CSCI 599: Deep Learning and its Applications

Lecture 3

Fall 2017
Joseph J. Lim

Disclaimer

- This course is taught for the 1st time @ USC. This course is 599, and thus an **experimental** course.
- The syllabus, course policy, and grading details **may change** over the semester (**check website!**)
- If you prefer a well-structured course, this is **NOT** a course for you, and I encourage you to take the course next year. We really mean this.
- But, it will be **fun** and **challenging**!

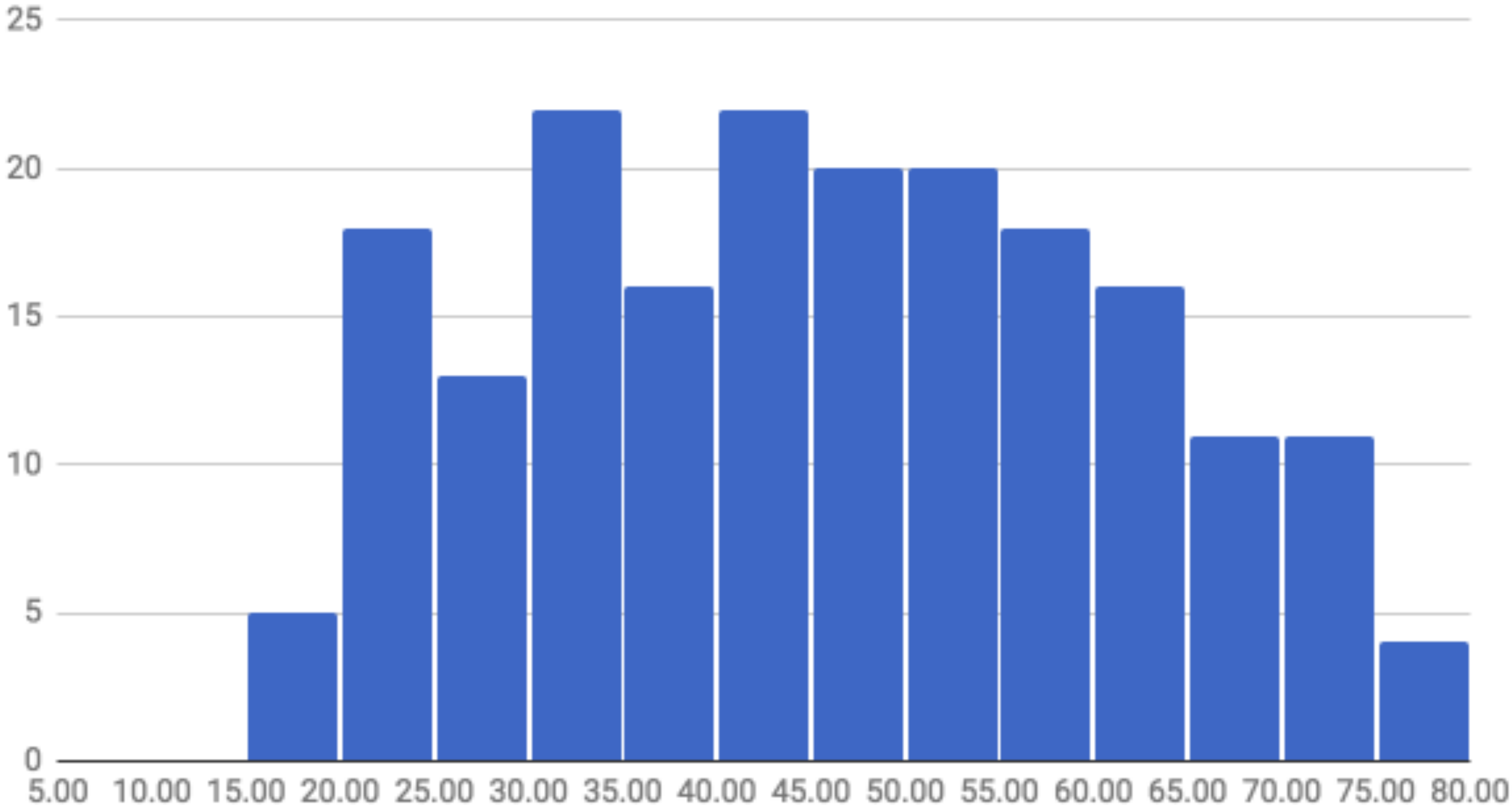
Today's agenda

- CSCI 599 overview
- Learning 101
- Course Entrance 1-1

Today's agenda

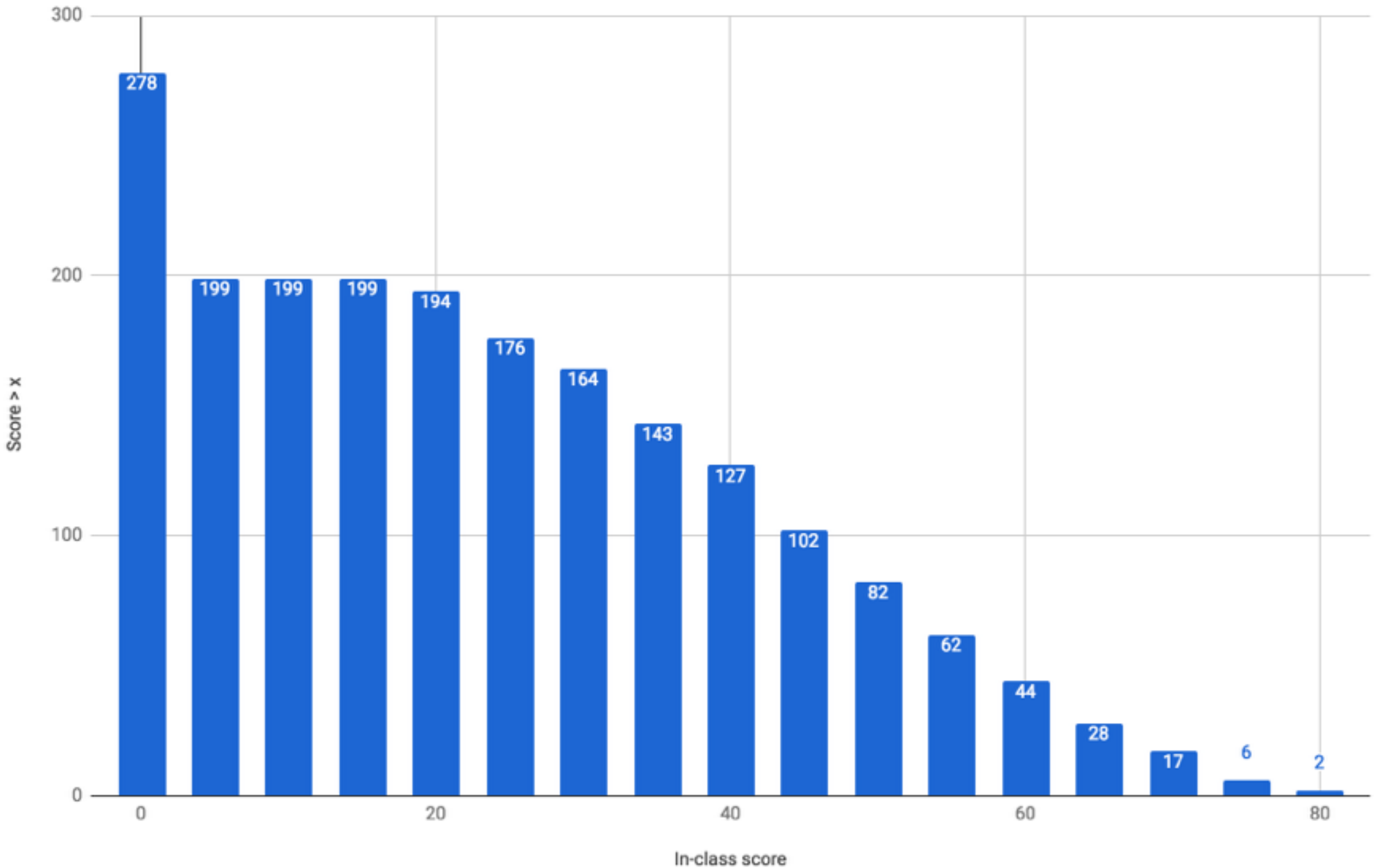
- CSCI 599 overview
- Learning 101
- Course Entrance 1-1

