

## An Efficient Deep Learning Ensemble Using a Distribution of Label Embedding

Saerom Park\*

\*Assistant Professor, Dept. of Convergence Security Engineering, Sungshin Women's University, Seoul, Korea

### [Abstract]

In this paper, we propose a new stacking ensemble framework for deep learning models which reflects the distribution of label embeddings. Our ensemble framework consists of two phases: training the baseline deep learning classifier, and training the sub-classifiers based on the clustering results of label embeddings. Our framework aims to divide a multi-class classification problem into small sub-problems based on the clustering results. The clustering is conducted on the label embeddings obtained from the weight of the last layer of the baseline classifier. After clustering, sub-classifiers are constructed to classify the sub-classes in each cluster. From the experimental results, we found that the label embeddings well reflect the relationships between classification labels, and our ensemble framework can improve the classification performance on a CIFAR 100 dataset.

▶ **Key words:** Deep Ensemble Learning, Clustering, Multi-class classification, Label embedding, Stacking Ensemble Model

### [요 약]

본 연구에서는 레이블 임베딩의 분포를 반영하는 딥러닝 모형을 위한 새로운 스택킹 앙상블 방법론을 제안하였다. 제안된 앙상블 방법론은 기본 딥러닝 분류기를 학습하는 과정과 학습된 모형으로부터 얻어진 레이블 임베딩을 이용한 군집화 결과로부터 소분류기들을 학습하는 과정으로 이루어져 있다. 본 방법론은 주어진 다중 분류 문제를 군집화 결과를 활용하여 소 문제들로 나누는 것으로 한다. 군집화에 사용되는 레이블 임베딩은 처음 학습한 기본 딥러닝 분류기의 마지막 층의 가중치로부터 얻어질 수 있다. 군집화 결과를 기반으로 군집화 내의 클래스들을 분류하는 소분류기들을 군집의 수만큼 구축하여 학습한다. 실험 결과 기본 분류기로부터의 레이블 임베딩이 클래스 간의 관계를 잘 반영한다는 것을 확인하였고, 이를 기반으로 한 앙상블 방법론이 CIFAR 100 데이터에 대해서 분류 성능을 향상시킬 수 있다는 것을 확인할 수 있었다.

▶ **주제어:** 딥 앙상블 학습, 군집화, 다중 분류 문제, 레이블 임베딩, 스택킹 앙상블 모형

---

• First Author: Saerom Park, Corresponding Author: Saerom Park  
\*Saerom Park (psr6275@sungshin.ac.kr), Dept. of Convergence Security Engineering, Sungshin Women's University  
• Received: 2020. 11. 04, Revised: 2021. 01. 04, Accepted: 2021. 01. 04.

## I. Introduction

최근 인공지능 알고리즘은 다량의 데이터로부터 복잡한 패턴들을 학습할 수 있게 하는 딥러닝 모형을 기반으로 빠르게 발전하고 있다 [1][2]. 앙상블 방법론은 여러 모형들의 결과를 잘 결합하는 방법으로 개별 모형을 사용할 때보다 더 안정적이고 좋은 성능을 나타내기 때문에 기계학습 모형의 실제 응용에서 활발하게 사용되고 있다 [3][4][5][6][7]. 딥러닝 모형은 한 개의 큰 모형을 학습하는데에도 오랜 시간과 노력이 필요하기 때문에, 여러 모형을 독립적으로 하여 학습시킨 후에 학습 결과를 종합하는 기본 앙상블 방법론을 적용하기 위해서는 사용되는 모형의 수에 비례하는 시간과 노력이 필요하다. 따라서 효율적인 앙상블 모형 구축 방법론이 필요하다.

딥러닝 모형은 각 층마다 주어진 데이터의 수치 표현들이 학습한다는 점에서 표현 학습 방법론으로 주목을 받고 있다. 특히, 다중 분류 문제를 위한 딥러닝 모형의 최종 출력은 가능한 출력 값들에 대한 확률로써 주어진다. 이러한 확률은 주로 로짓(logit) 벡터에 소프트맥스(softmax) 함수를 취함으로 얻어질 수 있다. 다중 분류 모형을 위한 딥러닝 모형에 대해서 클래스 레이블에 대한 임베딩을 활용하고자 하는 연구들이 진행되어 왔다 [8][9].

본 연구에서는 이러한 로짓 벡터를 계산하기 위해서 사용되는 마지막 층의 가중치 행렬이 출력 레이블들에 대한 임베딩으로 해석될 수 있다는 특징을 활용하고자 한다. 본 논문에서는 주어진 학습 데이터로부터 학습된 딥러닝 모형의 마지막 층의 가중치 행렬로부터 얻어진 레이블 임베딩의 특징들을 살펴보고 이를 활용하여 주어진 다중 분류 문제를 소문제로 나눌 수 있는 방법을 제안한다.

기존 앙상블 방법들은 주로 같은 문제를 해결하는 여러 모형들을 학습시키고, 이로부터 나온 결과들을 종합하여 최종 예측을 수행하였다 [10]. 하지만, 본 논문에서는 주어진 문제를 소문제로 나누고, 소문제를 해결하는 분류기들을 종합하는 앙상블 방법을 제안하여 최종 예측을 수행하고자 한다.

## II. Preliminaries

### 1. Related works

앙상블 방법은 주어진 여러 개의 기본 모형의 결과들을 조합하여 하나의 예측 모형을 만드는 것을 목표로 한다 [3][8]. 전통적인 기계학습 모형에서는 배깅(bagging)이나

부스팅(boosting)과 같은 방법을 통해서 간단한 의사 결정 트리 모형으로부터 강건하고, 성능이 향상된 앙상블 모형을 얻고자 하였다.

특히, 스택킹(stacking) 방법은 많은 학습된 모형들로부터의 예측 결과를 입력으로 하여 최종 예측 모형을 만드는 방법이다 [2][11]. 하지만, 스택킹 앙상블의 경우 최종 예측 모형을 구축할 때에 주로 기본 모형으로부터의 예측 결과들을 사용하게 되기 때문에 과적합(overfitting) 있고, 이를 해결하기 위한 cross-validation 기반의 스택킹 방법이 제안되었다.

기존의 앙상블 방법들에서는 주로 기본 모형으로부터의 예측 결과들을 어떻게 조합할 것인가가 중요해 지고, 결과적으로는 성능을 향상시킬 수 있는 예측 가중치를 구하는데에 초점을 맞추게 된다. 또한, 딥러닝 모형의 특성을 분석하여 학습된 모형으로부터 불확실성을 추정하고 이를 개선하기 위한 연구들이 제안되어왔다 [12][13][14].

본 연구 논문의 기여점은 딥러닝 모형의 특징에 초점을 맞추어 예측의 확신 정도를 활용하여 클래스 레이블을 기반으로 한 군집에 대한 소분류기를 종합하여 최종 예측을 수행하는 앙상블 방법을 제안하였다는 것이다. 딥러닝 모형의 예측 결과의 확률 분포로부터 예측의 확신 정도를 추정할 수 있는 간단한 방법을 제안하고, 실제 새로운 테스트 데이터에 대한 예측 시에 예측의 확신 정도가 높은 경우에는 기본 분류기를 통해 예측하고, 불확실성이 높을 경우에 앙상블 모형을 통해 예측을 수행하도록 하여 예측 모형의 성능을 개선하고자 하였다.

### 2. Classification Model Using Deep Learning

딥러닝 모형은 분류 기준이 많은 분류 문제에 성공적으로 적용되어 왔다. 특히, 이미지 분류 문제에서는 합성곱 신경망(Convolutional Neural Network: CNN) 모형을 적용하여 많은 응용 분야에서 뛰어난 성능을 보이고 있다 [1]. 딥러닝 기반의 분류 모형의 구조를 살펴보면, 마지막 층은 분류 레이블 수 만큼의 노드를 가지게 되며, 입력 데이터에 대해 네트워크의 출력을 계산한 후에 소프트맥스 함수를 적용하여 각 클래스의 속할 확률 값을 얻게 된다. 입력 데이터를  $x \in$ , 출력 데이터를  $y \in$  라고 하고, 주어진 딥러닝 네트워크 함수를  $f$ : 라고 하면, 최종 분류 예측기  $c(x)$ : 는 다음과 같이 얻어진다.

$$c(x) = \operatorname{argmax}_{k \in \{1, \dots, C\}} \frac{\exp(f_k(x))}{\sum_{l=1}^C \exp(f_l(x))}$$

딥러닝 예측 모형을 학습하기 위해서  $k$  번째 클래스에 속할 예측 확률인 다음의 식을 기반으로 소프트맥스 크로스 엔트로피 손실 함수를 계산한다.

$$p_k(x) = \text{softmax}(f(x))_k = \frac{\exp(f_k(x))}{\sum_{l=1}^C \exp(f_l(x))}$$

$$L(D) = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^C y_k^{(i)} \text{softmax}(f(x))_k$$

위의 손실 함수는 다음의 학습 데이터로부터 계산 된다.  $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\} \subset \mathbb{R}^d \times \{0, 1\}^C$  학습 시에는 손실 함수를 최소화 하는 방향으로 네트워크의 파라미터들이 학습된다. 하지만, 위에 딥러닝 모형을 학습 하는 최적화 문제가 여러 국지 최적해를 갖게 되기 때문에 초기 딥러닝 모형의 학습은 RBM 이나 Autoencoder 의 비교사 학습 방법론들로 네트워크를 사전 학습 (pre-training)한 후에 얻어진 파라미터에서 시작하여 위의 손실 함수로 예측 모형을 학습하여 더 나은 예측 성능을 지닌 파라미터를 학습하고자 하였다. 최근에는 최적화 방법론들의 발달과 rectified linear unit (ReLU), batch normalization, layer normalization, residual layer 와 같은 학습에 도움이 되는 구조들이 활용되면서 좋은 국지 최적해를 얻을 수 있게 되어 사전 학습 없이도 많은 층을 가진 분류 네트워크를 학습할 수 있게 되었다. 하지만, 사전 학습의 기본이 되는 좋은 파라미터 상태에서부터 시작 하는 아이디어는 전이 학습(transfer learning)과 같이 학습이 어려운 경우에 좋은 학습 시작 파라미터를 얻기 위해 관련된 데이터로 학습한 초기 값을 사용하기도 한다.

