Class Eleven: From Classical to Connectionist AI

Philosophy and Science Fiction - Ryan Simonelli

November 2, 2022

1 Classical AI

- **Symbol Manipulation:** What an computer program does is follow a definite algorithm (constructed by its human creators) by which it takes one sequence of symbols as an input and returns another sequence of symbols as an output. Recall the examples:
  - Long division—quite clearly a case of symbol manipulation.
  - Our toy conversational program, which is clearly just manipulating strings of symbols.
- **Three Main Worries:**
  - **Searle:** It seems like something could produce the right outputs, following the algorithm for producing outputs from inputs, without actually understanding either the inputs or the outputs.
  - **Dreyfus:** It's not clear how such an algorithmic approach could account for various aspects of our intelligence that seem crucial to our consciousness, particularly, our ability to interact with our environment in a skillful way.
  - **Churchlands:** This sort of classical AI doesn't at all replicate the sort of processes that actually go on in the brain that we know give rise to conscious intelligence.

2 Searle's "Chinese Room" Argument

- **Weak vs. Strong AI:** Searle makes the following helpful distinction:
  - **Weak AI:** Studying AI can be a useful tool to understanding the mind.
    - Searle has no objection.
  - **Strong AI:** Computers can genuinely *understand* things and have other conscious mental states.
    - Searle's main target.
  - **The Chinese Room:**
    - You're locked in a room, with a whole bunch of Chinese symbols.
    - You don't know any Chinese.
    - You're given a series of Chinese symbols through a slot in the room.
    - You have a set of instructions, written in English, that tells you, when given one series of symbols through the slot, which series of symbols to put through the slot yourself.
    - If a speaker of Chinese were to witness the exchanges, seeing just the symbols that go in and the symbols that go out, it would seem like the person in the room is engaging in a fluent conversation.
    - Clearly, however, you're just blindly following instruction for manipulating symbols; you don't actually understand Chinese.
  - **The Point:** We can imagine that the program implemented by such a "human computer" (recall Turing's appeal to this use of the term) passes the Turing Test. But we would not inclined to say that human computer understands language. Why should we say anything about a digital computer?

- **The Churchlands' Basic Response:** This argument begs the question against classical AI, assuming the negation of the very thing that classical AI is committed to: that symbol manipulation (syntax) can be sufficient for understanding of meanings (semantics). Still, classical AI is wrong for other reasons . . .

3 Dreyfus's "Embodied Coping" Argument

- Much of our intelligence is *bodily*, involving an intelligent interplay between *perception* and *action.*
- We competently go about our environment, recognizing all of the varied things in it, understanding what they do, and what they can be used for, and we are capable of using these things to accomplish our ends.
  - We do this all the time in day to day life without thinking about it, but when you do think about it, it becomes clear that it's an *extraordinary skill*.
- This sort of "embodied coping," as Dreyfus calls it, is an essential component of human intelligence, and is completely ignored by something like the Turing test.
- Attempts to apply symbolic AI to reproduce this sort of "embodied coping" have been quite rough.
- It seems like this is precisely the sort of intelligence that is *not governed by a definite set of rules.* We're always encountering radically new circumstances, and we use a huge store of implicit knowledge to practically cope with these new circumstances.

4 The Churchlands' "Wrong Kind of Computer" Argument

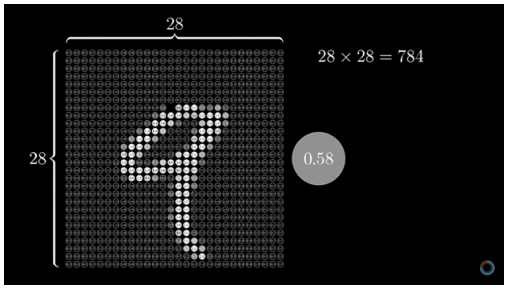
- If our goal is to recreate human-like intelligence, we're not going to do with software that is radically unlike the software implemented by the human brain.
- **Two Big Differences:**
  - Standard symbolic AI programs are *serial*, running processes in a sequence, one by one, whereas the nervous system is *massively parallel*, in that millions of signals are processed simultaneously.
  - Standard computer programs are *digital*, having *discrete* signals, whereas the neuronal signals are *analog*, having *continuous* output frequencies.
- If our goal is to replicate human-like intelligence, why not reverse-engineer the brain?

5 Connectionist AI

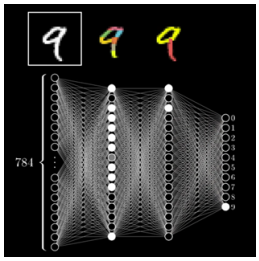
- **Neural Networks:** Two basic components:
  - **Layers of Neurons:**
    - **Input Layer:** Encodes the data that is given the network as an input.
    - *Hidden Layers:* Various transitory data needed as steps on the way to the final output.
    - **Output Layer:** Encodes the output of the network.
  - **Weighted Connections or Synapses:** Each neuron in every level is connected to each neuron in the next level by a *weighted connection.*
    - **Weights:** Determine how much the activation of a neuron in one level bears, positively or negatively, on the activation of a neuron in the next level.
- **Training:** A process by which the network is given inputs, produces outputs, those outputs are *evaluated*, the network is *modified*, and the process is *repeated*.

- This often requires a "training set," a bunch of correct input-output pairs, but it could also involve some other way of evaluating outputs, for instance, a score in a game.
- When a website makes you click all the boxes with a bus in them to "prove you're not a computer," you're actually just being mined for data that can be used to train computers.

- **A Simple Example:** From this video by 3Blue1Brown: https://www.youtube.com/watch?v=aircAruvnKk
  - We want to make a network that can recognize hand-written numbers from 0 to 9.
  - We can pixelate the images of these handwritten numbers on a 28x28 grid, so as to represent them in the *input layer* as a series of activation levels (between 0 and 1, depending on brightness) for 784 (28x 28) neurons, like so:



  - The *output layer* is just ten neurons, each one representing a digit. The "judgment" of the program is given by which neuron among these ten has the highest activation level.
  - The *hidden layers* encode intermediate steps along the way to the judgment of the number, for instance, judgments about the component parts. So, the whole network looks like this:



  - The *training set* is a large set of pairs of handwritten numbers and their correct numerical interpretations. The program is evaluated on the basis of how much the neurons for the correct number is rightly activated and how much the neuron's for the wrong numbers

- **Deepmind's Atari-Playing Programs:** https://youtu.be/rbsqaJwpu6A?t=562
  - Recall what Turing called "Lady Lovelace's Objection," that computer programs can't do anything new—they can only do what we program them to do:

    > "The Analytical Engine has no pretensions to *originate* anything. It can do *whatever we know how to order it* to perform."

    As Hassabis says here, however, the breakout playing program surprised programmers with this ingenious strategy!

6 Turning to Our Sci-Fi Scenarios

- **Asimov and Classical AI:** The most sci-fi writer who's most famously dealt with questions of artificial intelligience is Isaac Asimov who, writing in the 40s and 50s, clearly has a classical conception of AI.
  - The robots are explicitly programmed to follow the "three laws" of robotics, and conflicts between the rules can lead the programs getting "stuck," as we see in the story "Runaround."

- **How Ava was Made:** Ava's mind, we are led to believe, is a complex neural network trained on the data from the search engine BlueBook, as we'll as all of the data from people's cellphones when they make videocalls.
  - Using the entire internet to train a neural network is not unlike how current language models like LaMDA and GPT-3 are constructed. Compare Ava to the GPT-3 based bot, "Sophie": https://iamsophie.io/
- **The Scenario in "The Lifecycle of Software Objects":** Ted Chiang imagines "digients," artificial intelligences that exist solely in a virtual world.
  - They're initially programmed basically like Deepmind's Atari-playing program. They start out moving at random and eventually learn to practically cope with their environment through self-training.
  - They're then taught language and social interaction by people in basically the way suggested by Turing.
  - These purely virtual digients can then become physically embodied by being "uploaded" into real life robotic suits.