HHS Public Access

Author manuscript

IEEE Spectr. Author manuscript; available in PMC 2019 January 10.

Published in final edited form as:

IEEE Spectr. 2017 March ; 54(3): 32–37. doi:10.1109/MSPEC.2017.7864754.

Deep Learning Reinvents the Hearing Aid:

Finally, wearers of hearing aids can pick out a voice in a crowded room

DeLiang Wang

My mother began to lose her hearing while I was away at college. I would return home to share what I'd learned, and she would lean in to hear. Soon it became difficult for her to hold a conversation if more than one person spoke at a time. Now, even with a hearing aid, she struggles to distinguish the sounds of each voice. When my family visits for dinner, she still pleads with us to speak in turn.

My mother's hardship reflects a classic problem for hearing aid manufacturers. The human auditory system can naturally pick out a voice in a crowded room, but creating a hearing aid that mimics that ability has stumped signal processing specialists, artificial intelligence experts, and audiologists for decades. British cognitive scientist Colin Cherry first dubbed this the "cocktail party problem" (http://www.psypress.co.uk/common/supplementary/184169360x/ch6_194.pdf) in 1953.

More than six decades later, less than 25 percent of people who need a hearing aid (http://www.asha.org/public/hearing/Hearing-Aids-Overview/) actually use one. The greatest frustration among potential users is that a hearing aid cannot distinguish between, for example, a voice and the sound of a passing car if those sounds occur at the same time. The device cranks up the volume on both, creating an incoherent din.

It's time we solve this problem. To produce a better experience for hearing aid wearers, my lab at Ohio State University (http://web.cse.ohio-state.edu/~dwang/pnl/), in Columbus, recently applied machine learning based on deep neural networks to the task of segregating sounds. We have tested multiple versions of a digital filter that not only amplifies sound but can also isolate speech from background noise and automatically adjust the volumes of each separately.

We believe this approach can ultimately restore a hearing-impaired person's comprehension to match—or even exceed—that of someone with normal hearing. In fact, one of our early models (http://web.cse.ohio-state.edu/~dwang/papers/HYWW.jasa13.pdf) boosted, from 10 to 90 percent, the ability of some subjects to understand spoken words obscured by noise. Because it's not necessary for listeners to understand every word in a phrase to gather its meaning, this improvement frequently meant the difference between comprehending a sentence or not.

Without a better hearing aid, the world's hearing will get worse. The World Health Organization estimates that 15 percent of adults, or roughly 766 million people, suffer from hearing loss (http://www.who.int/pbd/deafness/news/Millionslivewithhearingloss.pdf). That number is rising as the population expands and the proportion of older adults becomes larger. And the potential market for an advanced hearing aid isn't limited to people with

hearing loss. Developers could use the technique to improve smartphone speech recognition. Employers could use it to help workers on noisy factory floors, and militaries could equip soldiers to hear one another through the noisy chaos of warfare.

It all adds up to a big potential market. The global US $6 billion hearing aid industry (http://www.marketsandmarkets.com/PressReleases/hearing-aids.asp) is expected to grow at 6 percent every year through 2020, according to the market research firm MarketsandMarkets, in Pune, India. Satisfying all those new customers, though, means finding a way to put the cocktail party problem behind us. At last, deep neural networks are pointing the way forward.

For decades, electrical and computer engineers tried and failed to achieve speech isolation through signal processing. The most popular approach has been to use a voice-activity detector to identify gaps between people's utterances as they speak. In this approach, the system designates the sounds captured within those gaps as "noise." Then, an algorithm subtracts the noise from the original recording—leaving, ideally, noise-free speech.

Unfortunately this technique, known as spectral subtraction (https://www.youtube.com/watch?v=g5oSqiZIc_w), is notorious for removing too much speech or too little noise. Too often, what results is an unpleasant artifact (called musical noise) that makes the audio sound as if it were recorded underwater. The problems are so serious that even after many years of development, this method does little or nothing to improve people's ability to recognize speech in noisy environments.

I realized we had to take a different approach. We began with a theory from Albert Bregman, (http://webpages.mcgill.ca/staff/Group2/abregm1/web/) a psychologist at McGill University in Montreal, Canada, who proposed in 1990 that the human auditory system organizes sounds into distinct streams. A stream essentially corresponds to sound emitted from a single source, such as a nearby friend. Each sound stream is unique in its pitch, volume, and the direction from which it comes.

