Intelligent Systems: Recognition and Reasoning

James L. Crowley

MoSIG M1 Second Semester 2020/2021
Lecture 1 2 Feb 2021

Intelligence: Recognition and Reasoning

Outline

Class notes and exercises on the web:
http://crowley-coutaz.fr/jlc/Courses/2020/MOSIG.SIRR/MoSIG.SIRR.html

The Science of Intelligent Systems.

AI as a Scientific Discipline

Artificial Intelligence is the science and technology of artificial systems that exhibit intelligent behavior.

Science: Science is the elaboration of theories and models that predict and explain phenomena (T Kuhn 76). Science is performed by communities who share paradigms (problems and problem solutions) and compete to publish in scientific journals and conferences.

Scientific method is composed of empirical observation and objective documentation, followed by definition of concepts, and creation of theories and models to explain and predict observed phenomena. The value of a theory or model is in its ability to predict and explain phenomena.

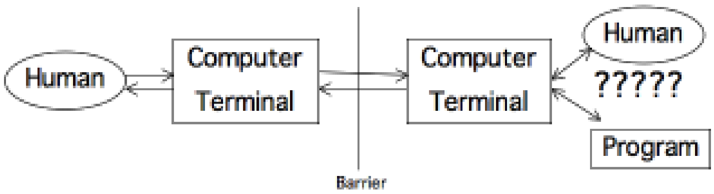
Theories and models are compared by scientific experiments. The model that best predicts the outcome of the experiment, and the theory that provide the simplest explanation, are adopted by the scientific communities.

For sciences of the artificial, such as AI, the theories and models prescribe the design of systems, and explain their performance. For example, in electronics, theories and models explain and predict the relation between electric and magnetic fields and prescribe how to build a motor or a radio. There are generally many possible models for building a system. The key to comparison is **performance evaluation**. Thus, in lecture 2 of this course, we will present techniques for performance evaluation. These are the techniques that the scientific community use to compare systems.

But what do we mean by Intelligence? Alan Turing asked this question in 1950 and defined intelligence as a description of behavior.

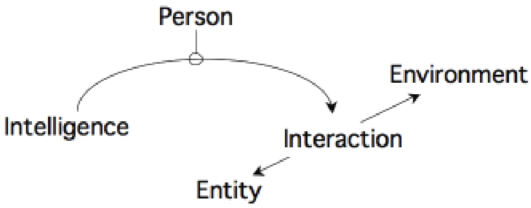
Intelligence as a Description of Behavior: The Turing Test

Alan Turing proposed a definition for intelligence based on a test. The Turing test defined Intelligence as human-level performance at text-based interaction.



To define intelligence, Turing proposed a test in which a human observer interacts with an unknown agent over a teletype terminal. The agent may be a human or a program. Turing proposed that the program (or machine) could be considered as intelligent if a human observer could not reliably tell whether they are interacting with a human or a machine.

The Turing Test gave an important insight: Intelligence is NOT an intrinsic property of an agent. Intelligence is a "DESCRIPTION" of behavior.



Intelligence describes the interaction of an entity with its environment. Intelligence is a description (a property assigned by an observer) of the behaviour of the entity.

In modern form, we can define intelligence as “Human level performance at tasks requiring perception, action, cognition or interaction”.

Intelligence: Human-level performance at tasks requiring perception, action, cognition or interaction

Thus an agent may be very intelligent at some tasks (such as math or chess) and much less intelligent at other tasks (such as social interaction or competitive sports).

Artificial Intelligence as a Scientific Domain

The modern scientific domain of AI emerged in the 1960s as a convergence of researchers from Cognitive Science, Logic, Planning, Pattern Recognition, Image Processing and other fields, driven by the emergence of Computer Science. The origin of the term "Artificial Intelligence" is generally credited to a Symposium at Dartmouth College in 1956, organized by John McCarthy, Marvin Minsky, and attended by the many AI pioneers, such as Arthur Samuel, Herb Simon, Allan Newell, etc.

AI Pioneers at the Dartmouth Symposium (1956)

The field of AI gone through several "epochs", each dominated by different paradigms (problems and problem solutions). Each epoch has left its trace on the technology of Artificial Intelligence.

