STAT 453: Introduction to Deep Learning and Generative Models

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching

Lecture 01

What Are Machine Learning And Deep Learning? An Overview.

Lecture Topics

1. **Course overview**
2. What is machine learning?
3. The broad categories of ML
4. The supervised learning workflow
5. Necessary ML notation and jargon
6. About the practical aspects and tools

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

Audio Classification Using Convolutional Neural Networks

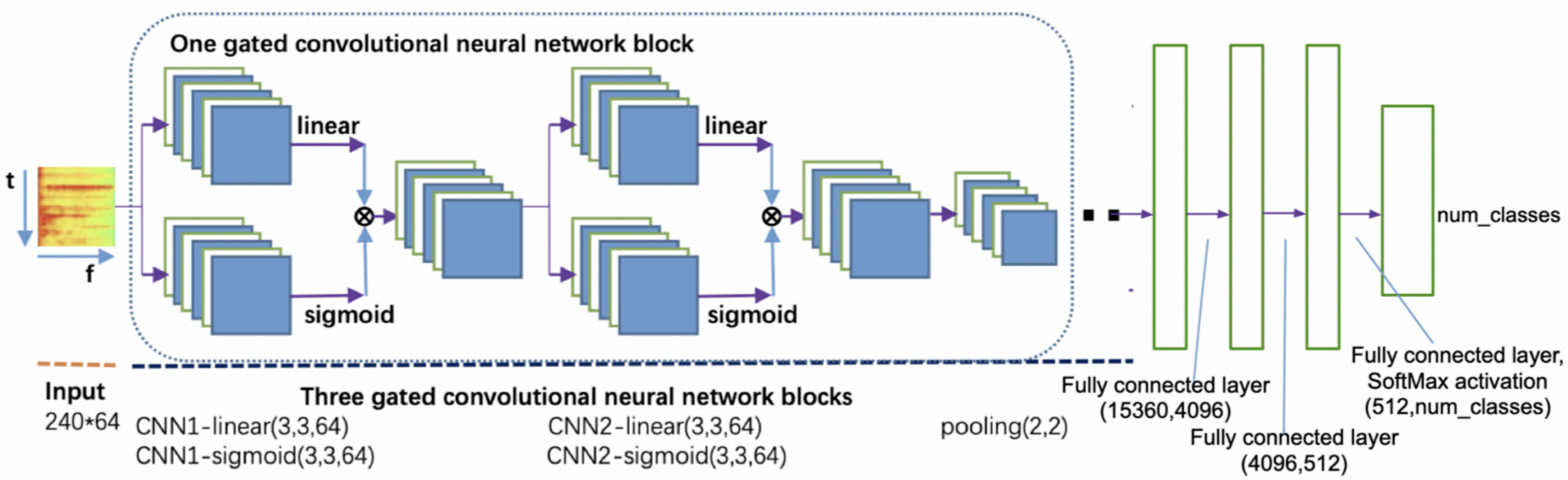
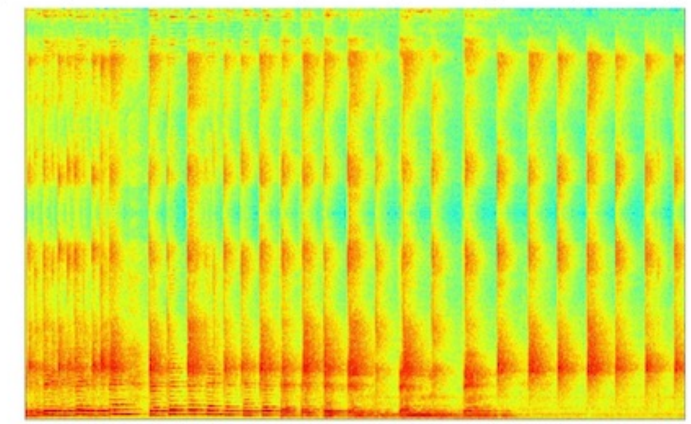
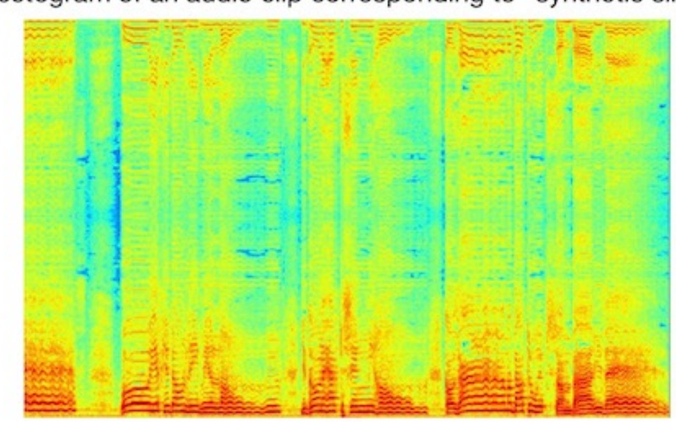


Figure 2. Gated Convolutional Neural Network with Multi-Layered Classification Model

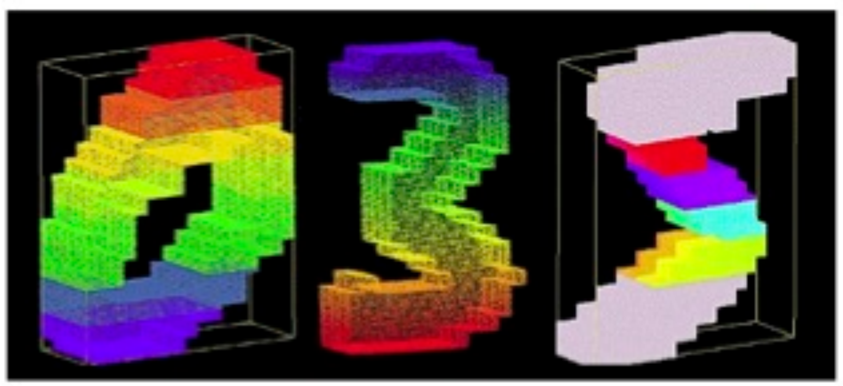
Spectogram of an audio clip corresponding to "finger snapping:"



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



3D Convolutional Networks



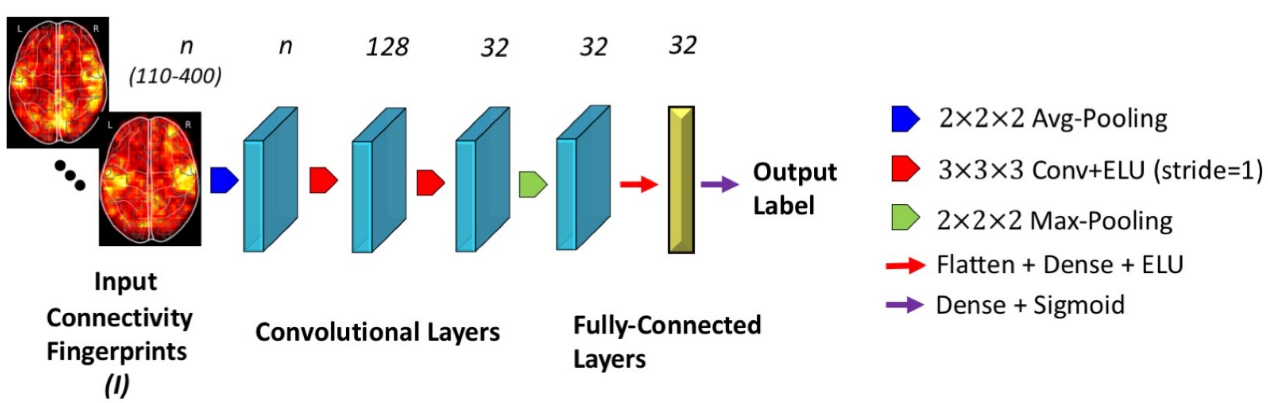
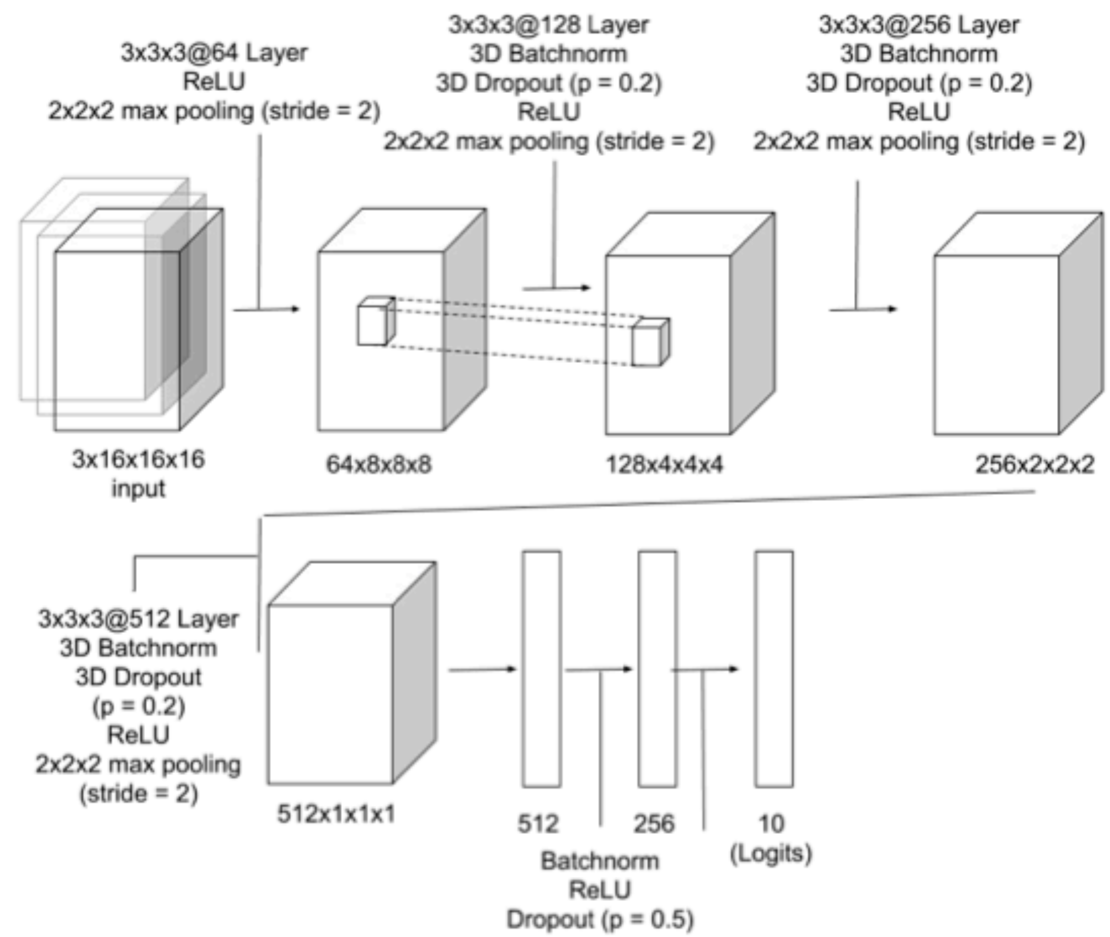
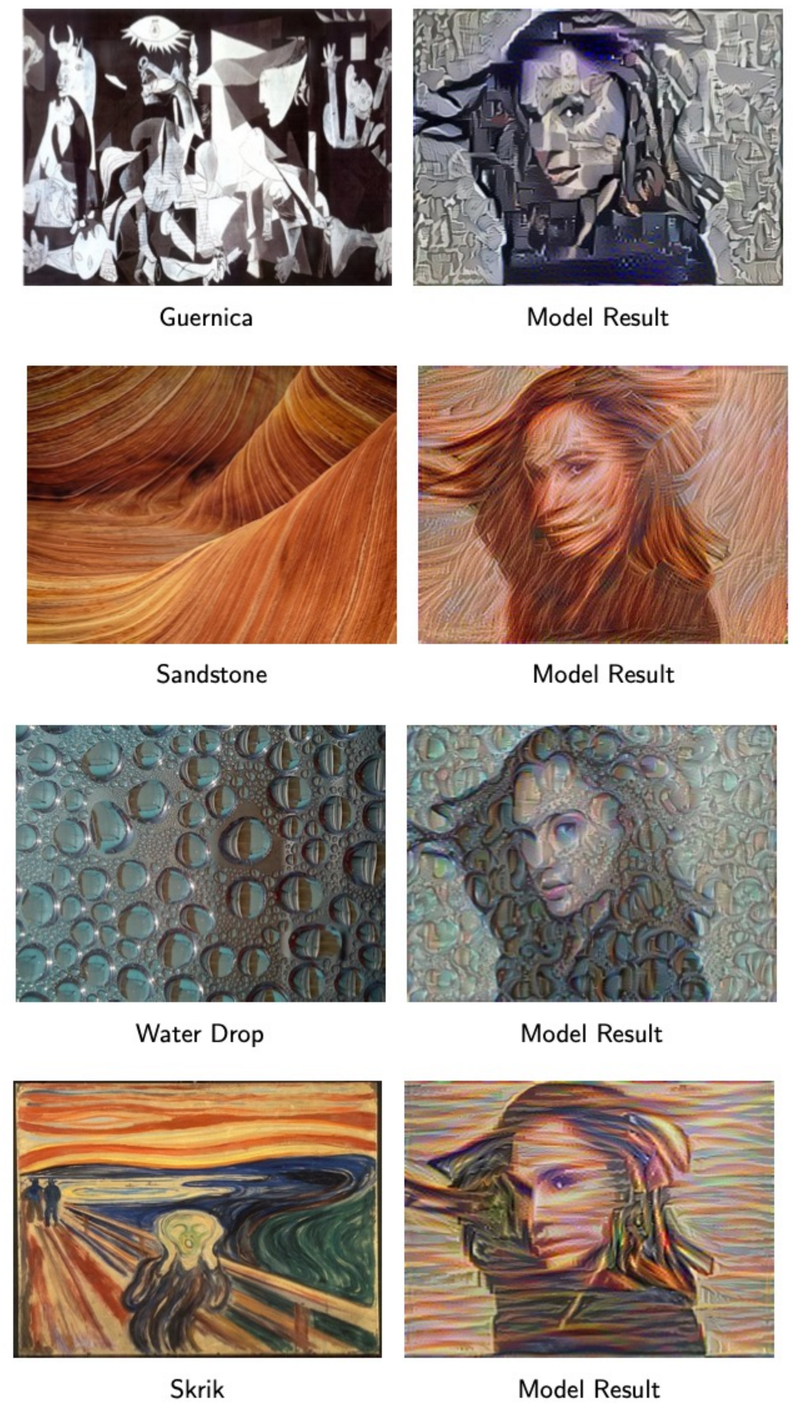


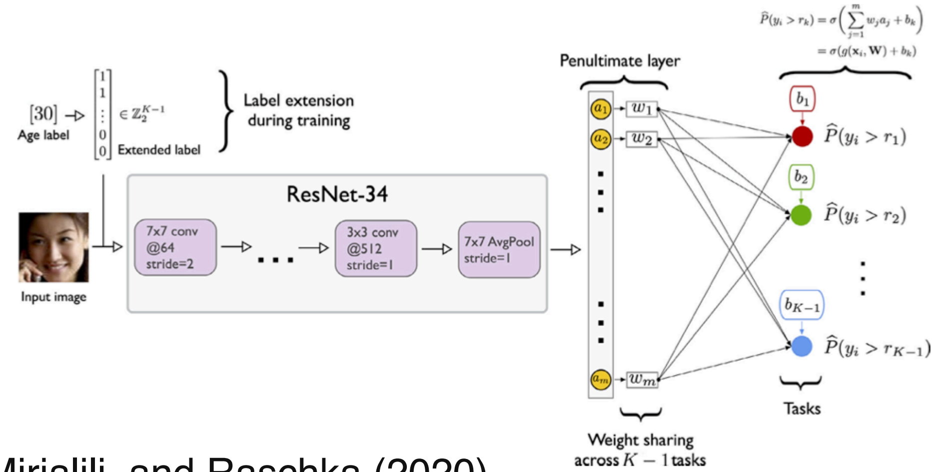
Image Source: 3D Convolutional Neural Networks for Classification of Functional Connectomes. https://arxiv.org/abs/1806.04209



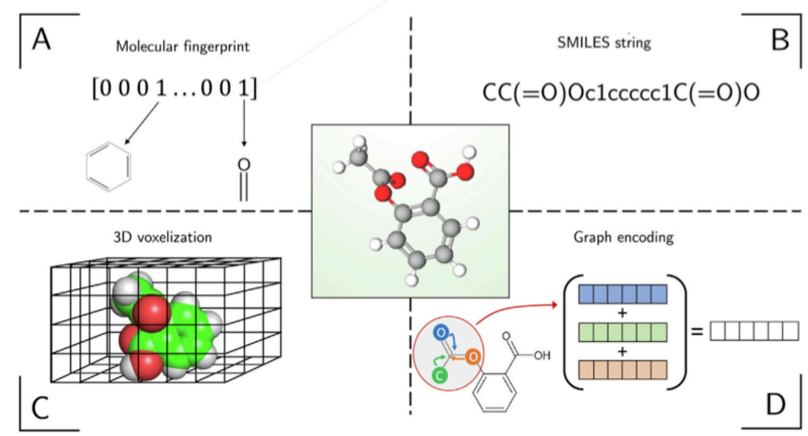
Photographic Style Transfer With Deep Learning



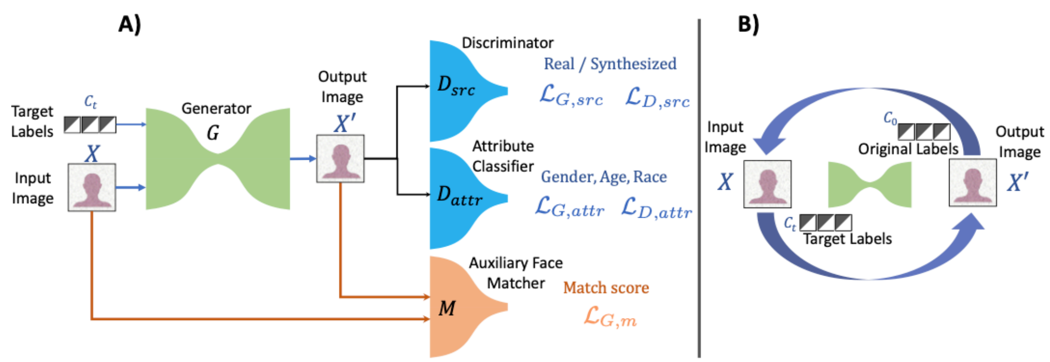
My Research Interests



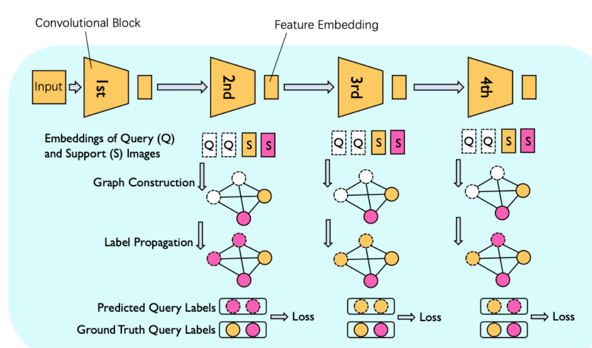
Cao, Mirjalili, and Raschka (2020)
Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation . Pattern Recognition Letters. 140, 325-331



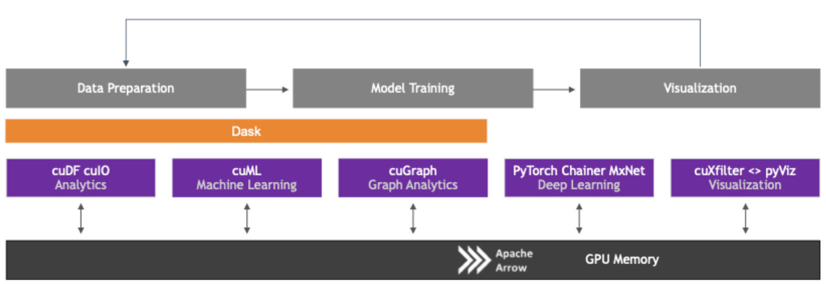
Raschka and Kaufman (2020)
Machine Learning and AI-based Approaches for Bioactive Ligand Discovery and GPCR-ligand Recognition
Elsevier Methods, 180, 89–110



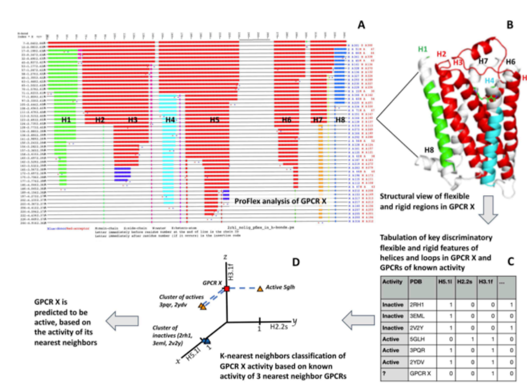
Mirjalili, Raschka, and Ross (2020)
PrivacyNet: Semi-Adversarial Networks for Multi-attribute Face Privacy
IEEE Transactions in Image Processing. Vol. 29, pp. 9400-9412, 2020



Yu and Raschka (2020)
Looking Back to Lower-level Information in Few-shot Learning
Information 2020, 11, 7



Raschka, Patterson, and Nolet (2020)
Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence
Information 2020, 11, 4



Bemister-Buffington, Wolf, Raschka, and Kuhn (2020)
Machine Learning to Identify Flexibility Signatures of Class A GPCR Inhibition
Biomolecules 2020, 10, 454.

