Lecture 23: Artificial intelligence, machine learning, natural language processing, ...

- **buzzwords, hype, real accomplishments, wishful thinking**
  - big data, deep learning, neural networks, chatbots, ...
- **brief history**
- **examples**
  - games (chess, Go)
  - classification (spam detection)
  - prediction (future prices)
  - recommendation systems (Netflix, Amazon, Goodreads, ...)
  - natural language processing (sentiment analysis, translation, generation)
  - large language models
- **issues and concerns**
  - accuracy
  - fairness, bias, accountability, explainability
  - appropriate uses
- **Beware: on this topic, I am even less of an expert than normal.**

Revisionist history of AI (non-expert perspective)

- **1950s, 1960s: naive optimism about artificial intelligence**
  - checkers, chess, machine translation, theorem proving, speech recognition, image recognition, vision, ...
  - almost everything proved to be much harder than was thought
- **1980s, 1990s: expert or rule-based systems**
  - domain experts create rules, computers apply them to make decisions
  - it's too hard to collect the rules, and there are too many exceptions
  - doesn't scale to large datasets or new problem domains
- **2010s: machine learning, big data, deep learning, ...**
  - provide a "training set" with lots of examples correctly characterized
  - define "features" that might be relevant, or let the program find them itself
  - write a program that "learns" from its successes and failures on the training data (basically by figuring out how to combine feature values)
- **2020s: large language models**
  - ChatGPT, Claude, ...
  - near-human performance on many text understanding and generation tasks

    including images, speech (multi-modal)

Examples of ML applications (a small subset)

- **games**
  - checkers, chess, Go
- **classification**
  - spam detection, digit recognition, optical character recognition, authorship, ...
  - image recognition, face recognition, ...
- **prediction**
  - house prices, stock prices, credit scoring, resume screening, ...
  - tumor probabilities, intensive care outcomes, protein structure, ...
- **recommendation systems**
  - e.g., Netflix, Amazon, Goodreads, ...
- **natural language processing (NLP)**
  - language translation
  - text to speech; speech to text
  - sentiment analysis
  - text generation (ChatGPT et al)
  - image generation (Dall-E, Stable Diffusion, Midjourney, etc)

Types of learning algorithms

- **supervised learning (labeled data)**
  - teach the computer how to do something with training examples
  - then let it use its new-found knowledge to do it on new examples
- **unsupervised learning (unlabeled data)**
  - let the computer learn how to do something without training data
  - use this to find structure and patterns in data
- **reinforcement learning**
  - some kind of "real world" system to interact with
  - feedback on success or failure guides / teaches future behavior
- **recommender systems**
  - look for similarities in likes and dislikes / behaviors / ...
  - use that to predict future likes / behaviors

