오픈소스로 여는 뉴노멀

2020 공개SW 페스티벌

과학기술정보통신부
Ministry of Science and ICT

nipa 정보통신산업진흥원
National IT Industry Promotion Agency

딥-러닝의 대중화
Keras & AutoKeras

케라스 코리아
전미정

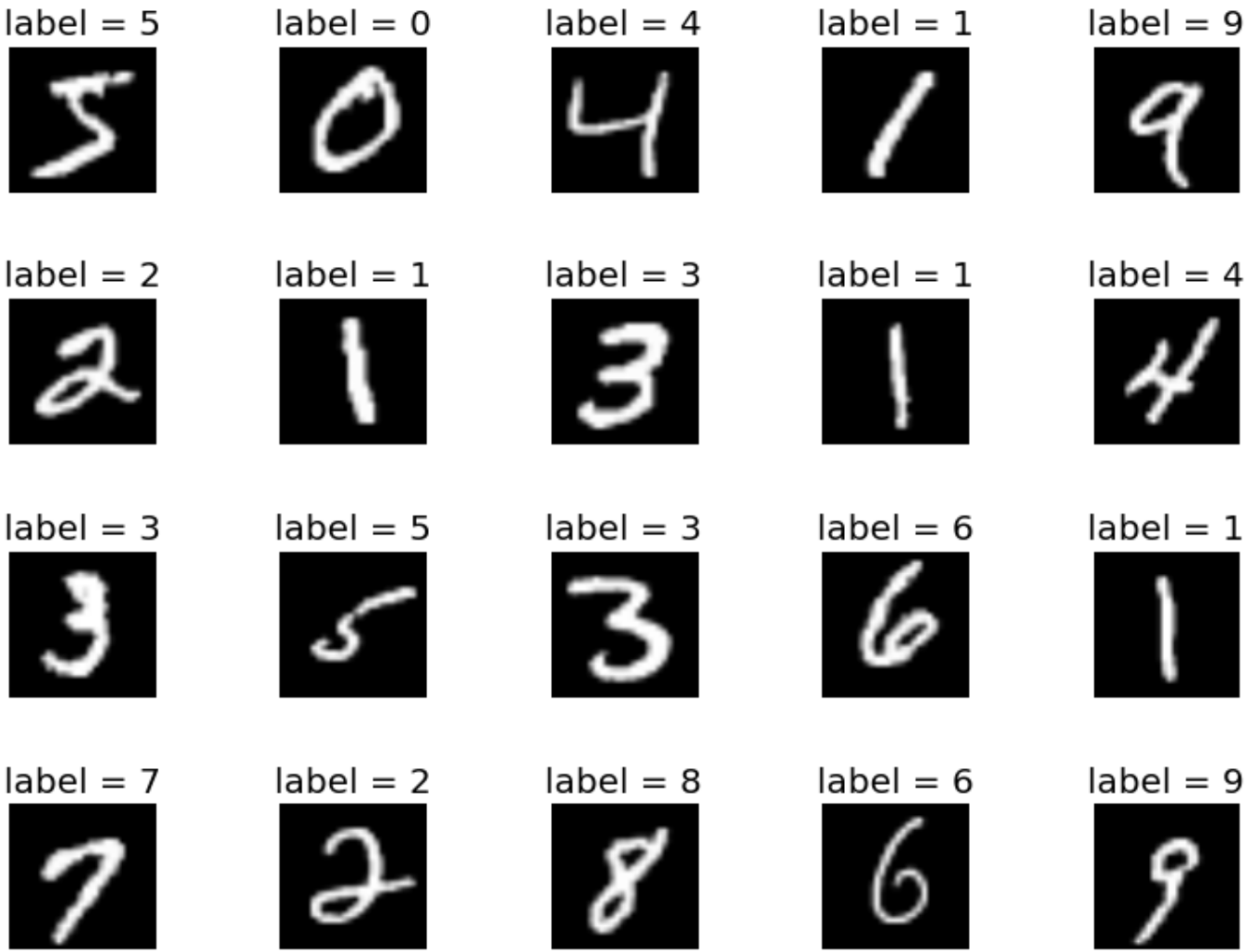
목차

Keras

κέρας = 그리스어 “뿔”

▶ 케라스

- 2016.04.16. ver.1.0.0
- 구글 엔지니어 Francois Chollet 개발한 오픈소스
- Python으로 구현된 high-level deep learning API
- TesnsorFlow Back-end
- 일관되고 간단한 문법, 사용자의 액션 최소화
- 풍부한 문서와 가이드 제공



https://t1.daumcdn.net/cfile/tistory/99BB78335993E64726

Tensorflow

```
hidden_activation = tf.nn.sigmoid(tf.matmul(x, weights) +
hidden_bias)
visible_reconstruction =
tf.nn.sigmoid(tf.matmul(hidden_activation,
tf.transpose(weights))
    + visible_bias)
final_hidden_activation =
tf.nn.sigmoid(tf.matmul(visible_reconstruction, weights)
    + hidden_bias)

positive_phase = tf.matmul(tf.transpose(x),
hidden_activation)
negative_phase =
tf.matmul(tf.transpose(visible_reconstruction),
final_hidden_activation)
LEARNING_RATE = 0.01
weight_update = weights.assign_add(LEARNING_RATE *
    (positive_phase - negative_phase))
visible_bias_update = visible_bias.assign_add(LEARNING_RATE
*
    tf.reduce_mean(x - visible_reconstruction, 0))
hidden_bias_update = hidden_bias.assign_add(LEARNING_RATE *
    tf.reduce_mean(hidden_activation -
final_hidden_activation, 0))

train_op = tf.group(weight_update, visible_bias_update,
hidden_bias_update)
loss_op = tf.reduce_sum(tf.square(x -
visible_reconstruction))

session = tf.Session()
session.run(tf.global_variables_initializer())
current_epochs = 0

for i in range(20):
    total_loss = 0
    while mnist.train.epochs_completed == current_epochs:
        batch_inputs, batch_labels =
mnist.train.next_batch(100)
        _, reconstruction_loss = session.run([train_op,
loss_op], feed_dict={input_placeholder: batch_inputs})
        total_loss += reconstruction_loss

    print("epochs %s loss %s" % (current_epochs,
reconstruction_loss))
    current_epochs = mnist.train.epochs_completed
```

Keras

```
model = Sequential()
model.add(Dense(hidden_neurons, input_dim=input_size))
model.add(Activation('relu'))
model.add(Dense(classes, input_dim=hidden_neurons))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
    metrics=['accuracy'], optimizer='adadelta')

model.fit(X_train, Y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1)

score = model.evaluate(X_test, Y_test, verbose=1)
print('\n''Test accuracy:', score[1])
```

Tensorflow

```
hidden_activation = tf.nn.sigmoid(tf.matmul(x, weights) +
hidden_bias)
visible_reconstruction =
tf.nn.sigmoid(tf.matmul(hidden_activation,
tf.transpose(weights))
    + visible_bias)
final_hidden_activation =
tf.nn.sigmoid(tf.matmul(visible_reconstruction, weights)
    + hidden_bias)

positive_phase = tf.matmul(tf.transpose(x),
hidden_activation)
negative_phase =
tf.matmul(tf.transpose(visible_reconstruction),
final_hidden_activation)
LEARNING_RATE = 0.01
weight_update = weights.assign_add(LEARNING_RATE *
    (positive_phase - negative_phase))
```

