ICLR 2018 REPRODUCIBILITY CHALLENGE REPORT

Twin Networks: Matching the future for sequence generation

Aditya Prakash - IIT Roorkee
Saurabh Mishra - IIT Roorkee

Paper Summary

The paper describes an approach to facilitate modeling of long-term dependencies in RNNs by implicitly forcing the forward states to hold information about the longer-term future, ie, effectively regularizing RNNs. This is done by using an additional backward RNN and jointly training both the networks. However, only the forward network is used during testing. The results are provided on various tasks like speech recognition, image captioning, language modelling and pixel-by-pixel generation on MNIST.

Target Questions

1. Verifying results for two tasks - image captioning on MSCOCO dataset and pixel-by-pixel generation for MNIST.
2. Implementation of baseline models to verify the scores mentioned in the paper.
3. Quantitative estimate of the training time taken by both networks.
4. Comparison of the nature of convergence of both networks.

Experimental Methodology

For implementation purposes, we use the code available on github:

- Image captioning : https://github.com/nke001/neuraltalk2.pytorch
- MNIST : https://github.com/nke001/Twinnet

Implementation Details

Image Captioning

- Implementation of both networks for show&tell and soft attention models
- Karpathy's training, validation and test split is used
- Resnet with 101 layers is used to extract features for attention models
- Input image size : 256 x 256 x 3
- Optimizer : Adam
- Learning rate : The paper mentions a fixed learning rate of 0.001. We experiment on both fixed and decaying learning rate (decay factor of 0.8 after every 3 epochs)
- The value of hyper-parameter α for twin loss is not mentioned in the paper. For implementation, α = 1.5 is used as mentioned for speech processing task
- Batch size is kept at maximum value possible with the GPU

Sequential MNIST

- Hugo’s binarized MNIST dataset is used
- 3 layer LSTM with 512 hidden units
- Optimizer : Adam
- Learning rate : 0.001
- Batch size : 50

Results

All the models are trained and tested on NVIDIA GeForce GTX 1080 Ti.

Image Captioning

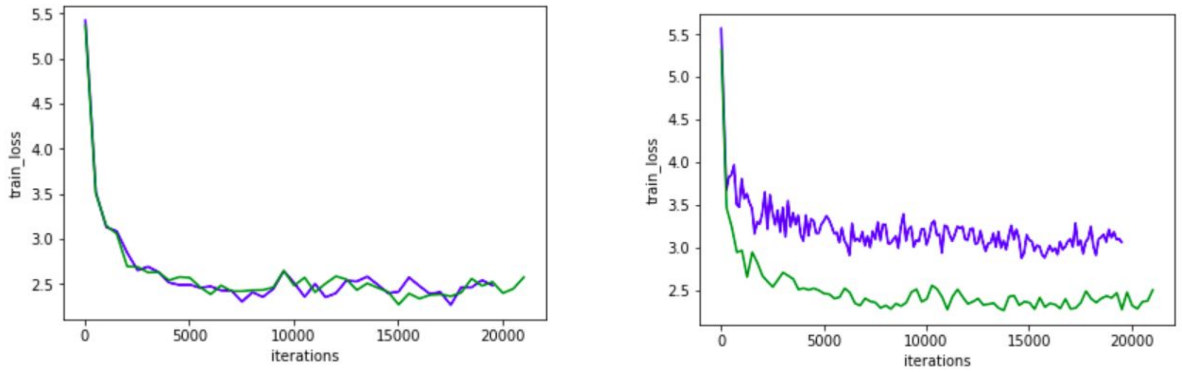
The results are reported on Bleu-1 to Bleu-4, Meteor, Rogue-L and CIDEr scores using the code available - https://github.com/tylin/coco-caption

	Bleu-1	Bleu-2	Bleu-3	Bleu-4	Meteor	Rogue-L	CIDEr
Show & Tell	0.725	0.555	0.415	0.310	0.249	0.529	0.953
Show & Tell Twin	0.705	0.531	0.390	0.287	0.239	0.517	0.884
Soft Attention	0.730	0.564	0.423	0.315	0.250	0.533	0.976
Soft Attention Twin	0.681	0.317	0.113	0.032	0.187	0.458	0.582

Training time :

Time per iteration (in seconds) for batch size - 64			
Show & Tell	Show & Tell Twin	Soft Attention	Soft Attention Twin
0.081	0.261	0.782	1.540

Nature of Convergence :



(a) Show and Tell

(b) Soft Attention

Figure 1: train_loss vs iterations plot for comparison of convergence between **baseline (green)** and **twin net (blue)**

Sequential MNIST

The results are reported in terms of negative log likelihood (NLL) and training time for both the baseline and twin networks.

