DEEP HIERARCHICAL MODEL FOR HIERARCHICAL SELECTIVE CLASSIFICATION AND ZERO SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Object recognition in real-world image scenes is still an open problem. With the growing number of classes, the similarity structures between them become complex and the distinction between classes blurs, which makes the classification problem particularly challenging. Standard N-way discrete classifiers treat all classes as disconnected and unrelated, and therefore unable to learn from their semantic relationships. In this work, we present a hierarchical inter-class relationship model and train it using a newly proposed probability-based loss function. Our hierarchical model provides significantly better semantic generalization ability compared to a regular N-way classifier. We further proposed an algorithm where given a probabilistic classification model it can return the input corresponding super-group based on classes hierarchy without any further learning. We deploy it in two scenarios in which super-group retrieval can be useful. The first one, selective classification, deals with the problem of low-confidence classification, wherein a model is unable to make a successful exact classification. The second, zero-shot learning problem deals with making reasonable inferences on novel classes. Extensive experiments with the two scenarios show that our proposed hierarchical model yields more accurate and meaningful super-class predictions compared to a regular N-way classifier because of its significantly better semantic generalization ability.

1 INTRODUCTION

Object recognition from images in real world scene, which requires drawing boundaries between groups of objects in a seemingly continuous space, is a highly complex problem. The visual world is populated with a vast number of diverse objects, where each object instance may pose many visual forms. Furthermore, with the growing number of classes, the similarity structures between them become complex and the distinction between classes blurs. Because of the high complexity of this task, most existing work focus on simplifying the problem to a supervised single-label classification problem. Unfortunately, since one-vs-all classifiers treat all classes as unrelated, visual recognition systems cannot transfer semantic information about learned classes.

One way of dealing with this issue is to respect the natural continuity of visual space instead of artificially partitioning it into disjoint categories. Classes have varying degrees of discriminability. Some classes have unique features while other classes might share similar features and are hence harder to distinguish from each other. Such similarity structures in the data are very valuable information that could potentially lead to improving classifiers (Frome et al., 2013; Rippel et al., 2015; Bilal et al., 2018).

We propose an approach that addresses these shortcomings by learning a hierarchical inter-class relationship model. We present a novel hierarchical probability-based loss function, which we call **soft-NLL**. This loss function gives a probability weight according to the inter-class taxonomy graph distances. We evaluate the model semantic generalization ability, which means for example that the model makes more semantically reasonable errors. Our model shows a significant improvement on the hierarchical semantic metric in an order of magnitude over existing methods, while it guarantees a minor decrease in the common top-k accuracy compared to the widely used hard-NLL loss. Our approach is independent of the specific prediction model and thus can always benefit from design progress.

Furthermore, we develop an algorithm where given a probabilistic model, based on its top-k predictions our algorithm returns the input corresponding super-group based on classes hierarchy without any further learning. We present two use cases for super-group retrieval and show that our soft-NLL outperforms in both applications. The first one, selective classification, deals with the problem of low-confidence classification, wherein a model is unable to make a successful exact classification. In this case, our algorithm returns a corresponding closest super-class. In the second scenario, the proposed method is used for the zero-shot learning problem. In this case, given a novel input, the algorithm returns its hierarchically related group, rather than generating a true unseen group.

2 RELATED WORK

Vision researchers have sought to exploit external semantic knowledge. Taxonomies, or class hierarchies were the first to be explored (Griffin & Perona, 2008; Marszalek et al., 2008). Taxonomy depends on a human-defined criteria, and thus implicitly consists of semantic knowledge. Early works (Bengio et al., 2010; Gao & Koller, 2011; Griffin & Perona, 2008; Marszalek et al., 2008) exploit the tree structure for efficient branch-and-bound training and prediction, while a few use taxonomies as sources of relational knowledge between categories (Deng et al., 2014; Hwang, 2013). Our method shares the same goal with the latter group of work.

Deep Convolutional Neural Networks which learn high-level features and a N-way flat softmax classifier have recently gained enormous popularity for their impressive performance on a number of visual recognition tasks. Since Krizhevsky et al. (2012) won the ImageNet ILSVRC challenge 2012 (Deng et al., 2009), many variants of this model have been proposed. GoogLeNet (Szegedy et al., 2015), Resnet (He et al., 2015) and VGGNet (Simonyan & Zisserman, 2014) focus on increasing the network depth by adding more convolutional layers to the original model. These models generalize better than smaller ones. Our goal is to benefit form the progress of CNN designs.

Some existing work has explored various approaches to learning semantic information in deep neural networks. One related approach is to learn a joint representation of images and labels in an embedding space. Frome et al. (2013); Norouzi et al. (2013) learn semantic relationships between classes using a semantic word embedding model, where each class represented by an embedding vector. In parallel, they pre-train a deep neural network for visual object recognition. Then, both methods develop a specific mapping of the images into the semantic embedding space. Both methods evaluated their semantic ability using a semantic metric. Moreover, they employed their models for zero-shot learning. However, Frome et al. (2013) show modest performance in its semantic relevance meaning because the word2vec embeddings are weak in the sense of capturing the inherent hierarchical semantics. Furthermore, it imposes structure only on the output space, and structure on the learned space is not explicitly enforced. However, our goal is to transfer direct inter-class hierarchical data to the learned space.

Yan et al. (2015) propose a two-staged CNN architecture named HD-CNN, which leverages the taxonomy to separate the categories into easy coarse-grained ones and confusing fine-grained ones, trained in separate networks. However, such separate learning for coarse and fine-grained categories results in larger memory footprints, while it improves performance with only a modest rate. Recently Bilal et al. (2018) designed a hierarchy-aware CNNs that accelerate model convergence and alleviate overfitting. They select AlexNet (Krizhevsky et al., 2012) as a reference architecture and created branches from multilevel deep features to perform group-level classification and back-propagate group error. The number of groups is gradually increased along the depth of the inner layers. However this scheme demands extensive analysis and consequently it cannot easily benefit from the progress of CNN designs. Moreover, the latter two works did not present an evaluation of how predictions are semantically meaningful.

Our goals are to directly exploit the class hierarchy information while benefit form the progress of CNN designs. We focus on large scale multi-class problems. In contrast to most previous works we evaluate the semantic ability of our model according to a semantic metric. We further show that such a metric is meaningful when we compare our soft semantic model to a regular hard-model on two applications.

3 THE PROPOSED METHOD

3.1 SOFT NEGATIVE LOG LIKELIHOOD LOSS FORMULATION

The problem we address in this paper is a supervised multiclass classification. Following common setup, we assume the input $X \in \mathcal{X}$ and label $Y \in \mathcal{Y} = \{1, ..., \mathcal{C}\}$ can be modelled by a joint distribution $\pi(X, Y)$. The labels are organized in a taxonomy graph G, where each directed edge represents an 'is a' relation, and the distance between nodes reflects a semantic relationship between them. Our goal is to model these semantic relationships, so that its top-k predictions will be closely related to the true label according to G.

The standard objective formulation used in the context of deep learning for a multiclass classification problem is the negative log likelihood (NLL) loss, which is also referred to as the cross entropy loss with the probability class indicator. Given a probabilistic model $\hat{\pi}(Y|X)$ and a sample (x_0, y_0), NLL loss is defined by

$$l(x_0, y_0) = -\sum_{y=1}^{\mathcal{C}} \pi(y|x_0) \log \hat{\pi}(y|x_0) \text{ , where } \pi(y|x_0) = \mathbf{1}_{(y=y_0)}$$

This formulation assigns the whole probability weight to a single class with all other classes having zero weight. This approach artificially partitions the visual space into disjoint categories and does not take into account semantic relationships between classes.

We propose to use a 'soft' (instead of 'hard') model $\pi(y|x)$ as a semantic relationship measure. This model can be learned as the class graph taxonomy G. More formally, let $\Pi \in \mathbb{R}^{\mathcal{C} \times \mathcal{C}}$ represent a semantic probability taxonomy relationship (or distance) map, where $\Pi_{i,j} \triangleq \pi(i|j)$. We define the soft NLL loss as:

$$l(x_0, y_0) = -\sum_{y=1}^{\mathcal{C}} \Pi_{y_0,y} \log \hat{\pi}(y|x_0) \tag{1}$$

Π is a class row-wise i.e. each ith row resembles the ith true-class probabilities, which are inversely proportional to the on-the-graph distance defined,

$$d_G(\text{class node i}, \text{class node j}) \triangleq \text{shortest path length between nodes}$$

Details about hyper-parameters selection is described in Section 4.4.

3.2 SUPER-GROUP RETRIEVAL ALGORITHM

In this section we present a super-group retrieval algorithm. Returning a super-group can be useful in hard scenarios for example, when given Object samples with high confusion between similar classes or when making an inference for novel classes. Soft-NLL has a better semantic generalization ability as we show below in Section 5.1. Its top-k predictions are more accurate with respect to the true class. Therefore, they can be used as a coarse-grained classifier and retrieve a better super group, compared to standard models.

