# Lecture 1 History

| | |
|---|---|
| Created | @May 20, 2021 3:57 PM |
| Tags | |

Week 1 - Lecture: History, motivation, and evolution of Deep Learning

Course website: http://bit.ly/pDL-homePlaylist: http://bit.ly/pDL-YouTubeSpeaker: Yann LeCunWeek 1: http://bit.ly/pDL-en-010:00:00 - Week 1 - LectureLECTURE ...

https://www.youtube.com/watch?v=0bMe_vCZo30

NEW YORK UNIVERSITY

Deep Learning

Week 1 – Lecture

Motivation of Deep Learning, and Its History and Inspiration

🎙 Yann LeCun Basics of Supervised Learning, Neural Nets, Deep Learning Backpropagation and architectural components Convolutional neural network and its applications More Deep Learning Architectures Regularization Tricks / Optimization Tricks / Understanding how Deep Learning works Energy-based models Self-supervised

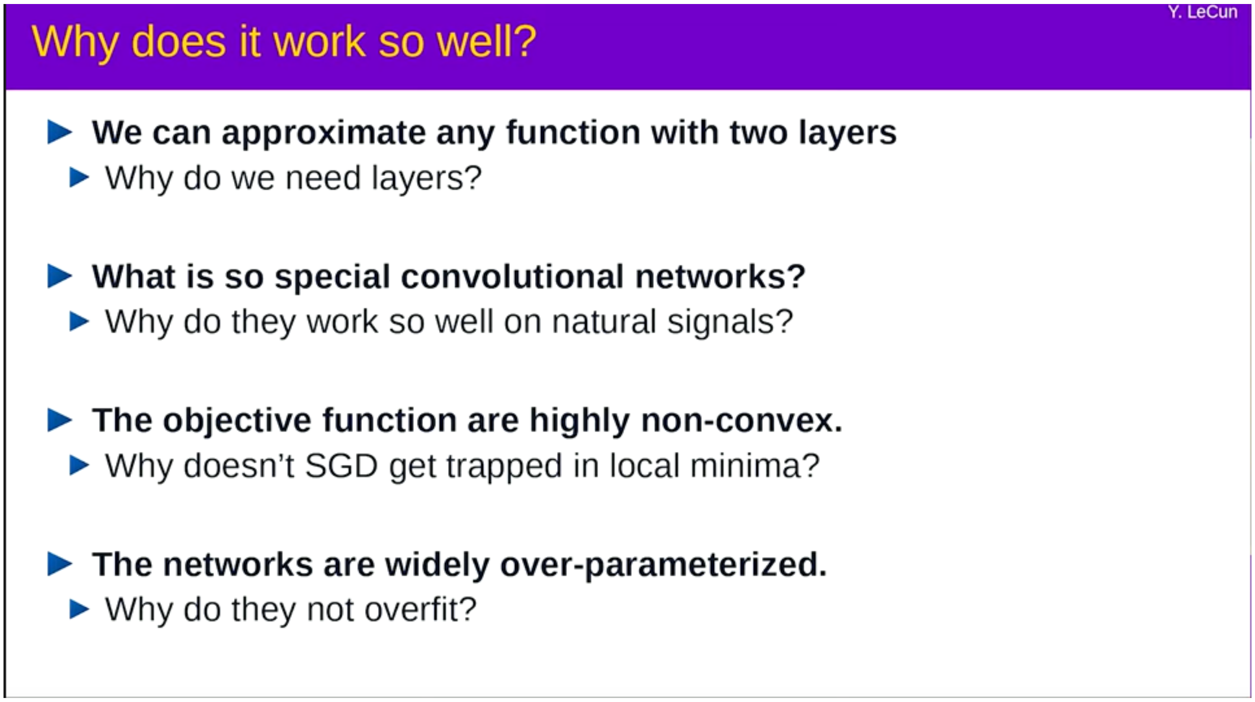
https://atcold.github.io/pytorch-Deep-Learning/en/week01/01-1/

Evolution and Uses of CNNs and Why Deep Learning?

