[[Profile photo for Brent Oster](https://www.quora.com/profile/Brent-Oster)](https://www.quora.com/profile/Brent-Oster" \t "_blank)

**[Brent Oster](https://www.quora.com/profile/Brent-Oster" \t "_blank)**

 ·

Follow

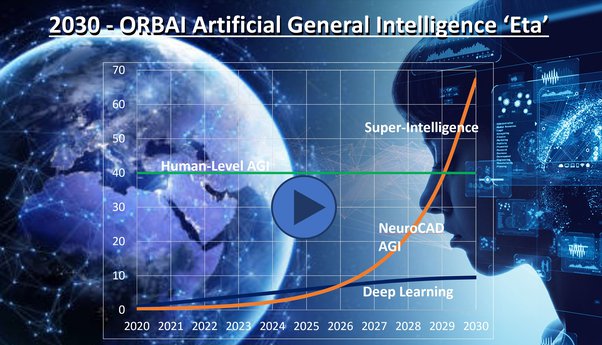
Former Deep Learning Solution Architect at NVIDIA (2008–2018)[Updated Jun 11](https://www.quora.com/Why-is-artificial-intelligence-a-hype/answer/Brent-Oster)

**[Why is artificial intelligence a hype?](https://www.quora.com/Why-is-artificial-intelligence-a-hype" \t "_blank)**

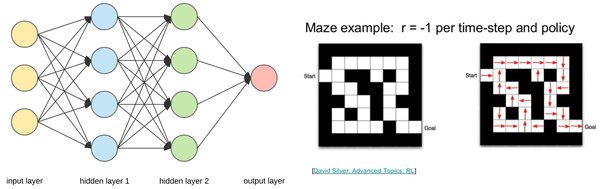
I worked for a decade on AI, first at NVIDIA, as a Solution Architect to research deep learning techniques, and present solutions to customers to solve their problems and to help implement those solutions. For the past 4 years I have been working with ORBAI on what comes next after DNNs and Deep Learning. I will cover both, showing how it is very difficult to scale DNNs to AGI, and what a better approach would be.

What we usually think of as Artificial Intelligence (AI) today, when we see human-like robots and holograms in our fiction, talking and acting like real people and having human-level or even superhuman intelligence and capabilities, is actually called Artificial General Intelligence (AGI), and it does NOT exist anywhere on earth yet.

What we actually have for AI today is much simpler and much more narrow Deep Learning (DL) that can only do some very specific tasks better than people. [It has fundamental limitations](https://www.scientificamerican.com/article/artificial-general-intelligence-is-not-as-imminent-as-you-might-think1) that will not allow it to become AGI, so if that is our goal, we need to innovate and come up with better networks and better methods for shaping them into an artificial intelligence.



Let me write down some extremely simplistic definitions of what we do have today, and then go on to explain what they are in more detail, where they fall short, and some steps towards creating more fully capable 'AI' with new architectures.



**Machine Learning**- Fitting functions to data, and using the functions to group it or predict things about future data. (Sorry, greatly oversimplified)

**Deep Learning** - Fitting functions to data as above, where those functions are layers of nodes that are connected (densely or otherwise) to the nodes before and after them, and the parameters being fitted are the weights of those connections.

Deep Learning is what usually gets called AI today, but it is really just very elaborate pattern recognition and statistical modeling. The most common techniques / algorithms are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Reinforcement Learning (RL).

**Convolutional Neural Networks (CNNs)** have a hierarchical structure (which is usually 2D for images), where an image is sampled by (trained) convolution filters into a lower resolution map that represents the value of the convolution operation at each point. In images it goes from high-res pixels, to fine features (edges, circles,….) to coarse features (noses, eyes, lips, … on faces), then to the fully connected layers that can identify what is in the image.

**Recurrent Neural Networks (RNNs)** work well for short sequential or time series data. Basically each 'neural' node in an RNN is kind of a memory gate, often an LSTM or Long Short Term Memory cell. RNNs are good for time sequential operations like language processing or translation, as well as signal processing, Text To Speech, Speech To Text,…and so on.

**Reinforcement Learning** is a third main ML method, where you train a learning agent to solve a complex problem by simply taking the best actions given a state, with the probability of taking each action at each state defined by a policy. An example is running a maze, where the position of each cell is the ‘state’, the 4 possible directions to move are the actions, and the probability of moving each direction, at each cell (state) forms the policy.

But all these methods just find a statistical fit of a simplistic model to data. DNNs find a narrow fit of outputs to inputs that does not usually extrapolate outside the training data set. Reinforcement learning finds a pattern that works for the specific problem (as we all did vs 1980s Atari games), but not beyond it. With today's ML and deep learning, the problem is there is no true perception, memory, prediction, cognition, or complex planning involved. There is no actual intelligence in today's AI.

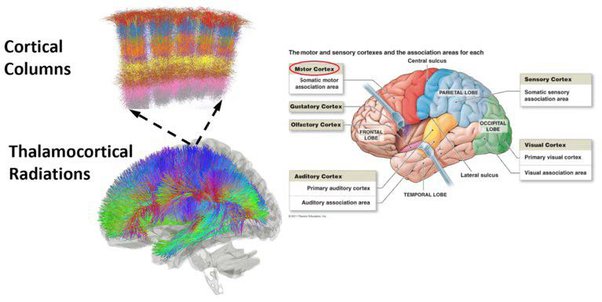
Here is a video of how deep learning could be overtaken by methods based on more flexible spiking neural networks (flexible analog neural computers), shaped by genetic algorithms, architected into an AGI, and evolved over the next decade into a superintelligence.

We propose an AI architecture that can do all the types of tasks required - speech, vision, and other sensors that could make a much more General Artificial Intelligence.

From our [AGI Patent:](https://www.orbai.ai/artificial-general-intelligence.htm) We specify a method for artificial general intelligence that can simulate human intelligence, implemented by taking in any form of arbitrary input data, the method comprising Learning to transform the arbitrary input data into an internal numerical format, then performing a plurality of numerical operations, the plurality of numerical operations comprises learned and neural network operations, on the arbitrary input data in the internal format, then transforming the arbitrary input data into output data having output formats using a reciprocal process learned to transform the output data from the arbitrary input data, wherein all steps being done unsupervised.

How does the brain handle vision, speech, and motor control? Well, it's not using CNNs, RNNs, nor Transformers, that's for sure. They are mere tinker toys by comparison.

First, the brain:

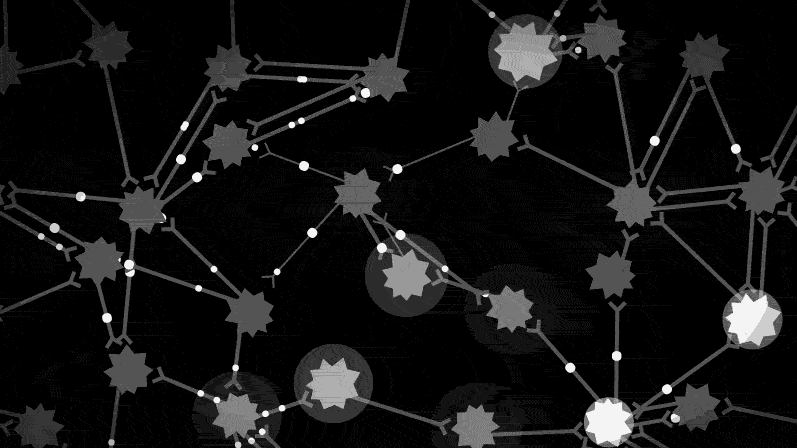


The brain is divided into distinct regions: the outer cerebral cortex is a sheet of neurons 4mm thick that is folded around the thalamocortical radiations below it like a pie crust around a head of broccoli. This cortex is divided into regions for vision, audio, speech, touch, smell, motor control, and our other external and internal senses and outputs. The cortex is composed of a million cortical columns, each with 6 layers and about 100,000 neurons and each column represents a computing unit for the cortex, processing a feature vector for the senses or motor control.

The cerebellum is like a second brain, tucked under and below the cerebrum, and it performs fine control of motor and emotional state. Many of the internal structures of the brain are more ancient and function independently of the cortex, like the brainstem, thalamus and other structures, controlling our core functions, drives, and emotions that are acted on by the rest of the brain.

The hippocampus and other parts of the brain’s memory system orchestrate stories or narratives from this representation to reconstruct memories of the past, predict fictional stories into the future. When we dream, our brain, directed by the hippocampus, creates fictional narratives that fill in the blanks in our waking knowledge and allow us to learn about and build models of our world that are much more complex and nuanced than we could without dreaming, helping us with planning our waking actions.

For our basic unit of synthetic neural computing, we will use spiking neural networks (SNNs), which model neurons as discrete computational units that work much more like biological neurons, fundamentally computing in the time domain, sending signals to travel before neurons, approximating them with simple models like Izhikevich ([Simple Model of Spiking Neurons](https://www.izhikevich.org/publications/spikes.htm)) or more complex ones like Hodgkin-Huxley ([Hodgkin–Huxley model - Wikipedia](https://en.wikipedia.org/wiki/Hodgkin%E2%80%93Huxley_model)) (Nobel Prize 1953).



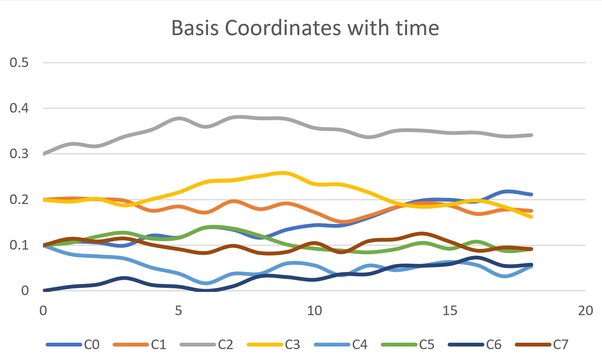
However, to date, application of spiking neural networks has remained difficult, as finding a way to train them to do specific tasks has remained elusive. Although Hebbian learning functions in these networks, there has not been a way to shape them so we can train them to learn to do specific tasks. Backpropagation (used in DNNs) does not work because all these spiking signals are one-way in time and are emitted, absorbed and integrated in operations that are non-reversible.

**Autoencoding Input and Output** - This section deals with how we transform real world data of various types - images, video, audio, speech, numerical data ... into a common internal format for the AGI core to process, and back to output again.

We need a more flexible connectome or network connection structure to train spiking neural networks. While DNNs only allow ‘neurons’ to connect to the next layer, connections in the visual cortex can go forward many layers, and even backwards, to form feedback loops. When two SNNs with complementary function and opposite signal direction are organized into a feedback loop like this, Hebbian learning now helps train them to become an autoencoder, that is able to encode spatial-temporal inputs such as video, sound or other sensors, and reduce them to a compact machine representation and then decode that representation into the original input and together provide feedback to train this process. We called this Bidirectional Interleaved Complementary Hierarchical Neural Networks or BICHNN.

The autoencoder learns to transform video, audio, numerical and other data against a learned set of feature or basis vectors stored internally within the autoencoder, outputting a set of basis coordinates that represent the weights of those features in the present input stream.

These weights become time-series basis coordinates (TSBCs) that represent a vector narrative or memory stream that can be processed by more conventional computer science and numerical methods as well as by specialized SNNs for predictors and other solvers internal to the AGI.

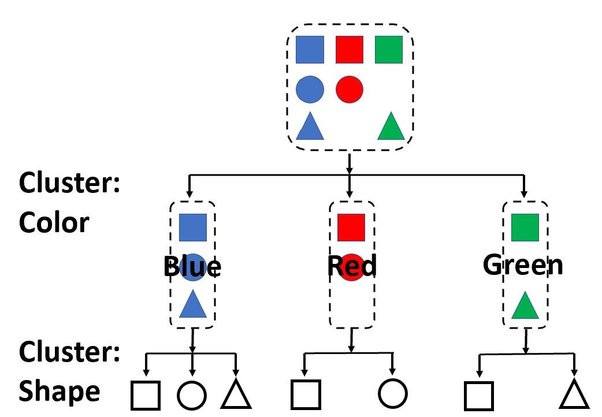


The autoencoder runs in reverse as well, transforming TSBCs back to native world data for output or to drive actuators in robotics and autonomous vehicle applications.

In practice, it may be more computationally tractable to use a hierarchy of autoencoders and PCA axes to break the encoding into a series of steps where intermediate results are sorted by the most defining feature, then further encoded to extract ever finer and more nuanced features from the data. We term this the Hierarchical Encoder Network or HAN in the book (see comment for details).

The cortical columns of the cerebral cortex are analogous to our terminal layer of autoencoders, a map storing the orthogonal basis vectors for reality and doing computations against them, including computing basis coordinates from input engrams. Our version of the thalamocortical radiations is a hierarchy of alternating autoencoders and principal component axes that we term the HAN.

Here is a simple example of the HAN learning to encode data with shape and color as the features it classifies them on. First the autoencoder runs on the input data and learns to transform it into an internal format and back, in the process, learning that there are 7 basis vectors, or types of data: blue squares, circles, and triangles; red squares and circles; and green squares and triangles.



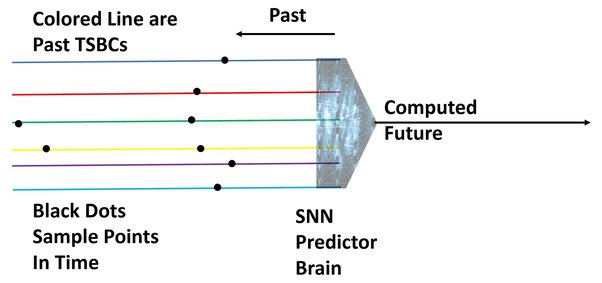
First it sorts the data based on one feature axis - color, then runs an autoencoder on those clusters of data and learns that the first cluster consists of squares, circles, and triangles (not encoding the color as a feature because it is common to all), and then learning that the second cluster consists of squares and circles, and the third cluster consists of squares and triangles.

Now by having an index at each of the bottom basis vectors, we can uniquely identify the original data, and we can reconstruct it from the features. This simple HAN learned that there were two feature axes - color and shape, and that there were three colors - blue, red, and green, and that there were three shapes of which some belonged in each color group but not others.

In real operation, there can be hundreds of axis, thousands of clusters, each picking out more and more detailed features about the data as it cascades down the HAN until it has been reduced to the finest possible feature set - the basis vectors.

**AGI Core** - Now we use these time-series basis coordinates (TSBCs) as code and data in sets of modules including SNN stream processors that operate on that data, including predictors, dreamers, and other methods for processing that data.

Predictor SNNs learn to predict future TSBCs from past TSBCs by training on known data and building a model of that data that is more than just a statistical data fit like DNNs, but rather a custom analog SNN computer that learns to model the data.

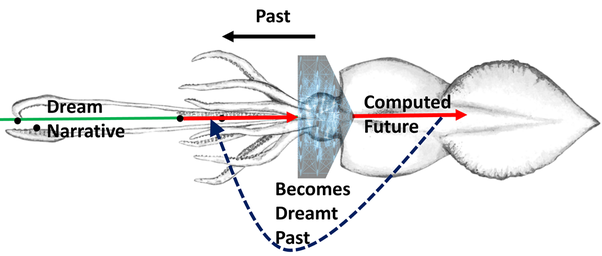


Dreamer SNNs are a predictor that is trained to model the reality behind the data in the same manner as the predictor, but who's input is short-circuited from their output prediction such that the entire TSBC they are creating is simulated, fictional, a dream.

The Dreaming SNN, it is based on real neuroscience. Dreaming fills in blanks between our memories from experienced reality by modeling reality in simulated narratives.

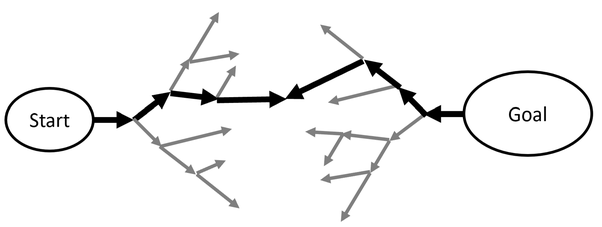
The book ”When Brains Dream” describes some very interesting neuroscience research in this area by Antonio Zadra and Robert Stickgold. They propose a model for memory and dreaming called NEXTUP which states that during REM sleep, the brain explores associations between weaker connected memories via fictional dream narratives that, while not meant to solve immediate problems, nor necessarily even incorporate waking experiences explicitly, lays down a network of associations that will aid in future problem solving, whether or not we consciously recall the dreams themselves or not.