About the course

Topics Planned 1/2

Part 1: Introduction

- L01: Course overview, introduction to deep learning
- L02: The brief history of deep learning
- L03: Single-layer neural networks: The perceptron algorithm

Part 2: Mathematical and computational foundations

- L04: Linear algebra and calculus for deep learning
- L05: Parameter optimization with gradient descent
- L06: Automatic differentiation with PyTorch
- L07: Cluster and cloud computing resources

Part 3: Introduction to neural networks

- L08: Multinomial logistic regression
- L09: Multilayer perceptrons and backpropration
- L10: Regularization to avoid overfitting
- L11: Input normalization and weight initialization
- L12: Learning rates and advanced optimization algorithms

Topics Planned 2/2

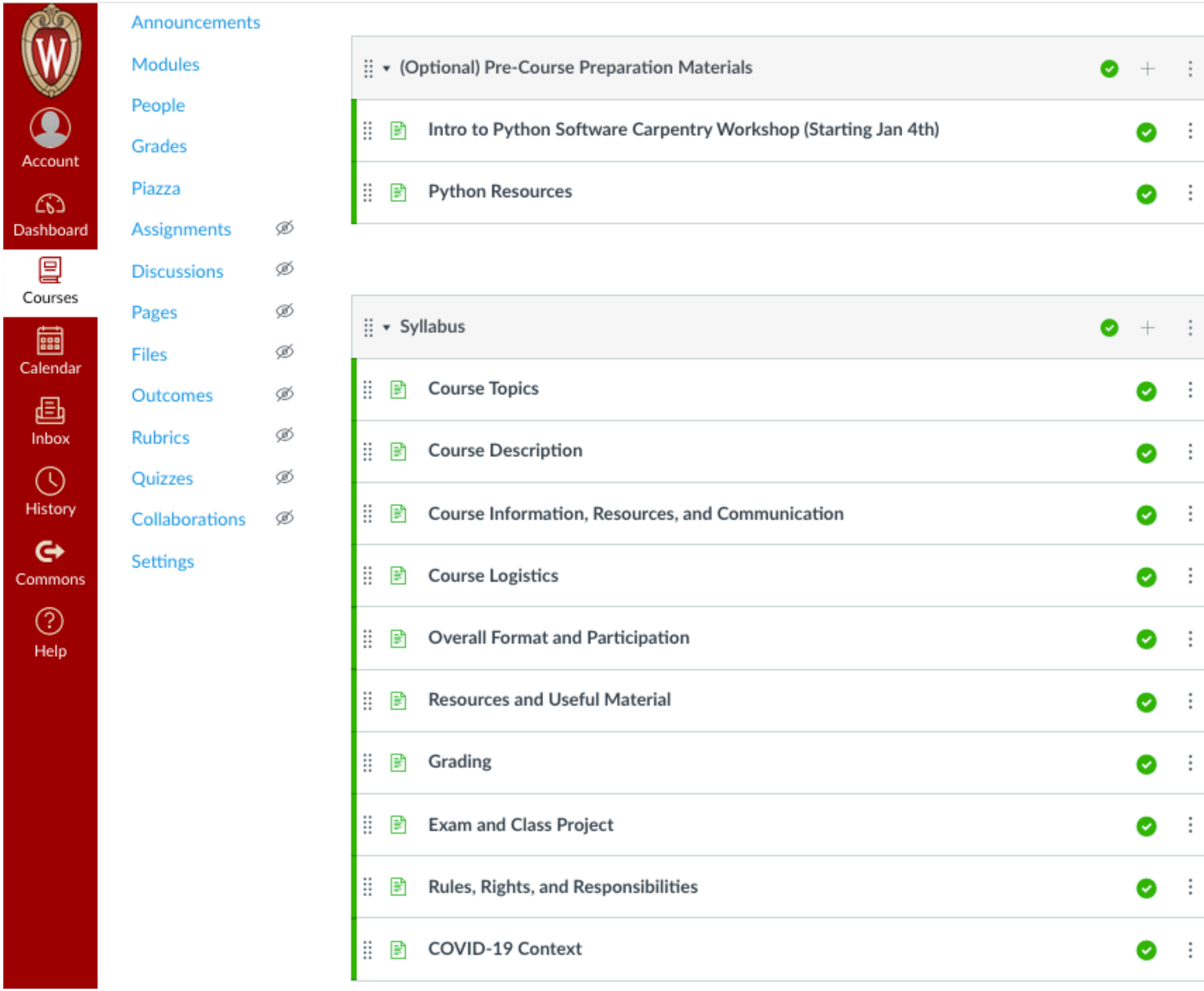
Part 4: Deep learning for computer vision and language modeling

- L13: Introduction to convolutional neural networks
- L14: Convolutional neural networks architectures
- L15: Introduction to recurrent neural networks

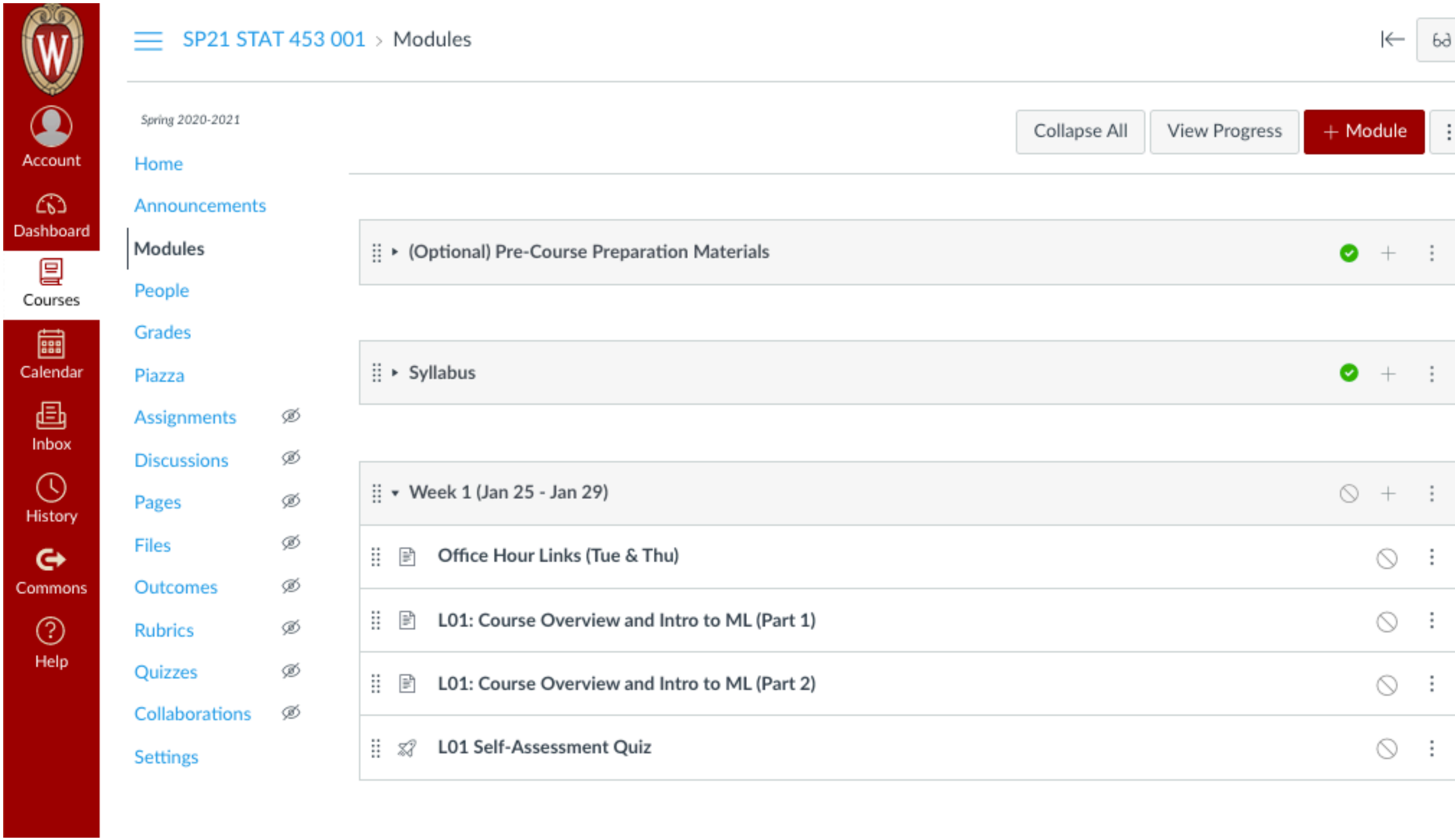
Part 5: Deep generative models

- L16: Autoencoders
- L17: Variational autoencoders
- L18: Introduction to generative adversarial networks
- L19: Evaluating generative adversarial networks
- L20: Recurrent neural networks for seq-to-seq modeling
- L21: Self-attention and transformer networks

Course Material and Info



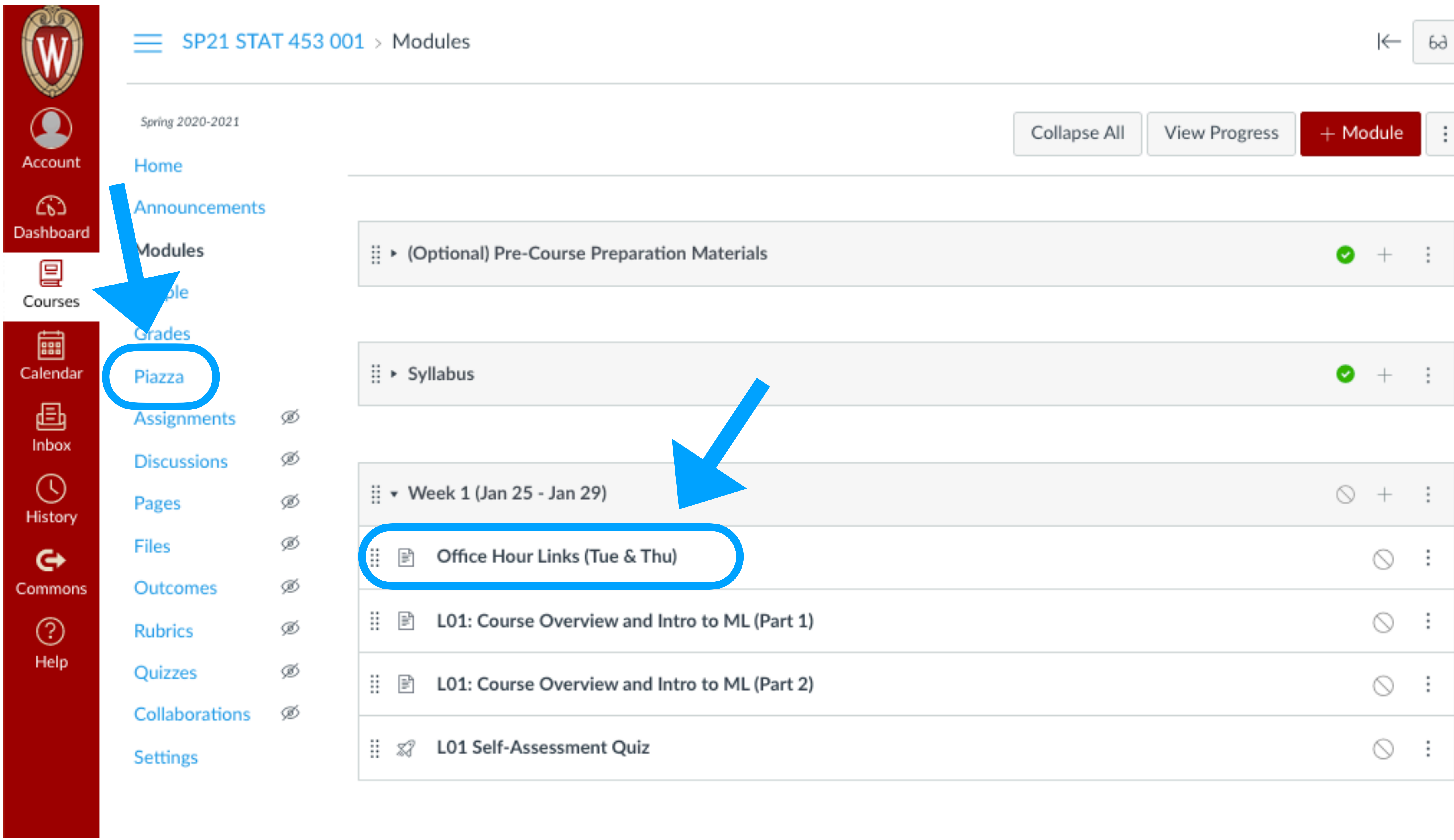
Weekly Content



Grading

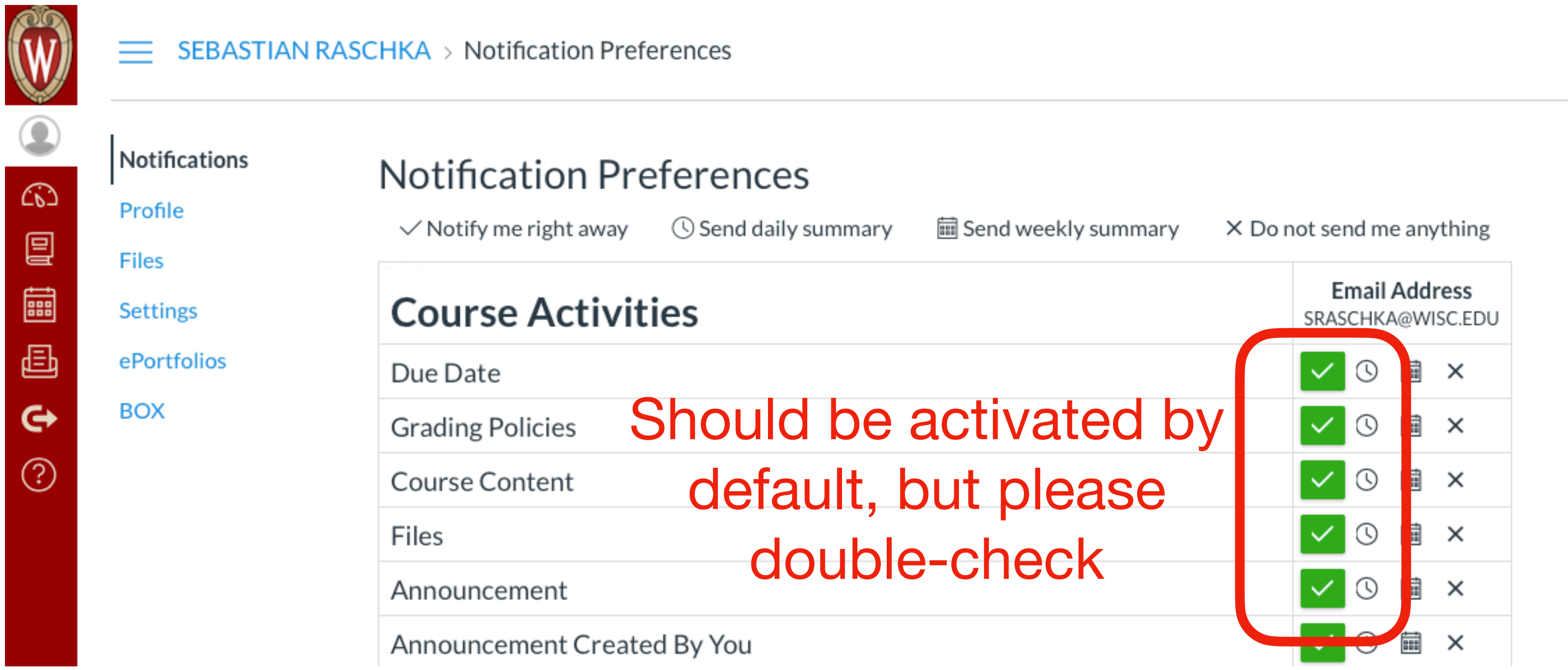
- 30% Problem Sets (HW assignments and quizzes)
- 20% Midterm Exam
- 50% Class Project:
  - 5% Project proposal
  - 20% Project presentation (+ peer review)
  - 25% Project report (+ peer review)

Questions & Discussions



Important!

