Overview of Robot Perception

Prof. Yuke Zhu

Fall 2020

Logistics

Office Hours

Instructor: 4-5pm Wednesdays (Zoom) or by appointment

TA: 10:15-11:15am Mondays (Zoom) or by appointment

Presentation Sign-Up: Deadline Today (EOD)

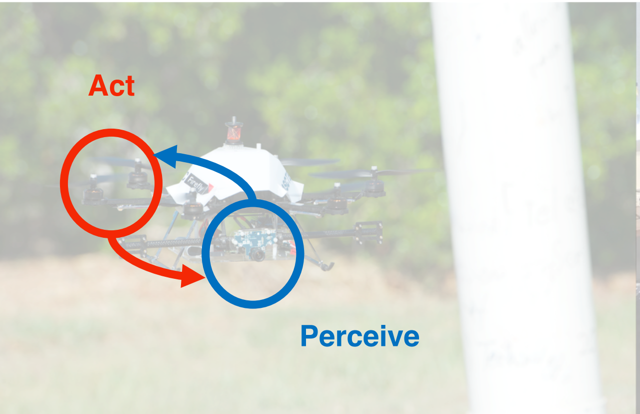
First review due: Wednesday 9:59pm (one review: Mask-RCNN or YOLO)

Student Self-Introduction

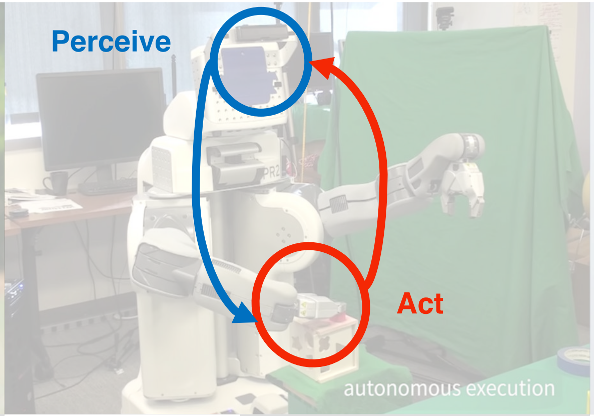
Today’s Agenda

- What is Robot Perception?
- Robot Vision vs. Computer Vision
- Landscape of Robot Perception
  - neural network architectures
  - representation learning algorithms
  - state estimation tasks
  - embodiment and active perception
- Quick Review of Deep Learning (if time permits)

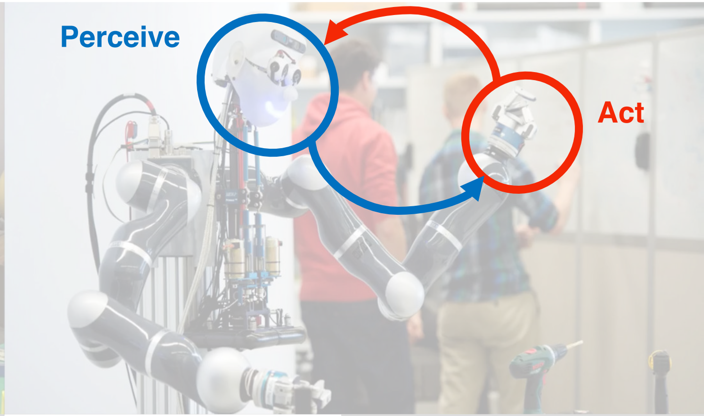
A key challenge in Robot Learning is to close the **perception**-action loop.



[Sa et al. IROS 2014]



[Levine et al. JMLR 2016]



[Bohg et al. ICRA 2018]

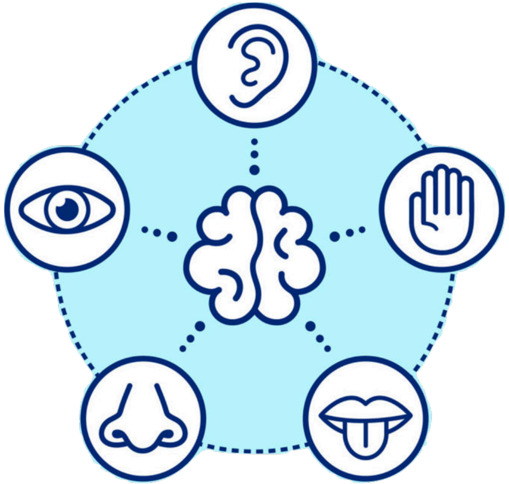
What is Robot Perception?

Making sense of the unstructured real world…

- Incomplete knowledge of objects and scene
- Imperfect actions may lead to failure
- Environment dynamics and other agents

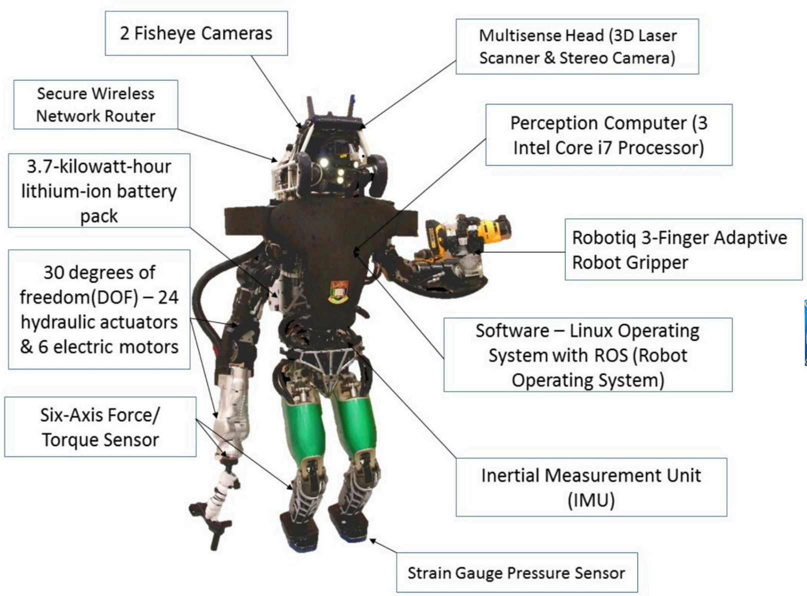
Robotic Sensors

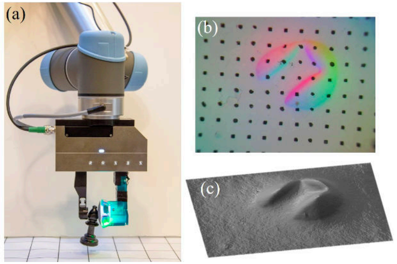
Making contact of the physical world through multimodal senses

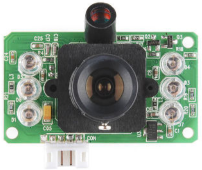


Robotic Sensors

Making contact of the physical world through multimodal senses







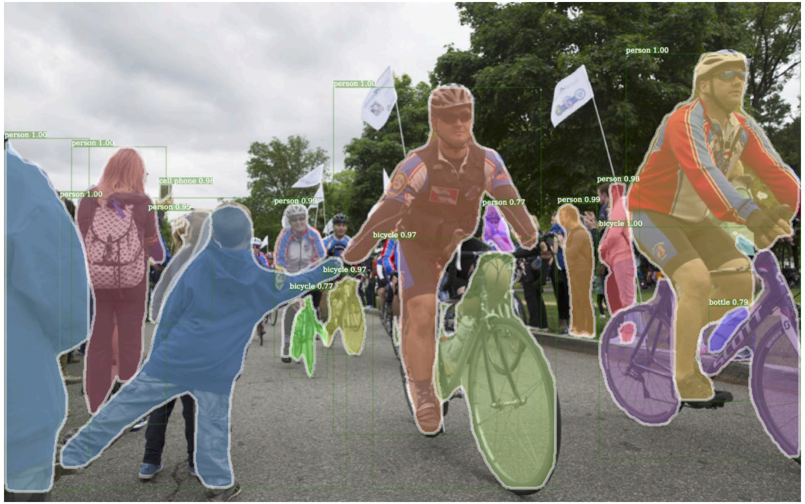
[Source: HKU Advanced Robotics Laboratory]

- The Limits and Potentials of Deep Learning for Robotics. Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, Peter Corke (2018)
  - A Sensorimotor Account of Vision and Visual Consciousness. Kevin O'Regan and Alva Noë (2001)

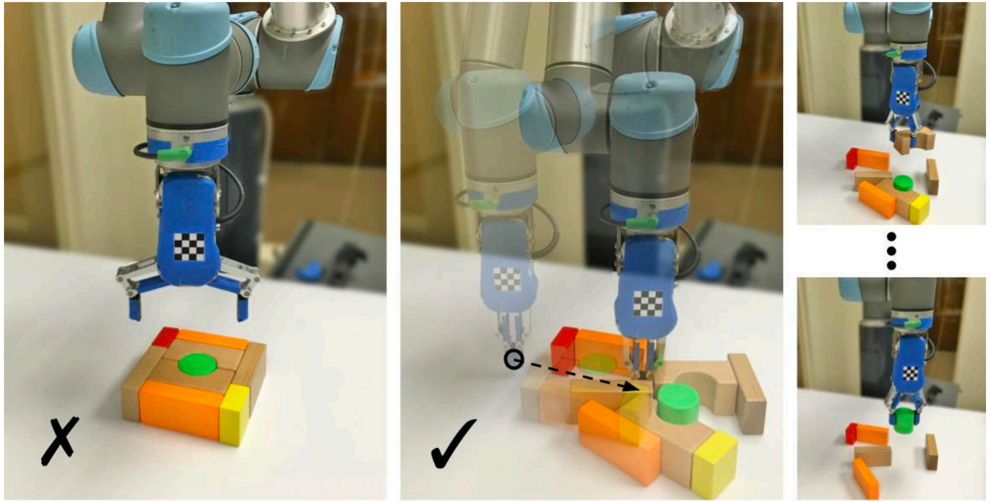
Robot Vision vs. Computer Vision

MIND THE GAP

Robot vision is **embodied**, **active**, and **environmentally situated**.



