

NLP CHALLENGE : TRIAL AND ERROR

Team : State_Of_The_Art
Dongju Park



School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology (GIST)
Meta-Evolutionary Machine Intelligence Laboratory (MEMI lab)



NLP Challenge

네이버, 창원대가 함께하는
NLP Challenge
18.11.16. Fri. - 12.14. Fri.

1st naver
nsmi
nlp
challenge
changwon
univ.

행사 일정
4주간 NLP에 관심 있는 분께 온라인 진행
- 11월 15일: NLP에 대한 이해를 돕기 위한 NLP 기초 강의
- 11월 16일: NLP에 대한 이해를 돕기 위한 NLP 기초 강의
- 11월 17일: NLP에 대한 이해를 돕기 위한 NLP 기초 강의

참가 신청
<http://naver.me/8e9a5gC>
11월 15일 14시까지 네이버 앱에 신청 가능

참가 대상
NLP에 관심이 있는 분
NLP에 대한 이해를 돕기 위한 NLP 기초 강의

시상 및 혜택
총 1000만원 상당의 상품 및 기념품
- 1등: 1000만원 상당의 상품 및 기념품
- 2등: 500만원 상당의 상품 및 기념품
- 3등: 200만원 상당의 상품 및 기념품
- 4등: 100만원 상당의 상품 및 기념품
- 5등: 50만원 상당의 상품 및 기념품
- 6등: 20만원 상당의 상품 및 기념품
- 7등: 10만원 상당의 상품 및 기념품
- 8등: 5만원 상당의 상품 및 기념품
- 9등: 2만원 상당의 상품 및 기념품
- 10등: 1만원 상당의 상품 및 기념품

상세 내용
<https://github.com/naver/nlp-challenge>
NLP에 대한 이해를 돕기 위한 NLP 기초 강의
NLP에 대한 이해를 돕기 위한 NLP 기초 강의
NLP에 대한 이해를 돕기 위한 NLP 기초 강의

NAVER Clova NSML

NLP Challenge

네이버, 창원대가 함께하는
NLP Challenge
18.11.16. Fri. - 12.14. Fri.

1st naver
nsmi
nlp
challenge
changwon
univ.

행사 일정
4주간 NLP에 관련된 다양한
· 11월 16일 (금요일) 12월 14일 (금요일)
· 네이버 클라우드, 창원대, NSML
· NSML, 네이버, 창원대, NSML

참가 신청
http://naver.me/nlpchallenge
11월 16일 (금요일) 12월 14일 (금요일)
· 11월 16일 (금요일) 12월 14일 (금요일)
· 11월 16일 (금요일) 12월 14일 (금요일)

참가 대상
· NLP에 관심이 있는 대학생, 대학원생
· NLP에 관심이 있는 대학생, 대학원생

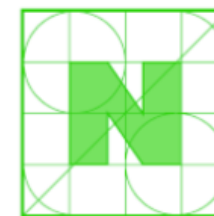
상장 및 혜택
· 총 1000만원 상당의 상품 및 기념품
· 1000만원 상당의 상품 및 기념품
· 1000만원 상당의 상품 및 기념품

상장 기준
· NLP에 관심이 있는 대학생, 대학원생
· NLP에 관심이 있는 대학생, 대학원생
· NLP에 관심이 있는 대학생, 대학원생

NAVER Clova NSML

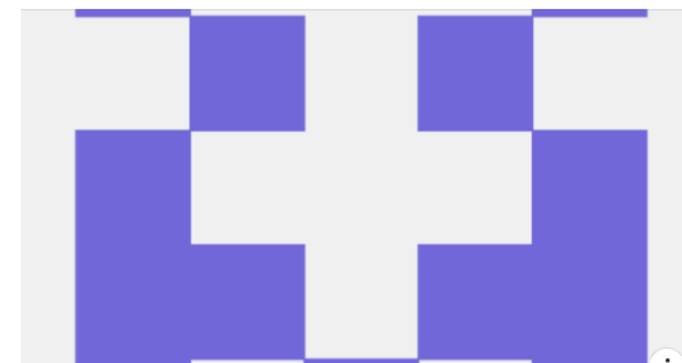


님이 그룹에 링크를 공유했습니다: TensorFlow KR.
11월 19일 · 🌐
<https://github.com/naver/nlp-challenge>



GITHUB.COM
naver/nlp-challenge
NLP Shared tasks (NER, SRL) using NSML.
Contribute to naver/nlp-challenge development by creating an account on GitHub.

님이 그룹에 링크를 공유했습니다: TensorFlow KR.
11월 20일 · 🌐
이미 광고된 것 같은데 추가로 첨언해서 복불합니다.
네이버 서치 앤 클로바 조직 NLP/대화 Inho Kang 리더님 뎬
"우수한 품질 내는 팀, NLP/대화 팀 입사 지원서 공식 코딩테스트 면제 등 특
혜드립니다."
"팀 TO 아직 여유 많으니 관심있으신 분들은 많은 지원 바랍니다."



GITHUB.COM
naver/nlp-challenge
NLP Shared tasks (NER, SRL) using NSML. Contribute to naver/nlp-

NLP Challenge



NLP Challenge

DEVIEW 2018

Excellence · Sharing · Growth

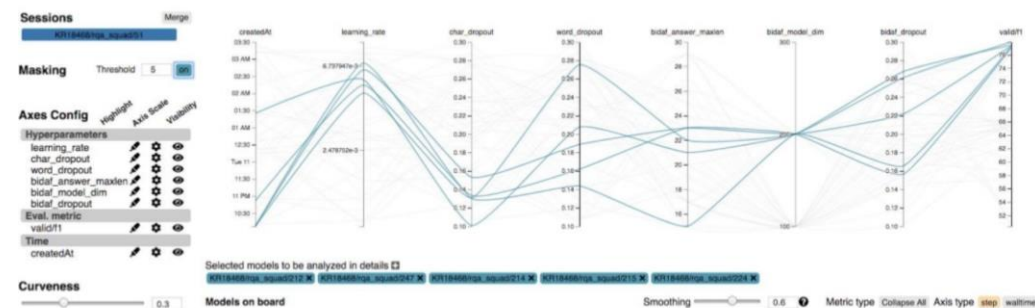
NSML:
머신러닝 플랫폼 서비스하기
& 모델 튜닝 자동화하기

김민규, 김진웅
NSML
Clova

DEVIEW
2018

Select best hyperparameters

DEVIEW
2018



Model

CoNLL 2003 (English)

The [CoNLL 2003 NER task](#) consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC). Models are evaluated based on span-based F1 on the test set.

Model	F1	Paper / Source	Code
Flair embeddings (Akbik et al., 2018)	93.09	Contextual String Embeddings for Sequence Labeling	Flair framework
BERT Large (Devlin et al., 2018)	92.8	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
CVT + Multi-Task (Clark et al., 2018)	92.61	Semi-Supervised Sequence Modeling with Cross-View Training	Official
BERT Base (Devlin et al., 2018)	92.4	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
BiLSTM-CRF+ELMo (Peters et al., 2018)	92.22	Deep contextualized word representations	AllenNLP Project AllenNLP GitHub
Peters et al. (2017)	91.93	Semi-supervised sequence tagging with bidirectional language models	
HSCRF (Ye and Ling, 2018)	91.38	Hybrid semi-Markov CRF for Neural Sequence Labeling	HSCRF
NCRF++ (Yang and Zhang, 2018)	91.35	NCRF++: An Open-source Neural Sequence Labeling Toolkit	NCRF++
LM-LSTM-CRF (Liu et al., 2018)	91.24	Empowering Character-aware Sequence Labeling with Task-Aware Neural Language Model	LM-LSTM-CRF

Model

CoNLL 2003 (English)

The [CoNLL 2003 NER task](#) consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC). Models are evaluated based on span-based F1 on the test set.

Model	F1	Paper / Source	Code
Flair embeddings (Akbik et al., 2018)	93.09	Contextual String Embeddings for Sequence Labeling	Flair framework
BERT Large (Devlin et al., 2018)	92.8	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
CVT + Multi-Task (Clark et al., 2018)	92.61	Semi-Supervised Sequence Modeling with Cross-View Training	Official
BERT Base (Devlin et al., 2018)	92.4	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
BiLSTM-CRF+ELMo (Peters et al., 2018)	92.22	Deep contextualized word representations	AllenNLP Project AllenNLP GitHub
Peters et al. (2017)	91.93	Semi-supervised sequence tagging with bidirectional language models	
HSCRF (Ye and Ling, 2018)	91.38	Hybrid semi-Markov CRF for Neural Sequence Labeling	HSCRF
NCRF++ (Yang and Zhang, 2018)	91.35	NCRF++: An Open-source Neural Sequence Labeling Toolkit	NCRF++
LM-LSTM-CRF (Liu et al., 2018)	91.24	Empowering Character-aware Sequence Labeling with Task-Aware Neural Language Model	LM-LSTM-CRF

알고있고

간단하고

쉽게 만들 수 있는

BILSTM-CRF + (ELMo)

Model

CoNLL 2003 (English)

The [CoNLL 2003 NER task](#) consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC). Models are evaluated based on span-based F1 on the test set.

Model	F1	Paper / Source	Code
Flair embeddings (Akbik et al., 2018)	93.09	Contextual String Embeddings for Sequence Labeling	Flair framework
BERT Large (Devlin et al., 2018)	92.8	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
CVT + Multi-Task (Clark et al., 2018)	92.61	Semi-Supervised Sequence Modeling with Cross-View Training	Official
BERT Base (Devlin et al., 2018)	92.4	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
BiLSTM-CRF+ELMo (Peters et al., 2018)	92.22	Deep contextualized word representations	AllenNLP Project AllenNLP GitHub
Peters et al. (2017)	91.93	Semi-supervised sequence tagging with bidirectional language models	
HSCRF (Ye and Ling, 2018)	91.38	Hybrid semi-Markov CRF for Neural Sequence Labeling	HSCRF
NCRF++ (Yang and Zhang, 2018)	91.35	NCRF++: An Open-source Neural Sequence Labeling Toolkit	NCRF++
LM-LSTM-CRF (Liu et al., 2018)	91.24	Empowering Character-aware Sequence Labeling with Task-Aware Neural Language Model	LM-LSTM-CRF

NER Model Baseline for NSML

NER baseline Model 구조

- Bidirectional RNN + CRF
- 어절, 음절(RNN) Concat 하여 사용

```
import tensorflow as tf

class Model:
    def __init__(self, parameter):
        self.parameter = parameter

    def build_model(self):
        self._build_placeholder()

        # { "morph": 0, "morph_tag": 1, "tag" : 2, "character": 3, .. }
        self._embedding_matrix = []
        for item in self.parameter["embedding"]:
            self._embedding_matrix.append(self._build_embedding(item[1], item[2], name="embedding_" + item[0]))

        # 각각의 임베딩 값을 가져온다
        self._embeddings = []
        self._embeddings.append(tf.nn.embedding_lookup(self._embedding_matrix[0], self.morph))
        self._embeddings.append(tf.nn.embedding_lookup(self._embedding_matrix[1], self.character))

        # 음절을 이용한 임베딩 값을 구한다.
        character_embedding = tf.reshape(self._embeddings[1], [-1, self.parameter["word_length"], self.parameter["embedding"][1][2]])
        char_len = tf.reshape(self.character_len, [-1])
```


Model

NER Model Baseline for NSML

NER baseline Model 구조

- Bidirectional RNN + CRF
- 어절, 음절(RNN) Concat 하여 사용

baseline

2018-11-19 21:05:54

모델번호:1
팀명: nsmllteam

88.0977

Hyper-parameters

Epochs	40
Batch size	20
Learning rate	0.01
Word embedding size	32
Char embedding size	32
Tag embedding size	32
Lstm units	32
Char lstm units	64

Model

NER Model Baseline for NSML

NER baseline Model 구조

- Bidirectional RNN + CRF
- 어절, 음절(RNN) Concat 하여 사용

baseline

2018-11-19 21:05:54

모델번호:1
팀명: nsmllteam

88.0977

88.0977

Hyper-parameters

Epochs	40
Batch size	20
Learning rate	0.01
Word embedding size	32
Char embedding size	32
Tag embedding size	32
Lstm units	32
Char lstm units	64

49.4154

Model

baseline	2018-11-19 21:05:54	모델번호: 1 팀명: nsmteam	88.0977
----------	---------------------	------------------------	---------

Hyper-parameters

Epochs	40
Batch size	20
Learning rate	0.01
Word embedding size	32
Char embedding size	32
Tag embedding size	32
Lstm units	32
Char lstm units	64

49.4154



Hyper-parameters

Epochs	20
Batch size	10
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

66.3427

Model

baseline	2018-11-19 21:05:54	모델번호: 1 팀명: nsmteam	88.0977
----------	---------------------	------------------------	---------

Hyper-parameters

Epochs	20
Batch size	10
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

66.3427



Hyper-parameters

Epochs	100
Batch size	1000
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

71.5061

Model

Tuning Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit

Model

Tuning Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit

```
nsml.report(summary=True, scope=locals(), train_loss=avg_cost, step=epoch)  
nsml.save(epoch)
```

모델번호 = epoch

baseline

2018-11-19 21:05:54

모델번호:1
팀명: nsmlteam

88.0977

Model

main.py

```
parser.add_argument('--train_lines', type=int, default=50, required=False, help='Maximum train lines')
```

dataset_batch.py

```
def get_data_batch_size(self, n, train=True):  
    if train:  
        for i, step in enumerate(range(0, self.parameter["train_lines"], n)):  
            if len(self.morphs[step:step + n]) == n:  
                yield self.morphs[step:step+n], self.ne_dicts[step:step+n], self.characters[step:step+n], \  
                    self.sequence_lengths[step:step+n], self.character_lengths[step:step+n], \  
                    self.labels[step:step+n], i  
    else:  
        for i, step in enumerate(range(0, self.parameter["train_lines"], n)):  
            if len(self.morphs[step:step+n]) == n:  
                yield self.morphs[step:step+n], self.ne_dicts[step:step+n], self.characters[step:step+n], \  
                    self.sequence_lengths[step:step+n], self.character_lengths[step:step+n], \  
                    self.labels[step:step+n], i
```

Model

baseline	2018-11-19 21:05:54	모델번호: 1 팀명: nsmteam	88.0977
----------	---------------------	------------------------	---------

Hyper-parameters

Epochs	100
Batch size	1000
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

75 epochs : 71.5061



Hyper-parameters

Train lines	90000
Epochs	20
Batch size	64
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

2 epochs : 85.5220

Model

baseline	2018-11-19 21:05:54	모델번호:1 팀명: nsmteam	88.0977
----------	---------------------	-----------------------	---------

keep_prob word_length batch_size

embedding_size lstm_units

Tuning Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Submit Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit
Tuning Submit Submit Submit

Model

baseline	2018-11-19 21:05:54	모델번호: 1 팀명: nsm1team	88.0977
----------	---------------------	-------------------------	---------

Hyper-parameters

Train lines	90000
Epochs	20
Batch size	64
Learning rate	0.02
Word embedding size	16
Char embedding size	16
Tag embedding size	16
Lstm units	16
Char lstm units	32

85.5220



Hyper-parameters

Train lines	90000
Epochs	20
Batch size	128
Learning rate	0.005
Word embedding size	128
Char embedding size	128
Tag embedding size	128
Lstm units	128
Char lstm units	128

88.1035

Model

Hyper-parameters

Train lines	90000
Epochs	20
Batch size	128
Learning rate	0.005
Word embedding size	128
Char embedding size	128
Tag embedding size	128
Lstm units	128
Char lstm units	128

88.1035

Hyper-parameters tuning

training data score



test data score

Model

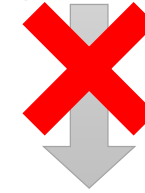
Hyper-parameters

Train lines	90000
Epochs	20
Batch size	128
Learning rate	0.005
Word embedding size	128
Char embedding size	128
Tag embedding size	128
Lstm units	128
Char lstm units	128

88.1035

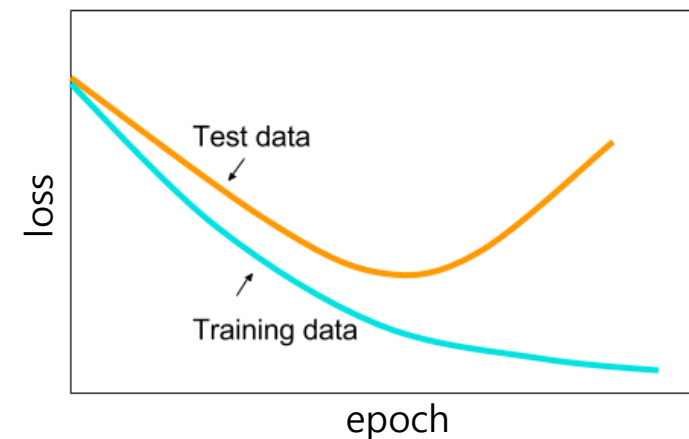
Hyper-parameters tuning

training data score



test data score

Overfitting



Model

MODEL 1

Training data : 90000

training data score



test data score

MODEL 2

Training data : 80000

Validation data : 10000

validation data score



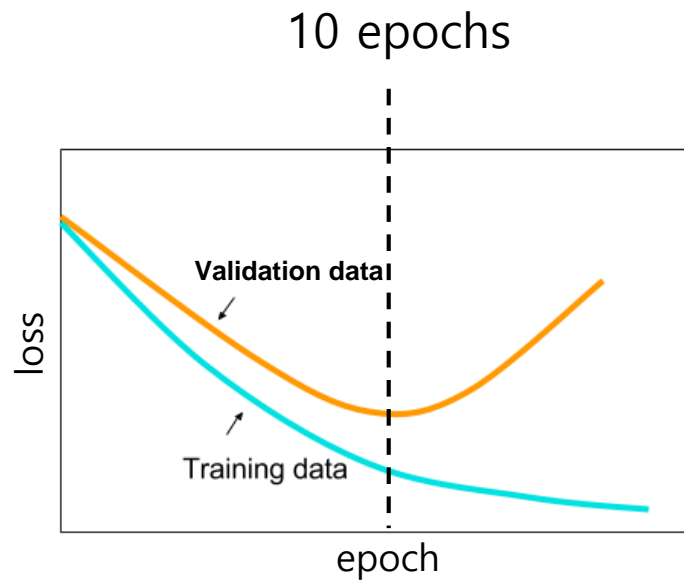
test data score

Model

MODEL 2

Training data : 80000

Validation data : 10000

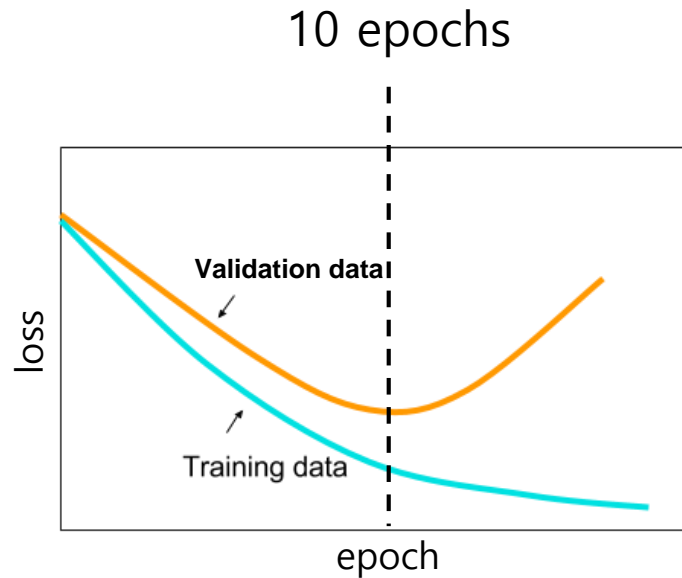


Model

MODEL 2

Training data : 80000

Validation data : 10000



MODEL 1

Training data : 90000

Submit

10 epochs \pm 1 epoch

Model

A word cloud consisting of the words "Tuning" and "Submit" in various sizes and orientations. The words are scattered across the frame, with "Submit" appearing slightly more frequently than "Tuning". The colors are black and white.

88.1035  88.2567

Model

Changing optimizer

AdamOptimizer



GradientDescentOptimizer

MomentumOptimizer

RMSPropOptimizer

Model

Changing optimizer

AdamOptimizer



GradientDescentOptimizer

MomentumOptimizer

RMSPropOptimizer

A word cloud consisting of the words "Submit" and "Tuning" in various sizes and orientations. The words are black and set against a white background. "Submit" appears more frequently and in larger sizes than "Tuning".

88.7138

⋮

89.2098

•
•
•

89.8737

•
•
•

90.1050

⋮

90.2499

Model 1
score

Model

MODEL 2

Training data : 80000

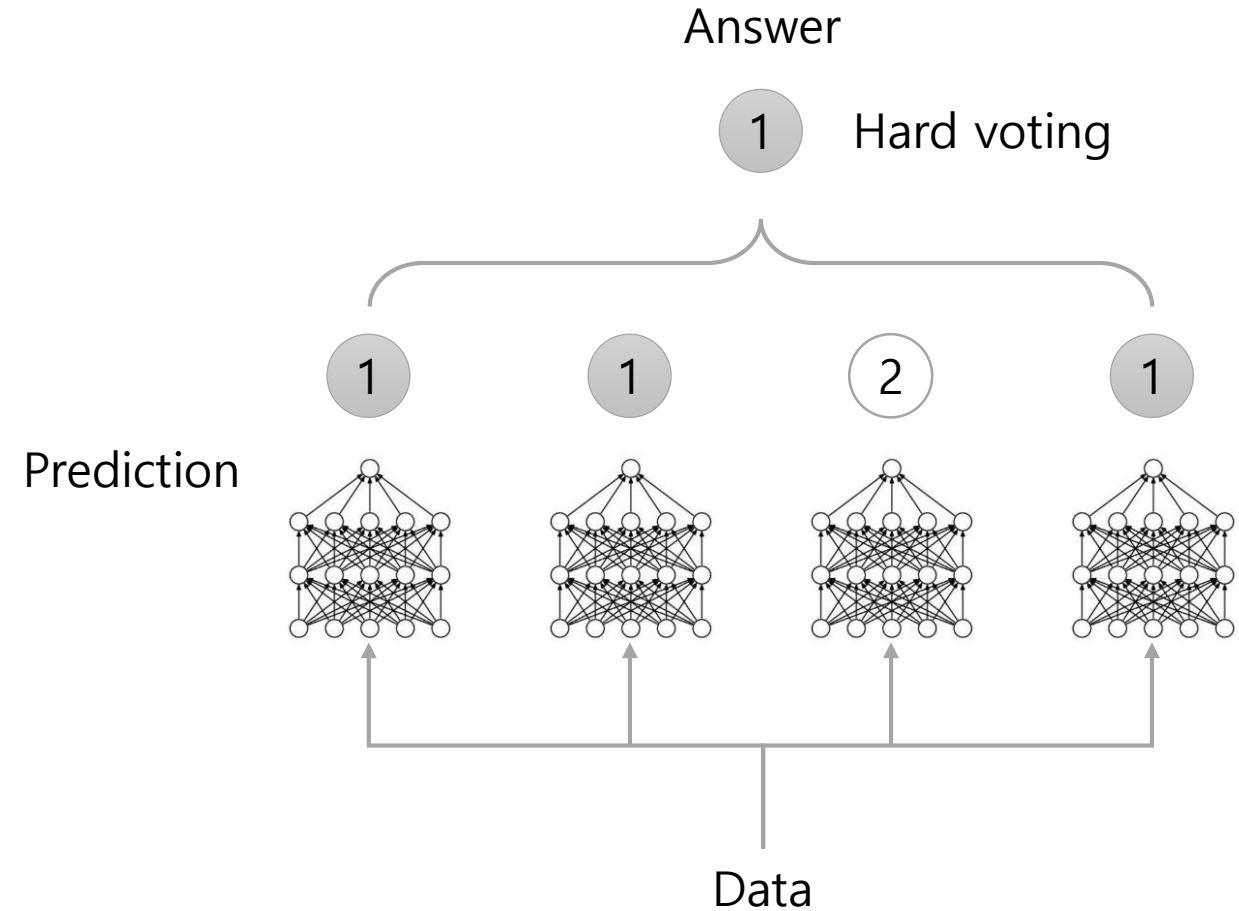
Validation data : 10000

Ensemble
MODEL

Training data : 80000

Validation data : 10000

Ensemble of **N** different models



Model

Ensemble
MODEL

Ensemble of **three** different models

90.2499



90.4219

Model

Ensemble
MODEL

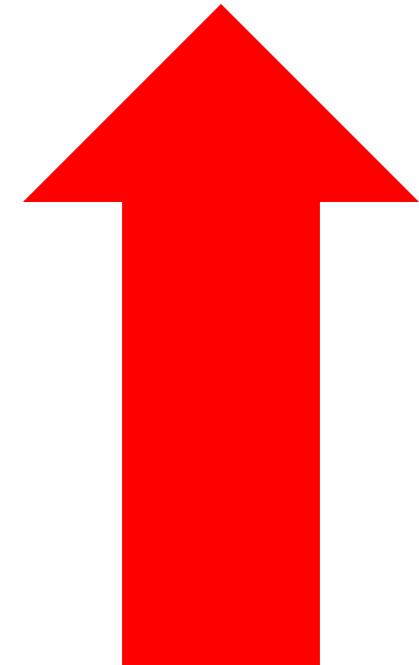
Ensemble of **five** different models

Ensemble
MODEL

Training data : 90000

~~Validation data : 10000~~

Ensemble of **N** different models



THANK YOU