3. Structuring Machine Learning Projects

Week	Week1

Why ML Strategy

시스템 개선을 위한 아이디어가 많은데

하나 선택해서 잘 못 한다면 6개월동안 시간만 쓰고 성능 향상이 안되는 경우가 있다.

Orthogonalization = 직교화

하이퍼파라미터 튜닝할게 너무 많음

뭘 튜닝해야할지 잘 알아야 한다. → 이걸 **orthogonalization 직교화** 라고 함.

예시1) tv 디자이너가 손잡이 어떻게 달지 고민하는데 0.1*높이 + 0.3*넓이 - 1.7*사다리꼴 + 0.8*수평방향이미지 이런걸로 공식을 세워놓으면 디자이너는 손잡이에 대해 하나만 고민하면 됨.

예시2) 차 핸들, 브레이크, 엑셀로 이루어져잇는데

조이스틱으로 차를 조종한다고 생각하면

이것도 0.3*핸들각도 - 0.8*속도 이거랑 2*핸들각도 + 0.9*속도 식 두개로 조종한다고 해보자.

이러면 손잡이가 2개인 셈이라서 더 조종하기는 힘들듯 → 직교화 특성을 갖게 되면 실제로 제어하려는 부분에 이상적으로 맞추어진 직교 컨트롤을 사용하면 손잡이 조절이 더 쉬워짐.

지도 학습이 잘 이루어지려면 아래 4개가 잘 유지되어야 한다.

1. Fit training set well on cost function
    - bigger network
    - Adam 더 나은 최적화 알고리즘

2. Fit dev set well on cost function
    - 정규화 (다른 손잡이 세트)
3. Fit test set well on cost function
4. Performs well in real world

training set에서 잘 되는게 real world의 성능으로 이어지지 않으면 dev, test set가 올바르지 않거나 cost function이 잘 못 된것.

Single Number Evaluation Metric

분류기	Precision	Recall
A	95%	90%
B	98%	85%

Precision 정확도 : 전체에서 정답으로 판단한 것의 갯수

Recall 재현율 : 정답으로 판단한 것 중에서 실제로 정답인 거

두개는 trade-off 가 발생함. 둘 다 잘 나올수는 없다.

그래서 F1 score만들어서 평가

F1 score = $\frac{2}{\frac{1}{P}+\frac{1}{R}}$ = $\frac{2(P+R)}{PR}$ = 정확도와 재현율의 조화평균

Satisficing and Optimizing Metric

분류기에서 정확도와 러닝타임을 고려한다고 해보자.

아무리 정확하게 측정해도 러닝타임이 길면 안된다. 우리가 러닝타임을 최대 100ms로 기준을 잡자.

이렇게 기준을 잡는게 러닝 타임과 정확성을 절충하거나 모두 고려하는 합리적인 방법이 된다.

Train/Dev/Test Distributions

Train/Dev/Test 어떻게 설정하는지에 따라 프로젝트 속도가 달라질 수 있다.

dev set = development set or hold-out cross validation set

팀에게 목표 과녁을 설정하는 것. dev/test set을 설정하면 프로젝트를 빠르게 cycle 돌릴 수 있고.

강의 예시에서는 us,uk,europe, america 를 dev 세트로, india,china,asia,australia를 test 세트로 했는데 이렇게 하면 distribution이 달라서 좋은 성능으로 이어지지 않는다. 이러면 과녁을 바꿔야 함. 시간은 시간대로 쓰고

→ 모든 데이터를 무작위로 섞고 이걸 dev/test 세트에 반영하기 → 데이터가 같은 분포도에서 온 것.

Choose a dev set and test set to reflect data you expect to get in the future and considef important to do well on.

