CS 559 Machine Learning

Introduction and Overview

Yue Ning
Department of Computer Science
Stevens Institute of Technology
Email: *yue.ning@stevens.edu*
https://yue-ning.github.io

Course Information

- Prerequisite course
  - MA 222 - Probability Theory
  - CS 556 - Mathematical Foundations of Machine Learning
- Meeting
  - Time: 3:30 PM - 6:00 PM
  - Date: Thursday
  - Location: Babbio 122
- Canvas: Announcements, Assignments, Discussions.
- How to find Canvas: login to `my.stevens.edu`

Instructor and TA Information

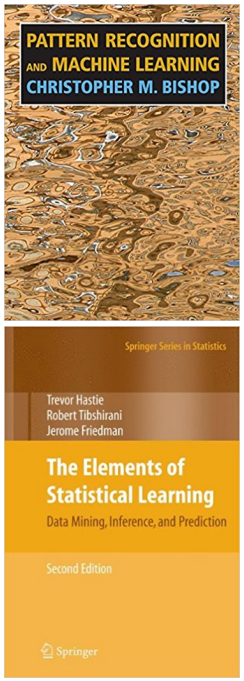
- Instructor: Yue Ning
  - Email: Yue.Ning@stevens.edu
  - Office hours: Thursday 1:00 pm - 3:00 pm
  - Office: Gateway South 448
  - You can also request Zoom
- Teaching Assistant: Bethel Hall

  - Email: bhall2@stevens.edu
  - Office hours: Wednesday 10am - 12pm
  - Office location: TBA
  - Zoom: `https://stevens.zoom.us/j/6482717382`

Textbooks

Recommended:

- Bishop, Christopher M., 2006.
  Pattern Recognition and Machine Learning.
  Springer-Verlag New York, Inc.
- Hastie, Trevor and Tibshirani,
  Robert and Friedman, Jerome,
  2001.
  The Elements of Statistical Learning.
  Springer New York Inc.
- (optional) Chris Bishop and
  Hugh Bishop, 2023.
  Deep Learning, Foundations and Concepts.
  Springer.



Course Prerequisites and Goals

Before this course, you should know.....

- Programming (Python)
- Linear Algebra (Vector, Matrix, Projection, Eigenvalues/vector)
- Probability and Optimization (distributions, expectation/variance, objective function)

At the end of this course, you should.....

- Be more proficient at math and programming
- Be able to recognize when and how a new problem can be solved with an existing technique
- Be able to implement general ML techniques for a variety of problem types

Alert

This class is not about:

- Introduction to machine learning tools/software

Coursework

Distribution

- Homework (20%)
- Exam (40%)
- Final Project (30%)
- Participation (5%)
- Quizzes (5%)

Policy

- Late days: maximum of two (2) late days in total without penalty.
- Late submissions: For those who have used up their late days: If you submit solutions late within 48 hours, you will receive 30% penalty. If you submit late within 48-72 hours, you will receive 50% penalty. After 72 hours, you will receive zero points.
- Double-check: You need to double-check your submissions after you upload them on Canvas.
- Questions: Check the lecture materials before you ask a question. **Use Canvas**: ask questions on Canvas. Some other students may know the answer!

Homework

- Goal: test your ability to understand knowledge of ML
- Format: mix of written and programming problems
- How to submit: Canvas (upload your e-copies)
- Programming Language: we recommend Python
- Jupyter Notebook: it is recommended to use for your programming assignments.

Exam

- Goal: test your ability to use knowledge of ML to solve new problems
- Format: all written problems. Similar to the written part of the homework
- Closed book: cheat sheets allowed
- Covers all material up to the preceding week
- Time: later part of the semester

Course Project

- Goal: select any problem you are interested and apply ML techniques to tackle it.
- Milestones: proposal report, final report and a 10-min presentation!
- Task is open, please follow: task definition, literature review, implement methods, evaluation, result analysis.
- Help: come to any office hours, TA, Internet.
- Submission: reports and code.

Plagiarism: Zero Tolerance

- Collaborate and discuss, but write code and reports independently.
- Do not look at classmates' writeup or code
- Do not share writeup/code (online and physical) ANYTIME
- Debugging: only look at input-output behavior
- We will detect plagiarism

Copyright

All course materials have copyright;
DO NOT share outside this classroom.

Coursework grading

Distribution of (0-100)

- A $(90 - 100)$
- A- $(85 - 90)$
- B+ $(80 - 85)$
- B $(75 - 80)$
- B- $(70 - 75)$
- C+ $(65 - 70)$
- C $(60 - 65)$
- F (< 60)

Coursework grading

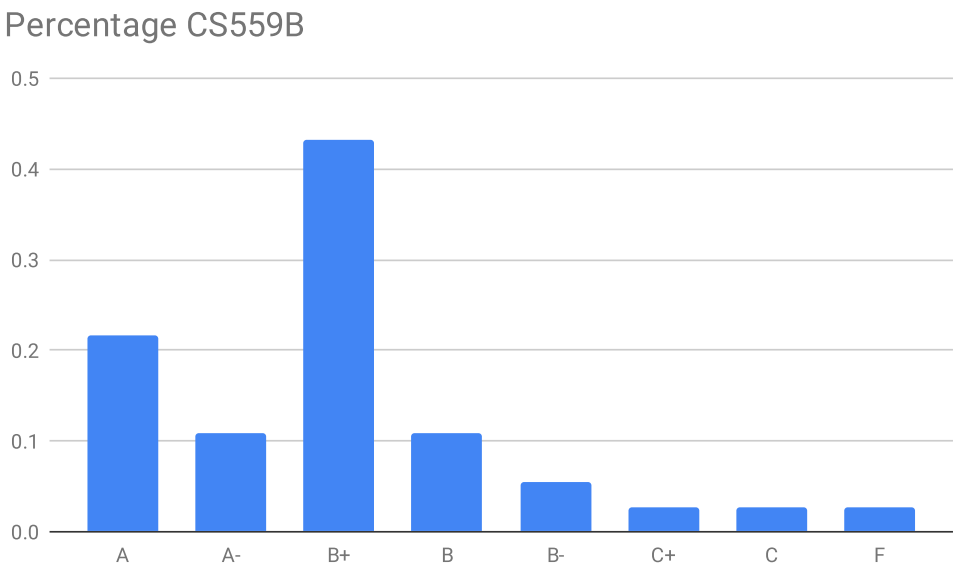


Figure: Students Performance Spring 2019

How to fail this course?

