Can multi-label classification networks know what they don't know?

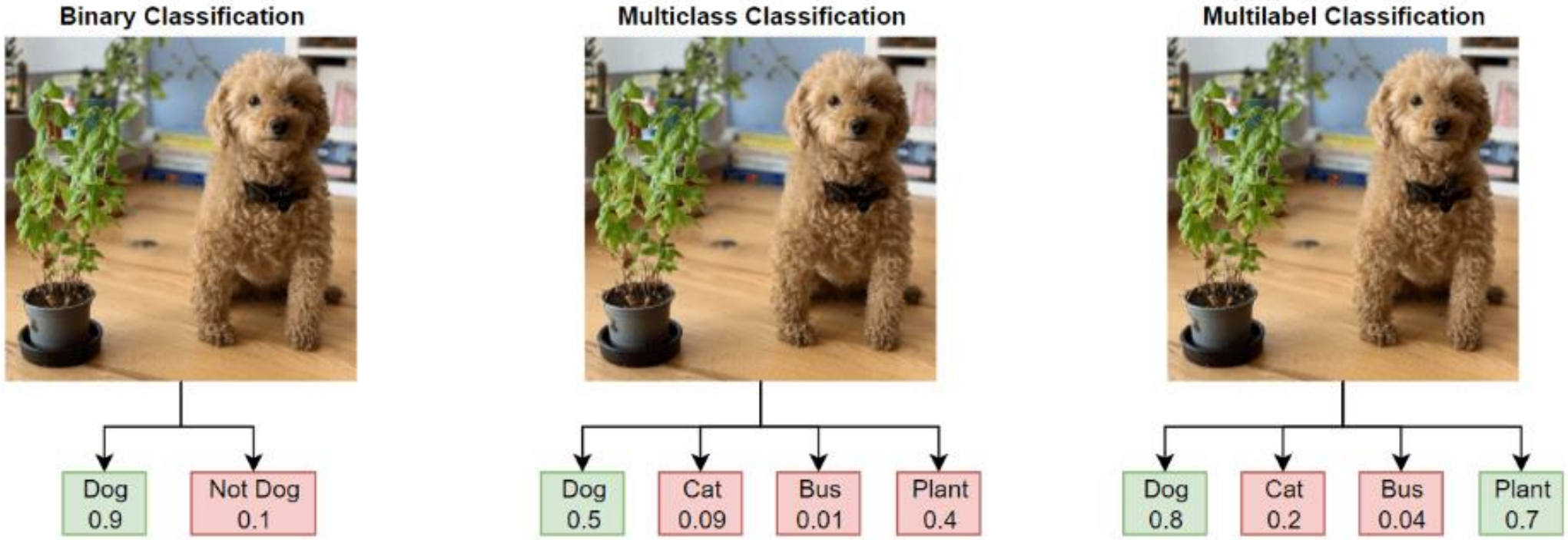
Haoran Wang, Weitang Liu, Alex Bocchieri, Yixuan Li

COLING 2022

박나현

Multi-label classification

- 입력 데이터가 여러 개의 label을 가지고 있어 이를 분류하는 문제



Introduction

- Previous studies
  - OOD sample을 multi-class classification 에서 찾는 것에 초점
    - 각각의 sample 은 single label
- Critical research gap
  - 실제 세계에서 multi-label classification에 대한 OOD detection을 발전시키고 평가하는 것에 대한 중요성 증가

Energy-based out-of-distribution detection

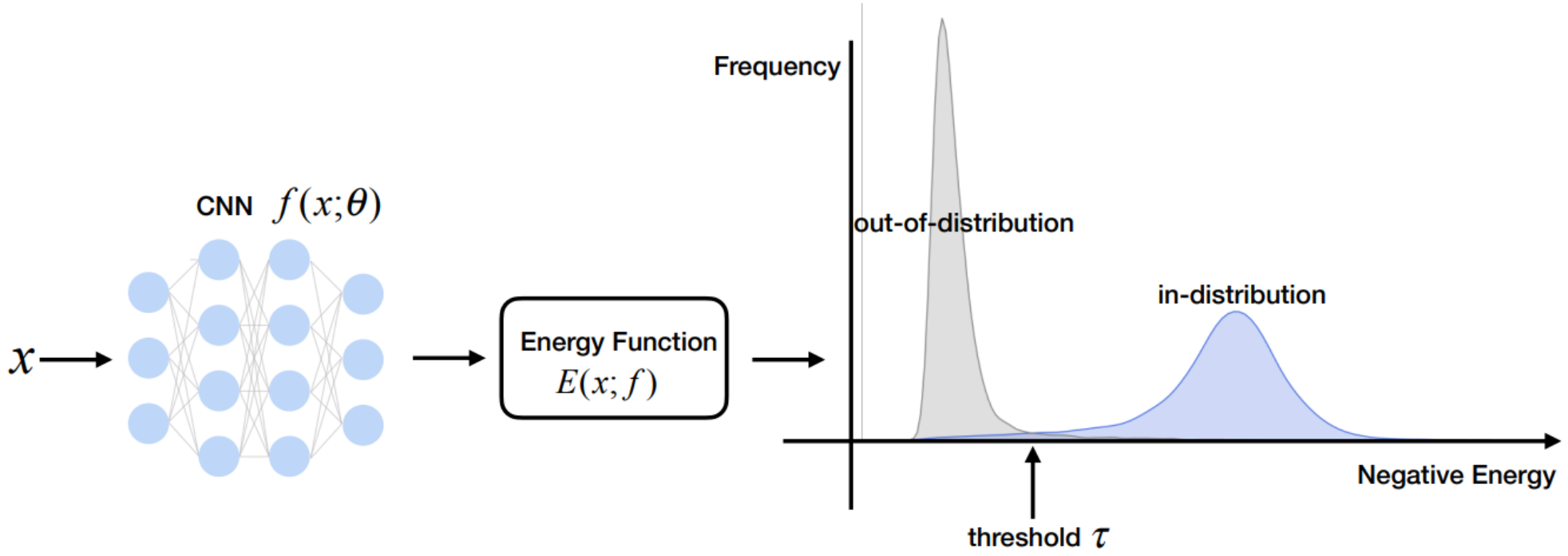


Figure 1: Energy-based out-of-distribution detection framework. The energy can be used as a scoring function for any pre-trained neural network (without re-training), or used as a trainable cost function to fine-tune the classification model. During inference time, for a given input $\mathbf{x}$, the energy score $E(\mathbf{x}; f)$ is calculated for a neural network $f(\mathbf{x})$. The OOD detector will classify the input as OOD if the negative energy score is smaller than the threshold value.

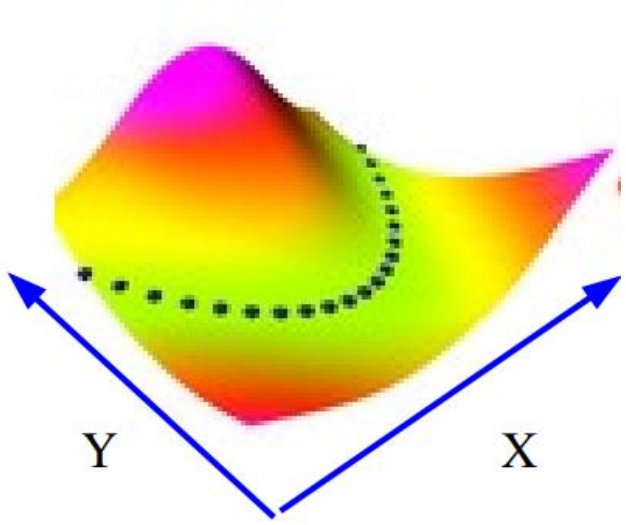
Liu, Weitang, et al. "Energy-based out-of-distribution detection." *Advances in neural information processing systems* 33 (2020): 21464-21475.

Energy

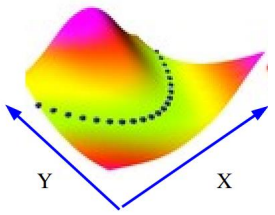
- **Energy-based model(EBM)**
  - **Observed data**에 대해 더 **낮은** single, non-probabilistic scalar mapping
  - **Unobserved data**에 대해 더 **높은** single, non-probabilistic scalar mapping

single, non-probabilistic scalar

= energy



Energy function



- Softmax
  - 단점
    - Training 때와 다른 data가 들어온 경우 임의로 확률 값 할당
    - overconfidence

$$p(y_i = 1 \mid \mathbf{x}) = \frac{e^{f_{y_i}(\mathbf{x})}}{\sum_{j=1}^{K} e^{f_{y_j}(\mathbf{x})}}$$

K real-valued numbers $f(\mathbf{x})$ Nerual classifier

- Energy function
  - Logits을 통한 확률 분포
  - Boltzmann distribution로 logit을 확률 분포로 변환
    - 해당 상태의 에너지와 온도의 함수로 특정 상태 에 있을 확률을 제공 하는 확률 분포 또는 확률 척도

$$p(y_i = 1 \mid \mathbf{x}) = \frac{e^{-E(\mathbf{x},y_i)}}{\int_{y'} e^{-E(\mathbf{x},y')}} = \frac{e^{-E(\mathbf{x},y_i)}}{e^{-E(\mathbf{x})}}$$

Multi-class classifier가 logit을 energy function 과 같다고 보며 Energy based 관점에서 해석 가능

$$E(\mathbf{x}, y_i) = -f_{y_i}(\mathbf{x})$$

Free energy function $E(\mathbf{x}) = -\log \sum_{i=1}^{K} e^{f_{y_i}(\mathbf{x})}$

Energy-based out-of-distribution detection

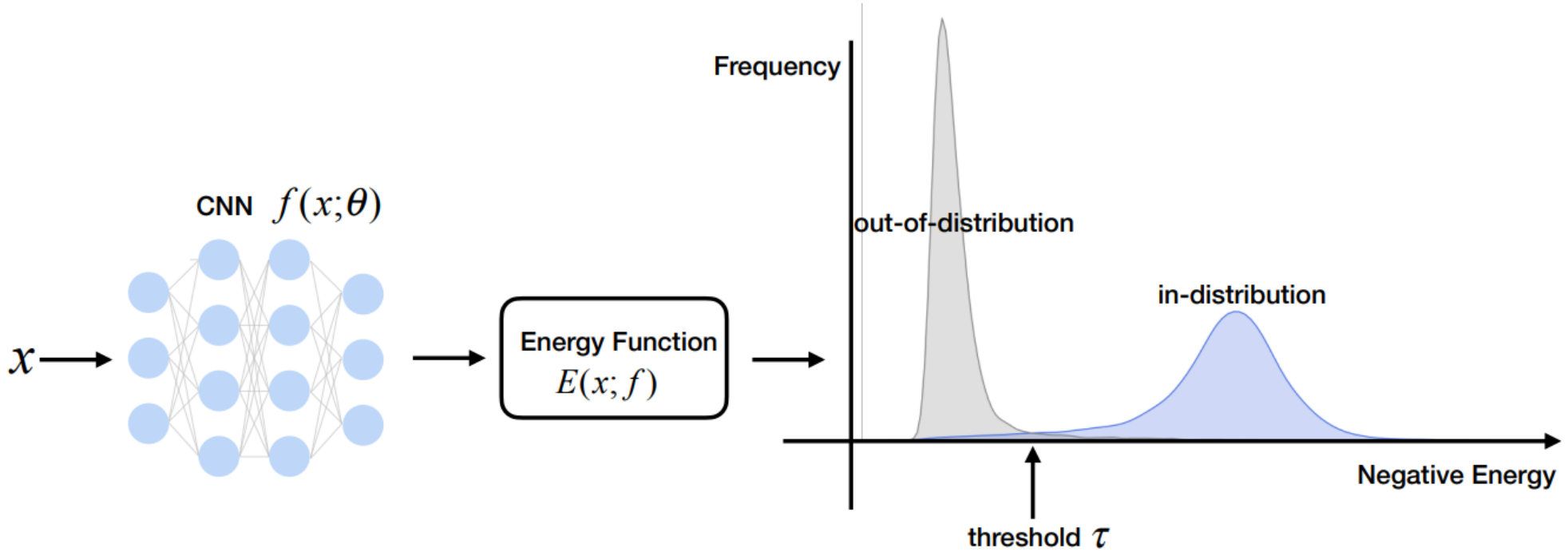
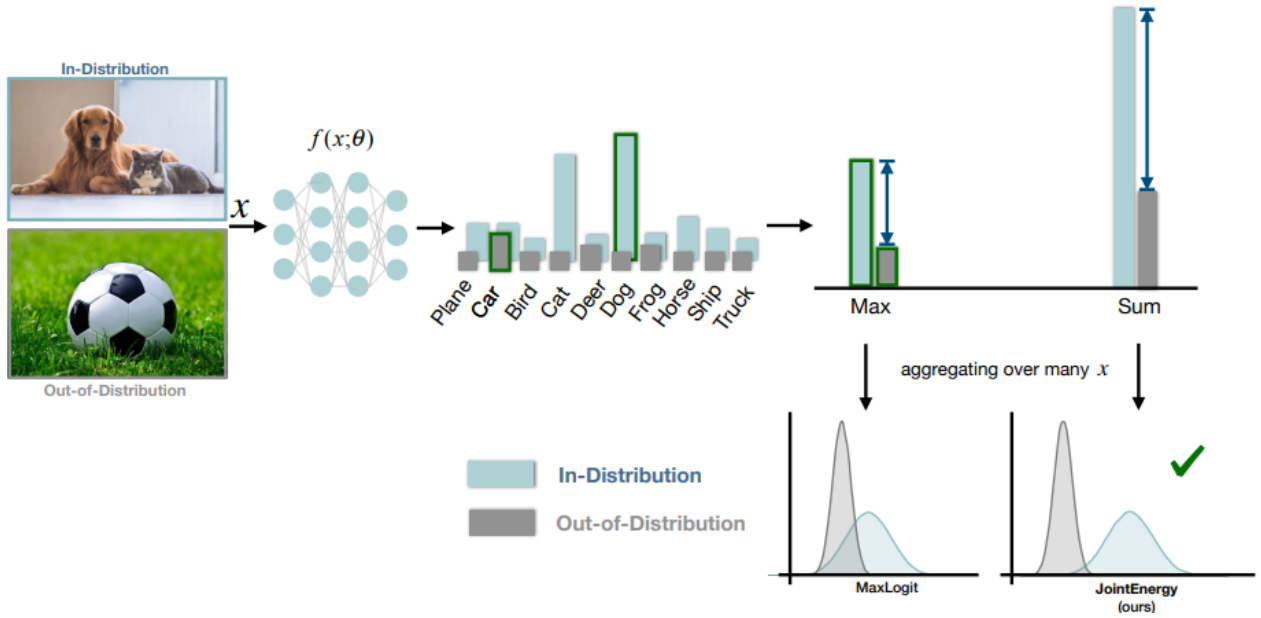


Figure 1: Energy-based out-of-distribution detection framework. The energy can be used as a scoring function for any pre-trained neural network (without re-training), or used as a trainable cost function to fine-tune the classification model. During inference time, for a given input $\mathbf{x}$, the energy score $E(\mathbf{x}; f)$ is calculated for a neural network $f(\mathbf{x})$. The OOD detector will classify the input as OOD if the negative energy score is smaller than the threshold value.