🎙 Yann LeCun In animal brains, neurons react to edges that are at particular orientations. Groups of neurons that react to the same orientations are replicated over all of the visual field. Fukushima (1982) built a neural net (NN) that worked the same way as the brain, based on two concepts.

https://atcold.github.io/pytorch-Deep-Learning/en/week01/01-2/

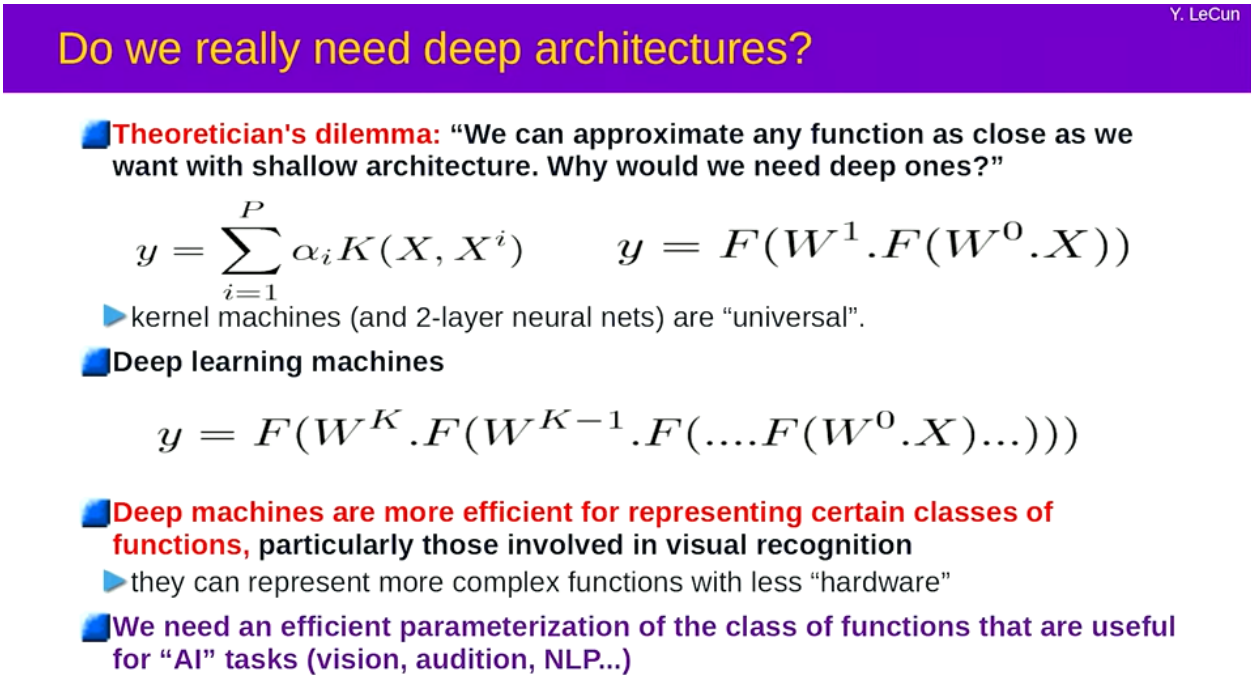
CV: semantic segmentation: mark the image's contents with different colors meaning different labels.



Y. LeCun

## Why does it work so well?

- **We can approximate any function with two layers**
  - Why do we need layers?
- **What is so special convolutional networks?**
  - Why do they work so well on natural signals?
- **The objective function are highly non-convex.**
  - Why doesn't SGD get trapped in local minima?
- **The networks are widely over-parameterized.**
  - Why do they not overfit?



Y. LeCun

## Do we really need deep architectures?

- **Theoretician's dilemma: "We can approximate any function as close as we want with shallow architecture. Why would we need deep ones?"**

$$y = \sum_{i=1}^{P} \alpha_i K(X, X^i) \qquad y = F(W^1.F(W^0.X))$$

  - kernel machines (and 2-layer neural nets) are "universal".
- **Deep learning machines**

$$y = F(W^K.F(W^{K-1}.F(....F(W^0.X)...)))$$

- **Deep machines are more efficient for representing certain classes of functions, particularly those involved in visual recognition**
  - they can represent more complex functions with less "hardware"
- **We need an efficient parameterization of the class of functions that are useful for "AI" tasks (vision, audition, NLP...)**

## Why does it  work so well

- We can approximate any function with two layers? Why do we need layers?

