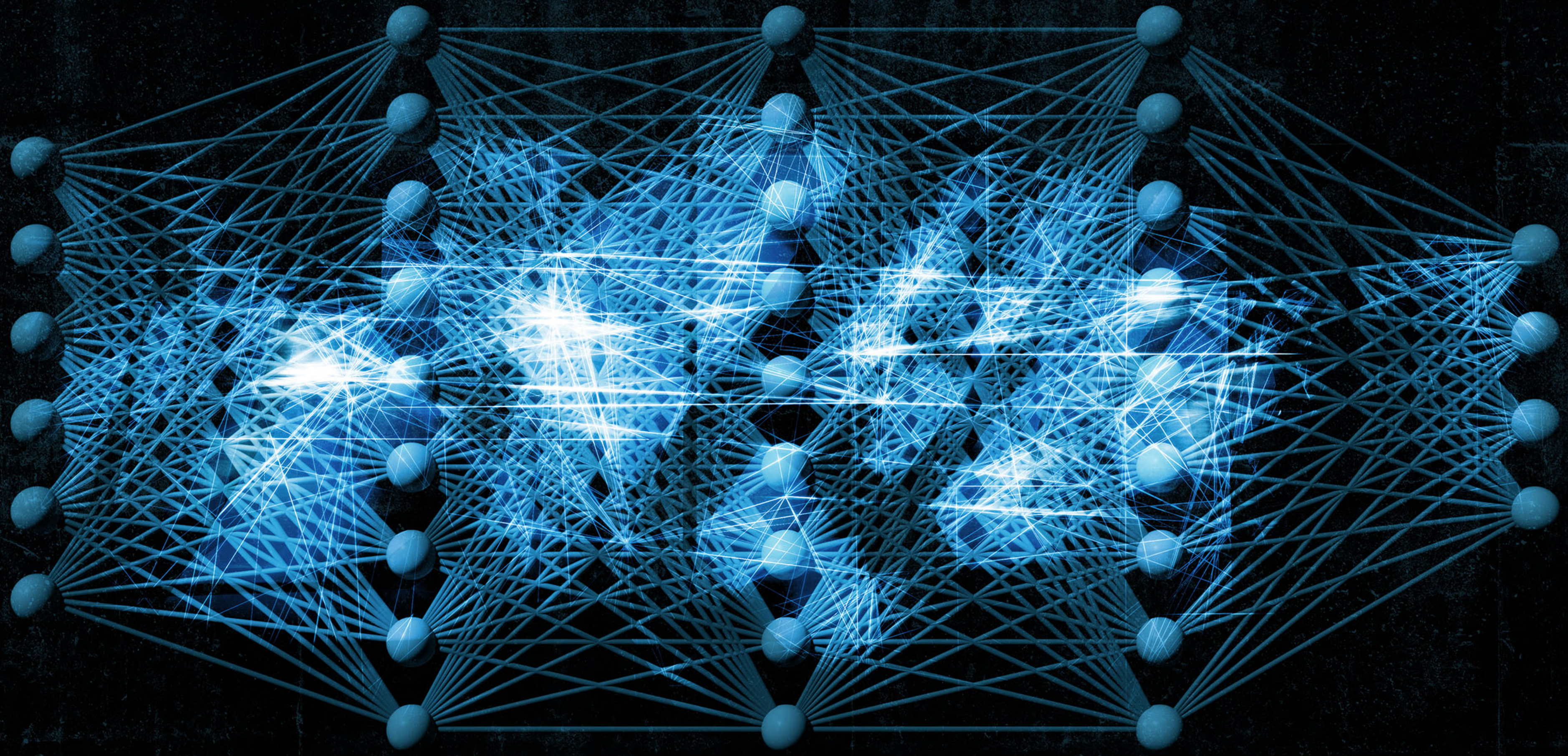
NeuroCAD for Spiking Neural Network



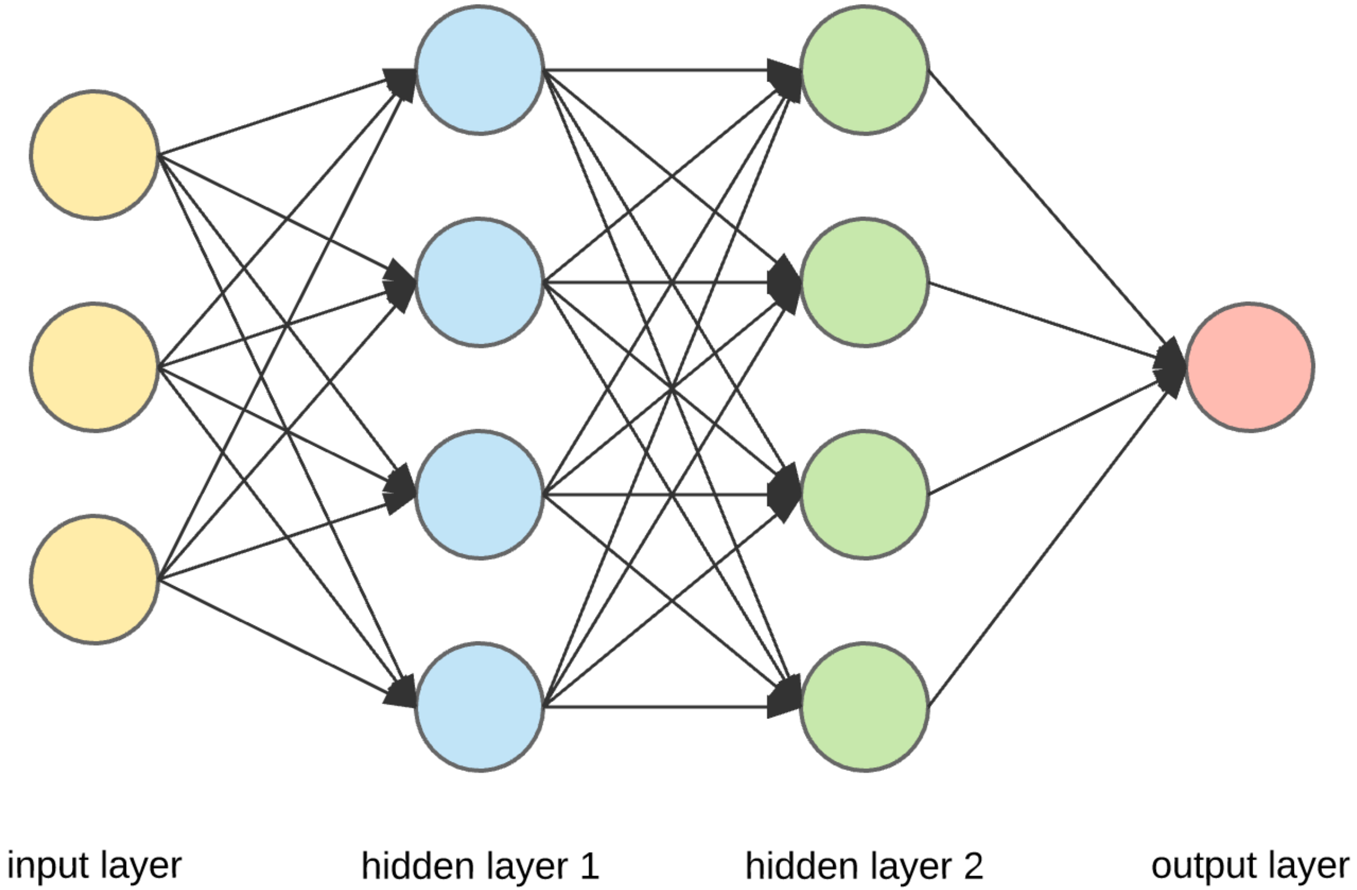
Bidirectional Interleaved Complementary Hierarchical Neural Networks

Brent Oster, SinduKumari, ORBAI

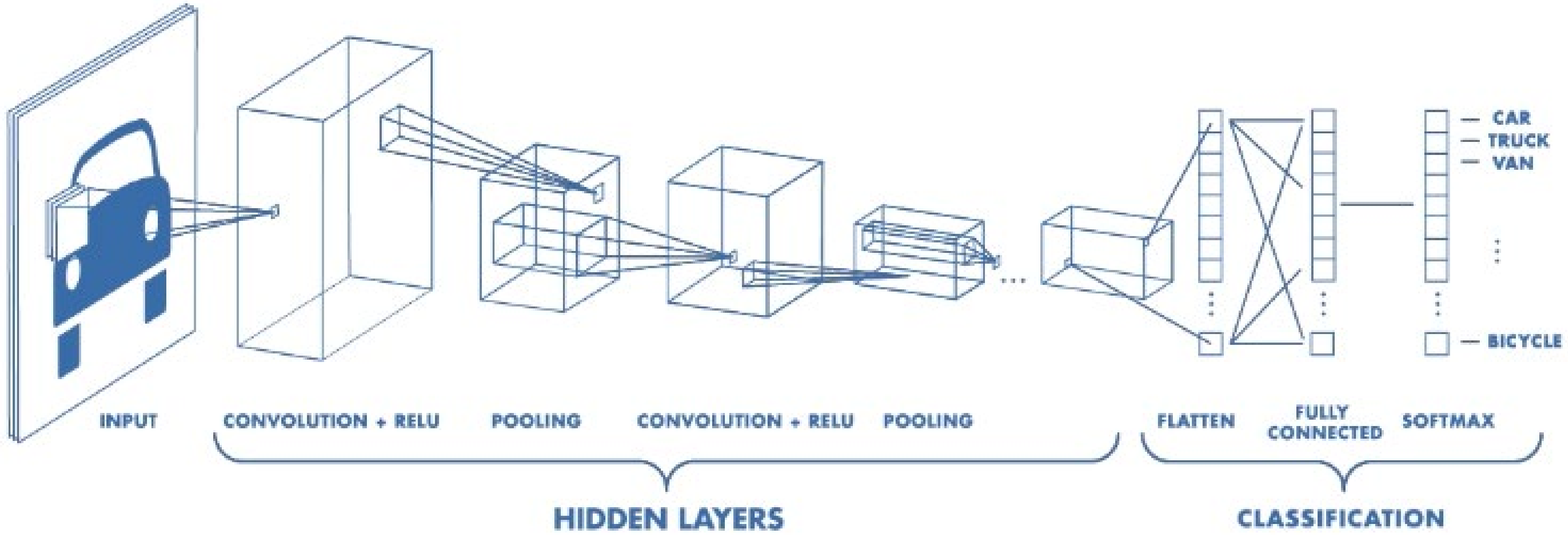
What is Artificial Intelligence?

- Computer simulation that can do useful operations and tasks
  - Learn how to perform these tasks without explicit instructions
  - Learn by doing, on-the fly, from practice and experience
  - Learn to do a wide variety of tasks that humans can do
  - Have cognition, intuition and able to estimate given sparse information
  - Be able to control physical robots, drones, etc. intelligently
- Is Deep Learning artificial intelligence?

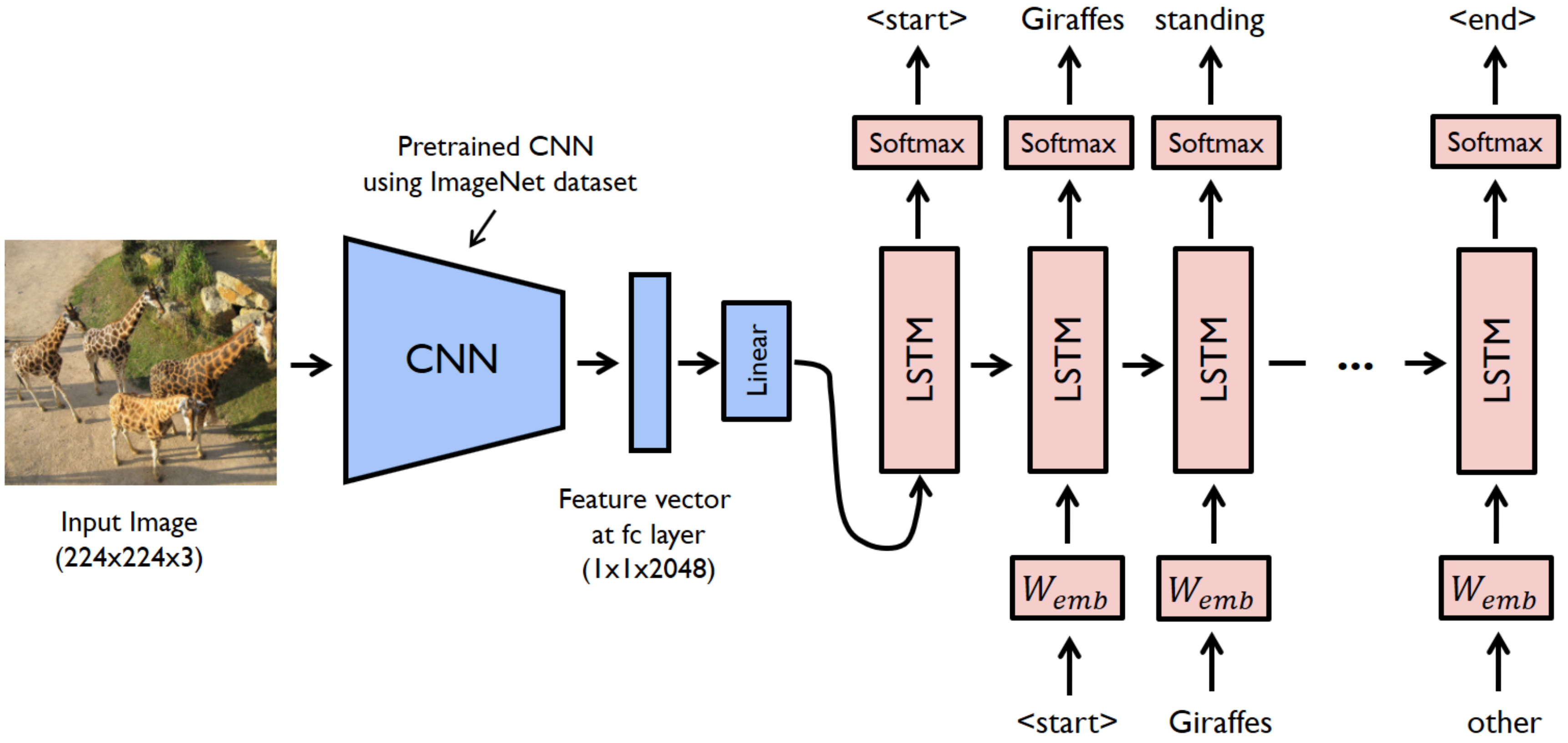
Deep Learning with 'Neural' Networks is State of Art Today



