ANTWORKS™

Reimagine IT

DEMYSTIFYING FRACTAL SCIENCE IN COGNITIVE AUTOMATION

CONTENTS

Jumping onto the AI Wagon 2
Demystifying Duality and Self-similarity of Fractals 3
Types of Networks 4
Fractal Architecture 5
Duality in Self-similar Syndromes 8
Hybrid Designs 8
Drop-Path 8
Block Functions 9
Topology Mapping and Deployment 10
Topology Mapping 11
Roadmap for the Way Ahead 13
References 15

JUMPING ONTO THE AI WAGON

What is the common thread between children in a playpen, trainees on their first day at work and a clueless person at a bank kiosk? It may not be obvious at first but their circumstances are remarkably similar. Unaware, afraid to ask and eager to learn, all three scenarios are perfect examples of how cognitive thinking helps us all grasp an idea, learn through observation and emulate it in our own ways. Cognitive thinking is the exact theory that artificial intelligence (AI) is based on. No, AI is not a new mantra or trend. After all, it has been around for the last 50 years or so, but never has it been as acclaimed as it has been since the last 2 years. With the increased investments in infrastructure and automation, everyone is realising that the AI band wagon has much more to offer than big data and analytics.

Today we are using cognitive thinking to help bots read, learn, contextualize , infer and automate front-, middle- and back office processes across industries. Learning through cognitive thinking is quickly becoming the preferred way to shape organizations, industries and customer experiences. With endless opportunities in real-life business applications, AI is the inevitable omnipresent, bursting forth with promises of the next wave in technology innovation.

However, as leaders look to get their questions answered and prepare a roadmap to integrate AI in their existing business processes, they are keen to understand how these cognitive skills can be leveraged for real world enterprises

Some frequently asked questions include:

- Based on a set of steps executed by users, can machine learning algorithms monitor and arrive at patterns? While these steps are very different between users, will they have the same outcomes?
- Is process reengineering a mandatory precursor to automation? How can we derive insights faster or more efficiently?
- Can we unlock hidden correlations between data elements without specifically querying for it?
- Is there a method or approach to automatically read and infer the context with a machine?
- Can we combine one or a set of algorithms to create a net that will orchestrate and form a decision matrix based on the cognitive ability gained by the very influx of data?
- Will these algorithms predict future patterns based on the historic trend of data and take actions proactively (self-heal)?

DEMYSTIFYING DUALITY AND SELF-SIMILARITY OF FRACTALS

Most cognitive computing and machine learning solutions use neural science as a foundation. Neural science is based on the premise that each neuron carries identical information. Recent studies show that fractals hold great promise when it comes to cognitive learning. Fractal science is based on the premise that networks of neurons carry similar but not identical signals about patterns, while neural networks read absolute data.

In this paper, we have set out to illustrate how fractal genesis can play an integral role in the field of machine learning. In effect, the duality and self-similarity offers a singleton value, which monotonically converges to the desired set of results. A synthesis of practical recommendations when using fractal dimensions on stochastic models in predicting the variances in the nearest neighborhood typically concludes with deterministic values.

Let us focus on two aspects of AI that are still relatively new, that have not been exploited to the core and can yield immense benefits when leveraged

1st

Let's assume that there are domains in machine learning that have an inherent fractal structure.

2nd

Most commonly used machine learning algorithms (associated algorithms of neural networks) do not exploit this after structure.

In addition to investigating these two aspects, we shall establish options for new algorithms that can exploit such fractal structures.

The first assumption suggests that in various learning tasks the dataset (the input from which we wish to learn) contains fractal characteristics. Broadly speaking, there are details at all scales. When you zoom in, even slightly, the data reveals a non-smooth structure. This lack of smoothness can be seen naturally in phenomena such as clouds, coastlines, mountain ranges, and the crests of waves.

If this detail is to be exploited, the object under study must also be self-similar, i.e. the large-scale features must mirror the small-scale features, if only statistically. And indeed, in most natural fractals, this is the case. The shape of a limestone fragment will be closely related to the ridges of the mountainside where it broke off originally, which in turn will bear resemblance to the shape of the mountain range as a whole.