Given the model top-k predictions the algorithm follows these steps:

1. **Clean the top-k predictions from less relevant predictions:**
   The model top-k predictions may be noisy. Based on the graph taxonomy we can generate a cleaner subset of the top-k predictions.
   More specifically, given the top-k predictions, we calculate a subset of them with each ith class l-hCorrectSet as follows,
   $$S_i(l, k) = \{\text{top-k predictions}\} \cap \{\text{l-hCorrectSet}_i\}$$
   Where for a given class its l-hCorrectSet is the l nearest set of classes gathered from the graph taxonomy as defined by Frome et al. (2013). We choose the highest matching set, $S_C(l, k)$ where $C = \underset{i}{\operatorname{argmax}} |S_i(l, k)|$.
2. **Generate super-group candidates:**
   The set $\{A_S\}$ equals to super-group nodes which are ancestors for each class in $S_C(l, k)$.

3. **Choose the most specific super-group:**
   The most specific super group is the $a \in \{A_S\}$ which is the lowest common ancestor (LCA) of $S_C(l,k)$ generated by $LCA = \min_{a \in \{A_S\}} \sum_{s \in S_C(l,k)} d_G(a,s)$ [1].

For example, suppose we have a simple taxonomy as illustrated in Figure 1, where the leaf nodes are valid classes and the other nodes are theirs super-classes. Suppose we got a sample from class 1, and our top-3 predictions are: 0,1 and 6. Each class has its own 3-hCorrestSet, where the true class hCorrestSet gives the maximal intersection set with the top3 predication i.e. $S_1(3,3) = 0,1$. We have three super-group candidates: A,B,D where D is the LCA generated super-group. In Section 4.5 we discuss ways of choosing the algorithm's hyper-parameters k and l.

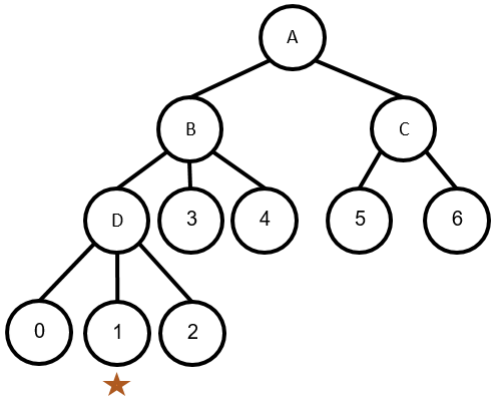

Figure 1: Super-class generation example

4 EXPERIMENTS

The objective of this work is to develop a method for generating a semantic vision model i.e. a model which makes semantically relevant predictions even when it makes errors. Moreover, we show its advantage in super group generation in two different scenarios of selective classification and zero shot learning.

4.1 DATASET

ImageNet is the largest publicly available labeled image dataset, encompassing more than 14 million images that belong to more than 21K object categories (Deng et al., 2009). The object categories are nouns in the WordNet database of the English language (Miller, 1995) . A fundamental property of WordNet is its hierarchical organization of concepts.

In our experiments we adopt a subset of ImageNet the ILSVRC12 dataset, which gather 1K classes that are randomly selected according to certain criteria that aim to reduce ambiguity. It consists of approximately 1.3 million training images and 50k validation labeled images from each category.

As illustrated in Figure 2, ILSVRC12 taxonomy is a complex directed graph, there may be multiple routes from the root to the leaf nodes. Its minimal spanning tree which contains the leaf nodes is highly not balanced, where routes depth range is between 6 and 18.

4.2 METRICS

We use several metrics in the evaluation process, each was averaged on test images. The *flat hit@k* is a standard metric 0/1 error used in large scale classification problems which returns a success if

[1] The taxonomy is a graph i.e. there may be more than one route connecting two nodes, therefore generating LCA in the mentioned way is better than by taking the deepest ancestor, $LCA = \max_{a \in \{A_S\}} \sum d_G(a, root)$.

the true label resides in the top k predictions. To measure the semantic quality of predictions beyond the true label, we also evaluate with the hierarchical-precision@k (*hp@k*) introduced in Frome et al. (2013), which is a semantic relevance of the model top-k predictions. It is computed as a fraction of the top-k predictions that overlap with the true class *k-hCorrectSet*:

$$hp@k = \frac{1}{N}\sum_{n=1}^{N}\frac{\text{number of model's top-k predictions in k-hCorrectSet for image i}}{k}$$

where for a true class k-hCorrectSet is the k nearest classes set gathered from the graph taxonomy as described in detail in Frome et al. (2013).

In order to evaluate super-group (SG) generation succession we adopt two metrics, $SG\text{-}hit$ is a 0/1 error metric, where given a predicated label, it returns success if there is a path between the SG candidate and the true class. The latter metric may be useless because the algorithm may return the root, therefore we calculate how much a candidate is specific by measuring its distance to the true class i.e. $SG\text{-}specificity = d_G(SG, truelabel)$. The last two metrics are based on the graph taxonomy. We check the SG generation succession in two scenarios: selective classification and zero shot learning.

4.3 Training Details

In our experiments we are comparing semantic ability of different loss paradigms while using the same core vision model. We train Resnet50 proposed by He et al. (2015) and Alexnet presented in Krizhevsky et al. (2012) covnvolutional networks as our core vision models. We use the data preprocessing, training procedures, and hyperparameters as described in these papers.

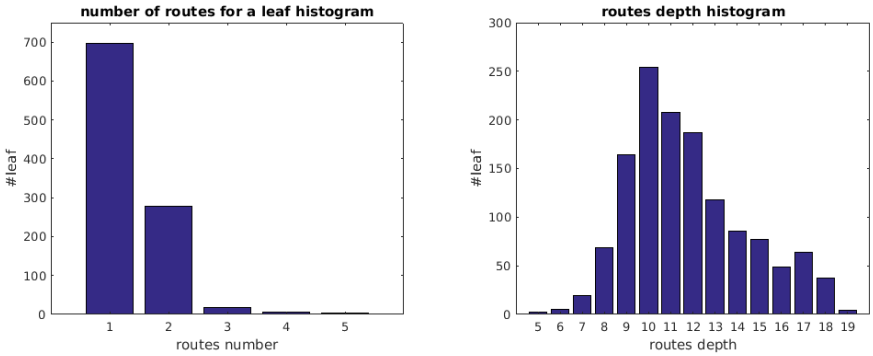

Figure 2: taxonomy statistics: number of different routes from root to leaf histogram (left), leaf routes depth histogram, remark: multiple routes per leaf included (right).

4.4 Soft Hierarchical Based Probabilities: Choosing Hyper-Parameters

In this section we deal with determining soft labels hierarchical probabilities hyper-parameters Π which is defined in Equation 1. Given the ith row which indicates the true class, the probabilities of all jth corresponding classes $\Pi_{i,j}$ satisfy

$$\Pi_{i,i}/\Pi_{i,j} = f_d,\ \forall\, i,j \ \text{ s.t } \ d_G(i,j) = d \tag{2}$$

where $\Pi_{i,i}$ is the true class probability. That is the probabilities of all jth classes with equal distance d to the ith true class, are degrade by a constant *shrink factor* f_d in relation to the true ith class probability. The values f_d are in direct relation to the on-graph distance metric. For instance, Table 1 describes the probability values given for the class 'tabby' with the shrink vector $[f_2, f_{3,4}, f_{5,6}] = [10, 100, 1000]$, where $f_{3,4}$ indicates that both distance level 3 and level 4 gets the same value.

We empirically investigate the impact of different sets of f values on the flat-hit and hierarchical-precision metrics on the ILSVRC 2012 1K dataset. To provide a baseline for comparison, we compared the performance of our model to a hard-NLL model, where all models trained with Renset50.

Table 1: Soft hierarchical probabilities for a specific class ('tabby') with the shrink vector $[f_2, f_{3,4}, f_{5,6}] = [10, 100, 1K]$

distance level $[d]$	#classes	shrink factor $[f_d]$	Probability weights
0	1	1	61.9%
2	4	10	6.19%
3	0	-	-
4	9	100	0.619%
5	29	1K	0.0619%
6	94	1K	0.0619%
≥ 7	863	∞	0

Table 2: Shirk factor f_d impact on model performance on ImageNet ILSVRC12 1K validation set trained with Resnet50

Model name	**Flat hit@k (%)**				**Hierarchical precision@k**			
	1	**2**	**5**	**10**	**2**	**5**	**10**	**20**
[0] hard-NLL	75.85	85.79	92.81	95.75	0.571	0.423	0.377	0.360
[1] $f_2 = 10$	75.12	84.47	91.48	94.85	0.696	0.566	0.482	0.427
[2] $f_2, f_{3,4} = 10, 100$	74.74	84.24	90.81	94.04	0.706	0.657	0.654	0.620
[3] $f_2, f_{3,4}, f_{5,6} = 10, 100, 1\text{K}$	74.59	84.04	90.81	93.91	0.702	0.655	0.662	0.684
[4] $f_2, f_{3,4}, f_{5,6} = 0, 100, 1\text{K}$	75.64	85.49	92.01	94.92	0.570	0.519	0.585	0.631
[5] $f_2, f_{3,4}, f_{5,6} = 25, 250, 1\text{K}$	75.73	85.63	92.04	95.95	0.683	0.615	0.614	0.623

Table 2 shows results for the mentioned models on both the flat and hierarchical metrics and Figure 3 (left) visualizes these results. First, Comparing models 1-3 to the baseline 0 shows that adding probability weight to far classes increases the semantic metric performance hp@k for growing k values. Second, if we skip distance levels the performance is degraded as can be showed in model 4, where we skip all classes with distance level d=2 giving these classes a zero probability. Moreover, using small weights yields a model which has only a minor decrease in flat-hit while still boosting the hp@k as can be seen in model 5 relative to model 3.

4.5 SUPER-GROUP RETRIEVAL: CHOOSING HYPER-PARAMETERS

The algorithm for super-group generation incorporates two hyper-parameters k and l, i.e the top-k predictions and the extent of the hCorrectSet. Clearly, increasing k and l improves SG-hit while hurting SG-Specificity. In this section we deal with choosing these parameters; in other words we investigate the impact of a combination of these parameters on super-group performance.

The use of super-group generation is for hard scenarios where the model is probable to miss its top-1 or when making an inference for novel classes. In this section we focus in the first case while taking into consideration the samples where the standard sotmax-NLL makes an error in it's top1 prediction. The experiments were performed with the same trained soft-NLL model used for results in Section 5.1. As baseline we compared the performance of our model to a standard softmax model. Both models were trained with resnet50 topology.

First, we observe that k should be determined adaptively according to the uncertainty in the model top-k predictions. This uncertainty can be measured by the model top-k probability coverage defined,

$$p_\theta = \sum_{i \in top\text{-}k} \hat{p}_i$$

where $\hat{p}_i$ is the model softmax response for the ith class. That is, if a small k say $k = 5$ gives high probability coverage say $p_\theta = 0.95$ it is considered as high certainty in the predictions. This adaptive approach is needed because increasing k in cases when p_θ is high may degrade $SG\text{-}Specificity$,

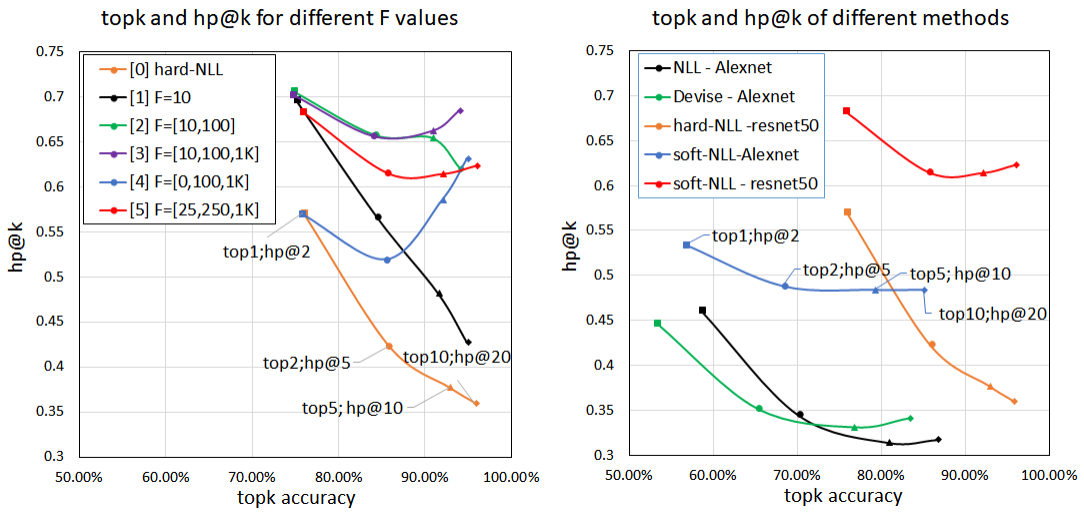

Figure 3: Visualization of flat-hit@k and hp@k metrics calculated on ILSVRC12 1K validation dataset. (Left) Performance for different f_d values trained model relative to baseline hard-NLL. (Right) Comparing different loss paradigms, where soft-NLL resnet50 is the same model like 5 in left figure. Each specific set of flat-hit and hp@k point is indicated by a different symbol. The sets are displayed near it's corresponding symbol for one curve in each figure and are equivalent to all other curves. We would like to get curves which are right and up i.e. with better topk accuracy and better hp@k

while when k values are low we see degrade in hitting the super-group. Thus, in the experiments we chose to take $k \geq 5$ while demanding that $p_\theta > p_{tresh}$, where p_{thresh} is a fixed threshold.