Altogether, many streams—such as that friend speaking over the roar of a hockey game—make up what Bregman calls an "auditory scene." (http://webpages.mcgill.ca/staff/group2/abregm1/web/pdf/2004_%20Encyclopedia-Soc-Behav-Sci.pdf) If sounds share the same frequency band at the same time, the loudest sound in a scene overpowers the others—a useful principle known as auditory masking. For example, someone may not notice a clock ticking in the corner of the room if rain is pattering on the roof. This principle, among others, is exploited in MP3 files to shrink the files to one-tenth of their original size by removing masked sounds (such as the ticking clock, in this case) without users noticing the omission.

Recalling Bregman's work, we wondered if we could build a filter to determine whether one sound stream dominates others at a given moment inside a specific frequency band. Psychoacousticians, who study sound perception, divide the average human's hearing range into about two dozen bands between 20 hertz and 20,000 Hz (https://www.youtube.com/watch?v=qNf9nzvnd1k). We wanted a filter to tell us whether a sound stream containing

speech or noise was stronger at certain times within these bands, as a first step toward separating the two.

My lab was the first, in 2001, to design such a filter, which labels sound streams as dominated by either speech or noise. With this filter, we would later develop a machine-learning program that separates speech from other sounds based on a few distinguishing features, such as amplitude (loudness), harmonic structure (the particular arrangement of tones), and onset (when a particular sound begins relative to others).

This original filter was what we called the ideal binary mask. It labels noise and speech that it finds within segments of sound called time-frequency units, which designate a particular brief interval within a specific frequency band. The filter analyzes each time-frequency unit in a sample of noisy speech and marks each as either 1 or 0. It records a 1 if the “target” sound (in this case, speech) is louder than noise, and a 0 if the target sound is softer. The result is a set of 1s and 0s that represent the dominance of noise or speech within a sample. Then, the filter tosses out all units labeled 0 and reconstructs the speech from those that scored 1. To reconstruct an intelligible sentence from noisy speech, a certain percentage of time-frequency units must be labeled 1.

We began testing the ideal binary mask in 2006 with the U.S. Air Force Research Laboratory, in Ohio. Around the same time, a team from Syracuse University (http://www.ncbi.nlm.nih.gov/pubmed/16957499), in New York, independently evaluated the ideal binary mask. In those trials, the filter helped people with a hearing impairment and also listeners with normal hearing to better understand sentences mixed with noise.

We had, basically, created a speech filter that performed flawlessly in the lab. But this filter enjoyed an unrealistic advantage. By design, we had provided it with samples of speech and noise separately and then tested it using mixtures of those same samples. Because it had been given the answers (that’s why it’s “ideal”), the filter knew when the speech was louder than the background noise. A practical speech filter must, entirely on its own and on the fly, separate a voice from the noise in a room.

Nevertheless, the fact that the ideal binary mask dramatically improved speech comprehension for both hearing-impaired listeners and those with normal hearing had a profound implication. It demonstrated that the technique of classification, a form of supervised learning, could be employed to approximate the ideal binary mask as a way of separating speech from noise. With classification, a machine mimics human learning, in effect, by completing exercises, receiving feedback, and drawing and remembering lessons from its experiences. That’s essentially the same way people learn from a young age to treat apples as a class distinct from oranges.

In the following years, my lab made the first attempt to approximate the ideal binary mask through classification. At about the same time we were developing our original classifier, a group at Carnegie Mellon University (http://www.cmu.edu/), in Pittsburgh, devised their own method, based on machine learning, to classify time-frequency units for another purpose: to improve automatic speech recognition. Later, a group at the University of Texas at Dallas led by the late Philipos Loizou used a different (http://ecs.utdallas.edu/loizou/

cimplants/GMM_intelligib_sept09.pdf) classification method. It became the first to show meaningful improvement in speech intelligibility for people with normal hearing by relying on only monaural features (as opposed to the binaural ones captured by two ears).

But these early machine-learning methods applied classification techniques that were not powerful or accurate enough to help hearing aid wearers. They could not yet handle the complex and unpredictable mixture of noises and voices that occur in the world. In order to do that, we would need something far more powerful.

Having demonstrated promising initial results with our early classification algorithms, we decided to take the next logical step—to improve the system so it could function in noisy real-world environments, and without training for specific noises and sentences. This challenge prompted us to try to do something that had never been done before: build a machine-learning program (http://web.cse.ohio-state.edu/~dwang/papers/HYWW.jasa13.pdf) that would run on a neural network and separate speech from noise after undergoing a sophisticated training process. The program would use the ideal binary mask to guide the training of the neural network. And it worked. In a study involving 24 test subjects, we demonstrated that this program could boost the comprehension of hearing-impaired people by about 50 percent.

