

OVERVIEW

An overview on the evolution and adoption of deep learning applications used in the industry

Sourav Dutta

IBM, India

Correspondence

Sourav Dutta, IBM, Kolkata 700091, India.

Email: s.dutta@ieee.org

With continuous improvements in performance of microprocessors over the years, they now possess capabilities of supercomputers of earlier decade. Further with the continuous increase in the packaging density on the silicon and General Purpose Graphics Processing Unit (GPGPU) enhancements, has led to utilization the deep learning (DL) techniques, which had lost steam during the last decade. A GPGPU is a parallel programming setup using a combination of GPUs and CPUs that can manipulate large matrices. Interestingly, GPUs were created for faster graphic processing, but found its way into relevant scientific computing. DL is a subset of the artificial intelligence (AI) domain and falls specifically under the set of machine learning (ML) techniques which are based on learning data representations rather than task-specific algorithms. It has been observed that the accuracy and the pragmatism of deploying DL at massive level was restricted by technological issues of executing DL based AI models, with extremely large training sessions running into weeks. DL applications can solve problems of very large order and areas like computer vision/image processing is one of the early successes and becoming quite a sensation in many areas such as natural language processing (NLP) with state of the art real-time translation capabilities, automatic game playing, optical character recognition especially handwritten text, and so on. This overview traverses the evolution and successful adoption in the various industry verticals.

This article is categorized under:

Application Areas > Industry Specific Applications

Application Areas > Business and Industry

Technologies > Machine Learning

KEYWORDS

artificial neural networks, CNN, deep learning, image recognition, LSTM, machine learning, text to speech

1 | INTRODUCTION

Artificial intelligence (AI) gained considerable momentum in the 1950s with pioneers from MIT, IBM, and CMU among others collaborating in the domain. Machine learning (ML), according to Arthur Samuel of IBM in 1959, gives "computers the ability to learn without being explicitly programmed." This generated a lot of interests within the U.S. and British governments with generous funding in furthering this field. It was a painfully a slow progress. In the beginning of 1980s and with the growing popularity of the microprocessors, AI research was given fresh lease by the commercial success of Expert Systems. Expert Systems is basically a kind AI platform that simulated the knowledge and analytical skills of human within a particular domain. This solution is to allow computers to learn from experience and understand the world in terms of a

hierarchy of concepts, with each concept defined through its relation to simpler. With enormous monetary gains, it looked quite formidable but with the fall of LISP Processor (LISP) around late 1980s, AI lost some traction. It was implicitly understood that the real challenge to AI proved to be solving the problems those were easy for people to perform but extremely difficult for people to describe them formally such as recognizing speech or faces in pictures.

Many of the early successes of AI took place in relatively sterile and formal environments and did not require computers to have much knowledge about the world. For example, IBM's Deep Blue chess-playing system defeated world champion Garry Kasparov in 1997. Chess has pretty straightforward rules, containing only 64 locations and 32 pieces that can move in strict direction and steps. Devising a successful chess strategy is a tremendous accomplishment, but the challenge is not due to the difficulty of describing these to chess pieces and allowable moves to the machine but the state space search required to check the various moves.

Computer scientists used neural networks (NNs) with many layers with new topologies and learning methods. This evolution of NNs has successfully solved complex problems in various domains. In the past decade, cognitive computing has emerged, the goal of which is to build systems that can learn and naturally interact with humans. Cognitive computing was demonstrated by IBM Watson by successfully defeating world-class opponents at the game Jeopardy (Kelly III, 2015).

Deep learning (DL) has already proved useful in many software disciplines, including computer vision, speech and audio processing, natural language processing (NLP), robotics, bioinformatics and chemistry, video games, search engines, online advertising, and finance (Bishop, 2006). The overview is organized in three sections. The evolution section gives a quick overview of the progress made till date. Section 2 is where all the DL applications are being explored and deployed in the various industries. Section 3 provides a glimpse of the future of DL.

2 | THE EVOLUTIONARY JOURNEY

DL networks before the 1960s were concentrated mainly in the variants of linear regressors popularly known as the supervised NNs (Rosenblatt, 1958). The general, workable algorithm for supervised deep feedforward multilayer perceptrons by Ivakhnenko and Lapa (1965) marked the origin of the DL networks, where the simple, connected processors used polynomial activation functions in conjunction with the Kolmogorov-Gabor polynomials. Ivakhnenko's NNs created hierarchical, distributed, internal representations of incoming data and approached in problem-dependent fashion to the numbers of layers and units per layer (Ivakhnenko, 1971). In the latter part of that decade, Fu's Syntactic pattern recognition methods start to learn from highly abstract, hierarchical data representations (Tikhonov, Arsenin, & John, 1977; Figure 1).

2.1 | Search tools

AI problems were fitted into the state space search for an optimized result. State space search is basically a process where successive configurations or states of an instance are considered, with the objective of finding a goal state with a desired

