Deep Learning

Lecture 1: Introduction to Deep Learning

National University of Singapore

CEG5304/EE6934

Instructor: Joey Tianyi Zhou

Background

Centre for Frontier AI Research
CFAR

Investigator
Principal Scientist

Email: zhouty@cfar.a-star.edu.sg

NUS National University of Singapore
Department of Electrical & Computer Engineering

Assoc. Prof.
(adj.)

Email: eleztyi@nus.edu.sg

Robustness

- Adversarial Attack/Defense
- Trustworthy Learning

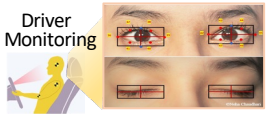
Efficiency

- Model/dataset Compression
- Efficient Training
- Transfer Learning

On-device AI Applications

Sony "Vision-S"

Eyeblink Detection*



Confidential

Centre for Frontier AI Research (CFAR)

- Established in 2022 (Planned in 2021)
- Aims to advance use-inspired basic research in AI
- Focus on the following strategically important fundamental research

Artificial General Intelligence (AGI)

- Neuro-symbolic AI
- Causal Reasoning
- General AI Agents
- Embodied AI
- AI for Metaverse

Resilient and Safe AI

- Adversarial attacks and defense
- Trustworthy AI
- AI for Safety and Security
- Self-aware Continuous Learning
- Sensemaking in AI

Sustainable AI

- Resource-efficient AI
- Data-efficient AI
- Quantum machine learning
- AI for Hardware
- AI for Sustainability

CFAR Internship Award for Research Excellence (CIARE)

Why?

- Provides undergraduate students an opportunity to gain research experience under supervision with CFAR
- Outstanding students may continue their research under local postgraduate degree

Who are we looking for:

- Students undertaking research programmes such as the Undergraduate Research Experience on NTU CAmpus (URECA) and Undergraduate Research Opportunity Programme (UROP)
- NUS Full Semester Full Time Internship Program
- SUTD Full Semester Full Time Internship Program

CIARE

- Internship Award for selected students
- Short-term Project Proposals from CFAR scientists
- Hand-on research for interns

Follow up

- Students have excellent outcomes from CFAR summer internship can be nominated for ARIA or other A*STAR internship programs for long-term internship
- For 3rd/4th students who have interests to pursuit PhD study, can be nominated for AGS/ACIS program.

CFAR PhD Support for Research Excellence (CPSRE)

Why?

- Attract Top-tier PhD candidates
- Provides additional support for Conference Trip/Research Visit
- Provides additional support for Research related equipment and expense

Who are we looking for:

- Applicants intending to pursue a Full-time PhD and existing PhD candidates (must be 2 years or less into PhD) in Singapore-based AUs
- Applicants would be new applicants for A*STAR PhD scholarship / AISG PhD scholarship affiliated with CFAR or existing PhD students receiving A*STAR PhD scholarship / AISG PhD scholarship will be affiliated with CFAR

CPSRE

- SGD6,000 annual support

More Awareness of Opportunities Under CFAR

CFAR aims to cultivate next-gen AI scientists in partnership with A*STAR Graduate Academy.

Post-PhD Investigatorship

- A*STAR CFAR Research Grant (A*CRG)
- A*STAR CFAR Early Career Research Grant (A*CECRG)

Internships

- A*STAR Research Internship Award (ARIA)
- A*STAR International Pre-Graduate Award (SIPGA)

PhD Scholarships

- A*STAR Computing & Information Science (ACIS)
- A*STAR Graduate Scholarship (AGS)
- A*STAR International Graduate Award (SINGA)
- A*STAR Research Attachment Programme (ARAP)

Monthly Sustenance Allowance Supplement available for select candidates

- AISG Ph.D Scholarship

Come join us at CFAR!
Email us at talent@cfar.a-star.edu.sg

Lecturers

- Dr. Joey Zhou, for Part I
- Dr. Xu Yuecong, for part II

Teaching Assistants

Mr. Qiqi Tao(email: tao.qiqi@u.nus.edu) (Project 1 for master students)
Mr. Ziteng Gao(email:ziteng.gao@u.nus.edu) (Project 1 for Ph.D students)
Dr. Xu Yuecong (email:yc.xu@nus.edu.sg) (Project 2 for both two modules)

Q&A Google doc:

https://docs.google.com/document/d/1MeICyhVWL8Kw400YEECtuvyFaA5BmFChCIW8ycrDSog/edit?usp=sharing
Mr. Yuchao Gu(email:yuchaogu@u.nus.edu) (General Q&A)

Acknowledgements

For various materials from:

- *COS 495 Introduction to Deep Learning,* Princeton University
- *EE 2211 Introduction to Machine Learning,* NTU
  - *11-785 Introduction to Deep Learning*, Carnegie Mellon University
  - *MIT 6.S191 Introduction to Deep Learning*, MIT
  - *C19 Machine Learning*, Oxford
  - *CS276 Information Retrieval & Web Search* , Stanford University
  - *CS213N Convolutional Neural Networks for Visual Recognition*, Stanford University
  - *CS294-129 Designing, Visualizing & Understanding Deep Neural Networks*, UC Berkeley
  - *CS498 Introduction to Deep Learning*, UIUC
  - *CSC321 Introduction to Neural Networks & Machine Learning*, University of Toronto
  - *CS 898: Deep Learning and Its Applications*
    Introduction to Neural Networks & Deep Learning, UVA
  - etc.

Module Learning Outcomes

- To provide a comprehensive understanding of current methodologies used in deep learning, particularly Deep Convolutional Neural Network and Transformer models;
- Be informed of recent progress within the deep learning field;
- To discuss the weaknesses and strengths of popular deep learning methods/architectures.
- Able to design, build and implement deep learning models;
- Apply deep learning methods to solve practical problems in areas such as image processing, natural language processing and computer vision.

Pre-requisites

- **EE5907 Pattern Recognition**
- **Proficiency in Python**, high-level familiarity with C/C++
  - assignments will be in Python with some deep learning libraries in C++.
- Calculus, Linear Algebra
- Probability Theory & Statistics

Textbook and materials

- Deep Learning:

  http://www.deeplearningbook.org/

PYTHON is a **general-purpose programming language** which becomes a powerful environment for scientific work with several libraries: `numpy`, `scipy`, `matplotlib`.

PYTHON is almost like pseudocode and allows expression of powerful ideas in very few lines of code.

- Other software frameworks: Pytorch, Tensorflow, Caffe, Torch, Marvin, …

Some Python Resources

Installing scikit-learn (Ref: [Book2] Andreas C. Muller and Sarah Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists", O'Reilly Media, Inc., 2017)

scikit-learn depends on two other Python packages, *NumPy* and *SciPy*. For plotting and interactive development, you should also install matplotlib, IPython, and the Jupyter Notebook. We recommend using the following prepackaged Python distributions, which provides the necessary packages:

Anaconda

A Python distribution made for large-scale data processing, predictive analytics, and scientific computing. Anaconda comes with NumPy, SciPy, matplotlib, pandas, IPython, Jupyter Notebook, and scikit-learn. Available on Mac OS, Windows, and Linux, it is a very convenient solution and is the one we suggest for people without an existing installation of the scientific Python packages. Anaconda now also includes the commercial Intel MKL library for free. Using MKL (which is done automatically when Anaconda is installed) can give significant speed improvements for many algorithms in scikit-learn.

Some tutorials that might be useful:

A quickstart tutorial on NumPy: https://numpy.org/devdocs/user/quickstart.html

Some community tutorials on Pandas: https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html

Scikit-learn tutorials: https://scikit-learn.org/stable/tutorial/index.html

Code of Conduct.

Students are encouraged to **actively participate**.

Your grade depends on what you do, not what others do..

Don't share your solutions/codes… what you submit should be your own work..

Academic Integrity:

Student's academic work & behavior are held to highest academic integrity standards. Cheating, plagiarism, unauthorized collaboration, and 'helping others commit these acts' are examples of academic misconduct, which can result in disciplinary action.

Any violation to academic honesty will be reported to the University's disciplinary panel.

Topics (PART I- Dr. Joey Zhou)

Lecture 1: Introduction & DL Basics: History of Deep Learning. Machine Learning Settings. Basic terminology and concepts: e.g., common learning methods, common network structures, etc.

Lecture 2: Neural networks, training pipeline and learning strategies: MLP, SGD, Loss Optimization, Gradient Descent. Regularization.

Lecture 3: Autoencoder, CNN and RNN and their Variants: Explain the reason for the rise and fall of CNNs and RNNs.

Lecture 4: Transformers : Introduction to self-attention and Transformer model, their applications to both NLP and CV.

Lecture 5: Prompting : Prompting in both NLP and CV domains.

Lecture 6: Model and Dataset Compression: Model Compression including pruning, distillation, and quantization. Dataset Distillation and Pruning.

Grading

- CEG5304 and EE6934
- Projects #1: 30%
- Projects #2: 30%
- Final: 40%

*Projects (EE6934 has extra question) and Final (Different)

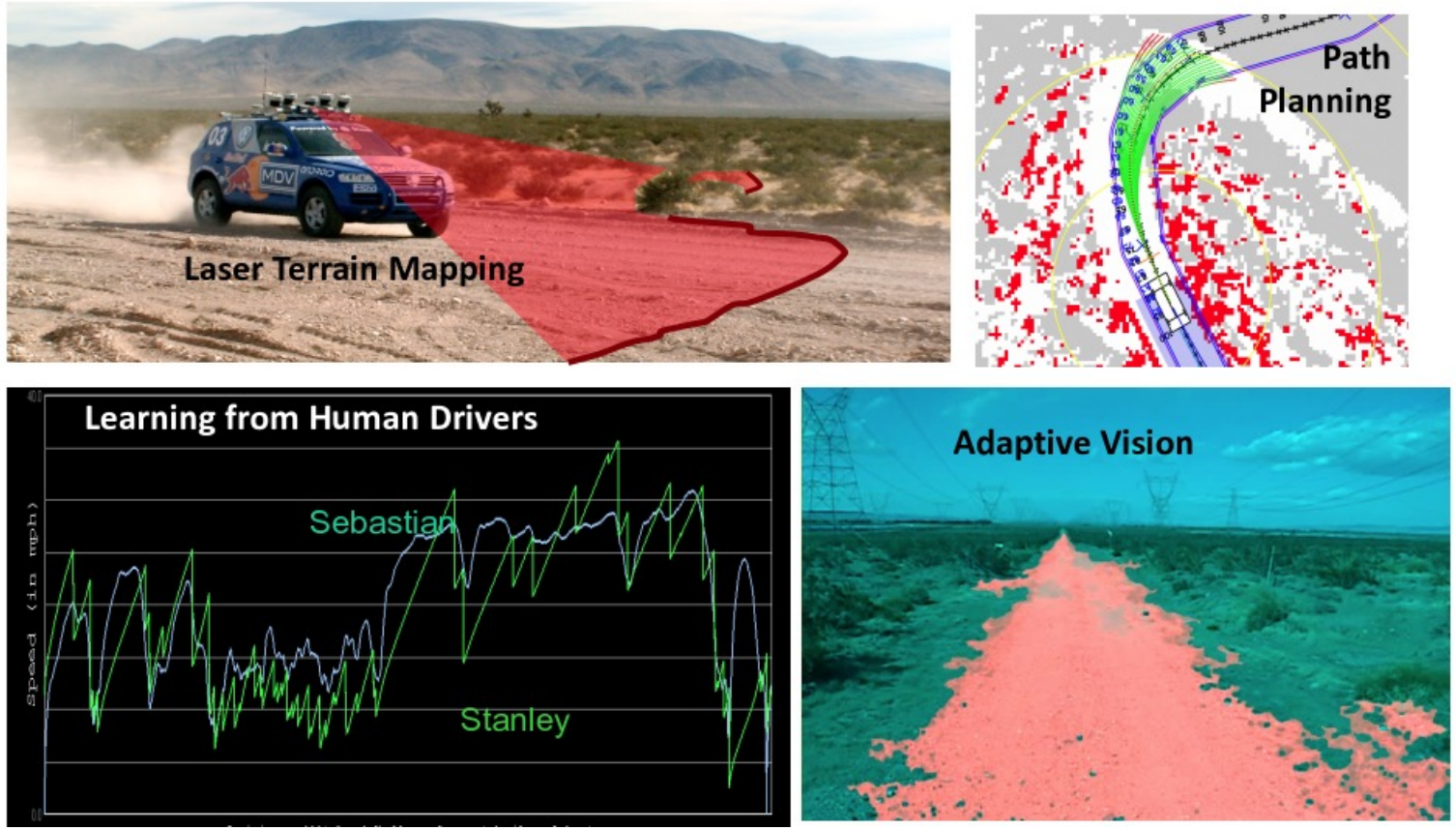
Project #1 (Deadline: 01 Mar 2024 6pm)
Project #2 (15 Mar 2024 – 19 Apr 2024 6pm)
Final Time: the week after recess week

Deadlines are strict. Late submission will be deducted 10 points (out of 100) for every 24 hours.

