भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

EE63050: DEEP LEARNING

Motivation and Introduction

OUTLINE

- Motivation
- Building blocks
- Training
- Generalisation

MOTIVATION

Introducing ChatGPT

We’ve trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

Try ChatGPT ↗   Read about ChatGPT Plus

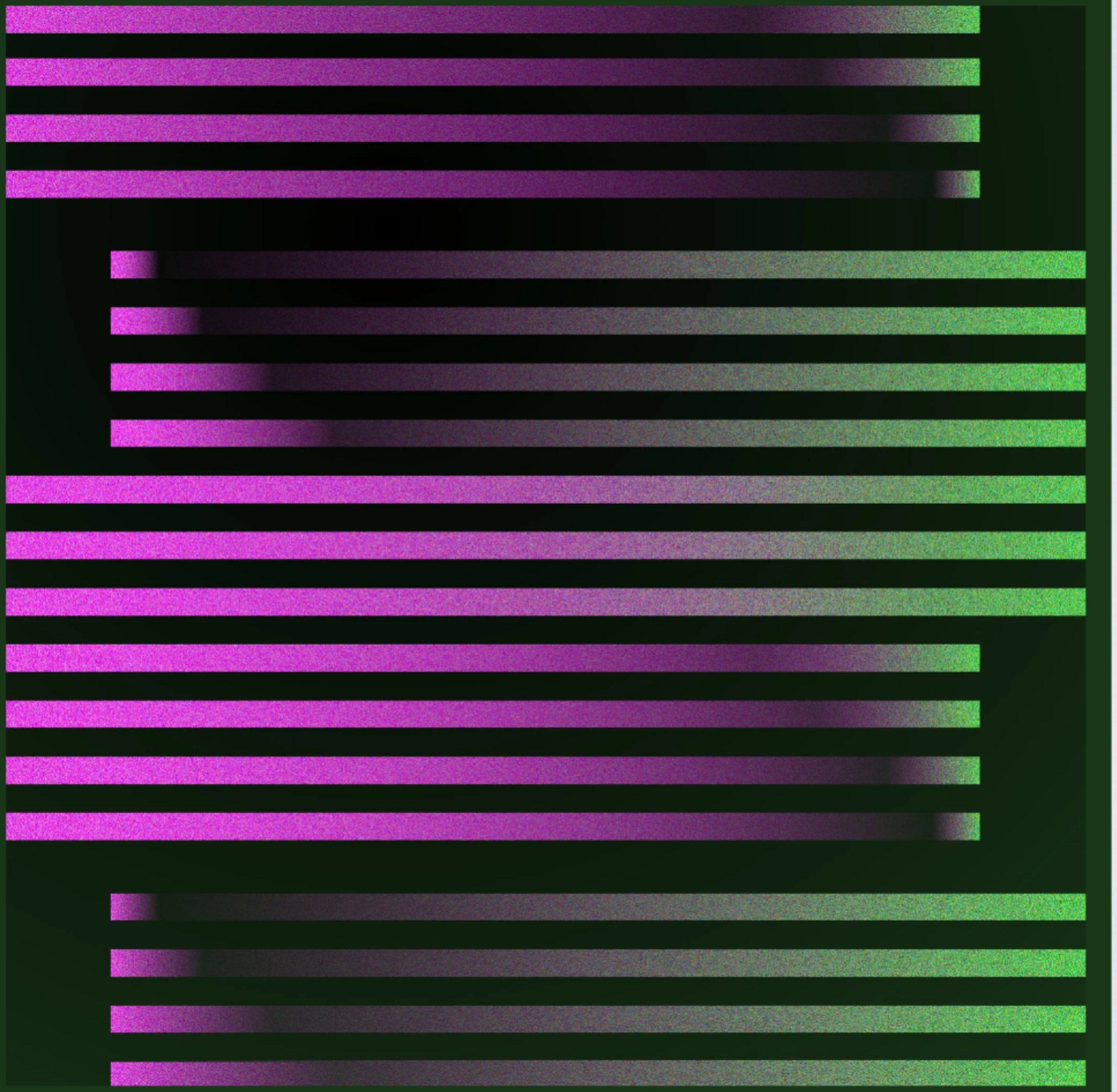


Illustration: Ruby Chen

MOTIVATION


DALL·E 2
DALL·E 2 is an AI system that can create realistic images and art from a description in natural language.
Try DALL·E ↗
Follow on Instagram ↗
car
car
car

MOTIVATION

RSNA
Radiological Society of North America

Membership | Annual Meeting | Journals | Education | Research | Practice Tools

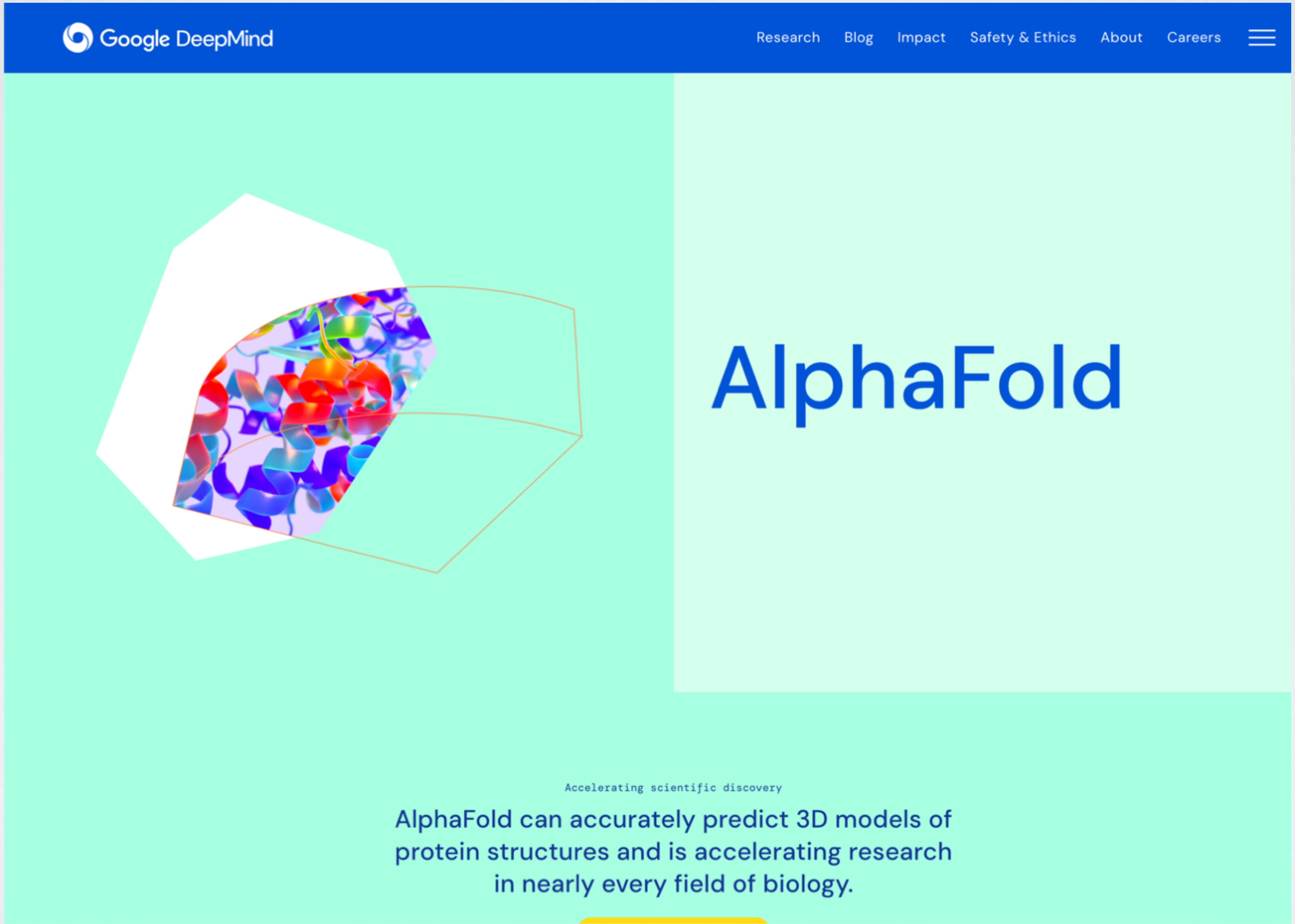
RSNA News

AI Outperformed Standard Risk Model for Predicting Breast Cancer

Algorithms identify both missed cancers and breast tissue features that help predict future cancers

June 06, 2023

MOTIVATION


Google DeepMind
Research
Blog
Impact
Safety & Ethics
About
Careers
AlphaFold
Accelerating scientific discovery
AlphaFold can accurately predict 3D models of protein structures and is accelerating research in nearly every field of biology.

