2020.02.21. HCAI Open Seminar

Open Set Recognition In Deep Networks

김상훈

CONTENTS

◆ **이해를 돕기 위한 배경지식 설명**

(1) 기계학습의 분류

(2) 심층신경망에서의 분류모델

◆ **심층신경망 분류모델의 한계**

◆ **Open Set Recognition의 개념**

◆ **OpenMax** : 실제 알고리즘을 통해 심층신경망에서의 거리기반 Open Set Recognition

배경지식

(1) 기계학습의 분류

훈련 데이터의 형태에 따라

비지도학습

	X_1	X_2	…	X_p
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_n	…	…	…	…

지도학습

	X_1	X_2	…	X_p
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_n	…	…	…	…

Function

$Y = F(X)$

Y

지도학습 내에서도 **Y변수의 형태에 따라**

회귀모델

예시)

- 기상 데이터로 주가를 예측하는 문제

분류모델

예시)

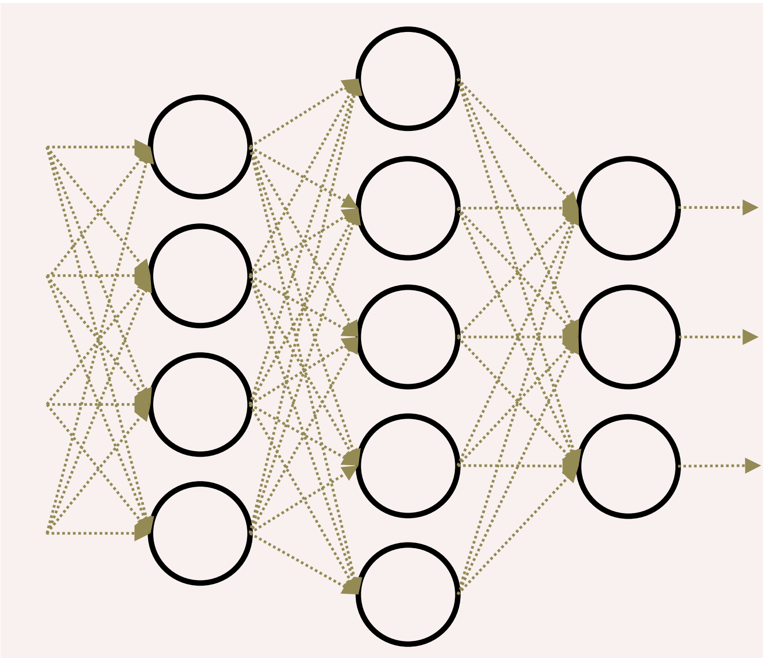
- 이미지 데이터로 고양이, 강아지를 분류하는 문제

배경지식

(2) 심층신경망에서의 분류모델

Input X

…



Neural Net

Output Y

Y
강아지
토끼
강아지
…
…
토끼
고양이
고양이

배경지식

(2) 심층신경망에서의 분류모델

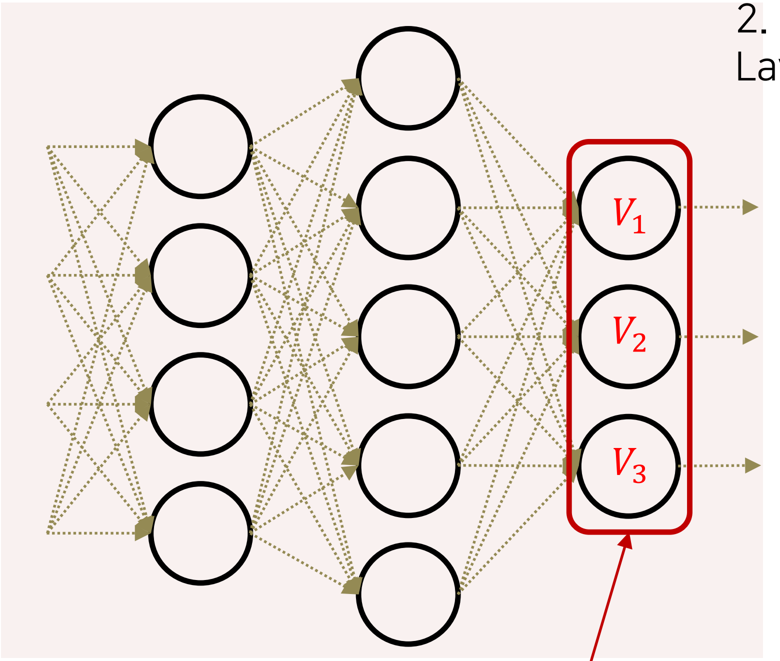
1. 신경망의 연산을 통해 Logit Vector로 축약

2. 각 클래스에 대응되는 Logit값을 SoftMax Layer에서 0~1 사이의 확률로 반환

	X_1	X_2	…	X_P
N_1	…	…	…	…

Pixel by pixel variable

…



Neural Net

$$\frac{\exp(V_1)}{\sum_k \exp(V_k)} \quad P(Y = \text{강아지})$$

$$\frac{\exp(V_2)}{\sum_k \exp(V_k)} \quad P(Y = \text{고양이})$$

$$\frac{\exp(V_3)}{\sum_k \exp(V_k)} \quad P(Y = \text{토끼})$$

SoftMax

Logit Vector = $\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$

Logits
값의 범위 : $(-\infty, \infty)$

SoftMax Activation
값의 범위 : $[0,1]$

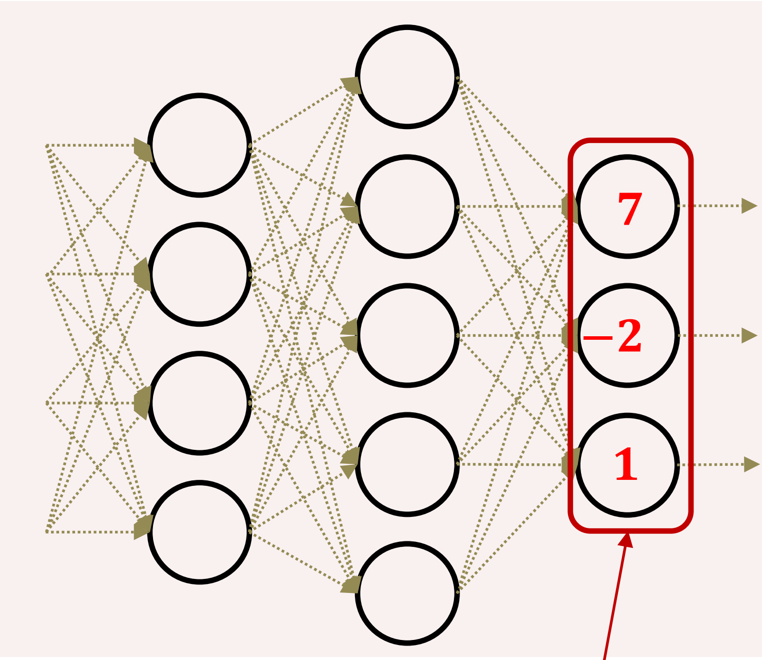
배경지식

(2) 심층신경망에서의 분류모델

	X_1	X_2	…	X_P
N_1	…	…	…	…

Pixel by pixel variable

…



Neural Net

Logit 값이 클수록, 더 큰 확률을 가짐
(Logit 값은 각 클래스의 확률에 대응하는 값)

$$\frac{\exp(V_1)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{강아지}) = 0.81$$

$$\frac{\exp(V_2)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{고양이}) = 0.12$$

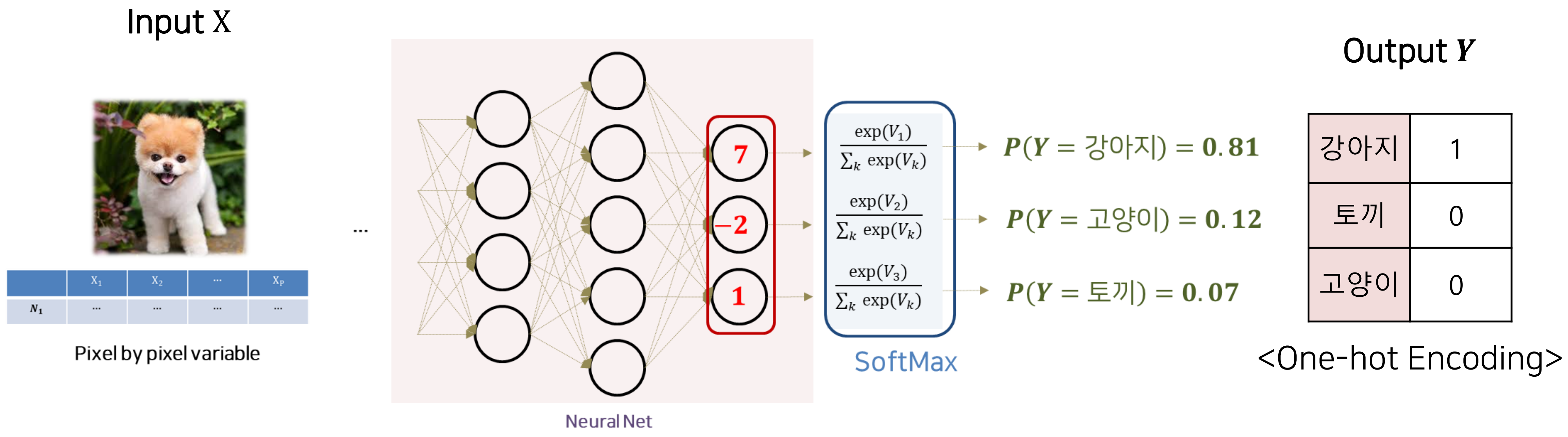
$$\frac{\exp(V_3)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{토끼}) = 0.07$$

SoftMax

Logit Vector = $\begin{bmatrix} 7 \\ -2 \\ 1 \end{bmatrix}$

배경지식

(2) 심층신경망에서의 분류모델



Output 값과 신경망이 출력하는 확률 중 가장 큰 확률을 가지는 클래스가 일치하도록 학습을 진행

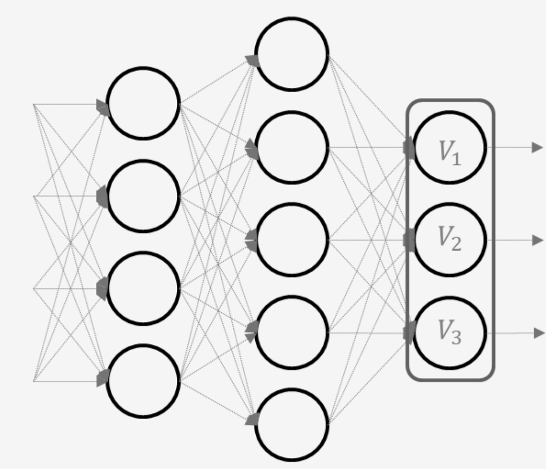
배경지식

(2) 심층신경망에서의 분류모델

Input X

	X_1	X_2	…	X_p
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_n	…	…	…	…

…



Neural Net

$$\frac{\exp(V_1)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

$$\frac{\exp(V_2)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

$$\frac{\exp(V_3)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

SoftMax

Output $\boldsymbol{Y}$

obs	강아지	고양이	토끼
N_1	1	0	0
N_2	0	1	0
…	…	…	…
N_n	0	0	1

모든 학습 데이터로
학습진행

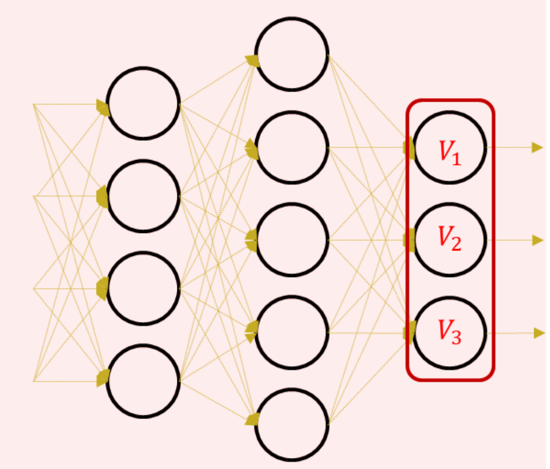
배경지식

(2) 심층신경망에서의 분류모델

Input X

	X1	X2	…	Xp
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_n	…	…	…	…

…



Neural Net

$$\frac{\exp(V_1)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

$$\frac{\exp(V_2)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

$$\frac{\exp(V_3)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

SoftMax

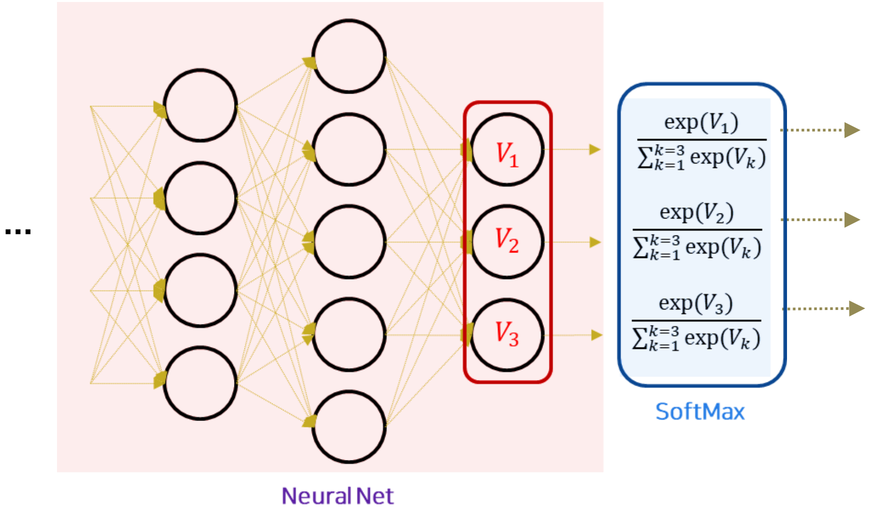
Output $\boldsymbol{Y}$

obs	강아지	고양이	토끼
N_1	1	0	0
N_2	0	1	0
…	…	…	…
N_n	0	0	1

모든 학습 데이터로
학습완료

배경지식

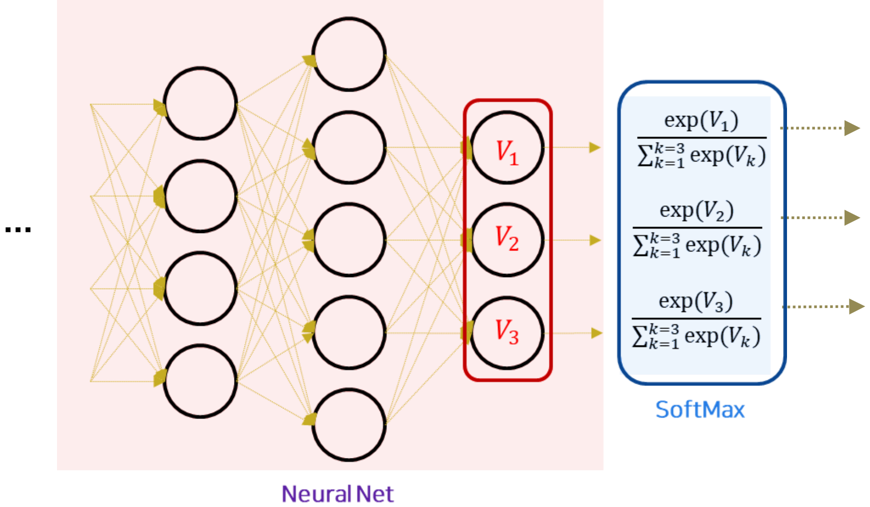
(2) 심층신경망에서의 분류모델



분류기(Classifier) 생성

배경지식

(2) 심층신경망에서의 분류모델



$\boldsymbol{P(Y = \text{강아지}) = 0.92}$

$\boldsymbol{P(Y = \text{고양이}) = 0.01}$

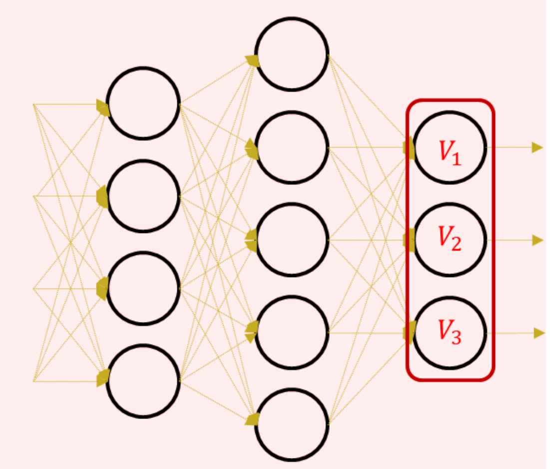
$\boldsymbol{P(Y = \text{토끼}) = 0.07}$

강아지/고양이/토끼 분류기

배경지식

(2) 심층신경망에서의 분류모델

...



Neural Net

$$\frac{\exp(V_1)}{\sum_{k=1}^{k=3}\exp(V_k)} \qquad \frac{\exp(V_2)}{\sum_{k=1}^{k=3}\exp(V_k)} \qquad \frac{\exp(V_3)}{\sum_{k=1}^{k=3}\exp(V_k)}$$

SoftMax

$$\boldsymbol{P(Y = \text{강아지}) = 0.01}$$

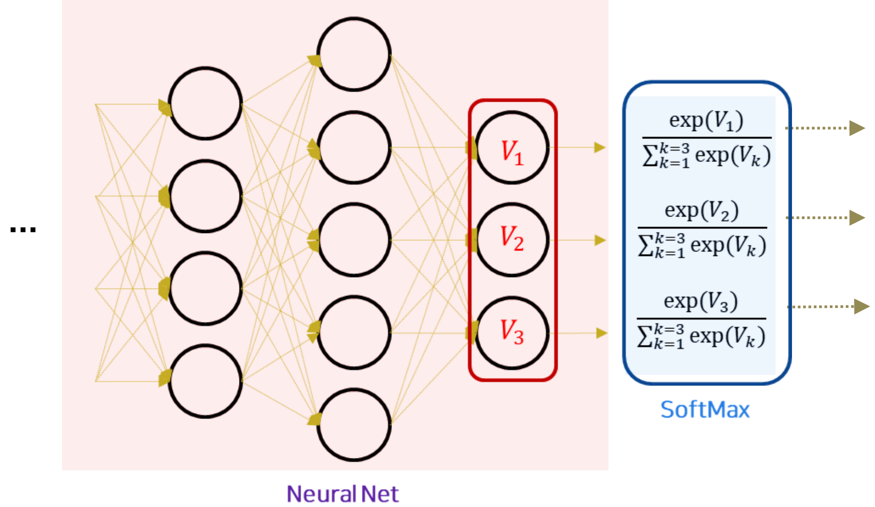
$$\boldsymbol{P(Y = \text{고양이}) = 0.96}$$

$$\boldsymbol{P(Y = \text{토끼}) = 0.03}$$

강아지/고양이/토끼 분류기

배경지식

(2) 심층신경망에서의 분류모델



$\boldsymbol{P(Y = \text{강아지}) = 0.001}$

$\boldsymbol{P(Y = \text{고양이}) = 0.009}$

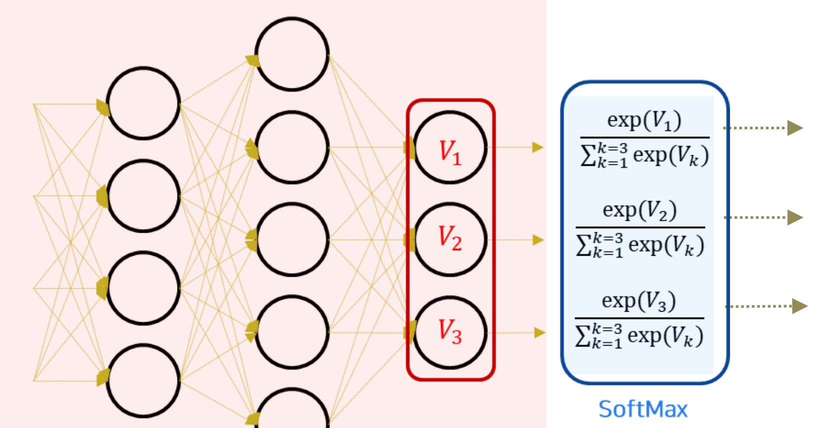
$\boldsymbol{P(Y = \text{토끼}) = 0.99}$

강아지/고양이/토끼 분류기

심층신경망 분류모델의 한계

학습 단계에서 학습하지 않은 클래스의 데이터가 들어온다면?

...

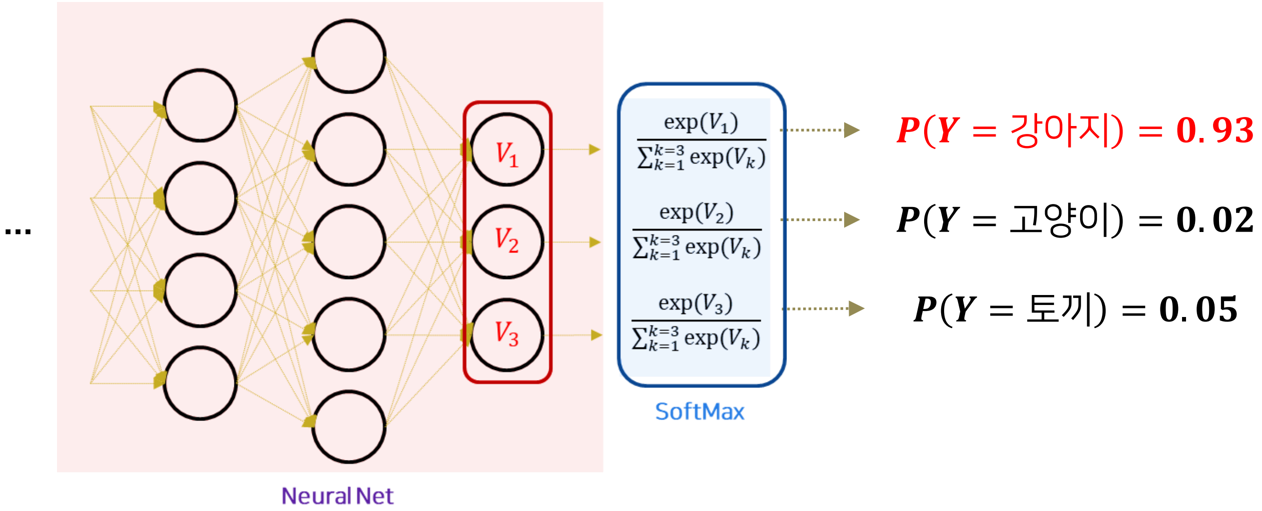


Neural Net

강아지/고양이/토끼 분류기

심층신경망 분류모델의 한계

학습 단계에서 학습하지 않은 클래스의 데이터가 들어온다면?



강아지/고양이/토끼 분류기

분류기는 학습한 클래스에 대한 확률만 출력할 수 있다.

심층신경망 분류모델의 한계

❖ **Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images**

- 2015년 CVPR(Computer Vision and Pattern Recognition)에서 발표된 논문
- 2020년 2월 20일 기준 **1,553**회 인용

Deep Neural Networks are Easily Fooled:
High Confidence Predictions for Unrecognizable Images

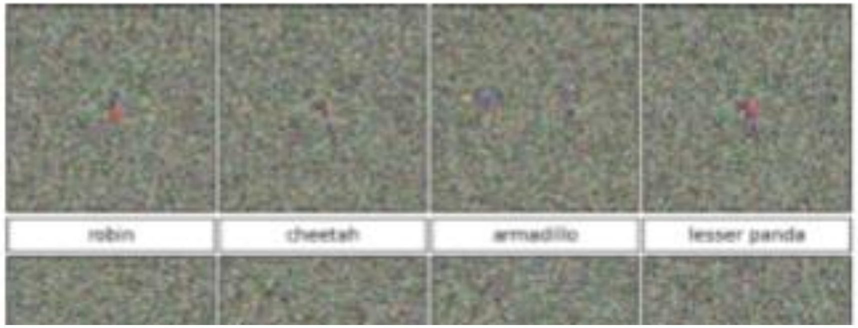
Anh Nguyen
University of Wyoming
anguyen8@uwyo.edu

Jason Yosinski
Cornell University
yosinski@cs.cornell.edu

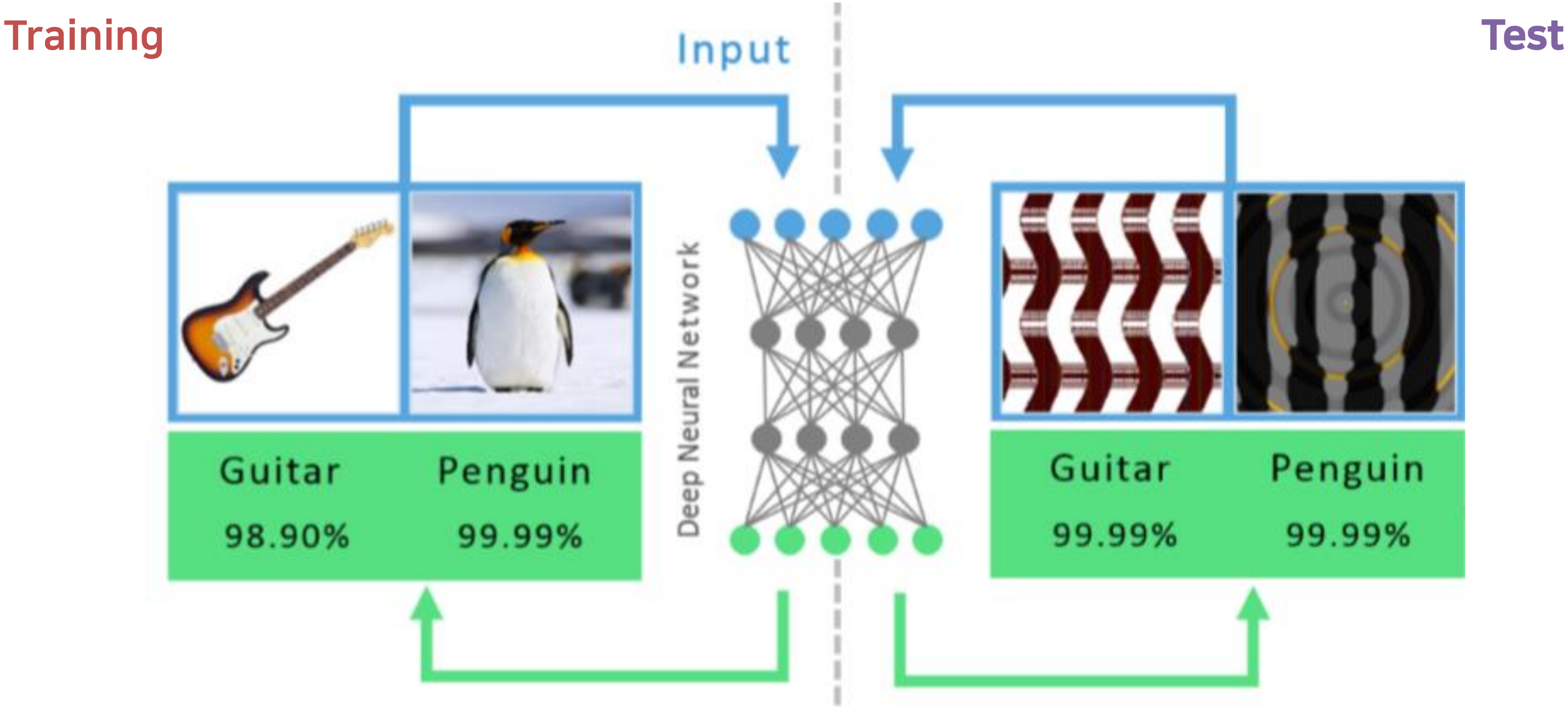
Jeff Clune
University of Wyoming
jeffclune@uwyo.edu

Abstract

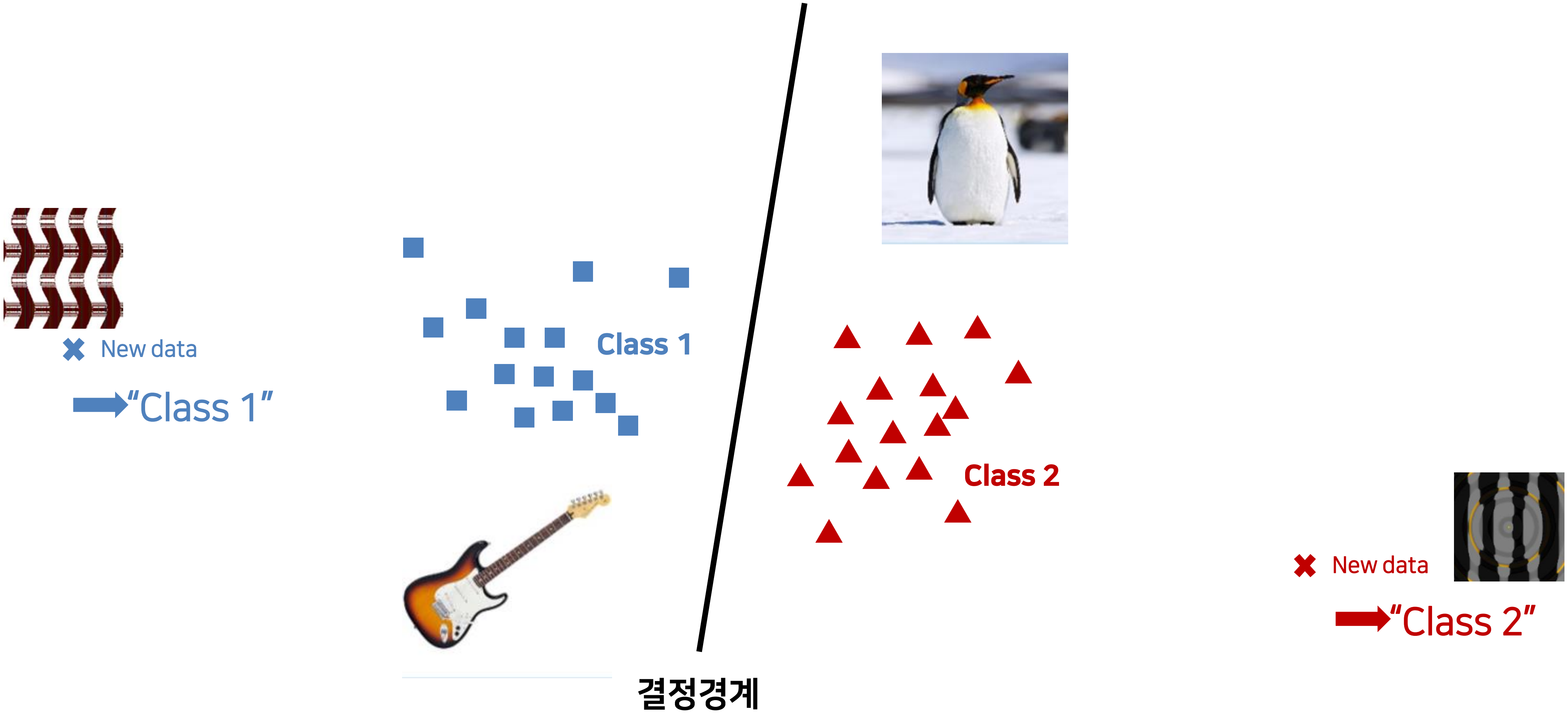
Deep neural networks (DNNs) have recently been achieving state-of-the-art performance on a variety of pattern-recognition tasks, most notably visual classification problems. Given that DNNs are now able to classify objects in images with near-human-level performance, questions



심층신경망 분류모델의 한계

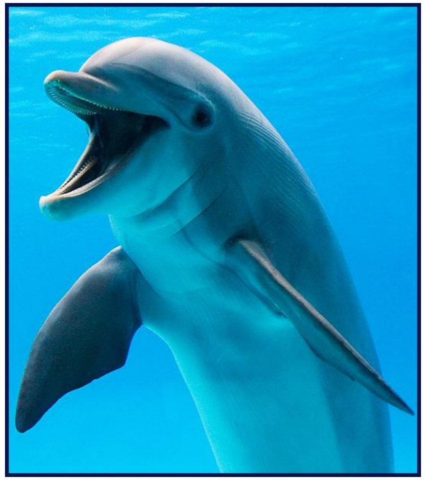


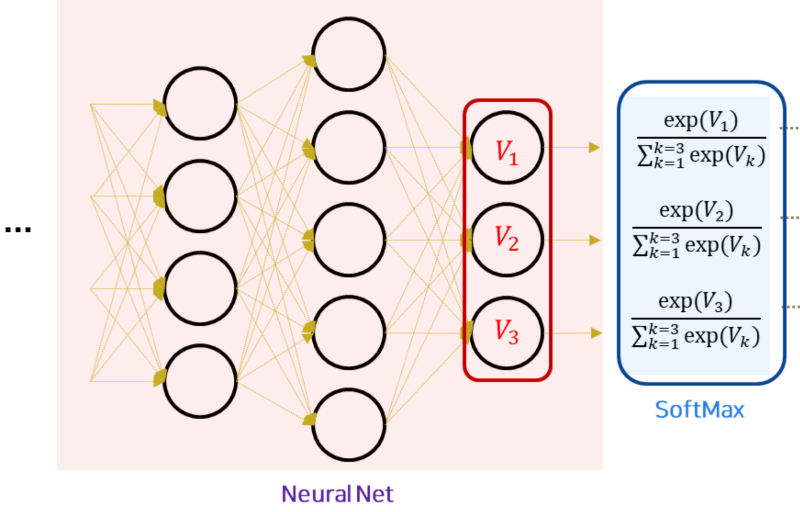
심층신경망 분류모델의 한계



심층신경망 분류모델의 한계

만약 SoftMax가 출력한 확률이 모두 낮다면?