Finding natural fractals is not difficult. Very few natural objects are smooth, and the human eye has no problem recognizing them as fractals. In the case of datasets used in machine learning, finding fractal structures is not as easy. Often these datasets are modelled on a Euclidean space of dimension greater than three, and some of them are not Euclidean at all, leaving us without our natural geometric intuition. The fractal structure may be there, but there is no simple way to visualize the dataset as a whole. We need to analyze various datasets to investigate their possible fractal structure.

HERE'S HOW

The fractal dimension is an interesting metrics because it is supposed to quantify by a single value, scale independence and roughness of ecological objects. However, in the following scenario, we will show that those two properties may be quantified by a single dimension only in some specific cases. In general, a non-integer quantifies only the roughness, and self-similarity needs to be evidenced or postulated by other means. Secondly, we revisited some aspects of the practical estimation of fractal dimensions, and have proposed a simplification of its estimation for 2D fields and discuss its possible relationship with self-similarity.

Our second assumption is that when this fractal structure and self-similarity exists, most commonly used machine learning algorithms cannot exploit it.

The geometric objects that popular algorithms use to represent their hypotheses are always linear in nature.

This means that, even if they represent the data in a narrow range of scales, they cannot do it on all scales, giving them an inherent limitation on how well they can model any fractal dataset. While the scope of this document does not allow a complete and rigorous investigation of these claims, we have endeavored to provide some initial research into this relatively unexplored area of machine learning.

TYPES OF NETWORKS

There are many different types of networks that assist in machine learning and can be modified to address the problem of training and recognition solutions.

- Feed-forward Fractal Network
- Training Algorithm: Backpropagation
- Convolutional Neural Network
- Deep Belief Network
- Parzen Probabilistic Neural Network
- Stacked Auto-Encoders
- Support Vector Neural Network
- Three Layer Neural Network