### III. The Proposed Scheme

#### 1. Distribution of label embedding

딥러닝 모형을 학습한 후에 예측을 수행하기 위해서 얻어지는 소프트맥스 함수를 취하기 전의 네트워크의 출력  $f(x) \in \mathbb{R}^C$  을 로짓(logit) 이라고 한다. 딥러닝 네트워크에서 어떠한 네트워크 구조를 사용하더라도 분류 문제를 위한 최종 예측 값을 계산하기 위해서 상위 1개 이상의 선형층을 사용하게 된다. 따라서, 로짓 값은 항상 선형층을 통해 얻어지게 된다. 총  $M$  개의 층을 가지고 있는 딥러닝 네트워크에서 마지막 선형층의 앞단의 층에서의 노드의 개수를  $p$  라고 하고,  $M-1$  층의 출력 값을  $z_{M-1} \in \mathbb{R}^p$  이라고 하면, 최종 층의 로짓 값은 다음과 같은 식을 통해서 얻어질 수 있다.

$$f(x) = W^T z_{M-1} + b$$

위의 식에서  $W \in \mathbb{R}^{p \times C}$  는 가중치 행렬이고,  $b \in \mathbb{R}^C$  는 절편 값이 된다. 위의 식의 계산 과정을 살펴보면 학습된 네트워크에 대해서 가중치 벡터와 절편 값은 고정되어 있고, 입력 데이터에 따라서  $z_{M-1}$  값이 달라지게 된다. 소프트맥스 함수를 적용했을 때 특정 클래스에 대한 로짓 값이 클수록 해당 클래스의 확률 값이 커지게 되므로  $k$  번째 클래스에 해당하는 로짓 값은 다음과 같은 식으로 얻어진다.

$$f_k(x) = W_{:,k}^T z_{M-1} + b_k$$

이 식을 통해서 가중치 행렬의  $k$  번째 열벡터와의 내적 값이 커질수록 해당 클래스의  $k$  번째 클래스의 로짓 값이 커질 수 있음을 확인할 수 있다. 즉, 가중치 행렬  $W$  의  $k$  번째 열벡터와의 내적 값이 클수록  $k$  번째 클래스에 속할 확률이 커지게 된다. 따라서, 우리는 학습된 네트워크의 마지막 층의 가중치 벡터의 열 벡터들을 레이블 임베딩으로 간주하고, 특정 클래스 레이블에 대한 대표 벡터로 설정하였다. 이렇게 얻어진 레이블 임베딩을 통해서 많은 클래스 레이블이 존재하는 문제에서의 클래스 간에 유사도를 측정할 수 있다. 기본적으로, 최종 로짓 값을 얻기 위해서 내적 연산을 수행하기 때문에, 레이블 간의 유사도를 측정하는 데에는 일반적인 유클리디안 거리(Euclidean Distance) 측도 대신에 코사인 유사도(Cosine Similarity)를 계산하였다.

본 앙상블 방법론에서는 클래스의 수가 많은 다중 분류 문제에서 기본 딥러닝 모형의 예측이 불확실할 경우 이에 대한 예측을 소문제에 대한 분류기를 활용해서 수행하는 방법론을 제안한다. 먼저, 주어진 학습 데이터에 대해 학습을 수행해서 얻어진 딥러닝 모형으로부터 레이블 임베딩을 얻게 되면 유사한 클래스 레이블들을 군집화 하여 유사한 클래스들을 분류하는 소문제들을 얻을 수 있다. 각 군집마다 얻어진 소문제들에 대한 분류기를 재학습하여 유사한 클래스끼리 분류하는 예측 성능을 높일 수 있고, 이를 예측이 불확실한 경우의 최종 예측 성능을 높이기 위해서 활용할 수 있다. 소문제에 대한 분류기를 학습할 때에는 전체 네트워크를 임의의 초기값으로부터 학습하는 것이 아닌 전이 학습에서와 같이 원래의 딥러닝 모형의 학습된 파라미터부터 시작하여 학습함으로써 더 적은 횟수의 반복 회수만으로도 효율적으로 학습을 할 수 있게 된다.

#### 2. Uncertainty estimation

본 앙상블 방법에서는 최종 예측을 수행하기 위해 모든 입력 데이터에 대해서 소문제에 대한 예측 결과를 적용하

는 것이 아니고, 기본 분류기의 예측 결과가 불확실할 경우에만 소분류기의 예측 결과를 활용한다. 여기에서 예측 결과가 불확실하다는 것은, 소프트맥스 함수를 통해 얻어진 예측 확률 분포를 통해서 측정할 수 있다. 우리는 불확실성 추정 지표 중에서도 가장 널리 사용되는 엔트로피 (entropy) 지수를 이용하였다. 클래스  $k$ 에 속할 확률을  $p_k$ 라고 할 때, 엔트로피 값은 다음과 같이 계산될 수 있다.

$$-\sum_{k=1}^C p_k \log(p_k)$$

엔트로피 값은 각 데이터 별로 얻어진 로짓 값으로부터 계산되므로, 데이터 배치에 대해서 동시에 계산될 수 있는 특징을 지닌다. 계산된 엔트로피 값이 크다는 것은 불확실성이 높은 것을 의미하고, 작다는 것은 불확실성이 낮고 확실하게 예측하였다는 것을 의미하게 된다. 따라서 위의 불확실성 지표로부터 기본 딥러닝 모형의 예측에 대한 확신이 있는지를 추정할 수 있고, 불확실성이 클 경우에만 앙상블 모형의 소문제에 대한 분류기를 적용할 수 있게 된다.

