https://introml.mit.edu/

6.390 Intro to Machine Learning

Lecture 7: Neural Networks II, Auto-encoders

Shen Shen

October 11, 2024

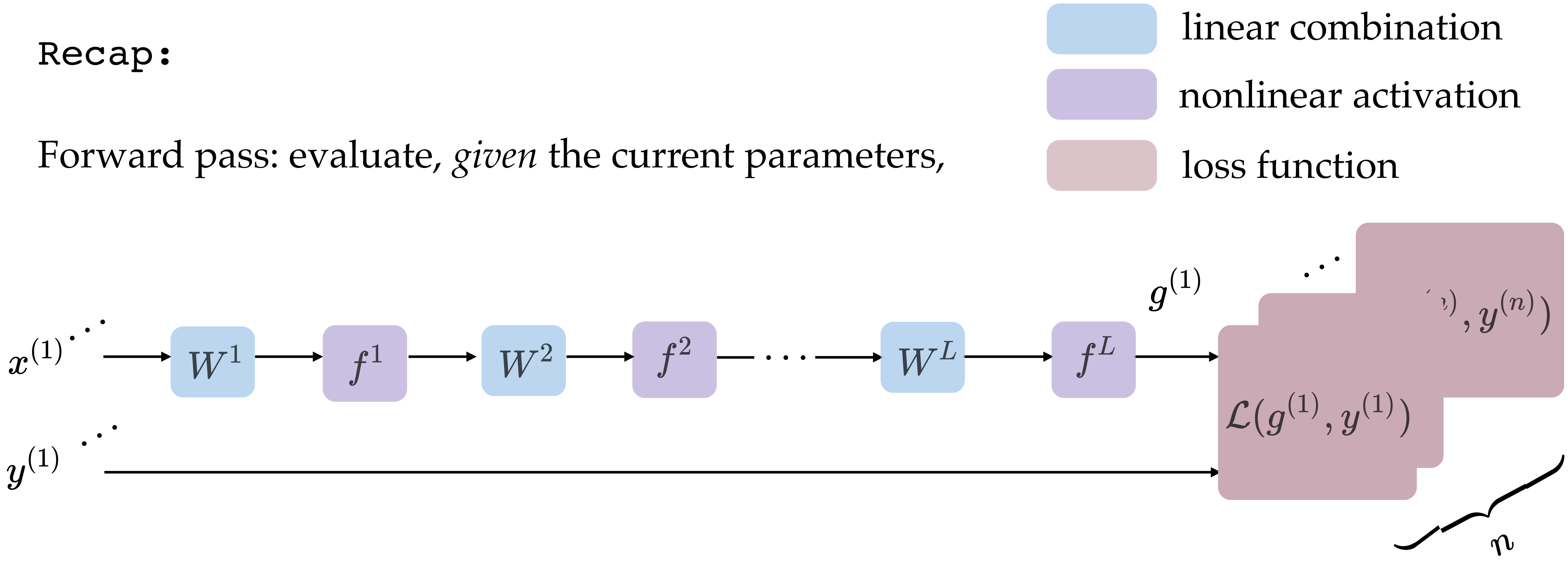
(slides adapted from Phillip Isola)

Outline

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

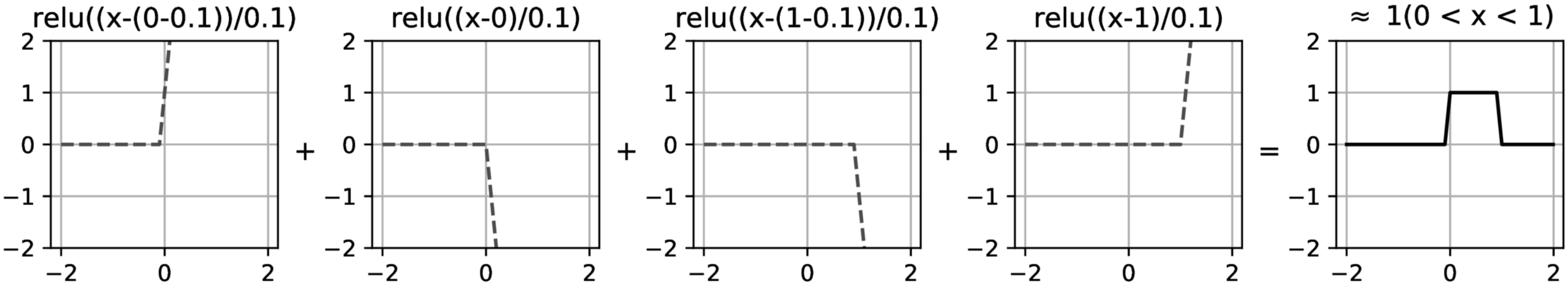
`Recap:`

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

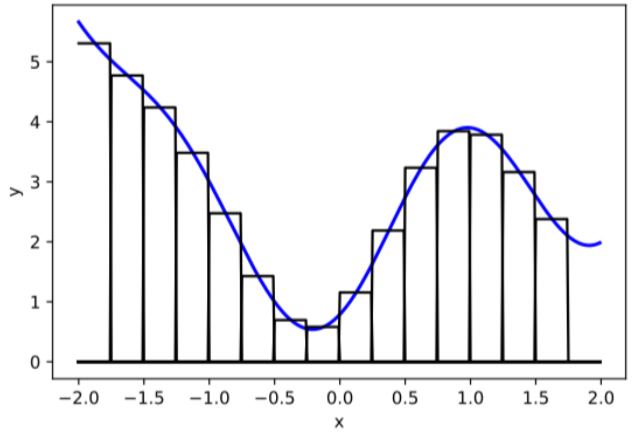


- the model output $g^{(i)} = f^L\left(\dots f^2\left(f^1(\mathbf{x}^{(i)}; \mathbf{W}^1); \mathbf{W}^2\right); \dots \mathbf{W}^L\right)$
- the loss incurred on the current data $\mathcal{L}(g^{(i)}, y^{(i)})$
- the training error $J = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(g^{(i)}, y^{(i)})$

compositions of ReLU(s) can be quite expressive

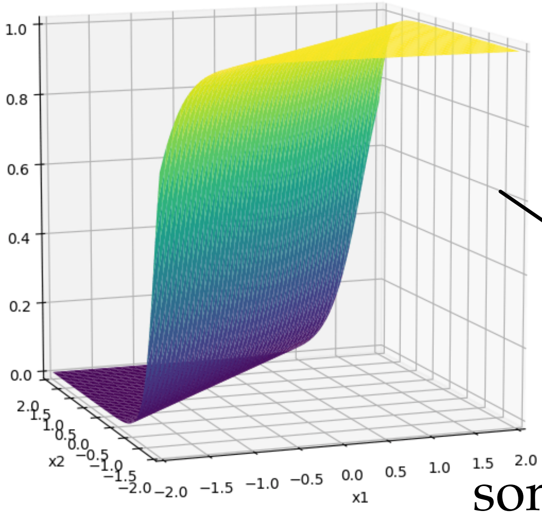


in fact, asymptotically, can approximate any function!

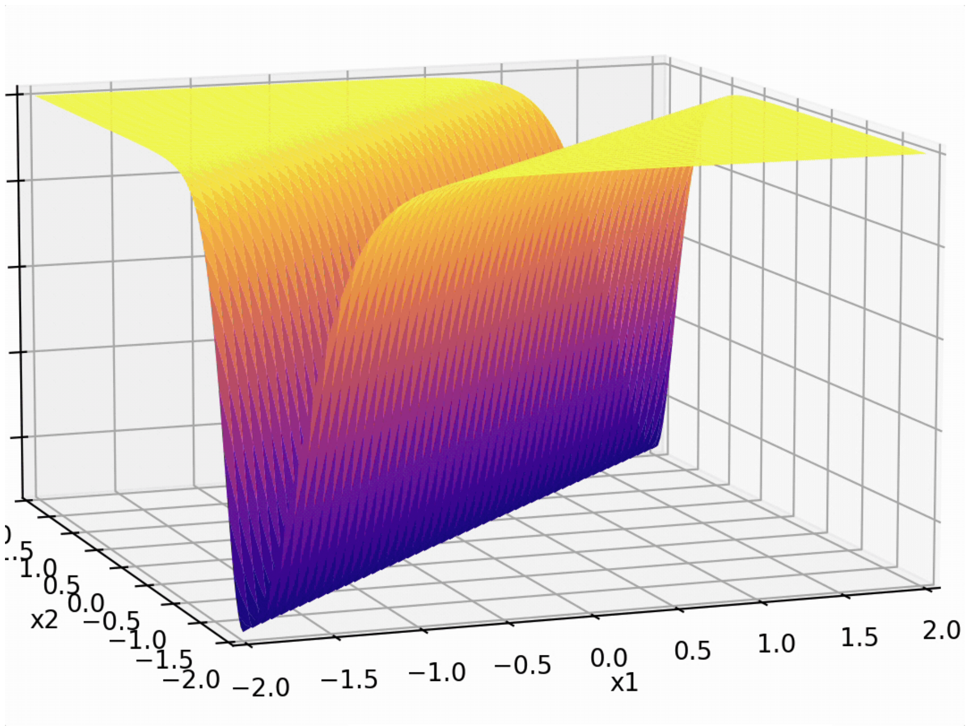


(image credit: Phillip Isola)

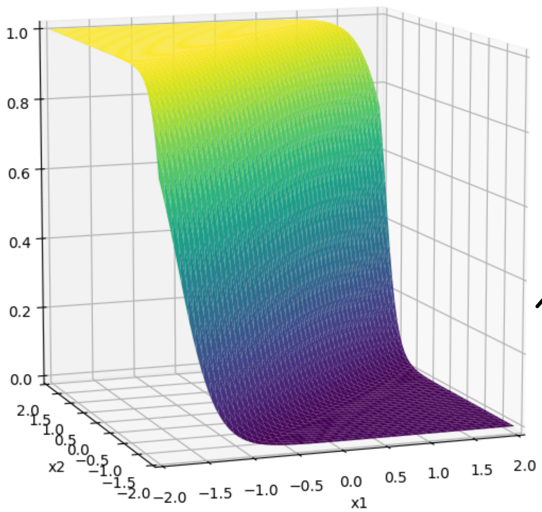
Recap: $\sigma_1 = \sigma(5x_1 + -5x_2 + 1)$

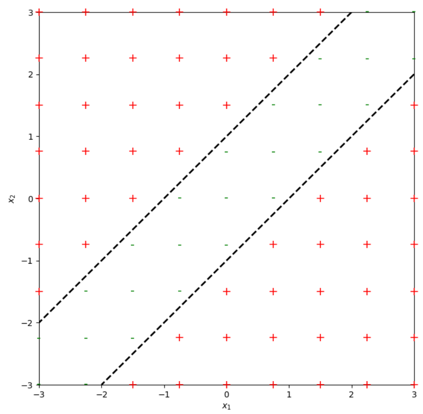


some weighted sum



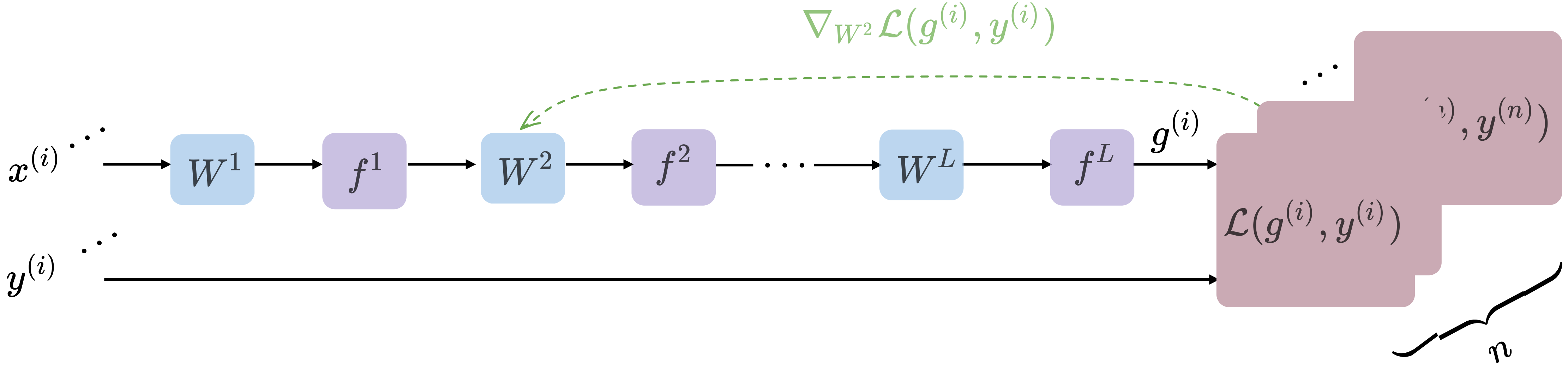
$\sigma_2 = \sigma(-5x_1 + 5x_2 + 1)$





`Recap:`

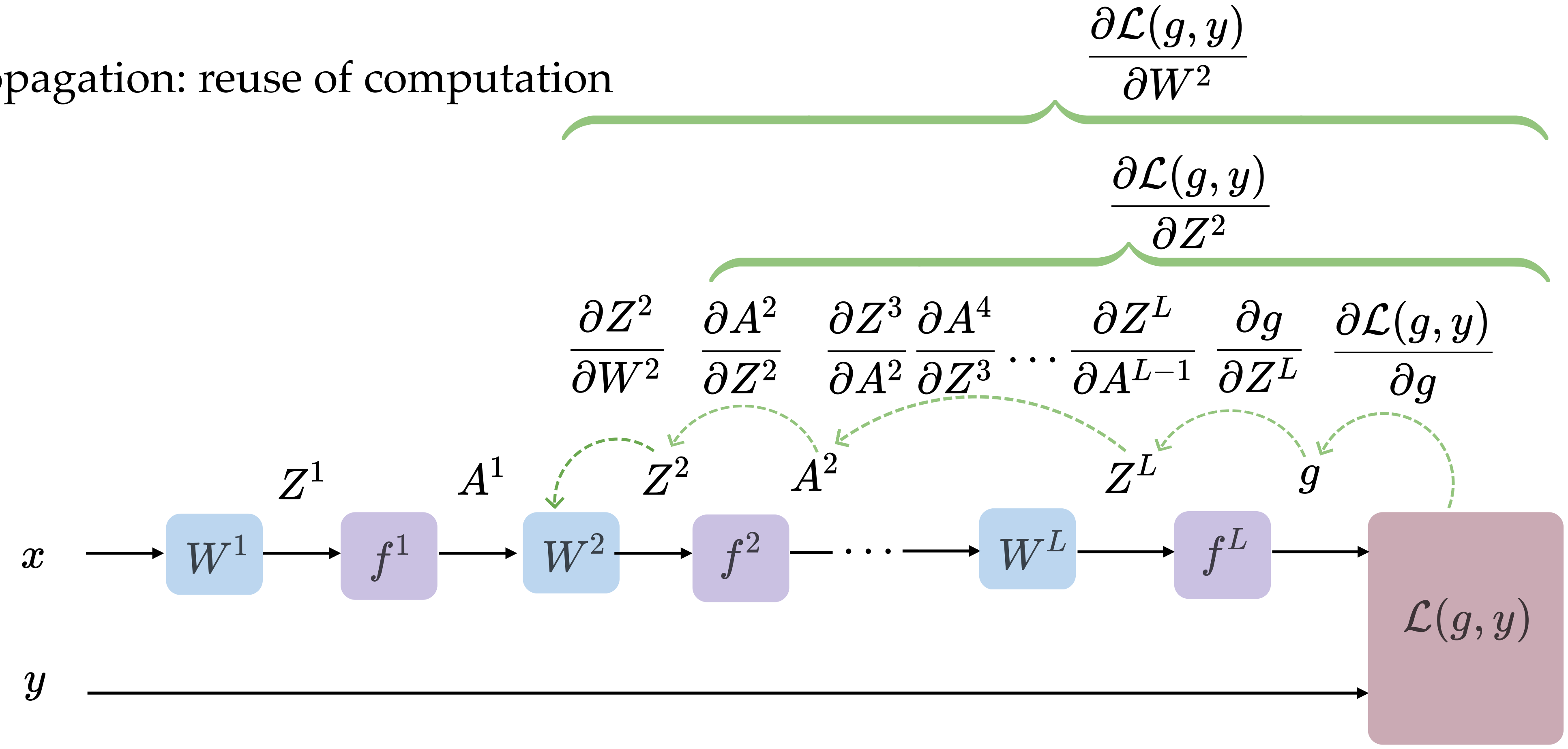
Backward pass: run SGD to update the parameters, e.g. to update W^2



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

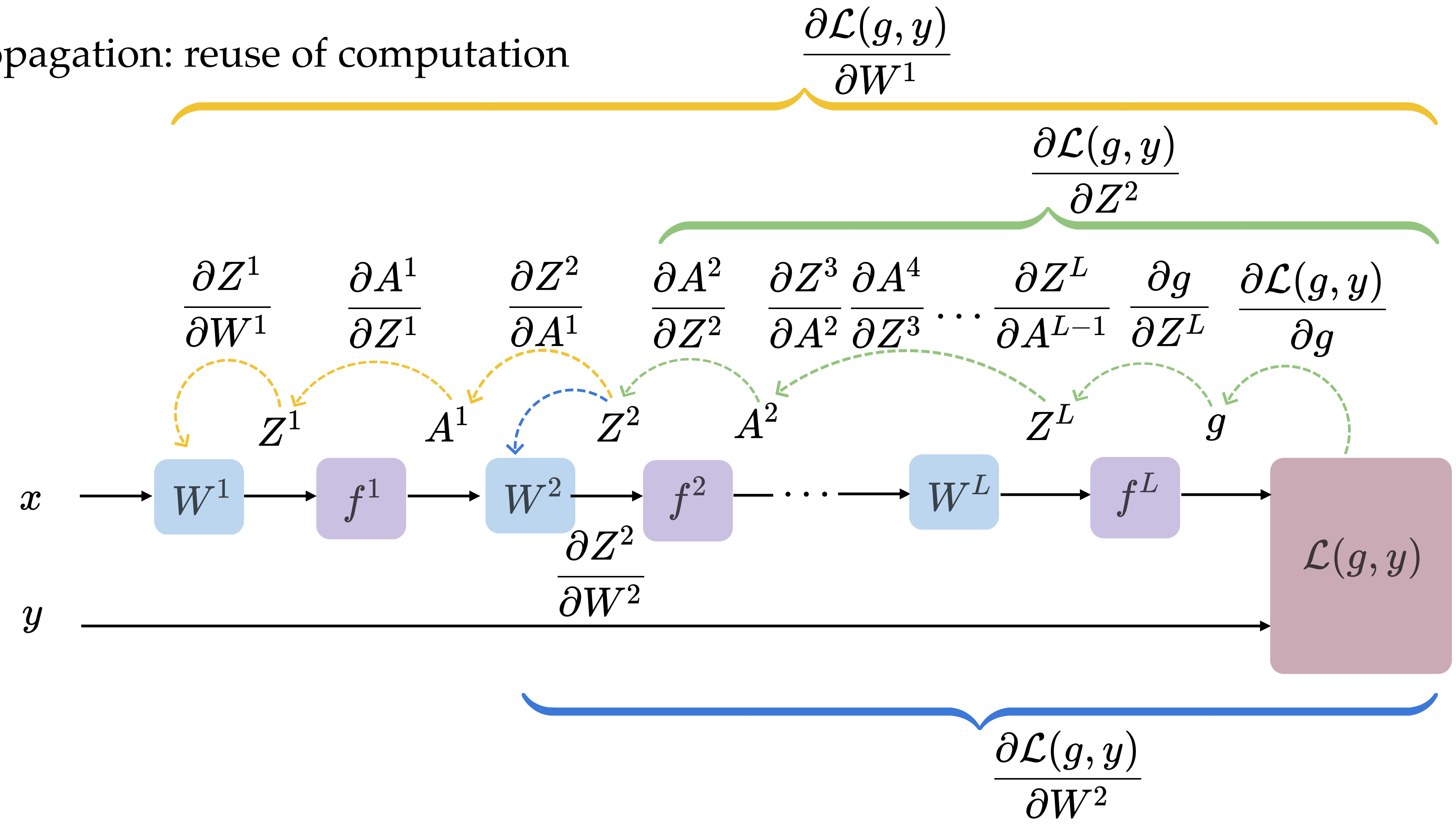
```
Recap:
```

back propagation: reuse of computation



Recap:

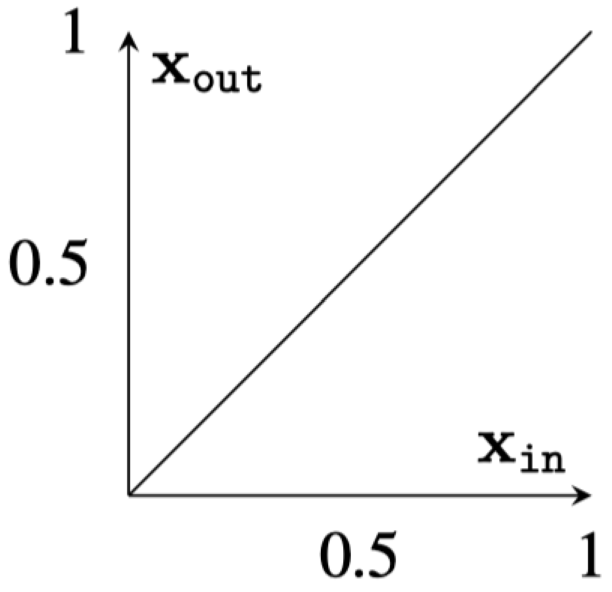
back propagation: reuse of computation



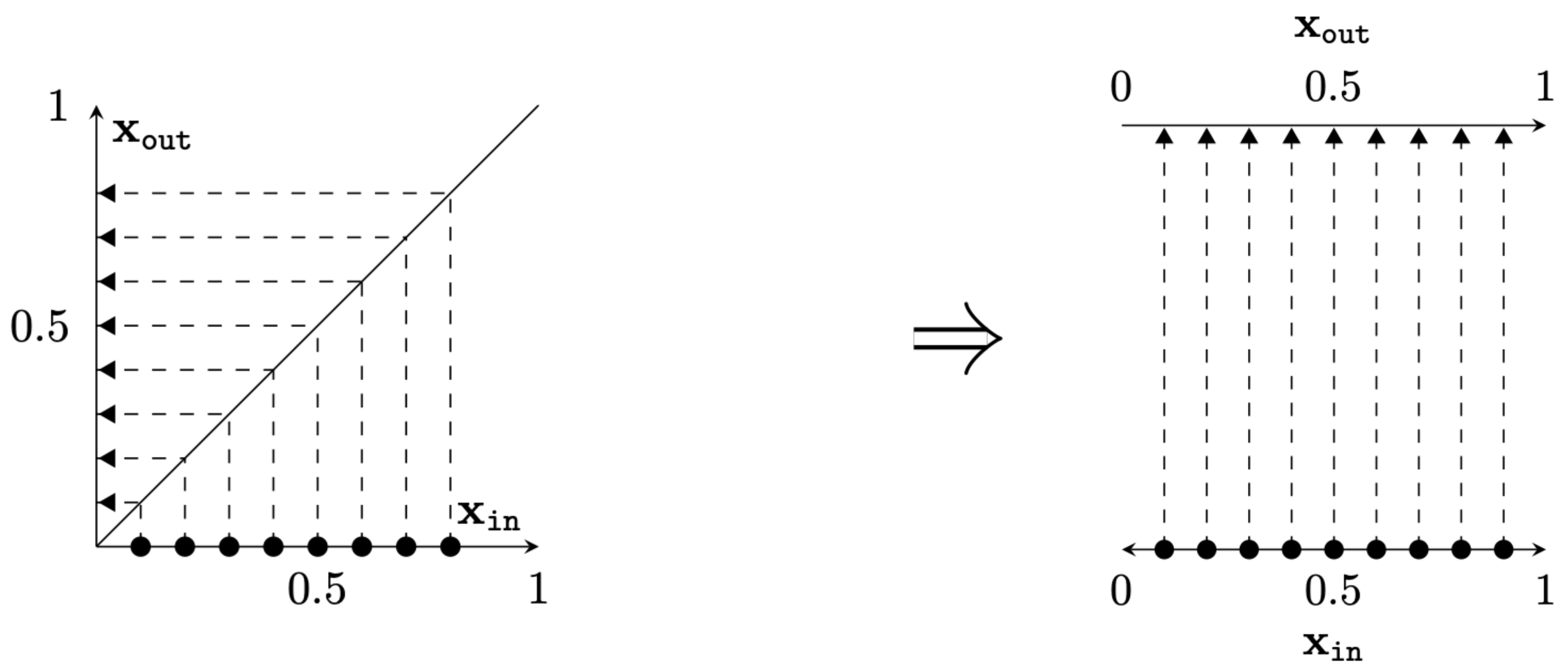
Outline

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

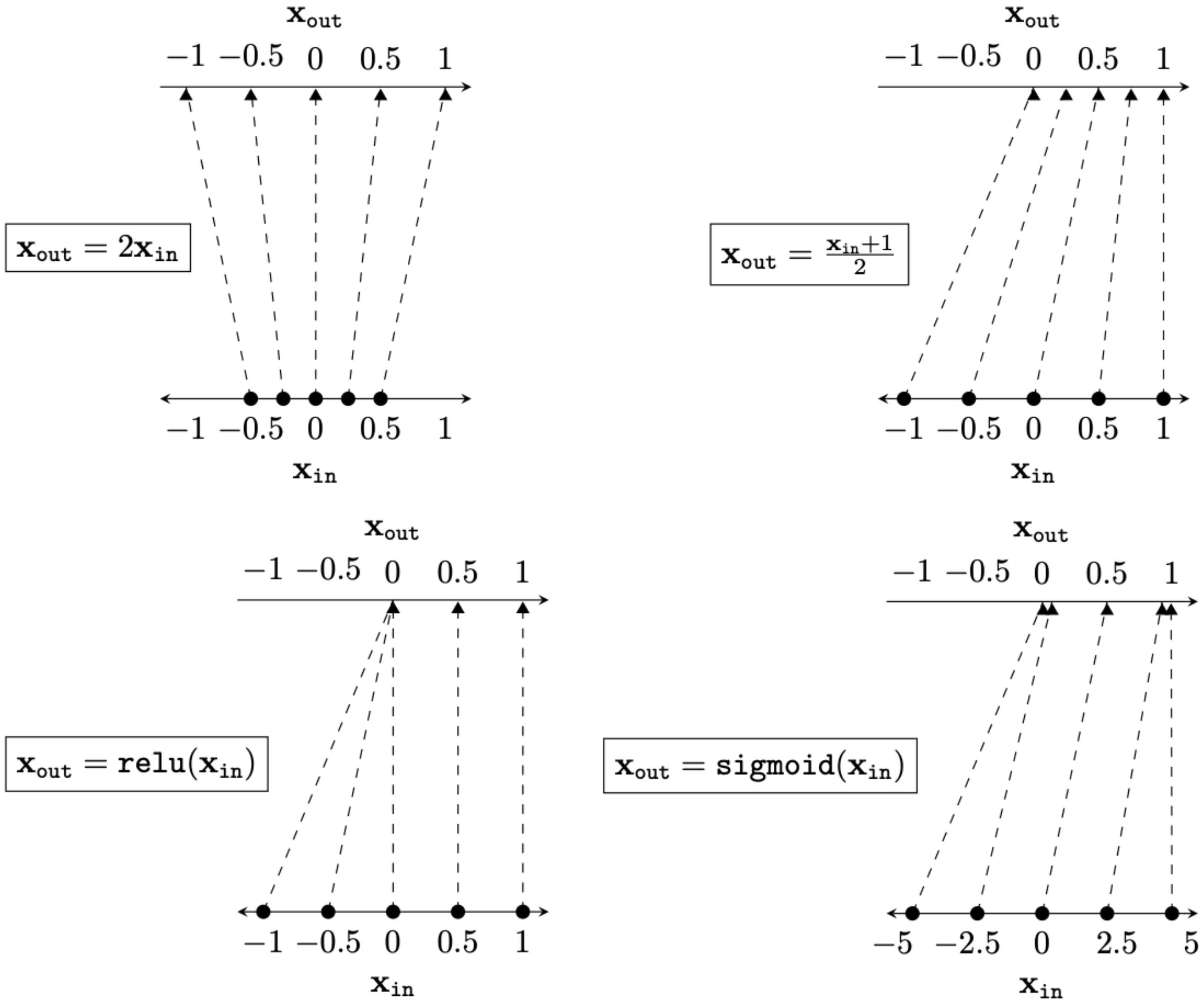
Two different ways to visualize a function



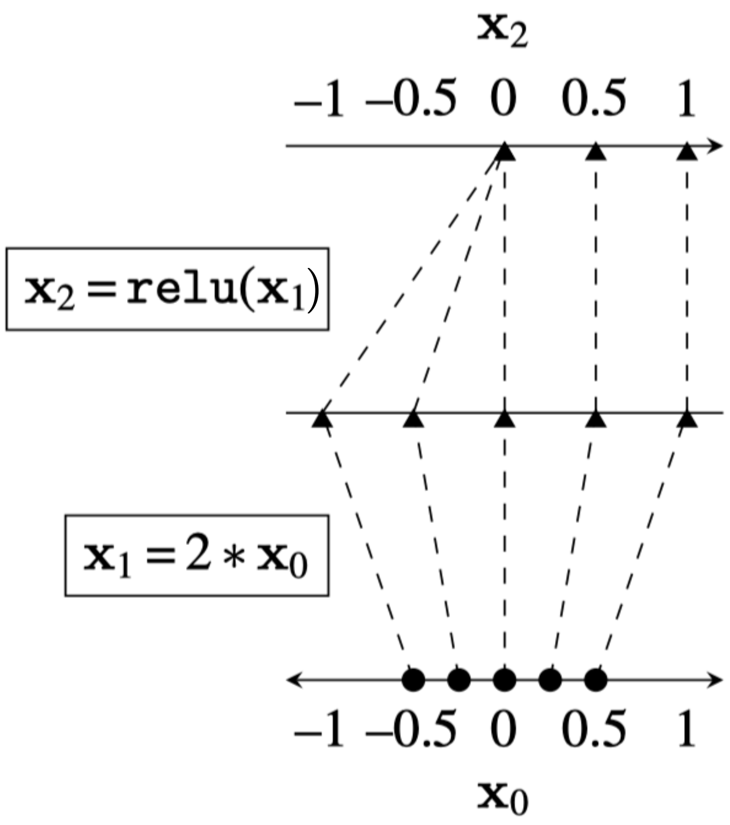
Two different ways to visualize a function



Representation transformations for a variety of neural net operations



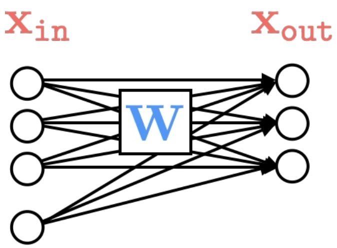
and stack of neural net operations



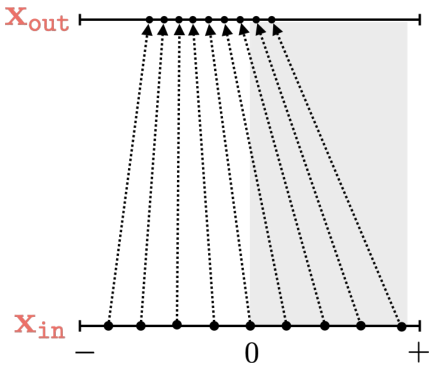
Parameters

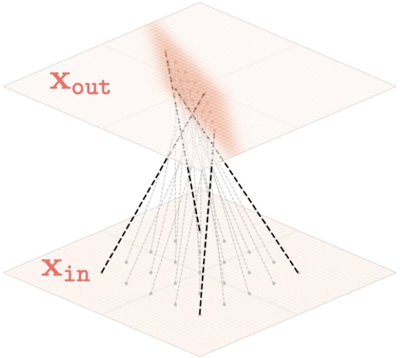
wiring graph | equation | mapping 1D | mapping 2D

linear

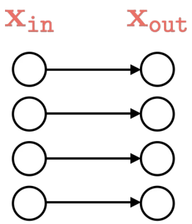


$$\mathbf{x}_{\mathtt{out}} = \mathbf{W}\mathbf{x}_{\mathtt{in}}$$

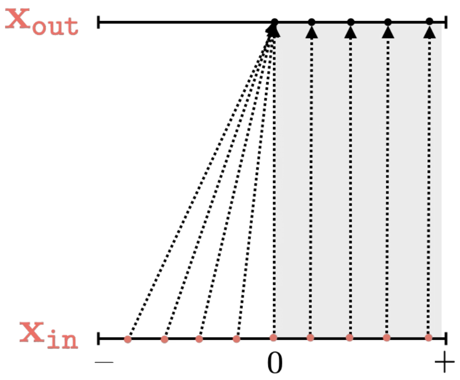


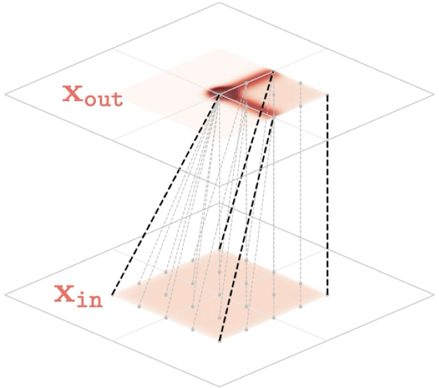


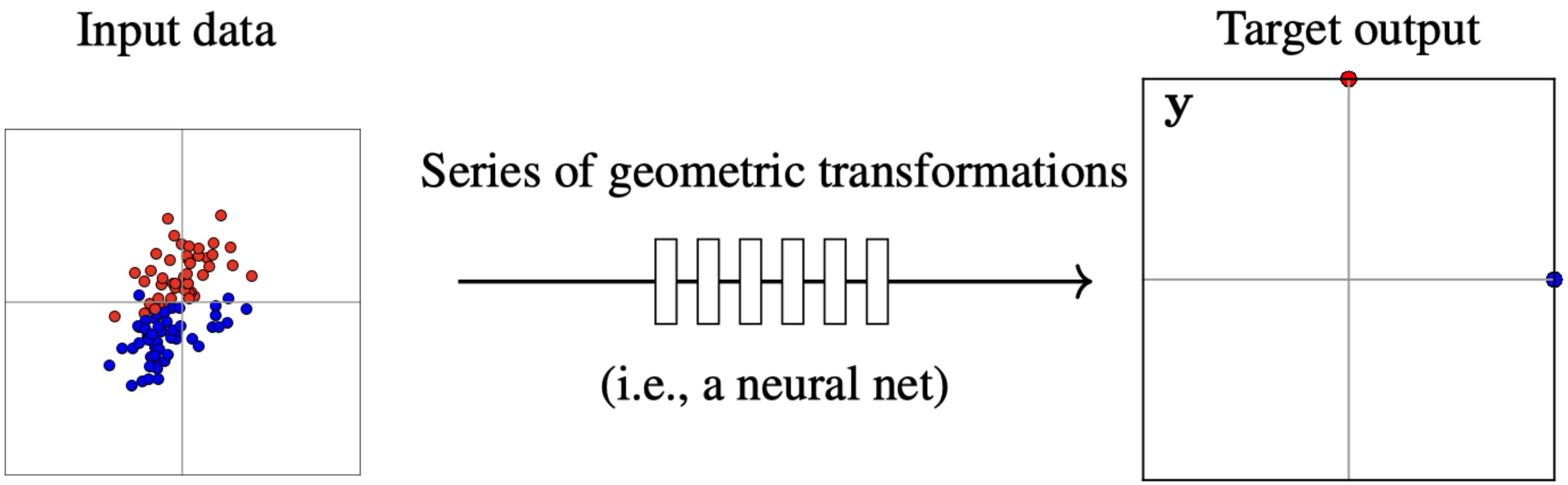
relu



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






Input data
Series of geometric transformations
(i.e., a neural net)
Target output
y

