Lecture 01

What Are Machine Learning And Deep Learning? An Overview.

STAT 453: Introduction to Deep Learning and Generative Models
Spring 2020

Sebastian Raschka
http://stat.wisc.edu/~sraschka/teaching/stat453-ss2020/

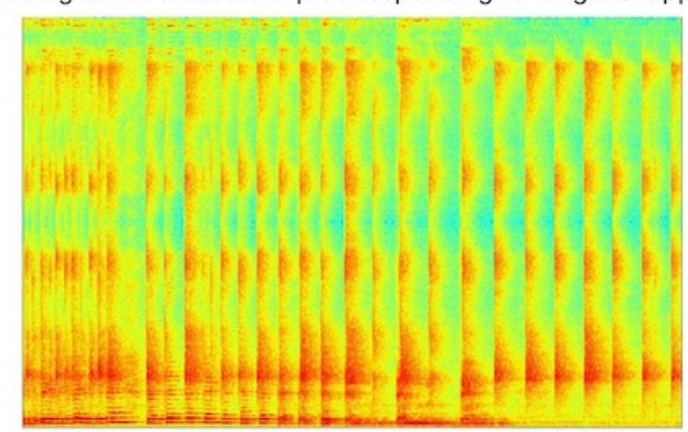
But first ...
a course overview

A short teaser: what you will be able to do after this course

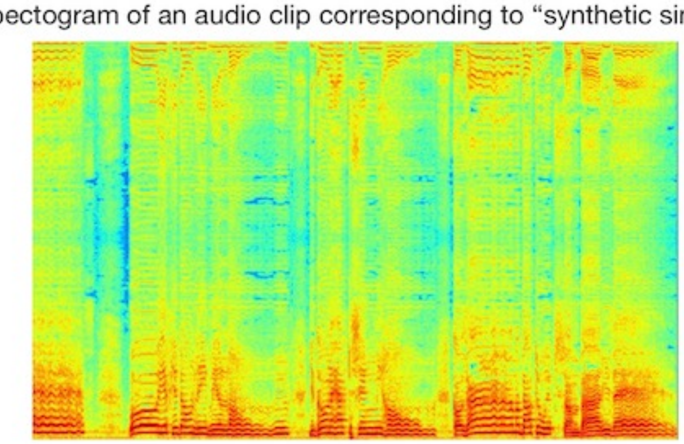
Audio Classification Using Convolutional Neural Networks

Poet Larsen, Reng Chiz Der, and Noah Haselow were exploring modern approaches for audio classification. In particular, Poet, Reng, and Noah used deep learning architectures that were originally developed for computer vision, namely convolutional neural networks (CNNs). The students implemented CNN architectures to capture temporal relationships in audio clips from spectrograms. They found that a Gated-CNN architecture was particularly helpful for mitigating vanishing gradient effects.

Spectogram of an audio clip corresponding to "finger snapping:"



Spectogram of an audio clip corresponding to "synthetic singing:"



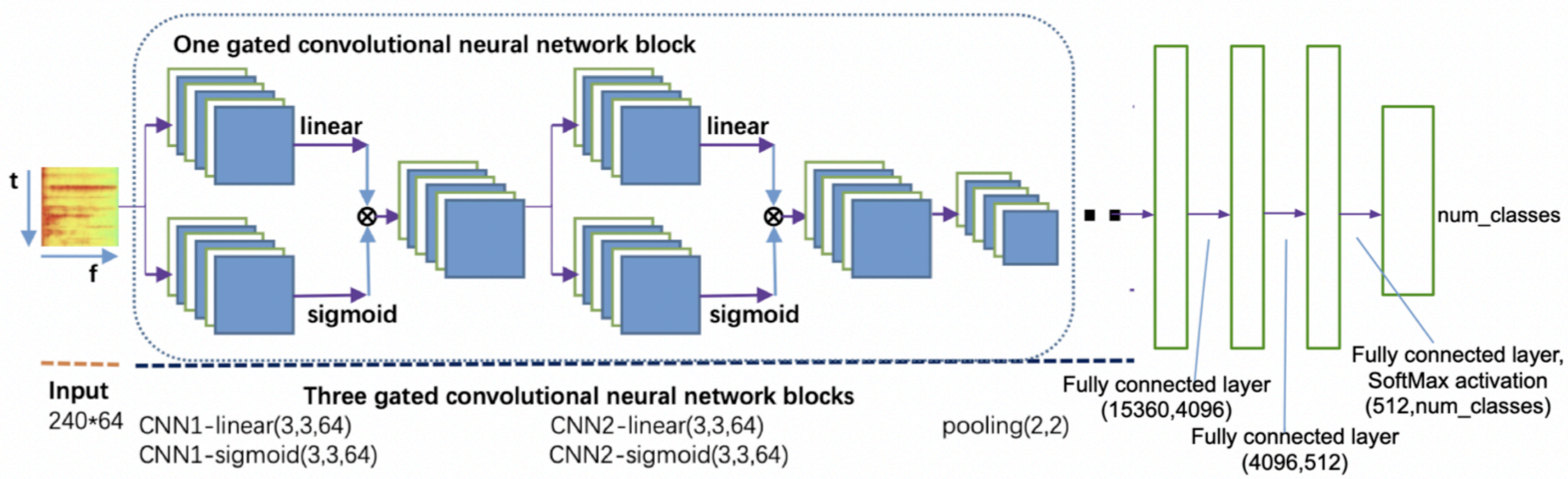
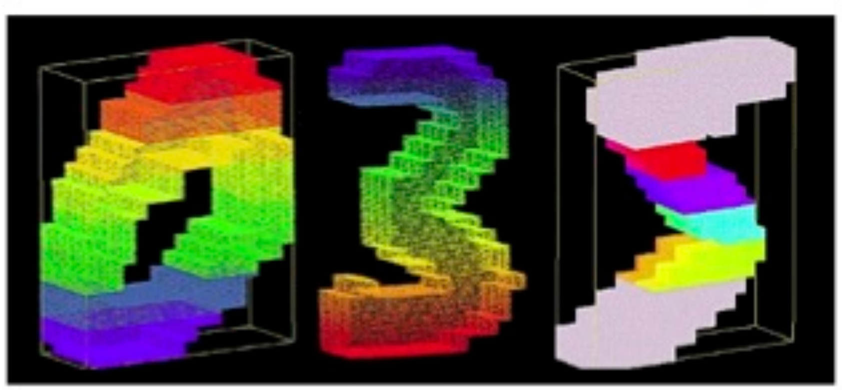
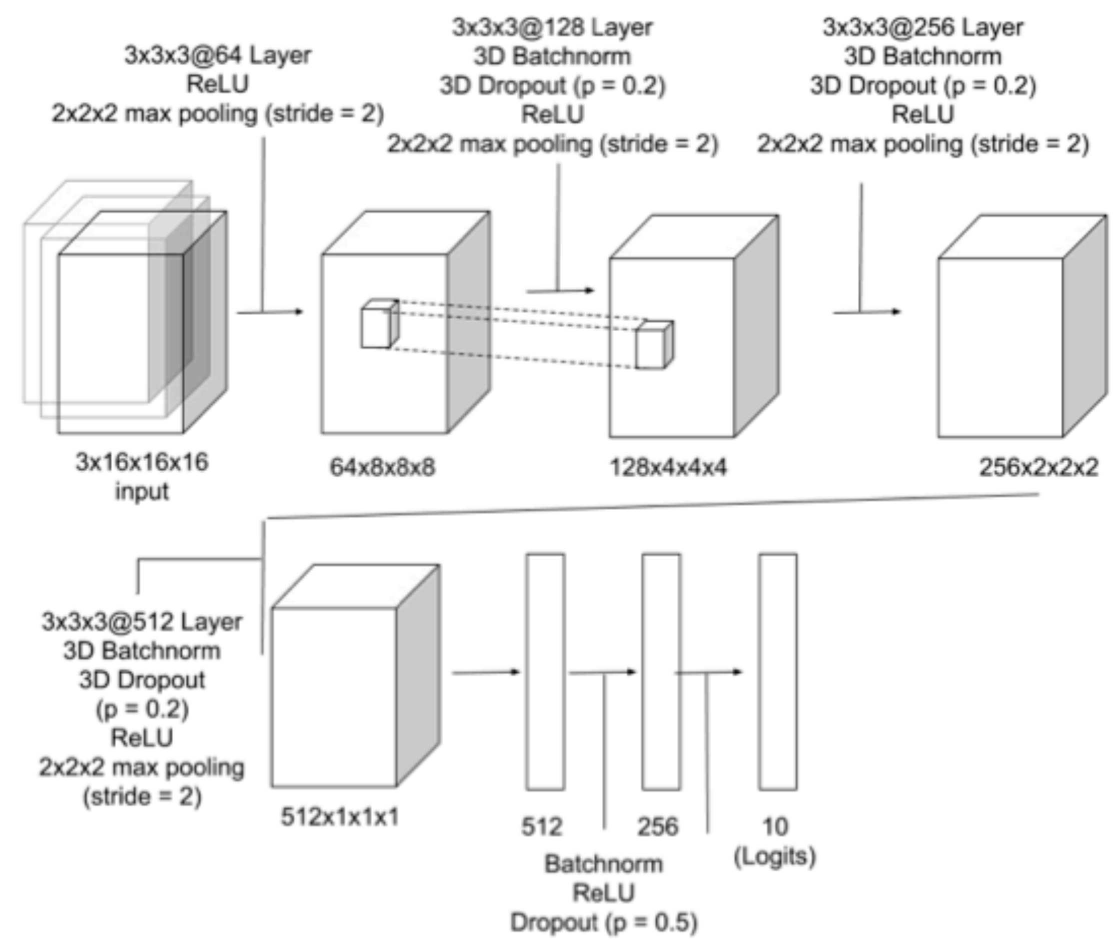


Figure 2. Gated Convolutional Neural Network with Multi-Layered Classification Model

3D Convolutional Networks

In this project, the students (Sam Berglin, Zheming Lian, and Jiahui Jiang) took the opportunity to explore 3D-convolutional neural networks a bit further, since we only covered them briefly in the lecture. While traditional photographs are inarguably more wide-spread, there are many domains where 3D-data analysis is especially useful (in contrast to 3D movies). For instance, applications involving spatially aware self-driving cars and advanced medicine (i.e., medical imaging involving CT or MRI) are areas that benefit from deep learning technology that leverages the 3D-structure of the data.





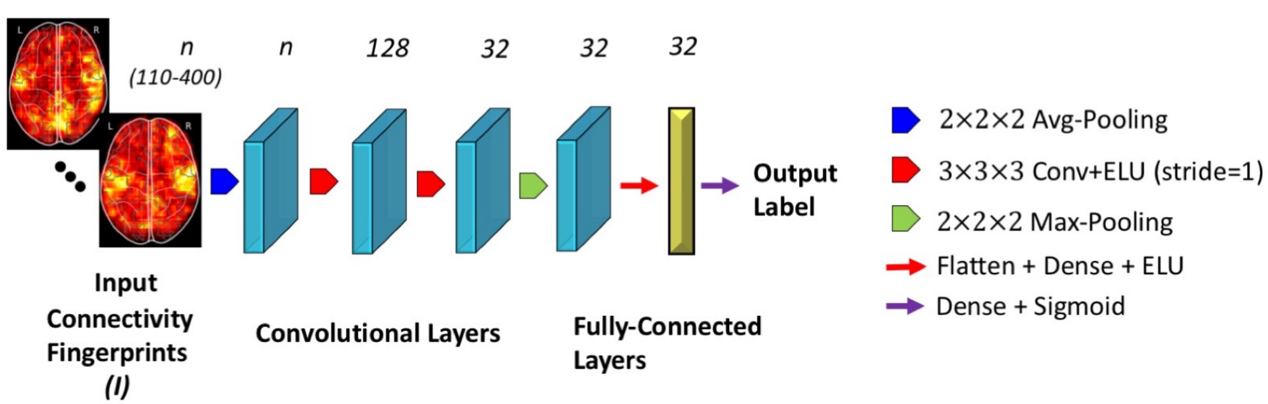
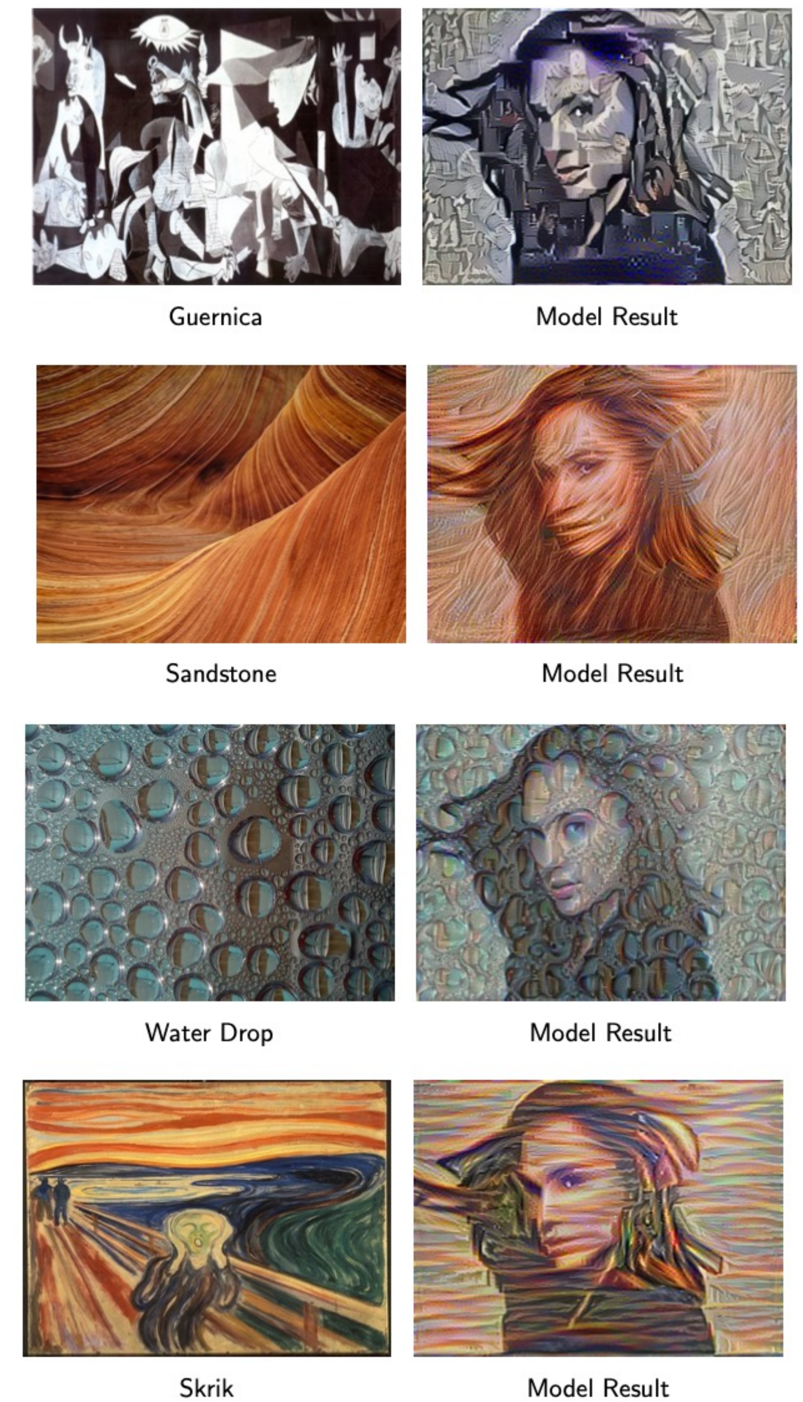


Image Source: 3D Convolutional Neural Networks for Classification of Functional Connectomes. https://arxiv.org/abs/1806.04209

Photographic Style Transfer With Deep Learning

For this part, after completing the face-matching subproject, the students (Lingfeng Zhu, Zhuoyan Xu, Cecily Liu) revisited their custom dataset consisting of 7500 images from Google Image Search – the dataset contains images of 15 movie actors. The style transfer approach, which the students outlined in more detail in their paper, yielded surprisingly nice results:



"It's no surprise that the artificial intelligence talent market is white-hot right now. The business value created by AI will reach $3.9 trillion in 2022, according to Gartner. IDC says worldwide spending on cognitive and artificial intelligence systems will reach $77.6 billion during the same year. And cognitive capabilities are poised to have an impact on nearly every corporate IT function."

