Text Classification

Kim, Ki Hyun and June Oh
FastCampus
2018.10.06

Index

- Problem Define
- Naïve Bayes
- Text Classification with RNN (Bi-LSTM)
- Text Classification with CNN [Kim 2014]
- Exercise

Objective

- Understand neural network architectures for text classification.

Problem Define

$$P(Y = c|X = w_1, w_2, \cdots, w_n)$$

Naïve Bayes

$$\underbrace{P(Y|X)}_{posterior} = \frac{\overbrace{P(X|Y)}^{likelihood}\overbrace{P(Y)}^{prior}}{\underbrace{P(X)}_{evidence}}$$

수식	영어 명칭	한글 명칭
P(Y|X)	Posterior	사후 확률
P(X|Y)	Likelihood	가능도(우도)
P(Y)	Prior	사전 확률
P(X)	Evidence	증거

MAP vs MLE

$$\hat{y}_{MAP} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, P(Y = y|X)$$

vs

$$\hat{y}_{MLE} = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, P(X|Y = y)$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, P(Y|X, \theta)$$

MAP vs MLE

$$P(Y = male|X = 155) \propto P(X = 155|Y = male)P(Y = male)$$

VS

$$P(X = 155|Y = male)$$

Naïve Bayes

$$P(Y = c) \approx \frac{\text{Count}(c)}{\sum_{i=1}^{|\mathcal{C}|} \text{Count}(c_i)}$$

$$P(w|c) \approx \frac{\text{Count}(w, c)}{\sum_{j=1}^{|V|} \text{Count}(w_j, c)}$$

Naïve Bayes

$$
\begin{aligned}
P(Y = c|X = w_1, w_2, \cdots, w_n) &\propto P(X = w_1, w_2, \cdots, w_n|Y = c)P(Y = c) \\
&\approx P(w_1|c)P(w_2|c)\cdots P(w_n|c)P(c) \\
&= \prod_{i=1}^{n} P(w_i|c)P(c)
\end{aligned}
$$

$$
\begin{aligned}
\hat{c}_{MAP} &= \underset{c \in \mathcal{C}}{\operatorname{argmax}}\, P(Y = c|X = w_1, w_2, \cdots, w_n) \\
&= \underset{c \in \mathcal{C}}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|c)P(c)
\end{aligned}
$$

Naïve Bayes

$$
\begin{aligned}
P(Y = c|X = w_1, w_2, \cdots, w_n) &\propto P(X = w_1, w_2, \cdots, w_n|Y = c)P(Y = c) \\
&\approx P(w_1|c)P(w_2|c)\cdots P(w_n|c)P(c) \\
&= \prod_{i=1}^{n} P(w_i|c)P(c)
\end{aligned}
$$

Why?

$$
\begin{aligned}
\hat{c}_{MAP} &= \underset{c \in \mathcal{C}}{\operatorname{argmax}}\, P(Y = c|X = w_1, w_2, \cdots, w_n) \\
&= \underset{c \in \mathcal{C}}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|c)P(c)
\end{aligned}
$$

Naïve Bayes

$$\mathcal{C} = \{pos, neg\}$$
$$\mathcal{D} = \{d_1, d_2, \cdots\}$$

$$P(pos|I, am, happy, to, see, this, movie, !) = \frac{P(I, am, happy, to, see, this, movie, !|pos)P(pos)}{P(I, am, happy, to, see, this, movie, !)}$$
$$\approx \frac{P(I|pos)P(am|pos)P(happy|pos)\cdots P(!|pos)P(pos)}{P(I, am, happy, to, see, this, movie, !)}$$

$$P(happy|pos) \approx \frac{\text{Count}(happy, pos)}{\sum_{j=1}^{|V|} \text{Count}(w_j, pos)}$$
$$P(pos) \approx \frac{\text{Count}(pos)}{|\mathcal{D}|}$$

Naïve Bayes

$$P(neg|I, am, happy, to, see, this, movie, !) = \frac{P(I, am, happy, to, see, this, movie, !|neg)P(neg)}{P(I, am, happy, to, see, this, movie, !)}$$
$$\approx \frac{P(I|neg)P(am|neg)P(happy|neg)\cdots P(!|neg)P(neg)}{P(I, am, happy, to, see, this, movie, !)}$$

$$P(happy|neg) \approx \frac{\text{Count}(happy, neg)}{\sum_{j=1}^{|V|} \text{Count}(w_j, neg)}$$

$$P(neg) \approx \frac{\text{Count}(neg)}{|\mathcal{D}|}$$

Add One Smoothing

$$P(happy|neg) \approx \frac{\text{Count}(happy, neg)}{\sum_{j=1}^{|V|} \text{Count}(w_j, neg)} = 0,$$

$$\text{where Count}(happy, neg) = 0.$$

Add One Smoothing

$$P(happy|neg) \approx \frac{\text{Count}(happy, neg)}{\sum_{j=1}^{|V|} \text{Count}(w_j, neg)} = 0,$$

$$\text{where Count}(happy, neg) = 0.$$

$$\tilde{P}(w|c) = \frac{\text{Count}(w, c) + 1}{\left(\sum_{j=1}^{|V|} \text{Count}(w_j, c)\right) + |V|}$$

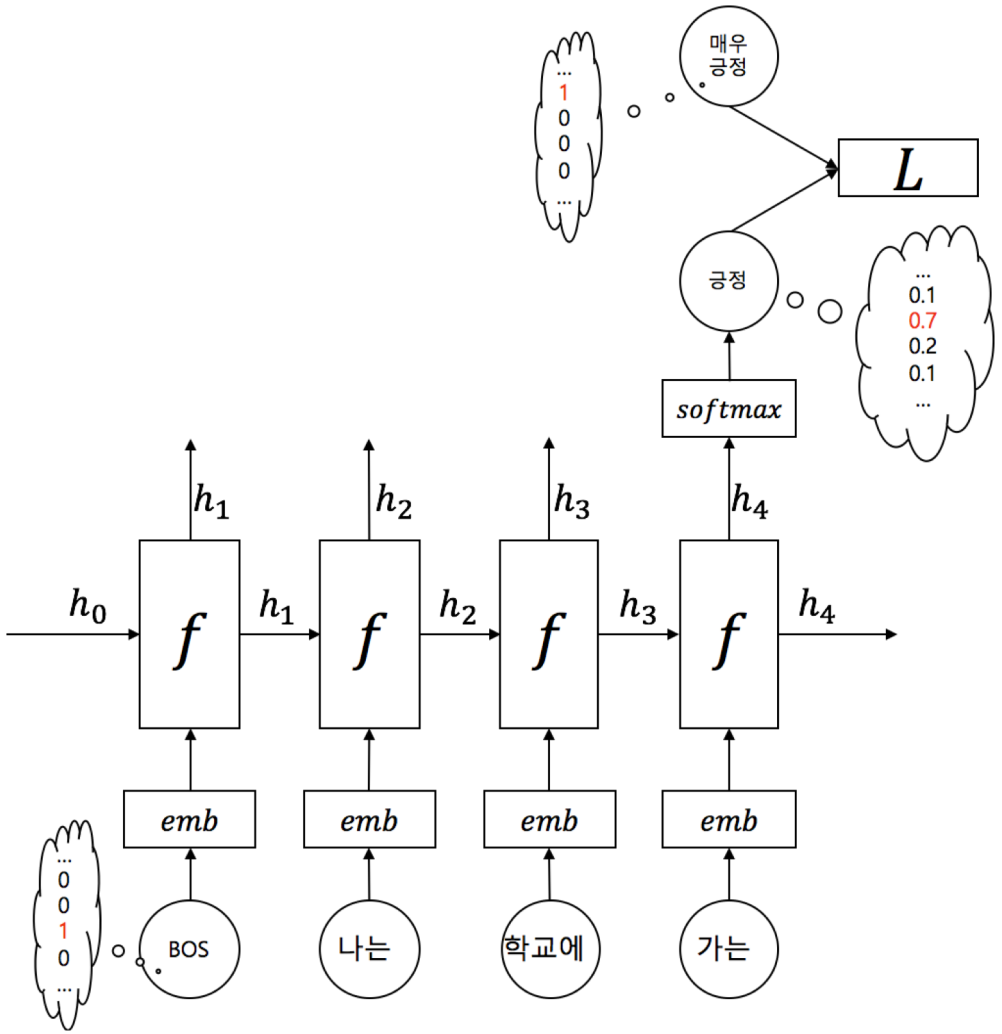
Naïve Bayes

- Pros and Cons
  - Easy and efficient, but fragile.
  - Straightforward because it is based on count.

$$P(pos|I, am, not, happy, to, see, this, movie, !)$$
$$P(neg|I, am, not, happy, to, see, this, movie, !)$$

$$P(not, happy) \neq P(not)P(happy)$$

Text Classification with RNN (Bi-LSTM)



Convolutional Neural Networks

- Convolution Filter

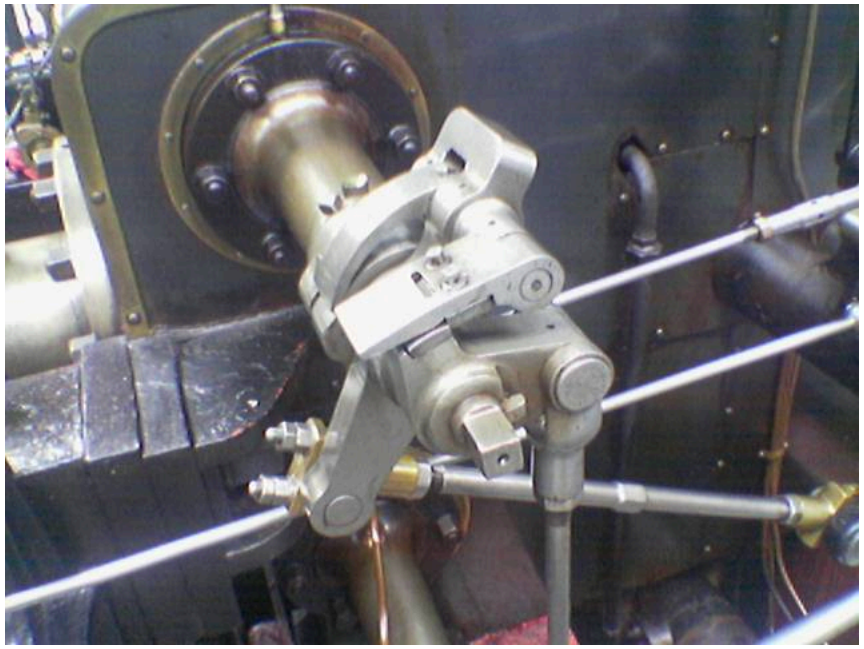
-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Convolutional Neural Networks

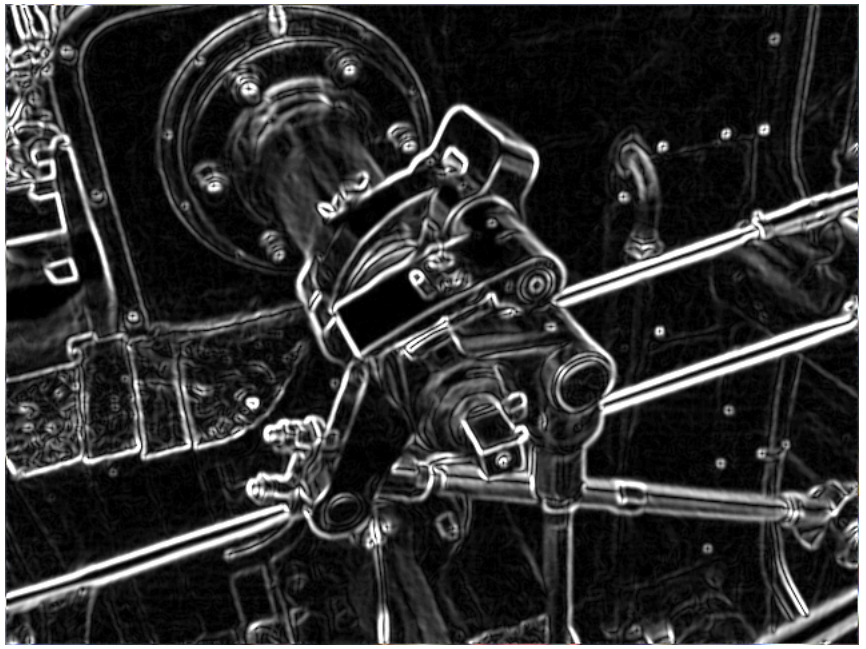


-1	0	+1
-2	0	+2
-1	0	+1

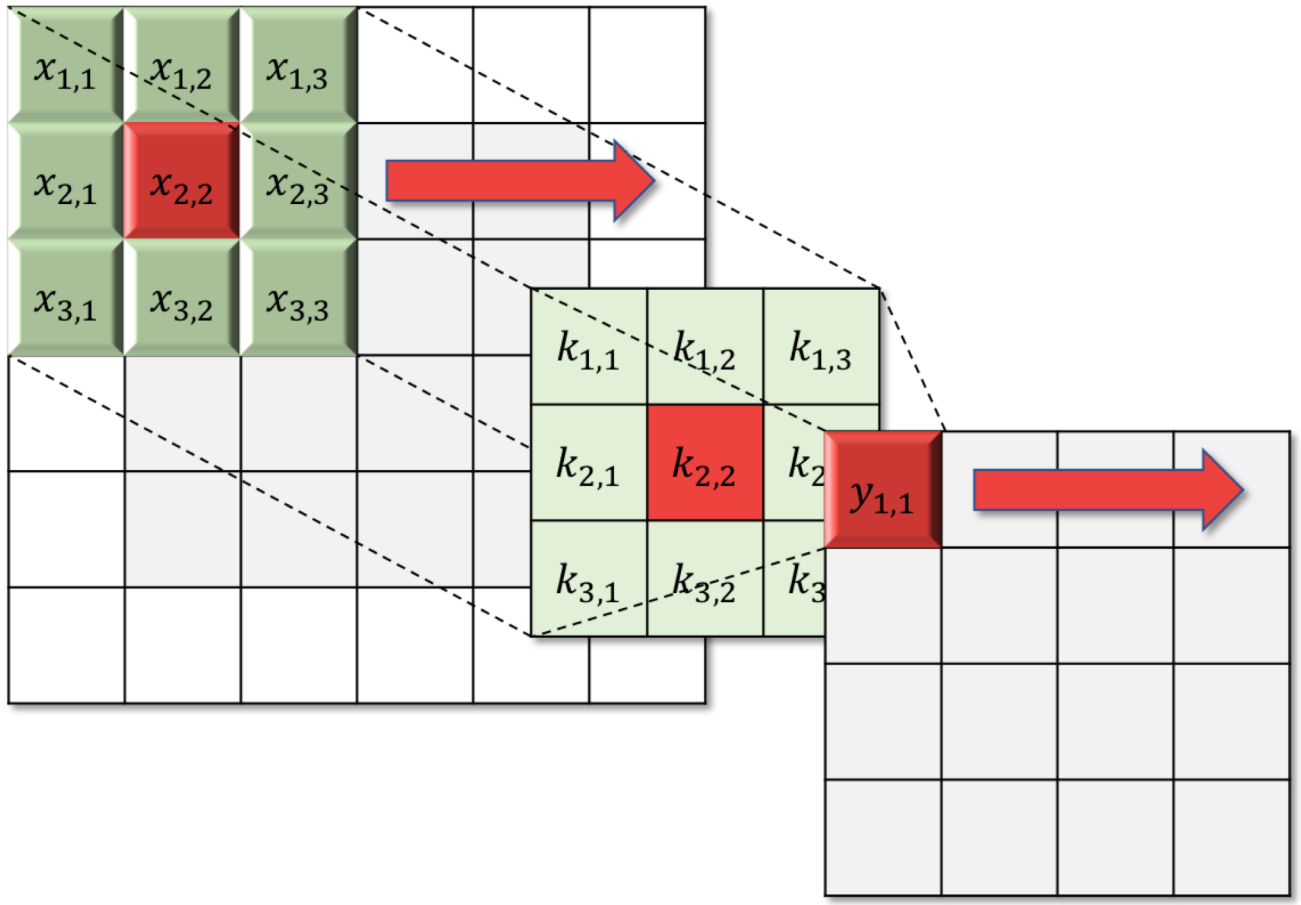
Gx

+1	+2	+1
0	0	0
-1	-2	-1

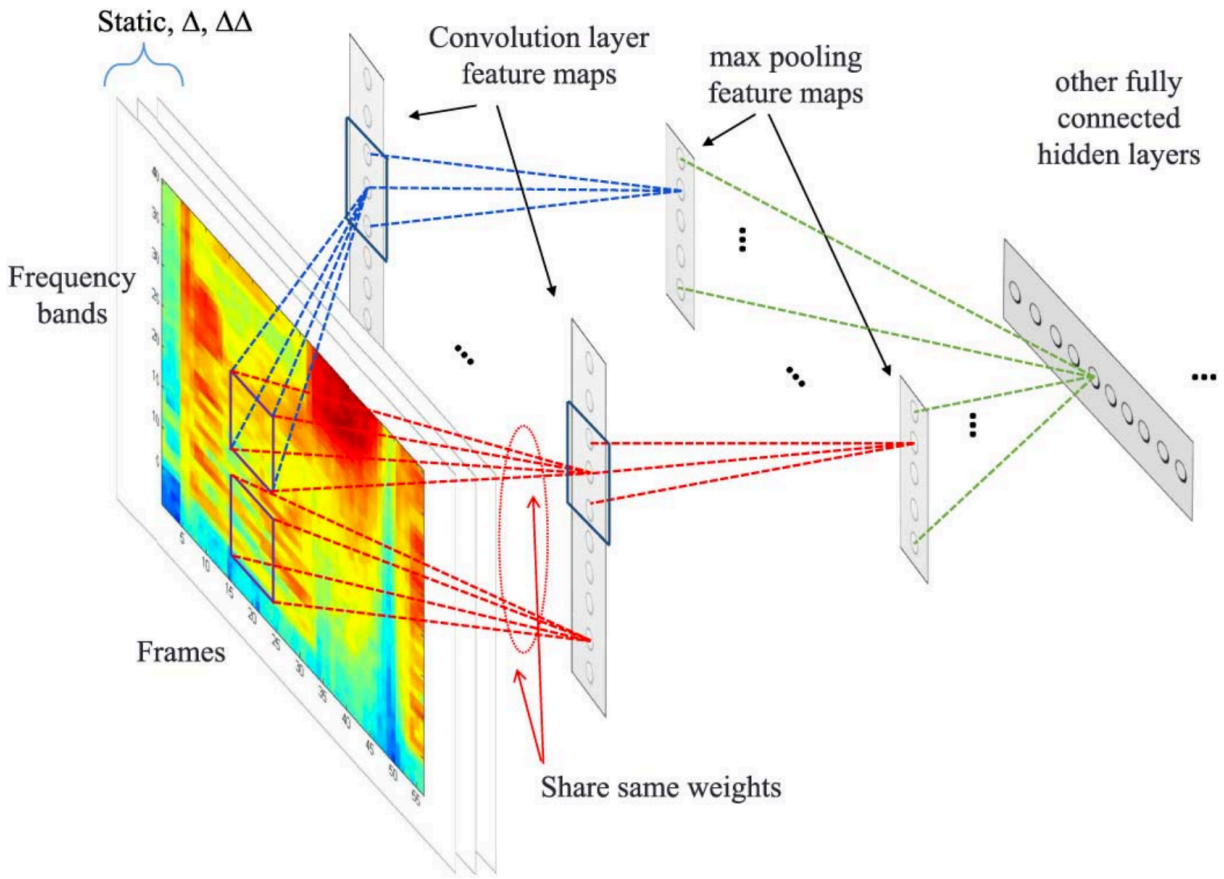
Gy



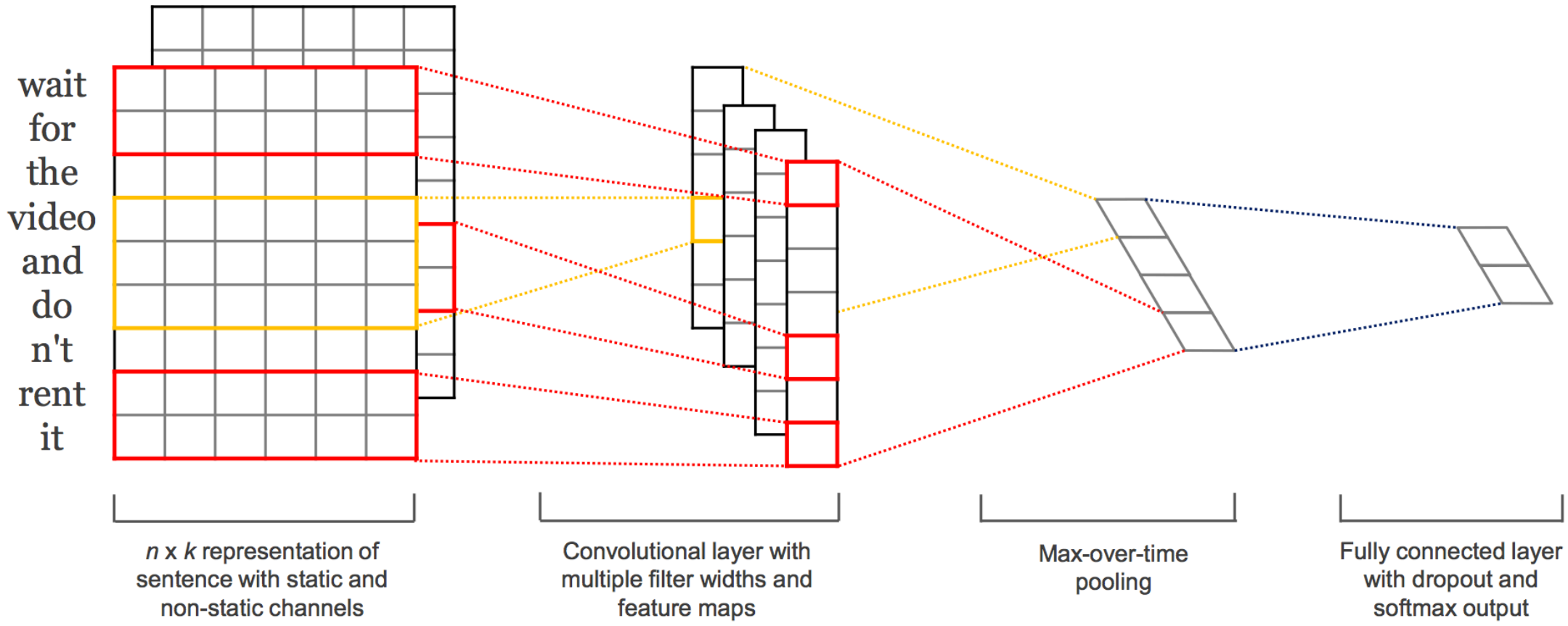
Convolutional Neural Networks



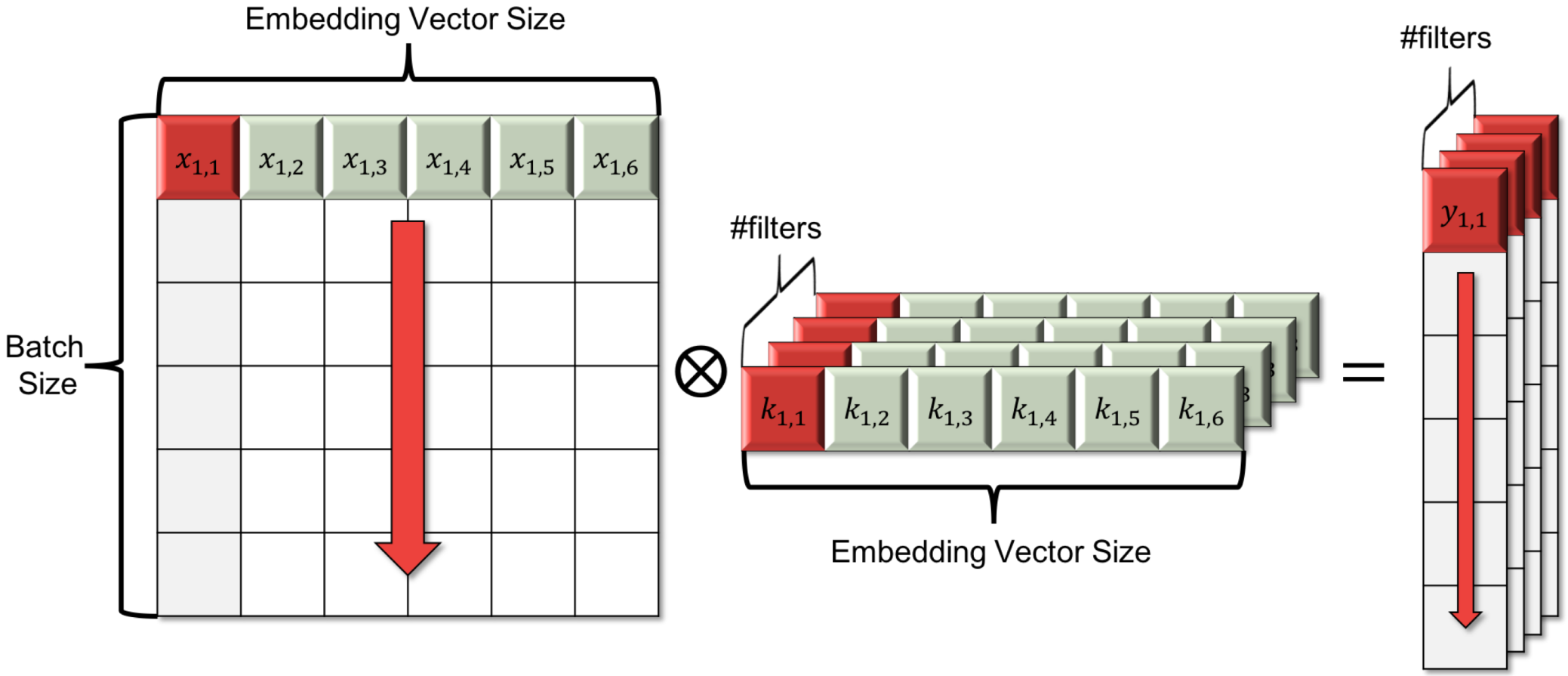
Convolutional Neural Networks



Text Classification with CNN [Kim 2014]



Text Classification with CNN [Kim 2014]



APPENDIX: Confusion Matrix

	oversea	vcoin	stock	havehome	lego	ku	nas	baby	iphonien	coffee	mac	andro	car	bike	gym
oversea	293	3	8	6	14	7	2	17	12	6	9	10	23	10	5
vcoin	9	848	38	2	6	4	0	5	7	3	8	3	8	9	8
stock	8	27	447	14	3	7	2	1	2	5	0	3	10	6	4
havehome	7	4	21	767	9	2	0	7	1	6	1	1	11	4	5
lego	34	37	23	12	899	107	8	90	41	129	41	28	63	149	114
ku	8	6	4	0	18	856	2	19	21	23	8	8	8	17	20
nas	12	7	4	2	2	16	541	2	19	7	32	22	11	10	2
baby	9	5	2	7	16	6	0	923	2	10	2	6	19	11	17
iphonien	25	11	4	1	10	18	9	6	871	7	66	101	22	18	10
coffee	18	21	13	17	64	33	6	65	20	1,038	16	6	34	73	59
mac	22	4	7	8	23	24	25	19	88	27	845	47	18	27	18
andro	17	4	11	2	13	35	3	7	139	10	42	1,244	17	8	6
car	43	4	8	9	13	14	3	22	15	17	8	10	1,189	51	16
bike	10	8	7	4	27	33	5	32	12	25	10	10	63	1,099	75
gym	9	3	5	1	16	14	0	26	7	24	7	3	13	50	937

APPENDIX: Confusion Matrix

		Actual class		
		Cat	**Dog**	**Rabbit**
Predicted class	**Cat**	**5**	2	0
	Dog	3	**3**	2
	Rabbit	0	1	**11**

		Actual class	
		Cat	**Non-cat**
Predicted class	**Cat**	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives

APPENDIX: Precision and Recall

		Actual class	
		Cat	Non-cat
Predicted class	Cat	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives

Recall=5/8

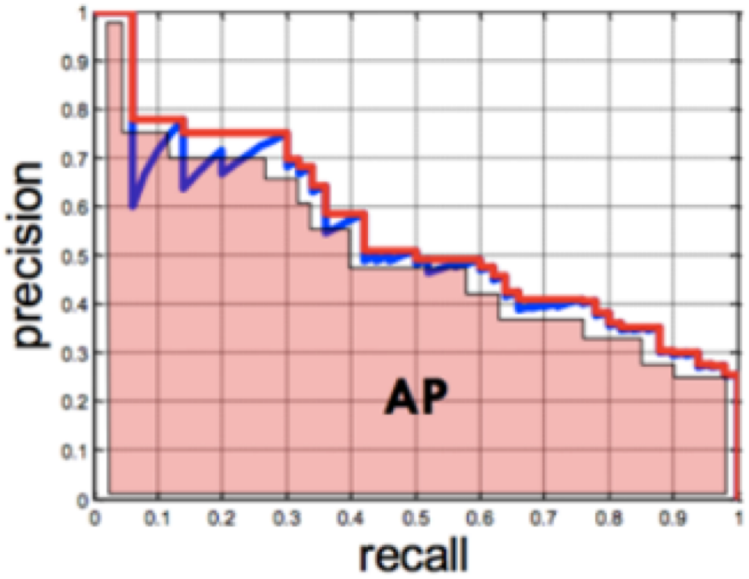
APPENDIX: Precision and Recall

		Actual class	
		Cat	Non-cat
Predicted class	Cat	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives

Precision=5/7

APPENDIX: Precision Recall Curve (PRC)

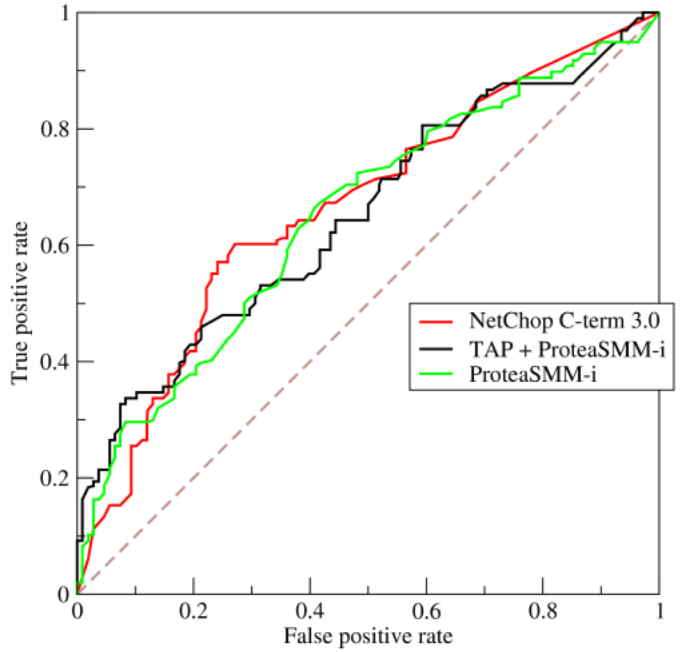
- http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html



APPENDIX: Receiver Operating Curve (ROC)

- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html

		Actual class	
		Cat	Non-cat
Predicted class	Cat	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives



APPENDIX: Area Under Curve (AUC)

- AUC ROC: Area Under Curve Receiver Operating Curve
- AUC PRC: Area Under Curve Precision Recall Curve

- http://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html

Exercise

- Github: https://github.com/kh-kim/simple-ntc
- Corpus: https://1drv.ms/f/s!Aii2Vw-ONsoGoC2QcejcOwfMHox_

Directory Architecture

- data/
- models/
- simple_ntc/
  - cnn.py
  - rnn.py
  - trainer.py
- classfiy.py
- data_loader.py
- get_confusion_matrix.py
- train.py
- utils.py

Training Procedure

- Train
  - Loop {1:n_epochs}
    - Loop{1:n_mini_batchs} # Train Epoch
      - Feed-forward
      - Get Loss and Back-propagate
      - Take a step with gradient-descent
    - Loop{1:n_mini_batchs} # Validate
      - Feed-forward
      - Get loss
    - Compare with lowest loss
    - Save Model
    - Check early stopping requirement

How to Train

- You can check default parameters on train.py.

```
$ python train.py -h
usage: train.py [-h] --model MODEL --train TRAIN --valid VALID
                [--gpu_id GPU_ID] [--verbose VERBOSE]
                [--min_vocab_freq MIN_VOCAB_FREQ]
                [--max_vocab_size MAX_VOCAB_SIZE] [--batch_size BATCH_SIZE]
                [--n_epochs N_EPOCHS] [--early_stop EARLY_STOP]
                [--dropout DROPOUT] [--word_vec_dim WORD_VEC_DIM]
                [--hidden_size HIDDEN_SIZE] [--rnn] [--n_layers N_LAYERS]
                [--cnn] [--window_sizes WINDOW_SIZES] [--n_filters N_FILTERS]
```

Train Example

```
$ python train.py --model ./models/model.pth --train ./data/corpus.train.txt --valid ./data/corpus.valid.txt --rnn --cnn --gpu_id 0
```

How to Inference

```
$ python classify.py -h
usage: classify.py [-h] --model MODEL [--gpu_id GPU_ID]
                   [--batch_size BATCH_SIZE] [--top_k TOP_K]
```

Inference Example

$ head -n 10 ./data/corpus.valid.txt | awk -F'₩t' '{ print $2 }' | python classify.py --model ./models/clien.pth --gpu_id -1 --top_k 3

```
$ head -n 10 ./data/corpus.valid.txt | awk -F'\t' '{ print $2 }' | python classify.py --model ./models/clien
cm_andro cm_iphonien cm_mac     갤 노트 잠금 화면 해제 어 플 사용 하 시 나요 ? : 클리앙
cm_baby cm_car cm_lego  [ 예비 아빠 ] 입당 신고 합니다 : 클리앙
cm_gym cm_oversea cm_vcoin      11 / 07 운동 일지 : 클리앙
cm_ku cm_baby cm_car    커플 이 알콩달콩 하 는 거 보 면 뭐 가 좋 습니까 . utb : 클리앙
cm_iphonien cm_mac cm_car       아이 포니 앙 분 들 께서 는 어떤 사이즈 의 아이 패드 를 더 선호 하 시 나요 ? : 클리앙
cm_coffee cm_lego cm_bike       잉여 잉여 ~ : 클리앙
cm_coffee cm_gym cm_lego        드 뎌 오늘 제대로 된 에스프레소 한잔 마셨 습니다 ! ! ^^ : 클리앙
cm_coffee cm_oversea cm_ku      동네 에 있 는 커피 집 에서 먹 는 커피 빙수 . .. : 클리앙
cm_car cm_oversea cm_bike       땡볕 에 두 시간 세차 하 기 : 클리앙
cm_gym cm_oversea cm_bike       268 . 1 / 22 생 서니 의 말랑말랑 클 핏 일지 ₩ 15 : 클리앙
```

Corpus

- Crawled from Clien

No	Class Name	#Samples	Topic
1	cm_andro	20,000	Android development
2	cm_baby	15,597	Raising baby
3	cm_bike	20,000	Bike hobby
4	cm_car	20,000	Car hobby
5	cm_coffee	19,390	Coffee hobby
6	cm_gym	20,000	Working out
7	cm_havehome	13,062	About having(or rent) home
8	cm_iphonien	20,000	About iPhone
9	cm_ku	20,000	About anime
10	cm_lego	20,000	Lego hobby
11	cm_mac	20,000	About Macintosh
12	cm_nas	11,206	About NAS(Network Attached Storage)
13	cm_oversea	10,381	About living in oversea
14	cm_stock	12,028	About stock trading
15	cm_vcoin	20,000	About crypto-currency trading
	Total	261,664	

Architecture

```
RNNClassifier(
  (emb): Embedding(35532, 128)
  (rnn): LSTM(128, 256, num_layers=4, batch_first=True, dropout=0.3, bidirectional=True)
  (generator): Linear(in_features=512, out_features=15, bias=True)
  (activation): LogSoftmax()
)
```

```
CNNClassifier(
  (emb): Embedding(35532, 128)
  (cnn-3-100): Conv2d(1, 100, kernel_size=(3, 128), stride=(1, 1))
  (cnn-4-100): Conv2d(1, 100, kernel_size=(4, 128), stride=(1, 1))
  (cnn-5-100): Conv2d(1, 100, kernel_size=(5, 128), stride=(1, 1))
  (relu): ReLU()
  (dropout): Dropout(p=0.3)
  (generator): Linear(in_features=300, out_features=15, bias=True)
  (activation): LogSoftmax()
)
```

Evaluation

Architecture	Valid Loss	Valid Accuracy
Bi-LSTM	7.9818e-01	0.7666
CNN	8.4225e-01	0.7497
Bi-LSTM + CNN		0.7679