알고 있으면 쓸모 있는 AI 지식

Transfer Learning

Meta Learning

Fewshot Learning

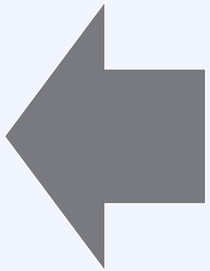
알고 있으면 쓸모 있는 AI 지식

Transfer Learning

Meta Learning

Fewshot Learning

...



인공신경망 학습이

가지고 있는 문제점

인공신경망 학습의 문제점

➢ 어떻게 신경망을 설계해야 좋은 성능을 보일지 모르겠다.

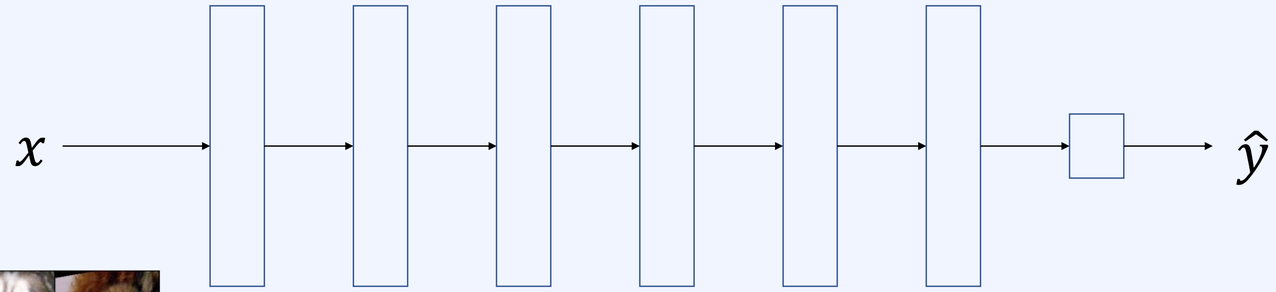
- 설정해야 할 하이퍼파라미터가 너무 많음

➢ 데이터가 충분하지 않다.

➢ 학습하는데 시간이 너무 오래 걸린다.

