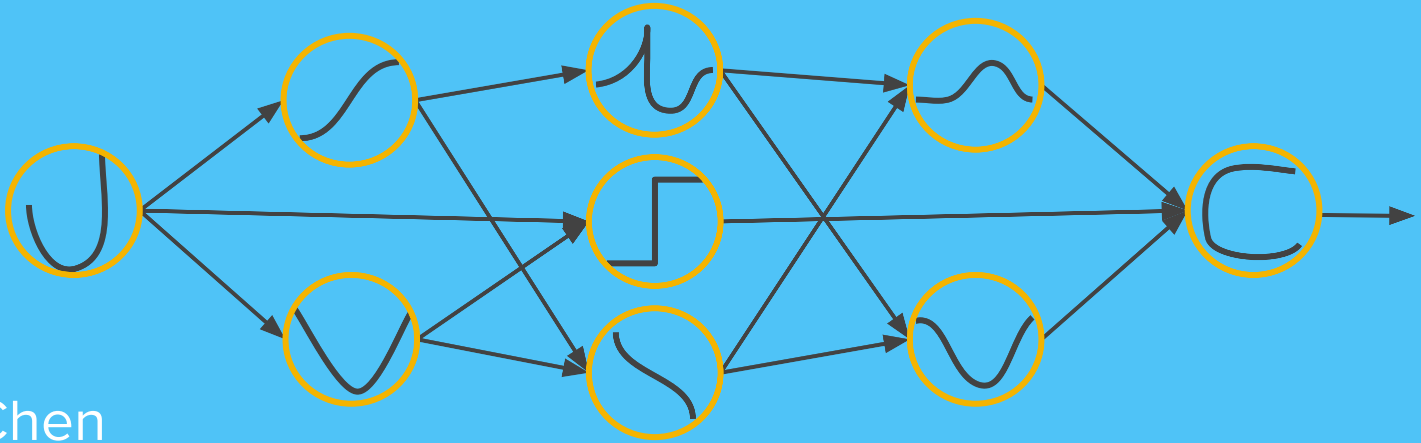
Revisiting NEAT Part II:

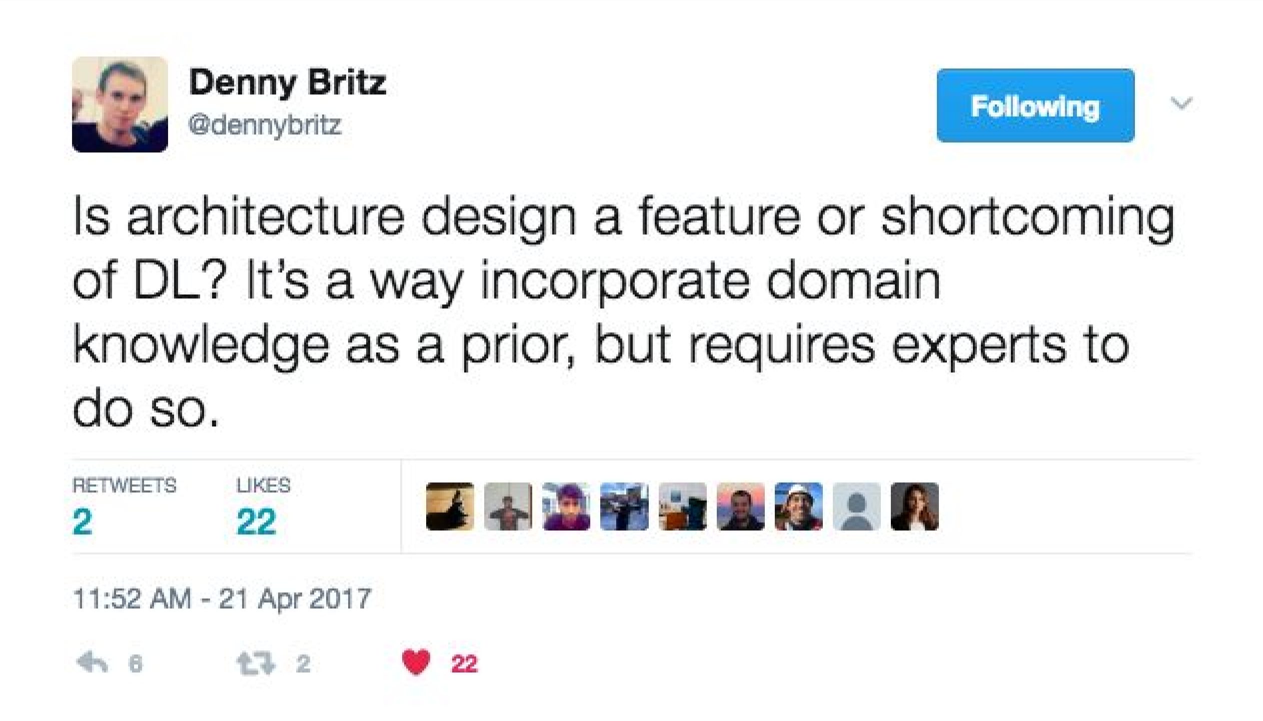
Compositional Pattern Producing Networks



Justin Chen

Machine **I**ntelligence **C**ommunity

Motivation

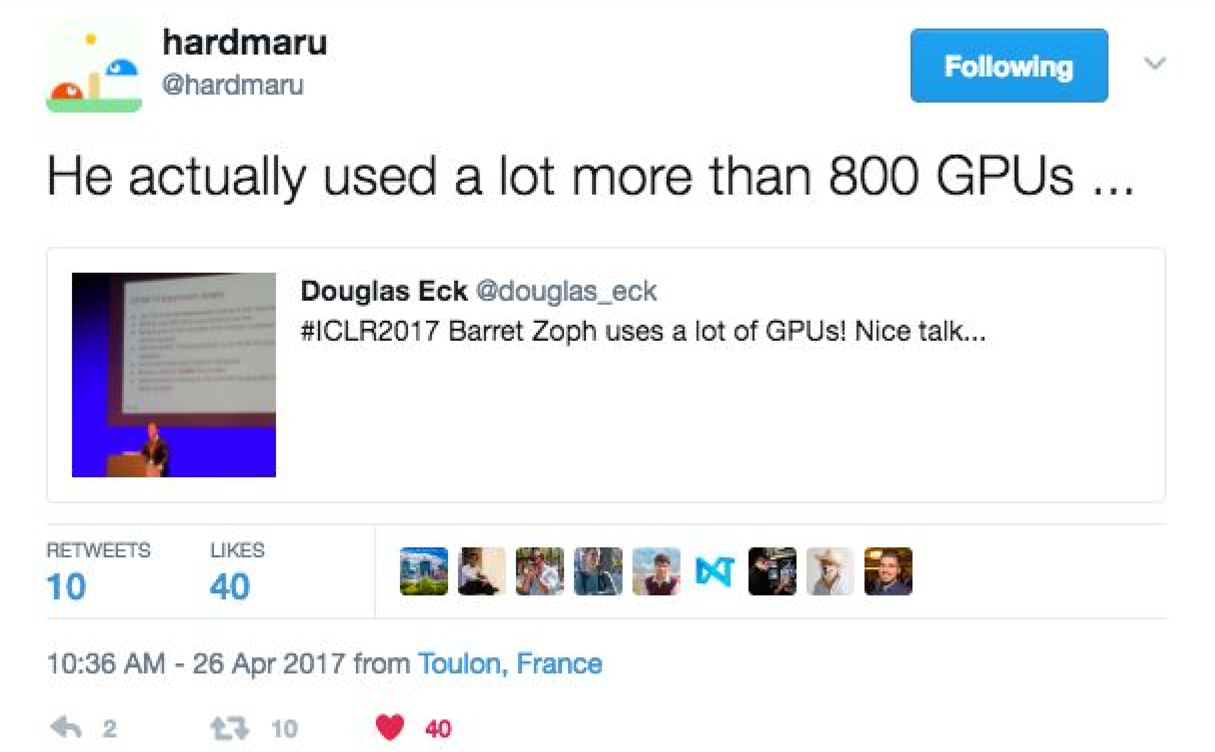

Denny Britz
@dennybritz

Is architecture design a feature or shortcoming of DL? It's a way incorporate domain knowledge as a prior, but requires experts to do so.

RETWEETS 2 LIKES 22

11:52 AM - 21 Apr 2017

Motivation



hardmaru
@hardmaru

He actually used a lot more than 800 GPUs ...

Douglas Eck @douglas_eck
#ICLR2017 Barret Zoph uses a lot of GPUs! Nice talk...

RETWEETS 10
LIKES 40

10:36 AM - 26 Apr 2017 from Toulon, France

2 10 40

Recap of NEAT

Neuroevolution of Augmenting Topologies

- Minimal Topology
- Incremental Growth
- Speciation

Recap of NEAT

https://twitter.com/dennybritz/status/855494508888203264

https://twitter.com/hardmaru/status/854524933187158016

Motivation

- Simultaneously discover and train neural networks
- Justify architectural components
- Novel architectures

Start from minimal topology

- reduce size of network and architectural search space

Incremental Growth

- Justify architectural components

Speciation

- Protect innovation

Compositional Pattern Producing Networks

A Novel Abstraction of Development

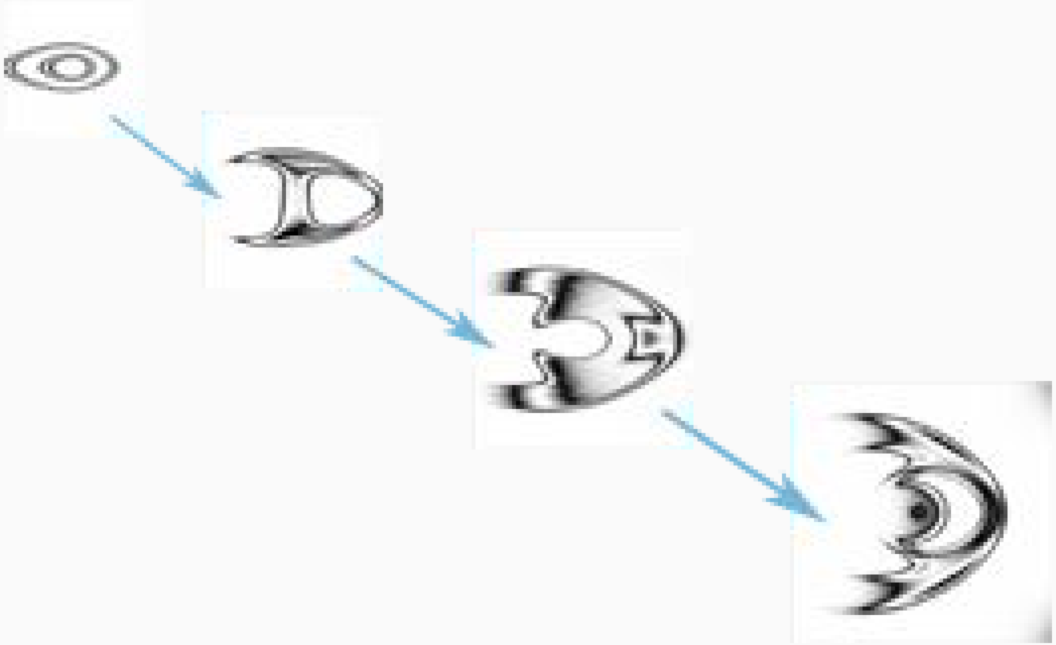
- Developmental Encoding
- Local Interaction
- Temporal Unfolding

Key Ideas and Properties

1. **Compact representations** for complex structures
2. Develop **structural motifs (regularities)**
3. Discover complex phenotypes via progressive **complexification** of genotype
4. Simulating **local interaction** and **temporal unfolding** is **unnecessary**
    a. Local Interaction - neighboring phenotypes are correlated
    b. Temporal Unfolding - state of a phenotype at one moment in time affects future states in the same area

Background

1. **Artificial Development**
2. **Patterns of Development**
   a. Repetitions
   b. Repetition with Variation
   c. Symmetry
   d. Imperfect Symmetry
   e. Elaborated Regularity
   f. Preservation of Regularity
3. **Complexification**



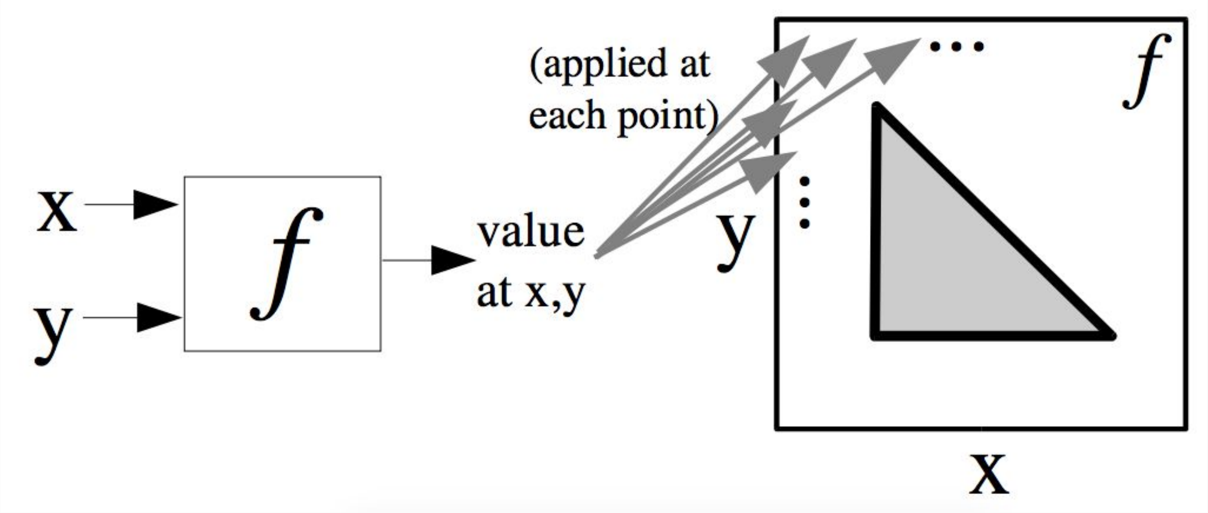
http://picbreeder.org/images/gui.png

Patterns Without Local Interaction

- Artificial Developmental Encodings
- Patterns of Development
- Complexification

Eliminating Temporal Unfolding

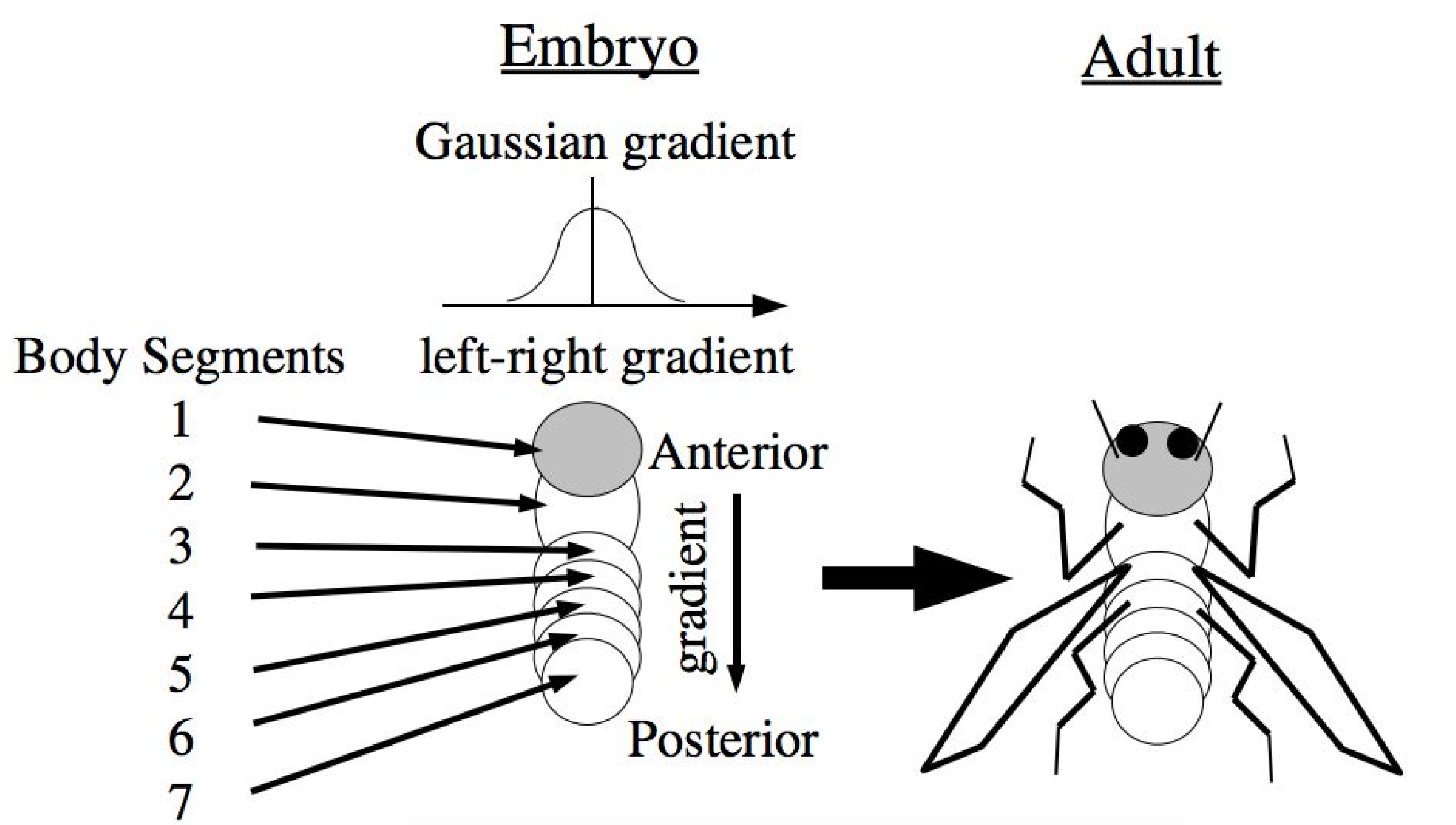
1. Describe phenotype **without explicitly** simulating temporal unfolding
2. Instead abstract development as a **composition of functions**
3. **Phenotype** now expressed as a **distribution of points**



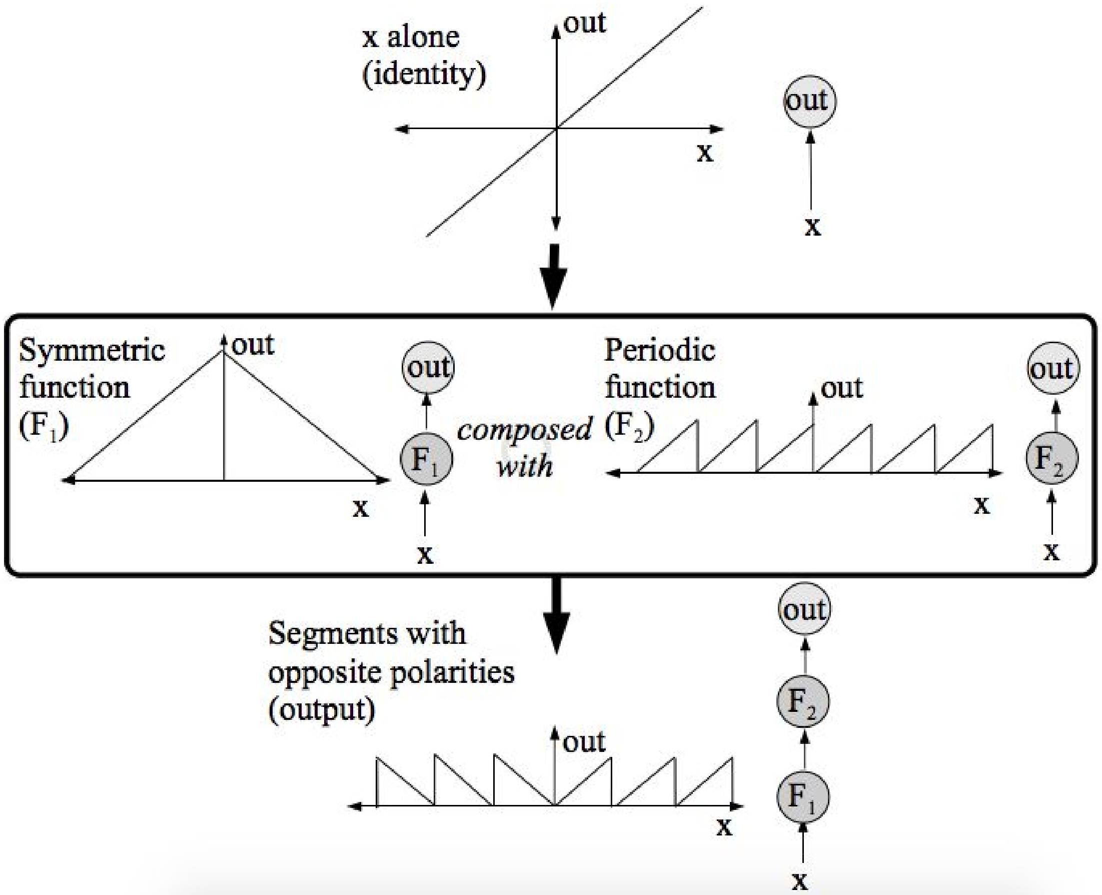
Functions and Regularities Without Local Interaction

- To eliminate temporal unfolding, must **mimic local interaction** to remain a **valid abstraction** of development
- **Coordinate frame** is established in early stages of natural embryo development
- **New coordinate frames** for defining **subregions**, **subfields**, build off initial axis
- **Assumes** boundaries between body segments becomes the region of subfield

Functions and Regularities Without Local Interaction

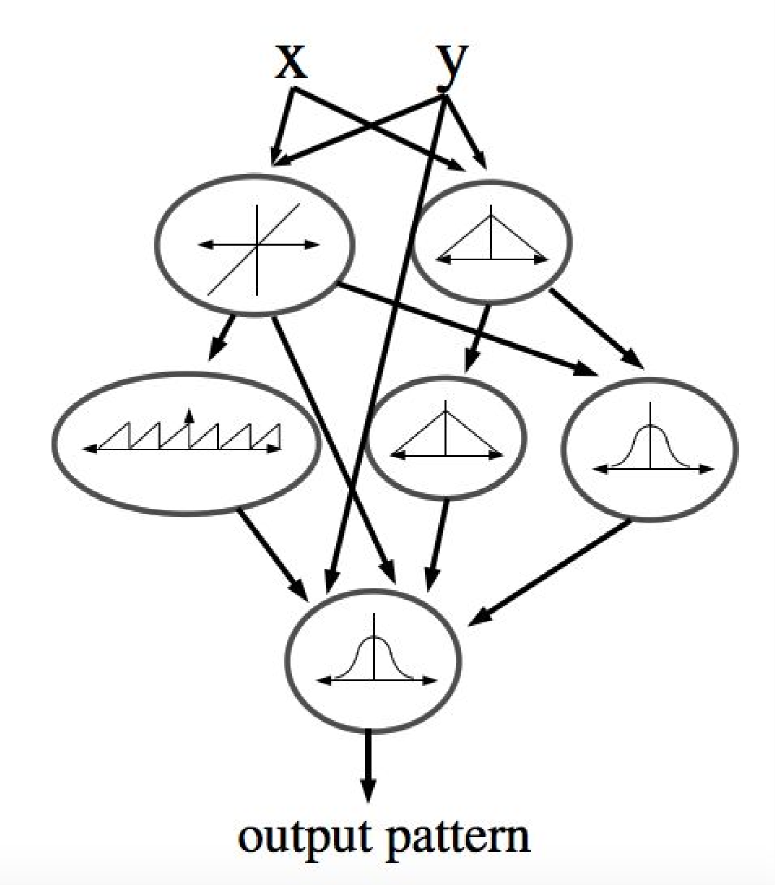


Functions and Regularities Without Local Interaction



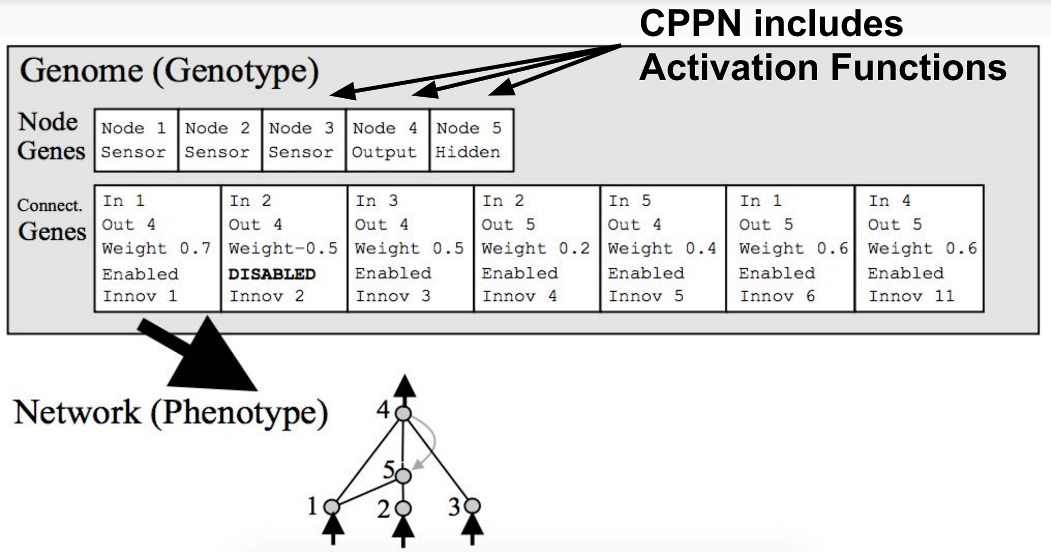
Functions and Regularities Without Local Interaction

- Describe process as a computation graph. *Look familiar?*
- Weighted connections leaving functions
- Multiple incoming connections summed
- X and Y are **initial coordinate axes**



CPPN-NEAT

- **Node genes** contain additional field for activation function
- Node assigned **random activation** from pool
- **Compatibility function** contains extra term to account for activation function



CPPN includes Activation Functions

Genome (Genotype)

Node Genes				
Node 1 Sensor	Node 2 Sensor	Node 3 Sensor	Node 4 Output	Node 5 Hidden

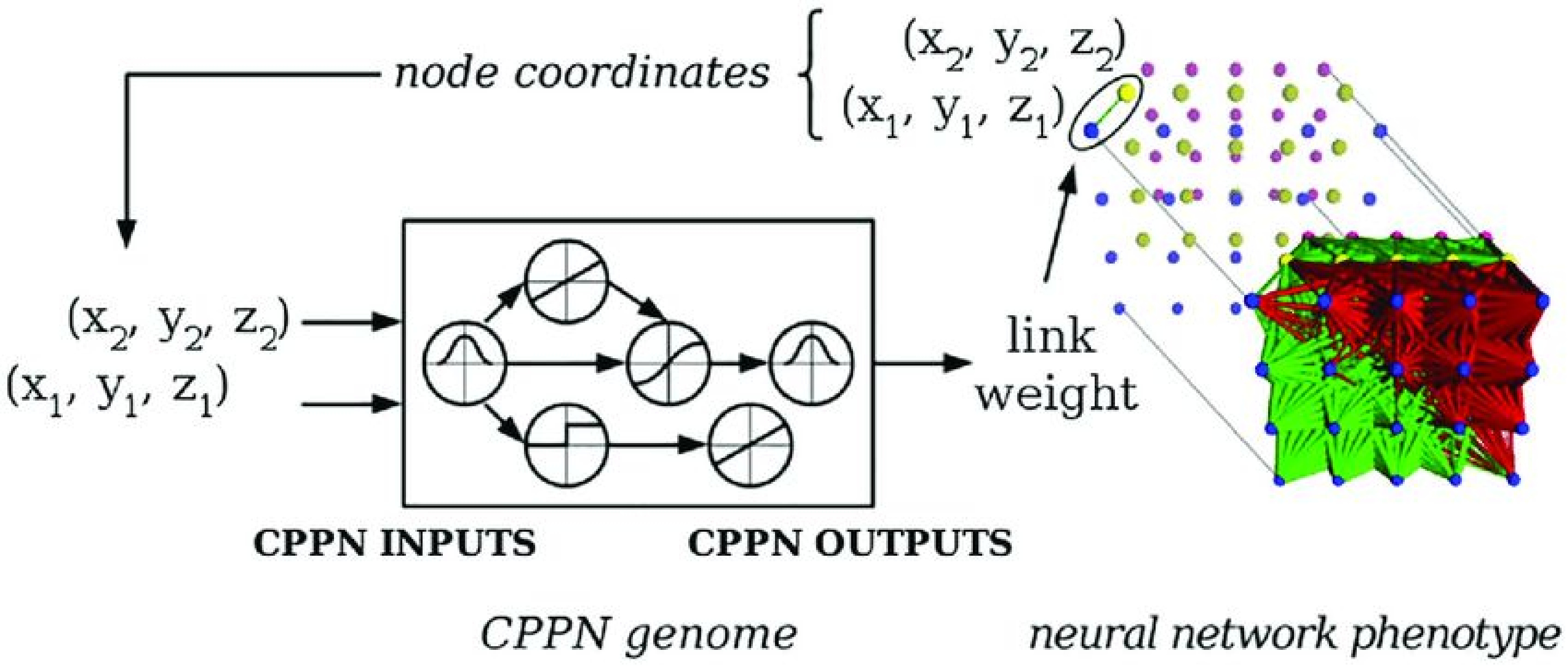
Connect. Genes						
In 1 Out 4 Weight 0.7 Enabled Innov 1	In 2 Out 4 Weight-0.5 **DISABLED** Innov 2	In 3 Out 4 Weight 0.5 Enabled Innov 3	In 2 Out 5 Weight 0.2 Enabled Innov 4	In 5 Out 4 Weight 0.4 Enabled Innov 5	In 1 Out 5 Weight 0.6 Enabled Innov 6	In 4 Out 5 Weight 0.6 Enabled Innov 11