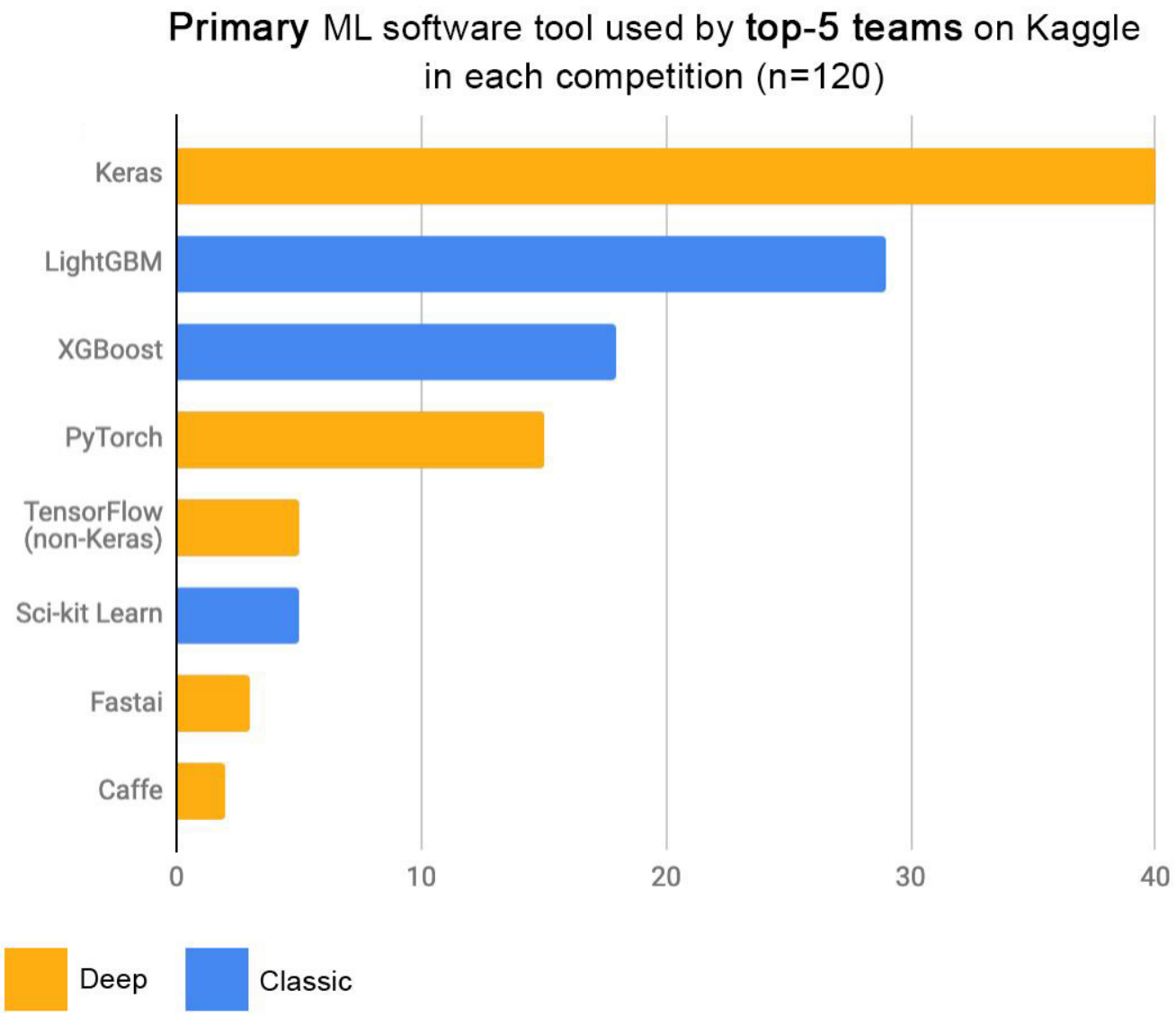
Keras

```
model = Sequential()
model.add(Dense(hidden_neurons, input_dim=input_size)
model.add(Activation('relu'))
model.add(Dense(classes, input_dim=hidden_neurons))
model.add(Activation('softmax'))

model.compile(loss='categorical_crossentropy',
    metrics=['accuracy'], optimizer='adadelta')

model.fit(X_train, Y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1)
```

AI 접근성 향상

딥러닝의 대중화

Keras



https://keras.io/why_keras/

Keras

Simple. Flexible. Powerful.

Get started | Guides | API docs

```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a trained vision model
vision_model = keras.applications.ResNet50()

# This is our video.encoding branch using the trained vision_model
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
embedded_question = layers.Embedding(10000, 256)(question_input)
encoded_question = layers.LSTM(256)(embedded_question)

# And this is our video question answering model:
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(1000, activation='softmax')(merged)
video_qa_model = keras.Model(inputs=[video_input, question_input],
                             outputs=output)
```

Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

https://keras.io/

AutoKeras

Version 1.0.0
2020.01.17.

AutoML

ML 아이템 추천

ML의 ML 추천!

