
Our Neural Network

Deep Learning basic

Announcement of Kangnam University in the Second semester of 2023

강동현, 박상진, 박은빈



01 introduction

02 research method

03 research results

04 Q&A

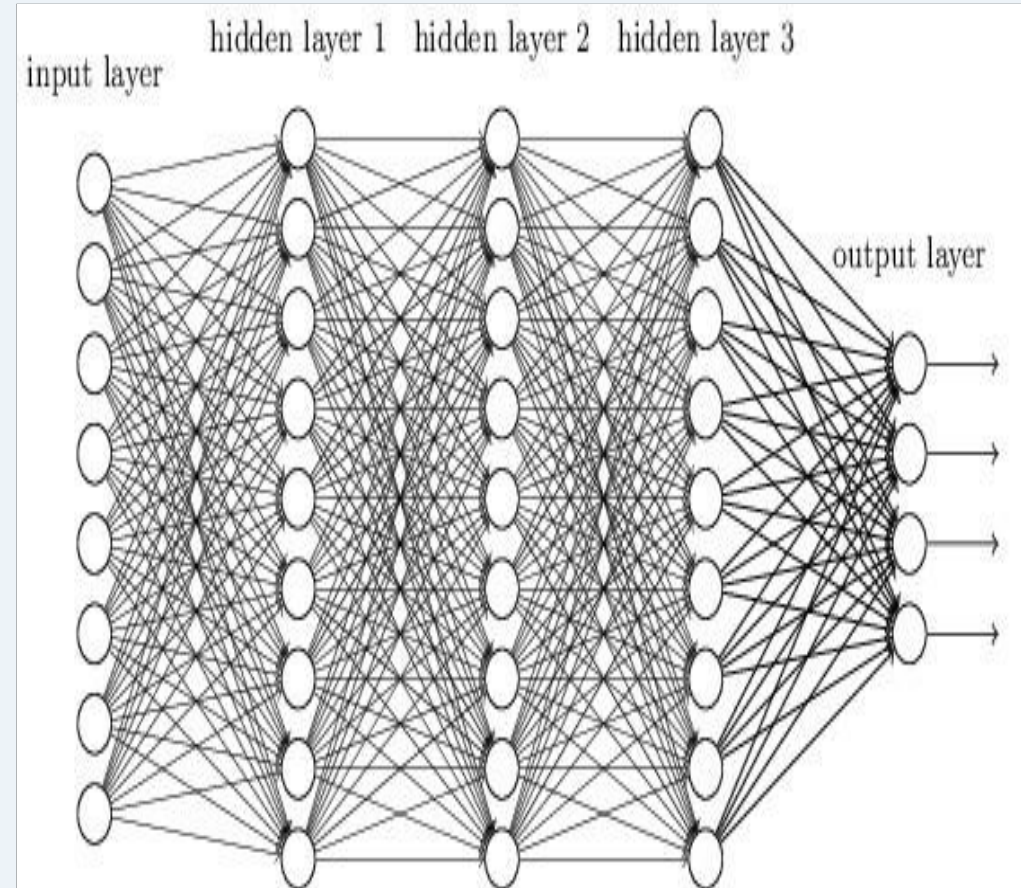
01

introduction

- purpose of study
- data

purpose of study

- Fashion-MNIST 데이터셋에 대해, 교재에서 학습한 최선의 신경망 구조와 학습 방법을 적용하여, 팀에서 할 수 있는 최고 성능의 Fashion-MNIST 데이터 분류 모델을 제시한다.
- 체계적이고 효과적인 실험을 통해, 제안하는 방법론이 합리적임을 입증한다.
- 배치 정규화를 사용하지 않고, 간단한 구조의 모델로 좋은 성능을 내는 자기 정규화 모델을 실험한다.



Fashion-MNIST

Fashion-MNIST는 머신 러닝 및 딥러닝 분야에서 널리 사용되는 이미지 분류 데이터셋으로, 특히 의류와 액세서리 관련 문제를 다루기 위해 제작되었다. 이 데이터셋은 28x28 크기의 흑백 이미지로 구성되어 있으며, 전체적으로 10개의 클래스로 나뉘어져 각각은 다양한 의류 아이템을 나타낸다.



Ankle Boot	6000	Ankle Boot	1000
T-shirt/top	6000	Pullover	1000
Dress	6000	Trouser	1000
Pullover	6000	Shirt	1000
Sneaker	6000	Coat	1000
Sandal	6000	Sandal	1000
Trouser	6000	Sneaker	1000
Shirt	6000	Dress	1000
Coat	6000	Bag	1000
Bag	6000	T-shirt/top	1000

02

Research method

- DataFrame sharing
- Self-Normalizing Neural Networks
- Data standardization
- regularization

Experiment

1. 실험에 사용한 모델의 하이퍼 파라미터, accuracy, 그리고 시간을 기록
2. Google Drive에 공유하여 다른 실험을 시작하기 전에 하이퍼 파라미터 설정에 도움
3. 동일한 seed 값을 설정하여 매 실험 결과를 동일하게 확인

	hidden_size_list	activation	weight_init_std	weight_decay_lambda	use_dropout	dropout_ratio	use_batchnorm	iters_num	batch_size	learning_rate	optimizer	train_acc	test_acc	time
0	[512, 256, 128]	selu_tmp	xavier	0.000002	True	0.30	False	10000	512	0.01	Momentum	0.980167	0.9062	824.248010
1	[256, 128]	selu_tmp	xavier	0.000000	True	0.25	False	10000	512	0.01	Momentum	0.978617	0.9057	346.187427
2	[512, 256, 128]	elu	he	0.000000	False	NaN	True	10000	512	0.01	Nadam	1.000000	0.9038	949.484395
3	[256, 64]	selu_tmp	xavier	0.000000	True	0.25	False	10000	512	0.01	Momentum	0.974417	0.9030	340.173420
4	[256, 32]	selu_tmp	xavier	0.000000	True	0.25	False	10000	512	0.01	Momentum	0.970800	0.9025	340.128564
...
87	[128, 32]	elu	he	0.000000	True	0.50	False	10000	512	0.01	Momentum	0.853458	0.8415	41.790797
88	[300, 150, 70]	sigmoid	xavier	0.000000	False	NaN	True	10000	256	0.01	SGD	0.864433	0.8388	182.931918
89	[128, 32]	elu	he	0.000000	True	0.50	False	10000	512	0.01	Momentum	0.849437	0.8335	34.266119
90	[200, 100, 50]	sigmoid	xavier	0.000000	False	NaN	True	10000	256	0.01	Adam	0.745667	0.7271	162.076558
91	[128, 32]	elu	he	0.000000	True	0.50	False	10000	512	0.01	Momentum	0.653229	0.6455	5.206525

92 rows × 14 columns

Self-Normalizing Neural Network

활성화 함수의 선택과 초기화 방법을 통해 네트워크가 자동으로 정규화되는 특징을 갖는 인공 신경망

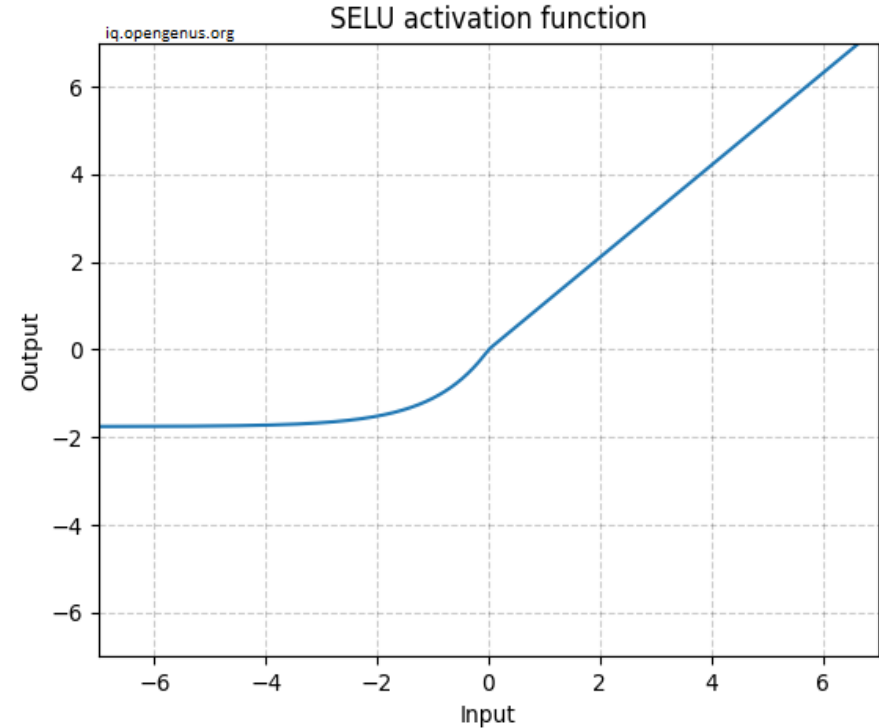
- SELU 활성화 함수
- Lecun 초기화
- 자기 정규화 특성
- 학습률 스케줄링

SELU

- SELU는 입력 데이터의 평균이 0이고 표준편차가 1이 되도록 수렴
- 활성화 함수식

$$f(x) = \begin{cases} \lambda x, & x \geq 0 \\ \lambda \alpha (e^x - 1), & x < 0 \end{cases}$$

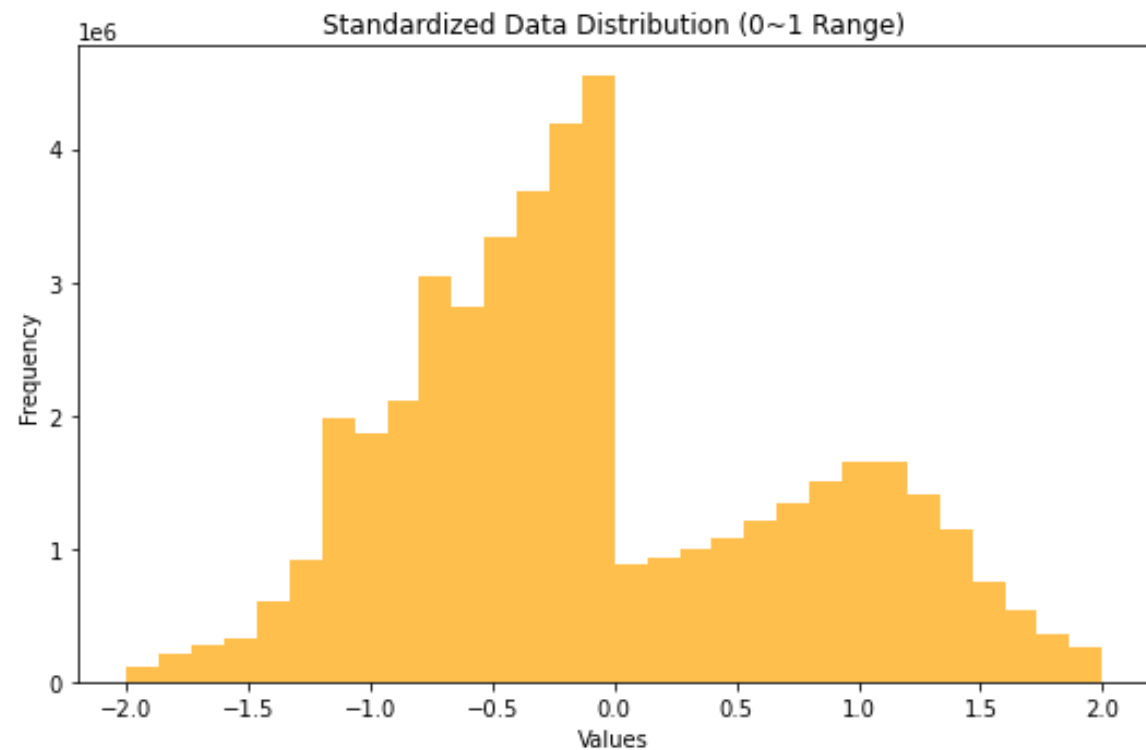
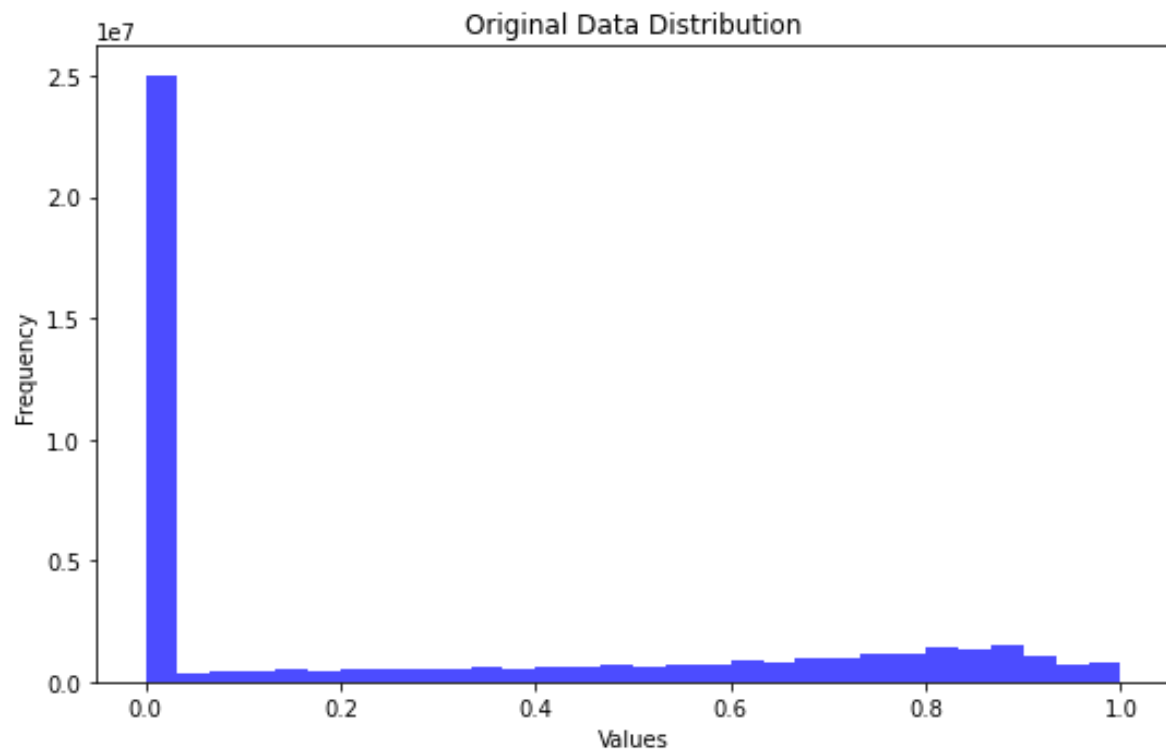
with $\alpha \approx 1.6733$ and $\lambda \approx 1.0507$



Normalize vs standardization

정규화: 데이터를 특정 범위로 변환하여 비교 가능하게 만드는 것이 목적

표준화: 데이터를 평균이 0이고 표준편차가 1인 분포로 변환하여 스케일링을 조절하는 것이 목적



regularization

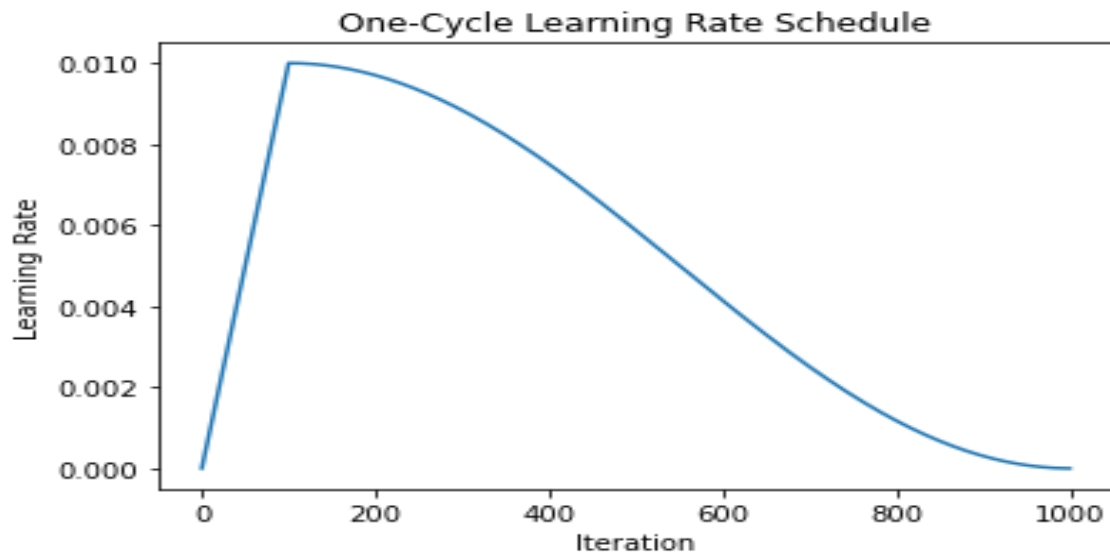
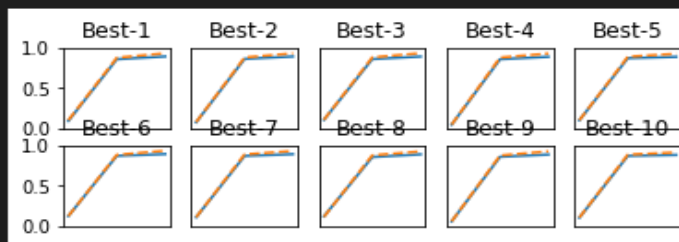
Alpha Dropout

- 자기 정규화 네트워크를 규제하고자 할 때, 일반 드롭아웃이 자기 정규화 기능을 방해
- 출력 값에 $(1 - \text{드롭아웃 확률})$ 을 곱하여 입력의 평균과 표준편차를 유지

1cycle scheduling

- 모델이 빠르게 수렴하면서도 안정적으로 학습하도록 도와주는 방법
- 학습률이 선형적으로 증가했다가, 일정 구간이 지나면 선형적으로 감소

```
===== Hyper-Parameter Optimization Result =====
Best-1(val acc:0.8920833333333333) | weight decay: 2.008200346026684e-07
Best-2(val acc:0.8918333333333334) | weight decay: 5.934726196444554e-07
Best-3(val acc:0.8911666666666667) | weight decay: 2.007712765963896e-07
Best-4(val acc:0.8878333333333334) | weight decay: 6.020991903738259e-08
Best-5(val acc:0.8874166666666666) | weight decay: 2.6139565619750825e-07
Best-6(val acc:0.8868333333333334) | weight decay: 2.2167598465839503e-07
Best-7(val acc:0.8860833333333333) | weight decay: 2.994715876079184e-06
Best-8(val acc:0.88375) | weight decay: 1.393546942970559e-07
Best-9(val acc:0.881) | weight decay: 4.45563082275952e-08
Best-10(val acc:0.87625) | weight decay: 7.855914588019412e-06
```



03

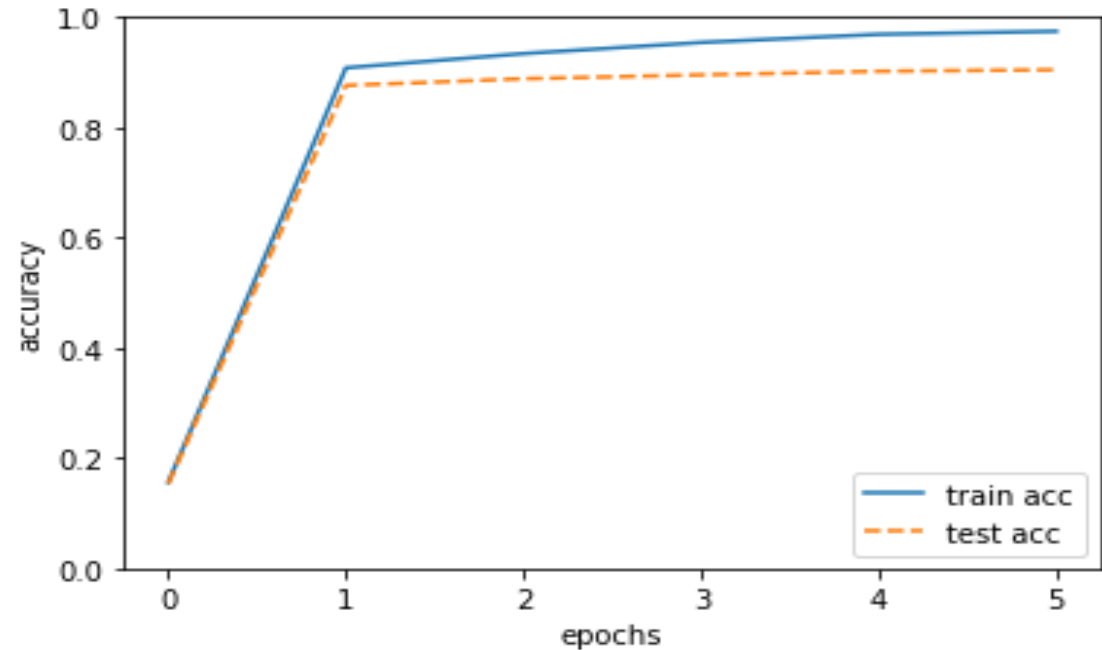
Research results

- Model architecture & result
- Detail & summary

Model architecture & result

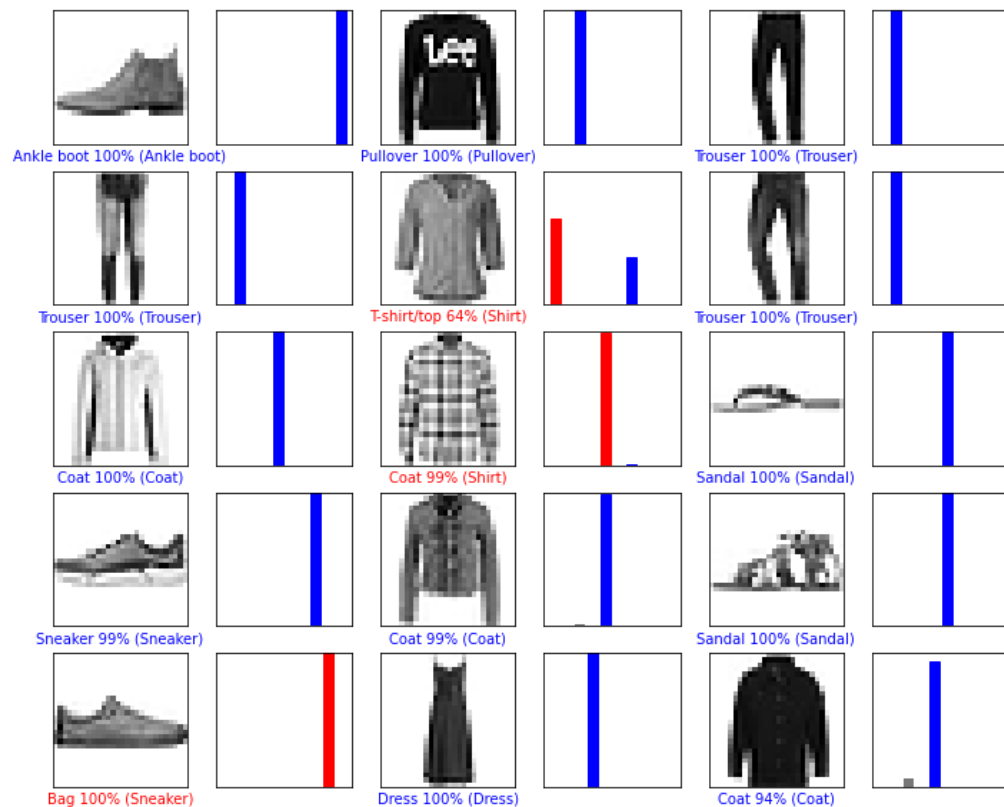
Final Network

- Input size = 784
- Hidden size list = [256, 64]
- Output size = 10
- Activation function = SELU
- Weight init: Lecun(Xavier)
- Alpha Dropout(p=0.25)
- Weight decay lambda = 0.000002
- Iteration = 10,000
- Batch size = 512
- Optimizer = Momentum
- Scheduling: 1 cycle scheduling



Train accuracy: 0.974467 Test accuracy: 0.9045

Detail & summary



Accuracy: 0.8019

CR:	precision	recall	f1-score	support
0	0.90	0.70	0.79	1293
1	0.98	0.96	0.97	1015
2	0.55	0.83	0.66	658
3	0.69	0.97	0.81	713
4	0.94	0.54	0.69	1739
5	0.76	1.00	0.86	758
6	0.40	0.70	0.51	568
7	0.84	0.93	0.88	906
8	0.98	0.85	0.91	1144
9	0.98	0.82	0.89	1206
accuracy			0.80	10000
macro avg	0.80	0.83	0.80	10000
weighted avg	0.85	0.80	0.81	10000

Summary

- ① 배치 정규화를 적용하지 않은 3층 모델로 좋은 성능을 도출
- ② 과적합 문제를 완전히 해결한 것은 아님
- ③ SELU activation function에서 지수 함수의 연산이 계산 비용을 높임

Thank you

Thank you for listening to my
presentation

강동현, 박상진, 박은빈