How to fail this course?

1. Miss the first homework assignment

How to fail this course?

2. Submit assignments late

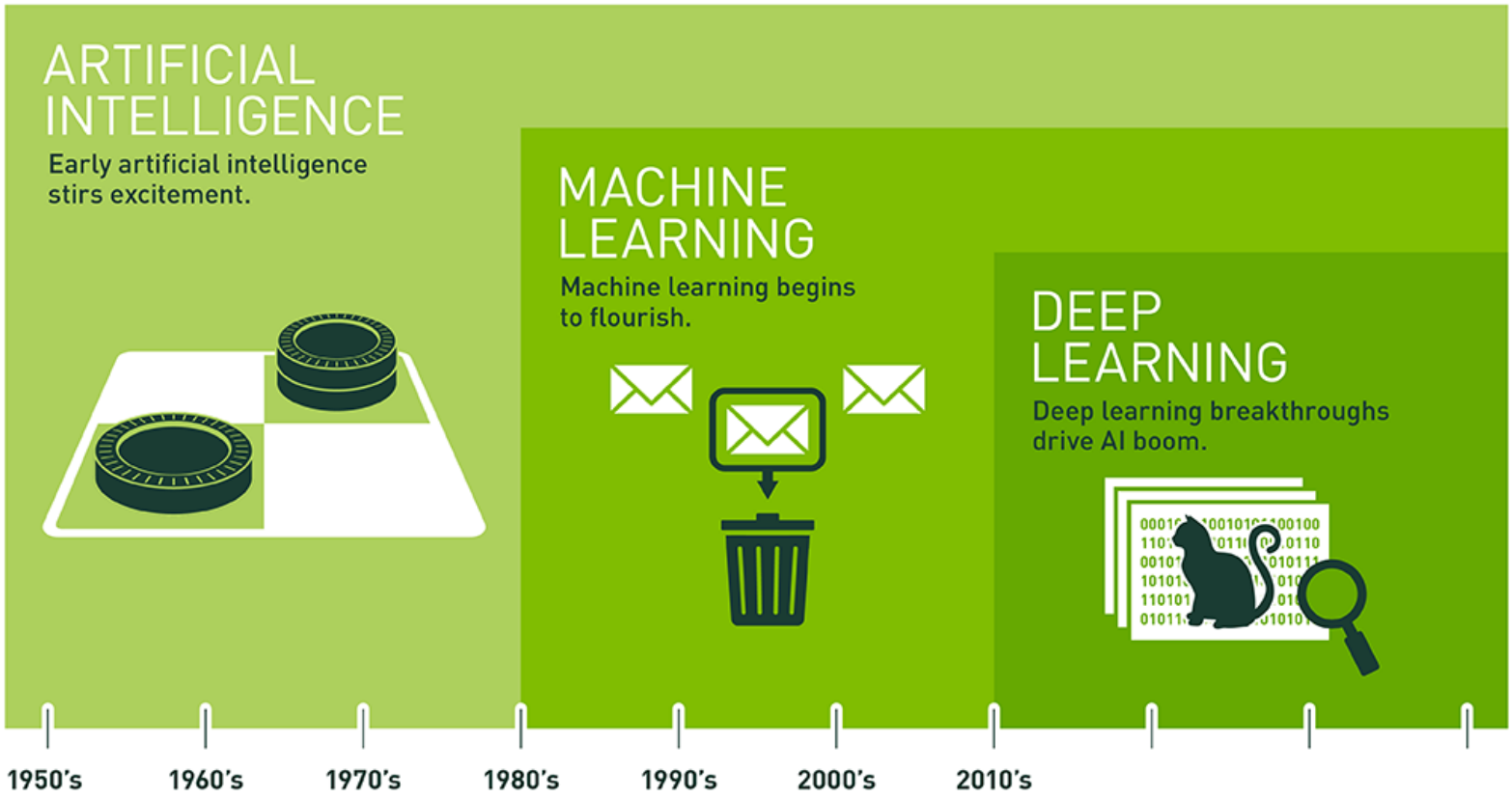
How to fail this course?

3. Miss classes

How to fail this course?

4. Miss exams

AI and Machine Learning



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Figure: Nvidia blog

What is Machine Learning for?

General Machine Learning Problems

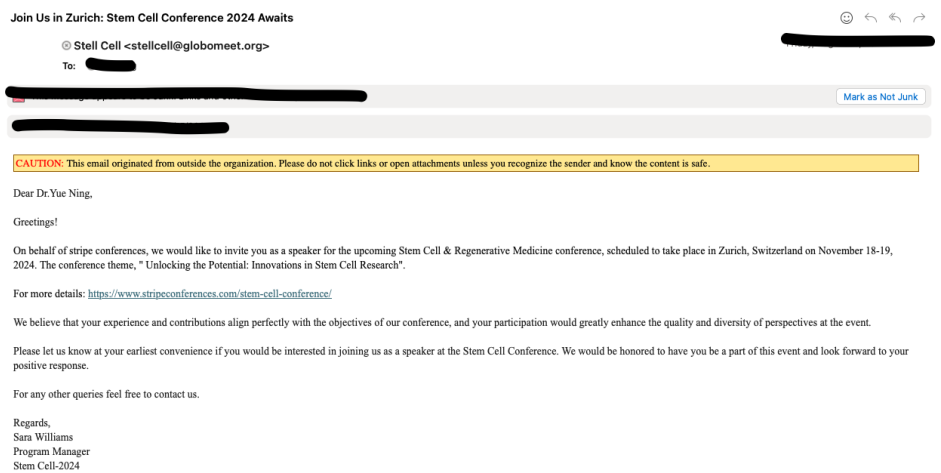
A prediction problem:

- Given an input x,
- Predict an "appropriate" output y.

Let's look at a few examples.

Example: Email Spam Prediction

- **Input**: an email



Join Us in Zurich: Stem Cell Conference 2024 Awaits

Stell Cell <stellcell@globomeet.org>

To: [redacted]

[redacted] Mark as Not Junk

[redacted]

CAUTION: This email originated from outside the organization. Please do not click links or open attachments unless you recognize the sender and know the content is safe.

