WiNG

The Future Of Machine Intelligence

MAY 11, 2015 JEFF SCHNEBLE SOFTWARE

"Machine intelligence" is a topic that's gotten a lot of press recently, from CNN predicting that robots will take your job (http://www.cnn.com/2014/11/10/opinion/ford-robots-jobs/) , to Elon Musk's concern that robots may take over the world (http://www.forbes.com/sites/ericmack/2015/01/15/elon-musk-puts-down-10-million-to-fight-skynet/) . All of this begs the question: what is machine intelligence? (Machine learning and artificial intelligence are more or less synonyms for machine intelligence, at least for the purposes of this discussion.)

This post provides a brief summary of: 1) what machine intelligence is technically; 2) the problems machine intelligence has a comparative advantage at solving; 3) recent developments that have made machine intelligence more applicable now than it was in the past; 4) examples of companies using a machine intelligence approach in practice; and 5) my thoughts on interesting areas for further exploration.

What is machine intelligence?

The Wikipedia definition (http://en.wikipedia.org/wiki/Machine_learning) of machine learning is, "an algorithm that can learn from data." While accurate, you may have already realized that this definition includes a lot of very simple models that are by no means new or particularly intelligent. For example, a linear regression model "learns" as new data points are added and the line of best fit is updated through the calculation of new coefficients.

A subset of machine learning is deep learning. Deep learning implies that the model includes multiple interacting layers, each of which potentially improves its performance based on past experience. For example, the output of one model may serve as the input to a second model, the output of which may serve as the input to a third model etc.

So why is deep learning considered interesting or important? While in principle a deep learning model may look like a sequence of simpler traditional models, in practice the system can exhibit

significantly more complex and nuanced behavior than any of its individual layers. An analogy helps make the point. The dynamics of a silicon transistor are fairly simple – in typical usage, a current between two gates is controlled by the voltage on a third gate. On its own, this behavior appears fairly straightforward. However, a few thousand transistors organized on a chip, in addition to a few similarly simple components, can create a microprocessor (for example, the Intel 4004 processor had 2,300 transistors). A microprocessor clearly exhibits more interesting behavior and can complete more useful work than an individual transistor.

This leads to the observation that a five- or ten-layer machine intelligence model can exhibit dramatically more interesting and potentially useful behavior than any of its individual layers could alone. For instance, a multi-layer model may be able to decompose an image into its constituent parts and accurately identify text and objects, even if it has never seen identical objects, or has not seen them from exactly the same angle. In principle this is similar to how our brains process complicated data streams such as those from our eyes or ears. This has led many to make comparisons between machine intelligence and human intelligence.

When most people talk about machine intelligence, they are referring to a model of some complexity, typically with multiple layers as described above. While the dividing line is blurry, this is most often the distinction between a “traditional” approach or algorithm and a "machine intelligence” or “artificial intelligence" approach.

It is worth noting here that machine intelligence has become a buzzword, and many companies and models cited as examples of machine intelligence do not meet the criteria defined above. It is tempting to describe a clever algorithm or well-designed product as “intelligent”; however the intelligence may be in the mind of the designer rather than in the program itself.

What is machine intelligence good for?

So why is machine intelligence interesting or important? There are certain classes of problems that lend themselves to a machine intelligence approach. At its core, well-implemented machine intelligence makes computer programs less fragile and more adaptable. Traditional computer programs are “hard coded” to solve a specific problem using a specific approach. If conditions change, the program breaks. For example, a program written by a human to play Pong could never play Breakout. However, an artificially intelligent program like DeepMind could learn to play

Breakout even if its creator had never heard of the game—more on that example below.

Problems with one or more of the following characteristics may be suitable to a machine intelligence approach:

1. **"Fuzzy" problems with unstructured data**. Unlike traditional models that directly map inputs to outputs, machine intelligence approaches can perform probabilistic and sometimes non-deterministic assessments. For example, a model may make a probabilistic "best guess" at the answer to a question, where both the question and the data may be unstructured and somewhat ambiguous. IBM's Watson is a good example of this approach (more on this below too).

2. **Changing conditions over time.** The "learning" aspect of machine intelligence stems from a model's use of previous data to improve the performance of future predictions. Unlike traditional approaches, where certain assumptions may be "hard coded" into the model, a true machine intelligence model will have significant degrees of freedom to adapt to changing conditions and to learn new behaviors. This is analogous to the way an intelligent creature can adapt to its environment. DeepMind is a good example of this.

3. **Large and dynamic data sets.** Like other automated IT solutions, machine intelligence can be highly scalable if implemented well. This makes the approach suited to data sets that are too large or too dynamic for humans to be "in the loop." This is particularly valuable when combined with the points above, as in some use cases tasks that used to require a human to make a subjective determination can now be fully automated at digital speeds and scale.

Some examples of use cases combining one or more of the above are provided in the last section.

Why now?

Machine intelligence is a field dating back to at least the 1950s (http://en.wikipedia.org/wiki/Machine_learning) , so we have to ask what's new now that makes the field an interesting area for exploration?

Perhaps the largest recent development has been the availability of ready-to-use machine intelligence libraries and services. In the past, using a machine intelligence approach required creating and coding a model from scratch, which often required a different skill set than typical

application development. Beginning in the late 1990s and early 2000s, libraries have been developed and made available which dramatically reduce the time to implement machine intelligence. For example, the OpenCV library, first released in 1999 and frequently updated since, provides a range of functions relevant to image processing. More recently, companies like Amazon and Google have begun offering machine intelligence as a service. Services like Google's Prediction API and Amazon Machine Intelligence offer a suite of machine-intelligence functions accessible through an API. Combined with cloud computing services themselves, these new services further reduce the amount it costs developers to begin integrating machine intelligence approaches into their applications.

A second development is the continued improvement in performance of microprocessor technology. Multi-layer machine intelligence models can be computational intensive, especially when run on large data sets. Many applications require some level of real-time performance to be useful, thereby limiting the complexity of models that can be used in practice. As microprocessors have continued to increase in performance, the complexity of models that can be realized has increased in step. (For example, current Intel microprocessors are approximately 30x faster (http://en.wikipedia.org/wiki/Instructions_per_second) than their counterparts from ten years ago.) Some believe we have recently crossed the threshold where computational power now allows for machine-intelligence models of significant complexity to outperform highly optimized traditional approaches. Graphical processing units have also recently begun being used to further accelerate machine intelligence models. It turns out that many of the matrix multiplication optimizations used to accelerate 3D rendering also apply to matrix multiplication problems in several machine intelligence approaches.

Examples of machine intelligence in the wild

So what are a few real-world examples of problems being solved by machine intelligence approaches today? I've included a few below, though this list is by no means exhaustive:

Netflix's recommendation engine: Netflix's engine is actually a linear combination of two models (http://www.quora.com/How-does-the-Netflix-movie-recommendation-algorithm-work) , one of which is a machine intelligence model. The machine intelligence model used is called a Restricted Boltzman Machine, and is essentially a two-layer graph model (http://www.machinelearning.org/proceedings/icml2007/papers/407.pdf) . This model uses a set of variables to

characterize each user. Expected movie ratings are then a function of these variables. For cases where a user has not yet watched a movie on Netflix, their expected rating for the movie is inferred from their personal variables, which are in turn inferred from other movies they have watched and rated. A simplified hill-climbing model is also used to improve the quality of forecasted ratings over time based on feedback. This model is on the simpler end of the machine intelligence spectrum we have defined here—it does use a multi-layer model, and it does improve with additional exposure to data. However, the model is only a two-layer one, and is operating on a highly defined set of inputs and outputs. Netflix's engine is an example of criteria #2 and #3 above.

DeepMind: DeepMind is a program that can learn to play Atari video games that it has never seen before. Over a period of hours, it went from not knowing how to play Breakout to setting a world record. The video (http://www.fastcompany.com/3042953/what-a-pong-beating-robot-says-about-where-silicon-valley-is-taking-ai?partner=rss&utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+fastcompany%2Fheadlines+%28Fast+Com is pretty amazing if you haven't seen it. DeepMind is programed with a goal—for example to increase its score in a game—and then experiments with different inputs to find a globally optimal solution to achieve the goal. This application is a particularly good example of criteria #2 above—the program starts with no knowledge of the game it is playing. It acquires knowledge over time through trial and error. If the game changes (for example, someone inserts a new cartridge), the program responds by learning the new game. This is clearly very different than a traditional program like DeepBlue, which was programed to play chess but could never learn to play checkers.

IBM's Watson: Watson (http://en.wikipedia.org/wiki/Watson_%28computer%29) is a machine-intelligence model that can answer natural language questions with natural language answers. Given the unstructured nature of both the question and the data set, Watson uses a probabilistic approach and suggests the most likely answer based on its analysis. Watson was able to beat the best human players at the gameshow Jeopardy. It is a particularly good example of criterion #1 above.

Spiderbook: Spiderbook is a startup that uses a machine-intelligence model to suggest sales leads based on a combination of information about your business and a scan of the entire internet. Spiderbook starts with unstructured inputs about the products your company sells, the types of customers you sell to, and who your competitors are. It then uses a multi-layer model to make a probabilistic assessment of who your most likely next customers are, based on publicly available data. Early customers have reported extremely high accuracy of these predictions, with the model

in many cases exceeding the accuracy of trained sales development reps. Spiderbook is an example of all three criteria above.

Interesting areas for further exploration

While machine intelligence could ultimately be useful in a wide range of applications, I personally am interested in areas where an intelligent machine can replace a human being in a workflow. Examples include:

Online and offline retail: Today, most stores have people on staff to answer our questions and recommend products for us to look at. Online, we're often provided recommendations that are similar to products we've already looked at. Well-implemented machine-intelligence models could likely recommend products and assist us with our shopping in both of these settings. A virtual assistant could provide an experience that feels like talking to a salesperson at a high-end department store, whether we're shopping online or in-store.

Customer support: Think back to your last customer support interaction—was it a good experience? Even if you spoke to a person, they were likely following a script and had limited degrees of freedom. An intelligent program could resolve your issue in real-time, with an interaction that feels like talking to a highly competent and motivated human. Not only does this application have the potential to dramatically improve our support experiences, it also has the potential to free thousands of real human beings from answering support calls in a call center.

Sales development: Most sales organizations have a front-end "sales development" function that performs research, identifies potential leads, and reaches out to them to test the water for interest. An intelligent machine could potentially identify prospects much more accurately, reducing the need to blast potential customers with phone calls and emails to find customers. Taken to its logical conclusion, machines could eventually reach out to prospects in a highly targeted way, provide product information, set up demos, and move an opportunity through the pipeline. Over time, perhaps machines will even be able to close their own deals! (No prediction on what their expectations around compensation will be though!)

© 1990 — 2018 Wing, Menlo Park, CA