D. Representation Learning

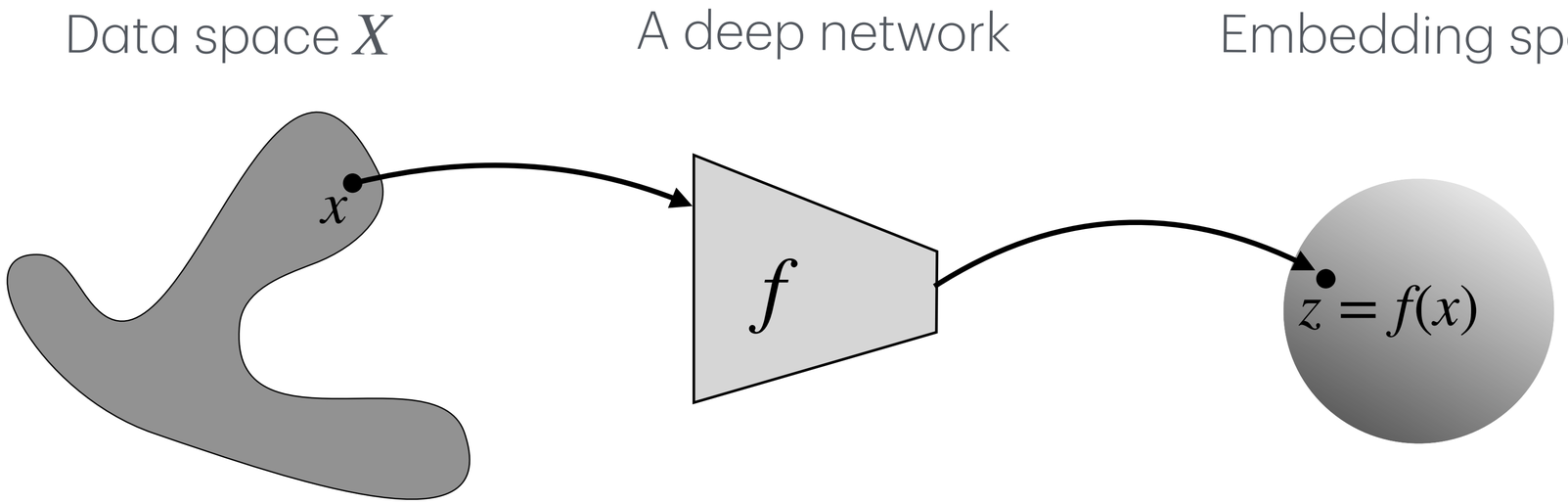
D.1 Representation Learning: Learning useful coordinates for data

D.2 Representation Learning: Evaluating and probing Representations

D.3 Representation Learning: How to learn a good Representation

Learning as a Mapping from Data to Representations

- Let's look at a deep neural network form a specific point of view:
  - Takes an input x
  - Transforms it through multiple layers
  - Builds a latent representation z
  - Uses z to perform a task



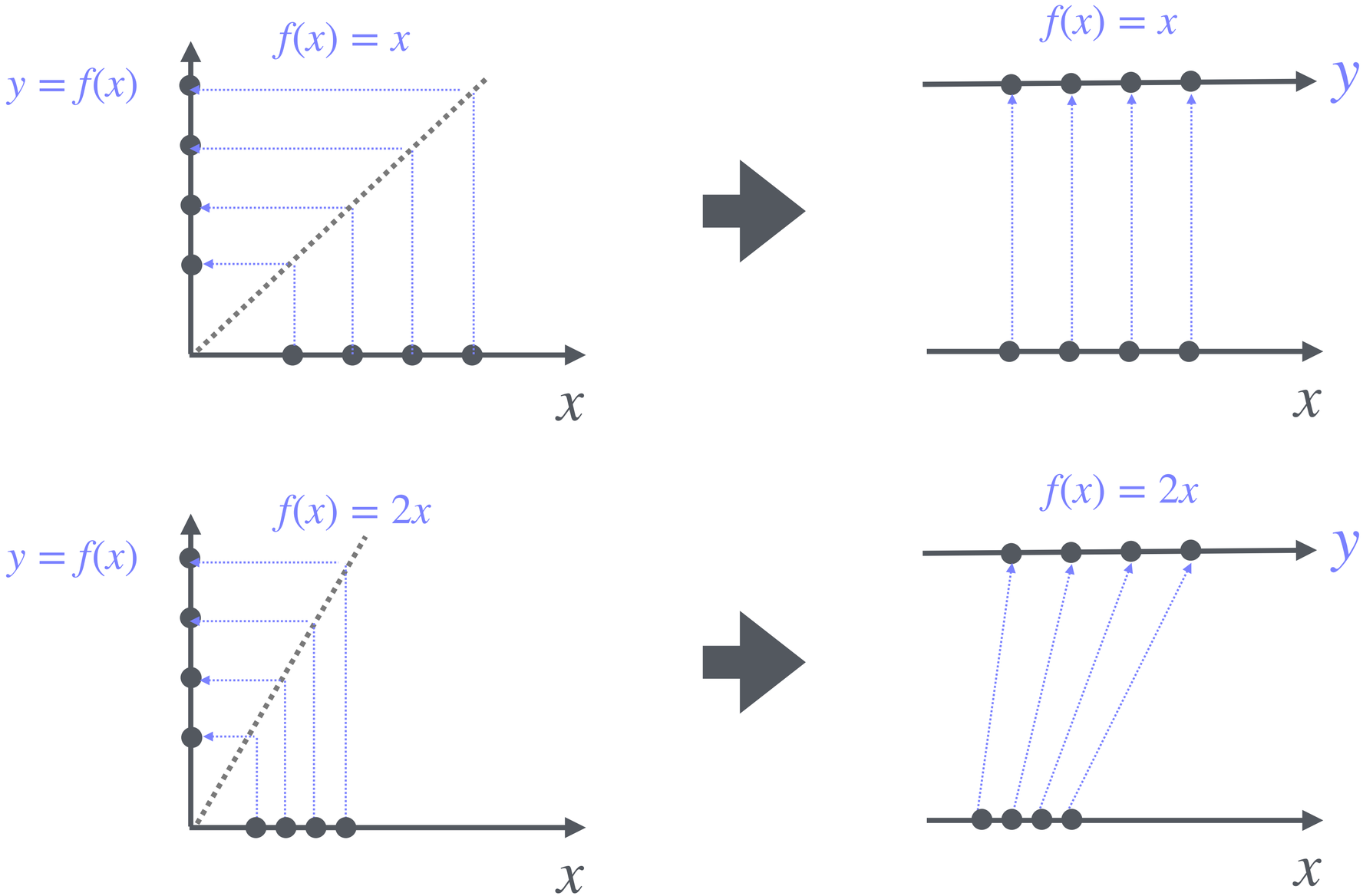
Data space:
Complex
Entangled
Mixture: useful and irrelevant information

