Deep Learning: Implications for Human Learning and Memory

James L. McClelland and Matthew M. Botvinick

Recent years have seen an explosion of interest in deep learning and deep neural networks. Deep learning lies at the heart of unprecedented feats of machine intelligence as well as software people use every day. Systems built on deep learning have surpassed human capabilities in complex strategy games like go and chess, and we use them for speech recognition, image captioning, and a wide range of other applications. A consideration of deep learning is crucial for a Handbook of Human Memory, since human brains are deep neural networks, and an understanding of artificial deep learning systems may contribute to our understanding of how humans and animals learn and remember.

Deep neural networks are complex, structured systems that process information in a parallel, distributed, and context sensitive fashion, and deep learning is the effort to use these systems to acquire capabilities we associate with intelligence through an experience dependent learning process. Within the field of Artificial Intelligence, work in deep learning is typically directed toward the goal of creating and understanding intelligence using all available tools and resources without consideration of their biological plausibility. Many of the ideas, however, at the heart of deep learning draw their inspiration from the brain and from characteristics of human intelligence we believe are best captured by these brain-inspired systems (Rumelhart, McClelland, and the PDP Research Group, 1986). Furthermore, ideas emerging from deep learning research can help inform us about memory and learning in humans and animals. Thus, deep learning research can be seen as fertile ground for cross-engagement between researchers who work on related issues with implications for both biological and machine intelligence.

We begin by introducing the basic constructs employed in deep learning and then consider several of the widely used learning paradigms and architectures used in these systems. We then turn to a consideration of how the constructs of deep learning relate to traditional constructs in the psychological literature on learning and memory. Next, we consider recent developments in the field of reinforcement learning that have broad implications for human learning and memory. We conclude with a consideration of areas where human capabilities still far exceed current deep learning approaches, and describe possible future directions toward understanding how these abilities might best be captured.

What is a Deep Neural Network?

A deep neural network is a system that processes inputs through multiple processing layers to produce outputs. In the simplest case, each layer takes a pattern of activation represented as a vector of numeric values over a set of neuron-like processing units and transforms this pattern into an output pattern using an array of real valued connection weights followed by the application of a non-linear operation. To make these ideas concrete, we consider a generic deep neural network that performs a handwritten digit classification task.[1] The network, diagrammed in Figure 1, processes input patterns derived from handwritten digits, and produces, for each input pattern, an output we can think of as representing, for each of the digits 0 through 9, an estimate of the probability that the input is an instance of that digit. In the network in question, there are three processing stages (the minimum most people would consider necessary for the network to be called deep). The input to the first is a pattern corresponding to the intensities of a 28 x 28 array of pixels (black = 0, white = 1; intermediate values allowed) is the digitized image of a handwritten digit, a pattern we can think of as an approximation of the retinal image someone would see upon fixating the digit. This image is transformed by multiplying it with a matrix of connection weights, which can be thought of as analogous to the array of synaptic connections from the neurons in a layer of sensory neurons in the eye to another layer of neurons deeper in the brain. This next layer is called a hidden layer since we do not specify either the activations or target outputs for this layer. Since it is the first such layer in our network we will call it $\mathbf{h}_1$**.** We adopt the convention of treating the input $\mathbf{x}$ as a pattern on layer 0 of the network, and representing the matrix to the first hidden layer from the input layer as $\mathbf{W}_{10}$**.** Still in the first stage of processing, we add a vector of learned bias parameters $\mathbf{b}_1$ to the output of the matrix multiplication. These biases can be thought of as determining the baseline excitabilities of each of the neuron-like units in layer $\mathbf{h}_1$. Finally, we apply a non-linear transformation to each value in the pattern produced by these operations. Many alternative non-linearities are used in deep learning research. Two basic ones are the rectified-linear (**relu)** function, which simply passes the value through if it is positive and otherwise sets the value to 0, and a function called **tanh** that keeps the activations bounded in the range between -1 and +1. In our case we apply the **relu** to produce the pattern $\mathbf{h}_1$**.** This completes the first stage of processing in our network.

We have introduced many concepts in the last paragraph, but this is nearly all we need to characterize the processing operations performed in our generic neural network, as almost all of the rest of the computations rely on the same ideas. Indeed, our second stage is very like the first, in that it multiplies the pattern $\mathbf{h}_1$ times a second matrix $\mathbf{W}_{21}$**,** adds an array of bias weights $\mathbf{b}_2$**,** and applies a **relu** non-linearity again to produce hidden layer pattern $\mathbf{h}_2$**.** The third

[1] This network is based on one created as an introduction to Tensorflow, a deep learning software toolkit. You can look at the code for the network, run it using Google Colab, and repeat the analyses shown in Figure 1 at https://github.com/jlmcc94303/TwoHiddenLayerMNISTDemo.

stage is a little different. We multiply $\mathbf{h}_2$ times a third weight matrix $\mathbf{W}_{32}$ and add a set of output bias weights $\mathbf{b}_3$ as before**,** but this time, we apply the non-linear **softmax** function to create estimates of the probability that the input **x** corresponds to each digit. The softmax function enforces the law of total probability which requires that the probability estimates add up to 1.[2] In concert with common conventions, we call the array of probability estimates **y**.

That's it! If we had such a network, we could study how well it classifies digits. One way this can be done is to present a set of test digits, and determine how often the correct digit is the one with the largest value in the output pattern **y**. In the case of the network in Figure 1, it can classify held out test items with an accuracy of about 98% correct after 5 training epochs (sweeps through all the items in a 50,000 example training set), a pretty high score for a very generic network. We can also study the network's representations -- for example, we can ask whether the representations at layer $\mathbf{h}_1$ capture the similarity relationships among the digits. In Figure 1B (lower panel), we show such an analysis based on a test set of 1000 images from each of the 10 digit classes. We see there that the network tends to treat 3's as similar to 2's, 5's and 8's. We can also consider the network's output confusability matrix -- how probable it thinks each of the possible class labels is on average, when shown the 1000 test examples from each of the digits classes. The results of this analysis, shown in Figure 1B (top panel), are largely consistent with the similarity relationships in the bottom panel, but one can see that the remaining two layers of the network have succeeded in reducing the similarity of the different classes, allowing the network to achieve a high level of accuracy.

As we have seen, a deep neural network processes input patterns through several layers or stages of processing, producing hidden layer patterns and, finally, outputs. The patterns inside the network are learned distributed representations, often called *hidden representations, latent representations,* or *embeddings.* In artificial networks they correspond to arrays of numerical values that we can think of in biological terms as corresponding abstractly to the pattern of firing rates over a set of neurons in the brain. In thinking about these patterns, we often do not think of the individual values of the elements in the arrays -- instead, we may think about the whole pattern of activation of the units at each layer. In that case, we generally think of the similarity relations among the patterns for different items when we seek to understand the content of these patterns, as described above, though some work does consider how individual elements might correspond to other things, such as conceptual primitives or individual neuron response rates (Yamins et al, 2014).

Why depth is important

[2] The softmax function takes a vector of real numbers **v** and converts it to a vector **p** of positive numbers that sum to 1. For each element v_i of **v,** we compute an intermediate value $\underline{s}_i = \exp(v_i)$. Then, we obtain each of the p_i values by dividing each of the s_i by the sum of all of the s_i values. See McClelland (2013) for a tutorial discussion of the probabilistic interpretation of these values and how neurons might compute them.

Now that we know what a deep network is, it is important to be clear about why depth is important. Why don't we simply have direct connections between inputs and outputs, rather than worrying about having multiple layers of units in between? The answer begins with the fact that limiting a network to direct connections limits the computations that the network can perform. With one or more hidden layers, a neural network can perform arbitrary input-output mapping tasks; without a hidden layer, this is not the case. Perhaps the simplest concrete example of these two points is provided by the exclusive-or (XOR) problem. In XOR, there are two input units and one output unit. The input units can each take values of 1 or 0. When either input unit takes value 1 and the other unit takes value 0 the output should be 1, but when both take the value 1 or the value 0, the output should be -1. To see why this cannot be computed without a hidden layer, let's assume we just have one weight from each input unit to the output unit, and one bias weight, and let's assume that the output unit takes value 1 if **Wx** + **b** is greater than 0, and value -1 if **Wx** + **b** is less than 0. There is no setting of the two weights and the bias parameter that can produce the correct answer for all four patterns. (For example, if both weights are positive, and the bias is negative, the network will produce a negative output when both inputs are off and a positive output when both are on, as desired, but the output will necessarily be positive when both input units are on). However, if we introduce two hidden units, one that comes on when either input unit is on and another than comes on when both hidden units are on, we can then set the weights from these units to the output units so that the output unit will come on if the first hidden unit is on, but not if neither is on, and not if both are on. This simple example illustrates an important role of a hidden layer: it performs a non-linear transformation of its input, recoding the input in a way that is useful for solving a given input-output mapping problem. (For a more detailed discussion, and an XOR network that works with just one hidden unit, see McClelland & Rumelhart, 1988, Chapter 4).

For many years, researchers debated whether it is useful or necessary to have more than one hidden layer, but now it is generally accepted that more than one is usually better, and in one case, a network's performance was shown to improve with greater depth up to as many as 1000 layers (He *et al.,* 2016). The reasons why depth is so important are not fully understood. One intuition about the importance of greater depth is the idea that it allows for multiple stages of recoding that are useful for capturing subtle nonlinear aspects of input-output mappings, allowing networks to disentangle underlying similarities among inputs that may not be easily extracted in a single layer. Furthermore, deeper networks often generalize better to items not included in their training set than shallower ones, perhaps in part because the recoding at each layer can allow less task-relevant aspects of variation in the input to be eliminated or separated out from other more task relevant input variation.

