AI 프로그래밍

ANN training과 Hyper Parameter

ANN Hyper Parameter

01

ANN Hyper Parameter

- Parameter
  - ANN의 weight와 bias
  - Training을 통해 최적의 값을 찾을 수 있음

- Hyper parameter
  - ANN의 parameter를 제외한 요소(learning rate 등)를 hyper parameter라 부름
    - Hyper parameter는 ANN의 training에 큰 영향을 끼침
    - 현재까지는 적절한 hyper parameter를 찾는 정형화된 방법은 거의 없음
      - 상용 ANN 시스템의 경우 여러 hyper parameter 조합으로 training 수행, 최적의 조합을 찾아 hyper parameter를 tuning함

ANN Hyper Parameter

- Hyper parameter의 종류
  - ANN architecture
    - Hidden layer 개수
    - Parameter 개수
    - Activation function
    - 기타 등등
  - Data 전처리
    - Data augmentation
    - Data normalization
    - 기타 등등
  - Training 방법
    - Learning rate 및 optimizer
    - Iteration 및 epoch
    - Mini-batch 크기
    - Parameter Initialization
    - Parameter regularization
    - 기타 등등

02

Training 시간과 Hyper parameter

Training 시간과 Hyper parameter

- Training 시간
  - Loss의 local minima를 빠르게 찾아 수렴하게 하려면 hyper parameter를 잘 설정해야 함

- Training 시간에 영향을 주는 hyper parameter
  - Training 방법
    - Learning rate 및 Optimizer
    - Iteration 및 epoch

Training 시간과 Hyper parameter

- Training 시간에 영향을 주는 hyper parameter
  - Learning rate 및 optimizer
    - Learning rate가 너무 작으면
      - False local minima에 갇히기 쉬움
      - Local minima에 도달하기까지 너무 오래 걸림
    - Learning rate가 너무 크면
      - Parameter가 빨리 변해 training이 빨라짐
      - Global minima 근처에서 진동이 발생하기 쉬움
        - Parameter가 수렴하지 못해 training이 오래 걸릴 수 있음
    - 최신 optimizer를 통해 learning rate의 의존도를 낮출 수 있음

Training 시간과 Hyper parameter

- Training 시간에 영향을 주는 hyper parameter
  - Iteration과 epoch
    - Training의 최대 반복 횟수를 지정
    - 횟수가 너무 적으면 local minima를 찾기 전에 training이 끝남
  - Iteration과 epoch의 차이점
    - Iteration : Parameter를 수정한 횟수
    - Epoch : Dataset의 모든 데이터를 한 번씩 training한 횟수
    - Dataset이 m개의 mini-batch로 이루어져 있을 때 모든 mini-batch를 다 training하면, iteration은 m번, epoch는 1번이 됨

03

ANN의 정확성과 Hyper parameter

ANN의 정확성과 Hyper parameter

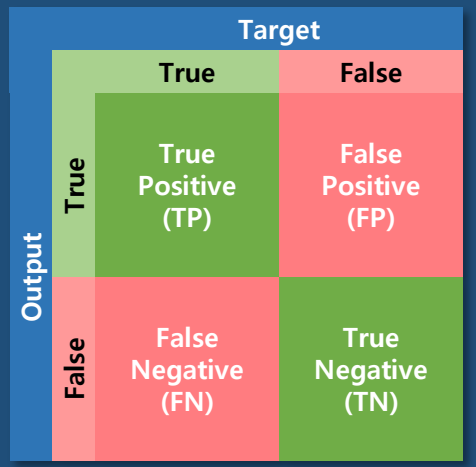
- ANN의 output layer에 따른 정확성 평가
  - Score
    - Output layer가 regression layer일 때 사용하는 평가 방법
    - Output 수치가 target 수치와 얼마나 비슷한지 평가
  - Accuracy
    - Output layer가 binary classifier 또는 softmax layer일 때 사용하는 평가 방법
    - 각 node 또는 전체 선택지에서 얼마나 정답을 많이 맞췄는지 평가

ANN의 정확성과 Hyper parameter

- Score
  - MAE (Mean Absolute Error)
    - $\frac{1}{n}\sum_i^n \|\vec{t_i} - \vec{y_i}\|$
  - MSE (Mean Squared Error)
    - $\frac{1}{n}\sum_i^n \|\vec{t_i} - \vec{y_i}\|^2$
  - RMSE (Root Mean Squared Error)
    - $\frac{1}{n}\sum_i^n \sqrt{\|\vec{t_i} - \vec{y_i}\|^2}$

ANN의 정확성과 Hyper parameter

- Accuracy
  - Accuracy (정확도)
    - 모든 경우 중 true/false를 정확히 예측한 비율
    - $\frac{TP+TN}{TP+FP+FN+TN}$
  - Precision (정밀도)
    - True라고 예측한 것 중 실제 true인 비율
    - $\frac{TP}{TP+FP}$
  - Recall (재현도)
    - True인 데이터 중 true를 맞춘 비율
    - $\frac{TP}{TP+FN}$



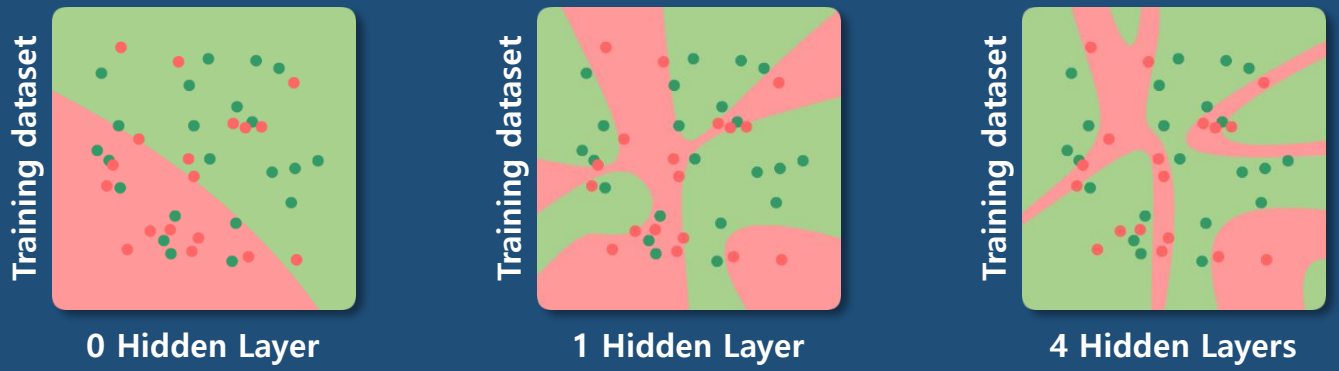
ANN의 정확성과 Hyper parameter

- Score와 accuracy를 높이기 위해선 hyper parameter를 잘 설정해야 함

- Score와 accuracy에 영향을 주는 hyper parameter
  - ANN architecture
    - Hidden layer 개수
    - Parameter 개수

ANN의 정확성과 Hyper parameter

- Score와 accuracy에 영향을 주는 hyper parameter
  - Hidden layer 개수



- Hidden layer가 많을 수록 decision boundary가 정밀해져 정확성이 증가함
- 단, hidden layer 수가 너무 많아지면 더 이상 정확성이 증가하지 않음

ANN의 정확성과 Hyper parameter

- Score와 accuracy에 영향을 주는 hyper parameter
  - Parameter 개수
    - ANN을 이루는 weight와 bias의 개수
    - ANN의 parameter가 늘어나는 경우
      - Hidden layer 추가
      - Layer의 width, 즉 node수가 증가
      - 연결되지 않았던 node 사이에 edge 추가
    - Parameter 개수가 많아질수록 feature가 다양해져 정확성이 증가함
    - 단, parameter 개수가 너무 많아지면 더 이상 정확성이 증가하지 않음

04

Generalization과 Hyper parameter

Generalization과 Hyper parameter

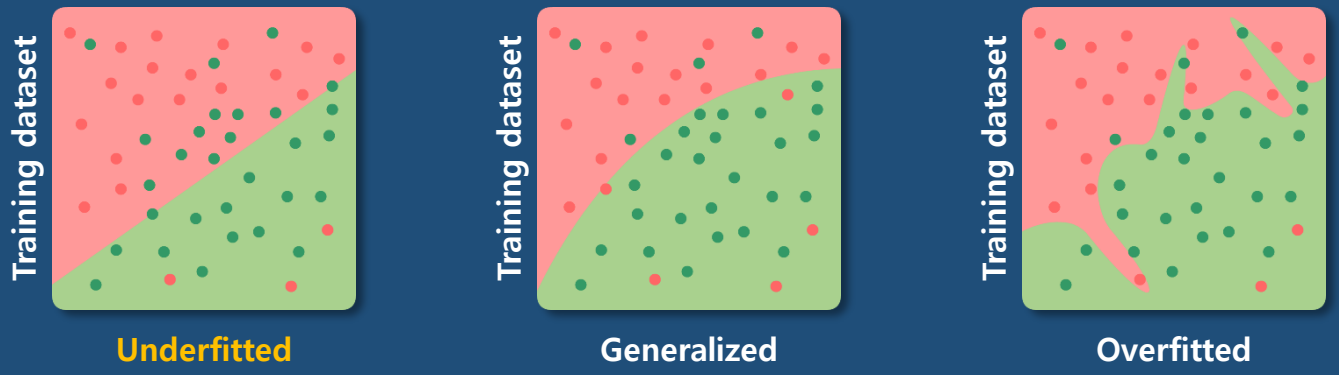
- Generalization
  - Training이 끝난 ANN이 실제 문제에서도 좋은 성능을 발휘할 수 있어야 함
  - Training 과정에서 보지 못한 데이터에서도 잘 작동하는 것을 generalization이라 함

  - Generalization 성능을 보기 위해, training dataset과 test dataset을 분리함
    - Training dataset : ANN training에 활용하는 dataset
    - Test dataset : Training이 끝난 후 generalization 시험을 위한 dataset

  - Generalization이 잘 된 ANN은 training 및 test dataset에 대한 정확성이 모두 높아야 함
    - 두 dataset에 대한 정확성이 모두 낮은 경우를 underfitting이라 함
    - Training dataset에 대한 정확성은 높지만, test dataset에 대한 정확성은 낮은 경우를 overfitting이라 함

Generalization과 Hyper parameter

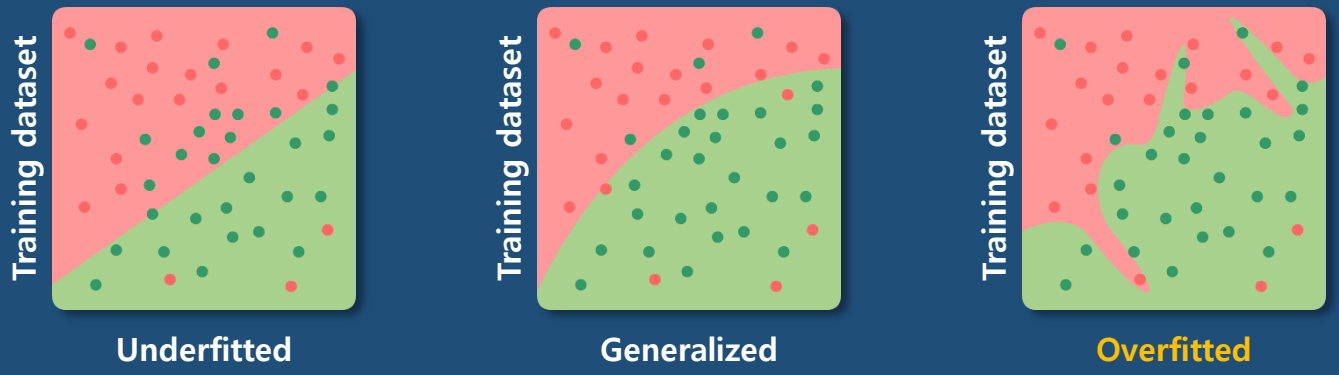
- Generalization
  - Underfitting



- Training dataset에 대한 정확성도 낮음
  - Dataset의 inlier도 맞추지 못함
- Training 시간이 부족하거나 ANN architecture가 너무 단순할 때 발생

