Deep Learning: Fundamentals and History

Definition and Functioning

Deep learning, in machine learning, focuses on utilizing multilayered neural networks to perform tasks such as classification, regression, and representation learning. The field takes inspiration from biological neuroscience and is centered around stacking artificial neurons into layers and "training" them to process data. The adjective "deep" refers to the use of multiple layers (ranging from three to several hundred or thousands) in the network. Methods used can be supervised, semi-supervised or unsupervised.

Fundamentally, deep learning refers to a class of machine learning algorithms in which a hierarchy of layers is used to transform input data into a progressively more abstract and composite representation. For example, in an image recognition model, the raw input may be an image (represented as a tensor of pixels). The first representational layer may attempt to identify basic shapes such as lines and circles, the second layer may compose and encode arrangements of edges, the third layer may encode a nose and eyes, and the fourth layer may recognize that the image contains a face.

Importantly, a deep learning process can learn which features to optimally place at which level on its own. Prior to deep learning, machine learning techniques often involved hand-crafted feature engineering to transform the data into a more suitable representation for a classification algorithm to operate on. In the deep learning approach, features are not hand-crafted and the model discovers useful feature representations from the data automatically. This doesn't eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited.

No universally agreed-upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth higher than two. CAP of depth two has been shown to be a universal approximator in the sense that it can emulate any function. Beyond that, more layers don't add to the function approximator ability of the network. Deep models (CAP > two) are able to extract better features than shallow models and hence, extra layers help in learning the features effectively. Deep learning architectures can be constructed with a greedy layer-by-layer method.

Deep learning helps to disentangle these abstractions and pick out which features improve performance. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data is more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are deep belief networks.

Architectures and Common Applications

Some common deep learning network architectures include fully connected networks, deep belief networks, recurrent neural networks, convolutional neural networks, generative adversarial networks, transformers, and neural radiance fields. These architectures have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Origins and Historical Evolution

Early forms of neural networks were inspired by information processing and distributed communication nodes in biological systems, particularly the human brain. However, current neural networks do not intend to model the brain function of organisms and are generally seen as low-quality models for that purpose.

Most modern deep learning models are based on multi-layered neural networks such as convolutional neural networks and transformers, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines.

The term deep learning was introduced to the machine learning community by Rina Dechter in 1986, and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons. Although the history of its appearance is apparently more complicated.

Deep neural networks are generally interpreted in terms of the universal approximation theorem or probabilistic inference. The classic universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions. In 1989, the first proof was published by George Cybenko for sigmoid activation functions and was generalized to feed-forward multi-layer architectures in 1991 by Kurt Hornik. Recent work also

showed that universal approximation also holds for non-bounded activation functions such as Kunihiko Fukushima's rectified linear unit (ReLU).

The universal approximation theorem for deep neural networks concerns the capacity of networks with bounded width but where depth is allowed to grow. Lu et al. proved that if the width of a deep neural network with ReLU activation is strictly larger than the input dimension, then the network can approximate any Lebesgue integrable function; if the width is smaller than or equal to the input dimension, then a deep neural network is not a universal approximator.

The probabilistic interpretation derives from the field of machine learning. It features inference, as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function. The probabilistic interpretation led to the introduction of dropout as a regularizer in neural networks. The probabilistic interpretation was introduced by researchers including Hopfield, Widrow and Narendra and popularized in surveys such as the one by Bishop.

Types of Artificial Neural Networks

There are two types of artificial neural network (ANN): feedforward neural networks (FNN) or multilayer perceptrons (MLP) and recurrent neural networks (RNN). RNNs have cycles in their connectivity structure, FNNs don't. In the 1920s, Wilhelm Lenz and Ernst Ising created the Ising model, which is essentially a non-learning RNN architecture consisting of neuron-like threshold elements. In 1972, Shun'ichi Amari made this architecture adaptive. His learning RNN was republished by John Hopfield in 1982. Other early recurrent neural networks were published by Kaoru Nakano in 1971.

Already in 1948, Alan Turing produced work on "Intelligent Machinery" that was not published in his lifetime, containing "ideas related to artificial evolution and learning RNNs". Frank Rosenblatt (1958) proposed the perceptron, an MLP with 3 layers: an input layer, a hidden layer with randomized weights that didn't learn, and an output layer. He later published a 1962 book that also introduced variants and computer experiments, including a version with four-layer perceptrons "with adaptive preterminal networks" where the last two layers have learned weights (here he credits H. D. Block and B. W. Knight). The book cites an earlier network by R. D. Joseph (1960) "functionally equivalent to a variation of" this four-layer system.

The first working deep learning algorithm was the Group method of data handling, a method to train arbitrarily deep neural networks, published by Alexey Ivakhnenko and Lapa in 1965. They regarded it as a form of polynomial regression, or a generalization of Rosenblatt's perceptron. A 1971 paper described a deep network with eight layers trained by this method, which is based on layer by layer training through regression analysis. Superfluous hidden units are pruned using a separate validation set. Since the activation functions of the nodes are Kolmogorov-Gabor polynomials, these were also the first deep networks with multiplicative units or "gates."

The first deep learning multilayer perceptron trained by stochastic gradient descent was published in 1967 by Shun'ichi Amari. In computer experiments conducted by Amari's student Saito, a five-layer MLP with two modifiable layers learned internal representations to classify non-linearly separable pattern classes. Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique. In 1969, Kunihiko Fukushima introduced the ReLU (rectified linear unit) activation function. The rectifier has become the most popular activation function for deep learning.

Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers began with the Neocognitron introduced by Kunihiko Fukushima in 1979, though not trained by backpropagation. Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673 to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt, but he didn't know how to implement this, although Henry J. Kelley had a continuous precursor of backpropagation in 1960 in the context of control theory. The modern form of backpropagation was first published in Seppo Linnainmaa's master thesis (1970). G.M. Ostrovski et al. republished it in 1971.

Paul Werbos applied backpropagation to neural networks in 1982 (his 1974 PhD thesis, reprinted in a 1994 book, did not yet describe the algorithm). In 1986, David E. Rumelhart et al. popularized backpropagation but didn't cite the original work.

The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel to apply CNN to phoneme recognition. It used convolutions, weight sharing, and backpropagation. In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition. In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days. In 1990, Wei Zhang implemented a CNN on optical computing hardware. In 1991, a CNN was applied to medical image object segmentation and breast cancer detection in mammograms. LeNet-5 (1998), a 7-level CNN by Yann LeCun et al., that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel images.

Recurrent neural networks (RNN) were further developed in the 1980s. Recurrence is used for sequence processing, and when a recurrent network is unrolled, it mathematically resembles a deep feedforward layer. Consequently, they have similar properties and issues, and their developments had mutual influences. In RNN, two early influential works were the Jordan network (1986) and the Elman network (1990), which applied RNN to study problems in cognitive psychology.

In the 1980s, backpropagation didn't work well for deep learning with long credit assignment paths. To overcome this problem, in 1991, Jürgen Schmidhuber proposed a hierarchy of RNNs pre-trained one level at a time by self-supervised learning where each RNN tries to predict its own next input, which is the next unexpected input of the RNN below. This "neural history

compressor" uses predictive coding to learn internal representations at multiple self-organizing time scales. This can substantially facilitate downstream deep learning. The RNN hierarchy can be collapsed into a single RNN, by distilling a higher level chunker network into a lower level automatizer network. In 1993, a neural history compressor solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time. The "P" in ChatGPT refers to such pre-training.

Sepp Hochreiter's diploma thesis (1991) implemented the neural history compressor, and identified and analyzed the vanishing gradient problem. Hochreiter proposed recurrent residual connections to solve the vanishing gradient problem. This led to the long short-term memory (LSTM), published in 1995. LSTM can learn "very deep learning" tasks with long credit assignment paths that require memories of events that happened thousands of discrete time steps before. That LSTM wasn't yet the modern architecture, which required a "forget gate," introduced in 1999, which became the standard RNN architecture.

In 1991, Jürgen Schmidhuber also published adversarial neural networks that contest with each other in the form of a zero-sum game, where one network's gain is the other network's loss. The first network is a generative model that models a probability distribution over output patterns. The second network learns by gradient descent to predict the reactions of the environment to these patterns. This was called "artificial curiosity." In 2014, this principle was used in generative adversarial networks (GANs).

During 1985–1995, inspired by statistical mechanics, several architectures and methods were developed by Terry Sejnowski, Peter Dayan, Geoffrey Hinton, etc., including the Boltzmann machine, restricted Boltzmann machine, Helmholtz machine, and the wake-sleep algorithm. These were designed for unsupervised learning of deep generative models. However, those were more computationally expensive compared to backpropagation. The Boltzmann machine learning algorithm, published in 1985, was briefly popular before being eclipsed by the backpropagation algorithm in 1986. A 1988 network became state of the art in protein structure prediction, an early application of deep learning to bioinformatics.

