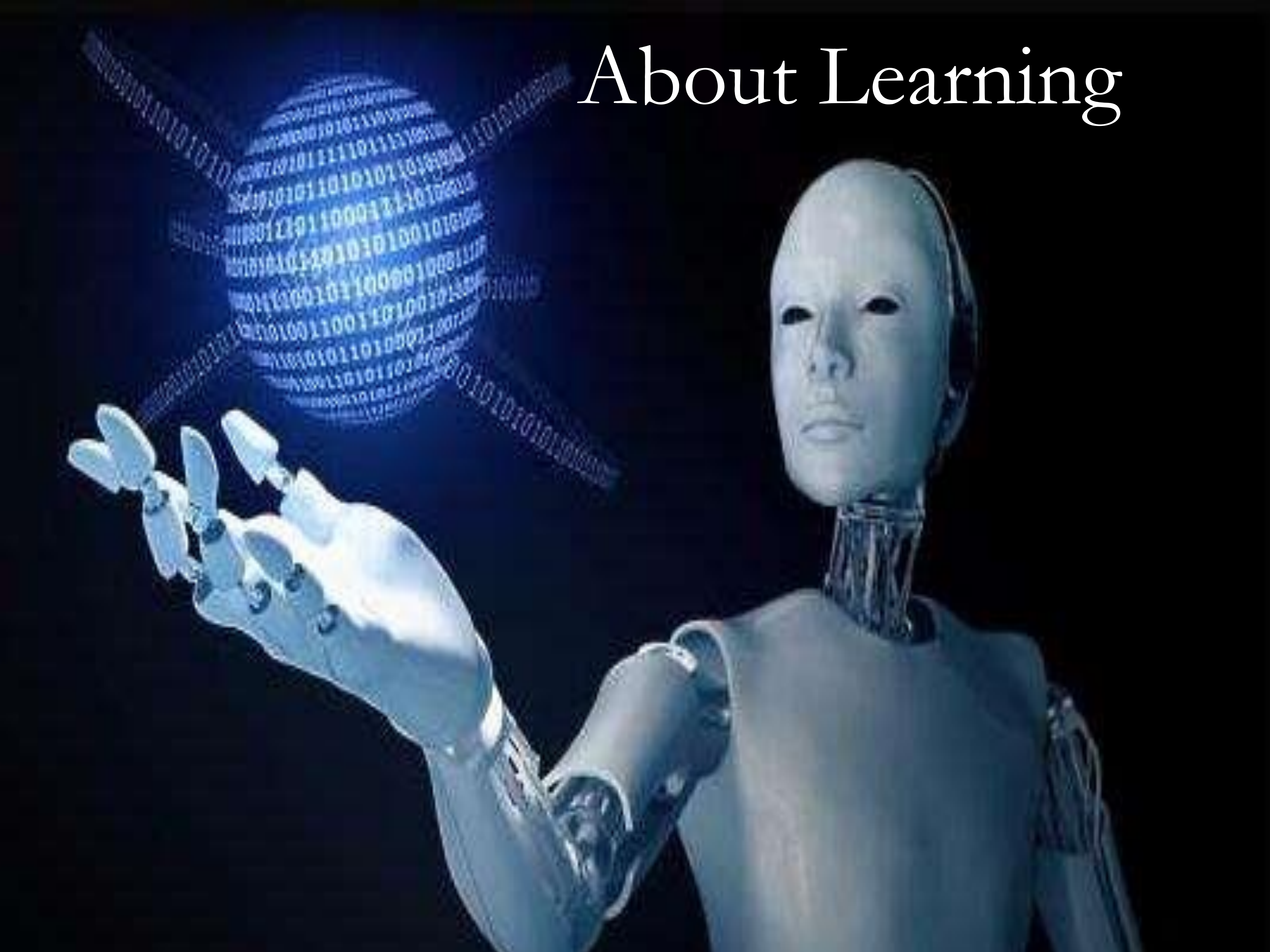
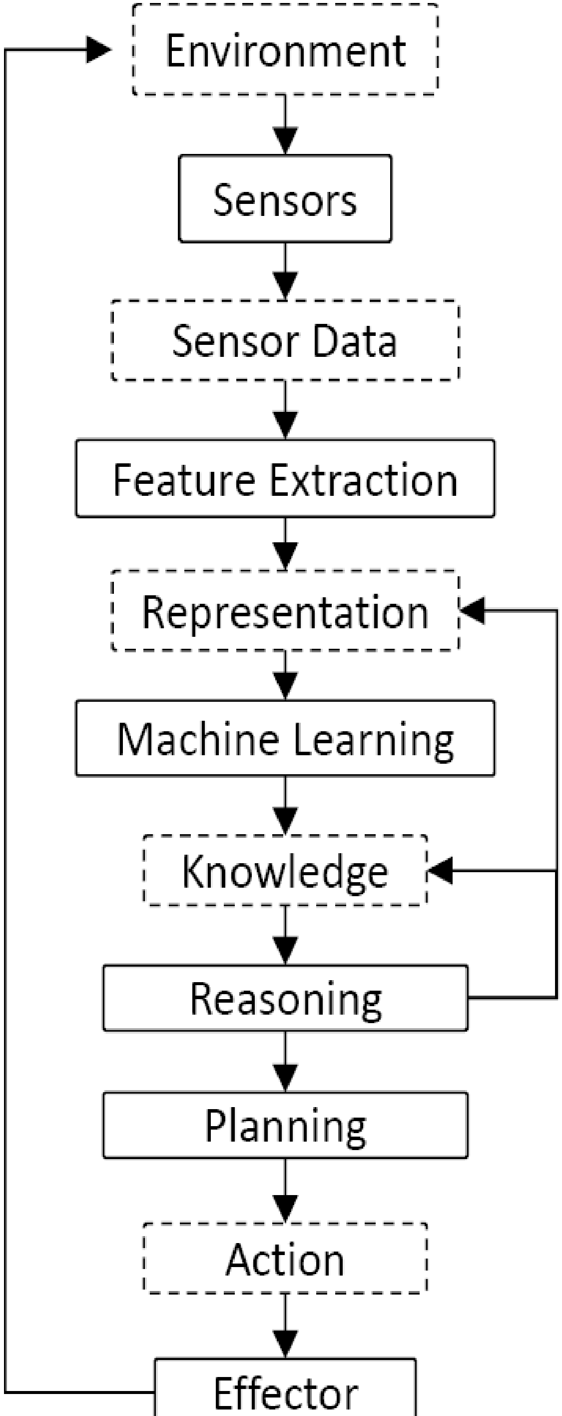
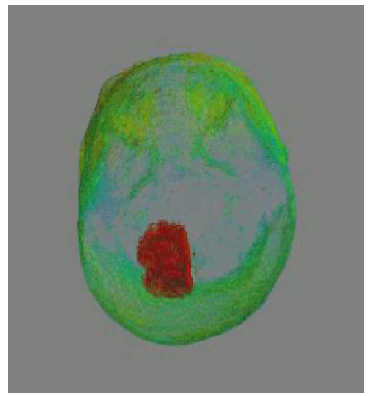
About Learning



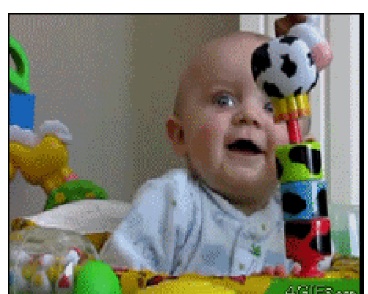


Formal tasks: Playing board games, card games. Solving puzzles, mathematical and logic problems.

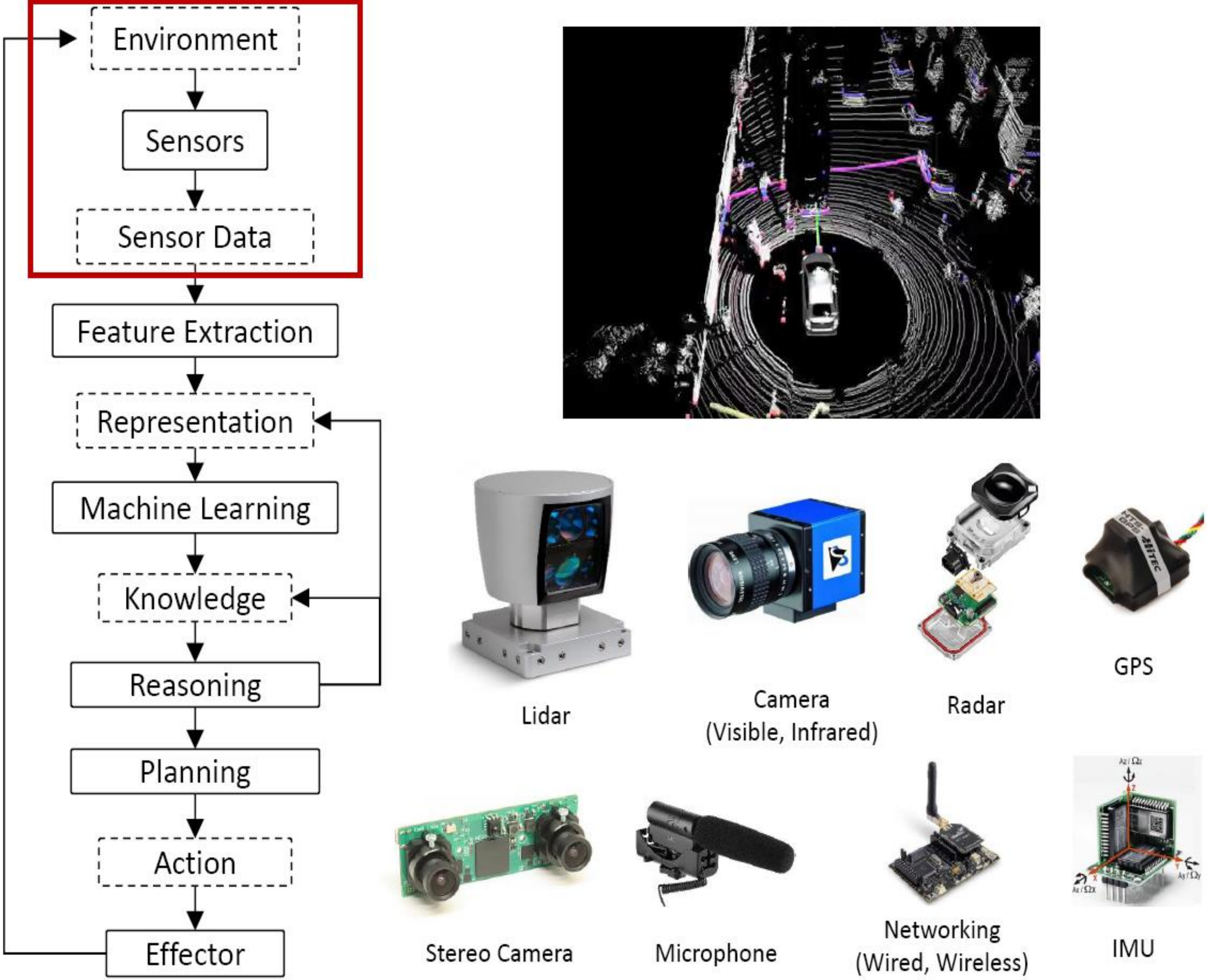


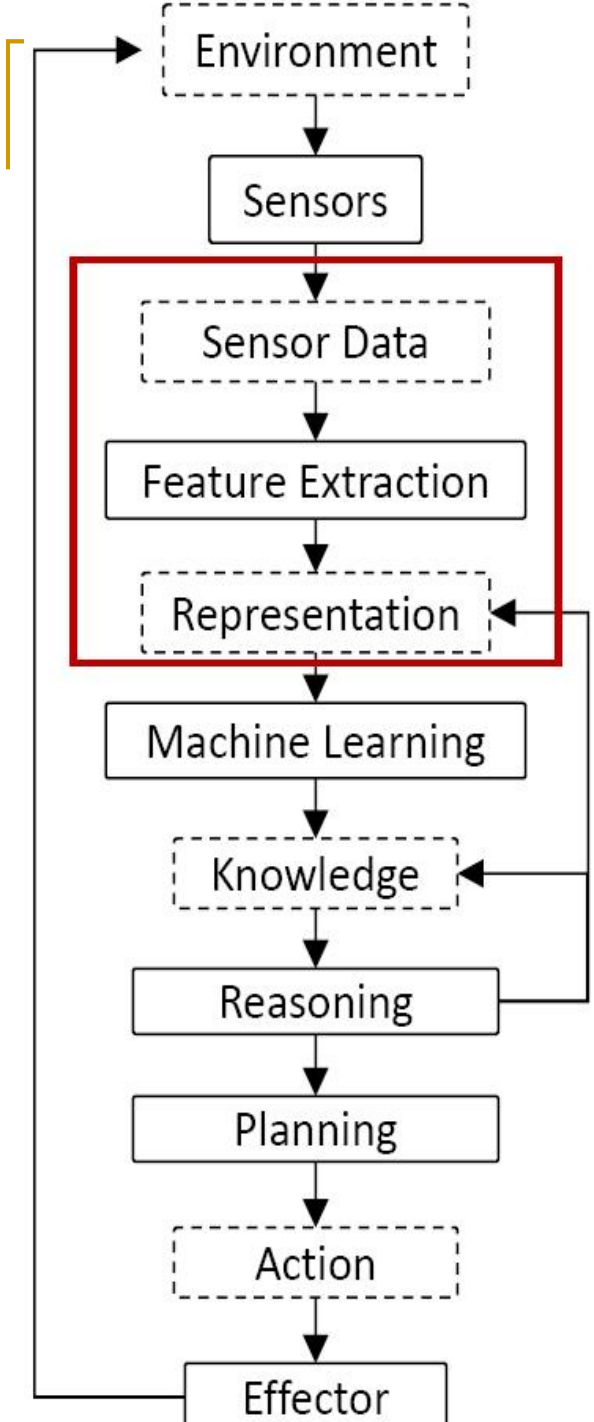
Expert tasks: Medical diagnosis, engineering, scheduling, computer hardware design.

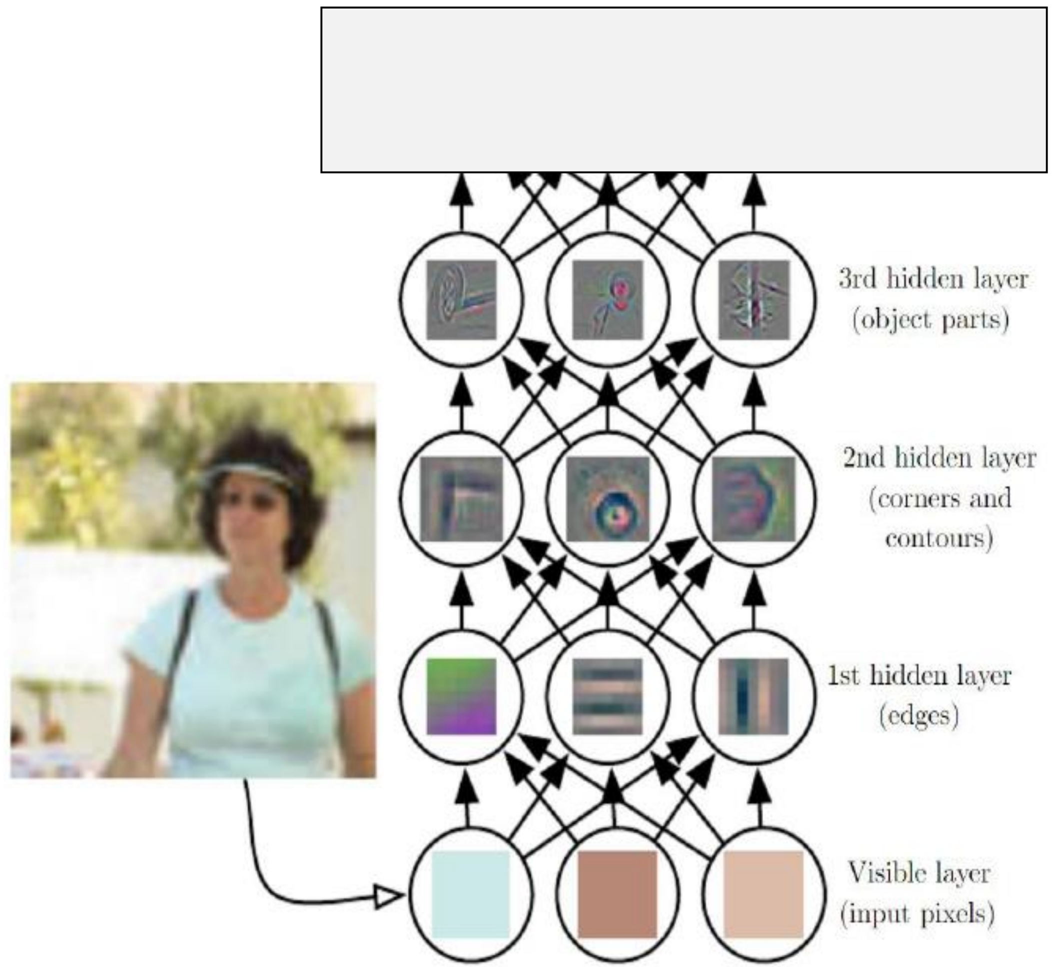
Mundane tasks: Everyday speech, written language, perception, walking, object manipulation.

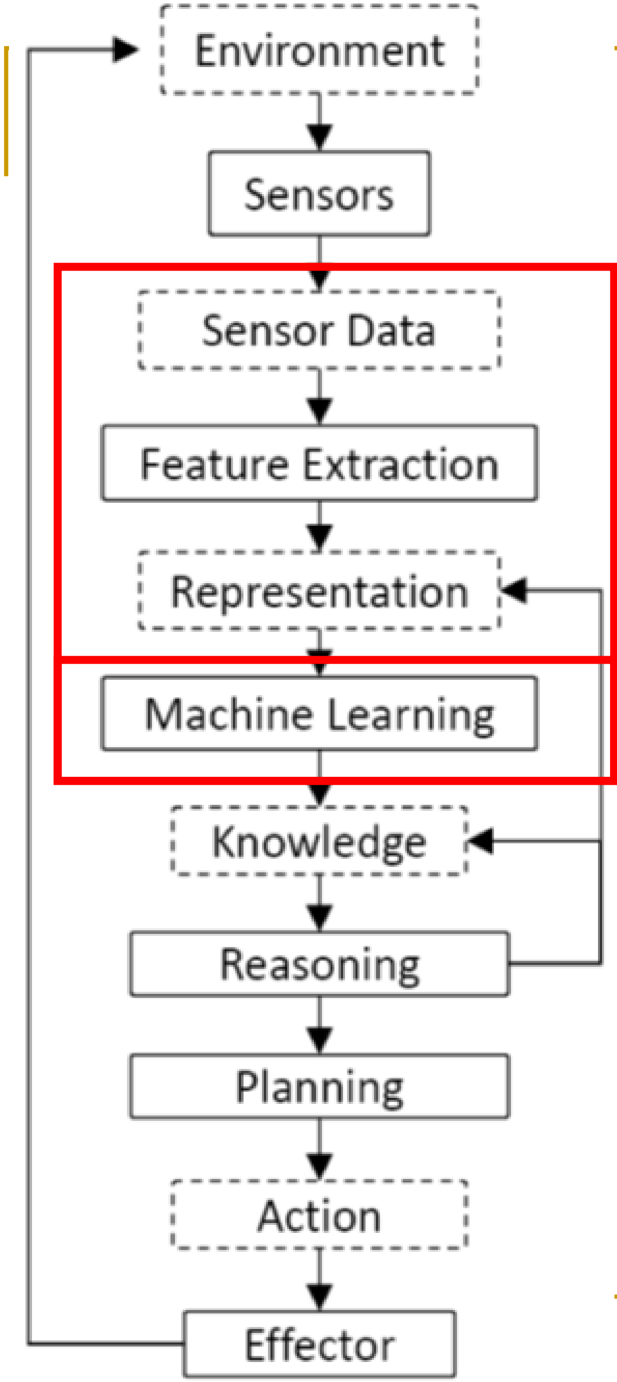


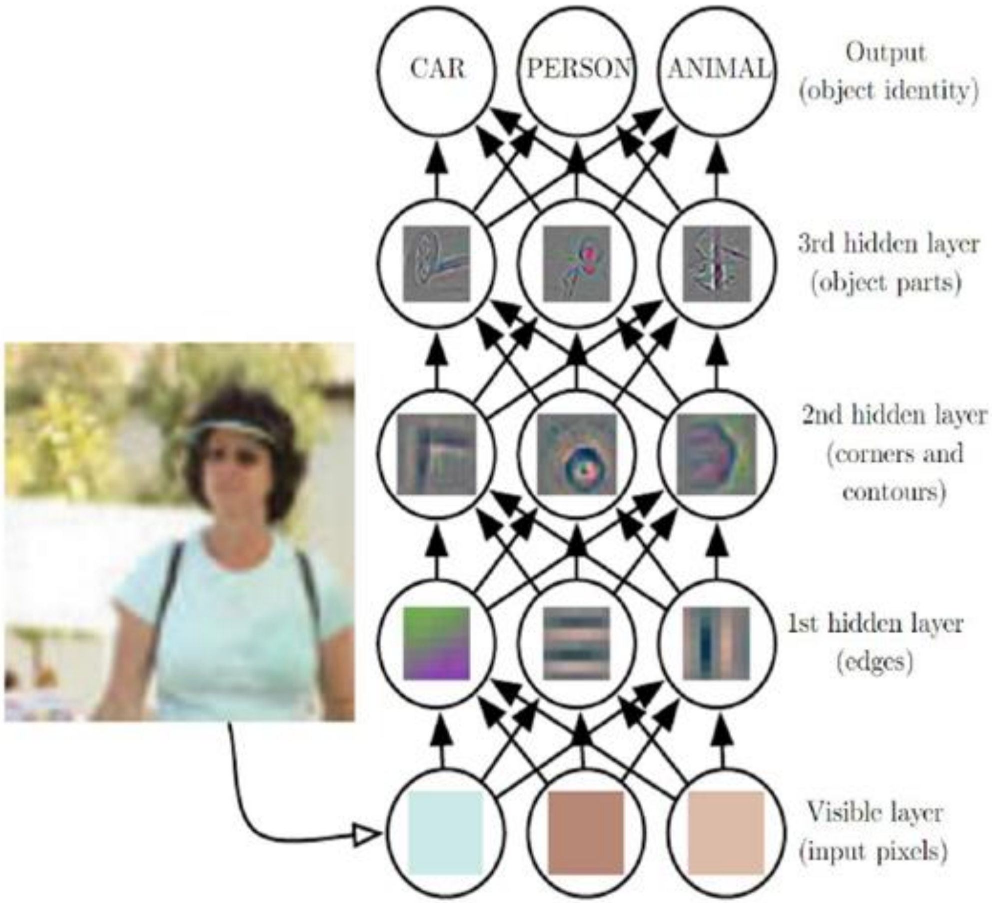
Human tasks: Awareness of self, emotion, imagination, morality, subjective experience, high-level-reasoning, consciousness.


Environment
Sensors
Sensor Data
Feature Extraction
Representation
Machine Learning
Knowledge
Reasoning
Planning
Action
Effector
Lidar
Camera
(Visible, Infrared)
Radar
GPS
Stereo Camera
Microphone
Networking
(Wired, Wireless)
IMU


Environment
Sensors
Sensor Data
Feature Extraction
Representation
Machine Learning
Knowledge
Reasoning
Planning
Action
Effector


3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)


Environment
Sensors
Sensor Data
Feature Extraction
Representation
Machine Learning
Knowledge
Reasoning
Planning
Action
Effector


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)

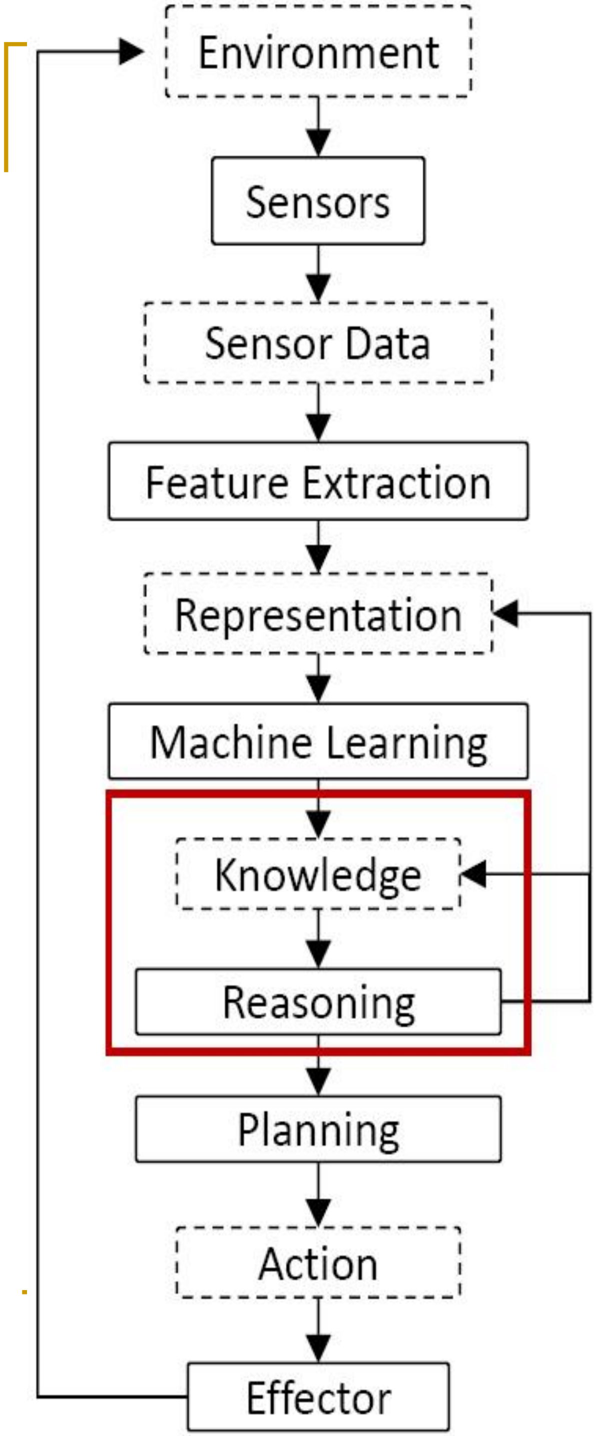
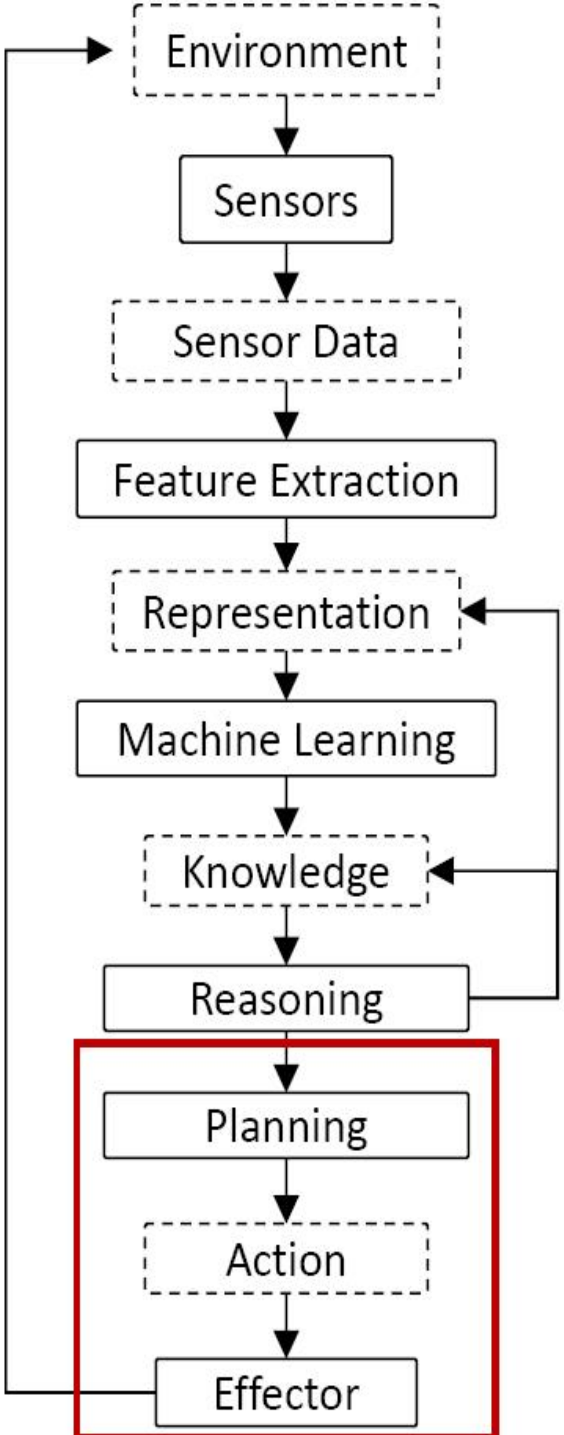
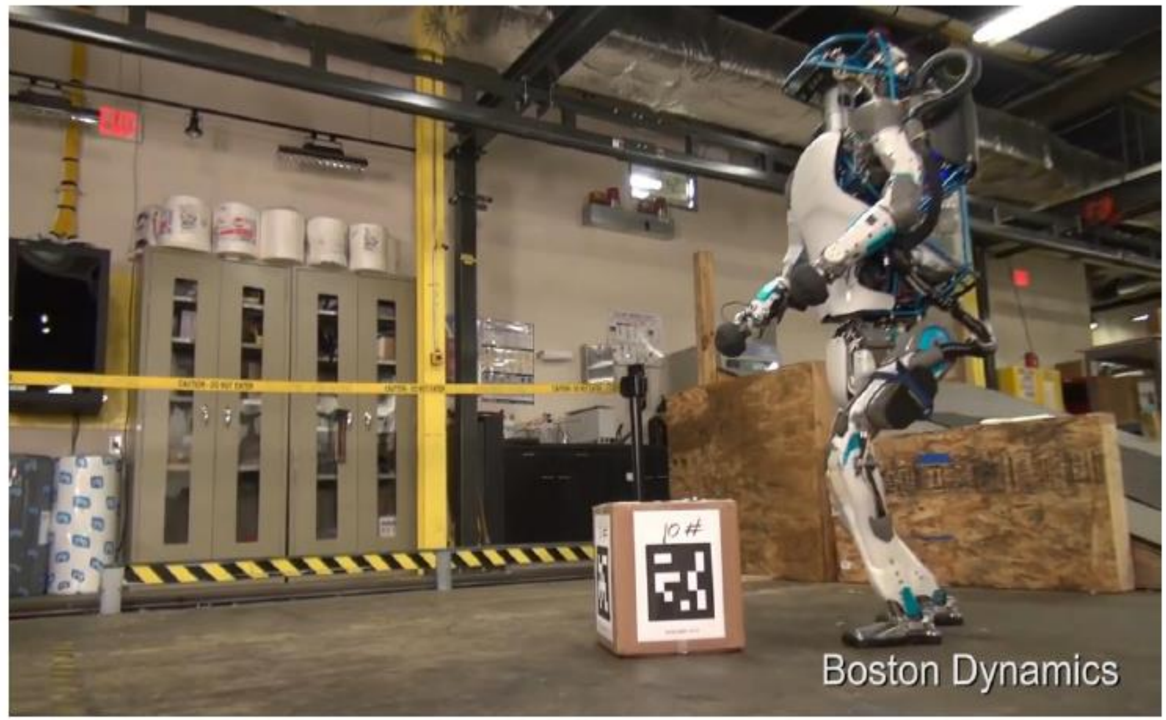


Image Recognition:
If it looks like a duck

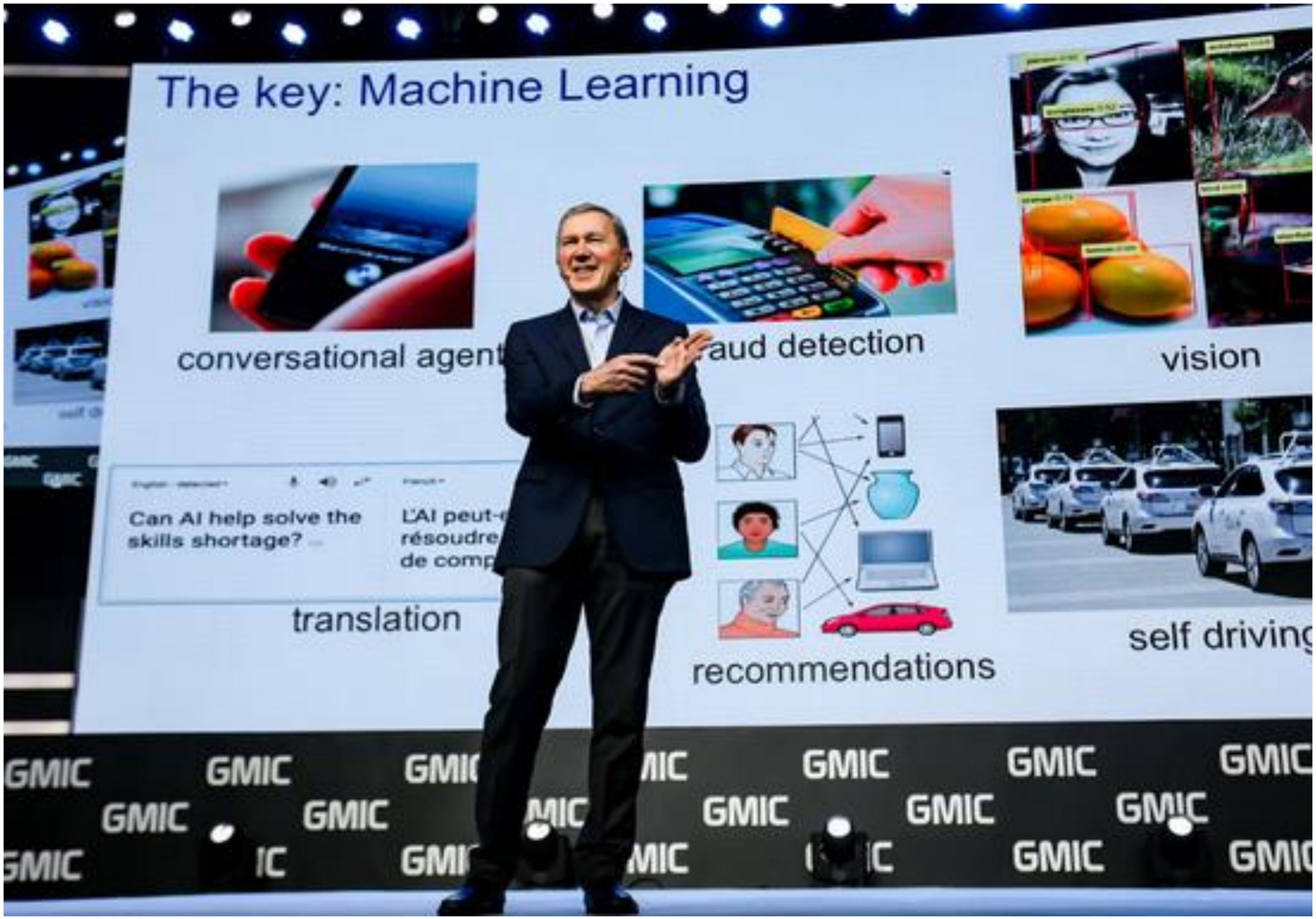
Audio Recognition:
Quacks like a duck

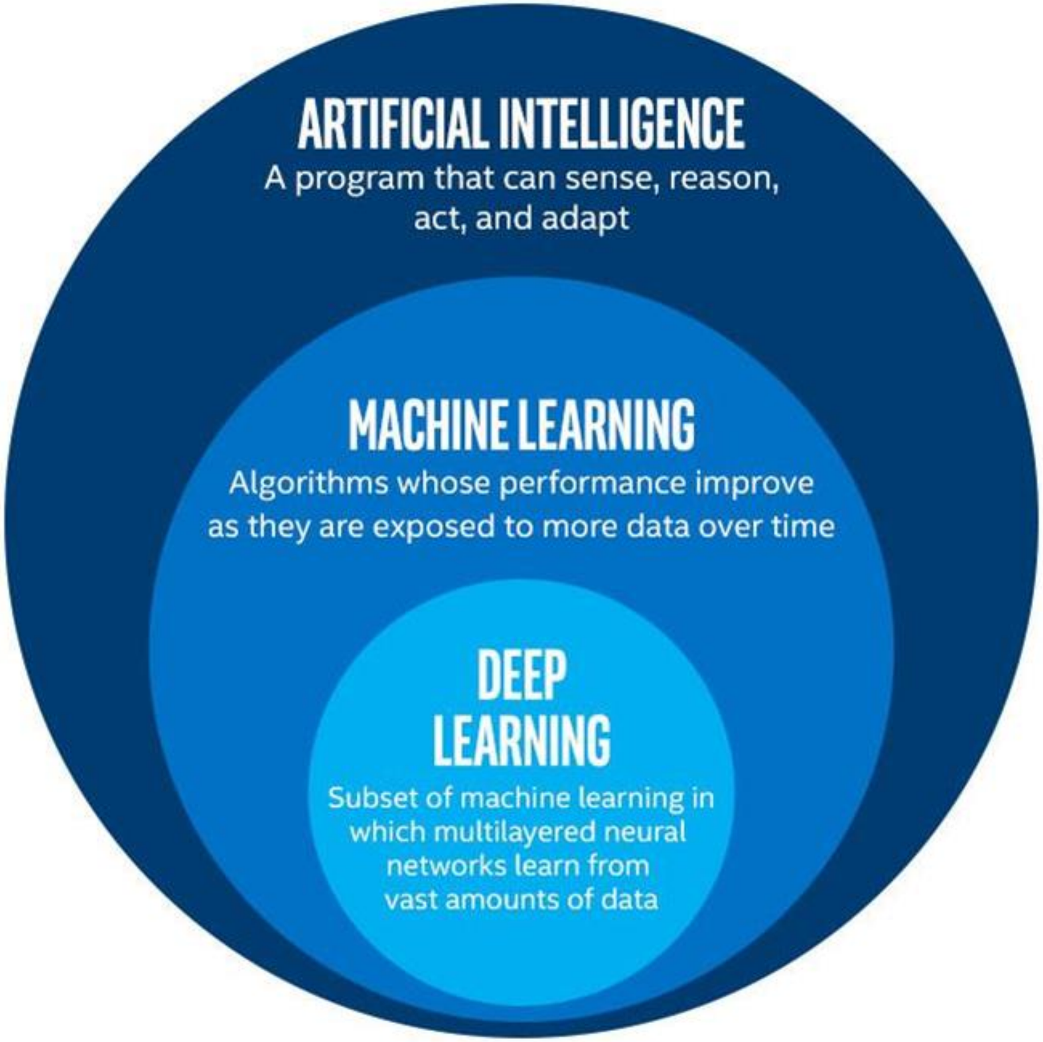
Activity Recognition:
Swims like a duck


Environment
Sensors
Sensor Data
Feature Extraction
Representation
Machine Learning
Knowledge
Reasoning
Planning
Action
Effector


Boston Dynamics

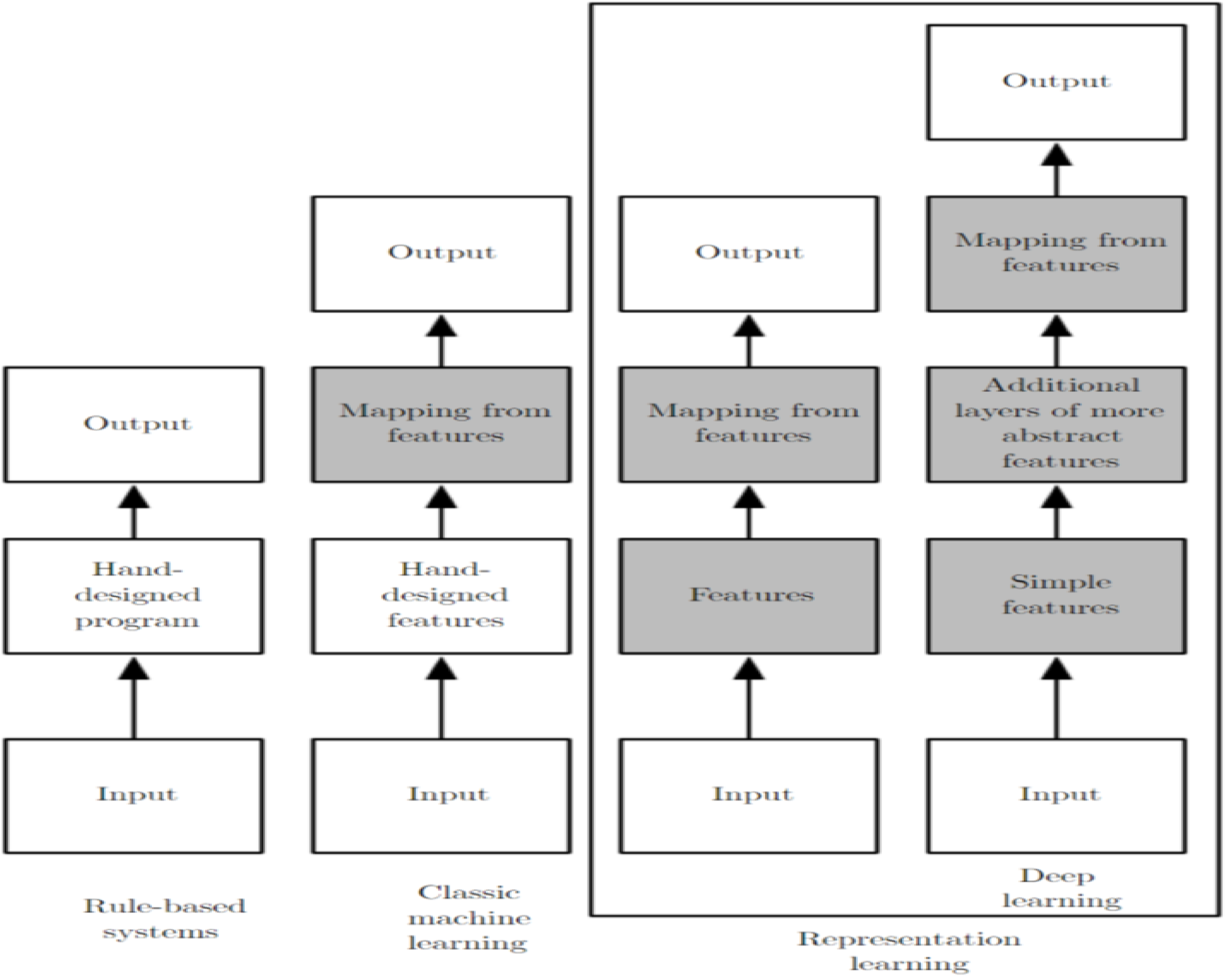
Tom Mitchell @ 2018 GMIC




ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

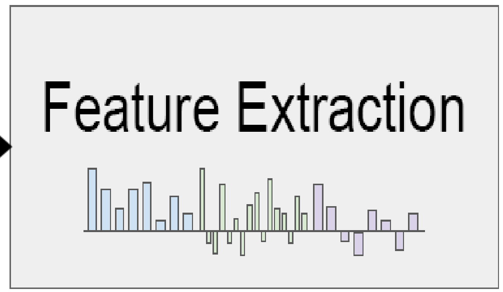
人工智能发展的四个阶段

- 初期阶段
  - 通用问题求解、机器翻译、定理证明、博弈、游戏……
- 知识时代
  - 专家系统、知识工程、知识表示、（不）确定性推理……
- 特征时代
  - 统计机器学习方法、优化技术、特征映射（浅层）、特征工程……
- 数据时代
  - 深度学习、表示学习、自动特征抽取、不同层次的抽象特征、特征映射（深层）……


Output
Hand-designed program
Input
Rule-based systems
Output
Mapping from features
Hand-designed features
Input
Classic machine learning
Output
Mapping from features
Features
Input
Output
Mapping from features
Additional layers of more abstract features
Simple features
Input
Deep learning
Representation learning

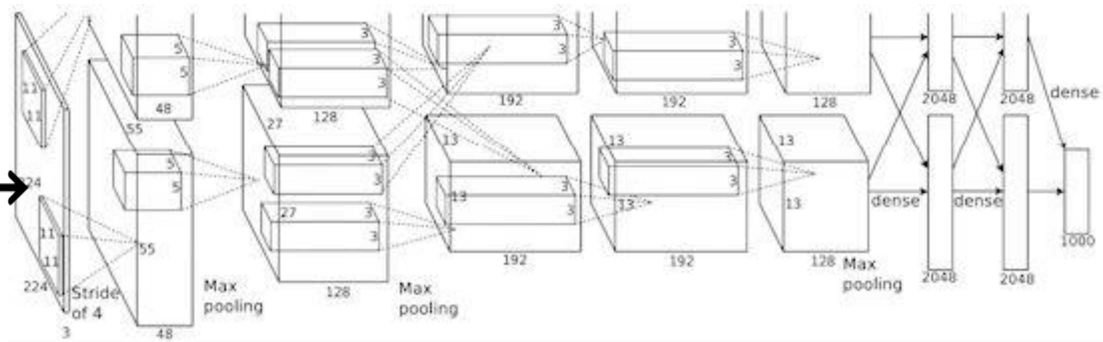
Classic Machine Learning

vs Deep Learning

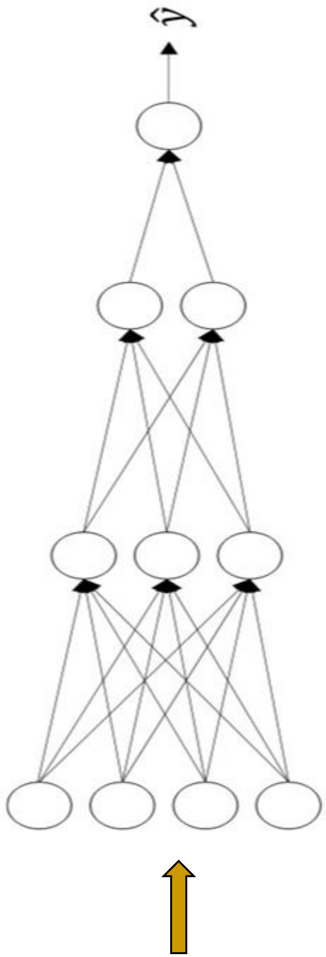


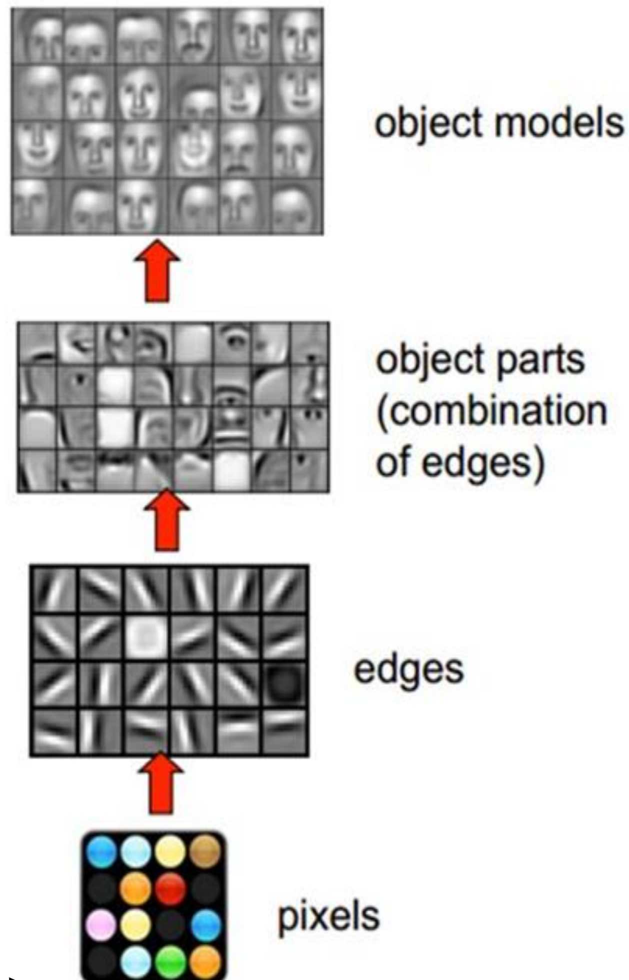
Classifier

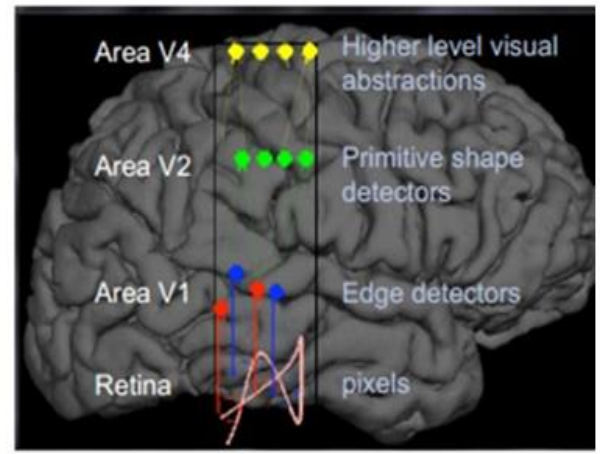
scores for classes



scores for classes

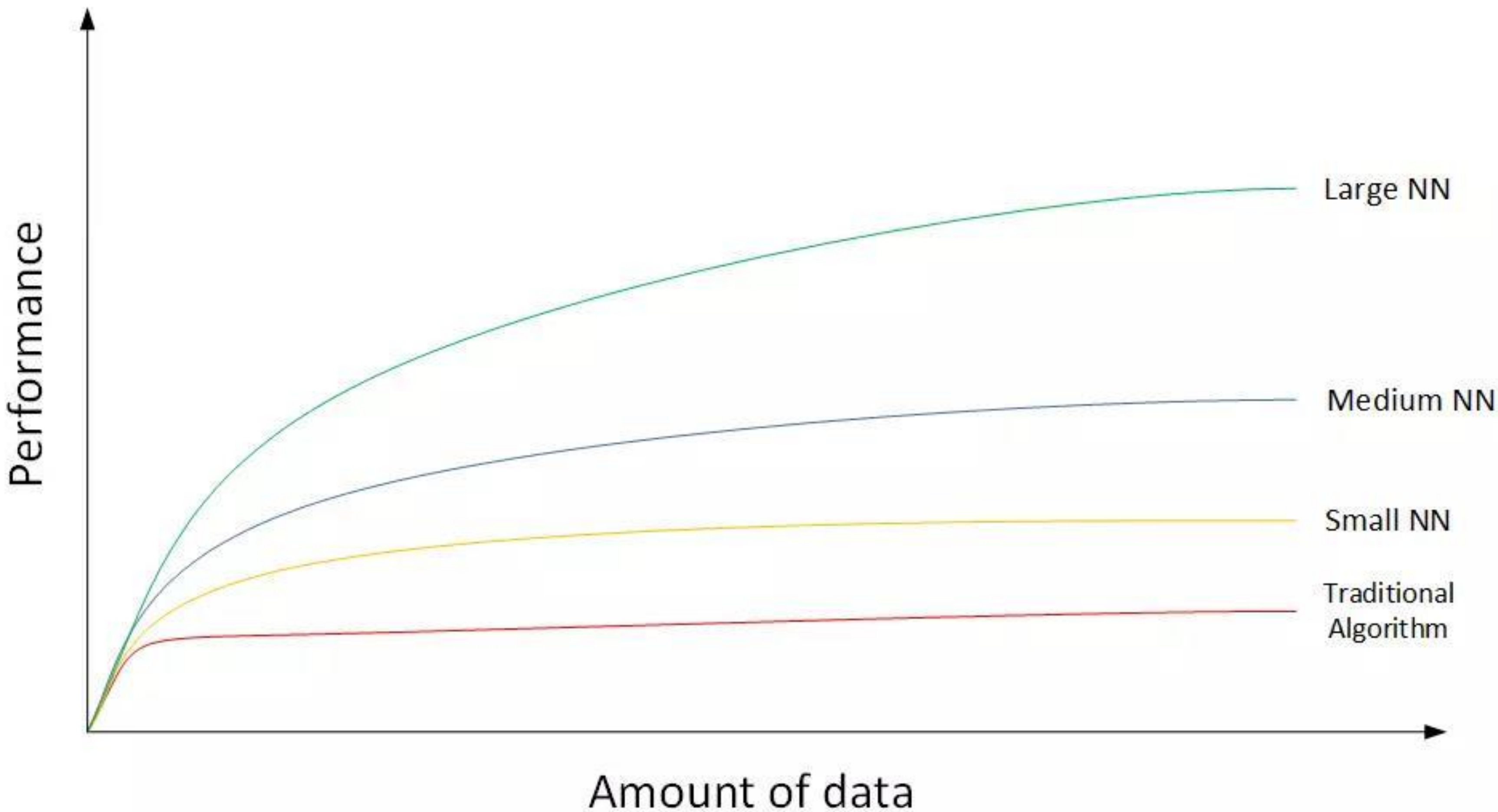






Intuition about Deep Representation

Scale drives deep learning progress



Types of learning task

by Geoffrey Hinton

- Supervised Learning
  - Learn to predict an output when given an input vector
- Unsupervised Learning
  - Learn to discover a good internal representation of the input
- Reinforcement Learning
  - Learn to select an action to maximize payoff

Yann Lecun的“学习蛋糕”

“Pure” Reinforcement Learning (cherry)

- The machine predicts a scalar reward given once in a while.
- **A few bits for some samples**

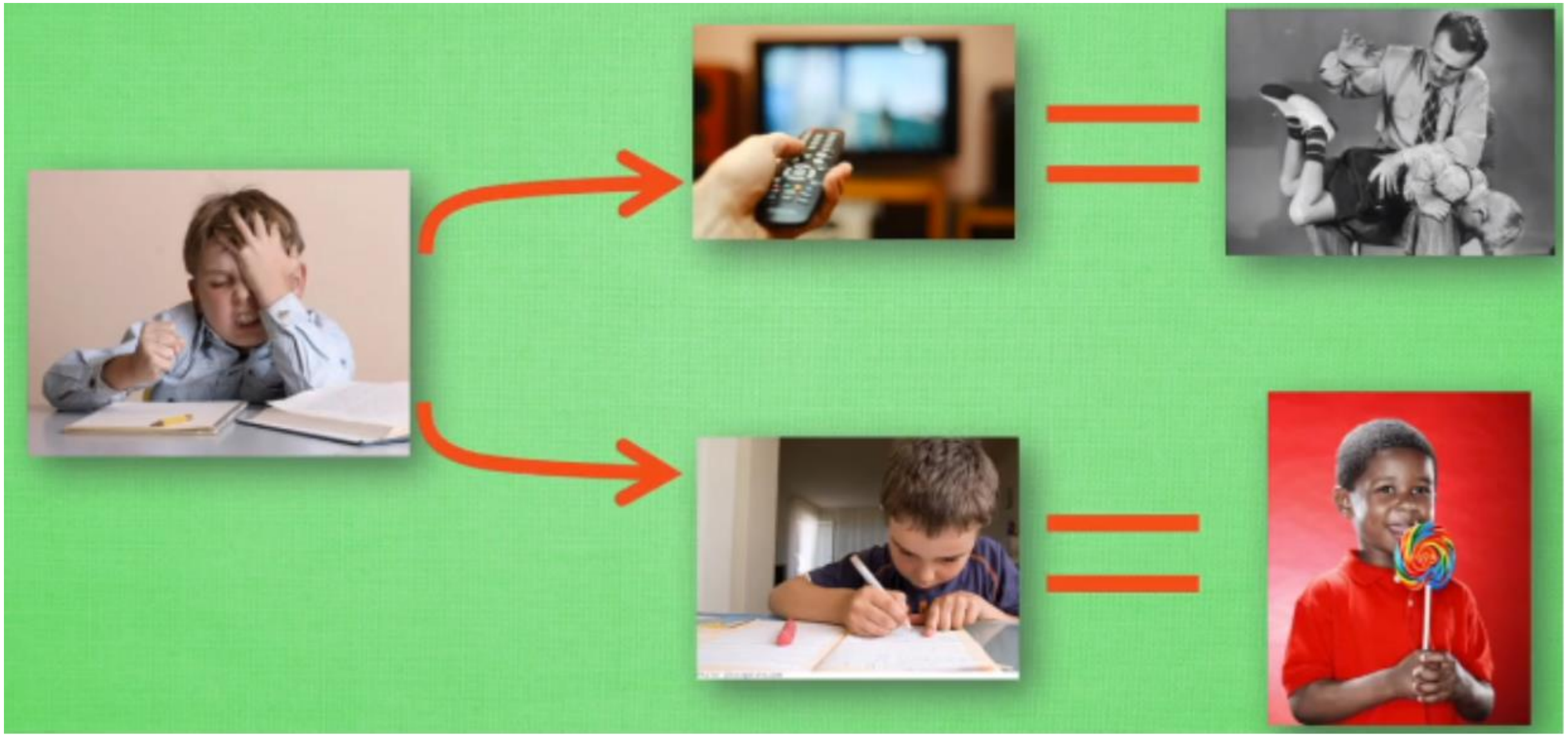
Supervised Learning (icing)

- The machine predicts a category or a few numbers for each input
- Predicting human-supplied data
- **10→10,000 bits per sample**

Self-Supervised Learning (cake génoise)

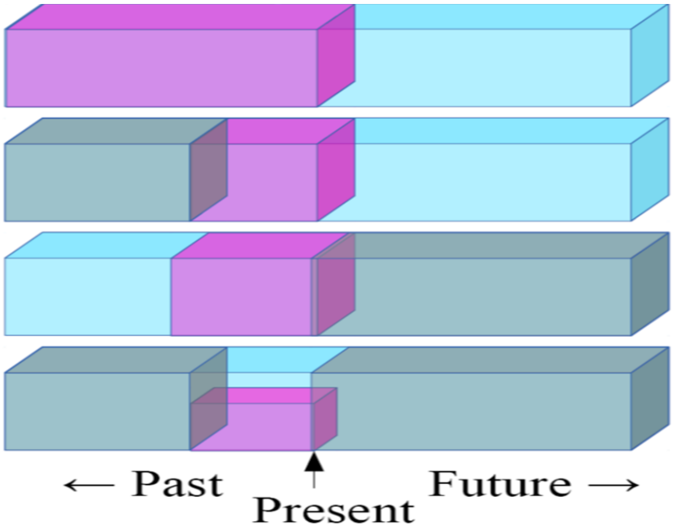
- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- **Millions of bits per sample**

Reinforcement Learning



Self-Supervised Learning

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible.



- Pretend there is a part of the input you don’t know and predict that.

Self-Supervised Learning: Filling in the Blanks

Examples of Supervised Learning

Input(x)	Output(y)	Application
Home features	Price	Real Estate
Ad,user info	Click on ad?(0/1)	Online Advertising
Image	Object(1,⋯,1000)	Photo tagging
Audio	Text transcript	Speech recognition
English	Chinese	Machine translation
Image,Radar info	Position of other cars	Autonomous driving

Data-driven approach

Supervised Learning

- **Collect** a **dataset** and labels
- **Design & Train** a **model**
- **Evaluate** the model on a withheld set of test **data**

Structured Data vs. Unstructured Data

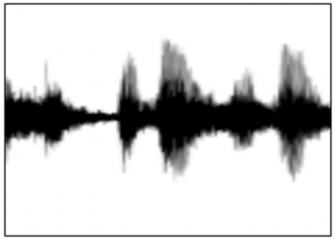
Supervised Learning

Structured Data

Size	#bedrooms	...	Price (1000$s)
2104	3		400
1600	3		330
2400	3		369
⋮	⋮		⋮
3000	4		540

User Age	Ad Id	...	Click
41	93242		1
80	93287		0
18	87312		1
⋮	⋮		⋮
27	71244		1

Unstructured Data



Audio

Image

Four scores and seven years ago...

Text

Regression vs. Classification

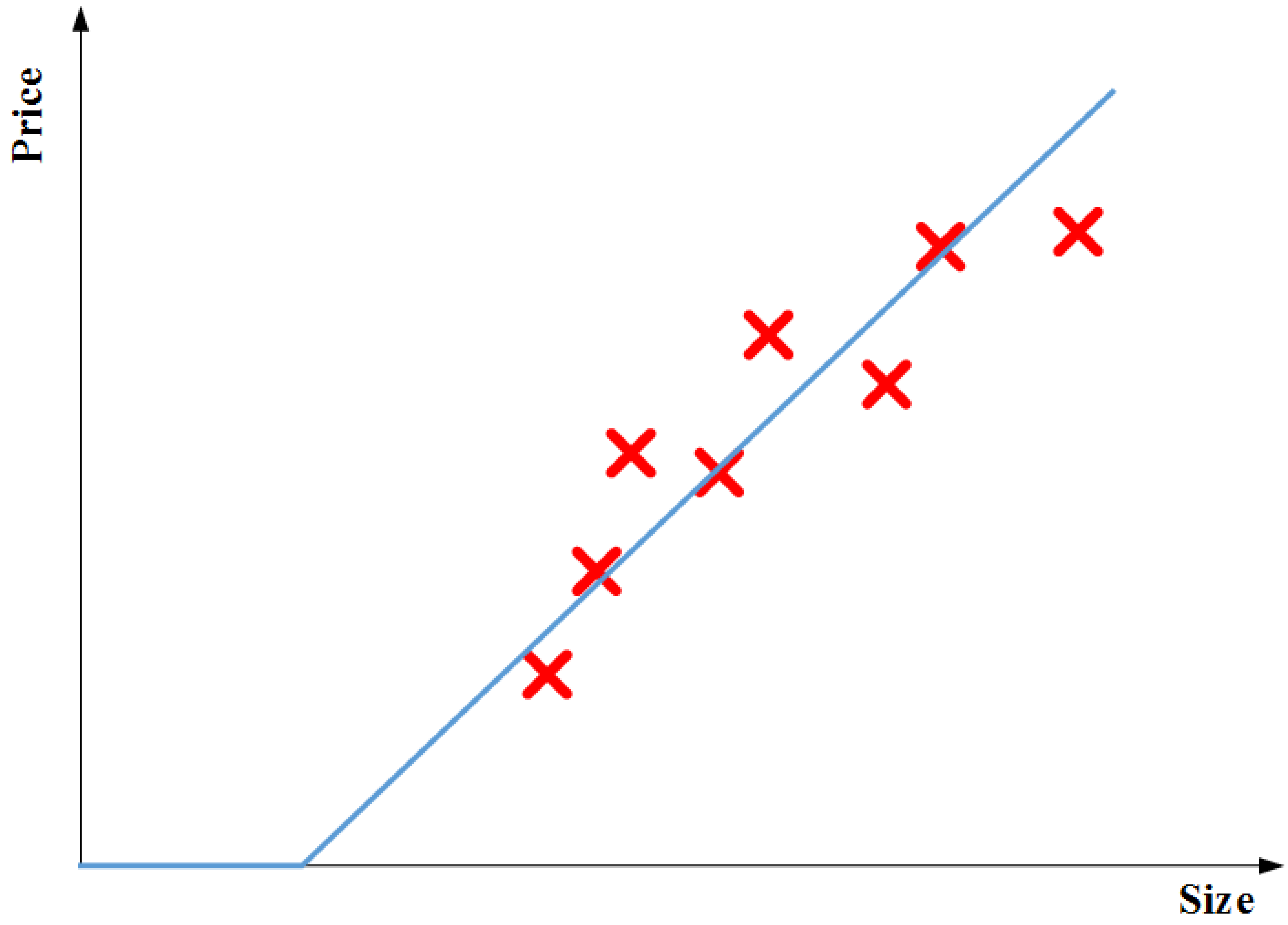
- Regression
  - Predict continuous valued output

Size	#bedrooms	...	Price (1000$s)
2104	3		400
1600	3		330
2400	3		369
⋮	⋮		⋮
3000	4		540