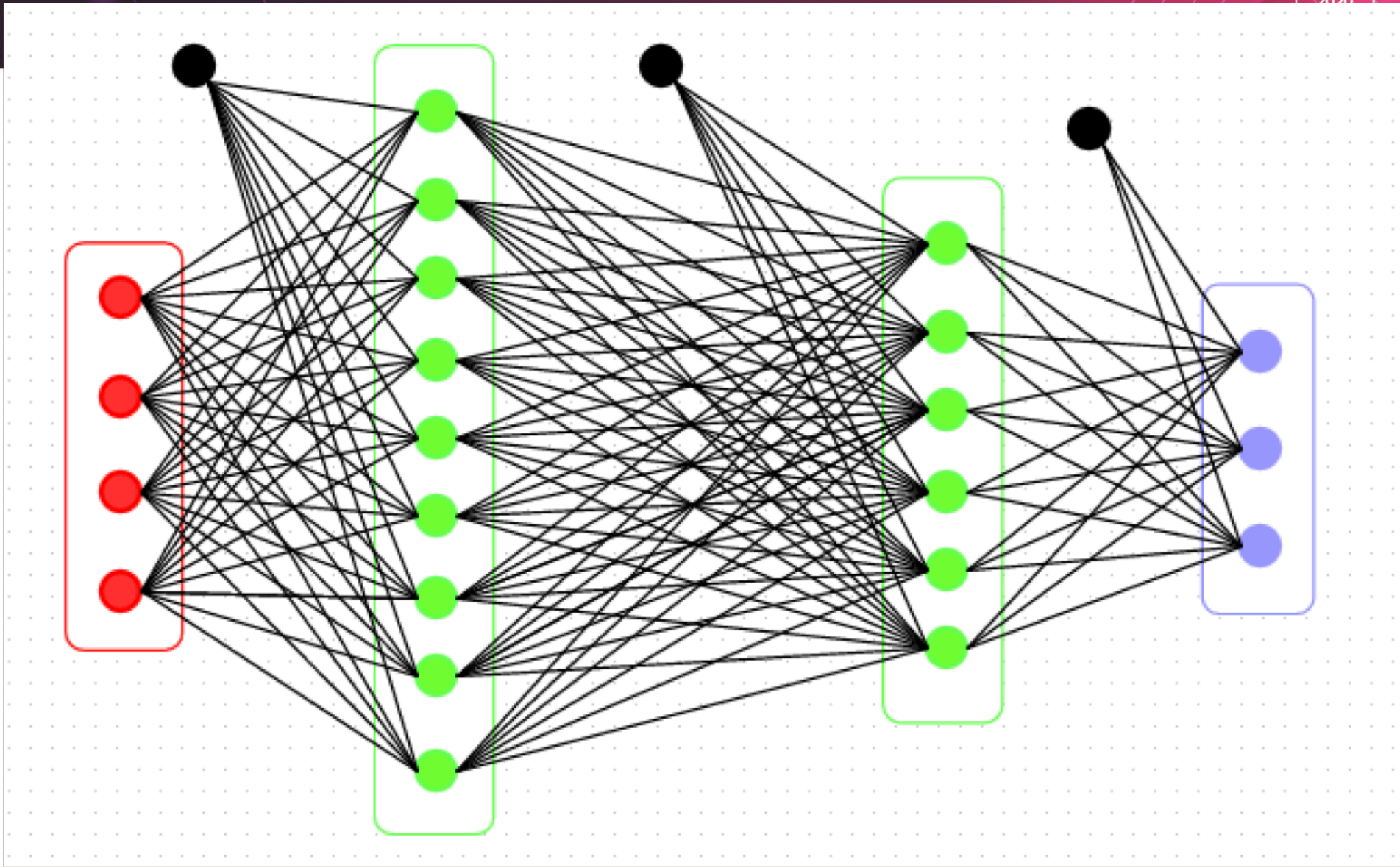
AutoKeras

AutoKeras

AutoKeras: An AutoML system based on Keras. It is developed by DATA Lab at Texas A&M University.

The goal of AutoKeras is to make machine learning accessible for everyone.

https://autokeras.com



http://www.parallelr.com/r-deep-neural-network-from-scratch/

- 몇 개의 layer를 쌓을 것인가?
- 하나의 layer에 몇 개의 node를 사용할 것인가?
- 어떤 종류의 layer를 활용할 것인가?
- Filter size, dropout percentage 등 parameter
- 일관된 모델 사용법

케라스 코리아
어떤 활동하나요?

Search Bar Text

케라스 사용자

안녕하세요, 혼자 딥러닝 공부하며 많은 도움을 받았던 좋은 자료 추천 드립니다.

1) 블록과 함께 하는 파이썬...
2) 파이썬과 케라스로 배우는...
3) 케라스로 구현하는 딥러닝과...

100 likes 100 Comments

Like Comment Share

Search Bar Text

케알못

이미지 분류 해보았습니다!

100 likes 100 Comments

Like Comment Share

Search Bar Text

케라스 사용자

딥러닝 컨퍼런스가 개최되네요! 링크 공유드립니다.

https://tykimos.github.io/2018/06/28/ISS_1st_Deep_Learning_Conference_All_Together/

100 likes 100 Comments

Like Comment Share

Search Bar Text

케알못

케라스 2.4.0 버전이 업데이트 되었습니다. 새로워진 내용 함께 살펴봐요~

1) ImageGenertor에 'new' 인자 추가
2) 버그 수정
3) 문서 업데이트

100 likes 100 Comments

Like Comment Share

Search Bar Text

케라스 사용자

object detection 할때 새로 데이터셋 모아서 학습하지 않고 클래스를 추가하는 방법이 있는지 궁금합니다.

100 likes 100 Comments

Like Comment Share

Search Bar Text

케알못

Autoencoder를 이용해 cluster 문제를 하고있습니다. ... coder에서 만들어진 데이터를 빼낼 수 있는 방법이 있을까요?

