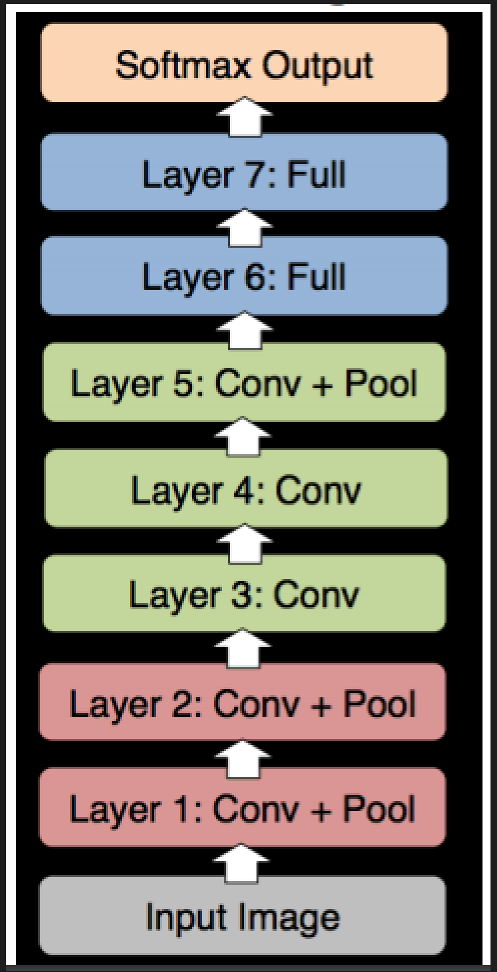
INTRODUCTION TO MACHINE LEARNING COMPSCI 4ML3

Lecture 24

Hassan Ashtiani

BREAKTHROUGH ON IMAGENET

- ImageNet classification challenge
  - Millions of images
  - Thousands of classes
- In 2012, AlexNet used won the competition by a high margin
  - ~15% error compared to ~25% of the next team
  - They used a convolutional architecture
  - They used GPUs for speedup
- CNNs became very popular



VGG NETWORK

- 2015

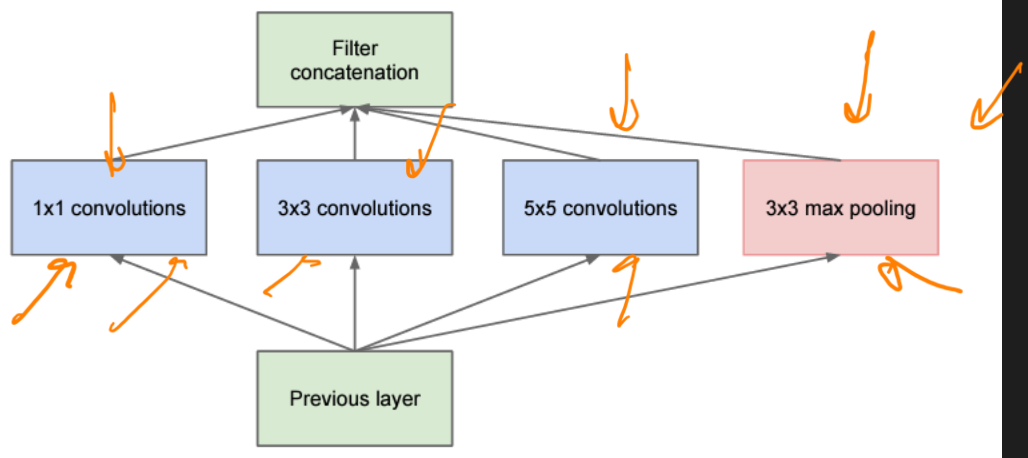
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

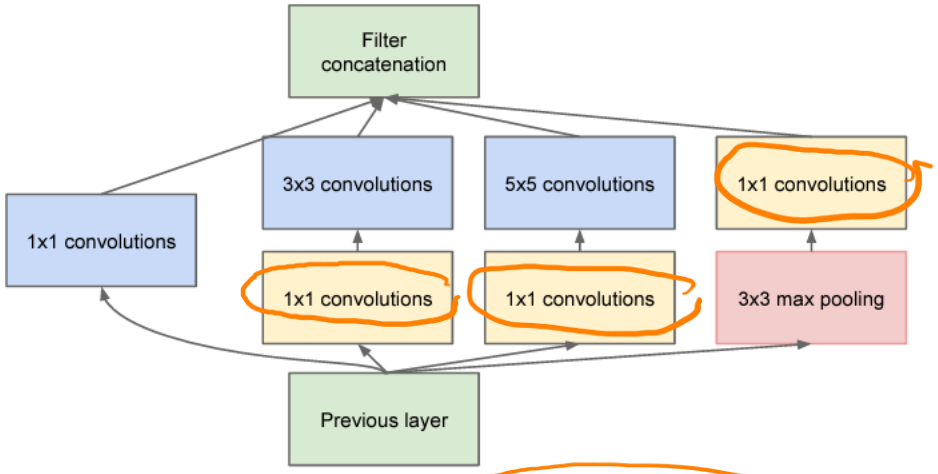
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

GOOGLENET

- INCEPTION MODULE
  - 2015



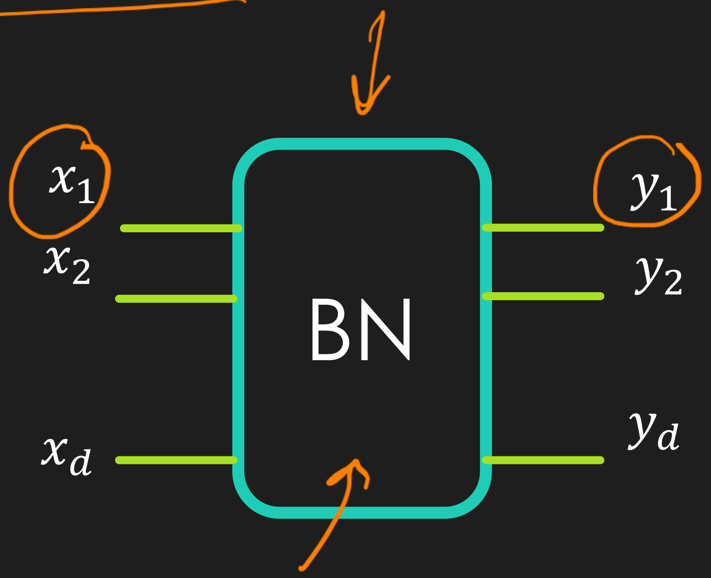
(a) Inception module, naïve version

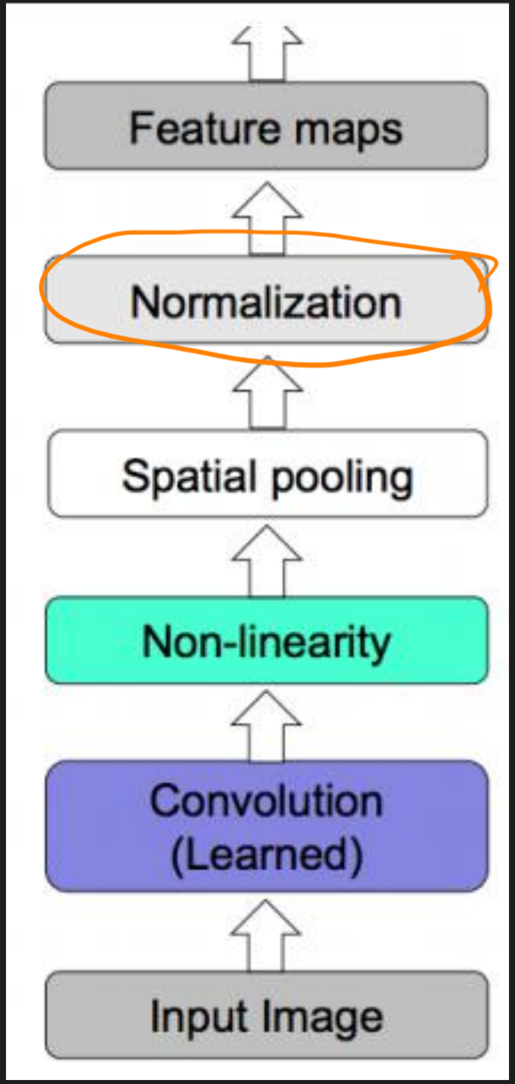


(b) Inception module with dimensionality reduction

BATCH NORMALIZATION

- INCEPTION WITH BATCH NORMALIZATION
  - 2015
- DATA POINT j
  - $x^j = [x_1^j, x_2^j, \ldots, x_d^j]$
- BATCH $X = [(x^1)^T \quad \ldots \quad (x^b)^T]^T$
  - BATCH SIZE = b
- $Y = ?$