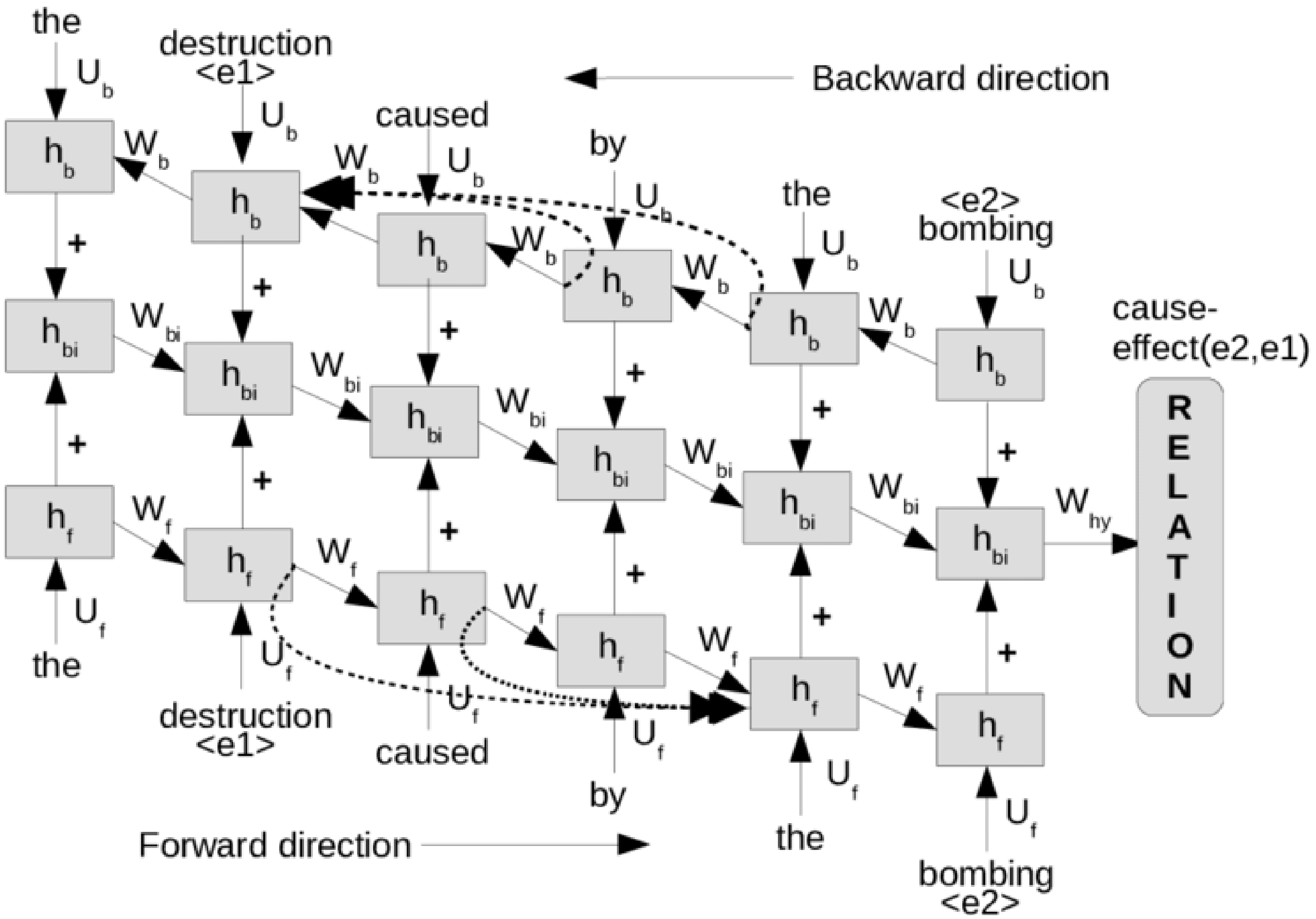
Convolutional Neural Networks – Image Recognition



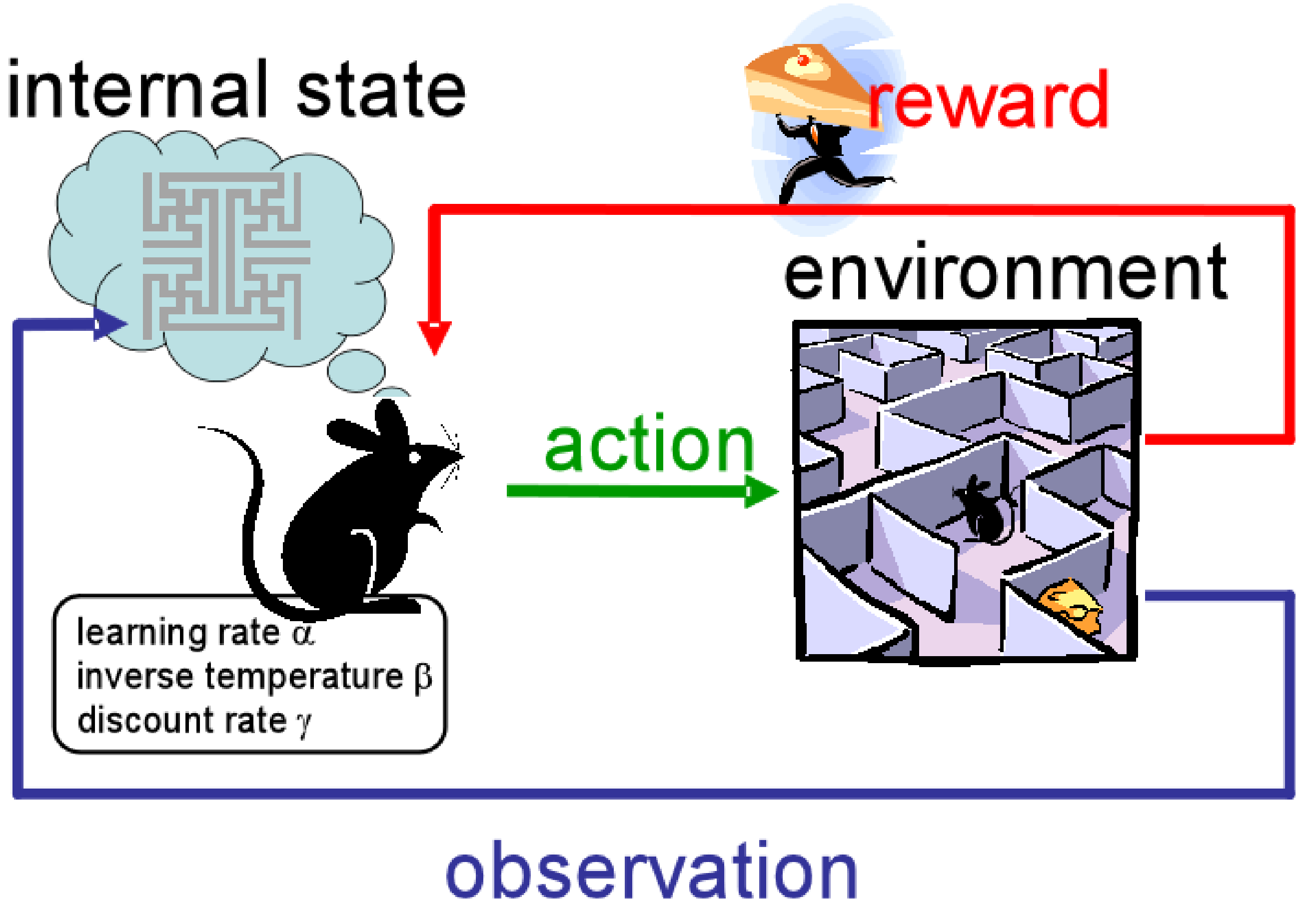
CNN –RNN Hybrid for Vision



Recurrent Neural Networks – Language, Speech



Reinforcement Learning – Control AI



Generative Adversarial Neural Networks (GAN) Unsupervised (Dynamic?) Learning?

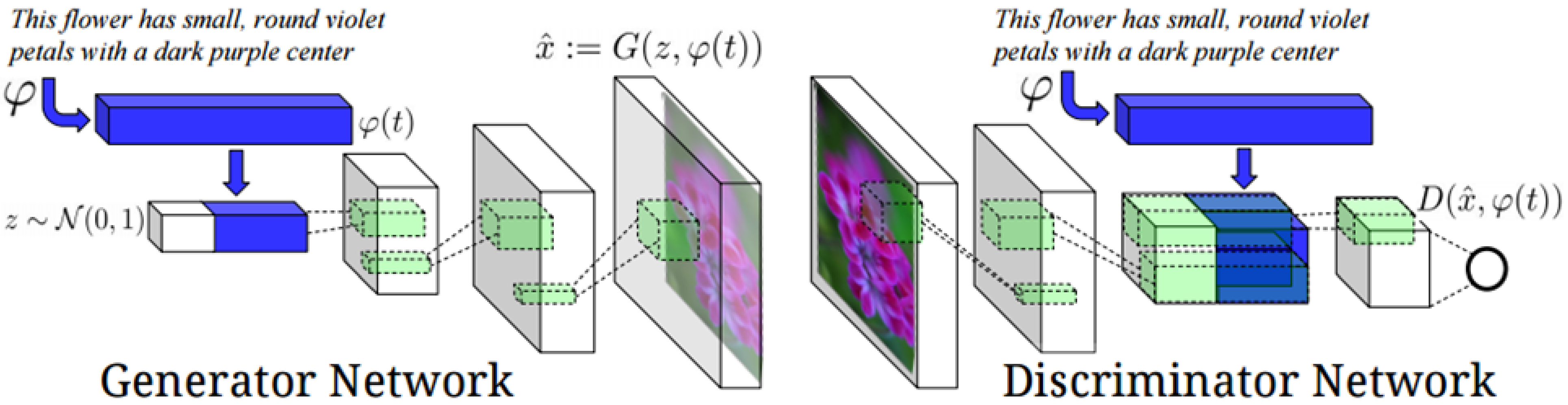


Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Network Architecture

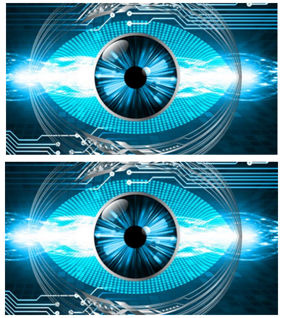
Performance Capture Human to Train Robot AI?