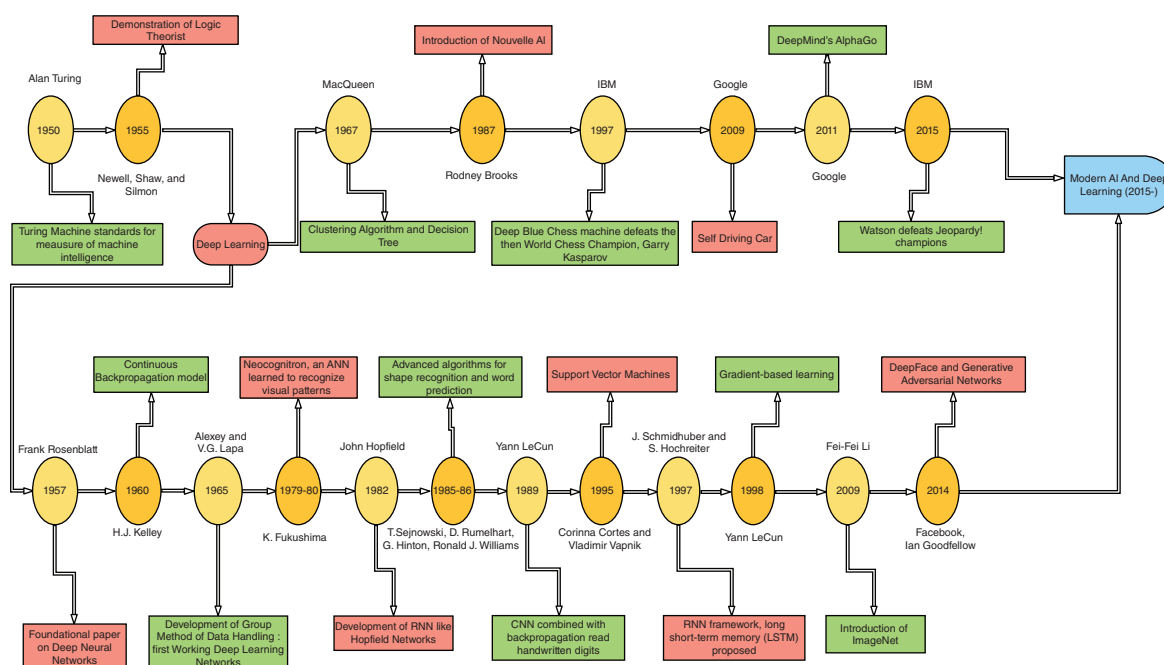


FIGURE 1 AI timeline

property. Problems are often modeled as a state space, a set of states that a problem can be in. The set of states forms a graph where two states are connected if there is an operation that can be performed to transform the first state into the second. State space search often differs from traditional computer science search methods because the state space is implicit: the typical state space graph is much too large to generate and store in memory. Instead, nodes are generated as they are explored, and typically discarded thereafter. A solution to a combinatorial search instance may consist of the goal state itself, or of a path from some initial state to the goal state (Haykin, 1999).

One of the earliest examples of AI as search was the development of a checkers-playing program. Arthur Samuel built the first such program on the IBM 701 Electronic Data Processing Machine, implementing an optimization to search trees called alpha-beta pruning. His program also recorded the reward for a specific move, allowing the application to learn with each game played (making it the first self-learning program). To increase the rate at which the program learned, Samuel programmed it to play itself, increasing its ability to play and learn. Although this can be applied to various problems, the main issue is of state space explosion.

2.2 | Perceptrons

The perceptron was an early supervised learning algorithm for single-layer NNs. Given an input feature vector, the perceptron algorithm could learn to classify inputs as belonging to a specific class. Using a training set, the network's weights and bias could be updated for linear classification. The perceptron was first implemented for the IBM 704, and then on custom hardware for image recognition.

As a linear classifier, the perceptron was capable of linear separable problems (Figure 2). The key example of the limitations of the perceptron was its inability to learn an exclusive OR (XOR) function. Multilayer perceptrons solved this problem and paved the way for more complex algorithms, network topologies, and DL.

2.3 | Clustering algorithms

The approach for the perceptrons was that of supervised type. Users provided data to train the network, and then test the network against new data. Clustering algorithms take a different approach called unsupervised learning. In this model, the algorithm organizes a set of feature vectors into clusters based on one or more attributes of the data (Hewlett Packard Enterprise, 2016; Figure 3).

One of the simplest algorithms is called *k*-means and requires very small amount of code. Although *k*-means is relatively efficient, one must specify *k* in advance. Depending on the data, other approaches might be more efficient, such as hierarchical or distribution-based clustering.

2.4 | Decision trees

Closely related to clustering is the decision tree. A decision tree is a predictive model about observations that lead to some conclusion. Conclusions are represented as leaves on the tree, while nodes are decision points where an observation diverges. Decision trees are built from decision tree learning algorithms, where the data set is split into subsets based on attribute value tests, through a process called recursive partitioning.

A useful aspect of decision trees is their inherent organization, which gives you the ability to graphically explain how you classified an item (Figure 4). Popular decision tree learning algorithms include C4.5 and the Classification and Regression Tree.

