Deep Transfer Learning for Classification

Shuyi Wang

Dec 2018

Introduction

Outline

- Classification
- Machine Learning for Classification
- Deep Learning for Classification
- Deep Transfer Learning for Classification

Classification

Definition

- Jacob, E. K. (2004). Classification and categorization: a difference that makes a difference.

- > Classification as process involves the orderly and systematic assignment of each entity to **one and only one** class within a system of **mutually exclusive** and **non overlapping** classes.

Application

Approaches

- human
- machine

human

- labor intensive
- cannot handle big data

machine

- rule-based system
- machine Learning

rule based system

- problem
  - machines are dumb
- human needs give them clear rules
  - expert system

Sometime doesn’t work

- if rule is not clear
- or simply wrong

example of failure

- Plato was applauded for his definition of man as a featherless biped
- Diogenes the Cynic “plucked the feathers from a cock, brought it to Plato’s school, and said, ‘Here is Plato’s man.’”

What if

- we cannot describe the rules
- let machine learn from data
- machine learning

Machine Learning for Classification

Machine learning

- Definition:
- Machine learning (ML) is the study of algorithms and mathematical models that computer systems use to progressively improve their performance on a specific task. (Wikipedia)

Illustration

- Source: https://goo.gl/bk6dUi

Different types

- Supervised
- Unsupervised
- Reinforcement

Supervised

- Labels need to be provided
- eg: classification

Unsupervised

- No Label needed
- eg: clustering

Comparison between them

exercise on clustering

- How to cluster / group the following animals?
- chicken, goldfish, goose, duck

Answer for Clustering

- {chicken, goose, duck} {goldfish}
- {chicken}, {goldfish, goose, duck}

exercise on classifciation

- How to classify the following animals?
- chicken, goldfish, goose, duck

Answer for classification

- You cannot do it
- No label provided!

Classification with Machine Learning

- Decision Tree
- Logistic Regression
- SVM
- Random Forest

Decision Tree

- Example: Titanic

Data of Titanic Passengers

The trained tree

-
- Source: https://goo.gl/YyagUZ

Decision Tree in Scikit-learn

```
from sklearn.datasets import load_iris
from sklearn import tree
iris = load_iris()
clf = tree.DecisionTreeClassifier()
clf = clf.fit(iris.data, iris.target)
```

- Source: https://goo.gl/dhQpTP

Logistic Regression

-
- Source: https://goo.gl/GXzumH

How it works

Logistic Regression in Scikit-learn

```
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
X, y = load_iris(return_X_y=True)
clf = LogisticRegression(random_state=0, solver='lbfgs', multi_class='multinomial').fit(X, y)
```

- Source: https://goo.gl/447efU

Problems of classical machine learning

- Can not handle complex task
    - hard to get features
    - non-linear separable

hard to get features

Feature Engineering

out-of-date

Example

- Please hand-craft feature for the following image of a panda:
-

Test if it works

- Try on the following images

non-linear separable

Let us play

- Tensorflow Playground

Link to Tensorflow Playground

- https://playground.tensorflow.org

Solution

- deep learning

Deep Learning for Classification

definition

- Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

Essentials for Deep Learning

- Data
- Structure
- Loss
    - evaluate if model works well

Data

- Tabular
- Image
- Sequencial (e.g., text)

Structure

- Dense Neural Network
- Convolutional Neural Network
- Recurrent Neural Network
- Combination of them

Examples

- Image Classification
- Text Classification

Image Classification

- End to end learning
- No feature engineering needed
- very complex structure

Text Classification

- Sequence as Input
- Judging from the current input and the state changed by previous inputs
- Recurrent Neural Network

Problem of Deep Learning

- Require a lot of data
- Easy to suffer from **Overfitting**
  - memorize the results, instead of learn the rules
- A lot of computing power needed (GPU/TPU)
- Time consuming

GPU

- Expensive
-

Example

- BERT (Bidirectional Encoder Representations from Transformers)
- State of the art of Language Modeling
- 4 TPU pods (**256 TPU** chips) in 4 days!

Deep Transfer Learning for Classification

Transfer

- People who has got plenty of computing power train a model from scrach
- release the model to public
- Others can borrow the model
- chop off several layers (near the output), replace with their own ones
- Train the new model on their tasks
- with small amount of data

Example of Image Classification

model we borrow

- imagenet

Our task

- Walle vs Doraemon

Experiment

- training set: 140 images for each class, 280 in total
- valid set: 16 images for Doraemon, 17 images for Walle
- Env: fast.ai 1.0 + Pytorch 1.0

Code

```
path = Path('imgs')
data = ImageDataBunch.from_folder(path, test='test', ds_tfms=get_transforms(), size=224)
learn = ConvLearner(data, models.resnet34, metrics=accuracy)
learn.fit_one_cycle(1)
```

Result

Time

-

Example of Text Classification

- Can we do transfer learning on text classification?
- Yes

language models

pretrained model we borrow

- WikiText-103 (WT103)

Experiment

- task
  - sentiment analysis
- data
  - Yelp reviews Polarity

State of the Art

-
- Source: Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., … & Carin, L. (2018). Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. arXiv preprint arXiv:1805.09843.

Code

```
data_lm = TextLMDataBunch.from_csv(path, valid='test')
data_clas = TextClasDataBunch.from_csv(path, valid='test', vocab=data_lm.train_ds.vocab)
learn = RNNLearner.language_model(data_lm, pretrained_fnames=['lstm_wt103', 'itos_wt103'],
  drop_mult=0.5)
learn.fit_one_cycle(1, 1e-2)
learn.unfreeze()
learn.fit_one_cycle(1, 1e-3)
learn.save_encoder('ft_enc')
learn = RNNLearner.classifier(data_clas, drop_mult=0.5)
learn.load_encoder('ft_enc')
learn.fit_one_cycle(1, 1e-2)
learn.freeze_to(-2)
learn.fit_one_cycle(1, slice(5e-3/2., 5e-3))
learn.unfreeze()
learn.fit_one_cycle(1, slice(2e-3/100, 2e-3))
```

Result

Tutorials

Links

- Deep Learning with Python, Part 0: Setup Fast.ai 1.0 on Google Cloud
- Deep Learning with Python and fast.ai, Part 1: Image classification with pre-trained model
- Deep Learning with Python and fast.ai, Part 2: NLP Classification with Transfer Learning

Questions?

The End

Thanks for your time!