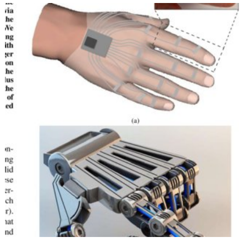
Intensive Performance Capture of Individual

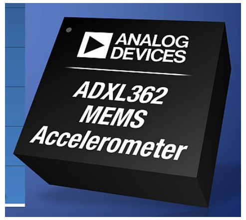
Use as Training Dataset for Android Mimic AI

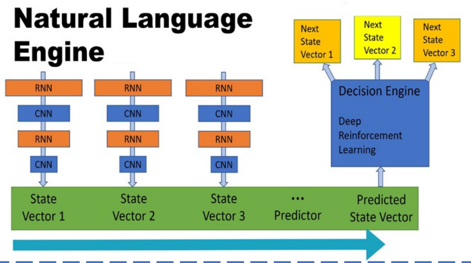
Motion
Facial Expressions
Voice & Speech
Mannerisms

Building a Humanoid Robot AI with Deep Learning Tech

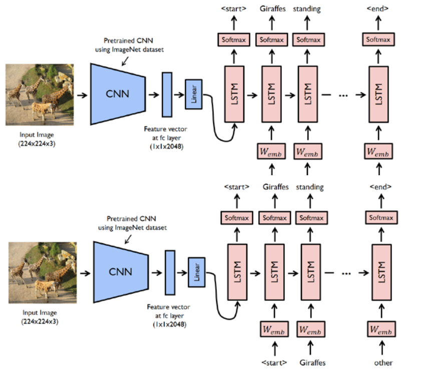






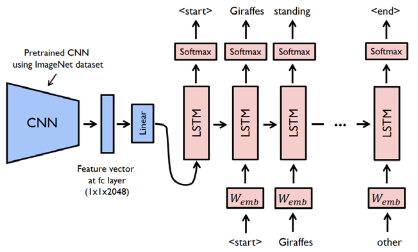


Stereo Vision

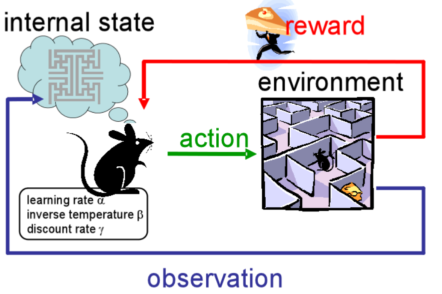


Dual CNN-RNN?

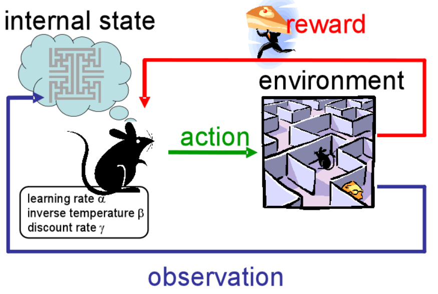
CNN-RNN for Sensors



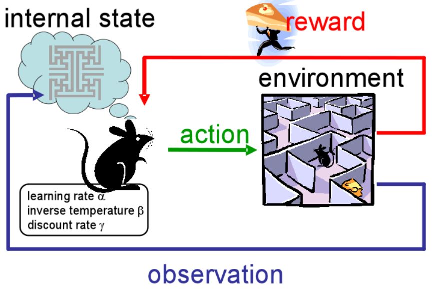
High Lvl Planning DRL



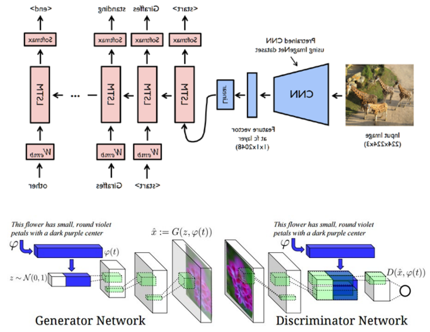
Macro Motion DRL



Animation Cntrl DRL

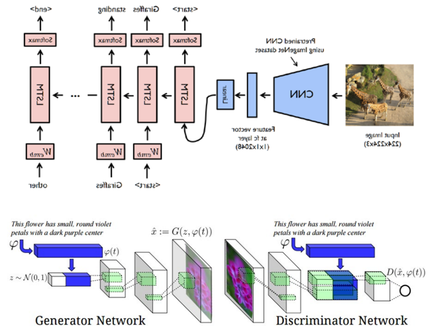


Facial Controller



Inv RNN-CNN-GAN?

Body Controller



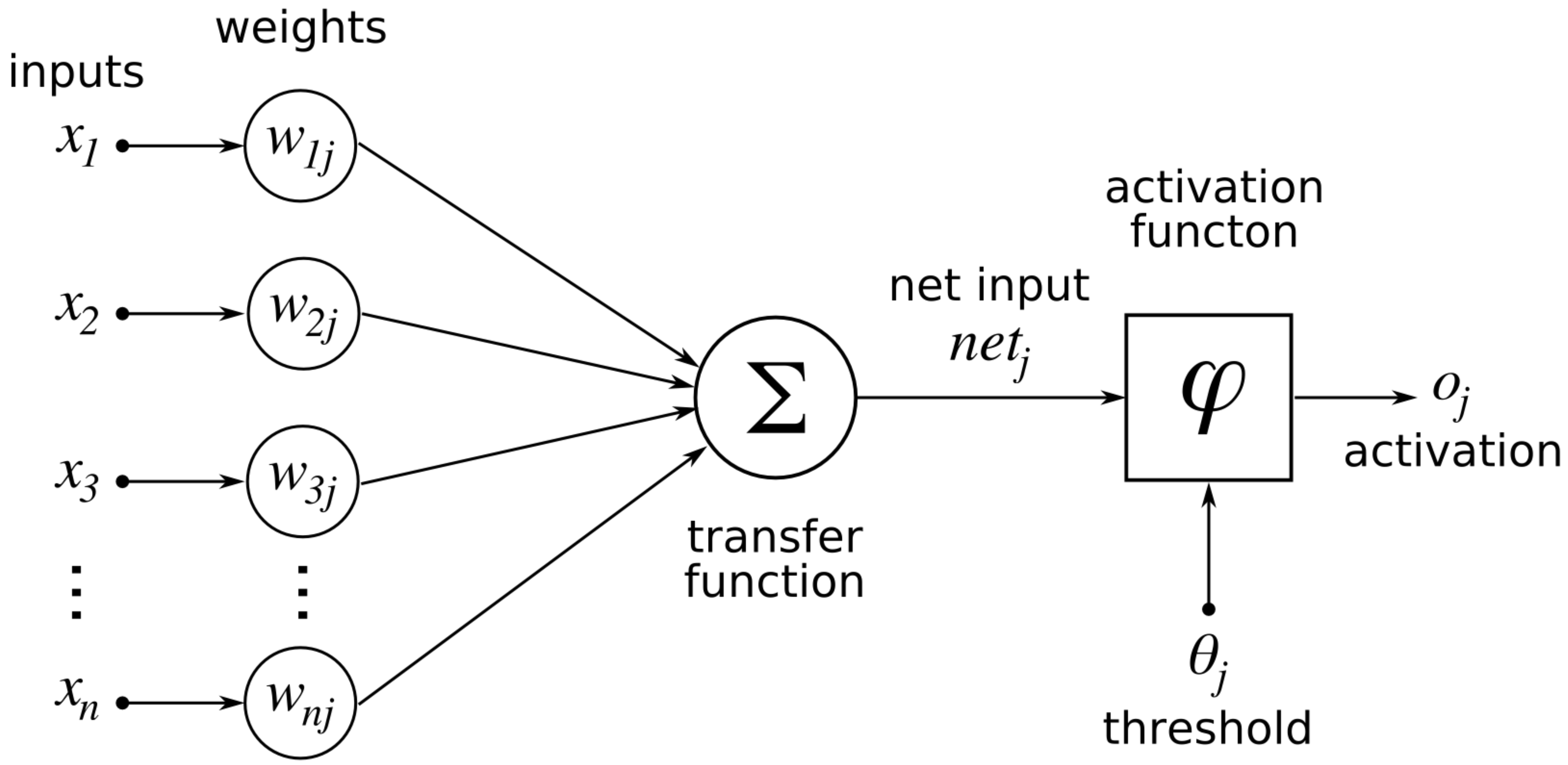
Inv RNN-CNN-GAN?

Deep Learning is NEVER Going Work for THAT!

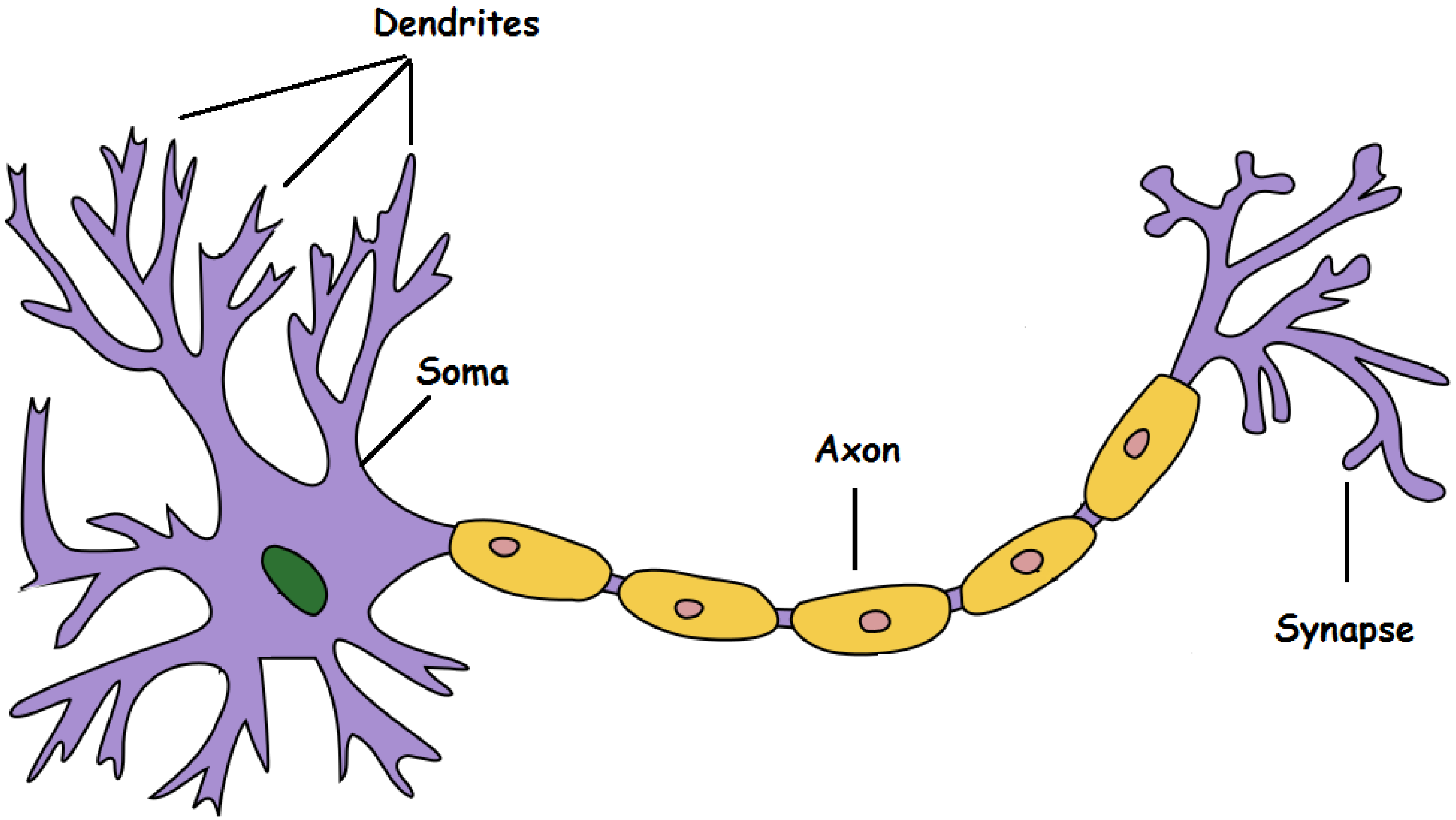
- **Deep Learning is only able to:**
  - **Learn from structured, formatted, and usually labelled data**
  - **Do very narrow tasks within the domain of that data**
  - **Requires large amounts of data to make accurate predictions**

- **Deep Learning CANNOT:**
  - **Learn to do general tasks or multiple tasks with same network architecture**
  - **Does not work well on unstructured real-world data**
  - **Can not stack multiple layers of DL implementations and have it train**
  - **Learn from experience in a real-life dynamic environment**
  - **Have cognition, intuition, or operate with sparse data, reach human AI**