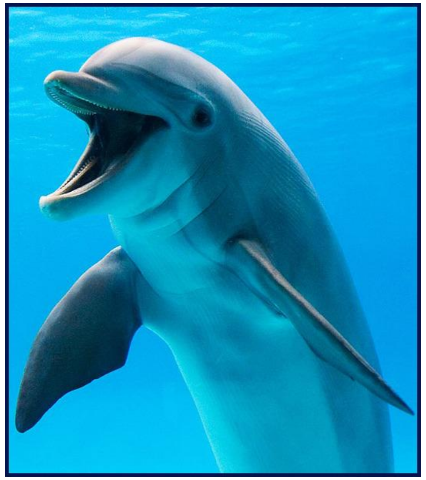
$\boldsymbol{P(Y=}$ 강아지$\boldsymbol{)=0.40}$

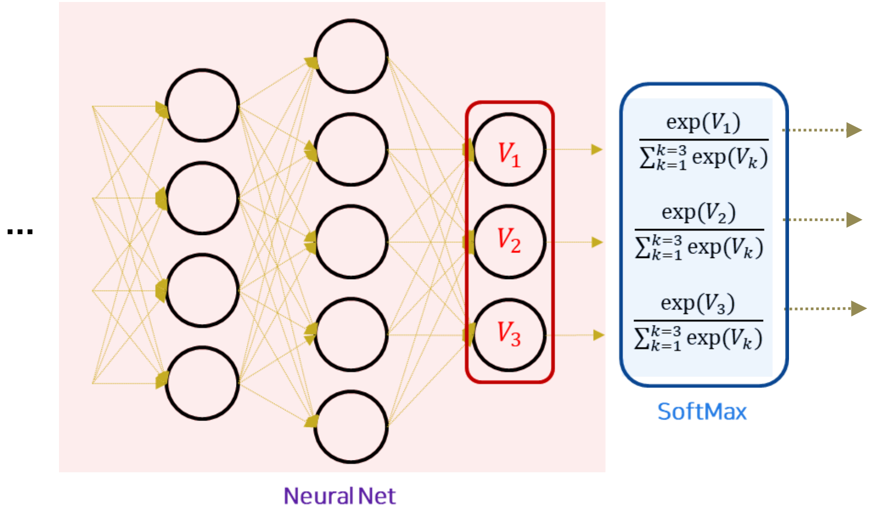
$\boldsymbol{P(Y=}$ 고양이$\boldsymbol{)=0.32}$

$\boldsymbol{P(Y=}$ 토끼$\boldsymbol{)=0.28}$

심층신경망 분류모델의 한계

만약 SoftMax가 출력한 확률이 모두 낮다면?





Threshold = 0.7

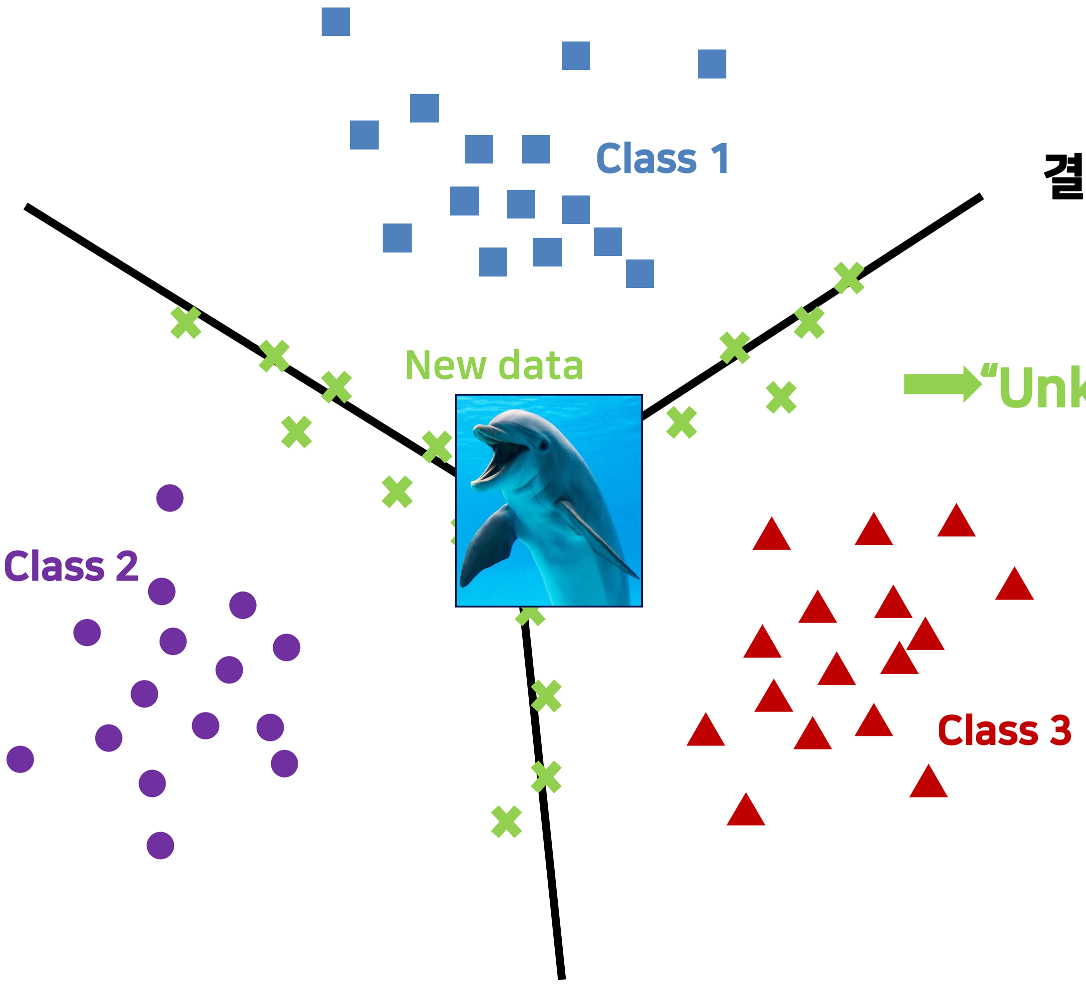
$\boldsymbol{P(Y=\text{강아지})=0.40}$ < 0.7 reject

$\boldsymbol{P(Y=\text{고양이})=0.32}$ < 0.7 reject

$\boldsymbol{P(Y=\text{토끼})=0.28}$ < 0.7 reject

➡ Unknown Class

심층신경망 분류모델의 한계



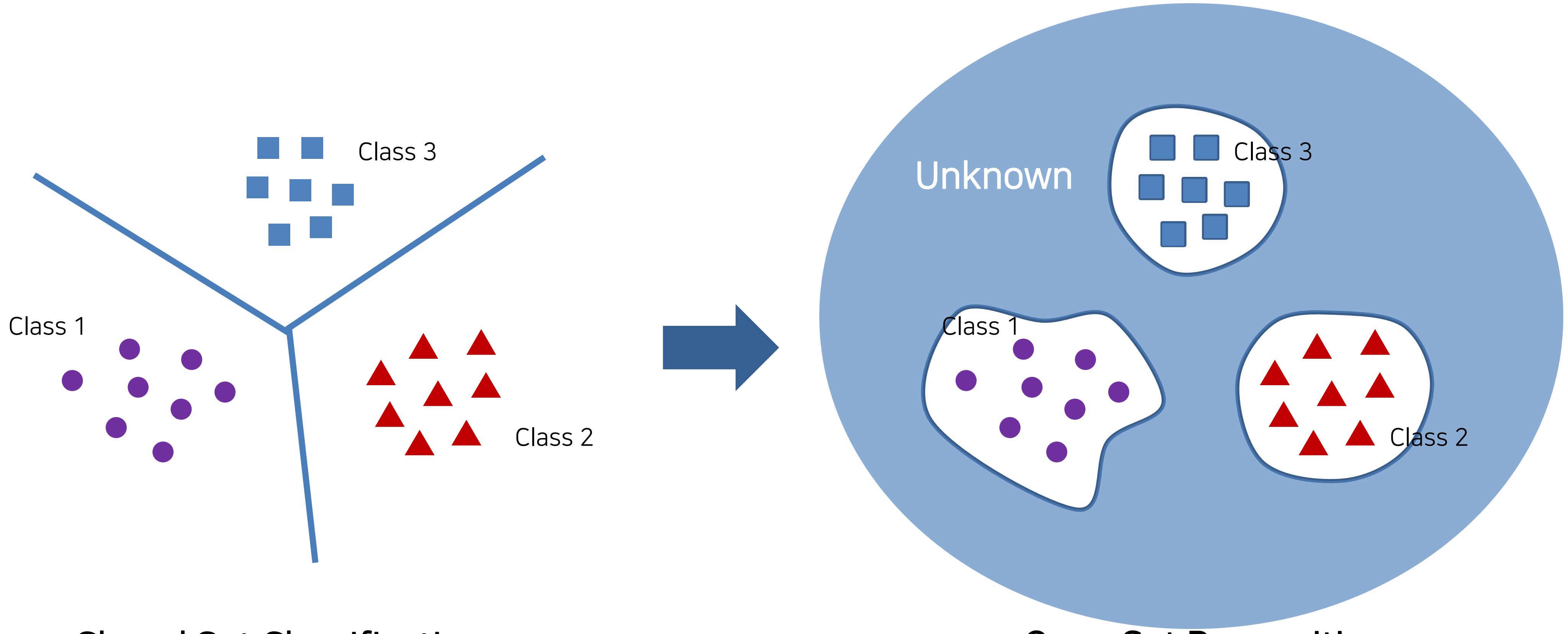
결정 경계 $y^* = \mathrm{argmax}_j \, P(y = j|\mathbf{x})$

➡"Unknown"

✖ New data

➡"Class 3"

Open Set Recognition의 개념



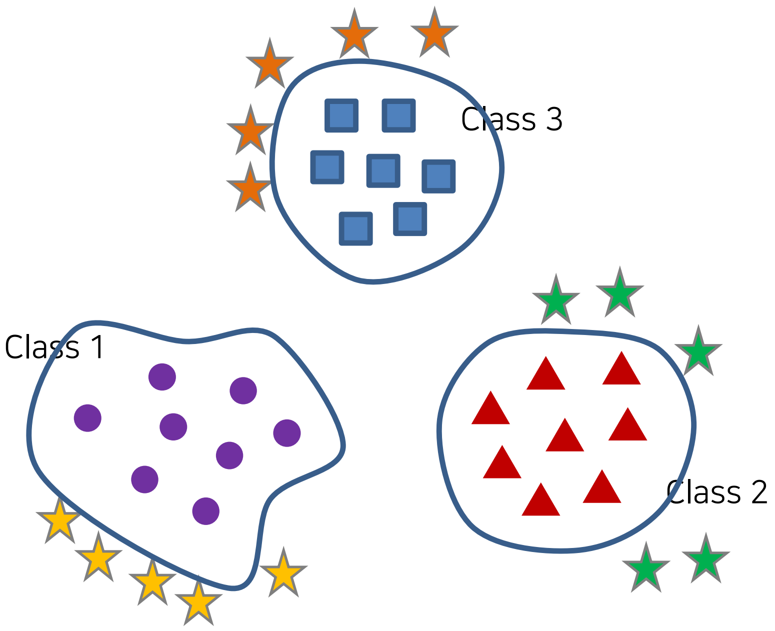
Closed Set Classification

Open Set Recognition

Open Set Recognition의 개념

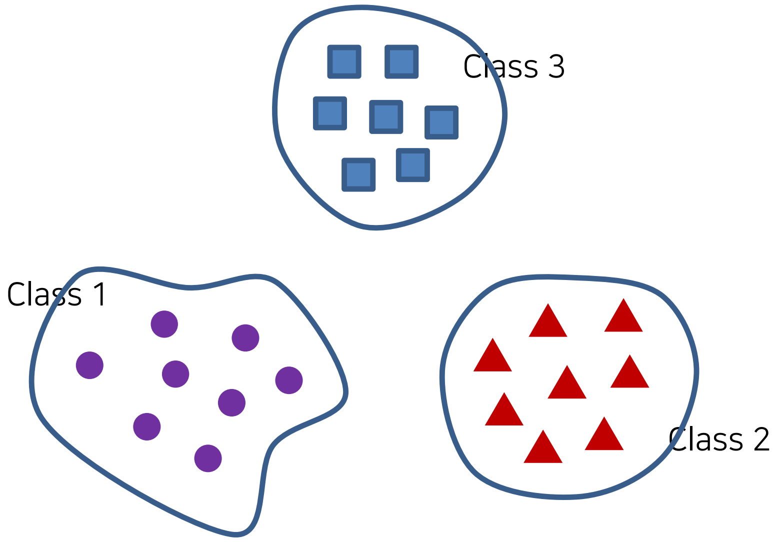
❖ Open Set Recognition의 분류

Adversarial Learning-based



GAN 등의 생성모델을 통해
각 클래스와 비슷한 다른 이미지들을
생성하여 새로운 클래스로 추가하여 학습

Distance-based



평균으로부터 떨어진 거리,
다른 클래스와의 마진 등을 활용하여
결정경계 생성

OpenMax

❖ Towards open set deep networks

- 2016년 CVPR(Computer Vision and Pattern Recognition)에서 발표된 논문
- 저자 Bendale A, **Boult T**
  - Toward Open Set Recognition, **Scheirer W J**, de Rezende Rocha A, Sapkota A, et al. (PAMI, 2013)에서 **Open Set Recognition을 처음 정의**
- Open Set Recognition을 **심층신경망에 처음 적용**한 연구
- 2020년 2월 20일 기준 **218**회 인용

Towards Open Set Deep Networks

Abhijit Bendale*, Terrance E. Boult
University of Colorado at Colorado Springs
{abendale,tboult}@vast.uccs.edu *