100 likes 100 Comments

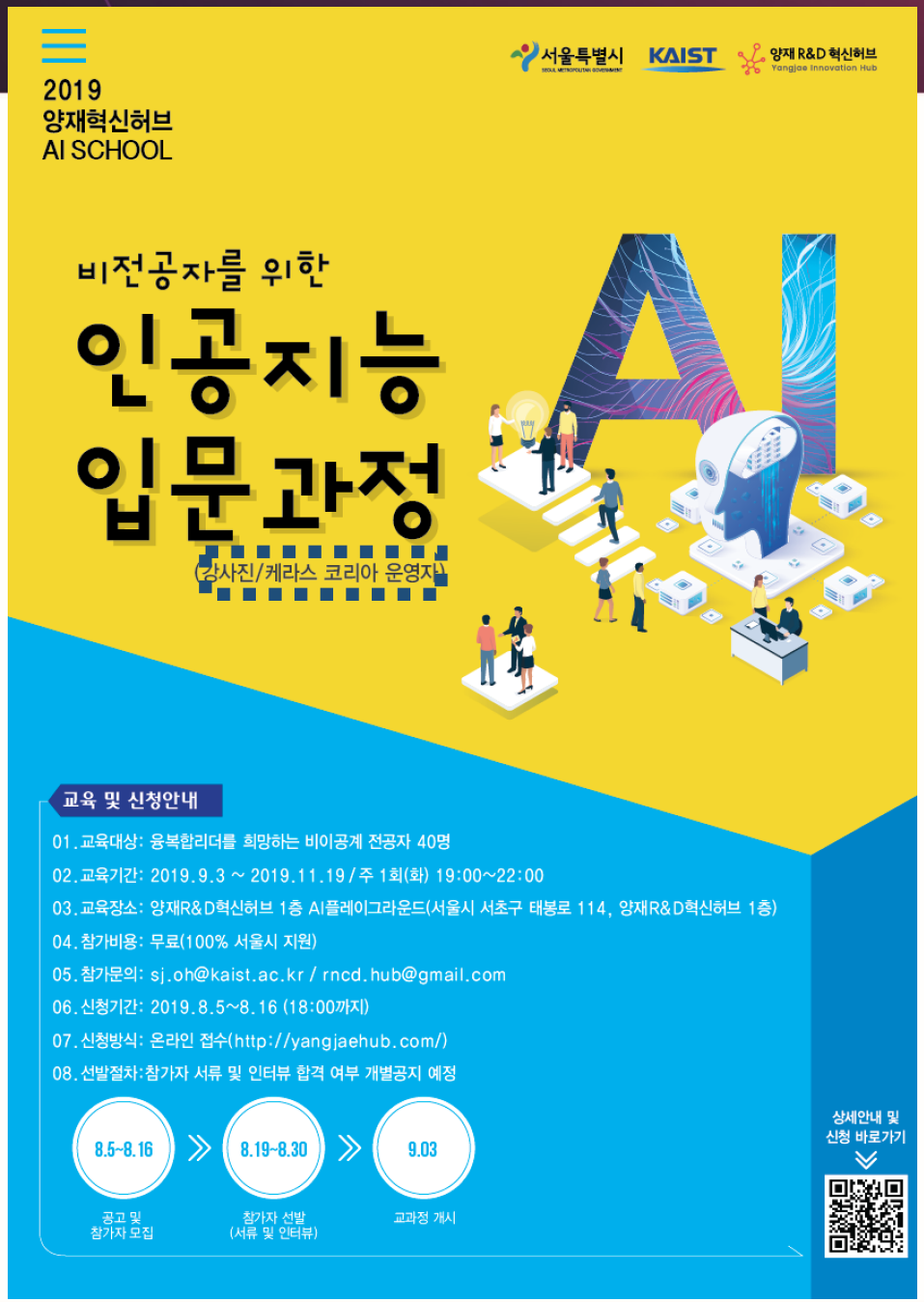
Like Comment Share

함께하는 딥러닝 컨퍼런스

함께하는 딥러닝 컨퍼런스

대전 러닝 데이


#DLD
대전러닝데이
2020년 9월 2일
10:00~18:00
28개 온라인발표
주최 | AIFactory, 유클리드소프트, 마인즈앤컴퍼니
주관 | AI프렌즈, 케라스 코리아, 캐글 코리아, 대덕넷
후원 | 한국마이크로소프트, NIPA, OpenUp, UST




전미정


박정현


이재원


강형석


이상훈


이태영


정연준

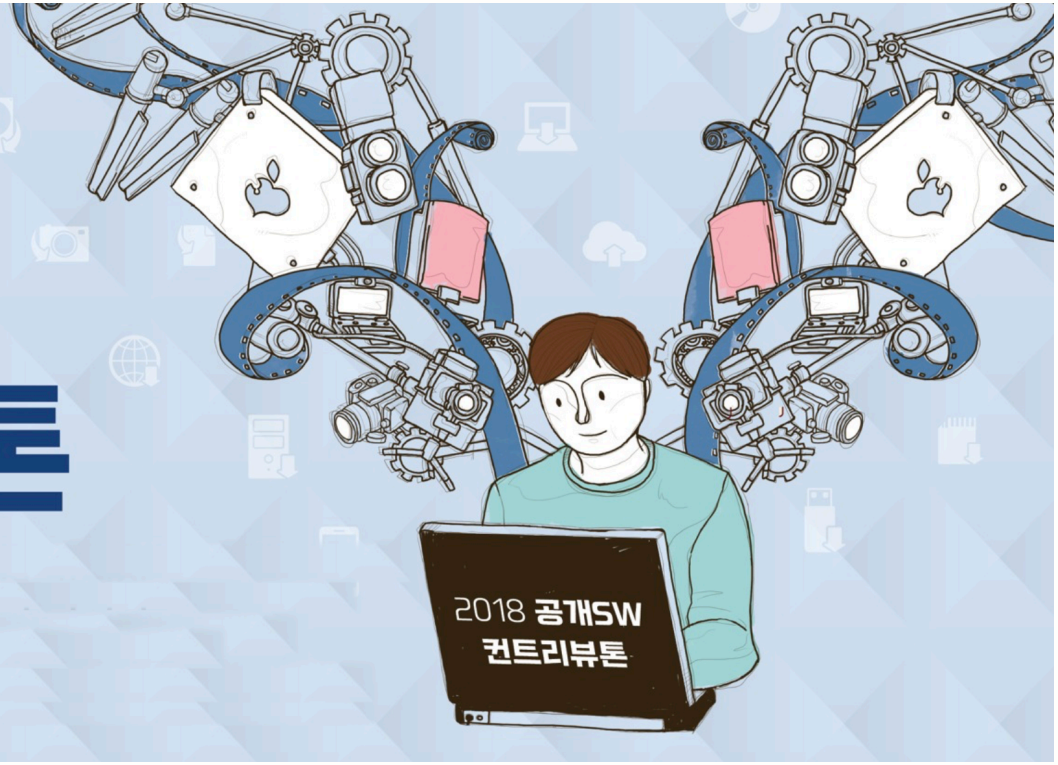

황준원

OSS OPEN SOURCE SOFTWARE

케라스 코리아와 오픈소스 컨트리뷰톤

OSS OPEN SOURCE SOFTWARE

2018 공개SW 컨트리뷰톤



▶ 첫 만남


2018 공개SW 컨트리뷰톤
참여·공유·협업의 공개SW 프로젝트 개발을
직접 경험해보세요!
Contribute to Keras
2018 공개SW
컨트리뷰톤

▶ 우리의 목표

Contribute to K

딥러닝(케라스)을 배울 수 있는 입문서와 영문 문서는 많지만

심화 이론, 실전 코드, 다양한 모델을 배울 수 있는

한글 문서가 부족하니

훌륭한 영문 튜토리얼들을 우리말로 번역해

모두의 손에 딥러닝 툴을 올릴 수 있도록 기여하자!

GitHub

Issue 등록

- 용어 관련 의견을 듣고 싶습니다. discussion good first issue help wanted
  #57 opened on Sep 8 by 5taku — 9
- Keras를 이용한 Realtime_Multi-Person_Pose_Estimation
  #43 opened on Aug 23 by MijeongJeon 2018_Contribut... — 1
- Keras와 Jupyter Notebook으로 시작하는 딥러닝
  #40 opened on Aug 22 by MijeongJeon 2018_Contribut...
- VGGNet, ResNet, Inception, Xception으로 ImageNet 분류하기
  #39 opened on Aug 22 by MijeongJeon 2018_Contribut...
- Keras와 텐서플로우를 이용한 손글씨 인식
  #38 opened on Aug 22 by MijeongJeon 2018_Contribut... — 1
- Keras를 이용한 Hyperparamter 최적화
  #36 opened on Aug 22 by MijeongJeon 2018_Contribut...
- Keras로 Multiple output. multiple loss 모델 설계하기
  #34 opened on Aug 22 by MijeongJeon 2018_Contribut...

Project 관리

2018_Contributhon

Updated 23 days ago

Filter cards

6 To do

- Keras를 이용한 Realtime_Multi-Person_Pose_Estimation
  #43 opened by MijeongJeon
- VGGNet, ResNet, Inception, Xception으로 ImageNet 분류하기
  #39 opened by MijeongJeon
- Keras로 Multiple output. multiple loss 모델 설계하기
  #34 opened by MijeongJeon
- Keras/examples (4): Examples demonstrating specific Keras functionality
  #26 opened by MijeongJeon
- 용어 관련 의견을 듣고 싶습니다.
  #57 opened by 5taku
  discussion good first issue help wanted

12 Doing

- 사전훈련된 단어삽입 모델로 문자 분류 문제 해결하기
  #31 opened by MijeongJeon
- 케라스를 이용한 콘텐츠 기반 이미지 검색
  #11 opened by MijeongJeon
- Keras/examples (1): Vision models examples
  #23 opened by MijeongJeon
- Keras와 Jupyter Notebook으로 시작하는 딥러닝
  #40 opened by MijeongJeon
- Keras와 텐서플로우를 이용한 손글씨 인식
  #38 opened by MijeongJeon
- Keras를 이용한 Hyperparamter 최적화
  #36 opened by MijeongJeon

36 Done

- Keras를 이용한 Multi-GPU 학습
  #37 opened by MijeongJeon
- issue #29
  0 of 3
  #81 opened by ohahohah
  Changes approved
- 샘플이 적은 이미지셋으로도 성능 내기 (transfer learning/fine tuning)
  #27 opened by MijeongJeon
- Keras vs PyTorch 어떤 플랫폼을 선택해야 할까?
  #42 opened by MijeongJeon
  good first issue
- Keras를 이용한 Multi-label 분류
  #35 opened by MijeongJeon

PR(Pull Request)

issue 27 #150

Merged **MijeongJeon** merged 94 commits into `master` from `issue_27` 27 days ago

Conversation 1 | Commits 94 | Checks 0 | Files changed 36

itsnamgyu commented on Oct 18 — Collaborator

리뷰 부탁드립니다!

mike2ox added some commits on Sep 3

- #15 : Title&subjects 한글화 완료. image 삽입 및 code 기입 예정 — 31c4be3
- #15 : 구현 앞 챕터까지 완료. 이미지 하단 한글화, 구현에서 model 내용 한글화 예정 — df16813
- #15 : 용어 통일 및 오타 수정. 구현 단계부터 번역 ... — dff8b2c
- #15 : 원문내 데이터 저장장 — cf01de6
- #15 : 콘텐츠 손실 함수 관련 부분까지 번역. 이미지 3개간의 추가 코멘트가 필요. 스타일 손실 함수부터 이어서 번역 예정정 — 9757598
- #15 : 컨텐츠 손실 함수 수식식 이미지 — aa8c51a

Review

130_뉴럴

89_Gen

...

tykimos commented on Oct 10 Member

왕성한 활동 하신다고 고생많으십니다. 작성하신 GAN 관련 코드에서 봐줬으면 하는 부분이 있습니다. 먼저 아래글 간단하게 읽어주세요~

https://tykimos.github.io/2017/12/12/One_Slide_GAN/

작성 코드에서, 학습 과정에 discriminator.trainable의 플래그를 바꾸는 부분이 나오는데, 이 플래그는 컴파일할 때만 적용되므로 컴파일 이후에는 아무 영향이 없을 듯 합니다. 한 번 확인 부탁드려요~

```
# Discriminator를 학습시킵니다.
discriminator.trainable = True
discriminator.train_on_batch(X, y_dis)

# Generator를 학습시킵니다.
noise = np.random.normal(0, 1, size=[batch_size, random_dim])
y_gen = np.ones(batch_size)
discriminator.trainable = False
```

Merge

429 lines (315 sloc) | 31 KB

Raw | Blame | History

케라스 LSTM 모델로 작곡하기(How to Generate Music using a LSTM Neural Network in Keras)

원문 링크

> 이 문서는 순환신경망(RNN)인 `LSTM`과 Python 음악 툴킷인 `music21`을 이용해 작곡하는 방법을 설명합니다. 2018년 8월을 기준으로, 동작하지 않는 코드는 동작하지 않는 부분을 동작하도록 변형하였기 때문에 코드는 원문과 같지 않을 수 있습니다. 또한 그대로 번역한 것이 아닌 필요한 설명과 합쳐서 다시 쓴 글이기 때문에 원문과 다를 수 있습니다. 모든 이미지들은 원문에서 가져온 이미지입니다. 원문에서 나온 코드의 이해를 돕기 위해 코드에 주석을 붙인 jupyter notebook 파일을 첨부합니다.

- 케라스
- LSTM
- Neural Networks layer

블로그 생성

HOME POSTS CATEGORIES TAGS

Keras for Everyone

KEKOxTutorial

Keras 모델을 REST API로 배포해보기

8 min read • 2018-10-27

Keras 모델을 REST API로 배포해보기 (Building a simple Keras + deep learning REST API)

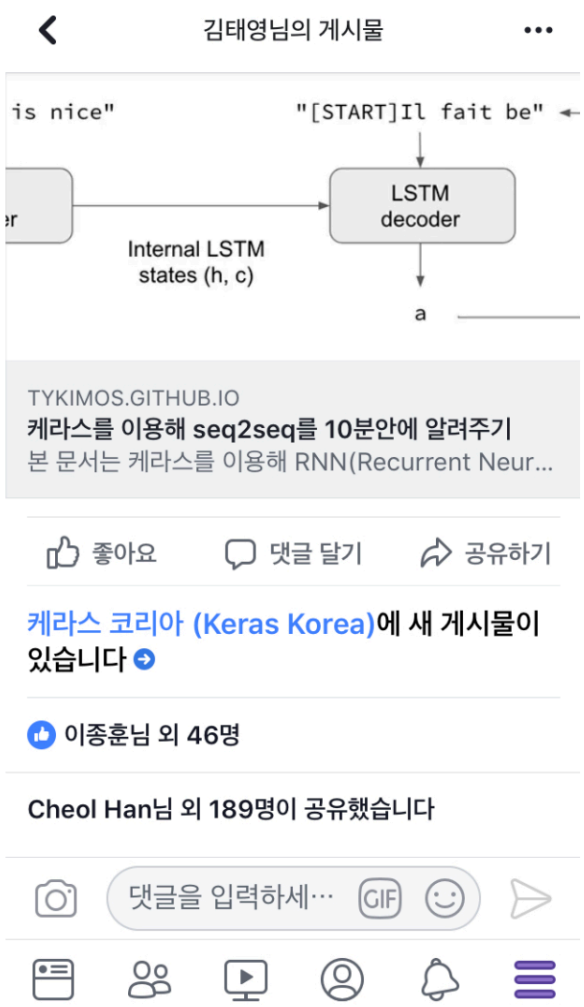
원문

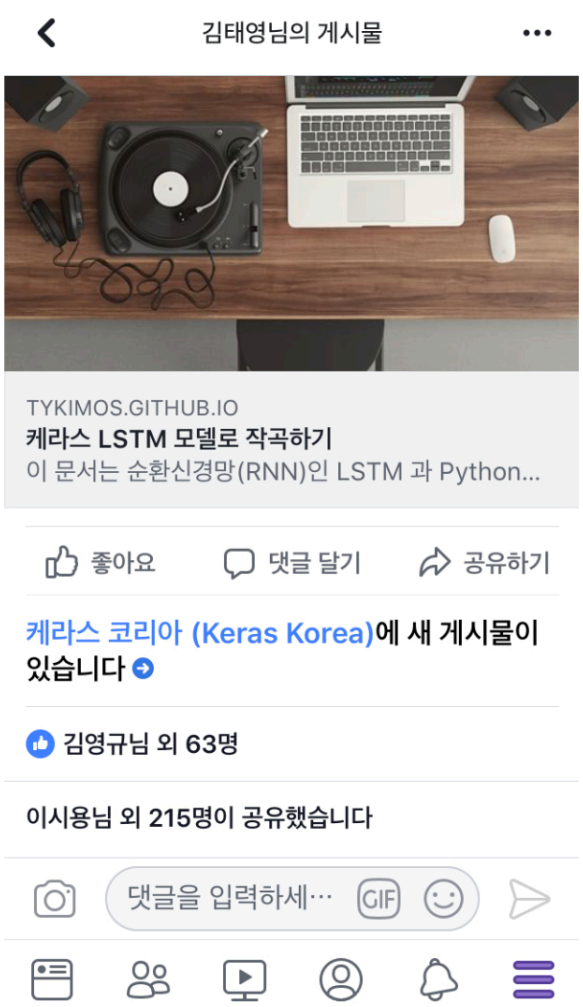
> 이 글은 Adrian Rosebrock이 작성한 안내 게시글로 Keras 모델을 REST API로 제작하는 간단한 방법을 안내하고 있습니다.

