03_XAI Methods 2: Global Explanation Methods

1. What is the Global Explanation Method?

Global Explanation Method

global의 의미는 모델의 평균적인 행동을 설명하고자 하는 방법이다. local은 개별 input으로 살펴보았다면, global은 dataset 전반에 걸쳐서 어떤 feature가 어떤 prediction을 내리는데 어떠한 영향을 미치는지를 살펴보는 것이다.

2. Model-agnostic Method

Model-agnostic

모델의 종류에 구애 받지 않고 다양한 모델에 적용될 수 있는 성질을 의미한다. 이는 2가지 특징을 지니고 있다.

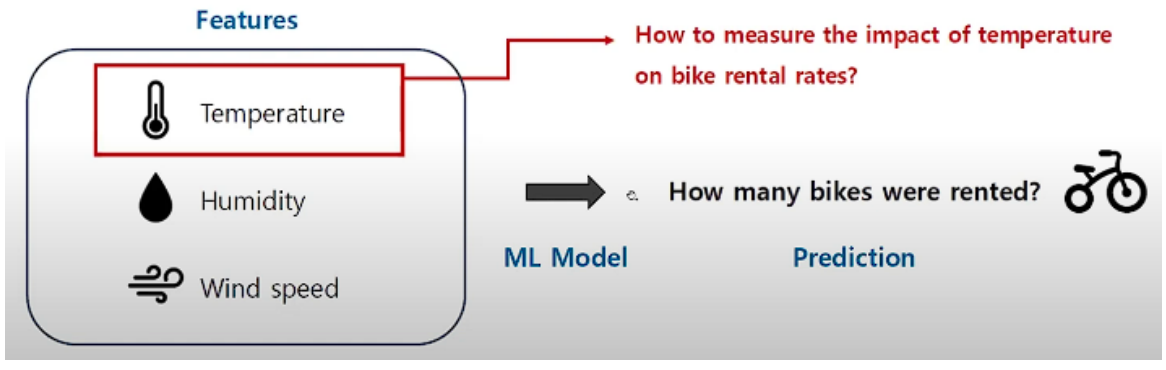
1. Flexibility : 다양한 상황과 모델 타입에 대해서 적용될 수 있는 점
2. Comparability : 각기 다른 모델들을 비교하는데 사용이 가능하다는 점

2-1. Partial Dependence Plot (PDP)

target feature와 모델의 예측값 간의 관계를 확인하는데 사용되는 기법이다.

예를 들어 살펴보자. 자전거가 얼마나 대여됐는지를 예측하는 ML 모델이 있다고 가정하자.

여기서 feature들은 기온, 습도, 풍속 등이 존재하는데, 아래와 같이 기온을 target feature로 설정하고 기온이 모델의 예측에 얼마만큼 기여했는지 marginalize하는 것이다.



PDP Method

수식을 살펴보자.

- $\hat{f}$: 머신러닝 모델
- x_s : target feature, X_c : 다른 피처들

머신러닝 모델을 설명하기 위해서 오로지 target feature에만 dependency를 갖도록

수식을 구성한다.

1. **Marginalize over other features**
   - 관심 있는 변수 x_s와 모델 출력값 간의 관계만 남기기 위해 다른 변수 X_c의 영향을 제거(평균화)하는 과정이다.
   - 관심 있는 변수가 모델 예측값에 미치는 영향을 분석하기 위해서 나머지 변수들의 영향을 줄여야 하고, 이를 위해서 그 변수들에 대한 평균을 구해야 하는 것이다.
     - 특정 변수 x_s가 바뀔 때, 나머지 변수 X_c가 어떤 값이든 **전체 데이터 분포에서 평균을 내서 x_s의 "순수한" 영향을 분석**한다!!
2. **실제 데이터 분포 $P(X_c)$를 알 수 없다.**
   - 이론적으로는 $P(X_c)$를 정확히 알아야 하지만, 실제로는 알 수 없으므로 데이터 기반으로 근사하여 계산한다.

$$\hat{f}_S(x_S) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

- $\hat{f}$: Machine learning model
- x_S: Target feature, X_C: Other features

⇨ **Marginalize over the other variables to retain only the dependency on** x_S (temperature).

⇨ However, we do not know the exact distribution of X_C.

- 실제로 적분을 계산하기 어려우므로, 머신러닝에서는 $P(X_C)$의 분포를 데이터 샘플로 근사하여 계산합니다.
- 기댓값을 근사적으로 계산하는 방식:

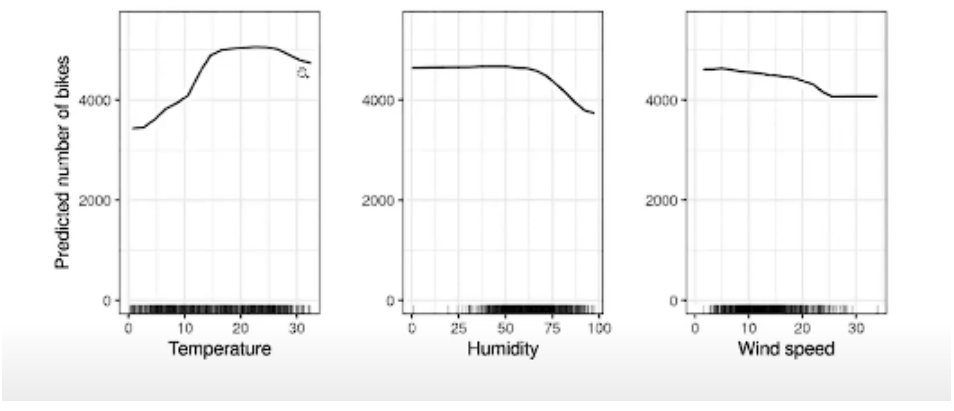
$$\hat{f}_S(x_S) \approx \frac{1}{n}\sum_{i=1}^{n} f(x_S, X_C^{(i)})$$

여기서:

- n: 샘플 수.
- $X_C^{(i)}$: 데이터 샘플 i에서의 다른 변수 값.

Results

- 장점 : 직관적으로 이해하기 쉽다, 구현하기 쉽다, target feature와의 최종 예측의 관계를 파악할 수 있다.
- 단점 : 나머지 feature들은 marginalize를 하기 때문에, target feature와의 다른 feature들의 상관관계에 대해서는 고려하기가 힘들다. 예를 들어, 현실에서는 온도와 습도가 강한 상관관계를 가질 수 있다. 그러나 PDP에서는 다른 변수들의 값을 평균화하므로 비현실적인 조합이 발생할 수 있는 것이다.



2-2. Accumulated Local Effect (ALE)

PDP의 단점을 보완하기 위해 고안된 기법으로, 특히 변수들 간 상관관계가 있을 때 더 적합한 해석 방법을 제공한다.

- **데이터 구간화(interval)**
  - 관심 변수 x_s의 값을 여러 개의 구간으로 나눈다.
- **각 구간 내에서 예측값 변화 계산**
  - 각 구간에서 변수 값이 x_s에서 $x_s + \Delta x$로 증가할 때 모델의 예측값 차이를 계산.
    - $f(x_s + \Delta x, X_c) - f(x_s, X_c)$
  - 즉, x_s가 조금 증가할 때 모델의 예측값이 얼마나 변하는지를 측정.
- **나머지 변수 X_c에 대해 평균 계산**
  - 각 x_s 구간에서, 나머지 변수 X_c를 고려한 여러 샘플의 차이값을 평균을 계산.
  - 이 과정에서 데이터 분포 내에서만 값이 사용되므로, 비현실적인 데이터 조합이 발생하지 않음.
- **변화량을 누적하여 ALE 계산**
  - ALE는 변화량을 누적하여 전체적인 트렌드를 형성한다.

ALE의 정의:

$$\hat{f}_{S,ALE}(x_S) = \int_{x_{S,\min}}^{x_S} \mathbb{E}_{X_C|X_S=z_S}\left[\frac{\partial \hat{f}(x_S, X_C)}{\partial x_S}\right] dz_S$$

1. $\hat{f}_{S,ALE}(x_S)$:
   - 변수 x_S에 대한 ALE 값.
   - 모델의 예측값이 x_S의 변화에 따라 얼마나 달라지는지를 누적한 결과.
2. $\mathbb{E}_{X_C|X_S=z_S}[\cdot]$:
   - 조건부 기댓값. 즉, 특정 x_S 값에서 나머지 변수 X_C의 영향을 평균화하는 과정.
3. **적분**:
   - x_S가 최소값부터 시작해 x_S 값까지, 각 구간에서 예측값 변화량을 누적.

ALE의 근사 계산 (Discrete Approximation):

데이터 구간화와 누적을 이용한 ALE 계산:

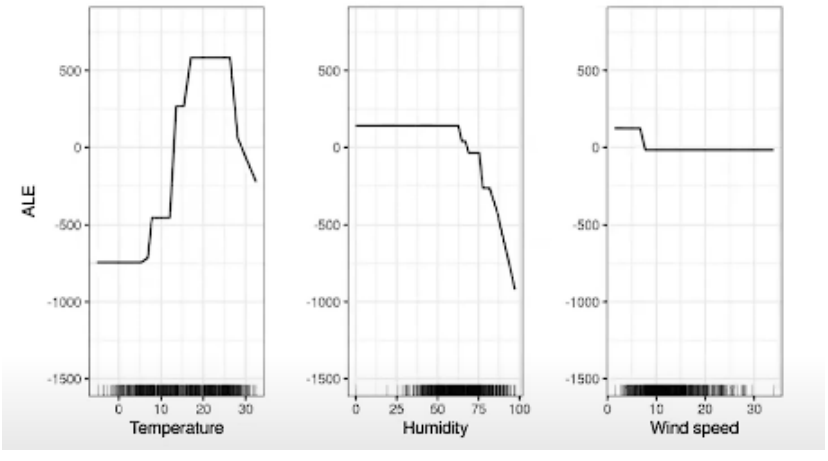
$$\hat{f}_{j,ALE}(x_S) = \sum_{k=1}^{v(x_S)} \frac{1}{n_k} \sum_{x \in N_k(x_S)} \left[\hat{f}(z_{S,k}^{(i)}, x_C^{(i)}) - \hat{f}(z_{S,k-1}^{(i)}, x_C^{(i)})\right]$$