Generalization과 Hyper parameter

- Generalization
  - Overfitting



- Training dataset에 대한 정확성이 너무 높음
  - Dataset의 outlier까지 맞추기 위해 decision boundary를 overfitting함
- Training 시간이 너무 길거나 ANN architecture가 너무 복잡할 때 발생

Generalization과 Hyper parameter

- Generalization
  - Overfitting을 방지해야 하는 이유
    - Training dataset은 현실 세계(real-world) 모집단 dataset의 일부일 수 밖에 없음
      - ANN을 현실 세계에서 활용했을 때, training 때 보지 못했던 상황을 맞이하기 쉬움
      - Training dataset에 overfitting하면, 실제 상황에서 틀린 판단을 내릴 확률이 높음
    - 현실 세계에서 sampling한 모든 dataset에는 outlier가 포함될 확률이 높음
      - Outlier까지 맞추기 위해 overfitting하면 일반적인 경향성을 놓치게 됨
    - 현실 세계에서는 training dataset이 완벽한 정답이 아님을 염두해 두어야 함

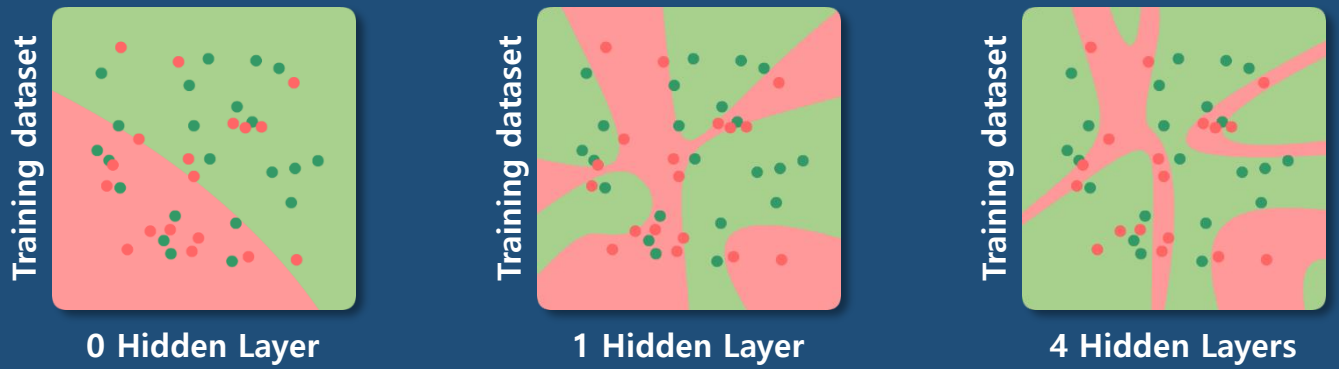
Generalization과 Hyper parameter

- Generalization
  - ANN의 generalization 성능을 높이기 위해선 hyper parameter를 잘 설정해야 함

- Generalization에 영향을 주는 hyper parameter
  - ANN architecture
    - Hidden layer 개수
    - Parameter 개수
  - Data 전처리
    - Data augmentation
  - Training 방법
    - Iteration 및 epoch
    - Parameter regularization

Generalization과 Hyper parameter

- Generalization에 영향을 주는 hyper parameter
  - Hidden layer 개수



- Hidden layer가 많을 수록 ANN architecture가 복잡해짐
- Hidden layer가 적으면 underfitting, 많으면 overfitting이 일어남

