



Module 1

Deep Networks

The differences between the three types of Deep Neural Networks and deep learning basics. Such deep neural networks (DNNs) have recently demonstrated impressive performance in complex machine learning tasks such as image classification, image processing, or text and speech recognition.

In particular, we will cover the following neural network types:

- Multi-Layer Perceptrons (MLP)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNN)

About us: Viso Suite allows machine learning teams to take control of the entire project lifecycle. By eliminating the need to purchase and manage point solutions, Viso Suite presents teams with a truly end-to-end computer vision infrastructure. To learn more, get a personalized demo from the Viso team.

What Is a Deep Neural Network (DNN) ?

Machine learning techniques have been widely applied in various areas such as pattern recognition, natural language processing, and computational learning. During the past decades, machine learning has had an enormous influence on our daily lives with examples including efficient web search, self-driving systems, computer vision, and optical character recognition (OCR).

Especially, deep learning models have become a powerful tool for machine learning and artificial intelligence. A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. Note that the terms ANN vs. DNN are often incorrectly confused or used interchangeably.

The success of deep learning neural networks has led to breakthroughs such as reducing word error rates in speech recognition by 30% over traditional approaches (the biggest gain in 20 years) or drastically cutting the error rate in an image recognition competition since 2011 (from 26% to 3.5% while humans achieve 5%).

Concept of Deep Neural Networks



Module 1

Deep neural network models were originally inspired by neurobiology. On a high level, a biological neuron receives multiple signals through the synapses contacting its dendrites and sending a single stream of action potentials out through its axon. The complexity of multiple inputs is reduced by categorizing its input patterns. Inspired by this intuition, artificial neural network models are composed of *units* that combine multiple inputs and produce a single output.

Deep Learning Layers explained

Neural networks target brain-like functionality and are based on a simple artificial neuron: a nonlinear function (such as $\max(0, \text{value})$) of a weighted sum of the inputs. These pseudo neurons are collected into layers, and the outputs of one layer become the inputs of the next in the sequence.

What makes a Neural Network “Deep”?

Deep neural networks employ deep architectures in neural networks. “Deep” refers to functions with higher complexity in the number of layers and units in a single layer. The ability to manage large datasets in the cloud made it possible to build more accurate models by using additional and larger layers to capture higher levels of patterns.

The two key phases of neural networks are called *training* (or learning) and *inference* (or prediction), and they refer to the development phase versus production or application. When creating the architecture of deep network systems, the developer chooses the number of layers and the type of neural network, and training data determines the weights.

3 Types of Deep Neural Networks (DNN)

Three following types of deep neural networks are popular today:

1. Multi-Layer Perceptrons (MLP)
2. Convolutional Neural Networks (CNN)
3. Recurrent Neural Networks (RNN)

Multilayer Perceptrons (MLPs)