Introduction to deep learning

Deep learning in our lives

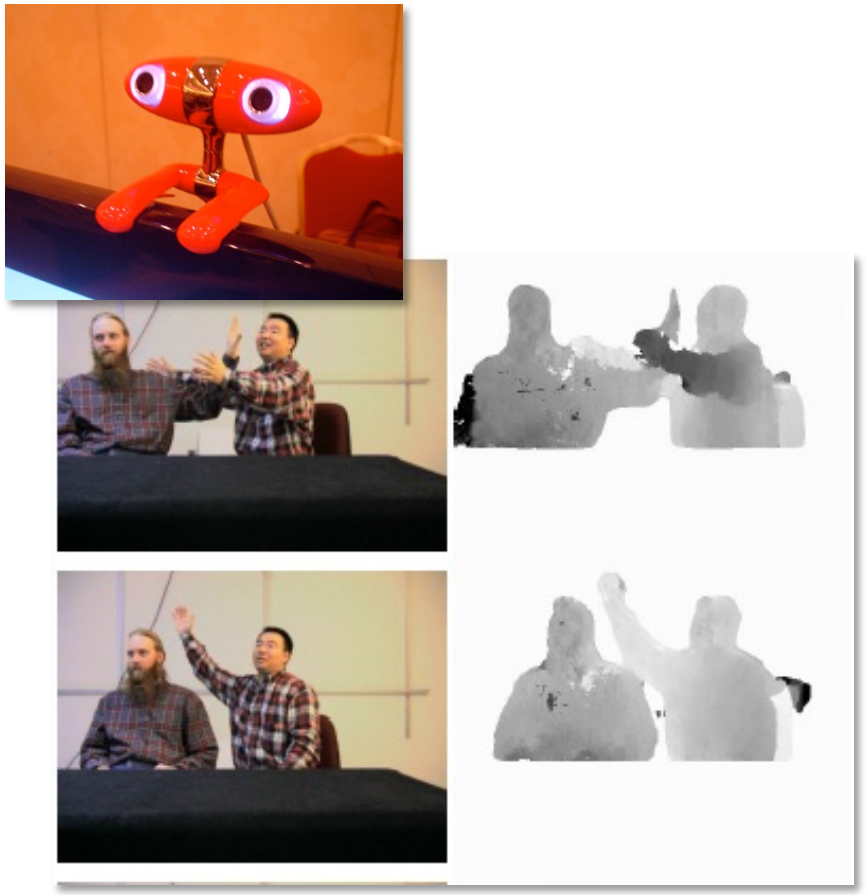
- Autonomous car technology

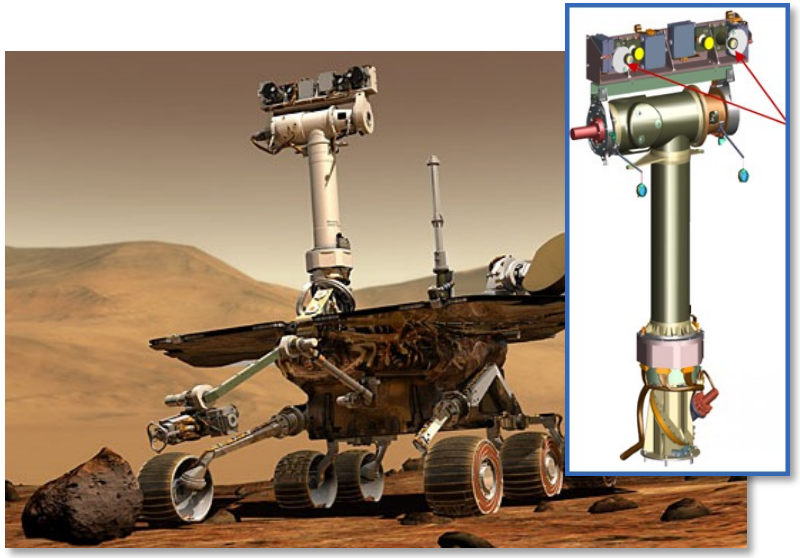


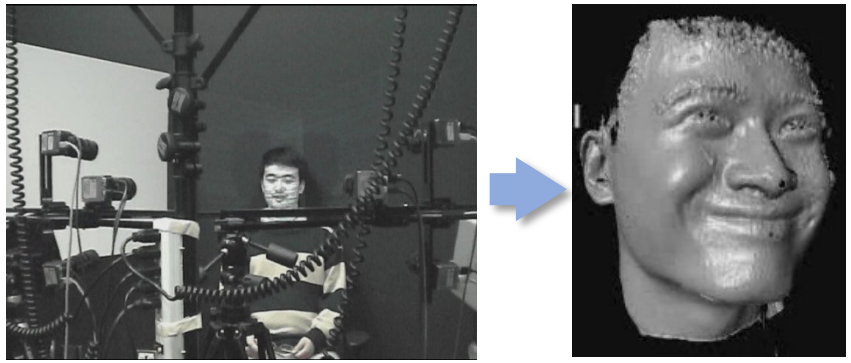
Deep learning in our lives

- Image Processing

Deep learning in our lives



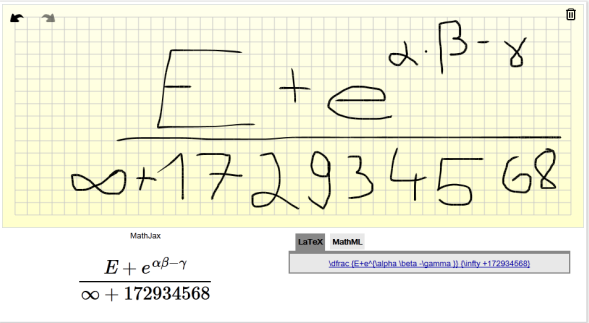




Deep learning in our lives

Automatic check processing

OCR



Equation Recognition

ChatGPT 3.5

ChatGPT

You

Please briefly introduce deep learning to my students.

ChatGPT

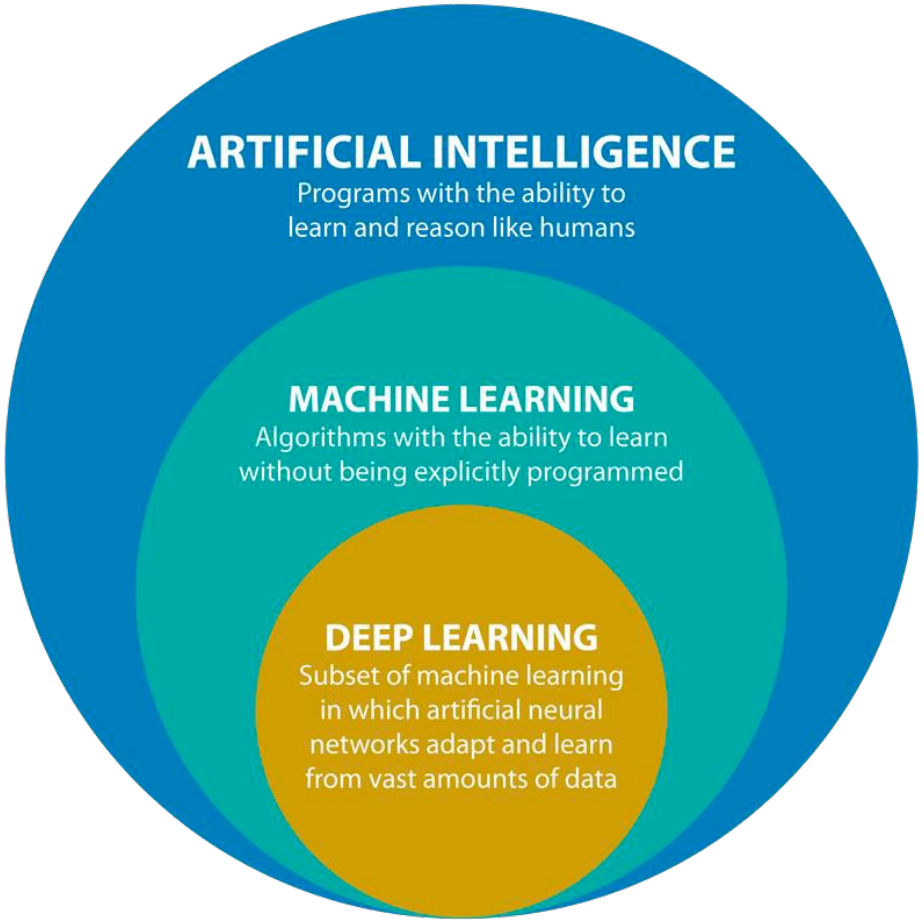
Certainly! Deep learning is a subset of machine learning, which is a broader field of artificial intelligence. It's inspired by the structure and function of the human brain, specifically neural networks. Here's a brief introduction to deep learning:

1. **Neural Networks:**
   - At the core of deep learning are neural networks, which are computational models inspired by the way biological neural networks in the human brain work.
   - These networks consist of layers of interconnected nodes (neurons). Information flows through the network, and each connection has a weight that is adjusted during training.
2. **Deep Neural Networks:**
   - "Deep" in deep learning refers to the depth of the neural networks, indicating the presence of multiple layers (deep architectures).
   - Deep neural networks are capable of learning intricate patterns and representations, allowing them to perform complex tasks.
3. **Training and Learning:**
   - Deep learning models learn from data through a process called training. During training, the model adjusts its internal parameters (weights) to minimize the difference between its predictions and the actual outcomes in the training data.
   - This is often done using a technique called backpropagation, where the model updates its weights based on the error it made during the prediction.
4. **Applications:**
   - Deep learning has demonstrated remarkable success across various domains, including image and speech recognition, natural language processing, and playing strategic games.
   - Examples of applications include image classification (identifying objects in images), speech recognition (converting spoken language into text), and language translation.
5. **Deep Learning Frameworks:**
   - There are several popular deep learning frameworks, such as TensorFlow and PyTorch, that provide tools and libraries to build and train deep neural networks.
6. **Challenges:**
   - Despite its success, deep learning faces challenges, including the need for large

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

What is the difference between DL (Deep Learning), ML (Machine Learning) and AI (Artificial Intelligence)?



- Artificial Intelligence(AI) is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”.
- Machine Learning (ML) is a current application of AI based around the idea that we should really just be able to give machines access to data and let them learn for themselves.
- Deep Learning (DL) is the most dominant method in ML. Artificial NNs adapt and learn from vast amounts of data.

What is human learning?

Human learning vs. Machine learning

- We can understand machine learning by reviewing <u>how we learn as a kid</u>.
- Your lecturers may use many different examples to help to learn some mapping relationship between 1, 2, 3,4… to their corresponding meanings. <u>Let us look at the video below</u>:

https://youtu.be/t7w3PHgvKJE

Human learning vs. Machine learning

- After the human learning, we know 8 means 8 objects (bees below) or 9 means we can count objects for nine times.

- The same process for machine learning. It aims at teaching the machines to understand what 8 and 9 in the two images mean.

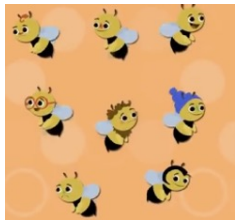
Human learning vs. Machine learning

- Our human brain is very powerful. After the human learning process, we know that the following digits all have the same meaning.