Knowledge and Learning in a Deep Neural Network

Knowledge in connections. It seems natural to say that the deep network we discussed above "knows" how to classify digits. However, the form of knowledge may not be the same as the form of knowledge we usually talk about when we ask if a person knows something. What exactly is the knowledge in the neural network? This knowledge consists primarily of the values of the connection weights in the weight arrays $\mathbf{W}_{10}$, $\mathbf{W}_{21}$, and $\mathbf{W}_{32}$, and in the bias arrays $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$. Collectively these values are often called the *parameters* of the neural network, and they constitute the knowledge it uses to perform its classification. Such networks have often been called *connectionist* networks -- because their knowledge is in their connections.

Deep learning. With the background laid out above, we are now ready to tackle deep learning, that is learning in deep neural networks. Essentially, deep learning is the process whereby experience shapes the values of the connections in a network. We first describe what is perhaps the most common approach, usually called supervised learning, considering extensions including reinforcement learning later.

Applying supervised learning to our deep network for digit classification, we use a set of images together with their corresponding labels to train the network. We can think of the process as an idealization of one that could apply to a biological system if it was shown a set of items and their names or labels. Indeed, a rudimentary version of this computation was proposed by Frank Rosenblatt, the pioneering neural network researcher of the mid-20th century (e.g., Rosenblatt, 1962). First, the network is initialized with random weights and biases, so that initially, its output is not based on any experience. Then, it is trained on items sampled from its training set. For each item, it takes its input pattern and processes it using the system we have described. Then, the item's label is compared to the value the network produces as its estimate of the probability of the correct label, and a measure called the ***loss*** reflecting how different this estimate is from 1.0 is calculated. Following this, a computation is performed to determine how each connection weight should change to reduce the loss, making the probability of the correct label higher and the probabilities of all other labels lower. For each parameter (i.e., each weight in each weight array and each bias in each bias array) this computation determines a quantity indicating how much (and in which direction) a change in the value of the parameter will affect the loss. Each of these values is called the ***gradient*** of the parameter with respect to the loss, and the entire ensemble of such values is collectively referred to as the gradient of the entire set of parameters. The connection weights are then adjusted by an amount proportional to the negative of the gradient, hence the name ***gradient descent.*** The weights move down the gradient, thereby reducing the loss.

The use of gradient descent has, as already indicated, been part of computational approaches to intelligence for a very long time. Gradient descent can be seen as a way of providing a means for optimizing the performance of a neural network in response to demands placed on it by its experiences. As we shall see, the principle can be extended in many ways, allowing learning systems to learn from experiences without labels, or with reward signals indicating how good the network's outputs are, without specifying what specific outputs are the correct ones.

Backpropagation, and why it is important. At first, gradient descent was only applied to a single layer of connection weights (the ones just before the network's output layer), in part because researchers tended to think of neurons as having discrete values (either 0 or 1), which prevented calculating gradient values for weights in the earlier processing stages of the network. The idea of treating unit activations as continuous valued allowed gradients to be determined for all of the parameters in a deep network like ours, according to a computation called *backpropagation* (Rumelhart, Hinton & Williams, 1986), because the signals indicating how weights coming into, for example, our $\mathbf{h}_2$ from $\mathbf{h}_1$ should change we determined by sending learning signals at the output layer backward through weight matrix $\mathbf{W}_{32}$ used to propagate error signals forward to the output layer of the network. The introduction of backpropagation was a crucial advance, because it could be applied repeatedly through any number of layers of weights, allowing researchers to learn connections weights at all layers of a neural network, not just at a final output layer. While multi-layer networks could be used without back-propagation, all of the connections in layers other than the last one had to be set by hand, making it impossible for networks to learn the non-linear transformations of inputs that would be most useful for performing the task at hand. Thus, we could say that back propagation really enabled deep learning, viewed as the process of setting the connection weights in all of the layers in a deep neural network.

Comment. It should be noted that what we have described here is a bare-bones implementation of gradient descent learning. Deep learning researchers employ many methods that enhance learning performance. Since our purpose here is to focus on fundamental properties, we omit discussion of these enhancements here. It may also be worth noting that backpropagation was initially criticized by some scientists interested in biological computation because the algorithm seemed to require precise and rapid reverse transmission across axons and synapses. Debate about this question is still underway, although current trends indicate more openness to exploring how computations approximating back propagation might actually be carried out in the brain (Richards & Lillicrap, 2019). In our view, there are exciting and interesting questions that remain to be addressed regarding how the biological brain learns. While these are being explored, it is useful to consider the consequences of deep learning based on the principle of gradient descent, as the principle allows us to explore computations that actually lead to solutions of challenging problems for perception, memory, and other functions of the mind and brain. Once these problems can be posed in terms of a training set and a loss measure, we can use gradient descent to find a solution.[3]

[3] An early concern with gradient descent was that the process would likely become stuck in a local minimum -- a set of parameters values less than all other neighboring values, but not as good as the best possible set of values overall. However, this problem is far less severe than was once thought. A more severe problem has turned out to be that neural networks can easily overfit their training data, limiting their ability to handle unseen examples. Fully

Architectures for Deep Learning

Understanding gradient descent in the simple-feedforward network considered in the previous section provides a starting place for work in deep learning, but it is important to recognize several of the more complex network architectures that are in use in deep learning as well. Here we consider three other architectures: *convolutional*, *recurrent*, and *attentional* neural networks. Deep-learning researchers create hybrids of these architectures to suit their needs in building models for artificial intelligence. It seems to us likely that the brain exploits aspects or variants of all of these architectures, though perhaps without some of the features that are exploited in artificial intelligence.

Convolutional neural networks. A very common type of artificial neural network, the convolutional neural network (CNN), was initially inspired by aspects of the neural networks in the brain (LeCun *et al,* 2015). It was observed that neurons in the visual pathway respond to similar stimulus properties at different positions in the visual field, leading to neural networks that use many identical copies of units with connection weights that are replicated at many positions across the visual field. For example, to represent the presence of line segments at eight different evenly spaced orientations at any position within an image, the same matrix of connection weights can be replicated many times to project every patch of an image to a patch-specific set of eight units, one for each orientation. A local pooling operation can then be applied from several adjacent sets of patch-specific units, resulting in a set of eight units in a pooling layer whose activation indicates to what extent the input contains a line of each of the orientations anywhere in the set of local patches. The types of units are inspired by the presence of simple and complex cells in primary visual cortex in primate brains, corresponding to the patch-specific units and local pooling units respectively, and the pattern of topographically-restricted connectivity known to characterize the visual pathway is captured by the topographically-restricted connectivity of the CNN.

Artificial CNN's are often trained to classify images into hundreds or even thousands of different classes using supervised learning. One trick they employ that is common in deep learning is to add the gradient signals driving each copy of the replicated connection weights at each position in a convolutional layer together and then use the summed gradient signal to update all the copies of the weights, so that all copies always remain exactly the same. (In a biological system, fixating on many different positions within the same image of an object and backpropagating the classification loss from each fixation would have approximately the same effect.)

Recurrent neural networks. The networks we have considered thus far have been useful for many tasks, but lack an important capability: the ability to use information distributed over time. Recurrent neural networks (RNNs) provide this capability. RNNs generally treat time as a

understanding and addressing the reasons why neural networks overfit remains an important and challenging issue.

sequence of discrete steps, each containing an element of a sequence, such as a frame of a video or audio stream, or a character in a printed text. The simplest form of RNN employs a stack of layers like a feed-forward network, but with the added feature that at each layer, its state at the end of the previous time step provides another source of input, via an additional connection weight matrix. For example, we could turn our basic feed-forward network into an RNN by adding new weight matrices $\mathbf{W}_{11}$ and $\mathbf{W}_{22}$, so that at time *t,* the input to hidden layer 1 would be $\mathbf{W}_{10}\mathbf{x}(t)+\mathbf{W}_{11}\mathbf{h}_1(t-1)+\mathbf{b}_1$**,** and the input to hidden layer 2 would be $\mathbf{W}_{21}\mathbf{h}_1(t)+\mathbf{W}_{22}\mathbf{h}_2(t-1)+\mathbf{b}_2$, as shown in Figure 2A. The simplest version of such a network, introduced by Elman (1990), demonstrated how this extension could capture key aspects of language structure. Two ways of looking at what is today called a 'Vanilla RNN' are presented in Figure 2. Conceptually, we can imagine that there is really only a single copy of the network, which simply updates its state as every time step, as shown in panel A, computer implementations of RNNs are best visualised as consisting of many separate copies of the same network, one for each of some specified number of time steps, as shown in panel B. The values of the connection weights are shared across all of the copies. Thus, RNN's are like CNN's in that they reuse the same connection weights -- here, for each point in time, rather than, as in the CNN, for each point in space. The computed weight update to each of the shared weight matrices is the sum of the updates to each instance of each weight matrix, based on back propagation (see figure caption for details).

While lots of interesting work was done with these networks over the years following the introduction of backpropagation, they fell into disuse in AI research because they struggled to master long-distance dependencies -- for example, in a sentence such as 'Inside the cave/dugout, John looked around and saw a bat' they could not span intervening words to use prior context (cave or dugout) to constrain the meaning assigned to a later word (bat). A major part of the problem is that the gradient of a loss generated at a particular time step tends to vanish rapidly as it is back propagated through each time step, making it very hard to learn to exploit long-distance dependencies. Another factor is that the hidden state of the network is continually updated with new information. A major breakthrough occurred with the development and subsequent refinement of long-short-term memory modules (LSTM's, Hochreiter & Schmidhuber, 1993), which came into wide use in language-related deep learning research early in this century. These modules include learnable connection weights that 'gate' information into and out of an internal context state within each neural network module and that can also learn to 'forget' or remove no longer needed information from the context state. An excellent introduction to these models is provided in Colah's blog (Olah, 2015).

LSTM-based models have contributed to many exciting developments in natural language processing starting about 10 years ago. For example, such models were able to learn to perform language prediction tasks: Using a large text corpus, an LSTM-based model could learn to use the preceding elements of the text to predict subsequent elements, or even to predict the translation of a text into another language. An accessible overview of work of this type is provided in a blob post by Andrej Karpathy (Karpathy, 2015). It has turned out, however, that the LSTM based networks still suffer to a degree from the same problems that plagued vanilla RNNs, and they have now been largely superseded by networks that use learned attention.

Neural networks with learned attention mechanisms. Since about 2015, another powerful architectural feature has been added to the deep learning toolkit -- a feature deep learning researchers call *attention*. The idea here is to learn patterns called *queries* that can be compared to other learned patterns called *keys,* to decide what set of units to attend to (that is, to draw information from), to constrain a computation of a representation in another set of units. For example, in a recent attention-based model of language processing called BERT (Devlin *et al,* 2018), a set of queries, keys and values is computed at each of several layers of processing at each position in a multi-word sequence. At each layer of processing, the queries from each position are compared to the keys at all other positions, and are used to weight the contributions of the values at each position to the computation of the representation of the meaning of the word at every other position (see Uszkoreit, J. 2017). Models based on learned attention are now the main drivers of language processing systems that have achieved state of the art performance in machine translation and several other tasks. Learned attention can also be used in vision and many other applications, and is likely to contribute to continued progress in a wide range of application areas.

Sources of Teaching Signals that Guide Learning

A key conceptual as well as practical issue for deep learning systems is the source of the teaching signals that drive learning. In our generic example, we considered the case where the source of the teaching signal was the label of each image of a hand-written digit. Cases like this are considered to be *supervised learning,* with the supervisory signal being the label or correct name of each digit in the training set. However, real life does not always present us with labels that tell us exactly what our response to an input should be. A great deal of deep learning research has been done with the idea that we should be able to learn without having an explicit supervisory signal, and we consider this idea here. In a later section, we consider cases where the supervisory signal is a reward signal, under the heading *Deep Reinforcement Learning.*

A very important class of models considers learning from images, texts, or other types of inputs, in the absence of any labels. Such models are often called *unsupervised,* but this word can actually cover two rather different approaches that we will distinguish. In the first approach, the task is to build a network that can reconstruct its input or predict some elements of its input from other elements. As we shall see this approach allows us to apply the same gradient-based approach we have considered thus far to minimize the reconstruction or prediction loss. In the second approach, we do not use a loss computed against external patterns of any kind, though even here, the learning can often be understood as gradient descent, in this case on some measure of system-internal behavior. In both cases, the key point is that no special supervisory signal is required -- we only have inputs or input streams as the source of experiences that drive learning.

Reconstruction- and prediction-error driven learning

Let's return to the case of hand printed digits and consider a situation in which we have many images of hand-printed digits, but no labels for any of the images. Here, we might seek to form an internal representation from each image that allows us to reconstruct the image. Of course, a simple element-wise copy of the input would be a trivial internal representation we could use for this purpose, but since the first report of backpropagation (Rumelhart *et al,* 1986), researchers have used a neural network to map from the input back to itself through one or more hidden layers, using gradient descent to learn internal representations that suffice to reconstruct the input, subject to the constraints imposed by the architecture of the neural network. The network architecture could be almost identical to the one we used for digit classification, with the simple change that we replace the output classification layer with an image reconstruction layer, where we use gradient descent on a loss that we calculate pixel by pixel, then sum across the output units as the total loss signal used to train the weights in the network.

Prediction-error driven learning works in a similar way. Elman's (1990) work on next-word prediction using his simple recurrent network exemplifies the approach. Elman's training data simply consisted of a long sequence of words, with the task being simply to predict the next word from the current word together with the previous hidden state, and this approach is in wide use in many deep language processing models today. An extension of this approach is to predict missing aspects of an input from other aspects. This is the approach used in attention-based language models such as BERT, where the network is given multiword sequences (e.g. a pair of sentences) at a time. Some of the words are deleted or replaced with random elements at the input level, and the task is to reconstruct the uncorrupted sequence at the output level.

Learning without relying on any external supervisory signal

It was a breakthrough for models of intelligent behavior when gradient-based methods were introduced. The use of a measure of loss against some external supervisory signal, even if it is just the input itself, is useful because we can then see connection weight changes as directly serving the goal of minimizing some externally observable behavior of our system, and such behavior is ultimately what intelligent machines must be optimized or biologically selected for. However, early biologically motivated approaches to learning usually were not formulated in terms of gradient descent in a loss measure. For example, Hebb's famous learning rule (1949/2005) tells us that the brain strengthens the connection from neuron A to neuron B when A takes part in causing B to fire, a matter that is entirely internal to the system. Hebb's idea has motivated and inspired research for over 70 years, but gaining traction toward building intelligent machines has depended on linking connection adjustment to the enhancement of performance as assessed by a measure usually called *loss*, or distance from some standard. That said, there are many interesting learning algorithms that do not rely on any external loss measure. Some have been used to address issues in biological and machine learning, and we argue that deep learning research could benefit from building on these ideas. In this context, it is useful to note that it is often possible to see such algorithms as performing gradient descent on some system-internal quantity which it might be useful to minimize, such as the utilization of limited resources. Indeed, including a resource-based loss (minimizing the total magnitude of connection weights, for example) is often combined with externally measured loss. Furthermore, learning rules such as Hebb's rule can be seen as performing a gradient computation; Linsker (1986) showed that his version of Hebb's rule maximizes the tendency of a neuron to respond to correlations among its inputs, subject to resource constraints, and such learning rules can also be understood as performing useful statistical computations, such as principal components analysis and independent components analysis (Bell & Sejnowski, 1997). When coupled with simple local connectivity constraints (for example, the constraint that units at a given position in one layer project only to neurons in a spatially restricted corresponding position in another layer), these approaches have given rise to models that show how features of neuronal responses can arise even from spontaneous random activity without any external sensory experiences at all (Linsker, 1986a,b; Miller, Keller & Stryker, 1989). It seems likely that processes like these continue to play some role in connection adjustment processes in the brain, and it has been suggested that they may have computational advantages as well (O'Reilly, 2001).

To summarize what we have discussed up to this point: Deep neural networks offer mechanisms for performing intelligent tasks that are inspired, but not constrained, by aspects of brain function. They achieve their success by optimizing the strengths of connections and other parameters using gradient descent in a measure of their success in performing some designated task. They exploit a number of enhancements over what we might call vanilla neural computation, including convolution, learned gating, and recurrence, all factors that are likely to be exploited (albeit perhaps in not exactly the same form) by biological learning systems.

Deep Learning and Human Learning and Memory

We next turn to a consideration of how learning in deep neural networks relates to issues and topics in human learning and memory. As we shall see, almost all aspects of human memory can be thought of as being influenced by a gradual learning process in real biological neural networks that are often simulated using deep neural networks.

Developmental learning and gradual acquisition of expertise

Learning via gradient descent is an intrinsically slow process, typically requiring many thousands to millions of training examples to fully capture the knowledge contained in a natural domain of experiences. As such, deep learning might seem like a poor model of human memory as it is traditionally studied in the laboratory, where participants' ability to learn and remember materials presented within a brief period of time is assessed. Indeed, as one of us has argued, deep learning might be a better model of the very gradual process whereby cognitive intuitions and abilities are acquired over developmental time or through immersion in a domain leading to the gradual emergence of expertise (McClelland, McNaughton & O'Reilly, 1995). Consider the ability to use our hands to grasp and manipulate objects; to control our bodies to walk, run, and dance; to comprehend and produce language; to read and to write; to intuitively understand physical relationships and the effects of our actions on the world; and to perform music or any other specialized skill expertly. All of these abilities appear to be acquired very gradually through very extensive exposure, making the acquisition of such systems of knowledge a plausible candidate for modeling with multi-layer neural networks that learn slowly via many small connection weight adjustments. Knowledge of the semantics of natural kinds develops slowly, too, over the first decade of life, making it another candidate for modeling using a deep learning system.

Shortly after the introduction of backpropagation, many researchers began to use it to address these aspects of human learning and development. Early applications included models that learned to map from inputs representing spellings of words to outputs representing their sounds (Sejnowsky and Rosenblatt, 1987, Seidenberg & Mcclelland, 1989), or to map from and input representing a physical situation such as the state of a balance scale to the outcome of the situation, in this case which side of the scale would go down (McClelland, 1989). Elman's (1990) work with simple recurrent networks showed how the paradigm could be extended, using a network's input and it internal state from the previous time step to predict the next element in a sequence, and a network that learned about living things first introduced by Rumelhart and Todd (1993) provided a springboard for an exploration of the development of knowledge in this domain by Rogers and McClelland (2004, 2008). These models were able to capture patterns of human performance and the patterns of developmental progressions seen in human learners, demonstrating their relevance to cognitive and psychological questions. According to this work, the gradual learning that occurs in deep neural networks could correspond to -- or at least capture many aspects of -- the learning process that occurs in human learners as they master cognitive domains of many different kinds over extended time periods.

The effects of brain damage on memory and acquired cognitive abilities further aid us in relating these models to brain structures and brain learning mechanisms. On the one hand, lesions to the hippocampus and overlying areas in the medial temporal lobes of the brain impair the ability to acquire new explicit knowledge quickly but can leave well-established prior knowledge and abilities nearly completely intact. Indeed, patient HM, who lost both medial temporal lobes to surgery to treat intractable epilepsy, was profoundly impaired in the ability to demonstrate learning with standard memory paradigms, and had no recollection of having participated in memory testing sessions even within minutes afterwards (Scoville & Milner, 1957). Yet his ability to read and write, his knowledge of words and their meanings, his semantic and general knowledge, and his intelligence as measured by IQ tests were all intact. On the other hand, damage to the neocortex of the brain tends to produce deficits in acquired abilities, with the content of the deficit depending on the functional role of the specific site of the lesion. As two examples, damage to the fusiform face area produces a deficit in the ability to recognize faces, and damage to the visual word-form area produces a deficit in the ability to read printed words. These findings are consistent with a theory of learning in the brain called *complementary learning systems theory* (Marr, 1971; McClelland, McNaughton & O'Reilly, 1995) which holds that the neocortex is a gradual learning system similar to a deep neural network, and that the hippocampus and related areas in the medial temporal lobes provide a fast learning system that allows humans and other animals to learn arbitrary aspects of new things relatively quickly -- a crucial ability we rely on every time we encounter someone or something new.

We now consider how deep learning may inform our perspective on aspects of memory as it can be studied in the laboratory.

Gradient-based changes as the substrate for implicit memory

According to complementary learning systems theory, each experience a person has gives rise to a pattern of activation distributed widely over many neocortical brain areas. Gradual learning occurs through adjustment of connections among the neurons in the neocortex, and these changes are typically small in magnitude during the learning of arbitrary associative relationships, so that they result in changes only detectable with very sensitive, implicit assessments of learning, such as, for example, measures of the probability of generating a particular word in a word stem completion task. The accumulation of such changes can give rise to the gradual improvement of a cognitive ability over a few laboratory sessions, such as the ability to read mirror reflected text, or the ability to trace a figure while viewing the figure and one's hand in a mirror. Learning in these sorts of tasks is spared in patients with MTL lesions (Schacter, 1987), supporting the view that these functions arise from the changes to connection weights in neocortical networks thought to be like deep learning systems. Apparently, accumulation of the effects of such changes can even give rise to the gradual acquisition of knowledge of facts and word meanings that have come into use since the onset of amnesia, assuming the information is repeated frequently. For example, HM learned to recognize the picture of John F. Kennedy and could report that he had been elected president and then assassinated, all very frequently discussed facts based on events that happened after his surgery (Milner, Corkin & Teuber, 1968). Priming effects observed in amnesic patients from a single encounter with a picture or word may also be due to such changes. According to this way of thinking, such priming effects might correspond to the after-effects of changes in neocortical connections arising from an individual experience.   These effects tend to decay fairly rapidly after exposure, suggesting that connection weight changes in the neural networks in the brain may have a fast-decaying component accounting for effects of immediate repetition, and a slower-decaying component capturing the gradual acquisition of a new skill or oft-repeated element of new semantic content (McClelland & Rumelhart, 1985).

According to complementary learning systems theory, gradual-learning based changes in the neocortex also influence performance in two other kinds of memory tasks, which we now discuss.

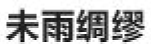
Immediate memory tasks

In some classical approaches to understanding memory, researchers have posited the existence of short term and/or working memory systems, sometimes conceived as containing a fixed number of slots (Atkinson & Shiffrin, 1968; Luck and Vogel, 1997), each holding a single item. In a neural network modeling context, it is more natural to think of memory in terms either of sustained activation within a neural network, or in terms of rapid changes in connection weights. Indeed, accounts of both types have been offered (e.g., Burgess & Hitch, 1999; Botvinick & Plaut, 2006). Importantly, the amount of information that an individual can retain for immediate recall appears to be crucially dependent on item familiarity, sequential statistical relationships between list elements, and the meaningfulness and coherence of the material to be remembered, and patients with cortical degeneration affecting semantic knowledge representations show deficits in exploiting item familiarity and meaningfulness effects (Patterson, Graham & Hodges, 1994). Thus, our ability to retain information in immediate memory is highly dependent on experience, and is generally thought to reflect recurrent computations that constrain neural activity patterns to conform to lexical, semantic, and syntactic constraints that have been learned through the gradual, frequency-dependent learning process that occurs in deep neural networks. Botvinick and Plaut (2006) simulate a wide range of aspects of immediate serial recall of short lists of items using the activation state within a generic recurrent neural network. Their model's success in accounting for human performance data supports the view that it may be useful to think of immediate serial recall and related verbatim verbal memory tasks as reflecting sustained activation based on a gradual learning process in a recurrent neural network, affected by experience statistics and drawing on connection based knowledge in a neocortex-like slow learning system. Indeed, as Botvinick & Plaut discuss, immediate serial recall within what is called the span of immediate memory seems not to be much affected by MTL lesions, though beyond the span of immediate memory, the MTL does appear to play a role. Notably, deep learning systems used for language tasks and many other types of tasks rely on maintained activation states dependent on gradual learning in recurrent neural networks as well. In all of these systems the activation state of the system can be viewed as a form of learning-dependent working memory.

Episodic memory tasks

In some theoretical frameworks, episodic memory and semantic memory are considered quite distinct, with memory for specific episodes thought to depend more crucially on the medial temporal lobe (though the MTL is usually thought to be important for initial acquisition of new semantic memories, see e.g. Squire, 1992). However, a newer perspective holds that episodic and semantic memory are intimately intertwined (Renoult, Irish, Moscovitch & Rugg, 2019). Here we highlight two crucial ways in which the neocortical and MTL learning systems work together in tasks that are typically thought of as episodic memory tasks.

First, it is important to understand that the input to the hippocampus is not raw sensory input; instead, the hippocampus receives input predominantly from cortical association areas, in which the brain representations are heavily dependent on gradual learning of the kind attributed to the neocortex in complementary learning systems theory. The consequence of this is that, for example, an individual who grew up in a Chinese cultural tradition and who had learned to read and write Chinese would likely have a richly elaborated pattern of phonological and semantic activity in their neocortical association areas upon processing the input below, whereas someone growing up in an English language cultural tradition would only perceive a sequence of four unrecognizable characters:



未雨绸缪

This sequence corresponds to a meaningful Chinese proverb, literally meaning ‘fix your house before the rain comes’ or more proverbially, prepare while you can for foreseeable adversities. According to the complementary learning systems theory, what will be available for storage after someone who is Chinese experiences the above sequence of characters will be quite different, compared to someone who does not understand either Chinese writing or culture. Even if written in a language that a participant in a memory experiment understands, an input that is not consistent with the learner’s pattern of prior experience or which lacks sufficient context to make sense will not result in a coherent understanding, and will not, therefore, be easy to remember (Bartlett, 1932/1995). Thus, what is available to the hippocampus depends critically on gradually acquired knowledge in deep neocortical networks.

Second, the hippocampus and MTL appear to work together synergistically in recall, making it far easier to learn, for example, pairs of words with strong pre-existing associations (locomotive-brakeman) than pairs of words with weak or no pre-existing associations (city-ostrich) (Cutting, 1978). The effect may be due in part to greater meaningfulness of related pairs resulting in better initial storage, but is likely to reflect the neocortical system’s prior knowledge of the relevant associative relationships complementing the retrieval of the association formed in the MTL (Kwok, 2003).

Complementing gradient-based learning systems with a hippocampus-like external memory

We end this section by noting that models of human and animal learning and memory as well as deep learning models in AI make extensive use of complementary learning systems-like architectures. As was noted first by McCluskey & Cohen (1989), deep networks struggle to integrate arbitrary new information quickly without interference with existing knowledge. To learn new things quickly without interference, a complementary learning system such as is provided by the hippocampus is needed (McClelland *et al.,* 1995). This issue is widely appreciated in the deep learning community, where a number of approaches have been introduced to overcome it. Deep learning models often employ rehearsal buffers (as introduced in Mnih *et al.,* 2015) or sophisticated external memory systems (as introduced in Graves *et al.,* 2016) inspired in part by the human hippocampus. These systems are used together with cortex-like deep neural networks, allowing Mnih *et al.* to achieve human level performance in the Atari suite of computer games and allowing Graves *et al.* to solve complex planning problems. For further discussion, see Kumaran, Hassabis & McClelland (2016). The issue remains an active area of ongoing research in the deep learning community. In models addressing biological learning and memory, there has also been extensive exploration of the roles of complementary learning systems. We mention two highlights from recent work here. First, it is now generally accepted that rapid statistical learning (e.g., from exposure 2 minutes of exposure to sequences of syllables with strong sequential dependencies) depends on the fast learning system in the hippocampus, as modeled by Shapiro *et al.* (2017). Second, there has been considerable interest over the last decade in the possibility that new information consistent with pre-existing schemas can be rapidly integrated into neocortical neural networks (Tse *et al.,* 2007), and computational investigations have now begun to explore the conditions under which this kind of rapid integration can occur (McClelland, McNaughton, & Lampinen, 2020).

Deep Reinforcement Learning

The discussion so far has focused on learning in situations where a neural network seeks to match some form of externally provided information, such as a classification response provided by an external teacher, or the next item in the input stream. If the system generates the wrong output, its parameters can be adjusted to move its output closer to the target.

While this form of learning has been a primary focus of deep learning research for most of its history, recent work has begun to focus also on a second area referred to as *reinforcement learning* (RL). Here again the learning system accepts inputs and produces outputs, but instead of receiving explicit output targets, as in supervised learning, the feedback signal takes the form of a single number, which scores how ‘good’ the output was. This numerical feedback signal is typically referred to as a ‘reward’ signal, with larger numbers indicating higher reward. The goal of learning is to select outputs that maximize the rewards received.

One fundamental difference between RL and supervised learning is the need for exploration. Even when an output yields a large quantity of reward, there is still the possibility that some other action might generate even more, and the only way to find out is to explore alternative actions. However, there comes a point where exploration is wasteful, and it makes more sense to leverage the information that has been gathered so far. Negotiating this balance between exploration and 'exploitation' is a central problem in RL.

In a further contrast with supervised learning, RL research has tended to focus on scenarios where the system's outputs are understood as actions that alter the environment and thus determine the system's next 'sensory' input, completing a sensory-motor loop. One ramification of this framing is that reward can be accrued not only through single actions, but through sequences of action, putting a focus on sequential decision-making problems.

It should be clearly noted that RL, by its definition, has no inherent connection with deep learning, and indeed RL research began as quite a separate endeavor. The development of computational algorithms for RL began in the 1980's, directly inspired by questions arising from animal conditioning research (Sutton & Barto, 1981). These early investigations gave rise to the *temporal difference* (TD) approach that is still central to most RL procedures (Sutton & Barto, 2018). Here, the system or 'agent' learns, for each input or state, an estimate of expected future reward, referred to as a 'state value.' An action is then selected and the agent takes note of both the reward received and the state value for the outcome state in which it arrives. A bit of simple arithmetic, based on these quantities, yields a *reward prediction error* (RPE) signal, indicating whether things turned out better or worse than was predicted based on the initial state. This RPE is then used to drive learning: Actions that yield positive RPEs are up-weighted so as to be selected more reliably in future; actions that yield negative RPEs are down-weighted.

The initial link between RL and animal learning was fortified in the 1990s, when it was noted that midbrain dopamine release displays a temporal profile quite closely matching what would be expected of an RPE signal (Montague. Dayan & Sejnowski, 1996). As we will discuss further below, RL has gone on to provide a rich framework for understanding learning and decision making in humans and other animals, as well as the underlying neural mechanisms.

Despite its interest, however, RL research up until around 2015 had one important limitation, which was connected to the issue of 'state representation.' The vast majority of RL algorithms required the current state or situation to be represented simply as an entry in a look-up table, or at most as a set of linearly combinable features. This made it impossible to select actions or compute value based on richer non-linear transformations of input, as would be necessary, for example, to deal with pixel-level visual inputs or relationally rich inputs such as board positions in chess.

Of course, as discussed earlier, learning useful non-linear transformations of input representations is exactly the focus in deep learning, and an aspiration arose early in the history of RL research to use deep neural networks to learn rich representations on top of which RL-driven action selection could operate. Some early success in pursuing this goal was attained in now-classic work on backgammon (Tesauro, 1994), but more typically the combination of RL with deep learning was found to be unstable, with learning diverging into failure modes rather than converging toward desired patterns of behavior.

A breakthrough occurred around 2015, in work that successfully applied "deep RL" -- as the combination of deep learning and RL has come to be called -- to a suite of classic Atari video games (Mnih et al., 2015). Here, a standard convolutional neural network took input in the form of pixel-level representations of the video-game screen, and output actions corresponding to joystick operations. Rewards corresponded to points in the game. A standard TD algorithm was used to compute RPE signals, and these were used to up- and down-weight actions as usual. However, this was accomplished through backpropagation: RPE signals were combined with backpropagation to adjust all of the weights in the network so as to up-weight actions yielding a positive RPE and down-weight actions yielding a negative one. Through gradual adjustments along these lines, the system was able to reach superhuman levels of performance on a large set of Atari games, often finding surprising solutions that to a human eye could even appear creative.

One ingredient that contributed to the more stable learning obtained in this Atari work was *experience replay*, using a hippocampus-like rehearsal buffer, as mentioned above. Action-outcome pairs were stored in memory and re-presented as training examples, alongside new experiences, throughout the learning process (a maneuver inspired by replay events observed in the brain). This kept training from focusing too exclusively on very recent experiences, avoiding the instability associated with earlier efforts.

This initial breakthrough work on deep RL has been followed by an explosion in deep RL research, which has extended the paradigm in a number of fundamental ways. One key development has involved the integration of *model-based* methods. The initial work on Atari had adopted a *model-free* strategy, learning to select actions directly from a representation of the current state. Model-based RL, in contrast, leverages a 'forward model' of the environment, which given an initial state and an action specifies the probabilities of outcome states. Such a model can be used for planning, i.e., explicit search over action-outcome trajectories just before selecting an action. Over the last few years, deep learning has been integrated into model-based RL in at least two ways. First, it has been used to recommend actions based on model-free learning, with these recommendations then being used to guide model-based search. This coordinated use of model-based and model-free learning formed the foundation for AlphaGo, a system that learned through RL to play the ancient board game of go at a super-human level (Silver et al., 2016). Second, deep learning has been used to learn the forward model itself, based on observed action-outcome associations (Schrittwieser et al., 2019).

The initial work on deep RL relied on ‘end-to-end’ RL, with backprop updates – and therefore the learning of internal representations -- driven solely by the RPE signal. However, there are two significant pitfalls associated with this end-to-end approach. First, in many problem domains, reward is ‘sparse’; that is, reward is only received rarely. This is true, for example, in board games like go, where reward (positive or negative) is only received at the end of the game. When reward is sparse, in this sense, it means that an RL agent can cast about for a long time without learning anything at all. A second problem with end-to-end learning is that it results in internal representations that are closely tied to a particular task, and which may therefore not be ideal for other tasks. These considerations have motivated work in which deep RL has incorporated ‘auxiliary’ learning signals alongside those arising from task-based reward. One example is an additional reward intended to drive exploration, often linked to novelty, surprise or successful control of the environment, independent of task-specific goals (e.g., Badia et al., 2020). Another common form of auxiliary learning signal encourages the agent to predict the outcomes of its actions, independent of whether they yield reward (e.g., Wayne et al., 2020; van den Oord, Li & Vinyals, 2018). This kind of learning, which extends the prediction-based learning often used in language modeling by including the agent’s action as part of the condition for the prediction, could be used for model-based RL, but even when it is not it has been found to engender rich internal representations useful for model-free RL.

Deep RL has been successfully applied in agent architectures containing recurrent neural networks including the long short-term memory networks introduced earlier. In this setting, RL can drive learning of working memory-like internal representations that carry forward action-relevant information about past events. One interesting side effect of this kind of learning is that it can give rise to deep RL systems that develop the ability to adapt their behavior rapidly based on reward feedback (Wang et al., 2016; see also Finn, Abbeel, Levine, 2017). The leveraging of past experience to accelerate new learning is referred to in machine learning as ‘meta-learning’. However, not surprisingly, the idea originates from psychology, where it has been called ‘learning to learn.’ Here, we describe the psychological principle and present a deep RL model that captures this behavior.

In the first paper to use this term, Harlow (1949) presented an experiment that captures the principle neatly (Figure 3, left). Here, monkeys were presented with two unfamiliar objects, and permitted to grab one of them. Beneath lay either a food reward or an empty well. The objects were then placed before the animal again, possibly left–right reversed, and the procedure was repeated for a total of six rounds. Two new and unfamiliar objects were then substituted in, and another six trials ensued with these objects. Then another pair of objects, and so forth. Across many object pairs, the animals were able to figure out that a simple rule always held:

one object yielded food and the other did not, regardless of left–right position. Thus, after training, when a monkey was presented with a new pair of objects it was able to learn in one shot which the preferable object was, providing a simple but vivid example of learning to learn. This pattern of behavior is shown in the middle panel, with the animals' 'insight' reflected most directly in their excellent performance on the second trial in each six-trial block.

Building on earlier work by Hochreiter (2001), Wang and colleagues (2018) showed that this same kind of learning to learn arises in a recurrent neural network trained on a task with the same logic as the one employed by Harlow (1949). Specifically, an LSTM network was trained, using reinforcement learning and backpropagation, to select between novel pairs of images, with the same six-trial block structure and rule-based pattern of reward. After training on a long series of different image pairs, the weights in the network were held fixed and a new image pair was presented for six trials. As shown in the right panel of the figure, under these conditions the network behaved very much like the monkeys in Harlow's experiment. In effect, reinforcement learning and backpropagation had 'tuned' the weights in the recurrent network in such a way that the *activation patterns* in the network were able to store information about which image had been selected on recent trials, and whether reward had been received. As discussed earlier, such an internal pattern of activity constitutes a form or working memory, one that the network learned to form, and then to use to guide adaptive behavior without weight updates.

In summary, the original RL algorithm had given birth to a second, emergent learning algorithm, implemented in the network's activation dynamics, an instance of learning to learn, or in this case, meta-reinforcement learning. This meta-reinforcement learning effect mirrors the working-memory based reward learning that has been observed in some recent investigations of human learning (Collins & Frank, 2012).

Deep RL: Implications for Psychology

Like deep learning at large, deep RL has interesting and wide-ranging implications for psychology and neuroscience. Deep RL of course brings together two computational frameworks that independently have tight preexisting links with psychology and neuroscience: deep learning, with its ties to representation learning in biological neural systems, and RL which as we have noted was inspired by animal learning phenomena and has been linked to dopaminergic function. Beyond this, a number of techniques that have been added to the deep RL toolbox, such as memory replay and model-based RL, have immediate connections with -- and in some cases were directly inspired by -- animal behavior or neuroscience (see e.g., Gershman & Daw, 2017).

However, while deep learning and RL have their own independent implications for psychology and neuroscience, deep RL is more than the sum of its parts: It raises issues that are not raised by either of its two ingredients alone. A good example is the meta-reinforcement learning phenomenon mentioned above, which has recently been used to account for a diverse range of behavioral observations and neuroscientific findings relating to the prefrontal cortex (Wang et al., 2018). We have already discussed how meta-reinforcement learning provided a novel explanation for the meta-learning effect observed by Harlow (1949). To take just one example on the neuroscience side, Wang and colleagues (2018) showed that the receptive fields of macaque prefrontal neurons during a reward-based learning task were closely mirrored by the response patterns of units inside a recurrent network trained on the same task.

A second example of a novel hypothesis suggested by deep RL is provided by recent work on 'distributional RL.' Here, the value and RPE signals from standard TD learning are replaced by richer, multi-dimensional representations capturing the full probability distribution over rewards and prediction-error signals (Dabney et al, 2018). Elaborating TD learning in this way enhances learning in deep RL systems by driving the emergence of richer internal representations, an effect not seen when the same approach is applied to RL systems that do not involve deep learning (Lyle, Bellemare & Castro, 2019). In recent experimental neuroscience work, evidence has been obtained that the mammalian dopamine system may use a distributional representation, as in distributional RL (Dabney et al., 2020).

While the field of deep RL is relatively new, the logic of the approach together with the initial steps we have described linking deep RL with behavioral and neuroscience findings suggests that it may provide a useful source of new hypotheses for how reward-based representation learning occurs in the brain, and how this manifests in behavioral patterns of learning and decision-making.

Discussion

In this article, we have considered the role of deep neural networks as models of human learning and memory. These models trace their roots to Hebb (1949), Rosenblatt (1962), and the PDP movement of the 1980s, and have become a major focus of research in artificial intelligence over the last decade. As researchers who have been involved in this tradition for decades, we see renewed motivation for ideas originating a quarter of a century ago, as well as exciting new directions that expand the fertile ground for productive interplay between research in artificial intelligence and research on biologically-based cognitive systems in animals and man and the biological implementation of such systems. We have seen how neural networks can be applied to the effort to understand many aspects of learning and memory, ranging from immediate recall of short lists of items to the gradual emergence of semantic knowledge during human development. We have seen how deep learning models exploit complementary learning systems to enhance their effectiveness at solving cognitively challenging tasks, and we have seen how gradient-descent based learning in advanced architectures with external memory demonstrate the potential for learning to shape what we store in memory and how we search memory to find what we need to perform skilled tasks. Building on Deep Reinforcement Learning, we have seen how neural network models can learn representations that support reward-driven behavior, and how such models can capture important behavioral phenomena such as the ability to learn to learn. We see an exciting prospect for the further development of models that rely on ideas from Deep Learning to inform our search for ways of understanding human learning and memory.

Limitations and Future Directions

Thus far, we have not addressed limitations of current deep learning models. We now turn to this important topic, acknowledging that current models do have limitations in their ability to capture aspects of human intelligence. We see the current state of development of models of intelligence as promising and exciting, but far from fully satisfactory. In spite of successes in creating artificial systems that are in use every day and that exceed human capabilities in games that take decades for human learners to master like chess and go, further progress is needed, and we do not believe it is possible to predict how long it will be before we have models that fully capture human mental abilities. Below we mention two recent challenges that point to areas where further progress is needed, along with tentative suggestions about the prospects for progress and the direction such progress might take. We preface these remarks by acknowledging that some critics of a deep learning approach have long argued that more explicit symbol-processing based approaches may be necessary as alternatives or complements to neural network based approaches (Fodor & Pylyshyn, 1988; Griffiths, Chater, Kemp, Perfors & Tenenbaum, 2010; Lake, Ullman, Tenebaum & Gershman, 2017). We still favor the view that we will achieve the most satisfactory long-term outcomes by continuing to explore how mechanisms that address the shortcomings of current models can arise without building too much structure in (McClelland, Botvinick, Noelle, Plaut, Rogers, Seidenberg & Smith, 2010), and we are excited that so many brilliant scientists are now exploring this approach.

One-shot learning. The first challenge we consider arises when we first see an instance of a new concept or type of object -- perhaps a new kind of vehicle such as a segway, or a character in an unfamiliar alphabet. After using the segway as a motivating example, Lake, Salakhutdinov, and Tenenbaum (2015) investigated the character example more fully, and demonstrated that humans were far better at several tasks after seeing a single example character than and of a set of neural-network models they considered. Among other things, after seeing an example of a character generated by one person, human adults were better at finding a second example of the character or generating new examples of the character than the neural networks.

Acquiring and adapting new skills quickly from observation and instruction. The second challenge arises when we first learn a new task or adapt our goals in a familiar task based on instructions. Lake *et al. (*2017) presented this challenge in the context of the videogame *Frostbite*, noting that, after reading the instructions on how to play the game, watching a couple of minutes of expert play, and then playing for just 15-20 minutes, humans achieved scores comparable to what improved versions of the network used by Mnih *et al* (2015) could achieve after thousands to millions of games. Furthermore, Lake *et al.* suggested a number of altered goals a human who knew how to play the game could readily adopt, such as trying to lose, trying to get as close as possible to a particular score without going over, or trying to get a score just slightly better than a friend's score.

We agree that these challenges address aspects of human intelligence not yet fully captured by Deep Learning systems. Below, we consider some what it will take to address the core issues behind these challenges.

Giving networks the same inductive biases that humans bring to a task. When humans learn how to perform a task, they rely on what are called *inductive biases* that constrain the space of possible solutions to a problem, allowing for rapid learning. Depending on their architecture and the optimization objective applied during learning, deep learning models can have strong inductive biases. Convolutional networks provide a clear example, since their structure builds in an inductive bias to assume the statistical structure of images remains the same even when images are translated (i.e., shifted) vertically and horizontally. As in other cases, the inductive bias in this case can be thought of as corresponding to one that in animals would have derived from evolution.

Inductive biases may also be provided by the resources we use to generate particular behaviors and by experience. Consider, for example, the task of producing a new example of a character from an human alphabet. Humans perceive characters using their visual systems, and produce characters by using their hands to guide a writing implement -- in the case of the Lake *et al.* (2015) study, a mouse -- to create a series of strokes to form a character. The neural networks Lake *et al.* considered did not have most of these characteristics. Their networks relied on simulated vision systems that processed characters as static images (arrays of pixel intensities, as in the digit recognition model we used to introduce deep networks). These networks have built-in visual biases including convolutional layers, but they lack the inductive bias that characters are formed out of a series of strokes that can be made by a hand using a writing implement. In fact, Lake *et al.* captured their own findings by creating an explicitly structured symbolic model that treats characters as hierarchically structured assemblies of strokes. Given that writing is a recent human adaptation that relies on characteristics of the human body (especially the human hand), it seems natural to explore the possibility that experience using the hand to control an implement to write could lead to connection-based knowledge with similar functional characteristics. To achieve human level performance in the various tasks Lake *et al.* (2015) demonstrated humans could perform without building in an explicit, character-specific hierarchical generative model in advance is a good challenge for deep learning research. We suggest that an important part of the solution may be to use networks that learn to produce drawings and characters using a human-like arm and hand controlling a pen or mouse. For a neural network system that suggests the promise of this approach, using a pen controlled by specifying stroke sequences, see Ganin *et al.* (2018).

Embedding networks that learn specific tasks in larger systems that perform many different tasks. One of the ways that most neural networks differ from humans is that a given neural network is typically trained on just one task or closely related suite of tasks. Highly successful convolutional networks for image classification, for example, are trained only to classify images, and Deep RL systems have generally been trained to perform only one task at a time. In Mnih *et al.* (2015) for example, the same architecture was used to play each of many different games, but a freshly initialized instance of this architecture was trained for each game. Similarly, completely distinct instances of the Alpha0 system (Silver *et al.,* 2018) are used to achieve superhuman performance at the separate games of chess, go, and the Japanese game shogi. This makes these systems completely unlike adult humans, who bring a lifetime of relevant experience to learning a new game such as Frostbite. In order to create systems that are able to learn a new game quickly from instruction, then, deep learning research will need to move toward using single systems that can effectively share what they have learned in each of many different tasks to help provide the starting place for learning new tasks (see Parisi *et al.*, 2019). Building such a system is an important step toward artificial general intelligence -- one that may also bring us closer to having neural networks that capture some of the aspects of the human ability to learn new tasks quickly from very limited experience.

Building networks that learn to follow open-ended instructions. Another way neural networks differ from people is that they do not generally make use of the open-ended medium of language to provide a way of guiding their performance and learning in a particular task. Throughout this article we have mentioned many networks, but virtually none of them rely on open-ended task specifications that can be provided through natural language. In this respect, they are really far more like non-human animal learners, or perhaps human learners brought up in total isolation from any form of interpersonal communication. To us, this is a profound limitation facing most Deep Learning systems -- one that will definitely have to be addressed before we can say we have a system that has the ability to learn and perform in all of the ways that humans can.

Some promising steps have recently been taken toward addressing this limitation. As one example, Radford *et al.* (2019) have recently described a neural network language processing system that can in principle learn an open ended range of language based tasks based on learning from texts that include examples of instances of language processing tasks, such as summarization and a wide range of different types of text-conditioned question answering tasks. A fertile extension of this general approach would be to train deep learning systems that live, play, and learn in a (perhaps virtual) environment that they share with humans, engaging in a give-and-take with others who can guide the behavior through discourse about their activities in the shared environment. Some steps moving part way in this direction are described in Hill *et al.* (2020).

Metacognition and discourse with others that can shape intelligent behavior. FInally, we make note of the human ability to think about our own behavior and exchange information with others -- information that we can put to use to increase our effectiveness. As one example, if someone is just learning English, one can tell them that the past tense of a verb is formed by adding 'ed', and this information can immediately be put to use, though nuanced mastery of the English past tense goes far beyond this simple 'rule', which should not be confused with expert language understanding. Furthermore, humans go beyond mere rule-following, often engaging in reflection on the tasks they seek to master, creating helpful guides for their own behavior. The developmental psychologist Annette Karmiloff-Smith was an early proponent of the view that humans rely on neural-network like processing systems complemented by metacognitive systems that observe and regulate the neural network (Clark & Karmiloff-Smith, 1993), and Anders Ericsson, who has spent his career studying the psychology of expertise has long argued that self-reflection and self-directed practice together with expert guidance by world-class coaches and role models are very important contributors to the emergence of world-class performance in all walks of life, ranging from athletics, to music, and to science and technology. It is likely that, to achieve human level performance, Deep Learning systems will need to exploit these characteristics of natural intelligence, and there are now several exciting new directions in deep learning research that seek to capture some aspects of these abilities (e.g., Lampinen & McClelland, 2019).

Final Comments and Conclusion

Given that the brain is a neural network, researchers at least since Hebb (1949) have sought to understand how human learning and memory might emerge from processes taking place in neural network-like systems. It was once possible, however, to argue that neural networks only address the implementation of cognitive abilities in brains, and that the true nature of such abilities needed to be sought in the characteristics of symbolic systems (Fodor & Pylyshyn, 1988). The unprecedented successes of deep learning models in the field of artificial intelligence have rekindled interest in thinking about the idea that mental abilities might arise as consequences of processing and learning in a neural network. In this article we have considered such systems, and we have seen how they can address human and animal performance in a wide range of different types of learning and memory tasks. We hope we have provided readers interested in human learning and memory with a starting place for understanding of some of the developments that have occurred in this field, and we hope we have indicated some of the directions research in this field must take to fully capture human learning and memory abilities. In closing, we emphasize that the effort to understand the nature of specifically human intelligence remains an important endeavor, quite apart from what can be achieved in artificial systems. Research that draws inspiration from research on artificial systems while still focusing on human performance and its biological substrate will be crucial to the long-term goal of understanding how learning and memory arise in natural biological systems, including humans.

References

Atkinson, R. C., & Shiffrin, R. M. (1968). Human Memory: A Proposed System and its Control Processes. In *Psychology of Learning and Motivation* (Vol. 2, pp. 89-195). Academic Press.

Badia, A.P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A. and Blundell, C. (2020). Never Give Up: Learning Directed Exploration Strategies. *arXiv preprint* arXiv:2002.06038.

Bartlett, F. C. (1932/1995). *Remembering: A study in experimental and social psychology*. Cambridge University Press.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, *7*(6), 1129-1159.

Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: a recurrent neural network model. *Psychological review, 113*(2), 201-233.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement Learning, Fast and Slow. *Trends in Cognitive Sciences, 23(*5).

Burgess, N., & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review, 106*, 551–581.

Clark, A., & Karmiloff-Smith, A. (1993). The cognizer's innards: A psychological and philosophical perspective on the development of thought. *Mind & Language*, *8*(4), 487-519.

Collins, A. G., & Frank, M. J. (2012). How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *European Journal of Neuroscience*, *35*(7), 1024-1035.

Cutting, J. (1978). A cognitive approach to Korsakoff's syndrome. *Cortex, 14,* 485-495.

Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., & Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. *Nature*, 1-5.

Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018, April). Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179-211.

Ericsson, K. A., & Charness, N. (1994). Expert performance: Its structure and acquisition. *American psychologist*, *49*(8), 725-747.

Finn, C., Abbeel, P., & Levine, S. (2017, August). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1126-1135). JMLR. org.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1-2), 3-71.

Ganin, Y., Kulkarni, T., Babuschkin, I., Eslami, S. M., & Vinyals, O. (2018). Synthesizing programs for images using reinforced adversarial learning. *arXiv preprint arXiv:1804.01118*.

Gershman, S. J., & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual review of psychology*, *68*, 101-128.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., ... & Badia, A. P. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, *538*(7626), 471-476.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in cognitive sciences*, *14*(8), 357-364.

Harlow, H. F. (1949). The formation of learning sets. *Psychological review*, *56*(1), 51.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hebb, D. O. (1949/2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

Hill, F., Lampinen, A., Schneider, R., Clark, S., Botvinick, M., McClelland, J. L., & Santoro, A. (2019). Emergent systematic generalization in a situated agent. *arXiv preprint arXiv:1910.00571*.

Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001, August). Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks* (pp. 87-94). Springer, Berlin, Heidelberg.

Karpathy, A. (2015) The unreasonable effectiveness of recurrent neural networks. A section of *Andrej Kaparthy's Blog.* http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in cognitive sciences*, *20*(7), 512-534.

Kwok, K. (2203). A computational investigation of the successes and failures of semantic learning in normal humans and amnesics. PhD dissertation, Department of Psychology, Carnegie Mellon University.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, *40*.

Lampinen, A. K., & McClelland, J. L. (2019). Zero-shot task adaptation by homoiconic meta-mapping. *arXiv preprint arXiv:1905.09950*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444.

Linsker, R. (1986). From basic network principles to neural architecture: Emergence of orientation columns. *Proceedings of the National Academy of Sciences*, *83*(22), 8779-8783.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*(6657), 279-281.

Lyle, C., Bellemare, M. G., & Castro, P. S. (2019, July). A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 4504-4511).

McClelland, J. L. (2013). Integrating probabilistic models of perception and interactive neural networks: A historical and tutorial review. *Frontiers in Psychology, 4*, 503.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., & Smith, L. B. (2010). Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends in cognitive sciences*, *14*(8), 348-356.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review, 102*(3), 419-459.

McClelland, J. L., McNaughton, B. L., & Lampinen, A. K. (2020). Integration of New Information in memory: New insights from a complementary learning systems perspective. *Philosophical Transactions of the Royal Society B.*, 375: 20190637.

McClelland, J. L., & Rumelhart, D. E. (1988). Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises. Cambridge, MA: MIT Press. Online edition available at https://web.stanford.edu/group/pdplab/pdphandbook.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (Vol. 24, pp. 109-165). Academic Press.

Miller, K. D., Keller, J. B., & Stryker, M. P. (1989). Ocular dominance column development: analysis and simulation. *Science*, *245*(4918), 605-615.

Milner, B., Corkin, S., & Teuber, H. L. (1968). Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of HM. *Neuropsychologia*, *6*(3), 215-234.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), pp.529-533.

Montague, P. R., Dayan, P., & Sejnowski, T. J. (1996). A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *Journal of neuroscience*, *16*(5), 1936-1947.

Olah, C. Understanding LSTM networks (2015). Section of *Colah's Blog.* https://colah.github.io/posts/2015-08-Understanding-LSTMs/,

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv preprint arXiv:1807.03748* (2018).

O'Reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and Hebbian learning. *Neural computation*, *13*(6), 1199-1241.

Patterson, K., Graham, N., & Hodges, J. R. (1994). The impact of semantic memory loss on phonological representations. *Journal of cognitive neuroscience*, *6*(1), 57-69.

Parisi, German I., Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Renoult, L., Irish, M., Moscovitch, M., & Rugg, M. D. (2019). From Knowing to Remembering: The Semantic–Episodic Distinction. *Trends in cognitive sciences*, *23*(12), 1041-1057.

Richards, B. A., & Lillicrap, T. P. (2019). Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, *54*, 28-36.

Rogers, T. T., Lambon Ralph, M. A., Garrard, P., Bozeat, S., McClelland, J. L., Hodges, J. R., & Patterson, K. (2004). Structure and deterioration of semantic memory: a neuropsychological and computational investigation. *Psychological review*, *111*(1), 205.

Rosenblatt, F. (1962). Principles of Neurodynamics, Washington, D. *C, Spartan*.

Rumelhart, D. E., McClelland, J. L. & The PDP Research Group(1986). *Parallel distributed processing: explorations in the microstructure of cognition, Vols I & II*. Cambridge: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533-536.

Schacter, D. L. (1987). Implicit memory: History and current status. *Journal of experimental psychology: Learning, memory, and cognition*, *13*(3), 501.

Schapiro, A. C., Turk-Browne, N. B., Botvinick, M. M., & Norman, K. A. (2017). Complementary learning systems within the hippocampus: a neural network modelling approach to reconciling episodic memory with statistical learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *372*(1711), 20160049.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T. and Lillicrap, T. (2019). Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*.

Scoville, W. B., & Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, *20*(1), 11.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. and Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), p.484.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Lillicrap, T. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140-1144.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: expectation and prediction. *Psychological review*, *88*(2), 135.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tesauro, G. (1994). TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, *6*(2), 215-219.

Tse, D., Langston, R. F., Kakeyama, M., Bethus, I., Spooner, P. A., Wood, E. R., ... & Morris, R. G. (2007). Schemas and memory consolidation. *Science*, *316*(5821), 76-82.

