Introduction to Deep Learning

AcadView

June 8, 2018

1 Overview

Deep learning is an aspect of artificial intelligence (AI) that is concerned with emulating the learning approach that human beings use to gain certain types of knowledge.

While traditional machine learning algorithms are linear, deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction. To understand deep learning, imagine a toddler whose first word is dog. The toddler learns what a dog is (and is not) by pointing to objects and saying the word dog. The parent says, ”Yes, that is a dog,” or, ”No, that is not a dog.” As the toddler continues to point to objects, he becomes more aware of the features that all dogs possess. What the toddler does, without knowing it, is clarify a complex abstraction (the concept of dog) by building a hierarchy in which each level of abstraction is created with knowledge that was gained from the preceding layer of the hierarchy.

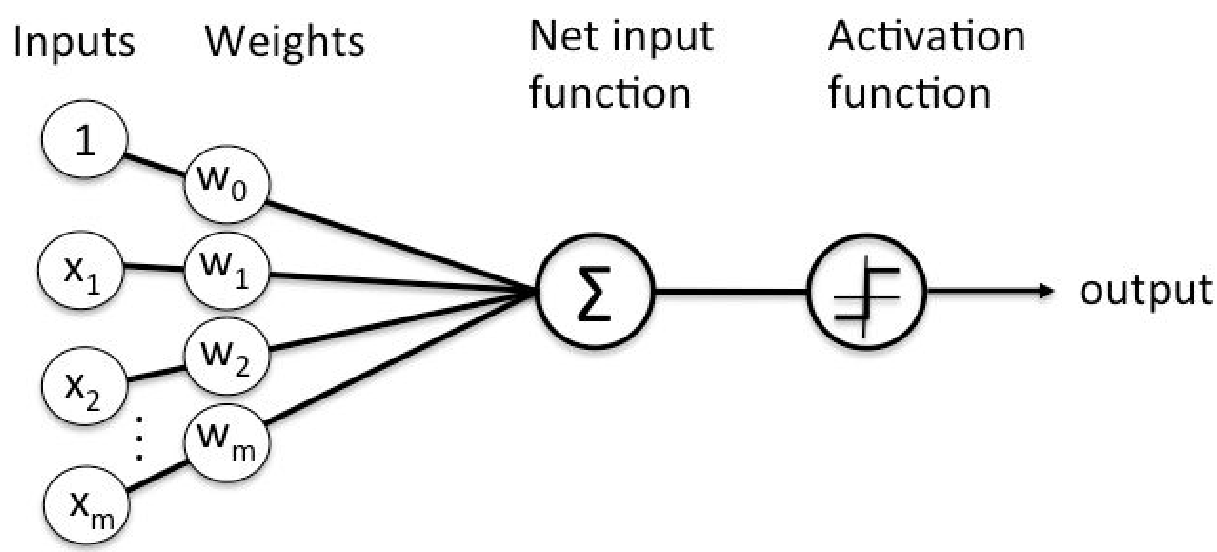
Use cases today for deep learning include all types of big data analytics applications, especially those focused on natural language processing (NLP), language translation, medical diagnosis, stock market trading signals, network security and image identification.

2 Deep Learning and Neural Networks

Deep learning is the name we use for ”stacked neural networks”; that is, networks composed of several layers.

The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs for the task the algorithm is trying to learn. (For example, which input is most helpful is classifying data without error?) These input-weight products are summed and the sum is passed through a nodes so-called activation function, to determine whether and to what extent that signal progresses further through the network to affect the ultimate outcome, say, an act of classification.

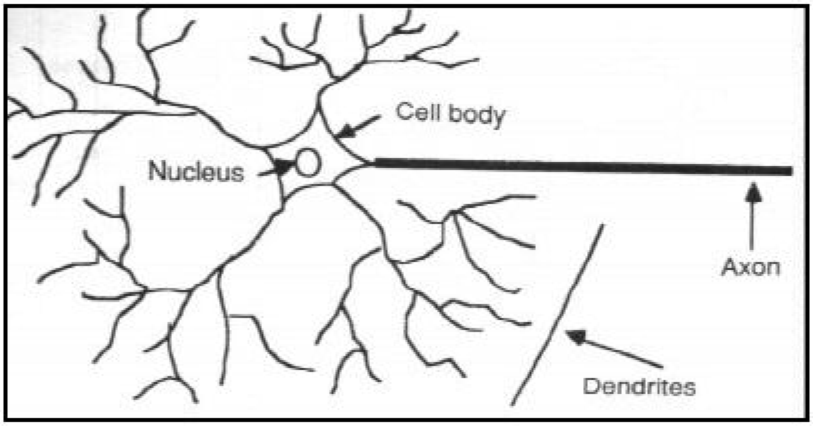
Here's a diagram of what one node might look like.



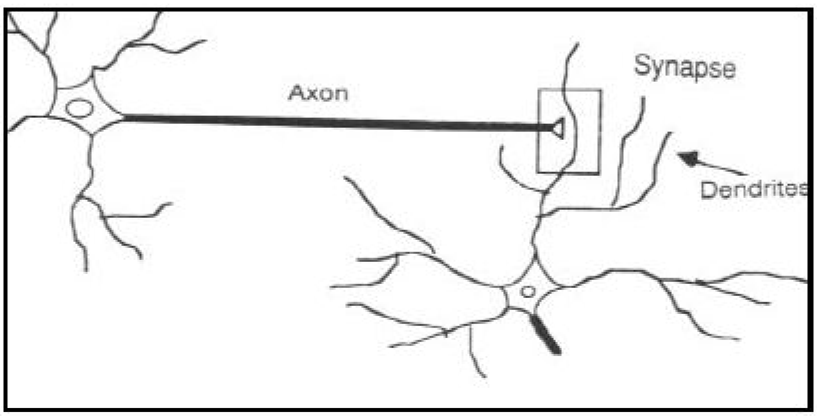
This model is an artificial way of modeling a human neuron which is the basic unit of a human nervous system.

2.1 Human neuron

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.



Components of a neuron



The synapse

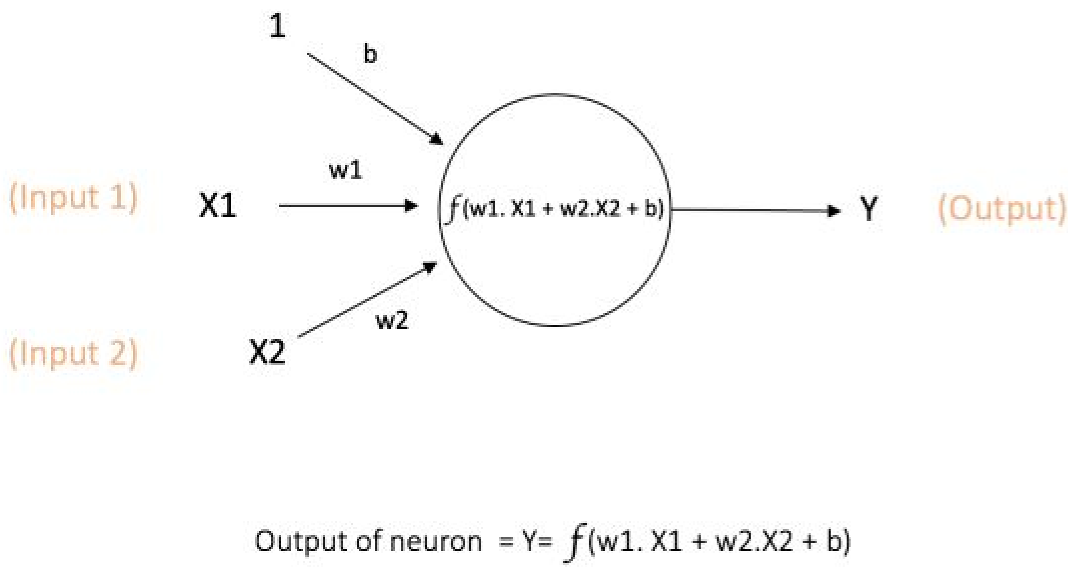
If you look closely that is the way the artificial neuron is modeled.

3 Artificial Neural Networks

An Artificial Neural Network (ANN) is a computational model that is inspired by the way biological neural networks in the human brain process information. Artificial Neural Networks have generated a lot of excitement in Machine Learning research and industry, thanks to many breakthrough results in speech recognition, computer vision and text processing. In this we will try to develop an understanding of a particular type of Artificial Neural Network called the Multi Layer Perceptron.

3.1 A Single Neuron

The basic unit of computation in a neural network is the neuron, often called a node or unit. It receives input from some other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs. The node applies a function f (defined below) to the weighted sum of its inputs as shown in Figure below:



The above network takes numerical inputs **X1** and **X2** and has weights **w1** and **w2** associated with those inputs. Additionally, there is another input 1 with weight b (called the Bias) associated with it.

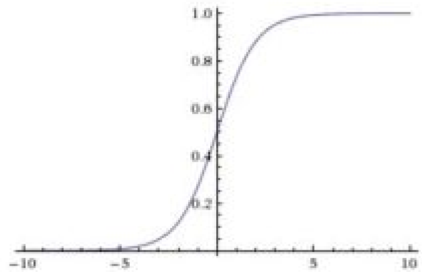
Importance of Bias: The main function of Bias is to provide every node with a trainable constant value (in addition to the normal inputs that the node receives)

The output Y from the neuron is computed as shown in the above figure. The function f is non-linear and is called the **Activation Function**. The purpose of the activation function is to introduce non-linearity into the output of a neuron. This is important because most real world data is non linear and we want neurons to learn these nonlinear representations.

Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it. There are several activation functions you may encounter in practice:

- **Sigmoid:** takes a real-valued input and squashes it to range between 0 and 1.

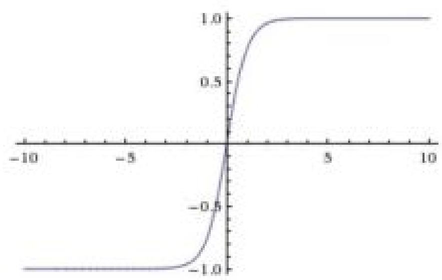
$$\sigma(x) = 1/(1 + \exp(-x))$$



Sigmoid

• **tanh:** takes a real-valued input and squashes it to the range $[-1,1]$.

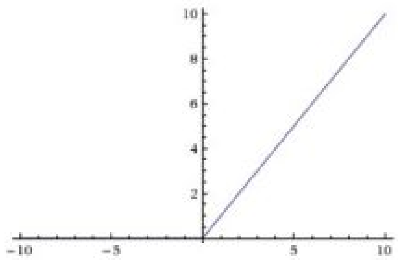
$$\tanh(x) = 2\sigma(2x) - 1$$



tanh

• **ReLU:** ReLU stands for Rectified Linear Unit. It takes a real-valued input and thresholds it at zero (replaces negative values with zero)

$$f(x) = \max(0,x)$$

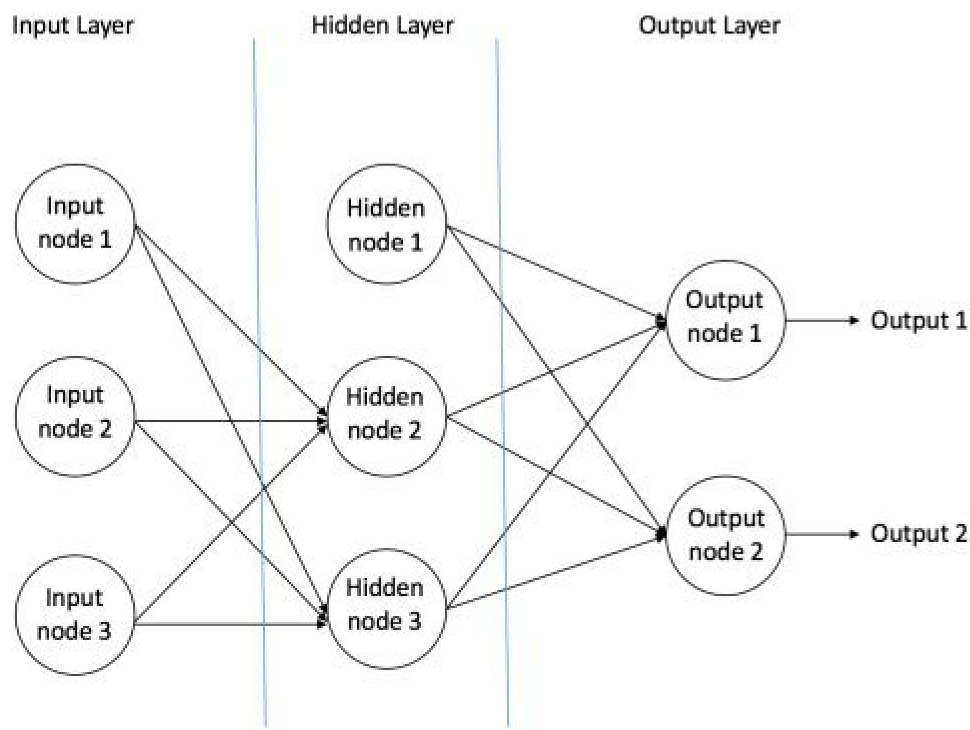


ReLU

3.2 Feedforward Neural Network

The feedforward neural network was the first and simplest type of artificial neural network devised. It contains multiple neurons (nodes) arranged in **layers.** Nodes from adjacent layers have **connections** or **edges** between them. All these connections have **weights** associated with them.

An example of a feedforward neural network is shown in figure below:



A feedforward neural network can consist of three types of nodes:

• **Input Nodes:** The Input nodes provide information from the outside world to the network and are together referred to as the "Input Layer". No computation is performed in any of the Input nodes,they just pass on the information to the hidden nodes.

• **Hidden Nodes:** The Hidden nodes have no direct connection with the outside world (hence the name ”hidden”). They perform computations and transfer information from the input nodes to the output nodes. A collection of hidden nodes forms a ”Hidden Layer”. While a feedforward network will only have a single input layer and a single output layer, it can have zero or multiple Hidden Layers.

• **Output Nodes:** The Output nodes are collectively referred to as the ”Output Layer” and are responsible for computations and transferring information from the network to the outside world.

In a feedforward network, the information moves in only one direction , forward - from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network (this property of feed forward networks is different from Recurrent Neural Networks in which the connections between the nodes form a cycle).

Two examples of feedforward networks are given below:

• **Single Layer Perceptron:** This is the simplest feedforward neural network and does not contain any hidden layer.

• **Multi Layer Perceptron:** A Multi Layer Perceptron has one or more hidden layers. We will only discuss Multi Layer Perceptrons below since they are more useful than Single Layer Perceptrons for practical applications today.

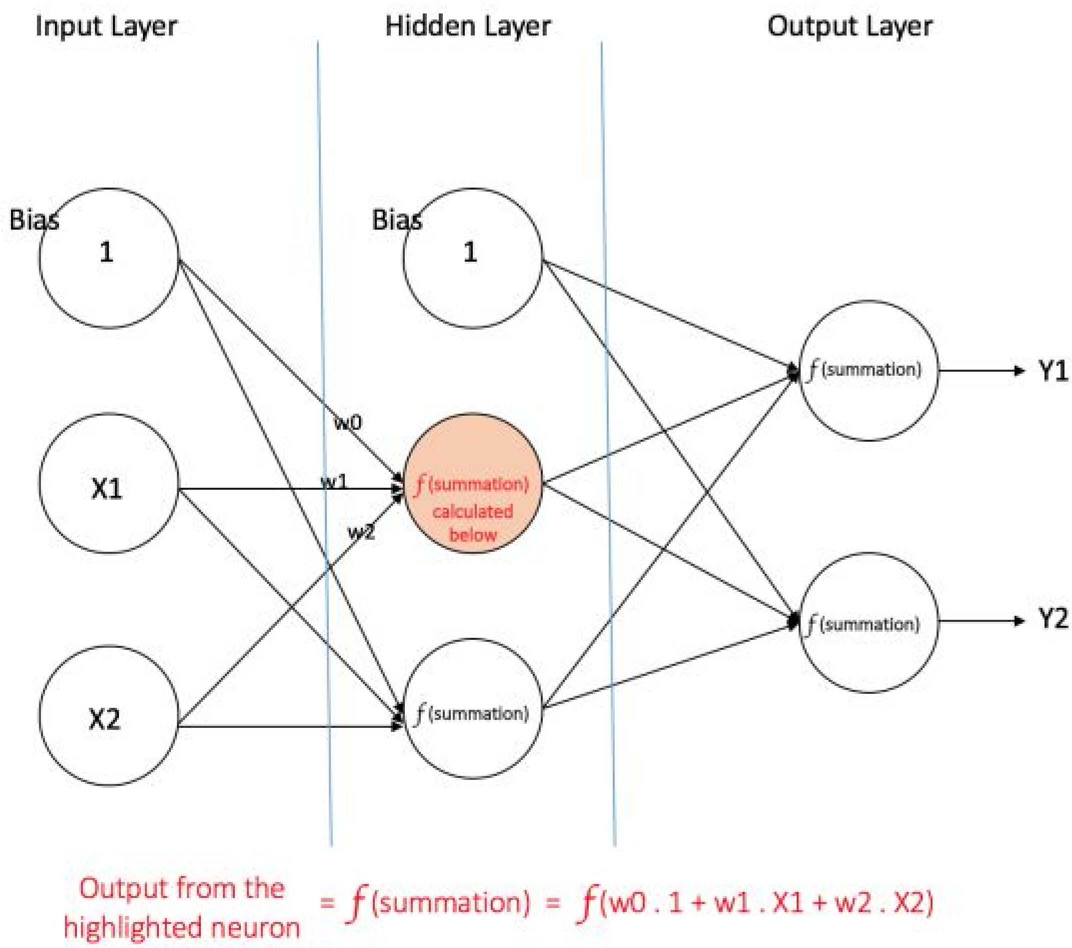
3.3 Multi Layer Perceptron

A Multi Layer Perceptron (MLP) contains one or more hidden layers (apart from one input and one output layer). While a single layer perceptron can only learn linear functions, a multi layer perceptron can also learn non linear functions.

Figure below shows a multi layer perceptron with a single hidden layer. Note that all connections have weights associated with them, but only three weights (w0,w1,w2) are shown in the figure.

Input Layer: The Input layer has three nodes. The Bias node has a value of 1. The other two nodes take X1 and X2 as external inputs (which are numerical values depending upon the input dataset). As discussed above, no computation is performed in the Input layer, so the outputs from nodes in the Input layer are 1, X1 and X2 respectively, which are fed into the Hidden Layer.

Hidden Layer: The Hidden layer also has three nodes with the Bias node having an output of 1. The output of the other two nodes in the Hidden layer depends on the outputs from the Input layer (1,X1,X2) as well as the weights associated with the connections (edges). Figure below shows the output calculation for one of the hidden nodes (highlighted). Similarly, the output from other hidden node can be calculated. Remember that f refers to the activation function. These outputs are then fed to the nodes in the Output layer.



Output Layer: The Output layer has two nodes which take inputs from the Hidden layer and perform similar computations as shown for the highlighted hidden node. The values calculated (Y 1andY 2) as a result of these computations act as outputs of the Multi Layer Perceptron.

Given a set of features X = (x1,x2,) and a target y, a Multi Layer Perceptron can learn the relationship between the features and the target, for either classification or regression.

3.4 How does a neural network learn things?

Information flows through a neural network in two ways. When it's learning (being trained) or operating normally (after being trained), patterns of information are fed into the network via the input units, which trigger the layers of hidden units, and these in turn arrive at the output units. This common design is called a feedforward network. Not all units ”fire” all the time. Each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections they travel along. Every unit adds up all the inputs it receives in this way and **(in the simplest type of network)** if the sum is more than a certain threshold value, the unit ”fires” and triggers the units it’s connected to (those on its right).

For a neural network to learn, there has to be an element of feedback involved–just as children learn by being told what they’re doing right or wrong. In fact, we all use feedback, all the time. Think back to when you first learned to play a game like ten-pin bowling. As you picked up the heavy ball and rolled it down the alley, your brain watched how quickly the ball moved and the line it followed, and noted how close you came to knocking down the skittles. Next time it was your turn, you remembered what you’d done wrong before, modified your movements accordingly, and hopefully threw the ball a bit better. So you used feedback to compare the outcome you wanted with what actually happened, figured out the difference between the two, and used that to change what you did next time (”I need to throw it harder,” ”I need to roll slightly more to the left,” ”I need to let go later,” and so on). The bigger the difference between the intended and actual outcome, the more radically you would have altered your moves.

Neural networks learn things in exactly the same way, typically by a feedback process called **backpropagation (sometimes abbreviated as ”backprop”)**. This involves comparing the output a network produces with the output it was meant to produce, and using the difference between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input unit –going backward, in other words. In time, backpropagation causes the network to learn, reducing the difference between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should.

3.5 How does it work in practice?

Once the network has been trained with enough learning examples, it reaches a point where you can present it with an entirely new set of inputs it’s never seen before and see how it responds. For example, suppose you’ve been teaching a network by showing it lots of pictures of chairs and tables, represented in some appropriate way it can understand, and telling it whether each one is a chair or a table. After showing it, let’s say, 25 different chairs and 25 different tables, you feed it a picture of some new design it’s not encountered before–let’s say a chaise longue–and see what happens. Depending on how you’ve trained it, it’ll attempt to categorize the new example as either a chair or a table, generalizing on the basis of its past experience–just like a human. Hey presto, you’ve taught a computer how to recognize furniture!

4 Advantages of Deep Learning

• **Self-driving cars**

Companies building these types of driver-assistance services, as well as full-blown self-driving cars like Google's, need to teach a computer how to take over key parts (or all) of driving using digital sensor systems instead of a humans senses. To do that companies generally start out by training algorithms using a large amount of data.

You can think of it how a child learns through constant experiences and replication. These new services could provide unexpected business models for companies.

• **Deep Learning in Healthcare**

Breast or Skin-Cancer diagnostics? Mobile and Monitoring Apps? or prediction and personalised medicine on the basis of Biobank-data? AI is completely reshaping life sciences, medicine, and healthcare as an industry. Innovations in AI are advancing the future of precision medicine and population health management in unbelievable ways. Computer-aided detection, quantitative imaging, decision support tools and computer-aided diagnosis will play a big role in years to come.

• **Voice Search & Voice-Activated Assistants**

One of the most popular usage areas of deep learning is voice search & voice-activated intelligent assistants. With the big tech giants have already made significant investments in this area, voice-activated assistants can be found on nearly every smartphone. Apple's Siri is on the market since October 2011. Google Now, the voice- activated assistant for Android, was launched less than a year after Siri. The newest of the voice-activated intelligent assistants is Microsoft Cortana.

• **Automatically Adding Sounds To Silent Movies**

In this task, the system must synthesize sounds to match a silent video. The system is trained using 1000 examples of video with sound of a drumstick striking different surfaces and creating different sounds. A deep learning model associates the video frames with a database of pre-recorded sounds in order to select a sound to play that best matches what is happening in the scene.

The system was then evaluated using a turing-test like a setup where humans had to determine which video had the real or the fake (synthesized) sounds. This uses application of both convolutional neural networks and Long short-term memory (LSTM) recurrent neural networks (RNN).

• **Automatic Machine Translation**

This is a task where given words, phrase or sentence in one language, automatically translate it into another language.

Automatic machine translation has been around for a long time, but deep learning is achieving top results in two specific areas:

– Automatic Translation of Text

– Automatic Translation of Images

Text translation can be performed without any preprocessing of the sequence, allowing the algorithm to learn the dependencies between words and their mapping to a new language.

• **Automatic Text Generation**

This is an interesting task, where a corpus of text is learned and from this model new text is generated, word-by-word or character-by-character.

The model is capable of learning how to spell, punctuate, form sentences and even capture the style of the text in the corpus. Large recurrent neural networks are used to learn the relationship between items in the sequences of input strings and then generate text.

• **Automatic Handwriting Generation**

This is a task where given a corpus of handwriting examples, generate new handwriting for a given word or phrase.

The handwriting is provided as a sequence of coordinates used by a pen when the handwriting samples were created. From this corpus, the relationship between the pen movement and the letters is learned and new examples can be generated ad hoc.

• **Image Recognition**

Another popular area regarding deep learning is image recognition. It aims to recognize and identify people and objects in images as well as to understand the content and context. Image recognition is already being used in several sectors like gaming, social media, retail, tourism, etc.

This task requires the classification of objects within a photograph as one of a set of previously known objects. A more complex variation of this task called object detection involves specifically identifying one or more objects within the scene of the photograph and drawing a box around them.

• **Automatic Image Caption Generation**

Automatic image captioning is the task where given an image the system must generate a caption that describes the contents of the image. In 2014, there was an explosion of deep learning algorithms achieving very impressive results on this problem, leveraging the work from top models for object classification and object detection in photographs.

Once you can detect objects in photographs and generate labels for those objects, you can see that the next step is to turn those labels into a coherent sentence description.

Generally, the systems involve the use of very large convolutional neural networks for the object detection in the photographs and then a recurrent neural network (RNN) like an Long short-term memory (LSTM) to turn the labels into a coherent sentence.

• **Automatic Colorization**

Image colorization is the problem of adding color to black and white photographs. Deep learning can be used to use the objects and their context within the photograph to color the image, much like a human operator might approach the problem. This capability leverage the high quality and very large convolutional neural networks trained for ImageNet and co-opted for the problem of image colorization. Generally, the approach involves the use of very large convolutional neural networks and supervised layers that recreate the image with the addition of color.