**Hierarchical Classification**

*Abstract*

Convolutional Neural Networks (CNNs) achieves the state-of-the-art accuracy in image classification. CNNs are able to extract features that make the classification problem linearly separable. However, as the number of classes grows, needless to say a Feed-Forward Neural Network, a simple linear classifier will need a large number of parameters. In fact, it is not CNN but the classifier consumes the majority of parameters. This problem will become more and more severe as the number of classes grows in large-scale fine-grained classification. We propose a succinct architecture which not only reduces the amount of model parameters but also attains the state-of-art performance. Furthermore, our method is compatible with any off-the-shelf CNN based feature extractors like VGG and ResNet.

*Introduction*

A very large amount of research machine learning and deep learning has focused on flat classification problems. Flat classification problem refers to standard binary or multi-class

classification problems which assumes the target distribution as Bernoulli or Categorical Distribution. However, we human beings classify objects based on naturally structured taxonomy. In fact, flat classes can be organized hierarchically into a Tree or DAG (Directed Acyclic Graph). Here we only focus on Tree in which each subclass is single labeled and only belongs to one superclass. DAG structured labels are related to another trend of problem called multi-label classification where each class can have different labels. So far, we only cope with supervised learning problem assuming a pre-defined class hierarchy is given.

Most of literature about this field target very different application domains. Despite several comprehensive surveys [1,2] written to facilitate this trend, very few efforts have been made to exploit hierarchical classification with deep learning framework. [3] proposes using a neural network as a local classifier and use the output of superclass classifiers as input of subclass classifiers. [4] proposes training coarse and fine classifiers separately and predict from the average of the results from those classifiers. [5] proposes adding a stop node after nodes in a neural network and conduct a special tracing algorithm to find the path of labels. [6] proposes using features from former layers of CNNs for superclasses and later for subclasses. However, all those methods are not flexible enough to adopt arbitrary feature extractors and hierarchical labels.

*Proposed method*

We propose replacing the widely used flatten classifier with a simple tree structured classifier. We aim to perform as competent as flatten classifier in terms of classifying the most specific classes while always consuming smaller number of parameters and also predict the hierarchical structure of labels. Our tree structured classifier can be appended to any proposed off-the-shelf CNN based feature extractors like VGG and ResNet. The predicted label is no longer a single digit but a list of indices indicating its path through the tree. During training, each sample passes forward through the path of those classifiers it belongs. Since the hierarchical labels are given, output of each superclass classifier will be guided to their subclass classifier accordingly. During testing, the model will predict the label the sample belongs at each level of the tree. So far, the performance is evaluated with flat classification accuracy measurement. A test sample is considered classified correctly only if it classifies its path of the label correctly. Figure 1 shows an example tree structured classifier. Each subclass classifier is noted as C, and the trailing indices are the path of labels. The number of classes to classify is indicated by the outgoing arrows. The leaf classifier can contain arbitrary number of classes and the outgoing arrows are omitted.

We also propose an alternative training strategy that we believe can train the hierarchical structure better. During training, output of each superclass classifier will be guided to their subclass classifier stochastically based on the probability of itself belonging to their own subclasses. This stochastic infiltration method prevents samples which are not likely correctly classified in the superclass to pass down to its subclass. As they are classified better at the superclass level, they will be more likely to infiltrate into the subclass level.

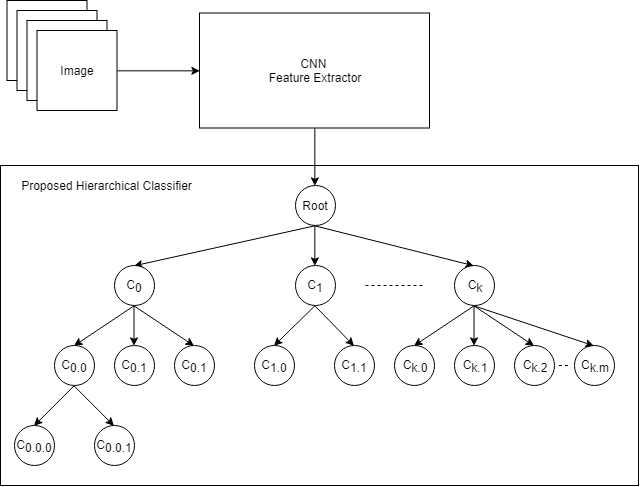


Figure 1. Architecture of proposed hierarchical classifier.

