# Statistical Machine Learning

Convolutional Neural Networks

Workshop 7

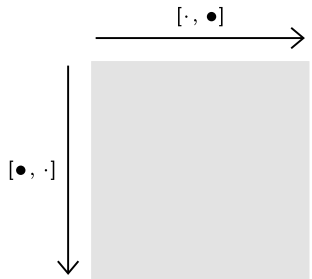
September 6, 2019

# (Py)Torch

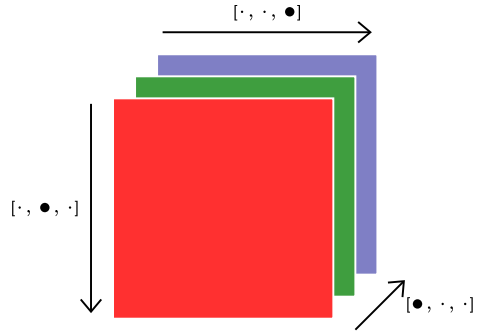

- Python library built on top of C++ backend of computational library Torch.
- Designed for efficient tensor operations on CPU/GPU.
- Automatically builds graph of operations to compute the gradient of any quantity with respect to any tensor involved.

# Image Tensors

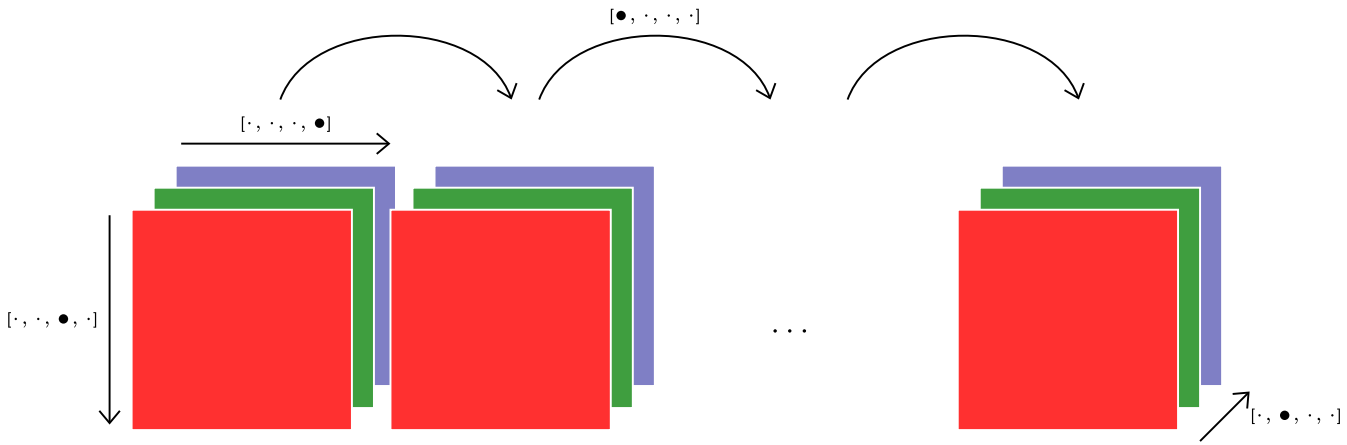
2d tensor (*e.g.* grayscale image)




3d tensor (*e.g.* rgb image)

