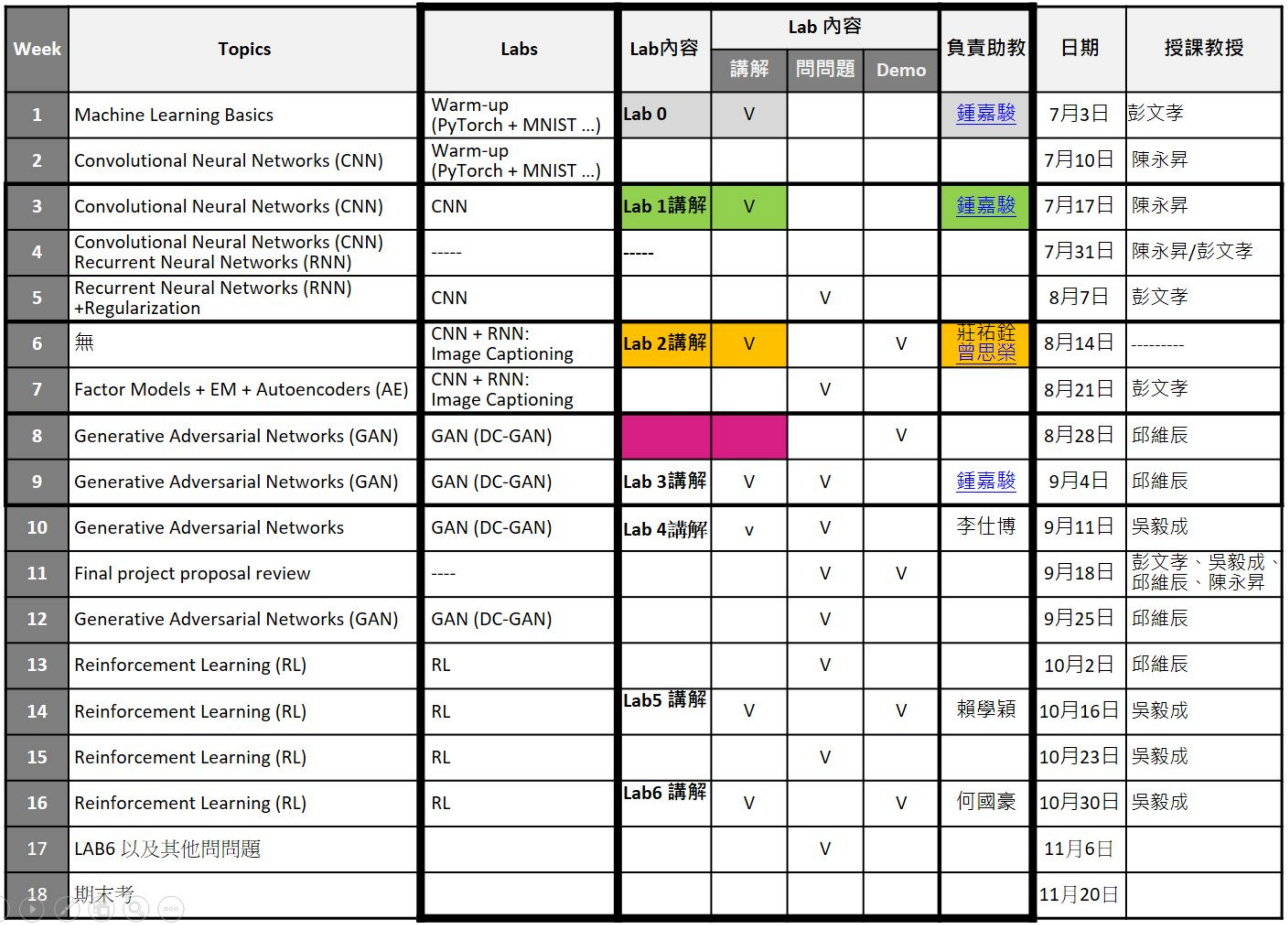
Lab3 : a LSTM Cell for Image Captioning

Department of Computer Science, NCTU

TA Ziv(鍾嘉峻)



Week	Topics	Labs	Lab內容	Lab 內容			負責助教	日期	授課教授
				講解	問問題	Demo			
1	Machine Learning Basics	Warm-up (PyTorch + MNIST ...)	Lab 0	V			鍾嘉駿	7月3日	彭文孝
2	Convolutional Neural Networks (CNN)	Warm-up (PyTorch + MNIST ...)						7月10日	陳永昇
3	Convolutional Neural Networks (CNN)	CNN	Lab 1講解	V			鍾嘉駿	7月17日	陳永昇
4	Convolutional Neural Networks (CNN) Recurrent Neural Networks (RNN)	-----	-----					7月31日	陳永昇/彭文孝
5	Recurrent Neural Networks (RNN) +Regularization	CNN			V			8月7日	彭文孝
6	無	CNN + RNN: Image Captioning	Lab 2講解	V		V	莊祐銓 曾思棨	8月14日	---------
7	Factor Models + EM + Autoencoders (AE)	CNN + RNN: Image Captioning			V			8月21日	彭文孝
8	Generative Adversarial Networks (GAN)	GAN (DC-GAN)	🟪	🟪		V		8月28日	邱維辰
9	Generative Adversarial Networks (GAN)	GAN (DC-GAN)	Lab 3講解	V	V		鍾嘉駿	9月4日	邱維辰
10	Generative Adversarial Networks	GAN (DC-GAN)	Lab 4講解	v	V		李仕博	9月11日	吳毅成
11	Final project proposal review	----			V	V		9月18日	彭文孝、吳毅成、邱維辰、陳永昇
12	Generative Adversarial Networks (GAN)	GAN (DC-GAN)			V			9月25日	邱維辰
13	Reinforcement Learning (RL)	RL			V			10月2日	邱維辰
14	Reinforcement Learning (RL)	RL	Lab5 講解	V		V	賴學穎	10月16日	吳毅成
15	Reinforcement Learning (RL)	RL			V			10月23日	吳毅成
16	Reinforcement Learning (RL)	RL	Lab6 講解	V		V	何國豪	10月30日	吳毅成
17	LAB6 以及其他問問題				V			11月6日	
18	期末考							11月20日	

Important Rules

- Important Date :
  - Report Submission Deadline: 10/16 (Wed) 11:59 AM
  - Demo date: 10/16 (Wed)

- Turn in :
  - Experiment Report (.pdf)
  - Source code (.py)

- Notice: zip all files in one file and name it like「DLP_LAB3_your studentID_name.zip」, ex: 「DLP_LAB3_0756172_鍾嘉峻.zip」

Important Rules

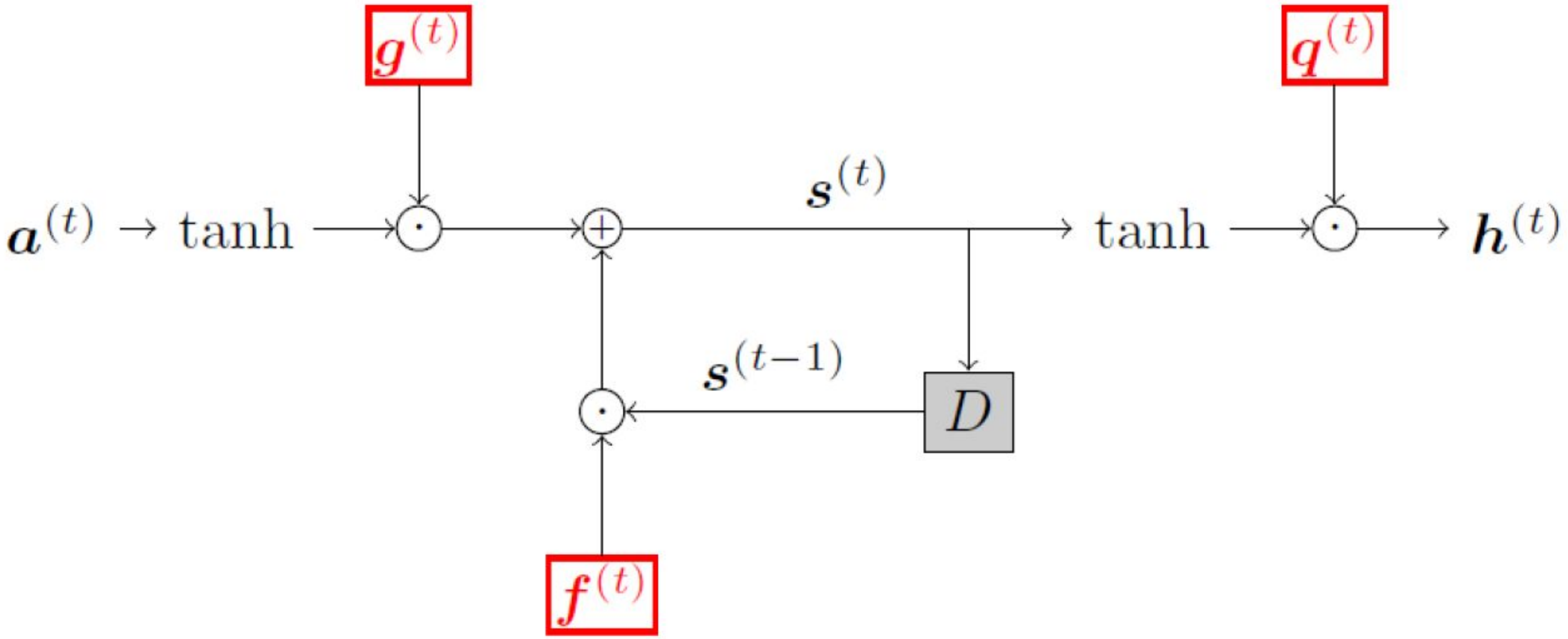
- Email To :
  - zivzhong.cs07g@nctu.edu.tw
  - Don’t CC other TA

- Email Tilte :
  - DLP_LAB3_your studentID_name

- Do not submmit your weight or dataset!!
  - But you should save the model weight for demo

Lab Objective

- In this lab, you have to implement a LSTM cell by yourselves
- And train an image caption model with your own LSTM cell



Lab Requierment

- Implement a LSTM cell
  - Please finish
  - Only Forward part , don’t worried

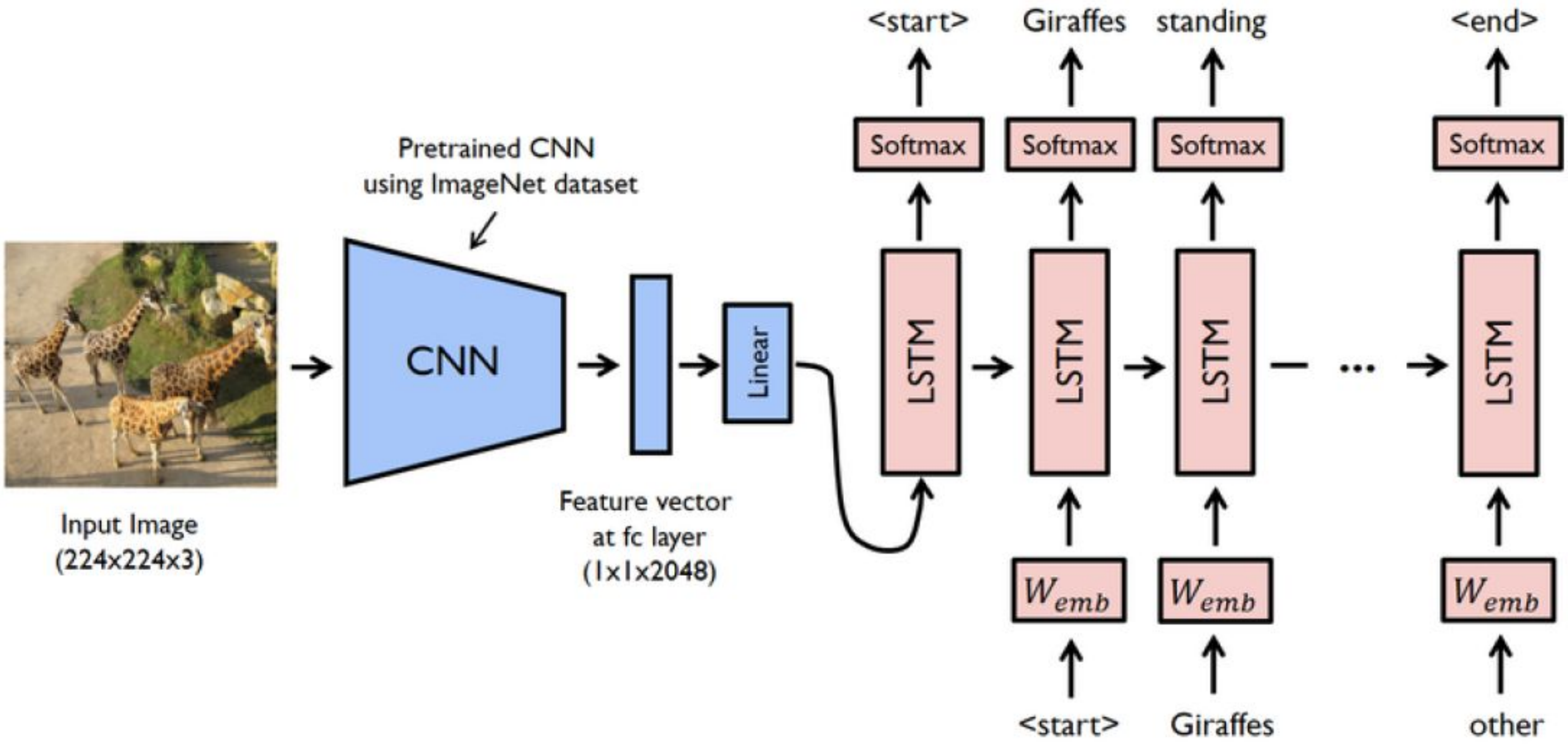
- Replace the LSTM cell in Pytorch image-caption example withthe one you implement.

- Train an image caption model

Lab Resource & Instruction

- Clone https://github.com/2019-dl-training-program/Lab3.git
  - Already on the sever
  - Please follow the Usage in Readme if you want to try on you own machine
- Get the data
  - Already on the sever
- Implement a LSTM cell (DIY_LSTM.py)
- Train the model
- You only need to start at “3. Preprocessing” in the Readme

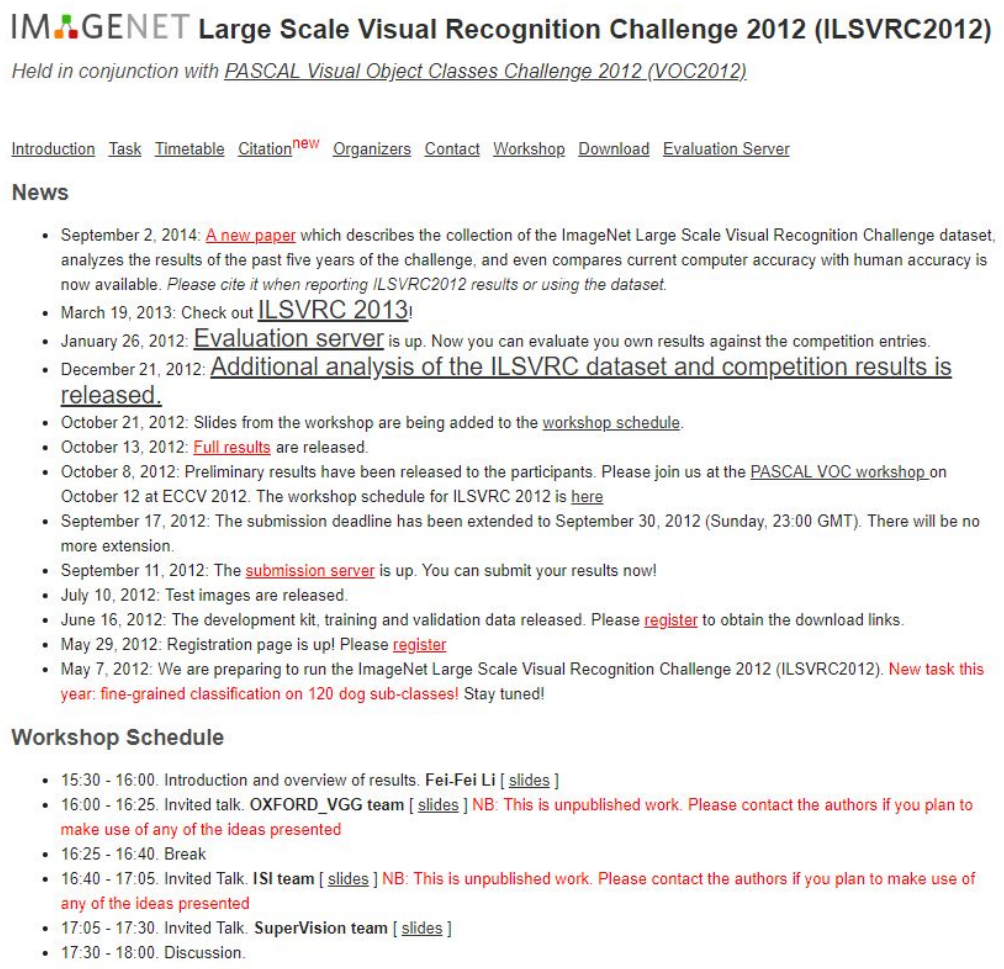
Image-Caption



Dataset

- ImageNet : Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)
- Very good dataset for
  - Classfication
  - localization
  - …..

- Dataset for this lab
  - Image
  - Annotation



IMAGENET Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)

Introduction Task Timetable Citation new Organizers Contact Workshop Download Evaluation Server

News

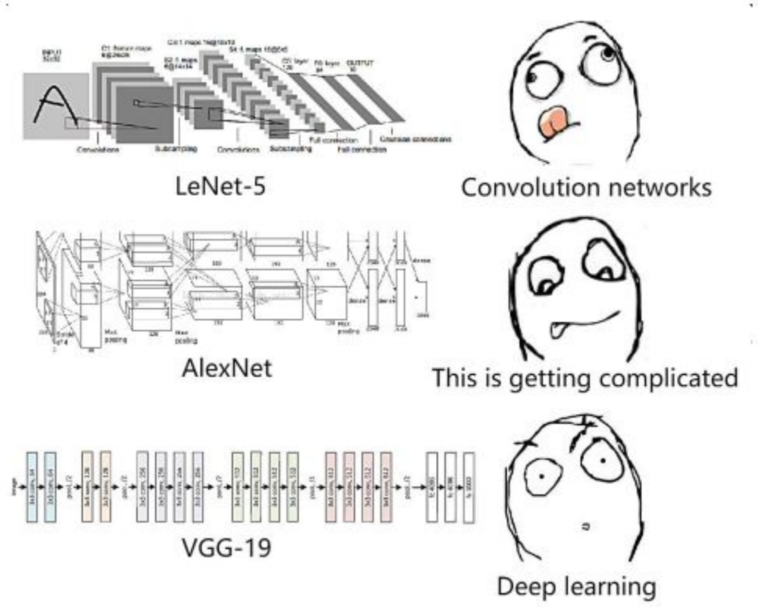
- September 2, 2014: A new paper which describes the collection of the ImageNet Large Scale Visual Recognition Challenge dataset, analyzes the results of the past five years of the challenge, and even compares current computer accuracy with human accuracy is now available. *Please cite it when reporting ILSVRC2012 results or using the dataset.*
- March 19, 2013: Check out ILSVRC 2013!
- January 26, 2012: Evaluation server is up. Now you can evaluate you own results against the competition entries.
- December 21, 2012: Additional analysis of the ILSVRC dataset and competition results is released.
- October 21, 2012: Slides from the workshop are being added to the workshop schedule.
- October 13, 2012: Full results are released.
- October 8, 2012: Preliminary results have been released to the participants. Please join us at the PASCAL VOC workshop on October 12 at ECCV 2012. The workshop schedule for ILSVRC 2012 is here
- September 17, 2012: The submission deadline has been extended to September 30, 2012 (Sunday, 23:00 GMT). There will be no more extension.
- September 11, 2012: The submission server is up. You can submit your results now!
- July 10, 2012: Test images are released.
- June 16, 2012: The development kit, training and validation data released. Please register to obtain the download links.
- May 29, 2012: Registration page is up! Please register
- May 7, 2012: We are preparing to run the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). New task this year: fine-grained classification on 120 dog sub-classes! Stay tuned!

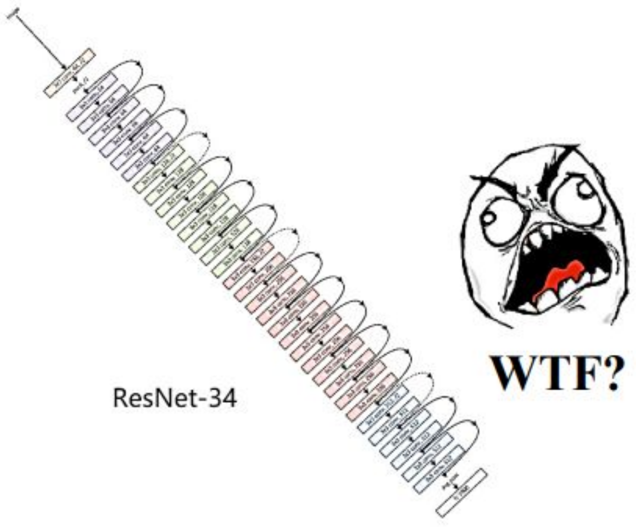
Workshop Schedule

- 15:30 - 16:00. Introduction and overview of results. **Fei-Fei Li** [slides]
- 16:00 - 16:25. Invited talk. **OXFORD_VGG team** [slides] NB: This is unpublished work. Please contact the authors if you plan to make use of any of the ideas presented
- 16:25 - 16:40. Break
- 16:40 - 17:05. Invited Talk. **ISI team** [slides] NB: This is unpublished work. Please contact the authors if you plan to make use of any of the ideas presented
- 17:05 - 17:30. Invited Talk. **SuperVision team** [slides]
- 17:30 - 18:00. Discussion.

Encoder

- Using pretrain models to extract feature vector from a given input image
  - Using pretrained ResNet-152
  - From Torchvision

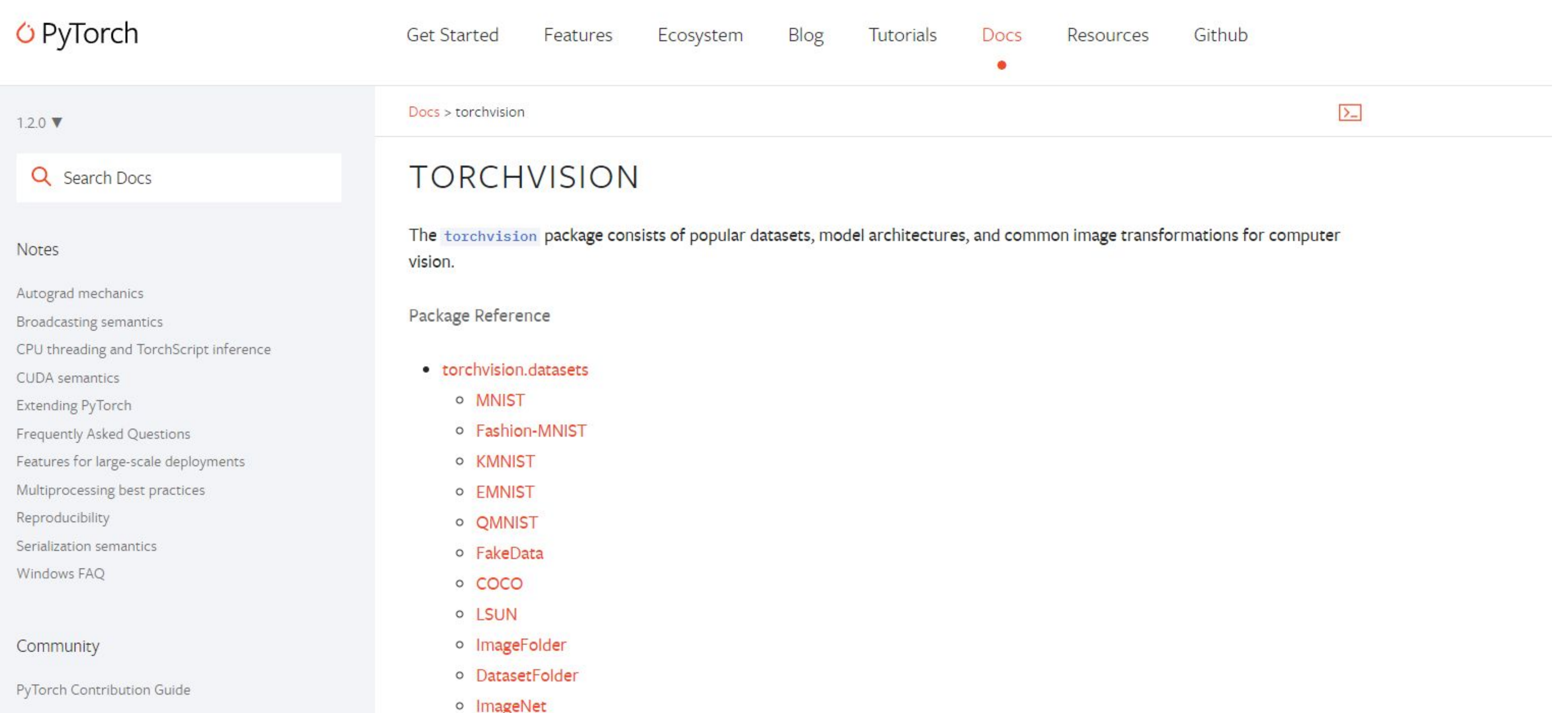




ResNet-152

Torchvision

- Pytorch official package consists of
  - popular datasets
  - model architectures
  - common image transformations for computer vision.



TORCHVISION

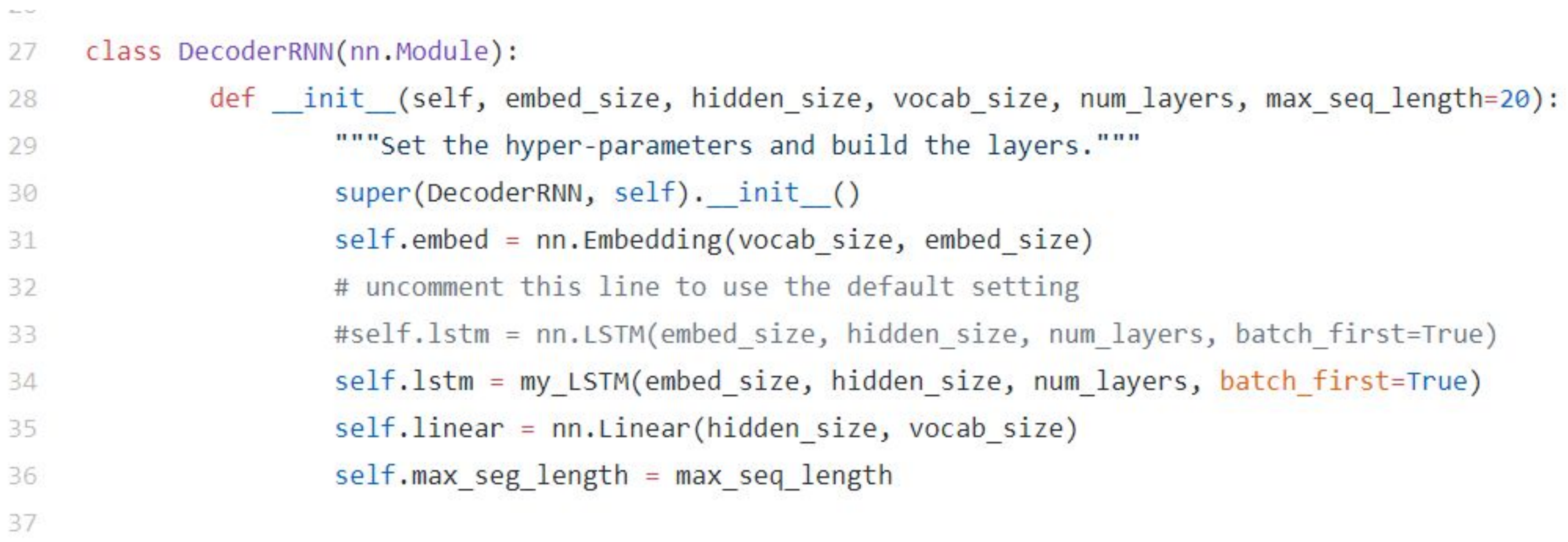
The `torchvision` package consists of popular datasets, model architectures, and common image transformations for computer vision.

Package Reference

- torchvision.datasets
  - MNIST
  - Fashion-MNIST
  - KMNIST
  - EMNIST
  - QMNIST
  - FakeData
  - COCO
  - LSUN
  - ImageFolder
  - DatasetFolder
  - ImageNet

Decoder

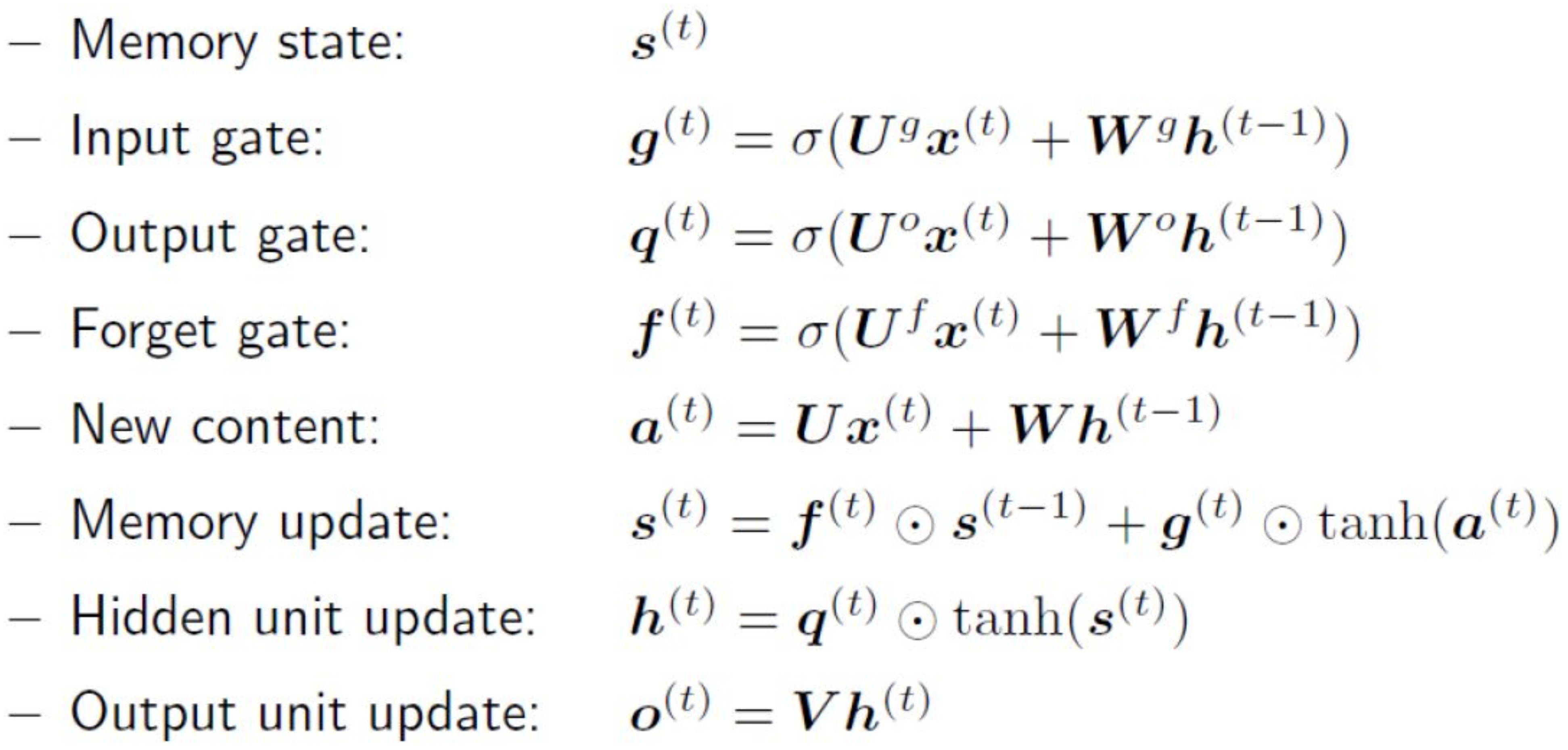
- Noticed that you will use your model in model.py line 34
  - You can use nn.LSTM to check your environment is OK or not



```
27  class DecoderRNN(nn.Module):
28          def __init__(self, embed_size, hidden_size, vocab_size, num_layers, max_seq_length=20):
29                  """Set the hyper-parameters and build the layers."""
30                  super(DecoderRNN, self).__init__()
31                  self.embed = nn.Embedding(vocab_size, embed_size)
32                  # uncomment this line to use the default setting
33                  #self.lstm = nn.LSTM(embed_size, hidden_size, num_layers, batch_first=True)
34                  self.lstm = my_LSTM(embed_size, hidden_size, num_layers, batch_first=True)
35                  self.linear = nn.Linear(hidden_size, vocab_size)
36                  self.max_seg_length = max_seq_length
37
```

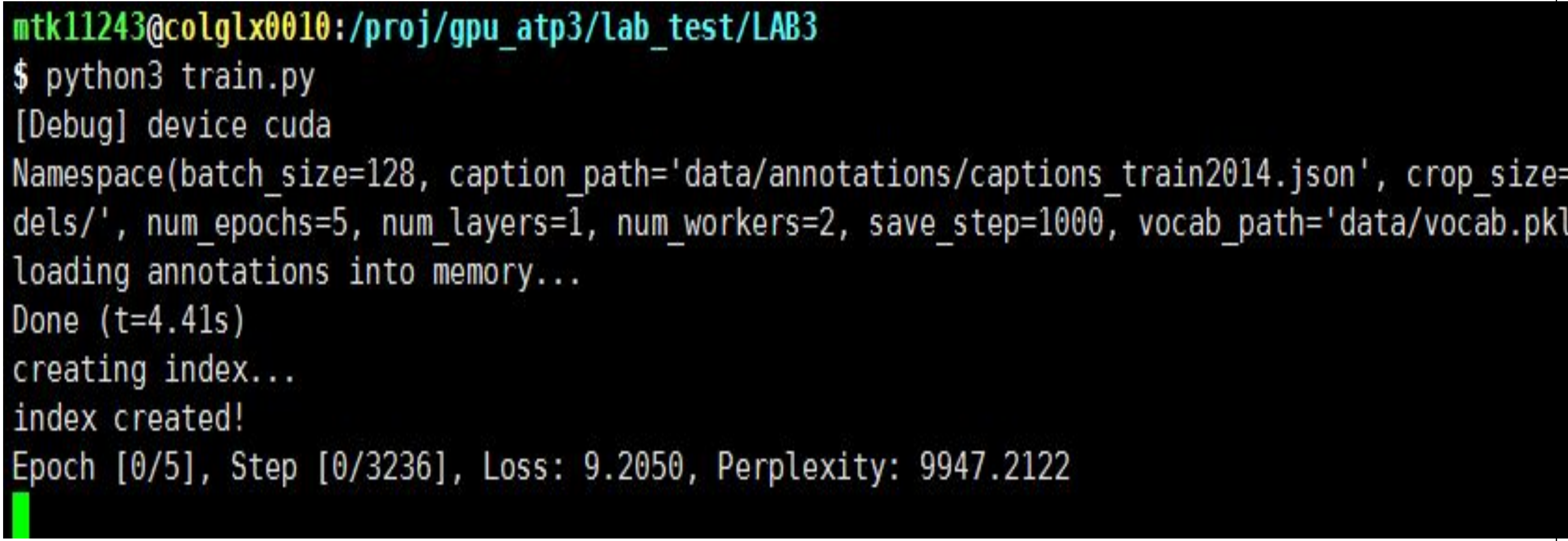
LSTM Recall

- At professor slide "RecurrentNeuralNetworks.pdf"



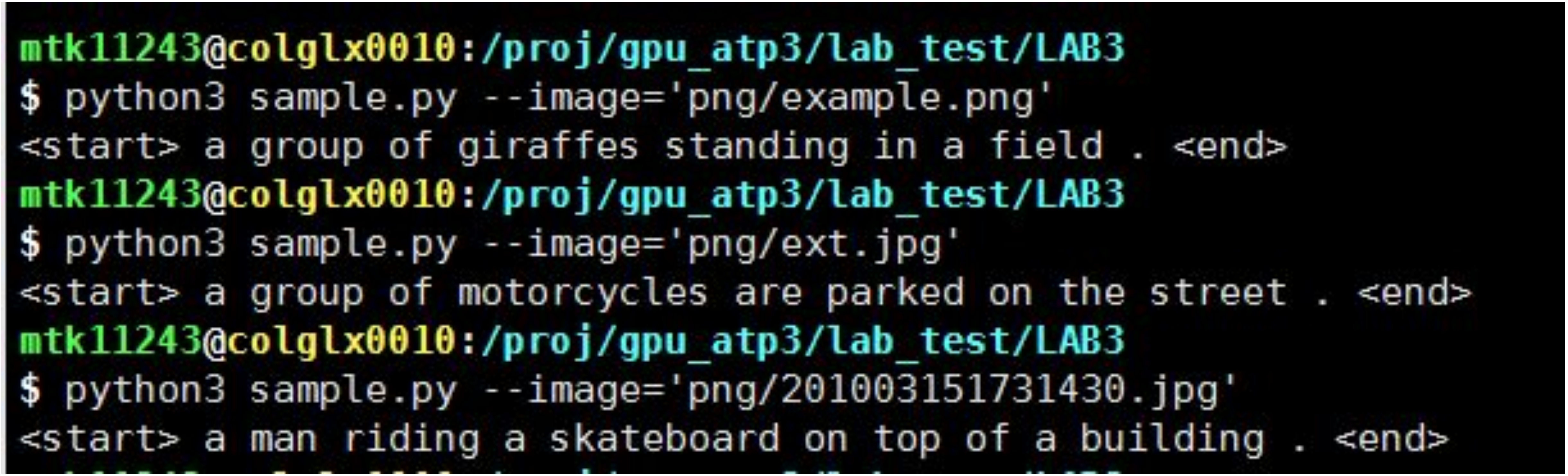
- Memory state: $\boldsymbol{s}^{(t)}$
- Input gate: $\boldsymbol{g}^{(t)} = \sigma(\boldsymbol{U}^g \boldsymbol{x}^{(t)} + \boldsymbol{W}^g \boldsymbol{h}^{(t-1)})$
- Output gate: $\boldsymbol{q}^{(t)} = \sigma(\boldsymbol{U}^o \boldsymbol{x}^{(t)} + \boldsymbol{W}^o \boldsymbol{h}^{(t-1)})$
- Forget gate: $\boldsymbol{f}^{(t)} = \sigma(\boldsymbol{U}^f \boldsymbol{x}^{(t)} + \boldsymbol{W}^f \boldsymbol{h}^{(t-1)})$
- New content: $\boldsymbol{a}^{(t)} = \boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)}$
- Memory update: $\boldsymbol{s}^{(t)} = \boldsymbol{f}^{(t)} \odot \boldsymbol{s}^{(t-1)} + \boldsymbol{g}^{(t)} \odot \tanh(\boldsymbol{a}^{(t)})$
- Hidden unit update: $\boldsymbol{h}^{(t)} = \boldsymbol{q}^{(t)} \odot \tanh(\boldsymbol{s}^{(t)})$
- Output unit update: $\boldsymbol{o}^{(t)} = \boldsymbol{V}\boldsymbol{h}^{(t)}$

Training



```
mtk11243@colglx0010:/proj/gpu_atp3/lab_test/LAB3
$ python3 train.py
[Debug] device cuda
Namespace(batch_size=128, caption_path='data/annotations/captions_train2014.json', crop_size=
dels/', num_epochs=5, num_layers=1, num_workers=2, save_step=1000, vocab_path='data/vocab.pkl
loading annotations into memory...
Done (t=4.41s)
creating index...
index created!
Epoch [0/5], Step [0/3236], Loss: 9.2050, Perplexity: 9947.2122
```

Testing



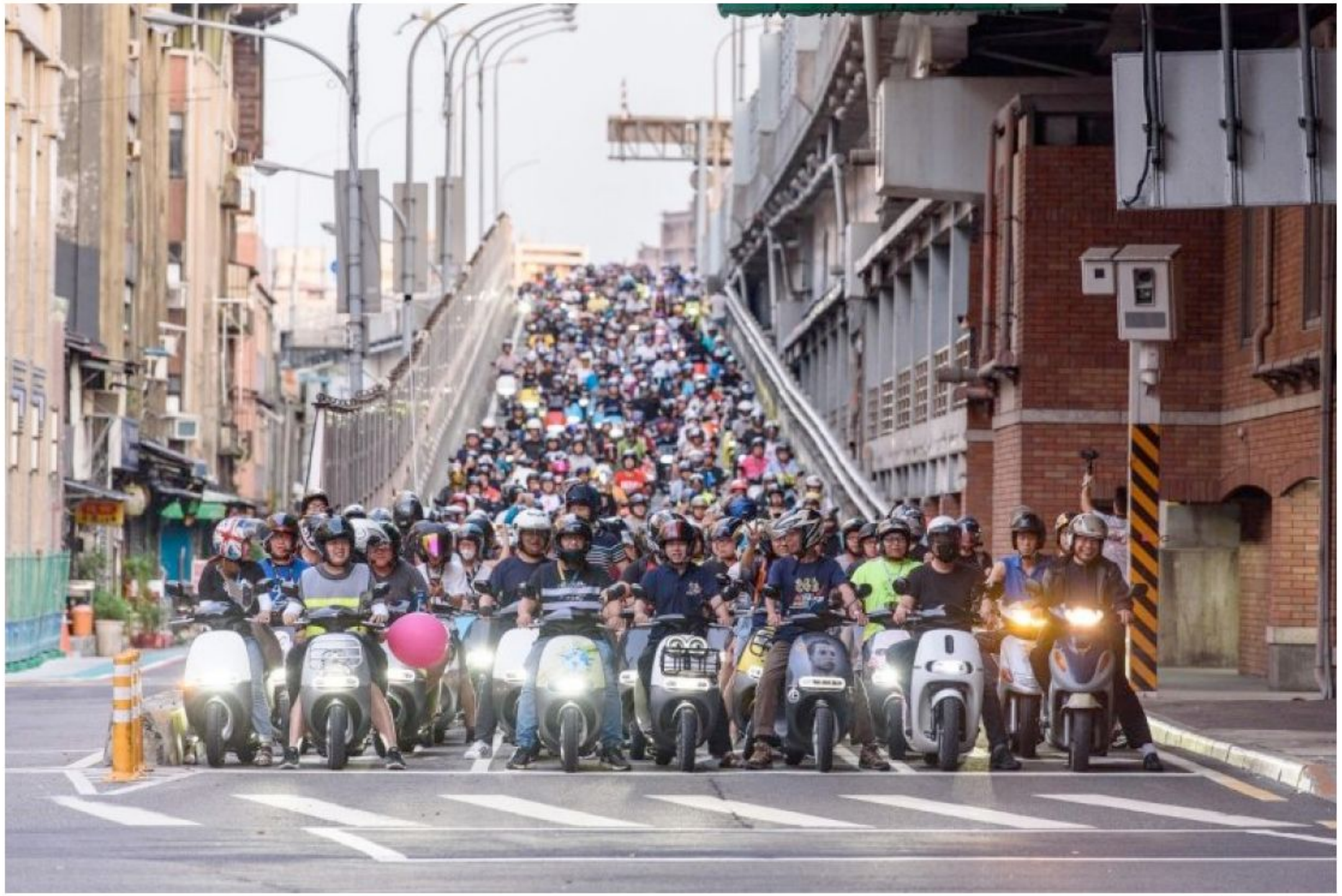
```
mtk11243@colglx0010:/proj/gpu_atp3/lab_test/LAB3
$ python3 sample.py --image='png/example.png'
<start> a group of giraffes standing in a field . <end>
mtk11243@colglx0010:/proj/gpu_atp3/lab_test/LAB3
$ python3 sample.py --image='png/ext.jpg'
<start> a group of motorcycles are parked on the street . <end>
mtk11243@colglx0010:/proj/gpu_atp3/lab_test/LAB3
$ python3 sample.py --image='png/20100315173l430.jpg'
<start> a man riding a skateboard on top of a building . <end>
```

Report Spec & Demo

- Introduction (5%)
- Explain how you implement LSTM (45%)
- Results – generating corresponding descriptions
  - A. example.png (10%)
  - B. ext.png (10%)
- Discussion (10%)
- Demo
  - Test your model on a given picture (10%)
  - Question (10%)

Demo example

- Like “Results” in Report
- The demo testing will be uploaded before Demo



<start> a group of people riding bikes down a street . <end>