Liu, Weitang, et al. "Energy-based out-of-distribution detection." *Advances in neural information processing systems* 33 (2020): 21464-21475.

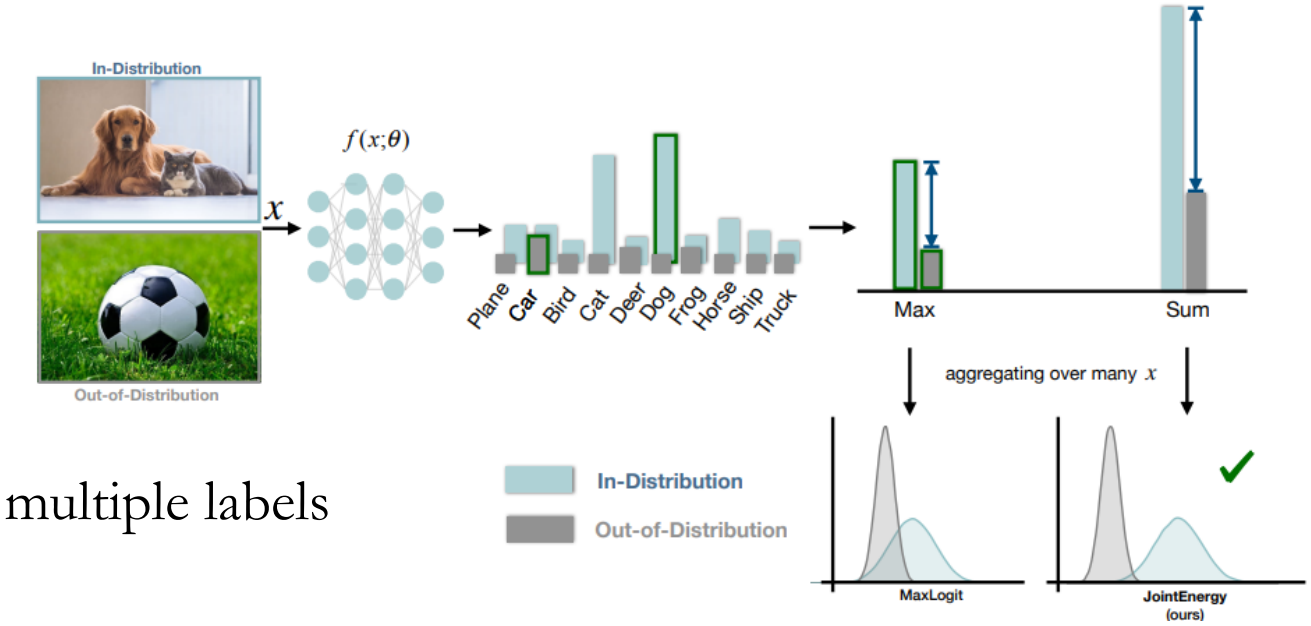
Main challenge

- Dominant한 label에 의존하는 것이 아니라 서로 다른 label을 함께 활용하여 uncertainty 를 평가할 필요성 존재



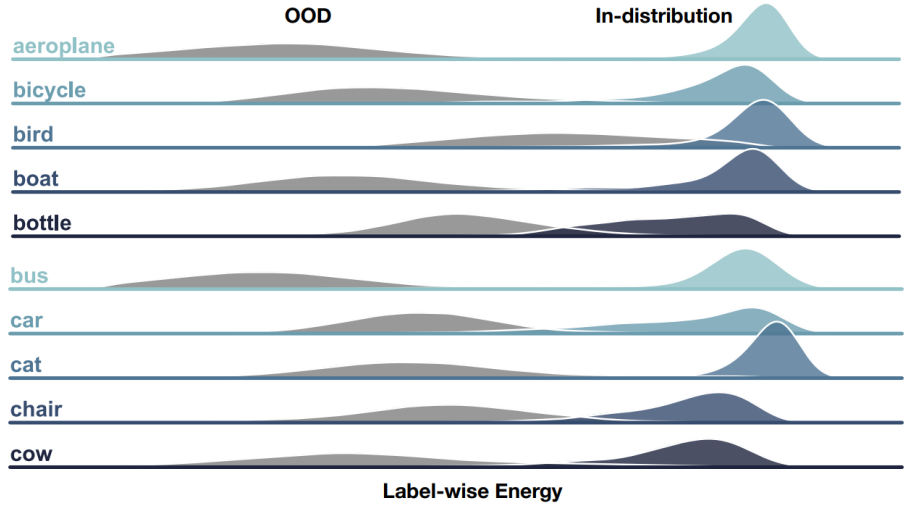
Maxlogit 사용할 경우,
Dog(IND)와 car(OOD)의 우세한 출력 차이만 캡쳐하고
Cat이라는 IND 정보 사라짐

제안한 방법



- **JointEnergy 제안**
  - Jointly characterize uncertainty from multiple labels
  - 장점
    - 생성모델을 사용하여 계산 복잡한 joint likelihood를 직접 계산하지 않음
      - Joint likelihood
        - IND와 OOD data 분리하여 따르게 됨
        - OOD data(즉, 더 높은 energy 가진 data)는 더 낮은 likelihood를 따른다고 예상하기 때문
    - 전체 label에 대한 label-wise energy 를 합쳐 새로운 OOD score 제안
    - Maxlogit 에 비해 IND와 OOD 사이의 점수차이를 효과적으로 증폭

Method



- **Label-wise Free Energy**
  - 기본 pre-trained multi-label neural classifier

$$f_{y_i}(\mathbf{x}) = h(\mathbf{x};\theta) \cdot \mathbf{w}^i_{\text{cls}},$$

Class i에 대한 weight

Feature vector in the penultimate layer

  - 각 binary label에 대한 예측 확률
    - Binary logistic classifier 로 생성 (OOD 인지 In-distribution 인지)
      - 두개의 logit(0, $f_{y_i}(\mathbf{x})$)을 가진 softmax로써 보여짐

$$p(y_i = 1 \mid \mathbf{x}) = \frac{e^{f_{y_i}(\mathbf{x})}}{1 + e^{f_{y_i}(\mathbf{x})}},$$

  - **Label-wise free energy**
    - Single label에 대한 OOD 불확실성 포착하지만 label 간 불확실성 포착 불가

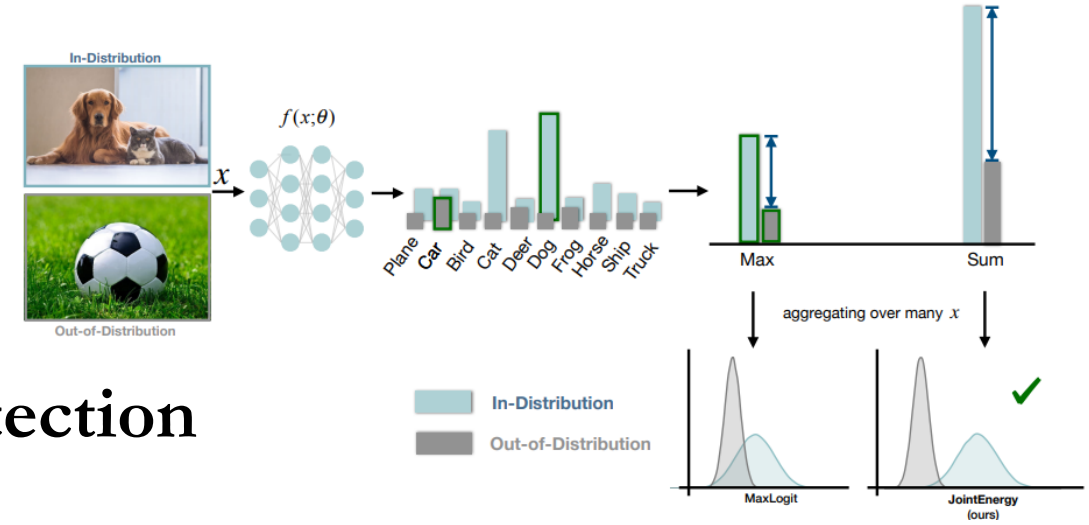
$$E_{y_i}(\mathbf{x}) = -\log(1 + e^{f_{y_i}(\mathbf{x})}),$$

Method

- **JointEnergy**
  - 새로운 novel scoring function
  - Label 간의 불확실성 고려
  - Label 간의 OOD 불확실성의 joint estimation을 고려한 첫번째 방법
  - 값의 음수이므로 점수가 클수록 분포내에 있음을 나타내는 기존 방법과 일치

$$E_{\text{joint}}(\mathbf{x}) = \sum_{i=1}^{K} -E_{y_i}(\mathbf{x})$$

Method



- **JointEnergy for multi-label OOD detection**
  - Energy threshold τ
    - In-distribution data 의 높은 비율이 $G(\mathbf{x};\tau)$ 에 의해 올바르게 분류하도록 선택
    - High fraction(95%) of in-distribution data 에서 선택됨

$$G(\mathbf{x};\tau) = \begin{cases} \text{out} & \text{if } E_{\text{joint}}(\mathbf{x}) \leq \tau, \\ \text{in} & \text{if } E_{\text{joint}}(\mathbf{x}) > \tau, \end{cases}$$

Experiments

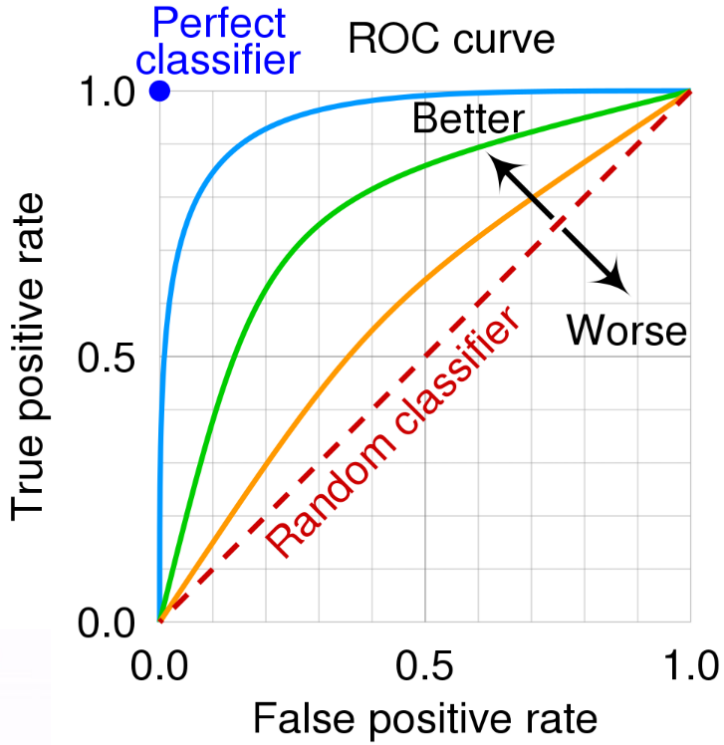
- **In-distribution Dataset (multi-label dataset)**
  - MS-COCO
    - 80 common object categories
    - 82,783/40,504/40,775
    - 고양이, 강아지, 차, 비행기, 사람…
  - PASCAL-VOC
    - 20개 class
    - 22,531
    - 자동차, 새, 고양이, 강아지, 사람…
  - NUS-WIDE
    - 119,986/80,283
    - 바다, 숲, 아기, 커피…

- **Out-of-distribution Dataset**
  - ImageNet
    - 20개 class
      - Multi-label classifier 가 이미 imagenet 으로 학습되어 있어 이와 겹치지 않는 class만 선택
      - ImageNet-22K에서 선택

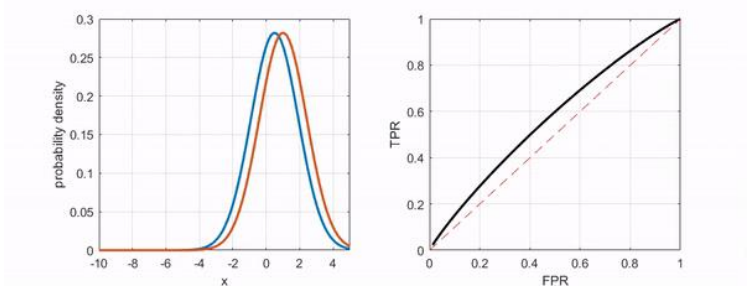
- **Training Details**
  - Densenet-121 + 2개 FC layer
  - Random crop 과 random flip 이용하여 color image 획득
    - Data augmentation

Evaluation metrics

- **ROC (Receiver Operating Characteristic)**
  - X 축: FPR(False Positive Rate)
  - Y 축: TRP(True Positive Rate)



Index	1	2	3	4	5	6	7	8	9	10
Actual	0	0	0	0	0	1	1	1	1	1
Predict	0.62	0.41	0.22	0.12	0.07	0.33	0.85	0.59	0.91	0.39



Threshold =0.1

Actual \ Predict	1	0
1	5	0
0	4	1

Threshold =0.2

Actual \ Predict	1	0
1	5	0
0	3	2

Threshold =0.3

Actual \ Predict	1	0
1	5	0
0	2	3

Threshold =0.4

Actual \ Predict	1	0
1	3	2
0	2	3

Threshold =0.5

Actual \ Predict	1	0
1	3	2
0	1	4

Threshold =0.6

Actual \ Predict	1	0
1	2	3
0	1	4

Threshold =0.7

Actual \ Predict	1	0
1	2	3
0	0	5

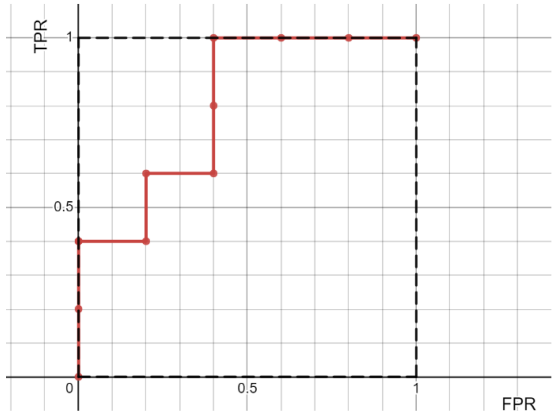
Threshold =0.8

Actual \ Predict	1	0
1	2	3
0	0	5

Threshold =0.9

Actual \ Predict	1	0
1	1	4
0	0	5

Threshold	FPR (x 축)	TPR (y 축)
0.1	0.8	1.0
0.2	0.6	1.0
0.3	0.4	1.0
0.4	0.4	0.6
0.5	0.2	0.6
0.6	0.2	0.4
0.7	0.0	0.4
0.8	0.0	0.4
0.9	0.0	0.2



$$TPR = \frac{TP}{TP + FN}$$

recall

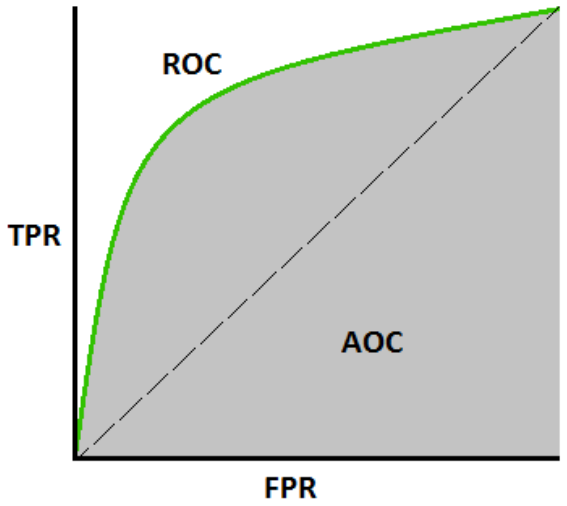
Actual \ Predict	1	0
1	TP	FN
0	FP	TN

$$FPR = \frac{FP}{FP + TN}$$

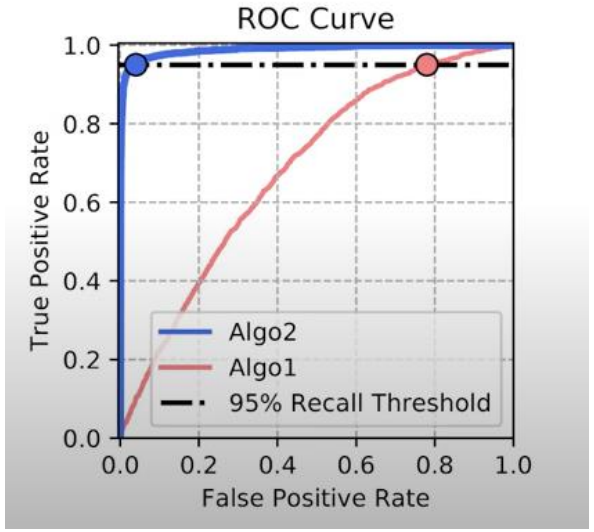
Actual \ Predict	1	0
1	TP	FN
0	FP	TN

Evaluation Metrics

- **AUROC(Area Under ROC Curve)**
  - AUC-ROC(Area Under Curve of ROC Curve)
  - ROC curve의 넓이



- **FPR95**
  - False positive rate at a 95% recall(True Positive Rate) threshold

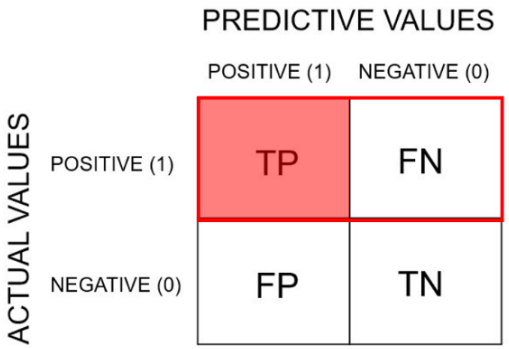


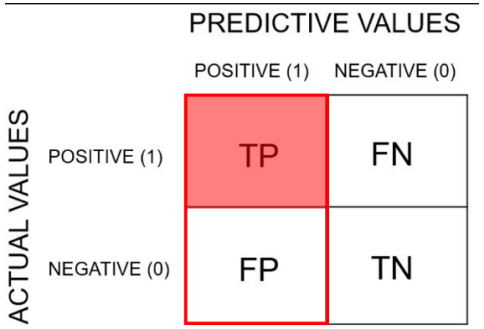
Evaluation Metrics

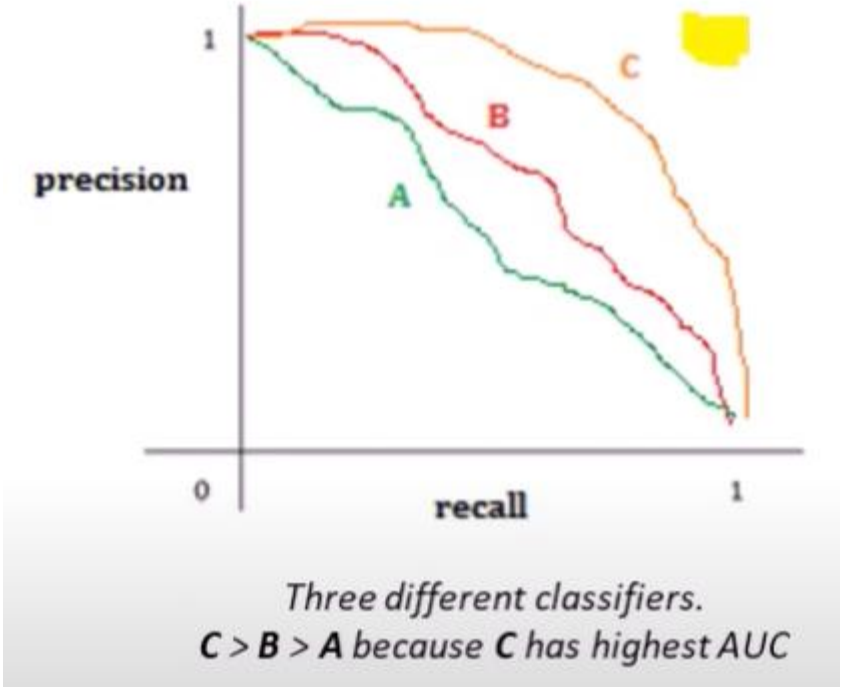
- AUPRC(area under the precision-recall curve)
  - X 축: Recall
  - Y 축: Precision

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$







Results

$\mathcal{D}_{\text{in}}$	MS-COCO	PASCAL-VOC	NUS-WIDE
		FPR95 / AUROC / AUPR	
OOD Score		↓ ↑ ↑	
MaxLogit [15]	43.53 / 89.11 / 93.74	45.06 / 89.22 / 83.14	56.46 / 83.58 / 94.32
MSP [16]	79.90 / 73.70 / 85.37	74.05 / 79.32 / 72.54	88.50 / 60.81 / 87.00
ODIN [28]	43.53 / 89.11 / 93.74	45.06 / 89.22 / 83.16	56.46 / 83.58 / 94.32
Mahalanobis [27]	46.86 / 88.59 / 93.85	41.74 / 88.65 / 81.12	62.67 / 84.02 / 95.25
LOF [3]	80.44 / 73.95 / 86.01	86.34 / 69.21 / 58.93	85.21 / 67.75 / 89.61
Isolation Forest [31]	94.39 / 49.04 / 66.87	93.22 / 50.67 / 35.78	95.69 / 53.12 / 83.32
JointEnergy	**33.48 / 92.70 / 96.25**	**41.01 / 91.10 / 86.33**	**48.98 / 88.30 / 96.40**

- **How does JointEnergy compare to common OOD detection methods?**
  - Energy based를 OOD detection method in literature 와 비교
  - Model
    - 대부분 baseline(maxlogit, msp, odin, mahalanobis): 최대값 기반으로 OOD 점수 계산
    - LOF: knn 사용하여 local densitiy 측정
      - OOD는 local 에 비해 낮은 밀도
    - Isolation Forest: root node에서 terminating node까지 path 길이로 이상치 탐지
      - Tree based approach

Results

- **How do different aggregation methods affect OOD detection performance?**
  - Label-wise energy score를 합치는 다른 방법과 비교
    - 최대값의 label만을 고려하는 것이 아니라 모든 label의 정보를 고려하는 것의 중요성 강조

$$E_{y_i}(\mathbf{x}) = -\log(1 + e^{f_{y_i}(\mathbf{x})}),$$

$$E_{\text{max}}(\mathbf{x}) = \max_i -E_{y_i}(\mathbf{x}),$$

auroc

$\mathcal{D}_{\text{in}}$	MaxEnergy	JointEnergy
MS-COCO	89.11	**92.70**
PASCAL-VOC	89.22	**91.10**
NUS-WIDE	83.58	**88.30**

Results

- **What is the effect of applying the aggregation method to prior methods?**
  - 이전 scoring function 에 aggregation 방법을 적용하는 효과 확인
    - Max를 sum으로 대신하여 실험
    - 각 label 에 대해 score 계산 후 label 별 점수를 OOD 전체 점수로 합산
  - Label 전체 logit을 단순히 합하면 양수와 음수의 혼합으로 성능 저하

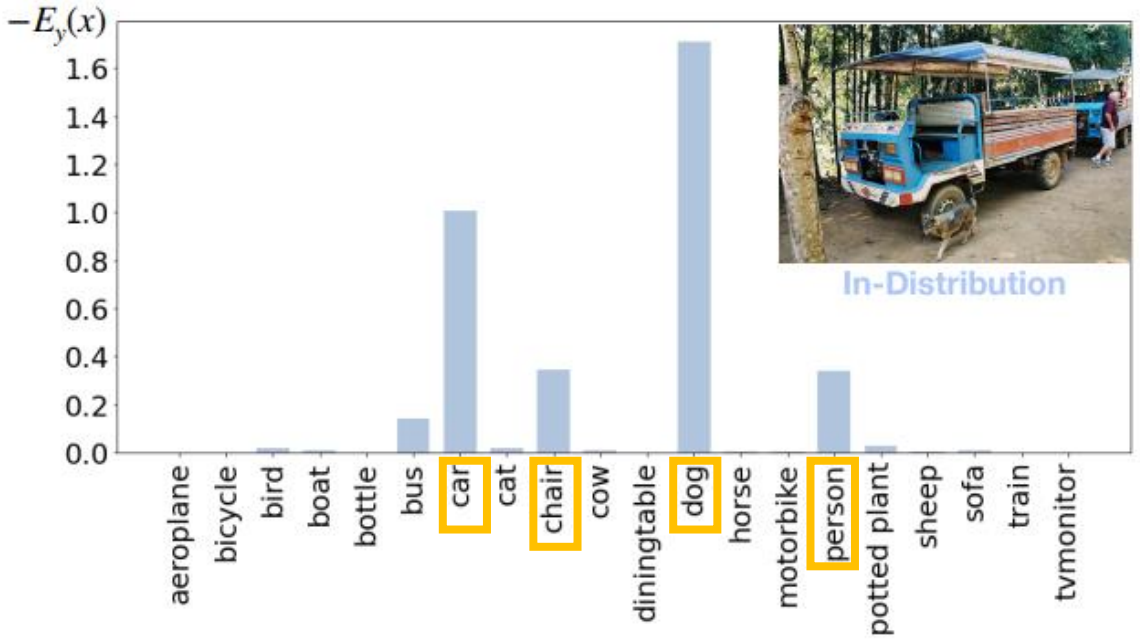
	$\mathcal{D}_{\text{in}}$	MS-COCO	PASCAL	NUS-WIDE
		FPR95 / AUROC / AUPR		
OOD Score	**Aggregation**	↓ ↑ ↑		
Logit	Sum	95.46 / 61.81 / 80.39	87.18 / 72.68 / 61.24	96.53 / 51.75 / 82.55
Prob	Sum	45.04 / 89.32 / 94.40	**38.57** / 86.53 / 79.10	50.84 / 83.82 / 95.15
ODIN	Sum	56.56 / 84.62 / 92.24	50.35 / 79.45 / 70.19	56.26 / 81.04 / 94.34
Mahalanobis	Sum	53.43 / 87.52 / 93.35	44.43 / 87.76 / 79.86	69.05 / 80.46 / 94.09
LOF	Sum	N/A	N/A	N/A
Isolation Forest	Sum	N/A	N/A	N/A
JointEnergy (ours)	Sum	**33.48 / 92.70 / 96.25**	41.01 / **91.10 / 86.33**	**48.98 / 88.30 / 96.40**

43.53%

적절한 scoring function 고르는 것이 더 중요

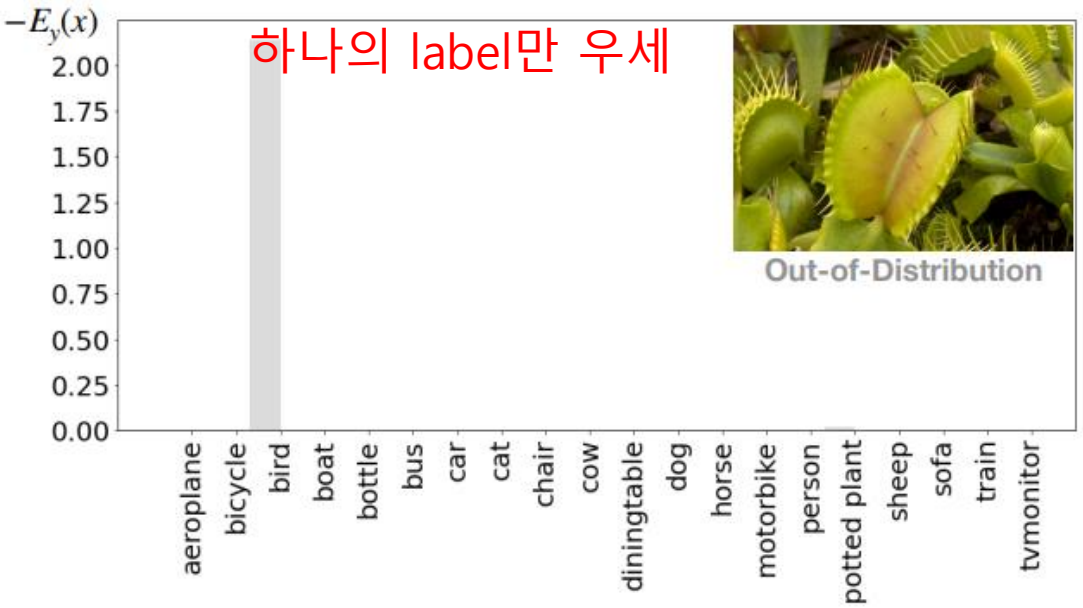
Results

- Qualitative Case Study
  - Jointenergy로 학습한 결과



MaxLogit socre: 1.63

Jointenergy score: 3.23



MaxLogit socre: 2.14

Jointenergy score: 2.19

Conclusion and Outlook

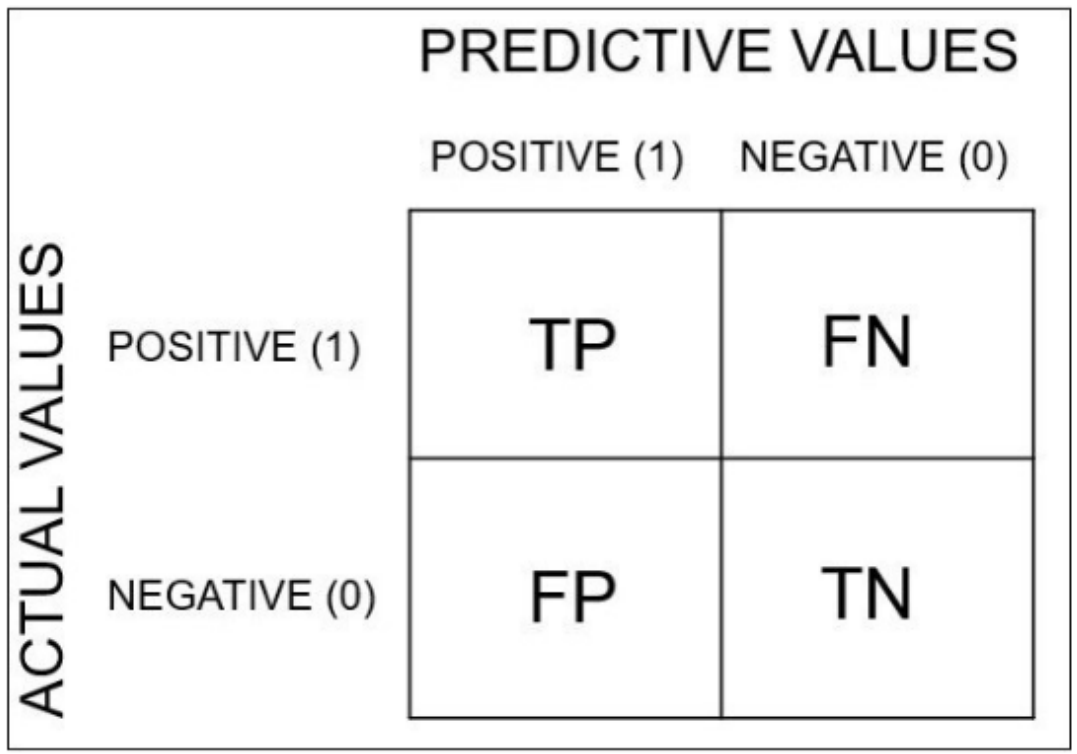
- Multi-label 분류에서 OOD 불확실성 추정에 대한 에너지 점수 제안
  - 하나의 label 정보만을 이용하는 것보다 모든 label에 대한 jointenergy를 사용하는 것이 in-distribution 과 OOD input를 구별하는 데 더 효과적
  - 수학적으로 입증

Thank you

1. Confusion Matrix (오차행렬)

위 네가지 지표를 설명하기 전에 Confusion Matrix를 먼저 설명하고자 한다.

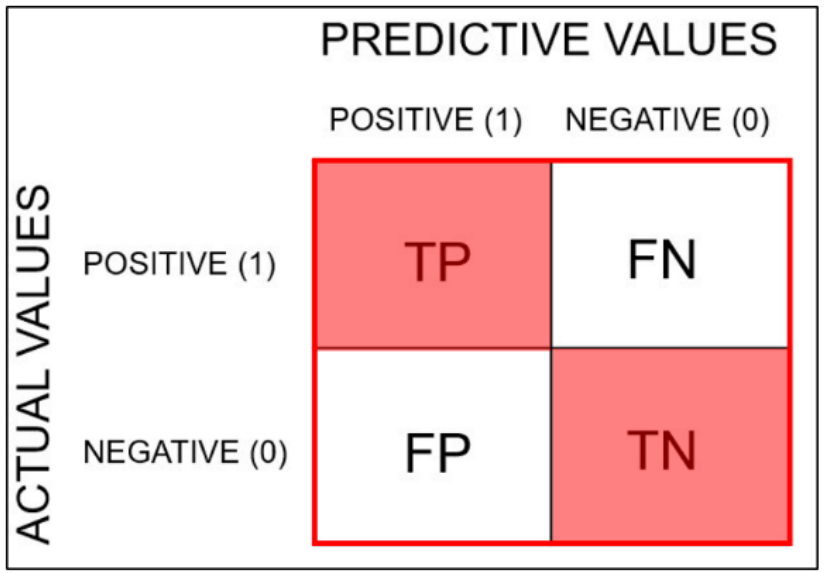
> **Confusion Matrix**란? Training 을 통한 Prediction 성능을 측정하기 위해 예측 value와 실제 value를 비교하기 위한 표



<Confusion Matrix>

2. Accuracy (정확도)

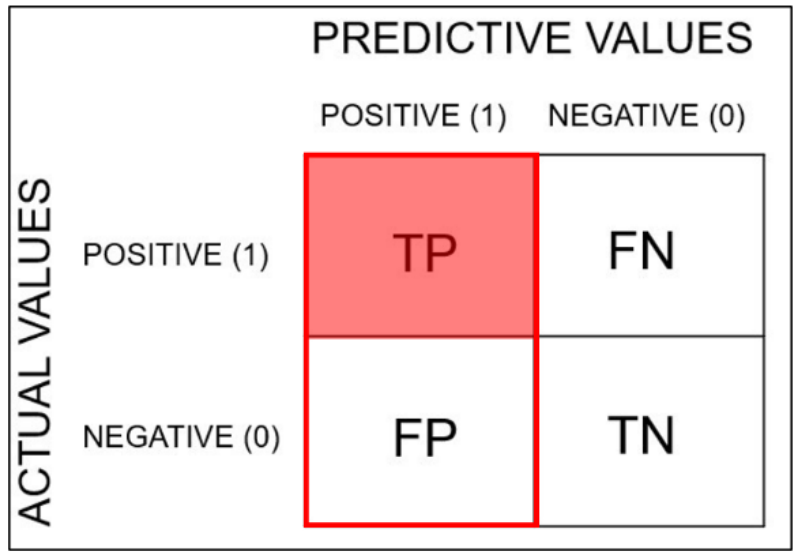
Accuracy(정확도)란? 전체 중 모델이 바르게 분류한 비율



<정확도>

3. Precision (정밀도)

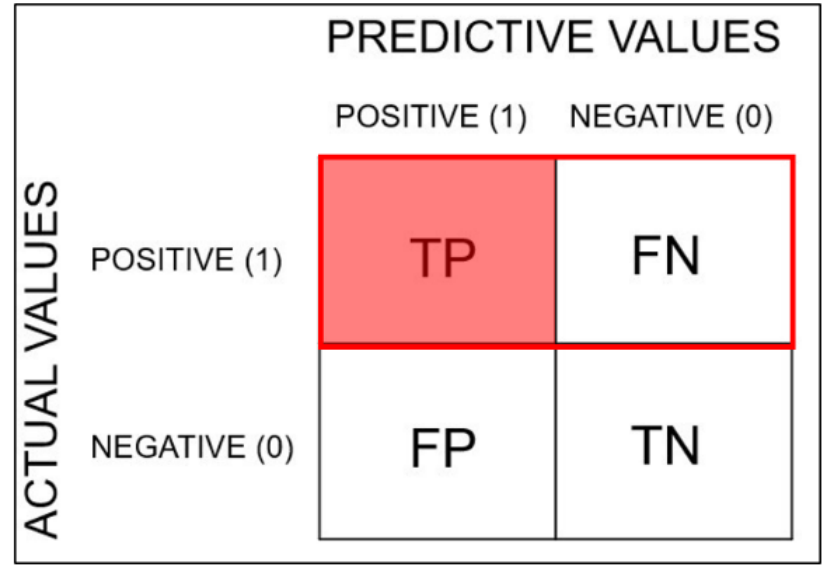
Precision(정밀도)란? 모델이 Positive라 분류한 것 중 실제값이 Positive인 비율



<정밀도>

4. Recall (재현도)

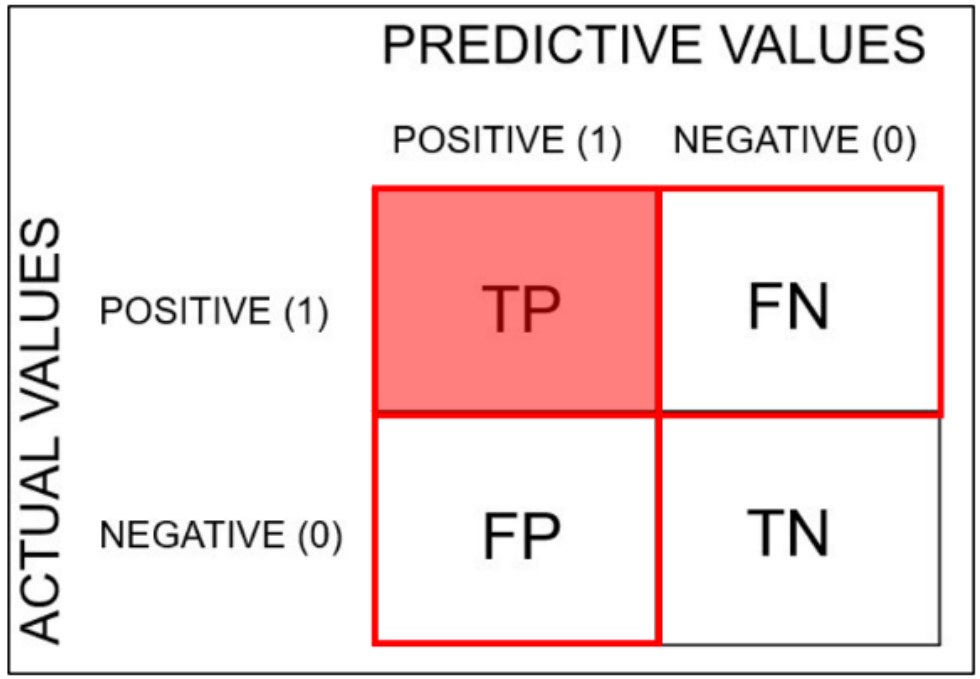
Recall(재현도)란? 실제값이 Positive인 것 중 모델이 Positive라 분류한 비율



<재현도>

5. F1 Score

F1 Score란? Precision과 Recall의 조화평균



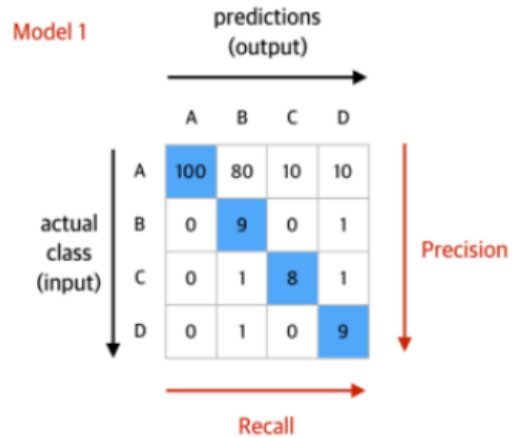
<F1-Score>

*** F1-Score**

$$F1-Score = \frac{2 * Presicion * Recall}{Precision + Recall}$$

* 다중 Class 에서 F1 Score 구하는 방법

각 Class에 대한 Precision과 Recall을 구한 후, 평균값으로 F1 Score 계산



Accuracy = (100+9+8+9) / 230 = 0.547

- Precision A =100 / (100+0) = 1
- Precision B = 9 / (9+81) = 9/91
- Precisian C = 8 / (8+10) = 8/18
- Precision D = 9 / (9+12) = 9/21

average precision = 0.492

- recall A =100 / (100+100) = 100/200
- recall B = 9 / (9+1) = 9/10
- recall C = 8 / (8+2) = 8/10
- recall D = 9 / (9+1) = 9/10

average recall = 0.775

F1 Score = 2 * ((0.492*0.775) / (0.492+0.775)) = 0.601

Accuracy = (198+1+1+1) / 230 = 0.87

- Precision A =198 / (198+9) = 198/207
- Precisian B = 1 / (1+13) = 1/14
- Precisian C = 1 / (1+4) = 1/5
- Precision D = 1 / (1+3) = 1/4

average precision = 0.369

- recall A =198 / (198+2) = 198/200
- recall B = 1 / (1+9) = 1/10
- recall C = 1 / (1+9) = 1/10
- recall D = 1 / (1+9) = 1/10

average precision = 0.323

F1 Score = 2 * ((0.369*0.323)/(0.369+0.323)) = 0.344

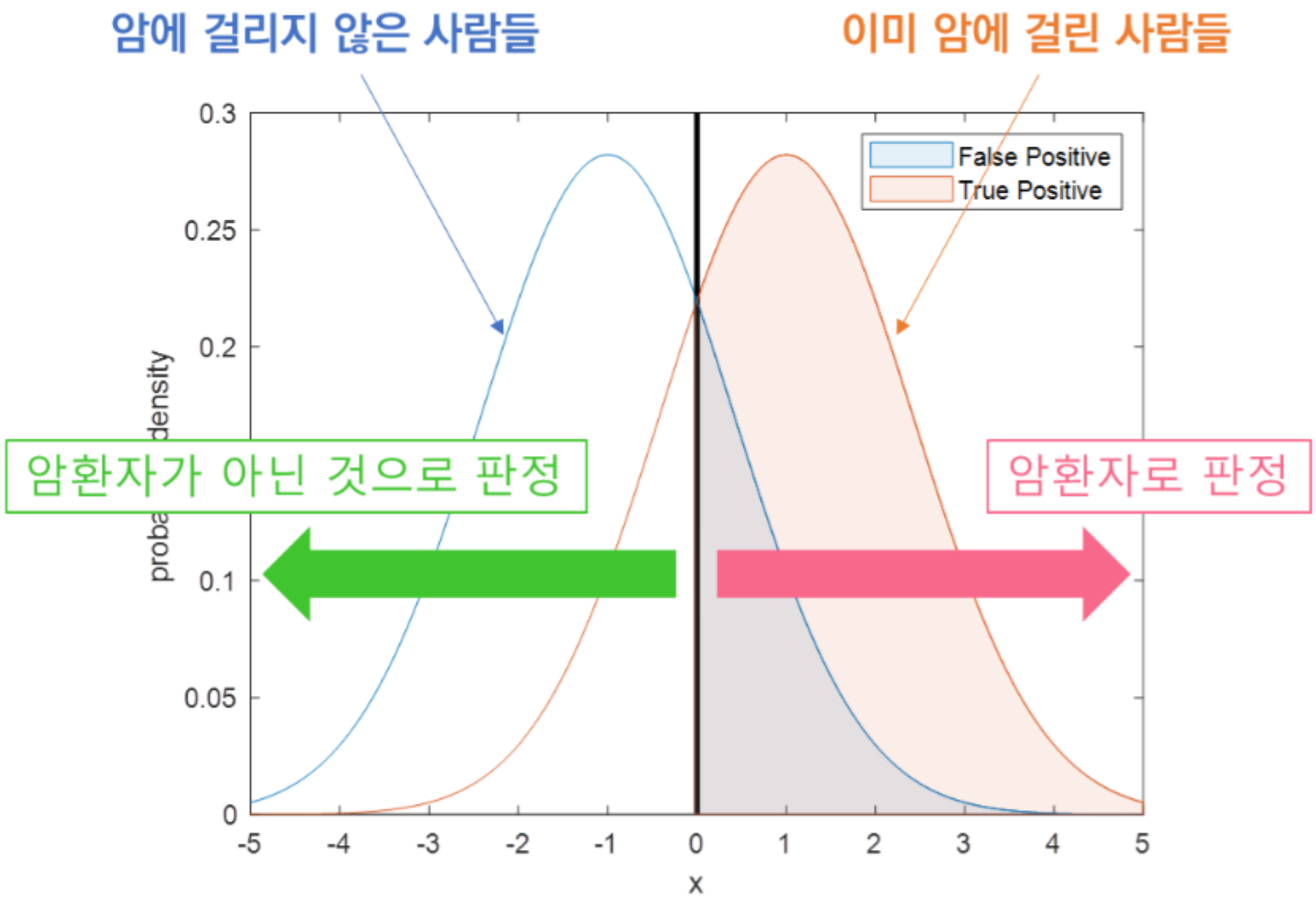


그림 3. True Positive와 False Positive를 그림으로 나타낸 것.