

Toy models of superposition

ANTHROPIC

Sept 14, 2022

Nelson Elhage*, Tristan Hume*, Catherine Olsson*, Nicholas Schiefer*, Tom Henighan,
Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen,
Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei,
Martin Wattenberg*, Christopher Olah[#]

Section 0: What is Superposition?

Superposition?

Why is it that neurons sometimes align with features and sometimes don't?

Why do some models and tasks have many of these clean neurons, while they're vanishingly rare in others?

In this paper, we use toy models — small ReLU networks trained on synthetic data with sparse input features — to investigate how and when models represent more features than they have dimensions. We call this phenomenon superposition.

왜 뉴런이 때때로 특징과 일치하고 때로는 그렇지 않은가?

왜 일부 모델과 작업에는 이러한 명확한 뉴런이 많이 존재하지만, 다른 모델에서는 그 수가 극히 적은가?

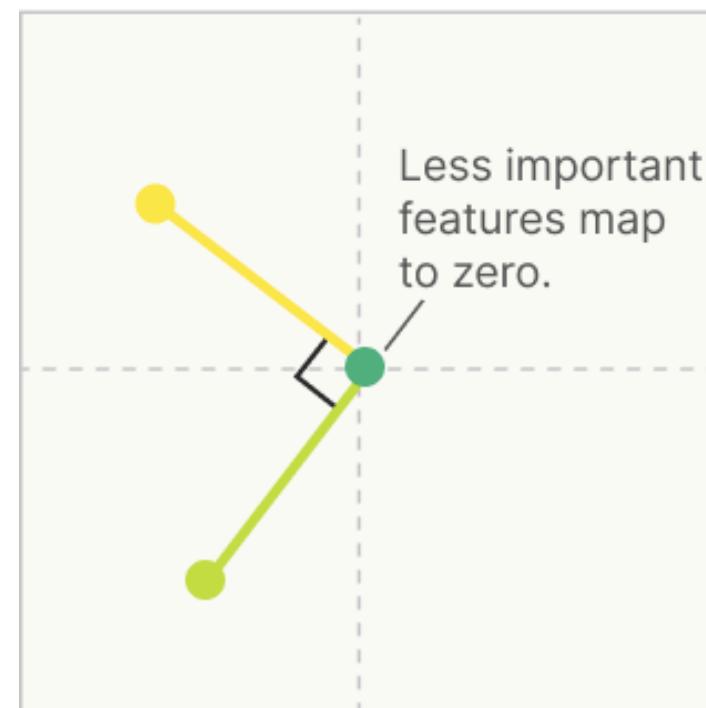
“언제, 어떻게 모델이 차원에 비해 더 많은 feature를 표현할 수 있는가”에 대한 고찰

Superposition?

Sparsity가 증가할수록, 모델은 “Superposition”을 사용해서 차원에 비해 더 많은 feature를 표현하고자 한다.

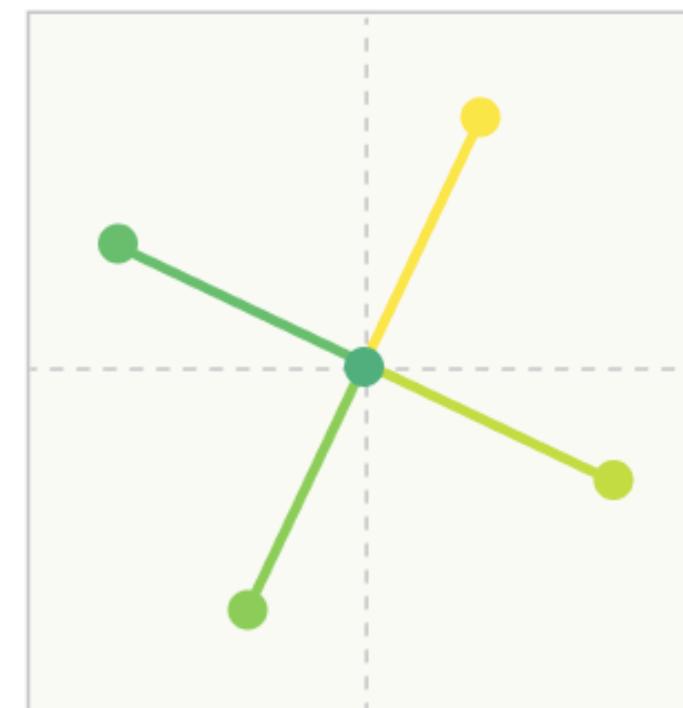
As Sparsity Increases, Models Use “Superposition” To Represent More Features Than Dimensions

Increasing Feature Sparsity



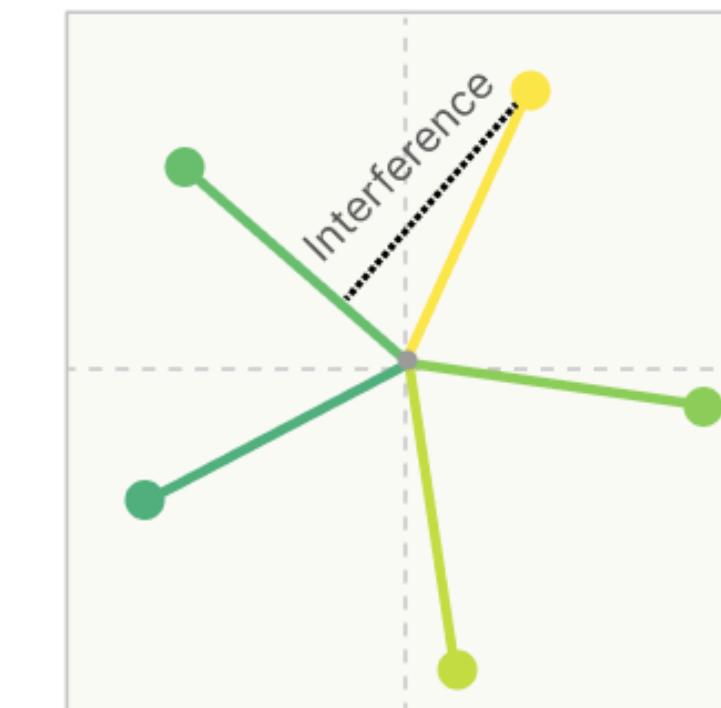
0% Sparsity

The two most important features are given **dedicated orthogonal dimensions**, while other features are **not embedded**.



80% Sparsity

The four most important features are represented as **antipodal pairs**. The least important features are **not embedded**.



90% Sparsity

All five features are embedded as a **pentagon**, but there is now “positive interference.”

Feature Importance

- Most important (Yellow)
- Medium important (Green)
- Least important (Teal)

Superposition? - What is Sparsity?

Toy Model들이 학습하는 synthetic data는 특정 sparsity를 가지게 된다.

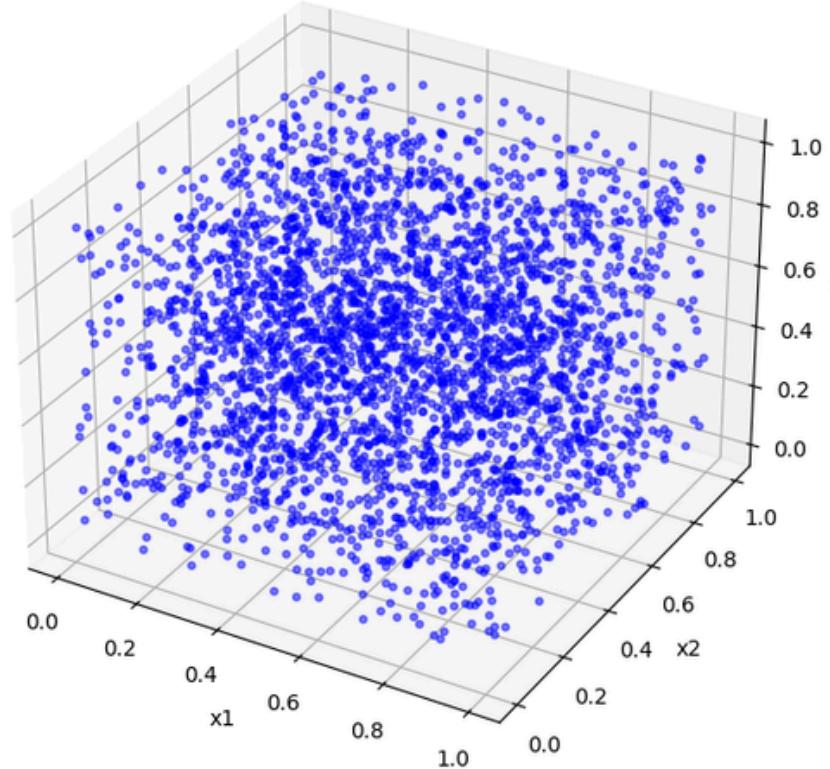
여기서 sparsity는 각 feature가 0이 될 확률을 의미한다.

만약 feature가 3개인 어떤 synthetic data를 시각화해보면, sparsity에 따른 분포는 다음과 같다.

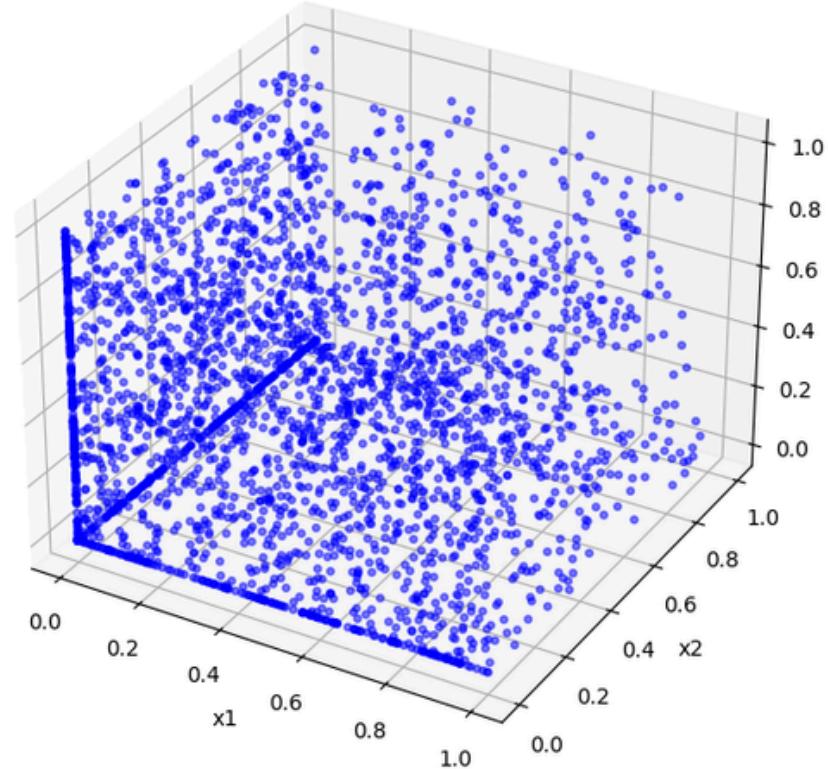
Increasing Feature Sparsity



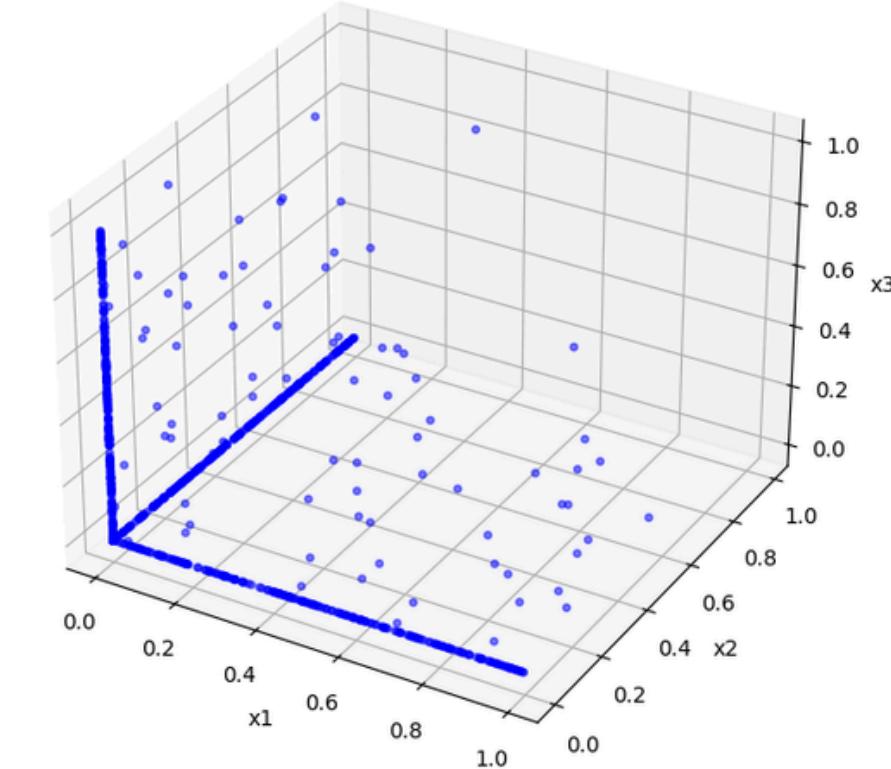
3D Distribution of Sparse Features ($1-S=1$)



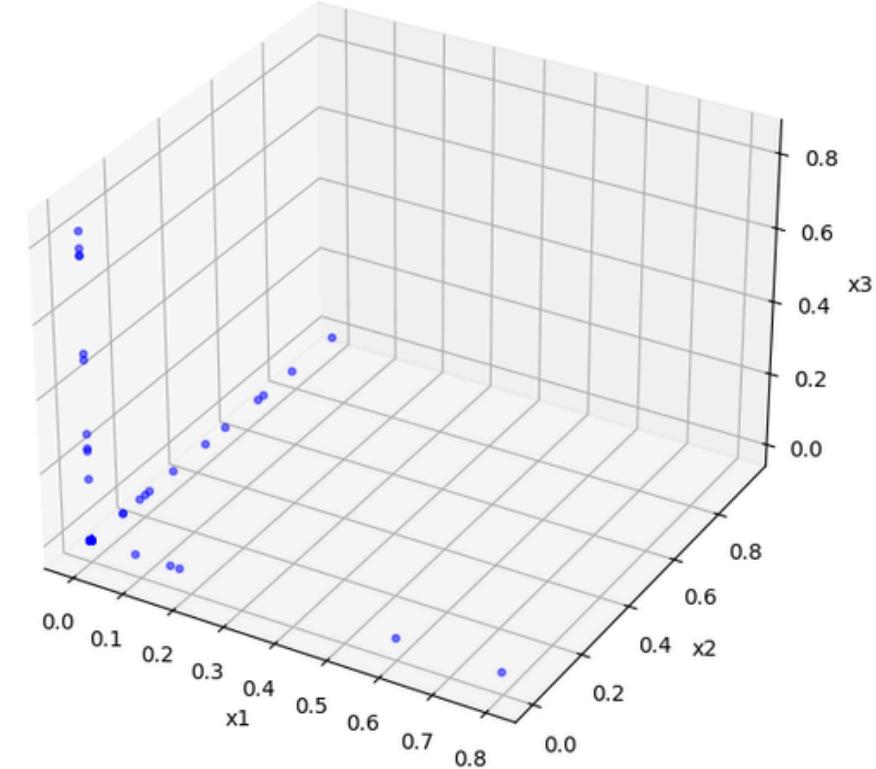
3D Distribution of Sparse Features ($1-S=0.7$)



3D Distribution of Sparse Features ($1-S=0.0999999999999998$)



3D Distribution of Sparse Features ($1-S=0.00300000000000027$)



Superposition? - Why sparseness?

Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?

BRUNO A. OLSHAUSEN,‡ DAVID J. FIELD†

Received 16 July 1996; in revised form 24 December 1996

WHY SPARSENESS?

The reason for conjecturing that sparseness is an appropriate prior for the a_i is based on the intuition that natural images may generally be described in terms of a small number of structural primitives—for example, edges, lines, or other elementary features ([Field, 1994](#)). In addition, one can see evidence for sparse structure in images by filtering them with a set of log-Gabor filters and collecting histograms of the resulting output distributions; these distributions typically show high kurtosis ([Field, 1993](#)), which is indicative of sparse structure.

Superposition? - Why sparseness?

Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?

BRUNO A. OLSHAUSEN,‡ DAVID J. FIELD†

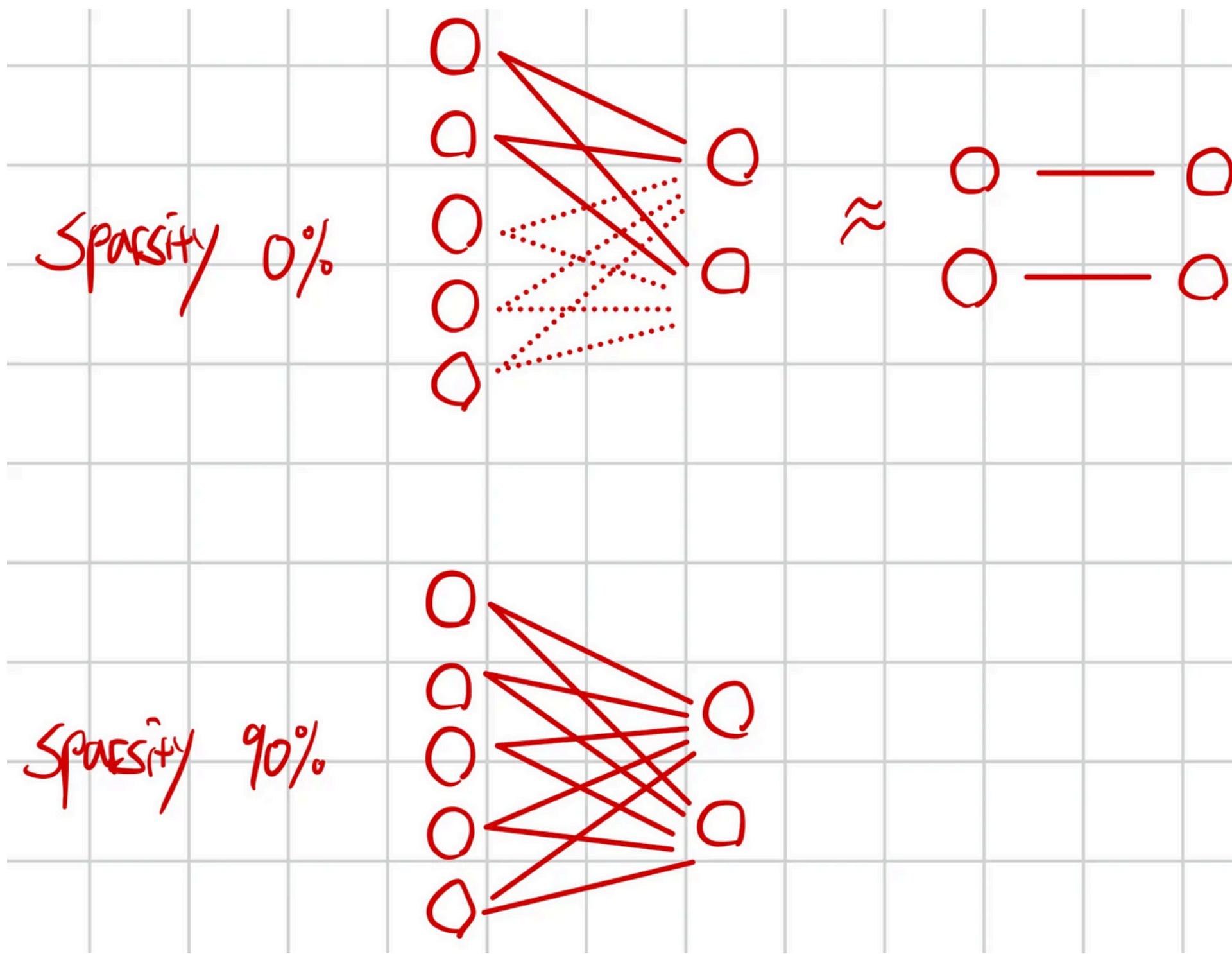
Received 16 July 1996; in revised form 24 December 1996

왜 희소성(sparseness)인가?

희소성이 인공지능(AI)에 적합한 사전 분포(prior)라고 추측하는 이유는, **자연 이미지가 일반적으로 소수의 구조적 기본 요소들(예: 모서리, 선분, 혹은 다른 기본적인 특징들)로 묘사될 수 있다는 직관에 기반한다** (Field, 1994).

또한, 로그-가보(Log-Gabor) 필터 세트를 이용해 이미지를 필터링하고, 그 결과 출력 분포의 히스토그램을 수집하면, 이 분포들이 주로 **높은 kurtosis를 보인다는 사실에서 희소적 구조의 증거를 확인할 수 있다** (Field, 1993). **높은 kurtosis는 희소적 구조를 나타낸다.**

Superposition?



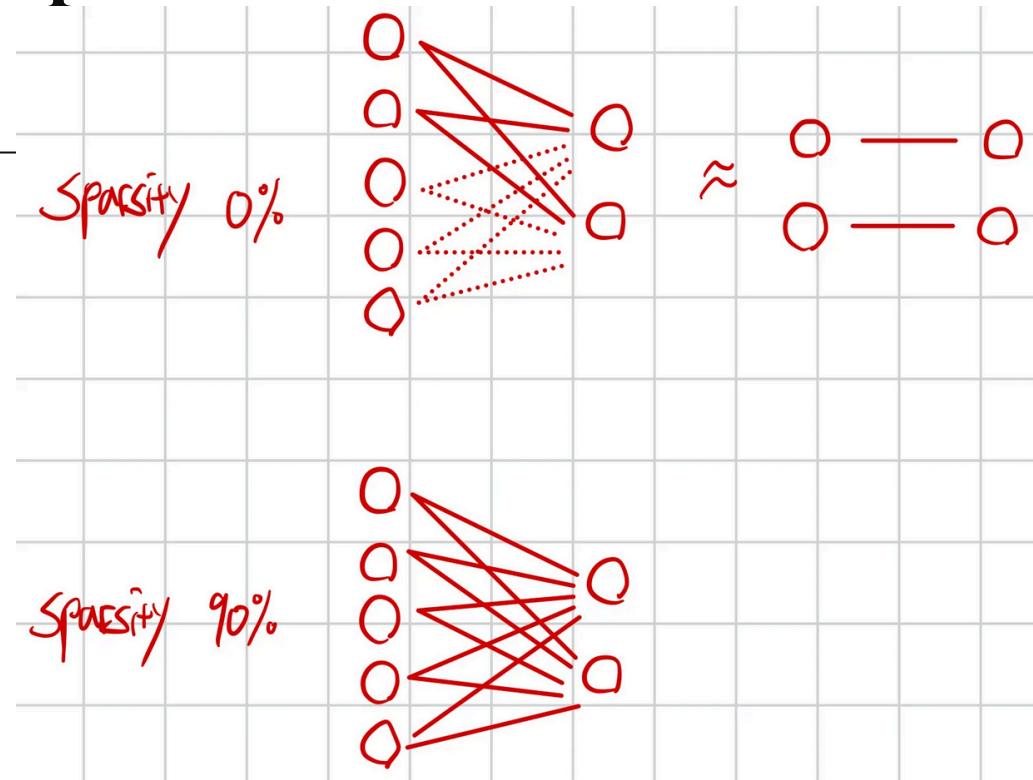
Feature 1: (0.99, 0.01)
Feature 2: (0.01, 0.99)
Feature 3: (-0.01, 0.04)
Feature 4: (-0.02, -0.01)
Feature 5: (0.01, -0.01)

2개 Feature만 살아남는다.
의미를 해석하기 쉽다.
PCA로 임베딩하는 것과 유사

Feature 1: (1.00, 0.00)
Feature 2: (0.31, 0.95)
Feature 3: (-0.81, 0.59)
Feature 4: (-0.81, -0.59)
Feature 5: (0.31, -0.95)

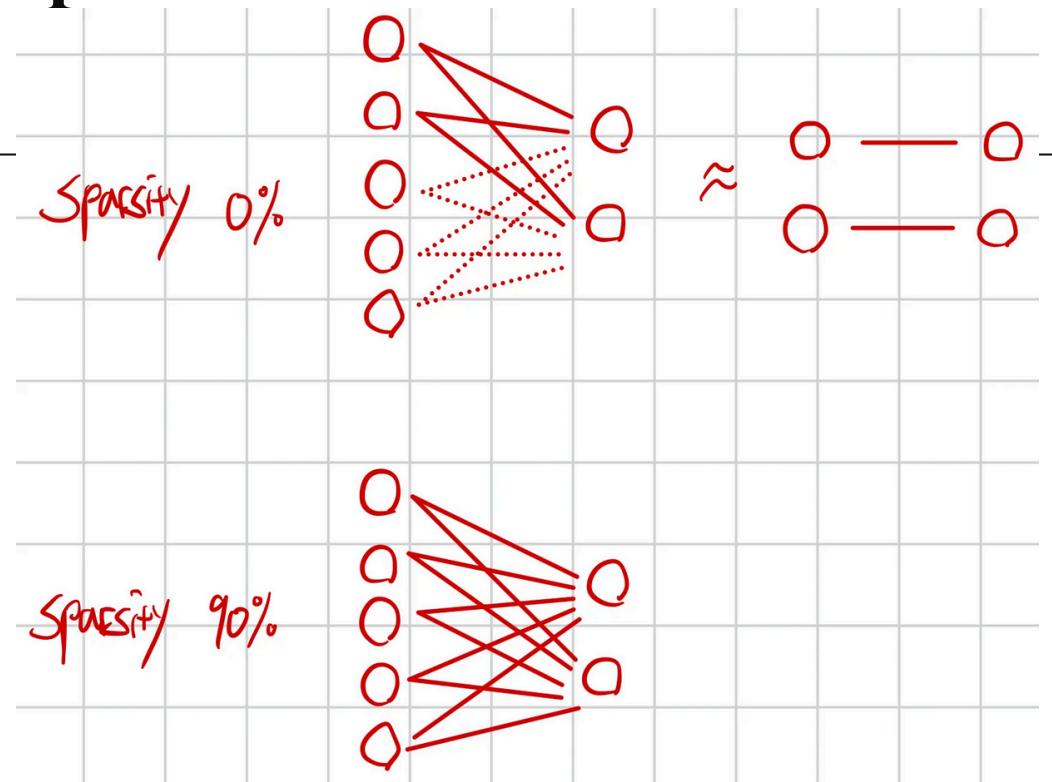
모든 feature를 포함한다.
의미를 알기 어렵다.
학습된 비선형 함수로 임베딩하는 것과 유사

Superposition?



- 0% Sparsity (PCA를 통해 2벡터로 데이터를 표현하는 것과 유사함)
 - 중요한 2가지 특징이 orthogonal dimension에 할당
 - 덜 중요한 특징 3가지는 0으로 매핑되어 표현되지 않음 (비활성화)
 - Sparsity가 없으므로 **독립적인 특징 표현이 가능**하지만, **전체 특징 공간을 비효율적으로 사용**한다.
- 80% Sparsity
 - 중요한 4가지 특징이 antipodal pairs로 표현
 - 덜 중요한 특징은 여전히 0으로 매핑 (비활성화)
 - 일부 특징은 **독립적이지 못**하지만, sparsity가 증가하면서 **공간 사용이 더 효율적**이다.
- 90% Sparsity (선형 모델에 ReLU를 추가한 모델을 통해 데이터를 표현하는 것과 유사함)
 - 5개의 모든 특징이 오각형으로 표현
 - 특징 간에 positive **interference**가 발생한다. (한 표현의 특징이 다른 표현의 특징에 영향을 미친다.)

Superposition?



Sparsity가 낮을수록

- 중요한 특징이 **독립적으로 표현**되므로 해석 가능성이 높다.

Sparsity가 높을수록

- 신경망은 더 많은 특징을 하나의 뉴런에 압축하여 **표현**할 수 있다.
- 이는 신경망이 더 **효율적으로 정보를 저장하고 처리**할 수 있게 하지 만, 특징들 간의 **간섭(interference)**이 발생할 수 있다.

Superposition?

이걸 어떻게 생각해 볼 수 있을까?

CLIP의 경우, $224 \times 224 \times 3$ 차원의 데이터를 512차원의 공간에서 표현한 것이다.

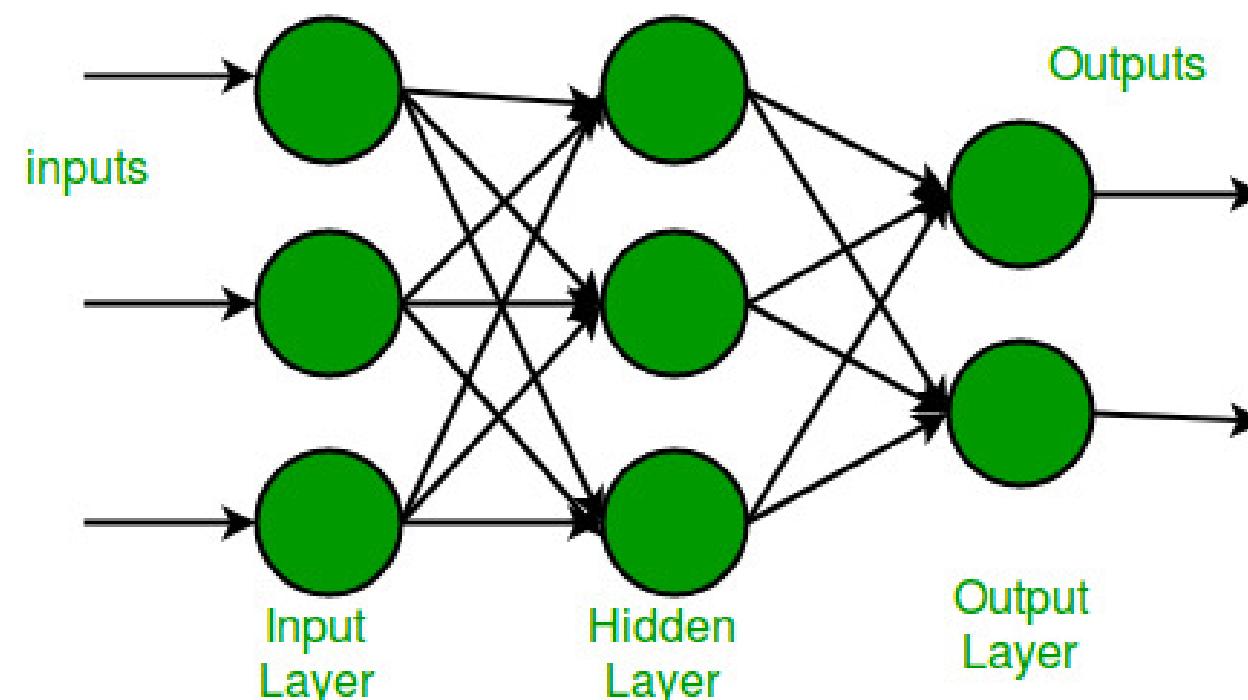
그렇다면 CLIP embedding은 오직 512가지(혹은 그 이하)의 요소에 의해서만 구분되어지는가? -> 아마도 아닐 것이다.

512차원의 CLIP embedding에는 512차원 이상의 basis로 구분되어질 것이다.

고차원의 정보를 저차원으로 옮기면서 이러한 현상이 발생한다면, mlp도 이러한 관점에서 해석이 가능하다.

차원을 증가시켰다가 감소시키면서, 본 차원이 가지고 있는 의미 정보보다 더 많은 의미 정보를 찾게 될 것으로 예상된다.

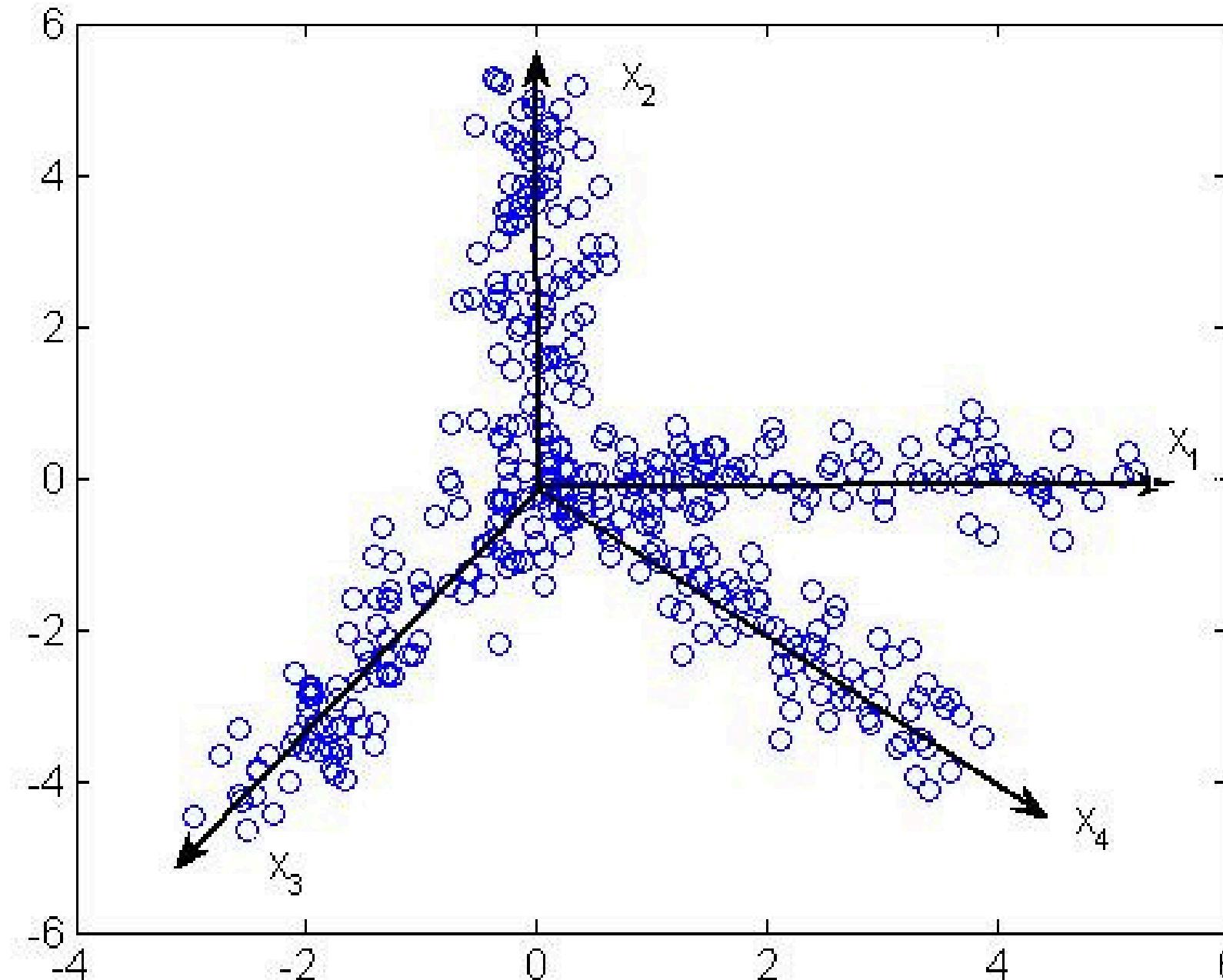
mlp에서 출력 차원을 n개로 조절한다는 것은, input vector를 구분지을 수 있는 n개의 feature를 찾아달라는 의미와 같을 것이다.



Superposition?

Overcomplete basis

데이터를 표현하기 위해 사용되는 벡터 공간의 기저(basis)가 데이터의 차원보다 더 많은 경우

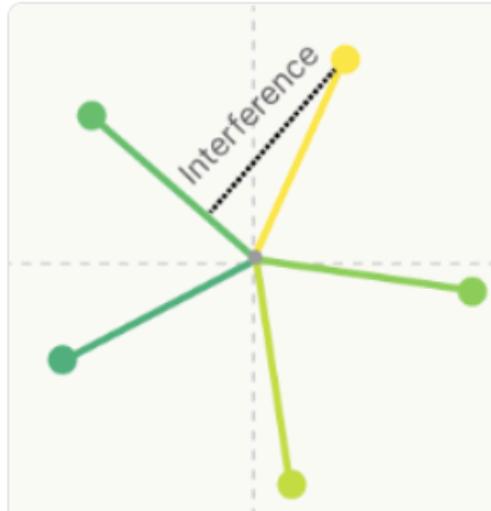


Sparse한 feature에 대해서, 모델은 이를 overcomplete basis를 찾고자 하며, 이는 activation function 때문에 발생하게 된다.

Key Results of this paper

- Superposition is a real, **observed phenomenon**
 - **Both** monosematic and polysemantic neurons can form
 - At least **some kinds of computation** can be performed in superposition
 - Whether features are stored in superposition is governed by a **phase change**
 - Superposition **organizes features into geometric structures** such as digons, triangles, pentagons, and tetrahedrons.
-
- Superposition은 **실제로 관찰된 현상**이다.
 - Monosematic 및 Polysemantic 뉴런이 **모두** 형성될 수 있다.
 - 적어도 **일부 종류의 계산**은 superposition 상태에서 수행될 수 있다.
 - Superposition 상태에서 feature가 저장되는지 여부는 **phase change(위상 변화)**에 의해 결정된다.
 - Superposition은 feature를 **기하학적 구조로 조직화**한다.