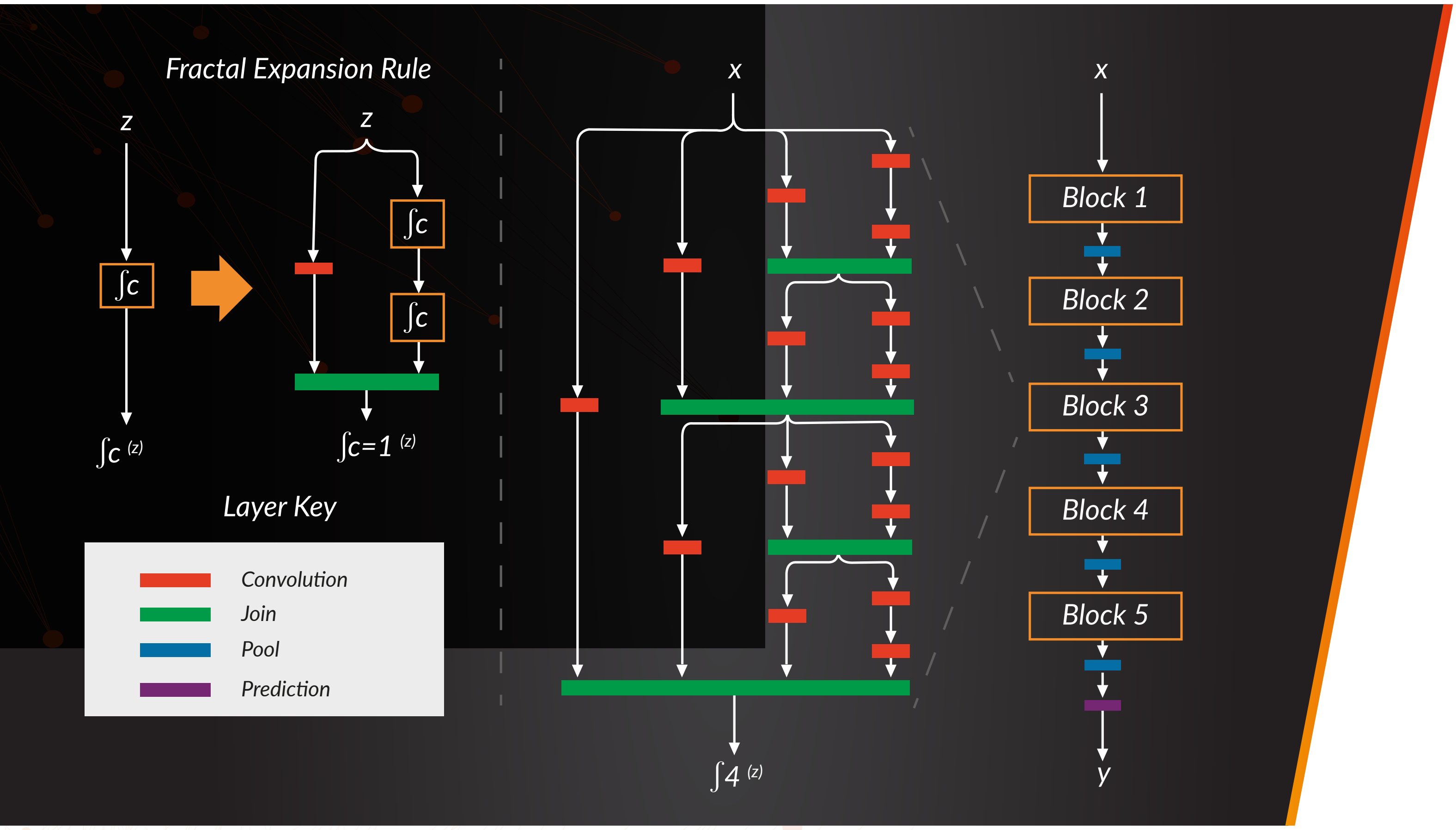
FRACTAL ARCHITECTURE

Convolutional neural networks (CNNs) have become a dominant machine learning approach for visual object recognition. Despite their introduction more than 20 years ago, improvements in computer hardware and network structure have enabled the training of truly deep CNNs only recently. The original LeNet5 consisted of 5 layers, VGG featured 19, and only last year Highway Networks and Residual Networks (ResNets) have surpassed the 100-layer barrier.

As CNNs become increasingly deep, a new challenge arises. As information about the input or gradient passes through many layers, it can vanish and “washout” by the time it reaches the end (or beginning) of the network. Many recent publications address this and/or related problems. ResNets and Highway Networks bypass signals from one layer to the next via identity connections. Stochastic depth shortens ResNets by randomly dropping layers during training to allow better flow of information and gradient.

Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output.

In this paper, we embrace this observation and introduce the Fractal Network (FNET) that connects each layer to the other in a feed-forward fashion. While traditional convolutional networks or neural networks with L layers have L connections — one between each layer and its subsequent layer — our network has L(L+1) 2 direct connections. The feature-maps of all preceding layers are used as inputs for each layer, and its own feature-maps are used as inputs into subsequent layers.



In the image above, a simple expansion rule generates a fractal architecture with C intertwined columns on the left. The base case, f1pzq, has a single layer of the chosen type (e.g. convolutional) between input and output. On the right, you can see that deep convolutional networks periodically reduce spatial resolution via pooling. A fractal version uses fC as a building block between pooling layers. Stacking B such blocks yields a network whose total depth, measured in terms of convolution layers, is B _ 2C_1. This example has depth 40 (B _ 5, C _ 4).

Let’s look at the benefits of FNET over Neural Networks to understand this better:

- A Fractal network is chosen to simulate and model ANN (Artificial neural Network) with a number of neurons which are monotonically divergent.
- Neurons in ANN tend to have fewer connections than biological neurons.
- Each neuron in ANN receives a number of inputs.
- An activation function is applied to these inputs which results in the activation level (output value) of the neuron.
- Knowledge about the learning task is given in the form of examples called training examples. These results are correlated by fractal nodes instead of possible neuron structures.

An Artificial Fractal Network (AFN) is specified by:

- A Fractal node (model): Information processing unit of the FN.
- Architecture: A set of neurons and links which connects neurons where each link has a separate weight.
- Learning algorithm: Used for training the FN by modifying the weights in order to model a particular learning task correctly on the training examples.
- The aim is to obtain a FN that is trained and generalizes well, as well as behave correctly on new instances of the learning task.

Benefits of FNET

- Alleviates the vanishing-gradient problem
- Strengthens feature propagation
- Encourages feature reuse
- Substantially reduces the number of parameters

DUALITY IN SELF-SIMILAR SYNDROMES

FNETs repeatedly combine several parallel layer sequences with different number of convolutional blocks to obtain a large nominal depth, while maintaining many short paths in the network.