Dominant Paradigms for Artificial Intelligence:
Pre-1960: Automata and Pattern Recognition
1960-1985: Planning, Problem Solving
1980-1995: Expert Systems
1985 -2000: Logic Programming
1995-2010: Bayesian methods for machine learning
2010 - present: Deep Learning with Neural Networks

Until recently, there were three fundamental barriers to artificial intelligence:

(1) Insufficient Computing Power.
(2) Prohibitive Cost of Encoding Domain Knowledge.
(3) Insufficient Labeled Data for Learning.

In the last 10 years we have seen a revolution. However this revolution has its roots in the beginnings of computer science.

Intelligence as Knowledge and Reasoning.

In the 1970's, AI researchers equated intelligence with knowledge. Some researchers, such as Minsky and McCarthy at MIT argued that intelligence required a powerful reasoning ability, guided by a small amount of knowledge. McCarthy argued that logic was the mathematics of symbolic computing, and that artificial intelligence could be provided by a suitable form of logic programming. This led to the creation of the PROLOG programming environment.

Other researchers such as Ed Feigenbaum and his colleagues at Stanford argued that intelligence could be achieved by symbolically encoding human knowledge in "knowledge bases" that would be used by intelligent agents aided by a small amount of reasoning. AI researchers argued about how Knowledge should be represented as structured knowledge representations and rules expressed in symbolic logic. Reasoning was provided by theorem proving or abductive reasoning.

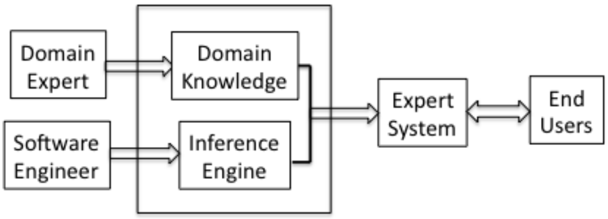
In his 1980 Turing award lecture Alan Newell unified these approaches by defining intelligence as **Competence**: the ability to solve problems.

Newell formulated this as the "Physical Symbol System" hypothesis: Intelligence required symbolic reasoning that could be programmed as a form of knowledge. The dominant paradigm accepted by AI scientists was that intelligence was based on symbol manipulation.

Anything that enabled solution of the problem could be considered to be a form of knowledge. In this view, knowledge could have many forms depending on the requirement of the problem. The challenge was to provide a symbolic encoding.

Expert Systems

In 1980, AI went through a revolution with the invention of Expert Systems. Expert systems use a general purpose "Inference Engine" to interpret symbolic expressions of Expert knowledge (a "Knowledge Base"). Expert systems provide expert advice to users based on the symbolic expression of expert knowledge. Expert systems are constructed by a computer scientist (Knowledge engineer) working with a Domain Expert encode and test the "knowledge base".



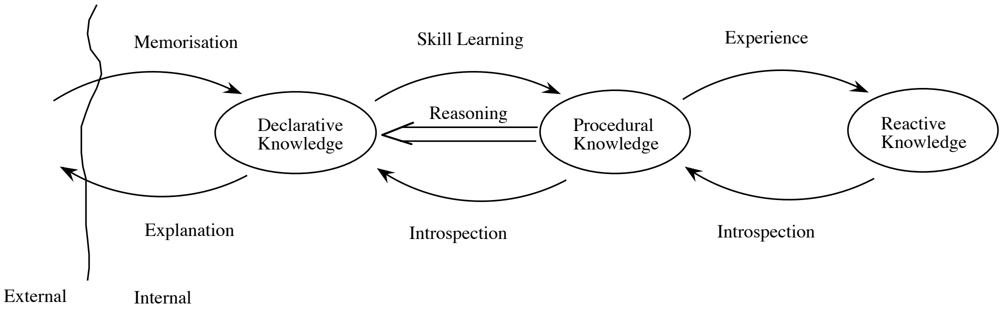
The original expert system was the Mycin Antibiotic Therapy Advisor (1980). The system could interactively provide therapy advice for doctors to prescribe antibiotics. Other successful expert systems found rare-earth mineral deposits, helped configure Digital Equipment co. Vax Computers, and provided logistics planning for NASA. These early systems often provided a 10 to 50 fold return on investment, creating instant wealth for investors.

