NEURAL DATASET

GENERALITY

RAGAV VENKATESAN
VIJETHA GATUPALLI
BAOXIN LI
SIFT  HOG  GABOR

ALL ABOUT THE FEATURES

DAISY
FEATURES COMES FROM DATA

• PCA
• Dictionaries
• Neural Networks

We know/posit that these features are the best representations of data for the dataset that we are currently concerned about.

What about off-the-shelf neural features?
OFF-THE-SHELF FEATURES

Extract features from a large dataset such as ImageNet
Make the feed forward network public.

Download off-the-shelf networks.
Extract features on user dataset.
Train a new classifier on top.
OFF-THE-SHELF FEATURES

Extract features from a large dataset such as ImageNet
Make the feed forward network public.

Download off-the-shelf networks.

Extract features on user dataset.
Train a new classifier on top.

Is there a guarantee that the downloaded features can capture the intricacies and idiosyncrasies of the user dataset?
OFF-THE-SHELF FEATURES

Most often not!
But we roll with it. – Because it works.

Is there a guarantee that the downloaded features can capture the intricacies and idiosyncrasies of the user dataset?
The ubiquity of downloaded CNNs
Unquestioned performance of networks trained on ImageNet

One network fits all.

But does it?
ATOMIC STRUCTURES

• CNN filters take some shapes due to the entropy of the dataset.
• Some datasets have some unique idiosyncrasies that show up as atomic structures.
• These may be edges and Gabor filters in the first layers and so on.
A GENERALITY RANKING METRIC

• Generality is not a rankable concept.
  • Due to the overlapping nature of feature expressions, representations aren’t usually nestable or complete.
  • Generality is only a relative concept.

• Can we use the neural training procedure and dataset performance to measure dataset generality? - Yes.
  • Very close corollary to network transferability and remembrance. [1].

DETOUR …. OBSTINATE LEARNING.

• An obstinate layer is a layer whose weights are not allowed to update during training.
• Gradients are simply ignored.
• An obstinate layer and all the layers that feeds into the obstinate layers must all be frozen.
  • Downloading a network and training only the softmax layer.
  • Layer-wise pre-training.
  • Dropouts (not exactly but similar).
EXPERIMENT SETUP

1. Consider two datasets $D_1$ and $D_2$.
2. Initialize a network with random weights and train with $D_i$.
   • This network is called the base network and is represented by $n(D_i|r)$.

\[
\begin{align*}
&n_1(D_j|D_i) \\
&n_2(D_j|D_i|D_i) \\
&n_3(D_j|D_i) \\
&n_k(D_j|D_i)
\end{align*}
\]
GENERALITY METRIC

- Performance of $n(D_j | r)$ is $\Psi(D_j | r)$.
- Performance of $n_k(D_j | D_i)$ is $\Psi_k(D_j | D_i)$.
- Dataset generality of $D_i$ with respect to $D_j$ at layer $k$ is:

$$g_k(D_i, D_j) = \frac{\Psi_k(D_j | D_i)}{\Psi(D_j | r)}$$
Performance that is achieved by $D_j$ using,

- $N - k$ layers worth of prejudice from $D_i$
- $k$ layers worth of features from $D_i$
- $k$ layers of novel knowledge from $D_j$

$$g_k(D_i, D_j) = \frac{\Psi_k(D_j | D_i)}{\Psi(D_j | r)}$$
PROPERTIES OF THIS GENERALITY METRIC

- \( g_k(D_i, D_j) > g_k(D_i, D_l) \rightarrow \) at \( k \) layers, \( D_i \) provides more general features to \( D_j \) than to \( D_l \).
  - Conversely, when initialized by \( n(D_i|r) \), \( D_j \) has an advantage in learning than \( D_l \).
- \( g_k(D_i, D_i) \geq 1 \ \forall k \).
- \( g_k(D_i, D_j) \) for \( i \neq j \) might or might not be greater than 1.
DATASETS CONSIDERED
REFERENCES TO DATASETS


SOME INTERESTING RESULTS
Generality curves for each dataset as base against all the other comparing datasets.
PARSING THE GRAPHS
SOME SURPRISING RESULTS

• No dataset is qualitatively the most general.

• MNIST dataset is the most specific.
  • Rather, MNIST dataset is one that is generalized by all datasets very highly at all layers.
  • MNIST dataset actually gives better accuracy when prejudiced with other datasets than with random intits or even when prejudiced with itself !!
  • This is a strong indicator that all datasets contain all atomic structures of MNIST.

• English and Digits are more general than Kannada !!
  • While MNIST and MNIST-rotated are not general, other MNIST with backgrounds, Google SVHN, NIST and Char74-English are all more general than Char74-Kannada.
INTER-CLASS DATASET GENERALITY

• $D_i$ and $D_j$ need not be entire datasets but can also be just disjoint class instances of the same dataset.

• For instance, we divided the MNIST dataset into two parts.
  • MNIST $[4, 5, 8]$ (base) and MNIST $[0, 1, 2, 3, 6, 7, 9]$ (retrain).

• Repeated this experiment several times with decreasing number of training samples per-class in the retrain dataset of MNIST $[0, 1, 2, 3, 6, 7, 9]$.
  • The testing set remained the same size.
  • We created seven such datasets with $7p$, $p \in [1, 3, 5, 10, 20, 30, 50]$ samples each.
INTRA-CLASS GENERALITY - RESULTS

- Initializing a network that was trained on only a small sub-set of well-chosen classes can significantly improve generalization performance on all classes.
  - Even if trained with arbitrarily few samples.
  - Even at the extreme case of one-shot learning.

<table>
<thead>
<tr>
<th>p</th>
<th>base</th>
<th>k = 0</th>
<th>k = 1</th>
<th>k = 2</th>
<th>k = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Random MNIST[458]</td>
<td>73.07</td>
<td>73.91</td>
<td>76.37</td>
<td>55.61</td>
</tr>
<tr>
<td>3</td>
<td>Random MNIST[458]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>73.34</td>
</tr>
<tr>
<td>5</td>
<td>Random MNIST[458]</td>
<td>90.98</td>
<td>92.98</td>
<td>92.6</td>
<td>83.32</td>
</tr>
<tr>
<td>10</td>
<td>Random MNIST[458]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>81.31</td>
</tr>
<tr>
<td>20</td>
<td>Random MNIST[458]</td>
<td>91.55</td>
<td>93.71</td>
<td>93.82</td>
<td>95.08</td>
</tr>
<tr>
<td>30</td>
<td>Random MNIST[458]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>87.77</td>
</tr>
<tr>
<td>50</td>
<td>Random MNIST[458]</td>
<td>95.52</td>
<td>95.52</td>
<td>97.07</td>
<td>96.78</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>88.62</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>90.78</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>97.38</td>
</tr>
</tbody>
</table>
While initially one would have assumed that Kannada itself does not contain all atomic structures of MNIST, we observed the contrary. SVHN, even with one-sample per class, would be a general dataset, we observed the contrary. SVHN, MNIST has a multitude of unique atomic structures. Even MNIST has 100 times less data than the full dataset and it already achieves close to state-of-the-art accuracy even when no layer is allowed to change. This is a remarkably strong indicator that the missing values.

The point of takeaway from this experiment is that if the classes chosen classes used as pre-training could build a network that generalized the entire dataset. We also provided some practical guidelines for a CNN engineer to adopt. After performing those classes and then learn the rest of the dataset even with very small number of samples, provided the generalities didn’t vary among the layers like it did when we initialized with data from outside the mother dataset. We also observed that the generalities didn’t remain close to state-of-the-art accuracy even when no layer is allowed to change.

This is a strong indicator that once initialized with a general enough subset of the classes, we observed that...
MORE RESULTS...

• Once initialized with a general enough subset of classes from within the same dataset, the generalities didn’t vary among the layers.

• The more the data we used, more stable the generalities remained.

• If the classes are general enough, one may initialize the network with only those classes and then learn the rest of the dataset even with very small number of samples.
Neural Dataset Generality

Ragav Venkatesan, Vijetha Gattupalli, Baoxin Li

(Submitted on 14 May 2016)

Often the filters learned by Convolutional Neural Networks (CNNs) from different datasets appear similar. This is prominent in the first few layers. This similarity of filters is being exploited for the purposes of transfer learning and some studies have been made to analyse such transferability of features. This is also being used as an initialization technique for different tasks in the same dataset or for the same task in similar datasets. Off-the-shelf CNN features have capitalized on this idea to promote their networks as best transferable and most general and are used in a cavalier manner in day-to-day computer vision tasks.

It is curious that while the filters learned by these CNNs are related to the atomic structures of the images from which they are learnt, all datasets learn similar looking low-level filters. With the understanding that a dataset contains many such atomic structures learn general filters and are therefore useful to initialize other networks with, we propose a way to analyse and quantify generality among datasets from their accuracies on transferred filters. We applied this metric on several popular character recognition, natural image and a medical image dataset, and arrived at some interesting conclusions. On further experimentation we also discovered that particular classes in a dataset themselves are more general than others.
Code for the paper, Neural Dataset Generality by Ragav Venkatesan, Vijetha Gatupalli and Baoxin Li — Edit

Latest commit 866576 on Feb 29

ragavvenkatesan updated to match samosa updates.

- .gitignore
  Added Gitignore
  11 months ago

- License.md
  Initialize code for GitHub Push.
  8 months ago

- README.md
  Update README.md
  8 months ago

- __init__.py
  updated to match samosa updates.
  7 months ago

- dataset_setup.py
  Initialize code for GitHub Push.
  8 months ago

To run the code first download the Samosa Toolbox (https://github.com/ragavvenkatesan/Convolutional-Neural-...
This work was supported in part by ARO grant W911NF1410371.

Fin.