Classification example: spam detection

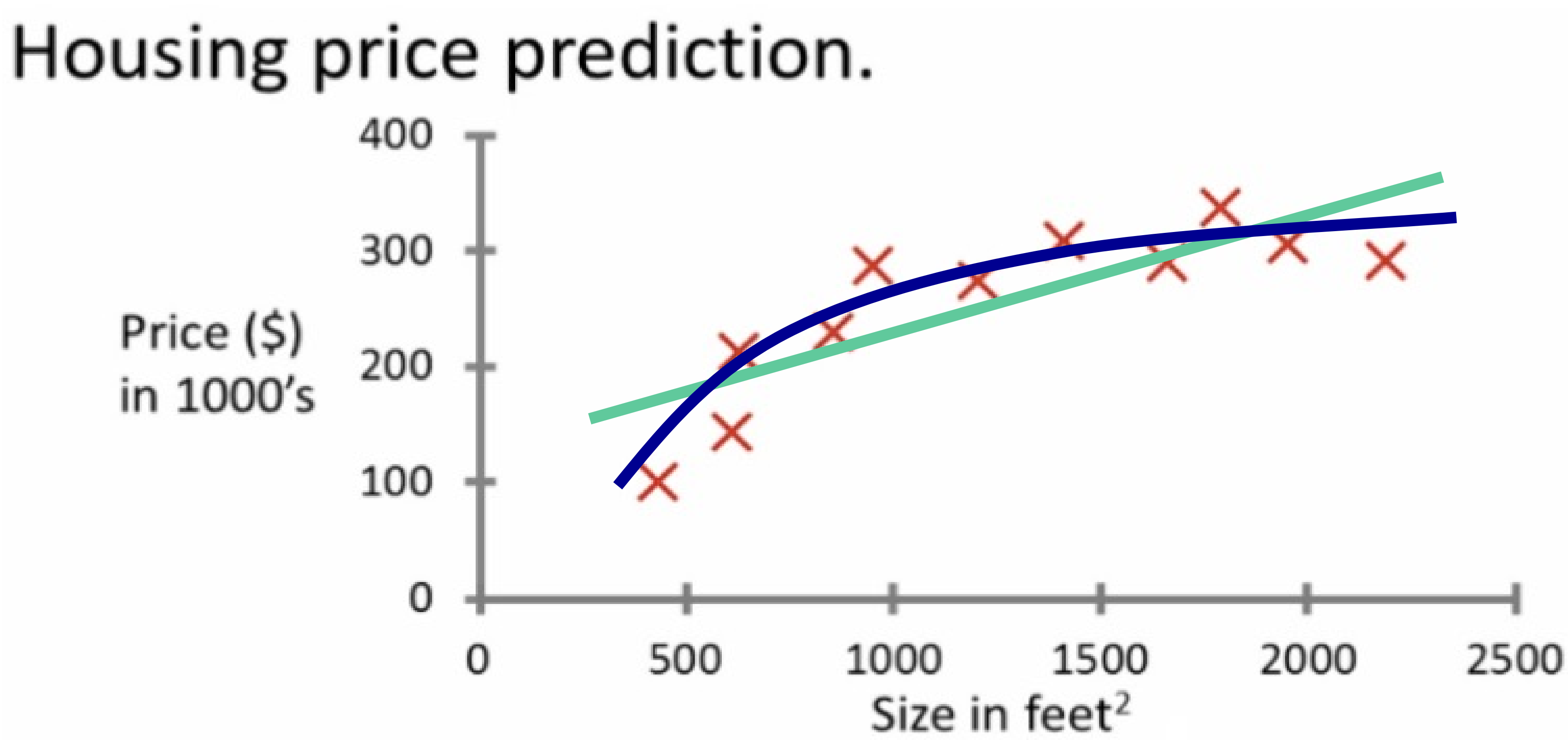
- **rule-based machine learning: choose a set of features like**
  - odd spelling, weird characters, language and grammar, origin, length, ...
- **provide a training set of messages marked as "spam" or "not spam"**

- **ML algorithm figures out parameter settings that let it do the best job of separating spam from not spam in the training set**
- **then apply that to real data**

- **potential problems:**
  - training set isn't good enough or big enough
  - creating it is may have to be done manually
  - "over-fitting": does a great job on training set but not on real data
  - spammers keep adapting so we always need new training material

Prediction example: house prices

- **only one feature here: square footage**
- **straight line? ("linear regression")**
- **some kind of curve?**



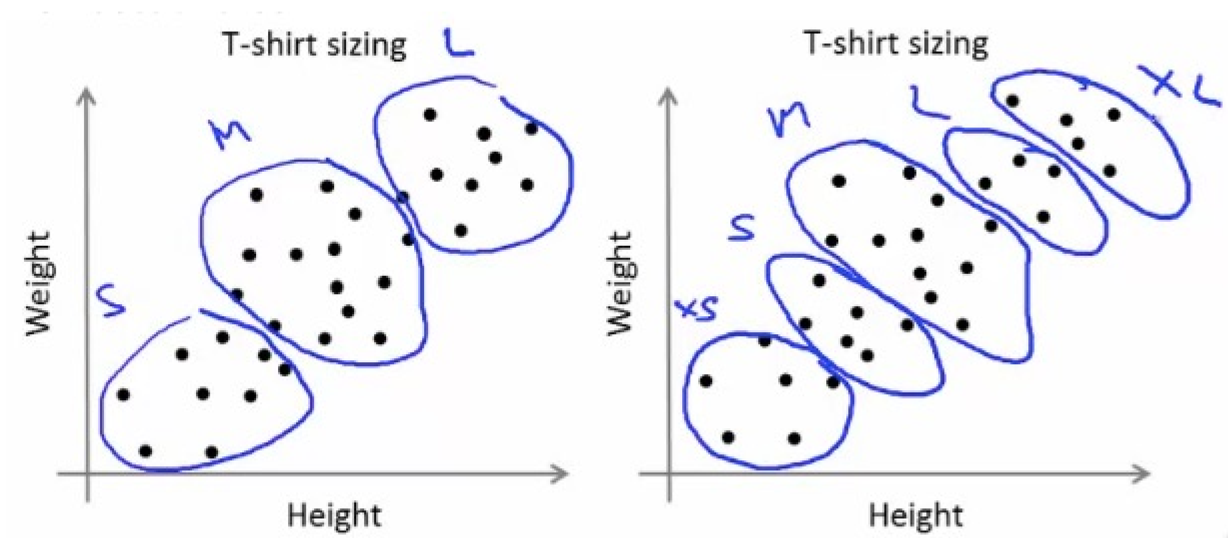
Clustering: learning from unlabeled data

- **contrast with supervised learning**
  - supervised learning:

    given a set of labels, fit a hypothesis to it
  - unsupervised learning:

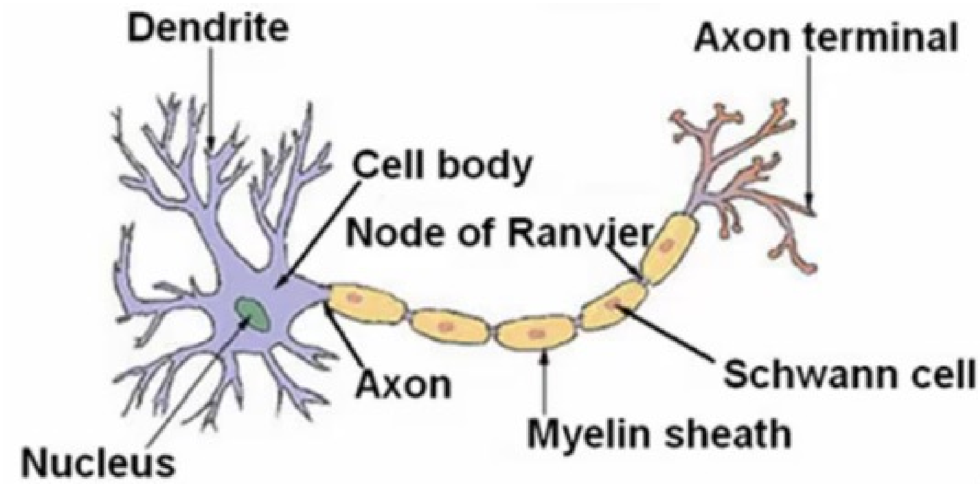
    try and determine structure in the data

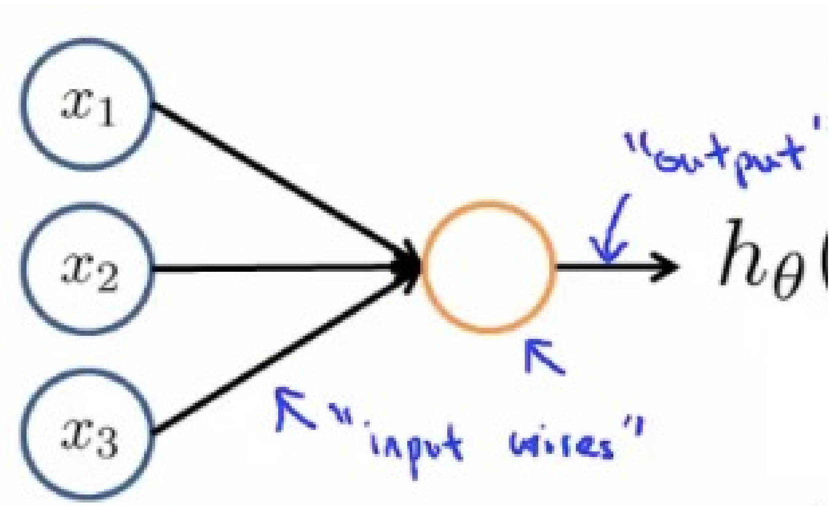
    clustering algorithm groups data together based on data features
- **clustering is good for**
  - market segmentation – group customers into different market segments
  - social network analysis – identify friend groups
  - topic analysis
  - authorship

