INSTITUTE FOR DEFENSE ANALYSES

Rapid Learning in Machines: Challenges and Responses

Brian A. Haugh, *Project Leader*
Patrick W. Langley
Daniel G. Shapiro

April 2021

IDA Document
D-14333

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

About This Publication

This work was conducted by the IDA Systems and Analyses Center under contract HQ0034-14-D-0001, Project AI-5-4458.02, "Artificial Intelligence and Autonomy," for the Office of the Undersecretary of Defense, Research and Engineering. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgements

Joshua Alspector, Kevin Garrison, Arun S. Maiya

For More Information

Brian A. Haugh, Project Leader
bhaugh@ida.org, 703-845-6678

Margaret E. Myers, Director, Information Technology and Systems Division
mmyers@ida.org, 703-578-2782

Copyright Notice

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-14333

Rapid Learning in Machines: Challenges and Responses

Brian A. Haugh, *Project Leader*

Patrick W. Langley

Daniel G. Shapiro

Executive Summary

Over the past four decades, the field of machine learning has produced deep insights into the nature of induction, as well as compelling applications for business and government. Interest in this topic has exploded in recent years, due to breakthroughs in computational methods, increased processing power, and the growing availability of large data sets. Much of the excitement has focused on learning in deep neural networks, which have produced substantial performance improvements in computer vision, speech recognition, language translation, and other arenas. However, this class of techniques typically relies on very large training sets, which are not available for all application areas. In response, researchers are developing new methods for deep learning that are less dependent on massive data repositories.

In this paper, we review the history of work in machine induction and draw lessons for the specialized but important task of rapid learning from small and moderately sized sample sets. We discuss the notion of learning as a search for models that fit the training cases, and we note two distinct paradigms: search through a space of model structures (which characterizes much of the early work in machine induction) and search through a parameter space (which includes deep learning methods). Because the older tradition emphasizes induction from very limited quantities of data, it offers lessons that can benefit ongoing work with deep networks.

We also review techniques that have successfully increased the rate of learning from limited data, including the use of constrained formalisms, guards against overfitting, feature selection, ensemble methods, background knowledge, and revision of existing models. These themes appear in the deep learning literature, but they were examined more fully in the structure-learning paradigm and many insights carry over to the parameter-learning framework. We also propose the adoption of earlier methodological insights, such as reporting learning curves, running experiments that reveal sources of power, measuring learning rate as a function of task characteristics, and examining tradeoffs among different approaches. In addition, we note that the older tradition offers relevant distinctions: separating learning systems from performance systems removes distractors from the problem statements of zero-shot, one-shot, and few-shot learning while highlighting the importance of reporting learning curves; referencing the bias-variance tradeoff will help explain rapid learning results as it relates the variability in error on new training cases to constraints on induction.

We close with several recommendations for the machine learning community and government agencies that fund its work: increase the diversity of induction methods under study, collect and popularize challenge problems for learning from small training sets, and encourage more systematic scientific studies of rapid learning. We also suggest organizing meetings that bridge the paradigms of structural and parametric learning and funding tutorials and summer schools to foster greater intellectual diversity. These steps should extend the reach of machine learning technology to application areas where training data are difficult to obtain.

Table of Contents

Tables

1. Background and Motivation

Recent years have seen growing excitement about advances in statistical machine learning, not only in university research laboratories but also in industry and government. Considerable attention has focused on deep neural networks, which can involve millions of numeric parameters that are tuned automatically. This approach requires powerful computers with large memories, but improvements in digital hardware have made them tractable, leading to successful applications in speech recognition, visual perception, and language processing, among other tasks. Naturally, these results have encouraged the U.S. Government to explore their use in military, security, and related settings.

Standard applications of deep neural networks rely on very large sets of training cases. Thus, it is not surprising that this paradigm been championed by companies like Google, Facebook, and Amazon, which collect such data repositories (e.g., text, audio, image, and video) as part of their core business models, including information about user behaviors (e.g., purchase records, social media). The approach has also received substantial use in China, which collects large data sets not only for commercial purposes but also to analyze the behavior of its citizens. However, there are many settings in which large training sets, especially ones with labeled cases, are both unavailable and impractical to create. As a result, there are increasing concerns about the reliance on large training sets, and there is a growing academic literature on ways to mitigate it. However, an older body of research addressed similar issues using quite different terminology, and it seems important to bridge this gap.

In the remainder of the paper, we discuss these and related topics. We start by stating the task of machine learning in terms of inputs provided and outputs generated. Next, we turn to the topic of rapid learning from small training sets and common terms used to describe this behavior. We then recount relevant results from both the early history of machine learning and more recent work on deep neural networks. Building on these reviews, we discuss a number of recurring themes shared by the paradigms and some lessons learned from the earlier research that can inform ongoing work. We close with recommendations for the machine learning community and ways the U.S. Government can foster further progress.

2. Statement of the Problem

We should start by reviewing the generic problem of machine learning, which we can state in terms of the inputs available and the outputs produced. We can specify this task as:

Given: A set of training cases *C* related to some performance task *T*;

Given: A performance system *P* that employs knowledge *K* to address *T*;

Given: A space of hypotheses or models for this knowledge;

Find: Knowledge *K′* that lets *P* perform well on *T* for *C* and that generalizes to new test cases.

Every machine learning system contains two components: a learning system that acquires knowledge, and a performance system that utilizes knowledge to accomplish a task. The performance system is best viewed as a constant (the mechanism that transforms task inputs into outputs does not change). Instead, the learning system extracts useful knowledge from training cases, which the performance system employs to improve behavior. Many machine learning systems start with no initial knowledge and effectively random task performance. Others seek to improve task performance given substantial initial knowledge. The most commonly studied performance task is classification, in which one assigns a new case to one of a number of known categories. This requires a classification mechanism that uses the learned model for this purpose. Most work focuses on supervised learning, in which a category label is provided with each training case, but unsupervised variants have also been studied. Learning can also improve performance on tasks that involve reactive control, language processing, and plan generation. In each setting, it is important to distinguish the learning system (which acquires expertise) from the performance system (which uses it), as the mechanisms are distinct. One can combine a given performance mechanism with distinct learning processes to produce different behaviors and vice versa.

Note that this statement of the problem given above does not specify the number of training cases available to drive the learning process. The system might be provided with thousands or millions of labeled instances for each category (or with fewer).[1] Typical machine learning papers report experimental results on training sets of whatever sizes are available. This was not a major issue through the early 1990s, when research focused on learning from hundreds to thousands of cases. However, the mid-1990s saw the emergence of the data-mining movement, which emphasized the collection and use of large data sets to create the most accurate models possible.

[1] In some cases, many training cases may be available but only a subset have labels, which raises different issues that we will not discuss here.

Since then, the advent of the World Wide Web, the widespread adoption of smart phones, the Internet of Things, and other developments have led to even larger collections of data. These technologies have enabled the creation of new data repositories (e.g., of Web pages, news stories, blogs, images, videos) and access to these information sources for purposes of model induction. Combined with substantial growth in storage capacity and impressive increases in processing speed, researchers and developers in both universities and companies have come to concentrate their energies on induction methods that assume the availability of very large training sets.