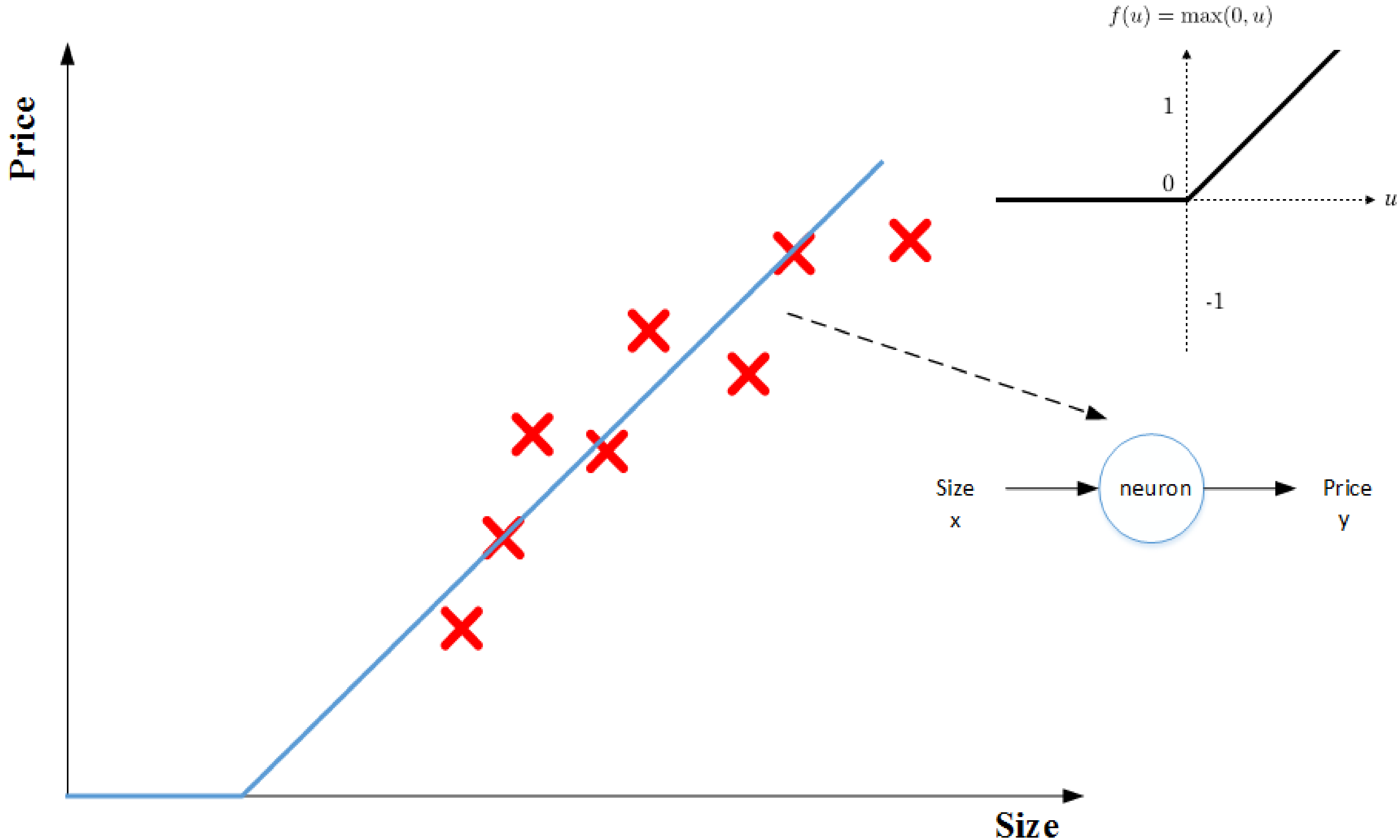
- Classification
  - Output a small number of discrete values

Regression

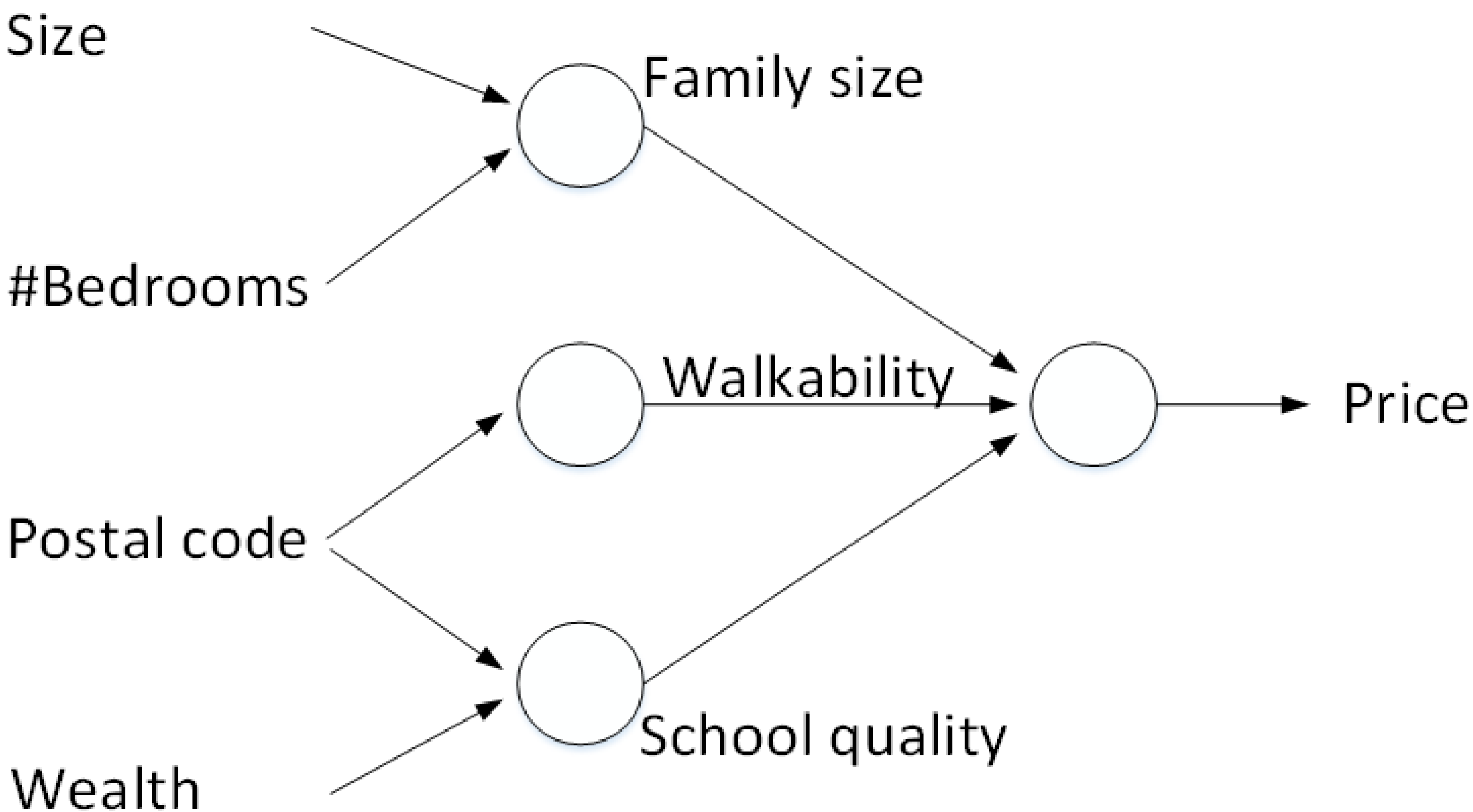
Housing Price Prediction

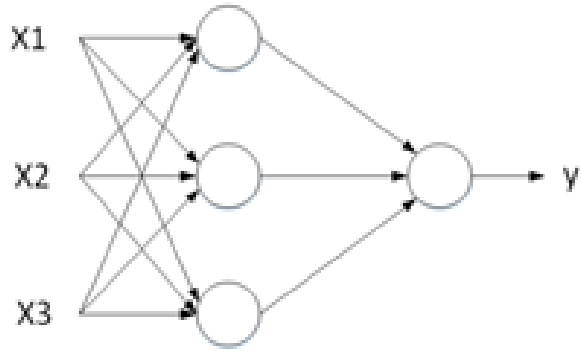


Housing Price Prediction

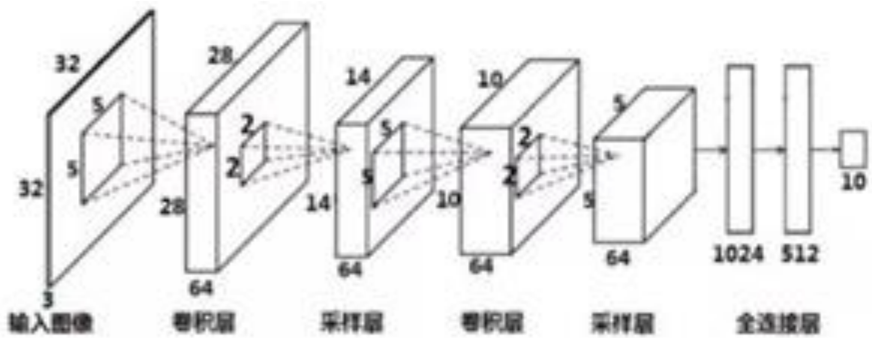


Housing Price Prediction

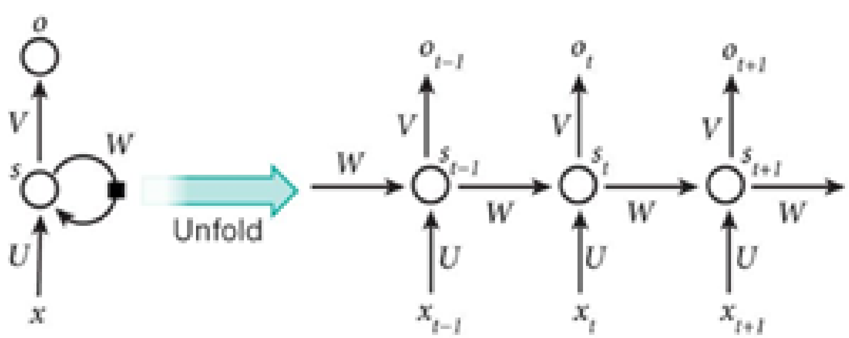




Standard Neural Network (NN)

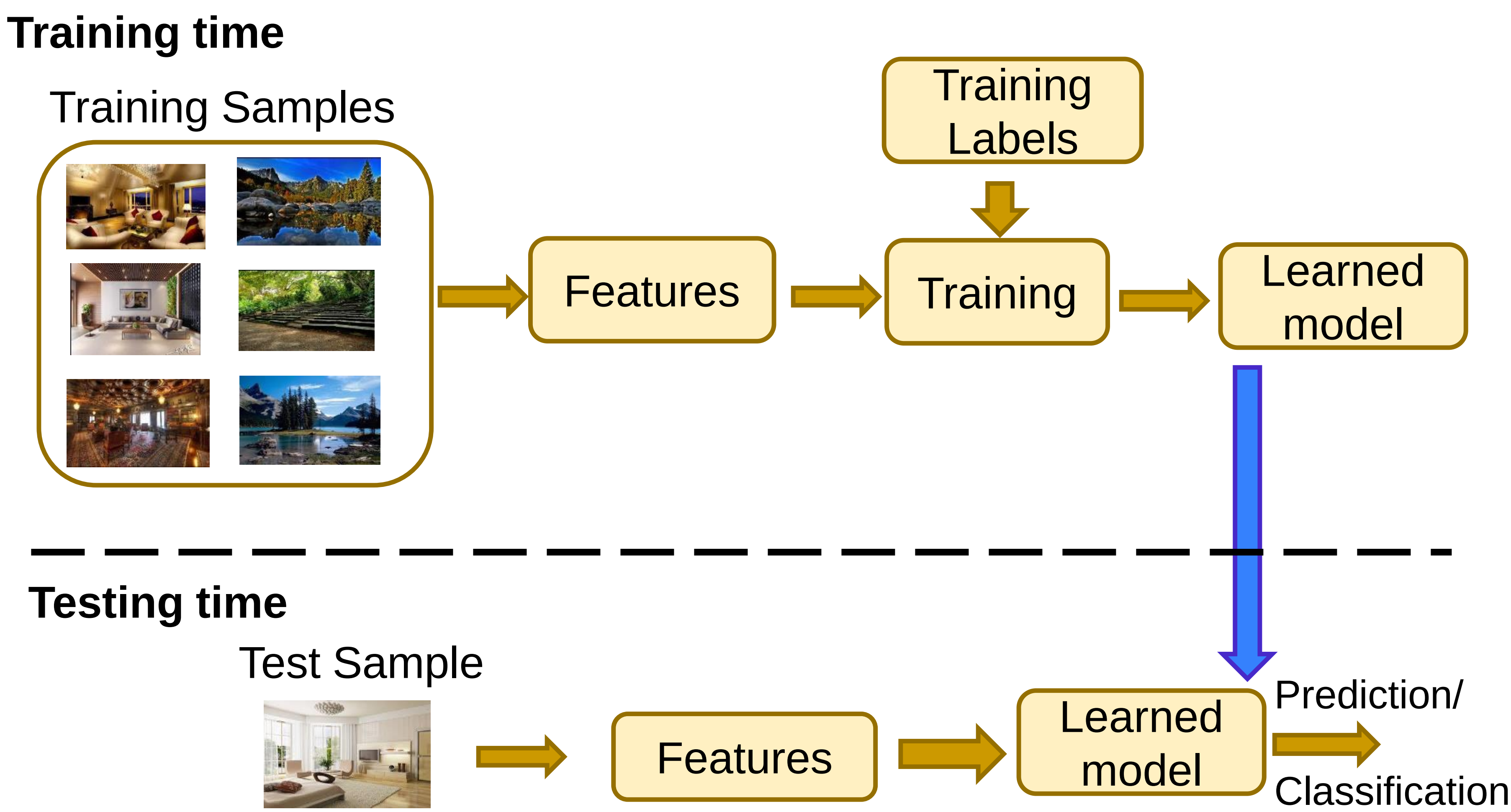


Convolutional Neural Network (CNN)



Recurrent Neural Network (RNN)

Basic supervised learning framework



Basic supervised learning framework

$$\mathrm{y} = \mathrm{f}(\mathbf{x})$$

output | prediction function | input

- **Training** (or **learning**): given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, instantiate a predictor f
- **Testing** (or **inference**): apply f to a new *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$
- What is the connection between training and test data?

