DOI:10.1145/3448250

How can neural networks learn the rich internal representations required for difficult tasks such as recognizing objects or understanding language?

BY YOSHUA BENGIO, YANN LECUN, AND GEOFFREY HINTON

Deep Learning for AI

TURING LECTURE

Yoshua Bengio, Yann LeCun, and Geoffrey Hinton are recipients of the 2018 ACM A.M. Turing Award for breakthroughs that have made deep neural networks a critical component of computing.

RESEARCH ON ARTIFICIAL neural networks was motivated by the observation that human intelligence emerges from highly parallel networks of relatively simple, non-linear neurons that learn by adjusting the strengths of their connections. This observation leads to a central computational question: How is it possible for networks of this general kind to learn the complicated internal representations that are required for difficult tasks such as recognizing objects or understanding language? Deep learning seeks to answer this question by using many layers of activity vectors as representations and learning the connection strengths that give rise to these vectors by following the stochastic gradient of an objective function that measures how well the network is performing. It is very surprising that such a conceptually simple approach has proved to be so effective when applied to large training sets using huge amounts of computation and it appears that a key ingredient is depth: shallow networks simply do not work as well.

We reviewed the basic concepts and some of the breakthrough achievements of deep learning several years ago.[63] Here we briefly describe the origins of deep learning, describe a few of the more recent advances, and discuss some of the future challenges. These challenges include learning with little or no external supervision, coping with test examples that come from a different distribution than the training examples, and using the deep learning approach for tasks that humans solve by using a deliberate sequence of steps which we attend to consciously—tasks that Kahneman[56] calls *system 2* tasks as opposed to *system 1* tasks like object recognition or immediate natural language understanding, which generally feel effortless.

From Hand-Coded Symbolic Expressions to Learned Distributed Representations

There are two quite different paradigms for AI. Put simply, the logic-inspired paradigm views sequential reasoning as the essence of intelligence and aims to implement reasoning in computers using hand-designed rules of inference that operate on hand-designed symbolic expressions that formalize knowledge. The brain-inspired paradigm views learning representations from data as the essence of intelligence and aims to implement learning by hand-designing or evolving rules for modifying the connec-

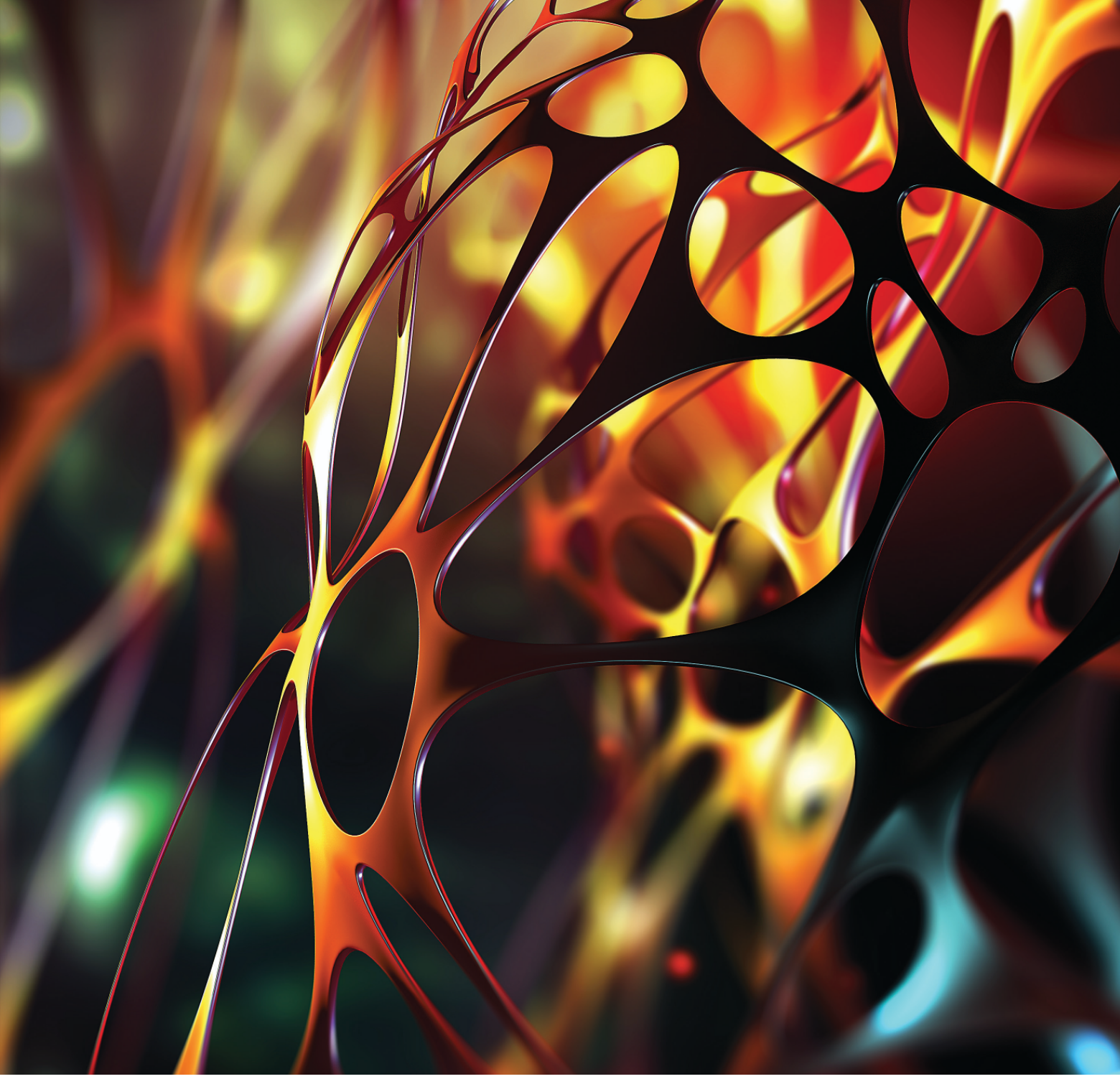

IMAGE BY YURCHANKA SIARHEI

tion strengths in simulated networks of artificial neurons.

In the logic-inspired paradigm, a symbol has no meaningful internal structure: Its meaning resides in its relationships to other symbols which can be represented by a set of symbolic expressions or by a relational graph. By contrast, in the brain-inspired paradigm the external symbols that are used for communication are converted into internal vectors of neural activity and these vectors have a rich similarity structure. Activity vectors can be used to model the structure inherent in a set of symbol strings by learning appropriate activity vectors for each symbol and learning non-linear transformations that allow the activity vectors that correspond to missing elements of a symbol string to be filled in. This was first demonstrated in Rumelhart et al.[74] on toy data and then by Bengio et al.[14] on real sentences. A very impressive recent demonstration is BERT,[22] which also exploits self-attention to dynamically connect groups of units, as described later.

The main advantage of using vectors of neural activity to represent concepts and weight matrices to capture relationships between concepts is that this leads to automatic generalization. If Tuesday and Thursday are represented by very similar vectors, they will have very similar causal effects on other vectors of neural activity. This facilitates analogical reasoning and suggests that immediate, intuitive analogical reasoning is our primary mode of reasoning, with logical sequential reasoning being a much later development,[56] which we will discuss.

The Rise of Deep Learning

Deep learning re-energized neural network research in the early 2000s by introducing a few elements which made it easy to train deeper networks. The emergence of GPUs and the availability of large datasets were key enablers of deep learning and they were greatly enhanced by the development of open source, flexible software platforms with automatic differentiation such as Theano,[16] Torch,[25] Caffe,[55] TensorFlow,[1] and PyTorch.[71] This made it easy to train complicated deep nets and to reuse the latest models and their building blocks. But the composition of more layers is what allowed more complex non-linearities and achieved surprisingly good results in perception tasks, as summarized here.

Why depth? Although the intuition that deeper neural networks could be more powerful pre-dated modern deep learning techniques,[82] it was a series of advances in both architecture and training procedures,[15,35,48] which ushered in the remarkable advances which are associated with the rise of deep learning. But why might deeper networks generalize better for the kinds of input-output relationships we are interested in modeling? It is important to realize that it is not simply a question of having more parameters, since deep networks often generalize better than shallow networks with the same number of parameters.[15] The practice confirms this. The most popular class of convolutional net architecture for computer vision is the ResNet family[43] of which the most common representative, ResNet-50 has 50 layers. Other ingredients not mentioned in this article but which turned out to be very useful include image deformations, dropout,[51] and batch normalization.[53]

We believe that deep networks excel because they exploit a particular form of compositionality in which features in one layer are combined in many different ways to create more abstract features in the next layer.

For tasks like perception, this kind of compositionality works very well and there is strong evidence that it is used by biological perceptual systems.[83]

Unsupervised pre-training. When the number of labeled training examples is small compared with the complexity of the neural network required to perform the task, it makes sense to start by using some other source of information to create layers of feature detectors and then to fine-tune these feature detectors using the limited supply of labels. In transfer learning, the source of information is another supervised learning task that has plentiful labels. But it is also possible to create layers of feature detectors without using any labels at all by stacking auto-encoders.[15,50,59]

First, we learn a layer of feature detectors whose activities allow us to reconstruct the input. Then we learn a second layer of feature detectors whose activities allow us to reconstruct the activities of the first layer of feature detectors. After learning several hidden layers in this way, we then try to predict the label from the activities in the last hidden layer and we backpropagate the errors through all of the layers in order to fine-tune the feature detectors that were initially discovered without using the precious information in the labels. The pre-training may well extract all sorts of structure that is irrelevant to the final classification but, in the regime where computation is cheap and labeled data is expensive, this is fine so long as the pre-training transforms the input into a representation that makes classification easier.