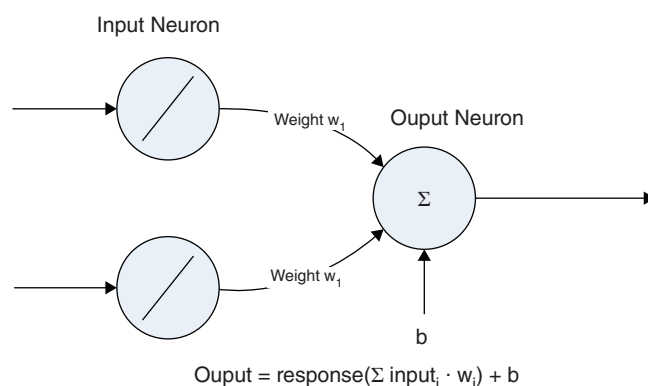


FIGURE 2 The basic perceptron

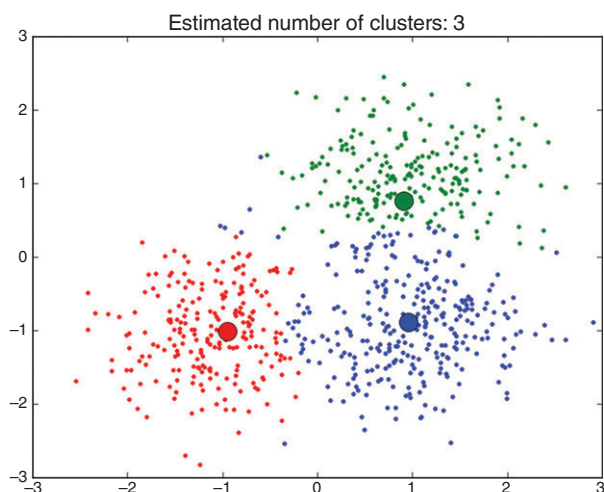


FIGURE 3 Output from a clustering classifier

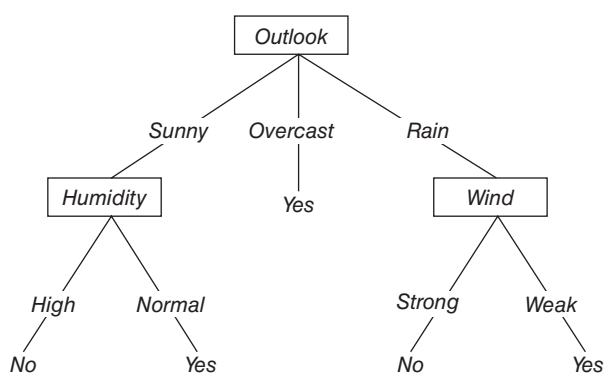


FIGURE 4 A sample decision tree

2.5 | Rule-based systems/Expert systems

The first system built on rules and inference, called Dendral, was developed in 1965, but it was not until the 1970s that these so-called expert systems hit their stride. A rules-based system is one that stores both knowledge and rules and uses a reasoning system to draw conclusions (Figure 5).

Rules-based systems have been applied to speech recognition, planning and control, and disease identification. One system developed in the 1990s for monitoring and diagnosing dam stability, called Kaleidos, is still in operation today.

2.6 | Machine learning

ML is a subfield of AI and computer science that has its roots in statistics and mathematical optimization. ML covers techniques in supervised and unsupervised learning for applications in prediction, analytics, and data mining. It is not restricted to DL, but covers other interesting areas as well.

2.7 | Backpropagation

The true power of NNs is their multilayer variant. Training single-layer perceptrons is straightforward, but the resulting network is not very powerful. The problem of how to train networks with multiple layers became the burning topic then. Backpropagation (BP) was developed to solve this. BP is an algorithm for training NNs that have many layers. It works in two phases. The first phase is the propagation of inputs through an NN to the final layer. This process is called feed forward. In the second phase, the algorithm computes an error, and then backpropagates this error (adjusting the weights) from the final layer to the first.

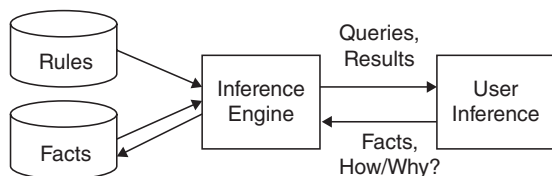


FIGURE 5 High-level rules-based system

The development of the continuous form of BP started in the early 1960s. In 1962, the refined derivation of BP based on the chain rule alone was published by Dreyfus, and the efficient, present form of BP was contributed by Linnainmaa (1970) for discrete sparse networks, which introduced a directly proportional relationship between the computation of the derivatives of the output error with respect to each weight and the number of weights.

Parallel to the introduction of cheaper GPUs in the market in the early 2000s, faster implementation of several different forms of NNs (Oh & Jung, 2004) and reduced error rates (Ciresan, Meier, Gambardella, & Schmidhuber, 2010) fostered revolutions in the NNs scholarship. The growing need to effectively utilize advanced computing units superseded the focus on advances in the algorithms.

During training, intermediate layers of the network organize themselves to map portions of the input space to the output space (Figure 6). BP, through supervised learning, identifies an error in the input-to-output mapping, and then adjusts the weights accordingly with a learning rate to correct this error. BP continues to be an important aspect of NN learning. With faster and cheaper computing resources, it continues to be applied to larger and denser networks.

2.8 | Convolutional NNs

Convolutional NNs (CNNs) are multilayer NNs that take their inspiration from the animal visual cortex. The architecture is useful in various applications, including image processing. The pioneering work of Fukushima (1975)) introduced the CNN architecture to the world. The first CNN was created by LeCun (1998) and LeCun et al. (1989), and at the time, the architecture focused on handwritten character-recognition tasks like reading postal codes.

The LeNet CNN (LeCun, 1998) architecture is made up of several layers that implement feature extraction, and then classification. The image is divided into receptive fields that feed into a convolutional layer that extracts features from the input image. The next step is pooling, which reduces the dimensionality of the extracted features (through down-sampling) while retaining the most important information (typically through max pooling). The algorithm then performs another convolution and pooling step that feeds into a fully connected, multilayer perceptron. The final output layer of this network is a set of nodes that identify features of the image (in this case, a node per identified number). Users can train the network through BP.

Commercialization of the CNN's started in the early 1990s with the application of BP to neocognitron-like CNNs (Ciresan et al., 2010) and using BPs for Max-Pooling CNNs (MPCNNs) (Ranzato, Huang, Boureau, & LeCun, 2007). The dramatic improvement in the record of MNIST handwritten digit dataset was attributed to efficient parallelized GPU-based MPCNNs. Computer vision communities benefited immensely from the revolutionary works on an ensemble of GPU-based MPCNNs and improved variants (Simonyan & Zisserman, 2015). Results from MPCNNs Image scanner were highly acceptable as compared to the previous work on CNNs without MP and speeded up the basic implementation by several orders of magnitude by ignoring the dispensable computations (Ciresan, Meier, Masci, Gambardella, & Schmidhuber, 2011).

Recent CNN applicatory usage in video classification (Karpathy et al., 2014) and multidigit numbers recognizer in Google Street View images (Goodfellow, Bulatov, Ibarz, Arnaud, & Shet, 2014) are credited to be among some of the most revolutionary computer vision scholarships made possible by the CNNs on GPU.

The use of deep layers of processing, convolutions, pooling, and a fully connected classification layer opened the door to various new applications of NNs (Figure 7). In addition to image processing, the CNN has been successfully applied to video recognition and many tasks within NLP. CNNs have also been efficiently implemented within General Purpose Graphics Processing Units (GPGPUs), greatly improving their performance.