Although these different approaches vary in network topology and training procedures, they share a key characteristic in that they create short paths from early layers to later duality in self-similar syndromes.

HYBRID DESIGNS

In fractal networks, simplicity of training mirrors its simplicity of design. A single loss, attached to the final layer, suffices to drive internal behavior mimicking deep supervision. Parameters are randomly overall depth, making them deep enough and training will carve out a useful assembly of subnetworks. The entirety of emergent behavior resulting from a fractal design may erode the need for recent engineering tricks intended to achieve similar effects. These tricks include residual functional forms with identity initialization, manual supervision, hand-crafted architectural modules, and student-teacher training regimes. Hybrid designs could certainly integrate any of them with a fractal architecture with an open-ended question about the degree to which such hybrids are synergistic.

DROP-PATH

Drop-path includes a regularization strategy and provides means of optionally imparting fractal networks with anytime behavior. A particular schedule of dropped paths during learning prevents subnetworks of different depths from co-adapting. As a consequence, both shallow and deep subnetworks must individually produce the correct output. In this section, we will elaborate upon the technical details of fractal networks and drop-path through experimental comparisons to residual networks across datasets.

Drop-path regularization forces each input to a join to be individually reliable. This reduces the reward for even implicitly learning to allocate part of one signal to act as a residual for another.

Experiments show that we can extract high-performance subnetworks consisting of a single column. This type of a subnetwork is effectively devoid of joins, as only a single path is active throughout. They do not produce any signals to which a residual could be added. Together, these properties ensure that join layers are not an alternative method of residual learning.

Dropout and drop-connect modify interactions between sequential network layers in order to discourage co-adaptation. Since fractal networks contain an additional macro-scale structure, we propose to complement these techniques with an analogous coarse-scale regularization scheme.

Just as dropout prevents co-adaptation of activations, drop-path prevents co-adaptation of parallel paths by randomly dropping operands of the join layers. This discourages the network from using one input path as an anchor and another as a corrective term (configuration that, if not prevented, is prone to over-fitting).

Consider two sampling strategies

- **Local:** A join drops each input with fixed probability, but we make sure at least one survives.
- **Global:** A single path is selected for the entire network. We restrict this path to a single column, thereby promoting individual columns as independently strong predictors.

BLOCK FUNCTIONS

A fractal network block functions with some connections between layers disabled, as long as some path from the input to the output is still available. Drop-path guarantees at least one such path while sampling a subnetwork with many other paths disabled. During training, presenting a different active subnetwork to each mini-batch prevents co-adaptation of parallel paths.

