A Gradient Boosting Approach for Training Convolutional and Deep Neural Networks

M2023082 김민형

Index

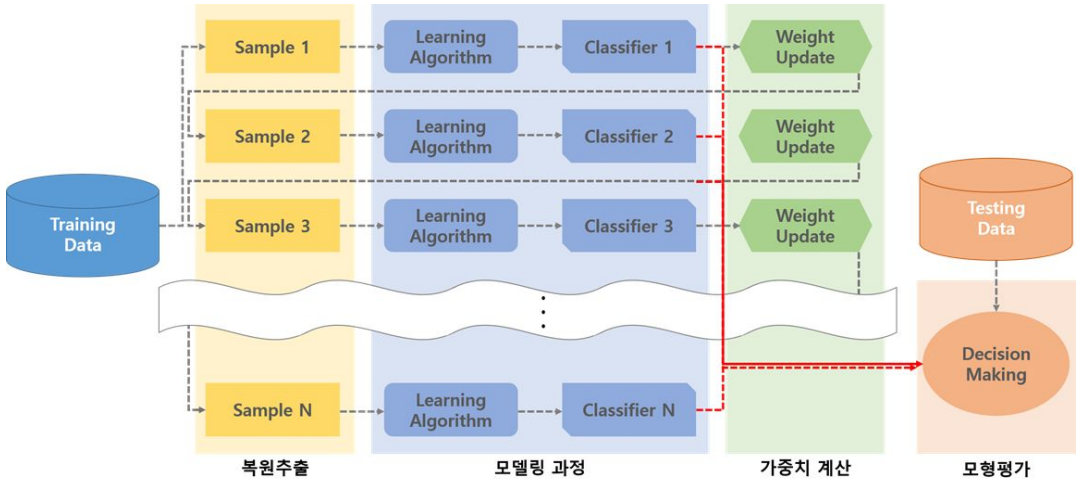
- Introduction
- Boosting
- GBM
- Proposed Method
- Experiments
- Conclusion

Introduction

- 기존 연구 방향 - Gradient Descent를 이용한 End-to-End 학습
- Gradient Boosting을 통한 단계적 학습
- 저자는 NN에 GB 방법론을 적용하고 싶었음
- TabNet, GBNN, DNF-net 등의 기존 연구가 존재하지만, 성능이나 모델 크기 등의 문제가 존재했음
- 저자가 제시하는 방법 GB-CNN, GB-DNN은 모델 크기와 상관 없는 학습 방법론

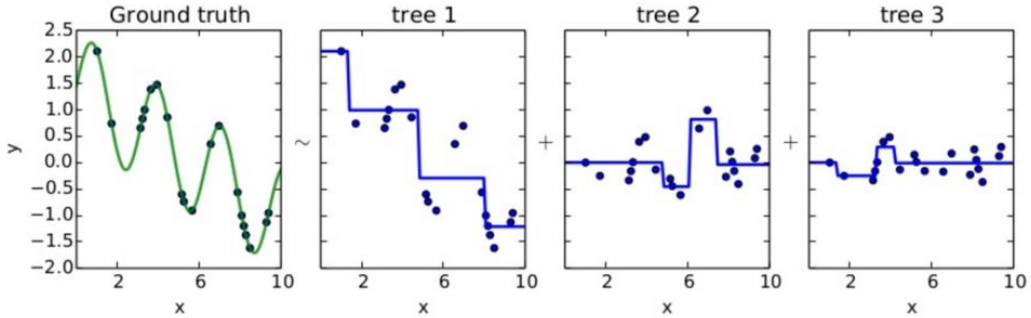
Boosting

- 머신러닝 앙상블 기법 중 하나로, Sequential한 Weak Learner를 여러 개 결합하여 Regression/Classification 성능을 높히는 알고리즘
- 대표적으로, Adaboost, GBM, XGBoost, LightGBM 등이 있음



Gradient Boosting

- Sequential한 Weak Learner들을 Residual을 줄이는 방향으로 결합
- 잔차 : $y_i - f(x_i)$
- 목적함수 $j(y_i, f(x_i)) = \frac{1}{2}(y_i - f(x_i))^2$ 를 최소화하고 얻어진 잔차를 다음 Weak Learner가 이를 예측하도록 함
- 최적화는 Gradient Descent로 진행

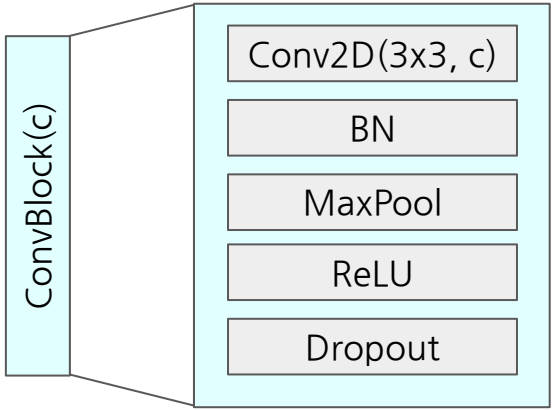


Proposed Method

- GB-CNN
- GB-DNN

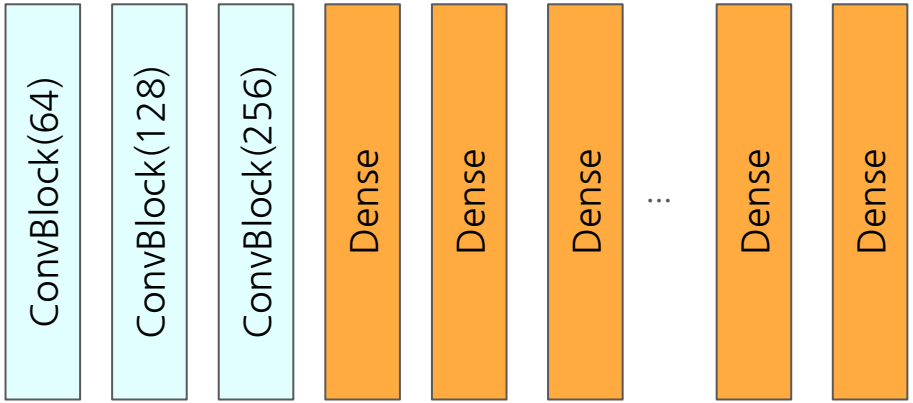
GB-CNN

- GB-CNN을 학습하는 방법은 다양한 구조의 CNN 네트워크에 적용할 수 있으나, 여기선 아래 설명할 구조의 모델을 통해 분류 문제를 처리함
- Convolution Block - 컨볼루션, 배치 정규화, 최대 풀링, 활성화, 드롭아웃을 순서대로 적용하는 하나의 블럭임



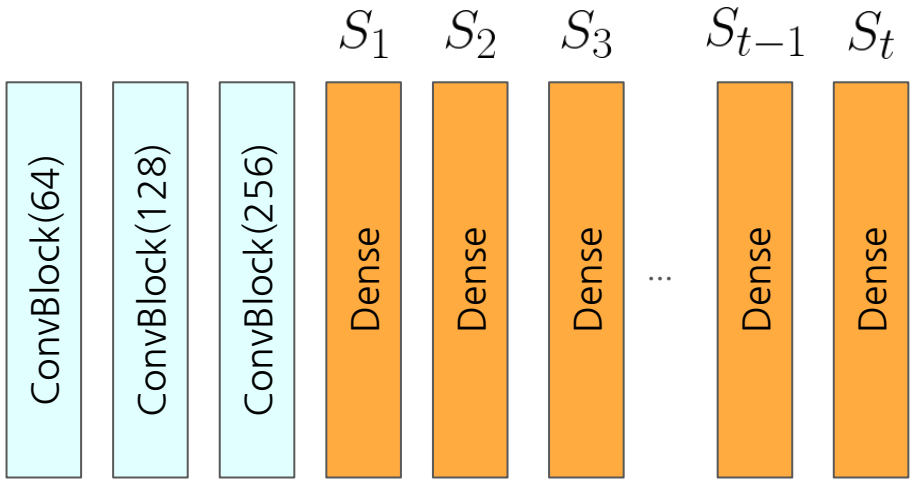
GB-CNN

- GB-CNN은 3개의 블럭으로 구성되며, 맨 앞부터 64채널, 128채널, 256채널의 컨볼루션을 갖는다
- 3개의 블럭 이후에 Flatten이 적용되며, 이후 Dense Layer를 쌓은 구조가 GB-CNN의 구조이다.