In addition to improving generalization, unsupervised pre-training initializes the weights in such a way that it is easy to fine-tune a deep neural network with backpropagation. The effect of pre-training on *optimization* was historically important for overcoming the accepted wisdom that deep nets were hard to train, but it is much less relevant now that people use rectified linear units (see next section) and residual connections.[43] However, the effect of pre-training on *generalization* has proved to be very important. It makes it possible to train very large models by leveraging large quantities of unlabeled data, for example, in natural language processing, for which huge corpora are available.[26,32] The general principle of pre-training and fine-tuning has turned out to be an important tool in the deep learning toolbox, for example, when it comes to transfer learning or even as an ingredient of modern meta-learning.[33]

The mysterious success of rectified linear units. The early successes of deep networks involved unsupervised pre-training of layers of units that used the logistic sigmoid nonlinearity or the closely related hyperbolic tangent. Rectified linear units had long been hypothesized in neuroscience[29] and already used in some variants of RBMs[70] and convolutional neural networks.[54] It was an unexpected and pleasant surprise to discover[35] that rectifying nonlinearities (now called ReLUs, with many modern variants) made it easy to train deep networks by backprop and stochastic gradient descent, without the need for layerwise pre-training. This was one of the technical advances that enabled deep learning to outperform previous methods for object recognition,[60] as outlined here.

Breakthroughs in speech and object recognition. An acoustic model converts a representation of the sound wave into a probability distribution over fragments of phonemes. Heroic efforts by Robinson[72] using transputers and by Morgan et al.[69] using DSP chips had already shown that, with sufficient processing power, neural networks were competitive with the state of the art for acoustic modeling. In 2009, two graduate students[68] using Nvidia GPUs showed that pre-trained deep neural nets could slightly outperform the SOTA on the TIMIT dataset. This result reignited the interest of several leading speech groups in neural networks. In 2010, essentially the same deep network was shown to beat the SOTA for large vocabulary speech recognition without requiring speaker-dependent training[28,46] and by 2012, Google had engineered a production version that significantly improved voice search on Android. This was an early demonstration of the disruptive power of deep learning.

At about the same time, deep learning scored a dramatic victory in the 2012 ImageNet competition, almost halving the error rate for recognizing a thousand different classes of object in natural images.[60] The keys to this victory were the major effort by Fei-Fei Li and her collaborators in collecting more than a million labeled images[31] for the training set and the very efficient use of multiple GPUs by Alex Krizhevsky. Current hardware, including GPUs, encourages the use of large mini-batches in order to amortize the cost of fetching a weight from memory

across many uses of that weight. Pure online stochastic gradient descent which uses each weight once converges faster and future hardware may just use weights in place rather than fetching them from memory.

The deep convolutional neural net contained a few novelties such as the use of ReLUs to make learning faster and the use of dropout to prevent overfitting, but it was basically just a feedforward convolutional neural net of the kind that Yann LeCun and his collaborators had been developing for many years.[64,65] The response of the computer vision community to this breakthrough was admirable. Given this incontrovertible evidence of the superiority of convolutional neural nets, the community rapidly abandoned previous hand-engineered approaches and switched to deep learning.

Recent Advances

Here we selectively touch on some of the more recent advances in deep learning, clearly leaving out many important subjects, such as deep reinforcement learning, graph neural networks and meta-learning.

Soft attention and the transformer architecture. A significant development in deep learning, especially when it comes to sequential processing, is the use of multiplicative interactions, particularly in the form of soft attention.[7,32,39,78] This is a transformative addition to the neural net toolbox, in that it changes neural nets from purely vector transformation machines into architectures which can dynamically choose which inputs they operate on, and can store information in differentiable associative memories. A key property of such architectures is that they can effectively operate on different kinds of data structures including sets and graphs.

Soft attention can be used by modules in a layer to dynamically select which vectors from the previous layer they will combine to compute their outputs. This can serve to make the output independent of the order in which the inputs are presented (treating them as a set) or to use relationships between different inputs (treating them as a graph).

The transformer architecture,[85] which has become the dominant architecture in many applications, stacks many layers of "self-attention" modules. Each module in a layer uses a scalar product to compute the match between its query vector and the key vectors of other modules in that layer. The matches are normalized to sum to 1, and the resulting scalar coefficients are then used to form a convex combination of the value vectors produced by the other modules in the previous layer. The resulting vector forms an input for a module of the next stage of computation. Modules can be made multi-headed so that each module computes several different query, key and value vectors, thus making it possible for each module to have several distinct inputs, each selected from the previous stage modules in a different way. The order and number of modules does not matter in this operation, making it possible to operate on sets of vectors rather than single vectors as in traditional neural networks. For instance, a language translation system, when producing a word in the output sentence, can choose to pay attention to the corresponding group of words in the input sentence, independently of their position in the text. While multiplicative gating is an old idea for such things as coordinate transforms[44] and powerful forms of recurrent networks,[52] its recent forms have made it mainstream. Another way to think about attention mechanisms is that they make it possible to dynamically route information through appropriately selected modules and combine these modules in potentially novel ways for improved out-of-distribution generalization.[38]

We believe that deep networks excel because they exploit a particular form of compositionality in which features in one layer are combined in many different ways to create more abstract features in the next layer.

