Machine Learning ZG565

Dr. Sugata Ghosal

Sugata.ghosal@pilani.bits-pilani.ac.in

BITS Pilani
Pilani Campus

BITS Pilani
Pilani Campus

Session 1
Date – 20th May 2023
Time – 2 PM to 4 PM

These slides are prepared by the instructor, with grateful acknowledgement of many others who made their course materials freely available online.

Session Content

- Objective of course
- Evaluation Plan
- What is Machine Learning?
- Application areas of Machine Learning
- Why Machine Learning is important?
- Design a Learning System
- Issues in Machine Learning

Objective of course

- Introduction to the basic concepts and techniques of Machine Learning
- Gain experience of doing independent study and research in the field of Machine Learning
- Develop skills of using recent machine learning software tools to evaluate learning algorithms and model selection for solving practical problems

What We'll Cover in this Course

- **Supervised learning algorithms**
  - Regression
  - Naïve Bayes
  - Logistic regression
  - Decision Tree and Random Forest
  - Support vector machines
- **Ensemble Techniques**
- **Unsupervised learning**
  - Clustering
- **Applications**

innovate achieve lead

Books

Text books and Reference book(s)

T1	Tom M. Mitchell: Machine Learning, The McGraw-Hill Companies

R1	Christopher M. Bishop: Pattern Recognition & Machine Learning, Springer
R2	**P. Tan, et al.** Introduction to Data Mining, **Pearson**
R3	**C.J.C. BURGES: A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer Academic Publishers, Boston.**

Evaluation Plan

Name	Type	Weight
3 Quiz, best 2 scores will be taken	Online	10%
Assignment-I	Take Home	10%
Assignment-II	Take Home	10%
Mid-Semester Test	Closed Book	30%
Comprehensive Exam	Open Book	40%

Please note there will be no change in submission dates for quiz and assignment

Lab Plan

Lab No.	Lab Objective
1	End to End Machine Learning
2	Linear Regression and Gradient Descent Algorithm
3	Logistic Regression Classifier
4	Decision Tree
5	Naïve Bayes Classifier
6	Random Forest

- **Labs not graded**
- **Most of the Lab recordings available at CSIS virtual labs**
- **Webinars will be conducted for lab sessions**
- **Labs will be conducted in Python**

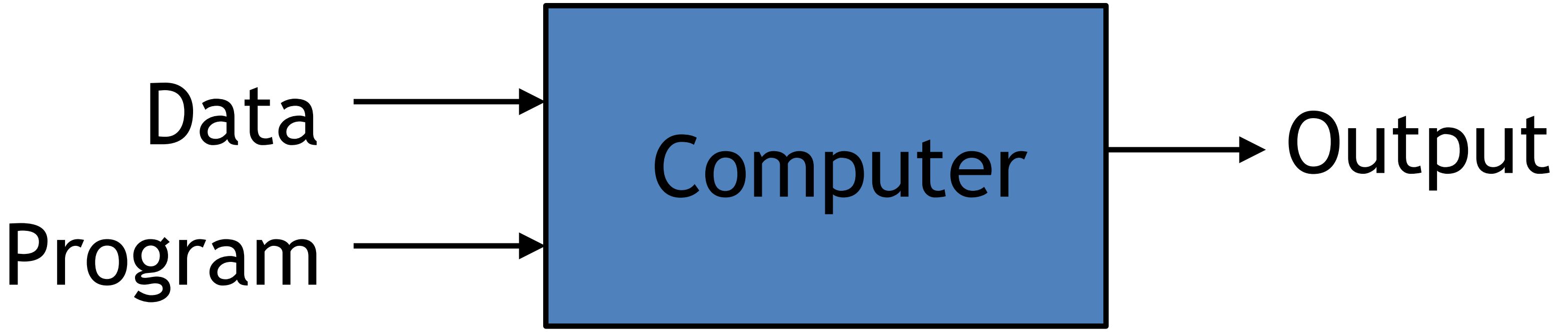
Machine Learning

- **Machine learning** is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions.

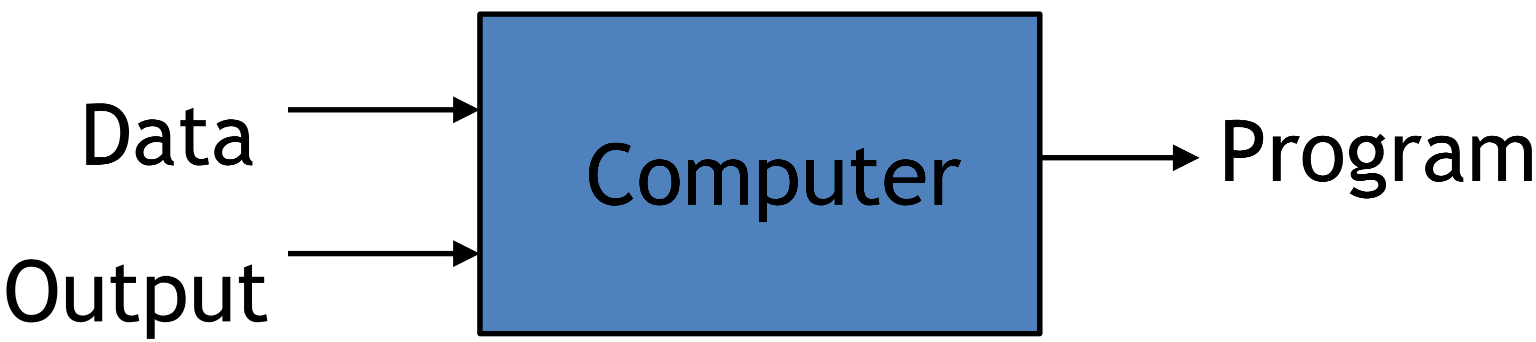
A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity” (Jerry Yang, CEO, Yahoo)

Traditional Programming



Machine Learning



Slide credit: Pedro Domingos

What is Machine Learning?

Definition by Tom Mitchell (1998):

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” Example: playing checkers.

E = the experience of playing many games of checkers

T = the task of playing checkers.

P = the probability that the program will win the next game.

What is Machine Learning?

