What is deep learning?

Deep learning is a subset of machine learning that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain. Some form of deep learning powers most of the artificial intelligence (AI) applications in our lives today.

The chief difference between deep learning and machine learning is the structure of the underlying neural network architecture. “Nondeep,” traditional machine learning models use simple neural networks with one or two computational layers. Deep learning models use three or more layers—but typically hundreds or thousands of layers—to train the models.

While supervised learning models require structured, labeled input data to make accurate outputs, deep learning models can use unsupervised learning. With unsupervised learning, deep learning models can extract the characteristics, features and relationships they need to make accurate outputs from raw, unstructured data. Additionally, these models can even evaluate and refine their outputs for increased precision.

Deep learning is an aspect of data science that drives many applications and services that improve automation, performing analytical and physical tasks without human intervention. This enables many everyday products and services—such as digital assistants, voice-enabled TV remotes, credit card fraud detection, self-driving cars and generative AI.

3D design of balls rolling on a track

The latest AI News + Insights

Discover expertly curated insights and news on AI, cloud and more in the weekly Think Newsletter.

Subscribe today

How deep learning works

Neural networks, or artificial neural networks, attempt to mimic the human brain through a combination of data inputs, weights and bias—all acting as silicon neurons. These elements work together to accurately recognize, classify and describe objects within the data.

Deep neural networks consist of multiple layers of interconnected nodes, each building on the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called visible layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

Another process called backpropagation uses algorithms, such as gradient descent, to calculate errors in predictions, and then adjusts the weights and biases of the function by moving backwards through the layers to train the model. Together, forward propagation and backpropagation enable a neural network to make predictions and correct for any errors. Over time, the algorithm becomes gradually more accurate.

Deep learning requires a tremendous amount of computing power. High-performance graphical processing units (GPUs) are ideal because they can handle a large volume of calculations in multiple cores with copious memory available. Distributed cloud computing might also assist. This level of computing power is necessary to train deep algorithms through deep learning. However, managing multiple GPUs on premises can create a large demand on internal resources and be incredibly costly to scale. For software requirements, most deep learning apps are coded with one of these three learning frameworks: JAX, PyTorch or TensorFlow.

Decoding AI: Weekly News Roundup

Join our world-class panel of engineers, researchers, product leaders and more as they cut through the AI noise to bring you the latest in AI news and insights.

Watch the latest podcast episodes

Types of deep learning models

Deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets. Here are six. Each has its own advantages and they are presented here roughly in the order of their development, with each successive model adjusting to overcome a weakness in a previous model.

One potential weakness across them all is that deep learning models are often “black boxes,” making it difficult to understand their inner workings and posing interpretability challenges. But this can be balanced against the overall benefits of high accuracy and scalability.

CNNs

Convolutional neural networks (CNNs or ConvNets) are used primarily in computer vision and image classification applications. They can detect features and patterns within images and videos, enabling tasks such as object detection, image recognition, pattern recognition and face recognition. These networks harness principles from linear algebra, particularly matrix multiplication, to identify patterns within an image.

CNNs are a specific type of neural network, which is composed of node layers, containing an input layer, one or more hidden layers and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

At least three main types of layers make up a CNN: a convolutional layer, pooling layer and fully connected (FC) layer. For complex uses, a CNN might contain up to thousands of layers, each layer building on the previous layers. By “convolution”—working and reworking the original input—detailed patterns can be discovered. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

CNNs are distinguished from other neural networks by their superior performance with image, speech or audio signal inputs. Before CNNs, manual and time-consuming feature extraction methods were used to identify objects in images. However, CNNs now provide a more scalable approach to image classification and object recognition tasks, and process high-dimensional data. And CNNs can exchange data between layers, to deliver more efficient data processing. While information might be lost in the pooling layer, this might be outweighed by the benefits of CNNs, which can help to reduce complexity, improve efficiency and limit risk of overfitting.