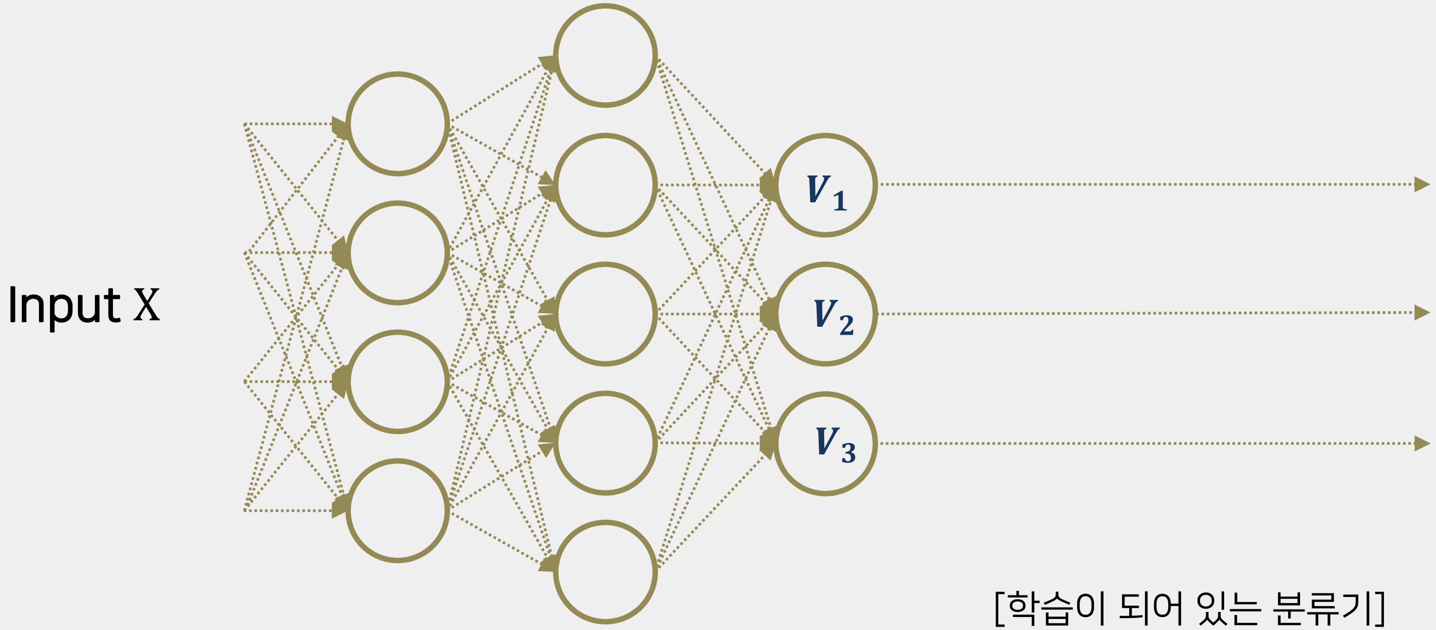
Abstract

Deep networks have produced significant gains for various visual recognition problems, leading to high impact academic and commercial applications. Recent work in deep networks highlighted that it is easy to generate images that humans would never classify as a particular object class, yet networks classify such images high confidence as that given class – deep network are easily fooled with images humans do not consider meaningful. The closed set nature of deep networks forces them to choose from one of the known classes leading to such artifacts. Recognition in

vision and learning. Recent research in deep networks has significantly improved many aspects of visual recognition [26, 3, 11]. Co-evolution of rich representations, scalable classification methods and large datasets have resulted in many commercial applications [5, 28, 16, 6]. However, a wide range of operational challenges occur while deploying recognition systems in the dynamic and ever-changing real world. A vast majority of recognition systems are designed for a static closed world, where the primary assumption is that all categories are known a priori. Deep networks, like many classic machine learning tools, are designed to perform closed set recognition.

OpenMax

❖ OpenMax 알고리즘



$$\frac{\exp(V_1)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{강아지}) = 0.26$$

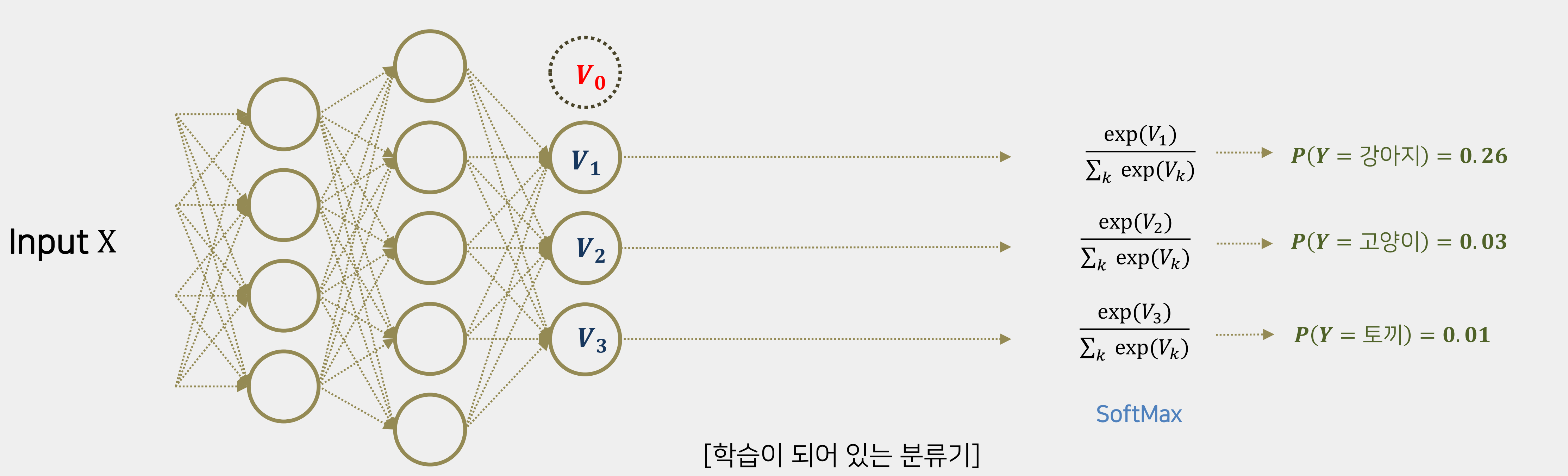
$$\frac{\exp(V_2)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{고양이}) = 0.03$$

$$\frac{\exp(V_3)}{\sum_k \exp(V_k)} \rightarrow P(Y = \text{토끼}) = 0.01$$

SoftMax

OpenMax

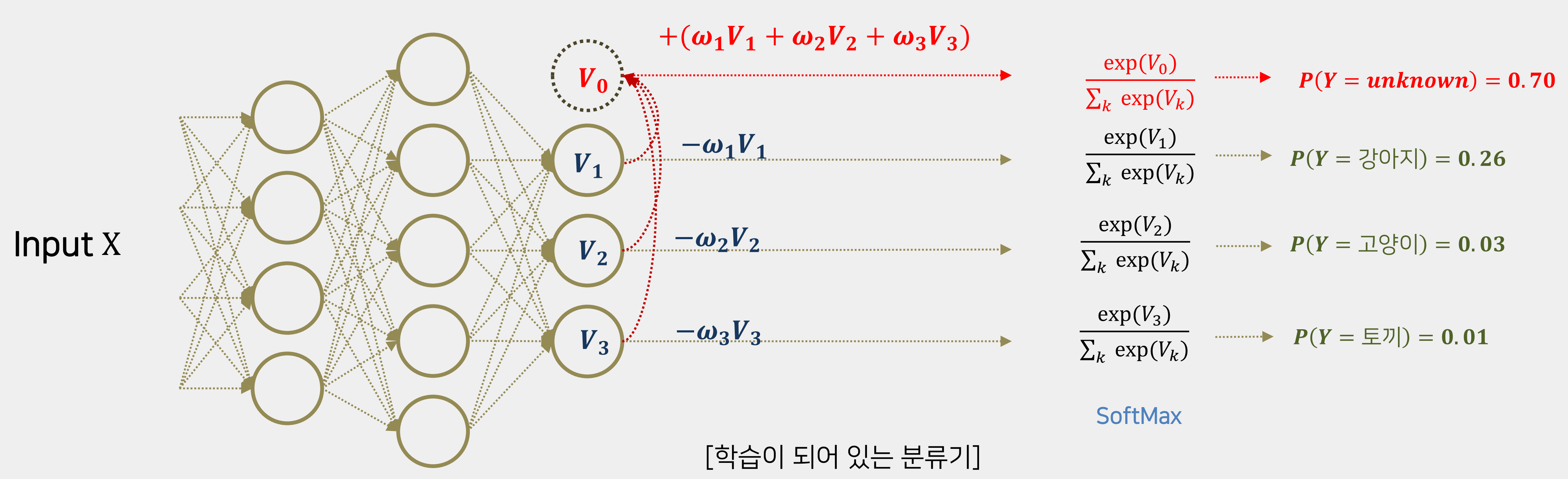
❖ OpenMax 알고리즘



- $V_0 = unknown\ class$에 대응되는 Logit 값

OpenMax

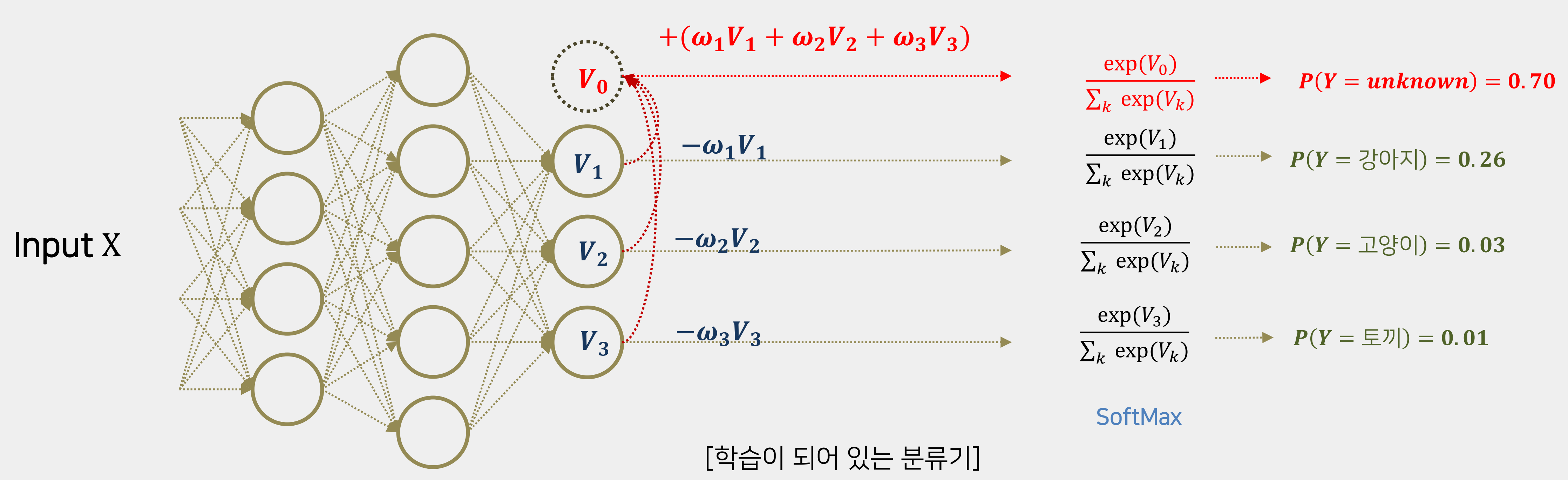
❖ OpenMax 알고리즘



- $V_0 = unknown\ class$에 대응되는 Logit 값
- $\omega_k =$ 분류기가 $k\ class$로 잘못 분류했을 확률에 대응하는 가중치

OpenMax

❖ OpenMax 알고리즘



ω_k를 어떻게 정의할 수 있을까?

➡ 본 논문에서는 **Extreme Value Theorem**에 기반하여 각 클래스 별 평균 Logit Vector로부터의 거리에 대한 **극단값(이상치)의 분포**를 통해 ω_k를 정의한다.

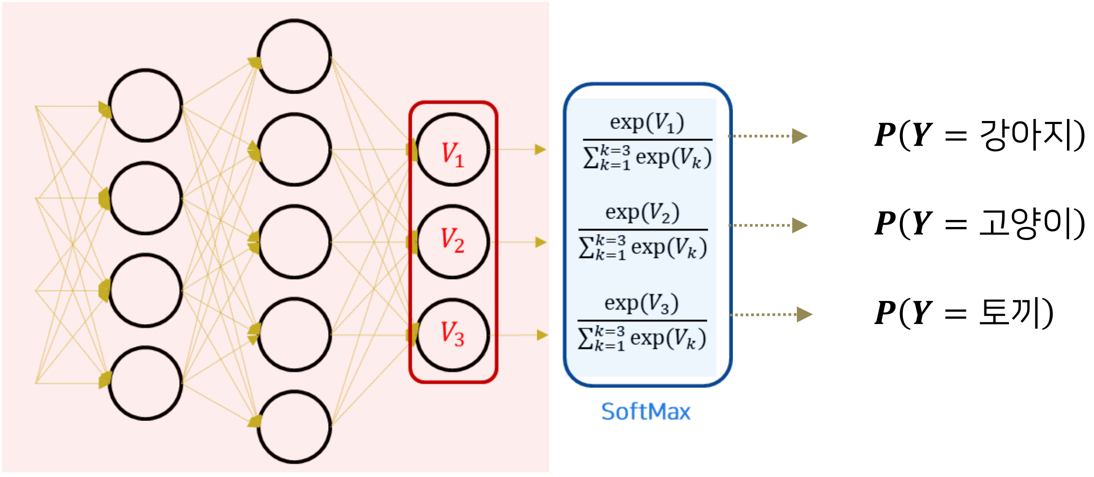
OpenMax : 준비 단계(Meta Recognition)

STEP1. **학습 데이터 중 분류기가 정확하게 분류한 데이터를 선별**

Input X in Training data

	X_1	X_2	…	X_P
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_n	…	…	…	…

…



강아지/고양이/토끼 분류기

OpenMax : 준비 단계(Meta Recognition)

STEP2. STEP1에서 선별된 데이터의 X데이터를 클래스별로 데이터를 분리

	X_1	X_2	…	X_P
N_{11}	…	…	…	…
N_{12}	…	…	…	…
…	…	…	…	…
N_{1a}	…	…	…	…

<강아지 클래스 데이터>

	X_1	X_2	…	X_P
N_{21}	…	…	…	…
N_{22}	…	…	…	…
…	…	…	…	…
N_{2b}	…	…	…	…

<고양이 클래스 데이터>

	X_1	X_2	…	X_P
N_{31}	…	…	…	…
N_{32}	…	…	…	…
…	…	…	…	…
N_{3c}	…	…	…	…

<토끼 클래스 데이터>

OpenMax : 준비 단계(Meta Recognition)

STEP3. 각 클래스 별로 선별된 데이터를 이용하여 Logit Vector 계산

	X_1	X_2	…	X_P
N_1	…	…	…	…
N_2	…	…	…	…
…	…	…	…	…
N_a	…	…	…	…

선별된 강아지 클래스 데이터

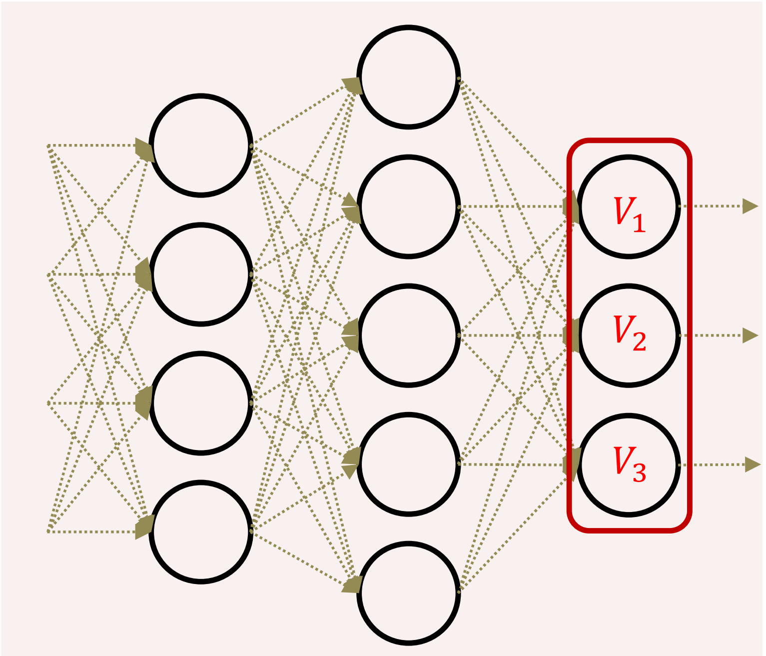
	X_1	X_2	…	X_P
N_{21}	…	…	…	…
N_{22}	…	…	…	…
…	…	…	…	…
N_{2b}	…	…	…	…

선별된 고양이 클래스 데이터

	X_1	X_2	…	X_P
N_{31}	…	…	…	…
N_{32}	…	…	…	…
…	…	…	…	…
N_{3c}	…	…	…	…

선별된 토끼 클래스 데이터

…



Neural Net

$$\frac{\exp(V_1)}{\sum_k \exp(V_k)}$$

$$\frac{\exp(V_2)}{\sum_k \exp(V_k)}$$

$$\frac{\exp(V_3)}{\sum_k \exp(V_k)}$$

SoftMax

강아지/고양이/토끼 분류기

OpenMax : 준비 단계(Meta Recognition)

STEP3. 각 클래스 별로 선별된 데이터를 이용하여 Logit Vector 계산

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

<강아지 클래스 Logit Vector Matrix>

Obs.	V_1	V_2	V_3
N_{21}	2.17	8.29	-1.32
N_{22}	1.89	9.13	2.11
…	…	…	…
N_{2b}	-3.14	5.42	1.35

<고양이 클래스 Logit Vector Matrix>

Obs.	V_1	V_2	V_3
N_{31}	-3.23	-1.23	6.98
N_{32}	1.52	1.52	9.78
…	…	…	…
N_{3c}	5.72	3.12	11.2

<토끼 클래스 Logit Vector Matrix>

OpenMax : 준비 단계(Meta Recognition)

STEP4. 각 클래스 별 Logit Vector의 평균 계산

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

$\overline{V_1} = 5.12$ $\overline{V_2} = -1.12$ $\overline{V_3} = 0.12$

<강아지 클래스 Logit Vector Matrix>

$$\mu_{rabit} = \begin{bmatrix} 1.32 \\ -2.35 \\ 9.32 \end{bmatrix}$$

<토끼 클래스 평균 Logit Vector>

$$\mu_{cat} = \begin{bmatrix} 0.12 \\ 11.27 \\ -3.53 \end{bmatrix}$$

<고양이 클래스 평균 Logit Vector>

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

OpenMax : 준비 단계(Meta Recognition)

STEP5. 각 클래스 별 평균 Logit Vector와의 거리 계산

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

$\overline{V_1} = 5.12$ $\overline{V_2} = -1.12$ $\overline{V_3} = 0.12$

<강아지 클래스 Logit Vector Matrix>

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

$$\left\| \begin{bmatrix} 9.87 \\ -2.13 \\ -6.23 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 40.3225$$

OpenMax : 준비 단계(Meta Recognition)

STEP5. **각 클래스 별 평균** Logit Vector**와의 거리 계산**

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

$\overline{V_1} = 5.12$ $\overline{V_2} = -1.12$ $\overline{V_3} = 0.12$

<강아지 클래스 Logit Vector Matrix>

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

$$\left\| \begin{bmatrix} 18.5 \\ 3.18 \\ 4.98 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 221.134$$

OpenMax : 준비 단계(Meta Recognition)

STEP5. 각 클래스 별 평균 Logit Vector와의 거리 계산

Obs.	V_1	V_2	V_3
N_{11}	9.87	-2.13	-6.23
N_{12}	18.5	3.18	4.98
…	…	…	…
N_{1a}	4.89	-3.91	1.01

$\overline{V_1} = 5.12$ $\quad \overline{V_2} = -1.12$ $\quad \overline{V_3} = 0.12$

<강아지 클래스 Logit Vector Matrix>

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

$$\left\| \begin{bmatrix} 4.89 \\ -3.91 \\ 1.01 \end{bmatrix} - \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \right\|_2 = 8.6291$$

OpenMax : 준비 단계(Meta Recognition)

STEP5. 각 클래스 별 평균 Logit Vector와의 거리 계산

Obs.	평균 Logit Vector와의 거리
N_{11}	221.134
N_{12}	217.532
…	…
N_{1a}	0.014

Obs.	평균 Logit Vector와의 거리
N_{21}	195.312
N_{22}	194.904
…	…
N_{2b}	0.082

Obs.	평균 Logit Vector와의 거리
N_{31}	139.641
N_{32}	121.896
…	…
N_{3c}	0.167

평균 Logit Vector와의 거리 내림차순 정렬

OpenMax : 준비 단계(Meta Recognition)

STEP6. **평균** Logit Vector**와의 거리 중 가장 큰** η(=20)**개** Sample **각 클래스 별로 추출**

Obs.	평균 Logit Vector와의 거리
N_{11}	221.134
N_{12}	217.532
…	…
N_{1a}	0.014

평균 Logit Vector와의 거리 중 가장 큰 η(=20)개 Sample 추출

$$\underbrace{[221.134, 217.532, \cdots, 197.423]}_{\eta(=20)\text{개}}$$

Extreme Value Theorem(the Fisher–Tippet Theorem)

Let $X_1, X_2 \ldots, X_n \ldots$ be a sequence of independent and identically-distributed random variables, and $M_n = \max\{X_1, \ldots, X_n\}$. If a sequence of pairs of real numbers (a_n, b_n) exists such that each $a_n > 0$ and $\lim_{n\to\infty} P\left(\frac{M_n - b_n}{a_n} \leq x\right) = F(x)$, where F is a non-degenerate distribution function, then the limit distribution F belongs to either the Gumbel, the Fréchet or the Weibull family[4]. These can be grouped into the generalized extreme value distribution.

→ 동일분포에서 독립적으로 추출한 변수의 샘플 중 가장 큰 값을 뽑으면, 가장 큰 값보다 클 확률은 Weibull 분포, Frechet 분포, Gumbel 분포의 형태로 만들 수 있다.

→ 극단값의 분포, 이상치의 분포 추정

$$F(x;\mu,\sigma,0) = e^{-e^{-(x-\mu)/\sigma}} \quad \text{for} \quad x \in \mathbb{R}.$$

Gumbel 분포

$$F(x;\mu,\sigma,\xi) = \begin{cases} e^{-y^{-\alpha}} & y > 0 \\ 0 & y \leq 0. \end{cases}$$

Frechet 분포

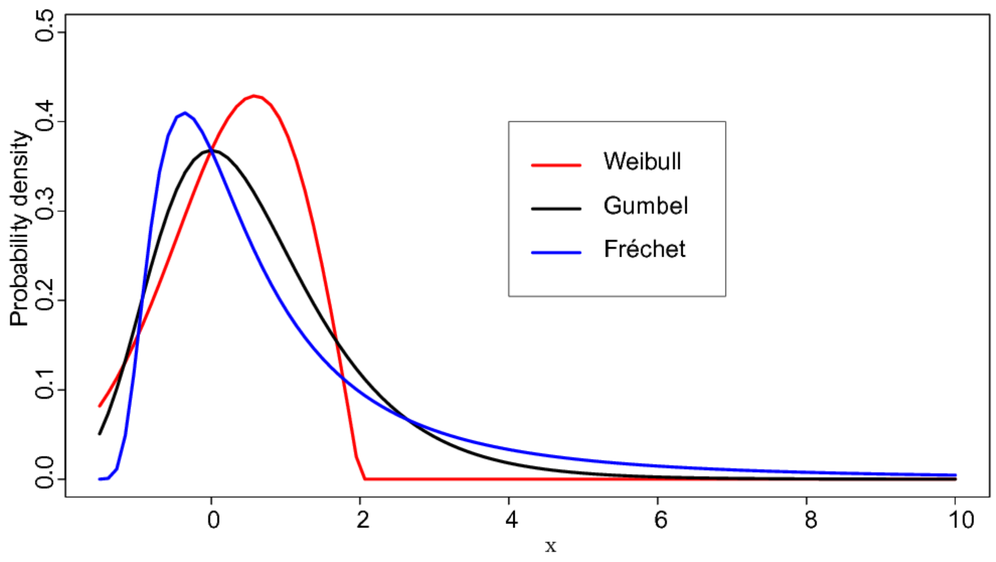
$$\boxed{F(x;\mu,\sigma,\xi) = \begin{cases} e^{-(-y)^{\alpha}} & y < 0 \\ 1 & y \geq 0 \end{cases}}$$

Weibull 분포

- Fisher, R. A. (1930). The Genetical Theory of Natural Selection. Oxford: Oxford University Press.
- de Haan, Laurens; Ferreira, Ana (2006). Extreme Value Theory: An Introduction. New York: Springer. pp. 6–12. ISBN 0-387-34471-3.

Extreme Value Theorem(the Fisher–Tippet Theorem)

Let $X_1, X_2 \ldots, X_n \ldots$ be a sequence of independent and identically-distributed random variables, and $M_n = \max\{X_1, \ldots, X_n\}$. If a sequence of pairs of real numbers (a_n, b_n) exists such that each $a_n > 0$ and $\lim_{n \to \infty} P\left(\frac{M_n - b_n}{a_n} \leq x\right) = F(x)$, where F is a non-degenerate distribution function, then the limit distribution F belongs to either the Gumbel, the Fréchet or the Weibull family[4]. These can be grouped into the generalized extreme value distribution.



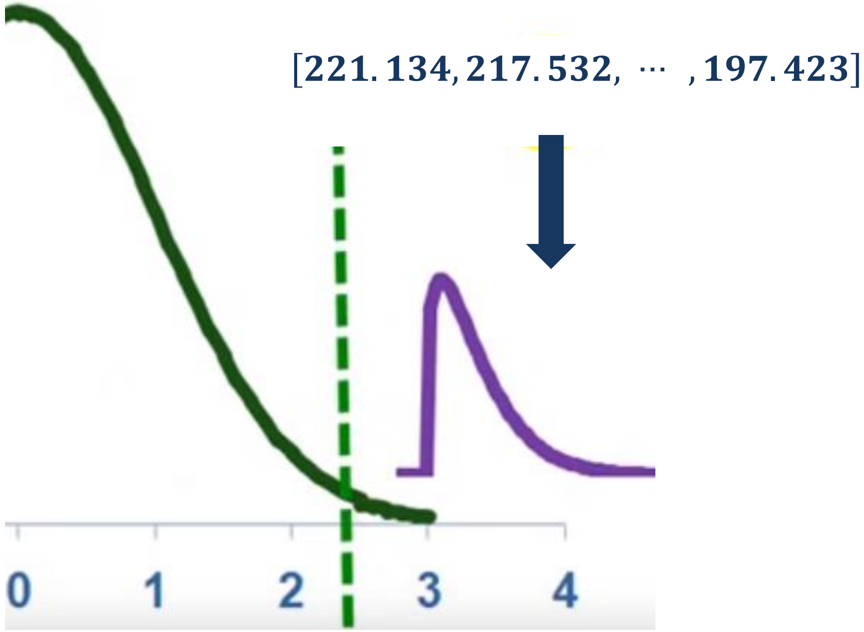
➡ 동일분포에서 독립적으로 추출한 변수의 샘플 중 가장 큰 값을 뽑으면, 가장 큰 값보다 클 확률은 Weibull 분포, Frechet 분포, Gumbel 분포의 형태로 만들 수 있다.

➡ 극단값의 분포, 이상치의 분포 추정

- Fisher, R. A. (1930). The Genetical Theory of Natural Selection. Oxford: Oxford University Press.
- de Haan, Laurens; Ferreira, Ana (2006). Extreme Value Theory: An Introduction. New York: Springer. pp. 6–12. ISBN 0-387-34471-3.

Extreme Value Theorem(the Fisher–Tippet Theorem)

Let $X_1, X_2 \ldots, X_n \ldots$ be a sequence of independent and identically-distributed random variables, and $M_n = \max\{X_1, \ldots, X_n\}$. If a sequence of pairs of real numbers (a_n, b_n) exists such that each $a_n > 0$ and $\lim_{n\to\infty} P\left(\frac{M_n - b_n}{a_n} \leq x\right) = F(x)$, where F is a non-degenerate distribution function, then the limit distribution F belongs to either the Gumbel, the Fréchet or the Weibull family[4]. These can be grouped into the generalized extreme value distribution.



→ 동일분포에서 독립적으로 추출한 변수의 샘플 중 가장 큰 값을 뽑으면, 가장 큰 값보다 클 확률은 Weibull 분포, Frechet 분포, Gumbel 분포의 형태로 만들 수 있다.

→ 극단값의 분포, 이상치의 분포 추정

가장 큰 η(=20)개 Sample로 최대 가능도 추정을 통해 분포의 파라미터를 추정.

- Fisher, R. A. (1930). The Genetical Theory of Natural Selection. Oxford: Oxford University Press.
- de Haan, Laurens; Ferreira, Ana (2006). Extreme Value Theory: An Introduction. New York: Springer. pp. 6–12. ISBN 0-387-34471-3.

OpenMax : 준비 단계(Meta Recognition)

STEP7. **각 클래스별로 평균** Logit Vector**와의 거리 중 가장 큰 η개로** Weibull **분포(극단치 분포) 생성**

Obs.	평균 Logit Vector와의 거리
N_{11}	221.134
N_{12}	217.532
…	…
N_{1a}	0.014

$$\underbrace{[221.134, 217.532, \cdots, 197.423]}_{\eta(=20)\text{개}}$$

Obs.	평균 Logit Vector와의 거리
N_{21}	195.312
N_{22}	194.904
…	…
N_{2b}	0.082

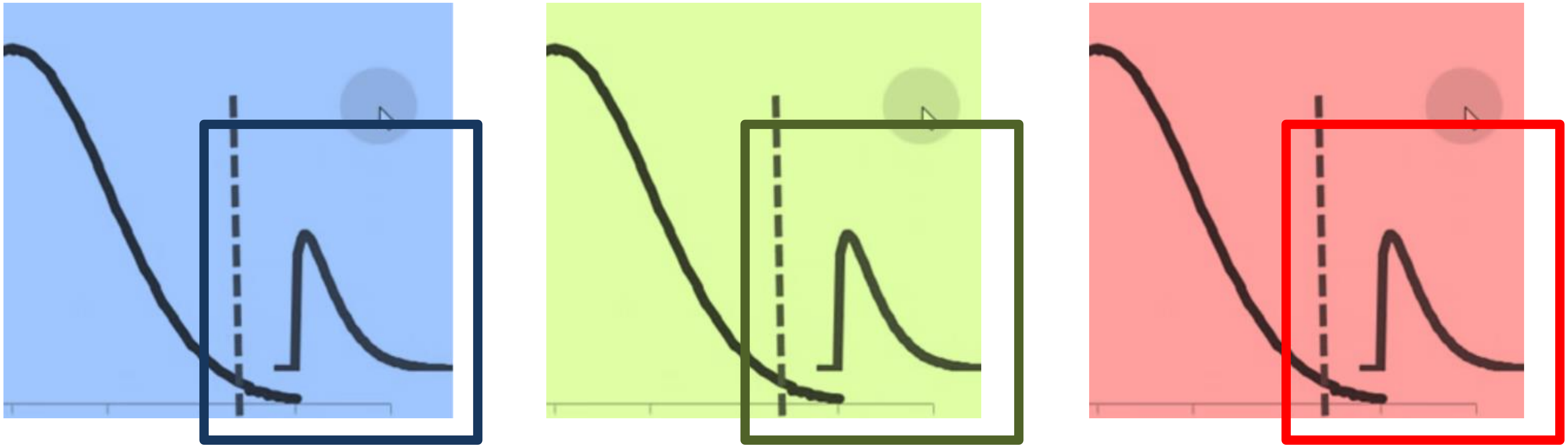
$$\underbrace{[195.312, 194.904, \cdots, 177.197]}_{\eta(=20)\text{개}}$$

Obs.	평균 Logit Vector와의 거리
N_{31}	139.641
N_{32}	121.896
…	…
N_{3c}	0.167

$$\underbrace{[139.641, 121.896, \cdots, 116.493]}_{\eta(=20)\text{개}}$$

OpenMax : 준비 단계(Meta Recognition)

STEP7. **각 클래스별로 평균** Logit Vector**와의 거리 중 가장 큰** η**개로** Weibull **분포(극단치 분포) 생성**

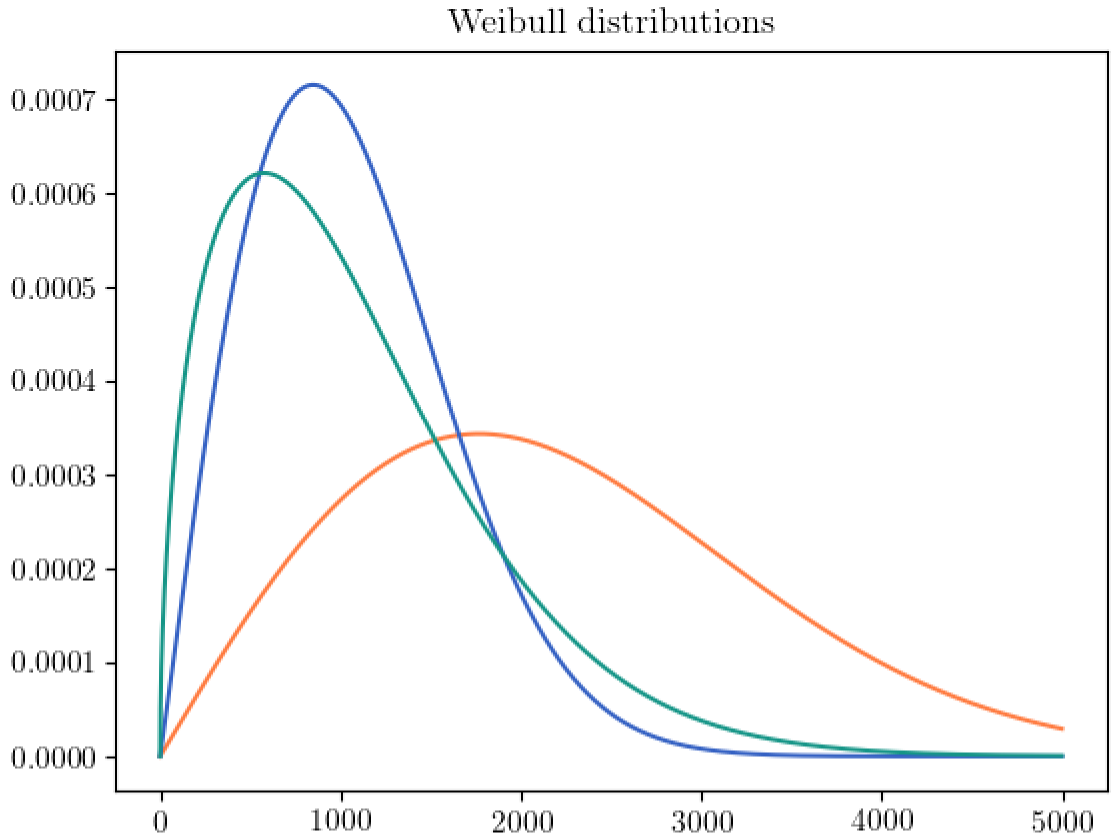


각 클래스별 평균 Logit Vector와의 거리의 극단 분포

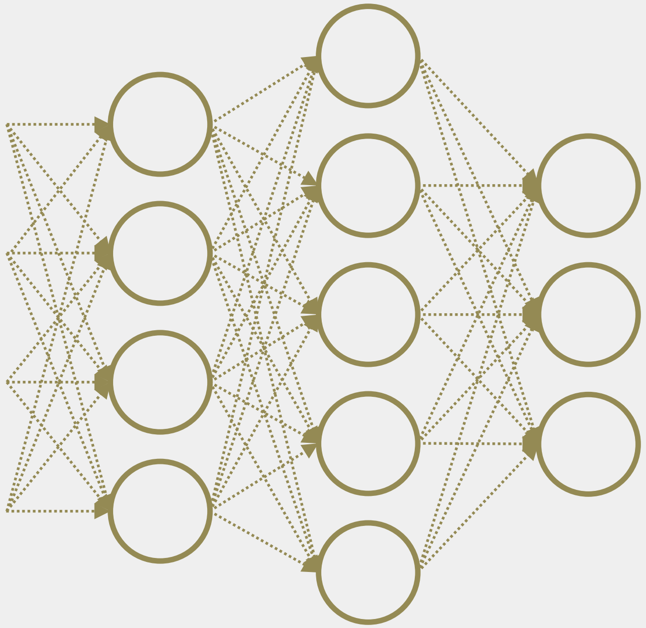
※ 주의. 각 분포는 모두 scale과 shape이 다른 분포

OpenMax : 준비 단계(Meta Recognition)

STEP7. **각 클래스별로 평균** Logit Vector**와의 거리 중 가장 큰** η**개로** Weibull **분포(극단값의 분포) 생성**



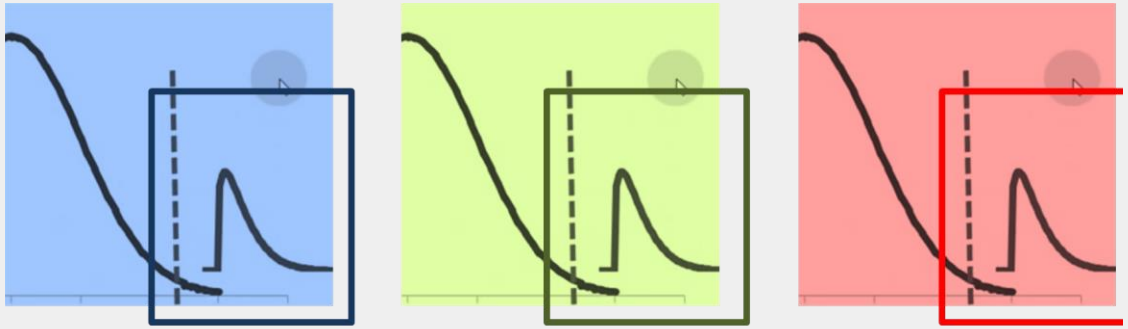
OpenMax : 실행단계



강아지/고양이/토끼 분류기

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix} \quad \mu_{cat} = \begin{bmatrix} 0.12 \\ 11.27 \\ -3.53 \end{bmatrix} \quad \mu_{rabit} = \begin{bmatrix} 1.32 \\ -2.35 \\ 9.32 \end{bmatrix}$$

각 클래스별 평균 Logit Vector

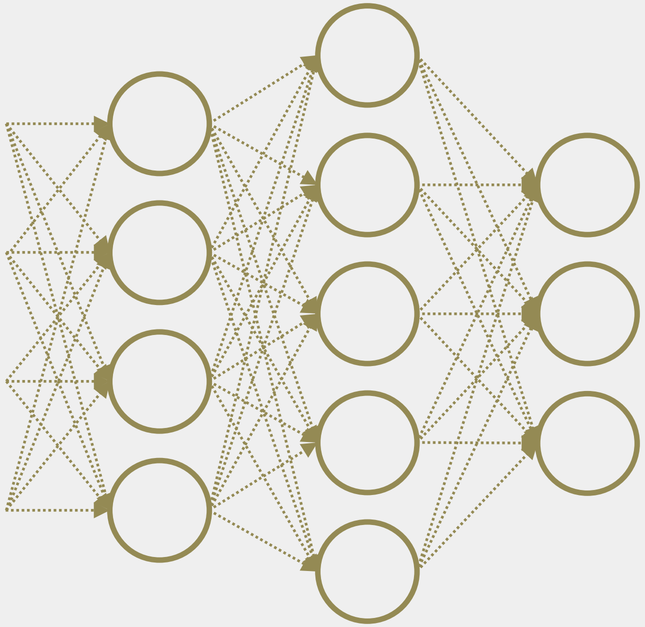


각 클래스별 평균 Logit Vector와의 거리에 대한 극단 분포

OpenMax : 실행단계

STEP1. **구축된 신경망에 테스트 데이터를 입력**

Input X

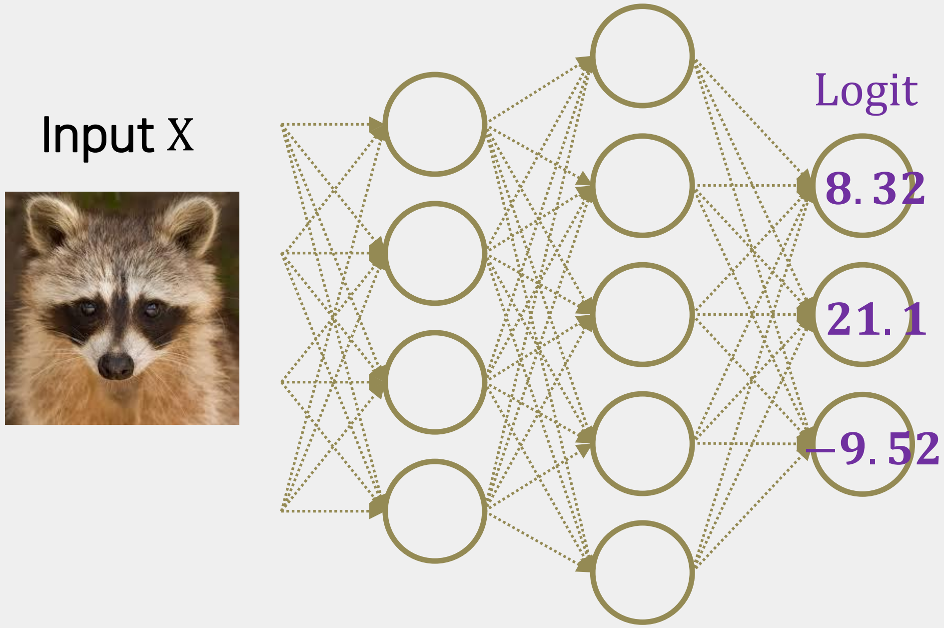


강아지/고양이/토끼 분류기

OpenMax : 실행단계

STEP2. **입력 데이터의** Logit Vector **계산 후 각 클래스별 평균** Logit Vector**와의 거리 각각 계산**

Input X

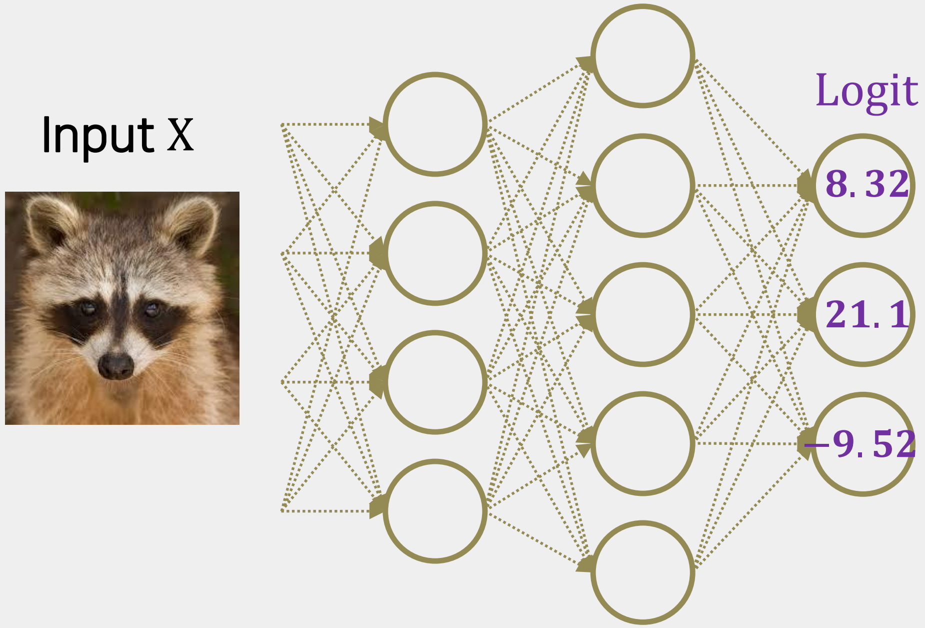


$$Logit\ Vector = \begin{bmatrix} 8.32 \\ 21.1 \\ -9.52 \end{bmatrix}$$

강아지/고양이/토끼 분류기

OpenMax : 실행단계

STEP2. **입력 데이터의** Logit Vector **계산 후 각 클래스별 평균** Logit Vector**와의 거리 각각 계산**



$$Logit\ Vector = \begin{bmatrix} 8.32 \\ 21.1 \\ -9.52 \end{bmatrix}$$

953.8481

$$\mu_{rabit} = \begin{bmatrix} 1.32 \\ -2.35 \\ 9.32 \end{bmatrix}$$

<토끼 클래스 평균 Logit Vector>

201.725

$$\mu_{cat} = \begin{bmatrix} 0.12 \\ 11.27 \\ -3.53 \end{bmatrix}$$

<고양이 클래스 평균 Logit Vector>

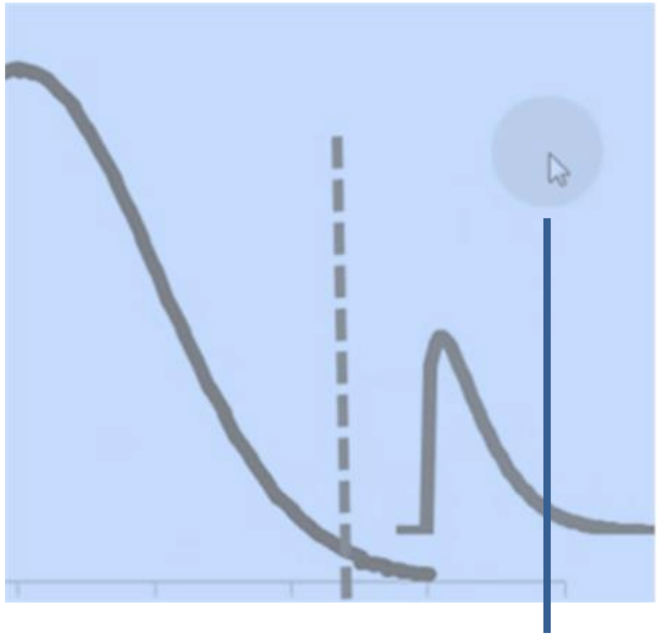
586.29

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

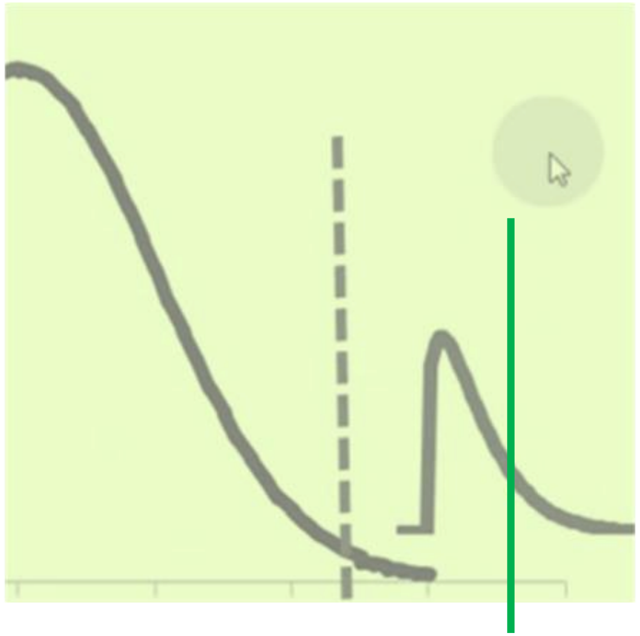
OpenMax : 실행단계

STEP3. **각각의 극단 분포의** CDF**를 통해 각 클래스 평균** Logit Vector**와의 거리의 극단 확률 계산**



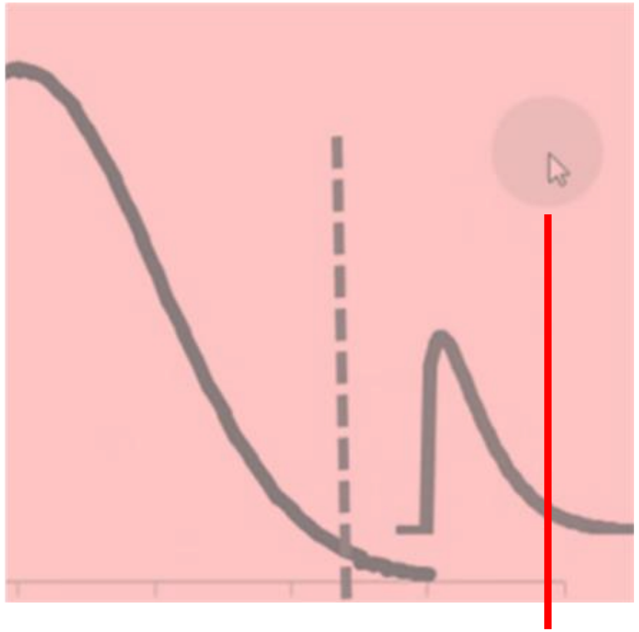
586.29

강아지 클래스



201.725

고양이 클래스



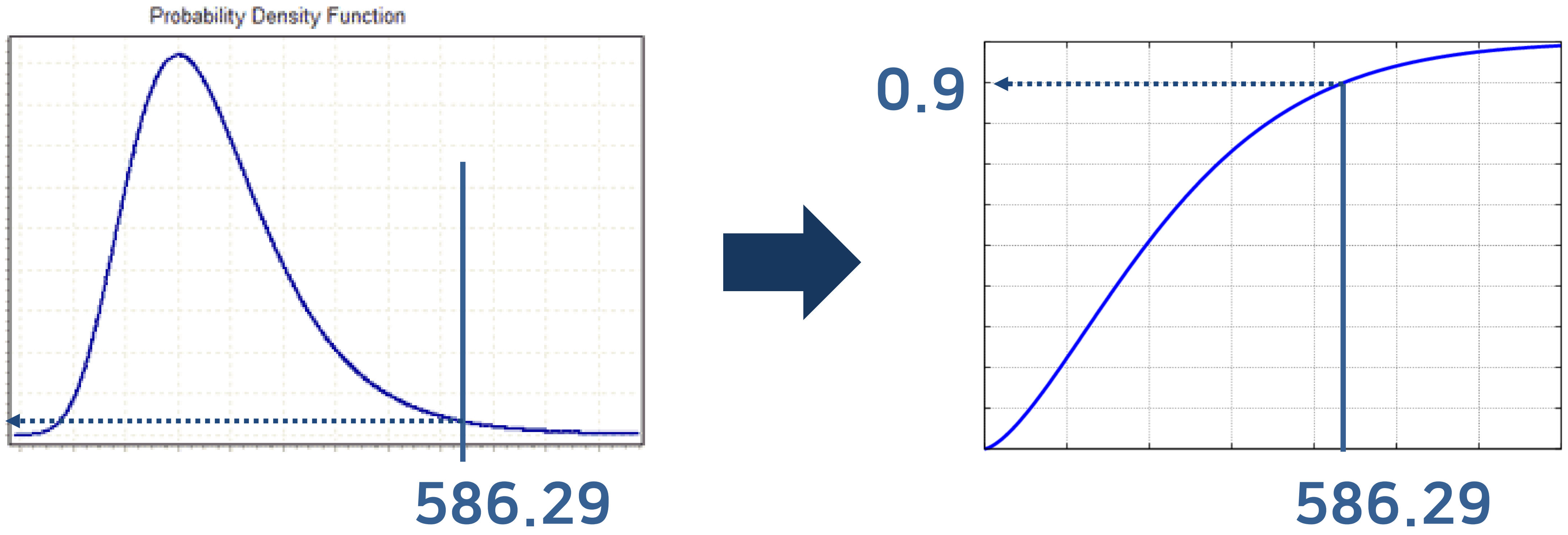
953.8481

토끼 클래스

※ 주의. 각 분포는 모두 scale과 shape이 다른 분포

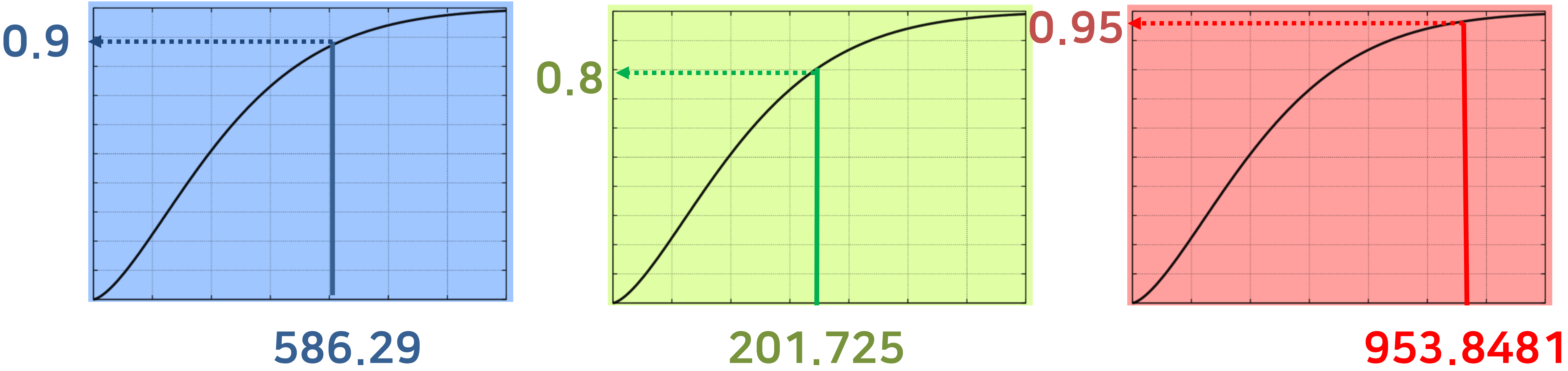
OpenMax : 실행단계

STEP3. **각각의 극단 분포의** CDF**를 통해 각 클래스 평균** Logit Vector**와의 거리의 극단 확률 계산**



OpenMax : 실행단계

STEP3. **각각의 극단 분포의** CDF**를 통해 각 클래스 평균** Logit Vector**와의 거리의 극단 확률 계산**

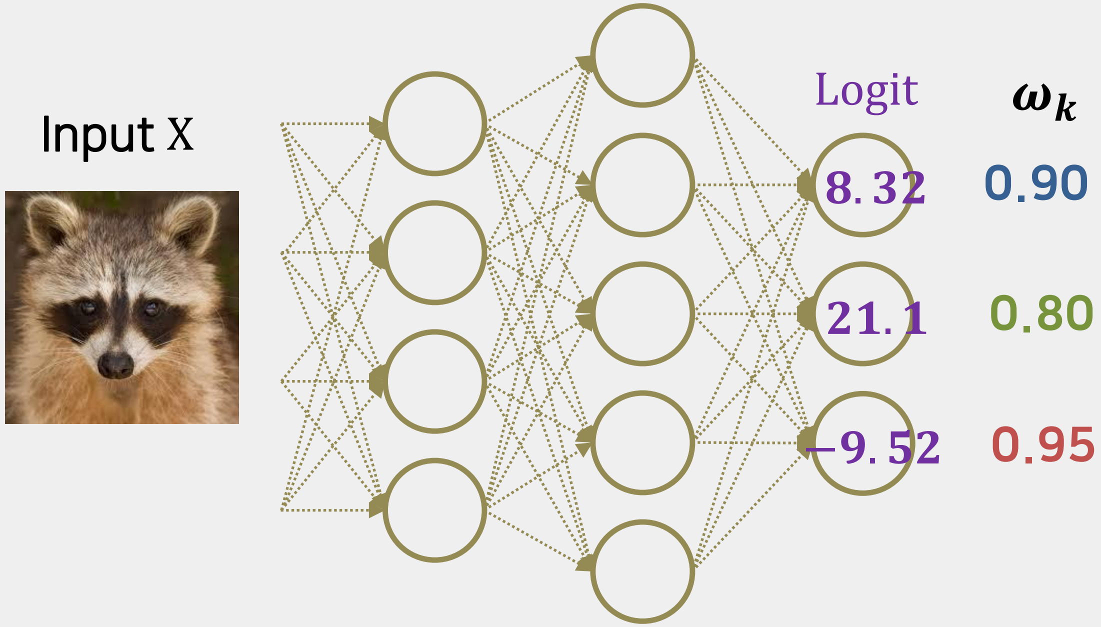


강아지 클래스 고양이 클래스 토끼 클래스

※ 주의. 각 분포는 모두 scale과 shape이 다른 분포

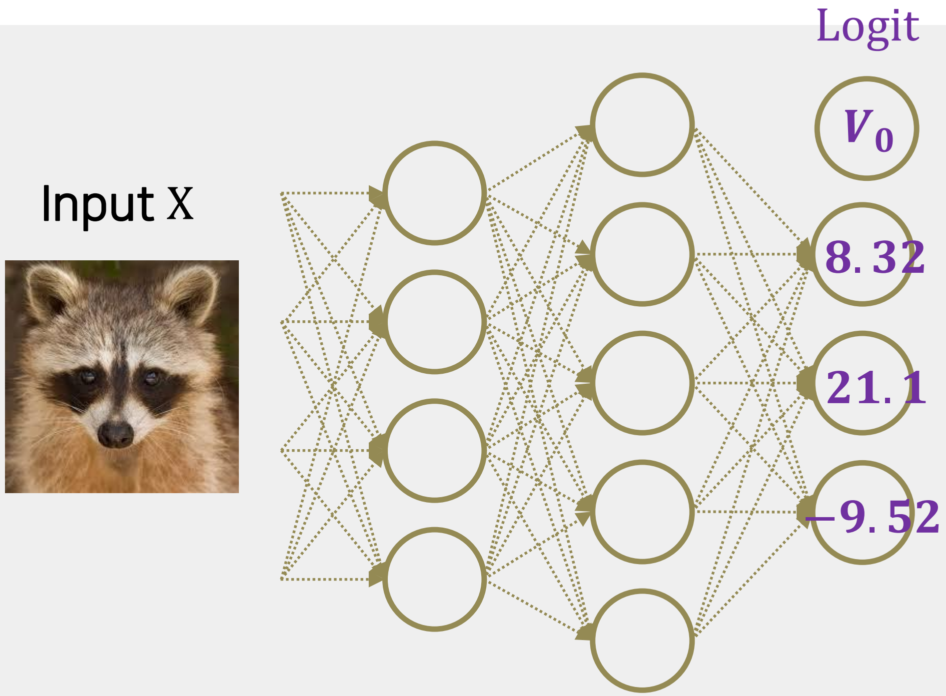
OpenMax : 실행단계

STEP4. **극단분포의** **CDF 값(Logit Vector가 해당 클래스가 아닐 확률)을 ω_k로** **두어** Logit Vector **업데이트**



OpenMax : 실행단계

STEP4. **극단분포의** CDF **값**(Logit Vector**가 해당 클래스가 아닐 확률**)**을** ω_k**로 두어** Logit Vector **업데이트**



$= 0.90 \times 8.32 + 0.80 \times 21.1 + 0.95 \times -9.52 = 15.324$

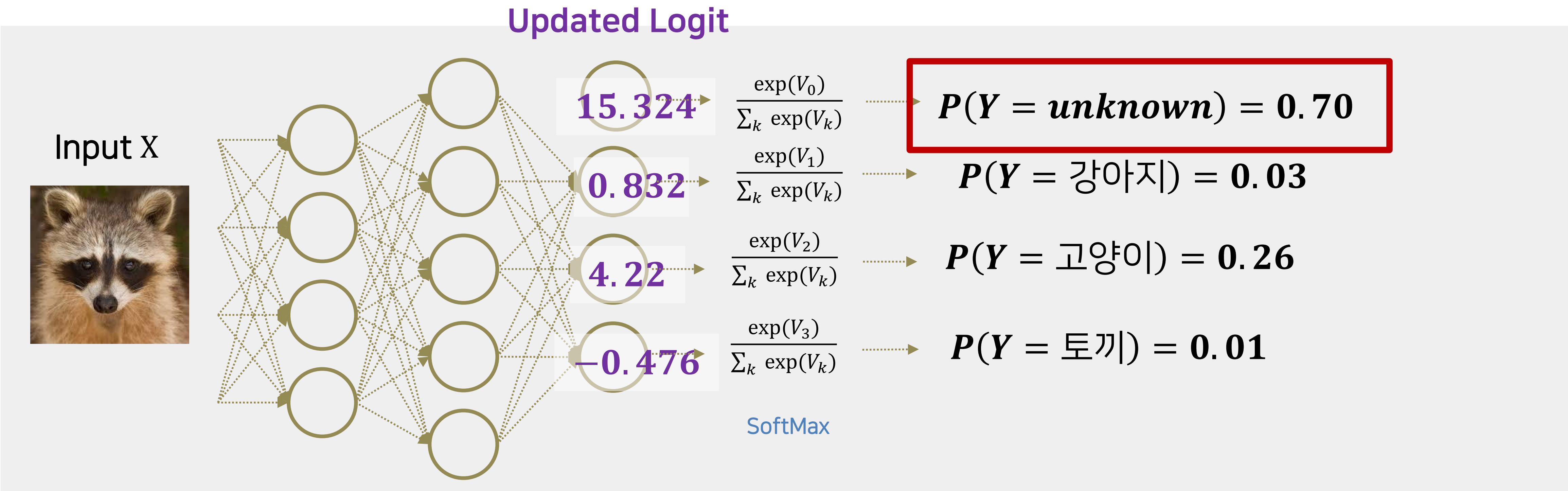
$= 8.32-(0.90 \times 8.32) = 0.832$

$= 21.1-(0.80 \times 21.1) = 4.22$

$= -9.52-(0.95 \times -9.52) = -0.476$

OpenMax : 실행단계

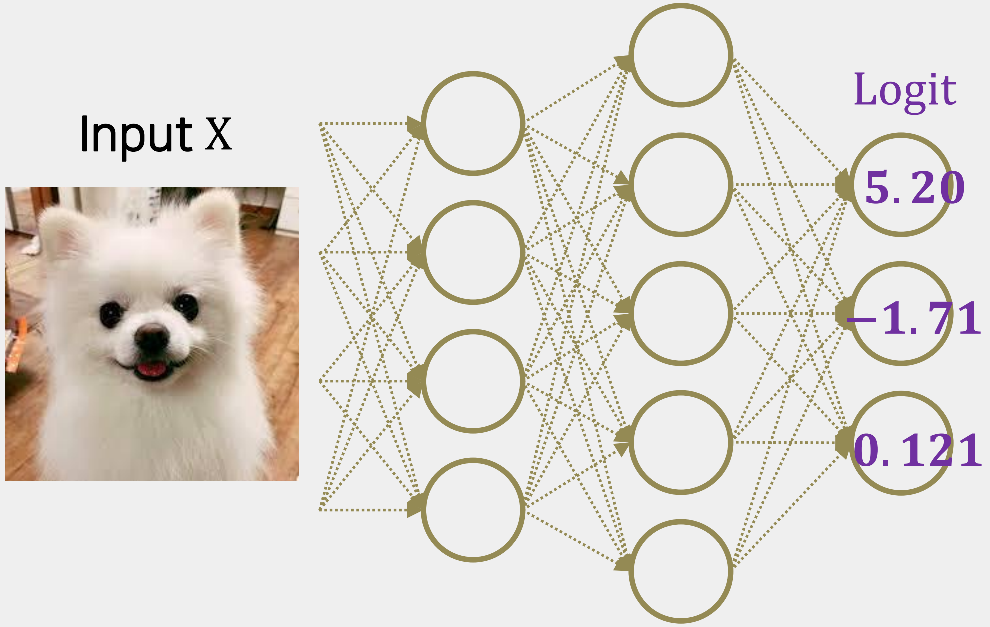
STEP5. **업데이트 된** Logit Vector**를** SoftMax Layer**에 통과 시켜 결과 도출**



➡Unknown Class

OpenMax : 실행단계

STEP4. **극단분포의** CDF **값**(Logit Vector**가 해당 클래스가 아닐 확률**)**을** ω_k**로 두어** Logit Vector **업데이트**



