A conversation with Chris Olah, Dario Amodei, and Jacob Steinhardt on March 21st and April 28th, 2015

Participants

- Chris Olah – http://colah.github.io/
- Dario Amodei, PhD – Research Scientist, Baidu Silicon Valley AI Lab; Scientific Advisor, Open Philanthropy Project
- Jacob Steinhardt – PhD Student, Computer Science, Stanford University; Scientific Advisor, Open Philanthropy Project
- Holden Karnofsky – Managing Director, Open Philanthropy Project

Note: This set of notes was compiled by the Open Philanthropy Project and gives an overview of the major points made by Chris Olah, Dario Amodei, and Jacob Steinhardt.

Summary

The Open Philanthropy Project spoke with Chris Olah, Dario Amodei, and Jacob Steinhardt as part of an investigation of machine learning. Topics covered included: areas of research in machine learning, past applications of machine learning, possible future applications, systemic issues in the field, and potentially important research and development (R&D) goals neglected by industry and other funders.

Topics in machine learning

Broad categories of work

Very broadly, machine learning generally involves designing an algorithm in order to make predictions from data. More narrowly, the task is often to learn to correctly "label" a data set (e.g. identify all of the images of dogs in a set of images).

Some broad categories of work in machine learning include:

- **Supervised learning** is the task of inferring a function from labeled training data. An example of this is learning how to distinguish between images of cats and dogs from a set of training data that consists of many images that are labeled as images of cats and images of dogs. Most of the recent advances in machine learning have been in this area.
- **Unsupervised learning** is the task of learning from unlabeled data. Instead of determining what label each data point should be given, the goal is to cluster data points together when they should be labeled *similarly* even though the precise nature of the label is not known. (For example, a system

might examine many images and determine that the ones with cats are similar to each other and the ones with dogs are similar to each other, even though it hasn't been explicitly given the mandate of classifying images as cats or dogs.) Google Brain's work to cluster images in YouTube videos with unlabeled data is an example of unsupervised learning.

- **Semi-supervised learning** could be thought of as "small supervised learning" and "large unsupervised learning." After unlabeled data points (which are often much more plentiful) are clustered together in meaningful ways using unsupervised learning, some labeled data can be used to provide information about the clusters of data. (To continue the above example, one might introduce one image labeled "cat" and one image labeled "dog" to the system described above, at which point it might be able to determine that many other of its images ought to be labeled as "cat" or "dog.")
- **Active learning** is a special case of semi-supervised learning in which a learning algorithm requests labeled examples (often from a human) with the aim of maximizing learning, especially in cases where it is costly to obtain labeled examples.
- **Reinforcement learning** is the task of learning to select actions in order to maximize total long-term "rewards," where some positive/negative value is assigned to each episode in a time series that represents the "reward" for that time series, and the history of positive/negative rewards up to the present is used to guide the decision. Reinforcement learning generally focuses on actions rather than predictions, and an important difference is that actions can have consequences for future actions in ways that predictions do not have consequences for future predictions. For example, in exploration/exploitation settings—where an algorithm takes some actions that are lower-expected-value in the short-term in hopes of identifying higher-value actions that can be repeated later—learning from early actions can affect later actions. For another example, putting gas in a car makes it possible to drive places later.

Some other types of work in machine learning (some of which overlap with the above categories) include:

- **Neural networks:** Neural networks take an input, which is treated as a many-dimensional vector (for example, if the input is a grayscale 64-pixel-by-64-pixel image, it might be treated as a 4096-dimensional vector where each coordinate represents the intensity of a particular pixel), and put it through a series of "layers." At each layer, the input vector is multiplied by a matrix of "weights" in order to produce another vector, which is in turn subject to some non-linear modification such as a rectifier function ($f(x) = \max(0,x)$ or sigmoid. These multiple stages (layers) of affine and non-linear transformations mean that the relationship between the input vector and the final "output" vector can be extremely complex and non-linear. The output vector generally represents the classification of the input (for example, if one

is trying to classify hand-written digits, the output vector might consist of 10 coordinates - one representing the probability that the digit is a 0, the next that it is a 1, etc.). The "neurons" in each layer are characterized by the weights and non-linear modification that maps the output of the previous layer (or initial input) to a new vector, with one neuron for each number in the new vector. One can "train" a neural network by entering many inputs whose correct classifications are known, and adjusting the weights associated with the "neurons" in order to get a higher quality set of outputs (where "quality" is measured by the aggregate score from the "loss function," which compares the neural network's output to the known correct output). A trained neural network, having been optimized to get good outputs from data whose classification is known, may then become useful for classifying data with unknown classifications. For example, one might feed in 10,000 images of handwritten digits (as vectors) and adjust the "neurons" to get this set of digits classified as accurately as possible, then use the trained "neurons" to classify further handwritten digits. The operation is somewhat analogous to running a linear regression, but by using multiple layers of different dimensionality, and in some cases by architecting the network so that some "neuron" values (matrix weights) depend on "neuron" values elsewhere in the network, one can model highly complex, non-linear relationships between input and output. Much of the recent progress in machine learning—particularly in image and speech recognition—has involved neural networks.

- **Generative models: a** generative model is a model that attempts to produce examples of data from a certain category, rather than just classifying it. So, for example, a generative model might look at many labeled hand-drawn numerals (as in MNIST), and then output many images of 3's (that are not the same as the 3's in the training set), or perhaps even try to represent all the variation in possible 3's. Generative models can be used directly (e.g. for text to speech systems) or indirectly (as an aid in providing examples that another model attempts to classify or discriminate). Generative models do not simply generate examples of data; rather, they explicitly model the "generating distribution." For example, in machine translation, researchers might either simply try to learn a function from English to Chinese, or else assume some latent "intended meaning" generates both the English and Chinese sentence, then infer the meaning from the English sentence and use that to generate the Chinese sentence. Only the latter would be a generative model.
- **Probabilistic graphical models:** Graphical models succinctly represent probabilistic dependence/independence relationships among a set of variables. They often allow computations which might normally require a lot of computational resources to be done more efficiently. Hidden Markov models (a specific kind of probabilistic graphical model) are often used for speech recognition. These models are appropriate because pronunciation of

a phoneme is often dependent on the pronunciation of phonemes used just before or after it, but fairly independent of the pronunciation of much earlier or later phonemes. They can also be useful for academic researchers working on subjects such as tomography.

- **Ensemble learning:** Ensemble learning involves combining different models (or sometimes variations on a single model) in order to get more predictive accuracy than what could be achieved using a single model. "Bagging" and "boosting" are examples of ensemble learning.
- **Structured output:** The goal of work on this topic is to produce machine learning models that output results with additional structure—such as a sentence, an image, a logical query, or a tree—rather than just a number corresponding to a classification.
- **Approximate inference:**
    - One example of approximate inference is the variational method (which is based on the calculus of variations). This approach is sometimes used to approximate the results of Bayesian updating. Other examples are discussed below under "Jacob Steinhardt's work."
    - Sampling is another approach to approximate inference, where instead of trying to approximate the entire probability distribution, one instead tries to draw approximate samples from the distribution. The two major approaches to sampling are Markov chain Monte Carlo (such as the Metropolis-Hastings algorithm) and sequential Monte Carlo. As one example of this type of algorithm, Gibbs sampling (a form of Metropolis-Hastings) involves re-sampling one parameter at a time from a high-dimensional distribution, and can produce samples that are correctly distributed asymptotically as the re-sampling process converges to a stationary distribution.

Most existing methods and current approaches to training machine learning models focus on accuracy. In addition to accuracy, sometimes there are additional goals for machine learning tasks, such as:

- **Resource efficiency:** e.g., decreasing the amount of memory used to accomplish the task.
- **Fairness:** for example, one might want to ensure that a machine learning system that is making decisions about who to approve for a loan is not implicitly racist or otherwise morally questionable.
- **Transparency:** ensuring that it is possible for the programmer to understand reasoning behind predictions.
- **Calibration:** ensuring that if an algorithm assigns X% confidence to a prediction, the prediction is true about X% of the time.

There is some work on these goals today, but it is fairly limited in comparison with work focused on accuracy. These problems pose additional challenges (in comparison with work focused on accuracy) because they are "ensemble"

properties, i.e. they depend on an entire model/set of predictions. In contrast, accuracy can be measured in terms of single predictions.

What Chris, Dario, and Jacob work on

Chris Olah's work
Chris has been working on machine learning for 2-2.5 years. One of his early projects involved investigating how changing the “hyperparameters” of a neural network affects its performance. Hyperparameters are parameters that are set in advance of optimizing the parameters in a model. Examples of hyperparameters include:

- The "learning rate": A common algorithm for training neural networks is "gradient descent." Under gradient descent, the neural network generates a model which it uses to make a prediction using training data and compares the prediction with the answer in the training data. The model then adjusts by taking a "step" in the direction that locally most improves the accuracy of the model. This prediction, comparison, step cycle repeats iteratively. The learning rate is a constant that affects the size of these steps.
- The "loss function": A function that takes a model prediction and a correct answer as input and outputs a positive number that is intended to capture the concept of how accurate the prediction was relative to the correct answer. The loss function affects the gradient descent process.
- The number of iterations of the training process described above.
- The number of neurons per layer of the neural network.
- The number of layers in the neural network.

This work primarily involved studying the consequences of changing hyperparameters in relatively small and simplistic neural networks. Yoshua Bengio has also worked on problems of this kind.

Another project involved visualizing neural networks in order to understand what they do internally. For example, one of Chris's blog posts looks at the topology of neural networks "thin" enough that it was possible to picture what the networks were doing at each layer. Data examined by neural networks consists of vectors in a high-dimensional space and the neural network "bends" the data in that space. Chris's work on visualization aims to improve understanding of how this process works. When he first worked at Google, Chris was developing techniques for understanding larger neural networks, and then applying them to models that Google had built (such as a translation algorithm).

Recently, Chris has been working on dynamic versions of neural nets, where the structure of the neural network changes in response to the input, with different computations being done for different inputs.

Jacob Steinhardt's work
Jacob has recently been working on probabilistic and statistical reasoning for problems that are very computationally difficult. The goal is to create approaches that can naturally incorporate computational approximations into the statistical model and learning process, as opposed to classical approaches such as variational inference and Markov chain Monte Carlo which make approximations externally to the model.

Percy Liang (Jacob's advisor), and his lab, work on question answering, latent structure learning, and other problems. In question answering, the task is to take a database with information about various objects and a query, and to output a collection of objects from that database that is a good answer to the query. For example, if the question was "Where was Barack Obama born?", the query would direct you to look for an object named "Barack Obama" and a relation "born in" and see what the object bears the "born in" relation to. In this case, the answer would be "Honolulu, HI."

This work on question-answering is different from work on IBM's Watson. Watson is an AI system that answers questions. It beat reigning human champions at Jeopardy on national television. Watson takes a large number of existing question-answering systems and intelligently combines them. Watson relies heavily on looking for word patterns, whereas Prof. Liang's approach relies more on logical reasoning.

Jacob's work is more conceptually focused and currently centers on building better frameworks for computationally-bounded statistical reasoning, especially in the context of approximate inference for structured prediction tasks (of which the question-answering task above is one example). His time is roughly divided between thinking at a high level about what aspects of existing methods are imperfect but seem like they could be improved, designing and implementing frameworks for realizing these improvements, and solving concrete tasks (to gain intuition and validate approaches).

Other questions Jacob works on include:
1. How much data is needed to solve learning problems under different resource constraints?
2. How can we reify computation as part of a statistical model?
3. How can we relax the supervision signal to aid computation while still maintaining consistent parameter estimates?

Dario Amodei's work
Machine learning researchers are attempting to solve speech recognition, i.e. to take in an audio recording that contains background noise and/or has a lot of "ums" and "ahs," and output a transcript that is better than what a human could make. The

state of the art gets ~30% word error rate in a noisy environment (e.g. background noise, other people talking, and/or low-quality microphones), and ~6-7% in a clean environment (one person talking at a time, no background noise, good microphone). These error rates are frustrating for users. But if an error rate that is about 4x lower can be produced, it seems possible that the product would be used much more widely.

There are a few directions along which progress could improve the error rate:

- Obtaining/improving data:
    - Getting more supervised data.
    - Making use of data that is not entirely supervised.
    - Data augmentation: For example, one approach is to take quiet data and put noise on it, e.g. by putting it through a low-quality phone microphone or superimposing two people speaking. This can help train a voice recognition system to work in noisy environments.
    - Improving ability to process and store data rapidly.
    - Data quality control.
- Network size: many challenges relate to getting the neural network to run fast, especially through parallelization. E.g., with 100 graphical processing units (GPU), is it possible to split the network (or the data going through it) among the 100 GPUs? Researchers make use of parallelism in both the model and the data.
- Small-scale changes to neural network architecture:
    - Changing the number of layers in the network.
    - Changing the shapes or connectivities of the layers.
    - Adding more, less, or different kinds of regularization,.
    - Changing the format of the input or output data.
    - Changing the loss-function.
    - Curriculum learning: In what order do you present data to a network? The order of presentation of the training data can affect results. For example, presenting "easy" data before "hard" data often works better, though researchers have a very limited theoretical understanding of why this is.
    - Increasing the number of nodes in a layer of a neural network.
- Large-scale architectural changes: For example, until three years ago, most speech systems used hidden Markov models to model the sequence of phonemes. This part of the model offered probabilities that a given phoneme would occur next given what came before. The output model offered a probability distribution (a Gaussian mixture model) over what phoneme was being pronounced at a given time, based on an audio recording input. In 2012-2013 there was a transition largely coming out of Geoffrey Hinton's lab in which people started using deep neural networks for the "acoustic model"—i.e. the part of the voice recognition system that transforms a 30-millisecond snippet of audio into a representation close to a plausible

phoneme—but continued to use hidden Markov models to model the transition between phonemes. There was another transition where Baidu and DeepMind moved toward using a recurrent neural network (rather than a hidden Markov model) for the transition between phonemes in addition to the acoustic model for identifying phonemes. The next step in this progression is unknown, but researchers look for opportunities to make additional transitions of this nature.

Day to day, a typical workflow looks like:

1. Build a system
2. Test the system on a demo server
3. Analyze its strengths and weaknesses
4. Consider test sets that the current data may not be representative of
5. Form a hypothesis about the causes of errors
6. Identify changes that could be made to the system that might address these issues

There are a number of challenges to making small-scale changes to network architecture. For example, there has been relatively little systematic thinking about how to make these changes. It is more of an art than a science right now. Systematic methods like gradient descent can't be used right now to improve many hyperparameters because the hyperparameters are non-continuous, making it impossible to take derivatives.

One common type of academic machine learning research

One common type of academic machine learning paper could be abstractly outlined as follows:

1. Algorithm A is the state-of-the-art, and it can do some task of interest, X.
2. However, A can't do a related task of interest, Y.
3. Y is an interesting task. If someone solved it, it might be useful/interesting in itself, or lead to something else that would be useful/interesting.
4. Algorithm A can be extended/altered to create an algorithm A' as follows.
5. On a toy example (described in the paper), algorithm A' does task Y.

Some relatively common variations on this theme is to argue that algorithm A' has similar performance to algorithm A, but is much simpler/more elegant/less over-engineered.

Trying to outperform state-of-the-art benchmarks, such as image classification on ImageNet (a standard test-set for image classification), is not a typical goal for an academic paper working outside of computer vision and natural language processing.

Potential impact of machine learning

Recent progress in neural networks

There has been a lot of progress in neural networks because machine learning researchers have learned that with enough computational power and data, there are many tasks that they can solve with supervised learning that people previously thought would be extremely challenging. For example, neural networks played an important role in advances in image classification and speech recognition. DeepMind combined reinforcement learning with a neural network in order to achieve human-level performance on a wide variety of Atari games using a single architecture. The neural network interprets the state of the game (using only observations of the screen and/or score, not of the game's internal state) and the reinforcement learning component decides which actions to take.

One consequence of this is that less research is happening outside of supervised learning than was happening previously, and that many people are trying to transform the problems they work on into supervised learning problems.

It is common to use neural networks where many neurons have the same parameters. Convolutional neural networks and recurrent neural networks both work this way. Convolutional neural networks use copies of a single neuron repeatedly for a given layer and are very effective for image classification. Recurrent neural networks often use copies of a single neuron and take sequential inputs (such as a series of words) and have been successful in tasks like predicting which word is likely to be next in a sentence given the previous words and speech recognition

In determining what kind of architecture to use, it is important to consider what regularities would be expected in a given problem domain. E.g., a researcher training a convolutional neural network, is (metaphorically) telling the neural network to focus on local relationships between images, colors, and textures in a particular section of an image. This kind of assumption implies that if cat fur looks a certain way in one part of the image, then it looks that way in other parts of that image.

Reasons for the recent progress in supervised learning

Convolutional neural networks were necessary for much of the recent progress, but they have been around for more than 20 years. They were invented by Yann LeCun in the 1990s at Bell Labs. Recently, researchers have made much larger convolutional nets and combined them with using large numbers of GPUs (with now much greater computing power than was available in 1990). A 2012 paper by Krizhevsky, Sutskever, and Hinton sparked much of this excitement.

Applications of machine learning

Applications of machine learning so far include:

- Recommendation systems
- Search
- Machine translation
- Speech recognition
- Handwriting recognition (e.g. for depositing checks)
- Self-driving cars
- Finance
- Recognizing street numbers from Google Maps's street view

Likely and/or hoped for progress in the next 5 years or so

The field is attracting additional talent and there has been increasing involvement from industry in machine learning. These trends are likely to continue.

Areas where there might be particularly interesting progress in the next several years include:

- Language translation.
- Voice recognition.
- Dynamic neural nets (neural nets that change architectures as they are trained).
- Novel architectures (analogous to past "large-scale architectural changes" discussed above).
- Big data (could be important because it would improve training).
- Approximate inference and computationally bounded reasoning in general. This may be very important for structured output problems.
- Transparent/robust AI (of the sort that some Future of Life Institute grantees are working on).
- Fairness (e.g., avoiding race-based discrimination when classifying data).
- Calibration (systems that are accurately able to distinguish which of their classifications are more vs. less likely to be correct).
- Causal inferences.
- Dialogue systems. Avoiding verbal dialogue systems that ask highly repetitive questions and don't allow for quick clarifications is a challenge in this area. For example, if you ask SIRI to find a flight to Bermuda and it returns you a flight to New Jersey, you can't fix it by saying, "No, I meant Bermuda." SIRI looks differentially for actions (go to the grocery story), times (tomorrow), and places (San Francisco), but the system that does this is fairly brittle.
- Text summary.
- Self-driving cars.
- In-principle capability to automate a significant fraction of factory work (though it would likely require additional translational work for specific types of factory work that would be automated).
- Online education – for example, it would probably be possible to give people targeted explanations based on the problems they are getting wrong. The

people interviewed were unsure about how much of this is already being done.
- Online dating (algorithms for assigning people to likely matches could be improved).
- Medicine – e.g. diagnosis and treatment selection.
- Credit – e.g. deciding whom to give loans to. There is some regulation that makes this hard, so many of the challenges are legal rather than technical.
- Energy – e.g. one could imagine monitoring cell phone transmissions and figuring out what factors affect energy use, and then use nudges to affect those factors in a way that decreases energy use. More ambitiously, a machine learning system might manage a power grid for a city.
- Science – representing scientific knowledge; automating steps in science (especially biology); structured search (e.g. identifying all of the proteins relevant to a given process).
- Customer service call centers.
- Making it easy for non-experts to use advanced techniques in machine learning. Because researchers made a lot of progress in machine learning recently, there is a large lag between cutting-edge techniques and the machine learning techniques that are used to solve many problems (such as matching on dating websites). Some current work is aimed at making it easier for people without expertise in machine learning to apply cutting-edge techniques in machine learning. Success in this area could significantly improve translational work in machine learning, or even eventually make it possible for non-experts to use cutting-edge machine learning techniques to train machines to do a wide variety of tasks.

Further down the line—as much of the low-hanging fruit from supervised learning gets plucked—the following areas may become increasingly important:
- Active learning
- Partially supervised / semi-supervised / unsupervised learning
- Systems that take actions in the world (such as reinforcement learning)
- Getting a neural network to learn general rules (such as laws of physics) that could then be applied to a different domain (such as celestial mechanics)
- Incorporating side information, such as neural Turing machines
- Change-point detection – i.e. detecting when a time-series or stochastic process changes its probability distribution
- Non-stationary/context-sensitive distributions – a probability distribution is "stationary" if the distribution does not change when shifted in time. Non-stationary distributions are harder to train.
- Heteroscedasticity – i.e. circumstances where the amount of error in a model varies over the range of values where the model makes predictions

What jobs might be automated?

With the visual processing that is possible today, there aren't deep obstacles (in terms of advances in algorithms) to automating most manufacturing. Rather, it is an extremely difficult engineering problem that will have to be done differently for each application.

Other areas where there could be automation include:

- Drug discovery.
- Tasks that involve driving cars or trucks.
- Biological experiments, especially the work of graduate students. It seems that progress could be made in this area with moderate effort, and could be highly impactful.
- Program synthesis and optimization. Program synthesis is the task of creating program code from data, such as output examples, constraints, or pseudocode. This could significantly increase productivity in software engineering.
- Routine mathematics.

A common theme with automation is that there are strongly non-linear returns from moving from 90% automation to 100% automation.

Robots can do a variety of tasks if they don't have to move around, but have a much harder time if they have to move around in an unfamiliar environment. For example, robots are much worse at walking than humans and usually can't get up if they fall over. They also are not very good at picking up objects. Robotics is a different field from machine learning, and the Open Philanthropy Project could likely get additional information by talking to a roboticist, especially someone who knows about machine learning.

Issues a philanthropist could focus on

Neglected goals

Some areas where progress could likely be made using machine learning, but might be neglected by industry, include:

- Operations of government.
- Technologies in the developing world. For example, it is common for internet bandwidth to drop to very low amounts for days in a row in Kenya. It may not be a machine learning problem, but optimizing that network could be a data processing problem.
- Automating elite jobs (such as lawyers and doctors) at a rate that keeps up with automation of less elite jobs, which could reduce the risks of growing inequality. It seems plausible that it would be possible to automate various aspects of legal work, such as finding relevant cases and checking for

whether a policy would comply with existing law with pattern recognition systems. It may be harder, however, to synthesize novel legal arguments or represent someone in a courtroom.
- Work aimed at reducing possible risks of artificial intelligence.

A general challenge is that attempts to automate the work of a given profession/industry may be resisted by members of that profession/industry.

Rather than focusing primarily on accelerating progress in machine learning, it seems more important to make the field put more energy into thinking through social impact and ensure that important problems get enough attention (such as transparent AI, potential social impacts of AI, and translational work).

A group of economists and machine learning researchers could try to forecast areas where processes and/or jobs were likely to be automated. It would probably not be very expensive, and they might have insights on this type of question.

There may be other opportunities for philanthropists in the general category of "translational machine learning." Automating specific tasks could take a great deal of time and money. In cases where the incentives are right, industry can deal with this. But in cases where the incentives are not right, there could be an interesting role for philanthropy. A general approach would be to think of what tasks it would be great to have automated, and then support research toward automating them. This seems like something that could be done today. For example, some people are working on using machine learning to create new diagnostics for malaria.

Systemic issues

Systemic issues in machine learning research—such as inefficiencies in the tenure, publication, and funding systems—seem much smaller than systemic issues in biology, though there are some problems of this type in the field.

Issues highlighted by Jacob

1. There are two main conferences (International Conference on Machine Learning and Neural Information Processing Systems) where researchers can submit papers in order to get prestigious publications. Every year about 1000 people submit to the conferences. They use machine learning to match papers to reviewers, and it enforces incrementalism because the person reviewing a paper is often someone who wrote the most recent related paper. This forces people to write papers in a very defensive way that is least likely to offend anyone. There have been experiments to assess the inter-rater reliability of reviews for these conferences.
2. Researchers tend to assume that every machine learning paper should have experiments showing that the authors have produced an algorithm with a performance metric higher than some other algorithm. However, these

experiments often are not very informative. One reason for this is that the performance metrics are often computed in very different ways. This can lead to a perception that the field is highly empirical when, in fact, it is often not.

3. Researchers are expected to have published an unreasonable number of papers. For example, if a PhD student wants to be eligible for a top academic job, they are expected to have about 15 first-author publications (or 3 publications per year of graduate school). This does not seem conducive to probing questions deeply, and incentivizes people to work on tasks where there is existing data, rather than putting together new data sets (which takes more time but is often very valuable). Consistent with this, deep work is fairly rare and most publications are driven by clever ideas which may not lead to progress on more fundamental issues. One potential driver for this issue is that a paper need not be exceptional in order to be published in the top two conferences in the field, and there are limited rewards for having a paper significantly better than what is necessary to be published in those conferences because people often simply count publications in top conferences when casually measuring researcher performance.
4. Another reason that deep papers are rare is that professors often come up with research ideas, but graduate students are primarily responsible for execution. A professor using this strategy can publish a significantly larger number of papers, but it means that research driven by deep intuition built up over a long period of time is underutilized in the research process. There are some exceptions to this, but overall it seems that such deep work should be more strongly encouraged.
5. There is a very limited understanding of almost every area of the field except for supervised learning. Working in these areas is risky, and making progress is not guaranteed, which may disincentivize people from doing high-risk, high-reward projects in these other areas.

Issues highlighted by Dario

Dario's perception is that there is a lot of overlap with the systemic issues in biology, but they aren't as bad and competition for funding isn't as severe. For example, a common issue in biology is that researchers have very strong incentives to submit every paper to *Nature* because there is a small chance that the paper will be accepted and that it will be a major success for the authors' careers. There are many special formatting requirements, and it often takes a very long time to get the paper in the correct format. When the paper is rejected, the team will then submit the paper to a specialized *Nature* publication (e.g. *Nature Neuroscience*). Then, revisions to the paper are often requested, often suggesting that the author should cite the paper of the reviewer and/or do additional experiments. This can significantly delay publication and often requires bringing on an additional graduate student. The result is that there is often over a year of work to do between submission of a paper, revisions, and acceptance.

Some people working in industry feel pressures toward incrementalism, though this may vary by company, and there are some related pressures within academia. (For example, DeepMind has been focused on some new and unusual approaches.) Instead of working on more incremental projects, it might be better if industry researchers had more opportunities to, e.g., explore conceptually different approaches to building deep learning systems, solve multimodality problems (including both video and audio), or tie neural networks to reinforcement learning.

Issues highlighted by Chris

It seems to Chris that trying to communicate and explain things well is underincentivized. It's common for people to do good work, but explain it poorly. For example, it is not uncommon for deep learning papers to omit details about how a model was trained that would be important for replication. In Chris's view, this is a general issue in mathematics education, including topics like calculus and information theory.

A large fraction of academia—especially in deep learning—are being pulled into corporate environments. This could be problematic if:

1. A lower fraction of research gets published.
2. Research becomes less transparent.
3. Research focuses on short-term goals at the expense of long-term goals. For example, Yoshua Bengio thinks that the current focus on supervised learning—and relative neglect of unsupervised learning—is an example of this dynamic. There are multiple orders of magnitude more data available for unsupervised learning than for supervised learning, so this gap could be important.

Researchers often lack tools (such as systems infrastructure) for large experiments. For example, Google has very powerful libraries that make this work easier to do, but many researchers outside of industrial settings lack access to equivalent tools.

Possible interventions for a philanthropist related to systemic issues

A philanthropist could:

1. Try to prevent machine learning from developing some of the systemic issues common in biology.
2. Try to separate expectations for researchers whose work involves a lot of "grind" and researchers whose work focuses on risky attempts at major breakthroughs.
3. Incentivize researchers to think more about impacts of machine learning on society. The field currently spends very little time thinking about societal impacts of what they are doing.

4. Improve incentives for researchers to take a more deep/thoughtful approach to research. For example, a funder could offer fellowships to people who have a track record of being more thoughtful about their research.
5. Create a research institute with substantially different incentives than either industry or academic lab, like the Allen Institute. This institute might be able focus on problems with high social value that are disincentivized under current structures, such as some of the "neglected goals" listed above. Alternatively, it could be modeled after HHMI and give outstanding researchers the freedom to work on what they think is best.
6. Help create public goods, such as high-quality datasets.
7. Seek to change the peer review process to avoid some of the issues highlighted by Jacob.

All Open Philanthropy Project conversations are available at
http://www.givewell.org/conversations