nlp4kor

Deep Learing for NLP: A to Z

이 문서는 초보가 제작하였습니다.

bage79@gmail.com

What do you need?

- Image (Video)
- Sound (Voice)
- Smell (?)
- Natural Language
- Inference (New Knowledge)
- Game (Strategy)
- Emotion
- And more…
    - Robotics, …

Study …

nlp4kor

- **Do your-self**
- **Study many Korean blogs & papers on the internet**
  - 모두의 딥러닝
    - https://hunkim.github.io/ml/
  - 페이스북 텐서플로우 코리아
    - https://www.facebook.com/groups/TensorFlowKR/

Sample Source Code

- **DeepLearningZeroToAll**
  - https://github.com/hunkim/DeepLearningZeroToAll
- **Tensorflow-101**
  - https://github.com/sjchoi86/Tensorflow-101
- **Googling…**

Utilities

- numpy
- pandas
- konlpy
- gensim
- bage_utils
  - https://github.com/bage79/nlp4kor/tree/master/bage_utils

Raw Data

- **Buy**
- **Obtain from your friends/collegues**
- **Make yourself**
- **Crawl the web**
  - requests
  - selinium
  - beautifulsoup4
  - lxml

Storing Text Data

- **File**
  - text, binary, object(pickle)
- **RDBMS**
  - MySQL, …
- **NoSQL**
  - MongoDB, …
- **Search Engine**
  - Elasticsearch, …

Convert to Dataset

- **File with gzip format**
  - 1/10 file size on text data
- **text or one-hot-vector or vector**
  - csv, tsv, pkl, npy, h5(HDF5) …
  - (e.g.) text file
    - features(input), labels(output) on one line
- **Similarly Distribution**
  - train set, validation set, test set

Coding Environment

- Python3 or Python2
- Anaconda or Individual package
- Tensorflow or Keras
- Ubuntu or Mac or Windows
- Native or Docker or AWS
- CPU or GPU
- CPU(GPU) clock or RAM size
- Github or Gitlab or Bitbucket or Google Drive

Coding Environment

nlp4kor

- **vi (vim), Atom, Eclipse…**
- **Jupyter Notebook (Ipython)**
  - for pilot program or presentation
  - Draw images, plots, dataframes…
  - Run on remote machine
- **Pycharm + Remote Interpreter**
  - for service
  - source navigating
  - Mac client + Ubuntu server (with GPU)

Modeling

- **Sparse vector or Dense vector**
  - One-hot-vector or Word2vec
- **Regression or Classification or Clustering**
- **FFNN**
- **CNN**
- **RNN**
- **RL**
- **GAN**
- **…**

Hyper-parameters

- **Optimizer**
  - AdamOptimizer
- **Regularizer**
  - l1_regularizer, l2_regularizer
- **Variable Initializer**
  - random_normal, truncated_normal, xavier_initializer, …
- **Activation functions**
  - tanh, sigmoid, relu, celu(?)…
- **Cost function**
  - Root Mean Square Error, softmax_crossentropy
- **Batch size, Total epochs, Learning rate, Dropout**
- **Input window size, Hidden size, Layers, …**

Testing with train set

- **Your model do work?**
  - use small train set, first
  - one character -> one word -> one sentence
- **Predicting Node**
  - output for test
- **Monitor overfitting**
  - observe cost of train set.
  - decide total epochs & learning rate.

Training (Long time)

- **Batch + background job**
- **Queue**
- **Logging**
  - tensorboard
- **Resources monitoring**
  - CPU, GPU, RAM
- **Push notification**
  - when all job is done

Testing with test set

- **Train set / Validation set / Test set**
    - with big train set
- **Visualize test result**
    - text
    - performance graph
- **Compare test results**
    - model
    - hyper-parameters

As a service

- **Your model is applicable to service?**
  - Accuracy or Precision, …
  - Throuput (speed)
  - CPU & GPU Memory Usage
  - Reusability (with data cache)
  - Flexibility (for various purposes and target languages)

As a service

- **For Demo (Web Interface)**
    - Bokeh
- **Restful API**
- **RPC API**