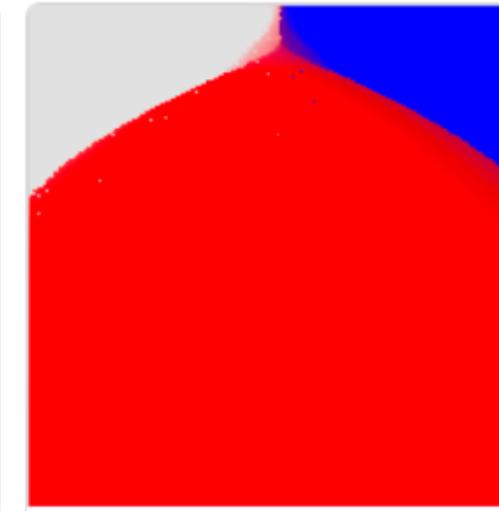
Index



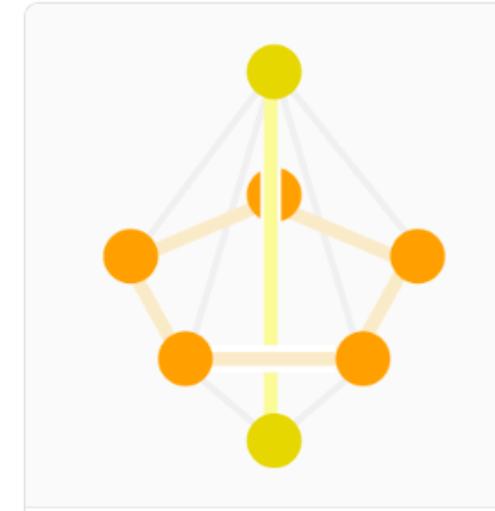
SECTION 1
Background & Motivation



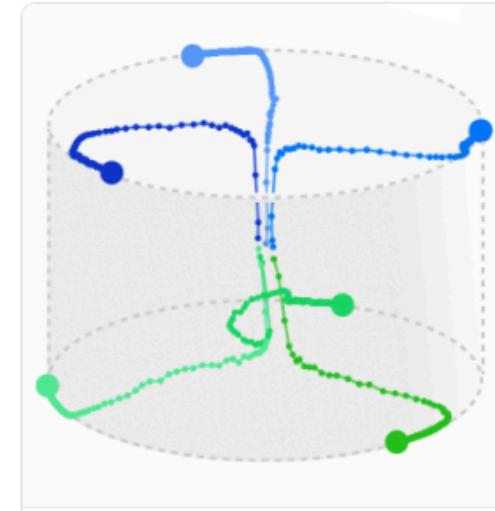
SECTION 2
Demonstrating Superposition



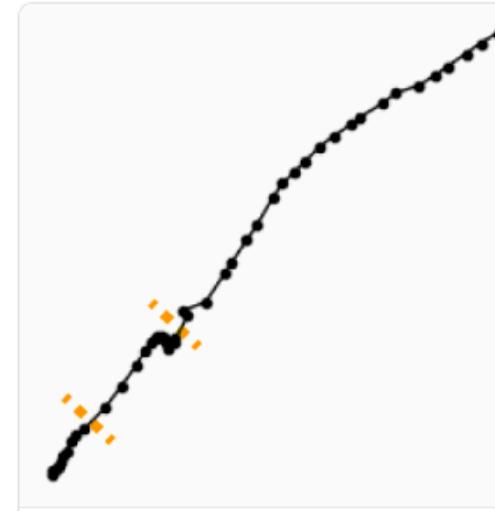
SECTION 3
Superposition as a Phase Change



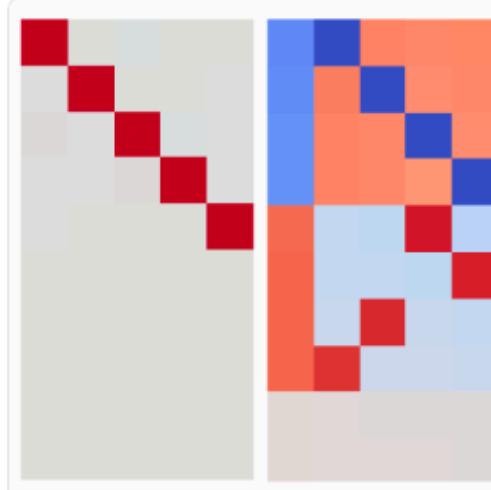
SECTION 4
The Geometry of Superposition



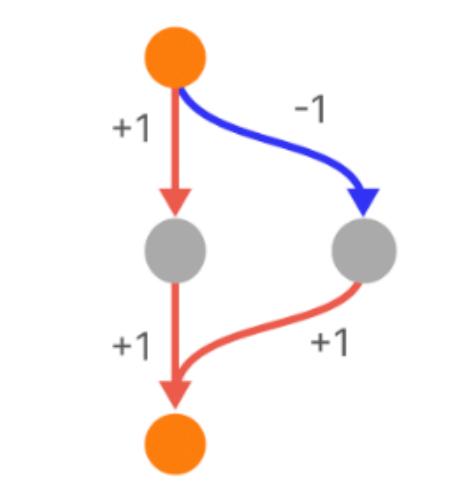
SECTION 5
Learning Dynamics



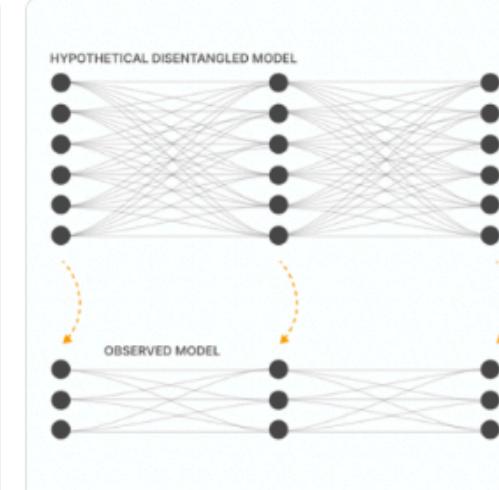
SECTION 6
Relationship to Adversarial Examples



SECTION 7
Superposition in a Privileged Basis



SECTION 8
Computation in Superposition



SECTION 9
The Strategic Picture

Discussion
Does this occur in real models?
Open Questions

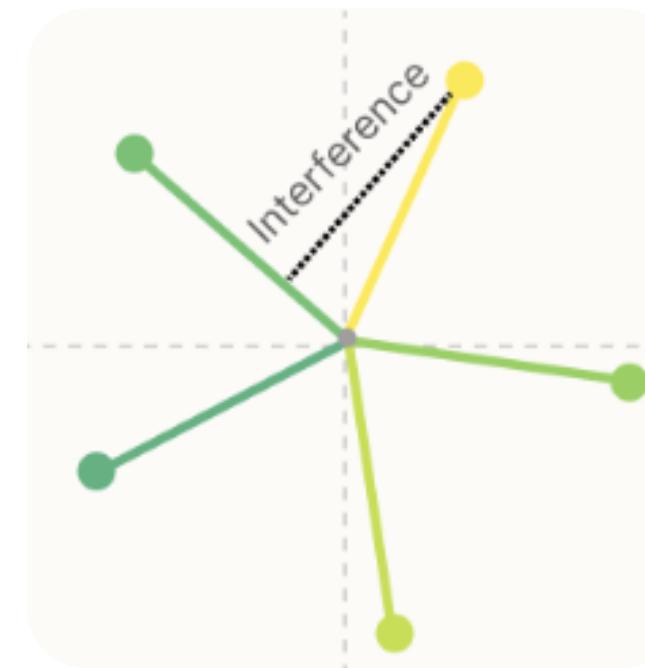
SECTION 10
Discussion

Related Work

SECTION 11
Related Work

Comments & Replications

SECTION 12
Comments & Replications



Section 1: Background & Motivation

Section 1: Background & Motivation

Linear representation hypothesis

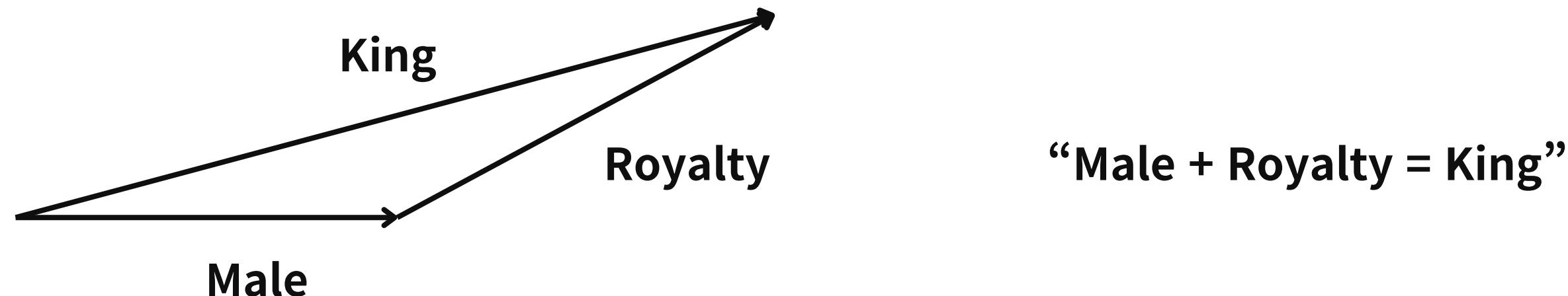
고차원 공간에서 언어 모델의 표현 공간에서 고수준의 개념들이 선형적으로 표현된다는 가설

Decomposability

Network representations can be described in terms of **independently understandable features**

Linearity

Features are represented by **direction**



Section 1: Background & Motivation

Superposition이 발생하는 원인에 대해, 2가지의 반대되는 힘이 있기 때문이라는 가정을 세웠다.

Privileged Basis - Feature들을 기저 방향과 정렬하도록 유도하는 힘

Only some representations have a privileged basis which encourages features to **align with basis directions** (i.e. to correspond to neurons)

Superposition - Feature들이 뉴런과 대응되지 않도록 밀어내는 힘

Linear representations can represent more features than dimensions, using a strategy we call superposition.

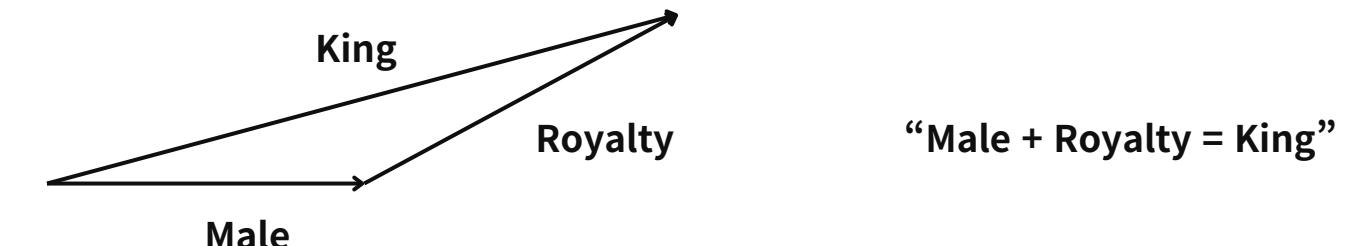
This can be seen as neural networks simulating larger networks.

This **pushes features away from corresponding to neurons**.

Section 1: Background & Motivation > Empirical Phenomena

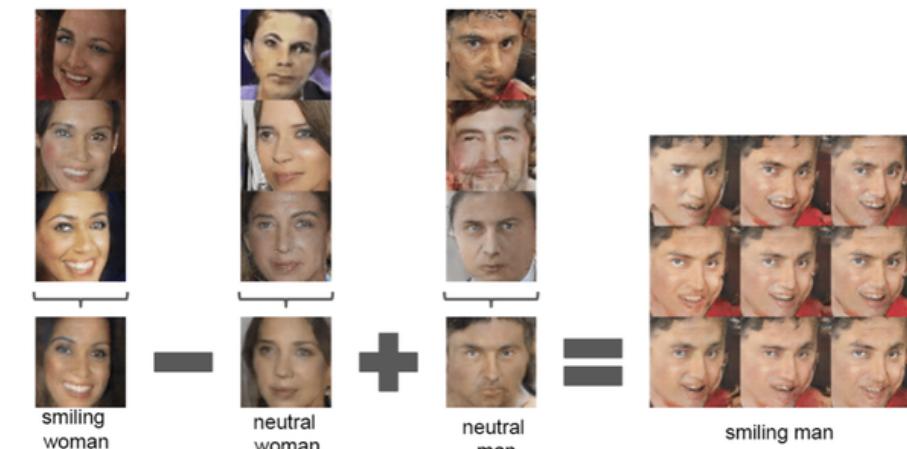
Word Embeddings

단어 임베딩에는 의미적 속성에 대응하는 방향이 존재, 이를 통해 임베딩 산술 벡터 연산이 가능하다.
 $V("king") - V("man") + V("woman") = V("queen")$



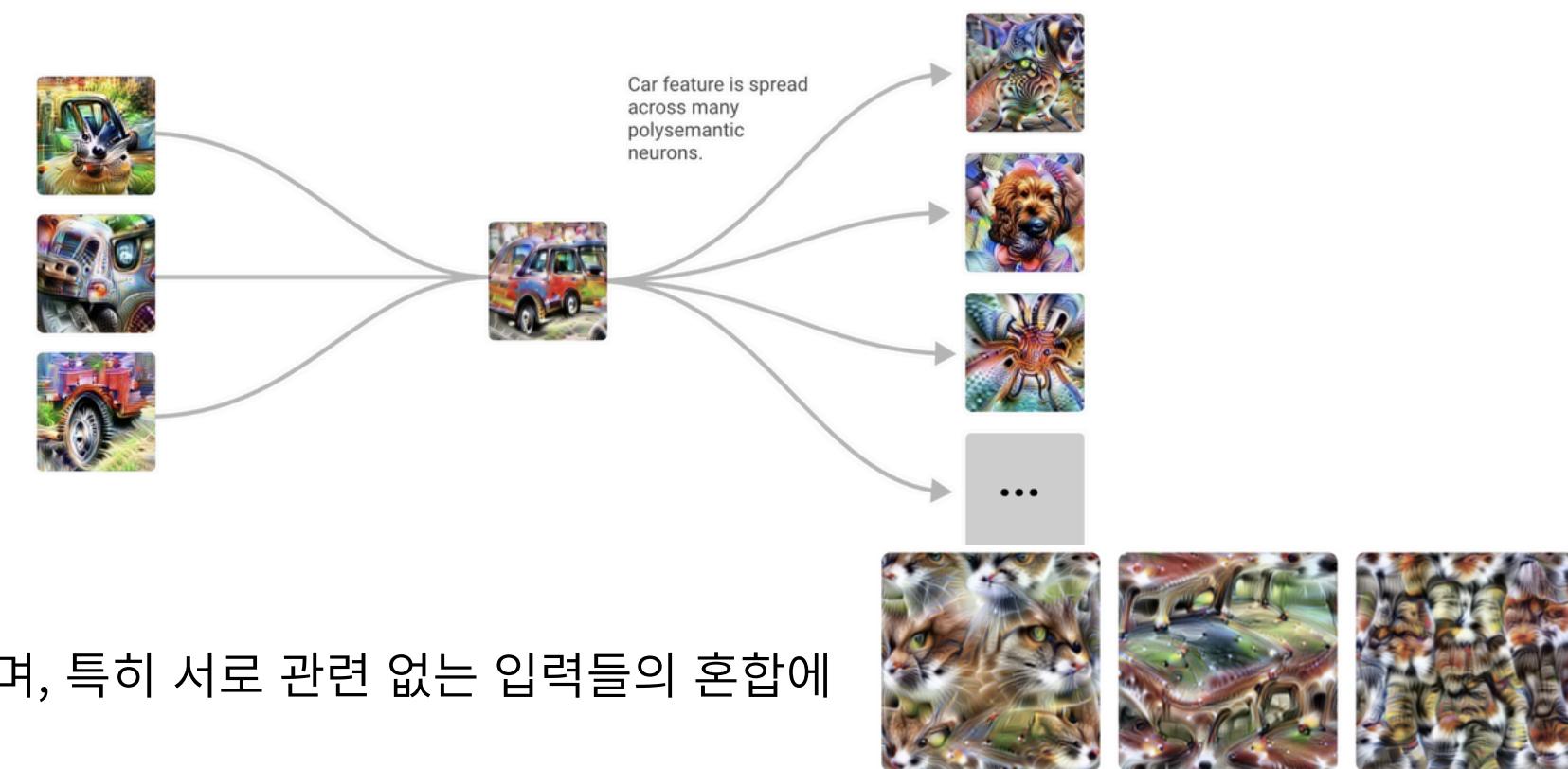
Latent Spaces

GAN에서도 비슷한 "벡터 연산"과 interpretable한 방향성이 발견되었다.



Interpretable Neurons

많은 연구에서 해석 가능한 뉴런들이 발견되었으며, 이들은 이해 가능한 특성에 반응한다.



Universality

동일한 특성에 반응하는 유사한 뉴런들이 여러 네트워크에서 발견될 수 있다.

Polysemantic Neurons

동시에, 입력의 interpretable 특성에 반응하지 않는 것처럼 보이는 많은 뉴런들이 있으며, 특히 서로 관련 없는 입력들의 혼합에 반응하는 것으로 보이는 다수의 polysemantic 뉴런들이 존재한다.

4e:55 is a polysemantic neuron which responds to cat faces, fronts of cars, and cat legs. It was discussed in more depth in Feature Visualization [4].

Section 1: Background & Motivation > What are Features?

~~Feature : 우리가 관찰하는 input의 interpretable한 속성~~

Features as arbitrary functions

Feature를 입력의 임의 함수로 정의하는 것은 불충분하다. 관찰된 feature들은 데이터에 대한 기본적 추상화이며 여러 모델에서 일관되게 나타난다. 또한 개별적으로 구분 가능하다 - 예를 들어 '고양이'와 '자동차'는 개별 feature지만, '고양이+자동차'는 feature의 조합이다.

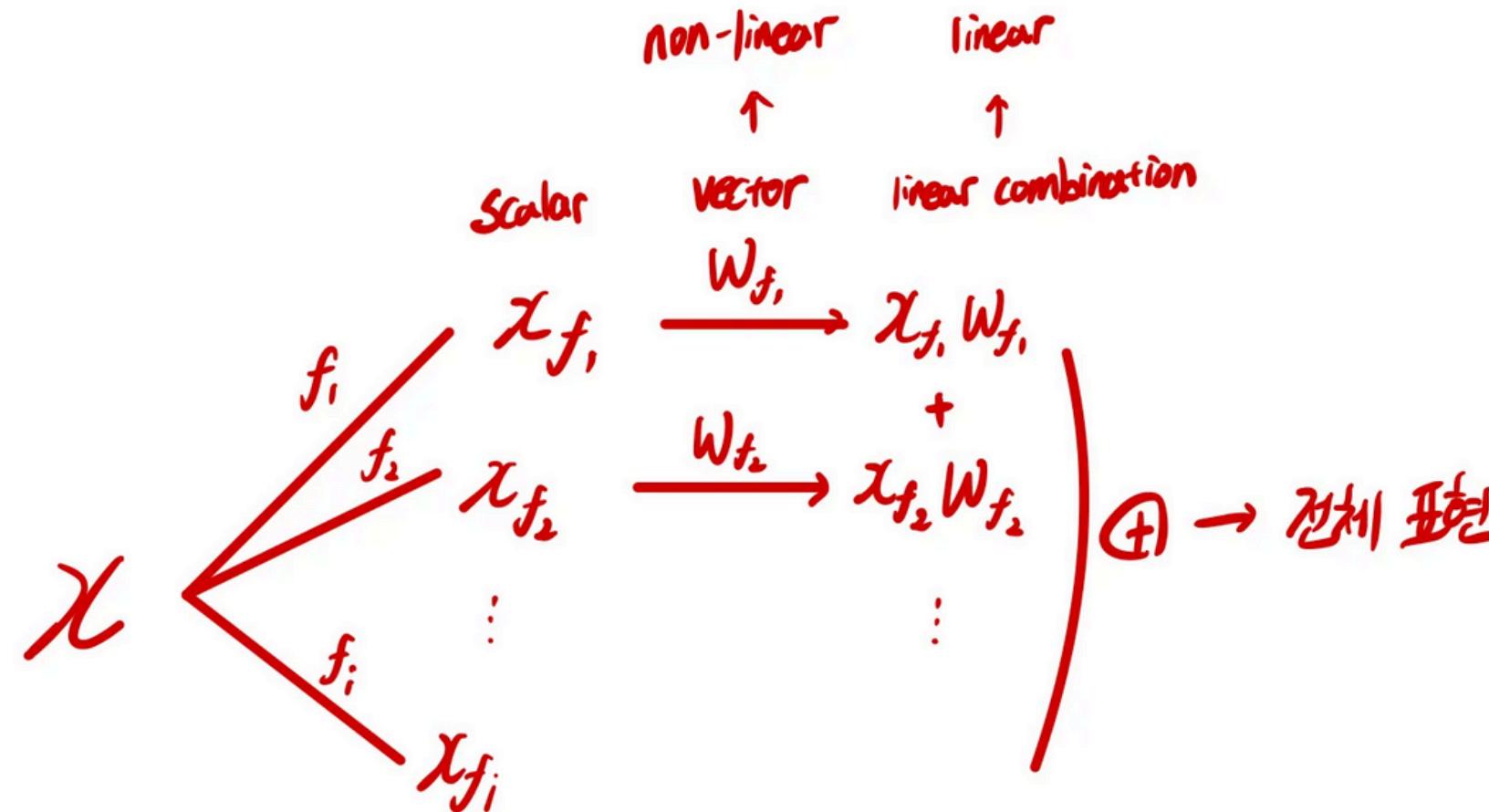
Features as interpretable properties

설명된 특징들은 모두 인간이 이해하기 쉽다. 이를 "인간이 이해할 수 있는 개념의 존재"로 정의할 수 있지만, AlphaFold와 같은 AI가 발견하는 단백질 구조처럼 우리가 처음에는 이해하지 못하는 특징들도 포함해야 한다.

Neurons in Sufficiently Large Models

마지막으로, 충분히 큰 신경망이 특정 뉴런을 할당하여 표현하는 입력의 속성을 feature로 정의한다. 충분히 큰 모델에서는 polysemantic 뉴런에서만 관찰되는 속성들도 전용 뉴런이 생길 것으로 기대된다.

Section 1: Background & Motivation > Features as Directions



표현되는 특징 w 들은 거의 확실히 입력의 비선형 함수
오직 feature에서 활성화 벡터로의 맵핑만이 선형 (선형 결합)

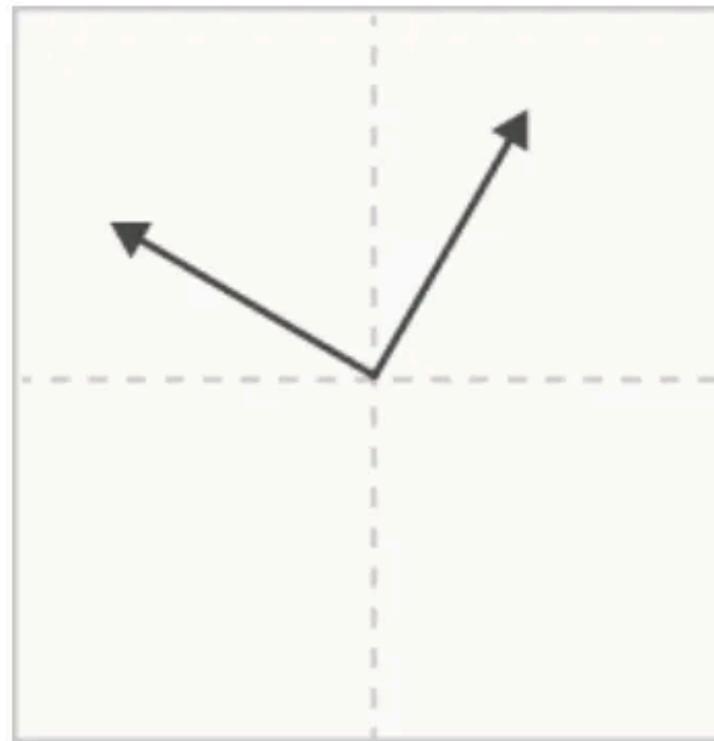
신경망은 **non-linearity**가 산재된 **linear function**들로 구성되어 있다.

$$\text{"King"} = x = x_{\text{gender}}W_{\text{gender}} + x_{\text{royalty}}W_{\text{royalty}}$$

$$x \begin{matrix} \nearrow w_{f_1} \\ \searrow w_{f_2} \end{matrix} \Rightarrow \text{non-linear}$$

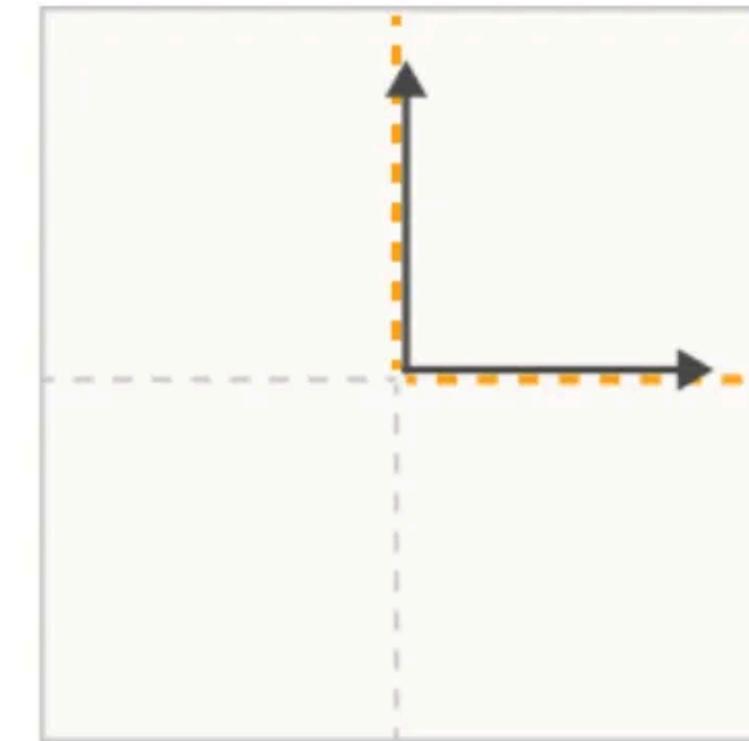
$$x_{\text{gender}}W_{\text{gender}} + x_{\text{royalty}}W_{\text{royalty}} \approx \text{"King"} \Rightarrow \text{linear}$$

Section 1: Background & Motivation > Privileged vs Non-privileged Bases



In a **non-privileged basis**, features can be embedded in any direction. There is no reason to expect basis dimensions to be special.

Examples: word embeddings, transformer residual stream



In a **privileged basis**, there is an incentive for features to align with basis dimensions. This doesn't necessarily mean they will.

Examples: conv net neurons, transformer MLPs

Non-privileged Basis (비특권적 기저)

- 비특권적 기저란, **특정 방향들이 특별하지 않은 기저**
- 만약 임베딩 공간에 임의의 선형 변환 M 을 적용하고, 그에 맞게 가중치에 M^{-1} 를 적용하면, 특성들은 여전히 같은 방식으로 표현되지만, 벡터 공간의 기저가 완전히 다르게 변할 수 있다.
- 이는 비특권적 기저의 대표적인 예로, 기저 자체는 해석과 상관없다는 뜻

Privileged Basis (특권적 기저)

- 특권적 기저란, **특정한 방향들이 특별한 의미를 가지는 기저**
- 신경망의 특정 레이어에서는 활성화 함수(activation function)와 같은 아키텍처적 요소가 대칭을 깨뜨리고, 특정 방향들을 더 중요한 방향으로 만들 수 있다.
- 신경망의 뉴런들은 종종 **해석 가능한 특성**과 직접 연결된다. 예를 들어, 곡선 탐지 뉴런이나 고양이 얼굴 탐지 뉴런과 같은 해석 가능한 뉴런들이 바로 이런 특권적 기저에 존재할 가능성이 크다.

Section 1: Background & Motivation > Privileged vs Non-privileged Bases

Privileged Basis (특권적 기저)

- 특권적 기저란, 특정한 방향들이 특별한 의미를 가지는 기저
- 신경망의 특정 레이어에서는 활성화 함수(activation function)와 같은 아키텍처적 요소가 대칭을 깨뜨리고, 특정 방향들을 더 중요한 방향으로 만들 수 있다.
- 신경망의 뉴런들은 종종 해석 가능한 특성과 직접 연결됩니다.
ex) 곡선 탐지 뉴런이나 고양이 얼굴 탐지 뉴런과 같은 해석 가능한 뉴런들이 바로 이런 특권적 기저에 존재할 가능성이 크다.



Neuron 4b:409



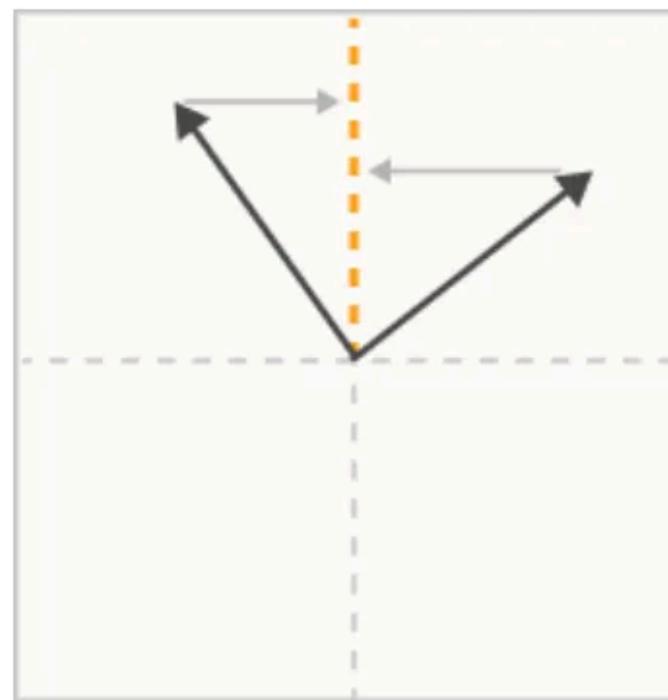
Dataset examples for neuron 4b:409

Section 1: Background & Motivation > The Superposition Hypothesis

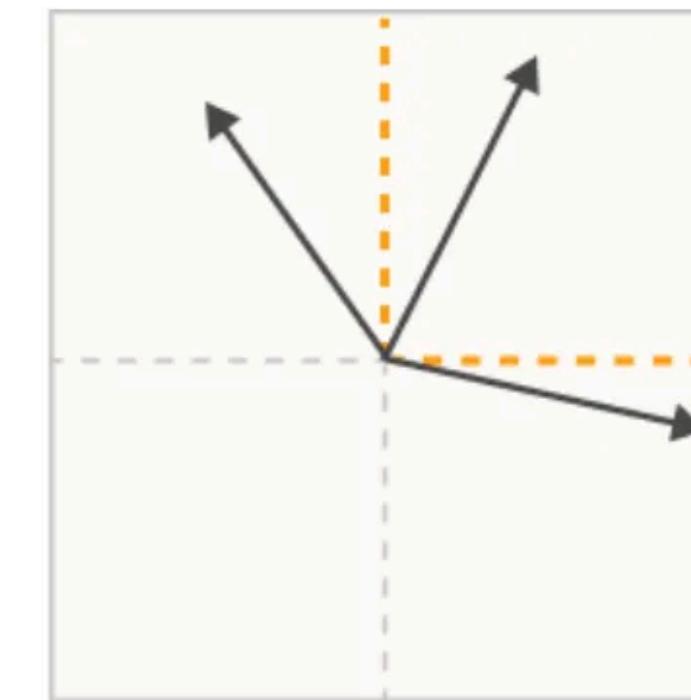
Privileged basis가 있을 때에도, 뉴런들이 "**polysemanticic**"해서 여러 무관한 특징에 반응하는 경우가 많다.

이에 대한 설명은 **superposition hypothesis**이 될 수 있다.

전반적으로 superposition의 아이디어는 신경망이 "**뉴런보다 더 많은 특징을 표현하고 싶어한다**"는 것이다.



Polysemanticity is what we'd expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.



In the **superposition hypothesis**, features can't align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

이것들을 수학적으로 해석해보자면,

Almost Orthogonal Vectors

n-dimensional 공간에서는 n개의 직교 벡터를 가질 수만 있지만, 고차원 공간에서는 $\exp(n)$ 만큼의 "**almost orthogonal**"한 벡터를 가질 수 있다.

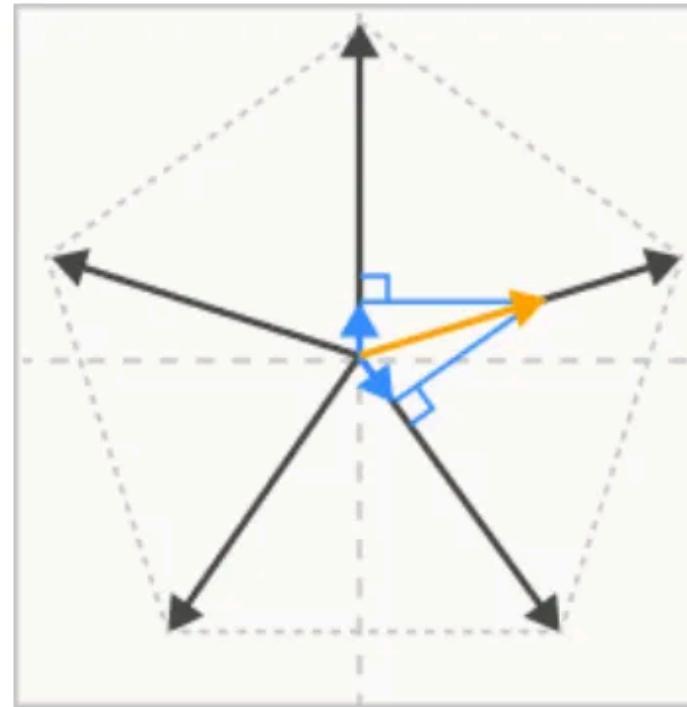
(Johnson-Lindenstrauss lemma: 고차원 공간의 점들을 측정한 거리를 거의 보존하면서 저차원 공간에 투영할 수 있음을 보장하는 수학적 결과)

Compressed Sensing

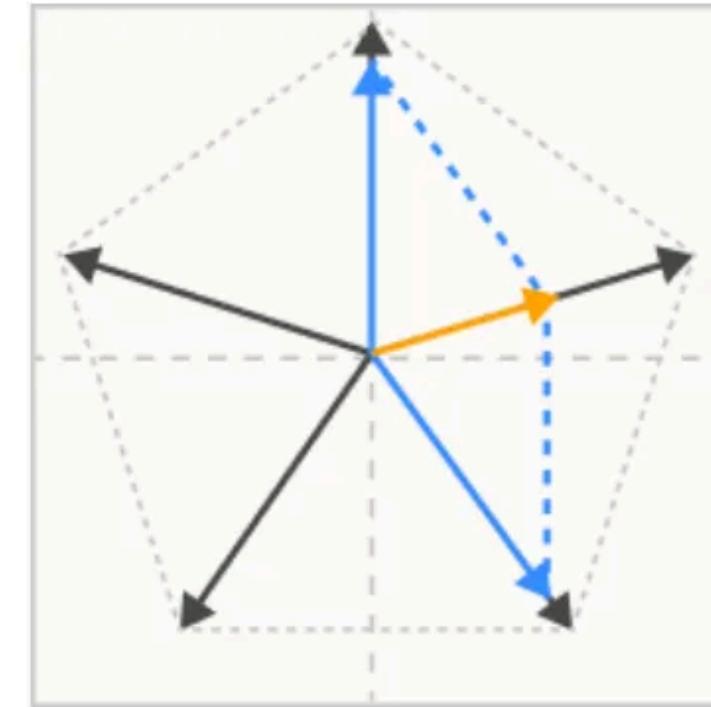
일반적으로, 벡터를 저차원 공간에 투영하면 원래 벡터를 재구성할 수 없다.

그러나 원래 벡터가 sparse하다는 것을 알면 원래 벡터를 복구할 수 있는 경우가 있다.

Section 1: Background & Motivation > The Superposition Hypothesis



Even if only **one sparse feature** is active, using linear dot product projection on the superposition leads to **interference** which the model must tolerate or filter.



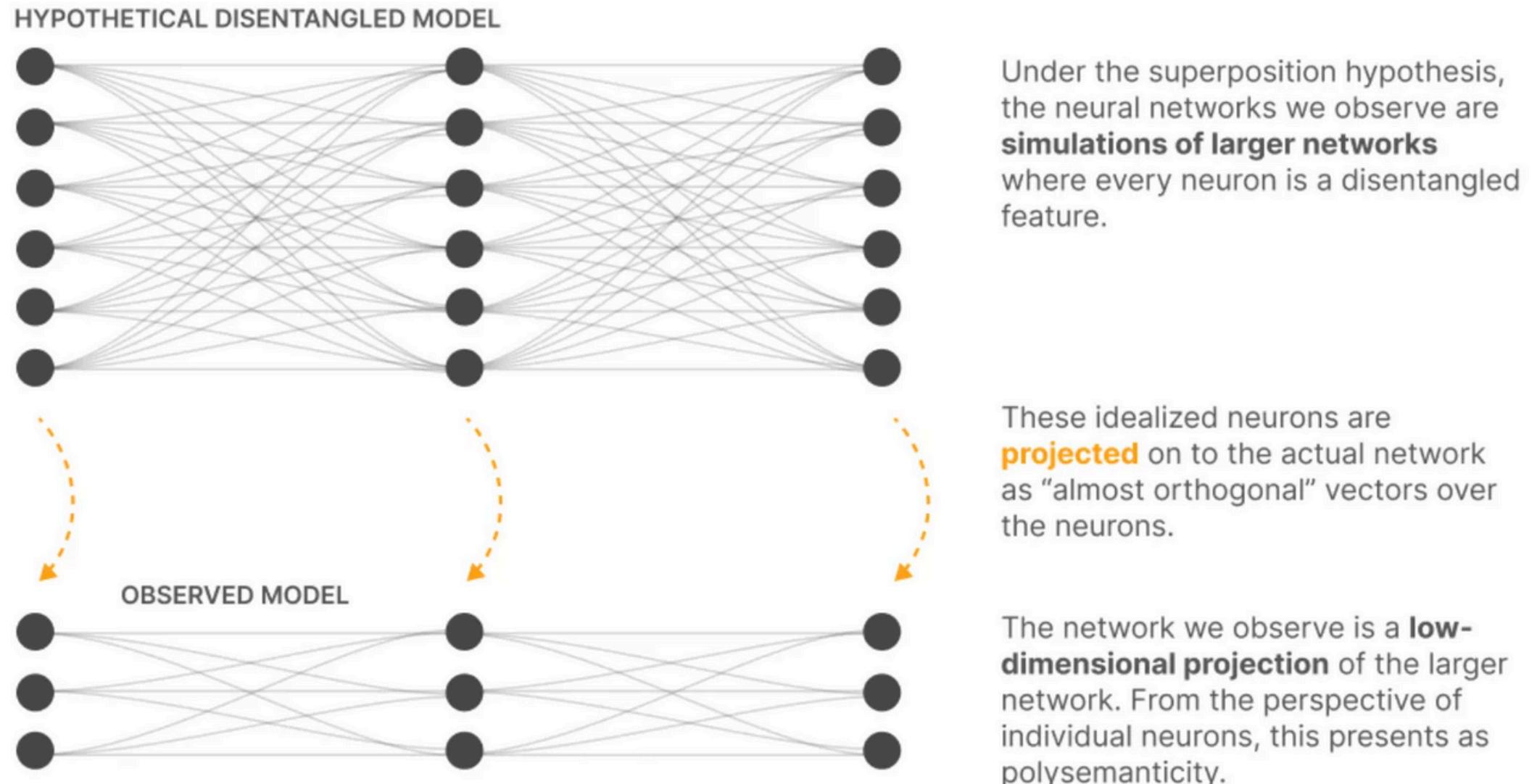
If the features aren't as sparse as a superposition is expecting, **multiple present features** can additively interfere such that there are multiple possible nonlinear reconstructions of an **activation vector**.

Superposition이 신경망에서 차원을 초과하는 특징을 압축적으로 표현할 수 있는 유용한 방법이다.

Superposition이 기대하는 대로 **특징이 충분히 희소하지 않으면** 간섭(interference)이 커지고, 재구성의 불확실성도 증가한다.

superposition hypothesis에서는 특징이 “**거의**” 직교하기 때문에, 하나의 특징이 활성화되면 다른 특징들이 **약간** 활성화되는 것처럼 보인다. 이러한 "noise" 또는 "interference"을 허용하는 데는 비용이 들 것이다. 하지만, 신경망이 sparse한 feature를 가지고 있다면, 이러한 비용은 더 많은 특징을 표현할 수 있는 이점에 의해 초과될 수 있다.

Section 1: Background & Motivation > The Superposition Hypothesis



- Hypothetical Disentangled Model
 - 모든 뉴런이 각각의 독립적인 특징(disentangled feature)을 표현한다고 가정하는 이상적인 구조
 - 각 뉴런은 superposition 없이 하나의 **명확하고 독립적인 의미**를 가진다.
- Observed Model
 - 실제 신경망은 가설적인 네트워크의 저차원 투영(low-dimensional projection)으로 나타난다.
 - 뉴런 간에 superposition이 발생하여 **다수의 특징이 동일한 뉴런에 인코딩**된다.

Section 1: Background & Motivation > Summary: A Hierarchy of Feature Properties

지금까지의 아이디어를 정리하자면, 신경망 표현이 가지는 속성으로 4가지가 있다.

Decomposability

분해 가능한 신경망 활성화는 다른 feature 값에 의존하지 않는 의미로서 **feature**로 분해될 수 있다.

Linearity

Feature는 방향에 해당한다. 각 특성 f 와 해당 표현 벡터 W 를 가지고 있고, 선형 결합으로 특성을 나타낼 수 있다.

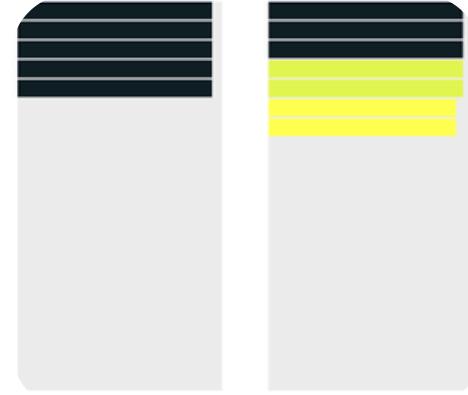
Superposition vs Non-superposition

선형 표현은 $W^T W$ 가 비가역적이면 superposition을 나타내고, 가역적이라면 superposition을 나타내지 않는다.

Basis-Aligned

표현이 basis-aligned되었다고 할 수 있는 경우, 모든 W_i 가 one-hot basis 벡터이다.

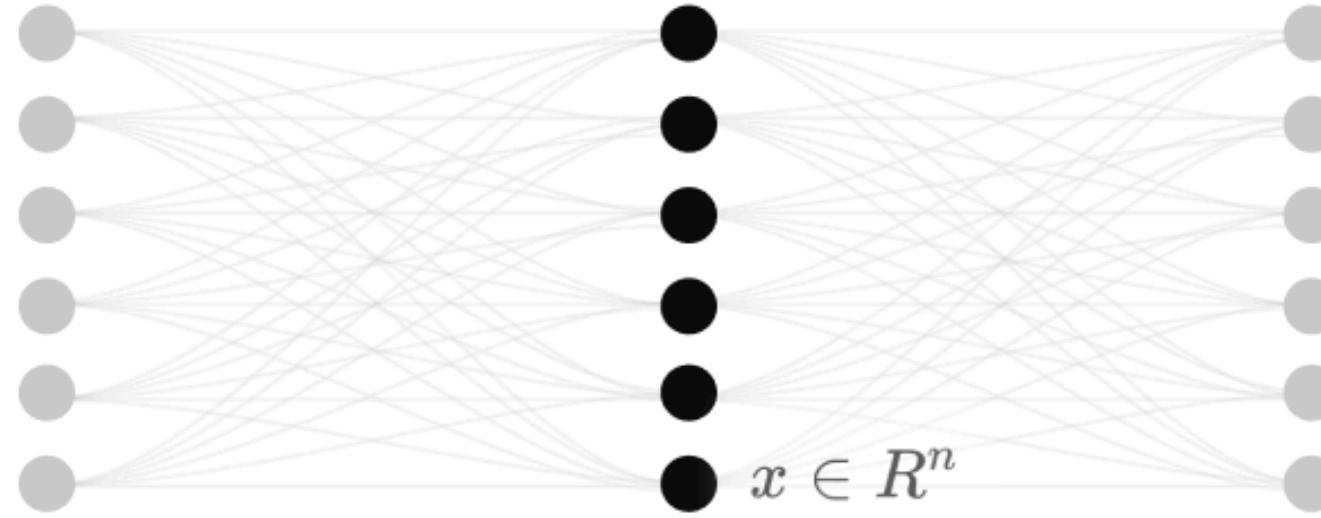
모든 W_i 가 sparse할 경우 표현은 부분적으로 기저 정렬되어 있다고 할 수 있다. 이는 Privileged Basis를 필요로 한다.



Section 2: Demonstrating Superposition

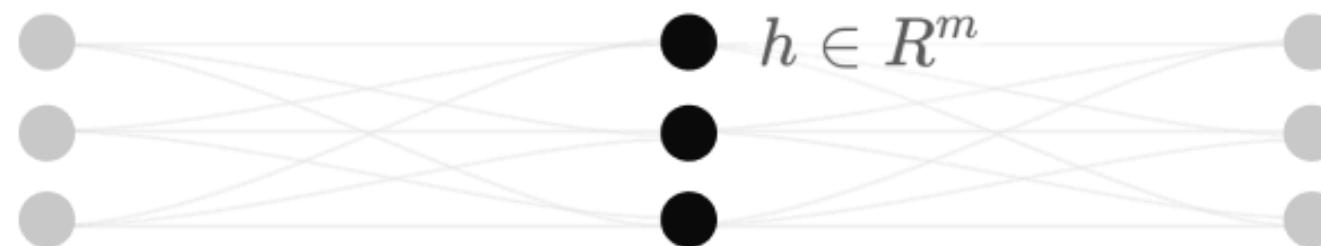
Section 2: Demonstrating Superposition > Experiment Setup

HYPOTHETICAL DISENTANGLLED MODEL



Our first experiments will test the extent to which the idealized activations of an imagined larger model can be **stored** and **recovered** from a lower-dimensional space.

OBSERVED MODEL



우리의 목표는 신경망이 고차원 벡터 $x \in R^n$ 를 저차원 벡터 $h \in R^m$ 로 투영한 후 다시 복원할 수 있는지를 탐구하는 것

→ 신경망에서, vector들은 sparse하게 저장될 것이고, sparse한 vector들은 고차원으로 복구가 가능하다는 것을 증명하고자 한 것 같다.

Section 2: Demonstrating Superposition > Experiment Setup

The Feature Vector (X)

고차원의 벡터 x : idealize된, hypothetical disentangled model의 활성화이다.

Feature가 가상의 더 큰 모델의 뉴런과 완벽하게 align되어 있다고 상상하고 있기 때문에, 각 요소 x_i 를 "feature"라고 부른다.

ex) Vision model에서는 이것이 gabor filter, a curve detector, or a floppy ear detector일 수 있고, 언어 모델에서는 특정 유명인을 언급하는 토큰이나 특정 종류의 설명이 되는 절을 나타낼 수 있다.

Section 2: Demonstrating Superposition > Experiment Setup

Feature Sparsity

자연 세계에서, 많은 feature는 드물게 발생한다는 점에서 sparse해 보인다.

ex) vision에서는 이미지의 대부분 위치에 수평 모서리, 곡선, 또는 개의 머리가 포함되지 않으며, 언어에서는 대부분의 토큰이 마틴 루터 킹을 언급하지 않거나 음악을 설명하는 절의 일부가 아니다. 이러한 이유로 우리는 **feature에 대해 sparse distribution을 선택할 것이다.**

More Features Than Neurons

모델이 표현할 수 있는 **잠재적으로 유용한 feature**가 엄청 많다.

- Feature: 모델이 학습해야 하거나 표현할 수 있는 잠재적인 정보, 패턴, 속성.
- Neuron: 이러한 feature를 표현하거나 처리할 수 있는 신경망의 구성 요소. = Dimension

Features Vary in Importance

모든 features가 주어진 작업에 대해 **동일하게 유용하지는 않다.**

일부는 다른 것보다 손실을 더 많이 줄일 수 있다. 서로 다른 개 품종을 분류하는 것이 main task인 ImageNet 모델의 경우, 늘어진 귀 탐지기는 그것이 가질 수 있는 가장 중요한 특성 중 하나일 수 있지만, 다른 특성은 성능을 아주 조금만 향상시킬 수 있다.

Section 2: Demonstrating Superposition > Experiment Setup

The Model ($x \rightarrow x'$)

- Linear model : superposition이 나타나지 않는, 잘 이해되는 baseline
- ReLU output model : superposition이 나타나는 아주 간단한 모델

두 모델은 마지막 activation function만 다르다.

Linear Model

$$h = Wx$$

$$x' = W^T h + b$$

$$x' = W^T W x + b$$

ReLU Output Model

$$h = Wx$$

$$x' = \text{ReLU}(W^T h + b)$$

$$x' = \text{ReLU}(W^T W x + b)$$

Superposition hypothesis에 따르면, higher-dimensional model의 각 **feature**는 lower-dimensional space의 **direction**에 해당

이것은 우리가 $h = Wx$ 의 선형 맵으로의 down projection이 가능하다는 것을 의미

각 열 W_i 가 lower-dimensional space에 **feature x_i** 를 의미한다는 것을 주목하자.

Section 2: Demonstrating Superposition > Experiment Setup

Linear Model

$$h = Wx$$

$$x' = W^T h + b$$

$$x' = W^T W x + b$$

ReLU Output Model

$$h = Wx$$

$$x' = \text{ReLU}(W^T h + b)$$

$$x' = \text{ReLU}(W^T W x + b)$$

W^T 의 사용

lower-dimensional space에서의 direction이 실제 feature에 해당하는 것인지에 관한 모호성을 피하는 것에 이점을 가지고 있다.

bias

Bias는 모델이 표현하지 않는 특성을 기댓값으로 설정할 수 있도록 해 준다.

나중에 보겠지만, negative bias를 설정하는 것은 superposition에 중요하다. (대략적으로 말하자면, 모델이 **약간의 noise를 무시할** 수 있게 해 준다.)

Activation function

superposition이 발생하는지의 여부와 매우 중요하다. 실제 신경망에서 특성이 실제로 모델에 의해 계산에 사용될 때, 활성화 함수가 존재할 것이므로, 마지막에 활성화 함수를 포함하는 것이 원칙적이다.

Section 2: Demonstrating Superposition > Experiment Setup

The Loss

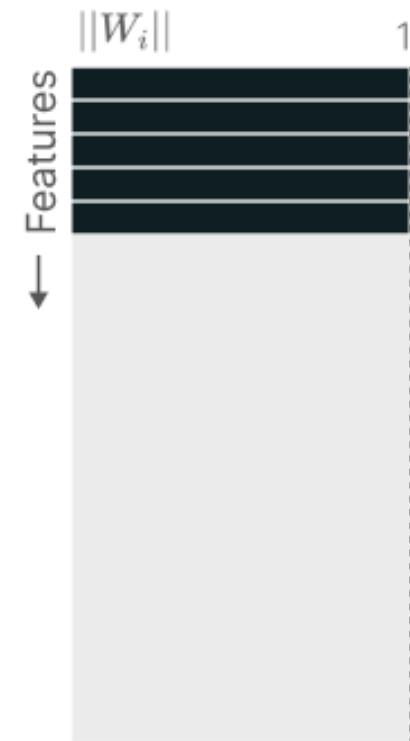
Loss는 위에서 설명한 특성 중요도 $|_i$ 로 가중치가 부여된 MSE(mean squared error) 사용한다.

$$L = \sum_x \sum_i I_i (x_i - x'_i)^2$$

Section 2: Demonstrating Superposition > Basic Results



It tends to be easier to visualize $W^T W$ than W . Here we see that $W^T W$ is an **identity matrix** for the most important features and **0** for less important ones.



We want to understand which features the model chooses to represent in its hidden representation, and whether they're orthogonal to each other.

To do this, we visualize the norm of each feature's direction vector, $\|W_i\|$. This will be ~ 1 if a feature is fully represented, and zero if it is not. For each feature, we also use color to visualize whether it is orthogonal to other features (i.e. in superposition).

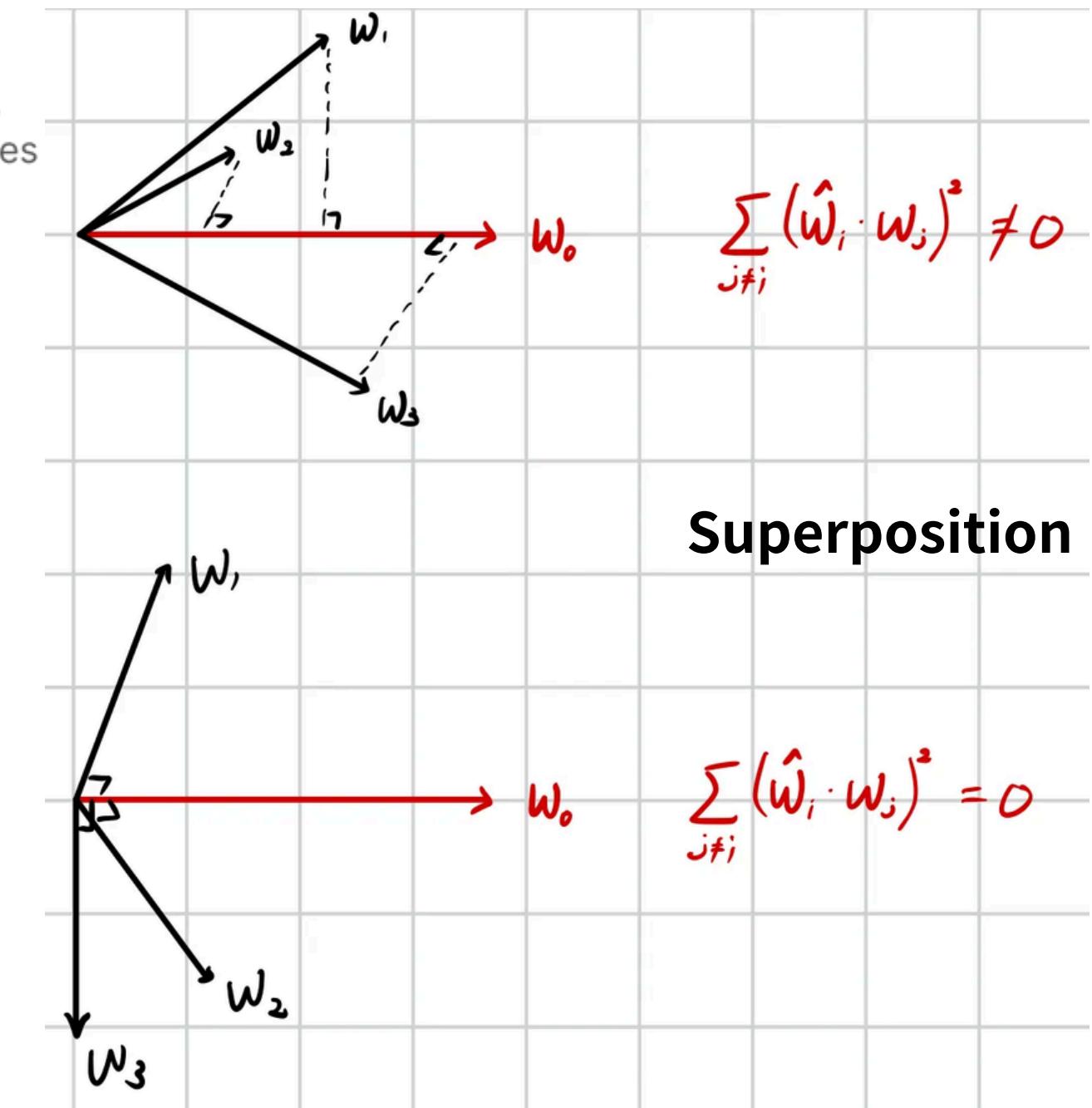
This model simply dedicates one dimension to each of the most important features, representing them orthogonally.

b

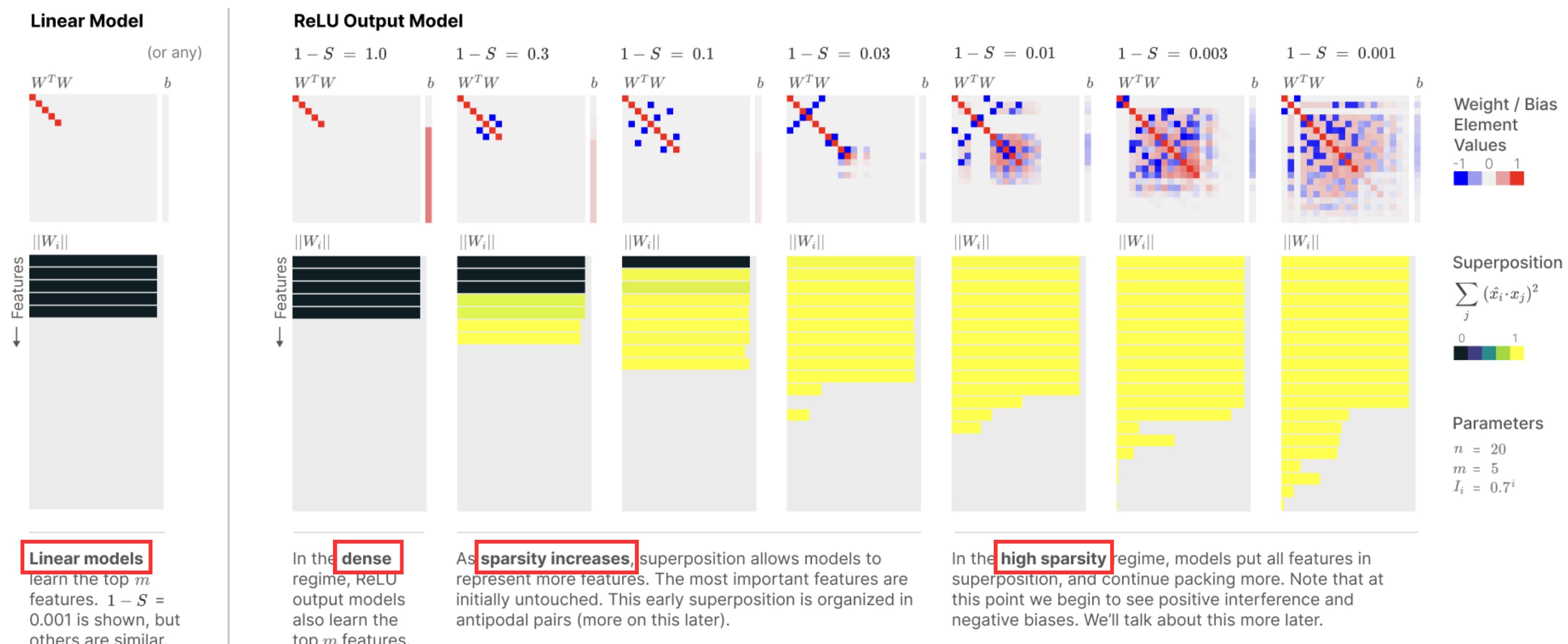
We can also look at the bias, b . The bias is **zero** for features learned to pass through, and the **expected value** (a positive number) for others.

Weight / Bias Element Values

-1 0 1



Section 2: Demonstrating Superposition > Basic Results

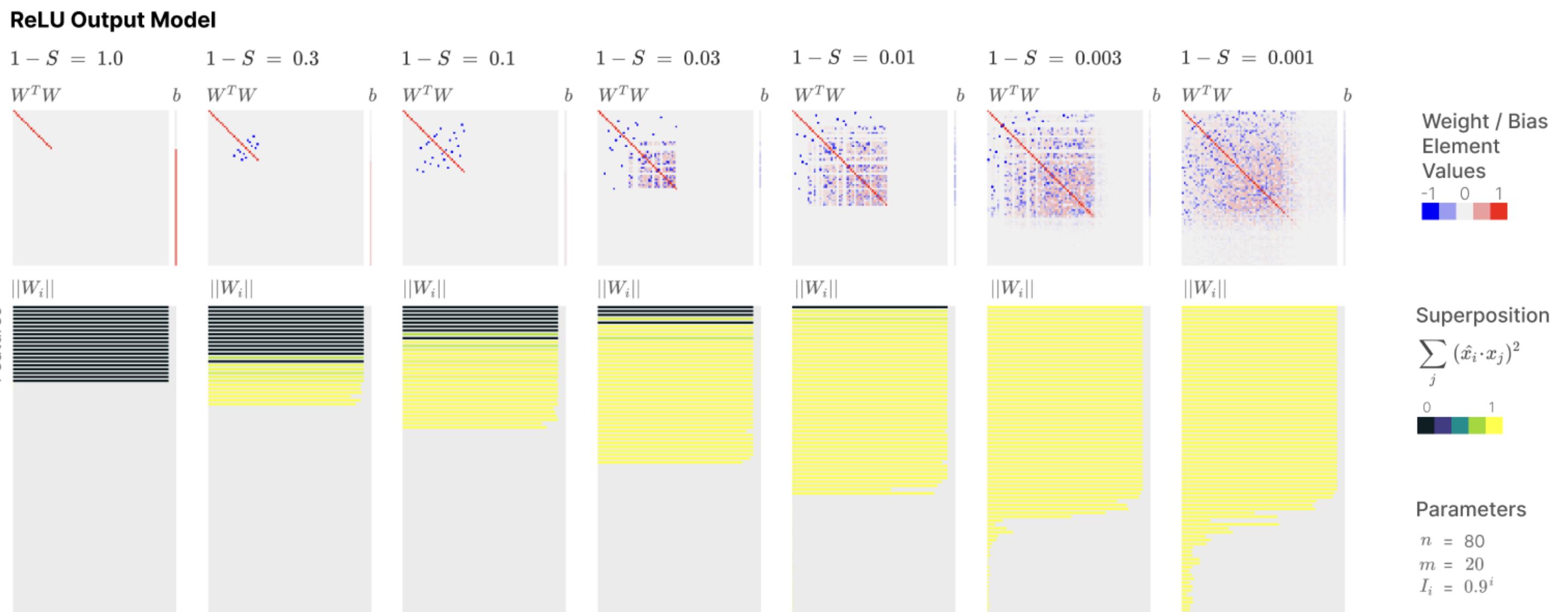
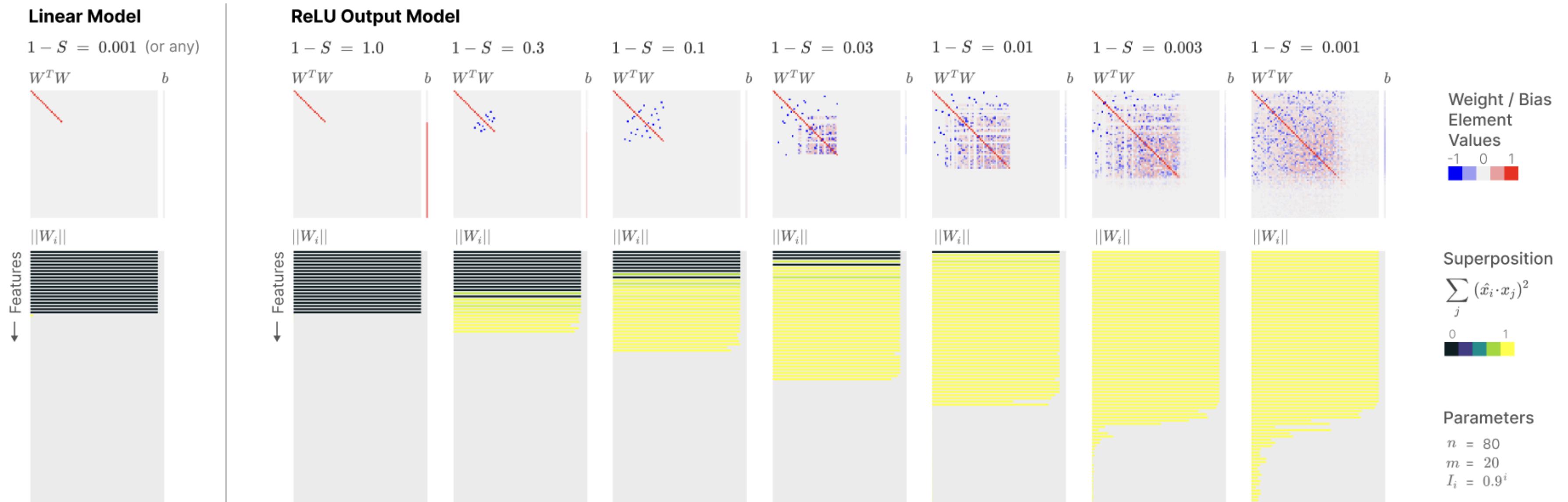


5차원의 공간에서 5개의 feature가 모두 orthogonal하게 배치된 상태

초기 단계에서는 antipodal pairs 존재, 일부만 orthogonal한 상태

모든 feature가 orthogonal하지 않은 상태

Section 2: Demonstrating Superposition > Basic Results



Weight / Bias
Element
Values

-1	0	1
■	■	■

Superposition

$$\sum_j (\hat{x}_i \cdot x_j)^2$$

0	1
■	■

Parameters

$n = 80$
 $m = 20$
 $I_i = 0.9^i$

Section 2: Demonstrating Superposition > Basic Results

$$L \sim \sum_i I_i (1 - \|W_i\|^2)^2 + \sum_{i \neq j} I_j (W_j \cdot W_i)^2$$

Feature benefit is the value a model attains from representing a feature. In a real neural network, this would be analogous to the potential of a feature to improve predictions if represented accurately.

Interference between x_i and x_j occurs when two features are embedded non-orthogonally and, as a result, affect each other's predictions. This prevents superposition in linear models.

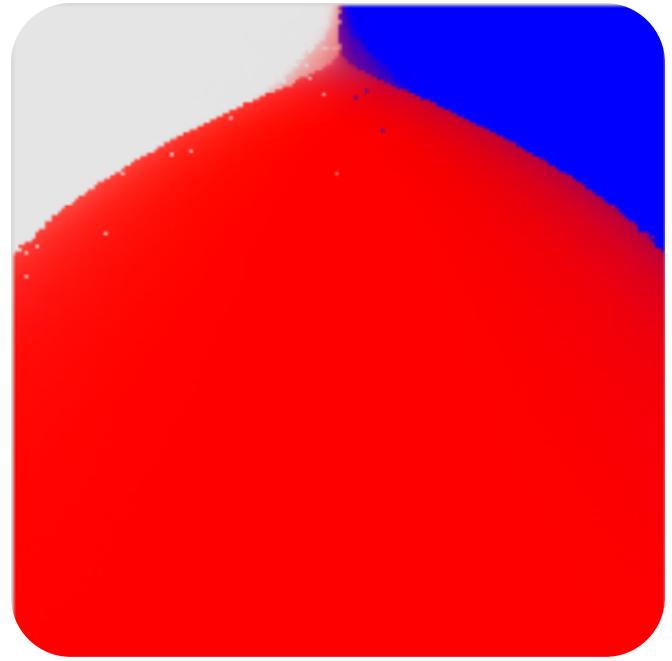
$$L_1 = \sum_i \int_{0 \leq x_i \leq 1} I_i (x_i - \text{ReLU}(\|W_i\|^2 x_i + b_i))^2 + \sum_{i \neq j} \int_{0 \leq x_i \leq 1} I_j \text{ReLU}(W_j \cdot W_i x_i + b_j)^2$$

If we focus on the case $x_i = 1$, we get something which looks even more analogous to the linear case:

$$= \sum_i I_i (1 - \text{ReLU}(\|W_i\|^2 + b_i))^2 + \sum_{i \neq j} I_j \text{ReLU}(W_j \cdot W_i + b_j)^2$$

Feature benefit is similar to before. Note that ReLU never makes things worse, and that the bias can help when the model doesn't represent a feature by taking on the expected value.

Interference is similar to before but ReLU means that negative interference, or interference where a negative bias pushes it below zero, is "free" in the 1-sparse case.

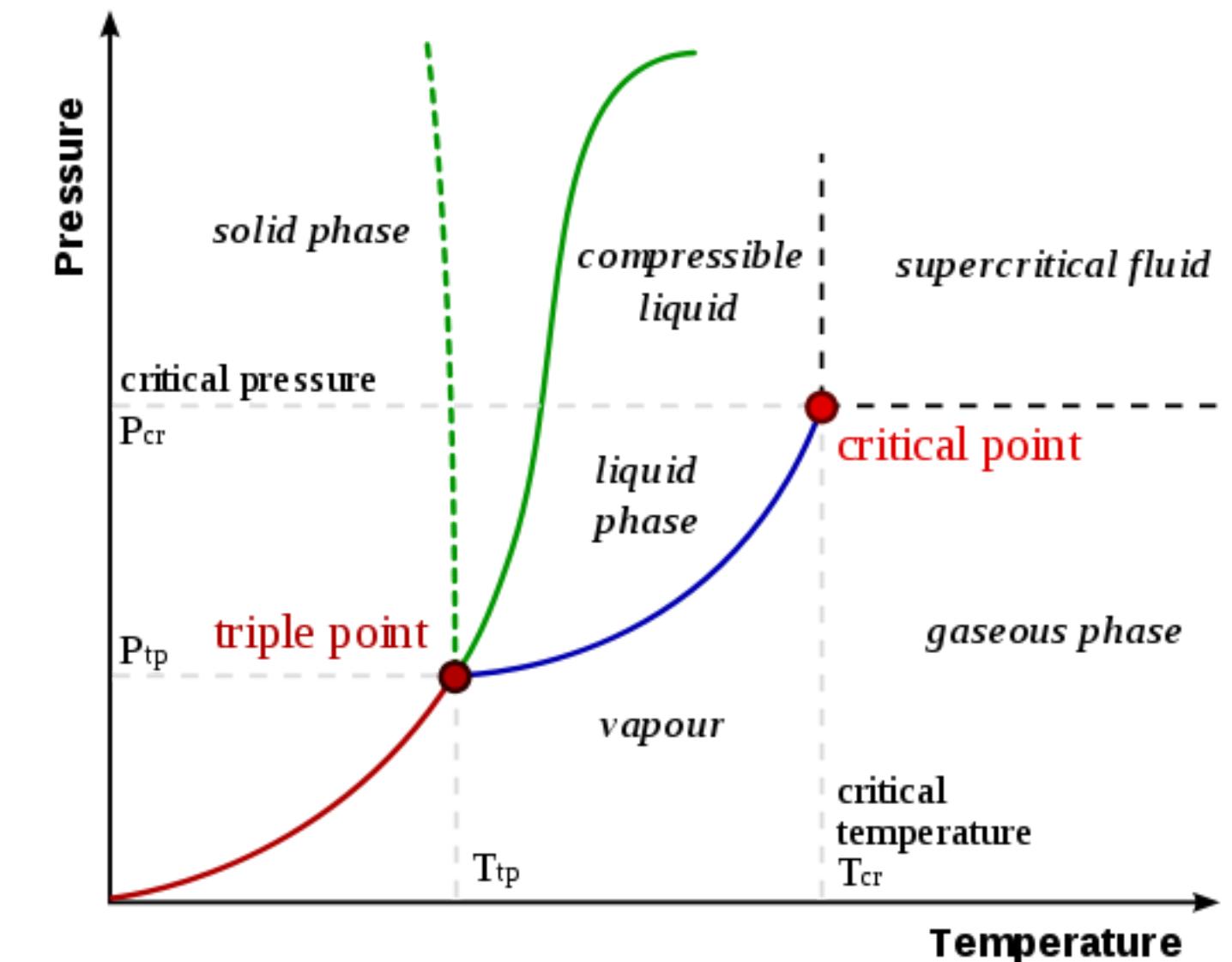


Section 3: Superposition as a Phase Change

Section 3: Superposition as a Phase Change

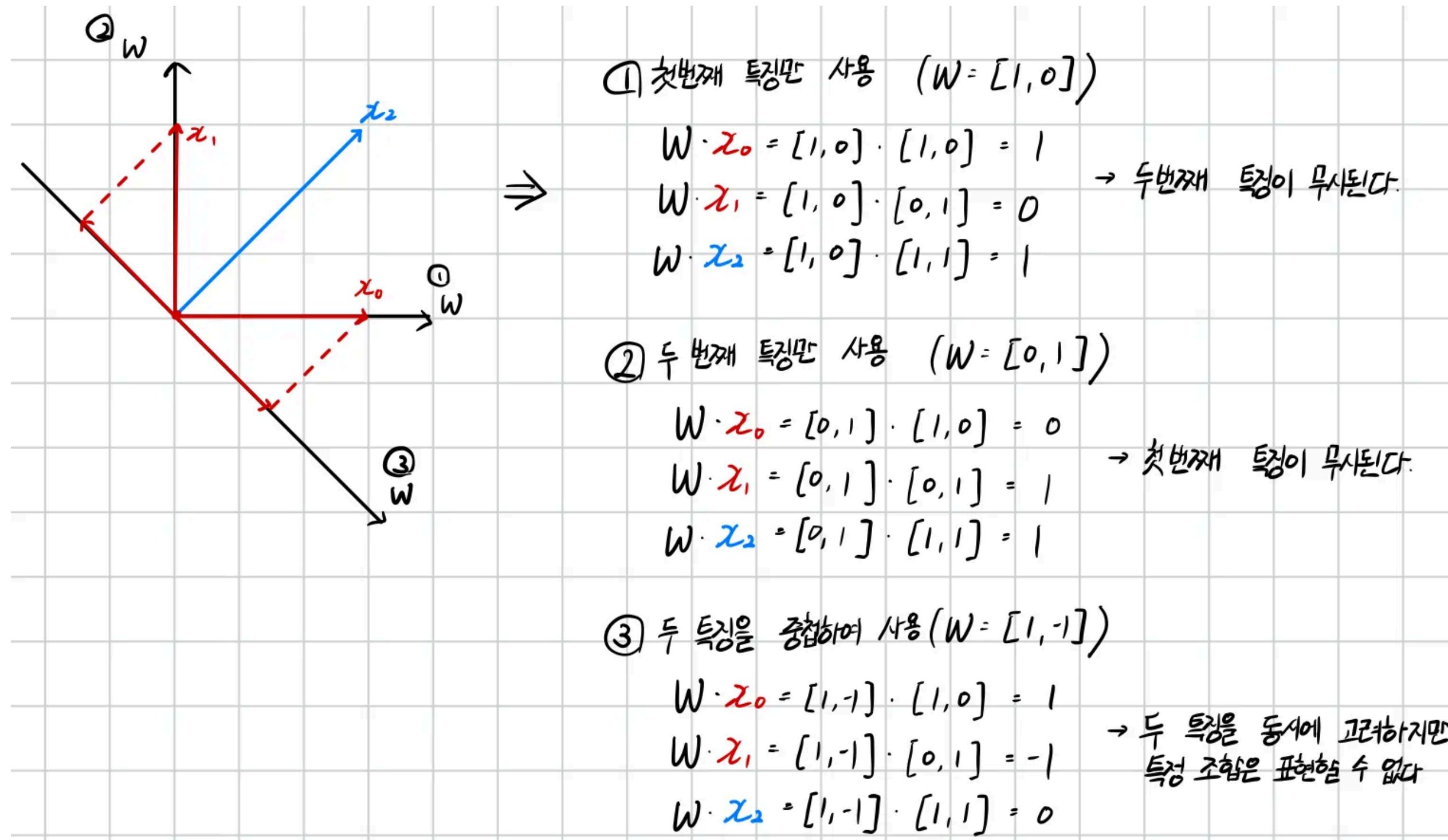
orthogonal \rightarrow antipodal \rightarrow superposition
아마도 어떤 형태의 위상 변화가 있을 것이다.

이걸 더 잘 이해하는 한 가지 방법은 물리학에서 "위상 도표" 같은 것이 있는지 탐구하여 특정 특징이 어느 이러한 영역에 있을 것으로 예상되는지를 파악하는 것이다.



Section 3: Superposition as a Phase Change

2차원의 feature를 1차원에 embedding 하는 상황, 임베딩이 아주 간단한 projection이라고 가정해보자.



-> ‘antipodal’
[1,0]과 [0,1]이 반대방향으로
embedding 된다

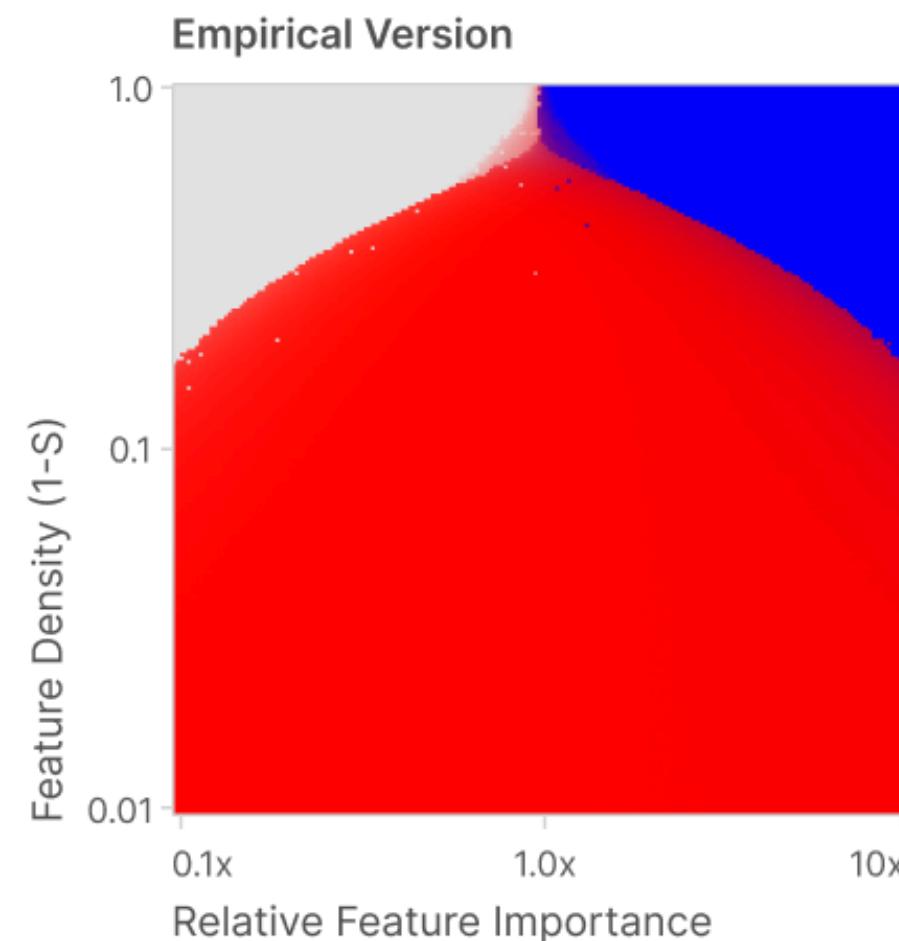
Section 3: Superposition as a Phase Change

Sparsity-Relative Importance Phase Diagram ($n=2, m=1$)

What happens to an “extra feature” if the model can’t give each feature a dimension? There are three possibilities, depending on feature sparsity and the extra feature’s importance relative to other features:

- Extra Feature is Not Represented
- Extra Feature Gets Dedicated Dimension
- Extra Feature is Stored In Superposition

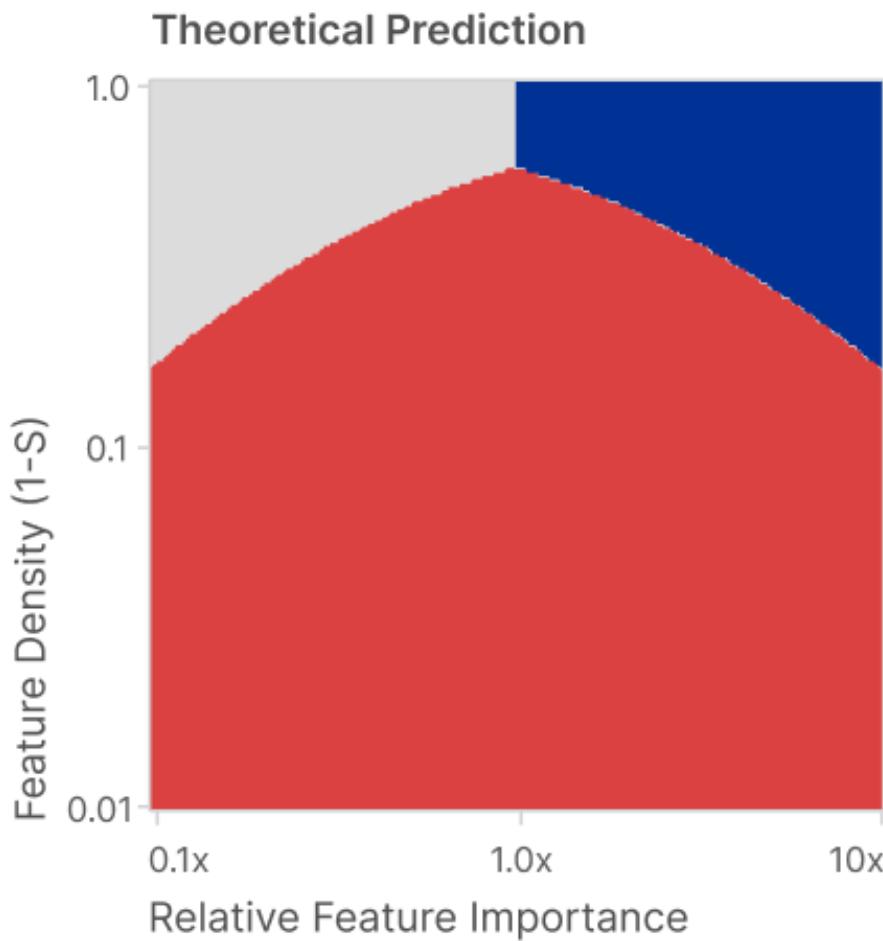
We can both study this empirically and build a theoretical model:



Each configuration is colored by the norm and superposition of the extra feature.

$$\sum_j (\hat{x}_i \cdot x_j)^2$$

A color bar legend for Feature Density (1-S) ranging from 0 to 1.0. It shows a gradient from blue (0) through green and yellow to red (≥ 1). The label $\|W_i\|$ is also present near the red end of the bar.



Not Represented
(Extra Feature is 0)

$$W = [1 \ 0]$$
$$W \perp [0 \ 1]$$

Dedicated Dimension
(Other Not Represented)

$$W = [0 \ 1]$$
$$W \perp [1 \ 0]$$

Superposition
(Antipodal Pair)

$$W = [1 \ -1]$$
$$W \perp [1 \ 1]$$

이번에는 학습을 있다고 가정해보자.

위에서부터 아래로 본다 (Sparsity가 낮았다가 높아지는 과정)

처음에는 중요한 특징은 표현되고, 덜 중요한 특징은 표현되지 않는다.

Superposition은 두 특징 사이에서 나타나다가, 결국에는 모든 특징을 표현하게 된다.

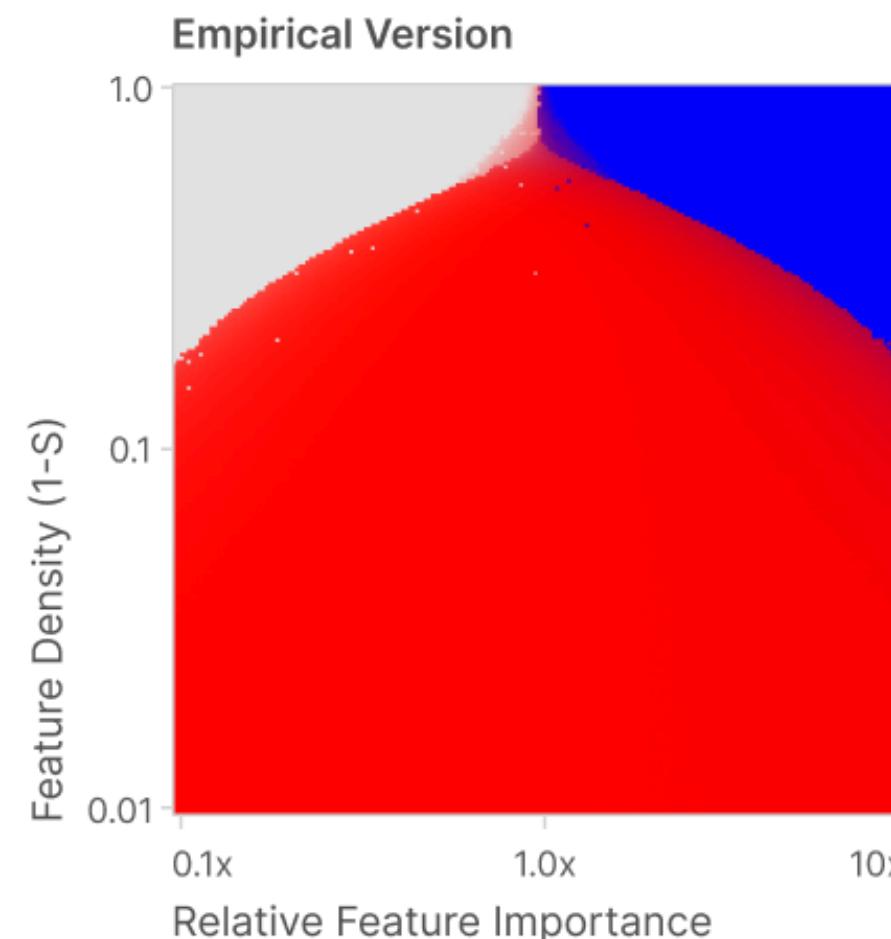
Section 3: Superposition as a Phase Change

Sparsity-Relative Importance Phase Diagram ($n=2, m=1$)

What happens to an “extra feature” if the model can’t give each feature a dimension? There are three possibilities, depending on feature sparsity and the extra feature’s importance relative to other features:

- Extra Feature is Not Represented
- Extra Feature Gets Dedicated Dimension
- Extra Feature is Stored In Superposition

We can both study this empirically and build a theoretical model:



Each configuration is colored by the norm and superposition of the extra feature.

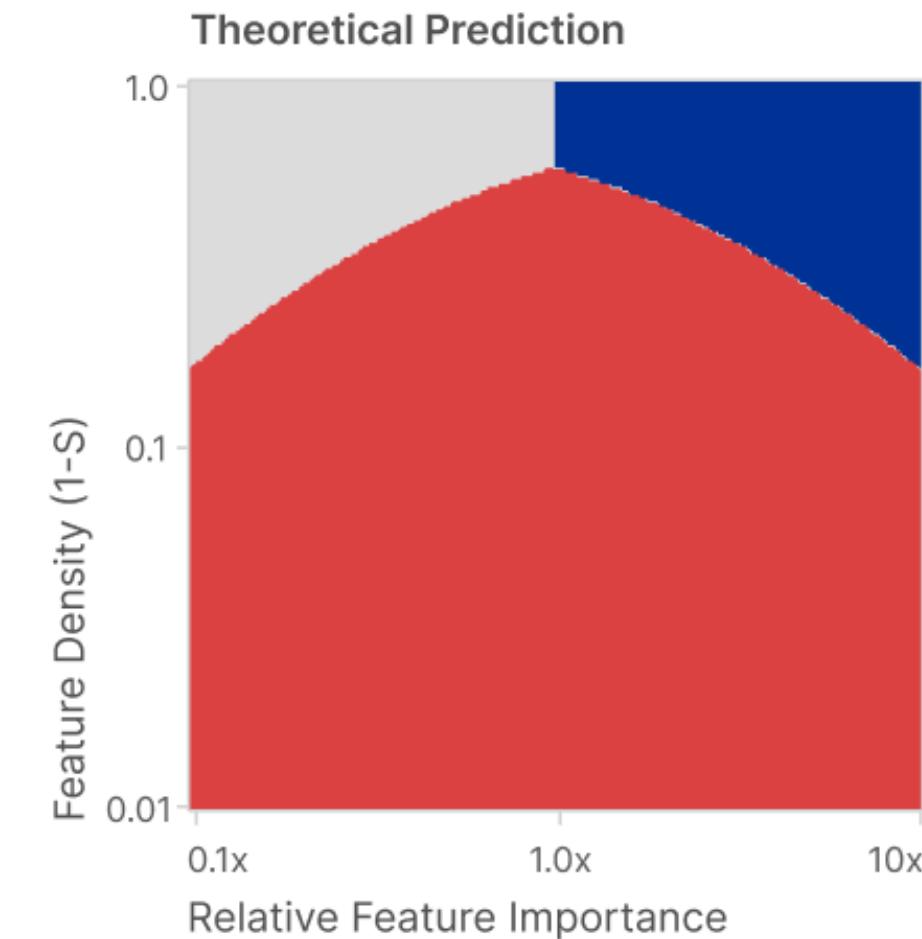
$$\sum_j (\hat{x}_i \cdot x_j)^2$$

A color scale legend for Feature Density (1-S), ranging from 0 (blue) to ≥ 1 (red). The label $||W_i||$ is also present.

≥ 1

$||W_i||$

0



Not Represented
(Extra Feature is 0)

$$W = [1 \ 0]$$
$$W \perp [0 \ 1]$$

Dedicated Dimension
(Other Not Represented)

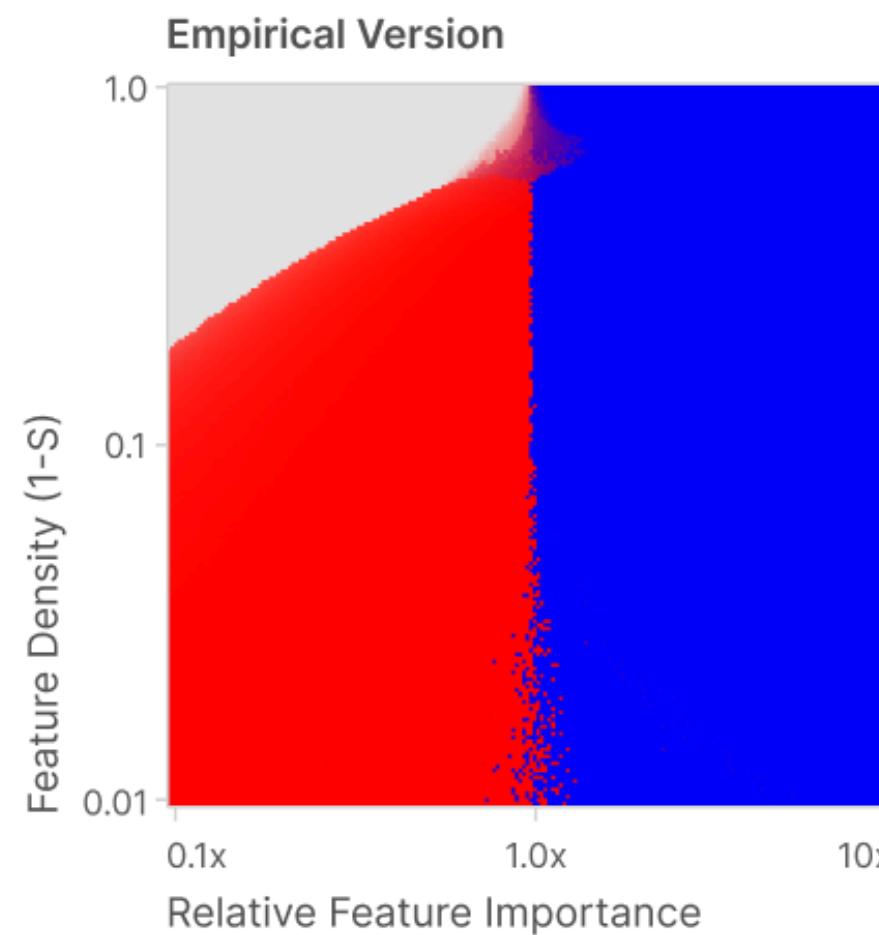
$$W = [0 \ 1]$$
$$W \perp [1 \ 0]$$

Superposition
(Antipodal Pair)

$$W = [1 \ -1]$$
$$W \perp [1 \ 1]$$

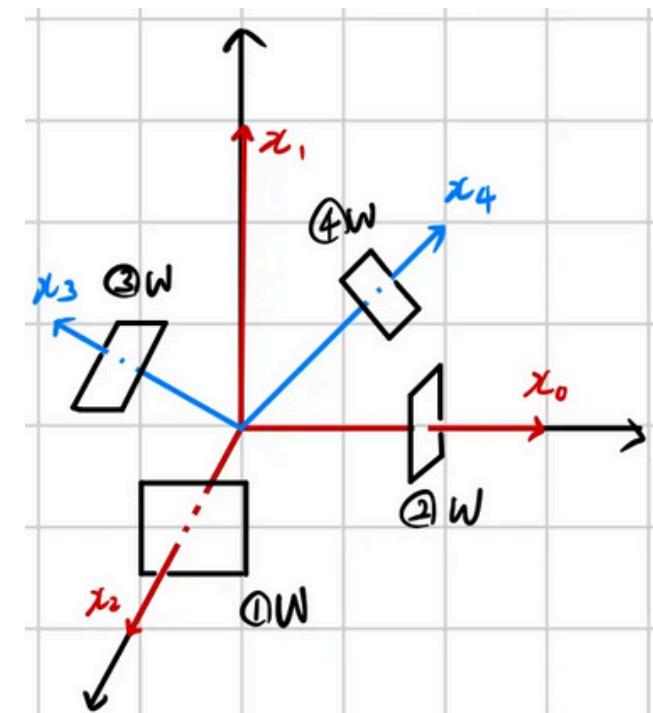
Section 3: Superposition as a Phase Change

Sparsity-Relative Importance Phase Diagram ($n=3, m=2$)



Each configuration is colored by the norm and superposition of the extra feature.

$$\sum_j (\hat{x}_i \cdot x_j)^2$$



① 추가 특징만 무시 ($w \perp [0, 0, 1]$)

$$w \cdot x_0 = [1, 0] \quad w \cdot w_3 = [0, 1]$$

$$w \cdot x_1 = [0, 1] \quad w \cdot w_4 = [1, 1]$$

$$w \cdot x_2 = 0$$

② 다른 특징중 1개 무시 ($w \perp [1, 0, 0]$ OR $w \perp [0, 1, 0]$)

$$w \cdot x_0 = 0 \quad w \cdot w_3 = [1, 1]$$

$$w \cdot x_1 = [0, 1] \quad w \cdot w_4 = [0, 1]$$

$$w \cdot x_2 = [1, 0]$$

③ 추가 특징을 다른 특징과 반대방향으로 중첩 ($w \perp [0, 1, 1]$)

$$w \cdot x_0 = [1, 0] \quad w \cdot w_3 = 0 \quad w = [0, 1, -1]$$

$$w \cdot x_1 = [0, 1] \quad w \cdot w_4 = [1, 1]$$

$$w \cdot x_2 = [0, -1]$$

dedicated Dim

④ 다른 두 특징을 중첩하고 추가특징에 전용차원 부여 ($w \perp [1, 1, 0]$)

$$w \cdot x_0 = [0, -1] \quad w \cdot w_3 = [1, 1]$$

$$w \cdot x_1 = [0, 1] \quad w \cdot w_4 = 0$$

$$w \cdot x_2 = [1, 0]$$

Section 3: Superposition as a Phase Change

$$W \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \quad x \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

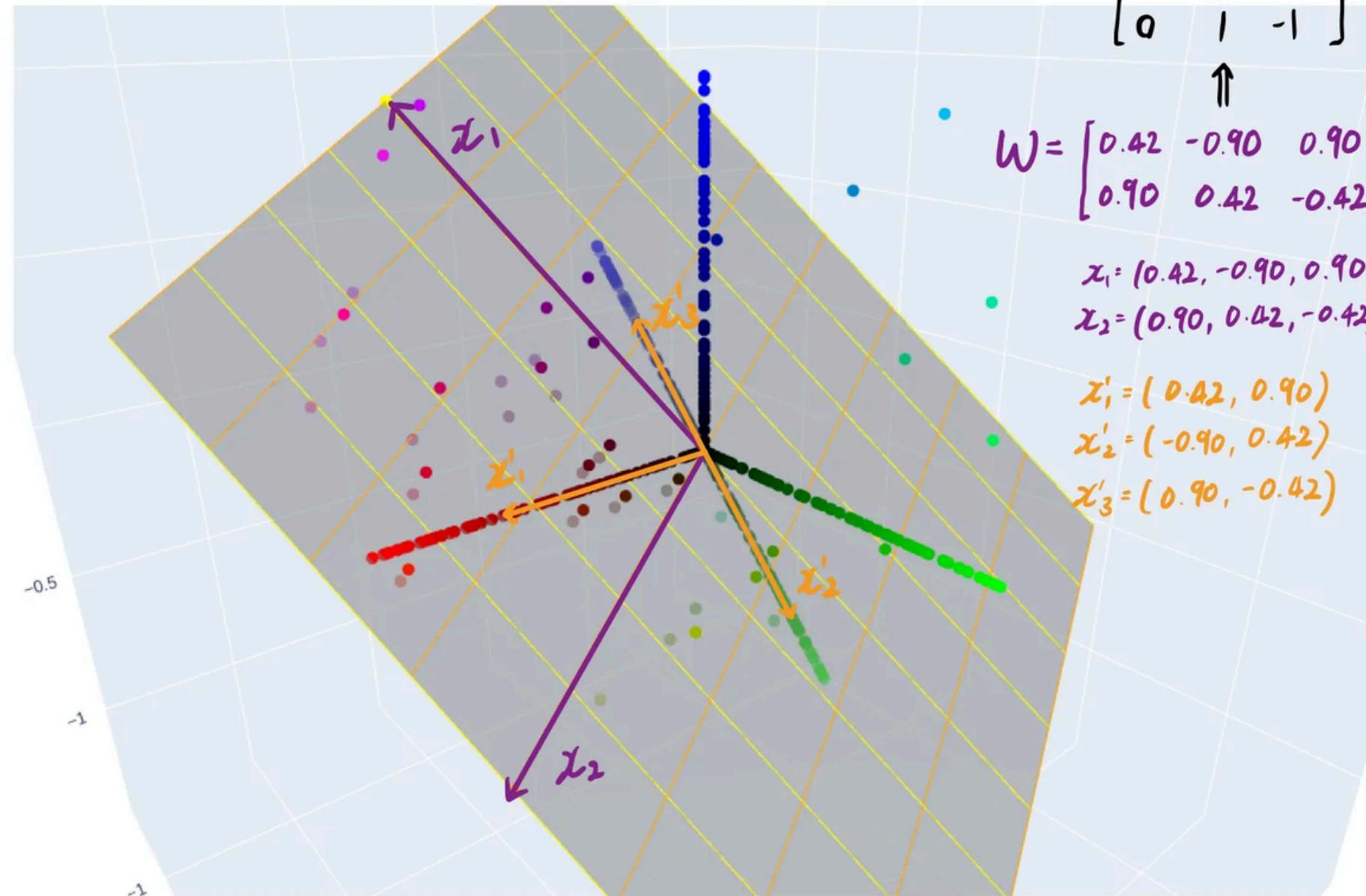
W 는 3차원 좌표를 임의의 2차원 평면으로 projection 시키는 선형변환

W 의 column vec 2개는 3차원에서 표현한 2차원 basis

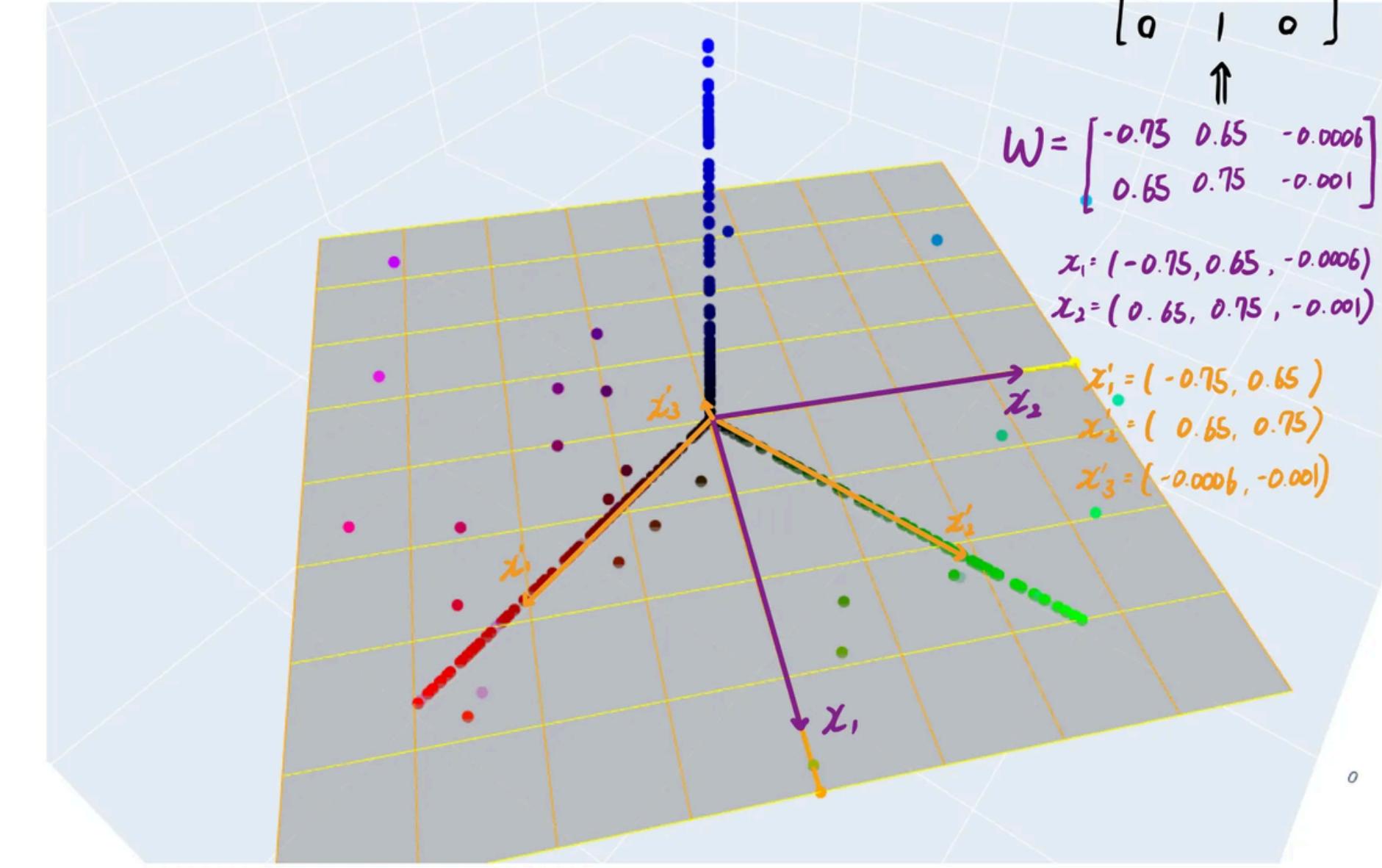
W 의 row vec 3개는 2차원에서 표현한 3차원 basis

Section 3: Superposition as a Phase Change

ReLU Model Projection (1-S=0.09999999999999998)



Linear Model Projection (1-S=0.0999999999999998)



평면은 x_1 과 x_2 를 span한 평면 (x_1 과 x_2 는 평면의 basis)

x'_1 은 x'_2 , x'_3 와 모두 orthogonal

x'_2 와 x'_3 는 서로 antipodal pair

평면은 x_1 과 x_2 를 span한 평면 (x_1 과 x_2 는 basis)

x'_1 은 x'_2 와 orthogonal

x'_3 는 거의 활성화되지 않음

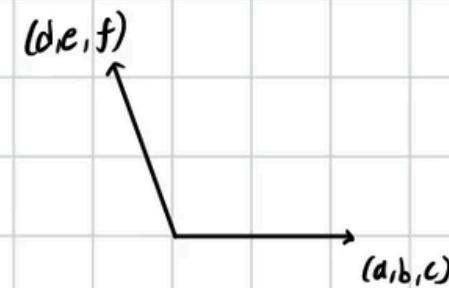
Section 3: Superposition as a Phase Change

$$x' = \text{ReLU}(W^T W z)$$

$$\text{Loss} = \sum_x (z - z')^2$$

$$\text{ReLU} \left(\begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \right) = \text{ReLU} \left(\begin{bmatrix} a^2+d^2 & ab+de & ac+df \\ b^2+e^2 & bc+ef & \cdot \\ c^2+f^2 & \cdot & \cdot \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \right)$$

ReLU
-가 되어도 0이 된다



I 가 되지 않아도 괜찮아진다 $\rightarrow W^T W$ linear dependent
 W 의 열벡터가 dependent

$$W = \begin{bmatrix} 1 & \alpha & \beta \\ 0 & 1 & r \end{bmatrix}$$

or

$$\begin{bmatrix} 1 & \alpha & \beta \\ 0 & r & 1 \end{bmatrix}$$

or

$$\begin{bmatrix} \alpha & 1 & \beta \\ 0 & r & 1 \end{bmatrix}$$

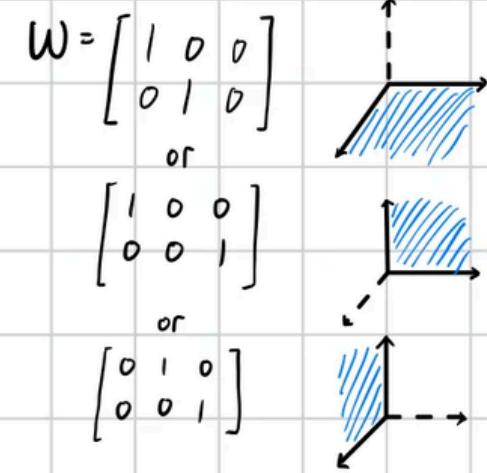
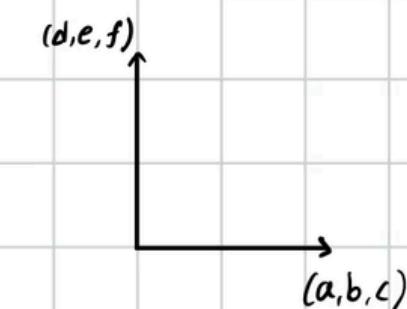
$$x' = W^T W z$$

$$\text{Loss} = \sum_x (z - z')^2$$

$$\begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} a^2+d^2 & ab+de & ac+df \\ b^2+e^2 & bc+ef & \cdot \\ c^2+f^2 & \cdot & \cdot \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$



I 가 되도록 바꿨다 $\rightarrow W^T W$ linear independent
 W 의 열벡터가 independent

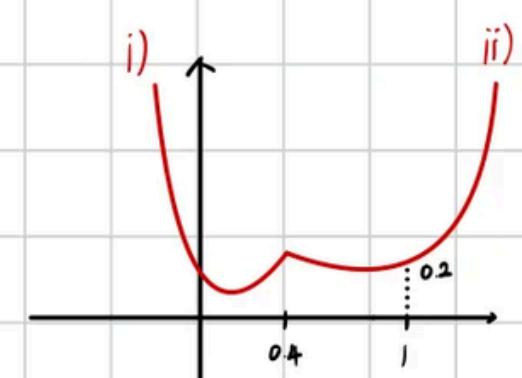
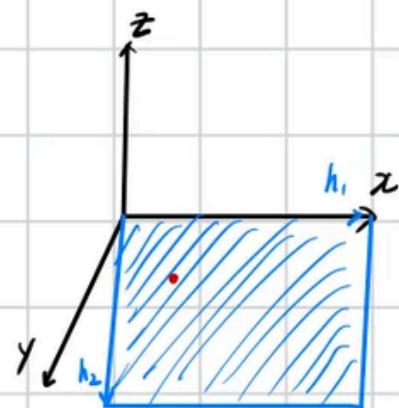


Section 3: Superposition as a Phase Change

$$x' = \text{ReLU}(W^T W x)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ K \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.4 \\ K \end{bmatrix} =$$

$$\begin{bmatrix} W^T \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.4-K \\ K \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.4-K \\ -0.4+K \end{bmatrix}$$



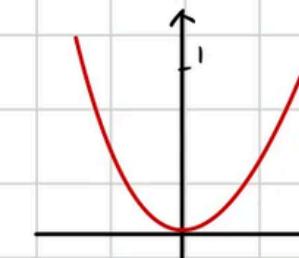
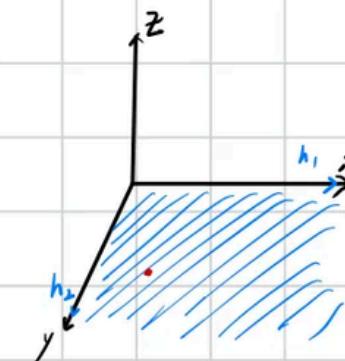
i) $L = 5K^2 - 1.6K + 0.16 \quad (K < 0.4)$
 ii) $L = K^2 - 1.6K + 0.8 \quad (K > 0.4)$

원래 좌표의 크가 늘어나도
 Loss는 상대적으로 천천히 증가

$$x' = W^T W x$$

$$\begin{bmatrix} W^T \\ W \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & K \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.4 \\ K \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0 \end{bmatrix}$$

$$L = (x - x')^2 = K^2$$



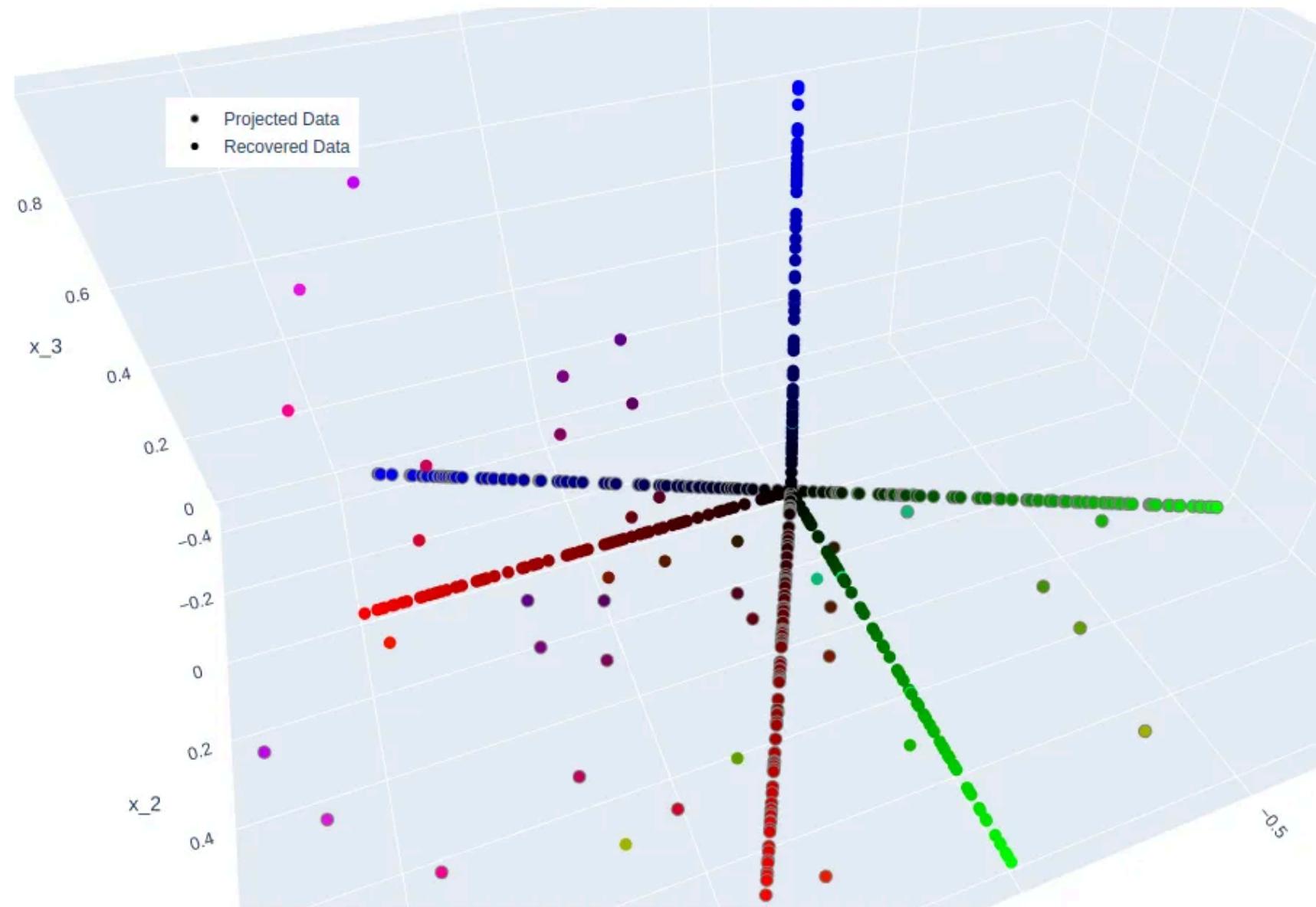
원래 좌표의 크가 늘어날수록
 Loss는 제곱으로 증가

Section 3: Superposition as a Phase Change

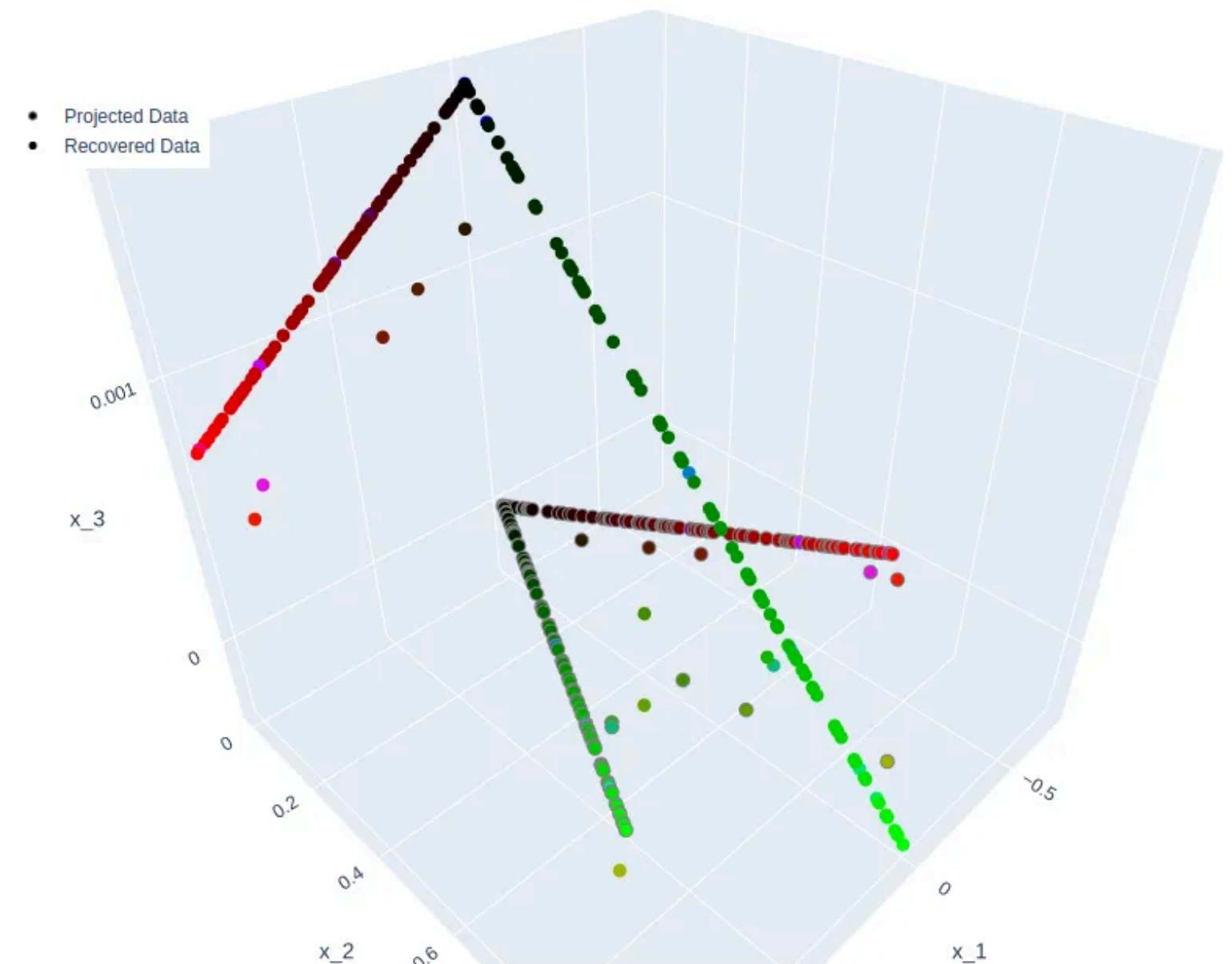
W^T 를 적용해서 projection된 2차원 데이터로부터 3차원 데이터를 복원 -> ReLU모델은 완벽하게 복원하는 반면, Linear 모델은 잘 복원시키지 못함

- 회색 테두리가 있는 점은 projection된 점들
- 테두리가 없는 점은 복원되는 점들

Linear Model Projection (1-S=0.0999999999999998)

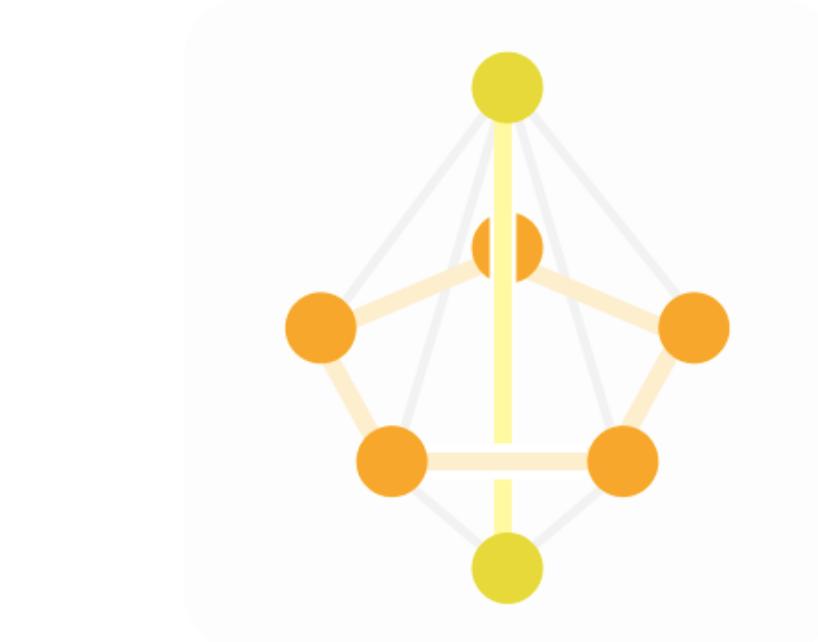


Linear Model Projection (1-S=0.0999999999999998)



Section 3: Superposition as a Phase Change > Summary of this section

**Superposition이 모델이 추가 특성(extra feature)을 나타낼 수 있게 하고,
추가 특성의 수가 희소성(sparsity)이 증가함에 따라 증가한다.**



Section 4: The Geometry of Superposition

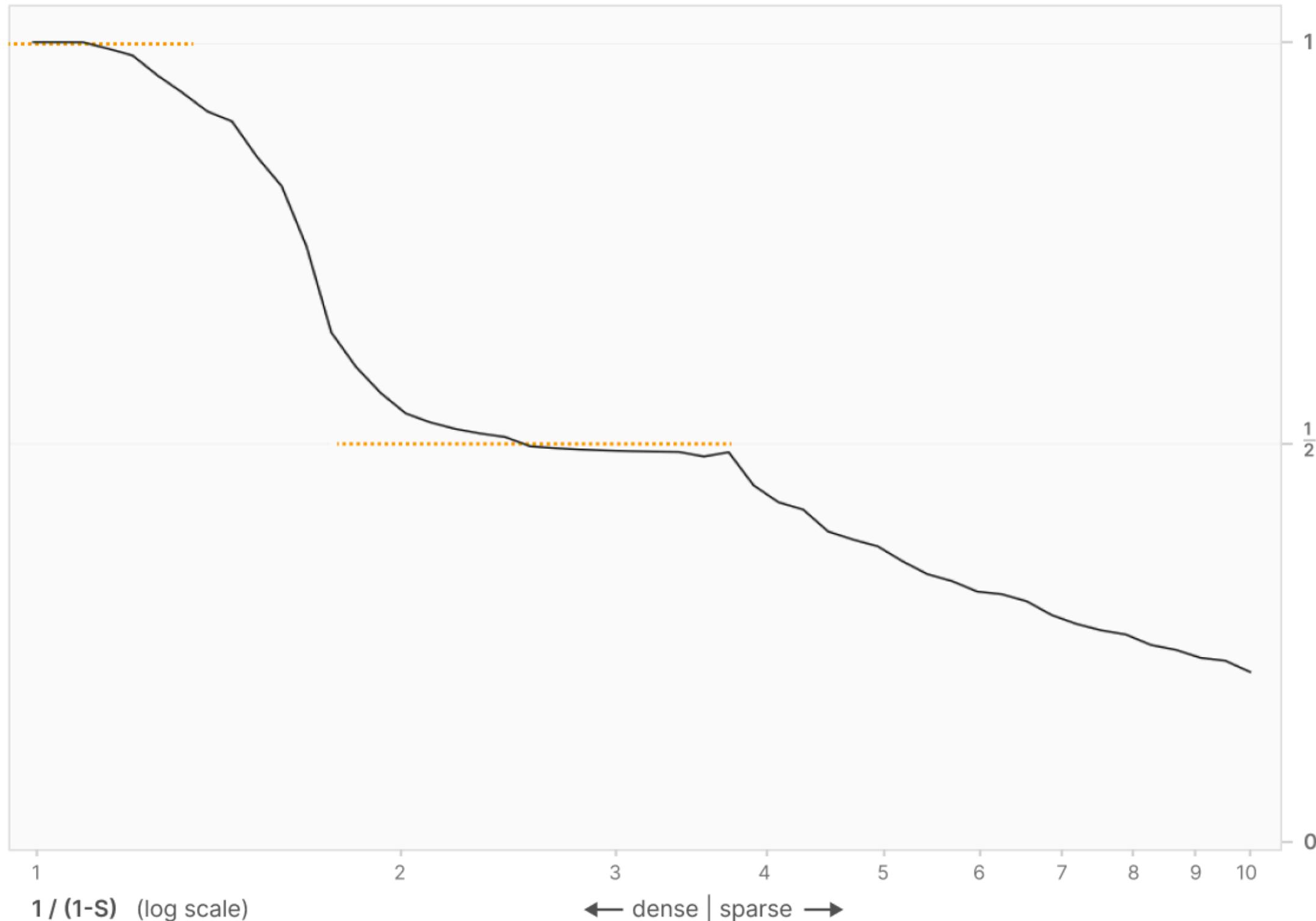
Section 4: The Geometry of Superposition > Uniform Superposition

in more detail, discovering an unexpected geometric story: features seem to organize themselves into geometric structures such as pentagons and tetrahedrons! In some ways, the structure described in this section seems "too elegant to be true" and we think there's a good chance it's at least partly idiosyncratic to the toy model we're investigating. But it seems worth investigating because if anything about this generalizes to real models, it may give us a lot of leverage in understanding their representations.

어떤 면에서는, 이 섹션에서 설명하는 구조가 "너무 우아해서 사실일 리가 없다"고 느껴지며, 우리가 조사하고 있는 장난감 모델에 적어도 부분적으로 특이할 가능성이 높다고 생각합니다. 그러나 이 부분이 실제 모델로 일반화된다면 이들의 representation을 이해하는 데 많은 도움이 될 수 있기 때문에 조사할 가치가 있어 보입니다.