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W} + c_4 \cdot \overline{A}$$

CPPN Computational Complexity

- **Must fully activate** for all N coordinates, so $O(N)$
  - e.g. a 2x2 grid, $O(N^2)$
- Can produce phenotype at **any resolution** by increasing number of coordinates
- CPPN is an **infinite resolution** mathematical structure
- Computational complexity minimized for a task by activating at **minimally necessary resolution**
- CPPN activated independently for every coordinate in phenotype allowing complete computation of **every coordinate to be parallelizable**

CPPN-NEAT Neural Network Phenotype



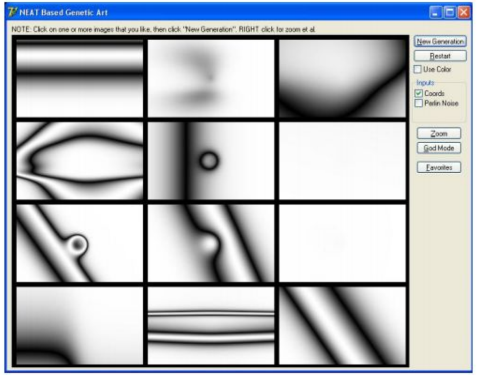
Experiments

- Picbreeder

Goals of experiments

1. CPPNs abstract natural development without depending on local interaction and unfolding over time
2. The way CPPNs solve tasks parallels natural evolution and development
3. Ability to establish and exploit regularities in general
4. **Establish relationship between CPPNs and development**

CPPN-NEAT-based Interactive Pattern Evolution



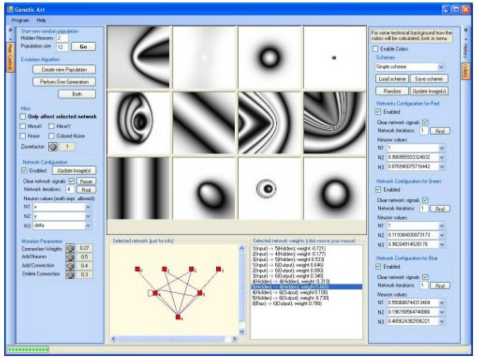
NOTE: Click on one or more images that you like, then click "New Generation". RIGHT click for zoom et al.

□ Use Color

Inputs

☑ Coords

□ Perlin Noise



Start new random population

Hidden Neurons: 2

Population size: 12

Evolution Algorithm

Misc

□ Only affect selected network

□ MirrorX □ MirrorY

□ Noise □ Colored Noise

Zoomfactor: 1

Network Configuration

☑ Enabled

Clear network signals: ☑

Network iterations: 4

Neuron values (math expr. allowed):

N1 x

N2 y

N3 delta

Mutation Parameters

Connection Weights 0.27

Add Neuron 0.5

Add Connection 0.4

Delete Connection 0.3

Selected network (just for info)

Selected network weights (click+move your mouse)

1(Input) -> 5(Hidden); weight:-0.721
2(Input) -> 4(Hidden); weight:-0.177
3(Input) -> 5(Hidden); weight:0.533
1(Input) -> 6(Output); weight:0.646
2(Input) -> 6(Output); weight:0.000
3(Input) -> 6(Output); weight:0.348
4(Hidden) -> 4(Hidden); weight:-0.319
5(Hidden) -> 4(Hidden); weight:[illegible]
4(Hidden) -> 6(Output); weight:0.100
5(Hidden) -> 6(Output); weight:-0.730
0(Bias) -> 6(Output); weight:0.780

For some technical background how the colors will be calculated, look in menu

□ Enable Colors

Schemes

Simple scheme

Networks Configuration for Red

☑ Enabled

Clear network signals: ☑

Network iterations: 1

Neuron values:

N1 1

N2 0.9068955933324832

N3 0.876940075716442

Network Configuration for Green

☑ Enabled

Clear network signals: ☑

Network iterations: 1

Neuron values:

N1 1

N2 0.1193848308873173

N3 0.982049145281178

Network Configuration for Blue

☑ Enabled

Clear network signals: ☑

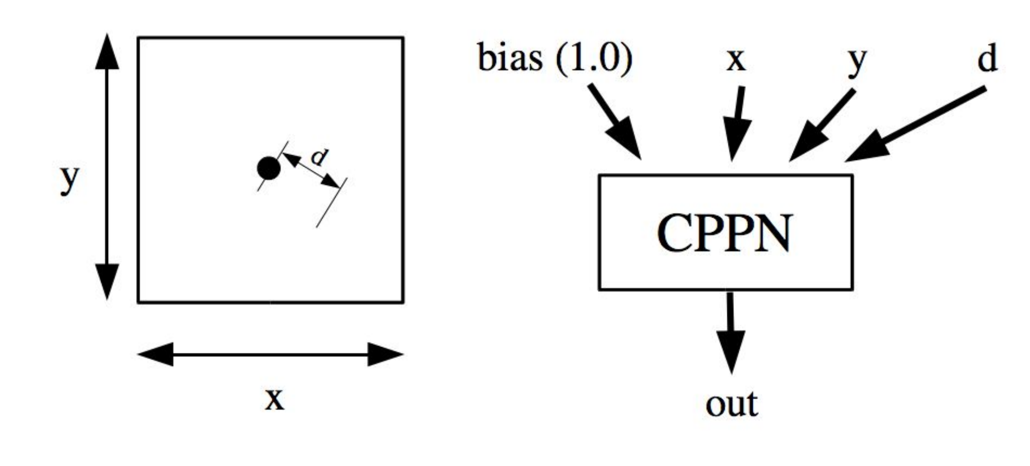
Network iterations: 1

Neuron values:

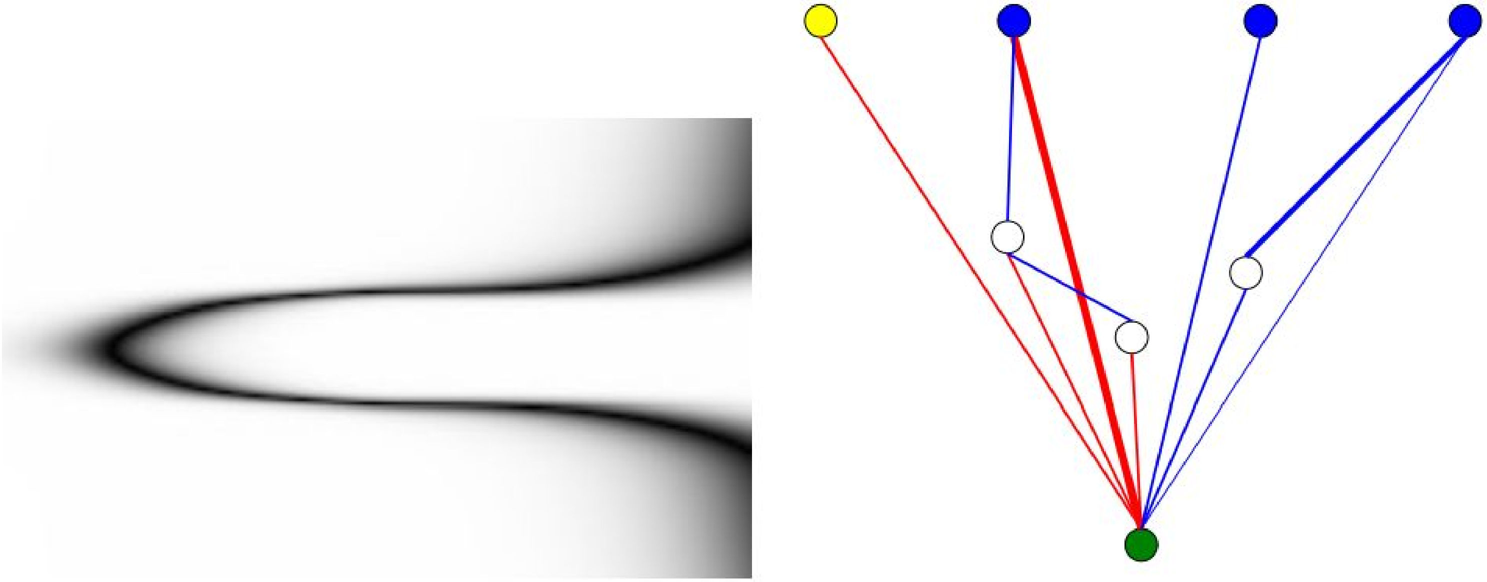
N1 0.550680744313404

N2 0.196150564740088

N3 0.465624382936221



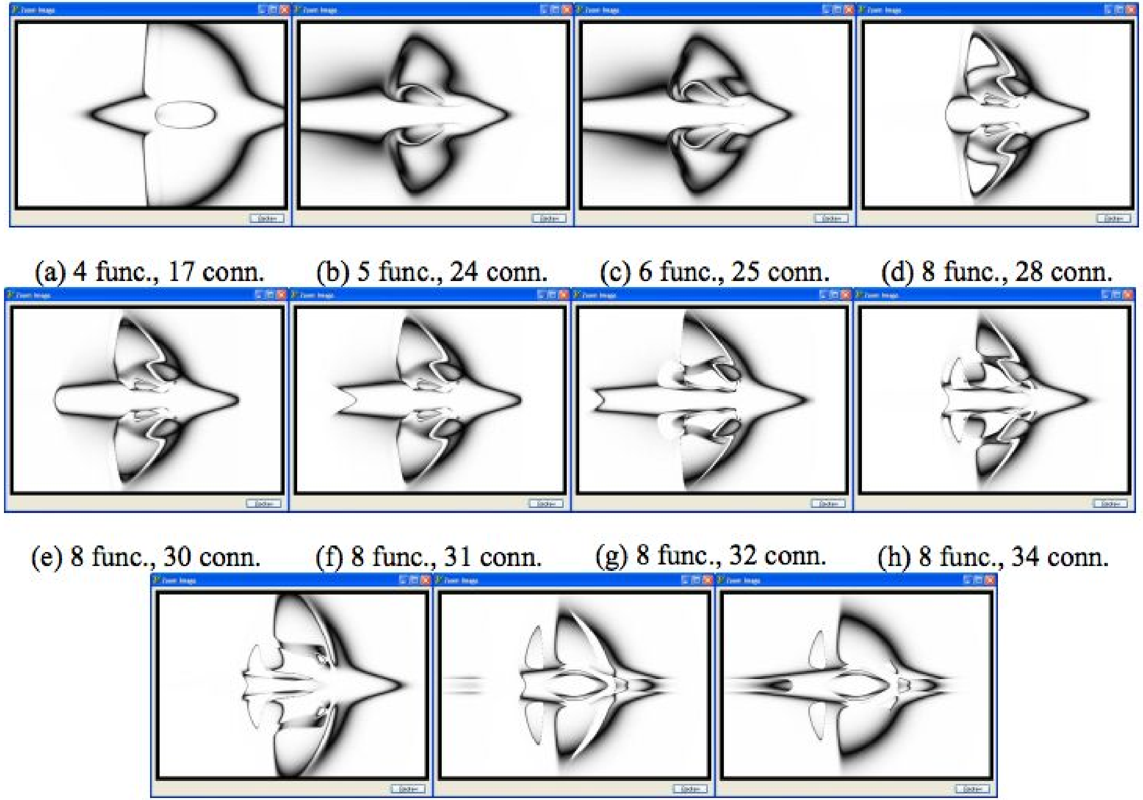
Elaboration on Pure Symmetry



(a) First Bilateral Symmetry

(b) CPPN

Elaboration on Pure Symmetry

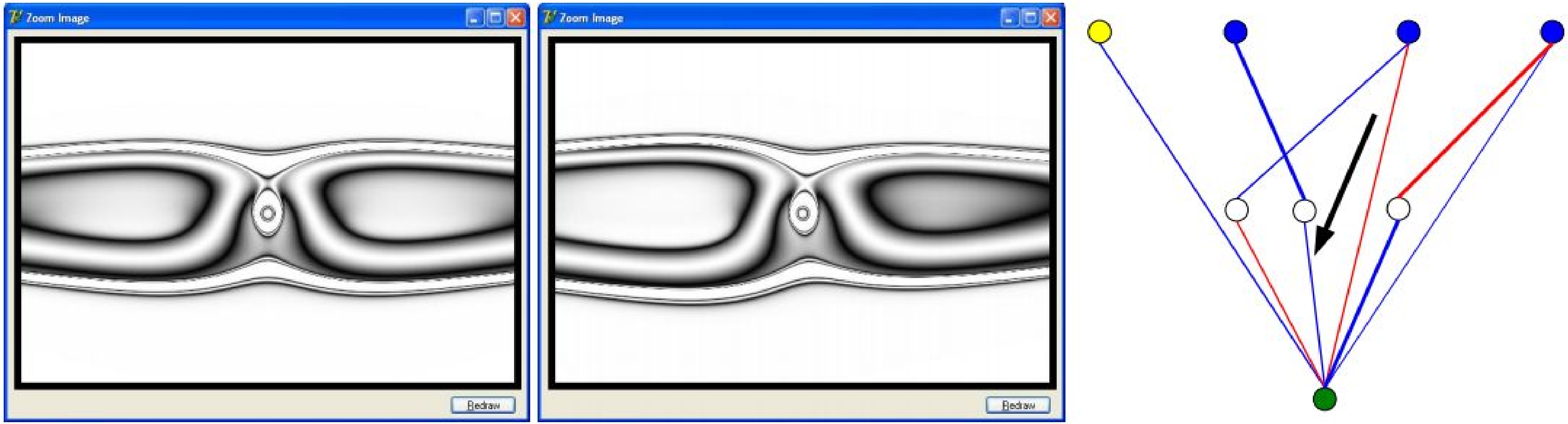


(a) 4 func., 17 conn.
(b) 5 func., 24 conn.
(c) 6 func., 25 conn.
(d) 8 func., 28 conn.

(e) 8 func., 30 conn.
(f) 8 func., 31 conn.
(g) 8 func., 32 conn.
(h) 8 func., 34 conn.

(i) 8 func., 36 conn.
(j) 9 func., 36 conn.
(k) 9 func., 38 conn.