[Detectron - Facebook AI Research]



[Zeng et al., IROS 2018]

Robot Vision vs. Computer Vision

Robot vision is **embodied**, **active**, and **environmentally situated**.

- **Embodied**: Robots have physical bodies and experience the world directly. Their actions are part of a dynamic with the world and have immediate feedback on their own sensation.
- **Active**: Robots are active perceivers. It knows why it wishes to sense, and chooses what to perceive, and determines how, when and where to achieve that perception.
- **Situated**: Robots are situated in the world. They do not deal with abstract descriptions, but with the here and now of the world directly influencing the behavior of the system.

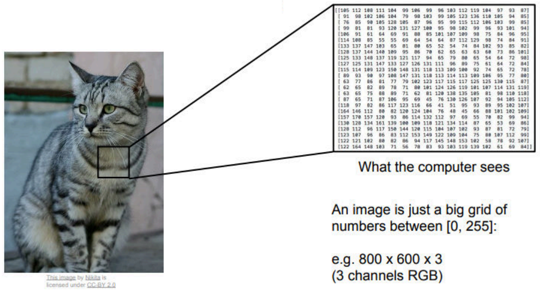
[Brooks 1991; Bajcsy 2018]

Robot Perception: Landscape

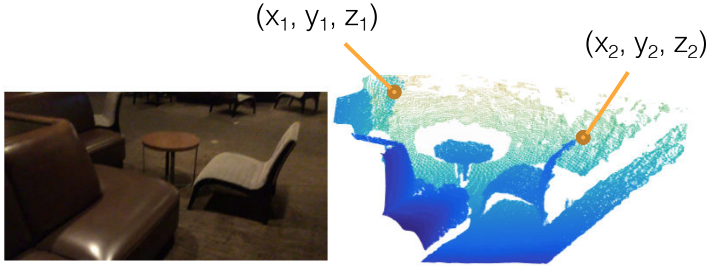
What you will learn in the chapter of Robotics and Perception

1. **Modalities**: neural network architectures designed for different sensory modalities
2. **Representations**: representation learning algorithms without strong supervision
3. **Tasks**: state estimation tasks for robot navigation and manipulation
4. **Embodiment**: active perception for embodied visual intelligence

Robot Perception: Modalities

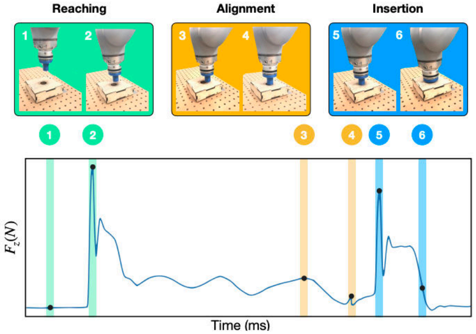


Pixels (from RGB cameras)



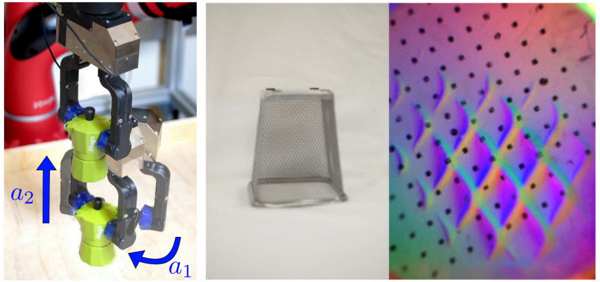
[Source: PointNet++; Qi et al. 2016]

Point cloud (from structure sensors)



[Source: Lee*, Zhu*, et al. 2018]

Time series (from F/T sensors)

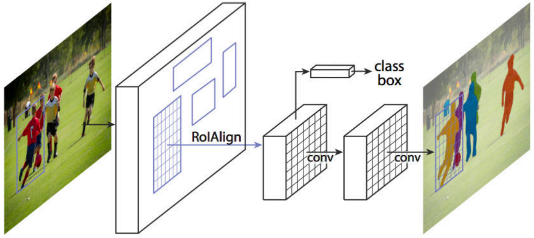


[Source: Calandra et al. 2018]

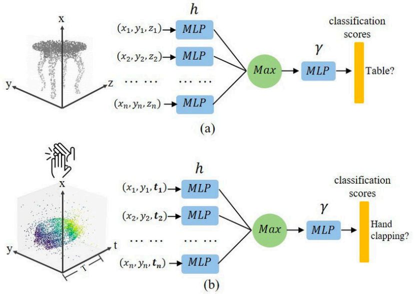
Tactile data (from the GelSights sensors)

Robot Perception: Modalities

How can we design the **neural network architectures** that can effectively process raw sensory data in vastly different forms?



Week 2: Object Detection (Pixels)

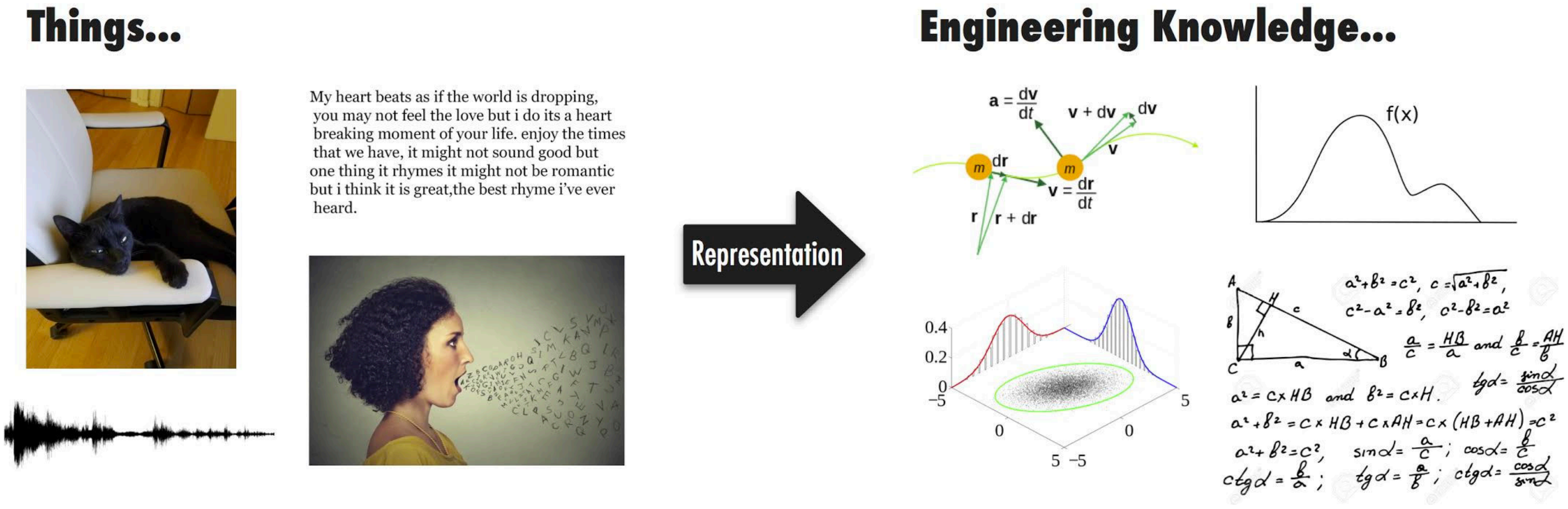


Week 3: 3D Point Cloud

More sensory modalities in later weeks…

Robot Perception: Representations

A fundamental problem in robot perception is to learn the proper **representations** of the unstructured world.