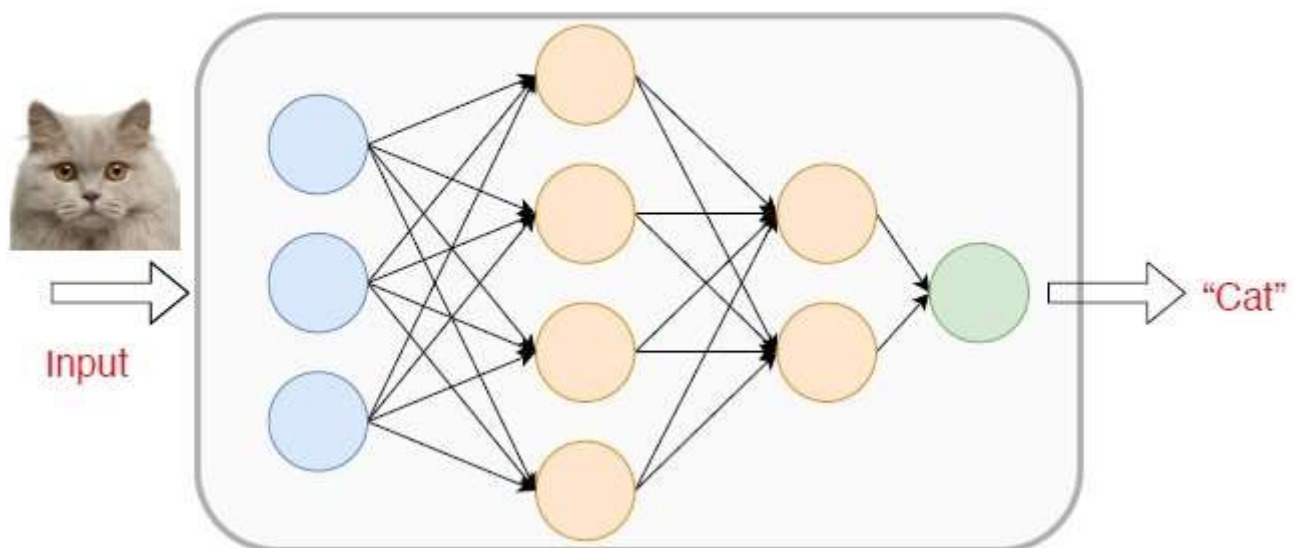


Module 1

A multilayer perceptron (MLP) is a class of a feedforward artificial neural network (ANN). MLP models are the most basic deep neural network, which is composed of a series of fully connected layers. Today, MLP machine learning methods can be used to overcome the requirement of high computing power required by modern deep learning architectures.

Each new layer is a set of nonlinear functions of a weighted sum of all outputs (fully connected) from the prior one.

Multilayer Perceptrons



Concept of Multilayer Perceptrons (MLP)

MLP Applications

MLPs are ideal for tasks involving structured data as they can handle input data with fixed dimensions and independent features, allowing them to learn complex patterns and relationships within the structured data. Additionally, their scalability and ease of implementation make them applicable to structured data tasks.

Some real-life applications of MLPs include:



Module 1

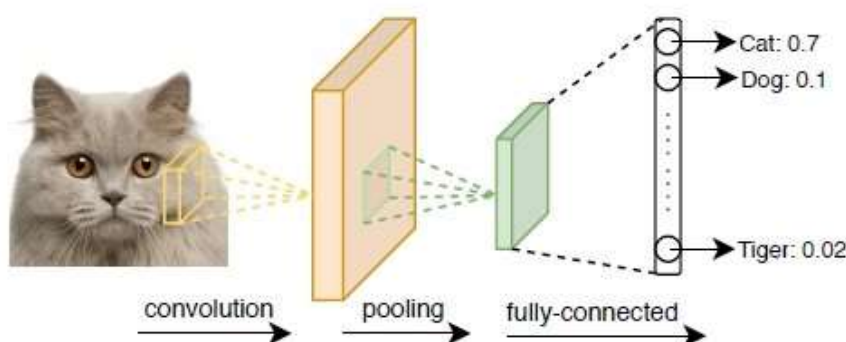
- Credit Scoring. Analysis of factors compiled as structured data such as credit history, income, and debt level to assess creditworthiness.
- Fraud Detection. Analysis of transactions as tabular data to detect potentially fraudulent activity, such as unauthorized access, identity theft, or unusual spending patterns.
- Customer Churn Prediction. Analysis of customer behavior, purchase history, and engagement metrics identifying customers at risk of leaving.

Convolutional Neural Network (CNN)

A convolutional neural network (CNN, or ConvNet) is another class of deep neural networks. We most commonly find CNNs in computer vision. Given a series of images or videos from the real world, with the utilization of CNN, the AI system learns automatic feature extraction of these inputs to complete a specific task, e.g., image classification, face authentication, and image semantic segmentation.