Input

Mapping

Label



8 objects

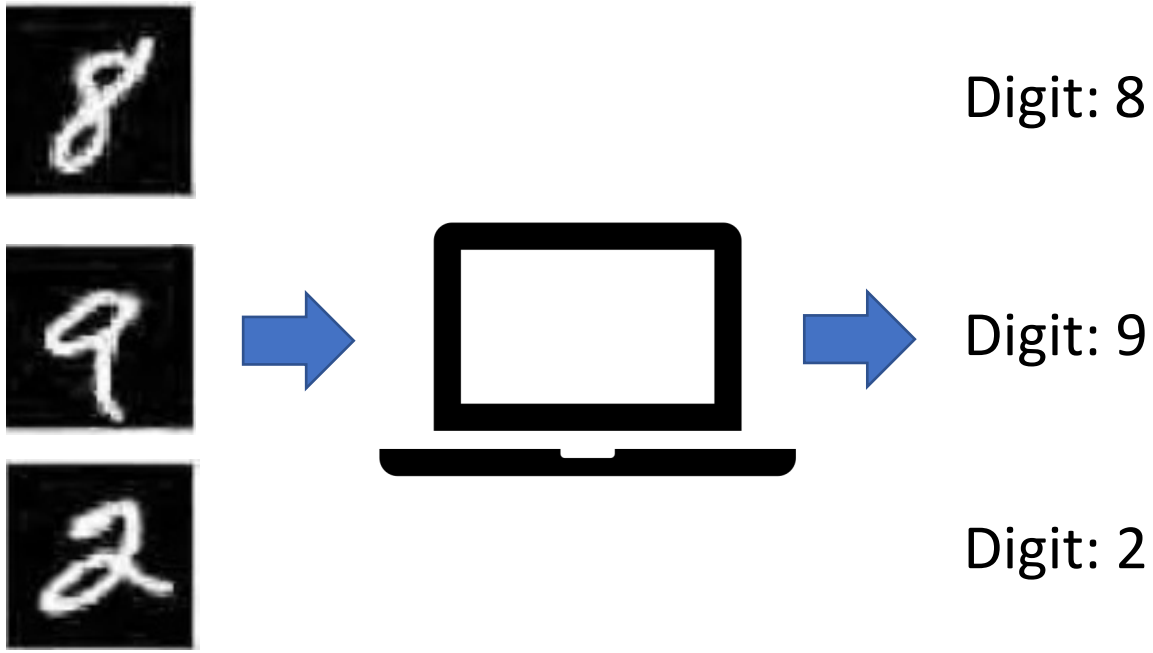
Human learning vs. Machine learning

- Our human brain is very powerful. After the human learning process, we know that the following digits all have the same meaning.

- You may also feel some difficulties in recognizing the last one. At this time, an experienced teacher will help you know its meaning again.
- Another example is our today's course, after the human learning, you can build a mapping between machine learning and its meanings in your mind.

Human learning vs. Machine learning

- This is same to the machine learning process. More specifically, we can also call this as pattern recognition.
- During the machine learning, we are training machines know the following mapping correlation:



Human learning vs. Machine learning

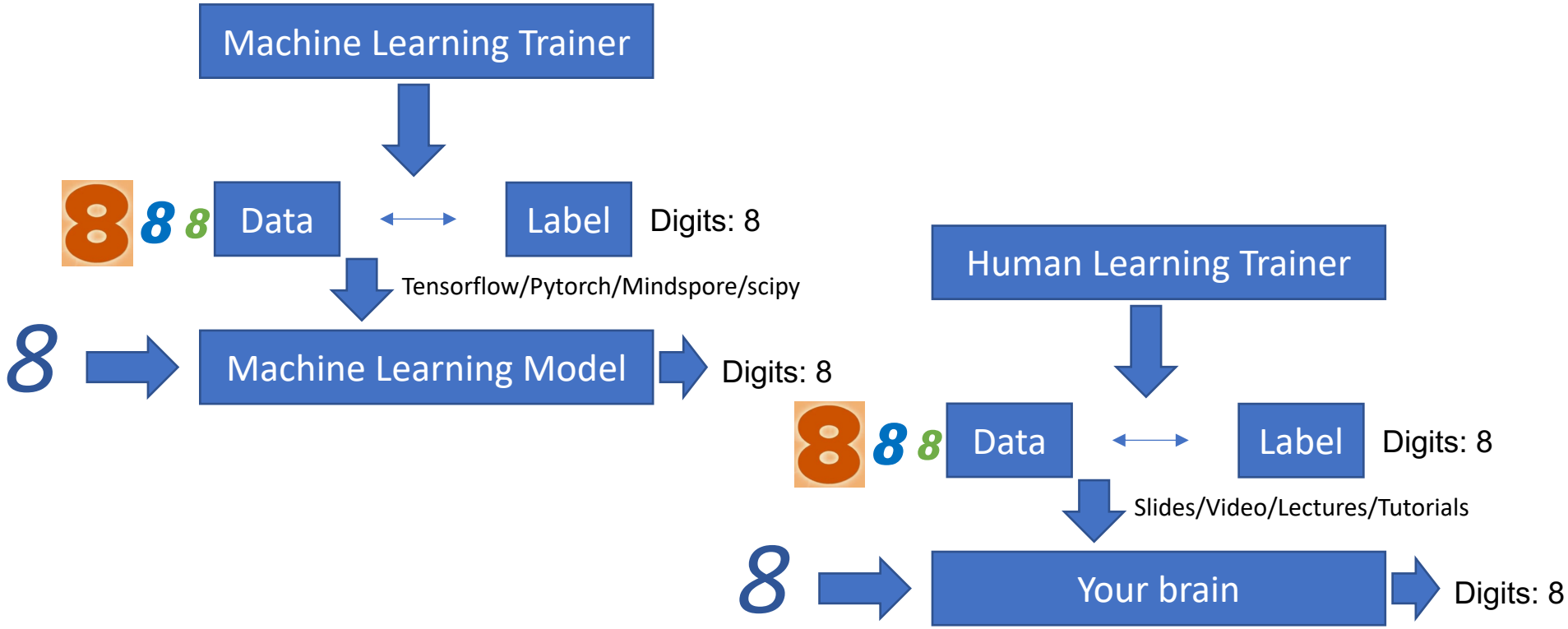
- However, the machine is very stupid. When you are training it by only using the first image, it cannot correctly recognize the second one.

This is overfit, since your machine can only recognize the digits you provide for training.

- We need to provide the machines with different examples to help it know different styles.

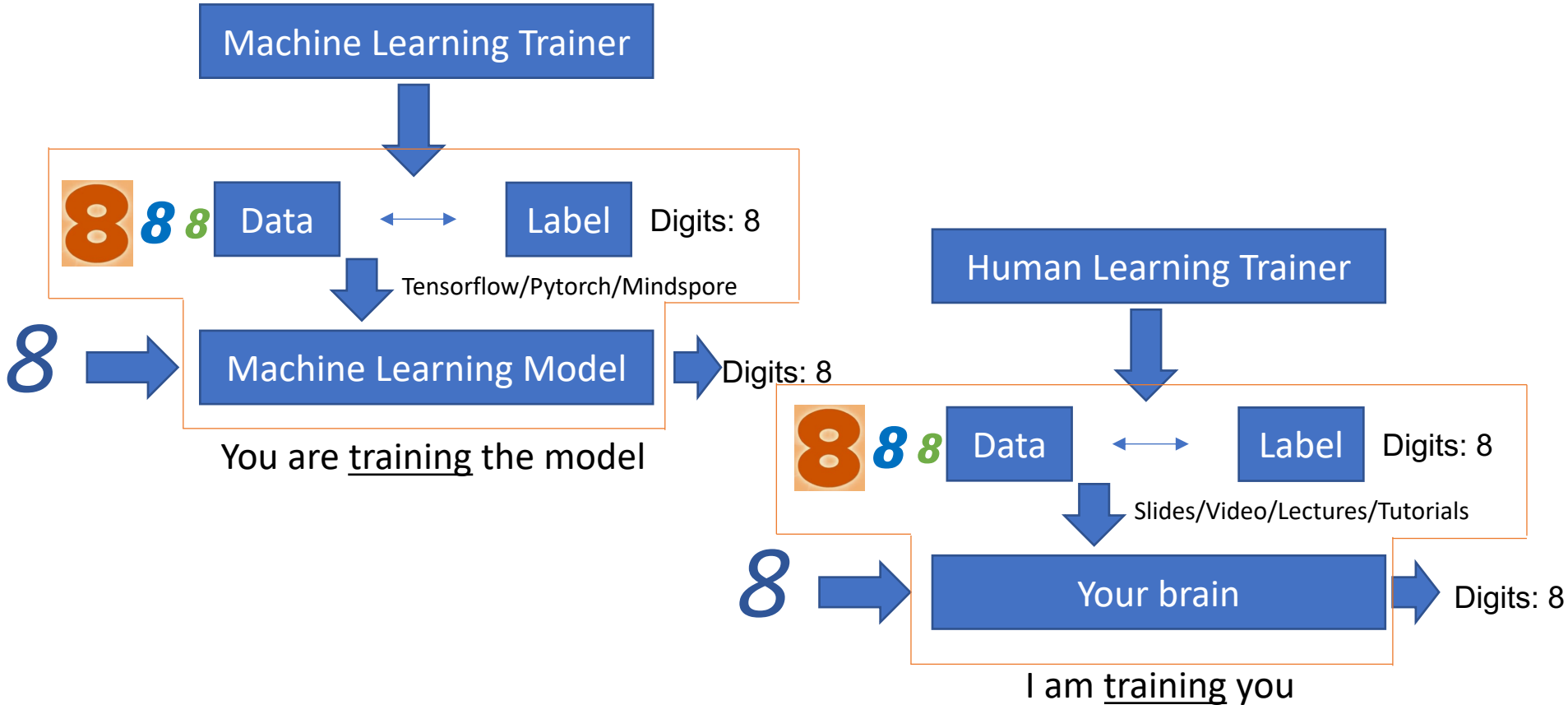
Human learning vs. Machine learning

- Let us summarize the difference between human learning and machine learning.



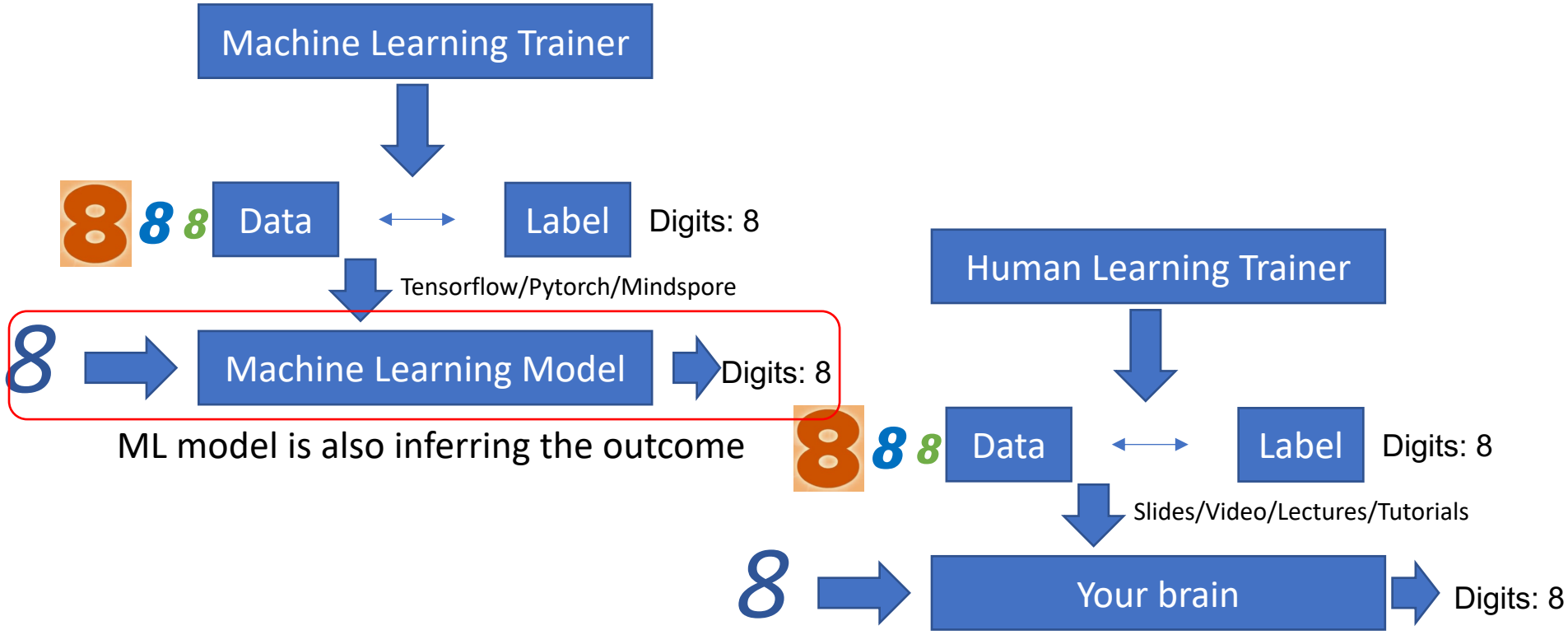
Human learning vs. Machine learning

- Let us summarize the difference between human learning and machine learning.



Human learning vs. Machine learning

- Let us summarize the difference between human learning and machine learning.



