The Obsidian Understanding Of What the Bojangles AI Is… and ‘Can’ Be

# Meta Explanation

Watch my videos, read my books, experiment with my sourcecode, don’t speak to my family!… I may fill this section in properly at a much later date, but for now, I feel like any more detail will simply either confuse my lovers, or better yet anger my haters.

The explanation will likely go something along the following lines (starting with the ‘least’ useful, and ending with the most useful):

(LEAST USEFUL TO MANKIND)

1. **Deep Learning** (Initial Operating Capability {IOC}, although mostly obsolete, or rather, currently misused… Please start pushing to Full Operating Capability {FOC})
2. **Relational Learning** (Development & Delivery)
3. **Signature Learning** (mostly late Theory & Research, with some successful D&D)
4. **Symbolic Learning** (straddled across mid T&R and mid D&D, a highly unusual situation… basically, the investment has run on ahead of the sensible academic science, for better or for worse)
5. **Abstractive Learning** (early T&R)
6. **Probabilistic Learning** (firmly in D&D, with a little more groundwork to be rolled out)
7. **Causal Learning** (mostly T&R, with not a little groundwork to fully enter D&D)
8. **Reward-Penalty Learning** (early D&D)
9. **(Mimic) Entropy Learning** (not even T&R yet)
10. **Objective Learning** (very early T&R)

(MOST USEFUL TO MANKIND)

Nota Bene 0001: And please remember, that this takes us merely to around mid 21st century. From around 25NC onwards, these core AI algorithms need to be composed into **AGI-Ensembles**, and eventually networked into **Collective Agent Teams**… I’m still sceptical as to whether a sort of **Overseer AI System** is plausible (at least using current and near-term technology).

# Macro Explanation

The acronym, **AI**, commonly stands for **Artificial Intelligence**.

**Artificial Intelligence** is a label which refers to any **Synthetic Behaviour** in either the physical or digital realms, which is not natively biological in origin.

**Synthetic Behaviour** can only currently be enabled via the combination of some form of **Mechanical Processor** and a **Digital or Robotic Platform** which the processor directly controls.

+**Mechanical Processors** – Currently consist of the following classes:

+**Analogue Processor** – Shaped physical components moved by a form of energy, in order to follow a program of **Computation Instructions**, which are themselves also built from shaped physical components; there will also be some form of input data, which is usually also in the form of shaped physical components… An example is a lock and key, used to secure a property.

+**Digital Processor** – Formed from elemental composite materials, that use electricity in order to open and close nanometre-sized **Binary Gates**, on a wafer chip consisting of usually a lot more than thousands of gates; the gates follow a program of **Computation Instructions**, which are held in memory which is also based on a massively gated elemental composite wafer chip {although is instead often capacitor or magnetic in nature}; there will also be some form of input data, which is also usually in the form of massively gated elemental composite wafer chips {although is instead often capacitor or magnetic in nature}… An example is the device you are possibly using to read this explanation (if it’s not on printed out sheets of paper).

+**Binary Gates** – Allow for two positions to be assumed… Open or closed… Which correspond to either the number one, or the number zero, respectively. These ‘ones’ and ‘zeros’ can then be used as **Computation Instructions**. (I encapsulate with apostrophes, because one and zero are human understandable interpretations of an open or closed gate, which computers don’t actually appreciate… They just chug through millions to quadrillions of them per second, simultaneously opening and closing likewise numbers of gates).

+**Computation Instructions** are sets of binary bits (that is, chains of successive ‘ones’ and ‘zeros’) which load data into working compute memory at initiation of an executable programme, and during the execution of the executable programme, by acting as addresses corresponding to some sort of high-performance rapid memory module (usually cache or RAM), that the processor understands and is compatible with; and then when actual computation chains of successive ‘ones’ and ‘zeros’ are reached by the active processor core, the processor passes along each step (steps being represented by a specific processor designed size of binary bits {the successive ‘ones’ and zeros in the chain}) and performs vast amounts of a small set of possible transformations to binary bits in the working compute memory; for example you could have 0001 (which in our example corresponds to a binary byte symbol for the number one) in address #0005 through #0008, and also you could have 0010 (which in our example corresponds to the binary byte symbol for the number two) in address #0009 through #0012; and when the processor reaches the first compute step in the instruction set (lets just say represented by 0000 which is in the case of this example telling the processor to use an add operand on the two next steps {the next two binary 4-bit steps… in our example holding the values 0101 [step offset 1] and 1001 which represents the index number 9 [step offset 2]}, which are actually treated by processor design as memory locations), then the processor will perform the necessary arithmetic to transform the number one represented in memory locations #0005 through #0008 and the number two represented in memory locations #0009 through #0012, in order to return the human readable number 3 to the user (the return command being in computation step four {**Computation Instruction** binary bit 13 through 16 [the fourth byte if you like in our example processor and instruction set, which could be represented by a value 1111]}). This is likely to cause all sorts of Catholic glee for some, and all sorts of anti-Christian uproar for others… Regardless, instruction sets ultimately allow a machine to learn i\_e\_ the scientific domain of **ML**, which I hasten to add, actually has far reaching ramifications outside of science, just waiting to be explored, discovered, experienced, and promoted or restricted!

