



Neural Network - Autoencoder

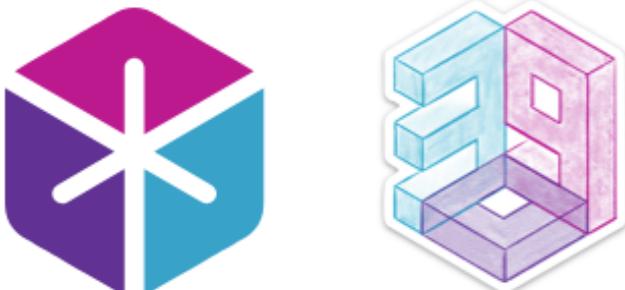
Rina BUOY, PhD



ChatGPT 4.0

Disclaimer

Adopted from



6.390

Introduction to Machine Learning (Fall 2024)

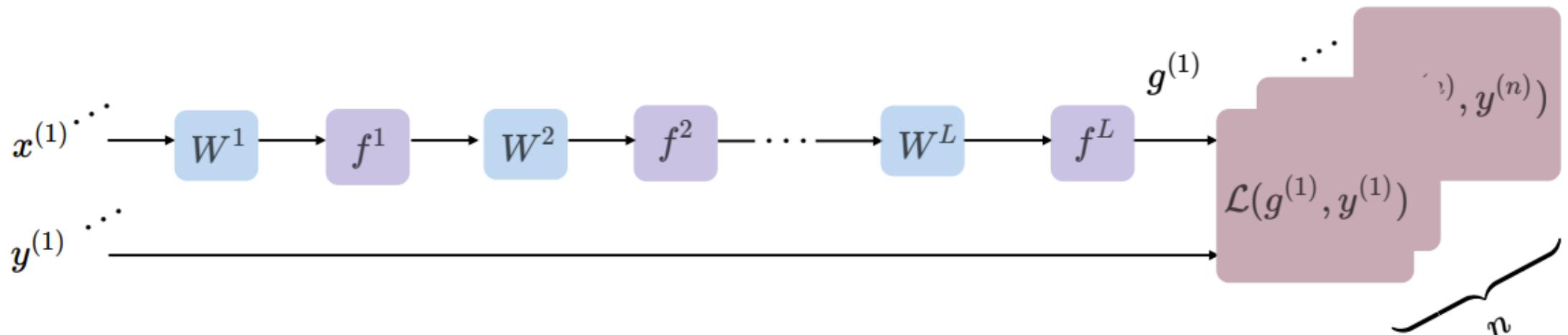
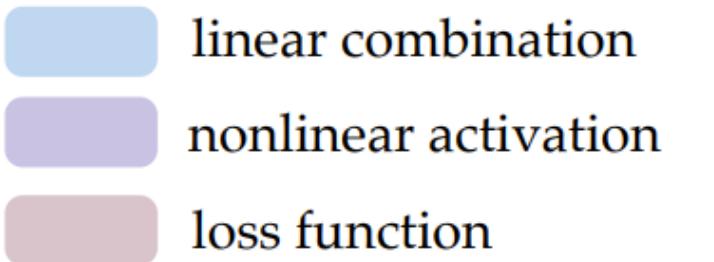
<https://introml.mit.edu/fall24>

Outline

- Recap, neural networks mechanism
- Neural networks are *representation* learners
- Auto-encoder:
 - Bottleneck
 - Reconstruction
- Unsupervised learning
- (Some recent representation learning ideas)

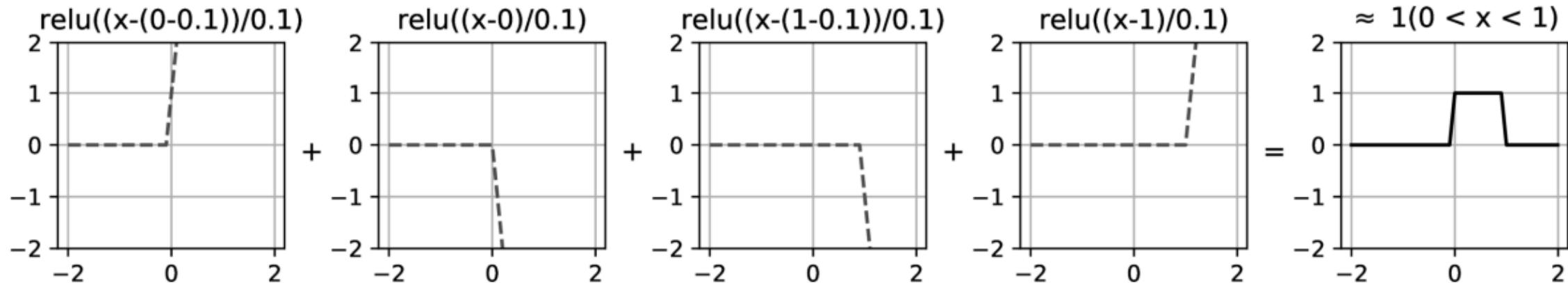
Recap:

Forward pass: evaluate, *given* the current parameters,

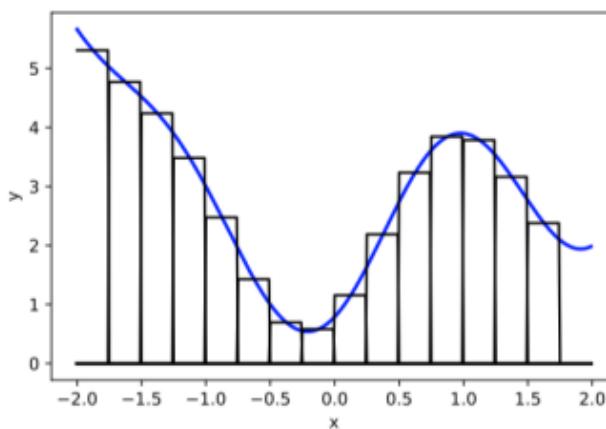


- the model output $g^{(i)} = f^L (\dots f^2 (f^1(\mathbf{x}^{(i)}; \mathbf{W}^1); \mathbf{W}^2); \dots \mathbf{W}^L)$
- the loss incurred on the current data $\mathcal{L}(g^{(i)}, y^{(i)})$
- the training error $J = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(g^{(i)}, y^{(i)})$

compositions of ReLU(s) can be quite expressive

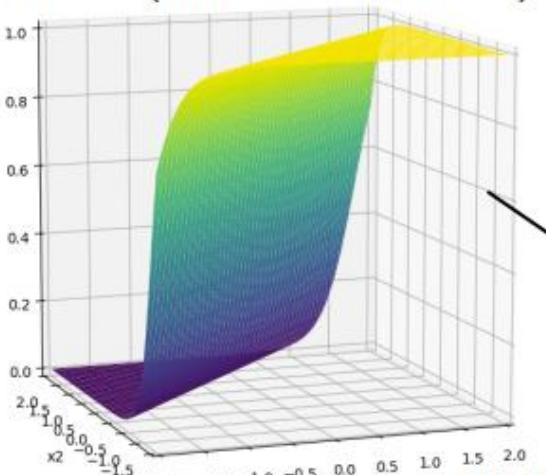


in fact, asymptotically, can approximate any function!



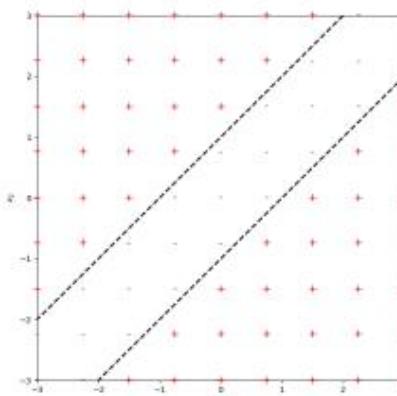
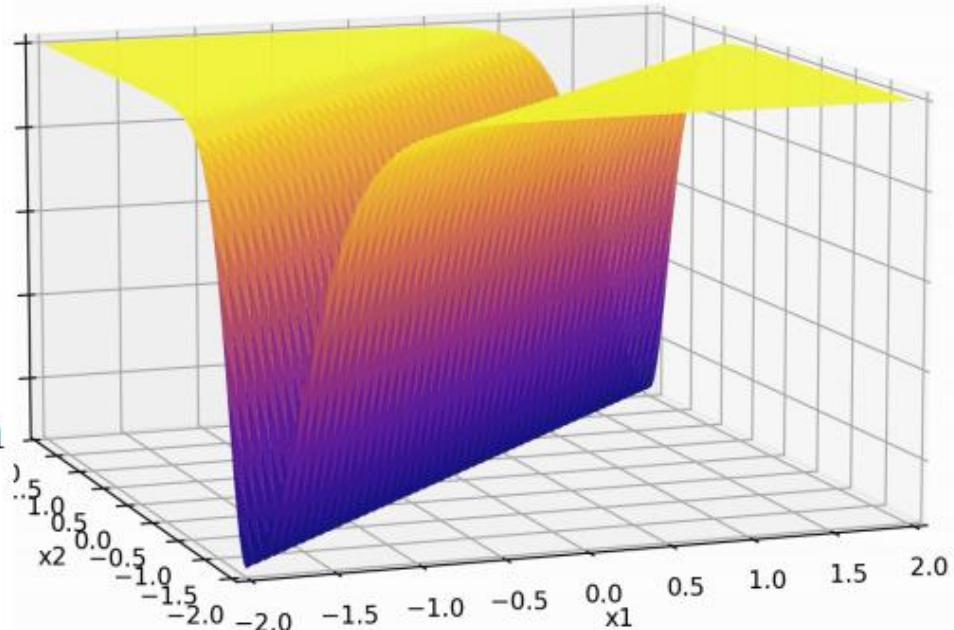
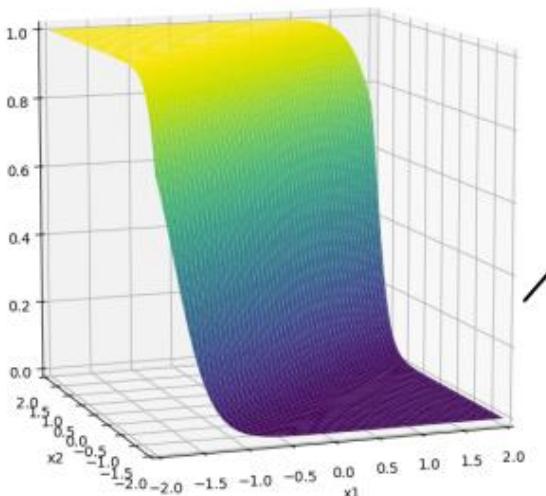
(image credit: Phillip Isola)

Recap: $\sigma_1 = \sigma(5x_1 + -5x_2 + 1)$



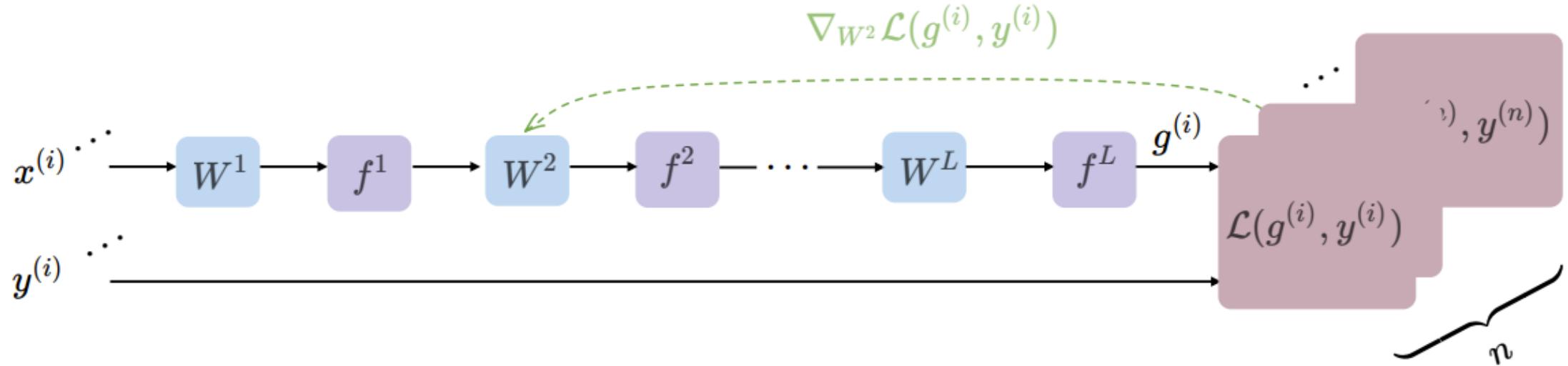
some weighted sum

$\sigma_2 = \sigma(-5x_1 + 5x_2 + 1)$



Recap:

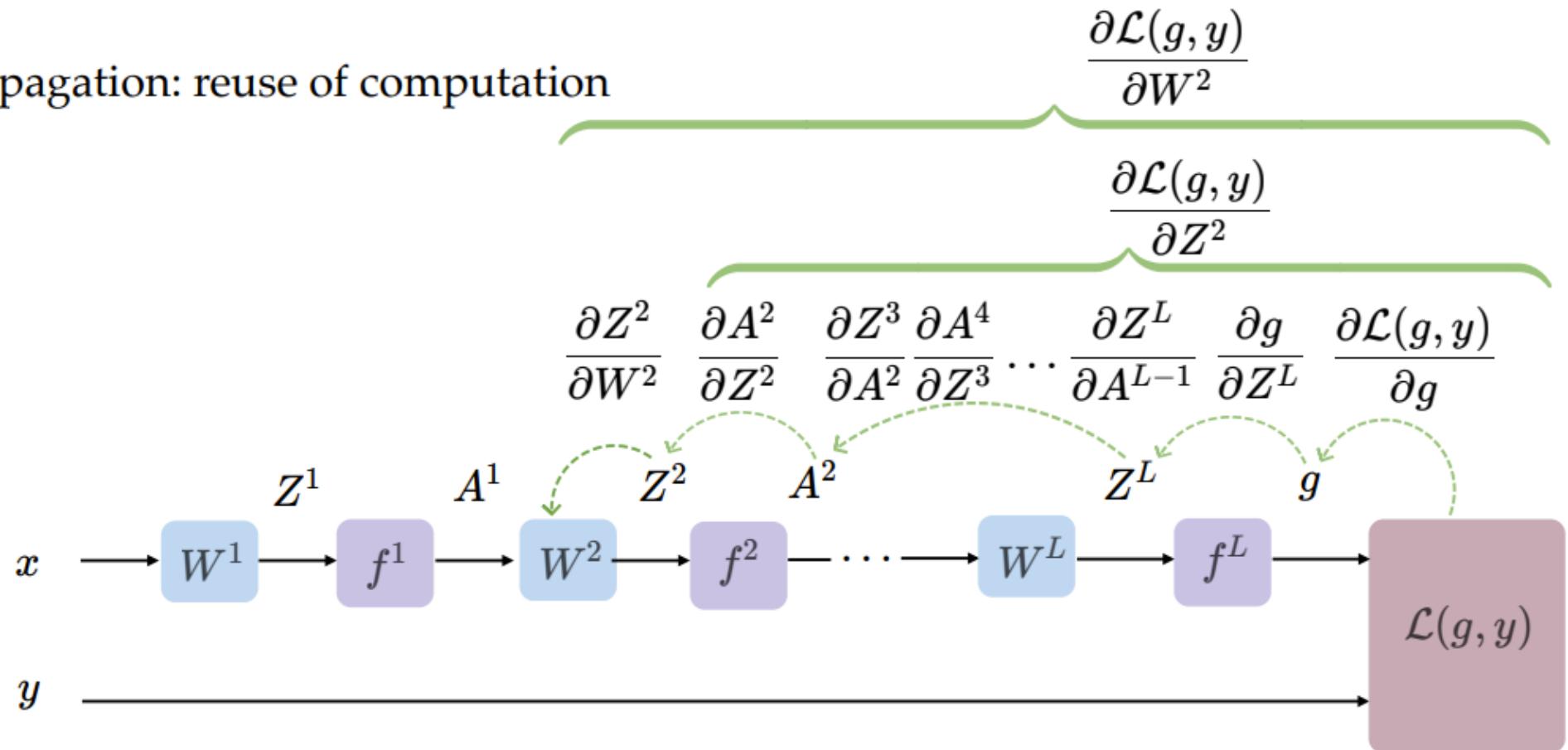
Backward pass: run SGD to update the parameters, e.g. to update W^2



- Randomly pick a data point $(x^{(i)}, y^{(i)})$
- Evaluate the gradient $\nabla_{W^2} \mathcal{L}(g^{(i)}, y^{(i)})$
- Update the weights $W^2 \leftarrow W^2 - \eta \nabla_{W^2} \mathcal{L}(g^{(i)}, y^{(i)})$