ML model is also inferring the outcome

You are <u>inferring</u> the outcome based on the training

Human learning vs. Machine learning

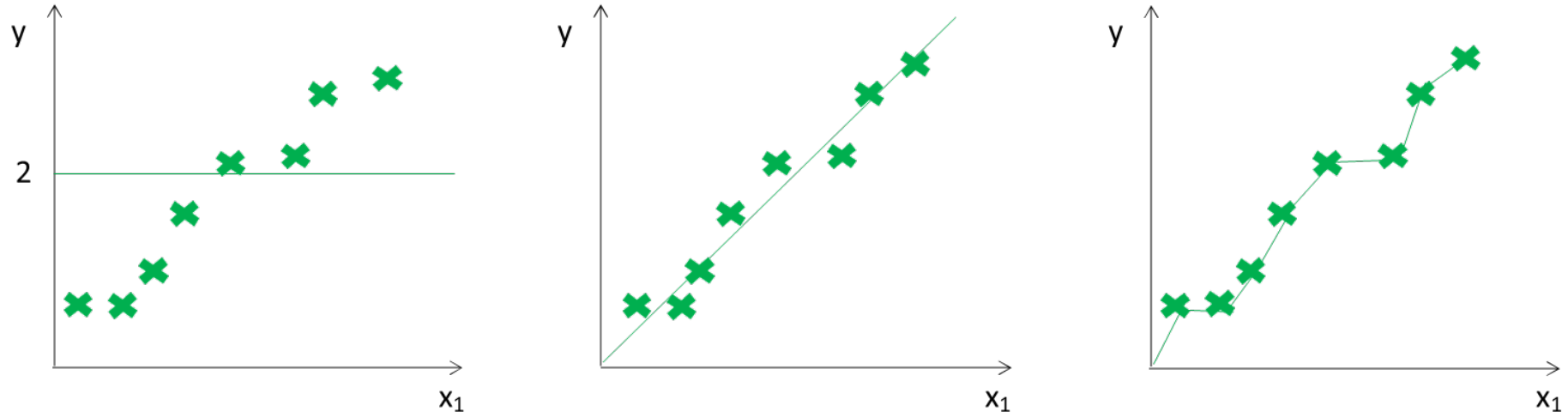
- Two stages in Machine Learning:
  - Training: The users train the model based on their needs by using different data and corresponding labels.
  - Inference: The trained ML models infer the outcome.
- Three outcomes for the training
  - Overfit: The production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit to additional data or predict future observations reliably.
  - Underfit: Underfitting is a scenario in data science where a data model is unable to capture the relationship between the input and output variables.
  - Wellfit: Your model can capture the data relationship well.

Human learning vs. Machine learning

- Let us make the three outcomes understandable
  - Overfit: You think machine learning = Pattern recognition, after taking our course
  - Underfit: After you taking the course, you know nothing about machine learning.
  - Wellfit: After you taking the course, you know all I teach today and you can also generalize what you learn today to new scenarios.

A small Q&A

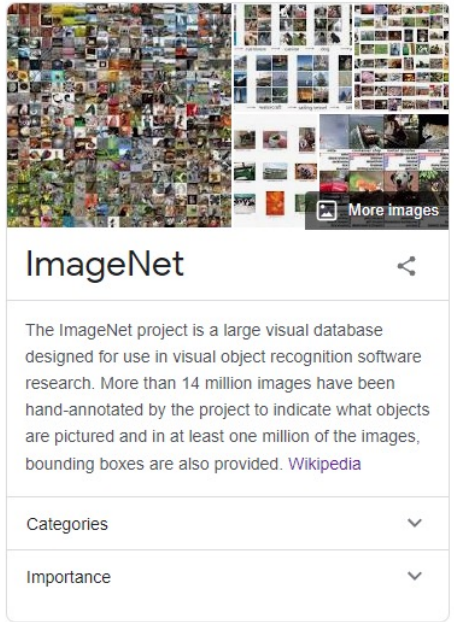
- Please point out the wellfit, underfit, and overfit.



- More data for a better model.

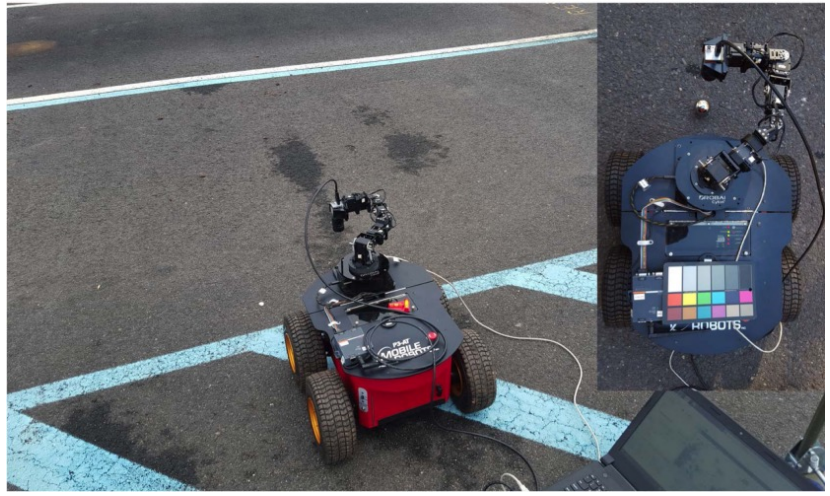
The data used for training

- Data is an important prerequisite of machine learning.
- Li Fei-Fei (李飞飞) launched a laundry store to support her data collection work.



The data used for training

- Assume that you want to train a model to recognize the materials. What you need to prepare for?



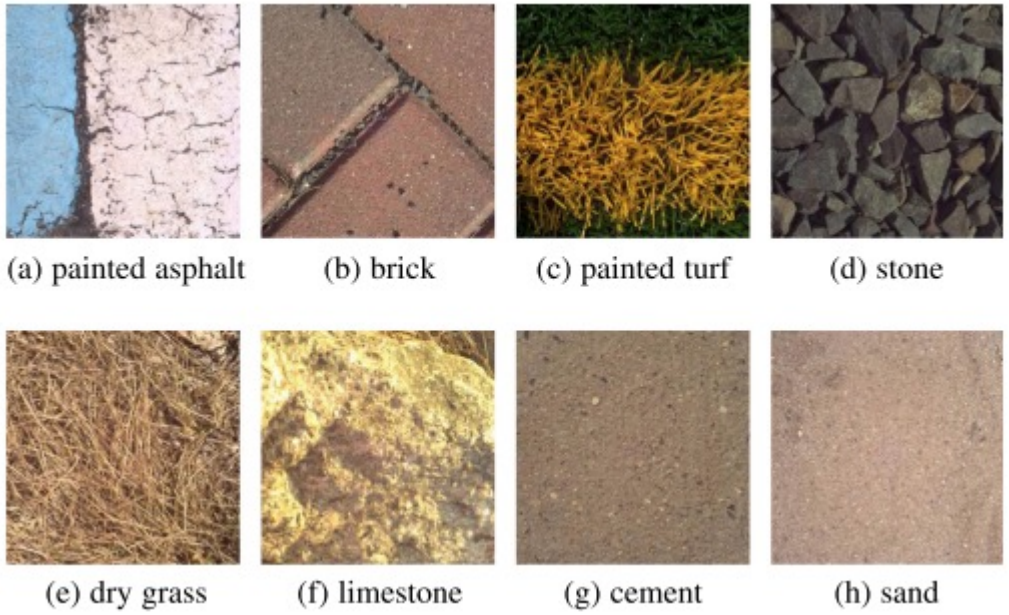
The robot in this figure needs to recognize the terrain material to do correct judgement.

How to build a corresponding dataset to train this ML model?

The data used for training

$\{x^i\}_{i=1....n}$

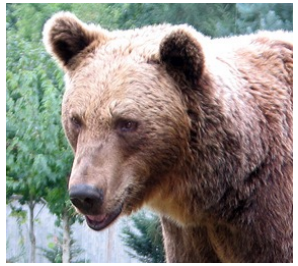
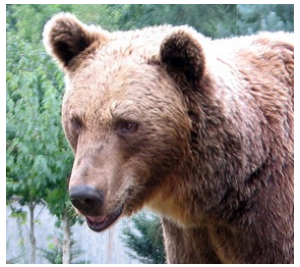
1. Capture training images from the real world.
2. Build a label manually for each image.



$\{x^i, y^i\}_{i=1....n}$

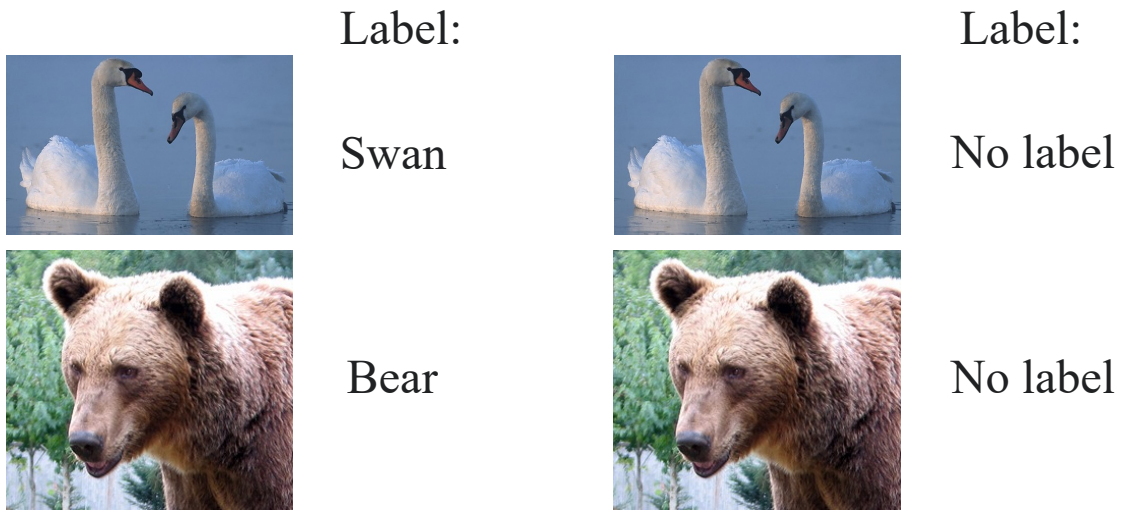
Different machine learning

- Supervised learning
  - Supervised learning (SL) is a machine learning paradigm for problems where the available data consists of labelled examples. (Label exists)
- Unsupervised learning
  - Unsupervised learning is a type of algorithm that learns patterns from unlabeled data.

	Label:		Label:
	Swan		No label
	Bear		No label

Different machine learning

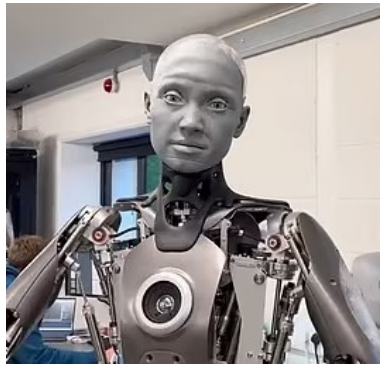
- Supervised learning
  - Supervised learning (SL) is a machine learning paradigm for problems where the available data consists of labelled examples.
- Unsupervised learning
  - Unsupervised learning is a type of algorithm that learns patterns from untagged data.



Different machine learning

- Supervised learning
  - Supervised learning (SL) is a machine learning paradigm for problems where the available data consists of labelled examples.
- Unsupervised learning
  - Unsupervised learning is a type of algorithm that learns patterns from unlabeled data.
- Reinforcement learning
  - learn to act based on feedback/reward

AI for gaming



Robots

Difference between Supervised Learning and Reinforcement learning?

- Use the example of training a robot to navigate through a maze to tell their difference

1.Supervised Learning:

1. In supervised learning, the robot is provided with a **labeled dataset** that includes **input-output pairs**. Each input corresponds to a specific state or configuration in the maze, and the output is the correct action to take at that state (e.g., move forward, turn left, turn right).
2. A human or an external system provides these labels by **manually guiding** the robot through the maze and recording the appropriate actions at each step.
3. The algorithm learns to map inputs (maze configurations) to outputs (actions) based on the provided labeled examples. Once trained, the robot can generalize its learning to new, unseen configurations in the maze.

2.Reinforcement Learning:

1. In reinforcement learning, the robot learns to navigate the maze **through trial and error without explicit guidance**. It receives feedback in the form of rewards or penalties based on the actions it takes in different states.
2. The robot explores the maze, **takes actions, and receives feedback** in the form of a reward or punishment based on the outcomes of its actions. The goal is for the robot to learn a policy (a mapping from states to actions) that maximizes the cumulative reward over time.
3. The learning process involves the robot making decisions, receiving feedback, and adjusting its strategy over multiple iterations until it discovers an optimal policy for navigating the maze.