GB-CNN

- GB-CNN의 출력은 Classification을 위한 원시 값을 출력한다
- 이 값은 각 Dense Layer 출력의 합산으로 구해진다
- $F_t(\mathbf{X}_i) = F_{t-1}(\mathbf{X}_i) + \boldsymbol{\rho}_t S_t(\mathbf{X}_i)$

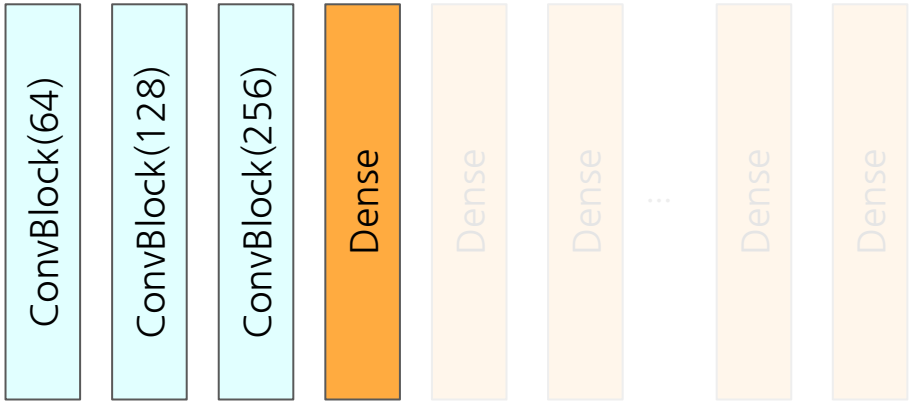


Learning Algorithm

1. CNN Blocks를 거친 Representation은 하나의 Dense Layer를 거쳐 분류를 위한 벡터를 생성한다. 이는 softmax를 통해 확률로 바뀌며, Cross-Entropy를 통해 CNN과 Dense 레이어를 학습한다.
2. 이제, 앞서 학습한 Dense 레이어의 가중치가 고정되며, 그 뒤에 새로운 Dense Layer를 붙여 다시 학습한다.
3. Loss는 이전 Dense 레이어의 출력과 현재 Dense Layer의 출력의 합산으로 정의된다

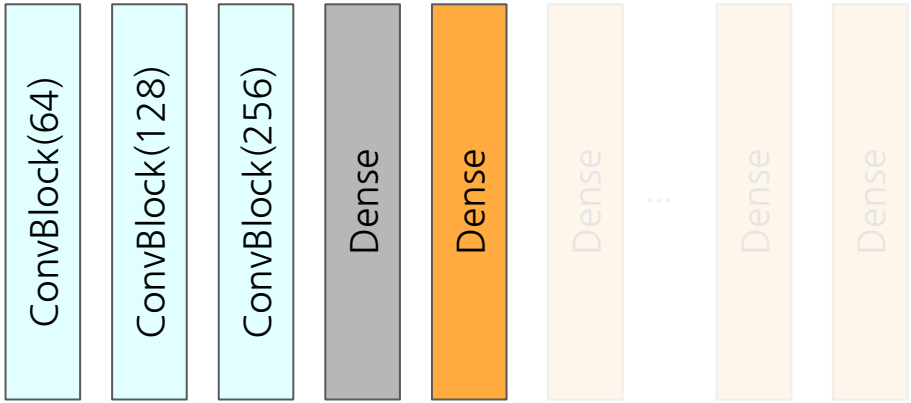
Learning Algorithm

- 초기 상태 - 하나의 Dense Layer로 동작하는 Classifier 학습



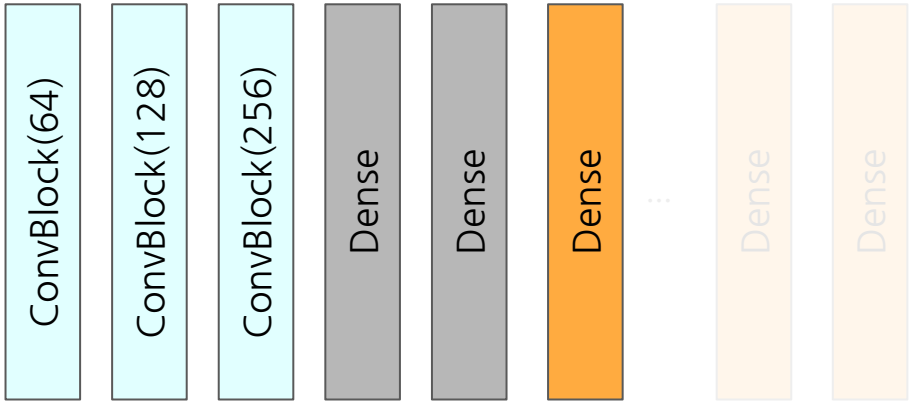
Learning Algorithm

- 학습이 완료된 후, 새로운 Dense Layer를 붙임
- 이때, 이전 Dense Layer는 가중치가 Freeze됨 (학습 X)