Transformers have produced dramatic performance improvements that have revolutionized natural language processing,[27,32] and they are now being used routinely in industry. These systems are all pre-trained in a self-supervised manner to predict missing words in a segment of text.

Perhaps more surprisingly, transformers have been used successfully to solve integral and differential equations symbolically.[62] A very promising recent trend uses transformers on top of convolutional nets for object detection and localization in images with state-of-the-art performance.[19] The transformer performs post-processing and object-based reasoning in a differ-

entiable manner, enabling the system to be trained end-to-end.

Unsupervised and self-supervised learning. Supervised learning, while successful in a wide variety of tasks, typically requires a large amount of human-labeled data. Similarly, when reinforcement learning is based only on rewards, it requires a very large number of interactions. These learning methods tend to produce task-specific, specialized systems that are often brittle outside of the narrow domain they have been trained on. Reducing the number of human-labeled samples or interactions with the world that are required to learn a task and increasing the out-of-domain robustness is of crucial importance for applications such as low-resource language translation, medical image analysis, autonomous driving, and content filtering.

Humans and animals seem to be able to learn massive amounts of background knowledge about the world, largely by observation, in a task-independent manner. This knowledge underpins common sense and allows humans to learn complex tasks, such as driving, with just a few hours of practice. A key question for the future of AI is how do humans learn so much from observation alone?

In supervised learning, a label for one of N categories conveys, on average, at most $log_2(N)$ bits of information about the world. In model-free reinforcement learning, a reward similarly conveys only a few bits of information. In contrast, audio, images and video are high-bandwidth modalities that implicitly convey large amounts of information about the structure of the world. This motivates a form of prediction or reconstruction called self-supervised learning which is training to "fill in the blanks" by predicting masked or corrupted portions of the data. Self-supervised learning has been very successful for training transformers to extract vectors that capture the context-dependent meaning of a word or word fragment and these vectors work very well for downstream tasks.

For text, the transformer is trained to predict missing words from a discrete set of possibilities. But in high-dimensional continuous domains such as video, the set of plausible continuations of a particular video segment is large and complex and representing the distribution of plausible continuations properly is essentially an unsolved problem.

A key question for the future of AI is how do humans learn so much from observation alone?

Contrastive learning. One way to approach this problem is through latent variable models that assign an energy (that is, a badness) to examples of a video and a possible continuation.[a]

Given an input video X and a proposed continuation Y, we want a model to indicate whether Y is compatible with X by using an energy function $E(X, Y)$ which takes low values when X and Y are compatible, and higher values otherwise.

$E(X, Y)$ can be computed by a deep neural net which, for a given X, is trained in a contrastive way to give a low energy to values Y that are compatible with X (such as examples of (X, Y) pairs from a training set), and high energy to other values of Y that are incompatible with X. For a given X, inference consists in finding one $\check{Y}$ that minimizes $E(X, Y)$ or perhaps sampling from the Ys that have low values of $E(X, Y)$. This energy-based approach to representing the way Y depends on X makes it possible to model a diverse, multi-modal set of plausible continuations.

The key difficulty with contrastive learning is to pick good "negative" samples: suitable points Y whose energy will be pushed up. When the set of possible negative examples is not too large, we can just consider them all. This is what a softmax does, so in this case contrastive learning reduces to standard supervised or self- supervised learning over a finite discrete set of symbols. But in a real-valued high-dimensional space, there are far too many ways a vector $\hat{Y}$ could be different from Y and to improve the model we need to focus on those Ys that should have high energy but currently have low energy. Early methods to pick negative samples were based on Monte-Carlo methods, such as contrastive divergence for restricted Boltzmann machines[48] and noise-contrastive estimation.[41]

Generative Adversarial Networks (GANs)[36] train a generative neural net to produce contrastive samples by apply-

a As Gibbs pointed out, if energies are defined so that they add for independent systems, they must correspond to negative log probabilities in any probabilistic interpretation.

ing a neural network to latent samples from a known distribution (for example, a Gaussian). The generator trains itself to produce outputs $\hat{Y}$ to which the model gives low energy $E(\hat{Y})$. The generator can do so using backpropagation to get the gradient of $E(\hat{Y})$ with respect to $\hat{Y}$. The generator and the model are trained simultaneously, with the model attempting to give low energy to training samples, and high energy to generated contrastive samples.