In summary, supervised learning involves learning from a labeled dataset with explicit input-output pairs provided by a supervisor, while reinforcement learning involves learning through exploration and receiving feedback in the form of rewards or penalties, allowing the system to discover optimal strategies on its own. In the context of the maze example, supervised learning requires a human guide providing explicit instructions, while reinforcement learning allows the robot to learn by interacting with the environment and receiving feedback based on its actions.

Still cannot understand?

- Supervised learning: In my class, I give you clear definitions about “machine learning” and help you build the mapping between the definitions and machine learning.
- Unsupervised learning: No clear definitions are given. I directly take you to lab session and you need to learn from the lab yourself.
- Reinforcement learning: I help you deepen the understanding by answering your questions.
- Weakly supervised learning: Lectures to build mapping and lab to learn yourself. That is what we want!!!

15-min Break

Let us summarize Machine learning 1-2-3

- Collect data, preprocess data and extract features
- Build model: choose hypothesis class $\mathcal{H}$ and loss function l
- Optimization: minimize the empirical loss

Data for training

- Continuous features
  - Numerical values that can take on any value within a certain range.
  - Represent things such as time, weight, income, temperature, etc.
  - With more details representation of data than discrete or categorical features.
- Categorical features
  - Data that can be divided into categories, such as “male” and “female” or “red” and “blue.”
  - Help predict what category something belongs to, based on other features.
  - Categorical data can be thought of as a set of categories, and each category can be represented by a number.

But some data may not be very easy to differentiate their category. For example!

Data for training

- Categorical attribute convert

City: "Beijing" , "Shanghai", "Shenzhen". **One-hot-encoding**

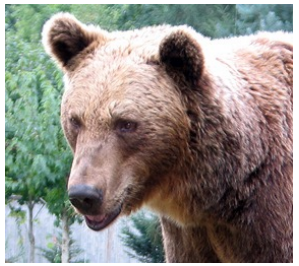
	f_Beijing	f_Shanghai	f_Shenzhen
Beijing	1	0	0
Shanghai	0	1	0
Shenzhen	0	0	1

Data for machine learning

- Some samples for (x_i, y_i)

Mixture image Background Reflection

Swan



Bear

Digit: 8

Data Pre-Processing

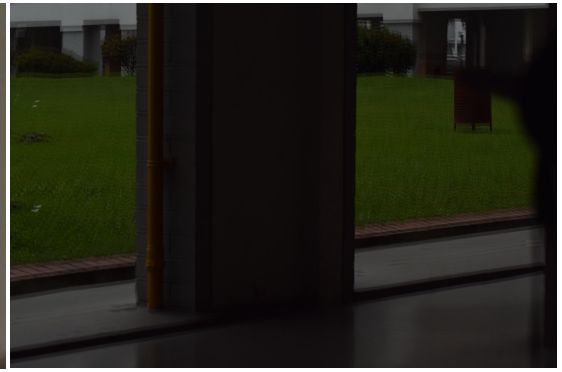
- What is outlier?
  - All things inconsistent with your purposes can be regarded as the outlier.

Student id	Name	Gender	Current GPA	CS Student	Course Enrolled
1034	Jerry	Male	3.2	Yes	Data Mining
1034	Jerry	Male	3.2	Yes	Physics
2578	Tom	Male	3.8	No	Data Mining
2578	Tom	Male	3.8	No	Linear Algebra
	Lion		2.9		
3729	Spike	Male	2.0	No	Linear Algebra
4280	Lily	Female	4.0	No	Physics
5530	Lucy	Female	3.7	Yes	Data Mining

Data Pre-Processing

- Here are some examples from the data I collected myself for research projects of Reflection Removal.
- This is an outlier! Can you tell me why this data is an outlier?

Mixture image



Reflection image

Background image

Da

- He ta I collected myself, when I was a P
- He y this data is an outlier?

Da

- He ta I collected myself, when I was a P
- He y this data is an outlier?

Data Pre-Processing

- For the examples shown in previous slides. Please discard them and collect a new one.
- In other scenarios, we have a plenty of methods to pre-process them.
  - Normalize
  - ToTensor
  - CenterCrop
  - ColorJitter
  - Grayscale
  - Pad
  - Resize

Data Pre-Processing

- Let us introduce them one-by-one
- Normalize: Normalize a tensor image with mean and standard deviation
- Why is such process important?

Label

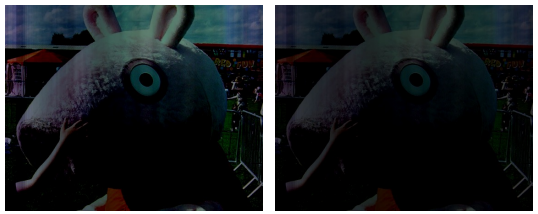
Peppa ☑

Label

Peppa ?

PSNR = 9.8

- Different illumination makes the accurate machine learning difficult



PSNR = 19.8

Two images after normalization

PSNR is an error metric used to measure the difference between two images. A higher PSNR value indicates a smaller difference between the two images.

Data pre-processing

- Can you find any problems in the data below?

	Data 1	Data 2	Data 3
1	0.24	10241	123
2	0.72	9867	432
3	0.88	2365	654
4	0.93	12117	345
5	0.26	7856	178

Too diverse

Data Pre-Processing

- Techniques in Normalization
  - Z-score normalization
  - Min-max normalization
  - Log Transformation
  - Square root normalization

Data pre-processing

- Z-score normalization

Data x_j → Mean: μ_j standard deviation : σ_j → $\dfrac{x_j - \mu_j}{\sigma_j}$

	Data 1	Data 2	Data 3
1	0.24	10241	123
2	0.72	9867	432
3	0.88	2365	654
4	0.93	12117	345
5	0.26	7856	178
Mean	0.61	8489.2	346.4
STD	0.30	3347.7	189.8

Data pre-processing

- Z-score normalization

Data x_j → Mean: μ_j standard deviation : σ_j → $\frac{x_j - \mu_j}{\sigma_j}$

	Da 1	Feature 2	Feature 3
1	-1.22	0.52	-1.17
2	0.38	0.41	0.45
3	0.91	-1.83	1.62
4	1.08	1.08	-0.01
5	-1.15	-0.19	0.88

Data pre-processing

- Min-max values
  - x_{max} is the maximum value of the data x
  - x_{min} is the minimum value of the data x

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Data pre-processing

- Min-max values

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

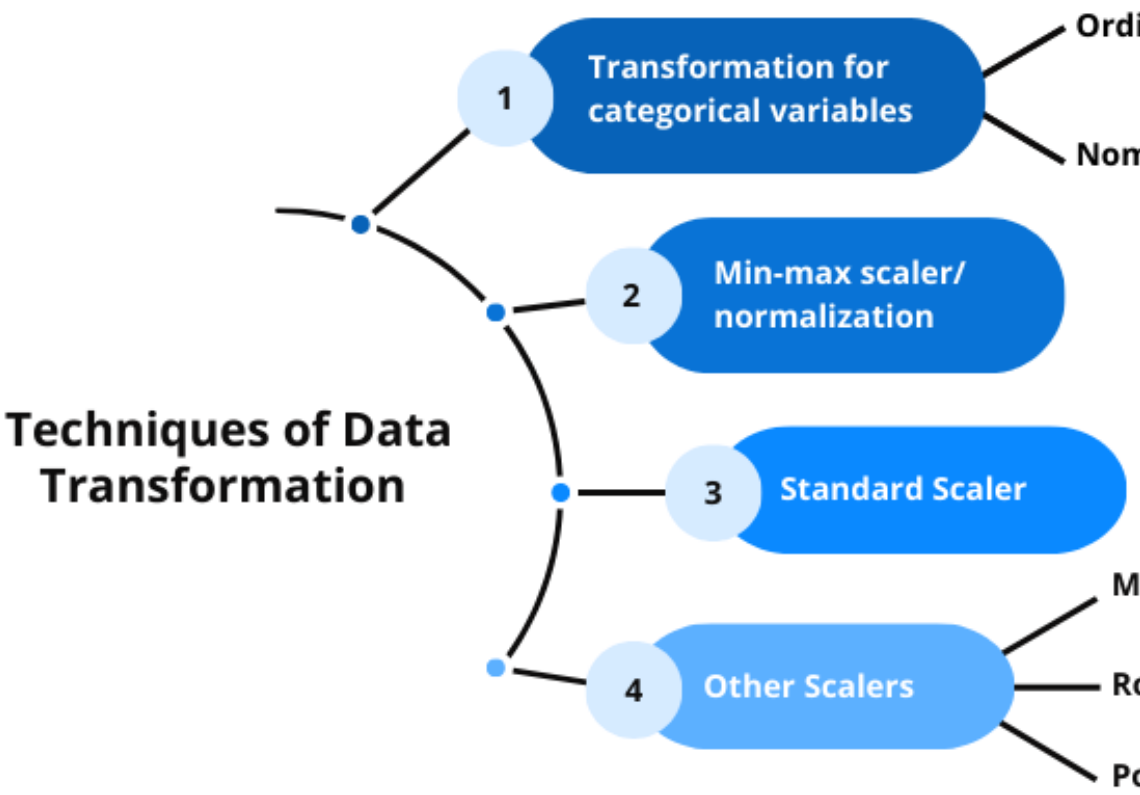
	Data 1	Data 2	Data 3
1	0.24	10241	123
2	0.72	9867	432
3	0.88	2365	654
4	0.93	12117	345
5	0.26	7856	178
Max	0.93	12117	654
Min	0.24	2365	123

Data pre-processing

- Min-max values

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

	Data 1	Data 2	Data 3
1	0.0	0.81	0.0
2	0.70	0.77	0.58
3	0.93	0.0	1.0
4	1.0	1.0	0.42
5	0.03	0.56	0.11



Ordinal Variables

{“small”: 1, “medium”: 2, “large”: 3}

Nominal Variables

One Hot Encoding, [0,0,1]

•The maxAbs scaler: This technique takes the absolute maximum value of the feature and divides each record by this max value, scaling the data in the range of -1 and 1.

MaxAbs scaler

Robust scaler

This technique removes the median from the data and scales it using the interquartile range (IQR). As the name suggests, it’s robust to outliers.

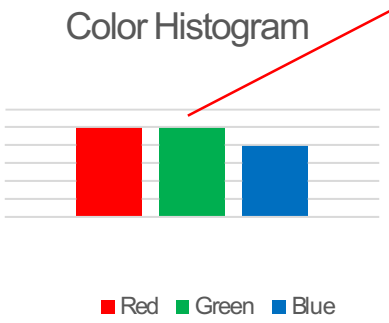
Power transformer scaler

This technique changes the data distribution, making it more like a normal distribution. It’s used most with heteroscedasticity data, which means that all variables don’t have the same variance.

Example 1: image classification

Hand-craft features

Extract features



Feature vector: x_i

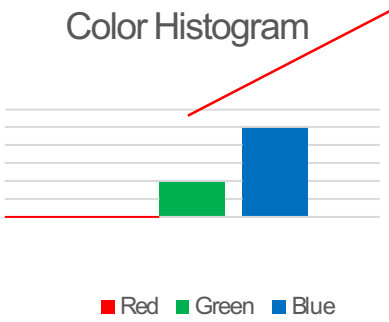
Indoor

0

Label: y_i

Hand-craft features

Extract features



Feature vector: x_j

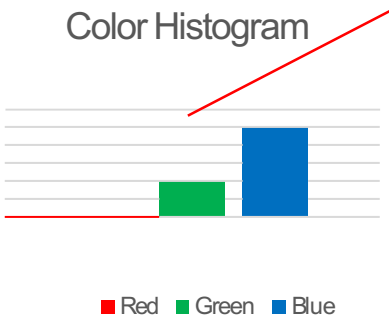
outdoor

1

Label: y_j

Hand-craft features

Extract features



Feature vector: x_j

indoor

1

Label: y_j

What if we face this kind of interior design?

Sheepdog or mop?

Chihuahua or muffin?

Data Representation

Many **AI** tasks can be solved by designing "**right set of features to extract**" for the task to be provided to a simple machine learning algorithm.

Example: "*Write a 'program' to detect cars in image*"

Cars have *wheels* ... use "presence of a *wheels* " as a feature? BUT difficult to describe what a *wheel* looks like in terms of **PIXEL** values. *Wheel* has a simple geometric shape BUT its image may be complicated by shadows, sun glaring off metal parts etc.

SOLUTION: use **ML** to **DISCOVER** not only the **MAPPING FROM REPRESENTATION** to output but also the representation itself. .. this approach is known as **REPRESENTATION LEARNING**.

