

Deep learning is a specialized part of machine learning that trains **Artificial Neural Networks (ANNs)** or **DNN** (deep neural network ) to solve complex tasks, like recognizing images. Here's how it works:

1. **ANNs with Multiple Layers**: Deep learning involves neural networks that have many layers, often called "deep" because of how many layers they have. These layers allow the network to automatically learn and understand complex data.
2. **Working with Raw Data**: ANNs can handle raw data, such as images, directly. The network doesn’t need to be told specifically what features to look for. Instead, it learns on its own.
3. **Example – Handwritten Digit Recognition**:
   * You have a set of handwritten images of digits from 0 to 9 (this dataset is known as MNIST). Each person writes the digits differently, so the challenge is for the ANN to identify which digit is shown in an image.
   * The network processes the pixel data (each image is made up of tiny pixels) and learns patterns, like edges or curves, that help define the shape of each digit.
   * Based on these patterns, the ANN can predict the outcome. For example, it can recognize that the digit in the image is a 4.

In essence, deep learning with ANNs learns **automatically from raw data**, recognizes patterns, and uses those patterns to make predictions — in this case, identifying what number is in a handwritten image!

**Difference b/w ML vs DL by IBM**

The chief difference between deep learning and machine learning is the structure of the underlying neural network architecture. “Nondeep,” [traditional machine learning](https://www.ibm.com/blog/machine-learning-types/) 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](https://www.ibm.com/consulting/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.

For more: <https://www.ibm.com/topics/deep-learning>

**Transformation from ML to DL and then Gen AI by Sir zia khan**

Indeed, Traditional Machine Learning (ML) or **Discriminative AI** also incorporated neural networks, which laid the foundation for more complex models like those used in Generative AI today. However, traditional neural networks in ML were simpler in architecture and scope compared to the deep neural networks driving modern generative models.

**Evolution into Deep Learning**

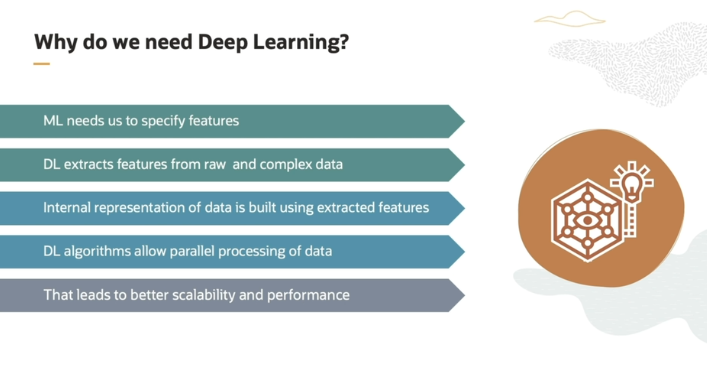
The limitations of early neural networks led to the development of **Deep Learning**, a subset of ML that focuses on deeper architectures with many hidden layers, enabling the modeling of more abstract and complex features.

**From Traditional Neural Networks to Generative Models**

The progression from traditional neural networks to generative models can be understood as an evolution in both the complexity of architectures and the types of tasks they were applied to:

* **Traditional ML Tasks (Discriminative Models):** Focused on tasks like classification and regression where the goal was to learn the boundary between different classes or to predict a continuous value based on input features.
* **Generative Models:** These models, like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), go beyond classification to learn the underlying distribution of the data itself. This allows them to generate new data instances that are similar to the original data.

For more: <https://github.com/panaversity/learn-applied-generative-ai-fundamentals/tree/main/19_genai_foundations/01_descriminative_ai#1-neural-networks-in-traditional-ml>



 **Machine Learning vs Deep Learning**:

* In **traditional machine learning**, when training an algorithm, we have to **manually select features** (important data points) that we think will help the model make predictions.
* In **deep learning**, the model **automatically finds and learns the important features** directly from the data. We don't need to specify them manually.

 **Internal Representation**:

* Deep learning builds its own **internal understanding** of the data. It looks at various patterns and combinations within the data to make predictions. This process is very complex and would be difficult for a human to do by hand.

 **Parallel Computations**:

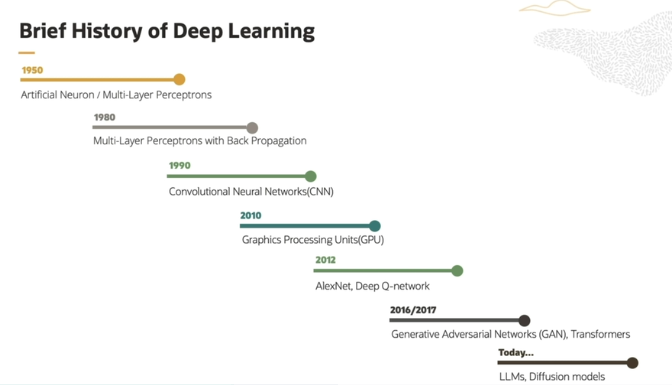
* Deep learning algorithms are good at handling large datasets quickly because they use **parallel processing**. This means they split the data into smaller parts (called batches) and process them at the same time.
* By doing this, deep learning can analyze a large amount of data in a **short amount of time**.

 **Scalability and Performance**:

* Since deep learning can automatically learn features and process data quickly, it’s **scalable** (it can handle larger amounts of data) and has **high performance** when working with **complex data** that is difficult to break down manually.

 **Conclusion**:

* In summary, deep learning is especially useful when the data is too complicated to describe with simple features. It complements traditional machine learning by handling more **complex** data and doing it **faster** thanks to its ability to automatically learn and process information.



**1. 1950s: Birth of Artificial Neurons**

* In the 1950s, researchers introduced the idea of **artificial neurons** as a way to simulate the workings of the human brain. These artificial neurons were the first building blocks of what we now call **neural networks**.
* The most basic model was the **perceptron**, which could take inputs and decide whether to activate or not, based on a simple rule. This was a very early concept of how machines could "learn" from data.

**2. 1980s: Introduction of Backpropagation**

* One of the biggest challenges in neural networks was figuring out how to train them effectively. In the 1980s, the idea of **backpropagation** was introduced.
  + Backpropagation is a method that allows a neural network to **learn** by adjusting its internal weights based on the errors it makes during training. It essentially "teaches" the network how to improve its predictions.
  + This method made it possible to train multi-layer networks, which became known as **multi-layer perceptrons**.

**3. 1990s: Convolutional Neural Networks (CNNs)**

* In the 1990s, **convolutional neural networks (CNNs)** were introduced. CNNs became popular for **image analysis** tasks.
  + CNNs are designed to automatically detect patterns like edges, shapes, and objects in images.
  + This innovation was important because it allowed neural networks to process visual data more effectively, paving the way for modern computer vision tasks.

**4. 2000s: Rise of GPUs**

* In the early 2000s, **Graphics Processing Units (GPUs)** were introduced for training neural networks.
  + **GPUs** could perform many operations in parallel, which made them ideal for speeding up deep learning computations.
  + As **GPUs became cheaper and more available**, researchers started using them to train larger and deeper neural networks, which allowed deep learning to scale and tackle more complex tasks.

**5. 2010s: Widespread Adoption**

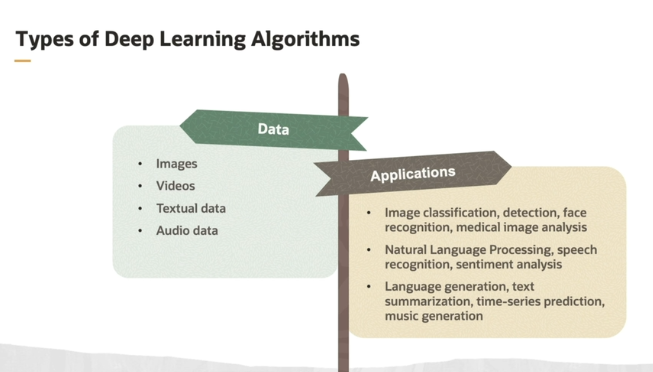
* **2010 onwards**, deep learning started becoming mainstream thanks to the availability of **cheap GPUs** and the development of better algorithms.
  + Deep learning was adopted in areas like **computer vision, natural language processing (NLP), speech recognition**, and **text translation**.
  + These applications are what you see today in technologies like **face recognition, voice assistants (Alexa, Siri)**, and **machine translation (Google Translate)**.