Learning Algorithm

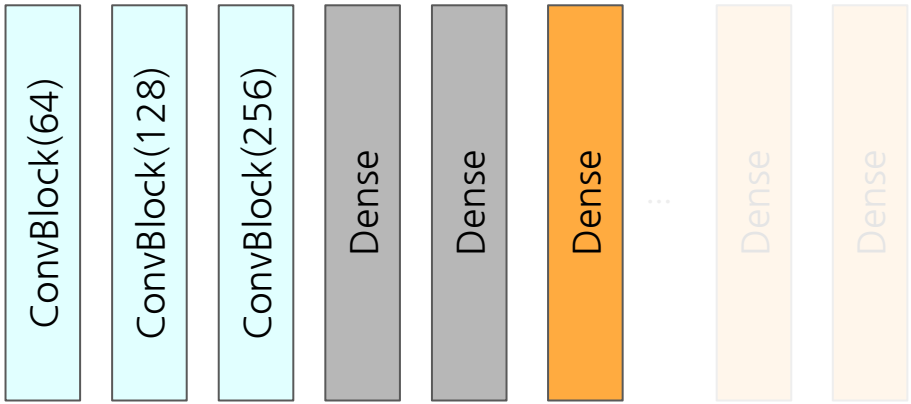
- 이 프로세스를 반복하는 것이 Boosting 과정임
- GB-CNN에서, ConvBlock은 Freeze되지 않고 Fine-tuning됨



Learning Algorithm

- Loss는 각 Boosting 단계까지 붙은 Dense Layer의 출력 합을 Softmax와 Cross-Entropy로 학습 (for classification)

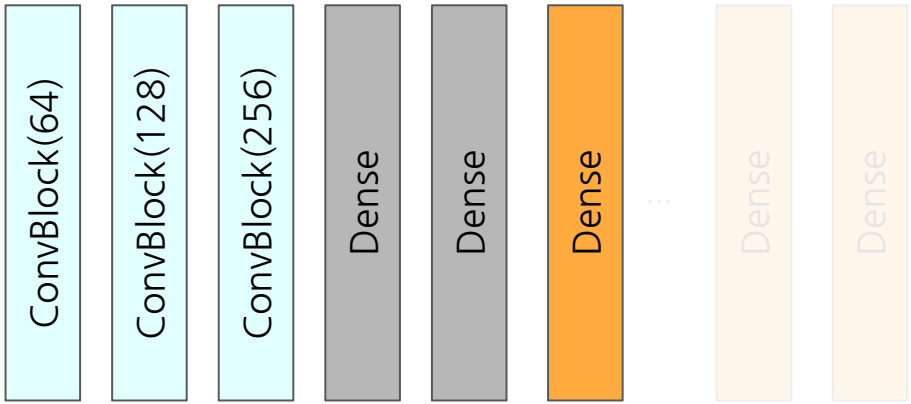
$$\ell(\mathbf{y}_i, \mathbf{P}_i) = -\sum_{k=0}^{K} y_i^k \log p_i^k \qquad p^k(\cdot) = \frac{\exp(F^k(\cdot))}{\sum_{l=1}^{K} \exp(F^l(\cdot))}$$



Learning Algorithm

- 모델 출력 합을 계산할때 사용된 앞서 본 모델 수식에서 ρ는 각 Dense Layer의 출력을 Class 마다 가중치를 주기 위해 사용되는 것으로 파악됨

$$F_t(\mathbf{X}_i) = F_{t-1}(\mathbf{X}_i) + \boldsymbol{\rho}_t S_t(\mathbf{X}_i)$$



GB-DNN

- GB-DNN은 GB-CNN에서 Convolution Block들을 제거하고, Dense Layer의 입력으로 Tabular Data를 직접 주는 방식임

