Neural Networks and Dynamic Bayesian Networks in Machine Learning

Selvakumar Vigneshwaran

NUS High School of Mathematics and Science

Abstract

This essay has been written on Dynamic Bayesian Networks and Neural Networks. It will explain the fundamental workings of these systems and the Mathematics behind them. It will definitely be a good guide to anyone who wants to learn these systems in a way in which words and Mathematics are combined and entwined to make the journey of understanding a simpler one. I have brought about this journey in the form of a trip in a technologically advanced world, where Image Analytics and Voice Recognition are involved in the reader's daily life. I hope you enjoy it!

~~~~~~

Artificial Intelligence: a name we might hear quite often in the current technologically advanced world. But what is it in true nature? Is it as simple as the word means? Just intelligence made artificially or, is there something more to it? Indeed, Artificial Intelligence is referring to an intelligent robot, capable of making self-directed decisions; for now. In future, we humans aim to make Artificial Intelligence capable of self-directed awareness of itself and the ability to mimic human neural networks. But for now, I would be looking into two specific types of AI tools and how they function in Machine Learning: Dynamic Bayesian Networks and Neural Networks.

Since early times, the need for efficient capture of prey by predators allowed the evolution of eyes and ears in complex organisms to be made possible. Our eyes and ears are needless to say, especially important and amazing. Imagine a world in which we do not have eyes and ears must go bouncing of walls. Yes, quite disastrous as you might say. However, the ability of our eyes and ears to conceive the surroundings is one thing; it is really the ability of our brains to *use* that information that is astounding in its own nature. A ground-breaking discovery of nature itself, the neural network and Bayesian networks in our brains has enabled us to differentiate objects seamlessly. Indeed, nature is amazing in its own sense.
~~~~~~

Humans have tried to achieve this unique ability in machines, and that has led to the birth of Image Analytics and Voice recognition.

Image Analytics and Voice Recognition are quickly becoming a particularly important part of modern life. The simplest forms of image analytics can be seen in the Quick Draw ! application made by Google. In this website, an Artificial Intelligence robot can use Image analytics to match the drawn object with a real-life object, using data received from millions of users' drawing world-wide. Though, Image Analytics also has some serious applications too, for example Face Recognition in high-security areas. Voice Recognition is also used in the AI experiments done by Google (again). These use sounds and the components of these sound waves to distinguish specific sounds but how does Mathematics play a crucial part in this? They do this by using Dynamic Bayesian Networks and Neural networks. Let us now start our intriguing journey in uncovering the hidden gems of Mathematics in these complex Artificial Intelligence systems!

(Hereafter, Artificial Intelligence will be addressed as AI at some points)

[A Brief Opening Paragraph on AI Classification]

Of the wide seas and open forests grown by AI, Machine Learning is a large savannah, divided into regions of constant development, renewed from age. Machine Learning has many sub-categories, one of which is Deep Learning. Deep learning is a type of learning that aims to use Neural networks like that of the brain to carry out learning. This learning can be supervised or semi-supervised, or unsupervised. Supervision is, in its essence, giving the labelled outputs together with the inputs for the Machine to learn from. Unsupervised, is to just give the data and let the Machine figure it out itself. Another type of model that is not given a name with a large category devoted to it, are Dynamic Bayesian Networks. Having explained these, the two types of Machine Learning that will be explained here will be Neural Networks and Dynamic Bayesian Networks. Now let us begin and step out into the world of AI, pursuing the flighty temptress, Knowledge!

You are in a technologically advanced supermarket, where AI are used for identifying what you buy and calculating the price. You take a can of sardine and place it in front of the scanner. You hope very much the AI will fail and take the can of sardine to be a can of tomato (cheaper). However, much to your disappointment, a label shows Sardine and the

price right beside it. But how do AI robots identify different objects using Image Analytics? They do this by using **Artificial Neural Networks** or more commonly known as **Connectionist Systems.** These systems are named for them being vaguely inspired by neural networks found in animal brains. These can use pre-set examples to derive their output from an input, rather than using a pre-programmed set of steps and rules that act on a limited set characteristic. For example, if we want the network to identify the cat, we will have a set of examples, which will be used in identifying it as "Cat". The network does not know that it has to identify certain characteristics , such as " Fur " and " Whiskers " ; it will just compare the Input with hidden exemplary variables , like color variables and types of edges and shape structures found , and create an output. This output can be as clear as " Cat " or " No Cat " or it could be a percentage measure such as " 75 % Cat " as shown in Fig 1.1 Now we know the basic overview of *what* these AI do , let's take a closer look at *how* they do it.

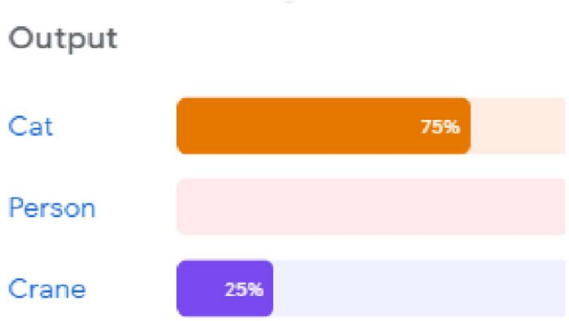


Figure 1.1 An example of percentage value given by an Image Analytic AI

Neural Networks in Image Analytics is divided into two broad categories, **feedforward neural network** and **recurrent neural network**. Feedforward neural networks have information flowing in one direction only, while recurrent neural networks have information flowing in both directions, correcting and analyzing itself. To get a good overview of basic neural networks, let us have a look at how a simple neural network works using an incredibly early type of "neuron"

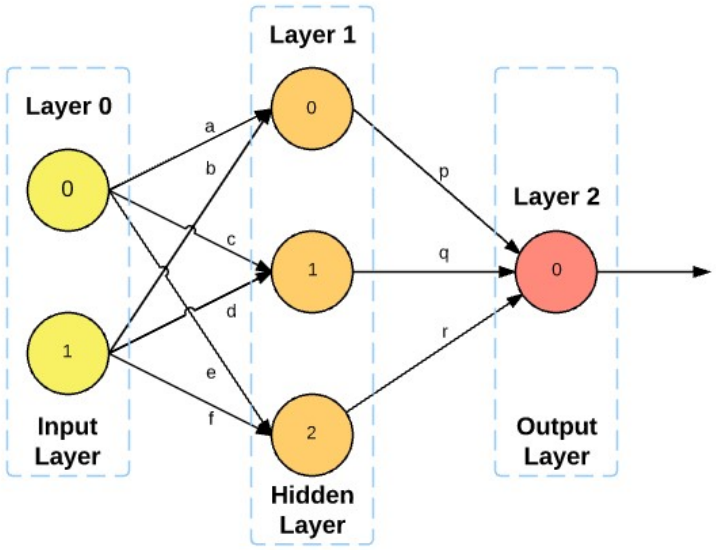
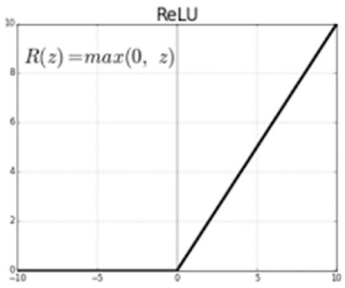


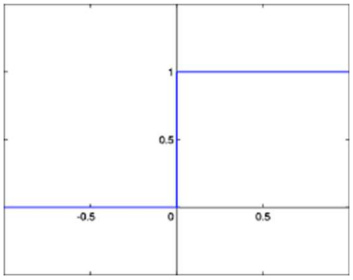
Fig 2.2 A diagram stating a very simple neural network

We will begin understanding these nodes by looking at the earliest type of such neurons known as a ***Perceptron.*** Perceptrons were some of the earliest "neurons" used The first layer is the input layer. The input layer consists of nodes that are assigned numbers. In multilayer networks there will be hidden layers too. These Hidden layers allow for the function of a neural network to be broken down into specific transformations of the data. Each hidden layer function is specialized to supply an outlined output. The output layer, well, gives the output. Perceptrons are basically mathematical functions, that follow the steps told below.

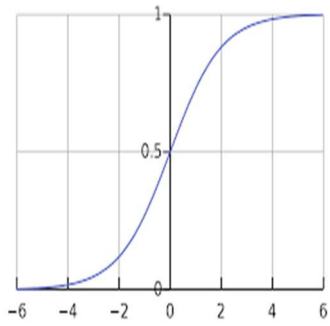
1. Multiply all the inputs by their **weights** ***w***, real numbers that express how important the corresponding inputs are to the output,
2. Add them together , referred as **weighted sum**: $\sum wj\ xj$,
3. Apply the activation function, in other words, determine whether the weighted sum is greater than a threshold value, where threshold is equivalent to bias, and assign 1, or less and assign 0 as an output. [Sometimes a bias is added to adjust the data given for an output] The activation functions can be (in modern neurons) …



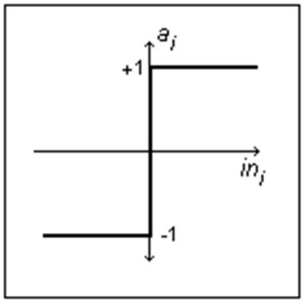
- The ReLU function, also known as the ramp function/



- The Step function, of which the most commonly used is the Heaviside Step function.



- The Sigmoid function



- The Sign function

- Or any other activation function that exists (yes, there are more activation functions than just these)

$$\sum_{i=1}^{m} w_i x_i$$

1,2b. where *m* is the number of inputs while *x* is the input and *w* is the weight

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

3b. where the number given by the activation are used to come up with a 1 or 0 for a result.

Before, we begin talking about Activation functions, what are functions? Mathematically, are binary relations over two sets that associate every element in the first set to that of the second set. In an earthier way of expressing that, functions are like machines that take in inputs and give out outputs.

The activation function can be a variety of functions, four of which are the **step function**, **sigmoid function** and **sign function** and **ReLU function** and the most common now is the ReLU function. The ReLU function is the function

$$\boldsymbol{f(x) = \max(0, x)} .$$

The ReLU function is a non-linear approach to using an activation function and is quite simple in its way. It basically follows the rule,

```
1  if input > 0:
2      return input
3  else:
4      return 0
```

Mathematically, it can be simply expressed as g(z) = max{0, z}. That means,

$$\max\{x_1,x_2\}=\{x_1,\text{if } x_1>x_2$$

$$=\{x_2,\text{otherwise}$$

This function has the many desirable properties of a linear activation function, giving multiple outputs. However, it is Mathematically classified as a non-linear function as for all negative inputs, it returns 0 as an answer.

One way to import ReLU in Keras , through the keras.layers.Activation module.

```
“ from keras.layers import Activation, Densemodel.add(Dense(64
,activation='relu')) “
```

Next, we will take a quick look at the Step and Sigmoid function. The step function and the sigmoid function are non-linear. The sigmoid function has many varieties, one of which is the logistics function,

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}},$$

where

e = the natural logarithmic base (also known as Euler’s number),

x_o = the value of the sigmoid's midpoint,

L= the curve's maximum value,

k = the logistic growth rate or steepness of the curve.

And the value of x varies from $-\infty$ to $+\infty$. The logistic function also has many unique properties, one of which states that. The logistic function has the symmetry property that

$1\text{-}f(x) = f(\text{-}x)$.

Thus, $x \mapsto f(x) - 1/2$ is an odd function.

The step function has many varieties too. As you might have noticed the name of an activation fucntion naturally refers to a class of activation functions quite close to each other in Mathematical nature. The Heavyside Step function is one of those, and its discrete form is similar to that of the ReLU function.

$$H[n] = \begin{cases} 0, & n < 0, \\ 1, & n \geq 0, \end{cases}$$

It is quite similar to understand and as an extra note, it is also derived from one of the ramp functions,

$$H(x) := \frac{d}{dx} \max\{x, 0\} \quad \text{for } x \neq 0$$

It states that *H*(*x*) can be derived from the ramp function as its derivative. What is a derivative? Derivatives are basically functions that measure the rate of change. The statement d/dx basically states "take the derivative of this thing I'm about to write after this". Therefore, we can quite easily define the statement above. X must not be equal to 0 due to the constant rule,

$$\frac{d}{dx}(c) = 0$$

The sign function is quite similar to that of the step function, just that the value of minimum goes till –1.

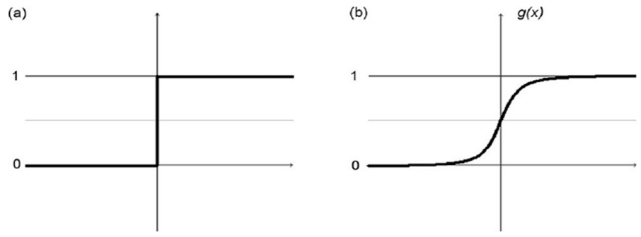


Figure 3.3 A diagram showing the graphs of the Step and Sigmoid function

Though, when dealing with Perceptrons at that time, the step and sign functions were very well equipped for good and clean answers, giving set results of -1 and 1 . The step function assumes the shape shown in Figure 3.3. The next stage of the more advanced neural network is the **Multilayer Perceptron Layer**. These however, required the use of more complex activation function that are more linear.

Multilayer Perceptron Layers have a hidden layer (which we talked about earlier) that enable it to have lesser error though the generalization of the machine as a whole will be quite diminished. As time passed by, better more powerful neurons came into play using activation functions other than the Step and Sign. These allowed a more varied number of outputs, allowing us to come up with better image analyzation features.

Now, we have looked at Feedforward neural networks, but let us just take a quick glimpse of recurrent neural networks. Recurrent neural networks use a special process known as backpropagation. Now, we've verified Feedforward neural networks, but let us now take a quick but enlightening look at recurrent neural networks. Recurrent neural networks use a special process mentioned as backpropagation. Essentially, backpropagation evaluates the expression for the derivative of the worth function as a product of derivatives between each layer backwards with the gradient of the weights between each layer being a modification of the partial products *(the product of one term of a multiplicand and one term of its*

multiplier.) . Now, there are some things you might not understand in this. What is a worth function? A worth function, also known as a loss function, works by mapping an event or values of one or more variables onto a real number plane, intuitively representing some "cost" associated with the actions of the event. This technique helps rectify weights without any manual help.

Now, that we know a fair much about neural networks, let us now exit the supermarket and return home. You can't wait to finish your delicious dinner that awaits you at home and sit down by the window, with your legs on the pouffe and a large cup of cocoa in your hands. You saunter down the street, humming your favorite tune and come to your car. You take out your keys and slid it into the car-door lock and hear a smooth mechanical click. Opening the door, you get in. "Welcome, please start the car using voice recognition" states a cool female voice. You jump a little at this sudden welcome

(This new feature takes some time getting used to) and grunt a "hello", murmuring under your breath and employing a few well-chosen words for the situation. How does an AI recognize your voice? They do this by using Dynamic Bayesian Networks. What is that? Before we delve into the depths of probability, let us scrape the surface, **Bayes theorem.**

What is Bayes theorem? Bayes theorem was first used by Reverend Thomas Bayes. This ingenious but simple theorem states the probability of an event based on some prior conditions. A common, but simple example can be used to explain these. Let us have this scenario where a patient has a symptom for cancer, but that might not necessarily mean he must have cancer! However, all those who do have cancer have this symptom. So, what is the probability that a person with the symptom has cancer? Let us have a set of data for this case …

Symptom	Cancer	Cancer	Total
	No	**Yes**	
No	99989	0	99989
Yes	10	1	11
Total	99999	1	100000

Figure 4.4 A table showing the values for Cancer study

So, the possibility that we have a symptom and have cancer can be derived if we take the

$$\frac{\text{Number of People who have symptom and have cancer}}{\text{Number of people who have symptom}}$$

However, there is a need for generalization. Therefore, we have to make some changes to this situation to fit that of others. In our first step to that, we will make the equation

$$= \frac{\text{Probability of Symptom having Cancer x Probability of having Cancer}}{\text{Probability of having Cancer}}$$

$$= \frac{\text{Probability of Symptom having Cancer x Probability of Cancer}}{\text{Probability Of Symptom having Cancer x Probability of Cancer + Probability of Symptom having No Cancer x Probability of having No Cancer}}$$

"Now we can't waste time writing such long sentences, can we?" might be a indignant Mathematical students question, and that is exactly what Mathematicians thought. They

decided to make the equation much easier to write and came up with Mathematical Notations to fit the sentences.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Where *P* means Probability, A and B refer to events and the line in-between refers to " given that ". This can be seen matching with the formula we got above. In simpler representation below …

- P(A\B) is a **conditional probability** : the likelihood of event A occurring given that B is true.
- P(B\A) is also a conditional probability: the likelihood of event B occurring given that A is true.
- P(A) and P(B) are the probabilities of observing A and B respectively; they are known as the **marginal probability**.

In this case P(A\B) is called the **Posterior**, P(B\A) is known as the **Likelihood** , P(A) known as Prior and P(B) is known as the **Normalizing Constant.** Though, the names of these variables vary in different places, the essence of the equation stays constant. Now we have a good understanding of Bayes theorem, and more importantly of the notation, let's take a look at Bayesian Networks.

