CE213 Artificial Intelligence – Lecture 10

Introduction to Machine Learning

Problems in the AI approaches learnt so far:
- Inefficient search strategies
- Ineffective evaluation functions
- Ineffective representation of knowledge
- Lack of knowledge

Machine learning - A new approach to AI
- data-driven vs. knowledge-driven
- knowledge representation learning
- new methods for problem solving

Why Machine Learning?

Dog or Cat?

Will the stock market go up or down?

Kaggle competition: Create a program to distinguish dogs from cats.
State of the art: a little kid may outperform the best computer (98.9%).

Can you use 'generate and evaluate' method for solving these problems?
Can you build expert systems for solving these problems?
Key issues: complex patterns, huge search space,
inefficient search methods,
ineffective knowledge representation,
ineffective evaluation functions.

What Is Machine Learning?

Machine learning is a branch of artificial intelligence focusing on **how to get computers to learn from data/experience**.

Very roughly:

Programming is telling a machine *what* to do and *how* to do it.

Machine learning is showing a machine *what* we want it to do and expecting it to figure out *how* to do it.

There could be more complicated situations:

Sometimes we cannot even show the machine what we want because we do not really know. A smart machine can even help in this situation through machine learning.

A short video introducing machine learning using layman's language:

https://www.youtube.com/watch?v=f_uwKZIAeM0

A More Specific Definition and Key Elements of Machine Learning

Machine learning can be defined as updating/optimising a model by a learning algorithm based on sample data and certain performance measure, so as to improve the performance of the model for a given task, so there must be

A ***task or problem*** and an associated ***performance measure***

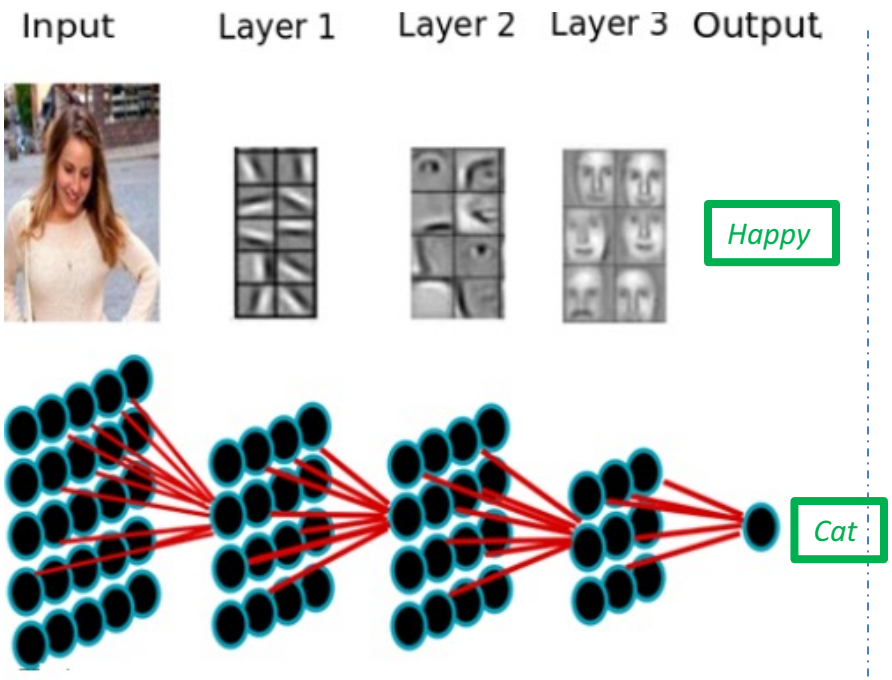
A ***learning environment*** or a ***set of sample data***

A ***model***

A ***learning algorithm***

Does a machine learner need a teacher? Where is the teacher?

How Does Machine Learning Work ?