GANs are somewhat tricky to optimize, but adversarial training ideas have proved extremely fertile, producing impressive results in image synthesis, and opening up many new applications in content creation and domain adaptation[34] as well as domain or style transfer.[87]

Making representations agree using contrastive learning. Contrastive learning provides a way to discover good feature vectors without having to reconstruct or generate pixels. The idea is to learn a feed-forward neural network that produces very similar output vectors when given two different crops of the same image[10] or two different views of the same object[17] but dissimilar output vectors for crops from different images or views of different objects. The squared distance between the two output vectors can be treated as an energy, which is pushed down for compatible pairs and pushed up for incompatible pairs.[24,80]

A series of recent papers that use convolutional nets for extracting representations that agree have produced promising results in visual feature learning. The positive pairs are composed of different versions of the same image that are distorted through cropping, scaling, rotation, color shift, blurring, and so on. The negative pairs are similarly distorted versions of different images which may be cleverly picked from the dataset through a process called hard negative mining or may simply be all of the distorted versions of other images in a minibatch. The hidden activity vector of one of the higher-level layers of the network is subsequently used as input to a linear classifier trained in a supervised manner. This Siamese net approach has yielded excellent results on standard image recognition benchmarks.[6,21,22,43,67] Very recently, two Siamese net approaches have managed to eschew the need for contrastive samples. The first one, dubbed SwAV, quantizes the output of one network to train the other network,[20] the second one, dubbed BYOL, smoothes the weight trajectory of one of the two networks, which is apparently enough to prevent a collapse.[40]

Variational auto-encoders. A popular recent self-supervised learning method is the Variational Auto-Encoder (VAE).[58] This consists of an encoder network that maps the image into a latent code space and a decoder network that generates an image from a latent code. The VAE limits the information capacity of the latent code by adding Gaussian noise to the output of the encoder before it is passed to the decoder. This is akin to packing small noisy spheres into a larger sphere of minimum radius. The information capacity is limited by how many noisy spheres fit inside the containing sphere. The noisy spheres repel each other because a good reconstruction error requires a small overlap between codes that correspond to different samples. Mathematically, the system minimizes a free energy obtained through marginalization of the latent code over the noise distribution. However, minimizing this free energy with respect to the parameters is intractable, and one has to rely on variational approximation methods from statistical physics that minimize an upper bound of the free energy.

The Future of Deep Learning

The performance of deep learning systems can often be dramatically improved by simply scaling them up. With a lot more data and a lot more computation, they generally work a lot better. The language model GPT-3[18] with 175 billion parameters (which is still tiny compared with the number of synapses in the human brain) generates noticeably better text than GPT-2 with only 1.5 billion parameters. The chatbots Meena[2] and BlenderBot[73] also keep improving as they get bigger. Enormous effort is now going into scaling up and it will improve existing systems a lot, but there are fundamental deficiencies of current deep learning that cannot be overcome by scaling alone, as discussed here.

Comparing human learning abilities with current AI suggests several directions for improvement:

1. Supervised learning requires too much labeled data and model-free reinforcement learning requires far too many trials. Humans seem to be able to generalize well with far less experience.

2. Current systems are not as robust to changes in distribution as humans, who can quickly adapt to such changes with very few examples.

3. Current deep learning is most successful at perception tasks and generally what are called system 1 tasks. Using deep learning for system 2 tasks that require a deliberate sequence of steps is an exciting area that is still in its infancy.

What needs to be improved. From the early days, theoreticians of machine learning have focused on the iid assumption, which states that the test cases are expected to come from the same distribution as the training examples. Unfortunately, this is not a realistic assumption in the real world: just consider the non-stationarities due to actions of various agents changing the world, or the gradually expanding mental horizon of a learning agent which always has more to learn and discover. As a practical consequence, the performance of today's best AI systems tends to take a hit when they go from the lab to the field.

Our desire to achieve greater robustness when confronted with changes in distribution (called out-of-distribution generalization) is a special case of the more general objective of reducing sample complexity (the number of examples needed to generalize well) when faced with a new task—as in transfer learning and lifelong learning[81]—or simply with a change in distribution or in the relationship between states of the world and rewards. Current supervised learning systems require many more examples than humans (when having to learn a new task) and the situation is even worse for model-free reinforcement learning[23] since each rewarded trial provides less information about the task than each labeled example. It has already been noted[61,76] that humans can generalize in a way that is different and more powerful than ordinary iid generalization: we can correctly interpret novel combinations of existing concepts, even if those combina-

tions are extremely unlikely under our training distribution, so long as they respect high-level syntactic and semantic patterns we have already learned. Recent studies help us clarify how different neural net architectures fare in terms of this systematic generalization ability.[8,9] How can we design future machine learning systems with these abilities to generalize better or adapt faster out-of-distribution?