Our Dreaming SNN Predictor (see the diagram) starts at the present experienced memories (or from a playback of past memories) on a TSBC, and predicts the next memory in a fictional narrative TSBC, then moves the whole pipeline one step into the future, treating the last-predicted memory as ‘present’, and just keeps on predicting into the future. Soon the future predictions are only indirectly based on the actual previously experienced reality and become a fictional (but consistent) narrative based on the Dreamer SNN's model of reality, that drifts in its own directions.



This mimics the process in REM dreaming, laying down fictional narratives or TSBCs in this case, filling in the blanks between experienced reality and making for a more complete model of that reality. That model, and the experienced and simulated memories can be used in future predictions and problem solving.

Lightning solvers start at two different points on TSBCs and branch towards each other like lighting leaders, with the first leaders to make contact forging the path to a solution.



All of these solvers are evolved in an internal genetic algorithm that the AGI uses to learn to solve problems. By running genetic algorithms to create and chain together modules like the above that operate on TSBCs, the AGI evolves a library of modules that can do arbitrary operations on data to produce the desired results, and accomplishes transfer learning by applying modules evolved on other problems to new, similar problems, getting exponentially better as it builds up its internal, evolved library of solver modules and configurations.

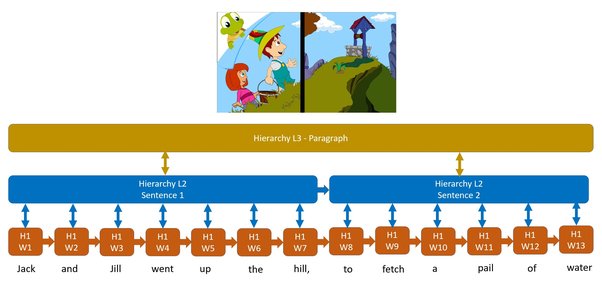
Then, the TSBC's can be transformed back to real-world data by the SNN Autoencoders and HAN and used as outputs or as signals to drive actuators, again with training by the same methods used as inputs, only back-driving the desired output signals.

This combination of I/O autoencoders that can transform diverse types of real-world data into a common internal TSBC format, and evolved families of process modules that learn to shape it into outputs and re-transform it into outputs via autoencoders provides the foundation for our AGI design, with all components learning their form and function from their environment and transferring their learning to new forms of data and tasks as the AGI advances and evolves.

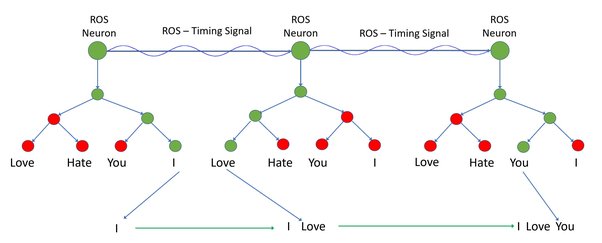
**Data Abstraction and Language** - This section is about abstract reasoning, hierarchical categorization of data, and language, and how we model them in our AGI.

As input comes in and is processed by the Hierarchical Autoencoder Network, it can be sub-sampled using temporal and relational autoencoders that condense the timeline and events into more abstracted versions, then output Hierarchical Temporal Basis Coordinates that incorporate this data so that the HTBSCs can be read at multiple levels of abstraction and detail, and even linked together at their higher levels of abstraction where they are temporally, spatially, or conceptually coincident.

Language is an example. It can be represented at the lowest level as a stream of letters (for text), or phonemes (for speech), which can be hierarchically structured as words, phrases, sentences, and paragraphs as in our Jack and Jill example, where the highest levels of abstraction are linked to visual information like in the picture (or video), and to similar narratives.



Another way of creating this hierarchical abstraction is to use an Rank Order Selective -Inhibitor Network or ROS-I to create a hierarchical inhibitory network of basis sets built up from the most granular components of memory (letters, phonemes in language), to higher abstractions that combine these basis to make words, phrases, sentences, and paragraphs.



