Low-Shot-Learning of Diseases in Chest X-Rays via Hallucination

Matan Harel, Uri Avron & Jonathan Somer

September 2, 2018

Abstract

One of the promises of the recent advancements in Artificial Intelligence is the ability to facilitate high precision computer aided diagnosis (CAD) systems and make such high precision diagnosis affordable and highly available. Medical imaging is one area where current deep learning and computer vision methods could possibly be applied. These methods utilize large sets of labelled images in order to acheive high accuracy. High quality labelled data is difficult to obtain due to a variety of factors.

Our objective is to answer whether we can use an existing large set of X-Ray images of healthy patients as well as a large of set of images of patients with some diseases, in order to improve the learning accuracy of new diseases for which we have only a few example images to learn from. This setting is known as Low-Shot-Learning.

We have implemented, extended and re-evaluated a method for performing low-shot-learning proposed by Bharath Hariharan and Ross Girshick from Facebook AI research, 2016 [1]. During this process we have also attempted to answer some questions that remained open while reading their innovative original paper and we will show some novel methods for evaluating this low-shot-learning setting.

1. Introduction

1.1. Low-Shot-Learning

The ability to learn from very few examples is a hallmark of human visual intelligence. Classical Machine Learning approaches fail to generalize from few examples so new techniques are required for performing Low-Shot-Learning.

The setting for low-shot-learning is composed of two phases. The first is a representation learning phase: the learner tunes its feature representation on an available set of base classes that have many training instances. In the low-shot learning phase, the learner is exposed to a set of novel classes with only a few examples per class and must learn a classifier over the joint label space of base and novel classes.

We evaluate the new classfier's accuracy over both the base and novel classes in order to see that higher accuracy was acheived on the novel classes but also that accuracy was not impaired for the base classes.

Low-Shot-Learning is a great challenge particularly in our setting as learning in a medical setting carries many unique challenges: Each disease category contains great intra-class variation: patients have varying anatomies, there are different methods of performing each examination, variation might be induced by different equipment and so on. Thus, very large labelled datasets are needed in order to capture this great variation. Obtaining high quality labelled datasets as such is very difficult. In addition, even if labelled data is obtained, the physician's diagnosis can be incorrect and in many cases is not validated or such validation does not get logged. In the dataset we chose to work with the researches who gathered the data employed a NLP text mining solution to procure labels from physicians written reports, a process which adds further noise to the labelling.

Thus, high quality labelled data is difficult to obtain and the ability to learn from little data is highly valuable.

1.2. Our Approach - High Level

As noted our approach is primarily based on the paper by Bharath Hariharan and Ross Girshick. At a very high level the method is as follows: given only a few images of a novel class, we train a Generator network to generate many new 'hallucinated' examples for this same class. We then use those new generated images, as well as those that were given to us, in order to train a nother Neural Network to perform the classification task.

The method for generating new training examples is based on the insight that variation within one category might be transferable to another category. For instance, a certain variation in anatomy may impact the chest images similarly regardless of the disease.

As we dive into the details of the method we will elaborate on the points in which we implemented novel solutions and advancements.

2. Our Method:

The method is composed of 3 parts:

1. Feature representation learning
2. Training an Example Generator
3. Training a classifier using the generated data

We will first introduce the datasets we used and then describe the method in detail.

2.1. Datasets used

We tested our results on 3 different datasets. The first 2 are the well researched MNIST and CIFAR10 datasets. The third is a new chest X-Ray image dataset.

In May 2017 the "ChestX-ray8" dataset was presented by a team of researchers from the NIH [2]. In the paper they present the methods used to generate the data. A short summary of the data is as follows:

- 108,948 images of 32,717 patients.
- 8 disease labels text-mined from radiological reports.
- Each image is labeled with one or more of these, or 'Normal'.
- Labeling: classes are very imbalanced. For example: 84K images were tagged 'Normal' and around 1K were tagged with 'Cardiomegaly '.
- Image sizes are 1024 by 1024 pixels. (big images, around 40GB of space for them all)

Along with the data, they also provide a benchmark for the task of classifying diseases using a DCNN they have trained.

