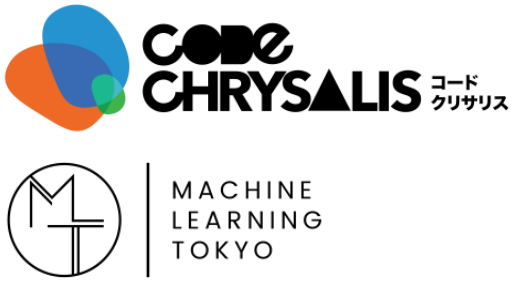
AI Open Education: Stanford Deep Learning Cheat Sheets in Japanese



30th October 2019, Code Chrysalis x MLT MiniConf #6
Yuta Kanzawa @yutakanzawa

SFE Senior Analyst at Janssen Pharmaceutical K.K., Tokyo
A Family Company of Johnson & Johnson

I am...

- 神沢雄大 Yuta Kanzawa (twitter: @yutakanzawa)
- Data scientist at Janssen Japan, Tokyo
- Opera & wine lover
  - Wagner
  - Bourgogne
- 7 languages
  - Human: Japanese, English, German
  - Computer: R, Python, SAS, SQL

Question:

**Do you know
any effective way to learn
machine learning &
deep learning?**

My answer:

Stanford ML/DL Cheat Sheets

Agenda

- Overview
  - Stanford ML/DL Cheat Sheets
  - Open education motivation
  - MLT Team
- How we have done
  - Self-assignment
  - Tools & process
  - Implementation
- What we have done
  - Results
  - Recognition

Overview

- Stanford ML/DL Cheat Sheets
- Open education motivation
- Translation Team in Machine Learning Tokyo

Stanford ML/DL Cheat Sheets by Afshine & Shervine Amidi

'Easy-to-digest study guides'*

VIP Cheatsheet: Supervised Learning

Afshine Amidi and Shervine Amidi

September 9, 2018

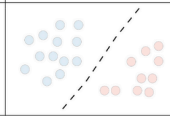
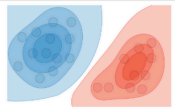
Introduction to Supervised Learning

Given a set of data points $\{x^{(1)},...,x^{(m)}\}$ associated to a set of outcomes $\{y^{(1)},...,y^{(m)}\}$, we want to build a classifier that learns how to predict y from x.

❒ **Type of prediction** – The different types of predictive models are summed up in the table below:

	Regression	Classifier
Outcome	Continuous	Class
Examples	Linear regression	Logistic regression, SVM, Naive Bayes

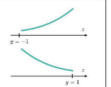
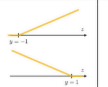
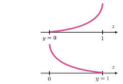
❒ **Type of model** – The different models are summed up in the table below:

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Notations and general concepts

❒ **Hypothesis** – The hypothesis is noted h_θ and is the model that we choose. For a given input data $x^{(i)}$, the model prediction output is $h_\theta(x^{(i)})$.

❒ **Loss function** – A loss function is a function $L:(z,y)\in\mathbb{R}\times Y\longmapsto L(z,y)\in\mathbb{R}$ that takes as inputs the predicted value z corresponding to the real data value y and outputs how different they are. The common loss functions are summed up in the table below:

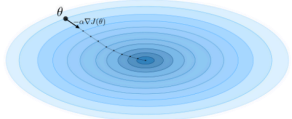
Least squared	Logistic	Hinge	Cross-entropy
$\frac{1}{2}(y-z)^2$	$\log(1+\exp(-yz))$	$\max(0,1-yz)$	$-\left[y\log(z)+(1-y)\log(1-z)\right]$
			
Linear regression	Logistic regression	SVM	Neural Network

❒ **Cost function** – The cost function J is commonly used to assess the performance of a model, and is defined with the loss function L as follows:

$$J(\theta)=\sum_{i=1}^{m}L(h_\theta(x^{(i)}),y^{(i)})$$

❒ **Gradient descent** – By noting $\alpha\in\mathbb{R}$ the learning rate, the update rule for gradient descent is expressed with the learning rate and the cost function J as follows:

$$\theta\longleftarrow\theta-\alpha\nabla J(\theta)$$



Remark: Stochastic gradient descent (SGD) is updating the parameter based on each training example, and batch gradient descent is on a batch of training examples.

❒ **Likelihood** – The likelihood of a model $L(\theta)$ given parameters θ is used to find the optimal parameters θ through maximizing the likelihood. In practice, we use the log-likelihood $\ell(\theta)=\log(L(\theta))$ which is easier to optimize. We have:

$$\theta^{\text{opt}}=\arg\max_\theta L(\theta)$$

❒ **Newton's algorithm** – The Newton's algorithm is a numerical method that finds θ such that $\ell'(\theta)=0$. Its update rule is as follows:

$$\theta\leftarrow\theta-\frac{\ell'(\theta)}{\ell''(\theta)}$$

Remark: the multidimensional generalization, also known as the Newton-Raphson method, has the following update rule:

$$\theta\leftarrow\theta-\left(\nabla^2_\theta\ell(\theta)\right)^{-1}\nabla_\theta\ell(\theta)$$

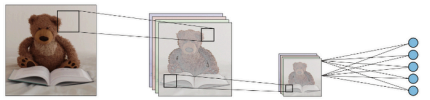
VIP Cheatsheet: Convolutional Neural Networks

Afshine Amidi and Shervine Amidi

November 26, 2018

Overview

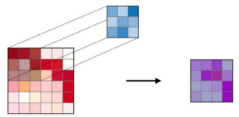
❒ **Architecture of a traditional CNN** – Convolutional neural networks, also known as CNNs, are a specific type of neural networks that are generally composed of the following layers:



The convolution layer and the pooling layer can be fine-tuned with respect to hyperparameters that are described in the next sections.

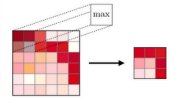
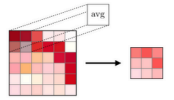
Types of layer

❒ **Convolutional layer (CONV)** – The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size F and stride S. The resulting output O is called *feature map* or *activation map*.

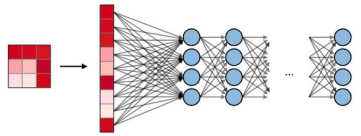


Remark: the convolution step can be generalized to the 1D and 3D cases as well.

❒ **Pooling (POOL)** – The pooling layer (POOL) is a downsampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.

	Max pooling	Average pooling
Purpose	Each pooling operation selects the maximum value of the current view	Each pooling operation averages the values of the current view
Illustration		
Comments	- Preserves detected features - Most commonly used	- Downsamples feature map - Used in LeNet

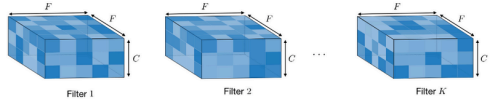
❒ **Fully Connected (FC)** – The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.



Filter hyperparameters

The convolution layer contains filters for which it is important to know the meaning behind its hyperparameters.

❒ **Dimensions of a filter** – A filter of size $F\times F$ applied to an input containing C channels is a $F\times F\times C$ volume that performs convolutions on an input of size $I\times I\times C$ and produces an output feature map (also called activation map) of size $O\times O\times 1$.



Remark: the application of K filters of size $F\times F$ results in an output feature map of size $O\times O\times K$.

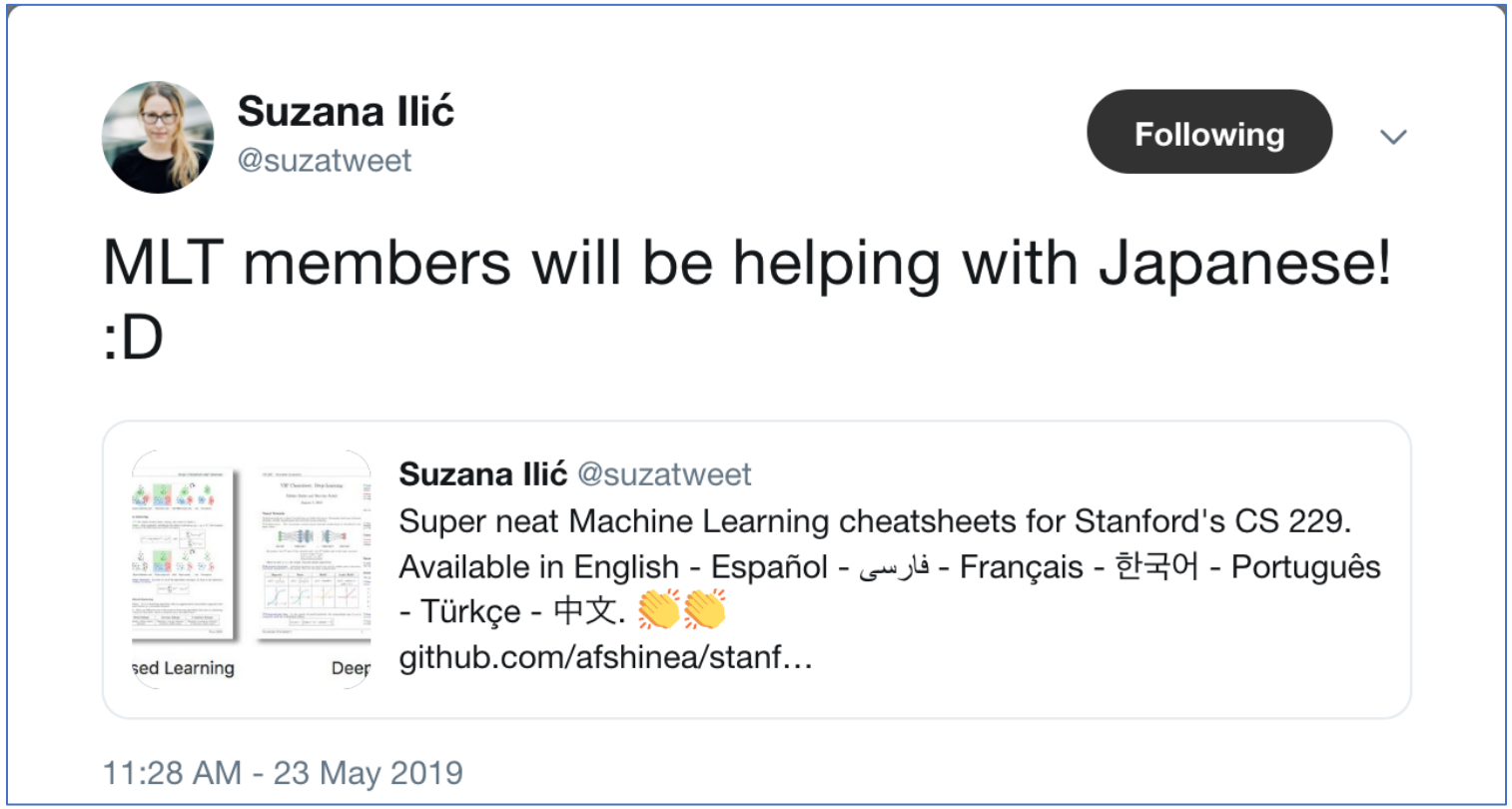
❒ **Stride** – For a convolutional or a pooling operation, the stride S denotes the number of pixels by which the window moves after each operation.

* Shervine Amidi, the author of the cheat sheets. https://stanford.edu/~shervine

Open education motivation

- High-quality contents from the top-level university.
- For machine learning beginners.
- Shape machine learning community.

Open education motivation (cont)



Me: ‘Sounds interesting!’

Translation Team in Machine Learning Tokyo

- 11 Machine Learning Tokyo members voluntarily joined!

Duy Linh Dang	Kamuela Lau	Suzana Ilić	Yoshiyuki Nakai
Hideaki Hamano	Nao Takatoshi	Tran Tuan Anh	Yuta Kanzawa
Hiroki Mori	Rob Altena	Wataru Oniki	

- Diversity in:
  - Native language
  - ML/DL experience
  - Academic background
  - Industry

How we have done

- Self-assignment
- Tools & process
- Implementation

Self-assignment

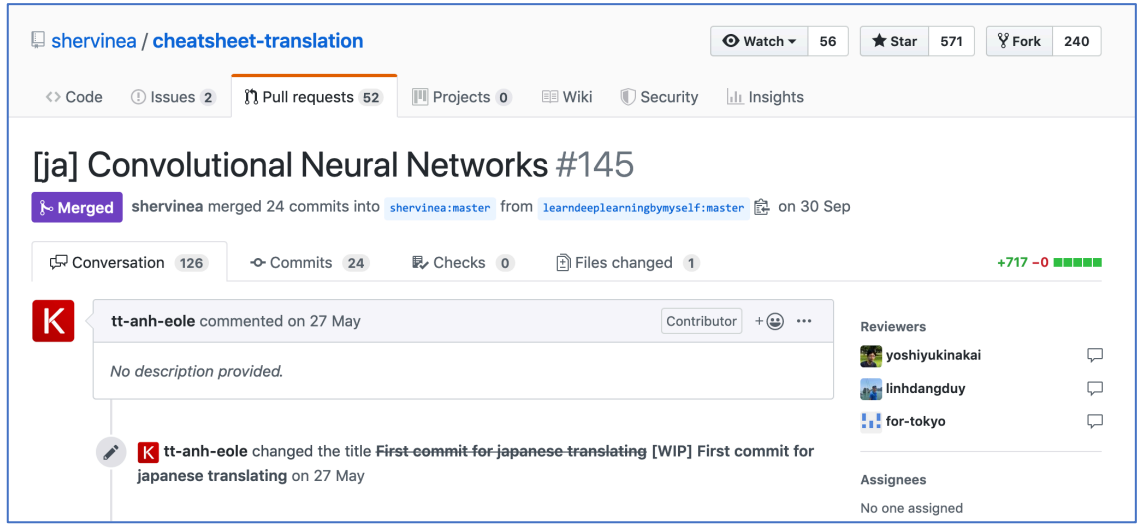
- All voluntary.

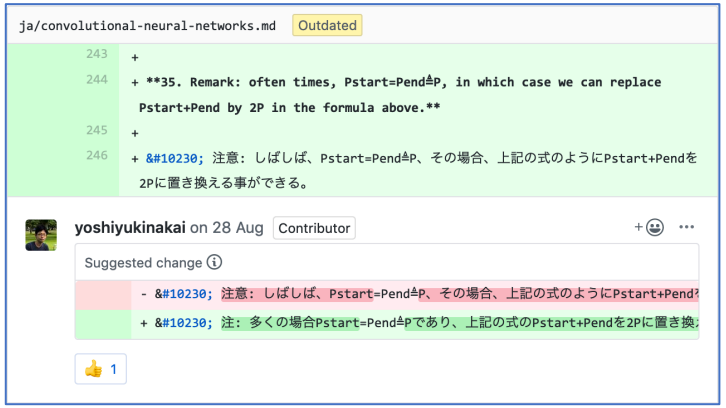
Cheat Sheet	Translator(s)	Reviewer(s)
Supervised Learning	Yuta	Tran
Unsupervised Learning	Tran, Wataru, Yoshiyuki, Yuta	Wataru, Yoshiyuki, Yuta
Algebra and Calculus	Rob	Kamuela
Probabilities and Statistics	Nao	Yuta
Convolutional Neural Nets	Tran, Yoshiyuki	Duy, Yoshiyuki, Wataru
Recurrent Neural Nets	Hideaki	Yoshiyuki
Deep Learning Tips and Tricks	Kamuela	Hiroki, Yoshiyuki

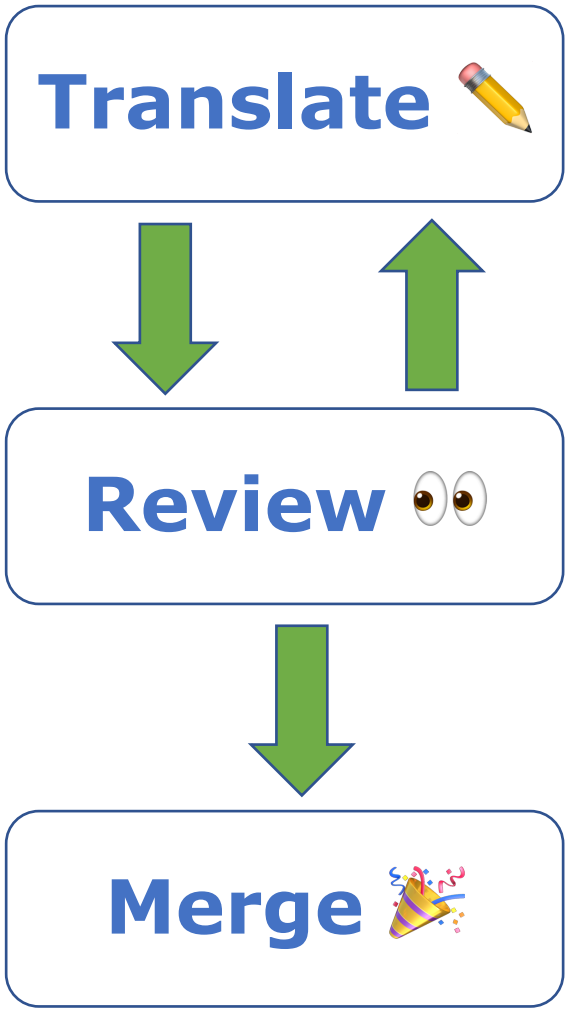
NB: Some of them had been translated by some people before.

Tools & process

- GitHub (repository): Useful to collaborate

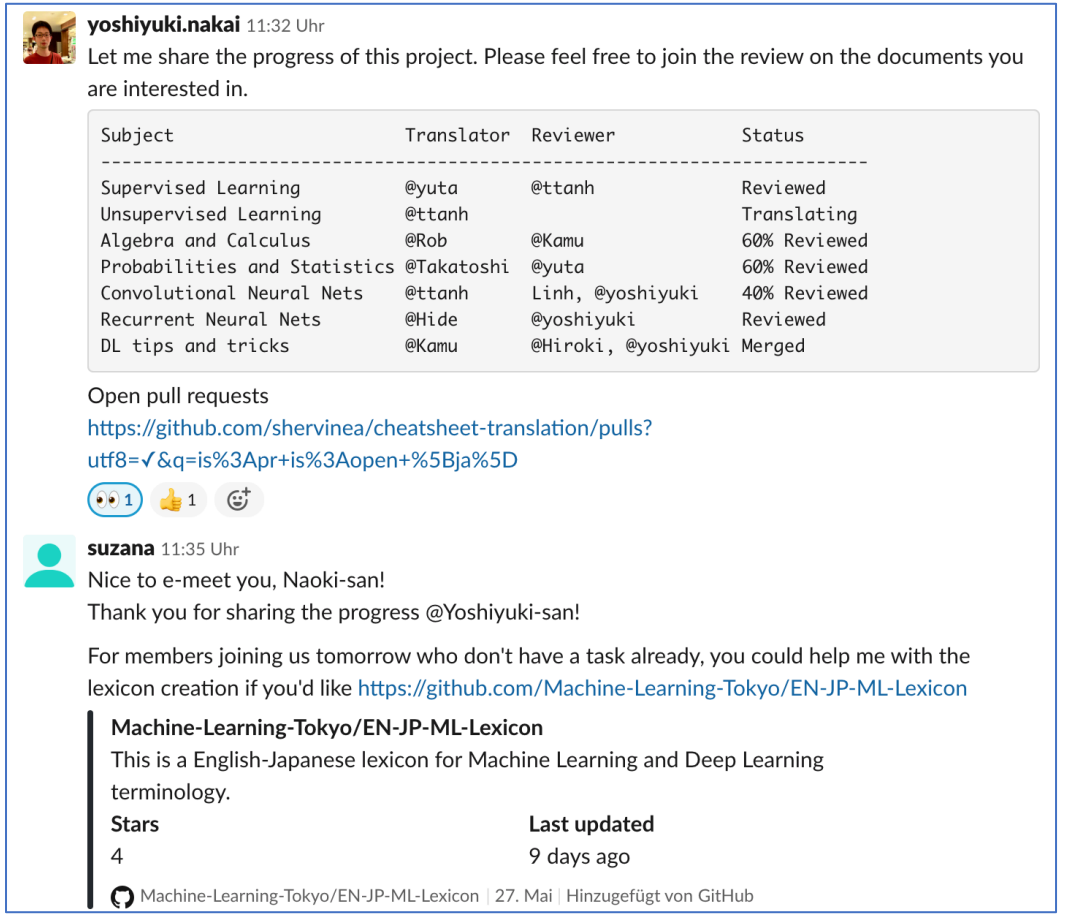






Tools & process (cont)

- Communications: Slack
- Search: Google (mainly)



yoshiyuki.nakai 11:32 Uhr

Let me share the progress of this project. Please feel free to join the review on the documents you are interested in.

Subject	Translator	Reviewer	Status
Supervised Learning	@yuta	@ttanh	Reviewed
Unsupervised Learning	@ttanh		Translating
Algebra and Calculus	@Rob	@Kamu	60% Reviewed
Probabilities and Statistics	@Takatoshi	@yuta	60% Reviewed
Convolutional Neural Nets	@ttanh	Linh, @yoshiyuki	40% Reviewed
Recurrent Neural Nets	@Hide	@yoshiyuki	Reviewed
DL tips and tricks	@Kamu	@Hiroki, @yoshiyuki	Merged

Open pull requests
https://github.com/shervinea/cheatsheet-translation/pulls?utf8=✓&q=is%3Apr+is%3Aopen+%5Bja%5D

👀 1 👍 1

suzana 11:35 Uhr

Nice to e-meet you, Naoki-san!
Thank you for sharing the progress @Yoshiyuki-san!

For members joining us tomorrow who don't have a task already, you could help me with the lexicon creation if you'd like https://github.com/Machine-Learning-Tokyo/EN-JP-ML-Lexicon

Machine-Learning-Tokyo/EN-JP-ML-Lexicon
This is a English-Japanese lexicon for Machine Learning and Deep Learning terminology.

Stars	Last updated
4	9 days ago

Machine-Learning-Tokyo/EN-JP-ML-Lexicon | 27. Mai | Hinzugefügt von GitHub

Implementation

- After slow progress of work at home…

- 5x workshops*1
  - Translation
  - Review
  - Lexicon*2

  +
- Meanwhile
  - Discussion in a Slack channel
  - Personal work at home

*1 Venues (cool meeting rooms!) are courtesy of Rakuten, Inc. and Safie Inc. See https://safie.link/safietimes/news/781
*2 https://github.com/Machine-Learning-Tokyo/EN-JP-ML-Lexicon

What we have done

- Results
- Recognition

Results

English (original)

CS 230 – Deep Learning https://stanford.edu/~shervine

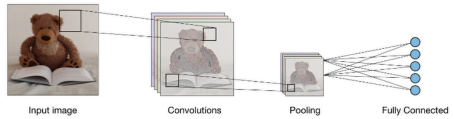
VIP Cheatsheet: Convolutional Neural Networks

Afshine Amidi and Shervine Amidi

November 26, 2018

Overview

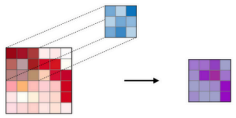
❒ **Architecture of a traditional CNN** – Convolutional neural networks, also known as CNNs, are a specific type of neural networks that are generally composed of the following layers:



The convolution layer and the pooling layer can be fine-tuned with respect to hyperparameters that are described in the next sections.

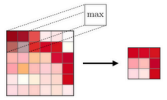
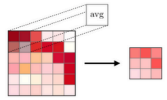
Types of layer

❒ **Convolutional layer (CONV)** – The convolution layer (CONV) uses filters that perform convolution operations as it is scanning the input I with respect to its dimensions. Its hyperparameters include the filter size F and stride S. The resulting output O is called *feature map* or *activation map*.

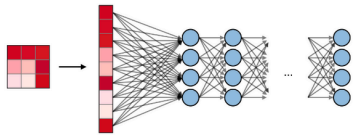


Remark: the convolution step can be generalized to the 1D and 3D cases as well.

❒ **Pooling (POOL)** – The pooling layer (POOL) is a downsampling operation, typically applied after a convolution layer, which does some spatial invariance. In particular, max and average pooling are special kinds of pooling where the maximum and average value is taken, respectively.

	Max pooling	Average pooling
Purpose	Each pooling operation selects the maximum value of the current view	Each pooling operation averages the values of the current view
Illustration		
Comments	- Preserves detected features - Most commonly used	- Downsamples feature map - Used in LeNet

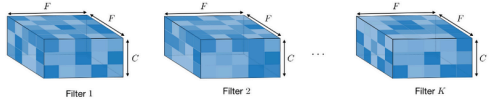
❒ **Fully Connected (FC)** – The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. If present, FC layers are usually found towards the end of CNN architectures and can be used to optimize objectives such as class scores.



Filter hyperparameters

The convolution layer contains filters for which it is important to know the meaning behind its hyperparameters.

❒ **Dimensions of a filter** – A filter of size $F \times F$ applied to an input containing C channels is a $F \times F \times C$ volume that performs convolutions on an input of size $I \times I \times C$ and produces an output feature map (also called activation map) of size $O \times O \times 1$.



Remark: the application of K filters of size $F \times F$ results in an output feature map of size $O \times O \times K$.

❒ **Stride** – For a convolutional or a pooling operation, the stride S denotes the number of pixels by which the window moves after each operation.

Stanford University 1 Winter 2019

Japanese (translated)*

CS 230 – 深層学習 https://stanford.edu/~shervine/l/ja

VIP チートシート: 畳み込みニューラルネットワーク

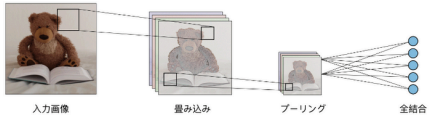
アフシンアミディ・シェルビンアミディ著

October 7, 2019

チャントウアンアイン・中井喜之訳

概要

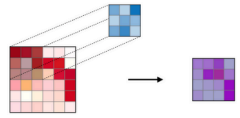
❒ **伝統的な畳み込みニューラルネットワークのアーキテクチャ** – CNNとしても知られる畳み込みニューラルネットワークは一般的に次の層で構成される特定種類のニューラルネットワークです。



畳み込み層とプーリング層は次のセクションで説明されるハイパーパラメータに関してファインチューニングできます。

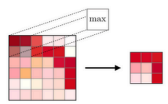
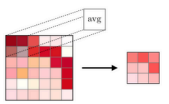
層の種類

❒ **畳み込み層(CONV)** – 畳み込み層(CONV)は入力Iを各次元に関して走査する時に、畳み込み演算を行うフィルタを使用します。畳み込み層のハイパーパラメータにはフィルタサイズFとストライドSが含まれます。結果出力Oは特徴マップまたは活性化マップと呼ばれます。

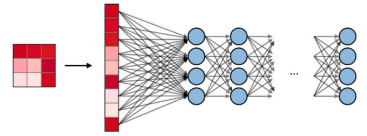


注: 畳み込みステップは1次元や3次元の場合にも一般化できます。

❒ **プーリング(POOL)** – プーリング層(POOL)は位置不変性をもつ縮小操作で、通常は畳み込み層の後に適用されます。特に、最大及び平均プーリングはそれぞれ最大と平均値が取られる特別な種類のプーリングです。

	最大プーリング	平均プーリング
種類	各プーリング操作は現在のビューの中から最大値を選ぶ	各プーリング操作は現在のビューに含まれる値を平均する
図		
コメント	- 検出された特徴を保持する - 最も一般的に利用される	- 特徴マップをダウンサンプリングする - LeNetで利用される

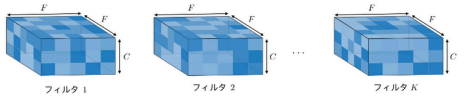
❒ **全結合(FC)** – 全結合(FC) 層は平坦化された入力に対して演算を行います。各入力は全てのニューロンに接続されています。FC層が存在する場合、通常CNNアーキテクチャの末尾に向かって見られ、クラススコアなどの目的を最適化するため利用できます。



フィルタハイパーパラメータ

畳み込み層にはハイパーパラメータの背後にある意味を知ることが重要なフィルタが含まれています。

❒ **フィルタの次元** – C個のチャネルを含む入力に適用される$F \times F$サイズのフィルタの体積は$F \times F \times C$で、それは$I \times I \times C$サイズの入力に対して畳み込みを実行して$O \times O \times 1$サイズの特徴マップ（活性化マップとも呼ばれる）出力を生成します。

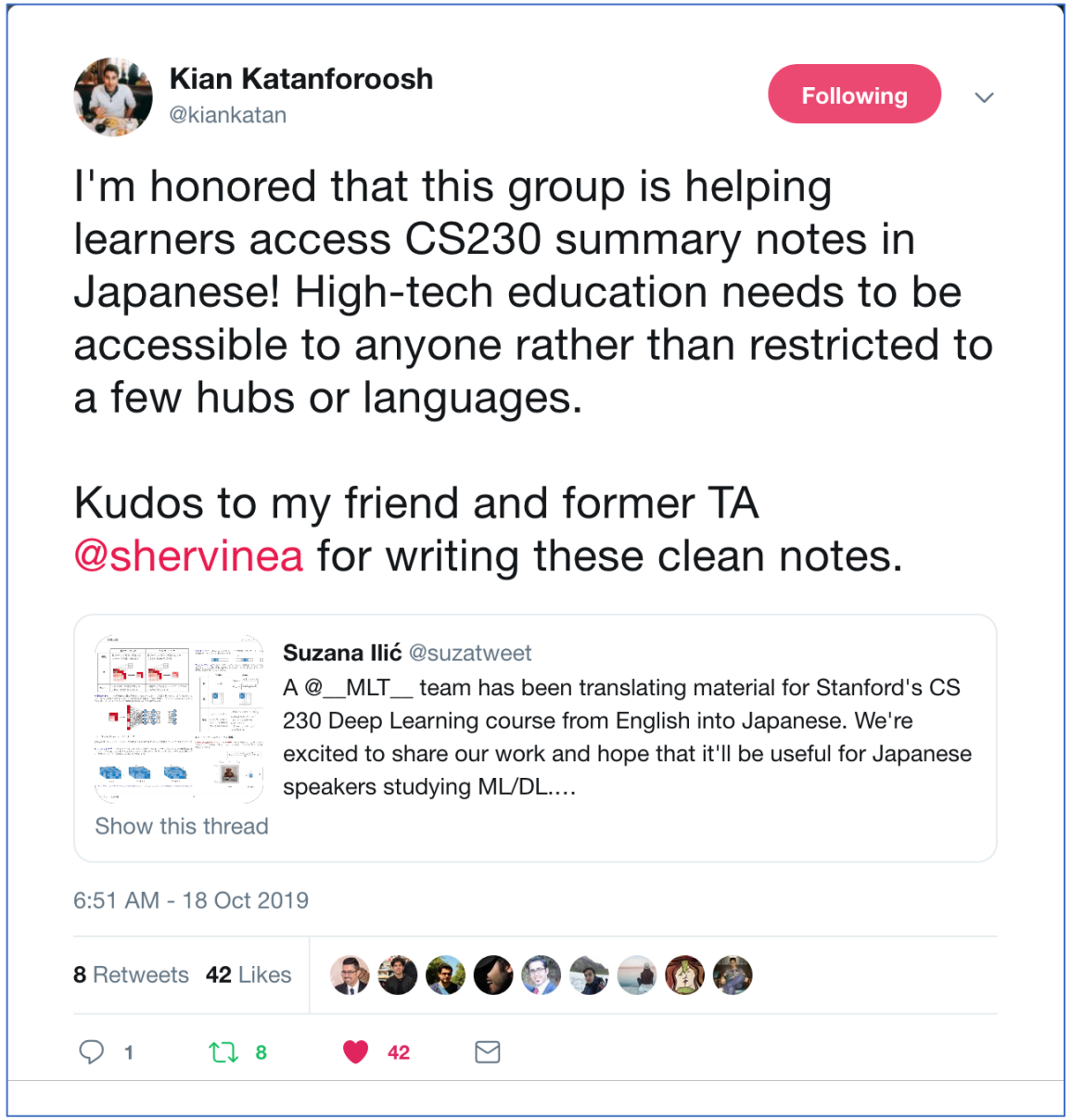


注: $F \times F$サイズのK個のフィルタを適用すると、$O \times O \times K$サイズの特徴マップの出力を得られます。

スタンフォード大学 1 冬2019

* https://github.com/afshinea/stanford-cs-230-deep-learning/blob/master/ja/cheatsheet-convolutional-neural-networks.pdf

Recognition 🎉



Still in progress 💪

- In other languages
- Catch-up with updates in the original

CS 230 (Deep Learning)

	Convolutional Neural Networks	Recurrent Neural Networks	Deep Learning tips
العَرَبِيَّة	not started	not started	not started
Català	not started	not started	not started
Deutsch	not started	not started	not started
Español	not started	not started	not started
فارسی	done	done	done
Suomi	not started	not started	not started
Français	done	done	done
עִבְרִית	not started	not started	not started
हिन्दी	not started	not started	not started
Magyar	not started	not started	not started
Bahasa Indonesia	done	in progress	in progress
Italiano	not started	not started	not started
日本語	done	done	done
한국어	in progress	in progress	in progress
Polski	not started	not started	not started
Português	done	not started	not started
Русский	not started	not started	not started
Türkçe	done	done	done
Українська	not started	not started	not started
Tiếng Việt	in progress	not started	in progress
简体中文	not started	in progress	not started
繁體中文	not started	not started	not started

https://github.com/shervinea/cheatsheet-translation

Enjoy!

Appendix: Open source contribution

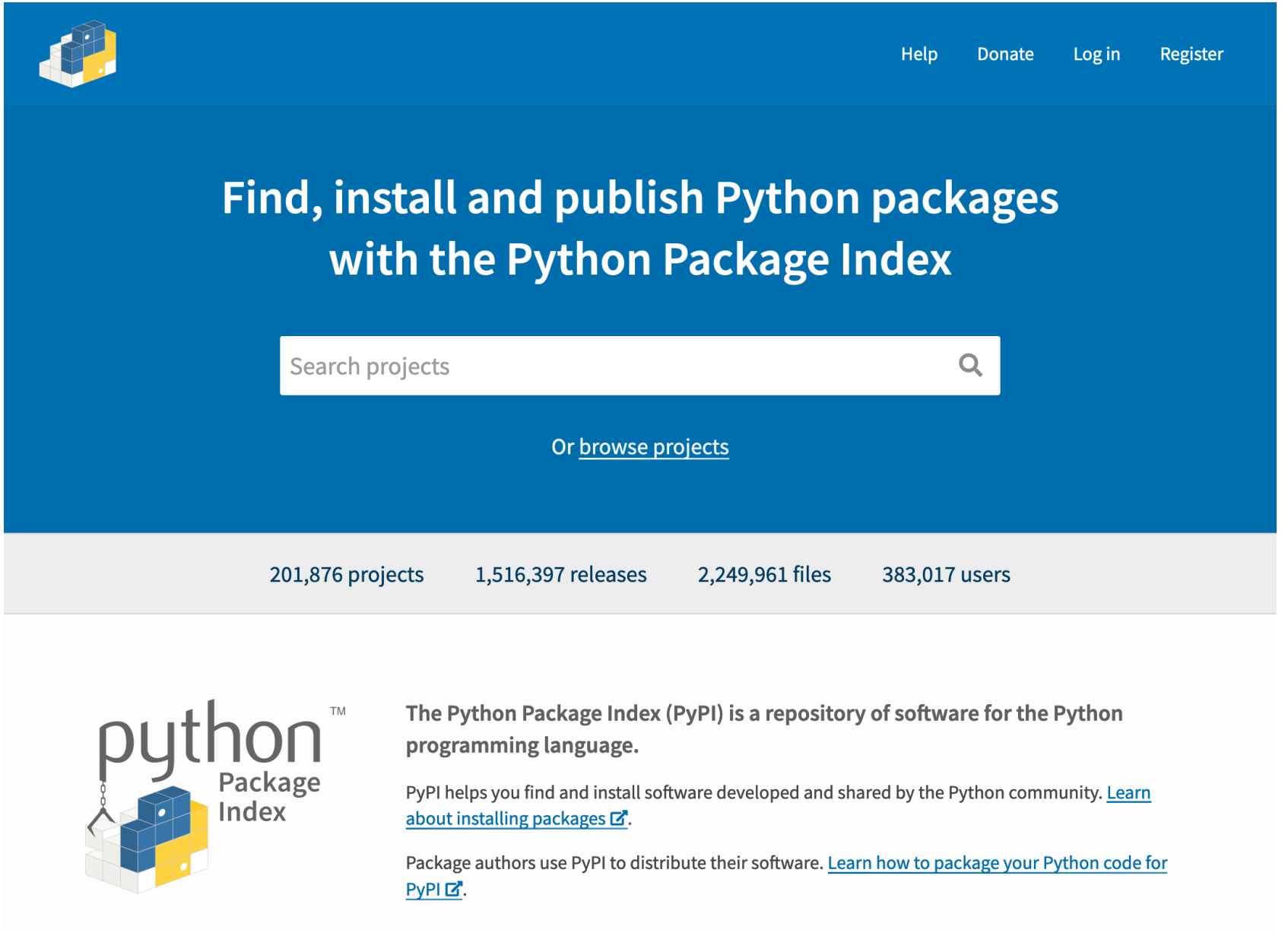
- Another case: PyPI translation
- Localisation

Another case: PyPI translation

- PyPI: Python Package Index https://pypi.org/
  `pip install`

- Implementation
  - Outside MLT
  - 6+ people (lead: @komo_fr)
  - One-day workshop + work at home
  - Weblate

English



https://pypi.org/

Japanese

https://pypi.org/

Recognition 👏👏👏

komo_fr@転職先ゆるぼ
@komo_fr
Following

🎉🎉🎉🎉🎉😂
#PyPI In Japan, we held a one-day workshop and translated! Thanks to @yutakanzawa @mssknd @u_shinji0612 ryok @mogutan88 !

Ghost Dusters @di_codes
Just added Brazilian Portuguese & Japanese as the 1st & 2nd fully translated languages for @pypi! 🇧🇷 🇯🇵

What language will be next!? We currently have Spanish: 88%,…
Show this thread

6:49 AM - 8 Oct 2019

3 Retweets **10** Likes

3 10

Ghost Dusters
@di_codes
Following

Just added Brazilian Portuguese & Japanese as the 1st & 2nd fully translated languages for @pypi! 🇧🇷 🇯🇵

What language will be next!? We currently have Spanish: 88%, Turkish: 82% and German: 70%.

Contribute to our translation effort via @weblateorg here: hosted.weblate.org/projects/pypa/…

Still in progress 💪

Language	Strings	Words	Needs editing	Checks	Suggestions	Comments	
French	100.0%	100.0%	0.0%	0.0%	0.0%	1.1%	Translate
German	100.0%	100.0%	0.0%	0.0%	2.0%	1.7%	Translate
Japanese	100.0%	100.0%	0.0%	0.0%	0.0%	2.5%	Translate
Portuguese (Brazil)	100.0%	100.0%	0.0%	0.0%	0.0%	1.1%	Translate
Spanish (Spain)	100.0%	100.0%	0.0%	0.0%	0.0%	1.1%	Translate
Ukrainian	100.0%	100.0%	0.0%	0.0%	1.5%	1.2%	Translate
Turkish	86.3%	72.5%	6.2%	1.7%	0.1%	1.5%	Translate
Chinese (Simplified)	64.6%	33.8%	3.2%	0.7%	0.5%	1.1%	Translate
Korean	63.6%	35.7%	3.5%	1.2%	1.5%	1.1%	Translate
Norwegian Bokmål	28.3%	8.9%	4.7%	1.2%	0.0%	1.1%	Translate
Arabic	21.0%	17.1%	1.4%	0.0%	0.0%	1.1%	Translate
Chinese (Traditional)	13.8%	8.9%	3.4%	0.1%	0.2%	1.1%	Translate
Polish	12.7%	7.8%	1.1%	0.1%	0.0%	1.1%	Translate
Indonesian	11.6%	8.0%	0.0%	0.0%	0.0%	1.1%	Translate
Dutch	10.5%	5.8%	0.9%	0.0%	0.0%	1.1%	Translate
Russian	9.0%	6.8%	0.7%	0.1%	0.5%	1.1%	Translate
Portuguese (Portugal)	8.3%	4.0%	0.5%	0.0%	0.0%	1.1%	Translate
Thai	7.8%	6.1%	0.2%	0.0%	0.0%	1.1%	Translate
Italian	5.4%	3.5%	0.5%	0.1%	1.5%	1.1%	Translate
Hungarian	0.6%	0.4%	0.0%	0.0%	0.0%	1.1%	Translate
Hebrew	0.1%	0.1%	0.0%	0.0%	0.6%	1.1%	Translate
Amharic	0.0%	0.0%	0.0%	0.0%	0.0%	1.1%	Translate
Romanian	0.0%	0.0%	0.0%	0.0%	0.0%	1.1%	Translate

https://hosted.weblate.org/projects/pypa/warehouse/

Localisation 🌏

- Many package documents are only in English.
  - Books, articles too.
  - Or need to be updated.

- You can contribute to open source too!
  - Translate into any language you use
  - Keep catching up with updates in the original
  - Help others following your path
  - Learning through translating