Uszkoreit, J. (2017). Transformer: A novel neural network architecture for language understanding. Google AI Blog, retrieved April 9, 2020. https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Wang, J.X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J.Z., Munos, R., Blundell, C., Kumaran, D. and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, J.X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J.Z., Hassabis, D. and Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature neuroscience*, *21*(6), pp.860-868.

Wayne, G., Hung, C.C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J., Mirowski, P., Leibo, J.Z., Santoro, A. and Gemici, M. (2018) Unsupervised predictive memory in a goal-directed agent. *ArXiv preprint arXiv:1803.10760*.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*(23), 8619-8624.

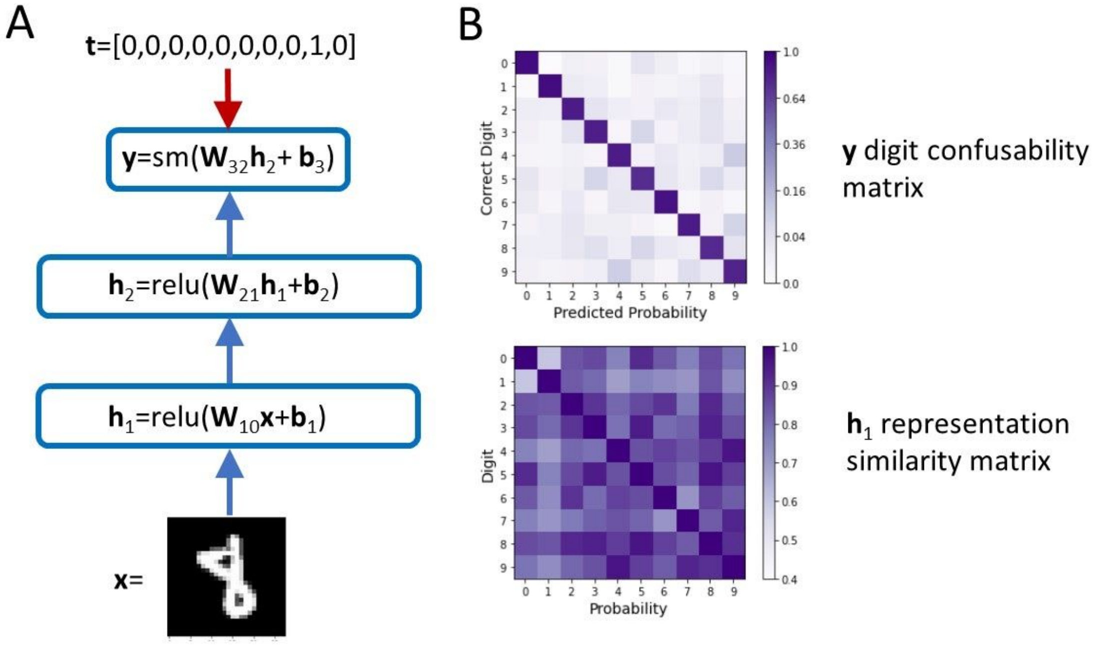

A
$\mathbf{t}$=[0,0,0,0,0,0,0,0,1,0]
$\mathbf{y}=\mathrm{sm}(\mathbf{W}_{32}\mathbf{h}_2+\mathbf{b}_3)$
$\mathbf{h}_2=\mathrm{relu}(\mathbf{W}_{21}\mathbf{h}_1+\mathbf{b}_2)$
$\mathbf{h}_1=\mathrm{relu}(\mathbf{W}_{10}\mathbf{x}+\mathbf{b}_1)$
$\mathbf{x}$=
B
Correct Digit
Predicted Probability
0 1 2 3 4 5 6 7 8 9
1.0
0.64
0.36
0.16
0.04
0.0
$\mathbf{y}$ digit confusability matrix
Digit
Probability
1.0
0.9
0.8
0.7
0.6
0.5
0.4
$\mathbf{h}_1$ representation similarity matrix

Figure 1. A: The deep neural network described in the text for handwritten digit classification. Each input image contains 28 x 38 grayscale values (turned into numbers between 0 and 1). This vector of values is passed through two hidden layers to a third output layer where the softmax function is applied to calculate the network's estimate of the probability that the correct label is each one of the possible digits. The network is trained by providing, with each of 50,000 training images, a teaching signal or target output value, which specifies which of the 10 labels (0 through 9) is correct. B: Output confusability and representation similarity analysis. Bottom: Representation similarities at the first hidden layer after the network was trained for one epoch, or sweep through the training set, and obtained a test accuracy of 90% correct. The mean activation pattern for each of 1000 test examples of each digit is first computed, and then compared to the mean activation pattern of every other digit, using the cosine similarity, which takes a value of 1 when two patterns are identical, and falls off from there as patterns become more different. Top: Digit confusability analysis of the network outputs. For each of the 10 digit classes (one per row), the mean predicted probabilities for each digit label are shown, one in each column. The actual color scale has been adjusted by taking the square root of each predicted probability, so that the network's slight tendency to confuse certain digits can be seen. The largest off-diagonal entry in the table corresponds to a probability of about .1.

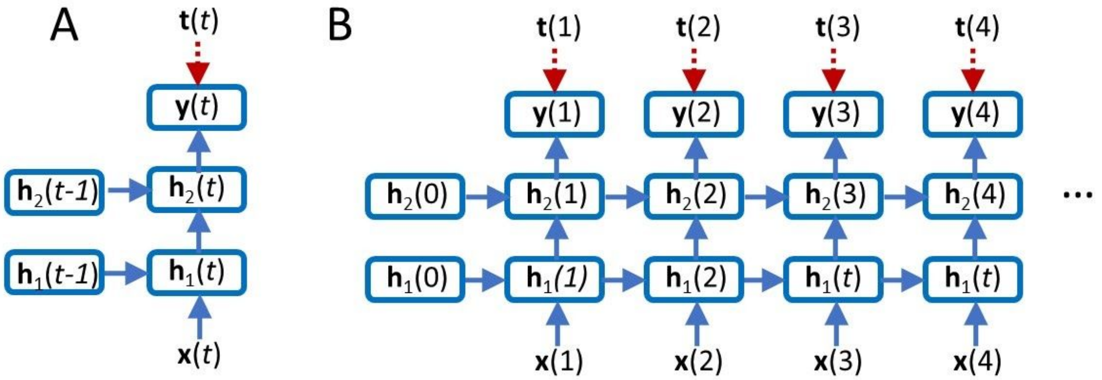


Figure 2. Two visualizations of a recurrent neural network. In A, we show the network as it is used to compute an output **y**(*t*) for the input **x**(*t*), using contextual influences provided by the previous state of each hidden layer, $\mathbf{h}_1$(*t*-1) and $\mathbf{h}_2$(*t*-1). Each blue arrow represents a distinct weight matrix, as described in the text. In B, the network is shown 'unrolled' for 4 time steps. This visualization is useful to see the full set of paths through which error signals can propagate to allow gradient-descent-based learning. For example, the error signal at time step 4, based on the loss computed between the teaching signal **t**(4) and the network output **y**(4), propagates downward through each upward blue arrow and leftward through each rightward blue arrow, specifying weight changes in all of the copies of matrix $\mathbf{W}_{10}$, the matrix mediating the influence of x(t) on h1(t) at each timestep. The total change to $\mathbf{W}_{10}$ is the sum of the changes specified by the back propagated error signal for each of the copies of the matrix.

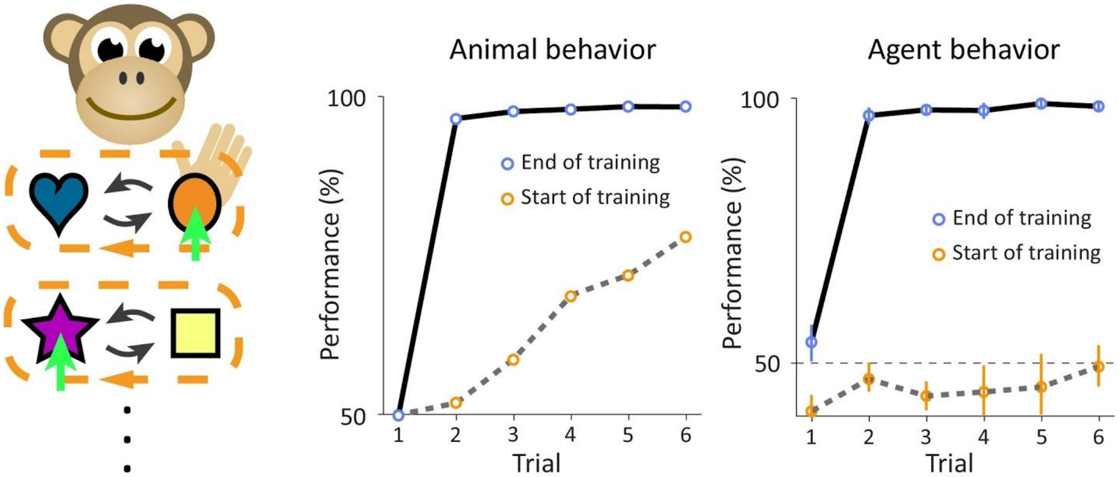


Figure 3. Meta-learning in recurrent neural networks. Left, illustration of the simple meta-learning task used with Harlow's monkeys. Middle, performance of monkeys across the six trials with each new object pair, at the start of training and at the end of training. Right, corresponding performance of the recurrent neural network model of Wang et al (2016). Figure adapted from Botvinick et al. (2019).