Formalization

- Given: training data $\{(x_i, y_i), i = 1, \ldots, n\}$
- Find $y = f(x)$
- S.t. f works well on *test* data

Source: [Y. Liang]

Formalization

- Given: training data $\{(x_i, y_i), i = 1, \ldots, n\}$
- Find $y = f(x) \in \mathcal{H}$
- S.t. f works well on *test* data

Hypothesis class

Source: Y. Liang

Formalization

- Given: training data $\{(x_i, y_i), i = 1, \ldots, n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$
- S.t. f works well on *test* data i.i.d. from distribution D

Have the same distribution

i.i.d.: independently

identically distributed

Source: Y. Liang

Formalization

- Given: training data $\{(x_i, y_i), i = 1, \ldots, n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$
- S.t. f works well on *test* data i.i.d. from distribution D

What kind of performance measure?

Source: Y. Liang

Formalization

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

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

Source: Y. Liang

Formalization

- Given: training data $\{(x_i, y_i), i = 1, \dots, n\}$ i.i.d. from distribution D
- Find $y = f(x) \in \mathcal{H}$ that minimizes

$$\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f, x_i, y_i)$$

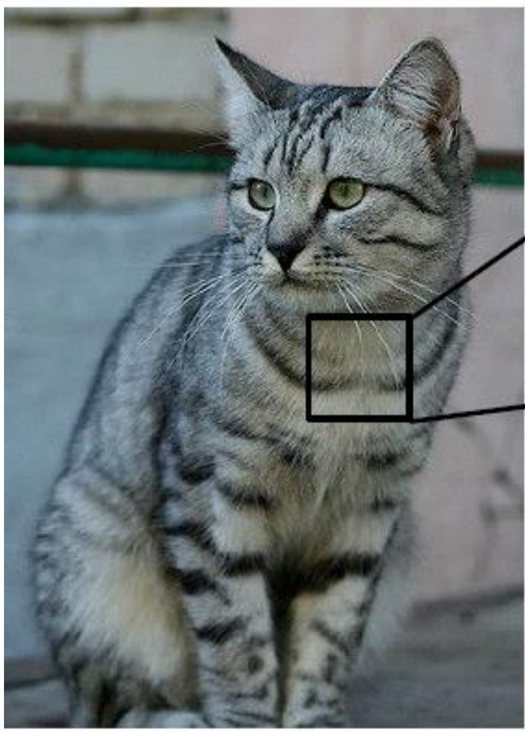
Empirical loss

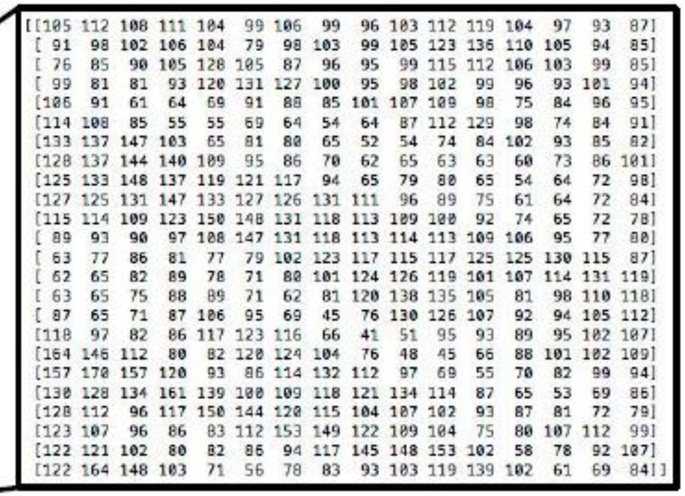
Source: Y. Liang

An Image Classification Example

A core task in Computer Vision - Image Classification

Assigning a single label to an image from a fixed set of categories





What the computer sees

Images are represented as 3D arrays of numbers, with integers between [0, 255]

0 - black   255 - white

eg： 248 x 400 x 3 = 297,600

(3 for 3 color channels RGB)

Linear Classifier

10x1

10x1

$f(x, W) = Wx \; (+\, b)$ → class scores (eg. 10 classes)

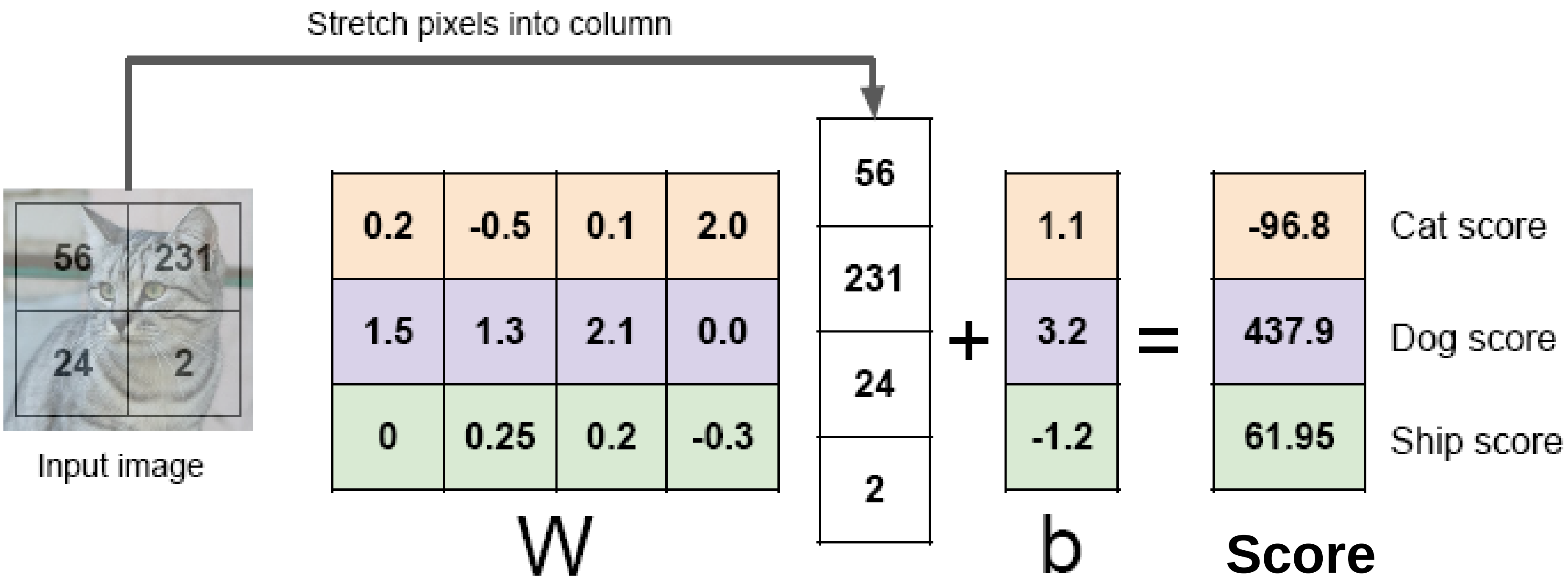
Array of 32x32x3 numbers (3072 numbers total)

Input 3072x1

Parameters or weights: 10x3072

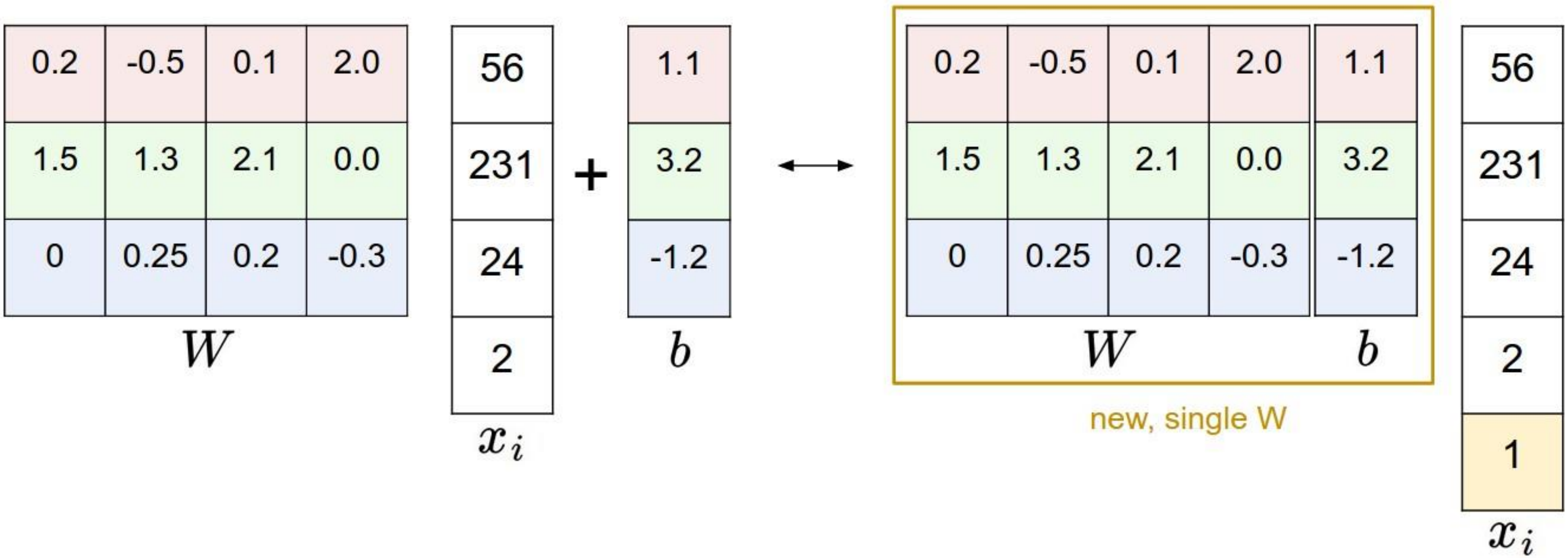
Example:

- An image with 4 pixels, and 3 classes (cat/dog/ship)



Bias trick

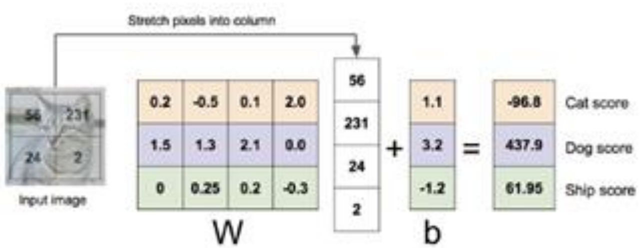
- Representing the two parameters W and b as one



Linear Classifier: Three Viewpoints

Algebraic Viewpoint

$f(x,W) = Wx + b$

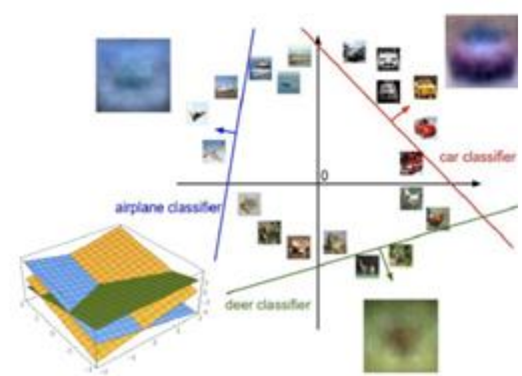


Visual Viewpoint

One template
per class

Geometric Viewpoint

Hyperplanes
cutting up space



How to tell whether W is good/bad?