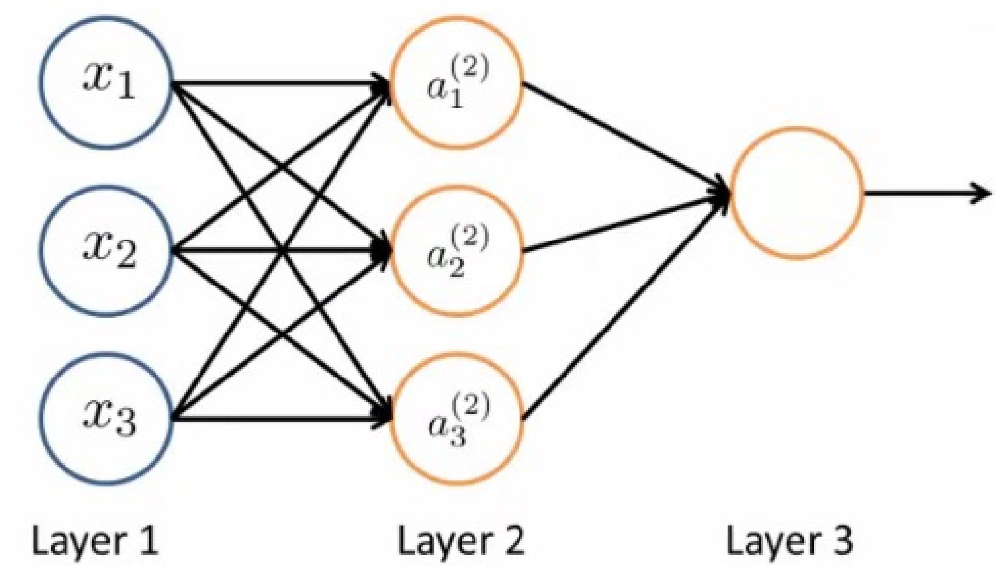


Neural networks, deep learning

- **simulate human brain structure**
  with artificial neurons
  in simple connection patterns







Large Language Models (LLM)