Entrance Exam



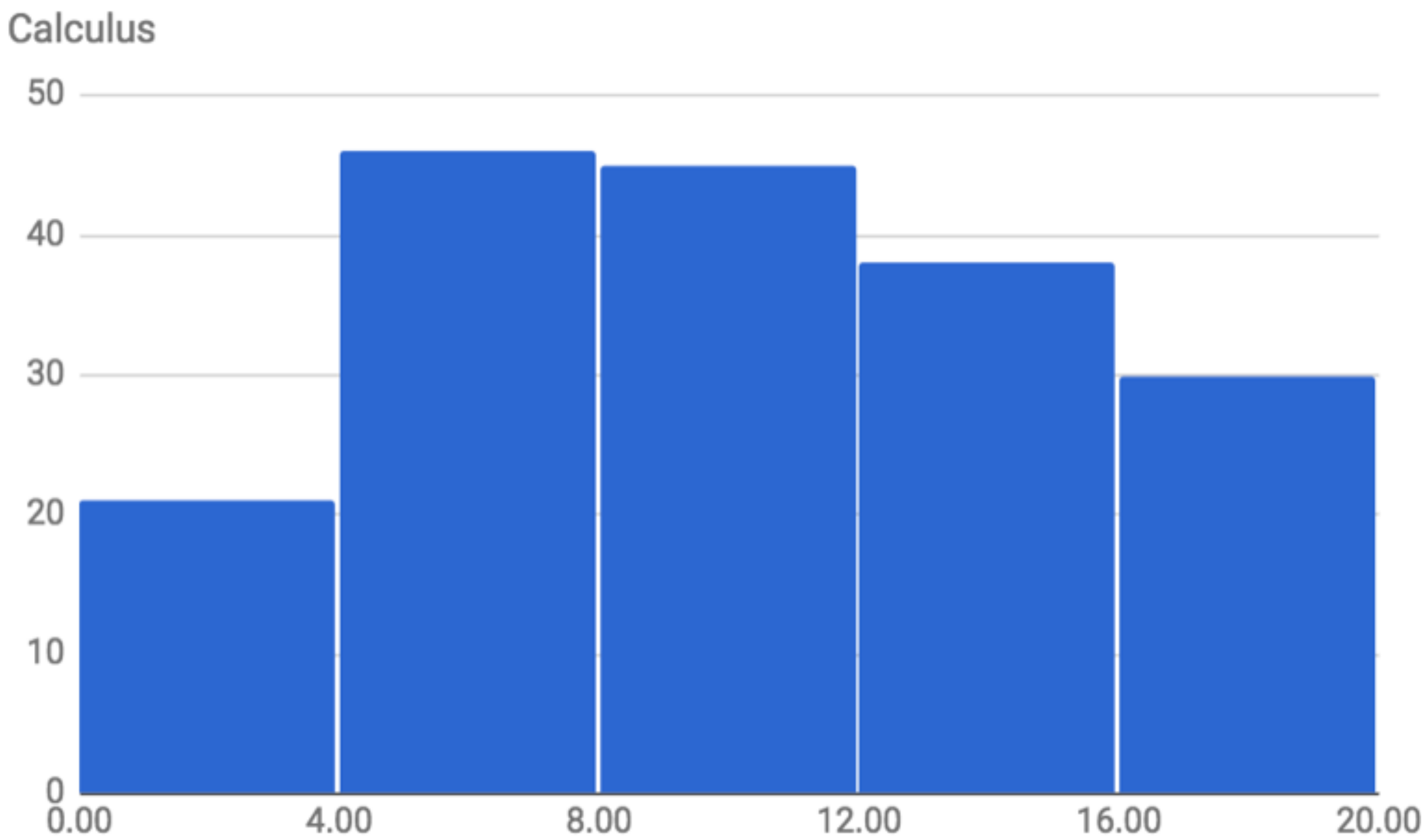
In-class Exam

Entrance Exam



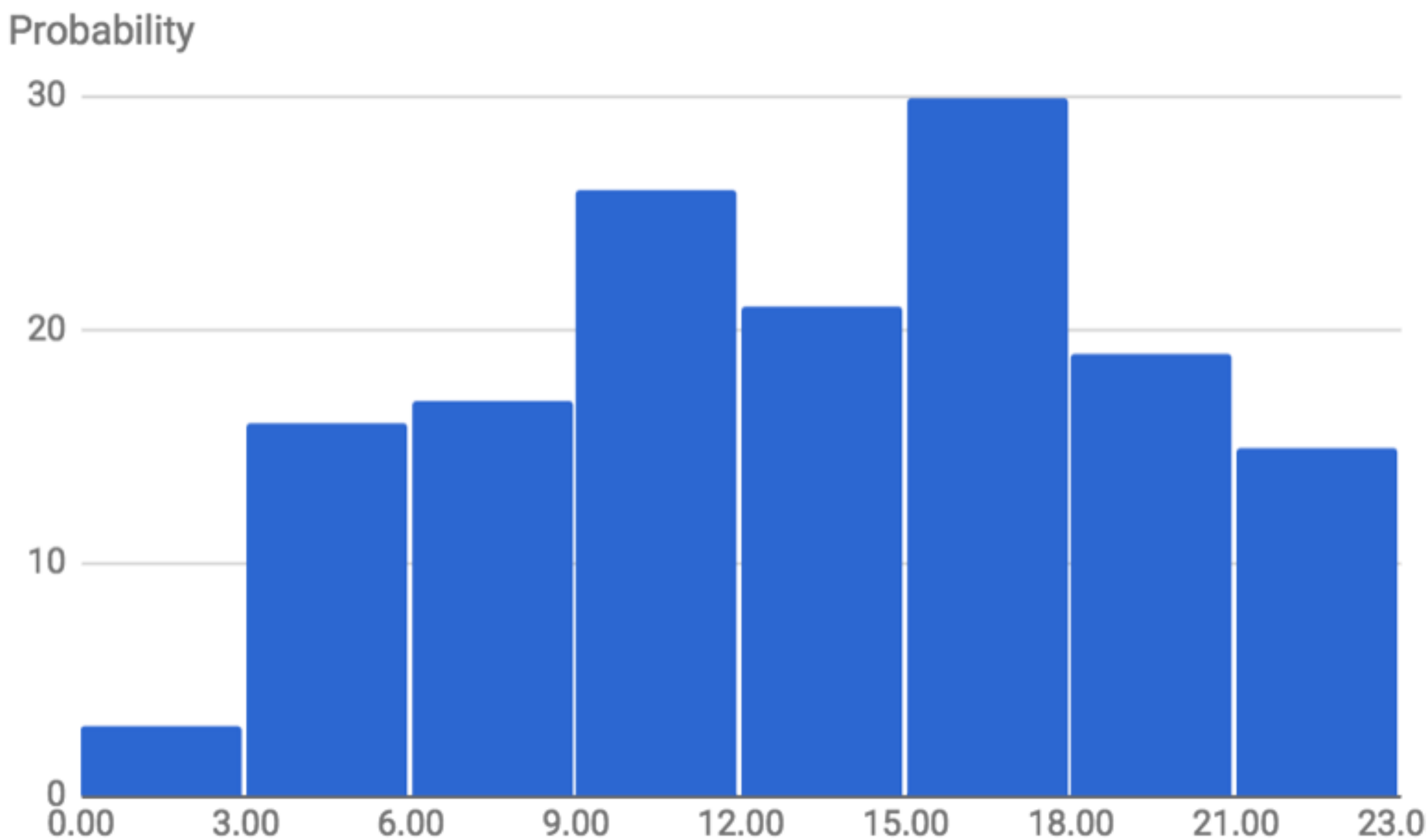
In-class Exam

Entrance Exam



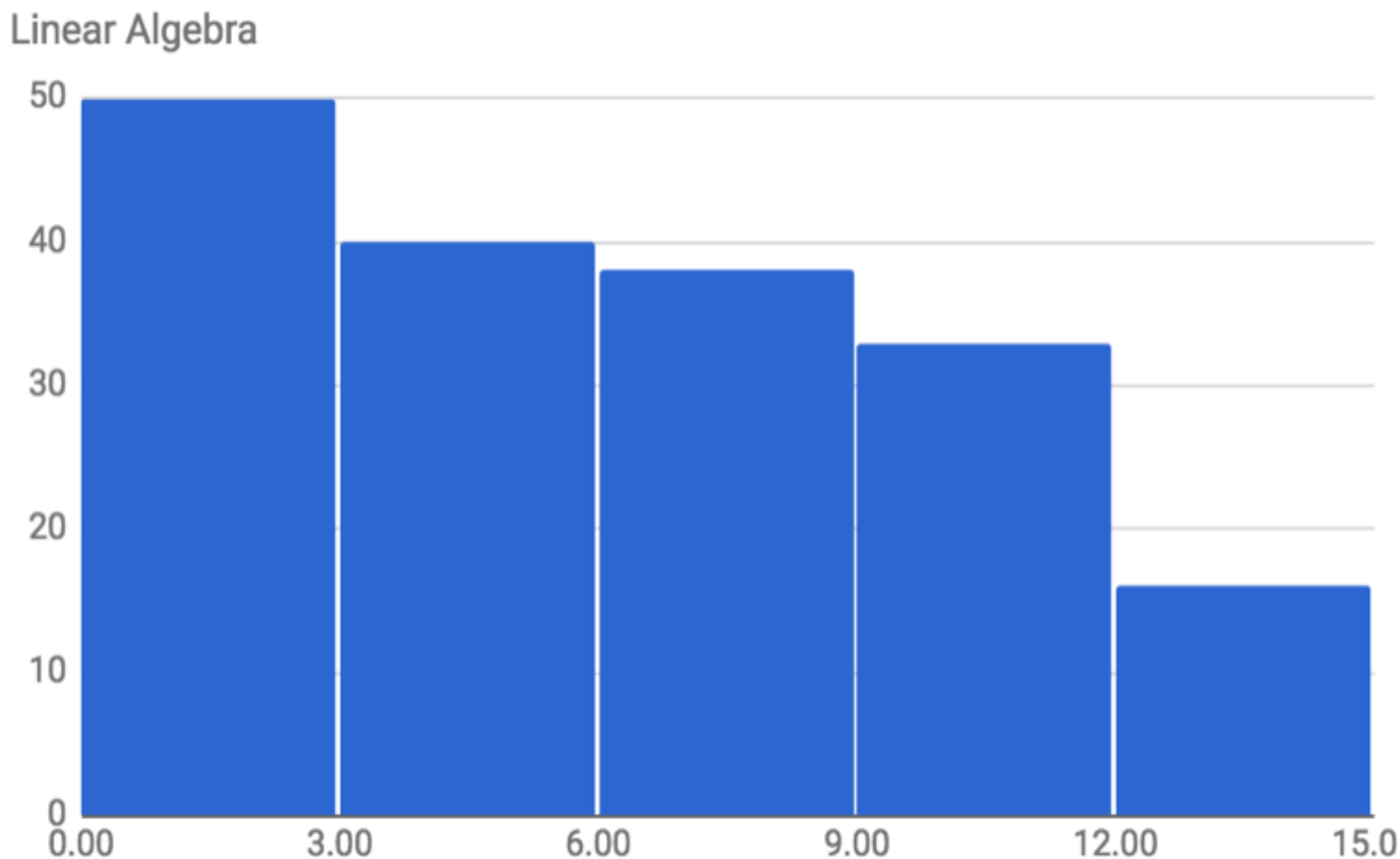
Calculus

Entrance Exam



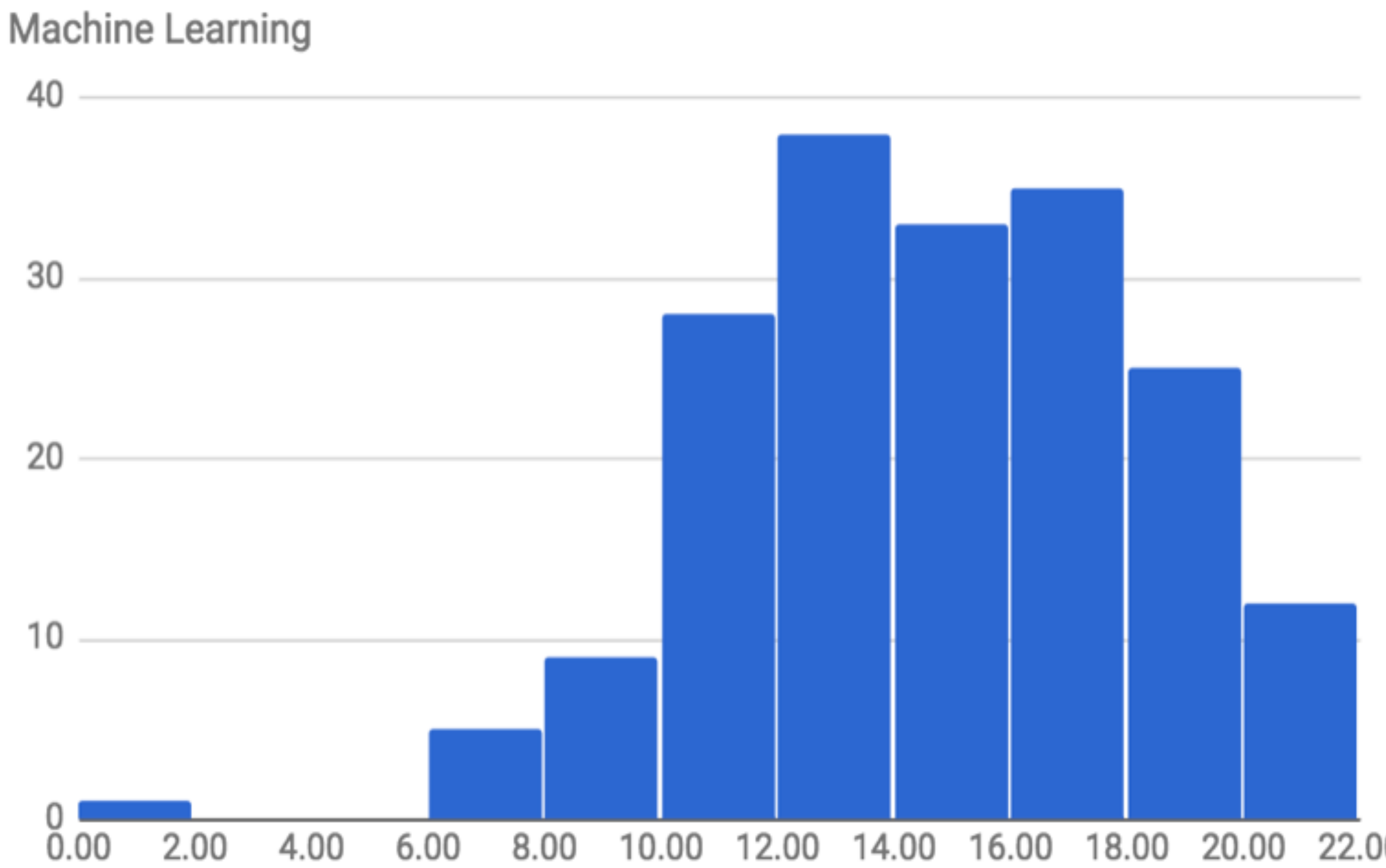
Probability

Entrance Exam



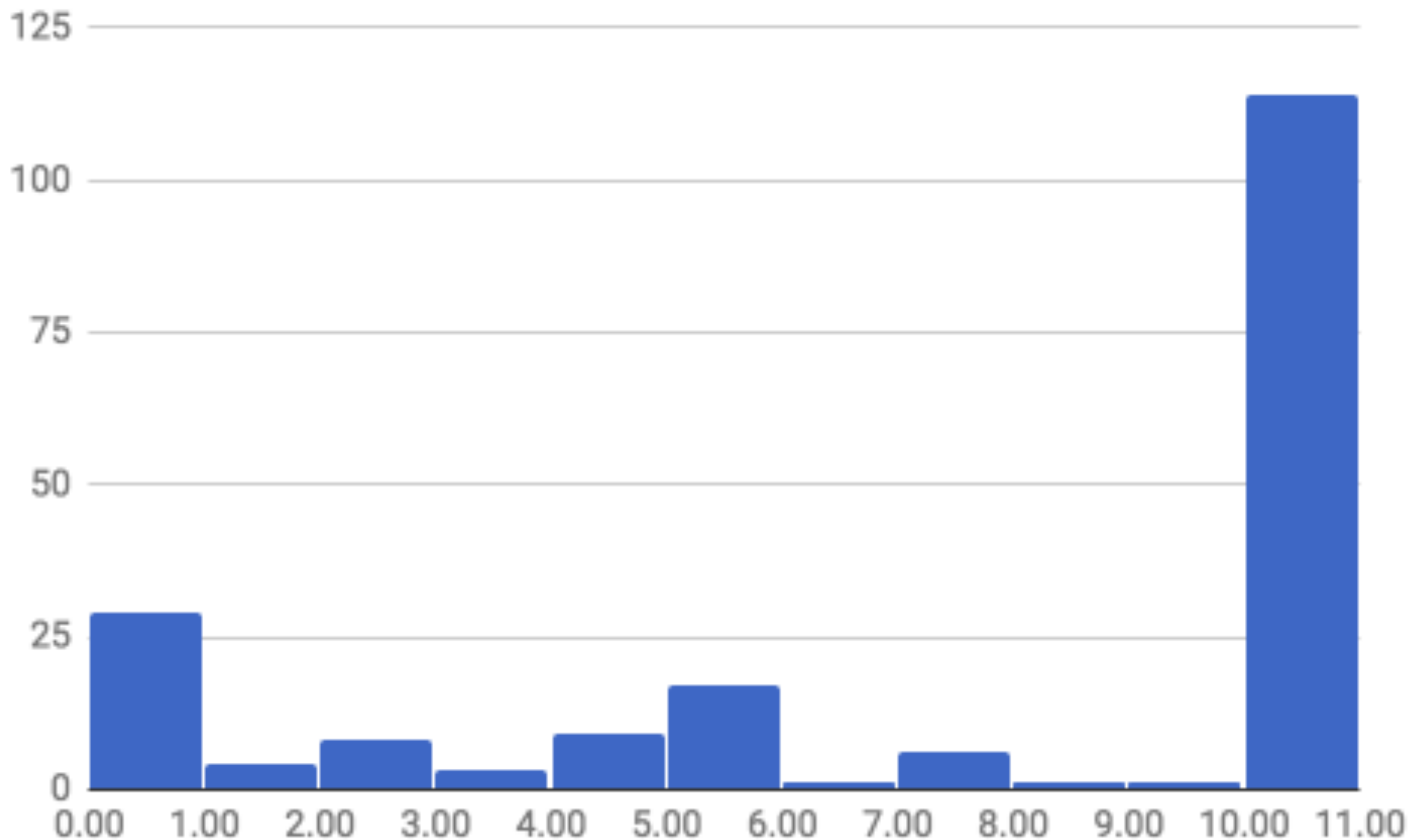
Linear Algebra

Entrance Exam



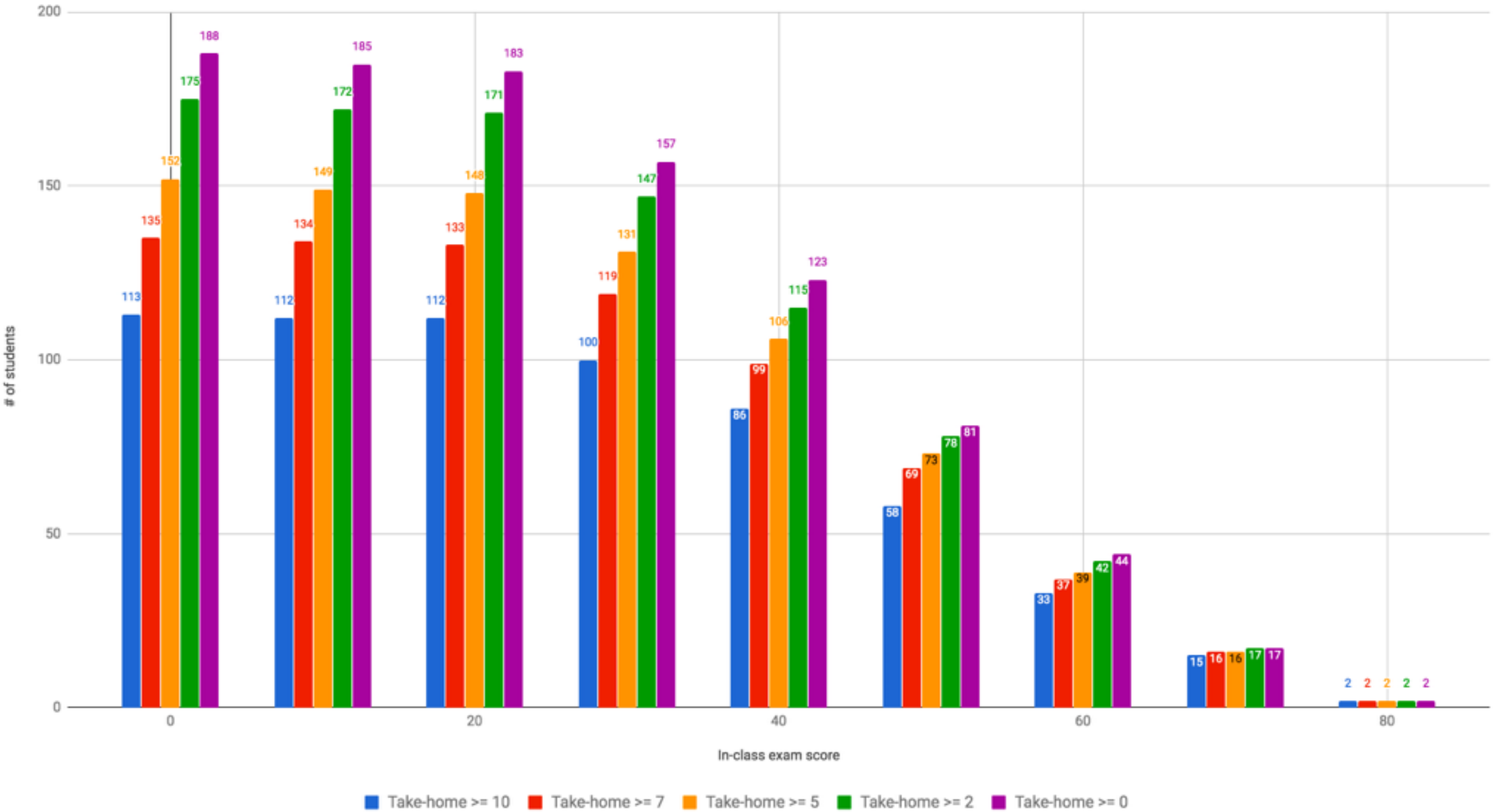
Machine Learning

Entrance Exam



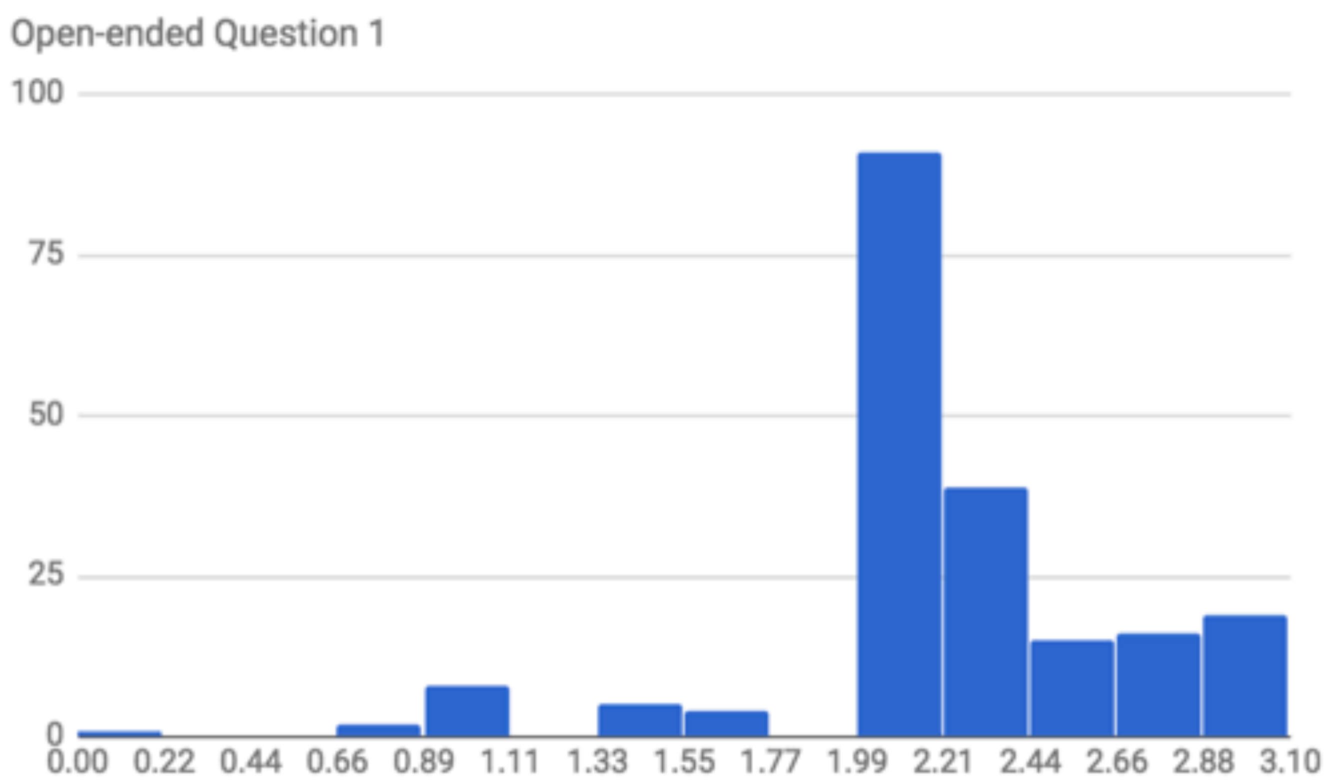
Take-home Exam

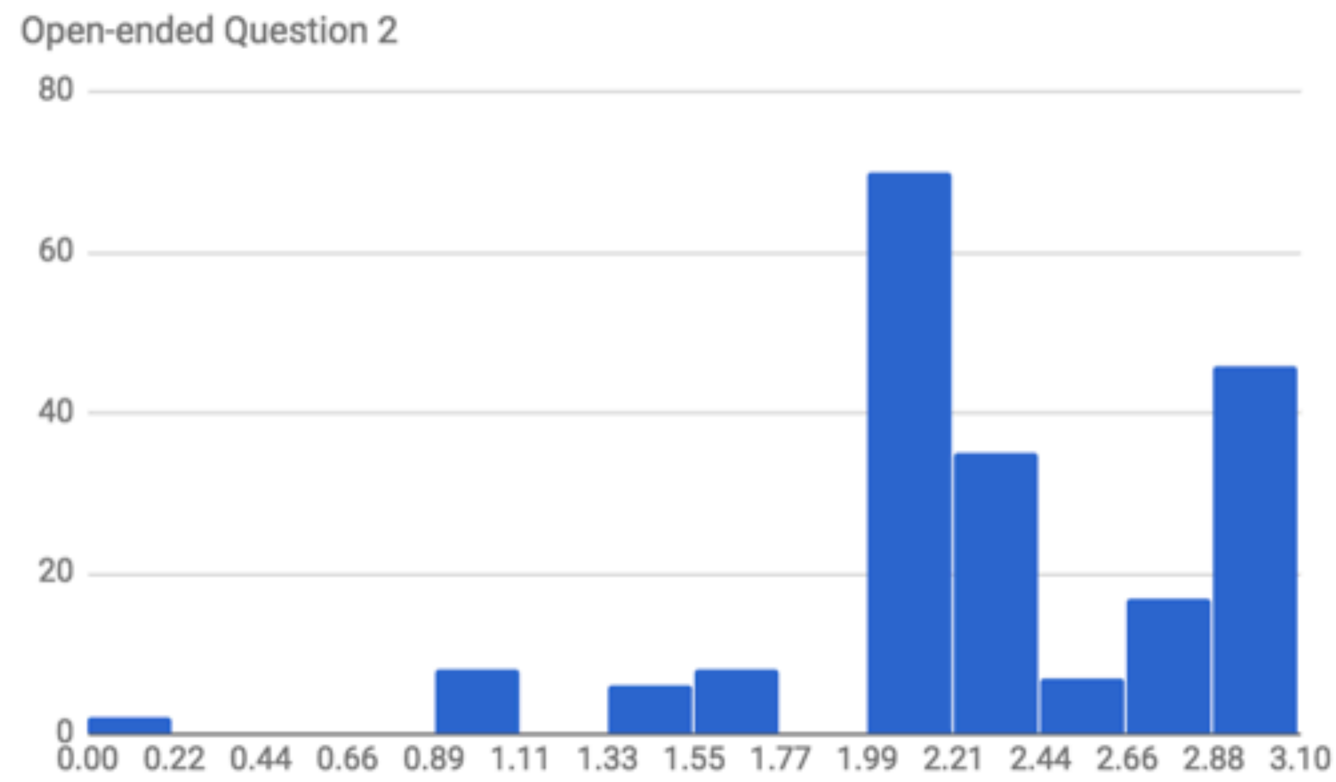
Entrance Exam



In-class + Take-home Exams

Entrance Exam





Open-ended Questions

Entrance Exam

- 5 groups based on exams
  - Group 1: excellent scores on take-home & in-class
  - Group 2: good scores on take-home & in-class
  - Group 3: well-rounded scores on in-class
  - Group 4: students who did NOT take exams
  - Group 5: none of the above
- Open-ended questions were NOT used. The main goal was to have you think about the project ahead.

Welcome to CSCI 599!

Office Hours

- Instructor OH @ SAL 214
  - Wednesday 2-3pm
  - This is NOT for homework related questions.

- TA OH @ SAL 125
  - Tuesday 1-5pm

Communication

- Please use **Piazza** for any general communication including questions
  https://piazza.com/usc/fall2017/csci599/home

- Use e-mail ONLY when it is necessary. Seriously I don't know when…
  But, the staff e-mail address is: ~~deeplearning-staff-l@usc.edu~~

- Any non-necessary e-mail will be ignored.

- **Register TODAY. Look for your project team mates!**

Communication

- Please do NOT
    - e-mail us individually (**we will not reply**)
    - come to our office without appointment

Syllabus

Week 3 9/6	Machine Learning 101 + Course registration	Attend Ian Goodfellow's talk (9/5)	