- To have a learning problem, we must identify
  - The class of tasks
  - The measure of performance to be improved
  - Source of experience

Example of Learning Problems

A Checker Learning Problem

- **Task T:** Playing Checkers
- **Performance Measure P:** Percent of games won against opponents
- **Training Experience E:** To be selected ==> Games Played against itself

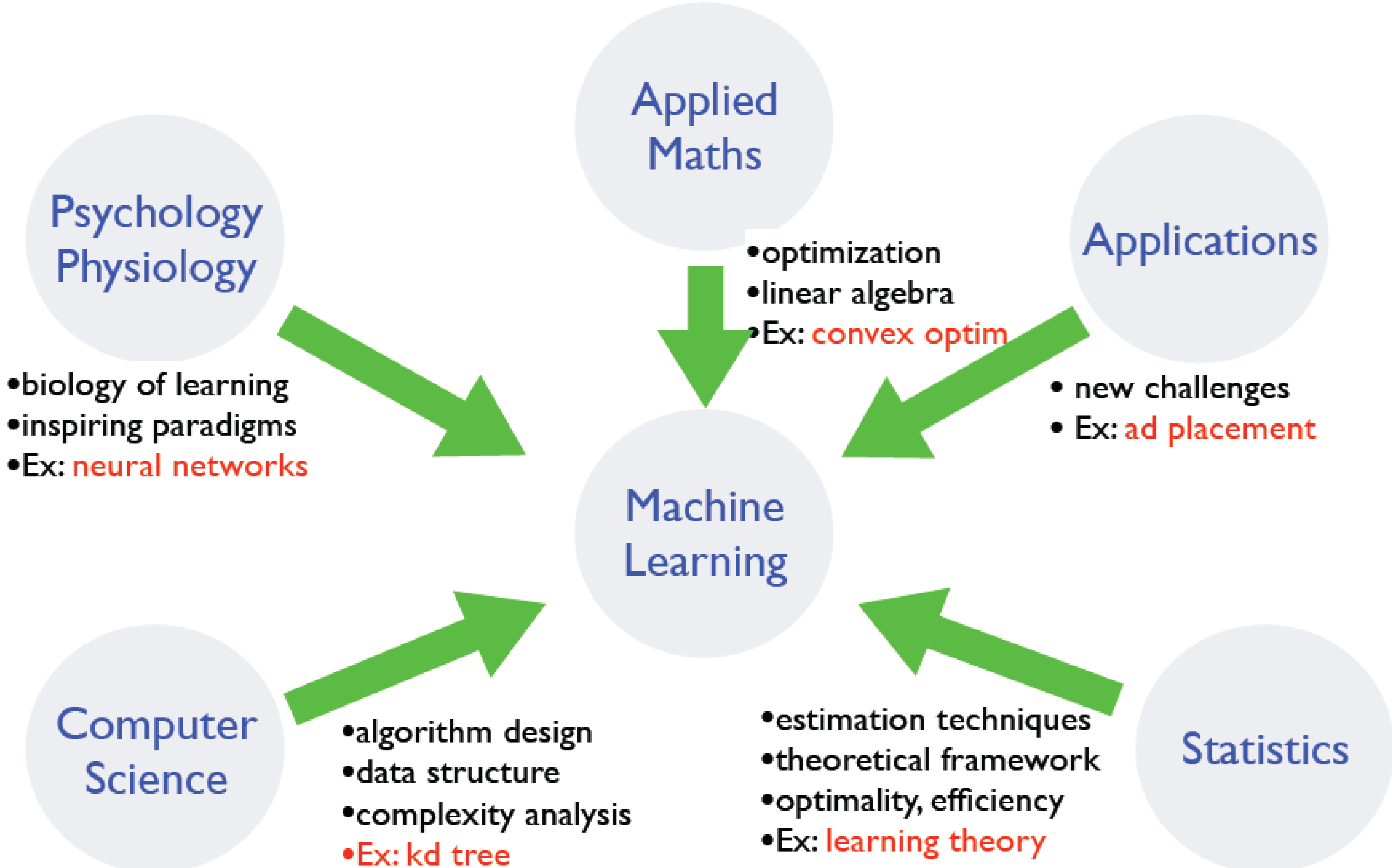
A handwriting recognition learning problem

- **Task T:** recognizing and classifying handwritten words within images
- **Performance measure P**: percent of words correctly classified
- **Training Experience E:** a database of handwritten words with given classifications

A robot driving learning problem

- **Task T:** driving on public four-lane highways using vision sensors
- **Performance measure P:** average distance travelled before an error (as judged by human)
- **Training experience E:** a sequence of images and steering commands recorded while observing a human driver

Where does ML fit in?



Slide credit: Dhruv Batra, Fei Sha

Why is Machine Learning Important?

- Some tasks cannot be defined well, except by examples.
- Relationships and correlations can be hidden within large amounts of data. Machine Learning may be able to find these relationships.
- Human designers often produce machines that do not work as well as desired in the environments in which they are used.

Why is Machine Learning Important ?

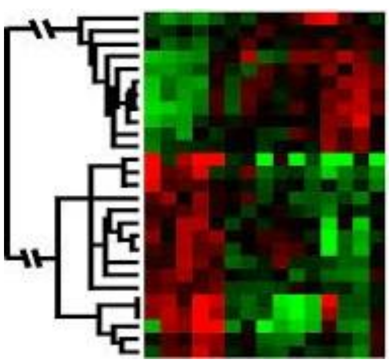
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems “by hand”.

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



Learning isn't always useful:

- There is no need to "learn" to calculate payroll

Slide Credit: Eric Eaton

Application Domains

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Language Processing

Many more emerging…

Slide credit: Pedro Domingos

State of the Art Applications of Machine Learning

•

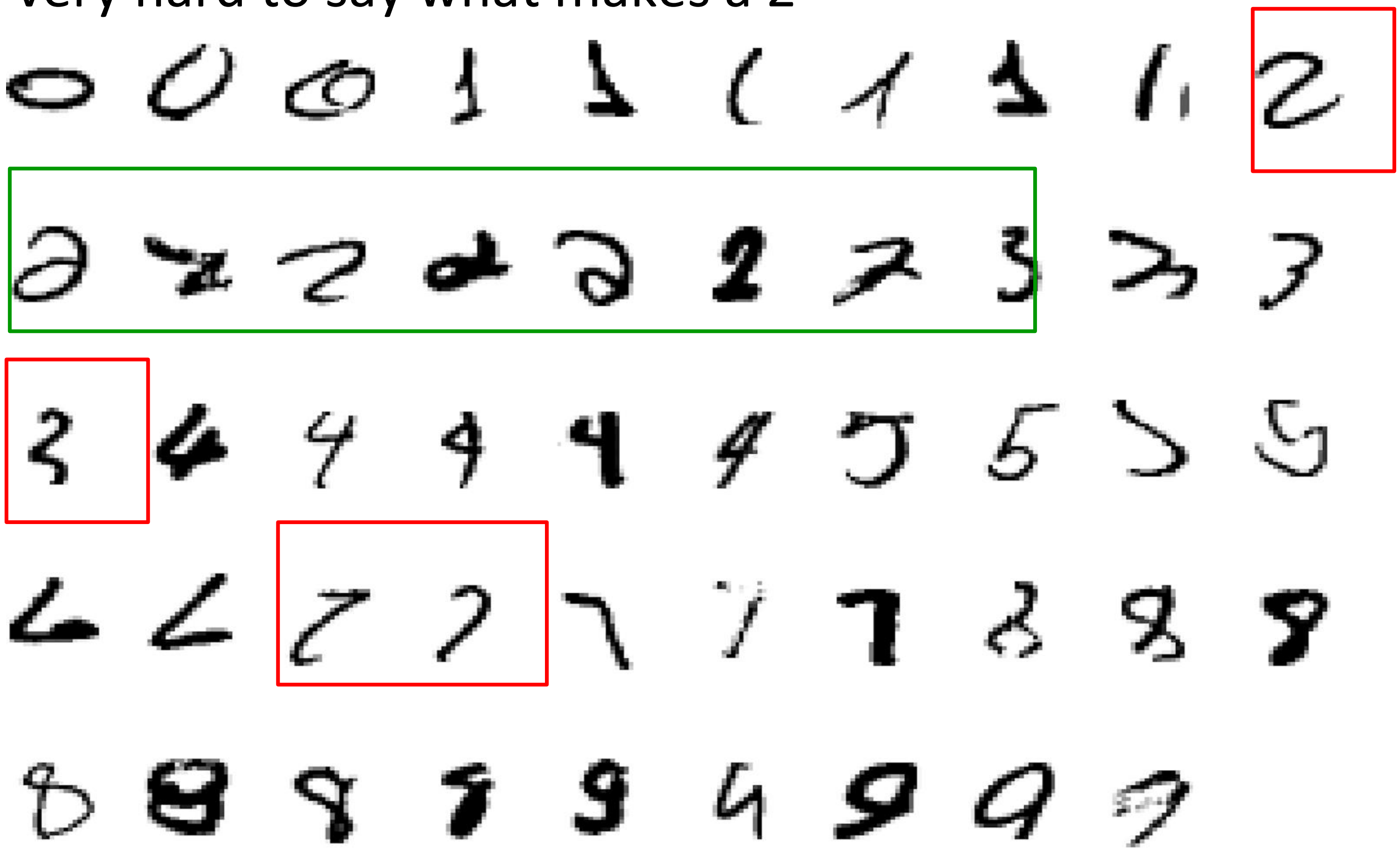
Application Types

- Medical diagnosis
- Credit card applications or transactions
- Fraud detection in e-commerce
- Worm detection in network packets
- Spam filtering in email
- Recommended articles in a newspaper
- Recommended books, movies, music, or jokes
- Financial investments
- DNA sequences
- Spoken words
- Handwritten letters
- Astronomical images