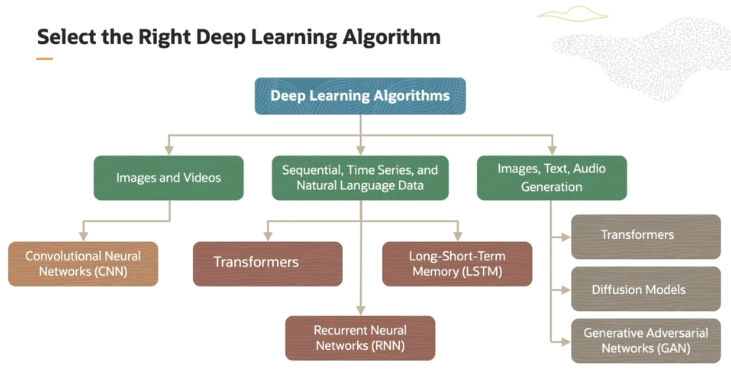
**6. 2012: Breakthrough with AlexNet and Deep Q-Network**

* In 2012, the **AlexNet** network achieved a breakthrough in **image classification** by winning the ImageNet competition, making deep learning much more popular.
* Around the same time, the **Deep Q-Network (DQN)** was developed for **reinforcement learning**, which allowed AI to learn how to play video games at superhuman levels. This was a major milestone in teaching machines how to learn from experience.

**7. 2016 Onward: Generative Deep Learning**

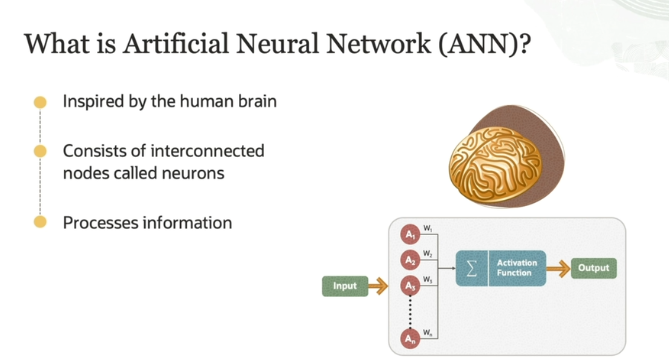
* From **2016**, deep learning started to be used for **generative tasks**, meaning AI could create new content, like images, text, or even music.
  + This is where models like **GANs (Generative Adversarial Networks)** and **large language models** started to gain attention. These models can generate realistic images, simulate human conversations, and more.





Now mainly here its telling that we have to identify **deep learning algo** on the basis of Data that which type of data we are going to process and on the basis of task/application that which type of work we are going to be performed by algo.

Selecting the right deep learning algorithm based on the data and application is important. For image, task like image classification, object detection, image segmentation, or facial recognition, CNN is a suitable architecture. For text, we have a choice of the latest transformers or LSTM or even RNN. For a generative task like text summarization, question answering, transformers is a good choice. For generating images, text-to-image generation, transformers, GANs, or diffusion models are available choice.



An Artificial Neural Network (ANN) works similarly to how the human brain processes information. Here's a simplified breakdown:

1. **Neurons**: Just like the brain, ANN consists of nodes called neurons. These are the building blocks that receive, process, and pass on information.
2. **Weights**: Each connection between neurons has a weight. Weights are important because they control how much influence one neuron has on the next one. If a connection has a higher weight, that input is considered more important.
3. **Processing Inputs**: Neurons take in inputs (like numbers representing data), multiply these inputs by their respective weights, and add them up.
4. **Threshold**: The neuron then checks if the sum of these weighted inputs is large enough to pass a certain threshold. If it is, the neuron "fires," meaning it activates and passes its output to the next layer.
5. **Layers**: The output of one neuron or layer becomes the input for the next. This process continues through several layers in the network, gradually transforming raw input data into more meaningful information to make predictions or decisions.

In short, ANN is a structure where neurons work together, adjusting their weights based on the input data, to "learn" patterns and make decisions!

**ANN by IBM**

Link: <https://www.ibm.com/topics/neural-networks>

A neural network is a machine learning model that works like a simplified version of the human brain. Here's an easy explanation:

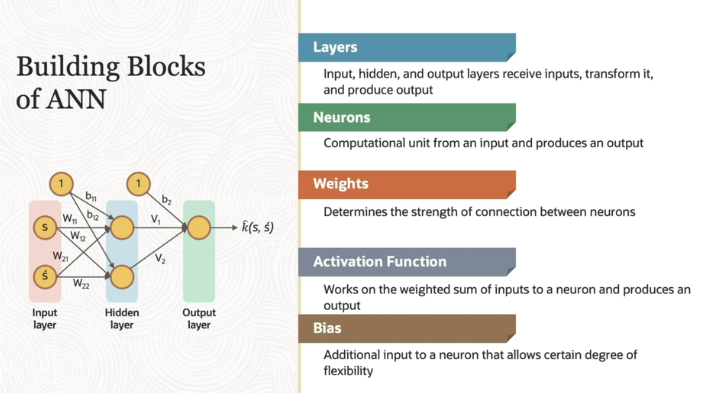
1. **Neural Network Structure**: It consists of layers of "neurons" (also called nodes) that are connected to each other. These layers include:
   * **Input Layer**: Takes in the raw data (like an image or text).
   * **Hidden Layers**: Perform calculations and transformations on the data to uncover patterns.
   * **Output Layer**: Provides the final decision or prediction (like recognizing an object in a picture).
2. **Connections and Weights**: Each neuron in a layer is connected to neurons in the next layer. Every connection has a weight, which shows how important that connection is.
3. **Thresholds and Activation**: Each neuron sums up the inputs it gets from the previous layer, multiplies them by their weights, and checks if the result passes a certain threshold. If it does, the neuron is "activated" and passes data to the next layer. If not, it remains inactive.
4. **Learning from Data**: Neural networks need **training data** to learn how to make decisions. During training, they adjust their weights and thresholds to improve their predictions. Over time, they become more accurate.
5. **Speed and Efficiency**: Once trained, neural networks can process data quickly. For example, they can recognize speech or images in seconds, which would take humans much longer to do manually.
6. **Applications**: Neural networks are used in many AI applications like Google’s search algorithm, speech recognition (like Siri or Google Assistant), and image recognition.

In summary, neural networks are like a digital brain that learns from data, identifies patterns, and makes decisions or predictions based on that learning. They are crucial in modern AI, especially in tasks where fast and accurate data processing is important.

Yt link: <https://youtu.be/rEDzUT3ymw4?si=JSQbxSAXEoXGsPVw>

Best resourse: <https://youtu.be/bfmFfD2RIcg?si=leDwaNdoLINcRuqT>

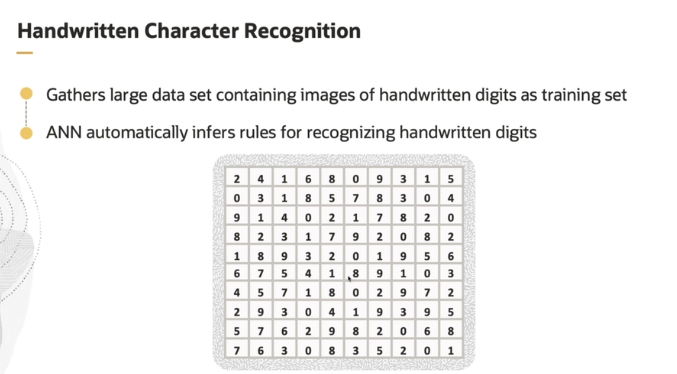
Sir zia link: <https://github.com/panaversity/learn-applied-generative-ai-fundamentals/tree/main/19_genai_foundations/00_neural_networks>