Basically, a neural network is a software system constructed of relatively simple elements that can achieve complex levels of processing by working together. (The system's structure is roughly modeled on how neurons and their networks work in the brain.) When presented with new examples, neural networks, like human brains, can "learn" by adjusting the weights of their connections.

Neural networks come in many shapes and sizes and with varying degrees of complexity. Deep neural networks are defined as having at least two "hidden" processing layers, which are not directly connected to a system's input or output. Each hidden layer refines the results fed to it by previous layers, adding in new considerations based on prior knowledge.

For example, a program designed to verify a customer's signature (http://www.cedar.buffalo.edu/~srihari/papers/ICGVIP2006-sig.pdf) might begin by comparing a new signature to a sample included in a training database. However, that program also knows from its training that the new signature does not need to precisely match the original. Other layers can determine if the new signature shares certain qualities that tend to remain consistent in a person's signature, such as the angle of slant, or the failure to dot the letter *i*.

To build our own deep neural network, we began by writing algorithms to extract features that could distinguish voices from noise based on common changes in amplitude, frequency, and the modulations of each. We identified dozens of attributes that could help our program discriminate between speech and noise to some extent, and we used all 85 of them to make the algorithms as powerful as possible. Among the most important attributes we identified were the frequencies of the sounds and their intensities (loud or soft).

Next, we trained the deep neural network to use these 85 attributes to distinguish speech from noise. This training occurred in two phases: First, we set the program's parameters

through unsupervised learning. This means we loaded many examples of the attributes into the program in order to prime it for the types of signals it would later have to classify on the fly.

Then we used samples of noisy speech and their corresponding results on the ideal binary mask to complete the second phase of training, which was the supervised learning. In particular, the set of 1s and 0s that make up the ideal binary mask was like an answer sheet that we used to test and improve our program's ability to separate speech from noise. For each new sample, the program would extract a set of attributes from the noisy speech. Then, after analyzing these attributes—frequencies, intensities, and so on—the filter performed a provisional classification—was it speech? was it noise?—and compared the result to what the ideal binary mask would determine in the same situation. If the result was different from the 1s and 0s within our perfect binary mask filter, we tweaked the neural network's parameters accordingly, so that the network would produce results closer to the 1s and 0s of the ideal binary mask on its next try.

To make these adjustments, we first calculated the error of the neural network, measured as the discrepancy between the ideal binary mask and the result at the neural network's final layer, which is known as the output layer. Once we computed this error, we would then use it to change the weights of the neural network's connections so that if the same classification was carried out again, the discrepancy would be reduced. The training of the neural network consisted of performing this procedure thousands of times.

One important refinement along the way was to build a second deep neural network that would be fed by the first one and fine-tune its results. While that first network had focused on labeling attributes within each individual time-frequency unit, the second network would examine the attributes of several units near a particular one. To understand why this helped, consider the following analogy: If the first network was like looking at the rooms of a house for sale, the second network was like walking around in the surrounding neighborhood. In other words, the second network provided the first network with extra context about the speech and noise it processed and further improved its classification accuracy. For example, a syllable may span many time-frequency units, but the background noise could change abruptly while it was being spoken. In our case, having contextual clues could help the program to more accurately separate speech from noise within the syllable.

At the end of the supervised training, the deep-neural-network classifier proved to be far superior to earlier methods at separating speech from noise. In fact, this algorithm was the first, of any technique relying on monaural techniques, to achieve major improvements in hearing-impaired listeners' ability to make sense of spoken phrases obscured by noise.

To test it with human subjects, we asked 12 hearing-impaired people and 12 with normal hearing to listen through headphones to samples of noisy sentences. The samples were in pairs: first the speech and noise occurring together, and then the same sample *after* it had been processed by our program running on the deep neural networks. The sentences, which included phrases such as "It's getting cold in here" and "They ate the lemon pie," were cluttered by two types of noise—a steady humming noise and the babble of many people

talking at once. The steady noise was similar to the sound of a refrigerator running, in which the audio waves are repetitive and the shape of the frequency spectrum does not change over time. We created the noisy background babble by adding utterances from four male and four female speakers, to mimic a cocktail party.