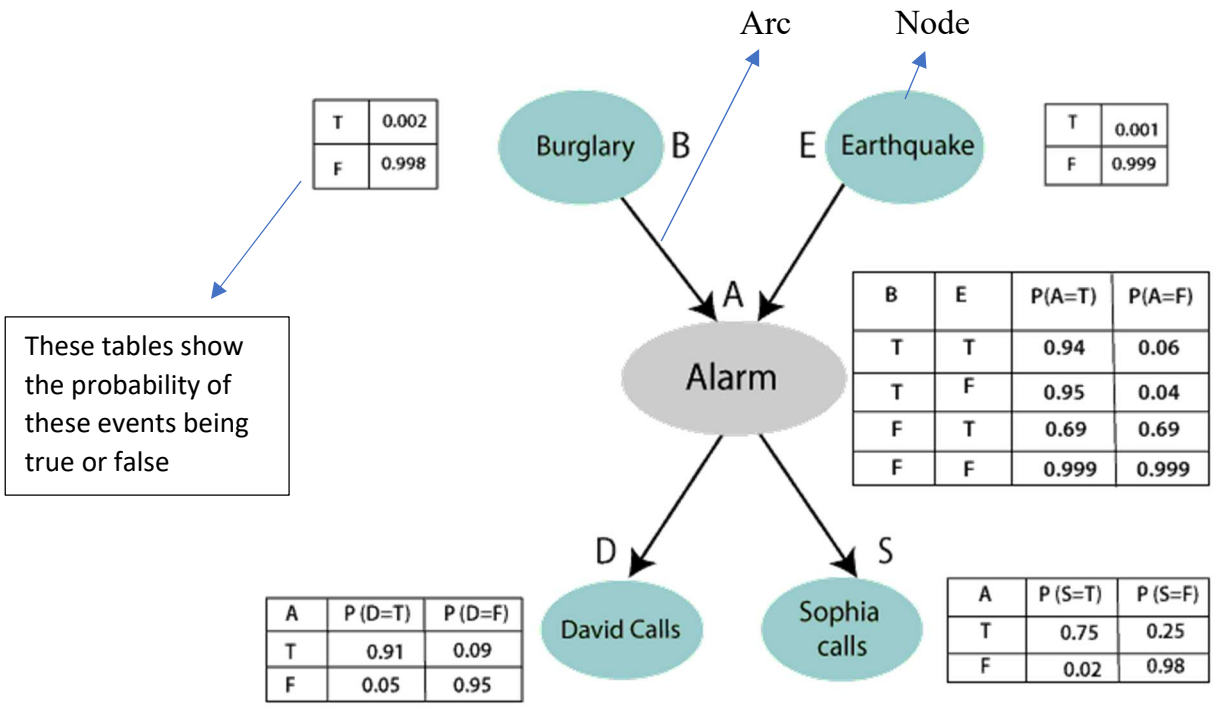


Figure 5.5 A Directed Acyclic Graph on a Burglar Alarm and its reactions

A **Directed Acyclic Graph (DAG)** is a type of graph in which it's impossible to come back to the same node by traversing the edges and it's so happens that Bayesian Networks are such graphs. Bayesian networks such as that seen in Figure 4.4, allows us to determine joint probabilities for events based on other occurrences. Knowledge about Bayes theorem will help us in understanding these. As seen above, we have the situation where an earthquake and a burglary trigger a burglar alarm. David and Sophia have the job of reminding their boss when a burglary happens. However, David and Sophia have chances of calling their boss even when the burglar alarm did not ring, due to their noisy neighborhood. So, what is the possibility that both David and Sophia call their Boss and there is no burglary or earthquake burglary?

Now we understood the Bayes theorem, and by doing that, we understood the notation. However, we will not be using the actual theorem here; that will be more for complex problems. However, the probability of that happening can be calculated by simple rules of multiplication of different events' probability. This we will do by using a **Joint Probability.**

P(S, D, A, ¬B, ¬E) = P (S|A) *P (D|A)*P (A|¬B ^ ¬E) *P (¬B) *P (¬E)

This is a joint probability

A Dynamic Bayesian Network is a Bayesian network that is constantly changing its connections and changing its arc lengths, as these may sometime represent the possibility of the event. A Dynamic Bayesian Network (DBN) is constantly changing and allows the Artificial Intelligence to use some learnt or supervised-and-taught characteristics to come up with an answer. DBN's use temporal nodes that are time varying. They allow constant change of values as well. This allows them to change their probabilities based on previous nodes and previous time-based nodes. That is one of the main differences between a DBN and a normal Bayesian Network.

There is also another key to DBN's and some normal Bayesian Networks, the Brier score. The Brier score is like a cost function of a DBN.

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2$$

Where f_t is the probability of the event, o_t the actual outcome of the event at instance t (o_t is 1 if it does happen and 0 if it does not happen) and N is the number of forecasting instances. In essence, it is the mean error squared.

Now, we have talked so much about how these systems work, yet we can't miss out their uses can we? DBN's and Neural Networks have played a huge role in Image Analytics and Voice Recognition. One of the main ways in which image recognition is used right now in the world is the usage of Face ID. Face ID is used in mobile phones and other personal electronic devices. Though, this might not be a good idea for security as it is a weak option. More uses include Airport Vehicle and Land Transport Vehicle Damage Assessment, providing second opinions for doctors and also for scanning areas for activities that are happening. Voice recognition can also be used, and this might be something slightly more theft-proof, it is still a weak option. Voice recognition can, however, be also used in Transport and also in Retail where customers can buy products without even touching their phone screens and data in danger of being hacked can be deleted in seconds with a simple voice command. However, there is a great solution to solving crimes by using a combination of these two technologies: Visual Speech recognition. Whatever the present of AI right now, we can say that the future is going to something exciting and great to look forward too. Now, why don't you go and catch up on that delicious plate of dinner.