[Source: Stanford CS331b]

Robot Perception: Representations

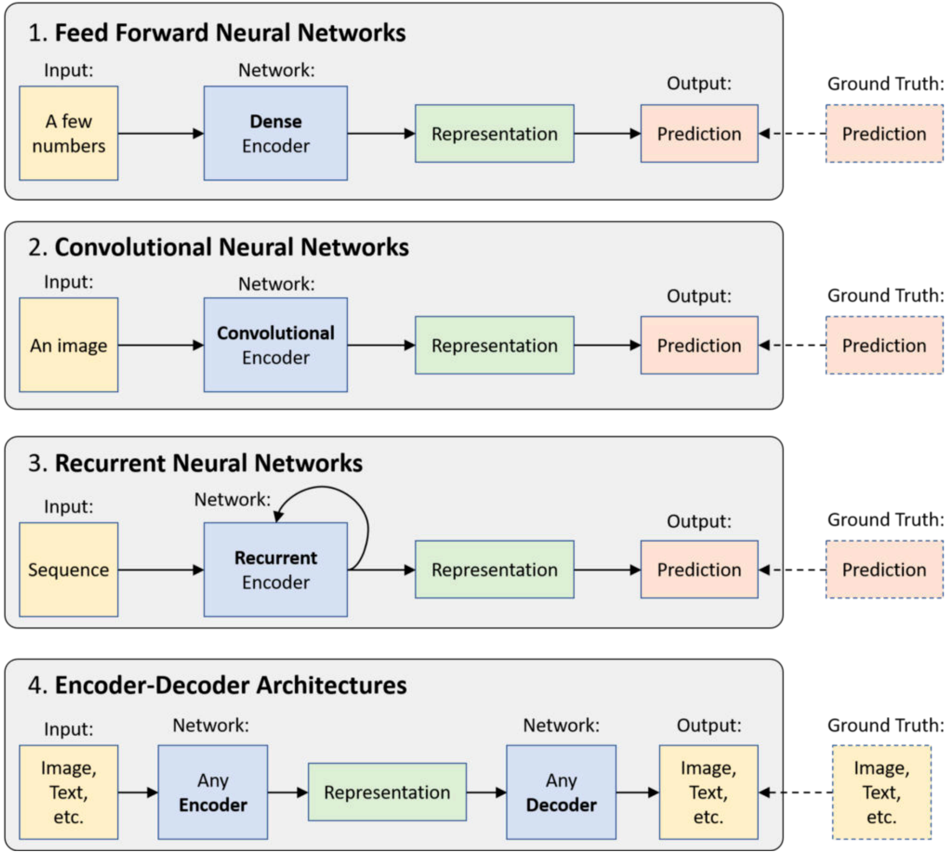
"Solving a problem simply means representing it so as to make the solution transparent."

Herbert A. Simon, Sciences of the Artificial

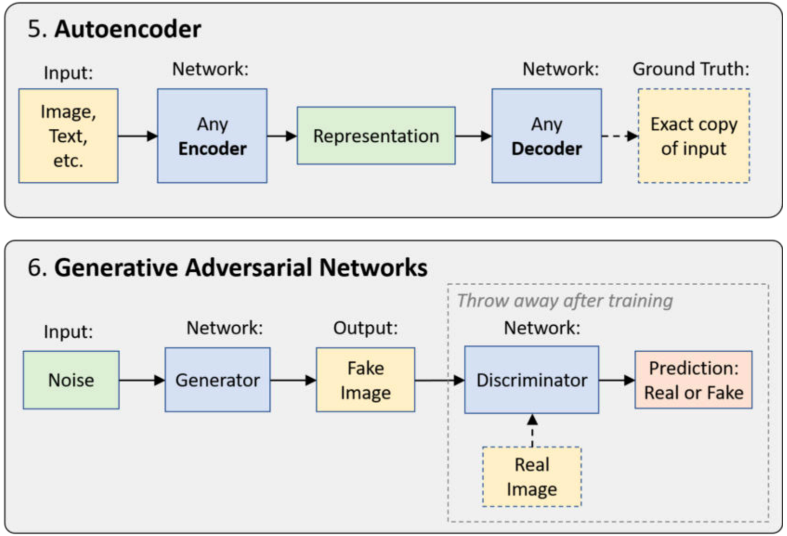
Our secret weapon? **Learning**

ICLR 2020
8th International Conference on Learning Representations
Addis Ababa, Ethiopia
April 26-30, 2020

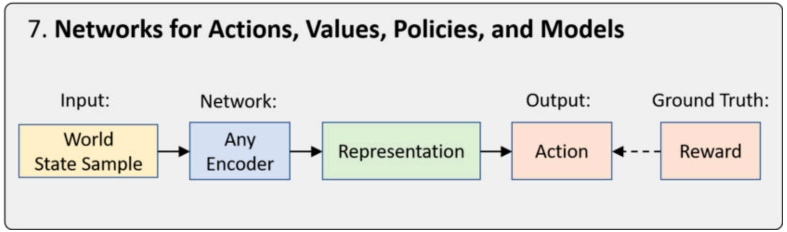
Supervised Learning



Unsupervised Learning



Reinforcement Learning



[6.S094, MIT]

Robot Perception: Representations

How can we learn **representations of the world** with limited supervision?

Week 3 (Thu)

"Nature" Structural priors (inductive biases)

+

"Nurture" Interaction and movement (embodiment)

Week 4 (Tue)

babies learning by playing

Robot Perception: Representations

How can we learn representations that fuse **multiple sensory modalities** together?

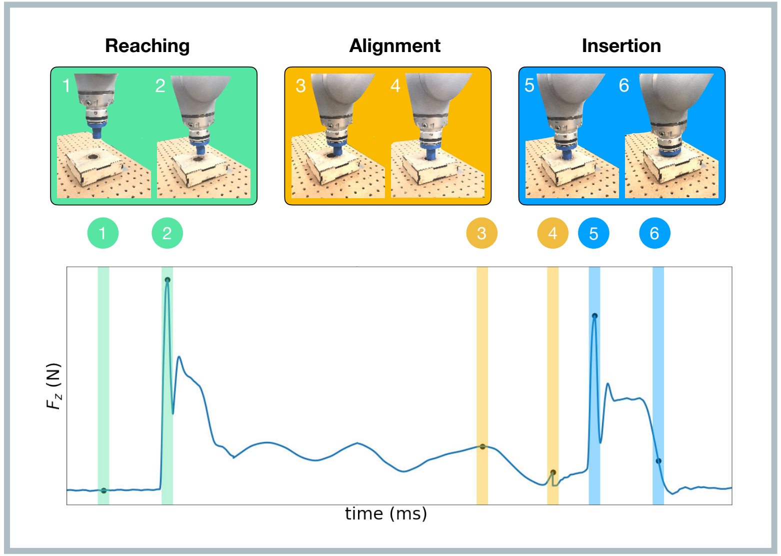
Is seeing believing?

[The McGurk Effect, BBC]

https://www.youtube.com/watch?v=2k8fHR9jKVM

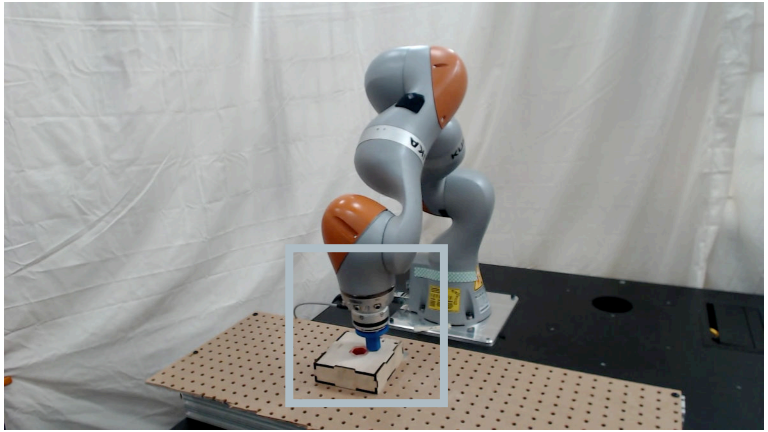
Robot Perception: Representations

How can we learn representations that fuse **multiple sensory modalities** together?



combining vision and force for manipulation

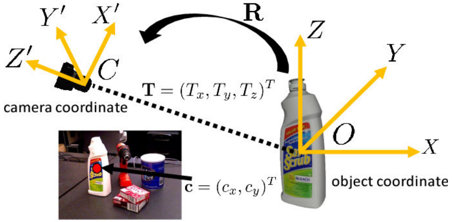
Week 4 Thu: Multimodal Sensor Fusion



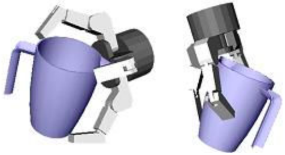
[Lee*, Zhu*, et al. 2018]

Robot Perception: Tasks

Noisy Sensory Data



State Representation

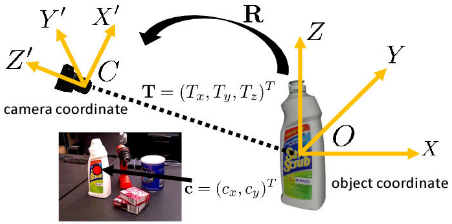


Perception & Computer Vision

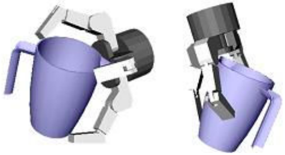
Robot Control & Decision Making

Robot Perception: Tasks

Noisy Sensory Data



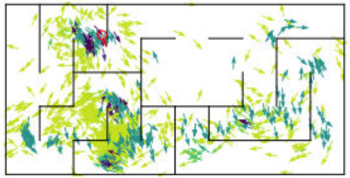
State Representation



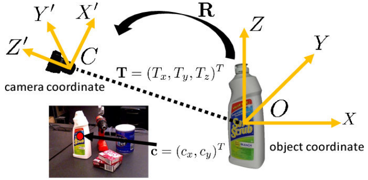
Perception & Computer Vision

Robot Control & Decision Making

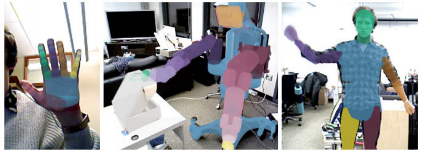
Localization (Week 5 Tue)



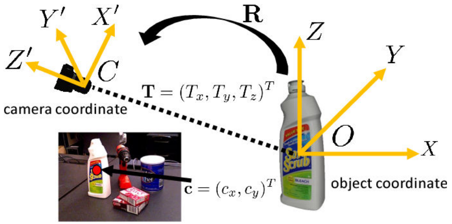
Pose Estimation (Week 5 Thu)



Visual Tracking (Week 6 Tue)

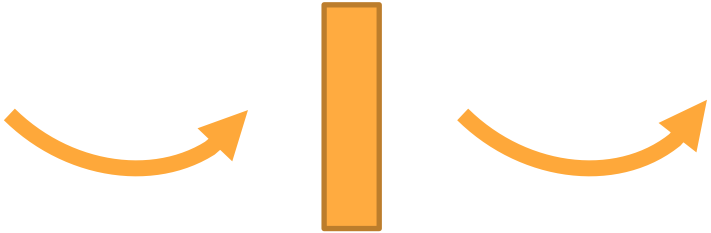


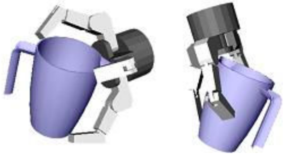
Robot Perception: Tasks



State Representation

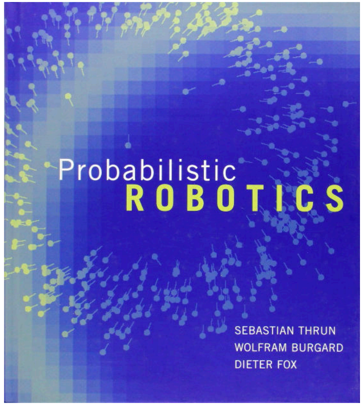
Noisy Sensory Data





Perception & Computer Vision

Robot Control & Decision Making



http://www.probabilistic-robotics.org/

Robot Perception: Tasks

State estimation methods: **Bayes Filtering**

Algorithm 1 The general algorithm for Bayes filtering

1: **for each** x_t **do**
2: $\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1}$ ▷ transition update
3: $bel(x_t) = \eta\, p(z_t|x_t)\, \overline{bel}(x_t)$ ▷ measurement update
4: **end for each**

x_t: state z_t: observation u_t: action $bel(x_t)$: belief

$p(x_t|u_t, x_{t-1})$: transition model (motion model)

$p(z_t|x_t)$: measurement model (observation model)

Robot Perception: Tasks

Week 5
Tue, Sept 22

Recursive State Estimation

- Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Rico Jonschkowski, Divyam Rastogi, Oliver Brock (2018)
- Particle Filter Networks with Application to Visual Localization. Peter Karkus, David Hsu, Wee Sun Lee (2018)
  - Differentiable Algorithm Networks for Composable Robot Learning. Peter Karkus, Xiao Ma, David Hsu, Leslie Pack Kaelbling, Wee Sun Lee, Tomas Lozano-Perez (2019)
  - Backprop KF: Learning Discriminative Deterministic State Estimators. Tuomas Haarnoja, Anurag Ajay, Sergey Levine, Pieter Abbeel (2016)

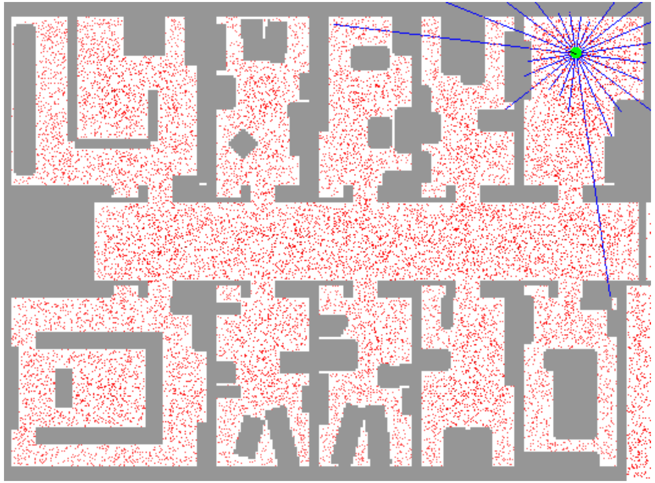
State estimation methods: **Bayes Filtering**

x_t: state z_t: observation u_t: action $bel(x_t)$: belief

$p(x_t | u_t, x_{t-1})$: transition model (motion model)

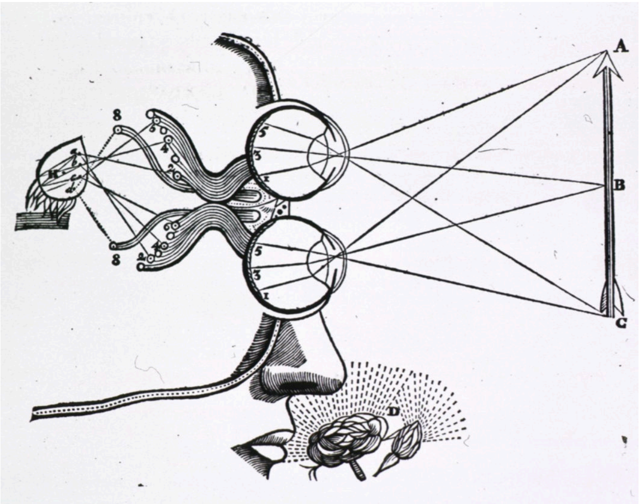
$p(z_t | x_t)$: measurement model (observation model)

What if models are hard to specify? **Learning**



Example: Particle Filter Localization

Robot Perception: Embodiment



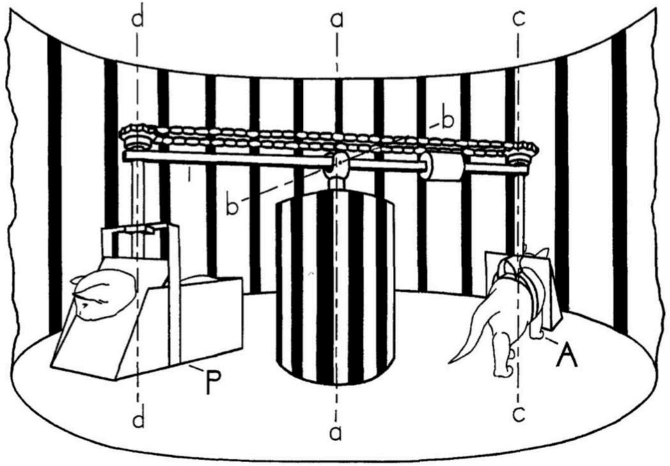
Input-Output Picture (Susan Hurley, 1998)

Conventional View of Perception

- Perception is the process of building an internal representation of the environment
- Perception is input from world to mind, and action is output from mind to world, thought is the mediating process.

[Action in Perception, Alva Noë 2004]

Robot Perception: Embodiment



Kitten Carousel (Held and Hein, 1963)

Embodied View of Perception

- As the active cat (A) walks, the other cat (P) moves and perceives the environment passively.
- Only the active cat develops normal perception through *self-actuated* movement.
- The passive cat suffers from perception problems, such as 1) not blinking when objects approach, and 2) hitting the walls.

Robot Perception: Embodiment

Pebbles (James J. Gibson 1966)

Embodied View of Perception

- Subjects asked to find a reference object among a set of irregularly-shaped objects
- Three groups
    a. Passive observers of one static image (49%)
    b. Observers of moving shapes (72%)
    c. Interactive observers (99%)
- The ability to condition input signals with actions is crucial to perception.

Robot Perception: Embodiment

Take-home messages

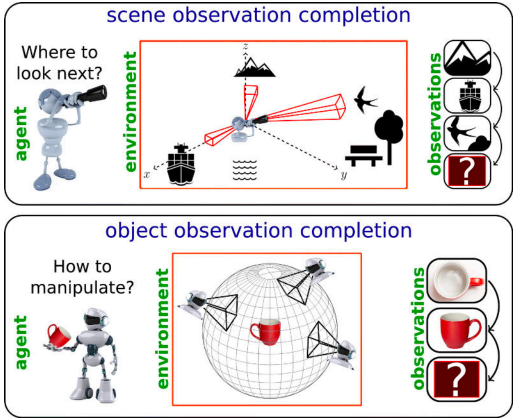
- Perceptual experiences do not present the sense in the way that a photograph does.
- Perception is developed by an embodied agent through actively exploring in the physical world.

- “We see in order to move; we move in order to see.” – William Gibson

Robot Perception: Embodiment

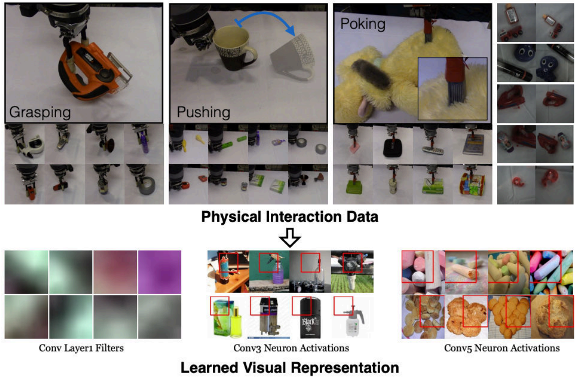
Week 6 (Thu) – Active Perception: How can embodied agents (robots) improve perception based on visual experiences through active exploration?

View Selection



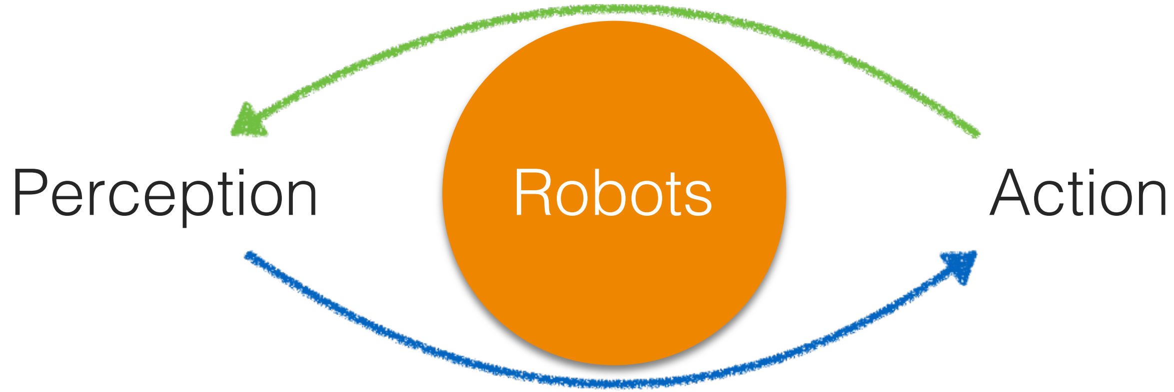
[Ramakrishnan et al. 2019]

Physical Interaction



[Pinto et al. 2016]

Research Frontier: Closing the Perception-Action Loop

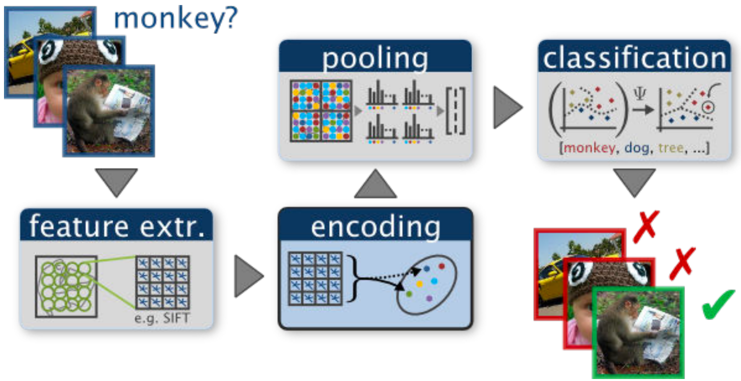


How robots develop better perception from embodied sensorimotor experiences

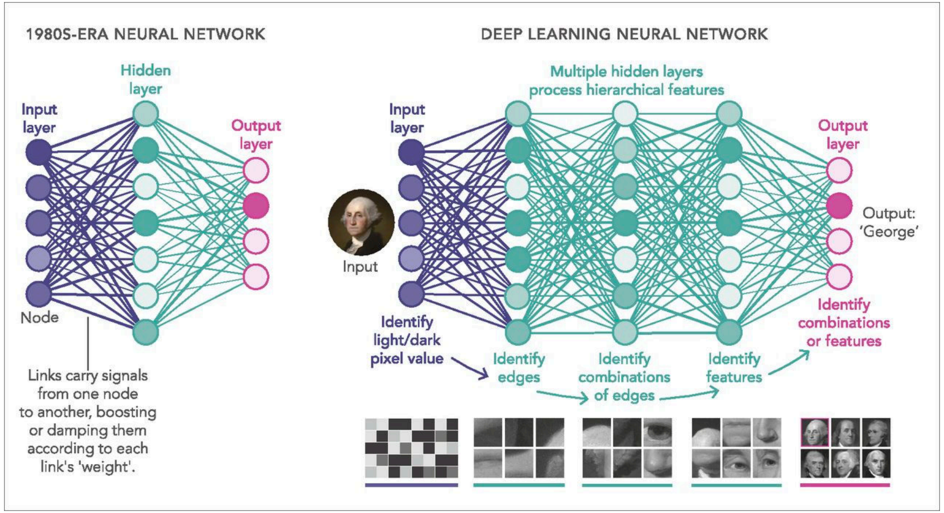
How robots' intelligent behaviors are guided by their interactive perception

Visual Processing Methods

What is new since 1980s?

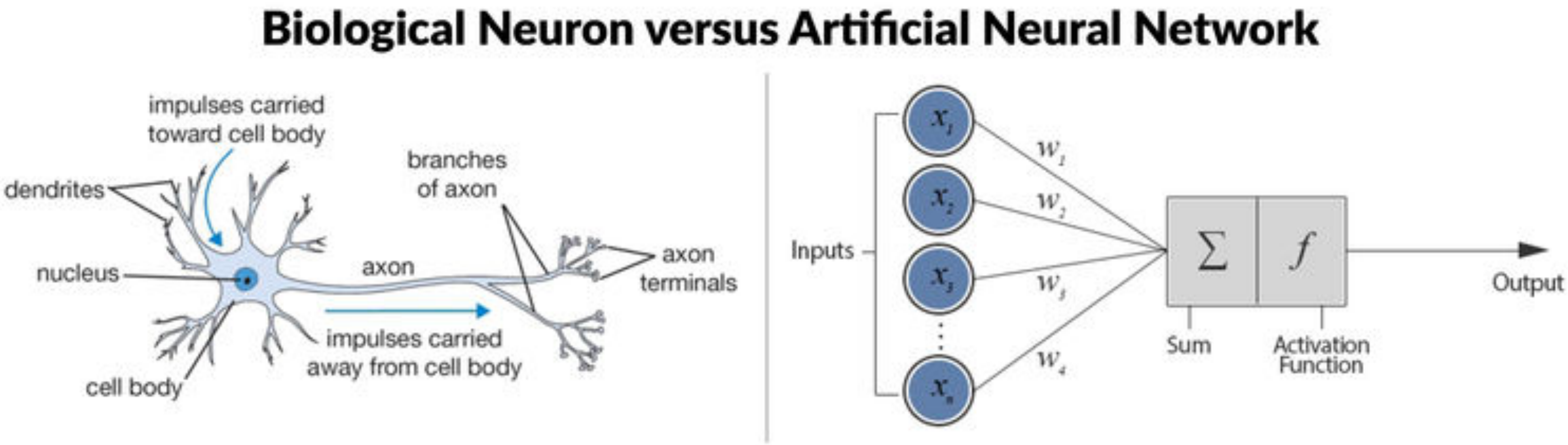


Staged Visual Recognition Pipeline



End-to-end Deep Learning

Quick Review of Deep Learning: Artificial Neurons



Biological Neuron

Computational building block for the brain

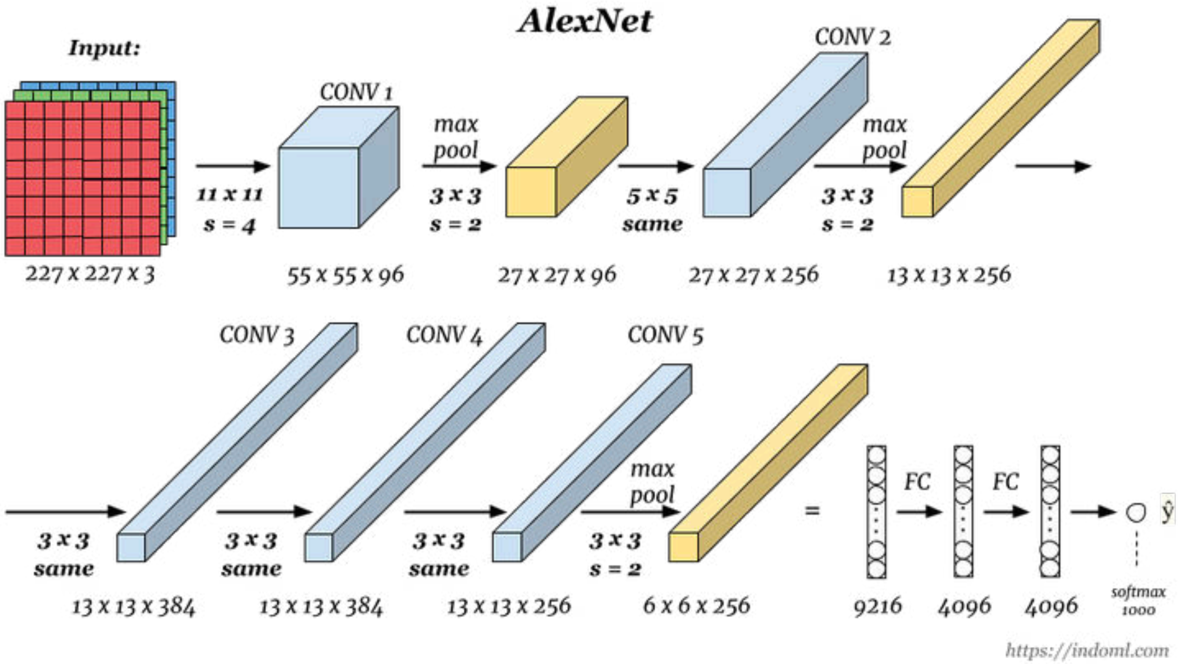
Artificial Neuron

Computational building block for the neural network

Note: Many differences exist – be careful with the brain analogies!