Dear Dr.Yue Ning,

Greetings!

On behalf of stripe conferences, we would like to invite you as a speaker for the upcoming Stem Cell & Regenerative Medicine conference, scheduled to take place in Zurich, Switzerland on November 18-19, 2024. The conference theme, " Unlocking the Potential: Innovations in Stem Cell Research".

For more details: https://www.stripeconferences.com/stem-cell-conference/

We believe that your experience and contributions align perfectly with the objectives of our conference, and your participation would greatly enhance the quality and diversity of perspectives at the event.

Please let us know at your earliest convenience if you would be interested in joining us as a speaker at the Stem Cell Conference. We would be honored to have you be a part of this event and look forward to your positive response.

For any other queries feel free to contact us.

Regards,
Sara Williams
Program Manager
Stem Cell-2024

- **Output**: "Spam" (1) or "Not Spam" (0)
- A **binary classification** (only 2 possible outputs)

Example: Medical Diagnosis

- **Input**: Symptoms (fever, cough, fast breathing, shaking, nausea,...)
- **Output**: Diagnosis (pneumonia, flu, common cold, ...)
- A **multiclass classification** problem: choosing one of several (discrete) outputs.

How to express uncertainty?

- Probabilistic classification or soft classification:

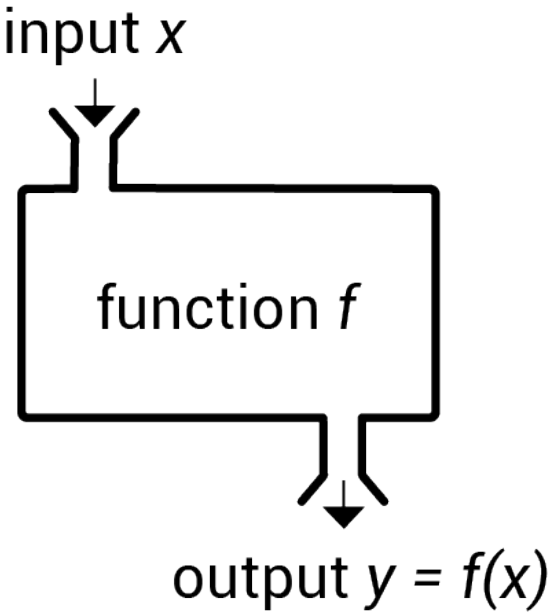
$$\begin{aligned} \mathbb{P}(\text{pneumonia}) &= 0.7 \\ \mathbb{P}(\text{flu}) &= 0.2 \\ \vdots \quad & \quad \vdots \end{aligned}$$

Example: Stock Price

- **Input**: History of stock's prices & news articles
- **Output**: Stock price at close of next day
- A **regression** problem (output is a number)

The Prediction Function

- A **prediction function**
  - takes input x and
  - produces an output y



- We're looking for prediction functions that solve particular problems.
- Machine learning helps find the best prediction function.

What is Machine Learning?

What is Not Machine Learning: Rule-based Methods

- Consider medical diagnosis
  - Build "Experts": Consult textbooks and medical doctors

  - Understand their diagnosis process
  - Implement this as an algorithm (a **"rule-based system"**)
- Does not sound too bad
- Popular in the 1980s
- Note: "experts" can be sophisticated than they sound here.

Rule-based Methods

Issues with rule-based systems:

- Very labor intensive to build
- Rules cannot generalize to new input
- Do not handle uncertainy well
- Expert systems seen as “brittle”

Modern AI: Machine Learning

- Machine learns on its own
- We provide **training data**, i.e. examples of (input x, output y) pairs.
  - e.g., A set of images, and whether or each contains a cat
  - e.g., A set of emails, and whether or not each is a SPAM
- Learning from training data of this pair form is called supervised learning

Machine Learning Process

A Machine Learning algorithm:

- **Input**: Training data
- "**Learns**" from training data
- **Output**: A "prediction function" that produces output y given input x

What is Machine Learning?

"Any regularity in the data can be used to compress the data, i.e. to describe it using fewer symbols than the number of symbols needed to describe the data literally. The more regularities there are, the more the data can be compressed. Equating 'learning' with 'finding regularity', we can therefore say that the more we are able to compress the data, the more we have learned about the data"

— Peter Grünwald

A Tutorial Introduction to the Minimum Description Length Principle

What is Machine Learning?

- **Machine Learning**: *A field of study that gives computers the ability to learn without being explicitly programmed.* – Arthur Samuel (1959).
- Automatically learn patterns from empirical data in order to improve prediction performance.
- An **interdisciplinary** (and relatively young) field focusing both on theoretical foundations of systems that learn, reason, and act as well as on practical applications of these systems.

Machine Learning for Speech Recognition

You said "Nice to meet you"

You said "I am glad we meet"

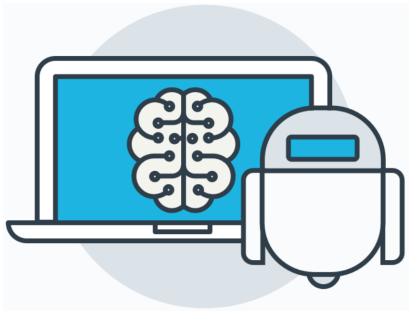
.

.

.

Training audio data

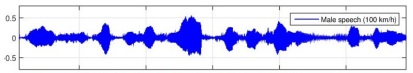
→



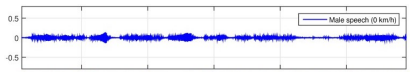
A machine learning model

→

"How is it going?"



"Glad to see you"



Test audio data

Machine Learning for Image Recognition

This is "cat"

This is "dog"

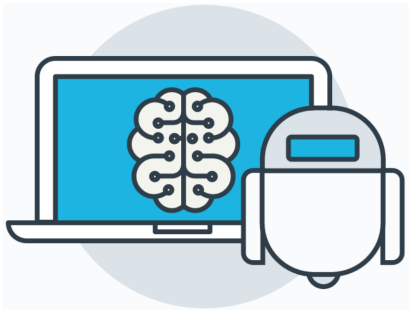
.