1. $v(x_S)$:
   - x_S의 값 범위를 나눈 구간의 개수.
2. $N_k(x_S)$:
   - x_S의 k-번째 구간에 해당하는 데이터 포인트.
3. $\hat{f}(z_{S,k}, x_C) - \hat{f}(z_{S,k-1}, x_C)$:
   - x_S 값이 구간 $k-1$에서 k로 이동할 때, 예측값의 변화량.
4. **누적(sum)**:
   - 각 구간에서의 예측값 변화량을 누적하여 ALE 값을 계산.

Results

- 장점 : 변수 간 상관관계를 반영한다. 실제 데이터 분포를 기반으로 계산되므로,
  비현실적인 변수 조합을 사용하지 않는다.

- 단점 : 변수 변화에 따른 예측값의 차이를 누적하여 계산하기 때문에 해석이 다소 어려울 수 있다. 또한, 고차원 데이터에서는 한계가 있을 수 있다.



3. Concept-based Method

What does concept mean?

인간이 직관적으로 이해할 수 있는 정보를 의미한다.

딥러닝 모델은 입력 데이터에서 특정 개념을 자동으로 학습하지만, 사용자가 직접 feature를 제공하는 것이 아니므로, 모델이 학습한 개념을 이해하는 것이 중요하다.

이를 통해 모델의 신뢰성과 해석 가능성을 높이는 데에 기여할 수 있기 때문이다.

Concept-based Explanation이 효과적이라면, 아래의 3가지 속성을 만족해야한다.

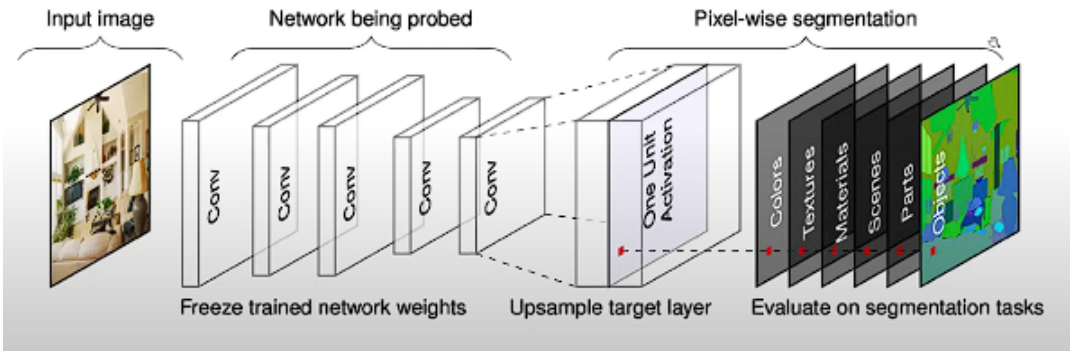
1. Meaningfulness
   - 개념이 스스로 의미를 가질 수 있어야 한다.
   - Ex) '얼룩말의 줄무늬 패턴', '고양이의 수염' 등과 같은 개념은 인간이 직관적으로 이해할 수 있다.
2. Coherency
   - 같은 개념에 속하는 데이터들은 서로 유사해야 한다.
3. Importance
   - 모델이 정확한 예측을 하기 위해서 반드시 필요한 개념이어야 한다.

3-1. Network Dissection

모델의 각각의 unit이 어떤 역할을 하는지 알아볼 수 있는 방법

아래와 같은 CNN 안의 convolution layer의 각각의 unit이 어떤 concept을 탐지하는지에 대하여 알아보는 것이다.

특정 뉴런이 어떤 개념을 학습했는지, CNN의 각 필터가 어떤 개념을 감지하는지..



Method

1️⃣ Broden Dataset 사용

- Broden Dataset은 이미지에 다양한 개념(concept)이 포함된 데이터셋이다.
- 이 데이터셋은 각 이미지 내 개념별로 픽셀 단위의 레이블(annotation)을 포함하고 있다.
- 개념은 Scene(장면), Object(객체), Part(부분), Texture(질감), Color(색상), Material(재질) 등으로 구성됨.

📌 핵심 포인트:

- CNN이 학습한 뉴런이 이런 개념들과 대응되는지 확인하기 위해 Broden Dataset사용
- 각 개념은 Binary mask로 표현, 즉 특징 개념이 존재하는 부분을 픽셀 단위로 표시함

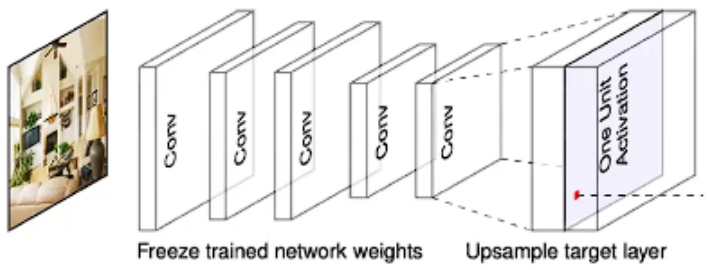
Figure 2. Samples from the **Broden** Dataset. The ground truth for each concept is a pixel-wise dense annotation.

2️⃣ 뉴런의 활성화(Activation MAP) 분석

- Broden Dataset에서 이미지를 입력하면, CNN 내부의 특정 뉴런(target unit)이 활성화됨.
- 활성화 맵은 해당 뉴런이 이미지의 어느 부분에 반응하는지 보여준다.

📌 핵심 포인트:

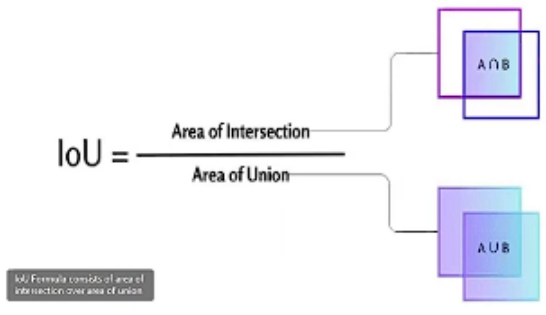
- 활성화 맵 : 뉴런이 이미지에서 강하게 반응하는 영역을 표시
- 해석 가능성 : CNN이 특정 개념에 반응하는지 확인하는 과정



3️⃣ IoU(Intersection over Union) 계산

- 뉴런의 활성화 영역과 Broden Dataset의 개념 레이블(ground truth)을 비교하여 IoU 점수를 계산한다.
- IoU (교집합 / 합집합) 값이 높을수록, 해당 뉴런이 특정 개념을 잘 학습한 것이다.

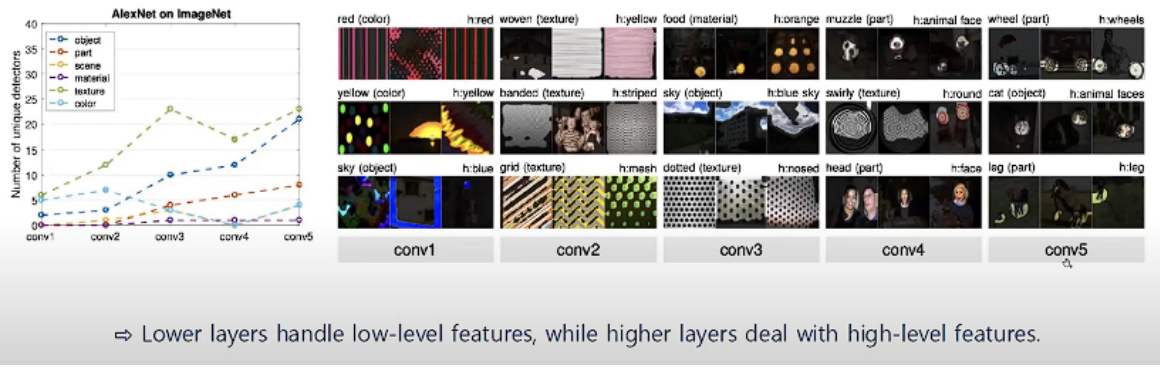
$$IoU_{k,c} = \frac{\sum |M_k(\mathbf{x}) \cap L_c(\mathbf{x})|}{\sum |M_k(\mathbf{x}) \cup L_c(\mathbf{x})|},$$



- $M_k(x)$: 뉴런 k의 활성화 맵
- $L_c(x)$: Broden 데이터셋의 개념 레이블
- 교집합 : 모델이 활성화한 영역과 실제 개념 레이블이 일치하는 부분

- 합집합 : 모델이 활성화한 영역과 개념 레이블의 총합

Results

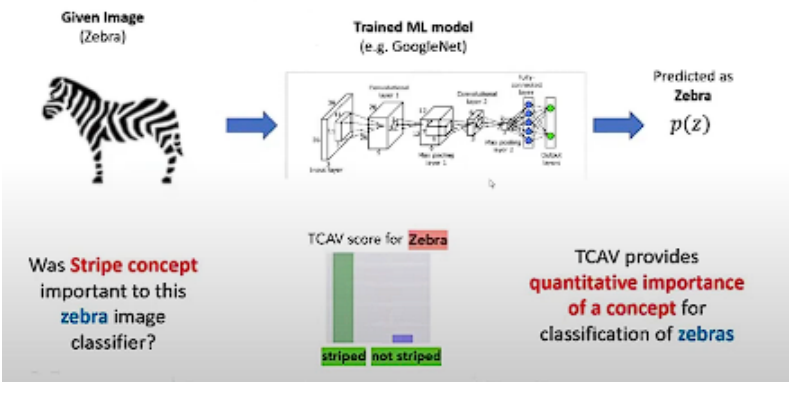


3-2. TCAV (Quantitative Test with Concept Activation Vectors)

딥러닝 모델이 특정 개념을 얼마나 중요하게 사용하는지 분석하는 방법이다.

기존의 해석 기법과 달리 모델의 내부 뉴런이 학습한 개념의 영향을 **정량화**할 수 있다.

- 아래 그림을 보자, 모델이 '얼룩말'을 예측할 때, '줄무늬' 개념이 중요할까? 라는 질문에
- TCAV는 줄무늬 개념이 모델의 예측에 얼마나 영향을 미치는지 측정하는 방법이다.



Method

1️⃣ 개념 데이터셋 구성

- 관심 있는 개념(Ex. 줄무늬)을 나타내는 Positive Set과 그렇지 않은 Negative Set을 구성.

2️⃣ 모델이 예측하는 대상 이미지 준비

- 분석하려는 이미지(Ex. 얼룩말)를 모델에 입력.

3️⃣ 모델의 내부 뉴런에서 활성화 값(Feature Representation) 추출

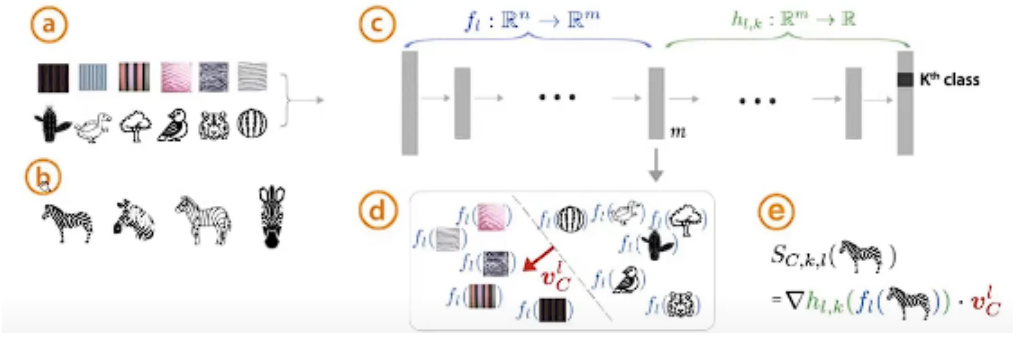
- CNN 모델의 중간층에서 활성화된 벡터(Feature Map)를 추출.
- 이는 모델이 해당 이미지를 어떻게 인식하는지를 보여준다.

4️⃣ 개념 벡터(Concept Activation Vector, CAV) 계산

- Positive Set과 Negative Set을 비교하는 SVM 분류기를 학습.
- SVM의 결정 경계(Decision Boundary)와 수직한 방향의 벡터를 CAV로 정의.
- 이 벡터는 특정 개념을 가장 잘 설명하는 방향을 나타낸다.

5️⃣ TCAV Score 계산

- 모델이 특정 개념(CAV 벡터 방향)으로 얼마나 민감하게 반응하는지 기울기 기반으로 측정한다.
- TCAV Score가 높을수록, 해당 개념이 모델 예측에서 중요한 역할을 한 것이다.



수식

TCAV Score $S_{C,k,l}(x)$는 특정 개념 C가 모델의 k번째 클래스의 예측값에 미치는 영향을 수치적으로 나타낸 것이다.

$$S_{C,k,l}(x) = \bigtriangledown h_{l,k}(f_l(x)) * v_C^l$$

1. $S_{C,k,l}(x)$

- 특정 개념 C가 모델의 k번째 클래스 예측값 $h_{l,k}$에 얼마나 영향을 미치는지 측정하는 값.
- 즉, **"이 개념이 모델의 최종 결정에 얼마나 중요한가?"**를 나타냄.

2. $f_l(x)$

- 모델의 l번째 층(layer)의 특징 표현(feature representation).
- 즉, 입력 x가 CNN 모델을 통과할 때, l번째 레이어에서 변환된 벡터.

3. $\bigtriangledown h_{l,k}(f_l(x))$

- 모델의 예측 $h_{l,k}$에 대한 기울기(gradient).
- 이는 입력 x의 내부 표현이 변화할 때, 모델 예측값이 얼마나 변화하는지를 나타냄.

4. v_C^l

- 개념 C에 대한 Concept Activation Vector(CAV).
- 즉, C라는 개념을 표현하는 방향 벡터.
- 이 벡터는 해당 개념을 표현하는 데이터들(Positive Set)과 그렇지 않은 데이터들을 구별하는 방향을 나타냄.

5. $\bigtriangledown h_{l,k}(f_l(x)) * v_C^l$

- C 개념 방향으로 입력이 변화할 때 모델 예측값이 얼마나 변하는지 내적 계산.
- 값이 클수록, 해당 개념이 모델의 예측에 큰 영향을 미친다는 의미.

[TCAV Score 미분 정의]

$$S_{C,k,l}(x) = \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(x) + \epsilon v_c^l) - h_{l,k}(f_l(x))}{\epsilon}$$

TCAV Score는 개념 방향으로 입력을 아주 조금 변경했을 때 모델 출력의 변화율을 측정하는 방식으로 정의할 수도 있다.

마찬가지로 이 미분 값이 크다면, 모델이 개념 C를 강하게 고려한다는 뜻이다!

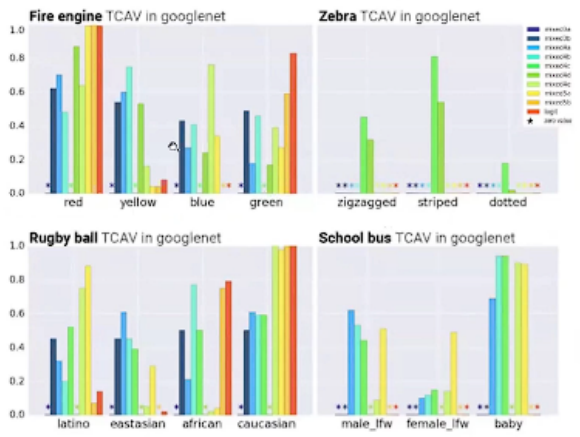
$$S_{C,k,l}(\text{zebra}) = \nabla h_{l,k}(f_l(\text{zebra})) \cdot \boldsymbol{v}_C^l$$

$$S_{C,k,l}(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{h_{l,k}(f_l(\boldsymbol{x}) + \epsilon \boldsymbol{v}_C^l) - h_{l,k}(f_l(\boldsymbol{x}))}{\epsilon} = \nabla h_{l,k}(f_l(\boldsymbol{x})) \cdot \boldsymbol{v}_C^l,$$

Results

⇨ The retrieval results of images similar or dissimilar to the target concept.

Results



⇨ It is possible to quantitatively measure which concepts are important for each class and which layers contain information about the target concept.

4. Prototype-based Method

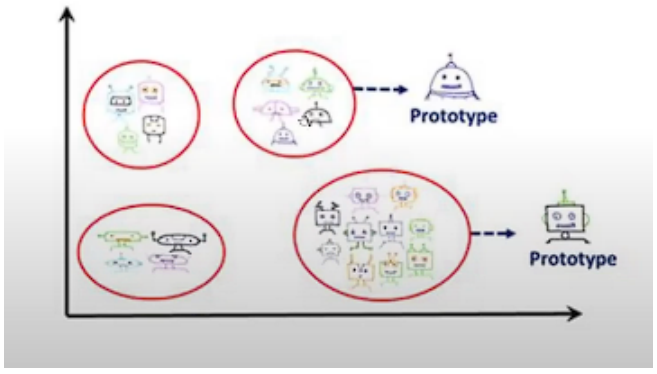
What does prototype mean?

Prototype이라는 것은 데이터를 대표할 수 있는 data example을 뜻한다.

즉, data distribution을 요약할 수 있는 subset을 뽑는 것이라고 생각하면 된다.

그렇다면, 왜 Prototype을 사용하는 걸까?

- 데이터의 대표적인 특징을 요약하는 소수의 샘플을 선택하여 데이터 전체를 대체할 수 있으므로, 학습 효율이 증가한다.
- 데이터의 구조를 유지하면서 설명 가능성을 향상시킬 수 있다.

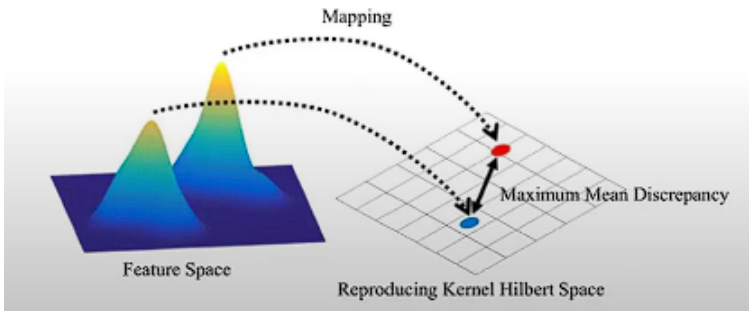


4-1. Maximum Mean Discrepancy (MMD)

MMD는 두 데이터 분포 X와 Y 사이의 차이를 측정하는 방법이다. 이는 머신러닝에서 두 개의 데이터 분포가 비슷한지를 평가할 때 사용된다.

그렇다면, MMD를 prototype 선택에 활용하는 이유는 무엇일까?
좋은 prototype은 전체 데이터 분포를 잘 반영해야 하는데, 원본 데이터 (
X)와 선택한 프로토타입 (Y) 사이의 MMD 값을 최소화하면, 프로토타입이 원본 데이터 분포를 잘 대표한다고 볼 수 있기 때문이다.