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

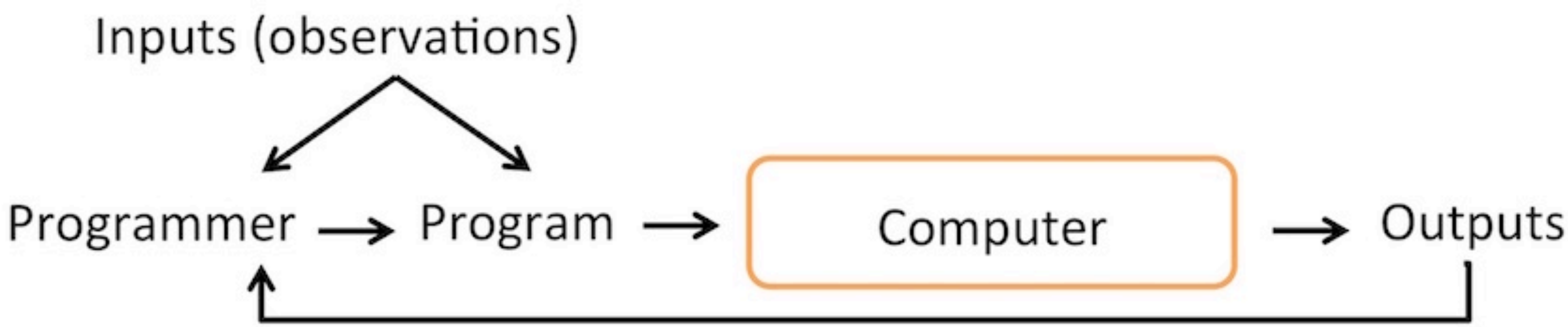


What Is Machine Learning?

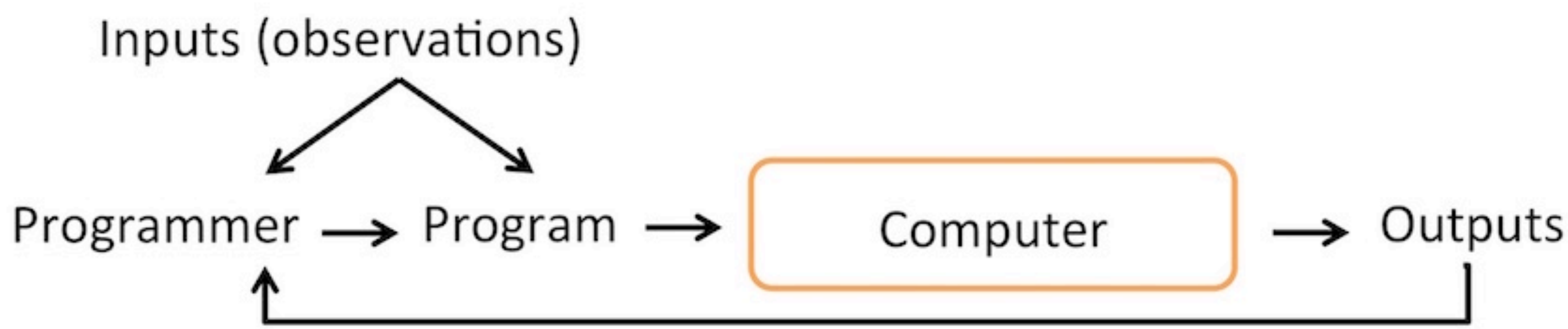
A short overview before we jump into Deep Learning

1. Course overview
2. **What is machine learning?**
3. The broad categories of ML
4. The supervised learning workflow
5. Necessary ML notation and jargon
6. About the practical aspects and tools

The Traditional Programming Paradigm



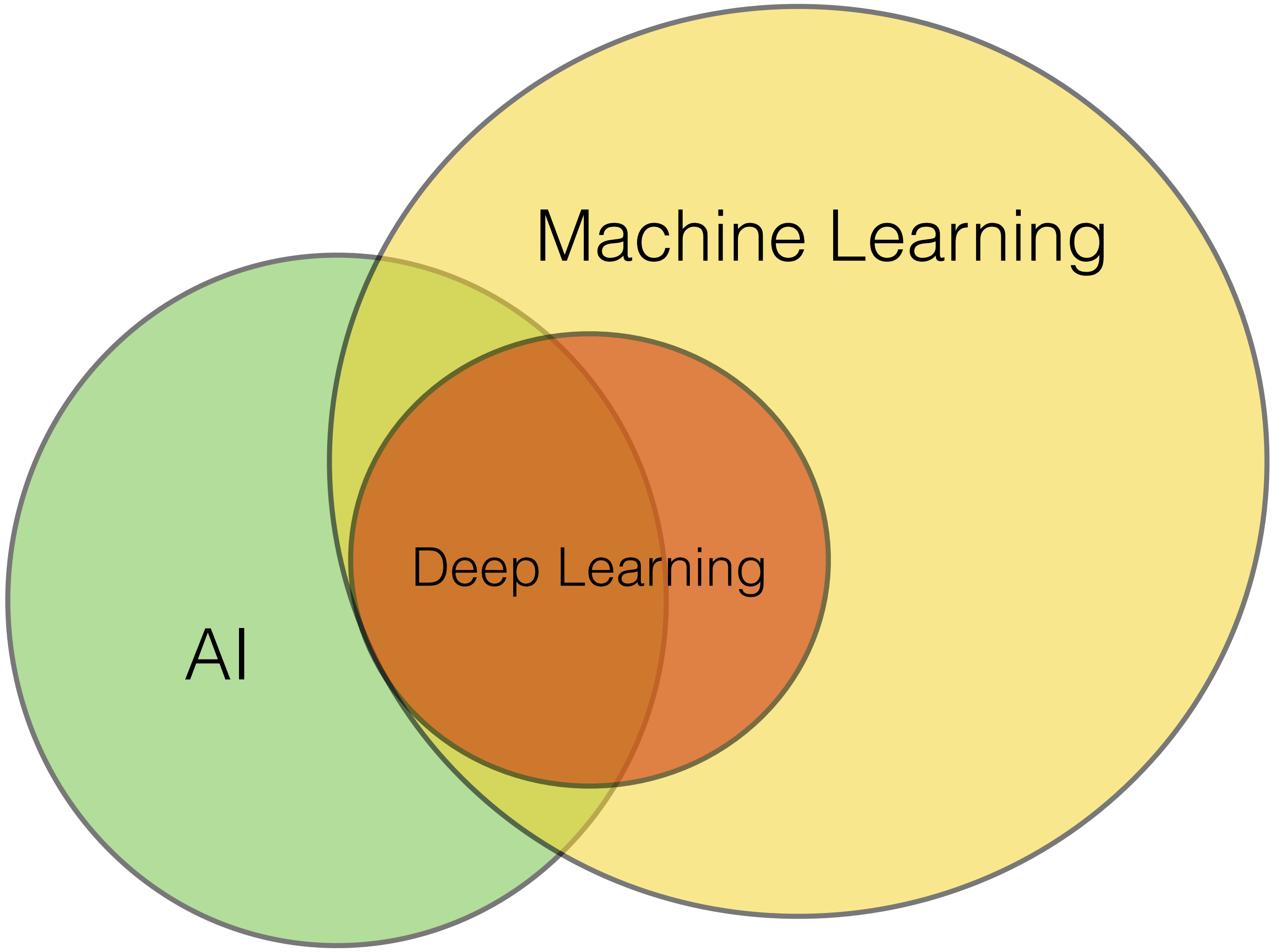
The Traditional Programming Paradigm



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

Machine Learning

The Connection Between Fields

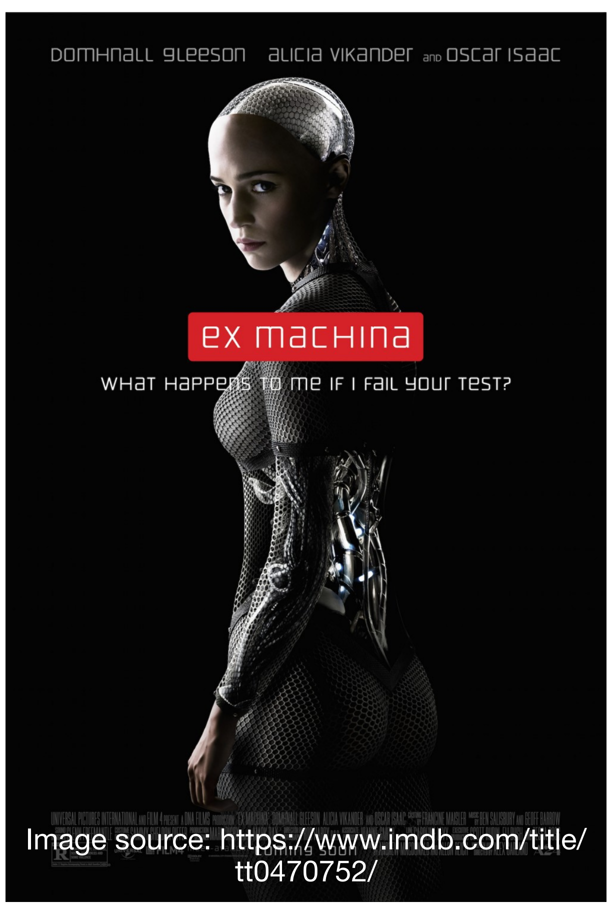


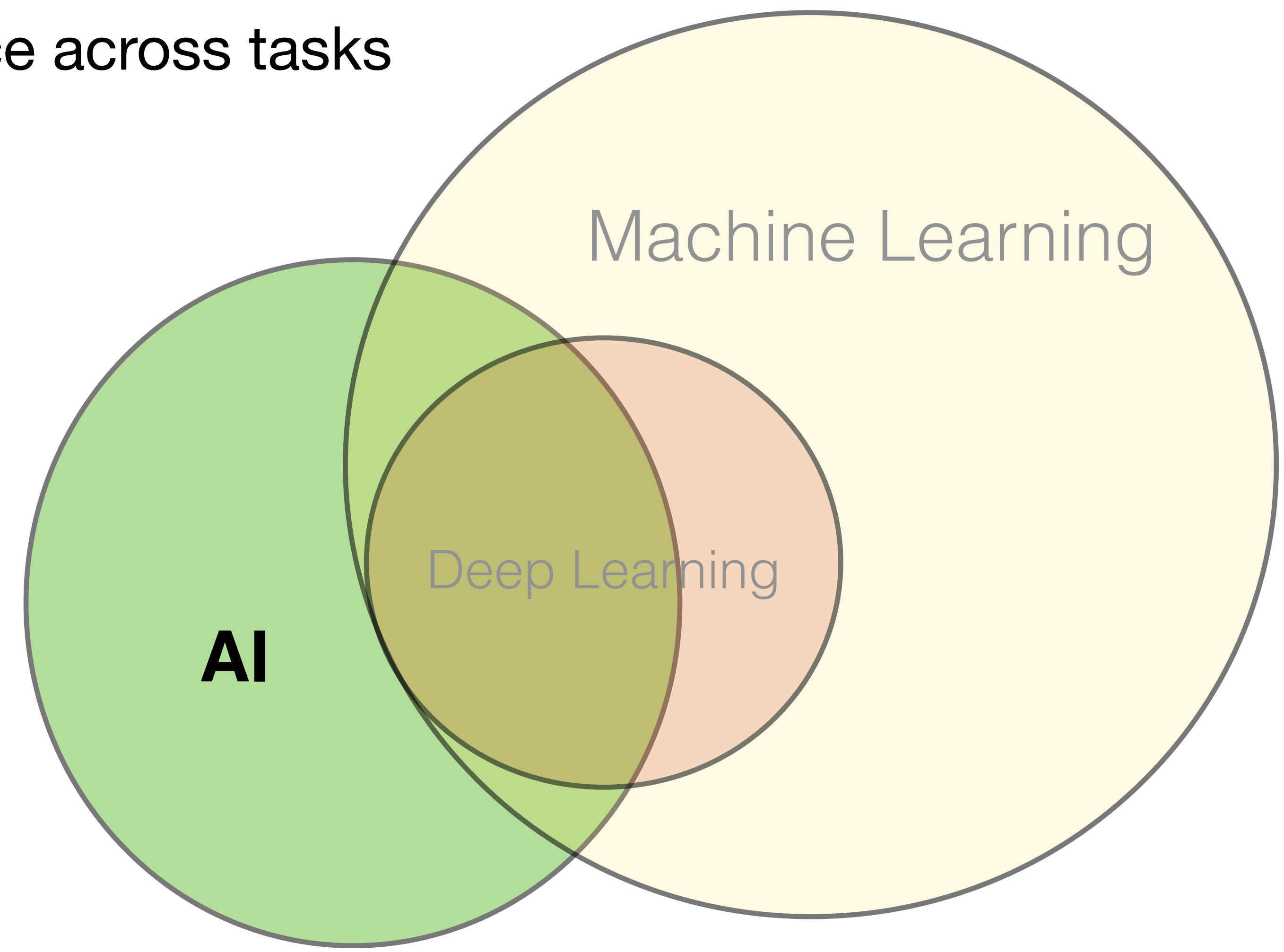
Different Types Of AI

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

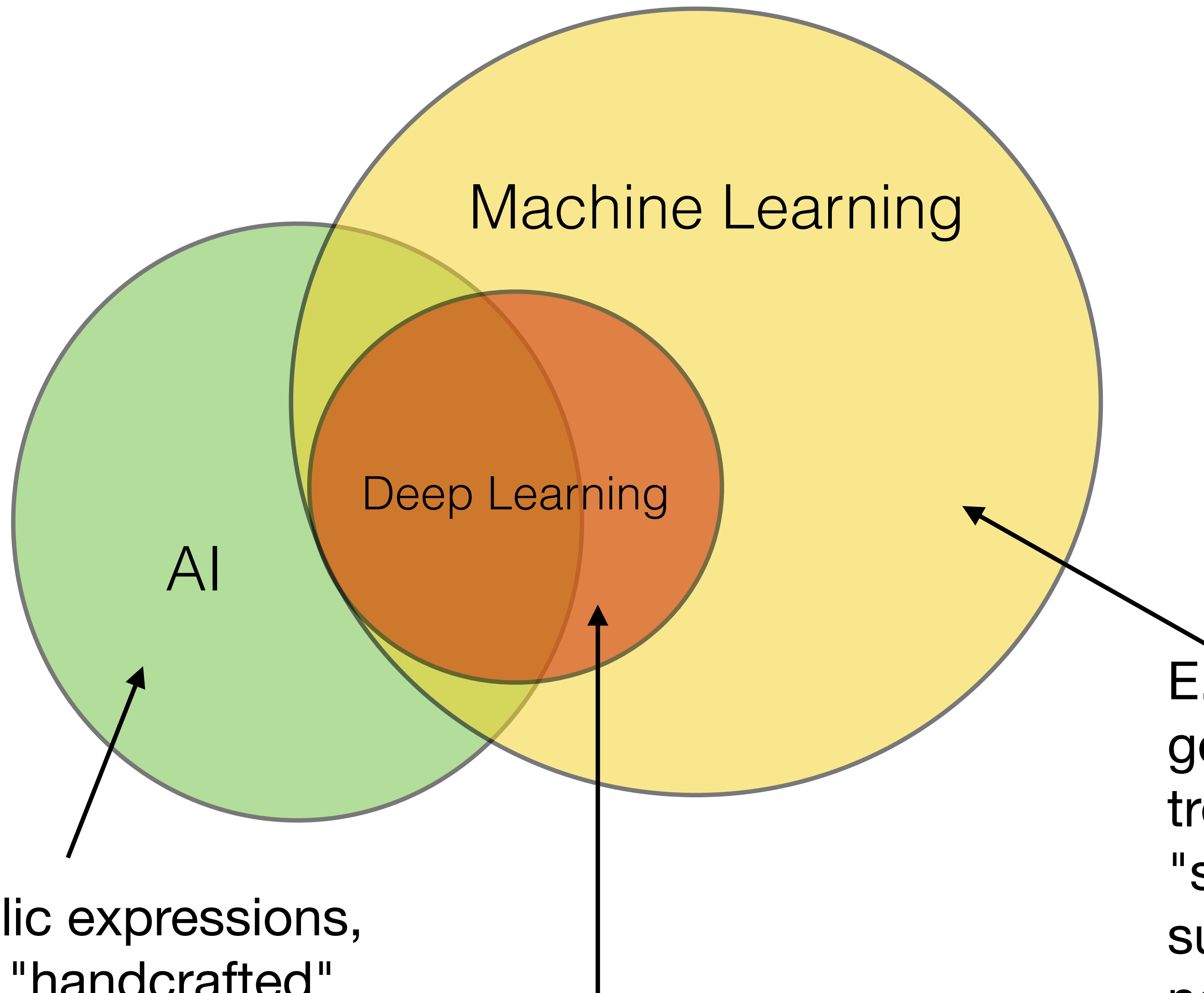
Narrow AI:
solving a particular task (playing a game, driving a car, ...)

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





What This Course Is About



E.g., symbolic expressions, 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, ...

Not all AI Systems involve Machine Learning

Deep Blue used custom VLSI chips to execute the **alpha-beta search** algorithm in parallel, an example of GOFAI
(Good Old-Fashioned Artificial Intelligence).