Week 4 9/13	Loss functions & Optimization + Neural Networks + Convolutional Neural Networks	**Assignment 1 OUT**	**Course Project Team**
Week 5 9/20	Training Neural Networks		
Week 6 9/27	CNN Architectures + Deep Learning Software		**Assignment 1 DUE**
Week 7 10/4	**In-class Midterm**		

Module 2			
Week 8 10/11	Recurrent Neural Networks		**Course Project Proposal DUE**
Week 9 10/18	Guest Lectures: Xiaodi Hou (TuSimple) Phillip Isola (OpenAI)	**Assignment 2 OUT**	
Week 10 10/25	Generative Models		
Week 11 11/1	Deep Reinforcement Learning		**Assignment 2 DUE**
Module 3: Advanced Topics			
Week 12 11/8	Advanced topics 1		**Course Project Mid-report**
Week 13 11/15	Advanced topics 2		
Week 14 11/22	No lecture (Thanksgiving)		
Week 15 11/29	**Term Project Presentation (4 hours)** **Spotlight + Poster**		
FINAL	**No Final**		

Subject to change!

Lecture format

- 1st module: mostly lectures
- 2nd/3rd module: lecture + TA's paper presentations

Important Dates

Subject to change!

- Assignment 1: week 6
- Midterm: week 7
- Assignment 2: week 11
- Project
  - Team formation: **week 4**
  - Project proposal: week 8
    - Project meeting with TA #1: between week 4 - week 8
    - Project meeting with Instructor #1: week 8 (M-W)
  - Project mid-report: week 12
    - Project meeting with TA #2: between week 8 - week 12
    - Project meeting with Instructor #2: week 11 (M-W)
  - Project report + Final presentation: week 15 (5-9:30pm) **4.5 hours**
    - Project meeting with TA #3: between week 12 - week 15

Course Project

- Team-based project (3-4 students per team)
- Each team will have at least 1 dedicated TA
  - Mandatory meeting with TA at least once every 3 weeks
- Create your own problems (extra points)
  - **Talk and discuss** with your TAs and me!
  - In the worst case, we will give a project idea
    - Less fun, Less points!