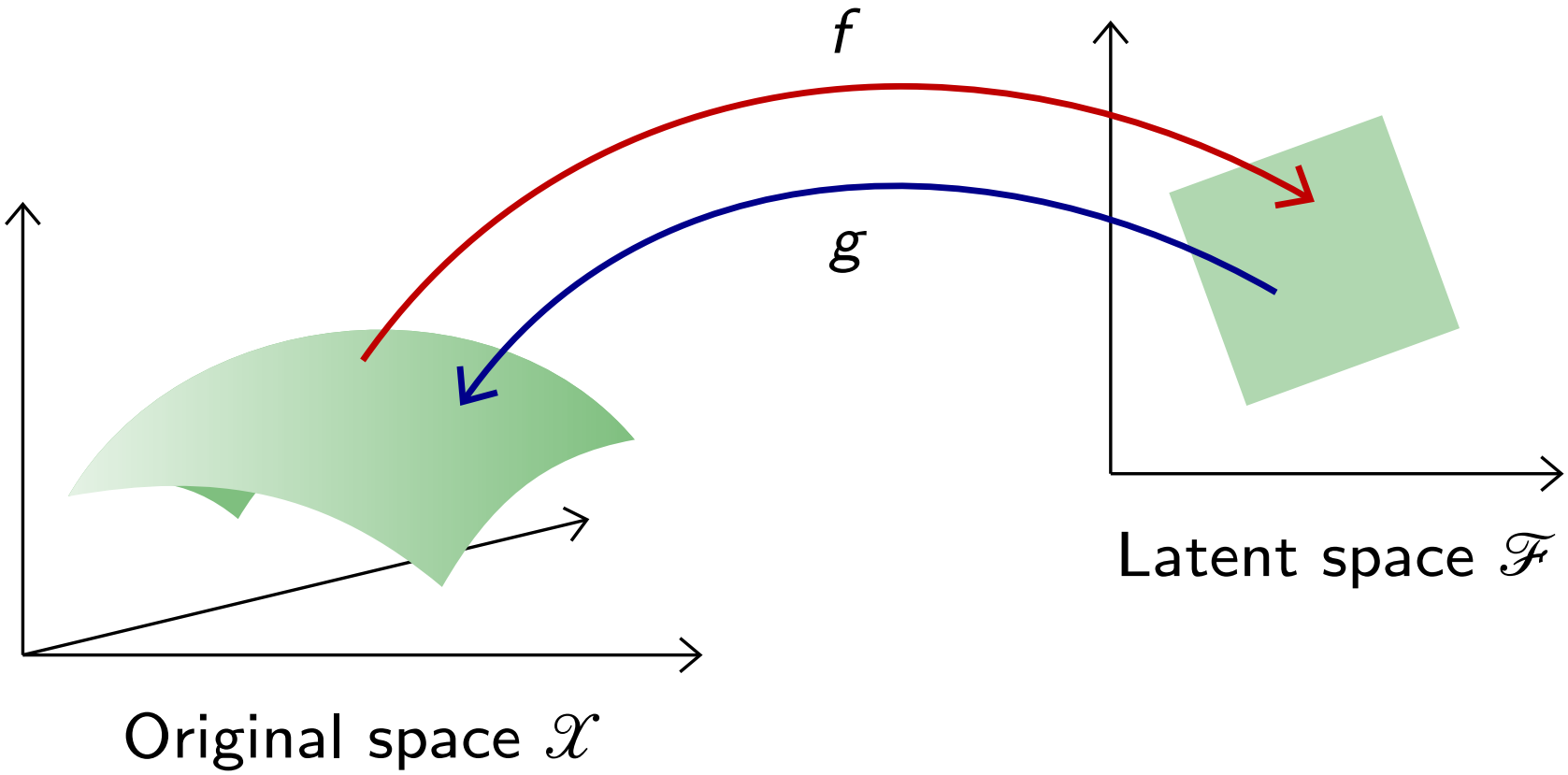



4d tensor (*e.g.* sequence of rgb images)




François Fleuret - Autumn 2019 Lecture slides.

# Autoencoders

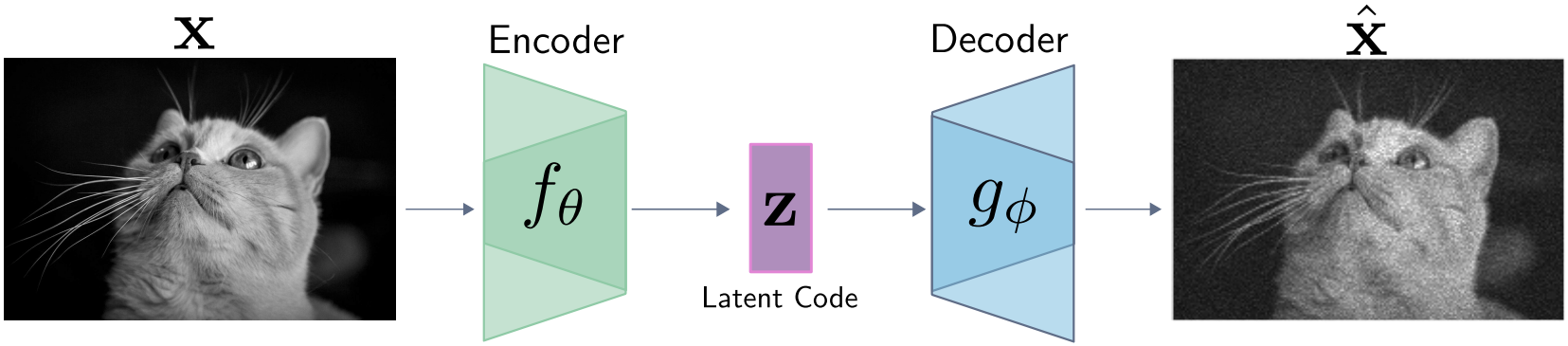



François Fleuret - Autumn 2019 Lecture slides.

# Autoencoders

- Learn encoder $f : \mathscr{X} \to \mathscr{F}$ from original to latent space.
- Learn decoder $g : \mathscr{F} \to \mathscr{X}$ from latent to original space.
- Unsupervised mapping. $g \circ f$ should be close to identity. Minimize quadratic loss over data:

$$\min_{f,g} \sum_{k} \|x_k - g \circ f(x_k)\|^2 \tag{1}$$



# Hands-on

1. Download `worksheet-xx.ipynb` from the LMS.
2. Move the worksheet to a working directory `$WORKDIR`
3. `cd $WORKDIR`
4. Start → Anaconda3 (64-bit) → Anaconda Prompt
5. Enter the following at the prompt:
   `conda install pytorch torchvision cpuonly -c pytorch`
6. Launch Jupyter
   `jupyter notebook`
7. The Jupyter UI should open in a web browser.
8. Click on `worksheet-xx.ipynb` to get started.

You can work on the notebooks at home if you install the Anaconda3 distribution on your machine.