Image Classification using Hierarchical Layers

Teja Kalvakolanu Marcus Karr Nagasai Chandra Roxanne Miller Justin Morgan

***Abstract*—Image classification performance is critical in Deep learning applications. Different type of structured data require different types of dedicated neural networks to achieve better results. But the existing CNN for image classification is a general flat N layer network that is general and does not use structure of the data set to achieve better results. This research is aimed to design a convolutional neural network that uses the hierarchical information of parent class to improve the classification of child classes. We modified the existing Resnet 18 by adding layers so that it uses hierarchical classification losses. This results are compared with a flat N layer Neural network. we ran this experiment 20 times each has 5 epochs for both the hierarchical network and Flat N layer network. Even though the network did not perform better than the flat model in detecting the most probable class but it performed better than the existing model in detecting the top 3 probable classes.**

I. Introduction

Image classification is one of the most popular applications of AI and machine learning models. With the advent of deep learning, convolutional neural networks (CNN) have shown splendid results in image classification. But, the traditional CNN's are trained as flat N-way classifiers and only produce results as a flat layer, and do not make use of any hierarchical information about the data. This is a research oriented experiment in improving the results of such classifications by using hierarchical and categorical information available during training the model. A neural network extending the existing resnet 18 to facilitate the hierarchical data. The proposed hypothesis is the child classes accuracy can be increased if the information about the parent class is sent through the neural network while predicting the child class. To do this we modified the existing loss function in order to train the network.

II. Background

One major-related work done to improve image classification results is HD-CNN. HD-CNN: Hierarchical Deep Convolutional Neural Network for Large Scale Visual Recognition, is a research paper that embeds deep convolutional neural networks into a categorical hierarchy.

Another related topic is one-shot learning. A 2012 paper by Salakhutdinov, et al., introduced hierarchical Bayesian networks in order to "leverage higher-order knowledge abstracted from previously learned categories". In that case, the goal was to classify with a single training example. By contrast, our goal is to use hierarchical information to increase the sample efficiency as well as the accuracy.

III. Experiment

A. Data set

The methodology we used when building our data set targeted a particular set of child to child class relations. This research sheds light on potential classification inaccuracies due to similar features being shared across two or more classes; a problem our research sets out to resolve. In particular, the data set we used contains a hierarchical set of cat and dog breeds. We chose to structure our classes as a two layer hierarchy: the first level would contain two species, cats and dogs as the parent classes, and the second layer of child classes would contain breeds of their respective parent classes one level above.

The selection process of cat and dog breeds evolved around our main child to child class relations methodology. We targeted breeds of cats and dogs that have similar physical body characteristics to establish these child to child relations. For instance, the Persian cat and Golden-retriever dog breeds have similar style of long and wavy fur. Each of our selected breeds contained child to child relations pertaining to physical body characteristics; such as the one outlined in the previous child to child, Persian cat and Golden-retriever dog example.

Networks with single-layer classifications will struggle more when two or more classes share similar features. In theory, our hierarchical classification will alleviate some of these cross-classification issues by eliminating sets of classes a network can infer; thus our child to child class relation data set selection methodology.

B. Architecture

This research will train a CNN using data set to recognize image hierarchies. The final goal of the system is to provide the system with a parent class and train it using relatively few images to recognize a child class. In order to achieve this the model must take a parent class as input, as well as the image to be classified.

Baseline accuracy and training speed will be established by building a neural network with a single classification layer containing all of the parent and child classes. This baseline architecture will not use a parent class as input. We will consider the accuracy of this network the minimum accuracy to meet with our new architecture.

Once baselines have been established, a neural network with the proposed architecture is built and compared to the first model. The same training and testing images are used with both architectures to ensure that differences in performance are not due to differences in data.

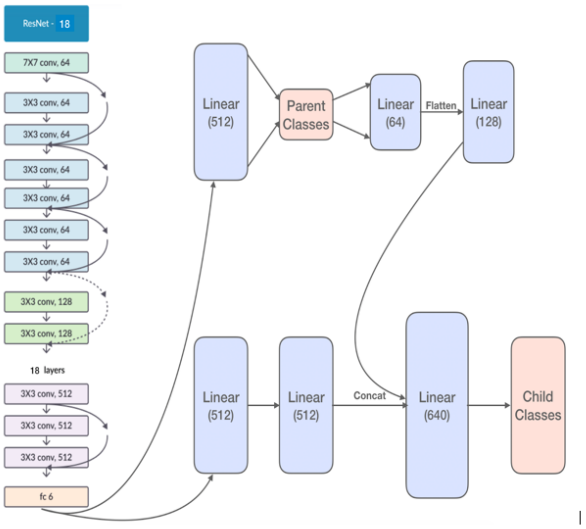


Fig. 1. Results

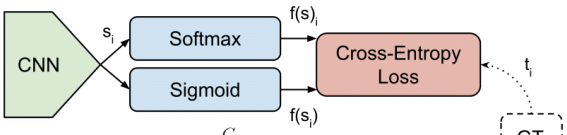
The architecture of this system involves training the model on images with multiple labels. These multiple labels correspond to the hierarchical properties of an item derived from hierarchical data set. This information acts as an additional attribute that should influence successive layers' confidence in producing labels for child classes.