From homogeneous layers to groups of neurons that represent entities. Evidence from neuroscience suggests that groups of nearby neurons (forming what is called a hyper-column) are tightly connected and might represent a kind of higher-level vector-valued unit able to send not just a scalar quantity but rather a set of coordinated values. This idea is at the heart of the capsules architectures,[47,59] and it is also inherent in the use of soft-attention mechanisms, where each element in the set is associated with a vector, from which one can read a key vector and a value vector (and sometimes also a query vector). One way to think about these vector-level units is as representing the detection of an object along with its attributes (like pose information, in capsules). Recent papers in computer vision are exploring extensions of convolutional neural networks in which the top level of the hierarchy represents a set of candidate objects detected in the input image, and operations on these candidates is performed with transformer-like architectures.[19,84,86] Neural networks that assign intrinsic frames of reference to objects and their parts and recognize objects by using the geometric relationships between parts should be far less vulnerable to directed adversarial attacks,[79] which rely on the large difference between the information used by people and that used by neural nets to recognize objects.

Multiple time scales of adaption. Most neural nets only have two timescales: the weights adapt slowly over many examples and the activities adapt rapidly changing with each new input. Adding an overlay of rapidly adapting and rapidly, decaying "fast weights"[49] introduces interesting new computational abilities. In particular, it creates a high-capacity, short-term memory,[4] which allows a neural net to perform true recursion in which the same neurons can be reused in a recursive call because their activity vector in the higher-level call can be reconstructed later using the information in the fast weights. Multiple time scales of adaption also arise in learning to learn, or meta-learning.[12,33,75]

Higher-level cognition. When thinking about a new challenge, such as driving in a city with unusual traffic rules, or even imagining driving a vehicle on the moon, we can take advantage of pieces of knowledge and generic skills we have already mastered and recombine them dynamically in new ways. This form of systematic generalization allows humans to generalize fairly well in contexts that are very unlikely under their training distribution. We can then further improve with practice, fine-tuning and compiling these new skills so they do not need conscious attention anymore. How could we endow neural networks with the ability to adapt quickly to new settings by mostly reusing already known pieces of knowledge, thus avoiding interference with known skills? Initial steps in that direction include Transformers[32] and Recurrent Independent Mechanisms.[38]

It seems that our implicit (system 1) processing abilities allow us to guess potentially good or dangerous futures, when planning or reasoning. This raises the question of how system 1 networks could guide search and planning at the higher (system 2) level, maybe in the spirit of the value functions which guide Monte-Carlo tree search for AlphaGo.[77]

Machine learning research relies on inductive biases or priors in order to encourage learning in directions which are compatible with some assumptions about the world. The nature of system 2 processing and cognitive neuroscience theories for them[5,30] suggests several such inductive biases and architectures,[11,45] which may be exploited to design novel deep learning systems. How do we design deep learning architectures and training frameworks which incorporate such inductive biases?

The ability of young children to perform causal discovery[37] suggests this may be a basic property of the human brain, and recent work suggests that optimizing out-of-distribution generalization under interventional changes can be used to train neural networks to discover causal dependencies or causal variables.[3,13,57,66] How should we structure and train neural nets so they can capture these underlying causal properties of the world?

How are the directions suggested by these open questions related to the symbolic AI research program from the 20th century? Clearly, this symbolic AI program aimed at achieving system 2 abilities, such as reasoning, being able to factorize knowledge into pieces which can easily recombined in a sequence of computational steps, and being able to manipulate abstract variables, types, and instances. We would like to design neural networks which can do all these things while working with real-valued vectors so as to preserve the strengths of deep learning which include efficient large-scale learning using differentiable computation and gradient-based adaptation, grounding of high-level concepts in low-level perception and action, handling uncertain data, and using distributed representations. C

References

1. Abadi, M. et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symp. Operating Systems Design and Implementation*, 2016, 265–283.
2. Adiwardana, D., Luong, M., So, D., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. Towards a human-like open-domain chatbot 2020; *arXiv preprint arXiv:2001.09977*.
3. Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 2019; *arXiv preprint arXiv:1907.02893*.
4. Ba, J., Hinton, G., Mnih, V., Leibo, J., and Ionescu, C. Using fast weights to attend to the recent past. *Advances in Neural Information Processing Systems*, 2016, 4331–4339.
5. Baars, B. *A Cognitive Theory of Consciousness*. Cambridge University Press, Cambridge, MA, 1993.
6. Bachman, P., Hjelm, R., and Buchwalter, W. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 2019, 15535–15545.
7. Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014; *arXiv:1409.0473*.
8. Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T., Vries, H., and Courville, A. Systematic generalization: What is required and can it be learned? 2018; *arXiv:1811.12889*.
9. Bahdanau, D., de Vries, H., O'Donnell, T., Murty, S., Beaudoin, P., Bengio, Y., and Courville, A. Closure: Assessing systematic generalization of clever models, 2019; *arXiv:1912.05783*.
10. Becker, S. and Hinton, G. Self-organizing neural network that discovers surfaces in random dot stereograms. *Nature 355*, 6356 (1992), 161–163.
11. Bengio, Y. The consciousness prior, 2017; *arXiv:1709.08568*.
12. Bengio, Y., Bengio, S., and Cloutier, J. Learning a synaptic learning rule. In *Proceedings of the IEEE 1991 Seattle Intern. Joint Conf. Neural Networks 2*.
13. Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of ICLR'2020; arXiv:1901.10912*.
14. Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *NIPS'2000*, 2001, 932–938.
15. Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In