### 3. Efficient Ensemble Model

앙상블 모형을 구축하기 위해서는 가장 먼저 기본 딥러닝 분류기를 학습하여야 한다. 기본 딥러닝 분류기로부터 얻어진 레이블 임베딩을 가지고 코사인 유사도를 활용하여 군집화를 진행하면 총  $S \ll C$  개의 소문제를 얻을 수 있다. 주어진 학습 데이터를 군집화 결과에 따라서 소문제들에 대한 학습데이터들로 분리  $\{1, \dots, C\} = C_1 \cup \dots \cup C_S$ 한다. 즉, 하나의 소문제는 기본 분류기가 헛갈리는 유사한 레이블들을 분류할 수 있도록 하는 것을 목적으로 한다. 분리된 학습데이터를 활용하여  $S$  개의 서로 다른 소문제에 대한 분류기  $c_j(x) : \mathbb{R}^d \rightarrow C_j, j \in \{1, \dots, S\}$ 를 학습하고, 소문제에 대한 분류기를 학습하기 위한 초기 값은 학습된 기본 딥러닝 분류기로부터 얻는다. 소문제에 대한 분류기들이 다 학습이 되면, 앙상블 모형 구축을 위한 기본 구성 요소들의 학습 과정은 완료된다.

학습이 완료 되면 최종 앙상블 모형은 기본 분류기와 소문제에 대한 분류기들로부터 최종 예측 결과를 도출한다. 새로운 테스트 데이터  $x_{test}$ 가 주어졌을 때, 다음의 과정을 통해서 클래스 레이블을 예측할 수 있다.

기본 분류기로부터 다음의 예측 확률을 계산한다.

$$p_k(x_{test}) = \text{softmax}(f(x))_k$$

얻어진 예측 확률 분포로부터 예측의 불확실성을 계산하기 위해서 엔트로피 값을 계산한다.

$$E = -\sum_{k=1}^C p_k(x_{test}) \log p_k(x_{test})$$

계산된 엔트로피 값이 만약 작다면, 기본 분류기의 예측 결과의 불확실성이 작다는 것을 알 수 있다. 이는 특정 클래스에 속할 확률이 높게 예측이 되었다는 것을 의미하게 된다. 따라서, 위에서 계산된 엔트로피 값이 임계값  $e$ 보다 작으면 ( $E < e$ ), 기본 분류기의 예측 결과를 최종 예측 결과로 출력한다.

$$\text{output: } c(x_{test})$$

하지만, 계산된 엔트로피 값이 임계값 보다 크게 된다면 ( $E \geq e$ ), 이는 예측의 불확실성이 크다는 것을 의미하게 된다. 따라서, 해당 경우에는 앞에서 얻어진 각 클래스에 대한 예측 확률들을 군집마다 합산하여, 해당 군집에 속할 확률들을 구한다. 얻어진 군집에 대한 확률을 통해 가장 높은 확률을 가지는 군집을 구하고, 해당 소문제에 해당하는 소분류기를 통해 최종 예측 값을 출력한다.

$$j = \text{argmax}_{t \in \{1, \dots, S\}} p_{C_t} = \sum_{k \in C_t} p_k(x_{test})$$

$$\text{output: } c_j(x_{test})$$

IV-3 에서의 실험을 통해 소분류기의 성능이 기본 분류기보다 우수함을 확인해 볼 수 있고, 이러한 성질을 활용하여 기본 분류기의 예측이 불확실한 경우에 소분류기들을 통해 개선된 예측을 할 수 있는 방법론을 제안하였다.

## IV. Experiments

### 1. Experimental design

본 앙상블 모형은 주어진 학습 데이터들로부터 기본 구성 요소들을 학습하는 학습 과정과 새로운 데이터에 대해서 클래스 레이블을 예측하는 예측 과정으로 나눌 수 있다. 학습 과정은 기본 분류기를 학습하는 과정, 기본 분류기로부터 얻어진 레이블 임베딩으로 클래스에 대한 군집화를 수행하는 과정, 군집화 결과로부터 얻어진 소문제들에 대한 분류기들을 학습하는 과정으로 이루어져 있다. 예측 과정은 앞에 III-3에 설명되어 있는 과정들로 이루어져 있다. 학습 과정에서 클래스에 대한 군집의 개수  $S$ , 예측 과정에서 엔트로피의 임계값  $e$ 를 결정하기 위해서는 학습 데이터로 사용되지 않은 검증(validation) 데이터가 필요하다. 따라서 제안된 앙상블 모형을 구축하기 위해서 주어진 전체 데이터를 학습 데이터, 검증 데이터, 시험 데이터로 분리하였다.

