Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks

Luca Bertinetto* Romain Mueller* Konstantinos Tertikas Sina Samangooei Nicholas A. Lord*
{luca.bertinetto, romain.mueller, konstantinos.tertikas, sina, nick.lord}@five.ai
www.five.ai

Abstract

Deep neural networks have improved image classification dramatically over the past decade, but have done so by focusing on performance measures that treat all classes other than the ground truth as equally wrong. This has led to a situation in which mistakes are less likely to be made than before, but are equally likely to be absurd or catastrophic when they do occur. Past works have recognised and tried to address this issue of mistake severity, often by using graph distances in class hierarchies, but this has largely been neglected since the advent of the current deep learning era in computer vision. In this paper, we aim to renew interest in this problem by reviewing past approaches and proposing two simple modifications of the cross-entropy loss which outperform the prior art under several metrics on two large datasets with complex class hierarchies: tieredImageNet and iNaturalist'19.

1. Introduction

Image classification networks have improved greatly over recent years, but generalisation remains imperfect, and test-time errors do of course occur. Conventionally, such errors are defined with respect to a single ground-truth class and reported using one or more top-k measures (k typically set to 1 or 5). However, this practice imposes certain notions of what it means to make a mistake, including treating all classes other than the "true" label as equally wrong. This may not actually correspond to our intuitions about desired classifier behaviour, and for some applications this point may prove crucial. Take the example of an autonomous vehicle observing an object on the side of the road: whatever measure of classifier performance we use, we can certainly agree that mistaking a lamppost for a tree is less of a problem than mistaking a person for a tree, as such a mistake would have crucial implications in terms both of prediction and planning. If we want to take such considerations into account, we must incorporate a nontrivial model of the relationships between classes, and accordingly rethink more broadly what it means for a network to "make a mistake". One natural and convenient way of representing these class relationships is through a taxonomic hierarchy tree.

*Equal contribution.

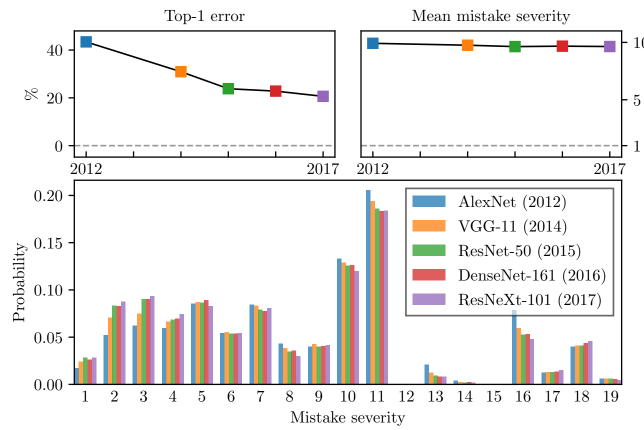

Figure 1: Top-1 error and distribution of mistakes w.r.t. the WordNet hierarchy for well-known deep neural network architectures on ImageNet/ILSVRC-12: see text for definition of mistake severity. The top-1 error has seen a spectacular improvement in the last few years, but though the *number* of mistakes has decreased in absolute terms, the *severity* of the mistakes made has remained fairly unchanged. Dashed lines denote the best achievable value of each metric.

This idea is not new. In fact, it was once fairly common across various machine learning application domains to consider class hierarchy when designing classifiers, as surveyed in Silla & Freitas [32]. That work assembled and categorised a large collection of hierarchical classification problems and algorithms, and suggested widely applicable measures for quantifying classifier performance in the context of a given class hierarchy. The authors noted that the hierarchy-informed classifiers of the era typically empirically outperformed "flat" (*i.e.* hierarchy-agnostic) classifiers even under standard metrics, with the performance gap increasing further under the suggested hierarchical metrics. Furthermore, class hierarchy is at the core of the ImageNet dataset: as detailed in Deng *et al*. [10], it was constructed directly from WordNet [20], itself a hierarchy originally designed solely to represent semantic relationships between words. Shortly after ImageNet's introduction, works such as Deng *et al*. [9], Zhao *et al*. [44], and Verma *et al*. [37] explicitly noted that the underpinning WordNet hierarchy suggested a way of quantifying the severity of mistakes, and experimented with hierarchical cost minimisation. Likewise, Deng *et al*. [8] presented a straightforward method for using a hierarchy-derived similarity matrix to define a more semantically meaningful compatibility function for image retrieval. Despite this initial surge of interest and the promising results accompanying it, the community effectively discarded hierarchical measures after deciding that they were not communicating substantially different information about classifier performance than top-1 and top-5 accuracies[1]. When the celebrated results in Krizhevsky *et al*. [16] were reported in flat top-k terms only, the precedent was firmly set for the work which followed in the deep learning era of image classification. Interest in optimising hierarchical performance measures waned accordingly.

We argue here that this problem is ripe for revisitation, and we begin by pointing to Fig. 1. Here, a *mistake* is defined as a top-1 prediction which differs from the ground-truth class, and the *severity* of such a mistake is the height of the lowest common ancestor of the predicted and ground-truth classes in the hierarchy. We see that while the flat top-1 accuracies of state-of-the-art classifiers have improved to impressive levels over the years, the distributions of the severities of the errors that *are* made have changed very *little* over this time. We hypothesise that this is due, at least in part, to the scarcity of modern learning methods which attempt to exploit prior information and preferences about class relationships in the interest of "making better mistakes", whether this information is sourced from an offline taxonomy or otherwise. The few exceptions of which we are aware include Frome *et al*. [12], Wu *et al*. [38], Barz & Denzler [3], and a passing mention in Redmon & Farhadi [27]. In Sec. 2, we suggest a framework for thinking about these pieces of work, their predecessors, and some of their conceptual relatives.

