Self-ensembling for object detection

Geoff French – g.french@uea.ac.uk

Colour Lab (Finlayson Lab)

University of East Anglia, Norwich, UK

Image montages from http://www.image-net.org
Thanks to

My supervisory team: Prof. G. Finlayson, Dr. M. Mackiewicz

Competition organisers and all participants
Overview
IN A NUTSHELL

Adapted self-ensembling – originally designed for classification – for object detection scenarios
We will set the scene by describing self-ensembling for classification and Faster R-CNN for object detection.
After which we will describe our object detection approach
Self-ensembling for classification
Self-ensembling is one of a class of algorithms that use \textit{consistency regularization} \cite{Oliver18}.
Self-ensembling developed for semi-supervised learning in [Laine17]

Further developed in [Tarvainen17] (mean teacher model)
We adapted it for use in domain adaptation [French18] and achieved 1st place in VisDa 2017 classification competition 😊
Mean-teacher model

Student and teacher networks

\( x_{Ti} \)

Self-Ensembling for Object detection
Mean-teacher model

**Student** is standard classifier DNN
Mean-teacher model

Weights of teacher network are exponential moving average of student network

\[
\begin{align*}
\text{stochastic aug.} & \quad \text{Student network} & \text{cross-entropy} & \quad \text{Squared diff} & \quad \text{Weighted sum} & \quad \text{loss} \\
\chi_{si} & \quad x_{si} & \quad x_{Ti} & \quad z_{Ti} & \quad \tilde{z}_{Ti} & \quad \chi_{Ti} \\
\text{Teacher network} &
\end{align*}
\]
Source domain sample:

Predict class probabilities with \textit{student} network and compute supervised cross-entropy loss (with data augmentation)
Target domain sample:

one sample

Self-Ensembling for Object detection
Target domain sample:
augment twice, differently each time
(translation, flip)
Target domain sample:

One path through **student** network
Second through **teacher**
(different dropout)

Self-Ensembling for Object detection
Target domain sample:

Result: two predicted probability vectors

Self-Ensembling for Object detection
Target domain sample:

Consistency loss: train **student** network to minimise squared difference between probability predictions.
Further adaptations for domain adaptation described in our earlier work [French18]

(separate batches for source/target, confidence thresholding, class balancing loss)
Faster R-CNN for object detection
Faster R-CNN [Ren15] is composed of two parts:

Region proposal network (RPN)
R-CNN head (final output)
RPN

Region proposal network (RPN) generates proposed boxes that may surround objects of interest
RPN

RPN is a fully convolutional network that generates predictions on a regular grid.
RPN

RPN Predictions correspond to anchor boxes; regularly spaced boxes across the image (constant for each resolution)
RPN

RPN predictions are combination of probability of presence of object and box-deltas that scale and move an anchor box to match that of a detected object.
RPN

Boxes from the RPN are filtered using non-maximal suppression (NMS), resulting in *proposals*.
R-CNN head

The *proposal* boxes are used to crop regions from upper layers of backbone network.
R-CNN head

These feature crops as passed to the R-CNN classification and regression network that determines the class of the detection and predicts final box deltas to refine the scale and position of the box.
R-CNN head

A final NMS filtering step yields the resulting detections
Self-ensembling for object detection
Model is Faster R-CNN that uses a ResNet-50 based feature pyramid network [Lin17] as a backbone.

We use mean-teacher, so two networks (teacher is EMA of student weights though).
For labelled (source domain images)

Data augmentation:

Random crop/translation
Horizontal flip
Uniform scale between 0.75x and 2.5x
For unlabelled (target domain images)

We augment the image twice (differently); one through teacher network, the other through student
For unlabelled (target domain images)

We found that limiting our target domain augmentation to translation/crops and horizontal flips worked best (no scaling).
We apply consistency regularization to the predictions from the R-CNN head of the network.
We found that applying consistency regularization to the output of the region proposal network (RPN) did not help.
We also found that attempting to use the predictions from the R-CNN head as pseudo-labels for the RPN didn’t help either.
Results
## VisDa 2018 detection results

<table>
<thead>
<tr>
<th>Team</th>
<th>Affiliation</th>
<th>Src mAP</th>
<th>Adapt mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>VARMS</td>
<td>JD AI Research, CV Lab</td>
<td>17.9</td>
<td>48.6</td>
</tr>
<tr>
<td>Ours</td>
<td>Colour Lab, UAE</td>
<td>10.2</td>
<td>13.5</td>
</tr>
<tr>
<td>UQ_SAS</td>
<td>University of Queensland</td>
<td>11.1</td>
<td>12.1</td>
</tr>
</tbody>
</table>

Self-Ensembling for Object detection
Conclusions
We have adapted self-ensembling to work in an object detection setting.
More work to do
See if we can improve performance

Analyse the effect of different parts of the approach
Test on different datasets
THANK YOU!
References