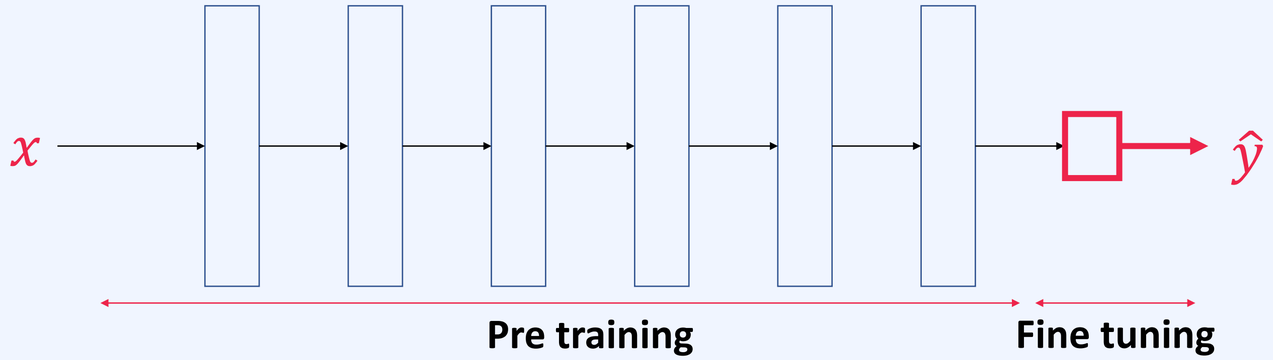
Transfer Learning

- Image recognition
  - cats vs dogs



Transfer Learning

- Image recognition
  - Normal vs Pneumonia



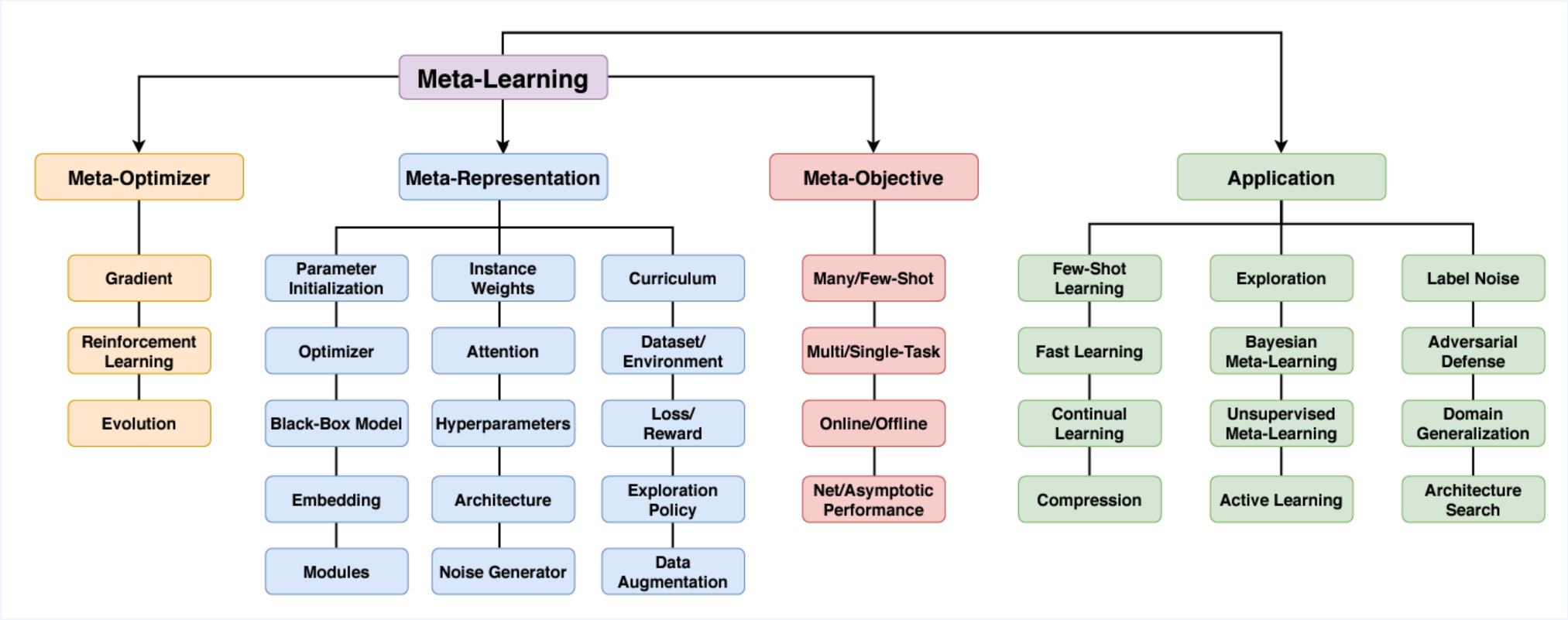
Meta Learning

- Learn to learn
  - 학습하는 방법을 배우자

- 인공신경망의 “학습“
  - 신경망을 구성하는 최적의 파라미터를 찾아가는 과정

Meta Learning

➢ Meta-Learning in Neural Networks: A Survey, Timothy Hospedales et al. 2020



Few shot Learning

- k-way n-shot
  - k: 클래스의 개수
  - n: 클래스가 가진 샘플 수

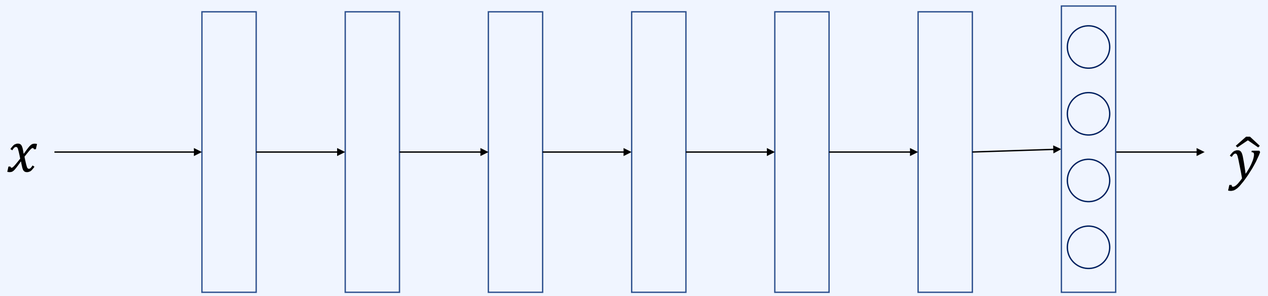
1 shot

4 way

이미지 출처: 위키백과

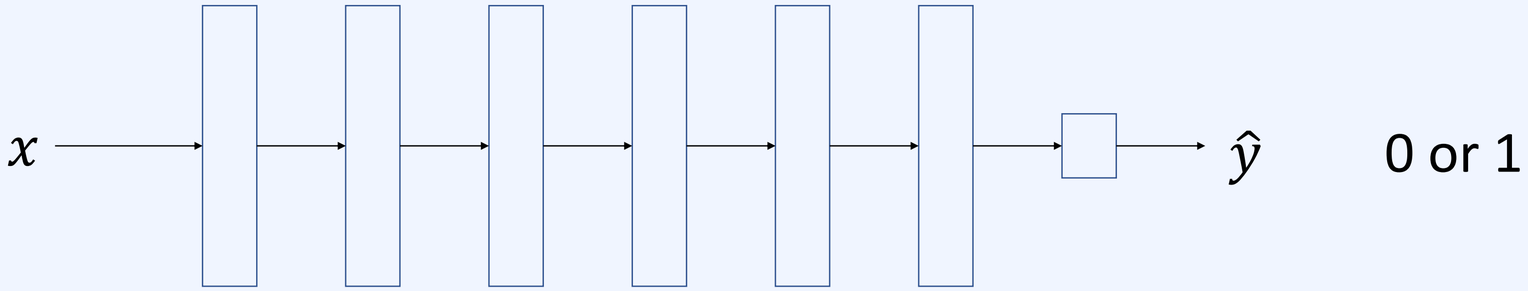
One shot Learning

- 일반적인 신경망의 학습
  - 지정된 범주에 속하는 정도를 판단



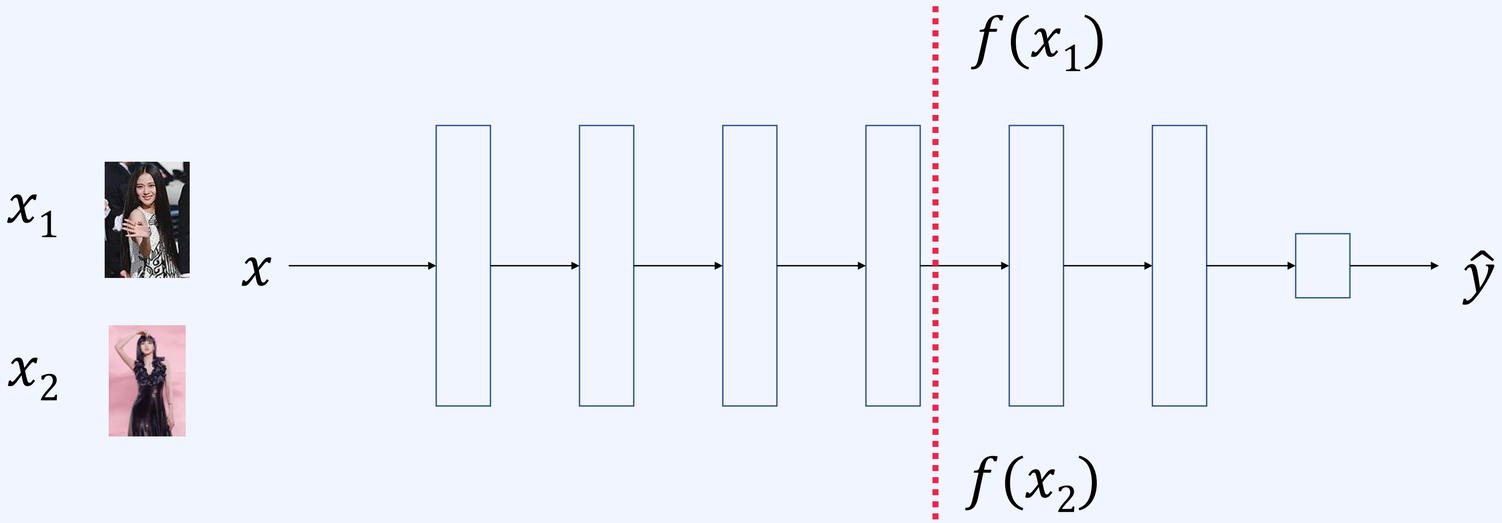
One shot Learning

- Few shot learning의 학습
  - 데이터 간의 유사도를 학습

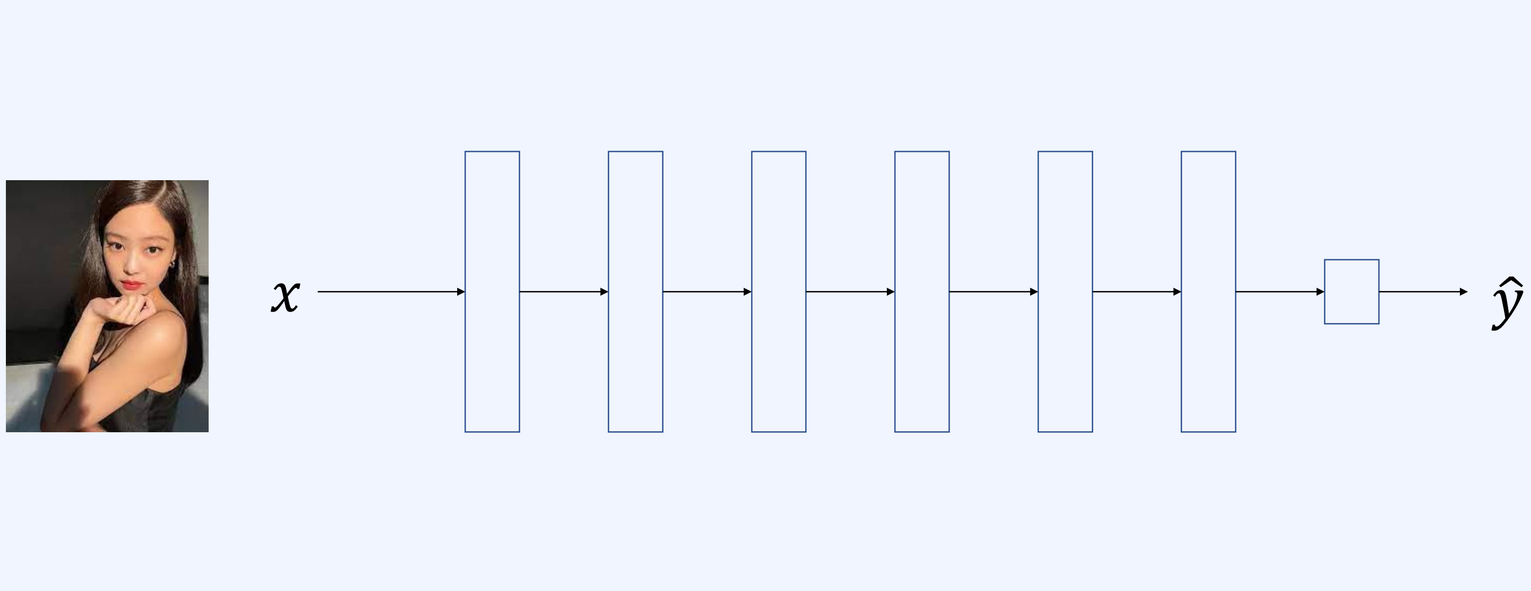


Siamese Network

$$||f(x_1) - f(x_2)||^2$$



Siamese Network



Summary

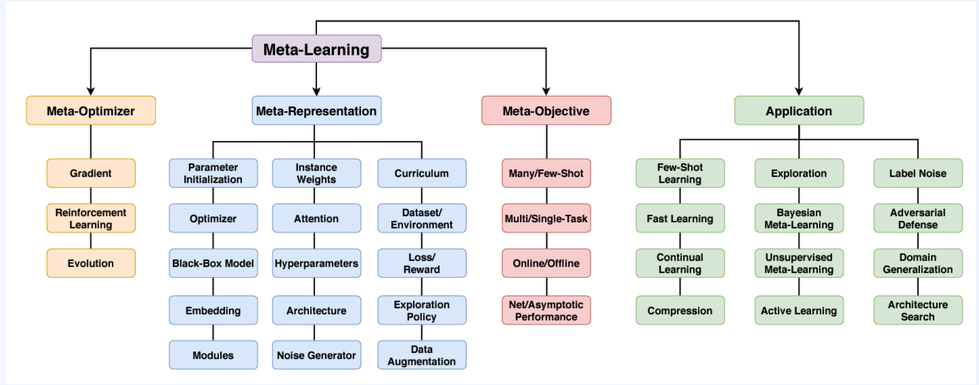
➢ 인공신경망 학습의 해결 과제

- 어떻게 신경망을 설계해야 좋은 성능을 보일지 모르겠다.
  - 설정해야 할 하이퍼파라미터가 너무 많음
- 데이터가 충분하지 않다.
- 학습하는데 시간이 너무 오래 걸린다.

➢ 인공신경망 학습의 해결 방법

- Transfer Learning
  - 유사한 Task를 수행하는 신경망의 학습 파라미터를 재사용

- Few shot Learning(Meta Learning)
  - 데이터 간의 유사도를 측정하는 함수를 학습



알고 있으면 쓸모 있는 AI 지식

XAI

인공신경망 학습의 문제점

➢어떻게 신경망을 설계해야 좋은 성능을 보일지 모르겠다.

