NLP Competence group

Quora Insignificance Challenge

Quora Challenge

The quora challenge comes from Kaggle.

The challenge is to classify a post by its title if it’s sincere or insincere. This competition poses a lot of challenges.

- Set time limit on GPU/CPU (2/6h)
- Humans agree about the class in 80 % of the cases making it a rather hard task
    - This means that if we get above 80 % on training data we’re certainly overfitting the model
- As mentioned earlier, text is a fun but hard problem as the combinations are infinite really.

Today

Quick walk-through of:

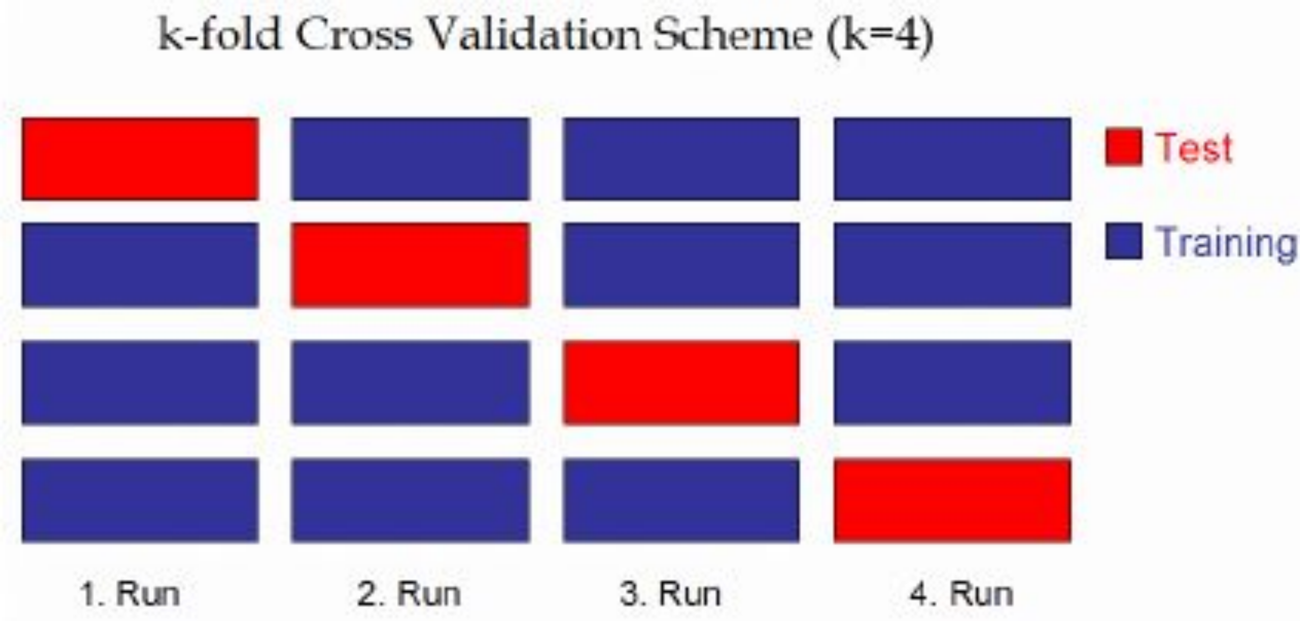
- **Validation** Kfold vs Split
- **Preprocessing** why is it important
- **Environment** and how we use it
- **Kaggle** how to use it

Splitting & K-folding during training

When training your model it’s very important to NEVER use test data.

To be able to actually evaluate your model you’ve to split your training data into train & validation. This is often done in 80/20.

To improve upon basic splitting you start to use “Cross validation” whereas one of the more common one is K-fold cross validation



k-fold Cross Validation Scheme (k=4)

Preprocessing

Time & time

Preprocessing “unique to” non-neural networks

- Want to decrease the dimensions of the input
    - To not “overflow with information”

- Stemming & lemming
    - Find the root word of the word
        - am, are, is -> be
        - car, car’s, cars -> car
- Remove uncommon words
- Lower case
- & more

Preprocessing common to all approaches

- Want to increase coverage

- Spell correction of text
- Tokenization
- De-contraction
- For word-embeddings:
  - Investigate how to increase coverage (common is to replace numbers with # as that's done in most embeddings)
- Clean text out of some really uncommon things, such as special punctuation
- & more

Transformation (quickly)

- Transform text to a format available for computers

- TfIdf
- Number misspellings
- Word Embeddings
- N-grams

Environment

I’ve built multiple baselines for you to expand upon to make prototyping simpler

1. spaCy
2. keras
3. sklearn (from the first meeting)

(4. fastText)

2 & 3 are built in the same manner.

As spaCy is built for use in the industry it’s a bit more locked down in how to use and was not worth the time to adjust to work with 2 & 3 “style”

The whole process

Input ->

Preprocess ->

Transform ->

Fit & Train ->

Evaluate ->

Repeat till happy ->

Submit ->

Neural Network Baseline

Input ->

Embedding ->

Dropout ->

BiRNN ->

BiRNN ->

Maxpool ->

Dense ->

Dense (half size) ->

Dense (sigmoid)

Kaggle

What ruined my day

Kaggle

Kaggle is kind enough to set up an environment for you, if Python either in the form of a Notebook or .py-script.

They’re not kind enough to let you upload your environment easily…

Howevery, they’ve the environment of Python with the most common packages fixed. They’ve the data available simplifying the flow.

In my opinion it’s nice to have things locally also

Let’s go!

(demo-time)