Proceedings of NIPS'2006, 2007.
16. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. Theano: A CPU and GPU math expression compiler. In *Proceedings of SciPy*. 2010.
17. Bromley, J., Guyon, I., LeCun, Y., Säkinger, E., and Shah, R. Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems*, 1994, 737–744.
18. Brown, T. et al. Language models are few-shot learners, 2020; *arXiv:2005.14165*.
19. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *Procedings of ECCV'2020*; *arXiv:2005.12872*.
20. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments, 2020;. *arXiv:2006.09882*.
21. Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020; *arXiv:2002.05709*.
22. Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning, 2020; *arXiv:2003.04297*.
23. Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T., and Bengio, Y. Babyai: First steps towards grounded language learning with a human in the loop. In Proceedings in ICLR'2019; *arXiv:1810.08272*.
24. Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conf. Computer Vision and Pattern Recognition 1*, 539–546.
25. Collobert, R., Kavukcuoglu, K., and Farabet, C. Torch7: A matlab-like environment for machine learning. In *Proceedings of NIPS Worskshop BigLearn*, 2011.
26. Collobert, R. and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML'2008*.
27. Conneau, A. and Lample, G. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems 32*, 2019. H. Wallach et al., eds. 7059–7069. Curran Associates, Inc.; http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf.
28. Dahl, G., Yu, D., Deng, L., and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech, and Language Processing 20*, 1 (2011), 30–42.
29. Dayan, P. and Abbott, L. *Theoretical Neuroscience*. The MIT Press, 2001.
30. Dehaene, S., Lau, H., and Kouider, S. What is consciousness, and could machines have it? *Science 358*, 6362 (2017, 486–492.
31. Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of 2009 IEEE Conf. Computer Vision and Pattern Recognition*, 248–255.
32. Devlin, J., Chang, M., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of ACL'2019*; *arXiv:1810.04805*.
33. Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks, 2017; *arXiv:1703.03400*.
34. Ganin, Y and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of Intern. Conf. Machine Learning*, 2015, 1180–1189.
35. Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of AISTATS'2011*.
36. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014, 2672–26804.
37. Gopnik, A., Glymour, C., Sobel, D., Schulz, L., Kushnir, T., and Danks, D. A theory of causal learning in children: causal maps and bayes nets. *Psychological Review 111*, 1 (2004).
38. Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms, 2019; *arXiv:1909.10893*.
39. Graves, A. Generating sequences with recurrent neural networks, 2013; *arXiv:1308.0850*.
40. Grill, J-B. et al. Bootstrap your own latent: A new approach to self-supervised learning, 2020; *aeXiv:2006.07733*.
41. Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th Intern. Conf. Artificial Intelligence and Statistics*, 2010, 297–304.
42. He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR'2020*, June 2020.
43. He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of CVPR'2016*, 770–778.
44. Hinton, G. A parallel computation that assigns canonical object-based frames of reference. In *Proceedings of the 7th Intern. Joint Conf. Artificial Intelligence 2*. 1981, 683–685.
45. Hinton, G. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence 46*, 1-2 (1990), 47–75.
46. Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing 29*, 6 (2012), 82–97.
47. Hinton, G., Krizhevsky, A., and Wang, S. Transforming auto-encoders. In *Proceedings of Intern. Conf. Artificial Neural Networks*. Springer, 2011, 44–51.
48. Hinton, G., Osindero, S., and Teh, Y-W. A fast-learning algorithm for deep belief nets. *Neural Computation 18* (2006), 1527–1554.
49. Hinton, G. and Plaut, D. Using fast weights to deblur old memories. In *Proceedings of the 9th Annual Conf. Cognitive Science Society*. 1987, 177–186.
50. Hinton, G. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science 313* (July 2006), 504–507.
51. Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. In *Proceedings of NeurIPS'2012*; arXiv:1207.0580.
52. Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (1997), 1735–1780.
53. Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.
54. Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. What is the best multi-stage architecture for object recognition? In *Proceedings of ICCV'09*, 2009.
55. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM Intern. Conf. Multimedia*, 2014, 675–678.
56. Kahneman, D. *Thinking, Fast and Slow*. Macmillan, 2011.
57. Ke, N., Bilaniuk, O., Goyal, A., Bauer, S., Larochelle, H., Pal, C., and Bengio, Y. Learning neural causal models from unknown interventions, 2019; *arXiv:1910.01075*.