Slide credit: Ray Mooney

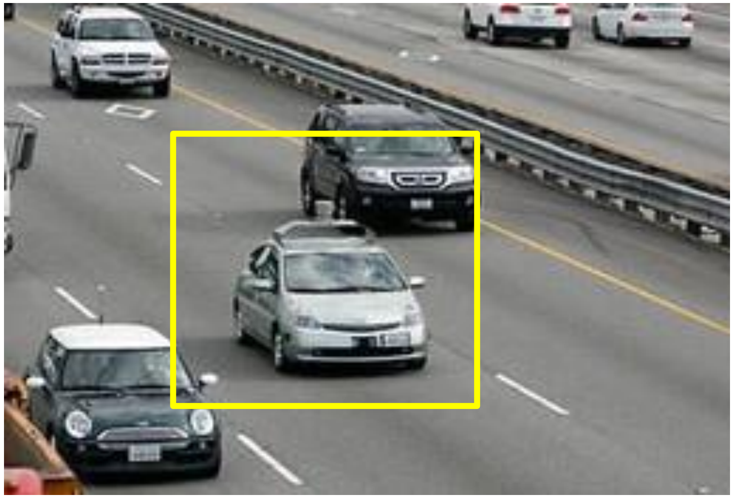
Pattern recognition

It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

Autonomous Cars

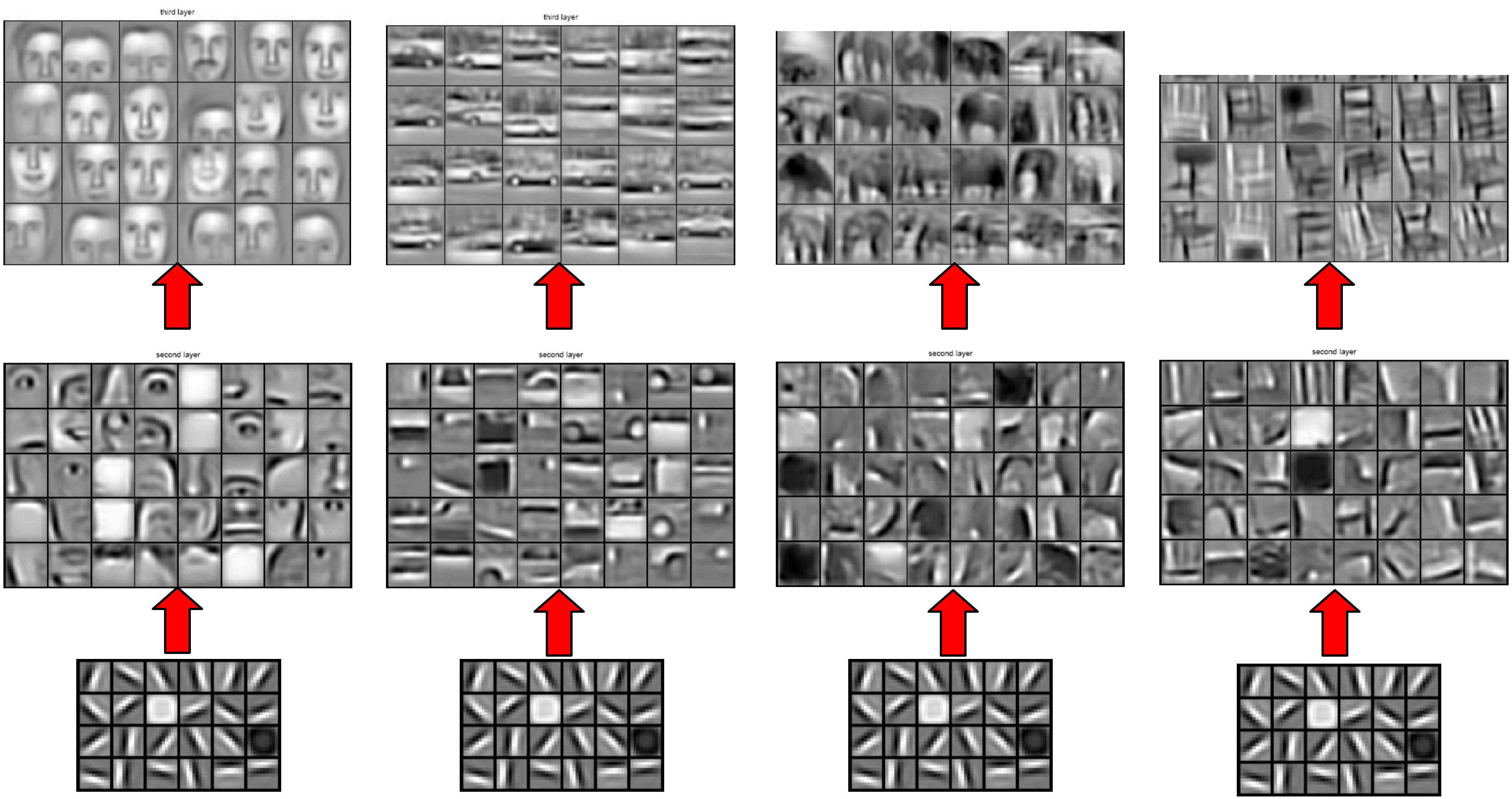


- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

UPenn's Autonomous Car →

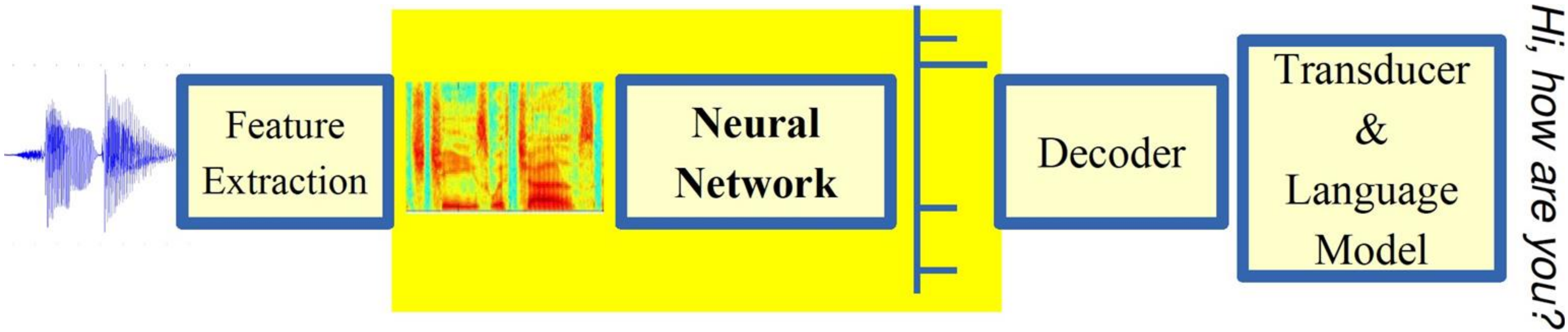
Slide credit: Eric Eaton

Learning of Object Parts

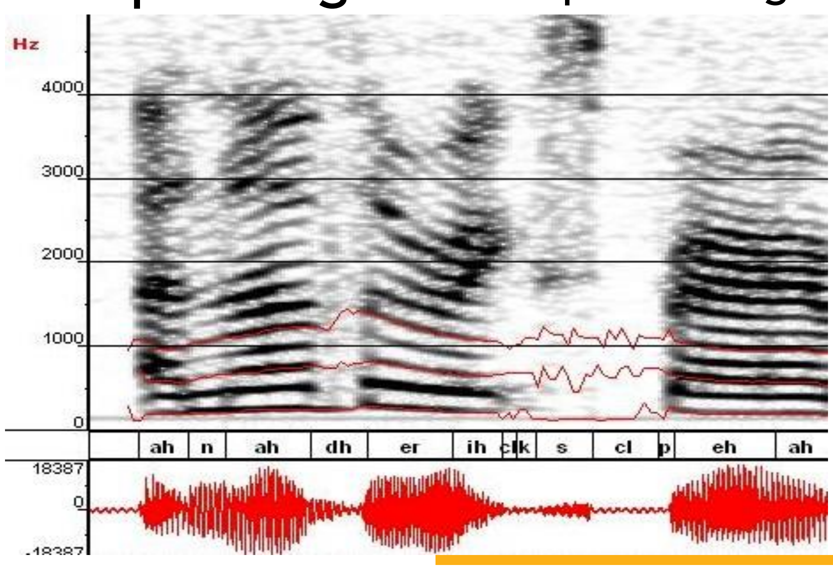


Slide credit: Andrew Ng

Automatic Speech Recognition



ML used to predict phoneme states from sound spectrogram



Deep Learning Based Results

# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline Gaussian Mixture Model based word error rate = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

