3 Applications of DL in the real world

1. **Self-Driving Cars:**
   A million sets of data are fed to a system to build a model, to train the machines to learn, and then test the results in a safe environment. The Uber Artificial Intelligence Labs at Pittsburg is not only working on making driverless cars humdrum but also integrating several smart features such as food delivery options with the use of driverless cars. A regular cycle of testing and implementation typical to deep learning algorithms is ensuring safe driving with more and more exposure to millions of scenarios.

2. **Virtual Assistants:**
   The most popular application of deep learning is virtual assistants ranging from Alexa to Siri to Google Assistant. Each interaction with these assistants provides them with an opportunity to learn more about your voice and accent, thereby providing you a secondary human interaction experience. Virtual assistants use deep learning to know more about their subjects ranging from your dine-out preferences to your most visited spots or your favourite songs. With deep learning applications such as text generation and document summarizations, virtual assistants can assist you in creating or sending appropriate email copy as well.

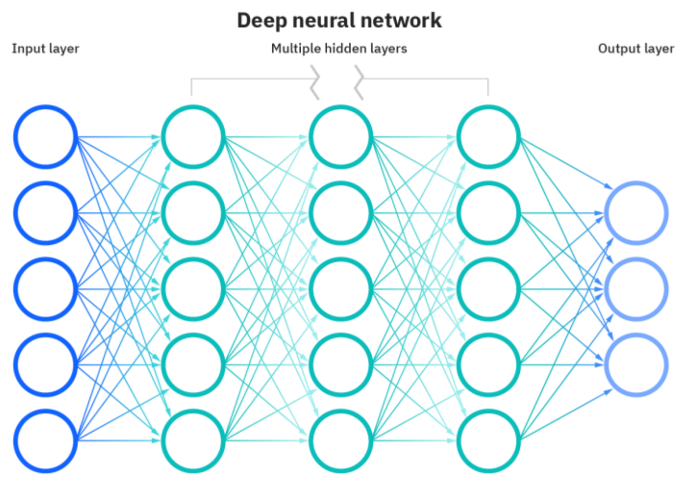
3. **Detecting Developmental Delay in Children:**
   Speech disorders, autism, and developmental disorders can deny a good quality of life to children suffering from any of these problems. An early diagnosis and treatment can have a wonderful effect on the physical, mental, and emotional health of differently-abled children. Hence, one of the noblest applications of deep learning is in the early detection and course-correction of these problems associated with infants and children. They use residual analysis that identifies the correlation between age, gender, and acoustic features of their speech to limit false positives. Autism is often detected by combining it with cofactors such as low birth weight, physical activity, body mass index, learning disabilities, etc.

What is Neural Network?

The computing systems inspired by biological neural networks to perform different tasks with a huge amount of data involved is called artificial neural networks or ANN. Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Structure of Neural Network?

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, 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.



Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. **One of the most well-known neural networks is Google's search algorithm.**

How do neural networks work?

Think of each individual node as its own linear regression model, composed of input data, weights, a bias (or threshold), and an output. Once an input layer is determined, weights are assigned. These weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs. All inputs are then multiplied by their respective weights and then summed. Afterward, the output is passed through an activation function, which determines the output. If that output exceeds a given threshold, it “fires” (or activates) the node, passing data to the next layer in the network. This results in the output of one node becoming in the input of the next node. This process of passing data from one layer to the next layer defines this neural network as a feedforward network.

There are three methods or learning paradigms to teach a neural network.

-Supervised Learning
-Reinforcement Learning
-Unsupervised Learning

Convolutional Neural Network (CNN) the DL Algorithm:

A convolutional neural network is a specific kind of neural network with multiple layers. It processes data that has a grid-like arrangement then extracts important features. One huge advantage of using CNNs is that you don't need to do a lot of pre-processing on images. CNNs can learn what characteristics in the filters are the most important. That saves a lot of time and trial and error work since we don't need as many parameters. The convolutional neural network algorithm's main purpose is to get data into forms that are easier to process without losing the features that are important for figuring out what the data represents. This also makes them great candidates for handling huge datasets. A convolution is used instead of matrix multiplication in at least one layer of the CNN.

How CNN works:

Convolutional neural networks are based on neuroscience findings. They are made of layers of artificial neurons called nodes. These nodes are functions that calculate the weighted sum of the inputs and return an activation map. This is the convolution part of the neural network. Each node in a layer is defined by its weight values. When you give a layer some data, like an image, it takes the pixel values and picks out some of the visual features. When you're working with data in a CNN, each layer returns activation maps. These maps point out important features in the data set. If you gave the CNN an image, it'll point out features based on pixel values, like colors, and give you an activation function.

Applications of CNN:

-Recognize images with little pre-processing
-Recognize different hand-writing
-Computer vision applications
-Used in banking to read digits on checks
-Used in postal services to read zip codes on an envelope

Example of CNN with Code:

Identify some handwritten numbers using the MNIST data set.
The first thing we do is define the CNN model. Next we separate our training and test data. Lastly, we use the training data to train the model and test that model using the test data.

CODE:

```
from keras import layers
from keras import models
from keras.datasets import mnist
from keras.utils import to_categorical

# Define the CNN model
model = models.Sequential()

model.add(layers.Conv2D(32, (5,5), activation='relu', input_shape=(28, 28,1)))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())
model.add(layers.Dense(10, activation='softmax'))

model.summary()

# Split the data into training and test sets
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

train_images = train_images.reshape((60000, 28, 28, 1))
train_images = train_images.astype('float32') / 255
```

```
test_images = test_images.reshape((10000, 28, 28, 1))
test_images = test_images.astype('float32') / 255

train_labels = to_categorical(train_labels)
test_labels = to_categorical(test_labels)

# Use the training data to train the model
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])

model.fit(train_images, train_labels,
          batch_size=100,
          epochs=5,
          verbose=1)

# Test the model's accuracy with the test data
test_loss, test_acc = model.evaluate(test_images, test_labels)

print('Test accuracy:', test_acc)
```