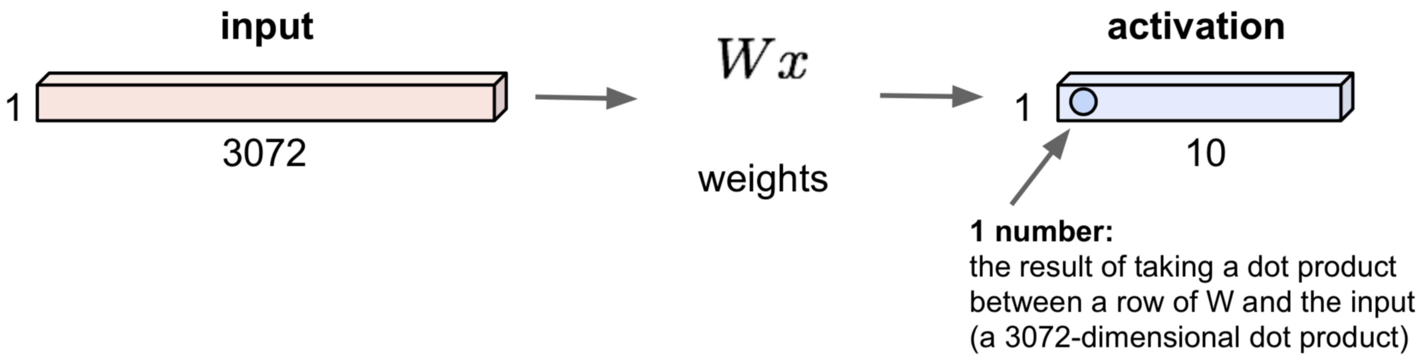
[Dendritic Computation, Michael London and Michael Hausser 2015]

Quick Review of Deep Learning: Convolutional Networks



Quick Review of Deep Learning: Fully-Connected Layers

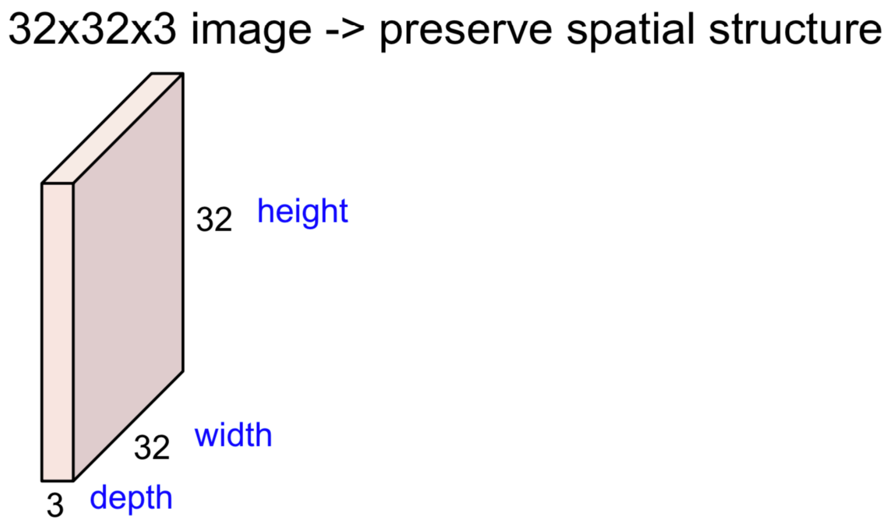
32x32x3 image -> stretch to 3072 x 1



What is the dimension of W?

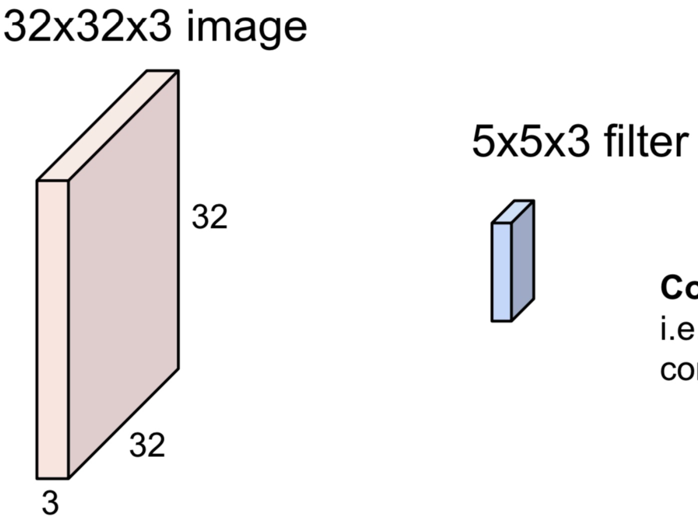
[Source: Stanford CS231N]

Quick Review of Deep Learning: Convolutional Layers



[Source: Stanford CS231N]

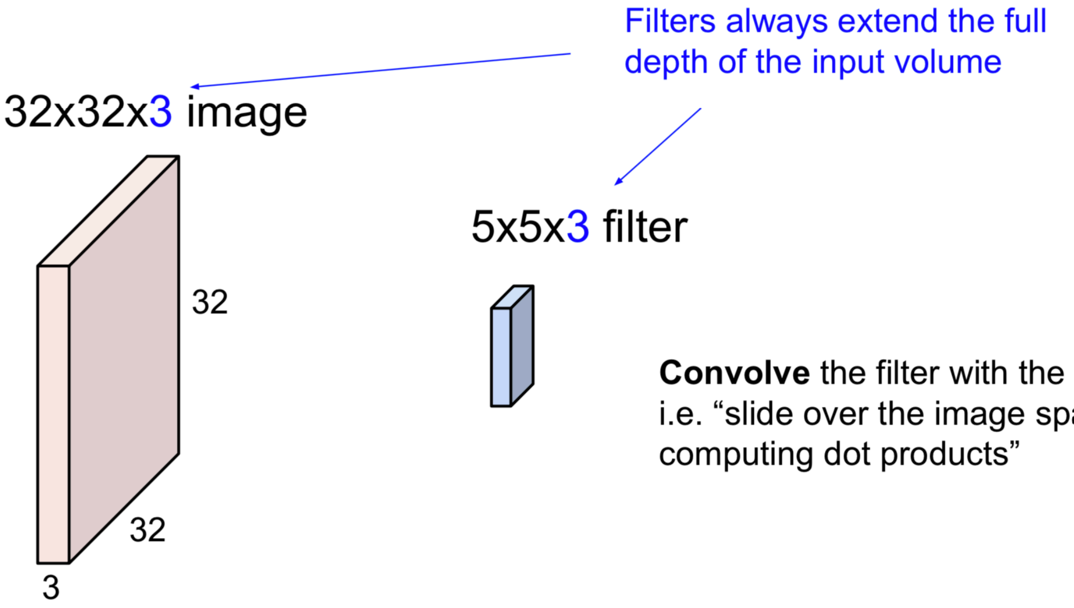
Quick Review of Deep Learning: Convolutional Layers



Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

[Source: Stanford CS231N]

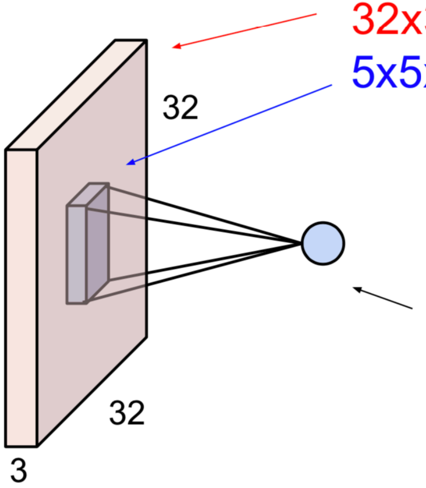
Quick Review of Deep Learning: Convolutional Layers



Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

[Source: Stanford CS231N]

