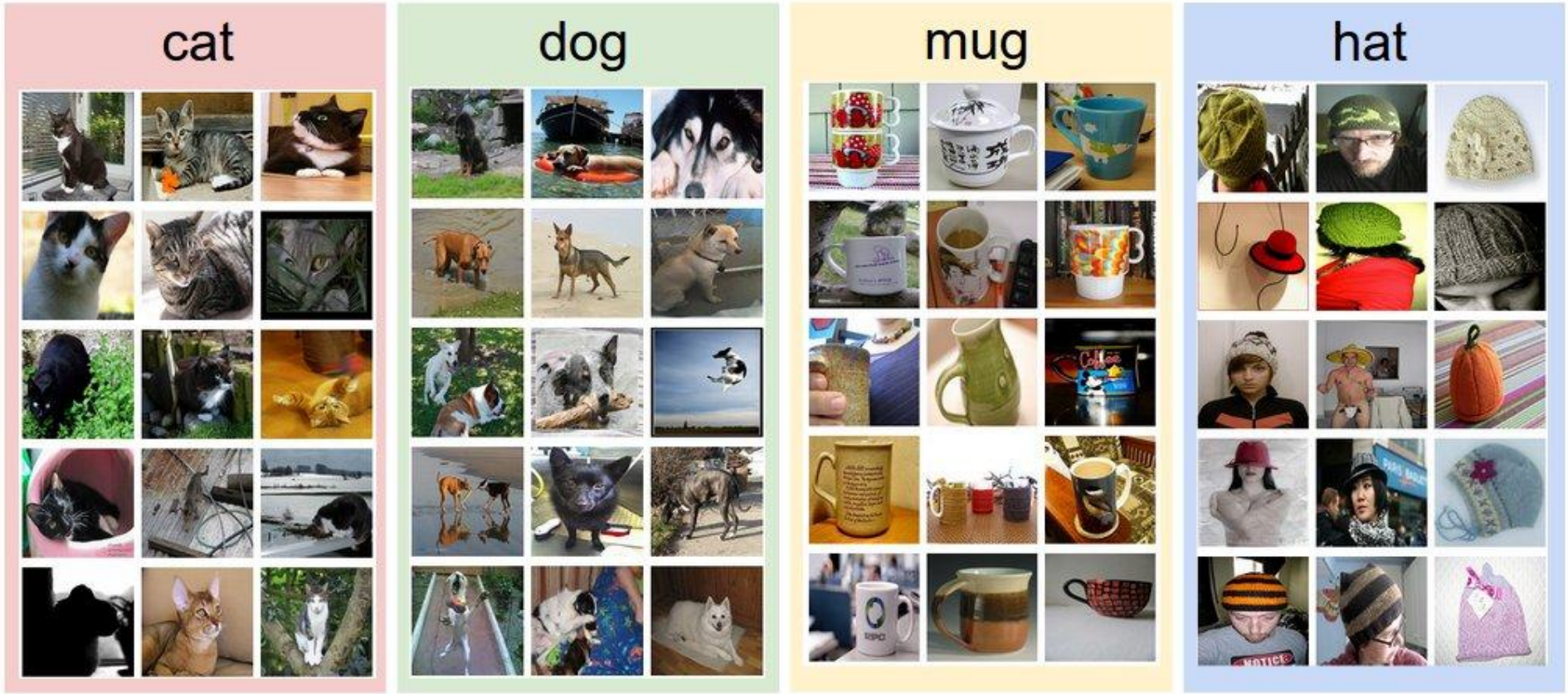
Assignment I: Image Classification

Tianshuo Yang (yangtianshuo@connect.hku.hk)

1 Image Classification

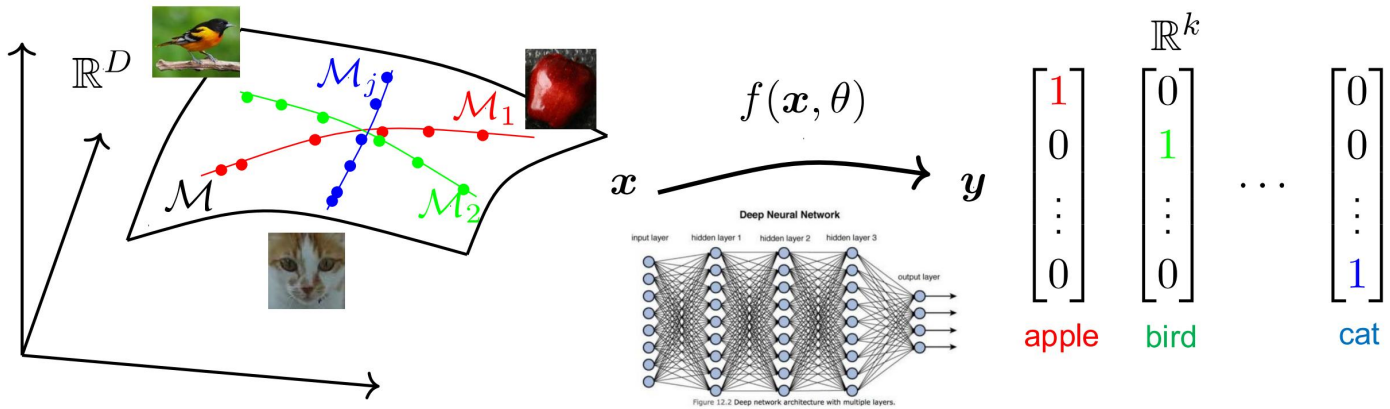


1.1 What will you learn

- Familiarize yourself with the usage of the HKU CS GPU Farm
- Gain experience in implementing neural networks with a popular deep learning framework PyTorch
- Develop a deep learning system (CNN) for recognition, including
  - network design & training
  - hyperparameter tuning
  - model inference
  - evaluation & visualization