The Deep Learning Resurgence

Both shallow and deep learning (e.g., recurrent nets) of ANNs for speech recognition have been explored for many years. These methods never outperformed non-uniform internal-handcrafting Gaussian mixture model / Hidden Markov model (GMM-HMM) technology based on generative models of speech trained discriminatively. Key difficulties have been analyzed, including gradient diminishing and weak temporal correlation structure in neural predictive models. Additional difficulties were the lack of training data and limited computing power. Most speech recognition researchers moved away from neural nets to pursue generative modeling. An exception was at SRI International in the late 1990s. Funded by the US government's NSA and DARPA, SRI researched in speech and speaker recognition. The speaker recognition team led by Larry Heck reported significant success with deep neural networks in speech processing in the 1998 NIST Speaker Recognition benchmark. It was deployed in the Nuance Verifier, representing the first major industrial application of deep learning.

The principle of elevating "raw" features over hand-crafted optimization was first explored successfully in the architecture of deep autoencoders on the "raw" spectrogram or linear filter-bank features in the late 1990s, showing its superiority over the Mel-Cepstral features that contain stages of fixed transformation from spectrograms. The raw features of speech, waveforms, later produced excellent larger-scale results.

Neural networks entered a lull, and simpler models that use task-specific handcrafted features such as Gabor filters and support vector machines (SVMs) became the preferred choices in the 1990s and 2000s, because of artificial neural networks' computational cost and a lack of understanding of how the brain wires its biological networks.

In 2003, LSTM became competitive with traditional speech recognizers on certain tasks. In 2006, Alex Graves, Santiago Fernández, Faustino Gomez, and Schmidhuber combined it with connectionist temporal classification (CTC) in stacks of LSTMs. In 2009, it became the first RNN to win a pattern recognition contest, in connected handwriting recognition.

In 2006, publications by Geoff Hinton, Ruslan Salakhutdinov, Osindero and Teh on deep belief networks were developed for generative modeling. They are trained by training one restricted Boltzmann machine, then freezing it and training another one on top of the first one, and so on, then optionally fine-tuned using supervised backpropagation. They could model high-dimensional probability distributions, such as the distribution of MNIST images, but convergence was slow.

The impact of deep learning in industry began in the early 2000s, when CNNs already processed an estimated 10% to 20% of all the checks written in the US, according to Yann LeCun. Industrial applications of deep learning to large-scale speech recognition started around 2010. The 2009 NIPS Workshop on Deep Learning for Speech Recognition was motivated by the limitations of deep generative models of speech, and the possibility that given more capable hardware and large-scale data sets that deep neural nets might become practical. It was believed that pre-training DNNs using generative models of deep belief nets (DBN) would overcome the main difficulties of neural nets. However, it was discovered that replacing pre-training with large amounts of training data for straightforward backpropagation when using DNNs with large, context-dependent output layers produced error rates dramatically lower than then-state-of-the-art Gaussian mixture model (GMM)/Hidden Markov Model (HMM) and also than more-advanced generative model-based systems. The nature of the recognition errors produced by the two types of systems was characteristically different, offering technical insights into how to integrate deep learning into the existing highly efficient, run-time speech decoding system deployed by all major speech recognition systems. Analysis around 2009–2010, contrasting the GMM (and other generative speech models) vs. DNN models, stimulated early industrial investment in deep learning for speech recognition. That analysis was done with comparable performance (less than 1.5% in error rate) between discriminative DNNs and generative models.

In 2010, researchers extended deep learning from TIMIT to large vocabulary speech recognition, by adopting large output layers of the DNN based on context-dependent HMM states constructed by decision trees.

The Deep Learning Revolution

The deep learning revolution started around CNN- and GPU-based computer vision. Although CNNs trained by backpropagation had been around for decades and GPU implementations of NNs for years, including CNNs, faster implementations of CNNs on GPUs were needed to progress on computer vision. Later, as deep learning becomes widespread, specialized hardware and algorithm optimizations were developed specifically for deep learning.

A key advance for the deep learning revolution was hardware advances, especially GPUs. Some early work dated back to 2004. In 2009, Raina, Madhavan, and Andrew Ng reported a 100M deep belief network trained on 30 Nvidia GeForce GTX 280 GPUs, an early demonstration of GPU-based deep learning. They reported up to 70 times faster training.

In 2011, a CNN named DanNet by Dan Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber achieved for the first time superhuman performance in a visual pattern recognition contest, outperforming traditional methods by a factor of 3. It then won more contests. They also showed how max-pooling CNNs on GPU improved performance significantly.

In 2012, Andrew Ng and Jeff Dean created an FNN that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images taken from YouTube videos. In October 2012, AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton won the large-scale ImageNet competition by a significant margin over shallow machine learning methods. Further incremental improvements included the VGG-16 network by Karen Simonyan and Andrew Zisserman and Google's Inceptionv3.