Slide credit: Eric Eaton

Robotics

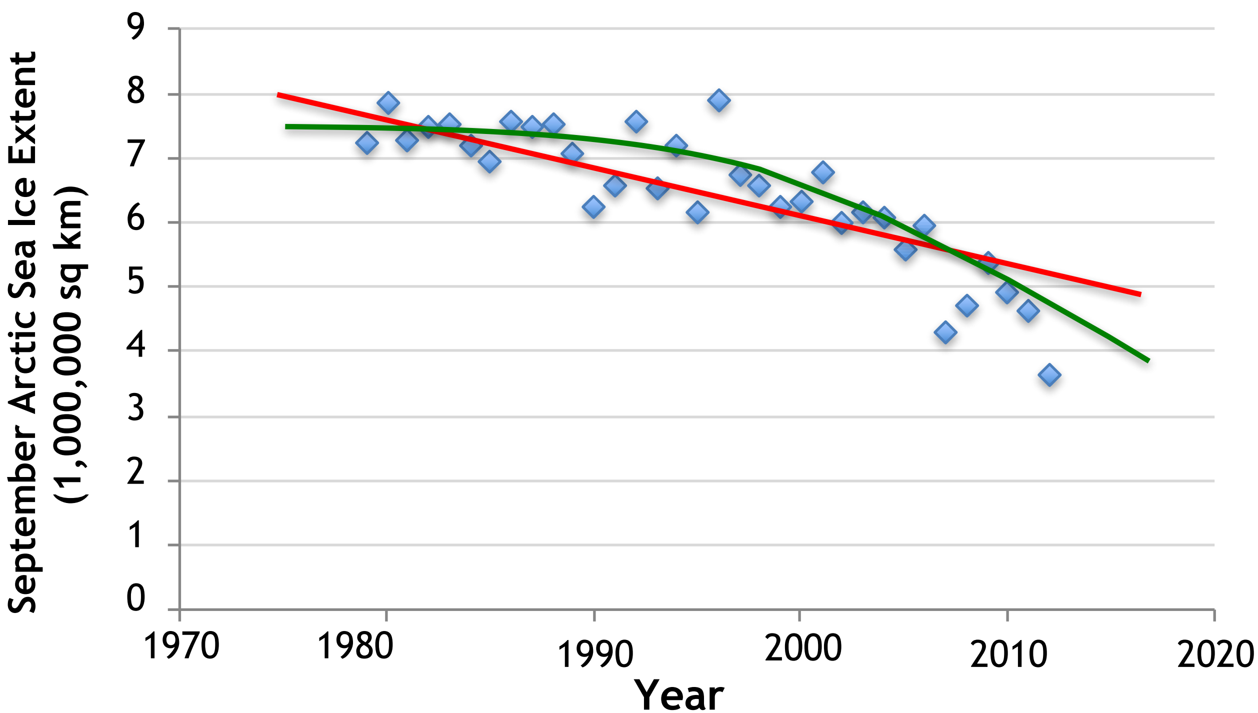
Types of Learning

- **Supervised (inductive) learning**
  - Given: training data, desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Given: rewards from sequence of actions

Slide Credit: Eric Eaton

Supervised Learning: Regression

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013) Slide Credit: Eric Eaton

Regression

$$y = f(\mathbf{x})$$

output

prediction function

features

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Classification Example

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

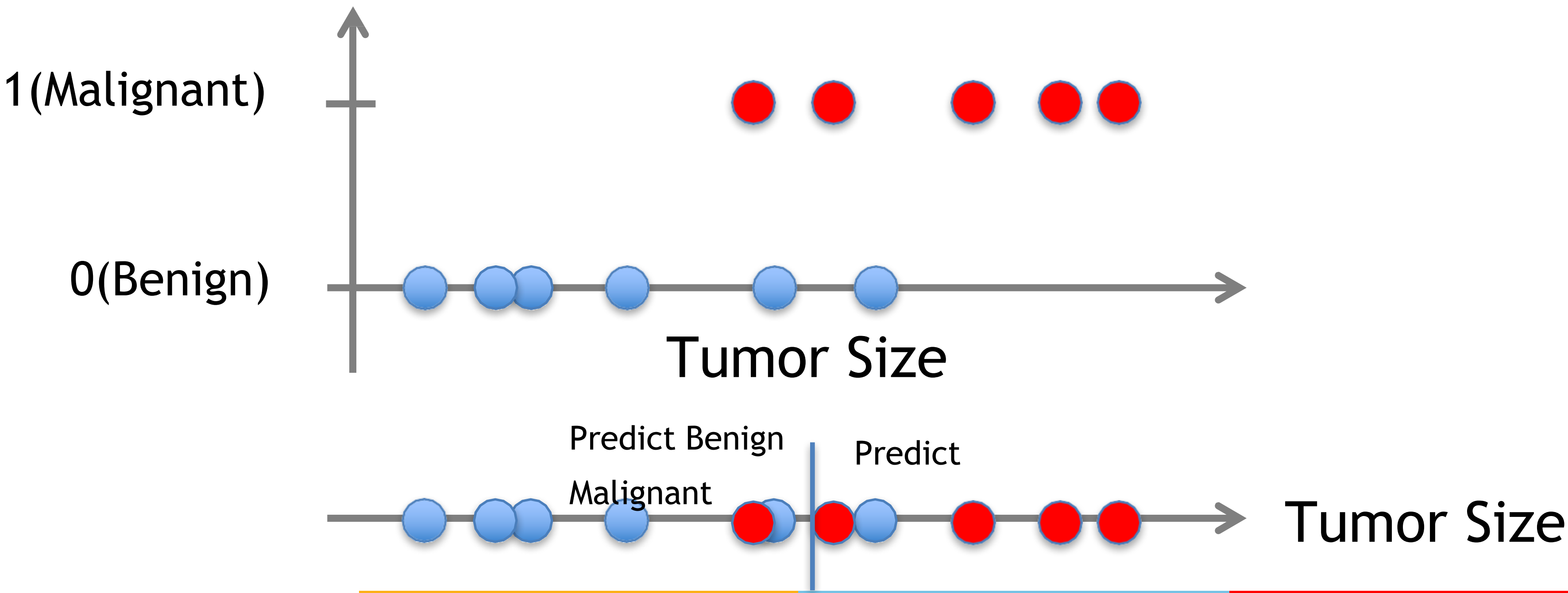
f() = “cow”

Slide credit: L. Lazebnik

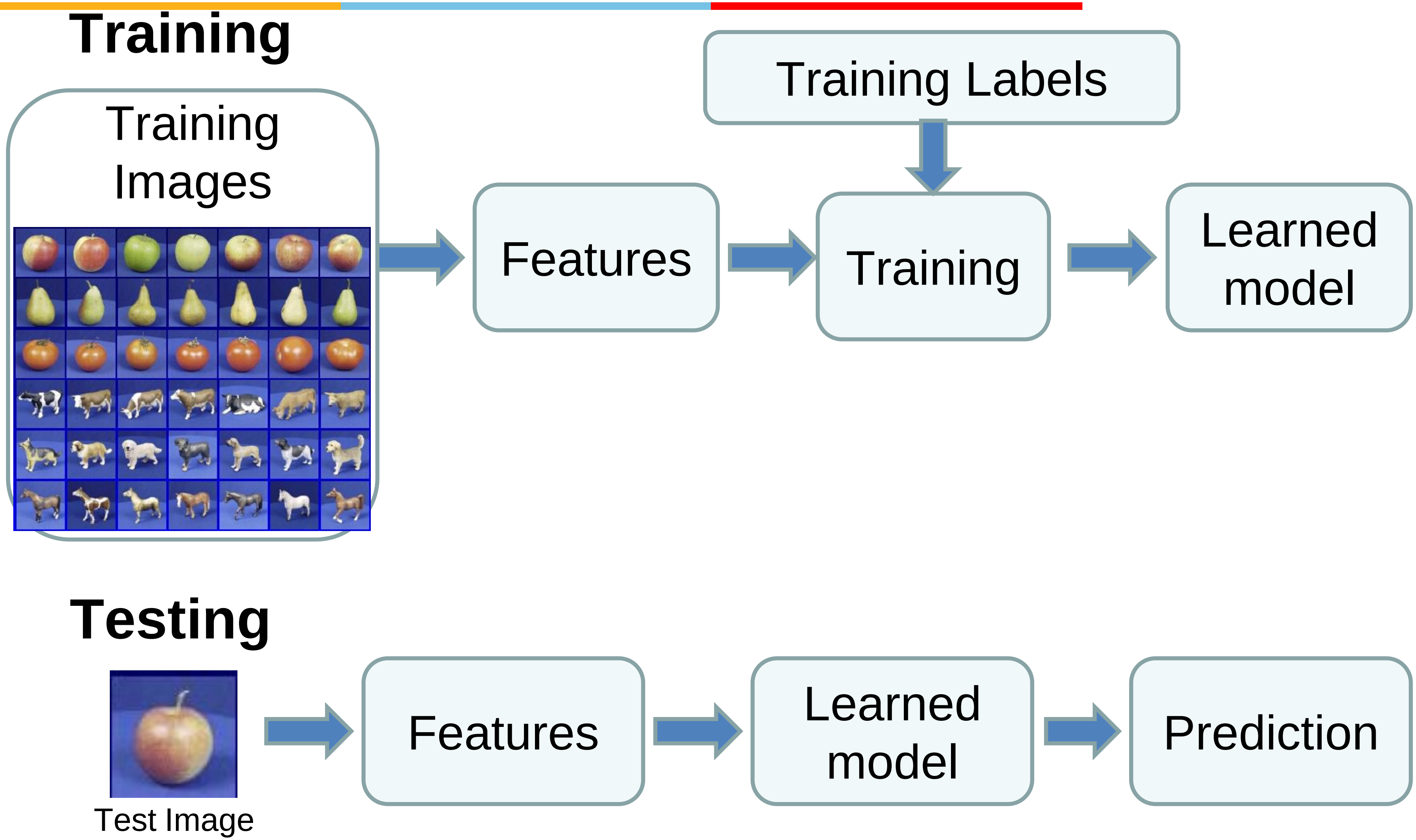
Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification

Breast Cancer (Malignant / Benign)



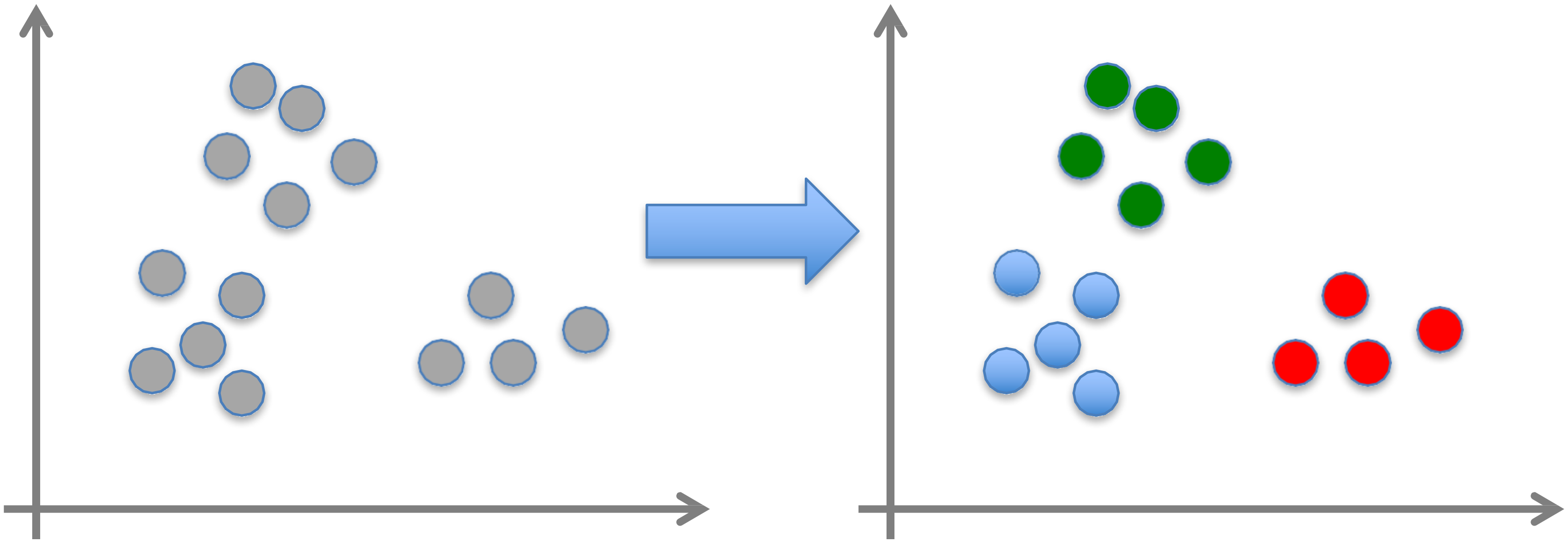
Classification



Slide credit: D. Hoiem and L. Lazebnik

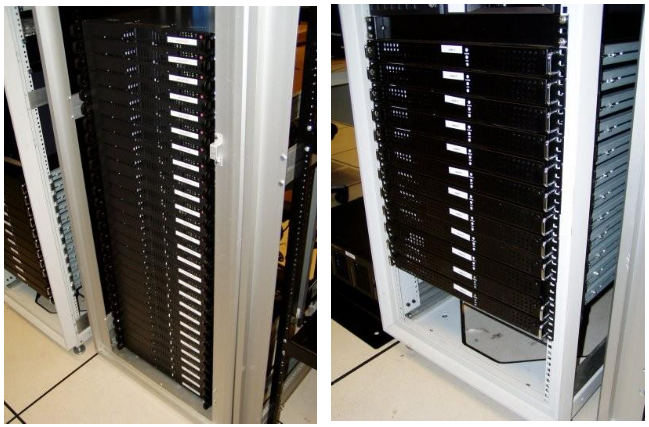
Unsupervised Learning

- Given x_1, x_2, …, x_n (without labels)
- Output hidden structure behind the x's
  - E.g., clustering

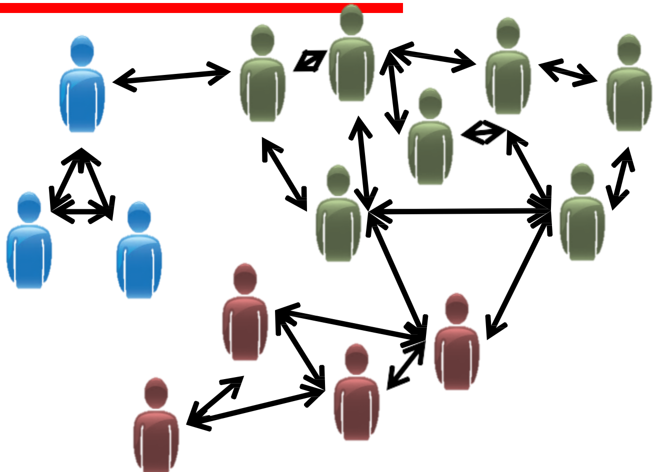


Slide Credit: Eric Eaton

Unsupervised Learning



Organize computing clusters



Social network analysis

Market segmentation

Astronomical data analysis

Slide credit: Andrew Ng

Reinforcement Learning

- No predefined data
- Semi-supervised learning model in machine learning,
- Allow an agent to take actions and interact with an environment so as to maximize the total rewards.
- Examples:
  – Resources management in computer clusters
  – Game playing
  – Robot in a maze

