Sparsity Normalization:
Stabilizing the Expected Outputs of Deep Networks

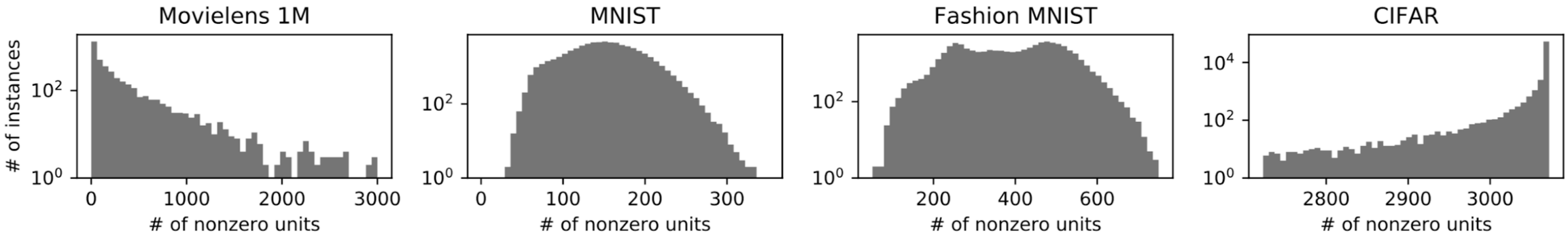
2019. 06. 07.

Joonyoung Yi
joonyoung.yi@kaist.ac.kr

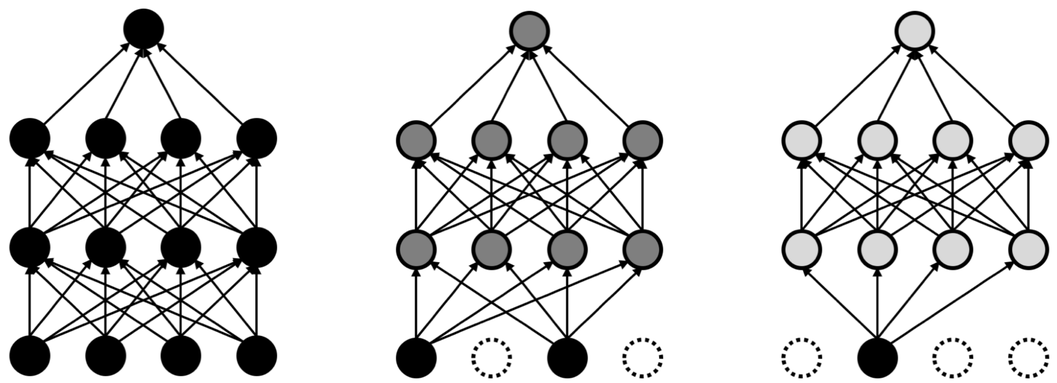
KAIST

VARIABLE SPARSITY PROBLEM

- Many benchmark datasets ***differ in the sparsity*** between the data instances.



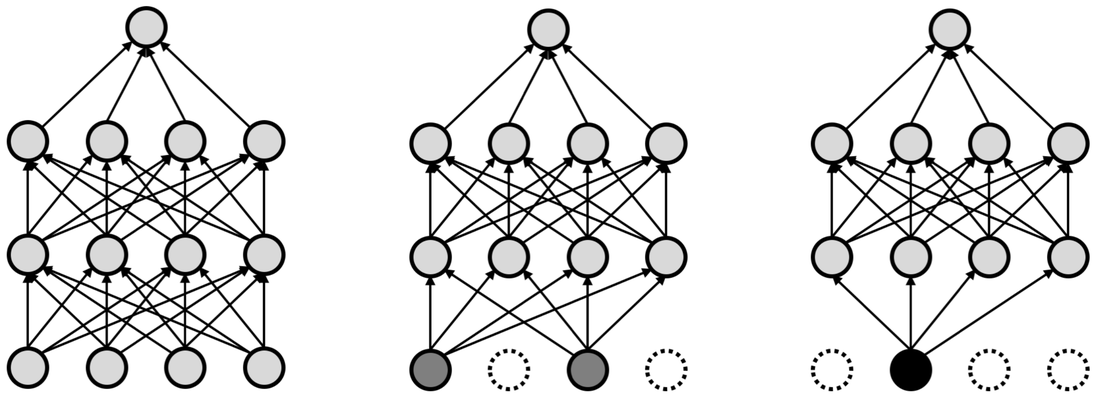
- ***Variable sparsity problem***: the expected value of the output layer depends on
  the sparsity of the input data instance which makes the training difficult.
- Varying outputs for data instances with similar characteristics under different sparsity.



(a) Standard Neural Net with variable sparsity

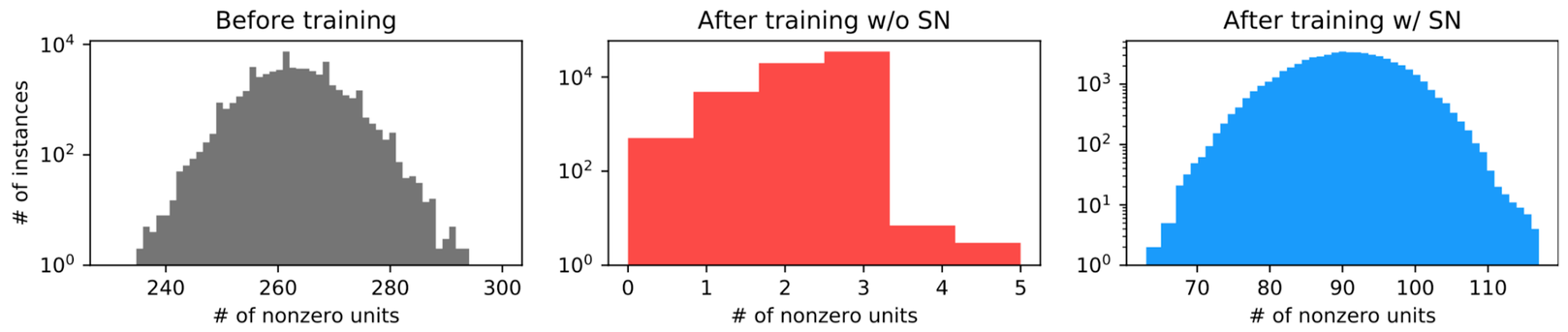
SPARSITY NORMALIZATION

- ***Divide*** each input data instance ***by l0***: $\mathbf{x}_{\mathrm{SN}} \leftarrow K'\mathbf{x} \,/\, \|\mathbf{x}\|_0$
  - So that outputs are not dependent on sparsity (can be applied to CNN similarly).



(b) After applying Sparsity Normalization

- ***Sparsity Normalization*** solves ***various sparsity problem*** (theoretically, experimentally).
  - Sparsity in a hidden layer is more stable after applying Sparsity Normalization.

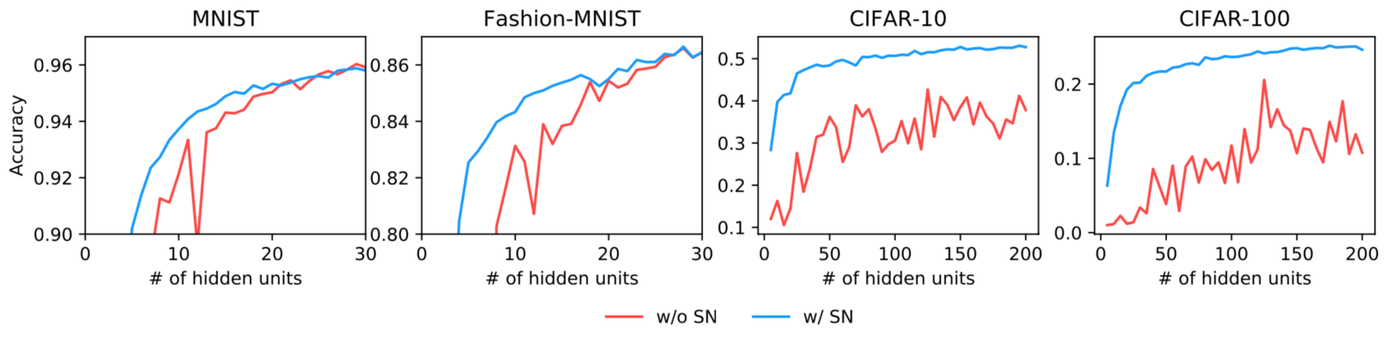


Experimental Results

- Collaborative filtering datasets: ***Achieved states-of-the-arts performance*** on Movielens 100K & 1M ***by simply applying Sparsity Normalization*** to non-states-of-the-arts model.
- Electronic health records (EHR) dataset: Better AUC & ***orthogonal to Dropout*.**

Dataset		Cardiovascular	Fatty Liver	Hypertension	Heart Failure	Diabetes
MLP	w/o SN	0.7057 ± 0.0027	0.6750 ± 0.0050	0.7977 ± 0.0027	0.7834 ± 0.0036	0.9121 ± 0.0097
	w/ SN	**0.7106 ± 0.0005**	**0.6911 ± 0.0022**	**0.8096 ± 0.0010**	**0.7914 ± 0.0012**	**0.9283 ± 0.0011**
MLP+Dropout	w/o SN	0.7084 ± 0.0005	0.6858 ± 0.0065	0.8023 ± 0.0054	0.7876 ± 0.0012	0.9263 ± 0.0026
	w/ SN	**0.7105 ± 0.0009**	**0.6941 ± 0.0011**	**0.8086 ± 0.0016**	**0.7922 ± 0.0015**	**0.9303 ± 0.0029**

- Vision datasets: Better accuracy with less capacity & ***orthogonal to BN.***



	VGG-16	VGG-16 + BN
w/o SN	0.8050 ± 0.0185	0.8755 ± 0.0066
w/ SN	**0.8094± 0.0173**	**0.8800 ± 0.0101**

- 6 UCI datasets: better performance even compared to ***other missing handling techniques.***