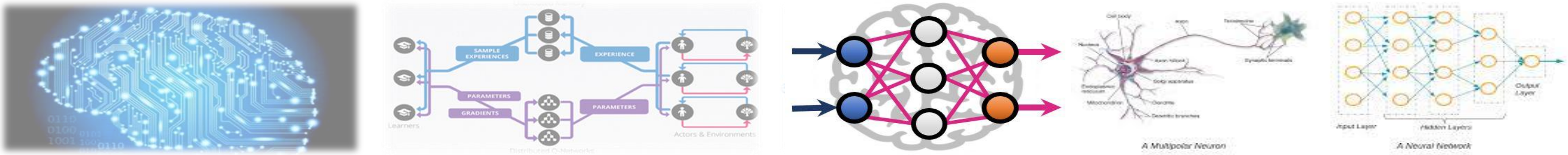


제 132회 석사 학위 졸업 논문 발표

Effective Front-end Architecture Search for Random Weight Network using Particle Swarm Optimization

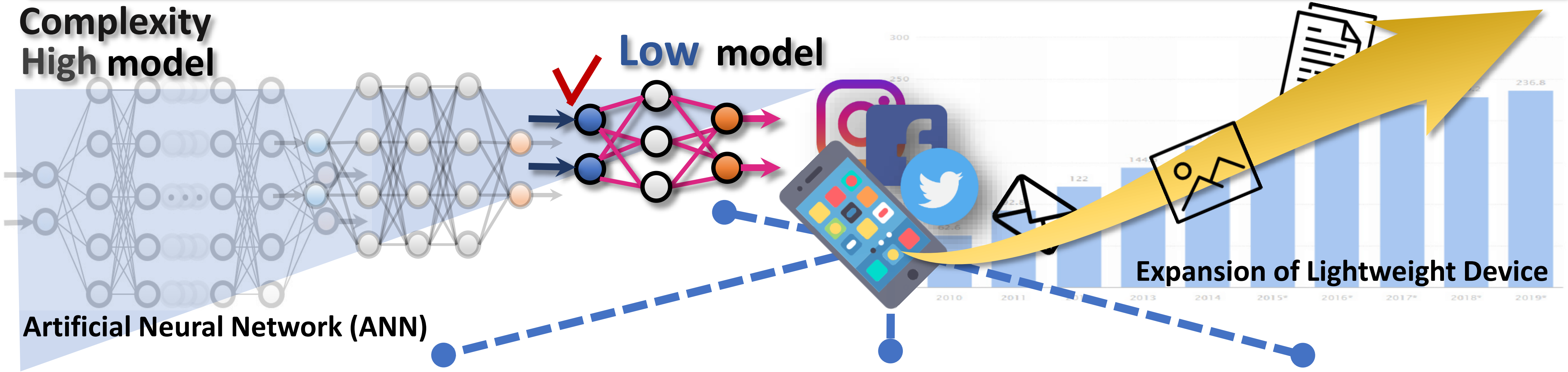
입자 군집 최적화를 이용한 효과적인 무작위 가중치 네트워크 전단 구조 탐색 기법

Jinhyeong Park,

jhpark.4745@gmail.com

2019. 11. 22.

AI with Edge Device



keep Privacy

삼성전자, "내 손안의 AI 대세 될 것"

삼성은 온디바이스 AI에 주목하고 있다.

클라우드 방식의 AI보다 개인화된 정보를 처리하는 데 유리

주목하고 있다. 서버에서 데이터를 처리하는 클라우드 방식의 AI보다 개인화된 정보를 처리하는 데 유리

모든 데이터를 완벽하게 보호

서버, 클라우드 등 다양한 방식으로 AI가 구동됨에 따라 각각의 특성을 잘 활용한 AI

삼성전자

more Stability

LG전자도 시스템반도체…AI칩 내재화

2019.05.16 10:08:54 / 유상호 crow@ddaily.co.kr

수행할 수 있고, 제품 내에서 개인정보에 해당되는 데이터를 안전하게 보호한다.

네트워크가 연결되지 않은 상황에서도 인공지능 기능

고성수 기자 ssgo@naeil.com

LG

more Data

터치보다 빠른 구글 어시스턴트 "1인 1 AI 시대 왔다"

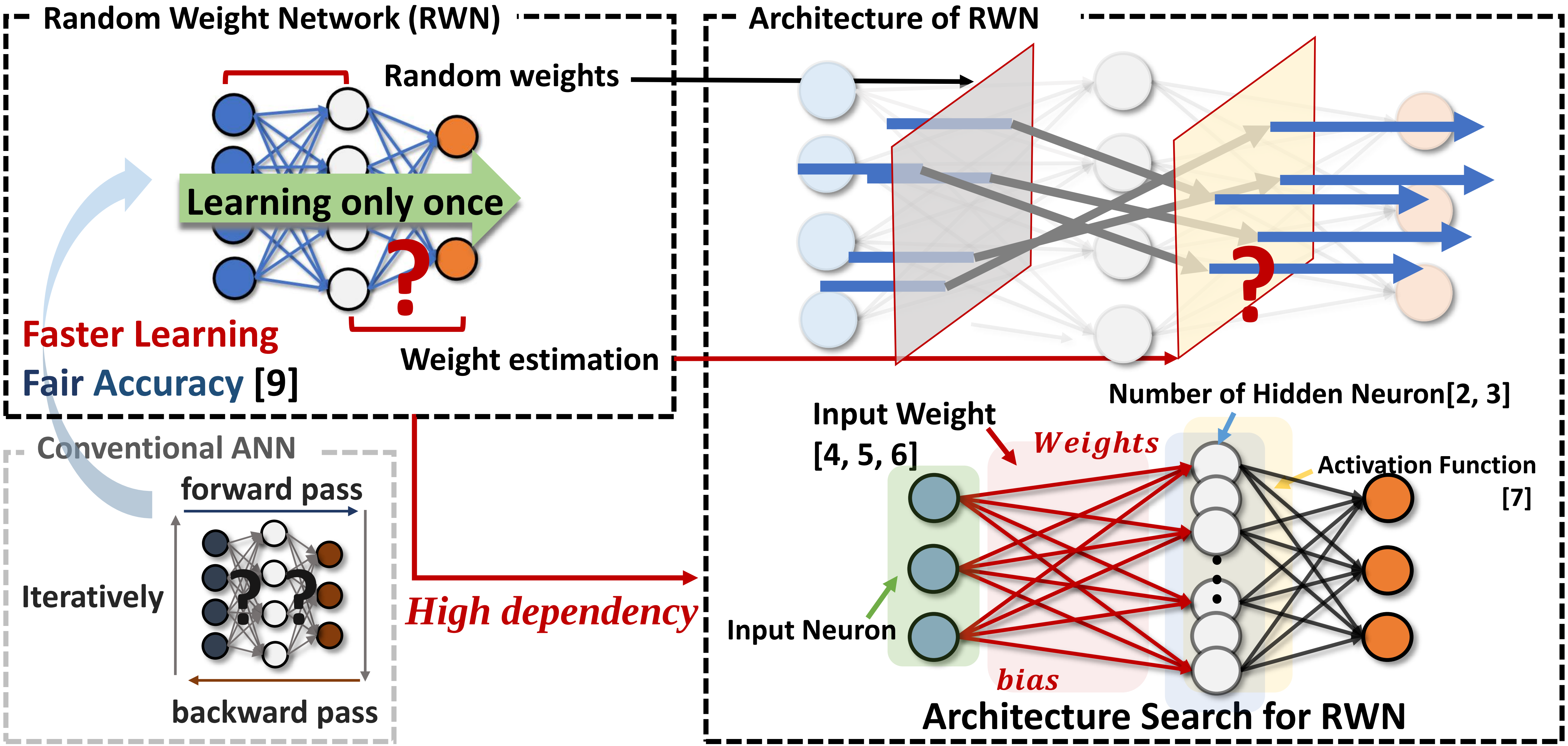
데이터의 양은 인간이 함께 지내는 시간에 비례

빅데이터를 수거하기 위한 장치들이다. 최근에는 인공지능 스피커가 여기

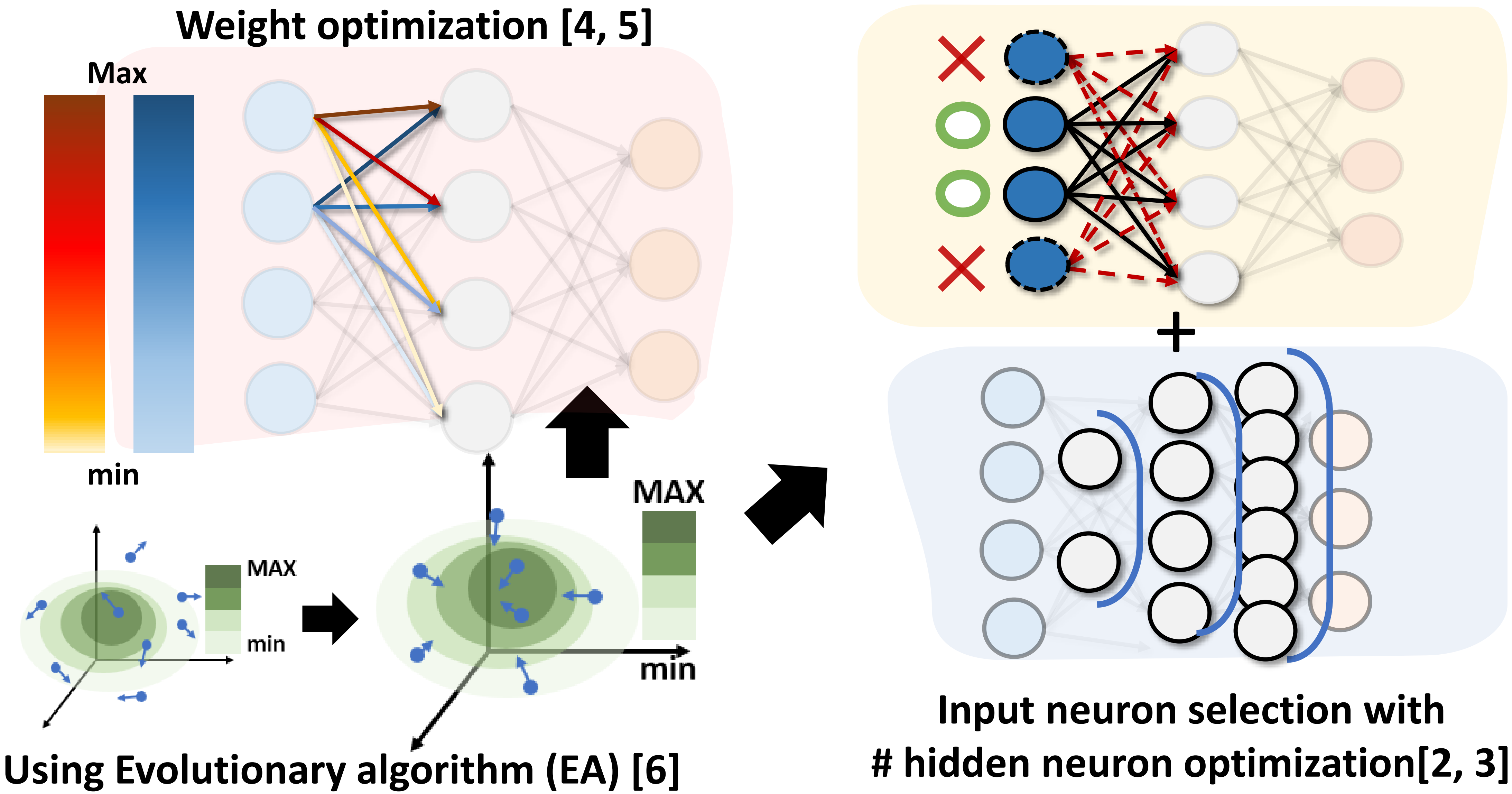
공지능 소프트웨어'가 필요하다.

Google

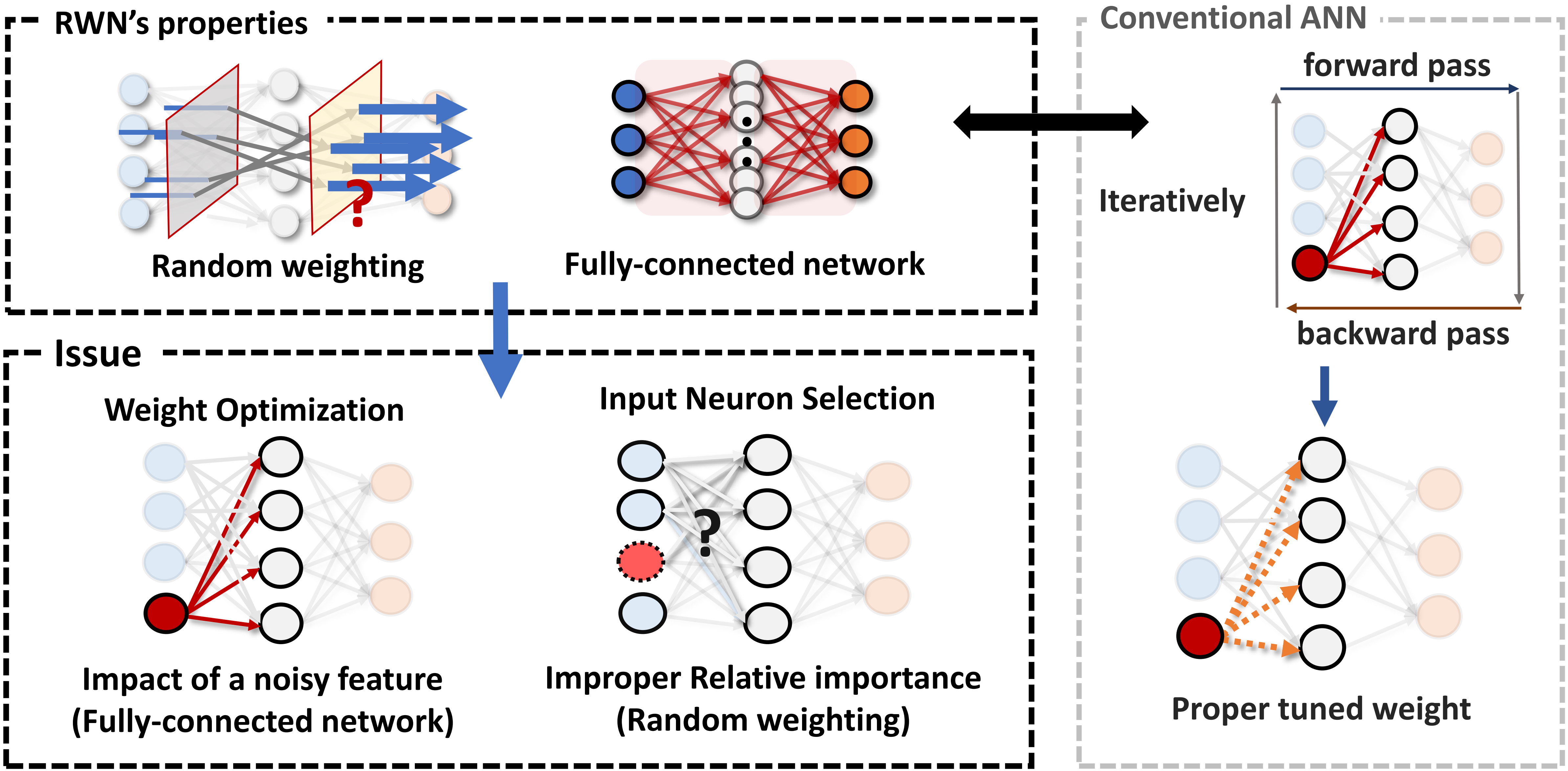
ANN Architecture for Edge Device



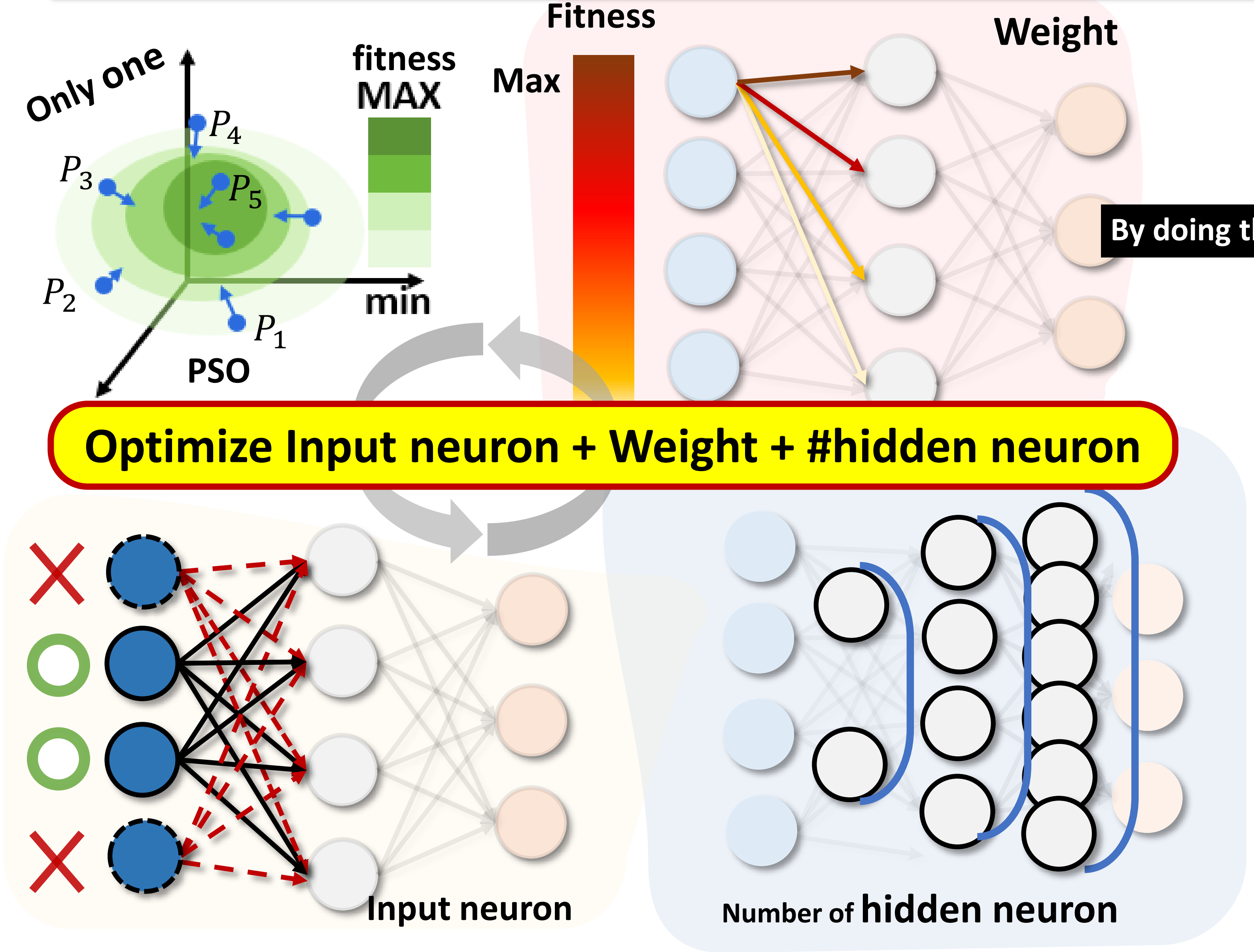
Conventional methods of RWN architecture search



Motivation


RWN's properties
Random weighting
Fully-connected network
Conventional ANN
forward pass
Iteratively
backward pass
Proper tuned weight
Issue
Weight Optimization
Impact of a noisy feature
(Fully-connected network)
Input Neuron Selection
Improper Relative importance
(Random weighting)

Proposed method



By doing this

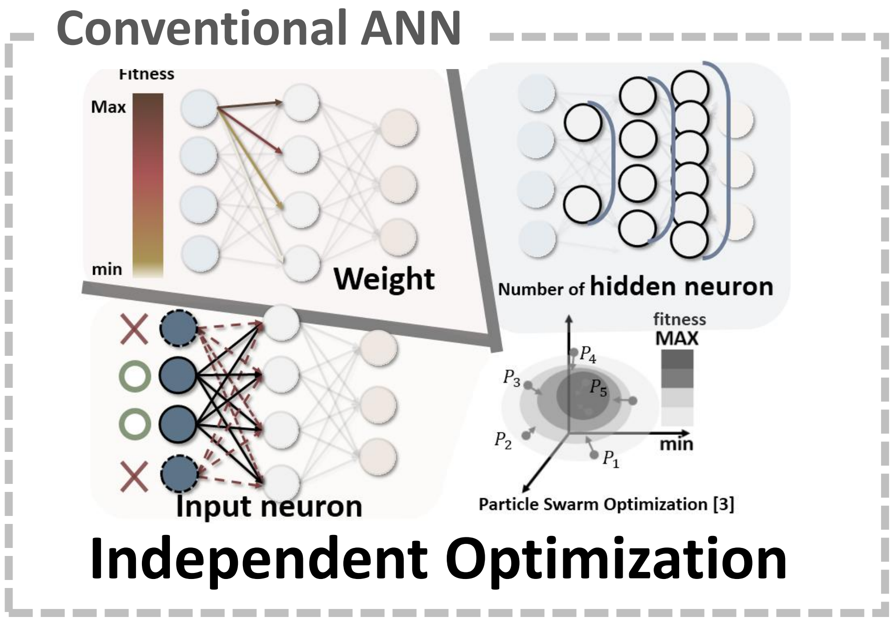
Expected Advantages

Better Learning Accuracy

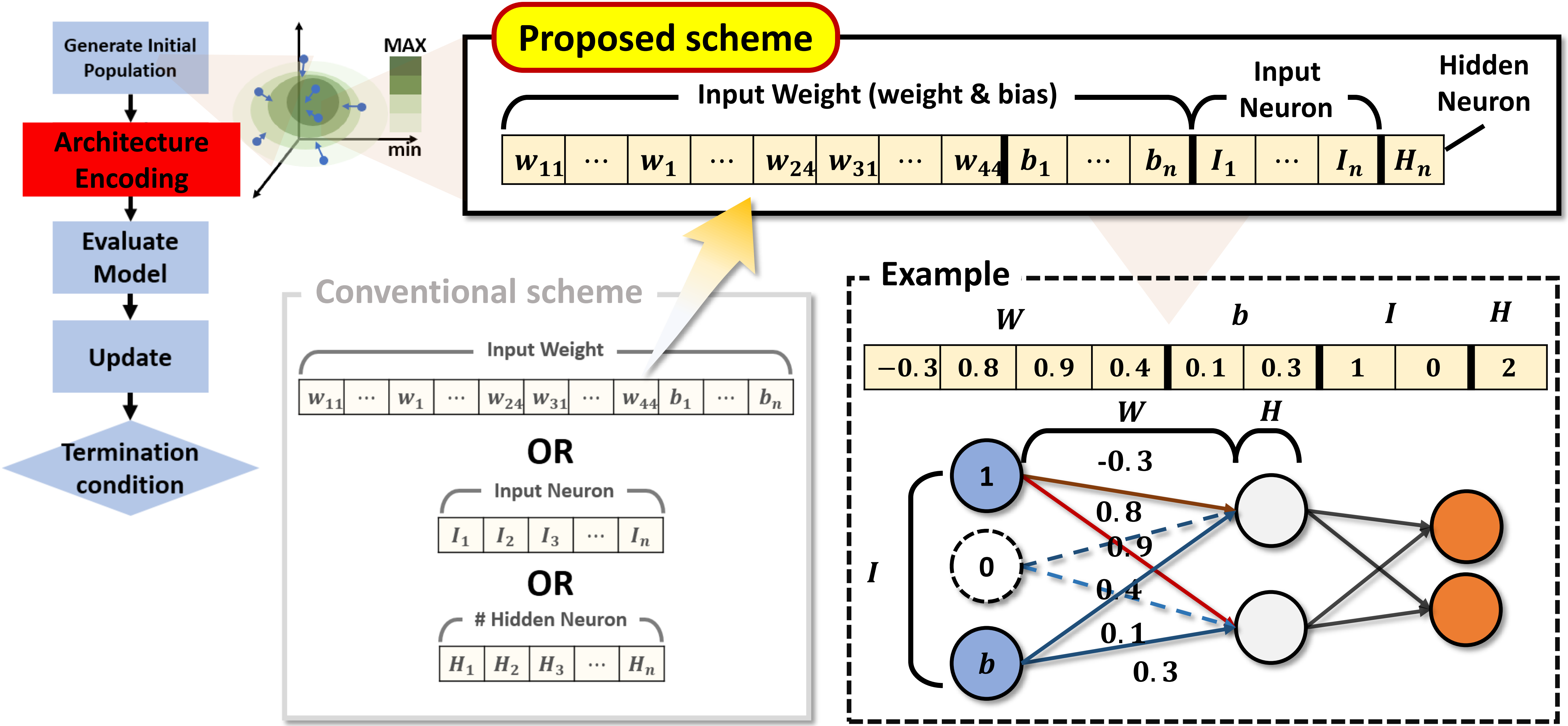
- Eliminating noisy input neurons
- Setting proper weights

In addition, …

- Low computational cost (LWD)



Proposed method: Encoding scheme



Experimental Setting

Datasets Description

- **10 multi-label text datasets from RCV1 and Yahoo collections**

Datasets	Pattern Size	Feature Size	Label Size	Cardinality	Distinct.
RCV1(S1)	6,000	945	101	2.880	1028
RCV1(S2)	6,000	945	101	2.634	954
RCV1(S3)	6,000	945	101	2.614	939
RCV1(S4)	6,000	945	101	2.484	816
Arts	7,484	1,157	26	1.654	599
Business	11,214	1,096	30	1.599	233
Education	12,030	1,377	33	1.463	511
Enron	1,702	1,001	53	3.378	753
Medical	978	1,494	45	1.245	94
Society	14,512	1,590	27	1.670	1054

Experimental Setting

Comparison algorithm

- IPE-ELM[3] : Input neuron selection and # hidden neuron optimization
- SaELM[4] : Input weight optimization
- CSO-ELM[5] : Input weight optimization

Performance Measure

$$hloss(T) = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{|L|}|\lambda_i \triangle Y_i|$$

Evaluates the fraction of misclassified instance-label pairs

➡ **Lower value, Higher Performance**

$$mlacc(T) = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{|\lambda_i \cap Y_i|}{|\lambda_i \cup Y_i|}$$

Evaluates the overall effectiveness of a model

➡ **Higher value, Higher Performance**

[3] Tansel, D., Ender, S. (**2019**). Evolutionary parallel extreme learning machines for the data classification problem. *Computers & Industrial Engineering*., 130:237-249.

[4] Nahvi, B., Habibi, J., Mohammadi, K., et al. (**2016**). Using self-adaptive evolutionary algorithm to improve the performance of an extreme learning machine for estimating soil temperature. *Comput. Electron. Agric.*, 124:150–160.

[5] Eshtay, M., Faris, H., Obeid, N. (**2018**). Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. *Expert Syst. Appl.*, 104:134-152.

Experimental Result

- Comparison results of four compared methods in terms

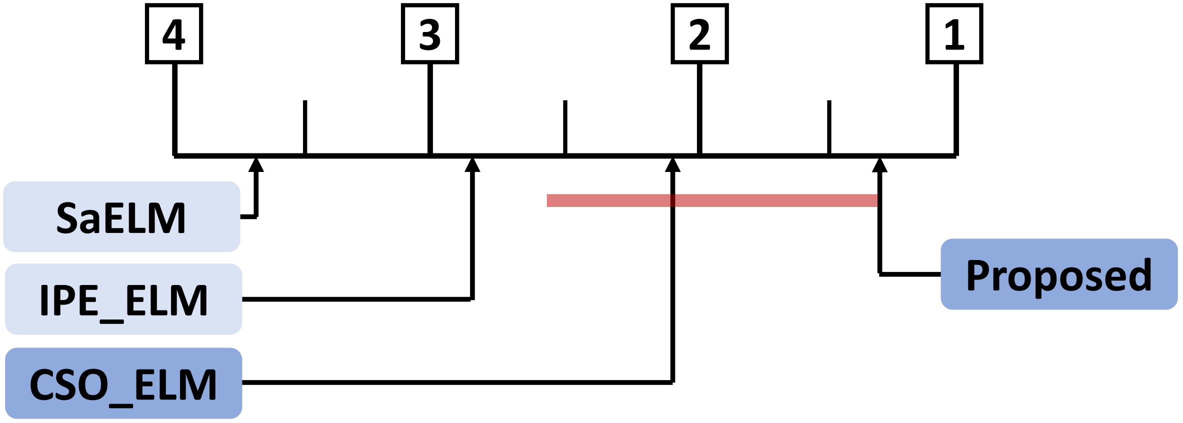
Datasets	Hamming loss				Multi-label accuracy			
	Proposed	**IPE-ELM[2]**	**SaELM[3]**	**CSO-ELM[4]**	**Proposed**	**IPE-ELM[2]**	**SaELM[3]**	**CSO-ELM[4]**
RCV1(S1)	**0.037±0.001**	0.044±0.001	0.044±0.001	0.042±0.001	**0.245±0.007**	0.155±0.010	0.153±0.009	0.163±0.007
RCV1(S2)	**0.035±0.001**	0.037±0.001	0.039±0.001	0.037±0.001	**0.281±0.012**	0.159±0.005	0.154±0.006	0.160±0.001
RCV1(S3)	**0.036±0.001**	0.038±0.001	0.038±0.001	0.037±0.002	**0.282±0.014**	0.161±0.007	0.159±0.008	0.165±0.005
RCV1(S4)	**0.033±0.000**	0.036±0.002	0.036±0.001	0.034±0.001	**0.306±0.219**	0.182±0.006	0.178±0.007	0.186±0.007
Arts	**0.087±0.003**	0.098±0.003	0.099±0.003	0.093±0.002	**0.315±0.005**	0.221±0.009	0.218±0.009	0.230±0.009
Business	0.030±0.001	0.030±0.001	**0.029±0.001**	0.030±0.001	0.677±0.008	0.676±0.008	**0.678±0.009**	0.676±0.010
Education	0.058±0.002	0.060±0.001	0.060±0.000	**0.057±0.002**	**0.348±0.002**	0.255±0.009	0.252±0.008	0.255±0.006
Enron	**0.056±0.002**	0.057±0.001	0.059±0.001	0.058±0.001	**0.428±0.012**	0.408±0.009	0.408±0.013	0.407±0.011
Medical	**0.016±0.002**	0.020±0.002	0.022±0.002	0.022±0.001	**0.640±0.027**	0.532±0.009	0.503±0.030	0.502±0.040
Society	0.062±0.001	0.062±0.001	0.062±0.000	**0.062±0.001**	**0.380±0.010**	0.370±0.009	0.370±0.010	0.371±0.010
Avg. rank	**1.30**	2.80	3.70	2.20	**1.1**	2.9	3.5	2.5

Experimental Result

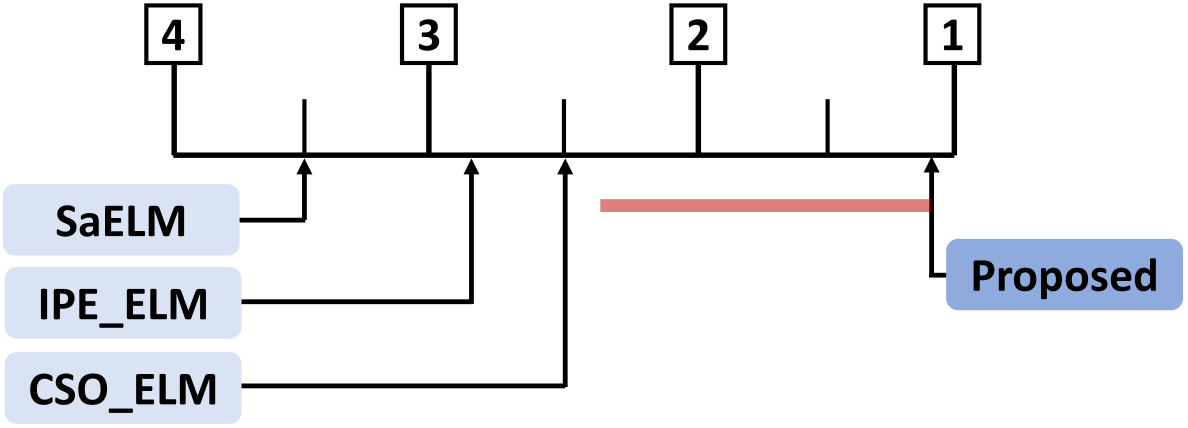
1. Friedman test

Evaluation measure	Friedman statistics	Critical values($\alpha = 0.05$)
Hamming loss	14.196	3.0724
Multi-label accuracy	14.936	

2. Bonferroni-Dunn test



(a) Hamming Loss

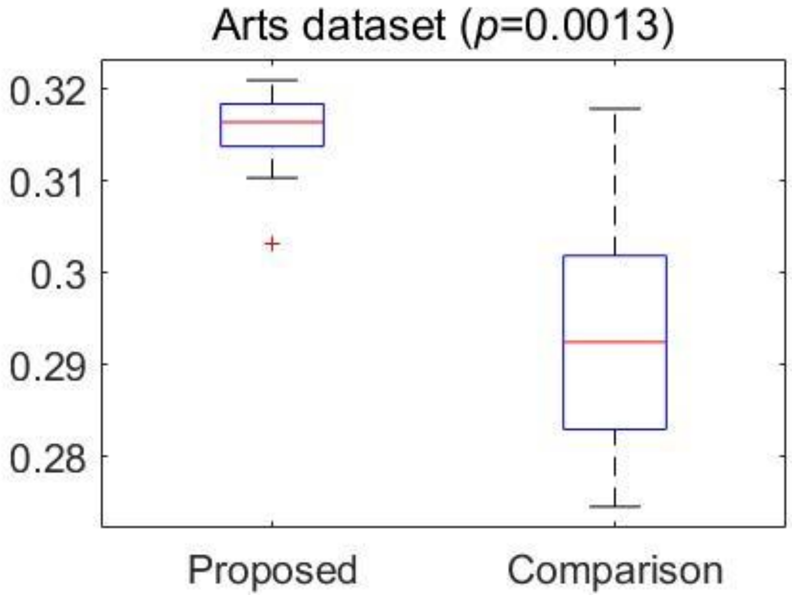


(b) Multi-label Accuracy

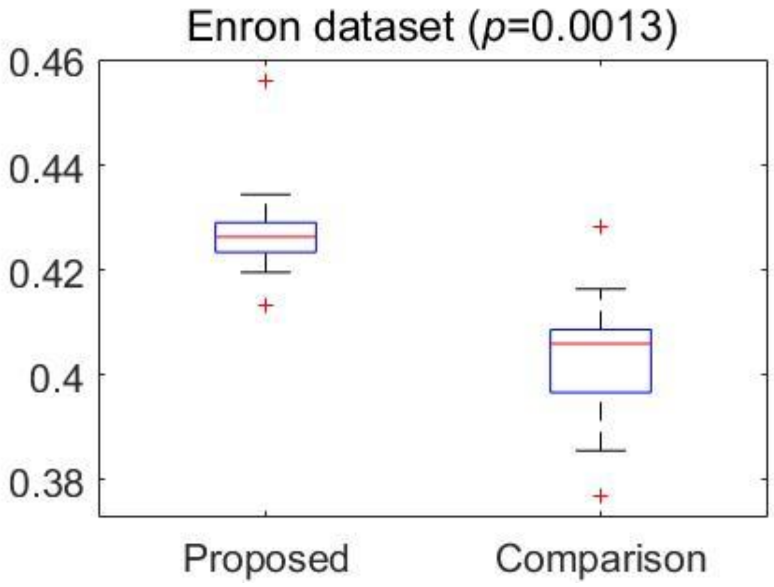
Analysis

Comparison results between two methods in multi-label accuracy

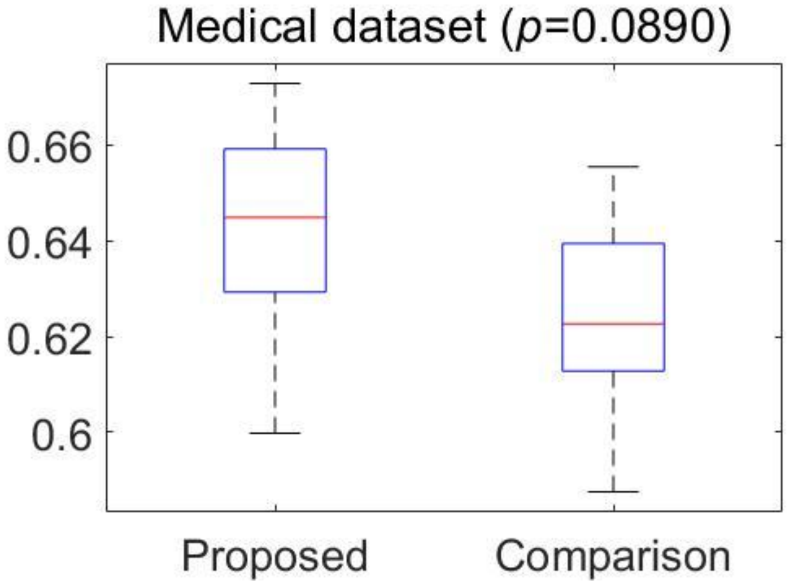
- **Proposed : Input neuron + Input weight + # of hidden neuron optimization**
- **Comparison : Input weight + # of hidden neuron optimization**



(a) Arts dataset



(b) Enron dataset



(c) Medical dataset

Example

Final architecture!

Removed neurons

pituitary
00
rounded
tonsils
rsv
x2
12-day
margin

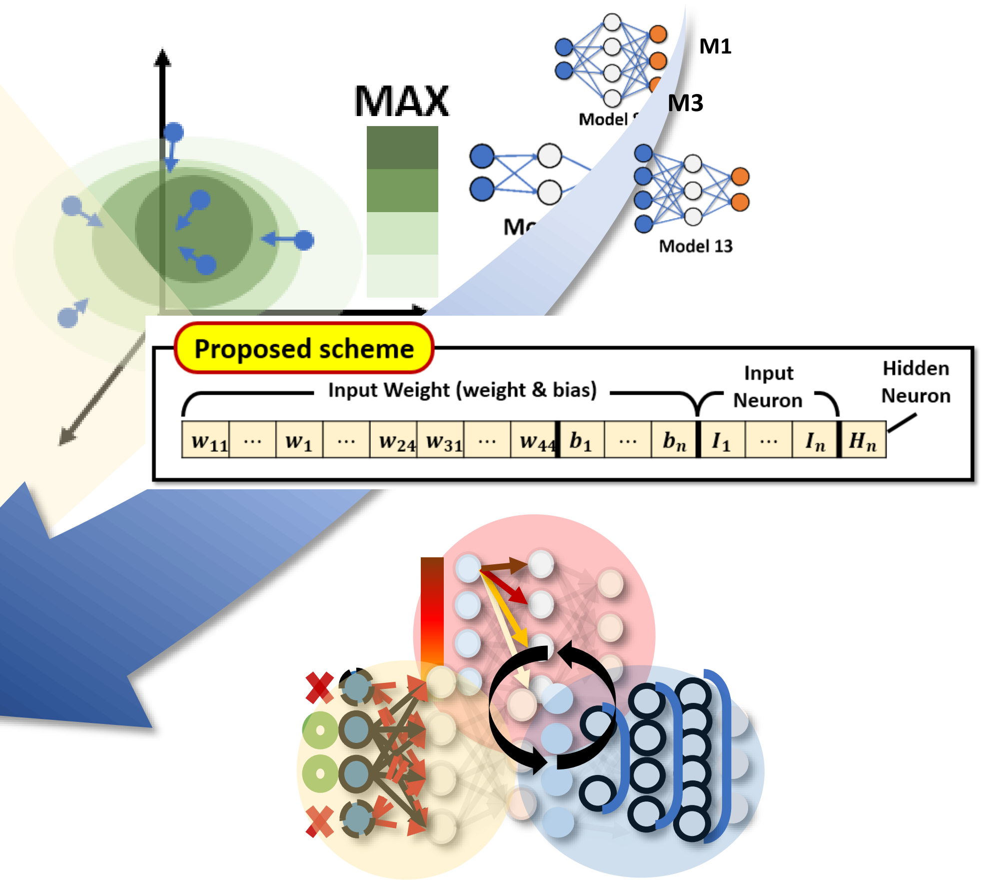
# of hidden neurons

340/724

Weight Top 10

bronchiectasis
filled
bladder
pyeloplastys
mother
region
obscuring
Scoliosis
Small
reflux

Datasets	Pattern Size	Feature Size	Label Size
Medical	978	1,449	45



Conclusion & Contribution

- Improved Architecture search method for RWN by combining input neuron selection with conventional methods.
- The experimental result show that the proposed method also can applied well to the multi-label classification problem.

- **Contribution**

  In International Publication (SCI(E))

  -Compact Feature Subset-based Multi-label Music Categorization for Mobile Devices, **Multimedia Tools and Applications**, 2019, **Co-author**

  In International Conferences

  -Multi Population Memetic Search for Effective Multi-label Feature Selection, **2019 Int. Conf. on Platform Technology and Service**, 2019, **1st author**

  In Domestic Conferences

  -Evolutionary Algorithm Design for Effective Multi-label Feature Selection, **The SEBS Spring Conf**, 2018, **1st author**

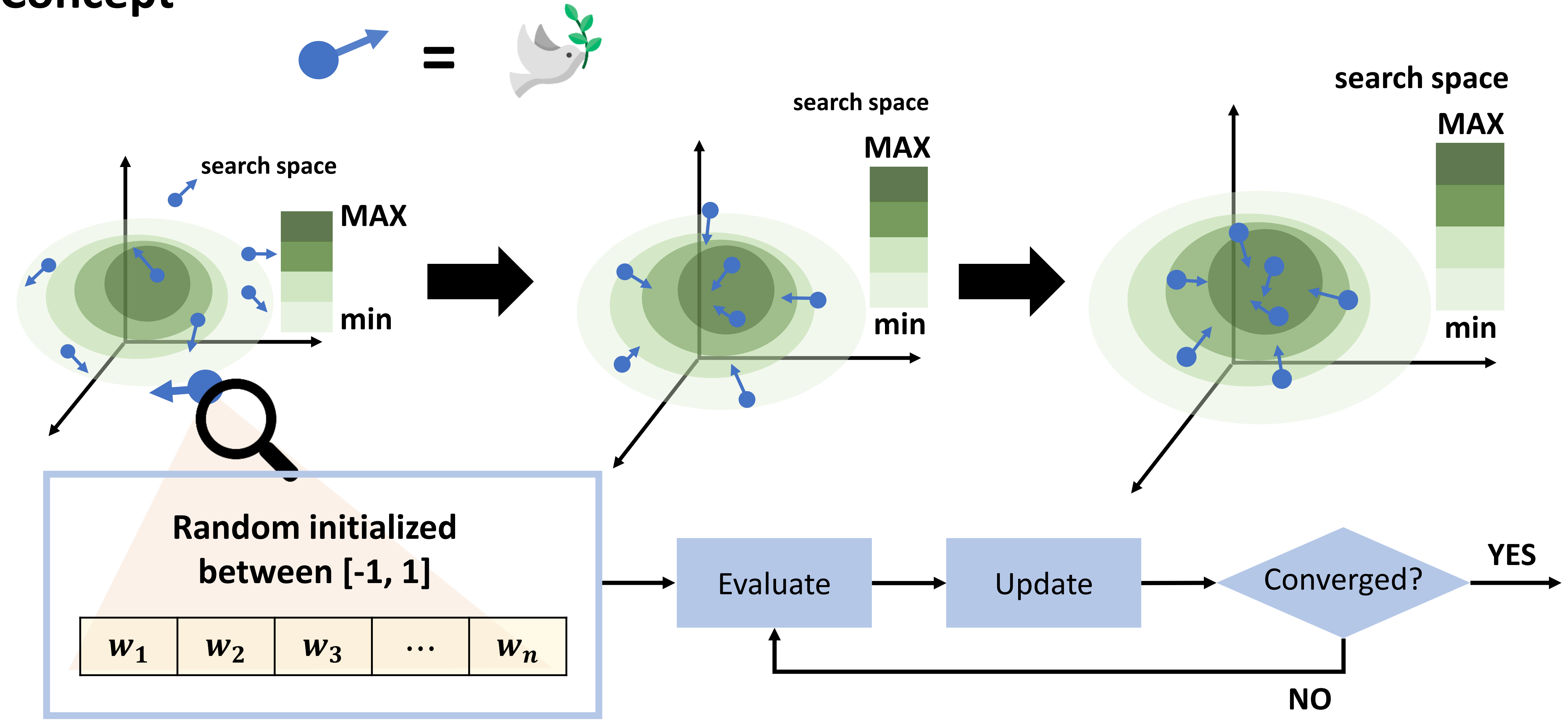
Thank you

Reference

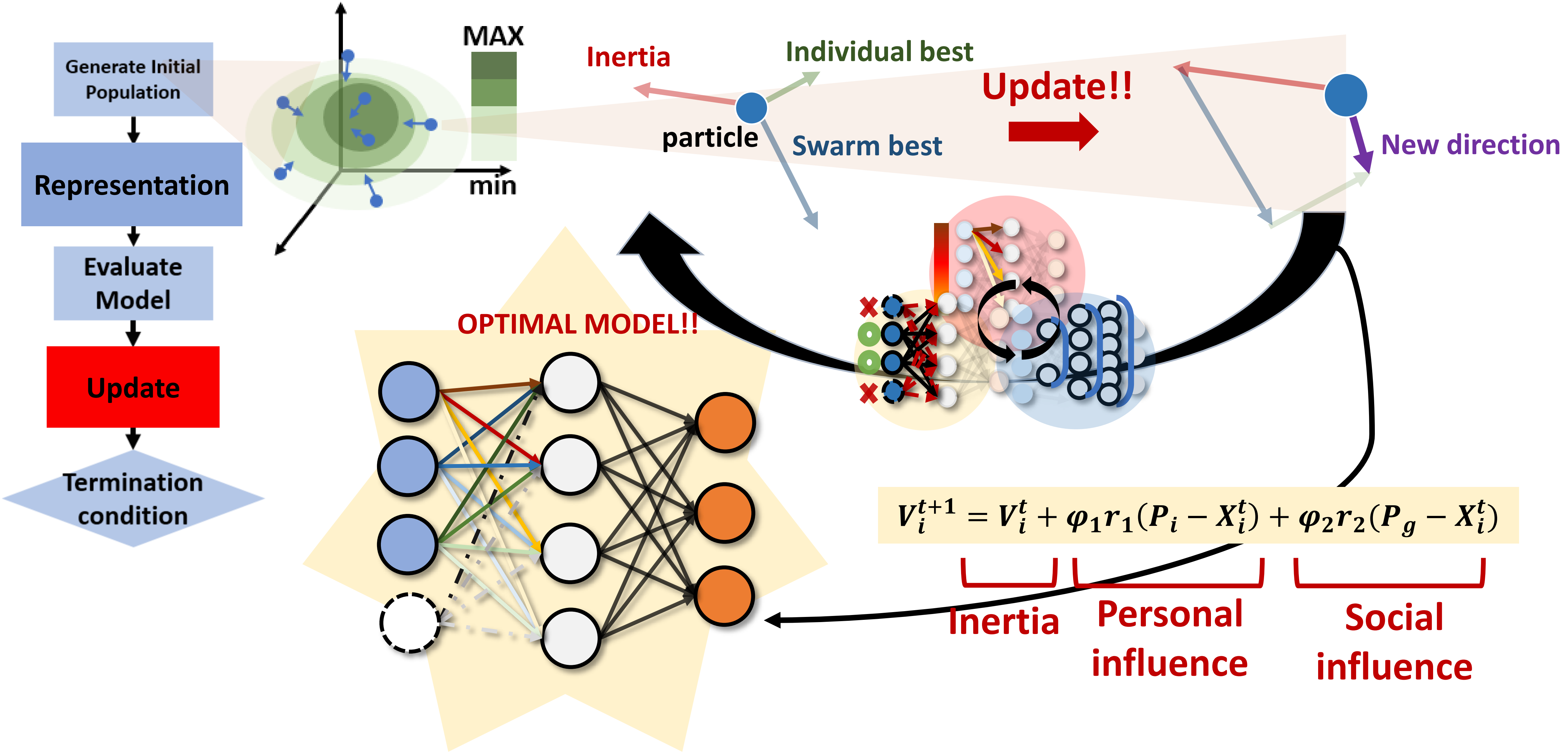
[1] Yunbin, D. (**2019**). Deep learning on mobile devices: a review. *Mobile Multimedia/Image Processing, Security, and Appl.* 2019., 109930A
[2] Hossam, F., Ala, M.A., Ali, A., et al. **(2019)**. An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks. *Information Fusion.*, 48:67-83
[3] Tansel, D., Ender, S. (**2019**). Evolutionary parallel extreme learning machines for the data classification problem. *Computers & Industrial Engineering.*, 130:237-249.
[4] Nahvi, B., Habibi, J., Mohammadi, K., et al. (**2016**). Using self-adaptive evolutionary algorithm to improve the performance of an extreme learning machine for estimating soil temperature. *Comput. Electron. Agric.*, 124:150–160.
[5] Eshtay, M., Faris, H., Obeid, N. (**2018**). Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. *Expert Syst. Appl.*, 104:134-152.
[6] Mohammed, E., Hossam, F., Nadim O. (**2019**). Metaheuristic-based extreme learning machines: a review of design formulations and applications. *International Journal of Machine Learning and Cybernetics.*, 10:1543-1561.
[7] Ojha, V. K., Abraham, A., Snasel, V. (**2017**). Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng. Appl. Artif. Intell.*, 60:97-116.
[8] Mohammed, E., Hossam, F., Nadim, O. (**2013**). Genetically optimized extreme learning machine. ETFA. *IEEE 18th conference on IEEE.*, 10:1543-1561.
[9] Guangbin, H., Qinyu, Z., Cheekheong, S. (**2006**). Extreme learning machine: Theory and applications. *Neurocomputing.*, 70:489-501.

Appendix: Particle Swarm Optimization (PSO)

Concept



Appendix: PSO Update



$$V_i^{t+1} = V_i^t + \varphi_1 r_1 (P_i - X_i^t) + \varphi_2 r_2 (P_g - X_i^t)$$

Appendix: Encoding scheme