Course Project

- Computational resource (**be conservative!**):
  $150 Google Cloud credit per student
  $125 Amazon AWS credit per student

Today's agenda

- CSCI 599 overview
- Learning 101
- Course Entrance 1-1

Deep Learning is impacting everywhere

- Machine Learning
- Computer Vision
- Natural Language Processing
- Robotics
- Medical Application
- Graphics
- Finance
- and many more

It's matter of one function

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

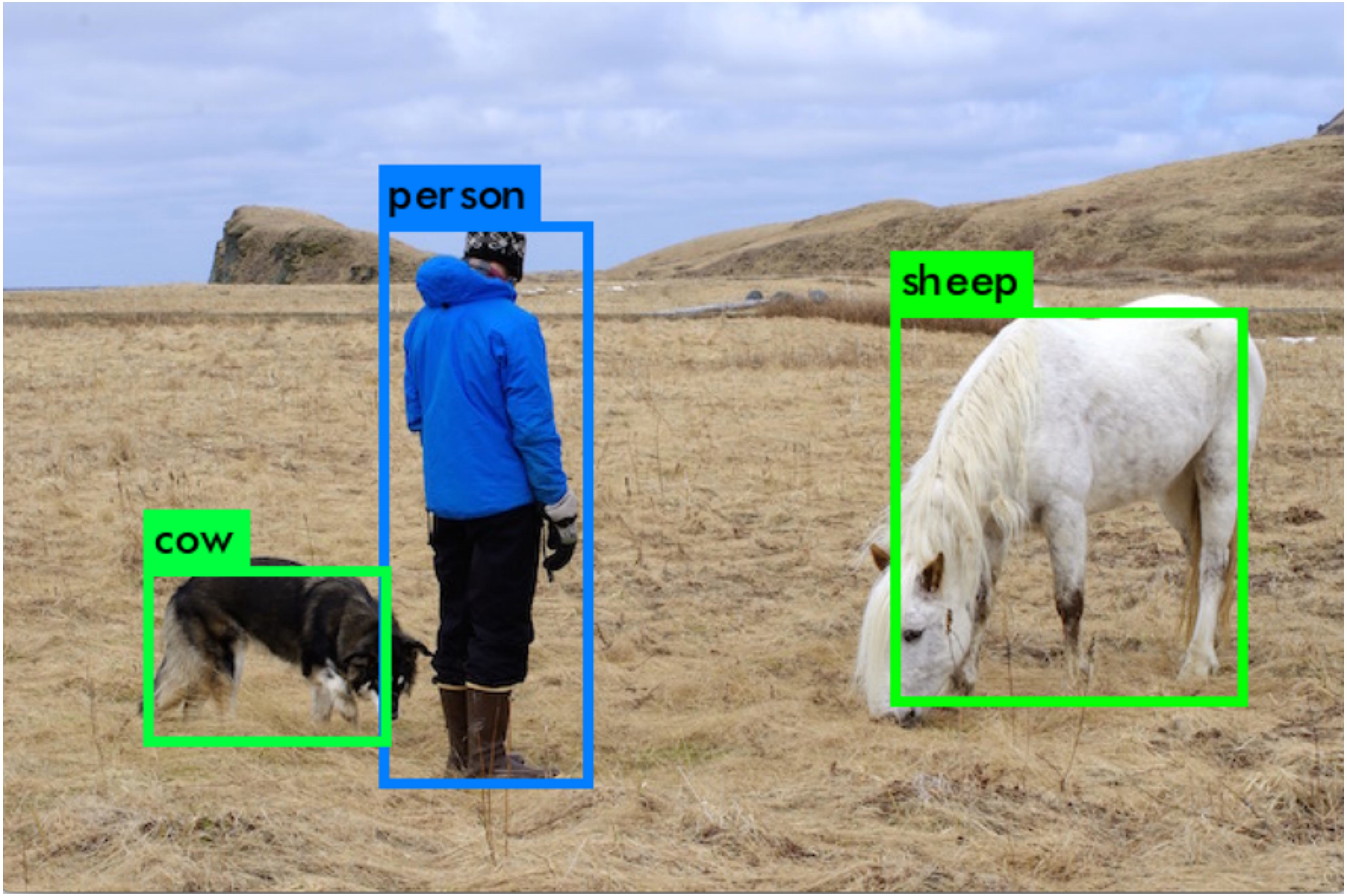
It’s matter of one function

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

Really…?

Let’s take a look

Object Detection



Object Detection in Video

J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. CVPR 2017.

Object Detection in Video

J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. CVPR 2017.

Object Detection

X:
Image

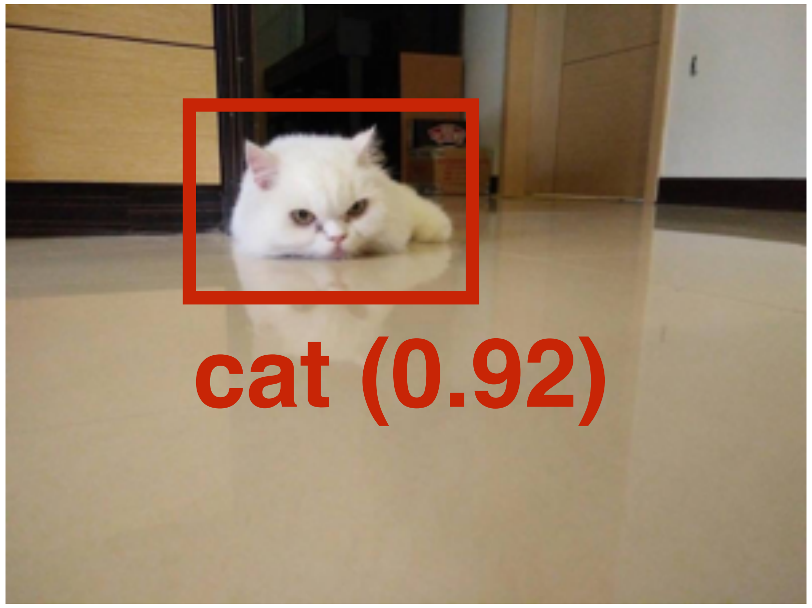
f(X)

Object Detection

X:
Image

f(X)

Y:
Object label
&
Bounding box



Semantic Segmentation

K. He, etl. al. Mask R-CNN. arXiv 2017.

Semantic Segmentation

X:
Image → f(X)

Semantic Segmentation

X:
Image

f(X)

Y:
Pixel-level
Label

3D Pose for Furniture

J. Lim, et. al. Parsing IKEA Objects: Fine Pose Estimation. ICCV 2013.

3D Pose for Furniture

J. Lim, et. al. Parsing IKEA Objects: Fine Pose Estimation. ICCV 2013.

3D Pose for Furniture

X:
Image

f(X)

3D Pose for Furniture

X:
Image

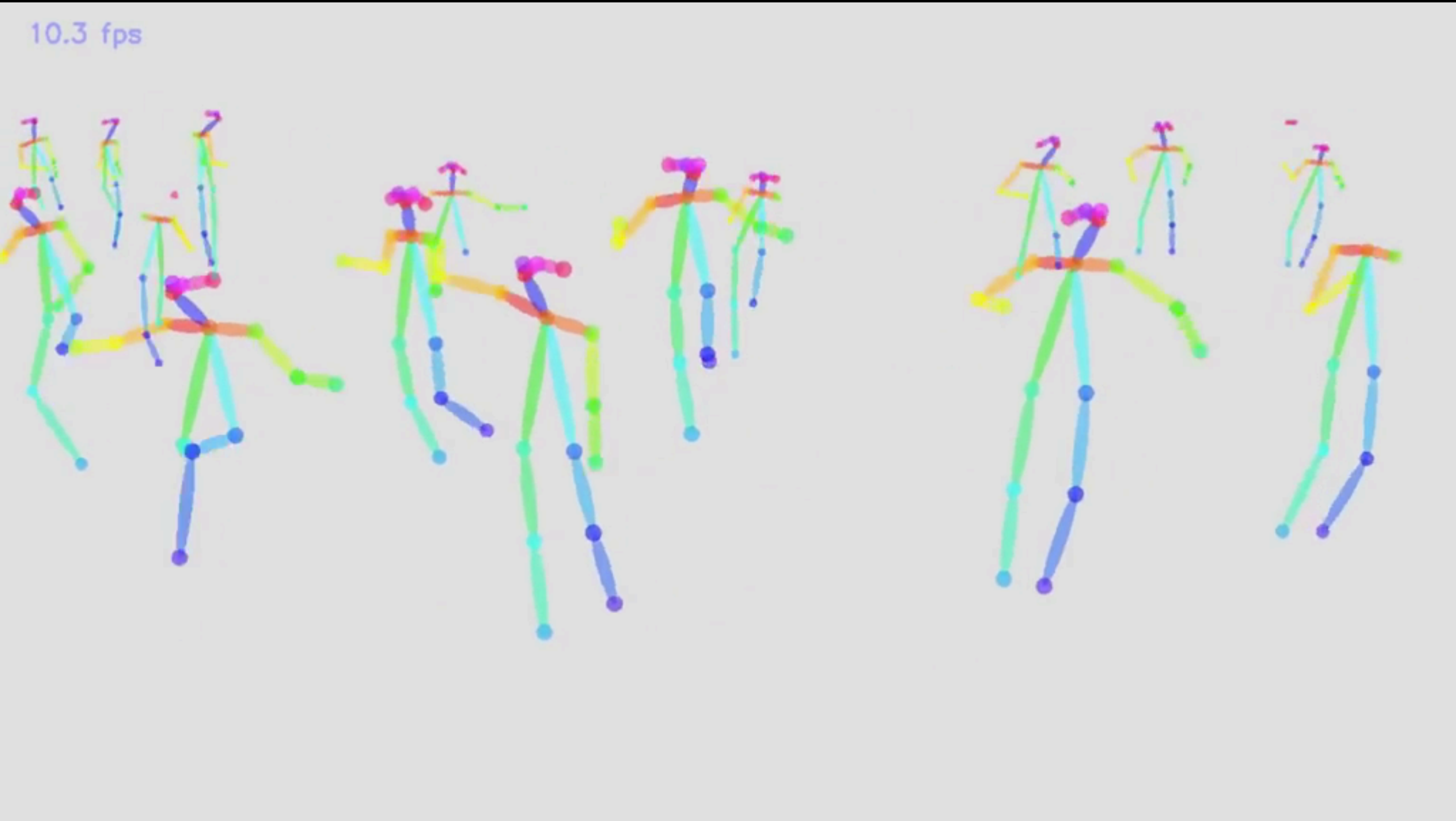
f(X)

Y:
Pose
(Scale, rotation,
and translation)

Human Pose

Z. Cao, et. al. Realtime Multi-person 2D Pose Estimation using Part Affinity Fields. CVPR 2017.

Human Pose



Z. Cao, et. al. Realtime Multi-person 2D Pose Estimation using Part Affinity Fields. CVPR 2017.

Human Pose

X:
Image

$\longrightarrow$ f(X)

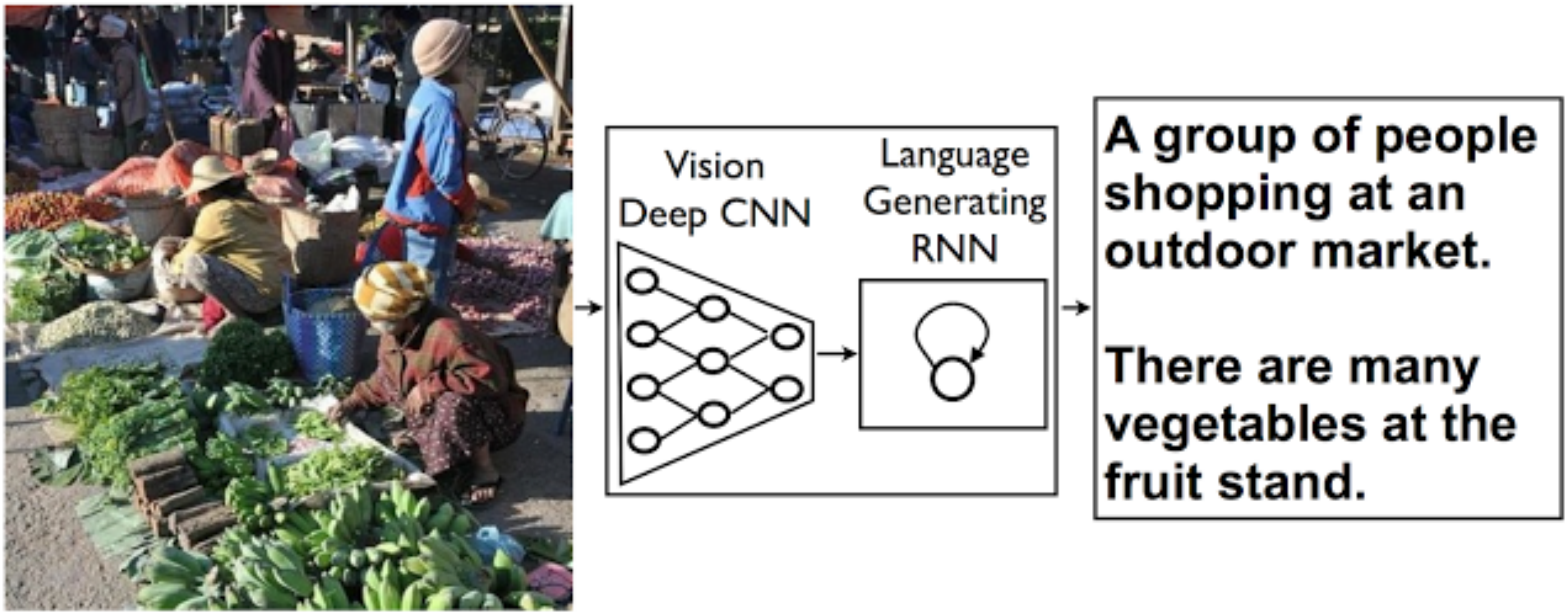
Human Pose

X:
Image

→ f(X) →

Y:
Pose of body joint

Image to Caption



From https://research.googleblog.com/2014/11/a-picture-is-worth-thousand-coherent.html

It’s matter of one function

X:
Image

$f(X)$

It’s matter of one function

X:
Image

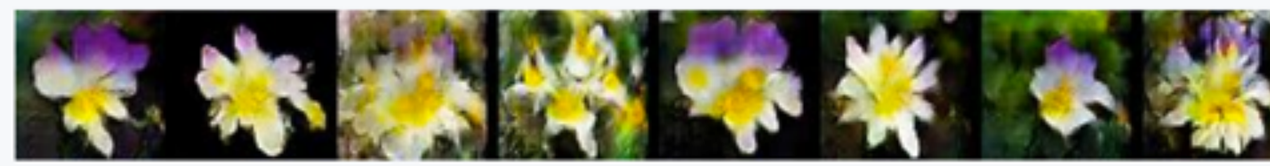
$f(X)$

Y:
Description

A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Caption to image

Caption	Generated Images
the flower shown has yellow anther red pistil and bright red petals	
this flower has petals that are yellow, white and purple and has dark lines	
the petals on this flower are white with a yellow center	

Reed, Scott, et. al. Generative Adversarial Text to Image Synthesis. ICML 2016.

It’s matter of one function

X:
Description

$\rightarrow$ f(X)

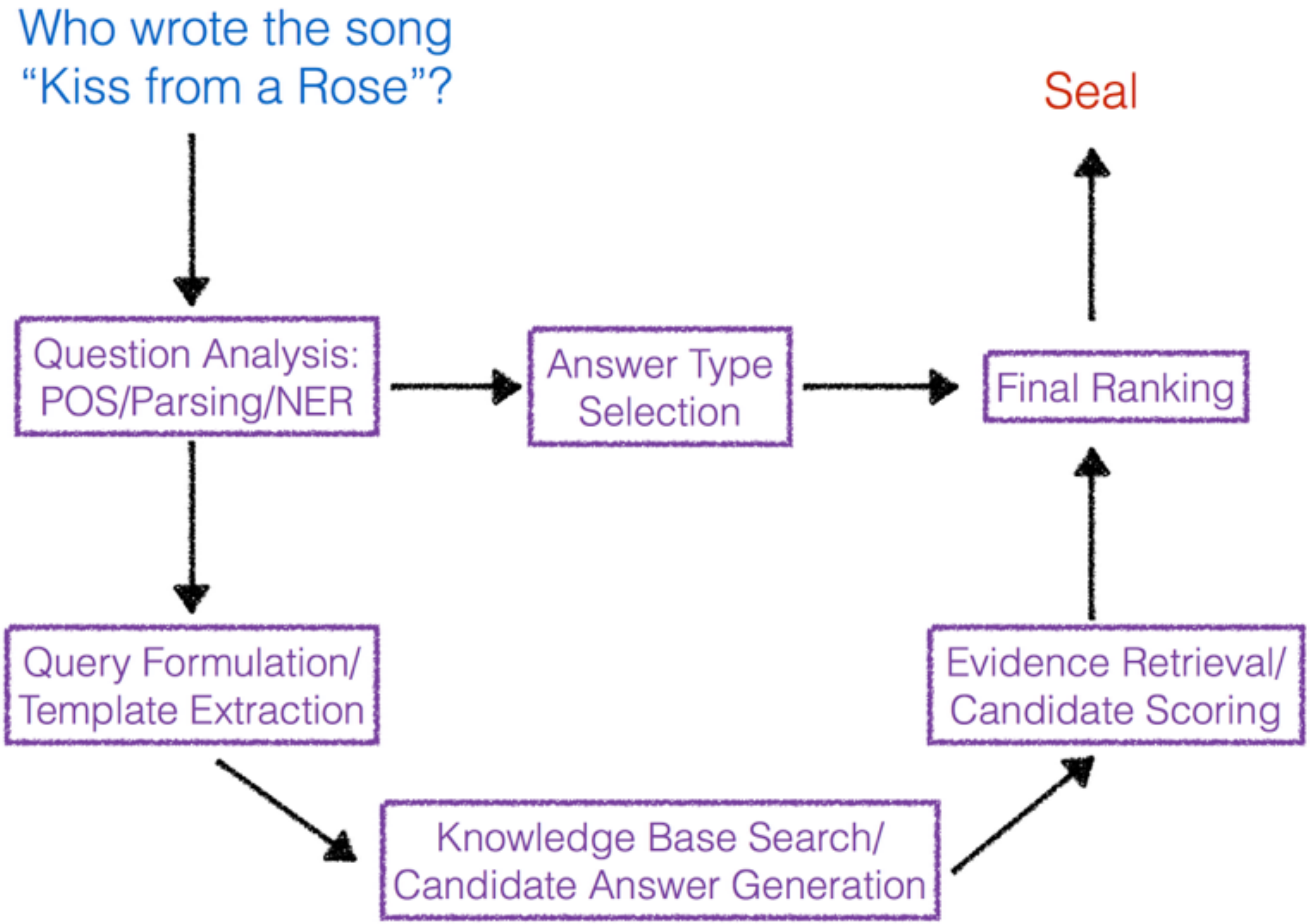
This flower has a
lot of small round
pink petals.

It's matter of one function

X: Description → f(X) → Y: Image

This flower has a lot of small round pink petals.

Question Answering



Kumar, Ankit, et. al. "Ask me anything: Dynamic memory networks for natural language processing." ICML 2016.

Question Answering

X:
Query

$\longrightarrow$ f(X) $\longrightarrow$

Y:
Answer

Who is the most
handsome person
in the world?

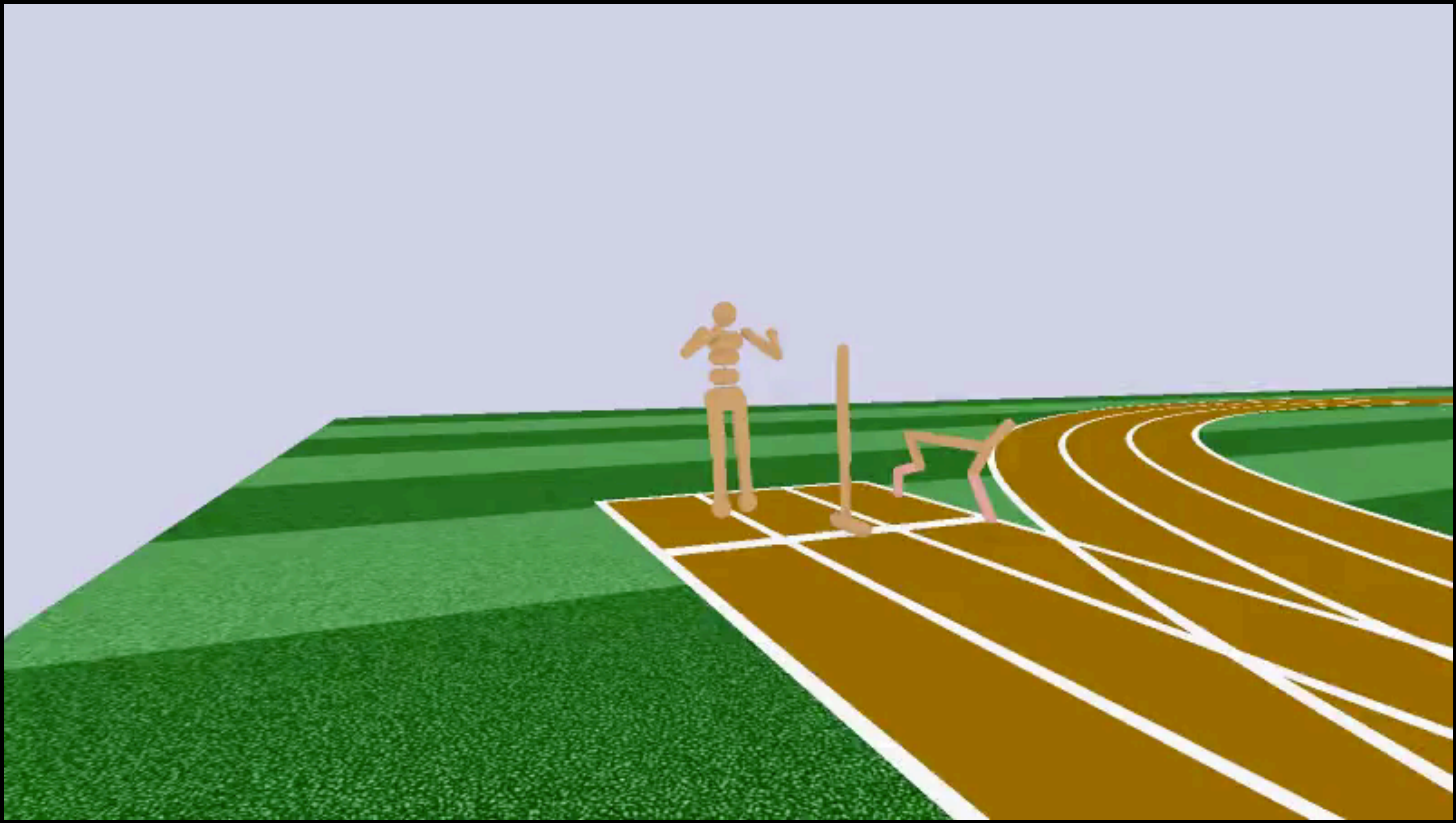
Question Answering

X: Query → f(X) → Y: Answer

Who is the most handsome person in the world?

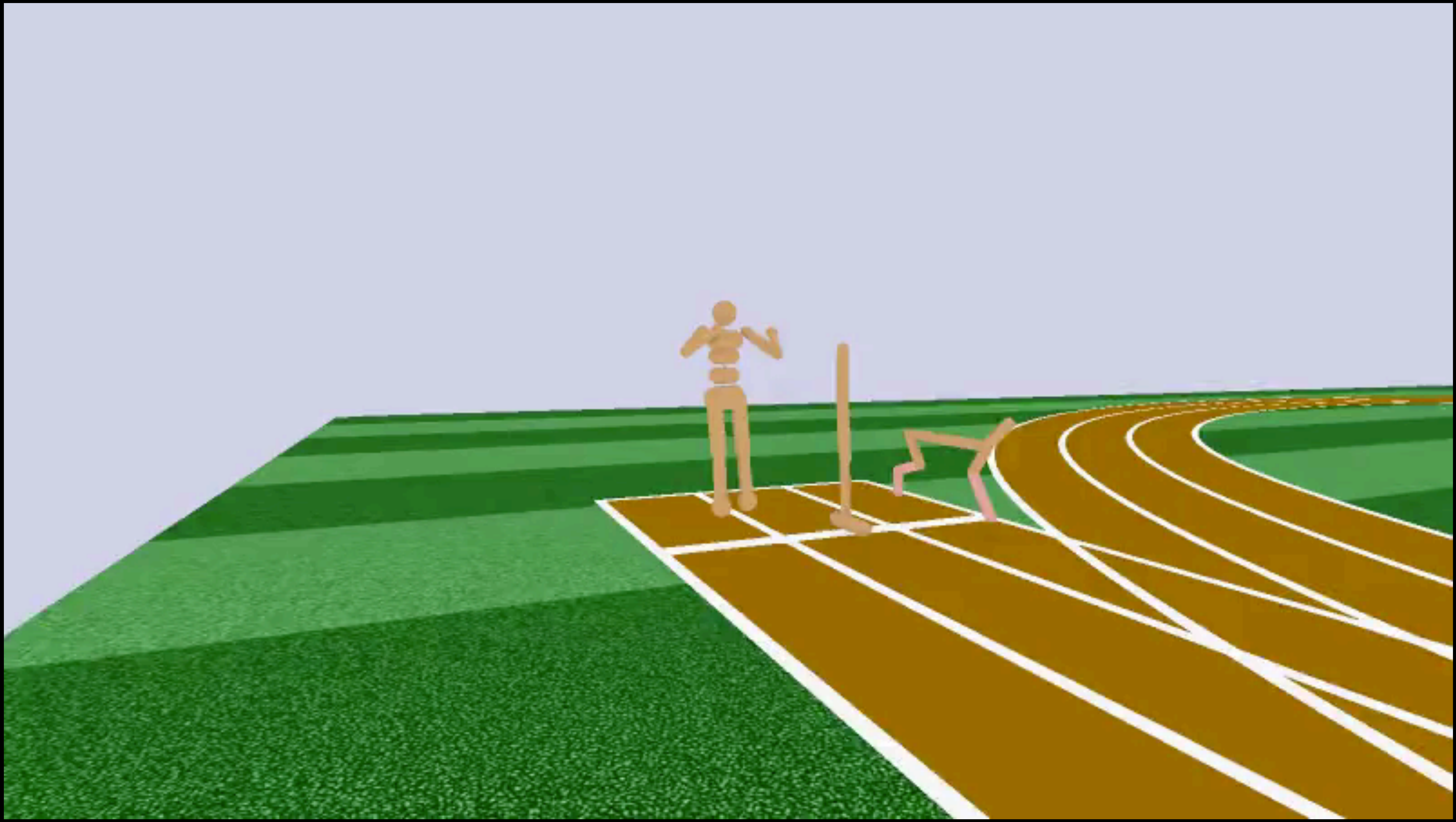
Me

Learning to Walk



From https://blog.openai.com/roboschool/

Learning to Walk

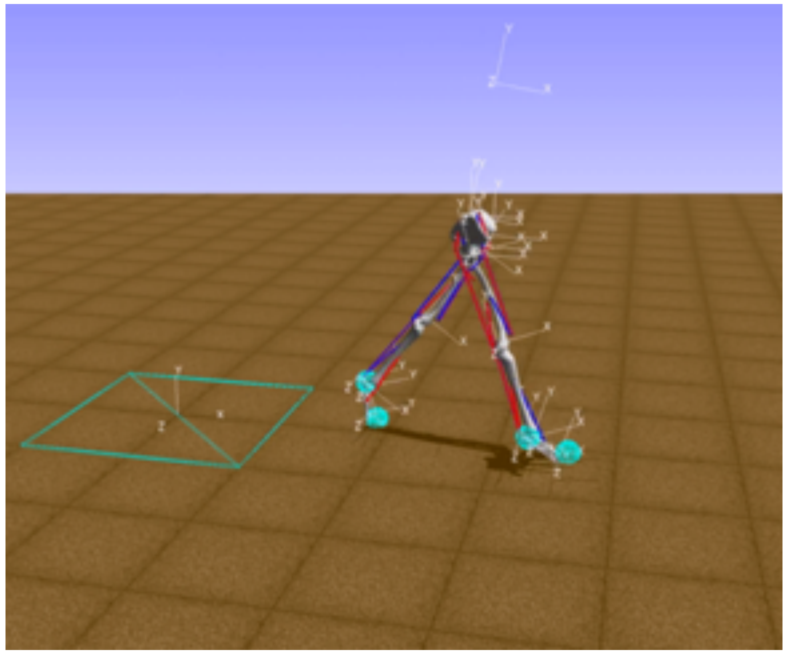


From https://blog.openai.com/roboschool/

Learning to Walk

X:
States

f(X)

