

# Machine Learning: Financial Data Takes a New Dimension

For professional investors only,  
Promotional document

**ram**  
ACTIVE INVESTMENTS



# Table of Contents

<b>Executive Summary</b>	<b>5</b>
<b>1 What's driving the growing popularity of machine learning ("ML")?</b>	
a. This current AI boom is rooted in research from the 1950s	6 – 8
b. The phenomenal acceleration of ML techniques is certainly no fad	9 – 11
<b>2 An Overview – So what is ML ?</b>	
a. The Principle : train the machine to recognise or predict	12
b. Various ML models for various objectives and applications	12 – 13
c. Adapting the models to abstract financial data	13
d. The common denominator for these techniques is the use of a training procedure to calibrate their parameters	14
e. But an ML construction process cannot be reduced to a single training procedure	14 – 15
<b>3 Why do we use ML techniques for financial data integration ?</b>	
a. ML is well adapted to high-dimensional and evolutive datasets	16
b. ML models can capture non-linear interactions	17 – 18
c. ML frameworks are powerful enough to handle unstructured datasets	19
d. ML frameworks also allow to prevent overfitting and interpret results	19
<b>4 How do we control the machine ?</b>	
a. Garbage in, Garbage out : Prepare the right learning material	20
b. Lost in higher dimension ? Select input features and model dimensions carefully	20 – 23
c. Generalisation : Train with care, make sure you are not training on "noise"	23 – 25
d. Understand what you are teaching : Establish the right objectives and focus on interpretability	26 – 29
<b>Conclusion</b>	<b>30</b>



# Executive Summary

Artificial intelligence (“AI”) has arguably become the most significant and disruptive general-purpose technology in recent years. Techniques like machine learning (“ML”) have taken giant strides forward; empowering computers and enabling them to learn and build models so that they can perform prediction across different domains. This technology is particularly pertinent to the portfolio management industry, where the integration of information from numerous, large – and often unstructured – datasets has become a key factor in determining success.

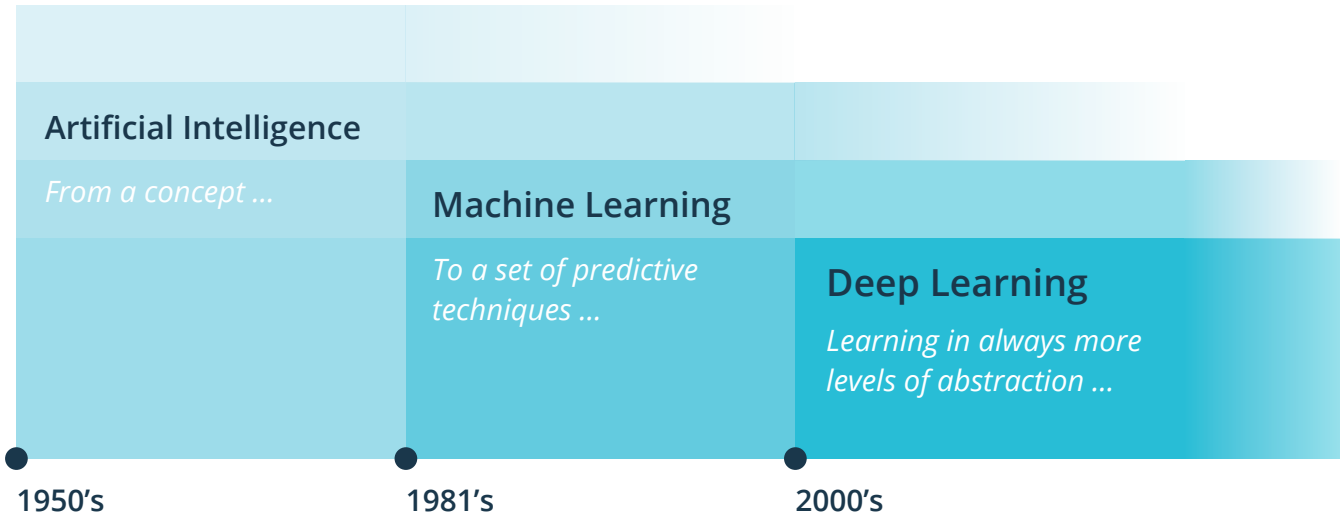
ML development frameworks accessible to the research community now offer solutions to both handle these datasets and interpret results, controlling how inferences are derived. We believe that what we teach the machine is only as important as the limits and constraints we set during its learning process, unlocking interpretability and generalisation.

In this paper we will look at the growing popularity of artificial intelligence and the potential for finance to prosper from machine learning implementations. As technology continues to push the boundaries of our imagination, new dimensions will undoubtedly emerge over time.



# What's driving the growing popularity of machine learning?

## a. This current AI boom is rooted in research from the 1950s



### AI emerges as a challenge for years ahead

Alan Turing, who is now regarded as the father of AI, creates the "Turing Test" to determine whether a computer is "able to think" (i.e. "imitate humans").

The machine is said to have passed the test if a human evaluator cannot reliably distinguish between the human and the machine when hearing a natural language conversation between them. Although the test cannot be used to evaluate all intelligent systems, it rationalises the debate as to whether machines can "think".

Even today the Turing Test is still an area of focus for big technology companies (e.g. Google's Duplex technology, Alibaba copywriting tool, IBM Project Debater, etc.), and several attempts have participated in the annual Loebner Prize competition, which elects the program that demonstrates the «most human» conversational behaviour, and which has only awarded bronze medals so far...

Technology heavyweights are leading the AI boom



Source: RAM Active Investments ("RAM AI")

1951



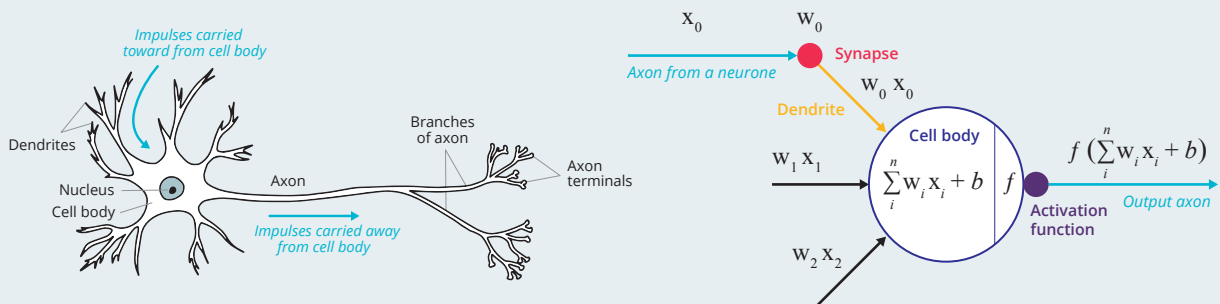
## Creation of the first artificial neural networks (“ANNs”) by cognitive scientist Marvin Minsky

The network is made of 40 neurons passing signals to each other through synapses. The synapses’ strengths are determined using a technique inspired from a neuroscientific theory called the “Hebbian Learning principle” (Donald Hebb, 1949). This principle attempts to explain synaptic plasticity, i.e. the adaptation of brain neurons during the learning process (simultaneous activation of cells leads to prominent increases in synaptic strength between those cells).

### Artificial Neural Networks (“ANNs”)

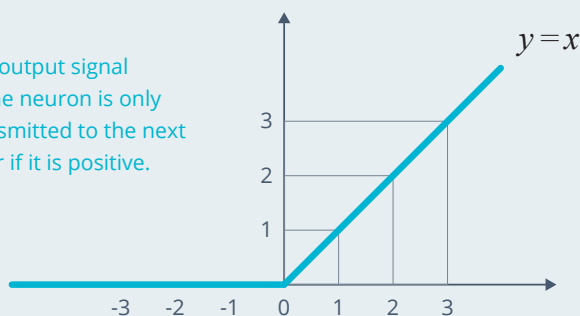
The flexible architecture of ANNs makes them powerful tools for solving a wide range of prediction-based problems. Each neuron aggregates several input signals, sending a signal to the next layer of neurons, but only if that signal is strong enough (this is what we call “activation”). Activation functions are non-linear, which allow the ANNs, layer after layer, to model complex representations. The most frequently used activation function is RELU.

#### Schematic analogies between biological and artificial neurons <sup>1</sup>



#### RELU activation function: $y = \max(0, x)$

The output signal of the neuron is only transmitted to the next layer if it is positive.



*“A single neuron in the brain is an incredibly complex machine that even today we don’t understand. A single “neuron” in a neural network is an incredibly simple mathematical function that captures a minuscule fraction of the complexity of a biological neuron. So, to say neural networks mimic the brain, that is true at the level of loose inspiration, but really artificial neural networks are nothing like what the biological brain does.”*

Andrew Ng, Google Brain founder

<sup>1</sup> Source : Stanford University. “A cartoon drawing of a biological neuron (left), and its mathematical model (right). 2017, (Electronic image). Available: <http://cs231n.github.io/neural-networks-1/>

1965



## The very start of deep learning

Mathematician Ivakhnenko creates the first working deep learning networks. They can be considered as a natural generalisation of ANNs, with several layers of neurons.

Ivakhnenko’s work paves the way for the development of Deep Learning (“DL”) techniques. “Deep” networks involve several hidden layers of neurons, or more generally, several stacks of parametrised processing.



1982

## ML models can now be trained efficiently ... ... at least from a theoretical standpoint

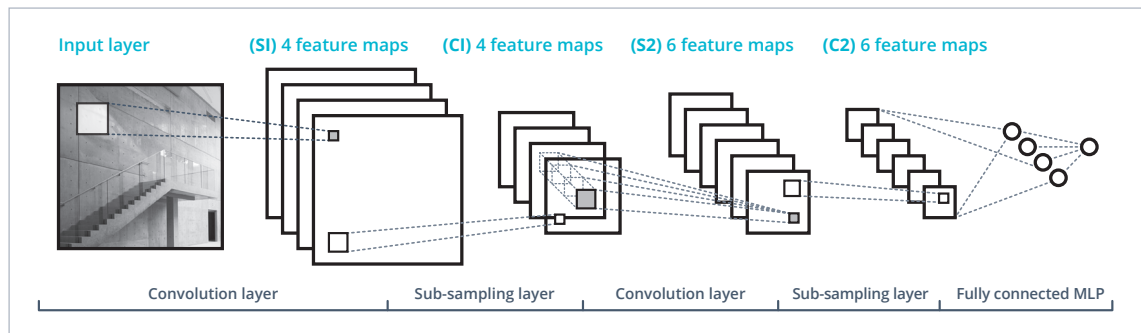
Scientist Paul Werbos employs an efficient optimisation procedure to train ANNs, the backpropagation of errors. "Machine learning" starts making sense and becomes a prominent area of research.

1982

## Learning getting deeper ...

Computer scientist LeCun, who is now Chief AI Scientist at Facebook AI Research, applies backpropagation to deep ANNs called "convolutional neural networks" ("CNNs").

Image representing a convolutional neural network for image recognition, each layer introduces a higher level of abstraction aimed at extracting semantics.



Source: LISA lab., "The Full Model: LeNet," 2013, [Electronic image]

1997

## ... and deeper

Computer scientist Schmidhuber, a pioneer in the field of deep learning models with memory, introduces the "long short-term memory" ("LSTM") network as a Recurrent Neural Network able to learn long-term relations between input features, which makes them very powerful for time series.

2015  
to ...

