

Assignment 3

Sang-gil Lee

Electrical and Computer Engineering
Seoul National University

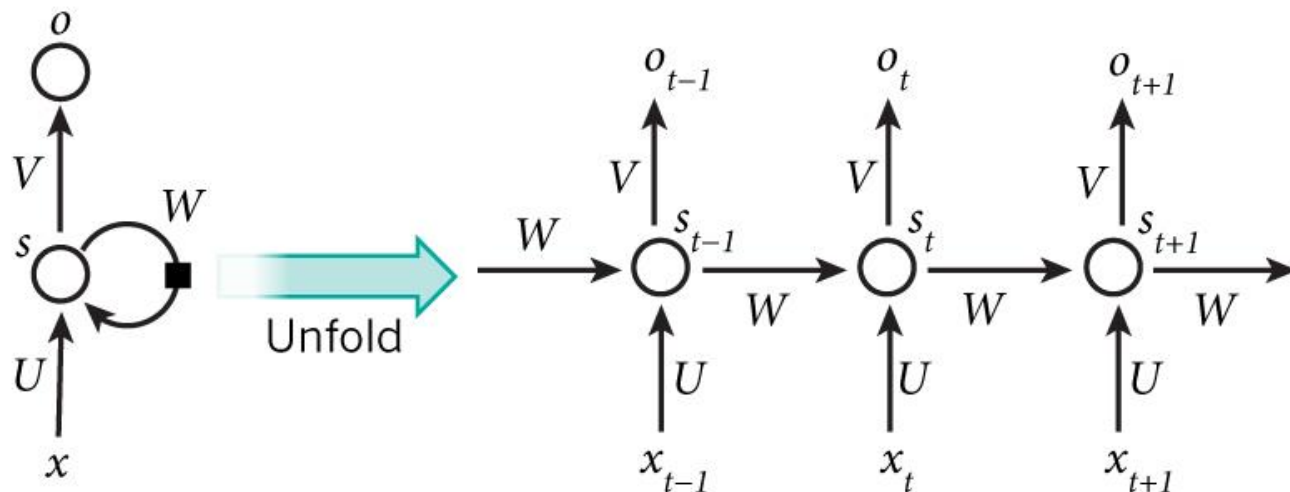
<http://data.snu.ac.kr>

Assignment Objectives

- Part 1: Implementing RNN
 - To understand RNN architecture before using TensorFlow
 - Implement forward/backward of
 - ✓ Single timestep
 - $\tanh'(x) = 1 - \tanh(x)^2$
 - ✓ Entire sequence based on single timestep
- Part 2: Image Captioning
 - Design RNN model for image captioning with TensorFlow
 - Explore various RNN structure and hyperparameters
- Part 3: Language Modeling
 - Learn probability distribution of characters from our language

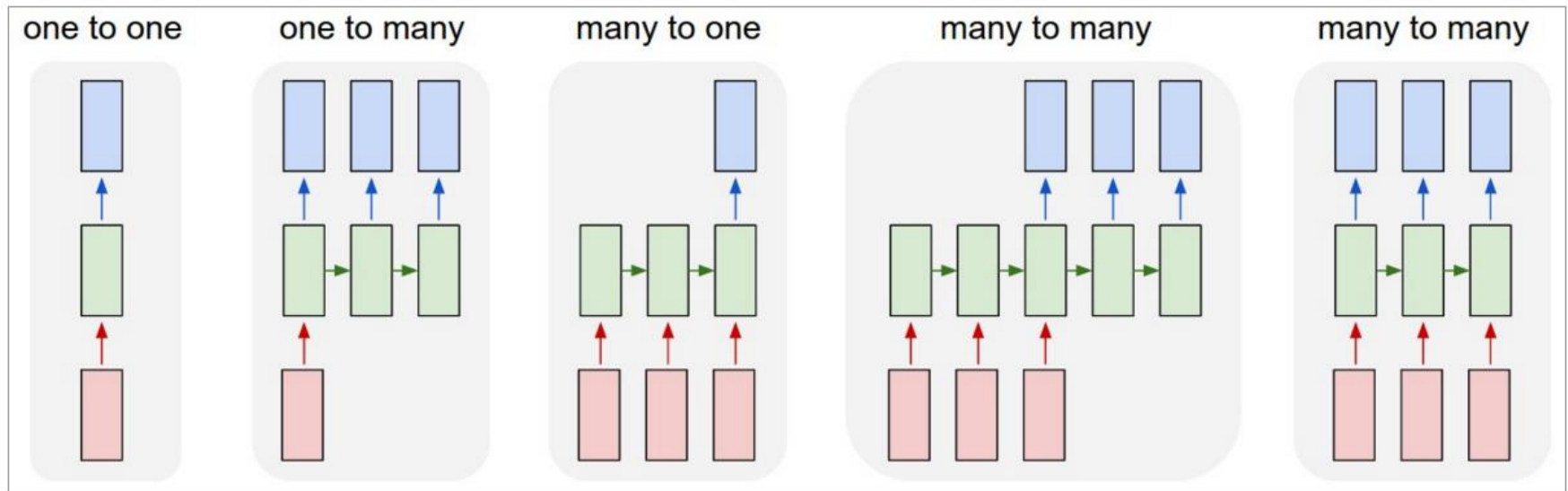
Recurrent Neural Networks

- RNN (Recurrent Neural Networks)



- RNN perform the same task for every element of a sequence
- Output depending on the previous computations, “memory”
- $s_t = f(Ux_t + Ws_{t-1})$, $o_t = \text{softmax}(Vs_t)$
 - ✓ f is nonlinearity function such as tanh
 - ✓ The same parameters (U, V, W)

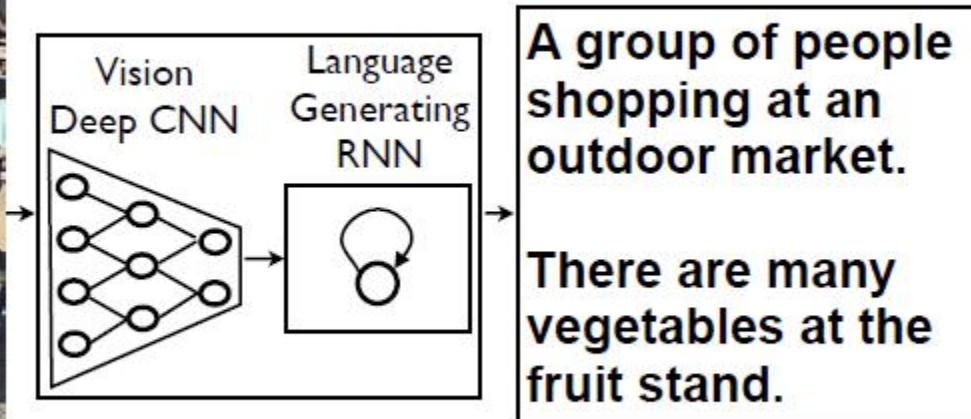
Recurrent Neural Networks



Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state (more on this soon). From left to right: **(1)** Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). **(2)** Sequence output (e.g. image captioning takes an image and outputs a sentence of words). **(3)** Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). **(4)** Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French). **(5)** Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case there are no pre-specified constraints on the lengths of sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

Part 2: Image Captioning

- Generating natural sentences describing an image
 - Connecting computer vision and natural language processing
 - A vision CNN followed by a language generating RNN

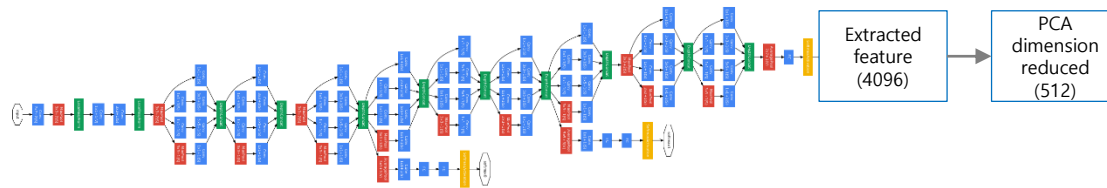


Part 2: Image Captioning

- Implement image captioning with TensorFlow RNN modules

- Input: extracted image feature from the VGG-16 network

- ✓ 4096 → 512



- Output: predicted captions

- Data: run the `./coco/get_coco_data.sh`

- Model: save your model in `model_path`

- Evaluation: BLEU score for validation and independent TA's test dataset

- ✓ Test dataset is not provided

- ✓ 주의사항

- 테스트는 저장된 모델을 로드해서 평가하므로 반드시 저장 및 로드 상태 확인

Microsoft COCO dataset



A man with pierced ears is wearing glasses and an orange hat.
A man with glasses is wearing a beer can crocheted hat.
A man with gauges and glasses is wearing a Blitz hat.
A man in an orange hat starring at something.
A man wears an orange hat and glasses.

- Microsoft coco dataset
 - Five independent human generated captions per image
- Dataset

```
coco_captioning/coco2014_vocab.json
coco_captioning/train2014_captions.h5
coco_captioning/train2014_images.txt
coco_captioning/train2014_urls.txt
coco_captioning/train2014_vgg16_fc7.h5
coco_captioning/train2014_vgg16_fc7_pca.h5
```

 - # train captions : 400,135
 - # validation captions: 10,000
- 주의사항
 - Full train 소요시간이 상당하므로 train 데이터의 일부 sample로 시험, 최종 제출시에 full train

```
train_data = load_coco_data(max_train=500)
```
 - 512 extracted image feature 사용할것
 - maxlen, n_words, input_dimension 변경하지 말것

Preprocessing of language model

- Preprocessing of training data of language model
 - Tokenize
 - Remove infrequent words
 - ✓ Limit vocabulary size to the most common words
 - ✓ <UNKNOWN>
 - Prepend special start and end token
 - ✓ <START> and <END> tokens
 - Word to index
 - Word to numeric vectors (Word2Vec)

<one-hot>

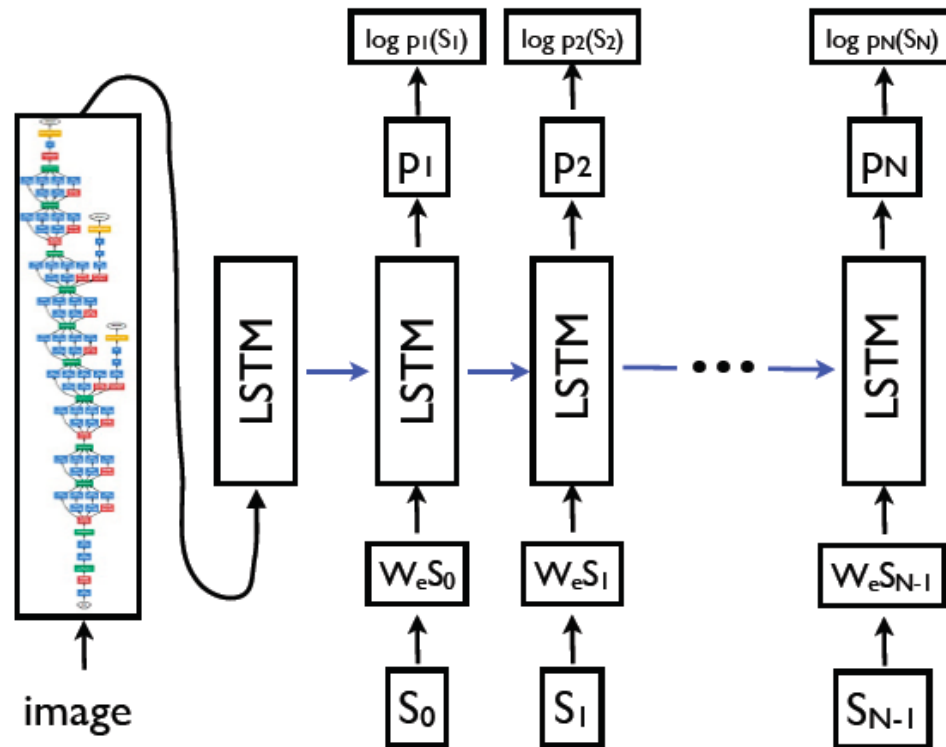
$$\begin{pmatrix} the \\ cat \\ sat \\ on \\ the \\ mat \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

<embedding>

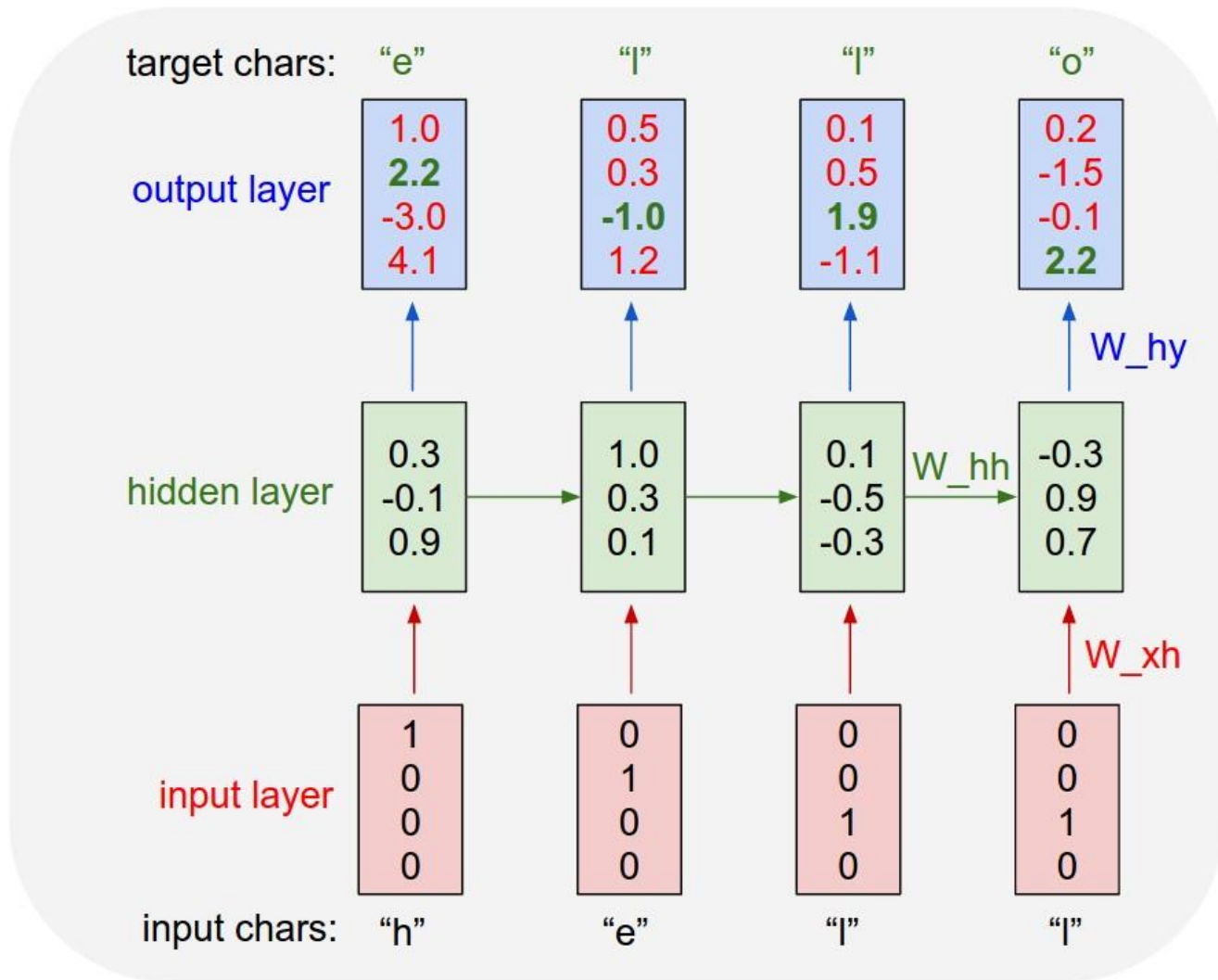
<i>anarchism</i>	0.5	0.1	-0.1
<i>originated</i>	-0.5	0.3	0.9
<i>as</i>	0.3	-0.5	-0.3
<i>a</i>	0.7	0.2	-0.3
<i>term</i>	0.8	0.1	-0.1
<i>of</i>	0.4	-0.6	-0.1
<i>abuse</i>	0.7	0.1	-0.4

Example of image captioning

- Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.



Character-level Language Modeling



Tinyshakespeare dataset

First Citizen:

Before we proceed any further, hear me speak.

All:

Speak, speak.

First Citizen:

You are all resolved rather to die than to famish?

All:

Resolved. resolved.

First Citizen:

First, you know Caius Marcius is chief enemy to the people.

All:

We know't, we know't.

First Citizen:

Let us kill him, and we'll have corn at our own price.

Is't a verdict?

All:

No more talking on't; let it be done: away, away!

Second Citizen:

One word, good citizens.

First Citizen:

We are accounted poor citizens, the patricians good.

What authority surfeits on would relieve us: if they

would yield us but the superfluity, while it were

wholesome, we might guess they relieved us humanely;

but they think we are too dear: the leanness that

afflicts us, the object of our misery, is as an

inventory to particularise their abundance; our

sufferance is a gain to them Let us revenge this with

our pikes, ere we become rakes: for the gods know I

speak this in hunger for bread, not in thirst for revenge.

Second Citizen:

Would you proceed especially against Caius Marcius?

All:

Against him first: he's a very dog to the commonalty.

Second Citizen:

Consider you what services he has done for his country?

Assignment 3

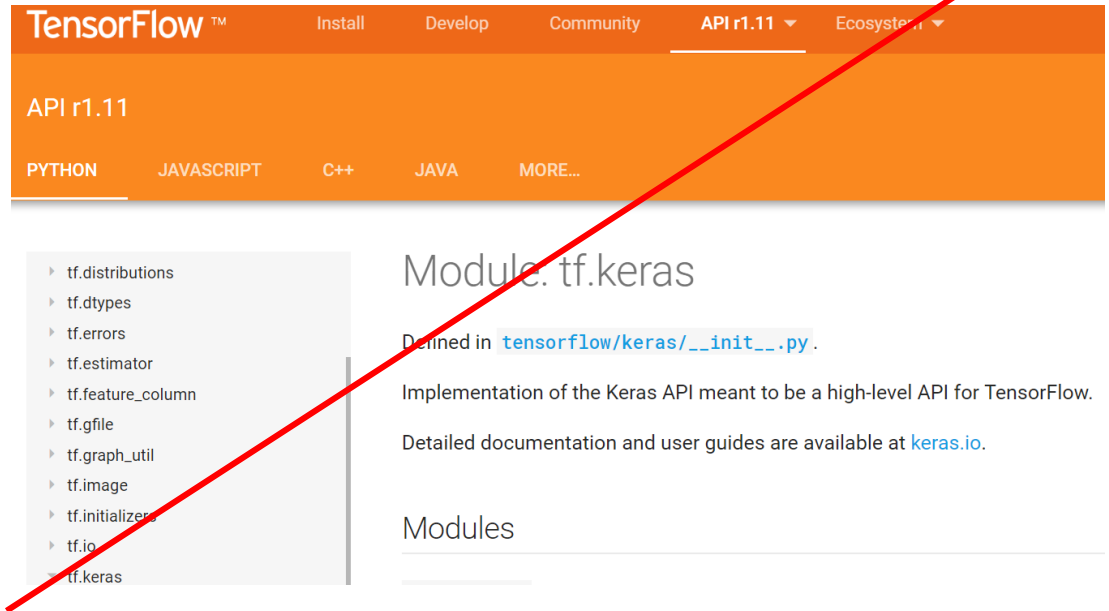
- 구현 및 평가 주의사항
 - 구현
 - ✓ Part1: rnn_layer.py
 - ✓ Part2: captioning.py, ipython 파일 (train, prediction, 구현내용 설명)
 - maxlen (17), n_words (1004), input_dimension(512) 변경 불가
 - coco_utils.py 변경불가
 - 구조, 함수, rnn 종류, 다른 hyperparameter 등 자유롭게 변경
 - ✓ Part 3: char_rnn.py, ipython 파일 (train, sampling, 구현내용 설명)
 - utils.py 변경불가
 - 구조, 함수, rnn 종류, 다른 hyperparameter 등 자유롭게 변경
 - 제출 모델 세이브 파일 (checkpoint)는 하나만! (많으면 용량이 너무 커짐)
 - ✓ Saver() 에서 max_to_keep 파라미터 활용

Assignment 3

- 구현 및 평가 주의사항

- 구현

tf.keras API 사용 금지



Keras: TensorFlow를 좀더 쉽게 쓰도록 도와주는 wrapper

해당 API를 쓰면 RNN을 1-line으로 구현 가능

그러나 공부 목적을 위해 해당 API는 사용할 수 없음

Assignment 3

- 구현 및 평가 주의사항

- 평가

- ✓ Part1 (20 point), Part2 (40 point), Part3 (40 point)

- ✓ Ipython 파일 로그 저장

- Part2 & Part 3

- » Train 과정에서 loss 변화를 최소 10개 정도 출력 할 것 (감소 상태 → 평가)

- 전체 epoch가 100이면 약 10구간마다 loss 변화 출력

- » 모델에 대한 설명: 미흡할 경우 감점요소

- Part 2

- » 독립 test data를 이용해서 저장된 모델 로드해서 평가

- 반드시 모델의 저장 및 로드 상태 확인

- .ipynb 파일의 초기화 뒤 Evaluation 부터 run 하더라도 돌아가야 함

- 약 200,000개의 captions을 가진 test_data로 평가

- » "타겟 성능?": validation set BLEU score **0.3 이상이면** full score 가능 (mostly)

- Part 3

- » Training & 샘플링 결과로 on/off 평가

- 반드시 모델의 저장 및 로드 상태 확인

- .ipynb 파일의 초기화 뒤 Evaluation 부터 run 하더라도 돌아가야 함

- 샘플링된 문장은 "명백히 이상하지만 않으면" on 처리

How to install assignment files

- Assignment files
 - **Models_captioning/ & models_char_rnn/** (모델 체크포인트가 저장될 폴더)
 - `Coco/ get_coco_data.sh`
 - **Coco_utils.py**
 - **Utils.py**
 - **Rnn_layers.py**
 - **Captioning.py**
 - **Char_rnn.py**
 - **Assignment3_Part1_Implementing_RNN.ipynb**
 - **Assignment3_Part2_ImageCaptioning.ipynb**
 - **Assignment3_Part3_CharRNN.ipynb**
 - `CollectSubmission.sh`
- 채점은 각 part의 .ipynb 파일로 수행할 예정: 제출 시 빠진 내용 및 파일이 없는지 반드시 체크
- Install assignment files
 - `tar -zxvf assignment3.tar.gz`
 - `sudo chmod 755 CollectSubmission.sh`
 - `./coco/get_coco_data.sh`
 - `jupyter notebook`
- Open the notebooks on your browser and get started

Important Notes

- PLEASE read the notes on the notebooks carefully
- Google first before mailing TAs
- Some details are missing, ambiguous, or even wrong on purpose
- Submitting your work
 - DO NOT clear the final outputs
 - After you are done **all three parts**
 - ✓ `$./CollectSubmission.sh team_#`
 - ✓ Upload the `team_#.tar.gz` on ETL
 - ✓ Ex: `team_1.tar.gz`
- TA email: `deeplearning.snu@gmail.com`

Thank You!

