

Michael Bleher

Institute for Mathematics, Heidelberg University

– YTM 2025 –

THE TANGLED WEB THEY WEAVE

EXPLORING NEURAL NETWORKS WITH DIRECTED TOPOLOGY

Artificial Neural Networks

Multi-Layer Perceptron (MLP)

$$f: \mathbb{R}^n \to \mathbb{R}^{k_1} \to \dots \to \mathbb{R}^{k_{L-1}} \to \mathbb{R}^m$$
$$a^{(0)} = x \in X \subset \mathbb{R}^n$$
$$a^{(\ell)} = \sigma(W^{(\ell)}a^{(\ell-1)} + b^{(\ell)}) \in \mathbb{R}^{k_\ell}$$

- σ activation function
- $W^{(\ell)}$ weight matrix

 $b^{(\ell)}$ – bias



Artificial Neural Networks

Multi-Layer Perceptron (MLP)

$$f: \mathbb{R}^n \to \mathbb{R}^{k_1} \to \ldots \to \mathbb{R}^{k_{L-1}} \to \mathbb{R}^m$$

$$a^{(0)} = x \in X \subset \mathbb{R}^n$$
$$a^{(\ell)} = \sigma(W^{(\ell)}a^{(\ell-1)} + b^{(\ell)}) \in \mathbb{R}^{k_\ell}$$

- σ activation function
- $W^{(\ell)}$ weight matrix

 $b^{(\ell)}$ – bias



Neural networks learn to

• recognize features



Olah et al. (2017)

Neural networks learn to

- recognize features
- and perform internal computations.



Olah et al. (2020)

Neural networks learn to

- recognize features
- and perform internal computations.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$



Olah et al. (2020)

Neural networks learn to

- recognize features
- and perform internal computations.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$

Feature Representations

• **Distributed**: Features span multiple neurons/layers

Windows (4b:237) excite the car detector at the top and inhibit at the bottom.	Tho Unit	"pe's "Local Code" Example Code Activations Meaning	Features: (Color, Shape) Pairs Superposition: None			
Car Body (4b:491) excites the car fetector, especially at the bottorn. Wheels (4b:373) excite the car detector at the pottorn and inhibit at the top.	ABCDMFGT	White Circle* TRed Circle* TRed Circle* Tree Circle* Tree Circle* Tree Circle* Tree Circle* Tree Circle* Tree Transle* Circle* Tree Transle* Circle* Tree Transle* Transl	Extends to Represent: Color and Shape Mixtures, Sets of (Color, Shape) Pairs Can Linearly Select: Colors, Shapes, Unions of Colors, Unions of Shapes, Any Union of (Color, Shape) Pairs Can't Linearly Select: None			
	-JKJZ20	Blue Tranger Blue Tranger Takes Tranger White Square White Square Takes Square Takes Square Takes Square Takes Square				
	Tho	rpe's "Highly Distributed Code" Example	Features: (Color, Shape) Pairs			
	Unit	Code Activations Meaning - O O O O O O O O O O O O O O O O O O O	Superposition: Yes Extends to Represent: None			
	A B C D		Can't Linearly Select (With threshold) Single (Color, Shape) Pair, Other Possibilities Based on Quirks of Superposition Can't Linearly Select: Colors, Shapes, Most (Color, Shape) sets			

Olah (2023) - informal note

Neural networks learn to

- recognize features
- and perform internal computations.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$

Feature Representations

- Distributed: Features span multiple neurons/layers
- Superposition: Neuron activation patterns overlap



Olah et al. (2022)

Neural networks learn to

- recognize features
- and perform internal computations.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$

Feature Representations

- Distributed: Features span multiple neurons/layers
- Superposition: Neuron activation patterns overlap
- **Polysemanticity**: Individual neurons respond to multiple unrelated concepts



Dreyer et al. (2024)

Task

Disentangle superpositions of distributed features that activate polysemantic neurons.

Status quo: *ICA, NMF, sparse autoencoders* \rightarrow recover *coordinate axes* of the feature space.

Why (directed) topology?

"What fires together, wires together"

- coactivation \rightarrow 'semantic neighbourhoods'
- distributed features \rightarrow unions and intersections
- causality \rightarrow directionality
- polysemantic neurons \rightarrow context-dependent



Idea

Build (filtered) **directed simplicial complex** that represents causal influence of neurons on each other.

Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 4 – Statistical significance test

Step 2 – Pick candidate simplex

Step 5 – Add simplex to complex \mathcal{K}_X

Step 3 - Coherent counterfactual ablation

Coherent counterfactual ablation

Three pieces

- Coherent: keep structural equations intact (stay on neural manifold M).
- Counterfactual: 'what if $a|_P$ was different?'
- Ablation: $a|_P \rightarrow 0$.

CCA = path $\gamma : [0, 1] \rightarrow M$, satisfying all three.

- 1. Fix $x_i \in X$ with activation $a(x_i) \in M \subset \mathbb{R}^N$
- 2. Find tangent vector $v \in T_{a(x_i)}M$ s.t. $a(x_i)|_P$ decreases.
- 3. Integrate to $\gamma(t)$ (flow equation $\dot{\gamma}(t) = v_i \gamma(0) = a(x_i)$)

The pullback $\gamma|_a$ quantifies the causal influence of P on the probe neuron q.





Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 4 – Statistical significance test

Step 2 – Pick candidate simplex

Step 5 – Add simplex to complex \mathcal{K}_X

Step 3 - Coherent counterfactual ablation

Step 4 Detail – Statistical Significance (Wilcoxon)

For probe neuron q (and $x_i \in X$)

- 1. Compute change of activation $\Delta q_i(t) = (\gamma_i(t) \gamma_i(0))|_q$
- 2. Perform Wilcoxon signed-rank test on $\{\Delta q_i(t)\}_{x_i \in X}$
 - Null Hypothesis: median $\Delta q_i(t) = 0$
 - Reject if $p < \alpha = 0.01$

3.
$$t^* = \min\{t : p(t) < \alpha\}$$

(earlier $t^* \implies$ more immediate influence)



Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 4 – Statistical significance test

Step 2 – Pick candidate simplex

Step 5 – Add simplex to complex \mathcal{K}_X

Step 3 - Coherent counterfactual ablation

- MLP Architecture 2 → 4 → 4 → 4 (8 internal neurons)
- Weights ±1, 0, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4



- MLP Architecture 2 → 4 → 4 → 4 (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4



- MLP Architecture $2 \rightarrow 4 \rightarrow 4 \rightarrow 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4



- MLP Architecture 2 → 4 → 4 → 4 (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4



- MLP Architecture 2 → 4 → 4 → 4 (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4







- MLP Architecture 2 → 4 → 4 → 4 (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons:
 x > 0, x < 0, y < 0, y > 0
- second layer neurons: Q1, Q2, Q3, Q4

Jaccard Matrix: All DSCs in Monosemantic Classifier														
Q1 -	1.00	0.00	0.00	0.00	0.12	0.22	0.05	0.00	0.03	0.00	0.22			1.0
Q2 -	0.00	1.00	0.07	0.00	0.04	0.02	0.00	0.13	0.00	0.01	0.02			
Q3 ·	0.00	0.07	1.00	0.03	0.01	0.00	0.00		0.02	0.10	0.00			0.8
Q4 ·	0.00	0.00	0.03	1.00	0.00	0.01	0.04	0.01	0.08	0.09	0.02			
Q1+Q2 ·	0.12	0.04	0.01	0.00	1.00	0.18	0.02	0.01	0.02	0.00	0.20			0.6 ද
Q1+Q3	0.22	0.02	0.00	0.01	0.18	1.00	0.06	0.00	0.04	0.00	0.75			al la
Q1+Q4 ·	0.05	0.00	0.00	0.04	0.02	0.06	1.00	0.00	0.18	0.02	0.07			0.4
Q2+Q3	0.00	0.13	0.32	0.01	0.01	0.00	0.00	1.00	0.02	0.05	0.00			
Q2+Q4	0.03	0.00	0.02	0.08	0.02	0.04	0.18	0.02	1.00	0.06	0.05			0.2
Q3+Q4	0.00	0.01	0.10	0.09	0.00	0.00	0.02	0.05	0.06	1.00	0.00			0.2
Random	0.22	0.02	0.00	0.02	0.20	0.75	0.07	0.00	0.05	0.00	1.00			
										0.0				
Or Or Or Or O' 63,														



- MLP Architecture $2 \rightarrow 3 \rightarrow 4 \rightarrow 4$ (7 internal neurons)
- Weights ±1, 0, one non-zero bias b, ReLU activation
- first layer neurons:
 x, y > 0, x < y, x, y < 0
- second layer (approx.): b < x + y, b < y - x, b < -(x + y),x - y < b



- MLP Architecture $2 \rightarrow 3 \rightarrow 4 \rightarrow 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons:

x, y > 0, x < y, x, y < 0

• second layer (approx.): b < x + y, b < y - x, b < -(x + y),x - y < b





- MLP Architecture $2 \rightarrow 3 \rightarrow 4 \rightarrow 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons:
 x, y > 0, x < y, x, y < 0
- second layer (approx.): b < x + y, b < y - x, b < -(x + y),x - y < b

Monosemantic vs Polysemantic Quadrant Classifier



Example 3: MNIST Digit Classifier

MLP

- MLP Architecture $784 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 10$ (240 internal neurons)
- trained to 97.6 % acc.
- Input samples X_i , i = 0, ..., 9, of 20 'one-hottest' inputs
- Calculated up to 2-simplices
- 7,700 1-simplices ($\approx 10\%$) and 360,000 2-simplices ($\approx 1\%$)



Example 3: MNIST Digit Classifier

MLP

- MLP Architecture $784 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 10$ (240 internal neurons)
- trained to 97.6 % acc.
- Input samples X_i , i = 0, ..., 9, of 20 'one-hottest' inputs
- Calculated up to 2-simplices
- 7,700 1-simplices (≈ 10%) and 360,000 2-simplices (≈ 1%)



Take-away

Directed simplicial complexes may represent distributed features over polysemantic neurons.

Next steps

- directed topological invariants?
- stability under weight perturbations?
- algebra of feature composition and interactions?
- unsupervised dictionary learning?