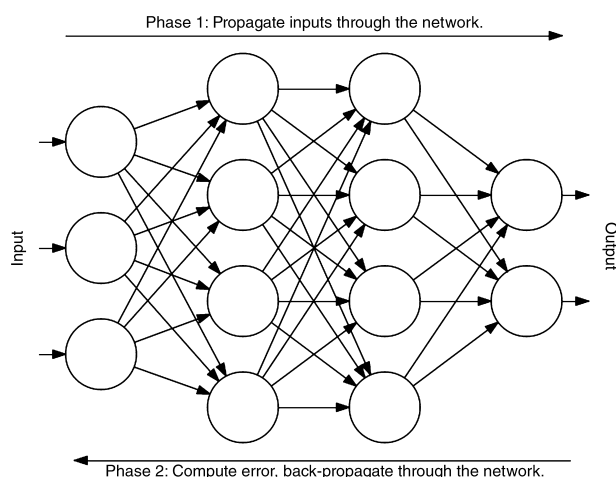


FIGURE 6 The process of backpropagation

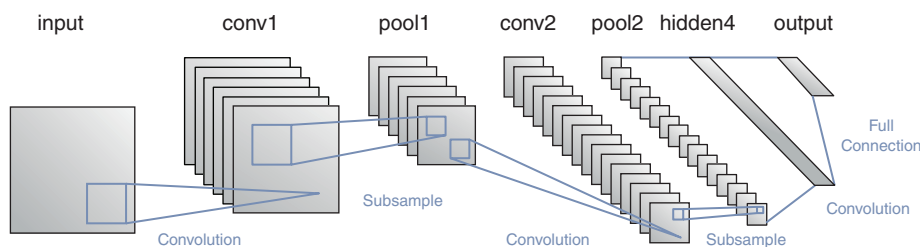


FIGURE 7 The LeNet convolutional neural network architecture

2.9 | Long short-term memory

In the case of BP, the network being trained was of feedforward type. In this architecture, users feed inputs into the network and propagate them forward through the hidden layers to the output layer. But, many other NN topologies exist. There is one that allows connections between nodes to form a directed cycle. These networks are called recurrent NNs, and they can feed backwards to prior layers or to subsequent nodes within their layer. This property makes these networks ideal for time series data. In 1997, a special kind of recurrent network was created called the long short-term memory (LSTM). The LSTM consists of memory cells that within a network remember values for a short or long time (Figure 8).

A memory cell contains gates that control how information flows into or out of the cell. The input gate controls when new information can flow into the memory. The forget gate controls how long an existing piece of information is retained. Finally, the output gate controls when the information contained in the cell is used in the output from the cell. The cell also contains weights that control each gate. The training algorithm, commonly BP-through-time (a variant of BP), optimizes these weights based on the resulting error. The LSTM has been applied to speech recognition, handwriting recognition, text-to-speech synthesis, image captioning, and various other tasks.

Training for NNs with unlimited depth involves RNNs learning to store the short-term memory representations of important previous observations for arbitrary time intervals. The active development of supervised LSTM RNNs made possible to advance problem-specific LSTM-like topologies (Bayer, Wierstra, Togelius, & Schmidhuber, 2009), wherein LSTM RNNs need to learn from the unlearnable DL tasks occurred hundreds or even thousands of discrete steps back from the present computation. The work of Graves et al. (Graves et al., 2009) underscored on the advanced applications of LSTM combined with recursive NNs, DAG-RNNs, and BRNNs.

A gradient-based method for finding RNN weights that maximize the probability of teacher-given label sequences, connectionist temporal classification (CTC) were used in the training of LSTM RNNs (Fernandez, Graves, & Schmidhuber, 2007). Progress was made where the speech recognition benchmark went beyond using the CTC-trained RNNs. LSTM based RNNs were successful in surpassing previous benchmarks for traditional HMMs in keyword spotting tasks (Fernandez et al., 2007) and in securing best results on TIMIT phoneme recognition benchmark (Graves, Mohamed, & Hinton, 2013). CTC-trained LSTM used by Google Voice has shown significant improvements and are also widely popular in syntactic parsing for NLP, text-to-speech synthesis, machine translation, and so on (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). The works on AMA memory (AMAMemory, 2015) proposed another novel approach for RNN based methods with faster memory control.

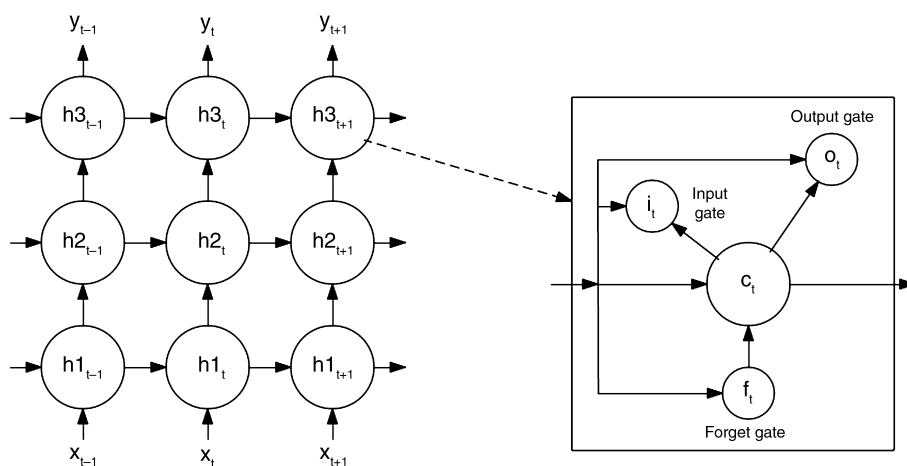


FIGURE 8 A long short-term memory network and memory cell

2.10 | Deep learning

DL is a relatively new set of methods that's changing ML in fundamental ways. DL is not an algorithm, per se, but rather a family of algorithms that implement deep networks with unsupervised learning. These networks are so deep that new methods of computation, such as GPGPUs, are required to build them in addition to clusters of compute nodes.

The CNN and LSTM algorithms have been combined to achieve several surprisingly intelligent tasks. As shown in Figure 9, CNNs and LSTMs have been used to identify, and then describe in natural language a picture or video.

DL algorithms have also been applied to facial recognition, identifying tuberculosis with 96% accuracy, self-driving vehicles, and many other complex problems.

However, despite the results of applying DL algorithms, problems exist that we have yet to solve. A recent application of DL to skin cancer detection found that the algorithm was more accurate than a board-certified dermatologist. But, where dermatologists could enumerate the factors that led to their diagnosis, there is no way to identify which factors a DL program used in its classification. This is called DL's black box problem.

Another application, called Deep Patient, was able to successfully predict disease given a patient's medical records. The application proved to be considerably better at forecasting disease than physicians—even for schizophrenia, which is notoriously difficult to predict. So, even though the models work well, no one can reach into the massive NNs to identify why.

3 | USE OF DL CONCEPTS IN THE INDUSTRY

This diagram shows the various interrelationship between the various domains—AI (which now getting morphed to Cognitive Computing) being the overarching domain (Figure 10).

Not surprisingly, the first companies to take advantage of the potential of AI at scale were large organizations such as Google, Facebook, and Amazon. The efforts of those early adopters attracted wide attention and inspired other organizations to begin exploring and developing AI solutions (Goodfellow, Bengio, & Courville, 2016; Schmidhuber, 2015). It would be interesting to note that Facebook is using LSTM for backend translation systems, which account for more than 2,000 translation directions and 4.5 billion translations each day in later half of 2017. It is also helping the speech recognition on over 2 billion Android phones. Google too uses LSTM for its translations. Apple, Microsoft, and Amazon have benefitted from the impeccable voice recognition capabilities using the LSTM. A good example of DL is perception, recognizing what is in an image, what people are saying when they are talking, helping robots explore the world and interact with it. DL is emerging as a central tool to solve perception problems in recent years. It is the state of the art having to do with computer vision and speech recognition. Increasingly people are finding that DL is a much better tool to solve problems.

Many companies today have made DL a central part of their ML toolkit. For example, Facebook, Google, and Uber are all using DL in their products. DL shines wherever there is lots of data and complex problems to solve and many companies today are facing lots of complicated problems. DL can be applied to many different fields.

As deep NNs become increasingly important to everything from self-driving cars to voice recognition, new libraries are making it much easier to use DL to solve real problems. Building a training a multilayer CNN would have taken hundreds of lines of code just a few years ago. We will have a quick look into some of the industries adopting DL applications

1. *Healthcare.* This is a big domain with application range from genomics, drug discovery, drug delivery, robot-aided surgery to diagnosis. One of the exciting areas at the very top end is radiological diagnosis. DL technology applied to medical imaging is one of the most disruptive technology radiology has seen since the advent of digital imaging. IBM Watson recently boosted itself with more than \$4 billion worth of new assets and the merge buyout represents Watson's full-force, billion-dollar entry into the imaging arena. Imaging may be a fairly small area compared to Watson's health care-wide sphere of activity, but it is certainly one of the most strategic for the group.

On the diagnostic image processing side, DL algorithms will help select and extract features from medical images as well

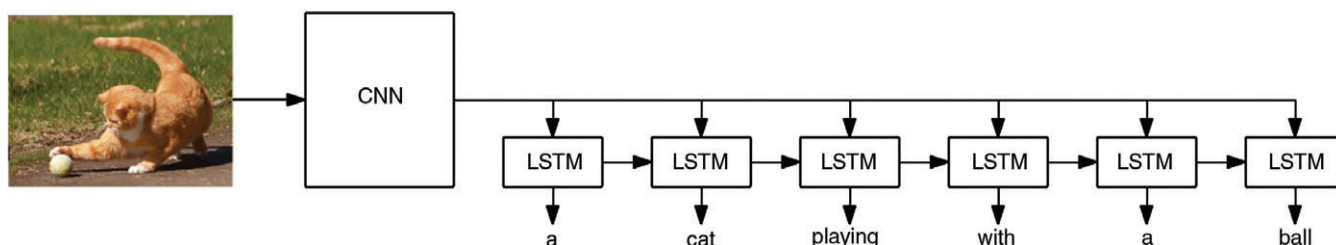


FIGURE 9 Combining convolutional neural networks and long short-term memory networks for image description

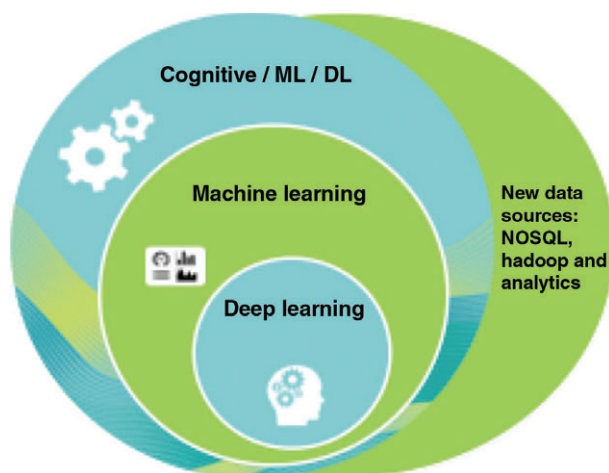


FIGURE 10 Understanding the inter domain relationships

as construct new ones; this will lead to representations of imaging studies never seen before. On the diagnostic image interpretation front, DL applications will help not only identify, classify, and quantify disease patterns from images, but will also allow to measure predictive targets and create actionable prediction models of care pathways (Terdiman, 2014). While the medical imaging community is contemplating whether this technology is a replacement for the radiologist, DL in health care and imaging is growing exponentially.

In a double-blinded study, the doctors at Manipal Hospitals found that Watson was concordant with the tumor board recommendations in 90% of breast cancer cases. The study was presented at the San Antonio Breast Cancer Symposium (Somashekhar, 2016). IBM Watson is being offered on trial basis for second opinions in oncological cases.

2. *Automotive—self-driving cars.* A self-driving car (also known as a driverless car, autonomous car, and robotic car) is a vehicle that is capable of sensing its environment and navigating without any human input (Dickmanns & Graefe, 1988). It has been an area of active interest since way back in 1930s. Autonomous cars use a variety of techniques to detect their surroundings, such as radar, laser, GPS, ultrasound, and most importantly computer vision. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage. There are already plenty of cars on the road with driver-assistance capabilities, but these cars still rely on users to take over when an unforeseen event occurs that the car is not programmed to respond to. Mercedes, Google, Tesla, and now Microsoft are companies investing heavily on self-driving cars. There are still many issues that need to be resolved such as software reliability, data security, reliable communication systems, navigation in bad/extreme weather conditions, and so forth.
3. *IT—cyber security.* As more devices become internet-enabled, hackers have an increasing number of entry points to infiltrate systems and cloud infrastructure. The best cyber security practices not only create more secure systems but can predict where the next attack will come from. This is critical since hackers are always on the hunt for the next vulnerable endpoint, so protecting against cyber-attack requires “thinking” like a hacker. IBM has developed the Watson Knowledge Studio that can be used to teach an ML model based on a labeled representative data subset. This model's machine-learning-based NLP capabilities can then be used to identify entities and relationships in a larger corpus of previously unseen documents. It is obvious that analysis of data deploying an ML language model would significantly decrease the time to research security threats. The model can work in tandem with other tools to research security threats in various connected sources of information and bring forth vital details about malware. A relationship between detection and research software can help address key challenges faced today in dealing with cyber threats. For the first time, output from Intrusion Detection Prevention Systems can automatically be researched against a plethora of information available in research papers and blogs. Furthermore, a cognitive system that is trained for the cyber security domain can adapt for evolving threats and needs of the industry.
4. *Weather forecast.* Predicting weather patterns has become increasingly important as governments and globe-spanning corporations prepare for broad-scale climate change and extreme weather events (Figure 11). The National Energy Research Scientific Computing Center, for example, used NEON, an open-source library from DL company to train a system to recognize extreme weather events based on visual pattern recognition. IBM too is using DL to help solar and wind companies better predict weather and improve alternative energy production. The results of such forecasts are increasingly being used in Air Transportation Industry, Military as well as Civilian early warning systems. Some of the benefits are elucidated below:

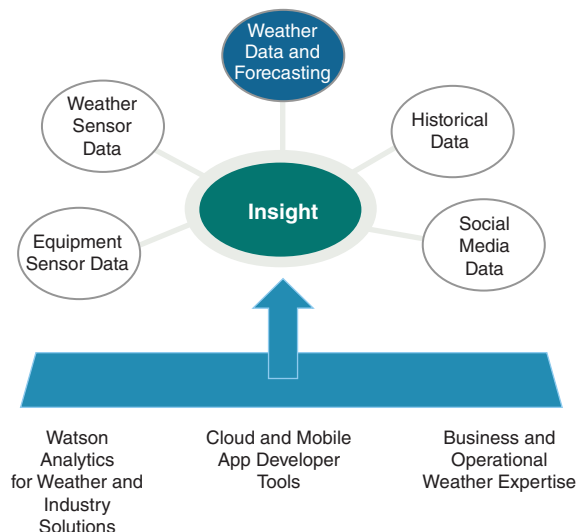


FIGURE 11 Components for weather forecasting

Increased lead time. The 15-day probabilistic tropical forecast can provide planners 1.5 days extra lead time to make better planning, positioning, and logistical decisions.

Rapid response. The algorithms track storms 24×7 and notify emergency planners within seconds of a disaster striking and provide a map-based visualization of first-pass view of the “damage zone,” so resources are pinpointed to the most serious impact zone first.

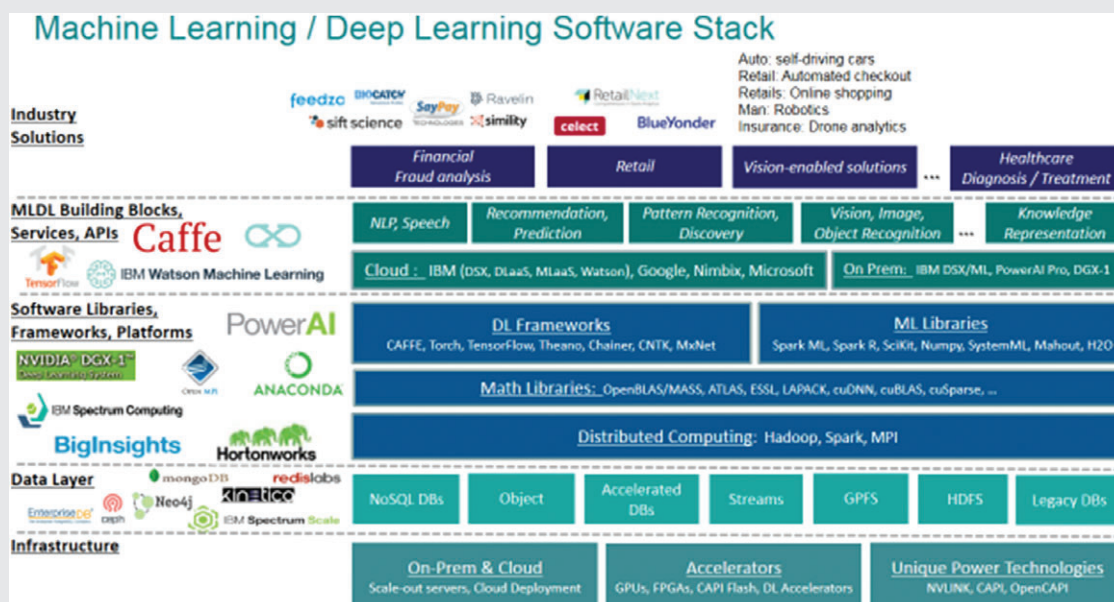
Scenario planning. Government agencies too often rely on a single forecast view when 2–5 realistic scenarios each having a 10–50% probability exist. Cities are obligated to plan for all scenarios to adequately protect citizens' life and property.