Section 4: The Geometry of Superposition > Uniform Superposition

Feature가 identical한, uniform superposition (independent, equally important, equally sparse)



Number of Hidden Dimensions per Embedded Feature

$$m/\|W\|_F^2$$

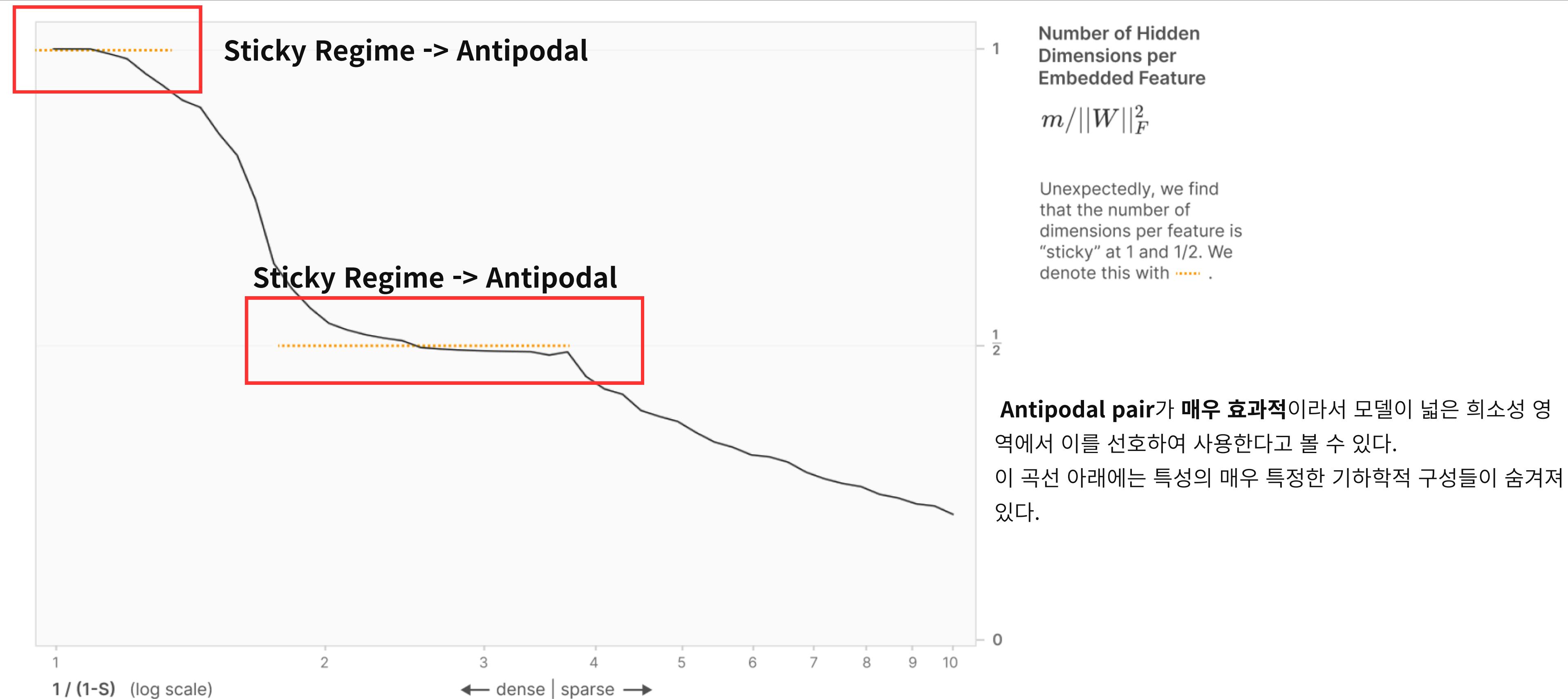
Unexpectedly, we find that the number of dimensions per feature is “sticky” at 1 and $\frac{1}{2}$. We denote this with

embedded feature당 (hidden) dimension

임베딩된 각 특징이 몇 개의 숨겨진 차원을 사용하는지

Frobenius norm은 특징이 표현되었는지 여부

Section 4: The Geometry of Superposition > Uniform Superposition



Section 4: The Geometry of Superposition > Uniform Superposition

Dimensionality

특정 특징이 차지하는 차원의 비율을 나타내는 척도

$$D_i = \frac{\|W_i\|^2}{\sum_j (\hat{W}_i \cdot W_j)^2}$$

$$\|W_i\|^2$$

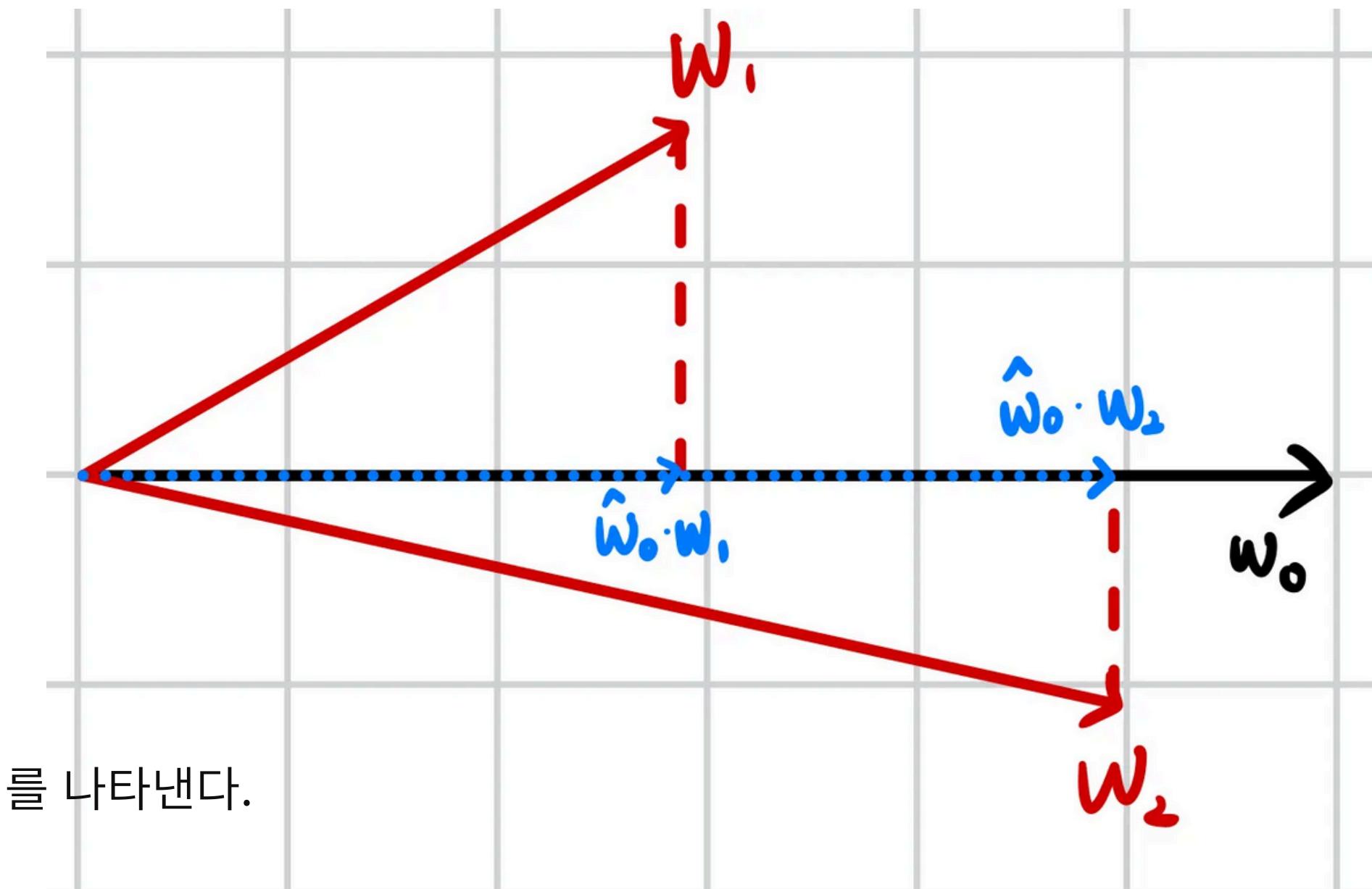
특징 i의 가중치 벡터의 크기

특징이 얼마나 강하게 표현되는지를 나타낸다.

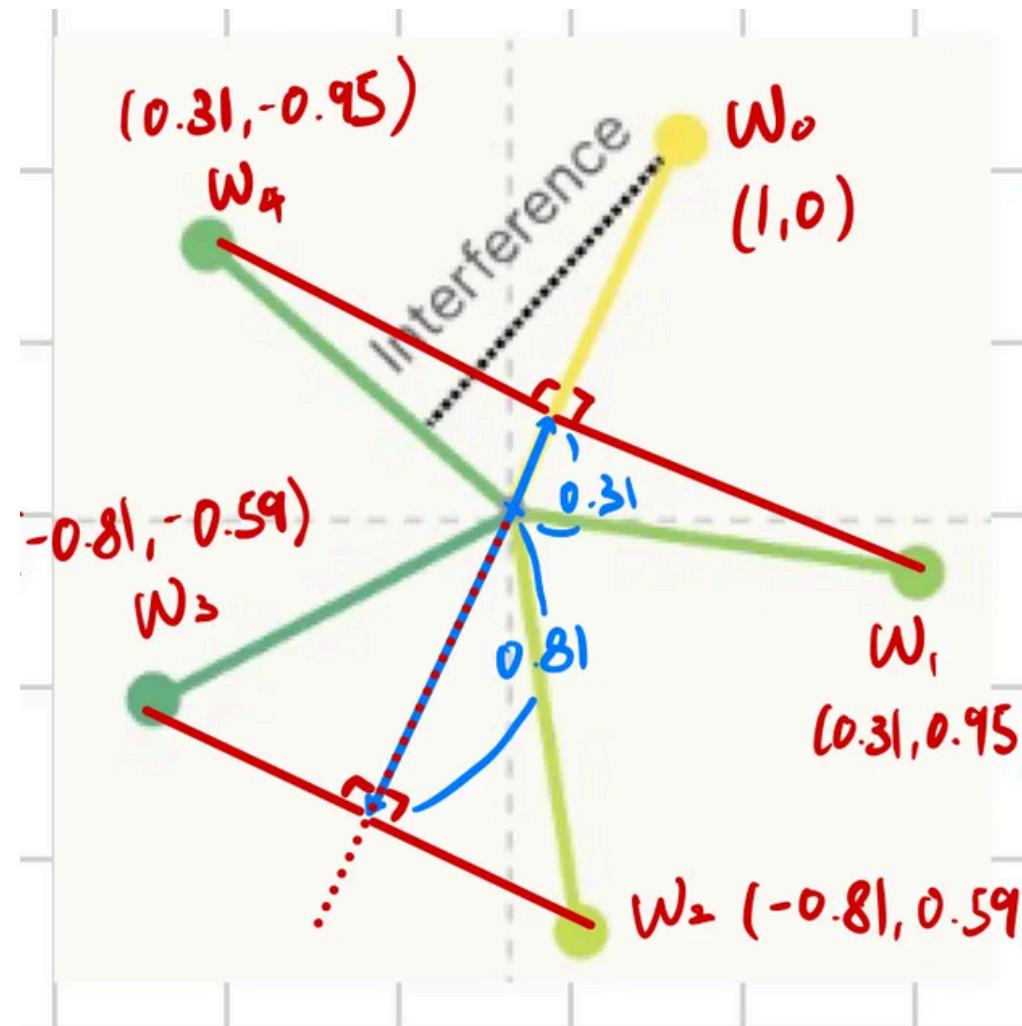
$$\sum_j (\hat{W}_i \cdot W_j)^2$$

모든 가중치 벡터 W_j 와 단위 벡터 W_i 의 내적의 제곱합

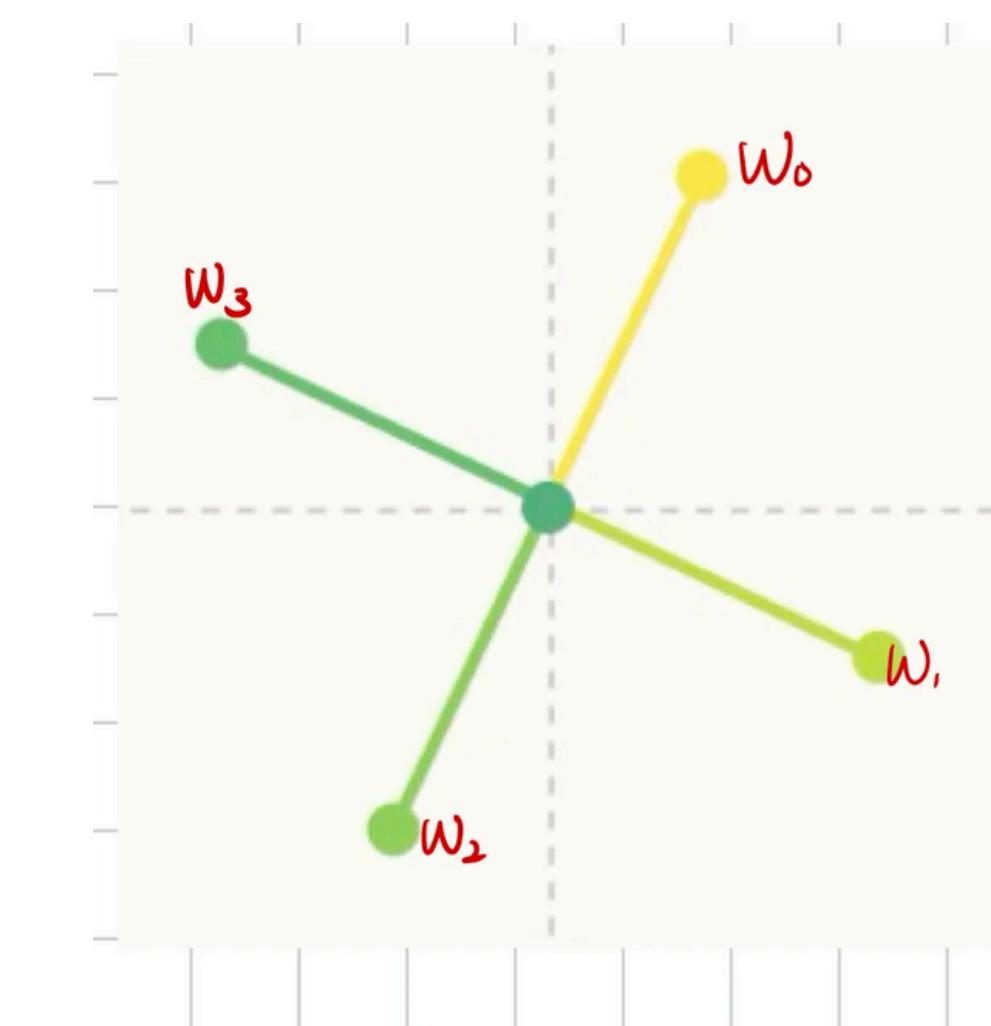
특징 i가 다른 특징들과 얼마나 관련이 있는지를 나타낸다.



Section 4: The Geometry of Superposition > Uniform Superposition



$$\begin{aligned} & \frac{1}{2(0.31)^2 + 2(0.81)^2 + 1} \\ &= \frac{1}{2.4984} \approx 0.4 \\ & 0.4 \times 5 = 2 \end{aligned}$$

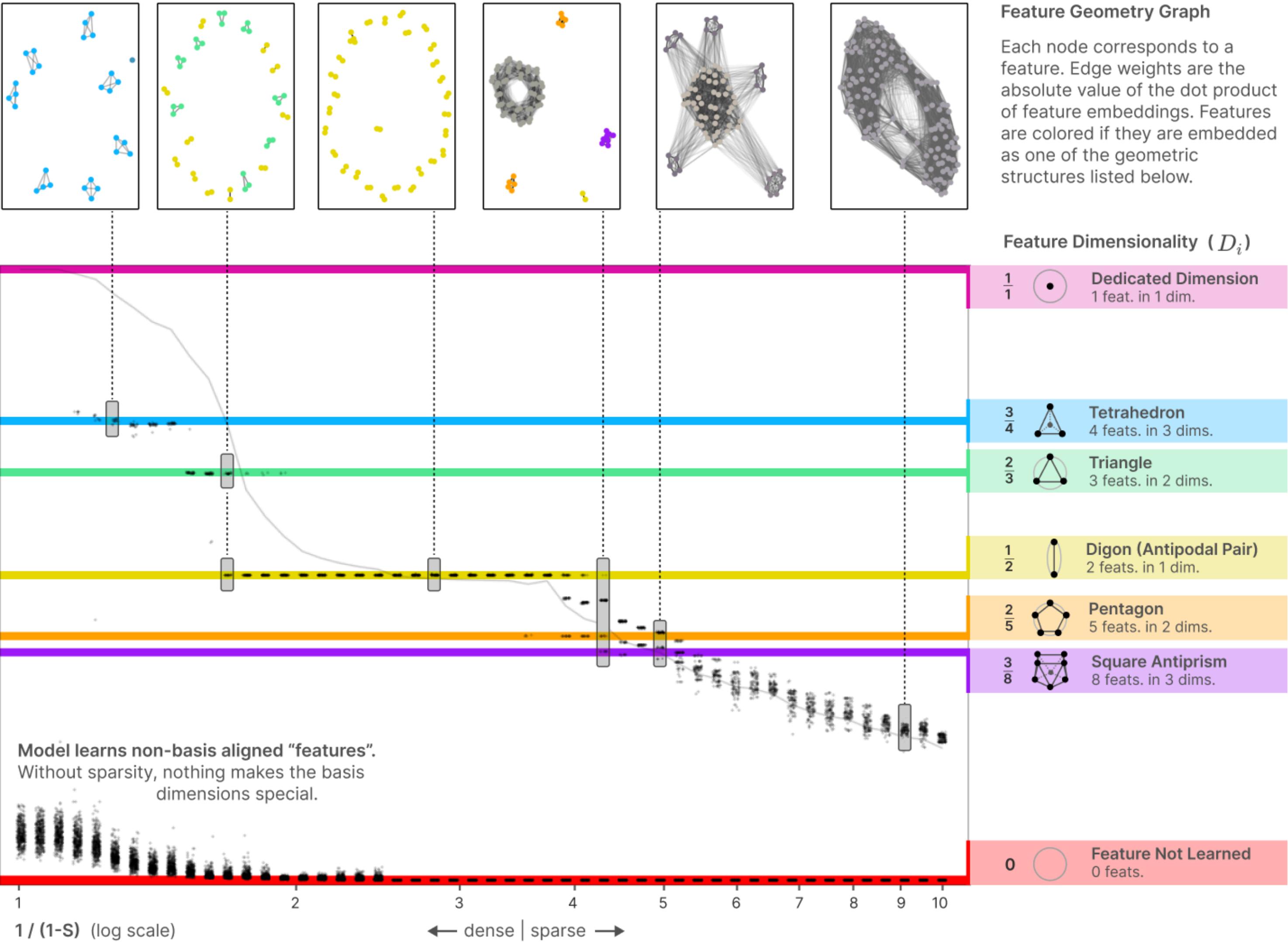


$$\begin{aligned} & w_0 \cdot w_1 = w_0 \cdot w_3 = 0 \\ & \frac{\|w_0\|^2}{(w_0 \cdot w_0) + (w_0 \cdot w_1)} \\ &= \frac{1}{1+1} = \frac{1}{2} \\ & \frac{1}{2} \times 4 + 0 = 2 \end{aligned}$$

Feature vector 1개가 0.4차원을 나타낸다.

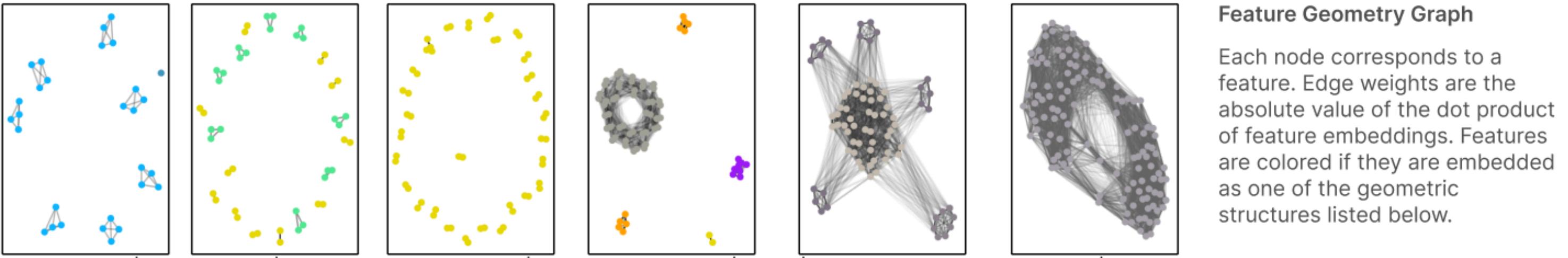
Feature vector 1개가 0.5차원을 나타낸다.

Orthogonal feature 2개일 경우 : Feature vector 1개가 전체 차원 중 1차원을 나타낸다.



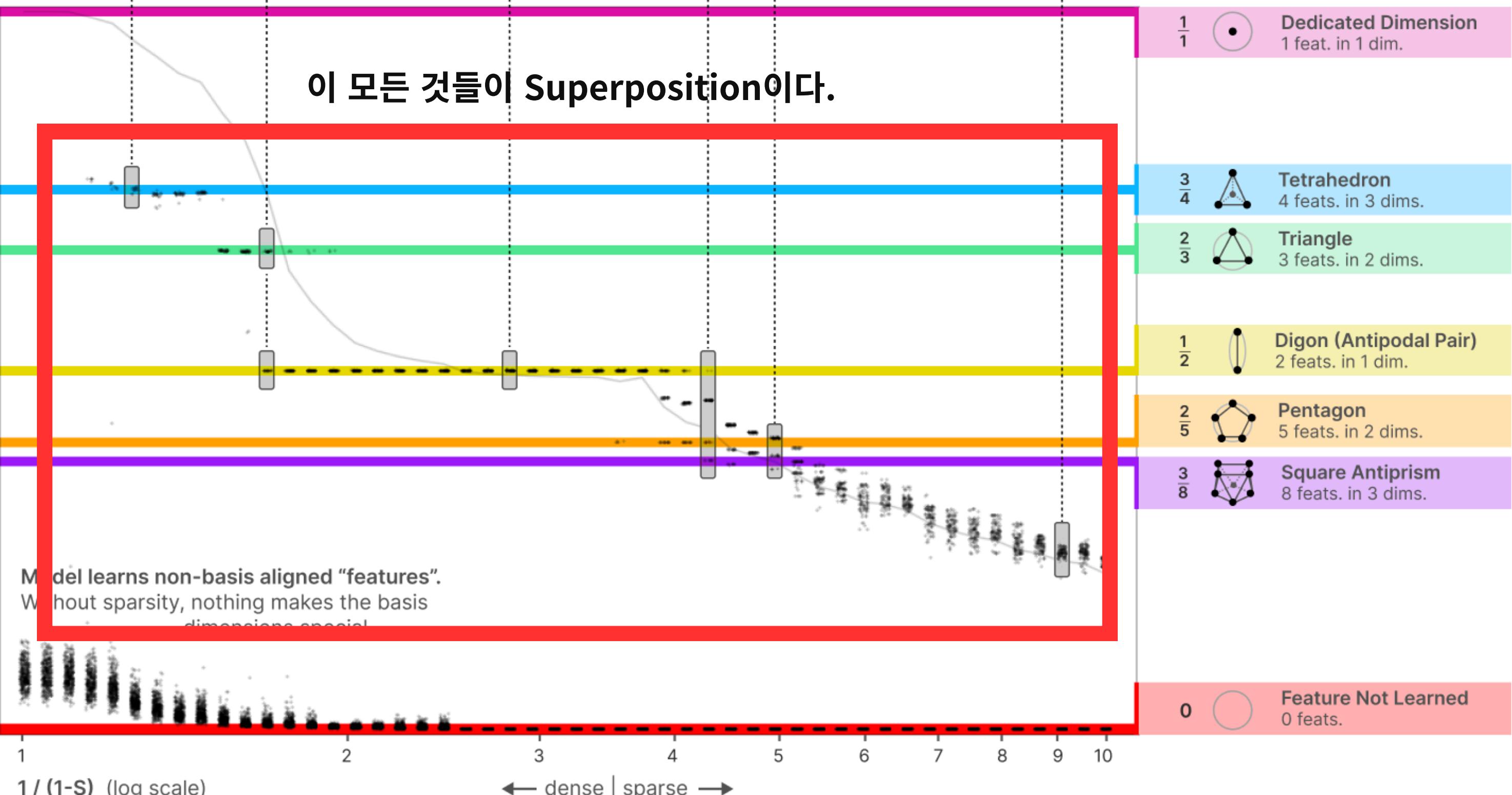
Node : Feature
Edge : |두 feature 내적|
(0인경우 연결 X)

Color : Feature들이 특정 기하학적 구조에 속하면 색상으로 표시

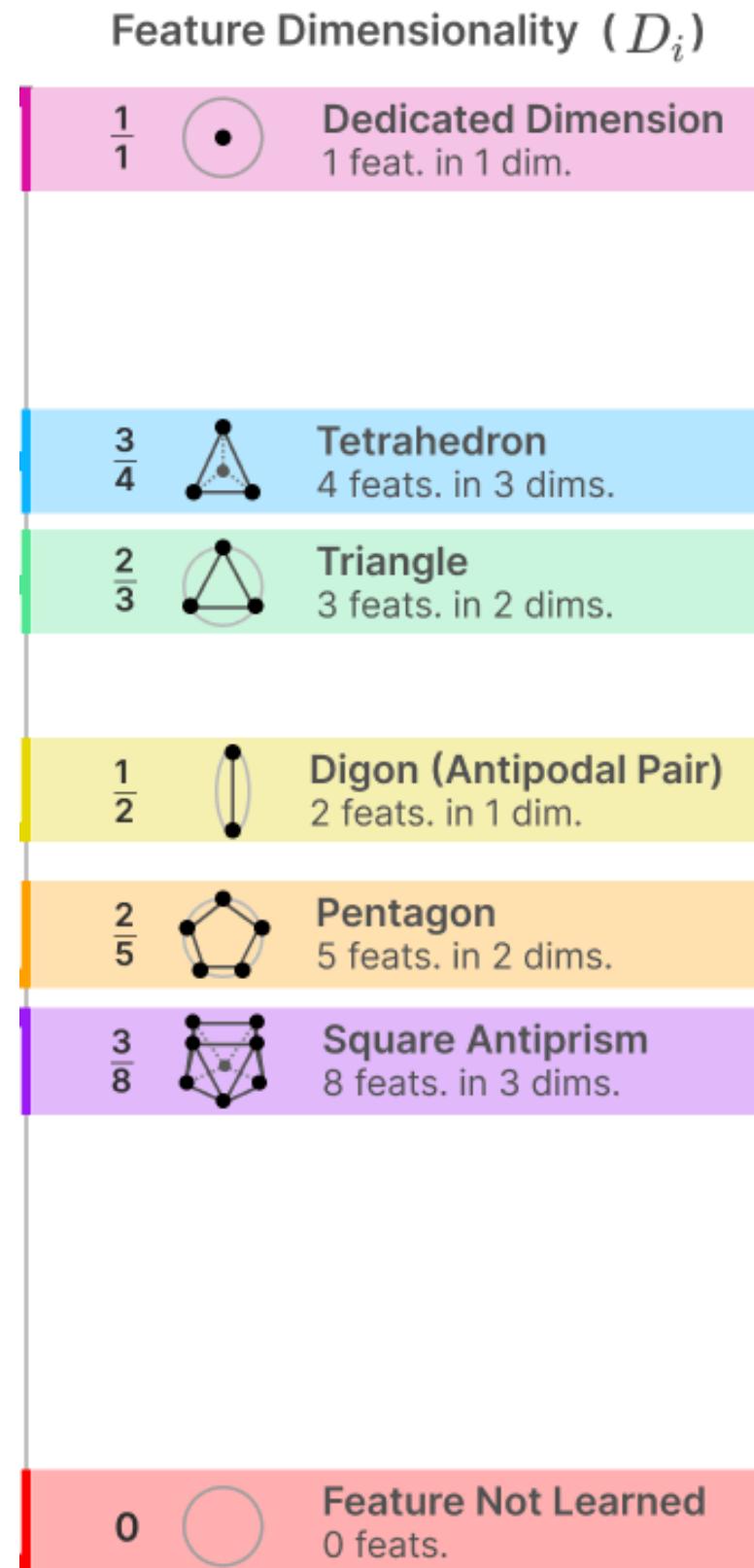


Feature Geometry Graph

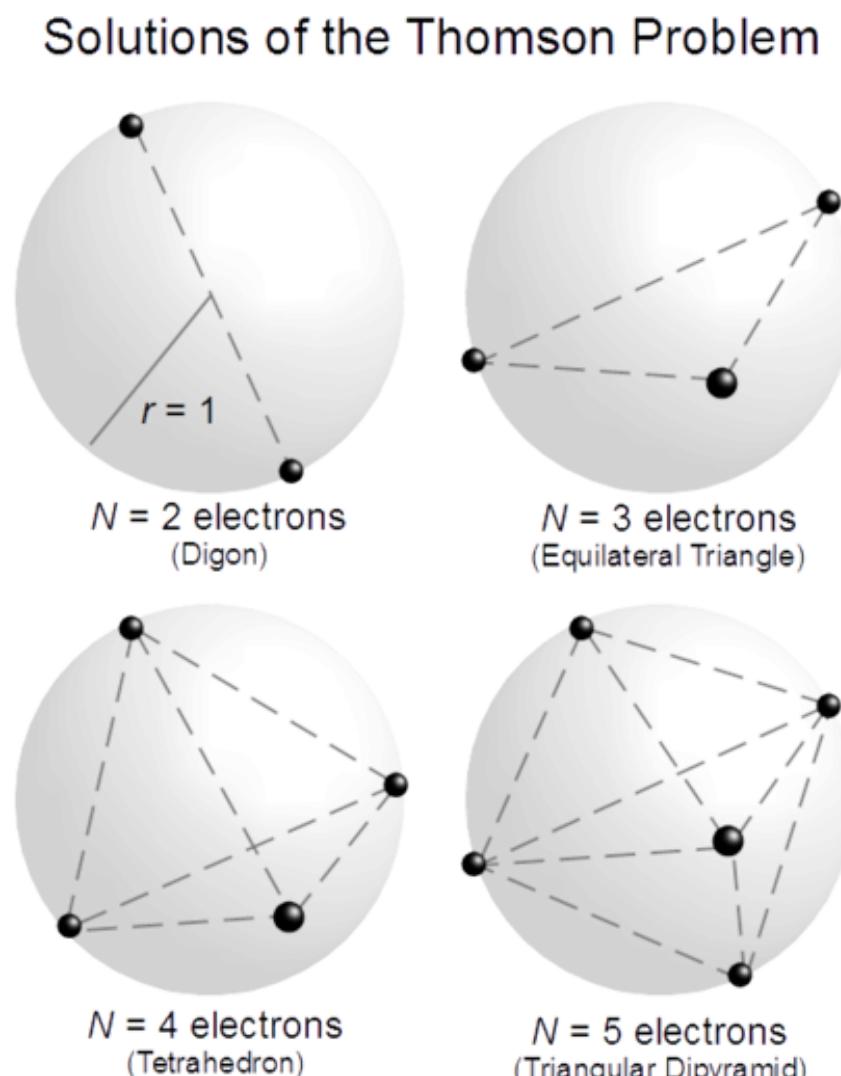
Each node corresponds to a feature. Edge weights are the absolute value of the dot product of feature embeddings. Features are colored if they are embedded as one of the geometric structures listed below.



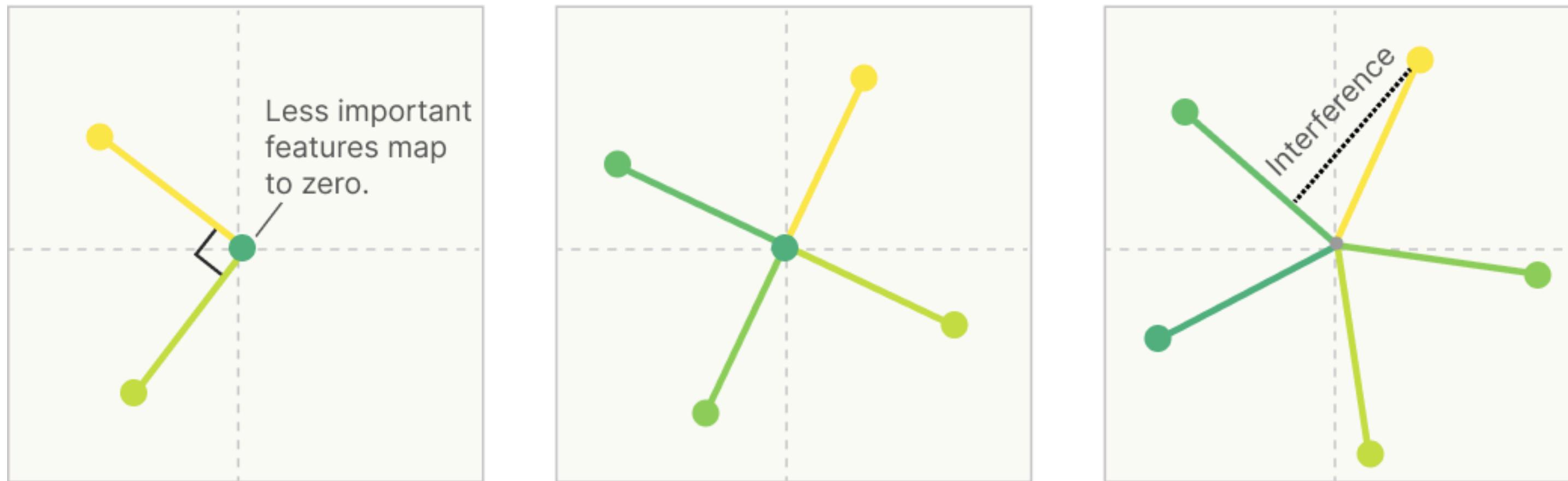
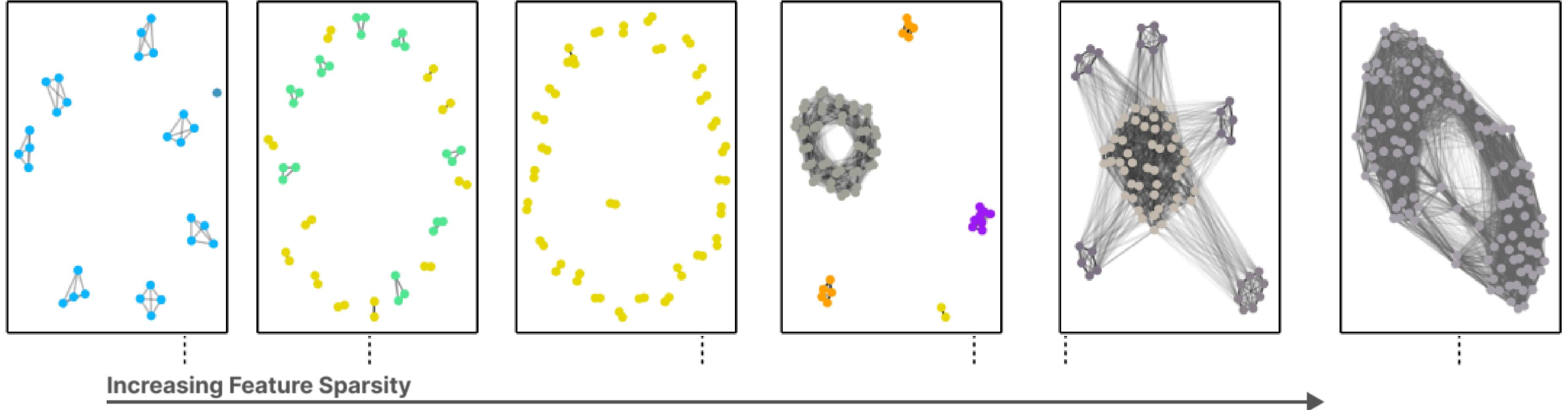
Section 4: The Geometry of Superposition > Uniform Superposition



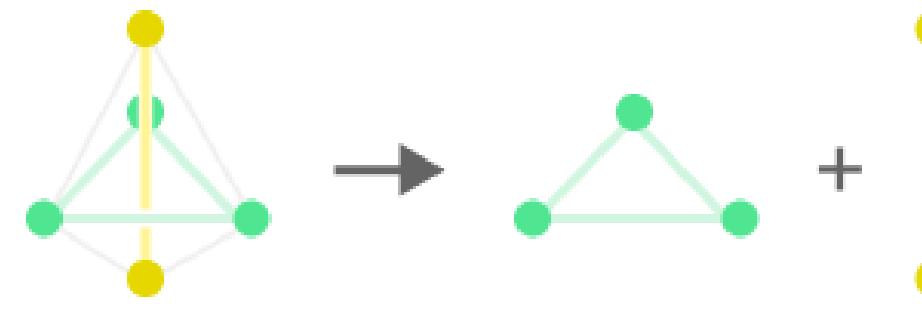
- **1/1 (Dedicated Dimension): 1개의 특징이 1차원을 사용.**
- **3/4 (Tetrahedron): 4개의 특징이 3차원을 사용.**
- **2/3 (Triangle): 3개의 특징이 2차원을 사용.**
- **1/2 (Digon, Antipodal Pair): 2개의 특징이 1차원을 사용.**
- **2/5 (Pentagon): 5개의 특징이 2차원을 사용.**
- **3/8 (Square Antiprism): 8개의 특징이 3차원을 사용.**
- **0 (Feature Not Learned): 특징이 학습되지 않음.**



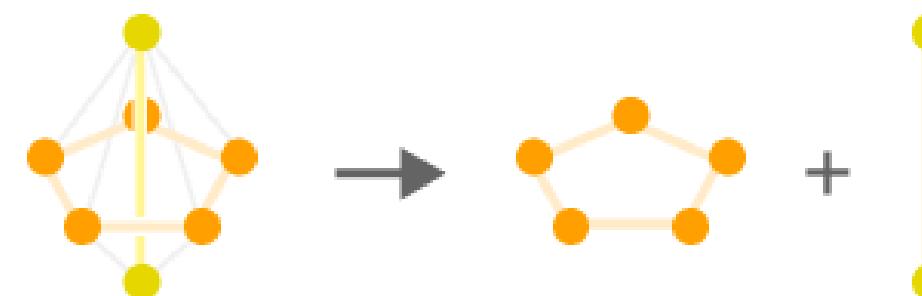
전자들이 단위 구면 위에서 최소 에너지 상태를 가질 수 있는 배치를 찾는 문제



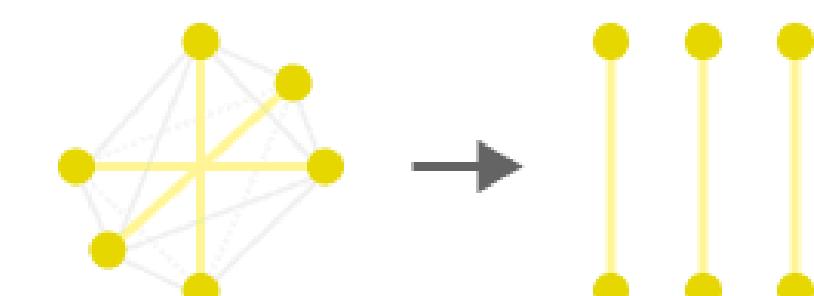
Section 4: The Geometry of Superposition



A triangular bipyramid is the tegum product of a triangle and an antipode. As a result, we observe $3 \times 2/3$ features and $2 \times 1/2$ features, rather than $6 \times 3/5$ features.



A pentagonal bipyramid is the tegum product of a pentagon and an antipode. As a result, we observe $5 \times 2/5$ features and $2 \times 1/2$ features, rather than $7 \times 3/7$ features.



An octahedron is the tegum product of three antipodes. This doesn't change the observed lines since $3/6=1/2$.

많은 톰슨 솔루션은 작은 균일 다면체의 orthogonal subspace에 두 개의 다면체를 임베드하여 tegum product(direct sum)한 것으로 이해할 수 있다.

ex) **triangular bipyramid = tegum product(triangle, antipode)**

3개의 특징이 2차원을 사용하고, 2개의 특징이 1차원을 사용. 6개의 특징이 3/5차원을 사용하는 것과는 다르다.

orthogonal 하므로, 간섭의 영향도 없어서 더 유리함.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

Features varying in importance or sparsity

지금까지는 S_i 가 S 로 일정한 경우만을 고려했다.

하지만 실제로는 feature마다의 Sparsity S_i 가 다를것이다.

이를 확인하기 위해, 한 개의 feature의 sparsity만 제어하고,

이 feature를 제외한 다른 feature의 sparsity를 일정하게 하여 결과를 보고자 한다.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

Features varying in importance or sparsity

중요성이나 희소성이 변동하는 특징들은 불균형이 형성될 때 다면체의 부드러운 변형을 발생시킨다.

이렇게 변형이 진행되다가 임계 파손점에 도달하면 다른 다면체로 전환된다.

Correlated features

Correlated features들은 종종 서로 다른 tegum 인자에서 형성되며 직교하는 것을 선호한다.

결과적으로 직교하는 지역 기저를 형성할 수 있다.

직교할 수 없는 경우에는 나란히 있는 것을 선호한다.

어떤 경우에는 상관된 특징들이 단일 특징으로 통합된다.

Anti-correlated features

Superposition이 필요한 경우 같은 tegum 인자에 있는 것을 선호한다.

그들은 이상적으로 반대 위치에서 음의 간섭을 가지는 것을 선호한다.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

Correlated features ?

이미지 분류기에서 발생한다고 가정하면,

동물을 식별하는 데 사용되는 특징의 묶음(털, 귀, 눈)과 **건물을 식별**하는 데 사용되는 다른 묶음(모서리, 창문, 문)이 있을 수 있다.

이러한 묶음의 특징은 **함께 나타날 가능성이 높다**.

Windows (4b:237)
excite the car detector
at the top and inhibit
at the bottom.



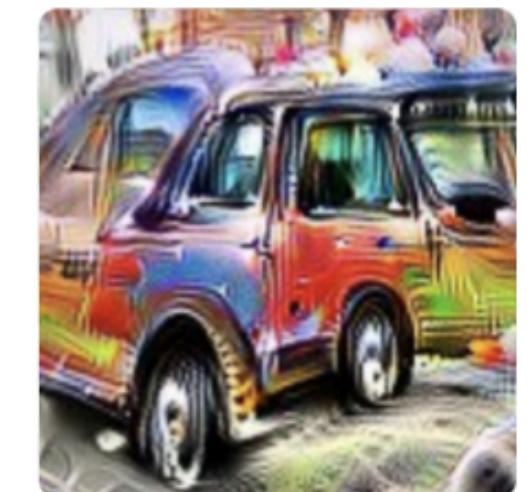
Car Body (4b:491)
excites the car
detector, especially at
the bottom.



Wheels (4b:373) excite
the car detector at the
bottom and inhibit at
the top.



