U-Net: Convolutional Networks for Biomedical Image Segmentation

May 30, 2019

This article is written as I read the U-Net paper by Ronneberger et al. [5]. U-Net is an architecture that, I think, is popular among later research works that perform image transformation. The most famous derivative work is the pix2pix paper [2]. By studying this paper, I will add a tool into my bag to tricks for network design.

1 Introduction

- Deep CNNs outperforms other methods when it comes to image classification. However, they require a lot of data to train. For example, AlexNet [3] was trained with the ImageNet dataset, which has 1M labelled training examples.
- Image segmentation, on the other hand, requires a label to be attached to each pixel. Image segmentation is common in biomedical imagae processing. However, biomedial datasets do not even have more than thousands of training examples.
- A work by Ciresan et al. trained a sliding-window CNN to segment patches instead of whole images [1]. This solves the data scarcity problem because there are more patches than whole images.
- However, Ciresan et al.'s approach has two drawbacks:
  - It is quite slow because you have to slide the network over the whole image.
  - There is a trade-off when choosing the patch size.
    - Larger patch size see more context but requires more max pooling layers, which reduce localization accuracy.
    - Small patch size affords better localization accuracy but it can only see little context.
- The authors propose a new architecture based on the "fully convolutional network" (FCN) [4]. They extend it so that it works with small training sets.
- The main ideas of the Long et al.'s paper are:
  - Make a classifier network convolutional by interpreting the last fully connected layers as a convolution filter that spans the whole image.
  - Upsample the final result of the classifier network to full resolution by a tranposed convolution layer. The layer is initialized to bilinear upsampling first, and the paper let the network learn the parameters.
  - Upscaled predictions from several layers are combined so that predictions from deeper layers (more context but less resolution) are combined with predictions for shallower layers (less context but more resolution).
- One feature of the FCN that the upsampled images does not have many channels. This is because the paper upsamples *predictions*: the number of channels is equal to the number of classes.

- The present paper's modifies the FCN by allowing the upsampling images to have a large number of feature channels. In this way, the network can propagate context information to higher resolution. The upsampling part is more or less equivalent to the downsampling part, yielding a U-shaped architecture.
- To segment a large image, the image is divided into tiles, where is tile is smaller than the input the network can take. The network is then fed the subimages in which the tiles are the center of the image. The missing area is taken from the part of the image that overlaps the receptive field. Pixels beyond the borders are filled by mirroring the source image.
- Biomedical data do not have many examples. The paper augments the data by applying elastic deformations.
- Cell segmentation tasks also have many touching instances of the same class. The paper uses a weighted loss where the background label between two touching instances have large weights.

2 Network Architecture

- The network takes as input a greyscale (1-channel) image of size 572×572. It outputs an image of size 388×388 with two channels. Supposedly, the first channel is the probability that the pixel is in a cell, and the second is the probability that it is outside a cell. The output 388×388 image corresponds to the center 388×388 image of the input image.
- The architecture is probably best described by a code snippet in Table 1.

3 Training

- To limit memory use, the paper uses the batch size of 1 and a large momentum factor of 0.99. This allows for previous samples to have impact on the optimization's trajectory.
- Let Ω denote the set of 2D positions in the output image. The loss used to train the network is the weighted cross entropy loss:

$$\mathcal{L} = -\sum_{\mathbf{x}\in\Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

  where $p_k(\mathbf{x}) = G_{1,k}(\mathbf{x})$ is the probability that Pixel $\mathbf{x}$ has label k, and $\ell(x)$ is the correct label of Pixel $\mathbf{x}$.
- The weight $w(\mathbf{x})$ of each pixel is computed as follows:

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 + \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right)$$

  where:
  - w_c is the weight function to balance the class frequecies. (I'm not actually so sure what this means. Is it a constant factor divided by the frequency of the class at that pixel?)
  - d_1 is the distance to the border of the nearest cell.
  - d_2 is the distance to the border of the second nearest cell.
  - $w_0 = 10$.
  - $\sigma = 5$ pixels.
- The paper uses He initialization.

Tensors	**Shape**
$A_0 = \text{input}$	$1 \times 572 \times 572$
$A_1 = \textsc{Relu}(\textsc{Convolve}(A_0, 3{\times}3, 64))$	$64 \times 570 \times 570$
$A_2 = \textsc{Relu}(\textsc{Convolve}(A_1, 3{\times}3, 64))$	$64 \times 568 \times 568$
$B_0 = \textsc{Max-Pool}(A_2, 2{\times}2)$	$64 \times 284 \times 284$
$B_1 = \textsc{Relu}(\textsc{Convolve}(B_0, 3{\times}3, 128))$	$128 \times 282 \times 282$
$B_2 = \textsc{Relu}(\textsc{Convolve}(B_1, 3{\times}3, 128))$	$128 \times 280 \times 280$
$C_0 = \textsc{Max-Pool}(B_2, 2{\times}2)$	$128 \times 140 \times 140$
$C_1 = \textsc{Relu}(\textsc{Convolve}(C_0, 3{\times}3, 256))$	$256 \times 138 \times 138$
$C_2 = \textsc{Relu}(\textsc{Convolve}(C_1, 3{\times}3, 256))$	$256 \times 136 \times 136$
$D_0 = \textsc{Max-Pool}(C_2, 2{\times}2)$	$256 \times 68 \times 68$
$D_1 = \textsc{Relu}(\textsc{Convolve}(D_0, 3{\times}3, 512))$	$512 \times 66 \times 66$
$D_2 = \textsc{Relu}(\textsc{Convolve}(C_1, 3{\times}3, 512))$	$512 \times 64 \times 64$
$E_0 = \textsc{Max-Pool}(D_2, 2{\times}2)$	$512 \times 32 \times 32$
$E_1 = \textsc{Relu}(\textsc{Convolve}(E_0, 3{\times}3, 1024))$	$1024 \times 30 \times 30$
$E_2 = \textsc{Relu}(\textsc{Convolve}(E_1, 3{\times}3, 1024))$	$1024 \times 28 \times 28$
$F_0 = \textsc{Up-Convolve}(E_2, 2{\times}2, 512)$	$512 \times 56 \times 56$
$F_1 = \textsc{Concat}(F_0, \textsc{Crop-Center}(D_2, 56{\times}56))$	$1024 \times 56 \times 56$
$F_2 = \textsc{Relu}(\textsc{Convolve}(F_1, 3{\times}3, 512))$	$512 \times 54 \times 54$
$F_3 = \textsc{Relu}(\textsc{Convolve}(F_2, 3{\times}3, 512))$	$512 \times 52 \times 52$
$G_0 = \textsc{Up-Convolve}(F_3, 2{\times}2, 256)$	$256 \times 104 \times 104$
$G_1 = \textsc{Concat}(G_0, \textsc{Crop-Center}(C_2, 104{\times}104))$	$512 \times 104 \times 104$
$G_2 = \textsc{Relu}(\textsc{Convolve}(G_1, 3{\times}3, 256))$	$256 \times 102 \times 102$
$G_3 = \textsc{Relu}(\textsc{Convolve}(G_2, 3{\times}3, 256))$	$256 \times 100 \times 100$
$H_0 = \textsc{Up-Convolve}(G_3, 2{\times}2, 128)$	$128 \times 200 \times 200$
$H_1 = \textsc{Concat}(H_0, \textsc{Crop-Center}(B_2, 200{\times}200))$	$256 \times 200 \times 200$
$H_2 = \textsc{Relu}(\textsc{Convolve}(G_1, 3{\times}3, 128))$	$128 \times 198 \times 198$
$H_3 = \textsc{Relu}(\textsc{Convolve}(G_2, 3{\times}3, 128))$	$128 \times 196 \times 196$
$I_0 = \textsc{Up-Convolve}(H_3, 2{\times}2, 64)$	$64 \times 392 \times 392$
$I_1 = \textsc{Concat}(I_0, \textsc{Crop-Center}(A_2, 392{\times}392))$	$128 \times 392 \times 392$
$I_2 = \textsc{Relu}(\textsc{Convolve}(I_1, 3{\times}3, 64))$	$64 \times 390 \times 390$
$I_3 = \textsc{Relu}(\textsc{Convolve}(I_2, 3{\times}3, 64))$	$64 \times 388 \times 388$
$G_0 = \textsc{Convolve}(I_3, 1{\times}1, 2)$	$2 \times 388 \times 388$
$G_1 = \textsc{Pixel-Wise-Softmax}(G_0)$	$2 \times 388 \times 388$

Table 1: U-Net architecture.

References

[1] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2843–2851.

[2] Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *CoRR abs/1611.07004* (2016).

[3] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., pp. 1097–1105.

[4] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *CoRR abs/1411.4038* (2014).

[5] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597* (2015).