Deeper analytics. Weather is a major factor in crime rates and fires, having insight into which weather patterns increase or decrease criminal activity helps focus law enforcement staffing. Finer grain historic weather data integrated with additional factors provide insight into which days to apply additional resources.

5. *Agriculture.* Agriculture is one of the oldest and most unpredictable industries. Of recent it is becoming increasingly data as well as technology driven. More predictable crop output, based on weather forecasting and data-based estimates, could potentially even take some of the uncertainty out of the commodities markets. Satellite imagery is being fed into the DL machines with lots of computing power to come up with real-time predictions eliminating the need for actual surveys and site visits. Weather forecast inputs are also plugged in and then, based on patterns over time, the technology is designed to predict crop yields and production.
6. *Manufacturing.* Robots are being deployed at some major warehouses, for stacking objects as well for packaging like those operated by large warehouse operators like Amazon. The digitization of the physical world is powering a radical reimagining of how organizations and individuals learn, decide, and act on opportunity. It is shifting the ways they come together, creating new ecosystems that drive competitive advantage. And it is forever changing how all of us experience the world. Engineering and product development can get products to the marketplace faster with responsive products designed using actual customer usage data. Operations and real estate can enhance asset performance and utilization as well as run energy- and cost-efficient facilities. Manufacturers can use device data to predict failures before they occur. Retailers can track customer movement through their stores to optimize merchandising. Lots of manufacturers and product companies use equipment that uses computer vision or ML algorithms to assess quality. Specifically, high quality consumer electronics products like smartphones, tablets, and laptops are likely inspected with equipment running on the lines. Manufacturing has a whole lot of potential in various legs of its chain. DL applications have use in each and every segment of the manufacturing process.
7. *Retail.* The retail industry is undergoing a major transformation. Smarter consumers combine sophistication and frugality—moving from retailer to retailer based on factors like price, availability, and peer reviews. There is an urgent need for developing personal relationships with consumers. The need to listen and respond to customers through channels like social media in real-time is absolutely essential. A clear brand differentiation need to be established and retailers slow to respond to disruptive players will eventually be out of business. Retailers need to re-imagine everything about the way they connect, transact, and engage with consumers to create brand value. A recent study shows 36% of consumers are willing to share their GPS location with retailers in exchange for localized communications, 41% of consumers use social, location, and mobile devices to research and purchase products across retail channels and 32% of

BOX 1

THE INFOGRAPHICS SHOWS THE ML/DL COMPONENTS AT VARIOUS LAYERS



consumers are willing to share their social handles with retailers. With so much data in hand, cognitive computing will power the transformation in the retail industry and bring in smarter shopping (Box 1).

4 | COGNITIVE COMPUTING

AI and ML are filled with examples of biological inspiration. And, while early AI focused on the grand goals of building machines that mimicked the human brain, cognitive computing is working toward this goal.

The cognitive computing era follows the eras of programmable and tabulating systems and represents a huge leap forward. This is a new era because there is a fundamental difference in how these systems are built and how they interact with humans. Traditional programmable systems are fed data, knowledge, and information, and they carry out and return results of processing that is preprogrammed by humans. In the programmable systems era, humans do most of the directing.

The cognitive era on the other hand is about thinking itself—how we gather information, access it and make decisions. Cognitive-based systems learn and build knowledge, understanding natural language, and reason and interact more naturally with human beings than traditional programmable systems. While the term “reasoning” refers to how cognitive systems “reasoning” is a slippery term, we mostly mean that such systems demonstrate insights that are very human-like.

Cognitive systems are able to put content into context, providing confidence-weighted responses, with supporting evidence. They are also able to quickly find the proverbial needle in a haystack, identifying new patterns and insights. Cognitive systems extend the capabilities of humans by augmenting human decision making capacity and helping us make sense of the growing amount of data germane to a situation. Over time, cognitive systems will simulate even more closely how the brain works. In doing so, they could help us solve the world's most complex problems by penetrating the complexity of big data and exploiting the power of NLP and ML.

While tremendous advancements have been made over the past 50 years, cognitive computing is virtually in its infancy in terms of how this exciting technology could potentially evolve. Adopting and integrating cognitive solutions into an organization is a journey and not a destination. Therefore, organizations need to set realistic expectations and develop long-term plans with incremental milestones to benefit from the technology's future progression. Based on experience with clients and extensive research, we have identified multiple opportunities across industries for innovative application of cognitive computing today, as well as examined how the technology might evolve in the future.

Cognitive computing, building on NNs and DL, is applying knowledge from cognitive science to build systems that simulate human thought processes. However, rather than focus on a singular set of technologies, cognitive computing covers several disciplines, including ML, NLP, vision, and human–computer interaction.

An example of cognitive computing is IBM Watson, which demonstrated state-of-the-art question-and-answer interactions on *Jeopardy* but that IBM has since extended through a set of web services. These services expose application programming interfaces for visual recognition, speech-to-text, and text-to-speech function; language understanding and translation; and conversational engines to build powerful virtual agents.

There are a whole lot of frameworks that have evolved over the period of time. Some of the popular ones are Caffe (Donahue et al., 2014), Tensor Flow, Theano, DL4J, Torch, Julia, WSP, and others.

5 | CONCLUSION

This review presents an overall view of the evolution and successive progress made in the domain of AI specially focussing on DL evolution and its applications. Increasing processing power per square millimeter along with faster bus communication innovations between the CPU and the GPGPU is fuelling the rapid implementation of DL applications across industries. The primary use being vision, image/object recognition, NLP, speech, recommendation, prediction, pattern recognition, discovery, and knowledge representation.