본 논문에서 제안한 앙상블 방법은 많은 수의 클래스 레이블이 있을 때에 적용하는 것이 적합하기 때문에 총 100개의 클래스 레이블에 대해서 예측하는 CIFAR100 이미지 데이터셋에 적용해 보았다. CIFAR100 데이터는 총 100개의 클래스 레이블에 대해서 20개의 상위 클래스(super class)를 가진다. CIFAR100 데이터가 포함하고 있는 클래스 레이블에 대한 정보는 다음 Table 1과 같다.

Table 1. Class label information for CIFAR100 dataset

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

실제 문제에서는 위의 CIFAR100 데이터와는 다르게 상위 클래스 정보가 없고, 더 적은 수로 소문제를 나누는 것이 적합할 수 있기 때문에 본 연구 방법론에서는 상위 클래스 정보를 사용하지 않고, 레이블 임베딩의 군집화를 통해서 소문제를 얻을 수 있었다.

CIFAR100 데이터에 대한 분류 문제를 해결하기 위해서 본 연구에서는 Wide ResNet(WRN) 구조의 네트워크를 사용하여 실험해 보았다. Wide ResNet는 28 x 10의 구조를 사용하여 학습을 수행하였다. 또한, 학습 시에는 기본적인 정규화 뿐만 아니라, random crop, random horizontal flip의 변환들을 수행하여 학습을 도왔다. CIFAR 100 데이터의 경우에는 학습 데이터가 60,000개이고, 시험 데이터가 10,000개 이므로 학습 데이터 중에 80%인 48,000개는 학습 데이터로, 12,000개는 검증 데이터로 사용하였다.

본 실험을 통해서 실제 이미지 데이터인 CIFAR100에 대해 앙상블 알고리즘을 적용해 봄으로써, 학습된 레이블 임베딩의 분포를 살펴보고, 소문제에 대한 분류기의 성능 및 앙상블 성능을 비교해 보았다.

## 2. Results for distribution of label embedding

본 실험에서는 학습된 기본 WRN 모형으로부터 얻어진 레이블 임베딩의 코사인 유사도 측도를 기반으로 하여 spectral clustering 방법론을 적용한 결과를 보여준다. 먼저, 학습된 레이블 임베딩의 코사인 유사도에 대한 유사도 행렬을 구하고 100개의 클래스에 대해서 9개의 군집으로 군집화를 수행하였을 때에 각 군집에 할당되는 클래스의 분포를 나타내면 다음 Fig. 1과 같다.

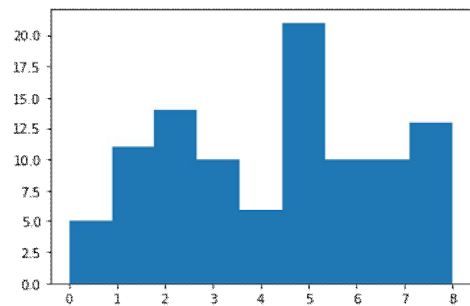


Fig. 1. Clustering results for CIFAR100 classes

위의 군집화 결과 분포를 살펴보면 군집 0과 군집 4에 속하는 클래스의 수가 적고, 군집 5에 속하는 클래스의 수가 가장 많음을 확인할 수 있다. 위의 군집들에 대한 실제 군집화 결과를 보면 다음 Table 2와 같다.

Table 2. Clustering results of clustering using label embeddings

cluster	classes
0	baby,boy,girl,man, woman
1	bridge,castle,cloud, forest,mountain,palm_tree, plain,road,rocket,sea,skyscraper
2	bee,beetle,butterfly,caterpillar,cockroach,crab,lizard,lobster,orchid,snail,snake,spider,sunflower,worm
3	apple,bowl,hamster,mushroom,orange,pear,plate,poppy,sweet_pepper,tulip
4	dinosaur,maple_tree, oak_tree,pine_tree,rose,willow_tree
5	bear,beaver,camel,cattle,chimpanzee,elephant,fox,kangaroo,leopard,lion,mouse,otter,porcupine,possum,rabbit,raccoon,shrew,skunk,squirrel,tiger,wolf
6	bicycle,bus,house,lawn_mower,motorcycle,pickup_truck,streetcar,tank,tractor,train
7	aquarium_fish,crocodile,dolphin,flatfish,ray,seal,shark,tROUT,turtle,whale
8	bed,bottle,can,chair,clock,coUCH,cup,keyboard,lamp,table,telephone,television,wardrobe'

위의 군집화 결과를 보면, 클래스 할당 수가 적었던 군집 0은 사람과 관련된 클래스가 하나의 군집을 이루었음을 확인해 볼 수 있고, 군집 4에는 실제 데이터들을 살펴보면, 초록색 배경들을 가지게 되는 나무에 해당하는 클래스들로 주로 이루어져 있음을 확인해 볼 수 있다. 또한 가장 많은 수의 클래스들이 포함되어 있는 군집 5의 경우에는 동물들로 이루어져 있음을 확인할 수 있었다. 위의 군집화 결과는 학습된 기본 딥러닝 모형으로부터 얻어진 레이블 임베딩에 대해 군집화를 수행하면 실제 분류 시에 헛갈릴 수 있는 클래스들끼리 같은 군집에 속하도록 만들어 줄 수 있다는 것을 보여준다.