Deep Learning ‘Neurons’ Are Too Simplistic



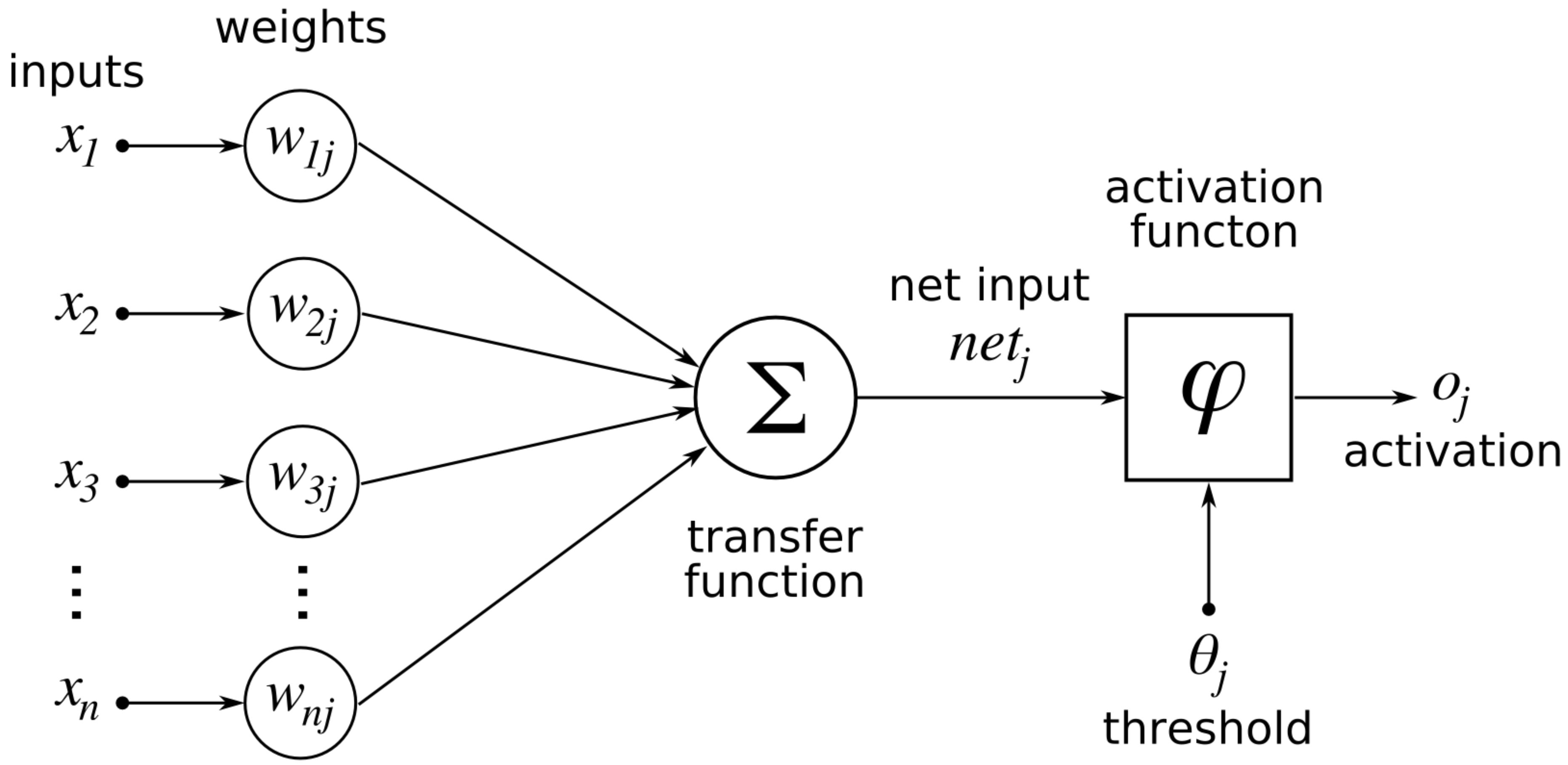
Real Biological Neurons are Very Sophisticated Electro-Chemical Computers

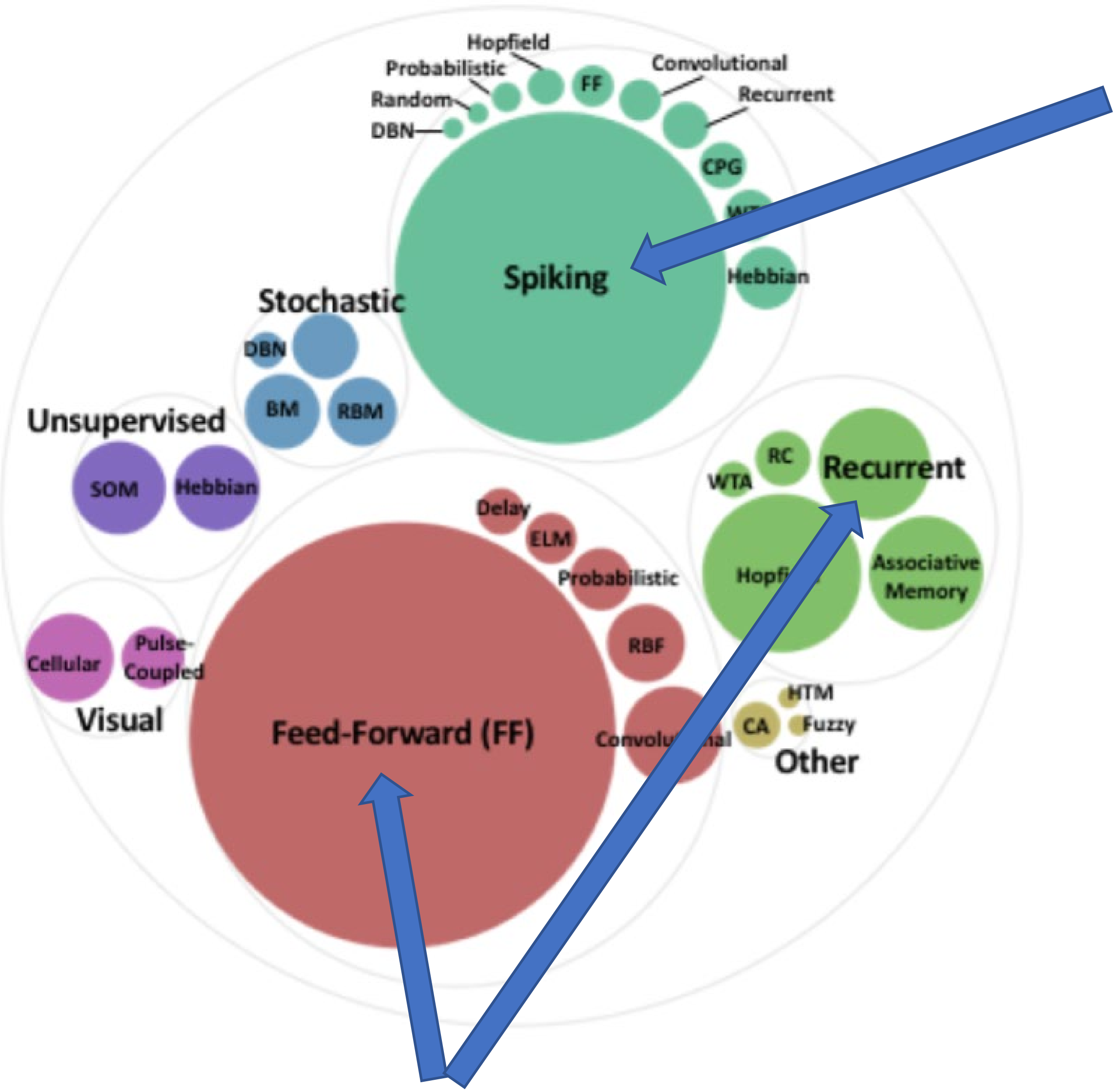


How Does a Biological Neuron Work (roughly)?

- The neuronal body integrates inputs from the dendrites coming into it
  - Integrates incoming signals in both **space and time**
  - Some dendrites excite, some inhibit, adding or subtracting from the potential
  - Neuronal body 'fires' when action potential (-55mv) is reached across cell wall
- When the neuronal body fires, a spike train is transmitted down the axon
  - Transmitted along axon, branches, and is amplified (and modified) along the way
  - Signal in **time and space** that carries more information than a simple amplitude
- Spiking signal is further modified at synapse
  - Axon spike train stimulates neurotransmitter release from pre-synaptic side
  - Neurotransmitters drift across synapse, modified by ambient neurochemistry
  - Receptors on post-synaptic side integrate chemical signal, firing at a threshold
  - A spike train propagates down the dendrite to the next neuron
  - If both the pre and post synaptic neuron fire close together: synapse strengthens

Do you still call this a ‘Neuron’?





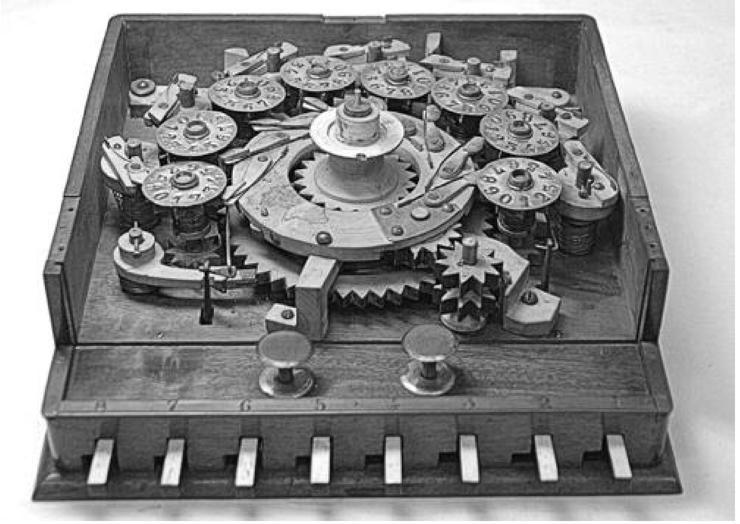
Spiking Neuron Models

Behave more like real neurons

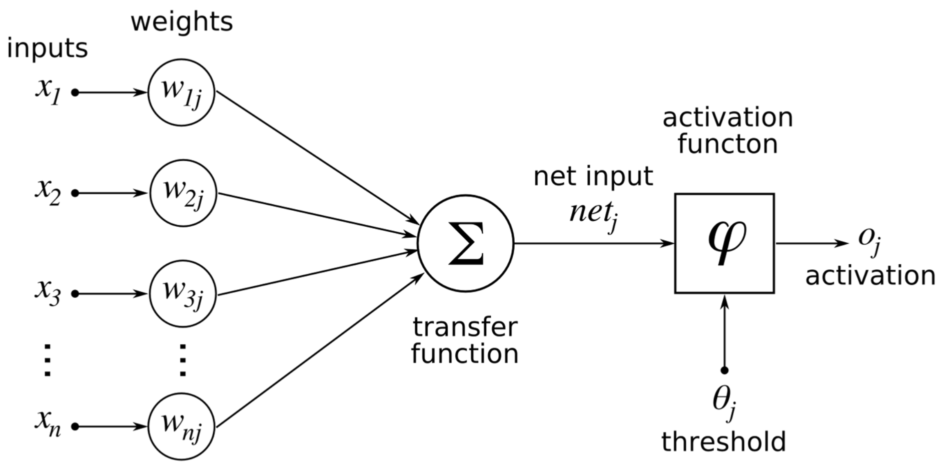
-Time-domain signals that propagate
- Information encoded in spikes
- Time-domain integration of spikes
- Integration in neuron and synapse
- Complex signal processing system
- Time dependency, lag in signals
- Allows waves, cascades, feedback
- Synapses that strengthen with use
    - Hebbian Learning
- Unsupervised associative learning

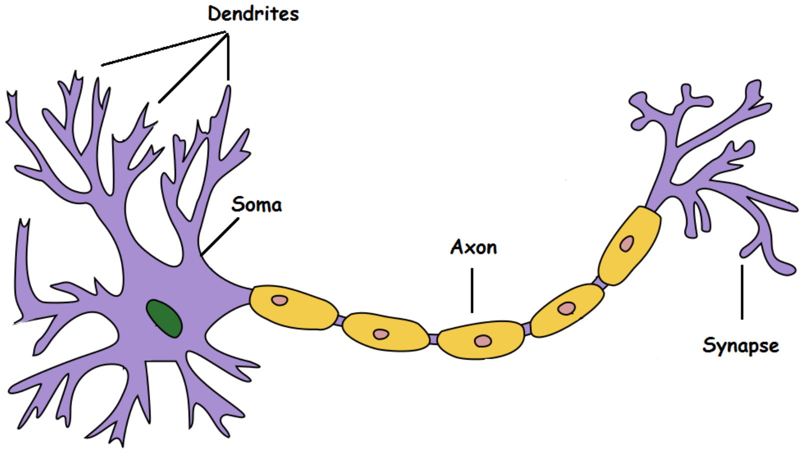
DL Uses Only a Subset of Artificial Neuron Models