2.2. Feature Representation Learning

For the xray data set we used a ResNet50 DCNN pretrained on the very large and diverse imagenet dataset, without the last dense layer, in order to generate the features for the images. This method was used in the chest x-ray paper in order to train a classifier on the data and implementing the same method enabled us to first reach their results and continue from there onto low shot learning.

For the mnist and cifar datasets we trained two different CNNs which acheived ~99% and ~90% accuracy on the datasets respectively. In the low-shot-learning setting we do not have access to data from the novel class during representation learning. Thus, we treated each class in turn as the novel class , and trained a classifier on the remaining classes. We then used this classifier, with the last layer removed, as a feature extractor in order to generate features for the novel class as well.

There is a significant difference between the two methods. In our method (the second), representation learning is performed ad hoc for the specific setting - “we learn to represent digits by learning to recognize digits”, whereas in the first setting a generic network is used to generate features for images from a very specific domain. As further research we propose to train a DCNN from scratch on the X-Ray dataset and compare the two methods. We presume that features learnt from data that is close to the domain at hand will be better than those obtained by generic models.

2.3. Learning to Generate new Examples

We now train a generator G for hallucinating images for novel classes. As we have noted, the method for generating new training examples is based on the insight that variation within one category might be transferable to another category. We want to use the knowledge of the intra-class variation of one class in order to generate a diverse set of examples for the novel class.

We train G to “solve analogies”: G will receive as input the concatenated feature vectors $\langle \phi(b_1), \phi(b_2), \phi(x) \rangle$ where b_1, b_2 are two images from the same base class and x is a novel image. For this input, G will output a vector who solves the analogy $b_1 : b_2 \Rightarrow x$:? Thus applying to x the $b_1 \rightarrow b_2$ transformation. Note the the $b_1 \rightarrow b_2$ transformation stays within class B and the generator should perform on x a transformation that does not result in an element of another class.

The method for training G is as follows: G will be a 3 layer multi-layer-perceptron. The training data for G is generated by creating completed analogies - quadruplets of feature vectors from the base classes. We start by clustering each of the base classes into k clusters. Then for each two classes A, B, for each pair of centroids, c_1^A, c_2^A from class A we find the pair c_1^B, c_2^B such that the cosine distance between $c_1^A - c_2^A$ and $c_1^B - c_2^B$ is minimized. Concatenating the 4 centroids results in one element in the dataset.

For each quadruplet $\langle c_1^A, c_2^A, c_1^B, c_2^B\rangle$ we feed the triplet: $\langle c_1^A, c_2^A, c_1^B\rangle$ into G. We want $G(\langle c_1^A, c_2^A, c_1^B\rangle)$ to be as close as possible to c_2^B and also remain within the class B. In order to do so we minimize the loss function:

$$\lambda MSE(G(\langle c_1^A, c_2^A, c_1^B\rangle), c_2^B) + L_{cls_{BASE}}(G(\langle c_1^A, c_2^A, c_1^B\rangle), B)$$

Where we have MSE the mean square error between the generator's output and the true target. And $L_{cls_{BASE}}$ the log-loss of the classifier w.r.t the true class of c_2^B.

One main concern we had while reading the paper was the following: we train the generator on one set of classes, but then use it to generate data for new classes. **Does this method really generalize?** Does a generator trained to perform transformations that stay within one set of classes respect this constraint on a novel class it has never seen before?

The authors of the paper in which this concept is presented do not address this issue directly and we wished to test this explicitly. In order to do so we trained another - "all knowing" classifier. This classifier was trained on the feature vectors of all classes including many examples of the novel class. We assume that if the generator generalizes well, then the transformations it performs on an example from a novel class it has never seen before the output will also remain within the same class.

The results show that this is indeed the case:

++BENCHMARK PLOT++

3. Experiments

1. describe the pipeline experiment. pit falls etc. the most we can squeeze out of the generator
2. figures and results.

References

[Low-shot Visual Recognition by Shrinking and Hallucinating Features]

[ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks]