Learning CNNs from Web Data

Rodrigo Santa Cruz and Stephen Gould
Australian Centre for Robotic Vision, Australian National University, Canberra, Australia
Outline

• WebVision Challenge
  • The large scale WebVision 2.0 dataset
  • Noisy labels
  • Imbalanced class distribution

• Approach
  • Intuition
  • Self-supervised pre-training
  • Base classifier
  • Weighted sampling
  • Dense prediction

• Results
Large Scale Problem

- WebVision 2.0 Dataset:
  - Noyse dataset generated from more than 12000 queries to **Google image search** and **Flickr social media**.
  - It contains **5,000 visual concepts** associated to synsets in ImageNet.
  - It has more than **16 millions training images**, 250 thousands validation images and 250 thousands test images (no public labels).
  - It also provides additional information such as title and description for Google images and title, description and tags for flickr images.
Noisy Labels
Imbalanced Class Distribution
Approach

- Deep Learning Formula: **Pretrain** on large dataset and then **finetune** on a smaller task-specific dataset. Ex: object detection.

- Initialization is **crucial** to non-convex optimization problems such as learning deep models.

- Does a good start point provide **robustness** to noisy labels?

[Exploring the Limits of Weakly Supervised Pretraining. Mahajan et al 2018]
Self-supervised learning

- The main idea is to exploit supervisory signals, intrinsically in the data, to guide the learning process.
- In practice, we define a supervised proxy task, where labels are obtained with almost zero cost, to train the model before finetune for the target task.

Example:

Question 1: [Doersch et al., ICCV 2015]

Question 2: [Zhang et al., ECCV16]

[Doersch et al., ICCV 2015]

[Zhang et al., ECCV16]
Visual Permutation Learning (VPL)

Ordering Criterion: Smiling

Image sequence $X$

Permutated sequence $\tilde{X} = P \ X$

Ordering Criterion: Spatial Position

Image sequence $X$

Permutated sequence $\tilde{X} = P \ X$

How to recover the original sequence?

$X = P^2 \tilde{X}$

We hypothesize that the model trained to solve such task is able to capture high-level semantic concepts, structure and shared patterns in visual data.

Remarks:

- We relax the inference over permutation matrices to inference over doubly-stochastic matrices (DSMs).
- We develop a neural network layer (Sinkhorn Layer) that approximates DSMs from CNN’s outputs.
- Incorporating the DSM structure in our predictors can avoid the optimizer from searching over impossible solutions.
Sinkhorn Layer

- Sinkhorn Normalization*: Any non-negative square matrix can be converted to a DSM by repeatedly rescaling its rows and columns.

- Function:

\[
R_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{i,k}}; \quad C_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{k,j}}
\]

\[
S^n(Q) = \begin{cases} 
Q, & \text{if } n = 0 \\
C \left( R \left( S^{n-1}(Q) \right) \right), & \text{otherwise.}
\end{cases}
\]

- Gradient (Row normalization):

\[
\frac{\partial \Delta}{\partial Q_{p,q}} = \sum_{j=1}^{l} \frac{\partial \Delta}{\partial R_{p,j}} \left[ \frac{[j = q]}{\sum_{k=1}^{l} Q_{p,k}} - \frac{Q_{p,j}}{\left( \sum_{k=1}^{l} Q_{p,k} \right)^2} \right]
\]

* [Sinkhorn and Knopp 1967][Knight 2008][Adams and Zamel 2011][Mena et al. 2018]
VPL Results

- It provides significant boost in performance compared to random initialization.
- This framework also presents good results for learning-to-rank problems such as image ranking.

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification (mAP%)</th>
<th>FRCN Detection (mAP%)</th>
<th>FCN Segmentation (%mIU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet</td>
<td>78.2</td>
<td>56.8</td>
<td>48.0</td>
</tr>
<tr>
<td>Random Gaussian</td>
<td>53.3</td>
<td>43.4</td>
<td>19.8</td>
</tr>
<tr>
<td>Context Prediction</td>
<td>55.3</td>
<td>46.6</td>
<td>-</td>
</tr>
<tr>
<td>Temporal coherence</td>
<td>58.4</td>
<td>44.0</td>
<td>-</td>
</tr>
<tr>
<td>In-painting</td>
<td>56.5</td>
<td>44.5</td>
<td>29.7</td>
</tr>
<tr>
<td>Colorization</td>
<td>65.6</td>
<td>47.9</td>
<td>35.6</td>
</tr>
<tr>
<td>Jigsaw Puzzle</td>
<td>68.6</td>
<td>51.8</td>
<td>36.1</td>
</tr>
<tr>
<td><strong>DeepPermNet</strong></td>
<td><strong>69.4</strong></td>
<td><strong>49.5</strong></td>
<td><strong>37.9</strong></td>
</tr>
</tbody>
</table>
VPL on the WebVision
VPL on the WebVision

Better Initialization
VPL on the WebVision

![Graph showing comparison between Faster Convergence and Better Initialization between DenseNet and DenseNet + Pretrain.]
VPL on the WebVision

- Faster Convergence
- Better Initialization

Similar local optimum
VPL on the WebVision

Similar behavior is observed under different learning rate schedule.
VPL on the WebVision

- We only achieved marginal improvements (< 1%) using visual permutation learning as pretraining procedure for the WebVision task.

- We can see significant improvements at the beginning of the training which is diluted as the training progresses, reaching similar performance after convergence.

- It is still useful when you need to train a model for few epochs.
VPL Regularizer
VPL Regularizer

- Pretrain in the visual permutation learning:
VPL Regularizer

- Pretrain in the visual permutation learning:
VPL Regularizer

- Pretrain in the visual permutation learning:

- Finetune on the target task:
VPL Regularizer

- Pretrain in the visual permutation learning:

- Finetune on the target task:
VPL Regularizer
Base Classifier - DenseNet121

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>value</th>
<th>Hyper-parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate</td>
<td>0.01</td>
<td>Optimizer</td>
<td>SGD</td>
</tr>
<tr>
<td>Lr. schedule</td>
<td>Decay by 0.1 every 6 epochs</td>
<td>Momentum</td>
<td>0.9</td>
</tr>
<tr>
<td>Batch size</td>
<td>320</td>
<td>Weight Decay</td>
<td>1e-4</td>
</tr>
<tr>
<td>Num. epochs</td>
<td>20</td>
<td>Framework</td>
<td>PyTorch</td>
</tr>
</tbody>
</table>
Weighted Random Sampling

“The network often memorizes the category with more instances when trained on an extremely imbalanced dataset.”

Then, we adopted a weighted sampling strategy which the probability of a image \( i \) been sampled is proportional to the inverse of its frequency,

\[
W_i = \frac{N}{N_{ci}}
\]

where \( N \) is the total number of images and \( N_{ci} \) is the number of images belonging to the same class of image \( i \).
Multiple Crops Prediction

1) Multiple Dimensions

2) Multiple Regions

3) Multiple Crops

+ Horizontal Flips + Resize (224px)

= 4 x 3 x (5 + 1) x 2 = 144 image crops

[Christian Szegedy et al. “Going deeper with convolutions”. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015 ]
## Results

### Results on the Validation Set

<table>
<thead>
<tr>
<th>Model Variation</th>
<th>Top-5 Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DenseNet 121 + Center Crop</td>
<td>0.733</td>
</tr>
<tr>
<td>DenseNet 121 + 10 Crops</td>
<td>0.748</td>
</tr>
<tr>
<td>DenseNet 121 + 144 Crops</td>
<td>0.750</td>
</tr>
<tr>
<td><strong>DenseNet 121 + Pretrain + 144 Crops</strong></td>
<td><strong>0.753</strong></td>
</tr>
</tbody>
</table>
Remarks

- We got the **third-place** in the competition scoring **69.56%** in top-5 accuracy on the **test set**.

- We are the only team in the top three **not using ensemble** of networks.

- We investigated **Self-supervised pre-training** as a tool to provide robust initialization for deep learning models.

- As recent papers suggest*, deep learning models seems to be reasonably **robust to some types of label noise**.

---


Learning CNNs from Web Data

Rodrigo Santa Cruz and Stephen Gould
Australian Centre for Robotic Vision, Australian National University, Canberra, Australia