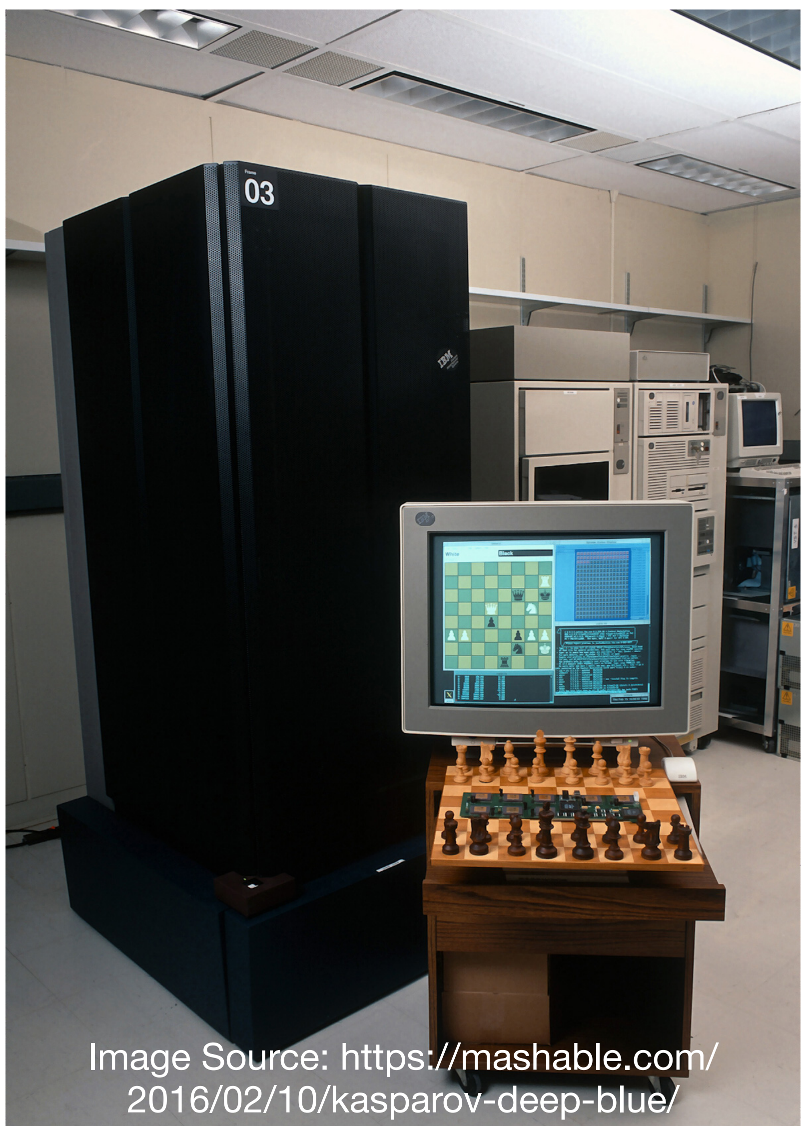
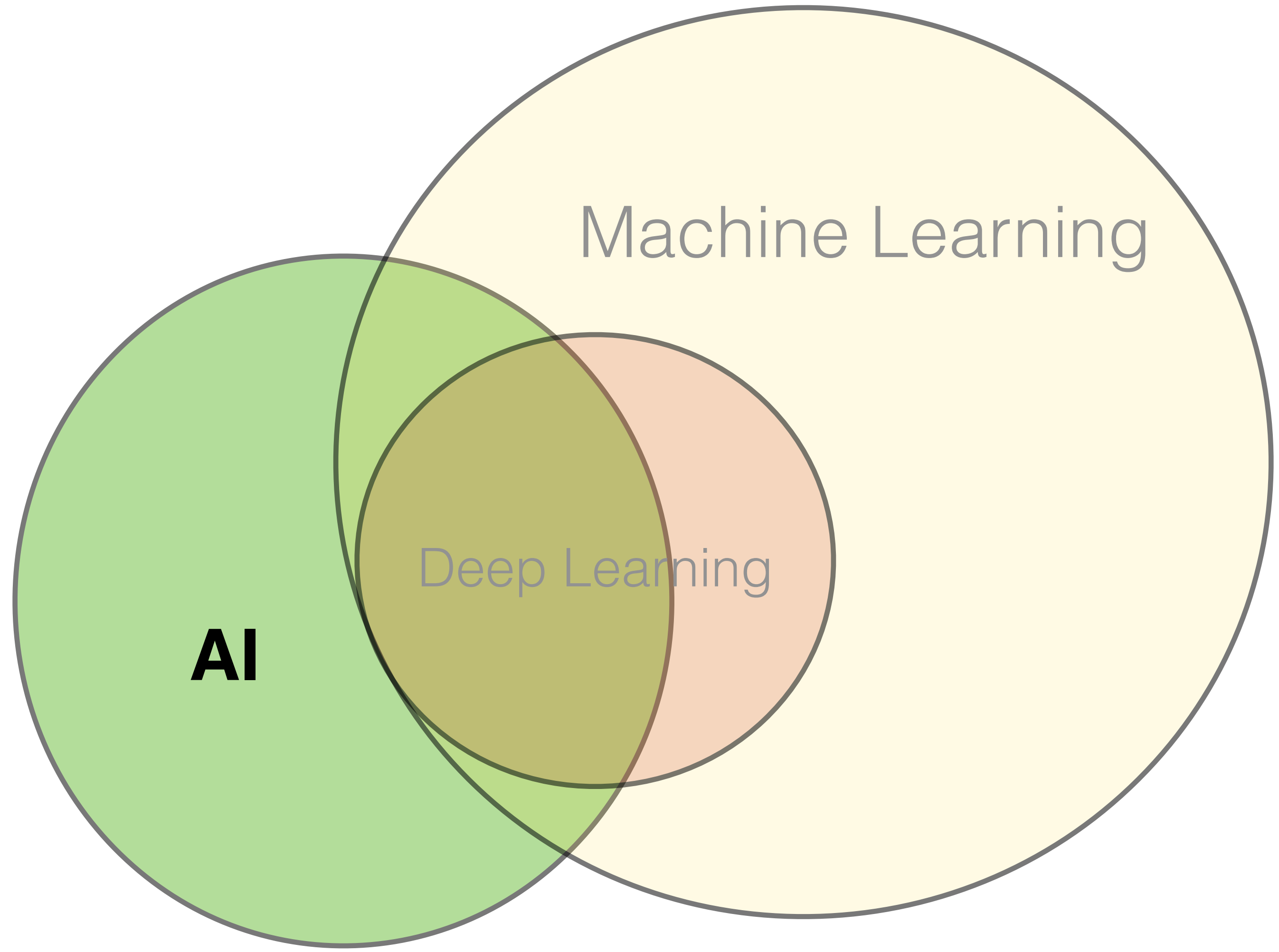
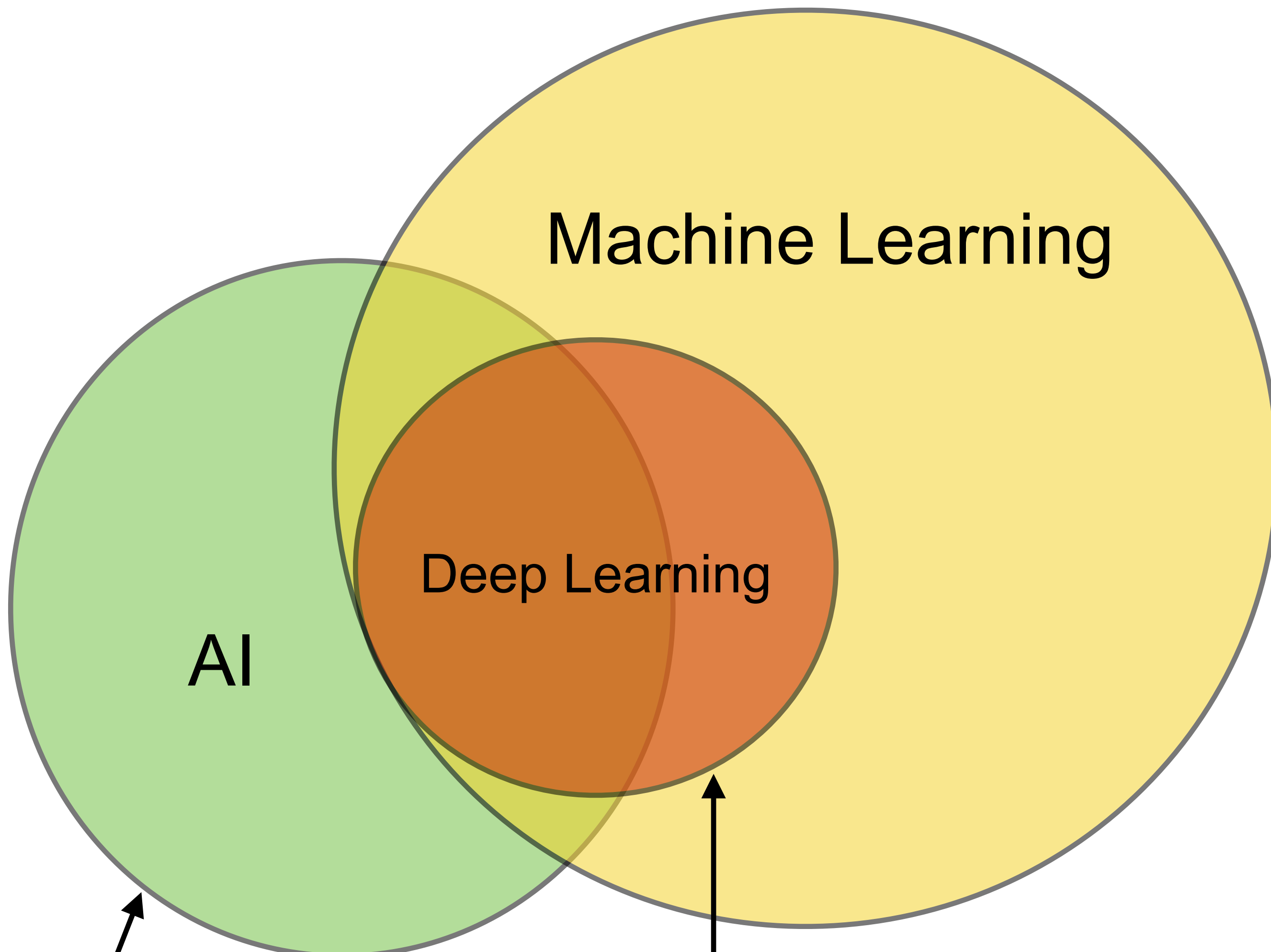


Image Source: https://mashable.com/2016/02/10/kasparov-deep-blue/



Examples From The Three Related "Areas"



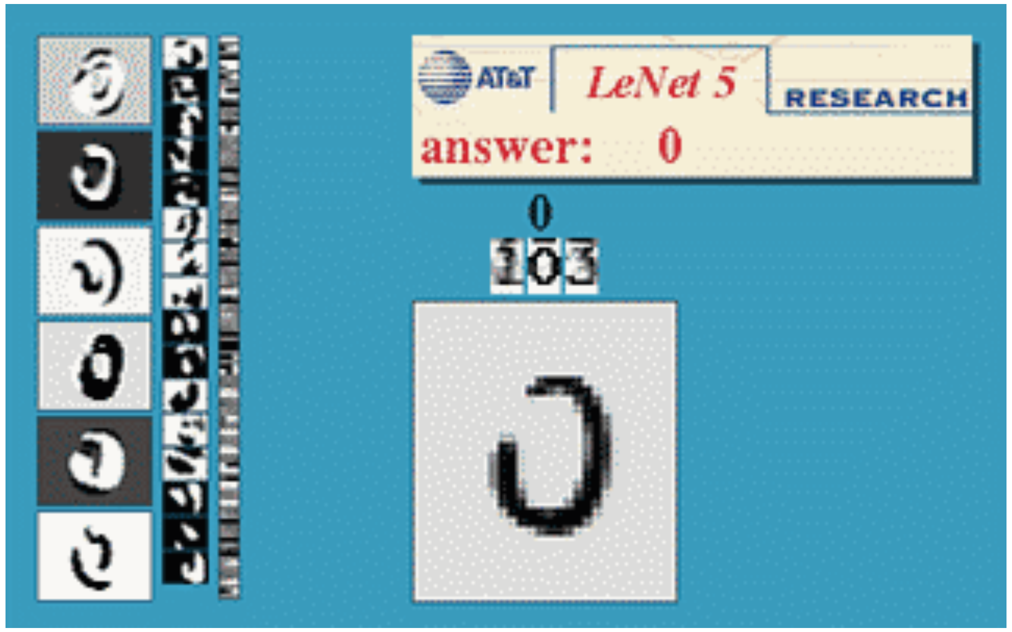
A non-biological system
that is intelligent
through rules

Algorithms that parameterize multilayer
neural networks that then learn
representations of data with multiple layers
of abstraction

Algorithms that learn
models/representations/
rules automatically
from data/examples

Some Applications Of Machine Learning/Deep Learning

- Email spam detection
- Fingerprint / face detection & matching (e.g., phones)
- 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



Source: http://yann.lecun.com/exdb/lenet/

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
- ...

Source: https://techcrunch.com/2017/11/07/waymo-now-testing-its-self-driving-cars-on-public-roads-with-no-one-at-the-wheel/

The 3 Broad Categories of ML

(This also applies to DL)

1. Course overview
2. What is machine learning?
3. **The broad categories of ML**
4. The supervised learning workflow
5. Necessary ML notation and jargon
6. About the practical aspects and tools

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*

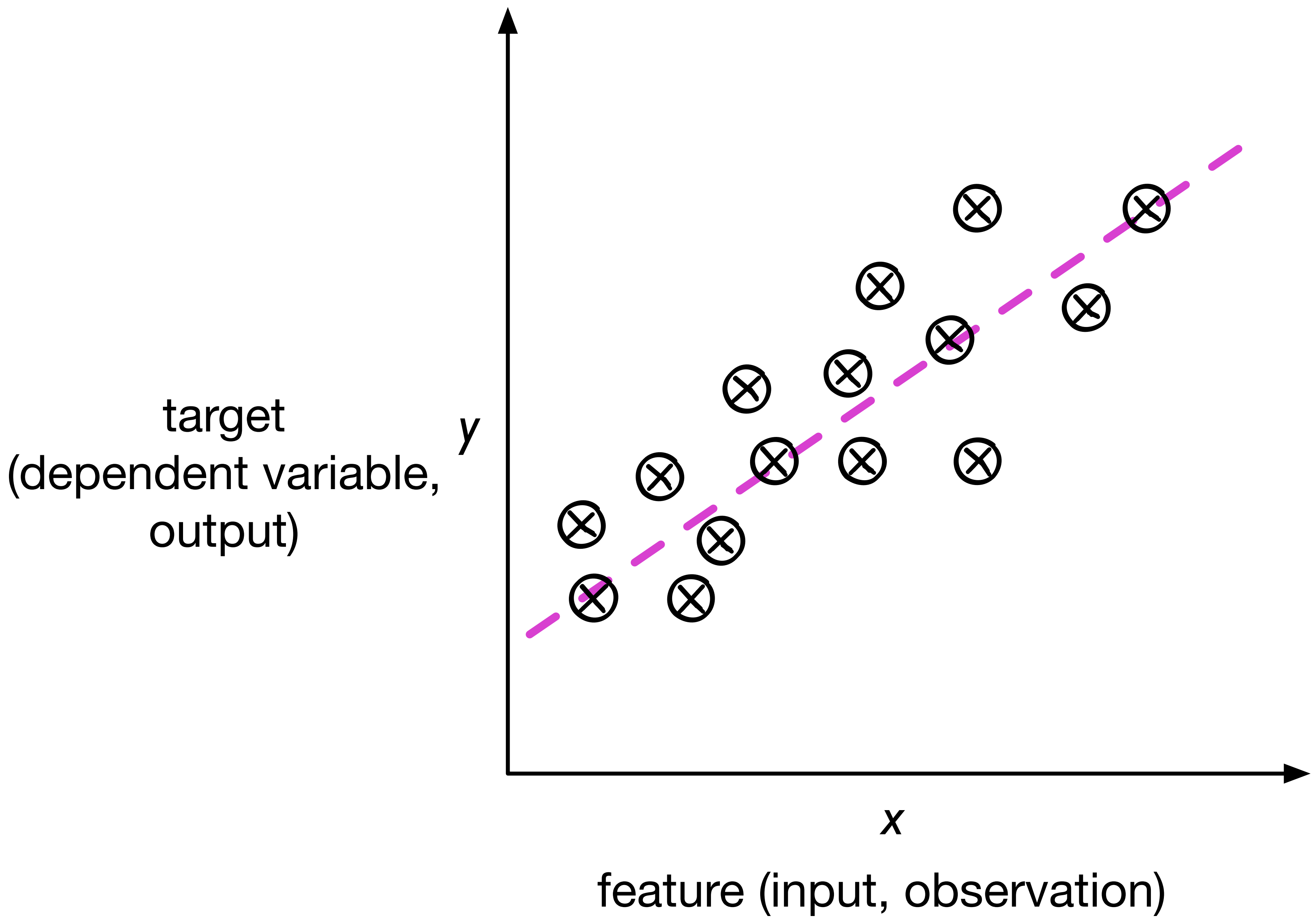
Supervised Learning Is The Largest Subcategory

Supervised Learning

- Labeled data
- Direct feedback
- Predict outcome/future

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

Supervised Learning 1: Regression

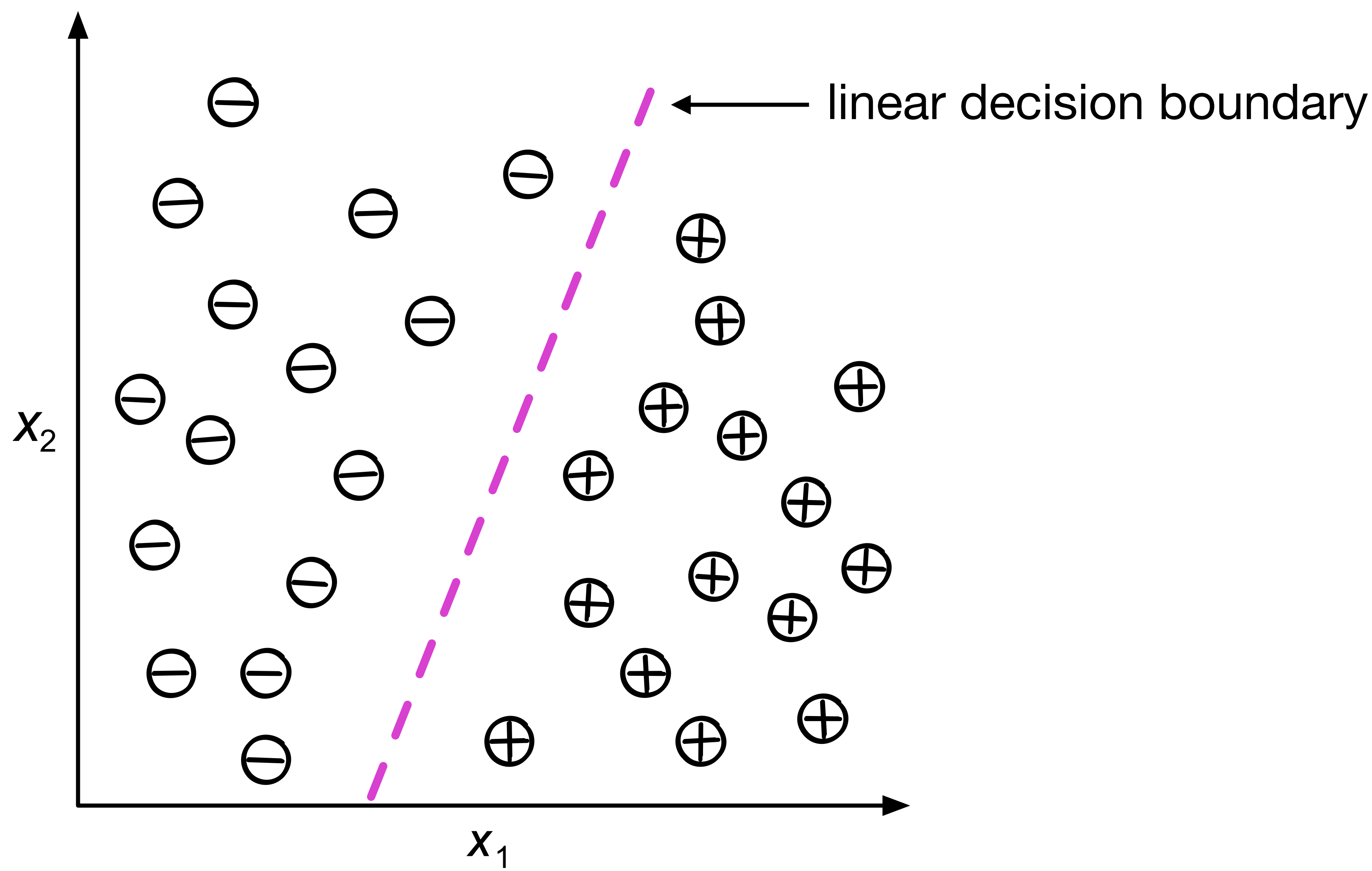


Source: Raschka and Mirjalili (2019). *Python Machine Learning, 3rd Edition*

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 ≻ good ≻ okay ≻ for genre fans ≻ bad*

