Why study machine learning

- AI is the enterprise of design and analysis of intelligent agents.
- Intelligent behavior requires knowledge (e.g., model of the environment)
- Explicitly specifying the knowledge needed for specific tasks is hard, and often infeasible
- How to acquire knowledge?
  - Learning
Why study machine learning

- Learning modifies the agent's decision mechanisms to improve performance
- Environment changes over time – adapt to changes
- Learning is essential for unknown environments, i.e., when designer lacks omniscience

Why study machine learning

- Applications
  - Medical diagnosis/image analysis (e.g. pneumonia, pap smears)
  - DNA sequence identification
  - Scientific discovery
  - Spam Filtering, fraud detection (e.g. credit cards, phone calls)
  - Search and recommendation (e.g. google, amazon)
  - Automatic speech recognition & speaker verification
  - Locating/tracking/identifying objects in images & video (e.g. faces)
  - ...
Data Mining

- Huge amounts of data available
  - Sources: business, science, medicine, economics, geography, environment, sports, ...
- Data is a potentially valuable resource
- Raw data is useless: need techniques to automatically extract information from it
  - Data: recorded facts
  - Information: patterns underlying the data
- Machine learning techniques: automatically finding patterns in data

Examples: The weather problem

- Conditions for playing a certain game

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>Yes</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Classification rule:

  If outlook = sunny and humidity = high then play = no
  If outlook = rainy and windy = true then play = no
  If outlook = overcast then play = yes
  If humidity = normal then play = yes
  If none of the above then play = yes
## The contact lenses data

<table>
<thead>
<tr>
<th>Age</th>
<th>Spectacle prescription</th>
<th>Astigmatism</th>
<th>Tear production rate</th>
<th>Recommended lenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Young</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>Soft</td>
</tr>
<tr>
<td>Young</td>
<td>Myope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Young</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Young</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>Soft</td>
</tr>
<tr>
<td>Young</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Young</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Normal</td>
<td>Hard</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>Soft</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Myope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>Soft</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Pre-presbyopic</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Myope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Myope</td>
<td>No</td>
<td>Normal</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Myope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Myope</td>
<td>Yes</td>
<td>Normal</td>
<td>Hard</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Hypermetrope</td>
<td>No</td>
<td>Normal</td>
<td>Soft</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Reduced</td>
<td>None</td>
</tr>
<tr>
<td>Presbyopic</td>
<td>Hypermetrope</td>
<td>Yes</td>
<td>Normal</td>
<td>None</td>
</tr>
</tbody>
</table>

---

## A decision tree for this problem

![Decision Tree Diagram]

- **Tear production rate**
  - Reduced
  - Normal
    - **Astigmatism**
      - **No**
        - **Spectacle Prescription**
          - **Soft**
            - **Hypermetrope**
              - **Hard**
              - **None**
          - **Myope**
          - **None**
      - **Yes**
        - **Spectacle Prescription**
          - **Soft**
          - **Myope**
          - **None**
Predicting CPU performance

- Example: 209 different computer configurations

<table>
<thead>
<tr>
<th>Cycle time (ns)</th>
<th>Main memory (Kb)</th>
<th>Cache (Kb)</th>
<th>Channels</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYCT</td>
<td>MMIN</td>
<td>MMAX</td>
<td>CACH</td>
<td>CHMIN</td>
</tr>
<tr>
<td>1</td>
<td>125</td>
<td>256</td>
<td>6000</td>
<td>256</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>8000</td>
<td>32000</td>
<td>32</td>
</tr>
<tr>
<td>...</td>
<td>208</td>
<td>480</td>
<td>512</td>
<td>8000</td>
</tr>
<tr>
<td>209</td>
<td>480</td>
<td>1000</td>
<td>4000</td>
<td>0</td>
</tr>
</tbody>
</table>

- Linear regression function

\[ PRP = -55.9 + 0.0489 \text{MYCT} + 0.0153 \text{MMIN} + 0.0056 \text{MMAX} + 0.6410 \text{CACH} - 0.2700 \text{CHMIN} + 1.480 \text{CHMAX} \]

Handwritten Digit Recognition

\[
\begin{array}{cccc}
0 & 1 & 2 & 3 \\
4 & 5 & 6 & 7 \\
8 & 9 & & \\
\end{array}
\]
Supervised Learning Problem

Machine learning approach to Handwritten Digit Recognition

- Input to learning: a set of labeled instances -> dataset
  - Instance
    - Individual, independent example of target concept
    - Characterized by a predetermined set of attributes/features
    - Represented as a feature vector \( x=(X_1, X_2, \ldots) \)
  - A training example is a pair \((x, t)\), \(t:\) target value (category of the digit)
  - Training dataset \( \{(x_i, t_i)\} \)

Supervised Learning (Cont.)