(preprocessing & feature extraction & classification)

(partly from www.kdnuggets.com)

Neural network model (one hidden layer) :

$$y_k = f(\sum_{i=1}^{n_h} w_{ki}^o \cdot h_i - \theta_k^o)$$

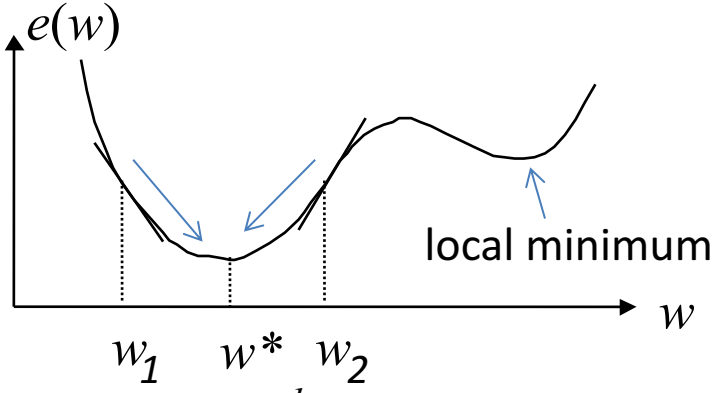
$$= f(\sum_{i=1}^{n_h} w_{ki}^o \cdot f(\sum_{j=1}^{n} w_{ij}^h \cdot x_j - \theta_i^h) - \theta_k^o)$$

Learning algorithm (gradient descent):

$$\Delta w = \alpha(z - y)\frac{dy}{dw} = -\alpha\frac{de}{dw}$$

$$e = \frac{1}{2}(z - y)^2$$

Error reduction by learning:



$$(\frac{de}{dw} < 0,\ \Delta w > 0) \quad (\frac{de}{dw} > 0,\ \Delta w < 0)$$

Learning Viewed as Search for Optimal Models

A model for performing a task can be defined by its structure and parameters,

$$\text{e.g., } y = f(x, W)$$

y – output, x – input, f – model structure, W – model parameters.

The process of learning can be viewed as ***searching*** the space of possible representations for a model (structure or/and parameters) that maximises the performance measure.

If we fix model structure f, then learning is to search for optimal model parameters W (example on previous slide). This is **parametric learning**.

In the lecture next week, the model involved is a decision tree that is not fixed, but constructed by learning. When the model structure f is not fixed, but updated/optimised by learning, it is **structural learning**.

A Taxonomy of Learning Tasks

Learning to classify

Given a set of training examples and their associated class labels,

Learn to correctly predict the classification of unclassified examples.

e.g., a parent teaching a child to recognise animals by showing the child pictures of animals with associated names.

Learning to predict numerical values (regression or approximation)

Given a set of training examples and associated numerical values,

Learn to correctly predict the numerical value for other examples in which it is not known.

e.g., learning to predict tomorrow's temperature from recorded weather data or learning to evaluate game states.

Both of these are sometimes called ***supervised learning***.

A Taxonomy of Learning Tasks (2)

Learning to form groups (clustering)

Given a set of unclassified/unlabelled examples,

Develop a "sensible" scheme for classifying them.

e.g., learning to group the cities around the world

Clustering is often called ***unsupervised learning***.

Learning what to do next (*reinforcement learning*)

Given the experience of engaging actively in a task,

Learn to improve performance when engaged in similar tasks in future

e.g., learning to play game of Go, robot navigation

Topics on Machine Learning for CE213

Decision Tree Induction (structural learning)

A classical 'learning to classify' approach

Neural Networks (parametric and/or structural learning)

McCulloch-Pitts neuron model

Multilayer neural networks and error back-propagation

Learning to predict or classify

Clustering (unsupervised learning)

K-means, Agglomerative hierarchical clustering

Reinforcement Learning

The Q learning algorithm

Genetic Algorithms

A Very Brief History of Machine Learning

1940s

McCulloch-Pitts neuron model. Hebb's learning rule.

Early 1950s

Turing's rebutting of Lady Lovelace's Objection suggested that computers could learn for themselves.

Middle 1950s – Middle 60s

Some neural network learning systems were developed, including Perceptron learning rule, Delta rule. There was also some work on symbolic learning systems.

Late 1960s

Eclipse of neural networks as a result of wildly over-optimistic claims about their capabilities and criticisms from Prof. Marvin Minsky at MIT.

Very Brief History of Machine Learning (2)

Early 1970s – Middle 1970s

Very little was done in machine learning – most researchers believed that they must first solve "the knowledge representation problem".

Late 1970s

Machine learning was viewed as potential solution to the "knowledge bottleneck" in expert systems. Decision tree was proposed.

Middle 1980s

Renaissance of neural network approaches with many new models and learning algorithms, including generalised Delta rule, error backpropagation learning algorithm, multilayer perceptron.