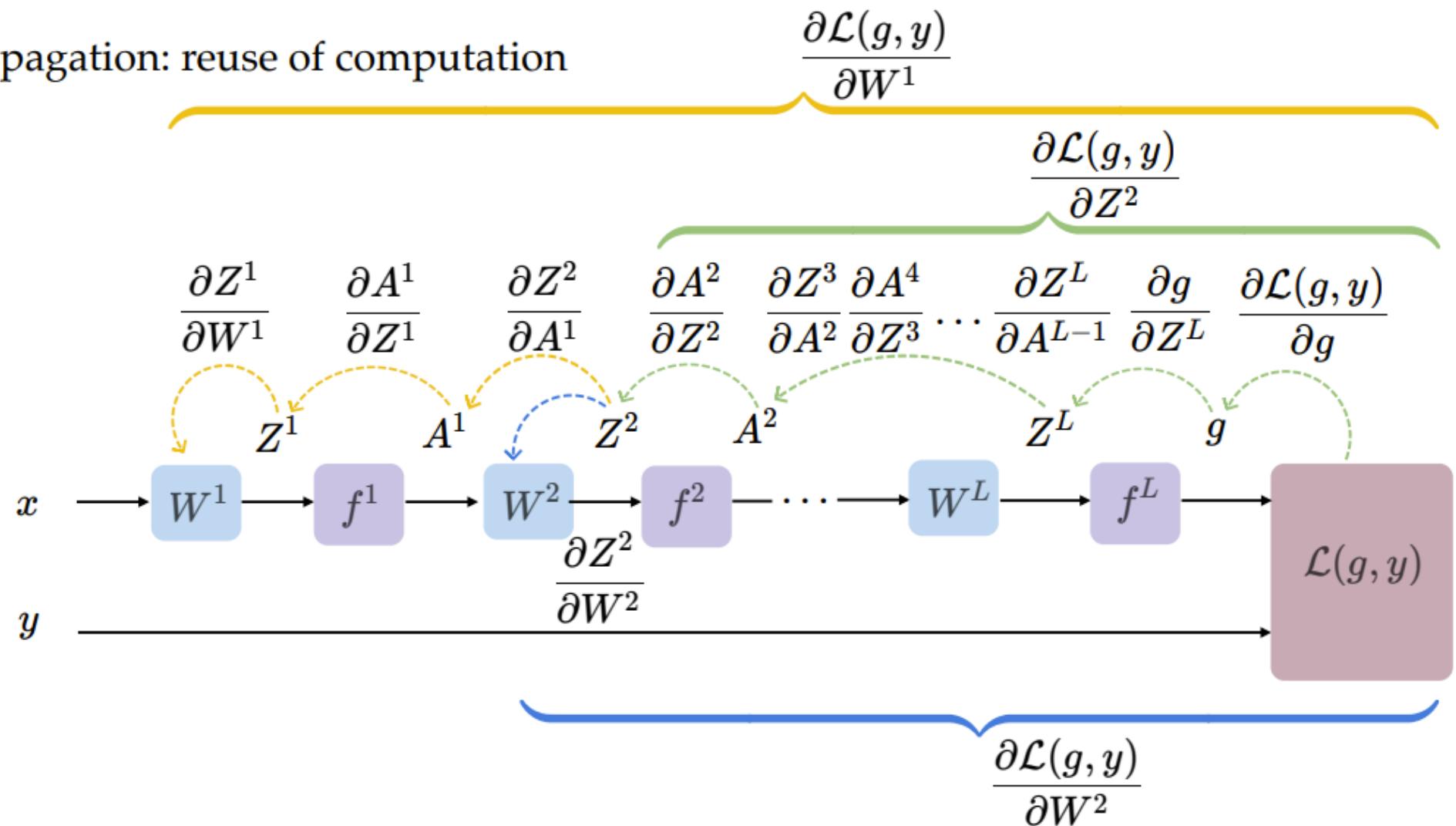
Recap:

back propagation: reuse of computation



Recap:

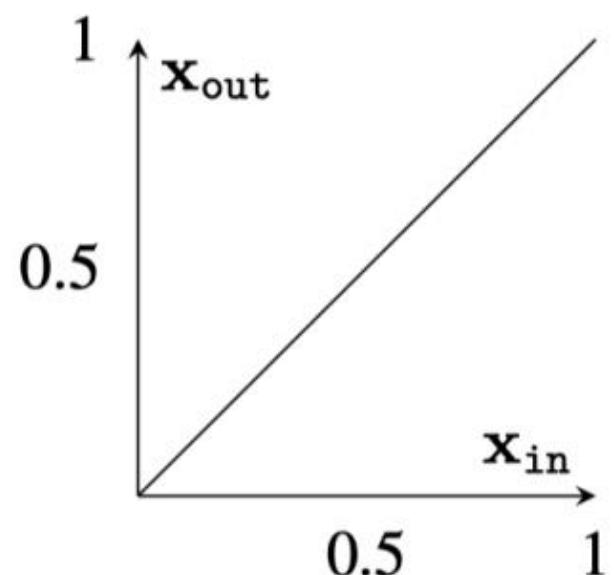
back propagation: reuse of computation



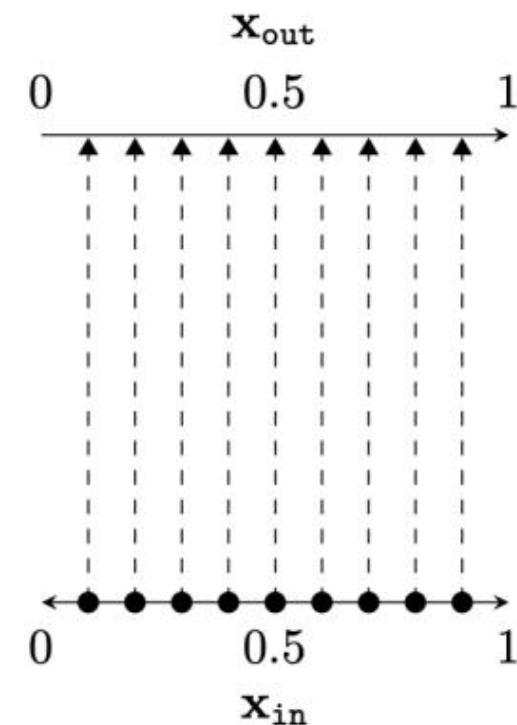
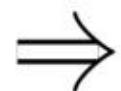
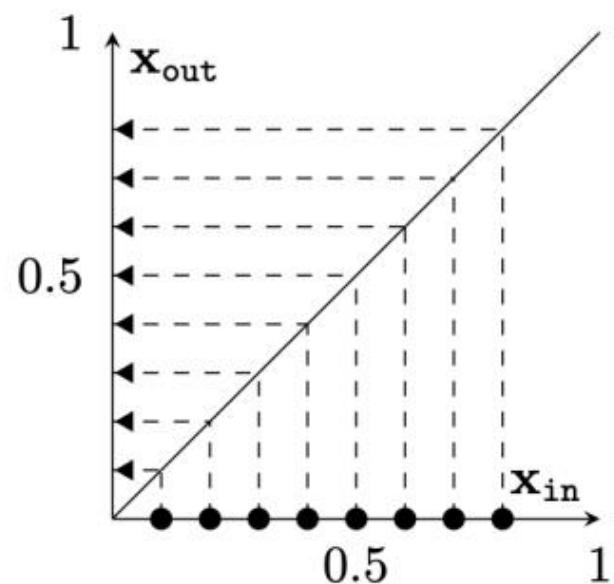
Outline

- Recap, neural networks mechanism
- Neural networks are *representation* learners
- Auto-encoder:
 - Bottleneck
 - Reconstruction
- Unsupervised learning
- (Some recent representation learning ideas)

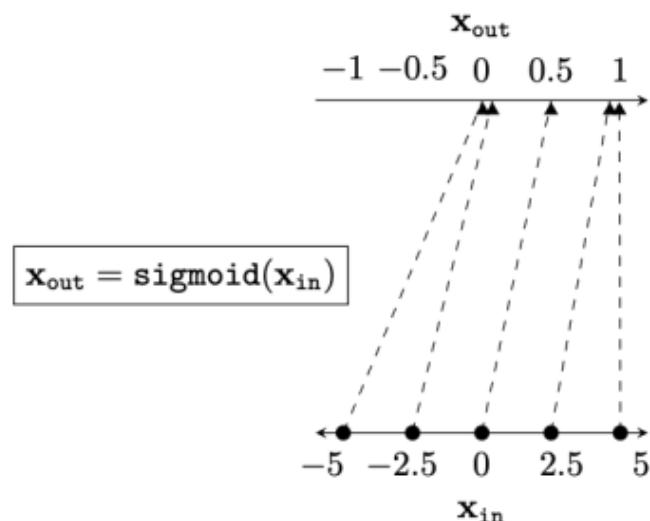
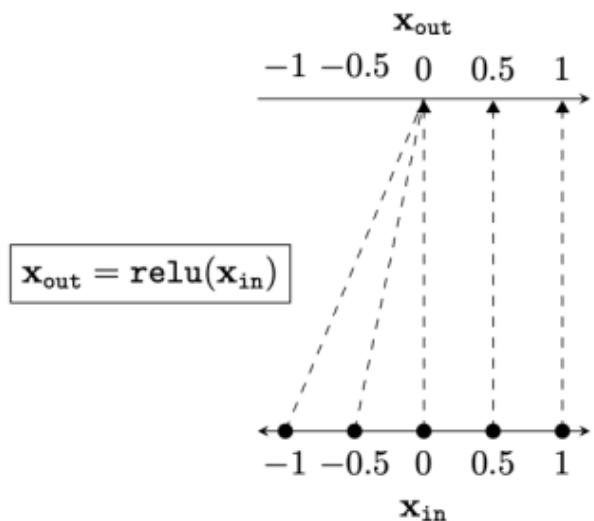
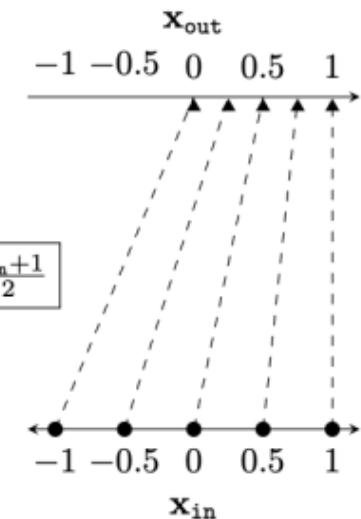
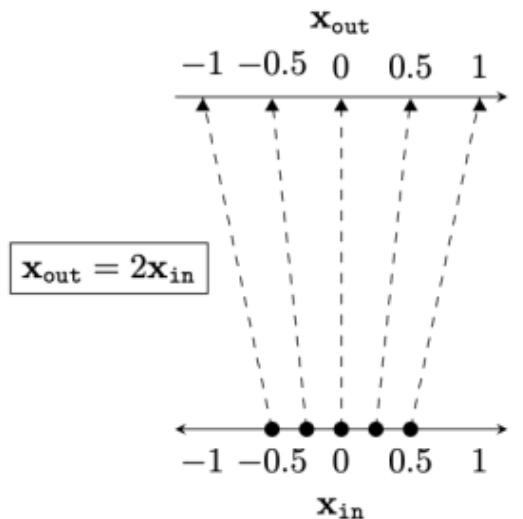
Two different ways to visualize a function



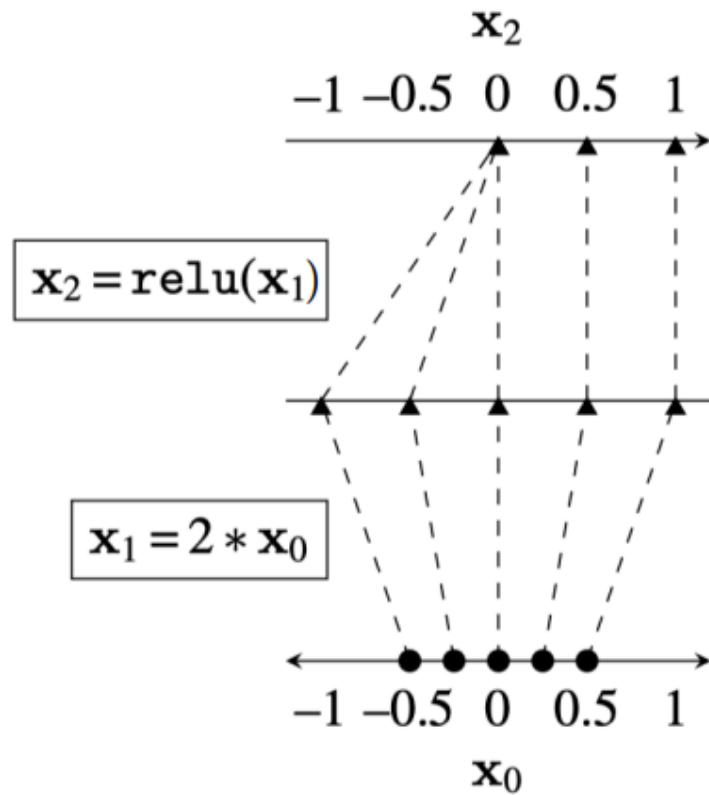
Two different ways to visualize a function



Representation transformations for a variety of neural net operations

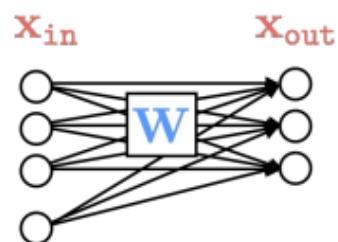


and stack of neural net operations



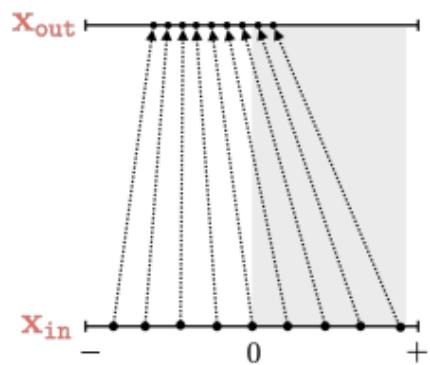
Parameters

linear

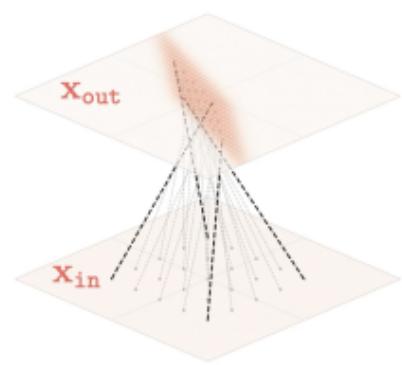


$$\mathbf{x}_{\text{out}} = \mathbf{W} \mathbf{x}_{\text{in}}$$

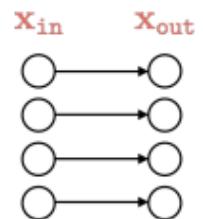
mapping 1D



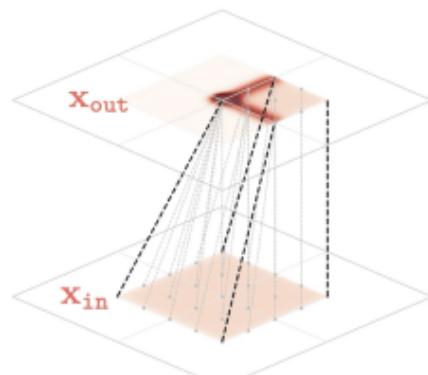
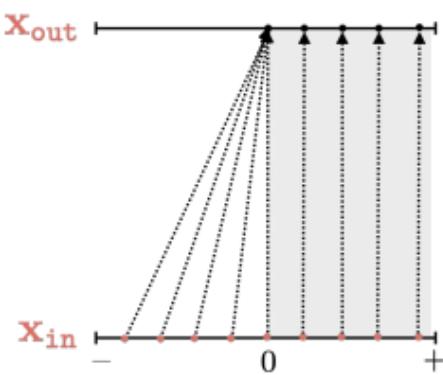
mapping 2D

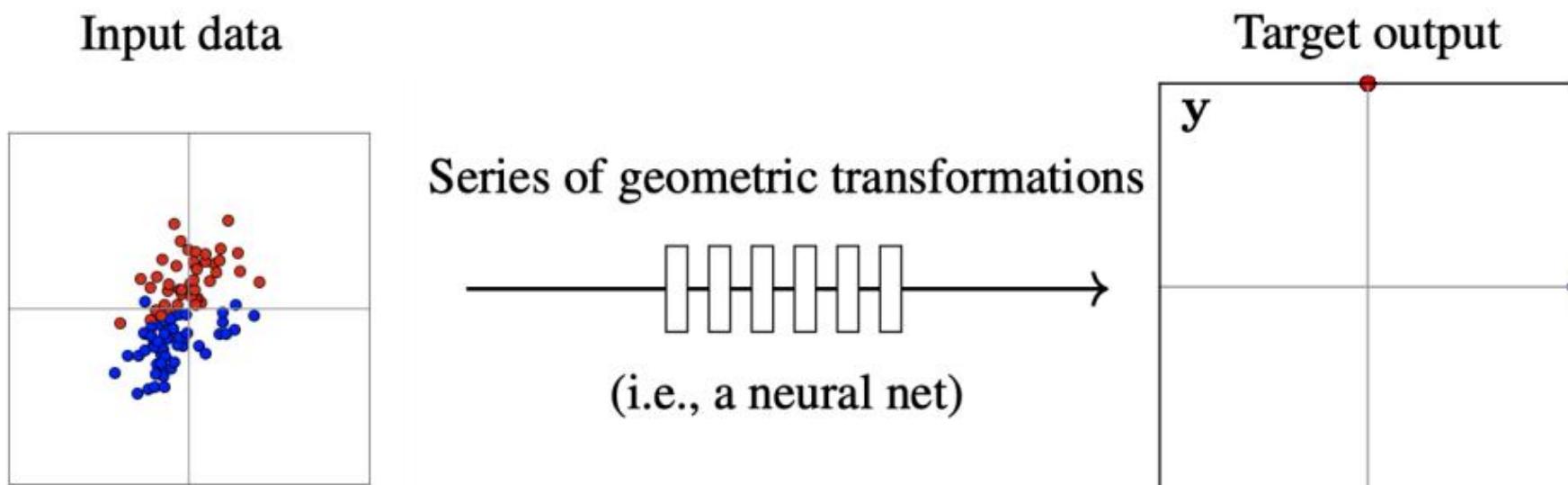


relu



$$x_{\text{out}_i} = \max(x_{\text{in}_i}, 0)$$





$$g = \text{softmax}(z_2)$$

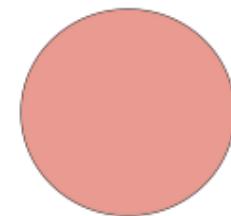
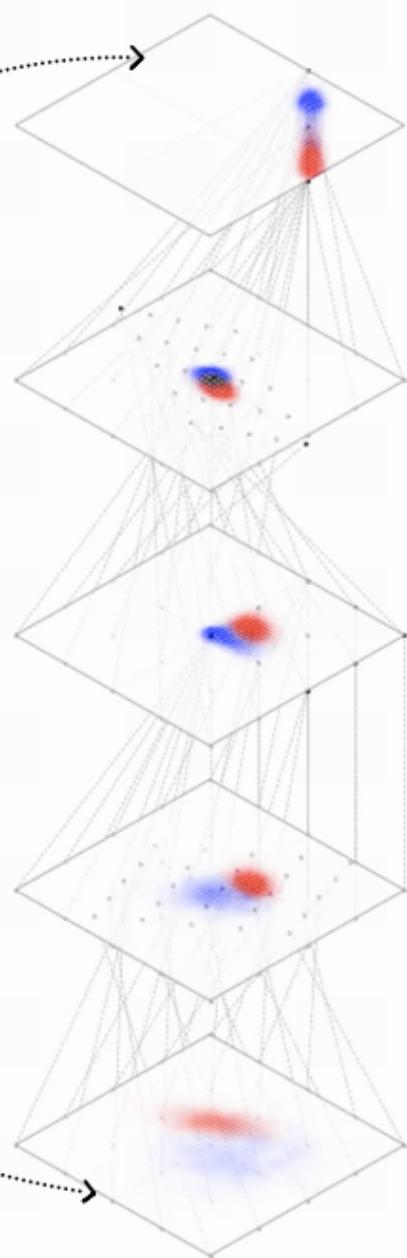
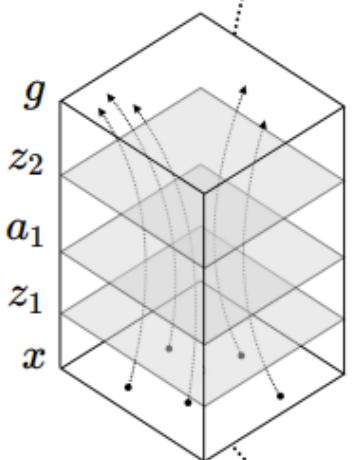
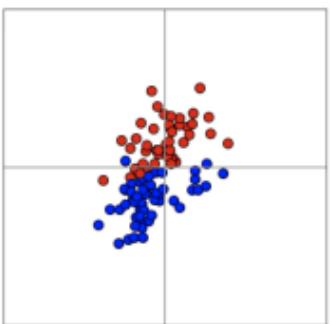
$$z_2 = \text{linear } (a_1)$$

$$a_1 = \text{ReLU}(z_1)$$

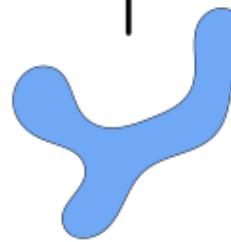
$$z_1 = \text{linear } (x)$$

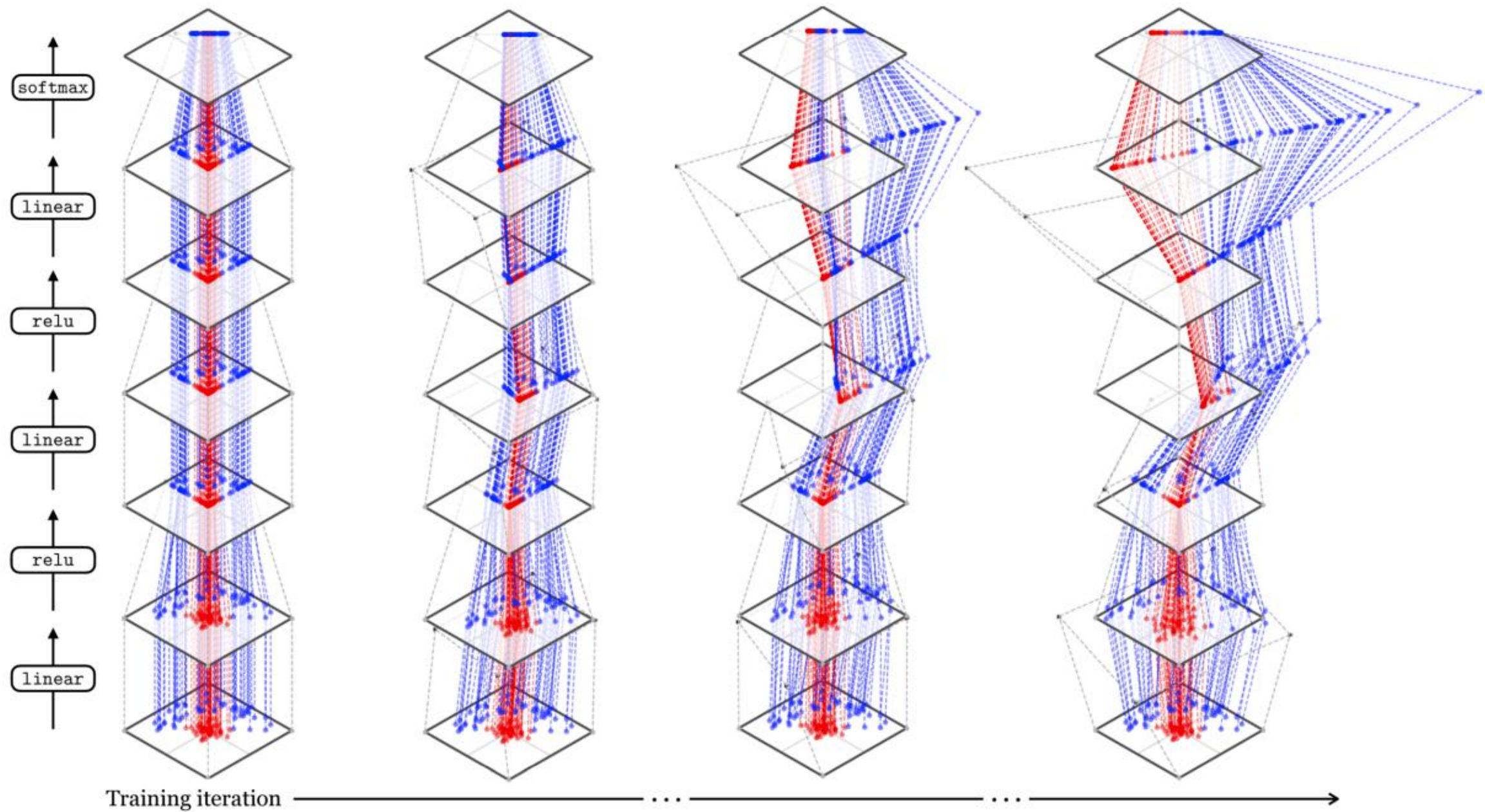
$$x \in \mathbb{R}^2$$

Training data



maps from
complex data
space to simple
embedding space

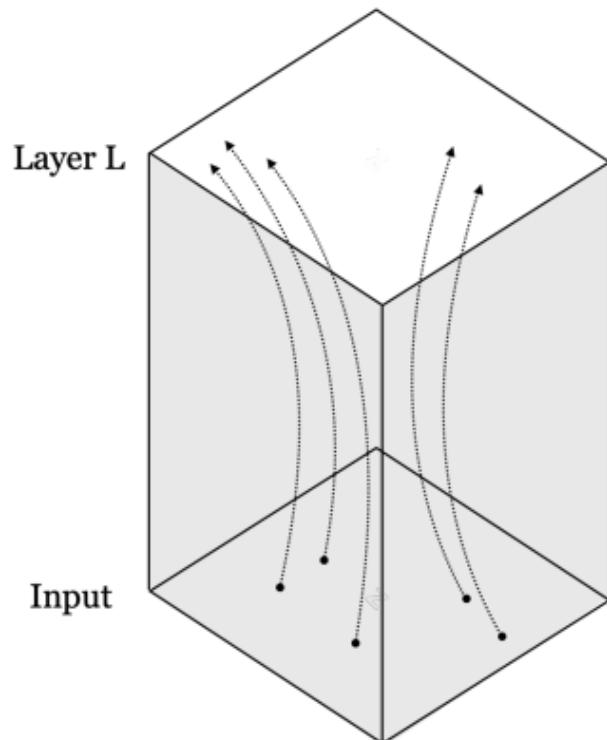




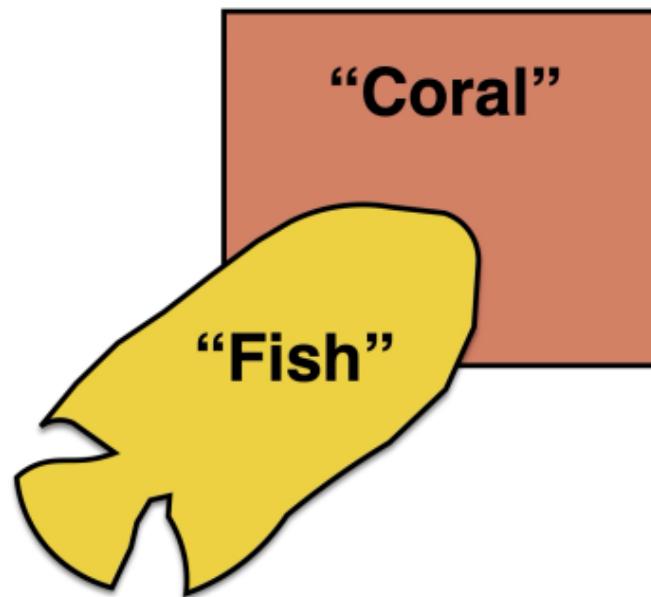
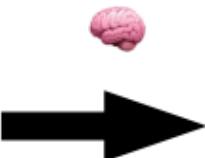
Neural networks are representation learners

Deep nets transform datapoints, layer by layer

Each layer gives a different *representation* (aka *embeddings*)
of the data



humans also learn representations



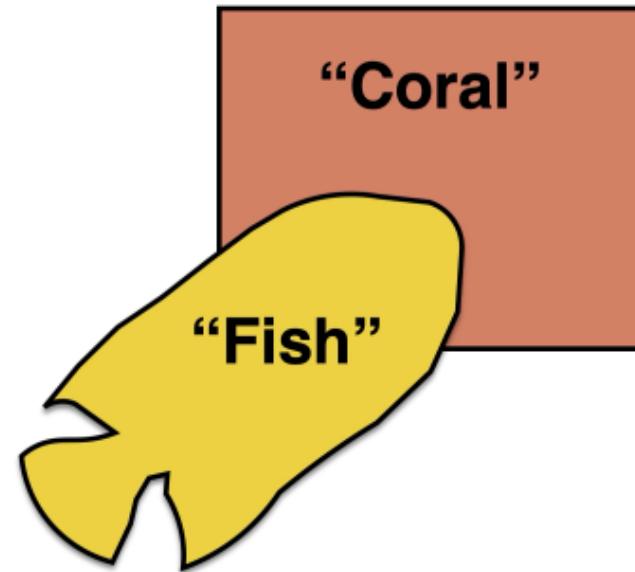


"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees."

— Max Wertheimer, 1923

Good representations are:

- Compact (*minimal*)
- Explanatory (*roughly sufficient*)

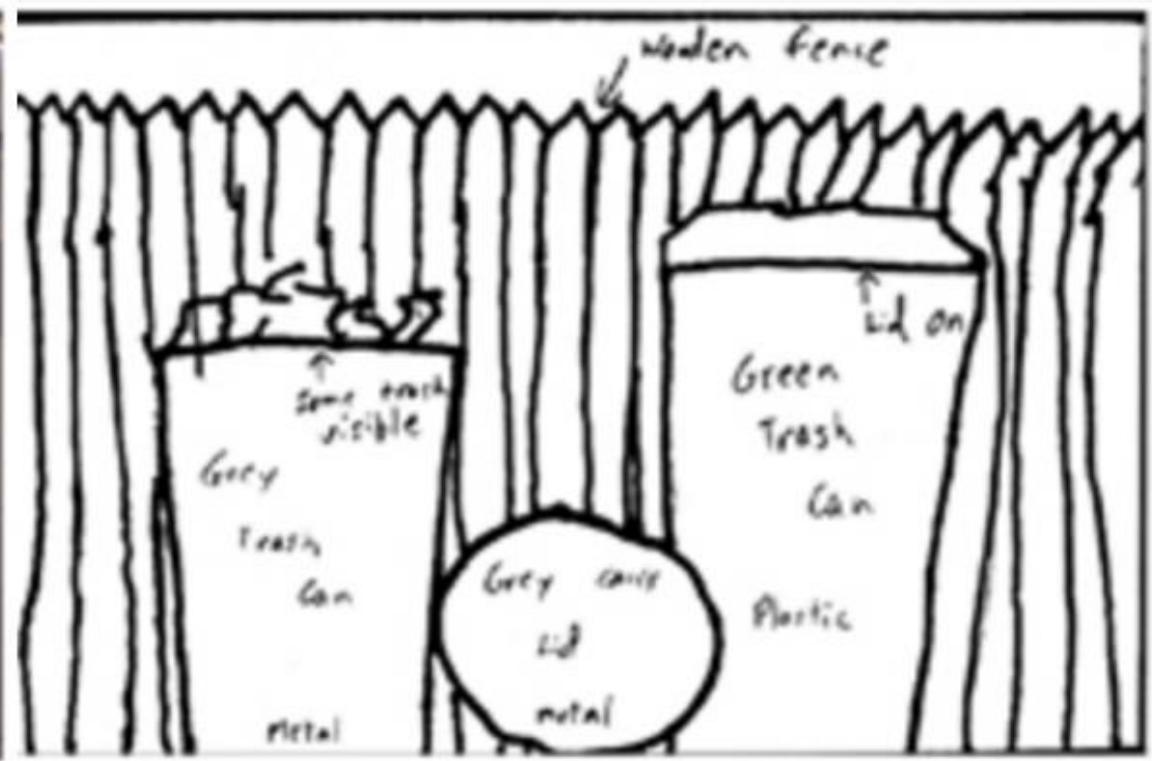


[See “Representation Learning”, Bengio 2013, for more commentary]

Observed image

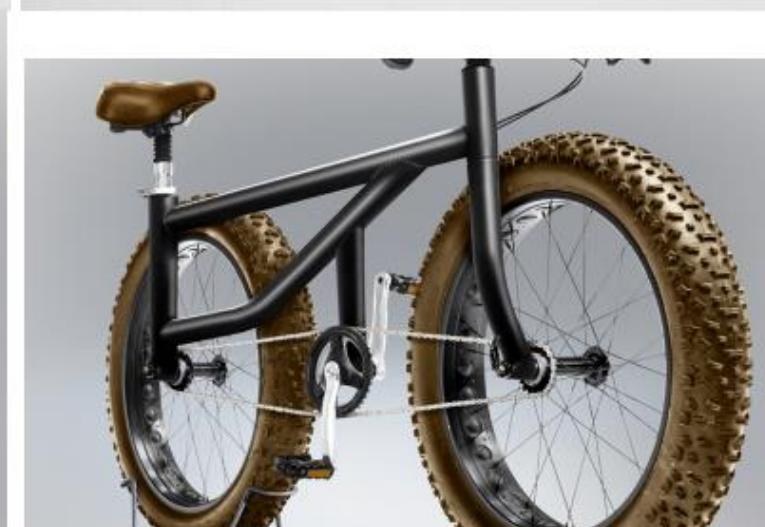


Drawn from memory



[Bartlett, 1932]

[Intraub & Richardson, 1989]



Outline

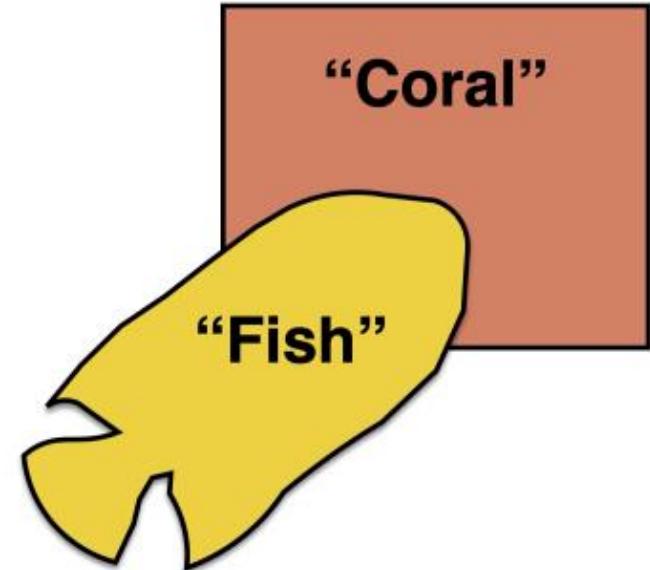
- Recap, neural networks mechanism
- Neural networks are *representation* learners
- Auto-encoder:
 - Bottleneck
 - Reconstruction
- Unsupervised learning
- (Some recent representation learning ideas)