따라서 각 군집마다의 소문제에 대해서 분류를 수행할 수 있는 분류기를 학습함으로써 헛갈리는 클래스들로 이루어진 작은 문제에 대해 분류 성능을 개선할 수 있는 소분류기들을 얻을 수 있을 것으로 기대된다.

### 3. Results for ensemble classification

앞에서 얻어진 군집화 결과들을 기반으로 각 군집에 해당하는 클래스들을 분류하는 소분류기들을 학습하였다. 소분류기들은 군집의 수만큼 얻어지고, 초기 기본 딥러닝 모형과 같은 구조를 갖지만, 최종 분류기의 출력 노드의 수가 다르고, 파라미터의 초기값으로 전체 학습 데이터로 학습했던 가중치와 편차 값을 사용하였다. 기본 분류기를 학습하기 위해서는 총 200 번의 epoch 으로 학습이 이루어졌지만, 소분류기의 경우에는 학습된 파라미터로부터 시작하여 학습이 이루어졌기 때문에 검증 데이터로 성능을 살펴보았을 때, 대부분 8~12회 정도의 epoch에서 학습이 안정됨을 확인할 수 있었다. 따라서 소분류기의 학습에는 15회의 epoch을 가지고 학습을 수행하였다.

다음의 결과는 소분류기를 학습한 후에 각 소문제에 대한 원래의 기본 분류기와 소분류기의 성능을 보여준다. 분류 성능을 보기 위해서 각 군집에서의 top1 정확도와 top3 정확도를 측정해 본 결과는 다음 Table 3 와 같다.

Table 3. Comparison of top1/top3 accuracies for baseline (basic classifier) and sub-classifier in 9 clusters

cluster	top1 accuracy (%)		top3 accuracy (%)	
	baseline	sub-classifier	baseline	sub-classifier
0	56.86	<b>62.77</b>	83.05	<b>92.78</b>
1	86.91	<b>90.12</b>	96.26	<b>99.48</b>
2	75.28	<b>88.95</b>	85.31	<b>96.23</b>
3	77.67	<b>88.15</b>	91.53	<b>97.88</b>
4	74.29	<b>80.60</b>	93.25	<b>97.67</b>
5	75.48	<b>79.27</b>	88.15	<b>92.86</b>
6	85.58	<b>89.64</b>	94.64	<b>98.49</b>
7	72.41	<b>87.21</b>	82.31	<b>95.37</b>
8	81.75	<b>89.13</b>	90.41	<b>96.09</b>

위의 결과로부터 모든 군집에 대해서 소분류기가 기본 분류기에 비해서 4~15% 정도의 정확도 개선이 있었다는 것을 확인해 볼 수 있다. 이는 전체 문제를 군집화를 통해서 소문제로 나누었기 때문에 100개의 클래스를 분류하는 것 보다 더 적은 수의 클래스를 분류하는 것이 학습이 용이하다는 것을 보여준다. 각 군집의 top3 정확도에서 baseline 의 경우에는 전체 클래스에 대해서 학습된 모형을 활용하였지만 sub-classifier 와 같이 소문제에 대한 분류 정확도만 보기 위해서 각 군집에 속한 클래스들에 대한 로짓 값들만 비교하여 군집 내의 클래스 중에서 로짓 값이 가장 큰 3개의 클래스 중에 실제 클래스가 존재하는지를 통해 계산하였다.

군집의 수에 따른 성능의 차이를 확인해 보기 위해서 위의 실험을 군집의 수가 8개인 경우와, 3개인 경우에도 수행해 보았다. Table 4 는 군집의 수가 8개인 경우의 결과를 보여주고, Table 5 은 군집의 수가 3개일 경우에 결과를 보여준다.

Table 4. Comparison of top1/top3 accuracies for baseline (basic classifier) and sub-classifier in 8 clusters

cluster	top1 accuracy		top3 accuracy	
	baseline	sub-classifier	baseline	sub-classifier
0	68.00	72.67	87.17	94.50
1	78.67	83.78	95.56	98.00
2	75.32	79.95	87.22	91.68
3	84.36	89.36	94.63	98.18
4	79.21	85.43	90.07	94.71
5	83.27	88.82	93.63	98.18
6	76.67	85.40	89.53	95.67
7	74.75	86.83	87.58	96.50

Table 5. Comparison of top1/top3 accuracies for baseline (basic classifier) and sub-classifier in 3 clusters

cluster	top1 accuracy		top3 accuracy	
	baseline	sub-classifier	baseline	sub-classifier
0	77.43	80.13	90.41	94.05
1	74.00	77.58	86.71	89.45
2	80.82	83.87	95.00	97.04

Table 4 와 Table 5 의 결과를 보면 군집의 수가 적으면 한 군집에 속하는 클래스의 수가 많아지므로 소문제의 난이도가 어려워져 소분류기로 인한 성능 향상이 줄어들을 확인해 볼 수 있다. 하지만, 군집의 수가 많아질수록 학습해야 하는 소분류기의 수가 많아지므로 학습 시의 비용도 증가하게 되고, 군집으로의 할당이 부정확해질 수 있으므로 문제에 따라서 적절한 수준의 군집 수를 결정하는 것이 중요하게 된다. 하지만 군집의 수가 8개일 경우에도 소분류기를 사용하였을 경우에 충분한 성능 개선이 있으므로 너무 작거나 너무 큰 정도의 군집 수를 제외하면, 성능 개선 효과를 기대해 볼 수 있다.

본 실험에서는 위와 같이 학습된 소분류기를 활용하여 III-3 에서 제안된 예측 알고리즘의 새로운 데이터에 대한 앙상블 모형의 예측을 수행해 보았다. 전체 시험 데이터에 대해서 앙상블 모형을 적용해 본 결과 예측 성능은 77.73% 로 기본 분류 모형의 예측 성능보다 1% 정도 향상되었다. 앙상블 모형의 성능 향상은 소분류기의 성능 향상에 비해서는 미미하지만 앙상블 모형에서 소분류기 모

형을 엔트로피가 충분히 클 경우에만 적용하였기 때문에 대부분의 예측에서 기본 모형과 앙상블 모형의 예측이 비슷하기 때문으로 이해될 수 있다. 아래의 Fig. 2 는 CIFAR100 시험 데이터 중에 기본 분류기의 예측과 앙상블 모형의 예측이 다른 orange 이미지의 예를 보여준다.



Fig. 2. An example for orange class

Fig. 3 은 Fig. 2. 의 데이터를 입력으로 주었을 때 기본 분류기가 예측한 예측 결과 로짓 값  $f(x)$  을 그래프로 나타낸 것이다. 해당 그래프에서 x 축은 100개의 클래스를 나타내고, 아래 리스트인 ['mushroom', 'hamster', 'orange', 'man', 'rabbit'] 은 로짓 값이 큰 클래스 5개를 보여준다. 그래프에서 확인할 수 있듯이 로짓 값이 분포가 특정 클래스를 뚜렷하게 강조해 주지 못하기 때문에 엔트로피 값이 크게 되고, 따라서 앙상블 모형을 적용하게 된다. 그래프 아래에 있는 5개의 클래스 정보는 가장 예측 확률이 높은 순서대로 다섯 개의 클래스를 보여주고 있다. 결과로부터 실제 클래스 레이블인 orange 는 세 번째로 높은 확률을 가짐을 확인해 볼 수 있다.

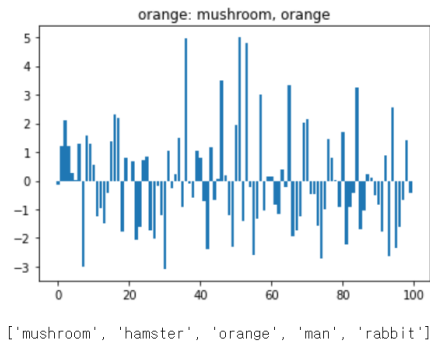
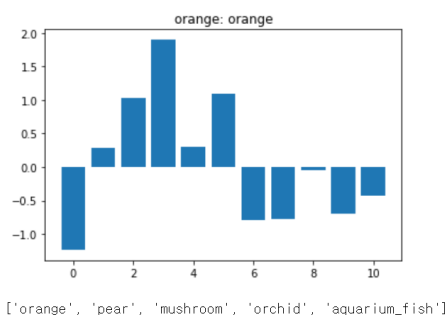


Fig. 3. Prediction results of baseline for the orange image

반면에, Fig. 4 는 앙상블 모형이 할당한 소분류기의 예측 결과를 보여주고 있다. 해당 그래프도 Fig. 3에서처럼 x 축은 해당 군집의 클래스들을 나타내고, y 축은 로짓 값  $f(x)$ 을 나타낸다. 또한 아래의 리스트인 ['orange', 'pear', 'mushroom', 'orchid', 'aquarium\_fish'] 는 가장 높은 로짓 값을 나타내는 상위 5개의 클래스를 보여준다. 소분류기에서는 실제 이미지의 클래스인 orange가 가장 높은 예

측 확률을 보이고 있고, 원래 기본 분류기에서 가장 높은 확률로 예측되었던 mushroom은 세 번째로 높은 확률로 예측됨을 확인해 볼 수 있다. 따라서 이렇게 기본 예측 분포의 엔트로피가 높은 경우에 가장 높은 확률을 가진 클래스로 예측을 수행하게 되면 위에 Fig. 3의 예에서처럼 확률이 높은 3개의 클래스가 비슷한 확률을 가짐에도 불구하고 가장 확률이 높은 클래스로 분류하게 되어 잘못된 클래스로 예측하게 될 수 있다. 하지만, 제안한 방법론을 통해 예측을 수행하게 되면, 조금 더 쉬운 소문제의 소분류기를 통해서 확신 있게 예측한다는 것을 확인해 볼 수 있다.