A Spiking Neuron is More Like a Biological Neuron

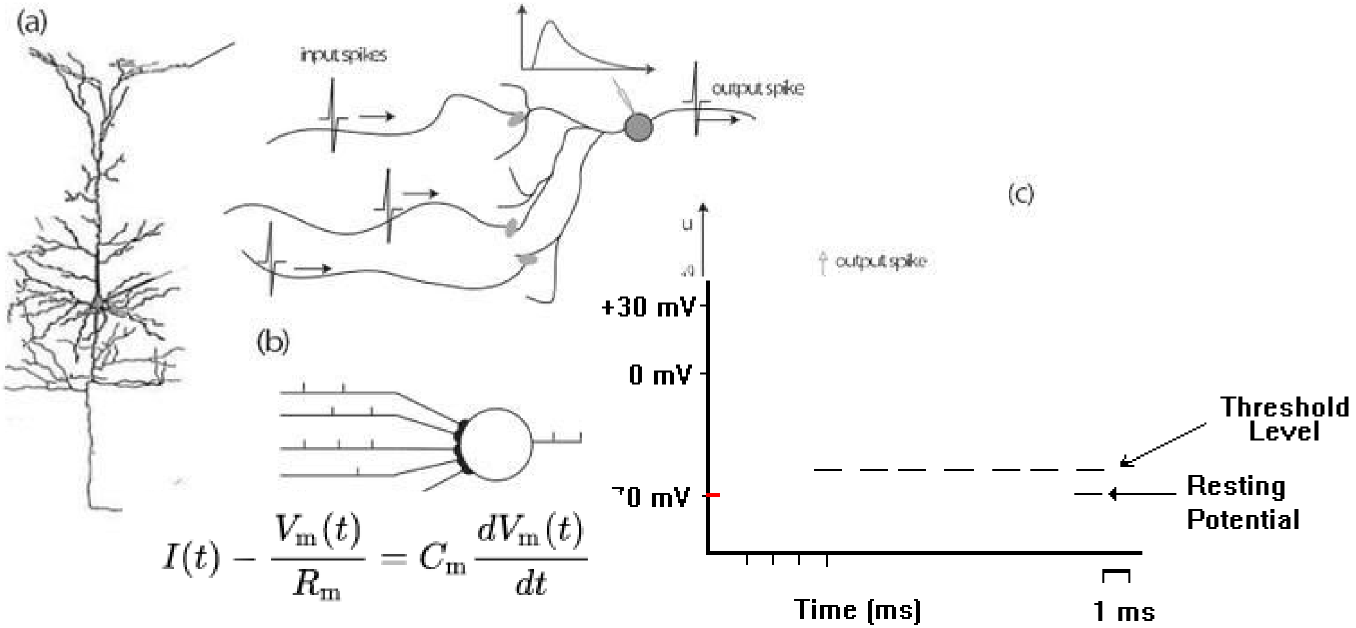


Deep Learning Neuron





Spiking Neuron



Link to BICHNN Demo

https://youtu.be/bthVbbbV_PM

NeuroCAD Synapse Model
‘Leaky Watering Can’

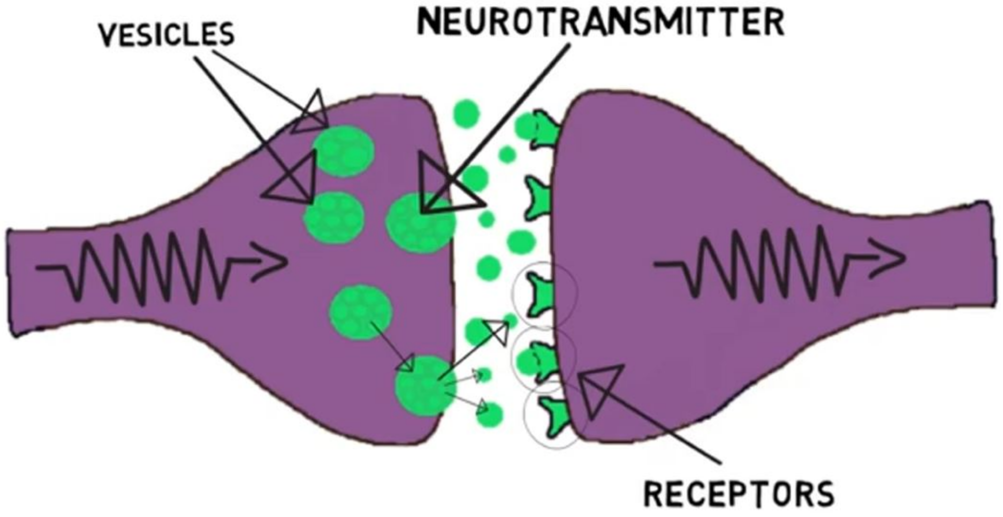
Variables
L = neurotransmitter level
Ls = neurotransmitter saturation level
ts = time since input spike

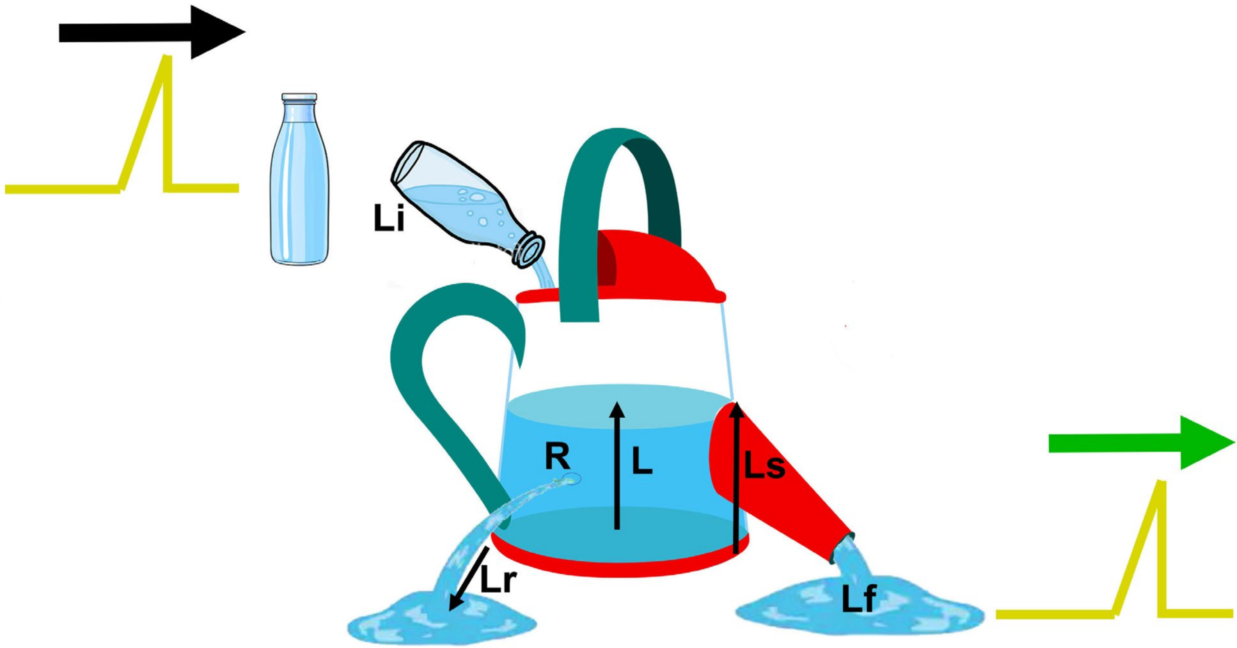
Constants
R = reuptake rate
Li = neurotransmitter input from spike
Lf = neurotransmitter 'consumed' when firing
tth = threshold time to learn within
Ft = fraction to reduce Ls by when training

```
L = L * (1 - R)
ts += dt;

If (spike comes in)
  L = L + Li
  ts = 0

if (L > Ls)
  (Fire spike down dendrite)
  L = L - Lf
  if (ts < tth)
    Ls = Ls * Ft
```





So, How Do We Train Spiking Neural Networks?

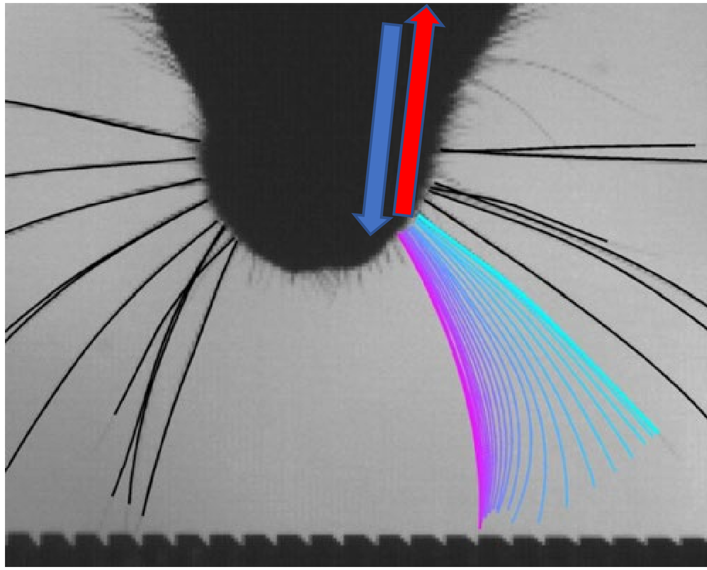
- This has remained an unsolved problem since they were developed in 1955
- Most Deep Learning uses back-propagation
  - Data is fed forward through the network and produces an output
  - A difference is computed between that output and a known label for the data
  - That difference is fed backwards through the network, adjusting the weights
  - This is repeated many times for the entire dataset till weights converge
- Back propagation does NOT generally work with spiking neural nets
  - SNN signals propagate in time, with complex integration at neuron and synapse
  - There is no way to back-drive these signals, compute derivatives and adjust weights
  - But somehow all moving life on earth manages to learn with a similar architecture
  - Hebbian learning – if pre & post synaptic neuron fire together, synapse strengthens
  - But this only allows the entire network to learn if it is first properly structured

The Quest for a Spiking Neural Net That Can Learn

Miguel Nicolelis – Brazilian Neuroscience Researcher
World expert in brain-machine interfaces, and measurement

Tickling a Rat's Whiskers

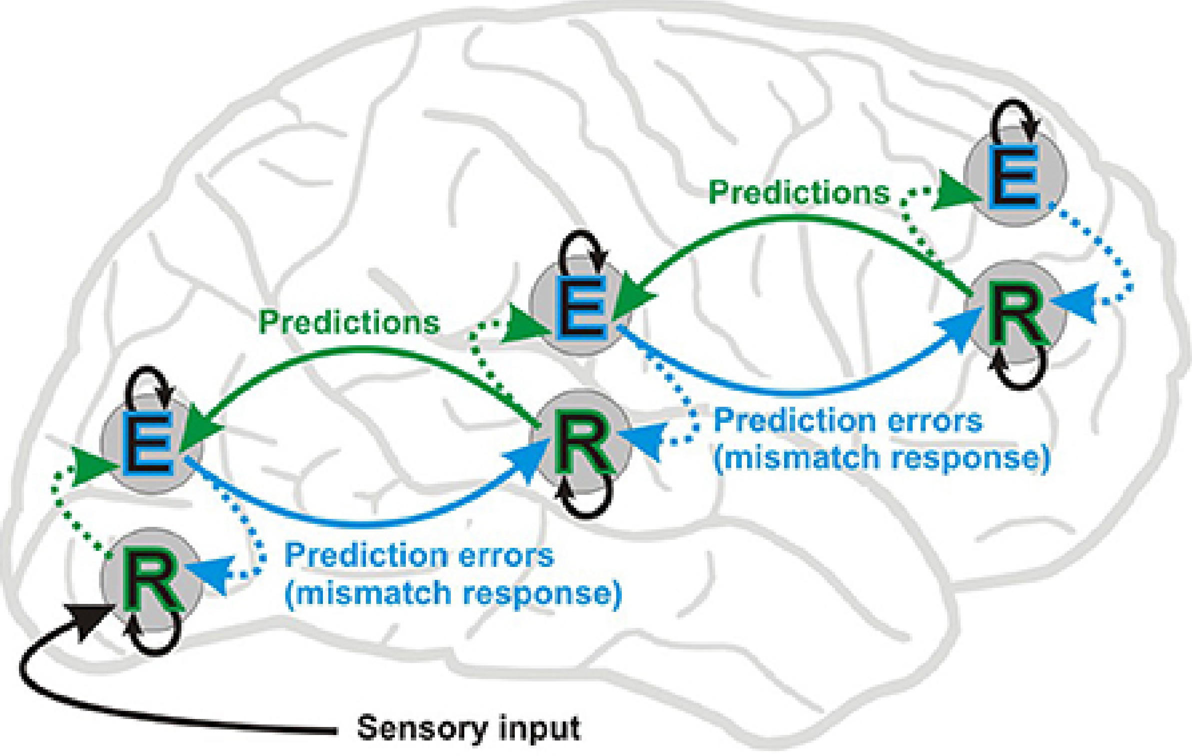
- Measuring neurological response to stimulating a rat's whiskers
- Probes were inserted at various spots in the neural path and brain
- Researcher would stimulate the rat's whiskers
- Probes could watch the signal travel from the whisker to the brain
- But there were also signals moving from the brain to the whisker
- Even when the whisker was not being stimulated, they were there
- The signal from brain to whisker was predicting the stimulus
- The two neural networks were interacting!
- Comparing the prediction and stimulus 'trains' the neural net how to perceive and predict the environment!
- EUREKA! Is this how the mammalian sensory cortex trains?