Model function:
Converts coordinates of data
Finds a useful map

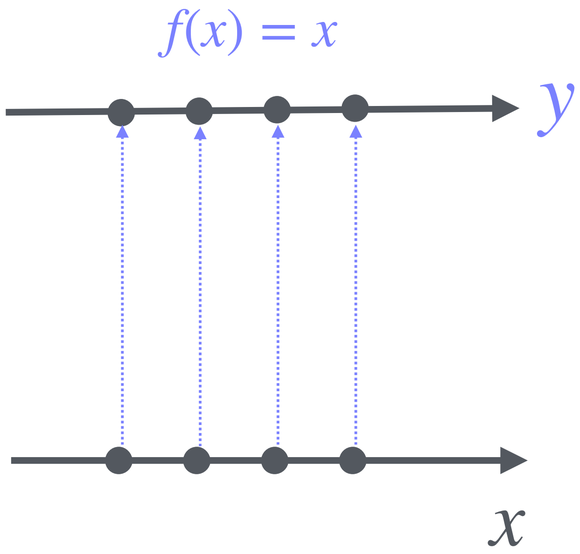
Representation space (embedding)
Usually simpler
Hopefully dis-entangled
Useful for the task

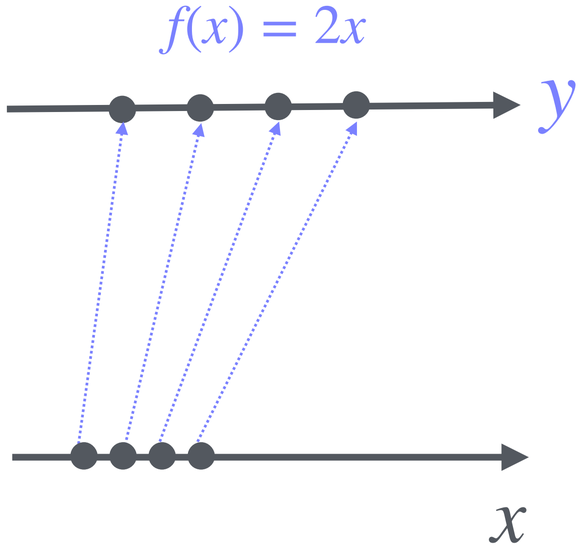
Inspired by Phillip Isola's book: https://visionbook.mit.edu/representation_learning.html

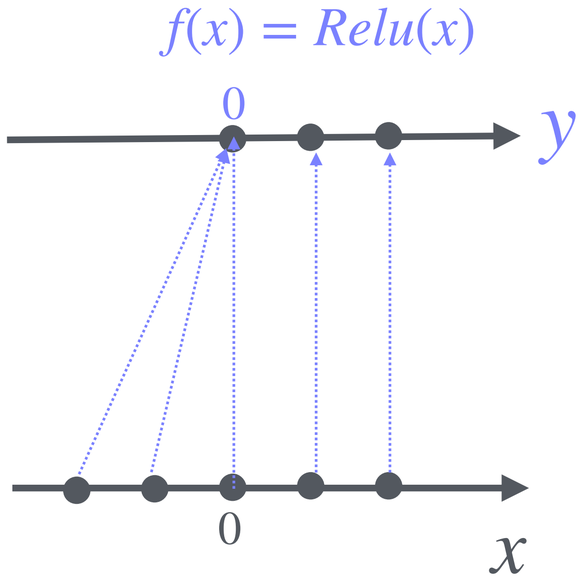
Representing functions as mapping from one space to another:

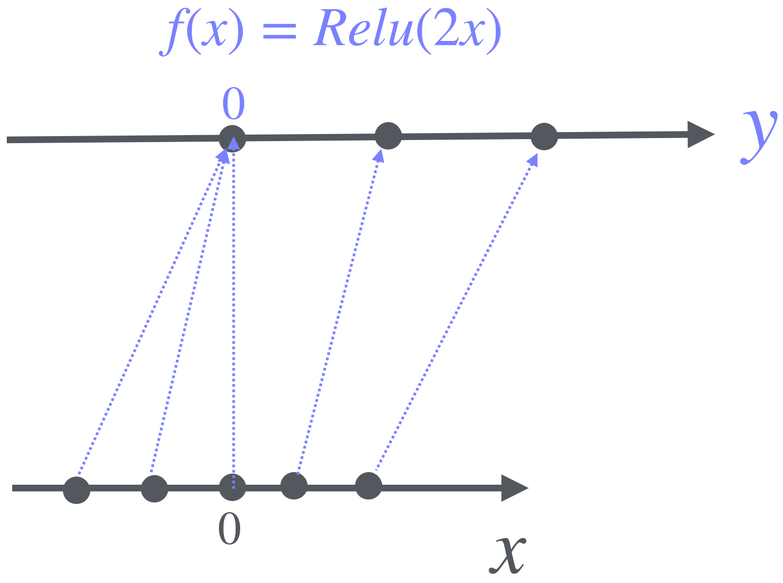


Representing functions as mapping from one space to another:





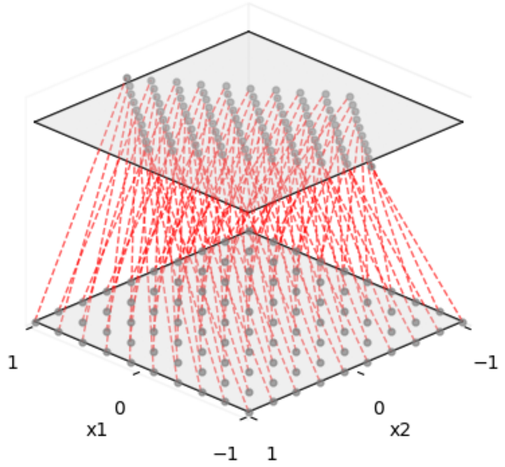




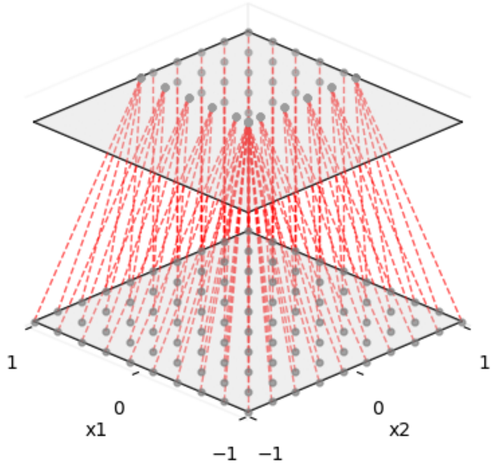
Representing functions as mapping from one space to another:

- This stays the same for function which map higher dimensions to each other.
- In cased of 2D, let's look at some useful maps to help our intuitive thinking:

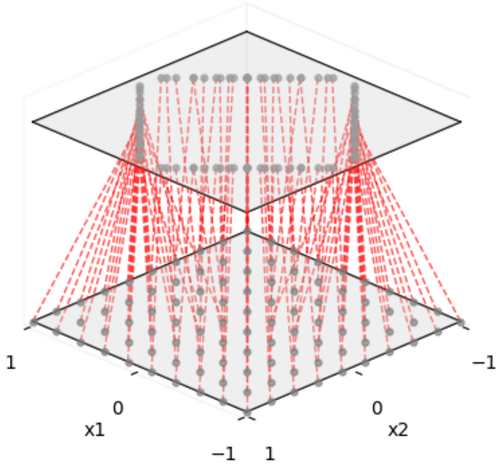
$Y = Linear(X)$



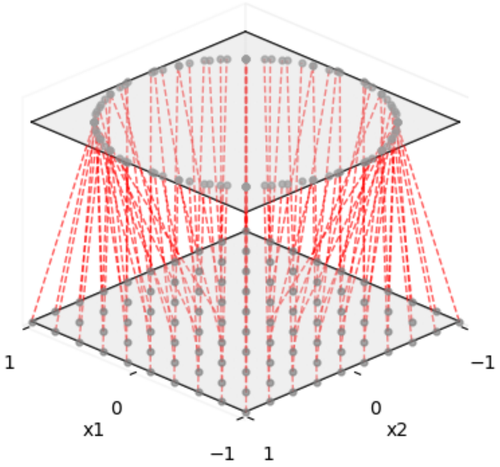
$Y = Relu(X)$



$Y = normalized_{L1}(X)$



$Y = normalized_{L2}(X)$



Representing functions as mapping from one space to another:

- This stays the same for function which map higher dimensions to each other.
- In cased of 2D, let's look at some useful maps to help our intuitive thinking:

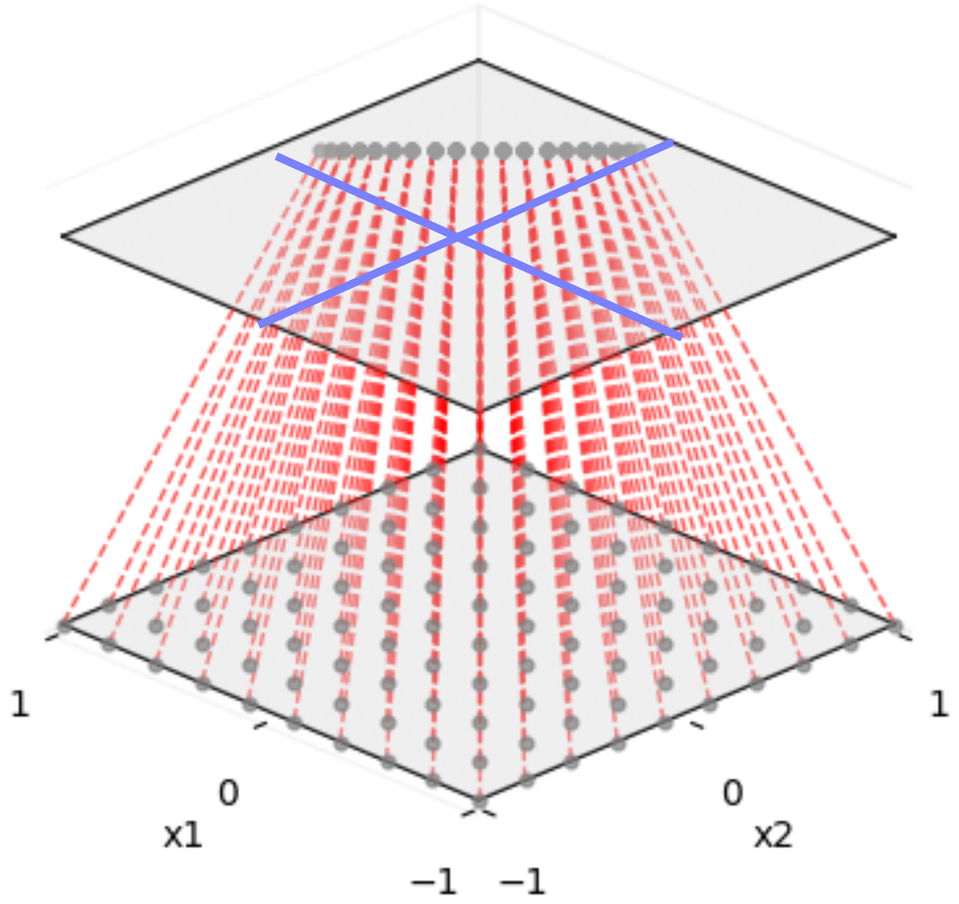
$$Y_i = softmax(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{d} e^{x_j}}$$

- In 2D (d=2), for each point, $x = (x_1, x_2)$:

$$y_1 = \frac{e^{x_1}}{e^{x_1} + e^{x_2}}$$

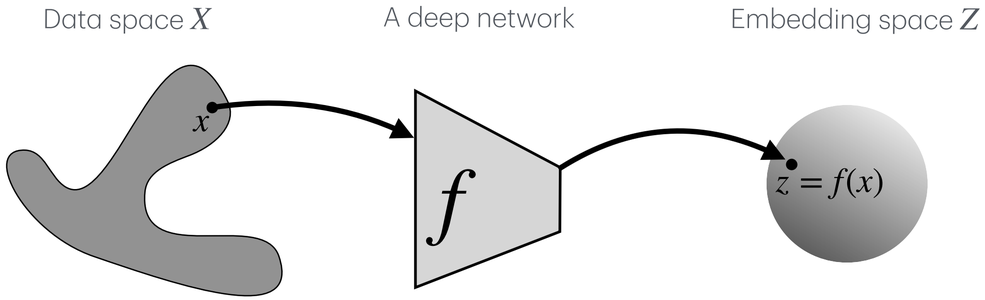
$$y_2 = \frac{e^{x_2}}{e^{x_1} + e^{x_2}}$$

$$y_1 + y_2 = 1$$

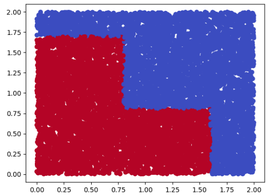


Deep neural network as mapping from one space to another

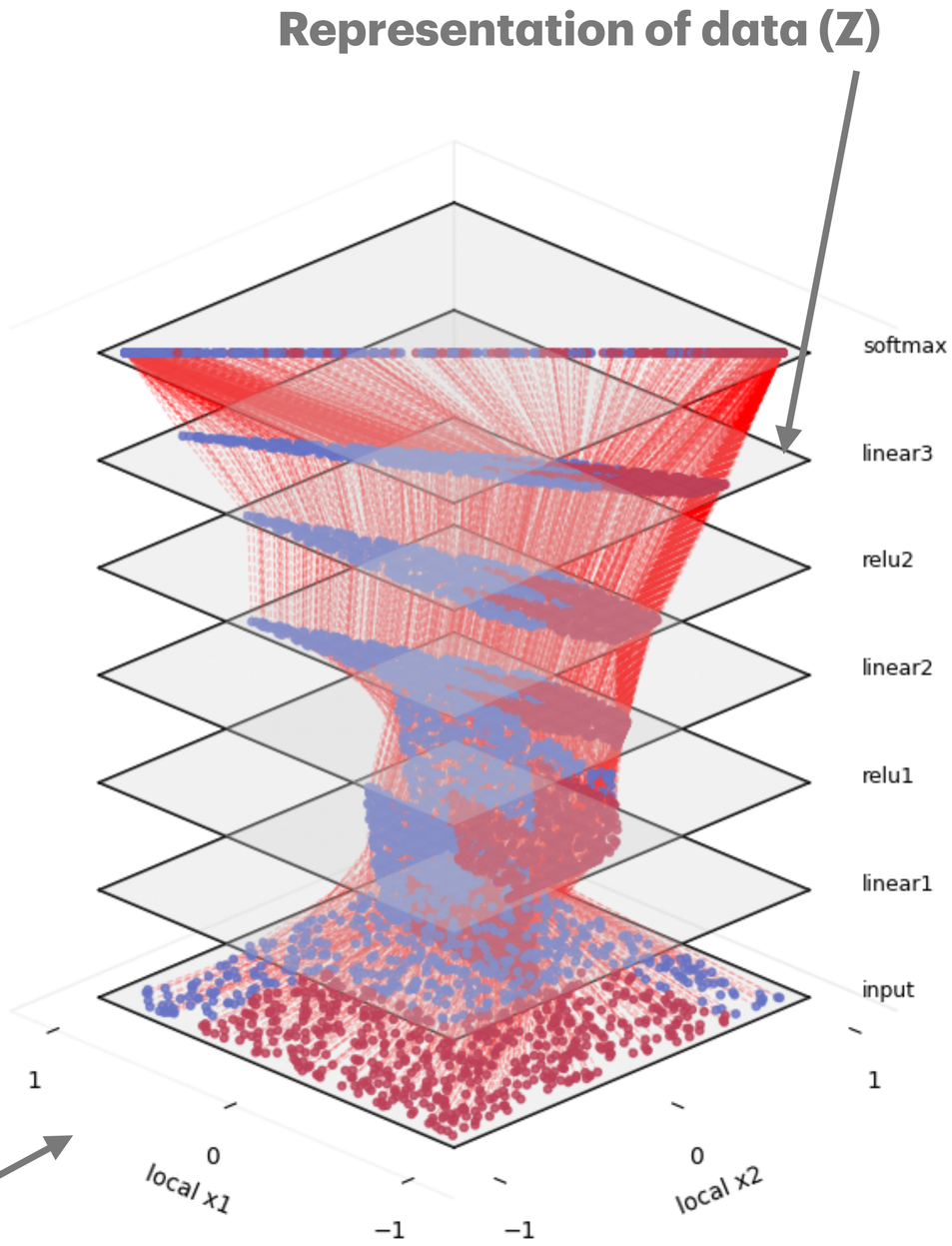
- We saw some example of how simple individual functions maps input space (X) to output space (Y)