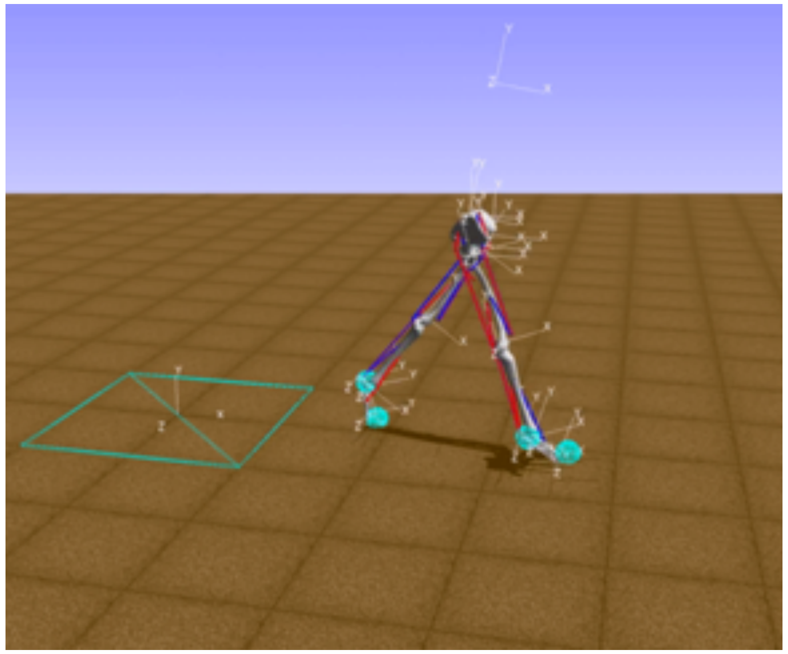


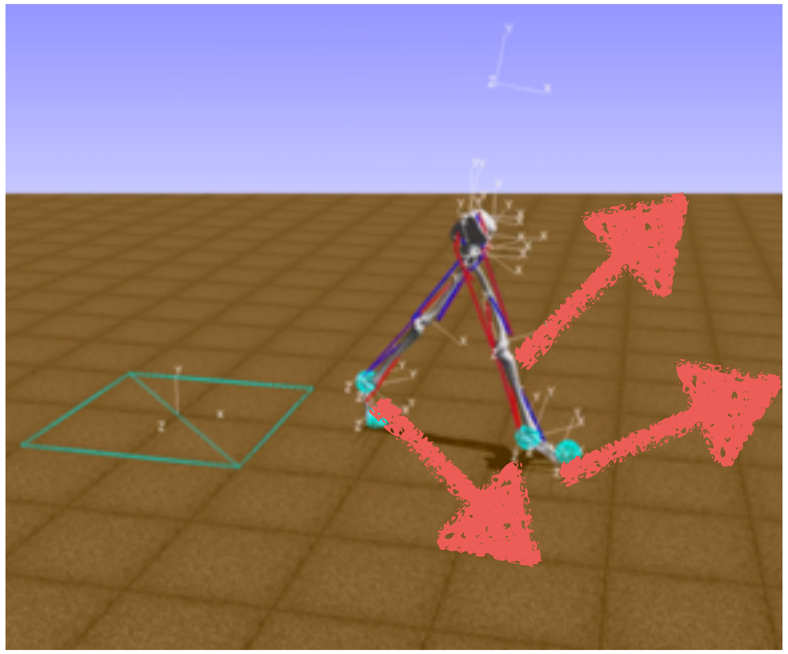
Learning to Walk

X:
States

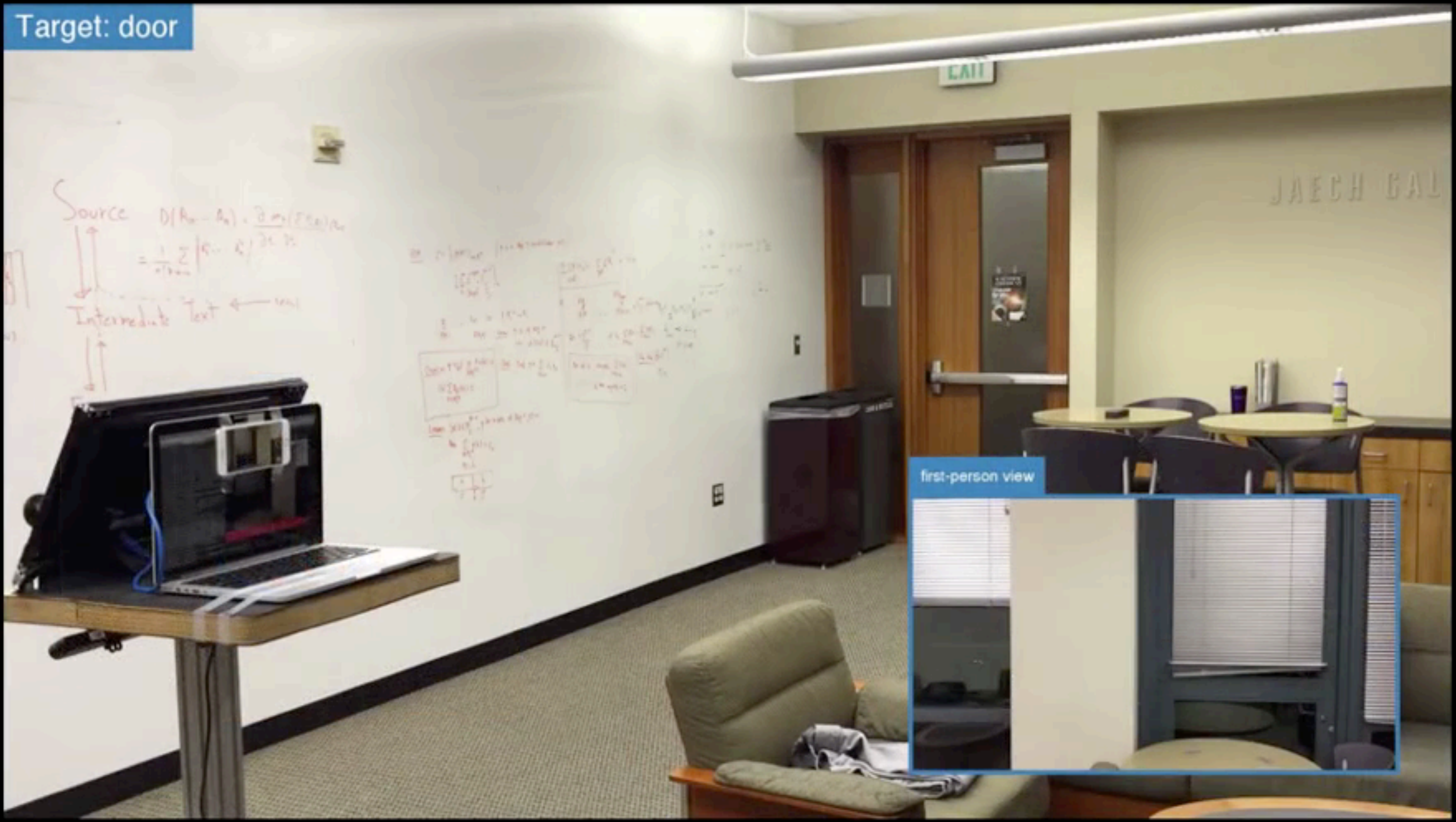
$f(X)$

Y:
Action



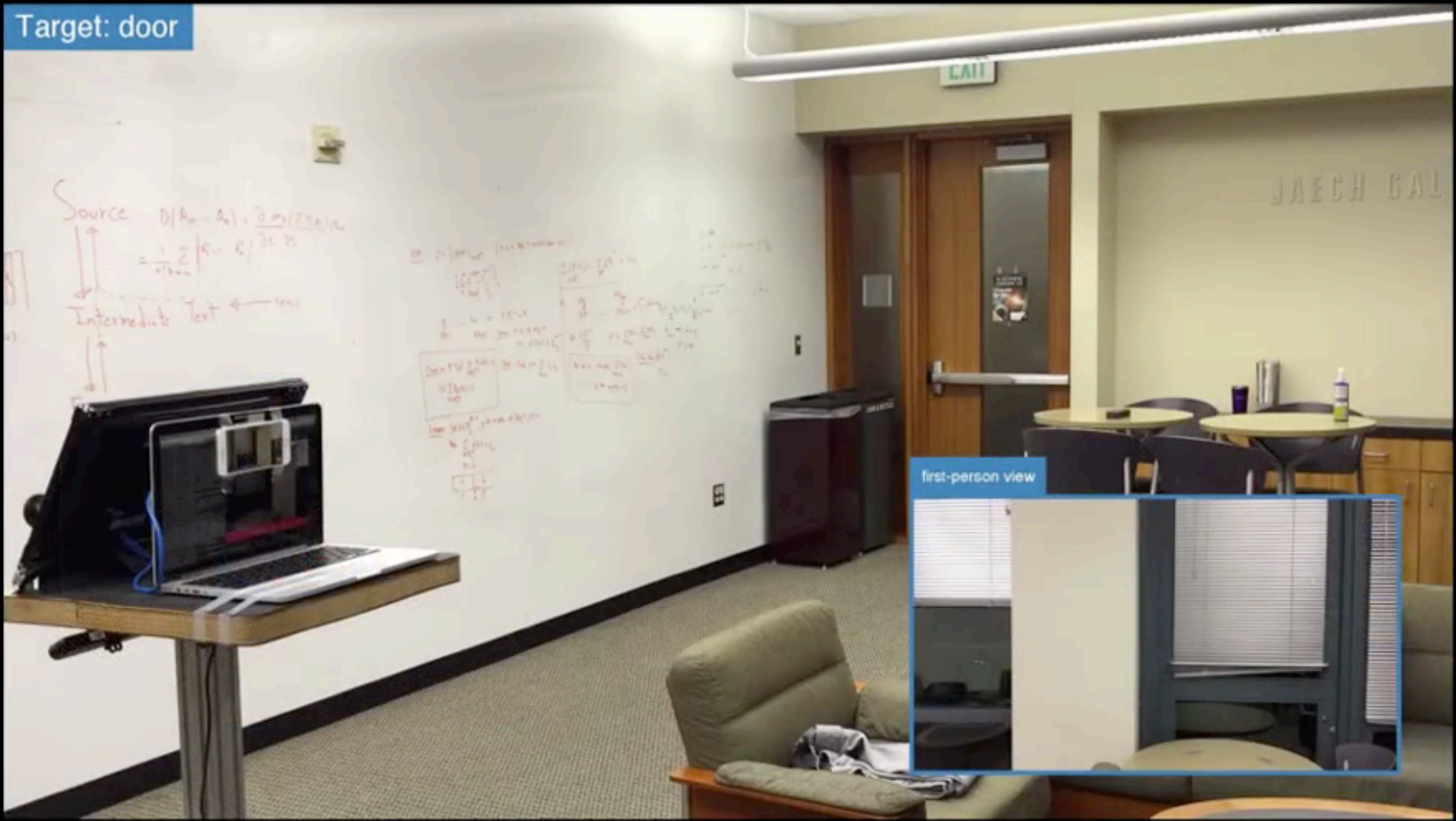


Navigation Robot



Y. Zhu, et. al. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. ICRA 2017.

Navigation Robot



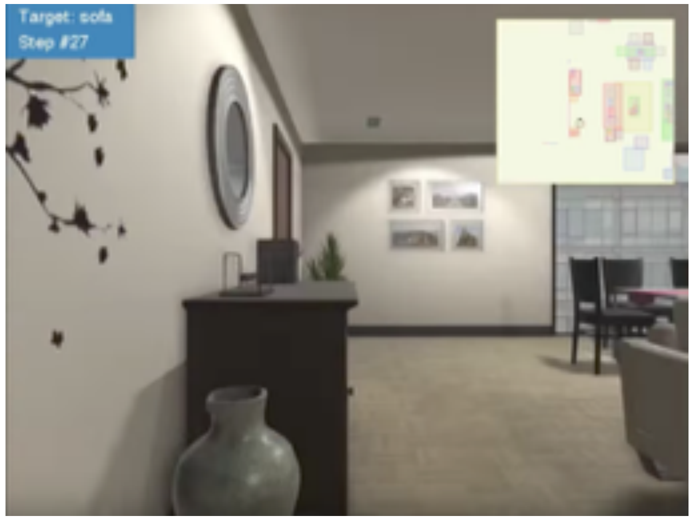
Y. Zhu, et. al. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. ICRA 2017.

Navigation Robot

X:
Goal &
Observation → f(X)