Expert systems were predicted to be a disruptive technology that would create an economic revolution and dominate all of computer science. This lead to enormous investment and huge conference attendance in the mid to late 1980s. When the prediction failed to come true in the 1990s, funding and conference attendance dropped dramatically and AI was said to have "failed".

However, research continued leading to a gradual convergence with cognitive science, and the emergence of symbolic computing techniques such as the semantic web and cognitive computing.

Kinds of Knowledge

Cognitive Psychologists identify different categories of knowledge. A common model from cognitive science recognizes Declarative, Procedural and Reactive knowledge as providing complementary abilities.



Reactive Knowledge provides fast automatic response to stimulus. Reactive knowledge automatic and enables fast reaction, as is needed for common abilities such as locomotion and manipulation. Riding a bicycle, driving a car, playing a piano or skiing all require reactive knowledge. Reactive knowledge is commonly acquired by experience and practice.

Declarative knowledge is a symbolic expression of competence. Declarative knowledge is commonly acquired using language (spoken or written language) and requires interpretation for use.

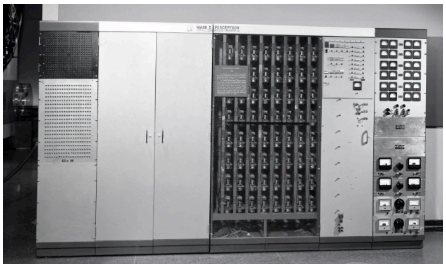
Procedural knowledge provides a series of steps to solve a problem. The procedures are commonly acquired as declarative knowledge but require reactive knowledge for interpretation.

However, a fundamental problem remained: How is knowledge acquired?

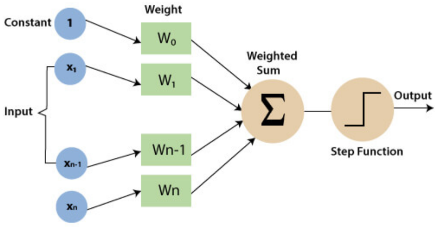
Encoding human knowledge in knowledge bases is prohibitively expensive. Attempts at automatic knowledge acquisition through machine learning were largely unsuccessful.

Machine Learning

One pioneer who was NOT invited to the Dartmouth Symposium was Frank Rosenblatt. In the 1950s, Rosenblatt invented a universal learning machine named the perceptron.



The Perceptron learning machine



The perceptron algorithm

The perceptron was a learning algorithm for a linear decision surfaces, and claimed to be a "universal learning machine".

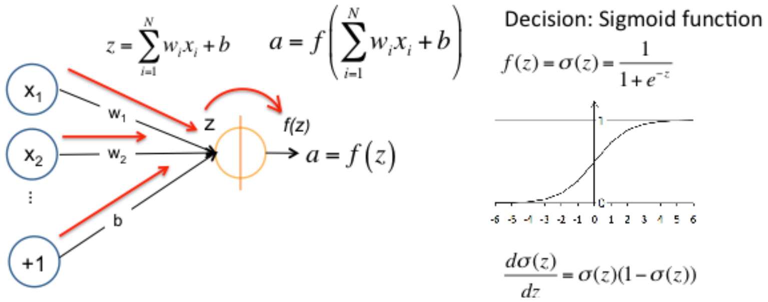
Problems: (1) Learning required labeled training data
(2) The perceptron could only classify linearly separable patterns

If the training data were not linearly separable, the algorithm would not terminate. Furthermore, perceptron learning is a random process, and experimental results were very difficult to reproduce.

In 1969, Marvin Minsky and Seymor Papert published a text entitled "Perceptrons" to document the limitations of the Perceptron. As an example, they showed that a perceptron could not learn to imitate a simple logic gates such as the Exclusive OR. Minsky and Papert book destroyed the reputation of the perceptron learning algorithm for many years. It became impossible to publish papers or obtain funding for research in perceptrons.

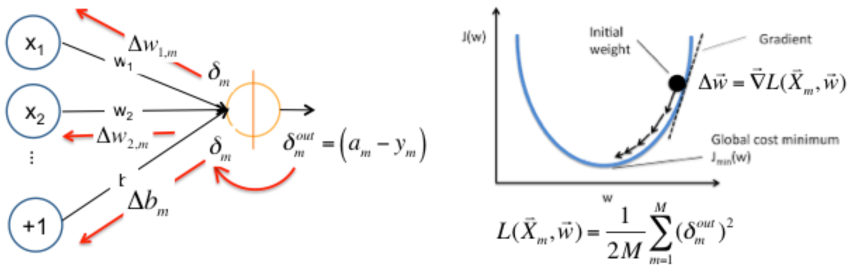
Artificial Neural Networks

During the 1980's, a small group of researchers continued to experiment with perceptrons. They found that problems with training with non-separable data could be overcome by using a soft decision surface, such as a sigmoid.


$z=\sum_{i=1}^{N} w_i x_i + b$
$a=f\left(\sum_{i=1}^{N} w_i x_i + b\right)$
Decision: Sigmoid function
$f(z)=\sigma(z)=\frac{1}{1+e^{-z}}$
x1
x2
+1
w1
w2
b
z
f(z)
$a=f(z)$
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
$\frac{d\sigma(z)}{dz}=\sigma(z)(1-\sigma(z))$

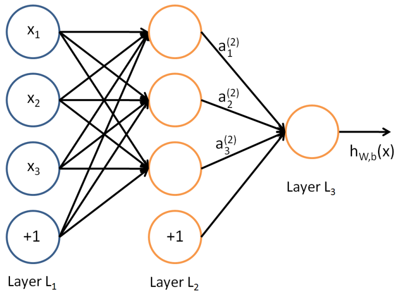
A feed forward neural network using a sigmoid decision functino

They found that the perceptron learning algorithm could be formulated as a form of optimization using gradient descent, and that gradient descent could be reformulated as a distributed parallel algorithm, known as back propagation.


x1
x2
+1
$\Delta w_{1,m}$
w1
w2
$\Delta w_{2,m}$
t
Δb_m
δ_m
δ_m
$\delta_m^{out}=(a_m-y_m)$
J(w)
Initial weight
Gradient
$\Delta\vec{w}=\vec{\nabla}L(\vec{X}_m,\vec{w})$
Global cost minimum
$J_{min}(w)$
w
$L(\vec{X}_m,\vec{w})=\frac{1}{2M}\sum_{m=1}^{M}(\delta_m^{out})^2$

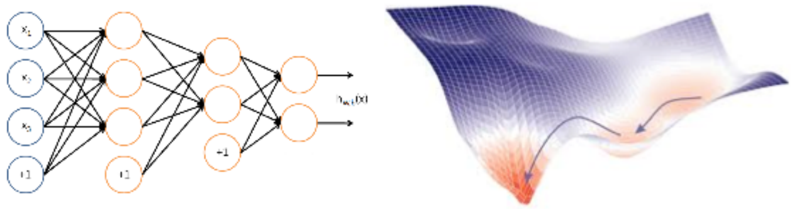
Back-propagation is a parallel form of Gradient Descent.

They found that limitations such as the inability to compute an XOR could be overcome by adding an additional "hidden" layer of neurons.



They were able to demonstrate that artificial neural networks were able to provide solutions to a number of problems in recognition for speech and computer vision that could not be solved by symbolic computing or expert systems. This lead to an second wave of enthusiasm for perceptrons, under the name "artificial neural networks".

Unfortunately attempt to generalize neural networks in the 1980s encountered several barriers.



Noise in the training data leads to local minima in the objective function blocking convergence and leading to non-reproduceable results.

1. Real-world problems required networks with hundreds of thousands (or millions) of parameters to tune, leading to excessively high cost in computation.
2. Training required massive amounts of labeled training data, and the required amount of data grew exponentially with the number of parameters leading to prohibitive cost in data collection and labeling.
3. Real training data was noisy making training very unreliable. System performance was impossible to reproduce.
4. Neural networks are black boxes. No one can explain why they work or when they do not work.

By the mid 1990s, Neural networks were largely abandoned in favor of mathematically sound Bayesian approaches to machine learning and logic programming for artificial intelligence.

Bayesian Approaches for Machine Learning

From the mid 1990s to 2010, machine learning was dominated by methods based on Bayes rule.

Consider two classes of events A and B.

Let P(A) be the probability that an event belongs to class A
Let P(B) be the probability that an event belongs to class B
Let P(A, B) be the probability that the event is in both A and B.

Bayes rule tells us that conditional probability is the fraction of events that are B that are also A and B: $P(A \mid B) \equiv \dfrac{P(A,B)}{P(B)}$

From this we can write: $P(A \mid B)P(B) = P(A,B)$

Similarly $P(B \mid A) \equiv \dfrac{P(A,B)}{P(A)}$ so that $P(B \mid A)P(A) = P(A,B)$

This gives the common definition of Bayes Rule:

$$\boxed{P(A \mid B)P(B) = P(A,B) = P(B \mid A)P(A)}$$

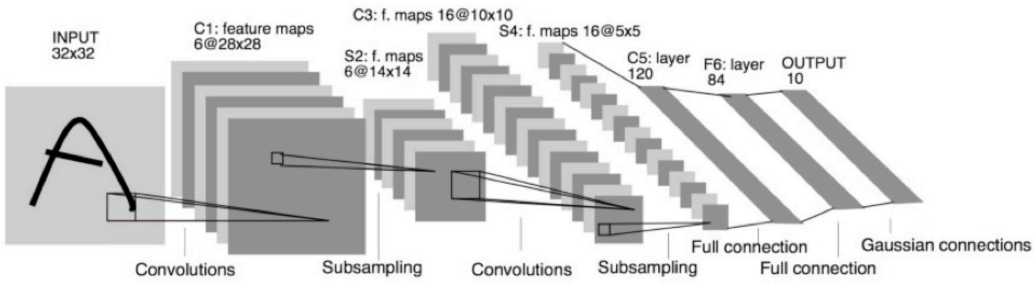
We can use this to construct systems that learn to assign events to classes with the lowest possible probability of error.

The Return of the Perceptron

In the late 1990s, a new form of machine learning became popular: Support Vector Machines (SVMs). SVMs are a form of maximal margin linear classifier, learned from data, similar to a perceptron. SVMs showed that machine learning approaches based on optimization using large amounts of data and computing could outperform elegant hand-crafted algorithms using logic programming or Bayes Rule. From 2000 to 2010, support vector machines were the dominant technique for learning for perception and recognition.

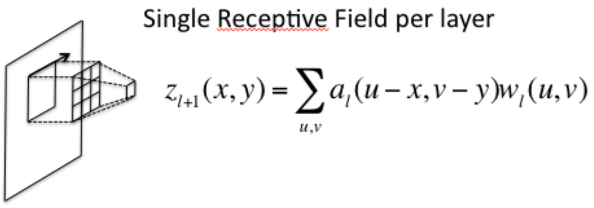
Convolutional Neural Networks

Although Neural Networks were largely abandoned in the 1990s, a small group of researchers continued to experiment. Using ideas from computer vision, Yann LeCun build a series of Convolutional Neural Networks. In 1994, one of his networks, LeNet5 won a competition for the best technique to recognize hand written characters on checks. This led to a commercial system for processing checks and sorting mail.

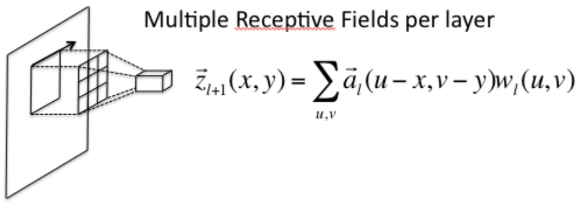


The LeNet 5 convolutional neural network - for recognizing hand-written digits.

Convolutional networks learn a 2-D pattern of weights, called a receptive field.



$$z_{l+1}(x,y) = \sum_{u,v} a_l(u-x, v-y) w_l(u,v)$$

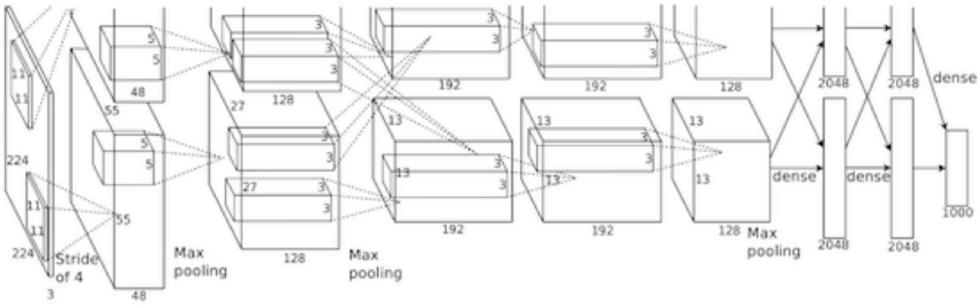


$$\vec{z}_{l+1}(x,y) = \sum_{u,v} \vec{a}_l(u-x, v-y) w_l(u,v)$$

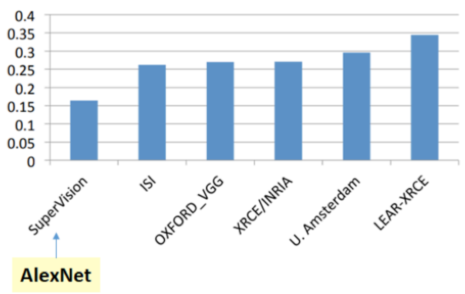
LeNet used back-propagation to learn several small receptive fields at each level.

In the early 2000's Computer Vision was increasingly driven by Challenge Based Research, in which researchers publish a data set and a base line algorithm and challenge the community to provide solutions with better performance.

The ImageNet Large Scale Visual Recognition Challenge was published in 2010. In 2012, the challenge was won by a network created by Alex Krizhevsky and Geoff Hinton based on the LeNet convolutional network. They called their network AlexNet.



AlexNet won by a large margin with an error of around 15%



This triggered a paradigm shift for Computer Vision, Speech Recognition, Machine Learning and (more recently) Artificial Intelligence.

So what changed between 1990 and 2010?

1. **Computing power**: Since 1948, as predicted by Moore's Law, computing power has doubled every 18 months, driven by the reduction in integrated circuit feature sizes (currently approx 10^{-5} meters (10 microns). In the 30 years since 1990, this has lead to a multiplication by 2^{20} in available computing power (aprox million fold growth!) With abundant GPUs and Cloud-based massively parallel Grid Computing massive parallel computing power is now available.

2. **Data**: The internet and the World Wide Web provide access to planetary scale data and the massive sharing of labeled training data. This is currently amplified by IOT and mobile computing.

3. **Reliability**: Improved understanding of optimization and learning algorithms have led to improved reliability and reproduceability of research results.

We will study neural networks in lectures 7 through 12. You will do a programming exercise in March, using Keras and Pytorch to recreate a network that learns to recognize handwritten characters using the LeNet 5 architecture.

Since 2010 a major break through has occurred with the development of techniques based on Deep Learning. Deep Learning has been found to provide reliable solutions to longstanding problems perception and problem solving.

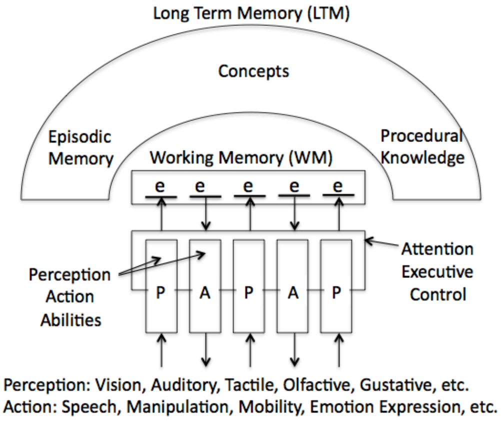
However, deep learning is only part of the solution. For human level performance at tasks requiring natural language requires explicit representation of knowledge. Natural language understanding offers access to the immense store of human knowledge coded in books and the internet, and to learn from and eventually teach humans. For this we need a convergence of machine learning and symbolic computing to provide cognitive systems.

Artificial Intelligence and Cognitive Science

Long Term Memory, Working Memory and Cognition

Modern cognitive scientists study intelligence as interaction of different forms of memories. Most models posit a cognitive architecture composed of a limited working memory interacting with perception and action based on episodic, procedural and conceptual memories.

A common cognitive model used for study of comprehension centers reasoning on processing places a limited working memory as the interface between perception, action and long term mermory.


Long Term Memory (LTM)
Concepts
Episodic Memory
Procedural Knowledge
Working Memory (WM)
e e e e e
Perception Action Abilities
P A P A P
Attention Executive Control
Perception: Vision, Auditory, Tactile, Olfactive, Gustative, etc.
Action: Speech, Manipulation, Mobility, Emotion Expression, etc.

(inspired by Rasmussen, Card-Moran-Newell, Anderson, Kintsch, many others)

Most models of human cognitive models share a number of common elements:

- Perception: Transforms and combines sensory stimuli to Phenomena
- Short Term Perceptual Memory: Temporary buffer holding recent stimuli
- Action: Activation patterns for muscle groups.
- Working Memory: 7+/-2 memory slots (perceived or remembered)
- Long Term Memory

Long-term memory (LTM) refers to memory structures used in several different cognitive abilities:

- Episodic Memories: recordings of significant sensory experiences
- Semantic Memory: Abstract representations for sensory experiences
- Procedural Memory: Sequences of operations to accomplish goals
- Spatial memory (Spatial relations between places)

Modern machine learning explore similar structures based on auto-encoders and self attention. Such architectures are currently to understand and manipulate natural language with systems as Google's Bidirection Encoding with Tranformers (BERT) and Open-AIs GPT-3. Such system are capable of self supervised learning by reading anything available on the internet.

Course Overview

This course is organised in two parts. We will first concentrate on machine learning for recognition. We will then cover techniques for reasoning with relations and knowledge.

Part 1 – Recognition and Machine Learning

1) Supervised learning and Performance Evaluation
2) Bayesian Learning, non-parametric methods.
3) Non-supervised learning with EM and K-Means
4) Support Vector Machines
5) Artificial Neural Networks, Back Propagation, and Architectures.

Part 2 – Reasoning

1) Knowledge Based Systems
2) Schema Systems, Frames, Structured Knowledge
3) Temporal and Spatial Reasoning
4) Planning and problem solving
5) Narrative and Causal Reasoning

We will start with a review Bayesian methods for machine learning in lectures 3, 4 and 5. In lecture 3 we will review Bayes Rule and probability theory. In lecture 4 we will see that a simple "frequency based" or "statistical" interpretation of Bayes rule can lead to some very practical algorithms for building non-parametric systems for learning and classification. In lecture 5 we will show how an axiomatic approach to probability can lead to powerful non-supervised methods for learning, clustering and data mining such as the algorithms K-means and Expectation-Maximization followed by methods based on Support Vector machines in lecture 6.

K-means, Expectation-Maximization and Support Vector Machines are all based on iterative algorithms for optimisation. In lesson seven we will return to the perceptron and show a universal optimization approach called Gradient Descent.

In lesson 8 we will see that a distributed form of Gradient Descent called "back-propagation". We will see how back propagation can be used to learn artificial neural networks. We will look see that such networks can be used both to recognize signals and to generate signals, and see that generative and discriminative networks can be combined to provide a powerful learning technology. We will study architectures for convolutional neural networks.

Much of the second half of the class will be devoted to symbolic reasoning and cognitive systems. We will review expert systems in lecture 13. We will see that schema systems can be used to represent both observed and abstract phenomena (concepts) and that these can be attached to methods for reasoning with Frames (lectures 15 and 16). We will see that relations between phenomena are fundamental to reasoning and study reasoning with spatial and temporal relations (lectures 13 and 17). Relations between phenomena are used to define a state based model and are fundamental both for planning and for Narrative and causal reasoning.

Grades

Unless the COVID pandemic forces a change, grades will be based on a neural network programming project (10%), and a final exam (90%). In the programming exercise, you will be asked to build and evaluate neural networks to recognize hand-written digits using python and keras. The emphasis for this exercise is on experimental performance evaluation.

The exam will be composed of a series of exercise problems based on the course material. Each week the course material will be illustrated with a set of exercise problems. These exercises will NOT be graded. However, the final exam will be (mostly) composed of (modified) versions of these same exercises! Do the exercises and the exam will be easy. Ignore the exercises and the exam will be challenging.

Exercises may be done individually or in a group. Exercises should be done within 2 weeks of assignment. Completed should be sent to me by email as a .pdf including the names of all persons who contributed to the solution. It is acceptable to work the exercise on paper and send a photo by email. The photo must be easily readable.

Feedback will be returned by email. Please allow at least 2 weeks for feedback.