HW5:
Representation Learning with Autoencoder

THE OHIO STATE UNIVERSITY

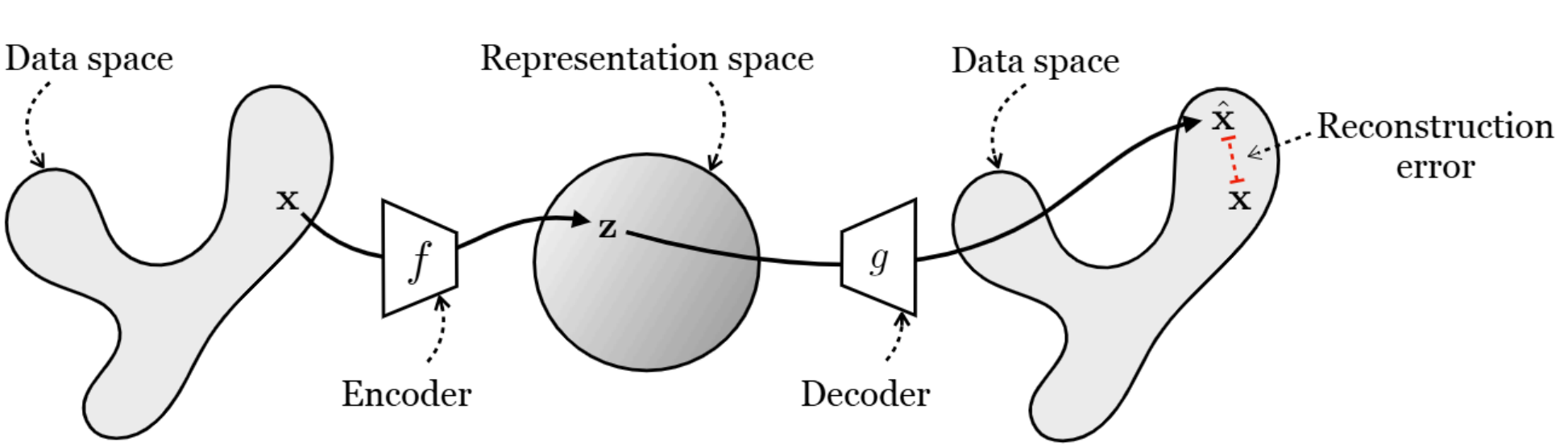
Notes

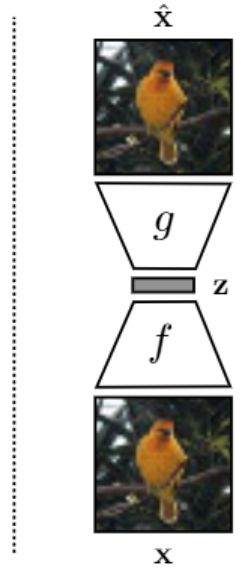
- Please make sure you read slide decks 17
- Please make sure you read chapter 30 of the textbook

Outline

- Recap of Autoencoder
- Recap of Convolution
- Homework description (programming part)
- Homework description (written part)