Late 1980s – Present

Research in machine learning techniques has been massively expanded, including deep learning in the past decade.

It is machine learning that has made breakthroughs in AI in recent years!

(AlphaGo, Face Recognition, Self-driving Car, Big Data Analysis, …)

Mathematical Preliminaries (mostly for self study)

There are a couple of branches of maths that occur repeatedly in machine learning:

Probability

Is needed because machine learning algorithms usually draw statistical conclusions from evidence.

Logarithm

Is fundamental to information theory that is widely used in machine learning – notably in **decision tree induction (information gain)**.

The next few slides provide a brief reminder of the fundamental ideas of these mathematical topics.

Probability

Probability is likelihood measured on a scale from 0 to 1.

Suppose E is some event

$P(E)$ = 1 means E certainly occurs/has occurred.

$P(E)$ = 0 means E certainly does not occur/has not occurred

$P(E)$ = 0.5 means it is equally likely that E does or does not occur.

If two events, X and Y, are ***statistically independent*** (i.e., neither helps you predict the other) then

$$P(X \wedge Y) = P(X) \times P(Y)$$

Conversely, if

$$P(X \wedge Y) \neq P(X) \times P(Y)$$

This indicates that the knowledge of one event can help you predict the other. This gives rise to the concept of conditional probability.

X^Y: X and Y occur simultaneously.

Conditional Probability

The ***conditional probability*** of event X to occur, given that event Y has occurred, is defined as

$$P(X|Y) \equiv \frac{P(X \wedge Y)}{P(Y)}$$

Do not confuse $P(X|Y)$ with $P(Y|X)$.

The probability that someone known to be an American is the US President is about 3×10^{-9}. (*X* – being the US President, *Y* – being an America (condition))

The probability that the US President is an American is 1.

Note that if X and Y are independent:

$$P(X|Y) = P(X)$$

which is, of course, another way of saying that knowing Y does not help you predict X.

Bayes Theorem

There is a simple relationship between $P(X|Y)$ and $P(Y|X)$:

$$P(X|Y) = P(Y|X) \times \frac{P(X)}{P(Y)}$$

This is known as ***Bayes Theorem***.

It can also be expressed as

$$P(X \wedge Y) = P(X|Y) \times P(Y) = P(Y|X) \times P(X)$$

Bayes theorem forms the basis of an important branch of machine learning, such as naïve Bayes learning and Bayesian networks, but they will not be covered in this module.

Logarithm

The ***logarithm*** of a number x is simply the power *p*, to which a fixed number *b* (the ***base***) must be raised to produce the number. $b^p = x \Leftrightarrow p = \log_b(x)$

Some simple examples

$$8 = 2 \times 2 \times 2 = 2^3$$

So logarithm of 8 to base 2 is 3, which is written $log_2(8) = 3$

$$100000 = 10 \times 10 \times 10 \times 10 \times 10 = 10^5$$

So logarithm of 100000 to base 10 is 5, which is written $log_{10}(100000) = 5$

$$log_2(32) = 5$$

$$log_{10}(100) = 2$$

$$log_2(2) = log_{10}(10) = 1$$

$$log_2(1) = log_{10}(1) = log_x(1) = 0$$

Logarithm (2)

Logarithms to **base 2** are particularly important in computer science.

Here are few more examples:

$$log_2(8) = log_2(2^3) = 3$$

$$log_2(0.5) = log_2\left(\frac{1}{2}\right) = log_2(2^{-1}) = -1$$

$$log_2(0.25) = log_2\left(\frac{1}{4}\right) = log_2\left(\frac{1}{2^2}\right) = log_2(2^{-2}) = -2$$

$$log_2(0.125) = log_2\left(\frac{1}{8}\right) = log_2\left(\frac{1}{2^3}\right) = log_2(2^{-3}) = -3$$

Summary

- **Why Do We Need Machine Learning?**
- **What Is Machine Learning?**
- **Key Elements of Machine Learning**
- **How Does Machine Learning Work?**
- **Learning Viewed as Search for Optimal Models**
- **A Taxonomy of Learning Tasks**

  Learning to classify, Learning to predict numerical values

  Clustering, Reinforcement learning
- **Brief History of Machine Learning**
- **Mathematical Preliminaries (mostly for self study)**

  Probability, Logarithms