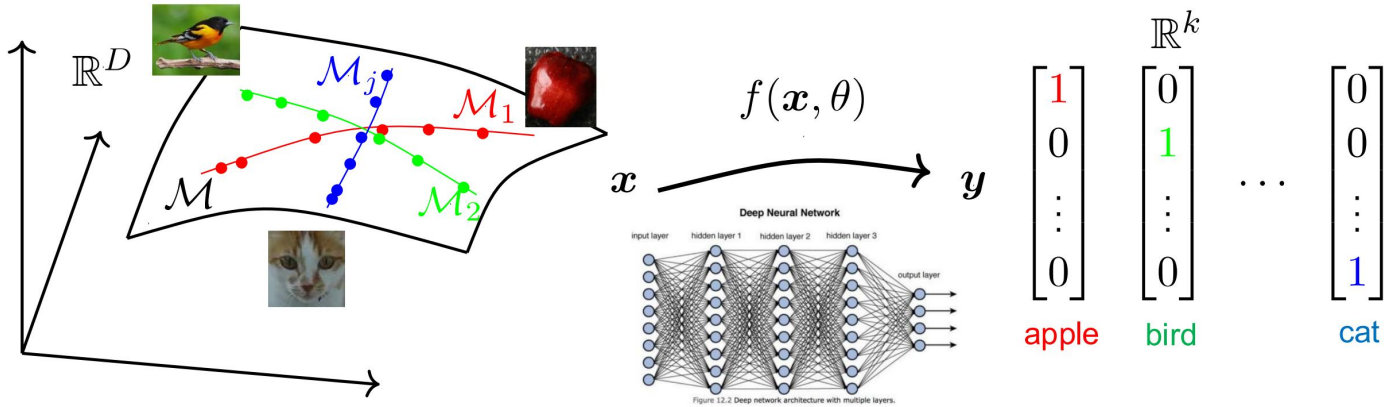
1.2 Why Recognition

Assignment **1**: Recognition

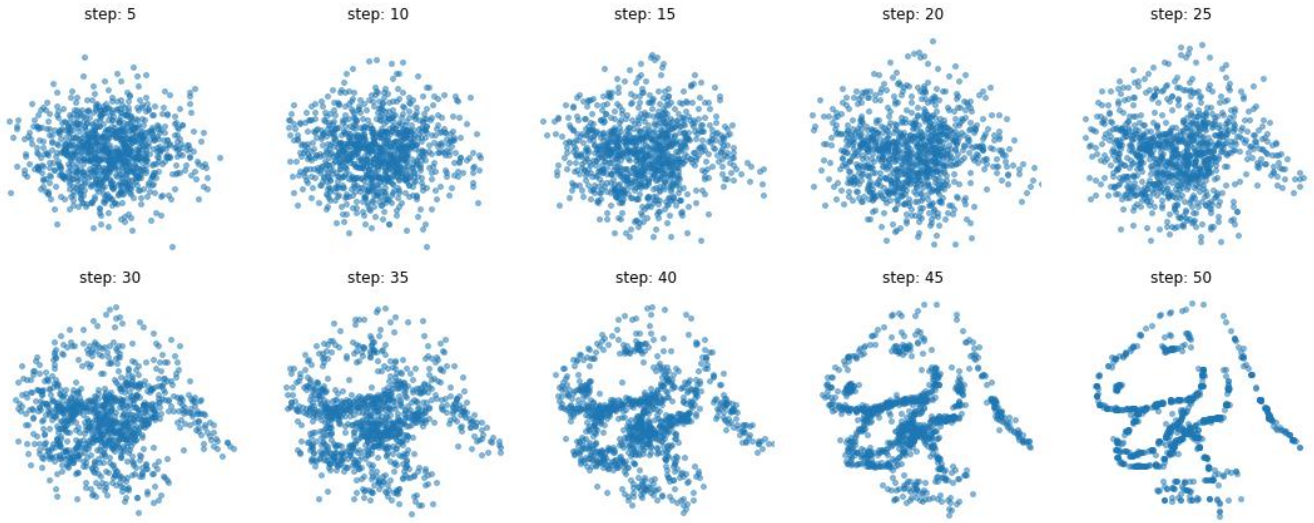


1.2 Why Recognition

Assignment **1**: Recognition



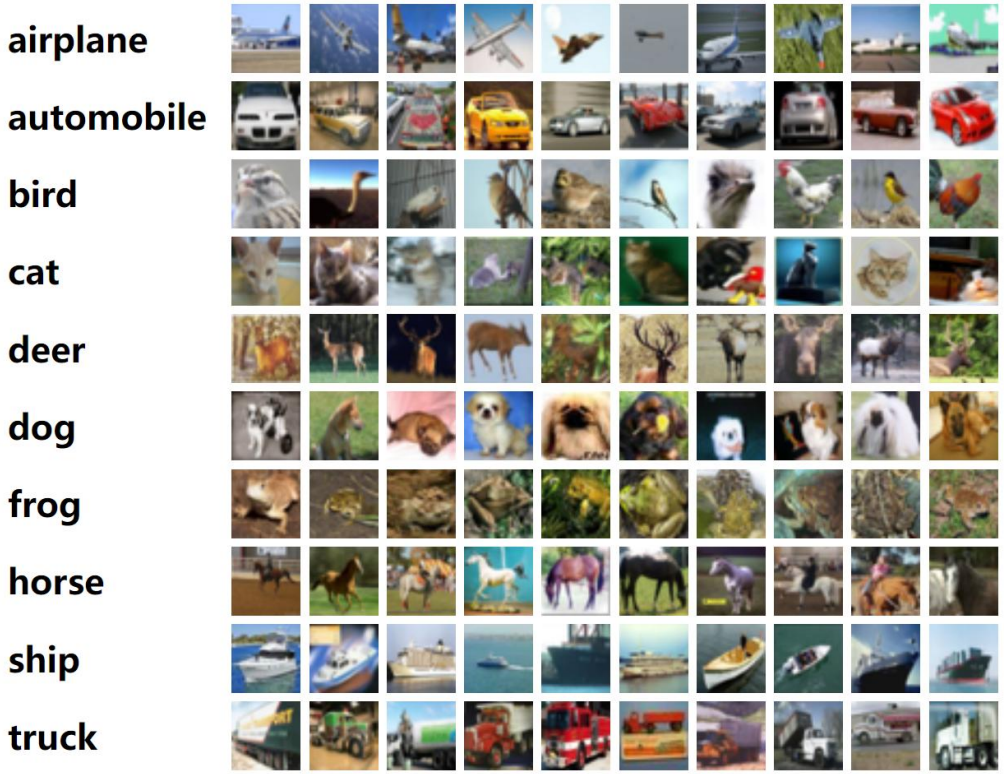
Assignment **2**: Generation



1.2 Why Recognition

Bi-directional Encoding and Decoding

(recognition and generation)



CIFAR-10 Dataset

1.2 Why Recognition

Bi-directional Encoding and Decoding

(recognition and generation)

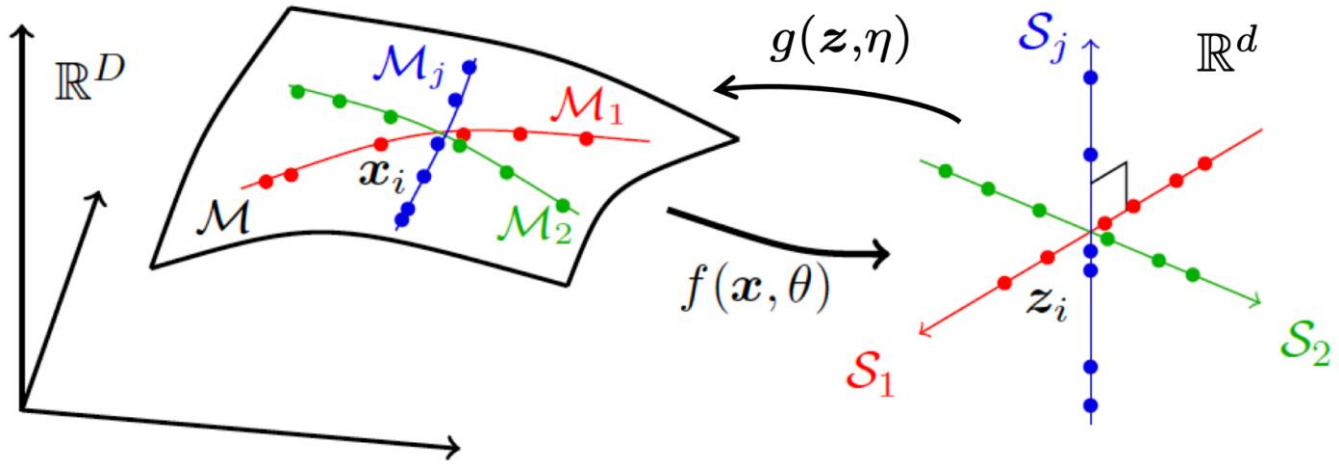


Image @ HKU DATA8014 Principles of Deep Representation Learning, Prof. Yi Ma

2 Working on the Assignment

- Set up the **HKU GPU Farm** (Recommended) to ensure you can access the GPU and run **Jupyter Lab**
- Learn the basic code and complete the assigned tasks in the Jupyter Notebook **Assignment_1.ipynb**
- Details on the Github Page: HKU-DASC7606-A1

2.1 HKU CS GPU Farm

- gpu-farm-quickstart

1. Applying for an Account (GPU Farm Phase 2)
2. Accessing the GPU Farm
3. Using GPUs in Interactive Mode
4. Environment Setup (On GPU compute node)
5. Set up Jupyter Lab

2.2 Assignment Tasks

- Task 1: Fill in the blank

There are ***9*** code blocks in the Jupyter notebook that require you to complete them.

- Task 2: Write a report (no more than 2 pages)

Your report should include three main sections: introduction, method, and experiment.

2.3 Task 1: Fill in the blank

Let's use BN

Add BatchNorm2d to the convolution neural network you implemented.

You should add batchnorm after the convolution operator and before the activation layer.

Please train this network and show the test accuracy.

```
import torch.nn as nn
import torch.nn.functional as F

class ConvolutionBNNet(nn.Module):
    def __init__(self):
        super().__init__()
        '''
        Implement here.
        '''
        pass

    def forward(self, x):
        '''
        Implement here
        '''
        pass
        return x
```

2.3 Task 1: Fill in the blank

- Save the model's training and prediction performance, or visualization results, in the notebook (or alternatively save them as a new .txt file, etc.).

- The last blank is an open-ended design, where you can apply various deep learning methods to try to improve performance.

- It is allowed to refactor the notebook into a set of well-organized Python files.

2.4 Task 2: Final Report (PDF, up to 2 pages)

- Introduction. Briefly introduce the task & background & related works.
- Methods. Improvements to the baseline model, including but not limited to the methods above.
- Experiments & Analysis (IMPORTANT). Analysis is the most important part of the report.
- Details on the Github Page

3 Files to submit

1. **Final Report** (PDF, up to 2 pages)

2. **Codes**
   a) All the code files including the Assignment_1.ipynb file with model output.

   b) README.txt if you added some python files.

3. Model Weights
   a) In the format of **model checkpoint link** (model_link.txt) due to the limitation on submission file size.

3 Files to submit

1. **Final Report**
2. **Codes**
3. **Model Weights**

If your student id is 30300xxxxx, then the compressed file for submission on Moodle should be organized as follows:

```
30300xxxxx.zip
├── report.pdf
├── your code (Must include the Assignment_1.ipynb file)
├── model_link.txt
└── (optional) README.md
```

4 Important Dates

- September 12, 2024 (Thu.): The assignment release.

- October 8, 2024 (Tue.): Submission deadline (23:59 GMT+8).

Late submission policy:

- 10% for late assignments submitted within 1 day late.
- 20% for late assignments submitted within 2 days late.
- 50% for late assignments submitted within 7 days late.
- 100% for late assignments submitted after 7 days late.

5 Marking Scheme

- Coding and Model Performance (80% of total marks)

  Marks will be given based on your coding and the performance of your model.

  (Accuracy on test set)

- Accuracy above 80% will get the full mark of this part.
- Accuracy between 70-80% will get 90% mark of this part.
- Accuracy between 65-70% will get 80% mark of this part.
- Accuracy between 60-65% will get 70% mark of this part.
- Accuracy between 50-60% will get 60% mark of this part.
- Others will get 0% mark.

- Final Report (20% of total marks)

5 Marking Scheme

- Coding and Model Performance (80% of total marks)

- Final Report (20% of total marks)

  The marks will be given mainly based on the richness of the experiments & analysis.

- Reasonable number of experiments + analysis: 90%-100% mark of this part.
- Basic analysis: 80%-90% mark of this part.
- Not sufficient analysis: lower than 80%.

6 Need More Support?

- For any questions about the assignment which potentially are common to all students, your shall first look for related resources as follows,
  - ✓ We encourage you to use GitHub Issues of this repository.
  - ✓ Or if you prefer online doc: Discussion doc.

- For any other private questions, please contact Tianshuo Yang via email.

Questions!

Any more questions, please contact yangtianshuo@connect.hku.hk