Supervised Learning 3: Ordinal regression

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

$\prec$

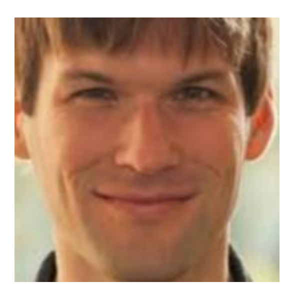
$\prec$

- **Ordinal regression:** Predict correct (ordered) label
  (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

$\prec$

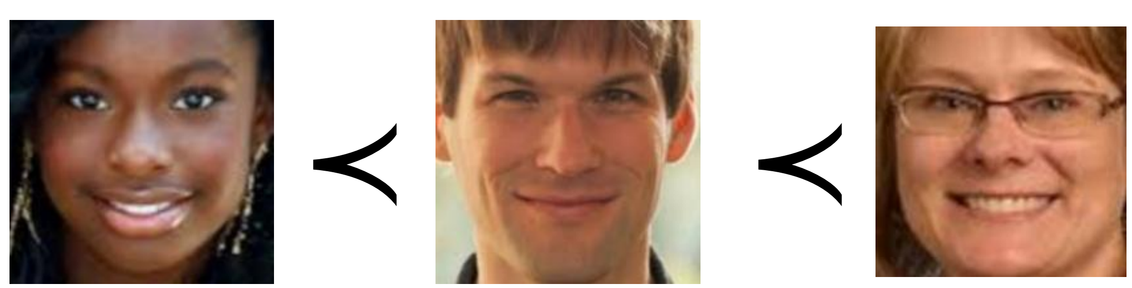
41

Supervised Learning 3: Ordinal regression

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

- **Ordinal regression:** Predict correct (ordered) label
  (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 2nd Subcategory Of ML (And DL)

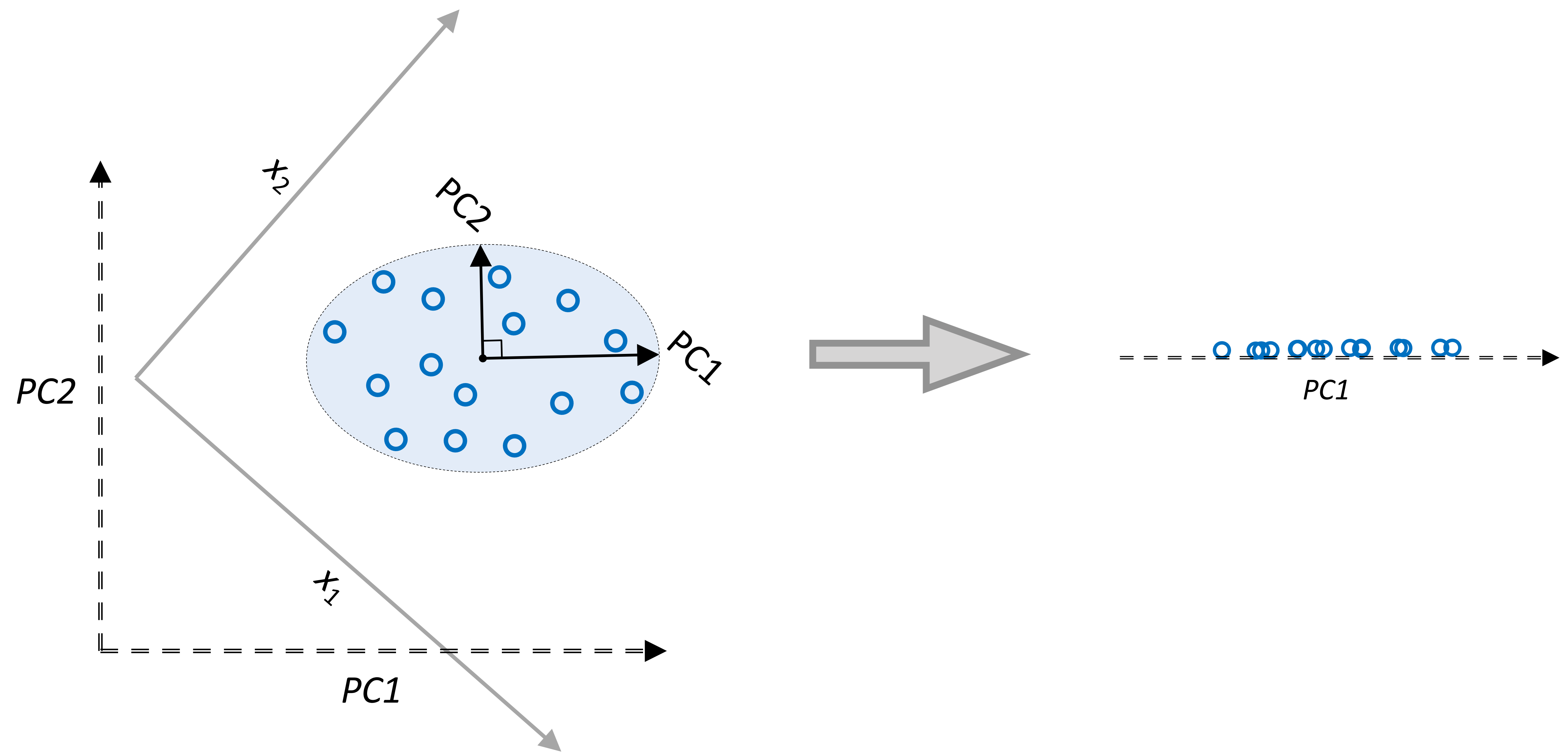
Unsupervised Learning

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

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

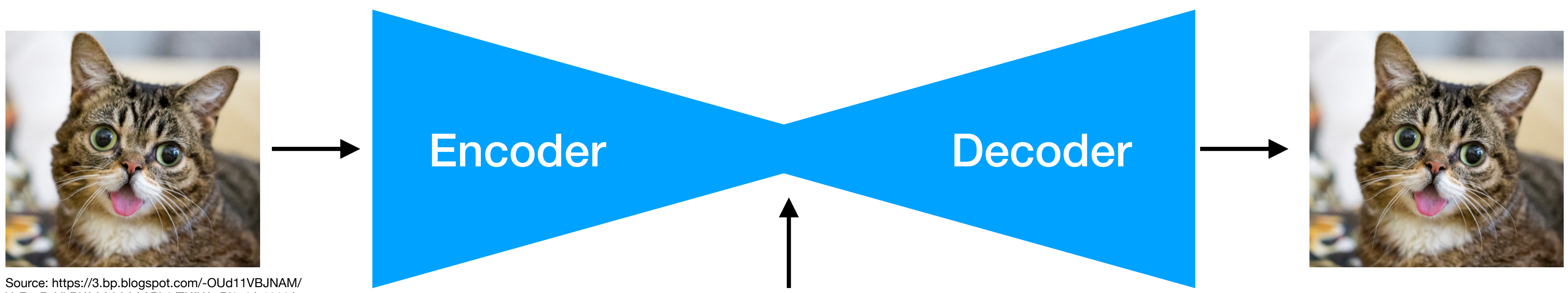
Unsupervised Learning 1: Representation Learning/Dimensionality Reduction

E.g., Principal Component Analysis (PCA)



Unsupervised Learning 1:
Representation Learning/Dimensionality Reduction

E.g., Autoencoders

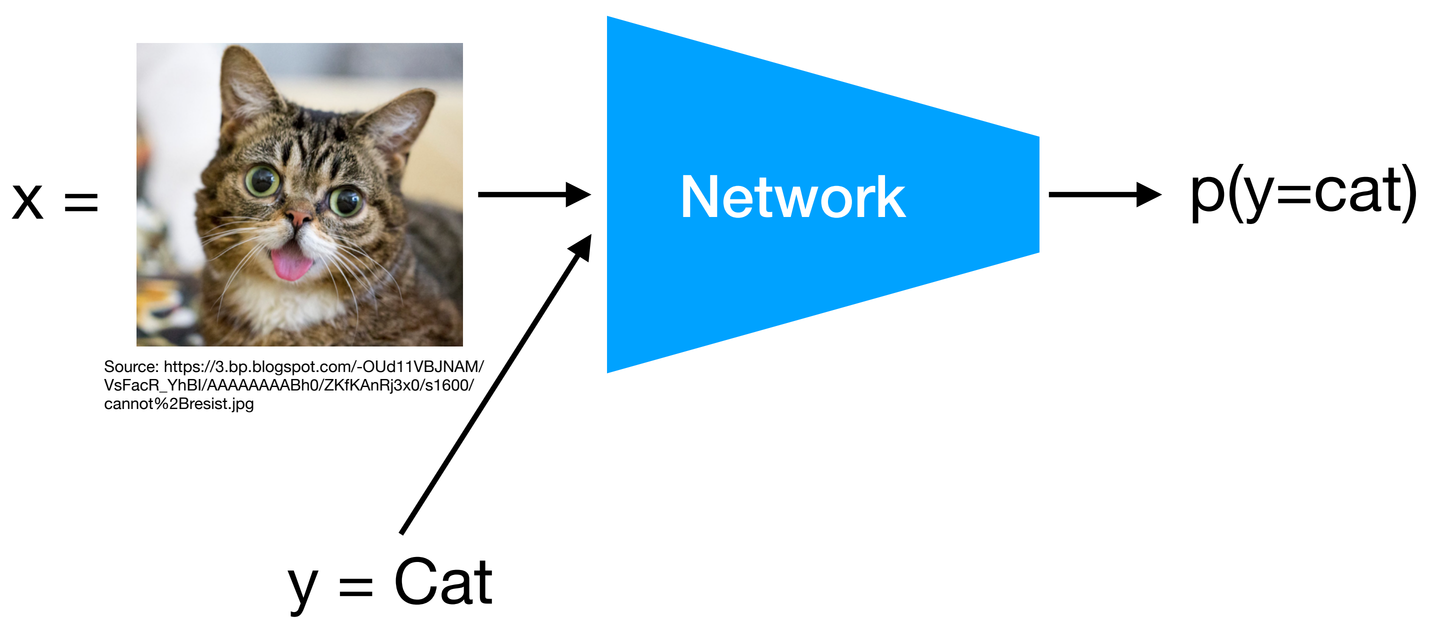


Source: https://3.bp.blogspot.com/-OUd11VBJNAM/VsFacR_YhBI/AAAAAAAABh0/ZKfKAnRj3x0/s1600/cannot%2Bresist.jpg

**latent representation/
feature embedding**

(covered later in this course)

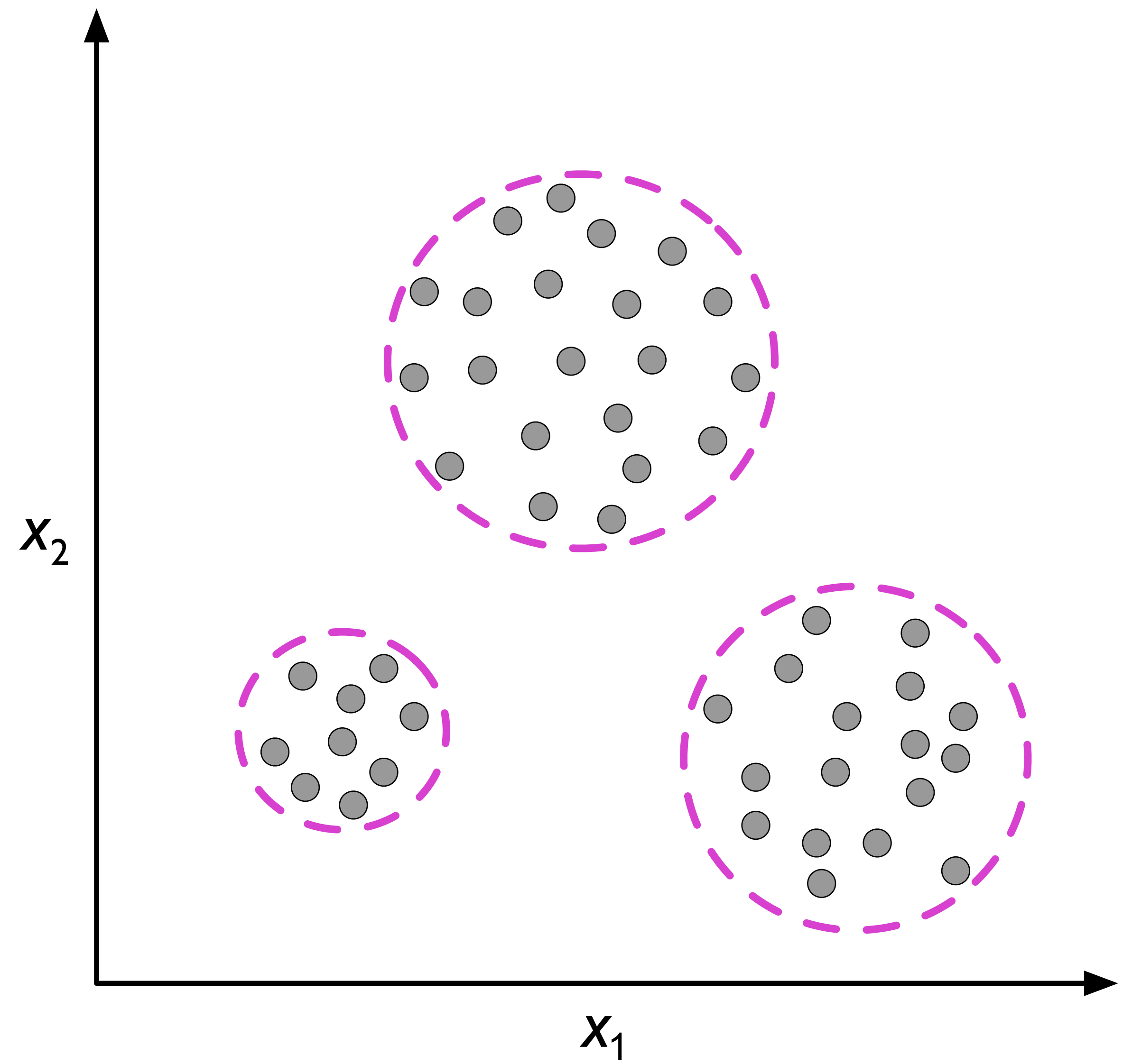
Reminder: Classification works like this



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*

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

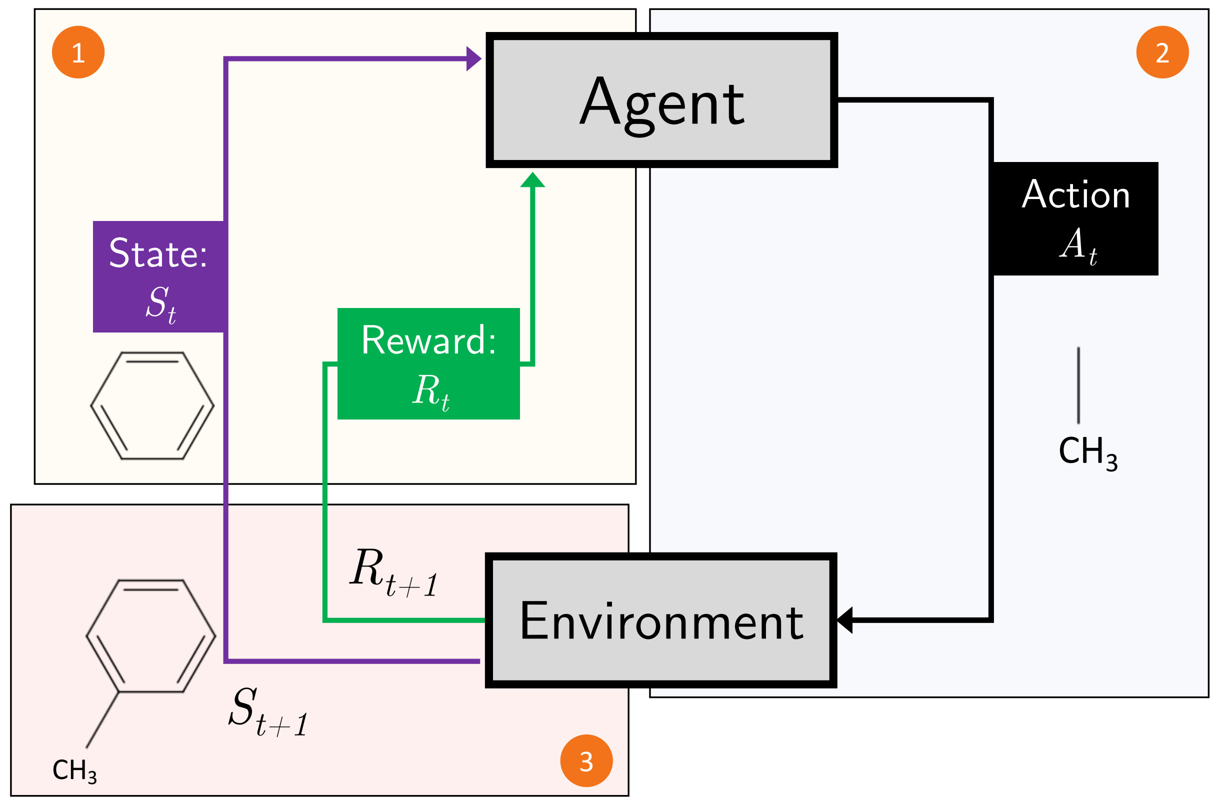
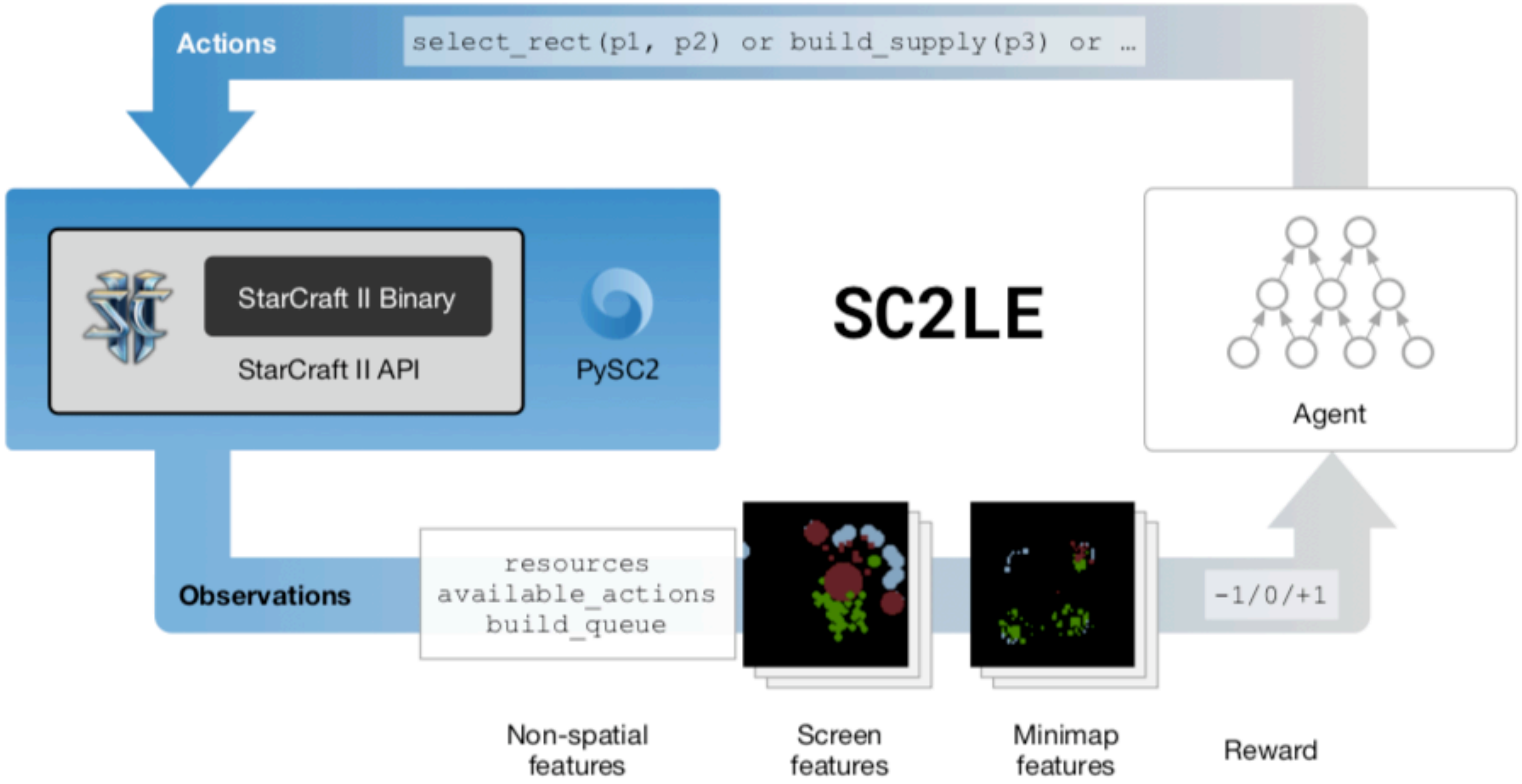


Figure 5: Representation of the basic reinforcement learning paradigm with a simple molecular example. (1) Given a benzene ring (state S_t at iteration t) and some reward value R_t at iteration t, (2) the agent selects an action A_t that adds a methyl group to the benzene ring. (3) The environment considers 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)



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).

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

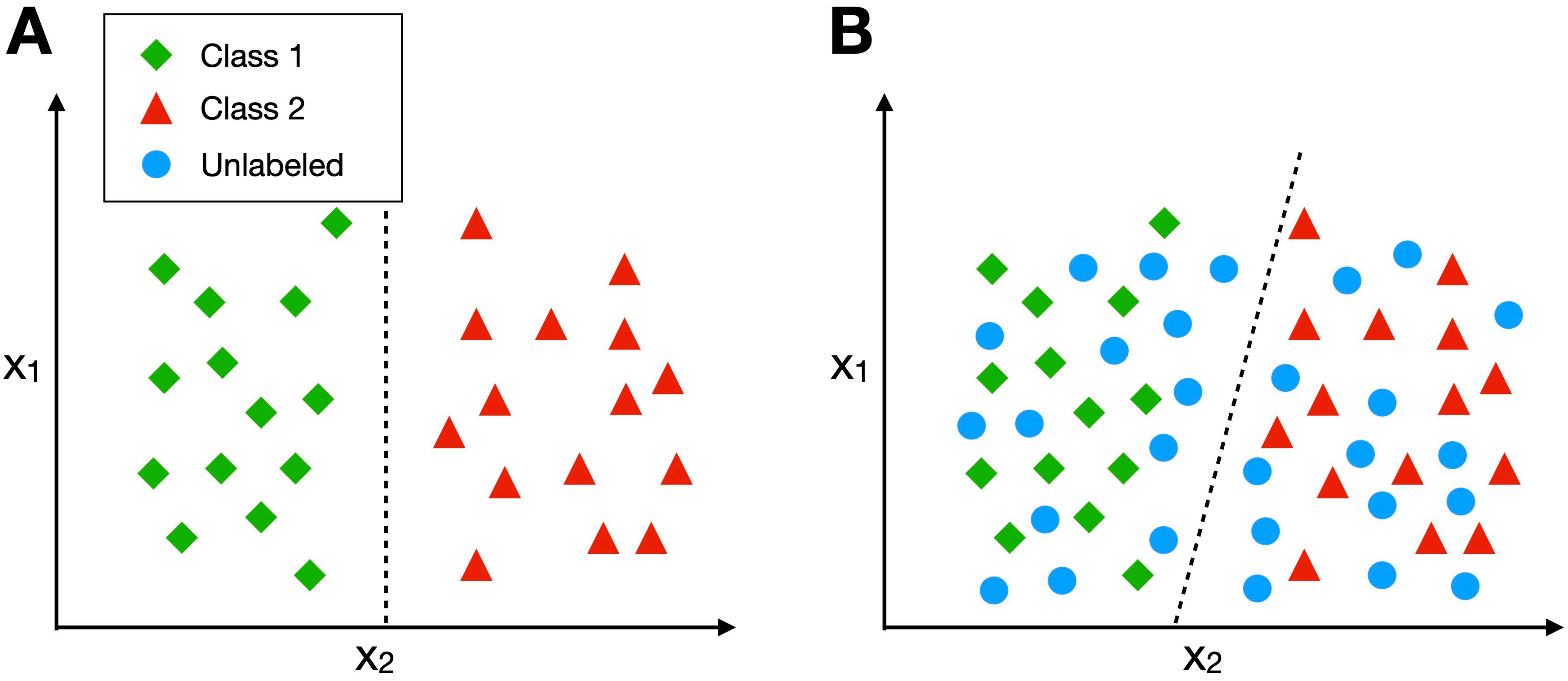
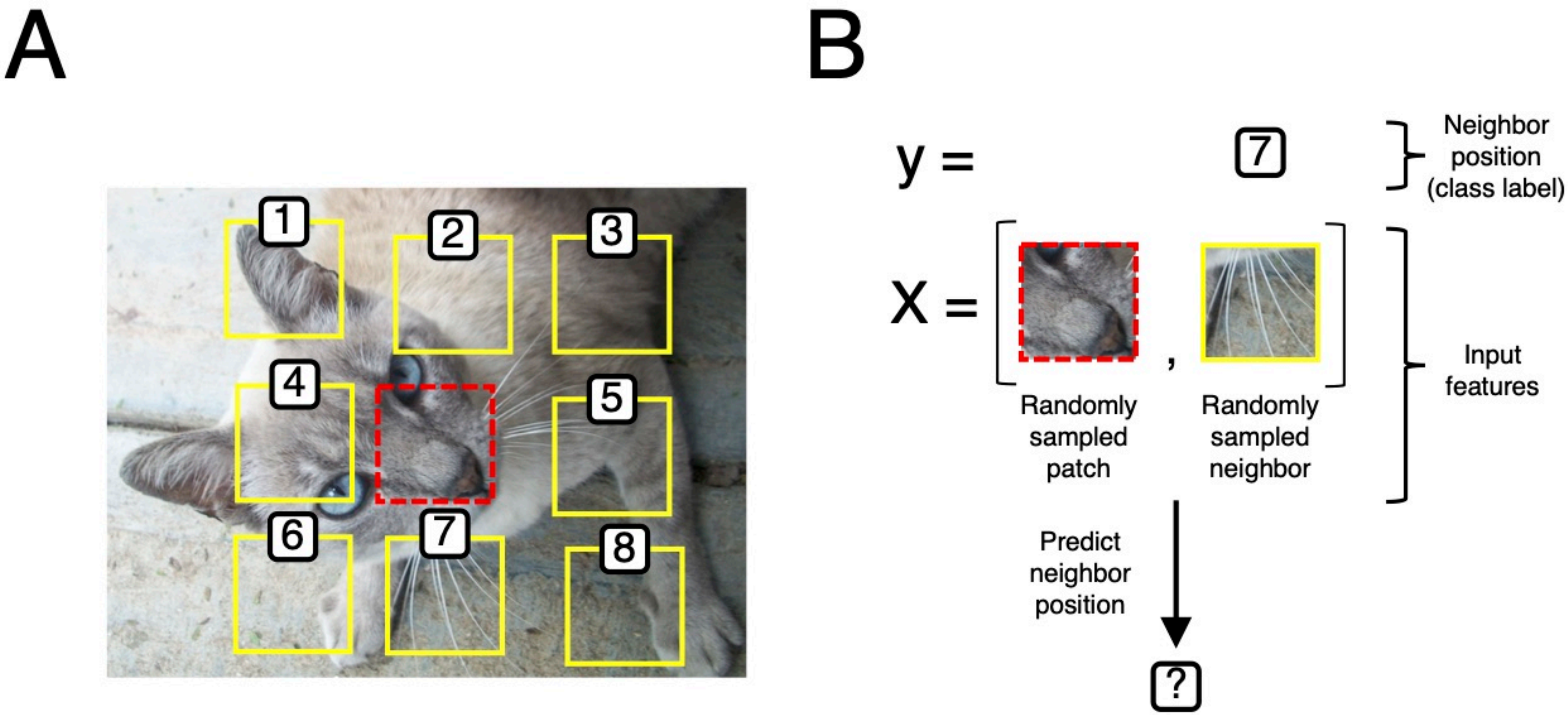


Illustration of semi-supervised learning incorporating unlabeled examples. (A) A decision boundary derived from the labeled training examples only. (B) A decision boundary based on both labeled and unlabeled examples.

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

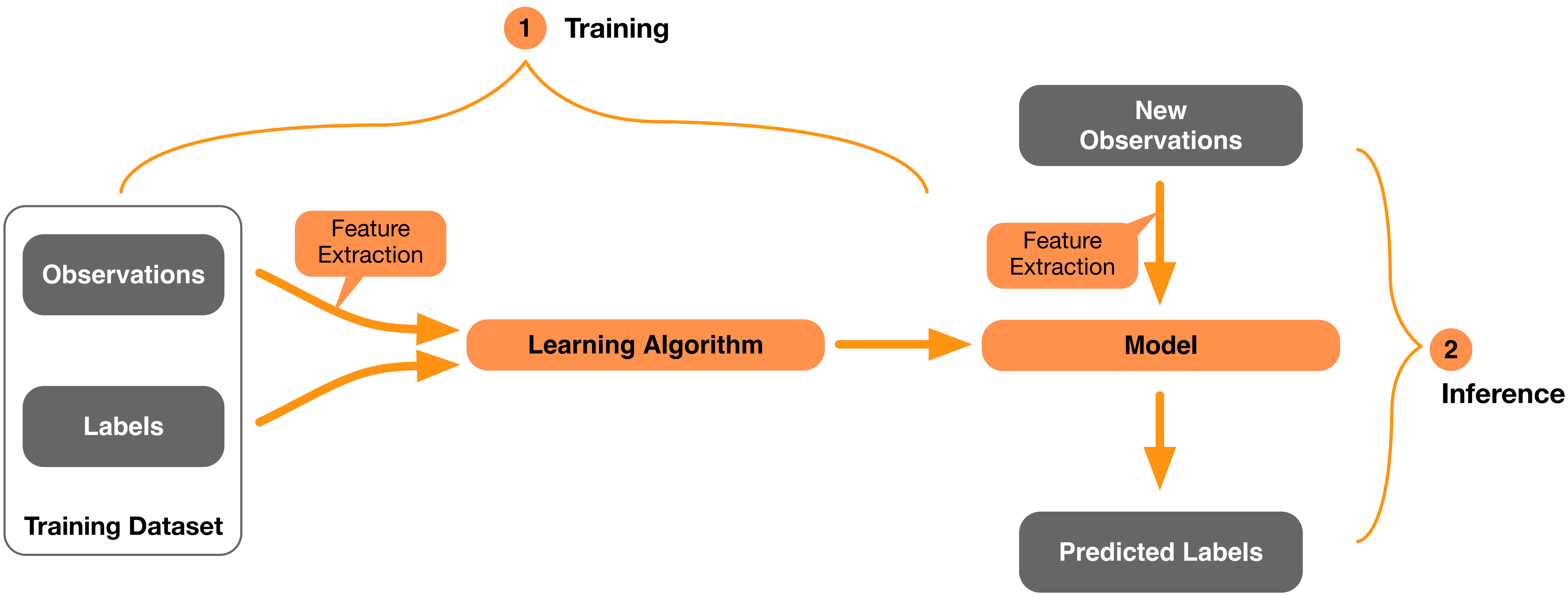


Self-supervised learning via context prediction. (A) A random patch is sampled (red square) along with 9 neighboring patches. (B) Given the random patch and a random neighbor patch, the task is to predict the position of the neighboring patch relative to the center patch (red square).

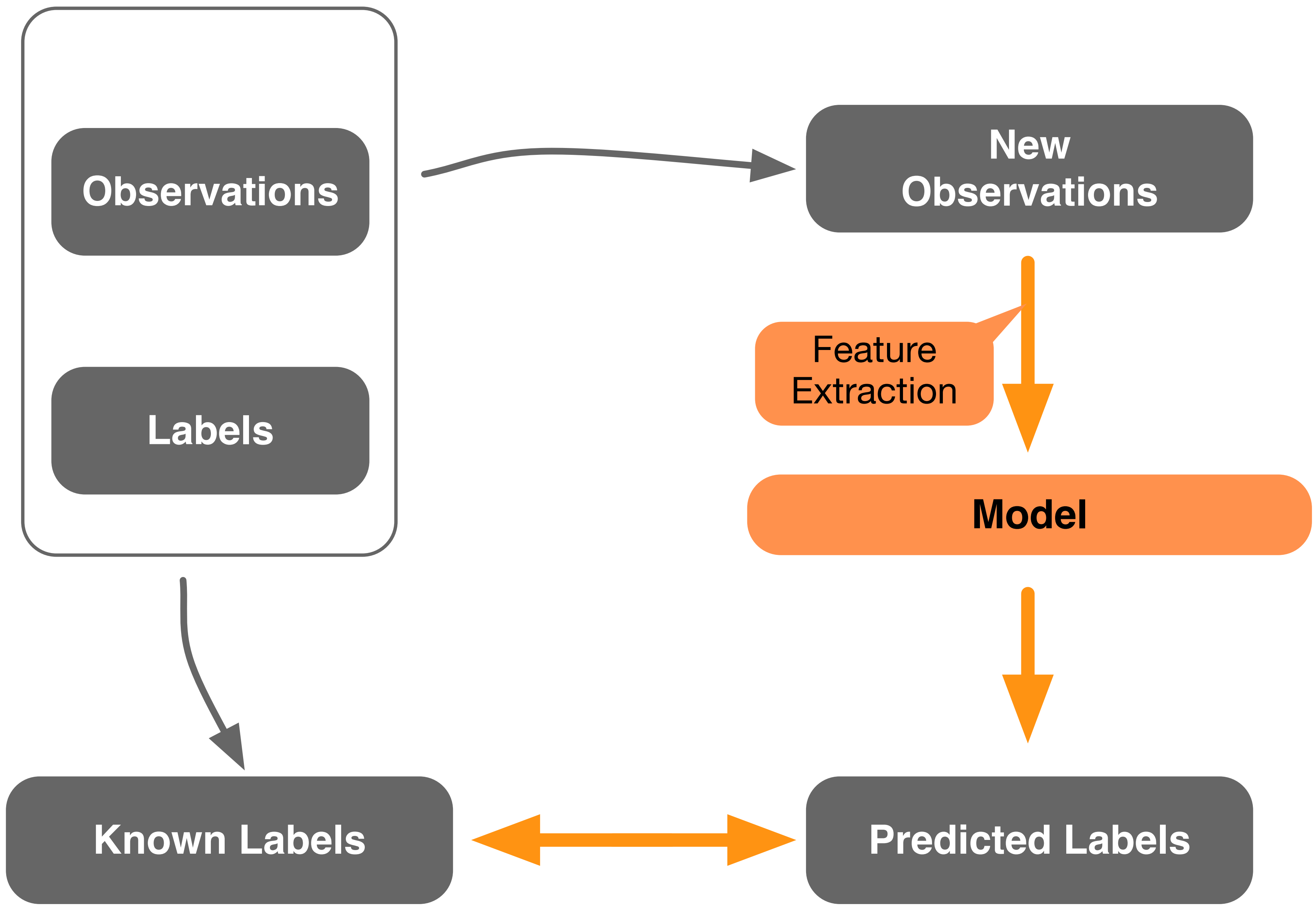
The Supervised Learning Workflow

1. Course overview
2. What is machine learning?
3. The broad categories of ML
4. **The supervised learning workflow**
5. Necessary ML notation and jargon
6. About the practical aspects and tools

Supervised Learning Workflow



Using a test dataset to evaluate the performance of a predictive model



Structured vs Unstructured Data

A

Feature vector of the 1st training example

Class label

Index	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
...	...	...	...	...	...
150	5.9	3	5.1	1.8	Iris-virginica

B

Machine Learning vs Deep Learning

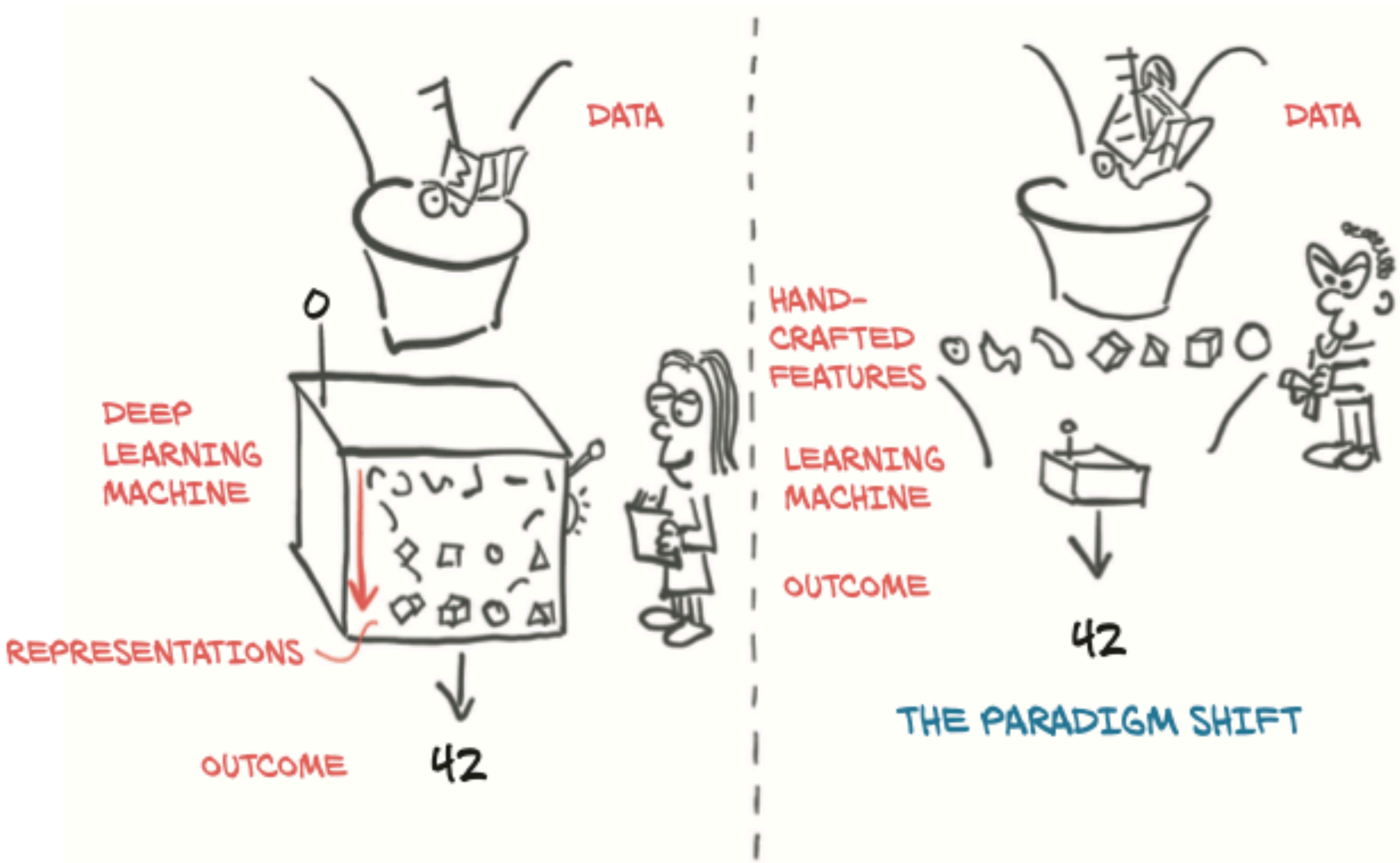
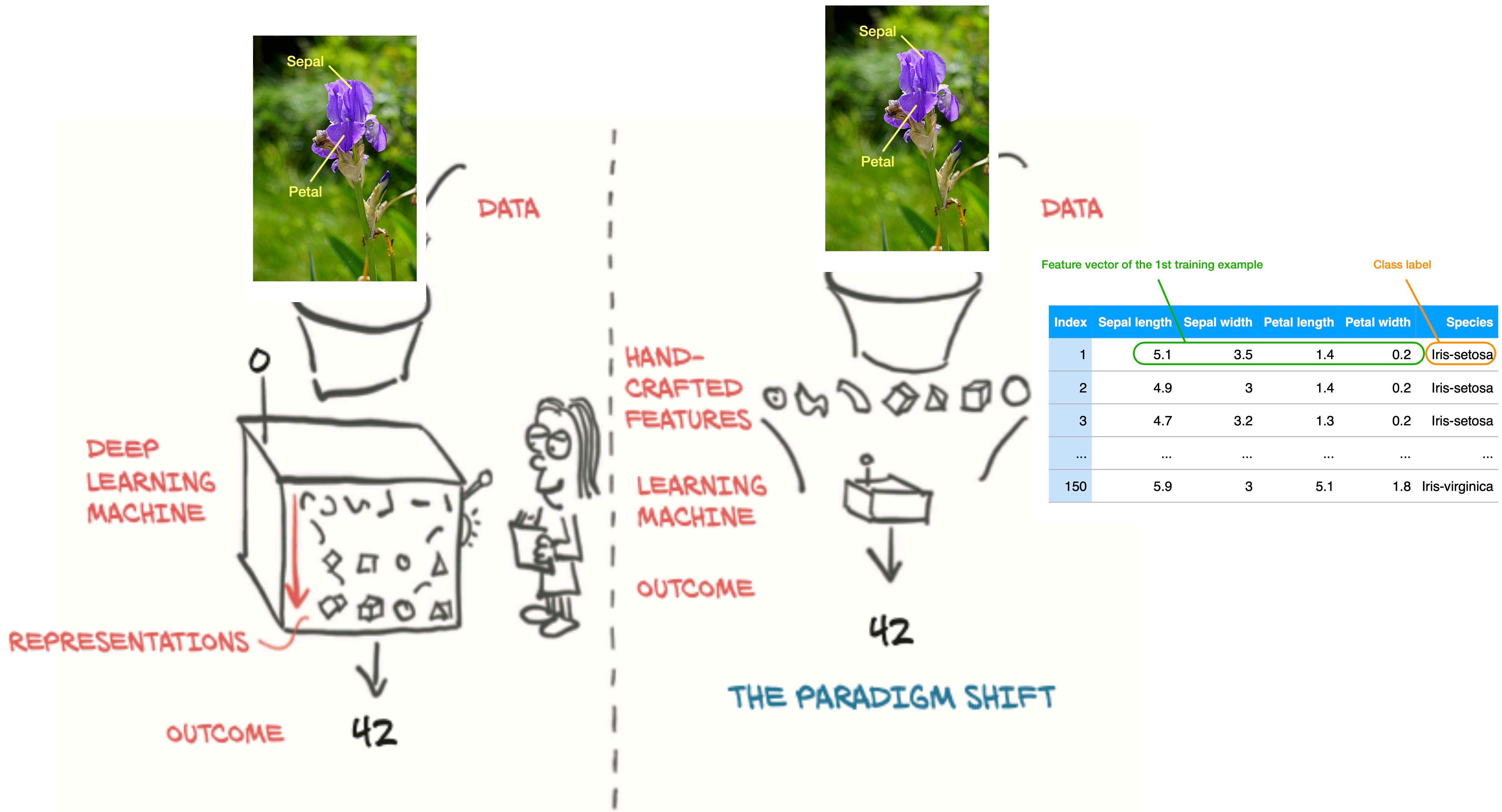


Image source: Stevens et al., *Deep Learning with PyTorch.* Manning*, 2020*

Machine Learning vs Deep Learning



Index	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
...	...	...	...	...	...
150	5.9	3	5.1	1.8	Iris-virginica

Image source: Stevens et al., *Deep Learning with PyTorch.* Manning, *2020*

Machine Learning Terminology and Notation

(Again, this also applies to DL)

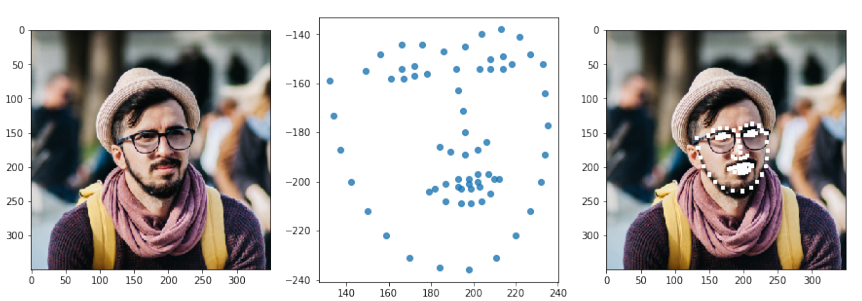
1. Course overview
2. What is machine learning?
3. The broad categories of ML
4. The supervised learning workflow
5. **Necessary ML notation and jargon**
6. About the practical aspects and tools

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
  (before DL, extensive feature engineering was required)



Source: http://rasbt.github.io/mlxtend/user_guide/image/extract_face_landmarks/

Supervised Learning (More Formal Notation)

"training examples"

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

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

Hypothesis: $h(\mathbf{x}) = \hat{y}$ ← sometimes t or o

Classification

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

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

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

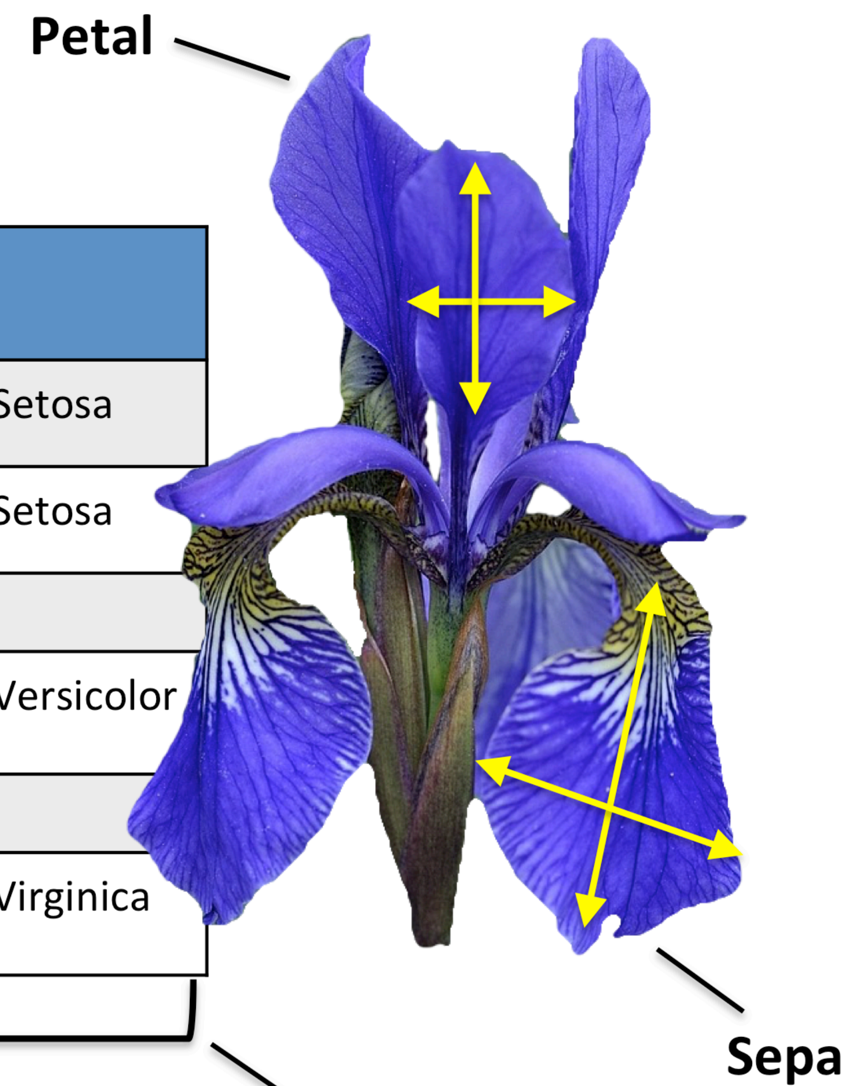
Design Matrix

Data Representation (structured data)

$m =$ ____

$n =$ ____

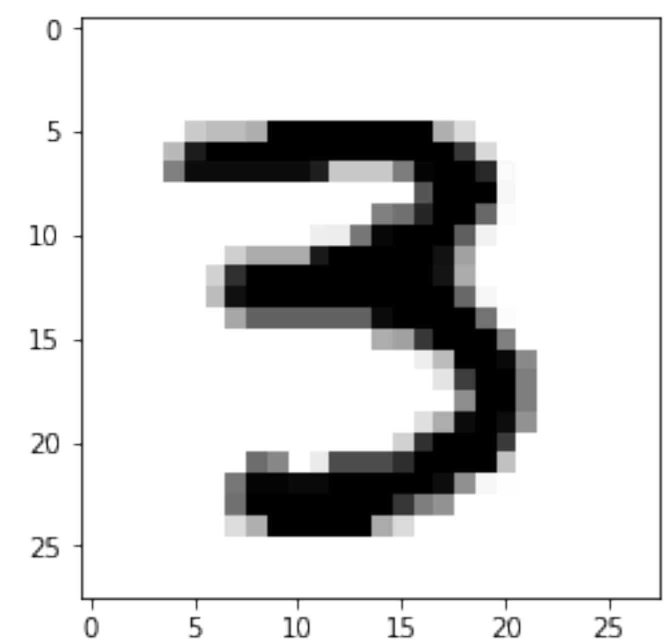
	Sepal length	Sepal width	Petal length	Petal width	
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			…		
50	6.4	3.5	4.5	1.2	Versicolor
			…		
150	5.9	3.0	5.0	1.8	Virginica



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, 0.44705883, 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])
```

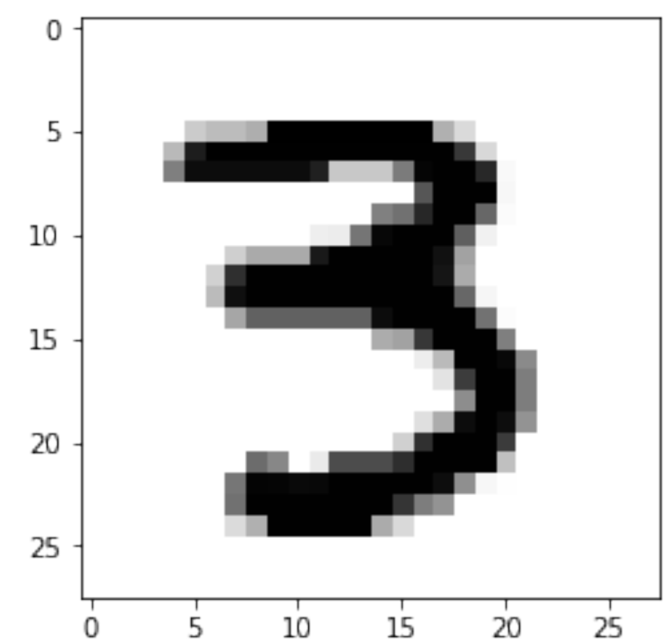
"NCHW" representation (more on that later)

```
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,
          0.9922, 0.9922, 0.6275, 0.0549, 0.0000, 0.0000, 0.0000, 0.0000,
```



Machine Learning Jargon 2/2

- **Training a model** = fitting a model = parameterizing a model = learning from data
- **Training example**, synonymous to
  training record, training instance, training sample (in some contexts, sample refers to a collection of training examples)
- **Feature**, synonymous to
  observation, 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

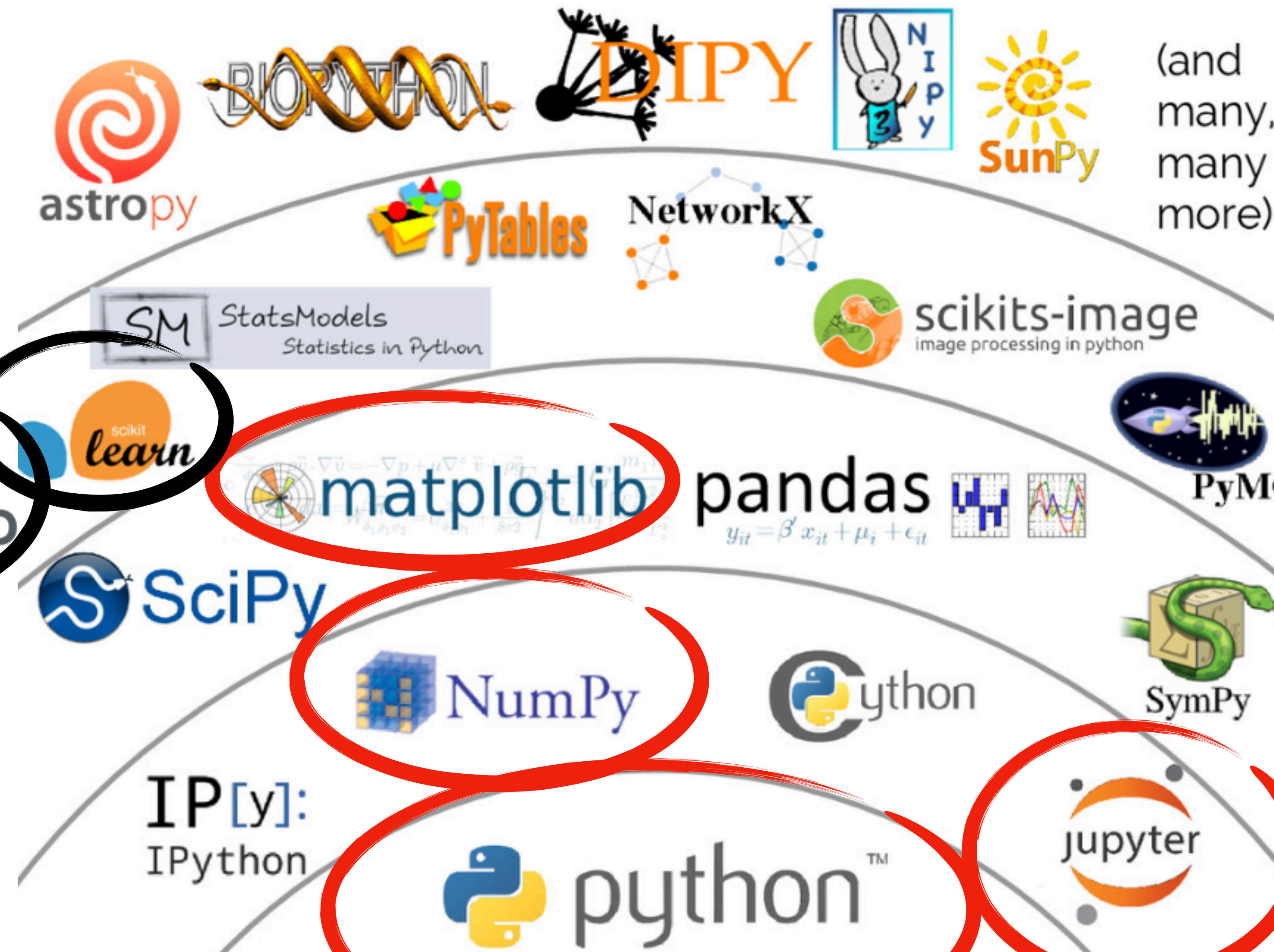
The Practical Aspects: Our Tools!

1. Course overview
2. What is machine learning?
3. The broad categories of ML
4. The supervised learning workflow
5. Necessary ML notation and jargon
6. **About the practical aspects and tools**

Main Scientific Python Libraries

Stat 451 FS2020
(Machine Learning)

MLXTEND

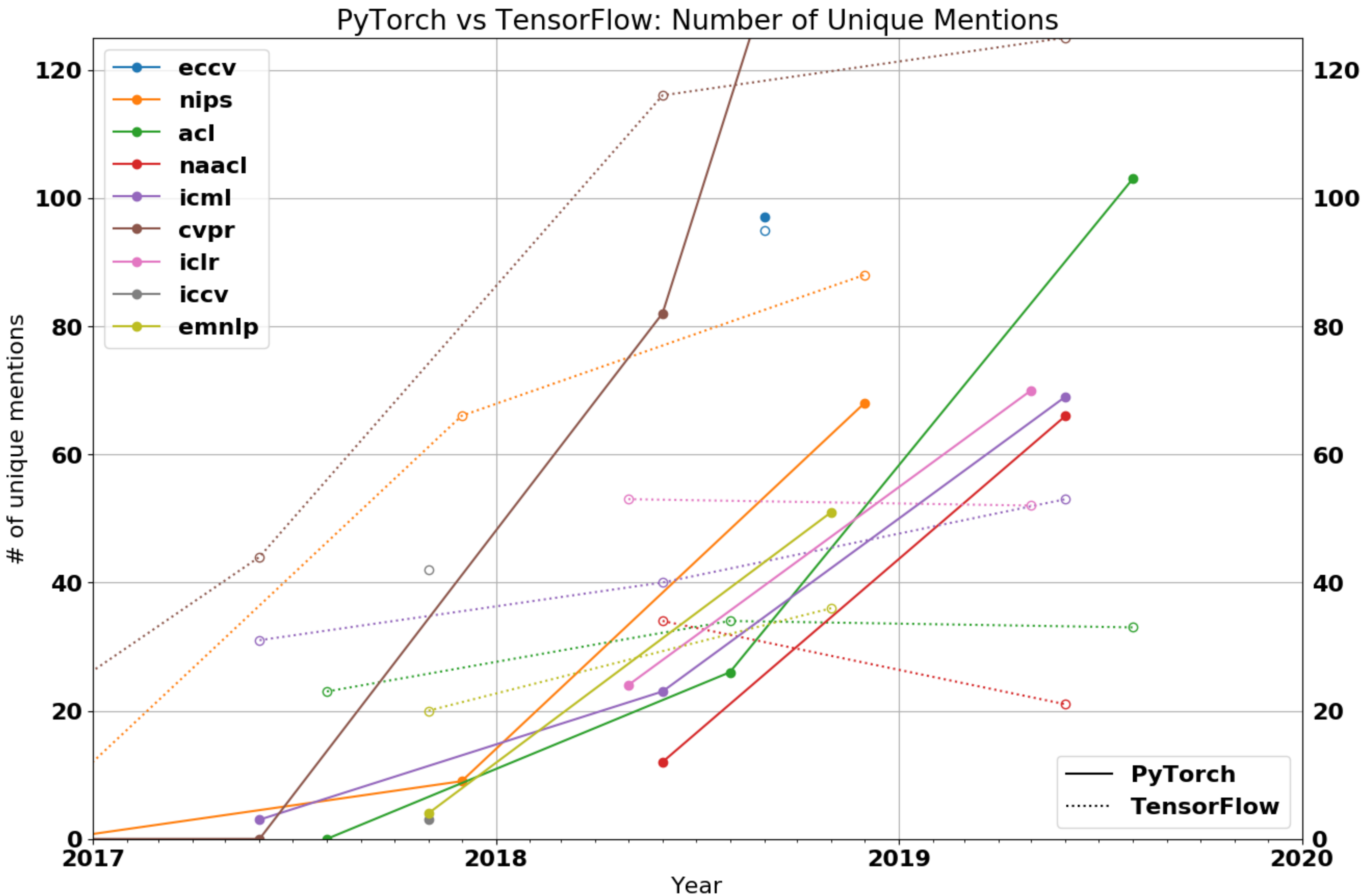


PyTorch

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/

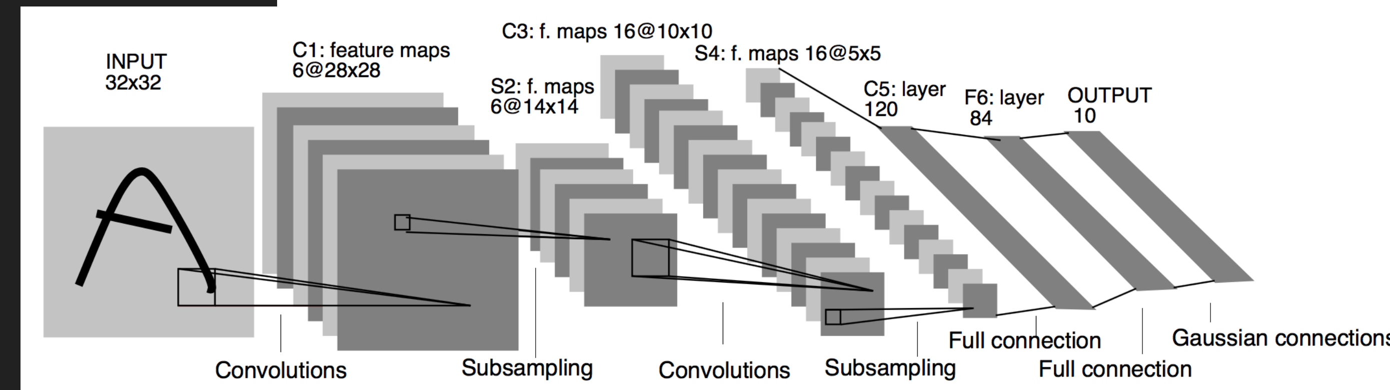
"The State of Machine Learning Frameworks in 2019"

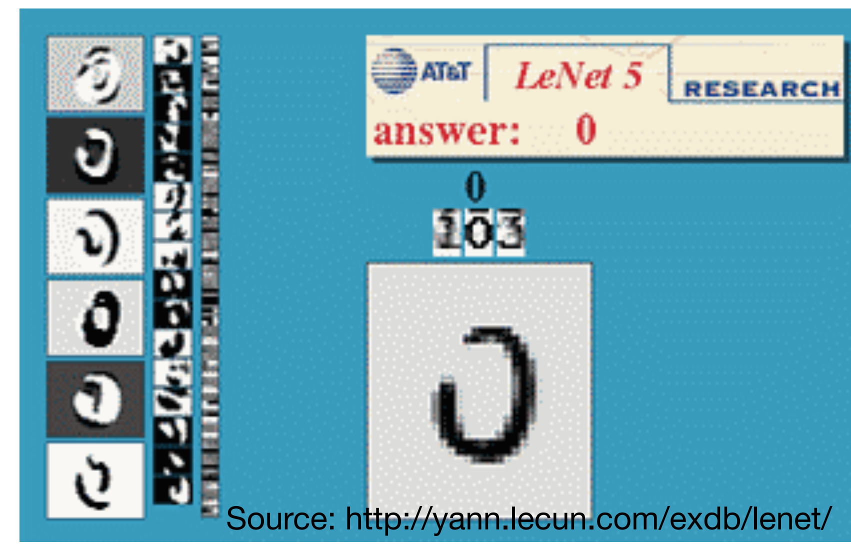
CONFERENCE	PT 2018	PT 2019	PT GROWTH	TF 2018	TF 2019	TF GROWTH
CVPR	82	280	240%	116	125	7.7%
NAACL	12	66	450%	34	21	-38.2%
ACL	26	103	296%	34	33	-2.9%
ICLR	24	70	192%	54	53	-1.9%
ICML	23	69	200%	40	53	32.5%

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/

```python
##########################
# ## MODEL
##########################

class LeNet5(torch.nn.Module):

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

        self.features = torch.nn.Sequential(
            torch.nn.Conv2d(1, 6, kernel_size=5),
            torch.nn.Tanh(),
            torch.nn.MaxPool2d(kernel_size=2),
            torch.nn.Conv2d(6, 16, kernel_size=5),
            torch.nn.Tanh(),
            torch.nn.MaxPool2d(kernel_size=2)
        )

        self.classifier = torch.nn.Sequential(
            torch.nn.Linear(16*5*5, 120),
            torch.nn.Tanh(),
            torch.nn.Linear(120, 84),
            torch.nn.Tanh(),
            torch.nn.Linear(84, num_classes),
        )

    def forward(self, x):
        x = self.features(x)
        x = torch.flatten(x, 1)
        logits = self.classifier(x)
        probas = torch.nn.functional.softmax(logits, dim=1)
        return logits, probas

```





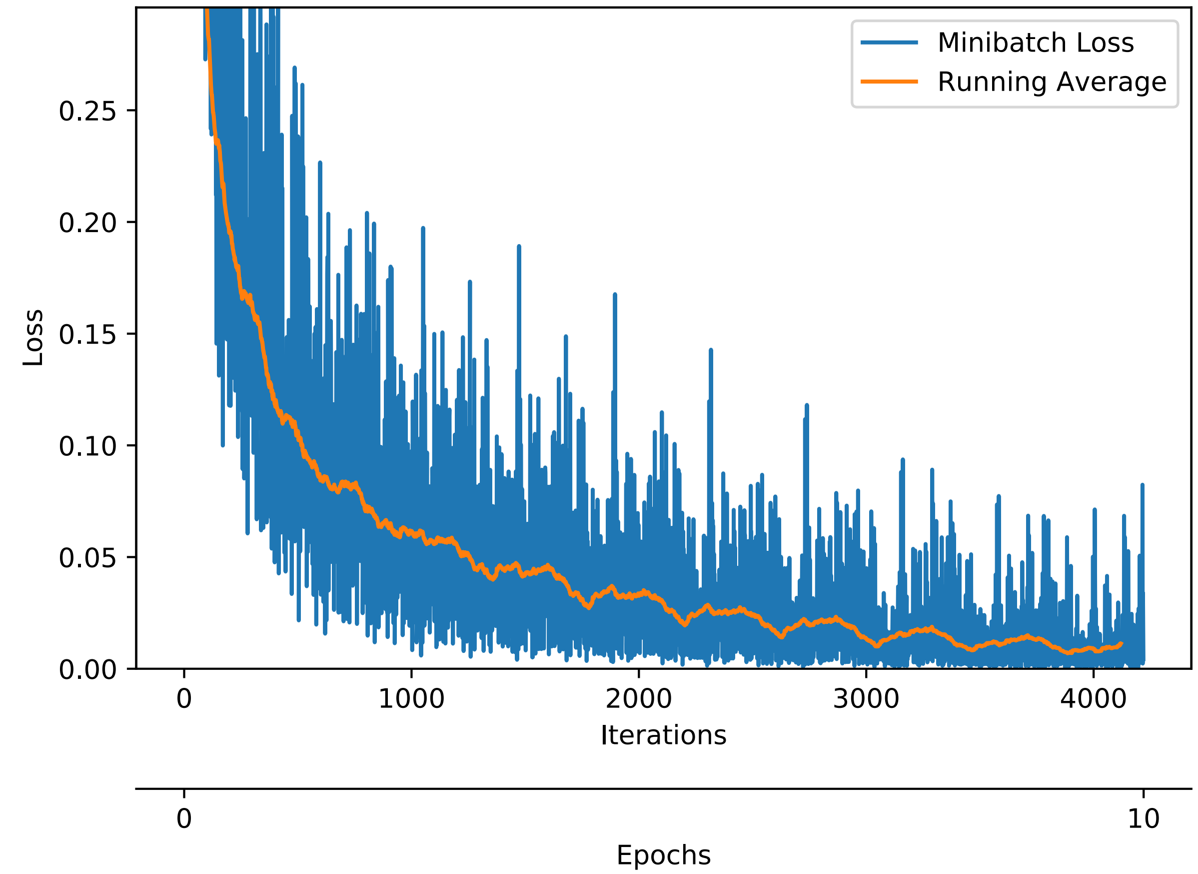
Visual Studio Code

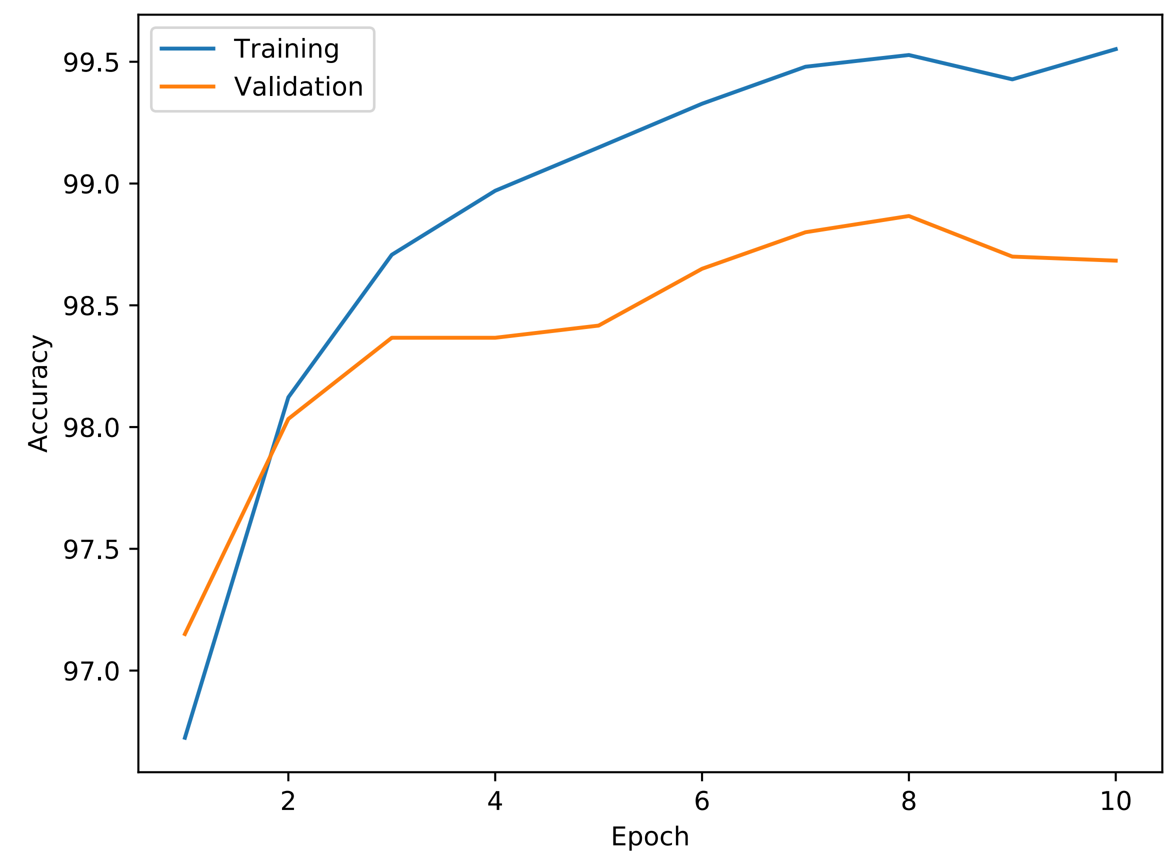
https://code.visualstudio.com

https://github.com/rasbt/stat453-deep-learning-ss21/tree/main/L01/code

```
(base) raschka@lambda-quad:~/code/stat453-ss21-exp$ python simple_cnn.py
PyTorch version: 1.7.0
Using cuda:0
[W Context.cpp:69] Warning: torch.set_deterministic is in beta, and its de
on operator())
Epoch: 001/010 | Batch 0000/0422 | Loss: 2.2935
Epoch: 001/010 | Batch 0050/0422 | Loss: 0.5462
Epoch: 001/010 | Batch 0100/0422 | Loss: 0.3154
Epoch: 001/010 | Batch 0150/0422 | Loss: 0.2551
Epoch: 001/010 | Batch 0200/0422 | Loss: 0.1792
Epoch: 001/010 | Batch 0250/0422 | Loss: 0.2210
Epoch: 001/010 | Batch 0300/0422 | Loss: 0.1551
Epoch: 001/010 | Batch 0350/0422 | Loss: 0.2155
Epoch: 001/010 | Batch 0400/0422 | Loss: 0.2306
Epoch: 001/010 | Train: 96.72% | Validation: 97.15%
Time elapsed: 0.09 min
Epoch: 002/010 | Batch 0000/0422 | Loss: 0.1028
Epoch: 002/010 | Batch 0050/0422 | Loss: 0.1167
Epoch: 002/010 | Batch 0100/0422 | Loss: 0.0660
Epoch: 002/010 | Batch 0150/0422 | Loss: 0.1024
Epoch: 002/010 | Batch 0200/0422 | Loss: 0.0847
Epoch: 002/010 | Batch 0250/0422 | Loss: 0.0905
Epoch: 002/010 | Batch 0300/0422 | Loss: 0.1024
Epoch: 002/010 | Batch 0350/0422 | Loss: 0.0719
Epoch: 002/010 | Batch 0400/0422 | Loss: 0.1302
Epoch: 002/010 | Train: 98.12% | Validation: 98.03%
Time elapsed: 0.18 min
Epoch: 003/010 | Batch 0000/0422 | Loss: 0.0720
Epoch: 003/010 | Batch 0050/0422 | Loss: 0.0984
Epoch: 003/010 | Batch 0100/0422 | Loss: 0.0373
Epoch: 003/010 | Batch 0150/0422 | Loss: 0.0685
Epoch: 003/010 | Batch 0200/0422 | Loss: 0.0511
Epoch: 003/010 | Batch 0250/0422 | Loss: 0.0617
Epoch: 003/010 | Batch 0300/0422 | Loss: 0.0748
```

```
Epoch: 010/010 | Batch 0000/0422 | Loss: 0.0095
Epoch: 010/010 | Batch 0050/0422 | Loss: 0.0113
Epoch: 010/010 | Batch 0100/0422 | Loss: 0.0135
Epoch: 010/010 | Batch 0150/0422 | Loss: 0.0028
Epoch: 010/010 | Batch 0200/0422 | Loss: 0.0019
Epoch: 010/010 | Batch 0250/0422 | Loss: 0.0049
Epoch: 010/010 | Batch 0300/0422 | Loss: 0.0132
Epoch: 010/010 | Batch 0350/0422 | Loss: 0.0114
Epoch: 010/010 | Batch 0400/0422 | Loss: 0.0270
Epoch: 010/010 | Train: 99.55% | Validation: 98.68%
Time elapsed: 0.88 min
Total Training Time: 0.88 min
Test accuracy 98.66%
```





Further Resources and Reading Materials

- "Introduction to Machine Learning and Deep Learning", article based on these slides https://sebastianraschka.com/blog/2020/intro-to-dl-ch01.html
- STAT451 FS2021: Intro to machine Learning, lecture notes: https://github.com/rasbt/stat451-machine-learning-fs20/blob/master/L01/01-ml-overview__notes.pdf
- *Python Machine Learning*, 3rd Ed. Packt 2019. *Chapter 1.*

Next Lecture:

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