"The number of AI jobs listed on Indeed from May 2018 to May 2019 grew 29 percent"

"Forty percent of respondents from Global 2000 organizations say they are adding more jobs as a result of AI adoption, according to a 2019 Dun & Bradstreet report"

"You will find a total of 7000 AI job openings in the United States, according to analysis by RPA firm UIPath."

Source: https://enterprisersproject.com/article/2019/8/ai-artificial-intelligence-careers-salaries-7-statistics

About the course

Topics Summary (Planned)

Part 1: Introduction

- Course overview, introduction to deep learning
- The brief history of deep learning
- Single-layer neural networks: The perceptron algorithm

Part 2: Mathematical and computational foundations

- Linear algebra and calculus for deep learning
- Parameter optimization with gradient descent
- Automatic differentiation
- Cluster and cloud computing resources

Part 3: Introduction to neural networks

- Multinomial logistic regression
- Multilayer perceptrons
- Regularization
- Input normalization and weight initialization
- Learning rates and advanced optimization algorithms
- Project proposal (online submission)

Part 4: Deep learning for computer vision and language modeling

- Introduction to convolutional neural networks 1
- Introduction to convolutional neural networks 2
- Introduction to recurrent neural networks 1
- Introduction to recurrent neural networks 2
- Midterm exam

Part 5: Deep generative models

- Autoencoders
- Autoregressive models
- Variational autoencoders
- Normalizing Flow Models
- Generative adversarial networks 1
- Generative adversarial networks 2
- Evaluating generative models

Part 6: Class projects and final exam

- Course summary
- Student project presentations 1
- Student project presentations 2
- Student project presentations 3
- Final exam
- Final report (online submission)

Course Website

http://pages.stat.wisc.edu/~sraschka/teaching/stat453-ss2020/

Course Material

- Field is relatively new and evolves quickly => No "good" textbook

- Main course material: Mainly slides
  + assigned reading (online references and papers)

- See course website for additional deep learning and Python suggestions

Course Logistics

When

- Tue 11:00-12:15 pm
- Thu 11:00-12:15 pm

Where

- SMI 331

Office Hours

- Prof. Sebastian Raschka:
  - Wed 3:05-4:05 pm, Room MSC 1171
- TA Zhongjie Yu:
  - TBD

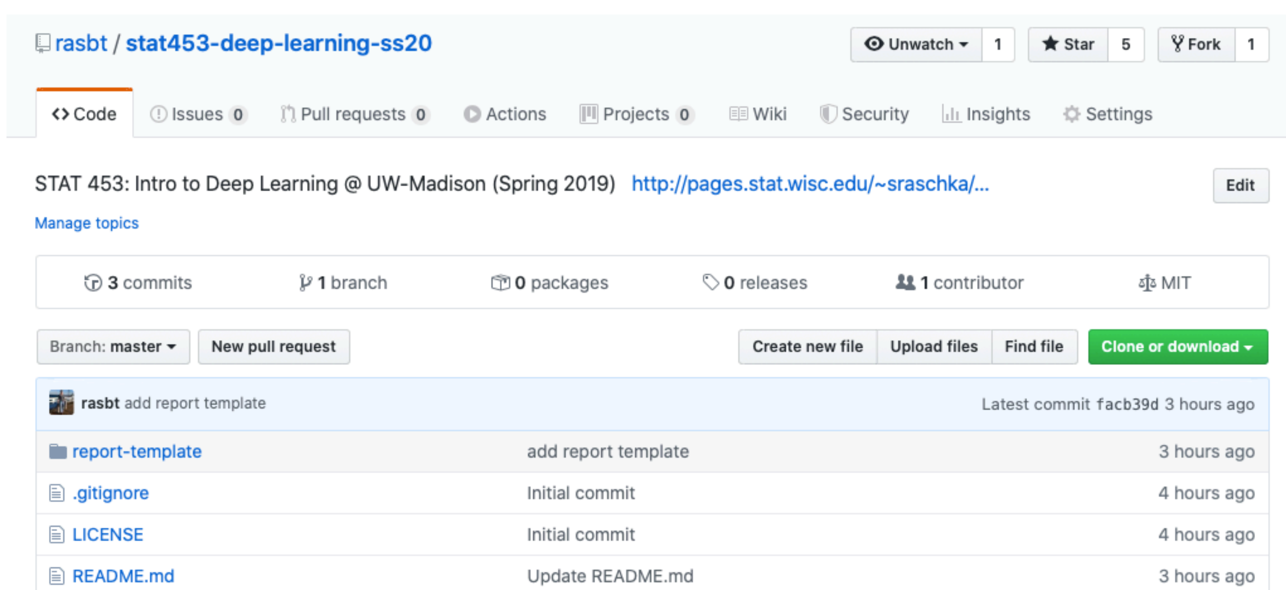
Important!

1) **Course Calendar:** Links to course material, info, etc., http://pages.stat.wisc.edu/~sraschka/teaching/stat453-ss2020/#calendar

Calendar

Date	Event	Description	Lecture Material	Announcements
Tue, Jan 21	Day 1			
Thu, Jan 23	Day 2			
Tue, Jan 28	Day 3			
Thu, Jan 30	Day 4			
Tue, Feb 04	Day 5			
Thu, Feb 06	Day 6			
Tue, Feb 11	Day 7			Deadline for group assignments
Thu, Feb 13	Day 8			
Tue, Feb 18	Day 9			

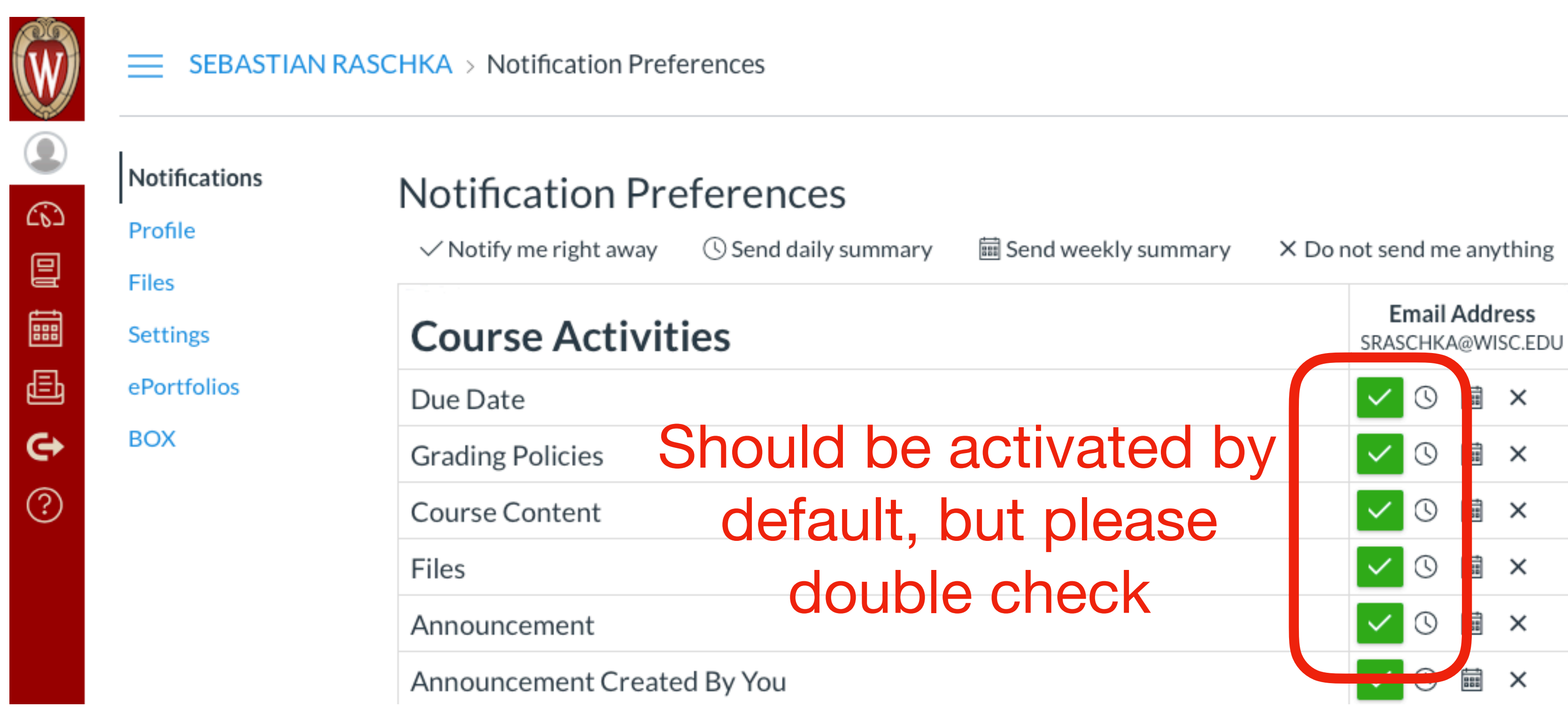
2) **Course material on GitHub:** https://github.com/rasbt/stat453-deep-learning-ss20



Important!

3) **Important info and announcements:** Canvas Announcements page

https://canvas.wisc.edu/courses/192139



Should be activated by default, but please double check

Grading

The final grade will be computed using the following weighted grading scheme:

- 20% Problem Sets
- 50% Exams:
  - 20% Midterm Exam
  - 30% Final Exam
- 30% Class Project:
  - 5% Project proposal
  - 10% Project presentation
  - 15% Project report

More about that later!

What You Will Learn Today

1/5 -- What Is Machine Learning?

2/5 -- The 3 Broad Categories of ML

3/5 -- Machine Learning Terminology and Notation

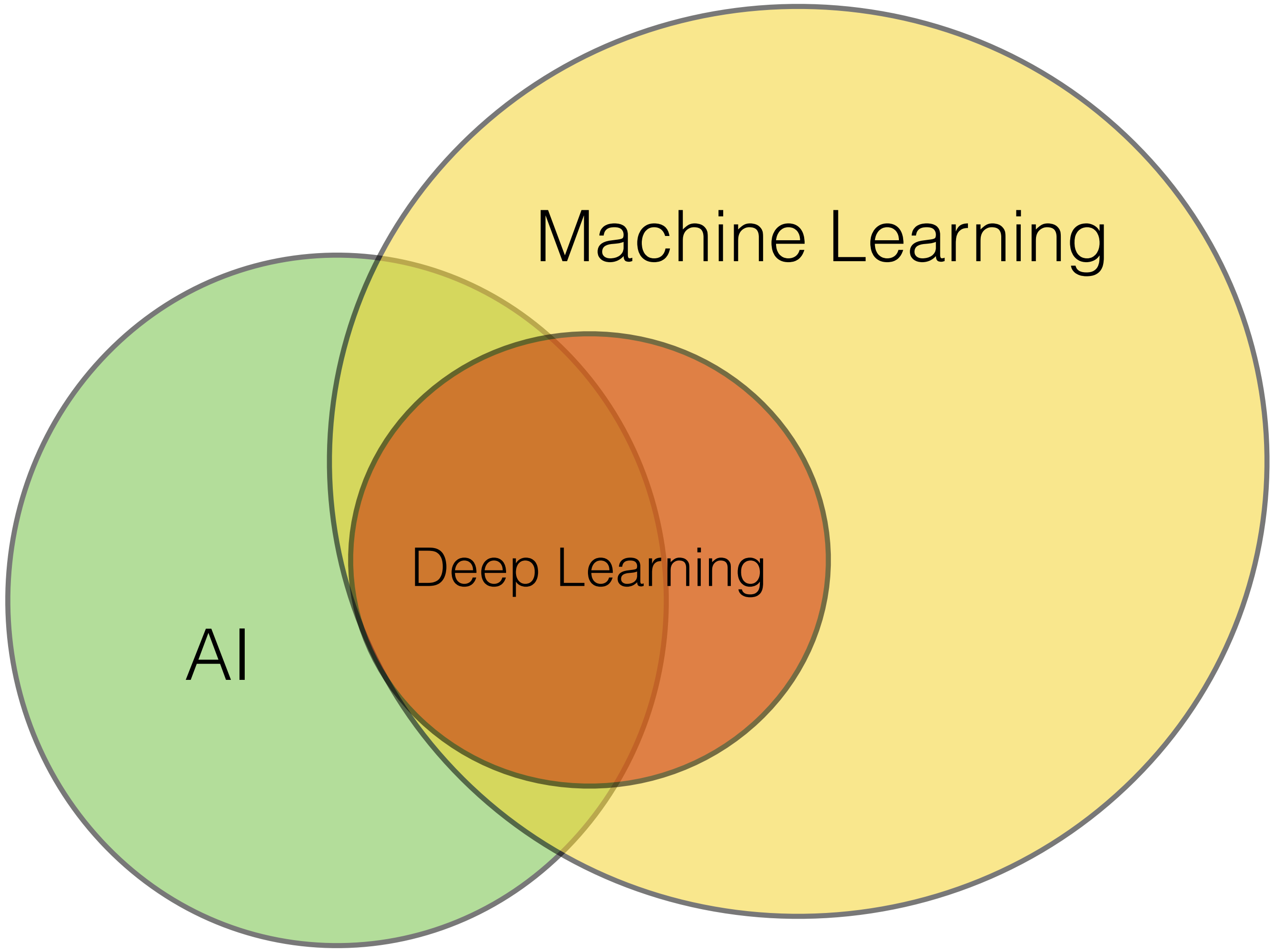
4/5 -- Machine Learning Modeling Pipeline

5/5 --The Practical Aspects: Our Tools!