Slide Credit: Eric Eaton

Supervised, Unsupervised and Reinforcement Learning Comparison

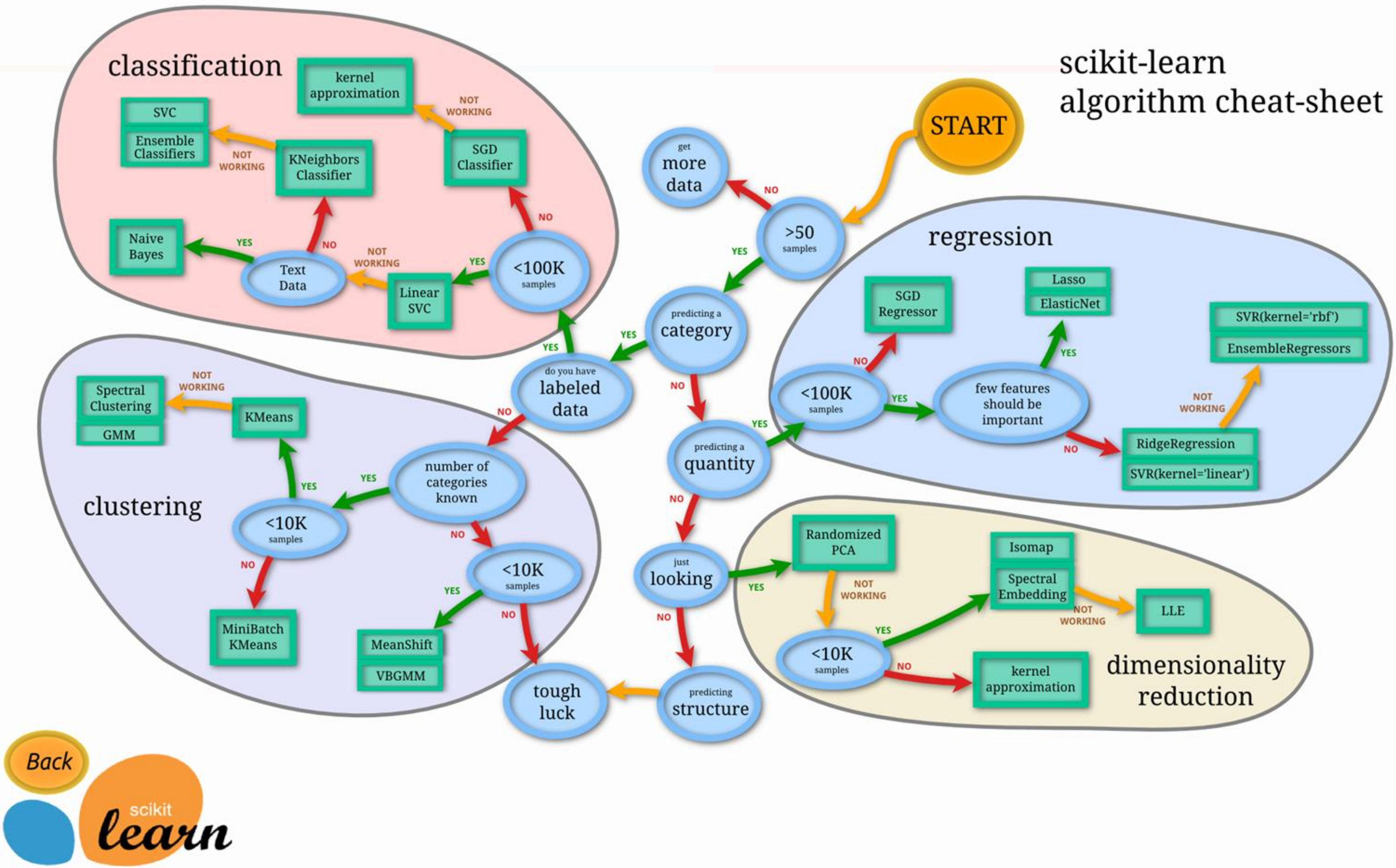
Criteria	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Definition	The machine learns by using labeled data	The machine is trained on unlabeled data without any guidance	An agent interacts with its environment by performing actions & learning from errors or rewards
Type of problems	Regression & classification	Association & clustering	Reward-based
Type of data	Labeled data	Unlabeled data	No predefined data
Training	External supervision	No supervision	No supervision
Approach	Maps the labeled inputs to the known outputs	Understands patterns & discovers the output	Follows the trial-and-error method