Autoencoder



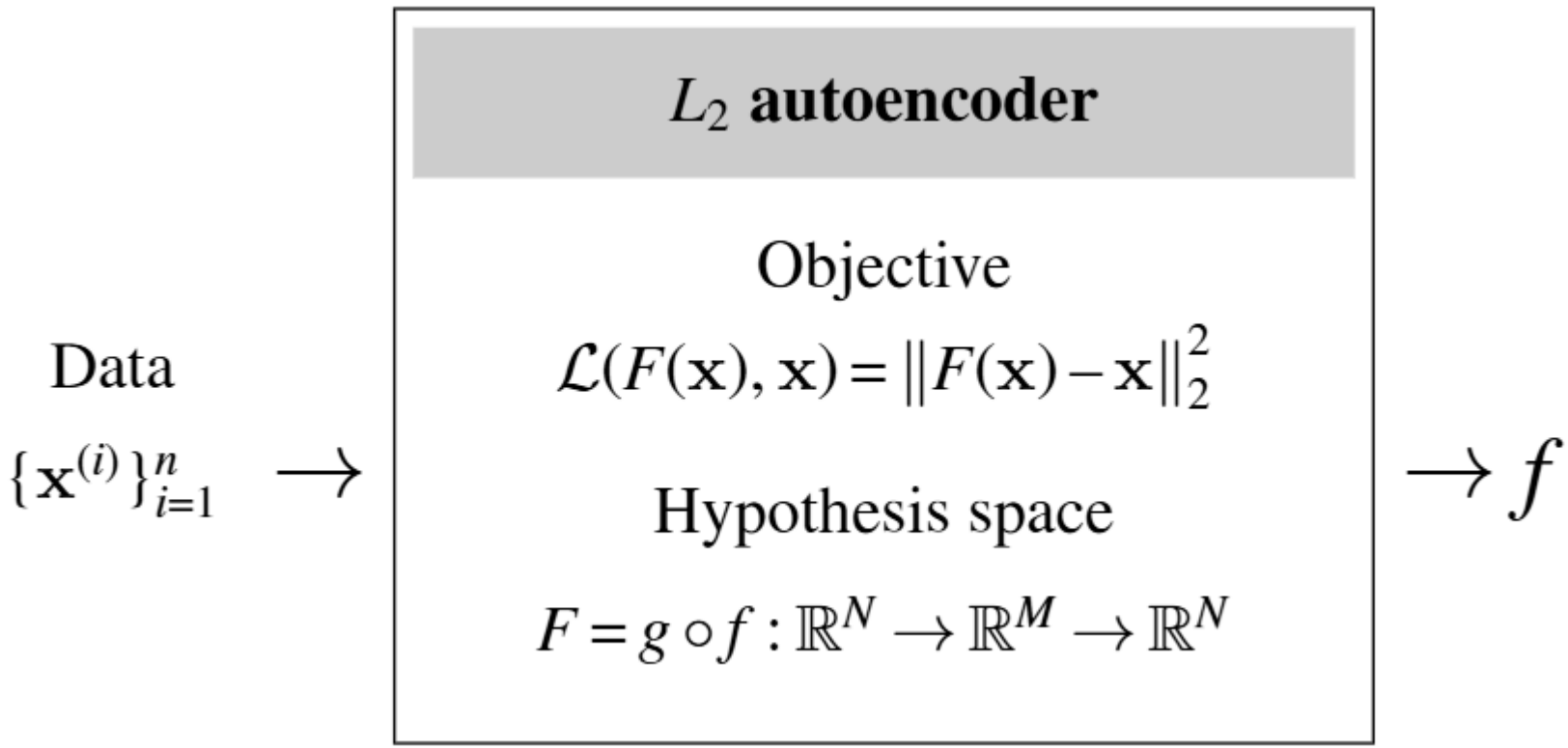


Objective function: $$f^*, g^* = \arg\min_{f,g} \mathbb{E}_{\mathbf{x}} \left\| g(f(\mathbf{x})) - \mathbf{x} \right\|_2^2$$

Design principle: **z** with lower dimensionality

[Figure credit: A. Torralba, P. Isola, and W. T. Freeman, Foundations of Computer Vision.]

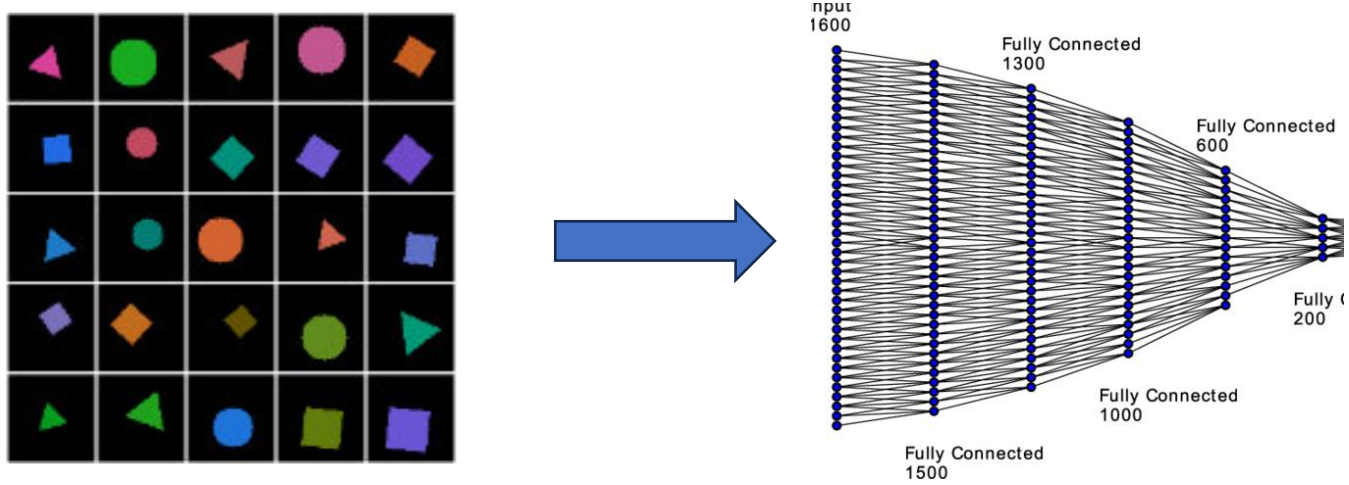
Autoencoder



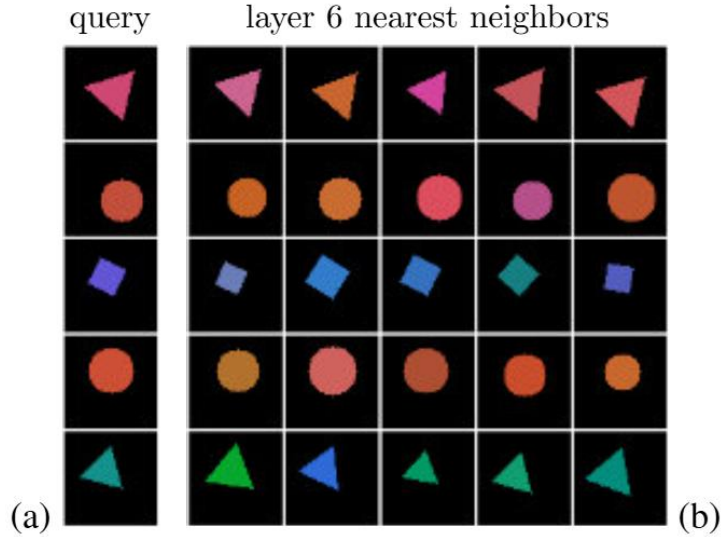
[Figure credit: A. Torralba, P. Isola, and W. T. Freeman, Foundations of Computer Vision.]