58. Kingma, D. and Welling, M. Auto-encoding variational bayes. In *Proceedings of the Intern. Conf. Learning Representations*, 2014.
59. Kosiorek, A., Sabour, S., Teh, Y., and Hinton, G. Stacked capsule autoencoders. *Advances in Neural Information Processing Systems*, 2019, 15512–15522.
60. Krizhevsky, A., Sutskever, I., and Hinton, G. ImageNet classification with deep convolutional neural networks. In *Proceedings of NIPS'2012*.
61. Lake, B., Ullman, T., Tenenbaum, J., and Gershman, S. Building machines that learn and think like people. *Behavioral and Brain Sciences 40* (2017).
62. Lample, G. and Charton, F. Deep learning for symbolic mathematics. In *Proceedings of ICLR'2020*; *arXiv:1912.01412*.
63. LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature 521*, 7553 (2015), 436–444.
64. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. Backpropagation applied to handwritten zip code recognition. *Neural Computation 1*, 4 (1989), 541–551.
65. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.
66. Lopez-Paz, D., Nishihara, R., Chintala, S., Scholkopf, B., and Bottou, L. Discovering causal signals in images. In *Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition*, 2017, 6979–6987.
67. Misra, I. and Maaten, L. Self-supervised learning of pretext-invariant representations. In *Proceedings of CVPR'2020*, June 2020; *arXiv:1912.01991*.
68. Mohamed, A., Dahl, G., and Hinton, G. Deep belief networks for phone recognition. In *Proceedings of NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*. (Vancouver, Canada, 2009).
69. Morgan, N., Beck, J., Allman, E., and Beer, J. Rap: A ring array processor for multilayer perceptron applications. In *Proceedings of the IEEE Intern. Conf. Acoustics, Speech, and Signal Processing*, 1990, 1005–1008.
70. Nair, V. and Hinton, G. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the ICML'2010*.
71. Paszke, A., et al. Automatic differentiation in pytorch. 2017.
72. Robinson, A. An application of recurrent nets to phone probability estimation. *IEEE Trans. Neural Networks 5*, 2 (1994), 298–305.
73. Roller, S., et al. Recipes for building an open domain chatbot, 2020; *arXiv:2004.13637*.
74. Rumelhart, D., Hinton, G., and Williams, R. Learning representations by back-propagating errors. *Nature 323* (1986), 533–536.
75. Schmidhuber, J. Evolutionary principles in self-referential learning. Diploma thesis, Institut f. Informatik, Tech.Univ. Munich, 1987.
76. Shepard, R. Toward a universal law of generalization for psychological science. *Science 237*, 4820 (1987), 1317–1323.
77. Silver, D., et al. Mastering the game of go with deep neural networks and tree search. *Nature 529*, 7587 (2016), 484.
78. Sukhbaatar, S., Szlam, A., Weston, J., and Fergus, R. End-to-end memory networks. *Advances in Neural Information Processing Systems 28*, 2015, 2440–2448. C. Cortes et al., eds. Curran Associates, Inc.; http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf.
79. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *Proceedings of ICLR'2014*; *arXiv:1312.6199*.
80. Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. Web-scale training for face identification. In *Proceedings of CVPR'2015*, 2746–2754.
81. Thrun, S. Is learning the n-th thing any easier than learning the first? In *Proceedings of NIPS'1995*. MIT Press, Cambridge, MA, 640–646.
82. Utgoff, P. and Stracuzzi, D. Many-layered learning. *Neural Computation 14* (2002), 2497–2539, 2002.
83. Van Essen, D. and Maunsell, J. Hierarchical organization and functional streams in the visual cortex. *Trends in Neurosciences 6* (1983), 370–375.
84. van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions, 2018; *arXiv:1802.10353*.
85. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, T., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, 5998–6008.
86. Zambaldi, V., et al. Relational deep reinforcement learning, 2018; arXiv:1806.01830.
87. Zhu, J-Y., Park, T., Isola, P., and Efros, A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the 2017 IEEE Intern. Conf. on Computer Vision*, 2223–2232.

Yoshua Bengio is a professor in the Department of Computer Science and Operational Research at the Université de Montréal. He is also the founder and scientific director of Mila, the Quebec Artificial Intelligence Institute, and the co-director of CIFAR's Learning in Machines & Brains program.

Yann LeCun is VP and Chief AI Scientist at Facebook and Silver Professor at New York University affiliated with the Courant Institute of Mathematical Sciences and the Center for Data Science, New York, NY, USA.

Geoffrey Hinton is the Chief Scientific Advisor of the Vector Institute, Toronto, Vice President and Engineering Fellow at Google, and Emeritus Distinguished Professor of Computer Science at the University of Toronto, Canada.

This work is licensed under a http://creativecommons.org/licenses/by/4.0/

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/deep-learning-for-ai