深度學習系統與實現

LAB01 – Train your first AI model

Dept. of Computer Science and
Information Engineering
National Chiao Tung University

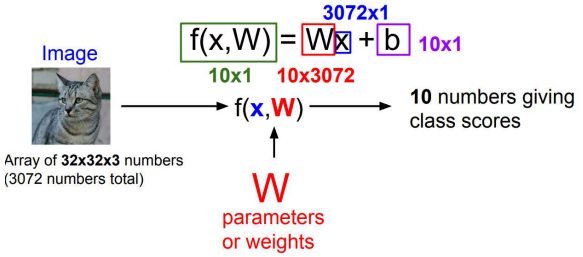
Outline

- Background
- Introduce pytorch framework
- LAB 1-1, Train a CNN model via pytorch
- LAB 1-2, Train a new dataset
- LAB 1-3, Tuning hyperparameters
- Notice

Background

- Task: classification
  - Linear classification (score function)
- Classification in DL
  - Score function + Loss function + Optimization
  - Loss function – softmax + cross entropy
  - Optimization – mini-batch gradient descent
- CNN architecture

Linear Classifier (score function)

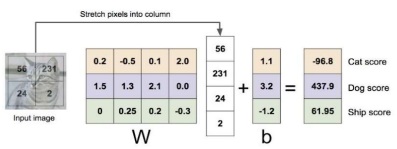


$$f(x,W) = Wx + b$$

$$\varphi(x) = [W \quad b]\begin{bmatrix} x \\ 1 \end{bmatrix} = Wx$$

Algebraic Viewpoint

f(x,W) = Wx

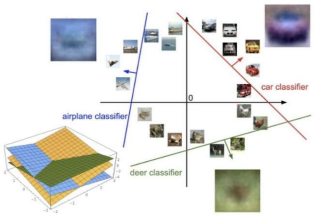


Visual Viewpoint

One template per class

Geometric Viewpoint

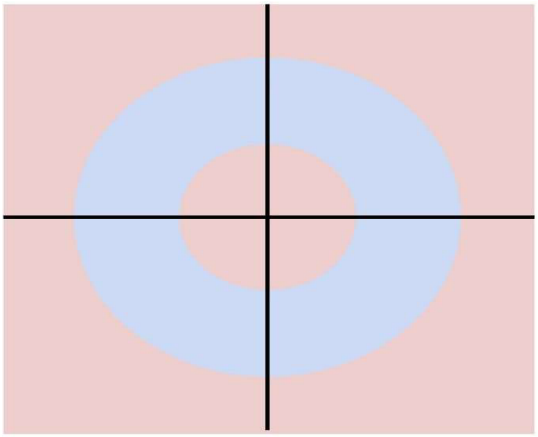
Hyperplanes cutting up space



Linear Classifier (score function)

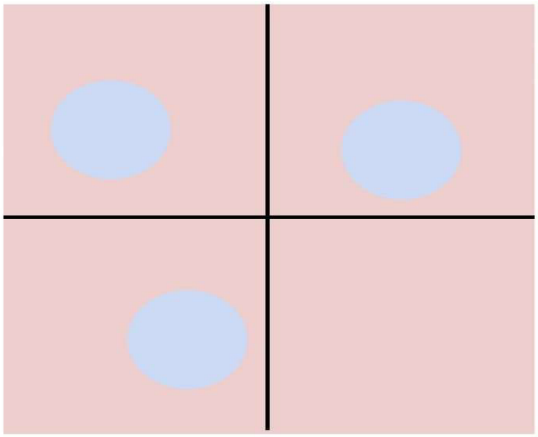
Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else