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

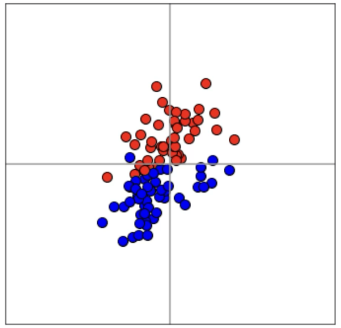
$$z_2 = \text{ linear } (a_1)$$

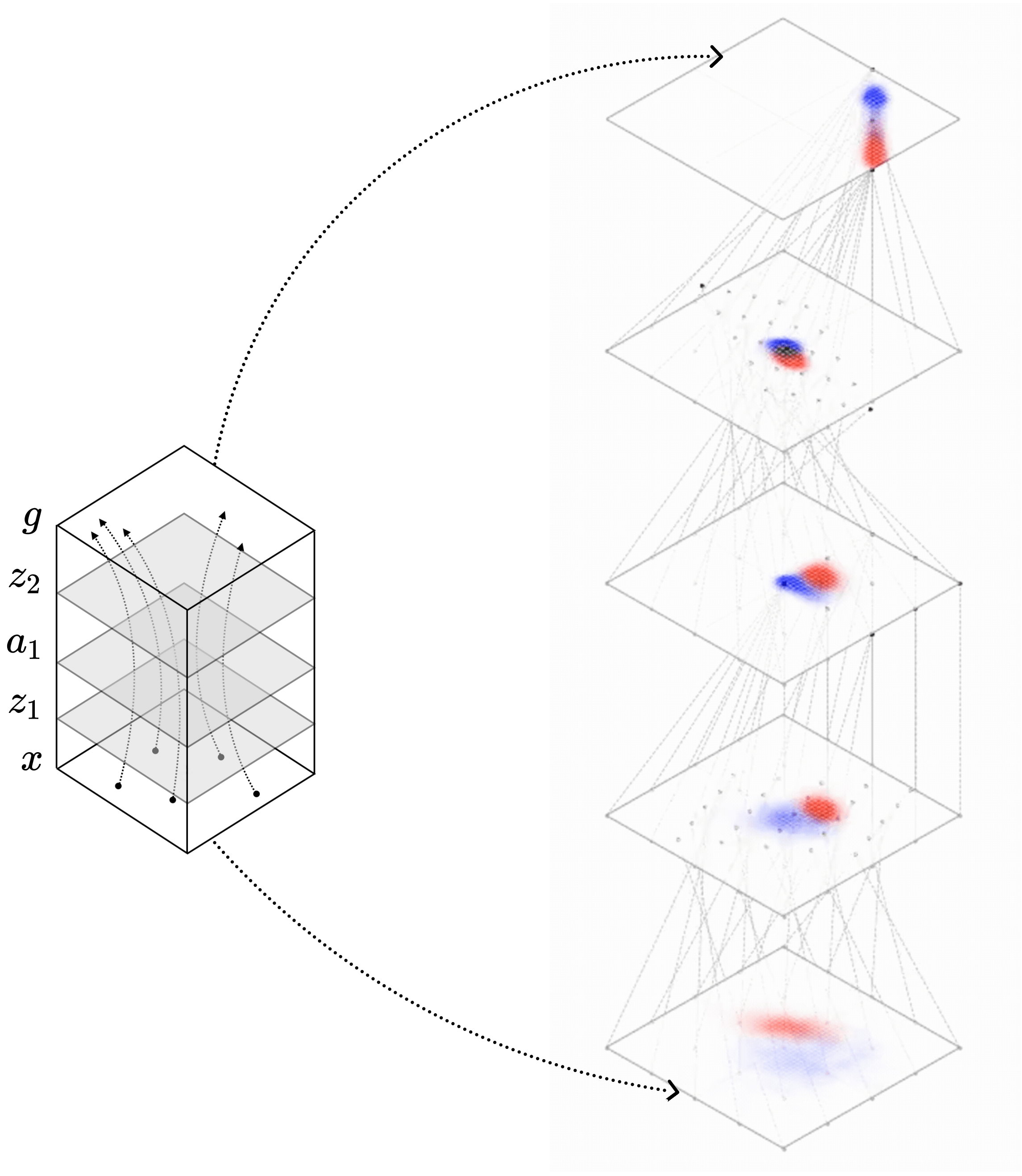
$$a_1 = \text{ ReLU}(z_1)$$

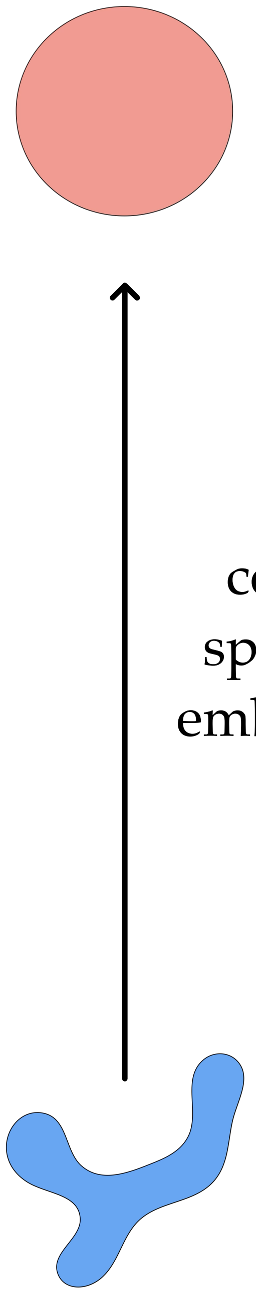
$$z_1 = \text{ linear } (x)$$

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

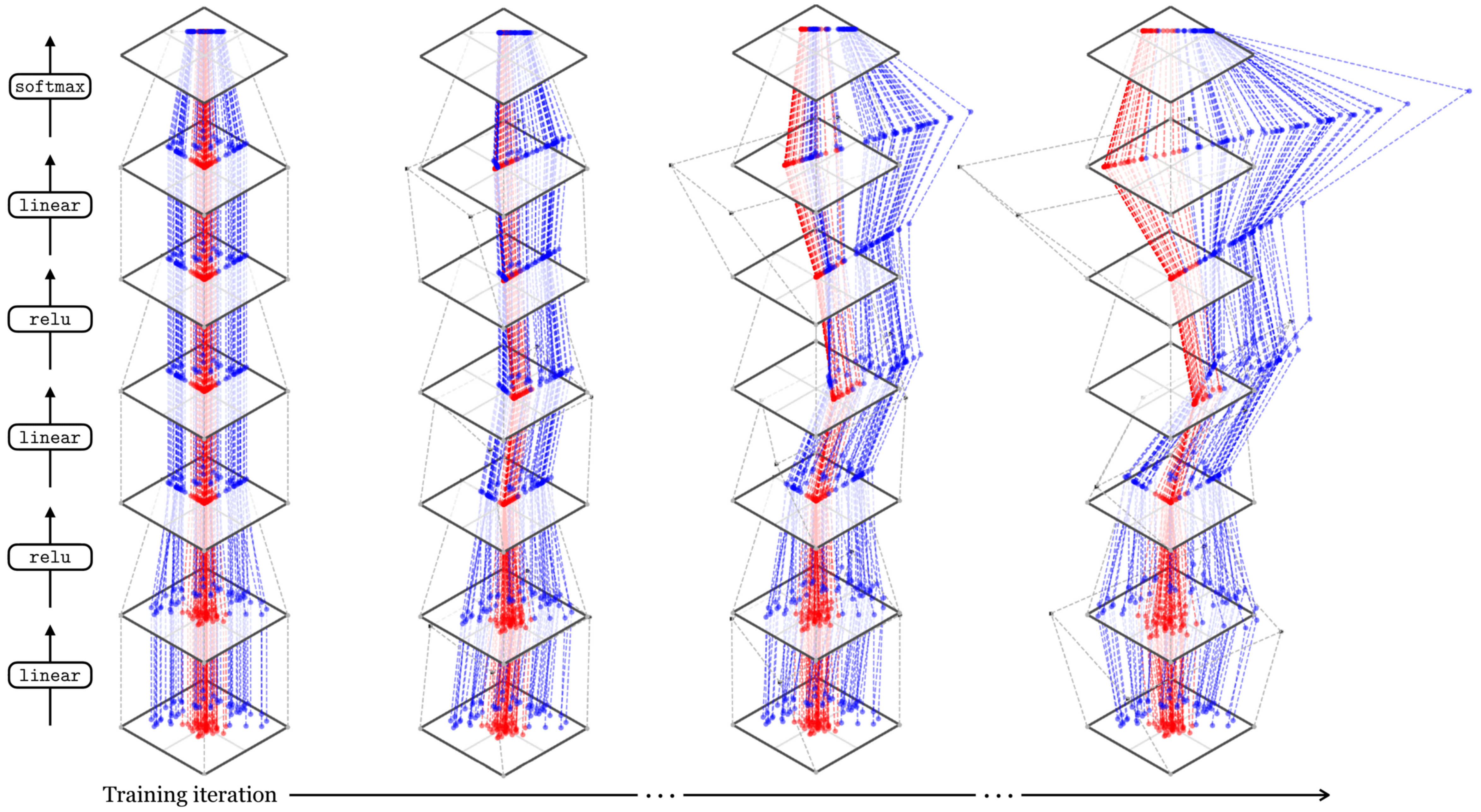
Training data







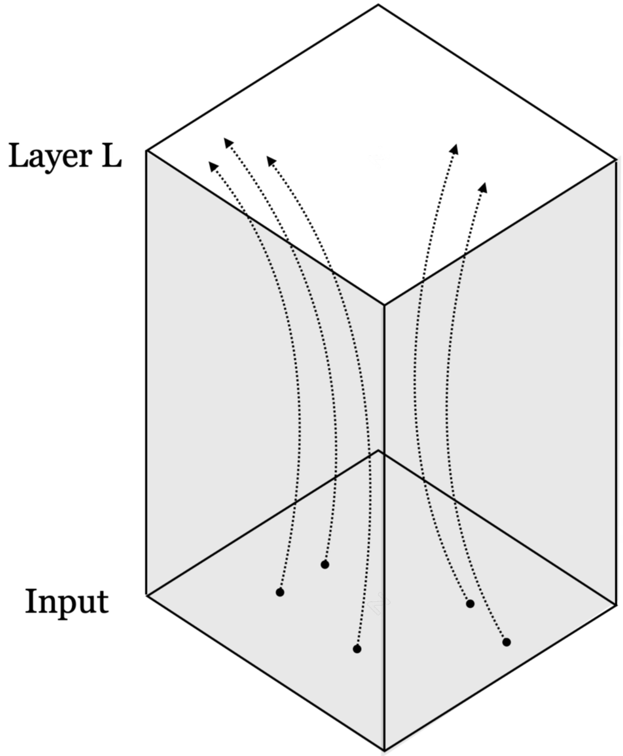
maps from complex data space to simple embedding space


softmax
linear
relu
linear
relu
linear
Training iteration

Neural networks are representation learners

Deep nets transform datapoints, layer by layer

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



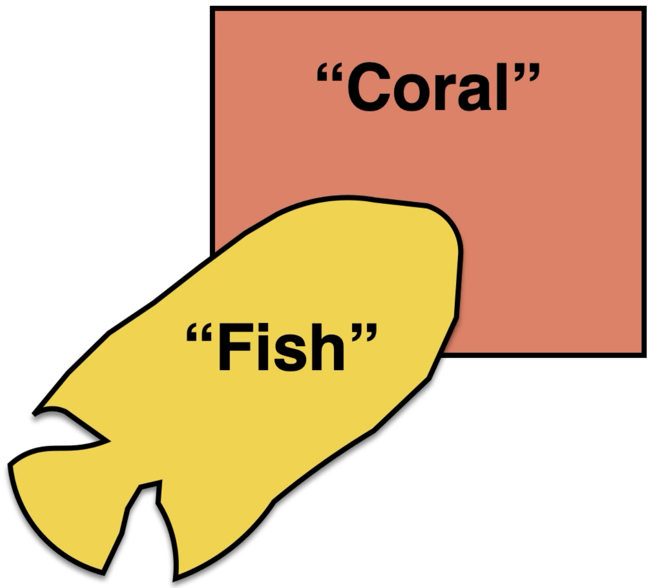
humans also learn representations

"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:

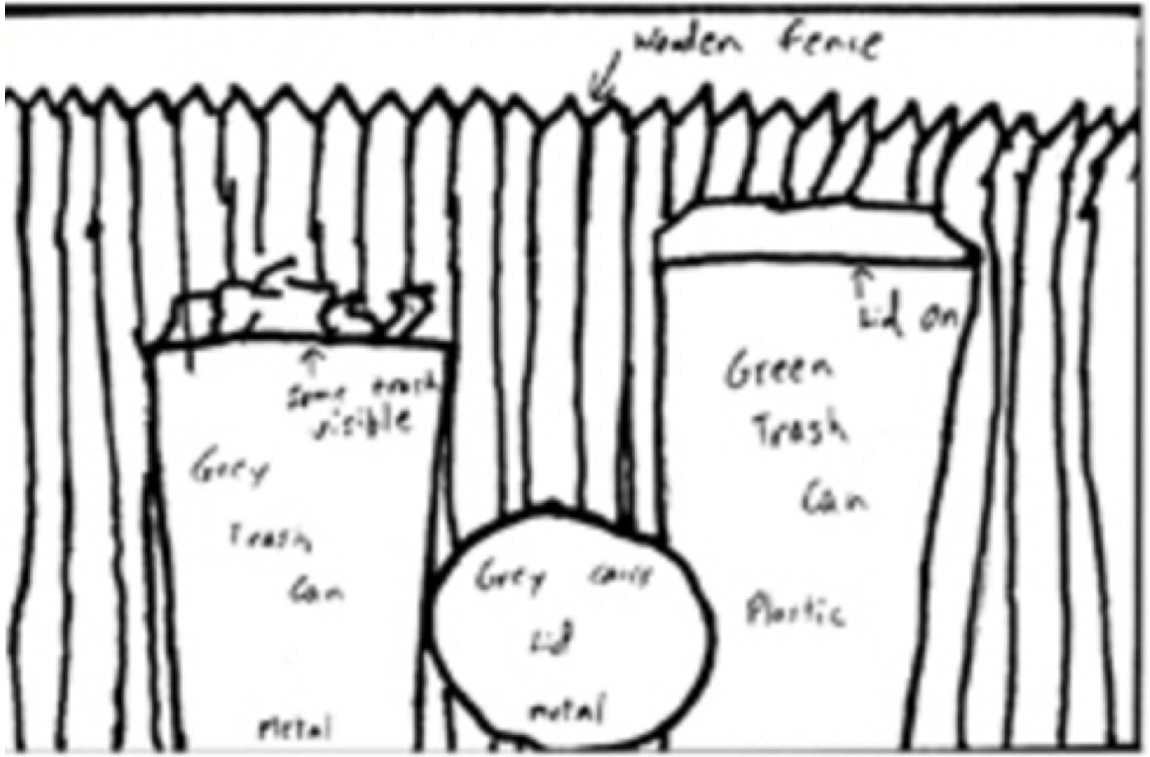
- Compact (*minimal*)
- Explanatory (*roughly sufficient*)



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

Observed image

Drawn from memory



[Bartlett, 1932]
[Intraub & Richardson, 1989]

[https://www.behance.net/gallery/35437979/Velocipedia]

Outline

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

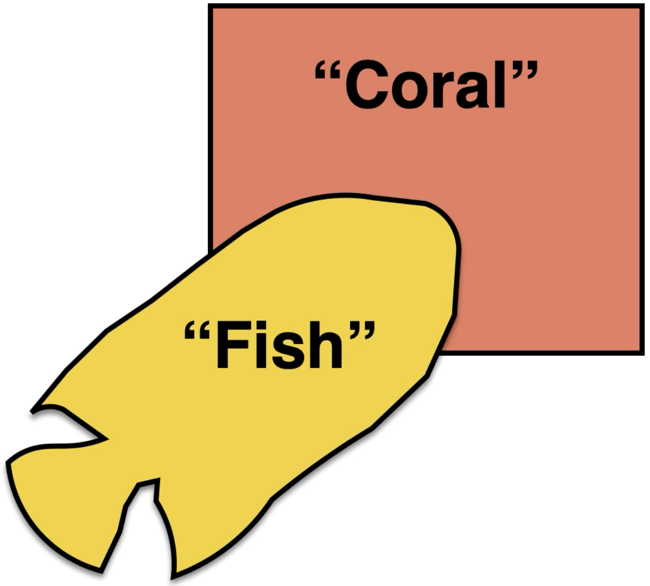
Good representations are:

Auto-encoders try to achieve these

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

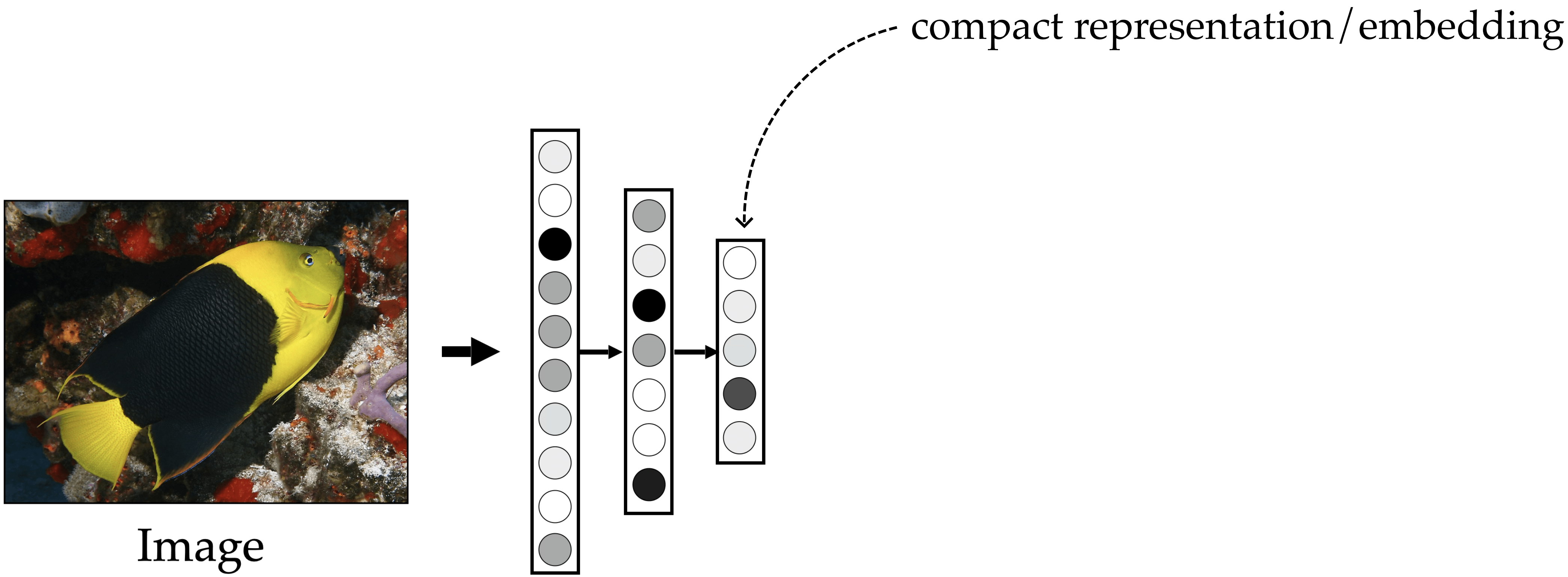
these may just emerge as well

- Disentangled (*independent* factors)
- Interpretable
- Make *subsequent problem solving easy*



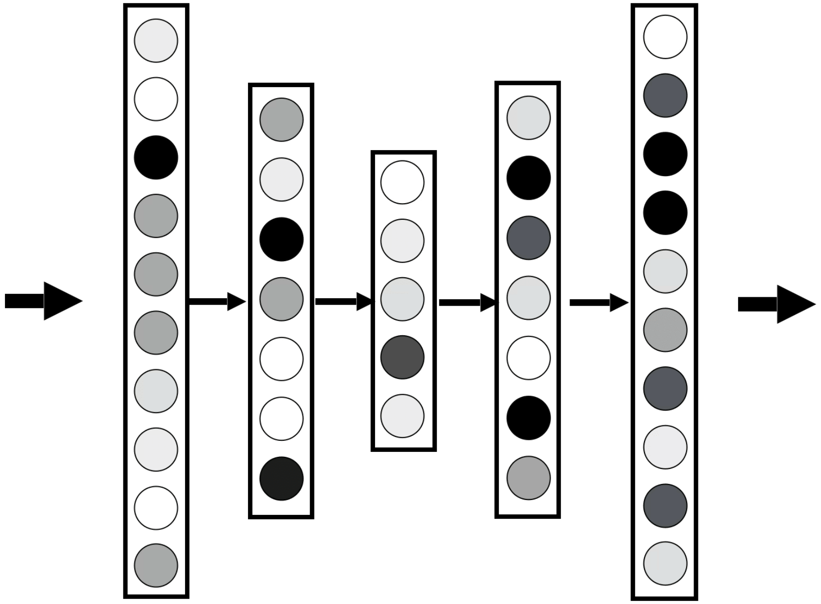
[See "Representation Learning", Bengio 2013, for more commentary]

Auto-encoder



Auto-encoder

Image



Reconstructed image

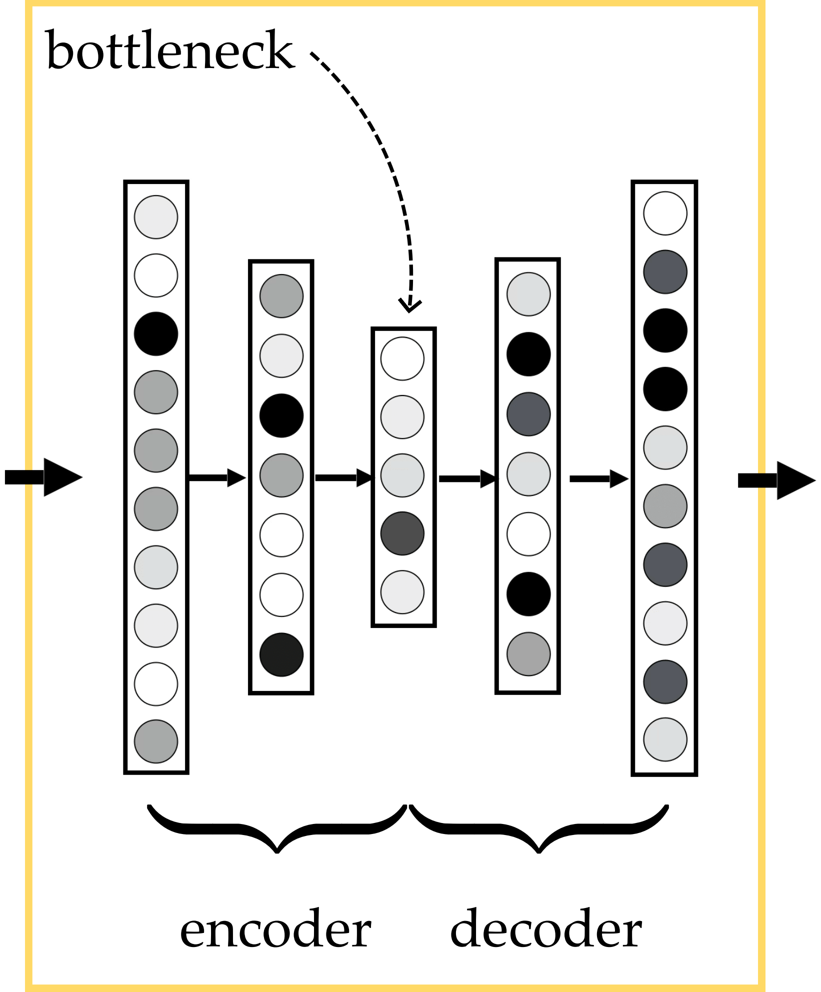
"What I cannot create, I do not understand." Feynman

Auto-encoder

$$\min_{W} ||x - \tilde{x}||^2$$

x

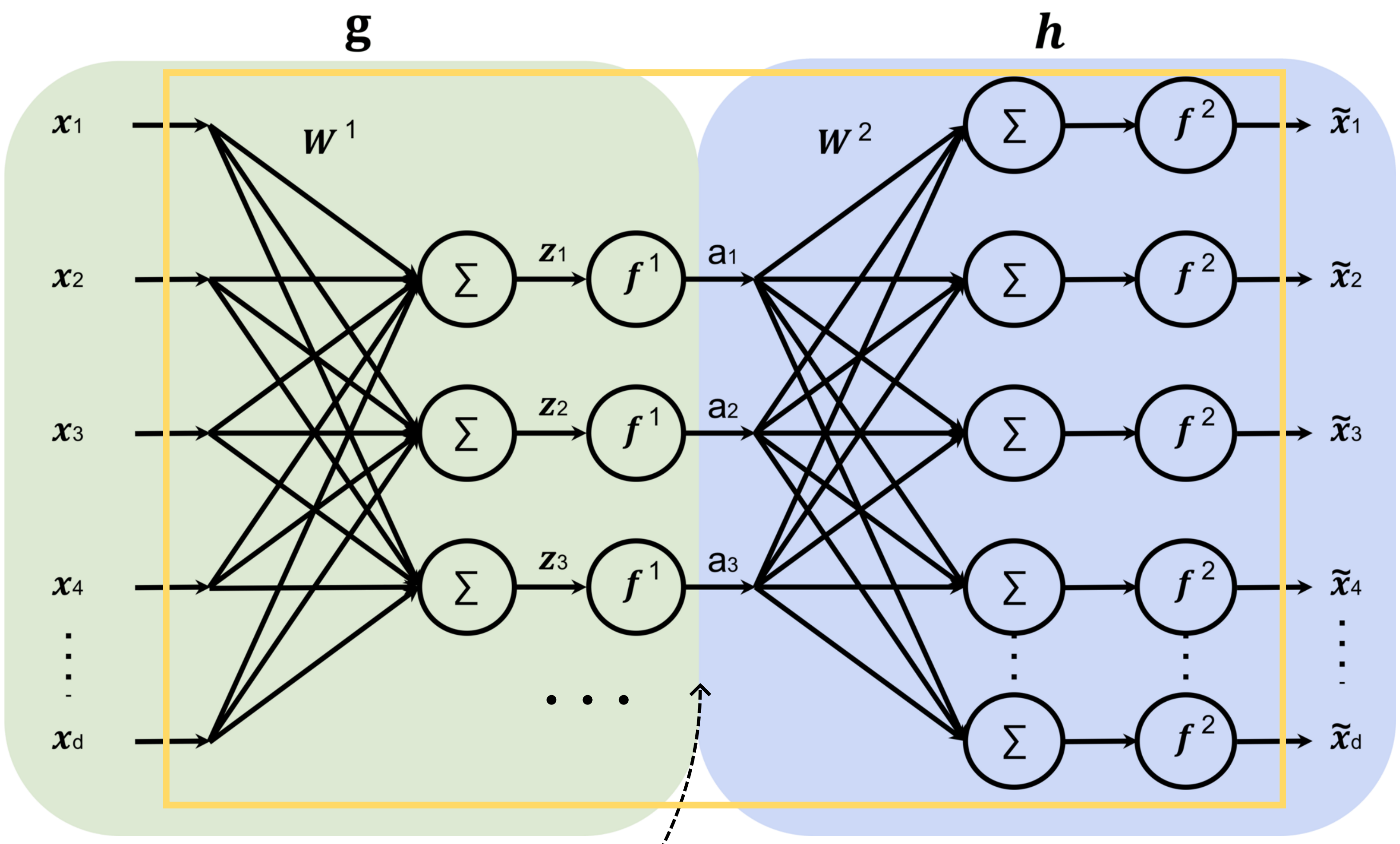
Image



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

Reconstructed image

input
$x \in \mathbb{R}^d$



output
$\tilde{x} \in \mathbb{R}^d$

bottleneck

typically, has lower dimension than d

Training Data

$$\left\{x^{(i)}\right\}_{i=1}^{n}$$

Auto-encoder

loss/objective

$$\mathcal{L}(F(\mathbf{x}), \mathbf{x}) = \|F(\mathbf{x}) - \mathbf{x}\|^2$$

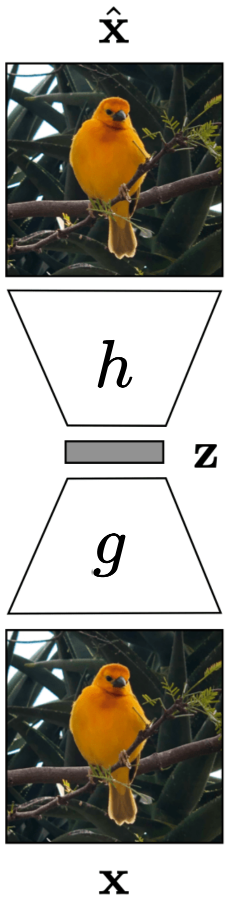
hypothesis class

$$F = g \circ h : \mathbb{R}^d \to \mathbb{R}^m \to \mathbb{R}^d$$

$m < d$

A model

f



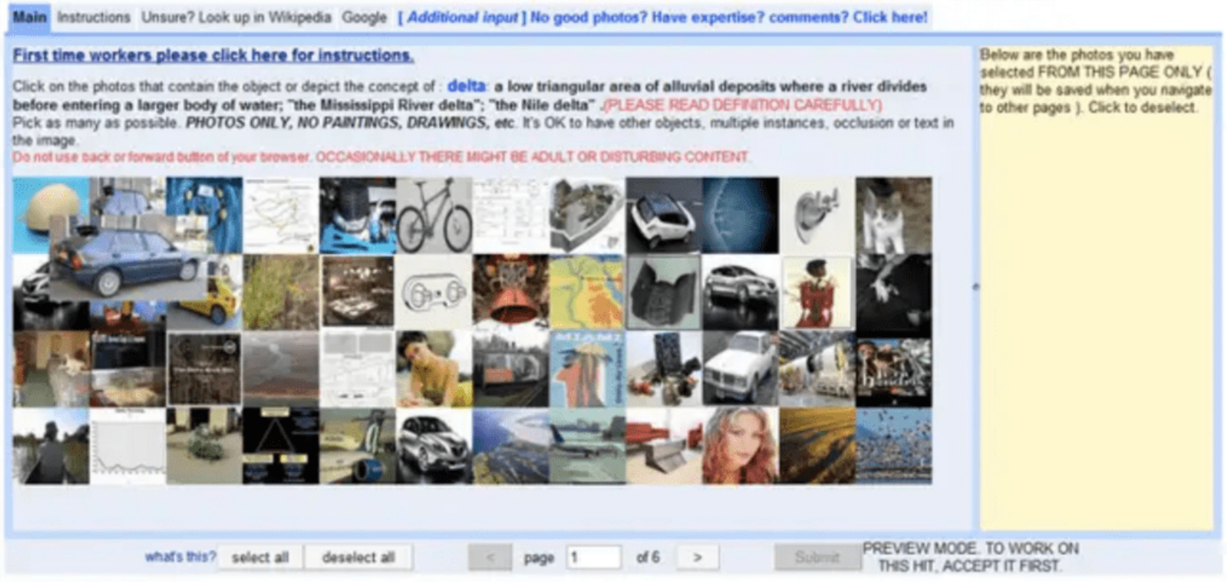
Supervised Learning

Training data

$$\{x^{(1)}, y^{(1)}\}$$
$$\{x^{(2)}, y^{(2)}\} \quad \longrightarrow \quad \boxed{\text{Learner}} \quad \longrightarrow \quad f : X \to Y$$
$$\{x^{(3)}, y^{(3)}\}$$
$$\ldots$$

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 surface
a graduate student who a
Amazon Mechanical Turk
sitting at computers arou
online tasks for pennies.



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

"He showed me the webs
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."