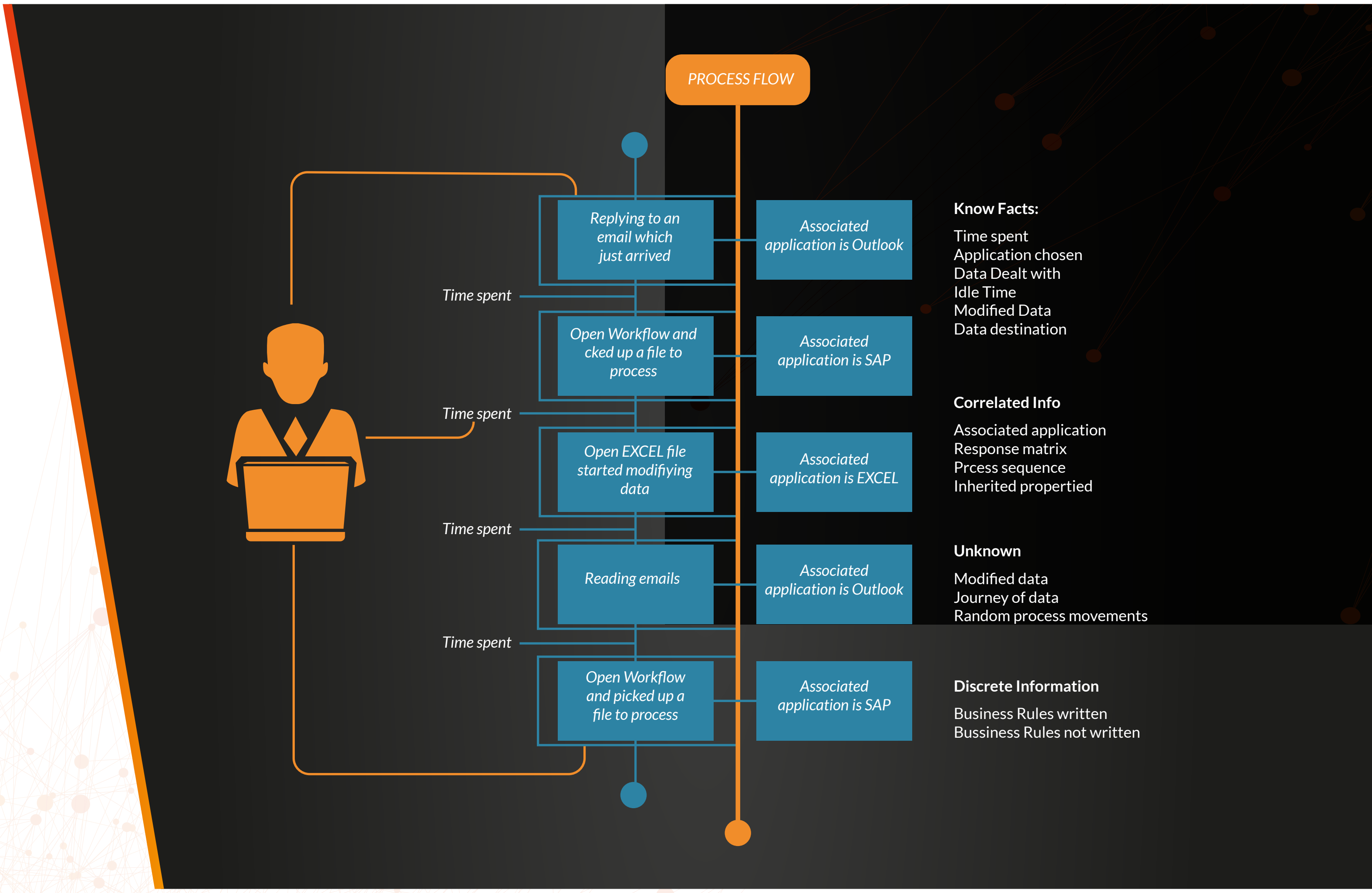
A global sampling strategy returns a single column as a subnetwork. Alternating it with local sampling encourages the development of individual columns as performant stand-alone subnetworks. As with dropout, signals may need appropriate rescaling. With element-wise means, this is trivial; each join computes the mean of its active inputs only.