Auto-encoders try
to achieve these

these may just
emerge as well

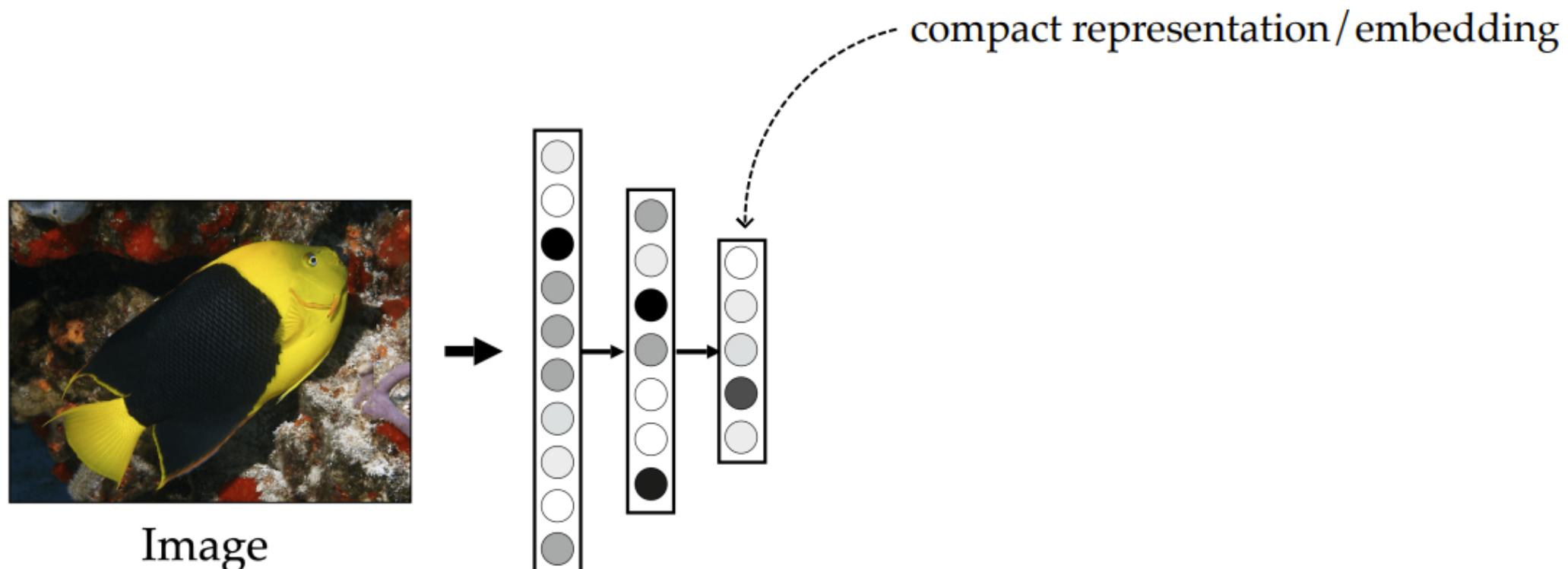
Good representations are:

- { • Compact (*minimal*)
- Explanatory (*roughly sufficient*)
- { • Disentangled (*independent factors*)
- Interpretable
- Make *subsequent problem solving easy*

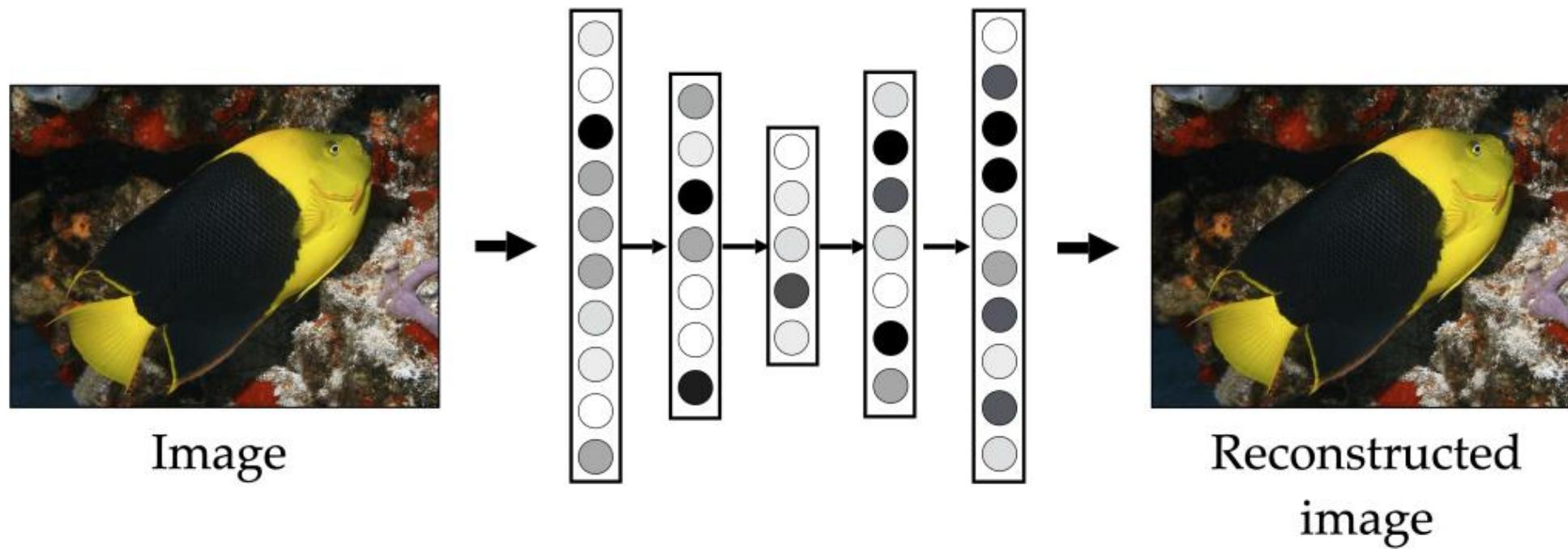


[See "Representation Learning", Bengio 2013, for more commentary]

Auto-encoder



Auto-encoder



"What I cannot create, I do not understand." Feynman

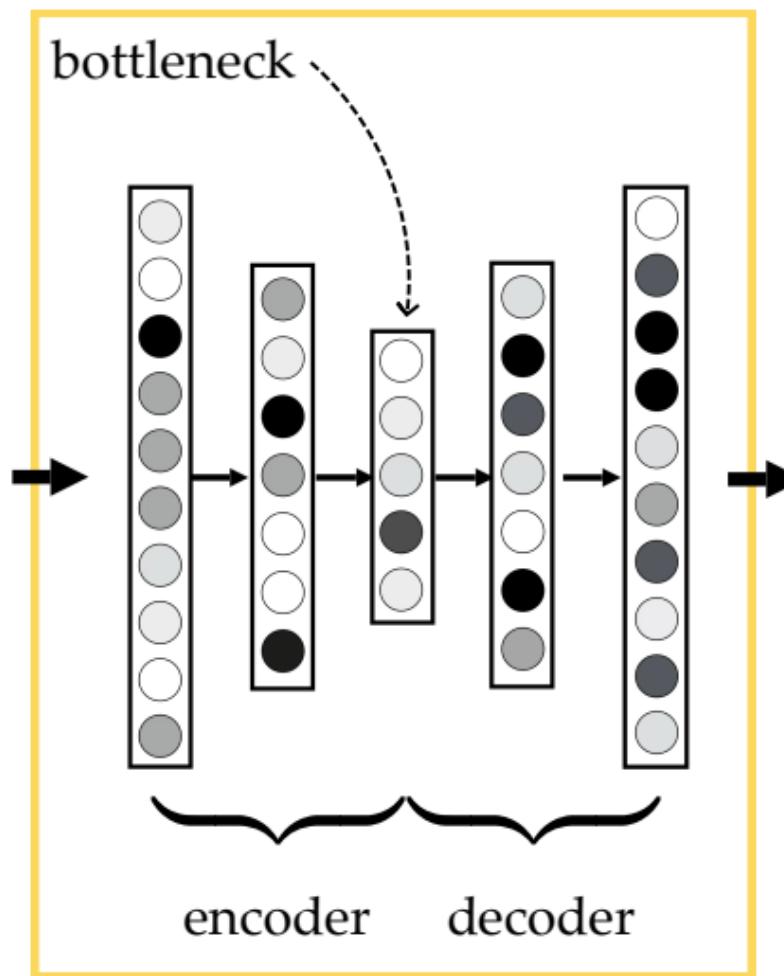
Auto-encoder

$$\min_W ||x - \tilde{x}||^2$$

x



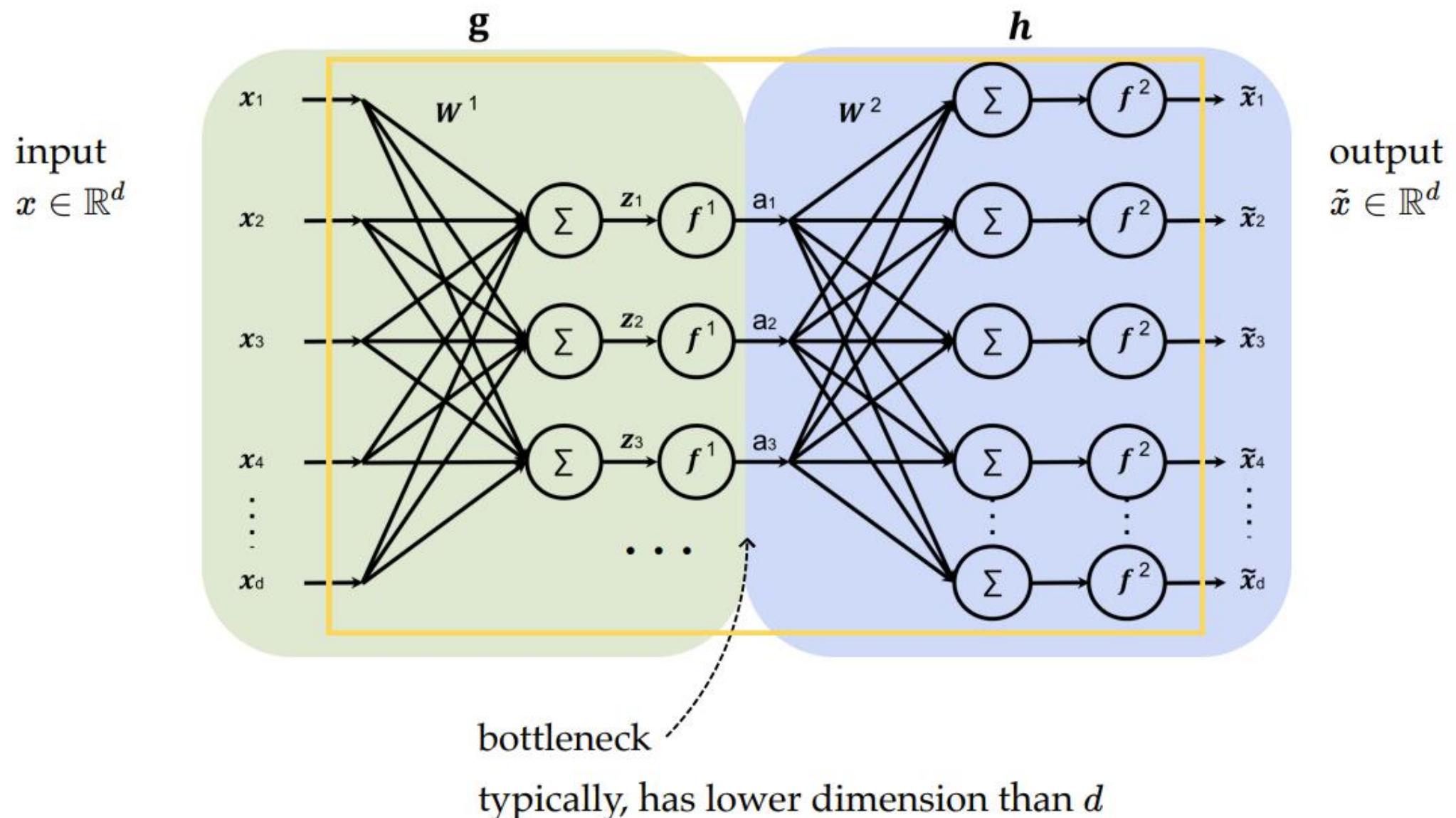
Image

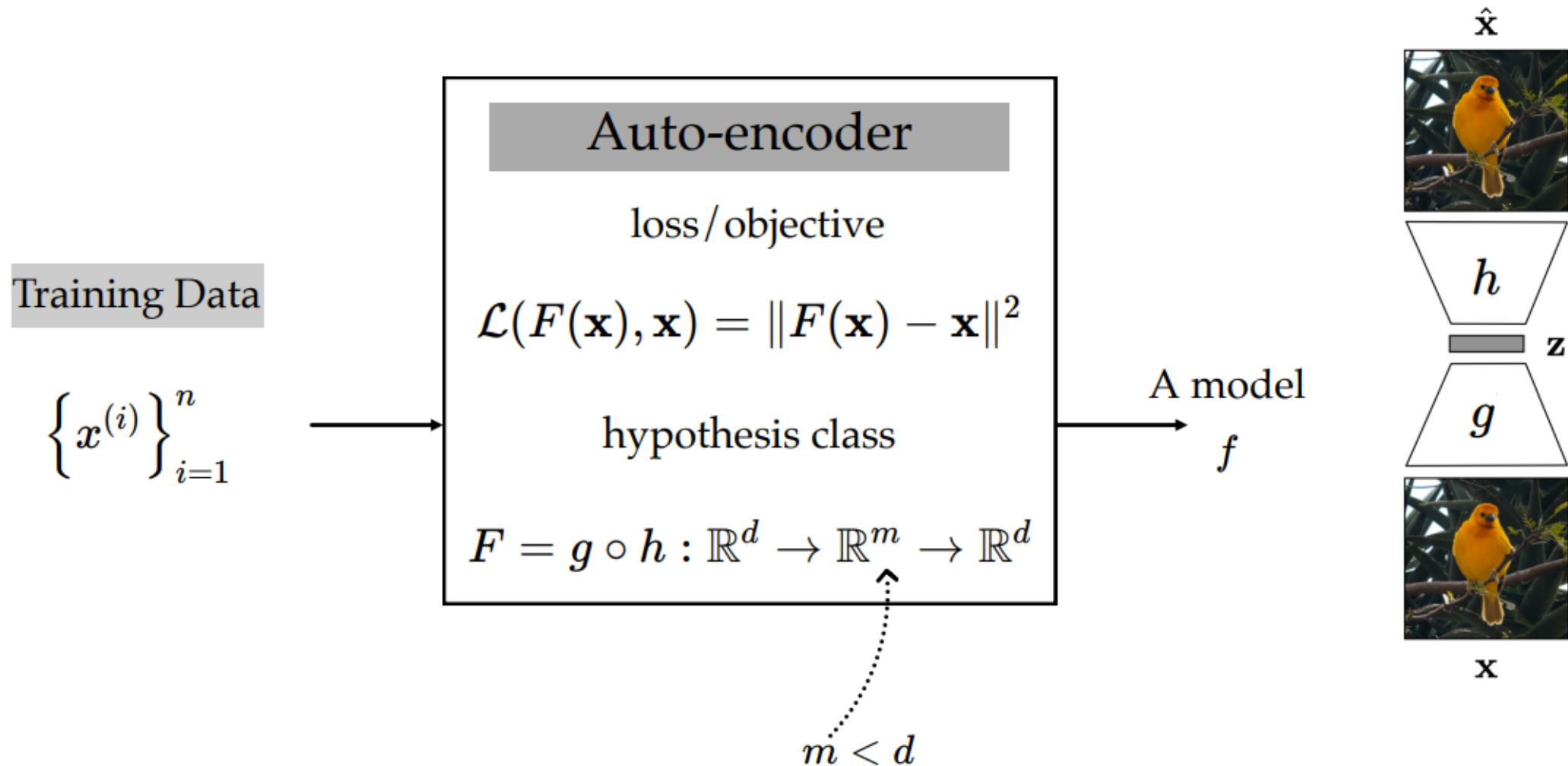


$$\tilde{x} = \text{NN}(x; W)$$



Reconstructed
image





Supervised Learning

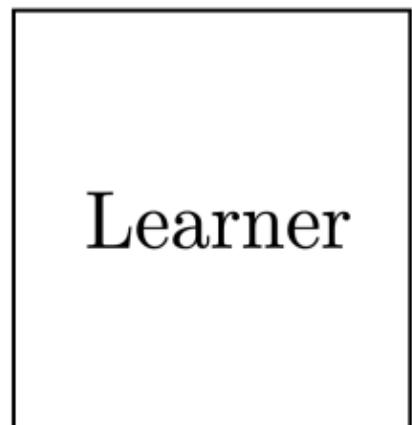
Training data

$$\{x^{(1)}, y^{(1)}\}$$

$$\{x^{(2)}, y^{(2)}\} \rightarrow$$

$$\{x^{(3)}, y^{(3)}\}$$

...



$$\rightarrow f : X \rightarrow Y$$

Undergrads were time-consuming, algorithms were flawed, and the team didn't have money—Li said the project failed to win any of the federal grants she applied for, receiving comments on proposals that it was shameful Princeton would research this topic, and that the only strength of proposal was that Li was a woman.

A solution finally surfaced: a graduate student who a Amazon Mechanical Turk sitting at computers around the world online tasks for pennies.

“He showed me the web, I knew the ImageNet project was going to happen,” she said. “Suddenly we found a tool that could scale, that we could not possibly dream of by hiring Princeton undergrads.”



The Amazon Mechanical Turk backend for classifying images. Image: ImageNet

Unsupervised Learning

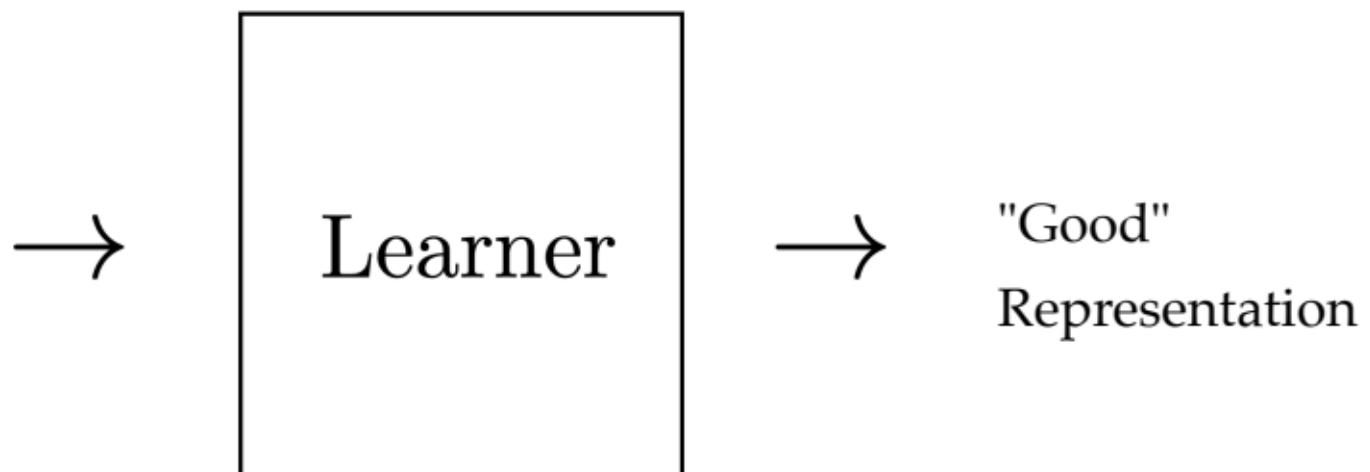
Training Data

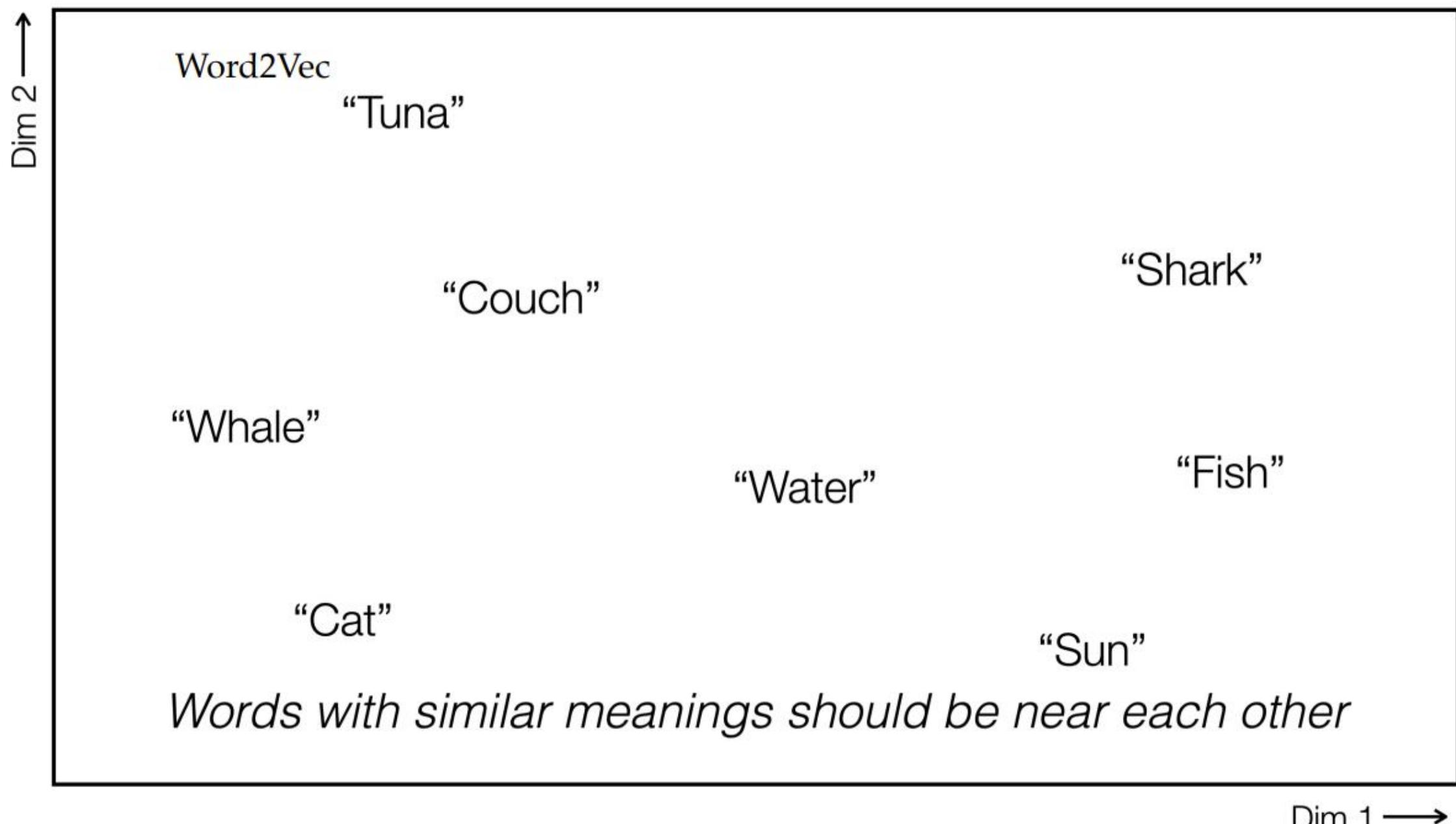
$$\{x^{(1)}\}$$

$$\{x^{(2)}\}$$

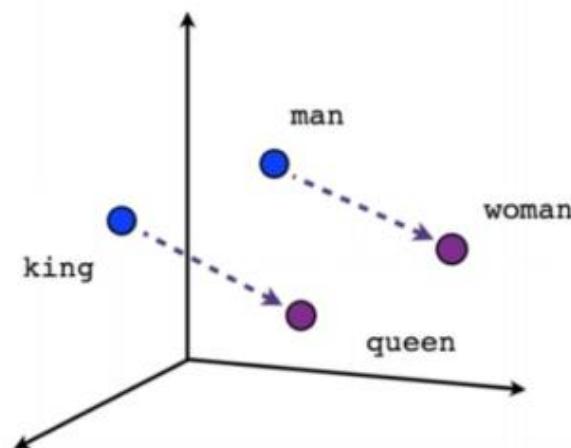
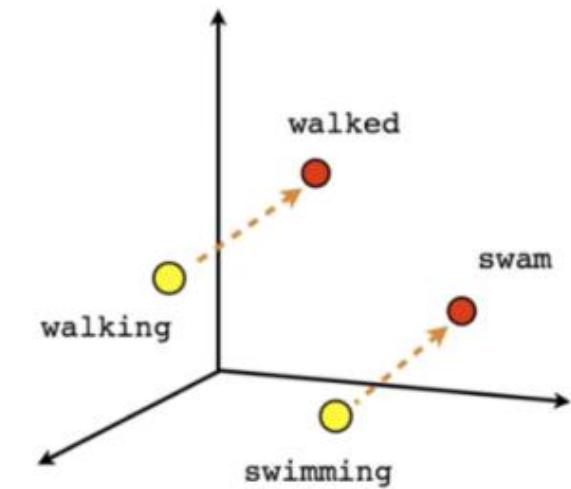
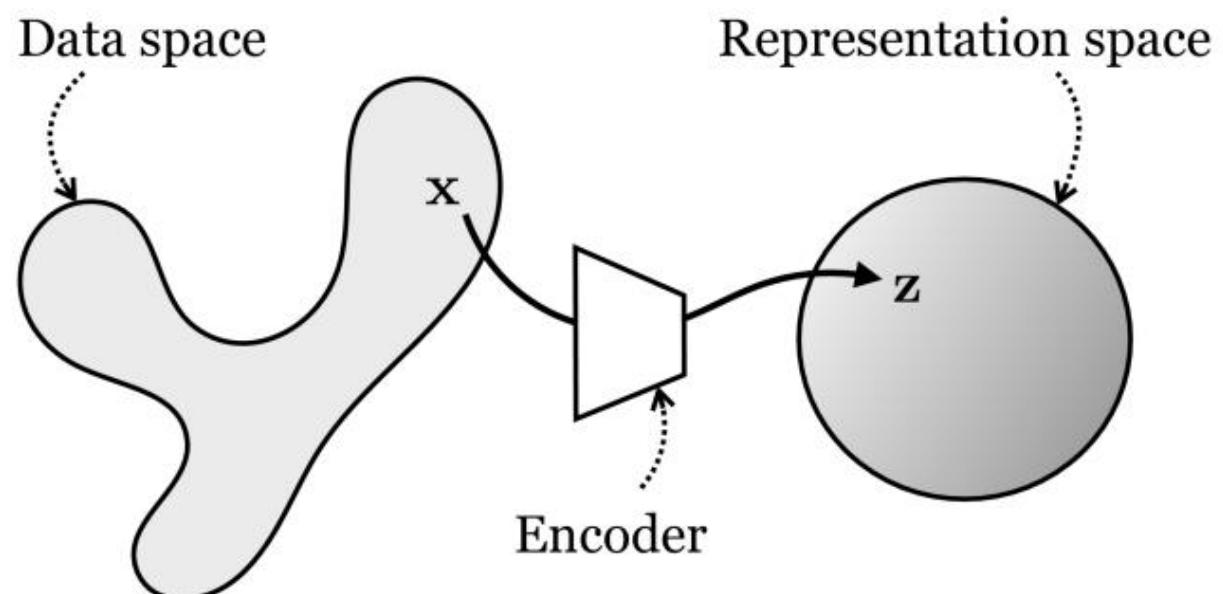
$$\{x^{(3)}\}$$

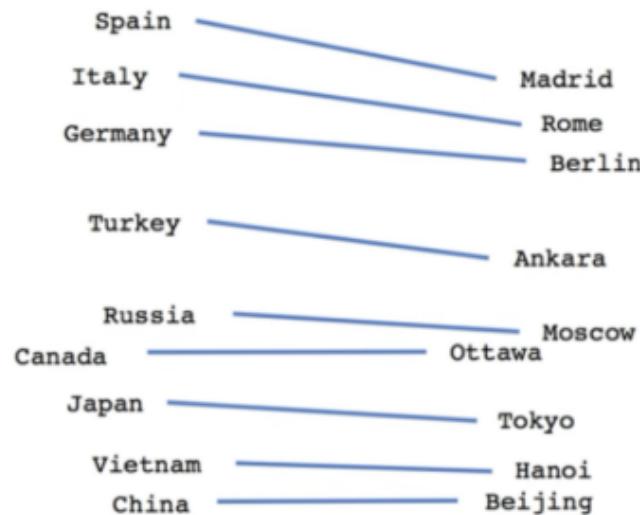
...





Word2Vec





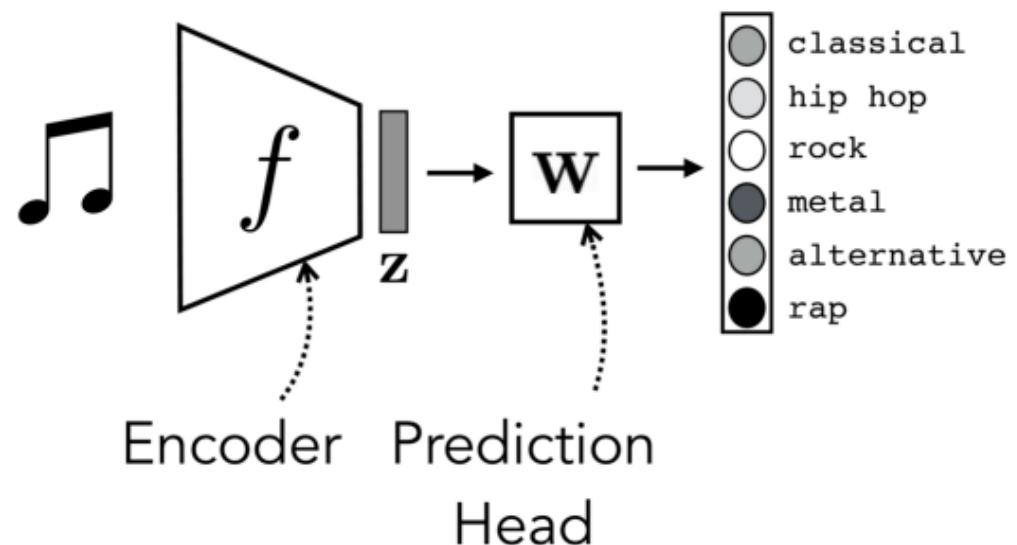
Country-Capital

$$\mathbf{X} = \text{Vector}(\text{"Paris"}) - \text{vector}(\text{"France"}) + \text{vector}(\text{"Italy"}) \approx \text{vector}(\text{"Rome"})$$

"Meaning is use" — Wittgenstein

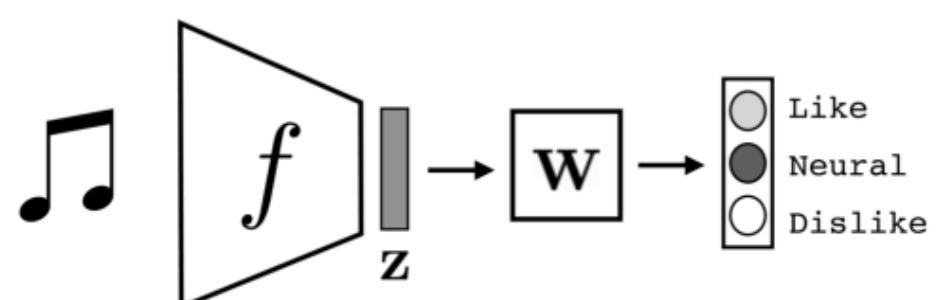
Training

Genre recognition



Testing

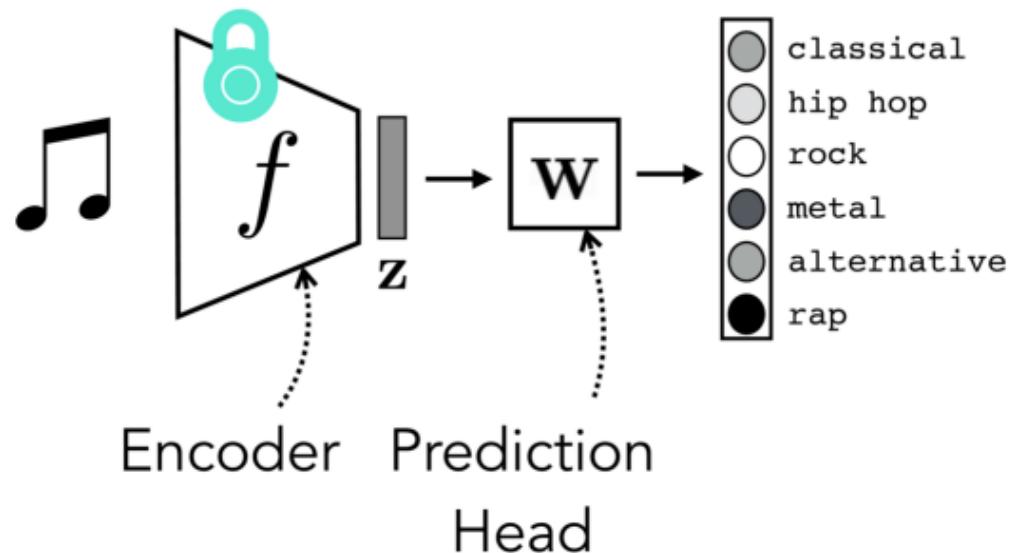
Preference prediction



Often, what we will be “tested” on is not what we were trained on.

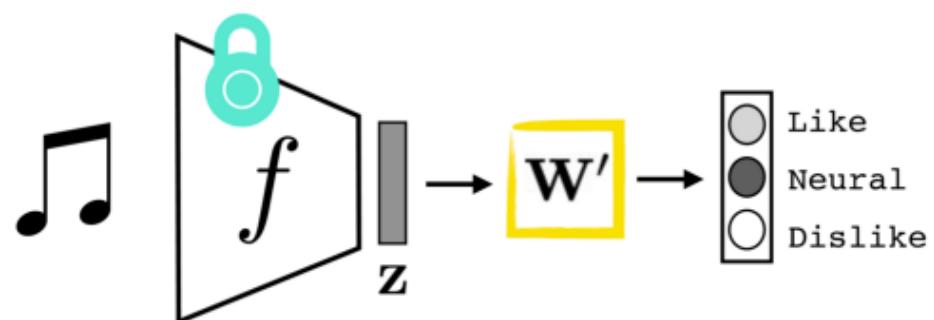
Training

Genre recognition



Adapting

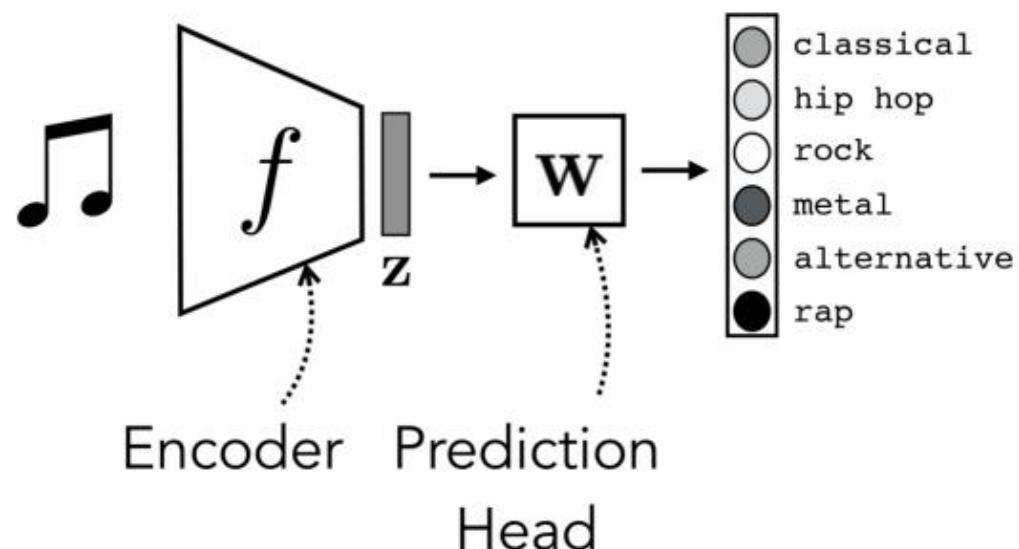
Preference prediction



Final-layer adaptation: freeze f , train a new final layer to new target data

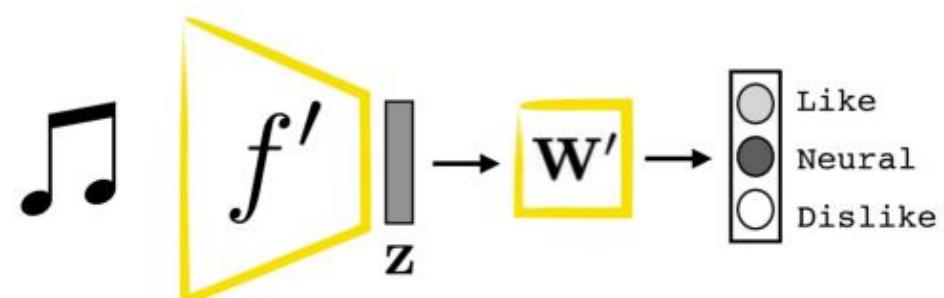
Training

Genre recognition



Adapting

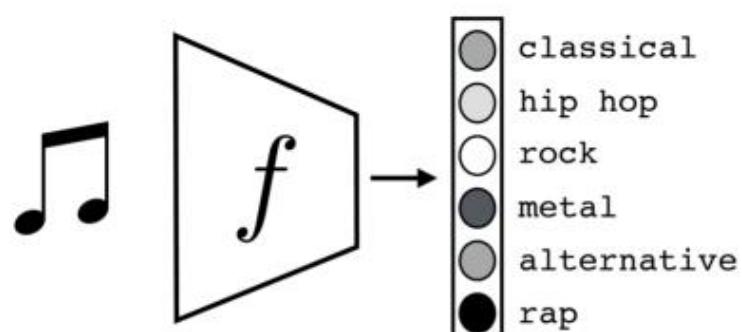
Preference prediction



Finetuning: initialize f' as f , then continue training for f' as well, on new target data

Pretraining

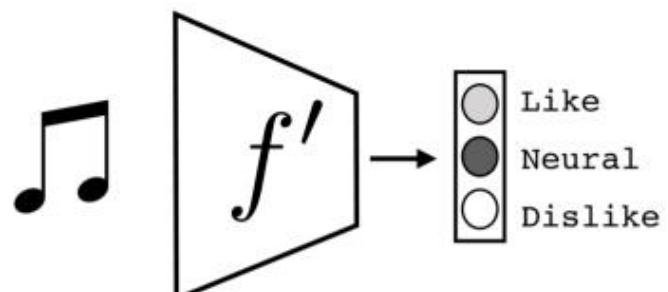
Genre recognition



A lot of data

Adapting

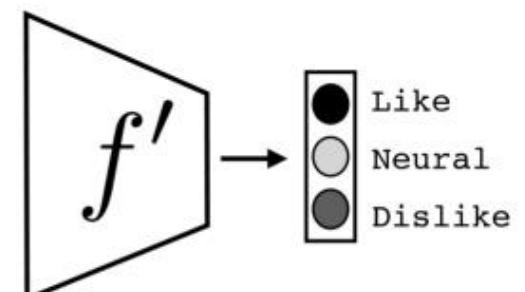
Preference prediction



A little data

Testing

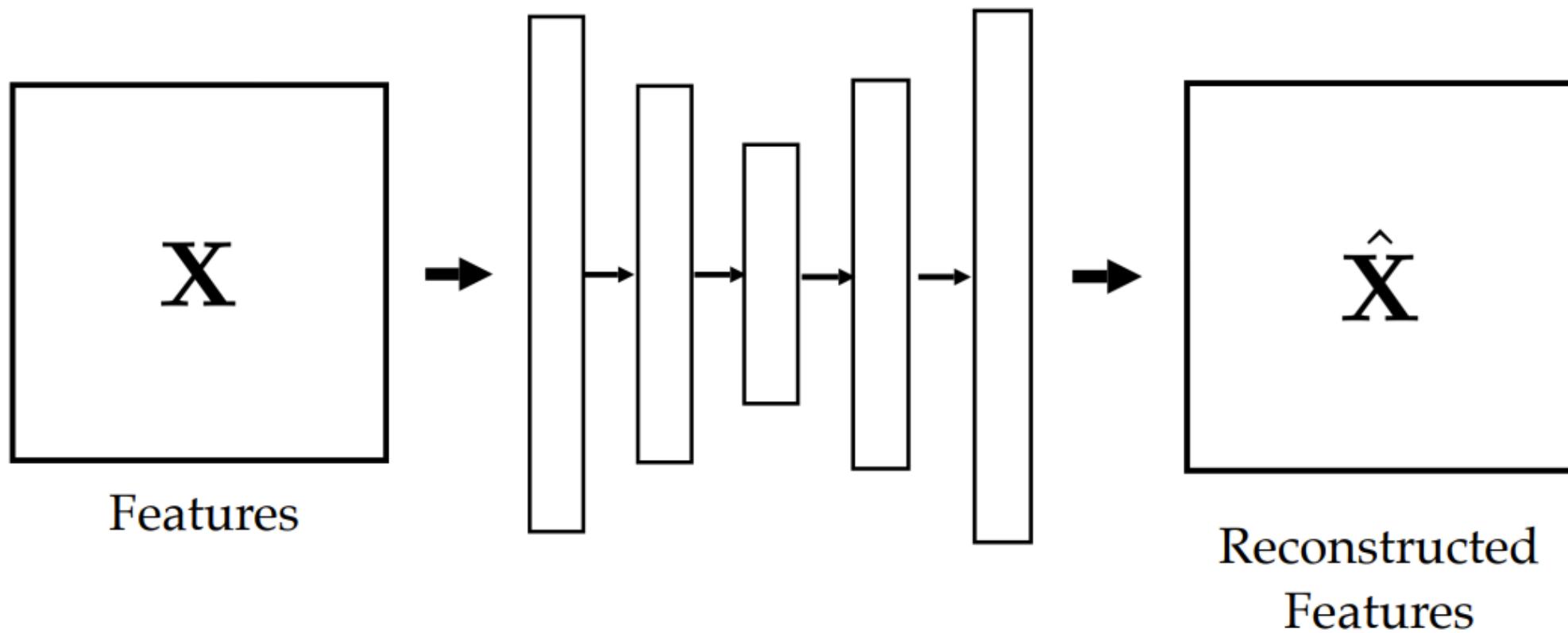
Preference prediction



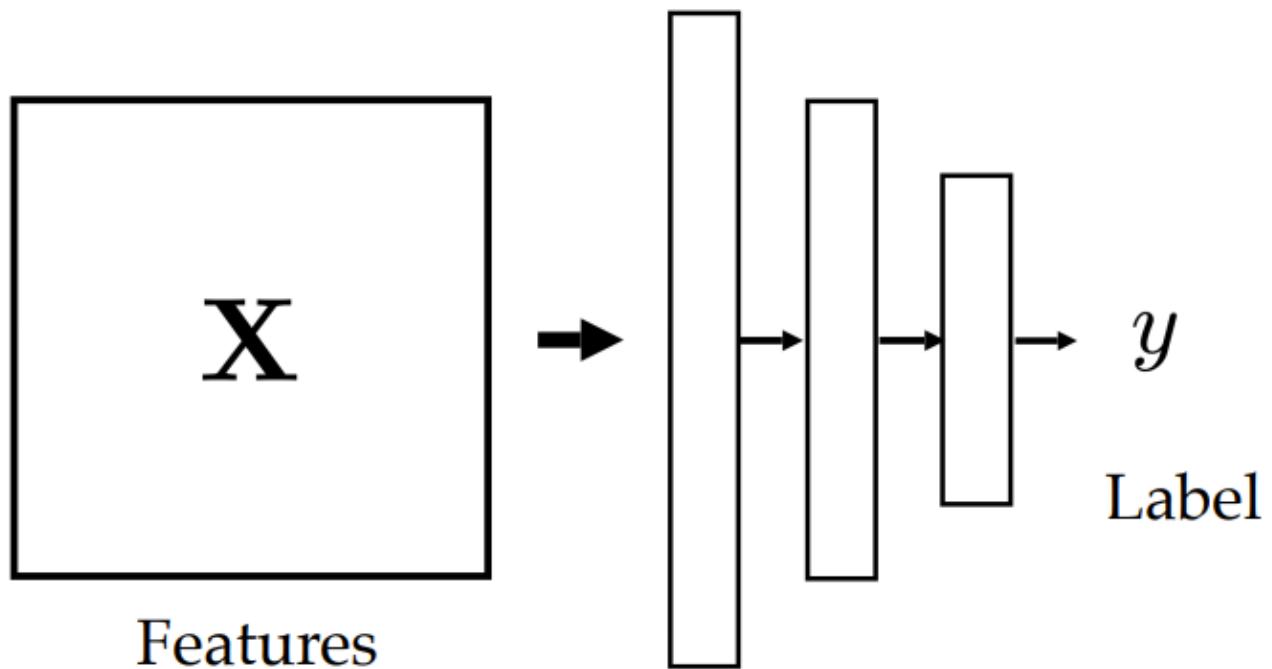
Outline

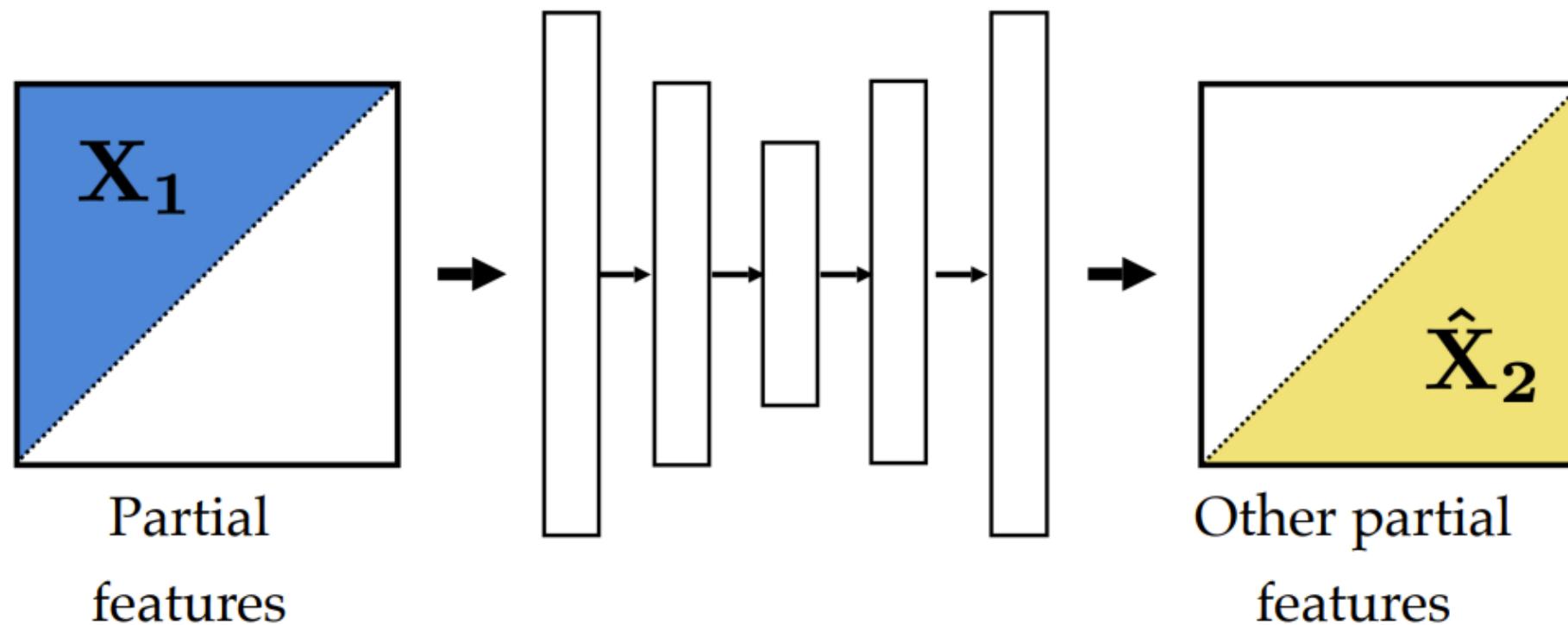
- Recap, neural networks mechanism
- Neural networks are *representation* learners
- Auto-encoder:
 - Bottleneck
 - Reconstruction
- Unsupervised learning
- (Some recent representation learning ideas)

Feature reconstruction (unsupervised learning)

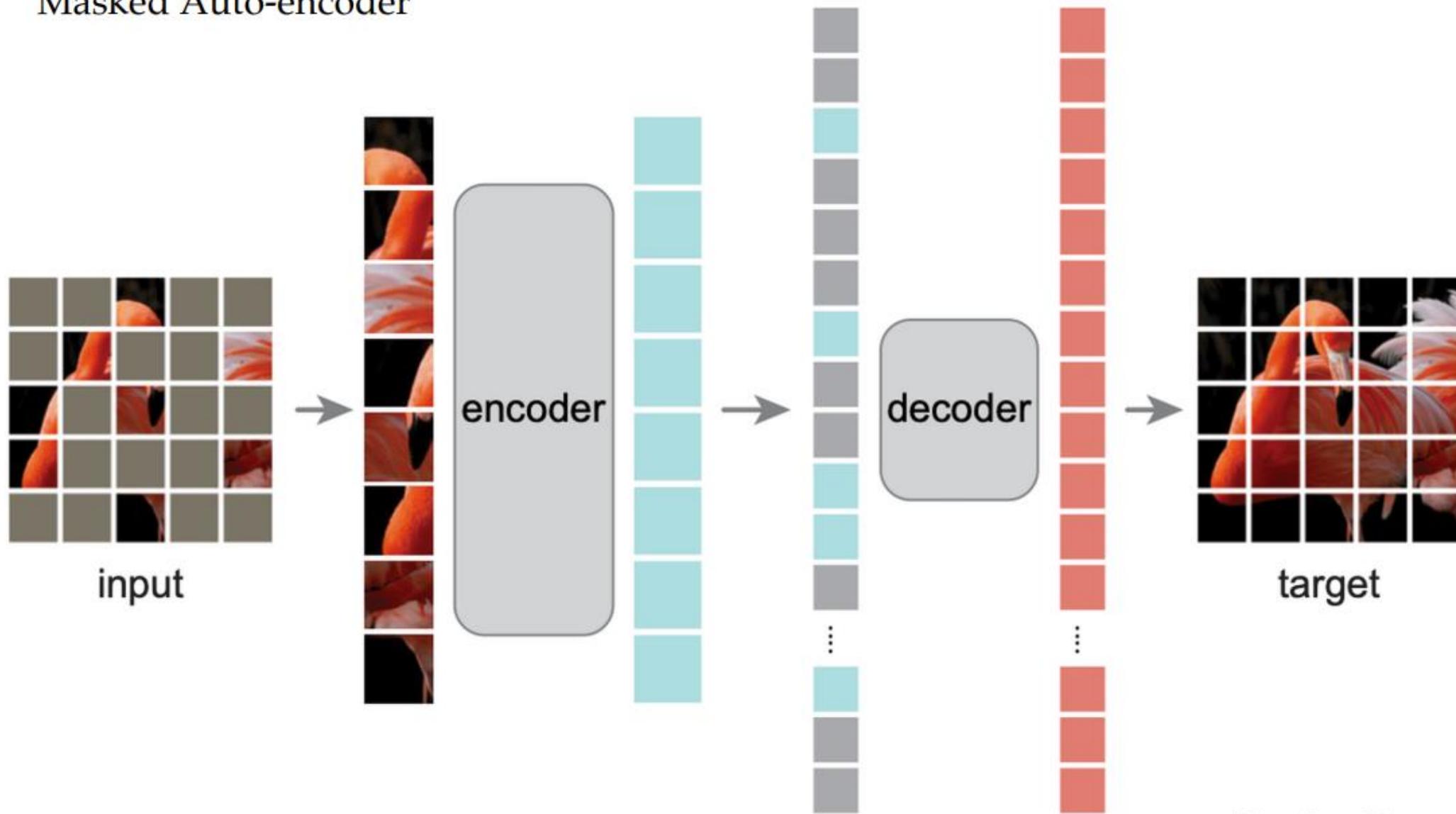


Label prediction (supervised learning)

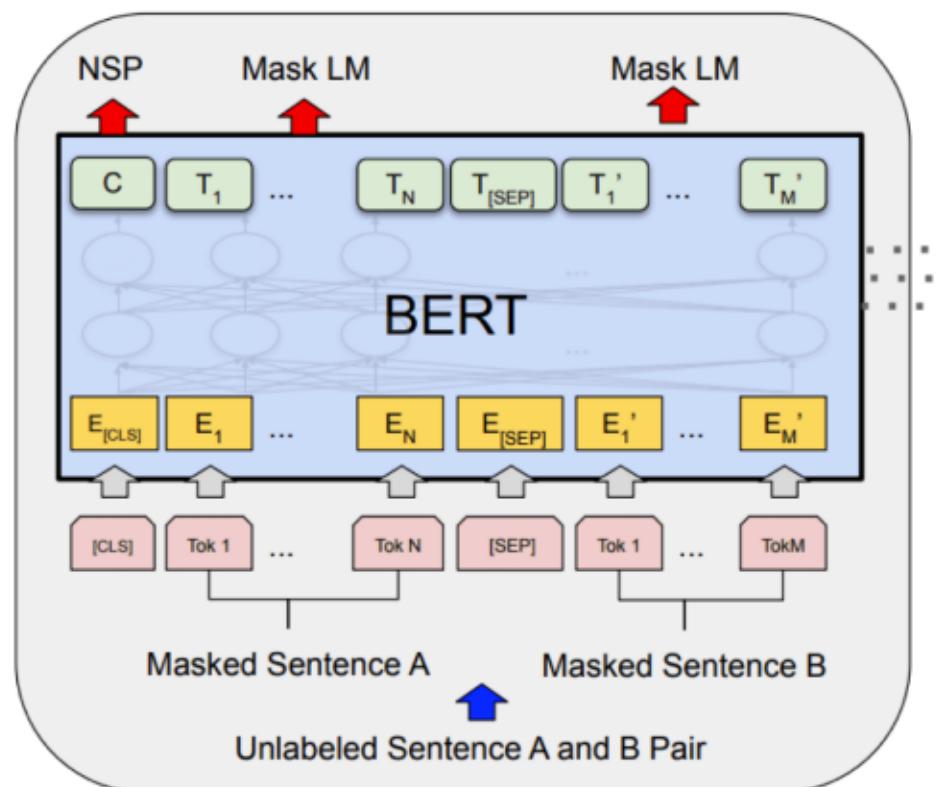




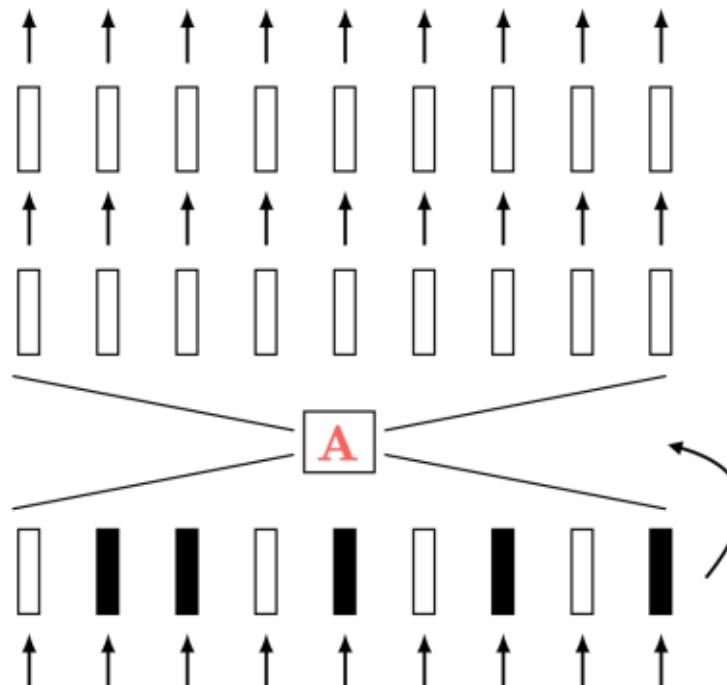
Masked Auto-encoder



Masked Auto-encoder



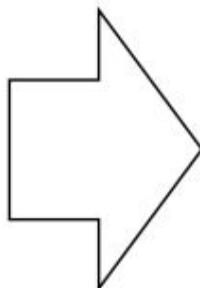
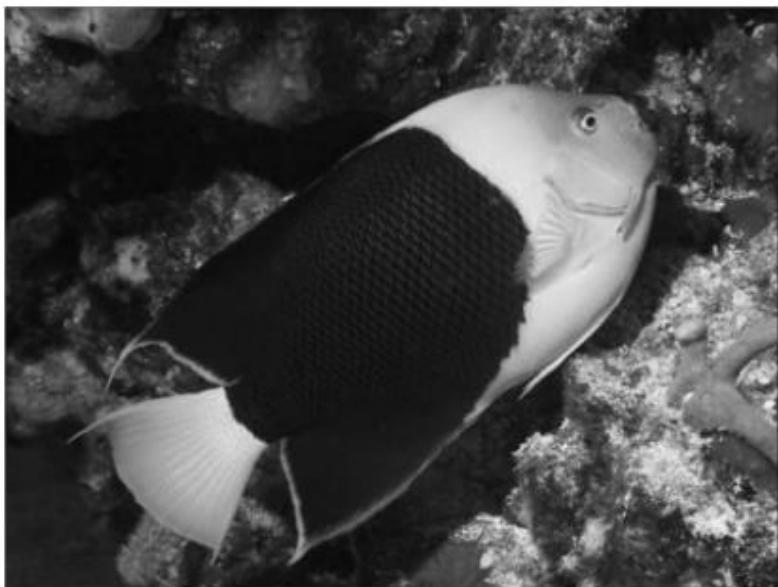
Colorless green ideas sleep furiously





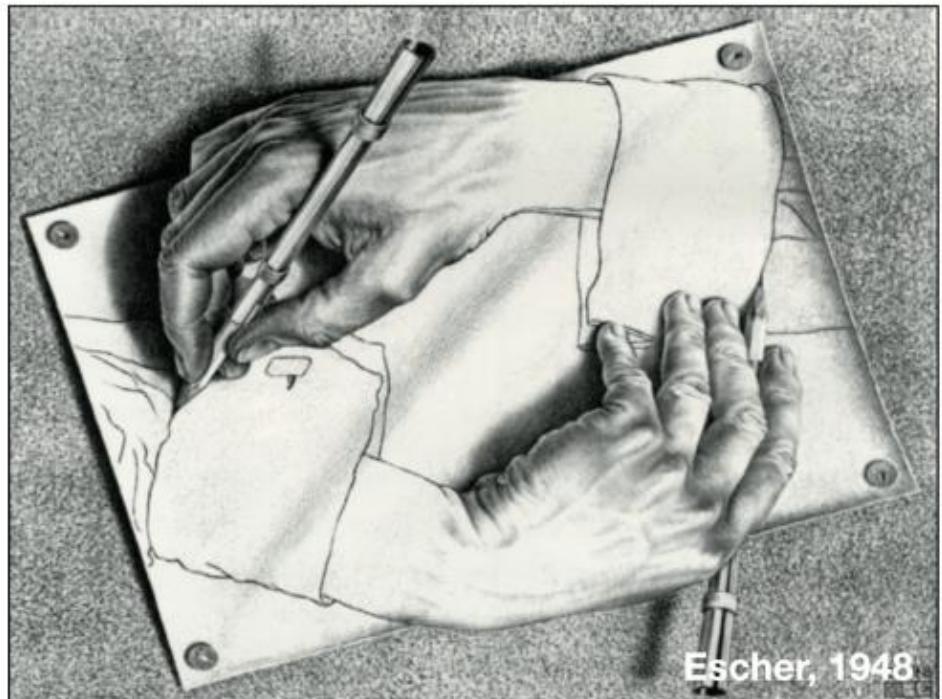
[Zhang, Isola, Efros, ECCV 2016]

predict color from gray-scale



[Zhang, Isola, Efros, ECCV 2016]

Self-supervised learning



Common trick:

- Convert “unsupervised” problem into “supervised” setup
- Do so by cooking up “labels” (prediction targets) from the raw data itself — called *pretext* task

How Much Information is the Machine Given during Learning?

► “Pure” Reinforcement Learning (**cherry**)

- The machine predicts a scalar reward given once in a while.

► **A few bits for some samples**



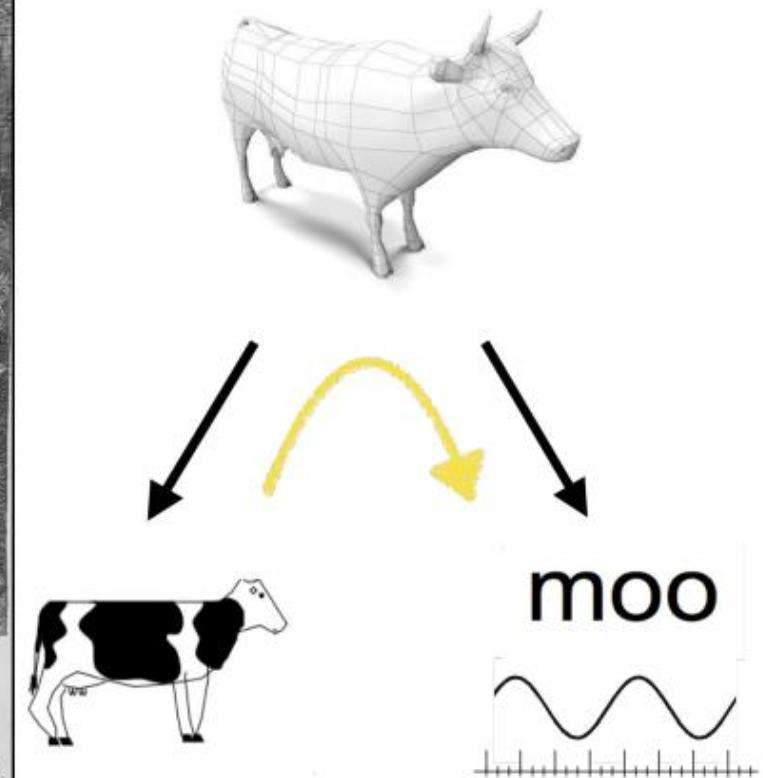
► Supervised Learning (**icing**)

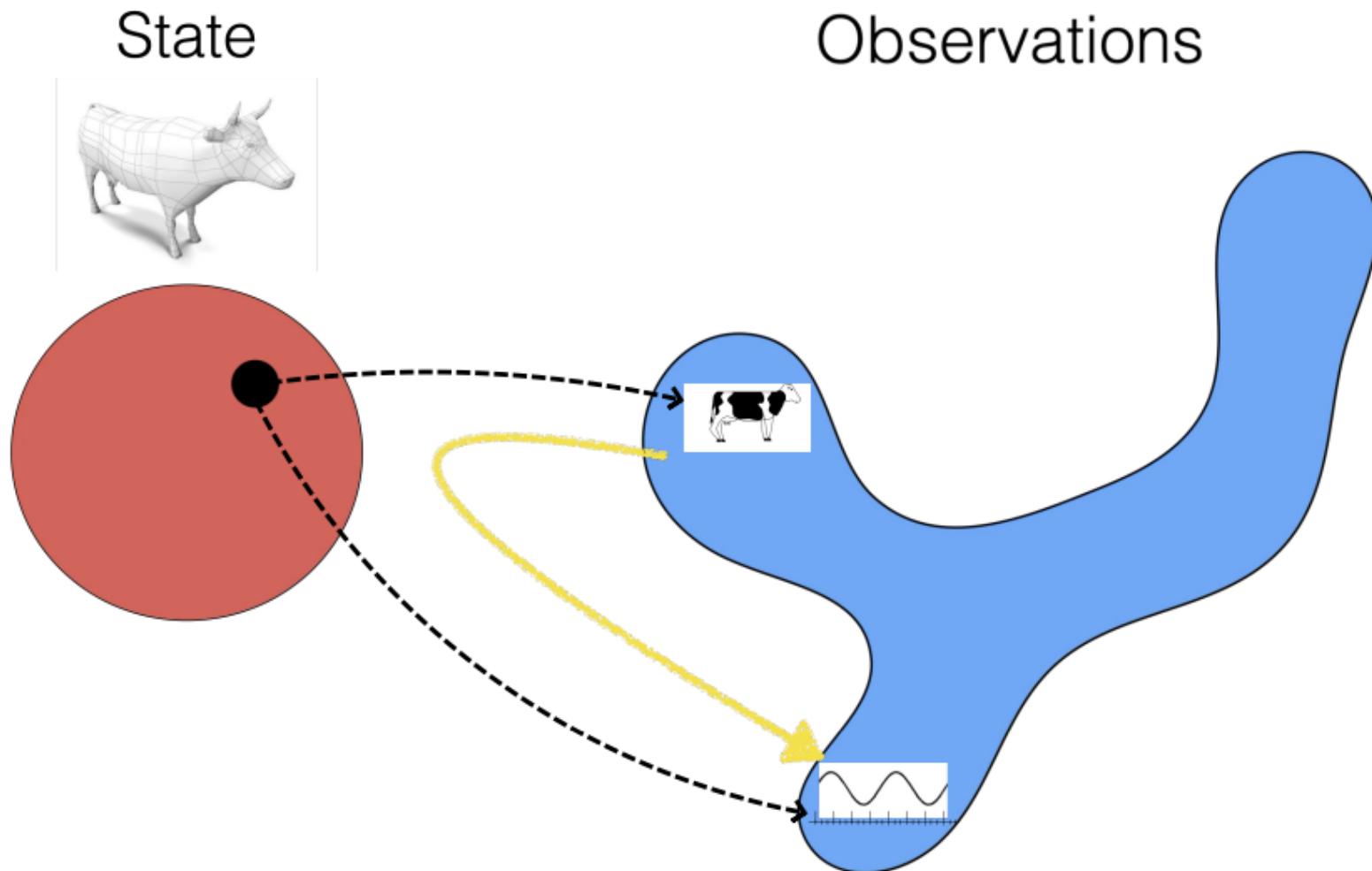
- The machine predicts a category or a few numbers for each input
- Predicting human-supplied data
- **10→10,000 bits per sample**

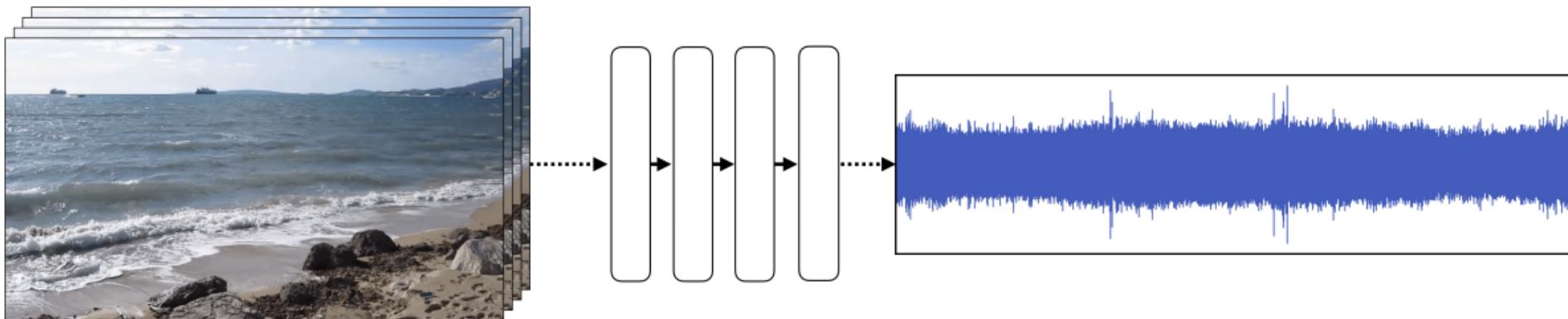
► Self-Supervised Learning (**cake génoise**)

- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- **Millions of bits per sample**

The allegory of the cave

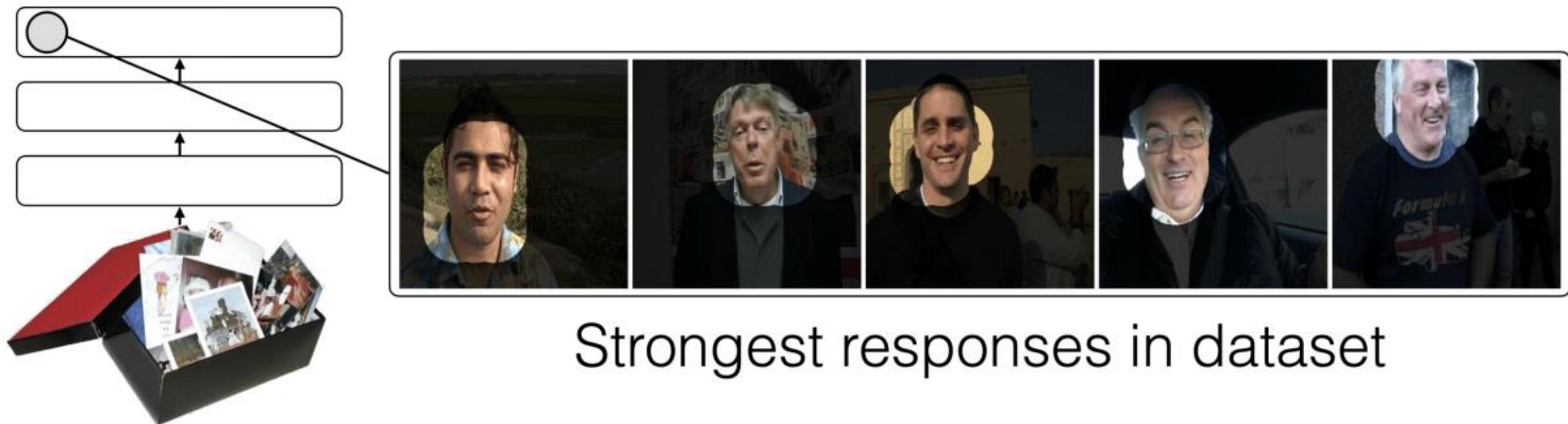






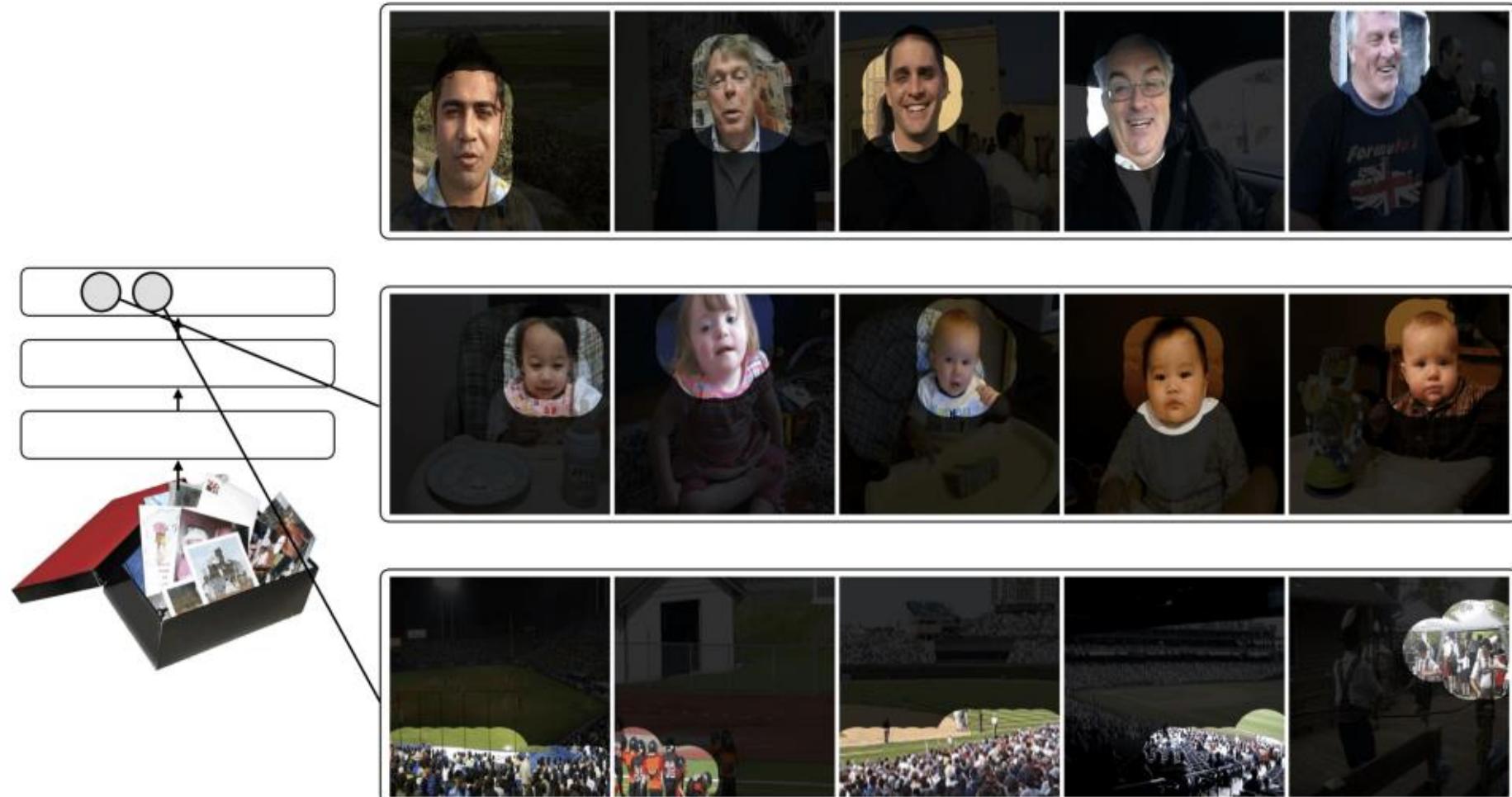
[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]
[Slide credit: Andrew Owens]

What did the model learn?

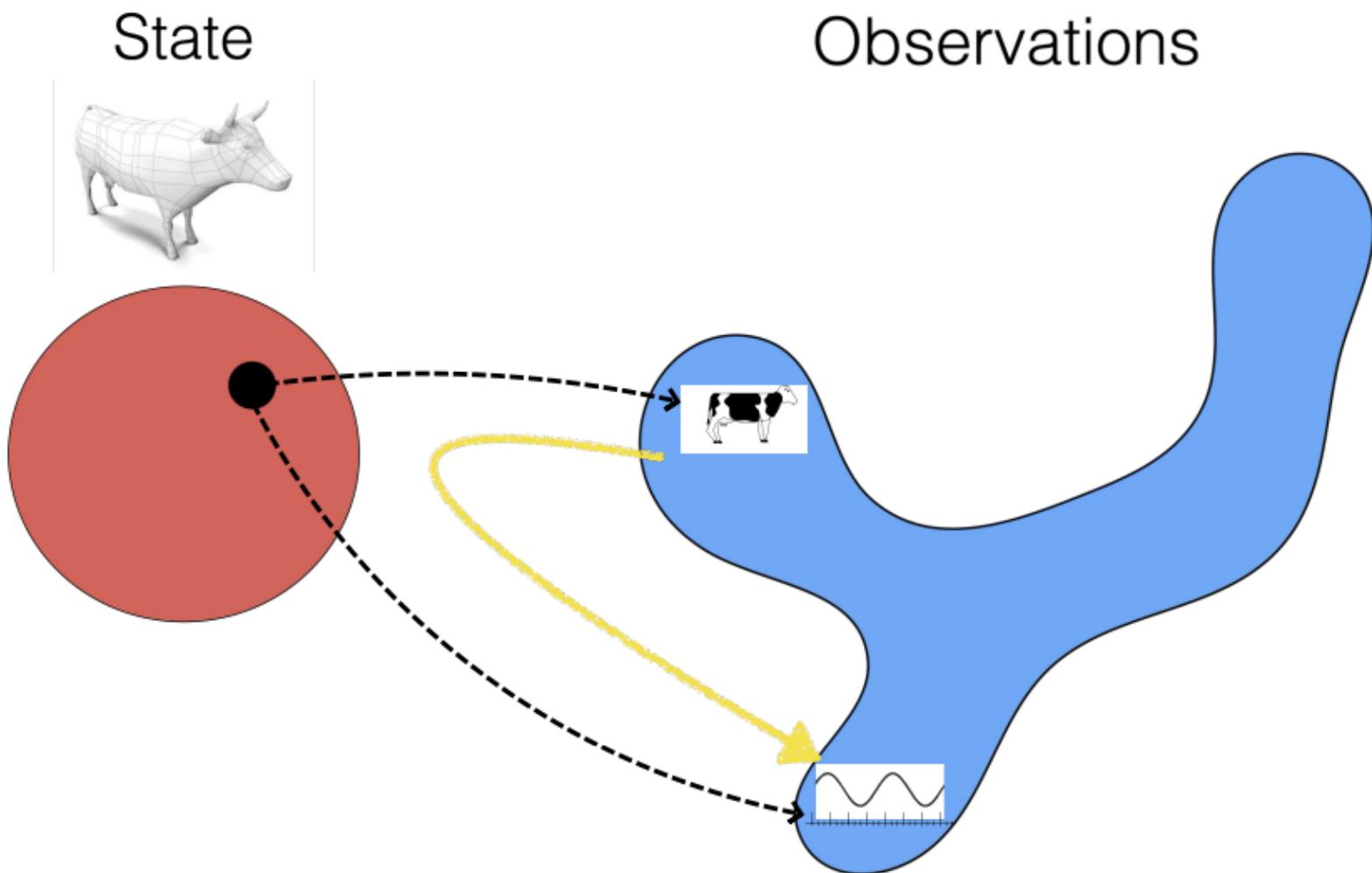


[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]

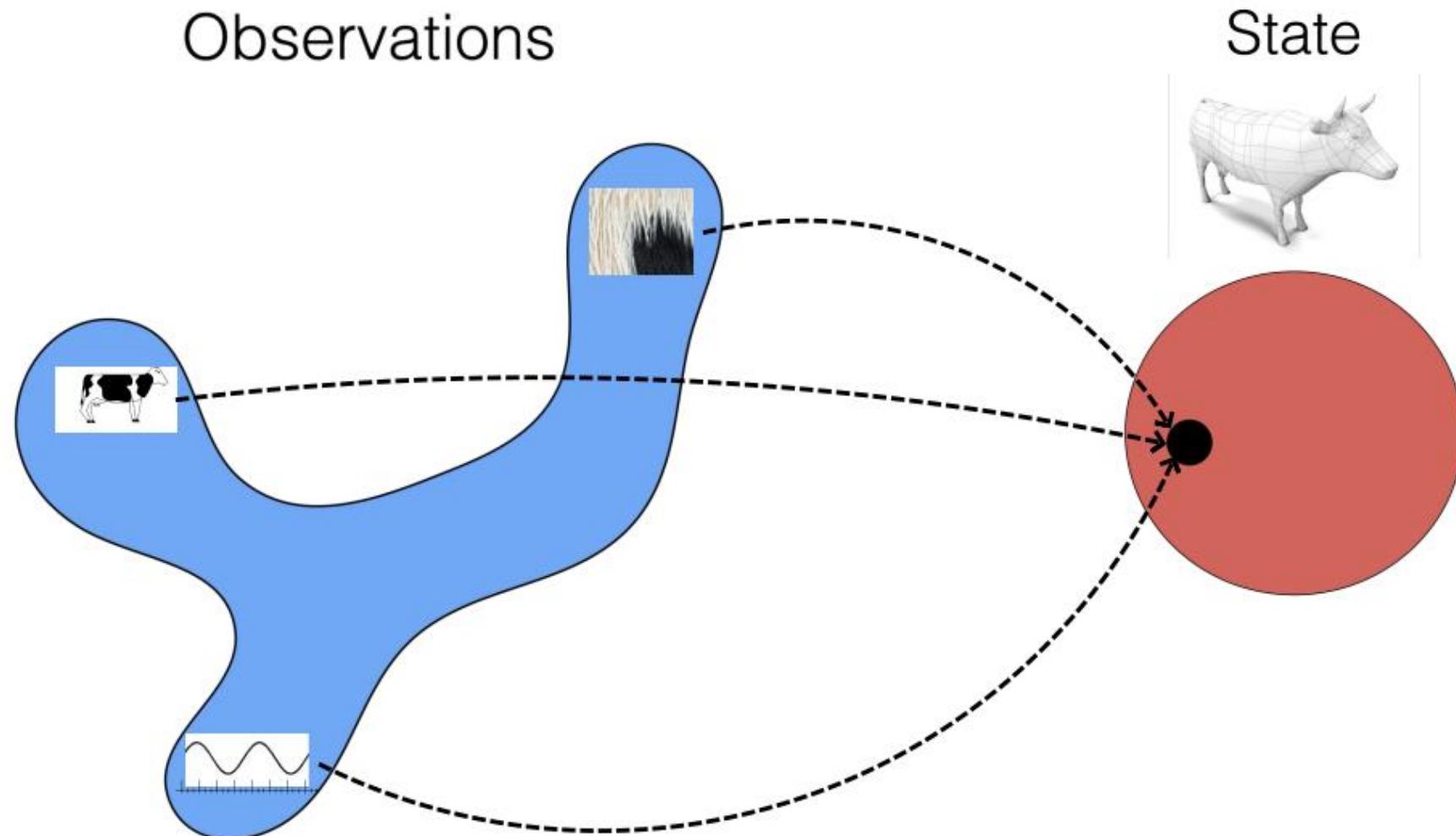
[Slide credit: Andrew Owens]



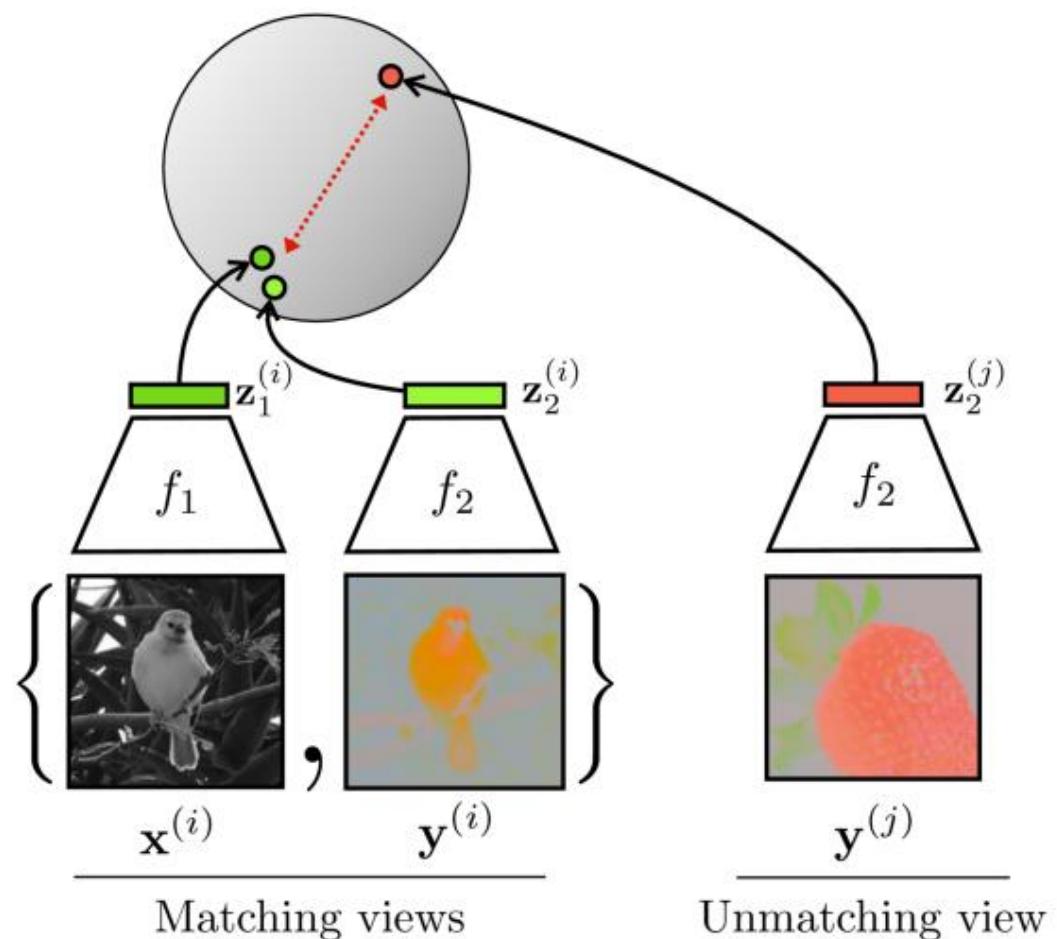
[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]
[Slide credit: Andrew Owens]

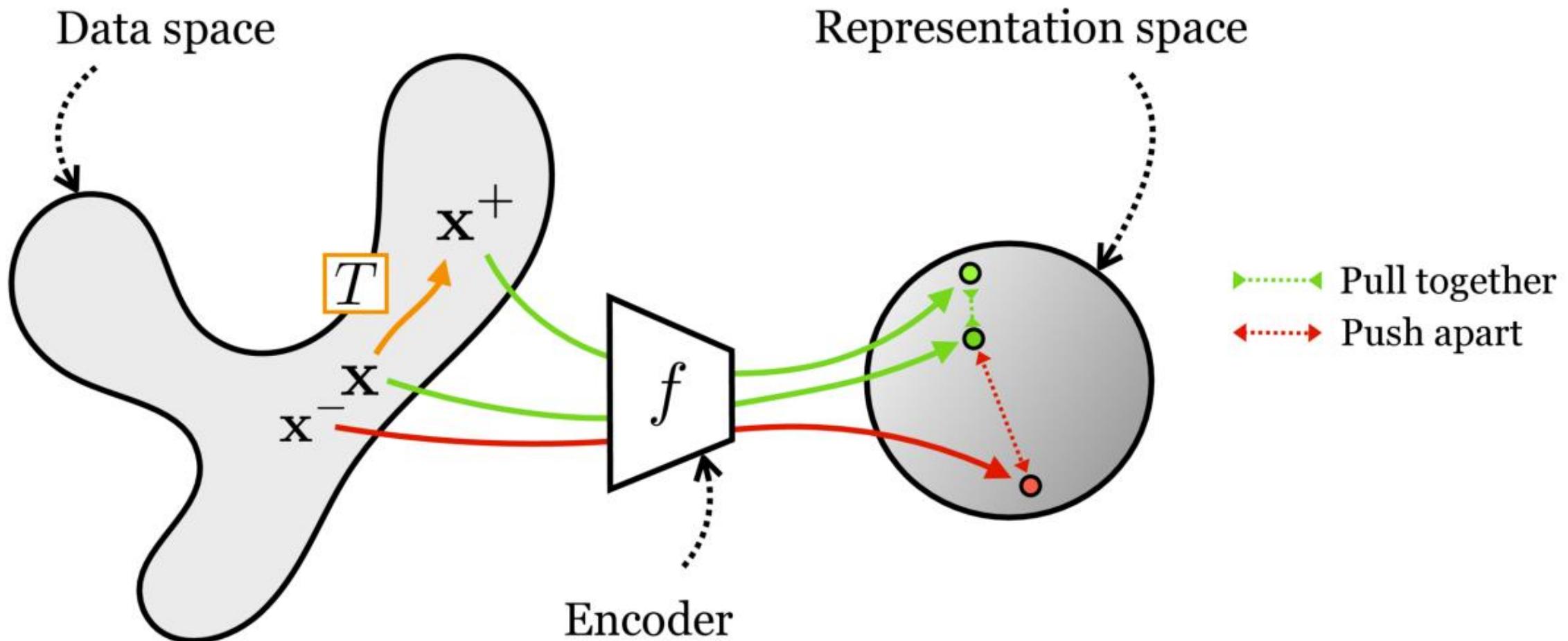


Contrastive learning



Contrastive learning





Summary

- We looked at the mechanics of neural net last time. Today we see deep nets learn representations, just like our brains do.
- This is useful because representations transfer — they act as prior knowledge that enables quick learning on new tasks.
- Representations can also be learned without labels, e.g. as we do in unsupervised, or self-supervised learning. This is great since labels are expensive and limiting.
- Without labels there are many ways to learn representations. We saw today:
 - representations as compressed codes, auto-encoder with bottleneck
 - (representations that are shared across sensory modalities)
 - (representations that are predictive of their context)