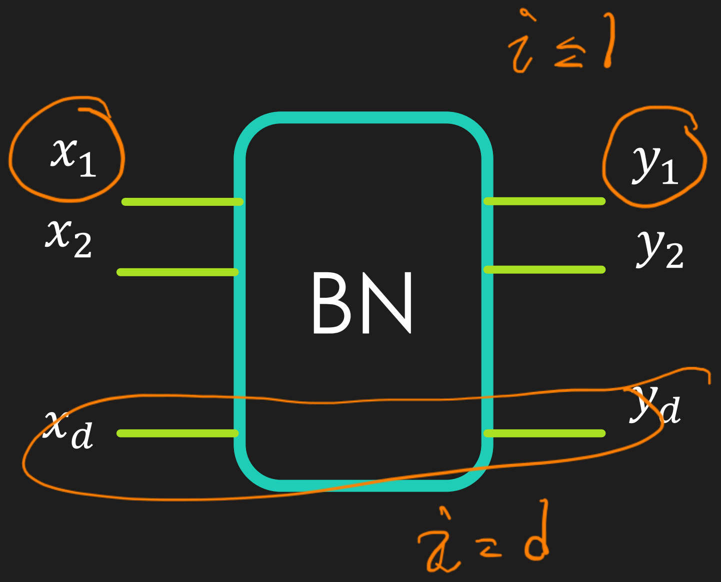
BATCH NORMALIZATION

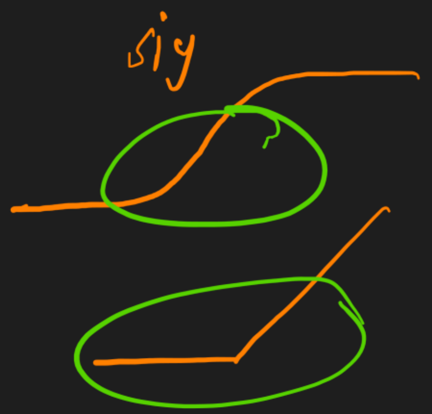
feature i :

$$y_i^j = \frac{x_i^j - \mu_i}{\sigma_i + \varepsilon}$$

$$\mu_i = \frac{1}{b} \sum_{j=1}^{b} x_i^j$$

$$\sigma_i^2 = \frac{1}{b} \sum_{j=1}^{b} (x_i^j - \mu_i)^2$$



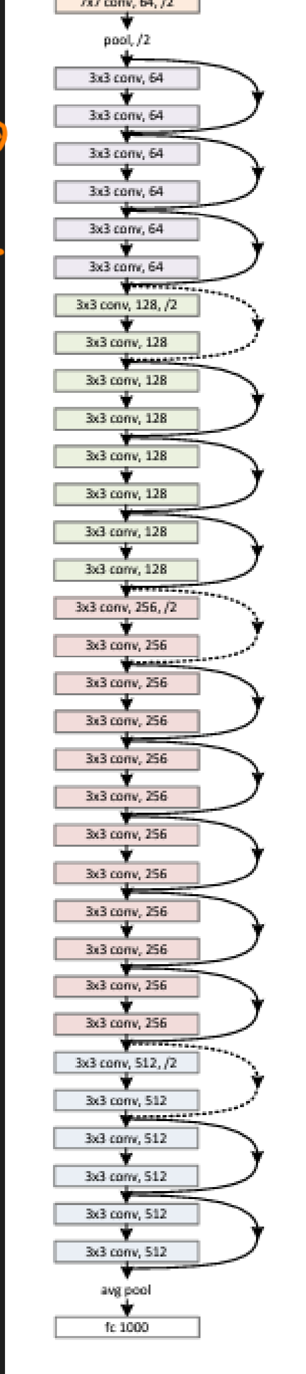


RESNET

- 34 RESIDUAL LAYERS
- 2016

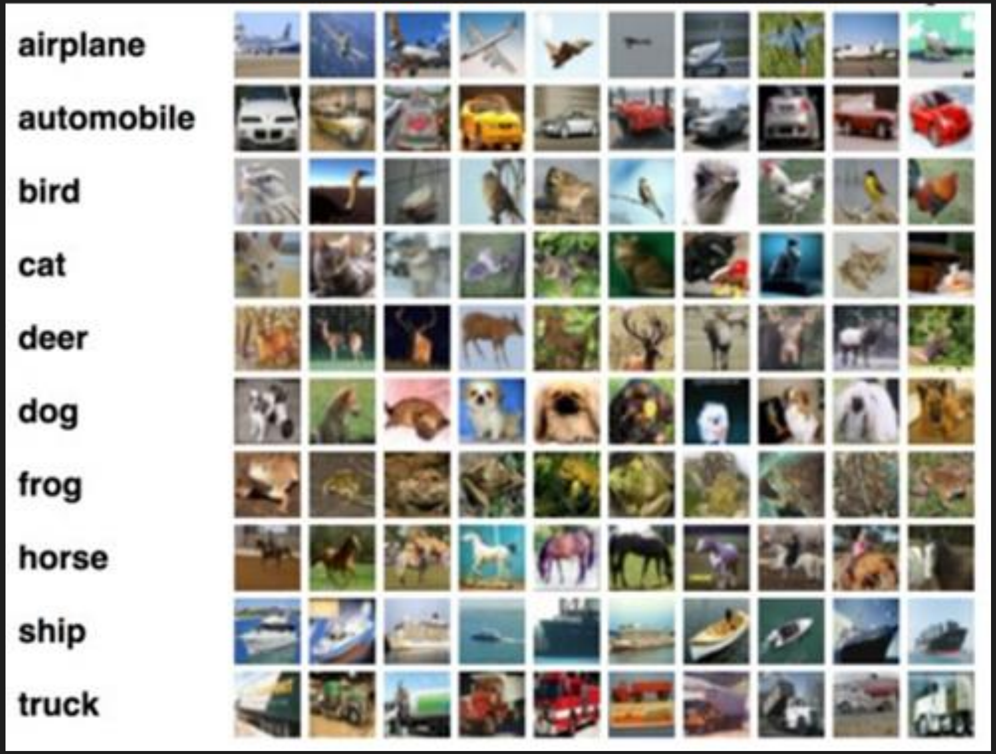
method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).



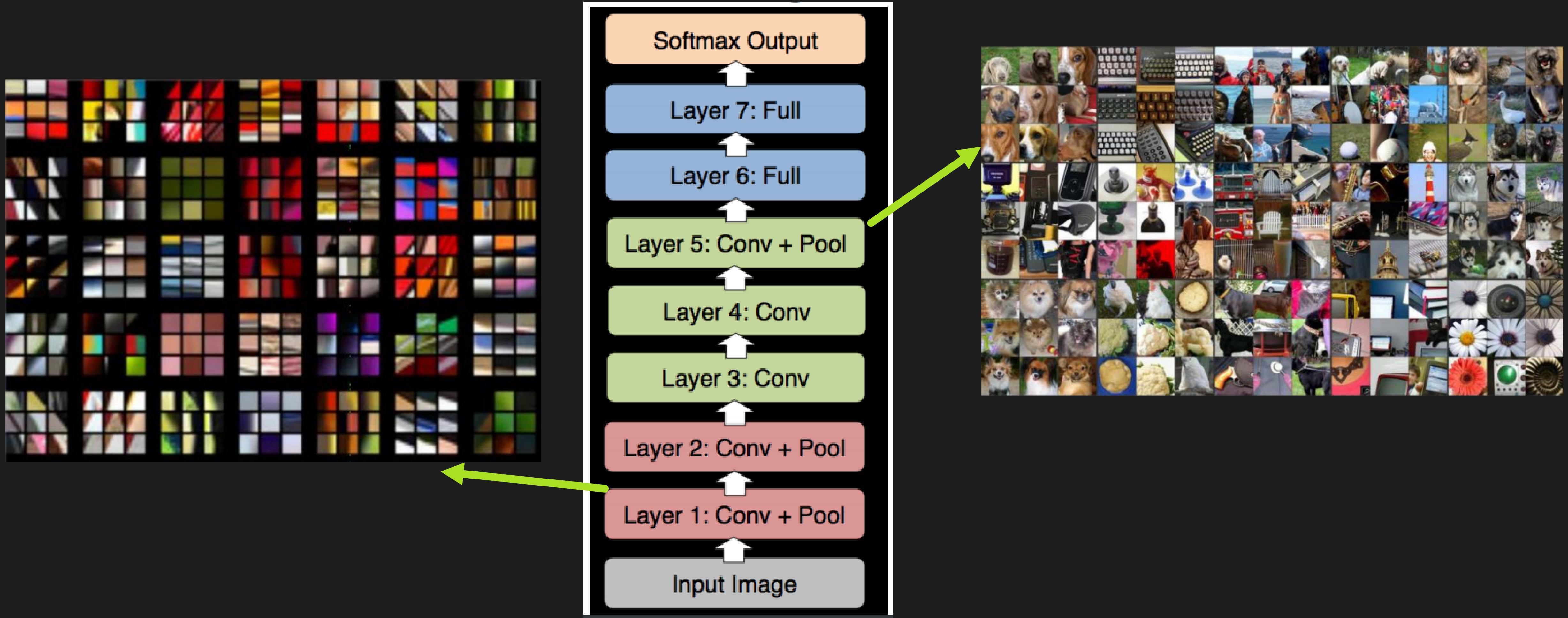
TRANSFER LEARNING

- USE IMAGENET DATA SET TO IMPROVE FOR CIFAR?