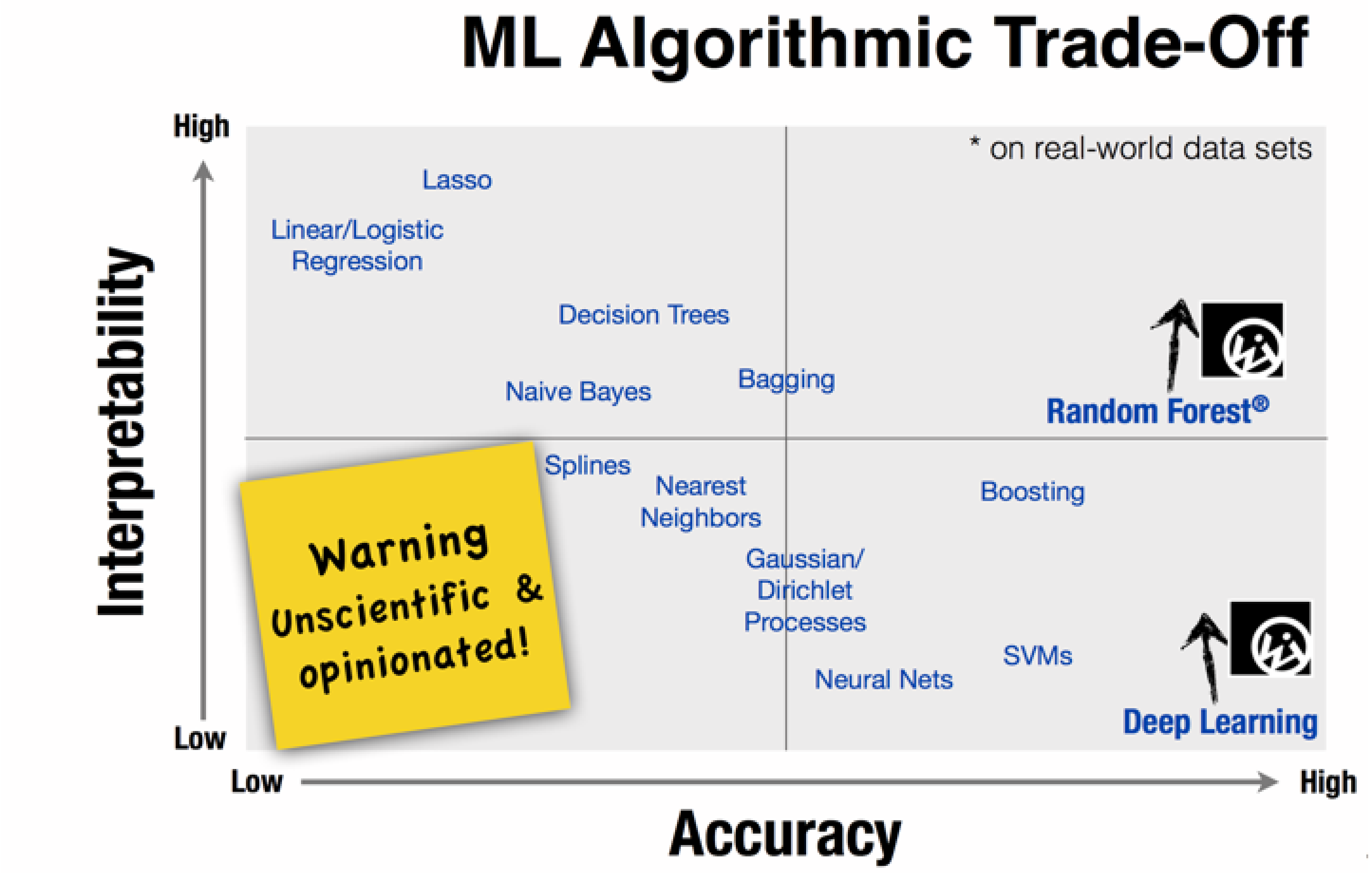
Open source ML programming tools

	Platform		Algorithms or Features
Scikit Learn	Linux, Mac OS, Windows	Python, C, C++	Classification, Regression, Clustering Preprocessing, Model Selection Dimensionality reduction.
PyTorch	Linux, Mac OS, Windows	Python, C++	Autograd Module, Optimization Module NN Module
TensorFlow	Linux, Mac OS, Windows	Python, C++	Provides a library for dataflow programming.
Weka	Linux, Mac OS, Windows	Java	Data preparation, Classification Regression, Clustering, Visualization Association rules mining

Open source ML programming tools

Colab	Cloud Service	-	Supports libraries of PyTorch, Keras, TensorFlow, and OpenCV
Apache Mahout	Cross-platform	Java Scala	Preprocessors,Regression Clustering, Recommenders Distributed Linear Algebra.
Accors.Net	Cross-platform	C#	Classification, Regression, Distribution Clustering, Hypothesis Tests & Kernel Methods, Image, Audio & Signal & Vision
Shogun	Windows Linux, UNIX Mac OS	C++	Regression, Classification, Clustering Support vector machines. Dimensionality reduction, Online learning etc.
Keras.io	Cross-platform	Python	API for neural networks


scikit-learn
algorithm cheat-sheet
START
>50 samples
NO
get more data
YES
predicting a category
YES
do you have labeled data
YES
<100K samples
YES
Linear SVC
NOT WORKING
Text Data
YES
Naive Bayes
NO
KNeighbors Classifier
NOT WORKING
SVC
Ensemble Classifiers
NO
SGD Classifier
NOT WORKING
kernel approximation
classification
NO
number of categories known
YES
<10K samples
YES
KMeans
NOT WORKING
Spectral Clustering
GMM
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift
VBGMM
NO
tough luck
clustering
NO
predicting a quantity
YES
<100K samples
NO
SGD Regressor
YES
few features should be important
YES
Lasso
ElasticNet
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
regression
NO
just looking
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction
NO
predicting structure
Back
scikit learn


ML Algorithmic Trade-Off
* on real-world data sets
Interpretability
High
Low
Accuracy
Low
High
Lasso
Linear/Logistic Regression
Decision Trees
Naive Bayes
Bagging
Random Forest®
Splines
Nearest Neighbors
Boosting
Gaussian/ Dirichlet Processes
SVMs
Neural Nets
Deep Learning
Warning
Unscientific &
opinionated!

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Every ML algorithm has three components
  - **Representation**
  - **Optimization**
  - **Evaluation**

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- etc.

Evaluating Performance

- If *y* is discrete:
  – Accuracy: # correctly classified / # all test examples
  – Precision/recall
    - True Positive, False Positive, True Negative, False Negative
    - **P**recision = TP / (TP + FP) = # predicted true pos / # predicted pos
    - **R**ecall = TP / (TP + FN) = # predicted true pos / # true pos
  – F-measure
    = 2PR / (P + R)

- Want evaluation metric to be in some range, e.g. [0 1]
  – 0 = worst possible classifier, 1 = best possible classifier

Evaluating Performance

- If *y* is continuous:
  – Sum-of-Squared-Differences (SSD) error between predicted and true *y*:

$$\mathbf{E} = \sum_{\mathbf{i}=1}^{\mathbf{n}} (\mathbf{f}(\mathbf{x}_{\mathbf{i}}) - \mathbf{y}_{\mathbf{i}})^2$$

Issues in Machine Learning

- What algorithms are available for learning a concept? How well do they perform?
- How much training data is sufficient to learn a concept with high confidence?
- When is it useful to use prior knowledge?
- Are some training examples more useful than others?
- What are the best tasks for a system to learn?
- What is the best way for a system to represent its knowledge?

Training vs Testing

- What do we want?
  - High accuracy on training data?
  - No, high accuracy on *unseen/new/test data!*
  - Why is this tricky?
- Training data
  - Features (x) and labels (y) used to learn mapping f
- Test data
  - Features used to make a prediction
  - Labels only used to see how well we've learned f!!!
- Validation data
  - Held-out set of the *training data*
  - Can use both features and labels to tune *parameters* of the model we're learning

Training vs. Test Distribution

- We generally assume that training and test examples are independently drawn from the same overall distribution of data
  – We call this “i.i.d” which stands for “independent and identically distributed”

Slide credit: Ray Mooney

ML in Practice

Loop

- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Recommended Readings

- Tom M. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc. International Edition 1997. **[Ch. 1]**
- **http://www.cs.princeton.edu/courses/archive/spr08/cos511/ [Web]**
- https://www.softwaretestinghelp.com/machine-learning-tools/

END
of
Session 1