Figure 4 displays $SG\text{-}Specificity$ vs $SG\text{-}hit$ for different values of k-probability coverage and l values. The curves gives a systematic way of choosing the best k and l. Given a demand on one metric dictates the sets of parameter to get the best other metric, e.g for $SG\text{-}hit = 0.7$ the best lowest $SG\text{-}Specificity$ is given for soft-NLL with $l = 20$, for $SG\text{-}Specificity = 3$ the highest $SG\text{-}hit$ is given for soft-NLL with $l = 20$. Moreover, Soft-NLL outperformed the standard softmax model with better hit and specificity for each such demand.

5 Results

5.1 Semantic relevance

This section presents flat and hierarchical results on the ILSVRC 2012 1K dataset. We compare our soft-NLL to standard softmax-NLL using two architectures Resnet50 and Alexnet, and to DeVise approach presented in Frome et al. (2013) trained with Alexnet. The hyper-parameter search setup used with soft-NLL loss is specified in details in Section 4.4

Table 3 shows results for the mentioned models on both the flat and hierarchical metrics. Figure 3 (right) visualizes these results. The soft-NLL shows a significant improvement in its semantic generalization ability while exhibiting only a minor decrease in the top-k accuracy compared to standard softmax-NLL on a same deep topologies. For Resnet50 at k=5 the soft-NLL gives 0.615 while hard-NLL gives 0.423 model which is about 45% relative improvement, at k=10 soft-NLL and hard-NLL gives 0.614 and 0.377 respectively which is a 63% relative improvement. Our relative improvement is in order of magnitude compared with the improvement given by DeVise, for k=5 and k=10 DeVise Frome et al. (2013) method gives a relative improvement of about 2% and 5% respectively over the standard model.

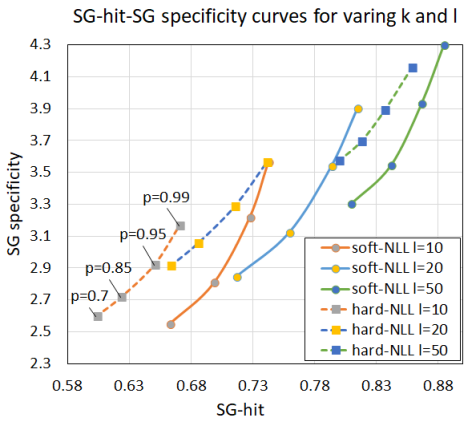

Figure 4: SG-hit and SG-specificity metrics calculated on top-1 miss cases of hard-NLL model in ILSVRC12 1K validation dataset. The effect of varying k and l on SG generation for soft-NLL and hard-NLL models trained with resnet50. For each l we use a set of k-probability coverage: 0.7,0.85,0.95,0.99 which are arranged from left to right for each curve. In order to do well We would like to get curves which are right and bottom i.e. with better hit and more specific

Table 3: Comparison of model performance on test set (ImageNet ILSVRC12 1K validation set)

Model name	**Flat hit@k (%)**				**Hierarchical precision@k**			
	1	**2**	**5**	**10**	**2**	**5**	**10**	**20**
NLL-Alexnet	58.6	70.1	80.8	86.7	0.461	0.345	0.314	0.317
DeVise(dim=500)	53.2	65.2	76.7	83.3	0.447	0.352	0.331	0.341
NLL-Resnet50	75.9	85.8	92.8	95.8	0.571	0.423	0.377	0.360
soft-NLL-Alexnet	57.7	69.2	79.7	85.6	0.542	0.493	0.491	0.490
soft-NLL-Resnet50	75.7	85.6	92.0	96.0	0.683	0.615	0.614	0.623

5.2 SELECTIVE CLASSIFICATION

Soft-NLL can be used to make reasonable inferences given hard candidate cases as shown in Section 5.1. Because it has a better semantic generalization ability, it can return the super group when it is improbable that we make successful exact classification. In this section we discuss the selection, i.e. how to identify the cases where top-1 prediction is improbable to make hit and in such cases propose to use the benefits of out model to return the super-class.

When we consider a large-scale problem ($|Y| \gg 1$), the standard approach is to consider the top-k predictions where $k > 1$. But in practice, given an image a desired need may be to get a single label and not k noisy predictions. On the other hand, considering only the top-1 prediction includes many mistakes even in the state-of-the-art models.

To mitigate the need in a single prediction we propose the following selection mechanism: select top-1 prediction only when it seemed to be 'promised' in success and in harder cases retrieves the super group (SG) in better accuracy. Our selection with guaranteed risk control problem is formalized like Geifman & El-Yaniv (2017), where instead of abstaining the predication we propose to retrieve the super group as follows,

$$(f, g) \triangleq \begin{cases} \textit{top-1}(f_{NLL}(x)), & \text{if } g(x) = 1 \\ SG(\textit{top-k}(f_{soft-NLL}(x))), & \text{if } g(x) = 0 \end{cases}$$

where, $f_{NLL}(x)$,$f_{soft-NLL}(x)$ are the standard and semantic classifiers and $g : X \to \{0, 1\}$ is a selection function.

As showed in Geifman & El-Yaniv (2017) the top-1 softmax model response can be used to control the guaranteed error that is,

$$g(x) = \begin{cases} 1, & \text{if } SR_{top\text{-}1}(x) > \theta \\ 0, & \text{otherwise} \end{cases}$$

where θ is determined according the desired error of top-1 predictions, e.g if the model gives a total 75% and the desired error is 5% we should take θ_o to guarantee this risk.

We calculated the risk-coverage curve for many θ values obtained for ILSVRC12 validation dataset. For each value we calculate top-1 error in the guaranteed-controlled set, and super group generation ability metrics in the complementary set as shown in 5. For example, for 65% coverage NLL gives about 92% top-1 accuracy, and in the complementary set (35% coverage) we get $SG\text{-}hit$ of 87% and 3.45 $SG\text{-}specificity$ using soft-NLL model, which outperforms standard NLL SG metrics for all θ-coverage values.

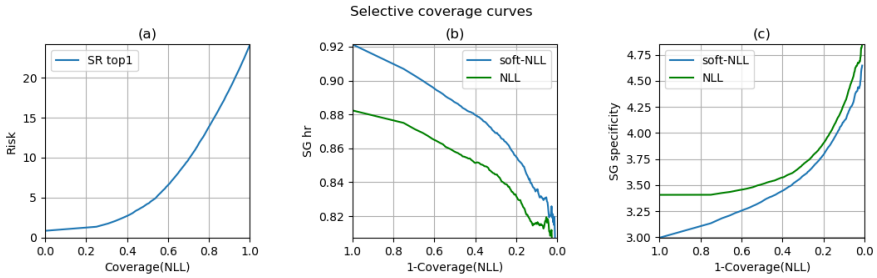

Figure 5: Selective coverage curves, (a) top-1 risk using standard NLL softmax, (b-c) super-group hr and specificity in the complementary sets. The x-axes of (b)-(c) was inverted for relation to (a) easier.

5.3 ZERO SHOT LEARNING

Soft-NLL model can be generalized as a coarse-grained descriptor to return a better super group. In this manner, our model can make reasonable inferences about candidate it has never visually observed and given a novel class return its super-class. To test this hypothesis, we extracted images from the ImageNet 2011 21K dataset with labels that were not included in the ILSVRC 2012 1K dataset on which our model was trained.

The zero-shot experiments were performed with the same trained Soft-NLL model used for results in Section 5.1. To provide a stronger baseline for comparison, we compared the performance of our model to standard softmax model. We again evaluate super group succession with SG-hit and SG-specificity metrics.

To quantify the performance of the model, we constructed from ImageNet 2011 21K zero-shot data two test data sets as defined in Frome et al. (2013). The dataset are defined with increasing difficulty based on the image labels' graph distance from ILSVRC 2012 1K labels in the ImageNet label hierarchy. '2-hop' is the easy dataset which comprised of 1,548 labels that are within two tree hops of the training labels, making them visually and semantically similar to the training set. A more difficult '3-hop' dataset comprised of 5,989 labels was constructed in the same manner. The datasets contains 1.3 million and 4.6 million images respectively.

Figure 6 compared between soft-NLL and hard-NLL $SG\text{-}Specificity$, $SG\text{-}hit$ metrics for different values of k-probability coverage and l values. Soft-NLL outperformed hard-NLL on both metrics on both datasets. It gets a better hit for each demand in the SG-specificity metric and a better specific super-class for each demand in the hit meter. In the harder '3-hop' dataset we get some degradation in the SG-hit metric but get a comparable results on the SG-specificity metric relative to '2-hop' dataset. The specificity of predictions is reasonable taking into consider the high degree of classes as shown in 2.

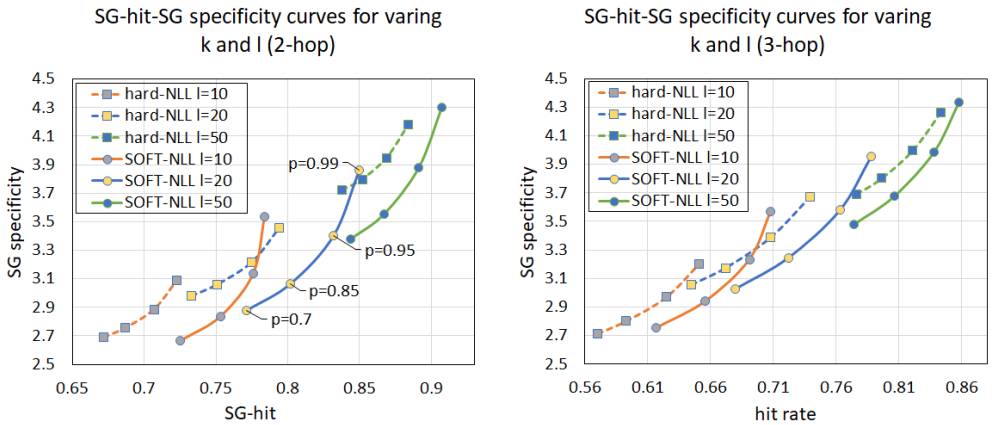

Figure 6: SG-hit and SG-specificity metrics calculated on zero-shot '2-hop'(left) and '3-hop' imagenet11 datasets. Comparision of hard-NLL and soft-NLL performance for varing k and l, where for each l we use a set of k-probability coverage: 0.7,0.85,0.95,0.99 which are arranged from left to right for each curve. In order to do well We would like to get curves which are right and bottom i.e. with better hit and more specific

6 Conclusion

In this work we have shown that our hierarchical probability based soft-NLL loss can be trained to give a comparable performance to the state-of-the-art hard-NLL softmax based model on flat classification metric, while simultaneously making more semantically reasonable errors, which is indicated by its significant improved performance on a hierarchical label metric relative to existing methods (Frome et al., 2013). Our model's learning process has no additional cost in training and classification time. This approach is independent of the specific deep architecture and thus can always benefit from design progress.

Furthermore, we present an algorithm where given a probabilistic model, based on its top-k predictions our algorithm returns the input corresponding super-group based on classes hierarchy without any further learning. We have shown that our soft-NLL retrieve a better coarse category in two hard use cases, where the model is probable to make a miss and for zero-shot scenario which deals with making an inference for novel classes.

Moving forward, we are experimenting an adaptive use of our loss function according to the confusion between classes. In some cases a hierarchical closeness may not coincide with visual closeness which is the integral issue when trying to differentiate between visual objects.

References

Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, pp. 163–171, 2010.

Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do Convolutional Neural Networks Learn Class Hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–165, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. *Cvpr*, pp. 248–255, 2009.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pp. 48–64. Springer, 2014.

Andrea Frome, Gs Corrado, Jonathon Shlens, Samy Bengio, Jeff Dean, and Marc Ranzato. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26*, pp. 2121–2129. 2013.

Tianshi Gao and Daphne Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *Computer Vision (ICCV), IEEE International Conference on*, pp. 2072–2079, 2011.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4878–4887. Curran Associates, Inc., 2017.

Gregory Griffin and Pietro Perona. Learning and using taxonomies for fast visual categorization. In *Computer Vision and Pattern Recognition. CVPR. IEEE Conference on*, pp. 1–8. IEEE, 2008.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2015.

Sung Ju Hwang. Discriminative object categorization with external semantic knowledge. 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

Marcin Marszalek, Cordelia Schmid, David A Forsyth, Philip H S Torr, Andrew Zisserman, and Marcin Marszaa lek. Constructing category hierarchies for visual recognition. In *European conference on computer vision*, pp. 479–491. Springer, 2008.

George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. pp. 1–9, 2013.

Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. pp. 1–15, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 2740–2748, 2015.