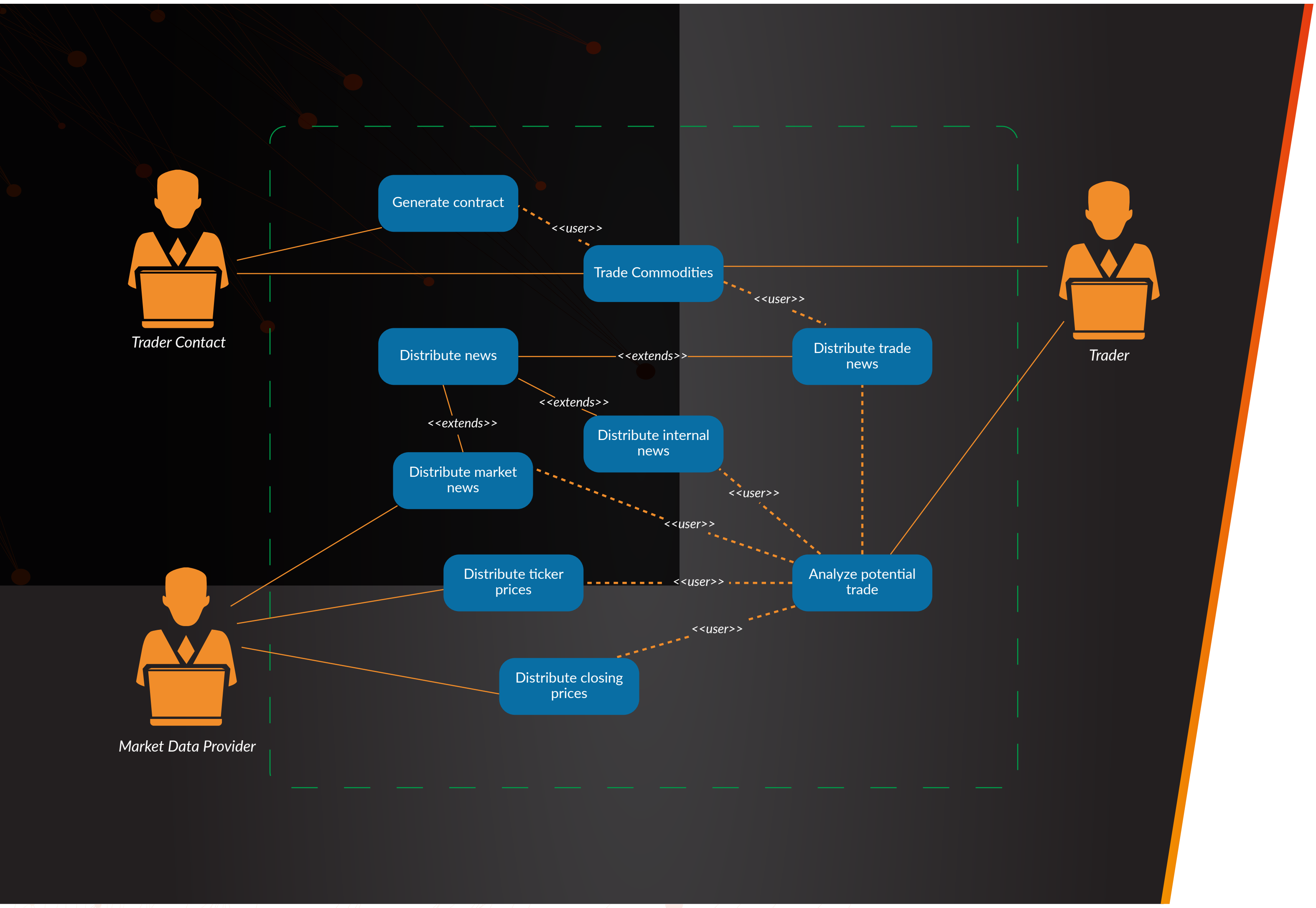
For example, we can train with dropout and a mixture model of 50% local and 50% global sampling or drop-path. We sample a new subnetwork each mini-batch. With sufficient memory, we can simultaneously evaluate one local sample and all global samples for each mini-batch by maintaining separate networks and tying them together via weight sharing.

While fractal connectivity permits the use of paths of any length, global drop-path forces the use of many paths whose lengths differ by orders of magnitude (powers of 2). Therefore, the subnetworks sampled by drop-path exhibit large structural diversity. This property stands in contrast to stochastic depth regularization of residual network, which, by virtue of using a fixed drop probability for each layer in a chain, samples subnetworks with a concentrated depth distribution.

TOPOLOGICAL MAPPING AND DEPLOYMENT

Demystify the relationship between action taken, data massaged (data cleaned/scrubbed), and the process through which the journey of data occurs. Here's how a topology model can be deployed and how the Same-Model Topology Processing algorithm can detoxify the bond between the massaged data, business rules, and the associated process.





TOPOLOGY MAPPING

Users often execute their tasks randomly, while depending on the availability of data, resources and their priorities. Though these appear random, there is a pattern at intervals, which repeats recursively either linearly or non-linearly. In order to map such occurrences and find the shortest or optimized path, a virtual topology can be crafted to represent the actions and associated impacts as shown in the figure above.

A virtual topology represents the way that processes communicate and their corresponding nodal points.

Nearest neighbor exchange in a mesh

Recursive doubling in an all-to-all exchange

During the process of spatial information collection, some inevitable problems may occur, such as the double emergency of the same node or line, emergencies of features, intersection and leaking polygon during the collection process of neighboring facial geometry objects of the nodes typically their properties, their availabilities and so on. Therefore, a topological process is needed to deal with these redundancies and errors.