The contributions of this work are as follows:

1. We review relevant literature within an explanatory framework which unifies a fairly disjoint prior art.
2. Building on the perspective gained from the preceding, we propose two methods that are both simple and effective at leveraging class hierarchies. Each uses a one-parameter drop-in generalisation of the standard cross-entropy loss. These loss variants can be tuned to produce different empirical tradeoffs between top-k and hierarchical performance measures, and reduce to the standard setup in their respective limits.
3. We perform an extensive experimental evaluation to both demonstrate the effectiveness of the said methods compared to prior art and to encourage future work.

The PyTorch [23] code for all experiments will be made available at github.com/fiveai/making-better-mistakes.

2. Framework and related work

We first suggest a simple framework for thinking about methods relevant to the problem of making better mistakes on image classification, beginning with the standard supervised setup. Consider a training set $\mathcal{S} = \{(x_i, C_i)\}_{i=1,\ldots,N}$ which pairs N images $x_i \in \mathcal{I}$ with class labels $C_i \in \mathcal{C}$. A network architecture implements the predictor function $\phi(x; \theta)$, whose parameters θ are learned by minimising

$$\frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\phi(x_i; \theta), y(C_i)) + \mathcal{R}(\theta), \qquad (1)$$

where the loss function $\mathcal{L}$ compares the predictor's output $\phi(x_i; \theta)$ to an embedded representation $y(C_i)$ of each example's class, and $\mathcal{R}$ is a regulariser.

Under common choices such as cross-entropy for $\mathcal{L}$ and one-hot embedding for y, it is easy to see that the framework is agnostic of relationships between classes. The question is how such class relationships $\mathcal{H}$ can be incorporated into the loss in Eqn. 1. We identify the following three approaches:

1. Replacing class representation $y(C)$ with an alternate embedding $y^{\mathcal{H}}(C)$. Such "label-embedding" methods, discussed in Sec. 2.1, can draw their embedding both from taxonomic hierarchies and alternative sources.
2. Altering the loss function $\mathcal{L}$ in terms of its arguments to produce $\mathcal{L}^{\mathcal{H}}(\phi(x; \theta), y(C))$, *i.e.* making the penalty assigned to a given output distribution and embedded label dependent on $\mathcal{H}$. Methods using these "hierarchical losses" are covered in Sec. 2.2.
3. Altering the function $\phi(x; \theta)$ to $\phi^{\mathcal{H}}(x; \theta)$, *i.e.* making hierarchically-informed architectural changes to the network, generally with the hope of introducing a favourable inductive bias. We cover these "hierarchical architectures" in Sec. 2.3 .

While a regulariser $\mathcal{R}^{\mathcal{H}}$ is certainly feasible, it is curiously rare in practice: [44] is the only example we know of.

2.1. Label-embedding methods

These methods map class labels to vectors whose relative locations represent semantic relationships, and optimise a loss on these embedded vectors. The DeViSE method of Frome *et al*. [12] maps target classes onto a unit hypersphere, assigning terms with similar contexts to similar

[1]From Russakovsky *et al*. [31]: *"[..] we found that all three measures of error (top-5, top-1, and hierarchical) produced the same ordering of results. Thus, since ILSVRC2012 we have been exclusively using the top-5 metric which is the simplest and most suitable to the dataset."*

representations through analysis of unannotated Wikipedia text [18]. The loss function is a ranking loss which penalises the extent to which the output is more cosine-similar to false label embeddings than to the correct one. They learn a linear mapping from a pre-trained visual feature pipeline to the embedded labels, then fine-tune the visual pipeline.

Romera-Paredes & Torr [29] note that their solution for learning an analogous linear mapping for zero-shot classification should easily extend to accommodating these sorts of embeddings. In Hinton *et al*. [14, Sec. 2], the role of the label embedding function is played by a temperature-scaled pre-existing classifier ensemble. This ensemble is “distilled” into a smaller DNN through cross-entropy minimisation against the ensemble's output For zero-shot classification, Xian *et al*. [39] experiment with various independent embedding methods, as is also done in Akata *et al*. [2]: annotated attributes, word2vec [19], glove [24], and the WordNet hierarchy. Their ranking loss function is functionally equivalent to that in Frome *et al*. [12], and they learn a choice of linear mappings to these representations from the output of a fixed CNN. Barz & Denzler [3] present an embedding algorithm which maps examples onto a hypersphere such that all distances represent similarities derived from lowest common ancestor (LCA) height in a given hierarchy tree. They proceed by minimising the sum of two rather different losses: (1) a linear loss based on cosine distance to the embedded class vectors and (2) the standard cross-entropy loss on the output of a fully-connected layer added after the embedding layer.

2.2. Hierarchical losses

In these methods, the loss function itself is parametrised by the class hierarchy such that a higher penalty is assigned to the prediction of a more distant relative of the true label. Deng *et al*. [9] simply train kNN- and SVM-based classifiers to minimise the expected WordNet LCA height directly. Zhao *et al*. [44] modify standard multi-class logistic regression by replacing the output class probabilities with normalised class-similarity-weighted sums. They also regularise feature selection using an “overlapping-group-lasso penalty” which encourages the use of similar features for closely related classes, a rare example of a hierarchical regulariser. Verma *et al*. [37] incorporate normalised LCA height into a “context-sensitive loss function” while learning a separate metric at each node in a taxonomy tree for nearest-neighbour classification. Wu *et al*. [38] implement granular classification of food images by sharing a standard deep network backbone between multiple fully-connected layers, each one outputting class probabilities at its respective hierarchy level. A separate label propagation step is used to smooth inconsistencies in the resulting marginal probabilities. Alsallakh *et al*. [4] likewise use a standard deep architecture as their starting point, but instead add branches strategically to intermediate pipeline stages. They thereby force the net to classify into offline-determined superclasses at the respective levels, backpropagating error in these intermediate predictions accordingly. At test time, these additions are simply discarded.

2.3. Hierarchical architectures

These methods attempt to incorporate class hierarchy into the classifier architecture without necessarily changing the loss function otherwise. The core idea is to “divide and conquer” at the structural level, with the classifier assigning inputs to superclasses at earlier layers and making fine-grained distinctions at later ones. In the context of language models, it was noted at least as early as Goodman [13] that classification with respect to an IS-A hierarchy tree could be formulated as a tree of classifiers outputting conditional probabilities, with the product of the conditionals along a given leaf's ancestry representing its posterior; motivated by efficiency, Morin & Bengio [21] applied this observation to a binary hierarchy derived from WordNet. Redmon & Farhadi [27] propose a modern deep-learning variant of this framework in the design of the YOLOv2 object detection and classification system. Using a version of WordNet pruned into a tree, they effectively train a conditional classifier at every parent node in the tree by using one softmax layer per sibling group and training under the usual cross-entropy loss over leaf posteriors. While their main aim is to enable the integration of the COCO detection dataset with ImageNet, they suggest that graceful degradation on new or unknown object categories might be an incidental benefit. Brust & Denzler [5] propose an extension of conditional classifier chains to the more general case of DAGs.

The above approaches can be seen as a limiting case of hierarchical classification, in which every split in the hierarchy is cast as a separate classification problem. Many hierarchical classifiers fall between this extreme and that of flat classification, working in terms of a coarser-grained conditionality in which a “generalist” makes assignments to groupings of the target classes before then distinguishing the group members from one another using “experts”. Xiao *et al*. [40], the quasi-ensemble section of Hinton *et al*. [14, Sec. 5], Yan *et al*. [41], and Ahmed *et al*. [1] all represent modern variations on this theme (which first appears no later than [15]). Additionally, the listed methods all use some form of low-level feature sharing either via architectural constraint or parameter cloning, and all infer the visual hierarchy dynamically through confusion clustering or latent parameter inference. Alsallakh *et al*. [4] make the one proposal of which we are aware which combines hierarchical architectural modifications (at *train* time) with a hierarchical loss, as described in Sec. 2.2. At test time, however, the architecture is that of an unmodified AlexNet, and all superclass “assignment” is purely implicit.

3. Method

We now outline two simple methods that allow to leverage class hierarchies in order to make better mistakes on image classification. We concentrate on the case where the output of the network is a categorical distribution over classes for each input image and denote the corresponding distribution as $p(C) = \phi_C(x;\theta)$, where subscripts denote vector indices and x and θ are omitted. In Sec. 3.1, we describe the *hierarchical cross-entropy* (HXE), a straightforward example of the hierarchical losses reviewed in Sec. 2.2. This approach expands each class probability into the chain of conditional probabilities defined by its lineage in a given hierarchy tree. It then reweights the corresponding terms in the loss so as to penalise classification mistakes in a way that is informed by the hierarchy. In Sec. 3.2, we suggest an easy choice of embedding function to implement the label-embedding framework of Sec. 2.1. The resulting *soft labels* are PMFs over $\mathcal{C}$ whose values decay exponentially w.r.t. an LCA-based distance to the ground truth.

3.1. Hierarchical cross-entropy

When the hierarchy $\mathcal{H}$ is a tree, it corresponds to a unique factorisation of the categorical distribution $p(C)$ over classes in terms of the conditional probabilities along the path connecting each class to the root of the tree. Denoting the path from a leaf node C to the root R as $C^{(0)} = C, \ldots, C^{(h)} = R$, the probability of class C can be factorised as

$$p(C) = \prod_{l=0}^{h-1} p(C^{(l)}|C^{(l+1)}), \tag{2}$$

where $h \equiv h(C)$ is the height of node C. Note that we have omitted the last term $p(C^{(h)}) = 1$. The conditionals can conversly be written in terms of the class probabilities as

$$p(C^{(l)}|C^{(l+1)}) = \frac{\sum_{A \in \text{Leaves}(C^{(l)})} p(A)}{\sum_{B \in \text{Leaves}(C^{(l+1)})} p(B)}, \tag{3}$$

where $\text{Leaves}(C)$ denotes the set of leaf nodes of the subtree starting at node C.

A direct way to incorporate hierarchical information in the loss is to hierarchically factorise the output of the classifier according to Eqn. 2 and define the total loss as the reweighted sum of the cross-entropies of the conditional probabilities. This leads us to define the *hierarchical cross-entropy* (HXE) as

$$\mathcal{L}_{\text{HXE}}(p, C) = -\sum_{l=0}^{h-1} \lambda(C^{(l)}) \log p(C^{(l)}|C^{(l+1)}), \tag{4}$$

where $\lambda(C^{(l)})$ is the weight associated with the edge node $C^{(l+1)} \to C^{(l)}$, see Fig. 2a. Although this loss is expressed in terms of conditional probabilities, it can easily be applied to models that output class probabilities using Eqn. 3. Note that $\mathcal{L}_{\text{HXE}}$ reduces to the standard cross-entropy when all weights are equal to 1. This limit case, which was briefly mentioned by Redmon & Farhadi in their YOLO-v2 paper [27], results only in architectural changes but does not incorporate hierarchical information in the loss directly.

Eqn. 4 has an interesting information-theoretical interpretation: since each term $\log p(C^{(l)}|C^{(l+1)})$ corresponds to the the information associated with the edge $C^{(l+1)} \to C^{(l)}$ in the hierarchy, the HXE corresponds to discounting the information associated with each of these edges differently. Note that since the HXE is expressed in terms of conditional probabilities, the reweighting in Eqn. 4 is not equivalent to reweighting the cross-entropy for each possible ground truth class independently (as done, for instance, in [17, 7]). A sensible choice for the weights is to take

$$\lambda(C) = \exp(-\alpha h(C)), \tag{5}$$

where $h(C)$ is the height of node C and $\alpha > 0$ is a hyperparameter that controls the extent to which information is discounted down the hierarchy. The higher the value of α, the higher the preference for "generic" as opposed to "fine-grained" information, because classification errors related to nodes further away from the root receive a lower loss. While such a definition has the advantages of interpretability and simplicity, one could think of other meaningful weightings (*e.g.* based on the branching factor of the hierarchy tree). We concentrate on Eqn. 5 here, while leaving the exploration of different strategies for future work.

3.2. Soft labels

Our second approach to incorporating hierarchical information, *soft labels*, is a label-embedding approach as described in Sec. 2.1. These methods use a mapping function $y(C)$ to associate classes with representations which encode class-relationship information that is absent in the trivial case of the one-hot representation. In the interest of simplicity, we choose a mapping function $y^{\text{soft}}(C)$ which outputs a categorical distribution over the classes. This enables us to simply use the standard cross-entropy loss:

$$\mathcal{L}_{\text{Soft}}(p, C) = -\sum_{A \in \mathcal{C}} y_A^{\text{soft}}(C) \log p(A), \tag{6}$$

where the soft label embedding is given componentwise by

$$y_A^{\text{soft}}(C) = \frac{\exp(-\beta d(A, C))}{\sum_{B \in \mathcal{C}} \exp(-\beta d(B, C))}, \tag{7}$$

for class distance function d and parameter β. This loss is illustrated in Fig. 2b. For the distance function $d(C_i, C_j)$, we use the height of $\text{LCA}(C_i, C_j)$ divided by the height of the tree. To understand the role of the hyperparameter β,

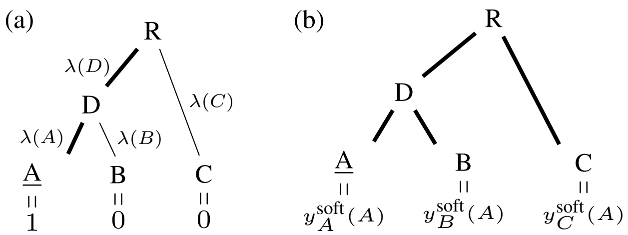

Figure 2: Representations of the *HXE* (Sec. 3.1) and *soft labels* (Sec. 3.2) losses for a simple illustrative hierarchy are drawn in subfigures (a) and (b) respectively. The ground-truth class is underlined, and the edges contributing to the total value of the loss are drawn in bold.

note that values of β that are much bigger than the typical inverse distance in the tree result in a label distribution that is nearly one-hot, *i.e.* $y_A(C) \simeq \delta_{AC}$, in which case the cross-entropy reduces to the familiar single-term log-loss expression. Conversely, for very small values of β the label distribution is near-uniform. Between these extremes, greater probability mass is assigned to classes more closely related to the ground truth, with the magnitude of the difference controlled by β.

We offer two complementary interpretations that motivate this representation (besides its ease). For one, the distribution describing each target class can be considered to be a model of the actual uncertainty that a labeller would experience due to visual confusion between closely related classes[2]. It could also be thought of as encoding the extent to which a common response to different classes is *required* of the classifier, *i.e.* the imposition of correlations between outputs, where higher correlations are expected for more closely related classes. This in turn suggests a connection to the superficially different but conceptually related *distillation* method of Hinton *et al.* [14, Sec. 2], in which correlations between a large network's responses to different classes are mimicked by a smaller network to desirable effect. Here, we simply supply these correlations directly, using widely available hierarchies.

Another important connection is the one to *label smoothing* [33], in which one-hot labels are combined with the uniform distribution. This technique has been used to regularise large neural networks (*e.g.* [33, 6, 36, 45]), but has only recently [22] been studied more thoroughly.