What Is
Machine Learning?

A short overview before we jump into Deep Learning

The Connection Between Fields

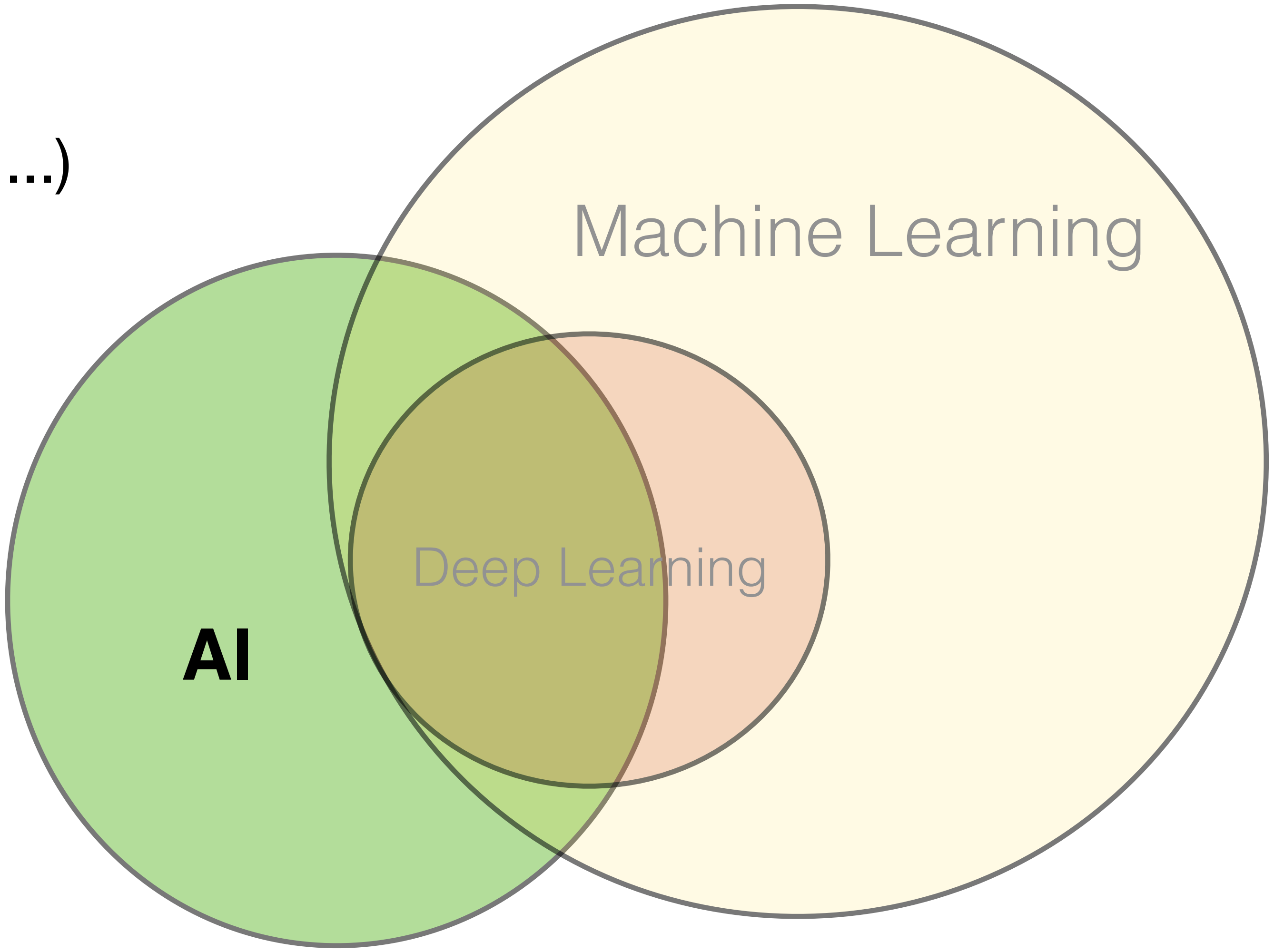


Different Types Of AI

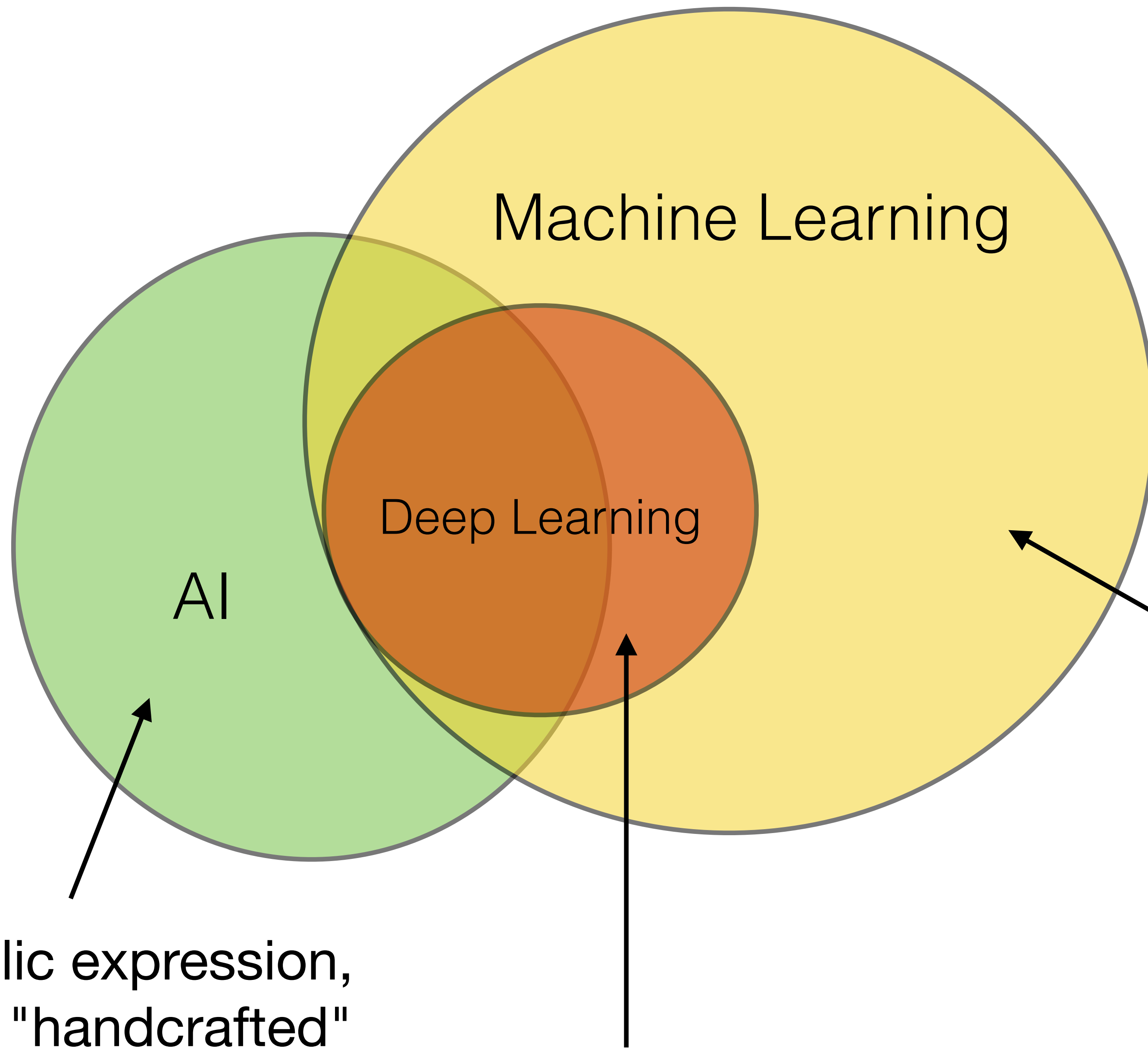
Artificial Intelligence (AI):
orig. subfield of computer science, solving tasks humans are good at (natural language, speech, image recognition, ...)

Artificial General Intelligence (AGI):
multi-purpose AI mimicking human intelligence across tasks

Narrow AI:
solving "a" task (playing a game, driving a car, ...)



What This Course Is About

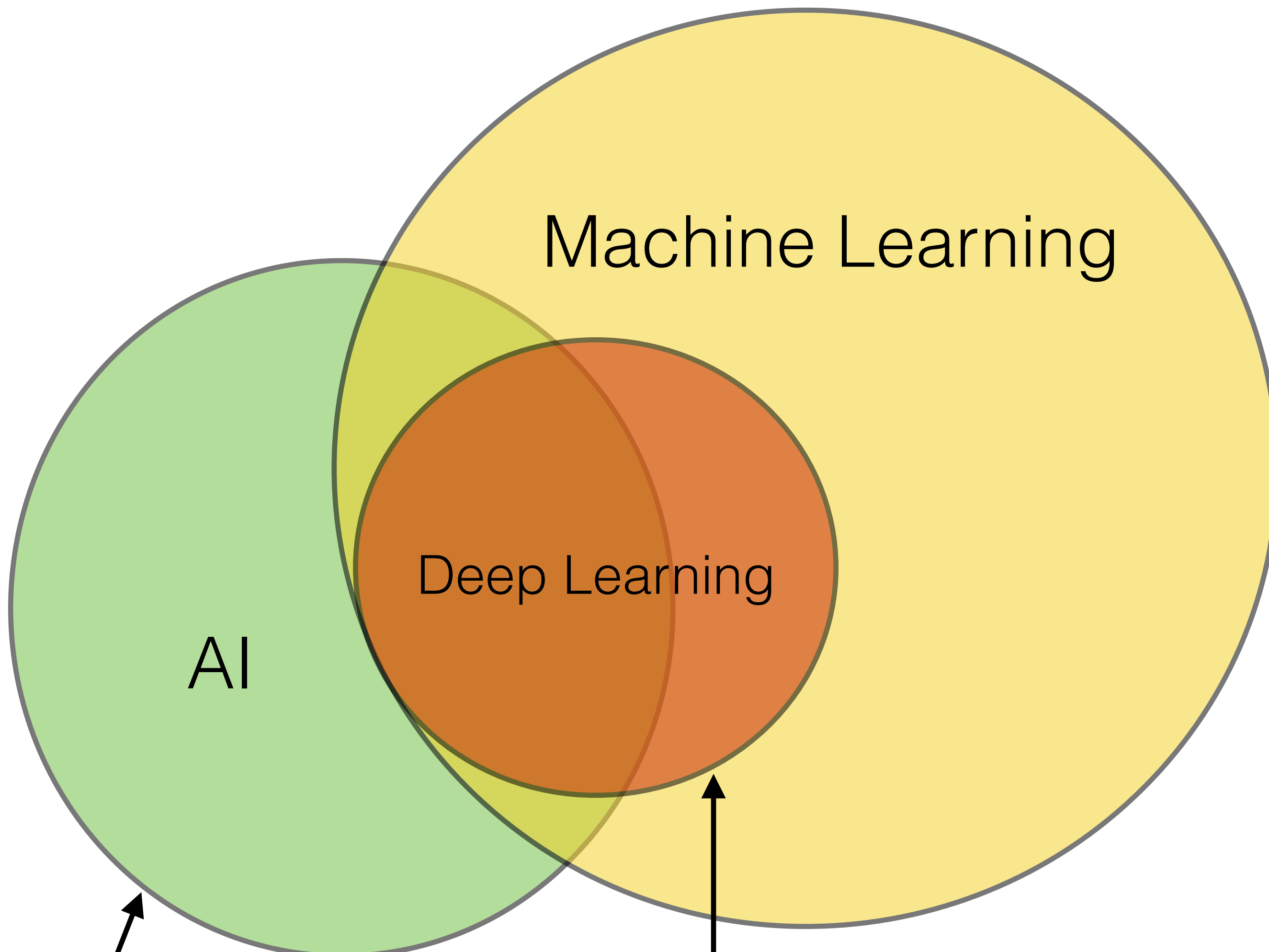


E.g., symbolic expression, logic rules / "handcrafted" nested if-else programming statements ...

Main focus of the course

E.g.,
generalized linear models,
tree-based methods,
"shallow" networks,
support vector machines,
nearest neighbors, ...

Examples From The Three Related "Areas"



= algorithms that learn models/representations/ rules automatically from data/examples

= a non-biological system that is intelligent through rules

= algorithms that parameterize multi-layers neural networks that then learn representations of data with multiple layers of abstraction

What Is Machine Learning?

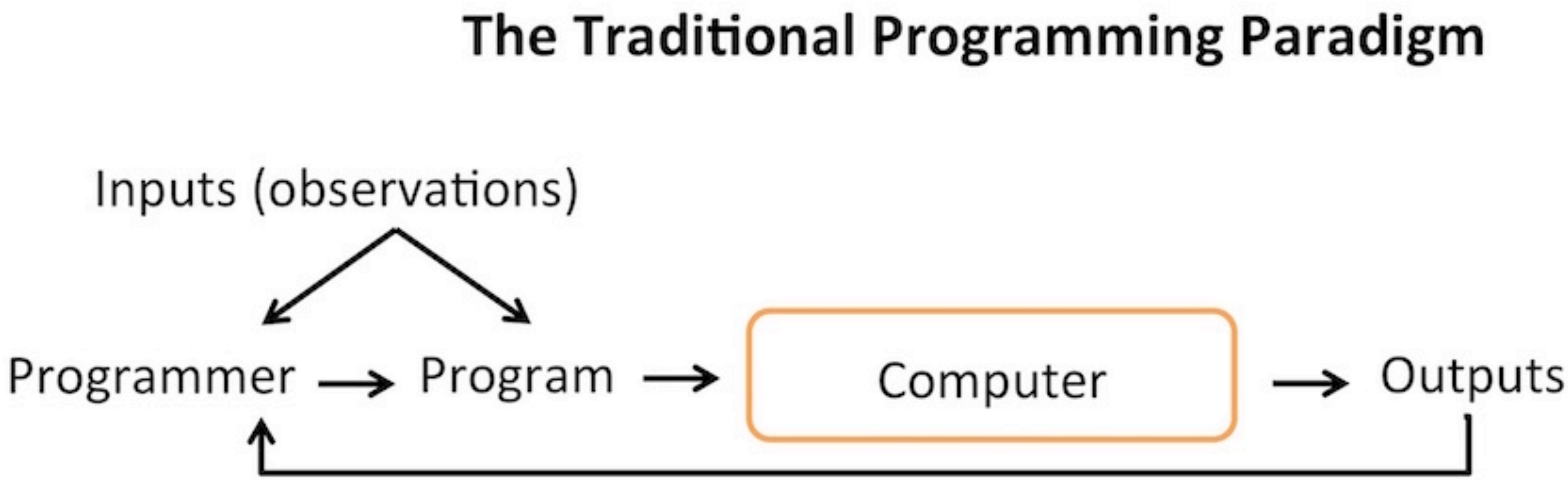
"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed"

— Arthur L. Samuel, AI pioneer, 1959

[probably the first, and undoubtedly the most popular definition]

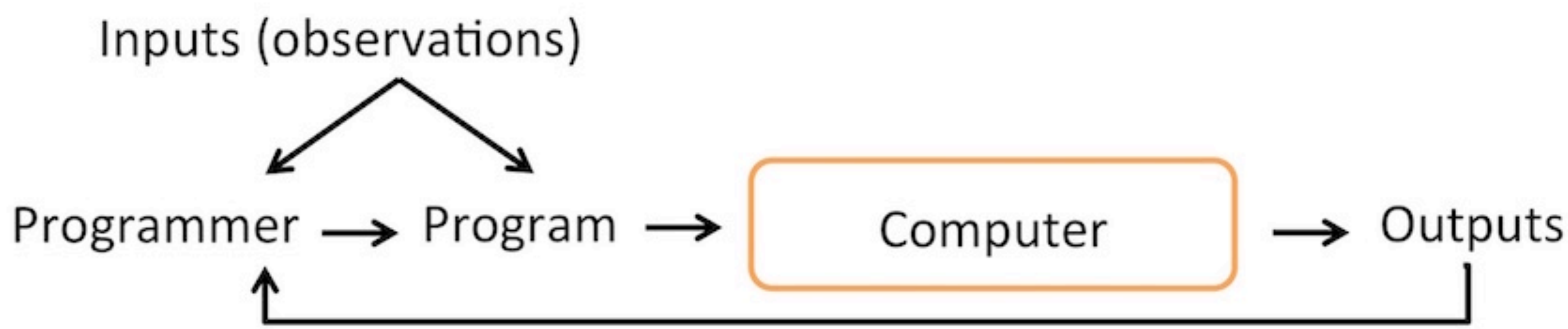
(This is likely not an original quote but a paraphrased version of Samuel's sentence "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.")

Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.



The Traditional Programming Paradigm
Inputs (observations)
Programmer
Program
Computer
Outputs

The Traditional Programming Paradigm



Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed
– Arthur Samuel (1959)

Machine Learning

Some Applications Of Machine Learning/Deep Learning

- Email spam detection
- Face detection and matching (e.g., iPhone X)
- Web search (e.g., DuckDuckGo, Bing, Google)
- Sports predictions
- Post office (e.g., sorting letters by zip codes)
- ATMs (e.g., reading checks)
- Credit card fraud
- Stock predictions

Some Applications Of Machine Learning/Deep Learning

- Smart assistants (Apple Siri, Amazon Alexa, ...)
- Product recommendations (e.g., Netflix, Amazon)
- Self-driving cars (e.g., Uber, Tesla)
- Language translation (Google translate)
- Sentiment analysis
- Drug design
- Medical diagnoses
- ...

The 3 Broad Categories of ML

(This also applies to DL)

The 3 Broad Categories Of ML (And DL)

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

Reinforcement Learning

- Decision process
- Reward system
- Learn series of actions

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

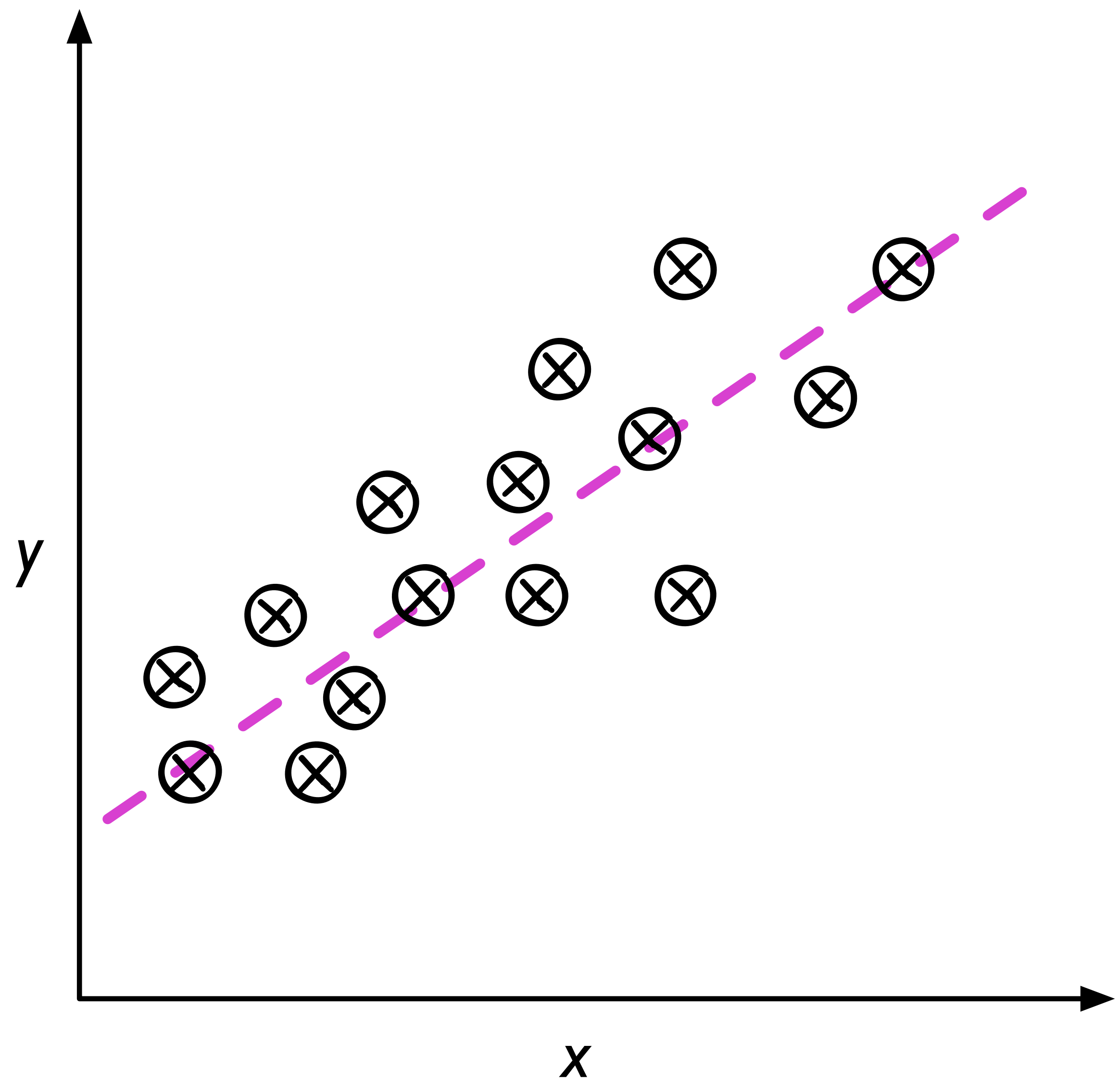
Supervised Learning Is The Largest Subcategory

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 1: Regression

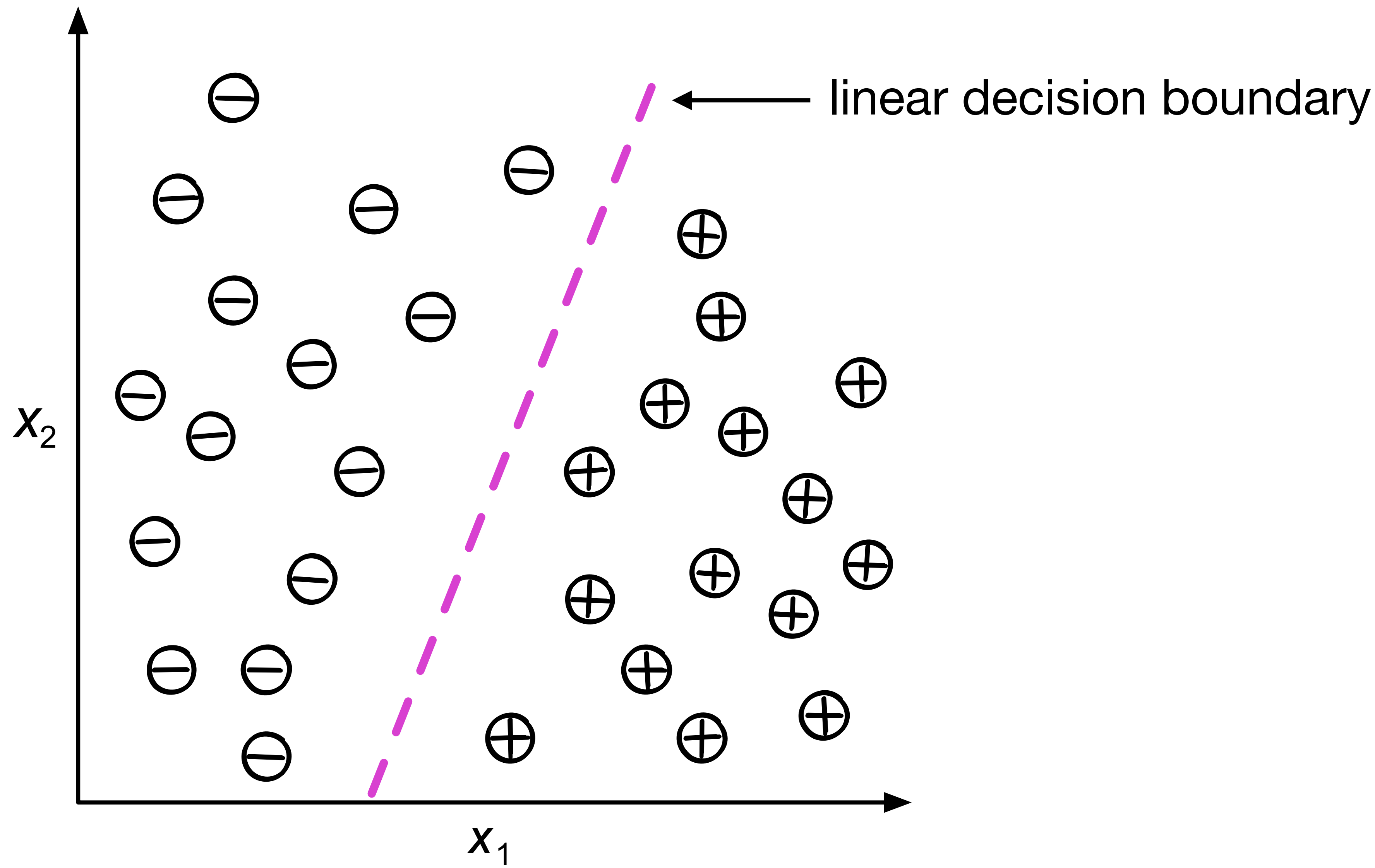


Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 2: Classification

Binary classification example with two *features* ("independent" variables, predictors)

What are the class labels (y's)?



Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Supervised Learning 3: Ordinal regression

- Ordinal regression also called *ordinal classification* or *ranking* (although ranking is a bit different)

Order dependence like in metric regression, but no metric distance

discrete values like in classification, but order dependence

$$r_K \succ r_{K-1} \succ \ldots \succ r_1$$

E.g., movie ratings: *great* $\succ$ *good* $\succ$ *okay* $\succ$ *for genre fans* $\succ$ *bad*

Supervised Learning 3: Ordinal regression

- **Ranking:** Correct order matters
  (0 loss if order is correct, e.g., rank a collection of movies by "goodness")

$\succ$

$\succ$

- **Ordinal regression:** Correct label matters
  (E.g., age of a person in years; here, regard aging as a non-stationary process)

Excerpt from the UTKFace dataset
https://susanqq.github.io/UTKFace/

18

$\prec$

29

$\prec$

41

The 2n Subcategory Of ML (And DL)

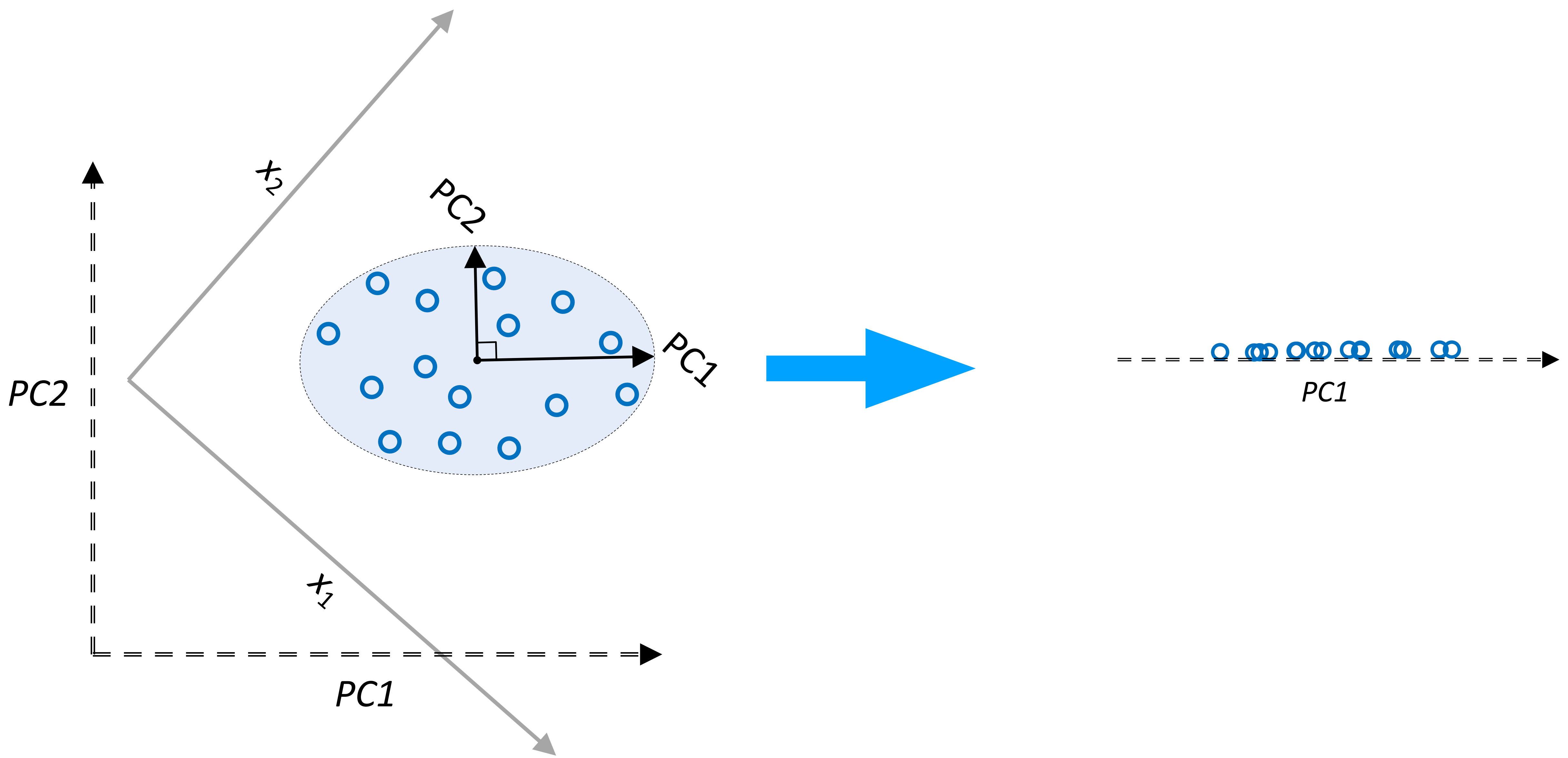
Unsupervised Learning

- No labels/targets
- No feedback
- Find hidden structure in data

Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

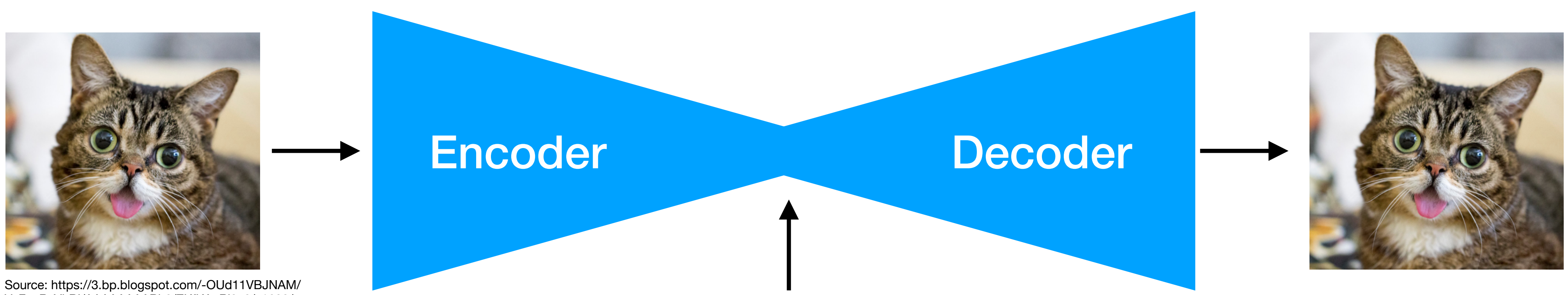
Unsupervised Learning 1: Representation Learning/Dimensionality Reduction

E.g., Principal Component Analysis (PCA)



Unsupervised Learning 1: Representation Learning/Dimensionality Reduction

E.g., Autoencoders

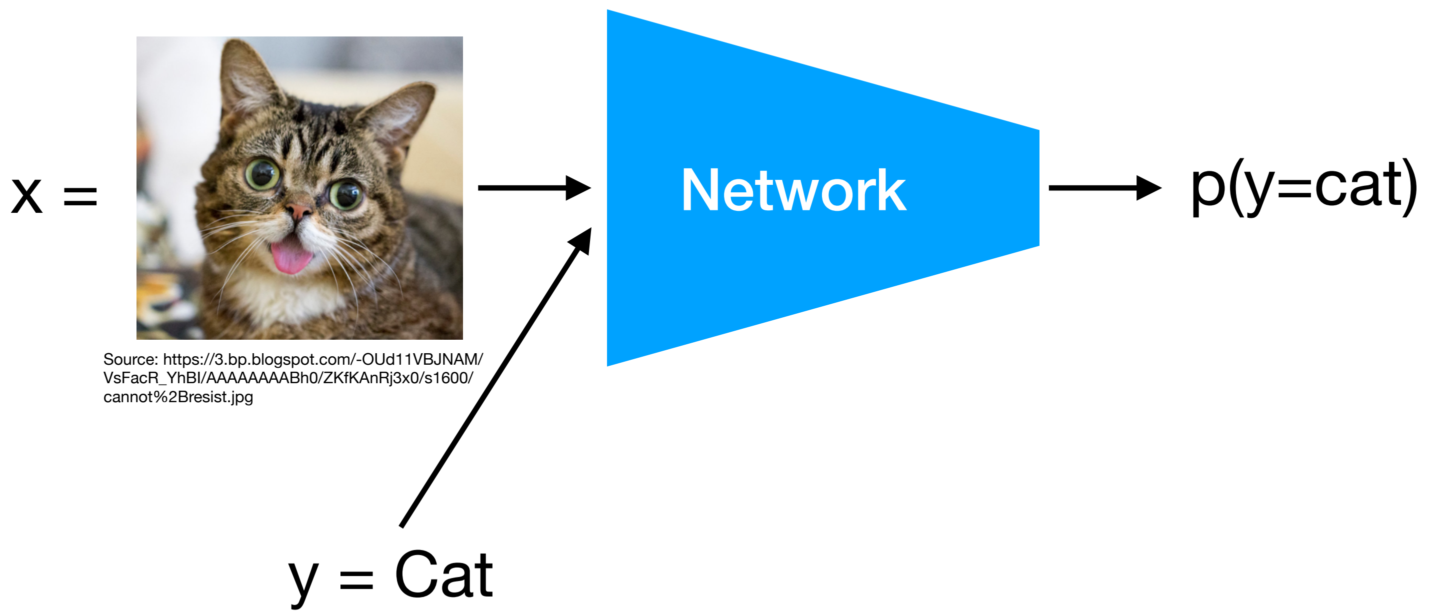


Source: https://3.bp.blogspot.com/-OUd11VBJNAM/VsFacR_YhBI/AAAAAAAABh0/ZKfKAnRj3x0/s1600/cannot%2Bresist.jpg

latent representation/ feature embedding

(covered later in this course)

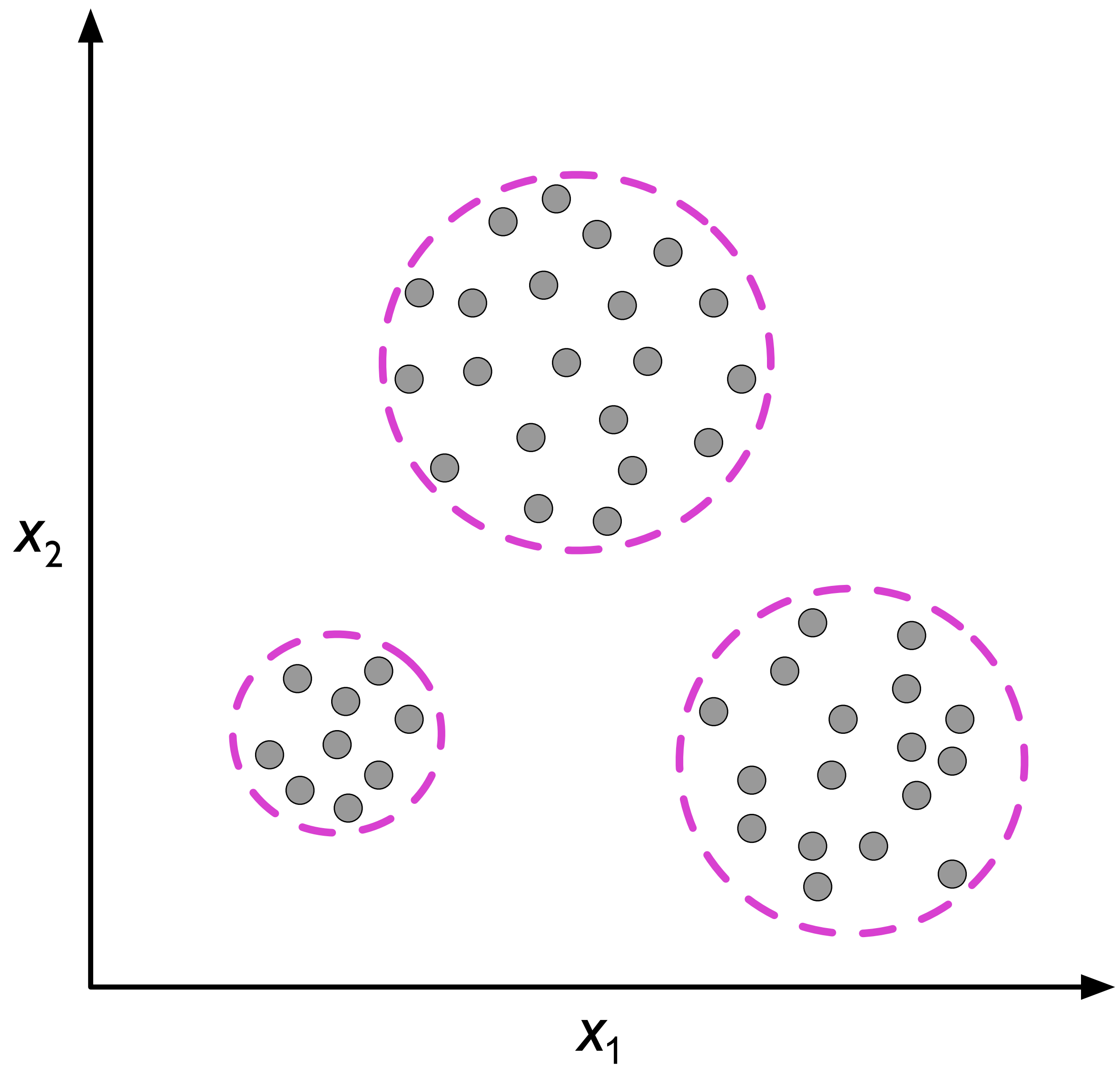
Reminder: Classification works like this



Source: https://3.bp.blogspot.com/-OUd11VBJNAM/VsFacR_YhBI/AAAAAAAABh0/ZKfKAnRj3x0/s1600/cannot%2Bresist.jpg

Unsupervised Learning 2: Clustering

Assigning group memberships to unlabelled examples (instances, data points)



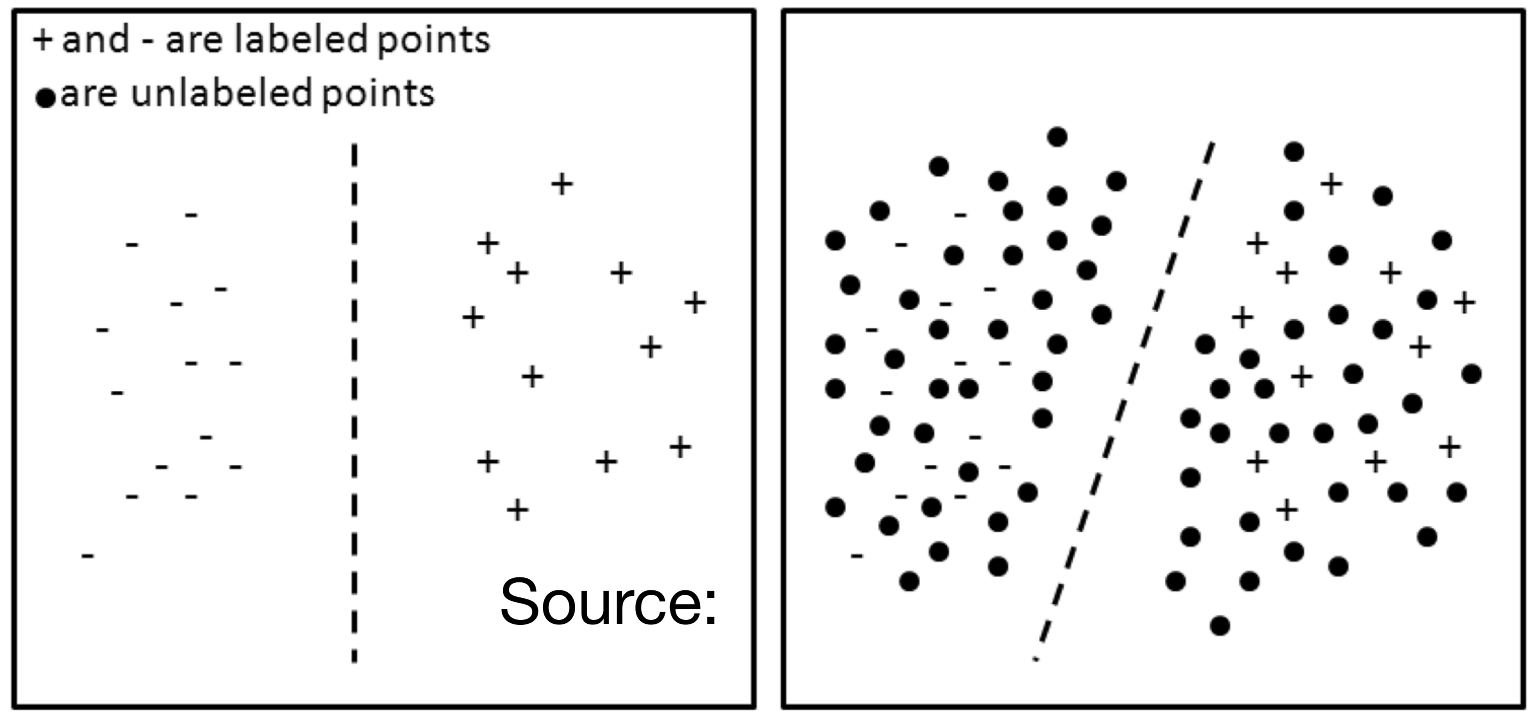
Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Semi-Supervised Learning

- mix between supervised and unsupervised learning
- some training examples contain outputs, but some do not
- use the labeled training subset to label the unlabeled portion of the training set, which we then also utilize for model training

Semi-Supervised Learning

www.nature.com/scientificreports/



(a) (b)

Figure 1. Semi-supervised learning tries to increase the generalization of classification performance by placing the decision boundary through the sparse regions in presence of both labeled and unlabeled data points. (**a**) The decision boundary in presence of labeled data points only, and (**b**) the decision boundary in presence of both labeled and unlabeled data.

In this paper, we present a semi-supervised learning method that analyzes groups of labeled and unlabeled points in multidimensional feature space in order to identify areas of high density and then guides the learning method to place decision boundaries through the regions with low density. We apply this technique to the analysis of digital pathology images of breast cancer.

Source: Peikari, M., Salama, S., Nofech-Mozes, S., & Martel, A. L. (2018). A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific reports*, *8*(1), 1-13.

Self-Supervised Learning

- A recent development and promising research trend in deep learning
- particularly useful if pre-trained models for transfer learning are not available for the target domain
- a process of deriving and utilizing label information directly from the data itself rather than having humans annotating it

Self-Supervised Learning

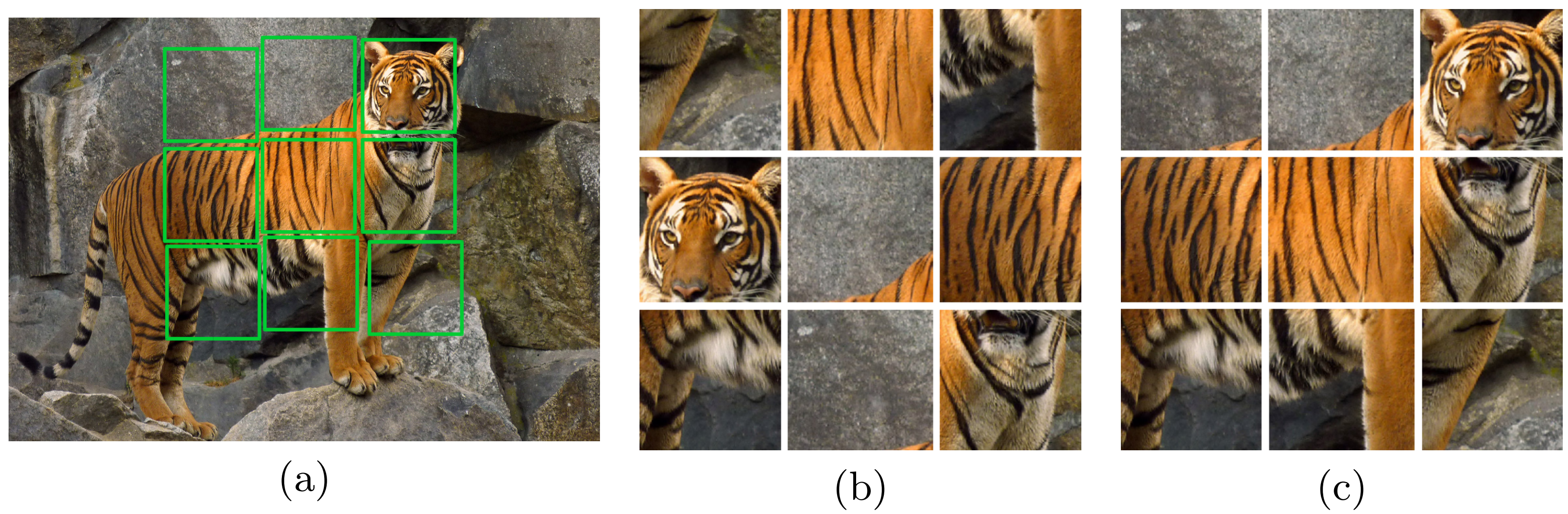


Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].

Source: Noroozi, Mehdi, and Paolo Favaro. "Unsupervised learning of visual representations by solving jigsaw puzzles." In *European Conference on Computer Vision*, pp. 69-84. Springer, Cham, 2016.

Reinforcement Learning: The third subcategory of ML (and DL)

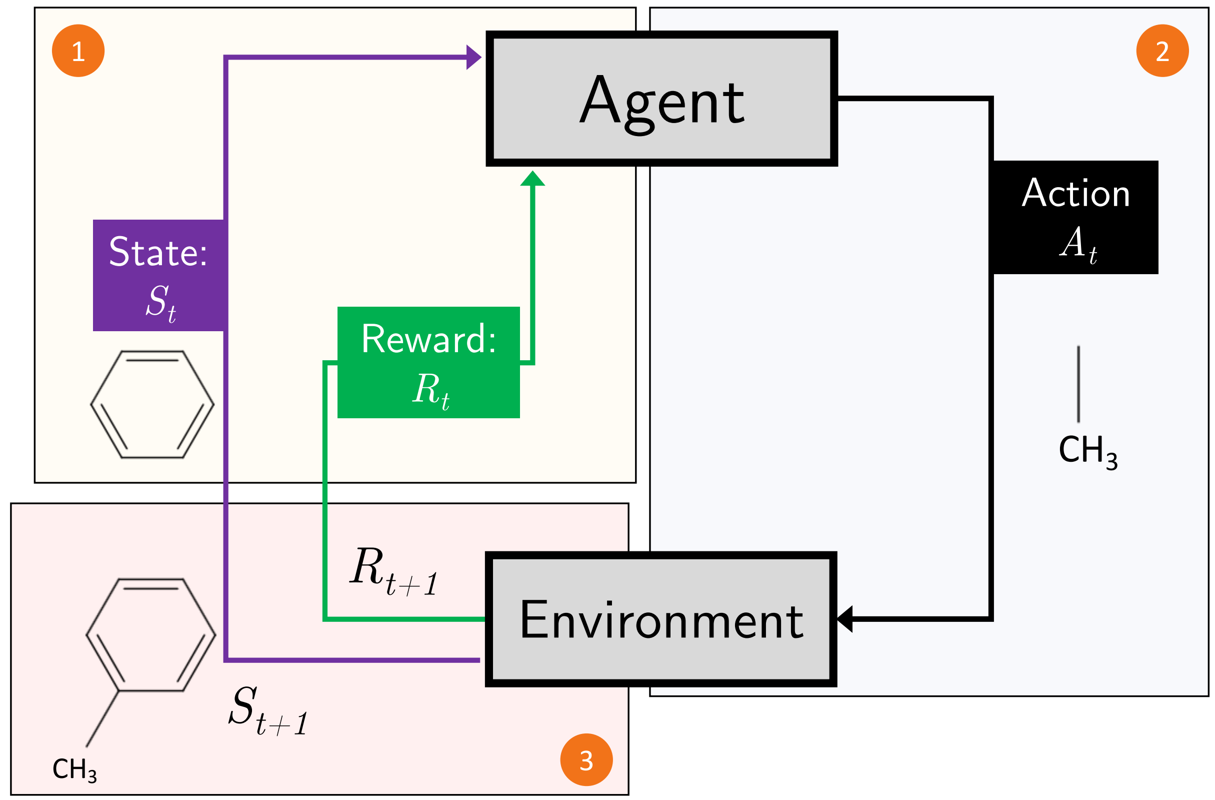


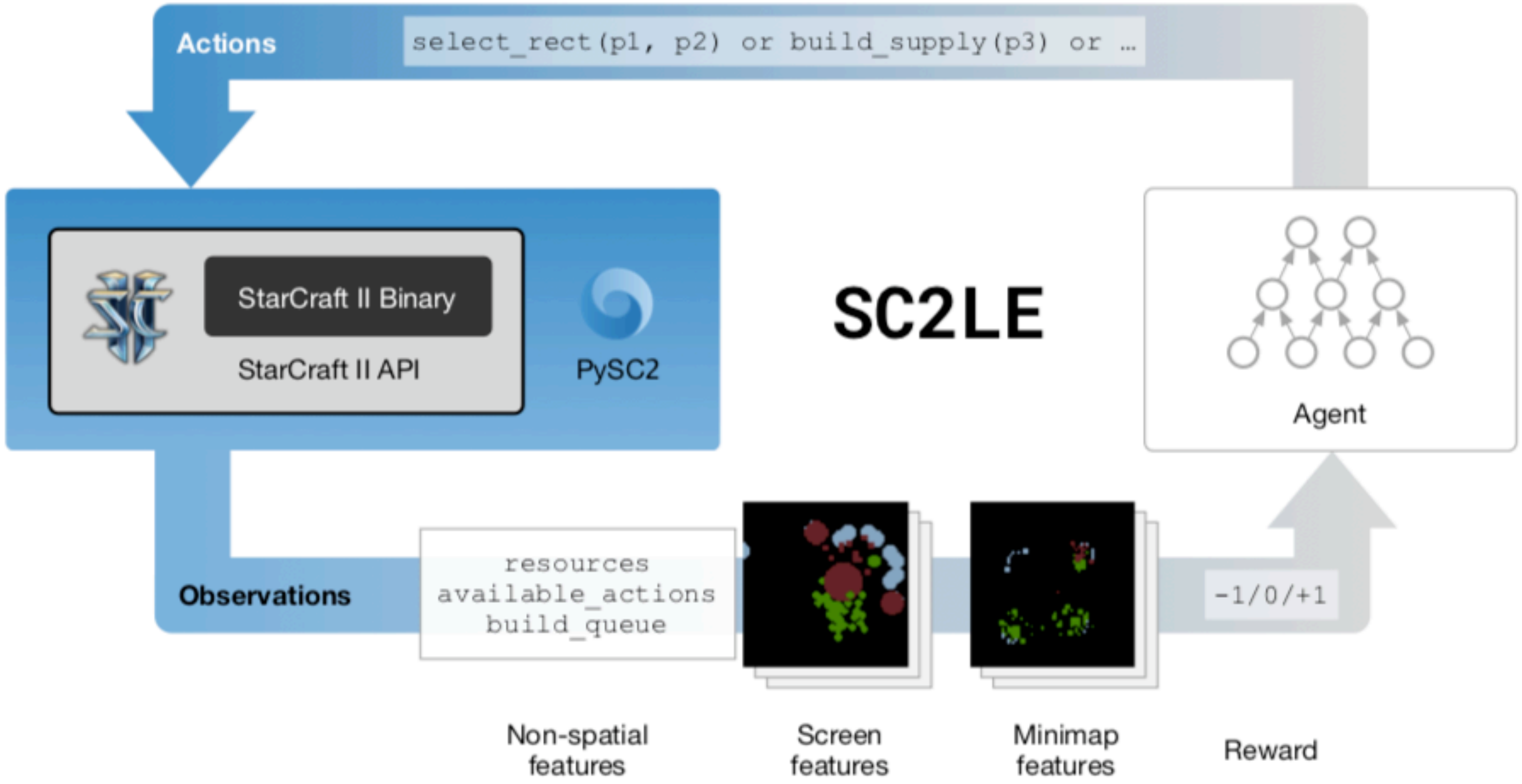
Figure 5: Representation of the basic reinforcement learning paradigm with a simple molecular example. (1) Given a benzene ring (state S_t at iteration t) and some reward value R_t at iteration t, (2) the agent selects an action A_t that adds a methyl group to the benzene ring. (3) The environment considers this information for producing the next state (S_{t+1}) and reward (R_{t+1}). This cycle repeats until the episode is terminated.

Source: Sebastian Raschka and Benjamin Kaufman (2020)
Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition

(Won't cover this in this course)

Reinforcement Learning: The third subcategory of ML (and DL)

Current state-of-the-art benchmark: StarCraft II



Vinyals, Oriol, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani et al. "Starcraft II: A new challenge for reinforcement learning." *arXiv preprint arXiv:1708.04782* (2017).

Machine Learning Terminology and Notation

(Again, this also applies to DL)

Machine Learning Jargon 1/2

- ***supervised learning:***
  learn function to map input *x* (features) to output *y* (targets)

- ***structured data:***
  databases, spreadsheets/csv files

- ***unstructured data:***
  features like image pixels, audio signals, text sentences
  (previous to DL, extensive feature engineering required)

Supervised Learning (More Formal Notation)

"training examples"

Training set: $\mathscr{D} = \{\langle \mathbf{x}^{[i]}, y^{[i]} \rangle, i = 1, \dots, n\},$

Unknown function: $f(\mathbf{x}) = y$

Hypothesis: $h(\mathbf{x}) = \hat{y}$

sometimes *t* or *o*

Classification

$h : \mathbb{R}^m \to \mathscr{Y}, \quad \mathscr{Y} = \{1, ..., k\}$

Regression

$h : \mathbb{R}^m \to \mathbb{R}$

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature vector

Data Representation

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Feature vector

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}$$

Design Matrix

$$\mathbf{X} = \begin{bmatrix} x_1^{[1]} & x_2^{[1]} & \cdots & x_m^{[1]} \\ x_1^{[2]} & x_2^{[2]} & \cdots & x_m^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{[n]} & x_2^{[n]} & \cdots & x_m^{[n]} \end{bmatrix}$$

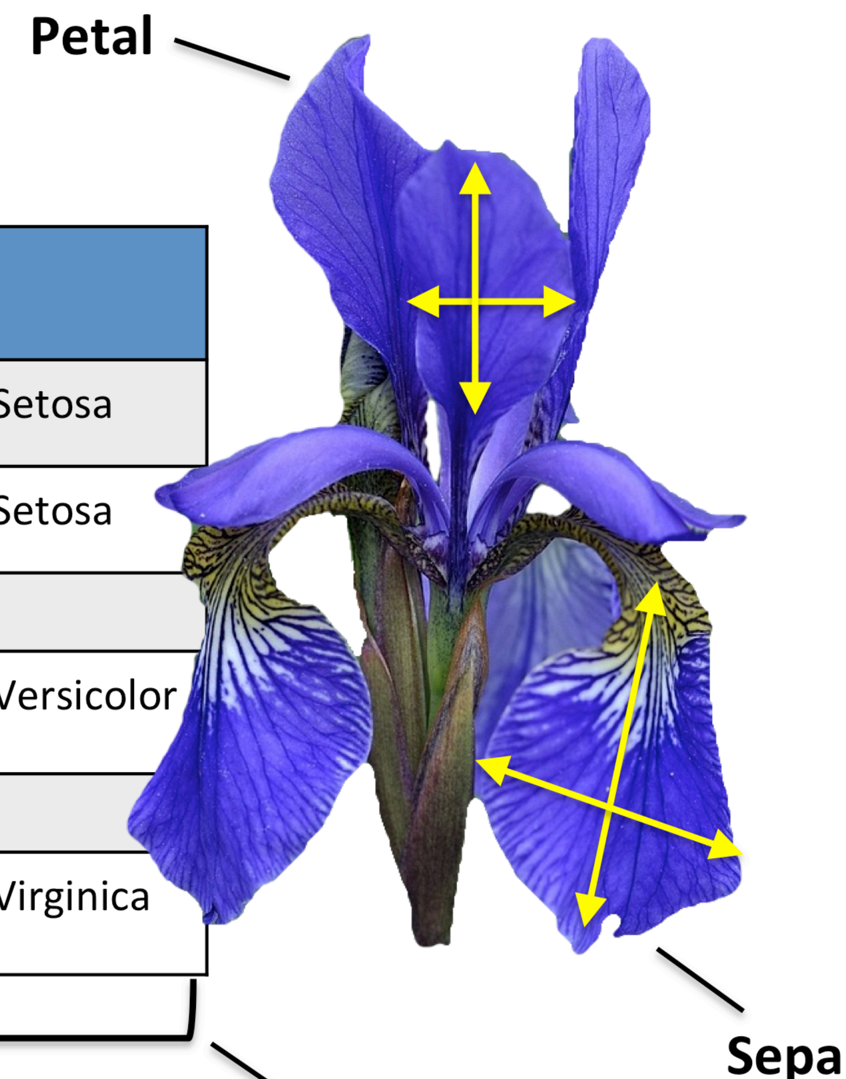
Design Matrix

Data Representation (structured data)

$m =$ ___

$n =$ ___

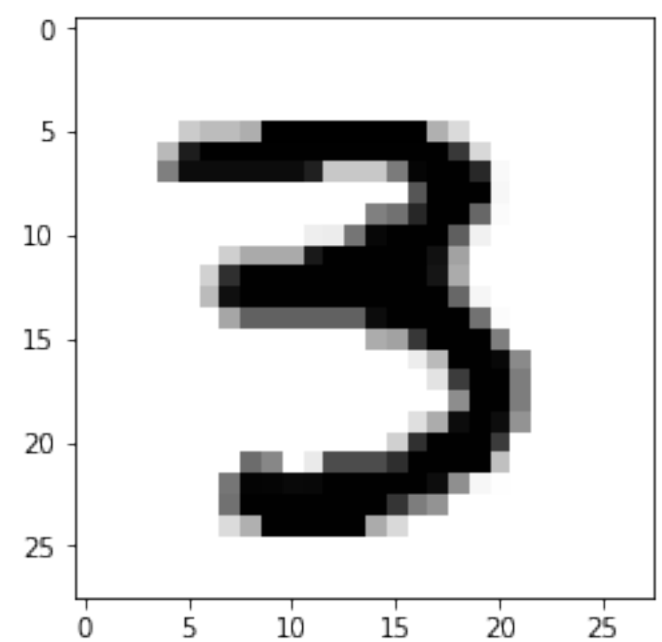
	Sepal length	Sepal width	Petal length	Petal width	
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Data Representation (unstructured data; images)

"traditional methods"

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.20392157, 0.2627451, 0.2627451, 0.31764707, 0.99607843, 0.99607843, 0.99607843
, 0.99607843, 1.0, 0.99607843, 0.99607843, 0.99607843, 0.30980393, 0.14509805, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.27450982, 0.8862745, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.9
9215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.7764706, 0.15294118, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5019608, 0.9529412, 0.9529412, 0.9529412, 0.9529412, 0.9529412,
0.9529412, 0.87058824, 0.21568628, 0.21568628, 0.21568628, 0.5176471, 0.98039216, 0.99215686, 0.99215686, 0.8392157
, 0.023529412, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.6627451, 0.99215686, 0.99215686, 0.99215686, 0.03137255, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.49803922, 0.5529412, 0.84705883, 0.99215686, 0.99
215686, 0.59607846, 0.015686275, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.06666667, 0.07450981, 0.5411765, 0.972549, 0.99215686, 0.99215686, 0.99215686, 0.627451, 0.05490196, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.1882353, 0.33333334, 0.33333334, 0.33333334
, 0.8980392, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.92941177, 0.3647059, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.18039216, 0.8156863, 0.99215686, 0.99215686, 0.99215686, 0
.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.91764706, 0.32941177, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.26666668, 0.9411765, 0.99215686, 0.99215686, 0.99215686, 0
.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.59607846, 0.03529412, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.34901962, 0.61960787, 0.61960787, 0.61960787,
0.61960787, 0.61960787, 0.61960787, 0.9529412, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.54901963, 0.011764
706, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.325
4902, 0.3647059, 0.78431374, 0.99215686, 0.99215686, 0.99215686, 0.5019608, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.07058824, 0.27058825, 0.99215686, 0.99
215686, 0.96862745, 0.45882353, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.10980392, 0.7647059, 0.99215686, 0.99215686, 0.50980395, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.44705883, 0.992156
86, 0.99215686, 0.50980395, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.1254902, 0.3254902, 0.9529412, 0.99215686, 0.9490196, 0.41960785, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.18431373, 0.8117647, 0.99215686,
0.99215686, 0.99215686, 0.76862746, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.57
64706, 0.47058824, 0.0, 0.08235294, 0.7019608, 0.7019608, 0.7019608, 0.81960785, 0.99215686, 0.99215686, 0.99215686
, 0.99215686, 0.24705882, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5372549, 0.984313
7, 0.98039216, 0.9607843, 0.9647059, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.9490196, 0.43921
57, 0.03137255, 0.003921569, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5529412, 0.992
15686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.7921569, 0.50980395, 0.41960785, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.14117648, 0.30980393, 0.99215
686, 0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.32941177, 0.14901961, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```



Data Representation (unstructured data; images)

Convolutional Neural Networks

```
Image batch dimensions: torch.Size([128, 1, 28, 28])
Image label dimensions: torch.Size([128])
```

"NCHW" representation (more on that later)

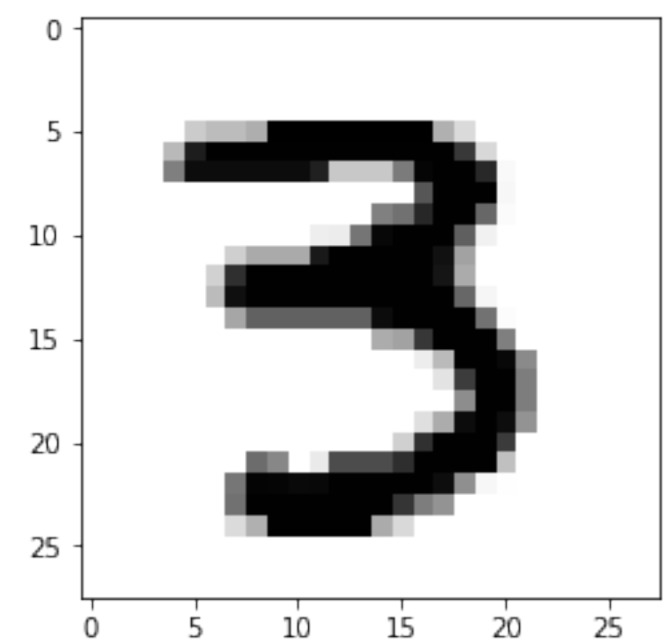
```
print(images[0].size())
```

```
torch.Size([1, 28, 28])
```

```
images[0]
```

```
tensor([[[0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],

         [0.0000, 0.0000, 0.0000, 0.0000, 0.5020, 0.9529, 0.9529, 0.9529,
          0.9529, 0.9529, 0.9529, 0.8706, 0.2157, 0.2157, 0.2157, 0.5176,
          0.9804, 0.9922, 0.9922, 0.8392, 0.0235, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.6627, 0.9922, 0.9922, 0.9922, 0.0314, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.4980, 0.5529,
          0.8471, 0.9922, 0.9922, 0.5961, 0.0157, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0000],
         [0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
          0.0000, 0.0000, 0.0000, 0.0667, 0.0745, 0.5412, 0.9725, 0.9922,
```



Machine Learning Jargon 2/2

- **Training example**, synonymous to
observation, training record, training instance, training sample (in some contexts, sample refers to a collection of training examples)
- **Feature**, synonymous to
predictor, variable, independent variable, input, attribute, covariate
- **Target**, synonymous to
outcome, ground truth, output, response variable, dependent variable, (class) label (in classification)
- **Output / Prediction**, use this to distinguish from targets; here, means output from the model

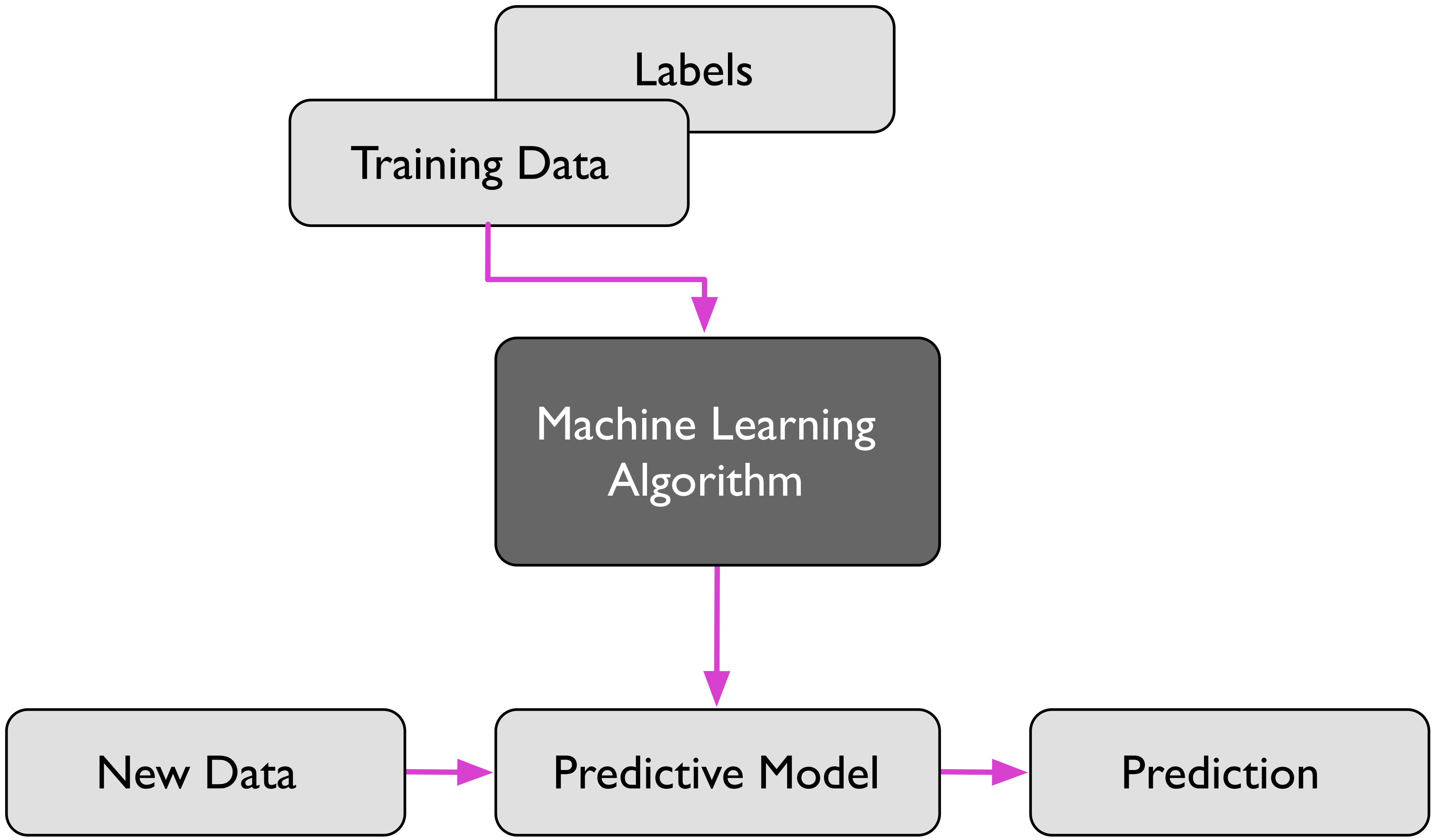
- use loss L for a single training example
- use cost C for the average loss over the training set
- use $\phi(\cdot)$, unless noted otherwise, for the activation function

(will make more sense later)

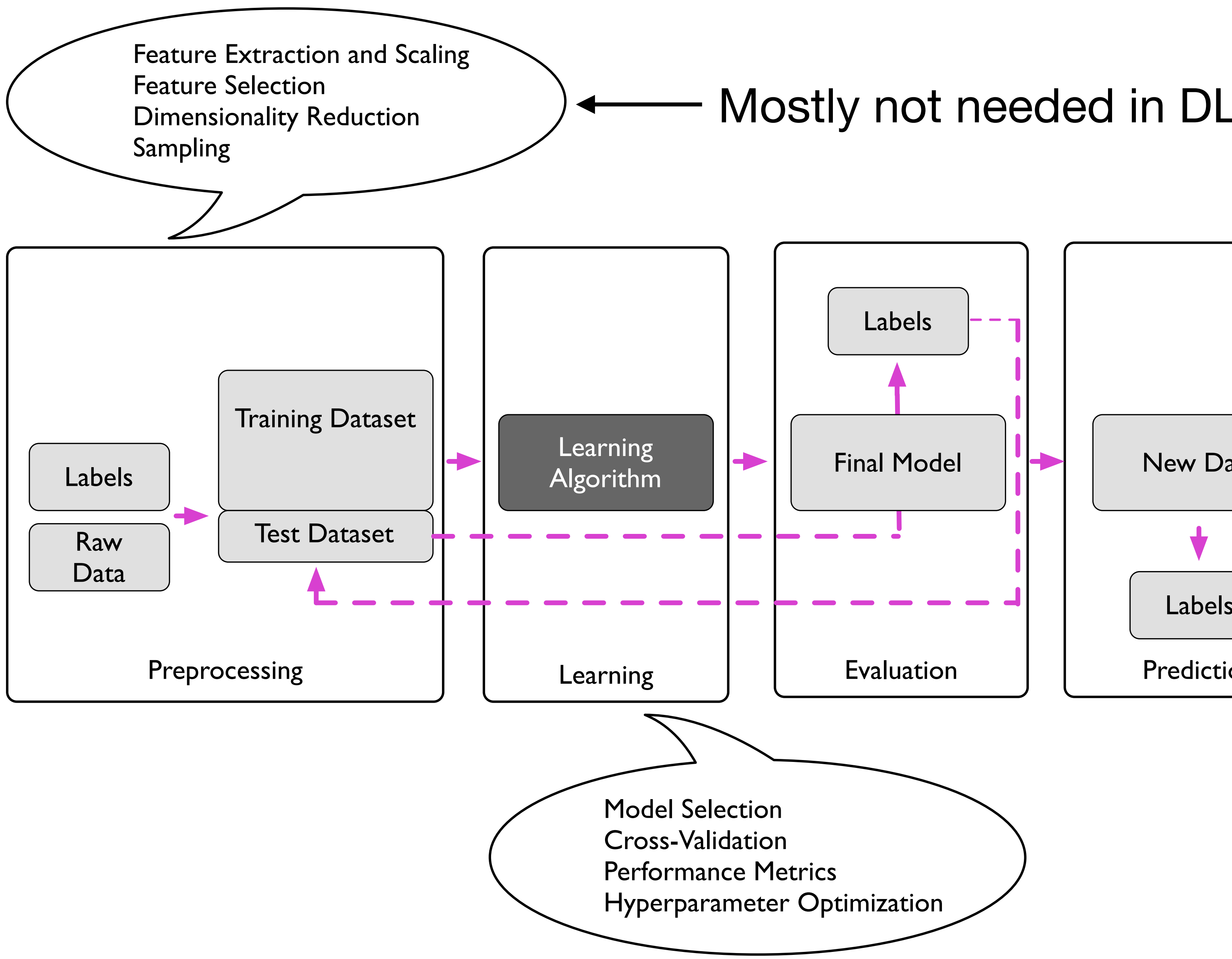
Machine Learning Modeling Pipeline

(Like before, this also applies to DL)

Supervised Learning Workflow



Supervised Learning Workflow (more detailed)



Source: Raschka and Mirjalily (2019). *Python Machine Learning, 3rd Edition*

Hypothesis Space

Entire hypothesis space

Hypothesis space
a particular learning
algorithm category
has access to

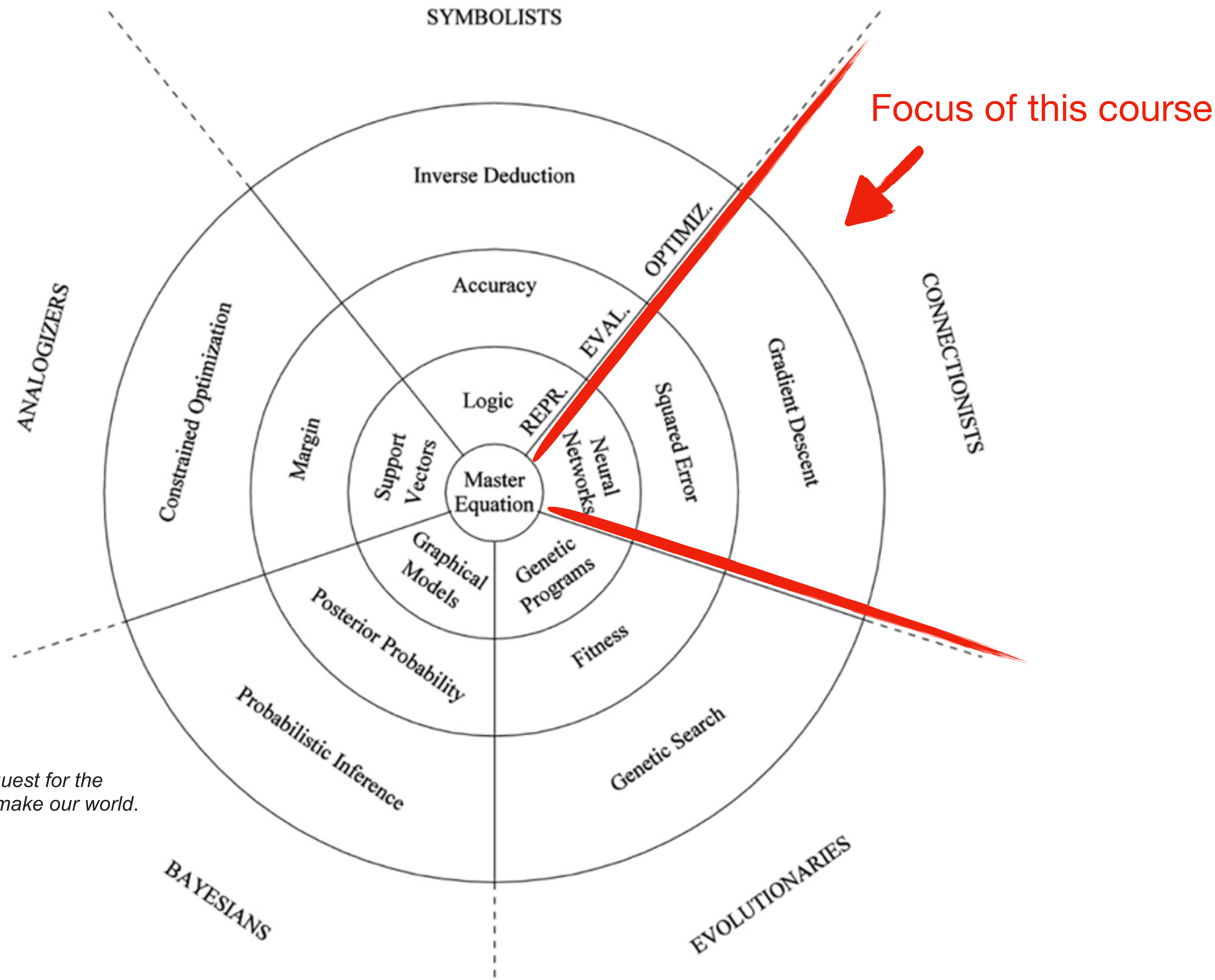
Hypothesis space
a particular learning
algorithm can sample

Particular hypothesis
(i.e., a model/classifier)

5 Steps for Approaching an ML/DL Problem

1. Define the problem to be solved.
2. Collect (labeled) data.
3. Choose an algorithm class.
4. Choose an optimization metric for learning the model.
5. Choose a metric for evaluating the model.

Pedro Domingo's 5 Tribes of Machine Learning



Source: Domingos, Pedro.
The Master Algorithm: How the quest for the ultimate learning machine will remake our world.
Basic Books, 2015.

Learning = Representation + Evaluation + Optimization

(Pedro Domingos, *A Few Useful Things to Know about Machine Learning* https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf)

Objective Functions / Surrogate Risk/Loss

- Maximize the posterior probabilities (e.g., naive Bayes)
- Maximize a fitness function (genetic programming)
- Maximize the total reward/value function (reinforcement learning)
- Maximize information gain/minimize child node impurities (CART decision tree classification)
- Minimize a mean squared error cost (or loss) function (CART, decision tree regression, linear regression, adaptive linear neurons, ...)
- Maximize log-likelihood or minimize cross-entropy loss (or cost) function
- Minimize hinge loss (support vector machine)

Optimization Methods

- Combinatorial search, greedy search (e.g., decision trees)
- Unconstrained convex optimization (e.g., logistic regression)
- Constrained convex optimization (e.g., SVM)
- Nonconvex optimization, here: using backpropagation, chain rule, reverse autodiff. (e.g., neural networks)
- Constrained nonconvex optimization (e.g., semi-adversarial nets)

0/1 Loss, Misclassification Error

$$L(\hat{y}, y) = \begin{cases} 0 & \textbf{if } \hat{y} = y \\ 1 & \textbf{if } \hat{y} \neq y \end{cases}$$

$$ERR_{\mathscr{D}^{\textbf{test}}} = \frac{1}{n} \sum_{i=1}^{n} L\left(\hat{y}^{[i]}, y^{[i]}\right)$$

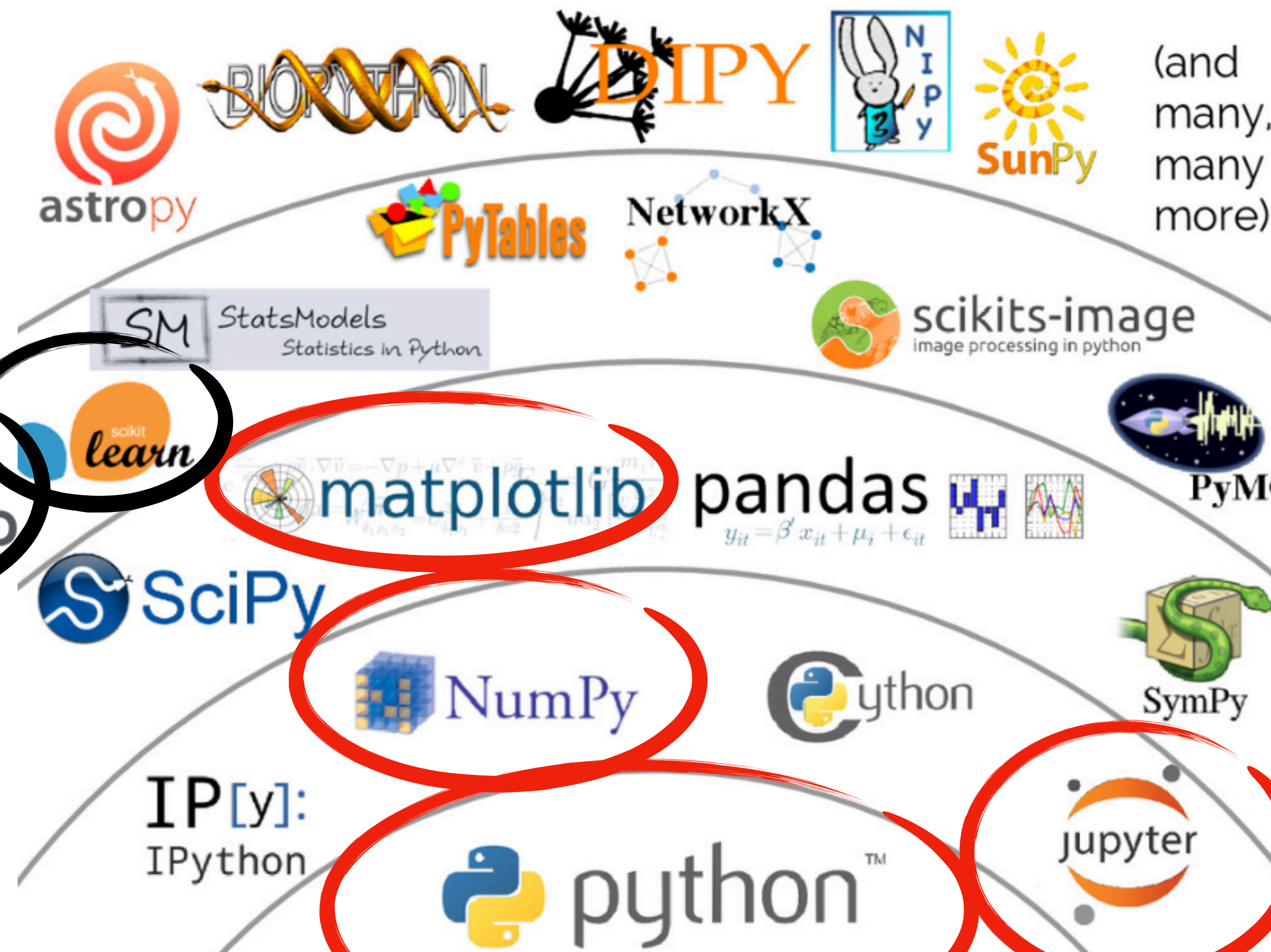
Other Performance Metrics

- Accuracy (1-Error)
- ROC AUC
- Precision
- Recall
- (Cross) Entropy
- Likelihood
- Squared Error/MSE
- L-norms
- Utility
- Fitness
- ...

The Practical Aspects: Our Tools!

Main Scientific Python Libraries

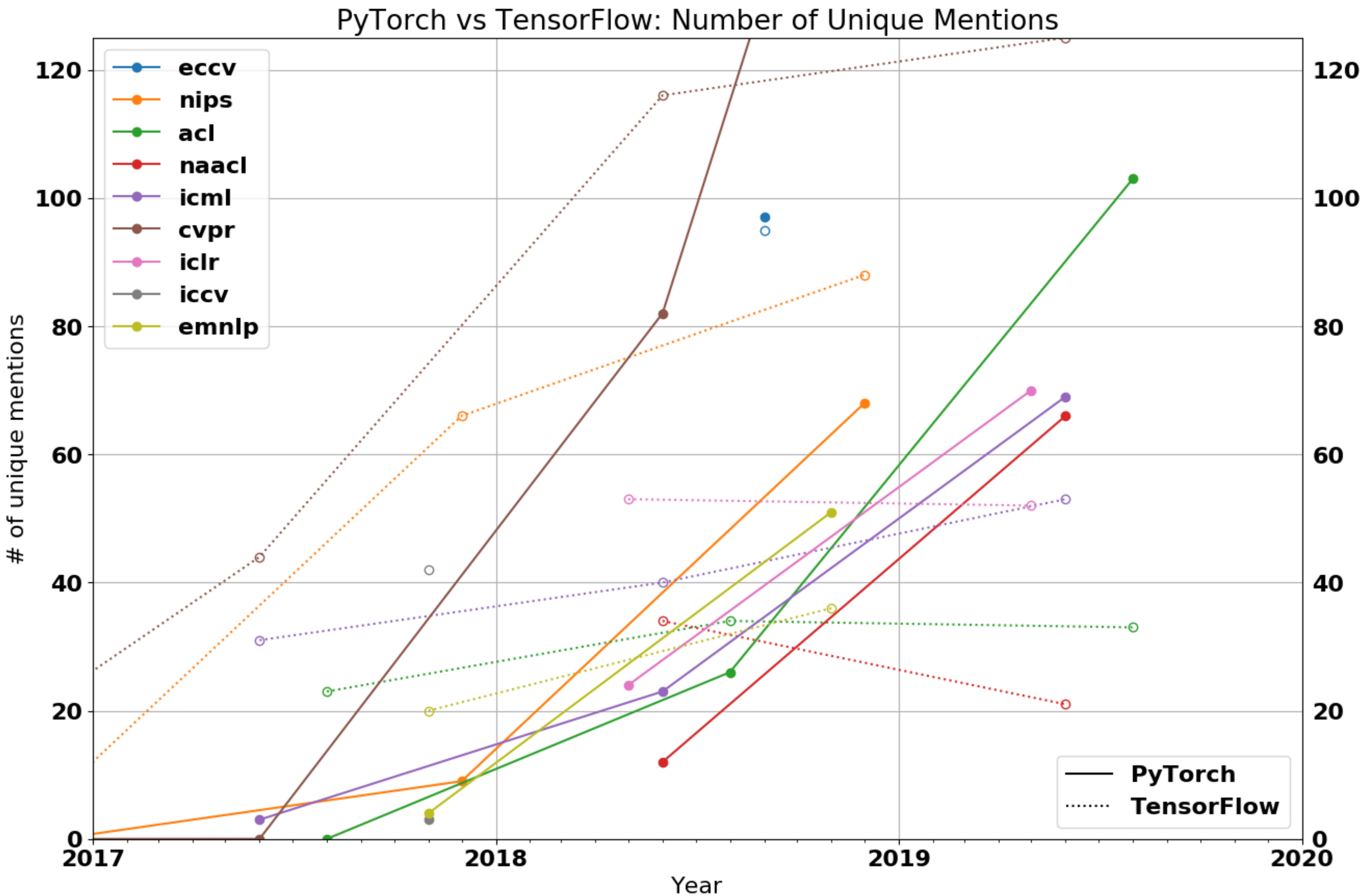
Stat 479 FS2019
(Machine Learning)



Main tools for this course

Image by Jake VanderPlas; Source: https://speakerdeck.com/jakevdp/the-state-of-the-stack-scipy-2015-keynote?slide=8)

"The State of Machine Learning Frameworks in 2019"



Source:
https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/

"The State of Machine Learning Frameworks in 2019"

CONFERENCE	PT 2018	PT 2019	PT GROWTH	TF 2018	TF 2019	TF GROWTH
CVPR	82	280	240%	116	125	7.7%
NAACL	12	66	450%	34	21	-38.2%
ACL	26	103	296%	34	33	-2.9%
ICLR	24	70	192%	54	53	-1.9%
ICML	23	69	200%	40	53	32.5%

In 2018, PyTorch was a minority. Now, it is an overwhelming majority, with 69% of CVPR using PyTorch, 75+% of both NAACL and ACL, and 50+% of ICLR and ICML. While PyTorch's dominance is strongest at vision and language conferences (outnumbering TensorFlow by 2:1 and 3:1 respectively), PyTorch is also more popular than TensorFlow at general machine learning conferences like ICLR and ICML.

Source:
https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/

Next Lecture:

A Brief Summary of the History of Neural Networks and Deep Learning

Reading Assignments

- Pedro Domingos, *A Few Useful Things to Know about Machine Learning*
  https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

- STAT479 FS2019: Machine Learning, lecture notes 01:
  https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/01_overview/01_ml-overview_notes.pdf

(exam questions also assume that you read the assigned reading materials)