Quick Review of Deep Learning: Convolutional Layers

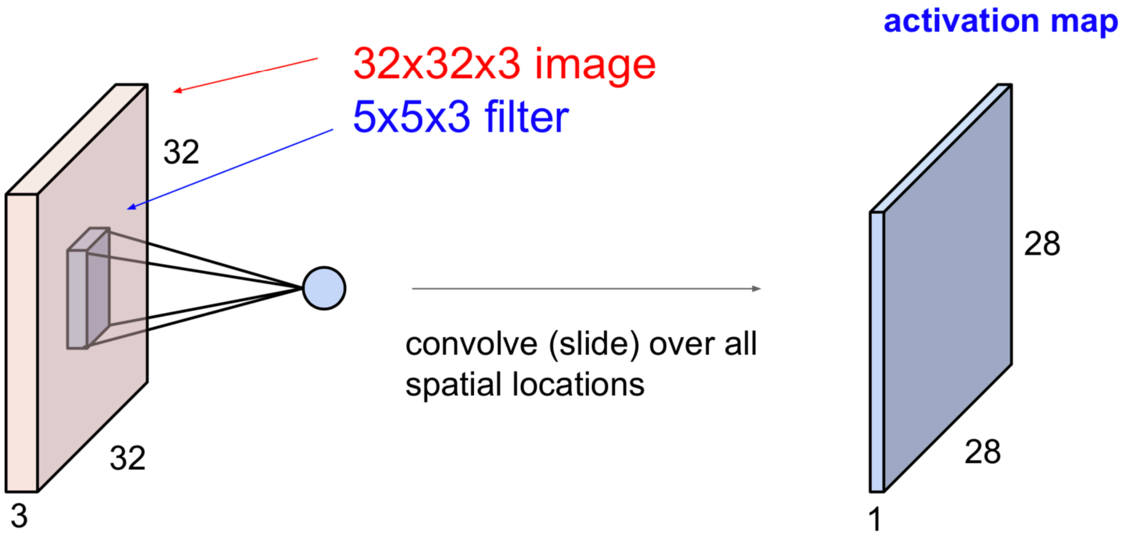


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

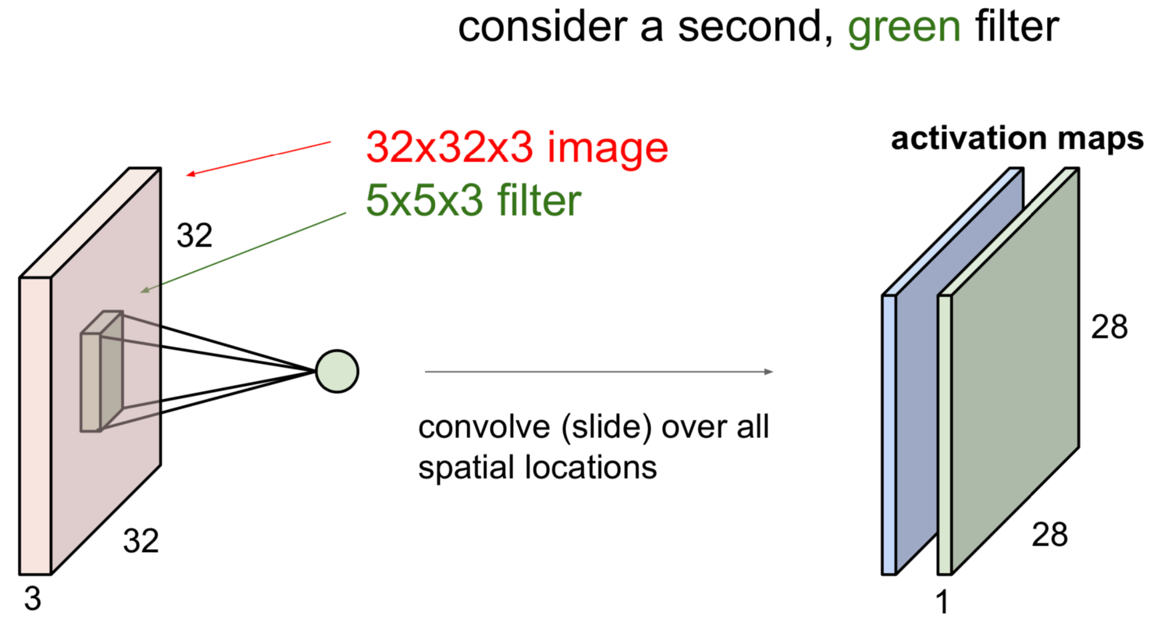
[Source: Stanford CS231N]

Quick Review of Deep Learning: Convolutional Layers



[Source: Stanford CS231N]

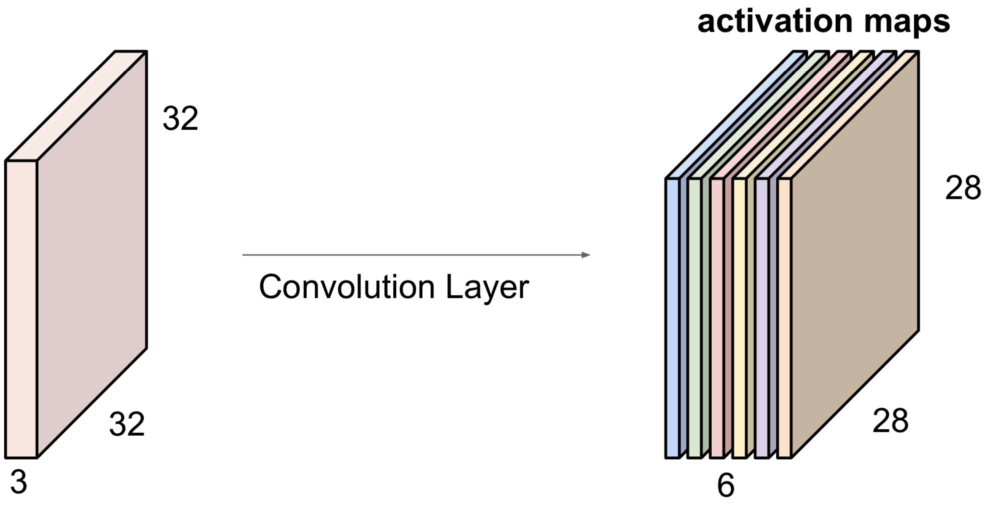
Quick Review of Deep Learning: Convolutional Layers



[Source: Stanford CS231N]

Quick Review of Deep Learning: Convolutional Layers

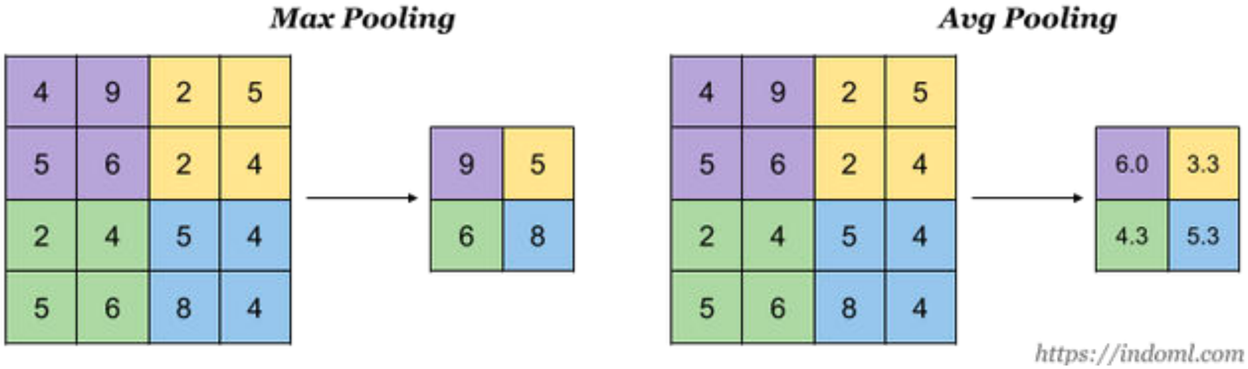
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

[Source: Stanford CS231N]

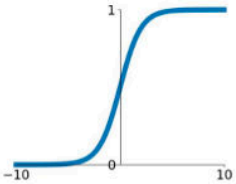
Quick Review of Deep Learning: Pooling Operations



Quick Review of Deep Learning: Activation Functions

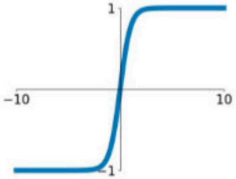
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



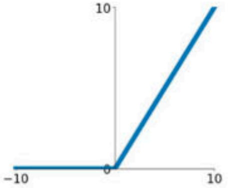
tanh

$\tanh(x)$



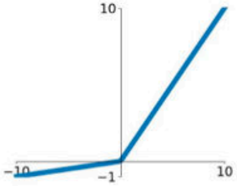
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

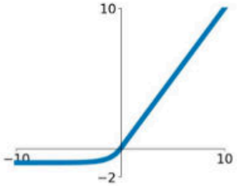


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

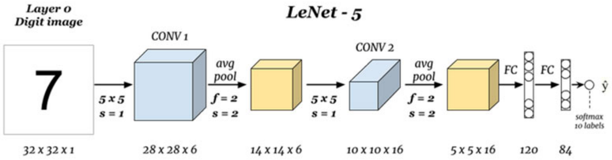
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