Learned representations often result in better performance than obtainable with "**hand-designed**" representations.. also enable **AI** systems to rapidly adapt to new tasks, with minimal human intervention.

a "**representation learning algorithm**" can discover a good set of features for a simple task in minutes or for a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time & effort.

Data Representation

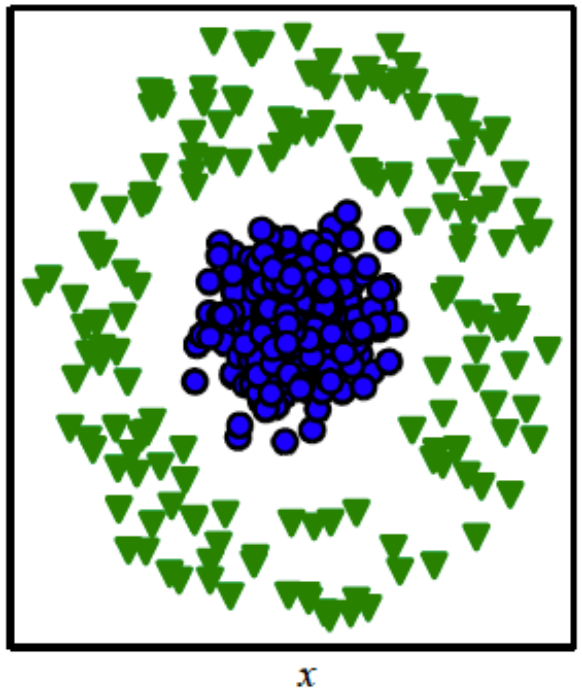
Machine learning algorithms depend heavily on the representation **of the data** they are given:

- **Deep Learning** concerned with **learning** how to represent data in a **Machine Learning** (**ML**) system

Example:

- scatter plot of two categories of data in 2D space .. y
- problem made easy by changing the way data is represented!

Cartesian Coordinates



Data Representation

Machine learning algorithms depend heavily on the representation **of the data** they are given:

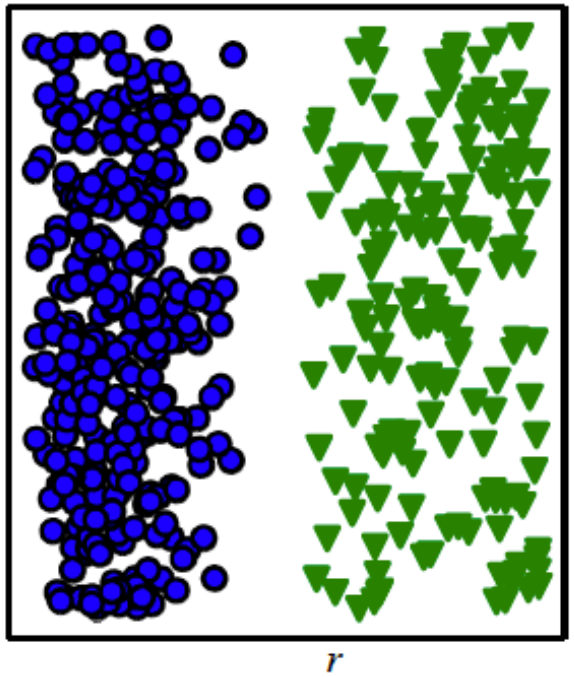
- Deep Learning concerned with **learning** how to represent data in a **Machine Learning** (**ML**) system

Example:

- scatter plot of two categories of data in 2D space ..
- problem made easy by changing the way data is represented!

ML has been applied successfully to challenging problems .. through changing representation of data...

Polar Coordinates



Glaucoma Screening from Fundus Image

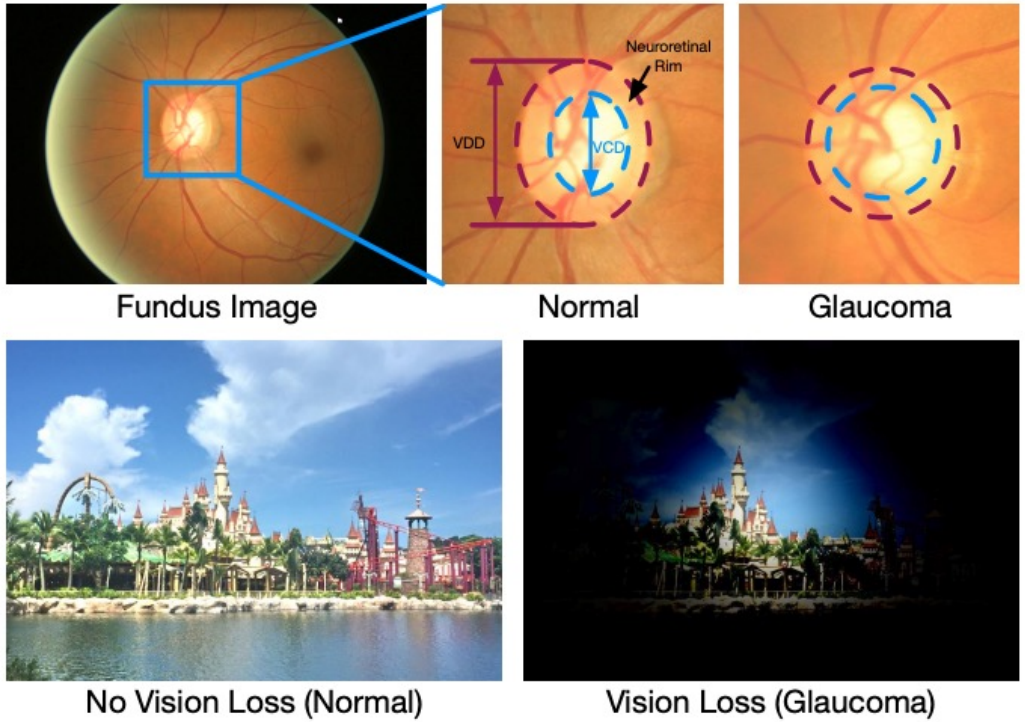


Fig. 1. Top: the whole fundus image and zoom-in normal/glaucoma disc regions, where the vertical cup to disc ratio (CDR) is calculated by the ratio of vertical cup diameter (VCD) to vertical disc diameter (VDD). Bottom: the visual fields with normal and glaucoma cases.

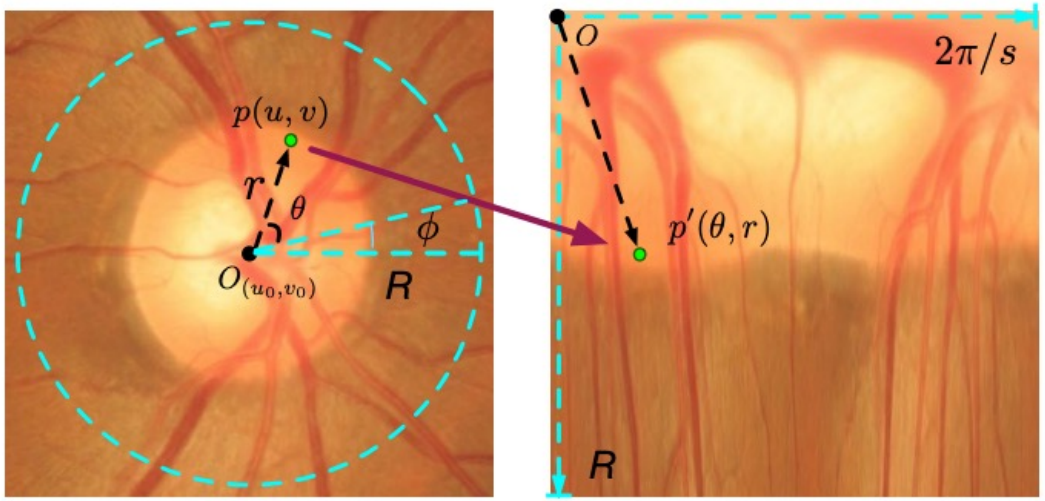


Fig. 4. Illustration of the mapping from Cartesian coordinate system (Left) to the polar coordinate system (Right) by using the polar transformation. The point $p(u,v)$ in Cartesian coordinate corresponds to the point $p'(\theta,r)$ in polar coordinate.

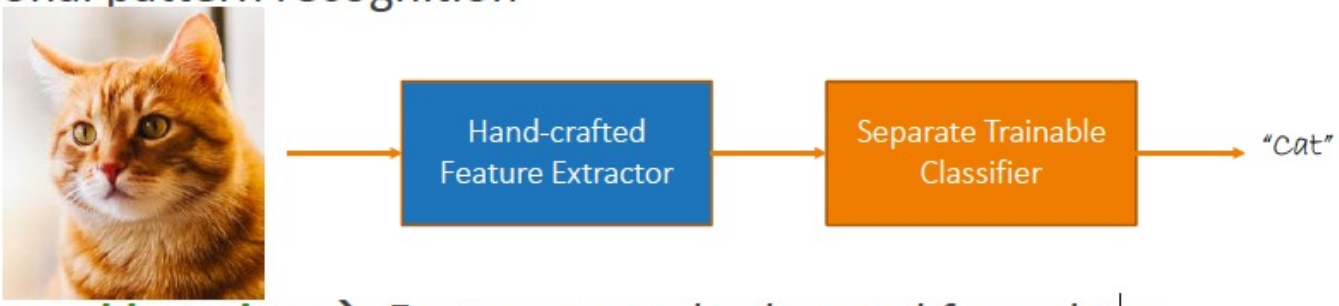
Wait…

- Why handcraft the feature vectors x, y?
  - Can use prior knowledge to design suitable features

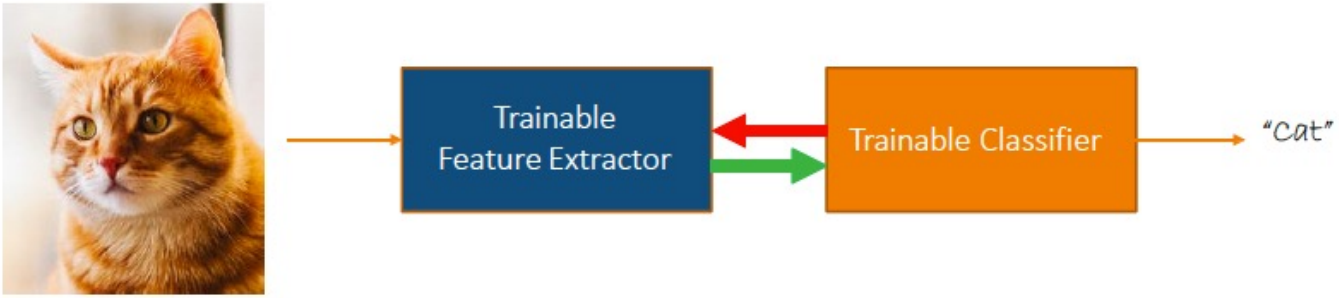
- Can computer learn the features on the raw images?
  - Learn features directly on the raw images: Representation Learning
  - Deep Learning $\subseteq$ Representation Learning $\subseteq$ Machine Learning $\subseteq$ Artificial Intelligence

Learning Representations & Features

Traditional pattern recognition



End-to-end learning → *Features are also learned from data*



Learning .. Features

Manually designed (*hand-crafted*) **features:**

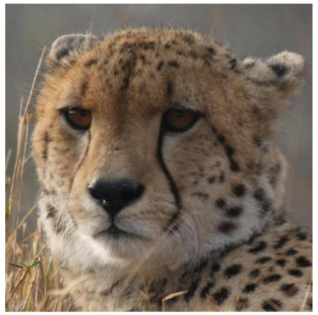
usually take much time to come up with, implement & validate
often incomplete; not easily known if they are optimal for the task

Learned features

easy to adapt
compact and specific to the task at hand
relatively easy and fast to optimize

Over-confidence Issue

Training data



Prediction: dog
Probability: 0.98

Prediction: dog
Probability: 0.95

The model used was Resnet-18, which yields ~99% accuracy on the validation set. Only using this for evaluation would have us believe it's an amazing model, but that's not why we're here. Below is the image of blogger and a dog, where apparently the blogger is more dog than this actual dog. Even worse, it's 98% confident that he is a dog, so he will be a dog even if we were only considering predictions with over 95% confidence.

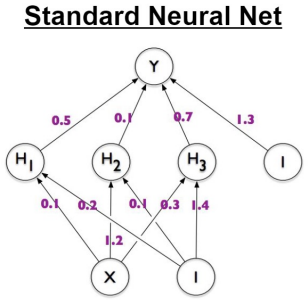
So how to solve this issue?