## AI programs beat humans in a series of game

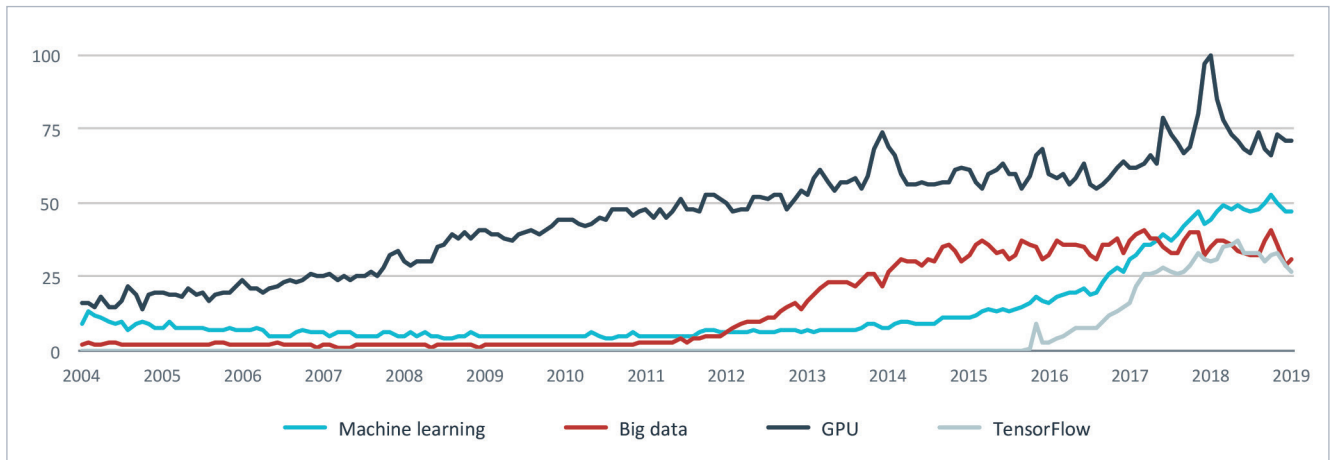
- Google's DeepMind uses reinforcement learning to beat champions at "perfect information games" such as go and chess. They have trained deep convolutional neural networks solely by reinforcement learning, a trial-and-error learning principle where the algorithm creates its own pathway to success from self-play games.
- Libratus, a software created by a group of computer scientists, defeated four top-class poker players at two-player Texas Hold'em poker, an imperfect-information game, with 99.98% statistical significance. This is the first time AI has beaten a champion in this game.



## b. The phenomenal acceleration of ML techniques is certainly no fad

Web-search data from Google Trends provides an interesting insight into what's driving the recent enthusiasm for machine learning techniques.

*Google Trends : Worldwide Interest Study*



Source: Google Trends

### ● Data, data, everywhere

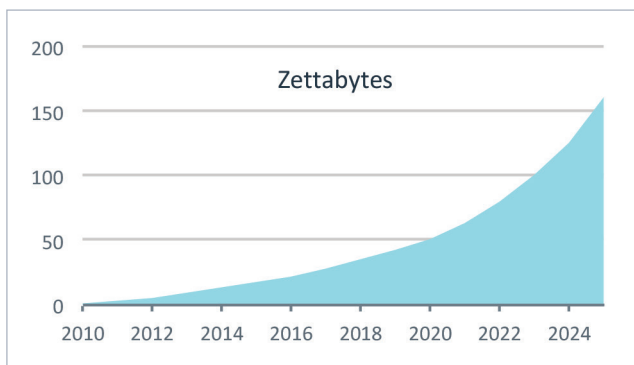
Digital data has seen exponential growth since the early 2000s. This growth has occurred across both structured data (highly organised contents such as relational databases) and, more importantly, unstructured data (such as emails, files, etc.). Unstructured data now represents roughly 80% of the total data but is harder to analyse. Most of this data is generated automatically by mobile devices and computers (e.g. social network platforms) but also by sensors; a modern car has more than 100 sensors!

The International Data Corporation (“IDC”) forecasts that by 2025 the global datasphere will grow to 163 zettabytes (i.e. 163 trillion gigabytes)<sup>1</sup>. The amount of data generated every day should be around 50 billion gigabytes, which is ten times the amount created between the dawn of civilisation and 2003 – this amount was estimated at

around 5 billion gigabytes by Hal Varian, Chief Economist at Google.

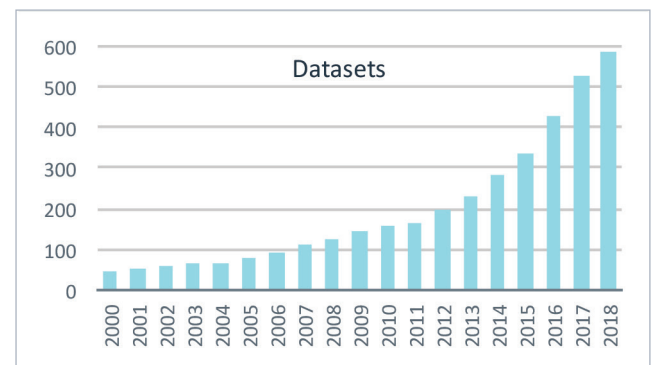
As investor focus has turned to higher frequency and alternative data sources, financial data has witnessed tremendous growth, enabling investors to take more informed decisions. In addition to traditional data sources (market prices, volumes, financial statements, analysts’ estimates, etc.), investors now have access to a wider range of alternative datasets enabling far greater insight. This alternative data comes in the form of more-or-less-structured datasets such as ESG indicators, management transactions, sentiment-driven data based on newsfeed and social media, earnings call transcripts, web-searches and web-traffic data, credit card, and – generally unstructured – sensor data records (e.g. geolocation, satellite, weather, etc.).

*Data Created Over Time*



Source: IDC’s Data Age 2025 study, sponsored by Seagate, April 2017

*Number of Alternative Datasets by Launch Date*



Source: Neudata & RAM AI

<sup>1</sup> Reference: <https://www.seagate.com/files/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf>



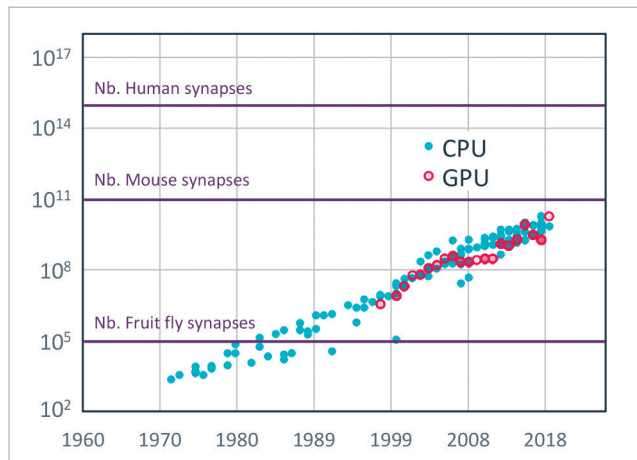
## Computer hardware advances have helped fuel the AI revolution

The research community now has access to significant processing power ...

The machine learning boom has benefited from the steady cheapening of hardware and developments achieved in recent years – sometimes to serve other purposes. Graphics Processing Units (GPUs), which were originally designed for rendering graphics for game players, can now be used for general purpose calculations, just like Central Processing Units (CPUs). This has helped to dramatically accelerate the training of machine learning algorithms.

Indeed, modern GPUs contain thousands of processing cores, many more than CPUs; allowing them to execute thousands of operations in parallel during one clock cycle. However, GPUs generally run at a lower clock speed than CPUs. For example, the NVIDIA GTX 1080Ti 11GB Graphics Controller contains 3'584 physical cores and 12bn transistors, whereas the Core Intel i9-7900X 3.3GHz processor has 10 physical cores and probably a few billion transistors, (although this number is undisclosed).

### Number of Transistors per CPU / GPU



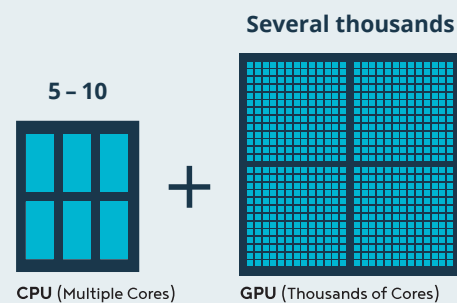
Source: Neudata & RAM AI

Similarly, the storage capacity of the various random-access memory (RAM) modules – used to store input data, temporary data, weights and activation parameters of ML algorithms – has increased steadily. These modules are essential in optimizing computing power: SRAM (Static RAM) is a low-density, fast storage solution which is generally embedded within processing units in order to feed them with the immediate data they need, while cheaper/slower DRAM (Dynamic RAM) are used as main memory to store less immediate data used in ML algorithms. These have also cheapened significantly!

### At the core : CPU vs GPU

The number of transistors on an integrated circuit (transistor count) continues to grow at a pace, albeit slightly slower than Moore's law (according to which it should double approximately every two or two and a half years).

These transistors are used both in processing and storage units. The chart below illustrates the number of transistors per processor.



Source: RAM AI

The number of transistors in modern processing units is fast approaching the number of synapses in a mouse's brain !

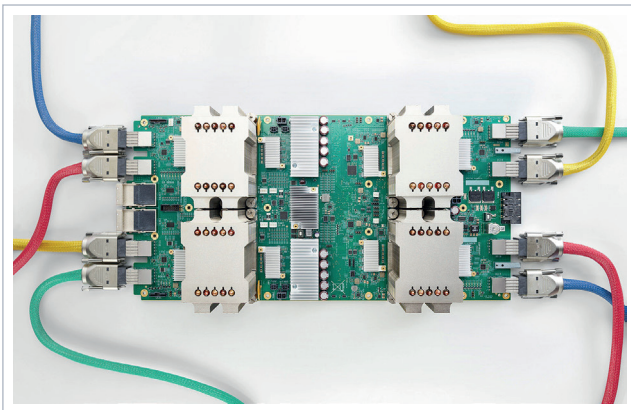
### ... And both hardware and cloud providers have joined the race !

GPU usage has been made possible through specific Application Programming Interfaces (APIs). APIs are a software intermediary that allows two applications to talk to one another. NVIDIA was the first GPU maker to propose their proprietary API called CUDA. NVIDIA have taken a quasi-monopoly (for now) – on the general-purpose GPU acceleration market. NVIDIA's dominant position has even enabled them to change the terms of their software licensing agreements. This change prohibits the use of consumer-grade (GeForce) cards by firms in data centers, forcing firms to utilise the more expensive institutional-grade (Tesla) cards.

In parallel, the world's major players of cloud computing (Google, Amazon and Microsoft) have seemingly joined a race towards offering ever-greater processing power on their cloud platforms –ensuring they stay current – to rival the best GPU infrastructures. Google was the first to provide access to NVIDIA's GPUs (P100), but Amazon responded by offering access to the more powerful Tesla Volta GPUs (V100-backed P3 instances that include up to eight GPUs connected by NVLink).

In February 2018, Google offered access to Tensor Processing Units (TPUs) through their cloud platform. TPUs are a form of application-specific integrated circuit (ASIC). Google designed these units specifically to optimise the use of their ML development framework, TensorFlow, and enabled the use of Compute Engine virtual machines to combine the processing power of several units. While Google's first-generation TPUs were designed to run only pre-trained ML models, their second-generation Cloud TPUs are more powerful and designed to accelerate both the training and the running of ML models. Each v2 TPU provides 180 teraflops (floating-point operations per second) of ML acceleration, which is the equivalent of 15 consumer-grade graphic cards such as the NVIDIA GTX 1080Ti. The latest water cooled V3 TPUs reach 420 teraflops, are equivalent to the recent NVidia DGX-Station (which delivers 500 teraflops for a purchase price of roughly \$50'000).

### Google's Cloud TPU2

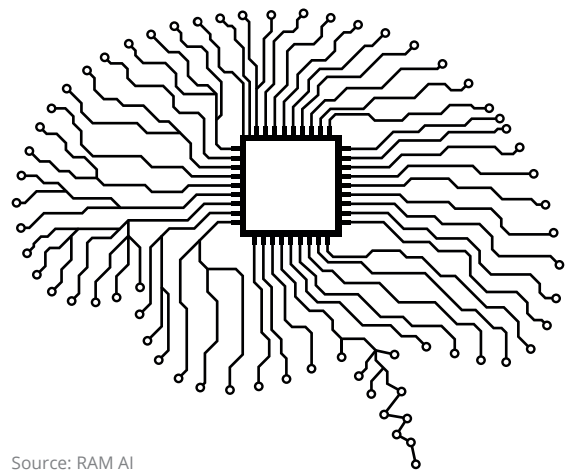


Source: Google

## ● The AI boom has also precipitated an open source software revolution

The bright outlook for AI applications has encouraged large tech corporations like Google, Facebook, Amazon and Microsoft to invest heavily in the field of machine learning. These names have committed resources to create ML development frameworks for their own needs but have also been enthusiastic in releasing them “open-source” to the research community. Popular ML frameworks include; PyTorch (released by Facebook in early 2017), TensorFlow (open-sourced by Google Brain in 2015) and MXNet (main backers being Amazon and Microsoft). These names have certainly contributed in developing a collaborative and active community, made up of researchers, practitioners and machine learning enthusiasts, and to the popularity of data science competition platforms like Kaggle.

These frameworks consist of a high-level interpreted language and are built upon a fast and low-level compiled backend to access computation devices including GPUs. They also generally offer data input/output solutions and convenient ways of modelling the building blocks of the network, commonly called “tensors”, as well as performing operations on them; tensors are used to encode the signal to process, but also the internal states and parameters of the ML algorithms.



Source: RAM AI

*“Every 5 years, computers are getting 10 times cheaper!”*

Schmidhuber

PYTORCH

TensorFlow

mxnet



# 2

## An Overview – So, what is ML ?

### a. The Principle : train the machine to recognise or predict

ML has often been defined as a “field of study that gives computers the ability to learn without being explicitly programmed” (Arthur Samuel, 1959). The general objective of ML is to capture regularity in data in order to make inferences or recognise certain patterns.

The principle of ML is that we let the machine learn from a dataset called the “training dataset”, but as the machine learns a new step to improve itself through training, we confront it with data it hasn’t seen before; the “validation dataset” which ensures each step forward is a tangible improvement. ML is basically an optimisation that corrects the overfitting biases the machine would take, with a “dumb” optimisation approach. We like to see it as the scientific concept of “trial and error”, through which science progresses, ultimately being communicated to the machine.

The concepts being well integrated, a well-worn adage of ML is “the more data, the better!”. The more the machine is confronted with different data samples and conditions as it learns, the better it will be with its predictions.

Indeed, many applications such as image recognition and language translation require the automatic extraction of “refined” information from raw signals.

In the context of finance, most references to AI techniques generally refer to a predictive ML model; the terms ML and AI are often used interchangeably and stand for a collection of statistical techniques ranging from regularised linear regression to Neural Networks and Deep Learning.

### b. Various ML models for various objectives and applications

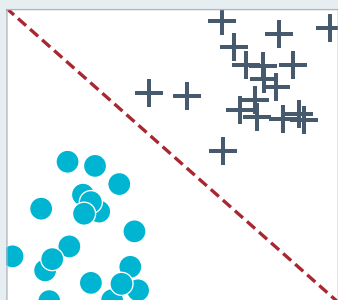
Machine learning can be reduced to two primary categories; supervised and unsupervised learning. The objective of both approaches is to make inferences from input variables, however :

#### ● Supervised learning ...

... seeks to map these input variables to an output variable; thus it requires training data to be labelled, i.e. to have, in addition to other features, a parameter of interest that can serve as the “correct” answer. Supervised learning is well-suited to :

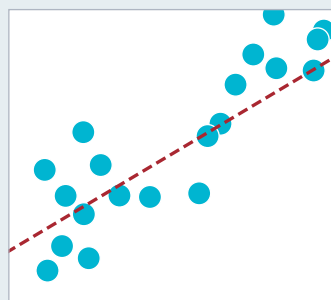
#### Classification problems (when the labels are discrete) :

- Prediction of top vs. bottom performing stocks
- Object recognition
- Cancer detection
- Credit card fraud detection



#### Regression problems (when the labels are continuous) :

- Customer satisfaction
- Macroeconomic forecasts
- Epidemiology



## ● Unsupervised learning ...

... seeks to describe the structure of unlabelled data. It is called “unsupervised” as there are no correct answers and no supervisor (teacher). For example :



**Netflix's** market segmentation through clustering instead of geographic, is to discover the inherent groupings in the data. Netflix observed that geography, gender, age and other demographics are not significantly indicative of the content a given user might enjoy.



**Amazon's** item-to-item collaborative filtering aims at discovering association rules that describes large portions of input data, such as customers who buy a given product may also tend to buy another related product.



**Google's** PageRank algorithm based on network analysis looks to discover the importance and the role of nodes in the network and measures the importance of website pages.

### c. Adapting the models to abstract financial data

Unsupervised learning can be applied to dimensionality reduction or feature selection on an input dataset such as the time series of standard or alternative factors across a universe of stocks; to find a smaller set of variables (either a subset of the observed variables or a set of derived variables) that captures the essential variations or patterns. The process can be dependent on the original learning task when it is known (e.g. predicting stock price, volatility or volume), or not (e.g. auto-encoders as non-linear alternative to PCAs for dimensionality reduction).

Supervised learning can be used for stock selection, the objective being to predict future stocks' behaviours based on an historical dataset of inefficiencies. These expected

behaviours can be either continuous (a function of expected return and risk, for example), or discrete (will a given stock outperform or underperform a given benchmark, etc.).

The recent development of ML frameworks for deep learning models also allows the efficient capturing of patterns in a time series. For example, convolutional neural networks (“CNNs”) commonly applied to image recognition, can also capture financial time series dynamics. Similarly, long short-term memory (“LSTM”) networks and other types of RNN (recurrent neural networks) can help capture events with long-term memory, corresponding to market corrections for example.

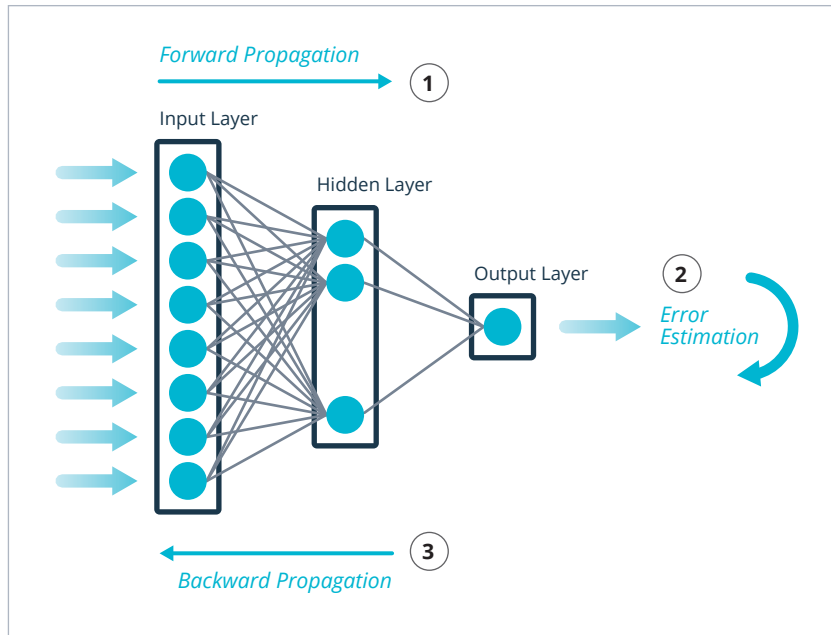


## d. The common denominator ? A training procedure to calibrate the parameters

This is similar to biological systems for which the model (e.g. brain structure) is DNA-encoded, and parameters (e.g. synaptic weights) are tuned through experiences. The most common training procedure is the “gradient descent optimisation”, which allows to estimate optimal model parameters by performing several training steps. At each step, the backpropagation reassesses the

sensitivity (gradient) of the prediction errors (between predictions and the actual values to predict) to changes in model parameters, in order to update model parameters and help them converge towards an optimum. The size of the shift in model parameters at each step is determined by a hyper-parameter known as the learning rate.

### Typical Training Procedure



Source: RAM AI

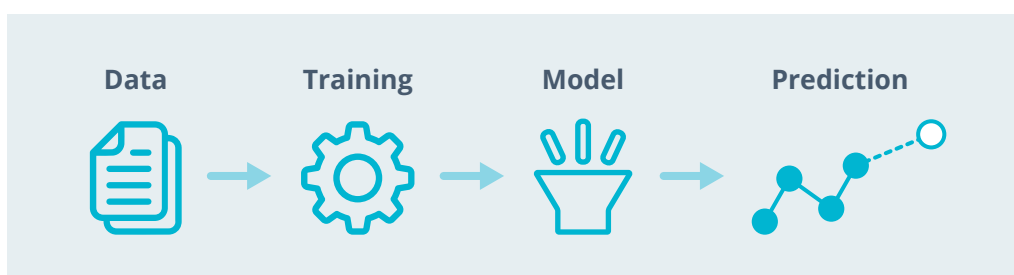
### Training Process

- ① **Forward Propagation**  
Input of sample data into the input layer  
Setting of the initial parameters
  - ② **Error Estimation**  
Compute losses  
(i.e. difference between prediction and realised)
  - ③ **Backward Propagation**  
Update parameters
- Repeat process until convergence

## e. But an ML construction process cannot be reduced to a single training procedure !

These models are generally complex and involve an additional set of parameters called hyper-parameters, which contribute to specify the model and define the way it trains :

- The network structure (e.g. number of neurons and hidden layers in a neural network)
- The learning rate
- The regularisation parameters, which allow to ensure the models extract regularity as opposed to noise
- The number of iterations