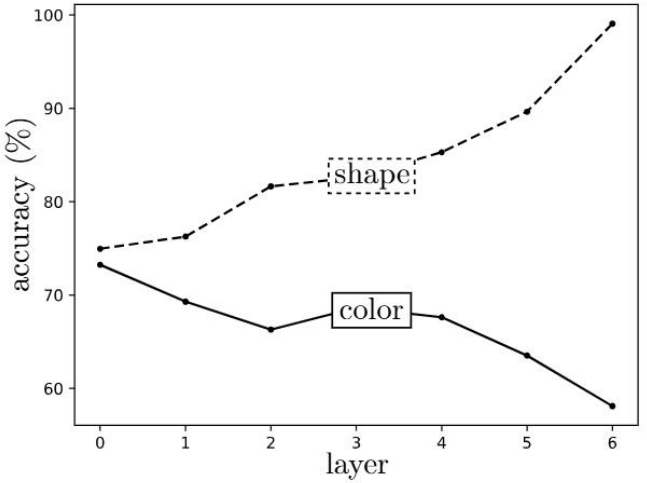
Example

Training data



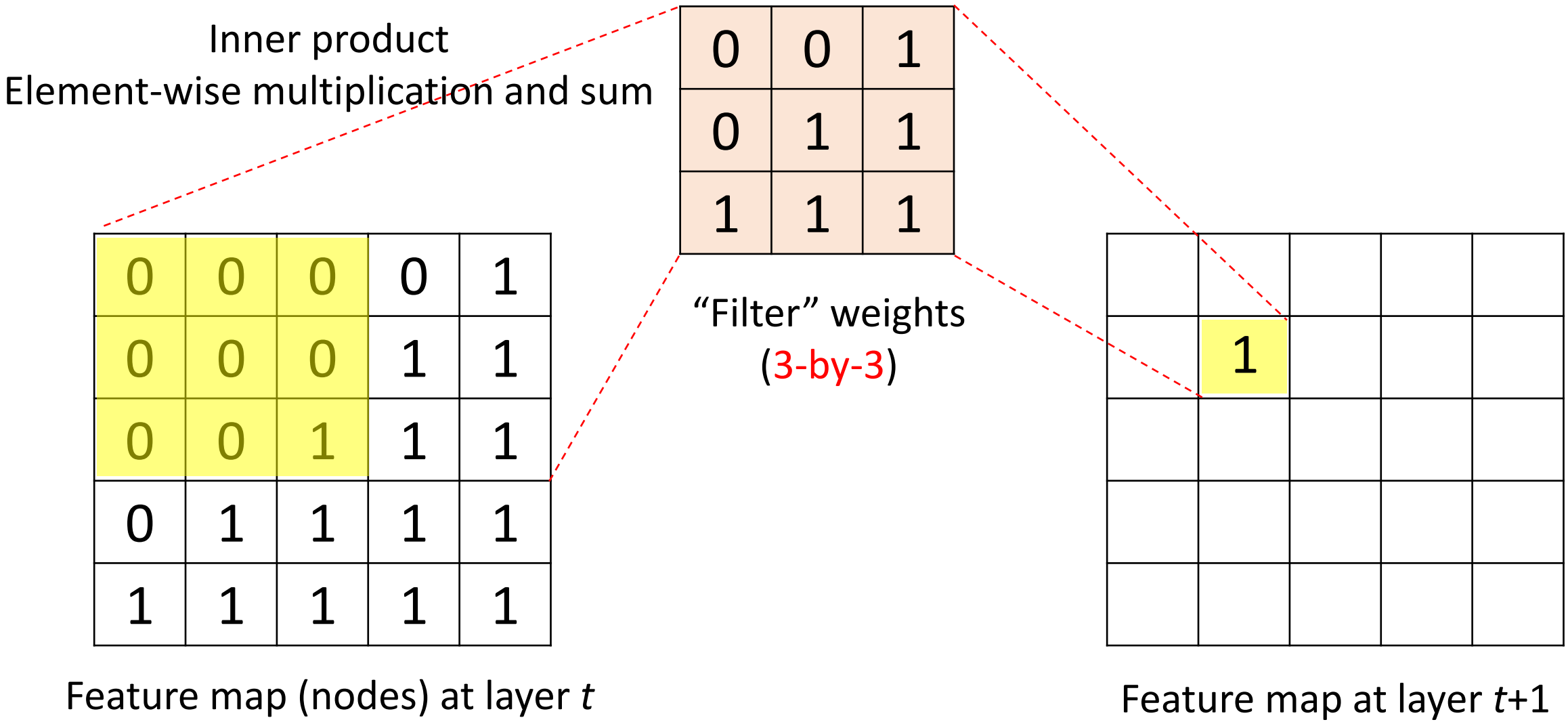
Nearest neighbor search