Representing Imperfect Symmetry

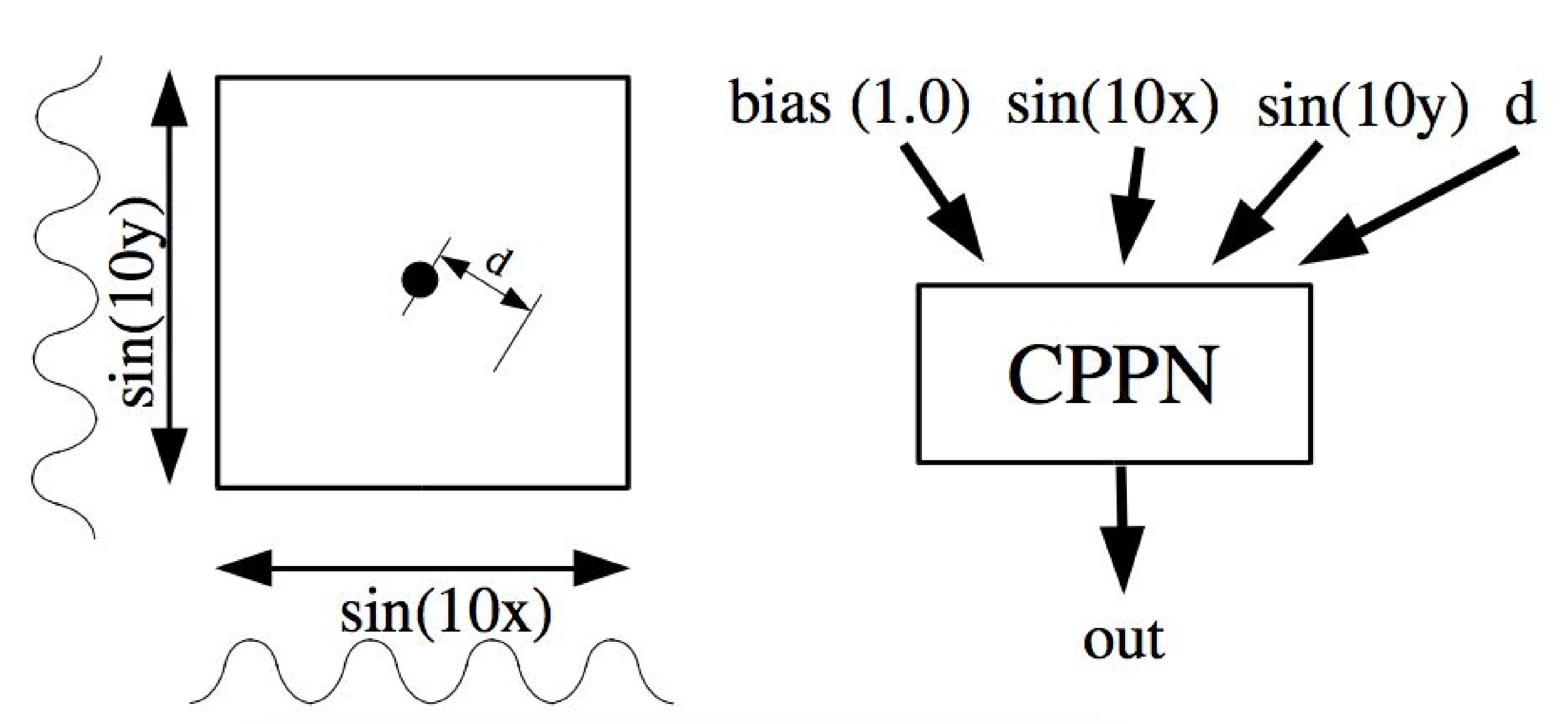


(a) Symmetric Sunglasses

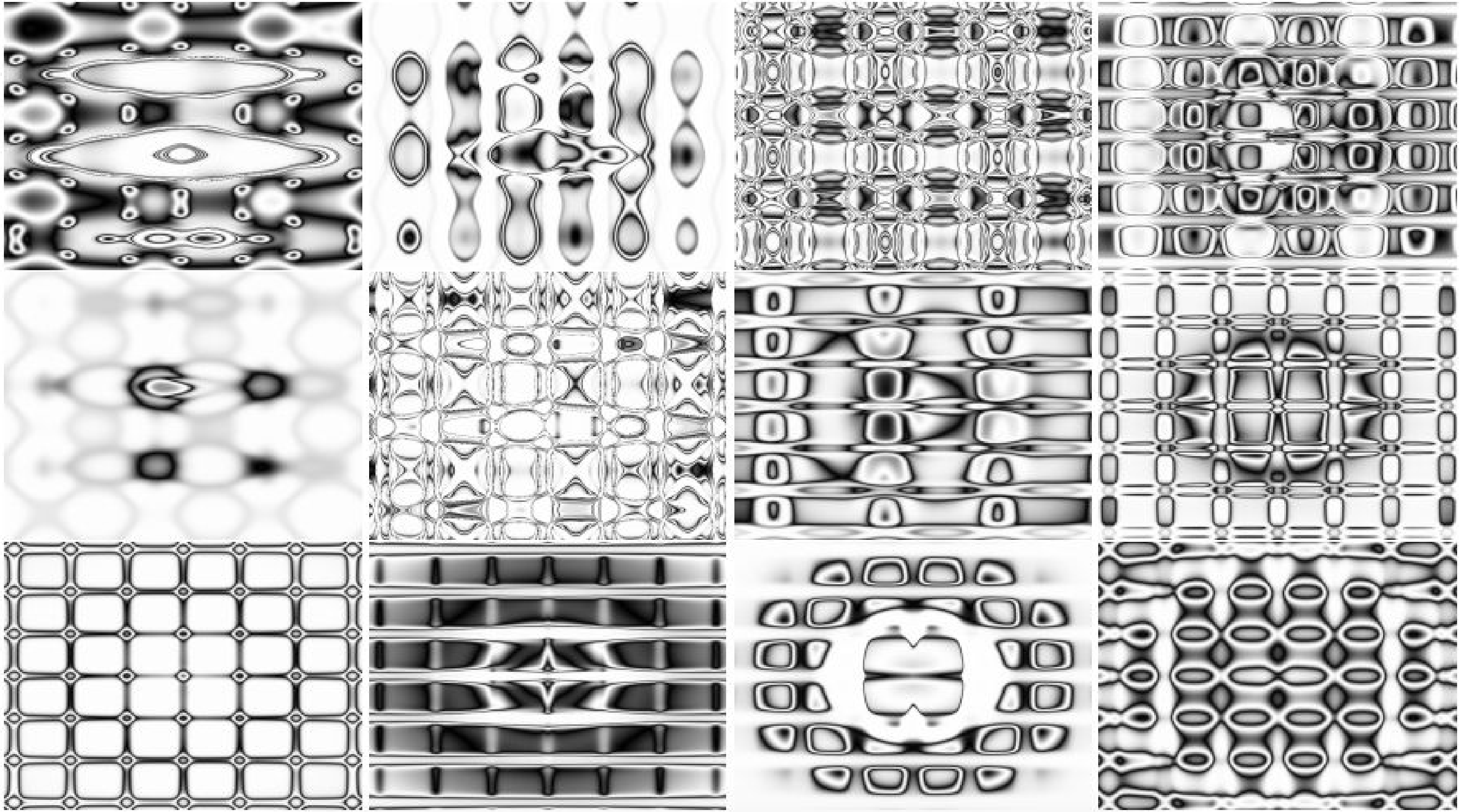
(b) Warped Symmetry

(c) CPPN

Repetition with Variation

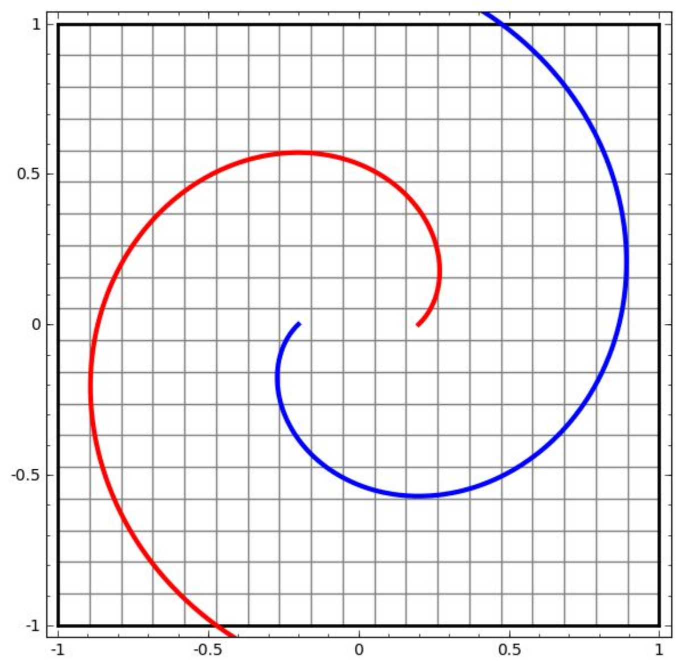


Repetition with Variation



Future work

1. Extending CPPNs as functions of their environments
2. Adaptation via requerying CPPN
3. Contiguous morphology problem and solutions in $\mathbb{R}^N$
4. Applying CPPNs to large-scale neural networks, physical morphologies, and building architectures



http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

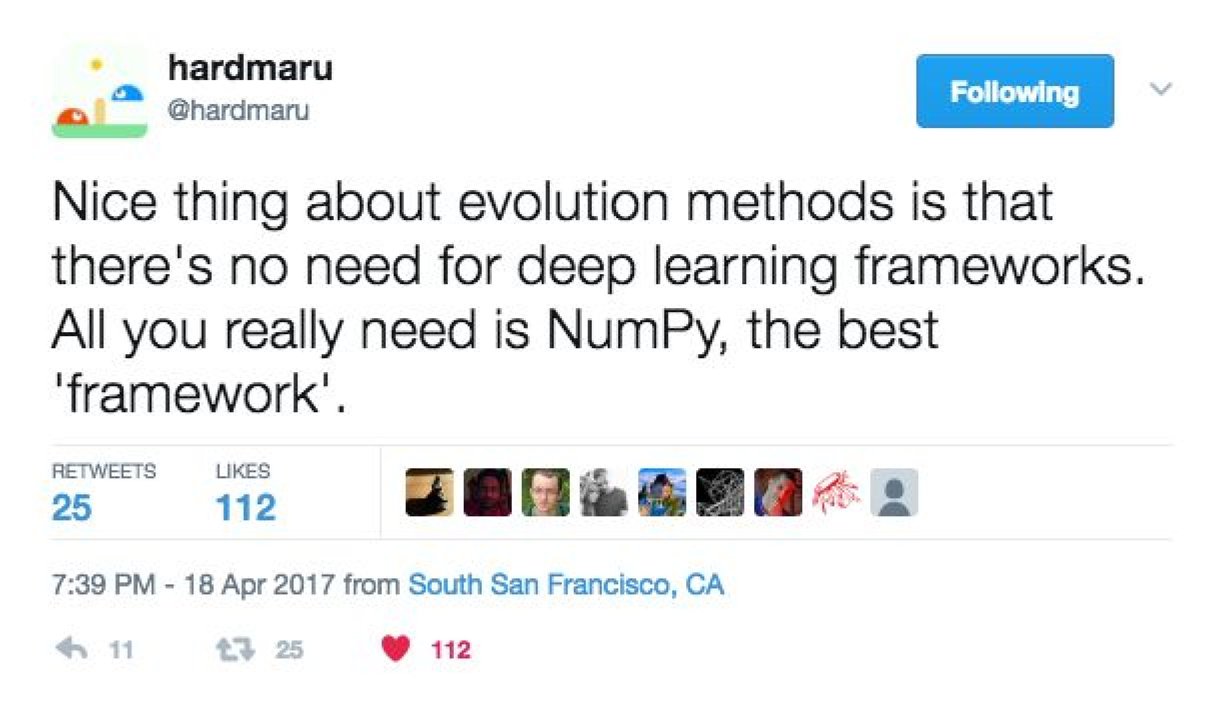
Conclusions

General pattern-generating mechanism

CPPNs exhibit fundamental properties of natural development

1. Symmetry
2. Imperfect symmetry
3. Repetition
4. Repetition with variation
5. Elaboration of existing regularities via complexification
6. Preservation of regularities

Final motivating tweet



hardmaru
@hardmaru

Nice thing about evolution methods is that there's no need for deep learning frameworks. All you really need is NumPy, the best 'framework'.

RETWEETS 25
LIKES 112

7:39 PM - 18 Apr 2017 from South San Francisco, CA

Citations

[0] Stanley, Kenneth O. "Compositional pattern producing networks: A novel abstraction of development." *Genetic programming and evolvable machines* 8.2 (2007): 131-162.

[1] Helms, Lucas, and Jeff Clune. "Improving HybrID: How to best combine indirect and direct encoding in evolutionary algorithms." PloS one 12.3 (2017): e0174635.

[2] http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/