.

.

Training image data

→



A machine learning model

→

"Monkey"

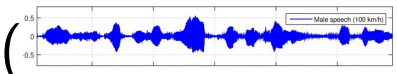
"Dog"

"Cat"

Test image data

Machine Learning $\approx$ Looking for a function

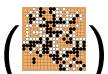
- Speech recognition

$$f(\text{  }) \rightarrow \text{“Nice to meet you”}$$

- Image recognition

$$f(\text{  }) \rightarrow \text{“Cat”}$$

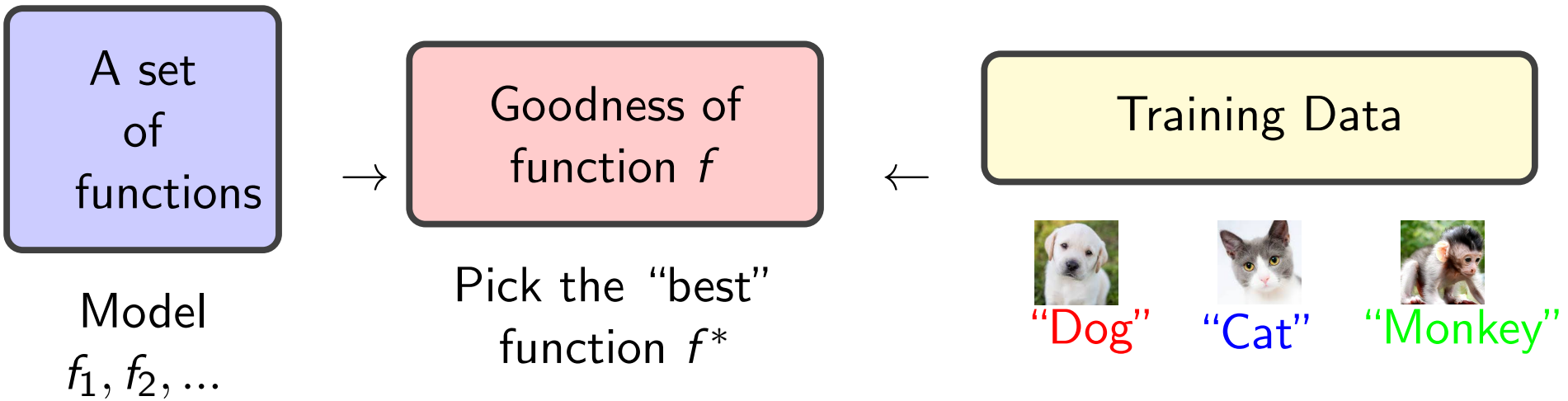
- Playing Go

$$f(\text{  }) \rightarrow \text{“5-5” (next move)}$$

- Dialogue System

$$f(\text{“Hi”}) \rightarrow \text{“Hello”}$$

Training


A set of functions
Model
$f_1, f_2, \ldots$
Goodness of function f
Pick the "best" function f^*
Training Data
"Dog"
"Cat"
"Monkey"

Inference

→ Using f^* → "Cat"

Inference (Test) process

Elements of ML Pipeline

- Raw input:
  - Text (a document, a paragraph, a sentence)
  - Time series (prices, measurements, etc)
  - Image files
  - Audio files
  - Videos
  - DNA sequences
- But most ML prediction functions need input as
  - fixed-length arrays of numbers
  - $\textsc{double}[\textsc{d}]$ for computer programmers
  - $\mathbb{R}^d$ for mathematicians

Feature Extraction

Definition

Mapping raw input to a vector $\mathbb{R}^d$ is called feature extraction or featurization.

- Better features $\Rightarrow$ less "smart" ML needed (makes things easier)
- Feature vectors are often called input vectors

Example: Detecting Email Addresses

- Task: Predict whether a string is an email address
- Use some domain knowledge to build features:

length $>$ 10	:	1
fracOfAlpha	:	0.85
contains(@)	:	1
endsWith(".com")	:	1
endsWith(".org")	:	0

- This is a bit ad-hoc. Can we be more systematic?

Feature Template: Last three characters equal

- Don't think about which 3-letter suffixes are meaningful...
- Just include them all:

endsWith(".aaa")	:	0
endsWith(".aab")	:	0
...		
endsWith(".org")	:	1
...		
endsWith(".zzz")	:	0

One-hot encoding

- One-hot encoding: a set of binary features that always has exactly one nonzero value.
- categorical variable: a variable that takes one of several discrete possible values: NYC Boroughs: "Brooklyn", "Bronx", "Queens", "Manhattan", "Staten Island"
- Categorical variables can be encoded numerically using one-hot encoding. In statistics, called a dummy variable encoding

Concept Check: How many features to one-hot encode the boroughs?

Labeled Data

- Package feature vectors together with output "labels":

Feat 1	Feat 2	...	Feat D	Y
0	1.54	...	932	0
1	-1.9	...	300	1
0	2.3	...	0	0

Figure: A set of examples with labels

- Each row is an "example" or "labeled datum".
- The last column is the output or "label" column.

Unlabeled Data

- Just feature vectors:

Feat 1	Feat 2	...	Feat D	Y
0	1.54	...	932	?
1	-1.9	...	300	?
0	2.3	...	0	?

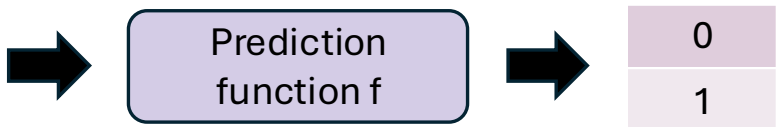
Figure: A set of examples without labels

- We want to be able to predict the missing labels.

Prediction Functions

- A prediction function has
  - Input: a feature vector
  - Output: a "label" (or prediction, output, response)

Feat 1	Feat 2	...	Feat D
0	1.54	...	932
1	-1.9	...	300
0	2.3	...	0



Y
0
1
0

Figure: Prediction Function

- The prediction function is what gets deployed

Learning algorithm

- A (supervised) learning algorithm has
  - Input: labeled training data (i.e., pairs of inputs and outputs)
  - Output: a prediction function

Feat 1	Feat 2	...	Feat D	Y
0	1.54	...	932	0
1	-1.9	...	300	1
0	2.3	...	0	0

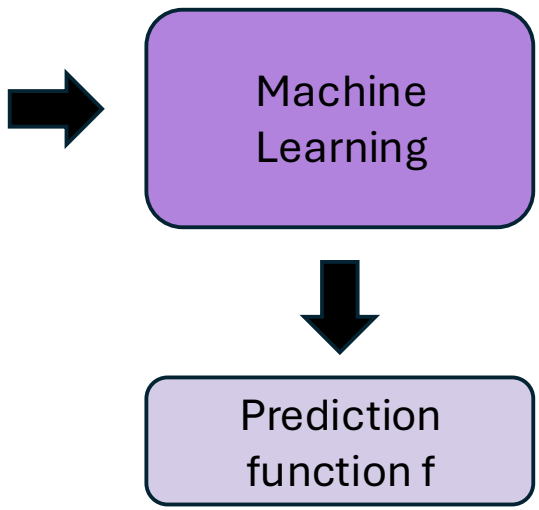


Figure: Supervised Learning Process

- Today is about what's outside the "purple box". Rest of course is about the inside.

Evaluating a Prediction Function

- Brilliant intern gives you a prediction function.

- How do we evaluate performance?
- Very important part of machine learning.
  - It can be subtle
  - Evaluation should reflect (business) goals as closely as possible.

Evaluating a Single Prediction: The Loss Function

- A loss function scores how far off a prediction is from the desired "target" output.
  - loss(prediction,target) returns a number called "the loss"
  - Big loss = Bad error
  - Small loss = Minor error
  - Zero loss = No error

Classic Loss Function

- Classification loss or "0/1 loss"
  - loss = 1 if prediction is wrong
  - loss = 0 if prediction is correct
- Square loss for regression
  - loss = (predicted - target)^2

Evaluating a Prediction Function

- Intern gives you a prediction function f (x).
  - "Average classification loss on training data was 0.01" (i.e. 1% error)
- Product Manager says "we can deploy if $\leq 2\%$ error."
- Deploy this prediction function?
  - No!
- Prediction function needs to do well on new inputs.
- (Don't test somebody with problems they've seen in advance.)

The Test Set

- A "test set" is labeled data that is independent of training data.
  - e.g. Split labeled data randomly into 80% training and 20% test.
- Training set: only for training prediction functions.
- Test set: only for for assessing performance.
- Larger test set gives more accurate assessment of performance.
- How big? We can review "confidence intervals" from statistics.

Train/Test vs Train/Deploy

- Train/Test (in classroom or lab):
  - Build model on training data (say 80% of all labeled data).
  - Get performance estimate on test data (remaining 20%).
- Train/Deploy (reality):
  - Build model on all labeled data.
  - Deploy model into wild.
  - Hope for the best.
- A large part of real-world machine learning is ensuring that
  - Test performance is a good estimate of deployment performance.
- How can we do this, and what can go wrong?

Main Principal of Train/Test Splitting

- **Train/Test setup should represent Train/Deploy scenario as closely as possible.**
- Random split of labeled data into train/test is usually the right approach.
- But consider time series prediction: 1000 days of historical data
  - Should we randomly split the days into training and test?

Train/Test Split for Time Series

- Consider Train/Deploy scenario: Prediction function trained on days occurring before deployment time period.
- Consider Train/Test scenario with random splitting:
  - Some test days occur before some training days.
  - No good!
- What can go wrong with random splitting of time series?
  - Suppose time series changes slowly over time.
  - To predict at test day d, just predict value at training day closest in time.
  - That trick won't work for very long during deployment
- Create train/test split by splitting in time:
  - Training set is everything before time T
  - Test set everything after time T

Summary: What to Give your Data Science Intern

- Split data into train and test.
- Give training set to intern, you keep the test set.
- Intern gives you a prediction function.
- You evaluate prediction function on test set.
- No matter what intern did with training set,
  - test performance should give you good estimate of deployment performance.

What Should the Intern Do?

- Intern wants to try many fancy ML models.
- Each gives a different prediction function.
- Intern needs her own test set to evaluate prediction functions.
- Intern should randomly split data again into
  - Training set and
  - Validation set
- This split could again be 80/20.
- Validation set is like test set, but used to choose best among many prediction functions.
- Test set is just used to evaluate the final chosen prediction function.

k-Fold Cross Validation

K-fold Cross-Validation on N training examples

- Create K equal sized partitions of the training data
- Each partition has N/K examples
- Train using $K-1$ partitions, validate on the remaining partition
- Repeat the same K times, each with a different validation partition
- Choose the model with the smallest average validation error
- Usually K is chosen as 10

k-Fold Cross Validation

K-fold Cross-Validation Visualization

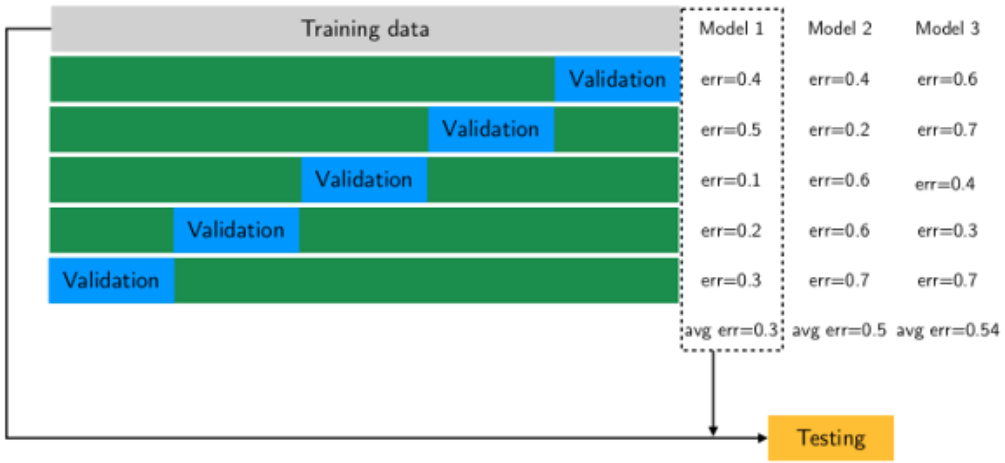


Figure: K-fold cross validation

Forward Chaining/Validation (Cross Validation for Time Series)

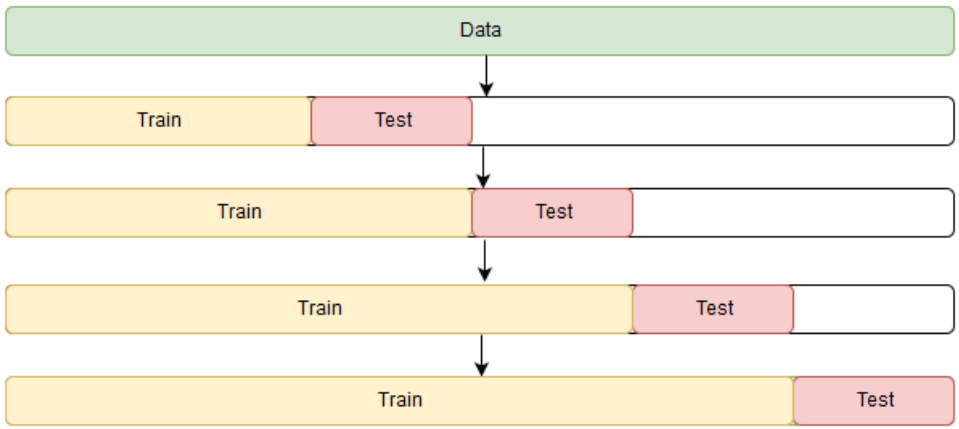


Figure: (Walk) Forward Validation for Time Series

Summary

- loss functions
  - 0/1 loss (for classification)
  - square loss (for regression)
- training set, validation set, test set
  - train/test should resemble train/deploy as closely as possible
  - random split often reasonable
  - for time series, split data in time, rather than randomly
  - validation and test sets are often called "hold-out data"
- k-fold cross validation for small datasets

Other Sources of Test $\neq$ Deployment

Leakage

- Leakage: Information about labels sneaks into features.
- Examples:
  - identifying cat photos by using the title on the page
  - including sales commission as a feature when ranking sales leads
  - using star rating as feature when predicting sentiment of Yelp review

Sample Bias

- Sample bias: Test inputs and deployment inputs have different distributions.
- Examples:
  - create a model to predict US voting patterns, but phone survey only dials landlines
  - building a stock forecasting model, but training using a random selection of companies that exist today – what's the issue?
  - US census slightly undercounts certain subpopulations in a way that's somewhat predictable based on demographic and geographic features
    - If predictable, can it be corrected? Hotly debated topic in 2000 – some of the world's top statisticians couldn't agree (Stephen Fienberg vs David Freedman).

Nonstationarity

- Nonstationarity: when the thing you're modeling changes over time
- Nonstationarity often takes one of two forms:
  - Covariate shift: input distribution changed between training and deployment.
    - (covariate is another term for input feature)
    - e.g. once popular search queries become less popular – new ones appear
    - mathematically similar to sample bias
  - Concept drift: correct output for given input changes over time
    - e.g. season changes, and given person no longer interested in winter coats
    - e.g. last week I was looking for a new car, this week I'm not

Learning Map

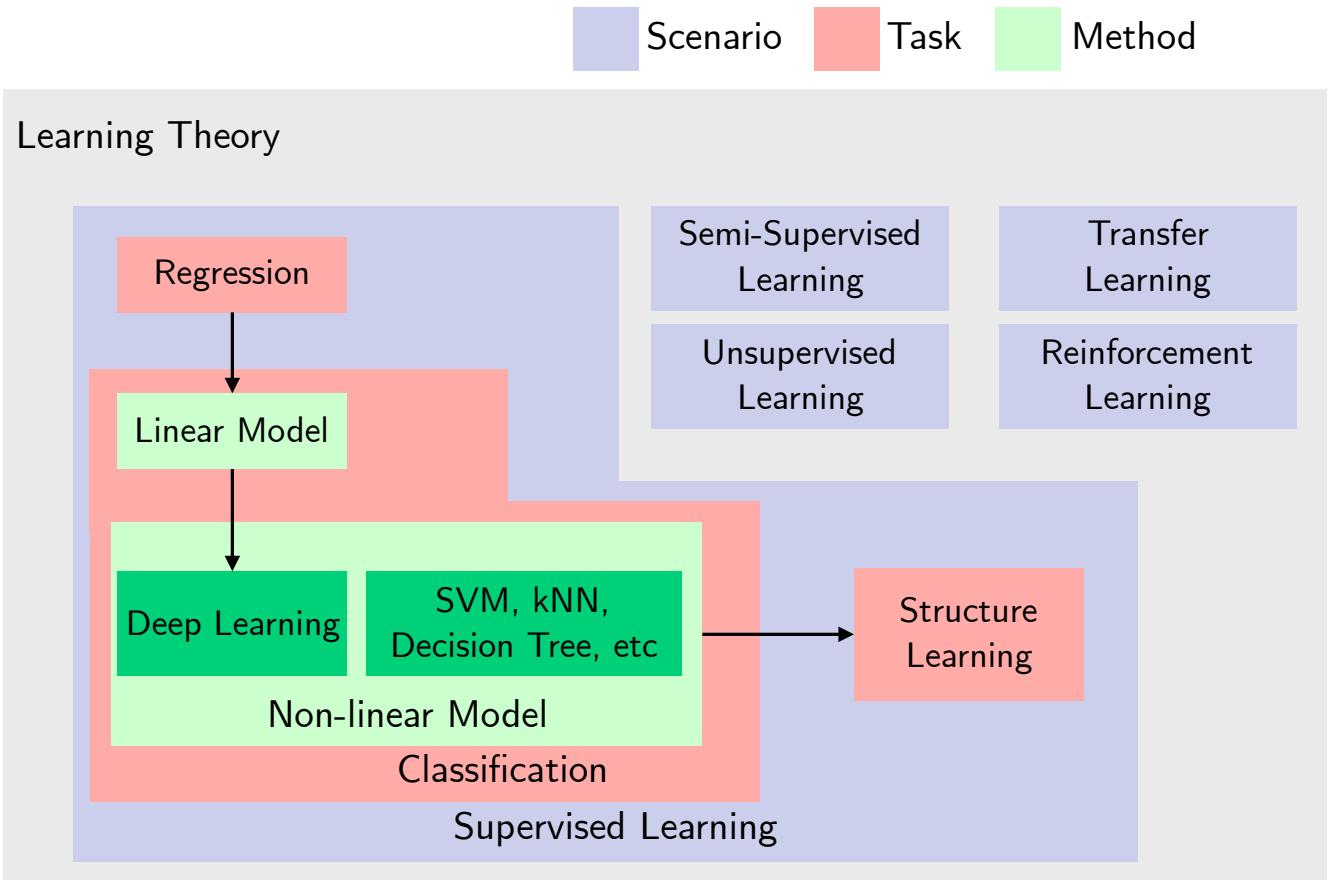


Figure: Learning Map of Machine Learning

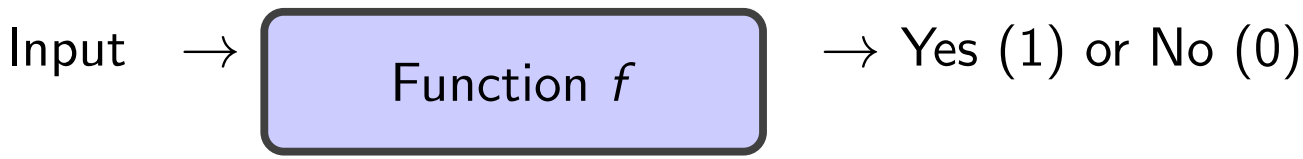
Supervised Learning

- Sample data comprises input vectors along with the corresponding target values (labels).
- Supervised learning uses the given labeled data to find a model (hypothesis) that predicts the target values for previously unseen data.

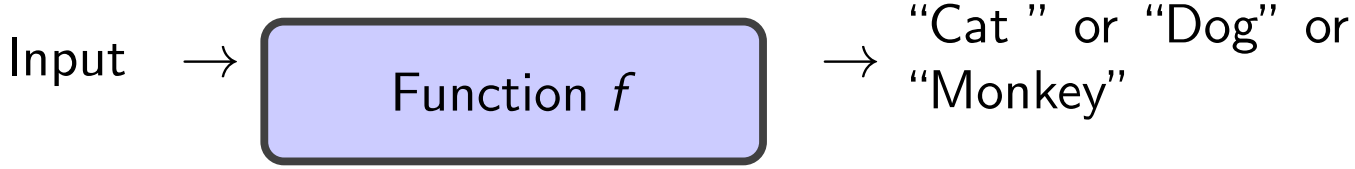
Supervised Learning: Classification

Classification: each element in the sample is labeled as belonging to some class. There is no order among classes.

- Binary classification. (Examples: spam classification)



- Multi-class classification. (Examples: document topic classification, image object classification, etc)



Classification - Deep Learning

- Image Recognition

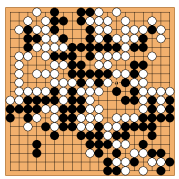
$\rightarrow X \rightarrow$ Function f $\rightarrow$

Hierarchical Structure

$\mathbb{P}(\text{Cat}) = 0.9$
$\mathbb{P}(\text{Dog}) = 0.05$
.....
$\mathbb{P}(\text{Monkey}) = 0.01$

Classification - Deep Learning

- Playing Go

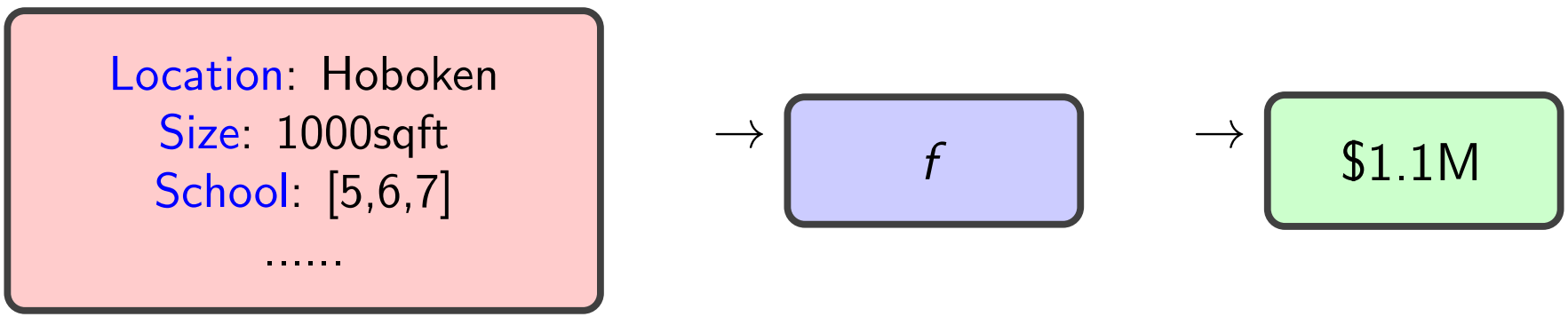


$\rightarrow X \rightarrow$ Function f $\rightarrow$ Next move
Each position is a class
(19*19 classes)

Hierarchical Structure

Supervised Learning: Regression

Regression: each element in the sample is associated with one or more continuous variables. Unlike classes, values have an order among them.
Example: predict house price



Semi-supervised Learning

Unlabeled data may be easily available, while labeled data maybe expensive to obtain.
Semi-supervised learning: it exploits unlabeled examples, in addition to labeled ones, to improve the generalization ability of the resulting classifier.

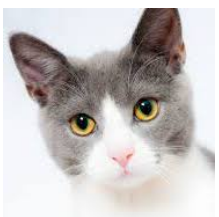
- For example, recognizing cats and dogs

Labelled data

"Cat"

"Dog"

Unlabelled data



Transfer Learning

- For example, recognizing cats and dogs

Figure: Labelled data and Semi-labeled data

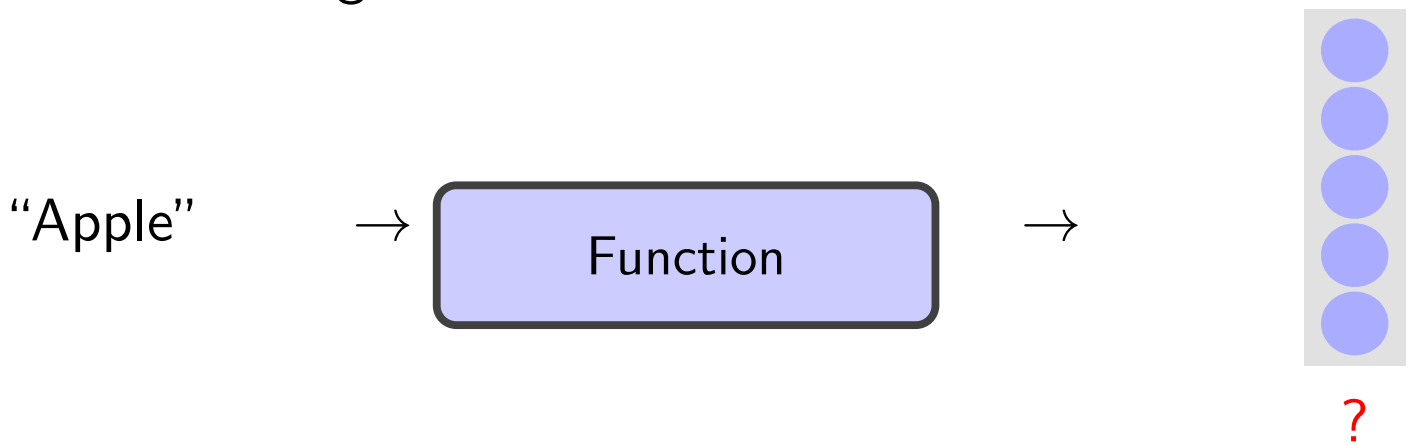
Data not related to the task considered (can be either labeled or unlabeled)

Unsupervised Learning

- The given data consists of input vectors without any corresponding target values.
- The **goal** is to discover groups of similar examples within the data (clustering), or to determine the distribution of data within the input space (density estimation).

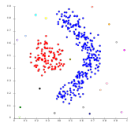
Unsupervised Learning

- Machine reading/understanding: learns the meaning of words from reading a lot of documents

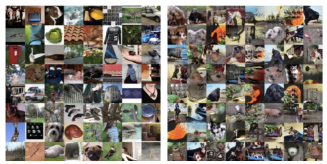


Unsupervised Learning

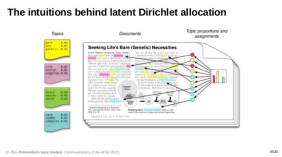
- Clustering



- Machine drawing [Salimans et al 2016]

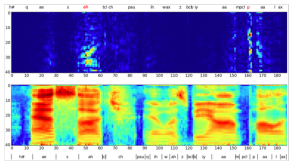


- Machine reading [Blei et al 2003]



Structure Learning - Beyond Classification

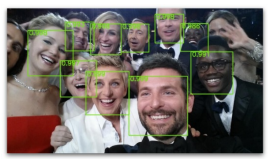
- Speech recognition [Graves et al 2013]



- Machine Translation [Google Brain 2016]

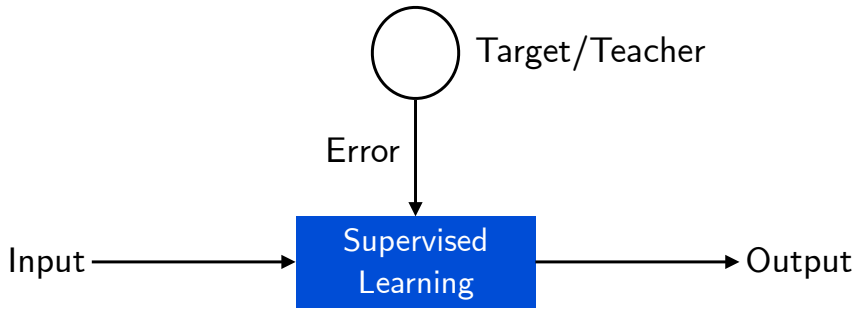
Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

- Face recognition [Farfade et al 2015]

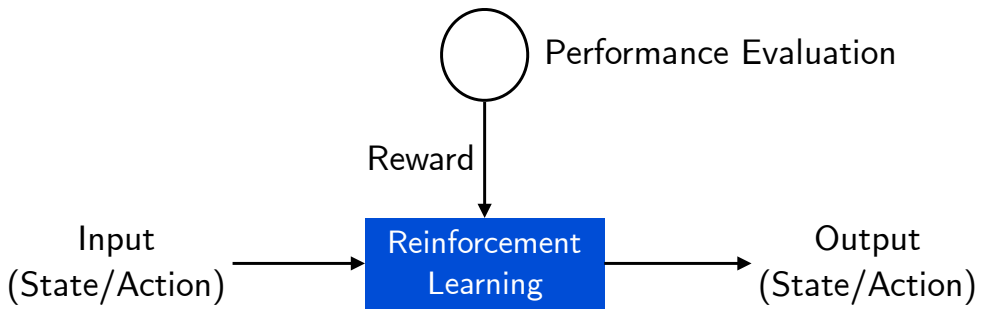


Reinforcement Learning - Beyond Classification

- Supervised: learning from teacher



- Reinforcement: learning from critics



Stevens offers two reinforcement learning courses:

- MA 661 - Dynamic Programming and Reinforcement Learning
- BIA 665 - Applied Reinforcement Learning

Reinforcement Learning - Beyond Classification

- The problem here is to find suitable actions to take in a given situation in order to maximize a reward.
- Trial and error: no examples of optimal outputs are given.
- Trade-off between exploration (try new actions to see how effective they are) and exploitation (use actions that are known to give high reward).

Reinforcement Learning - Example

- Reinforcement

  First move $\rightarrow$ .....many moves.....$\rightarrow$ Win!

Alpha Go is supervised learning + reinforcement learning

Quiz

Readings

- Bishop Chapter 1&2
- For next week: Bishop Chapter 3

Acknowledgements

Slides adapted from Dr. David S. Rosenberg, *Foundations of Machine Learning* at New York University.

Thank you!
Questions?