Tag Archive

이미지처리	6	CNN	6	Keras	6
LSTM	4	기초	3	Neural Style Transfer	2
NLP	2	GAN	2	eager execution	2
Autoencoder	2	강화학습	2	Classification	2
music21	1	음악	1	신경망	1
OpenCV	1	seq2seq	1	RNN	1
Fashoin-MNIST	1	Embedding	1	Neural Network	1
추천시스템	1	DCGAN	1	Ensemble	1
앙상블	1	ML Kit	1	Android	1
MNIST	1	모바일	1	AlphaZero	1
AlphaGo	1	Monte Carlo	1	Residual Convolutional Network	1
GRU	1	가상화폐	1	Fine Tuning	1
Data Augmentation	1	CycleGAN	1	Style Transfer	1
EyeNet	1	의료	1	TensorFlow	1
게임	1	Q-Learning	1	Gym	1
금융	1	Siraj Raval	1	주식	1
텍스트	1	GPU	1	Kaggle	1
PyTorch	1	Framework	1	Multi-label	1

문서 공유





작업문서 SNS 공유 화면 (총 602회 공유)

▶ 수상

참여·공유·협업의 공개SW 프로젝트 개발방법 체험과
최고급 개발자 멘토로부터 개발 노하우를 경험할 수 있는
2018 공개SW 컨트리뷰톤

2018 공개SW 컨트리뷰톤 수상자 발표

수상자 여러분께 축하의 박수를 보내드리며,
참여·협업의 정신으로 끝까지 수고해주신 모든 멘티, 컨트리뷰터 여러분께 감사의 인사를 드립니다.

훈격	상급	수상팀/개인
과학기술정보통신부장관상	최우수상	Chromium / Blink
정보통신산업진흥원장상	우수상	uftrace
		GitShare
		구름입력기
	장려상	Contribute to Keras(+Tensorflow) Documentation
		it-chain
		캐글 속 한글화 작업을 통한 데이터 사이언스 대중화
		OpenStack 2팀

▶ 컨트리뷰톤 후기

"기계적으로 문장 번역을하는 것이 아니라 쉽고, 정확하게 전달하기 위해 공부하고 고민하면서 딥러닝에 대해서 더 깊이 생각할 수 있는 기회였다"

"원문 블로그에 있는 코드를 직접 따라서 작성하고 실행해보며 한 줄 한 줄 코드를 더 잘 이해할 수 있었다."

"git으로 협업하는 과정에서 Branch가 섞이고, 충돌을 해결하는 시행착오를 겪으며 git 기능과 사용법에 대해 많이 배울 수 있었다."

"작업을 하다가 버그를 발견, 수정해 Keras 자체에 기여하는 성과를 냈다!"

"오픈 소스에 기여하는 자신감이 생겨 다른 딥러닝 관련 기여를 계획하고 있다!"


MICRO SOFTWARE


MICRO SOFTWARE

OSS OPEN SOURCE SOFTWARE

2020 공개SW 페스티벌

www.oss.kr

2019 공개SW 컨트리뷰톤

참여 · 공유 · 개방 · 협업의 공개SW프로젝트에 직접 참여하고
선배 개발자의 가이드와 함께 공개 SW 개발에 대한 진입장벽을 뚫고
다양한 기여(contribution)를 경험할 수 있는 최고의 기회!!!

참가신청 2019.8.5월 – 8.25일


2019
공개SW 컨트리뷰톤 발대식

옵디팀 ObDe

Object
Detection API

전미정 멘토

코코팀 KoKo

우리말
케라스 공식
문서

갓 정연준 멘토

▶(기존) 끊임없는 설치 문제

Import "object_detection/protos/ssd.proto" was not found or had errors.

models: research | stat:awaiting response | stat:awaiting tensorflower

#4538 opened on 14 Jun 2018 by edisonchensy

"protoc --python_out =. .\object_detection\protos\calibration.proto " Error

#6405 opened on 20 Mar by smalgan

▶ (해결) 도커를 활용한 편리한 설치

```
# pull yolk docker image
$ docker pull kerasyolk/yolk
```