Size of dev and test sets

일반적으로 7:3, 6:2:2

데이터 많으면(100만개 이상) 98:1:1

Size of dev set

> Set your dev set to be big enough to detect differences in algorithm/models you are trying out

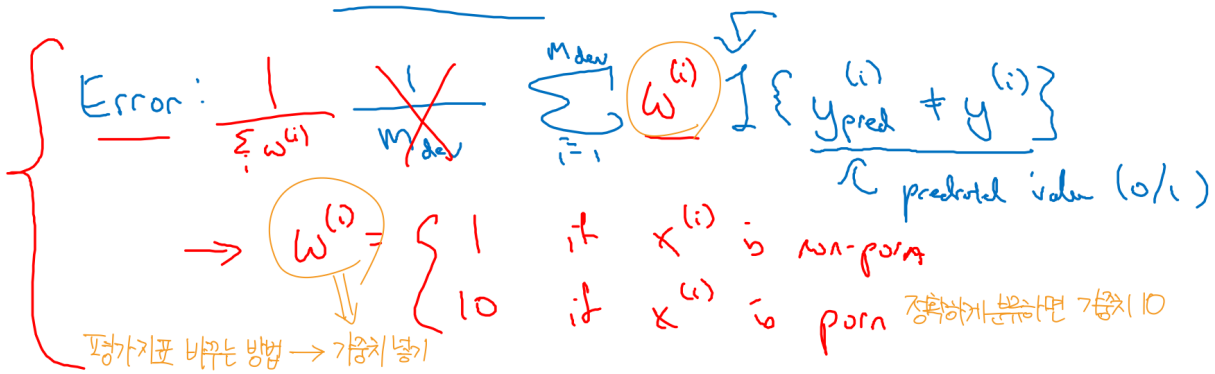
Size of test set

> Set your test set to be big enough to give high confidence in the overall performance of your system
>
> 데이터가 매우 크다면 test set이 없어도 된다(추천하진 않음)

When to Change Dev/Test Sets and Metrics?

프로젝트 진행 도중에 target을 잘못 설정했다는것을 깨달았다면 이때 target을 바꿔야한다.

이전 오류 지표가 만족스럽지 못한 경우 그 지표를 계속 유지하지 말고 어떤 것이 더 뛰어난 알고리즘인지에 대한 선호도를 잘 포착하는 새로운 지표 정의를 시도해 보세요



Orthogonalization for cat pictures: anti-porn

1. So far we've only discussed how to define a metric to evaluate classifiers
    - 목표 지점을 정한 뒤에 target
2. Worry separately about how to do well on this metric.
    - 어떻게 target을 맞출것인가.

하지만 유저들이 정말 원하는 것은 앱이 실제 올라가는 이미지를 잘 인식하는 것으로 이런 사진들은 엉성하거나, 흐릿하거나 프레임이 낮은 사진일 수도 있습니다

따라서 지표나 현 개발 세트 또는 개발 및 시험 세트 배포에서 좋은 결과가 나오더라도 앱이 실제로 잘 작동하지 않는다면 지표와 개발 시험 세트를 바꿔야 합니다.

다시 말해, 개발 시험 세트는 고품질 이미지를 다뤘는데 앱은 저품질 이미지를 다뤄야 해서 실제 평가 결과가 앱 성능을 제대로 예측하지 못한다면 실제 대상 데이터가 잘 반영될 수 있도록 개발 시험 세트를 바꿀 때입니다.

그래서 실제 상황의 데이터가 더 잘 반영되도록 해야 합니다.

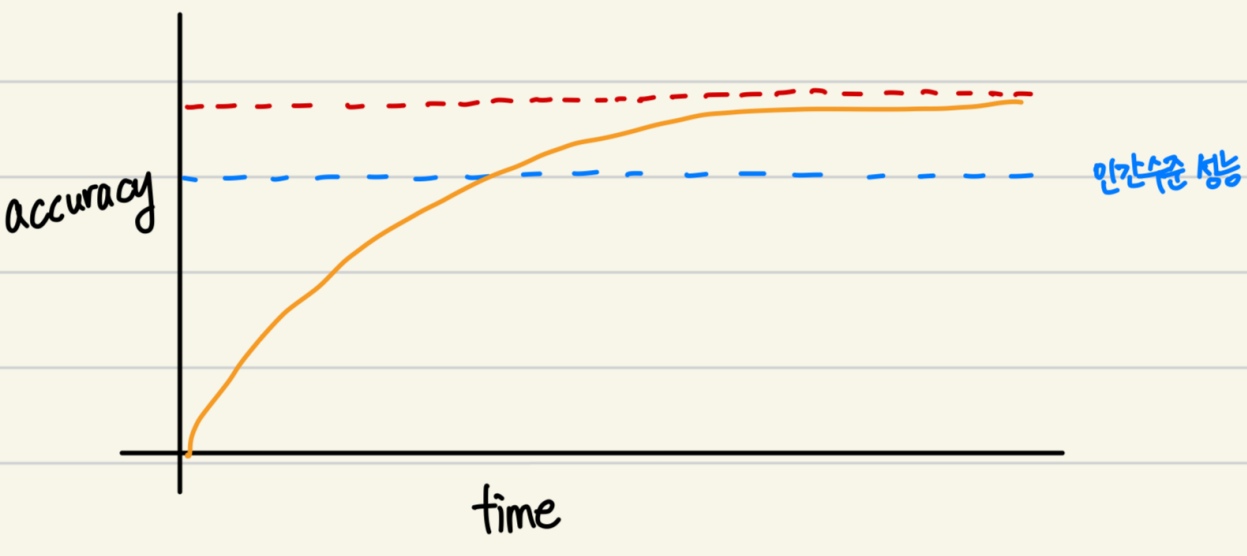
하지만 전반적으로 따라야 할 방향은 만약 현재 평가 중인 지표와 데이터가 실제 상황의 데이터에 대응되지 않는다면 지표 및/또는 개발/시험 세트를 바꿔 알고리즘을 실제 상황에 잘 대응시키는 것입니다.

제가 추천 드리는 방법은 완벽한 평가 지표와 개발 세트를 정의할 수 없더라도 이를 대강이라도 빠르게 설정해 팀의 반복 수행 속도를 높이는 것입니다.

나중에 더 나은 옵션이 있다면 그 때 가서 바꿔도 전혀 늦지 않습니다. 하지만 대부분의 팀에게 드리는 조언은 평가 지표나 개발 셋업 없이 너무 오래 진행하지 마시라는 겁니다 이렇게 하면 속

도가 늦춰지고 팀 생산성에 문제가 생기고 알고리즘 개선 기회가 저해될 수 있습니다.

Why Human-level Performance?



human-level을 뛰어넘으면서 진행속도나 정확도가 정체되기 시작.

알고리즘을 계속 훈련시키면서 더 많은 데이터를 더 큰 모델에 사용할 수는 있겠지만 성능은 이론적인 한계점에 다가갈 뿐 절대로 도달할 수 없다. ← Bayes optimal error 베이즈 최적 오류

human-level을 넘으면 진행속도가 떨어지는 이유

1. human-level과 베이즈 최적 오류 지점이 큰 차이가 없기 때문
2. human-level을 달성할때까지는 여러 도구를 이용해 성능을 개발하는데 그 지점을 넘으면 도구들을 못 써서

Avoidable Bias

human-level error는 베이즈 오류, 최적의 오류율을 추정하는데 도움이 될 수 있다.

베이즈오류와 훈련 오류 사이의 차이를 나타내는 'avoidable bias'와 훈련 오류와 개발 오류 사이의 차이를 나타내는 "bias"

Understanding Human-level Performance

?

인간이 이미 잘 수행하는 업무에 대해 인간 수준 오류의 추정치를 가지고 있는 경우 편향와 편차를 이해하기 위해서는 인간수준 오류의 프록시 또는 베이즈 오류의 추정값을 사용할 수 있습니다

베이즈 오류에 대한 추정치를 더 정확히 알면 회피가능 편향과 분산을 더 정확히 추정할 수 있습니다 따라서 편향 감소에 집중할지, 또는 분산 감소에 집중할지 적절한 결정을 내려야 합니다

복습하자면, 인간 수준 성능의 추정치를 알면 베이즈 오류의 예상값을 알 수 있습니다 그 다음 알고리즘에서 편향값을 줄일지 분산값을 줄일지 결정할 수 있습니다

Surpassing Human-level Performance

구조화된 데이터들은 ml이 더 잘 처리함.

인간은 자연적 인식을 더 잘함. 근데 요새는 nlp cv medical에서 인간보다 더 좋은 성능을 냄

Improving your Model Performance

1. You can fit the training set pretty well
2. The training set performance generalizes pretty well to the dev/test set.

Reducing (avoidable) bias and variance

- Train bigger model
- Train longer/better ooptimization algorithms
    - momentum, RMSProp, Adam
- NN architecture/hyperparameters search
- More data

- Regularization
  - L2, dropout, data augmentation
- NN architecture/hyperparameters search