REUSING FEATURE EXTRACTORS

- Earlier layers extract more generic features



FREEZING VS FINE TUNING

- Assume input images are of the same size

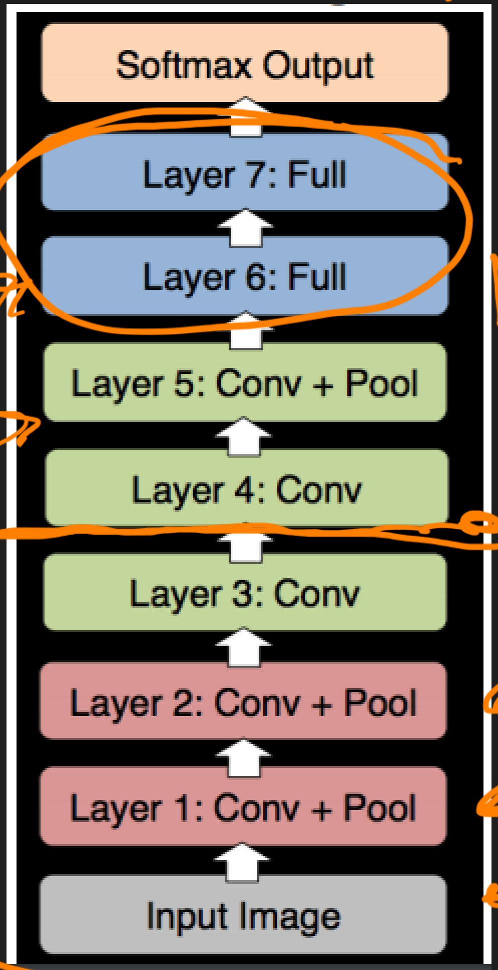
For CIFAR

- Reuse the first few layers from ImageNet
  - Initialize the last few layers randomly

CIFAR Training

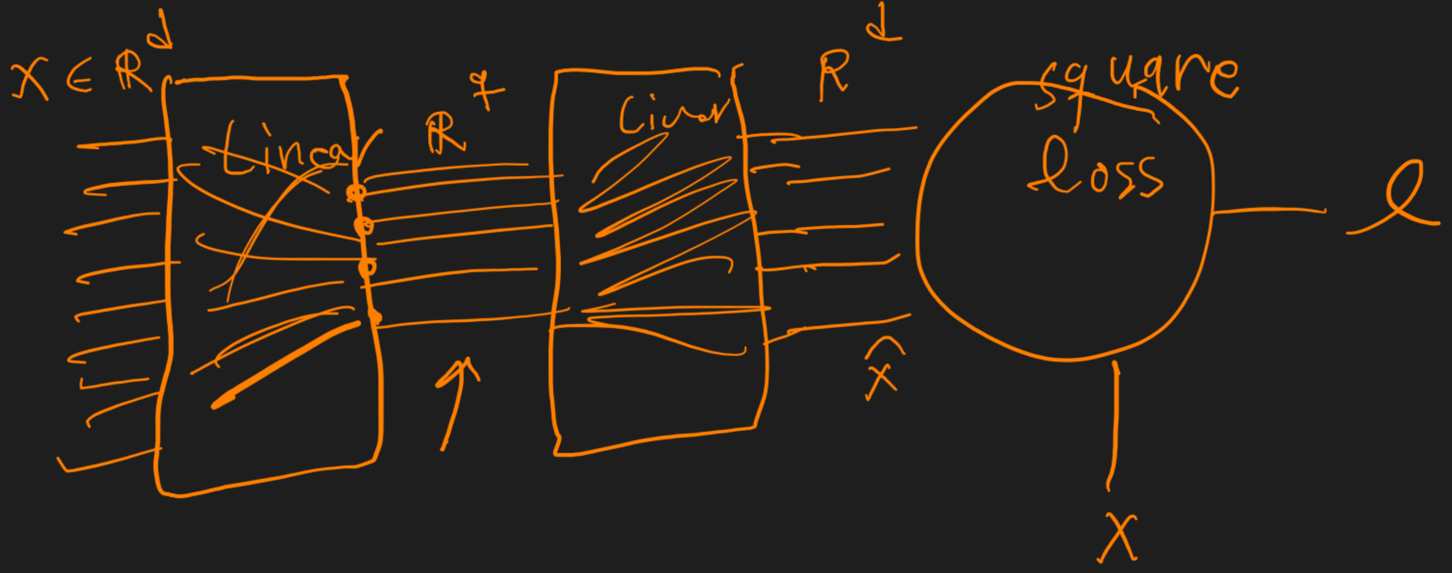
- Freezing: only update the last layers
- Fine Tuning: update the first layers as well