Whilst following a simple connectivity rule, FNET naturally integrates the properties of identity mappings, deep supervision, and diversified depth. They allow feature reuse throughout the networks and can consequently learn more compact and accurate models. Because of their compact internal representations and reduced feature redundancy, FNET can be leveraged as feature extractors for various computer vision tasks that build on convolutional features. Going forward, it will be interesting to study such feature transfers with FNET.

ROADMAP FOR THE WAY AHEAD

While many organization consider leveraging 'fractal' dimensions of a dataset as a good approximation of its intrinsic dimension, and to drop attributes that do not affect it, we have applied our method to real and synthetic datasets which produced accurate results with amazing speed, efficiency and clarity.

As an AI product company, AntWorks has helped many clients run their businesses leveraging our integrated stack of Cognitive Machine Reading, RPA and actionable Insights technologies. The ANTStein platform from AntWorks enables users to integrate AI into their business processes for data ingestion and harnessing actionable insights to drive faster, more accurate, intelligent automation.

ANTStein uses fractal networks to discern patterns, equipping it with superior machine learning abilities. While learning, the platform exhibits scale invariance where it can retain the self-similar resonating properties of shape while removing the dependence on scale dimension, making adaptive learning much easier with ANTStein. While it gets progressively accurate over time (it requires far smaller sample sets to learn from), training cycle times are reduced which result in far smaller infrastructure requirements.

ANTStein has achieved state-of-the-art results across several highly competitive datasets.

- No rotation of attributes, thus leading to easier interpretation of the resulting attributes
- Ability to identify attributes that have nonlinear correlations
- Constant number of passes over the dataset
- Accurate estimate on how many attributes we should keep

Leveraging FNET for ANTStein introduces direct connections between any two layers with the same feature-map size. With ANTStein we can show that FNET scales naturally to hundreds of layers, while exhibiting no optimization difficulties. In our experiments, FNET tends to yield consistent improvement in accuracy with growing number of parameters without any signs of performance degradation or overfitting.

REFERENCES

[Aha_1995] D. W. Aha and R. L. Bankert, “A Comparative Evaluation of Sequential Feature Selection Algorithms,” in Proceedings of the Fifth Intl. Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, FL, 1995.

ALLIGOOD, K.T., SAUER, T., & YORKE, J.A. 1996. Chaos: an introduction to dynamical systems. Springer Verlag.

ANDRLE, R. 1996. The west coast of Britain: statistical self-similarity vs. char- acteristic scales in the landscape. Earth surface processes and landforms, 21(10), 955–962.

ASUNCION, A., & NEWMAN, D.J. 2007. UCI Machine Learning Repository. BEYER, H.G., & SCHWEFEL, H.P. 2002. Evolution strategies–A comprehensive

introduction. Natural Computing, 1(1), 3–52.

BISHOP, C.M., et al. 2006. Pattern recognition and machine learning. Springer New York:.

CHAN, K.S., & TONG, H. 2001. Chaos: a statistical perspective. Springer Verlag.

CHANG, S.H., CHENG, F.H., HSU, W., & WU, G. 1997. Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes. Pattern Recognition, 30(2), 311–320.

[Belussi_1995] A. Belussi and C. Faloutsos, “Estimating the Selectivity of Spatial Queries Using the ‘Correlation' Fractal Dimension,” in 21th Intl. Conf. on Very Large Data Bases (VLDB), Zurich, Switzerland, 1995.

[Berchtold_1998] S. Berchtold, C. Böhm, H.-P. Kriegel, “The Pyramid-Tree: Breaking the Curse of Dimensionality,” in ACM SIGMOD Intl. Conf. on Management of Data, Seattle, Washington, 1998.

[Blum_1997] A. L. Blum and P . Langley, “Selection of Relevant Features and Examples in Machine Learning,” Artificial Inteligence, vol. 97, pp. 245-271, 1997.

[Böhm_2000] C. Böhm and H.-P. Kriegel, “Dynamically Optimizing High-Dimensional Index Structures,”

[Faloutsos_1996] C. Faloutsos, Searching Multimedia Databases by Content. Boston: Kluwer Academic Publishers, 1996.

[Faloutsos_1995] C. Faloutsos and K.-I. Lin, “FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets,” in ACM SIGMOD Intl. Conf. on Management of Data, San Jose, California, 1995.