Deep machines are more efficient for representing certain classes of functions, particular those involved in visual recognition. They can **representation more complex functions**

**with less hardware(fewer parameters).(less data?)**

We need an efficient parameterization of the class of functions that are useful for AI task.

▼ What is so special convolution networks? Why do they work so well on natural signals

MLP → CNN → Transformer → MLP-mixer (Lecun argued on twitter that 1∗1 is also convolution.)

▼ The objective function are highly non-convex. Why doesn't SGD get trapped in local minima?

▼ The networks are widely over-parameterized? Why do they not overfit?

## Learning representation

Basic principle: expand the dimension of the representation so that things are more likely to become linearly separable

space tiling(空间平铺)

random projections

polynomial classifier (feature cross-products)

radial basis functions(rbf)

kernel machines

## What are good features?

Discover the hidden structure in high-dimensional

discover & disentangle the independent explanatory factors.

The Manifold hypothesis:

Natural data lives in a low-dimensional (non-linear) manifold.

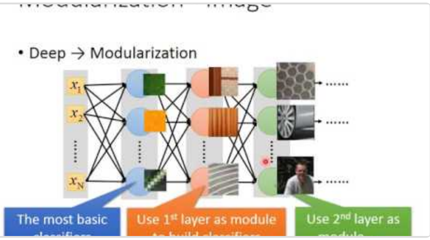
Variables in natural data are mutually dependent.

The number of independently movable muscle, about 50.

## Why Deep?

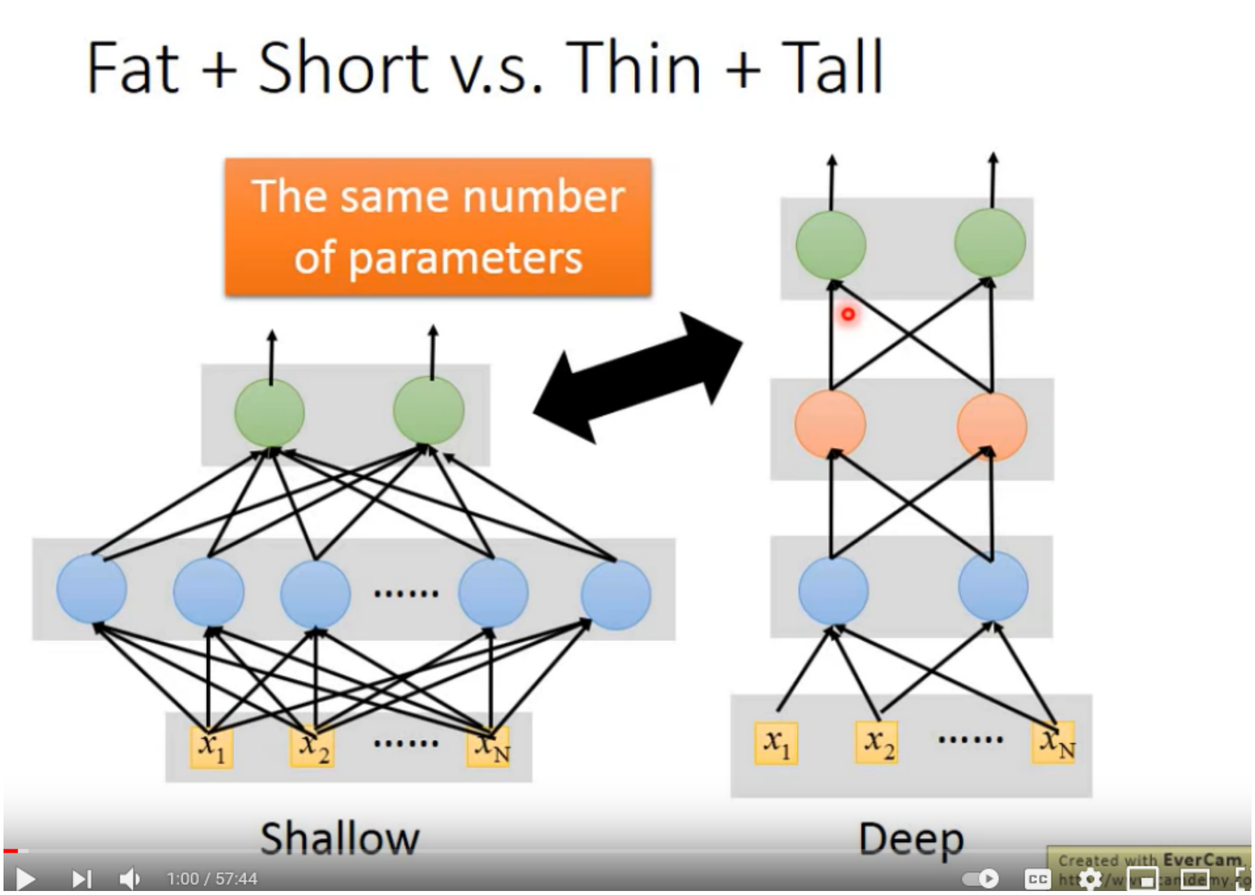
ML Lecture 11: Why Deep?

https://www.youtube.com/watch?v=XsC9byQkUH8



Modularization

Deep → Modularization → Less training data?

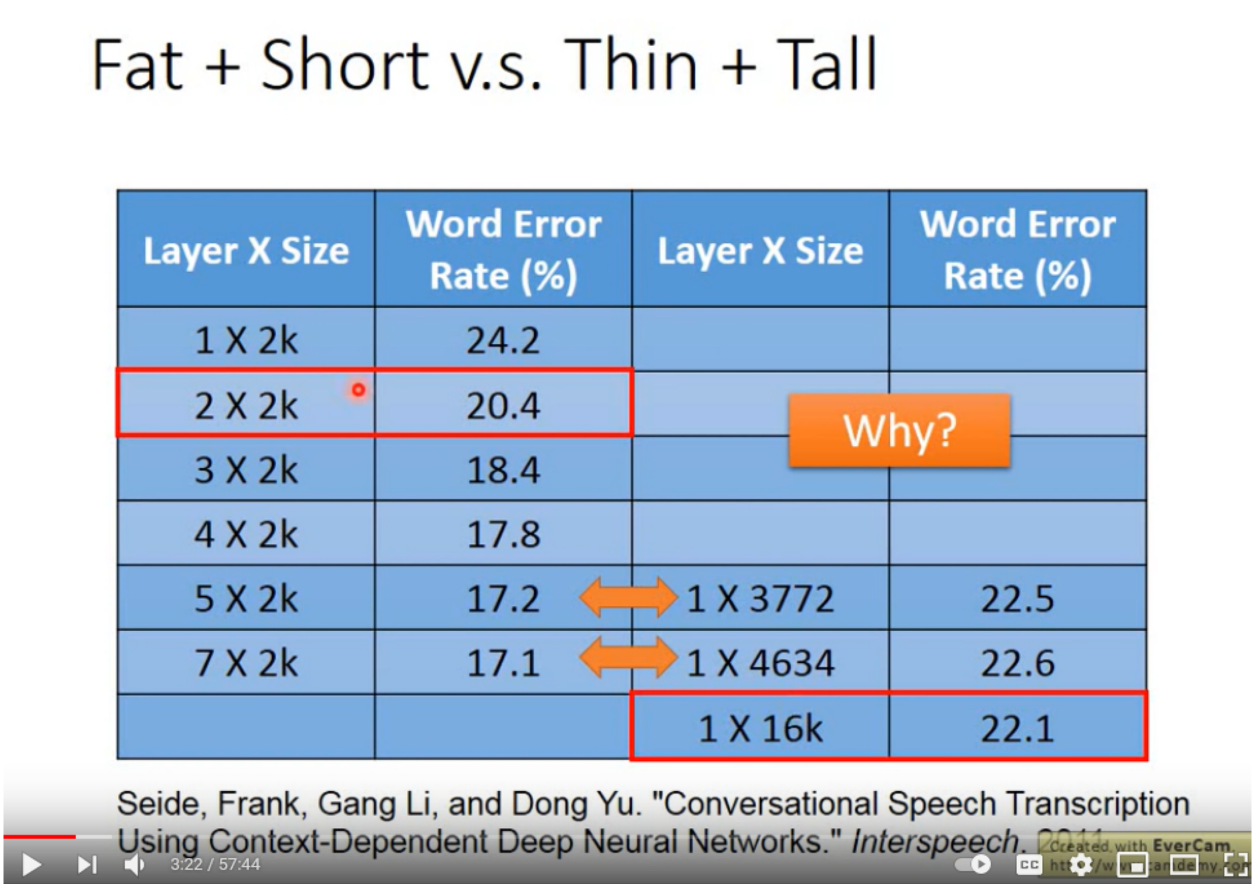


Fat + Short v.s. Thin + Tall

The same number of parameters

Shallow

Deep



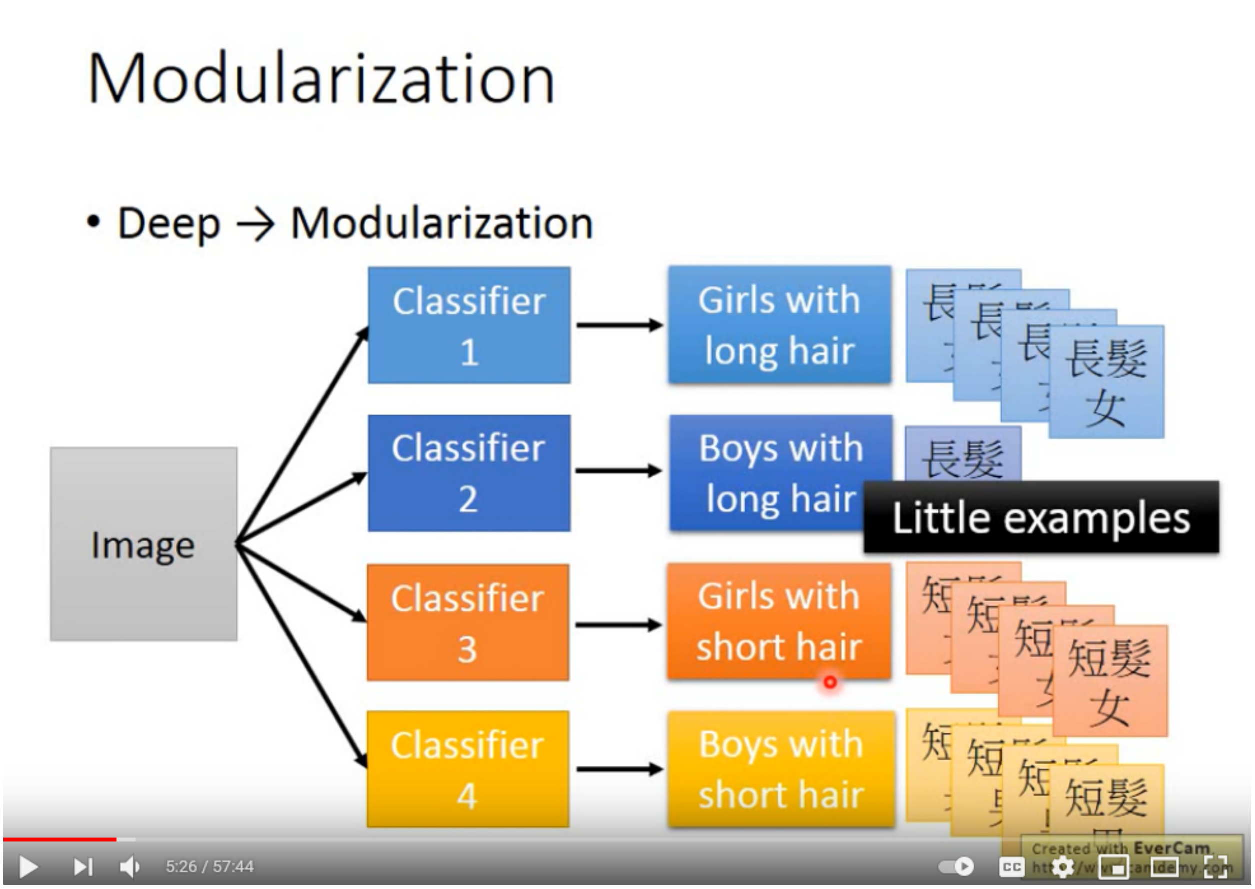
# Fat + Short v.s. Thin + Tall

| Layer X Size | Word Error Rate (%) | Layer X Size | Word Error Rate (%) |
|---|---|---|---|
| 1 X 2k | 24.2 | | |
| 2 X 2k | 20.4 | | |
| 3 X 2k | 18.4 | | |
| 4 X 2k | 17.8 | | |
| 5 X 2k | 17.2 | 1 X 3772 | 22.5 |
| 7 X 2k | 17.1 | 1 X 4634 | 22.6 |
| | | 1 X 16k | 22.1 |

Why?

Seide, Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011

Why Fat short neural network works worse than thin tall? even with more parameters?



# Modularization

- Deep → Modularization

Image

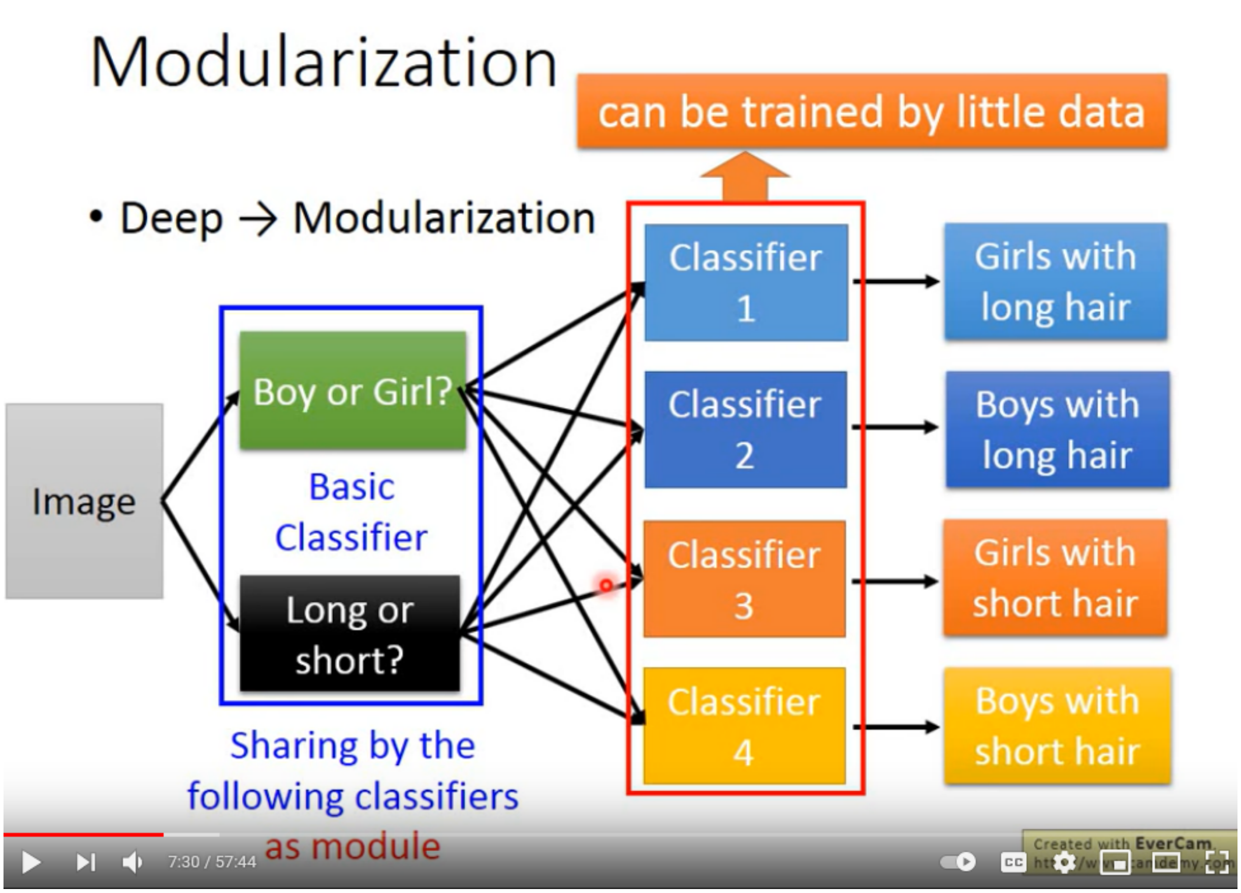
Classifier 1 → Girls with long hair

Classifier 2 → Boys with long hair

Classifier 3 → Girls with short hair

Classifier 4 → Boys with short hair

Little examples



# Modularization

can be trained by little data

- Deep → Modularization

Image

Boy or Girl?

Basic Classifier

Long or short?

Sharing by the following classifiers as module

Classifier 1

Classifier 2

Classifier 3

Classifier 4

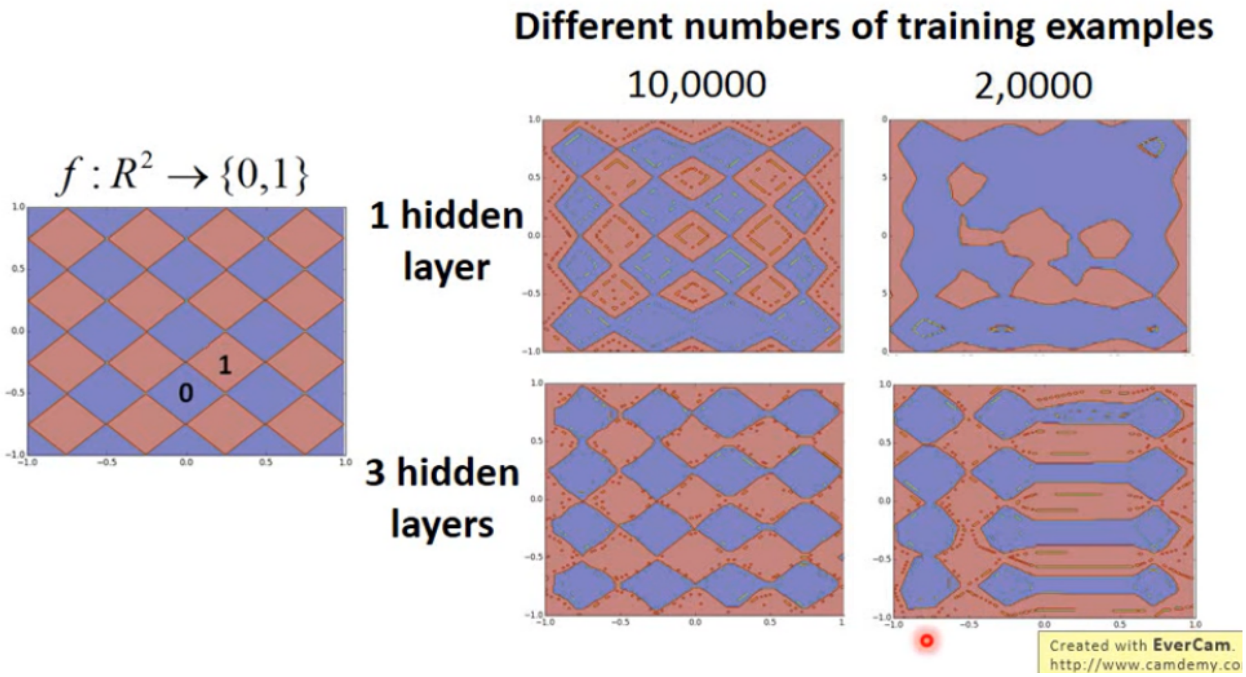
Girls with long hair

Boys with long hair

Girls with short hair

Boys with short hair

# More Analogy - Experiment



Different numbers of training examples

10,0000

2,0000

$f: R^2 \rightarrow \{0,1\}$

1 hidden layer

3 hidden layers