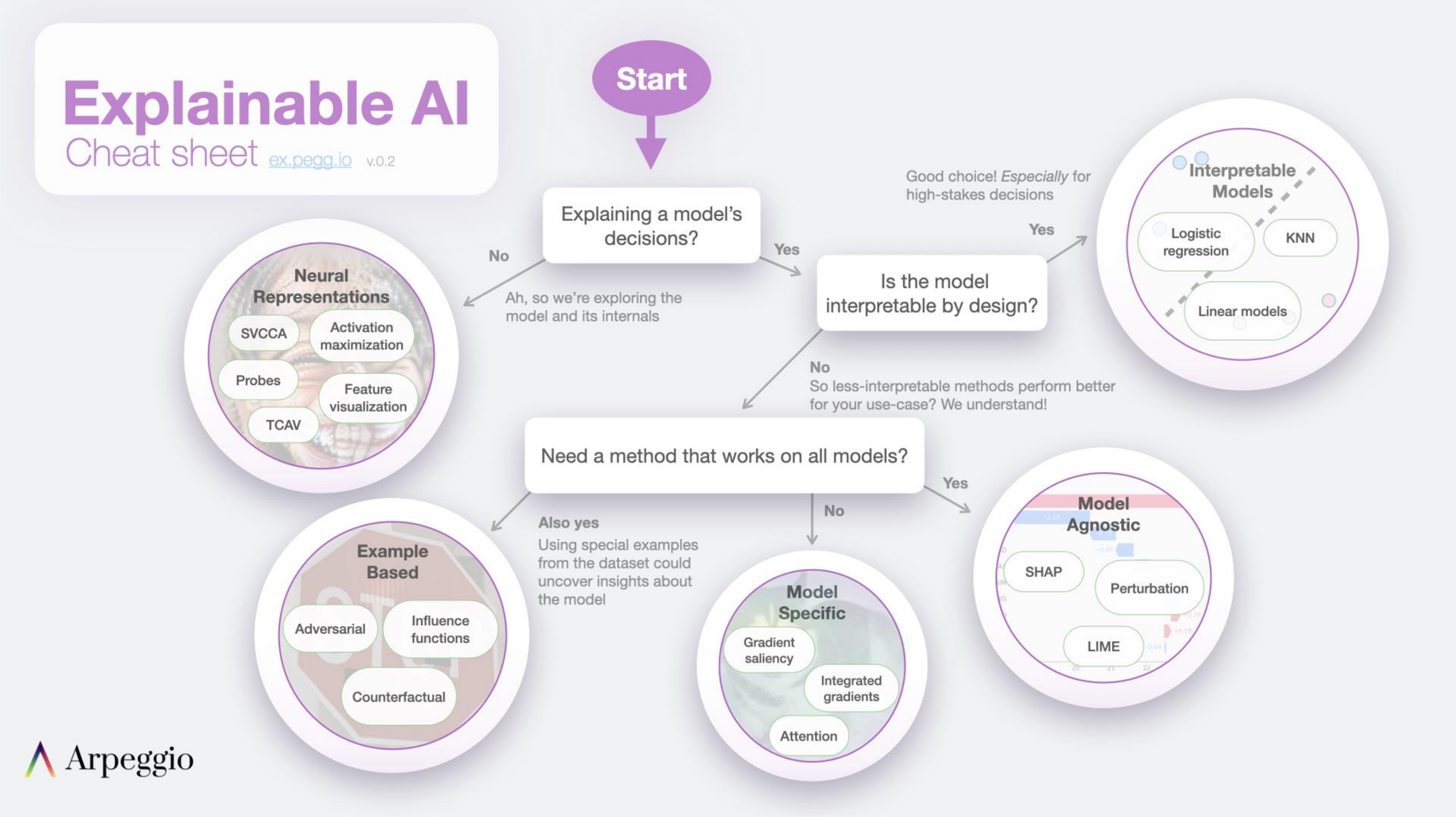
- 설정해야 할 하이퍼파라미터가 너무 많음

➢데이터가 충분하지 않다.

➢학습하는데 시간이 너무 오래 걸린다.

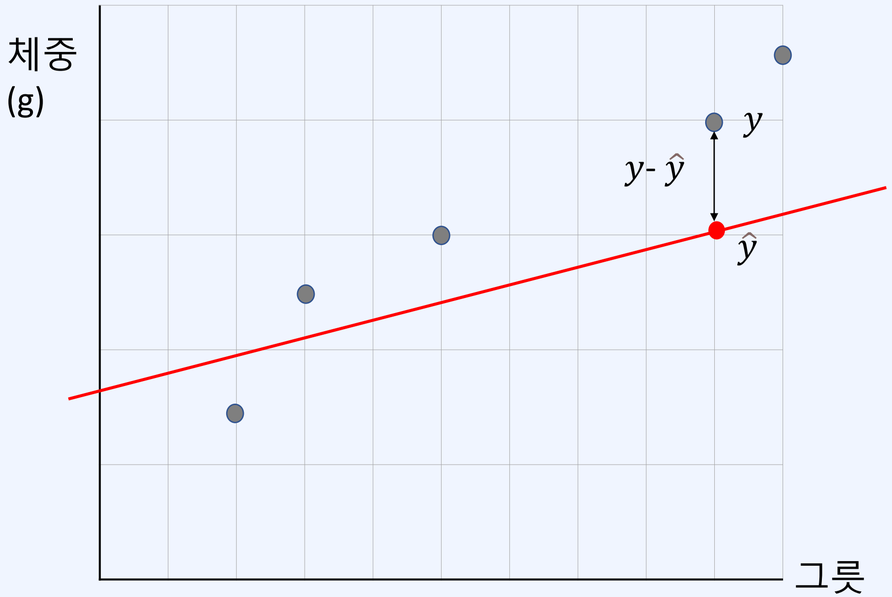
➢왜 이런 결과가 나왔는지 알 수 없다.