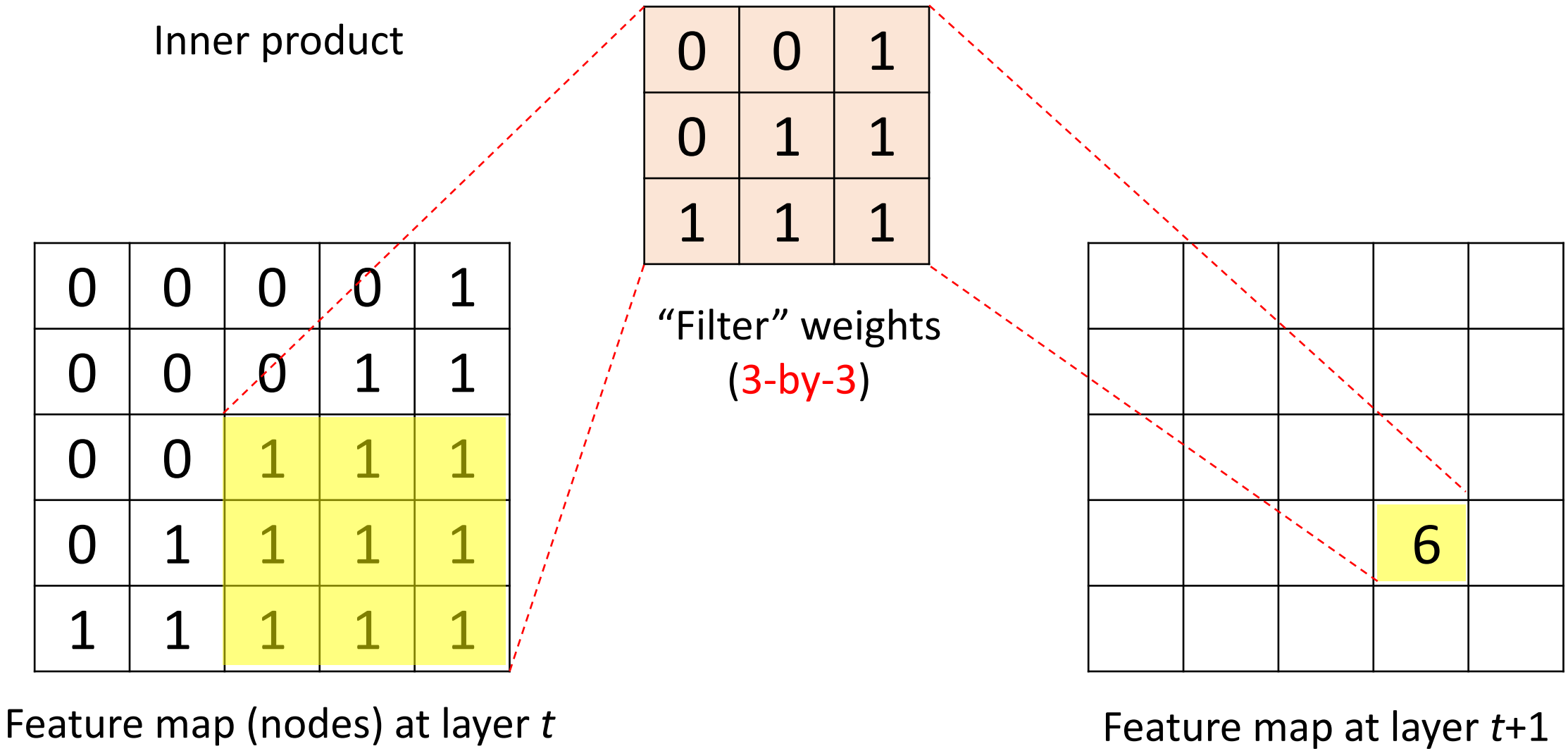


[Figure credit: A. Torralba, P. Isola, and W. T. Freeman, Foundations of Computer Vision.]

