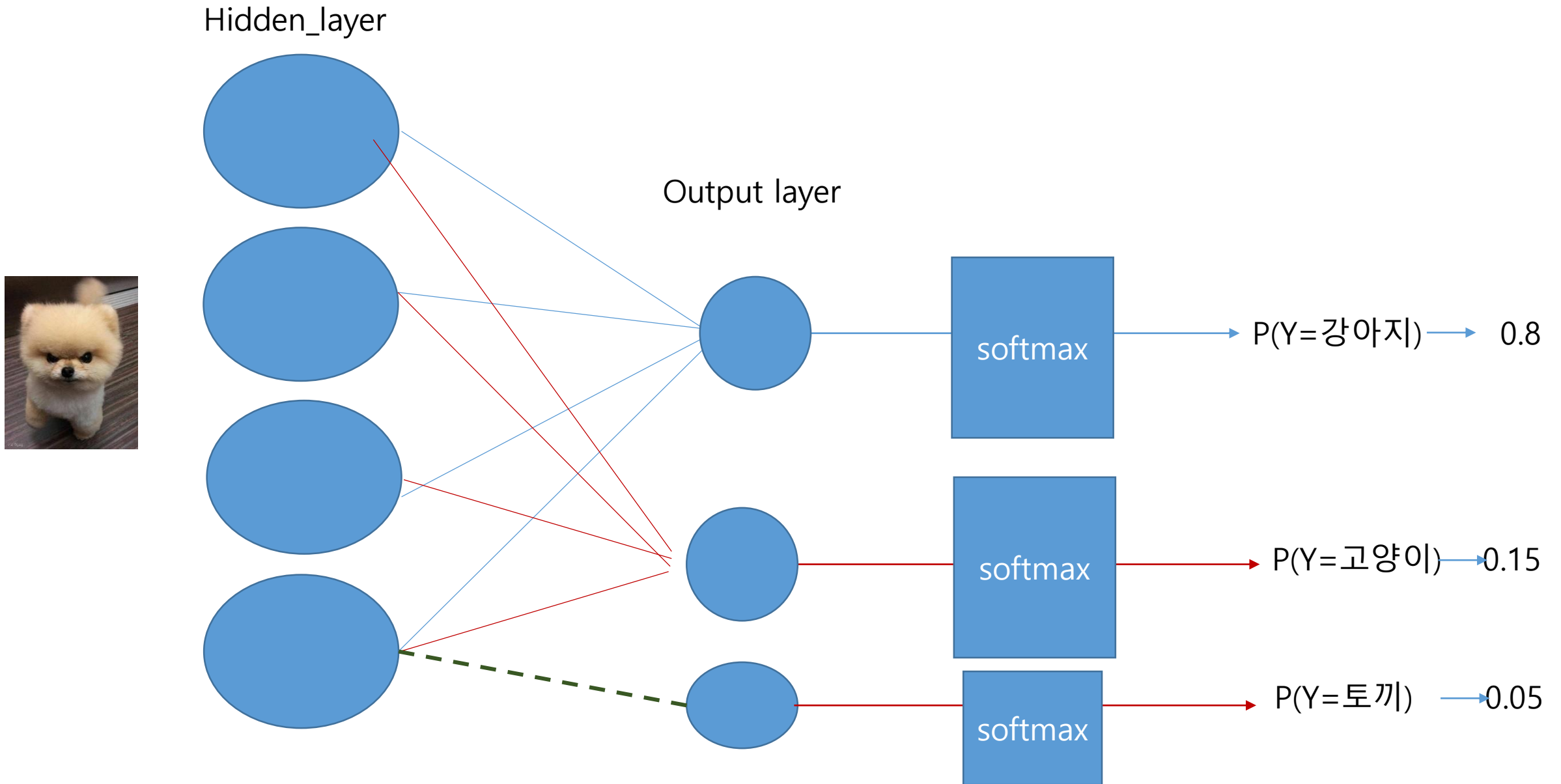


분류 모델의 한계

Towards open set deep networks(CVPR 2016)

분류 심층 신경망의 구조



Softmax, Logit vector

softmax

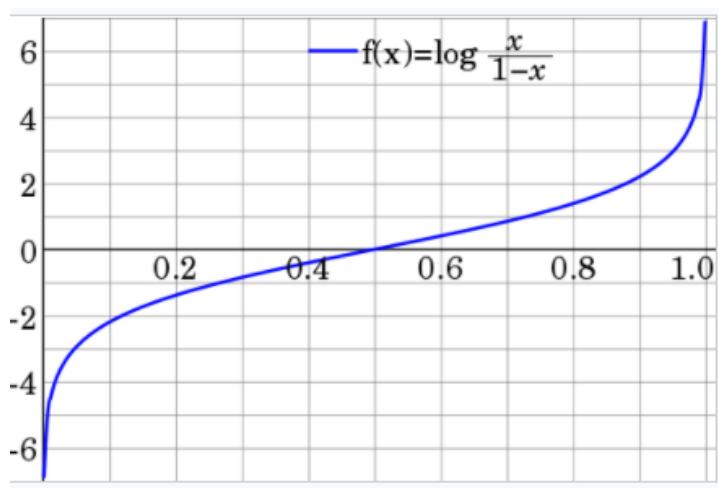
$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)}$$

```
>>> a = np.array([0.3, 2.9, 4.0])
>>> y = softmax(a)
>>> print(y)
[ 0.01821127  0.24519181  0.73659691]
>>> np.sum(y)
1.0
```

①: y[0]의 확률, ②: y[1]의 확률 ...

각 클래스일 확률을 반환해줌

Logit vector



Sigmoid 함수의 역함수
신경망의 마지막 계층은 로짓 계층으로 예측에 대한
원시 값을 반환

1. 신경망의 연산을 통해 Logit Vector로 축약

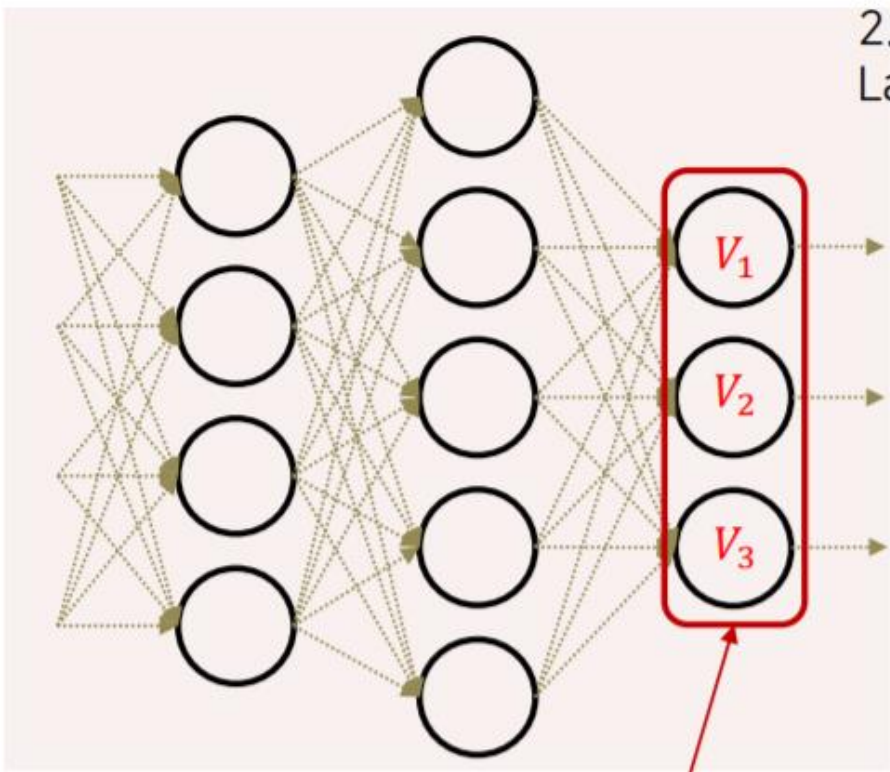
2. 각 클래스에 대응되는 Logit값을 SoftMax Layer에서 0~1 사이의 확률로 반환



	x_1	x_2	...	x_p
N_1

Pixel by pixel variable

...



Neural Net

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

Logits
값의 범위 : $(-\infty, \infty)$

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

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

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

SoftMax

SoftMax Activation
값의 범위 : $[0,1]$

$P(Y = \text{강아지})$

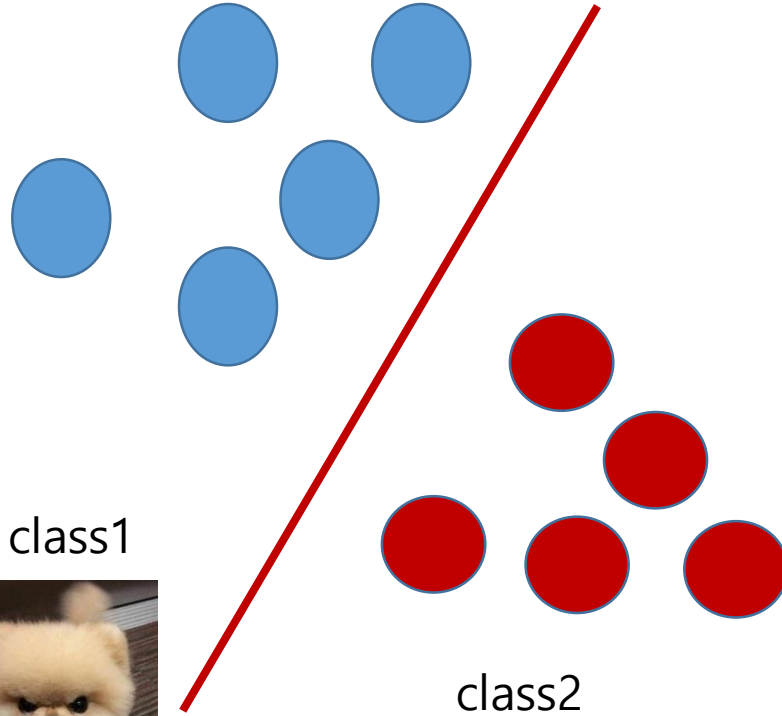
$P(Y = \text{고양이})$

$P(Y = \text{토끼})$

분류 모델의 한계점



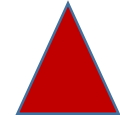
New class



class1



class2

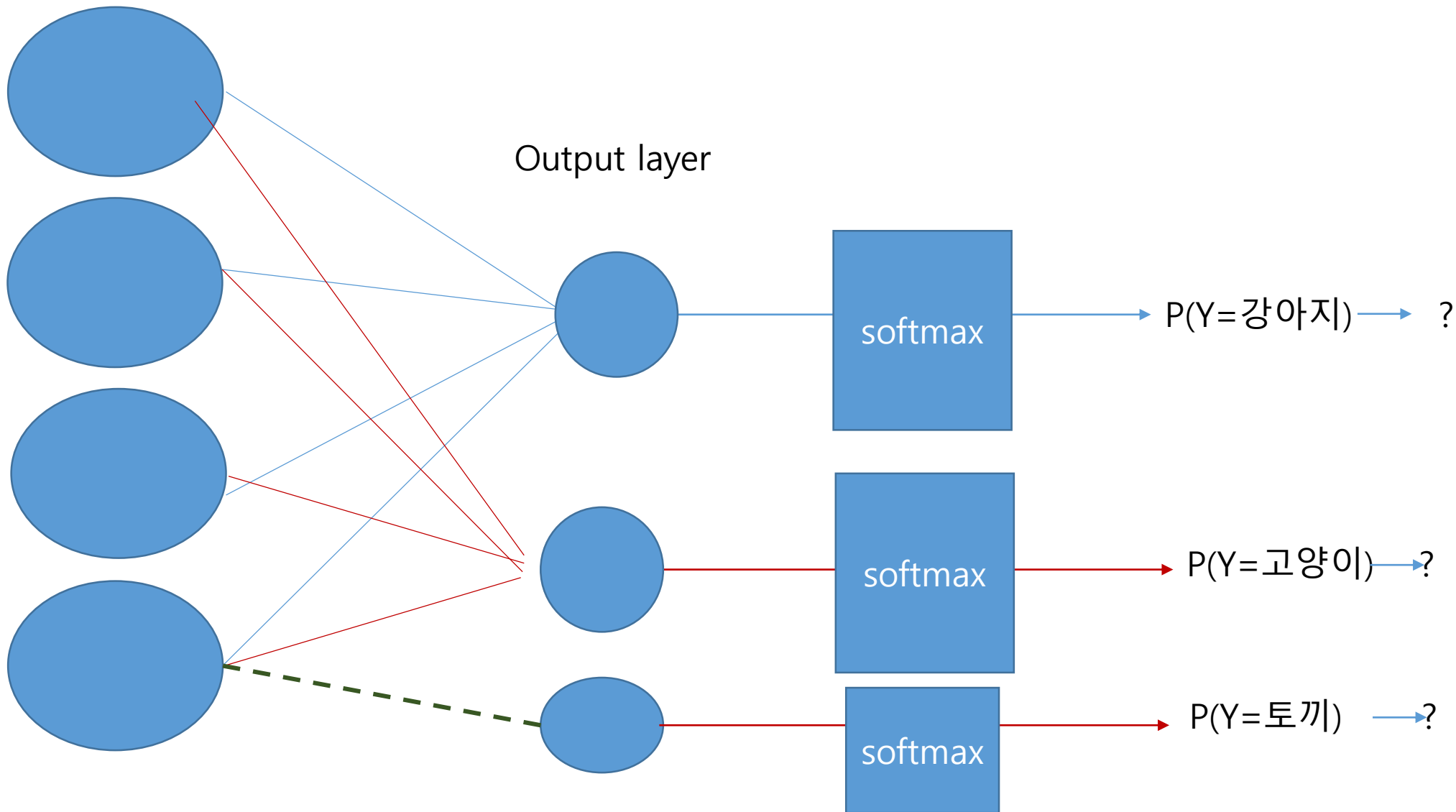


New class

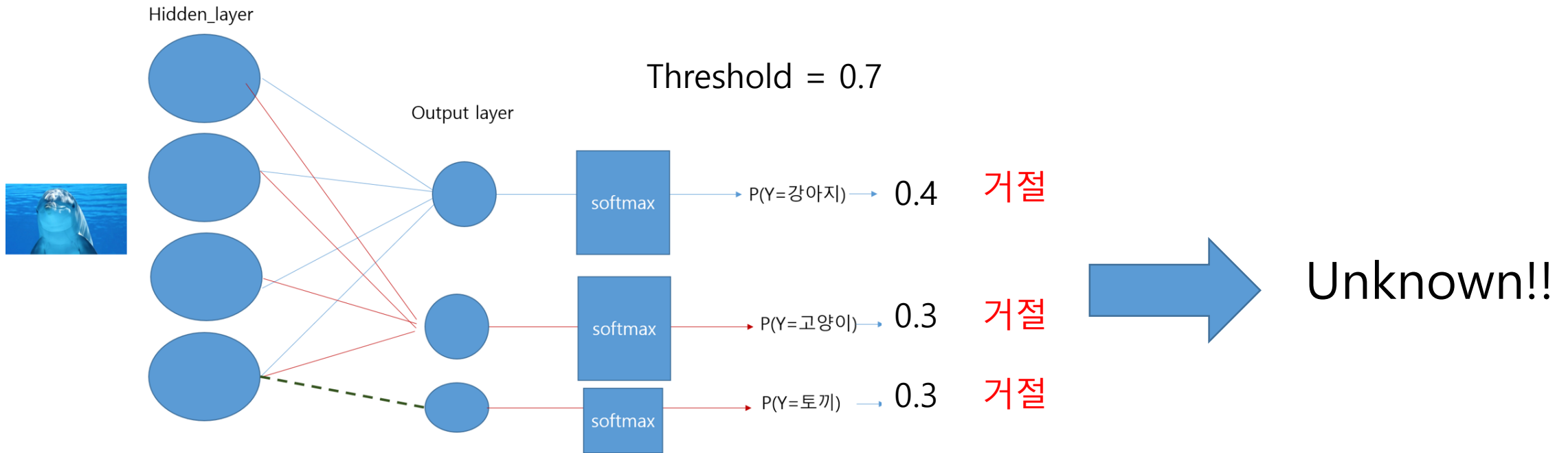
만약 전혀 다른 동물이 들어간다면?

Hidden_layer

Output layer



이를 극복하기 위해선..?

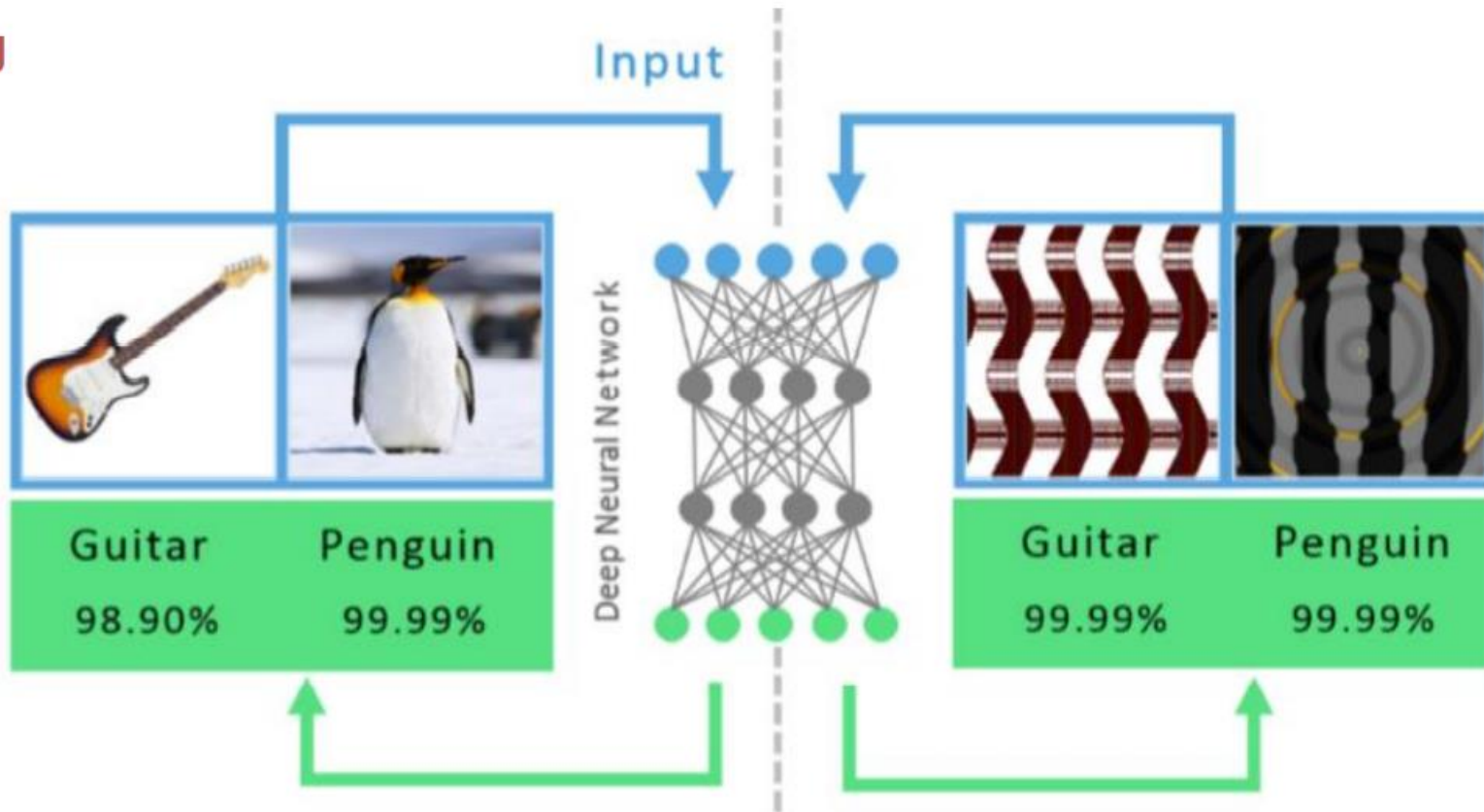


But

Training

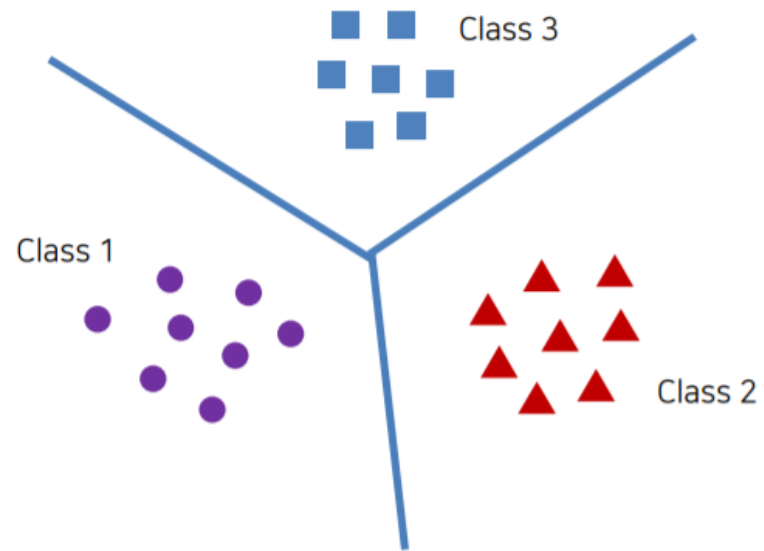
Input

Test

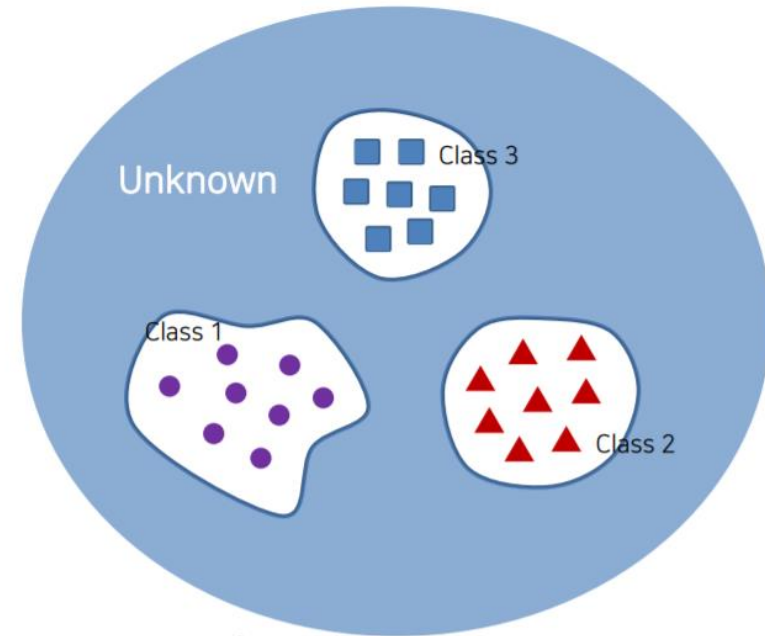


Threshold 적용 시킬 수 없다.

Closed Set vs OpenSet Recognition



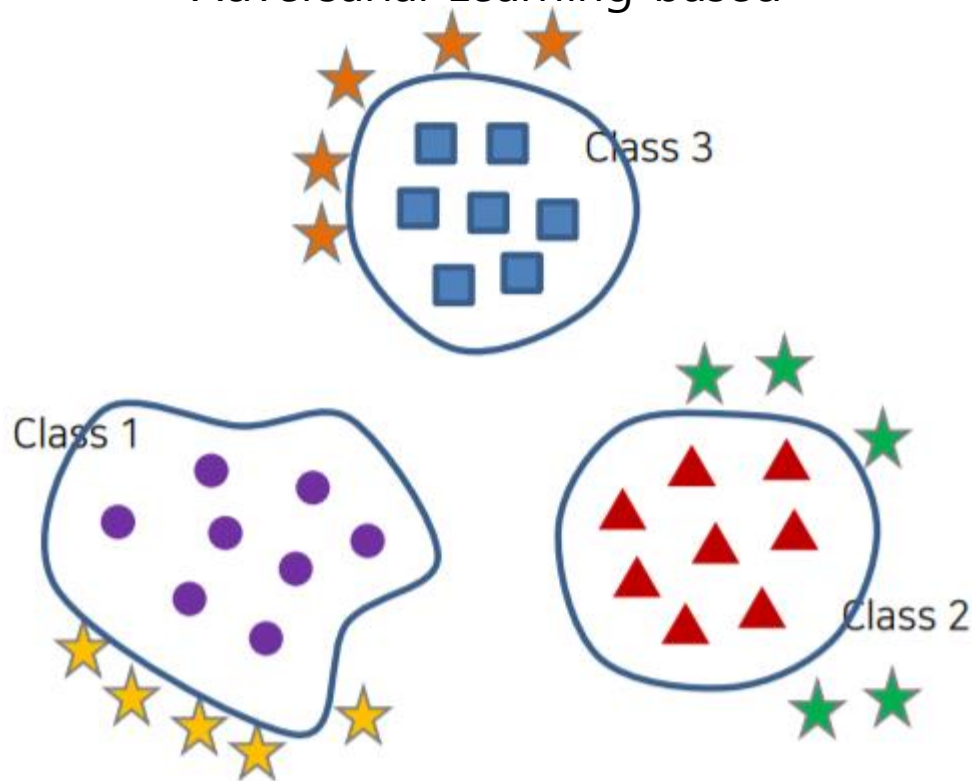
Closed Set Classification



Open Set Recognition

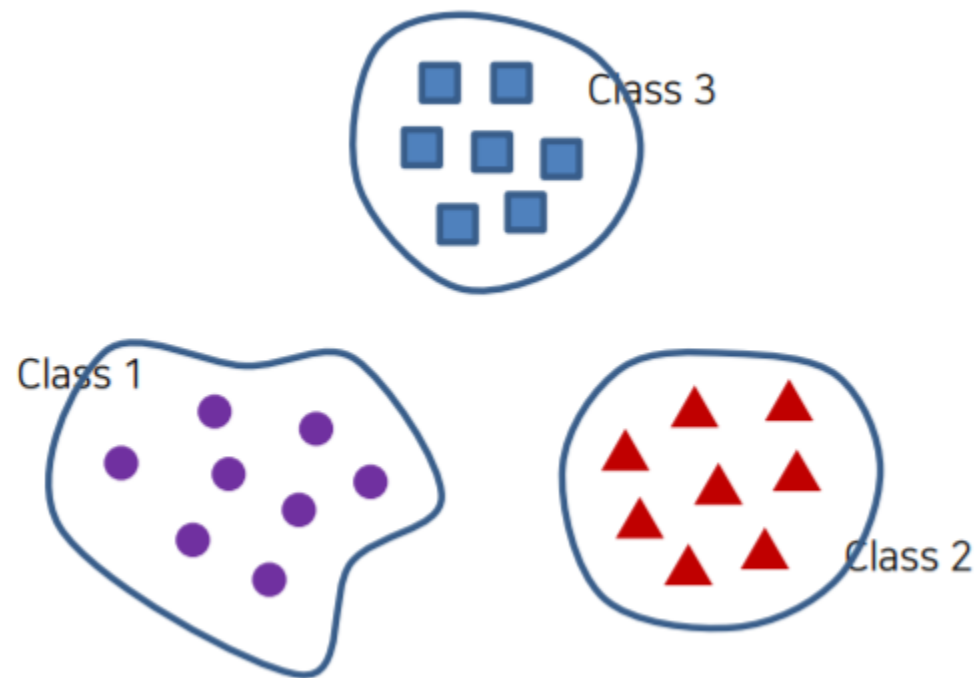
Open Set Recognition 종류

Adversarial Learning-based



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

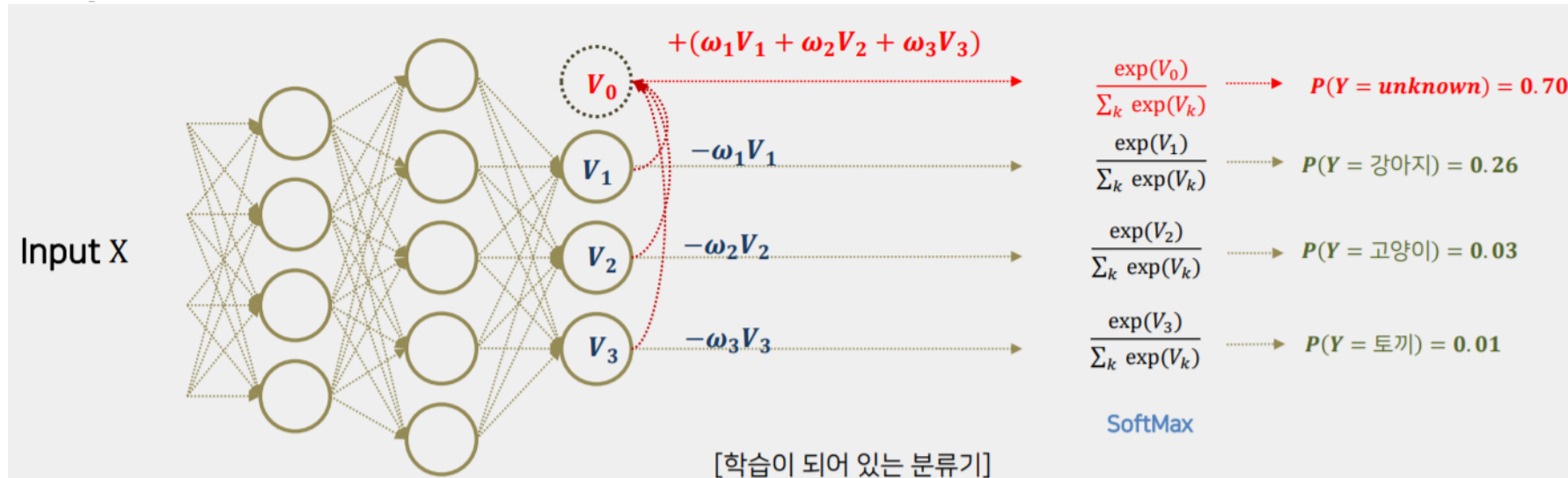
Distance-Based



평균부터 떨어진 거리, 마진 등을 통하여 결정 경계 생성

논문에선 EuCos 거리 사용

OpenMAX



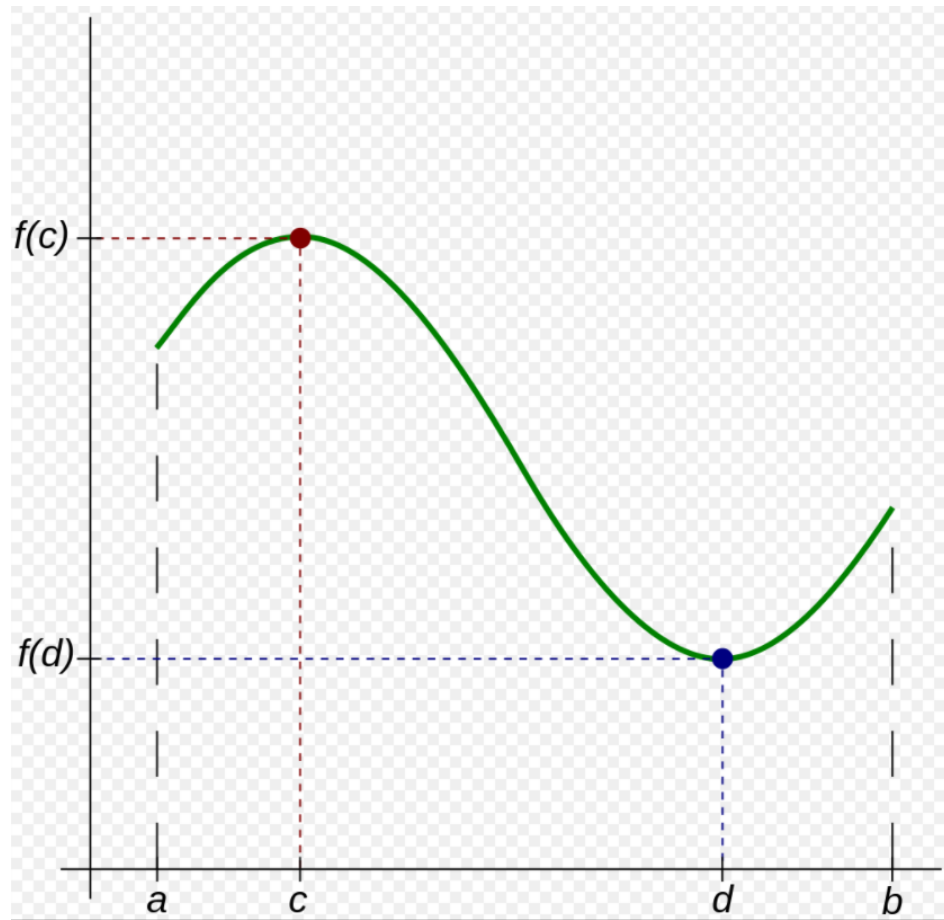
V_k = logit 값

W_k = 분류기가 k class로 잘못 분류했을 때 대응하는 가중치

W_k 를 어떻게 정의하냐?

Extreme value theorem(최대최소정리)에 기반하여
평균 Logit Vector로부터의 거리에 대한 극단값(이상치)의 분포를 통
해 w_k 를 정의한다

Extreme value theorem(1)



닫힌구간 $[a, b]$ 에서 연속인 함수 f 는
최댓값 $f(c)$ 와 최솟값 $f(d)$ 를 반드시
갖는다

OpenMAX Flow

1. 학습 데이터 중 분류기가 정확하게 선별한 데이터 선별
2. 선별된 데이터의 X(input)데이터를 클래스별로 분리
3. 각 클래스 별로 선별된 데이터를 이용하여 Logit Vector 계산
4. 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

$$\bar{V}_1 = 5.12 \quad \bar{V}_2 = -1.12 \quad \bar{V}_3 = 0.12$$

<강아지 클래스 Logit Vector Matrix>

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

<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>

4번 과정 수행 후 출력값

OpenMAX Flow

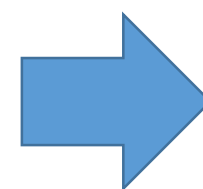
5. 각 클래스 별 평균 Logit Vector와의 거리 계산(반복)

Ex.

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

$$\bar{V}_1 = 5.12 \quad \bar{V}_2 = -1.12 \quad \bar{V}_3 = 0.12$$

<강아지 클래스 Logit Vector Matrix>


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

N11 Logit_avg

간단하게 유클리디안 거리 사용

OpenMAX Flow

6. 각 클래스 별로 계산된 거리 Matrix를 거리 기준 내림차순으로 정렬 후 평균 Logit Vector와 가장 거리가 큰 n개를 각 클래스 별로 추출

왜 굳이 뽑을까???

Extreme value theorem(2) The Fisher-Tippett Theorem

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.

➡ 극단값, 이상치의 분포를 추정할 수 있다!!!!

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

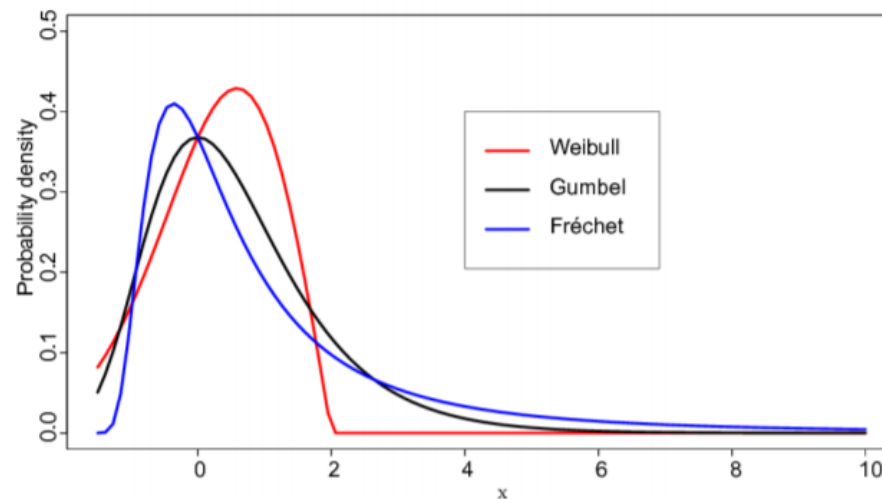
Gumbel 분포

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

Frechet 분포

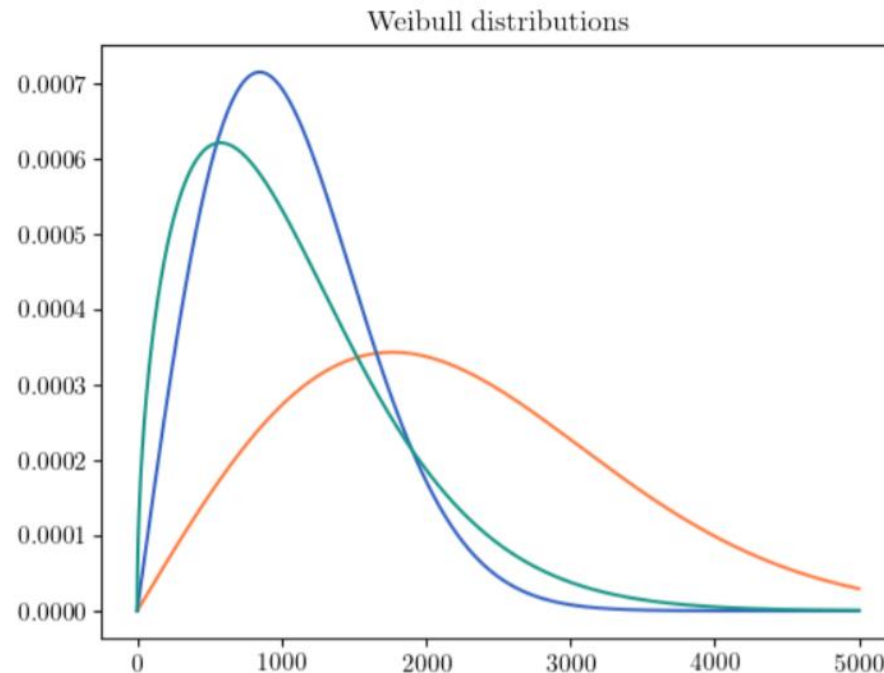
$$F(x; \mu, \sigma, \xi) = \begin{cases} e^{-(-y)^{\alpha}} & y < 0 \\ 1 & y \geq 0 \end{cases}$$

Weibull 분포



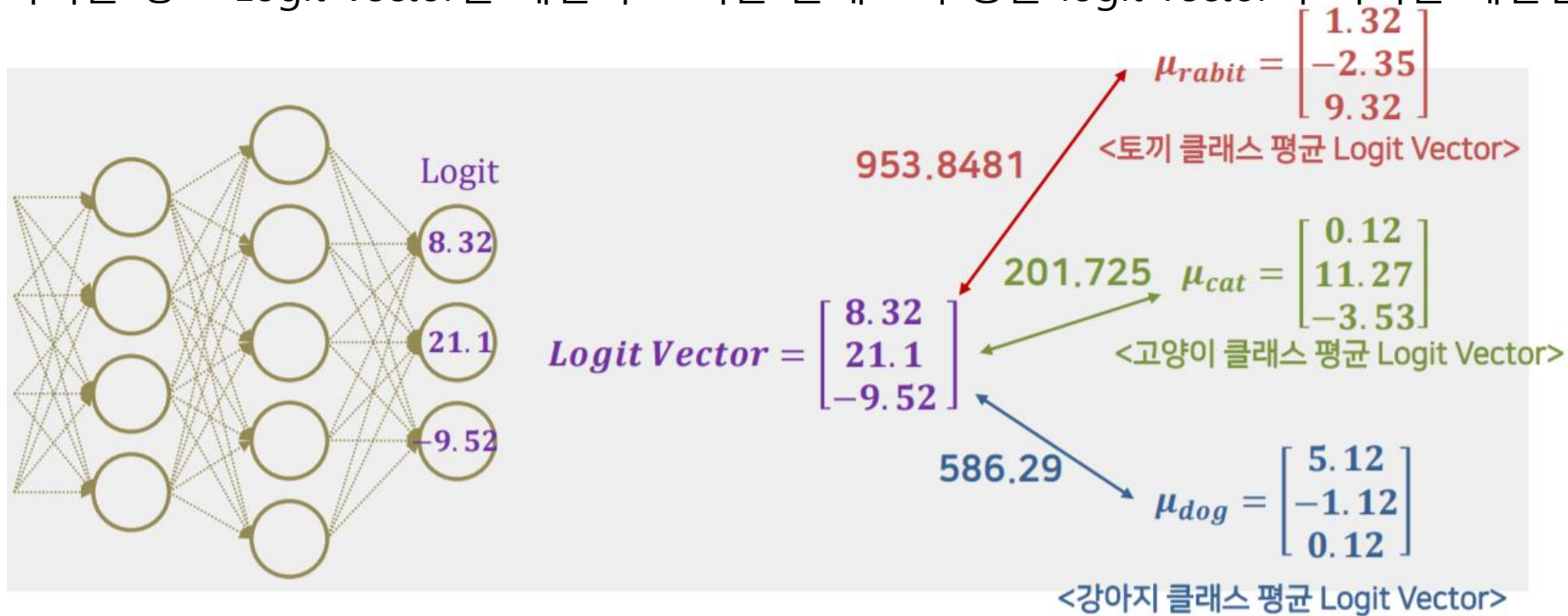
OpenMAX Flow

7. 각 클래스별로 거리가 가장 큰 n 개의 샘플로 **최대 가능도 추정**을 통해 **극단치 분포의 파라미터를 추정**한다.



OpenMAX Flow

8. 새로운 데이터를 넣고 Logit Vector를 계산하고 기존 클래스의 평균 logit vector와 거리를 계산한다

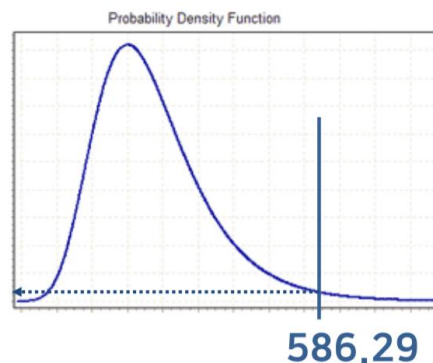


OpenMAX Flow

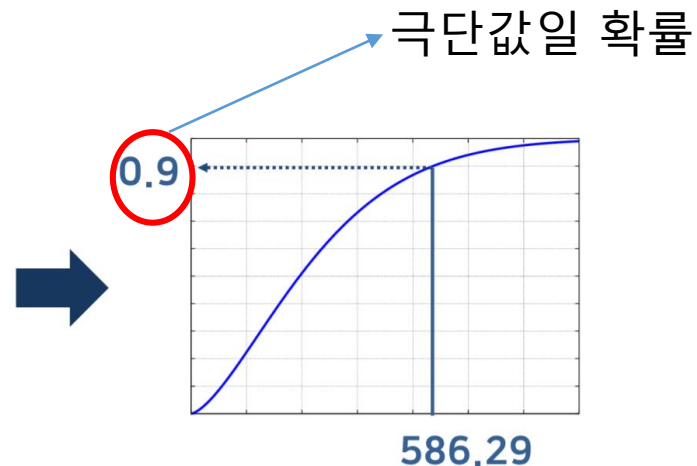
- 각 클래스 별 생성된 극단분포의 CDF(누적 분포)를 통해 평균 Logit Vector와의 거리 극단 확률 계산



클래스 별 극단값 분포도



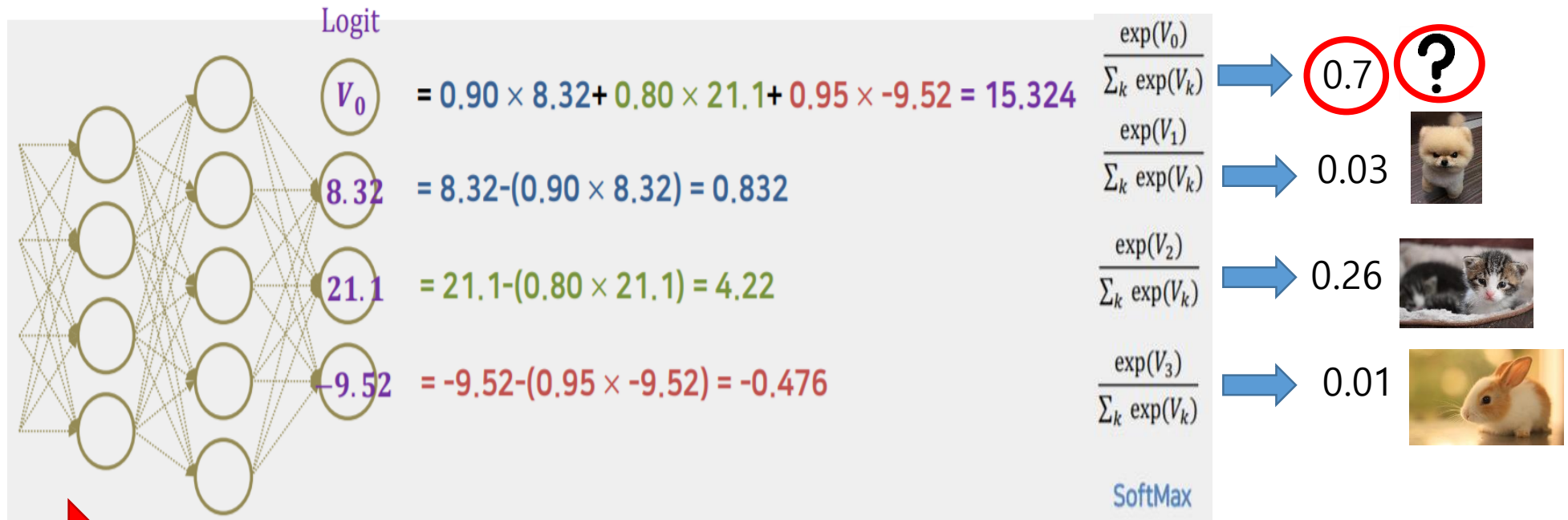
강아지 클래스의
확률 밀도 함수



강아지 클래스의
CDF

OpenMAX Flow

8. 극단 분포의 CDF값(극단값일 확률)을 w_k 로 두어 Logit Vector 업데이트



UnKnown으로 예측!!!