4. Evaluation

In the following, we first describe the datasets (Sec. 4.1) and metrics (Sec. 4.2) of the setup common to all of our experiments. Then, in Sec. 4.3, we evaluate our two simple proposals and compare them to the prior art. Finally, we experiment with random hierarchies to understand when information on class relatedness can help classification.

4.1. Datasets

In our experiments, we use tieredImageNet [28] (a large subset of ImageNet) and iNaturalist'19 [35], two datasets with hierarchies that are *a)* significantly different from one another and *b)* complex enough to cover a large number of visual concepts. ImageNet aims to populate the WordNet [20] hierarchy of nouns, with WordNet itself generated by inspecting IS-A lexical relationships. By contrast, iNaturalist'19 [35] has a biological taxonomy [30] at its core.

tieredImageNet was originally introduced by Ren *et al.* [28] for the problem of few-shot classification, in which the sets of classes between dataset splits are disjoint. The authors' motivation in creating the dataset was to use the WordNet hierarchy to generate splits containing significantly different classes, facilitating better assessment of few-shot classifiers by enforcing problem difficulty.

Although our task and motivations are different, we chose this dataset because of the large portion of the WordNet hierarchy spanned by its classes. To make it suitable for the problem of (standard) image classification, we re-sampled the dataset so as to represent all classes across the train, validation, and test splits. Moreover, since the method proposed in Section 3.1 and YOLO-v2 [27] require that the graph representing the hierarchy is a tree, we modified the graph of the spanned WordNet hierarchy slightly to comply with this assumption (more details available in the supplementary material, Sec. D). After this procedure, we obtained a tree of height 13 covering 608 classes. We refer to this dataset as *tieredImageNet-H*.

iNaturalist is a dataset of images of organisms that has mainly been used to evaluate fine-grained visual categorisation methods. The dataset construction protocol differs significantly from the one used for ImageNet in that it relies on passionate volunteers instead of workers paid per task [35]. Importantly, for the 2019 edition of the CVPR Fine-Grained Visual Categorization Workshop, metadata with hierarchical relationships between species have been released. In contrast to WordNet, this taxonomy is an 8-level complete tree spanning 1010 classes that can readily be used in our experiments without modifications. Since the labels for the test set are not public, we randomly re-sampled three splits from the original train and validation splits into a new training, validation and test set (with respective probabilities of 0.7, 0.15, and 0.15) We refer to this modified version of iNaturalist'19 as *iNaturalist-H*.

4.2. Metrics

We consider three measures of performance, covering different interpretations of a classifier's *mistakes*.

[2] In a recent work, Peterson *et al.* [25] make use of soft labels expressing the distribution of human labellers for a subset of CIFAR-10, showing strong generalisation for classifiers trained on them.

Top-k error. Under this measure, an example is defined as correctly classified if the ground truth is among the top k classes with the highest likelihood. This is the measure normally used to compare classifiers, usually with k=1 or k=5. Note that this measure considers all mistakes of the classifier equally, irrespective of how "similar" the predicted class is to the ground truth.

Hierarchical measures. We also consider measures that, in contrast to the top-k error, do weight the severity of mistakes. We use the height of the lowest common ancestor (LCA) between the predicted class and the ground truth as a core severity measure, as originally proposed in the papers describing the creation of ImageNet [10, 9]. As remarked in [9], this measure should be thought of in logarithmic terms, as the number of confounded classes is exponential in the height of the ancestor. We also experimented with the Jiang-Conrath distance as suggested by Deselaers & Ferrari [11], but did not observe meaningful differences wrt. the height of the LCA.

We consider two measures that utilise the height of the LCA between nodes in the hierarchy.

- The **hierarchical distance of a mistake** is the height of the LCA between the ground truth and the predicted class *when the input is misclassified*, *i.e*. when the class with the maximum likelihood is incorrect. Hence, it measures the severity of misclassification when only a single class can be considered as a prediction.
- The **average hierarchical distance of top-k**, instead, takes the mean LCA height between the ground truth and each of the k most likely classes. This measure can be important when multiple hypotheses of a classifier can be considered for a certain downstream task.

4.3. Experimental results

In the following, we analyse the performance of the two approaches described in Sec. 3.1 and Sec. 3.2, which we denote by *HXE* and *soft labels*, respectively. Besides a vanilla cross-entropy-based flat classifier, we also implemented and compared against the methods proposed by Redmon & Farhadi [27] (YOLO-v2)[3], Frome *et al*. [12] (DeViSE), and Barz & Denzler [3]. As mentioned in Sec. 1, these methods represent, to the best of our knowledge, the only modern attempts to deliberately reduce the semantic severity of a classifier's mistakes that are generally applicable to any modern architecture. Note, though, that we do not run DeViSE on *iNaturalist-H*, as the class IDs of this dataset are alien to the corpus used by word2vec [18].

Finally, we do not compare against the "generalist/expert" architectures surveyed in Sec. 2.3 for reasons explained in the supplementary material, Sec. B.

[3]Note that this refers to the conditional classifier subsystem proposed in Sec. 4 of that work, not the main object detection system.

Since we are interested in understanding the mechanisms by which the above metrics can be improved, it is essential to use a simple configuration that is common between all of the algorithms taken into account. We use a ResNet-18 architecture (with weights pretrained on ImageNet) trained with Adam [26] for 200,000 steps and mini-batches of size 256. We use a learning rate of $1\mathrm{e}{-5}$ unless specified otherwise. Further implementation details are deferred to the supplementary material, Sec. C.

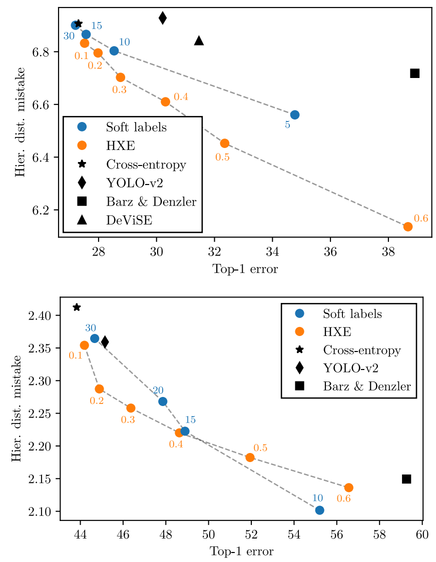

Figure 3: Top-1 error vs. hierarchical distance of mistakes, for *tieredImageNet-H* (top) and *iNaturalist-H* (bottom). Points closer to the bottom-left corner of the plot are the ones achieving the best tradeoff.

Main results. In Fig. 3 and 4 we show how it is possible to effectively trade off top-1 error to reduce hierarchical error, by simply adjusting the hyperparameters α and β in Eqn. 5 and 7. Specifically, increasing α corresponds to (exponentially) discounting information down the hierarchy, thus more severely penalising mistakes where the predicted class is further away from the ground truth. Similarly, decreasing β in the soft-label method amounts to progressively shifting the label mass away from the ground truth and towards the neighbouring classes. Both methods reduce to the cross-entropy in the respective limits $\alpha \to 0$ and $\beta \to \infty$. Moreover, notice that varying β affects the entropy of the distribution representing a soft label, where the two limit cases are β=∞ for the standard one-hot case and β=0 for the uniform distribution. We experiment with $0.1 \leq \alpha \leq 0.6$ and $5 \leq \beta \leq 30$.

To limit noise in the evaluation procedure, for both of our methods and all of the competitors, we fit a 4th-degree polynomial to the validation loss (after having discarded the first 50,000 training steps) and pick the epoch corresponding to

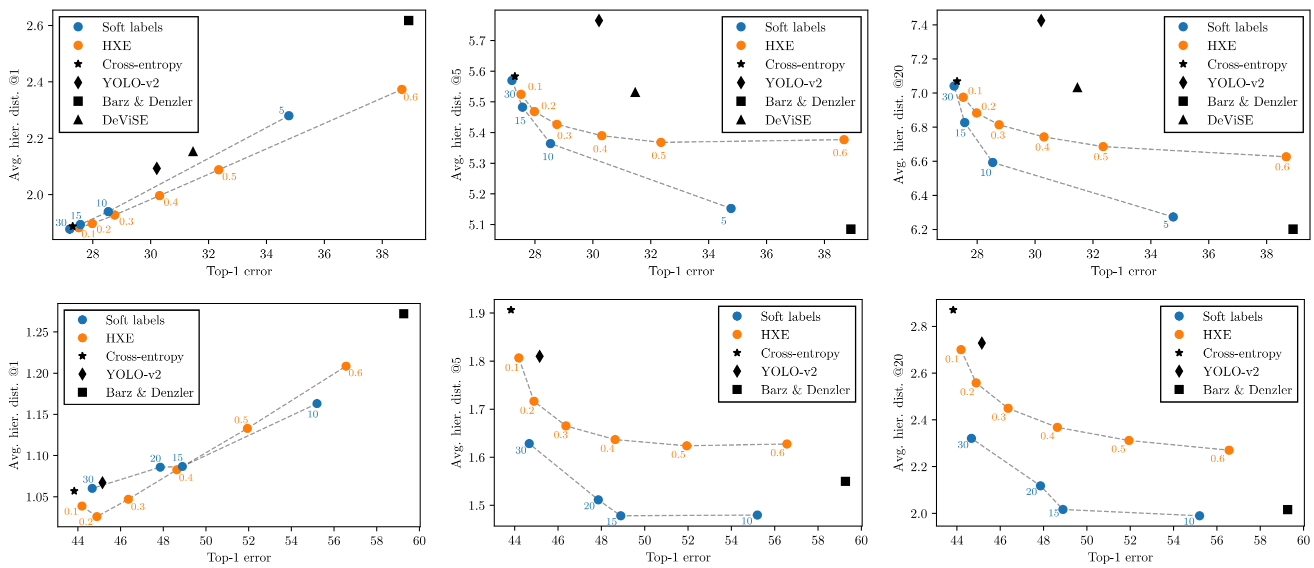

Figure 4: Top-1 error vs. average hierarchical distance of top-k (with $k \in \{1, 5, 20\}$) for *tieredImageNet-H* (top three) and *iNaturalist-H* (bottom three). Points closer to the bottom-left corner of the plot are the ones achieving the best tradeoff.

its minimum along with its four neighbours. Then, to produce the points reported in our plots, we average the results obtained from these five epochs on the validation set, while reserving the test set for the experiments of Table 1. Notice how, in Fig. 4, when considering the hierarchical distance with k=1, methods are almost perfectly aligned along the plot diagonal, which demonstrates the strong linear correlation between this metric and the top-1 error. This result is consistent with what is observed in [31], which in 2011 led the organisers of the ILSVRC workshop to discard rankings based on hierarchical distance.

When considering the other metrics described in Sec. 4.2, a different picture emerges. In fact, a tradeoff between top-1 error and hierarchical distance is evident in Fig. 3 and in the plots of Fig. 4 with k=5 and k=20. Notice how the points on the plots belonging to our methods outline a set of tradeoffs that subsumes the prior art. For example, in Fig. 3, given any desired tradeoff betweeen top-1 error and hierarchical distance of mistakes on *tieredImageNet-H*, it is better to use HXE than any other method. A similar phenomenon is observable when considering the average hierarchical distance of top-5 and top-20 (Fig. 4), although in these cases it is better to use the *soft labels*. The only exception to this trend is represented by Barz & Denzler [3] on *tieredImageNet-H*, which can achieve slightly lower average hierarchical distance for k=5 or k=20 at a significant cost in terms of top-1 error.

Using the results illustrated in Fig. 3 and 4, we pick two reasonable operating points for both of our proposals: one for the high-distance/low-top1-error regime, and one for the low-distance/high-top1-error regime. We then run both of these configurations on the test sets and report our results in Table 1. Means and 95% confidence intervals are obtained from the five best epochs.

The trends observed on the validation set largely repeat themselves on the test set. When one desires to prioritise top-1 error, then soft labels with high β or HXE with low α are more appropriate, as they outperform the cross-entropy on the hierarchical-distance-based metrics while being practically equivalent in terms of top-1 error. In cases where the hierarchical measures should be prioritised instead, it is preferable to use soft labels with low β or HXE with high α, depending on the particular choice of hierarchical metric. Although the method of Barz & Denzler is competitive in this regime, it also exhibits the worst deterioration in top-1 error with respect to the cross-entropy.

Our experiments generally indicate, over all tested methods, an inherent tension between performance in the top-1 sense and in the hierarchical sense. We speculate that there may be a connection between this tension and observations proceeding from the study of adversarial examples indicating a tradeoff between robustness and (conventional) accuracy, as in *e.g.* [34, 43].

Can hierarchies be arbitrary? Although the lexical WordNet hierarchy and the biological taxonomy of iNaturalist are not visual hierarchies per se, they reflect visual relationships between the objects represented in the underlying datasets. Since deep networks leverage visual features, it is interesting to investigate the extent to which the structure of a particular hierarchy is important. The connection between visual and semantic proximity has also been explored in such works as [9, 11]. But what happens if we impose an arbi-

Table 1: Results on the test sets of *tieredImageNet-H* (top) and *iNaturalist-H* (bottom), with 95% confidence intervals. For each column of each dataset, the best entry is hightlighted in yellow, while the worst is highlighted in gray.

	Hier. dist. mistake ↓	Avg. hier. dist. @1 ↓	Avg. hier. dist. @5 ↓	Avg. hier. dist. @20 ↓	Top-1 error ↓
CROSS-ENTROPY	6.89 ± 0.004	1.90 ± 0.002	5.59 ± 0.004	7.07 ± 0.007	27.55 ± 0.038
BARZ&DENZLER [3]	6.72 ± 0.017	2.62 ± 0.014	5.09 ± 0.009	6.21 ± 0.007	39.03 ± 0.157
YOLO-V2 [27]	6.91 ± 0.006	2.10 ± 0.002	5.77 ± 0.012	7.42 ± 0.018	30.43 ± 0.030
DEVISE [12]	6.83 ± 0.005	2.17 ± 0.003	5.54 ± 0.003	7.04 ± 0.002	31.69 ± 0.058
HXE α=0.1 (ours)	6.83 ± 0.009	1.89 ± 0.003	5.53 ± 0.004	6.98 ± 0.008	27.68 ± 0.066
HXE α=0.5 (ours)	6.46 ± 0.026	2.11 ± 0.021	5.37 ± 0.003	6.69 ± 0.008	32.61 ± 0.443
SOFT-LABELS β=15 (ours)	6.83 ± 0.005	1.90 ± 0.004	5.49 ± 0.002	6.83 ± 0.002	27.78 ± 0.063
SOFT-LABELS β=5 (ours)	6.56 ± 0.009	2.29 ± 0.008	5.16 ± 0.006	6.28 ± 0.005	35.00 ± 0.096
CROSS-ENTROPY	2.41 ± 0.003	1.05 ± 0.004	1.90 ± 0.004	2.87 ± 0.006	43.77 ± 0.138
BARZ&DENZLER [3]	2.19 ± 0.008	1.27 ± 0.007	1.56 ± 0.006	2.03 ± 0.005	57.83 ± 0.137
YOLO-V2 [27]	2.37 ± 0.006	1.07 ± 0.007	1.81 ± 0.008	2.73 ± 0.009	45.23 ± 0.202
HXE α=0.1 (ours)	2.35 ± 0.007	1.04 ± 0.004	1.80 ± 0.004	2.70 ± 0.009	44.28 ± 0.171
HXE α=0.6 (ours)	2.13 ± 0.003	1.21 ± 0.004	1.62 ± 0.003	2.68 ± 0.003	56.61 ± 0.241
SOFT-LABELS β=30 (ours)	2.35 ± 0.002	1.05 ± 0.005	1.62 ± 0.005	2.32 ± 0.004	44.75 ± 0.139
SOFT-LABELS β=10 (ours)	2.10 ± 0.005	1.16 ± 0.006	1.47 ± 0.004	1.99 ± 0.003	55.16 ± 0.196

trary hierarchy that potentially subverts this relationship?

