From Neural Networks to Deep Learning: Zeroing in on the Human Brain

Pondering the brain with the help of machine learning expert Andrew Ng and researcher-turned-author-turned-entrepreneur Jeff Hawkins.

By Jonathan Laserson

DOI: 10.1145/2000775.2000787

In the spring of 1958, David Hubble and Torsten Wiesel stood helpless in their lab at John Hopkins University. In front of them lay an anesthetized cat. In the cat's skull, just above its visual cortex (an area in the back of the brain), there was a 3mm-wide hole through which electrodes recorded the response of several neurons. Hubble and Wiesel had conjectured that these neurons would fire when shown the right stimulus on the retina. If only they had any idea what that stimulus could be. For days they had shown the poor cat different shapes and shiny spots of light, testing different areas of the retina, but nothing elicited a response. In their book, *Brain and the Visual Perception*, Hubble and Wiesel give a personal account of this experience:

> *The break came one long day in which we held onto one cell for hour after hour. To find a region of retina from which our spots gave any hint of responses took many hours, but we finally found a place that gave vague hints of responses. We worked away, in shifts. Suddenly, just as we inserted one of our glass slides into the ophthalmoscope, the cell seemed to come to life and began to fire impulses like a machine gun. It took a while to discover that the firing had nothing to do with the small opaque spot—the cell was responding to the fine moving shadow cast by the edge of the glass slide as we inserted it into the slot. It took still more time and groping around to discover that the cell gave responses only when this faint line was swept slowly forward in a certain range of orientations [1].*

That cell became the first discovered example of what is now known as "orientation selective cells," a type of cell which is actually prevalent in the visual cortex. These cells fire when they detect an edge oriented at a particular angle, in a specific area of the retina. Why orientations? Many researchers agreed that this type of pattern was reoccurring frequently in natural visual scenes, which might justify why the brain would have an explicit representation of such patterns, but for many years no one was able to show that oriented lines are more important than other representations like circles, blobs of light or stripes.

A MATHEMATICAL MODEL

To answer this question, Bruno Olshausen and David Field, then working at Cornell University, decided to model the problem mathematically using the following experiment. They collected a bunch of natural images—trees, lakes, leaves, and so on—and extracted thousands of small image patches, 16x16 pixels each, from random locations within these images. To make things simple, they only used gray-scale pictures. They proceeded to ask the following question: "Say your goal is to reconstruct each one of these image patches, but all you can use is a fixed set of 400 slides of size 16x16 pixels each. You're allowed to put slides on top of each other to approximate each patch, but cannot add anything extra, only play around with the slides [mathematically, this means that each image patch has to be a linear combination of slides]. If you could pick any 400 slides, which ones would you choose?"

Since the goal is to reconstruct 16x16

patches and you're allowed to pick 400 slides, there is a simple answer to this question: There are 256 pixels in a 16x16 patch; number their locations from 1 to 256, now let each slide depict a single pixel out of these 256 pixels. The first slide will thus depict a small black square in location #1, the second slide will depict a small black square in location #2, and so on (144 slides will be left blank). Now, when a black-and-white image patch comes along, you can reconstruct it exactly by stacking all the slides corresponding to the black pixels in the patch one on top of another.

The described method undoubtedly works. In fact, using this representation you could reconstruct any random black and white patch, even those that look like white noise. Yet somehow this doesn't feel like the right solution. First of all, it doesn't seem efficient. In nature, the patches are seldom completely random. There are correlations to exploit—patterns such as surfaces and lines that appear over and over again in visual scenes. Furthermore, as we learned from Hubel and Wiesel's experiment, the brain chose to represent such patches as combinations of oriented lines, not pixels. Why? If all the brain was trying to achieve was an accurate reconstruction of the patches, then the above 256 single-pixel slides do the trick (and with 144 slides to spare). Clearly the brain must be optimizing over additional considerations.

SPARSE CODING

Olshausen and Field came up with the following proposal: *sparseness*. When neurons fire in the brain they consume energy and other resources. It would be desirable to have a good representation of the visual data coming from the retina, while reducing neuron firing to a minimum. Going back to our slides problem, we want a pool of 400 slides that would enable us to build a reasonable reconstruction of each patch, but we'd like to use as few slides as possible per patch. That way we wouldn't have to work so hard collecting all the necessary slides for each patch. Mathematically, the function being optimized has two terms: One term rewards solutions that give accurate reconstructions; a second term rewards solutions that induce sparse reconstructions.

The algorithm that optimizes this function is called "sparse coding." It starts with a completely random set of slides and iteratively improves their content. In the first part of each iteration, the algorithm keeps the slide content fixed, and finds the optimal sparse representation (i.e., the optimal linear combination coefficients) for each patch using the fixed slides. In the second part of each iteration, the coefficients are fixed so that we know which slides are used to approximate each patch, while the slide content is optimized. When you run the algorithm, it's hard not to feel excited as the random pixels start forming patterns that end up being very close to, you guessed it, oriented lines.

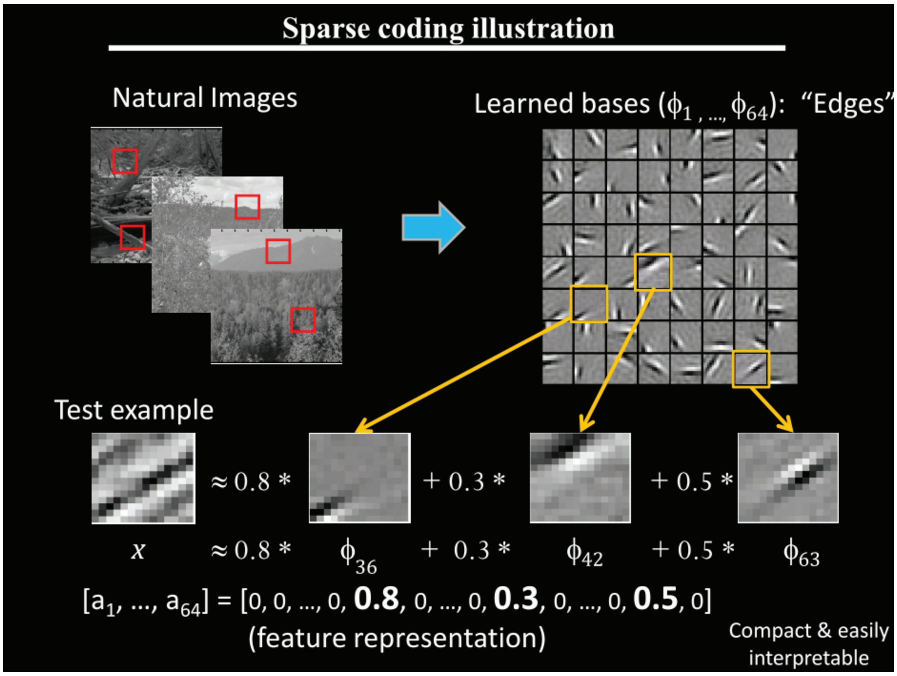
Following these exciting developments, an old idea reemerged in the neuroscience community. Could it be that something even deeper was discovered? Maybe, thought researchers, the same optimization happens in all layers of the visual cortex. Moreover, maybe the same learning algorithm is used in many regions of the brain to discover the building blocks of speech, sound, touch, language, and everything else our senses record.

A WHOLE NEW APPROACH TO AI

When Olshausen and Field published their paper in 1996, their goal was limited to illustrating a possible way in which the brain may be processing information. Little did they know that their paper would lead to a whole new approach to artificial intelligence (AI). However, it took almost 10 years before the AI community even took notice. "It no longer surprises me when great ideas lie fallow for so many years," says Andrew Ng, a professor at Stanford University and one of the leaders of the "Deep Learning" paradigm that came out of sparse coding. Ng explains that in 1996, limited computing power allowed only a small-scale implementation of the algorithm, not enough for it to work well on real-world problems and for the research community to realize its full potential.

But in 2005, a remarkable thing happened. An object recognition algorithm built upon sparse coding was able to beat all other algorithms on the most

Figure 1: Sparse Coding Illustration. Thousands of 16x16 image patches are extracted from pictures of natural scenes (top left). From this data, a fixed set of 64 slides (also called bases) is learned (top right). Each image patch is represented as a linear combination of slides (bottom). The slides are chosen so that these linear combinations are sparse, i.e., most of the coefficients are zero. The resulting slides correspond to oriented lines, similar to the receptive fields of cells in the visual cortex.


Sparse coding illustration
Natural Images
Learned bases ($\phi_1, ..., \phi_{64}$): "Edges"
Test example
≈ 0.8 *
+ 0.3 *
+ 0.5 *
$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$
$[a_1, ..., a_{64}] = [0, 0, ..., 0, \mathbf{0.8}, 0, ..., 0, \mathbf{0.3}, 0, ..., 0, \mathbf{0.5}, 0]$
(feature representation)
Compact & easily interpretable

challenging benchmark of that time—the Caltech 101 dataset [2]. This dataset includes pictures of objects from 101 categories (such as planes, motorcycles, elephants, etc.), and the brain-inspired algorithm correctly classified 42 percent of them. (Indeed, current vision algorithms correctly classify around 80 percent, indicating how much computer vision has improved, but back then 42 percent was the very best.)

"It was a surprise for me too," admits Ng. "I've been very skeptical about biological exploration most of my professional life. What we are seeing is that by taking ideas from neuroscience and incorporating them into Deep Learning algorithms, it actually increases the performance of the algorithm." Professor Ng was not the only one who was skeptical. The idea that we can look to the brain for inspiration on how to build intelligent machines is not very popular in the AI circles. "I faced a ton of skepticism and advice from well meaning colleagues, not to do this. I get almost none of that now," says Ng.

ON INTELLIGENCE

The person who has perhaps done the most to advance the idea that the brain can teach us to do better AI is Jeff Hawkins, who in 2004 published his book *On Intelligence*. "There was an institutional problem," explains Hawkins, recalling his struggle with the academic world when his grad-school application was rejected by MIT in 1981. "These people thought that studying brains would limit your thinking. You cannot do this easily in the normal channels of the academia. That is why I created RNI. That is why I created Numenta. That is why I wrote *On Intelligence*." Hawkins is referring to the Redwood Neuroscience Institute in Berkeley he founded in 2002, which is now run by Bruno Olshausen, and to his own start-up company Numenta Inc.

In *On Intelligence*, deliberately written at an undergraduate level, Hawkins focuses on the neocortex—a large sheet of neurons folded into the outer layer of the brain. He points out two main concepts that govern the neocortex—*sequence prediction* and *hierarchies*—and makes a compelling case that we can use the same principals to build intelligent machines. "Many people thought the brain is so complex and messy, we know so little about it, that it will be fruitless to work on it. And we said, that's not true—we know a lot, and we can make progress." The book eventually made its way to the libraries of many AI professors. "I was buying stacks of his book to give out to incoming students," says Andrew Ng, "it's a hugely inspirational book."

DEEP LEARNING 101

So how does Deep Learning work? Recall that in sparse coding, the idea is to come up with a basis, a pool of shared building blocks, so that every data instance can be reconstructed as a different linear combination of the same building blocks. You can think of sparse coding as a two-layer Deep Learning algorithm, where the data is in layer 0 and the building blocks are in layer 1. Now treat the layer-1 building blocks as your new data instances and try to find layer-2 building blocks to reconstruct the layer-1 building blocks. You now have a three-layer Deep Learning model. Pile on a few more layers and you end up with a nice deep representation model, in which the data lies at the bottom, and every layer maintains only the essential ingredients required to reconstruct the patterns in the layer beneath it.

Once the whole model is trained, you can feed it with a data instance and it will find the activation levels (the coefficients of the linear combinations) of all the building blocks in all the layers. These activation levels form a new representation of the picture, no longer in the language of pixels but in terms of a weighted sum of the different building blocks. The hope is that once you learn the model using a dataset like Caltech 101, the building blocks at higher levels will be activated only when a specific type of object appears in the image, for example a cat or a dog. Hence by looking at these high-level building blocks, one will be able to tell which type of object appears in the picture.

Deep Learning algorithms are already achieving state-of-the-art results, bypassing methods incorporating hand-engineered representations obtained through years of domain-specific expertise.

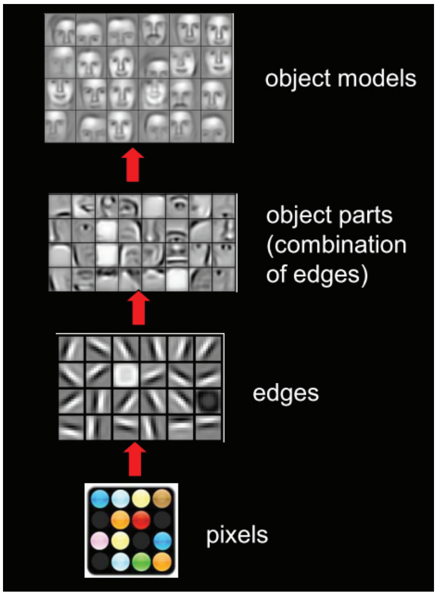
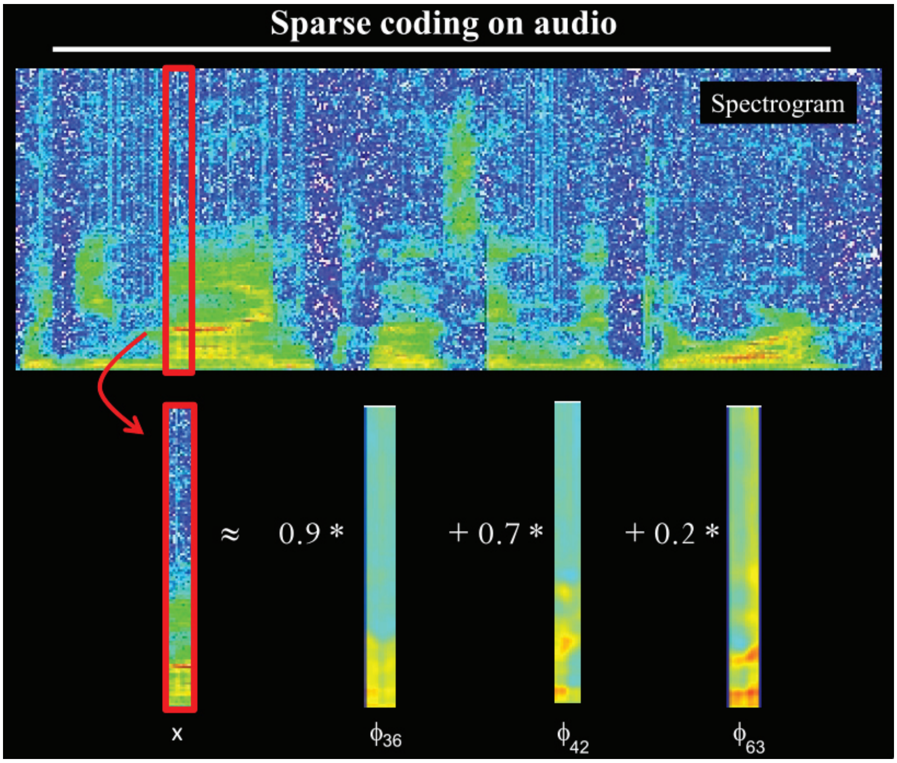


Figure 2: A Deep Learning model applied to pictures of faces stored as pixels in layer 0 at the bottom; the model discovers oriented lines in layer-1, face parts in layer-2, and templates of whole faces in layer-3.

In AI-speak, this type of learning is called "unsupervised learning," because we never give the model any information about the data except for the data itself. In the above example, we did not tell the algorithm about entities like cats and dogs, but simply threw a bunch of unlabeled pictures at it and took a "leap of faith" that the algorithm would find the two pets statistically different—sufficiently so that some building blocks would become specialized for cats while others would specialize in dogs. This is in contrast to "supervised learning," where each training example comes with a label stating its content. For example, a training image of a cat would be labeled "cat," hence the algorithm would know right off the bat that it had better find what makes a "cat" sta-

Figure 3: Sparse Coding for Phoneme Recognition. The data, representing uttered phonemes, is passed through a spectrogram (top) that shows the energy in each frequency (y-axis) through time (x-axis). A set of bases is learned so that each audio segment (bottom left) is represented as a sparse linear combination of bases (bottom right). A Deep Learning algorithm currently shows the best performance on the benchmark speech recognition dataset TIMIT.



tistically different because in the future it will be asked to tell if an unlabeled picture has a cat in it.

STATE-OF-THE-ART RESULTS

In 2006, Geoffrey Hinton of The University of Toronto published his seminal paper on *Deep Belief Nets*, laying the foundations of Deep Learning. Following his work, the last few years have seen a huge growth in the number of publications describing Deep Learning implementations. Many of the AI researchers who use these algorithms care less about mimicking the human brain and more about nailing down notoriously difficult tasks such as object recognition, speech recognition and action classification in video. On many benchmarks, Deep Learning algorithms are already achieving state-of-the-art results, bypassing methods incorporating hand-engineered representations obtained through years of domain-specific expertise.

Jonathan Laserson: **Is this as easy as it sounds? In a new domain, how much work do you need to put in to make a Deep Learning algorithm become the state-of-the-art?**

Andrew Ng: Depends on the domain and state-of-the-art. In the early days of Deep Learning, three to four years ago, the algorithms had more parameters and required an in-depth practitioner knowledge to get things to work. But as the field made progress in the last few years, the algorithms have become much simpler. The odds of someone new to this applying the algorithm and getting reasonable results is much higher. A naive implementation will make up a fairly good algorithm that will beat the baseline result. Still, if you want to get state-of-the-art results you need to play with the architecture. Judgment has to be made regarding how much data to train on vs. how large a model to build, how many hidden layers and the number of connections, so that you don't run out of memory and computational time.

JL: **Geoffry Hinton, world expert on both Neural Networks and Deep Learning, coined Deep Learning as "the next generation of Neural Networks." How is Deep Learning different from Neural Networks? And what makes it attractive now?**

AN: Scalability. The huge change between learning Neural Networks in the 1980s and now is labeled vs. unlabeled data. Back in the '80s, the majority of learning was supervised learning. Now there is a realization that we can learn from unlabeled data. This is exciting because in Machine Learning it is often the case that it's not the one with the best algorithm who wins, but the one with the most data. If you are able to use unlabeled data then for many problems you essentially have an unlimited source of data. [Hence you're only limited by your computational capacity. Indeed, much of the current work in Ng's lab is devoted to scaling up Deep Learning, for example by using graphics processing units.]

The other attractive side of Deep Learning is the neuroscience link. Deep Learning has more similarities to the way we think the brain "does" intelligence. First it's the learning from unlabeled data. We get far more data from unlabeled data, walking around hearing and seeing, far greater than what we get from parents and teachers. Also we see phenomenas of the brains rise naturally in these algorithms. If you rewire the optic signal to the auditory cortex, the auditory cortex learns to see. If you rewire the optic signal to somatosensory cortex it learns to see as well. It raises the hypothesis that a single underlying algorithms does perception even in different modalities.

JL: **In what way is Deep Learning *not* doing what the brain is doing?**

AN: The computation in a biological neuron is far more complicated than a sigmoid on a linear function [a sigmoid is a non-linear function often added to the output of artificial neurons to enhance the network expression power]. The neuron has less digits of precision than what we use. The brain communicates in spikes while in Deep Learning the model communicates in floating point numbers, and it is not entirely clear what they are analogous to. Also none of these algorithms is a dynamical system, while the brain is a dynamical system continuous through time.

JL: **Could any of these be key elements that will take Deep Learning to the next level?**

AN: It's hard to tell which of the differences between Deep Learning and the brain are fundamental and which are artifacts of the hardware. For example, no one knows if having a physical body, and the ability to see the same thing for multiple perspective is necessary to build a intelligent perceptual system. There is theory that children learn about objects by having binocular stereo vision that helps them do segmentation. Is it possible for a one-eyed organism with no ability to move its head (e.g., a computer) to learn the same sort of vision system? Is feedback and the ability to reach out and touch the world essential? Is attention crucial? These are all open questions, no one knows the answer to that.

NUMENTA: SEQUENCE PREDICTION RATHER THAN STATIC REPRESENTATION

In the Downtown Redwood City office of his company Numenta, Jeff Hawkins and his team are also working on a hierarchical model of perception. But while in Deep Learning (as in Neural Networks), each artificial neuron is not doing much more than to compute linear combinations, Hawkins' neurons behave significantly more like real neurons. They can inhibit each other, form synapses, construct dendritic segments, and receive inputs from both the lateral connections and feed-forward connections. The most important difference, however, is that they can predict their own activation. This allows the model to learn sequences of events—patterns of data across time in addition to space. Thus Hawkins' model not only tries to represent the current input, but also to tell what the next one is going to be.

The company was founded in 2005, shortly after the publication of *On Intelligence.* During the few years of its existence, Numenta has published a number of documents describing its model in detail, down to the level of pseudo-code. In addition, it released a software library and tried to cultivate a community of users. However, during the last five years, while papers on the merits of Deep Learning were accumulating in top AI conferences, Numenta was generating less news than anticipated. As one of the students inspired by Hawkins' refreshing approach, I was curious to know why. I was even more curious about a recent breakthrough that Hawkins had mentioned in the correspondence before our meeting.

Jonathan Laserson: **Jeff, something didn't work the way you expected?**

Jeff Hawkins: Initially, we thought it would be three to five years before we had a commercial application. It took us six years. In January of 2011 we switched from a "research" company to a "product" company. After a year of experimenting with the new algorithms, a business opportunity surfaced and we decided to take it. We are working on a product now, and when it's ready we'll see how it works. But you all will have to wait since we're in stealth mode.

JL: **Tell me about the breakthrough**.

JH: The big advance 18 months ago is about taking temporal sequences of data and forming a stable representation for them, and then using it to make predictions. It has a tight relationship to neuroscience. The new algorithm, which we call the Cortical Learning Algorithm (CLA), is a beautiful mesh between top-down theoretical needs and bottom-up biological detail. They both inform one another. One thing that came out of this is a very detailed model on the role of dendritic segments. Most of the computation in the neuron is in the dendrites.

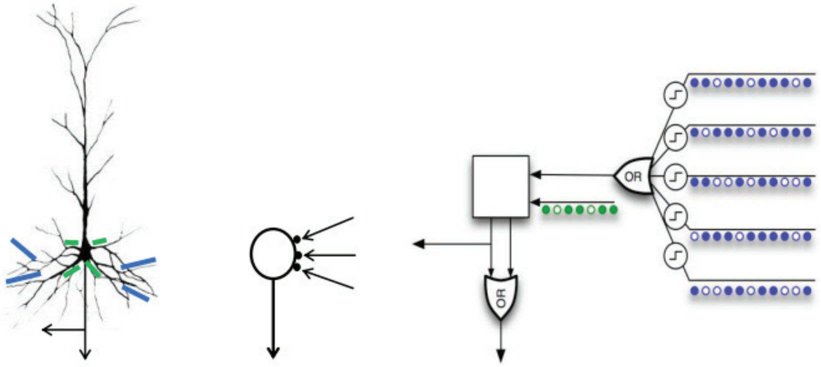
Thanksgiving 2009, the concepts behind CLA started. I remember this because over the holiday I read the book *Dendrites*. Read it cover to cover twice. I had a theory about how neurons will learn sequences and I wanted to see if it is biologically real, so this was a test for the theory. The breakthrough was that the same cells that learn to represent the input have to also learn the sequences. There aren't any other cells—biology tells you that. We spent a year implementing the algorithms that followed from this breakthrough, and at the end of 2010 we set out to build a business out of that, including outside funding.

One exciting thing was to learn that each cell could look at a sub-sample of the cells nearby, and form connections to those that were active prior to this one becoming active. Mathematically you only need to connect to 20 or so of the active cells—that is sufficient to predict your own activity even though the overall pattern you are trying to recognize can consist of hundreds or thousands of cells. Each cell can participate in many different patterns resulting in a distributed sequence memory. That was one key insight, and was different than what we had implemented before.

JL: **What was the crisis that led to the breakthrough?**

JH: We had spent five years trying to figure out how to learn sequences of patterns in complex large data streams. We tried many approaches, most of them from machine learning. At the time, one of the co-founders of the company was Dileep George. He is more of a math guy, and would come up with mathematical techniques, but it turned out to be a really hard problem. Finding temporal structure and stable representation in a messy data stream—that's hard. In the fall of 2009

Figure 4: From left to right—a real neuron, a classic neural network neuron, a Numenta neuron. Dendritic segments in real and Numenta neurons are marked in color. Green segments form feed-forward connections, and blue form lateral connections.



I said, "Let me go back to biology. Here is how I think neurons would do it."

JL: **Isn't it ironic? Earlier you mentioned how AI professors said that studying brains would limit your thinking. It turned out that relying on machine learning methods could limit your thinking.**

JH: You have to do both. You want to understand the concepts of machine learning—it helped seeing why all the other techniques didn't work. You have to have a conceptual framework of the problem you are trying to solve. Then you can look at the neuroscience and take a guess on how to do this.

JL: **How do you know if the changes you are making to the model are good or not?**

JH: There are two categories for the answer: one is to look at neuroscience, and the other is methods for machine intelligence. In the neuroscience realm there are many predictions that we can make, and those can be tested. If our theories explain a vast array of neuroscience observations then it tells us that we're on the right track. In the machine learning world they don't care about that, only how well it works on practical problems. In our case that remains to be seen. To the extent you can solve a problem that no one was able to solve before, people will take notice.

JL: **But you are not trying to optimize any particular task?**

JH: Is your brain optimizing a particular task? There's something called the "no free lunch theorem." It says that no one algorithm is best for everything. Generally, if you take a particular task and you put five Ph.D.s in a room, they will come to a pretty good solution. But from a business perspective that is not a scalable solution. That is not how the brain does it. The neocortex uses a pretty generic algorithm. It's not the best algorithm but it can solve a large class of problems up to a certain level of proficiency.

JL: **A one-size-fits-all algorithm?**

JH: The neocortex is like that, not necessarily the rest of the brain (for example, the retina is very specific). If you are born without sight, your visual cortex becomes sensitive to touch or sound. If I practice how to use a new tool, an area of the brain becomes dedicated to it. Today machine learning is not easy—it has a broken business model. You've got to have a bunch of Stanford graduates to solve problems. The question is if we can make it any easier.

JL: **Do you think that the Deep Learning approach is addressing these issues?**

JH: Conceptually it's similar. I am happy to see the interest in Deep Learning. I was also happy to see the interest in neural networks, but they didn't go far enough. They stopped too soon. The risk with Deep Learning is that they will have quick early success and then they'll stop there, which is what happened with neural networks.

JL: **They stopped too soon?**

JH: Early neural network researchers had success on simple problems, but they didn't continue to evolve the technology. They got hung up on doing better on very simple tasks. Clearly the brain is a neural network, right? But most artificial neural networks are not biological at all. I don't think that approach can succeed with such simple networks. I determined very early that any brain model has to explain how the brain makes huge amounts of predictions. It requires a temporal memory that learns what follows what. It's inherent in the brain. If a neural network has no concept of time, you will not capture a huge portion of what brains do. Most Deep Learning algorithms do not have a concept of time.

JL: **Is it more important to you to understand the brain better or to build better algorithms for AI?**

JH: My No. 1 has always been to understand how the brain works. I want to understand what my brain is, who I am and how my brain works. I wrote in my book about my eye opening experience reading the Scientific American article by Francis Crick in September 1979. I was 22 at the time. Crick said that we have lots of data but "no theoretical framework" for understanding it. I said to myself, "Oh gosh, we can do this. What else could be more interesting to work on in your life?" It became a personal goal, and I've learned so much by now that I feel I've met that goal. There is no reason we can't build a machine that will work like this, and that's exciting. We can build machines that are faster and smarter than humans, and that can solve problems that we have difficulty with. I find discovery of knowledge is the most exciting thing.

FURTHER READING

B.A. Olshausen and D.J. Field. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature 381* (1996),607-609.

Sparse Coding and Deep Learning tutorial http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial

Numenta Current Model http://www.numenta.com/htm-overview/education/HTM_CorticalLearningAlgorithms.pdf

A.W. Roe, S.L. Pallas, Y.H. Kwon and M. Sur. Visual projections routed to the auditory pathway in ferrets: receptive fields of visual neurons in primary auditory cortex. *J. Neurosci. 12* (1992) 3651-3664.

H. Lee, Y. Largman, P. Pham and A.Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *NIPS 2009.*

M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision* (2007).

Biography

Jonathan Laserson is a Ph.D. candidate in the Computer Science Artificial Intelligence Lab at Stanford University. His research interests include probabilistic models, Bayesian inference, and nonparametric priors over domains with unknown complexity. In his current research he is trying to make sense of high-throughput sequencing data from diverse biological environments. He can be contacted at jonil@stanford.edu.

References

[1] Hubel, D.H. and T. N. Wiesel. *Brain and Visual Perception: The Story of a 25-Year Collaboration.* Oxford University Press, 2004.

[2] Computational Vision at Caltech; http://www.vision.caltech.edu/Image_Datasets/Caltech101/