Different from fully connected layers in MLPs, in CNN models, one or multiple convolution layers extract features from input by executing convolution operations. Each layer is a set of nonlinear functions of weighted sums at different coordinates of spatially nearby subsets of outputs from the prior layer, which allows for the reuse of the weights.

Convolutional Neural Network



Concept of a Convolution Neural Network (CNN)



Module 1

Applying various convolutional filters, CNN machine learning models can capture the high-level representation of the input data, making CNN techniques widely popular in computer vision tasks.

Examples of Convolutional Neural Network (CNN) Models

Convolutional neural network example applications include image classification (e.g., AlexNet, VGG network, ResNet, MobileNet) and object detection (e.g., Fast R-CNN, Mask R-CNN, YOLO, SSD).

- AlexNet. For image classification, as the first CNN neural network to win the ImageNet Challenge in 2012, AlexNet consists of five convolution layers and three fully connected layers. Thus, AlexNet requires 61 million weights and 724 million MACs (multiply-add computation) to classify the image with a size of 227×227 .
- VGG-16. To achieve higher accuracy, VGG-16 is trained to a deeper structure of 16 layers consisting of 13 convolution layers and three fully connected layers. This requires 138 million weights and 15.5G MACs to classify the image with a size of 224×224 .
- GoogleNet. To improve accuracy while reducing the computation of DNN inference, GoogleNet introduces an inception module composed of different-sized filters. As a result, GoogleNet achieves a better accuracy performance than VGG-16 while only requiring seven million weights and 1.43G MACs to process the image with the same size.
- ResNet. ResNet, the state-of-the-art effort, uses the “shortcut” structure to reach a human-level accuracy with a top-5 error rate below 5%. In addition, the “shortcut” module can solve the gradient vanishing problem during the training of the model, making it possible to train a DNN model with a deeper structure.

The performance of popular CNNs applied for AI vision tasks gradually increased over the years. At this time, CNNs have surpassed human vision (5% error rate in the chart below).

Performance of current popular Deep Neural Networks on ImageNet. Humans achieve an error rate of 5%. – [Source](#)

CNN Applications

CNNs are useful for tasks involving spatial or hierarchical structure of input data, such as visual, audio, or time-series data. These models are very useful for image



Module 1

classification, object detection, and image segmentation, where spatial relationships between pixels or features matter. CNNs automatically learn hierarchical features from raw input data, making them prime for tasks requiring feature extraction from complex inputs.

- **Art Restoration and Preservation.** By analyzing high-resolution images of paintings or sculptures, CNNs detect missing or worn-down parts of pieces, thus aiding in reconstructing deteriorated parts of artwork.
- **Wildlife Conservation.** Conservationists and environmentalists can use CNNs to analyze camera trap images from remote locations. With images from remote locations, it omits the need for humans to make the difficult journey into the wilderness. The images from these locations can track animals, estimate population sizes, and detect poaching activities.
- **Fashion Design and Trends Prediction.** CNNs can be trained on datasets of fashion show imagery, social media, and e-commerce sites. With this info, CNNs can then identify emerging fashion trends, predict styles, and provide personalized fashion recommendations.

Recurrent Neural Network (RNNs)

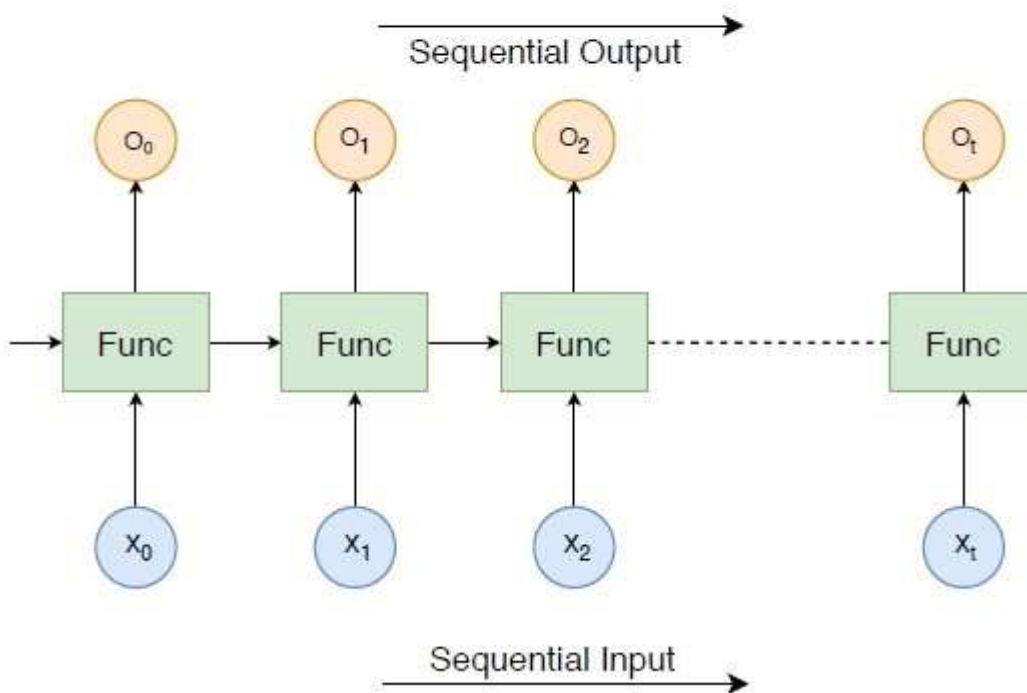
A recurrent neural network (RNN) is another class of artificial neural networks that use sequential data feeding. RNNs have been developed to address the time-series problem of sequential input data.

The input of RNN consists of the current input and the previous samples. Therefore, the connections between nodes form a directed graph along a temporal sequence. Furthermore, each neuron in an RNN owns an internal memory that keeps the information of the computation from the previous samples.



Module 1

Recurrent Neural Network



Concept of a

Recurrent Neural Network (RNN)

RNN models are widely used in Natural Language Processing (NLP) due to the superiority of processing the data with an input length that is not fixed. The task of the AI here is to build a system that can comprehend natural language spoken by humans. For example, natural language modeling, word embedding, and machine translation.

In RNNs, each subsequent layer is a collection of nonlinear functions of weighted sums of outputs and the previous state. Thus, the basic unit of RNN is a “cell”, consisting of layers and series of cells enabling the sequential processing of recurrent neural network models.

Examples of Recurrent Neural Network (RNN) Models

1. Long Short-Term Memory (LSTM). LSTM models address the vanishing gradient problem. The incorporation of specialized memory cells and gating mechanisms makes the learning of long-term dependencies in sequential data possible.



Module 1

2. Gated Recurrent Unit (GRU). Similar to LSTMs, GRU networks capture long-range dependencies in sequential data. GRU architecture is simpler when compared to LSTMs, with fewer parameters, thus, making them more computationally efficient in some cases.
3. Bidirectional RNNs. Process input sequences both forward and backward allowing them to capture dependencies from past and future contexts. In turn, making them useful in tasks such as speech recognition and machine translation.

RNN Applications

RNNs are useful for sequential data or data with temporal dependencies, like time-series data, text, or speech. With this in mind, RNNs tasks take into consideration input data order such as language modeling, sentiment analysis, and machine translation. RNNs can capture temporal dynamics and long-range dependencies in sequential data. Thus, making them valuable for tasks involving sequential prediction or generation.

- Music Composition and Generation. RNNs can generate new music by mimicking different genres or composers through pattern and structure analysis of existing music data.
- Personalized Storytelling and Interactive Fiction. By analyzing user inputs and interactions, RNNs can create storylines, characters, and plot twists adapting and evolving based on user decisions.
- Predictive Text-based Adventure Games. By analyzing actions and dialogue choices, RNNs can generate storylines, character interactions, and narratives based on players' decisions.