However, this assumption only holds for certain types of applications. In many other settings, there is little data available, and machine learning must extract what it can from much smaller training sets. This is especially true in some scientific disciplines, where experiments can be time consuming and expensive. A similar situation holds in medicine, where privacy restrictions limit accessibility to hospital databases, rare diseases produce only small data sets, and novel ailments like COVID-19 require rapid response based on limited samples. Similar issues arise in the military, where vision systems must recognize newly deployed weapon systems from limited experience with them and autonomous vehicles must adapt rapidly to changed environments. In contrast, machine learning and data mining have concentrated on large data, which has distracted developers from devising methods that extract as much as possible from small samples. The challenge in such cases is that, because there are fewer training cases to guide search through the space of hypotheses, one must draw on other sources of power to take their place. Fortunately, there is growing recognition that we need more research on induction from small amounts of experience, which we call *rapid learning*.

3. Terminology for Rapid Learning

We cannot discuss the target of rapid learning without vocabulary to describe this behavior. The past decade has seen researchers introduce a number of terms, and we will consider whether such terminology lends itself to scientific insights. We focus on a recent article by Wang et al. (2019), which states that:

> A computer program is said to learn from experience *E* with respect to some classes of task *T* and performance measure *P* if its performance can improve with *E* on *T* measured by *P*. (p. 4)

The authors then define *few-shot learning* as a "type of machine learning problem (specified by *E*, *T*, and *P*) where *E* contains only a limited number of examples with supervised information for the target *T*" (p. 4). They further define the special cases of *one-shot learning*, in which *E* contains only one example per category, and *no-shot learning*, in which *E* has no training cases, although other information may be available. In other words, one-shot, few-shot, and no-shot refer to more than learning tasks. They also specify experimental regimens that provide a certain number of training cases for each category and then measure the learned classifier's accuracy on new test cases. Thus, it may be more accurate to use "one-shot, few-shot, and no-shot training," which have fewer connotations about the results of learning.

A drawback of one-shot learning is that, to someone from outside the field, it suggests that a system can learn all it needs to know from one training case for each category, but this will almost never occur.[2] Consider the *nearest neighbor* algorithm (Cover & Hart, 1967), one of the earliest induction techniques that merely stores each training instance and, when asked to classify a new item, retrieves the most similar (or "nearest") according to some metric and predicts the retrieved case's class. Clearly, this method learns something from each training example, but it will only achieve high accuracy on new test instances for the simplest learning tasks. The same observation holds for more sophisticated approaches to induction, like few-shot learning, which hints that learning is complete after a few instances.

A more constructive response is to examine the *rate* at which a learning system improves performance as a function of the number of training cases. This distinguishes clearly between the task (e.g., supervised learning for classification) and how well a particular system does on the task (e.g., exhibits slow or rapid improvement). This idea relates to the notion of *sample complexity* in theoretical analyses of learning (Haussler, 1990). Expectations about the rate of learning should be

[2] Although "one-shot" or "one-trial" learning has a long history in psychology (Guthrie, 1946), "few-shot learning" is a more recent AI coinage of questionable syntax.

conditioned on the difficulty of the performance task. For instance, if each category has multiple, well-separated subclasses, then we will need at least one training case per subclass. Other factors, such as mislabeled samples and irrelevant features, can also affect learning rate. This clarifies that the number of training cases needed to learn an effective classifier is a matter of degree.

However, formal analyses of sample complexity typically predict rates of learning that are orders of magnitude slower than those observed in practice. A more empirical approach, with much older origins in psychology, involves the collection of *learning curves* (Langley, 2000). Rather than measuring performance for training sets of a single size, often selected arbitrarily, learning curves plot performance as a function of the number of training cases. This clarifies that performance typically improves gradually with training and distinguishes clearly between the rate, intercept, and asymptote of learning. Moreover, learning curves let us examine learning rate as a function of task difficulty, which can be influenced by factors like complexity of the target concept, number of irrelevant features, and noise level.[3] Learning curves may also cross when comparing two or more different methods, i.e., when one approach learns more rapidly early on but has a lower asymptote. It is crucial to understand such tradeoffs if we want to apply machine induction in data-sparse settings.

We should also discuss a third term—*zero-shot learning*—that has received considerable attention (Wang et al. 2019). In this paradigm, one provides a system with a description of some new category (e.g., a face) based on its component features (e.g., eyes, nose, and mouth), which have been learned earlier on simpler tasks. The system combines the features in this description and uses the result to classify cases of the new category with no additional training. The adjective *zero-shot* is somewhat misleading, because the system has been given a labeled training case for the class, but one specified in abstract terms. According to this view, the approach is actually a variant on one-shot induction, but it nevertheless examines the important idea of cumulative learning, in which later acquisition builds on earlier experience.

In summary, the recent interest in learning from small numbers of training cases is a positive sign, but the terminology that many researchers have adopted is misleading, and the experimental method associated with it reveals less information than desired. To develop, understand, and evaluate systems that learn from small training sets, we must examine their rate of learning. The older paradigm of reporting learning curves provides details about the rate of performance improvement, and we recommend that the community adopt this approach instead.

[3] Some average-case analyses (e.g., Langley, Iba, & Thompson, 1992; Pazzani & Sarrett, 1992) address these issues, but such work is rare.

4. Early Research on Rapid Learning

The concern with rapid learning is far from new. Much of the early research on machine learning during the 1970s and 1980s addressed the issue. During this period, the great majority of publications focused on implemented systems and their experimental evaluation. The scientific aim was to understand how characteristics of induction algorithms and induction tasks map onto characteristics of learning behavior. Developers typically demonstrated their system's abilities on small training sets—often well under a hundred cases. Early methods often learned very rapidly, but many of them also relied on unrealistic assumptions, such as conjunctive concepts and the absence of noise. Many techniques were also incremental, in that they processed one instance at a time and learned something from each one. Nevertheless, this era saw many intellectual advances and impressive results, which can inform current research in this area.

The 1980s saw widespread adoption of more sophisticated induction methods that handled more complex concepts and dealt with noisy training data, such as decision-tree construction (Quinlan, 1983, 1992), rule induction (Clark & Niblett, 1989), and backpropagation in neural networks. Even so, experiments showed that these techniques could learn accurate models from reasonably few training cases,[4] leading to a variety of applications that were deployed successfully in the commercial sector (Langley & Simon, 1995).

We can divide these early approaches to automated induction into two broad categories. One class, originally associated with the field of pattern recognition, assumed models of fixed structure and focused on estimating their parameters. This paradigm included early work on neural networks.[5] The other class, initially associated with the term machine learning, emphasized determination of model structure as part of the learning task. This paradigm was sometimes referred to as "symbolic learning"; it included rule induction, decision-tree construction, and many other techniques, as well as some parameter estimation methods (generally embedded within the identification of model structures).

Researchers during the 1980s and 1990s devised a variety of responses to the challenge of rapid learning, although not always stated in these terms. These responses all addressed two related issues. The first is to reduce the *variance* of learned models: the degree to which learning on different training samples produces different performance. The second is to reduce *overfitting*: the degree to which a learned model does better on training cases than on test cases. We discuss

[4] For instance, Quinlan (1983) reported that his approach to decision-tree induction required about four cases per terminal node to learn a complex classifier for chess end games.

[5] It also included classic statistical techniques like linear regression, as well as simple probabilistic methods like naive Bayes.

strategies for addressing these issues below. We emphasize methods investigated in the structure-learning paradigm, as many ideas from the parametric paradigm have found their way into the deep learning literature.

A. Constraining Formalisms

One response has employed restricted representations to scope the learning problem. Examples have included decision trees with only one test (Holte, 1993) and constrained Bayesian networks (Friedman, Geiger, & Goldszmidt, 1997). Despite their inability to encode many possible models, such methods have fared much better in practice, achieving higher accuracy and lower variance than many expected, especially on training sets of limited size. Studies of one-layer perceptrons (Shavlik, Mooney, & Towell, 1991) and naive Bayesian classifiers (Langley et al., 1992) have produced similar results within the parametric paradigm. These early results were counterintuitive to many researchers at the time, but they continue to offer valuable insights with direct relevance to induction from small training sets.

B. Guarding Against Overfitting

Another way to reduce variance and speed up learning is to modify more powerful methods to guard against overfitting the training data. In decision-tree and rule induction, this typically involves pruning learned structures to eliminate questionable elements (Quinlan, 1992). Pre-pruning methods involve early halting, before finding a model that overfits the training data, whereas post-pruning instead eliminates elements of a complex model after it has been found. This serves the same role as adding a regularization term in the parameter-estimation paradigm (Bishop, 2007), which can drive many weights to zero.

C. Selecting Features

A different way to reduce model complexity, and thus lower variance, is to select a subset of features to use during induction cases (Blum & Langley, 1997). One common technique, *backward elimination*, starts with all features and eliminates those with low predictive utility. Another approach, *forward selection*, instead starts with no attributes and adds ones that improve predictive ability.[6]

D. Creating Ensembles

A fourth response counters high variance by constructing a number of candidate models from different subsets of the training data and then combining their predictions at performance time. A common technique for this is *bagging* (Breiman, 1996), which takes the majority vote of the

[6] An analogous technique in the parametric paradigm, *principal components analysis*, acts to prioritize and reduce the number of features in a very different, algorithmic, manner.

learned models. Although originally designed for structural learning methods like decision-tree induction, the approach is equally relevant to neural networks and other parametric techniques.

E. Using Background Knowledge

Another response uses background knowledge about a domain, typically stated as rules, to enable learning from relatively few samples. For instance, *analytical* or *explanation-based learning methods* (e.g., DeJong, & Mooney, 1986) construct an explanation for each training case and learn a new rule from each one. A related scheme provides knowledge known as *declarative bias* (e.g., Adé, de Raedt, & Bruynooghe, 1995) to specify a constrained space of candidate hypotheses.

F. Revising Existing Models

A final class of techniques initiates the learning process from an existing model and revises it in response to training data. The initial structure may be handcrafted, as in early work on the revision of logic programs (e.g., Ourston & Mooney, 1994), but it can also be the result of prior learning on a related task, as in work on structural transfer (e.g., Shapiro, Könik, & O'Rorke, 2008). Both approaches make learning easier by reducing the distance between the initial and target model.

Research on each of these strategies focused on improving the predictive accuracy (or reducing the error) of learned classification models. Some efforts, especially on model revision, explicitly addressed the challenge of increasing learning rates and reported learning curves, whereas others did so only implicitly, reporting performance on whatever size data sets were available. By the late 1990s, these positive results were widely interpreted as resulting from a "bias-variance tradeoff." We can define the variance of a learning system as the degree to which the error on new test cases changes in response to changes in the samples (drawn from some distribution) on which it is trained. We can define the bias of a learner as the degree to which it produces minimal error on new test cases when given infinite training data, which relates to its ability to create arbitrary decision boundaries. We refer to a tradeoff because, in most situations, altering the induction method to reduce its bias will increase its variance and vice versa. This empirical relationship is relevant to learning from small training sets because, given unlimited data, the variance approaches zero. Thus, the tradeoff is far stronger when only small samples are available. The success of the techniques above can be attributed to reducing variance in ways that more than offset the cost of increasing bias. For this reason, they each remain relevant to the objective of rapid learning and deserve the attention of today's researchers.

5. Recent Research on Rapid Deep Learning

Researchers in deep learning have begun to examine the problem of acquiring expertise from limited data. The most common setting is classification, utilizing a deep neural net containing thousands to millions of free parameters. In this context, the terms *zero-shot*, *one-shot*, and *few-shot learning* refer to the quantity of supervised training data available to the classifier. These methods often compensate by drawing on a body of related knowledge and expertise. For example, an image, such as a tiger, can be recognized without any training data for that class (zero-shot learning) because tigers are known to be large, striped, and carnivorous—features that are recognizable from imagery (Markowitz et al., 2017). A new image class can be recognized from a single positive example (one-shot learning) using the features extracted from an image classifier for related tasks together with its training data as negative examples (Kozerawski and Turk, 2018). Hand-written characters from 50 alphabets can be recognized given one to five examples of each class (few-shot learning) after pre-training on a large body of related discrimination tasks (Snell et al., 2017). All of these approaches apply prior expertise to simplify the target learning task.[7]

Much of the work in rapid deep learning focuses on the *N*-way *K*-shot classification problem, which we segment into performance and learning tasks:

***N*-way classification**

Given a data sample drawn from one of *N* classes

Determine which class the sample belongs to in *N*

***K*-shot learning**

Given prior expertise encoded in some form

Given K supervised samples of each class

Learn to perform *N*-way classification

The assumption in rapid deep learning is that direct experience with the target classification task is very limited, meaning *K* and *N* are small, and those samples are insufficient to train a deep network alone. As a result, prior expertise plays a critical role in work on rapid deep learning. The survey by Wang et al. (2019) categorizes research in this area by the mechanisms used to exploit prior knowledge and identifies three main approaches:

[7] It is worth noting that the terms *zero-shot*, *one-shot*, and *few-shot learning* are not synonymous with work in deep neural nets. For example, Fei-Fei et al. (2006) build a prior over image features in a source task, and use Bayesian updating to learn a classifier given one to six instances of each target class.

- adapt training data from related tasks to bias the classifier,
- modify the model (the neural net architecture) to make the classifier easy to learn, and
- use knowledge from related tasks to inform parameter search.

A few examples may help to clarify these strategies. Methods of adapting training data are prevalent in deep learning, where the models contain very large numbers of free parameters. Researchers in deep learning frequently stretch training data by translating, flipping, scaling, cropping, and rotating images, by taking rolling windows from sequential data, or by simply adding noise to existing training data (among other transformations). Attribute strengths learned from a large set of scene images can be used to augment training data while preserving their labels (Kwitt et al., 2016). New training data for a target class in a few-shot learning context can be generated by employing feature models acquired from background knowledge (Schwartz et al., 2018). These techniques were much less common in classical machine learning research, which emphasized constrained representations that required limited training data.

Many approaches act by modifying the model to make it easier to learn. For example, prior experience can be used to constrain a deep network by directly sharing model parameters (Zhang et al., 2018). They address few-shot classification by importing model structure from ImageNet to obtain low-level features and then training new layers to adapt that knowledge for fine-grained image classification. The work by Snell et al. (2017) exploits prior knowledge by training a deep network on discrimination tasks drawn from a hand-written character dataset. It identifies class prototypes in the compact representation produced by this network, then matches the representation of a new sample against the prototypes to retrieve a classification vector.

Methods that use knowledge learned on related tasks to inform parameter update are beginning to appear in the deep learning literature. For example, prior experience can be used to initialize parameter values throughout a deep network before optimizing them by standard methods (e.g., gradient descent) (Donahue et al., 2013). The researchers trained a deep convolutional network on a 1000-way object recognition task and used the resulting parameter values to seed a classifier for subcategory and scene recognition tasks, resulting in state-of-the-art accuracy given limited amounts of supervised data. In contrast, Finn et al. (2017) used prior knowledge directly to speed search; they employed experience from a family of *K*-shot learning tasks to compute a direction of change for updating model parameters on a different *K*-shot learning task. They showed rapid learning from as little as one gradient update step in regression, classification and reinforcement learning tasks.

As a whole, research in rapid deep learning faces the issue that small data sets invite overfitting in models with thousands to millions of parameters. The methods of exploiting prior experience compensate for that tendency and simultaneously cast rapid deep learning as a form of knowledge transfer. Although many papers focus on learning from a few samples in the target task, the quantity of prior experience they employ is not always clear. Papers typically report classification accuracy obtained after training on zero, one, or a few samples of the target class, whereas a small number provide learning curves that measure accuracy on a test set after each sample is presented to the learning system. This emphasis on prediction accuracy is most likely inherited from research on deep learning where large amounts of training are available or assumed. In Chapter 7, we argue that the use of learning curves will influence research on rapid deep learning in productive directions.

6. Recurring Themes

We should also consider the metaphor of induction as search and its implications for rapid learning, especially as it plays a central role in both the structural and parametric learning paradigms. The idea is far from new, dating back at least to Simon and Lea (1974), who viewed induction as search through two related spaces of hypotheses and instances. Mitchell (1982) popularized the notion in the context of supervised learning of conjunctive rules, and by the mid-1980s, it had become widely adopted by researchers in the machine learning community.

To characterize any problem in terms of search, we must specify some notation for representing candidates, along with an initial candidate, operators for generating new candidates from existing ones, criteria to select among operators or candidates, and a termination criterion. For learning, this means specifying a formalism for models, an initial model or hypothesis, operators for generating or updating models, criteria for selecting among them, and conditions on when to halt.

We can describe many classic induction tasks in these terms, although many systems have relied on quite simple search methods. Most techniques for searching a space of model structures, such as those for inducing decision trees (Quinlan, 1992) and Bayesian networks (Heckerman, Geiger, & Chickering, 1995), rely on "greedy" search, which iteratively selects the best-scoring successor candidate until termination. The search metaphor has been equally important in work on statistical induction, including neural networks, but these approaches typically explore a space of parametric values for structures that are fixed in advance. Rather than generating different structural candidates and using heuristics to select the best alternative, they rely on schemes like gradient descent, which iteratively computes the next parametric candidate from the current one.

The model spaces that arise in machine induction are generally very large, which means they include many candidate solutions. However, the difficulty lies not in finding models that score well on some evaluation criterion (e.g., error on the training set), but in finding ones that will also generalize well to novel test cases. Larger search spaces typically hold more candidates that do well on the training data but poorly on test data. This is the underlying cause of overfitting and high variance, which slow the rate of learning and require more training cases.

The implications are intuitive and straightforward. We can mitigate overfitting and improve learning rate by modulating the search process. One approach is to reduce the size of the space by adopting constrained representations, selecting features, or using background knowledge. Another is to decrease the depth of search, which we can achieve by starting closer to the target model, as demonstrated in work on model revision. A third option is to improve the evaluation function that guides search, for example, by finding a better criterion for deciding how to extend a decision tree or a better update function for neural networks. A fourth alternative is to improve the termination criterion, as in pruning during the induction of decision trees, which attempts to reduce overfitting by eliminating questionable branches.[8]

Although deep learning focuses on parameter estimation rather than inducing model structures, these approaches to modulating search are equally relevant to reducing reliance on large training sets. Wang et al. (2019) recently explored this idea in very similar terms. They focused on "few-shot" scenarios in which the number of training cases is held constant, but the intent to increase learning rate (task performance as a function of training sample size) is the same. They discuss constraining the hypothesis space to reduce the search required to find a good model (e.g., by adding structure to the network) and guiding the search more effectively (e.g., by improving meta-parameters). Their terminology differs from the literature on structural learning, but they propose similar ideas adapted in ways appropriate to deep neural networks.

This suggests that we should reexamine the strategies from Chapter 4 in terms of the search metaphor and attempt to identify direct analogs between the two paradigms. The first column of Table 6-1 lists the six strategies for reducing variance, the second column presents examples from the structural learning paradigm, and the third column gives examples from the deep learning movement. There has been considerable diversity within the structural paradigm because different groups have focused on different structural formalisms (e.g., decision trees, rule sets, Bayesian networks). The deep learning community instead designs a neural network for a given task and searches for parameter values, but with millions of such parameters, this can have a similar effect to structural search. As a result, insights about the latter remain relevant to deep learning.

Table 6-1. Themes and Approaches to Modulating Search across the Structural Learning and Deep Learning Traditions

Theme	Structural Learning	Rapid Deep Learning
Constraining formalisms	Decision trees with one test	Custom model architectures and standard component layers (e.g., convolutional, recurrent)
Guarding against overfitting	Pre-pruning model structures Post-pruning model structures	Dropout layers Imposing regularization terms Detecting overfitting during training
Selecting features	Backward elimination of features Forward selection of features	Embedded representations Distillation (reducing model size)

[8] Introducing more sophisticated search methods does not help on its own, as this only increases the chances of finding a model that does well on training data and poorly on test data. There is evidence that more extensive search can increase variance and reduce generalization to new cases.

Theme	Structural Learning	Rapid Deep Learning
Creating ensembles	Combining multiple decision trees or rule sets	Combining multiple neural networks
Using background knowledge	Analytical learning methods Constructive induction Declarative bias to bound space	Using features from pre-trained networks Training on related tasks Informing parameter update
Revising existing models	Handcrafting initial model Structural transfer from other tasks	Initializing parameters Importing model substructure

Constrained formalisms restrict the degrees of freedom within a learning system and reduce variance in the resulting performance as the training data changes. Where work in structural learning employs restricted representations to accomplish this goal, research in deep learning shapes information flow across model layers and time. As an illustration, an autoencoder is a deep neural net trained to predict its input after forcing the signal through a network layer of reduced size (called an *embedded representation*). This bottleneck plays the role of a constrained formalism in structural machine learning; it captures the key information and causes the autoencoder to have less variance than a fully connected design for the same task. Variations on this theme are common in rapid deep learning. For example, Vinyals et al. (2016) compared embedded representations of images to perform *N*-way classification and generate the embedding by considering training cases in serial, thus ensuring that key features were remembered over time. More broadly, researchers in deep learning shape information flow (the analog of imposing a constrained formalism) by building networks from known structural elements (such as convolutional and fully connected layers) and primitive elements with special properties (such as recurrent neural nets and long short term memory layers that incorporate feedback and memory).

Guarding against overfitting prevents a learning system from capturing unnecessary detail in the training data or, equivalently, forces it to generalize from the data at hand. In the early rapid learning literature, the key adaptation is to prune learned structures (e.g., by penalizing conjunctive concept length). In the deep learning (and rapid deep learning) literature, the analogues are to introduce regularization terms into the optimization function that enforce a level of generality and to detect/use the onset of overfitting to terminate the training process. In addition, *dropout layers* (where a portion of the neurons turn off on each parameter update cycle) create multiple network variants whose output is combined by subsequent layers. This reduces variance by making predictions rely on different training distributions.

Feature selection facilitates hypothesis search by reducing model size. Structural learning techniques select features that are often hand-engineered or that explicitly transform representations to reduce dimensionality (e.g., by employing principal components analysis). In contrast, deep learning systems infer higher-level features from raw inputs as part of learning to perform the target task. At minimum, the learned features summarize inputs. For example, a convolutional network combines windows of input data (typically in a cascade across layers). Learned representations can also be *embedded* (i.e., forced to lower dimensionality), as in the case of an autoencoder described above. The work by Snell et al. (2017) is an example; it relies heavily on embedded representations for feature reduction to perform character recognition.

Ensemble methods combine the predictions of multiple learned classifiers, which reduces variance by partitioning the available training data. This technique is common in the structural learning literature, although it is infrequently employed in rapid deep learning. Wasay et al. (2020) argued that the training cost of multiple deep networks is the barrier and responded with a method of sharing those costs by generating multiple classifiers from a small number of more general parents.

Using background knowledge facilitates search by restricting the space of possible hypotheses. The effect is similar to using constrained formalisms, although the emphasis is on focusing the search within an existing model structure. As mentioned above, research in structural learning frequently specified a constrained space of candidate hypotheses via logical rules. In the context of rapid deep learning, background knowledge takes the form of features inferred by other learners or data from related tasks that a learning system transfers into its parameter set through training. For example, the work by Zhang et al. (2018) incorporates background knowledge by importing model structure.

The approach of *revising existing models* takes on somewhat different forms. In non-parametric learning, model revision acts to refine a prior structure that has been hand-engineered or transferred from a related task. In rapid deep learning, the model architecture is task-specific but fixed, so model revision has the character of parameter tuning. As mentioned in Chapter 5, Donahue et al. (2013) imported model layers from a related task (incorporating background knowledge) and then initialized the free variables from prior knowledge to speed parameter search (revising the existing model). Work by Finn et al. (2017) is something of an outlier here, as it employs background knowledge to directly inform gradient update steps through a deep network. In contrast, most of the work in both structural learning and rapid deep learning drives hypothesis search from objective functions but makes no special mention of search control.

In summary, the perspective of learning, as search developed in the structural learning literature from ~1980 to the present, supplies distinctions that span work on rapid deep learning. The lessons from classical work on learning as search are also relevant to work on rapid deep learning going forward.

7. Lessons Learned

Despite the increased interest that machine learning has received recently, it is a mature discipline with decades of prior accomplishments. Thus, it seems reasonable to extract lessons that can inform and guide research on rapid learning in deep neural networks. Here, we identify a number of such insights, which we state in generic terms as they apply equally to all induction paradigms.

A. Separate Learning from Performance

Many ideas about machine induction come from psychology, which has long distinguished between performance on a task and learning, which it defines as change in this performance. Early research in machine learning retained this separation, but it has become less common in recent years. This distinction makes it clear that we can combine a given performance mechanism (e.g., decision-tree classification) with different induction techniques (e.g., inducing a decision tree with or without pruning). Moreover, we can combine a particular learning process (e.g., storing cases) with different performance methods (e.g., 1 nearest neighbor vs. 5 nearest neighbors). We must examine both elements to draw conclusions about the sources of system power.

B. Record and Analyze Learning Curves

Measuring performance after *N* training cases, which is the norm in the deep learning literature, provides very limited information regardless of whether *N* is small or large. In contrast, collecting learning curves reveals the rates at which performance improves as a function of the amount of training data and reveals the intercepts and asymptotes of learning. These observations offer both theoretical insights about when rapid learning is possible and practical support for rapid learning in applied settings.

C. Take Advantage of Classic Insights

Four decades of research have revealed many ways to make induction more effective and increase the rate of learning. The community should take advantage of constrained formalisms, guards against overfitting, feature reduction techniques, ensemble methods, background knowledge, and revision of existing models to make learning effective when only small data sets are available. All are known in the deep learning community, but they are especially important as aids for induction from small training sets.

D. Understand Sources of Power

Empirical studies of machine learning have the potential to elucidate underlying causes of behavior, but all too often they only show incremental improvement without offering clear explanations. Treating data repositories as "benchmarks" encourages a competitive mindset that produces uninformative "bake offs." In contrast, controlled experiments can reveal the reasons for observed differences, not just their existence, and suggest general design principles that apply in many different situations.

E. Realize Not All Learning Tasks Are Created Equal

Some problems are inherently more difficult than others and require more training data. Experiments with synthetic domains, especially when combined with learning curves, can systematically vary factors such as target complexity and noise level that can affect task difficulty. Such careful studies can help explain variations in results observed across different natural data sets, including the reasons that some tasks support faster rates of learning or higher asymptotic behavior than observed on other problems.

F. Recognize and Utilize Tradeoffs

No approach or paradigm for machine learning will always produce the best results—they all have their strengths and weaknesses. This makes it essential that we understand the conditions under which an induction method works well or poorly. Some approaches may handle noise better but require more training data, whereas others may learn more rapidly but give lower asymptotic performance. Scientific understanding must precede engineering confidence about how different learning methods will fare in real-world settings and thus guide the selection of a technique that matches the task at hand.

These lessons have proved useful during the extended history of machine induction. They are not limited to either neural networks or rapid learning, but they are highly relevant to both of these contexts.

8. Recommendations

In the preceding sections, we examined the challenge of rapid learning, the terminology used to describe it, prior work in two major paradigms, themes that cut across those paradigms, and some lessons from both lines of research. In closing, we should consider ways in which the machine learning community, including researchers and government sponsors, should alter its future trajectory. We begin with recommendations for researchers, then consider ways the U.S. Government could encourage machine learning researchers and developers to pursue these additional activities. In each case, we provide an action item and explain the reasons behind it.

A. Devote More Attention to Domains That Require Rapid Learning

Not all applications come with large training sets available and, in many cases, they would be prohibitively expensive or even impossible to collect. Rare diseases, almost by definition, produce few samples for training, and responses to novel contagions such as COVID-19 would benefit from early analyses of limited data before the contagions spread widely. Adversaries' weapon systems and computer viruses typically remain secret until deployed or released, making it highly desirable to create models from as few instances as possible. Improved methods for rapid induction will extend the reach of machine learning to such high-risk settings.

B. Adopt Classic Lessons about Empirical Study of Learning

As noted earlier, machine learning has been a mature discipline for decades and has developed many experimental tools for understanding the behavior of induction systems. Learning curves, although not currently popular, are especially relevant for the situations with small samples, as they reveal performance as a function of training set size. Such plots can vary with problem characteristics such as noise level, can exhibit tradeoffs among different abilities, and can indicate that no approach is uniformly better. Robust engineering of systems for rapid learning depends on understanding strengths, weaknesses, and sources of power.

C. Incorporate Insights from Structural Induction into Deep Learning

Recent excitement about one paradigm for learning in deep neural networks (gradient descent through a parameter space) does not detract from the many successes of a different paradigm (structural learning, which instead searches a space of distinct model structures). The parametric and structural frameworks draw upon quite different metaphors, but each has lessons to offer about how to increase the rate of learning. Researchers in both communities would benefit from mastering the others' literature and attempting to incorporate these insights into their own approach to machine induction.

D. Increase the Diversity of Induction Methods under Study

If we believe that no approach to machine learning is always superior, then the community would benefit from exploring a broader range of techniques. Methods for inducing decision trees, logical rules, Bayesian networks, and case libraries introduce different biases than deep neural networks, many of them relevant to learning from small training sets. Experimental comparisons of alternative mechanisms will reveal when each produces better results and may point the way to hybrid frameworks that combine the best of different paradigms.