A combination of the techniques and methods used in AI are broadly classified as cognitive computing. The opportunities for cognitive computing are compelling. There are four recommended steps for moving forward based upon the valuable insights and lessons learned from pioneering organizations those have been early adopters of this capability. Plan the cognitive journey—identifying candidate opportunities (use cases) for cognitive computing across mission and functional areas in the organization. This includes identifying target processes to be disrupted with cognitive solutions. Next is prototyping—the purpose of developing a prototype is to allow users to see what the end state of their developed use case could look like using visual design treatments and focusing on the workflow for the use case scenarios. This is a critical step in validating and refining the use case, increasing user understanding and gaining buy-in and further test the business case hypotheses. After the prototype has been validated, the complete solution needs to be developed and “the team” needs to be trained. The focus of this step is developing the solution around the priority use cases(s) defined in the earlier two stages. Investments required will be driven by the requirements and analysis conducted in these steps. As previously discussed, the system training process is an ongoing process that will continue on well after the initial deployment. Lastly, deploy the solution and find ways to evolve the organization's cognitive capabilities. Once the solution is deployed, even intensive learning can begin—for both the system and the solution users and stakeholders. This can be called the “Deploy, Explore Evolve” stage which comprises (a) deploying the solution, (b) continuous learning (for both the system and the system's users and stakeholders), (c) continuous improvement of the corpus, (d) further evolution of the system and domain processes, and (e) exploration of further use cases for the application of cognitive computing. Continuous tracking of business benefits and the accuracy levels of the solution is critical to assess and evaluate progress against key metrics.

Cognitive systems along with DL are speeding up the automation and accuracy required in many areas which till now was dependent on human skills. However, these are to improve the productivity and to deepen the relationship between the humans and the world. There has tremendous growth in the program development domain as well with more to be seen in near future.

CONFLICT OF INTEREST

The author has declared no conflicts of interest for this article.

RELATED WIREs ARTICLE

[Deep learning for sentiment analysis: successful approaches and future challenges](#)

REFERENCES

- AMAmemory. (2015). *Answer at reddit AMA (Ask Me Anything) on "memory networks" etc (with references)*.
- Bayer, J., Wierstra, D., Togelius, J., & Schmidhuber, J. (2009). *Evolving memory cell structures for sequence learning*. Paper presented at the Proceedings of Artificial Neural Networks (Vol. 2, pp. 755–764), ICANN Cyprus.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. ISBN 0387310738, 9780387310732.
- Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep big simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12), 3207–3220.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). *Flexible, high performance convolutional neural networks for image classification*. Paper presented at the International Joint Conference on Artificial Intelligence (pp. 1237–1242), Barcelona, Spain.

- Dickmanns, E., & Graefe, V. (1988). Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1, 241–261. <https://doi.org/10.1007/BF01212362>
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). *Decaf: A deep convolutional activation feature for generic visual recognition*. arxiv.org/pdf/1310.1531.pdf
- Fernandez, S., Graves, A., & Schmidhuber, J. (2007). *An application of recurrent neural networks to discriminative keyword spotting*. Paper presented at the Proceedings of Artificial Neural Networks (Vol. 2, pp. 220–229), Porto, Portugal.
- Fukushima, K. (1975). Cognitron: A self organizing multilayered neural network. *Biological Cybernetics*, 20, 121–136.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2014). *Multi-digit number recognition from street view imagery using deep convolutional neural networks*. <http://dblp.uni-trier.de/rec/bibtex/journals/corr/GoodfellowBIAS13>. arXiv:1312.6082 v4.
- Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 855–868.
- Graves, A., Mohamed, A.-R., & Hinton, G. E. (2013). *Speech recognition with deep recurrent neural networks*. Paper presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 6645–6649), Vancouver, Canada.
- Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Prentice Hall.
- Hewlett Packard Enterprise. (2016). *Augmented intelligence, helping humans make smarter decisions*. Retrieved from <http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA6-4478ENW.pdf>
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. <http://dblp.uni-trier.de/rec/bibtex/journals/corr/abs-1207-0580>. arXiv:1207.0580.
- Ivakhnenko, A. G. (1971). Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(4), 364–378.
- Ivakhnenko, A. G., & Lapa, V. G. (1965). *Cybernetic predicting devices*. New York: CCM Information Corporation.
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). *Large-scale video classification with convolutional neural networks*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio.
- Kelly, J., III. (2015). *Computing, cognition and the future of knowing*. IBM Research Cognitive Computing. USA: IBM Corporation.
- LeCun, Y. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Back-propagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551.
- Linnainmaa, S. (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors*. (Master's thesis). University of Helsinki.
- Oh, K.-S., & Jung, K. (2004). GPU implementation of neural networks. *Pattern Recognition*, 37(6), 1311–1314.
- Ranzato, M. A., Huang, F., Boureau, Y., & LeCun, Y. (2007). *Unsupervised learning of invariant feature hierarchies with applications to object recognition*. Paper presented at the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–8), Minneapolis, MN.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. <http://dblp.uni-trier.de/rec/bibtex/journals/corr/SimonyanZ14a>. arxiv.org/abs/1409.1556
- Somashekhar, S. P. (2016, December 9). SABCS 2016 Paper presented at the San Antonio Breast Cancer Symposium. San Antonio, USA.
- Terdiman, D. (2014). *IBM's TrueNorth processor mimics the human brain*. Retrieved from <http://www.cnet.com/news/ibms-truenorth-processor-mimics-the-human-brain/>
- Tikhonov, A. N., Arsenin, V. I., & John, F. (1977). *Solutions of ill-posed problems*. Washington, DC: V. H. Winston & Sons.

How to cite this article: Dutta S. An overview on the evolution and adoption of deep learning applications used in the industry. *WIREs Data Mining Knowl Discov*. 2018;8:e1257. <https://doi.org/10.1002/widm.1257>