- Learning/training phase: find a model \( h(x) \)
  - Choose model/hypothesis space
    - Linear models
    - Decision trees
    - Support vector machines
    - Neural networks
  - Problem: find the best model/hypothesis \( h(x) \) given the training dataset
  - Use learned \( h(x) \) to categorize/predict new instances
Canonical Learning Problems

- Supervised Learning: given examples of inputs and corresponding desired outputs, predict outputs on future inputs.
  - Classification: target has finite domain - categories
  - Regression: target has continuous domain

- Unsupervised Learning: given only inputs, automatically discover representations, features, structure, etc.
  - Clustering: group similar instances, e.g. automatically group (unlabeled) handwritten digits

- Reinforcement Learning: occasional rewards or punishments
  - how an agent ought to take actions in an environment so as to maximize some notion of long-term reward in sequential decision problems, e.g., learn to play chess without human instruction

Mainstream Machine Learning
List of Topics

- Bayesian decision theory.
- Maximum-Likelihood and Bayesian parameter estimation.
- Model Evaluation
- Supervised learning
  - Naive Bayes Classifier
  - Nearest neighbor methods
  - Linear models
  - Decision trees
  - (Deep) Neural networks
  - Support vector machines
  - Ensemble learning

List of Topics (cont.)

- Probabilistic graphical models: Bayesian networks, Markov random fields
- Unsupervised learning:
  - Clustering: mixture models, K-means, Hierarchical Clustering
  - PCA, Linear Factor Models, ICA
- Sequential data: HMMs, Kalman Filters, Recurrent Neural Networks
- Markov decision process and Reinforcement Learning
- Selected applications throughout
The neural network “renaissance”

- A massive resurgence of interest in neural networks
- Deep learning “revolution”
- The striking successes of deep learning techniques on key benchmark problems
- Starting around 2012, impressive results were achieved on long-standing problems in speech recognition and computer vision
- Deep learning techniques won most of competitive machine learning challenges
Feedforward Network

- A three-layer neural network consists of an input layer, a hidden layer and an output layer interconnected by modifiable weights represented by links between layers.

- "neuron" or "unit" are used interchangeably.

- Multiple layers of cascaded linear neurons still produce linear functions.

- The novel computational power provided by multilayer neural nets can be attributed to the nonlinear mapping of the input to the representation at the hidden units, since the hidden-to-output layer leads to a linear discriminant.

Deep Neural Network

[Diagram of a deep neural network with multiple hidden layers and input/output nodes.]
Network Structures

- *Feed-forward* networks: no feed-back loops

- Layered networks: having successive layers of units, with connections running from every unit in one layer to every unit in the next layer.

- More than one hidden layer; skip-layer connections

- *Convolutional networks* connect to a subset of units in the next layer: for computer vision

- *Recurrent networks*: for sequential data such as speech recognition, natural language applications

Expressive Power of multi-layer Networks

- Universal Function Approximation Theorem
  
  “Any bounded continuous function from input to output can be approximated with arbitrarily small error by a network with one hidden layer, given sufficient number of hidden units $n_H$, proper nonlinearities, and weights.”

- Any decision boundaries can be implemented as a three-layer neural network

- Any function can be approximated to arbitrary accuracy by a network with two hidden layers.
Deep Learning Topics

- Adversarial Machine Learning
- Interpretable ML, interpretation of learned representations
- Architectures: CNN, RNN GNN, GCN, Attention Models
- Encoding the domain symmetries: equivariance
- Physics-guided NN: combining knowledge of physics-based models
- Multitask and Transfer Learning, Few-Shot Learning
- Semi-Supervised Learning
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoder