Ghi chú của một coder

Vũ Anh

Tháng 01 năm 2018

Mục lục

Phần I

Lập trình

Chương 1

Python

1.1 Cơ bản

Vấn đề với mảng

Random Sampling [1] - sinh ra một mảng ngẫu nhiên trong khoảng (0, 1), mảng ngẫu nhiên số nguyên trong khoảng (x, y), mảng ngẫu nhiên là permutation của số từ 1 đến n

1.2 Quản lý gói với Anaconda

Cài đặt package tại một branch của một project trên github

```
$ pip install git+https://github.com/tangentlabs/django-oscar-
    ↪ paypal.git@issue/34/oscar-0.6#egg=django-oscar-paypal
```

Trích xuất danh sách package

```
$ pip freeze > requirements.txt
```

Chạy ipython trong environment anaconda
Chạy đống lệnh này

```
conda install nb_conda
source activate my_env
python -m IPython kernelspec install-self --user
ipython notebook
```

Interactive programming với ipython
Trích xuất ipython ra slide (không hiểu sao default '–to slides' không work nữa, lại phải thêm tham số '–reveal-prefix' [1]

[1] tham khảo [pytorch](http://pytorch.org/docs/master/torch.html?highlight=randntorch.randn), [numpy](https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.random.html))

```
jupyter nbconvert "file.ipynb"
  --to slides
  --reveal-prefix "https://cdnjs.cloudflare.com/ajax/libs/reveal.
      ↪ js/3.1.0"
```

Tham khảo thêm

* https://stackoverflow.com/questions/37085665/in-which-conda-environment-is-jupyter-executing * https://github.com/jupyter/notebook/issues/541issuecomment-146387578 * https://stackoverflow.com/a/20101940/772391

python 3.4 hay 3.5

Có lẽ 3.5 là lựa chọn tốt hơn (phải có của tensorflow, pytorch, hỗ trợ mock)

Quản lý môi trường phát triển với conda

Chạy lệnh 'remove' để xóa một môi trường

```
conda remove --name flowers --all
```

1.3 Test với python

Sử dụng những loại test nào?

Hiện tại mình đang viết unittest với default class của python là Unittest. Thực ra toàn sử dụng 'assertEqual' là chính!

Ngoài ra mình cũng đang sử dụng tox để chạy test trên nhiều phiên bản python (python 2.7, 3.5). Điều hay của tox là mình có thể thiết kế toàn bộ cài đặt project và các dependencies package trong file 'tox.ini'

Chạy test trên nhiều phiên bản python với tox

Pycharm hỗ trợ debug tox (quá tuyệt!), chỉ với thao tác đơn giản là nhấn chuột phải vào file tox.ini của project.

1.4 Xây dựng docs với readthedocs và sphinx

20/12/2017: Tự nhiên hôm nay tất cả các class có khai báo kế thừa ở project languageflow không thể index được. Vãi thật. Làm thằng đệ không biết đâu mà build model.

Thử build lại chục lần, thay đổi file conf.py và package_reference.rst chán chê không được. Giả thiết đầu tiên là do hai nguyên nhân (1) docstring ghi sai, (2) nội dung trong package_reference.rst bị sai. Sửa chán chê cũng vẫn thế, thử checkout các commit của git. Không hoạt động!

Mất khoảng vài tiếng mới để ý thằng readthedocs có phần log cho từng build một. Lần mò vào build gần nhất và build (mình nhớ là) thành công cách đây 2 ngày

Log build gần nhất

```
Running Sphinx v1.6.5
making output directory...
loading translations [en]... done
```

```
loading intersphinx inventory from https://docs.python.org/
    ↪ objects.inv...
intersphinx inventory has moved: https://docs.python.org/objects.
    ↪ inv -> https://docs.python.org/2/objects.inv
loading intersphinx inventory from http://docs.scipy.org/doc/
    ↪ numpy/objects.inv...
intersphinx inventory has moved: http://docs.scipy.org/doc/numpy/
    ↪ objects.inv -> https://docs.scipy.org/doc/numpy/objects.
    ↪ inv
building [mo]: targets for 0 po files that are out of date
building [readthedocsdirhtml]: targets for 8 source files that
    ↪ are out of date
updating environment: 8 added, 0 changed, 0 removed
reading sources... [ 12%] authors
reading sources... [ 25%] contributing
reading sources... [ 37%] history
reading sources... [ 50%] index
reading sources... [ 62%] installation
reading sources... [ 75%] package_reference
reading sources... [ 87%] readme
reading sources... [100%] usage

looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [ 12%] authors
writing output... [ 25%] contributing
writing output... [ 37%] history
writing output... [ 50%] index
writing output... [ 62%] installation
writing output... [ 75%] package_reference
writing output... [ 87%] readme
writing output... [100%] usage
```

Log build hồi trước

```
Running Sphinx v1.5.6
making output directory...
loading translations [en]... done
loading intersphinx inventory from https://docs.python.org/
    ↪ objects.inv...
intersphinx inventory has moved: https://docs.python.org/objects.
    ↪ inv -> https://docs.python.org/2/objects.inv
loading intersphinx inventory from http://docs.scipy.org/doc/
    ↪ numpy/objects.inv...
intersphinx inventory has moved: http://docs.scipy.org/doc/numpy/
```

```
    ↪ objects.inv -> https://docs.scipy.org/doc/numpy/objects.
    ↪ inv
building [mo]: targets for 0 po files that are out of date
building [readthedocs]: targets for 8 source files that are out
    ↪ of date
updating environment: 8 added, 0 changed, 0 removed
reading sources... [ 12%] authors
reading sources... [ 25%] contributing
reading sources... [ 37%] history
reading sources... [ 50%] index
reading sources... [ 62%] installation
reading sources... [ 75%] package_reference
reading sources... [ 87%] readme
reading sources... [100%] usage

/home/docs/checkouts/readthedocs.org/user_builds/languageflow/
    ↪ checkouts/develop/languageflow/transformer/count.py:
    ↪ docstring of languageflow.transformer.count.
    ↪ CountVectorizer:106: WARNING: Definition list ends without
    ↪  a blank line; unexpected unindent.
/home/docs/checkouts/readthedocs.org/user_builds/languageflow/
    ↪ checkouts/develop/languageflow/transformer/tfidf.py:
    ↪ docstring of languageflow.transformer.tfidf.
    ↪ TfidfVectorizer:113: WARNING: Definition list ends without
    ↪  a blank line; unexpected unindent.
../README.rst:7: WARNING: nonlocal image URI found: https://img.
    ↪ shields.io/badge/latest-1.1.6-brightgreen.svg
looking for now-outdated files... none found
pickling environment... done
checking consistency... done
preparing documents... done
writing output... [ 12%] authors
writing output... [ 25%] contributing
writing output... [ 37%] history
writing output... [ 50%] index
writing output... [ 62%] installation
writing output... [ 75%] package_reference
writing output... [ 87%] readme
writing output... [100%] usage
```

Đập vào mắt là sự khác biệt giữa documentation type
Lỗi

```
building [readthedocsdirhtml]: targets for 8 source files that
    ↪ are out of date
```

Chạy

```
building [readthedocs]: targets for 8 source files that are out
    ↪ of date
```

Hí ha hí hửng. Chắc trong cơn bất loạn sửa lại settings đây mà. Sửa lại nó trong phần Settings (Admin gt; Settings gt; Documentation type)

Khi chạy nó đã cho ra log đúng

```
building [readthedocsdirhtml]: targets for 8 source files that
    ↪ are out of date
```

Nhưng vẫn lỗi. Vãi!!! Sau khoảng 20 phút tiếp tục bấn loạn, chửi bới readthedocs các kiểu. Thì để ý dòng này

Lỗi

```
Running Sphinx v1.6.5
```

Chạy

```
Running Sphinx v1.5.6
```

Ngay dòng đầu tiên mà không để ý, ngu thật. Aha, Hóa ra là thằng readthedocs nó tự động update phiên bản sphinx lên 1.6.5. Mình là mình chúa ghét thay đổi phiên bản (code đã mệt rồi, lại còn phải tương thích với nhiều phiên bản nữa thì ăn c** à). Đầu tiên search với Pycharm thấy dòng này trong ‘conf.py‘

```
# If your documentation needs a minimal Sphinx version, state it
    ↪ here.
# needs_sphinx = '1.0'
```

Đổi thành

```
# If your documentation needs a minimal Sphinx version, state it
    ↪ here.
needs_sphinx = '1.5.6'
```

Vẫn vậy (holy sh*t). Thử sâu một tẹo (thực sự là rất nhiều tẹo). Thấy cái này trong trang Settings

Ờ há. Thằng đần này cho phép trỏ đường dẫn tới một file trong project để cấu hình dependency. Haha. Tạo thêm một file ‘requirements‘ trong thư mục ‘docs‘ với nội dung

```
sphinx==1.5.6
```

Sau đó cấu hình nó trên giao diện web của readthedocs

Build thử. Build thử thôi. Cảm giác đúng lắm rồi đấy. Và... nó chạy. Ahihi

Kinh nghiệm

* Khi không biết làm gì, hãy làm 3 việc. Đọc LOG. Phân tích LOG. Và cố gắng để LOG thay đổi theo ý mình.

PS: Trong quá trình này, cũng không thèm build thằng PDF với Epub nữa. Tiết kiệm được bao nhiêu thời gian.

1.5 Pycharm Pycharm

01/2018: Pycharm là trình duyệt ưa thích của mình trong suốt 3 năm vừa rồi.

Hôm nay tự nhiên lại gặp lỗi không tự nhận unittest, không resolve được package import bởi relative path. Vụ không tự nhận unittest sửa bằng cách xóa file .idea là xong. Còn vụ không resolve được package import bởi relative path thì vẫn chịu rồi. Nhìn code cứ đỏ lòm khó chịu thật.

1.6 Vì sao lại code python?

01/11/2017 Thích python vì nó quá đơn giản (và quá đẹp).

[1] : *https* : *//github.com/jupyter/nbconvert/issues/*91*issuecomment* − 283736634

Chương 2

Java

01/11/2017: Java đơn giản là gay nhé. Không chơi. Viết java chỉ viết thế này thôi. Không viết hơn. Thề!

Chương 3

PHP

PHP là ngôn ngữ lập trình web dominate tất cả các anh tài khác mà (chắc là) chỉ dịu đi khi mô hình REST xuất hiện. Nhớ lần đầu gặp bạn Laravel mà cảm giác cuộc đời sang trang.

Cuối tuần này lại phải xem làm sao cài được xdebug vào PHPStorm cho thằng em tập tành lập trình. Haizzz

Tương tác với cơ sở dữ liệu

Liệt kê danh sách các bản ghi trong bảng groups

“‘ $sql = "SELECT * FROM `groups`"$;groups = mysqli$_query$(conn, sql); “‘

Xóa một bản ghi trong bảng groups

“‘ $sql = "DELETE FROM `groups` WHERE id = `5`"; mysqli_query$(conn, sql); “‘

Cài đặt debug trong PHPStorm

https://www.youtube.com/watch?v=mEJ21RB0F14

(1) XAMPP

- Download XAMPP (cho PHP 7.1.x - do XDebug chưa chính thức hỗ trợ 7.2.0) https://www.apachefriends.org/xampp-files/7.1.12/xampp-win32-7.1.12-0-VC14-installer.exe - Install XAMPP xampp-win32-7.1.12-0-VC14-installer.exe - Truy cập vào địa chỉ http://localhost/dashboard/phpinfo.php để kiểm tra cài đặt đã thành công chưa

(2) Tải và cài đặt PHPStorm

- Download PHPStorm https://download-cf.jetbrains.com/webide/PhpStorm-2017.3.2.exe - Install PHPStorm

(3) Tạo một web project trong PHPStorm - Chọn interpreter trỏ đến PHP trong xampp

(4) Viết một chương trình add.php

“‘php $a = 2$;b = 3; c =a + b;

echo c; “‘

Click vào `add.php`, chọn Debug, PHPStorm sẽ báo chưa cài XDebug

(5) Cài đặt XDebug theo hướng dẫn tại https://gist.github.com/odan/1abe76d373a9cbb15bed

Click vào add.php, chọn Debug

(6) Cài đặt XDebug với PHPStorm Marklets Vào trang https://www.jetbrains.com/phpstorm/markle
Trong phần Zend Debugger - chọn cổng 9000 - IP: 127.0.0.1 Nhấn nút Generate

Bookmark các link quot;Start debuggerquot;, quot;Stop debuggerquot; lên trình duyệt

(7) Debug PHP từ trình duyệt

* Vào trang http://localhost/untitled/add.php * Click vào bookmark Start debugger * Trong PHPStorm, nhấn vào biểu tượng quot;Start Listening for PHP Debug Connectionsquot; * Đặt breakpoint tại dòng thứ 5 * Refresh lại trang http://localhost/untitled/add.php, lúc này, breakpoint sẽ dừng ở dòng 5

Phần II

Xác suất

Chương 4

Xác suất

Phần này có thêm khảo Goodfellow u.a. (2016) và giáo trình xác suất thống kê của thạc sỹ Trần Thiện Khải, đại học Trà Vinh [1]

4.1 Các hàm phân phối thông dụng

17/01/2018 Lòng vòng thế nào hôm nay lại tìm được của bạn Đỗ Minh Hải [2], rất hay

4.1.1 Biến rời rạc

Phân phối đều - Discrete Uniform distribution

Là phân phối mà xác suất xuất hiện của các sự kiện là như nhau.
Biến ngẫu nhiên X tuân theo phân phối đều rời rạc

$$X \sim \mathcal{U}(a, b)$$

với tham số $a, b \in \mathbb{Z}; a < b$ là khoảng giá trị của X, đặt $n = b - a + 1$
Ta sẽ có:

Định nghĩa	Giá trị
PMF	$p(x) \mid \frac{1}{n}, \forall x \in [a, b]$
CDF - $F(x; a, b)$	$\frac{x - a + 1}{n}, \forall x \in [a, b]$
Kỳ vọng - $E[X]$	$\frac{a + b}{2}$
Phương sai - $Var(X)$	$\frac{n^2 - 1}{12}$

Ví dụ: Lịch chạy của xe buýt tại một trạm xe buýt như sau: chiếc xe buýt đầu tiên trong ngày sẽ khởi hành từ trạm này vào lúc 7 giờ, cứ sau mỗi 15 phút sẽ

[1] http://www.ctec.tvu.edu.vn/ttkhai/xacsuatthongke_dh.htm
[2] https://dominhhai.github.io/vi/2017/10/prob-com-var

có một xe khác đến trạm. Giả sử một hành khách đến trạm trong khoảng thời gian từ 7 giờ đến 7 giờ 30. Tìm xác suất để hành khách này chờ:

a) Ít hơn 5 phút.

b) Ít nhất 12 phút.

Giải

Gọi X là số phút sau 7 giờ mà hành khách đến trạm.

Ta có: $X \sim R[0; 30]$.

a) Hành khách sẽ chờ ít hơn 5 phút nếu đến trạm giữa 7 giờ 10 và 7 giờ 15 hoặc giữa 7 giờ 25 và 7 giờ 30. Do đó xác suất cần tìm là:

$$P(0 < X < 15) + P(25 < X < 30) = \frac{5}{30} + \frac{5}{30} = \frac{1}{3}$$

b) Hành khách chờ ít nhất 12 phút nếu đến trạm giữa 7 giờ và 7 giờ 3 phút hoặc giữa 7 giờ 15 phút và 7 giờ 18 phút. Xác suất cần tìm là:

$$P(0 < X < 3) + P(15 < X < 18) = \frac{3}{30} + \frac{3}{30} = \frac{1}{5}$$

Phân phối Béc-nu-li - Bernoulli distribution

Như đã đề cập về phép thử Béc-nu-li rằng mọi phép thử của nó chỉ cho 2 kết quả duy nhất là A với xác suất p và $\bar{A}$ với xác suất $q = 1 - p$ Biến ngẫu nhiên X tuân theo phân phối Béc-nu-li

$$X \sim B(p)$$

với tham số $p \in \mathbb{R}, 0 \leq p \leq 1$ là xác suất xuất hiện của A tại mỗi phép thử

Định nghĩa		Giá trị
PMF	p(x)	$p(x) \mid p^x(1-p)^{1-x}, x \in \{0, 1\}$
CDF	$F(x; p)$	$\begin{cases} 0 & \text{for } x < 0 \\ 1-p & \text{for } 0 \leq x < 1 \\ 1 & \text{for } x \geq 1 \end{cases}$
Kỳ vọng	$E[X]$	p
Phương sai	$Var(X)$	$p(1-p)$

Ví dụ

Tham khảo thêm các thuật toán khác tại Hai (2018)

Phần III

Khoa học máy tính

Chương 5

Hệ điều hành

Những phần mềm không thể thiếu

* Trình duyệt Google Chrome (với các extensions Scihub, Mendeley Desktop, Adblock) * Adblock extension * Terminal (Oh-my-zsh) * IDE Pycharm để code python * Quản lý phiên bản code Git * Bộ gõ ibus-unikey trong Ubuntu hoặc unikey (Windows) (Ctrl-Space để chuyển đổi ngôn ngữ) * CUDA (lập trình trên GPU)

Xem thông tin hệ thống

Phiên bản 'ubuntu 16.04'

```
sudo apt-get install sysstat
```

Xem hoạt động (

"' mpstat -A "'

CPU của mình có bao nhiều core, bao nhiêu siblibngs

"' cat /proc/cpuinfo

processor : 23 vendor$_id$: $GenuineIntelcpufamily$: 6$model$: 62$modelname$: $Intel(R)Xeon(R)CPUE5-2430v2@2.50GHzstepping$: 4$microcode$: 0$x428cpuMHz$: 1599.707$cachesize$: 15360$KBphysicalid$: 1$siblings$: 12$coreid$: 5$cpucores$: 6$apicid$: 43$initialapicid$: 43fpu : $yesfpu_exception$: $yescpuidlevel$: 13wp : $yesflags$: $fpuvmedepsetscmsrpaemcecx8apicsepmtrrpgemcacmovpatpse36clflushdtsacpimmxfxsrssesse2sshttm$ 5005.20$clflushsize$: 64$cache_alignment$: 64$addresssizes$: 46$bitsphysical, 48bitsvirtualpowermanagement$: "'

Kết quả cho thấy cpu của 6 core và 12 siblings

Chương 6

Ubuntu

Chuyện terminal

Terminal là một câu chuyện muôn thưở của bất kì ông coder nào thích customize, đẹp, tiện (và bug kinh hoàng). Hiện tại mình đang thấy combo này khá ổn Terminal (Ubuntu) (Color: Black on white, Build-in schemes: Tango) + zsh + oh-my-zsh (fishy-custom theme). Những features hay ho

* Làm việc tốt trên cả Terminal (white background) và embedded terminal của Pycharm (black background) * Hiển thị folder dạng ngắn (chỉ ký tự đầu tiên) * Hiển thị brach của git ở bên phải

![Imgur](https://i.imgur.com/q53vQdH.png)

Chuyện bộ gõ

Làm sao để khởi động lại ibus, thỉnh thoảng lại chết bất đắc kì tử [1]"‘$ibus - daemonibusrestart$"‘

Chuyện lỗi login loop

Phiên bản: ‘ubuntu 16.04‘

27/12/2017: Lại dính lỗi không thể login. Lần này thì lại phải xóa bạn KDE đi. Kể cũng hơn buồn. Nhưng nhất quyết phải enable được tính năng Windows Spreading (hay đại loại thế). Hóa ra khi ubuntu bị lỗi không có lancher hay toolbar là do bạn unity plugin chưa được enable. Oài. Sao người hiền lành như mình suốt ngày bị mấy lỗi vớ vẩn thế không biết.

20/11/2017: Hôm nay đen thật, dính lỗi login loop. Fix mãi mới được. Thôi cũng kệ. Cảm giác bạn KDE này đỡ bị lỗi ibus-unikey hơn bạn GNOME. Hôm nay cũng đổi bạn zsh theme. Chọn mãi chẳng được bạn nào ổn ổn, nhưng không thể chịu được kiểu suggest lỗi nữa rồi. Đôi khi thấy default vẫn là tốt nhất.

21/11/2017: Sau một ngày trải nghiệm KDE, cảm giác giao diện mượt hơn GNOME. Khi overview windows với nhiều màn hình tốt và trực quan hơn. Đặc biệt là không bị lỗi ibus nữa. Đổi terminal cũng cảm giác ổn ổn. Không bị lỗi suggest nữa.

[1] : $https://askubuntu.com/questions/389903/ibus-doesnt-seem-to-restart$

Phần IV

Học máy

- Vấn đề với HMM và CRF?
- Học MLE và MAP?

Machine Learning

Có bao nhiêu thuật toán Machine Learning?

Có rất nhiều thuật toán Machine Learning, bài viết [Điểm qua các thuật toán Machine Learning hiện đại](https://ongxuanhong.wordpress.com/2015/10/22/diem-qua-cac-thuat-toan-machine-learning-hien-dai/) của Ông Xuân Hồng tổng hợp khá nhiều thuật toán. Theo đó, các thuật toán Machine Learning được chia thành các nhánh lớn như 'regression', 'bayesian', 'regularization', 'decision tree', 'instance based', 'dimesionality reduction', 'clustering', 'deep learning', 'neural networks', 'associated rule', 'ensemble'... Ngoài ra thì còn có các cheatsheet của [sklearn](http://scikit-learn.org/stable/tutorial/machine$_learning_map/index.html$).

Việc biết nhiều thuật toán cũng giống như ra đường mà có nhiều lựa chọn về xe cộ. Tuy nhiên, quan trọng là có task để làm, sau đó thì cập nhật SOTA của task đó để biết các công cụ mới.

Xây dựng model cần chú ý điều gì?

Khi xây dựng một model cần chú ý đến vấn đề tối ưu hóa tham số (có thể sử dụng [GridSearchCV](sklearn.model$_selection.GridSearchCV$))

Bài phát biểu này có vẻ cũng rất hữu ích [PYCON UK 2017: Machine learning libraries you'd wish you'd known about](https://www.youtube.com/watch?v=nDF7$_8FOhpI$).$C\emptyset cp\emptyset n$

* [DistrictDataLabs/yellowbrick](https://github.com/DistrictDataLabs/yellowbrick) (giúp visualize model được train bởi sklearn) * [marcotcr/lime](https://github.com/marcotcr/lime) (giúp inspect classifier) * [TeamHG-Memex/eli5](https://github.com/TeamHG-Memex/eli5) (cũng giúp inspect classifier, hỗ trợ nhiều model như xgboost, crfsuite, đặc biệt có TextExplainer sử dụng thuật toán từ eli5) * [rhiever/tpot](https://github.com/rhiever/tpot) (giúp tối ưu hóa pipeline) * [dask/dask](https://github.com/dask/dask) (tính toán song song và lập lịch)

Ghi chú về các thuật toán trong xử lý ngôn ngữ tự nhiên tại [undertheseа.flow/wiki](https://github.com/magizbox/underthesea.flow/wiki/Develop)

Framework để train, test hiện tại vẫn rất thoải mái sklearn. tensorboard cung cấp phần log cũng khá hay.

[Câu trả lời hay](https://www.quora.com/What-are-the-most-important-machine-learning-techniques-to-master-at-this-time/answer/Sean-McClure-3?srid=5O2u) cho câu hỏi [Những kỹ thuật machine learning nào quan trọng nhất để master?](https://www.quora.com/What-are-the-most-important-machine-learning-techniques-to-master-at-this-time), đặc biệt là dẫn đến bài [The State of ML and Data Science 2017](https://www.kaggle.com/surveys/2017) của Kaggle.

Tài liệu học PGM

[Playlist youtube](https://www.youtube.com/watch?v=WPSQfOkb1M8amp;list=PL50E6E80E8525B59C) khóa học Probabilistic Graphical Models của cô Daphne Koller. Ngoài ra còn có một [tutorial](http://mensxmachina.org/files/software/demos/bayesnetdemo.html) dở hơi ở đâu về tạo Bayesian network

[Chưa biết] Tại sao Logistic Regression lại là Linear Model?

Trong quyển Deep Learning, chương 6, trang 165, tác giả có viết

“‘ Linear models, such as logistic regression and linear regression, are appealing because they can be t eciently and reliably, either in closed form or with convex optimization “‘

Mình tự hỏi tại sao logistic regression lại là linear, trong khi nó có sử dụng hàm logit (nonlinear)? Tìm hiểu hóa ra cũng có bạn hỏi giống mình trên [stats.stackexchange.com](https://stats.stackexchange.com/questions/93569/why-is-logistic-regression-a-linear-classifier). Ngoài câu trả lời trên stats.stackexchange, đọc một số cái khác [Generalized Linear Models, SPSS Statistics 22.0.0](https://www.ibm.com/support/k $IntroductiontoGeneralizedLinearModels, AnalysisofDiscreteData, PennsylvaniaStateUniversity](h$ $//onlinecourses.science.psu.edu/stat504/node/216)cngvnchahiulm.$

Hiện tại chỉ hiểu là các lớp model này chỉ có thể hoạt động trên các tập linear separable, có lẽ do việc map input x, luôn có một liên kết linear $latexwx$, trước khi đưa vào hàm non-linear.

Các tập dữ liệu thú vị

Iris dataset: dữ liệu về hoa iris

Là một ví dụ cho bài toán phân loại

Weather problem: dữ liệu thời tiết. Có thể tìm được ở trong quyển Data Mining: Practical Machine Learning Tools and Techniques

Là một ví dụ cho bài toán cây quyết định

Deep Learning

Tài liệu Deep Learning

Lang thang thế nào lại thấy trang này [My Reading List for Deep Learning!](https://www.microsoft.co us/research/wp-content/uploads/2017/02/DL$_{R}eading_{L}ist.pdf)camtanhMicrosoft.TrongØ, (Øngnhin)$

Các layer trong deep learning [2]

Sparse Layers

[**nn.Embedding**](http://pytorch.org/docs/master/nn.htmlembedding) ([hướng dẫn](http://pytorch.org/tutorials/beginner/nlp/word$_{e}mbeddings_{t}utorial.html)) $grepcode$: [$Shawn1993/cnn-text-classification-pytorch$]($https://github.com/Shawn1993/cnn-text-classification-pytorch/blob/master/model.pyL18$)Œ$engvaitrnhmtlookuptable, mapmtwordviden$

Convolution Layers

[**nn.Conv1d**](http://pytorch.org/docs/master/nn.htmlconv1d), [**nn.Conv2d**](http://pytorch. [**nn.Conv3d**](http://pytorch.org/docs/master/nn.htmlconv3d) [1] $grepcode$: [$Shawn1993/cnn-text-classification-pytorch$]($https://github.com/Shawn1993/cnn-text-classification-pytorch/blob/master/model.pyL20-L24$), [$galsang/CNN-sentence-classification-pytorch$]($https://github.com/galsang/CNN-sentence-classification-pytorch/blob/master/model.pyL36-L38$)

Các tham số trong Convolution Layer

* ‘kernel$_{s}ize$‘($haylfiltersize$)

Đối với NLP, kernel$_{s}izethngbngregion_{s}ize*word_{d}im$(Øiviconv1d)$hay(region_{s}ize, word_{d}im)$Øiviconv

<small>Quá trình tạo feature map đối với region size bằng 2</small>

* ‘in$_{c}hannels$‘, ‘$out_{c}hannels$‘($lslng$‘$featuremaps$‘)

Kênh (channels) là các cách nhìn (view) khác nhau đối với dữ liệu. Ví dụ, trong ảnh thường có 3 kênh RGB (red, green, blue), có thể áp dụng convolution giữa các kênh. Với văn bản cũng có thể có các kênh khác nhau, như khi có các

kênh sử dụng các word embedding khác nhau (word2vec, GloVe), hoặc cùng một câu nhưng biểu diễn ở các ngôn ngữ khác nhau.

* ‘stride‘

Định nghĩa bước nhảy của filter.

Hình minh họa sự khác biệt giữa các feature map đối với stride=1 và stride=2. Feature map đối với stride = 1 có kích thước là 5, feature map đối với stride = 3 có kích thước là 3. Stride càng lớn thì kích thước của feature map càng nhỏ.

Trong bài báo của Kim 2014, ‘stride = 1‘ đối với ‘nn.conv2d‘ và ‘stride = word$_d$$im$‘$Øivi$‘$nn.conv1d$‘

Toàn bộ tham số của mạng CNN trong bài báo Kim 2014,

| Description | Values | |—————|————| | input word vectors | Google word2vec | | filter region size | (3, 4, 5) | | feature maps | 100 | | activation function | ReLU | | pooling | 1-max pooling | | dropout rate | 0.5 | | $latexlamp; s = 22$ norm constraint | 3 |

Đọc thêm:

* [Lecture 13: Convolutional Neural Networks (for NLP). CS224n-2017](http://web.stanford.edu/class/cs224n/lect 2017-lecture13-CNNs.pdf) * [DeepNLP-models-Pytorch - 8. Convolutional Neural Networks](https://nbviewer.jupyter.org/github/DSKSD/DeepNLP-models-Pytorch/blob/master/notebooks/08.CNN-for-Text-Classification.ipynb) * [A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Zhang 2015](https://arxiv.org/pdf/1510.03820.pdf)

BTS

22/11/2017 - Phải nói quyển này hơi nặng so với mình. Nhưng thôi cứ cố gắng vậy. 24/11/2017 - Từ hôm nay, mỗi ngày sẽ ghi chú một phần (rất rất nhỏ) về Deep Learning [tại đây](https://docs.google.com/document/d/1KxDrw5s6uYHNLda7t0rhp0RM$_T$$lUGxydQ-Qi1JOPFr8/edit?usp = sharing)

[1] : [$UnderstandingConvolutionalNeuralNetworksforNLP$]($http : //www.wildml.com/2015/11/understandi convolutional-neural-networks-for-nlp$)[2] : $http : //pytorch.org/docs/master/nn.html$

Phần V

Linh tinh

Chương 7

Nghiên cứu

Các công cụ

[Google Scholar](https://scholar.google.com.vn/) vẫn là lựa chọn tốt

* Tìm kiếm tác giả theo lĩnh vực nghiên cứu và quốc gia: sử dụng filter label: + đuôi * ví dụ: [danh sách các nhà nghiên cứu Việt Nam thuộc lĩnh vực xử lý ngôn ngữ tự nhiên (label:natural$_l$anguage$_p$rocessing + .vn)](*https* : *//scholar.google.com.vn/citations?hl* = *enamp*; *view*$_o$*p* = *search*$_a$*uthorsamp*; *mauthors* = *label* ∗ *danhschnyØspxptheolngtrchdn*

Bên cạnh đó còn có [semanticscholar](https://www.semanticscholar.org/) (một project của [allenai](http://allenai.org/)) với các killer features

* [Tìm kiếm các bài báo khoa học với từ khóa và filter theo năm, tên hội nghị](https://www.semanticscholar.org/search?venue* [Xem những người ảnh hưởng, ảnh hưởng bởi một nhà nghiên cứu, cũng như xem co-author, journals và conferences mà một nhà nghiên cứu hay gửi bài](https://www.semanticscholar.org/author/Christopher-D-Manning/1812612)

Mendeley rất tốt cho việc quản lý và lưu trữ. Tuy nhiên điểm hạn chế lại là không lưu thông tin về citation

Các hội nghị tốt về xử lý ngôn ngữ tự nhiên

* Rank A: ACL, EACL, NAACL, EMNLP, CoNLL * Rank B: SemEval

Các tạp chí

* [Computational Linguistics (CL)](http://www.mitpressjournals.org/loi/coli)

Câu chuyện của Scihub

Sci-Hub được tạo ra vào ngày 5 tháng 9 năm 2011, do nhà nghiên cứu đến từ Kazakhstan, [Alexandra Elbakyan](https://en.wikipedia.org/wiki/Alexandra$_E$*lbakyan*)

Hãy nghe chia sẻ của cô về sự ra đời của Sci-Hub

> Khi tôi còn là một sinh viên tại Đại học Kazakhstan, tôi không có quyền truy cập vào bất kỳ tài liệu nghiên cứu. Những bai bao tôi cần cho dự án nghiên cứu của tôi. Thanh toán 32 USD thi thât la điên rô khi bạn cần phải đọc lướt hoặc đọc hàng chục hoặc hàng trăm tơ để làm nghiên cứu. Tôi có được những bai bao nhơ vao trôm chúng. Sau đó tôi thấy có rất nhiều và rất nhiều nhà nghiên cứu (thậm chí không phải sinh viên, nhưng các nhà nghiên cứu trường đại học) giống như tôi, đặc biệt là ở các nước đang phát triển. Họ đã tạo ra các

cộng đồng trực tuyến (diễn đàn) để giải quyết vấn đề này. Tôi là một thành viên tích cực trong một cộng đồng như vậy ở Nga. Ở đây ai cần có một bài nghiên cứu, nhưng không thể trả tiền cho nó, có thể đặt một yêu cầu và các thành viên khác, những người có thể có được những giấy sẽ gửi nó cho miễn phí qua email. Tôi có thể lấy bất cứ bai nao, vì vậy tôi đã giải quyết nhiều yêu cầu và người ta luôn rất biết ơn sự giúp đỡ của tôi. Sau đó, tôi tạo Sci-Hub.org, một trang web mà chỉ đơn giản là làm cho quá trình này tự động và các trang web ngay lập tức đã trở thành phổ biến.

Về phần mình, là một nhà nghiên cứu trẻ, đương nhiên phải đọc liên tục. Các báo cáo ở Việt Nam về xử lý ngôn ngữ tự nhiên thì thường không tải lên các trang mở như arxiv.org, các kỷ yếu hội nghị cũng không public các proceedings. Thật sự scihub đã giúp mình rất nhiều.

Scihub bị chặn*

Vào thời điểm này (12/2017), scihub bị chặn quyết liệt. Hóng được trên page facebook của scihub các cách truy cập scihub. Đã thử các domain khác như .tw, .hk. Mọi chuyện vẫn ổn cho đến hôm nay (21/12/2017), không thể truy cập vào nữa.

Đành phải cài tor để truy cập vào scihub ở địa chỉ http://scihub22266oqcxt.onion/https://dl.acm.org/c
Và mọi chuyện lại ổn.

Làm sao để nghiên cứu tốt

* Làm việc mỗi ngày * Viết nhật ký nghiên cứu mỗi tuần (tổng kết công việc tuần trước, các ý tưởng mới, kế hoạch tuần này) * Cập nhật các kết quả từ các hội nghị, tạp chí

Sách giáo khoa

* [Machine Learning Yearning, by Andrew Ng](https://gallery.mailchimp.com/dc3a7ef4d750c0abfc192

Lượm lặt

* Review các khóa học Deep Learning: https://www.kdnuggets.com/2017/10/3-popular-courses-deep-learning.html

Mở đầu

(01/11/2017) Không biết mình có phải làm nghiên cứu không nữa? Vừa kiêm phát triển, vừa đọc paper mỗi ngày. Thôi, cứ (miễn cưỡng) cho là nghiên cứu viên đi.

Chương 8

Nghề lập trình

Chân kinh con đường lập trình: [Teach Yourself Programming in Ten Years. Peter Norvig](http://norvig.com/21-days.html)

Trang web hữu ích

* Chia sẻ thú vị: [15 năm lập trình ở Việt Nam](https://vozforums.com/showthread.php?t=3431312) của Blanic (vozfourm) * Trang web chứa cheatsheet so sánh các ngôn ngữ lập trình và công nghệ http://hyperpolyglot.org/

01/11/2017

Vậy là đã vào nghề (đi làm full time trả lương) được 3 năm rưỡi rồi. Thời gian trôi qua nhanh như *ó chạy ngoài đồng thật. Tâm đắc nhất với câu trong một quyển gì đó của anh lead HR google. Có 4 level của nghề nghiệp. 1 là thỏa mãn được yêu cầu cả bản. 2 là dự đoán được tương lai. 3 là cá nhân hóa (ý nói là tận tình với các khách hàng). 4 là phiêu diêu tự tại. Hay thật! Bao giờ mới được vào mức 4 đây.

Tài liệu tham khảo

Goodfellow, Ian / Bengio, Yoshua / Courville, Aaron (2016): *Deep Learning.* , MIT Press.

Hai, Do (2018): *Một số phân phối phổ biến* .