Goal

Observation



Navigation Robot

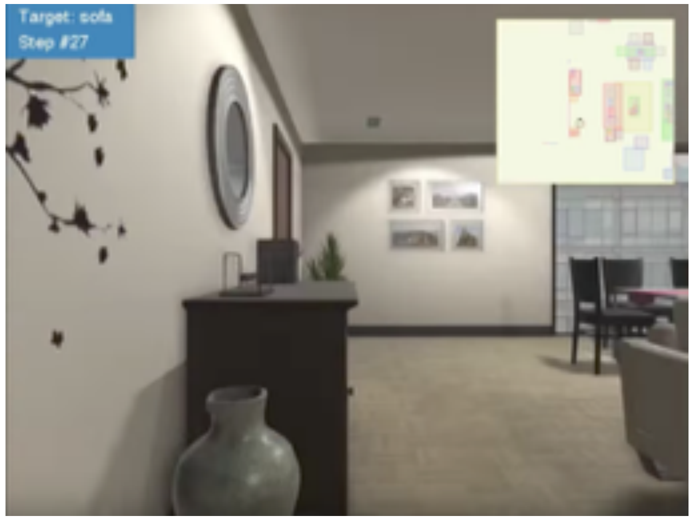
X:
Goal &
Observation

→ f(X) →

Y:
Action

Goal

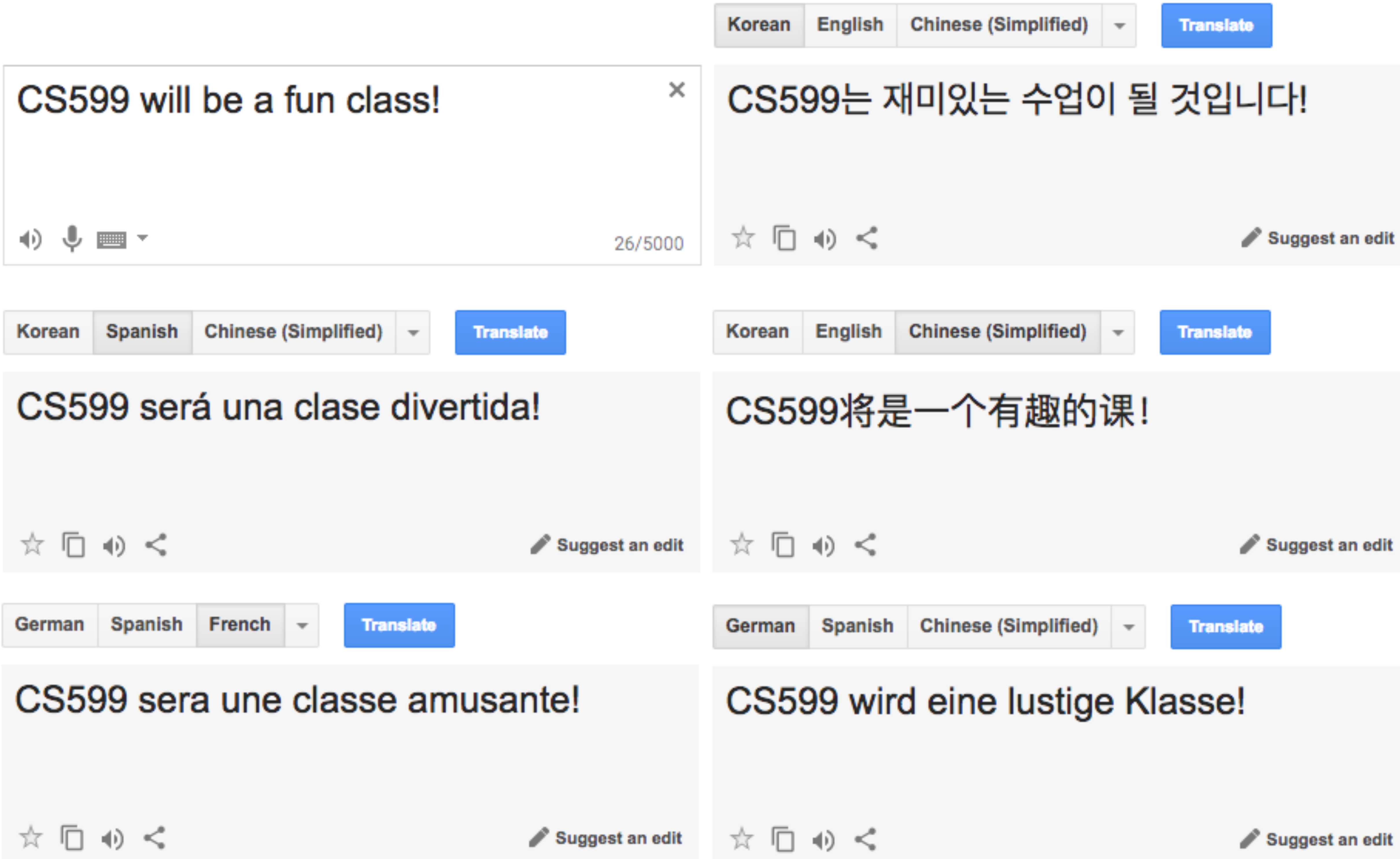
Observation



turn left

Practice

Machine Translation



Google Translate

Machine Translation

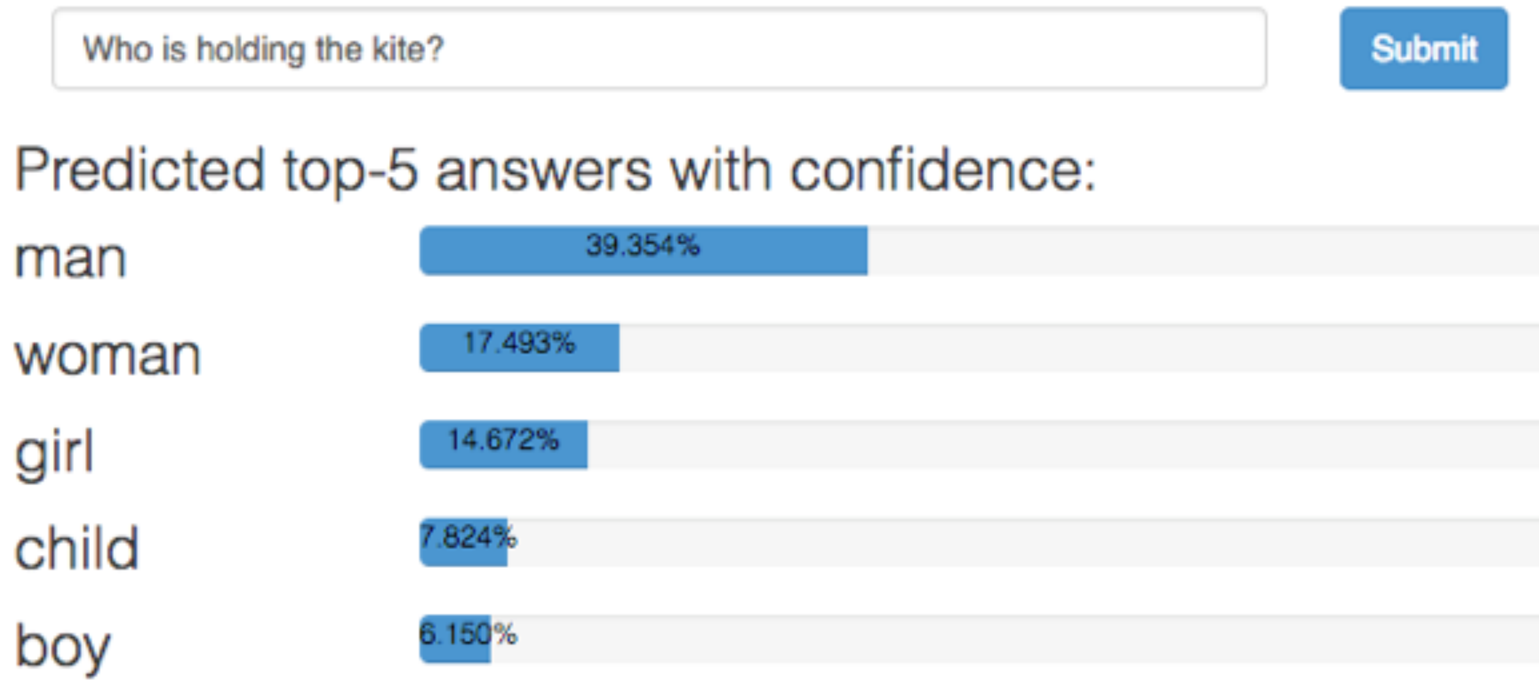
X:
Text in
Language A

f(X)

Machine Translation

X: Text in Language A → f(X) → Y: Text in Language B

Visual Question Answering (VQA)



From http://www.visualqa.org/

Visual Question Answering (VQA)

X:
Image and
Query

$\longrightarrow$ f(X)

Who is holding
the kite?

Visual Question Answering (VQA)

X:
Image and
Query

f(X)

Y:
Answer

Who is holding
the kite?

Answer: Man

Object Picking Robot

S. Levine, et. al. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.. IJRR 2017.

Object Picking Robot

S. Levine, et. al. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection.. IJRR 2017.

Object Picking Robot

X: Observation

f(X)

Object Picking Robot

X:
Observation

$\longrightarrow$ f(X) $\longrightarrow$

Y:
Action

Autonomous Driving

From AutoX

Autonomous Driving

From AutoX

Autonomous Driving

X:
Camera Input,
Speed, etc

→

f(X)

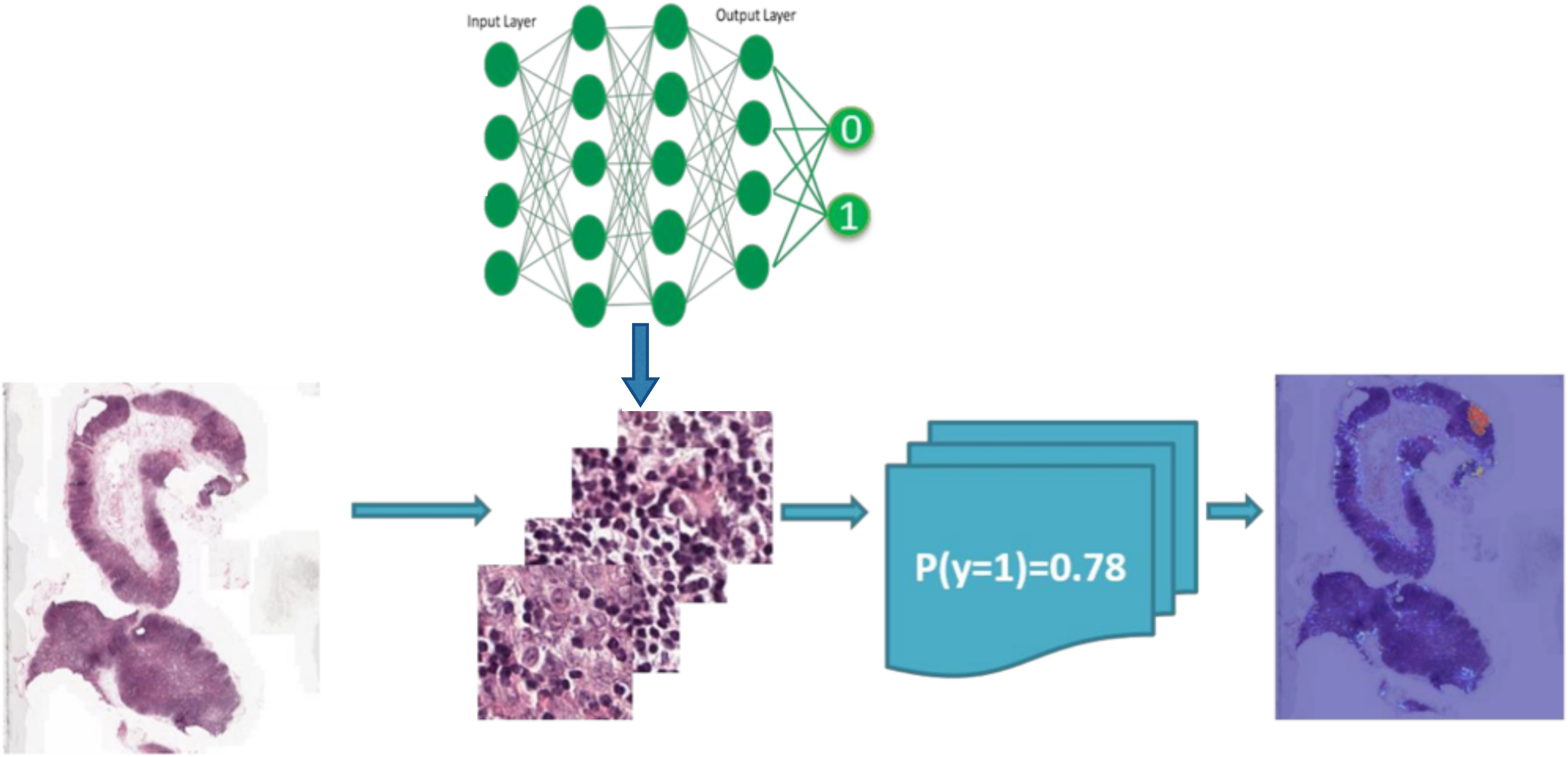
Autonomous Driving

X:
Camera Input,
Speed, etc

→ f(X) →

Y:
Action

Cancer Metastases Detection



From PathAI's submission to CAMELYON16.

Cancer Metastases Detection

X: Image → f(X)

Cancer Metastases Detection

X: Image → f(X) → Y: Prevalence

It's matter of one function

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

Ok. It is true…

It's matter of one function

Question

- How is this related to intelligence?
- What does it mean to have one function for all intelligence?

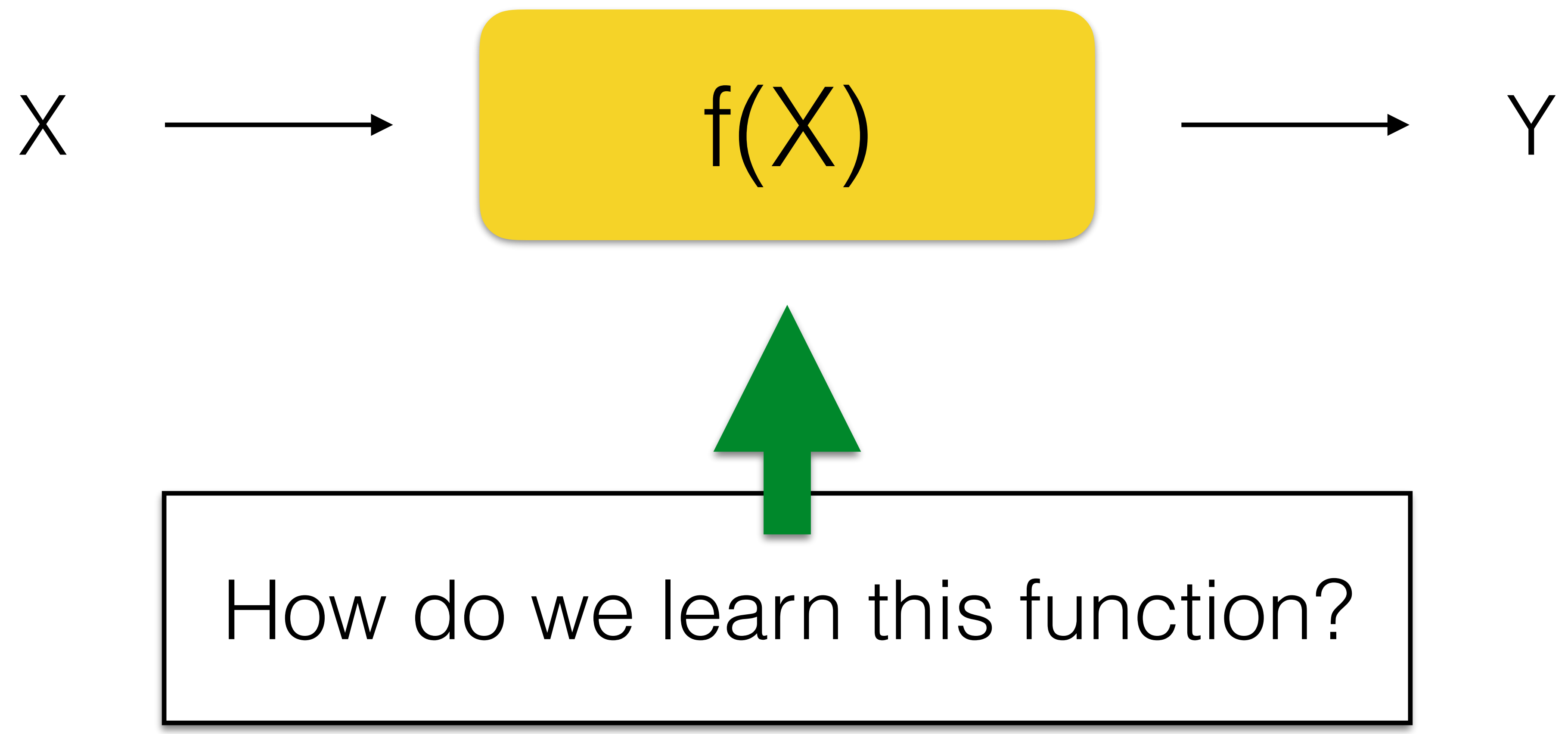
$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

Ok. It is true…

What's the challenge then?

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

What's the challenge then?



Types of Learning

$X \longrightarrow \boxed{f(X)} \longrightarrow Y$

- Supervised Learning
- Unsupervised Learning
- Weakly / Semi-supervised Learning
- Reinforcement Learning

Definition from Dhruv Batra's deep learning course (ECE 5604)

Types of Learning

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

- Supervised Learning — desired output (**Y**) in training data
- Unsupervised Learning
- Weakly / Semi-supervised Learning
- Reinforcement Learning

Definition from Dhruv Batra's deep learning course (ECE 5604)

Types of Learning

$X \longrightarrow \boxed{f(X)} \longrightarrow Y$

- Supervised Learning — desired output (**Y**) in training data
- Unsupervised Learning — **Y** not in training data
- Weakly / Semi-supervised Learning
- Reinforcement Learning

Definition from Dhruv Batra's deep learning course (ECE 5604)

Types of Learning

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

- Supervised Learning — desired output (**Y**) in training data
- Unsupervised Learning — **Y** not in training data
- Weakly / Semi-supervised Learning — some of **Y** in training data
- Reinforcement Learning

Definition from Dhruv Batra's deep learning course (ECE 5604)

Types of Learning

$$X \longrightarrow f(X) \longrightarrow Y$$

- Supervised Learning — desired output (**Y**) in training data
- Unsupervised Learning — **Y** not in training data
- Weakly / Semi-supervised Learning — some of **Y** in training data
- Reinforcement Learning — rewards based on a set of actions

Definition from Dhruv Batra's deep learning course (ECE 5604)

Types of Learning

X → f(X)

Question

- Ian Goodfellow's talk yesterday
- Where does GAN fall into?

- Supervised Learning — desired output (**Y**) in training data
- Unsupervised Learning — **Y** not in training data
- Weakly / Semi-supervised Learning — some of **Y** in training data
- Reinforcement Learning — rewards based on a set of actions

Definition from Dhruv Batra's deep learning course (ECE 5604)

Our goal is “to approximate”

$X \longrightarrow \boxed{f(X)} \longrightarrow Y$

There may exist an exact function (f*) mapping from X to Y.

Our goal is not to find this exact function.

Rather, we are happy as long as f(X) can **approximate** f*(x). f does NOT have to be exactly f*.

This course will be about

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

(1) How do we learn this function (using deep learning)?
(2) How to formulate a problem into this

Our goal is "to approximate"

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

For f(X) to approximate any f*(X), f is better to be highly capable.

Our goal is “to approximate”

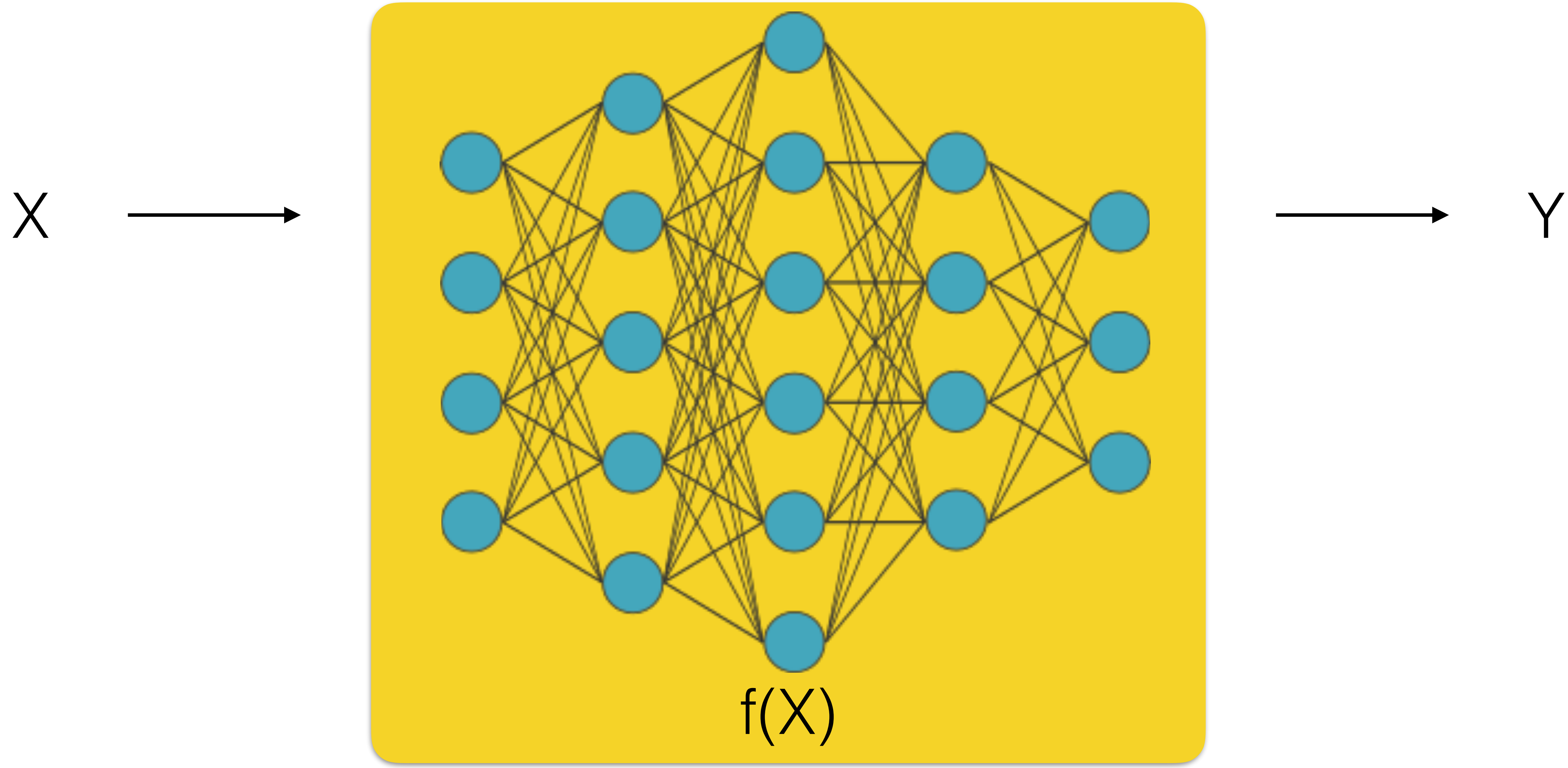
$X \longrightarrow \boxed{f(X)} \longrightarrow Y$

For f(X) to approximate any f*(X), f is better to be highly capable.

Deep learning is an effective method for this

- Non-linear (high capacity)
- Hierarchical
- End-to-End learning

Deep Learning is



- Non-linear (high capacity)
- Hierarchical
- End-to-End learning

Linear Classification

$$X \longrightarrow \boxed{f(X)} \longrightarrow Y$$

Let's first talk about learning a simple function.

Recap: Image classification

X → f(X) → Y

50x50x3 numbers
(7500 numbers)

One-hot class vector
(e.g. <0,0,0,1,0>)

Modified from CS 231N @ Stanford

Parametric Approach

$$f(x; W) = Wx$$

X → $f(X; W)$ → Y

50x50x3 numbers
(7500 numbers)

Weights
(that we want to learn)

One-hot class vector
(e.g. <0,0,0,1,0>)

Modified from CS 231N @ Stanford

Linear Classification

$f(x; \color{red}{W}) = \color{red}{W}x$

$X \longrightarrow f(X; \color{red}{W}) \longrightarrow Y$

50x50x3 numbers
(7500 numbers)

Weights
(that we want to learn)

One-hot class vector
(e.g. <0,0,0,1,0>)

Modified from CS 231N @ Stanford

Linear Classification

$$\overset{\text{5x1}}{\boxed{f(x; W)}} = \overset{\text{5x7500}}{\boxed{W}}\overset{\text{7500x1}}{\boxed{x}}$$

X → f(X; W) → Y

50x50x3 numbers
(7500 numbers)

Weights
(that we want to learn)

One-hot class vector
(e.g. <0,0,0,1,0>)

Modified from CS 231N @ Stanford

Linear Classification

$$\underbrace{\boxed{f(x; W)}}_{5\text{x}1} = \underbrace{\boxed{W}}_{5\text{x}7500}\,\underbrace{\boxed{x}}_{7500\text{x}1} + \underbrace{\boxed{b}}_{5\text{x}1}$$

X → f(X; W) → Y

50x50x3 numbers
(7500 numbers)

Weights
(that we want to learn)

One-hot class vector
(e.g. <0,0,0,1,0>)

Modified from CS 231N @ Stanford

Linear Classification

$$\underbrace{\boxed{f(x; W)}}_{5\times1} = \underbrace{\boxed{W}}_{5\times7500}\underbrace{\boxed{x}}_{7500\times1} + \underbrace{\boxed{b}}_{5\times1}$$

X → f(X; W) → Y

50x50x3 numbers
(7500 numbers)

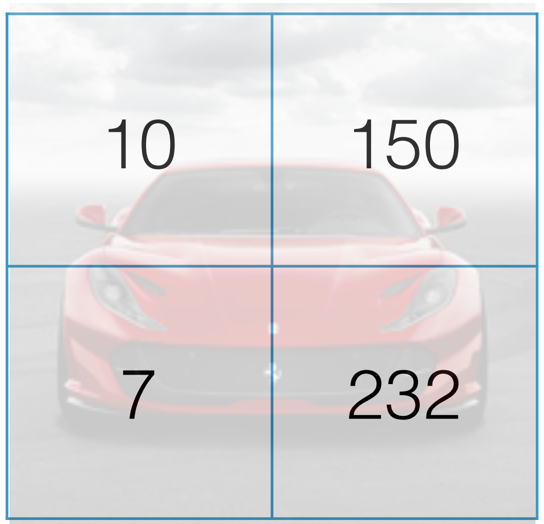
Weights
(that we want to learn)

One-hot class vector
(e.g. <0,0,0,1,0>)

Y: output
X: input
W, b: **learned weight**

Modified from CS 231N @ Stanford

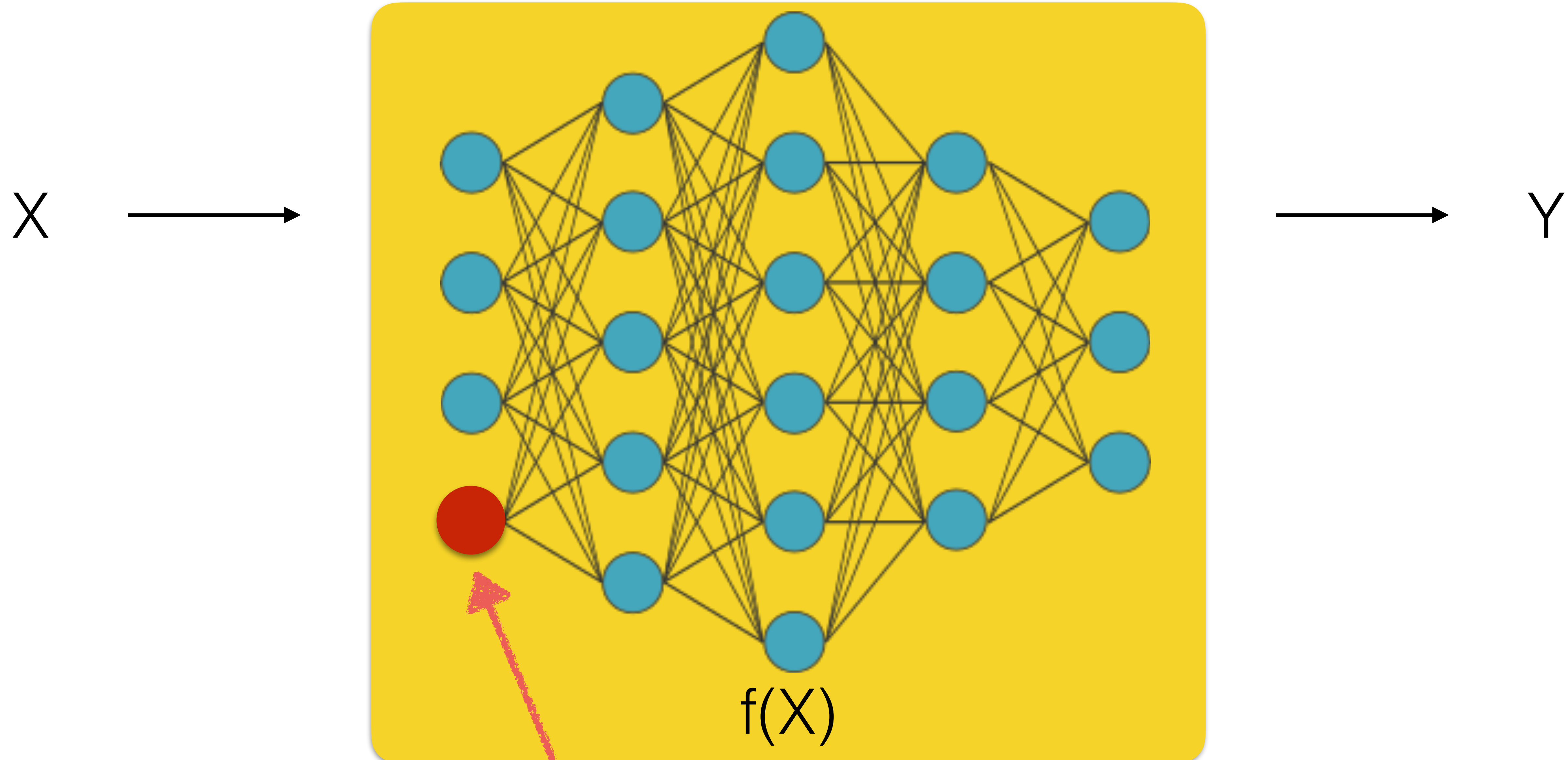
Linear Classification



$$
\underbrace{\begin{bmatrix} 0.2 & -0.3 & 0 & 0.6 \\ -1 & 0.4 & 0.2 & 0.7 \\ -0.9 & 0.8 & 0.2 & 0.3 \end{bmatrix}}_{W}
\underbrace{\begin{bmatrix} 10 \\ 150 \\ 7 \\ 232 \end{bmatrix}}_{x}
+
\underbrace{\begin{bmatrix} 0.2 \\ -1 \\ -0.9 \end{bmatrix}}_{b}
=
\underbrace{\begin{bmatrix} 96.4 \\ 212.8 \\ 181.1 \end{bmatrix}}_{f(x;\, W,b)}
$$

Modified from CS 231N @ Stanford

Linear Classification



Apparently, this is an unit used in deep neural networks too.

Today's agenda

- CSCI 599 overview
- Learning 101
- Course Entrance 1-1

Course Entrance 1-1

- If you talked me individually for any exception, e-mail me now (again even if you have done so).
- Others who need to talk with me, come to me after this lecture.

Todo

- Form your project team (use Piazza if needed)!

Questions?