▶(기존) 일치하지 않는 버전들

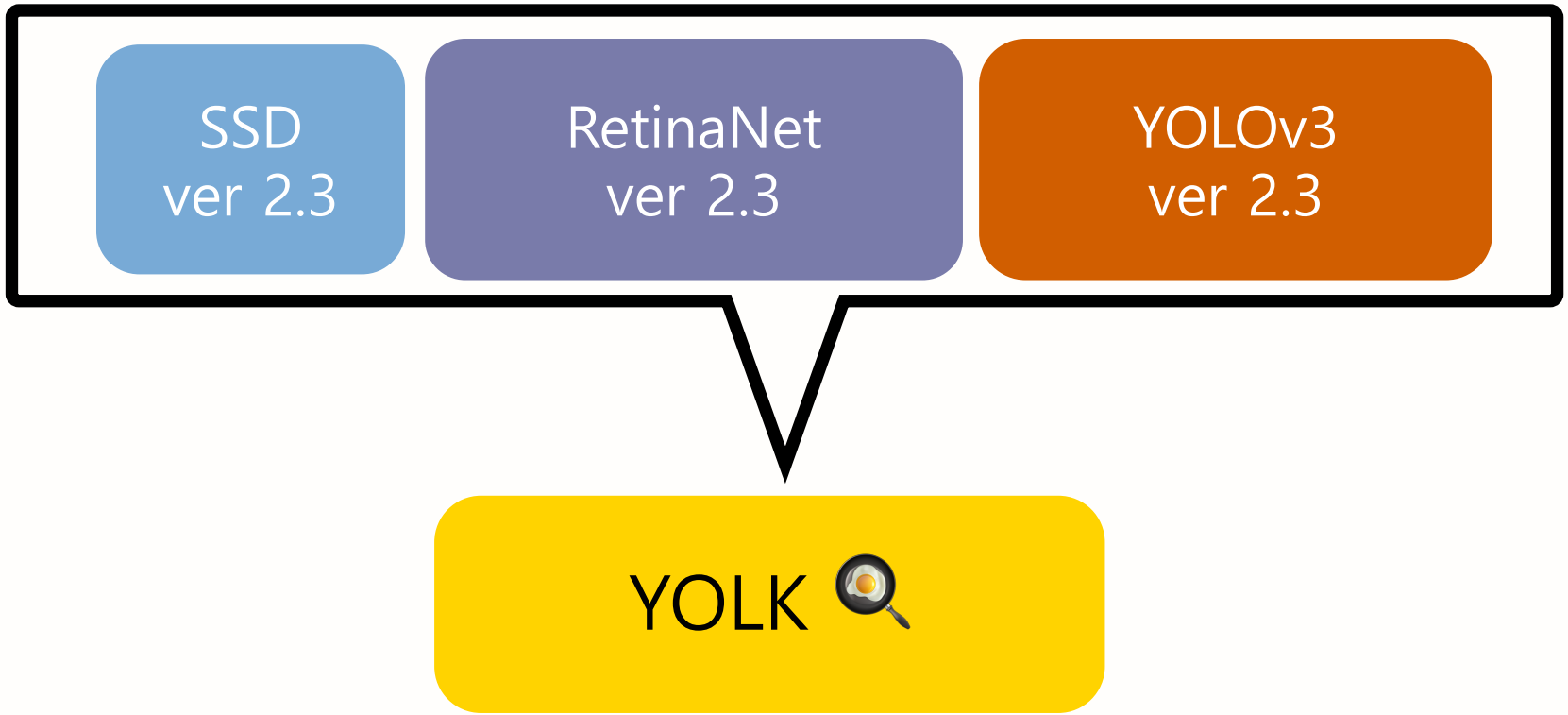
SSD
ver 2.1

RetinaNet
ver 2.3

YOLOv3
ver 1.9

```
$ git clone github/ssd
$ git clone github/retinanet
$ git clone github/yolov3
```

▶(해결) 동일한 Dependency



▶(기존) 이해하기 너무 어려운 코드

```
def run_inference_for_single_image(image, graph):
  with graph.as_default():
    with tf.Session() as sess:
      # Get handles to input and c
      ops = tf.get_default_graph()
      all_tensor_names = {output.n
      tensor_dict = {}
      for key in [
          'num_detections', 'detection_boxes', 'detection_scores',
          'detection_classes', 'detection_masks'
      ]:
        tensor_name = key + ':0'
        if tensor_name in all_tensor_names:
          tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(
              tensor_name)
      if 'detection_masks' in tens
        # The following processing
        detection_boxes = tf.squee
        detection_masks = tf.squee
        # Reframe is required to t
the image size.
        real_num_detection = tf.ca
        detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection, -1])
        detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -1])
        detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
            detection_masks, detection_boxes, image.shape[1], image.shape[2])
        detection_masks_reframed = tf.cast(
            tf.greater(detection_m
        # Follow the convention by
        tensor_dict['detection_mas
            detection_masks_refram
      image_tensor = tf.get_defaul

      # Run inference
      output_dict = sess.run(tensor_dict,
                             feed_dict={image_tensor: image})

      # all outputs are float32 numpy arrays, so convert types as appropriate
      output_dict['num_detections'] = int(output_dict['num_detections'][0])
      output_dict['detection_classes'] = output_dict[
          'detection_classes'][0].astype(np.int64)
      output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
      output_dict['detection_scores'] = output_dict['detection_scores'][0]
      if 'detection_masks' in output_dict:
        output_dict['detection_masks'] = output_dict['detection_masks'][0]
  return output_dict
```

```
ops = tf.get_default_graph().get_operations()
```

```
# The following processing is only for single image
detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
```

```
tensor_dict['detection_masks'] = tf.expand_dims(
    detection_masks_reframed, 0)
image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')
```

▶ (해결) 일관된 keras 스러운 코드

```
model = yolk.detector.load_inference_model(model_path)

model.compile(optimizer=sgd, loss=loss)

y_pred = model.predict(image)

yolk.detector.show_result(image_path, y_pred_thresh)
```


맥윙
콘파이
더욱 든든해진
USA

코코팀 KoKo

우리말
케라스 공식 문서

정연준 멘토

우리말 케라스 공식 문서

문서도 부실해
근데 영어야!

봐도 모르겠다 ...

▶ 내용이 누락된 문서

```
class Tokenizer:
  def fit_on_texts()
  def fit_on_sequences()
  def texts_to_sequences()
  def texts_to_sequences_generator()
  def sequences_to_texts()
  def sequences_to_texts_generator()
  def texts_to_matrix()
  def sequences_to_matrix()
  def get_config()
  def to_json()

def hashing_trick()
def one_hot()
def text_to_word_sequence()
```

Tokenizer 메소드

fit_on_texts

```
fit_on_texts(texts)
```

입력된 텍스트를 바탕으로 단어를 추출, 정수 인덱스를 부여하여 `Tokenizer` 내부에 딕셔너리 형태의 단어 목록을 생성합니다. 이 목록은 `word_index` 메소드로 불러올 수 있습니다. 목록은 전체 단어의 종류 만큼 생성되지만 이 가운데 다른 메소드에서 사용되는 단어 개수는 최대 `Tokenizer` 에서 지정한 `num_words-1` 개입니다(`num_words-1` 개인 까닭은 `0` 번 인덱스는 단어에 배정되지 않기 때문입니다). 기본적으로 문자열로 이루어진 리스트를 텍스트로 입력받으며, 텍스트 리스트가 하위 리스트들로 구성된 경우list of lists 각 하위 리스트들에 포함된 문자열들을 각각 하나의 토큰으로 취급합니다. 이후 `texts_to_sequences` 메소드와 `texts_to_matrix` 메소드를 사용하기 위해서는 먼저 `fit_on_texts` 메소드로 단어 목록을 생성해야 합니다.

인자 - texts: 문자열의 리스트, (메모리 절약을 위한)문자열의 제너레이터, 또는 문자열 리스트로 이루어진 리스트.

---

fit_on_sequences

```
fit_on_sequences(sequences)
```

단어 인덱스로 이루어진 리스트를 입력받아 `Tokenizer` 내부에 인덱스 목록을 생성합니다. `fit_on_text` 를 따로 사용하지 않는 경우, 이후 `sequences_to_matrix` 메소드를 사용하려면 먼저 `fit_on_sequences` 메소드를 실행해야 합니다.

인자 - sequences: 순서형 데이터의 리스트. 여기서 리스트 내의 각 순서형 데이터는 해당 문장의 단어 순서에 따라 각 단어의 인덱스를 나열한 리스트입니다.

설명이 너무 부족한 문서

hashing_trick

```
keras.preprocessing.text.hashing_trick(text, n, hash_function=None, f
```

Converts a text to a sequence of indexes in a fixed-size hashing space.

Arguments

- **text**: Input text (string).
- **n**: Dimension of the hashing space.

hashing_trick

```
keras.preprocessing.text.hashing_trick(text, n, hash_function=None, filt
', lower=True, split=' ')
```

텍스트에 일반적인 정수 대신 지정한 크기의 해시공간에 맞추어 인덱스를 부여합니다. 이때 해시 함수의 충돌로 인해 서로 다른 단어가 같은 인덱스 값을 갖는 경우가 생길 수 있습니다. 충돌 확률은 해시 공간의 차원과 개별 객체의 개수에 따라 달라집니다. `0` 은 어떤 단어에도 배정되지 않는 예비 인덱스입니다.

