Lecture 2

Algorithms and Computers

Rahul Bhattacharya

Machine Learning: The New Science

Part I

What is this New Science of Machine Learning?

Putting Machine Learning in the context of Artificial Intelligence (AI)

- 💻 Introduction, Lecture 0

Part II

Foundational Concepts of Machine Learning

- 💻 Lectures 1 – 12.

Part III

Solving Real Life Problems with Machine Learning

- 💻 Lectures 13 – 25.

Part IV

Machine Learning: The Changing Landscape of Knowledge

(Near and the Not So Distant Future)

- 💻 Lectures 26, 27.

Pedro Domingos, the noted computer scientist at the University of Washington, opens his book, *The Master Algorithm* with the statement, “We live in the age of algorithms”. He is spot on when he says that algorithms are “in every nook and cranny of civilization”. Anything and everything that we do at our home or in our workplace are governed by algorithms. In Lecture 1, we talked a bit about algorithms, let’s talk some more and in somewhat greater detail. Let’s try to understand how they work inside a computer.

An algorithm is a set of step by step instructions and given in a certain order to a computer telling it what to do, how to execute a task. The instructions have to be precise and unambiguous and must be given in a particular order. A cooking recipe is also a set of instructions but it is not an algorithm. The computer needs to understand in what sequence the order has to be executed. An example of an algorithm could be:

Generate a uniform random number, $U(0,1)$*, between 0 and 1*

Assign $U(0,1)$ *to* y

If $U(0,1) > 0.5$ *Then* $y = 1$

Else $y = 0$

Of course, a computer does not understand English and hence it will not be able to comprehend what we mean by the above statements (the sequence of instructions). So, we have to write the above instructions in a specific language (called a computer language) and present them in a certain specific format which is known as a **computer program**. But a computer program is a piece of software for communicating with the computer. It is not the computer's brain which resides inside that box (comprising the screen, keyboard and the machinery inside) that sits on our desks or on our laps. Instructions – the algorithms – have to be processed by a computer's brain. So all the instructions in a computer program needs to be translated into a language that the machine – i.e. the computer – can understand and operate upon (the **machine language**). This machine language operates a computer's brains, where our algorithm – the one that we fed to the computer along with data via a computer program – is decoded by the computer and executed.

A computer brain is made up of transistors – or, switches – that flip "on" and "off" as electric current flows through them. That is the only language they understand, the language of "on" and "off". In other words, a computer lives in a binary world where it understands everything through only "on" or "off". The state of a transistor, represented by an "on" or an "off" represents one bit of information. An "on" can be represented as 1 and an "off" can be represented as 0.

Let's try to understand what algorithms are in the context of computers. Pedro Domingos, has given an excellent exposition of algorithms in his book *The Master Algorithm*. Let's try to take the road he has taken, The simplest of all algorithms is this: *Switch on* or a *Switch off*. A transistor inside a computer either switches on or switches off. If it switches on then a value of 1 appears (one bit of information) and if it switches off then a value of 0 appears (another bit of information). Combining two bits of information gives us another simple algorithm; and, thereby helps to lay the fundamental architecture of modern-day computers. In modern day computers, a transistor switches on or off in response to other transistors inside the computer. This creates a fabric of reasoning which helps the computer to understand – in fact, translate – all the instructions that we give in English. The building blocks of this fabric, the hardware or the brains of the computer, are known as logic gates and there are three main logic gates: AND, OR and NOT, each in itself an algorithm.

If say, a transistor A switches on only when transistors B and C are both on then it creates an AND gate. This is an algorithm: *If B and C on then A is on*. If transistor A switches on when either transistor B or transistor C is on then it is an OR gate. The algorithm is: *If B or C on then A is on*. Finally, if transistor A switches on when transistor B is off creates the NOT gate. The algorithm is: *If B is off then A is on*. You may find this absolutely wonderous but all algorithms in this world that you can think of, from the simplest to the most challenging and complex can actually be broken down to these three basic algorithms: AND, OR and NOT. That is why computers can computer such difficult problems in no time.

Say, an oncologist – a doctor who specializes in cancer diagnosis and treatment – treating a Diffuse Large B Cell Lymphoma (DLBCL), a type of non-Hodgkin's lymphoma (cancer) wants to decide whether to go for R-CHOP chemotherapy (treatment) or R-EPOCH chemotherapy. In the absence of very clear evidence to the contrary (at the time of this writing), it is generally believed by the oncologists that both R-CHOP and R-EPOCH yield similar results for DLBCL treatment, though in the recent years, many doctors have preferred R-EPOCH treatment for certain rarer sub-types of DLBCL, such as what is known as the Double Hit Lymphoma. The general belief amongst doctors and the industry best practice has so evolved that in the case of double hit lymphoma R-EPOCH chemo yields better results.

There is a fine line between double hit lymphoma and the other types lymphoma. A critical gene mutation known as Myc/8q24 has to happen for the pathologist to diagnose that disease as double hit. Further, generally speaking all double hit lymphomas are also double expressor lymphoma where there is gene expression – two of these three genes are expressed Bcl2, Bcl6 and cMyc – but there is no mutation or rearrangement of the genes.

The oncologist may use a simple algorithm to find out if he should be using R-CHOP or R-EPOCH treatment. *If it is double hit then go for R-EPOCH otherwise go for R-CHOP*. In terms of logic gates AND and OR we can write this algorithm (in a diagrammatic format) as follows.

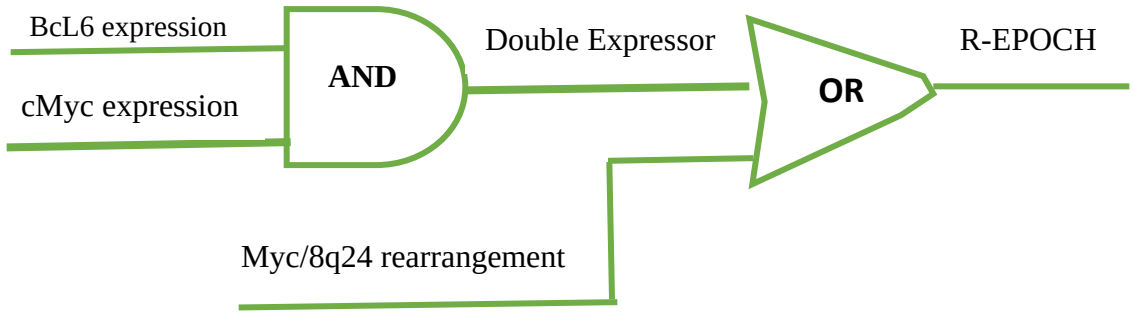

Figure 2.1 An algorithm (inside a computer's brain) for treating Lymphoma

Far more complex problems in cancer research, space physics, quantum mechanics or any other field that you can think of can be completely composed of just three operations AND, OR and NOT. Whatever instructions that we give to a computer is ultimately broken down into these three operations by the computer.

In the previous lecture we briefly talked about how an algorithm acts a process of learning from the data. Let's elaborate a bit more on that idea. For that, let's fall back on something that we learnt in our middle school: algebra. Indeed, as middle school students when we first learnt algebra and started solving simple problems in algebra, we were engaging in what is known as algorithmic thinking, i.e. trying to think about solutions of problems using an algorithm. So, how is algorithm a process? The following is referenced from Mark Driscoll's excellent book *Fostering Algebraic Thinking*.

Think of a common word problem (the type that we all solved in school).

Word Problem (Pre-algebra) in Math: Sarah bought 40 oranges for 30 cents each. She ate 5 of them and she then sold the rest for 50 cents each. What was her profit?

This is in fact a problem in arithmetic (what education experts call prealgebra problems) and can be done using simple computational steps.

Step 1: subtract 5 from 40.

Step 2: Then multiply the answer (obtained in step 1) by 50

Step 3: From that answer (obtained in step 2) subtract the product of 30 and 40

We could have broken down the above algorithm in four steps instead of three (rather than combining the third and the fourth step into one step). Anyway, this sequence of steps – instructions, if you will – yields the result 550 cents as the answer. When as school kids we solved this problem, we thought of each step as an independent action and the series of the steps as a sequence of independent actions. First do this and then do that and finally do this. Each step, in our mind, was an independent action and we did not think of it as whole action that takes and input and gives an output.

Now consider the same problem that appears in algebra.

Algebra (Word) Problem in Math: Sarah bought some oranges at 30 cents each. She ate 5 of them and then she sold the rest for 50 cents each. If her profit was $5.50, how many oranges did she buy?

This is also a word problem but it is now in the domain of algebra and not arithmetic. Here we are given the result – the output of the problem – and asked to find the input; the reverse of what we did in our

arithmetic word problem above. The way the problem is presented is upside down: start with the output and then find the input. It is an inverse problem. This kind of problems requires a different type of thinking. We can no longer visualize the computation process as a step by step sequence starting from the input and going all the way down to the output (result). That way we will not be able to find the answer, i.e. the correct input. The thinking requires us to create a kind of mapping from the output to the input and this mapping process creates what we call a *function* (as we have learnt in the first lecture). The entire sequence of calculations in the first problem, step 1, step 2 and step 3, needs to be "packaged" to form an appropriate function.

In the above problem, the entire calculation comprising step 1 to step 3 should be looked upon as one whole operation – or, a process – whereby the output is being mapped to the input. If we taken the unknown input as x, a variable whose value needs to be determined then the entire process can be written down as:

$$50(x - 5) - 30x = 550$$

Here, the left-hand side, LHS (all terms appearing on the left side of the "equal to" sign) encompasses all the steps that would go into an arithmetic calculation of the problem (step 1, step 2 and step 3) and the right hand side, RHS (the term appearing on the right side of the "equal to" sign) gives the output. By solving for x in the above equation we will arrive at the input, i.e. 40. The algorithm is then the process that maps the RHS to the LHS: *RHS* → *LHS*. Here, what we do is to employ reasoning to get back to input from the output, which is known in advance.

Going from the Word Problem in arithmetic to the one in algebra marks a tremendous shift in our thinking. We no longer think of the computation-based sequence of steps, i.e. an algorithm, as a series of independent operations or actions but rather as one unified process, or what Driscoll calls "an entity unto itself".

This shift in thinking on a much grander scale has given rise to the discipline of machine learning.

References:

- Driscoll, Mark, *Fostering Algebraic Thinking*, Heinemann, 1999
- Domingos, Pedro, *The Master Algorithm*, Basic Books, 2015

Machine Learning: The New Science is an **online course** developed by Risk Latte AI to increase awareness and a very basic understanding of the field of Machine Learning, the new science that is taking our world by storm. In the next ten years, the world that we see around us would have changed completely, thanks to machine learning. In the next 20 to 30 years at most we, the human beings, would be in a totally uncharted territory, once again thanks to machine learning and the much bigger and related discipline of artificial intelligence (AI). In more ways than one, machine learning is the stepping stone to understanding artificial intelligence. Machine learning is a very important sub-field of artificial intelligence.

This online course prepares a person to start off with a more formal **Machine Learning 101** course. This course contains very easy to understand lectures and simple tutorials implemented in Microsoft Excel™ to illustrate real life problem solving using machine learning algorithms.

- This course is primarily targeted towards high school, college and university students and middle to senior level working professionals and executives around the world who have a very basic knowledge of mathematics and a very limited or even **no knowledge** of computer programming.
- For certain segments of our target audience, such as certain specific groups of high school, college and university students in India and other developing countries this online course costs only US$8.00 (USD Eight only).
- This course comes with on-demand and customized onsite, classroom lectures and workshops at a minimal cost.
- A lot of the lectures and many Excel spreadsheets containing worked examples are **FREE** and can be easily downloaded from our website. These convey the general flavor of the course.

Risk Latte AI is a unit of **Risk Latte Americas Inc.**, a Montreal, Canada incorporated company, that is in the business of developing machine learning algorithms and software for banks, financial institutions, healthcare and the education industry as well as developing gamification and social learning applications for the general public.