Generalization과 Hyper parameter

- Generalization에 영향을 주는 hyper parameter
  - Parameter 개수
    - Parameter가 많을 수록 ANN architecture가 복잡해짐
    - Parameter가 적으면 underfitting, 많으면 overfitting을 일으킴

Generalization과 Hyper parameter

- Generalization에 영향을 주는 hyper parameter
  - Data augmentation
    - Dataset의 데이터 양 충분하지 않거나, 데이터의 다양성이 부족할 때 이를 해결하는 방법

- 기존 데이터에 각종 변형을 가해 다양한 데이터를 추가함
- 데이터의 다양성을 늘리면 데이터간 관계성이 줄어 overfitting을 방지할 수 있음

Generalization과 Hyper parameter

- Generalization에 영향을 주는 hyper parameter
  - Iteration 및 epoch
    - Training이 길어지면 parameter가 training dataset의 outlier에도 fitting할 시간을 줌
    - Parameter가 충분히 작은 값으로 수렴하지 않아도 training을 중단해 overfitting을 방지

Generalization과 Hyper parameter

- Generalization에 영향을 주는 hyper parameter
  - Parameter regularization
    - Parameter의 절대값이 너무 커지지 않도록 제한하는 방법
    - Weight decay라 하며, weight decay rate($\lambda < 1$)를 통해 제한 정도를 조절할 수 있음

$$L(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{\Theta}) = L(\boldsymbol{t}, \boldsymbol{y}) + \frac{1}{2}\lambda \|\boldsymbol{\Theta}_t\|^2$$

$$\boldsymbol{\Theta}_{t+1} = (1 - \gamma\lambda)\boldsymbol{\Theta}_t - \gamma \nabla \boldsymbol{\Theta}_t$$

    - 일반적으로 parameter는 0과 가까운 값으로 초기화
    - Parameter의 크기를 제한하면 parameter가 너무 많이 training되지 않게 됨
    - 결과적으로 overfitting을 방지하는 효과가 있음

04

Gradient와 Hyper parameter

Gradient와 Hyper parameter

- Training 중 gradient 크기의 변화
  - Gradient explosion
    - Training 도중 gradient의 크기가 너무 커져버리는 현상
    - Parameter 진동이 심해져 local minima을 전혀 찾지 못함
    - Parameter에 overflow 또는 underflow가 발생해 training이 중단 될 수 있음
  - Gradient vanishing
    - Training 도중 gradient가 0에 수렴해 사라지는 현상
    - Training을 계속 진행해도 parameter를 더 이상 수정하지 못함

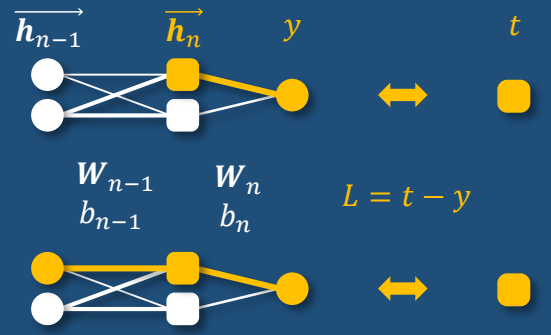
Gradient와 Hyper parameter

- Training 중 gradient 크기의 변화
  - Gradient explosion과 vanishing을 방지하기 위해선 hyper parameter를 잘 설정해야 함

- Gradient에 영향을 주는 hyper parameter
  - ANN architecture
    - Hidden layer 개수
  - Data 전처리
    - Data normalization
  - Training 방법
    - Learning rate
    - Parameter Initialization
    - Parameter regularization

Gradient와 Hyper parameter

- Gradient에 영향을 주는 hyper parameter
  - Hidden layer 개수
    - $\frac{\delta L}{\delta w_{n,1}} = \frac{\delta L}{\delta y}\frac{\delta y}{\delta w_{n,1}} = -1 \cdot h_{n,1}$
    - $\frac{\delta L}{\delta w_{n-1,1,1}} = \frac{\delta L}{\delta y}\frac{\delta y}{\delta h_{n,1}}\frac{\delta h_{n,1}}{\delta w_{n-1,1,1}} = -1 \cdot w_{n,1} \cdot (1 - \tanh(g_{n,1})^2) \cdot h_{n-1,1}$



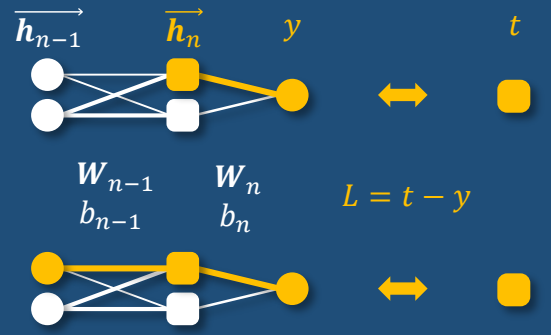
- Input layer로 갈 수록 gradient에 앞(forward) layer의 weight값($w_{n,1} < 1$)이 곱해짐
- 따라서 Hidden layer가 너무 많은 경우 input layer에 가까워질 수록 gradient가 0에 수렴함
- 다행히, 현재는 ResNet, DepthNet, ContextNet 등 hidden layer를 많이 쌓아도 gradient vanishing을 방지하는 방법이 많이 발표됨

Gradient와 Hyper parameter

- Gradient에 영향을 주는 hyper parameter
  - Data normalization

$$\frac{\delta L}{\delta w_{n,1}} = \frac{\delta L}{\delta y}\frac{\delta y}{\delta w_{n,1}} = -1 \cdot h_{n,1}$$

$$\frac{\delta L}{\delta w_{n-1,1,1}} = \frac{\delta L}{\delta y}\frac{\delta y}{\delta h_{n,1}}\frac{\delta h_{n,1}}{\delta w_{n-1,1,1}} = -1 \cdot w_{n,1} \cdot (1 - \tanh(g_{n,1})^2) \cdot h_{n-1,1}$$



- Gradient에는 node의 값을 곱함
- 따라서 데이터의 input과 target의 크기가 큰 경우 gradient의 크기가 커짐
- 데이터 값의 크기를 적당한 크기로 정규화하는 것을 data normalization이라 함
  - 예시: $X \in [0,255]$ 인 image 데이터를, 각 픽셀의 값을 255로 나누어 $X \in [0,1]$로 정규화

Gradient와 Hyper parameter

- Gradient에 영향을 주는 hyper parameter
  - Learning rate
    - Learning rate가 큰 경우
      - 처음 0에 가까이 초기화 된 parameter들이 쉽게 큰 값으로 변함
      - Parameter의 크기가 커지면 각 node의 값이 커져 gradient explosion이 발생하기 쉬움
    - Learning rate가 작은 경우
      - Parameter의 변화량이 더 적어지기 때문에 gradient vanishing 현상이 더 쉽게 발생

Gradient와 Hyper parameter

- Gradient에 영향을 주는 hyper parameter
  - Parameter Initialization
    - Parameter의 크기를 작게 초기화하여 gradient explosion이 일어날 확률을 줄임
      - 일반적으로 bias는 모두 0 으로 초기화
      - 일반적으로 weight는 layer에 node가 n개 있을 때, $\left[-\frac{1}{n^2}, \frac{1}{n^2}\right]$중 임의의 값으로 설정
        - Weight가 1보다 충분히 작은 값이 되어 gradient explosion을 예방
        - Weight를 0으로 초기화할 경우 gradient vanishing 발생
      - 임의의 값을 추출하는 방법으로 Xavier initializer와 He initializer 등을 활용함
  - Parameter regularization
    - Weight decay를 통해 parameter의 크기가 커지는 것을 막아 gradient explosion을 방지함
    - Weight decay가 너무 크면 parameter가 너무 작아져 gradient vanishing의 원인이 됨

감사합니다

Fin