MOTIVATION

REVIEW

doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

MOTIVATION

[Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury]

Deep Neural Networks for Acoustic Modeling in Speech Recognition

[The shared views of four research groups]

MOTIVATION

nature COMMUNICATIONS

ARTICLE

Received 19 Feb 2014 | Accepted 4 Jun 2014 | Published 2 Jul 2014

DOI: 10.1038/ncomms5308

Searching for exotic particles in high-energy physics with deep learning

P. Baldi[1], P. Sadowski[1] & D. Whiteson[2]

MOTIVATION

Journals & Magazines > IEEE Wireless Communications > Volume: 27 Issue: 1

Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions

Publisher: IEEE Cite This PDF

Hongji Huang ; Song Guo ; Guan Gui ; Zhen Yang ; Jianhua Zhang ; Hikmet Sari ; Fumiyuki Adachi **All Authors**

Abstract

Document Sections

- Introduction
- Overview of Deep Learning for Wireless Communication
- Deep Learning for 5G: An Alternative Approach
- Future Challenges and Opportunities
- Deep Reinforcement Learning for the Wireless Physical Layer

Show Full Outline

Authors

Abstract:
The new demands for high-reliability and ultra-high capacity wireless communication have led to extensive research into 5G communications. However, current communication systems, which were designed on the basis of conventional communication theories, significantly restrict further performance improvements and lead to severe limitations. Recently, the emerging deep learning techniques have been recognized as a promising tool for handling the complicated communication systems, and their potential for optimizing wireless communications has been demonstrated. In this article, we first review the development of deep learning solutions for 5G communication, and then propose efficient schemes for deep learning-based 5G scenarios. Specifically, the key ideas for several important deep learning-based communication methods are presented along with the research opportunities and challenges. In particular, novel communication frameworks of NOMA, massive multiple-input multiple-output (MIMO), and millimeter wave (mmWave) are investigated, and their superior performances are demonstrated. We envision that the appealing deep learning- based wireless physical layer frameworks will bring a new direction in communication theories and that this work will move us forward along this road.

Published in: IEEE Wireless Communications (Volume: 27 , Issue: 1, February 2020)

Page(s): 214 - 222

Date of Publication: 05 August 2019

INSPEC Accession Number: 19429398

DOI: 10.1109/MWC.2019.1900027

More Like This

A Comprehensive Survey on Millimeter Wave Communications for Fifth-Generation Wireless Networks: Feasibility and Challenges
IEEE Access
Published: 2020

Coordinated Beamforming for UAV-Aided Millimeter-Wave Communications Using GPML-Based Channel Estimation
IEEE Transactions on Cognitive Communications and Networking
Published: 2021

Show More

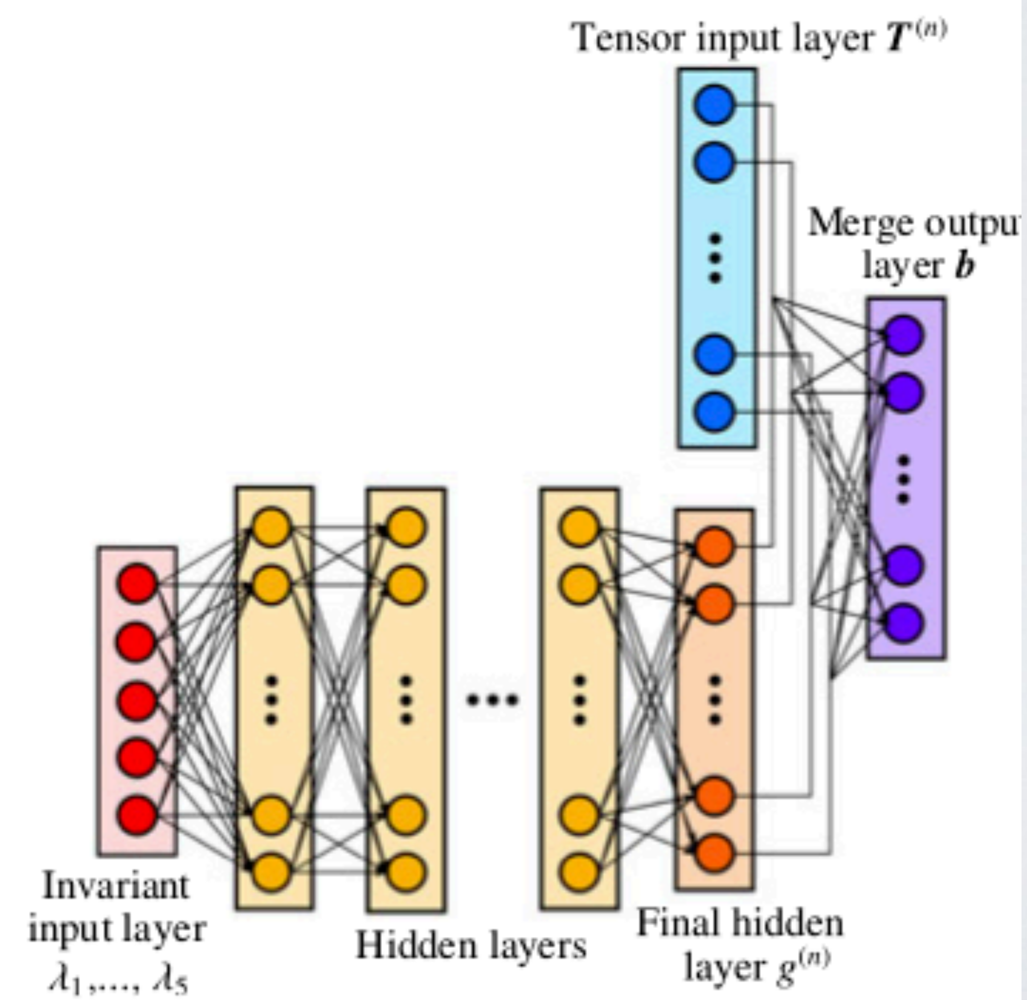
MOTIVATION

Focus on Fluids journals.cambridge.org/focus

Deep learning in fluid dynamics

J. Nathan Kutz†

Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA



MOTIVATION

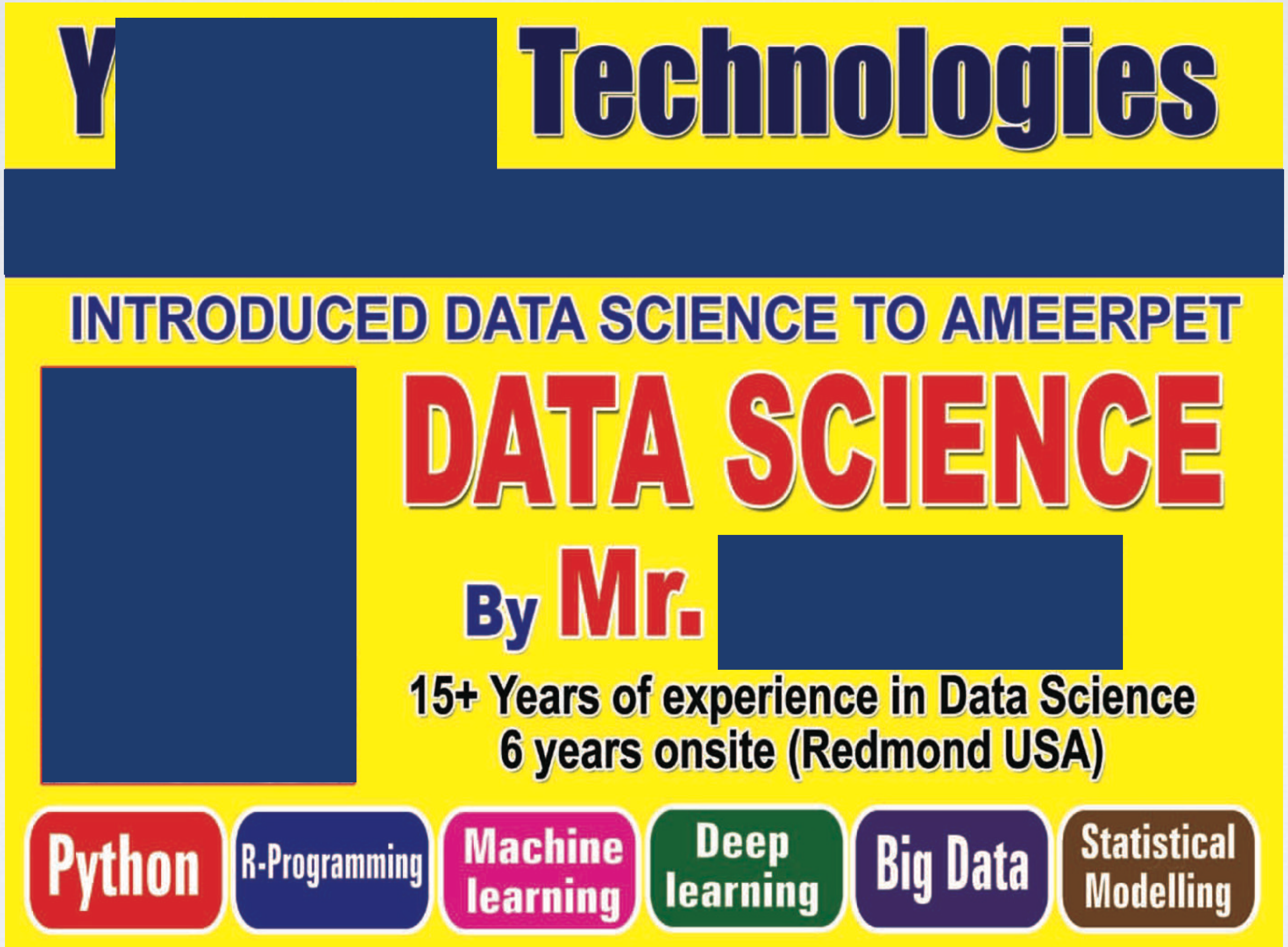
Collaborative Deep Learning for Recommender Systems

