CS1003/1004: Intro to CS, Spring 2004

Lecture #13: Networks, AI

Janak J Parekh
janak@cs.columbia.edu

Administrivia

- HW#5 due today
- HW#6 out tonight
  - Homework topics feedback?
  - It's not always easy coming up with "interesting stuff" that isn't very hard.
- Maryam will be out *next* week
  - William will be teaching her lectures
  - There may be some OH rescheduling, so be sure to check the webboard
- Grades
- Review session?

IP addressing

- IPv4: "dotted-quad notation"
  - Each machine has an address of the form xxx.yyy.zzz.www
  - Many "restricted" addresses
  - DNS (domain name service) maps a name to an IP address
  - chambers.psl.cs.columbia.edu → 128.59.14.155
- LANs typically have contiguous IP addresses
  - Columbia (wired): 128.59.*.*
  - Columbia (wireless): 160.39.*.*
  - We're getting slowly more fragmented
- *Routers* "route" packets between one LAN to another based on addresses and a "routing table"

IP "packets"

- A *packet* is a bag of data, typically up to 1500 bytes
- Contains some *headers* specifying things like source and destination, and some *data*
- The Internet is a "packet-switched" network
- TCP (Transmission Control Protocol) is one protocol that takes large amount of data to be sent and breaks them up into these small packets
- TCP/IP – the most common combination (RFC 793)
- I can take a look at the packets if I'm bored…

What services run on the Internet?

- E-mail: specified by its own protocols
  - SMTP (RFC 821, 2821) – Specifies how to transfer email from a source to a destination via a chain of mail servers
  - POP3/IMAP are simply *retrieval* protocols to retrieve your mail from a mailbox
- Web: two main standards
  - HTTP: Hypertext Transfer Protocol (RFC 2616)
  - HTML: Hypertext Markup Language
- Both work over TCP/IP
  - "Stacking" protocols on top of each other
  - *Port* abstraction to separate services over TCP/IP

Other services

- Telnet: simple text over TCP/IP
  - In fact, I can telnet to an HTTP server and talk HTTP or SMTP if I know how to
- FTP: File Transfer Protocol
- ssh: like telnet, but encrypted for security's sake
  - I can actually read the data typed over telnet or ftp using tcpdump… if I'm root or have control over a switch
- Others?
  - kazaa, IRC, AIM, MSN, you name it
  - Worms
  - Once you learn more, you can make your own

So how do you stay secure?

- Effective password management
  - Change your passwords every so often
  - Don't use your last name as the password
- Use secure protocols
  - These use *encryption*, which makes it difficult for a third-party
  - SSL, ssh are two of several out there
- Don't run random programs on your computer
  - Viruses and spyware can do network traffic communication behind your back, and convey your own data to other parties

What does this mean for you?

- OSes and networks are the context of all the work we do with computers nowadays
- If you program in the future, you'll likely have to interact with both in a more involved form
- Both C and Java have ways of communicating with the operating system and with other computers on LANs and the Internet, so you can write your own Kazaa's or webbrowsers…

Transition…

- We've already talked about…
  - Hardware basics
  - Software basics
  - Systems and networks
  - How to build solutions from these (albeit simple)
- This and the next lecture talk about more open-ended areas of Computer Science
  - But still very legitimate!

Artificial Intelligence

- Perhaps one of the most misunderstood Computer Science concepts
- "… to develop machines that communicate with their environments through traditionally human sensors means and proceed intelligently without human intervention."
- In other words:
  - Algorithms to understand human communication
  - Algorithms to process information unattended
- Once something "works", it's no longer "AI"
  - Voice recognition is here, and it works (mostly)

What's an AI?

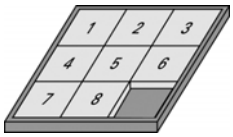
- In order to accomplish the task, do we just use a clever combination standard computing algorithms (performance), or do we actually try to "model" the mind (simulation)?
- Is intelligence measured by the ability to win (at a game) or to be humanlike?
  - Turing test
  - Turing supposed that by the year 2000, machines would have a 30% chance of passing a 5-minute Turing test
  - DOCTOR/ELIZA: free copy in emacs!

Various AI methodologies

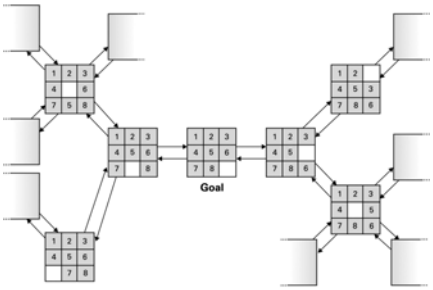
- Reasoning/production systems
- Neural networks
- Genetic algorithms
- Natural language processing
- Robotics, vision
- Databases/expert systems

Reasoning

- Common problem domain – the 8-puzzle
- There are 181,440 different configurations of the 8-puzzle
- Given a random configuration, can we compute the moves necessary to restore to this state?



