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 *consistency regularization* [Oliver18]

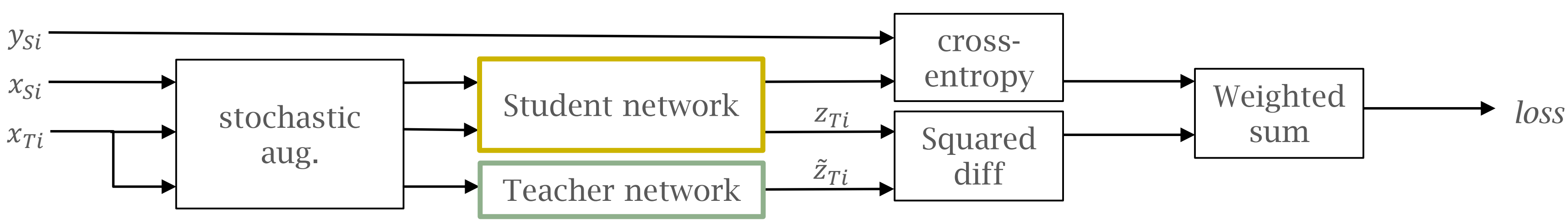
Self-ensembling developed for semi-supervised learning in [Laine17]

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

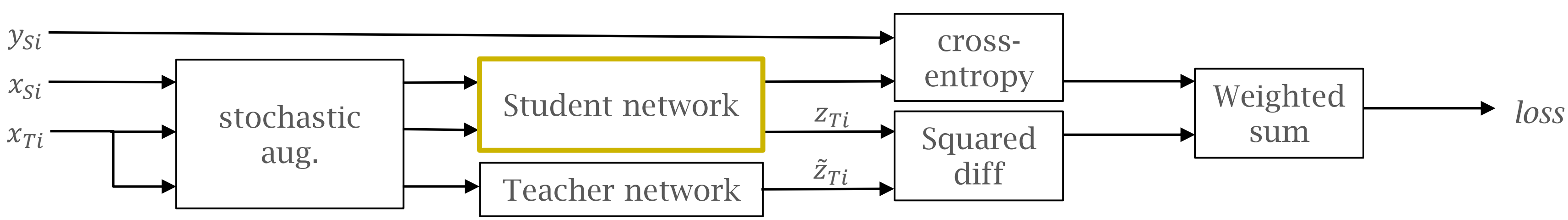
Student and teacher networks



x_{Ti}

Mean-teacher model

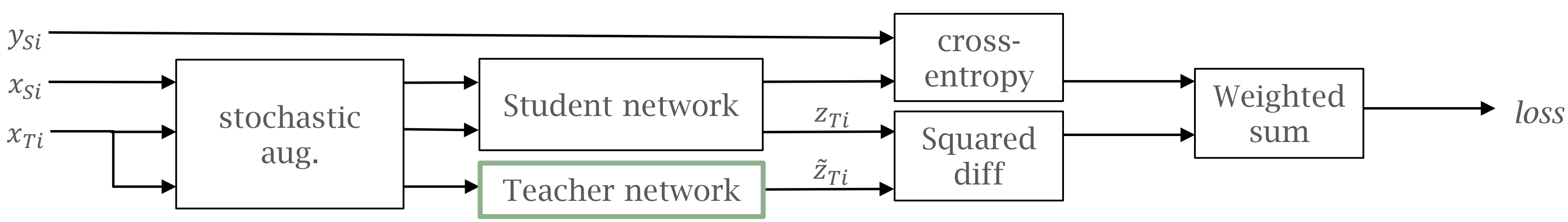
Student is standard classifier DNN



x_{Ti}

Mean-teacher model

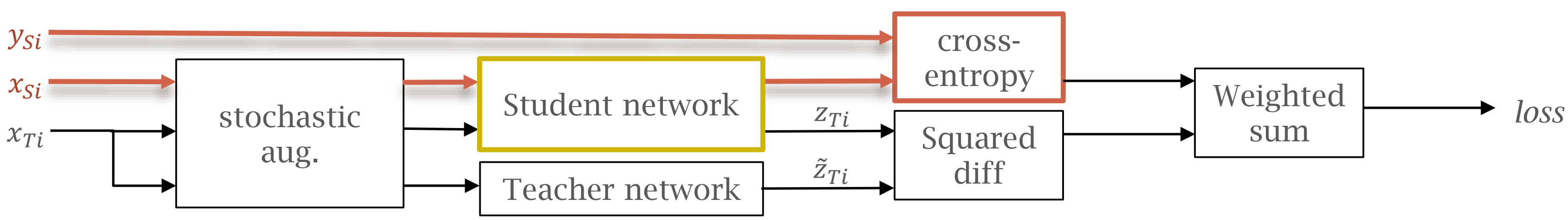