This architecture will be implemented by adding dense layers between each hierarchical level that are given the results from the previous layer. The dense layers output an attribute that helps in the successive layers' classification. The inclusion of this inference parameter derived from parent layers should improve the accuracy.

The existing resnet is used and the output of the fully convolutional layer is divided in to two branches one to detect the parent class and one to detect the child class . A linear layer with 512 neurons is used at the start of both branches. This is used to predict the parent classes and is again sent through the linear layer to convert it in to 64*64 which is flattened to 128*128. This output is concatenated with the child branch which has 512 *512 linear layer and both are concatenated to a linear layers of 640 neurons which is used to predict the child classes.

C. Loss function

The loss function we used is Cross entropy . Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events.Entropy is the number of bits required to transmit a randomly selected event from a probability distribution. A skewed distribution has a low entropy, whereas a distribution where events have equal probability has a larger entropy.Cross-entropy builds upon the idea of entropy from information theory and calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution.In this case one distribution is the final dense layer output of input image and other is final dense layer output of the target is the target image. since we plan to use the heirarchical information to



$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1-t_1) log(1-f(s_1))$$

Fig. 2. Loss entropy

predict the child classes the loss function is modified.

Total loss = (C.E(child-pred, child-labels) + C.E(parent-pred, parent-labels) + 0.1*mse(child-pred.softmaxlog(0), target))

we add the cross entropy loss of the child class prediction , cross entropy loss of parent class prediction and 0.1 times the mean square error of the child prediction and target. The child predication array represents the parent information i.e it is an array of 0 or 1 of size 10 . Based on the parent class this array is modified by calculating the soft max this is compared with target by calculating the mean square error. The whole loss is back propagated.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where N is the number of data points,
f_i the value returned by the model and
y_i the actual value for data point i.

Fig. 3. Mean Squared Error

IV. Validation

To measure the success of the new architecture, it was compared against the results from a normal convolutional neural network architecture. Both networks were run 20 times for 5 epochs each and accuracy results were obtained for single class classification as well as for estimation of the three most likely classes. The accuracy results were as follows The mean and standard deviation for accuracy obtained after running over 20 times each with 5 epochs are as represented in the fig 2. Even though the mean and standard deviation accuracy for the most probable class for Hierarchical model is not better than the base model the mean and standard deviation for the top 3 classes is certainly better than the existing resnet model. This infers that the architecture we developed certainly boosted

	Mean, the standard deviation of accuracies over 20 epochs for most probable class		Mean, the standard deviation of accuracies over 20 epochs for the top 3 probable classes	
	Mean	Std	Mean	Std
Base Model	88.9	0.37	98.27	0.25
Hierarchical Model	88.6	0.55	98.47	0.12

Fig. 4. Results

the accuracy of the right child class by using the hierarchical information.

V. Conclusion

Even though the hierarchical architecture did not outperform the flat layer network in predicting the most probable class, it did do better at predicting the top three most probable classes. This is because when the network predicts the parent class, it boosts the probability of all of that parent's child classes. Child classes of the wrong type will therefore be much less likely to be in the top three. We think the network did worse than the baseline at predicting a single child class because the loss function had to balance too many different variables. When the parent class was provided as one of the initial inputs instead of at the end, the network performed slightly better than baseline.

VI. Future Work

This work can be extended to modify the loss function and perform all the regularization techniques to make this better than the existing resnet for hierarchical data. This network can also be used for larger data sets that has hierarchical information available and child specific classes having similarities across different parent classes to give more generalized and improved results.

References

[1] 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.

[2] Liu, Yunhong, and Yizhu Huang. "MC-HDCNN: Computing the Stereo Matching Cost with a Hybrid Dilated Convolutional Neural Network." International Conference on Neural Information Processing. Springer, Cham, 2019.

[3] Penalty, Temporal Sparsity. "HD-CNN: Hierarchical Deep Convolution

[4] Zhu, Xinqi, and Michael Bain. "B-CNN: branch convolutional neural network for hierarchical classification." arXiv preprint arXiv:1709.09890 (2017).

[5] Roy, Deboleena, Priyadarshini Panda, and Kaushik Roy. "Tree-CNN: a hierarchical deep convolutional neural network for incremental learning." Neural Networks 121 (2020)