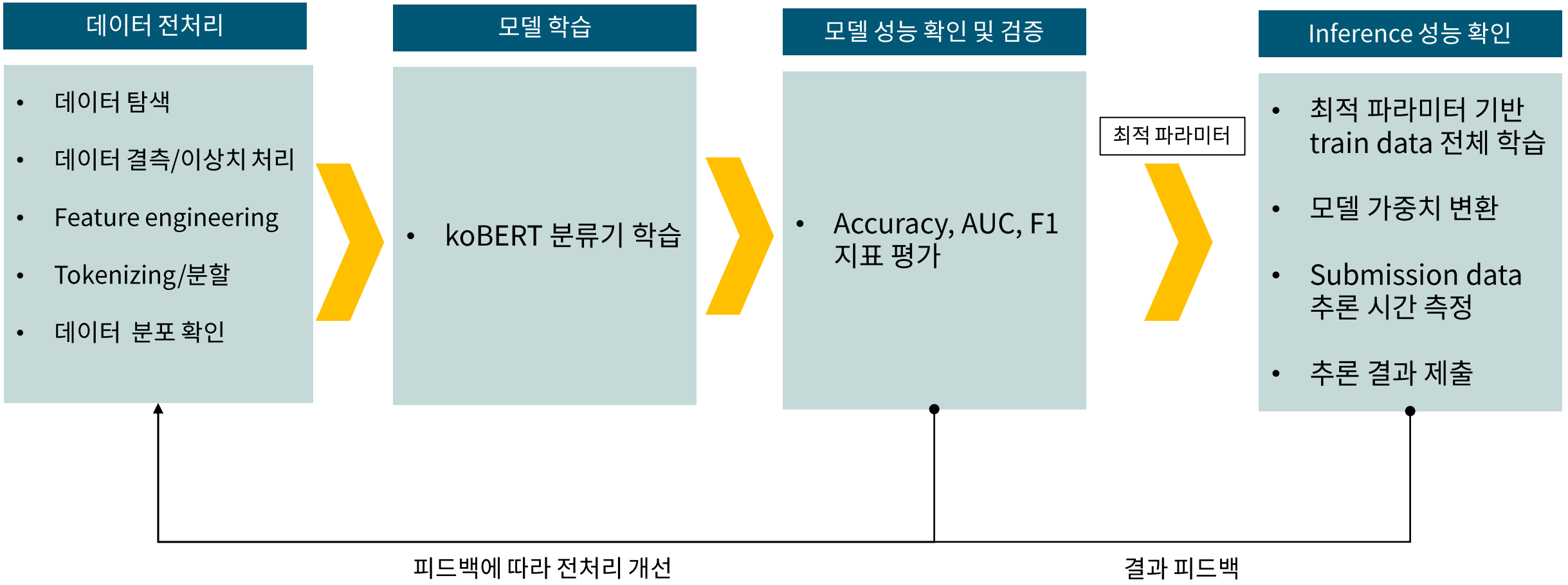
AI야,
진짜 뉴스를 찾아줘!

BERT 모델을 이용한 뉴스 이진 분류

발표자: 이준혁(Team torque)

Contents

Part 1. 모델링 프로세스



Part 2. 데이터 전처리-탐색

◆ **데이터 변수**

- n_id : string, NEWS00000~NEWS10002
- Date : 2020/01/01~2020/06/30
- Title : string
- Content : string
- Ord : int, 0~396
- Info : 0 & 1

◆ **대회 주제를 단일 문장 분류 과제로 가정**

- Ex) 네이버 영화 댓글 감정분석(Naver Movie Sentiment Classification)

Part 2. 데이터 전처리-결측/이상치 처리

◆ 데이터의 content, title 변수의 결측치 체크

- 결측치는 없는 것을 확인
- 뉴스는 검증된 사람이 쓰는 글이므로 문법/오타 등의 처리는 불필요하다고 판단

◆ Content, title이 동일하지만 info가 틀린 이상치 체크 및 제거

- NEWS08586에서 4개의 이상치 제거

◆ 데이터에 포함된 HTML 특수문자 변환(ex: 00주 신고가#&33 → '00주 신고가!')

- Tokenizing 시 문제가 발생할 수 있기 때문
- 변환 전*: 00주 신고가#&33 → _00, 주, _신고,가, #, &, 33(분해된 html 특수문자는 의미 소실)
- 변환 후*: 00주 신고가!→ _00, 주, _신고,가, !

*예시 문장으로 실제 tokenizing 결과와는 다를 수 있습니다.

Part 2. 데이터 전처리-feature engineering

◆ 변수 간 상관관계 조사

- 중복되는 Content 내용 중 일부 항목은 Title과 관계 없이 info가 달라지지 않는 것을 확인
- 예시*

a)
Title: 0000그룹, IT솔루션 총판사업분할 '0000' 출범
Content: "실적기반" 저가에 매집해야 할 0월 급등유망주 TOP 5 전격공개
Info: 1

b)
Title: 발표 임박!! 즉시 매수.
Content: "실적기반" 저가에 매집해야 할 0월 급등유망주 TOP 5 전격공개
Info: 1

◆ 데이터의 중복을 줄이기 위해 Title과 Content 둘 다 사용

*예시 문장으로 실제 기사 데이터에 해당 내용은 존재하지 않습니다.

Part 2. 데이터 전처리-feature engineering

◆ 변수 간 상관관계 조사

- Ord와 info의 유의미한 관계 확인

독립-종속변수 간 상관계수(pearson)

	date	ord	info
date	1	0.013	0.028
ord		1	0.124
info			1

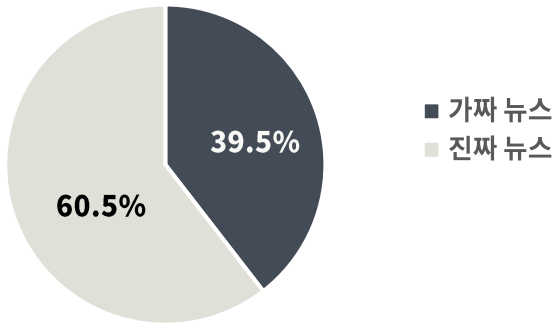
→ **Date 변수는 종속변수 info와 2%의 관계가 있음**

Ord 변수는 info와 12%의 관계가 있음

(경험상 기사 뒤쪽에 광고가 많음)

- 데이터의 분포 확인

Full train data 레이블 분포



→ **불균형이 크지 않은 데이터, no sampling**
(sampling시 오버피팅 유발)

Part 2. 데이터 전처리-feature engineering

◆ Multimodal fusion 기법은 모델 비대화 및 추론 시간 악화

- 그림 1에서 볼 수 있듯, 모델에 레이어가 추가됨에 따라 비대해짐
- 정확도는 증가할 수 있지만, 속도 및 용량, 리소스 사용 면에서는 나쁜 결과를 불러옴
- Content와 Title, Ord를 전처리 과정에서 병합해 사용

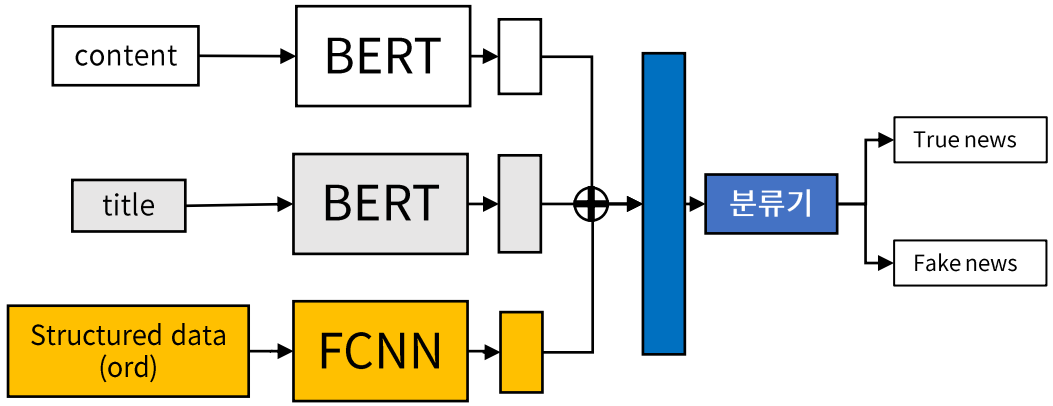


그림 1) BERT-FCNN multimodal fusion model

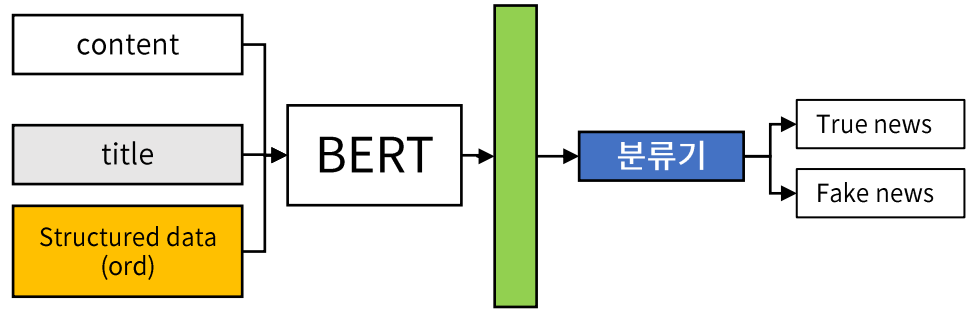


그림 2) single BERT model

Part 2. 데이터 전처리-tokenizing/분할

◆ **Feature engineering 결과에 따라 문장 전처리**

◆ **전처리 전***

- TITLE:’0000년 한국 TV 0대중 0대 인터넷 연결된다 #&33’,
- CONTENT: ‘OOO, 00주 신고가‘
- ORD: 01

◆ **전처리 후***

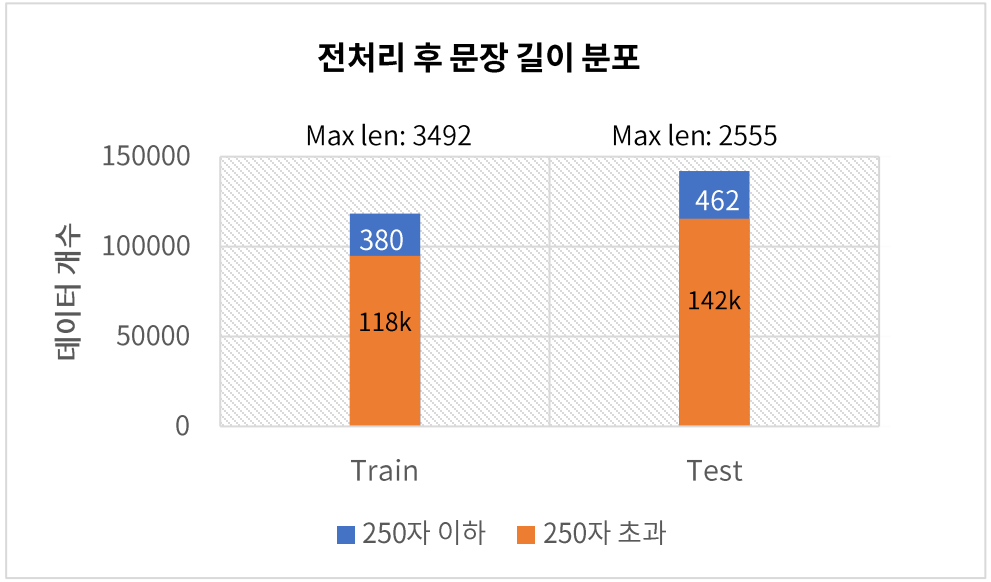
- ‘0000년 한국 TV 0대중 0대 인터넷 연결된다! OOO, 00주 신고가. 01’

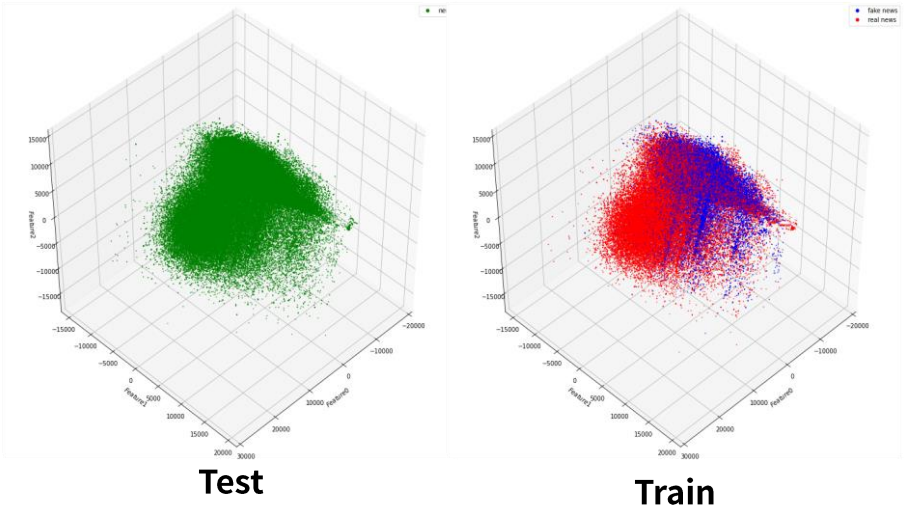
◆ **중복되는 값 제거 및 tokenizing(pretrained sentencepiece tokenizer)**

◆ **데이터 분할: Full train data(100%) → train 80%, valid 10%, test 10%**

*예시 문장으로 실제 기사 데이터에 해당 내용은 존재하지 않습니다.

Part 2. 데이터 전처리-분포 확인





토크나이징 시 64개 변수 내에 효과적으로 임베딩 가능

경향성 확인, 적절한 분류 가능

Part 3. 모델 소개-BERT 소개

◆ What is BERT?

- Bidirectional Encoder Representations from Transformers → 양방향적 Transformer 기반의 인코더
- Transformer 아키텍쳐를 이용, 양방향성(bidirectional)을 가지고 문장을 파악할 수 있는 모델
- 기존 단방향성 모델의 문맥 파악 및 성능 향상 한계 해결
- Pre-trained 한국어 모델 있음(Kobert*)
- Fine-tuning 용이

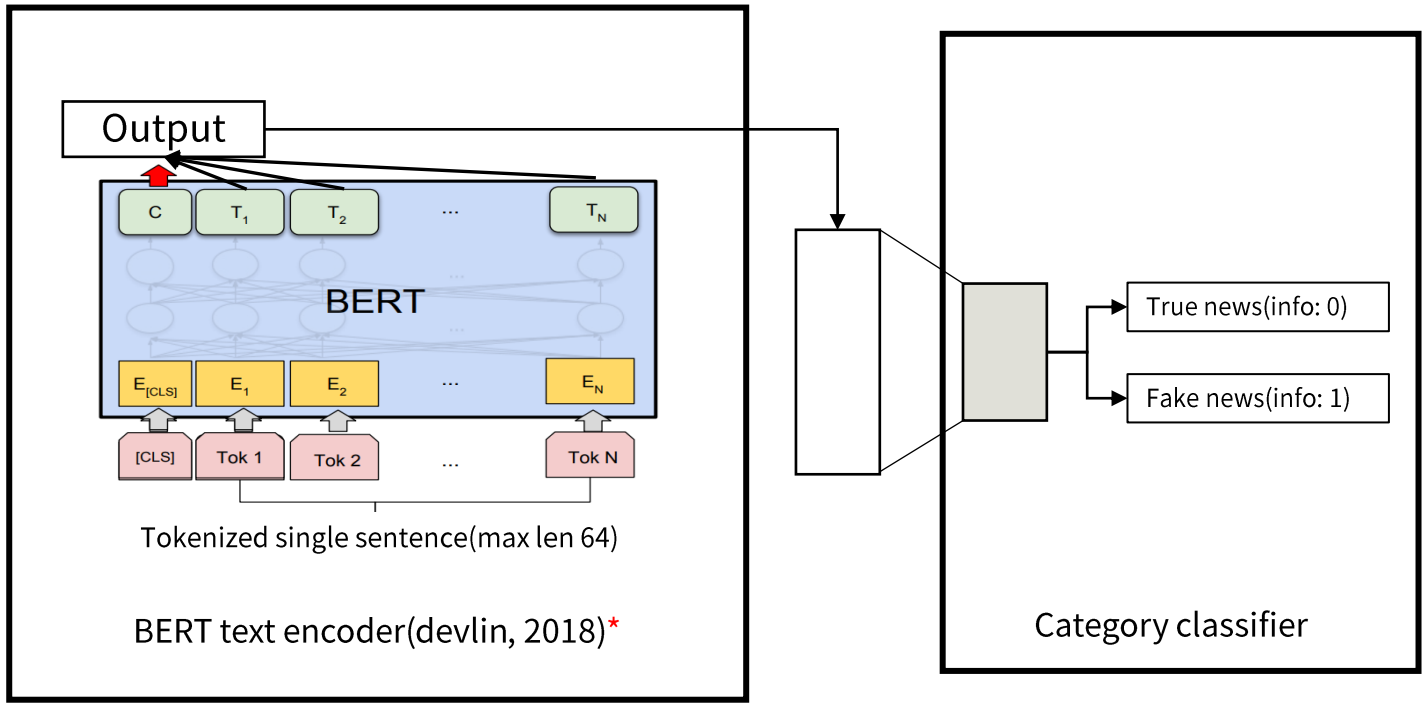
◆ Where to use?

- 문장 질의 task
- 기계 번역
- 문장 분류

Part 3. 모델 소개-BERT 분류기 구조

Model architecture

- Pre-trained BERT(KoBERT) + FCNN classifier



*picture source: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, devlin et al

Part 3. 모델 소개-koBERT 분류기 학습

◆ Model 상세

- BERT-Base 기반 pretrained model(Kobert)
- 학습 가능한 parameter 개수: 약 92,200,000개

◆ Pretrain data

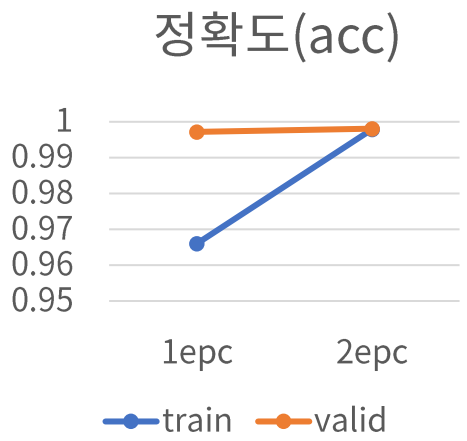
- 한국어 위키(약 5,000,000개 문장)
- 한국어 뉴스(약 20,000,000개 문장)

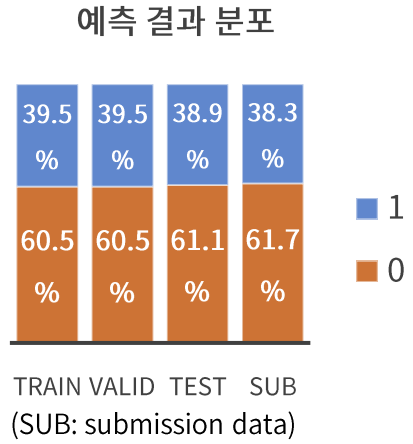
◆ Finetuning(학습 및 검증) data

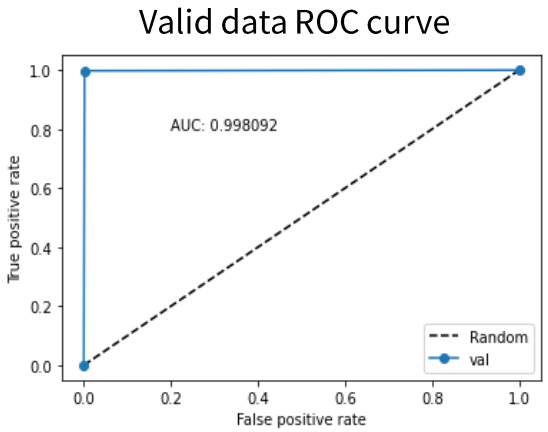
- Train data(94996개 문장)
- Valid(11874개 문장), test data(11875개 문장)

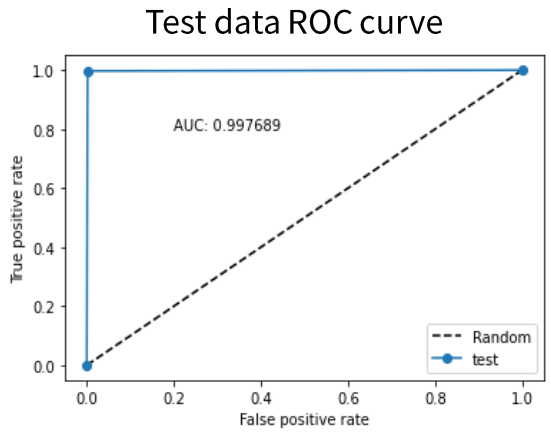
→ 모델 미세 조정 학습을 통해 얻은 최적 파라미터: 2epoch, 5e-5 learning rate, 128 batchsize

Part 4. 성능 검증-지표 평가









Score	Validation data	Test data
ACCURACY	0.9981	0.9978
AUC	0.9980	0.9976
F1 score	0.9976	0.9971

Validation data	Predicted: 0	Predicted: 1	Error
Actual: 0	7146	12	0.17%
Actual: 1	10	4665	0.21%

Test data	Predicted: 0	Predicted: 1	Error
Actual: 0	7215	13	0.18%
Actual: 1	13	4593	0.28%

Part 4. 성능 검증- inference 성능 확인

◆ 모델 추론 속도 가속을 위한 가중치 변환

- 정밀도: 실수를 표현할 때 사용하는 형식
- 정밀도가 커질수록 정확하게 표현 가능한 자릿수가 많지만, 사용되는 메모리의 크기가 커진다
- 그래픽 처리 장치(GPU)의 정밀도(precision)별 초당 처리 용량*

Performance	NVIDIA T4	NVIDIA V100	NVIDIA A100
Double precision	No data	7 Tera FLOPS	9.7 Tera FLOPS
Single precision	8.1 Tera FLOPS	14 Tera FLOPS	19.5 Tera FLOPS
Half precision	65 Tera FLOPS	No data	624 Tera FLOPS
INT8	130 Tera OPS	No data	1248 Tera OPS

Tera: 10^12, 1조
OPS: 초당 정수 연산 횟수
FLOPS: 초당 실수 연산 횟수

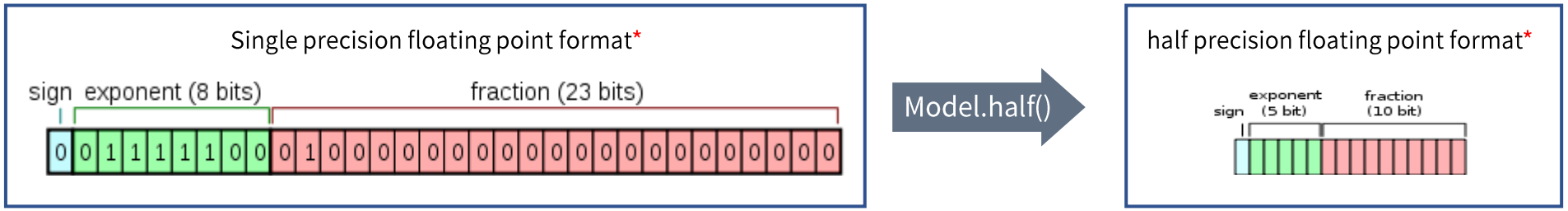
- 정밀도가 내려갈수록 초당 처리 용량이 커짐
- Half precision(반정밀도)을 사용해 속도 향상이 가능하다고 판단

* 출처: xiilab.com, nvidia 제품 브로슈어, PCIe 제품 기준

Part 4. 성능 검증- inference 성능 확인

◆ 모델 추론 속도 가속을 위한 가중치 변환

- 학습된 모델 로드 후 가중치를 single precision(FP32) → half precision(FP16)로 변경



- 가중치 변환은 정확도에 큰 영향을 주지 않음.

Model	Baseline	Mixed Precision	Reference
AlexNet	56.77%	56.93%	(Krizhevsky et al., 2012)
VGG-D	65.40%	65.43%	(Simonyan and Zisserman, 2014)
GoogLeNet (Inception v1)	68.33%	68.43%	(Szegedy et al., 2015)
Inception v2	70.03%	70.02%	(Ioffe and Szegedy, 2015)
Inception v3	73.85%	74.13%	(Szegedy et al., 2016)
Resnet50	75.92%	76.04%	(He et al., 2016b)

원 가중치(FP32)/혼합 가중치(FP16+FP32)에서 여러 모델의 top-1 정확도 비교(Narang, 2018)**

- 변환 전/후 prediction 결과 비교: 정확도 차이 없음, 속도 빨라짐
- 추론에 half precision 사용

* picture source: en.Wikipedia.org
** Table source: MIXED PRECISION TRAINING, Narang et al

Part 4. 성능 검증- inference 성능 확인

◆ **최적 파라미터를 사용한 full train data 학습(FP32, 118745개 문장)**

◆ **Submission data 추론 시간* 측정(batch size: 256, NVIDIA® T4 GPU)**

- FP32: 0.00424sec/sample → FP16: 0.00164sec/sample
- Half precision에서 38%의 속도 개선 보임

◆ **최적 환경에서 추론 시간* 측정(batch size: 256, NVIDIA® V100 GPU)**

- FP16: 0.00088sec/sample

◆ **리더보드 정확도: private 기준 99.382%**

* 자체 기준으로 측정된 시간, 주최측 측정 시간과 차이가 있을 수 있음

Part 5. 적용 효과 및 개선사항

◆ 적용 효과

- 정보 전달 시 test data 기준 가짜 뉴스 99.72% 제거 가능
- 투자 분석가, 고객 등 의사 결정에 긍정적 효과

◆ 개선사항

- 비식별화된 뉴스 회사 이름(1~n) 변수 추가
- Jit.script 을 이용한 c++ 활용 및 추론 시간 개선 가능
- INT8 quantization(가중치 변경)을 통한 추가 가속

THANK YOU

leejunhyuk2526@gmail.com