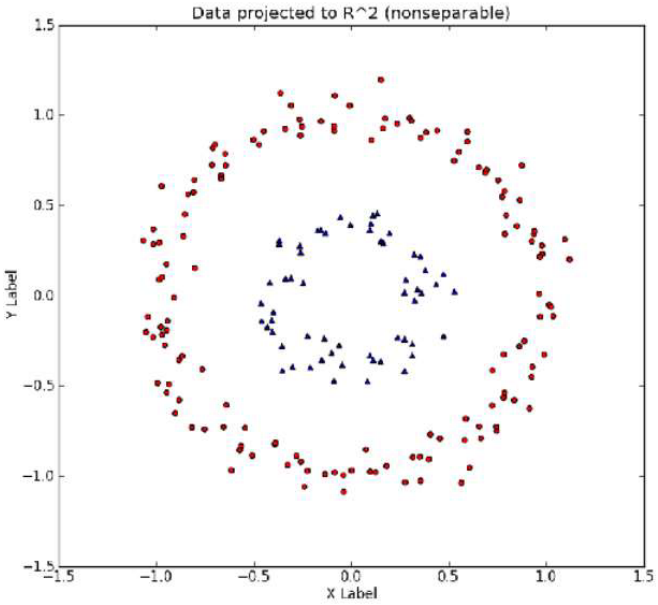


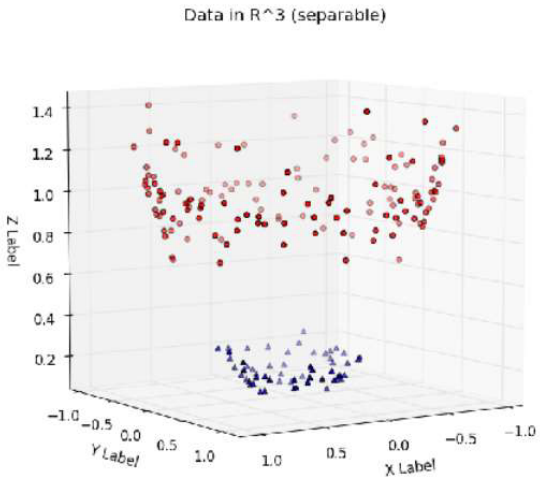
Class 1:
Three modes

Class 2:
Everything else

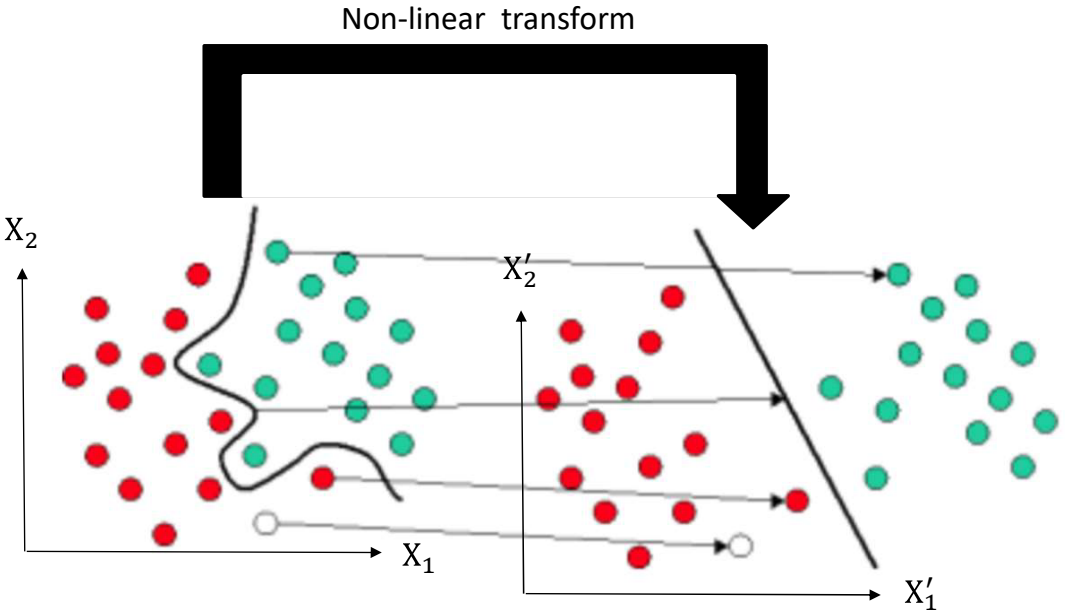


Linear Classifier





Add some Non-linear function

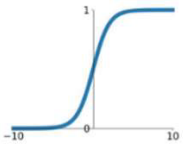


Need activation function !!

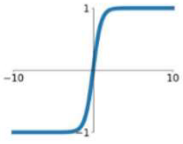
Add some Non-linear function

Activation Functions

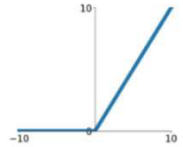
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



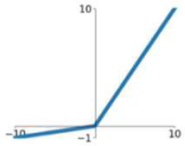
tanh
$\tanh(x)$



ReLU
$\max(0, x)$

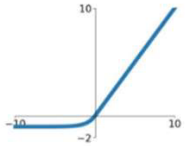


Leaky ReLU
$\max(0.1x, x)$



Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Different Activation Functions and their Graphs

Loss function

- **Score function:** map raw data to class scores
  - Scoring function, classification function
  - Max score refers to the class of the prediction
  - Linear transformation
- **Loss function:** quantify the difference between scores and ground truth labels
  - Cost function, objective
  - Evaluate how well or how bad the model perform
  - Only required during training

Loss function

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$s = f(x_i; W)$

$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$ Softmax Function

Probabilities must be >= 0

Probabilities must sum to 1

$L_i = -\log P(Y = y_i|X = x_i)$

	Unnormalized log-probabilities / logits	exp	unnormalized probabilities	normalize	probabilities		Correct probs
cat	3.2		24.5		0.13	compare	1.00
car	5.1		164.0		0.87		0.00
frog	-1.7		0.18		0.00		0.00

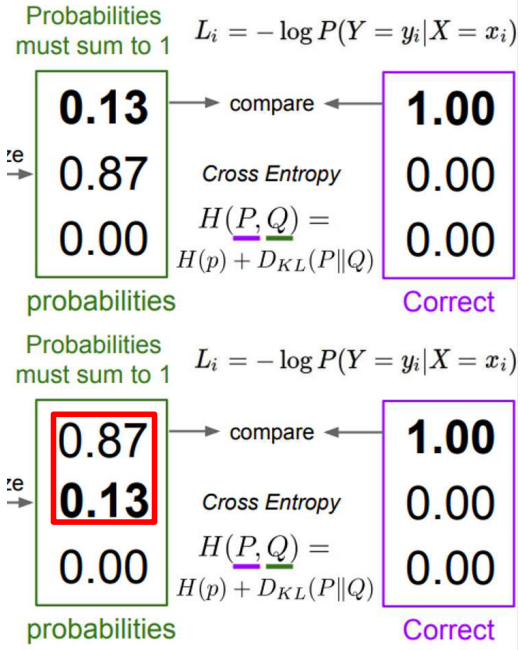
Cross Entropy

$H(P, Q) = H(p) + D_{KL}(P||Q)$

$H(p, q) = -\sum_x p(x) \log q(x)$

Loss function

Softmax Classifier (Multinomial Logistic Regression)



$$H(p,q) = -\sum_x p(x) \log q(x)$$

High Loss

Case1: -(1 × log(0.13) + 0 + 0) = 0.886

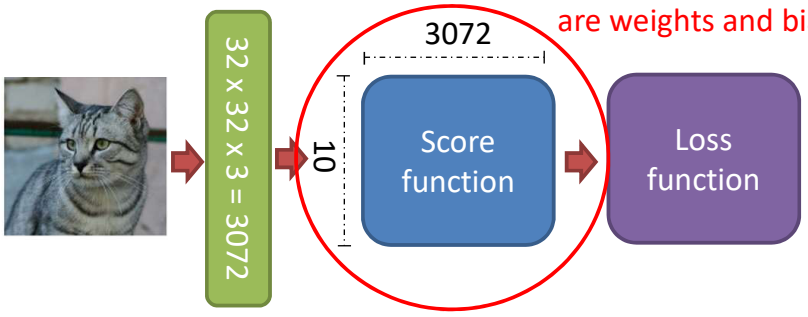
Low Loss

Case2: -(1 × log(0.87) + 0 + 0) = 0.060

Optimization

- **Minimize Loss:**
  - A way to adjust the parameters of score function
  - Find a minimum loss is object
  - High dimensional loss surface

All params you can adjust are weights and biases

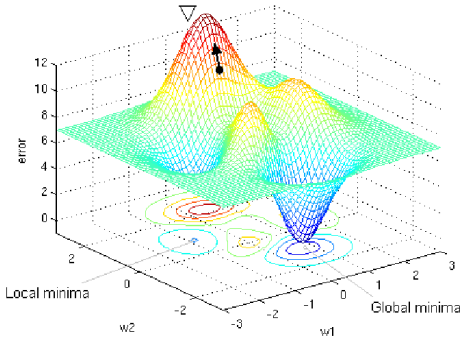


Can't understand high dimension (30720) space !!

Optimization

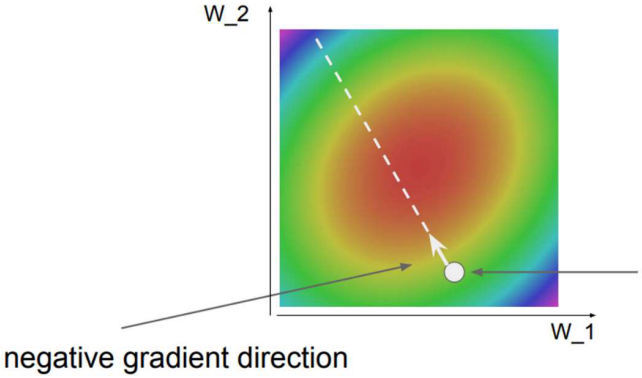
Also complex if slicing in two dimensions !

Like These:

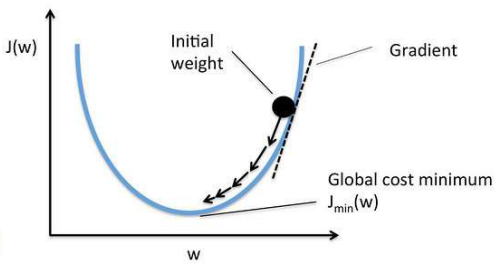


Optimization

- Strategy #1: Random search (bad !)
- Strategy #2: Random Local Search
- Strategy #3: Following the Gradient (good !)

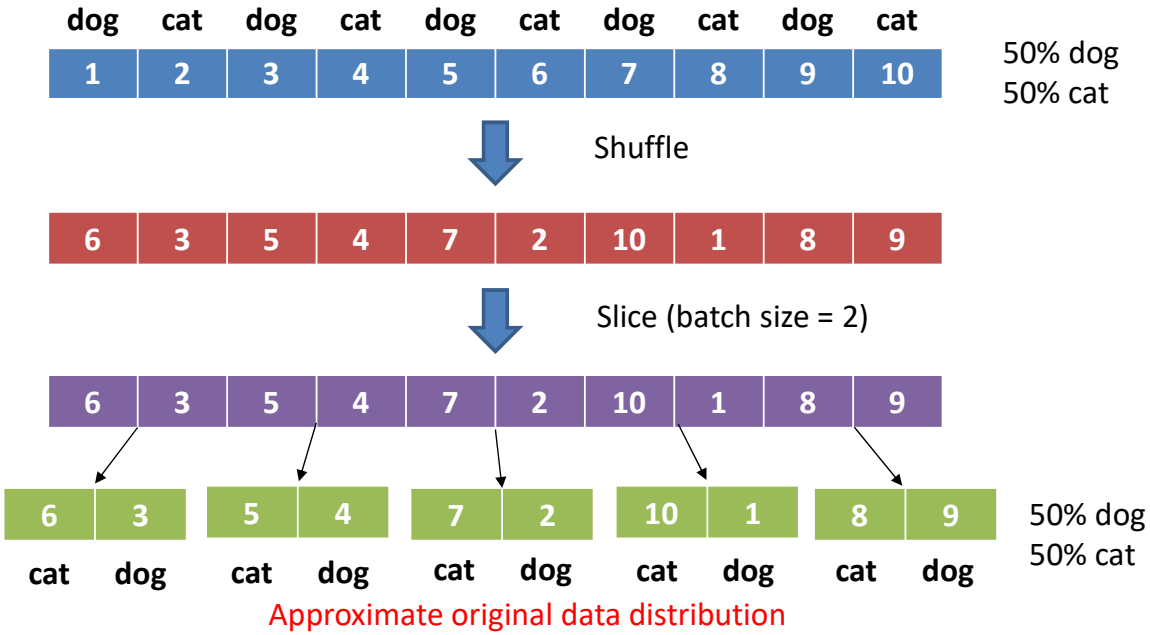


origi



Optimization

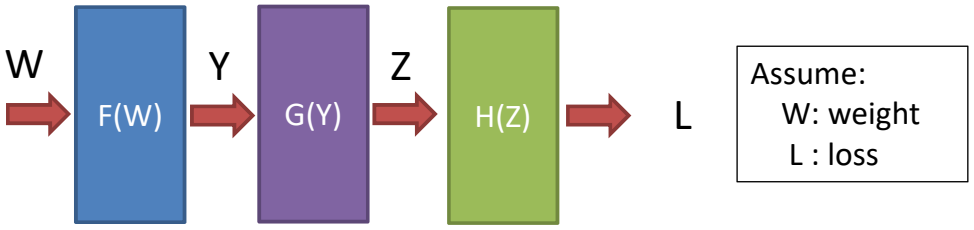
- Mini-batch gradient descent



- SGD (stochastic gradient descent) batch size= 1

Optimization

- How to compute gradient
  - Back - propagation



How to get $\frac{\partial L}{\partial W}$?

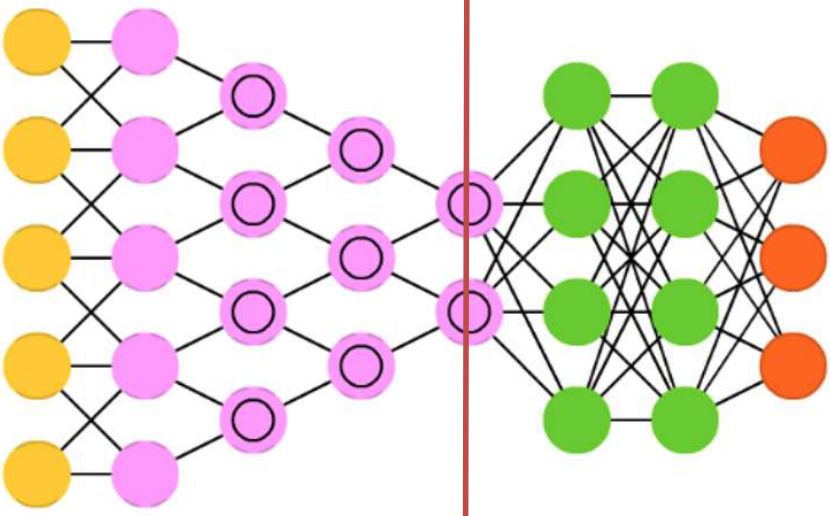
Chain-rule: $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Z} \times \frac{\partial Z}{\partial W} = \frac{\partial L}{\partial Z} \times \frac{\partial Z}{\partial Y} \times \frac{\partial Y}{\partial W}$

$H'(Z)$ $G'(Y)$ $F'(W)$

CNN architecture

Feature Extractor
Convolution layers

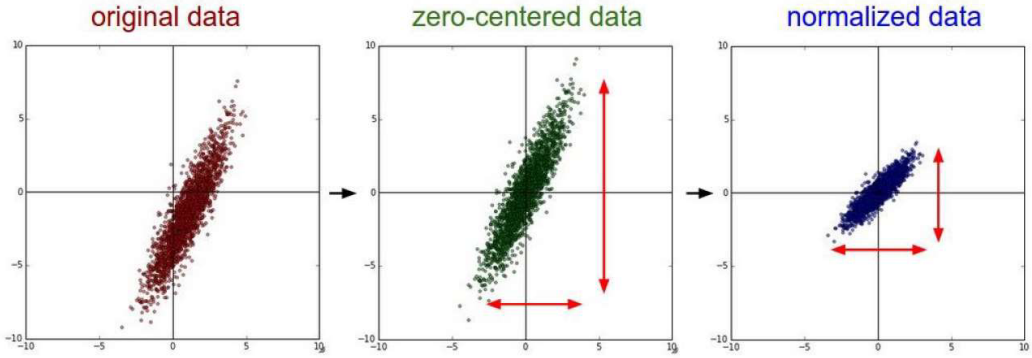
Classifier
Fully Connect layers



Sparse !

Normalization

Step 1: Preprocess the data




```
X -= np.mean(X, axis = 0)
```

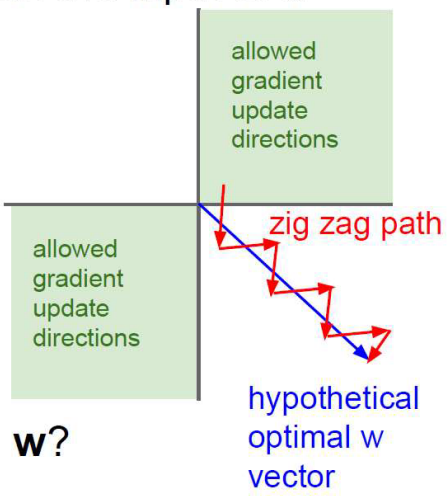
```
X /= np.std(X, axis = 0)
```

(Assume X [NxD] is data matrix,
each example in a row)

Zero mean

Remember: Consider what happens when the input to a neuron is always positive...

$$f\left(\sum_i w_i x_i + b\right)$$



What can we say about the gradients on **w**?
Always all positive or all negative :(
(this is also why you want zero-mean data!)

Prepare your dataset

Setting Hyperparameters

Idea #1: Choose hyperparameters that work best on the data

BAD: K = 1 always works perfectly on training data

Idea #2: Split data into **train** and **test**, choose hyperparameters that work best on test data

BAD: No idea how algorithm will perform on new data

Idea #3: Split data into **train**, **val**, and **test**; choose hyperparameters on val and evaluate on test

Better!

More keywords

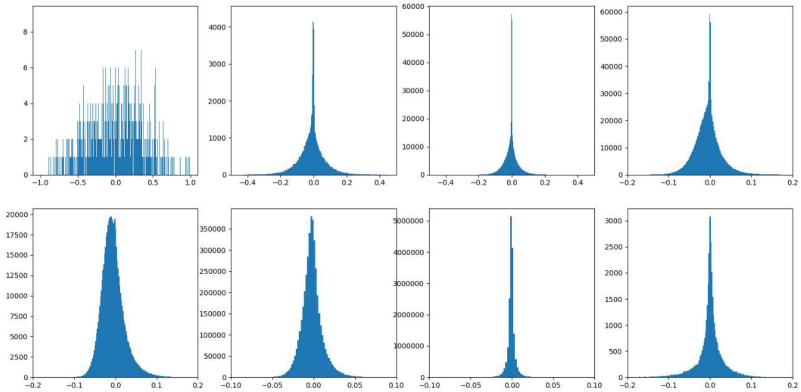
- Learning rate
- Learning rate decay schedule
- Epoch

Introduce Pytorch framework

- Dynamic computation graph
- Supply many common model, common dataset
- Tensor
  - Tensor type (CPU vs CUDA)
  - Tensor flag – grad_fn , requires_grad
  - How to compute gradient

LAB 1-1
Train a CNN model via Pytorch

- We already supply a example
- Just need to make up the test part
- You should show us the weight distribution (each layer)
- You should show total loss and accuracy (test case)
- You should show accuracy for each class



LAB 1-2
Train a new dataset

- You should show total loss and accuracy (test case)
- You should show what model you use

LAB 1-3
Tuning hyper-parameters

- Learning rate
- Learning rate decay method
- Show your experiment and result
  - How to decide your initial learning rate
  - Or, How to decide your decay method

Notice

- NCHC container : (use jupyter notebook to complete lab)
  - https://aitrain.nchc.org.tw
  - Use matplotlib to show the chart
  - don't need to download the cifar10 & Food11 dataset
- The cifar10 & Food-11 dataset is already in /tmp/datasets-nctu (only-read)
  - Food-11 download link : https://mmspg.epfl.ch/food-image-datasets
- The R/W directory : /tmp/work
  - If need to change the directory tree of Food-11 (LAB 1-2), please copy to this directory
  - Ex: /tmp/work/Food-11_{student id}
- (LAB 1-2 hint) ImageFolder : https://pytorch.org/tutorials/beginner/data_loading_tutorial.html#afterword-torchvision
- (LAB 1-2 hint) Command : ls /tmp/datasets-nctu/Food-11/training/7_*.jpg

Grading

- LAB 1-1 (60%)
- LAB 1-2 (35%)
- LAB 1-3 (5%)
- Bonus (5%)
  - In LAB 1-2, do normalize according the mean and standard deviation of Food-11 dataset

Total: 105

- Submission: source code + report (E3)
  - zip format (ex: dllab_{student id}.zip)
- Deadline : 2018/9/30, 23:59
- (暫) Demo : 2018/10/1, 1G

Report Spec.

- EX:
  - Problems & solutions
  - Experiment setup
  - Results
  - Analysis

Reference

- Stanford Course : CS231n
  - http://cs231n.stanford.edu/
- Pytorch Document:
  - https://pytorch.org/docs/stable/index.html