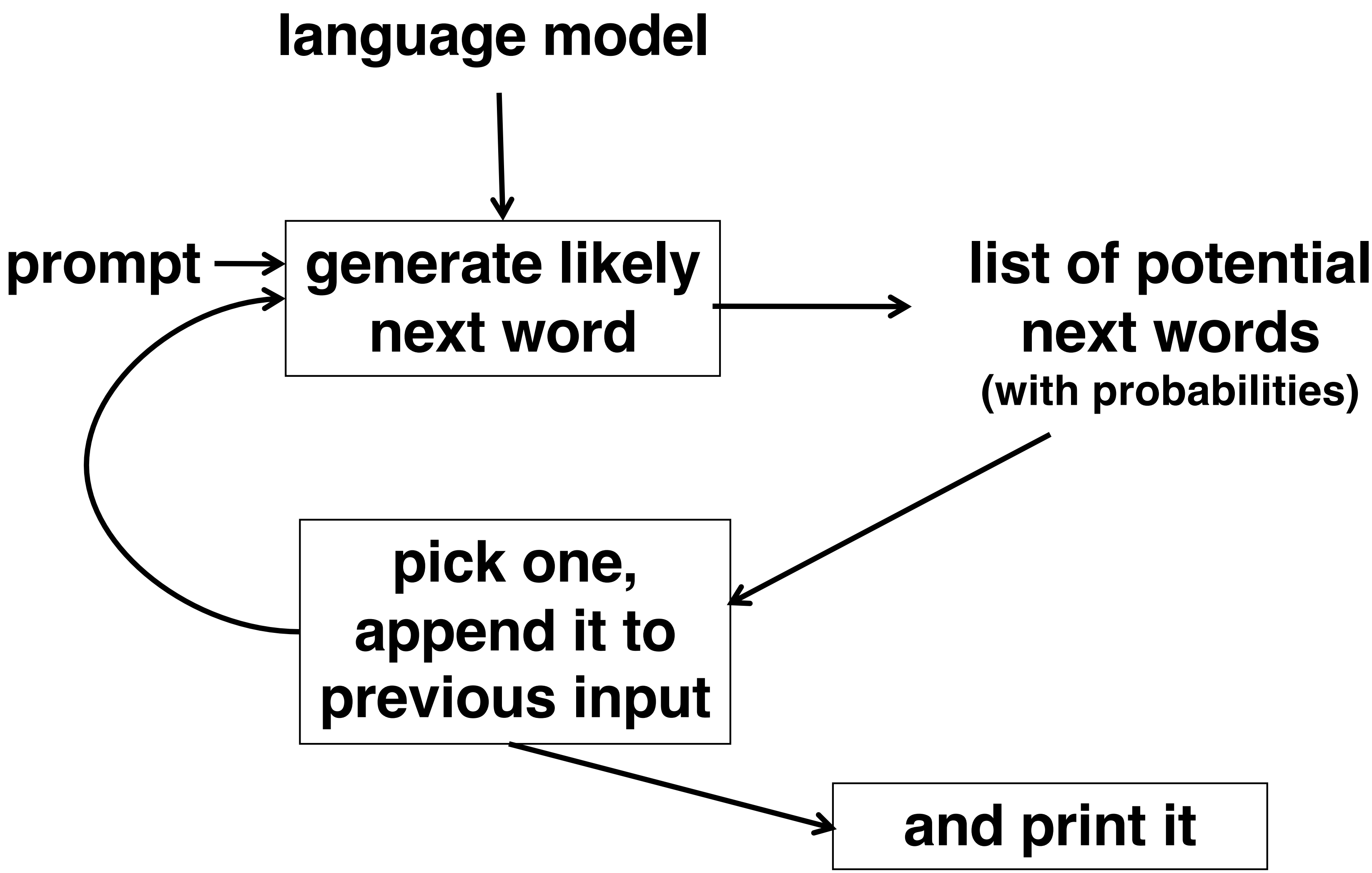
- **language models based on very large text corpus**
  - use deep learning to learn how language is used
  - use that to generate text that seems human-written
  - and give the (strong) impression of understanding

- **models are proprietary (mostly)**
  - e.g., GPT-3, -4 licensed by Microsoft from OpenAI
  - in part because they cost a *lot* to create, plus competitive value

- **GPT = Generative Pre-trained Transformer**
  - transformer is a particular architecture for training

- **ChatGPT is based on GPT-3, -4   (chat.openai.com)**
  - tuned for conversational style
  - can remember previous parts of a conversation
  - very new: became available Nov 30, 2022
  - has already revolutionized the field and public perception of AI

How LLMs work (layman's view)



Training a language model (layman's view)

- **read an enormous amount of text (trillions of words)**
- **learn how words fit together, what phrases make sense, what information is likely true (because it appears often?)**

- **training comes from massive amounts of text (no labeling)**
  - e.g., wikipedia, gutenberg, newspapers, programs, ...
  - in any language
  - GPT-3 training corpus approximately 500 billion words

- **this creates billions of parameters (numbers) that enable prediction of next word given a sequence of words**

- **doesn't really "understand" but is very good at figuring out context well enough that its predictions make sense**

Training and using a model

- **train model on say 1 trillion words from Internet, etc.**
- **language model is a very deep and wide neural net**
  - e.g., 10,000 wide, 100 deep, lots of internal connections
  - 100 billion or more parameters