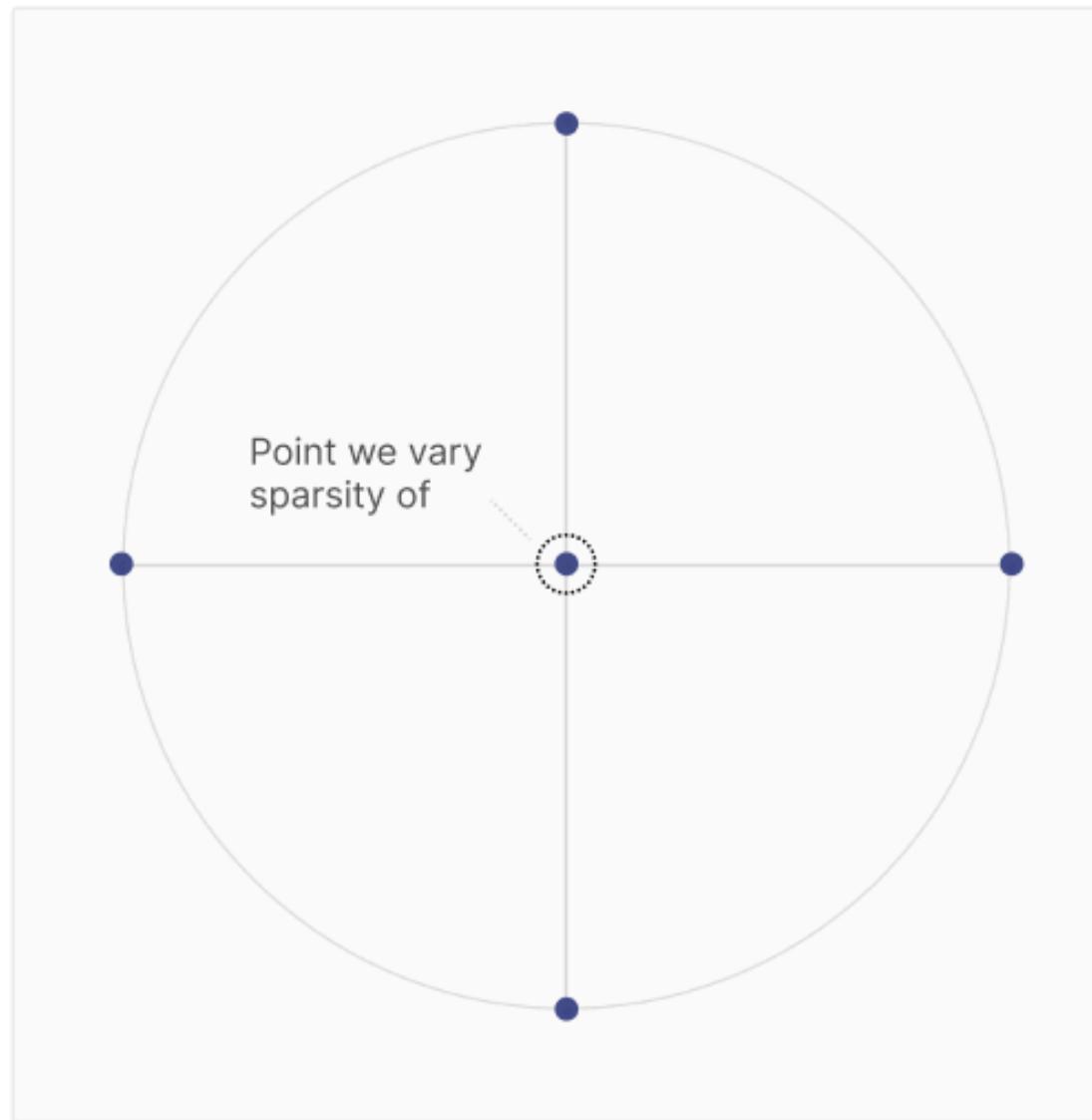
- positive (excitation)
- negative (inhibition)



A **car detector** (4c:447)
is assembled from
earlier units.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

Digon (Square) Solutions



When the sparsity of the varied point falls below a certain critical threshold (~2.5x less than others) the pentagon solution changes to two digons.

Pentagon Solutions



Note how vertices shift as sparsity changes

To study non-uniform sparsity, we consider models with five features, varying the sparsity of a single feature and observing how the resulting solutions change. We observe a mixture of continuous deformation and sharp phase changes.

Parameters

$n = 5$
 $m = 2$
 $I_i = 1$
 $1-S = 0.05$ (baseline)

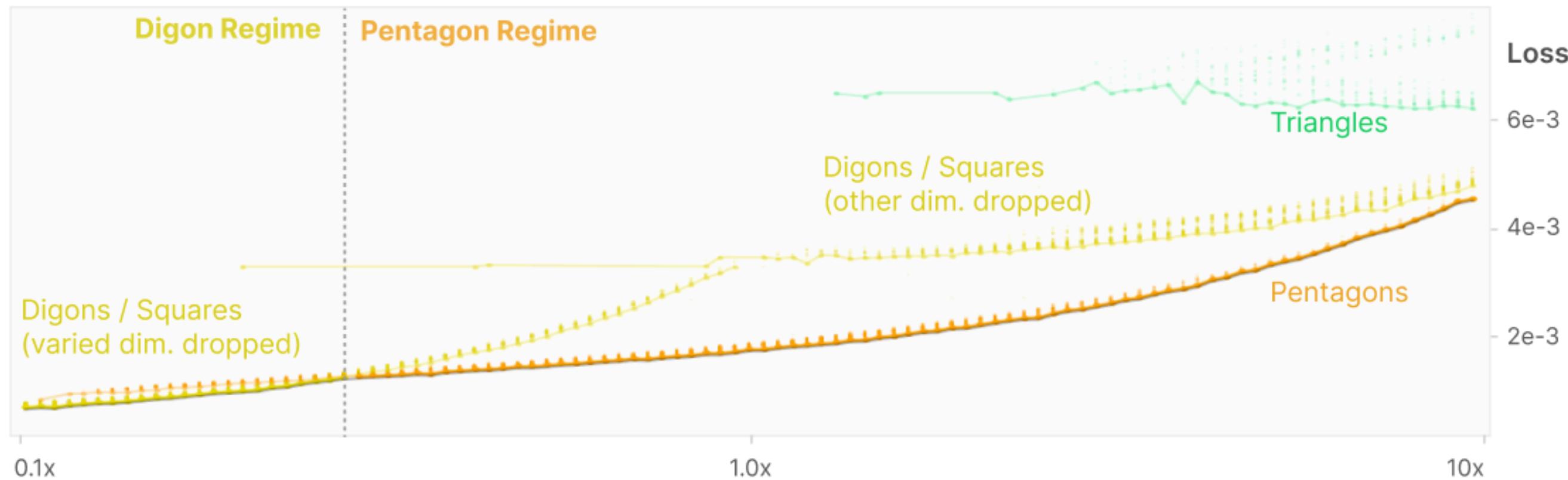
Relative Feature Density ($1-S$)



회소성이 일정 이상 감소하면 오각형이 깨진다.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

The Pentagon-Digon Phase Change Corresponds to a Loss Curve Crossover

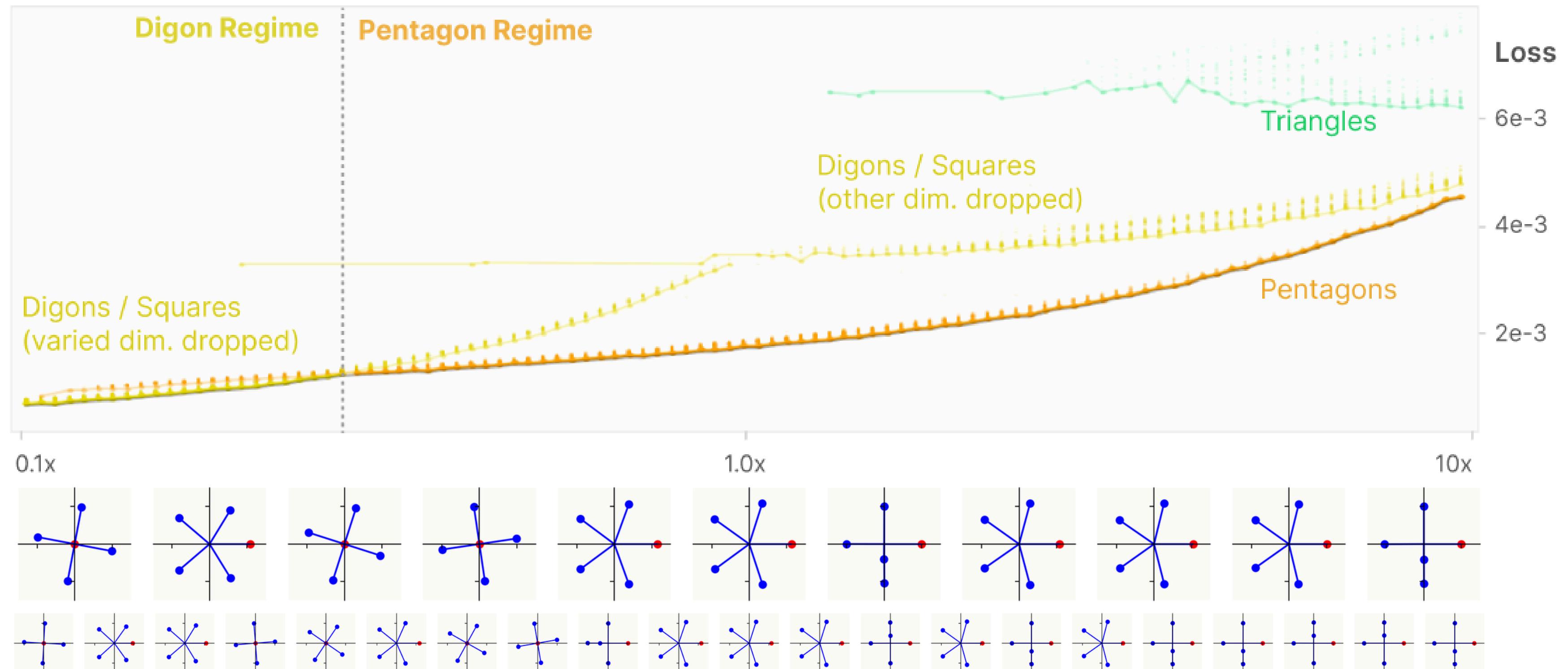


Gradient descent has trouble moving between solutions associated with different geometries. As a result, fitting the model will often produce non-optimal solutions. By characterizing and plotting these, we can see that each geometry creates a different loss curve, and that the pentagon-digon phase change corresponds to a cross over between the curves.

모델이 교차점에서 2개의 digons를 고수하기보다는, 오각형으로 상 변환을 하는 모습
굵은 선이 loss 함수이다.

Section 4: The Geometry of Superposition > Non-Uniform Superposition

The Pentagon-Digon Phase Change Corresponds to a Loss Curve Crossover

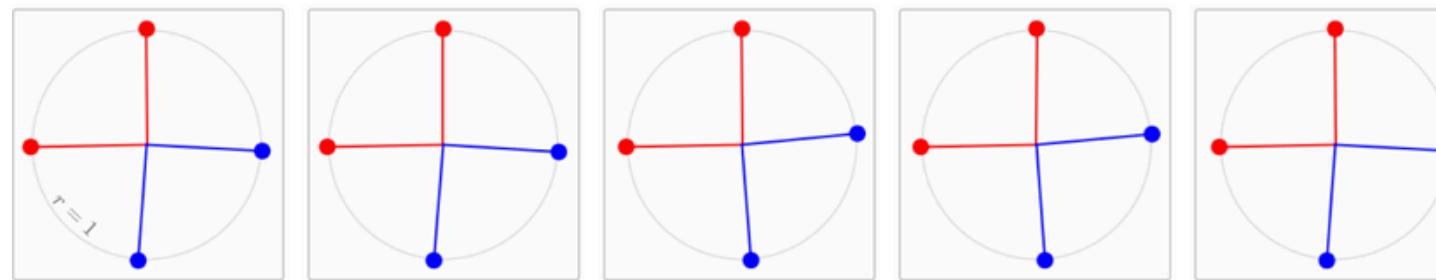


Section 4: The Geometry of Superposition > Correlated and Anticorrelated Features

비균등 중첩의 더 복잡한 형태는 특징 간의 상관 관계가 있을 때 발생한다.

Models prefer to represent correlated features in orthogonal dimensions.

We train several models with 2 sets of 2 correlated features ($n=4$ total) and a $m=2$ hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation.

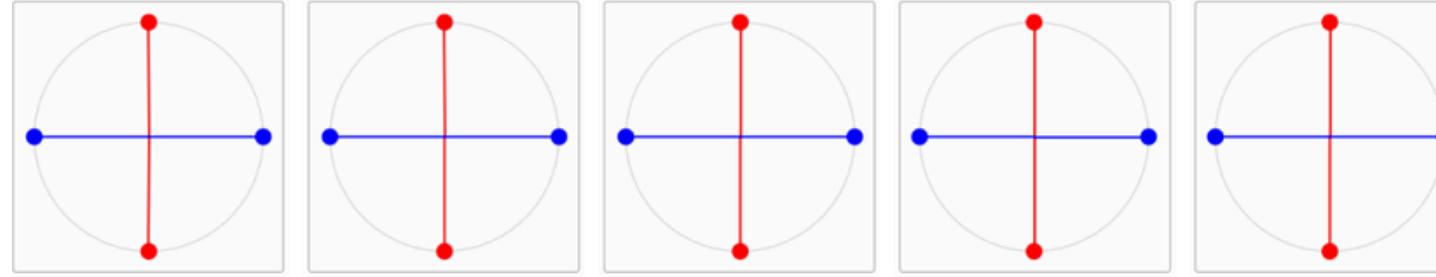


•• and •• denote correlated feature sets.

Correlated feature sets are constructed by having them always co-occur (ie. be zero or not) at the same time.

Models prefer to represent anticorrelated features in opposite directions.

We train several models with 2 sets of 2 anticorrelated features ($n=4$ total) and a $m=2$ hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation.

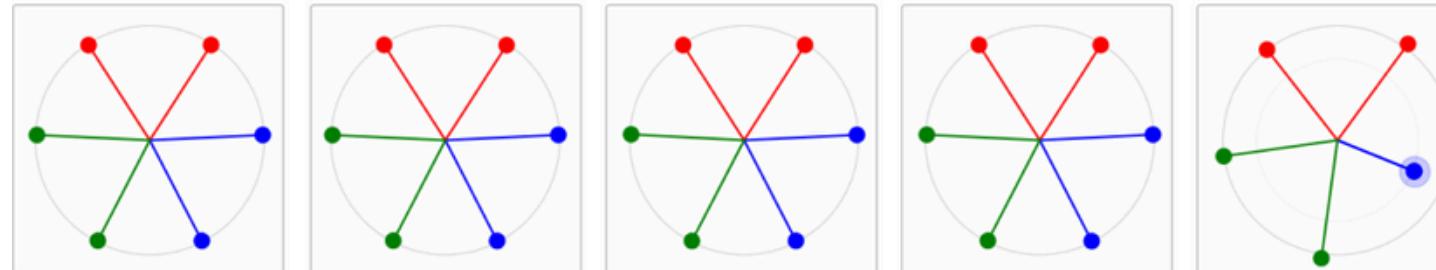


•• and •• denote anticorrelated feature sets.

Anticorrelated feature sets are constructed by having them never co-occur (ie. be zero or not) at the same time.

Models prefer to arrange correlated features side by side if they can't be orthogonal.

We train several models with 3 sets of 2 correlated features ($n=6$ total) and a $m=2$ hidden dimensions. We then visualize the weight column for each feature. For ease of comparison, we rotate and flip solutions to have a consistent orientation. (Note that models will not embed 6 independent features as a hexagon like this.)



••, ••, and •• denote correlated feature sets.

Sometimes correlated feature sets "collapse". In this case it's an optimization failure, but we'll return to it shortly as an important phenomenon.

Correlated features

Correlated features들은 직교하는 것을 선호한다.

결과적으로 직교하는 지역 기저를 형성할 수 있다.

직교할 수 없는 경우에는 나란히 있는 것을 선호한다.

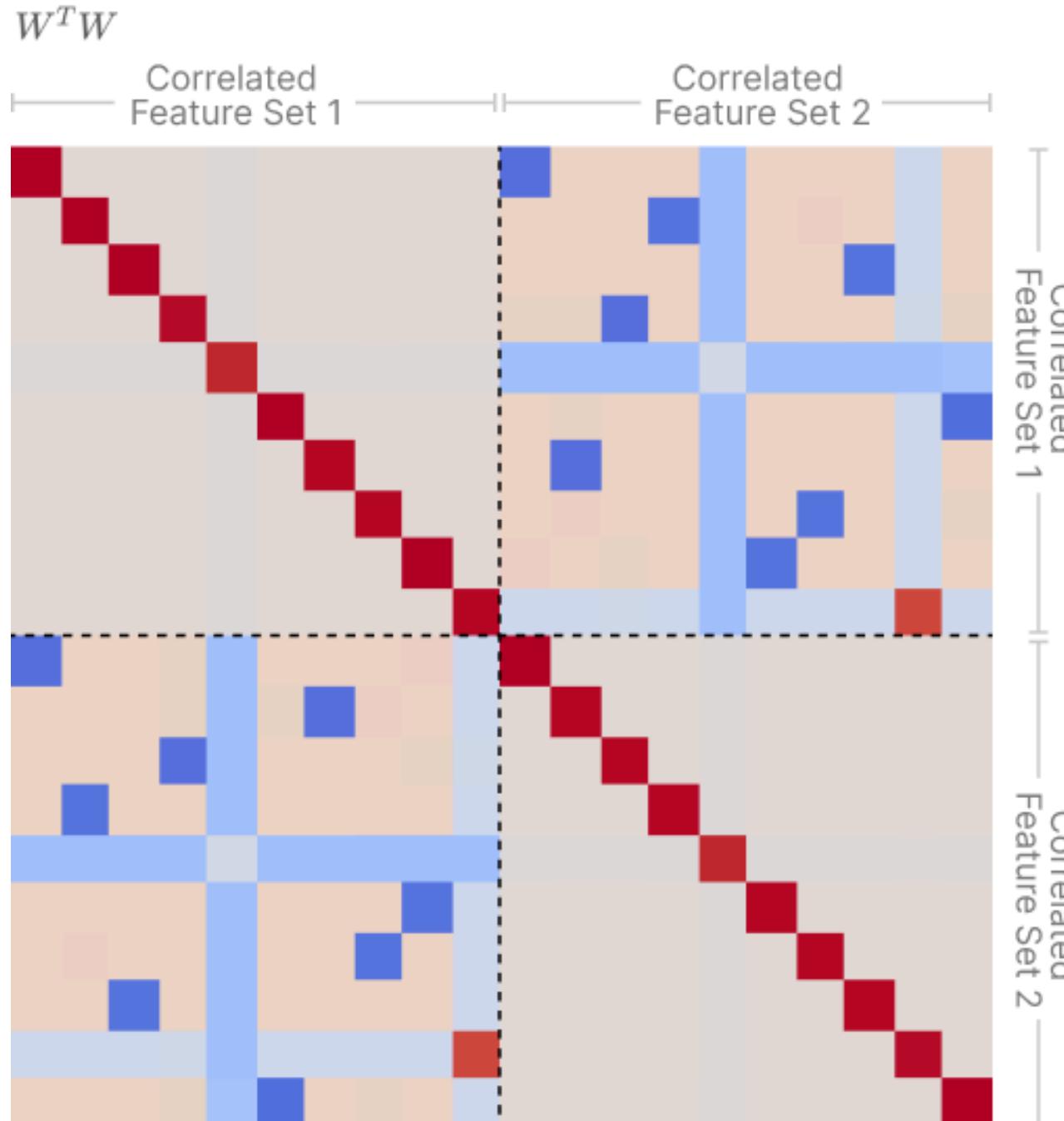
어떤 경우에는 Correlated feature들이 단일 특징으로 통합된다.

Anti-correlated features

Superposition이 필요한 경우 같은 tegum 인자에 있는 것을 선호한다.

그들은 이상적으로 반대 위치에서 음의 간섭을 가지는 것을 선호한다.

Section 4: The Geometry of Superposition > Correlated and Anticorrelated Features



Models prefer to represent correlated features in orthogonal dimensions, creating “local orthogonal bases”.

We train a model with 2 sets of 10 correlated features ($n=20$ total) with $m=10$ hidden dimensions.

Within each set of correlated features, the model creates a *local orthogonal basis*, having each feature be represented orthogonally.

Correlated features

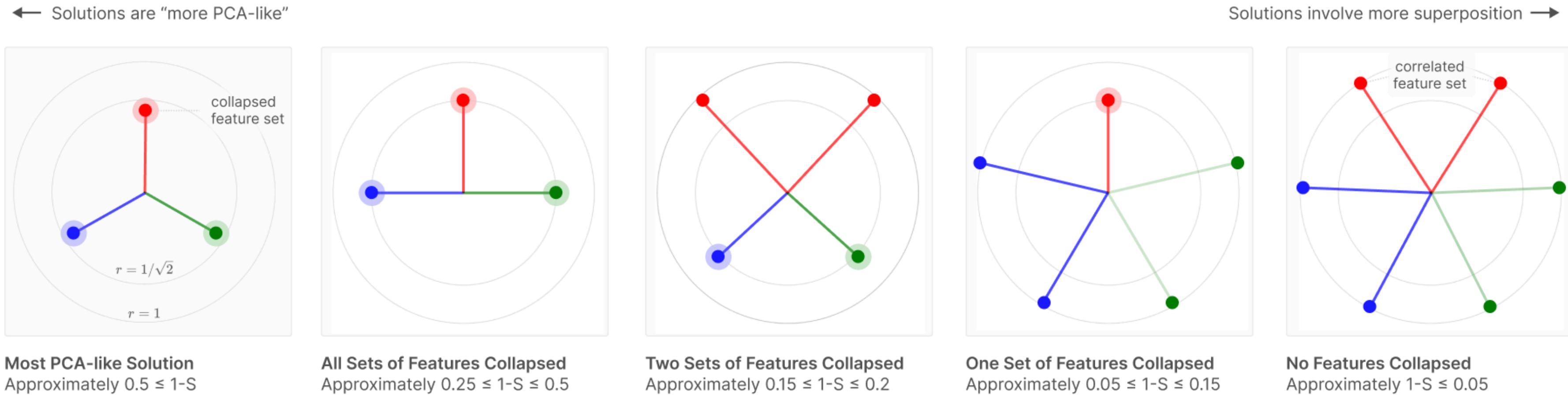
Correlated features들은 직교하는 것을 선호한다.

결과적으로 직교하는 지역 기저를 형성할 수 있다.

직교할 수 없는 경우에는 나란히 있는 것을 선호한다.

어떤 경우에는 상관된 특징들이 단일 특징으로 통합된다.

Section 4: The Geometry of Superposition > Correlated and Anticorrelated Features



가장 흥미로운 점은, **PCA와 superposition 사이에 trade-off가 존재하는 것처럼 보인다는 것**

feature가 2개의 correlated feature를 하나로 표현해야 하는 경우, 모델은 이들의 PC를 표현하고자 한다

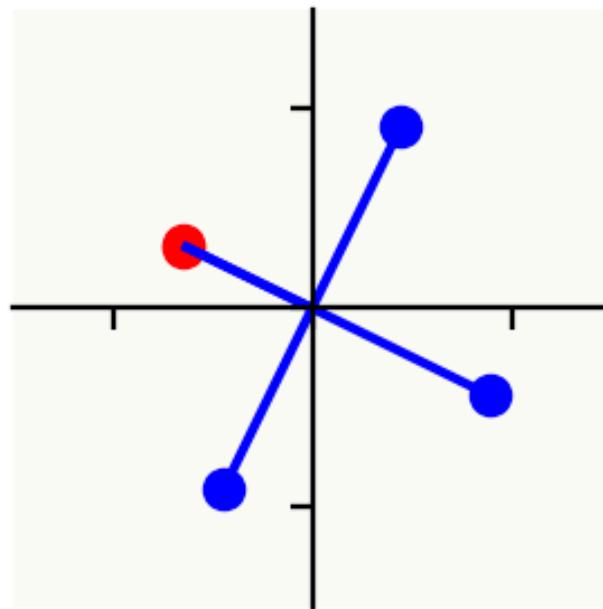
Sparsity를 줄이면, feature는 점진적으로 주성분으로 “붕괴”되며, 해결책이 PCA와 동등해진다.

Feature가 더 상관관계가 생길수록 PCA는 더 나은 전략이 되고, 특징이 더 sparse해질수록 superposition은 더 나은 전략이 된다.

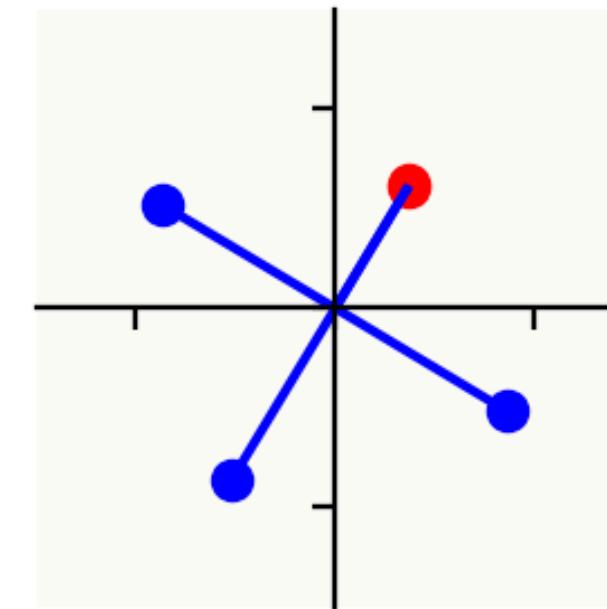
Feature가 모두 sparse하고 상관관계가 있을 때는 각 전략의 혼합이 나타난다.

Section 4: The Geometry of Superposition > Correlated and Anticorrelated Features

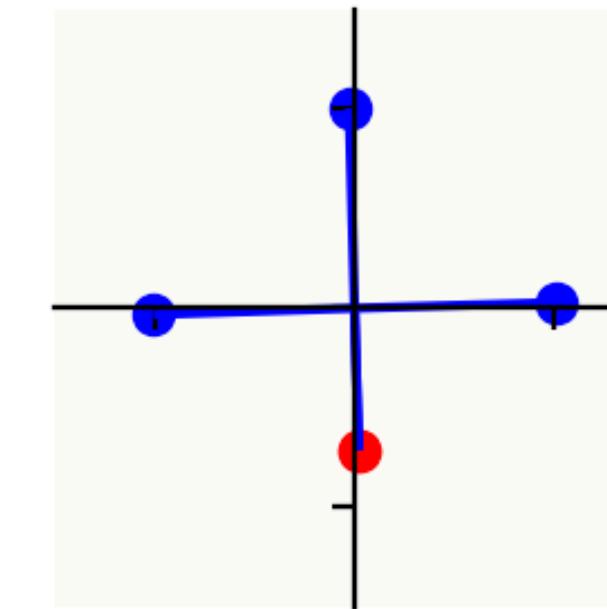
← Solutions are “more PCA-like”



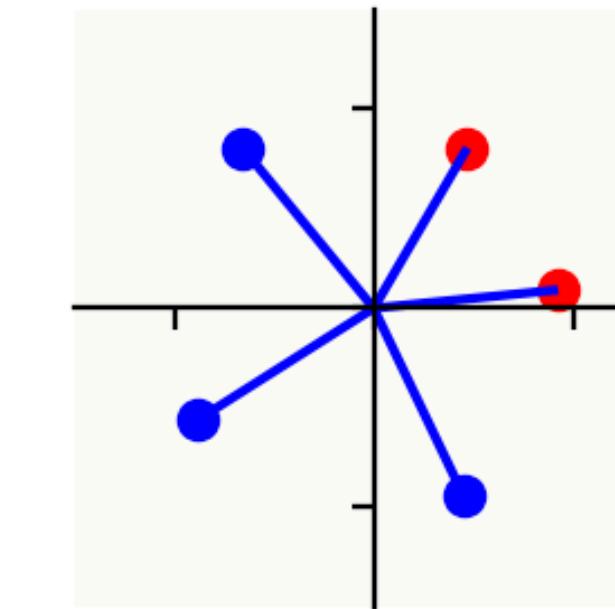
Most PCA-like Solution
Approximately $0.5 \leq 1-S$



All Sets of Features Collapsed
Approximately $0.25 \leq 1-S \leq 0.5$

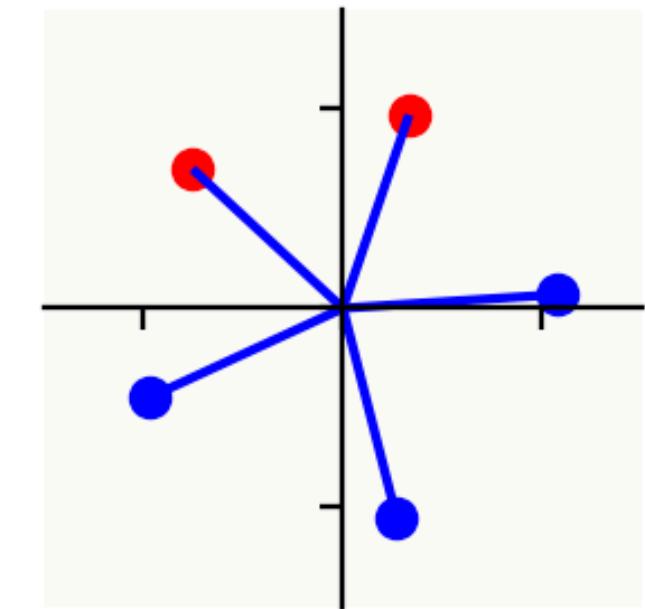


Two Sets of Features Collapsed
Approximately $0.15 \leq 1-S \leq 0.2$

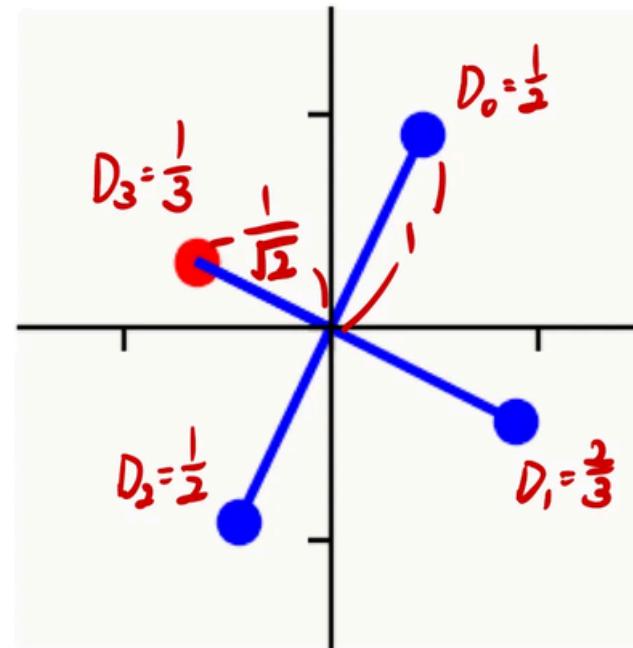


One Set of Features Collapsed
Approximately $0.05 \leq 1-S \leq 0.15$

Solutions involve more superposition →



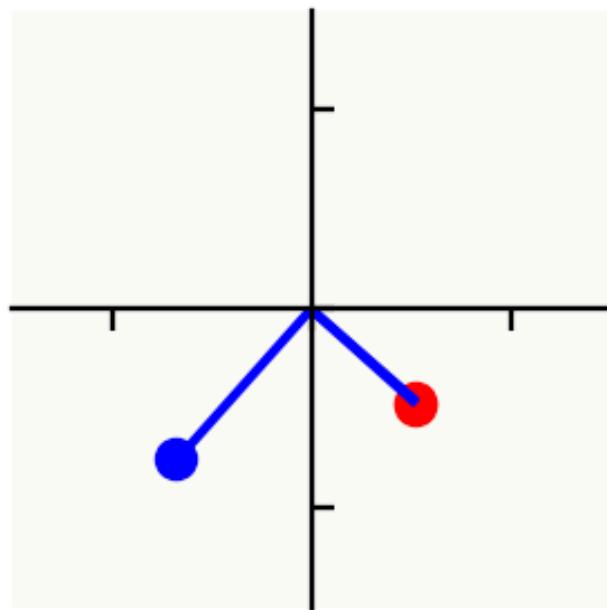
No Features Collapsed
Approximately $1-S \leq 0.05$



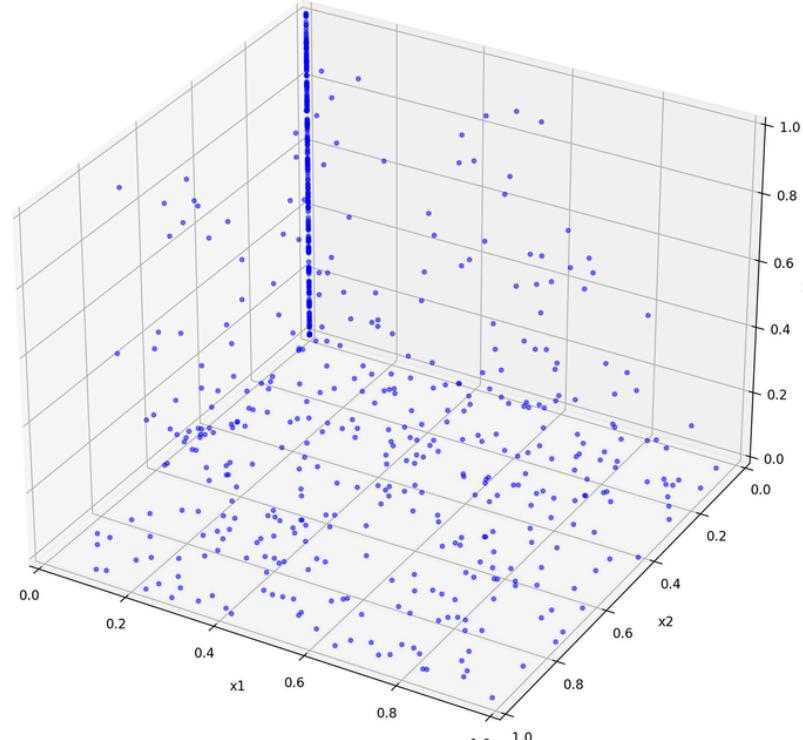
$$D_i = \frac{\|w_i\|^2}{\sum_j (\hat{w}_i \cdot w_j)^2}$$

Section 4: The Geometry of Superposition > Correlated and Anticorrelated Features

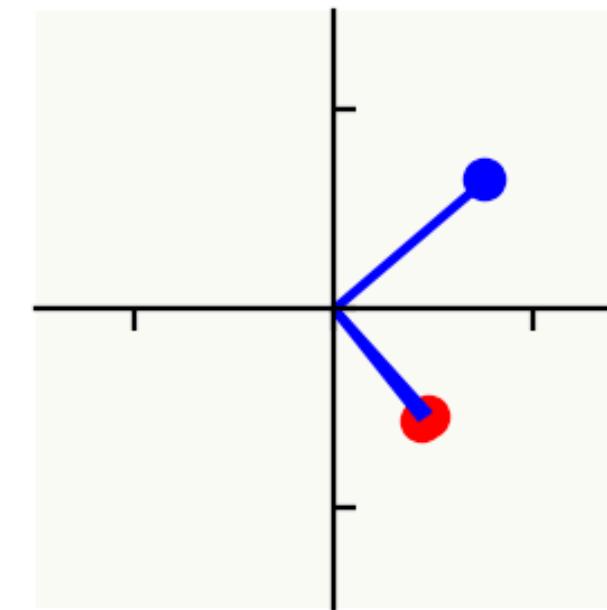
← Solutions are “more PCA-like”



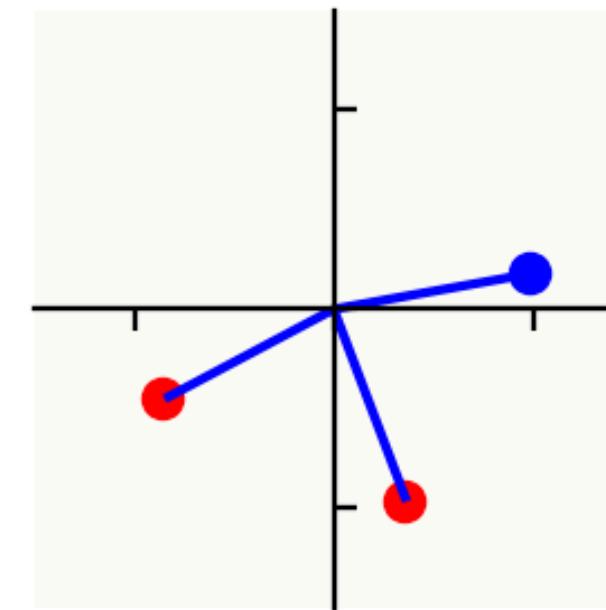
3D Distribution of Batch [0.59 1. 0.59]



One Set of Features Collapsed
ximatively $0.25 \leq 1-S \leq 0.5$

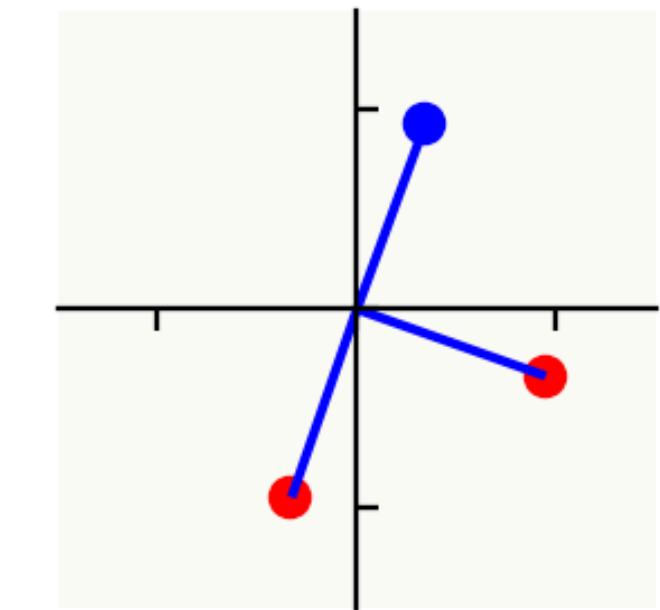


Two Sets of Features Collapsed
Approximately $0.15 \leq 1-S \leq 0.2$

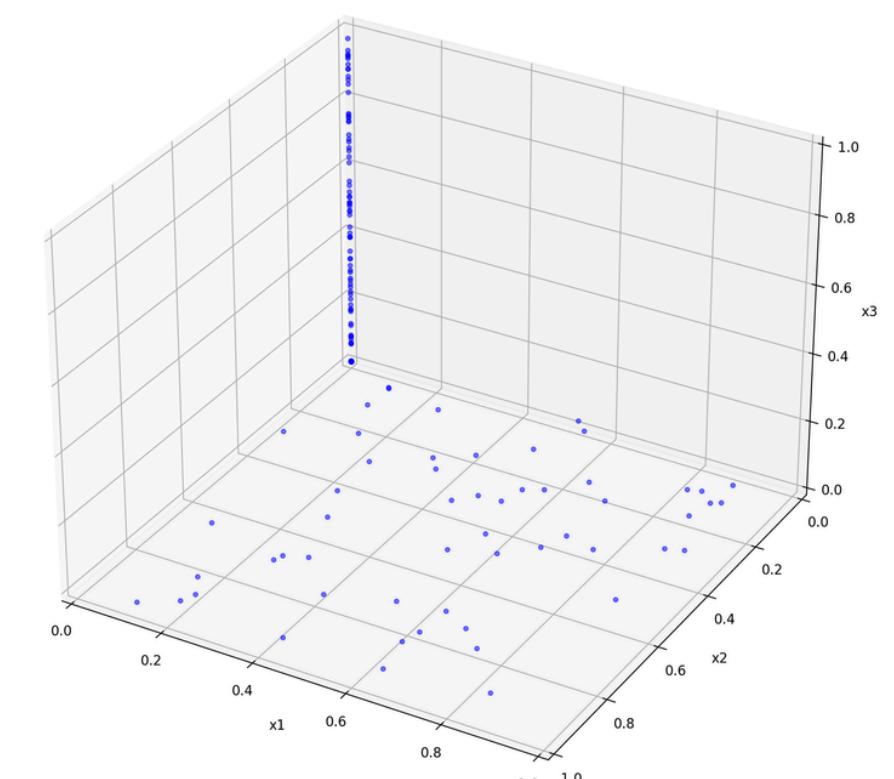


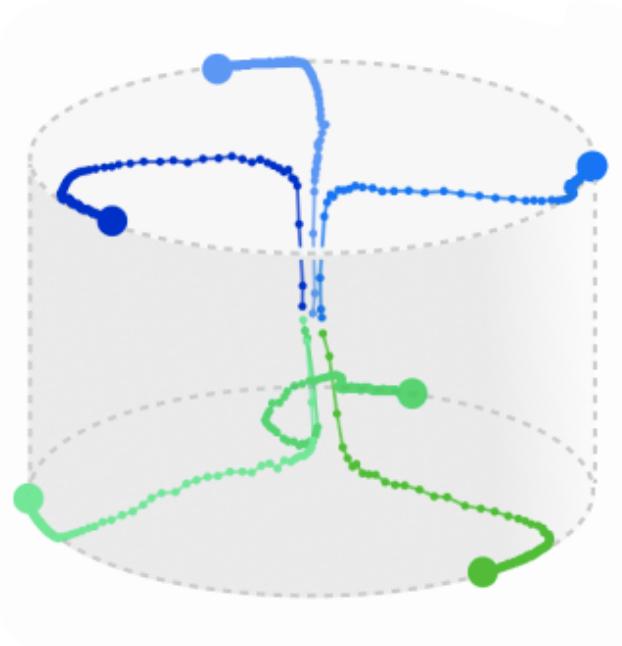
One Set of Features Collapsed
Approximately $0.05 \leq 1-S \leq 0.1$

Solutions involve more superposition →



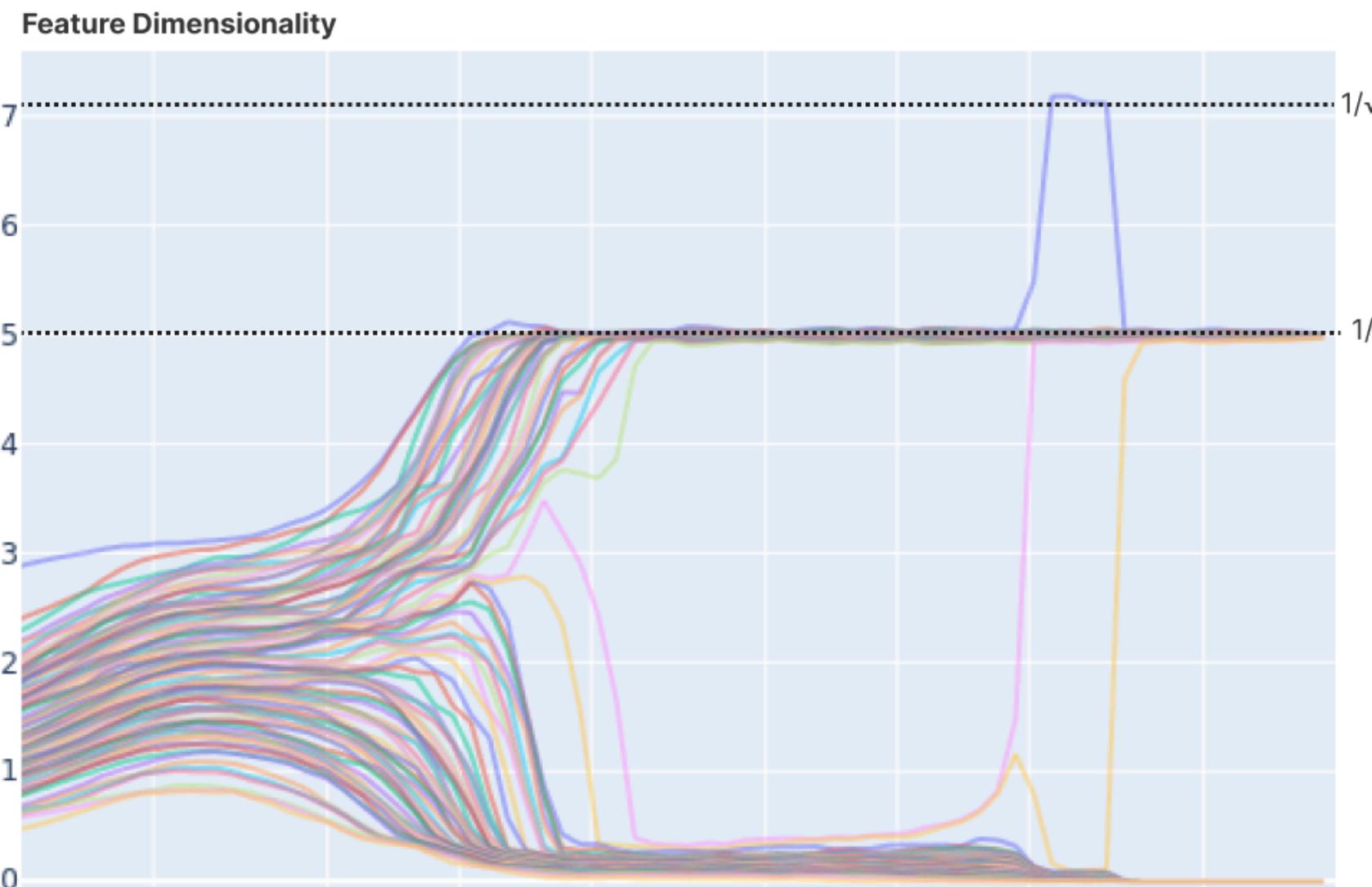
3D Distribution of Batch [0.94 1. 0.94]





Section 5: Learning Dynamics

Section 5: Superposition and Learning Dynamics



Phenomenon 1: Discrete "Energy Level" Jumps

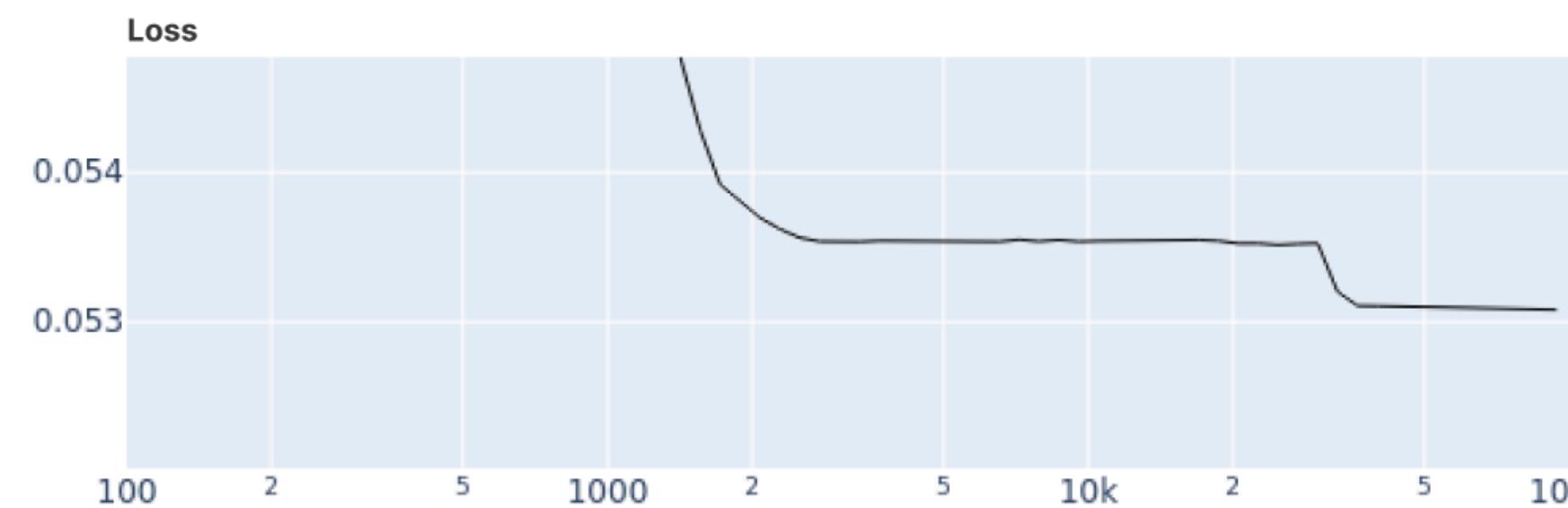
특징의 수가 많은 장난감 모델의 학습 역학이 "에너지 레벨 점프"에 의해 지배되는 것으로 보인다.

학습이 진행됨에 따라 **feature**가 완전히 무시되거나,
antipodal pairs로 가게 된다.

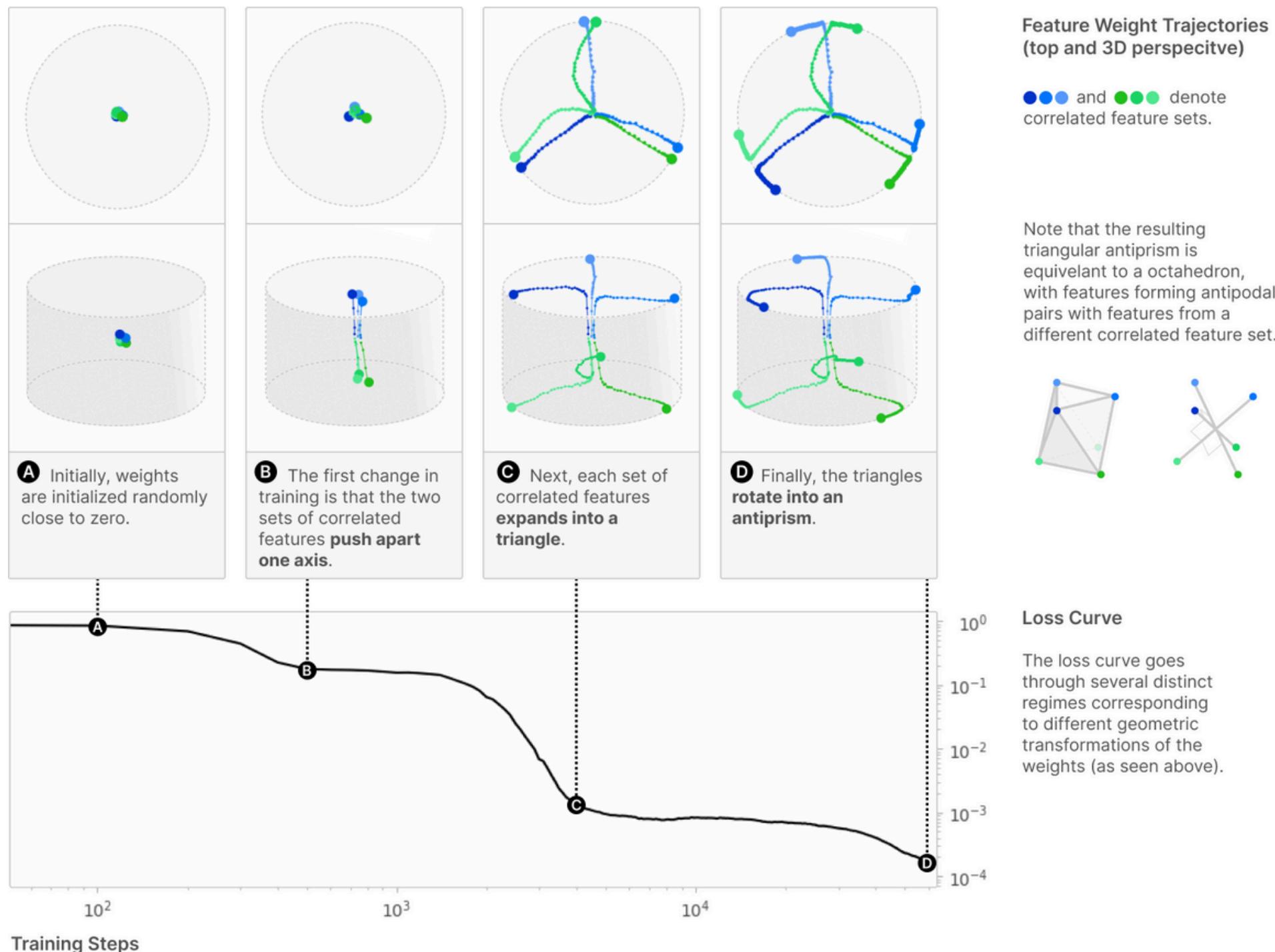
이때 loss가 급격히 감소하는 모습을 보인다.

더 큰 모델에서 손실 곡선이 겉보기에 매끄럽게 감소하는 것은 실제로 서로 다른 구성 간의 작은 특징 점프들로 구성되어 있을 것이다.

어쩌면 **grokking**과 관련이 있을지도 모른다.



Section 5: Superposition and Learning Dynamics



Phenomenon 2: Learning as Geometric Transformations

6개의 특징 벡터를 3차원으로 표현하고자 할 때의 상황이다.

정팔면체의 표현은 3개의 antipode로 표현된다.

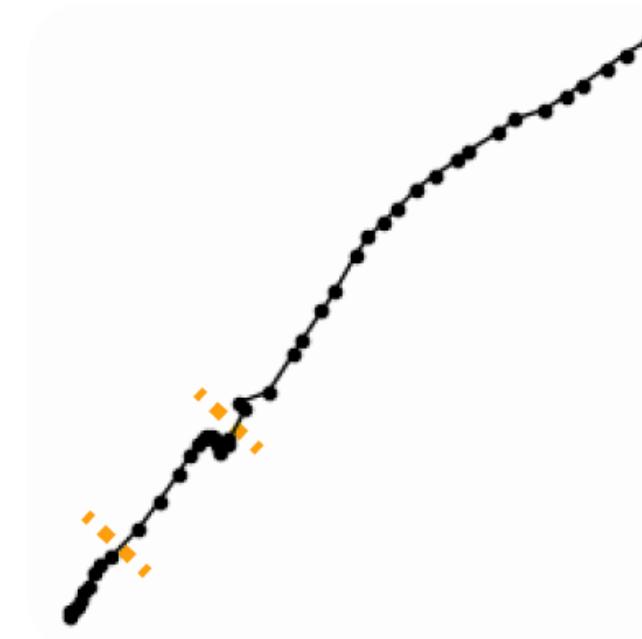


이전 섹션들에서 언급했던 기하학적 구조를 가지면서 loss가 급격히 감소하는 모습을
볼 수 있다.



An octahedron is the tegum product of three antipodes.
This doesn't change the observed lines since $3/6=1/2$.





Section 6: Relationship to Adversarial Examples

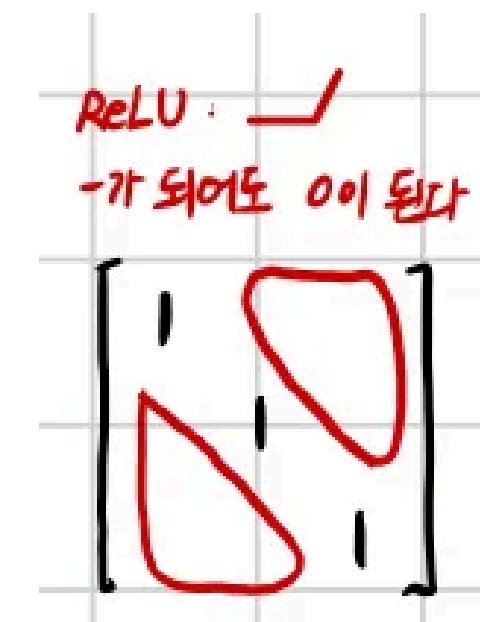
Section 6: Relationship to Adversarial Examples

In a model without superposition, the end-to-end weights for the first feature are:

$$(W^T W)_0 = (1, 0, 0, 0, \dots)$$

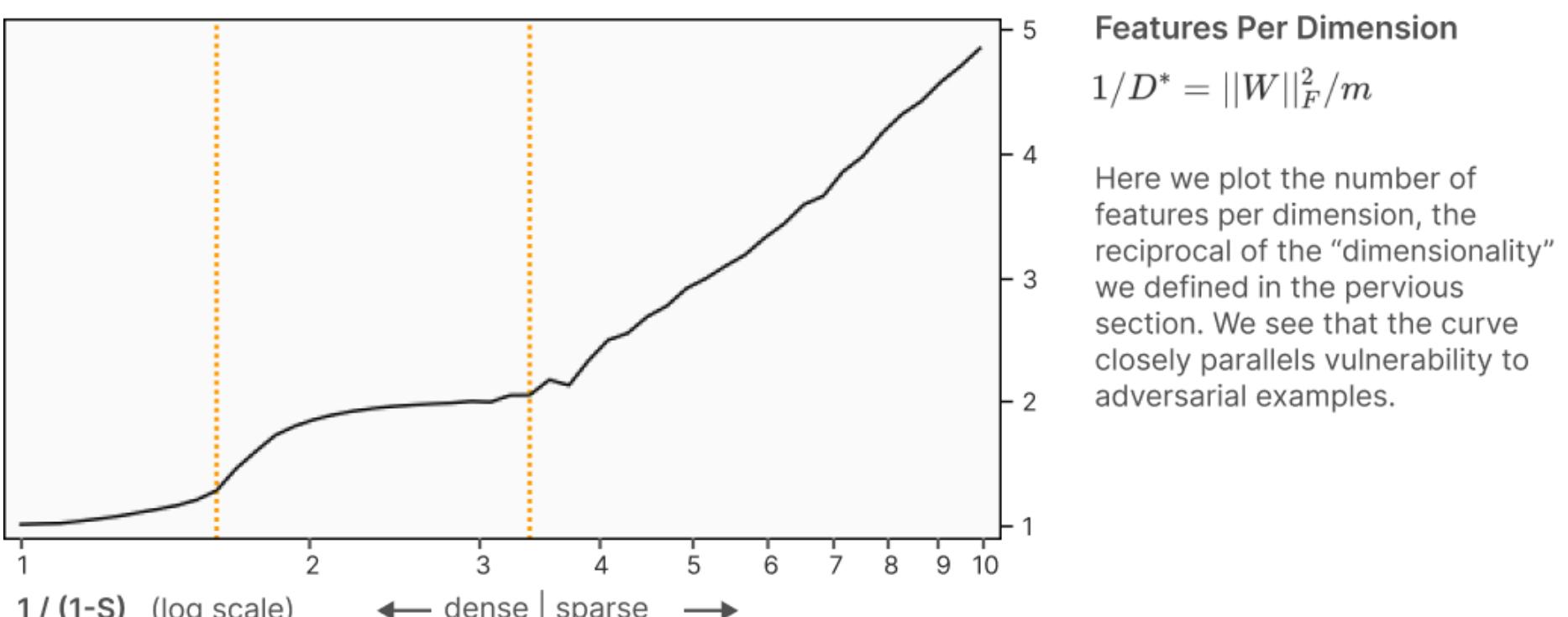
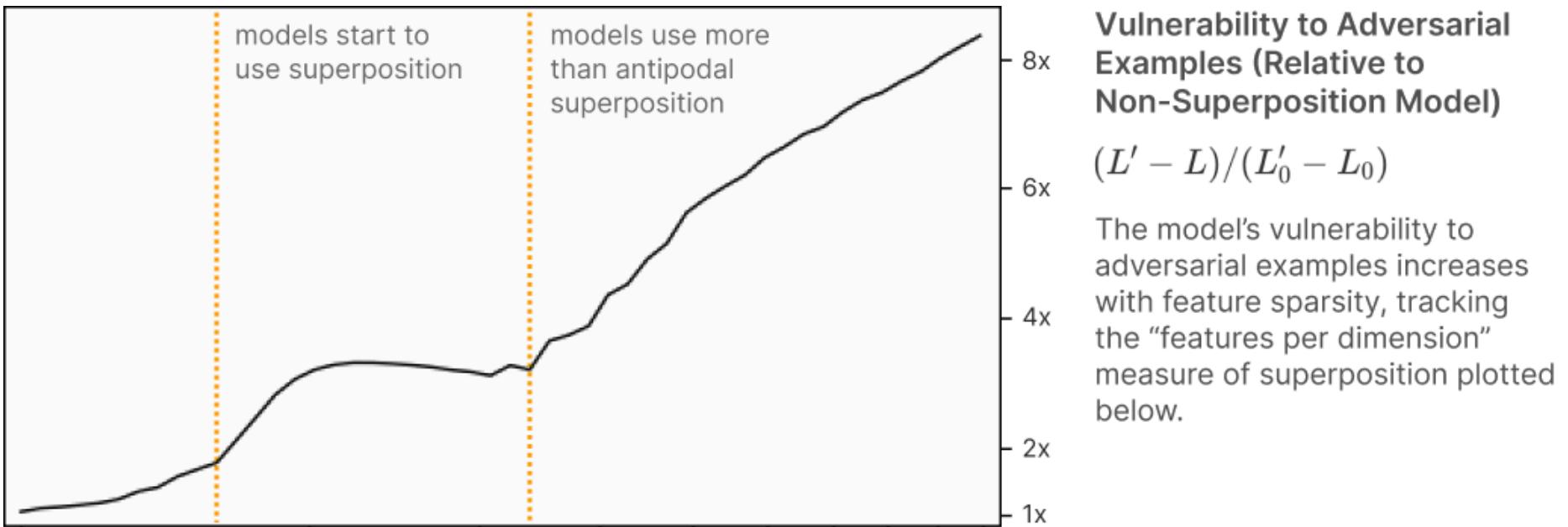
But in a model with superposition, it's something like:

$$(W^T W)_0 = (1, \epsilon, -\epsilon, \epsilon, \dots)$$

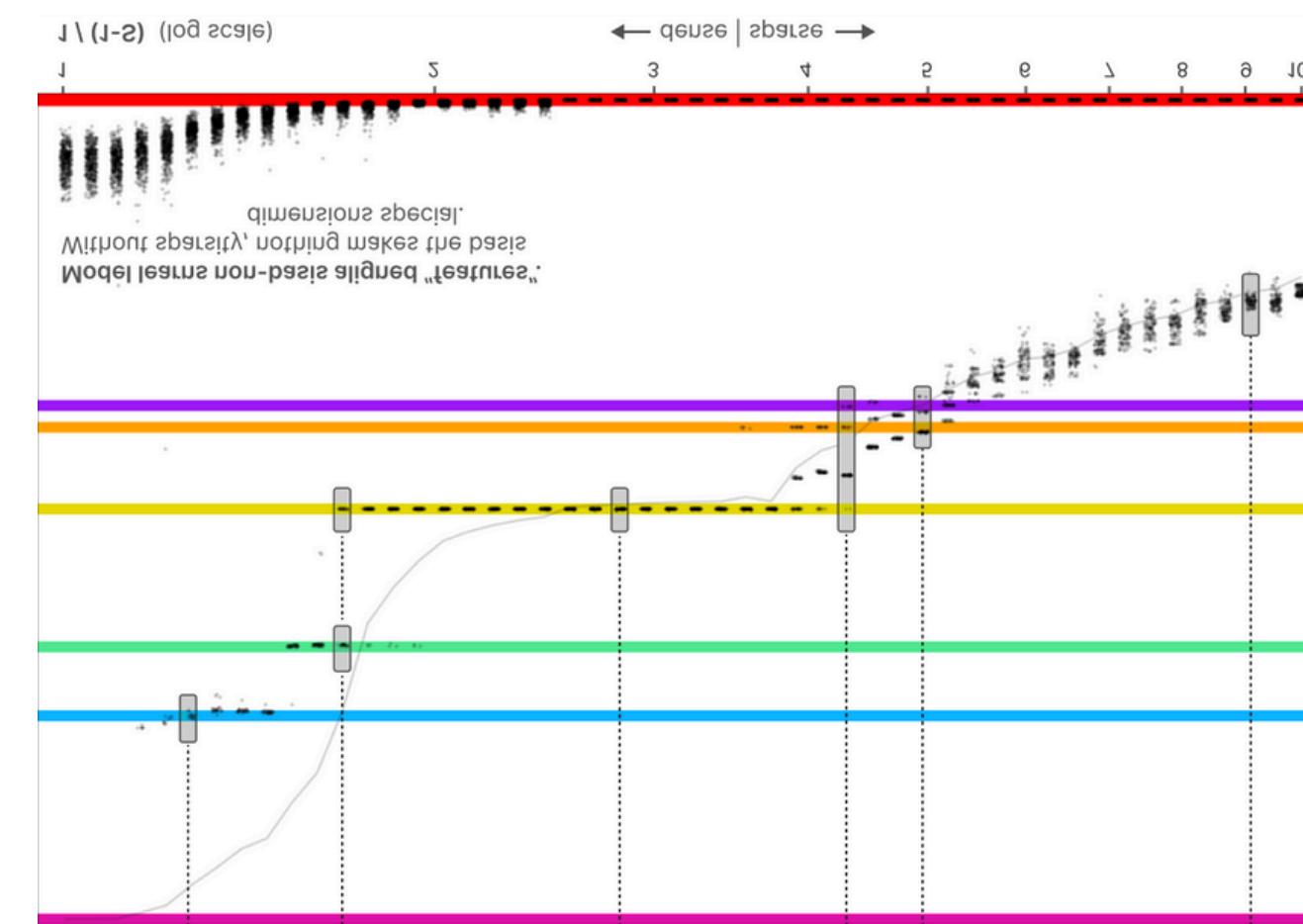


희소한 데이터에 맞춘 모델의 최적 행동은 더 많은 특성을 표현하기 위해 중첩을 사용하는 것이며, 이는 공격에 취약하게 만든다.

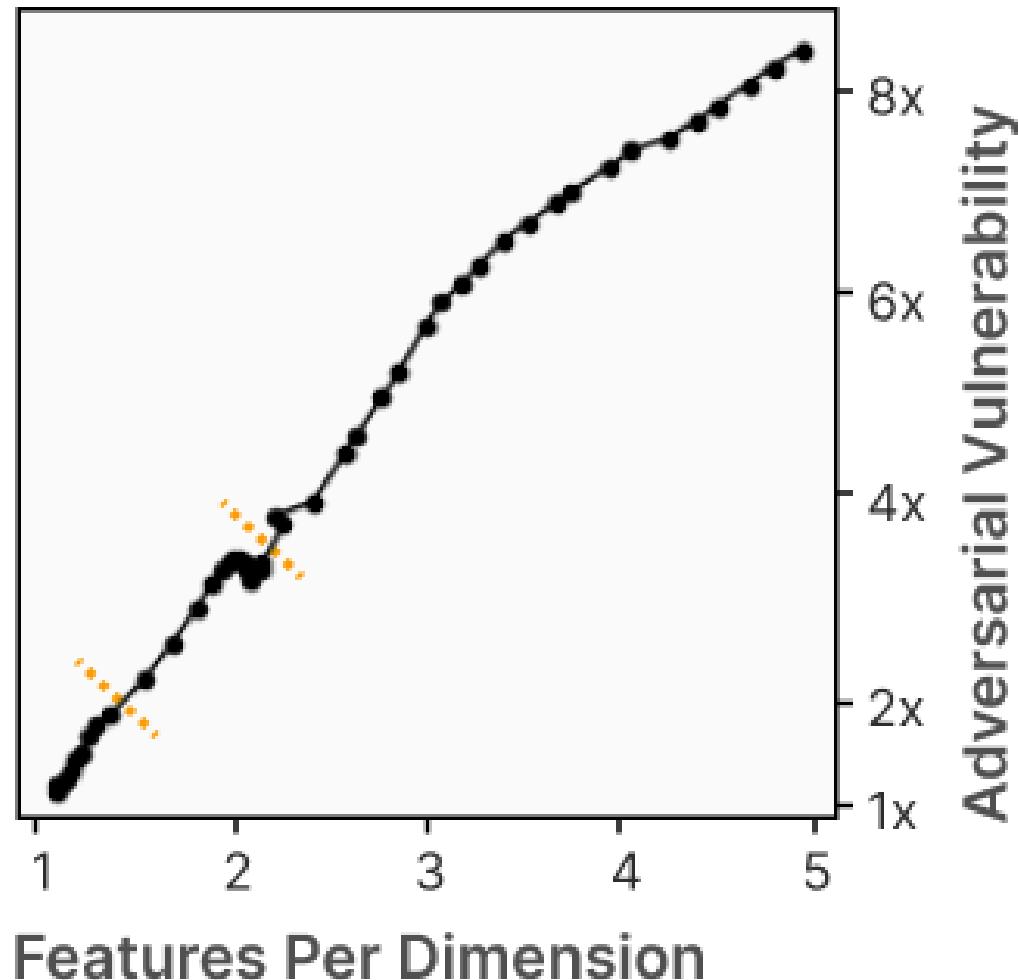
Section 6: Relationship to Adversarial Examples



Model이 **superposition**을 사용하면서부터 attack에 대한 취약성이 증가한다. 특히 **antipodal** 을 벗어나면서부터 그 취약성이 더 증가하게 된다.

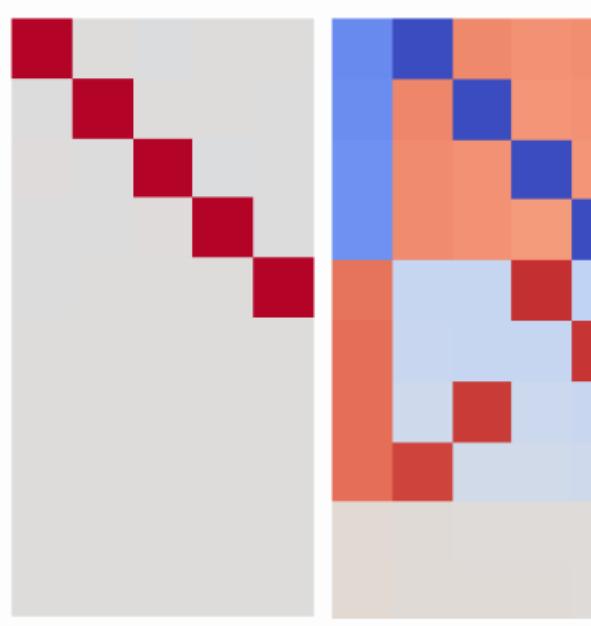


Section 6: Relationship to Adversarial Examples



We can also directly plot adversarial vulnerability against the number of features per dimension. This reveals that adversarial vulnerability is highly correlated with the number of features stored in superposition per dimension.

“차원 당 superposition으로 저장된 feature의 갯수”와 취약성은 연관관계가 있어 보인다.



Section 7: Superposition in a Privileged Basis

Section 7: Superposition in a Privileged Basis

Privileged basis를 만드는 방법:

Activation function을 추가하거나, **hidden layer**에 L1 정규화를 적용시키는 것

MLP의 구조에 대해서 잘 이해하는 것을 목적으로 하므로, 전자의 방법을 선택한다.

전에 있던 구조에서, **projection 시의 과정에도 ReLU를 사용하는 것**에 주목.

ReLU Output Model

$$h = Wx$$

$$x' = \text{ReLU}(W^T h + b)$$

$$x' = \text{ReLU}(W^T Wx + b)$$

This gives us the following "ReLU hidden layer" model:

$$h = \text{ReLU}(Wx)$$

$$x' = \text{ReLU}(W^T h + b)$$

We'll train this model on the same data as before.

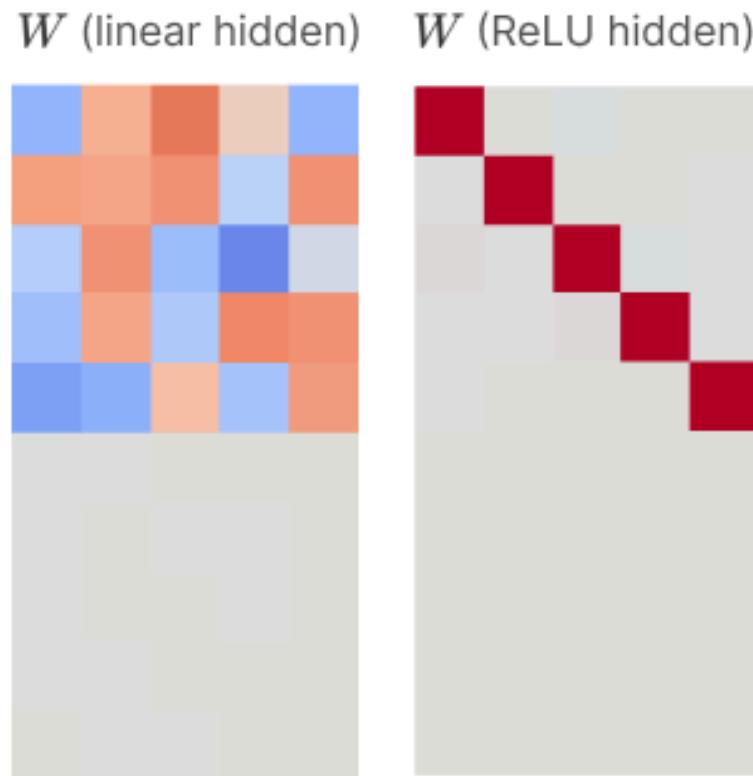
Section 7: Superposition in a Privileged Basis

이전에는 $W^T W$ 를 시각화하면서 해석하고자 했지만, 이번에는 W 를 직접 해석할 수 있게 된다.

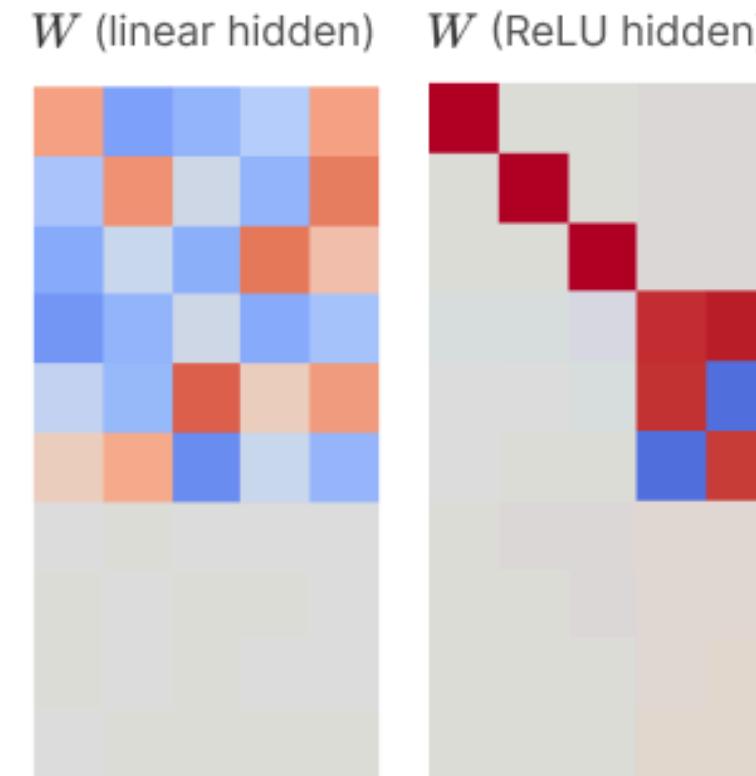
-> 이전에 시각화했던 것들은 hidden layer가 2차, 3차원 상에서만 해석이 가능합니다. (제가 했던 것들)

A Privileged Basis Makes W Directly Interpretable

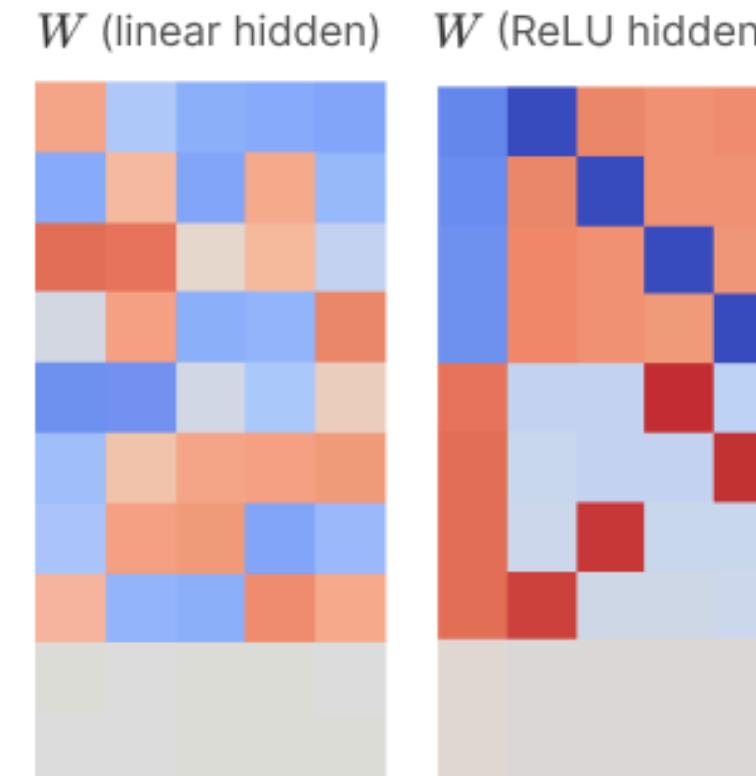
No Superposition



6 Features in 5 Dimensions



8 Features in 5 Dimensions

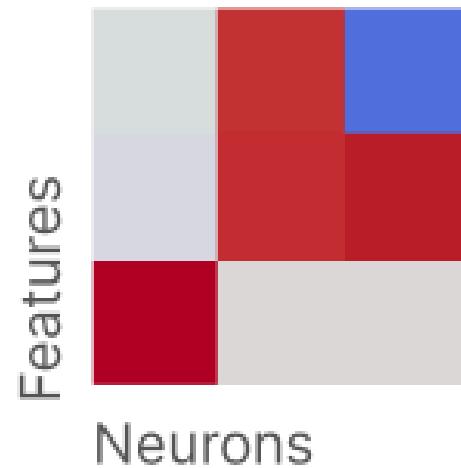


Weight / Bias
Element
Values
-1 0 1

Section 7: Superposition in a Privileged Basis

W as Matrix

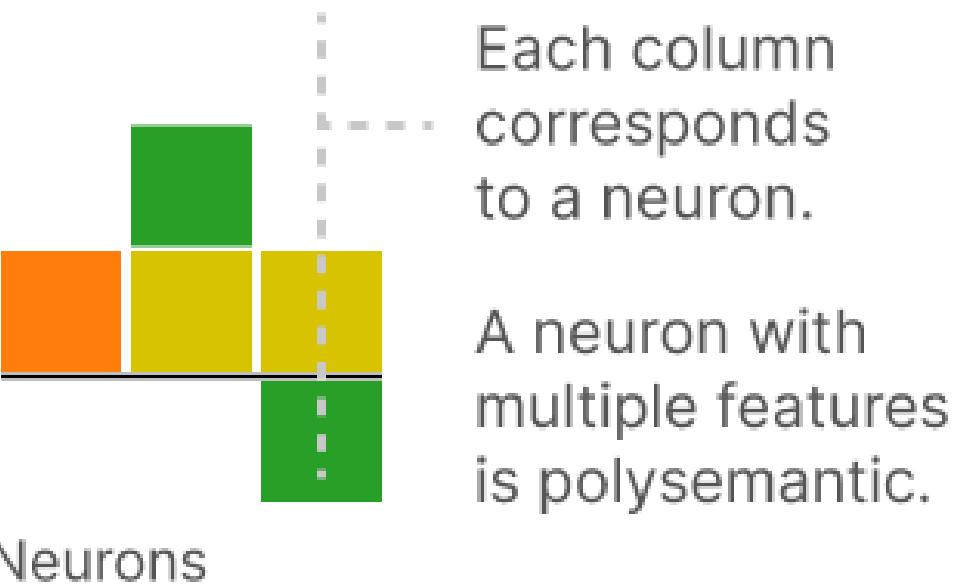
Since the hidden layer now has a privileged basis we can visualize the raw weight matrix.



- Let's assign a color to each feature for the stack plot.

W as Stack Plot

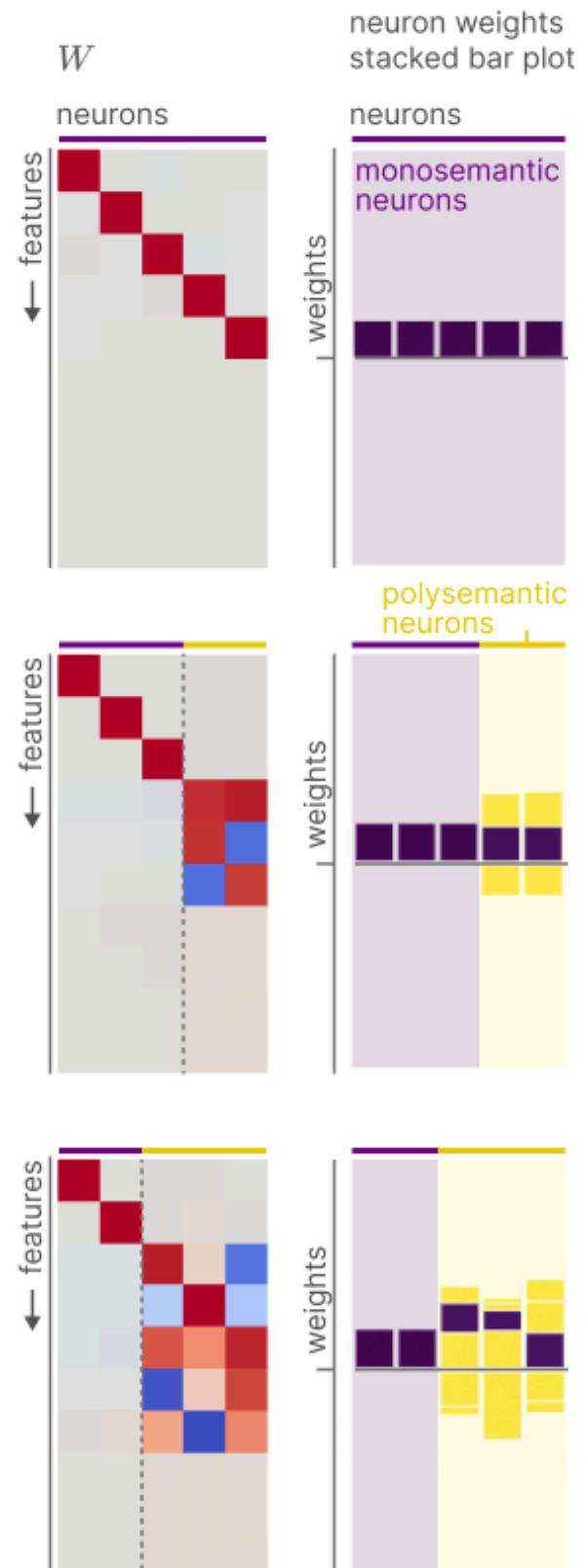
Instead of showing a matrix, we can map features to colors and stack the weights per neuron.



Each column corresponds to a neuron.

A neuron with multiple features is polysemantic.

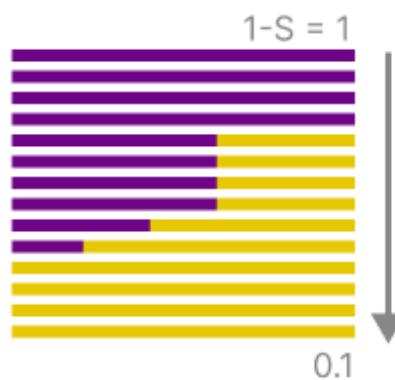
Section 7: Superposition in a Privileged Basis



All Monosemantic Neurons (approximately $0.5 \leq 1-S$)

In this regime, every feature gets a dedicated neuron. That is, the model consists entirely of monosemantic neurons.

As feature sparsity increases, we see neurons shift from being **monosemantic** to being **polysemantic**.

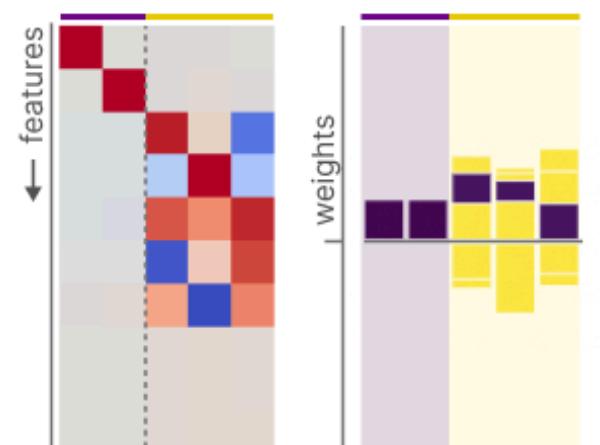


Three Features in Two Neurons (approximately $0.2 \leq 1-S \leq 0.5$)

In this regime, the three most important features still get dedicated neurons, but the next three are represented in a kind of binary code by two neuron's activations.