+last

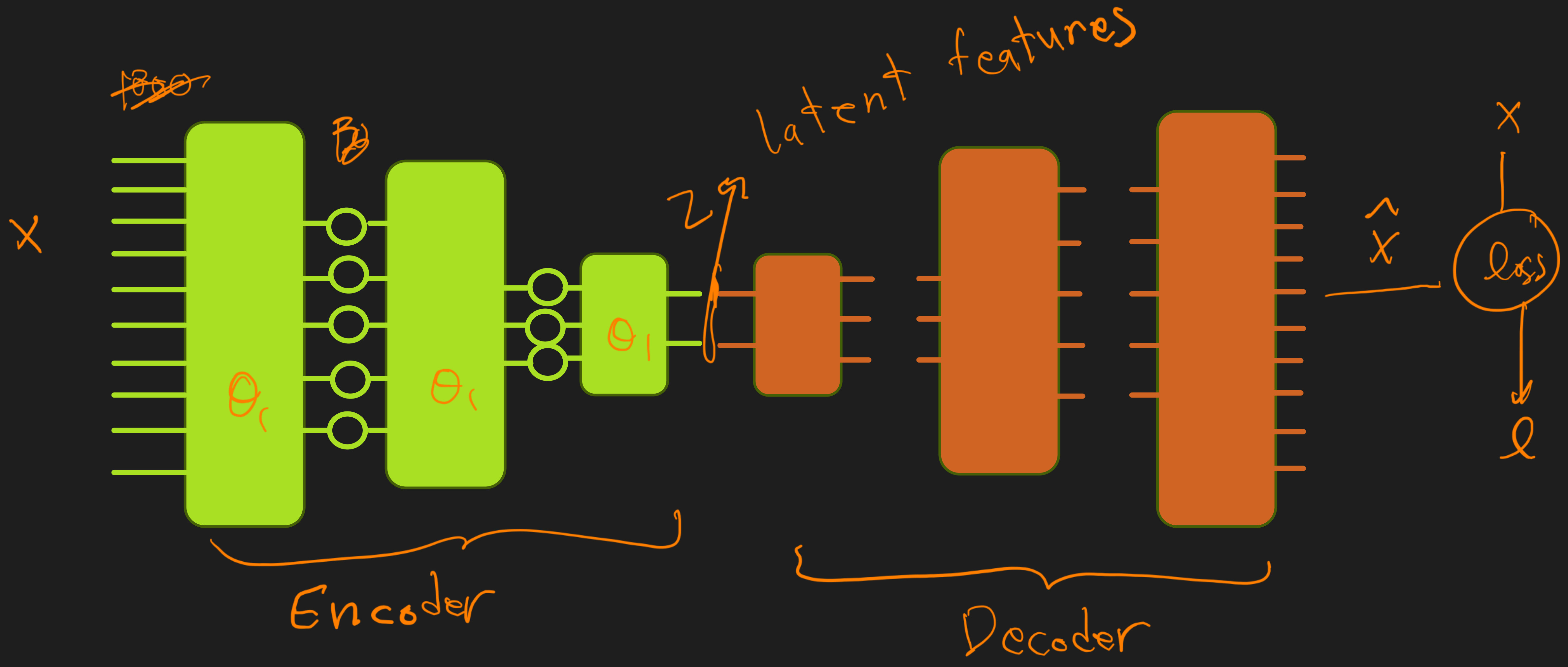


FEATURE EXTRACTION USING NEURAL NETS

- Using Neural Nets to find good "representations" of data?
- First step:
  - Can we implement PCA with neural nets?



AUTOENCODERS



AUTOENCODERS

- $Enc_{\Theta_1}: \mathbb{R}^d \to \mathbb{R}^q$ $\quad$ $Dec_{\Theta_2}: \mathbb{R}^q \to \mathbb{R}^d$
  - $q \ll d$
- $\text{MIN}_{\Theta_1,\Theta_2} \sum_x \ell(x)$
- $\ell(x) = \left\| Dec_{\Theta_2}\left(Enc_{\Theta_1}(x)\right) - x \right\|$

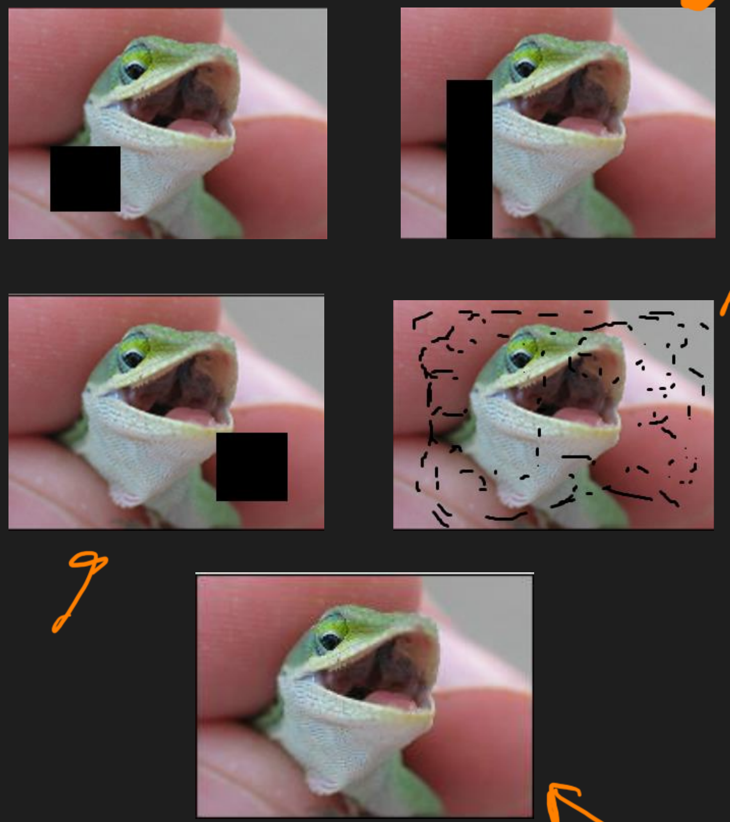
- CAN BE USED FOR COMPRESSION
  - E.G., IMAGE COMPRESSION
- CAN BE USED FOR DIMENSIONALITY REDUCTION

