strong AI is machine intelligence with full range of human intelligence.

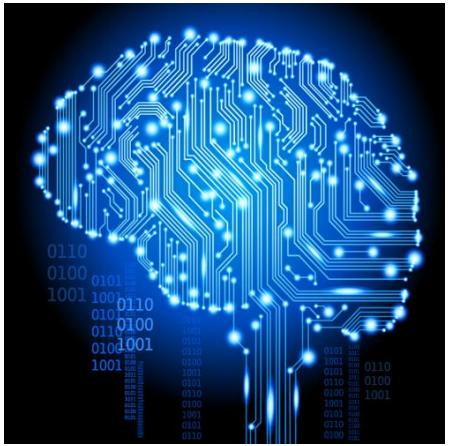
- Weak AI:

Weak AI is a system that uses a set of pre-programmed rules to apply them to any tasks, to reach a successful fulfillment.

Example:

iPhone 3g Commercial in 2009:

"There's an app for that"



In 2019

"There's a DeepNet for that"

Tools

Caffe (C++,matlab,Python)

Caffe2 (C++, Python)

~~cuda-convnet2 (C++,Python)~~

~~Theano (Python)~~

~~Pylearn2 (Python)~~

~~Lasagne (Python)~~

Torch (lua)

PyTorch (Python)

MatConvNet (Matlab)

TensorFlow (Python, C++)

Keras(Python)

Mxnet(R, Python, C++, Julia, Android deployable)

Ref: https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

Resources

- http://deeplearning.net/
  - Hub to many other deep learning resources
- https://github.com/ChristosChristofidis/awesome-deep-learning
  - A resource collection deep learning
- https://github.com/kjw0612/awesome-deep-vision
  - A resource collection deep learning for computer vision
- http://cs231n.stanford.edu/syllabus.html
  - Nice course on CNN for visual recognition
- http://gitxiv.com/category/deeplearning
  - Everything: paper, code, data, community
- http://cs231n.github.io/python-numpy-tutorial/
  - Python tutorial

Talks about Deep Learning

- https://www.youtube.com/watch?v=40riCqvRoMs
  - TED talk: How We Teach Computers to Understand Pictures by Fei-Fei Li
- https://www.youtube.com/watch?v=n1ViNeWhC24
  - Deep Learning, Self-Taught Learning and Unsupervised Feature Learning by Andrew Ng
- https://www.youtube.com/watch?v=_Zfm0Rz6svY
  - Deep Learning by Geoffrey Hinton
- https://www.youtube.com/watch?v=zY0qX4lbjYY
  - Deep Learning by Yann LeCun
- https://www.youtube.com/watch?v=zY0qX4lbjYY
  - A Tutorial on Deep Learning by KaiYu

The End