XAI

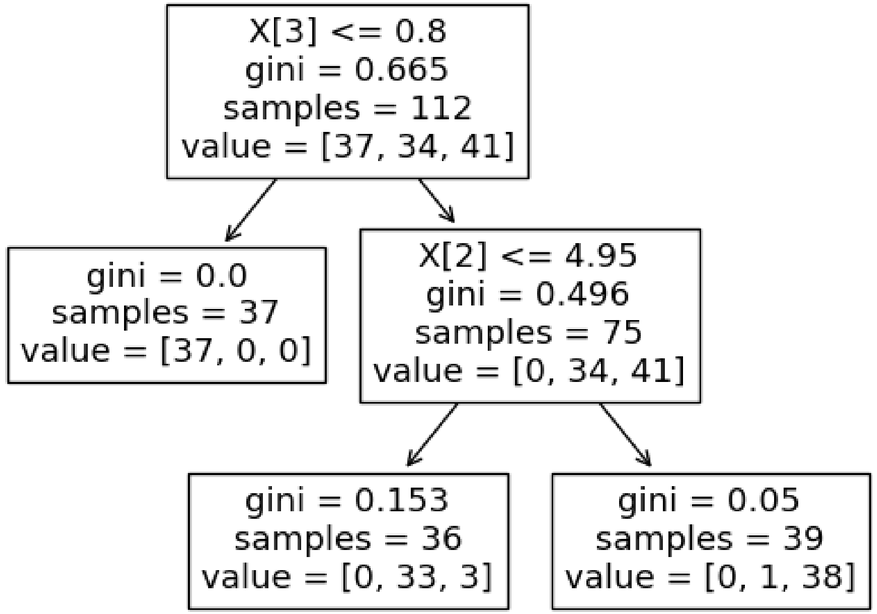


https://ex.pegg.io/

Interpretable Model



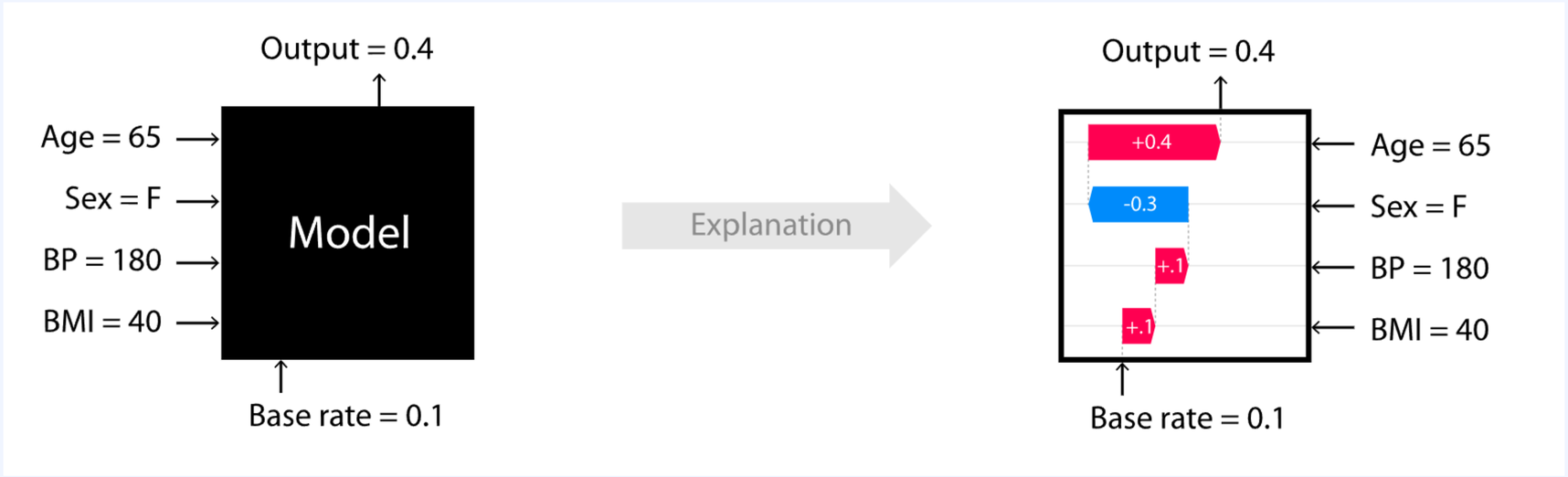
$$H(x) = \hat{y} = wx + b$$



Model Agnostic

➢ Model Agnostic
- 학습에 사용된 model이 무엇인지 상관없이 모델을 해석할 수 있다

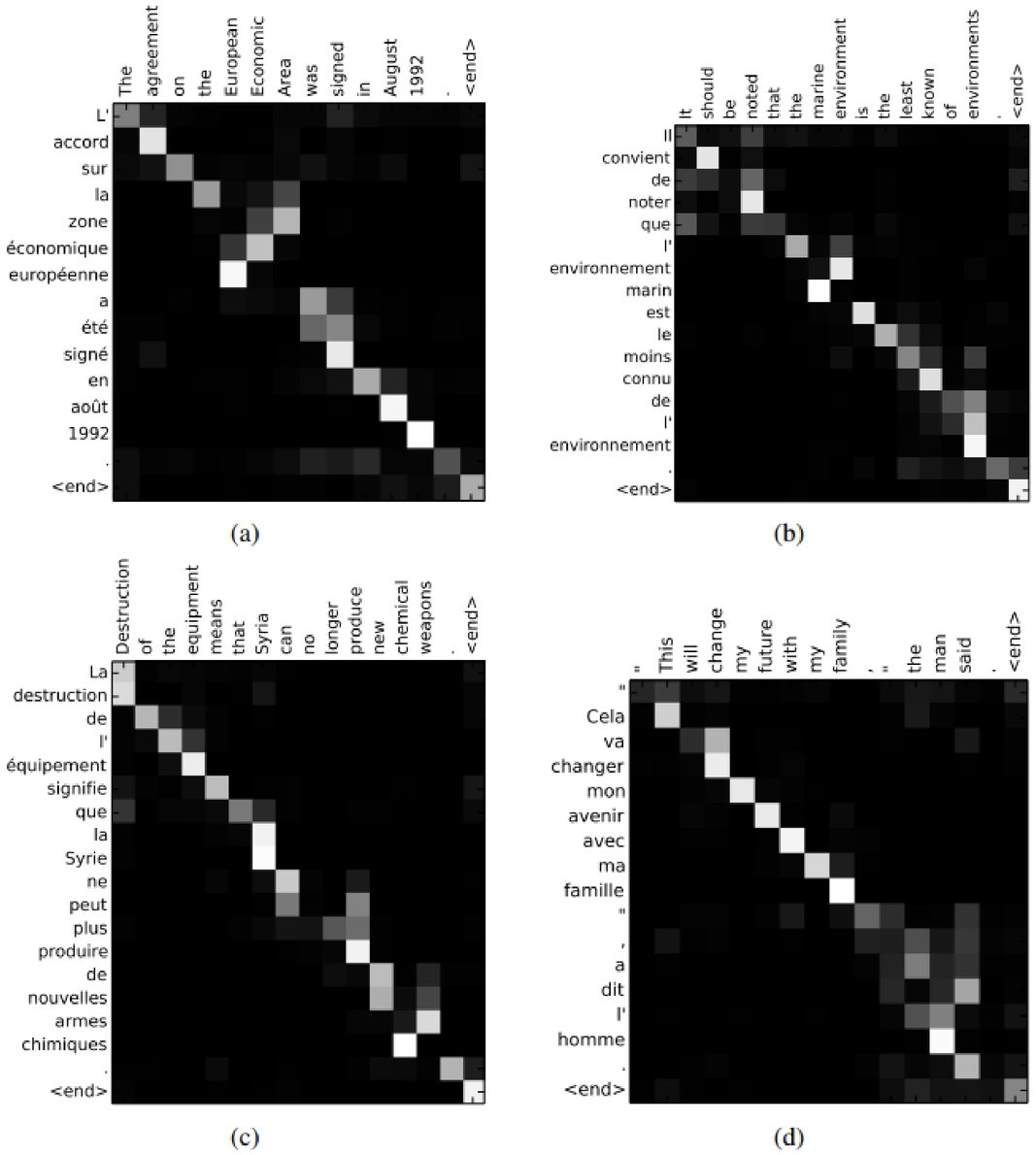
➢ SHAP Values(SHapley Additive exPlanations)
- 어떤 특정 feature가 어떤 값을 취했을 경우, 모델을 통해서 얻게 되는 예측 결과를 바탕으로 특정 feature가 모델에 미치는 영향력을 해석
  - 예측을 세분화하여 각 피처의 영향을 확인 가능



https://github.com/slundberg/shap

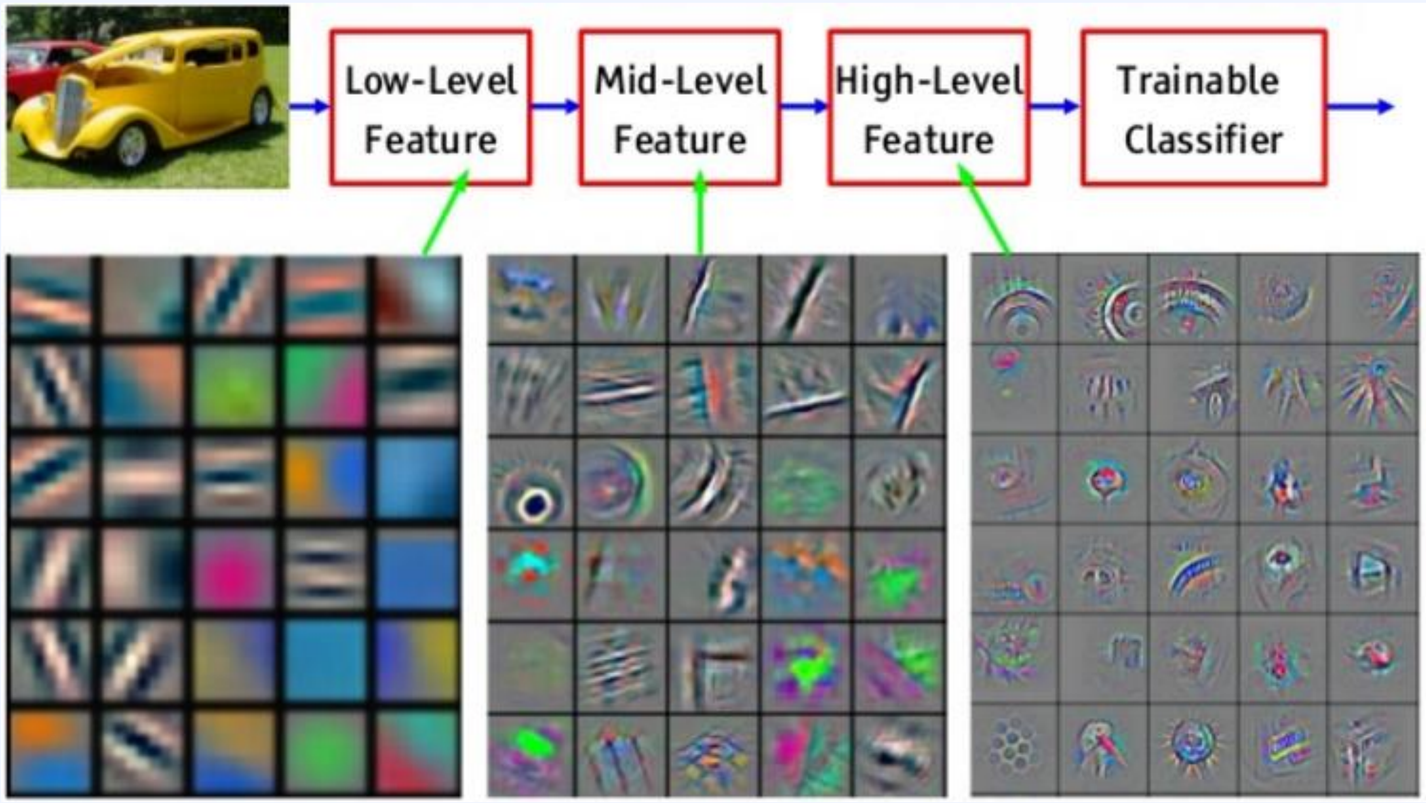
Model Specific

- Attention
  - Machine Translation
    - English - French



Neural Representations

➢ Feature visualization



XAI의 효과

- Model은 어떤 feature를 가장 중요하게 생각하는가?
- 각각의 feature가 Model 예측 결과에 어떠한 영향력을 주는가?
- 각 feature의 일반적인 효과는 어떻게 되는가?

XAI가 필요한 경우

- 디버깅
- 피처 엔지니어링
  - 하나의 피처와 다른 피처를 조합하여 연산하여 새로운 피처를 도출하는 작업을 할 때 인사이트를 줄 수 있음
- 데이터 수집을 위한 가이드
  - 가치있고 필요한 데이터만 수집할 수 있음
- Model의 예측 결과를 바탕으로 중요 의사결정을 할 때
  - model을 실제로 적용하기 위한 신뢰

Summary

➢ XAI(eXplainable AI)

- 설명가능한 인공지능

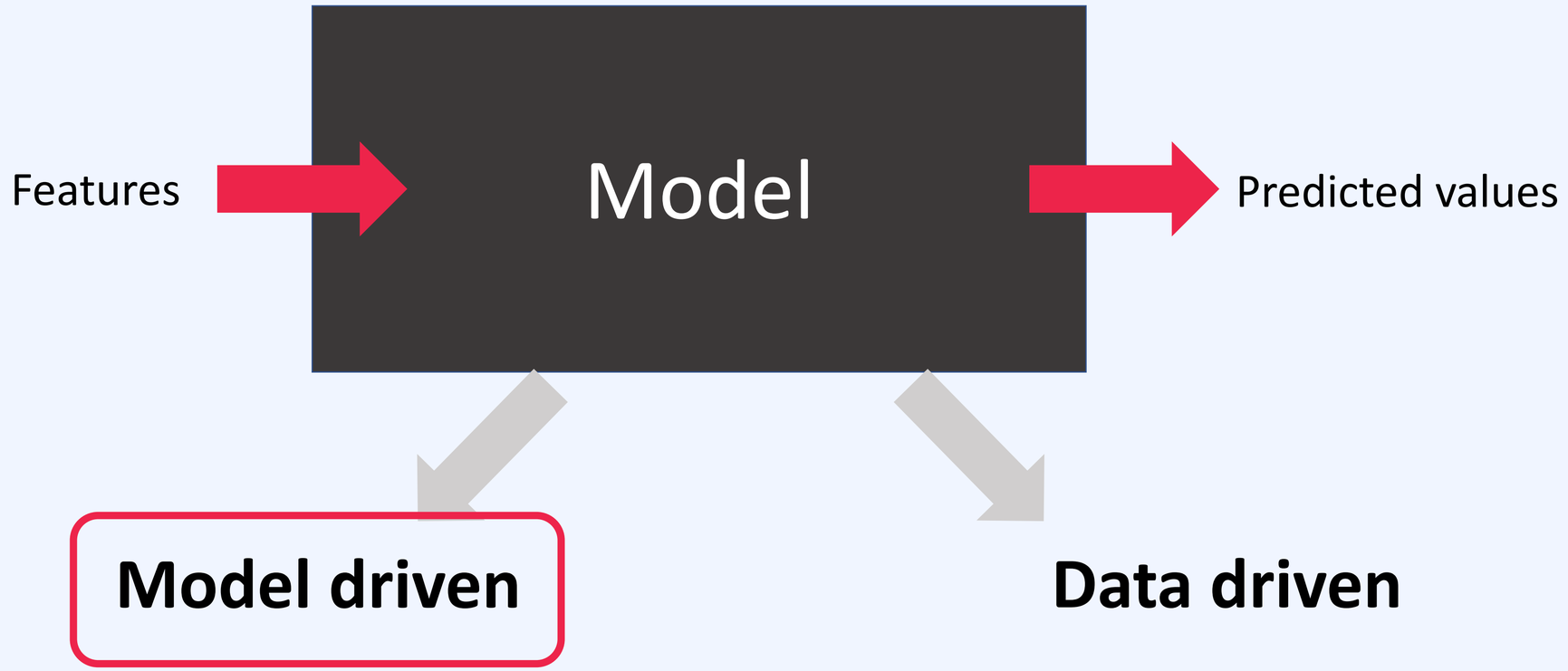
- 모델의 학습 결과를 해석할 수 있는 방법
  - 모델 성능 개선을 위한 방법을 찾거나 예측 모델을 실제로 사용할 수 있게 하는데(=모델의 신뢰도 상승) 도움

- 방법
  - Interpretable Models
    - 해석이 가능한 형태의 모델
  - Model Agnostic
    - 모델의 종류와 상관없이 해석할 수 있는 방법 (예: SHAP)
  - Model Specific
    - 모델 자체의 동작을 바탕으로 해석할 수 있는 방법
  - Neural Representations
    - 신경망의 학습 과정에서 추출한 feature의 시각화를 통해 신경망의 학습 내용을 이해

알고 있으면 쓸모 있는 AI 지식

신경망의 성능 개선 방법
(신경망 설계 관점)

Model의 성능 개선 방법



신경망의 성능 개선 방법

- **model driven** vs data driven
  - weight initialization
  - drop out
  - batch normalization
  - early stop
  - transfer learning
  - end to end vs part
  - ml vs dl

신경망의 성능 개선 방법

➢ **model driven** vs data driven
- **weight initialization**
- drop out
- batch normalization
- early stop
- transfer learning
- end to end vs part
- ml vs dl

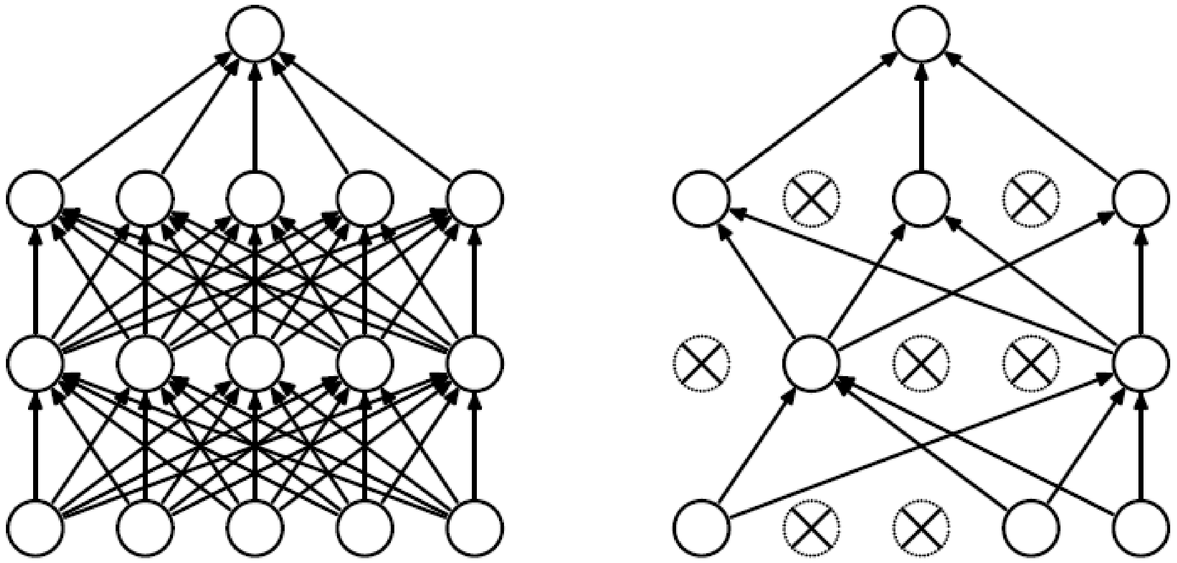
- **Xavier Initialization**
  - **activation function** : tanh, sigmoid
- **He Initialization**
  - **activation function** : relu

```
# xavier initialization
torch.nn.init.xavier_uniform_(linear1.weight)
torch.nn.init.xavier_uniform_(linear2.weight)
torch.nn.init.xavier_uniform_(linear3.weight)
```

신경망의 성능 개선 방법

➢ **model driven** vs data driven
- weight initialization
- **drop out**
- batch normalization
- early stop
- transfer learning
- end to end vs part
- ml vs dl

- **학습 중에 은닉층의 뉴런을 무작위로 삭제하여 과적합을 예방하는 기법**
- **p라는 확률로 출력 노드의 신호를 보내다 말다 함**
  - 드롭아웃을 적용한 다음에 오는 계층은 앞 계층의 일부 노드들의 신호가 p라는 확률로 단절되기 때문에 훨씬 더 견고하게 신호에 적응

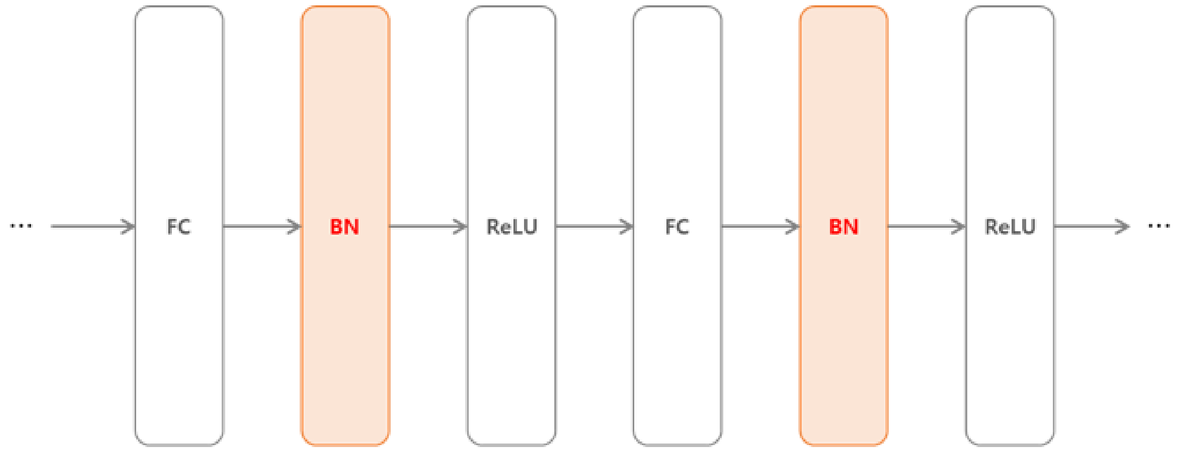


(a) Standard Neural Net (b) After applying dropout.

신경망의 성능 개선 방법

➢ **model driven** vs data driven
- weight initialization
- drop out
- **batch normalization**
- early stop
- transfer learning
- end to end vs part
- ml vs dl

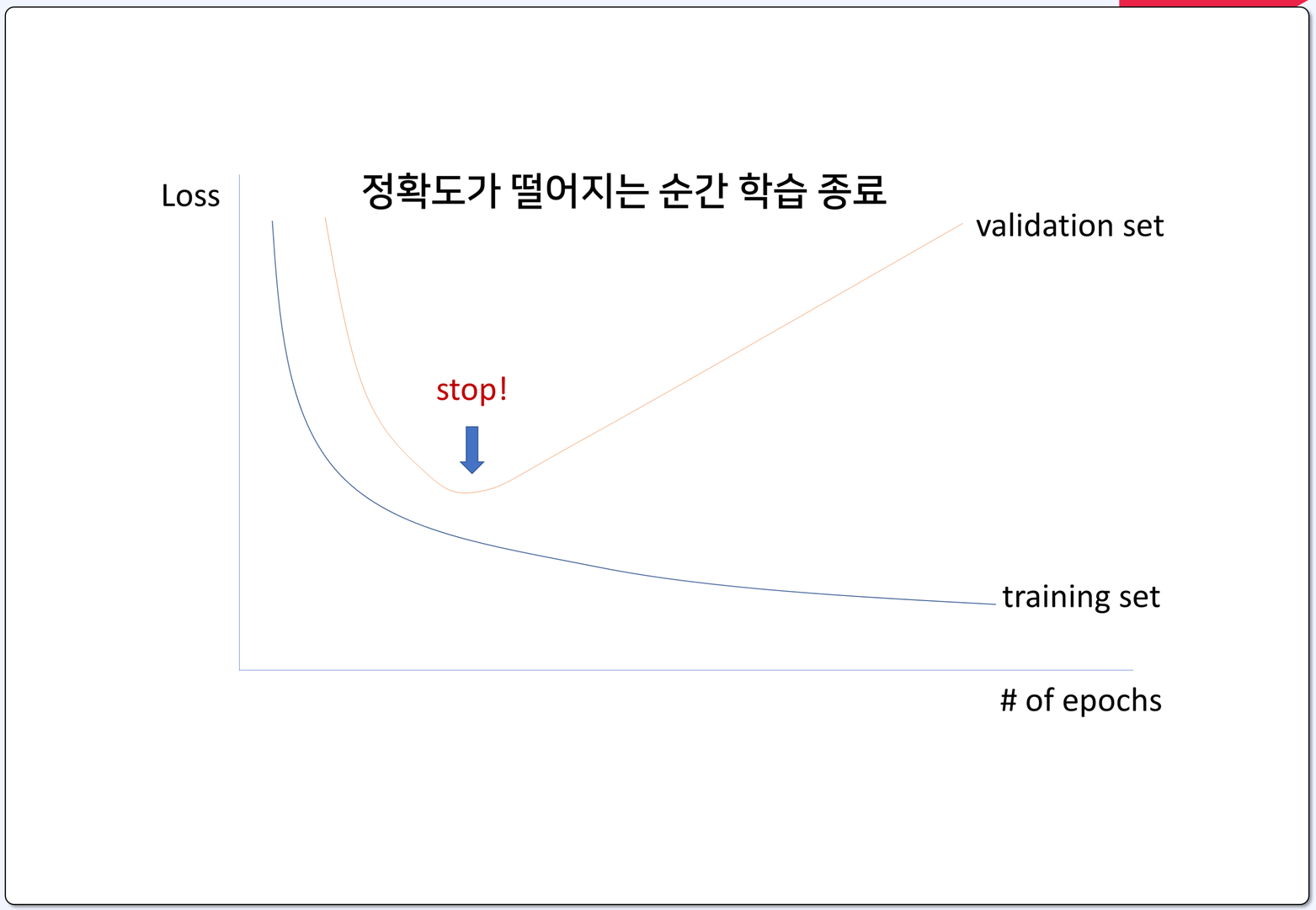
- 한 번에 입력으로 들어오는 **배치 단위로 데이터 분포의 평균이 0, 분산이 1**이 되도록 정규화 진행
- **빠른 학습 가능:** learning rate를 높게 잡을 수 있음
- **자체적인** regularization 효과



[BN을 사용한 신경망 예]

신경망의 성능 개선 방법

- **model driven** vs data driven
  - weight initialization
  - drop out
  - batch normalization
  - **early stop**
  - transfer learning
  - end to end vs part
  - ml vs dl



신경망의 성능 개선 방법

TIP

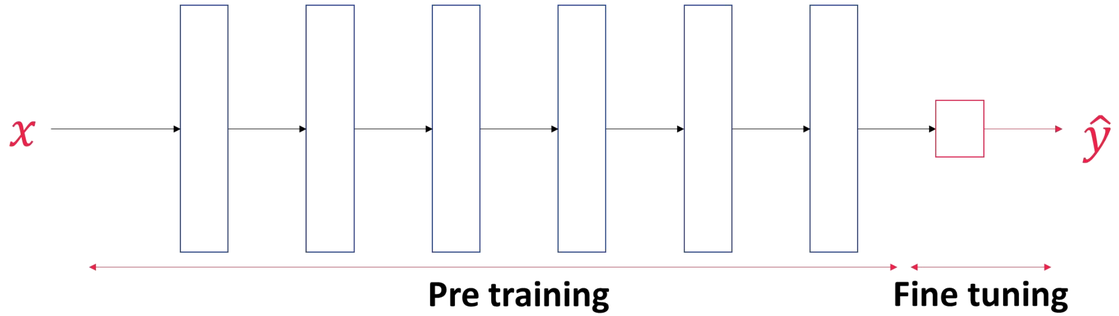
- **model driven** vs data driven
  - weight initialization
  - drop out
  - batch normalization
  - **early stop**
  - transfer learning
  - end to end vs part
  - ml vs dl

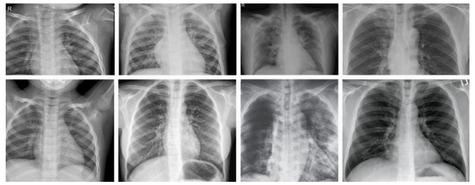
```python
# Train
def traindata(device, model, epochs, optimizer, loss_function, train_loader, valid_loader):
    # Early stopping
    last_loss = 100
    patience = 2
    triggertimes = 0

    for epoch in range(1, epochs+1):
        model.train()

        for times, data in enumerate(train_loader, 1):
            input, label = data[0].to(device), data[1].to(device)

            # Zero the gradients
            optimizer.zero_grad()

            # Forward and backward propagation
            output = model(input.view(input.shape[0], -1))
            loss = loss_function(output, label)
            loss.backward()
            optimizer.step()

            # Show progress
            if times % 100 == 0 or times == len(train_loader):
                print('[{}/{}, {}/{}] loss: {:.8}'.format(epoch, epochs, times, len(train_loader), loss.item()))

        # Early stopping
        current_loss = validation(model, device, valid_loader, loss_function)
        print('The Current Loss:', current_loss)

        if current_loss > last_loss:
            trigger_times += 1
            print('Trigger Times:', trigger_times)

            if trigger_times >= patience:
                print('Early stopping!₩nStart to test process.')
                return model
        else:
            print('trigger times: 0')
            trigger_times = 0

        last_loss = current_loss

    return model
```

신경망의 성능 개선 방법

➢ **model driven** vs data driven
- weight initialization
- drop out
- batch normalization
- early stop
- **transfer learning**
- end to end vs part
- ml vs dl

- 기존의 만들어진 모델을 사용하여 새로운 모델을 만드는 방법
- 학습을 빠르게 하며 예측 성능을 더 높임
- 이미 잘 훈련된 모델이 있고, 특히 해당 모델과 유사한 문제를 해결 시 효과적

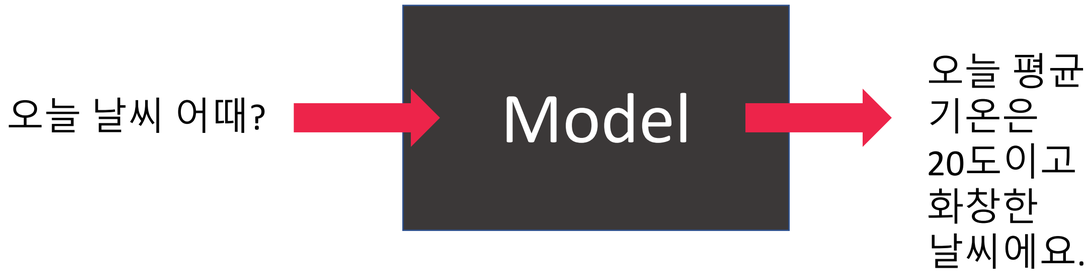




신경망의 성능 개선 방법

➢ **model driven** vs data driven
- weight initialization
- drop out
- batch normalization
- early stop
- transfer learning
- **end to end vs part**
- ml vs dl

- All in one?



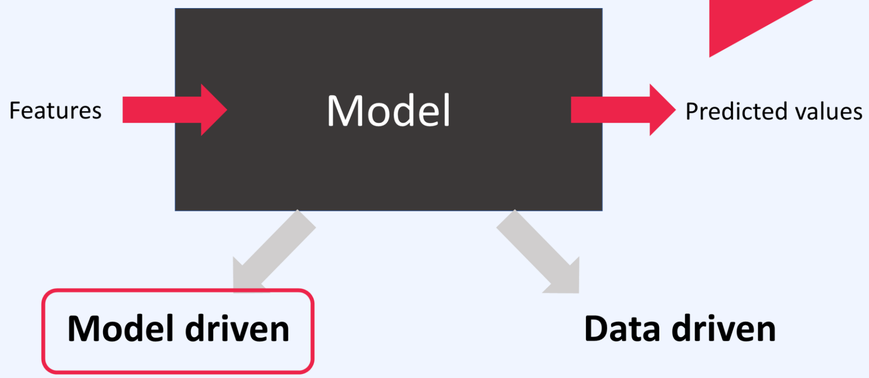
신경망의 성능 개선 방법

➢ **model driven** vs data driven
- weight initialization
- drop out
- batch normalization
- early stop
- transfer learning
- end to end vs part
- **ml vs dl**

- Rule based
- Statistical based
- Learning based

Summary

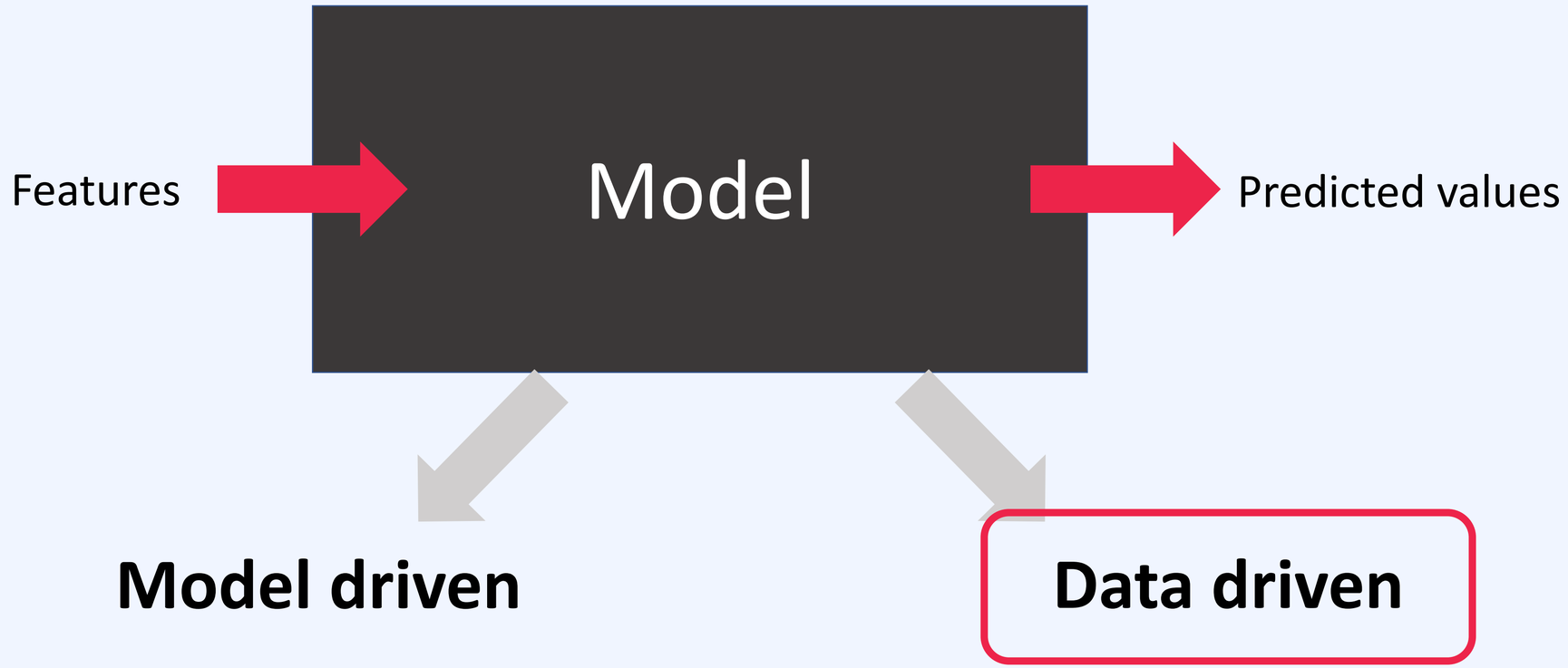
- **model driven** vs data driven
  - weight initialization
  - drop out
  - batch normalization
  - early stop
  - transfer learning
  - end to end vs part
  - ml vs dl