+**Capacitor Memory** – Based on the coupling of many transistor-capacitor pairs (memory cells) in an integrated circuit. Each pair acts as a store for a single binary bit. The transistor has three functions: 1. To read the memory cell i\_e\_ the state of the capacitor (full {one} or empty {zero}); 2. To rapidly fill the memory cell capacitor with electrons; 3. To rapidly empty the memory cell capacitor of electrons. Please also be aware, that the capacitor leaks relatively rapidly (many times per second if it keeps getting refilled with electrons each time). As such there is a high-frequency of background memory-controller recharge of full capacitors, usually thousands of times per second.

+**Magnetic Memory** – Based on dipole switching, instead of **Binary Gates**. Data is read by sensing the resistance of dipole storage elements. Data is written, by inducing a nanometre-localised strong electrical field corresponding to the orientation desired in the dipole storage element.

+**Magnetic Processor** – I won’t discuss them much yet, because it is very early days, and I suspect that all current form-factors will be obsolete well before magnetic processors reach mass-market... All I will say, is that they are currently mainly used as a form of medium-term, high-performance data storage. This is only a small part of their computation usefulness for mankind in my opinion.

+**Optical Processor** – I won’t discuss them much yet, because it is very early days, and I suspect that all current form-factors will be obsolete well before optical processors reach mass-market... All I will say, is that they are currently seemingly ignored, in comparison to how vital they are to the next generation of computation for AI. You see, most of the digital **Machine Learning** algorithms are best suited for optical computation. The reason, is that optical processors are magnitudes better than digital processors in terms of energy efficiency, and also magnitudes better than digital processors in terms of size efficiency. The current problem is caused, by ‘intentional’ vested-interest or ignorance-based avoidance, of funding for the research and development activities related to optical processors. This is a massive shame for the technological future of mankind this century!

+**Quantum Processor** – I won’t discuss them much yet, because it is very early days, and I suspect that all current form-factors will be obsolete well before quantum processors reach mass-market... All I will say, is that they are basically souped-up digital processors, which can compute based on usually sub-nanometre-sized **Particle Distributions**… As opposed to the digital processor, which is based ‘merely’ on **Binary Gates**.

**Particle Distributions** – Allow for many (an almost infinite number) of pseudo gate positions to be assumed… Basically it works like a pseudo **Binary Gate** which can partially open or close, to any degree in between fully open or fully closed… These partial gate positions correspond to fractional numbers between and including one and zero… But please remember that the computation is not actually in the form-factor of a gate, but rather, it is in the form-factor of collections of sub-atomic particles such as electrons, protons, or ions, which at rest, are travelling so fast within a specific computation chamber, that the resolution of their position, velocity, trajectory, and spin, forms a distribution, when measured over a successive number of measurement instances. These computation chambers can compute, because they are initialised to a specific state preceding each measurement, and the particles involved in the computation chamber probabilistically undergo extremely similar, if not exactly the same interactions each time, which leads to the statistically significant repetition of a particular position, velocity, trajectory, and spin distribution, over a collection of measurements… Try that one on for size O Einstein!

+**Platform** – At least one software executable file with its constituent variable files (usually at least an environmental variables file, an initiation variables file, and a global variables file) {this is the potential of the system}; along with its collateral initiation data, operational data, and termination data {this is the cause of the system}; and finally, a native structure composed of at least a compute core with networking ports, but usually a lot more, like rapid access memory, long-term storage, mainframe (**Digital** {physically stationary but digitally mobile}), androidframe (**Robotic** {both physically and digitally mobile}), sensory devices, and actuator devices {this is the effect of the system}.

# Micro Explanation

The acronym, **ML**, commonly stands for **Machine Learning**.

**Machine Learning** is a label which refers to any computer algorithm that can be computed in order to **Train** on input data samples, and thereby create a **Model Of Patterns** within the aggregate input **Latent-Space** range of all the combined input data samples, which results in the storage of **Relationship Representations** of each input data sample to at least one **Supervised**, **Semi-Supervised**, or **Unsupervised** output **Classification**.

+**Train/Training** – The computation of sets of averages in order to describe aggregate characteristics of sets of input data samples… For example, a trendline which plots the average across variable Y, as variable X increases.

+**Model Of Patterns** – Stored descriptions of the **Relationship Representations** between aggregate characteristics of variables; these models consist of **Graphs Of Correlation Nodes**… The difference between this and **Traditional Correlation Analysis** (which is pretty much obsolete, because it is a less efficient and less effective version of graphs of correlation nodes {being merely a subset thereof}), is that here groups of variables can be described as having aggregate effects on other groups of variables, in a **Non-Linear** fashion… For example, if variable Y (beers drunk) is currently going up by 20% per hour, but will peak and come down after sunrise and if variable Z (women present) goes up by 40% directly after this, then variable A (Michael’s ‘botheration’) will skyrocket by 99% well before he has fully digested breakfast.

+**Graphs Of Correlation Nodes** span at least two, although commonly tens or hundreds of variables, which are systematically constructed, often with a threshold. Each node describes the relationship between hierarchically adjacent nodes, with a trendline of numbers built from averages, which is the ‘computer understood’ description of how the output variable changes as the input variable changes.

+**Traditional Correlation Analysis** – Allows for the modelling of **Linear** variable pairs only.

+ **Non-Linear** – Is a form of pattern-recognition which can accommodate the description of the more complex, usually non-symmetrical, data relationships that are often curved and/or choppy in profile.

+**Latent-Space** – This is bounded by the greatest and least limit values within which the **Model Of Patterns** describes data input sample **Relationship Representations**... Essentially, your model can forecast **Output Classifications** for any **Inference** data sample which is previously seen or unseen in the training input data samples and their characteristics and associated relationships, to a specific **Degree Of Significance**.

+**Inference** – Executing a similarity algorithm which checks Latent-Space for any reasonably close pattern matches for the new inference data sample, in order to obtain a close or average output **Classification**.

+**Degree Of Significance** – Indicates how much a human should trust the outputs of a given model… Usually on a scale of 0.000001% (6-Sigma Or Loose : Never/Always trust, depending on the **Polarity Of the Decision**) to 100% (6-Sigma Or Loose : Never/Always trust, depending on the **Polarity Of the Decision**).

**Polarity Of the Decision** – For example, if your decision is choosing between investing and divesting, and your model tells you that the price of your asset in question will be going up and so is profitable to invest in, but the model is based on a **Number Of Input Data Samples** which only give the suggestion a **Degree Of Significance** of 0.00001% (Lean), then you clearly want to think at least twice before following its suggestions… Conversely if your model tells you that war with China will happen this century, but this time the model is based on a **Number Of Input Data Samples** which actually give the suggestion a **Degree Of Significance** of 80%, then you clearly want to immediately and massively increase your military budget, and then also probably inaugurate specific, frequent, and ongoing homeland **Preparatory Taskforces**, **Diplomatic Missions**, and even **Political Meetings**, with China and its peers, in order to try and avoid the worst case scenario, and to also try and delay anything that is clearly unavoidable!

+ **Supervised** – Humans or animals {or of course angels or even God himself… would be kind of weird, but ‘could’ in fact easily happen at many points in the past/future!} have provided the **Relationship Representations** between training input data sample and output **Classification**.

+**Semi-Supervised** – Pre-existing person-originated output classifications have been combined with similarity matching algorithms in order to automatically generate richer classification labels {and some times even the only classification labels} for new **Relationship Representations** between training input data sample and output **Classification**.

+**Unsupervised** – This is child’s play… Go ask #AOC or something!

+**Classification** – Ultimately this process of **Machine Learning** allows for vast numbers of **Inferences** to be elicited, which return at least one value, at the moment only either textual or numerical, corresponding to the **Inference** data sample’s most similar peers within the **Training** input data samples.

Once you have a **Classification** automatically generated from a **Machine Learning** algorithm, this can be used to carry out future computation steps which for example would move the connected **Digital Or Robotic Platforms**… Example a **Robot** moving from the charging pad to the loading pad in a warehouse; or a **Bot-Trader** automatically placing thousands of buy and sell commands on varying amounts of **Obsidian** stock on an equities exchange in the not too distant future, all within the course of a typical ‘casual’ trading day… However, given current advancement levels in AI, classifications seem to either be ignored, because there isn’t the funding to do anything about them, or they are simply fed back into more and more AI systems, leading to catastrophic pollution of publicly available data (which we really need to avoid going forward by the way).

Nota Bene 0002: Please be aware that some AI academic, and even corporate, incumbents will likely be upset with me, because I have omitted to mention their darling catchphrase : ‘**Neural Nets**’. This is water of my camels back, because in my opinion it is a highly misleading term which is poorly understood in the AI community… The fundamental reason being that no digital algorithm is actually capable of even similarly computing in a way that we commonly understand to be a biological neuron computation (in fact we still don’t have more than the slightest idea how biological neurons really work, from a detailed scientific perspective)… I mean, as an analogy, it would be like someone saying that ‘Dunking Nets’ work in the same way as ‘Fishing Nets’… Anyone with any sense can immediately work out that if used a fishing net in a basketball game, in place of a dunking net, then neither team would ever be able to score points, and if you use a dunking net instead of a fishing net in order to fish, then your likelihood of catching any fish, would be rather uncomfortably close to zero!