These two neurons form a binary code for three features. Note that the (1,1) feature is orthogonal to the others, while the other two are an antipodal pair.

Individual features are colored based on whether they're in superposition



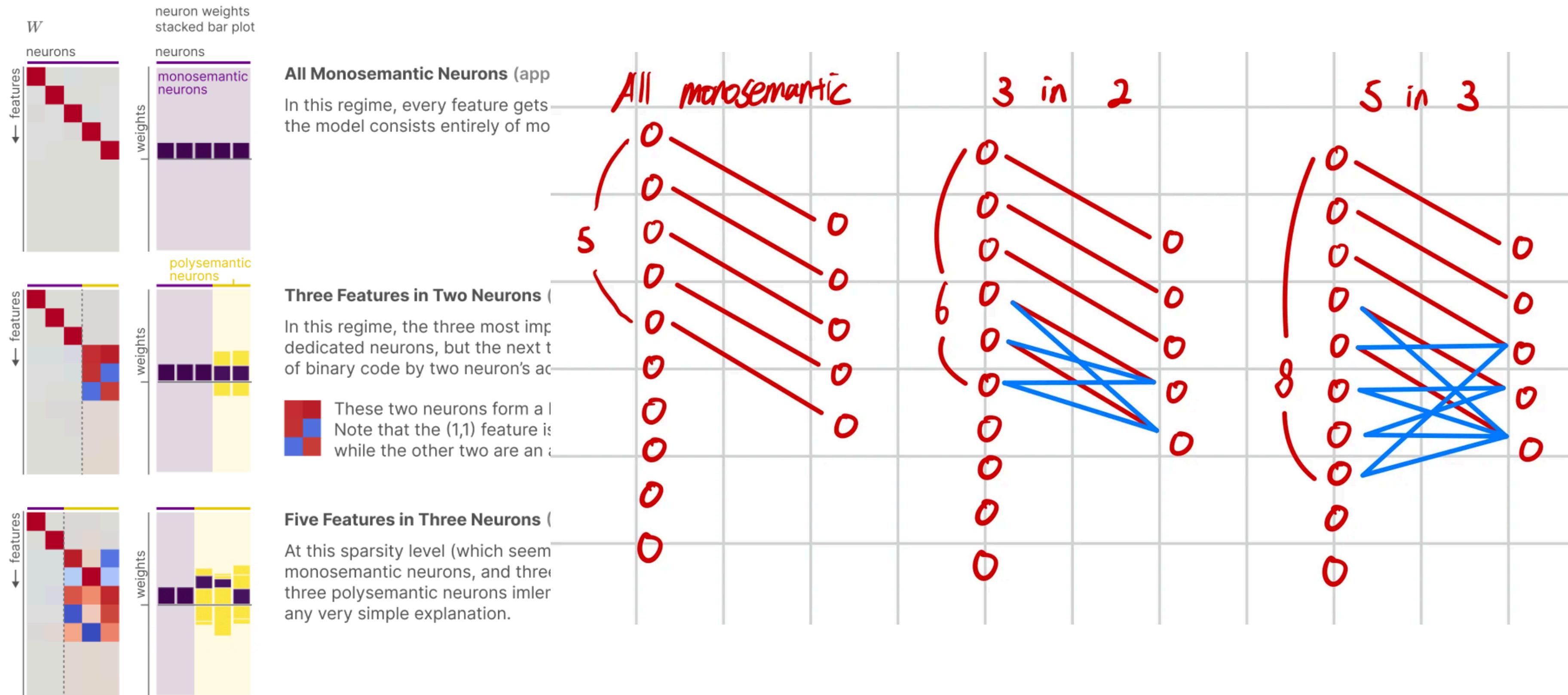
Five Features in Three Neurons ($1-S = 0.15$)

At this sparsity level (which seems quite narrow), we still see two monosemantic neurons, and three polysemantic neurons. The three polysemantic neurons implement a code that doesn't have any very simple explanation.

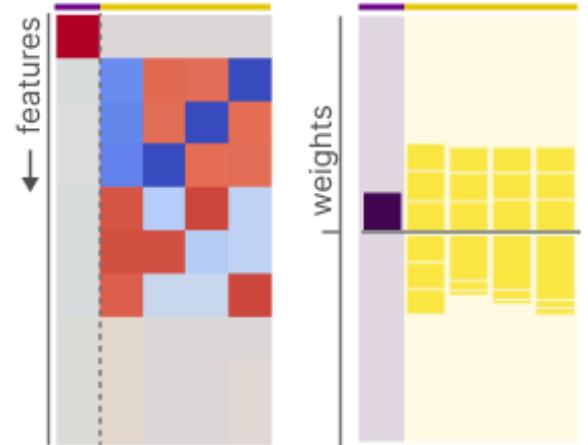
$$\sum_j (\hat{x}_i \cdot x_j)^2$$

0	1
dark purple	yellow

Section 7: Superposition in a Privileged Basis

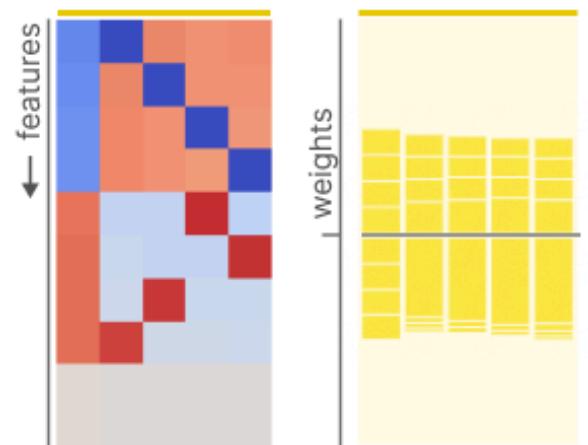


Section 7: Superposition in a Privileged Basis



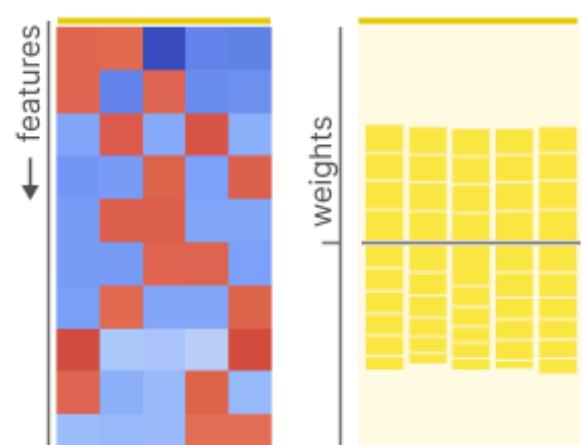
Six Features in Four Neurons ($1-S = 0.12$)

In this regime, we have one monosemantic neuron. The four polysemantic neurons implement an interesting code. One neuron seems to distinguish between important and unimportant features. Important features are then encoded as sets of two of the other three neurons, while unimportant features are represented by one neuron.



Eight Features in Five Neurons ($0.05 \leq 1-S \leq 0.08$)

In this regime, all neurons are polysemantic. The code is a kind of extension of the one described previously: one distinguishes important and unimportant features. Then either sets of three of the other neurons, or a single other neuron, are used to distinguish the specific feature.



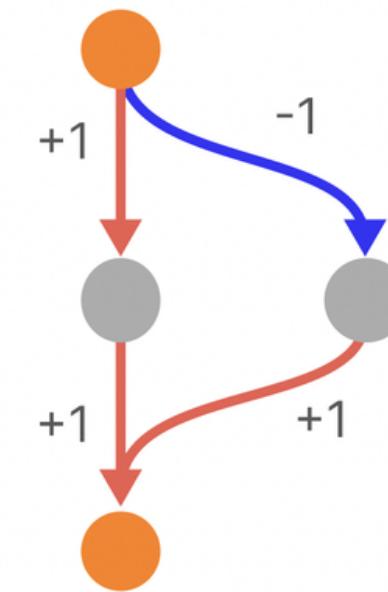
Features are Pairs of Neurons ($1-S \leq 0.04$)

In this regime, each feature simply corresponds to a *pair of neurons*.

Presumably if there were more features, increasing sparsity would eventually produce a dense binary code. But with only 10 features, this is the densest code that forms.

Section 7: Superposition in a Privileged Basis

Sparsity가 증가함에 따라 **monosemantic neuron**에서 **polysemantic neruon**으로의 전환이 있다는 것



Section 8: Computation in Superposition

Section 8: Computation in Superposition

지금까지는 신경망이 sparse feature를 superposition로 저장하고 이를 복원할 수 있다는 점을 보여주었다.

그러나 모델은 단순히 저장 방식으로 superposition을 사용하는 것이 아니라,
superposition 상태에서 완전히 계산을 수행할 수 있다고 생각한다.

Section 8: Computation in Superposition

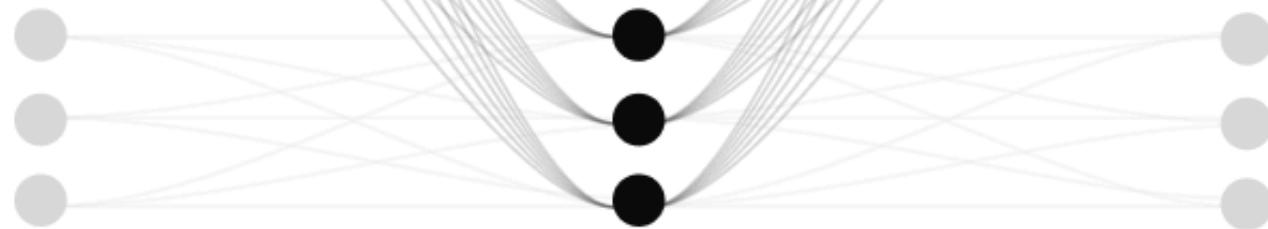
HYPOTHETICAL DISENTANGLLED MODEL



Can neural networks do useful computation in superposition, in addition to just representing features?

모델이 단순히 feature를 표현하는 것으로 superposition을 사용할 뿐 아니라, superposition 상태에서 **유용한 계산을 수행할 수 있을까?**

OBSERVED MODEL



We replace a single layer of an imagined larger network with a lower dimensional model and study what happens.

이번에는 간단한 비선형 함수를 계산하고, 그것이 superposition을 활용할 수 있는지를 탐구한다.

Section 8: Computation in Superposition > Experiment Setup

입력의 절댓값을 출력하는 간단한 비선형 함수

$$\text{abs}(x) = \text{ReLU}(x) + \text{ReLU}(-x)$$

$W_2^T W_1$ 을 연구할 수 없으며,

W 와 h 도 독립적으로 학습되기 때문에, W 의 열을 그냥
연구할 수 없다.

Following the previous section, we'll consider the "ReLU hidden layer" toy model variant, but no longer tie the two weights to be identical:

$$h = \text{ReLU}(W_1 x)$$

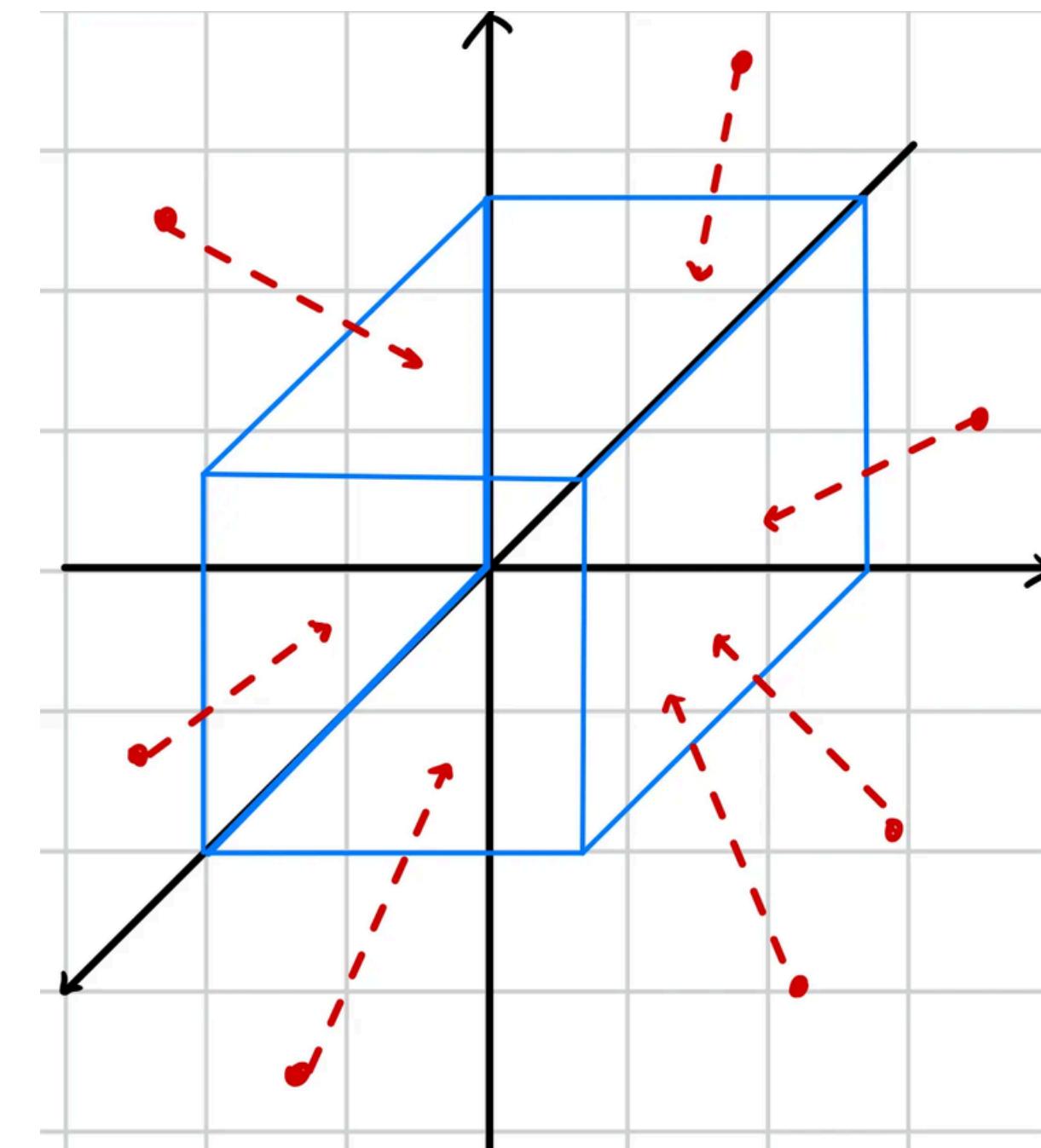
$$y' = \text{ReLU}(W_2 h + b)$$

The loss is still the mean squared error weighted by feature importances I_i as before.

Section 8: Computation in Superposition > Experiment Setup

입력의 절댓값을 출력하는 간단한 비선형 함수

$$\text{abs}(x) = \text{ReLU}(x) + \text{ReLU}(-x)$$



Section 8: Computation in Superposition > Basic Results

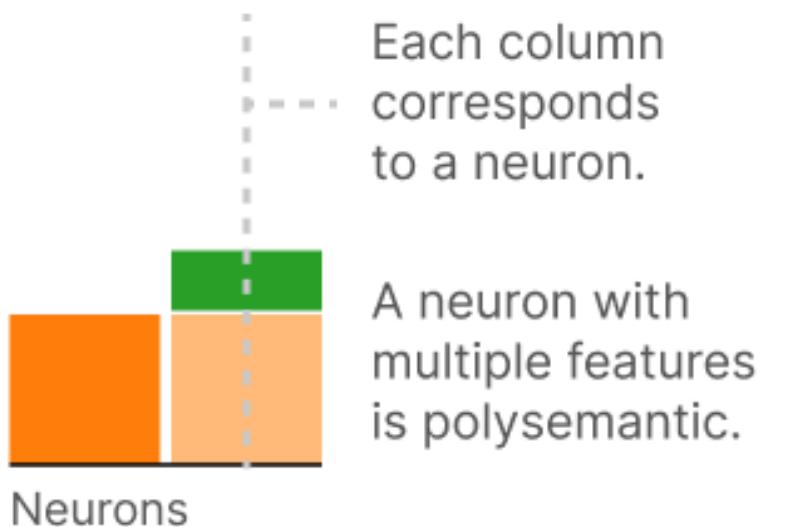
W as Matrix

Since the hidden layer now has a privileged basis we can visualize the raw weight matrix.



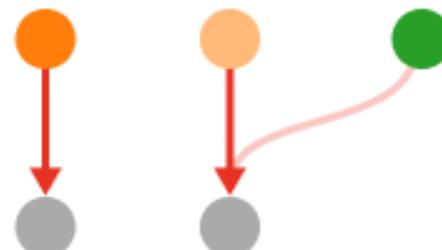
W as Stack Plot

Instead of showing a matrix, we can map features to colors and stack the weights per neuron.



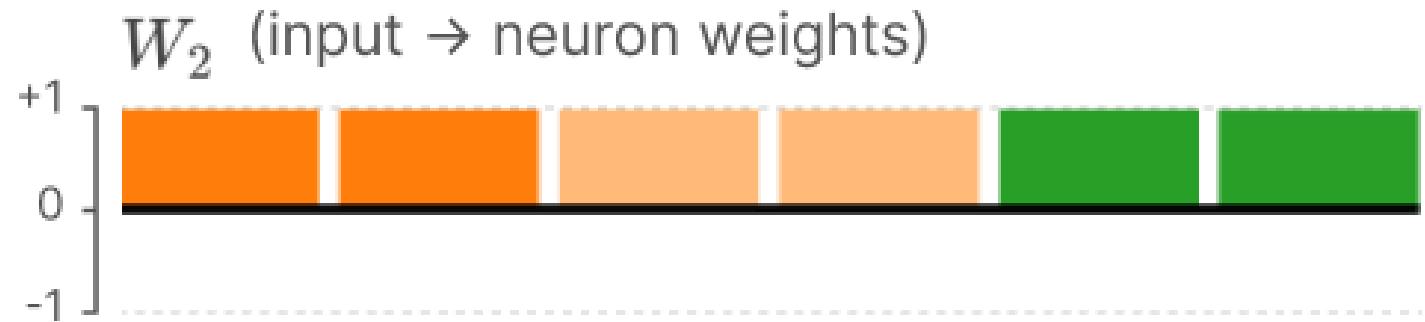
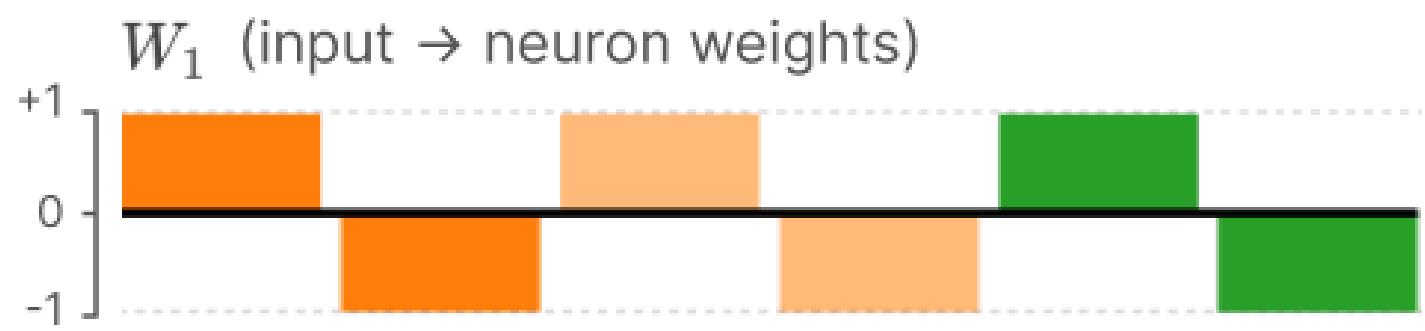
W as Graph

We can also visualize weights as the edges of a graph, as is often done for neural nets.

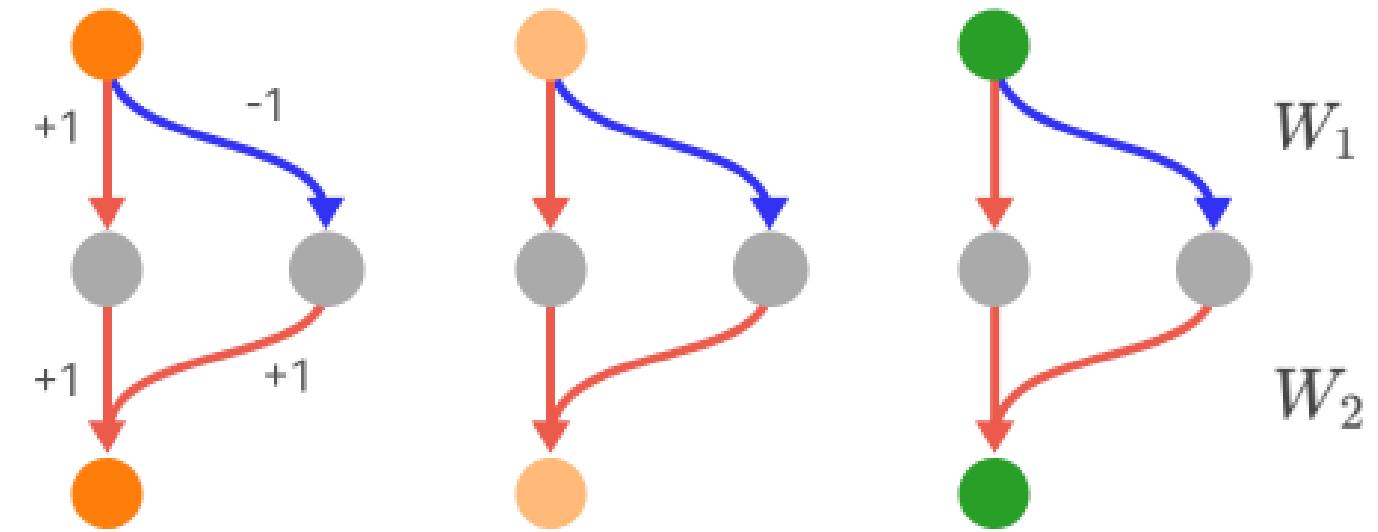


Section 8: Computation in Superposition > Basic Results

Raw Weights Visualization



Graph Visualization



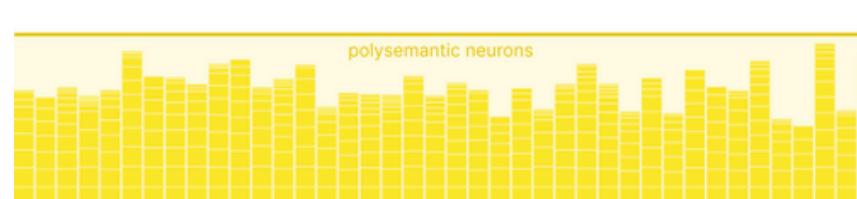
입력을 양의 영역과 음의 영역으로 나누어 계산하는 모습

Section 8: Computation in Superposition > Superposition vs Sparsity

ReLU Hidden Layer Toy Model on Absolute Value Task

$n = 100$; $m = 40$; $I_i = 0.8^i$

Neurons (sorted by importance of largest feature)

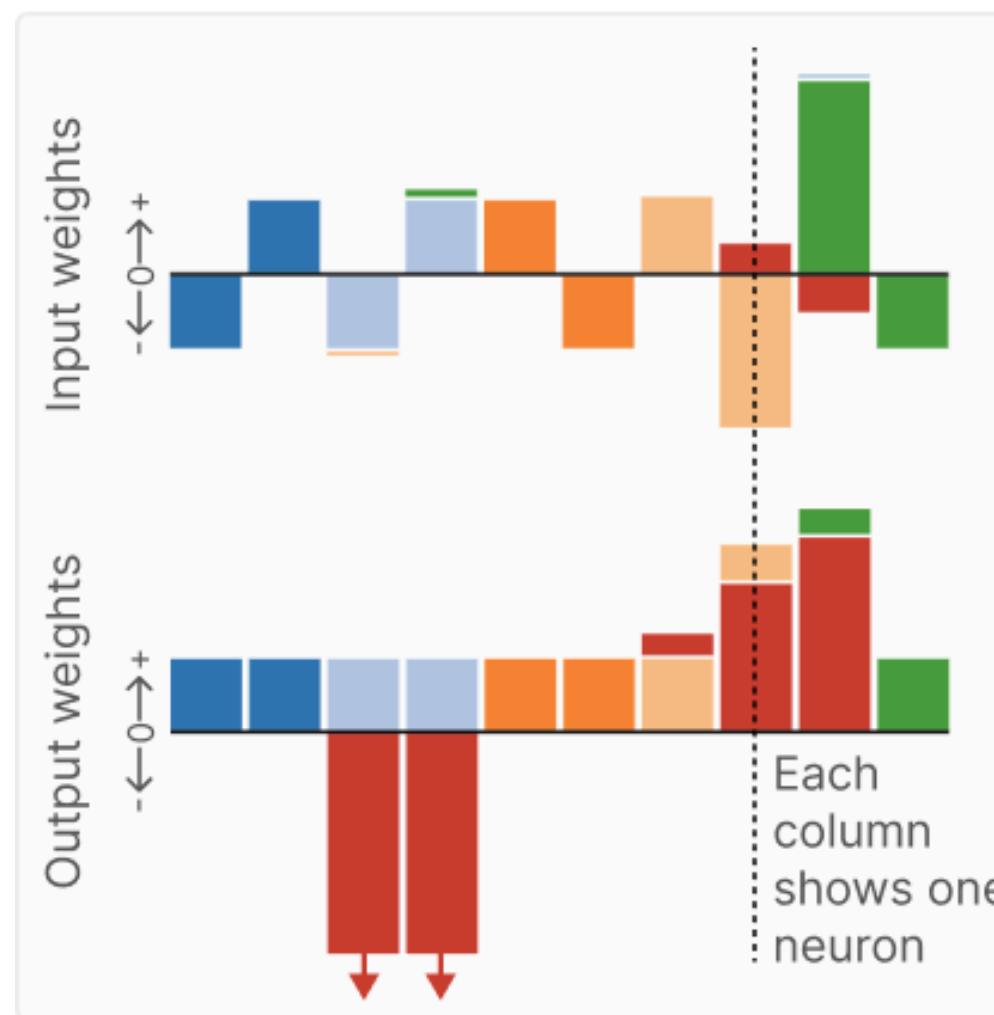


Neurons (sorted by importance of largest feature)

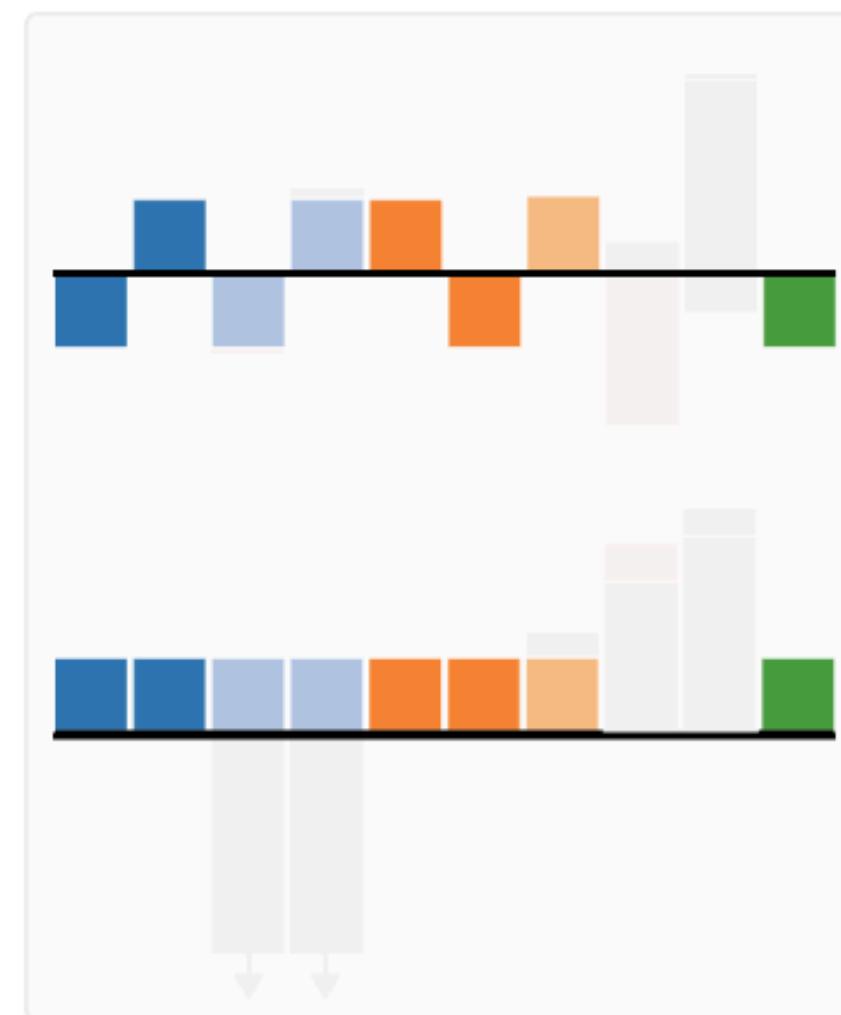


Section 8: Computation in Superposition > The Asymmetric Superposition Motif

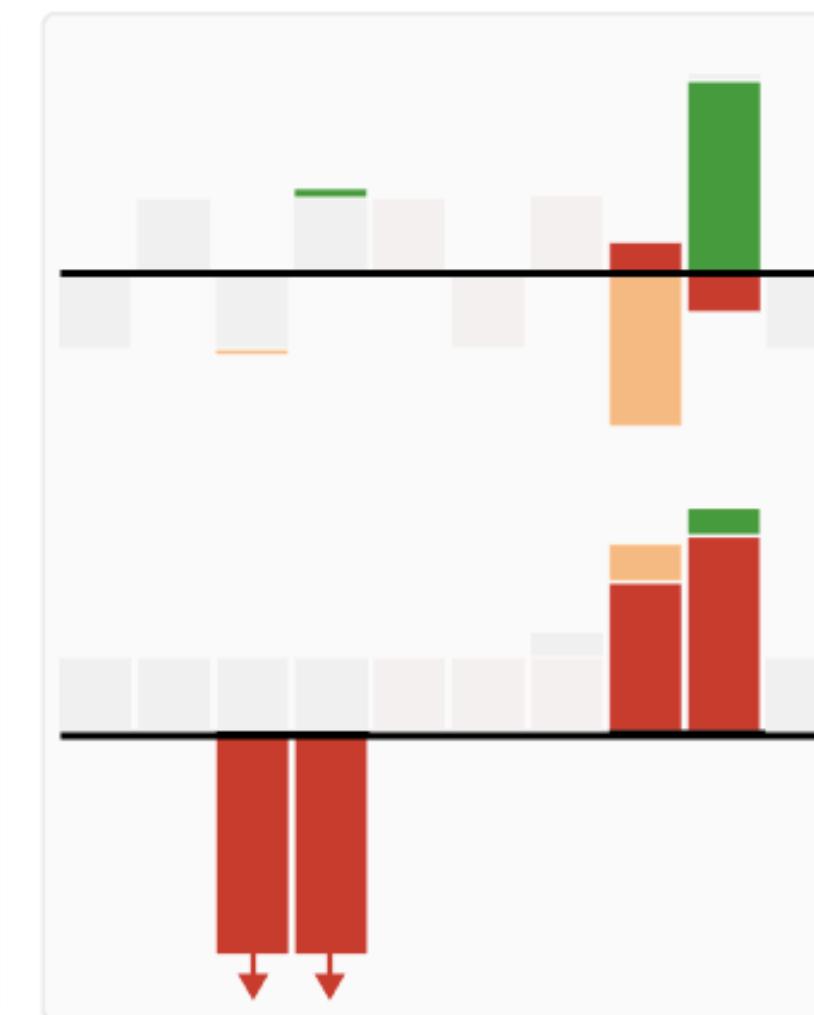
At first glance, this model is quite complicated and tricky to understand. However, we can (mostly) decompose it into two pieces...



Many weights are simply implementing absolute value, or a single side of absolute value, in the expected way.



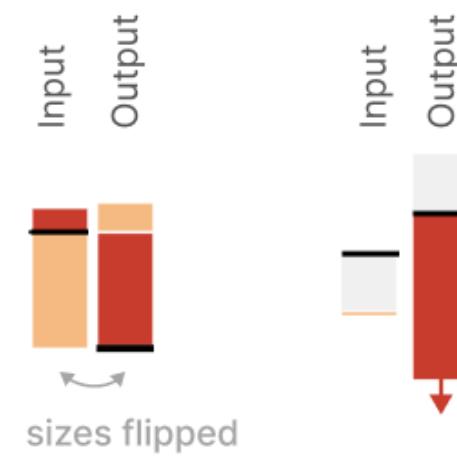
The main other thing is **asymmetric superposition with inhibition**. The model has two instances of this motif.



Section 8: Computation in Superposition > The Asymmetric Superposition Motif

Asymmetric Superposition with Inhibition Instance 1

Asymmetric Superposition Inhibition



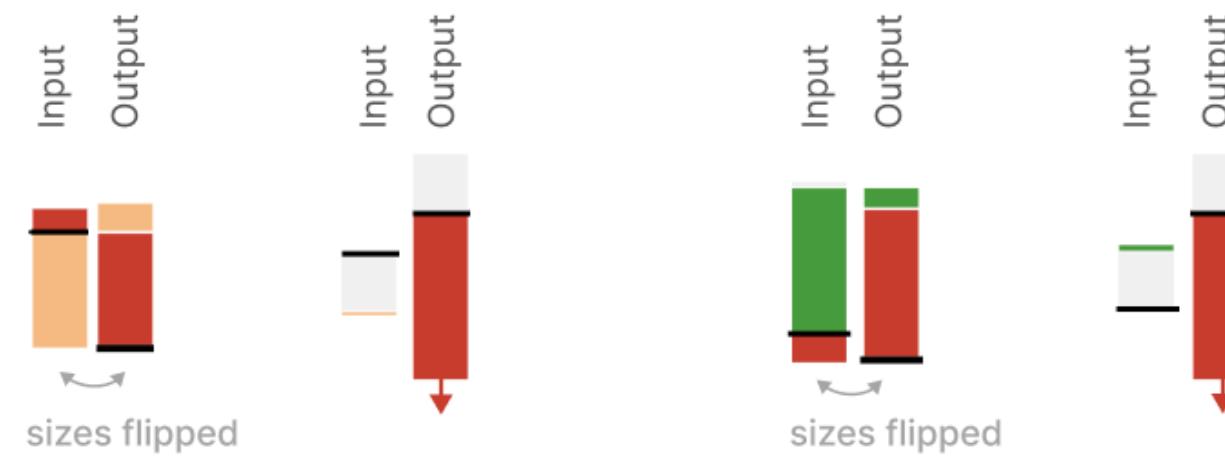
sizes flipped

One neuron represents two features and with *asymmetric superposition*. This causes to heavily interfere with , but not the reverse.



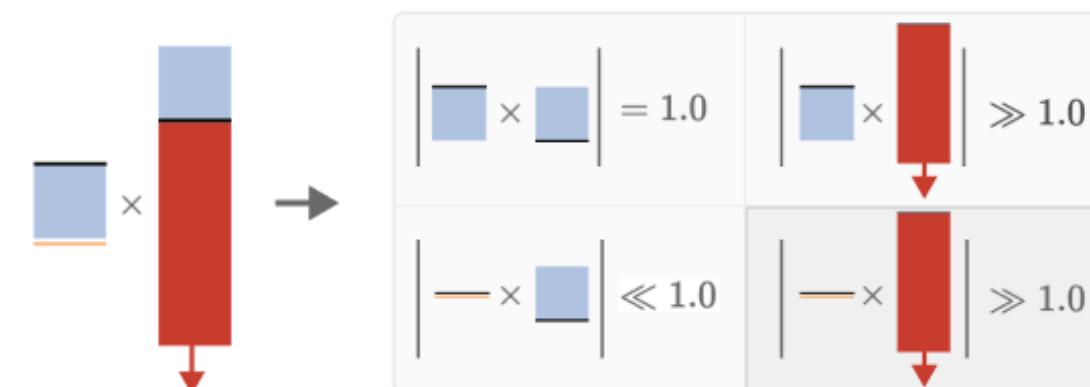
Asymmetric Superposition with Inhibition Instance 2

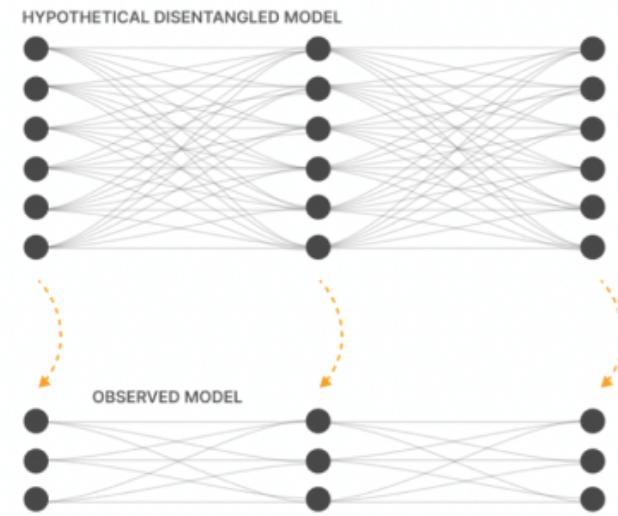
Asymmetric Superposition Inhibition



sizes flipped

Large amounts of positive interference are bad, so the model then puts a small amount of into a neuron and uses it to massively inhibit . This also forces the main feature the neuron is operating on () to inhibit .





Section 9: The Strategic Picture

Section 9: The Strategic Picture > Safety, Interpretability, & "Solving Superposition"

특징을 나열할 수 있는 능력을 부여하고, 동시에 활성화를 펼칠 수 있는 모든 방법을 "중첩의 해결책"이라고 부르기로 하자.

Decomposing Activation Space

해석 가능성의 가장 기본적인 도전 과제는 차원의 저주를 극복하는 것

기계적 해석 가능성의 경우, 이는 궁극적으로 활성화 공간을 독립적으로 이해할 수 있는 구성 요소로 분해할 수 있는지 여부로 축소된다.

Describing Activations in Terms of Pure Features

Activation들을 순수한 특징에 따라 설명할 수 없다는 점이 문제이다.

Understanding Weights (ie. Circuit Analysis)

신경망의 가중치는 일반적으로 이해할 수 있는 특징들을 연결할 때만 이해될 수 있다.

Even very basic approaches become perilous with superposition

아주 간단한 접근이더라도, 관계 없는 특징들이 서로 긍정적인 내적 곱을 가지게 되기 때문에, superposition은 코사인 유사성이 오해를 불러일으킬 가능성이 있다는 것을 의미

Section 9: The Strategic Picture > Three Ways Out

1. Create models without superposition

숨겨진 층의 활성화에 L1 정규화 항을 적용하기만 하면 superposition을 없앨 수 있다.

하지만 모델들이 superposition을 통해 상당한 이점을 얻고 있기 때문에, 고려해봐야 한다.

2. Find an overcomplete basis

Non-superposition 모델을 만드는 반대 전략은 superposition이 있는 일반 모델을 가져와서 그 특징이 어떻게 후에 embedding 되는지를 설명하는 overcomplete basis를 찾는 것

3. Hybrid approaches

Section 9: The Strategic Picture > Three Ways Out

1. Create models without superposition

Section 9: The Strategic Picture > Three Ways Out

2. Find an overcomplete basis

that describes how features are represented in models with superposition.

- It's no longer easy to know how many features you have to enumerate
- Solutions are no longer integrated into the surface computational structure
- It's a different, major engineering challenge

Section 9: The Strategic Picture > Three Ways Out

3. Hybrid approaches

in which one changes models, not resolving superposition, but making it easier for a second stage of analysis to find an overcomplete basis that describes it.

Discussion

Does this occur
in real models?

Open Questions

Section 10: Discussion

Section 10: Discussion > Does this occur in Real Models?

Polysemantic neurons exist

polysemantic neuron은 존재한다.

Neurons are sometimes "cleanly interpretable" and sometimes "polysemantic", often in the same layer

뉴런은 때때로 "명확히 해석 가능"하고 때때로 "다의적"이며, 종종 같은 layer에서 나타난다.

InceptionV1 has more polysemantic neurons in later layers

고차원 특성이 될수록 탐지되는 자극이 더 드물고 따라서 더 희소해지기 때문에, 다의성인 뉴런의 비율은 깊이가 증가함에 따라 증가

Early Transformer MLP neurons are extremely polysemantic

Transformer 언어 모델의 첫 번째 MLP 층의 뉴런은 종종 매우 다의적이다.

첫 번째 MLP 층의 목표가 동일한 토큰의 다양한 해석(예: 영어, 독일어, 네덜란드어, 아프리칸스어에서 "die")을 구별하는 것이라면, 이러한 특성은 매우 희소하여 많은 다의성을 갖게 될 것으로 예측된다.