The image and explanation describe the **building blocks of an Artificial Neural Network (ANN)**, which consists of several key elements that work together to make predictions or decisions. Let's break it down:

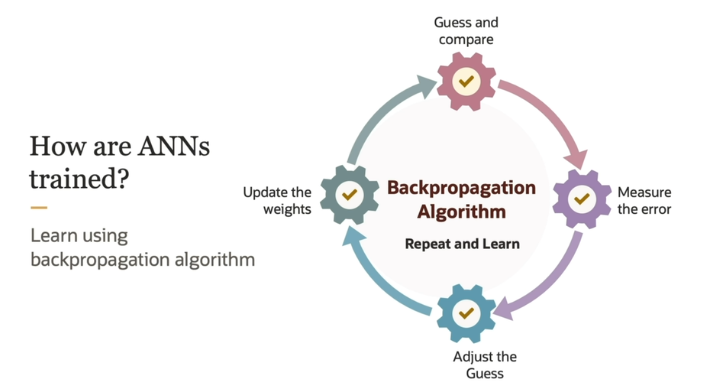
1. **Layers**:
   * **Input Layer**: This layer receives raw data (like numbers, pixels from images, etc.). It acts as the starting point of the network.
   * **Hidden Layer(s)**: These layers are optional but often multiple hidden layers are used in deep learning. They process the inputs from the previous layer and pass transformed data to the next layer.
   * **Output Layer**: The final layer that provides the result or prediction based on the data processed through the network.
2. **Neurons**:
   * These are the computational units or "nodes" that do the actual processing in a layer. Each neuron takes an input, processes it, and generates an output that is passed to the next neuron in the subsequent layer.
3. **Weights**:
   * **Weights** represent the strength of the connection between two neurons. Each connection between neurons has a weight associated with it, which determines how much influence the input from one neuron has on the next neuron.
   * For example, if a connection has a high weight, it means that input has a strong influence on the next neuron’s output.
4. **Activation Function**:
   * After calculating the weighted sum of inputs, the **activation function** decides if the neuron should be "activated" (produce an output) or not. It helps introduce non-linearity, allowing the network to learn complex patterns.
   * The activation function works by taking the weighted sum and applying a function to determine if the neuron "fires" (passes the data to the next layer) or not.
5. **Bias**:
   * **Bias** is an additional input added to the neuron that allows it to be more flexible in decision-making. It gives the network the ability to better adjust its outputs, even when all inputs are 0.
   * Think of bias as a constant value added to shift the output, helping the network to make better predictions by adding flexibility.

So in summary, the input data moves through the layers of neurons, where each connection has a weight. The activation function helps the neurons decide when to activate, and the bias adds a bit of flexibility to the network. All of these building blocks work together to make predictions, such as recognizing an image or classifying data.



A screen shot of a computer

Description automatically generated



Artificial Neural Networks (ANNs) are trained through a process called backpropagation. Here's how it works in a simple way:

1. **Start with an Input**: Imagine you show the ANN an image, say of the digit "2." The network processes this image and produces an output. Ideally, the output neuron for "2" should fire, meaning it recognizes the digit correctly.
2. **Mistakes Happen**: Sometimes, the network gets it wrong. Let's say instead of firing the "2" neuron, the network fires the neuron for "6." This is an error.
3. **Fixing the Error (Backpropagation)**: To correct this error, we adjust the "weights" (which are like the strength of connections between neurons). The backpropagation algorithm calculates how much each connection contributed to the mistake. It then tweaks the weights to reduce the error.
4. **Repeat and Improve**: By showing the network thousands of images and using backpropagation to adjust the weights after each error, the ANN gradually learns to predict the correct digit for most images.

This process of adjusting weights through backpropagation is what we call **model training**. The more images the ANN sees, the better it gets at making correct predictions!