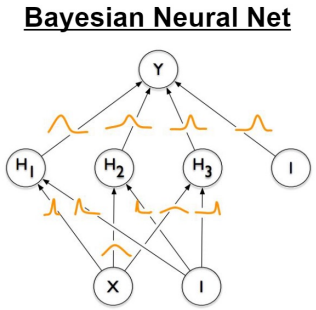
https://jramkiss.github.io/2020/07/29/overconfident-nn/

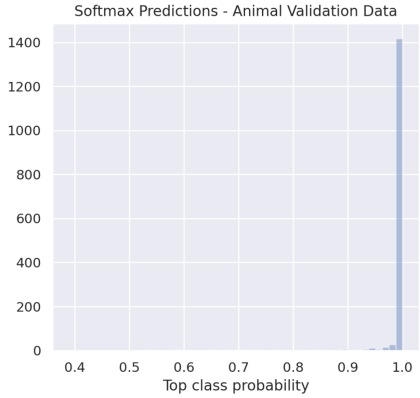
Possible Solutions

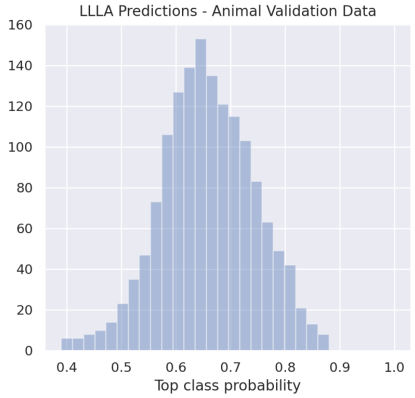
1. Collect more data on the human and build up human classifier.

2. Bayesian methods are great for quantifying uncertainty.





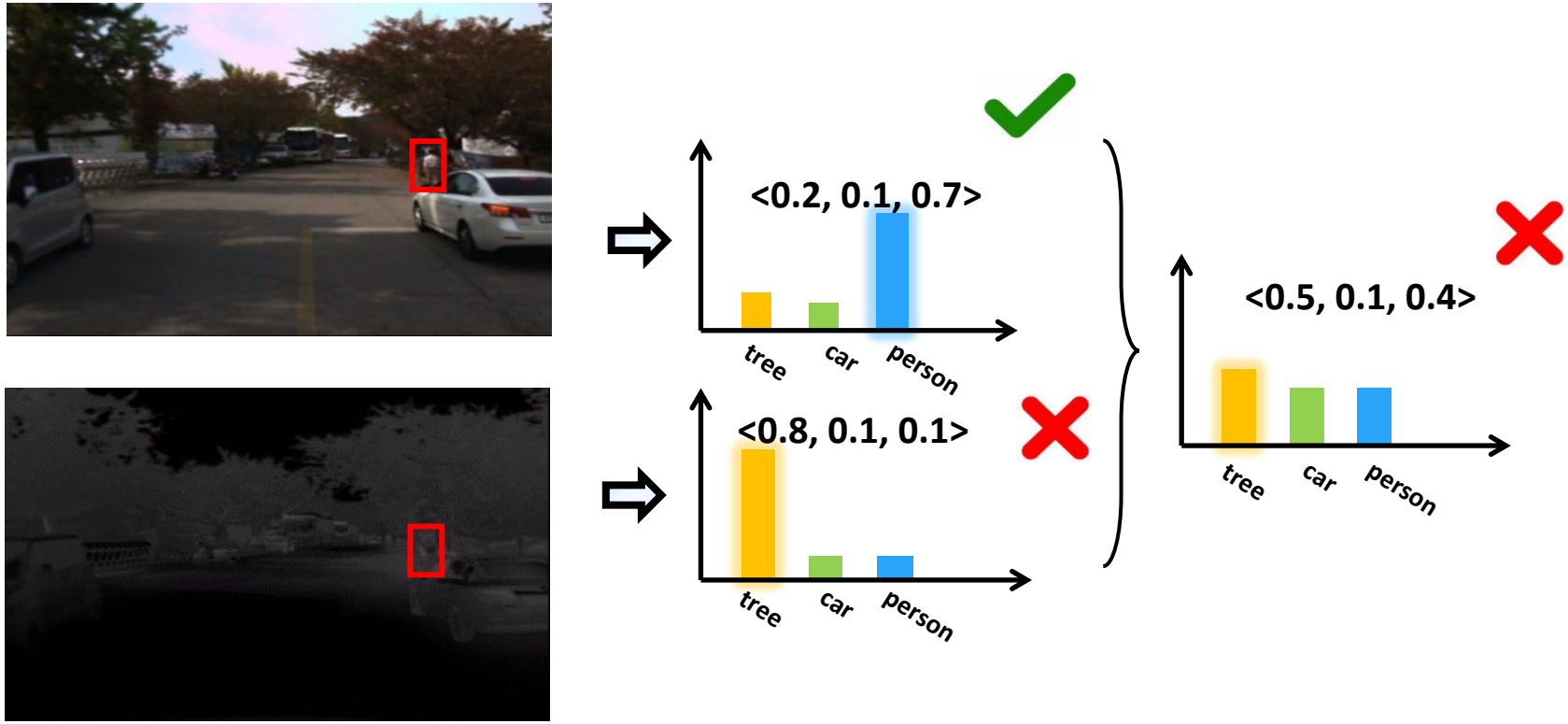




Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Network, ICML 2020.

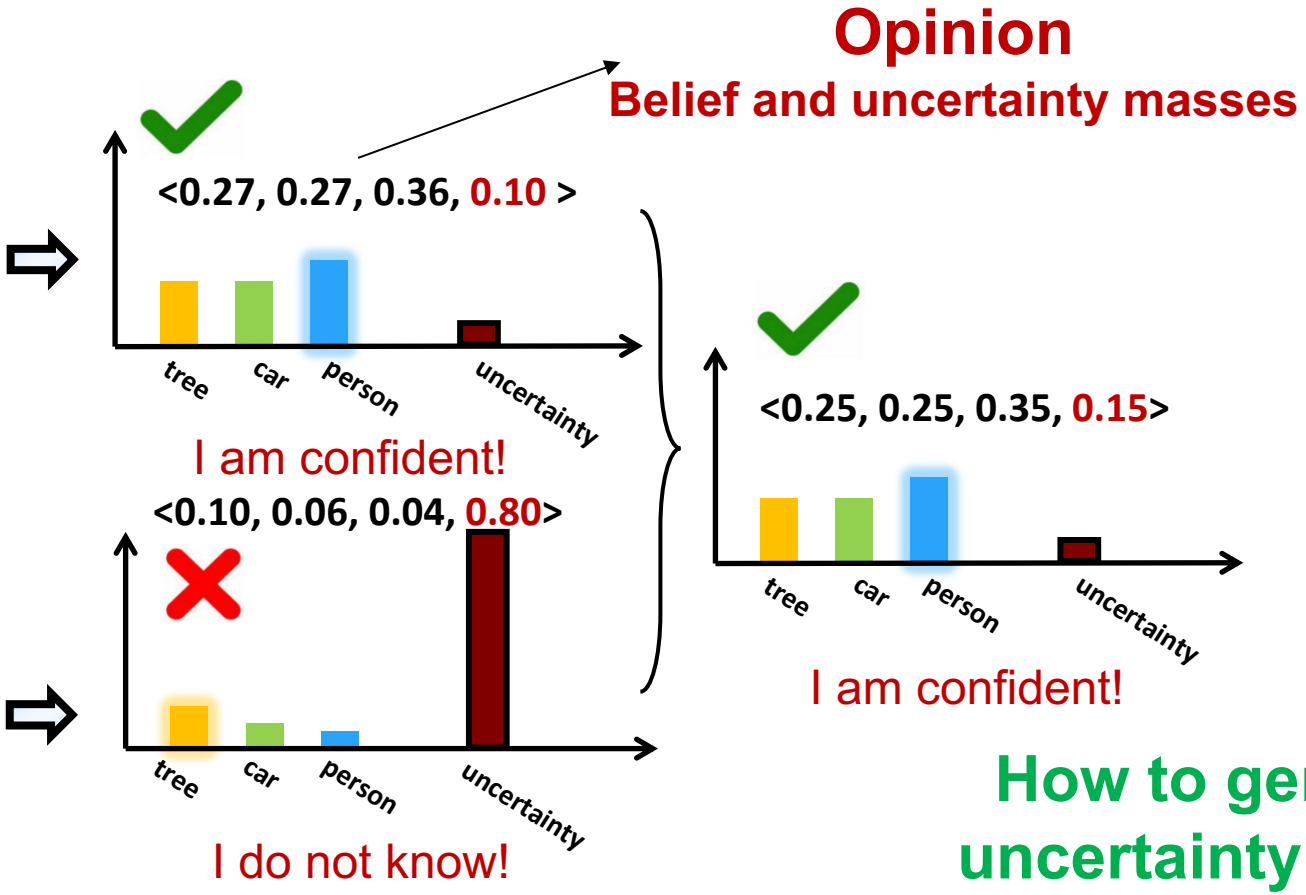
Issue of Traditional Multi-Modal Classification

- Traditional fusion methods are not trustworthy
  - The *Softmax* output is usually lead to over confidence.
  - Uncertainties of different modalities are not considered in the fusion.



Method: Trustworthy Multi-Modal Classification

Predictive uncertainty is directly modeled.



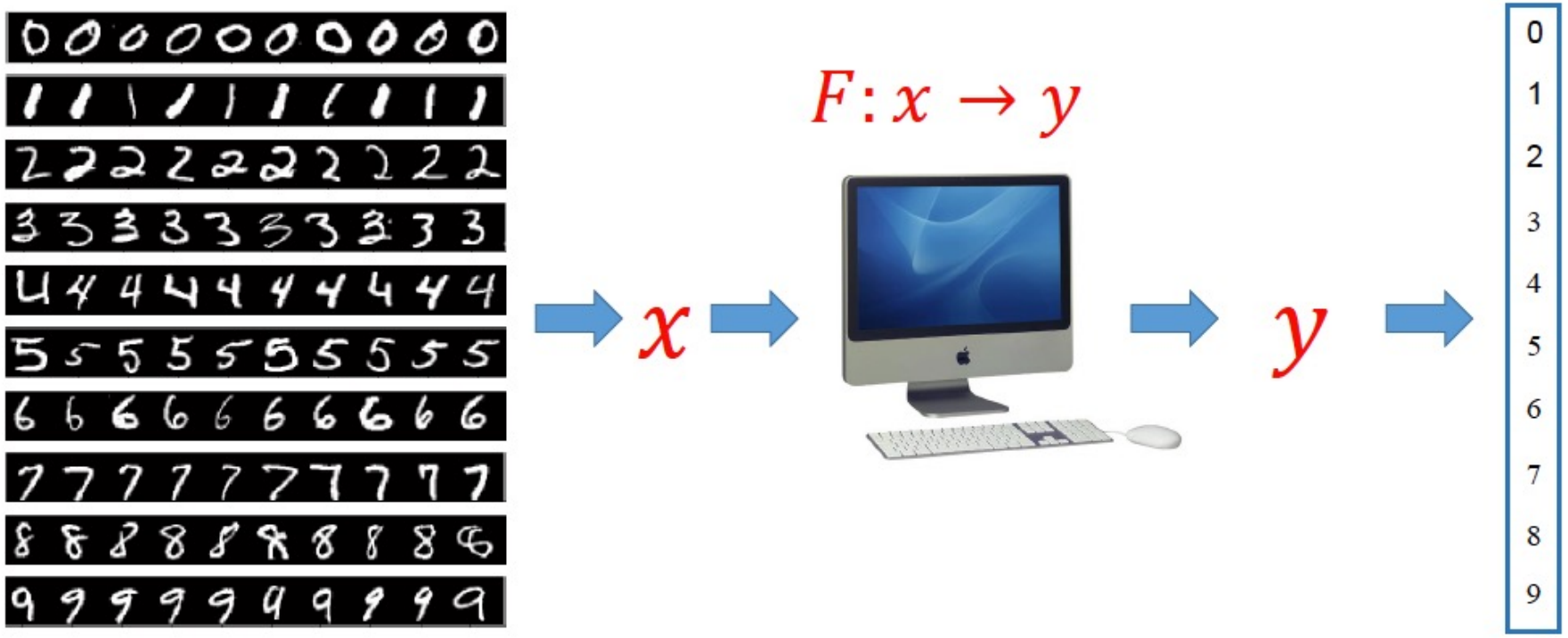
How to generate uncertainty score?

Z. Han, C. Zhang, H. Fu and J. Zhou, "Trusted Multi-View Classification With Dynamic Evidential Fusion" in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 45, no. 02, pp. 2551-2566, 2023.