A large search problem



"Production system"

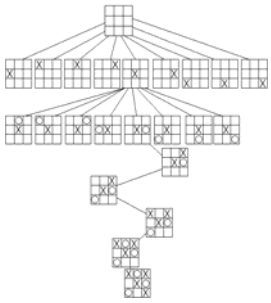
- Consists of three things:
  1. A number of *states*
  2. A number of *productions* or *rules* to transition between states
  3. A *control system* to decide which rule to follow
- Given these elements, the problem reduces to a *search problem*
- One way of modeling this is a *search tree*, consisting of part of the state graph

Search tree for 9-puzzle

- This is just a partial search tree
- Represents one initial configuration
- Goal: to traverse the tree quickly enough and find the correct state
- Problem: tree can be very "wide"

Search tree for Tic-Tac-Toe

- Again, partial search tree
- User might be the first move, followed by a computer move, etc.
- Goal: find a *winning* state
- Problem reduced to a data structure and a set of *search algorithms*
- Still many choices…

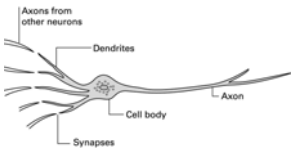


Search strategies

- *Breadth-first*
  - Look at the first row, then the second row, then the third row…
- *Depth-first*
  - Go all the way to one leaf, then backtrack and resume
- *Heuristic*
  - Have a special piece of code that "tells" you a preferred choice
  - A directed search – not always foolproof, but reduces amount of nodes searched
  - For 8-puzzle: "# of tiles *out of place*" – take move that minimizes this value

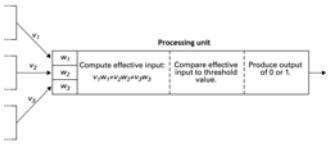
Neural networks

- Idea modeled after neurons
- Given some inputs and a configuration, the neuron *fires* with the appropriate stimuli
- Neurons may "learn" which stimuli to fire on



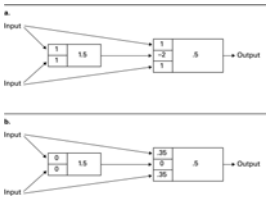
Artificial neural networks

- Difference: we use numbers, not electrical impulses
- "Compute effective unit" uses weights w_x
- Goal: arrange a network of these that produces the result that we want, and adjust the weights so it gives the correct answer



Artificial neural networks (II)

- Challenge: Given such networks, we don't want to adjust the weights manually
- A technique called *backpropagation* allows the machine to be given "training data", and it adjusts its weights to match the desired output
- Example: face, voice recognition



Genetic Algorithms

- Have programs *evolve*; mix-and-match them to produce the best result
  - Common in building game players: mix-and-match players to produce desirable output
- Need a very focused language that you can "mix-and-match"
- Generally a very slow process to evolve

Natural Language Processing

- Syntactic analysis
  - Apply grammar rules
  - For example, identify the subject of the sentence "Mary gave John a birthday card."
- Semantic analysis
  - Identify the semantic role of each word, i.e., action, agent of action, object of action
- Contextual analysis
  - "I ate a bag of chips."
- Applications
  - Information retrieval and information extraction
  - Particularly important for web-based applications

Robotics/vision

- Historically focused on mechanical and electrical engineering aspects
- We can already do set tasks, but what about modifications?
  - Objects on a conveyor belt at irregular intervals/orientation
  - Navigate around a room with obstructions
- Need to take images of scenes, compute boundaries, detemine paths
- Goal: autonomous robots

Database/expert systems

- Context drives a huge problem: how to encode context and knowledge that the human mind possesses, and retrieve said information?
- "Associative memory systems"
- Web search is just a start – keyword-based searching, not semantic-based searching
- Expert systems: encode domain-specific knowledge to help solve problems

Weak vs. Strong AI

- All of these applications are essentially *weak*: we tell the computer what to do, and we solve problems
  - Not really "AI", per se – useful solutions to solve real-world problems
- *Is* Strong AI, i.e., sentience/consciousness, possible?
  - If so, we're still quite a long way away
  - On the other hand, there's the Turing test

Next time

- In labs:
  - C – File I/O
  - Java – GUI-based event programming
- Last lecture: computation theory