DENOISING AUTOENCODERS

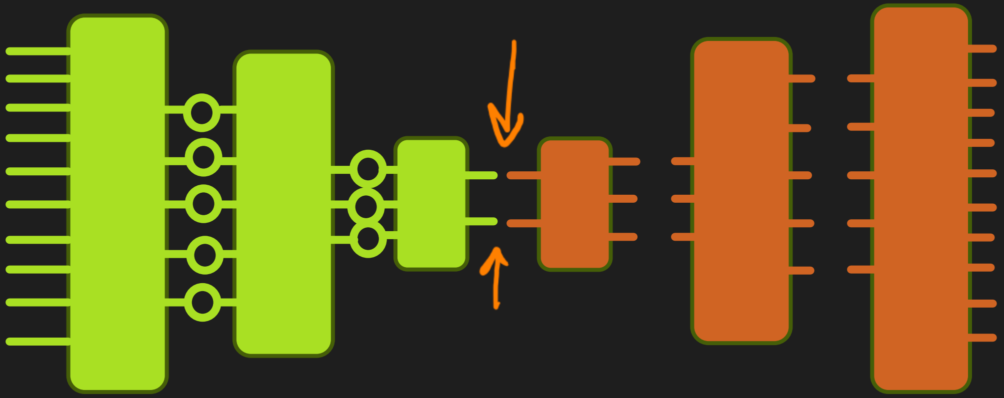
- Ideally, we want "semantically similar" data points to be close to each other in the latent space
  - In the transfer learning example, the representation was learned in a supervised manner
  - What can we can do if we have no labels?
    - For images?

USE OF IMAGE INVARIANCES

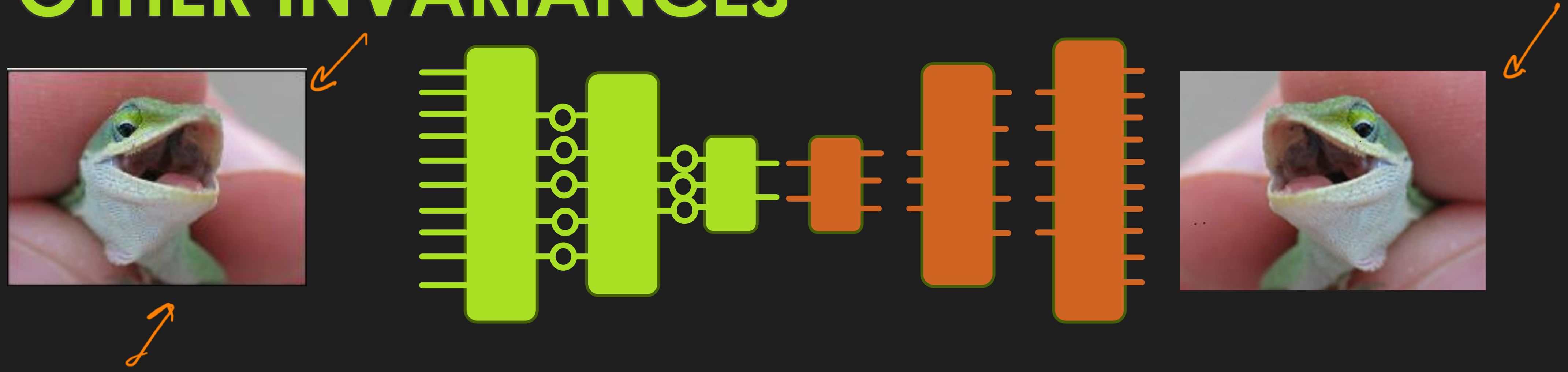
- Left images should all have the same representation



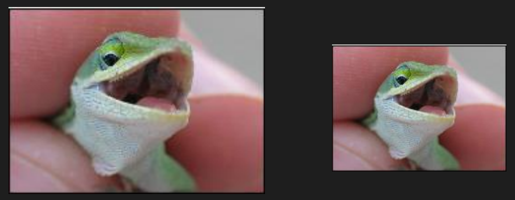
- Train a denoiser!
- $\ell(x) = \left\| Dec_{\Theta_2}\left(Enc_{\Theta_1}(noise(x))\right) - x \right\|$



