Dependable and Secure AI

Introduction

Can AI-based Components be Part of Dependable Systems?

Dependable Systems

Dependable Systems can be found in many forms and application domains, but especially in transportation systems, medical systems and recently in the domain of IoT and Industry 4.0.*

*Industry 4.0 has been defined as “a **name for the current trend of automation and data exchange in manufacturing technologies**, including cyber-physical systems, the Internet of things, cloud computing and cognitive computing and creating the smart factory”

“Dependability”

- **Reliability** how often is the system allowed to fail
- **Availability** to which extend is the system usable, when it is needed
- **Maintainability** how intense is the maintenance of the system
- **Safety** how must the environment be secured against the system
- **Security** how must the system be protected against the environment

Safety Integrity Level : Standard

- Safety Integrity Levels (SIL) define the criticality of the component,
- Each SIL requires different development techniques as well as testing or verification methods and techniques.

- The SILs are defined by the probability of failure, a risk reduction factor (can the risk of failure be reduced by a certain amount, using multiple instances, redundancy, etc), probability of failure per hour and the meantime between failure.

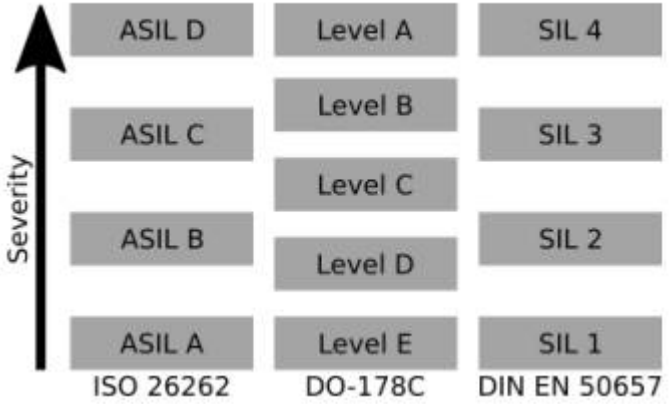
SIL Safety Integrity Level	PFDavg Average probability of failure on demand per year (low demand mode)	RRF Risk Reduction Factor	PFDavg Average probability of failure on demand per hour (high demand or continuous mode)
SIL 4	$\geq 10^{-5}$ and $< 10^{-4}$	100000 to 10000	$\geq 10^{-9}$ and $< 10^{-8}$
SIL 3	$\geq 10^{-4}$ and $< 10^{-3}$	10000 to 1000	$\geq 10^{-8}$ and $< 10^{-7}$
SIL 2	$\geq 10^{-3}$ and $< 10^{-2}$	1000 to 100	$\geq 10^{-7}$ and $< 10^{-6}$
SIL 1	$\geq 10^{-2}$ and $< 10^{-1}$	100 to 10	$\geq 10^{-6}$ and $< 10^{-5}$

IEC 61508
Automatic protection systems called safety-related systems

Standards: **Safety integrity level (SIL)**

SIL1 being the lowest and SIL4 the highest severity level:

- For example in SIL4 :
  - high coverage of branches in the source code of a component
  - ensures adequate testing of the most critical components in the system.
  - standard procedure in avionic or automotive applications.



- Civil avionic systems : regulated by DO178c
- train applications DIN EN 50657
- Medical devices are certified under IEC 82304, 2018 [4]
- Automotive under ISO 26262 [5].
- Each of these standards defines strict requirements with the goal to ensure the the functional safety of each component

Ensuring dependability in critical systems

- **Analytical Approaches:** strict and rigorous review of specification, design and implementation.
- **Constructive Approaches:** These techniques and patterns can be used as a guideline to ensure safety during the design and implementation phase of a project, for example safety cases. These scenarios can be used to directly derive the design or even parts of the implementation of the system.
- Fault tolerant system with **redundancy** concepts can be implemented to increase the reliability and availability of the system.
- In addition a **fault containment** strategy can be developed. If a fault occurs the consequences spread only to specific predefined boundaries, as a result, the system can stay intact.

Risk mitigation mechanisms

From a more technical point of view dependability properties of a system can be improved by adding risk mitigation mechanisms:

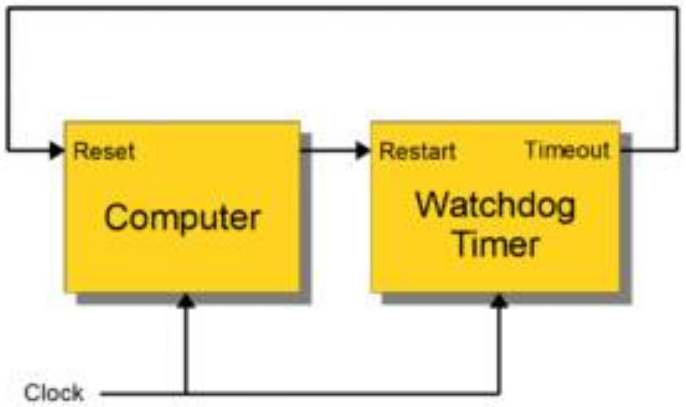
- watchdog or
- brownout detection.

Watchdog

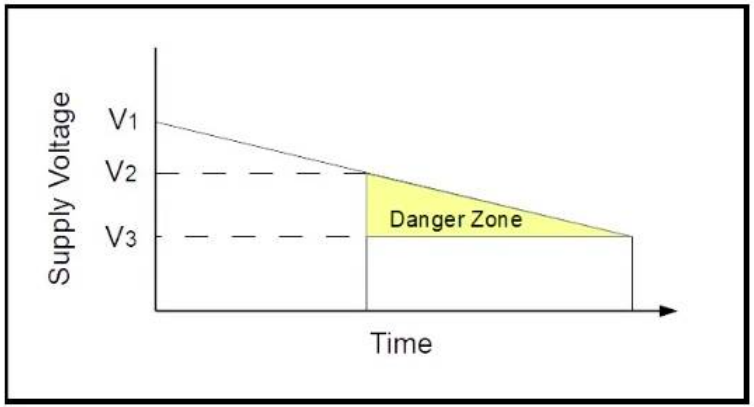
- The hardware component of a watchdog is a counter:
  - set to a certain value
    then counts down towards zero.

- It is the responsibility of the software:
  to set the count to its original value so that it never reaches zero.

- If timer reaches zero:
  - it is assumed that the software has failed in some manner
  - CPU is reset.

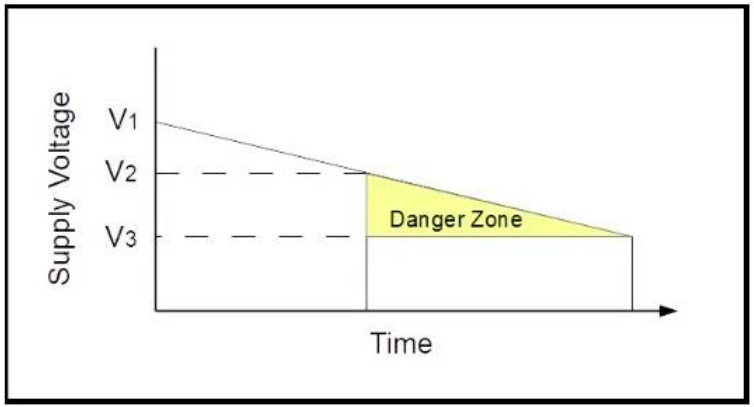


Brownout Detection



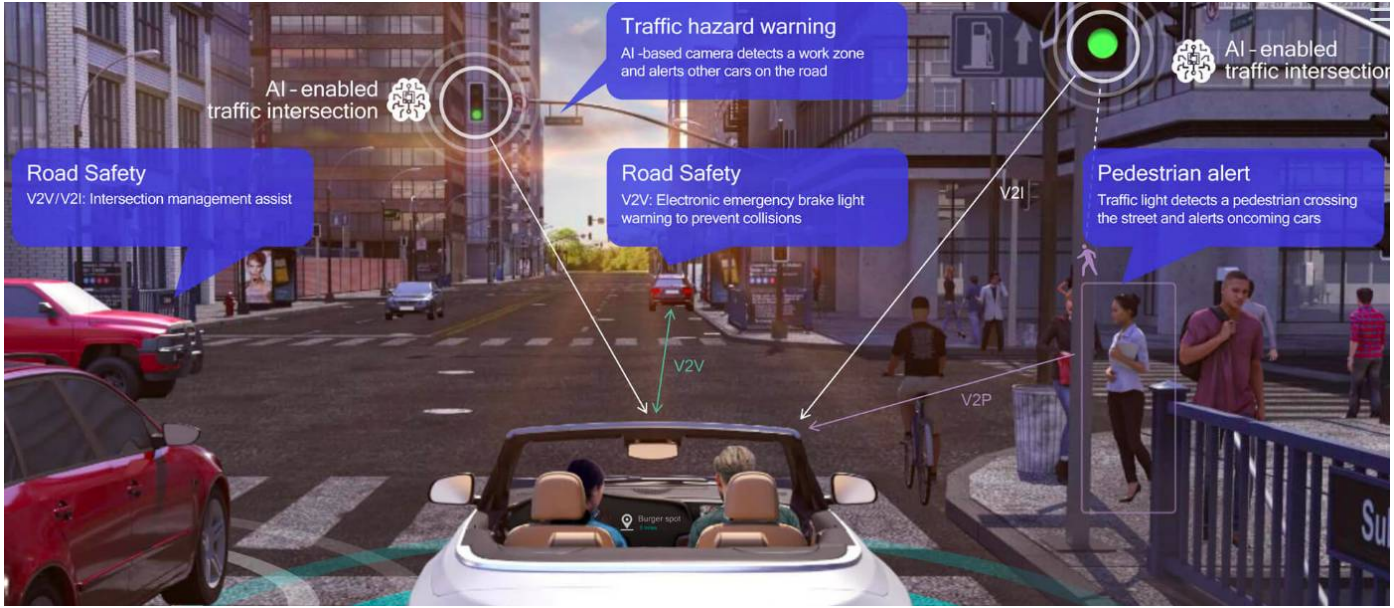
- A “brown out” of a microcontroller is a partial and temporary reduction in the power supply voltage below the level required for reliable operation.
- Many microcontrollers have a protection circuit which detects when the supply voltage goes below this level
- puts the device into a reset state to ensure proper startup when power returns. This action is called a “Brown Out Reset” or BOR.

Brownout Detection



- V1: is the normal power supply voltage. V2: is the point where the microcontroller may not operate reliably.
- V3: a point where operation stops entirely.
- Between V2 and V3 is a “danger zone” where things can go wrong and operation is unreliable.
- The device could work correctly for years while the power supply goes in and out of the danger zone and then there is a failure.
- The BOR level is set above V2 and replaces the danger zone with a reset of the device.
- Reset is not good but (usually) better than uncertain.

Artificial Intelligence in Critical Systems



- Autonomous driving : prominent example for systems incorporating critical components derived using ML.
- The capability of an **AI system to react in complex scenarios in a short amount of time** is unique
- Enables the system to identify pedestrians or traffic signs in a fraction of classic image recognition methods used before.

Problems with AI-ML

- ML methods are b**ased on probabilities**.
- They are **stochastic principles**, which can only estimate the correct answer with a specific certainty.
- Even though the ML algorithms might be 100% sure that the outcome is correct, the answer can still be wrong. This can e.g. happen if the **quality of the training** data is too low, or the data does not even contain all possible scenarios

Introduction to Machine Learning

Slide Ref: Introduction to Machine Learning, Eric Eaton

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

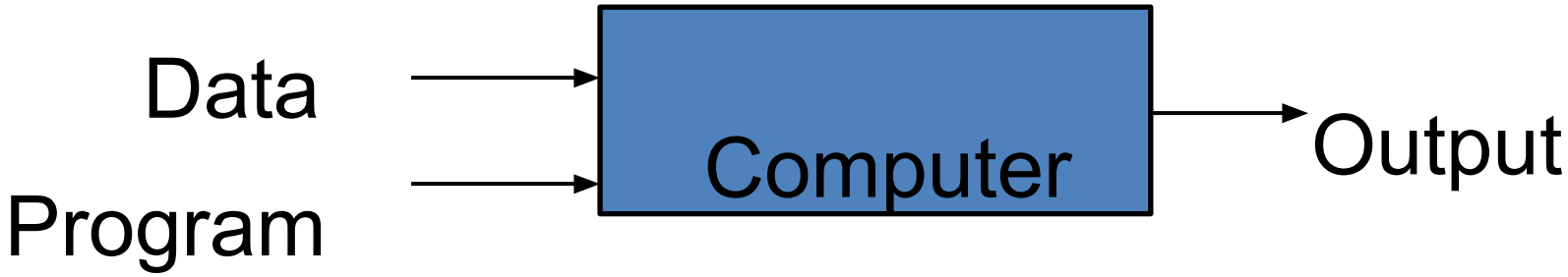
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

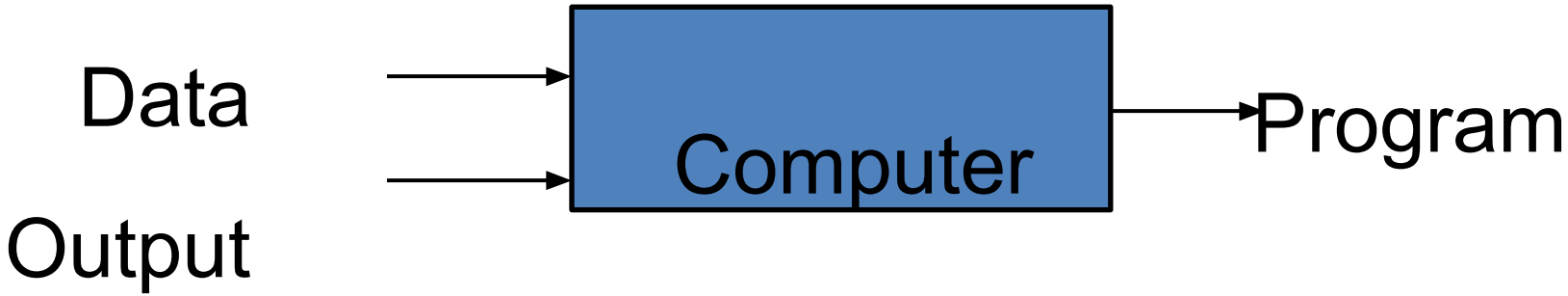
- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

Traditional Programming



Machine Learning



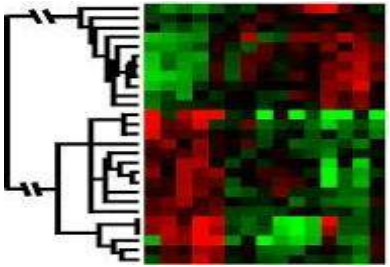
Slide credit: Pedro

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)



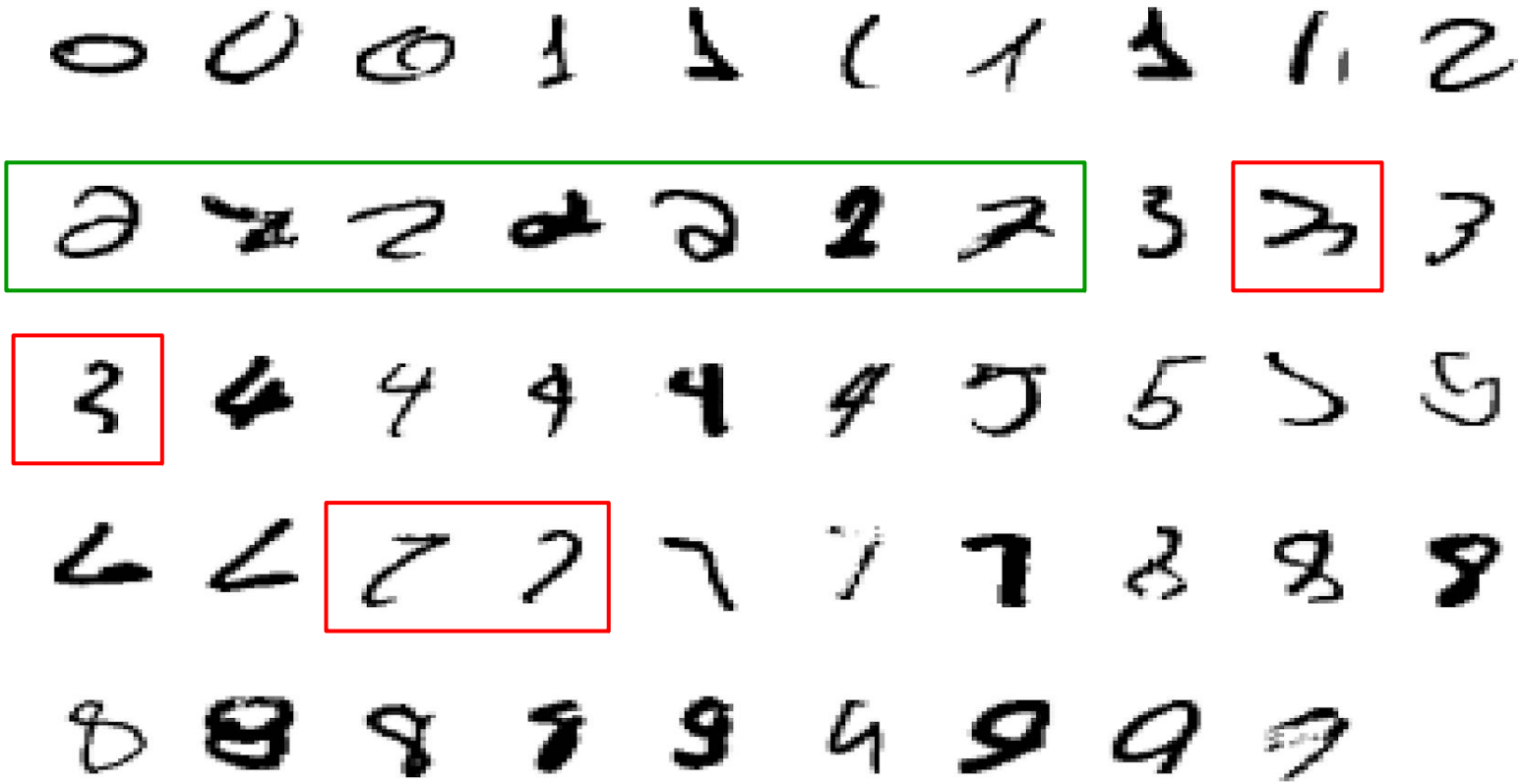



Learning isn't always useful:

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

Based on slide by E. Alpaydin

A classic example : It is very hard to say what makes 2



Slide credit: Geoffrey

Some more examples of Task

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Slide credit: Geoffrey

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Slide credit: Pedro

Samuel's Checkers-Player

"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

Defining the Learning Task

Improve on task T, with respect to performance metric P, based on experience E

T: Playing checkers

P: Percentage of games won against an arbitrary opponent

E: Playing practice games against itself

T: Recognizing hand-written words

P: Percentage of words correctly classified

E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors

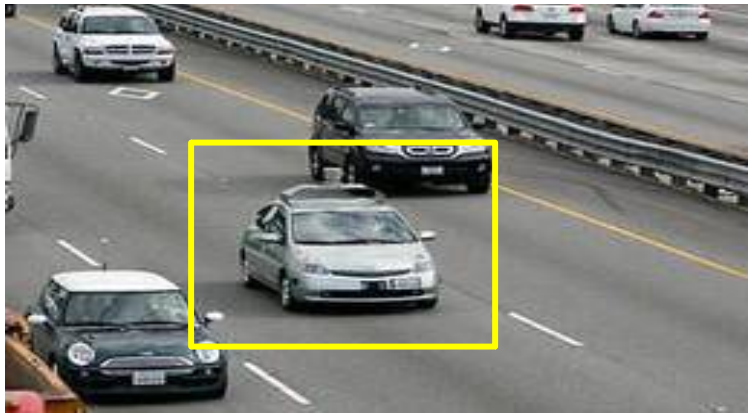
P: Average distance traveled before a human-judged error

E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate. P: Percentage of email messages correctly classified. E: Database of emails, some with human-given labels

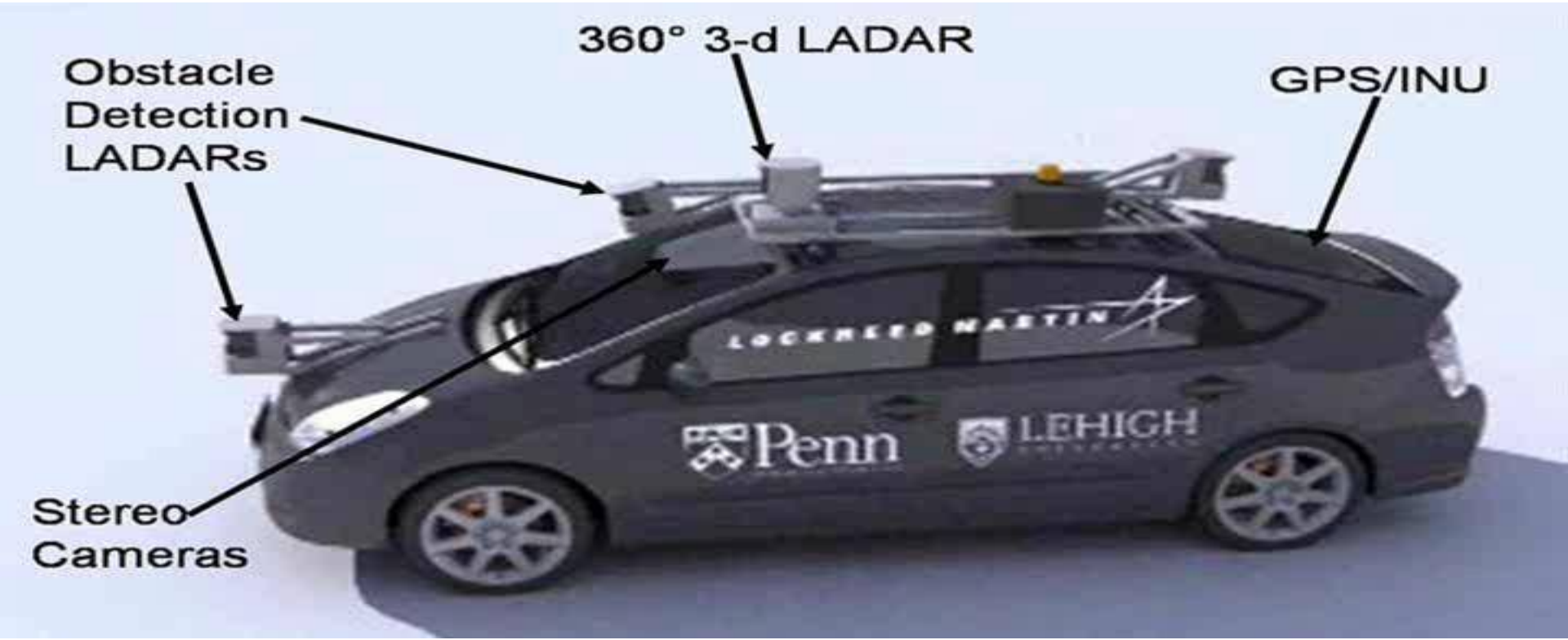
State of the Art Applications of Machine Learning

Autonomous Cars

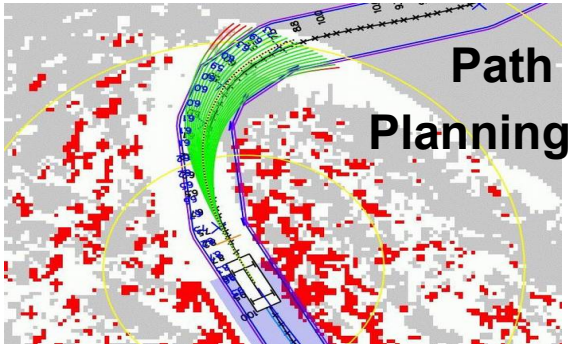


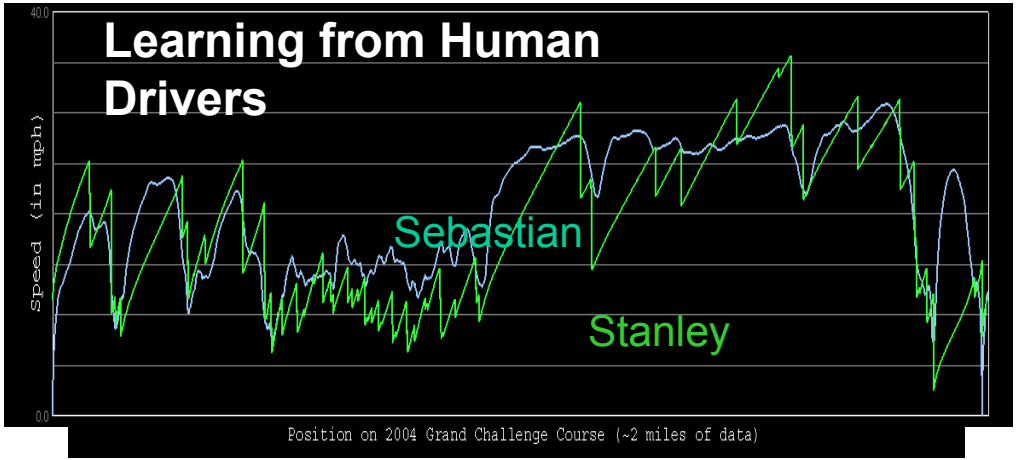
- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars
- Penn's Autonomous Car
  (Ben Franklin Racing Team)

Autonomous Car Sensors



Autonomous Car Technology





Images and movies taken from Sebastian Thrun's multimedia

Deep Learning in the Headlines

BUSINESS NEWS

MIT Technology Review

Is Google Cornering the Market on Deep Learning?

A cutting-edge corner of science is being wooed by Silicon Valley, to the dismay of some academics.

By Antonio Regalado on January 29, 2014

How much are a dozen deep-learning researchers worth? Apparently, more than $400 million.

This week, Google reportedly paid that much to acquire DeepMind Technologies, a startup based in

BloombergBusinessweek Technology

Acquisitions

The Race to Buy the Human Brains Behind Deep Learning Machines

By Ashlee Vance | January 27, 2014

intelligence projects. "DeepMind is bona fide in terms of its research capabilities and depth," says Peter Lee, who heads Microsoft Research.

According to Lee, Microsoft, Facebook (FB), and Google find themselves in a battle for deep learning talent. Microsoft has gone from four full-time deep learning experts to 70 in the past three years. "We would have more if the talent was there to

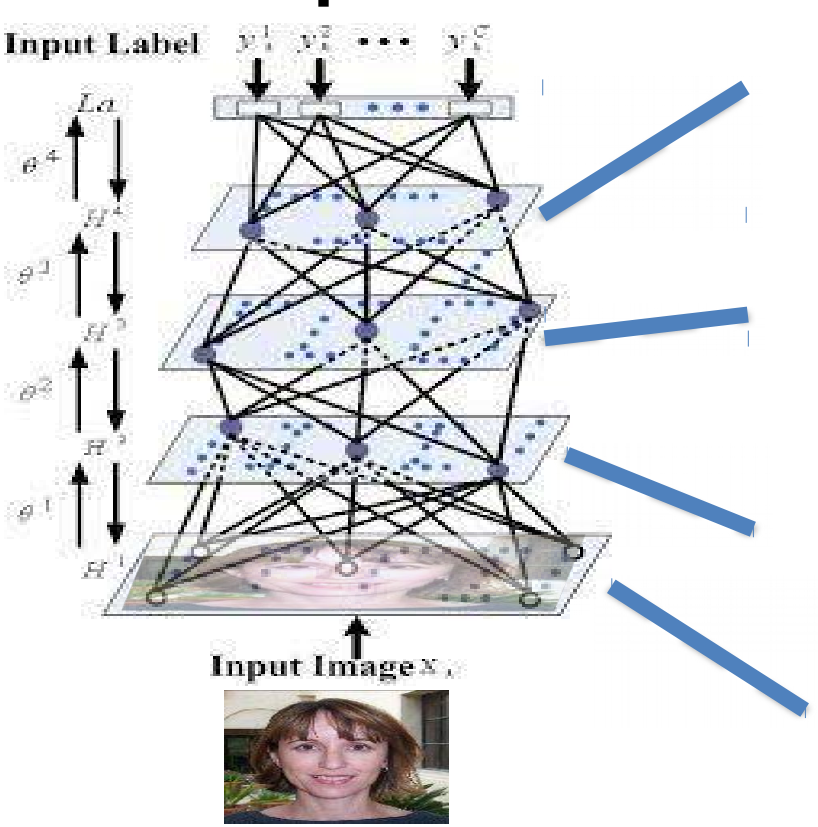
WIRED GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIG

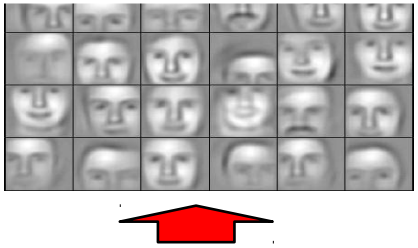
INNOVATION INSIGHTS | community content | featured

Deep Learning's Role in the Age of Robots

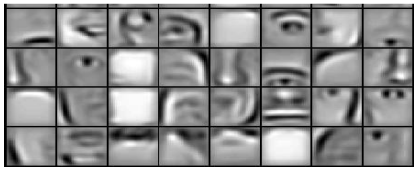
BY JULIAN GREEN, JETPAC 05.02.14 2:56 PM

Deep Belief Net on Face

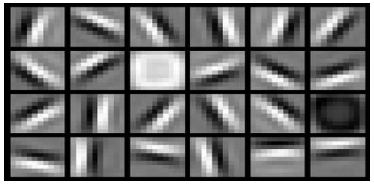




object models



object parts (combination of edges)

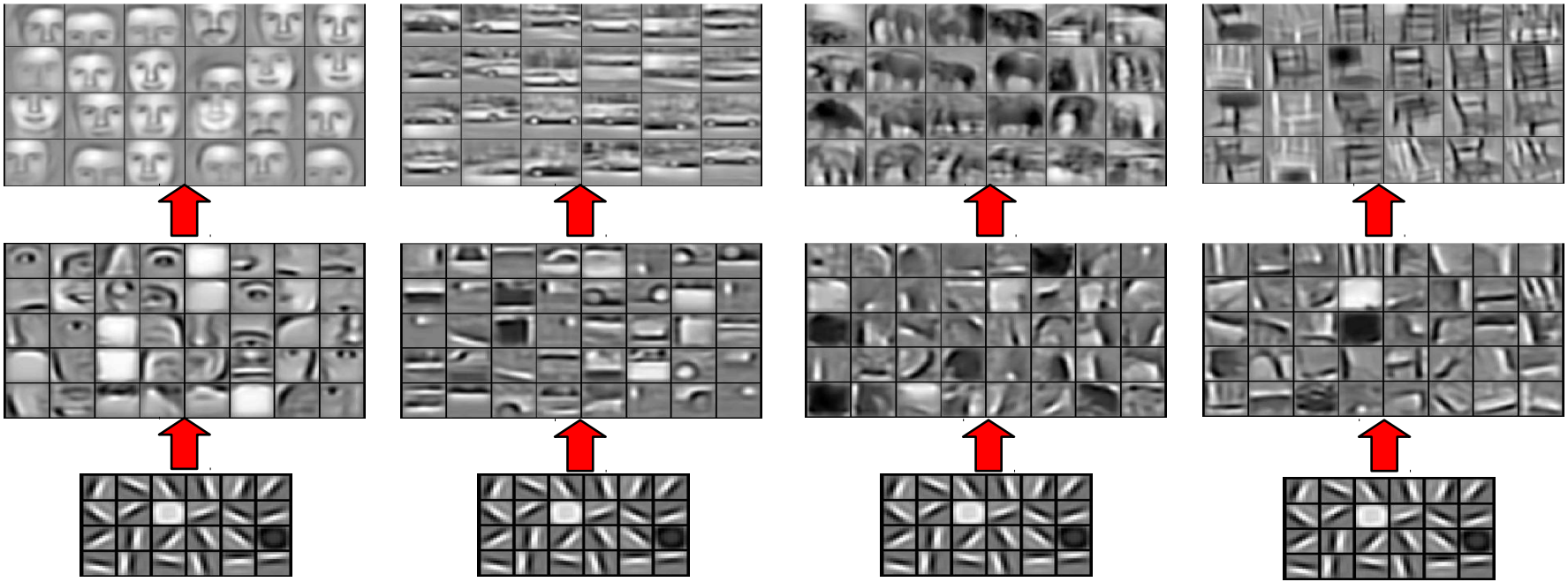


edges

pixels

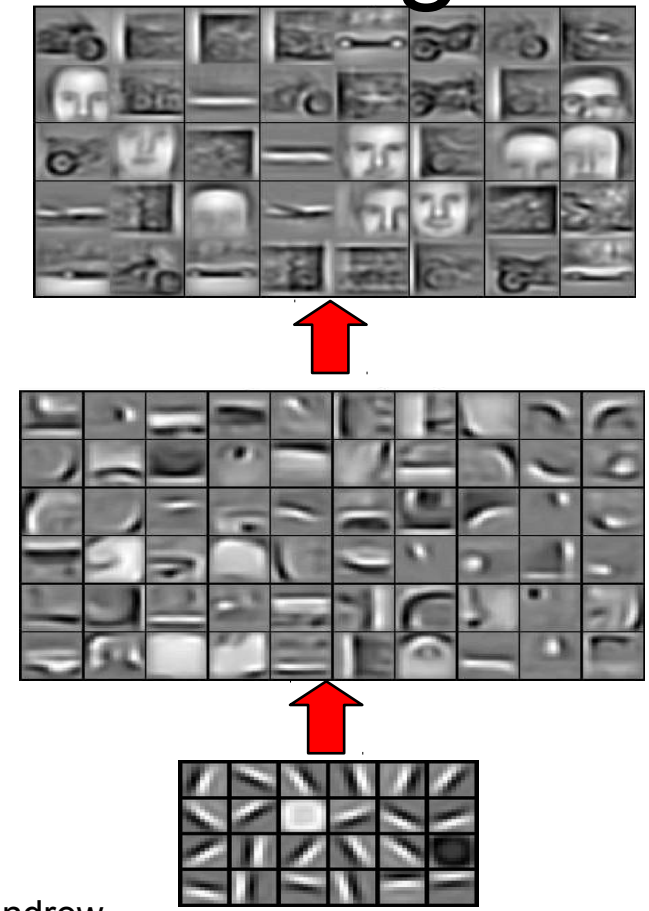
Based on materials by

Learning of Object Parts



Slide credit: Andrew

Training on Multiple Objects



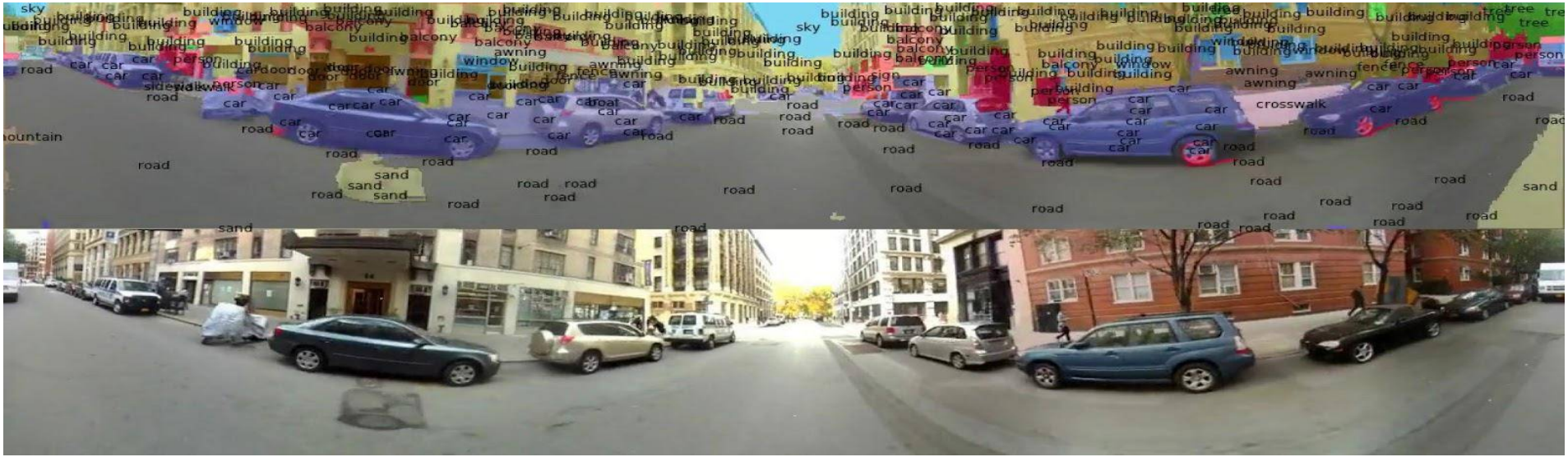
Trained on 4 classes (cars, faces, motorbikes, airplanes).

Second layer: Shared-features and object-specific features.

Third layer: More specific features.

Slide credit: Andrew

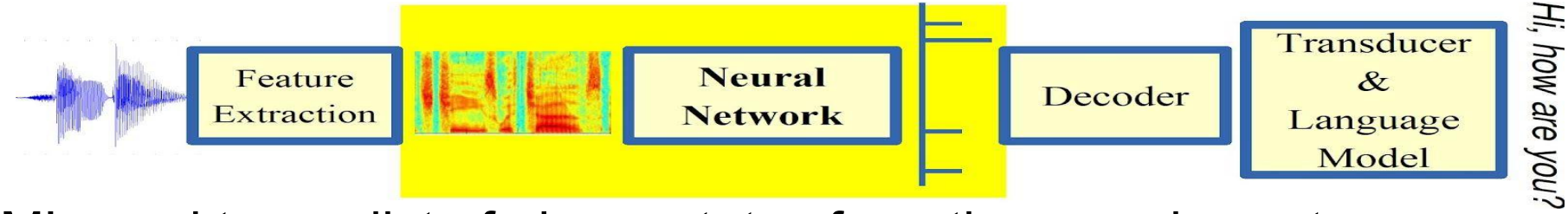
Scene Labeling via Deep Learning



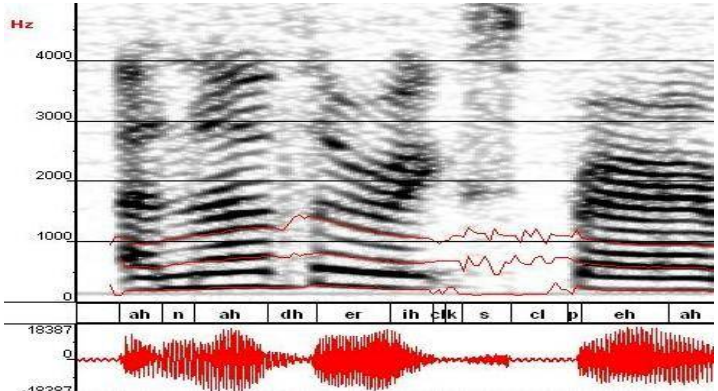
[Farabet et al. ICML 2012, PAMI 2013]

Machine Learning in Automatic Speech Recognition

A Typical Speech Recognition System



ML used to predict of phone states from the sound spectrogram



Deep learning has state-of-the-art

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

Baseline GMM performance = 15.4%

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

Impact of Deep Learning in Speech Technology

Slide credit: Li Deng, MS

Types of Learning

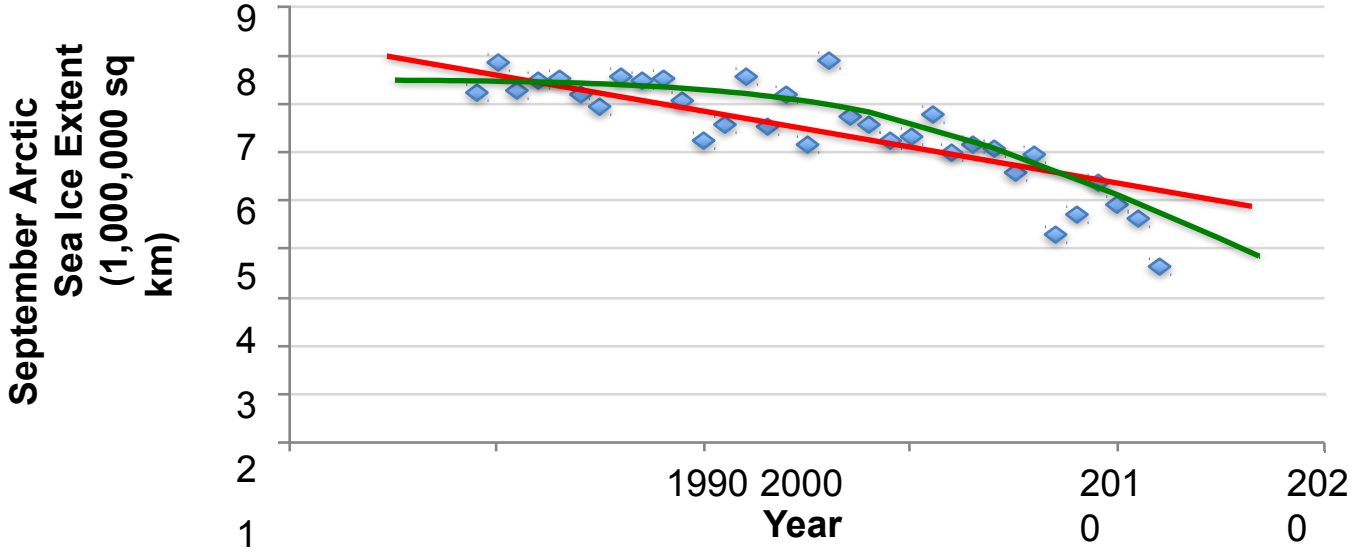
Types of Learning

- **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**
  - Rewards from sequence of actions

Based on slide by Pedro Domingos

Supervised Learning: Regression

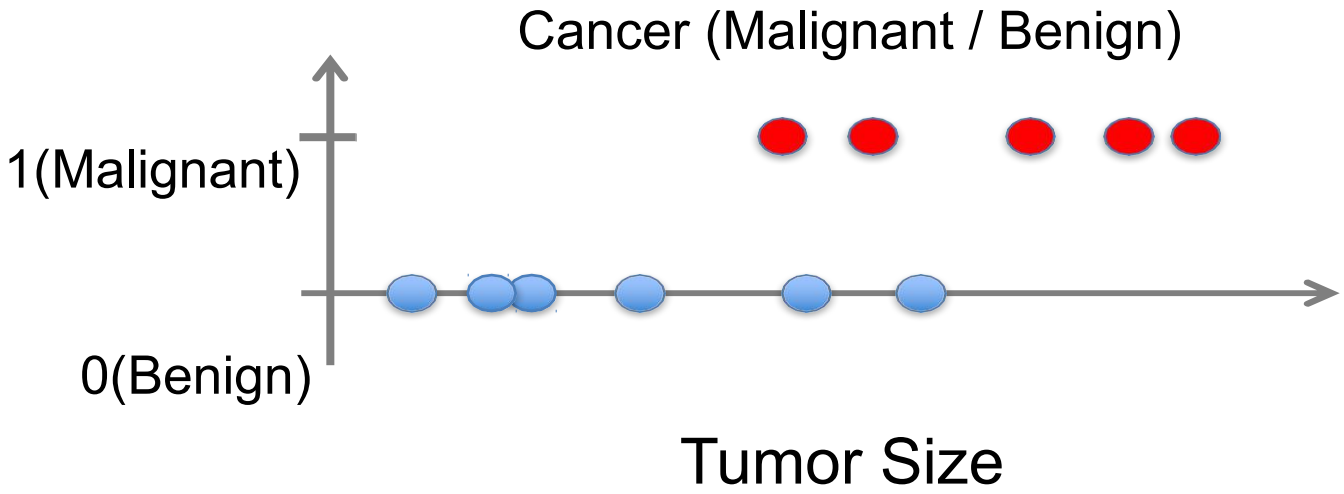
- Given (x_1, y_1), (x_2, y_2), …, (x_n, y_n)
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



Data from G. Witt, Journal of Statistics Education, Volume 21, Number

Supervised Learning: Classification

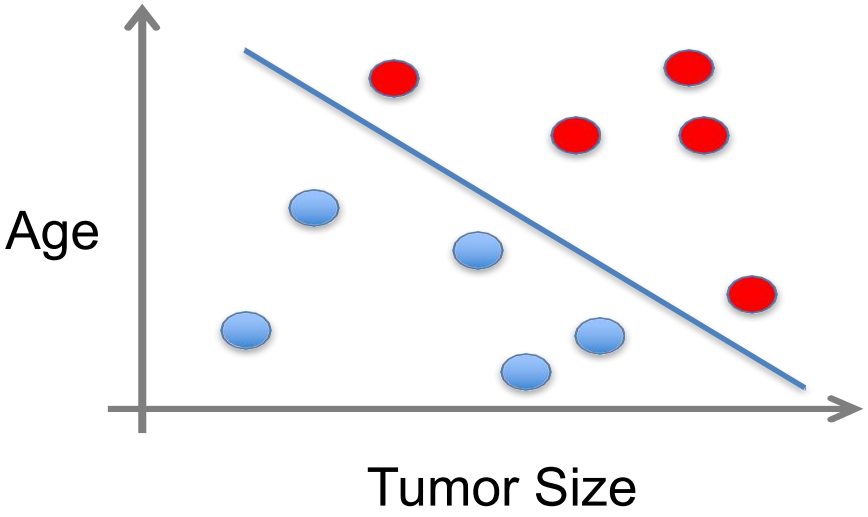
- Given (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew

Supervised Learning

- x can be multi-dimensional
  - Each dimension corresponds to an attribute



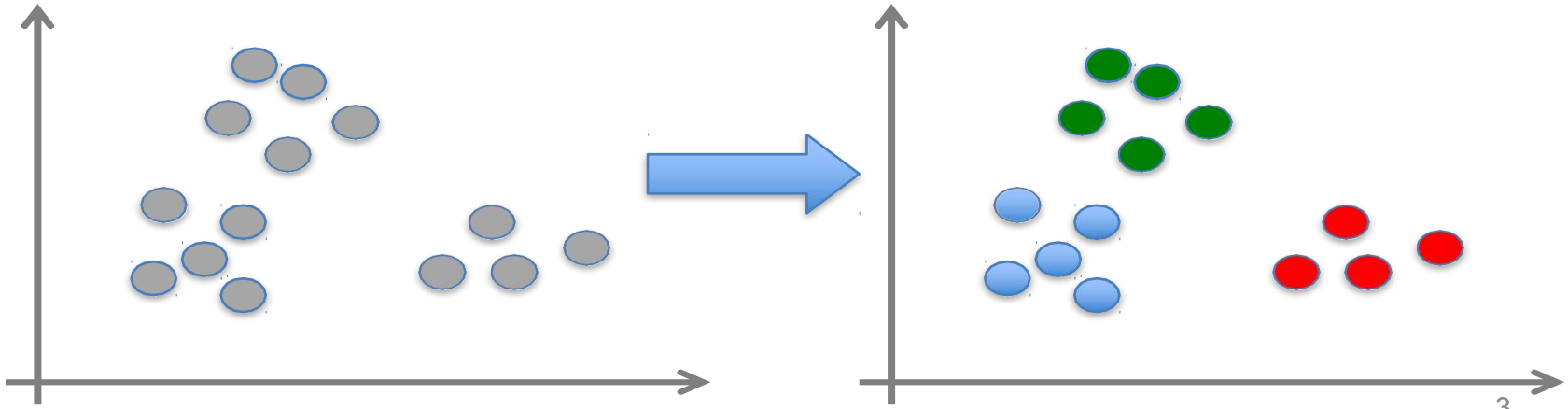
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

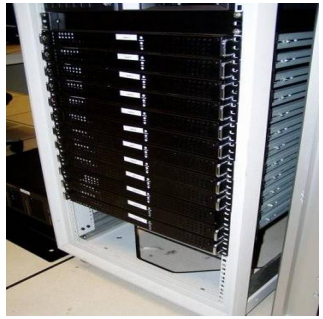
Based on example by Andrew

Unsupervised Learning

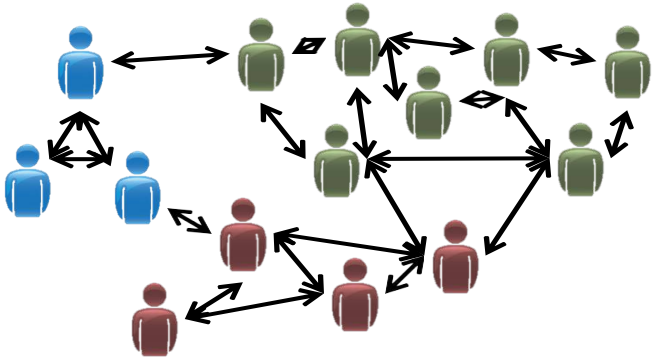
- Given x_1, x_2, ..., x_n (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised Learning



Organize computing clusters



Social network analysis

Market segmentation

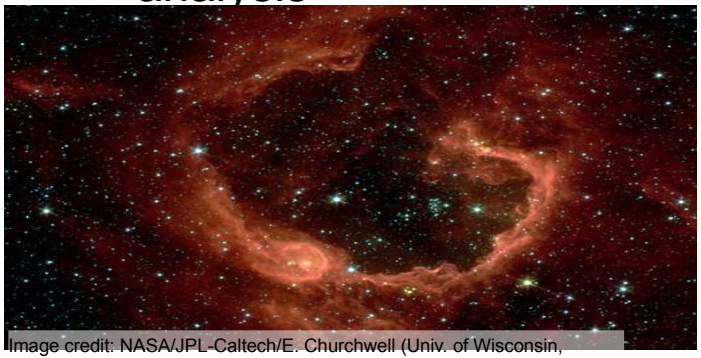


Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsin, Madison)

Astronomical data

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

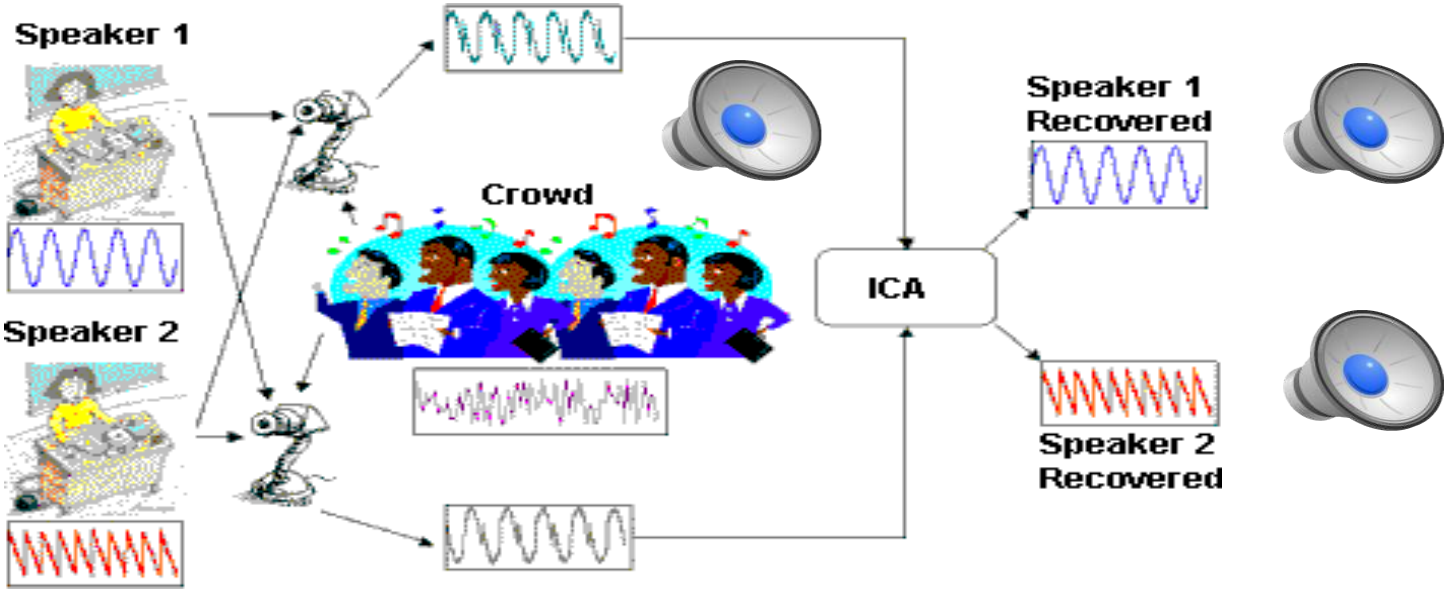


Image credit: statsoft.com Audio from

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

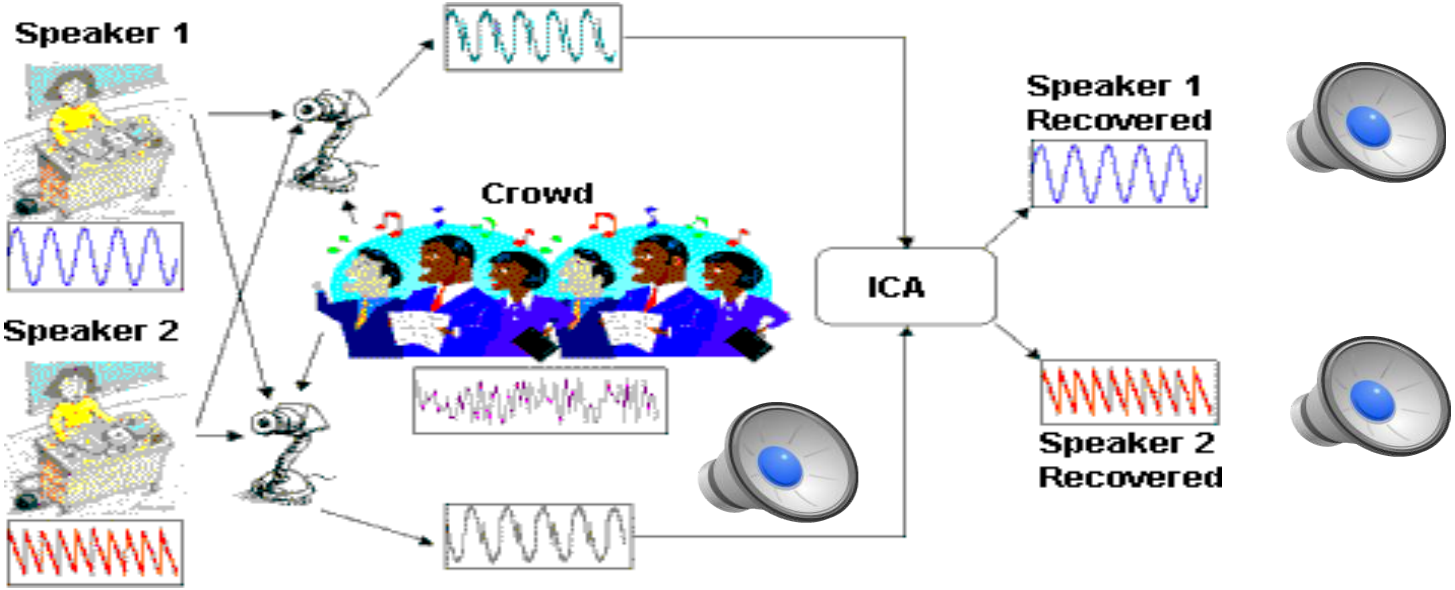
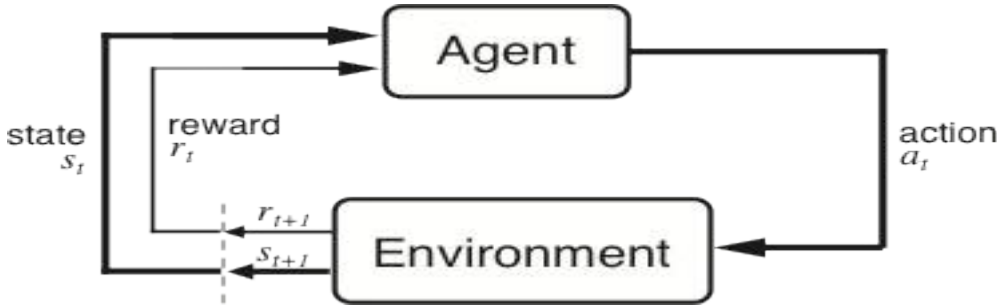


Image credit: statsoft.com Audio from

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state
- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand



Reinforcement Learning

https://

Inverse Reinforcement Learning

- Learn policy from user

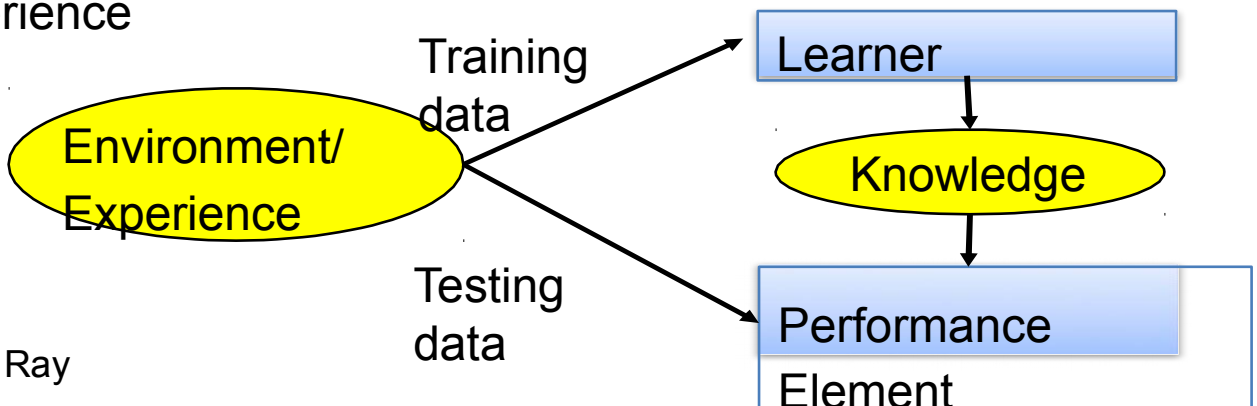
Stanford Autonomous Helicopter

http://heli.stanford.edu/ https://www.youtube.com/watch?v=VCdxqn0fcnE

Framing a Learning Problem

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned
  - i.e. the ***target function***
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Based on slide by Ray Mooney

Training vs. Test Distribution

- We generally assume that the 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"

- If examples are not independent, requires

  collective classification

- If test distribution is different, requires ***transfer learning***

Slide credit: Ray Mooney

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year
- Every ML algorithm has three components:
  - **Representation**
  - **Optimization**
  - **Evaluation**

Slide credit: Pedro

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

Slide credit: Ray

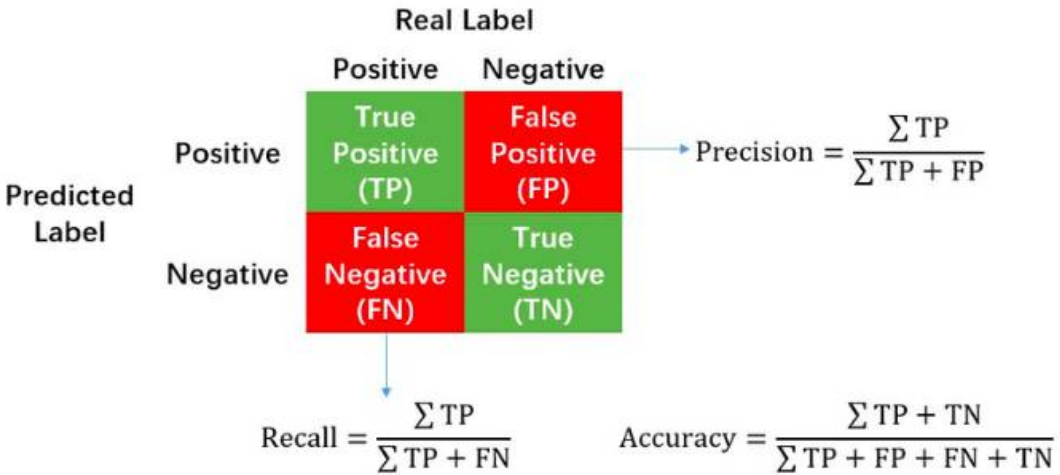
Various Search/Optimization Algorithms

- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

Slide credit: Ray

Evaluation

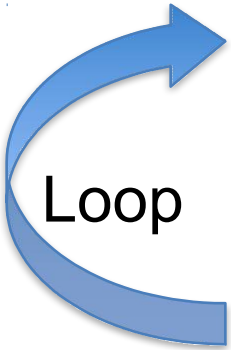
- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence etc



[A **true positive** is an outcome where the model *correctly* predicts the *positive* class. Similarly, a **true negative** is an outcome where the model *correctly* predicts the *negative* class.]

Slide credit: Pedro

ML in Practice



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

Based on a slide by Pedro

A Brief History of Machine Learning

History of Machine Learning

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

Slide credit: Ray

History of Machine Learning (cont.)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology

Slide credit: Ray

History of Machine Learning (cont.)

- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

Slide credit: Ray

History of Machine Learning (cont.)

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision

Based on slide by Ray

History of Machine Learning (cont.)

- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.
  - Many More To Come!!!!!

Based on slide by Ray

Summary

Machine learning seeks to **develop methods for computers to improve their performance at certain tasks** on the basis of observed data.

Typical example:

- detecting pedestrians in images taken from an autonomous vehicle,
- classifying gene-expression patterns from leukaemia patients into subtypes by clinical outcome,
- or translating English sentences into French.
- scope of machine-learning tasks is even broader than these pattern classification or mapping tasks, and can include optimization and decision making, compressing data and automatically extracting interpretable models from data.

Data: The Treasure

- Almost all machine-learning tasks can be formulated as making inferences about missing or latent data from the observed data [inference, prediction or forecasting]

  Example :

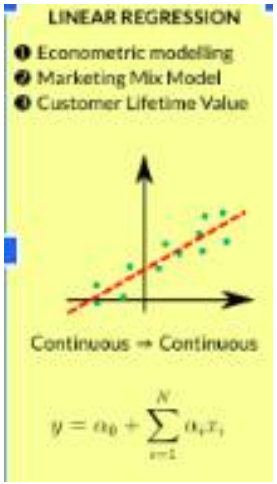
  - Consider classifying people with leukaemia into one of the four main subtypes of this disease on the basis of each person's measured gene-expression patterns. Observed data : pairs of gene-expression patterns and labelled subtypes, Unobserved or missing data: subtypes for new patients.
    - **To make inferences about unobserved data from the observed data:**
      - **the learning system needs to make some assumptions**
      - **taken together these assumptions constitute a model.**

Model

- A model can be very simple and rigid [classic statistical linear regression model]
- Model can be complex and flexible [llarge and deep neural network]
- A model is considered to be well defined if it can make forecasts or predictions about unobserved data having been trained on observed data
    - if the model cannot make predictions it cannot be falsified, in the sense of the philosopher Karl Popper's proposal for evaluating hypotheses, or as the theoretical physicist Wolfgang Pauli said the model is "not even wrong").

Any sensible model will be uncertain when predicting unobserved data, uncertainty plays a fundamental part in modelling.

Rigid Linear Regression

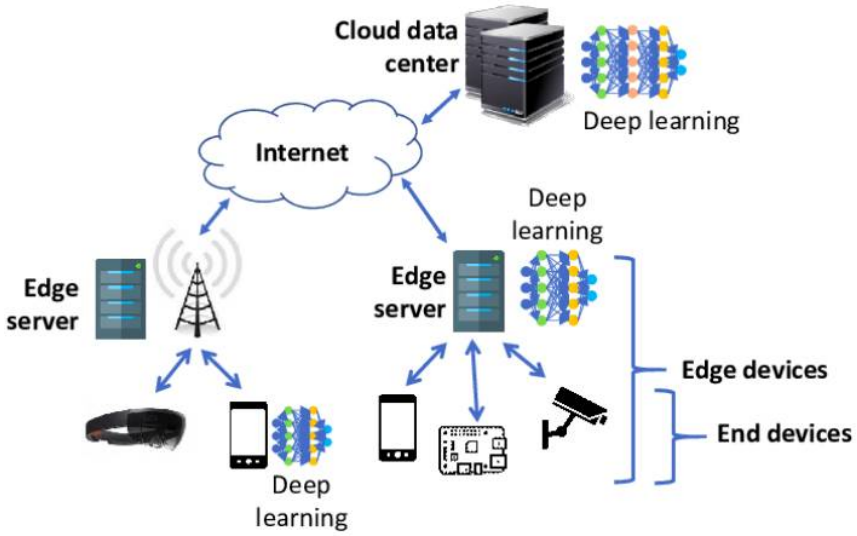


- Linear Regression is greatly affected by the presence of Outliers, linearity assumption
  - An outlier is an unusual observation of response y, for some given predictor x.
- Logistic regression assumes **linearity of independent variables and log odds**.

Flexible deep NN

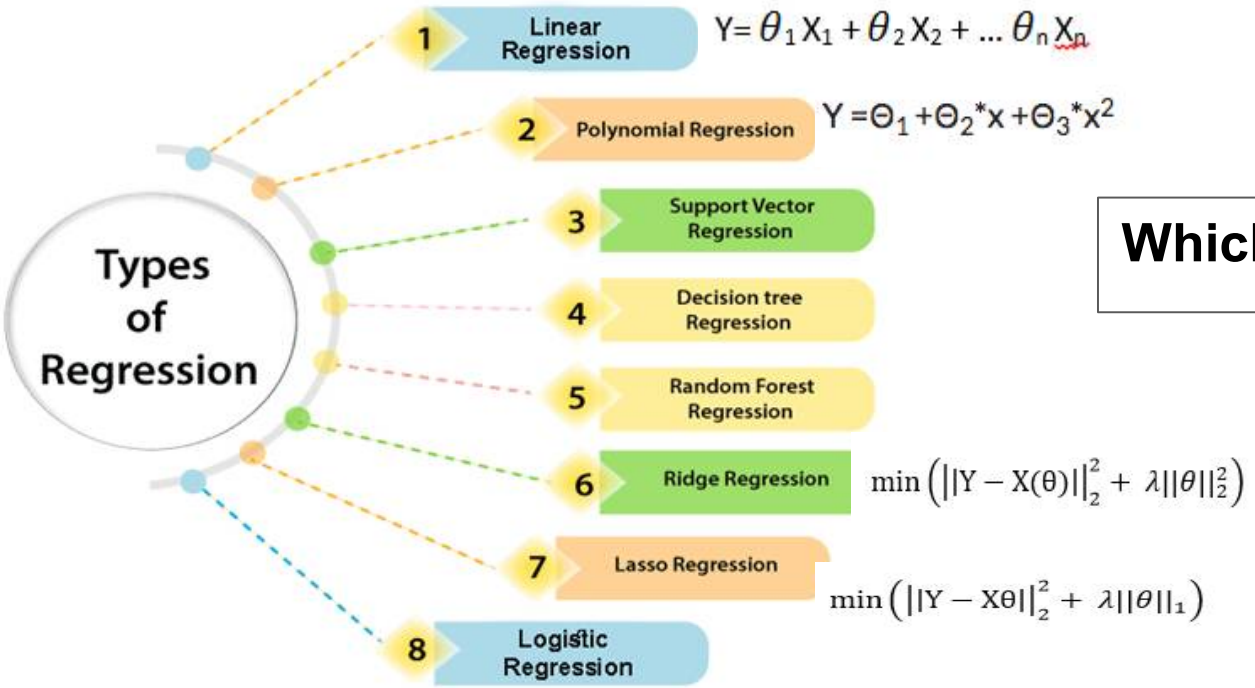
DNNs is a challenge since they typically require billions of expensive arithmetic computations.

- DNNs are typically deployed in ensemble to boost accuracy performance, which further exacerbates the system requirements.
- This computational overhead is an issue for many platforms, e.g. data centers and embedded systems, with tight latency and energy budgets.
- Flexible DNNs: achieves large reduction in average inference latency while incurring small to negligible



Deep learning can execute on edge devices (i.e., end devices and edge servers) and on cloud data centers.

Regression analysis helps in the prediction of a continuous variable



Which model to choose?

Problem 1: Uncertainty in Modelling

Uncertainty in Modelling

- At the lowest level, model uncertainty is introduced from measurement noise, for example, pixel noise or blur in images.
- At higher levels, a model may have many parameters, such as the coefficients of a linear regression, and there is uncertainty about which values of these parameters will be good at predicting new data.
- Finally, at the highest levels : uncertainty about even the general structure of the model:
    - Is linear regression or a neural network appropriate, if the latter, how many layers should it have, and so on.

Probabilistic approach to modelling

- uses probability theory to express all forms of uncertainty .
- probability distributions are used to represent all the uncertain unobserved quantities in a model
    - structural, parametric and noise-related
    - how they relate to the data.

Probabilistic approach to modelling

- basic rules of probability theory are used to infer the unobserved quantities given the observed data.
- Learning from data occurs through the transformation of the prior probability distributions (defined before observing the data), into posterior distributions (after observing data).
- **The application of probability theory to learning from data is called Bayesian learning**

Bayesian machine learning: How to choose ?

There are two simple rules that underlie probability theory: the sum rule:

$$P(x) = \sum_{y \in Y} P(x,y)$$

and the product rule:

$$P(x,y) = P(x)P(y \mid x).$$

- x and y: observed or uncertain quantities, taking values in some sets X and Y, respectively.
- For example, x and y might relate to the weather in Cambridge and London, respectively, both taking values in the set X=Y={rainy,cloudy,sunny}
- P(x) : probability of x
- P(x,y) is the joint probability of observing x and y,
- P(y|x) is the probability of y conditioned on observing the value of x

Bayesian machine learning: How to choose?*

- Since P(x,y) and P(y,x) are commutative: $$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)} = \frac{P(x \mid y)P(y)}{\sum_{y \in Y} P(x,y)}$$

- Apply probability theory to machine learning by replacing the symbols above:
  - Replace x by D to denote the observed data
  - Replace y by θ to denote the unknown parameters of a model

$$P(\theta \mid D, m) = \frac{P(D \mid \theta, m)P(\theta \mid m)}{P(D \mid m)}$$

where $P(D \mid \theta, m)$ is the likelihood of parameters θ in model m, $P(\theta \mid m)$ is the prior probability of θ and $P(\theta \mid D, m)$ is the posterior of θ given data D.

Ref: *https://www.nature.com/articles/nature14541.pdf

Compositional probabilistic models

- Simple probability distributions over single or a few variables can be composed to form the building blocks of larger, more complex models.
- Representing such compositional probabilistic models: graphical models

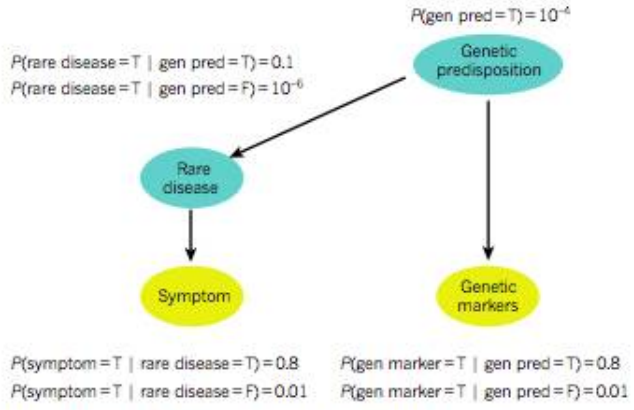


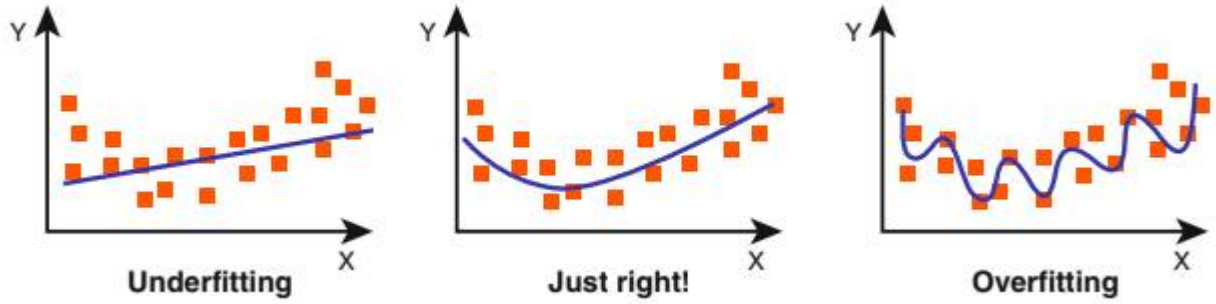
Figure 1 | Bayesian inference. A simple example of Bayesian inference applied to a medical diagnosis problem. Here the problem is diagnosing a rare disease using information from the patient's symptoms and, potentially, the patient's genetic marker measurements, which indicate predisposition (gen pred) to this disease. In this example, all variables are assumed to be binary. T, true; F, false. The relationships between variables are indicated by directed arrows and the probability of each variable given other variables they directly depend on is also shown. Yellow nodes denote measurable variables, whereas green nodes denote hidden variables. Using the sum rule (Box 1), the prior probability of the patient having the rare disease is: $P(\text{rare disease} = T) = P(\text{rare disease} = T|\text{gen pred} = T)\,p(\text{gen pred} = T) + p(\text{rare disease} = T|\text{gen pred} = F)\,p(\text{gen pred} = F) = 1.1 \times 10^{-5}$. Applying Bayes rule we find that for a patient observed to have the symptom, the probability of the rare disease is: $p(\text{rare disease} = T|\text{symptom} = T) = 8.8 \times 10^{-4}$, whereas for a patient observed to have the genetic marker (gen marker) it is $p(\text{rare disease} = T|\text{gen marker} = T) = 7.9 \times 10^{-4}$. Assuming that the patient has both the symptom and the genetic marker the probability of the rare disease increases to $p(\text{rare disease} = T|\text{symptom} = T, \text{gen marker} = T) = 0.06$. Here, we have shown fixed, known model parameters, that is, the numbers $\theta = (10^{-4}, 0.1, 10^{-6}, 0.8, 0.01, 0.8, 0.01)$. However, both these parameters and the structure of the model (the presence or absence of arrows and additional hidden variables) could be learned from a data set of patient records using the methods in Box 1.

Problem 2: Overfitting of ML nets

Why Overfitting?

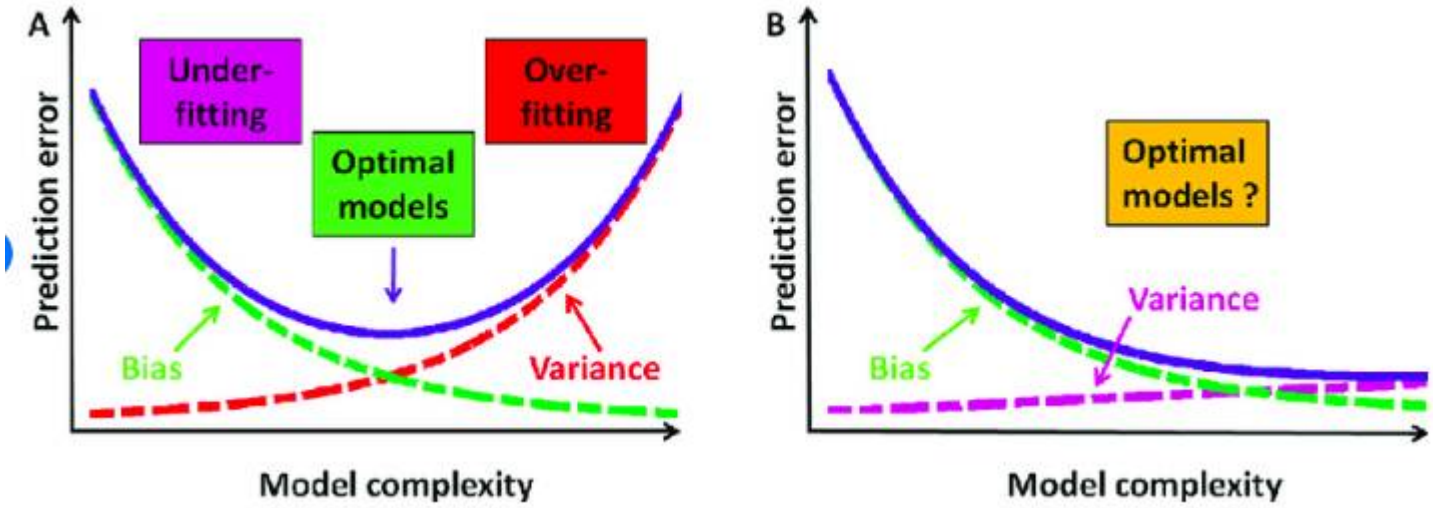
- Overfitting is **a modeling error in statistics** that occurs when a function is too closely aligned to a limited set of data points
- Example:
  - A traffic sign recognition system has to learn signs, but all STOP signs are only captured in an alley.
  - The ML algorithm is likely to learn, that a sign in an alley is a STOP sign and therefore detecting every traffic sign in an alley as such. Or a STOP sign, that is not located in an alley can not be detected.
  - This is caused by a **low variance in the data, with a high bias**

Overfitting underfitting details



A network is overfitting when a model's training error (computed on a training set) is much lower than its generalization error (computed on a test or validation set).

This is opposite to underfitting, when a model is not able to obtain a sufficiently low error value on its training set.



- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.
- Variance refers to the changes in the model when using different portions of the training data set [**variance** is the expectation of the squared deviation of a random variable from its population mean or sample mean.]

Problem 3: Edge Cases

Edge Cases

- Unexpected scenarios that occur so irregularly or even seldom are called Edge Cases
- often not covered in any training data, simply because they are not anticipated during the system development.
- All can possibly lead to accidents, when the systems deals with them in a wrong way.

Example:

- A vehicle that is only used in sunny weather for a long time in the same environment, will override some of the learned features trained by the manufacturer.
- handling intersections and performing turns as well as crossing the intersection.

Solution

- Learning algorithms could continue to **evolve while in operation**
- this approach is also called continuous or on-the-fly learning.
- Actually this approach is often seen as similar to human learning processes.
- Drawback:
  - potentially unwanted behavior
  - the training can not be influenced in a suitable way
  - dynamic changes to a certified system in general is not permitted by the current regulations and standards

Test Case

- To understand the implication of critical components based on machine learning, HAW Hamburg uses miniature vehicles

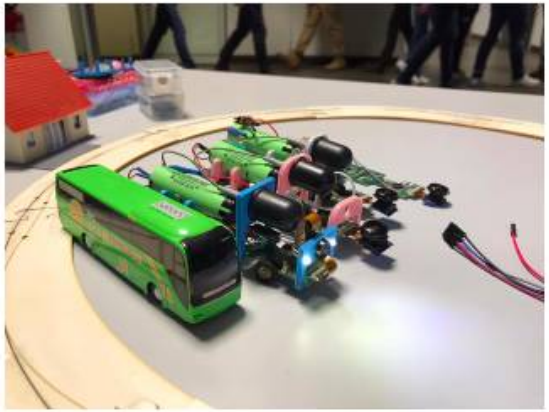


Fig. 3: Autonomous miniature vehicle type 2 (“truck”, large). The latter is designed to carry an FPGA board while the former is controlled by an ESP32 micro controller. (©Wunderland, 2019; Tiedemann, 2019, under CC-BY 4.0)

Testing on use cases:

- obstacle detection for autonomous driving.
- recognition of traffic signs
- obstacles in the driving lane
- route detection
- Fault tree analysis

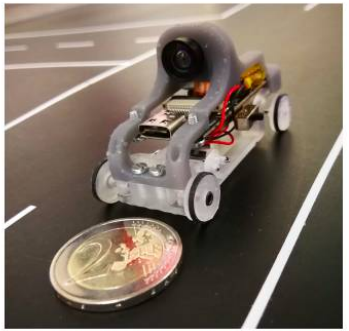


Fig. 2: Autonomous miniature vehicle type 1 (“sedan”, small) (©Tiedemann, 2019, under CC-BY 4.0)

In Summary

As a concrete example, consider *adversarial examples*, small perturbations of input examples that make even a highly accurate ML model give incorrect predictions.

- Adversarial examples can be used to regularize the training procedure and make a model robust to small perturbations of data (which is *a special case of stability*).
- Adversarial examples can be used as explanations by providing the minimal changes in the input that would alter the model prediction on it (*counterfactual explanations*).
- Adversarial examples that only change certain protected attributes like gender or race can be used to verify and optimize for fairness (*fairness audit*).

Refs

- Torge Hinrichs, Bettina Buth:**Can AI-based Components be Part of Dependable Systems?** IV 2020: 226-231

- **Reliable Machine Learning,** https://www.microsoft.com/en-us/research/group/reliable-machine-learning/