Unsupervised Learning

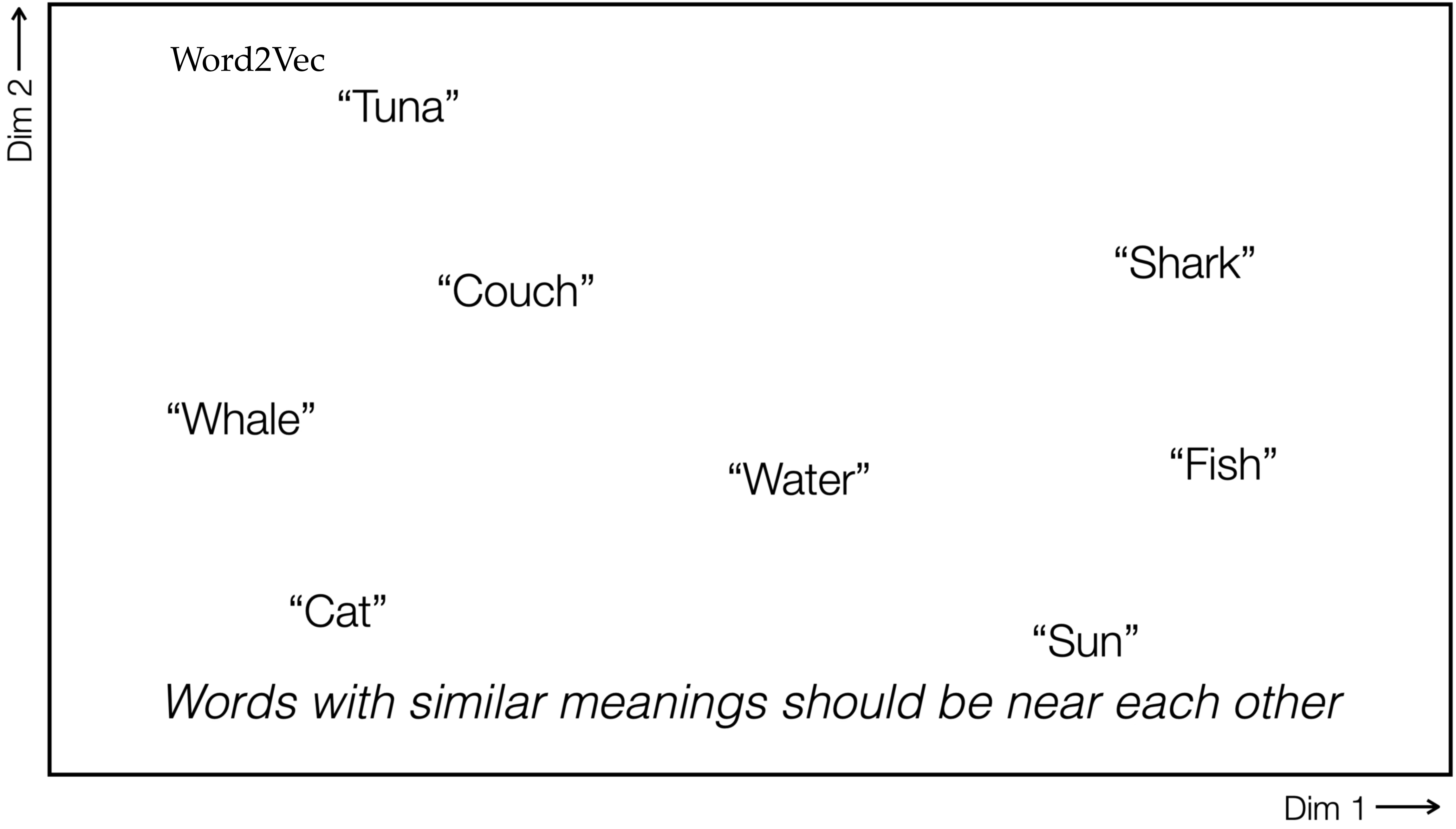
Training Data

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

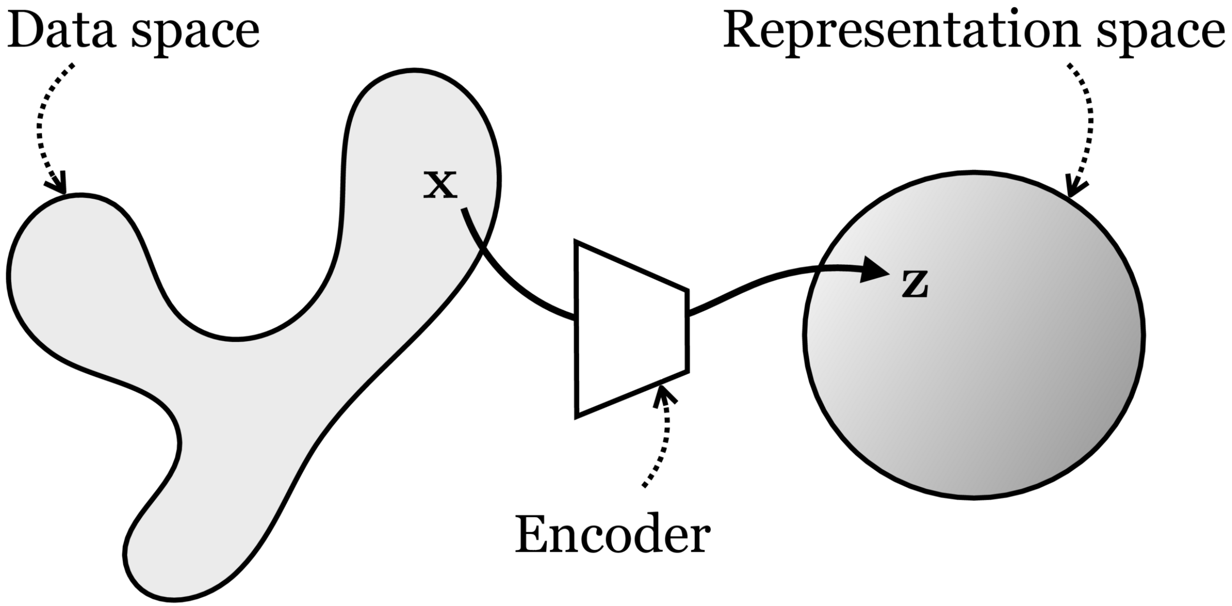
$\{x^{(2)}\}$ $\rightarrow$ Learner $\rightarrow$ "Good" Representation

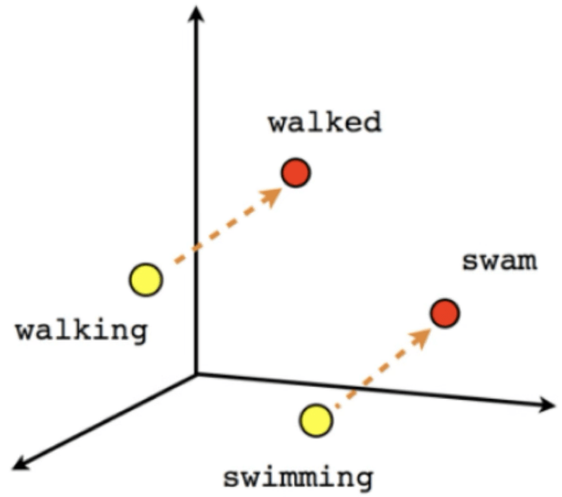
$\{x^{(3)}\}$

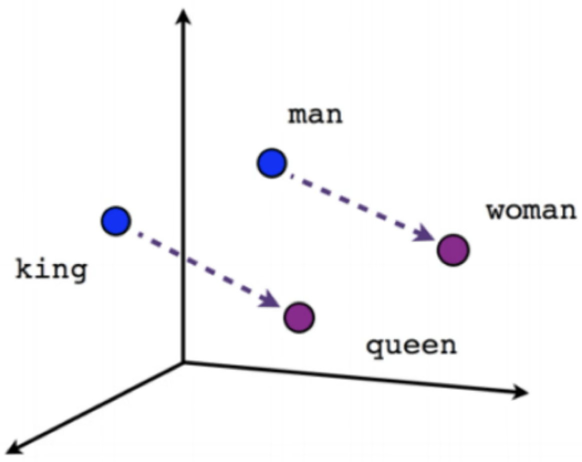
$\ldots$


Word2Vec
“Tuna”
“Shark”
“Couch”
“Whale”
“Fish”
“Water”
“Cat”
“Sun”
Words with similar meanings should be near each other
Dim 2
Dim 1

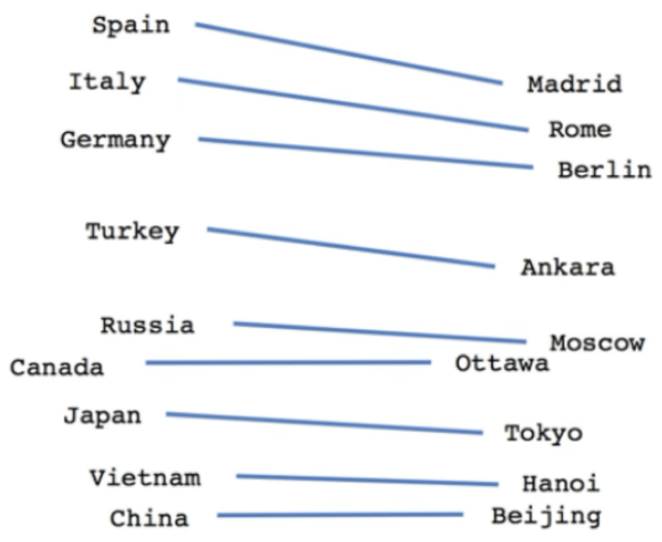
Word2Vec







https://www.tensorflow.org/text/tutorials/word2vec



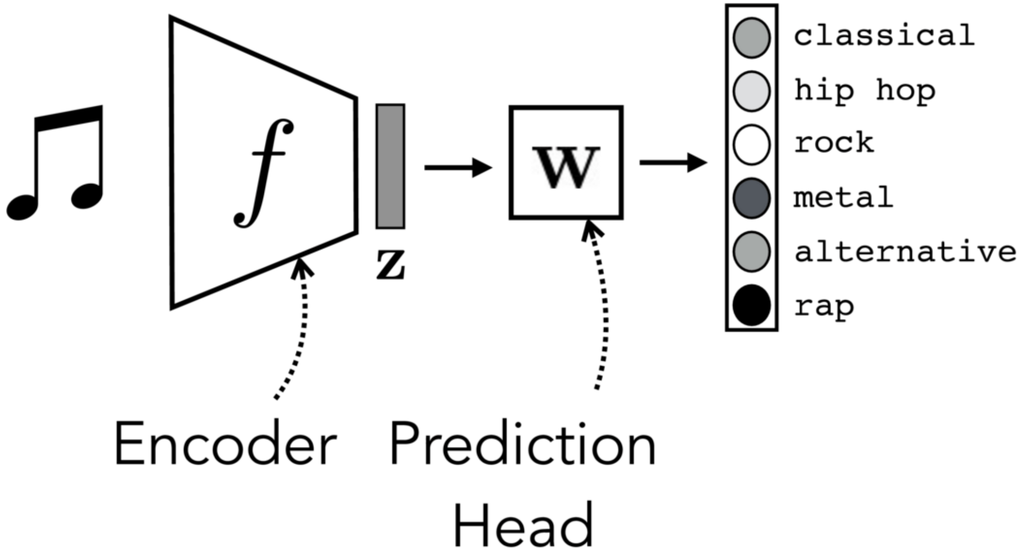
Country-Capital

X = Vector(“Paris”) – vector(“France”) + vector(“Italy”) ≈ vector("Rome")

“Meaning is use” — Wittgenstein

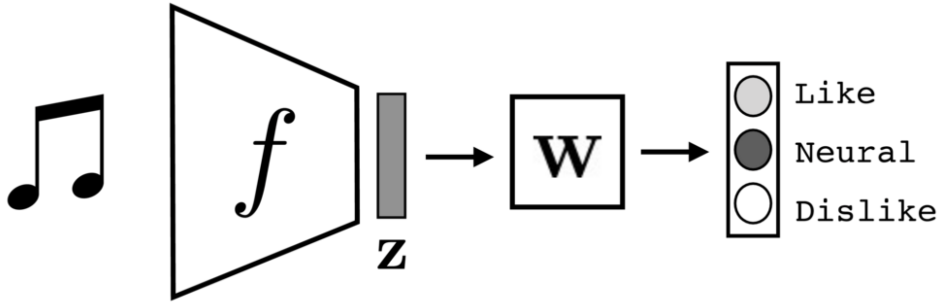
Training

Genre recognition



Testing

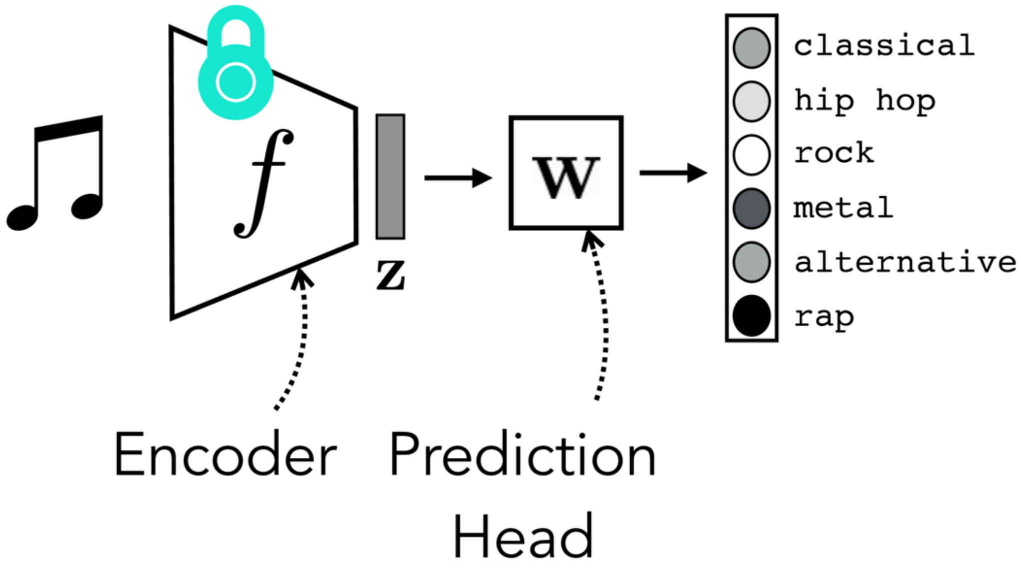
Preference prediction



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

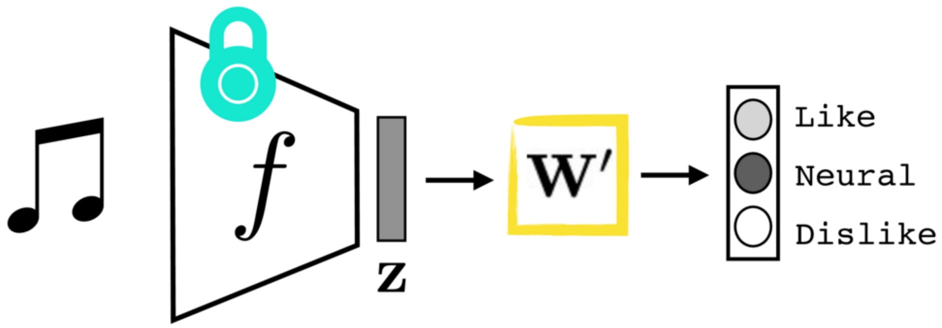
Training

Genre recognition



Adapting

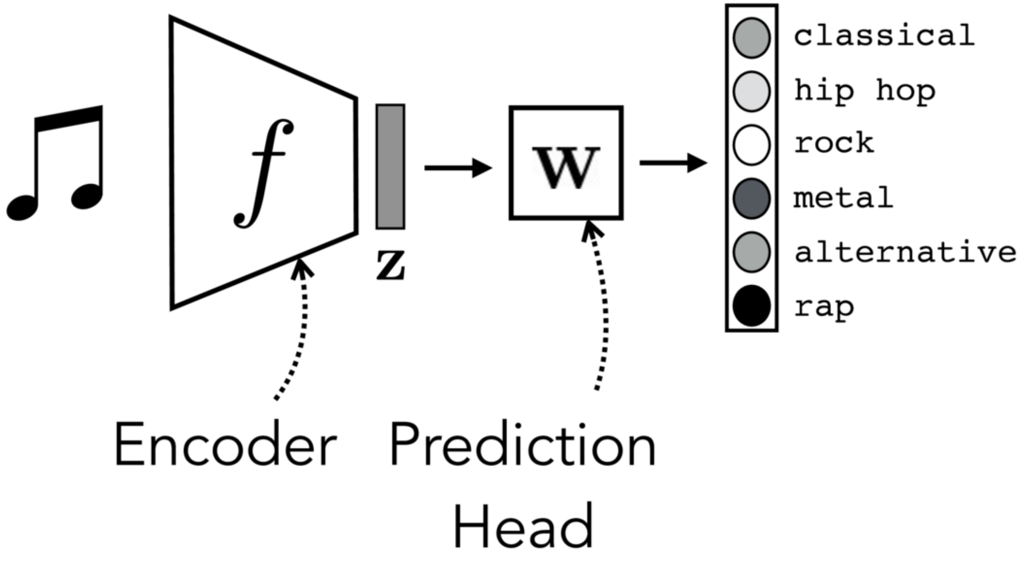
Preference prediction



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

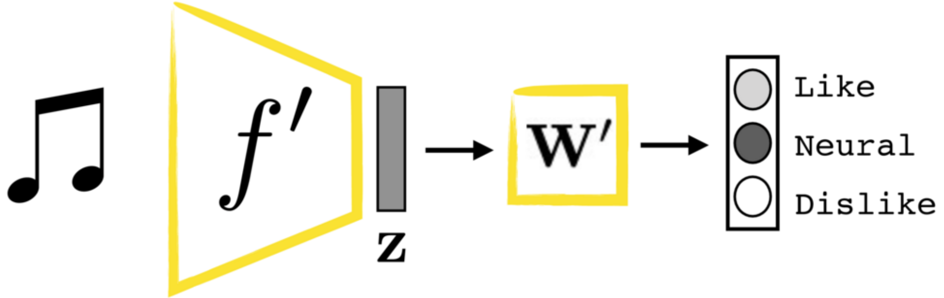
Training

Genre recognition



Adapting

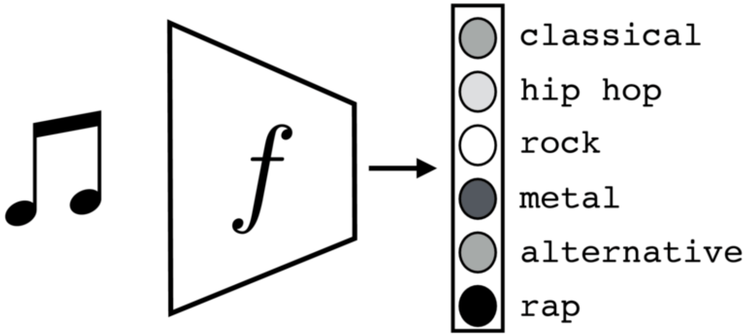
Preference prediction



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

Pretraining

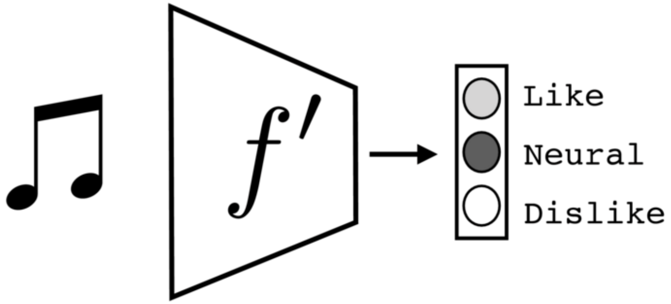
Genre recognition



A lot of data

Adapting

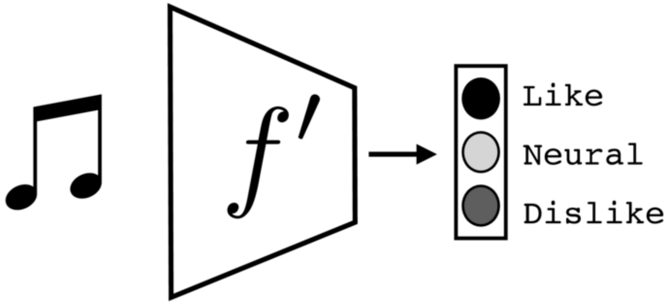
Preference prediction



A little data

Testing

Preference prediction

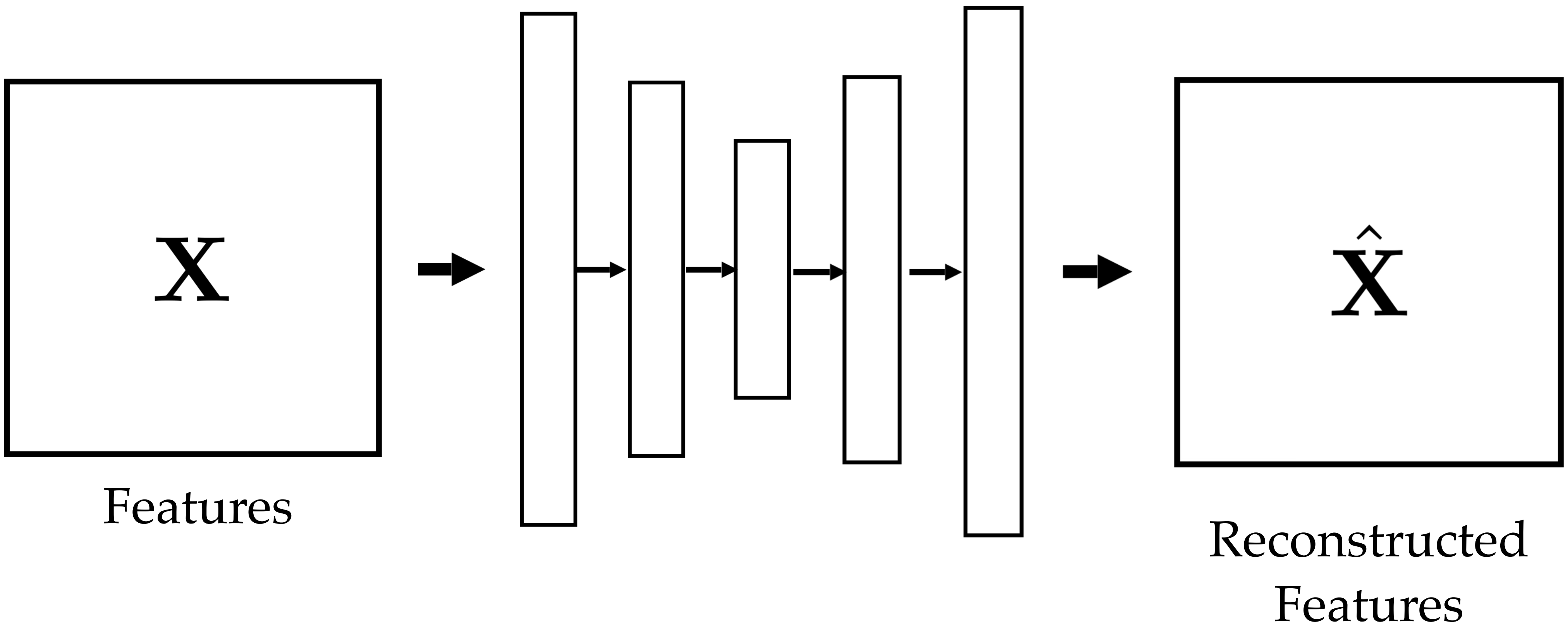


Outline

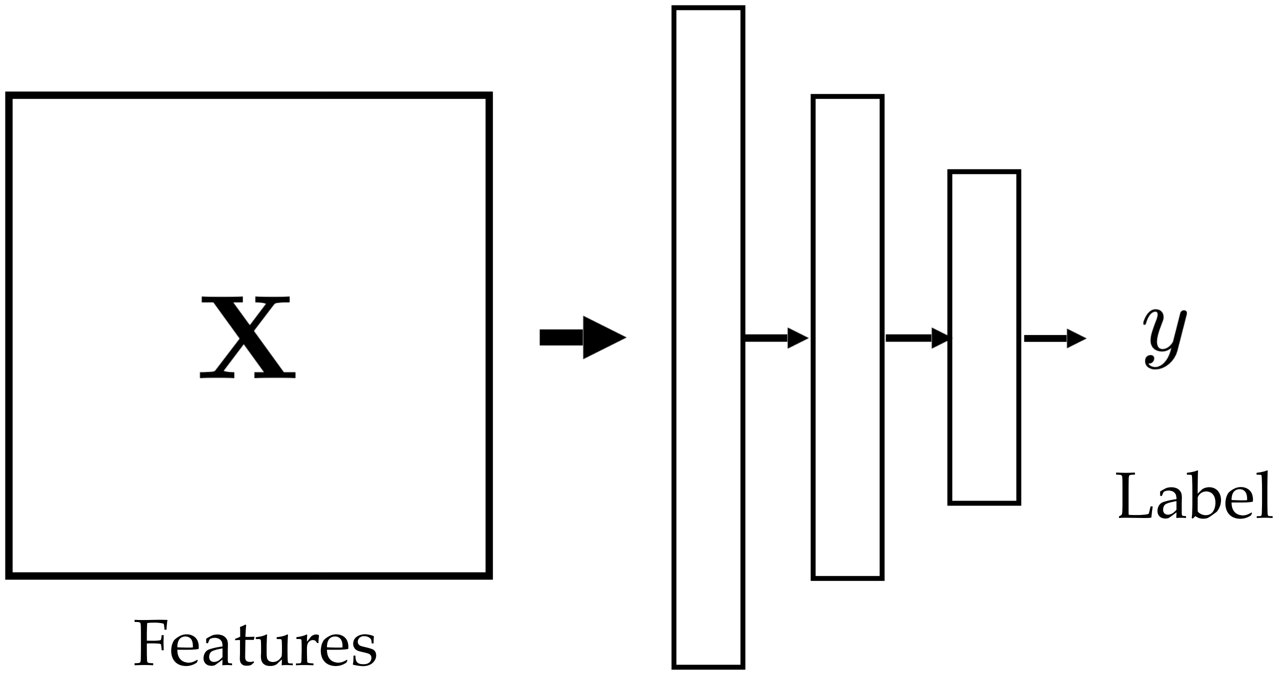
- Recap, neural networks mechanism
- Neural networks are *representation* learners
- Auto-encoder:
  - Bottleneck
  - Reconstruction
- Unsupervised learning
- (Some recent representation learning ideas)

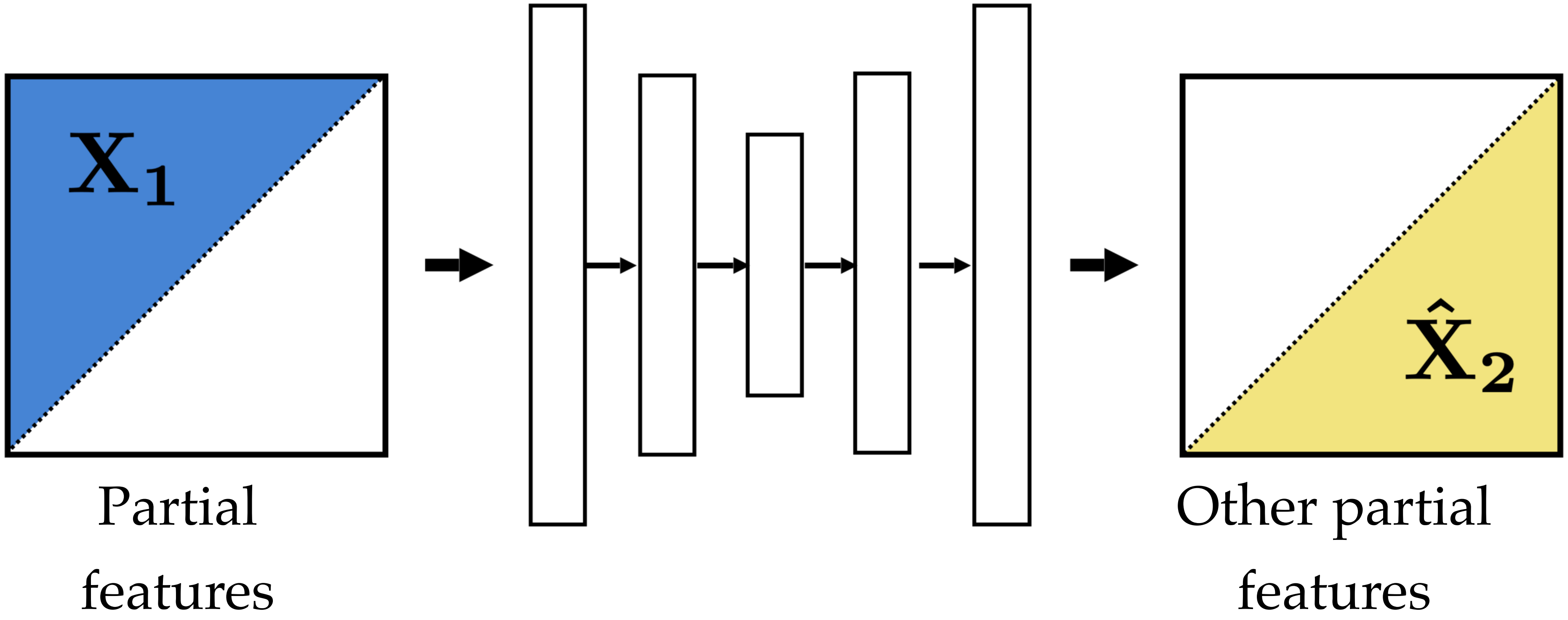
(

Feature reconstruction (unsupervised learning)

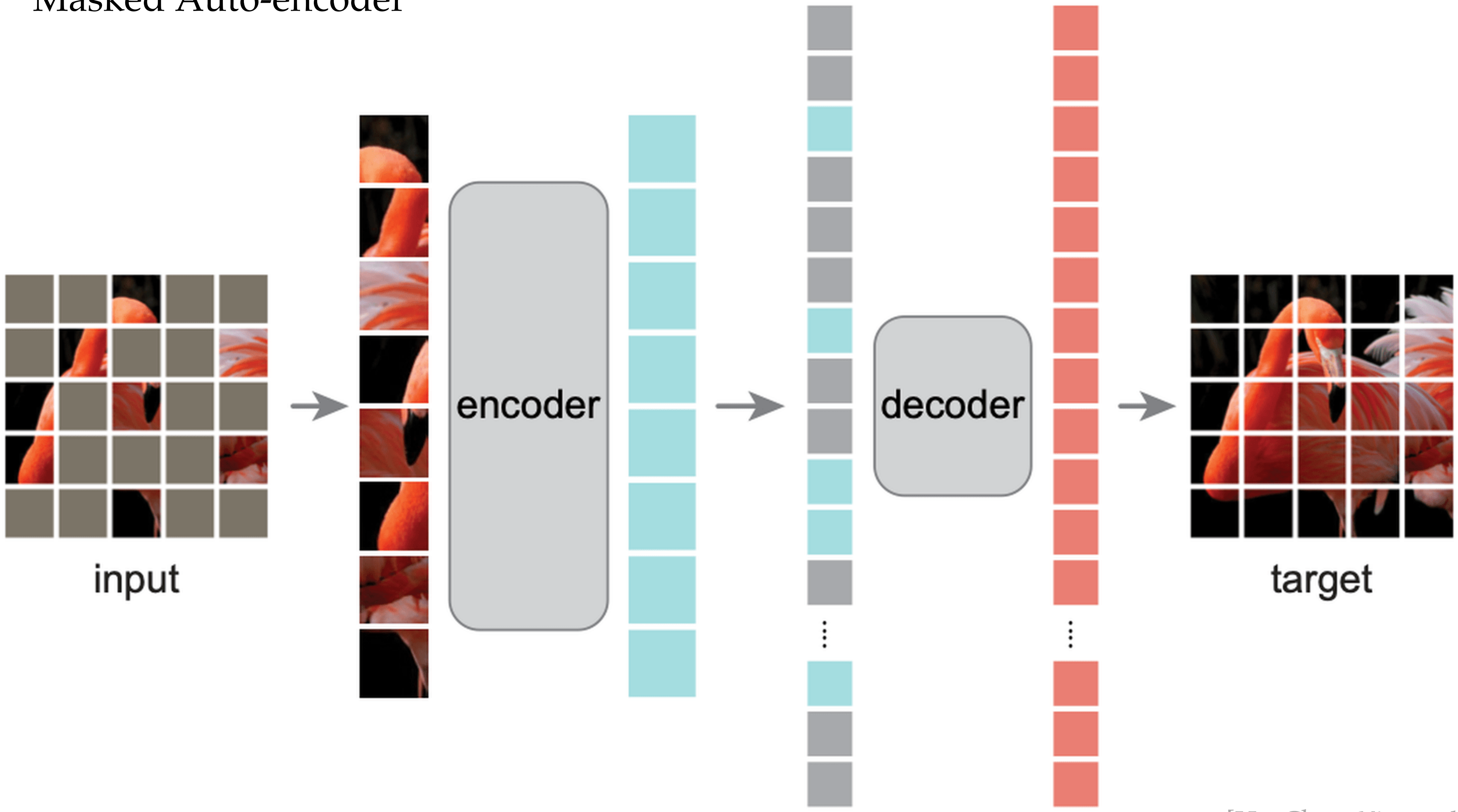


Label prediction (supervised learning)



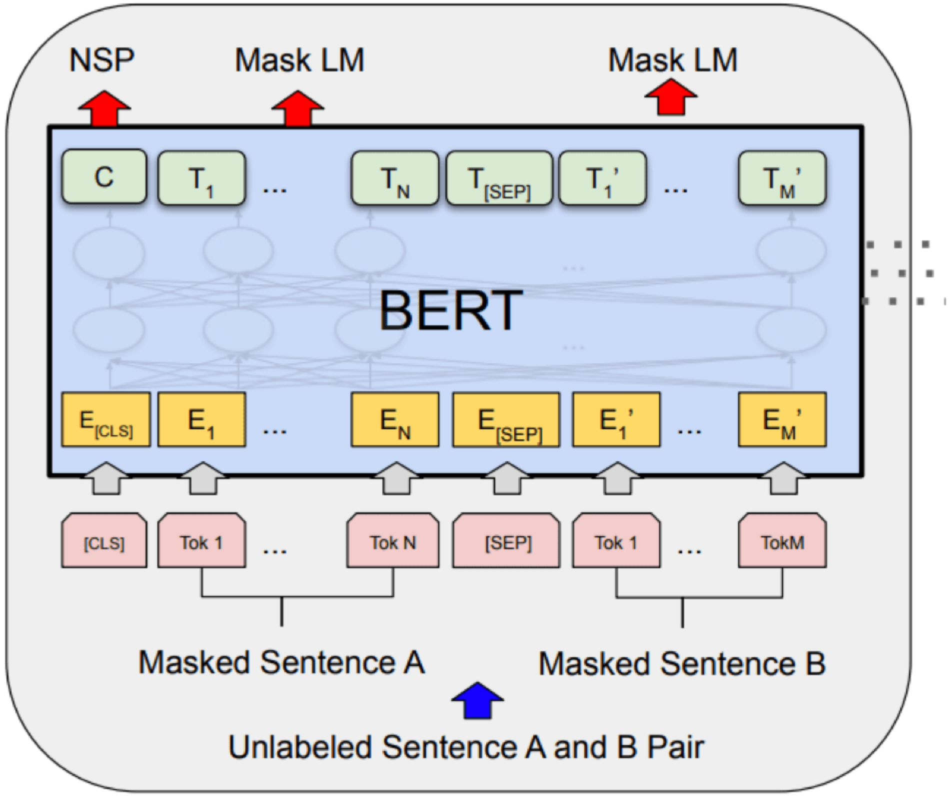

$\mathbf{X}_1$
$\hat{\mathbf{X}}_2$
Partial features
Other partial features

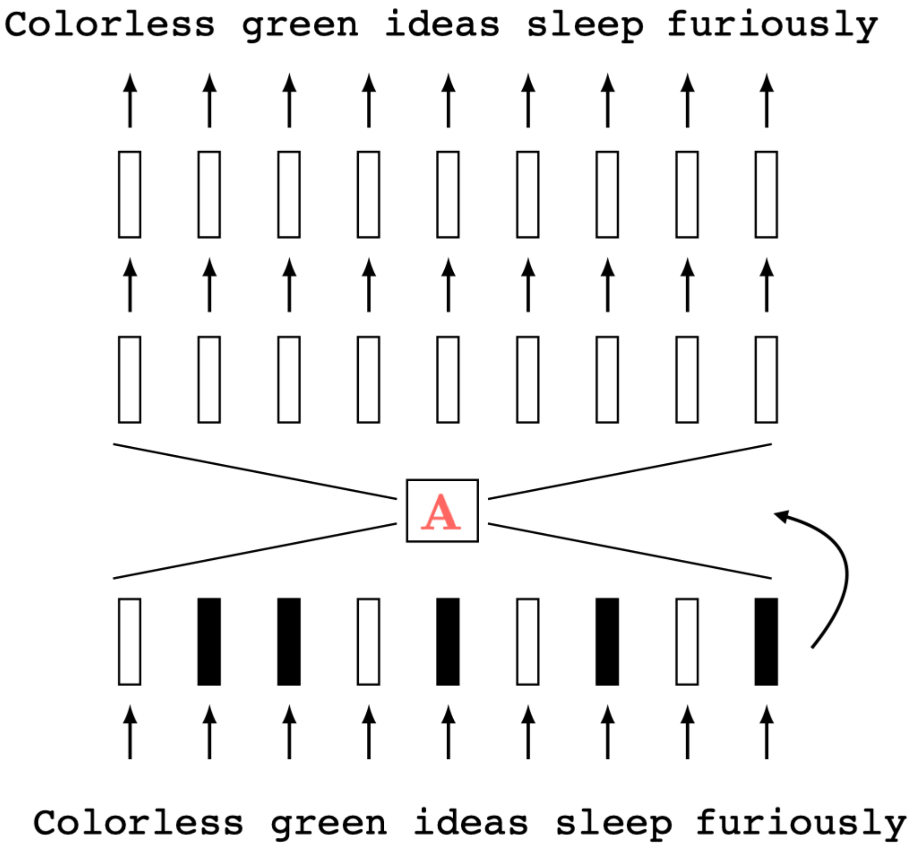
Masked Auto-encoder



[He, Chen, Xie, et al. 2021]

Masked Auto-encoder

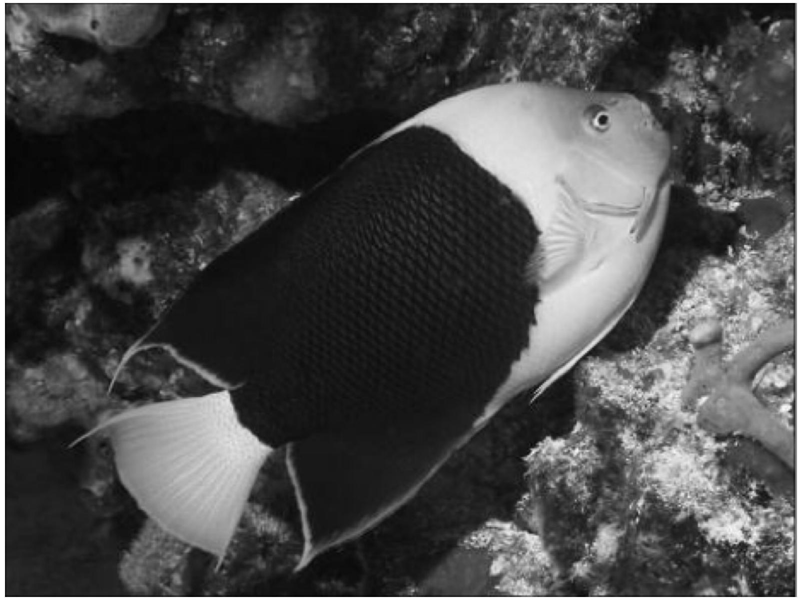


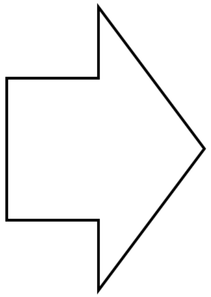


[Devlin, Chang, Lee, et al. 2019]

[Zhang, Isola, Efros, ECCV 2016]

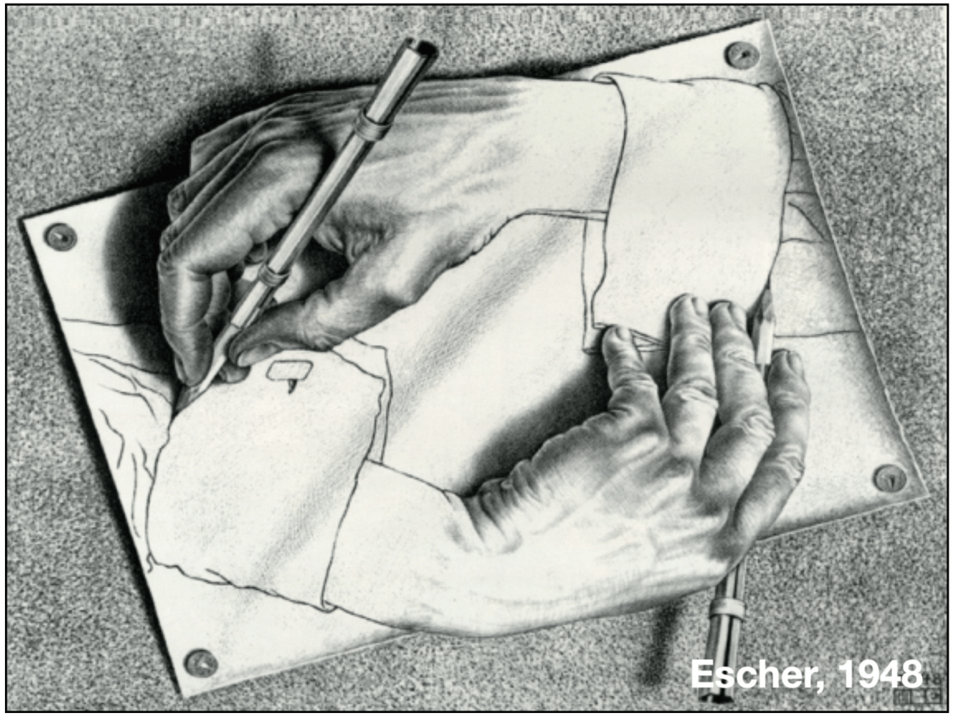
predict color from gray-scale





[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

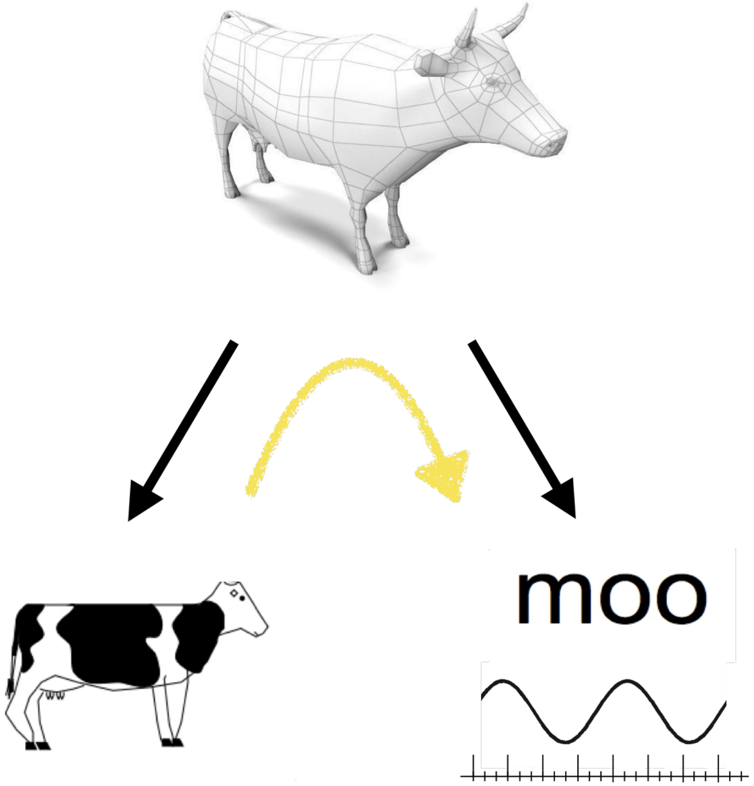
Y. LeCun

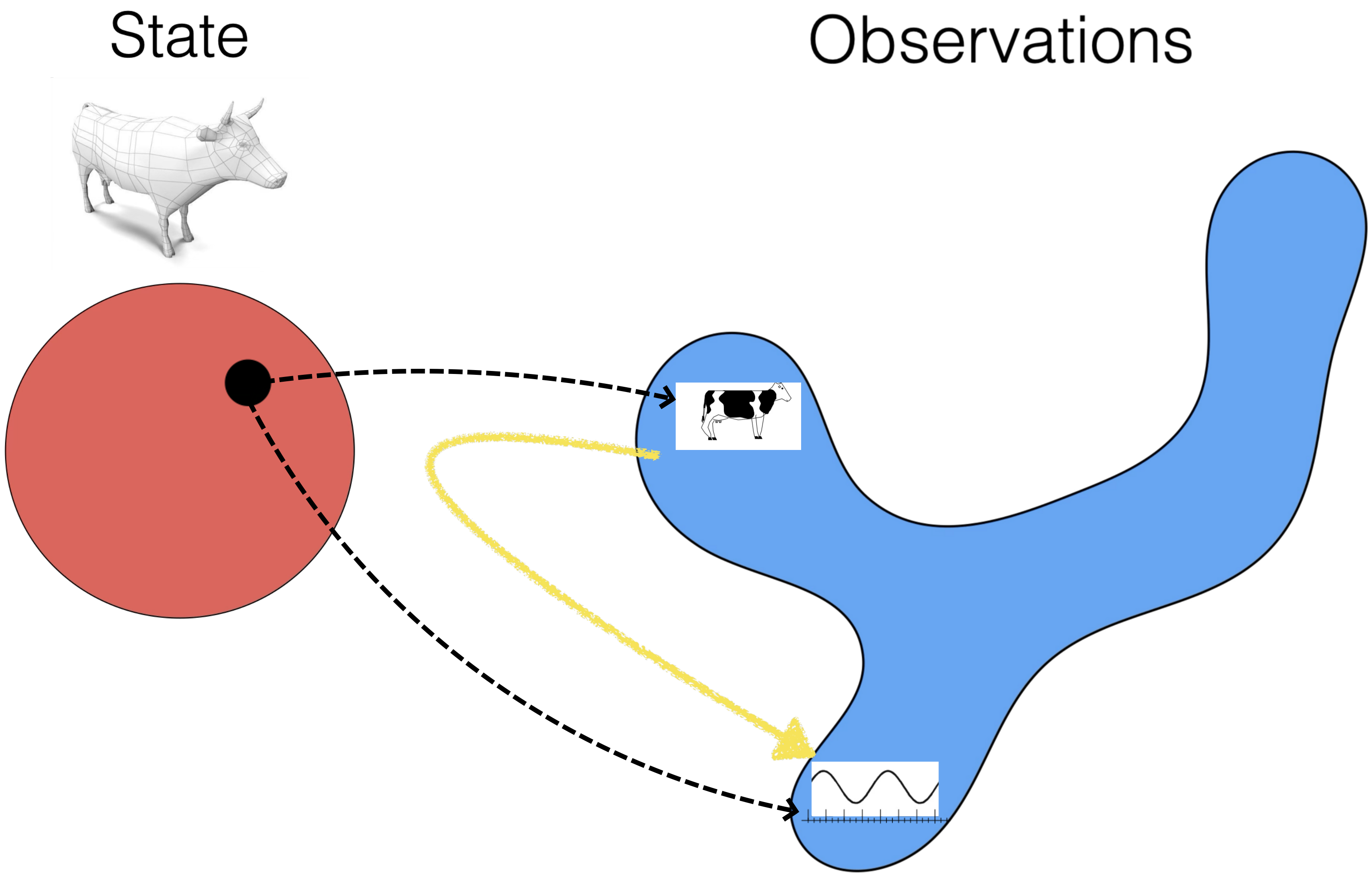
How Much Information is the Machine Given during Learning?

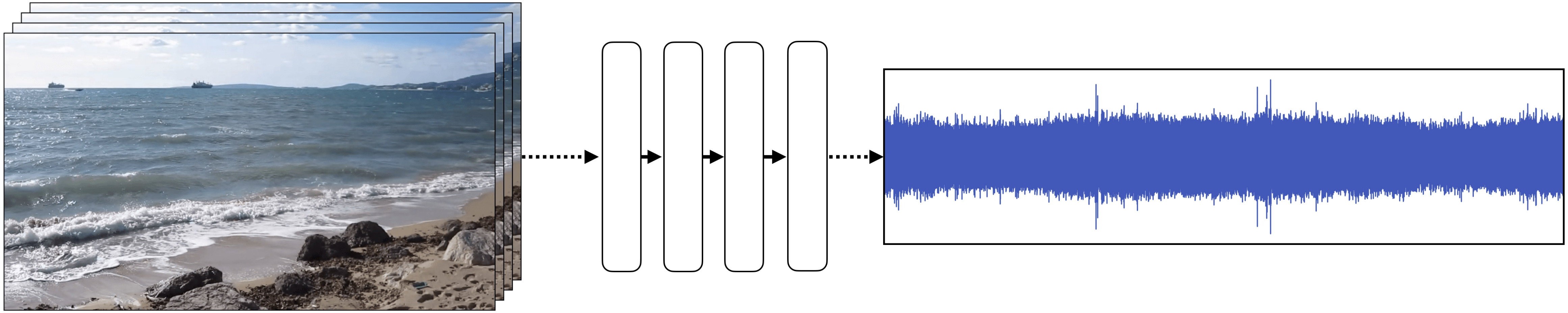
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits 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**

[Slide Credit: Yann LeCun]

The allegory of the cave

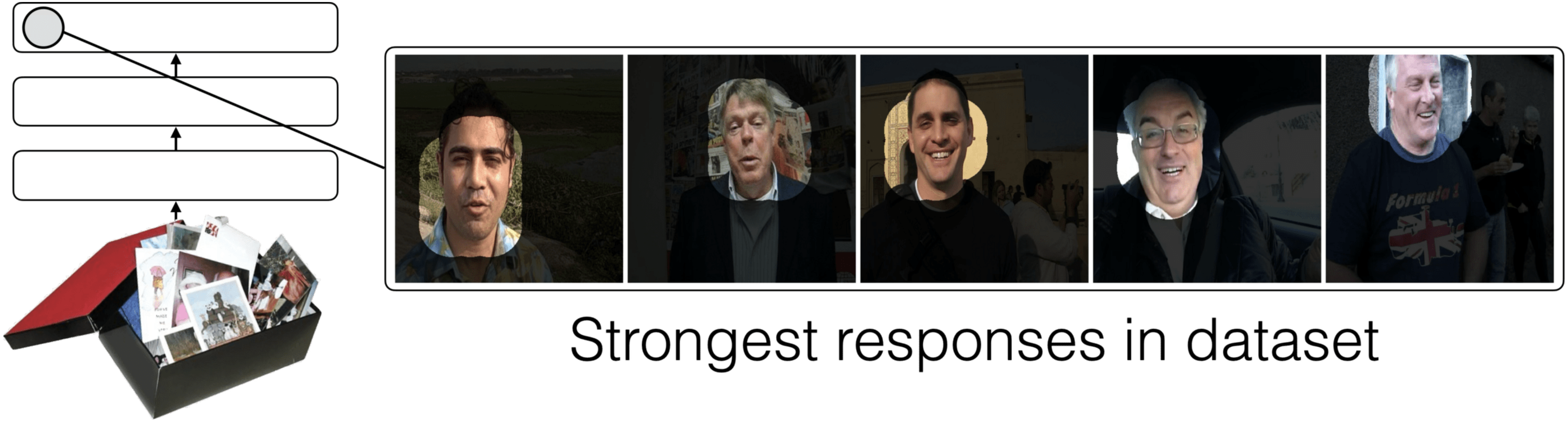



State
Observations

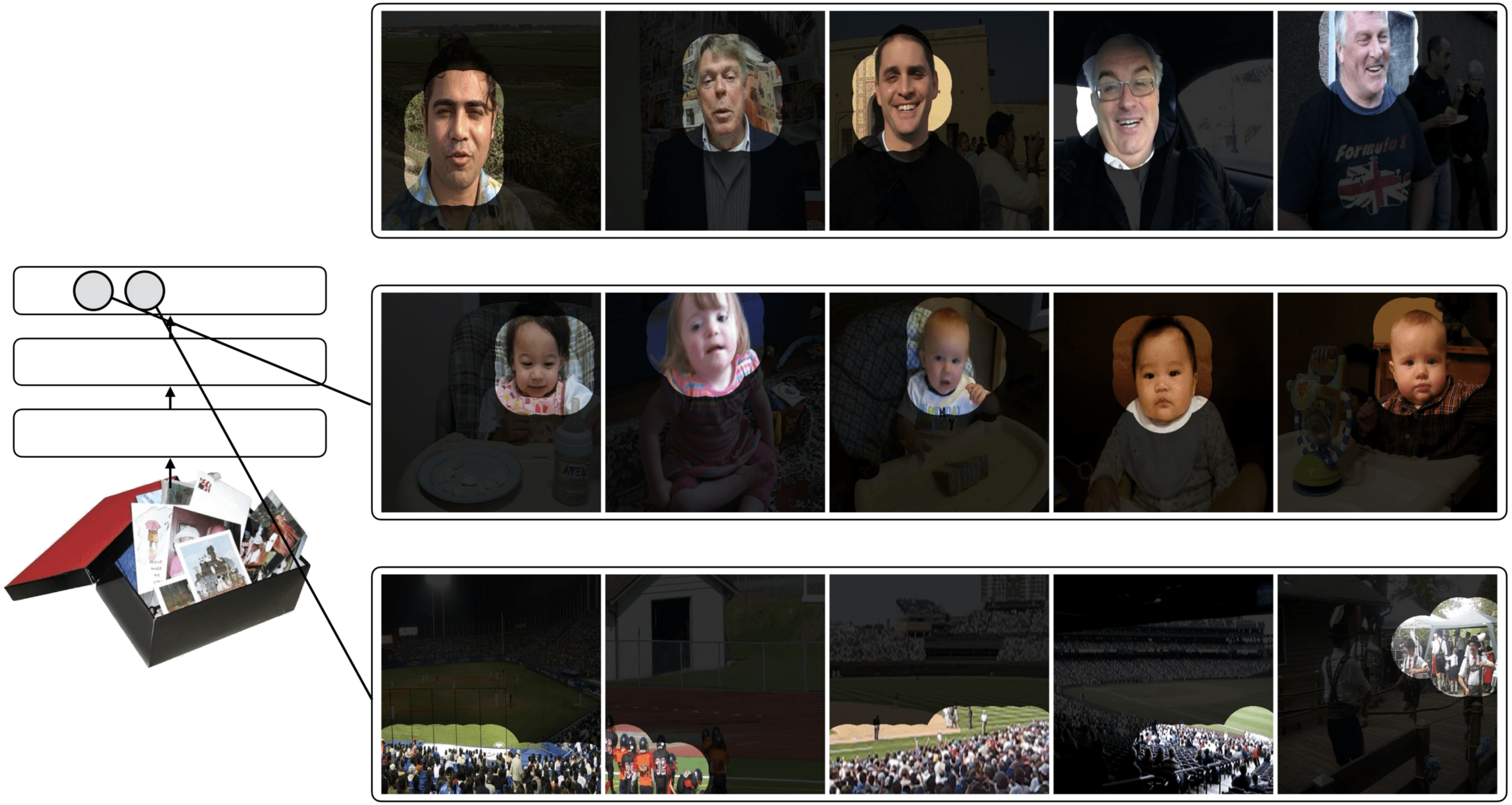


[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]
[Slide credit: Andrew Owens]

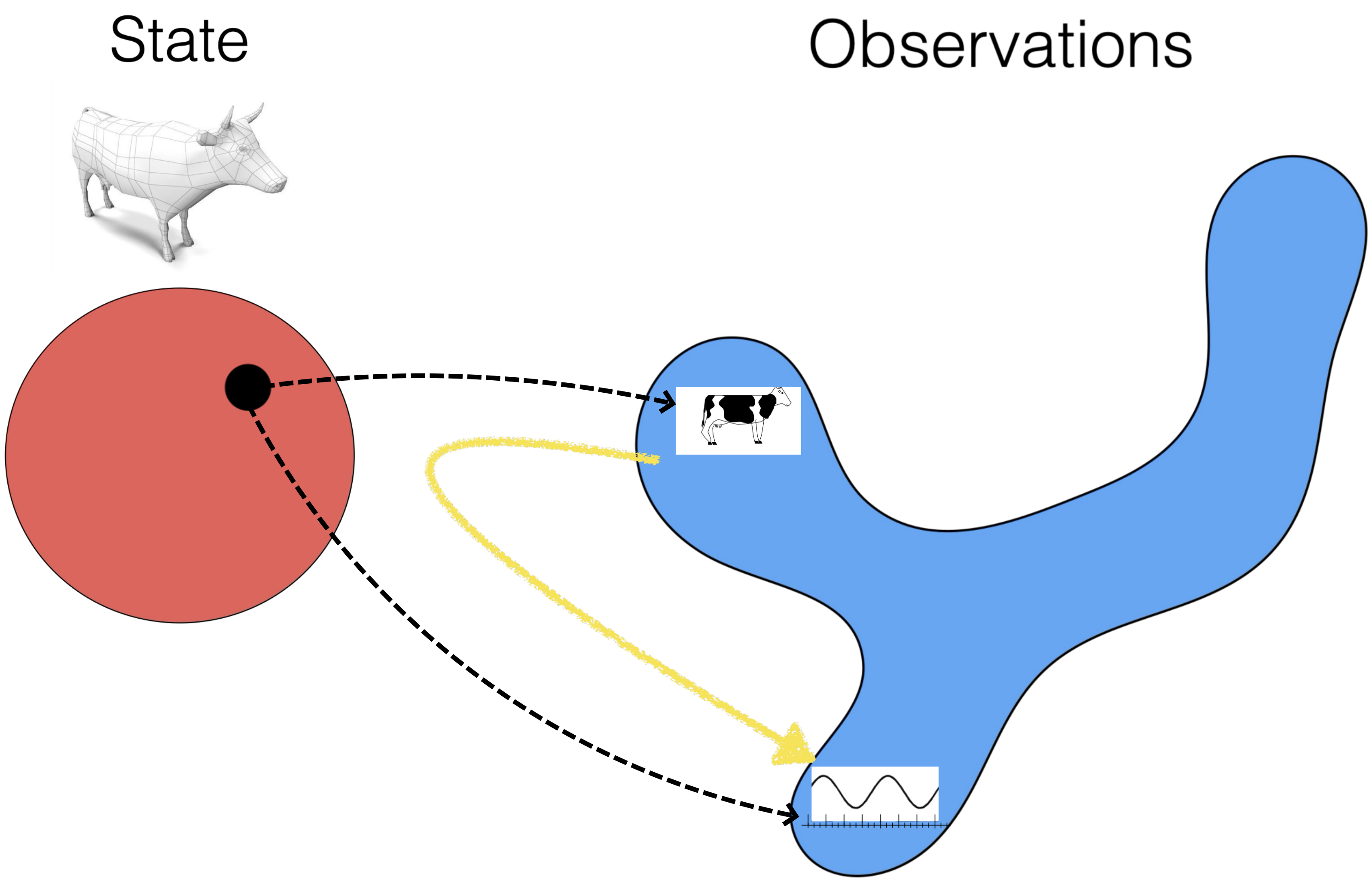
What did the model learn?



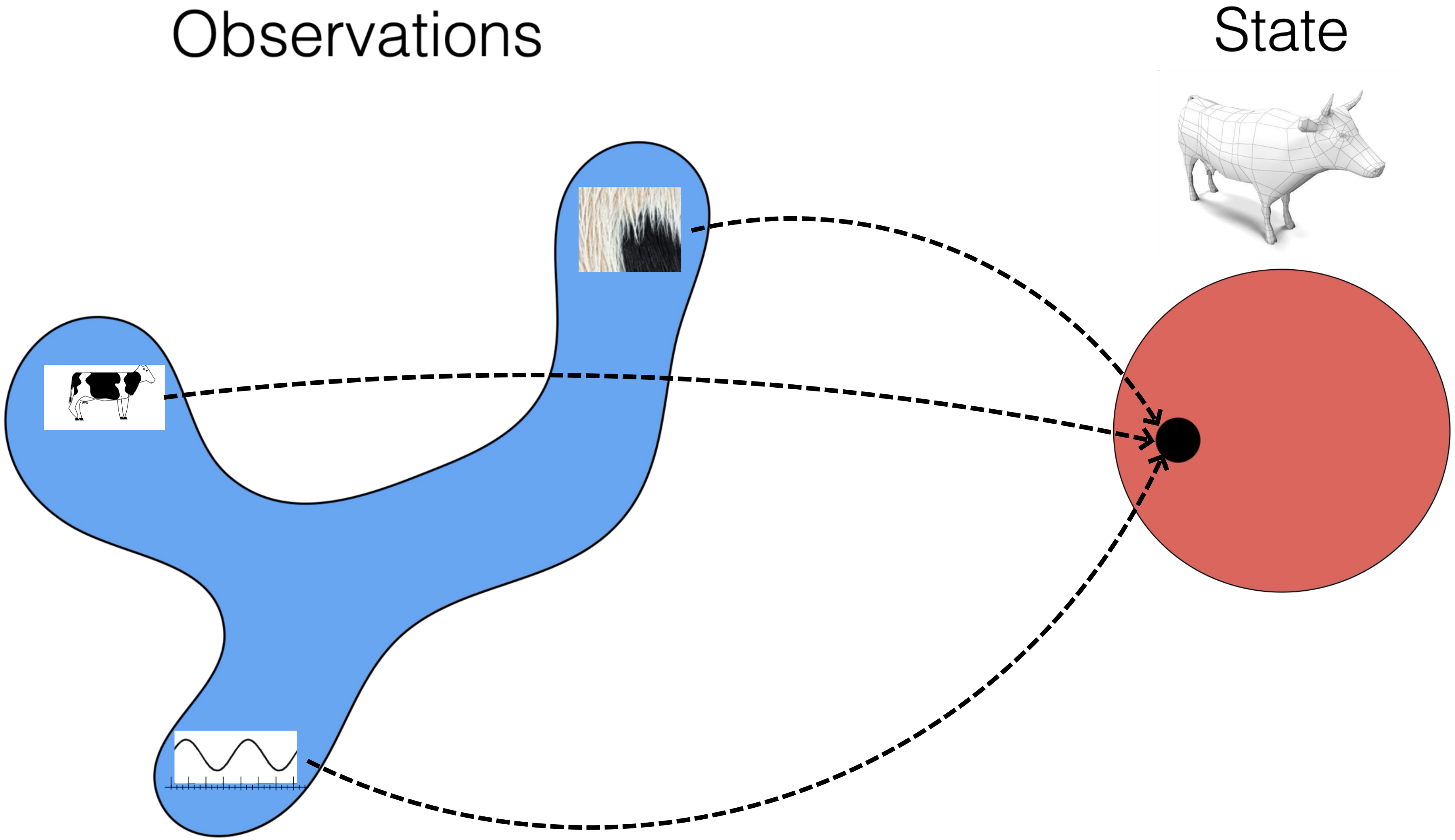
[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]
[Slide credit: Andrew Owens]



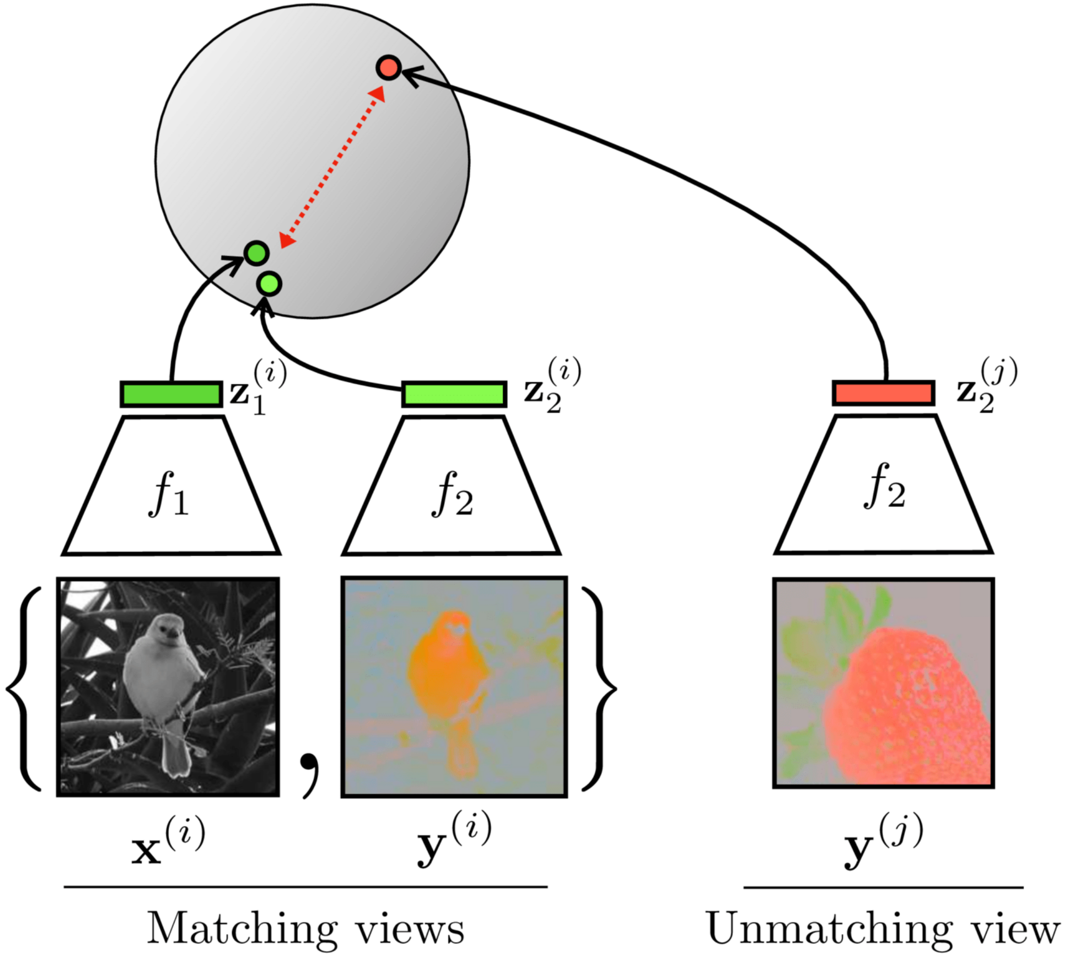
[Owens et al, Ambient Sound Provides Supervision for Visual Learning, ECCV 2016]
[Slide credit: Andrew Owens]

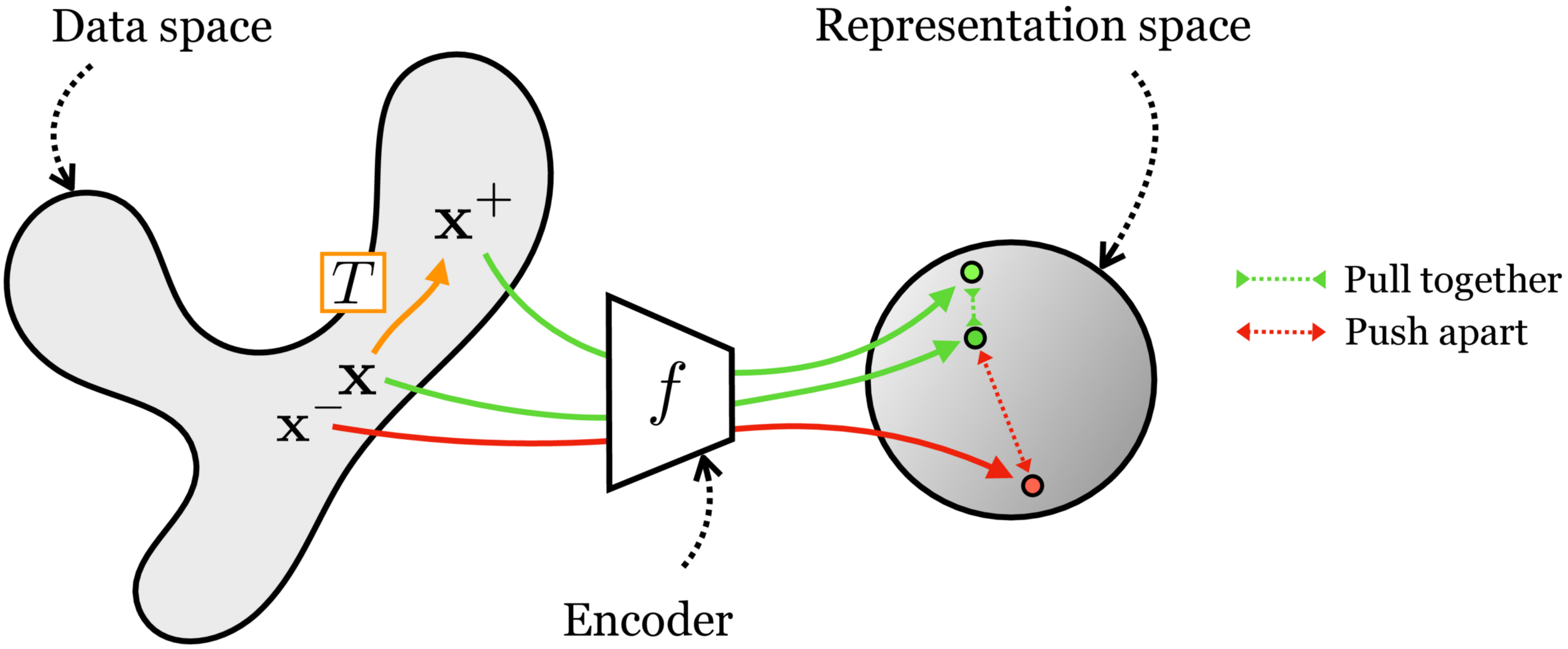

State
Observations

Contrastive learning



Contrastive learning




Data space
Representation space
$\mathbf{x}^+$
T
$\mathbf{x}$
$\mathbf{x}^-$
f
Encoder
Pull together
Push apart

[Chen, Kornblith, Norouzi, Hinton, ICML 2020]

DallE

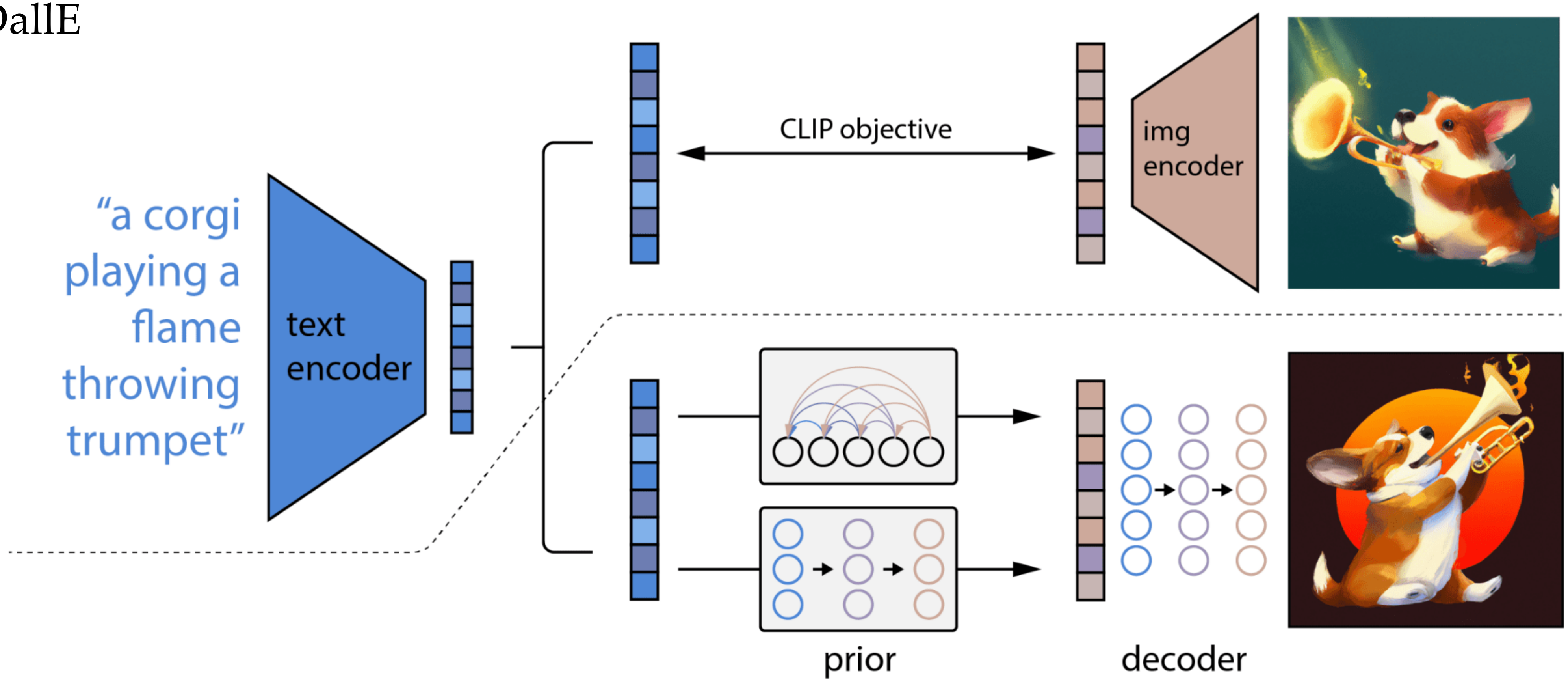


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

[https://arxiv.org/pdf/2204.06125.pdf]

$$\Bigg)$$

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)

https://forms.gle/36SX9pqCTWpp323N8

We'd love to hear your thoughts.

Thanks!