OTHER INVARIANCES



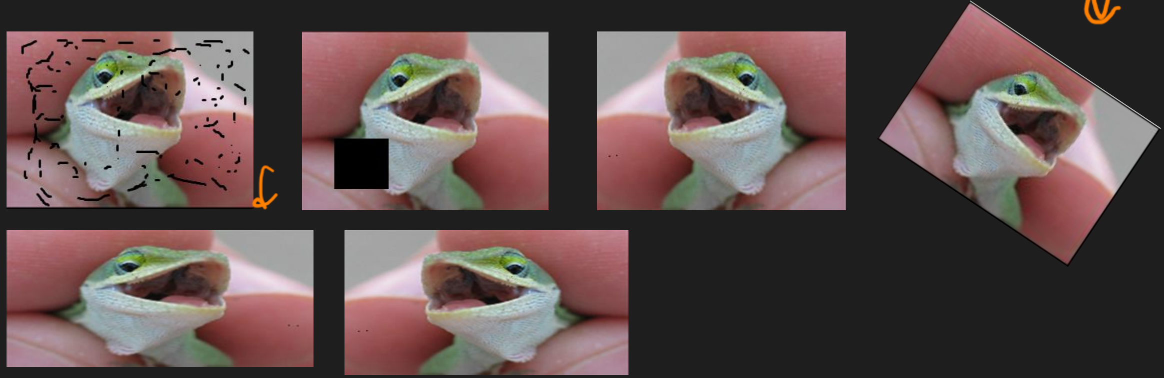
- FLIPPING AN IMAGE DOES NOT CHANGE ITS SEMANTIC
  - ONE CANNOT RECOVER THE UNFLIPPED IMAGE….
  - STILL, THESE SHOULD BE MAPPED CLOSE TO EACH OTHER
- SAME FOR SCALING, ETC.



CONTRASTIVE LEARNING

- ORIGINAL IMAGE x

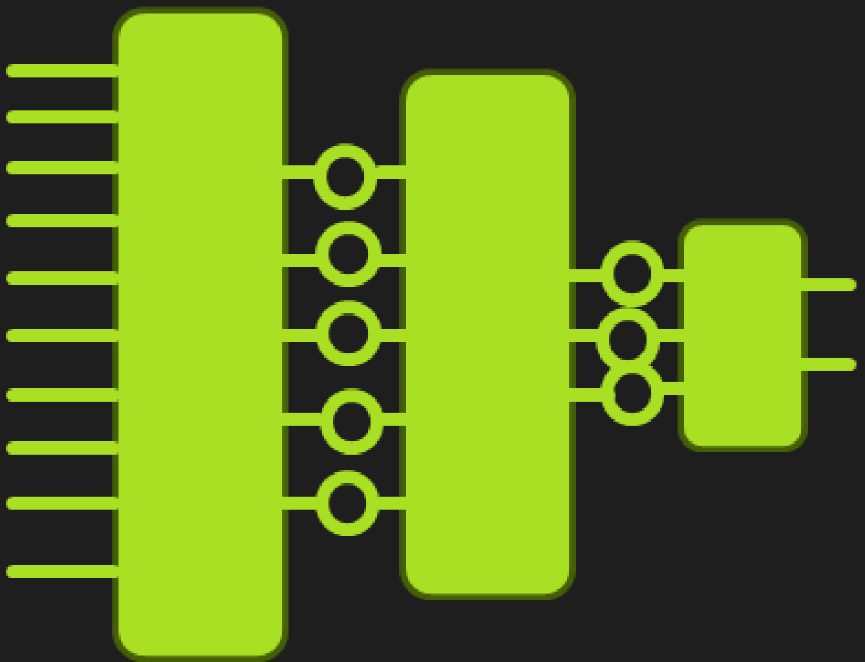
- SIMILAR IMAGES TO x
  - y SUCH THAT $(x, y) \in pos$
  - (x, y) IS A "POSITIVE PAIR"



- DISSIMILAR IMAGES TO x
  - z SUCH THAT $(x, z) \in neg$
  - (x, z) IS A "NEGATIVE PAIR"

CONTRASTIVE LEARNING

- FIND A MAPPING (ENCODER) THAT
  - MAPS SIMILAR POINTS CLOSE TO EACH OTHER AND DISSIMILAR POINTS FAR FROM EACH OTHER
- LOSS FOR POINT x
  - $(x, y) \in pos$
  - $\left(x, z_j\right)_{j=1}^{M} \in neg$
  - $-\log \dfrac{e^{\frac{<Enc(x),Enc(y)>}{t}}}{e^{\frac{<Enc(x),Enc(y)>}{t}} + \Sigma_{z_j}\, e^{\frac{<Enc(x),Enc(z)>}{t}}}$



z_3

z_1 z_2

- Contrastive + linear classification

- Figure taken from
- Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere