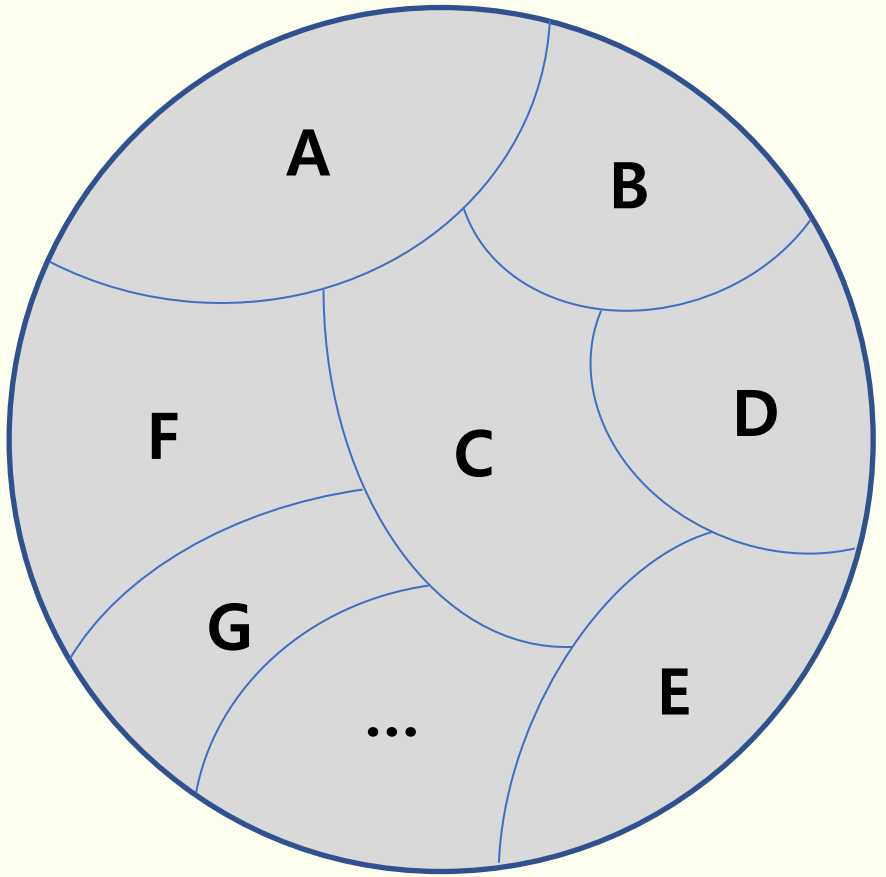
Towards Open Set Deep Networks

Juhyeong Lee, Undergraduate Researcher @ MLC Lab.

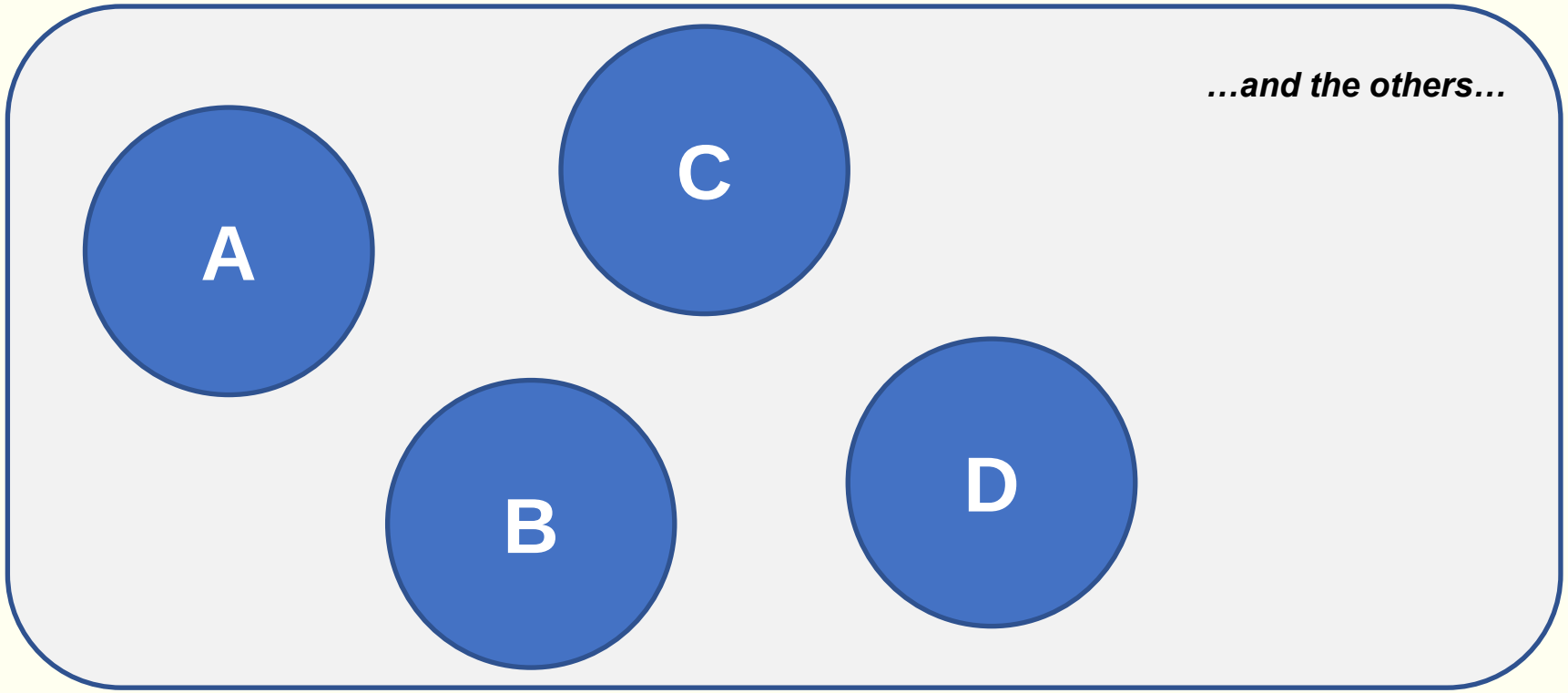
Agenda

I. Motivation
II. Open Set Deep Networks
    I. OpenMax

지식의 분류 - 희망편



지식의 분류 - 절망편

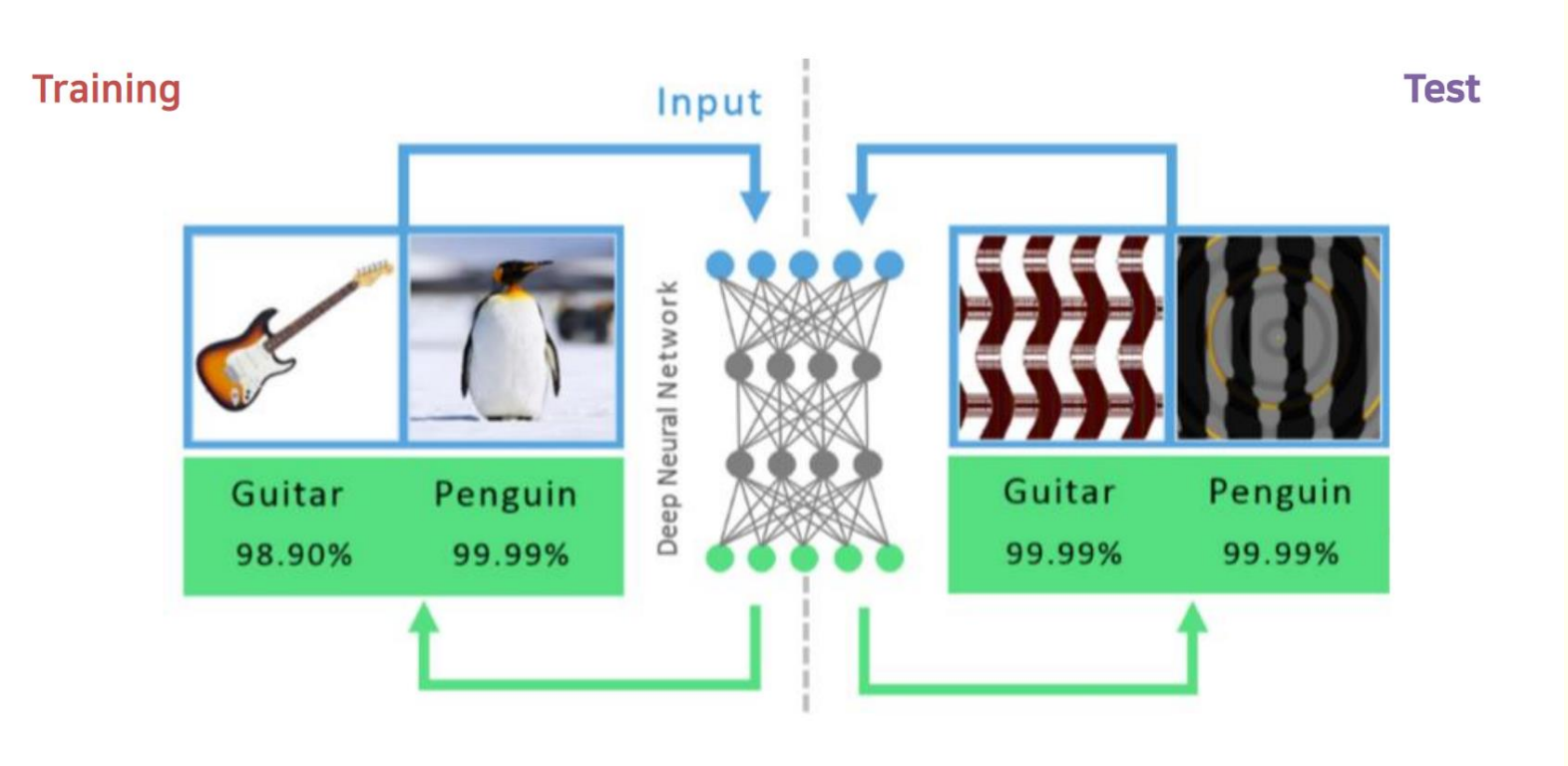


Motivation

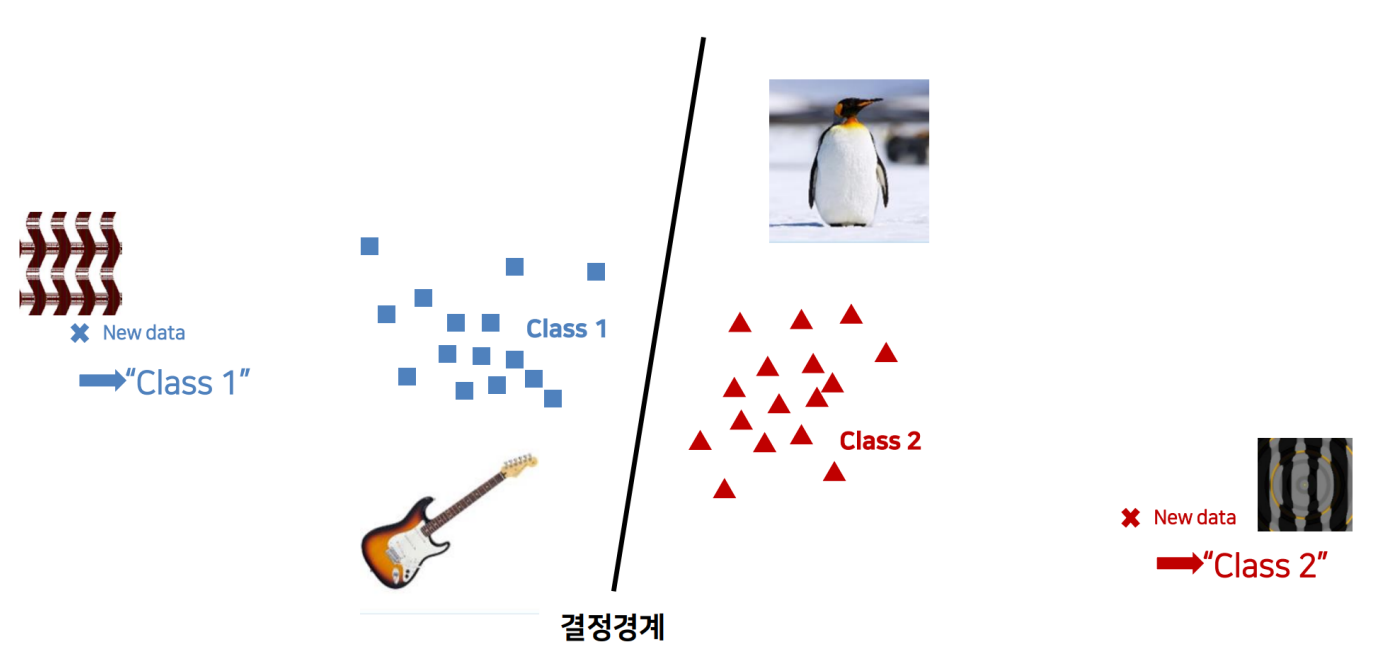
- The ***closed set*** nature of deep networks ***forces*** them to ***choose from one of the known classes.***

- Recognition in the real world is open set, i.e. the recognition system **should** ***reject unknown/unseen classes*** **at test time.**

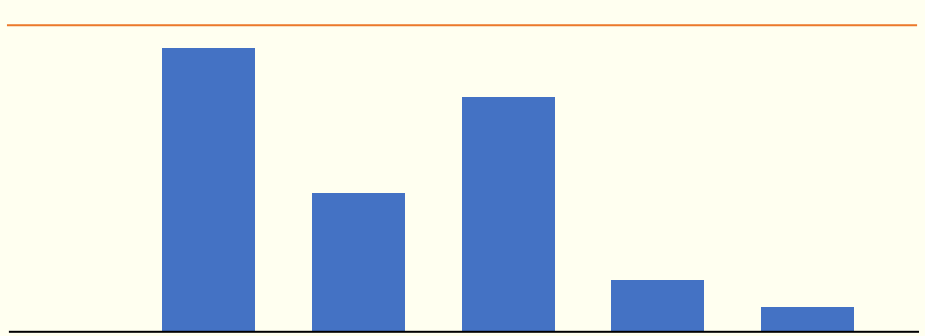
Deep Networks are Easily Fooled



Deep Networks are Easily Fooled



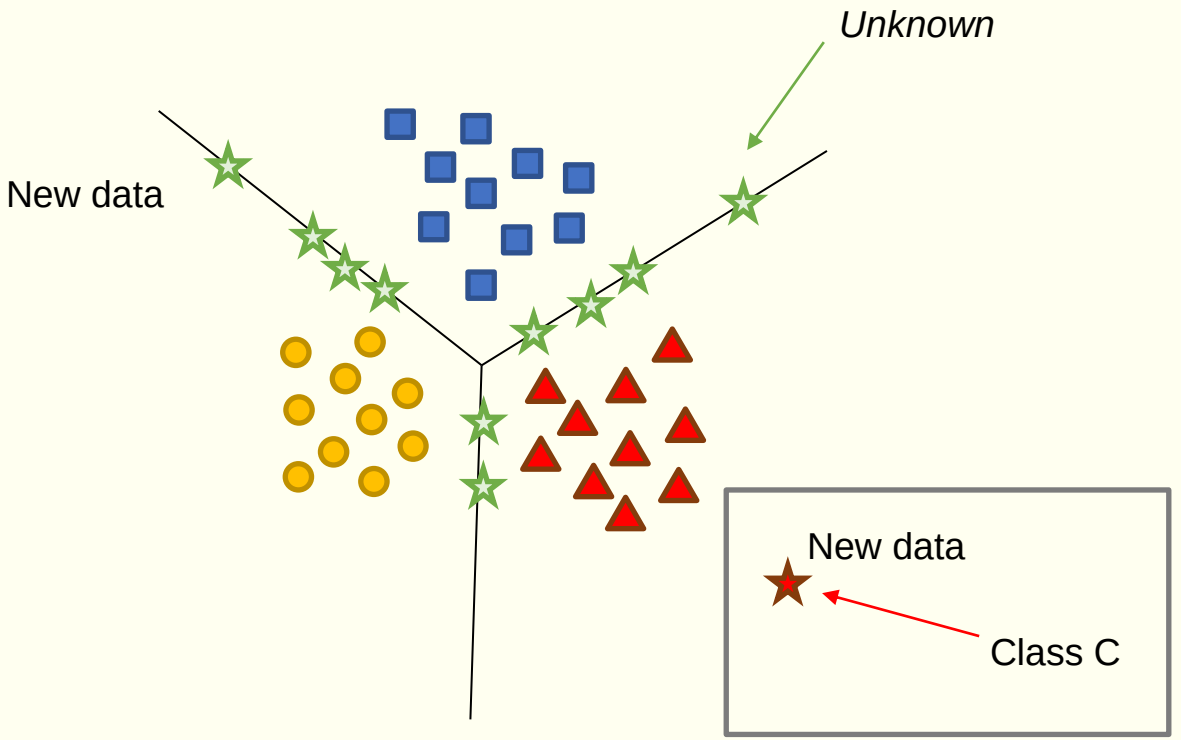
Naïve Approach : Threshold



Reject!

thresholding on uncertainty is not sufficient to determine what is unknown.

Naïve Approach : Threshold

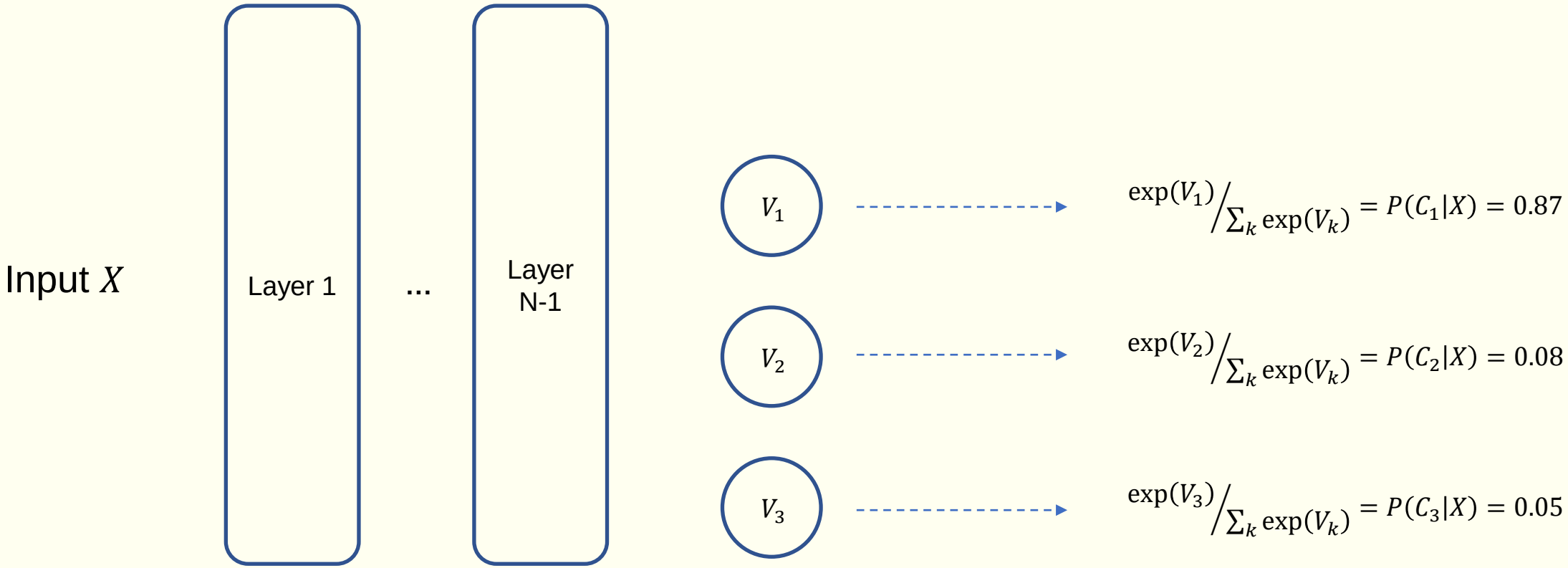


thresholding on uncertainty is not sufficient to determine what is unknown.

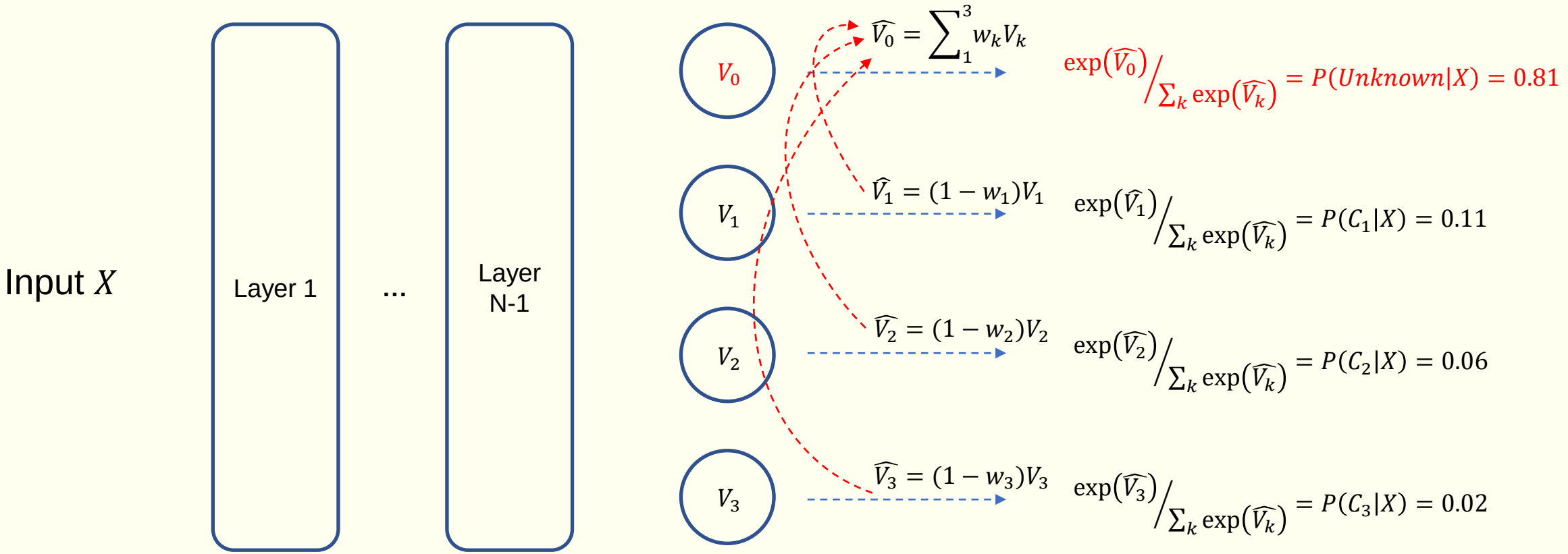
Open Set Deep Networks : OpenMax

- Extends ***SoftMax layer*** by ***enabling it to predict an unknown class***
- OpenMax incorporates ***likelihood*** of ***the recognition system failure***. This likelihood is used to ***estimate the probability for a given input belonging to an unknown class.***

Open Set Deep Networks : OpenMax



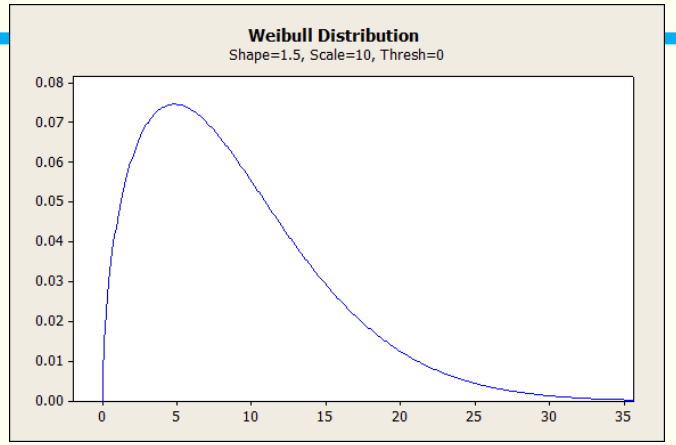
Open Set Deep Networks : OpenMax



OpenMax

- Defining w_k
  - Probability of NOT in that class k
- MAV : Mean Activation Vector
  - Activation Vector (AV) = values from logit layer
  - 기존 : logit layer → softmax 후 **확률**로 해석
  - 논문 : logit layer → 어떤 클래스와 연관되어 있는지에 대한 **분포**를 제공한다고 해석

OpenMax

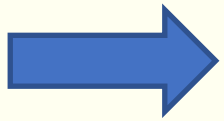


- Estimating Outlier Distribution
  1. Collect distances between ***all correct positive training instances*** and ***MAV***
  2. Per-class Weibull fit to n-largest distances to MAV
     - 동일 분포에서 **독립적으로 추출한 샘플 중 가장 큰 값**을 뽑으면, *가장 큰 값보다 큰 확률은 Weibull 분포의 형태로 만들 수 있음(Extreme Value Theorem)*
  3. $w_k = weibull_k.cdf(d_{k,x}),$
     $where\ d_{k,x} = (\text{distance between } MAV_k \text{ and input X})$

Example – Animal Classifier

- Animal Classifier (Dog / Cat / Deer)

Distance (Dog)	Distance (Cat)	Distance (Deer)
5.5533	5.6833	5.1219
5.0953	5.5163	4.7848
4.9558	5.4040	4.5958
…	…	…
0.6255	0.5465	0.5633



Estimate top-n
Weibull Distribution per Class

Open Set Deep Networks : OpenMax

Layer 1

...

Layer N-1

5.20

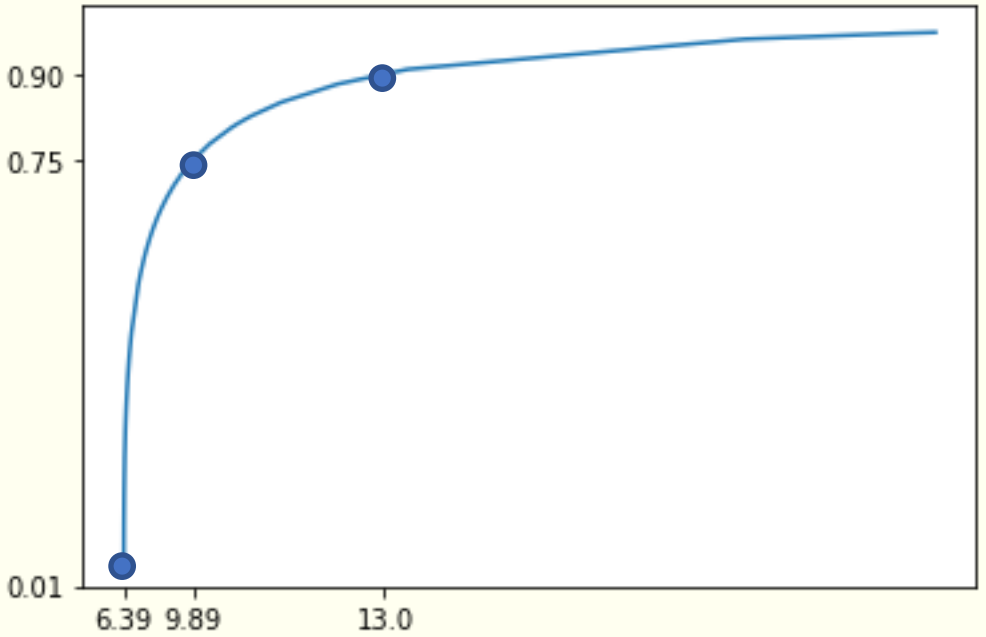
-1.71

0.12

Calculate distance with MAVs

Class	MAVs	Distance
MAV_{dog}	[5.12, -1.12, 0.12]	**6.39**
MAV_{cat}	[0.12, 11.27, -3.53]	**13.00**
MAV_{deer}	[1.32, -2.35, 9.32]	**9.89**

Open Set Deep Networks : OpenMax



Class	MAVs	Distance	CDF
MAV_{dog}	[5.12, -1.12, 0.12]	6.39	0.01
MAV_{cat}	[0.12, 11.27, -3.53]	**9.89**	**0.75**
MAV_{deer}	[1.32, -2.35, 9.32]	**13.00**	**0.90**

Open Set Deep Networks : OpenMax

Layer 1 ... Layer N-1

V_0 V_1 V_2 V_3

$\widehat{V_0} = \sum_1^3 w_k V_k = -0.000968$

$\widehat{V_1} = (1 - 0.01) \times 5.20 = 5.14$

$\widehat{V_2} = (1 - 0.75) \times (-1.71) = -0.42$

$\widehat{V_3} = (1 - 0.90) \times 0.12 = 0.01$

$$\exp(\widehat{V_0}) \Big/ \sum_k \exp(\widehat{V_k}) = 0.02$$

$$\exp(\widehat{V_1}) \Big/ \sum_k \exp(\widehat{V_k}) = 0.93$$

$$\exp(\widehat{V_2}) \Big/ \sum_k \exp(\widehat{V_k}) = 0.01$$

$$\exp(\widehat{V_3}) \Big/ \sum_k \exp(\widehat{V_k}) = 0.04$$

Open Set Deep Networks : OpenMax

Layer 1

...

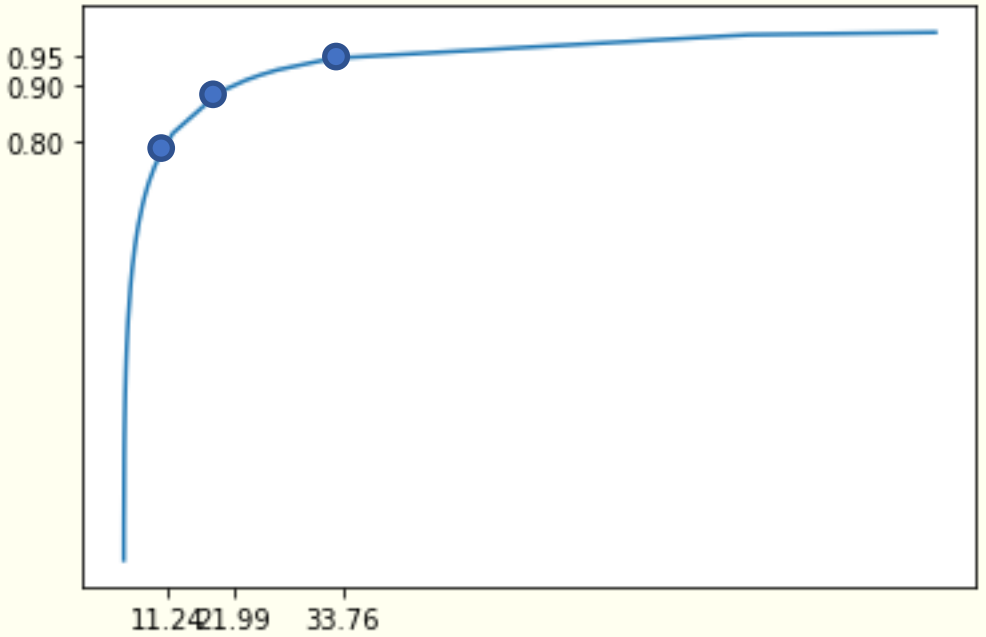
Layer N-1

8.32
21.1
-9.52

Calculate distance with MAVs

Class	MAVs	Distance
MAV_{dog}	[5.12, -1.12, 0.12]	**11.24**
MAV_{cat}	[0.12, 11.27, -3.53]	**33.76**
MAV_{deer}	[1.32, -2.35, 9.32]	**21.99**

Open Set Deep Networks : OpenMax



Class	MAVs	Distance	CDF
MAV_{dog}	[5.12, -1.12, 0.12]	**11.24**	**0.9**
MAV_{cat}	[0.12, 11.27, -3.53]	**33.76**	**0.8**
MAV_{deer}	[1.32, -2.35, 9.32]	**21.99**	**0.95**

Open Set Deep Networks : OpenMax

Layer 1 ... Layer N-1

V_0 V_1 V_2 V_3

$\widehat{V_0} = \sum_1^3 w_k V_k = 15.32$

$\widehat{V_1} = (1 - 0.9) \times 8.32 = 0.83$

$\widehat{V_2} = (1 - 0.8) \times 21.1 = 4.22$

$\widehat{V_3} = (1 - 0.95) \times (-9.52) = -0.47$

$\exp(\widehat{V_0}) / \sum_k \exp(\widehat{V_k}) = 0.70$

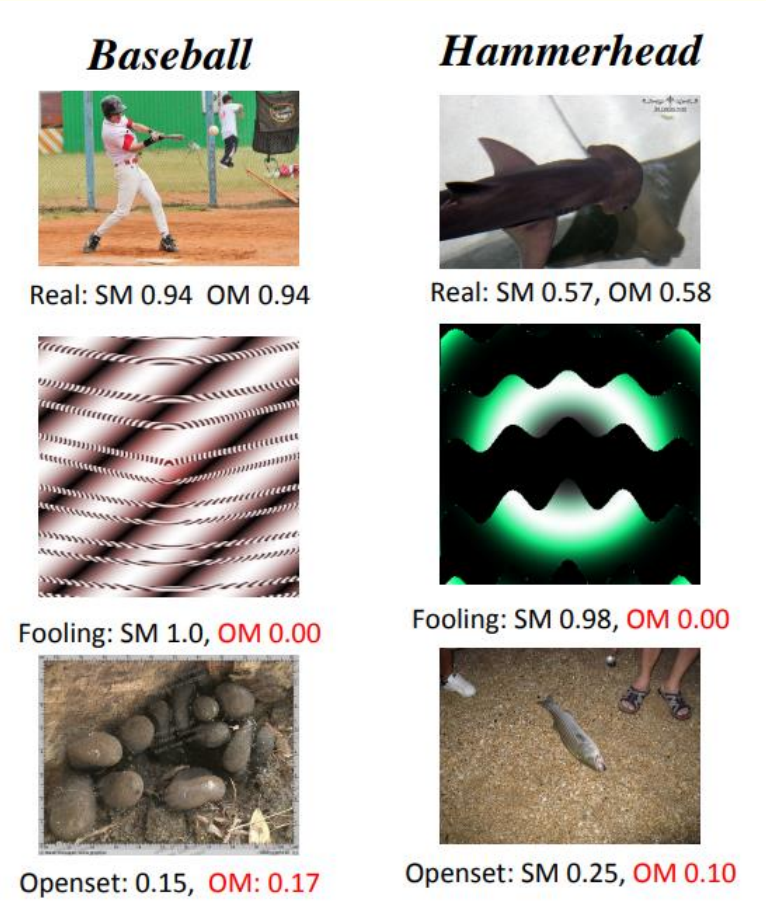
$\exp(\widehat{V_1}) / \sum_k \exp(\widehat{V_k}) = 0.03$

$\exp(\widehat{V_2}) / \sum_k \exp(\widehat{V_k}) = 0.26$

$\exp(\widehat{V_3}) / \sum_k \exp(\widehat{V_k}) = 0.01$

OpenMax : Summary

- Inner-Domain Classes
  - Simmilar to Softmax results
- Outer-Domain Classes
  - Softmax classifier ***always predicts to one of classes*** with high-confidence,
  - While OpenMax ***can reject for outer-domain classes*** with great performance



OpenMax : Summary

- After training neural network(classifier),
  do post-process with open set recognition
- Estimate extreme distribution w.r.t. distance between MAV and samples
- Define logits for unknown class(outer-domain class), and update existing logits