즉, MMD를 최소화하는 방식으로 프로토타입을 선택하면, 전체 데이터 분포를 가장 잘 나타내는 대표 샘플을 선택 가능하다!



Method

$$MMD^2(F,X,Y) = [\frac{1}{n}\sum_{i\in[n]}\phi(x_i) - \frac{1}{m}\sum_{j\in[m]}\phi(y_j)]^2$$

1. $\phi(x)$ **변환 (Mapping)**

- 데이터를 고차원 힐베르트 공간(Hilbert Space)으로 변환하여 분포 간 차이를 측정.
- 이 변환을 통해, 복잡한 분포 차이를 더 정확하게 측정할 수 있다.

2. **평균 벡터 차이 계산**

- 원본 데이터 X의 평균 벡터와 프로토 타입 Y의 평균 벡터 차이를 계산.
- 즉, X와 Y가 비슷한지 비교.

3. **커널 트릭(Kernel Trick) 적용**

- 실제로 $\phi(x)$를 계산하는 대신, 커널 함수 k(x, y)를 사용하여 계산을 단순화함.

$$k(X,Y) =< \phi(X), \phi(Y) >_F$$

4. **최종 근사화 수식**

- 힐베르트 공간에서의 거리 계산을 커널 함수로 근사

$$MMD^2 \approx \frac{1}{n^2}\sum_{i,j\in[n]}k(x_i,x_j) - \frac{2}{nm}\sum_{i\in[n],j\in[m]}k(x_i,y_j) + \frac{1}{m^2}\sum_{i,j\in[m]}k(y_i,y_j)$$

How to calculate the distance between distributions X and Y over a function space F

$$MMD^2(\mathcal{F}, X, Y) = \left(\frac{1}{n}\sum_{i\in[n]}\phi(x_i) - \frac{1}{m}\sum_{j\in[m]}\phi(y_j)\right)^2$$

Measure the difference between the mean of X and the mean of Y in Hilbert space.

$$k(X,Y) = \langle\phi(X), \phi(Y)\rangle_{\mathcal{F}}$$

Kernel Trick

$$\approx \frac{1}{n^2}\sum_{i,j\in[n]} k(x_i, x_j) - \frac{2}{nm}\sum_{i\in[n],j\in[m]} k(x_i, y_j) + \frac{1}{m^2}\sum_{i,j\in[m]} k(y_i, y_j)$$

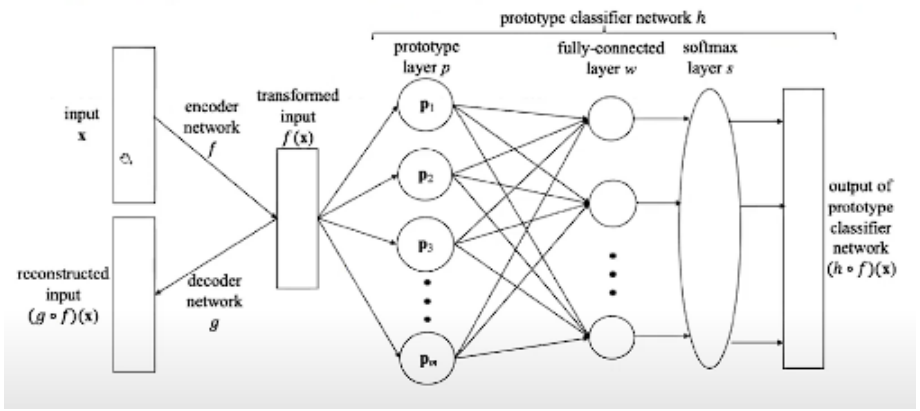
→ Treat the variables for the original data X as constants and find the prototype set Y that minimizes the loss.

4-2. Generating the prototype

Deep Learning for Case-Based Reasoning through Prototypes(AAAI, 2018) 논문은 위에서 알아본 기존의 프로토타입을 선택하는 방법이 아니라, 새로운 데이터를 생성하여 설명 가능성을 높이는 방법을 제안하였다.

이는 기존 데이터에서 프로토타입을 선택하는 것이 아닌, 생성하는 방식으로 모델의 해석력을 향상시키고 데이터셋의 중심적인 개념을 명확히 설명할 수 있게 한다.

Prototype 기반 신경망 : 설명 및 학습 손실 구성 요소



1. **네트워크 구조 및 프로토타입 레이어**

- 입력 x를 인코더 네트워크 $f(x)$를 통해 잠재 공간(latent space)으로 변환.
- 각 클래스별 대표적인 프로토타입 벡터 p_i들이 존재하며, 입력이 이 프로토타입들과 얼마나 가까운지 계산한다.
- 프로토타입과의 거리를 특징으로 활용하여 분류를 수행한다.

$$p(z) = [\|z - p_1\|_2^2, \|z - p_2\|_2^2, ..., \|z - p_m\|_2^2]^T, \quad \text{where } z = f(x_i)$$

- 각 입력이 프로토타입과 얼마나 가까운지 L2 거리를 계산하여 새로운 특징으로 사용.

2. **학습 손실 함수 구성**

- 이 모델은 3가지 주요 손실 함수를 사용한다.
  1. Cross-Entropy Loss

$$E(h \circ f, D) = \frac{1}{n}\sum_{i-1}^{n}\sum_{k-1}^{K} -1[y_i = k]\log((h \circ f)_k(x_i))$$

  - 모델이 정확하게 분류할 수 있도록 학습하는 손실 함수.
  - 표준 크로스 엔트로피 손실을 사용하여 프로토타입 기반 분류기의 정확도를 높인다.

  2. Regularization Loss, R1 & R2

  a. 적어도 하나의 훈련 샘플과 가까워야 함

$$R_1(p_1, ..., p_m, D) = \frac{1}{m} \sum_{j-1}^{m} \min_{i \in [1,n]} \|p_j - f(x_i)\|_2^2$$

- 각 프로토타입 벡터 p_j는 반드시 어떤 훈련 데이터 포인트와 가까워야 한다.
- 즉, 프로토타입이 실제 데이터와 관련이 있는 유의미한 벡터가 되도록 유도.

b. 모든 입력 데이터가 적어도 하나의 프로토타입과 가까워야 함

$$R_2(p_1, ..., p_m, D) = \frac{1}{n} \sum_{i-1}^{n} \min_{j \in [1,m]} \|f(x_i) - p_j\|_2^2$$

- 각 입력 데이터 x_i가 적어도 하나의 프로토타입과 가까운 거리에 위치해야 한다.
- 즉, 훈련 데이터가 프로토타입 주변으로 잘 클러스터링(clusterinig)되도록 유도.

→ 이 두 가지 손실이 함께 작동하여, 프로토타입이 의미 있는 위치에 있도록 보장한다!

3. Reconstruction Loss

$$R(g \circ f, D) = \frac{1}{n} \sum_{i-1}^{n} \|(g \circ f)(x_i) - x_i\|_2^2$$

- 오토인코더의 디코더 $g(f(x))$가 입력을 정확히 재구성하도록 유도.
- 즉, 잠패 표현 $f(x)$이 입력 데이터를 충분히 보존할 수 있도록 학습.

→ 이는 프로토타입을 생성하는데 중요한 역할을 하며, 프로토타입 기반의 데이터 생성을 가능하게 한다!

Results

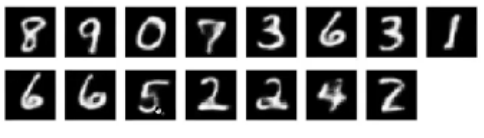


Figure 3: 15 learned MNIST prototypes visualized in pixel space.

Table 2: The (rounded) distances between a test image 6 and every prototype in the latent space.

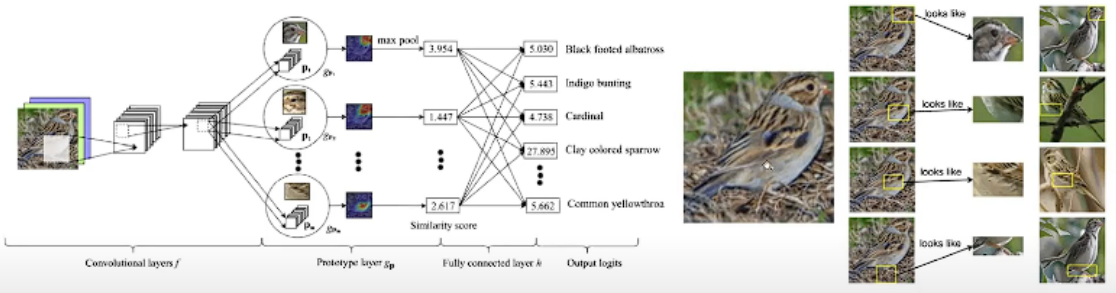
4-3. ProtoPNet (local explanation)

기존 방식에서는 프로토타입을 전체 데이터 수준에서 학습했지만, ProtoPNet은 이미지를 패치(patch) 단위로 나누어 프로토타입을 학습한다.

- CNN을 통해 특징을 추출, 각 특징이 훈련 데이터의 특정 부분과 얼마나 유사한지를 평가
- 훈련 데이터의 가장 유사한 부분을 찾아서 해석 가능성을 높임
- 최종적으로, 가장 유사한 프로토타입의 영향을 반영하여 분류를 수행

→ 즉, ProtoPNet은 **"이 샘플이 특정 클래스로 분류된 이유"**를 훈련 데이터의 특정 부분과 비교하여 설명할 수 있다!

Advancement: ProtoPNet (local explanation)



- **Train proto vectors on a patch level** and **provide the most similar part of the training example** without using an autoencoder.