Day 1: Introduction to AI

Part 1: AI Overview

2021

Qualcomm

Part 1: AI Overview

Qualcomm

Content

◆ 1.1 AI Overview

- 1.1.1 AI: What? How? Where?

- 1.1.2 Qualcomm AI

- 1.1.3 AI vs. Machine learning vs. Deep learning

- 1.1.4 Different types of machine learning

- 1.1.5 Basic concepts in Machine learning & Deep learning

◆ 1.2 Quick Tour of Deep Learning

◆ 1.3 Foundation of Deep earning

1.1 AI Overview

◆ 1.1.1 AI: What? How? Where?

- Definition of AI

- Cloud AI, Embedded AI

- AI Application Scene

Definition of AI

Artificial intelligence (AI) is in contrast to the natural intelligence of creatures, especially humans.

Artificial intelligence is "the designing and building of intelligent agents that receive percepts from the environment and take actions that affect that environment." [1]

In 1956, John McCarthy, a young Assistant Professor of Mathematics at Dartmouth College, decided to organize a seminar to discuss and develop ideas about "thinking machines". He named 'Artificial Intelligence' for the new field. Dartmouth Workshop is the birthplace of Artificial intelligence.

> These attendees won Turing Award in the future:
>
> Dr. Marvin Minsky, 1969
> Mr. John McCarthy, 1971
> Allen Newell, 1975
> Herbert A. Simon, 1975

[1] Russell, Stuart J., and Peter Norvig. Artificial intelligence: a modern approach. 1994.

[2] The Dartmouth CollegeArtificial Intelligence Conference: The Next Fifty Years, AAAI 2006

Qualcomm

Cloud AI, Embedded AI

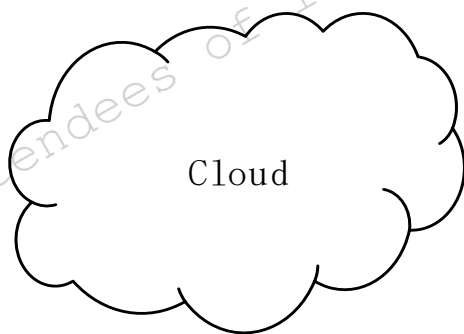At first, AI programs were developed and run on PC.

In recent years, the operating environment of AI has begun to shift from computers to cloud and embedded devices, not just PC.
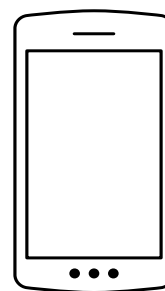
Let's compare their characteristics and differences.

Computer

Cloud

Device
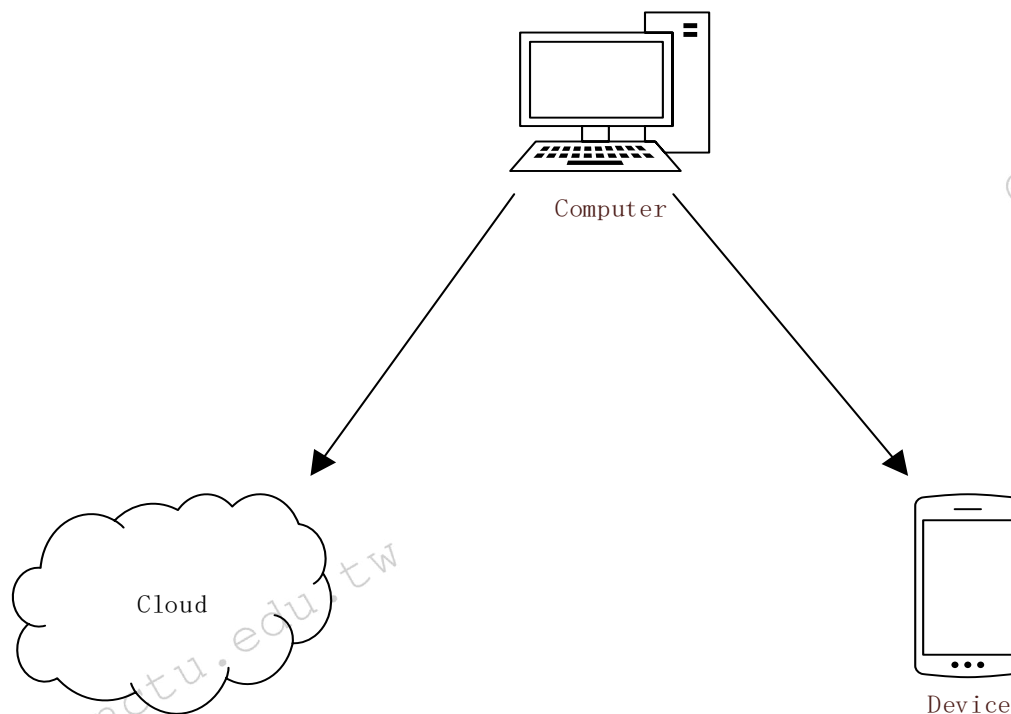
Cloud AI, Embedded AI

Cloud AI and Embedded AI have their own applicable scenarios.

Cloud

Device

- Powerful computing power

- Easy access to big data

- Strong scalability

- Suitable for sharing as a service

- User privacy and data security

- Work offline, quick response

- Small, lightweight and easy to carry

- Low power consumption

- Naturally suitable for IoT and Edge computing

Qualcomm

AI Application Scenes

Computer vision

- Image classification
- Object detection
- Fingerprint recognition and Face-ID
- Self-driving car

Automatic speech recognition

- Virtual assistant and smart speaker
- Simultaneous translation

Natural language processing

- Spam filtering
- Machine translation

Cross-cutting areas

- Smart City
- Unmanned store
- AlphaFold for "protein folding problem" (Biology science)
- Discovery and design new material (Materials science)

Game, Strategy and Decision

- AlphaGo (Google DeepMind)
- Deep Blue (IBM)

Business Intelligence

- Recommendation system
- Data mining

Qualcomm

1.1 AI Overview

◆ 1.1.2 Qualcomm AI

- Software

- Hardware

- Thundercomm AI Kit

https://www.qualcomm.com/solutions/mobile-computing/features/mobile-ai

Qualcomm AI - Software

Qualcomm is aiming to make on-device AI ubiquitous — expanding beyond mobile and powering other end devices, machines, vehicles, and things.

Machine Vision SDK supplies the cutting edge computer vision algorithms to provide localization, feature recognition and obstacle detection for Snapdragon platforms. It supports the following algorithms:

- Visual-Inertial Simultaneous Localization and Mapping (VISLAM)
- Depth From Stereo (DFS)
- Downward Facing Tracker (DFT)
- Camera Parameter Adjustment (CPA)
- …

Hexagon Neural Network (NN) library is an offloadable NN inference framework based on Hexagon DSP. Hexagon NN provides optimized deployment of basic machine learning modules and significantly accelerates deep learning operations such as

- Convolution
- Pooling
- Activation
- …

https://developer.qualcomm.com/software/machine-vision-sdk
https://developer.qualcomm.com/software/hexagon-dsp-sdk/dsp-processor
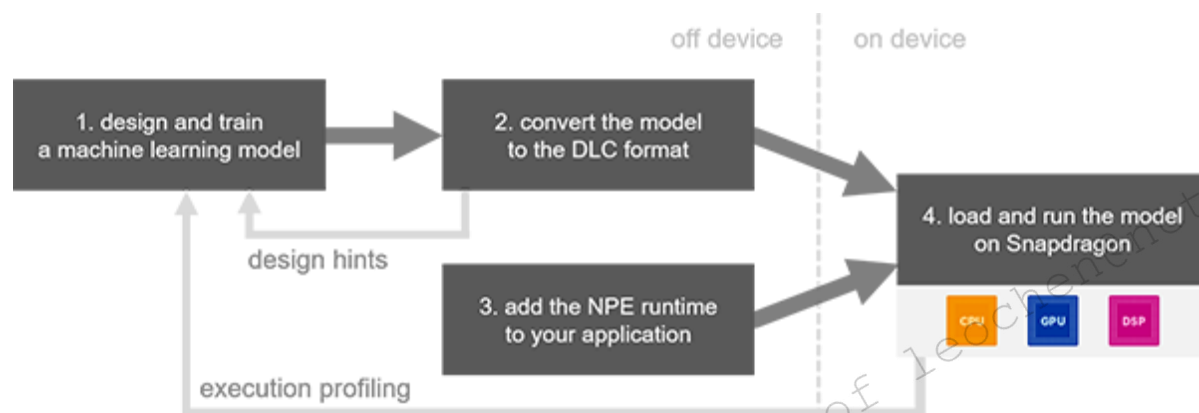
Qualcomm

Qualcomm AI - Software

Qualcomm is aiming to make on-device AI ubiquitous — expanding beyond mobile and powering other end devices, machines, vehicles, and things.

Snapdragon Neural Processing Engine (SNPE) is a software accelerated runtime for the execution of deep neural networks. With SNPE, users can:

- Execute an arbitrarily deep neural network
- Execute the network on the Snapdragon™ CPU, the Adreno™ GPU or the Hexagon™ DSP.
- Convert Caffe, Caffe2, ONNX™ and TensorFlow™ models
- Integrate a network into applications via C++ or Java

off device | on device

1. design and train a machine learning model
2. convert the model to the DLC format
4. load and run the model on Snapdragon

design hints

3. add the NPE runtime to your application

CPU GPU DSP

execution profiling

https://developer.qualcomm.com/docs/snpe/overview.html

Qualcomm

Qualcomm AI - Hardware

Snapdragon heterogeneous computing platform, equipped with Kryo CPU, Adreno GPU and Hexagon DSP, is aiming to run AI applications quickly and efficiently on-device.

This heterogeneous computing approach gives developers and OEMs the ability to optimize AI user experiences on smartphones and other edge devices.
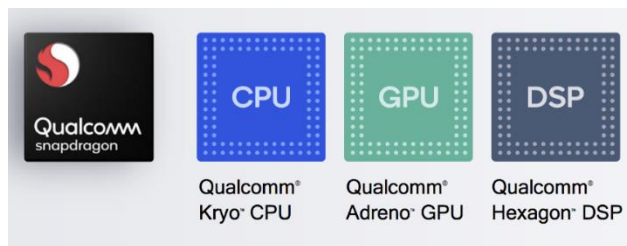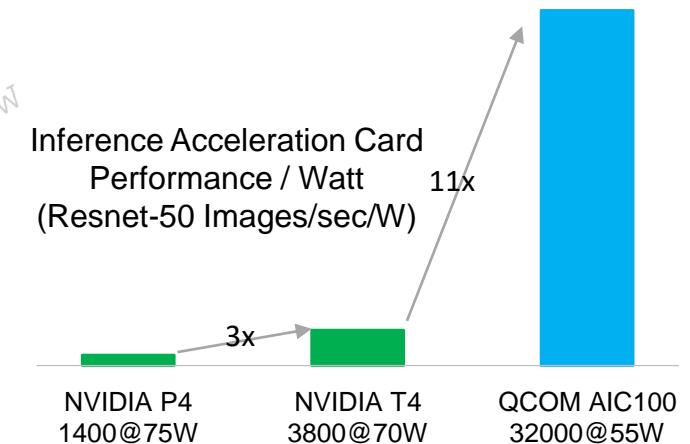
Neural processing unit (NPU) AI accelerator is specially designed for mobile neural network. It accelerates AI features including face recognition, facial landmark detection, object detection and so on.
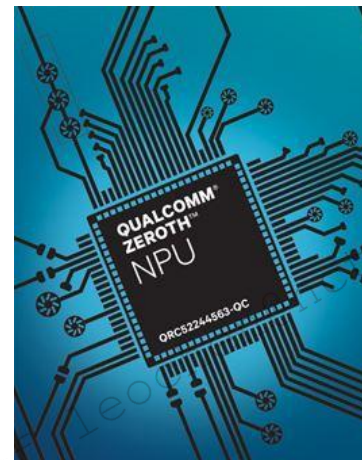
Cloud AI 100 is a kind of AI Inference Acceleration Cards for cloud servers.

With 7nm process node, Cloud AI 100 is more than 10x performance per watt over the industry's most advanced AI inference solutions.

Inference Acceleration Card Performance / Watt (Resnet-50 Images/sec/W)

11x

3x

NVIDIA P4 1400@75W

NVIDIA T4 3800@70W

QCOM AIC100 32000@55W

AI Kit

The TurboX™ AI Kit is a high-performance embedded development device based on the Qualcomm® SDA845 processors. The kit is designed to support on-device AI application development ranging from robotics, AR/VR, smart camera, automotive, smart retail, smart factory, smart home and smart city.

AI kit supports various AI algorithms utilizing the Qualcomm® AI Engine, running on a heterogeneous architecture featuring Qualcomm® Hexagon™ DSP, Qualcomm® Adreno™ GPU and Qualcomm® Kryo™ CPU

It has a rich interface and good scalability.

1.1 AI Overview

◆ 1.1.3 AI vs. Machine learning vs. Deep learning

Qualcomm
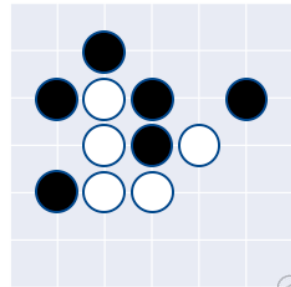
Gomoku AI Game v.s. AlphaGo

AlphaGo
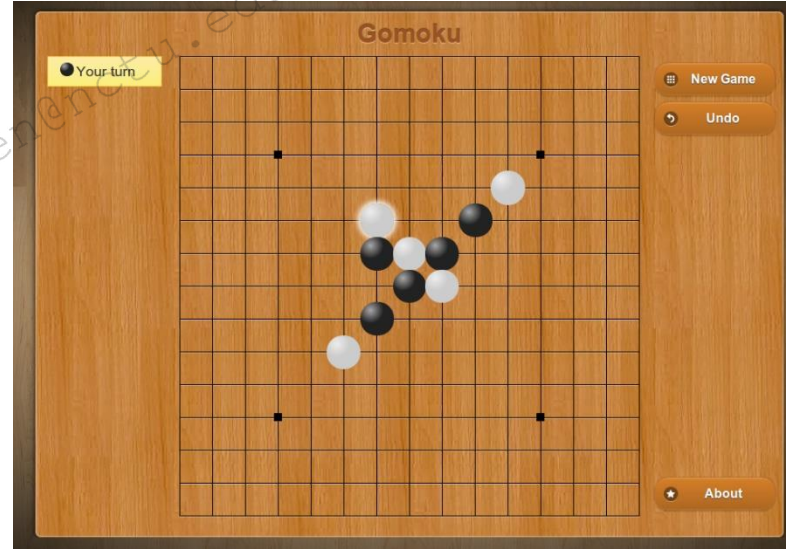
Gomoku Game (Five in a Row)

Question:

1. Are they AI program?
2. Do they use Machine learning techniques?
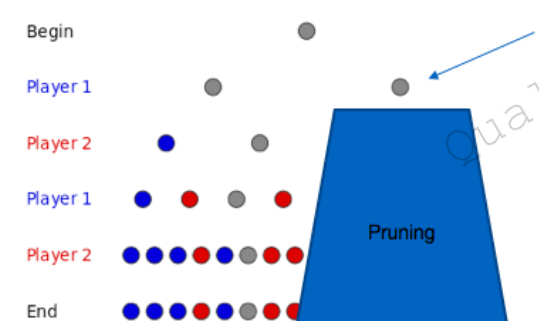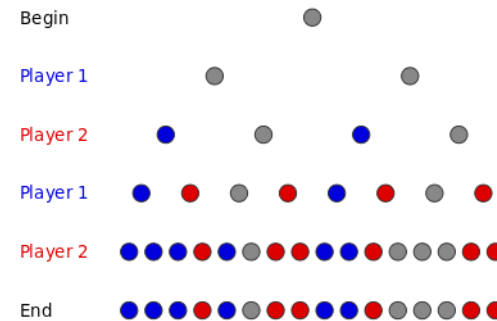3. Do they use Deep learning techniques?

https://en.wikipedia.org/wiki/File:FloorGoban.JPG
https://github.com/yyjhao/HTML5-Gomoku

Qualcomm

Brief introduction to Gomoku AI Game

A fragment of chess board

Scores for White

1. Define a score function.

2. Create a game tree.

3. Depth-First-Search (DFS) + Pruning.

Qualcomm

Brief introduction to AlphaGo

The rules of Go is quite complicated than Gomoku. So score and decision functions are implemented by CNN; search all nodes and pruning is impossible, so Monte Carlo Tree Search (MCTS) is performed.

Reinforcement learning + Convolutional neural network (CNN) + Monte Carlo Tree Search (MCTS)

1. **Score function:** Value Network estimates probability that current move leads to win, implemented by 12 layers CNN.

2. **Decision function: Policy** Network decides next chess piece, implemented by 12 layers CNN.

3. **Game tree search:** MCTS randomly search.

4. Train the reinforcement learning model on 30M positions from human expert games.

https://www.nature.com/articles/nature16961
https://icml.cc/2016/tutorials/AlphaGo-tutorial-slides.pdf
https://zhuanlan.zhihu.com/p/20607684

Qualcomm

AI vs. Machine learning vs. Deep Learning

Artificial Intelligence (AI): the ability that a machine to perceive, learn, think, resolve problems, and communicate like human beings. It includes a lots of fields like machine learning, natural language processing (listening and reading), computer vision (observing), robotics (moving) and so on. In a broad sense, all techniques to make a machine "smart" are all AI.

Machine Learning: A road leading to artificial intelligence. The algorithm "**learns**" the intrinsic rules by itself, rather than hardcoded rules by hand.

Deep Learning: A field of machine learning. Literally, the term "deep" means the model is combined by a stack of "shallow" models.

AI

Machine Learning

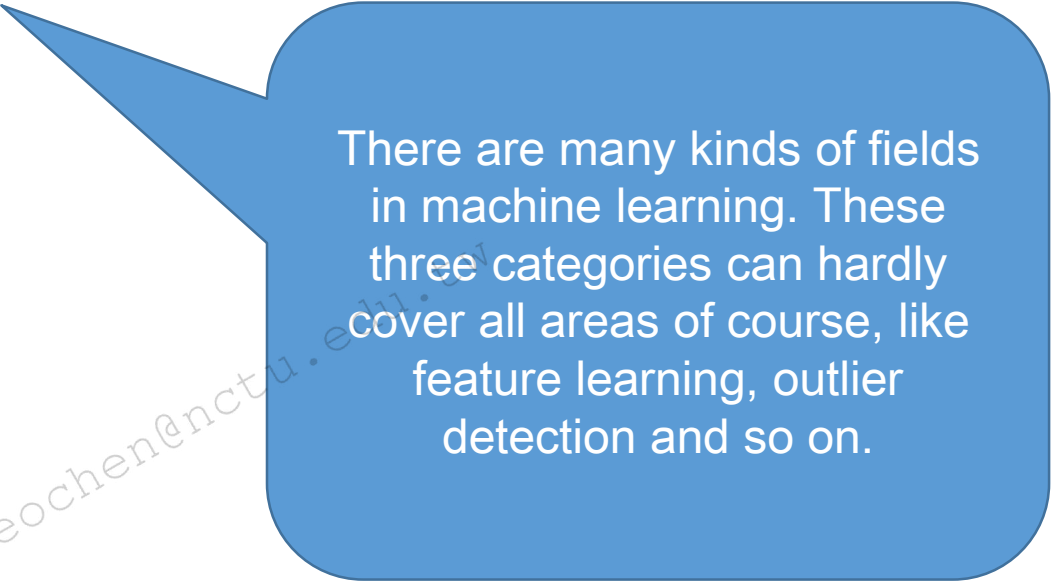Deep Learning

Qualcomm
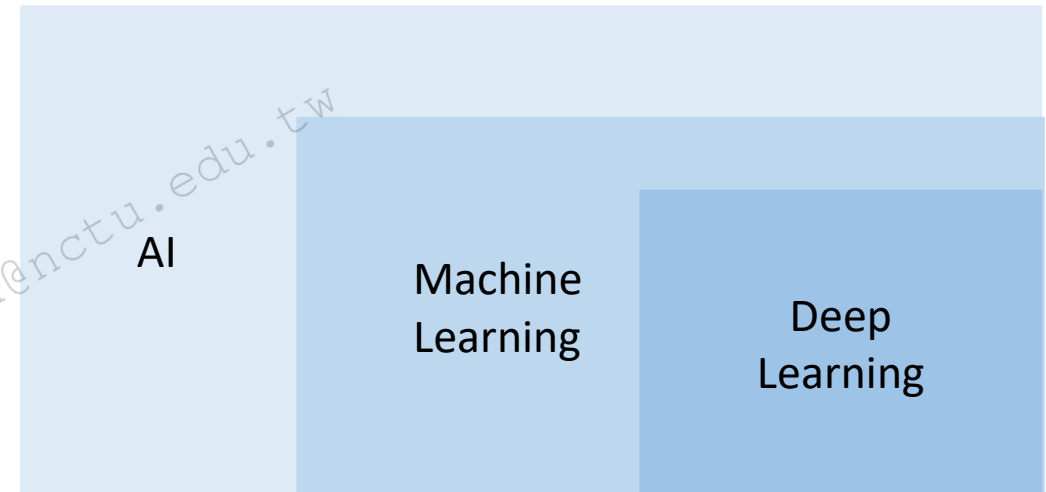
1.1 AI Overview

◆ 1.1.4 Different types of machine learning

- Supervised learning

- Unsupervised learning

- Reinforcement learning

Different types of machine learning

Machine learning can be **mainly** divided into three types according to the type of model's input and output.
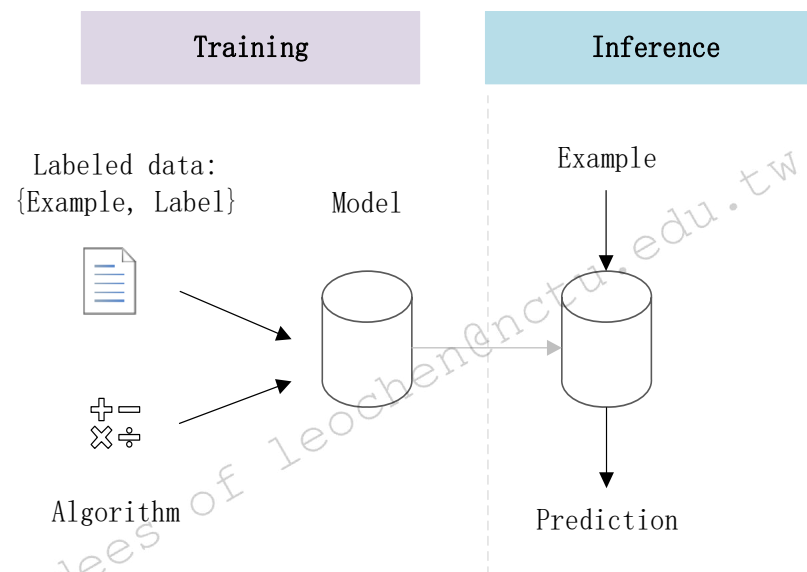
- Supervised learning

- Unsupervised Learning

- Reinforcement Learning

There are many kinds of fields in machine learning. These three categories can hardly cover all areas of course, like feature learning, outlier detection and so on.

Qualcomm

Supervised learning

- The input of supervised learning model are examples and desired output pairs, which is named as training data. The corresponding desired output is called as label or ground truth.

- During training, the algorithm learns a function (model) from training data. After training, the function is used to infer the labels of unseen examples in training data, as known as test data.

Training	Inference

Labeled data:
{Example, Label}

Model

Example

Algorithm

Prediction

Unsupervised learning

Unsupervised learning involves to learn intrinsic but unknown pattern in input data without labels. Unsupervised learning algorithms identify inherent commonalities in the data itself and react based on the presence or absence of such commonalities in each piece of data.
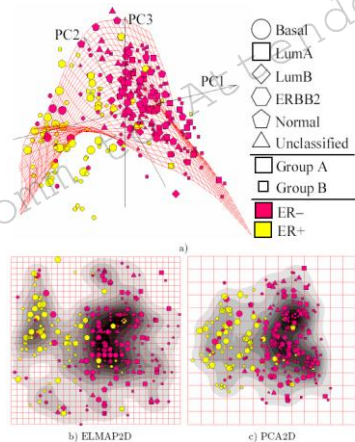
A common case of unsupervised learning is dimensionality reduction.
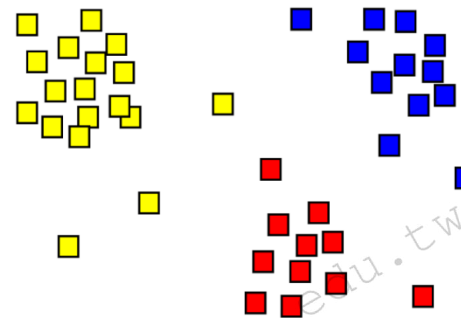
Another typical case is clustering. Cluster analysis is dividing a set of similar samples into several groups, , according to inherent commonalities of the data itself.
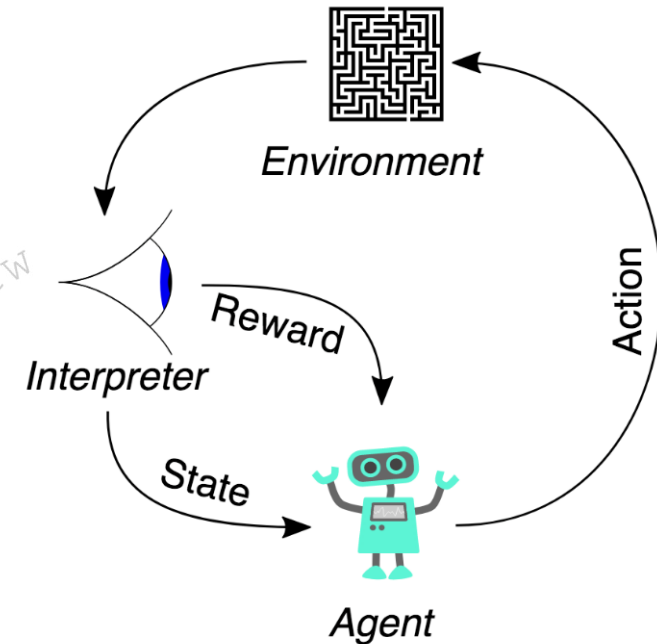
https://en.wikipedia.org/wiki/File:Elmap_breastcancer_wiki.png
https://en.wikipedia.org/wiki/File:Cluster-2.svg

Reinforcement Learning

- Reinforcement learning defines interactive learning based on agents and environments.

- The typical framing of a Reinforcement Learning (RL) scenario: an **agent** takes **actions** in an **environment**, and a short-term **reward** and a long-term **value** are fed back into the agent, and the agent adjust itself to suit the environment, until it's intelligent enough.

- **Agent** : The algorithm is an agent that Interact with environment.

- **Action** : The set of all possible moves that the agent may make.

- **Environment** : The world where the agent is.

- **State** : An immediate situation in which the agent is.

- **Policy**: The strategy that the agent uses to determine the next action based on the current state, mapping from state to action.

- **Reward (Feedback)**: measures the quality of a state, representing the short-term metric of the agent's last action.

- **Value** : As opposed to the short-term reward, value measures the value of the long-term reward under the policy.

https://en.wikipedia.org/wiki/File:Reinforcement_learning_diagram.svg

Qualcomm

1.1 AI Overview

◆ 1.1.5 Basic concepts in Machine learning & Deep learning

- Training vs. Inference

- Training, test, and validation data sets

- Basic tasks of machine learning

- Model Evaluation

- Underfitting & Overfitting

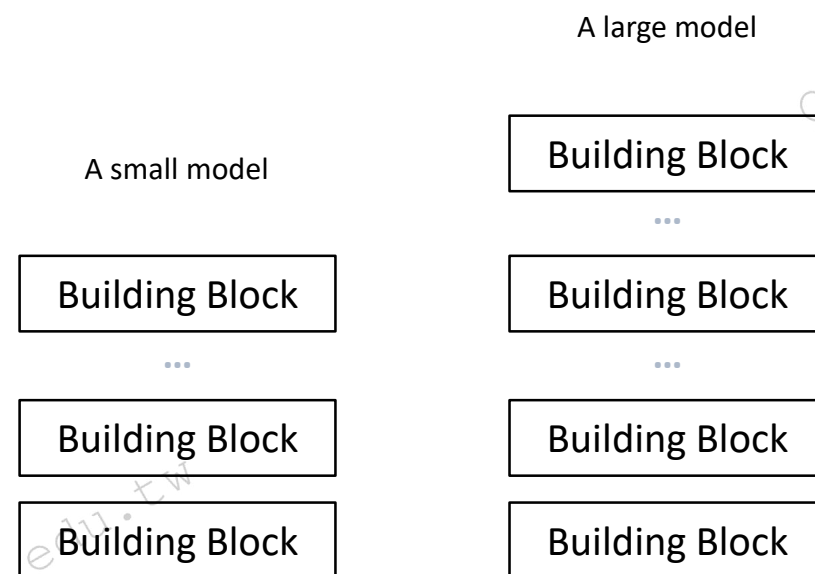Note: Most of the following concepts apply to supervised learning.

Training vs. Inference

- Training is "teaching" the model by feeding pairs of examples and corresponding labels, in order to obtain the ability to predict new data.

- Inference can't happen without or before training. After training, the "trained" model is used to infer the label that is unseen in training stage.

Training	Inference

Labeled data:
{Example, Label}

Model

Example

Algorithm

Prediction

Qualcomm
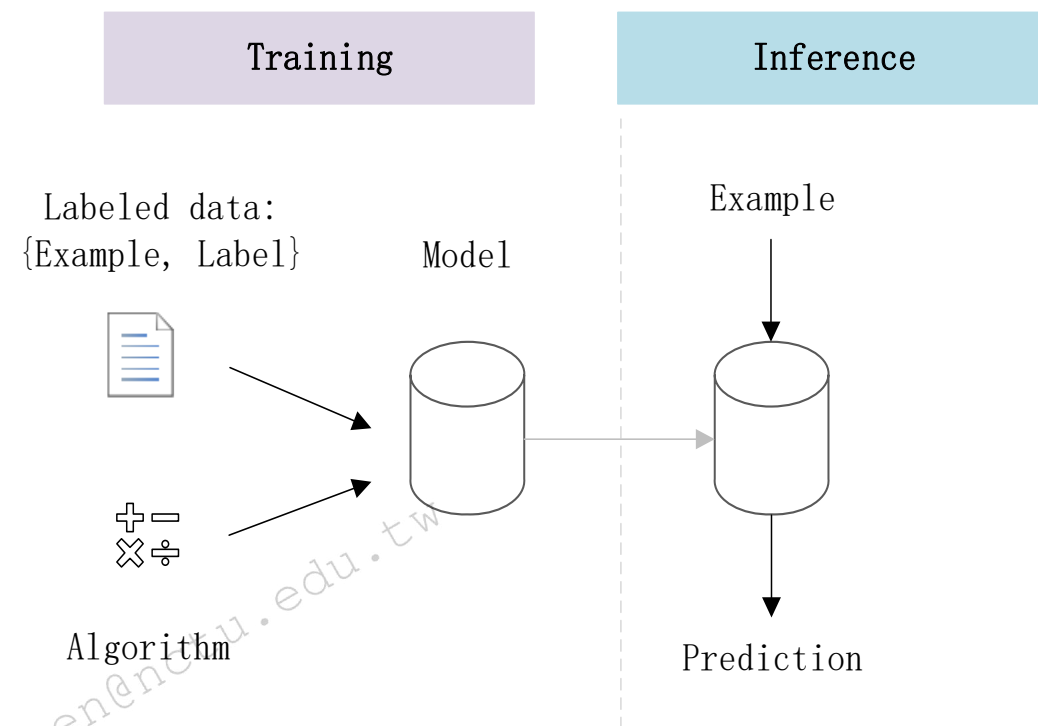
Training, test, and validation data sets

- We need to split the whole dataset into training, test, and validation data sets for different purpose. Generally speaking, training set is for training, and inference happens on test set as well as deployment. But what is validation set?

- First, we need to learn what is **hyper-parameter**. Hyper-parameter is a kind of special parameters about the architecture of the model, which is manually set before the training starts. For example, two models below have the same architecture but different numbers of layers. Layer number is a kind of hyper-parameters.

A large model

A small model

Building Block

Building Block

...

Building Block

...

Building Block

...

Building Block

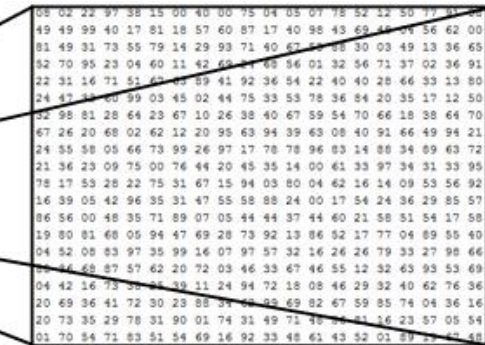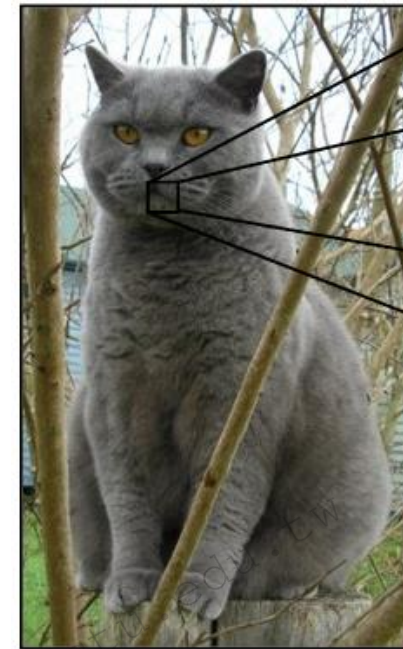Building Block

Building Block

Qualcomm

Training, test, and validation data sets

- We often try to design different models in the same architecture, but with different hyper-parameters. These different models are performed on one and the same dataset – validation set, and the **best model** with the highest performance stands out from the crowd.
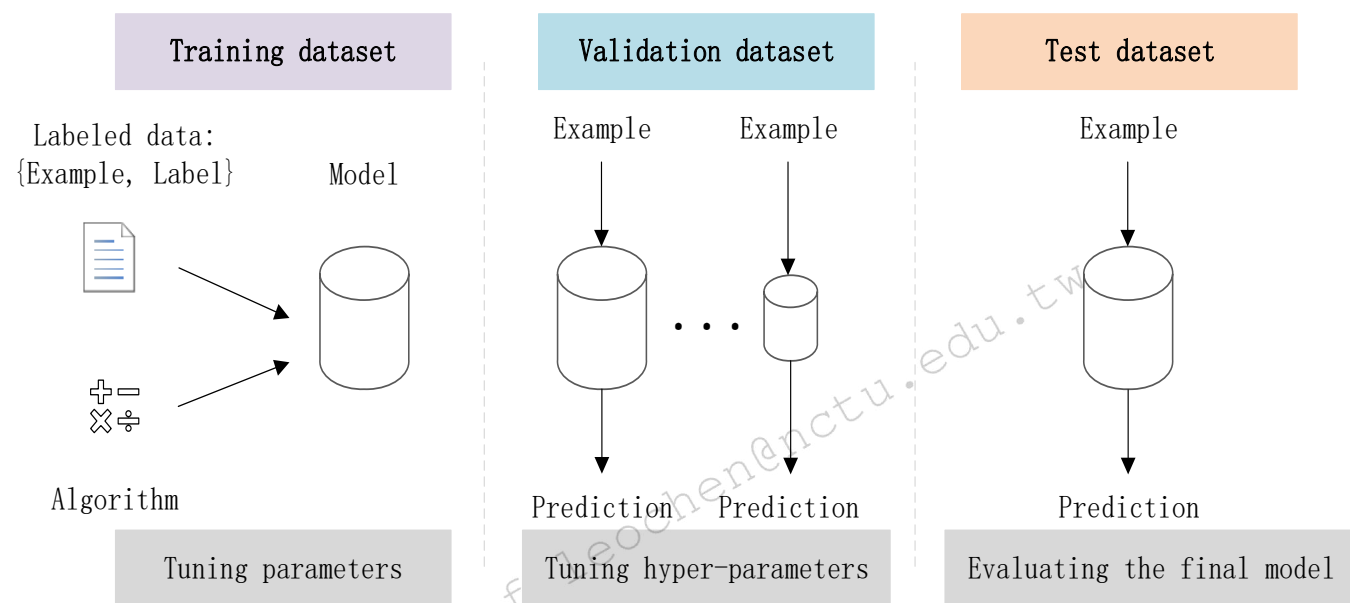
Basic tasks of machine learning

- Classification: predict the category of input example. Categories are sematic words, and are encoded as discrete finite numbers, representing the index of categories. There is a finite set and the result does not exceed this finite set.

- For example, we predict an image is **cat** or **not cat**, output are 1(is cat) and 0(not cat). The finite set is {0, 1}.
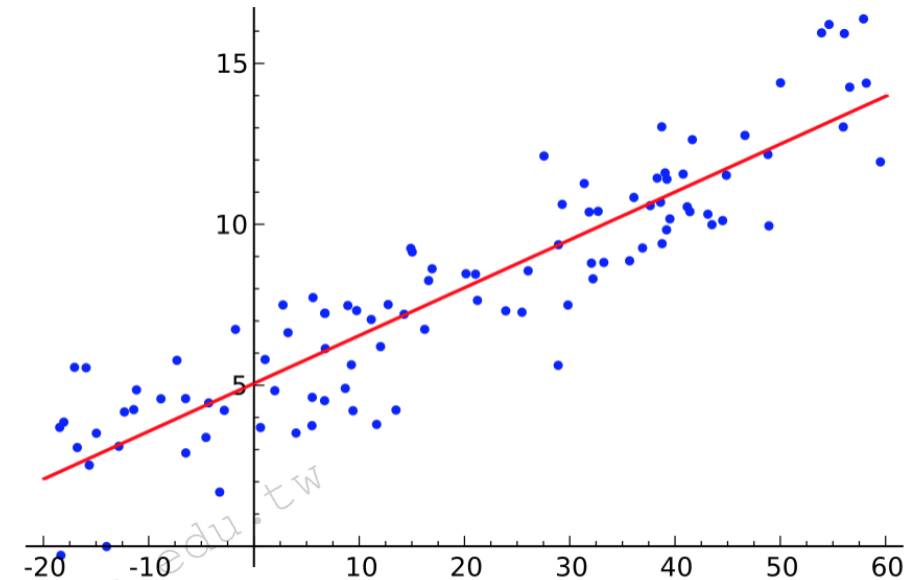
What the computer sees

image classification → 82% cat
15% dog
2% hat
1% mug

https://github.com/cs231n/cs231n.github.io/blob/master/assets/classify.png

Qualcomm

Basic tasks of machine learning

- Regression: the output is a real number, like length, area, coordinate points and so on.

- For example, we estimate the price of a house with the area of it.

Qualcomm

Basic tasks of machine learning

- Object detection: detect instances of semantic categories like animals, persons.

- Image segmentation: divide an image into multiple segments pixel by pixel.
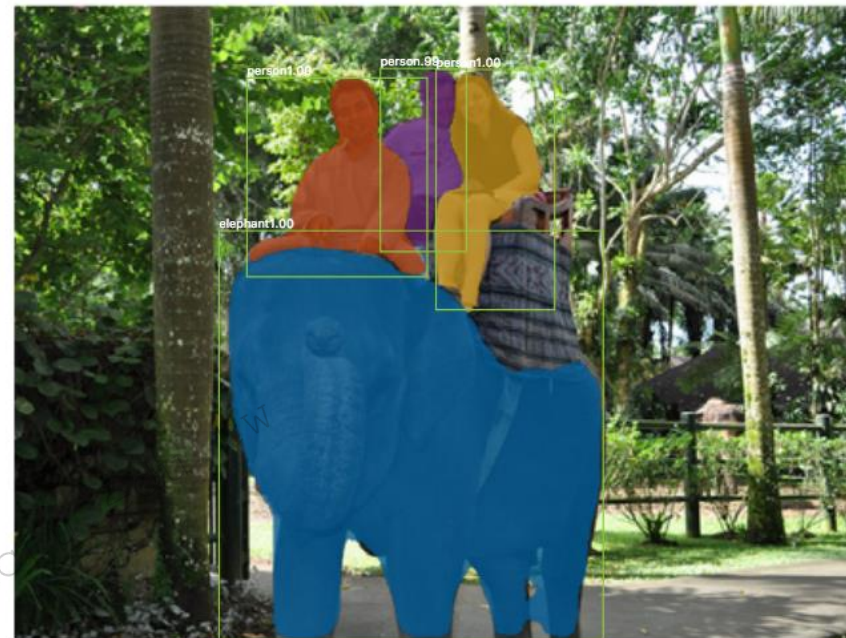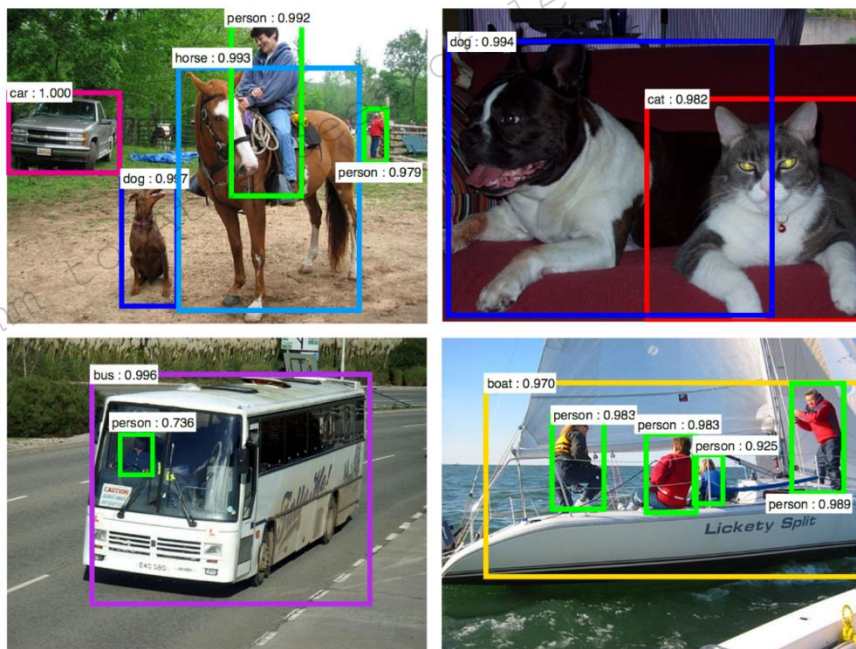
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arxiv 1506.01497
Mask R-CNN, arxiv 1703.06870

Model Evaluation

- **Classification: Precision, Recall, Accuracy.**

- Example: predict an image is cat or not cat.

- **Precision:** Among all images you predict as cat, how many images are real cat?

- **Recall:** Among all real cat images, how many real cat images are picked up?

- **Accuracy:** In your whole predictions, how may images are predicted correctly? (Including is-cat and not-cat)
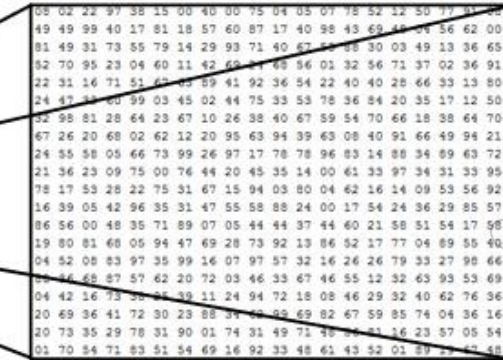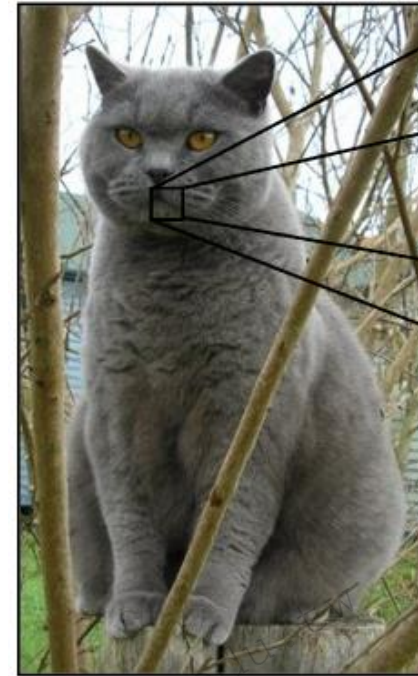
What the computer sees

image classification → 82% cat
15% dog
2% hat
1% mug

Qualcomm

Model Evaluation

- Classification: Precision, Recall, Accuracy.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{60}{60 + 20} = 0.75$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{60}{60 + 40} = 0.6$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{80 + 60}{80 + 60 + 40 + 20} = 0.7$$

Actual labels

	Is-Cat (1)	Not-Cat (0)	
Is-Cat (1)	Type I correct — TP: 60	Type I error — FP: 20	80
Not-Cat (0)	Type II error — FN: 40	Type II correct — TN: 80	120
	100	100	

Predictions

Model Evaluation

- **Regression: Mean squared error (MSE)**

- For a single example of regression, we measure the quality of prediction by the square of difference between the label and the predicted value.

- The performance of a model on validation or test set is calculated by the average error of the entire dataset.

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2$$
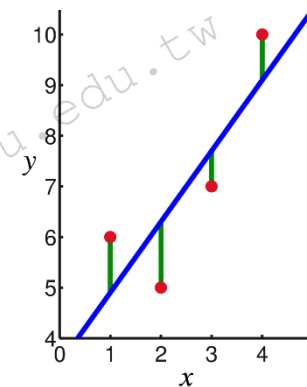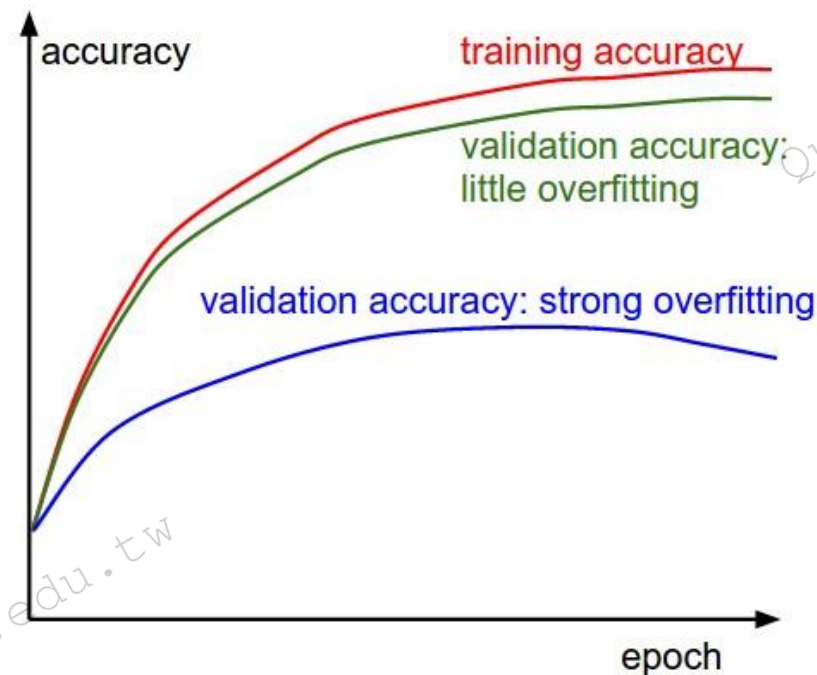
Under-fitting & Over-fitting

What is under-fitting and over-fitting?

Over-fitting:

- What: The prediction of a model is very closely or exactly to the ground truth, but it's poor to generalize to additional data reliably.
- Why: May be the model is so complicated that the training data is less than the model fitting ability.
- How: Loss on training set is small while loss on validation/test set is big.

Under-fitting:

- What: A model couldn't or temporarily cannot learn the underlying law of the data.
- Why: Maybe the training data is too complex and beyond to its fitting capabilities, or the training is just starting and it needs more time.
- How: Loss on training set and validation/test set are both big.

https://github.com/cs231n/cs231n.github.io/blob/master/assets/nn3/accuracies.jpeg

How to deal with Under-fitting & Over-fitting?

Under-fitting

- 1. Wait a moment.
- 2. Increase learning rate.
- 3. Use a new model with more complexity.

Over-fitting

- 1. More training data.
- 2. Data augmentation.
- 3. Reduce the model complexity.
- 4. Add regularization item to loss function.
- 5. Other regularization methods. (like Dropout / Batch norm in deep learning)

Thank You

Qualcomm