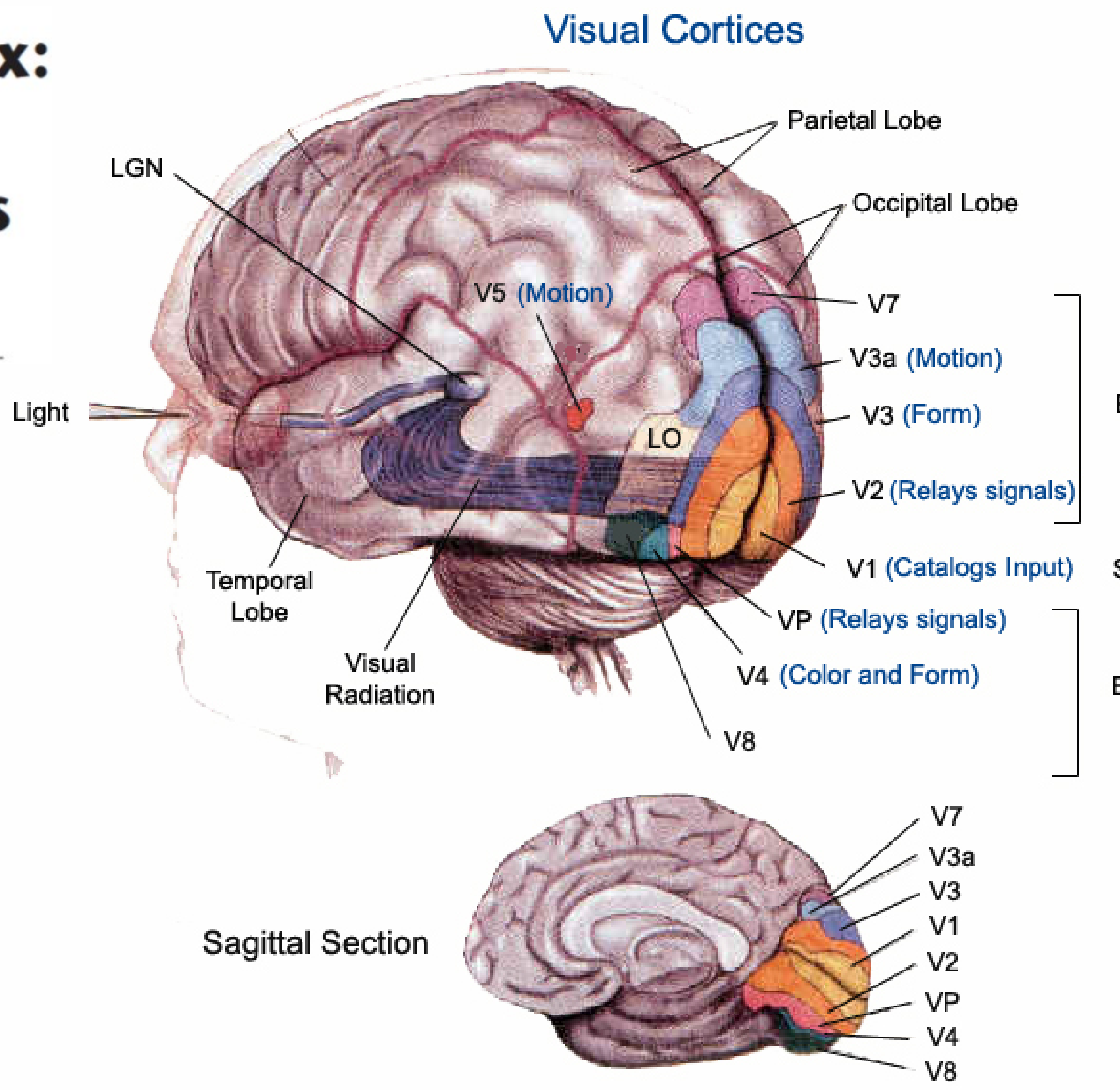


The Biological Inspiration for BICHNN

Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects

Rajesh P. N. Rao[1] and Dana H. Ballard[2]





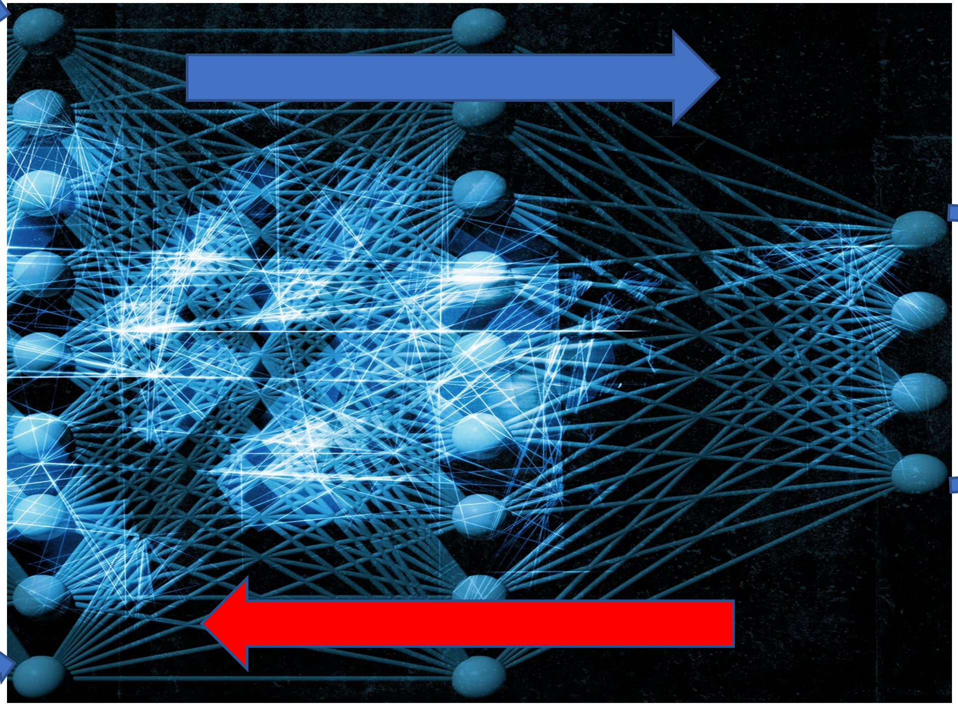
Bidirectional **I**nterleaved **C**omplementary **H**ierarchical **N**eural **N**ets

- **Sensory perception is a dynamic, interactive process, NOT static**
- **Signals from the sensor are hierarchically processed into abstractions**
- **Abstractions are processed in the opposite direction into sensory output**
- **Close your eyes, picture a ‘Fire Truck’. Your visual cortex works in reverse!**
- **These Bidirectional Interleaved Complementary Networks interact**
- **The two networks train each other to do their complementarity tasks**
- **Basically like the generator and discriminator of a GAN, only interleaved**
- **Signals can be bounced between sensor and abstract, like dreaming**
- **What we expect to sense actually influences what we really sense**

Bidirectional **I**nterleaved **C**omplementary **H**ierarchical **N**eural **N**ets

Sensory
Input

Processing Input
Sensory -> Abstract



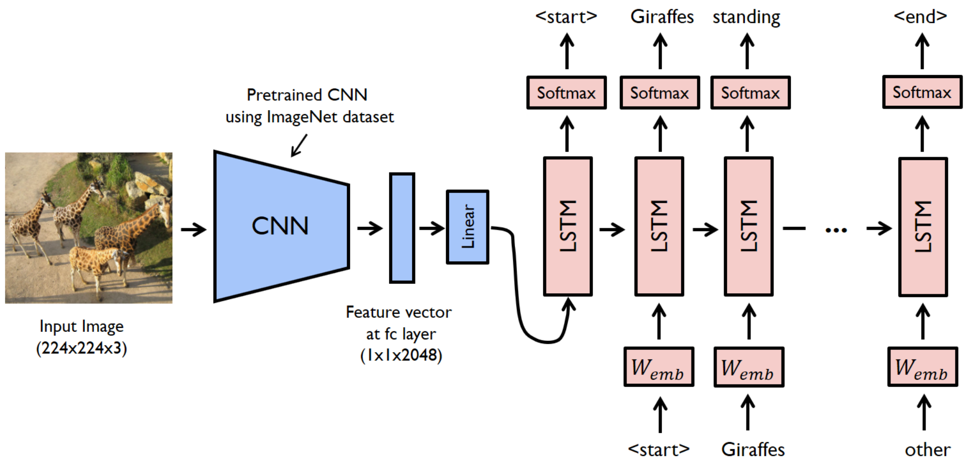
Predicting Input
Abstract -> Sensory

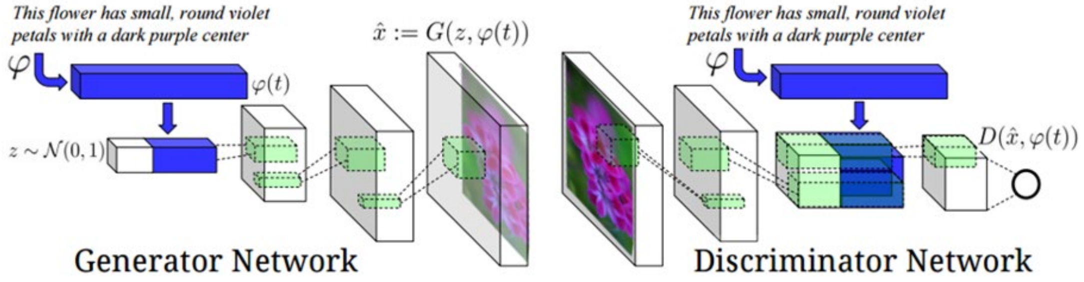
Abstract
Encoding

BICHNN – A Useful New Tool For AI

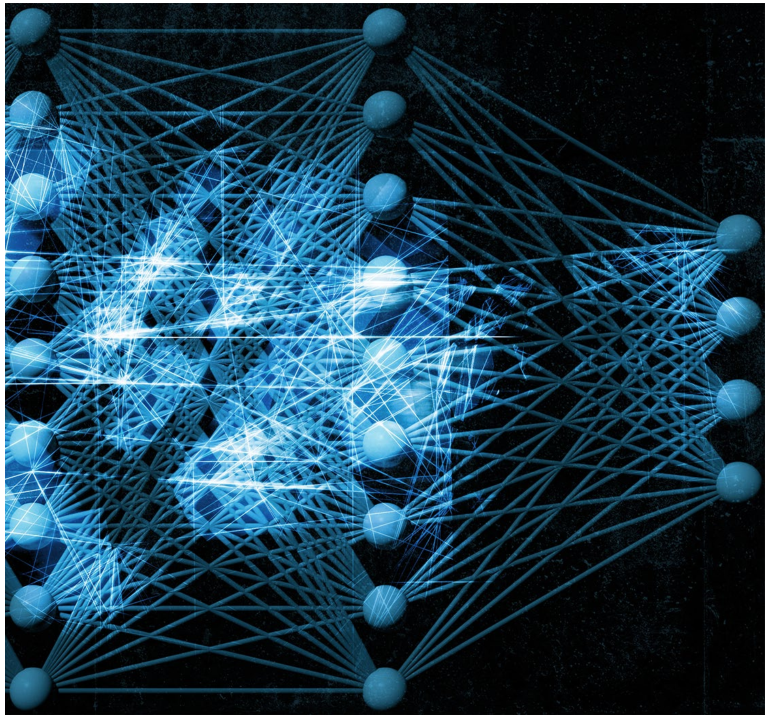
- Can replace CNN, RNN, and make them self-training
- Replaces GANs and Autoencoders, is more accurate, and powerful
- More powerful and easier to train for sensory applications as well
- Network architecture that can perform useful operations and tasks
- Learns how to perform these tasks without explicit instructions
- Learn by doing, on-the fly, from practice and experience
- Can be combined into multi-modal sensory systems to learn associatively
- Learn to do a wide variety of tasks that humans can do (in time)
- Generally applicable to speech, vision, sensory, and control

CNN + RNN + GAN





BICHNN

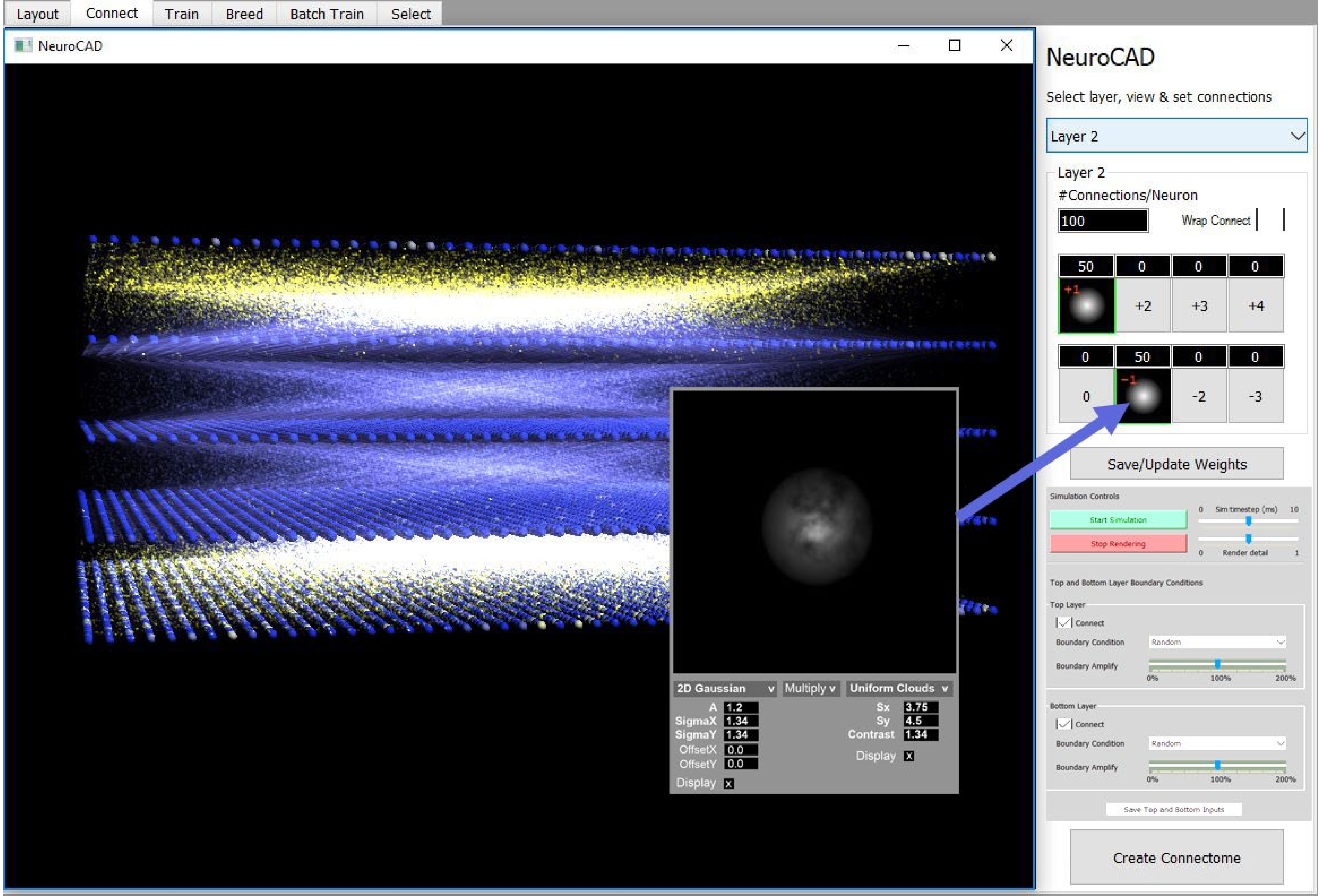


Architecting Spiking Neural Nets is Difficult

- Moderate sized spiking NN: 1 million spiking neurons
- 1 Billion connections & synapses
- 3D geometry is important because signals travel
- Time-dependent circuits, complex relationships
- NO design methodologies, intuition how to connect
- Like throwing 1 billion strands of spaghetti at a wall
- Never going to come up with functional architectures
- Especially not ones that can train and learn
- We need new design tools, new methodologies

NeuroCAD

- **Design software for architecting and testing Spiking Neural Networks**
- **NeuroCAD - UI workflow for SNN design using Genetic Algorithms**
  - **Layout – Lay out layers of neurons and position them**
  - **Connection – Connect the layers of neurons stochastically**
  - **Testing – Run simulations of the SNN in your test harness**
  - **Selection – Select the best performing versions of your network**
  - **Breeding – Cross-breed and mutate the best performing nets**
  - **Iterate – Run testing on the new batch till converged to solution**
- **Build more advanced AI than has ever been possible**

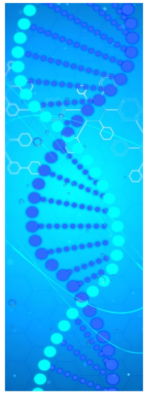

Layout
Connect
Train
Breed
Batch Train
Select
NeuroCAD
NeuroCAD
Select layer, view & set connections
Layer 2
Layer 2
#Connections/Neuron
100
Wrap Connect
50
0
0
0
+1
+2
+3
+4
0
50
0
0
0
-1
-2
-3
Save/Update Weights
Simulation Controls
Start Simulation
Stop Rendering
Sim timestep (ms)
Render detail
Top and Bottom Layer Boundary Conditions
Top Layer
Connect
Boundary Condition
Random
Boundary Amplify
0%
100%
200%
Bottom Layer
Connect
Boundary Condition
Random
Boundary Amplify
0%
100%
200%
Save Top and Bottom Inputs
Create Connectome
2D Gaussian
Multiply
Uniform Clouds
A 1.2
SigmaX 1.34
SigmaY 1.34
OffsetX 0.0
OffsetY 0.0
Display
Sx 3.75
Sy 4.5
Contrast 1.34
Display

NeuroCAD Genome – Connectome Expansion

- **The human brain has 100B Neurons, 100T Connections**
- **All of this grows from the blueprint of only 8000 genes**
- **8000 genes -> 100 trillion neural connection connectome**
- **This is one heck of a decompression algorithm!**
- **You need genes to do genetic algorithms, to breed and mutate**
- **NeuroCAD uses a few hundred parameters as genome**
- **These are expanded into 2D procedural maps and mixed in tree**
- **Output is a 2D probability map for connection of LayerN -> LayerM**
- **Genome Parameters -> 2D Procedural Maps -> Connectome**

Defining the NeuroCAD Connectome Algorithmically

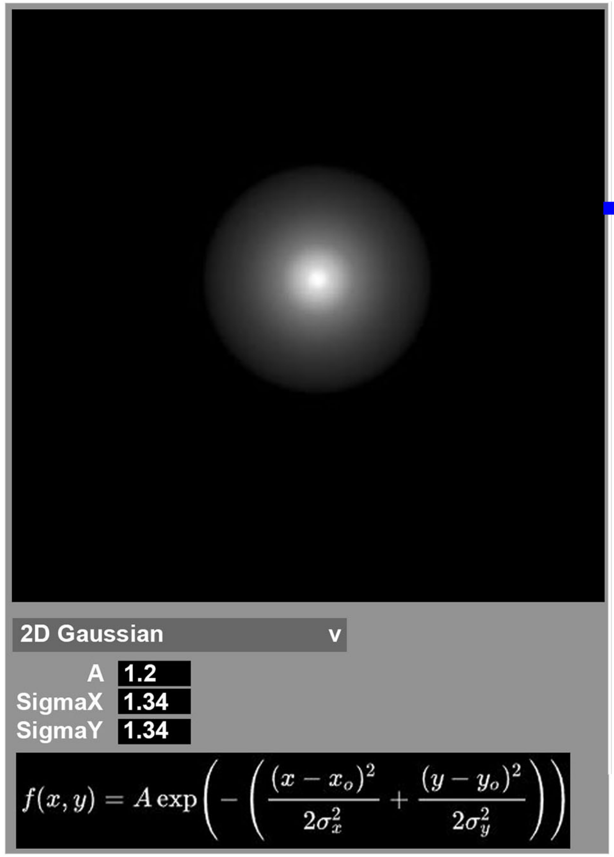
Parameters (Genome) -> 2D Algorithms -> 2D Probability Maps -> Connectome



Genome (Parameters)