['orange', 'pear', 'mushroom', 'orchid', 'aquarium\_fish']

Fig. 4. Prediction results of sub-classifier for the orange image

## V. Conclusions

본 연구 논문은 주어진 클래스 레이블의 수가 많은 경우의 다중 분류 문제에서 불확실한 예측에 대한 문제를 해결하기 위해서 전체의 큰 문제를 소문제로 분리한 후에 이를 해결하는 소분류기를 앙상블 하여 예측을 수행할 수 있는 방법론을 제안하였다. 딥러닝 모형을 기반으로 하였기 때문에 소문제로 분리하기 위해 딥러닝 모형의 학습 결과로 나올 수 있는 레이블 임베딩을 통해서 군집화를 수행하였다. 또한, 모형의 불확실성을 측정하기 위해서는 엔트로피 측도를 사용하였다. 그 결과 전체 문제를 소문제로 나누어 학습한 소분류기가 기본 분류기에 비해 4~15% 정도의 성능 향상을 보였고, 이로부터 구축된 앙상블 모형의 경우에도 1% 정도의 성능 향상을 보였다.

본 연구 논문에서는 기본적인 군집화 방법으로 spectral clustering 방법론을 사용하였지만, 추후 연구 방향으로 다양한 군집화 방법론을 적용해 볼 수 있을 것이다. 또한, 예측 과정에서도 군집화 결과를 기준으로 얻어진 확률을 통해서 하나의 소분류기에 할당하여 최종 예측을 수행하는 간단한 방법을 적용하였지만, 추후 연구를 진

행하여 여러 개의 소분류기의 결과들을 종합할 수 있는 방향으로 발전시키면, 더 큰 성능 향상을 이룰 수 있을 것으로 기대된다.

## ACKNOWLEDGEMENT

This work was supported by the Sungshin Women's University Research Grant of 2019

## REFERENCES

- [1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Communications of the ACM* Vol. 60, No. 6, pp.84-90 May. 2017. DOI: 10.1145/3065386
- [2] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. Vol. 27, No. 1, pp.3104-3112, Dec. 2014.
- [3] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. "The elements of statistical learning." New York: Springer series in statistics, Vol. 1. No. 10. 2001.
- [4] Malinin, Andrey, Bruno Mlodozeniec, and Mark Gales. "Ensemble distribution distillation." *arXiv preprint arXiv:1905.00076*, 2019.
- [5] Ko, Hyungjin, et al. "Loss-Driven Adversarial Ensemble Deep Learning for On-Line Time Series Analysis." *Sustainability* Vol. 11, No. 12, pp. 1-24 Jun. 2019 DOI: DOI:10.3390/su11123489.
- [6] Sesmero, M. Paz, Juan M. Alonso-Weber, and Araceli Sanchis. "CCE: An ensemble architecture based on coupled ANN for solving multiclass problems." *Information Fusion* Vol. 58, pp. 132-152. Jun. 2020. DOI: 10.1016/j.inffus.2019.12.015
- [7] Park, S., J. Hah, and J. Lee. "Inductive ensemble clustering using kernel support matching." *Electronics Letters* Vol. 53, No. 25, pp. 1625-1626 Dec. 2017. DOI: 10.1049/el.2017.2159
- [8] Sun, Xu, et al. "Label embedding network: Learning label representation for soft training of deep networks." *arXiv preprint arXiv:1710.10393*, 2017.
- [9] Sellah, Smail, and Vincent Hilaire. "Label Clustering for a Novel Problem Transformation in Multi-label Classification." *J. UCS* Vol.26 No.1 pp. 71-88. Jan. 2020,



- [10] Tao, Sean. "Deep neural network ensembles." *International Conference on Machine Learning, Optimization, and Data Science*. Springer, Cham, 2019.
- [11] Low, Cheng-Yaw, Jaewoo Park, and Andrew Beng-Jin Teoh. "Stacking-based deep neural network: Deep analytic network for pattern classification." *IEEE Transactions on Cybernetics*, Vol. 50, No. 12, Dec. 2019. DOI: 10.1109/TCYB.2019.2908387
- [12] Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., & Lakshminarayanan, B. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness." *Advances in Neural Information Processing Systems* Dec. 2020.
- [13] Gabrié, M., Manoel, A., Luneau, C., Barbier, J., Macris, N., Krzakala, F., & Zdeborová, L. "Entropy and mutual information in models of deep neural networks." *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 50, No. 12, pp. 5021-5034 Dec, 2020. DOI: 10.1109/TCYB.2019.2908387
- [14] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... & Nahavandi, S. A. "Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges." *arXiv preprint arXiv:2011.06225*. 2020

## Authors



Saerom Park received the B.S. and Ph.D. degrees in industrial engineering from Seoul National University, in 2013 and 2018, respectively. Dr. Park is currently an Assistant Professor with the Department of

Convergence Security Engineering, Sungshin Women's University, Seoul, South Korea. Her research interests include kernel machines, representation learning, transfer learning, and secure machine learning.