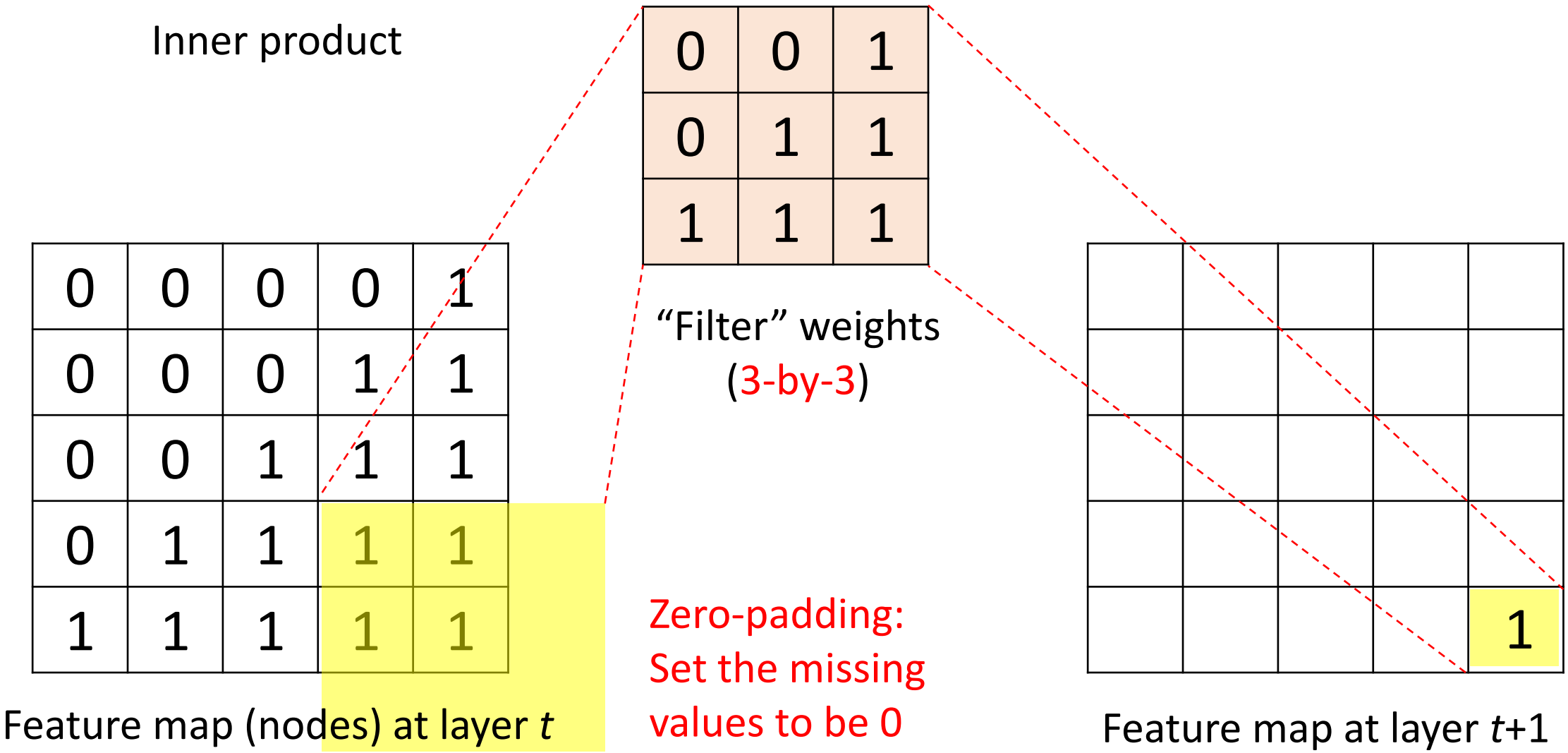
Convolution



Convolution



Convolution



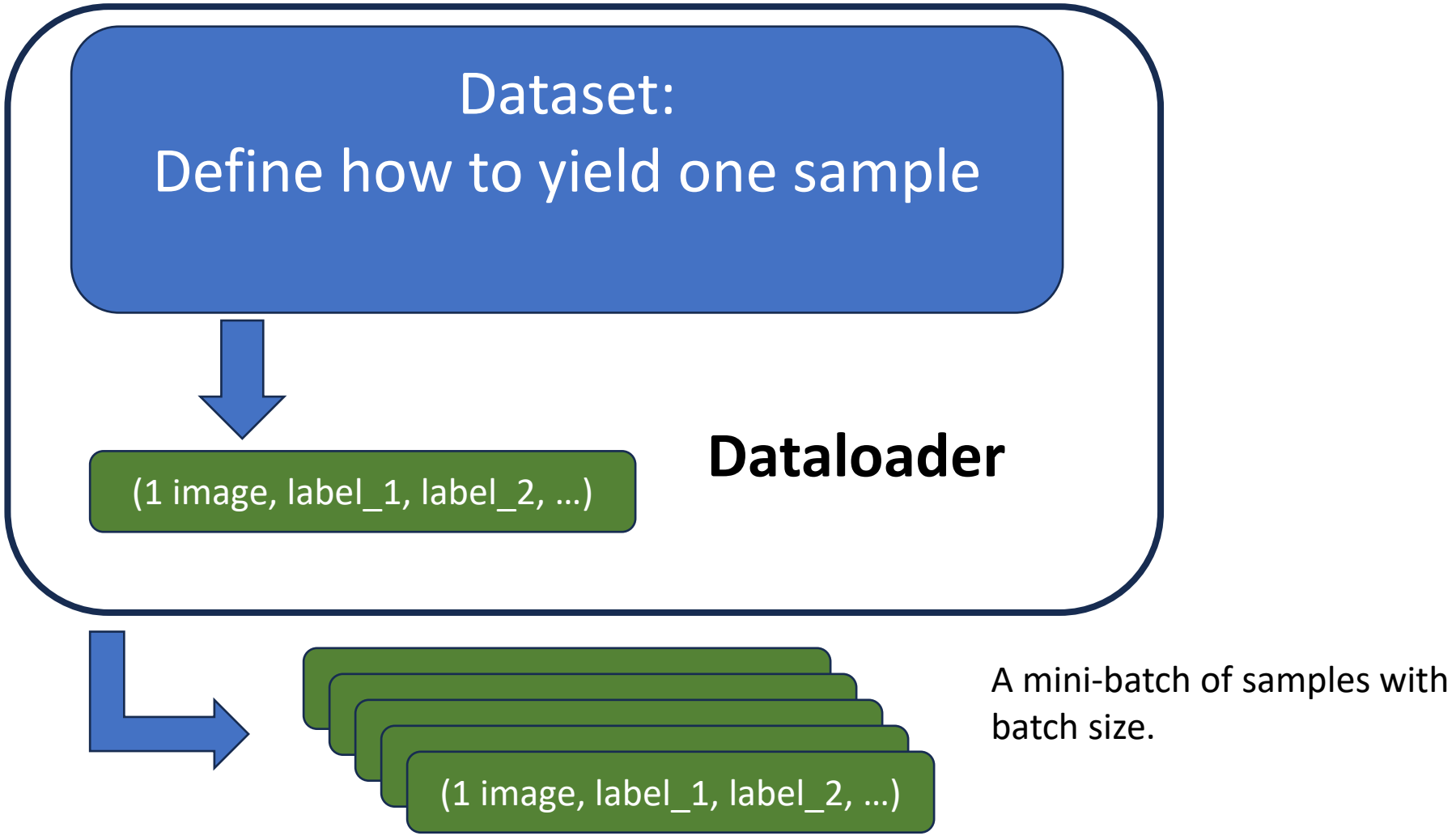
Outline

- Recap of Autoencoder
- Recap of Convolution
- Homework description (programming part)
- Homework description (written part)

Q1: Generate the Training Dataset

- In Q1, you are to implement the following steps
  - Generate the training data with **THREE SHAPES** and **EIGHT COLORS**
  - You will be given the following
    - s: scale for the shape
    - cx and cy: the center of the shape
  - Shapes:
    - Circle: Draw a circle centered at (cx, cy) with a diameter of s. Use d.ellipse().
    - Square: Draw a square centered at (cx, cy), with side length s, and rotated by a random angle. Use d.polygon().
    - Triangle: Draw an equilateral triangle centered at (cx, cy), scaled by s, and rotated by a random angle. Use d.polygon().

Dataset and Dataloader



Q2 Create the AutoEncoder and Train It

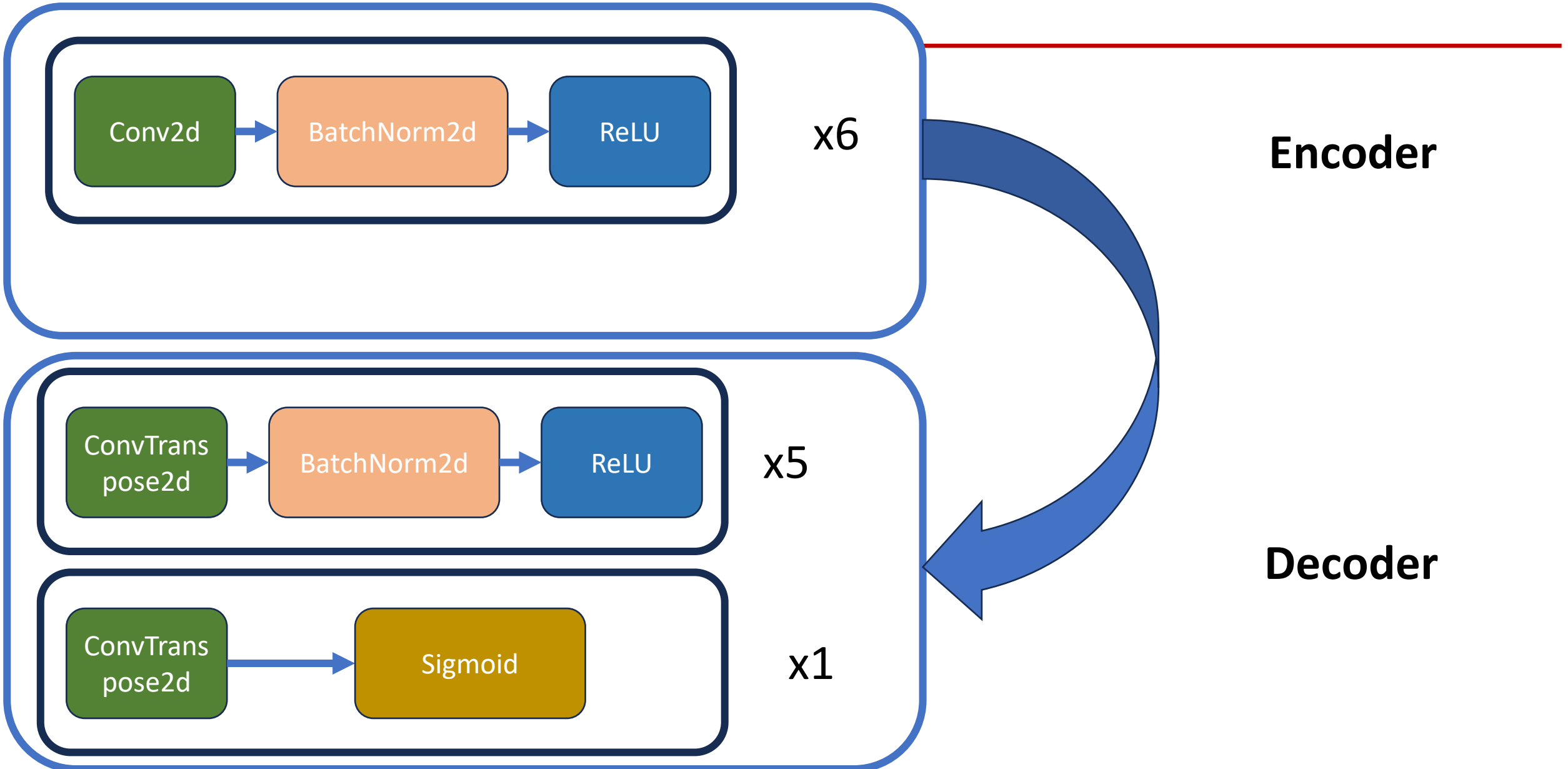
ColorShapeDataset is responsible for loading each image (given an index) and preparing the labels

- In __getitem__, you need to load one path from all_paths using idx
- Load the image with PIL.Image, apply the transform
- Parse shape & color labels from the filename using parse_labels.

build_dataloader is responsible for creating an instance of the Dataset and wrapping it inside the DataLoader to yield data batch by batch:

- Instantiate a ColorShapeDataset with build_transform().
- Wrap it in a DataLoader with the given batch_size, shuffle, and num_workers.

Q2 AutoEncoder



Q3 K Nearest Neighbor

Distance matrix ‘dist’

Num_ref_embeddings

Num_query_embeddings	0.1	0.01	0.052	0.12	0.04	0.5	0.3	0.03	0.06

For each query embedding,

- You will calculate the distance between this query embedding and all the reference embedding
- Thus, ‘dist’ has shape num_query_embembedings x num_ref_embeddings
- For K= 3, you will select the closest 3 reference embeddings to plot.
- For the distance, please use **cosine distance**, that is, pre-normalize each vector to have its L2 norm equal to 1, and compute the L2 distance

Q4 Achieve the best shape accuracy

- You **must NOT** use any images from the **test set** (data_folder_test) for training.
- You should **NOT** do any **supervised** training using **labels** from the training data

Potential directions

- Modify the autoencoder architecture
  - Larger model?
  - Better architecture?
- Change the training setup
  - Better loss?
  - Adjusting hyper-parameters?
- Change the training set or adding data augmentations:
  - More data?
  - Data augmentation?