Hao Wang
Hong Kong University of Science and Technology
hwangaz@cse.ust.hk

Naiyan Wang
Hong Kong University of Science and Technology
winsty@gmail.com

Dit-Yan Yeung
Hong Kong University of Science and Technology
dyyeung@cse.ust.hk

MOTIVATION


Y[redacted] Technologies
[redacted]
INTRODUCED DATA SCIENCE TO AMEERPET
[redacted]
DATA SCIENCE
By Mr. [redacted]
15+ Years of experience in Data Science
6 years onsite (Redmond USA)
Python
R-Programming
Machine learning
Deep learning
Big Data
Statistical Modelling

MOTIVATION


P.G GENTS
EMINENT IT INFO
Weekday / Weekend Batch
DATA SCIENCE
R, STASTICS, PHYTHON, MACHINE LEARNING
DEEP LEARNING, NLP, AI
Python
R [redacted]
934 934 6666

BASIC MATH REVIEW

- A **vector** is denoted by: $\vec{x} = [x_1, x_2, ..., x_n]^T$
- Equation of a **straight line**: $\hat{y} = mx + c$
- Example of a **non-linear function**: $\sigma(x) = \frac{1}{1+e^{-x}}$
- A **parameterised function** is denoted by:

$$\hat{y} = f(x;\theta)$$

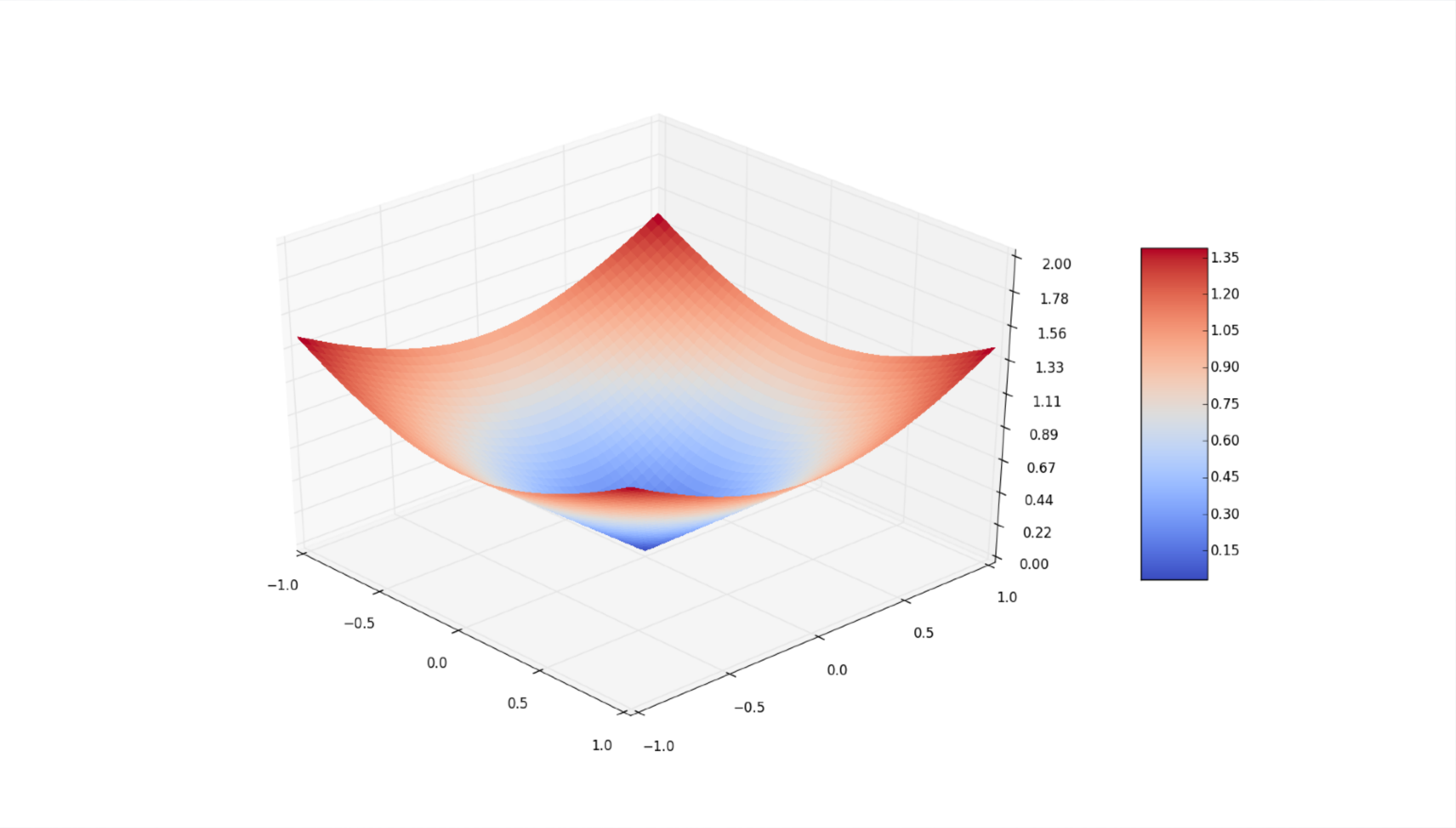
BASIC MATH REVIEW

- The **derivative/slope** of a function is zero at its maximum/minimum (if it exists): $\frac{d}{d\theta}[R(\theta)]=0$

- The **gradient** of a multivariable function is like the **slope** of a single variable function

BASIC MATH REVIEW

- Descending against the gradient takes us to the local-minimum



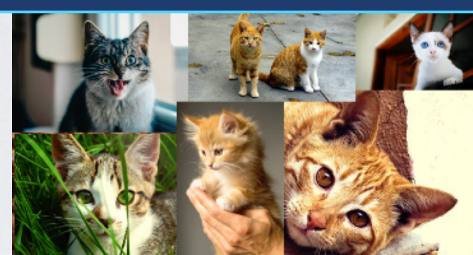
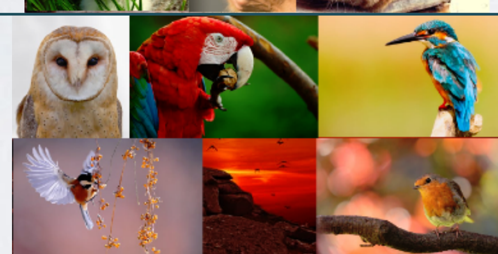
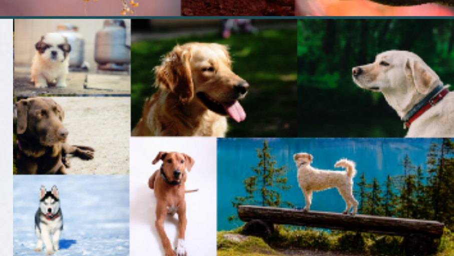
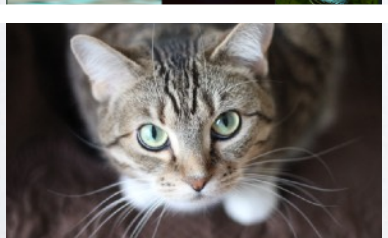
WHAT IS MACHINE LEARNING?

- **Learning from data:** a method of teaching machines to improve predictions from data
- A method for computers to **recognise patterns**
- "A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**." - *T. M. Mitchell, CMU*

MACHINE LEARNING FLAVOURS

- **Supervised Learning**
- Unsupervised Learning
- Reinforcement Learning

SUPERVISED LEARNING

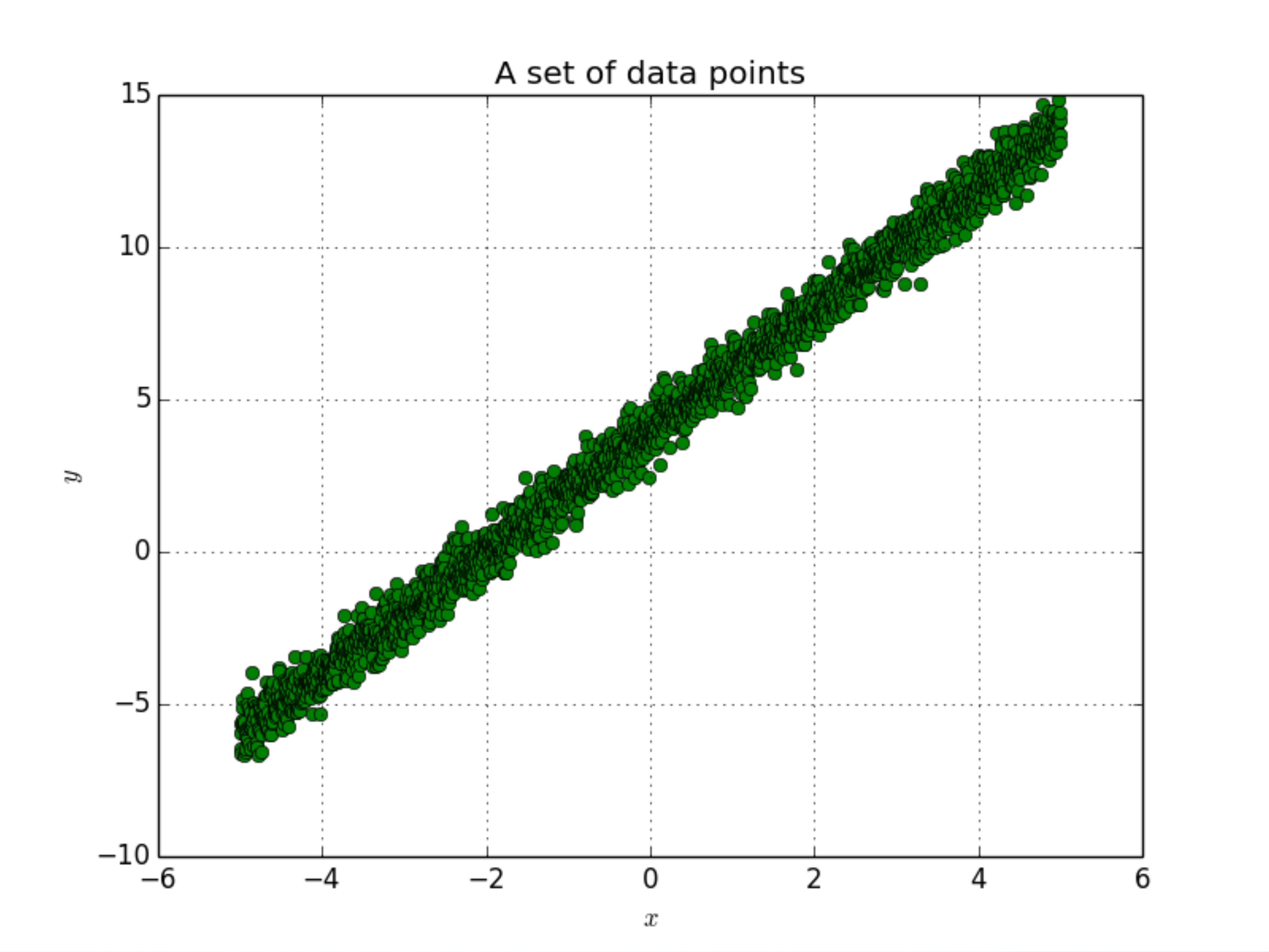
Input (**x**)	Label (y)
	Cat
	Bird
	Dog
	Fish
	?

SUPERVISED LEARNING

- Given a set of **input points** and corresponding **ground truth labels:** $\left\{\left(\vec{x}_1, y_1\right)\left(\vec{x}_2, y_2\right)...\left(\vec{x}_n, y_n\right)\right\}$
- Discrete labels: **Classification**
- Continuous labels: **Regression**
- Learn a **parameterised functional relationship** to **estimate** label from input: $\hat{y} = f(x;\theta)$
-

EXAMPLES

- Example 1:

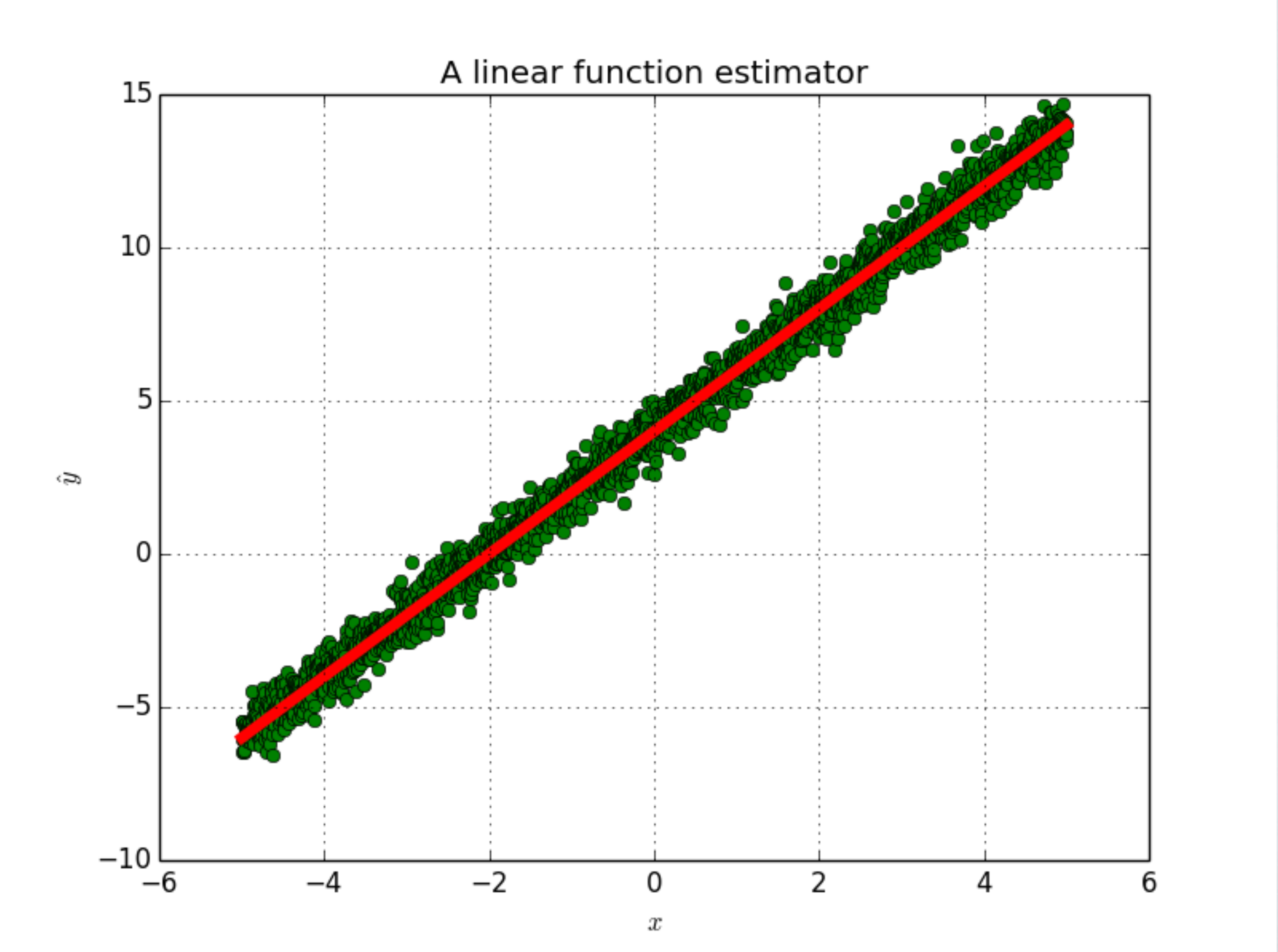


-

EXAMPLES

$$\hat{y} = mx + c \qquad \theta = \{m, c\}$$

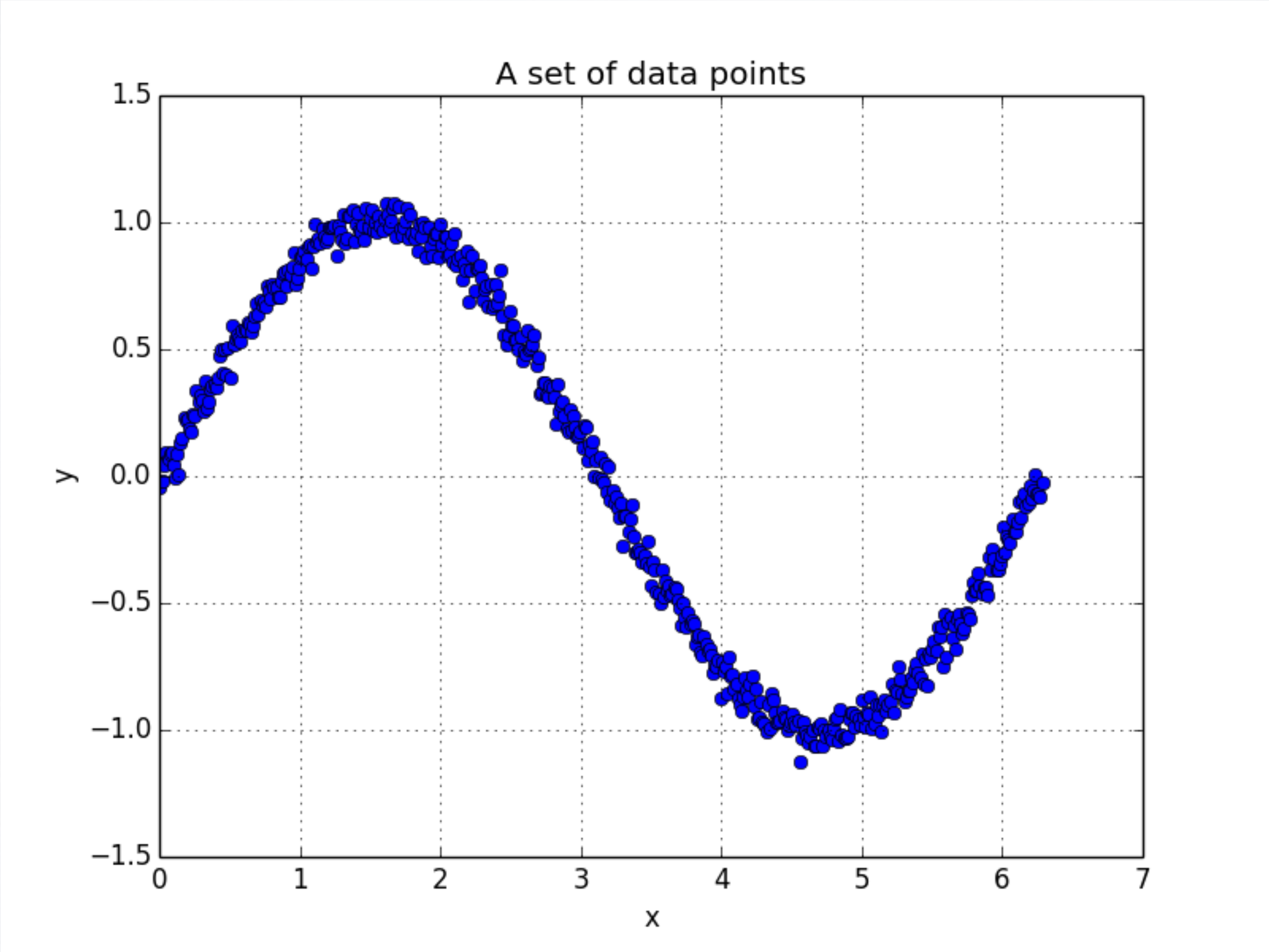
- Linear:



-

EXAMPLES

- Example 2:



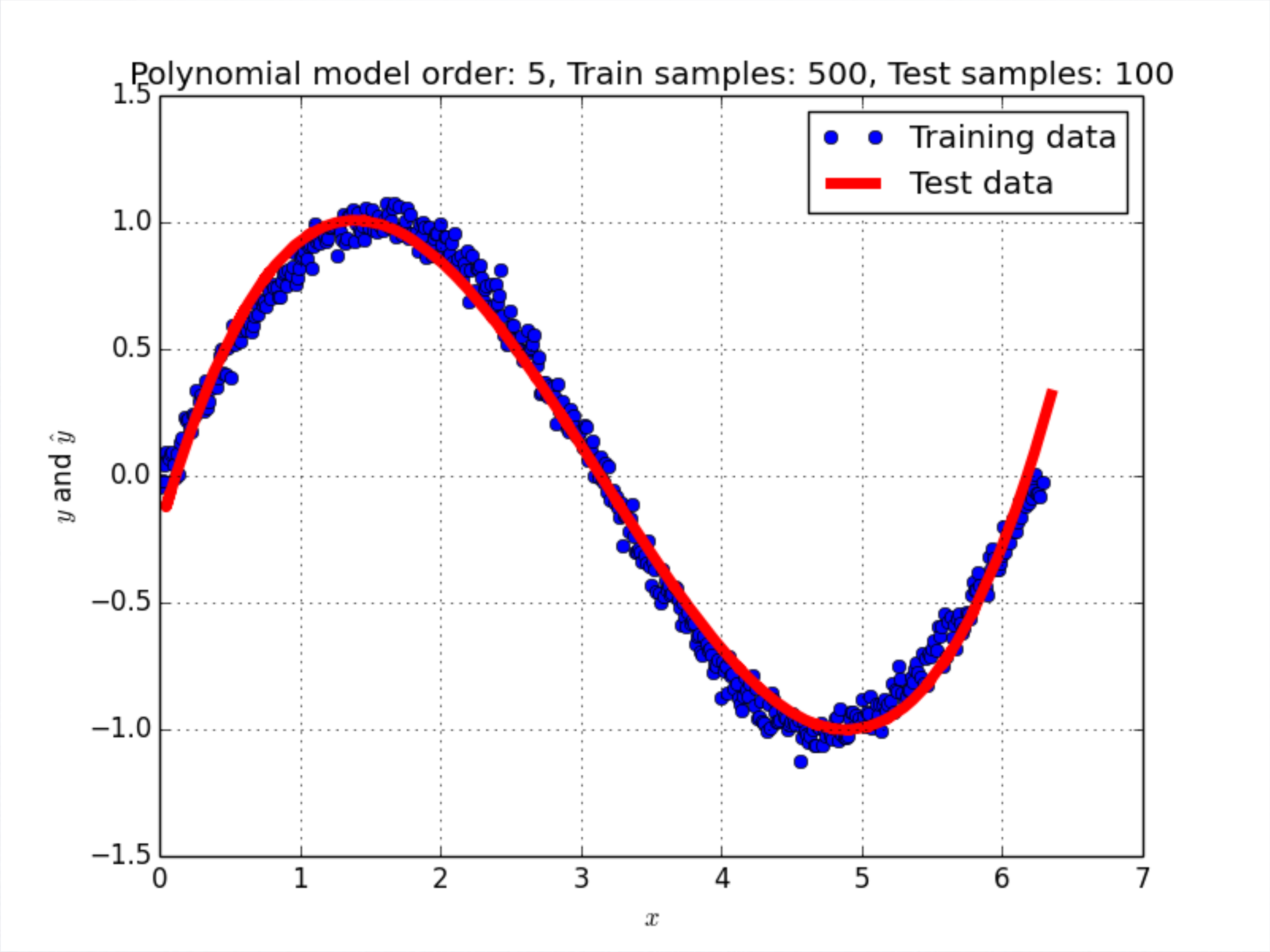
-

EXAMPLES

$$\hat{y} = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \text{---} w_n x^n$$

$$\theta = \{w_0, w_1, \ldots, w_n\}$$

- Linear:



-

EXAMPLES

- Example 3:

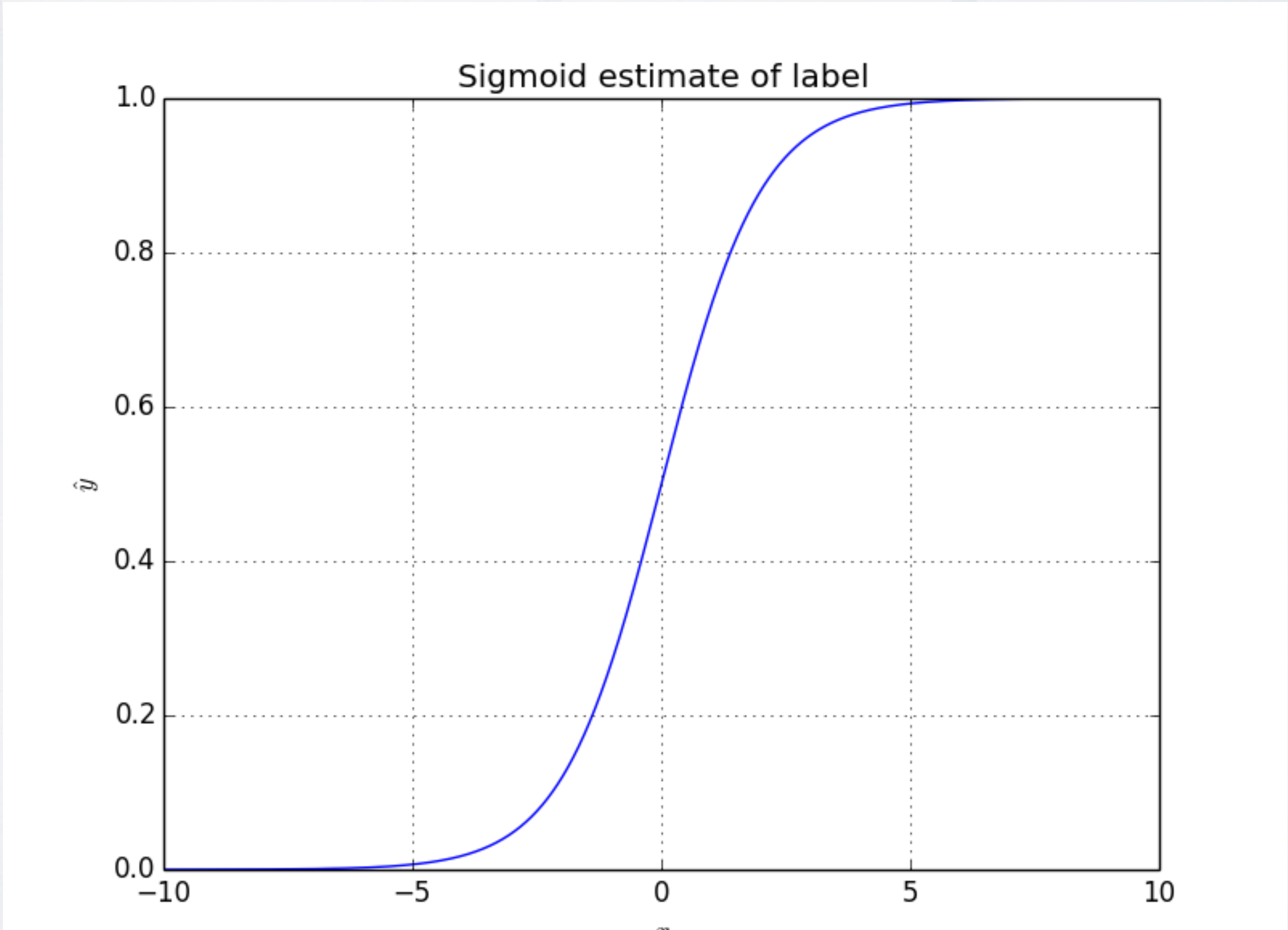
Input (**x**)		Label (y)
0	0	0
0	1	1
1	0	1
1	1	0

-

EXAMPLES

$$\hat{y} = \sigma(mx + c) \qquad \theta = \{m, c\} \qquad \sigma(x) = \frac{1}{1+e^{-x}}$$

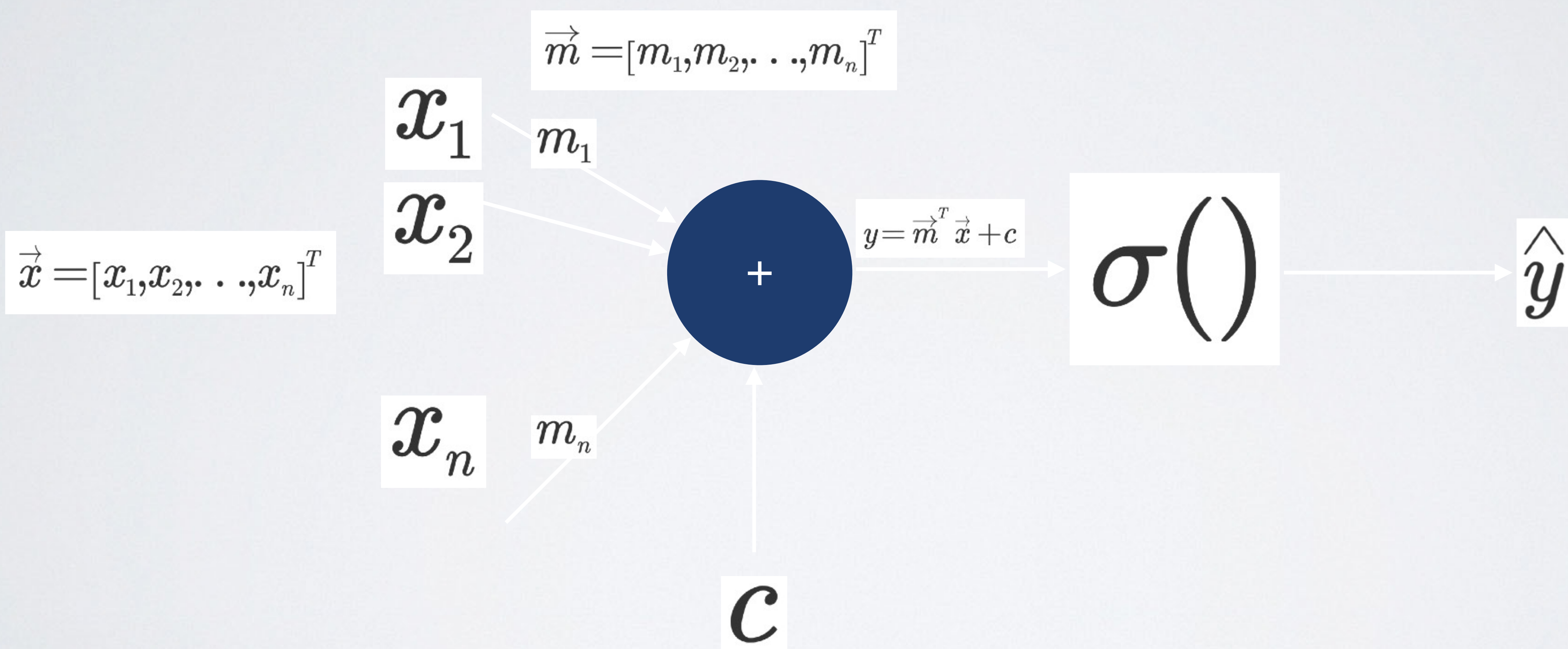
- Non-linear:



-

EXAMPLES

- Artificial Neuron: $\hat{y} = \sigma\left(\vec{m}^T \vec{x} + c\right)$ $\theta = \{\vec{m}, c\}$



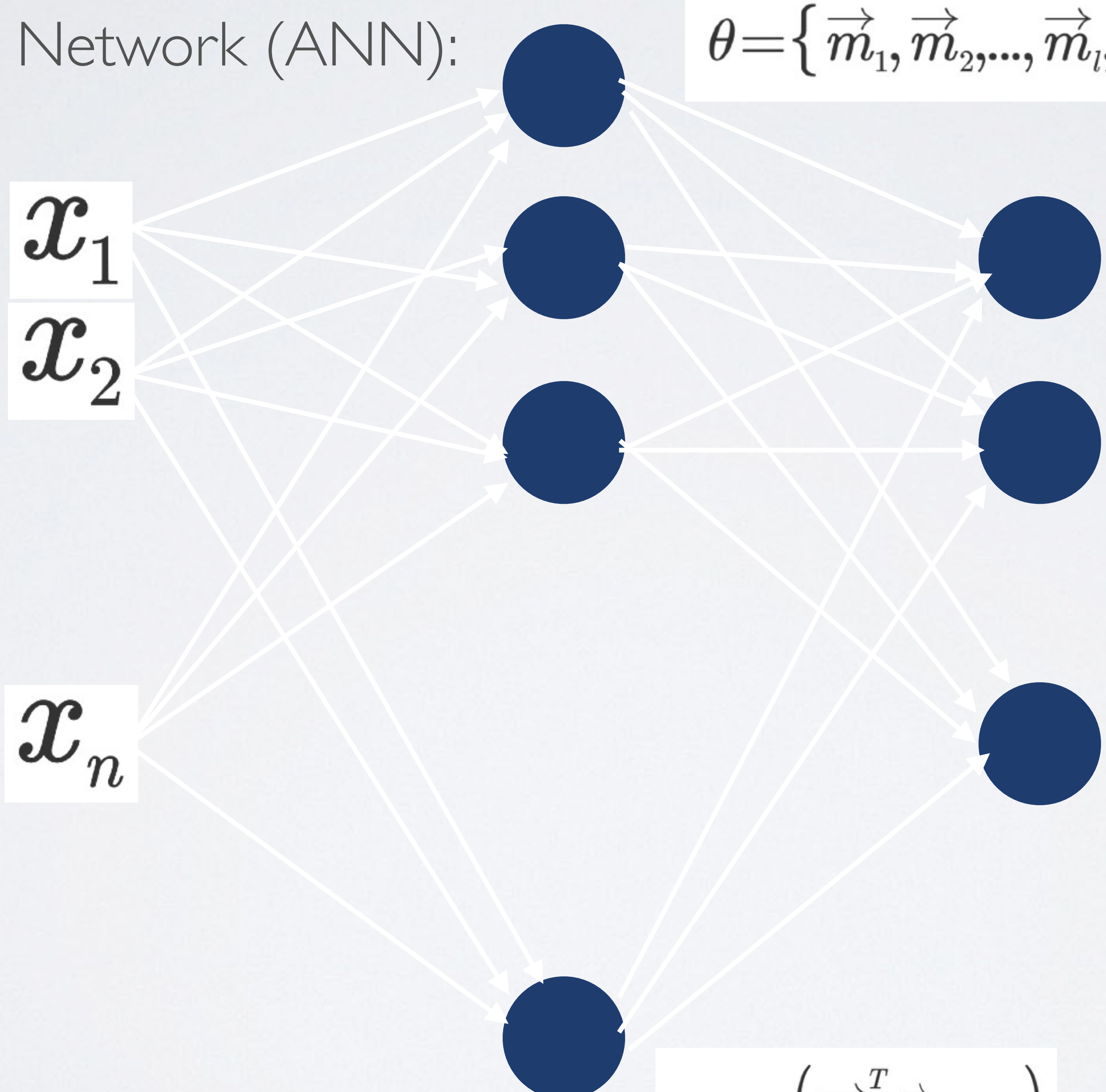
-

EXAMPLES

$$\hat{y}_k = g_k\left(\vec{w}_k^{\,T} \vec{z} + d_k\right) \quad z_i = \sigma\left(\vec{m}_i^{\,T} \vec{x} + c_i\right)$$

$$\theta = \left\{\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_l, c_1, c_2, \ldots c_l, \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_k, d_1, \ldots, d_k\right\}$$

- Artificial Neural Network (ANN):



$$\hat{y}_k = g_k\left(\vec{w}_k^{\,T} \vec{z} + d_k\right)$$

$$z_i = \sigma\left(\vec{m}_i^{\,T} \vec{x} + c_i\right)$$

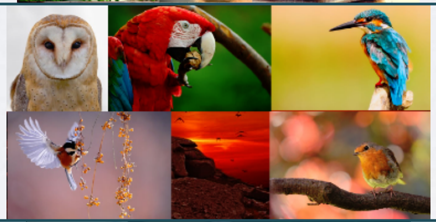
-

EXAMPLES

- ANN:
  - 2 input nodes
  - 2 hidden nodes
  - 1 output node
- Performance: very high accuracy of prediction!

EXAMPLES

- Example 4:
-

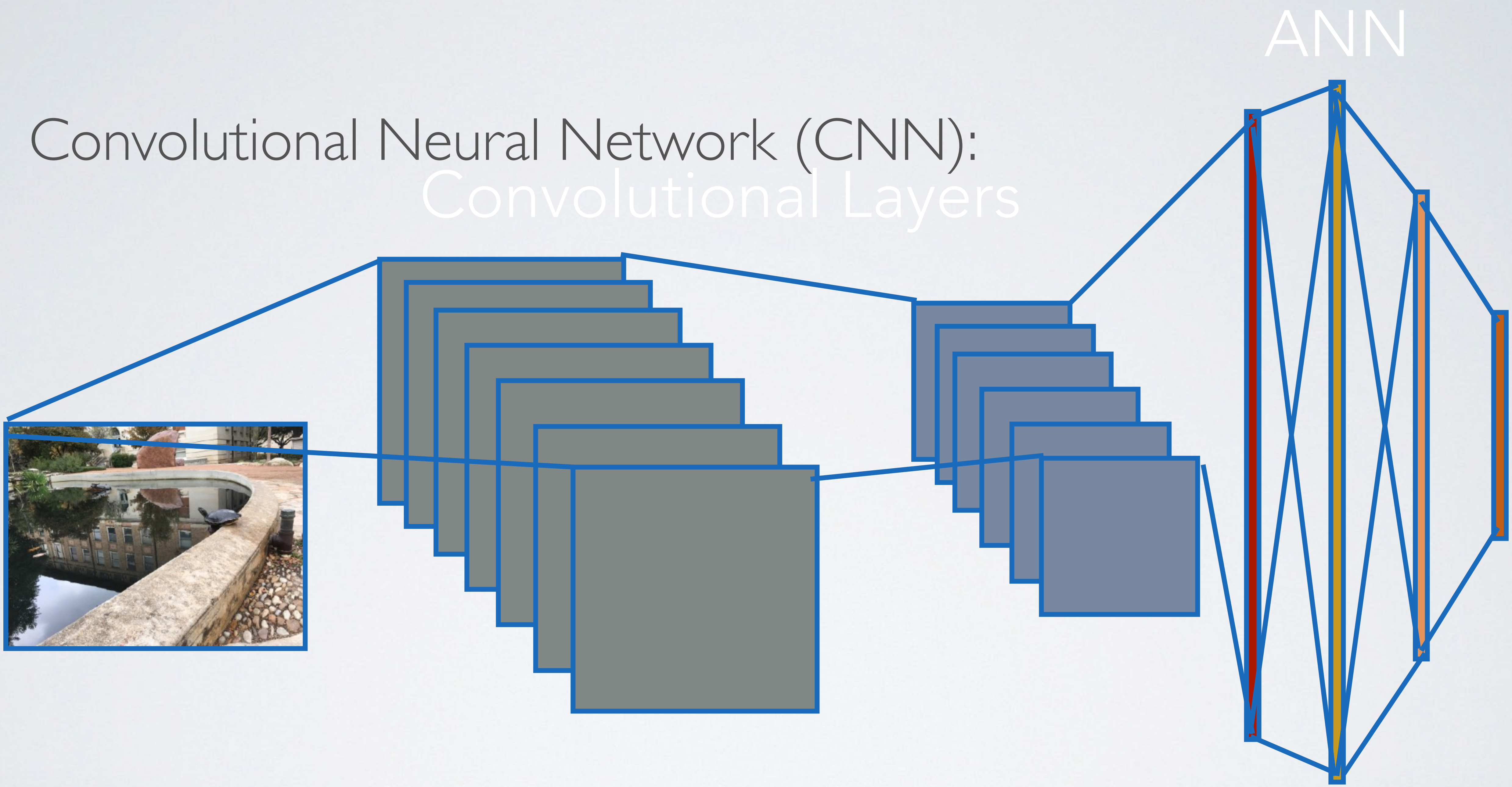
Input (**x**)	Label (y)
	Cat
	Bird
	Dog
	Fish
	?

EXAMPLES

- Questions:
  - Can we **directly** relate 2D inputs to labels?
  - Are all image pixels **equally important** for learning?
    - Can we learn an **effective representation** of images that is smaller (or much smaller) than the image?

EXAMPLES

- Convolutional Neural Network (CNN):

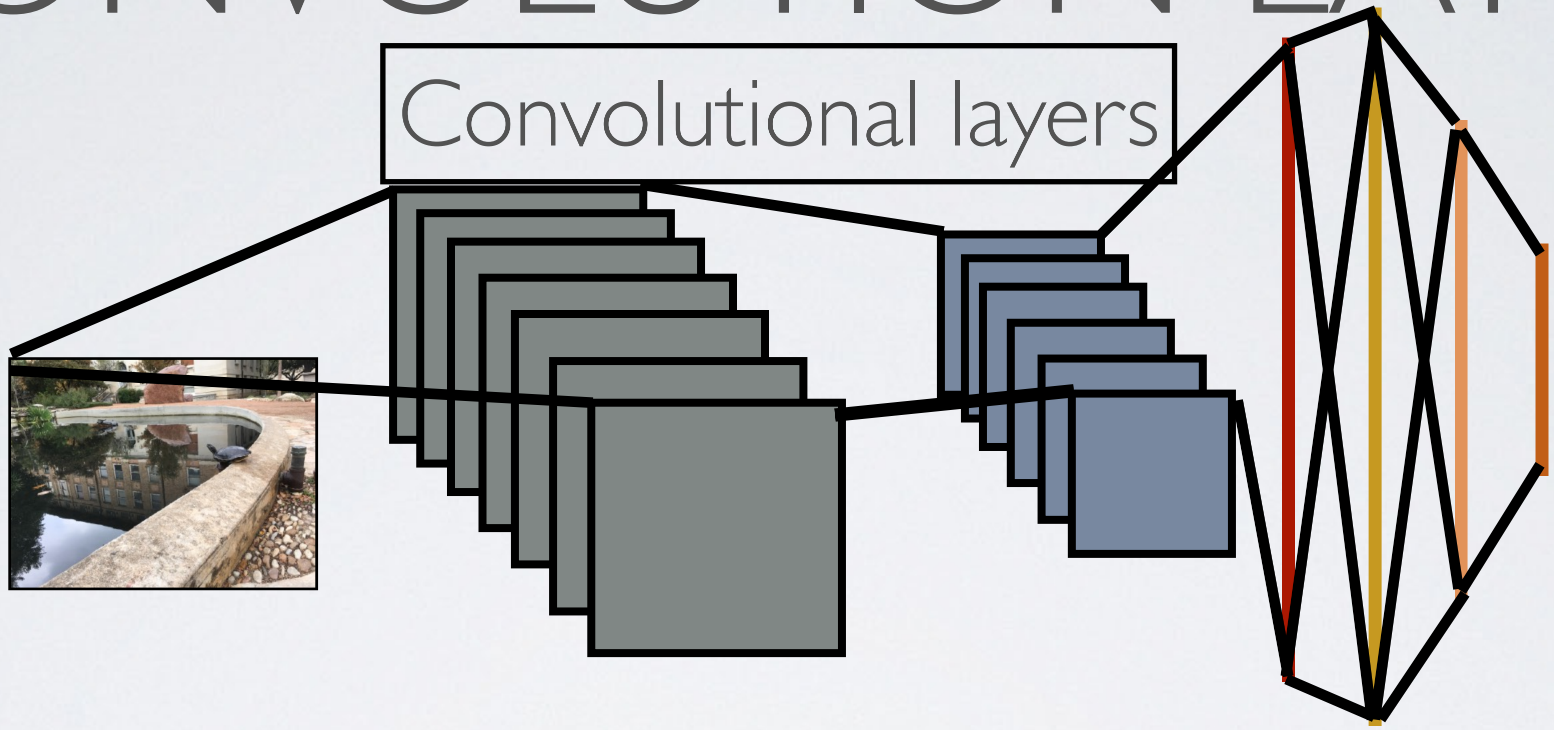


- Parameters: ANN + lots of filter weights!

EXAMPLES

- ResNet50 [https://arxiv.org/pdf/1502.01852.pdf]:
  - 152 layers!
  - ~25 million parameters!
- Won the ImageNet 2015 image recognition challenge
- **Surpasses** human recognition performance!

BUILDING BLOCK: CONVOLUTION LAYER



- **Linear component** of the network
- Parameter **sharing**
- **Local** connectivity

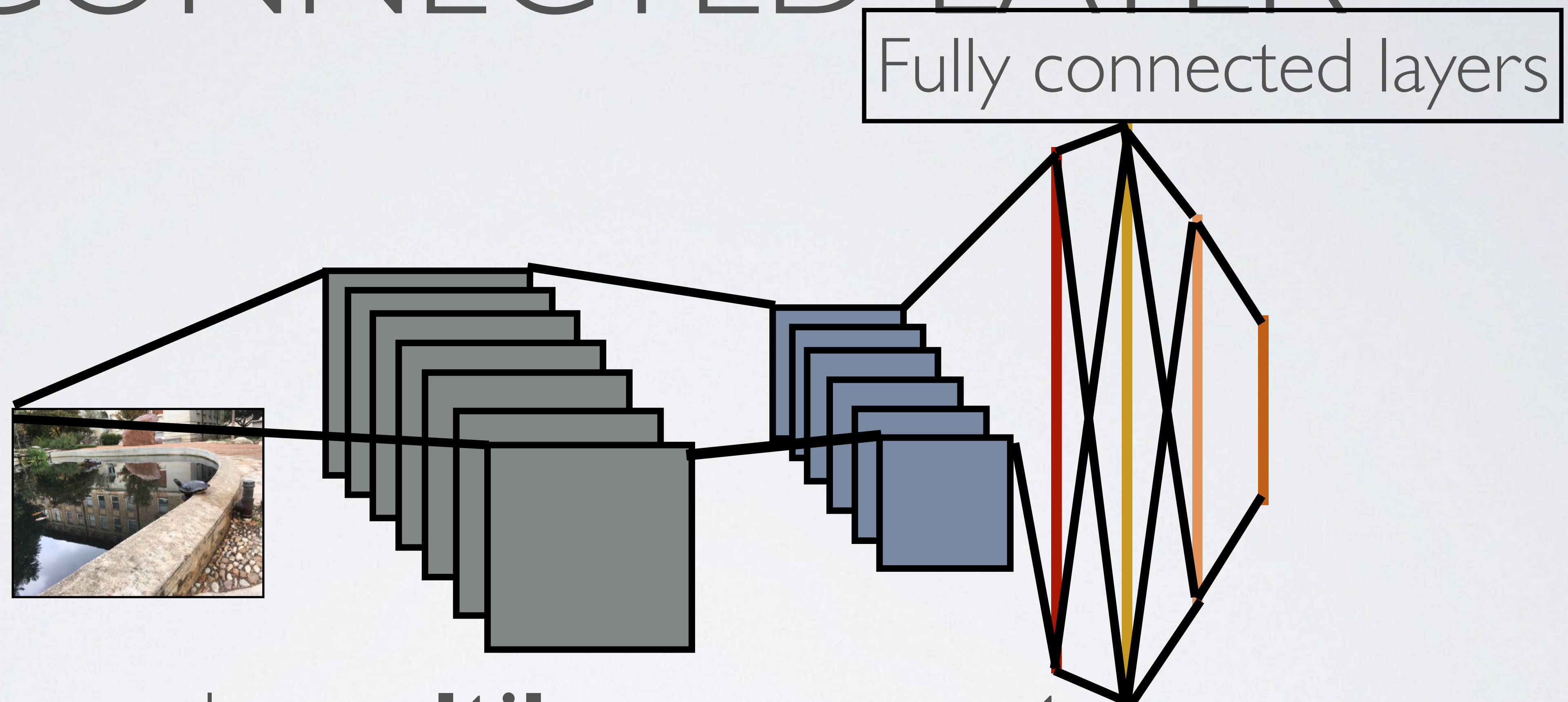
BUILDING BLOCK: NON-LINEARITY LAYER

- Fundamental to NN
  - Allows for **modelling complex input/output relations**
  - Inspired by the **neuron**
  - Examples: sigmoid, tanh, ReLU etc.
  - Also called **Activation Layer**

BUILDING BLOCK: POOLING LAYER

- Pooling of output
  - **Dimensionality reduction**
  - Control **overfitting**
  - **Invariant to small translation**
  - Examples: max, l2 norm, average etc.

BUILDING BLOCK: FULLY CONNECTED LAYER



- Essentially, a regular **multilayer perceptron**
- Input is a vector formed by **flattening** pooling layer output
- Output is vector whose size equals **number of classes**

HOW DO WE LEARN PARAMETERS?

- How **good** is our **estimate** compared to **ground truth**? Quantify error: $d\left(y, \hat{y}\right) = \left(y - \hat{y}\right)^2$

- Find **error** between **estimate** and **ground truth** over **all training data points**:

$$R(\theta) = \sum_{i=1}^{n} d\left(y_i, f(x_i; \theta)\right)$$

- Find **parameters** such that **error is optimised:** $\frac{\partial}{\partial \theta}[R(\theta)]$

TRAINING: LOSS FUNCTION

- Standard loss functions such as **softmax** or ***cross entropy***
- Loss functions are **non-convex** leading to **locally optimal** solutions

TRAINING: GRADIENT DESCENT BASED METHODS

- **Several** optimisation methods can be applied:
  - Gradient descent, Stochastic gradient descent
  - Momentum, Nesterov momentum
  - AdaGrad, RMSProp, Adam
- **Stopping** condition based on **training/validation error**
- Choice of **initialisation** important
- **Backpropagation** of forward loss used for gradient computation

TRAINING: PARAMETERS AND DATA

- Similar rules as in neural network training:
  - Weights typically **initialised randomly**
  - **Pre-trained weights** also used commonly
  - Feed data in **small batches (mini batches)**
  - **Epoch** is one forward and backward pass of all training data points

GENERALISATION

- Any model's performance is measured on **previously unseen inputs**
- This is also known as **generalisation**
- Typically, the dataset is divided into **training, validation and test set**
- Generalisation performance is reported on the **test set**

WHY DID DEEP LEARNING SUCCEED?

- **Non-linearity** key to success of neural networks
- Deeper networks allow for **greater flexibility** to learn complex input-label relations
- Excellent **generalisation** ability

WHY DID DEEP LEARNING SUCCEED?

- The **backpropagation algorithm** led to the resurgence of neural networks
  - An **iterative approach** to **updating parameters**
  - Estimates gradients by **back-propagating errors**

WHY DID DEEP LEARNING SUCCEED?

- **Lots of data** available to learn the parameters of the network
- Availability of powerful **Graphics Processing Units (GPUs)** for training deep networks

IS DL RIGHT FOR YOUR PROBLEM?

- Do you have **lots of labeled data** (preferably images) and want to make **predictions** on new (unseen) data?
- Do you have **some labeled data** that is **similar** to larger labeled databases?
- Do you want to learn **efficient representations** of your data?
- Do you want to **generate new data samples** from existing samples to augment your data?

SUMMARY

- CNNs form the **fundamental building blocks** of modern machine learning models
- Allow for **feature learning**
- **Extremely successful** in solving several machine learning problems: image recognition, video analysis, natural language processing, drug discovery, visual system modelling etc.
- **Reason for the deep learning revolution**

REFERENCES

- https://www.deeplearningbook.org
- https://cs231n.github.io/convolutional-networks/