To answer this question, we randomised the nodes of the hierarchies and repeated our experiments. Results on *iNaturalist-H* are displayed in Fig. 5 (*tieredImageNet-H* exhibits a similar trend). Again, we report tradeoff plots showing top-1 errors on the x-axis and metrics based on the height of the LCA (on the randomised hierarchy) on the y-axis. It is evident that the hierarchical distance metrics are significantly worse when using the random hierarchy. Although this is not surprising, the extent to which the results deteriorate is remarkable. This suggests that the inherent nature of the structural relationship expressed by a hierarchy is paramount for learning classifiers that, besides achieving competitive top-1 accuracy, are also able to make better mistakes. Thus, while one may wish to enforce application-specific relationships using this approach (as motivated in Sec. 1), the effectiveness of doing so may be constrained by underlying properties of the data.

Curiously, for the *soft labels*, the top-1 error of the random hierarchy is consistently *lower* than its "real" hierarchy counterpart. We speculate this might be due to the structural constraints imposed by a hierarchy anchored to the visual world, which can limit a neural network from opportunistically learning correlations that allow it to achieve low top-1 error (at the expense of ever more brittle generalisation). Indeed, the authors of [42] noted that it is more difficult to train a deep network to map real images to random labels than it is to do so with random images. The most likely explanation for this is that common visual features, which are inescapably shared by closely related examples, dictate common responses.

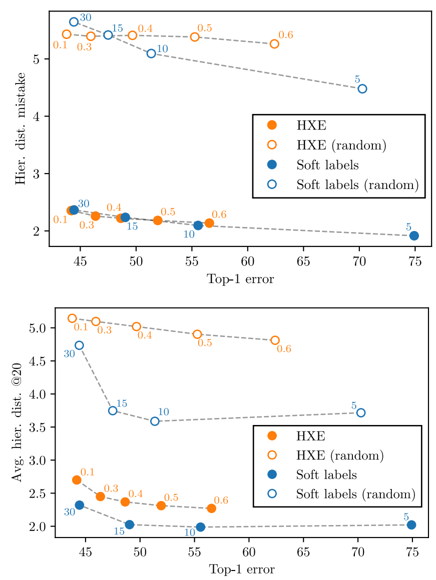

Figure 5: Top-1 error vs. hierarchical distance of mistakes (top) and hierarchical distance of top-20 (bottom) for *iNaturalist-H*. Points closer to the bottom-left corner of the plots are the ones achieving the best tradeoff.

5. Conclusion

Since the advent of deep learning, the community's interest in making better classification mistakes seems to have nearly vanished. In this paper, we have shown that this problem is still very much open and ripe for a comeback. We have demonstrated that two simple baselines that modify the cross-entropy loss are able to outperform the few modern methods tackling this problem. Improvements in this task are undoubtedly possible, but it is important to note the delicate balance between standard top-1 accuracy and mistake severity. As it stands, it appears that it is possible to make better mistakes, but the nature of the class relationships defining the concept of "better" is crucial. Our hope is that the results presented in this paper are soon to be surpassed by the new competitors that it has inspired.

References

[1] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani. Network of experts for large-scale image categorization. In *European Conference on Computer Vision*, 2016.

[2] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[3] Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[4] 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*, 2017.

[5] Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: using hierarchies to improve deep classifiers. *arXiv preprint arXiv:1811.07125*, 2018.

[6] Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. In *Proc. Interspeech*, 2017.

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] Jia Deng, Alexander C Berg, and Li Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *2011*, 2011.

[9] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European Conference on Computer Vision*, 2010.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[11] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2011.

[12] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, 2013.

[13] J Goodman. Classes for fast maximum entropy training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al. Adaptive mixtures of local experts. *Neural computation*, 1991.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

[20] George A Miller. *WordNet: An electronic lexical database*. 1998.

[21] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Aistats*. Citeseer, 2005.

[22] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, 2014.

[25] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *IEEE International Conference on Computer Vision*, 2019.

[26] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2019.

[27] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[28] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[29] Bernardino Romera-Paredes and Philip Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, 2015.

[30] Michael A Ruggiero, Dennis P Gordon, Thomas M Orrell, Nicolas Bailly, Thierry Bourgoin, Richard C Brusca, Thomas Cavalier-Smith, Michael D Guiry, and Paul M Kirk. A higher level classification of all living organisms. *PloS one*, 2015.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[32] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 2011.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[34] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2018.

[35] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[37] Nakul Verma, Dhruv Mahajan, Sundararajan Sellamanickam, and Vinod Nair. Learning hierarchical similarity metrics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[38] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[39] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[40] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[41] 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 *IEEE International Conference on Computer Vision*, 2015.

[42] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2016.

[43] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

[44] Bin Zhao, Fei Li, and Eric P Xing. Large-scale category structure aware image categorization. In *Advances in Neural Information Processing Systems*, 2011.

[45] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.