인자

- **text:** `str`. 텍스트 입력.
- **n:** `int`. 해시 공간의 차원. 해싱 함수를 통해 부여하는 인덱스의 최댓값을 지정합니

기능을 알 수 없는 문서

Keras Documentation

Search docs

Docs » Losses

Available loss functions

mean_squared_error

```
keras.losses.mean_squared_error(y_true, y_pred)
```

mean_absolute_error

```
keras.losses.mean_absolute_error(y_true, y_pred)
```

mean_absolute_percentage_error

```
keras.losses.mean_absolute_percentage_error(y_true, y_pred)
```

mean_squared_logarithmic_error

```
keras.losses.mean_squared_logarithmic_error(y_true, y_pred)
```

사용 가능한 손실 함수

mean_squared_error

```
keras.losses.mean_squared_error(y_true, y_pred)
```

예측값과 목표값의 평균 제곱 오차(MSE, mean squared error)를 계산합니다.

`(square(y_pred - y_true))/len(y_true)`

인자

- **y_true**: 목표값 텐서.
- **y_pred**: 예측값 텐서.

반환값

샘플당 하나의 스칼라 손실값 텐서.

mean_absolute_error

```
keras.losses.mean_absolute_error(y_true, y_pred)
```

예측값과 목표값의 평균 절대 오차(MAE, mean absolute error)를 계산합니다.

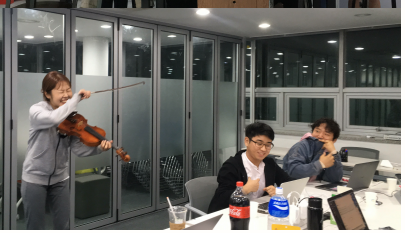
`(abs(y_pred - y_true))/len(y_true)`

인자

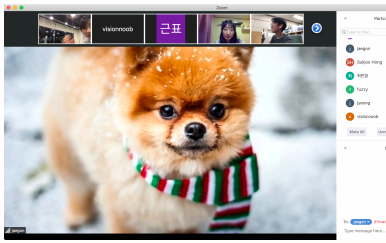
- **y_true**: 목표값 텐서.
- **y_pred**: 예측값 텐서.

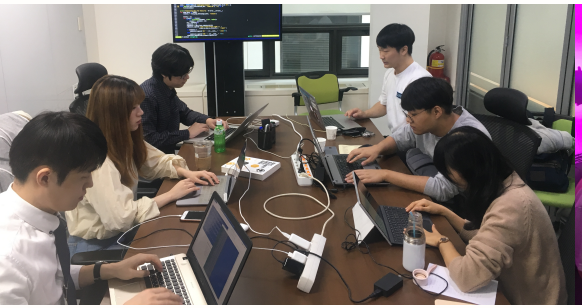




10회 오프 모임
20회 랜선 모임

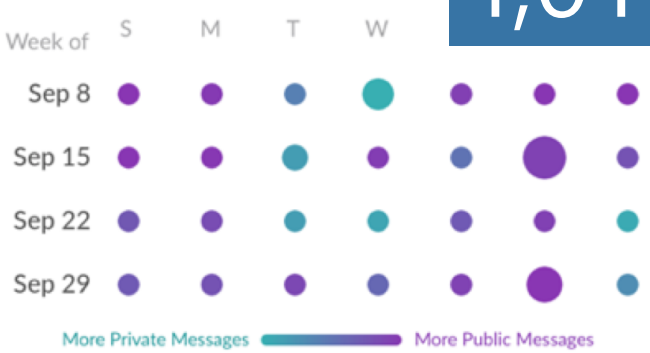



Your members sent a total of **3,730 messages** last week (that's 2,072 more than the week before). Of those, **60% were in public channels** and **40% were direct messages**. Your membe
files (that's 50 more than the week before).

3,730

Your members sent a total of **4,012 messages** last week (that's 500 more than the week before). Of those, **74% were in public channels** and **26% were direct messages**. Your membe
files (that's 67 more than the week before).

4,012

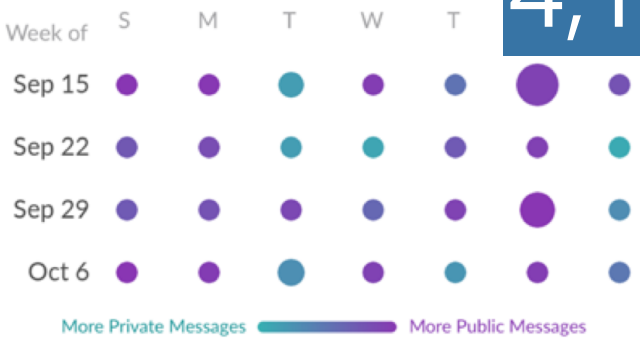


Your members sent a total of **4,316 messages** last week (that's 586 more than the week before). Of those, **68% were in public channels** and **32% were direct messages**. Your members als
(that's 8 fewer than the week before).

4,316

Your members sent a total of **4,170 messages** last week (that's 158 more than the week before). Of those, **62% were in public channels** and **38% were direct messages**. Your members also uploaded **91 files** (that's 18 fewer than the week before).

4,170



keras-team / **keras**

keras-team / **keras-docs-ko**

- PR 70회
- 커밋 272개
- 라인 5334
- 욜커 10명

KerasKorea / **KerasObjectDetector**

- 이슈 42건
- 커밋 54개
- PR 24개
- 욜커 11명

2020 공개SW 페스티벌

2019 공개SW 컨트리뷰톤 프로젝트 수상팀 발표

대상	과학기술정보통신장관상	uftrace
최우수상	정보통신산업진흥원장상	YOLK(You Only Look Keras) RustPython/gpython

아동 · 청소년 〉 생계 및 기초생활지원

보호 종료 아동이라는 말을 들어보신 적이 있으신가요

모금함 소개 | 기부자 현황 3,410 | 모금 소식 0

✓ 네티즌 기부내역 기업 기부내역 후원금액 10,340,400 원

2020.08.14 - 2020.09.06 검색 초기화

총 3,405개의 네티즌 기부내역과 응원메시지가 있습니다.

보호 종료아동들에게 조금이나마 도움이 되었으면 좋겠습니다 -케라스 코리아 욜크팀 일동 (전미정 외 21인) (전미정, 정연준, 김민우, 김준영, 박진이, 유원상, 이혜리, 전지영, 정성훈, 조휘용, 최민영, 황성주, 김수정, 김현규, 박근표, 박아정, 박정현, 송문혁, 이설희, 이재원, 임재곤, 홍석주)

2020.09.06 16:11

제라님 1,100,000 원 후원

104%

2020.08.14 ~ 2020.09.06까지

D-day

10,340,400 원

네티즌 10,340,400 원
기업 0 원

9,900,000 원 목표

기부하신 금액은 수수료 없이 100% 전달됩니다

정기기부 신청 시, 매달 10%를 기부콩으로 적립해드립니다

모금단체
대한사회복지회

세상의 모든 아이들이 행복했으면 좋겠습니다

단체안내 ›

해피로그 가입	2005년
총 모금액	986,346,700원
필수 모금후기 등록률	83%
현재 정기기부자	69명

공유 & 성장

with 오픈소스, 커뮤니티