알고 있으면 쓸모 있는 AI 지식

신경망의 성능 개선 방법
(데이터 관점)

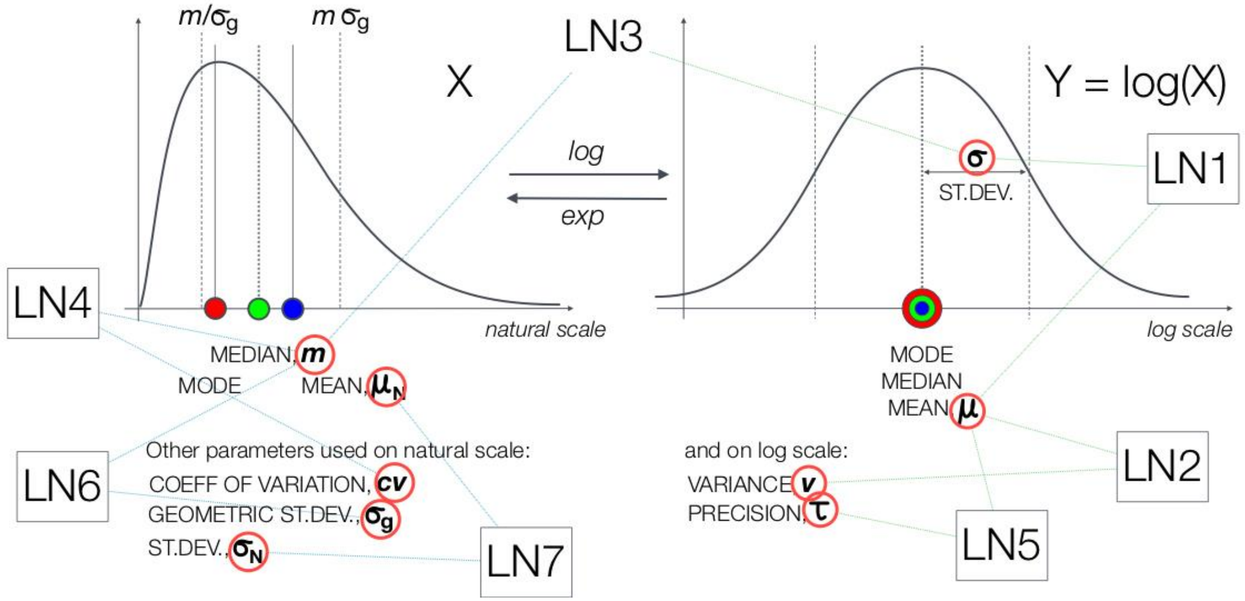
Model의 성능 개선 방법



Model의 성능 개선 방법

- model driven vs **data driven**
  - **data distribution**
  - data scale
  - domain knowledge
    - binning
  - feature engineering
  - dimension reduction
  - train set vs test set vs real data

- Log-normal distribution

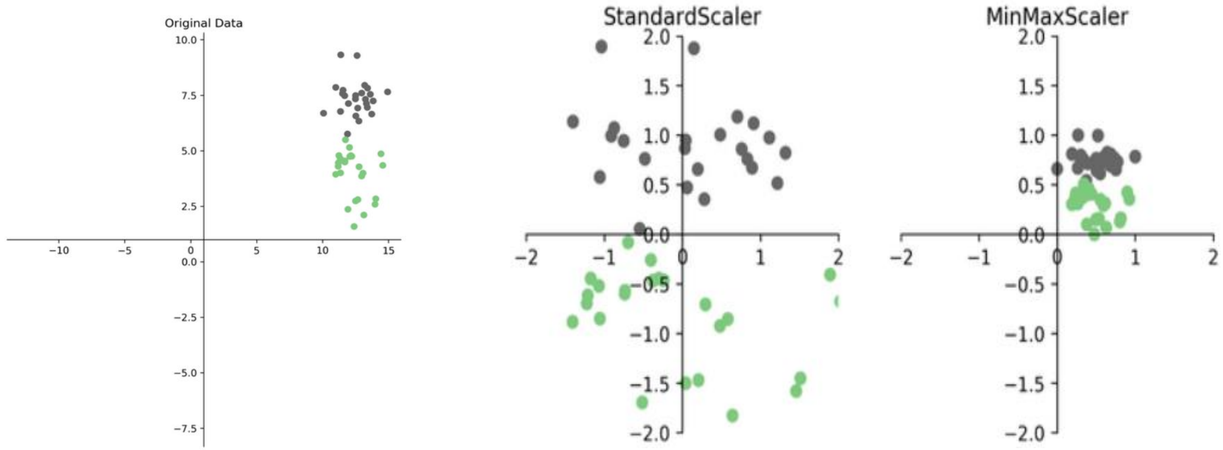


https://en.wikipedia.org/wiki/Log-normal_distribution#/

Model의 성능 개선 방법

- model driven vs **data driven**
  - data distribution
  - **data scale**
  - domain knowledge
    - binning
  - feature engineering
  - dimension reduction
  - train set vs test set vs real data

- Data scaling



https://amueller.github.io/aml/01-ml-workflow/03-preprocessing.html

Model의 성능 개선 방법

- model driven vs **data driven**
  - data distribution
  - data scale
  - **domain knowledge**
    - binning
  - feature engineering
  - dimension reduction
  - train set vs test set vs real data

- domain knowledge

단계	과정	결과값
원본	데이터	12, 28, 0, 16, 4, 18, 16, 26, 24
1	속성값 정렬	0, 4, 12, 16, 16, 18, 24, 26, 28
2-1	동일 간격 기반 평활화 Bin: [-, 10), [10, 20), [20, +)	0, 4 / 12, 16, 16, 18 / 24, 26, 29
2-2	동일 빈도 기반 평활화 Bin 밀도=3	0, 4, 12 / 16, 16, 18 / 24, 26, 29

Model의 성능 개선 방법

- model driven vs **data driven**
  - data distribution
  - data scale
  - domain knowledge
    - binning
  - **feature engineering**
  - dimension reduction
  - train set vs test set vs real data

- **feature engineering**
  - 주어진 Feature를 변형하여 Target값과 더 관련있게 만들기 위한 작업
  - Mutual Information, 수학공식 적용(예: 비율), count, 데이터 파싱, 데이터 결합, 그룹 변환, K-means, Encoding …

Model의 성능 개선 방법

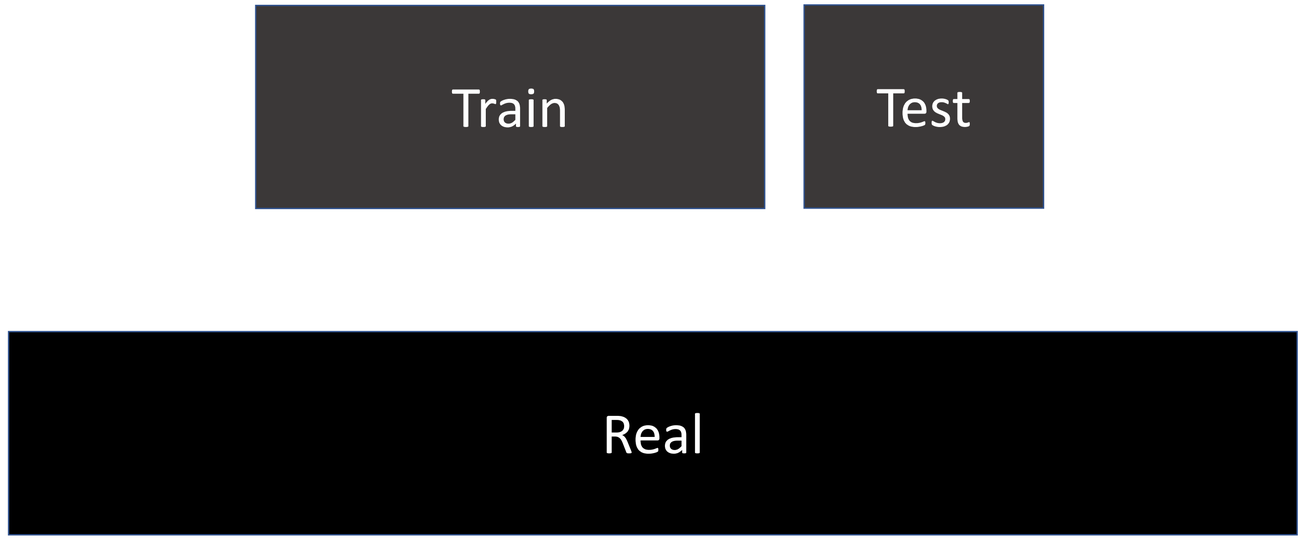
- model driven vs **data driven**
  - data distribution
  - data scale
  - domain knowledge
    - binning
  - feature engineering
  - **dimension reduction**
  - train set vs test set vs real data

- **dimension reduction**
  - feature selection
    - 특정 피처에 종속성이 강한 불필요한 피처는 제거하고, 데이터의 특징을 잘 나타내는 주요 피처만 선택
  - feature extraction
    - 피처를 함축적으로 더 잘 설명할 수 있는 또 다른 공간으로 매핑해서 추출

Model의 성능 개선 방법

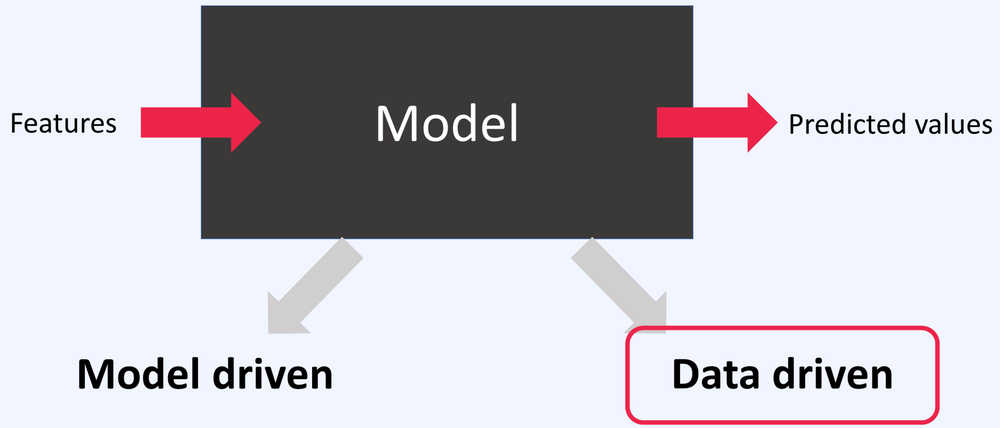
- model driven vs **data driven**
  - data distribution
  - data scale
  - domain knowledge
    - binning
  - feature engineering
  - dimension reduction
  - **train set vs test set vs real data**

- **train set vs test set vs real data**



Summary

- model driven vs **data driven**
  - data distribution
  - data scale
  - domain knowledge
    - binning
  - feature engineering
  - dimension reduction
  - data collection
  - train set vs test set vs real data



알고 있으면 쓸모 있는 AI 지식

시스템 관점에서의 인공지능의 이해

Machine Learning System

- Paper: Hidden Technical Debt in Machine Learning Systems, D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Franc¸ois Crespo, Dan Dennison, Google, In.

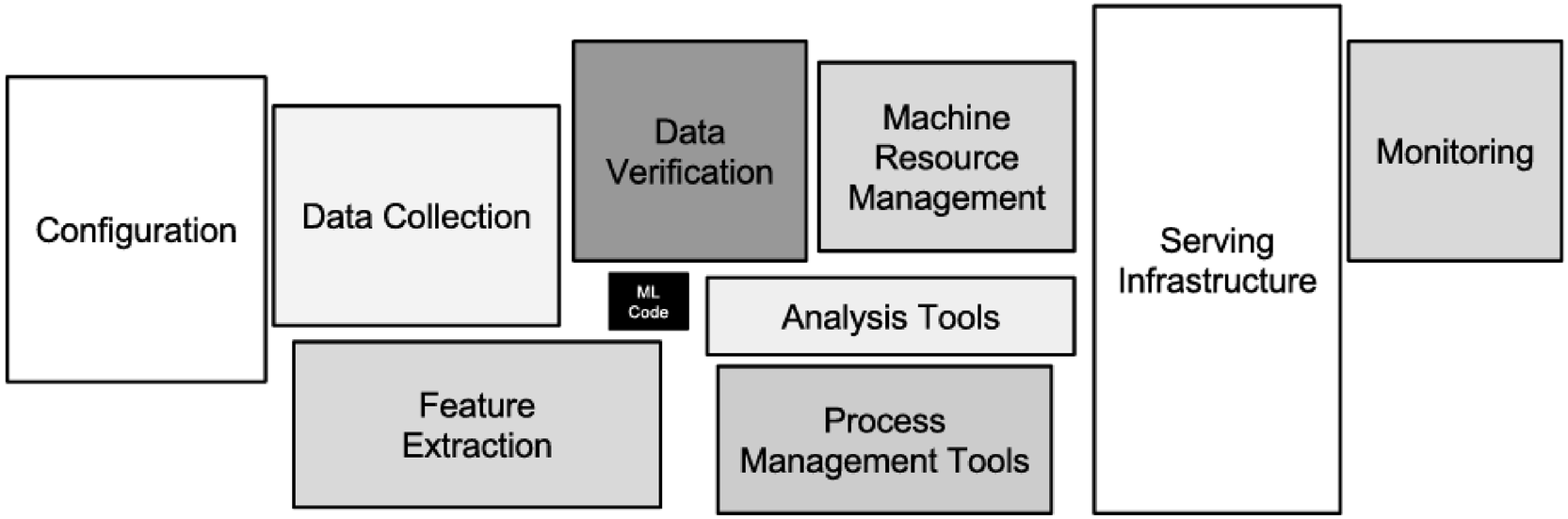


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Machine Learning Lifecycle

- 머신러닝으로 해결할 수 있는 문제인가?
- 데이터는 수집이 가능한가?
- 가용한 HW는?
- 가용한 인력은?
- 일정은?
- 평가 기준
- …

- 데이터 수집
- Labeling, annotation
- visualization
- cleansing
- …

- 모델 설계
- 학습
- 평가
- 성능 개선

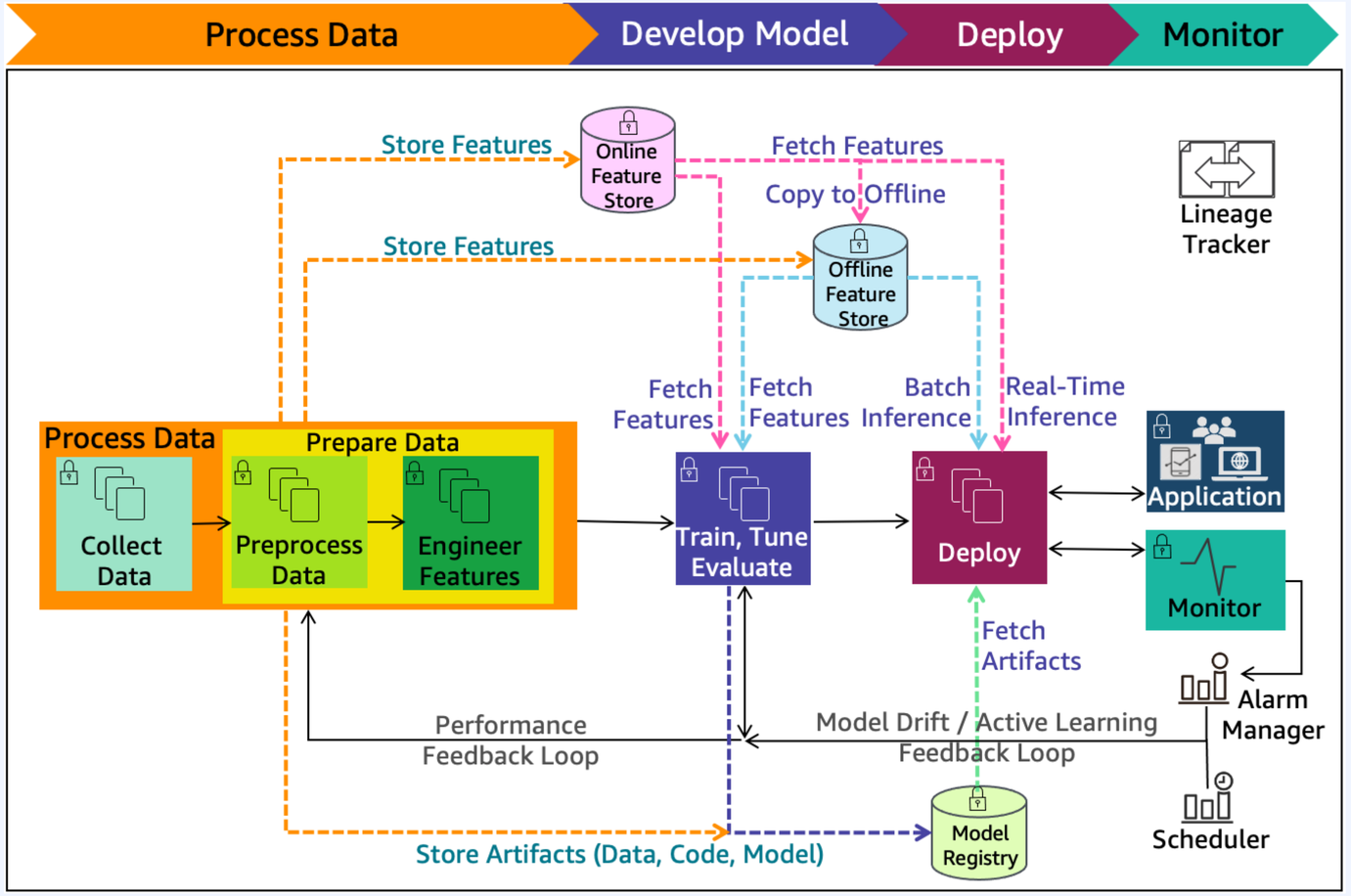
- 모델 베포
- 모니터링
- 모델 업데이트

Machine Learning Lifecycle

AI 프로젝트 구성원

AI 서비스 기획자
데이터 관리자
데이터 어노테이션 관리자
데이터 분석가
데이터 분석 툴 개발자
데이터 분석 검증 도구 개발자
데이터 어노테이션 도구 개발자
AI 모델러
AI 서비스 개발자
...

Machine Learning Architecture



https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/ml-lifecycle-architecture-diagram.html

Summary

➢ AI 프로젝트에서 숲을 봐야하는 이유

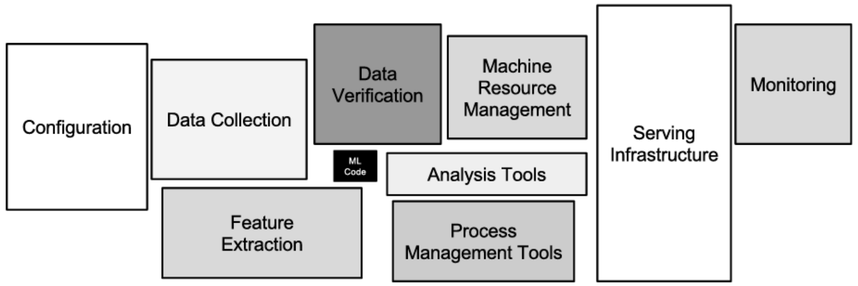


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

➢ Tech vs Governance

➢ Machine learning vs Data analysis