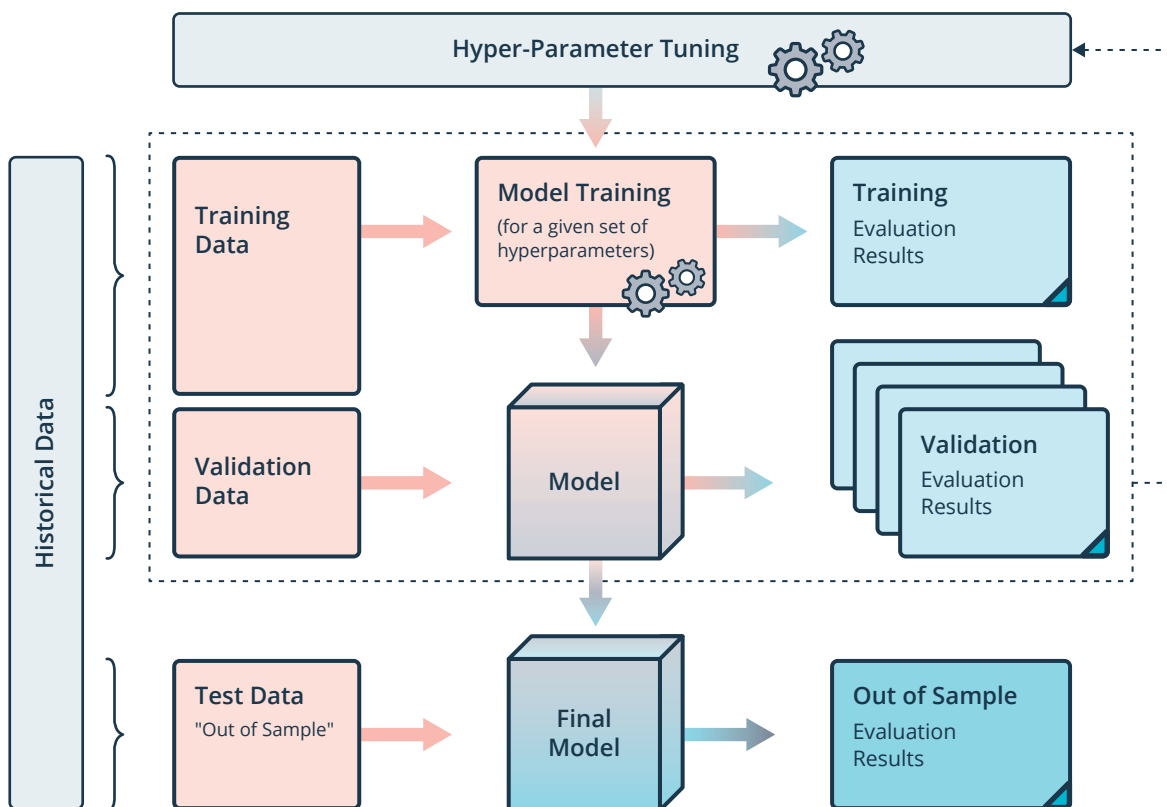
Source: RAM AI

These hyper-parameters are not set during the training procedure; they must be tuned before the training, either manually or by using grid search or Bayesian optimisation techniques. The picture below shows a typical calibration process for a ML model, in which its parameters (e.g. neuron weights) are set through training, and its hyper-parameters are set through an iterative process.

At RAM AI, we believe a robust way of deploying a ML model consists of implementing such an iterative process through a cross-validation technique, which, at each iteration (which corresponds to a full training for a given set of hyperparameters), randomly generates a training sample (sample of data used to fit the model) and an unseen validation sample (sample of data used to provide an unbiased evaluation of the model fit on the training sample).

The presence of a training set and a validation set as well as the potential use of cross-validation techniques to generate them are essential in ensuring the generalisation of the model (i.e. detecting potential overfitting). It is also key to initially isolate a test dataset, aimed at providing an unbiased evaluation of the final model. This is typical of a trial-and-error process, where the system tries to extract regularity from raw input data by performing several feedback loops.

### Typical Prototyping Phase of a ML Model



Source: RAM AI



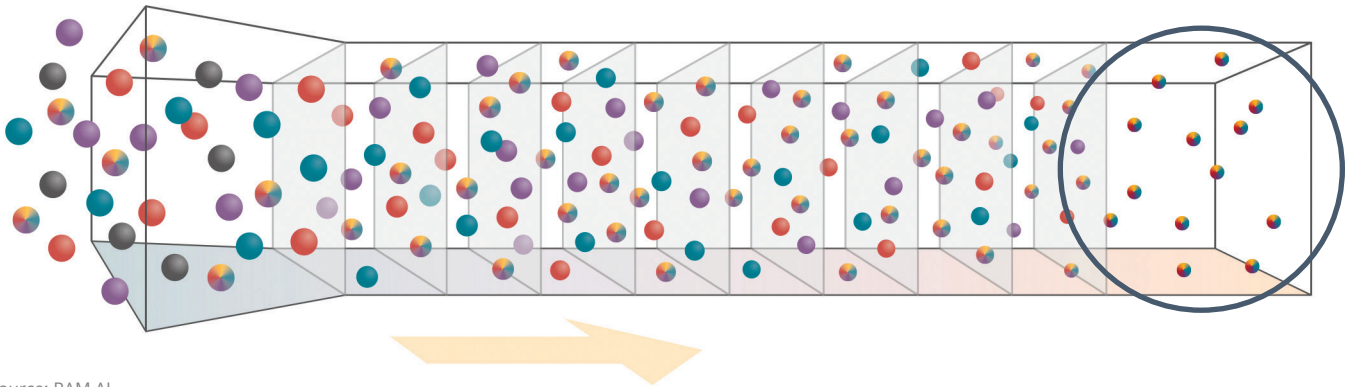
# 3

## Why do we use ML techniques for financial data integration ?

### a. ML is well adapted to high-dimensional and evolutive datasets

At RAM AI, our systematic stock picking methodology seeks to capture persistent inefficiencies across a given universe of stocks. To do so, we need to abstract data from financial datasets such as time series of stocks' features (i.e. "factors"). These factors can be classified as "traditional" (value, low-risk, momentum, quality, etc.) or more "alternative" (ESG indicators, management transactions, sentiment data from newsfeeds and social media, etc.). High-dimensional data processing allows us to make informed predictions, simultaneously integrating information from across all these factors.

ML techniques help to capture stocks that may not be a pure Value, Low-Risk, Growth Momentum profile, but are highly alpha generative being great by all standards.



Source: RAM AI

In addition to the increasing number of variables, we also must treat a large amount of data points across several geographical regions, long-term time horizons and various frequencies (some of our datasets have more than 20 years of daily history).

Machine learning models can be seen as a natural generalisation of data processing techniques; perfectly adapted to the high dimensionality of our datasets, combined with the flexibility they require :

- ML development frameworks are optimised to manipulate tensors (i.e. multi-dimension arrays), and benefit from the high processing power offered by CPUs, GPUs and TPUs (and their high-capacity RAMs).
- The optimisers embedded in these frameworks are extremely powerful to train deep learning models, they have been developed specifically to solve multi-dimensional problems (i.e. picture recognition, consumer clustering, etc.).
- The process of adding new variables to an existing dataset is very easy, and several ML techniques allow quick assessment if such variables bring new information to our stock selection process; this is convenient to test the potential added-value of alternative datasets.
- Unsupervised machine learning techniques and features selection techniques can also be used to reduce the dimensionality of the problem, if needed.

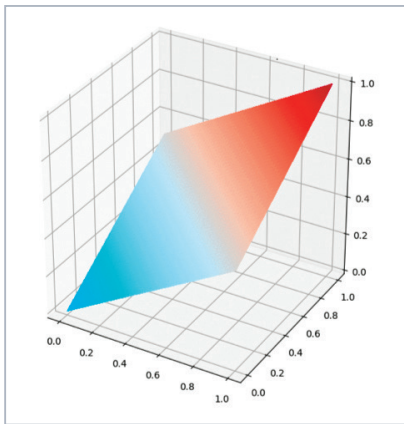


## b. ML models can capture non-linear interactions

The input factors we use are obviously not independent variables. They interact with each other in a way which is a-priori unknown, although these interactions might make intuitive sense. For example, low-quality stocks which have high earnings accruals tend to have inflated earnings, therefore lower price-earnings ratios, which makes them cheaper according to one measure of value. One way of modelling interactions in standard data

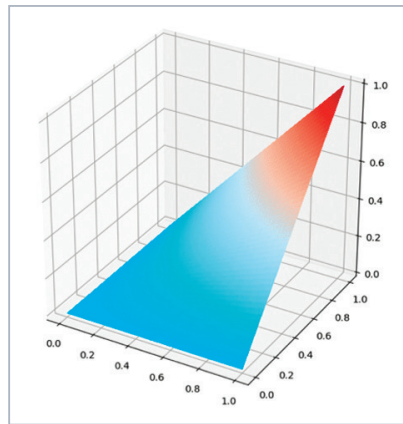
analysis, such as linear regressions, is to add several first-order interaction terms in the form of products of two variables into the input dataset. For example, charting the preference curve (y) using ranked factors for value (x) and quality (z), we can see that a name with average value and average quality ( $x=50\%$ ,  $z=50\%$ ) will score higher than an extremely cheap name of extremely bad quality ( $x=100\%$ ,  $z=0$ ).

### Modelling of Interactions Between Value & Quality



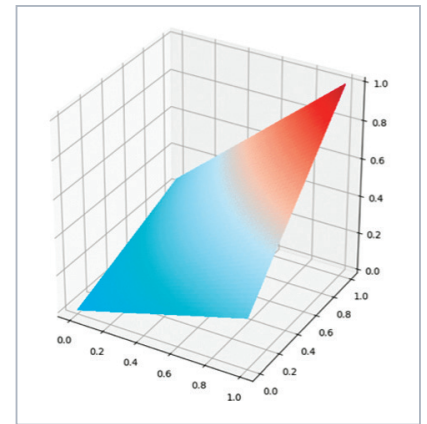
1

Ignoring interactions, an extremely cheap name but of extremely bad quality ( $x=100\%$ ,  $z=0$ ) will score as high as a name with average value and average quality ( $x=50\%$ ,  $z=50\%$ ).



2

Intuitively the bad quality name should score lower than the average name (this assumption deserves at least to be tested), therefore we add an interaction term.



3

The resulting preference curve now looks more realistic, with the bad quality name scoring lower than the average quality name.

However, we can anticipate that these interactions are unlikely to be linear, and that some higher-order interaction terms could be required in some instances. For example, the interactions between leverage (as measured by Debt/Equity) and future returns are clearly not linear: on the one hand too much debt can be a consequence of over-investment and result in high interest costs, (weighing on future returns); on the other, too little debt might dampen growth. There must be an optimal amount of debt somewhere between these two extremes. It is obviously difficult to model the structure of dependencies within our input dataset given the high diversity of input factors and non-linear causality relationships.

A copula-based technique (some form of parameterisation of the joint cumulative distribution function) could be used to model these interactions, but at the cost of high calibration complexity – and subsequently a high risk of poor generalisation.

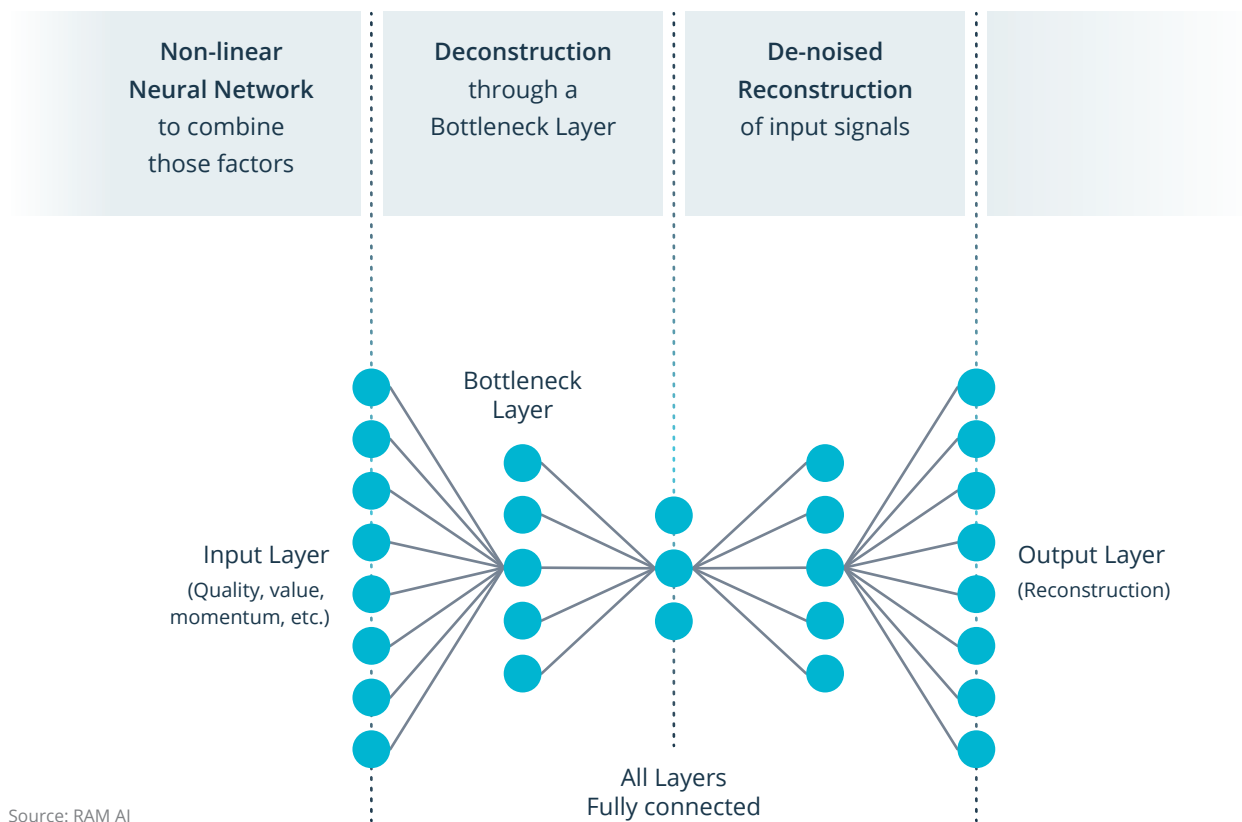
One of the advantages of ML techniques is that they require almost no assumptions on the structure of interactions. We have observed that random forests and neural networks work well to abstract these dependencies and make predictions: when calibrated accurately, they generally outperform regressions and logistic regressions with first-order interaction terms.



Machine learning techniques also provide non-linear alternatives to standard dimensionality reduction techniques such as principle component analysis (“PCA”), sometimes used to remove “noise” from input signals. We can use, for example, an unsupervised machine learning technique called auto-encoder (“AE”); AE also refers to manifold learning. We can illustrate this comparison by using an auto-encoder based on a neural network showing a bottleneck configuration, which seeks to predict the identity function.

We can use PCA and AE to reduce the dimensionality of a dataset of more than 200 factors, to a limited number of factors, thus improving the generalisation capabilities of a subsequent model. This can prove useful if the model is complex (tends to overfit, e.g. boosted trees) or if input data has many outliers.

### Noise Removal



Source: RAM AI

### c. ML frameworks are powerful enough to handle unstructured datasets

All data that does not qualify as structured is unstructured, this describes newly created data that hasn't been (or fully) explored, cleaned and processed. It is usually laden with text but may also include audio and/or visual data. In the context of finance, textual datasets include news, social media, earnings calls transcripts, etc.

Various alternative and unstructured datasets are now available to the investment community, and extracting information from them can clearly provide an edge for stock selection strategies. ML frameworks provide tools to structure these datasets and extract semantics from them.

At RAM AI, we are exploring ways to unveil the hidden information in some of these datasets, (e.g. to extract the sentiment of an earnings' call to enhance a stock selection strategy). ML development frameworks provide powerful techniques to handle these through Natural Language Processing ("NLP"), (i.e. leveraging computing power and statistics to process and clarify language in a systematic and sensible way that doesn't involve human intervention (beside coding)).

There are generally three major steps in NLP: text preprocessing, text to features and testing and refinement. Text preprocessing includes noise removal, lexicon normalisation and objective standardisation. Text to features includes syntactical parsing, entity parsing, statistical features and word embedding.

Structuring the preprocessed text through word embedding (a method of using a vector of numbers to capture different dimensions of a word) can typically transform it in a set of a few hundred numerical features (dimensions), making subsequent analysis easier!

### d. ML frameworks also allow to prevent overfitting and interpret results

The first objective when fitting a ML model must be to ensure that the model is sufficiently robust (i.e. it must generalise well on unseen data). To do so, many techniques can be used in combination: using cross-validation (i.e. isolating various training and validation samples as described previously), using large enough datasets, and using regularisation techniques. Regarding cross-validation, it is vital to compare the speed and smoothness of the convergence of the various training and validation datasets, as well as the distribution of model weights throughout the various training iterations; various visualisation tools such as Google's Tensorboard can help. ML frameworks also make it easier to use and compare several regularisation techniques. For example, the various optimisers generally embed a weight decay parameter, which allows to implement ridge regularisation (i.e. to force the weights to be smaller without being zero) during the training procedure.

At RAM AI, one aspect of the application of ML techniques that we believe should not be underestimated is interpretability (i.e. understanding how and why results

are generated). Model interpretability aims to help us understand why a machine learning model made a given decision and gain comfort that the model functions correctly (and for the right reasons). Model explainability has been an increasing area of research, especially in the last two to three years, as demonstrated by the many tutorials on model interpretability in the recent flagship conferences on machine learning. In particular, many topics were covered at the 2017 International Conference on Machine Learning (ICML) and the 2017 Neural Information Processing Systems (NIPS), including sensitivity analysis, mimic models, and hidden layers investigation.

Numerous competing packages that enhance model explainability were presented at these conferences, including the SHapley Additive exPlanations ("SHAP"). SHAP purports to be a unified approach to explain the output of any machine learning model. SHAP connects game theory with local explanations, uniting several previous methods and representing only possible consistent and locally accurate additive feature attribution methods based on expectations.



# 4

## How do we control the machine ?

### a. Garbage in, garbage out : Prepare the right learning material

At RAM AI, we have been working for over 12 years on building several factors for stock selection, which seek to capture various inefficiencies such as momentum, growth, quality, value, low-risk, etc.

Although we have designed every single factor with the aim to extract information on a given inefficiency as efficiently as possible, all these factors need to be pre-processed so that our machine learning models can infer accurate predictions from them. Preprocessing is crucial, especially for deep learning models which use neural networks, for which convergence is clearly helped when data is cleaner. Cleaning the data means working on the distribution of inputs, the potential presence of non-available data, and identifying causes for outliers, in order to potentially retreat them.

Among other techniques, we use Scikit-Learn package for these processing steps, in particular their various scalers :

**Quantile transformer :** Rank transformation in which distances between marginal outliers and inliers are shrunk.

**Gaussian transformer :** Another non-linear transformation in which distances between marginal outliers and inliers are shrunk.

**Standard Scaler :** Removes the mean and scales the data to unit variance.

**Robust scaler :** Removes the median and scales the data according to the quantile range (by default the interquartile range).

### b. Lost in higher dimension ? Select input features and model dimensions carefully

#### ● Feature selection

As discussed previously, over the course of the last 12 years, we at RAM AI have built a repository of numerous features capturing a variety of market inefficiencies on the back of traditional and alternative data sources. We use deep learning to combine all these stock features when predicting a stock's return, risk and liquidity dynamic. Deep learning approaches provide us with an optimal non-linear combination of these features, revealing their non-linear interactions.

We face numerous challenges when building a deep learning model combining a multitude of features :

- As the number of features and hence the number of model parameters grows, it becomes increasingly difficult to prevent overfitting when training a deep learning algorithm. Very large artificial neural networks lead much more easily to an overfitting of a training dataset, making generalisation capability of the model poor.

- The more input features in a deep learning algorithm the higher the likelihood of having highly correlated features in the input set. This makes the optimisation convergence of the model much less stable as it can lead to large variations in the local gradient of these variables.
- Larger input sets naturally require much more computing power to be trained, which makes the process of building the model extremely time-consuming.
- The interpretation of the model's complexity increases as the number of inputs grows.

To overcome all these issues, a feature selection before training a model becomes XXX when one deals with large number of inputs (hundreds, thousands or more).

The feature selection process begins with the identification of the most informative and relevant features to predict the output variables. They are the set of variables that should together help predict in the most robust manner the output variable.

## ● Univariate feature selection

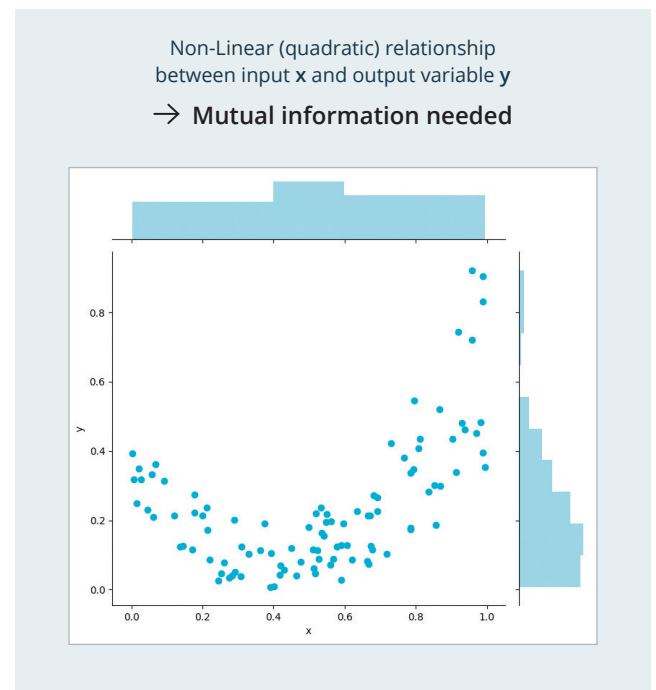
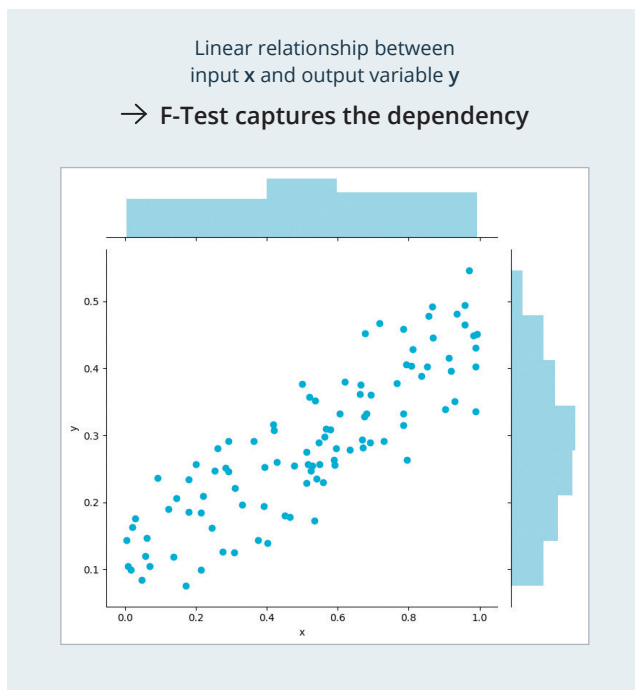
The simplest way to select input features is to retain those that have the strongest relationship with the output variable. The most basic approach is to select the features that are most highly correlated with the output variable. To better assess the linear relationship and consider the significance of this correlation, one can use a F-test which estimates the degree of linear dependency between each input and the output variable, selecting the top ranked ones.

But the limit of correlation or F-test scoring is that they only reflect linear relationships between the input and output variables, when there are often non-linear interactions at play in our data. In this case it would be more useful to conduct a more general univariate test looking at mutual information between each input and the output variable. Mutual information being entropy-based, can uncover any statistical dependency to help select the most relevant features.



Source: RAM AI

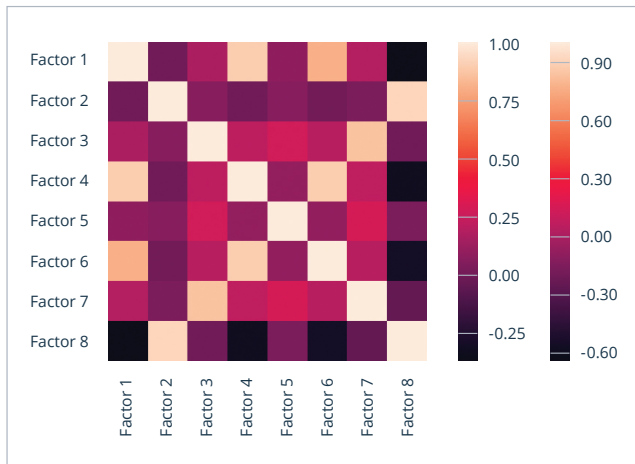
### *Linear vs Non-Linear Relationships*



Source: RAM AI



## Correlation Heatmap of a Set of Eight Input Market Factors



Source: RAM AI

### Regression-based ranking

We can fit a multivariate linear regression model to explain the output variable with all features and rank each feature along its statistical significance to retain the most explanatory ones.

The downside in this feature selection approach lies in the fact that a number of highly correlated input factors can still be retained, as often they would tend to have large betas with opposite signs and high statistical significance in a multivariate regression. This subsequently increases the risk of having a sub-optimal set of features in the deep learning model.

### Lasso

A feature selection based on a Lasso model can help to overcome the limits of a linear regression by reducing the likelihood of multiple highly correlated features being retained. A Lasso adds a penalty that sums up the absolute scale of the regression betas, a "L1" penalty, which forces many parameters to be zero as the strength of the regularisation increases. Weak and redundant features rapidly end up with zero coefficients, making the Lasso well-suited for a robust feature selection.

### Recursive feature elimination ("RFE")

Recursive feature elimination ("RFE") is a process that aims to ensure one misses the least interactions between features, by gradually taking into account how those interactions change as we reduce the size of the feature set.

It starts by fitting a model with all input variables, then iteratively eliminates groups of  $k$  variables at each step with the least importance in the model. One typically evaluates the importance of a variable by measuring the reduction of accuracy of the model when it is eliminated.

There is, however, a very strong limit to univariate tests to achieve a robust feature selection. Univariate tests retain the best features to explain the output on a stand-alone basis but totally ignore the complementarity or the redundancy of the features retained. In practice, the input features often exhibit a significant degree of correlation, making univariate tests sub-optimal.

Hence the need for feature selection with a multivariate angle capturing the complementarity of the inputs to identify an overall optimal group of variables.

The model is then fit again with the reduced feature set, to then exclude  $k$  variables again, and so on, until we reach the desired number of input features.

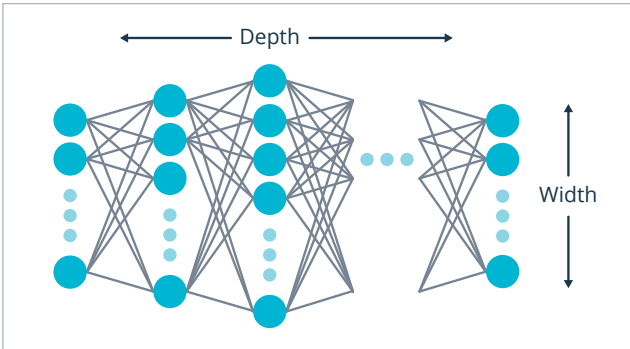
We can also select the right input features into our model using a more general non-linear model capturing the non-linear interactions between the input features, sometimes even using our deep learning model itself (e.g. an Artificial Neural Network). The main downsides here are heavy computing times and the need for very large datasets to complete this RFE feature selection. Indeed, to ensure no overfitting with the large number of parameters involved, one should use a cross-validation process that ensures generalisation, with large data samples required. It is also highly convoluted to have to set the hyperparameters for a more general feature selection model.

RFE is a computing-intensive feature selection technique as multiple fits are required to reach the targeted number of features, but it should best capture the non-linear interaction dynamics of the input factors selected and their complementarity.

### Deep Learning Model : what dimension?

Before diving into the engineering of the deep learning model, the tunings of the model parameters, the work of a data scientist setting the building blocks of a deep learning model is not dissimilar to those an architect. One of the important elements of a deep learning model architecture is its dimension. This dimension will depend a lot on the size of the input, i.e. the number of features retained after our feature selection process. But an equally important element will be the size of the data sample we rely on to train the algorithm.

## Artificial Neural Network Dimensions



Source: RAM AI

The way we approach dimensioning at RAM AI is “constructive” in the sense that we tend to start with simple and small artificial neural networks, and then increase width and depth incrementally until we find what seems to be an optimal network size. Given the size of our datasets, the largest which is typically dozens of GB with millions of single stock observation periods, we never reach the thousands of layers found in some of the most recent image recognition deep neural networks.

In the case of an artificial neural network, dimension both represents the width and the depth (number of layers) of a neural network. When one scales up a network from low dimensions, it is either possible to increase the width or the depth of the network. There is solid theoretical foundations for the strong performance of deep networks.

Montufar, Pascanu, Cho and Bengio (2014) demonstrate the strength of a combination of layers to capture complexity, identifying a number of input regions exponential to the depth of the network.<sup>1</sup>

In practice however it does seem that deep learning practitioners tend to exaggerate the number of layers in their networks, sometimes massively over-dimensioning them, while much smaller wider alternatives would both improve accuracy and save them computing costs, see for instance Wide Residual Networks by Zagoruyko & Komodakis, 2016.<sup>2</sup>

Schmidhuber, Greff and Srivastava also published a very interesting article aiming to overcome the difficulty and limits of very-deep neural network training (up to close to arbitrary size) in their Highway Networks (2015).<sup>3</sup>

The more data we have in our training and validation samples, the more parameters the algorithm can handle without overfitting and loss of generalisation.

<sup>1</sup> Reference: <https://arxiv.org/pdf/1402.1869.pdf>

<sup>2</sup> Reference: <https://arxiv.org/pdf/1605.07146.pdf>

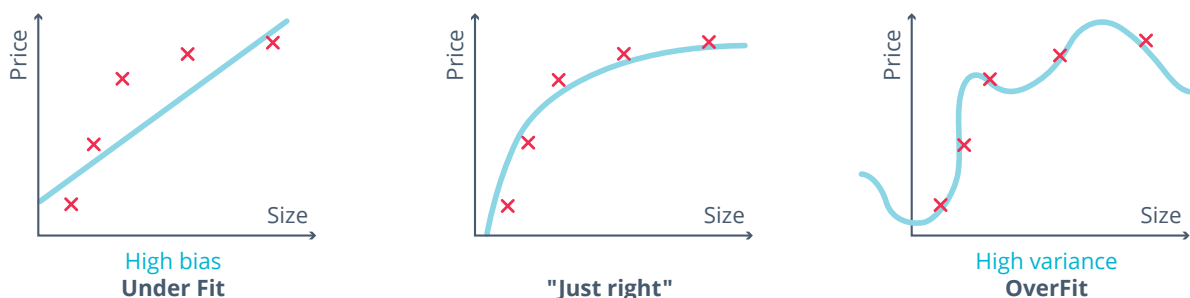
<sup>3</sup> Reference: <https://arxiv.org/pdf/1505.00387.pdf>

## c. Generalisation : Train with care, make sure you are not training on “noise”

When designing a machine learning model, it is crucial to consider how well the model can generalise to new data. If the model is appropriately fitted, it will be able to make predictions based on data it has never seen before. If the model is under or over fitted, generalisation won't be possible. The control of the fit is based on the bias-variance tradeoff.

- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict.
- Variance is the variability of model prediction for a given data point.

As illustrated below, high bias is a characteristic of an underfitted model and high variance is a characteristic of an overfitted model.



Source: RAM AI

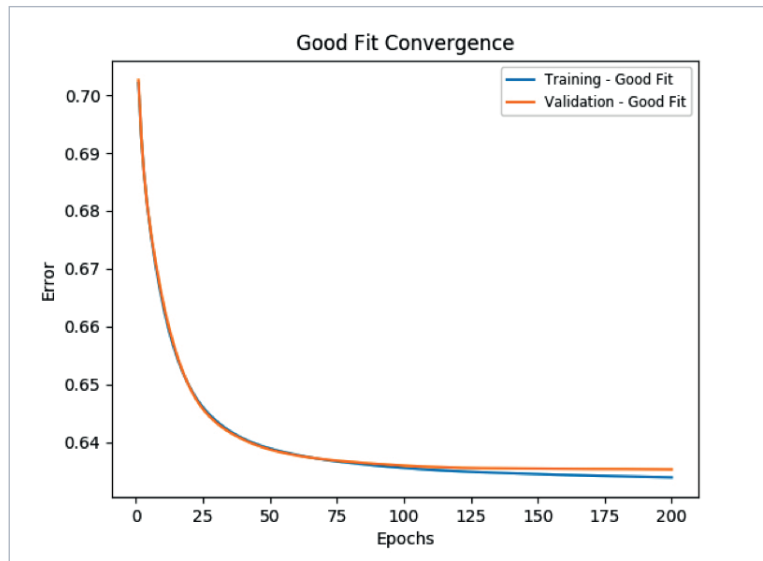


## ● How to control the generalisation

To control the generalisation of the model, we analyse the convergence of the loss function over training and validation samples. A loss function is a measure of how well the algorithm models the given data.

### Good Fit

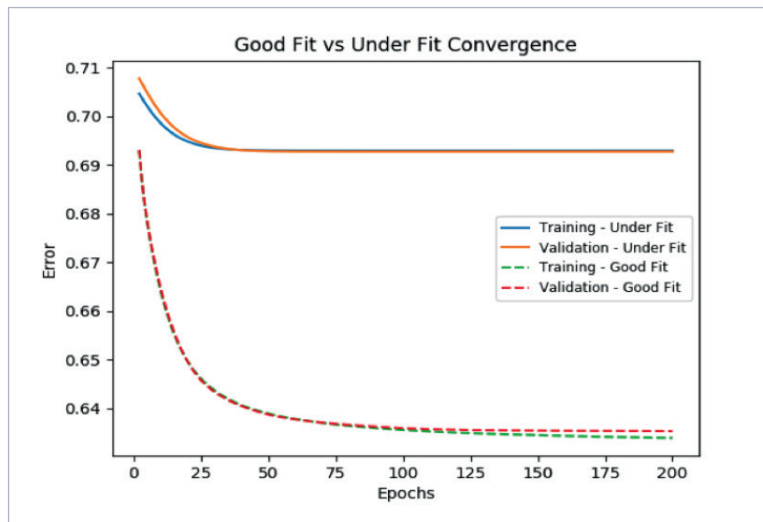
Ideally, the situation when the loss function reaches a global minimum, the model is said to have a good fit on the data. This case is achievable at the point between overfitting and underfitting. With the passage of time, our model will keep on learning and will be able to generalise. As such, the error for the model on the training and the validation data will keep on decreasing.



Source: RAM AI

### Under Fit

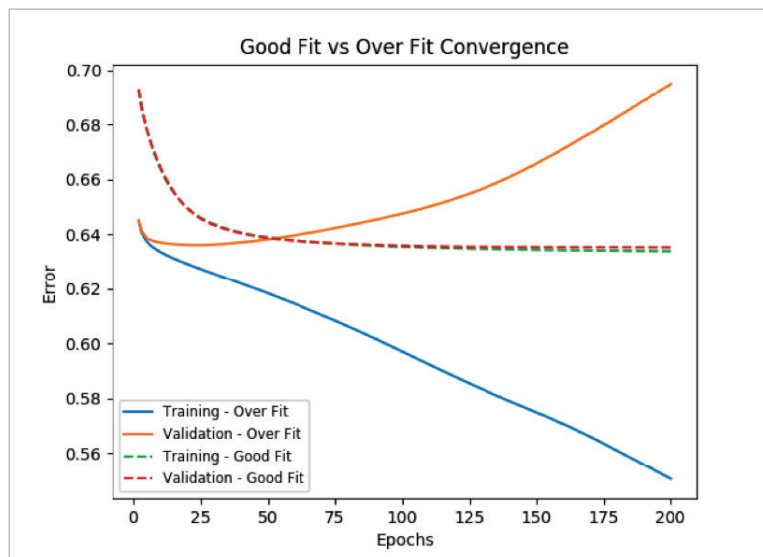
Models with a high bias will not learn enough from the training data and will oversimplify the model. Consequently, it cannot capture the underlying trend of the data in the chart. This leads to a high error count on both the training and validation data.



Source: RAM AI

### Over Fit

Models with high variance over-learn on the training data, capturing all its specificities (including true properties but also noise), and does not generalise on new data. As a result, such models get opposite convergence patterns on training and validation loss function: the more the model learns (decrease in training error), the less it is able to generalise (increase in validation error).



Source: RAM AI



## ● How to improve the generalisation

All the steps described in previous chapters are important to help the generalisation of the model, including data related matters (size of the dataset, data preprocessing, features selection) and algorithm specific issues (type and dimension of the algorithm). Additionally, several techniques can be used to control the machine during the learning process, to achieve an optimal fit.

### K Fold cross validation

In K Fold cross validation, the data is divided into k subsets. A training/validation process is repeated k times, such that each time, one of the k subsets is used as the validation set and the other k-1 subsets are put together to form a training set. The error is defined as the averaged error over all k trials. This helps to reduce bias/underfitting risk as most of the data is used for training and reduces variance/overfitting risk as most of the data is also being used for validation.

### Regularisation

One of the most common mechanisms for avoiding overfitting is called regularisation. Many regularisation approaches have been developed and below we are presenting two internally used techniques.

Ridge Regression :

To reduce the impact of irrelevant features on the trained model, another element is added to the loss function. This element leads to the loss function being negatively impacted by high features coefficients. As the function is minimised, the coefficients tend to be reduced, resulting in a simplified model, thus less prone to overfitting.

$$\text{Loss} = \sum (\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2$$

with

$\sum (\hat{Y}_i - Y_i)^2$  = Original Loss

$\lambda \sum \beta^2$  = Regularisation term

Dropout :

Deep neural networks have many parameters and as such, are prone to overfitting. The Dropout is a technique for addressing this problem. The concept is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much<sup>1</sup>.

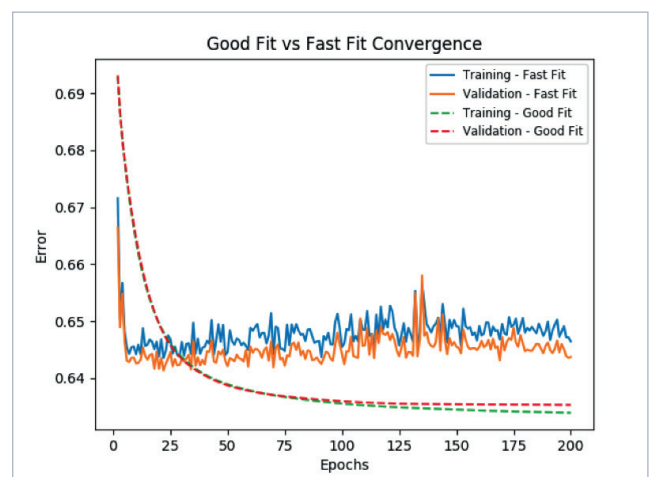
<sup>1</sup> Source : A Simple Way to Prevent Neural Networks from Overfitting. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov; 15(Jun):1929–1958, 2014.

### Early Stopping

During the training of the machine, there will be a point when the model will stop generalizing and start learning the statistical noise in the training dataset. A technique called Early Stopping is used to avoid overtraining by stopping the training when a trigger is reached. For instance, this trigger can be the point where the validation error starts increasing.

### Learning rate

Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network following a training step. The lower the value, the lower the impact of the correction. Our approach is to avoid the risk of setting a rate that would be too high or too low by using an adaptive learning rate which evolves during this training process. While using a low learning rate might seem a good idea in terms of ensuring that we do not miss the global minimum, it could also mean that the convergence will take a long time, especially if it gets stuck on a region with the same error (plateau). An increasing learning rate helps the convergence to progress faster and allows for traversal of plateau and saddle points (critical point in which some dimensions observe a local minimum, while other dimensions observe a local maximum). However, when the learning rate is too high, huge steps will be taken and the minimum will never be reached. As observed on the chart below, the loss is reduced very quickly, but then the parameters become unstable and the loss diverge from the minimum on both training and validation set.



Source: RAM AI



### d. Understand what you are teaching : Establish the right objectives and focus on interpretability

At RAM AI, we believe focusing on model interpretability is key when developing stock selection strategies which combine a high number of factors. Interpretability means understanding why a model makes a certain prediction. It's even more important since :

- It can highlight the reasons why a given stock has been selected (or not) at a given date and to better understand the drivers of our selections (and their dynamics over time).
- It can help detect the presence of potential outliers in input data, which would require further investigation.
- It helps provide comfort with the predictions of the models and may also teach humans some basic interactions between certain input features and expected forward stock behaviours.
- It can identify potential biases when constructing the input dataset used for backtesting purposes.

However, we are using deep learning models which are, by their very nature, complex. Complex models generally allow to obtain good predictive accuracies, which often comes at the expense of interpretability.

Different approaches can be used. The most direct ones are perturbation-based methods; they directly compute the attribution of an input feature (or set of features) by removing or altering them (adding noise for example) and running a prediction (with a simple forward pass) on the new input, measuring the difference with the original prediction. Other popular approaches include the "gradient-based" methods; they compute the attribution by calculating the partial derivatives of the prediction with respect to each input feature and multiplying them with the input itself. Other popular methods include "relevance backpropagation" and "attention-based methods", however we will not discuss these here.

Interpretability has been an increasing area of research recently, and many packages have been proposed to enhance model explainability, such as SHAP - SHapley Additive exPlanations, which is an attempt to unify some of these attribution methods.

#### Factor Biases

It is also important to monitor input factor biases across a stock selection.

At RAM AI, we also gain comfort in the predictions of our stock selection models by looking at the average factor biases for a selection at a given date. They must make sense on average and be relatively stable historically. The chart below shows an example of average factor biases for a long-only stock selection model based on 51 factors within emerging markets (the higher the better), at a given date.

Factor	Bias	Factor	Bias
VLM_FACTOR_1	0,20	EST_FACTOR_1	-0,02
VLM_FACTOR_2	0,48	EST_FACTOR_2	0,06
VLM_FACTOR_3	0,18	EST_FACTOR_3	0,44
SIZ_FACTOR_1	0,10	EST_FACTOR_4	0,50
SIZ_FACTOR_2	0,06	EST_FACTOR_5	0,75
QLT_FACTOR_1	0,56	EST_FACTOR_6	0,68
QLT_FACTOR_2	0,35	MOM_FACTOR_1	0,26
QLT_FACTOR_3	0,14	MOM_FACTOR_2	0,28
QLT_FACTOR_4	0,21	MOM_FACTOR_3	0,11
QLT_FACTOR_5	0,11	MOM_FACTOR_4	0,30
QLT_FACTOR_6	0,11	MOM_FACTOR_5	-0,07
FIN_FACTOR_1	0,25	MOM_FACTOR_6	0,29
FIN_FACTOR_2	0,13	RSK_FACTOR_1	0,27
FIN_FACTOR_3	0,50	RSK_FACTOR_2	0,69
FIN_FACTOR_4	0,59	RSK_FACTOR_3	0,19
GTH_FACTOR_1	0,17	RSK_FACTOR_4	-0,52
GTH_FACTOR_2	0,07	RSK_FACTOR_5	-0,28
VAL_FACTOR_1	0,42	RSK_FACTOR_6	-0,03
VAL_FACTOR_2	0,38	RSK_FACTOR_7	0,51
VAL_FACTOR_3	0,56	MOM_FACTOR_1	0,17
RSK_FACTOR_1	0,17	MOM_FACTOR_2	0,33
RSK_FACTOR_2	0,37	MOM_FACTOR_3	0,15
RSK_FACTOR_3	0,77	MOM_FACTOR_4	0,22
RSK_FACTOR_4	0,77	MGT_FACTOR_1	0,04
		MGT_FACTOR_2	0,15
		BOOL_FACTOR_1	0,38
		BOOL_FACTOR_2	0,36

Source: RAM AI



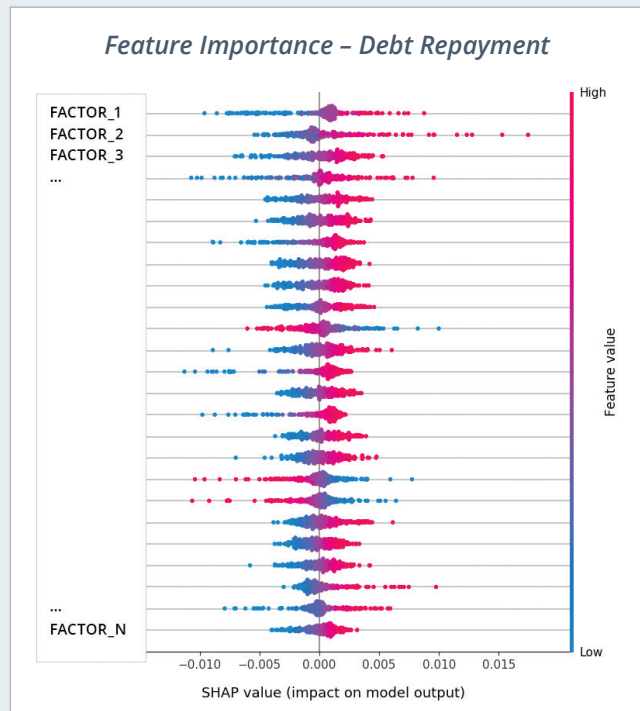


## The SHAP package : an efficient model interpretation

The SHAP library is designed to calculate Shapley values significantly faster than if a model prediction had to be calculated for every possible combination of features. By implementing model specific algorithms, it takes advantage of different model structures. For example, SHAP's integration with gradient boosted decision trees takes advantage of the hierarchy in a decision tree's features to calculate the SHAP values, in polynomial time instead of the classical exponential runtime. For Python, the SHAP package provides Shapley values calculations and (partly interactive) plots to explore Shapley values across different features and/or observations.

The figure on the right shows the distribution of Shapley values, for each feature, across the observations in a test sample used for a stock selection strategy. The most discriminative feature (factor) appears at the top of the chart. The sign shows whether the factor value moved the prediction – a measure of expected forward stock behaviour – toward positive (positive SHAP) or negative (negative SHAP).

The colour reflects the order of magnitude of the factor values (low, medium, high). This way the plot immediately reveals that low (blue) values on the top seven factors deteriorate prospects for a given stocks.



Source: RAM AI

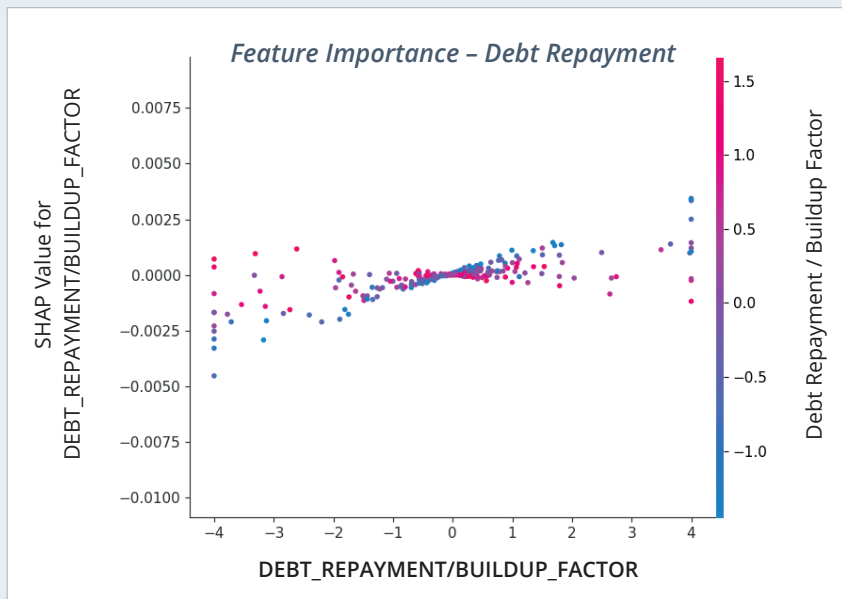
### ● The SHAP unified approach to interpretability

SHAP is presented as a unified approach to interpreting model predictions, we use it to measure variable importance, and to visualise interactions between some input variables and the response variable. This approach comes from game theory and is based on Shapley values (named after game theorist and Nobel prize winner Lloyd Shapley). They are used to fairly distribute credit or value to each individual player/participant in cooperative situations, or in short, to design a fair payout scheme. In summary, Shapley values calculate the importance of a feature by comparing what a model predicts with and without the feature. However, the order in which a model sees features may have an impact on its predictions, thus this comparison is done in every possible order.

In their paper presented at NIPS 2017, Lundberg and Lee present an approach to compute "Shapley Additive exPlanations", an additive feature importance measure, and theoretical results (derived from a proof from game theory on the fair allocation of profits). This showed that, in the class of additive feature attribution methods, their approach is the only one which verifies a number of desirable properties, in particular local accuracy (the sum of all the feature importance should sum up to the total importance of the model) and consistency (if we change a model such that it relies more on a feature, then the attributed importance for that feature should not decrease).

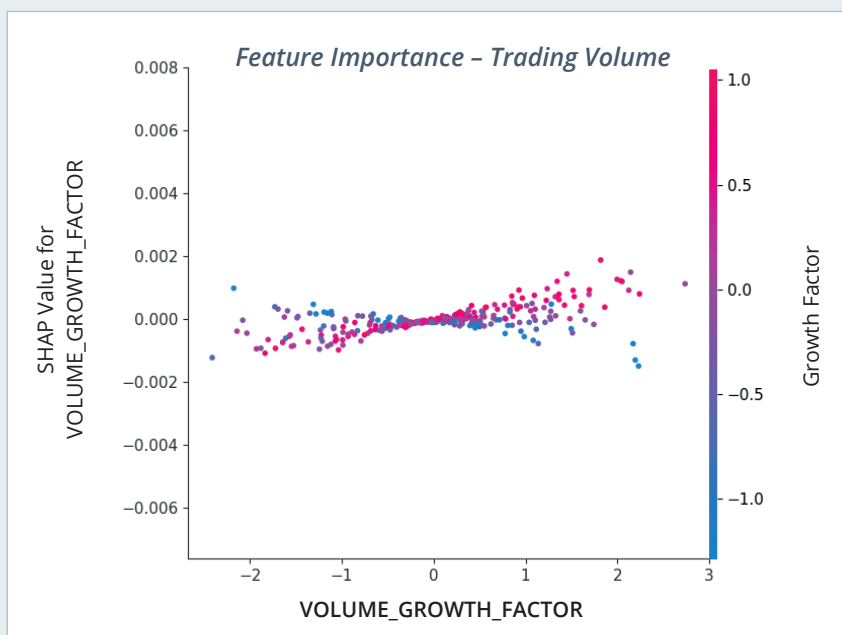
Additional interactions charts are available, such as the one below representing SHAP value for the repayment / build-up of debt in our stock selection model.

We were mentioning previously that debt levels and debt dynamics have an unclear impact on expected forward stock behaviour, as their interactions with future returns are not linear. The chart below shows that although the repayment / build-up of debt has a slight positive impact on expectations on average, the model distinguishes between stocks ranking positively on a growth factor (in red) and the others (in blue): the latter ones are clearly expected to be rewarded by the market when they are able to repay their debt, or conversely, to be punished if they accumulate excessive debt.



Source: RAM AI

Below we present another interaction chart showing that trading volume growth has an impact on future returns after conditioning on whether a stock ranks positively on a growth factor :



Source: RAM AI



## Conclusion

The general enthusiasm for machine learning techniques will not turn out to be a short-lived fashion trend. It is proving to be, across numerous fields of application, the most efficient tool to extract information and build predictive models from large datasets.

The significant investments of tech giants into the technology and hardware supporting deep learning will help maintain this trend. The flexibility and robustness of their open-source development frameworks, as well as the significant processing power accessible (either directly or through their highly evolutive cloud infrastructures), make machine learning techniques efficient data processing techniques. This is increasingly welcome at a time when digital data is exploding, alternative or unstructured financial datasets.

But we must be conscious that the history of artificial intelligence is awash with over-exuberant expectations. Here, the most stunning example could be the optimistic forecast of cognitive scientist Minsky in 1970: "In from three to eight years we will have a machine with the general intelligence of an average human being". Who would only accurately predict today when fully autonomous self-driving cars will be reliable?

We prefer to regard machine learning techniques as a generalisation of traditional data processing techniques, and our research efforts are equally focused on testing models and controlling them by making sure they generalise and provide tangible results. At a time when the number of attempts to generate truly human-like AI is increasing and excitement about fully-autonomous vehicles is growing, we embrace initiatives about "Human trust in AI".



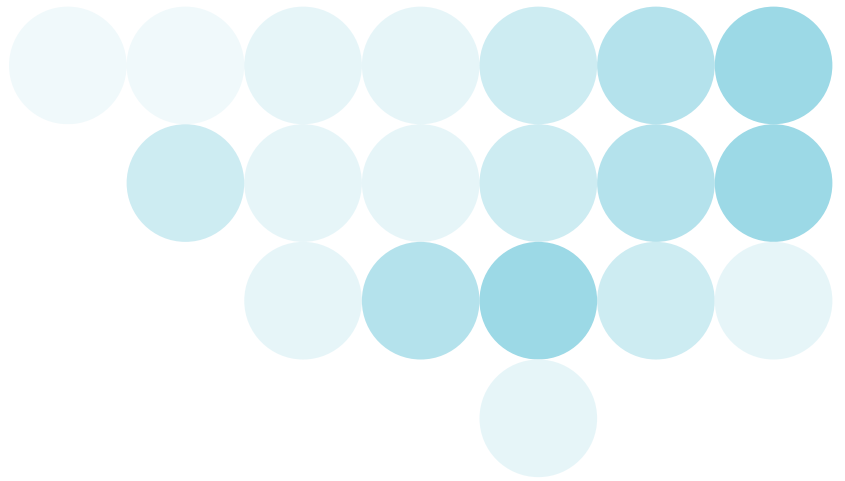
Many thanks to François Fleuret, head of the Machine Learning group at IDIAP Research Institute and adjunct faculty at EPFL, for his precious insights and advices on our deep learning setup. (<https://www.idiap.ch/~fleuret/>)

## References

- RAM's Systematic Equity : A leading approach to ESG Integration (RAM Active Investments, 2018)
- The Digitization of the World, From Edge to Core (Gantz, Reinsel, Rydning, 2018)
- On the Number of Linear Regions of Deep Neural Networks (Bengio, Cho, Montufar, Pascanu, 2014)
- Wide Residual Networks (Komodakis, Zagoruyko, 2017)
- Highway Networks (Greff, Schmidhuber, Srivastava, 2015)
- Dropout : A Simple Way to Prevent Neural Networks from Overfitting (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014)
- A Unified Approach to Interpreting Model Predictions (Lee, Lundberg, 2017)

## Disclaimer

This material is addressed to professional clients for informative purposes only. It is neither an offer nor an invitation to buy or sell investment products and may not be interpreted as investment advice. It is not intended to be distributed, published or used in a jurisdiction where such distribution, publication or use is forbidden, and is not intended for any person or entity to whom or to which it would be illegal to address such a material. In particular, investment products are not offered for sale in the United States or its territories and possessions, nor to any US person (citizens or residents of the United States of America). The opinions herein do not consider individual clients' circumstances, objectives, or needs. Before entering into any transaction, clients are advised to form their own opinion and consult professional advisors to obtain an independent review of the specific risks incurred (tax, financial etc.). Upon request, RAM AI Group is available to provide more information to clients on risks associated with investments. The information and analysis contained herein are based on sources deemed reliable. However, RAM AI Group does not guarantee their accuracy, correctness or completeness, and it does not accept any liability for any loss or damage resulting from their use. All information and assessments are subject to change without notice. Changes in exchange rates may cause the NAV per share in the investor's base currency to fluctuate. There is no guarantee to get back the full amount invested. Past performances, whether actual or back-tested, are not necessarily indicative of future performance. Without prejudice of the due addressee's own analysis, RAM understands that this communication should be regarded as a minor non-monetary benefit according to MIFID regulations. Clients are invited to base their investment decisions on the most recent prospectus, key investor information document (KIID) and financial reports which contain additional information relating to the investment product. These documents are available free of charge from the SICAV's and Management Company's registered offices, its representative and distributor in Switzerland, RAM Active Investments S.A. and at Macard Stein & Co AG, Paying and Information Agent in Germany; and at RAM Active Investments (Europe) SA - Succursale Milano in Italy. This marketing material has not been approved by any financial Authority, it is confidential and addressed solely to its intended recipient; its total or partial reproduction and distribution are prohibited. Issued in Switzerland by RAM Active Investments S.A. which is authorised and regulated in Switzerland by the Swiss Financial Market Supervisory Authority (FINMA). Issued in the European Union and the EEA by the Management Company RAM Active Investments (Europe) S.A., 51 av. John F. Kennedy L-1855 Luxembourg, Grand Duchy of Luxembourg. The reference to RAM AI Group includes both entities, RAM Active Investments S.A. and RAM Active Investments (Europe) S.A.



**RAM Active Investments SA**  
**Geneva**

Rue du Rhône 8  
1204 Geneva – SWITZERLAND  
Tel : +41 58 726 87 00

**RAM Active Investments AG**  
**Zurich**

Talstrasse 65  
8001 Zurich – SWITZERLAND  
Tel : +41 58 717 97 97

**RAM Active Investments (Europe) SA**  
**Luxembourg**

51, avenue John F. Kennedy  
1855 Luxembourg – LUXEMBOURG  
Tel : +352 28 56 14 1

**RAM Active Investments (Europe) SA**  
**Milan**

Via Montenapoleone 14  
20121 Milan – ITALY  
Tel : +39 027 788 4001