Deep Learning Lab: Assignment #3

Francisco Rivera Valverde
francisco.rivera@students.uni-freiburg.de

***Abstract*—The present work summarizes the use of a fully convolutional neural network to perform image segmentation. The architecture employed uses inverted bottlenecks with stride of 2 to reduce the dimensionality of the input image by 16x and then restore the dimensionality with transposed convolutions. Four configurations that analyze the impact of skipped connections are evaluated based on the Intersection over Union (IoU) metric. The maximum IoU obtained is 0.37 when the architecture used 3 skip connections on every 2x up sampling.**

***Index Terms*—Deep Learning, CNN**

I. Introduction

The surge of convolutional neural networks allowed developers to extract high level features of their dataset, on a very similar way as per how neurons interpret data [1]. On the field of computer vision, Convolutional Neural Networks like [2] were the epoch that motivated computer scientist to start employing deep neural network against the traditional object detection techniques.

This document relies on the convolutional neural network principles to extract dimensionality features that allow image segmentation. With image segmentation, the use can label each pixel belonging to a predefined category. Particularly, the architecture developed consisted of a Fully Convolutional Neural Network with 16x dimensionality reduction of the CamVid dataset, and it's corresponding upsizing.

The goal of the exercise is to compare how different skip connections from the down sampling, improve the performance of the net [3]. Hence, this document would use the Intersection over Union as a metric to compare a network with up to three stages of skip connections. To accomplish this goal, section II would summarize the architecture employed. Section III would elaborate on the training and dataset. Section IV would present the results of the architecture tested, and finally section V would present some conclusions and recommendations.

II. Fully Convolutional architecture

The architecture under discussion employs a similar concept to [3] in which an input image is down sampled by a ratio of 16x to extract dimensionality feature. Those features are then up sampled using the same ratio principle (16x upsampling) with transposed convolutions. The first down sampling convolutions, referred to as encoder, make use of the inverted bottleneck principle in which a wide-narrow-wide convolution allows to reduce the number of parameters to train[4].

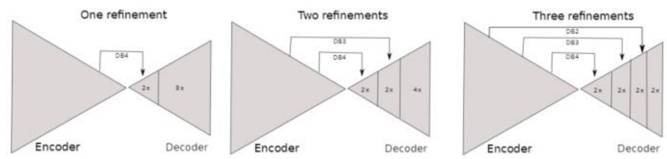

Fig. 1. Configurations with skip connections. A encoder structure reduces the dimensionality of the input image by 16x. On the last 3 down samplings of factor of 2x, a tensor is fed back to the up to the respective up sampling decoder as on [5]

Subsequent stages of inverted convolution with stride of 2 reduce the dimensionality of the model by 16x. While a a plain transposed convolution can restore the dimensionality, this document analyses the impact of concatenating skip connections during the up sampling. Such connections, as reported by [3], should help improve the recomposed image. Figure 1 shows three out of the 4 configurations of interest. Particularly relevant is that the skip connection come from the corresponding con/transposed convolution pair. The extra configuration to be tried out is having no skip connection at all.

To incorporate such a skip connection, the main stream is first upsampled by the desired factor. The biggest dimensionality signal, between the main stream and skip connection is cropped and concatenated into one before a 3x3 kernel convolution. The configurations tried out can be summarized as following:

1. Configuration 1: No skip connection and a transition up of 16x.
2. Configuration 2: Transition up of 2x with skip connection concatenated and a remaining 8x upsample.
3. Configuration 3: 2 blocks of 2x upsampling with respective skip connection concatenated. Additional remaining 4x upsampling at the end.
4. Configuration 4: 3 blocks of 2x upsampling with respective skip connection concatenated. Additional remaining 2x upsampling at the end.

III. Training the architecture

The above architecture is coded in TensorFlow and employs the Supervisor managed infrastructure to handle training. It used a CamVid dataset[1] with 41000 iterations with 500 steps per epoch. The baseline run for the experiment, which is a run without no skip connection, employed the following settings:

- Adam optimizer with 0.9 beta1

[1] Available from https://lmb.informatik.uni-freiburg.de/data/DL_Lab/data/assignment3_DL.tar.gz

- 1 image (300x300) as batch size
- 12 classes to predict

The methodology of the test was to train 4 different models and made use of the restore model functionality of TensorFlow to evaluate the performance of each configuration by iteration. The metric of performance used in here is IoU, so in other words, to test the model, a checkpoint of every training iteration is restored and IoU is calculated per iteration.

IV. Results

Such IoU per training iteration can be plotted over an IoU vs epoch point graph. Test data was used to create Figure 2, which shows how each extra skip connection slightly improves the Intersection over Union as a metric. To conceptualize this better, Table I one summarizes the top IoU seen across the epochs, where one can clearly see that using 4 skip connections increases IoU by more than 900 percent.

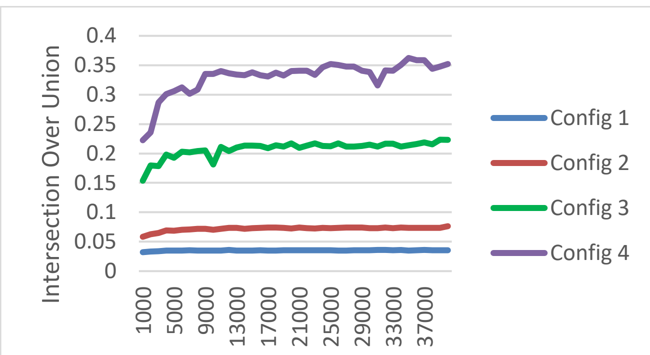

Fig. 2. Intersection Over Union at each one of the 41000 model checkpoints created during training. Configuration 1 means no skip connection was used, whereas config 2 to 4 increase 1 skip connection as per described in the architecture. There is noticeable improvement over each extra skip connection.

TABLE I. IOU OF THE DIFFERENT SKIP CONETIONS IMPLEMENTATIONS

configuration	***IOU***	***Improvement over Baseline (%)***
No Skip Conection	0.0359140	0
1 Skip Connection	0.0763853	113
2 Skip Connection	0.2234399	522
3 Skip Connection	0.3623841	909

Additionally, it is interesting to understand the impact during training of adding this skip connections. Figure 3 can help to observe a decrease in loss when 3 skip connections are added. With just one skip connection, there is a small improvement over the baseline, but incremental stack of skip connections shows a benefit. One can also see that even on early iterations, the model is able to converge nicely to a loss that is even lower than the final no skip connecting configuration

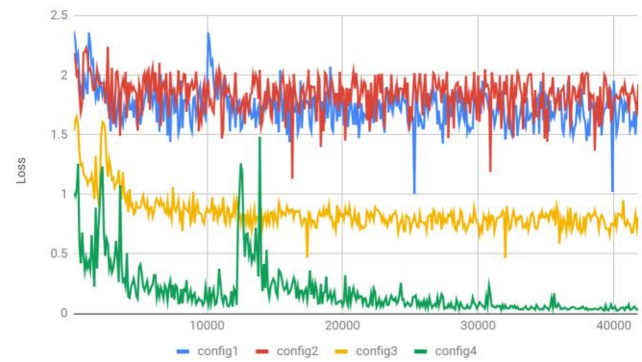

Fig. 3. Training Loss impact of incorporating skip connection. increase 1 skip connection as per described in the architecture. Loss is significantly decreased with skip connections

V. Conclusion

The use of skip connection on fully convolutional neural network can see a significant benefit of adding skip connections. Such skip connection complicates the network, but with the use of frameworks like TensorFlow, the complexity of the task is on the architectural design and not on the implementation.

However, even though the skip connections significantly improve the performance, the final IoU is still very bad yielding result lesser than 0.4. One can continue to increase the number of skip connections and define if there is an upper bound where this is no longer true, but this is a task left for future works.

.

References

[1] Cecotti, Hubert, and Axel Graser. "Convolutional neural networks for P300 detection with application to brain-computer interfaces." IEEE transactions on pattern analysis and machine intelligence 33.3 (2011): 433-445.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[4] Sandler, Mark, et al. "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation." arXiv preprint arXiv:1801.04381 (2018).

[5] Albert Ludwigs Universitat Freiburg. Masterpraktikum Deep Learning lab. Taken from http://dl-lab.informatik.uni-freiburg.de(2018)

[6] GoodFellow, Ian, et al. Deep learning. Cambridge: MIT press, 2016.

[7] Min Lin. Network In Network. URL: https://arxiv.org/abs/1312.4400. 2013