- **training: give it some text, ask it to predict the next word**
- **if it's wrong, adjust model parameters so it will do better**
- **repeat for several months of computation**

- **in use, give it a prompt, ask it to predict next word**
- **it generates a list of potential next words with probabilities**
- **pick one of the more likely continuations, randomly**
- **print it**
- **feed the result back into the input**
- **repeat**

Hardware

- **logical/functional/architectural structure**
  - bus connects processor, primary memory, disks, other devices
  - caching
  - CPU cycle: fetch-decode-execute; kinds of instructions
    - toy machine as an example
    - different processor families are incompatible at the instruction level
  - von Neumann: architecture; Turing: equivalence of all machines
- **physical implementation; sizes and capacities**
  - chips; Moore's law, exponential growth
- **analog vs digital**
- **representation of information**
  - bits, bytes, numbers, characters, instructions
  - powers of 2; binary and hexadecimal numbers
  - interpretation determined by context
- **it's all bits at the bottom**

Software

- **algorithms: sequence of defined steps that eventually stops**
  - complexity: how number of steps is related to amount of data

    linear: searching, counting, …

    quadratic: simple sorting

    logarithmic: binary search (logarithm = number of bits needed to store)

    n log n: quicksort

    exponential: towers of Hanoi, traveling salesman problem, …
- **programs and programming languages:**
  - evolution, language levels: machine, assembly, higher-level
  - translation/compilation; interpretation
  - a program can simulate a machine or another program
- **basic programming, enough to figure out what some code is doing**
  - variables, constants, expressions, statements, loops & branches (if-else, while), functions, libraries, components
- **operating systems: run programs, manage file system & devices**
  - file systems: logical: directories and files; physical: disk blocks
- **application programs, interfaces to operating system, APIs**

Communications, etc.

- **local area networks, Ethernet, wireless, broadcast media**
- **Internet: IP addresses, names & DNS, routing; packets**
  - bandwidth
- **protocols: IP, TCP, higher-level; layering**
  - synthesis of reliable services out of unreliable ones
- **Web: URLs, HTTP, HTML, browser**
  - caching
- **security & privacy: viruses, cookies, spyware, …**
  - active content: Javascript, plugins, addons
- **cryptography**
  - secret key; public key; digital signatures; secure hashes
- **compression; error detection & correction**
- **wireless, cell phones, GPS, …**
- **AI/ML**

Real world issues

- **legal**
  - intellectual property: trademarks, patents, copyrights, licenses
  - jurisdiction, especially international
- **social**
  - privacy, security
- **economic**
  - open source vs proprietary
  - who owns what
- **political**
  - policy issues
  - balancing individual, commercial and societal rights and concerns

Things to take away

- **some skills, some specific technical knowledge**
  - how computers and communications work today
  - what's ephemeral, what's likely to still be true in the future
- **improved numeracy / quantitative reasoning**
  - what makes sense, what can't possibly make sense, and why
    plausible estimates, engineering judgment, enlightened skepticism
- **another way of thinking**
  - how do things work?
  - how *might* something work?
  - you can often figure it out
- **some appreciation of tradeoffs & alternatives**
  - you never get something for nothing
- **some historical perspective**
  - everything derives from what came before

- **informed opinions about the role of technology**

Final exam (watch the web page for updates)

- **Exam will be emailed to you early on Saturday Dec 14**
  - must be returned by Thu Dec 19, 5 PM EST in person / email / pony express
- **similar to midterm but twice as long**
- **open book, as with midterm:**

  open notes, book, problem sets, labs, old exams, ..., but no Internet
- **see instructions on web site**

- **I'm usually looking for something <u>brief</u> that shows that you understand or can reason**
- **if you're writing or calculating a lot, you're likely on the wrong track**
- **questions try to test your understanding of basic ideas**
  - meant to be simple and straightforward, <u>if you understand</u>
  - not meant to be tricky or rely on obscure facts
- **think about plausibility and where I'm likely coming from**
- **if it still seems ambiguous, say "I'm assuming this..." and carry on**