[Faloutsos_2000] C. Faloutsos, B. Seeger, A. J. M. Traina, C. T. Jr., “Spatial Join Selectivity Using Power Laws,” in to be published in the ACM SIGMOD Intl. Conf. on Management of Data, Dallas, Texas, USA, 2000.

[Fayyad_1998] U. Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery," Bulletin of the IEEE Technical committee on Data Engineering, vol. 21, pp. 39-48, 1998.

[John_1994] G. H. John, R. Kohavi, K.Pfleger, "Irrelevant Features and the Subset Selection Problem," in 11' Intl. Conf. on Machine Learning, New Brunswick, NJ, USA, 1994.

[Kira_1992] K. Kira and L. L. Rendell, "A Practical Approach fo Feature Selection," in 9' Intl. Conf. on Machine Learning, Aberdeen Scotland, 1992.

[Langley_1997] P . Langley and S. Sage, "Scaling to Domains with many Irrelavant Features," in Computational learning theory and natural learning systems, vol. 4, R. Greiner, Ed. Cambridge, MA: MIT Press, 1997.

[Pagel_2000] B.-U. Pagel, F. Korn, C. Faloutsos, "Deflating the Dimensionality Curse Using Multiple Fractal Dimensions," in 16th Intl. Conf. on Data Engineering (ICDE), San Diego, CA - USA, 2000.

[Scherf_1997] M. Scherf and W. Brauer, "Feature Selection by Means of a Feature Weighting Approach," Technische Universität München, Munich 1997.

[Schroeder_1991] M. Schroeder, Fractals, Chaos, Power Laws, 6 ed. New York: W.H. Freeman and Company, 1991.

[Singh_1995] M. Singh and G. M. Provan, " A Comparison of Induction Algorithms for Selective and Non- selective Bayesian Classifiers," in 12' Intl. Conf. on Machine Learning, Lake Tahoe, CA, USA, 1995.

[Traina_1999] C. Traina, A. J. M. Traina, C. Faloutsos, "Distance Exponent: A New Comcept for Selectivity Estimation in Metric Trees," Carnegie Mellon University, Pittsburgh, P A CMU-CS-99-110, March 1, 1999 1999.

[Traina_2000] C. Traina, A. J. M. Traina, C. Faloutsos, "Distance Exponent: a New Concept for Selectivity Estimation in Metric Trees," in IEEE Intl. Conf. on Data Engineering (ICDE), San Diego - CA, 2000.

[Turk_1991] M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol. 3, pp. 71-86, 1991.

[Vafaie_1993] H. Vafaie and K. A. D. Jong, "Robust Feature Selection Algorithms," in Intl. Conf. onTools with AI, Boston, MA, 1993.

[Wactlar_1996] H. D. Wactlar, T. Kanade, M. A. Smith, S. M. Stevens, "Intelligent Access to Digital Video: Informedia Project," IEEE Computer, vol. 29, pp. 46-52, 1996.

AUTHOR PROFILE

DR. VENKATANATHAN DWARKANATHAN...

ANTWORKS CTO

Dr. Venkatanathan Dwarkanathan is an accomplished Mathematician with active engagement in research on topics like Chaos theory, fractal based pattern recognition and several others in applied mathematics, including dyna-systems theory, Fractal compositions and geophysical image dynamics.

Prior to joining Antworks, Dr. Venkatanathan Dwarkanathan held a broad range of leadership positions in mission critical portfolios in large organizations like National Aeronautics and Space Administration, Boeing, Aetna, Ausys Automation, Maples and iData Sciences.

He holds a Ph.D. degree in Mathematics (Astro-pad dynamics) and a post-doctoral work in Fractal Topology with profound expertise in building software frameworks which generates math models for applications in various verticals.

During the past 25+ years, he has developed many theories and built models for different verticals including; Aerospace (Image Sequencing), Roadways (Crack detection in JPEG images),Market Intelligence (Fractal Genesis), Healthcare (Mathematical Models for payers) and generic Data Capture Framework using Content Based Object Retrieval Methods

GET IN TOUCH

Headquarters:
Level 39, 10 Marina Boulevard,
Marina Bay Financial Centre Tower 2,
Singapore – 018983
Phone: +65 6818 5724

www.ant.works | hello@ant.works

ANTWORKS™
Reimagine IT