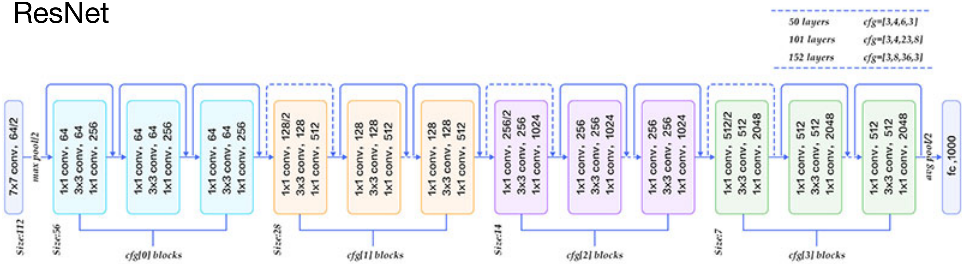


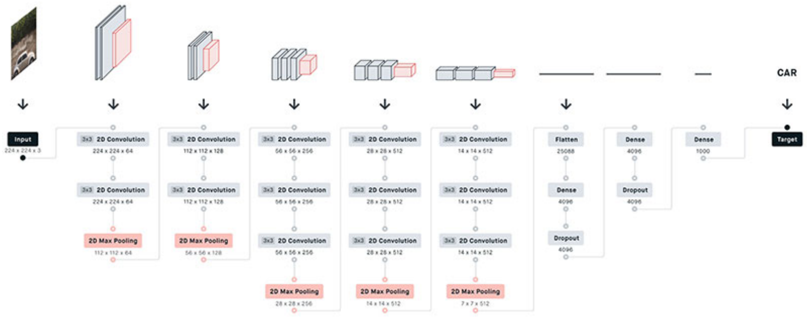
Quick Review of Deep Learning: CNN Architectures

LeNet

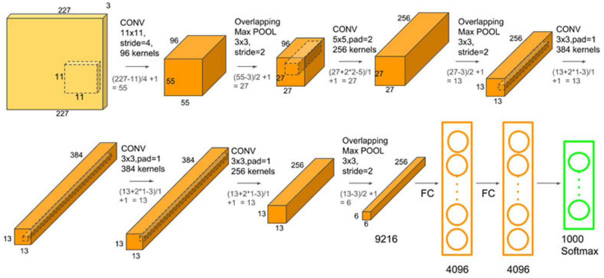


ResNet





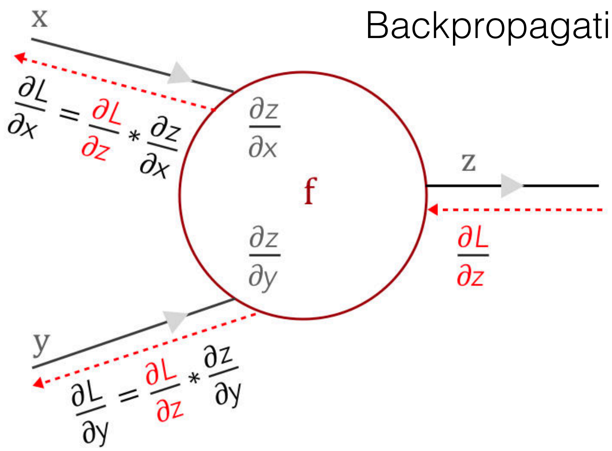
VGG-16



AlexNet

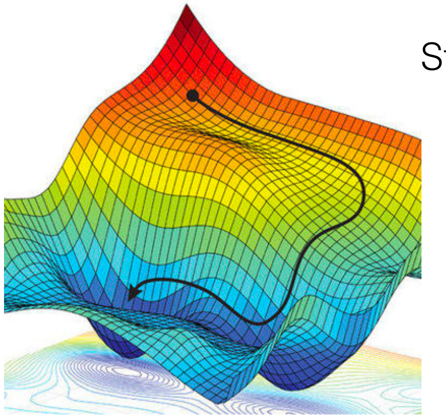
Quick Review of Deep Learning: Optimization

Backpropagation



$\frac{\partial z}{\partial x}$ & $\frac{\partial z}{\partial y}$ *are local gradients*

$\frac{\partial L}{\partial z}$ *is the loss from the previous layer which has to be backpropagated to other layers*



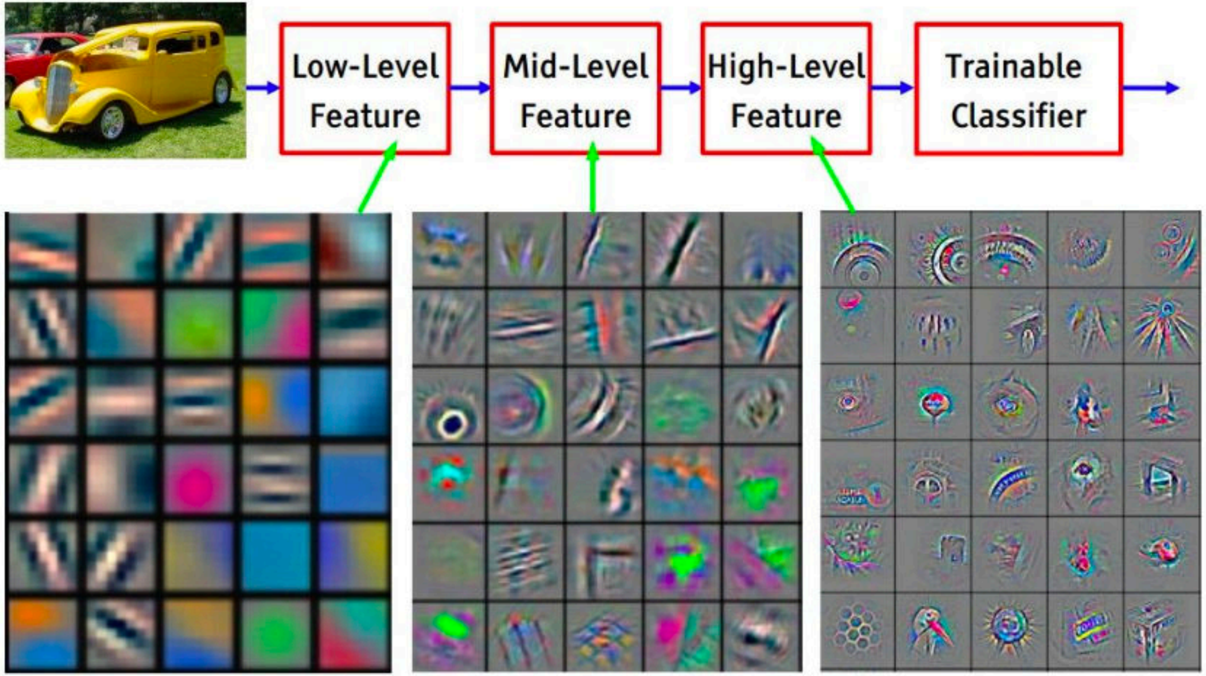
Stochastic Gradient Descent (SGD)

learning rate

$$\theta = \theta - \eta \nabla_\theta J(\theta; x^{(i)}; y^{(i)})$$

weights input label

Quick Review of Deep Learning: Features



[Source: Stanford CS231N]

Quick Review of Deep Learning: Implementation

PyTorch

TensorFlow

Tutorial coming in late September / early October

```
[ ] import torch
    from torch import nn

    class MNISTClassifier(nn.Module):

      def __init__(self):
        super(MNISTClassifier, self).__init__()

        # mnist images are (1, 28, 28) (channels, width, heig
        self.layer_1 = torch.nn.Linear(28 * 28, 128)
        self.layer_2 = torch.nn.Linear(128, 256)
        self.layer_3 = torch.nn.Linear(256, 10)

      def forward(self, x):
        batch_size, channels, width, height = x.size()

        # (b, 1, 28, 28) -> (b, 1*28*28)
        x = x.view(batch_size, -1)

        # layer 1
        x = self.layer_1(x)
        x = torch.relu(x)

        # layer 2
        x = self.layer_2(x)
        x = torch.relu(x)

        # layer 3
        x = self.layer_3(x)

        # probability distribution over labels
        x = torch.log_softmax(x, dim=1)

        return x
```

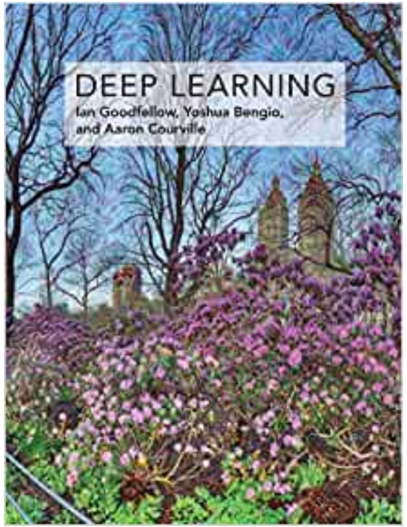
Quick Review of Deep Learning: Resources

Online Courses

- CS231N: Convolutional Neural Networks for Visual Recognition
  http://cs231n.stanford.edu/
- MIT 6.S191: Introduction to Deep Learning
  http://introtodeeplearning.com/

Textbooks:

- Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville
  http://www.deeplearningbook.org/



Resources

Related courses at UTCS

- CS342: Neural Networks
- CS 376: Computer Vision
- CS 378 Autonomous Driving
- CS 393R: Autonomous Robots
- CS394R: Reinforcement Learning: Theory and Practice

Extended readings:

- Action-based Theories of Perception, Stanford Encyclopedia of Philosophy
- Action in Perception, Alva Noë