- Quantifying what it means to have a "good" W

- **Loss function：**
  - Measure the quality of a particular set of parameters W
  - Based on how well the induced scores agreed with the ground truth labels in the training data

Loss Function

- cost function / objective
- A loss function tells how good a model is
  - high： a poor job
  - low： doing well

Examples: Loss Function

- Two commonly seen loss functions
  - **Hinge loss**
    - Multiclass Support Vector Machine (SVM) Loss
  - **Cross-entropy loss**
    - Softmax classifier

Multiclass Support Vector Machine (SVM) Loss

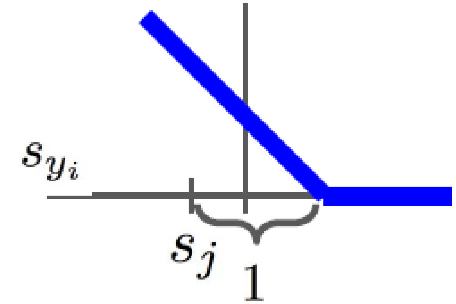
- Target：wants the correct class for each sample/data to a have a score higher than the incorrect classes by at least a margin of delta

- If any class has a score inside the red region (or higher), then there will be accumulated loss. Otherwise the loss will be zero.
- The objective will be to find the weights that will simultaneously satisfy this constraint for all the examples in the training data and give a total loss that is as low as possible.

Multiclass SVM Loss

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq y_i}\max(0, f(x_i;W)_j - f(x_i;W)_{y_i} + 1)$$

（delta=1）



cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

$$\boxed{L_i = \sum_{j\neq y_i}\max(0, s_j - s_{y_i} + 1)}$$

= max(0, 2.2 - (-3.1) + 1)
  +max(0, 2.5 - (-3.1) + 1)
= max(0, 6.3) + max(0, 6.6)
= 6.3 + 6.6
= 12.9

Loss over full dataset is :

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i$$

L=(2.9+0+12.9)/3=5.27

- Suppose that we found a W such that L = 0. Is this W unique?

- **No! e.g. 2W is also has L = 0!**

W:

= max(0, 1.3 - 4.9 + 1)
+max(0, 2.0 - 4.9 + 1)
= max(0, -2.6) + max(0, -1.9)
= 0 + 0
= 0

With W twice as large:

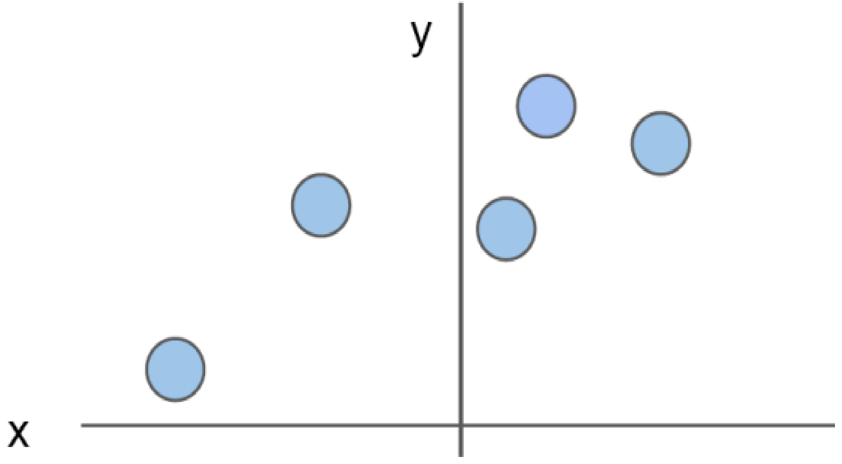
= max(0, 2.6 - 9.8 + 1)
+max(0, 4.0 - 9.8 + 1)
= max(0, -6.2) + max(0, -4.8)
= 0 + 0
= 0

- How do we choose between W and 2W?

- Regularization！

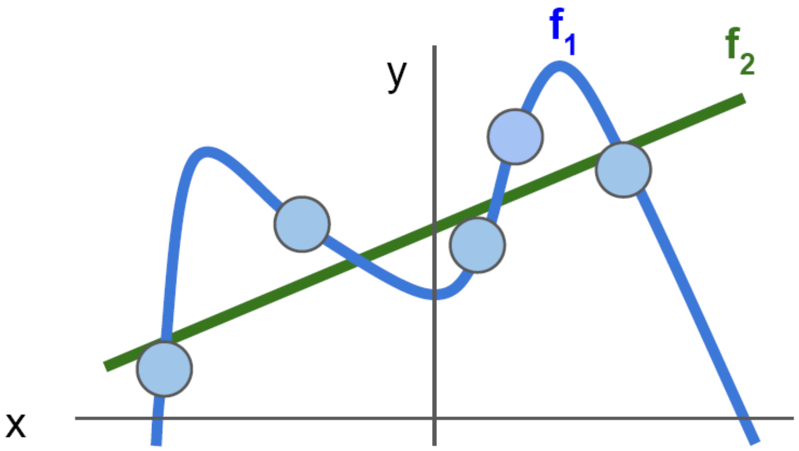
$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

- **Data loss**: Model predictions should match training data



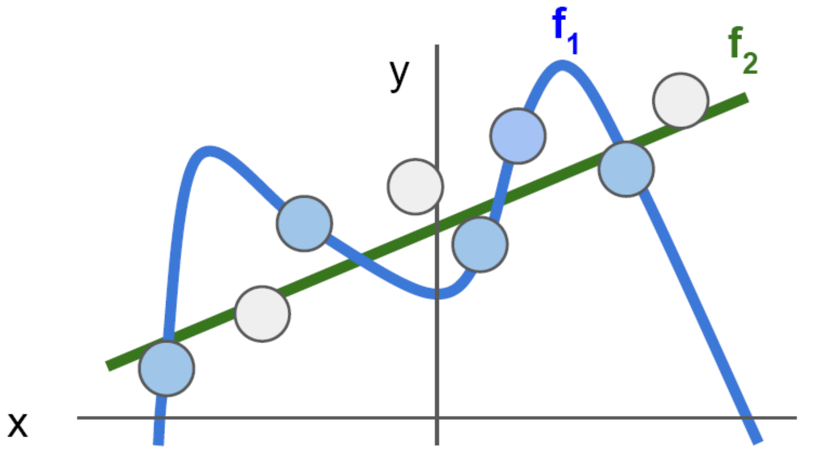
$$\boxed{L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)}$$

- **Data loss**: Model predictions should match training data



$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

- **Data loss**: Model predictions should match training data



Regularization

regularization strength
(hyperparameter)

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \boxed{\lambda R(W)}$$

Data loss: Model predictions should match training data

Regularization: Prevent the model from doing too well on training data

- Regularize
  - Express preferences over weights
  - Make the model simple so it can work on test data
  - Improve optimization by adding curvature

Regularization

- Simple examples
  - L2 regularization: $R(W) = \sum_k \sum_l W_{k,l}^2$
  - L1 regularization: $R(W) = \sum_k \sum_l |W_{k,l}|$
  - Elastic net (L1 + L2): $R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$
- More complex:
  - Dropout
  - Batch normalization
  - Stochastic depth, fractional pooling, etc

$$x = [1, 1, 1, 1]$$

$$w_1 = [1, 0, 0, 0]$$

$$w_2 = [0.25, 0.25, 0.25, 0.25]$$

$$\Rightarrow \quad w_1^T x = w_2^T x = 1$$

Use L2 Regularization $R(W) = \sum_k \sum_l W_{k,l}^2$

Which W will be chosen?

- L2 regularization prefers w2, because it likes to “spread out” the weights.

Softmax Classifier (Multinomial Logistic Regression)

- Generalization of binary Logistic Regression classifier to multiple classes

Softmax Classifier (Multinomial Logistic Regression)

- Generalization of binary Logistic Regression classifier to multiple classes

- Interpret raw classifier scores as **probabilities**

- **score:** $s = f(x_i; W)$
  - **probability:** $P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$ Softmax Function
    - **loss**: $L_i = -\log P(Y = y_i | X = x_i)$

- This can be viewed as the **cross-entropy** between the "empirical" distribution $\hat{P}(c|x_i)$and the "estimated" distribution $P_W(c|x_i)$: $-\sum_c \hat{P}(c|x_i) \log P_W(c|x_i)$

Example

Probabilities must be >= 0

Probabilities must sum to 1

$s = f(x_i; W)$

$p_i = \exp(s_i)$

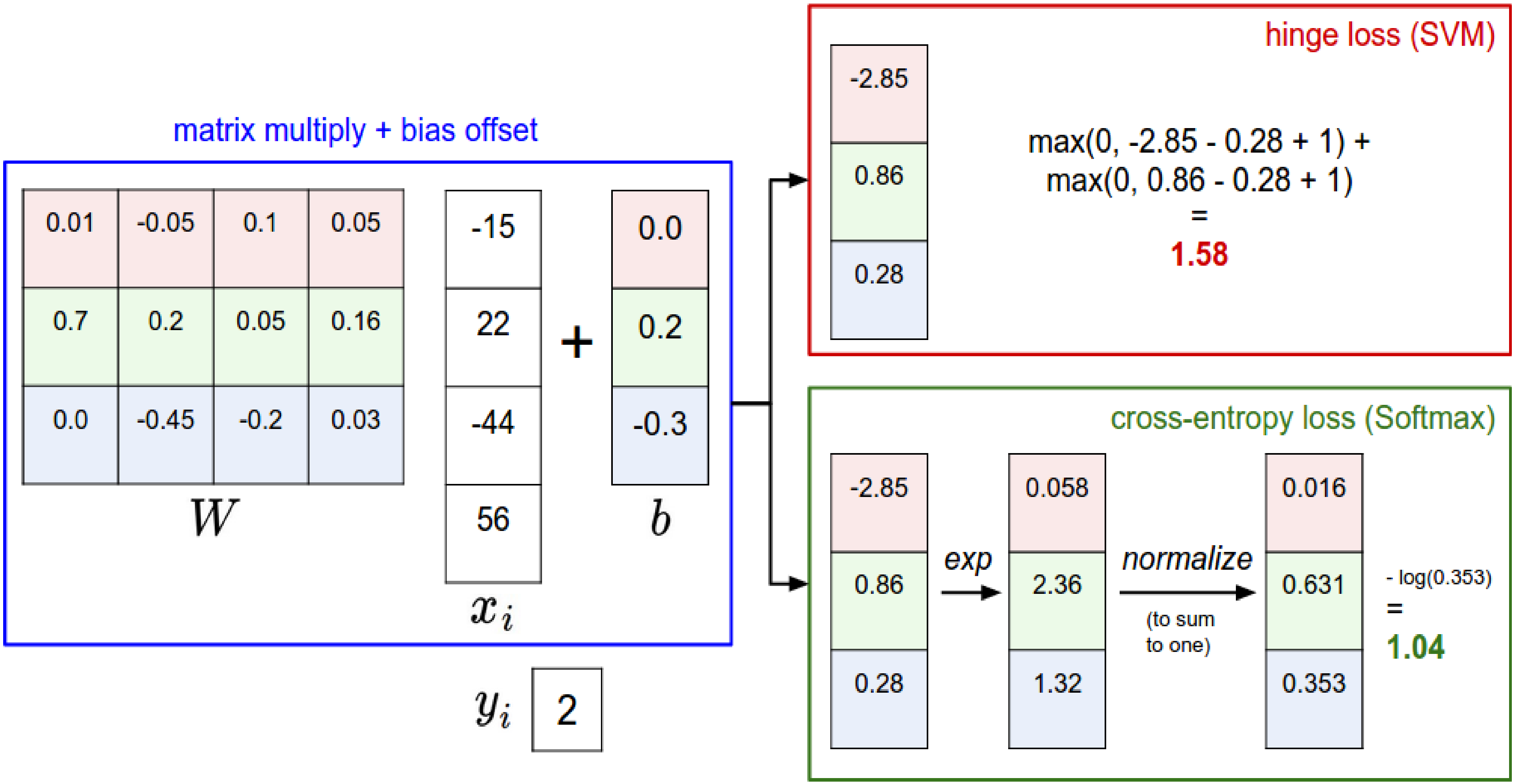
$$\frac{p_i}{\sum p_i}$$

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

	score	unnormalized probabilities	normalized probabilities
Cat	**3.2**	**24.5**	**0.13**
Car	5.1	164.0	0.87
Frog	-1.7	0.18	0.00

$\rightarrow$ Li = -log(0.13) = 0.89

SVM vs. Softmax



Summarize

- **Machine Learning 1-2-3**
  - **Collect a dataset** (and labels: for supervised learning) and **extract features**
  - **Build a model**:
    - Choose hypothesis class $\mathcal{H}$ and loss function l
  - **Optimization**:
    - Minimize the loss

Feature Extraction

- Handcraft the feature vectors (x, y)
  - Classic machine learning
  - Can use prior knowledge to design suitable features
- Learn the features directly from the raw data
  - Representation Learning
  - Deep Learning ⊆ Representation Learning ⊆ Machine Learning

课程部分材料来自他人和网络，仅限教学使用，请勿传播，谢谢！