In our artificial ROS-Inhibitory network, a linear series of artificial ROS neurons fires in sequence, generating an excitatory signal when each one fires, causing each root neuron in the attached inhibitory neural network to fire, and as the signal cascades down that inhibitory neural network, it is selectively inhibited by an external, time domain control signal at each neuron, by modulating the neuron’s outgoing signal by its inhibitory signal. Overall, this selects which branches of the hierarchy are activated, by controlling the inhibition at each neuron in that hierarchy.

By repeatedly training this system on a set of speech inputs, with the input to the terminal branches of the ROS-Inhibitor network reaching and training the lower levels first, then percolating upward, it would first learn a sequence of phonemes, then progressively whole words, phrases, sentences, and larger groupings, like a chorus in a song, or repeated paragraphs in legal documents. Or the commands for the actuators could be back-driven through the motor control ROS-Inhibitor network to train control signals for robotics applications.

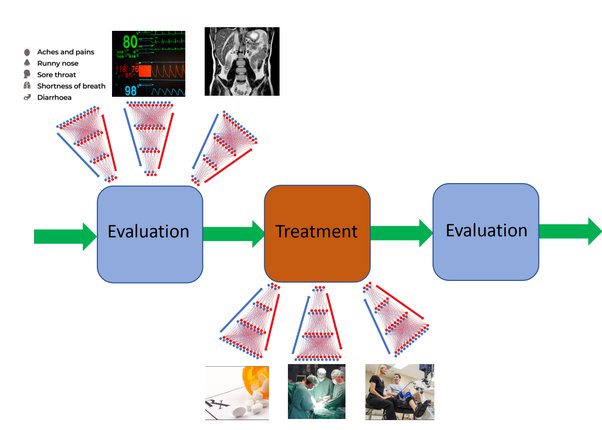
Once trained, our system can be run forward, with the ROS / excitatory neurons firing in sequence, and playback of the trained inhibitory signals modulating the activity of the neurons in the network to create a sequence of phonemes, words, phrases and paragraphs, to reproduce video from synthetic memories, and control motion by blending hierarchical segments (directed by the AI) to generate the words or motion.

The temporal inhibitory signals are a transformation of the TSBCs that puts them into a hierarchical format that can form temporal basis sets whose hierarchical combinations via the ROS-I system can simulate more complex output for motion, text, speech and other temporal data.

Language is a type of memory narrative (or HTSBC in our AGI) that forms the backbone for all other forms of narratives, not only labelling the data with that language, but forming a cognitive monologue by which we construct our thoughts and actions – the same language monologue that our AGI’s methods and processes operates on.

By organizing the internal data hierarchically, with the higher levels of the hierarchy abstracted and cross-linked to similar abstract data, and language being the backbone of that data, we go beyond a simple computer crunching series of numbers and allow our AGI to explore the higher-level relationships between objects, sequences, events, and the language describing them and tying them together.

This use of such abstraction and language leads to an AGI that can converse naturally with a human with fluid and fluent speech, and also allows reasoning and planning in many human professions like medicine, finance, and law.



So what impact will a AGI superintelligence have on the world? By developing an AGI that can perceive the real world, reduce those perceptions to an internal format that computers can understand, yet still plan, think and dream like a human, then convert the results back to human understandable form, and even converse fluently using human language, enabling online professional services in finance, medicine, law, and other areas. It can also add these enhanced analytics, forecasting, and decision-making capabilities to financial forecasting and enterprise software - where it can be used by businesses large and small.

Problems of wealth inequality, poverty, hunger, injustice, and lack of basic services for healthcare and information services are the norm for 3/4 of the people in the world. For millennia, human civilization has been unable to solve these basic problems, no matter the form of government or choice of deity and belief system, people are just unable to see the larger picture, and helpless to do anything about it.

Over the next decade, a superintelligence, a Strong Artificial General Intelligence, will evolve to oversee a global network augmenting the systems of Law, Medicine, Education, Finance and all previous human administrative functions. With its vast, wide, and deep knowledge reach, the wisdom to draw on all this past knowledge to plot possible paths into the future, this superintelligence will serve all of humanity and to help us make carefully measured & unbiased choices to guide us, and help us govern our affairs with focused, insightful information and an unprecedented ability to look into the future.

To see more: ORBAI ([About Us](http://www.orbai.ai/about-us.htm)) or for any questions, feel free to contact me at: [brent.oster@orbai.com](mailto:brent.oster@orbai.com)