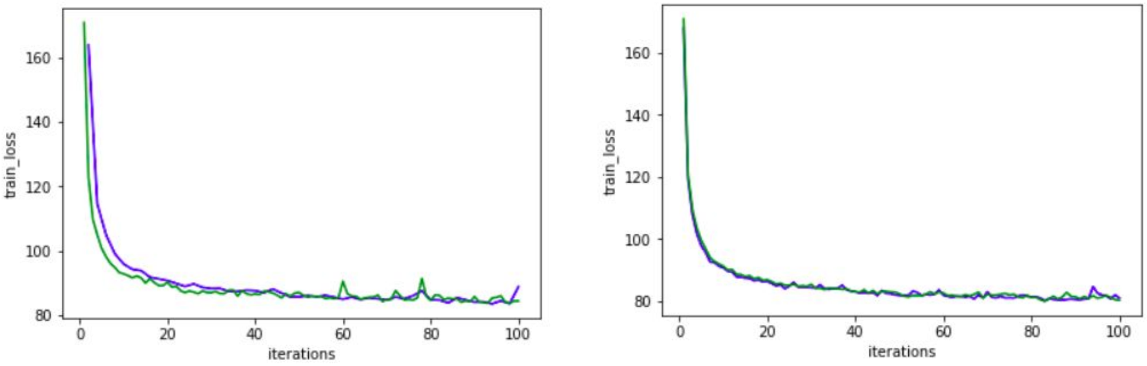
Unconditional Generation :

	Baseline	TwinNet
Negative log likelihood (NLL)	79.870	79.350
Training time per iteration (sec)	0.423	0.862

Conditional Generation :

	Baseline	TwinNet
Negative log likelihood (NLL)	83.710	80.870
Training time per iteration (sec)	0.423	0.871

Nature of Convergence :



(a) Unconditional Generation

(b) Conditional Generation

Figure 2: train_loss vs iterations plot for comparison of convergence between **baseline (green)** and **twin net (blue)**

Analysis and Discussion:

The idea is intuitive and well motivated and the paper is explained in great detail. However, based on our results, we believe that there is need for extensive experimentation and testing.

- For image captioning, we observe baseline values around 2-3 % higher whereas the twin net values show little change while using the same values of parameters as those mentioned in the paper. However, we believe that extensive hyperparameter search may lead to better results.

- The observed values for the soft attention twin net is quite less than those reported in the paper. This indicates that the convergence is highly affected, probably due to increased parameters and non-convexity in the optimization surface. This is also evident in the loss curve in Figure 1 (b) where we observe increased fluctuations.

- For unconditional generation of MNIST, we observe higher values of NLL which is probably because the parameters are not the same as mentioned in the paper. However, the values for baseline and twin net are almost the same in accordance with the paper.

- We also report values for the conditional generation of MNIST. The improved value for the twin net reaffirms the notion that the proposed approach works better for the conditional case. Also, the values for conditional case show improvement over the unconditioned case as expected.

We would also like to point out few **limitations**:

- Major downside of the approach is the cost in terms of resources. The twin model requires large memory and takes longer to train (~ 2-4 times) while providing little improvement over the baseline.

- During evaluation we found that the attention twin model gives results like “a woman at table a with cake a”, where it forces the model to look like a sentence from the backward side too. This might be the reason for low metric values observed in soft attention twin net model.

- The effect of twin net as a regularizer can be examined against other regularization strategies for comparison purposes.

Conclusion

The paper presents a novel approach to regularize RNNs and give results on different datasets indicating wide range of application. However, based on our results, we believe that further experimentation and extensive hyperparameter search is needed. Overall, the paper is detailed, simple to implement and positive empirical results support the described approach.