People in both groups showed a big improvement (http://web.cse.ohio-state.edu/~dwang/papers/HYWW.jasa13.pdf) in their ability to comprehend sentences amid noise after the sentences were processed through our program. People with hearing impairment could decipher only 29 percent of words muddled by babble without the program, but they understood 84 percent after the processing. Several went from understanding only 10 percent of words in the original sample to comprehending around 90 percent with the program. There were similar gains for the steady-noise scenario with hearing-impaired subjects—they went from 36 percent to 82 percent comprehension.

Even people with normal hearing were able to better understand noisy sentences, which means our program could someday help far more people than we originally anticipated. Listeners with normal hearing understood 37 percent of the words spoken amid steady noise without the program, and 80 percent with it. For the babble, they improved from 42 percent of words to 78 percent.

One of the most intriguing results of our experiment came when we asked, Could people with hearing impairment who are assisted by our program actually outperform those with normal hearing? Remarkably, the answer is yes. Listeners with hearing impairment who used our program understood nearly 20 percent more words in the babble and about 15 percent more words in steady noise than those with normal hearing who relied solely on their own auditory system to separate speech from noise. With these results, our program built from deep neural networks has come the closest to solving the cocktail party problem of any effort to date.

There are, of course, limits to the program’s abilities. For example, in our samples, the type of noise that obscured speech was still quite similar to the type of noise the program had been trained to classify. To function in real life, a program will need to quickly learn to filter out many types of noise, including types different from the ones it has already encountered. For example, the hiss of a ventilation system is different from the hum of a refrigerator compressor. Also, the noisy samples we used did not feature reverberations from the walls and objects in a room, which compounds the noise problem at any cocktail party.

Since we published those early results, we’ve purchased a database of sound effects designed for filmmakers and used its 10,000 noises to further train the program. This year, we found that the retrained program (http://web.cse.ohio-state.edu/~dwang/papers/CWYWH.jasa16.pdf) [PDF] could encounter completely new noises and achieve meaningful improvement in comprehension for both hearing-impaired listeners and those with normal hearing. Now, with funding from the National Institute on Deafness and Other Communication Disorders, (https://www.nidcd.nih.gov/) we are pushing the program to operate in more environments and test it with more listeners who have hearing loss.

Eventually, we believe the program could be trained on powerful computers and embedded directly into a hearing aid, or paired with a smartphone via a wireless link, such as Bluetooth, to feed the processed signal in real time to an earpiece. Periodically, hearing aid wearers could update their devices as manufacturers release new versions after retraining the system on new noises. We have filed several patents for the technique and are working with partners to commercialize it, including Starkey Hearing Technologies (http://www.starkey.com/), in Eden Prairie, Minn., a leading hearing aid manufacturer in the United States.

With this approach, the cocktail party problem does not look nearly as daunting as it did just a couple of years ago. We, and others, can now create software that we expect will ultimately overcome it through more extensive training in more noisy situations. In fact, I suspect this process is similar to the way children learn to separate speech from noise early in life—through repeated exposure to a wide range of both. With more experience, the approach can only get better. That's the beauty of it. As is also true for a youngster, time is on our side.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Biography

DeLiang Wang (http://web.cse.ohio-state.edu/~dwang/) is a professor in the department of computer science and engineering and the Center for Cognitive and Brain Sciences at Ohio State University in Columbus, Ohio. He is a University Distinguished Scholar and directs OSU's Perception and Neurodynamics Laboratory (http://web.cse.ohio-state.edu/~dwang/pnl/), which focuses on developing algorithms to solve problems related to machine perception. Wang received his Ph.D. in computer science from the University of Southern California, Los Angeles, in 1991 after completing his bachelor's and master's degrees at Peking University, in Beijing. He is co-editor in chief of the journal *Neural Networks* (http://www.journals.elsevier.com/neural-networks).