Let us summarize Machine learning 1-2-3

- Collect data, preprocess data and extract features
- Build model: choose hypothesis class $\mathcal{H}$ and loss function l
- Optimization: minimize the empirical loss

Handwritten Digits Recognition



Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n \}$
- Find $y = f(x)$ using training data
- s.t. f correct on test data

What kind of functions?

Handwritten Digits Recognition

digitizing

$$F: x \rightarrow y$$

representation

0
1
2
3
4
5
6
7
8
9

Problem:
How to construct the mapping F?

Handwritten Digits Recognition

$$x = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ \vdots \end{bmatrix}$$

28x28

Problem:
How to construct the mapping F from 784 dimensional space to 10 dimensional space?
It is almost impossible manually!

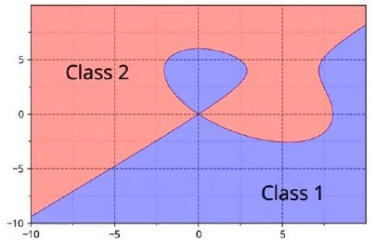
$$F : x \rightarrow y$$

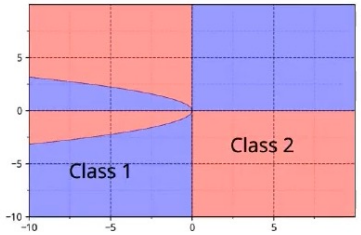
$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

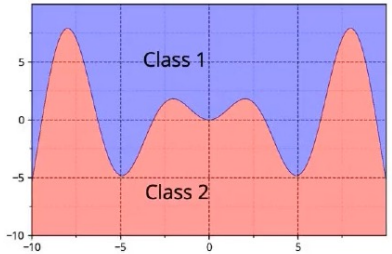
Universal approximation theorem

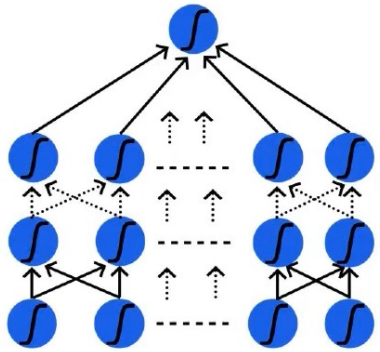
‘‘a single hidden layer neural network with a linear output unit can approximate any continuous function arbitrarily well, given enough hidden units’’

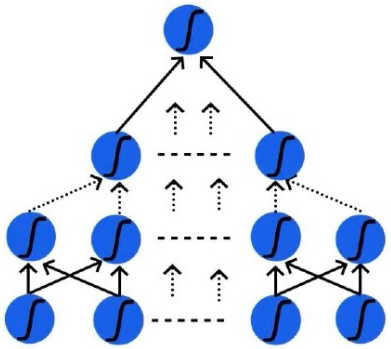
Hornik, 1991

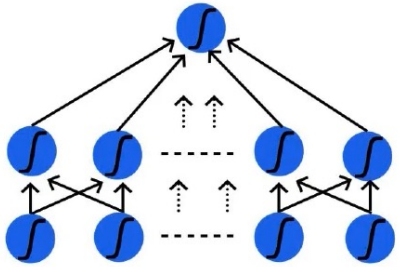




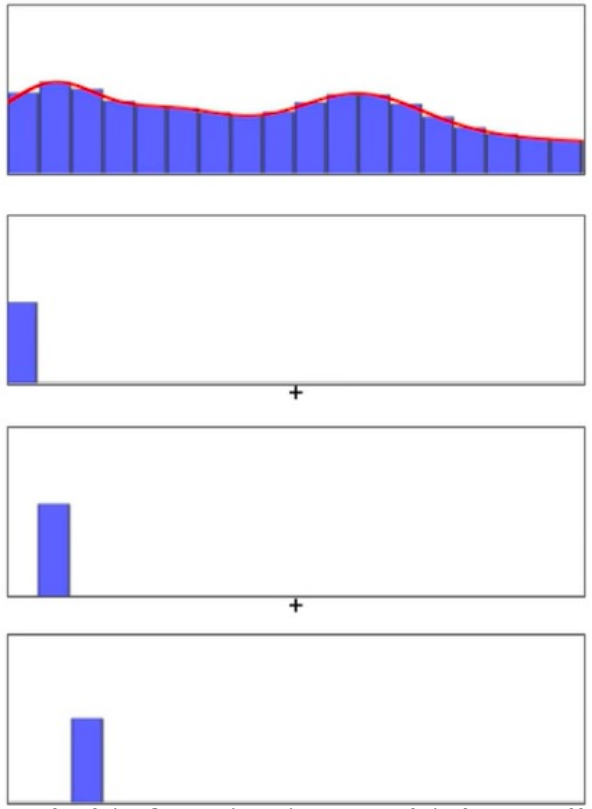


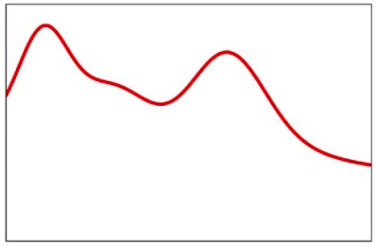


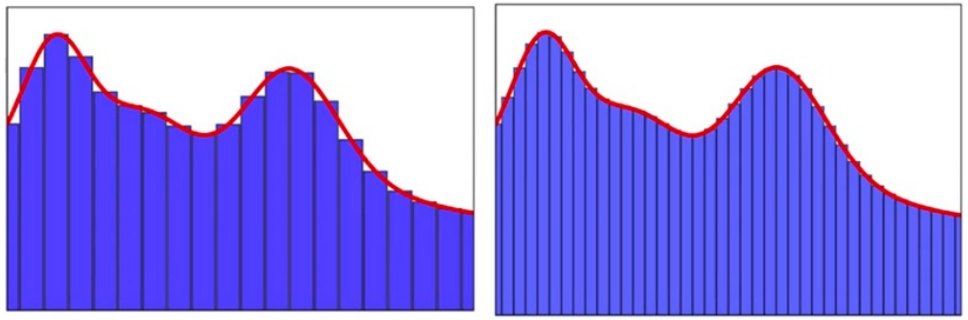




Illustrative Proof of Universal Approximation Theorem





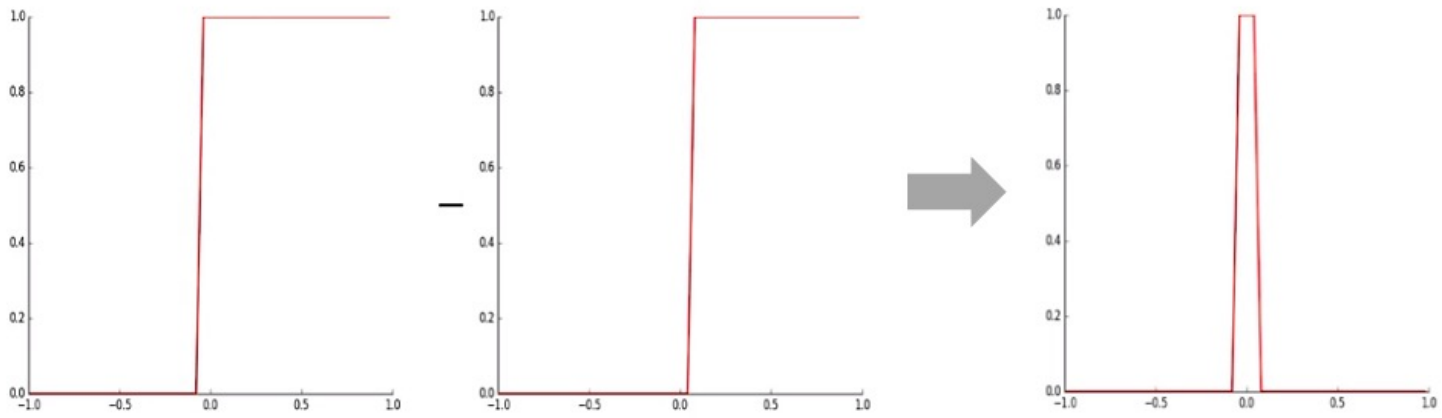


The more functions that I choose in this method the better will be my approximation.

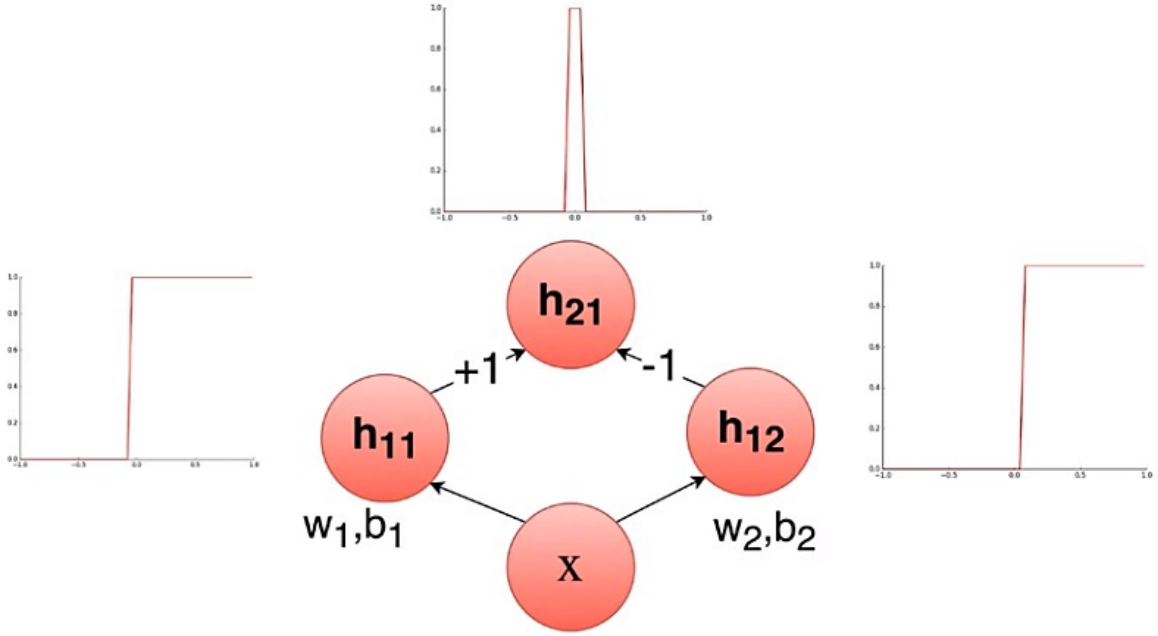
break this function into multiple smaller parts so that each part is represented by a simpler function.

How do we come up with these rectangles/towers and how it will tie back to the sigmoid neuron?

Let's take two sigmoid functions having a very steep slope and notice that they have a different place at which they peak. The left sigmoid peaks just before zero and right sigmoid peaks just after zero. If I just subtract these two functions the net effect is going to be a tower (rectangular output).



If we can get the series of these towers, then we can approximate any true function between input and output.



if we have an input **x** and it is passed through the two sigmoid neurons and the output from these two neurons are combined in another neuron with weights +1 and -1 i.e... same is as subtracting these two outputs, then we will get our tower. Now you can see that we have our building block ready which is a connection of three sigmoid neurons. If we can construct many such building blocks and add all of them up, we can approximate any complex true relationship between input and output. This is called the ***representational power of deep neural networks***, to approximate any kind of relationship between input and output.

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n \}$
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. f correct on test data

Connection between training data and test data?

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. f correct on test data i.i.d. from distribution D

They have the same distribution

i.i.d.: independently identically distributed

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. f correct on test data i.i.d. from distribution D

What kind of performance measure?

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Various loss functions

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$L(f) = \quad = \mathbb{E}_{x,y \sim D}[l(f, x, y)]$$

- Examples of loss functions:
  - 0-1 loss: $l(f, x, y) = \mathbb{I}[f(x) \neq y]$ and $L(f) = \Pr[f(x) \neq y]$
  - l_2 loss: $l(f, x, y) = [f(x) - y]^2$ and $L(f) = \mathbb{E}[f(x) - y]^2$

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ using training data
- s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

How to use?

Math formulation

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ that minimizes $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$
- s.t. the expected loss is small

$$L(f) = \mathbb{E}_{(x,y)\sim D}[l(f, x, y)]$$

Empirical loss

Linear regression

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$ i.i.d. from distribution D
- Find $f_w(x) = w^T x$ that minimizes $\hat{L}(f_w) = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - y_i)^2$

l_2 loss; also called mean square error

Hypothesis class $\mathcal{H}$

All of today’s class