There are other disadvantages to CNNs, which are computationally demanding—costing time and budget, requiring many graphical processing units (GPUs). They also require highly trained experts with cross-domain knowledge, and careful testing of configurations, hyperparameters and configurations.

RNNs

Recurrent neural networks (RNNs) are typically used in natural language and speech recognition applications as they use sequential or time-series data. RNNs can be identified by their feedback loops. These learning algorithms are primarily used when using time-series data to make predictions about future outcomes. Use cases include stock market predictions or sales forecasting, or ordinal or temporal problems, such as language translation, natural language processing (NLP), speech recognition and image captioning. These functions are often incorporated into popular applications such as Siri, voice search and Google Translate.

RNNs use their “memory” as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of RNNs depends on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions.

RNNs share parameters across each layer of the network and share the same weight parameter within each layer of the network, with the weights adjusted through the processes of backpropagation and gradient descent to facilitate reinforcement learning.

RNNs use a backpropagation through time (BPTT) algorithm to determine the gradients, which is slightly different from traditional backpropagation as it is specific to sequence data. The principles of BPTT are the same as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. BPTT differs from the traditional approach in that BPTT sums errors at each time step, whereas feedforward networks do not need to sum errors as they do not share parameters across each layer.

An advantage over other neural network types is that RNNs use both binary data processing and memory. RNNs can plan out multiple inputs and productions so that rather than delivering only one result for a single input, RNNs can produce one-to-many, many-to-one or many-to-many outputs.

There are also options within RNNs. For example, the long short-term memory (LSTM) network is superior to simple RNNs by learning and acting on longer-term dependencies.

However, RNNs tend to run into two basic problems, known as exploding gradients and vanishing gradients. These issues are defined by the size of the gradient, which is the slope of the loss function along the error curve.

When the gradient is vanishing and is too small, it continues to become smaller, updating the weight parameters until they become insignificant—that is: zero (0). When that occurs, the algorithm is no longer learning.

Exploding gradients occur when the gradient is too large, creating an unstable model. In this case, the model weights grow too large, and they will eventually be represented as NaN (not a number). One solution to these issues is to reduce the number of hidden layers within the neural network, eliminating some of the complexity in the RNN models.

Some final disadvantages: RNNs might also require long training time and be difficult to use on large datasets. Optimizing RNNs add complexity when they have many layers and parameters.

Autoencoders and variational autoencoders

Deep learning made it possible to move beyond the analysis of numerical data, by adding the analysis of images, speech and other complex data types. Among the first class of models to achieve this were variational autoencoders (VAEs). They were the first deep-learning models to be widely used for generating realistic images and speech, which empowered deep generative modeling by making models easier to scale—which is the cornerstone of what we think of as generative AI.

Autoencoders work by encoding unlabeled data into a compressed representation, and then decoding the data back into its original form. Plain autoencoders were used for a variety of purposes, including reconstructing corrupted or blurry images. Variational autoencoders added the critical ability not just to reconstruct data, but also to output variations on the original data.

This ability to generate novel data ignited a rapid-fire succession of new technologies, from generative adversarial networks (GANs) to diffusion models, capable of producing ever more realistic—but fake—images. In this way, VAEs set the stage for today’s generative AI.

Autoencoders are built out of blocks of encoders and decoders, an architecture that also underpins today’s large language models. Encoders compress a dataset into a dense representation, arranging similar data points closer together in an abstract space. Decoders sample from this space to create something new while preserving the dataset’s most important features.

The biggest advantage to autoencoders is the ability to handle large batches of data and show input data in a compressed form, so the most significant aspects stand out—enabling anomaly detection and classification tasks. This also speeds transmission and reduces storage requirements. Autoencoders can be trained on unlabeled data so they might be used where labeled data is not available. When unsupervised training is used, there is a time savings advantage: deep learning algorithms learn automatically and gain accuracy without needing manual feature engineering. In addition, VAEs can generate new sample data for text or image generation.

There are disadvantages to autoencoders. The training of deep or intricate structures can be a drain on computational resources. And during unsupervised training, the model might overlook the needed properties and instead simply replicate the input data. Autoencoders might also overlook complex data linkages in structured data so that it does not correctly identify complex relationships.

GANs

Generative adversarial networks (GANs) are neural networks that are used both in and outside of artificial intelligence (AI) to create new data resembling the original training data. These can include images appearing to be human faces—but are generated, not taken of real people. The “adversarial” part of the name comes from the back-and-forth between the two portions of the GAN: a generator and a discriminator.

The generator creates something: images, video or audio and then producing an output with a twist. For example, a horse can be transformed into a zebra with some degree of accuracy. The result depends on the input and how well-trained the layers are in the generative model for this use case.

The discriminator is the adversary, where the generative result (fake image) is compared against the real images in the dataset. The discriminator tries to distinguish between the real and fake images, video or audio.

GANs train themselves. The generator creates fakes while the discriminator learns to spot the differences between the generator's fakes and the true examples. When the discriminator is able to flag the fake, then the generator is penalized. The feedback loop continues until the generator succeeds in producing output that the discriminator cannot distinguish.

The prime GAN benefit is creating realistic output that can be difficult to distinguish from the originals, which in turn may be used to further train machine learning models. Setting up a GAN to learn is straightforward, since they are trained by using unlabeled data or with minor labeling. However, the potential disadvantage is that the generator and discriminator might go back-and-forth in competition for a long time, creating a large system drain. One training limitation is that a huge amount of input data might be required to obtain a satisfactory output. Another potential problem is “mode collapse,” when the generator produces a limited set of outputs rather than a wider variety.

Diffusion models

Diffusion models are generative models that are trained using the forward and reverse diffusion process of progressive noise-addition and denoising. Diffusion models generate data—most often images—similar to the data on which they are trained, but then overwrite the data used to train them. They gradually add Gaussian noise to the training data until it’s unrecognizable, then learn a reversed “denoising” process that can synthesize output (usually images) from random noise input.

A diffusion model learns to minimize the differences of the generated samples versus the desired target. Any discrepancy is quantified and the model's parameters are updated to minimize the loss—training the model to produce samples closely resembling the authentic training data.

Beyond image quality, diffusion models have the advantage of not requiring adversarial training, which speeds the learning process and also offering close process control. Training is more stable than with GANs and diffusion models are not as prone to mode collapse.

But, compared to GANs, diffusion models can require more computing resources to train, including more fine-tuning. IBM Research® has also discovered that this form of generative AI can be hijacked with hidden backdoors, giving attackers control over the image creation process so that AI diffusion models can be tricked into generating manipulated images.

Transformer models

Transformer models combine an encoder-decoder architecture with a text-processing mechanism and have revolutionized how language models are trained. An encoder converts raw, unannotated text into representations known as embeddings; the decoder takes these embeddings together with previous outputs of the model, and successively predicts each word in a sentence.

Using fill-in-the-blank guessing, the encoder learns how words and sentences relate to each other, building up a powerful representation of language without having to label parts of speech and other grammatical features. Transformers, in fact, can be pretrained at the outset without a particular task in mind. After these powerful representations are learned, the models can later be specialized—with much less data—to perform a requested task.

Several innovations make this possible. Transformers process words in a sentence simultaneously, enabling text processing in parallel, speeding up training. Earlier techniques including recurrent neural networks (RNNs) processed words one by one. Transformers also learned the positions of words and their relationships—this context enables them to infer meaning and disambiguate words such as “it” in long sentences.

By eliminating the need to define a task upfront, transformers made it practical to pretrain language models on vast amounts of raw text, enabling them to grow dramatically in size. Previously, labeled data was gathered to train one model on a specific task. With transformers, one model trained on a massive amount of data can be adapted to multiple tasks by fine-tuning it on a small amount of labeled task-specific data.

Language transformers today are used for nongenerative tasks such as classification and entity extraction as well as generative tasks including machine translation, summarization and question answering. Transformers have surprised many people with their ability to generate convincing dialog, essays and other content.

Natural language processing (NLP) transformers provide remarkable power since they can run in parallel, processing multiple portions of a sequence simultaneously, which then greatly speeds training. Transformers also track long-term dependencies in text, which enables them to understand the overall context more clearly and create superior output. In addition, transformers are more scalable and flexible in order to be customized by task.

As to limitations, because of their complexity, transformers require huge computational resources and a long training time. Also, the training data must be accurately on-target, unbiased and plentiful to produce accurate results.

Deep learning use cases

The number of uses for deep learning grows every day. Here are just a few of the ways that it is now helping businesses become more efficient and better serve their customers.

Application modernization

Generative AI can enhance the capabilities of developers and reduce the ever-widening skills gap in the domains of application modernization and IT automation. Generative AI for coding is possible because of recent breakthroughs in large language model (LLM) technologies and natural language processing (NLP). It uses deep learning algorithms and large neural networks trained on vast datasets of existing source code. Training code generally comes from publicly available code produced by open-source projects.

Programmers can enter plain text prompts describing what they want the code to do. Generative AI tools suggest code snippets or full functions, streamlining the coding process by handling repetitive tasks and reducing manual coding. Generative AI can also translate code from one language to another, streamlining code conversion or modernization projects, such as updating legacy applications by translating COBOL to Java.

Computer vision

Computer vision is a field of artificial intelligence (AI) that includes image classification, object detection and semantic segmentation. It uses machine learning and neural networks to teach computers and learning systems to derive meaningful information from digital images, videos and other visual inputs—and to make recommendations or take actions when the system sees defects or issues. If AI enables computers to think, computer vision enables them to see, observe and understand.

Because a computer vision system is often trained to inspect products or watch production assets, it usually can analyze thousands of products or processes per minute, noticing imperceptible defects or issues. Computer vision is used in industries that range from energy and utilities to manufacturing and automotive.

Computer vision needs lots of data, and then it runs analyses of that data over and over until it discerns and ultimately recognizes images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.

Computer vision uses algorithmic models to enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will “look” at the data and teach itself to tell one image from another. Algorithms enable the machine to learn by itself, rather than with someone programming it to recognize an image.

Computer vision enables systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, to take action. This ability to provide recommendations distinguishes it from simple image recognition tasks. Some common applications of computer vision today can be seen in:

Automotive: While the age of driverless cars hasn’t entirely arrived, the underlying technology has started to make its way into automobiles, improving driver and passenger safety through features such as lane line detection.

Healthcare: Computer vision has been incorporated into radiology technology, enabling doctors to better identify cancerous tumors in healthy anatomy.

Marketing: Social media platforms provide suggestions on who might be in a photograph that has been posted on a profile, making it easier to tag friends in photo albums.

Retail: Visual search has been incorporated into some e-commerce platforms, enabling brands to recommend items that would complement an existing wardrobe.

Customer care

AI is helping businesses to better understand and cater to increasing consumer demands. With the rise of highly personalized online shopping, direct-to-consumer models, and delivery services, generative AI can help further unlock a host of benefits that can improve customer care, talent transformation and the performance of applications.

AI empowers businesses to adopt a customer-centric approach by harnessing valuable insights from customer feedback and buying habits. This data-driven approach can help improve product design and packaging and can help drive high customer satisfaction and increased sales.

Generative AI can also serve as a cognitive assistant for customer care, providing contextual guidance based on conversation history, sentiment analysis and call center transcripts. Also, generative AI can enable personalized shopping experiences, foster customer loyalty and provide a competitive advantage.

Digital labor

Organizations can augment their workforce by building and deploying robotic process automation (RPA) and digital labor to collaborate with humans to increase productivity, or assist whenever backup is needed. For example, this can help developers speed the updating of legacy software.

Digital labor uses foundation models to automate and improve the productivity of knowledge workers by enabling self-service automation in a fast and reliable way—without technical barriers. To automate task performance or calling APIs, an enterprise-grade LLM-based slot filling model can identify information in a conversation and gather all the information required for completing an action or calling an API without much manual effort.

Instead of having technical experts record and encode repetitive action flows for knowledge workers, digital labor automations built with a foundation of model-powered conversational instructions and demonstrations can be used by the knowledge worker for self-service automation. For example, to speed app creation, no-code digital apprentices can help end-users, who lack programming expertise, by effectively teaching, supervising and validating code.

Generative AI

Generative AI (also called gen AI) is a category of AI that autonomously creates text, images, video, data or other content in response to a user’s prompt or request.

Generative AI relies on deep learning models that can learn from patterns in existing content and generate new, similar content based on that training. It has applications in many fields—including customer service, marketing, software development and research—and offers enormous potential to streamline enterprise workflows through fast, automated content creation and augmentation.

Generative AI excels at handling diverse data sources such as emails, images, videos, audio files and social media content. This unstructured data forms the backbone for creating models and the ongoing training of generative AI, so it can stay effective over time. Using this unstructured data can enhance customer service through chatbots and facilitate more effective email routing. In practice, this might mean guiding users to appropriate resources, whether that’s connecting them with the right agent or directing them to user guides and FAQs.

Despite its much-discussed limitations and risks, many businesses are forging ahead, cautiously exploring how their organizations can harness generative AI to improve their internal workflows, and enhance their products and services. This is the new frontier: How to make the workplace more efficient without creating legal or ethical issues.

Natural language processing and speech recognition

NLP combines computational linguistics—rule-based modeling of human language—with statistical and machine learning models to enable computers and digital devices to recognize, understand and generate text and speech. NLP powers applications and devices that can translate text from one language to another, respond to typed or spoken commands, recognize or authenticate users based on voice. It helps summarize large volumes of text, assess the intent or sentiment of text or speech and generate text or graphics or other content on demand.

A subset of NLP is statistical NLP, which combines computer algorithms with machine learning and deep learning models. This approach helps to automatically extract, classify and label elements of text and voice data and then assign a statistical likelihood to each possible meaning of those elements. Today, deep learning models and learning techniques based on RNNs enable NLP systems that “learn” as they work and extract ever more accurate meaning from huge volumes of raw, unstructured and unlabeled text and voice datasets.

Speech recognition—also known as automatic speech recognition (ASR), computer speech recognition or speech-to-text—is a capability that enables a program to process human speech into a written format.

While speech recognition is commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user’s voice.

Industry applications

Real-world deep learning applications are all around us, and so well integrated into products and services that users are unaware of the complex data processing that is taking place in the background. Some of these examples include:

Customer service deep learning

Many organizations incorporate deep learning technology into their customer service processes. Chatbots are often used in various applications, services and customer service portals. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more sophisticated chatbot solutions attempt to determine, through learning, if there are multiple responses to ambiguous questions in real time. Based on the responses it receives, the chatbot then tries to answer these questions directly or routes the conversation to a human user.

Virtual assistants such as Apple's Siri, Amazon Alexa or Google Assistant extend the idea of a chatbot by enabling speech recognition functionality. This creates a new method to engage users in a personalized way.

Financial services analytics

Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

Healthcare record-keeping

The healthcare industry has benefited greatly from deep learning capabilities ever since the digitization of hospital records and images. Image recognition applications can support medical imaging specialists and radiologists, helping them analyze and assess more images in less time.

Law enforcement uses deep learning

Deep learning algorithms can analyze and learn from transactional data to identify dangerous patterns that indicate possible fraudulent or criminal activity. Speech recognition, computer vision and other deep learning applications can improve the efficiency and effectiveness of investigative analysis by extracting patterns and evidence from sound and video recordings, images and documents. This capability helps law enforcement analyze large amounts of data more quickly and accurately.

*Source: https://www.ibm.com/think/topics/deep-learning#:~:text=Deep%20learning%20is%20a%20subset,applications%20in%20our%20lives%20today.*