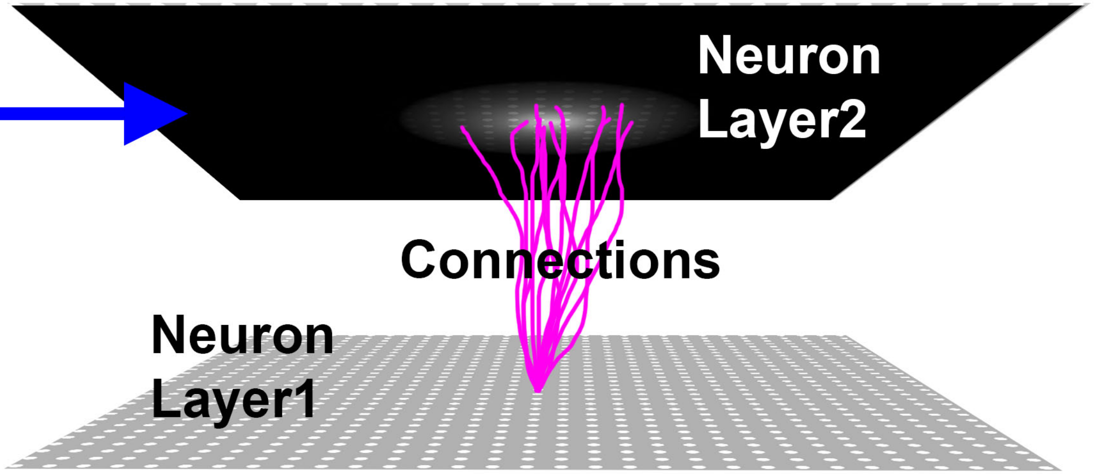
A
SigmaX
SigmaY

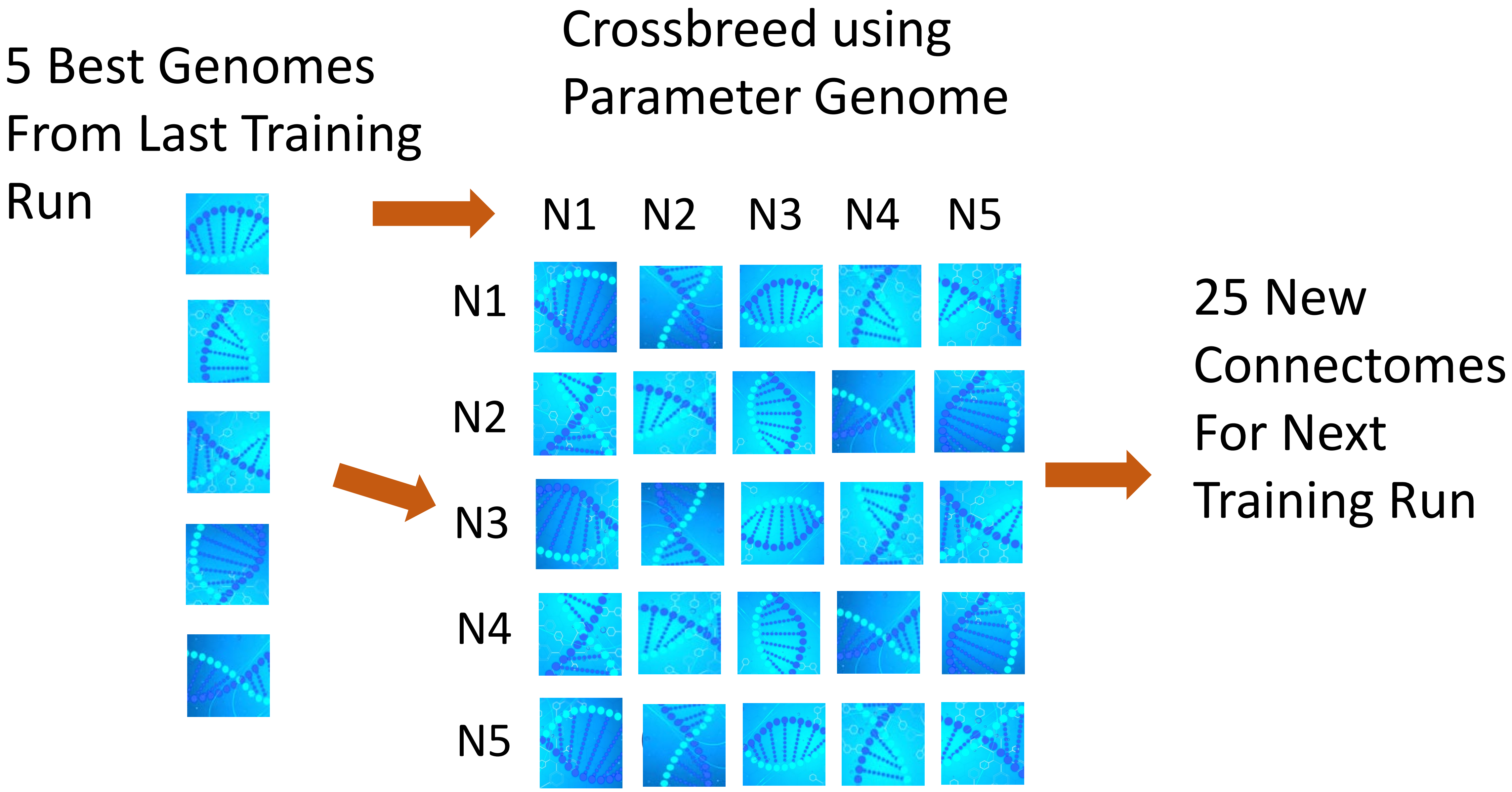
Mapping 2D Maps



Connectome

Map Shows Connection Probability



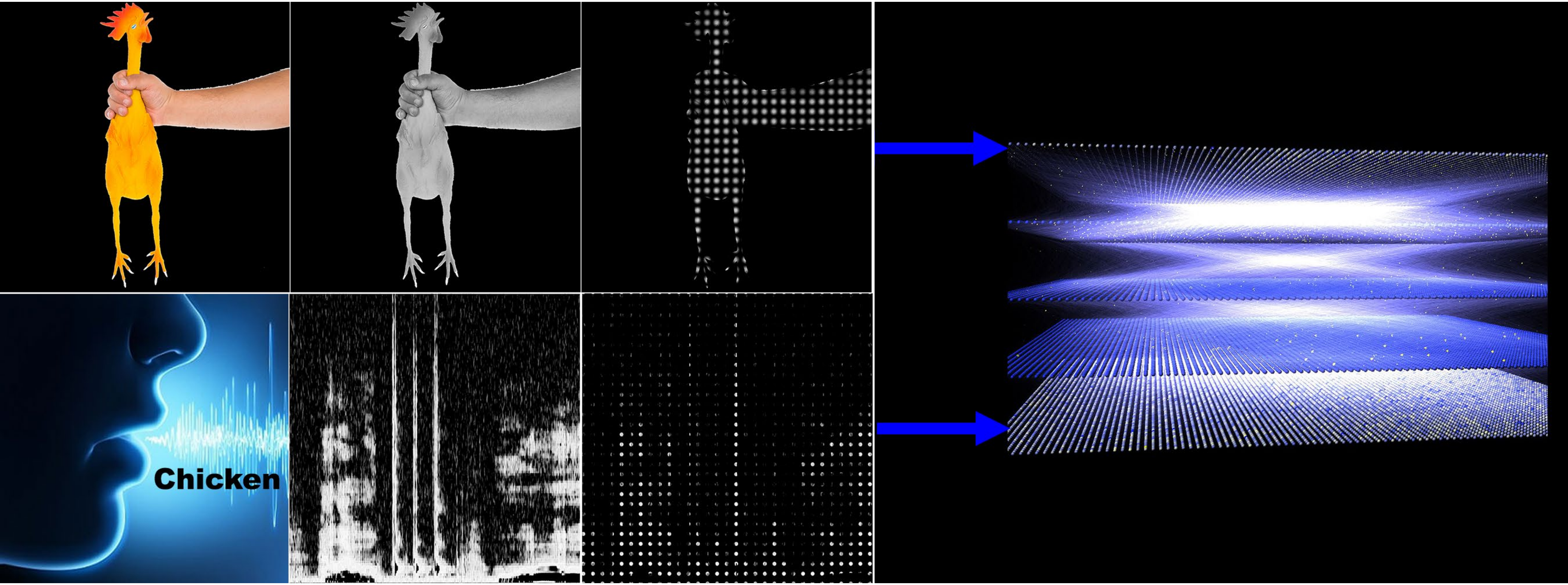


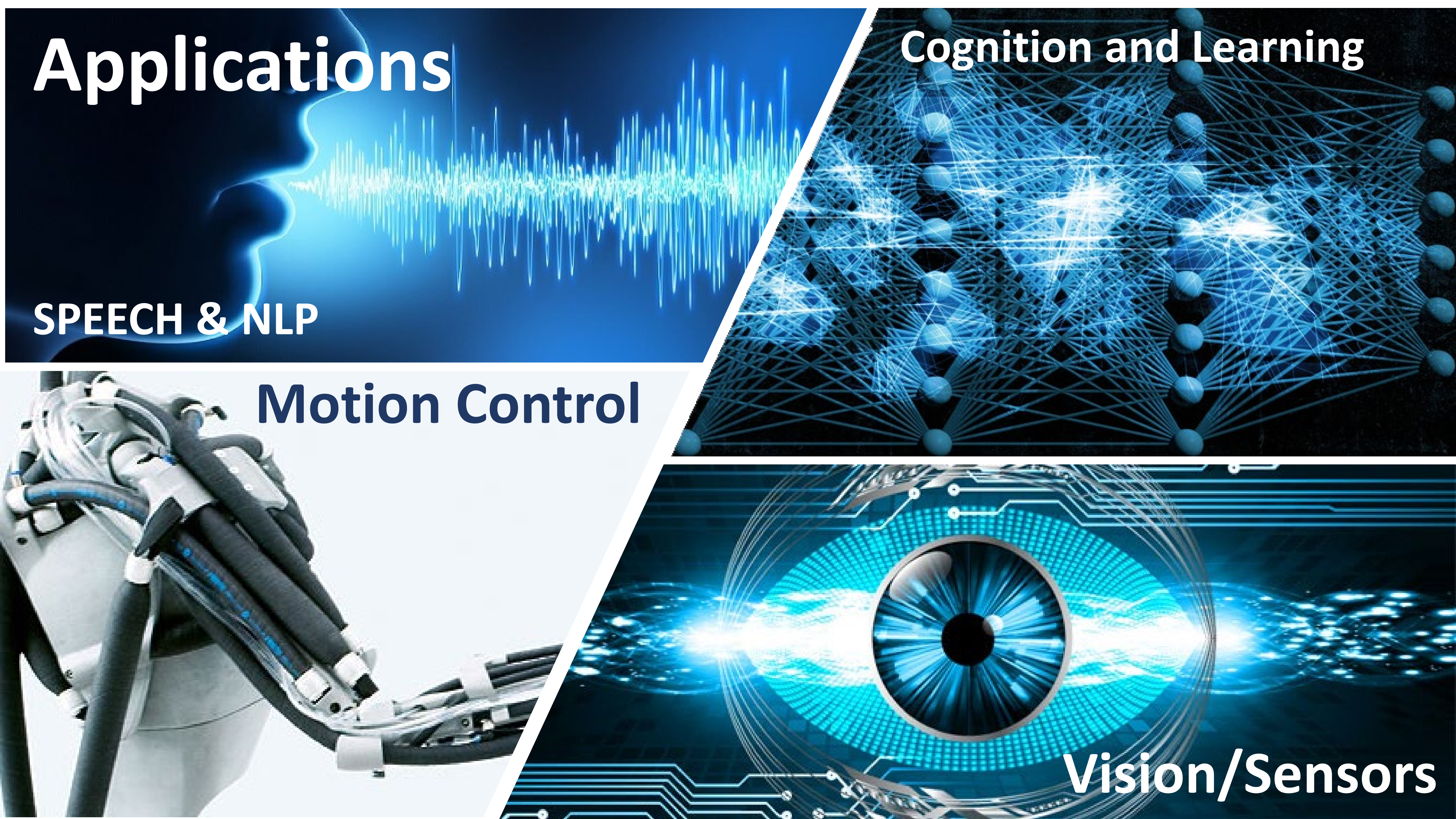
5 Best Genomes From Last Training Run
Crossbreed using Parameter Genome
N1 N2 N3 N4 N5
N1
N2
N3
N4
N5
25 New Connectomes For Next Training Run

NeuroCAD

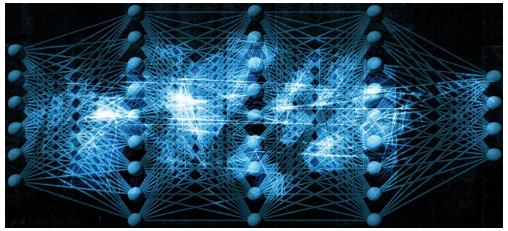
NeuroCAD is a software tool with a GUI for designing Spiking Neural Networks. It allows the user to lay out the layers of spiking neurons, connect them up algorithmically, crossbreed and mutate them to generate a population of similar neural nets, then run simulations on them, train them, cull the underperformers, and then crossbreed the top performing designs and continue the genetic algorithms till a design emerges that meets the performance criteria set by the designer.

Associative Learning with BICHNN

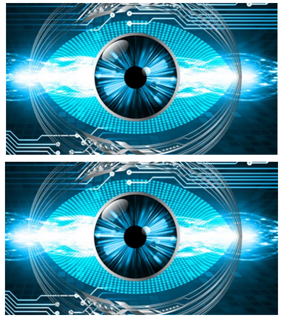


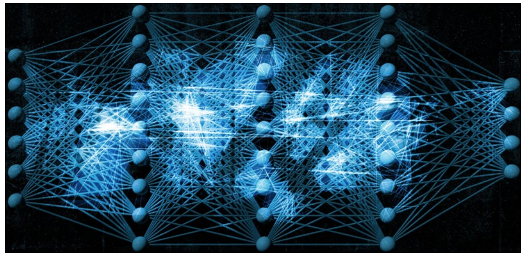

Applications
SPEECH & NLP
Cognition and Learning
Motion Control
Vision/Sensors

Building a Humanoid Robot AI with BICHNN

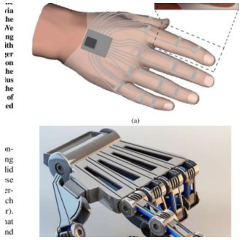


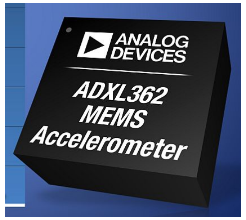
BICHNN Speech

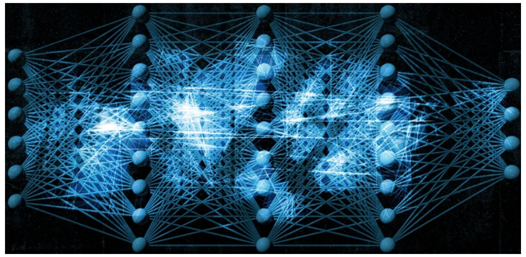




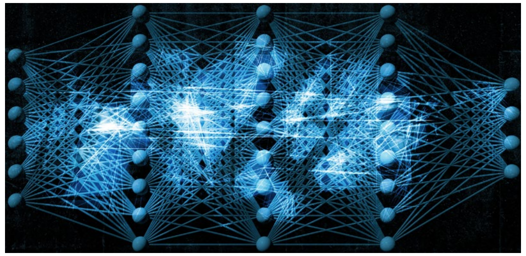
BICHNN Vision



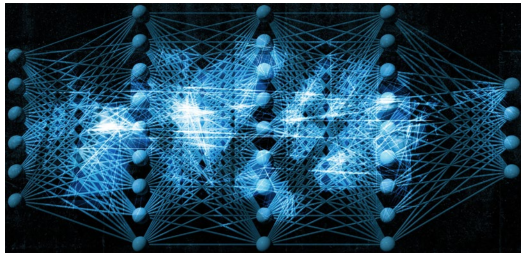




BICHNN Sensory

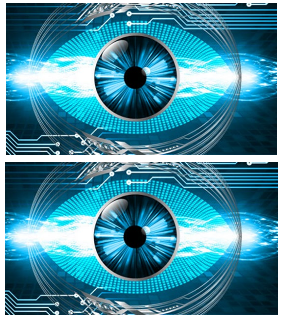


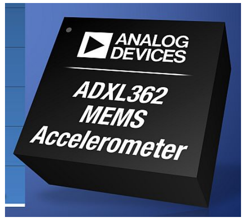
BICHNN Facial Animation

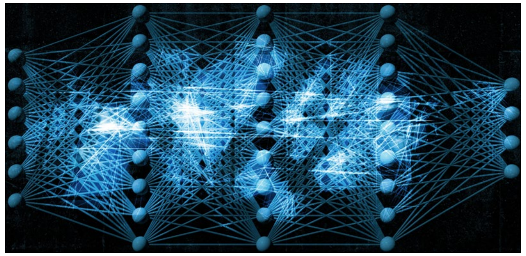


BICHNN Robot Controller

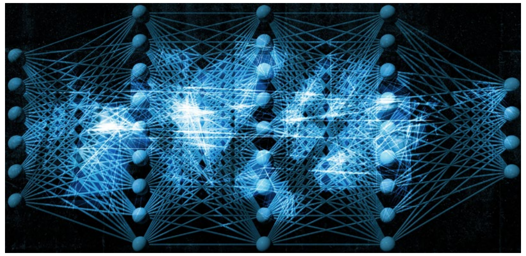
Building an Auto (or Drone) AI with BICHNN



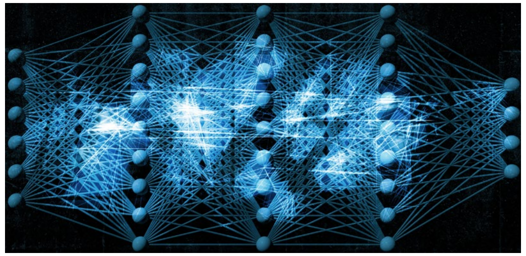




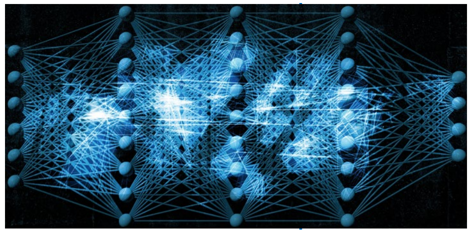
BICHNN Speech



BICHNN Vision



BICHNN Sensory



BICHNN Auto-Driving Controller

BICHNN Drone Controller