- A toy illustrative example: an MLP with hidden dimension (d_h=2) transforming a 2D points into a classification:



$linear$ $Relu$ $linear$ $Relu$ $linear$ $softmax$



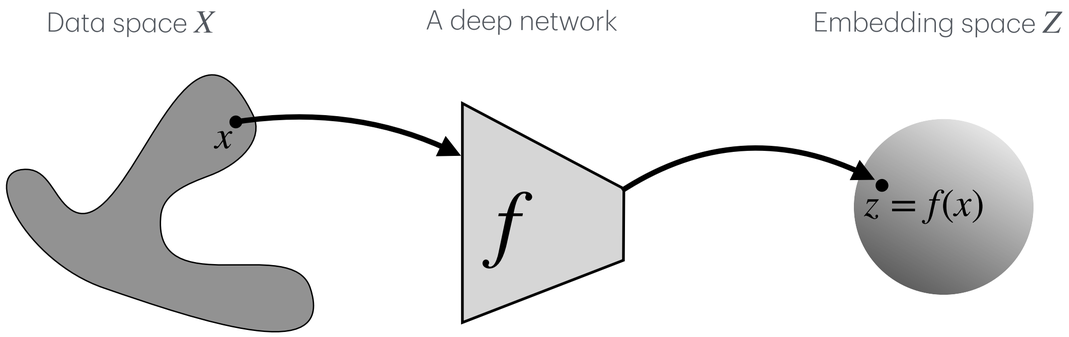
What is a Representation?

- A representation is a way of describing data.
- In this course, we focus on vector embeddings as representations.
- An encoder maps each input (vector) into a vector space:

$$f_\theta : \mathcal{X} \to \mathbb{R}^d$$

=For one data point x, its representation is: $z = f_\theta(x), \qquad z \in \mathbb{R}^d$

- So, a representation is a new coordinate description of the original input. Ideally this new description makes downstream tasks easier.
- Tasks such as: *classification, prediction, comparison, clustering, or generation*



Locatello, Francesco, et al. PMLR, 2019.
Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35.8 (2013): 1798-1828.

Representation Learning and Manifold Hypothesis

- So far we saw via example(s) that:

Representation: mapping from complex data space → simpler embedding space

- But, why should we expect a simpler representation to even exist? It is hypothetical:

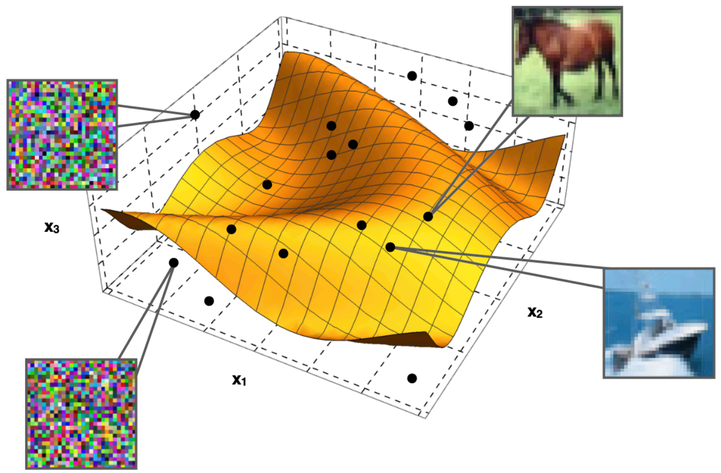
> *"...the idea that the dimensionality of many data sets is only artificially high; though each data point consists of perhaps thousands of features, it may be described as a function of only a few underlying parameters.*
>
> *That is, the data points are actually samples from a low-dimensional manifold that is embedded in a high-dimensional space."*
>
> Cayton, "Algorithms for manifold learning", UCSD Tech. Rep., 2005.

The manifold hypothesis: Is a widely accepted tenet of machine learning which asserts that nominally high- dimensional data are in fact concentrated near a low-dimensional manifold, embedded in high-dimensional space

Whiteley, Nick, Annie Gray, and Patrick Rubin-Delanchy., *Statistical Methodology* (2026)
Cayton, Lawrence. *Univ. of California at San Diego Tech.*(2005):

Manifold Hypothesis

- High-dimensional data concentrates near low-dimensional manifolds
  - CIFAR10: 3072 dimensions → intrinsic dimension ≈ 35
  - MNIST: 784 dimensions → intrinsic dimension ≈ 14
- Labels depend on manifold position, not ambient coordinates
- Why this matters: justifies the "simpler embedding" goal
- Makes learning tractable despite high nominal dimensionality



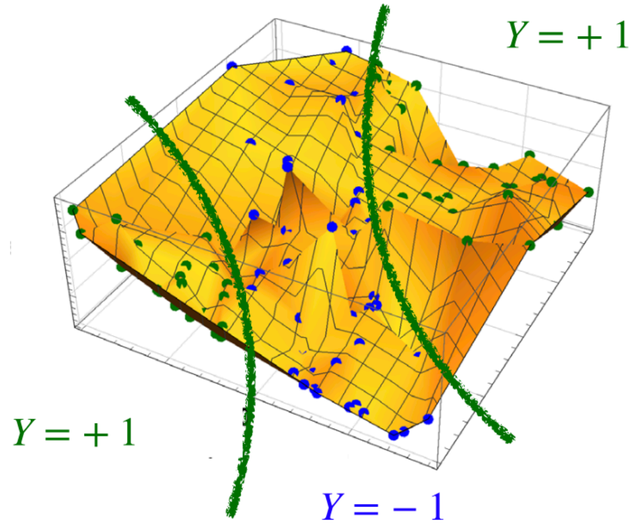


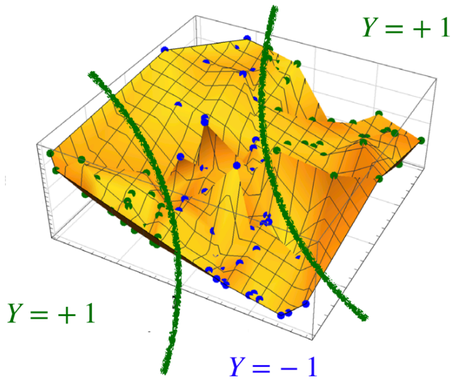
Illustration of the notion of a hidden manifold in input space using CIFAR10 example images

Goldt, Sebastian, et al. "Modeling the influence of data structure on learning in neural networks: The hidden manifold model." *Physical Review X* 10.4 (2020): 041044.

Manifold Hypothesis

Note: A very hand-wavy way of estimating dimenssionality of manifolds in the representation space is via looking at nearest neighbors. If points lie locally on a d dimensional manifold, then the number of nearby points within radius $r$scales like:

$$\#\{neighborswithinr\} \propto r^d$$



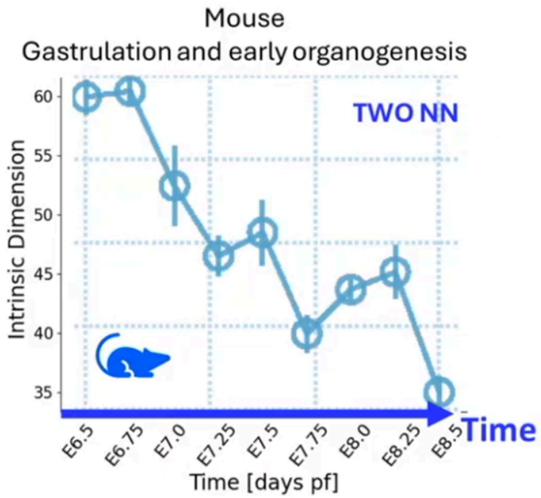
Other more rebust methods also exist:

Levina–Bickel maximum-likelihood estimator, etc

https://www.nature.com/articles/s41598-017-11873-y

https://scikit-dimension.readthedocs.io/en/latest/

https://dadapy.readthedocs.io/en/latest/



Application in biology: *Biondo, Marta, et al. "The intrinsic dimension of gene expression during cell differentiation." Nucleic Acids Research 53.16 (2025): gkaf805.*

Representation Learning and Manifold Hypothesis

lets look at the setup one more time:

- A data point lives in an input space:

  $x \in \mathcal{X} \subseteq \mathbb{R}^{D_{input}}$

  Where D_{input} determined by how the data is structured. for example: image height × width × channels.

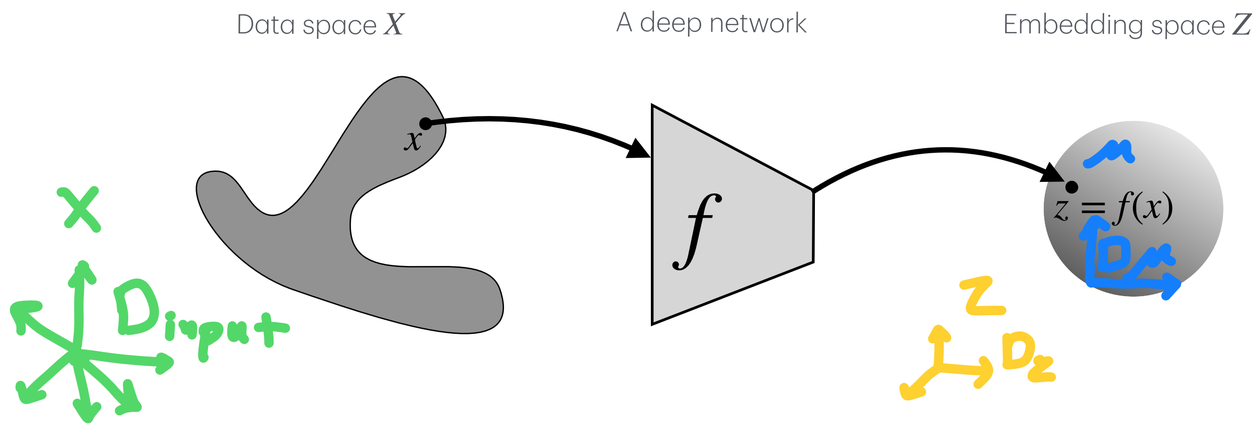
- A neural network maps the input into a latent space:

  $z = f_\theta(x), z \in \mathcal{Z} \subseteq \mathbb{R}^{D_z}$

  where D_z is determined by the architecture. for example: the width of a hidden layer or embedding layer.

- The manifold hypothesis says that although data may live in a high-dimensional space, real samples often occupy a much smaller structured region. So the observed representations may lie near a lower-dimensional manifold:

  $\mathcal{M}_z \subset \mathbb{R}^{D_z}, \qquad d_{\mathcal{M}} \ll D_{input}$



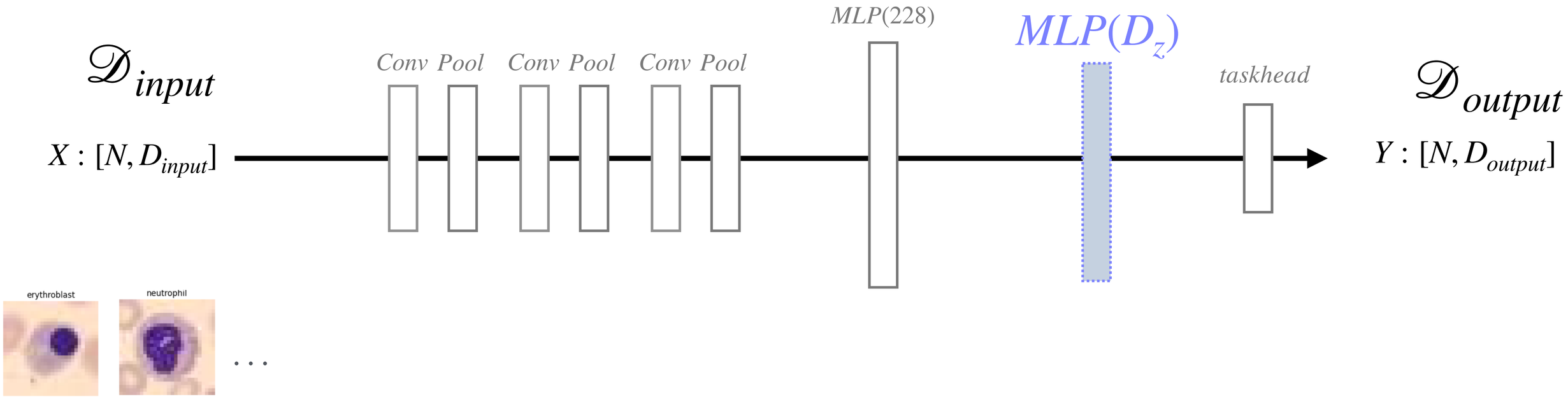
Representation Learning and Manifold Hypothesis

- Representation learning tries to reshape the data so this hidden structure becomes more useful for downstream tasks.

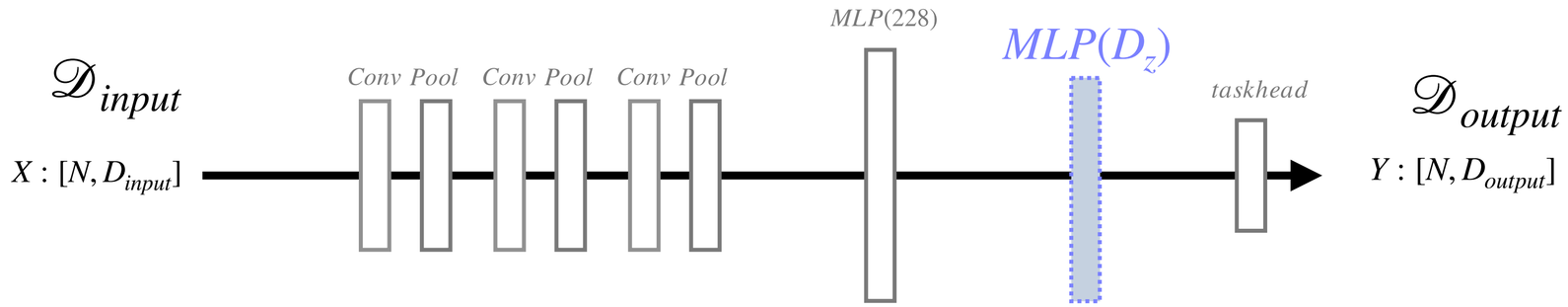
What affects the learned representation?

1. **Data:** Natural structure, noise, labels, augmentations
2. **Architecture:** CNN locality, pooling, weight sharing, depth, bottlenecks
3. **Training:** Loss function, labels, optimizer, regularization

We can try to investigate this in a simple toy setup:



Representation Learning and Manifold Hypothesis



Toy setup :

- We use a small CNN encoder to map images into a penultimate embedding:

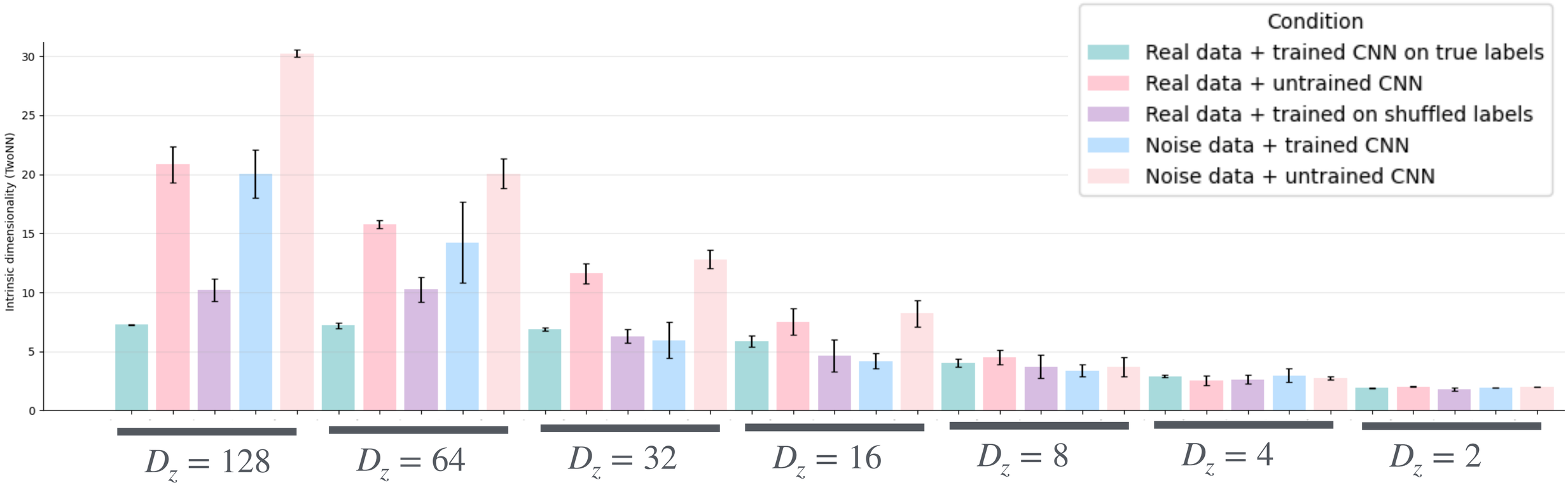
$$x \in \mathbb{R}^{D_{input}} \rightarrow z \in \mathbb{R}^{D_z} \rightarrow \hat{y}$$

  - D_{input} is fixed by the data format: image height × width × channels.
  - D_z is fixed by the architecture but we sweep the penultimate width D_z =64,32,16,8,4,2.
- For each condition, we extract the penultimate representation z and estimate: intrinsic dimensionality of z linear probe accuracy from z.
- We want to ask: *How do data, architecture, and training change the geometry of representations?*
- *We compare five conditions:*

1. ***Real data + untrained CNN*** *Measures the effect of architecture alone.*
2. ***Real data + trained on true labels*** *Measures architecture + real data + meaningful supervision.*
3. ***Real data + trained on shuffled labels*** *Keeps the images real, but breaks the semantic label relationship.*
4. ***Noise data + untrained CNN Removes*** *real image structure while keeping the same model architecture.*
5. ***Noise data + trained CNN*** *Tests whether training can create apparent structure even from weak/random data.*

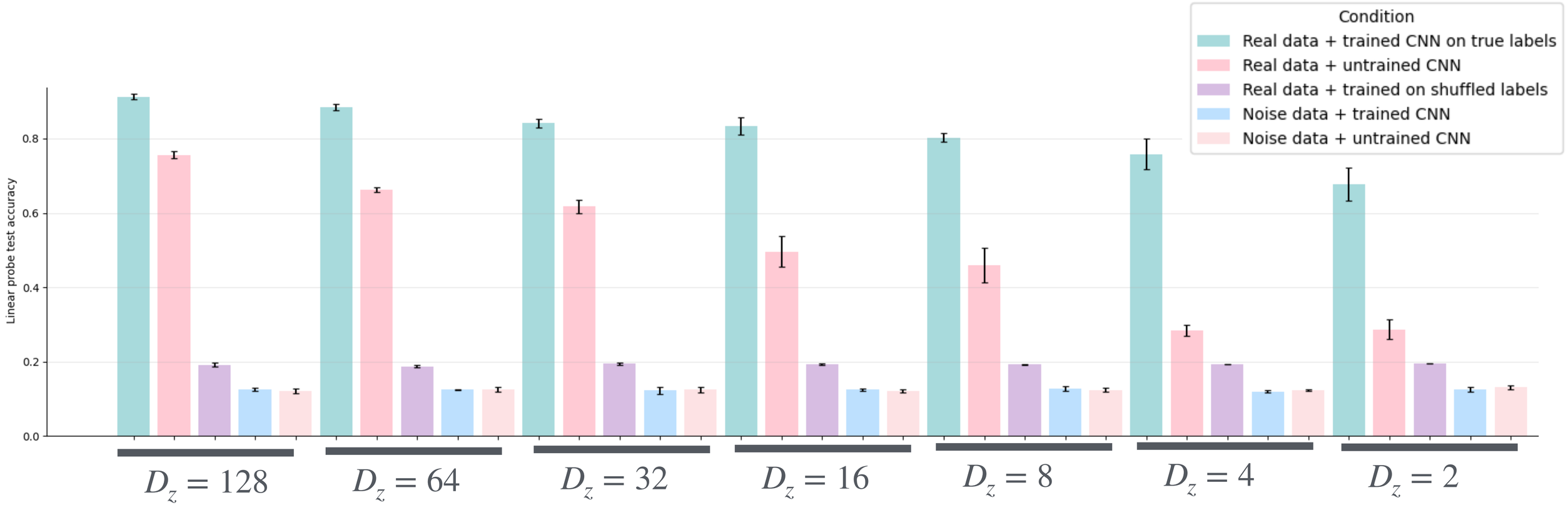
Representation Learning and Manifold Hypothesis

1. ***Real data + trained CNN on true labels*** *Measures architecture + real data + meaningful supervision.*
2. ***Real data + untrained CNN*** *Measures the effect of architecture alone.*
3. ***Real data + trained on shuffled labels*** *Keeps the images real, but breaks the semantic label relationship.*
4. ***Noise data + trained CNN*** *Tests whether training can create apparent structure even from weak/random data.*
5. ***Noise data + untrained CNN Removes*** *real image structure while keeping the same model architecture.*



Representation Learning and Manifold Hypothesis

1. ***Real data + trained CNN on true labels*** *Measures architecture + real data + meaningful supervision.*
2. ***Real data + untrained CNN*** *Measures the effect of architecture alone.*
3. ***Real data + trained on shuffled labels*** *Keeps the images real, but breaks the semantic label relationship.*
4. ***Noise data + trained CNN*** *Tests whether training can create apparent structure even from weak/random data.*
5. ***Noise data + untrained CNN Removes*** *real image structure while keeping the same model architecture.*



Takeaway: Even the dimensionality of the learned representation is not determined by the architecture alone. It emerges from the interaction between the data distribution, the architecture, and the training objective.

What makes a representation good?

A good representation makes a downstream task easier. (Goodfellow et al. 2016)

Broadly speaking, it is:

1. **Sufficient**: keeps task-relevant information
2. **Compact**: removes irrelevant variation
3. **Structured**: similar examples are nearby
4. **Separated**: task-relevant groups are easier to distinguish
5. **Robust**: ignores irrelevant perturbations, but preserves meaningful differences

Next we will look at methods to evaluate the learned representation