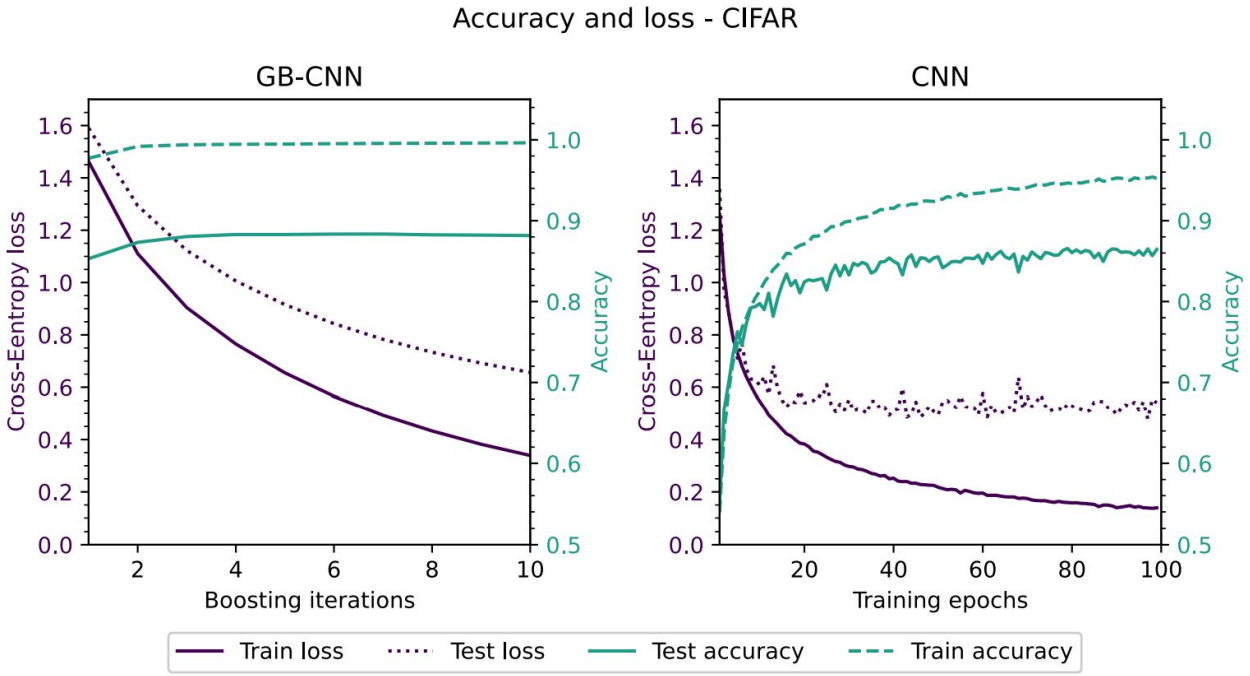
Experiments

- 모델 학습 시 10-fold cross validation을 통해 early stopping을 적용했음
- GB-CNN의 수렴은 2~3 Boosting 내에 이루어졌음 -> 최대 부스팅 수 10으로 제한

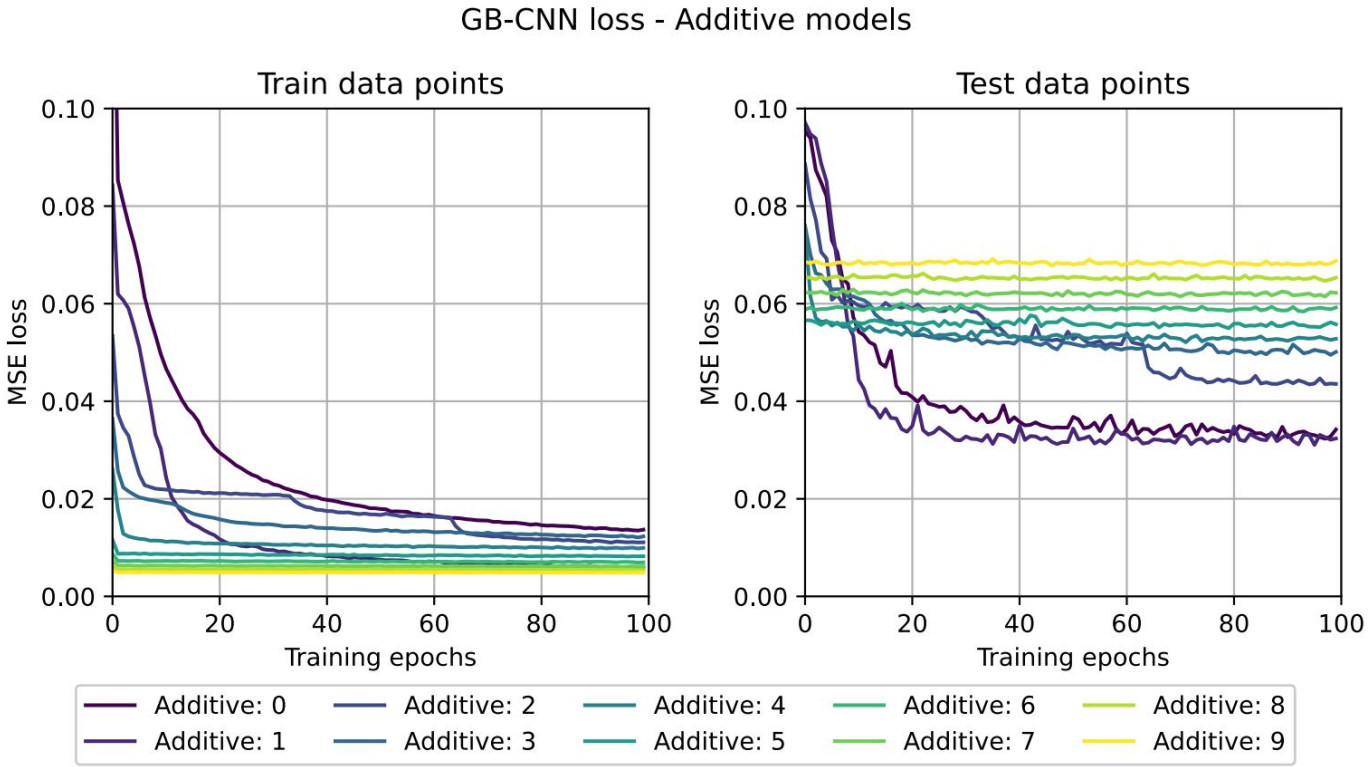
Name	**train/test**	$P_H \times P_W$	K	Ch
MNIST [28]	60,000/10,000	28×28	10	1
CIFAR-10 [29]	50,000/10,000	32×32	10	3
Rice varieties [30]	56,250/18,750	32×32	5	3
Fashion-MNIST [31]	60,000/10,000	28×28	10	1
Kuzushiji-MNIST [32]	60,000/10,000	28×28	10	1
MNIST-Corrupted [32]	60,000/10,000	28×28	10	1
Rock-Paper-Scissors [33]	2,520/370	32×32	3	3

Name	**instances**	**Features**	**class labels**
Digits [34]	1797	64	10
Ionosphere [34]	351	34	2
Letter-26 [34]	20,000	16	26
Sonar [34]	208	2	60
USPS [35]	9,298	256	10
Vowel [34]	990	10	11
Waveform [34]	5,000	21	3

Experiments



Experiments



Experiments

2D-Image dataset	GB-CNN	CNN
MNIST	99.66%	99.40%
CIFAR-10	86.72%	86.30%
Rice varieties	99.33%	94.68%
Fashion-MNIST	94.02%	92.38%
Kuzushiji-MNIST	98.82%	97.21%
MNIST-Corrupted	99.62%	99.45%
Rock-Paper-Scissors	89.78%	69.35%

Dataset	GB-DNN	DNN	GBNN	XGBoost
Digits	98.17%	97.72%	97.27%	96.83%
Ionosphere	95.38%	92.04%	90.69%	91.77%
Letter-26	95.08%	95.18%	75.63%	96.28%
Sonar	85.85%	85.98%	78.18%	83.94%
USPS	97.15%	96.06%	94.04%	95.94%
Vowel	97.58%	97.17%	83.30%	93.02%
Waveform	84.90%	82.65%	87.01%	84.74%

Conclusion

- 본 논문의 핵심 아이디어는 Gradient Boosting 방법론을 신경망 학습에 적용하는 것임
- 본 연구가 제시한 모델은 이미지 분류를 위한 GB-CNN과 테이블 데이터 분류를 위한 GB-DNN임
- 두 모델은 일종의 학습방법론이며, 다양한 구조의 모델에도 적용될 수 있음
- 두 모델은 일반적인 방법으로 학습한 동일 모델 대비 조금 더 나은 성능을 냄