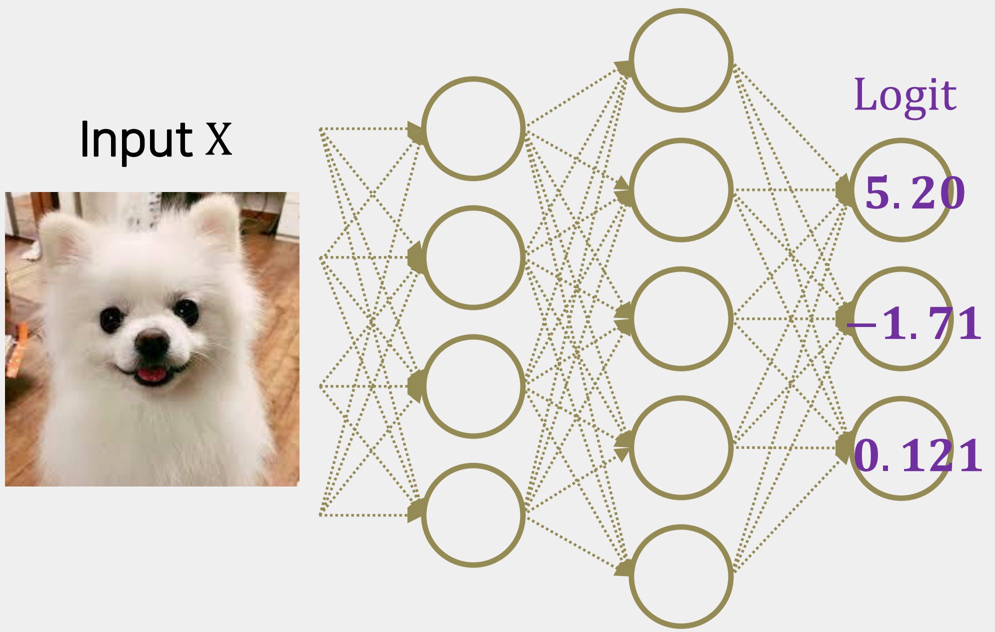
$$Logit\ Vector = \begin{bmatrix} 5.20 \\ -1.71 \\ 0.121 \end{bmatrix}$$

$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

<강아지 클래스 평균 Logit Vector>

OpenMax : 실행단계

STEP4. **극단분포의** CDF **값**(Logit Vector**가 해당 클래스가 아닐 확률)을** ω_k**로 두어** Logit Vector **업데이트**



$$\mu_{rabit} = \begin{bmatrix} 1.32 \\ -2.35 \\ 9.32 \end{bmatrix}$$

<토끼 클래스 평균 Logit Vector>

$$\boldsymbol{Logit\ Vector} = \begin{bmatrix} 5.20 \\ -1.71 \\ 0.121 \end{bmatrix}$$

$$\mu_{cat} = \begin{bmatrix} 0.12 \\ 11.27 \\ -3.53 \end{bmatrix}$$

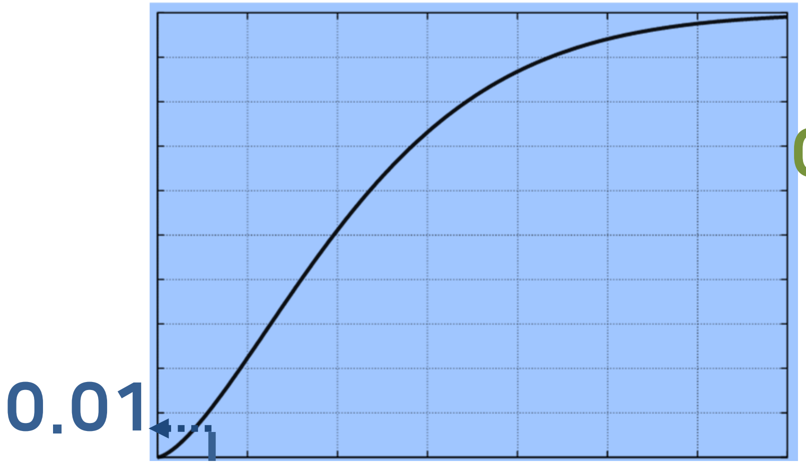
<고양이 클래스 평균 Logit Vector>

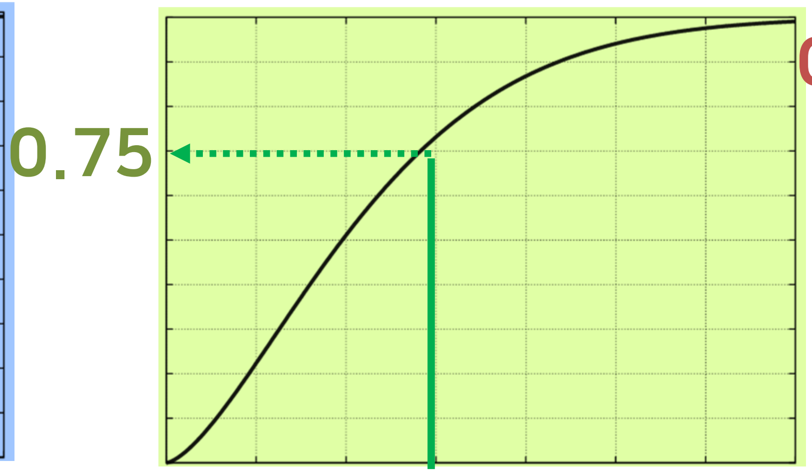
$$\mu_{dog} = \begin{bmatrix} 5.12 \\ -1.12 \\ 0.12 \end{bmatrix}$$

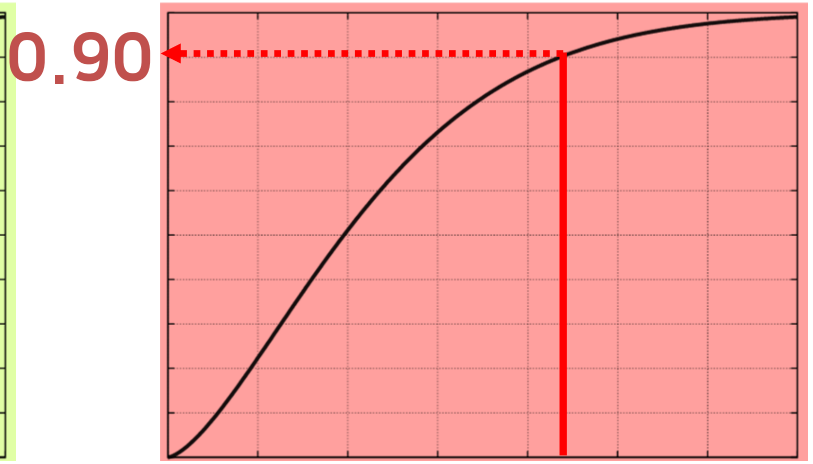
<강아지 클래스 평균 Logit Vector>

OpenMax : 실행단계

STEP3. **각각의 극단 분포의** CDF**를 통해 각 클래스 평균** Logit Vector**와의 거리의 극단 확률 계산**







강아지 클래스

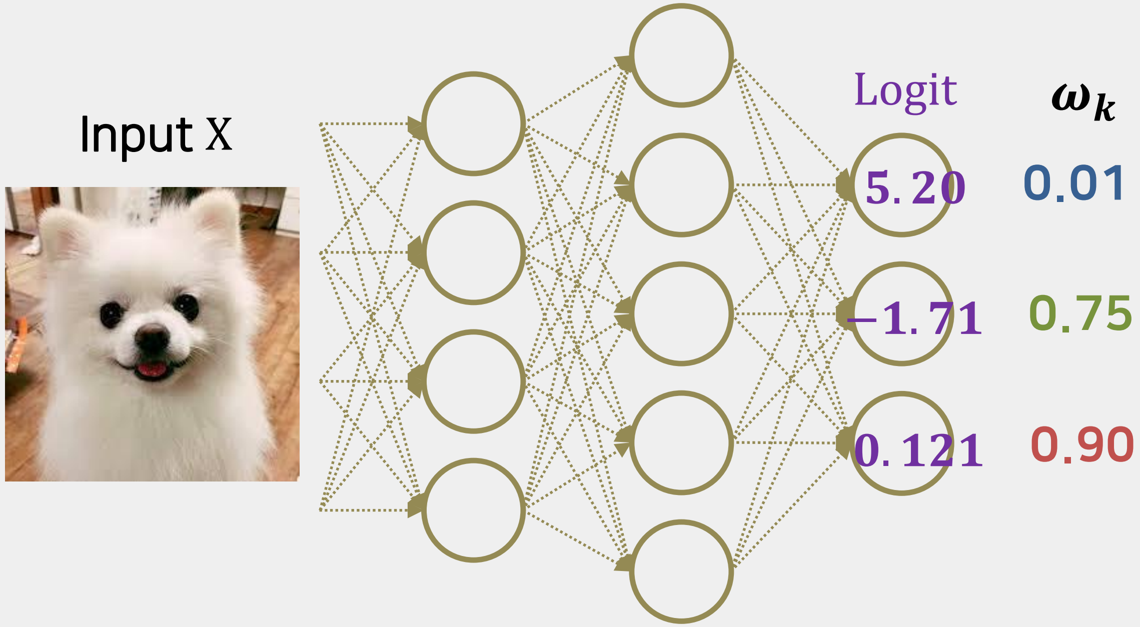
고양이 클래스

토끼 클래스

※ 주의. 각 분포는 모두 scale과 shape이 다른 분포

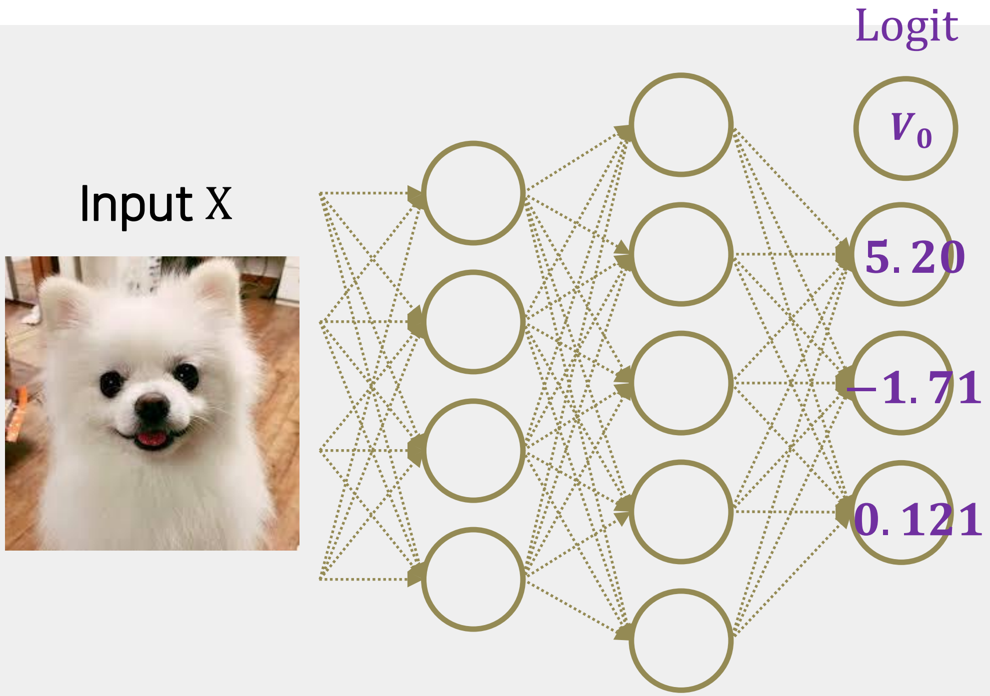
OpenMax : 실행단계

STEP4. **극단분포의** CDF **값**(Logit Vector**가 해당 클래스가 아닐 확률**)**을** ω_k**로 두어** Logit Vector **업데이트**



OpenMax : 실행단계

STEP4. **극단분포의** CDF **값**(Logit Vector**가 해당 클래스가 아닐 확률**)**을** ω_k**로 두어** Logit Vector **업데이트**



$= 0.01 \times 5.20 + 0.75 \times -1.71 + 0.90 \times 0.121 = -0.000968$

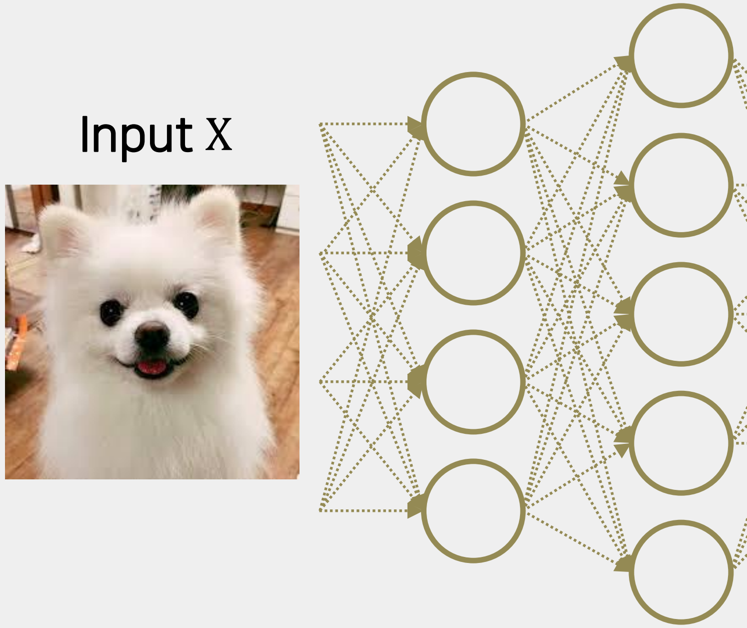
$= 5.20-(0.01 \times 5.20) = 5.148$

$= -1.71-(0.75 \times -1.71) = -0.4275$

$= 0.121-(0.90 \times 0.121) = 0.0121$

OpenMax : 실행단계

STEP5. 업데이트 된 Logit Vector를 SoftMax Layer에 통과 시켜 결과 도출



Updated Logit

Updated Logit	SoftMax	
−0.000968	$\frac{\exp(V_0)}{\sum_k \exp(V_k)}$	$P(Y = unknown) = 0.02$
5.148	$\frac{\exp(V_1)}{\sum_k \exp(V_k)}$	$\boxed{P(Y = \text{강아지}) = 0.93}$
−0.4275	$\frac{\exp(V_2)}{\sum_k \exp(V_k)}$	$P(Y = \text{고양이}) = 0.01$
0.0121	$\frac{\exp(V_3)}{\sum_k \exp(V_k)}$	$P(Y = \text{토끼}) = 0.04$

SoftMax

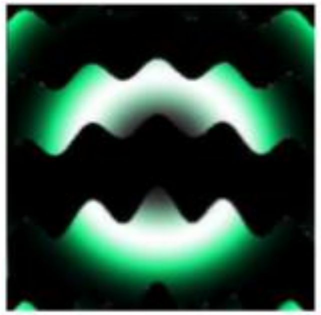
➡Dog Class

OpenMax : 실험결과

	Baseball	Hammerhead
		
SoftMax	0.94	0.57
OpenMax	0.94	0.58

학습 데이터에 포함된 클래스

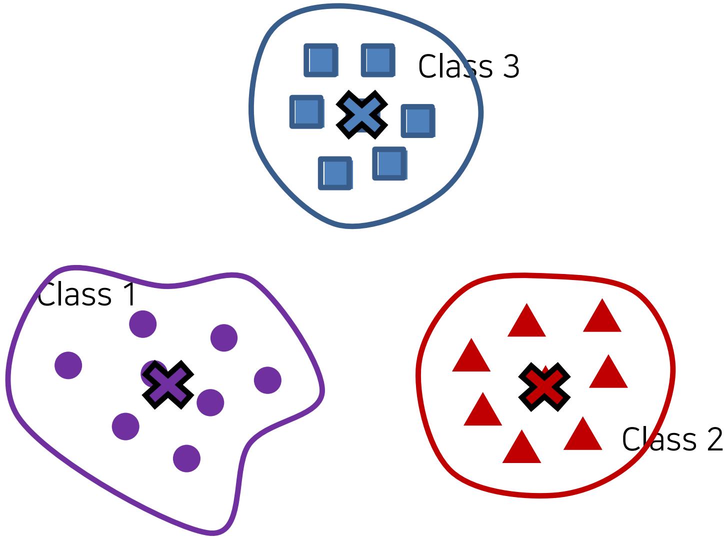
훈련데이터에 포함된 클래스에 대한 분류 성능은 OpenMax와 SoftMax 간의 큰 차이가 없다

	Baseball	Hammerhead
		
SoftMax	**1.0**	**0.98**
OpenMax	**0.00**	**0.00**

Unknown 클래스

훈련데이터에 포함되지 않은 클래스에 대한 분류 성능은 SoftMax의 경우 높은 확신으로 오분류한 데이터에 대해 OpenMax는 우수한 성능으로 거부한다.

OpenMax : 요약



OpenMax

- 신경망의 학습이 모두 끝난 후, 후처리를 통한 Open Set Recognition
- 각 클래스 별 평균으로부터의 거리에 대한 극단분포를 이용하여 결정경계 제한
- Unknown Class에 대한 Logit 을 정의하고, 기존의 Logit Update

Q&A