The success in image classification was then extended to the more challenging task of generating descriptions (captions) for images, often as a combination of CNNs and LSTMs. In 2014, the state of the art was training "very deep neural network" with 20 to 30 layers. Stacking too many layers led to a steep reduction in training accuracy, known as the "degradation" problem. In 2015, two techniques were developed to train very deep networks: the Highway Network was published in May 2015, and the residual neural network (ResNet) in Dec 2015. ResNet behaves like an open-gated Highway Net.

Around the same time, deep learning started impacting the field of art. Early examples included Google DeepDream (2015), and neural style transfer (2015), both of which were based on pretrained image classification neural networks, such as VGG-19.

Generative adversarial networks (GANs) by (Ian Goodfellow et al., 2014) (based on Jürgen Schmidhuber's principle of artificial curiosity) became state of the art in generative modeling during the 2014-2018 period. Excellent image quality is achieved by Nvidia's StyleGAN (2018) based on the Progressive GAN by Tero Karras et al. Here the GAN generator is grown from small to large scale in a pyramidal fashion. Image generation by GAN reached popular success and provoked discussions concerning deepfakes. Diffusion models (2015) have eclipsed GANs in generative modeling since then, with systems such as DALL·E 2 (2022) and Stable Diffusion (2022).

In 2015, Google's speech recognition improved by 49% by an LSTM-based model, which they made available through Google Voice Search on smartphone.

Deep learning is part of state-of-the-art systems in various disciplines, particularly computer vision and automatic speech recognition (ASR). Results on commonly used evaluation sets such as TIMIT (ASR) and MNIST (image classification), as well as a range of large-vocabulary speech recognition tasks have steadily improved. Convolutional neural networks were superseded for ASR by LSTM, but are more successful in computer vision. Yoshua Bengio, Geoffrey Hinton and Yann LeCun were awarded the 2018 Turing Award for "conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing".

Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming.

An ANN is based on a collection of connected units called artificial neurons (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream.

Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input) to the last (output) layer, possibly after traversing the layers multiple times. The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information.

Neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. As of 2017, neural networks typically have a few thousand to a few million units and millions of connections. Despite this number being several orders of magnitude less than the number of neurons in a human brain, these networks can perform many tasks at a level beyond that of humans (e.g., recognizing faces, or playing "Go").

Deep Neural Networks (DNNs)

A deep neural network (DNN) is an artificial neural network with multiple layers between the input and output layers. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions. These components as a whole function in a way that mimics functions of the human brain, and can be trained like any other ML algorithm. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNNs have many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights," to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn't accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential until it determines the correct mathematical manipulation to fully process the data.

Recurrent neural networks, in which data can flow in any direction, are used for applications such as language modeling. Long short-term memory is particularly effective for this use. Convolutional neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR).

Challenges of Deep Learning

As with ANNs, many issues can arise with naively trained DNNs. Two common issues are overfitting and computation time. DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Regularization methods such as Ivakhnenko's unit pruning or weight decay (L2-regularization) or sparsity (L1 -regularization) can be applied during training to combat overfitting. Alternatively, dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies. Another interesting recent development is research into models of just enough complexity through an estimation of the intrinsic complexity of the task being modeled. This approach has been successfully applied for multivariate time series prediction tasks such as traffic prediction. Finally, data can be augmented via methods such as cropping and rotating such that smaller training sets can be increased in size to reduce the chances of overfitting.

DNNs must consider many training parameters, such as the size (number of layers and number of units per layer), the learning rate, and initial weights. Sweeping through the parameter space for optimal parameters may not be feasible due to the cost in time and computational resources. Various tricks, such as batching (computing the gradient on several training examples at once rather than individual examples) speed up computation. Large processing capabilities of many-core architectures (such as GPUs or the Intel Xeon Phi) have produced significant speedups in training because of the suitability of such processing architectures for matrix and vector computations.

Alternatively, engineers may look for other types of neural networks with more straightforward and convergent training algorithms. CMAC (cerebellar model articulation controller) is one such kind of neural network. It doesn't require learning rates or randomized initial weights. The training process can be guaranteed to converge in one step with a new batch of data, and the computational complexity of the training algorithm is linear with respect to the number of neurons involved.

Since the 2010s, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks that contain many layers of non-linear hidden units and a very large output layer. By 2019,

graphics processing units (GPUs), often with AI-specific enhancements, had displaced CPUs as the dominant method for training large-scale commercial cloud AI. OpenAI estimated the hardware.