Weights of teacher network are exponential moving average of student network



x_{Ti}

Source domain sample:

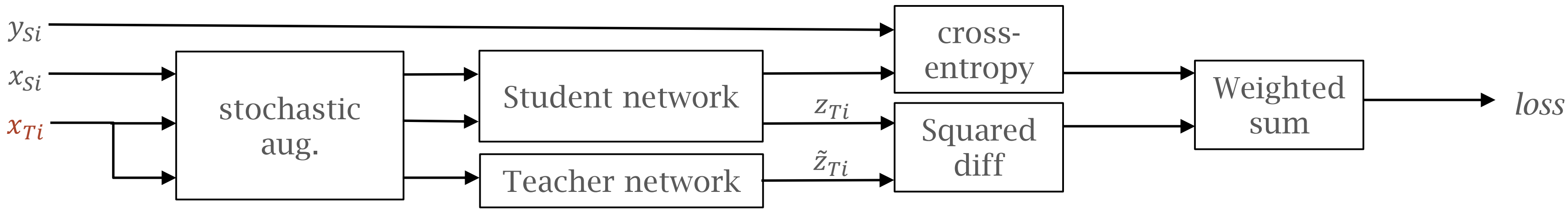
Predict class probabilities with student network and compute supervised cross-entropy loss (with data augmentation)



y_{Si}

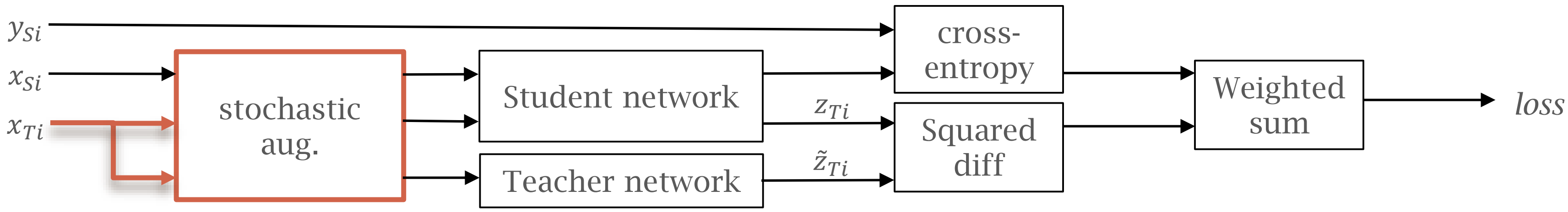
Target domain sample:

one sample



Target domain sample:

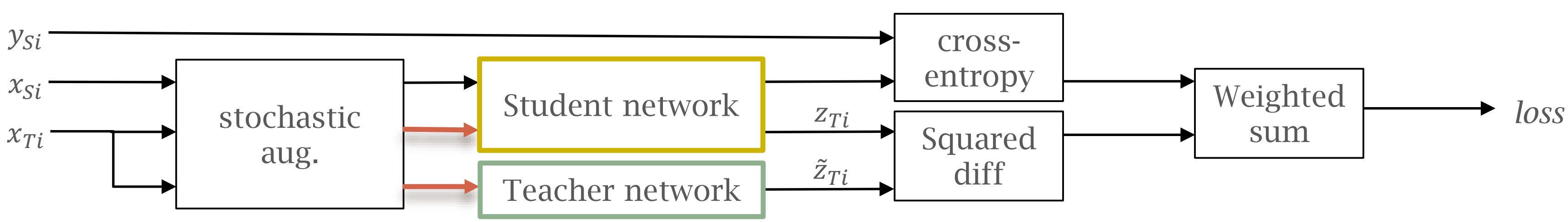
augment twice, differently each time (translation, flip)



x_{Ti}

Target domain sample:

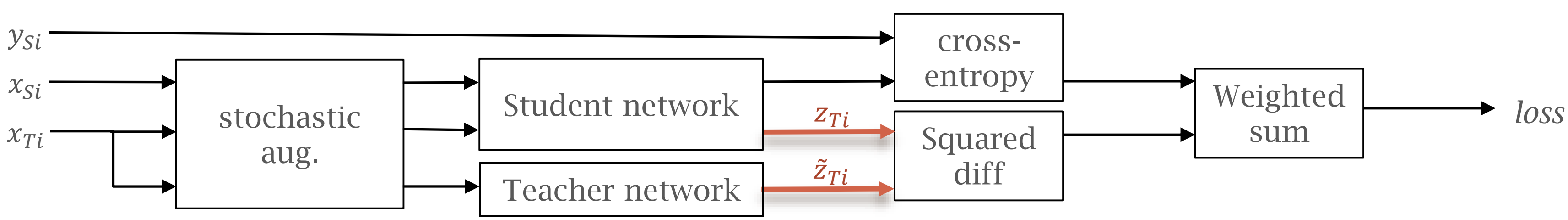
One path through student network
Second through teacher
(different dropout)



x_{Ti}

Target domain sample:

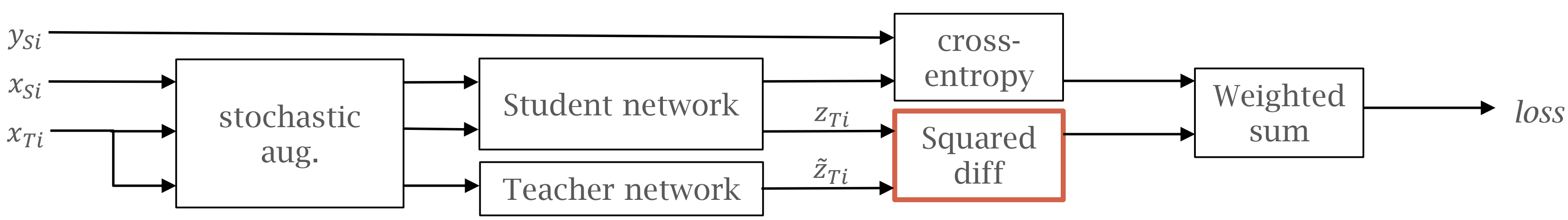
Result: two predicted probability vectors



x_{Ti}

Target domain sample:

Consistency loss: train student network to minimise squared difference between probability predictions



x_{Ti}

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

	Team	Affiliation	Src mAP	Adapt mAP
1	VARMS	JD AI Research, CV Lab	17.9	**48.6**
2	Ours	Colour Lab, UAE	10.2	13.5
3	UQ_SAS	University of Queensland	11.1	12.1

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

[French18] Geoff French, Michal Mackiewicz, Mark Fisher “Self-ensembling for visual domain adaptation.“ *ICLR 2018*.

[Laine17] Samuli Laine and Timo Aila. "Temporal Ensembling for Semi-Supervised Learning.“ *ICLR* 2017.

[Li16] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. "Revisiting batch normalization for practical domain adaptation.“ 2016.

[Lin17] Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B. and Belongie, S.J., "Feature Pyramid Networks for Object Detection" CVPR 2017

[Oliver18] Oliver, A., Odena, A., Raffel, C., Cubuk, E.D. and Goodfellow, I.J., "Realistic Evaluation of Semi-Supervised Learning Algorithms" 2018.

[Ren15] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” NIPS 2015.

[Tarvainen17] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.“ 2017.