Samples of Speech

Before and After Filtering

provided by DeLiang Wang, OSU

The Man Called the Police

It’s Getting Cold in Here

They Ate the Lemon Pie

It’s Time to Go to Bed

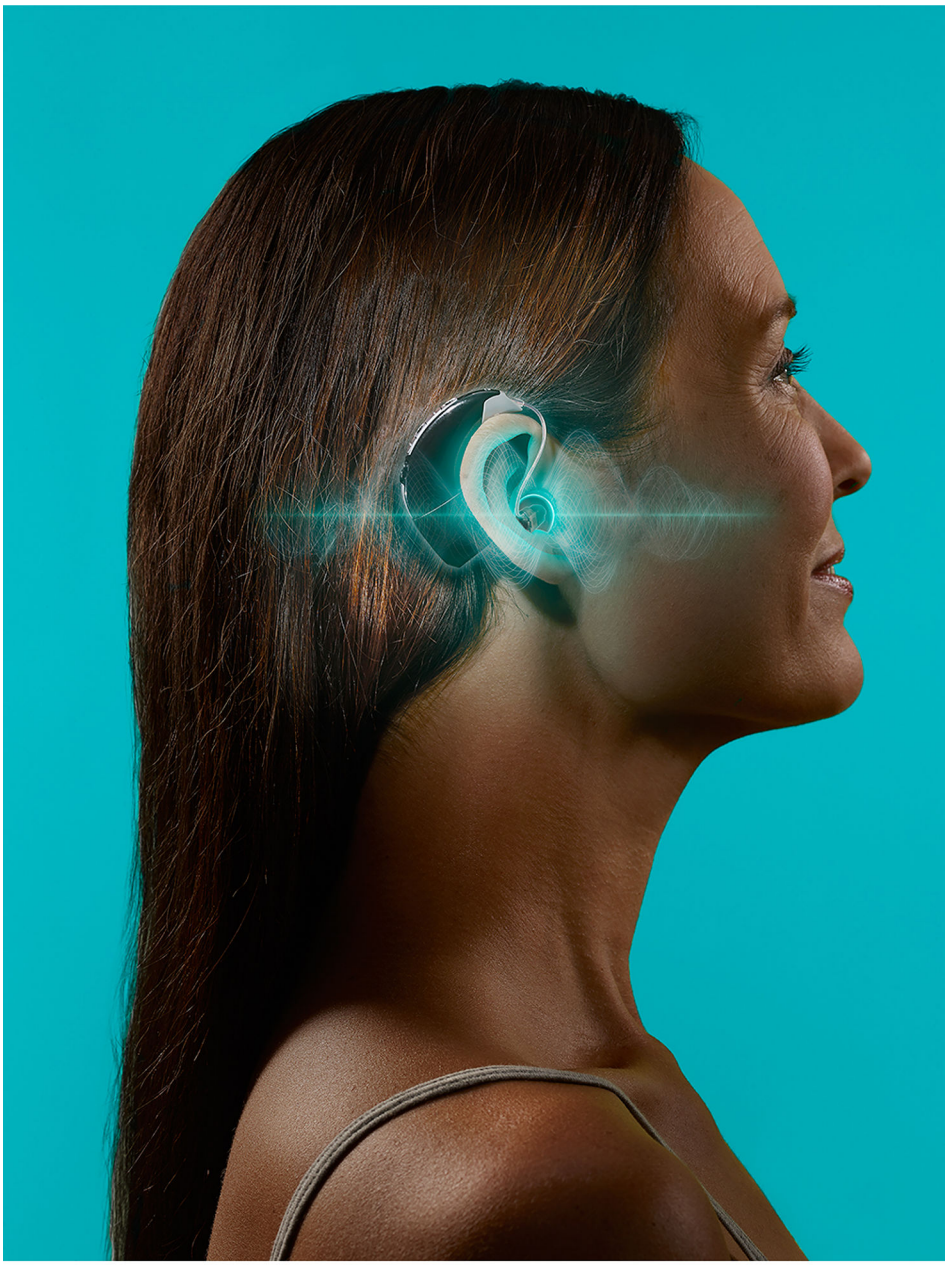

Photo: Dan Saelinger/Trunk Archive

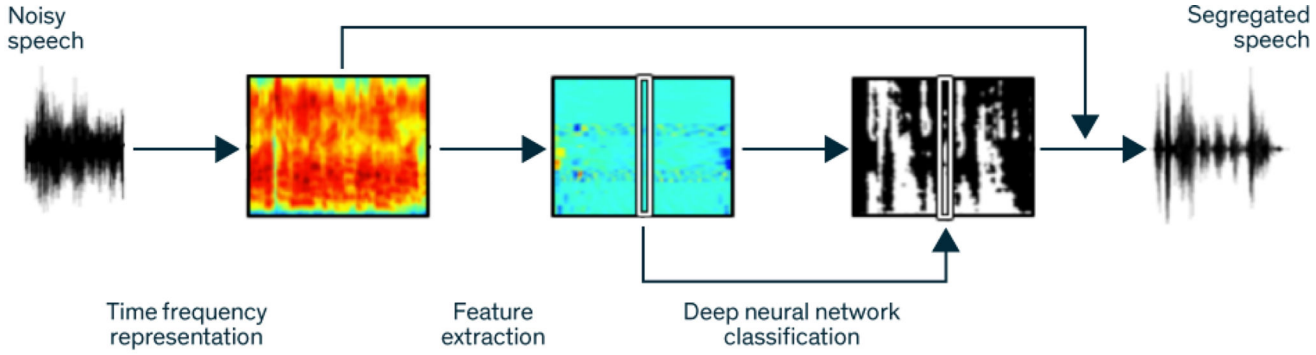

Illustration: Erik Vrielink

Clean Speech: To separate speech from noise, a machine learning program breaks a noisy speech sample into a collection of elements called time-frequency units. Next, it analyzes these units to extract 85 features known to distinguish speech from other sounds. Then, the program feeds the features into a deep neural network trained to classify the units as speech or not based on past experience with similar samples. Lastly, the program applies a digital filter that tosses out all the nonspeech units to leave only separated speech.

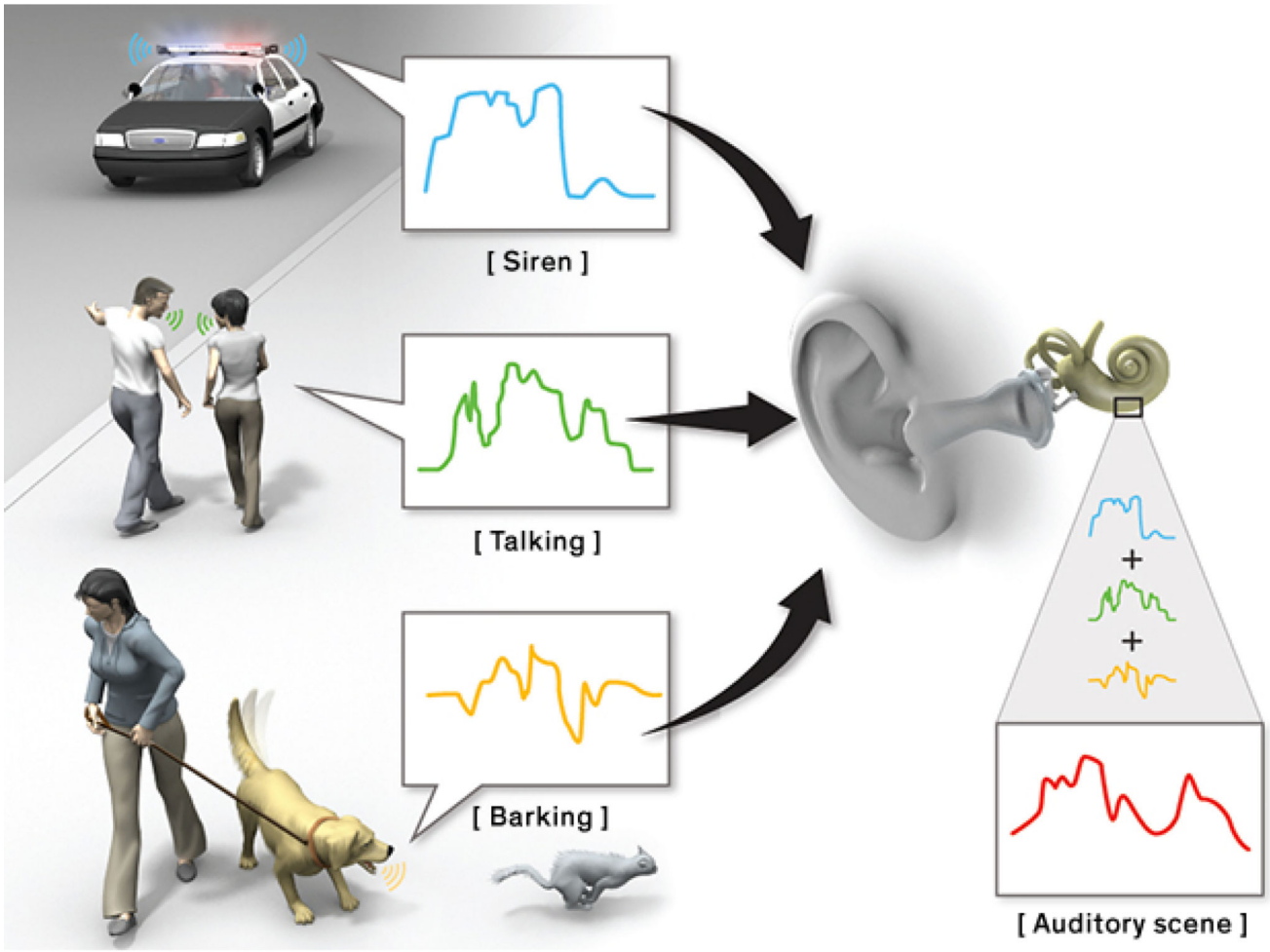

Illustration: Emily Cooper

A Noisy World: Thanks in part to its odd shape, the human ear captures many sound streams at once. A stream is all the sound waves that emanate from a single source, such as a dog. Together, these streams make up an auditory scene (barking + siren + talking).

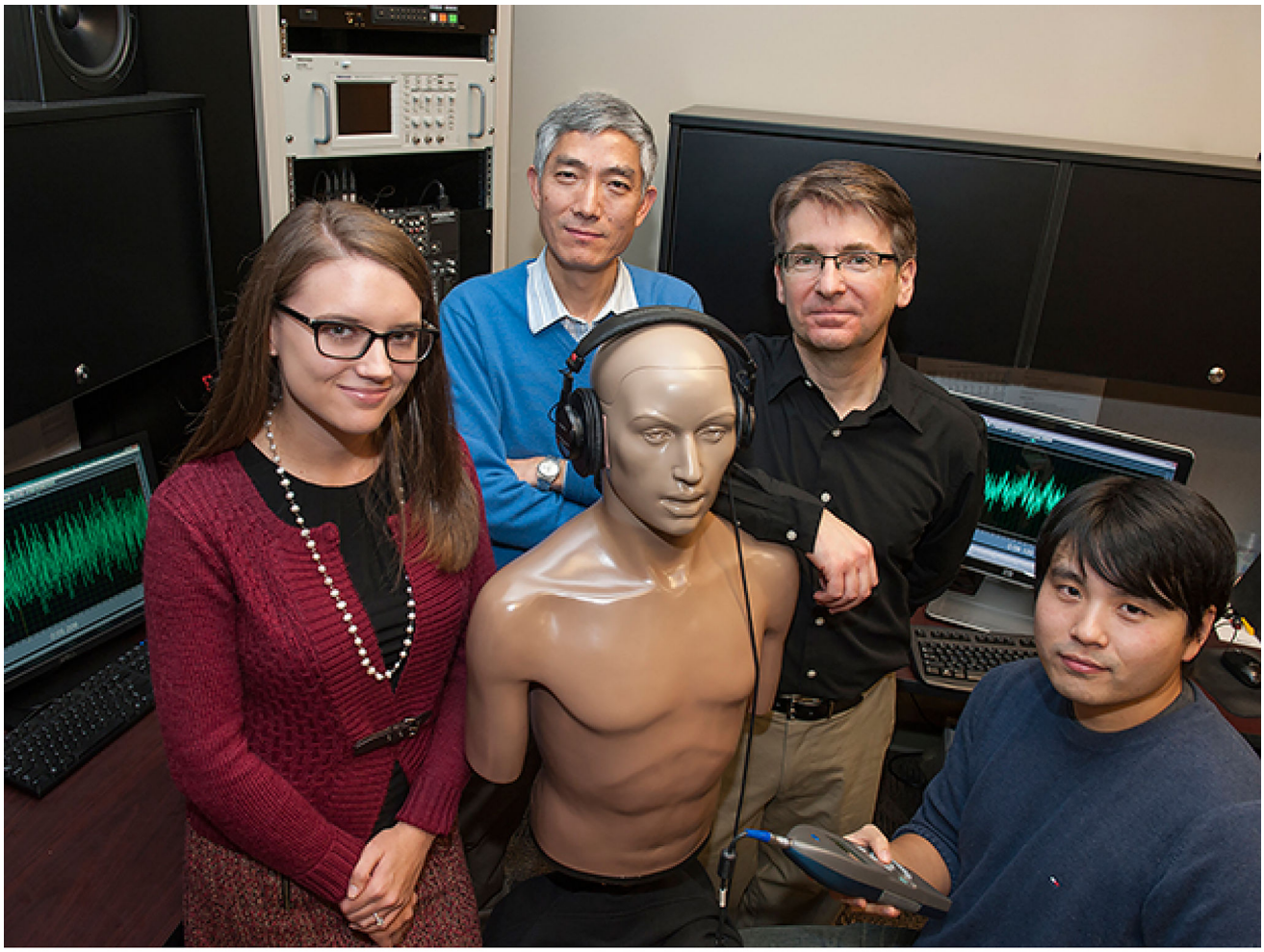

Photo: Jo McCulty

Listen Up: In this 2013 photo, a machine-learning program for speech separation built on deep neural networks is tested by [from left to right] Sarah Yoho, DeLiang Wang, Eric Healy, and Yuxuan Wang of Ohio State University.

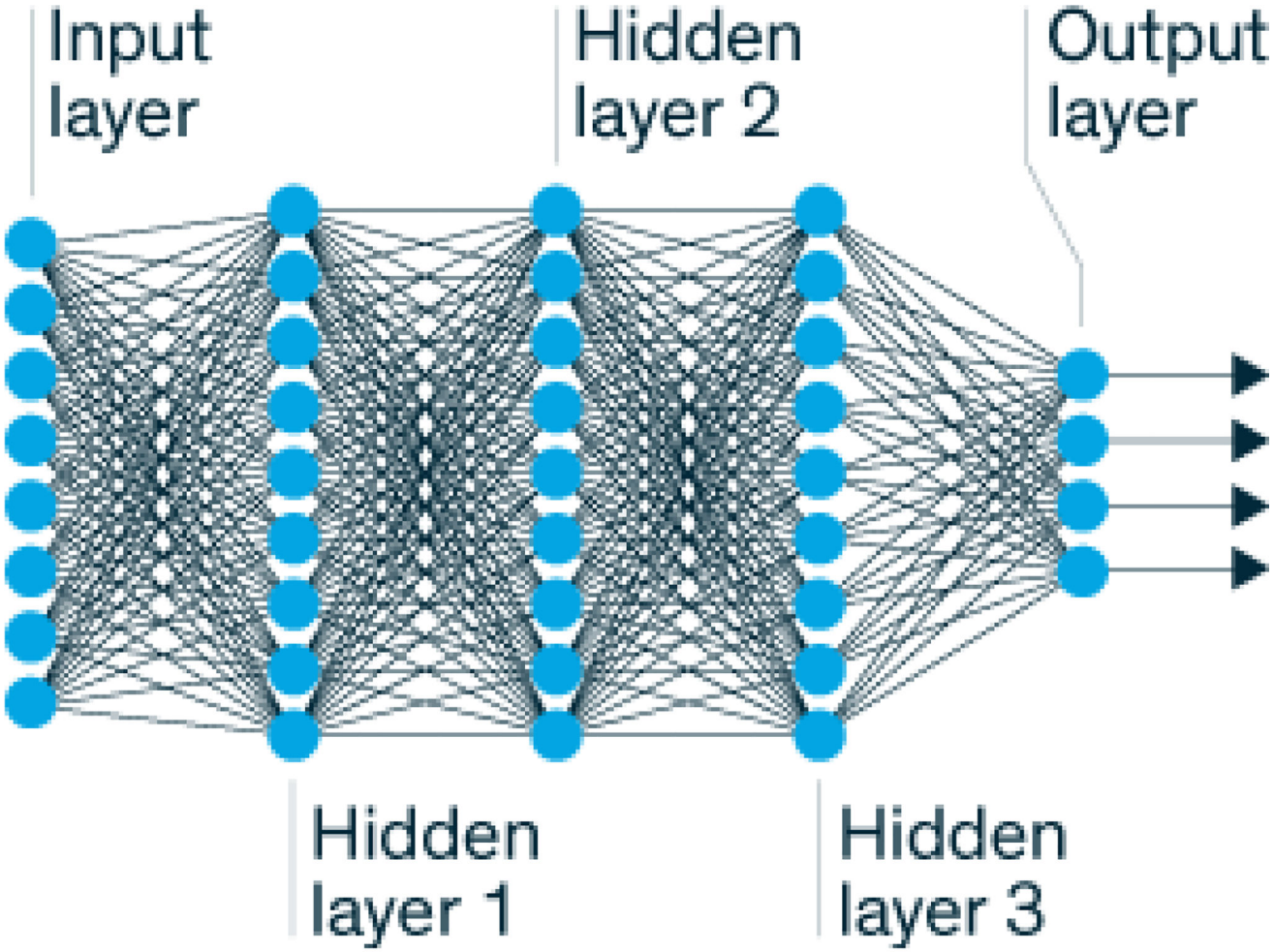

Illustration: Erik Vrielink

Smart Layers: A deep neural network consists of two or more processing layers in between the input layer, through which information is fed into the system [left], and the output layer, which reveals the results [right]. To improve performance, researchers can adjust the system's parameters and tweak the connections between layers.