The computation cost of our method is smaller than flatten classifier in terms of both time and space. The memory consumption of our method is in O(k(sum(c\_i)) and flatten classifier is O(k(Mult(c\_i). k indicates the number of features extracted from CNN network and c\_i indicates the number of classes to classify in i\_th subclass classifier. Technically, since the extracted features are shared by all classifiers, we can pass data samples through them in parallel and mask out those that are not valid according to the given labels. Therefore, the time cost of our method is only the cost of the largest subclass classifiers.

*Application*

We conduct experiments with two dataset EMNIST (balanced) and CIFAR100. EMNIST (balanced) and CIFAR100 contain 47 and 100 categories respectively [7, 8]. EMNIST dataset contains two superclasses, digit and letter, and digit contains 10 classes while letter contains 37 classes. We also resize EMNIST data from 28x28 to 32x32 and apply standard data augmentation strategy like random horizontal flip to CIFAR100 data. We train totally 350 epochs. The SGD (Stochastic Gradient Descent) optimizer is used with scheduled decaying learning rate at 150th, 250th epochs. All experiments are run on NVIDIA TITAN Xp with batch size 128. We also experiment with two different feature extractors ResNet18 and ResNet34. Since we have multiple classifiers, the loss of each subclass classifier is scaled down by the number of subclasses at its same level. We maintain a similar loss scale as using the flatten classifier in order to avoid fine tuning learning rate. As the results shown in Table I, both our methods achieve comparable results as using the flatten classifier.

Tabel I. Flat labels classification accuracy

*Future work*

We plan to experiment our methods with more sophisticated dataset like ImageNet and DMOZ. ImageNet and its successor WebVision, although mostly classified with flatten classifier, are organized according to the WordNet hierarchy. They contain more than 1.2/1.6 million images with 1000/5000 categories. DMOZ dataset is a multilingual open-content directory of World Wide Web links. It contains 3.5 million links with one sentence description and about 1 million categories.

Apart from enlarging the problem scale, another possible track is to expand single labeled class into multi-labeled class. We can also explore unsupervised hierarchical classification. Recently, VAE (Variation AutoEncoder) has been used to cluster MNIST data with GMM (Gaussian Mixture Model) []. By switching the categorical distribution inside GMM into a series of categorical distributions with different number of classes inside Hierarchical GMM, we may be able to do unsupervised hierarchical classification.

Finally, we also plan to explore assigning different part of CNN feature extractor to different classifier. In spirit, we want to build a dictionary between classifiers and feature extractors. For instance, classifiers with more classes to classify may need more filters while smaller ones can use fewer filters. They may also share part of their own filters to control the total parameters they use.

*Conclusion*

We propose a succinct hierarchical classifier with two different training strategies. Our method attains the state-of-art image classification result by appending our classifier directly to the well-known feature extractors. Our method is able to train the hierarchical structure of classes. Meanwhile, our classifier has less computation cost than flatten classifier both in terms of time and space.

*Reference*

[1] Silla, Carlos N., and Alex A. Freitas. "A survey of hierarchical classification across different application domains." Data Mining and Knowledge Discovery 22.1-2 (2011): 31-72.

[2] Sturm, Bob L. "A survey of evaluation in music genre recognition." International Workshop on Adaptive Multimedia Retrieval. Springer, Cham, 2012.

[3] Cerri, Ricardo, Rodrigo C. Barros, and André CPLF De Carvalho. "Hierarchical multi-label classification using local neural networks." Journal of Computer and System Sciences 80.1 (2014): 39-56.

[4] Yan, Zhicheng, et al. "HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition." Proceedings of the IEEE international conference on computer vision. 2015.

[5] Wu Z, Saito S. HiNet: Hierarchical Classification with Neural Network[J]. arXiv preprint arXiv:1705.11105, 2017.

[6] Bilal, Alsallakh, et al. "Do convolutional neural networks learn class hierarchy?." IEEE transactions on visualization and computer graphics 24.1 (2018): 152-162.

[7] Cohen, Gregory, et al. "EMNIST: an extension of MNIST to handwritten letters." arXiv preprint arXiv:1702.05373 (2017).

[8] Krizhevsky, Alex, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Vol. 1. No. 4. Technical report, University of Toronto, 2009.