E. Collect and Popularize Challenge Problems for Rapid Learning

Government agencies should identify challenging and impactful problems that require rapid learning and fund research on them. Examples might be safe autonomous vehicle navigation in rare near-accident scenarios and discovery of effective treatments for rare diseases. In both cases, training data are difficult to obtain, but benefits of successful rapid learning would be high. Prizes are another option, as rapid learning tasks support well-defined achievement goals.

F. Encourage Careful Scientific Studies of Rapid Learning

Although competitions can galvanize interest in a topic, they can also distract a community from efforts to understand the reasons for any mechanism's success or failure, including relations between features of the task, characteristics of the learning method, and overall system behavior. Government agencies can discourage "bake offs" and reward researchers for carrying out balanced studies that reveal limitations and tradeoffs. Work on adversarial learning has been a step in the right direction, but it should be combined with careful studies that examine factors influencing its effects.

G. Organize Meetings on Bridging Structural and Deep Learning

Many deep-learning researchers are unfamiliar with methods for structural induction and many structural induction researchers are unfamiliar with deep-learning techniques. Given that both paradigms have insights that would benefit the other, Government agencies should support meetings that bring together the two communities and expose them to each other's concepts, terminology, techniques, and experimental methods. This will help bridge the intellectual gap and encourage new work that crosses paradigm boundaries.

H. Fund Tutorials and Summer Schools to Foster Intellectual Diversity

Although cross-paradigm meetings are an important step, they will not be sufficient on their own. U.S. agencies should complement them by funding other activities, such as tutorials and summer schools that aim explicitly to increase intellectual diversity within the machine learning community. Participants should come away with a clear view of the field's long and successful history, the great variety of techniques it has developed, and the benefits of having a large repertoire of methods from which to draw.

These activities could be encompassed by new funding programs that are aimed specifically at addressing challenges that arise in learning from small amounts of training data. Taken together, these and similar efforts should strengthen and broaden the machine learning community, improving its ability to develop robust methods for rapid learning, understand when these methods are effective, and apply them to problems of national interest.

Appendix A: Recent Papers in Rapid Learning

Although this paper has primarily examined broad strategies for rapid learning that can be shared across the structural and parametric learning traditions, this section illustrates how those strategies are realized in a selection of recent technical papers. We have organized this discussion into two symmetric subsections describing recent work in each tradition.

1. Recent Papers in Rapid Parametric Learning

A good deal of current research in rapid parametric learning is motivated by recent DARPA programs.[9] Table A-1 identifies a selection of papers that illustrate themes presented in Table 6-1, which we discuss below.

Table A-1. Recent Papers in Parametric Learning that Illustrate Rapid Learning Themes

Theme	Description	References
Constraining formalisms	Embed parametric methods in constrained formalisms to enable rapid learning	Pfeffer, 2017; Wang & Yeung 2016
Selecting features	Exploit embedded representations to speed learning	Brown et al., 2020; Bommasani, Katiyar, & Cardie, 2019
Using background knowledge	Apply ontological knowledge to speed learning Select transfer tasks that facilitate rapid learning	Zhou, Khashabi, Tsai, & Roth, 2018; Yin, Hay, & Roth, 2019; Nayak & Bach, 2020; Su, Maji, & Hariharan, 2020
Revising existing models	Augment training data to refine learned models (and speed learning)	Beck, Papakipos, & Littman, 2019; Wang, Girshick, Hebert, & Hariharan, 2018

a. Constraining Formalisms

The first strategy embeds parametric learning in constrained formalisms to achieve rapid learning.

Pfeffer (2017) learns from data in the context of a hand-coded program expressed in a probabilistic programming language. Statements in this language define a probability density function over values produced by the program, and learning is motivated by an error function that measures how well this density function models a training set.

[9] Examples include Science of Artificial Intelligence and Learning for Open-world Novelty (SAIL-ON), Lifelong Learning Machines (L2M), Explainable AI (XAI), and Machine Common Sense (MCS).

Although this work is in its early stages, it suggests a method of merging classical programming with statistical learning that speeds learning by constraining tasks.

Work on Bayesian deep learning (Wang and Yeung, 2016) mates deep learning models for perception tasks to probabilistic graphical models for causal inference and dealing with uncertainty. Many variants have been proposed for recommendation, topic modeling, and control tasks, along with multiple learning algorithms for acquiring free parameters (e.g., expectation management, Bayesian conditional density filtering, and stochastic gradient variational Bayes). Overall, these approaches support rapid learning by constraining the learning tasks, enabling model acquisition from smaller datasets.

b. Selecting Features

Several recent papers in natural language processing rely on embedded representations to speed learning. This is a form of feature selection, as discussed in Chapter 6.

Brown et al. (2020) noted that language models learned from very large bodies of text implicitly acquire many distinctions that make them applicable to new linguistic tasks. They showed that GPT-3 (with 175 billion free parameters that are trained autoregressively on a language prediction task) achieves strong performance as a few-shot learner on new datasets and tasks including translation, question answering, unscrambling words, using a novel word in a sentence, and performing three-digit arithmetic.

Bommasani et al. (2019) developed techniques for encoding relational data and making it useful in sentiment analysis tasks; given a text snippet, is a named entity positive, negative, or neutral towards a target relation or event? Their approach acquires embedded representations of entity mentions, target mentions, and context from text, and passes those encodings to a simple fully connected network for sentiment classification. This work contributes to rapid deep learning in the same sense as Brown et al. [3]; it enables language learners to develop rich representations that are immediately applicable to new domains.

c. Using Background Knowledge

A number of recent papers illustrate the theme of using background knowledge to speed learning.

Zhou et al. (2018) employed ontological knowledge to address the zero-shot entity typing problem; given a reference in a sentence and a taxonomy of entity types, identify a set of types appropriate to the mention. Their approach links context words to type-compatible Wikipedia pages and exploits human-curated information in the Wikipedia pages to infer the entity type.

Yin et al. (2019) addressed zero-shot text labeling tasks. Their approach trains the BERT language model (Devlin et al., 2019) on entailment examples and exploits the learned ontological knowledge to perform new entailment tasks (e.g., to identify the topics, emotions, or situations implied by a text snippet).

Nayak and Bach (2020) exploited common-sense knowledge embedded in hand-constructed graphical structures like ConceptNet[10] to perform classification tasks. Their approach learns to translate the structure of a knowledge graph into a neural net and then employs the neural net for several classification tasks. The resulting system shows strong performance on zero-shot intent classification, fine-grained entity typing, and object recognition tasks.

Su et al. (2020) examined the best pairing of source and target tasks when exploiting background knowledge. They considered a scenario composed of pretraining on unlabeled image rotation tasks (e.g., assembling a jigsaw puzzle composed of birds and animals), followed by few-shot learning on a target classification task using images from the same domain. The work shows that (a) with no additional training data, adding an auxiliary self-supervised task improves the performance of existing few-shot techniques, and (b) the benefits of self-supervision increase with the difficulty of the target task (e.g., classifying increasingly noisy images). It is not entirely clear why the relation in (b) exists, but it provides guidance for phrasing a variety of rapid deep learning tasks.

d. Revising Existing Models

Several recent papers illustrate the theme of achieving rapid learning by revising existing models.

Beck et al. (2019) refined learned models for automated driving tasks by training on additional data that would otherwise be discarded. Their method associates numeric feedback with human piloted examples (e.g., negative feedback for swerves, positive for pristine lane changes) and acquires deterministic continuous control programs from any range of good and bad behavior demonstrations.

Wang et al. (2018) speeded learning in few-shot classification tasks by (in the author's words) "hallucinating" new training data that is specifically useful for learning. Their approach couples an image generator (that maps real samples into new variants) to an image classifier and achieves targeted hallucination by passing the classification loss into the generator during training.

In summary, many recent research efforts in rapid parametric learning can be appreciated as an exploration of historical themes discussed in this document that unify the traditions of statistical and parametric learning.

[10] https://conceptnet.io/

2. Recent Papers in Rapid Structural Learning

Despite the attention that deep neural networks have received, there has continued to be active work on structural learning, much of it focused on effective induction from small training sets. Table A-2 identifies a selection of recent papers that illustrate themes presented in Table 6-1, which we discuss below.

Table A-2. Recent Papers in Structural Learning that Illustrate Rapid Learning Themes

Theme	Description	References
Constraining formalisms	Use constrained structural notations to limit the space of hypotheses	McFate & Forbus, 2016; Schede, Kolb, & Teso, 2019
Using background knowledge	Provide structural building blocks to generate candidate hypotheses	Muggleton, Dai, Sammut, Tamaddoni-Nezhad, Wen, & Zhou, 2018; Mitra & Baral, 2016
Revising existing models	Identify which model elements lead to errors and repair them	Klenk, Piotrowski, Stern, Mohan, & de Kleer, 2020; Arvay & Langley, 2016

a. Constraining Formalisms

One common strategy used to support rapid learning in the structural paradigm is to adopt a constrained formalism for representing models and hypotheses. Here are two recent examples of this approach.

McFate and Forbus (2016) reported a system that encodes categories as conjunctions of features in predicate logic, each with an associated conditional probability. An incremental learning process updates these probabilities, adds new relations, and introduces new categories when training cases are sufficiently dissimilar. The system learned the accurate semantics for denominal verbs like *giving* from only nine sentences per word and generated correct inferences for new instances that occurred in very different constructions.

Schede et al. (2019) described an approach to inducing constraint programs that are stated as a collection of linear inequalities over a set of continuous variables. Their system learns accurate and interpretable constraints from substantially fewer training cases than its predecessors, needing only 200 samples on standard test problems.

b. Using Background Knowledge

Another important approach to increasing learning rates within the structural paradigm is to provide generic background knowledge that informs the induction process. Here are two instances of this technique.

Muggleton et al. (2018) reported a novel method for computer vision that uses meta-interpretive learning to invent useful predicates and induce logical models from small training sets. Their system uses background knowledge about shadows and reflection, stated in relational logic, to speed the acquisition of visual object categories on both synthetic and natural images—exhibiting one-shot learning in some cases.

Mitra and Baral (2016) presented an approach to language processing that learns logical rules for use in question answering, reference resolution, and inference. They provided two very general axioms in the event calculus that let their system use learned rules more effectively. The authors did not report results on small data sets, but their method fared substantially better than neural networks, which suggested more rapid learning.

c. Revising Existing Models

Another line of research starts with a structural model that is consistent with past observations but revises the model's content when it no longer fits the data. Two recent efforts illustrate this approach.

Klenk et al. (2020) reported an approach to learning in environments that change over time. Their HYDRA system starts with an accurate model of the physical world stated in PDDL+, a planning formalism that includes discrete and continuous elements. When anomalies arise, it generates structural and quantitative hypotheses, runs a small number of experiments to evaluate them, and selects the candidate with the most support to revise its model. Their system produced encouraging results in the Science Birds domain.

Arvay and Langley (2016) described a system that encodes models as sets of processes, each with an algebraic rate expression and one or more derivatives that are proportional to it. When given multivariate time series that diverge from a model's predictions, it identifies which processes are responsible, then either revises their parameters or replaces them with other processes that better fit the observations. The system needs only tens of data points to detect anomalies and guide the search for process models.

Other recent work in the structural learning framework has continued to explore themes that support rapid learning. For example, Painsky and Rosset (2017) examined induction of random forests (a widely used ensemble method based on decision trees), which guards against overfitting. Although this research was not motivated by the desire to improve rapid learning, it should have that effect as it acts to decrease the variance on predictions learned from small quantities of data.

3. Summary

Recent papers in both the parametric and structural paradigms have advanced the technology for learning effective models from small training sets. Despite their differences, they all draw on the long history of insights on this topic. Some recent work integrates ideas from both paradigms. For example, Ross et al. (2018) presented a system that uses the output of a symbolic parser to determine the structure of a deep neural network that in turn learns grounded representations from captioned videos. Hybrid approaches of this form hold the promise to combine the benefits of structural and parametric methods for rapid learning.

References

Adé, H., de Raedt, L., & Bruynooghe, M. (1995). Declarative bias for specific-to-general ILP systems. *Machine Learning*, *20*, 119–154.

Arvay A., & Langley, P. W. (2016). “Heuristic Adaptation of Scientific Process Models,” *Adv. Cogn. Syst.*, vol. 4, pp. 207–226.

Beck, J., Papakipos, Z., & Littman, M. (January 2019). “ReNeg and Backseat Driver: Learning from Demonstration with Continuous Human Feedback,” *ArXiv190105101 Cs Stat*, Accessed: Sep. 14, 2020. [Online]. Available: http://arxiv.org/abs/1901.05101.

Bishop, C. (2007). *Pattern recognition and machine learning*. Berlin: Springer. ISBN-10: 0387310738, ISBN-13: 978-0387310732

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*, 245–271.

Bommasani, R., Katiyar, A., & Cardie, C. (June 2019). “SPARSE: Structured Prediction using Argument-Relative Structured Encoding,” in *Proceedings of the Third Workshop on Structured Prediction for NLP*, Minneapolis, Minnesota, pp. 13–17, doi: 10.18653/v1/W19-1503.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Brown, T.B. et al. (July 2020) “Language Models are Few-Shot Learners,” *ArXiv200514165 Cs*, Accessed: Feb. 20, 2021. [Online]. Available: http://arxiv.org/abs/2005.14165.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Clark, P., & Niblett, T (1989). The CN2 induction algorithm. *Machine Learning*, *34*, 261–283.

DeJong, G., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning*, *1*, 145–176.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (May 2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *ArXiv181004805 Cs*, Accessed: Feb. 20, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*, 594–611. DOI:https://doi.org/10.1109/TPAMI.2006.79

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta learning for fast adaptation of deep networks. *Proceedings of the Thirty-Fourth International Conference on Machine Learning* (pp. 1126–1135). Sydney, NSW.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131–163.

Guthrie, E. R. (1946). Psychological facts and psychological theory. *Psychological Bulletin*, *43*, 1–20.

Haussler, D. (1990). Probably approximately correct learning. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 1101–1108). Boston, MA: AAAI Press.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197–243.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 63–90.

Ju, C., Bibaut, A., & Laan, M. (2017). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, *45*. 10.1080/02664763.2018.1441383.

Klenk, M., Piotrowski, W., Stern, R., Mohan, S. & de Kleer, J. (2020). "Model-Based Novelty Adaptation for Open-World AI," *Proc. Eighth Annu. Conf. Adv. Cogn. Syst.*, p. 4.

Kozerawski, J., & Turk, M. (2018). CLEAR: Cumulative LEARning for one-shot one-class image recognition. *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 3446–3455. Salt Lake City, UT: IEEE.

Kwitt, R. Hegenbart, S., & Niethammer, M. (2016). One-shot learning of scene locations via feature trajectory transfer. *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 78–86. Las Vegas, NV: IEEE.

Langley, P. (2000). Crafting papers on machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 1207–1211). Stanford, CA: Morgan Kaufmann.

Langley, P. (2011). The changing science of machine learning. *Machine Learning*, *82*, 275–279.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 223–228). San Jose, CA: AAAI Press.

Langley, P., & Simon, H. A. (1995). Applications of machine learning and rule induction. *Communications of the ACM*, *38*, 55–64.

Markowitz, J., Schmidt, A., Burlina, P., & Wang, I-Jeng. (2017). *Combining deep universal features, semantic attributes, and hierarchical classification for zero-shot learning*. arXiv:1712.03151v1 [cs.CV].

McFate C., & Forbus, K. D. (2016). "Analogical Generalization and Retrieval for Denominal Verb Interpretation," *Proc. Thirty-Eighth Annu. Meet. Cogn. Sci. Soc.*

Mitra A., & Baral C. (February 2016). "Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, pp. 2779–2785, Accessed: Apr. 26, 2021. [Online].

Muggleton, S., Dai, W.-Z., Sammut, C., Tamaddoni-Nezhad, A., Wen, J., & Zhou, Z.-H. (2018). "Meta-Interpretive Learning from noisy images," *Mach. Lang.*, vol. 107, no. 7, pp. 1097–1118, doi: 10.1007/s10994-018-5710-8.

Nayak, N. V., & Bach, S. H. (June 2020). “Zero-Shot Learning with Common Sense Knowledge Graphs,” *ArXiv200610713 Cs Stat*, Accessed: Sep. 14, 2020. [Online]. Available: http://arxiv.org/abs/2006.10713.

Ourston, D., & Mooney, R. J. (1994). Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, *66*, 273-309.

Painsky A., & Rosset, S. (2017). “Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2142–2153, doi: 10.1109/TPAMI.2016.2636831.

Pazzani, M., & Sarrett, W. (1992). A framework for average-case analysis of conjunctive learning algorithms. *Machine Learning*, *9*, 349–372.

Pfeffer, A. (May 2017). “Learning Probabilistic Programs Using Backpropagation,” *ArXiv170505396 Cs Stat*, Accessed: Feb. 13, 2021. [Online]. Available: http://arxiv.org/abs/1705.05396.

Quinlan, J. R. (1992). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Ross, C., Barbu, A., Berzak, Y., Myanganbayar, B., & Katz, B. (2018). “Grounding language acquisition by training semantic parsers using captioned videos,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 2647–2656, doi: 10.18653/v1/D18-1285.

Schede, E. A., Kolb, S., & Teso, S. (November 2019). “Learning Linear Programs from Data,” in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, Portland, OR, USA, pp. 1019–1026, doi: 10.1109/ICTAI.2019.00143.

Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., & Bronstein, A. (2018). Delta-encoder: An effective sample synthesis method for few-shot object recognition. *Advances in Neural Information Processing Systems*, *31*, 2850–2860.

Shapiro, D., Könik, K., & O’Rorke, P. (2008). Achieving far transfer in an integrated cognitive architecture. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* (pp. 1325–1330). Chicago, IL: AAAI Press.

Shavlik, J. W., Mooney, R. J. & Towell, G. G. (1991). Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning*, *6*, 111–143.

Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and cognition*. Hillsdale, NJ: Lawrence Erlbaum.

Simons, T. & Lee, D. (2019). A review of binarized neural networks. *Electronics*, *8*, 661.

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *31st Conference on Neural Information Processing Systems* (NIPS 2017), Long Beach, CA.

Su, J.-C., Maji, S., & Hariharan, B. “When Does Self-supervision Improve Few-shot Learning?”, *ArXiv191003560 Cs*, Jul. 2020, Accessed: Sep. 14, 2020. [Online]. Available: http://arxiv.org/abs/1910.03560.

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, *29*, 3630–3638.

Wang, H., & Yeung, D.-Y. (2016). “Towards Bayesian Deep Learning: A Survey.”

Wang, Y., Yao, Q., Kwok, J., & Ni, L., (2019) *Generalizing from a few examples: A survey on few-shot learning*. https://arxiv.org/abs/1904.05046

Wang, Y.-X., Girshick, R., Hebert, M., & Hariharan, B. (April 2018). “Low-Shot Learning from Imaginary Data,” *ArXiv180105401 Cs*, Accessed: Sep. 14, 2020. [Online]. Available: http://arxiv.org/abs/1801.05401.

Wasay, A., Hentschel, B., Liao, Y., Chen, S., & Idreos, S. (2020). Mothernets: Rapid deep ensemble learning. *Proceedings of the Third Conference on Machine Learning and Systems*. Austin, TX.

Yin, W., Hay, J., & Roth, D. “Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3912–3921, doi: 10.18653/v1/D19-1404.

Zhou, B., Khashabi, D., Tsai, C.-T., & Roth, D. “Zero-Shot Open Entity Typing as Type-Compatible Grounding,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2065–2076, doi: 10.18653/v1/D18-1231.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)
00-04-21

2. REPORT TYPE
Final

3. DATES COVERED (From – To)

4. TITLE AND SUBTITLE
Rapid Learning in Machines: Challenges and Responses

5a. CONTRACT NUMBER
HQ0034-14-D-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBERS

6. AUTHOR(S)
Patrick W. Langley, Daniel G. Shapiro

5d. PROJECT NUMBER
AI-5-4458.02

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES
Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NUMBER
D-14333

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Dai Hyun Kim, Director for Autonomy & Artificial Intelligence
C5ISREW
DDR&E(R&T)/RT&L
Office of the Undersecretary of Defense, Research and Engineering
Pentagon 3C168, Washington, DC, 20301

10. SPONSOR'S / MONITOR'S ACRONYM
OUSD(R&E)

11. SPONSOR'S / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Project Leader: Brian A. Haugh

14. ABSTRACT

Recent breakthroughs have generated an explosion of interest in machine learning, especially with deep neural networks. However, this class of techniques typically relies on very large training sets, which are not available for all application areas. In response, researchers are investigating new methods that are less dependent on massive data repositories. This paper reviews the 40-year history of research on machine learning, which developed a variety of techniques for induction from limited amounts of data and draws lessons for the generic task of rapid learning from small and moderately sized sample sets. We compare two paradigms for inductive learning: search through a space of model structures (which characterizes much of the early work) and search through a parameter space (which includes deep learning methods). We review techniques that have increased the rate of learning from limited data in both paradigms, highlight common themes, and propose adoption of technical and methodological insights obtained from the prior tradition. We close with recommendations for the machine learning community and government agencies funding its work that should extend the reach of this technology to application areas where training data are difficult to obtain.

15. SUBJECT TERMS
machine learning, rapid learning, rapid deep learning zero-shot learning, one-shot learning, few-shot learning

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
Unclassified	Unclassified	Unclassified	Unlimited	28	Dr. Dai Hyun Kim, Director for Autonomy & Artificial Intelligence

19b. TELEPHONE NUMBER (Include Area Code)
571-372-6714