Department of Computing

Imperial College of Science, Technology and Medicine

What is a U-Net?

Author: Anton Zhitomirsky

Contents

1 The architectural background **2**
- 1.1 Method . . . 2
- 1.2 Training . . . 3
  - 1.2.1 Energy Function . . . 3
- 1.3 Conclusion . . . 3
- 1.4 Considerations . . . 4

2 Implementation **4**

3 Proposed extensions for the U-Net **4**
- 3.1 Residual connections . . . 4
  - 3.1.1 [2] . . . 4
  - 3.1.2 [5] . . . 5
- 3.2 Dense connections . . . 5
- 3.3 Attention Mechanisms . . . 5

This is a sumamry of the contents of the file [7] and example of its implementation (from scratch and from the github)

1 The architectural background

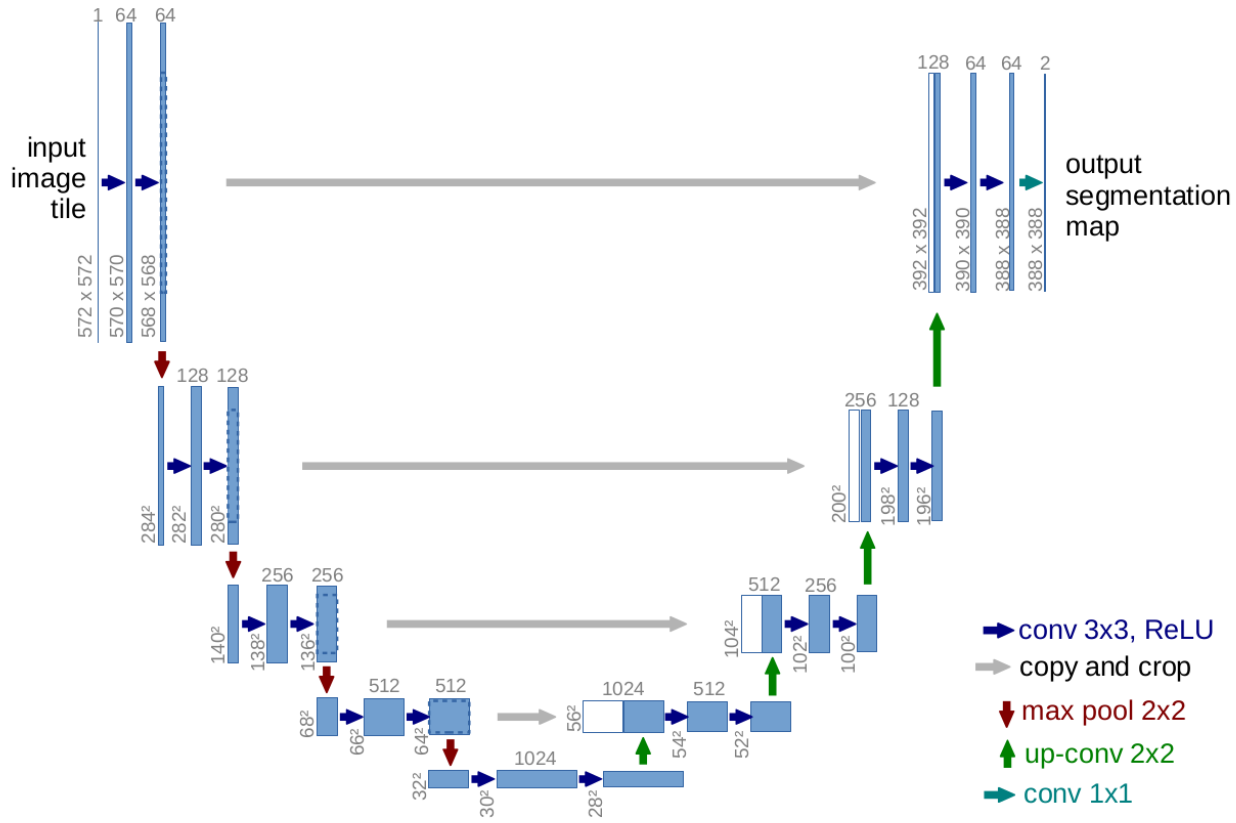

Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Figure 1: U-Net architecture taken from [7]

The U-Net is an adaption of another network [4]. We first discuss the architecture of the approach which is shown in Figure 1.

1.1 Method

Most of the method can be seen by referencing the Legend of Figure 1. It's worth noting that when we expand our features out we concatenate it with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions. The cropping is required due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes.

It is advised that you choose dimensions of an image that are even so that there is seamless tiling of the output segmentation map.

1.2 Training

This network favours large input tiles over a large batch size and hence they have reduced the batch to a single image. Accordingly they use a high momentum (0.99) such that a large number of the previously seen training samples determine the update in the current optimization step.

1.2.1 Energy Function

The energy function is computed by a pixel-wise soft-max (1) over the final feature map combined with the cross entropy loss function (2).

$$p_k(\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_{k'=1}^{K} \exp(a_{k'}(\mathbf{x}))} \tag{1}$$

- $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega, \Omega \subset \mathbb{Z}^2$
- K is the number of classes
- $p_k(\mathbf{x})$ is the approximated maximum-function (≈ 1 for the k that has the maximum activation $a_k(\mathbf{x})$ and ≈ 0 for all other k)

$$E = \sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{l(\mathbf{x})}(\mathbf{x})) \tag{2}$$

- Cross Entropy penalizes at each position the deviation of $p_l(\mathbf{x})(\mathbf{x})$ from 1 using E
- $l : \Sigma \to \{1, \cdots, K\}$ is the true label of each pixel
- $w : \Sigma \to \mathbb{R}$ is a weight map which gives some pixels more importance in training. More details in paper.

1.3 Conclusion

The resulting architecture is symmetric in its expansive path as it is to its contracting path which yields its notorious u-shaped name. It doesn't contain any fully connected layers and only uses the valid part of each convolution (the segmentation map only contains the pixels, for which the full context is available in the input image).

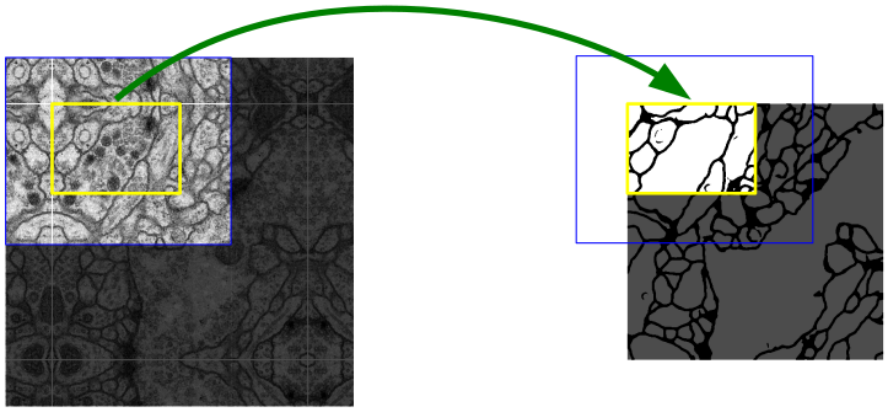

Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

Figure 2: Overlap figure taken from [7]

It takes advantage of the overlap-tile strategy (Figure 2). This is result of passing through the features received at each level and passing it horizontally through to the upsampled level on the opposite side of the network.

1.4 Considerations

"Tasks where little training data is available, we use excessive data augmentation by applying elastic deformations to the available training images. This allows the network to learn invariance to such deformations, without the need to see those transformations in the annotated image corpus".

This might mean that for our purpose we may squash, stretch and transform the inputs to simulate different human phsiology yet maintain the same relative architecture.

2 Implementation

3 Proposed extensions for the U-Net

This section has been left as a TODO.

3.1 Residual connections

3.1.1 [2]

This paper suggests that the forward and backward signals can be directly propagated from one block to any other block when using identity mappings as the skip conenctions and after-addition activation. code available at GitHub.

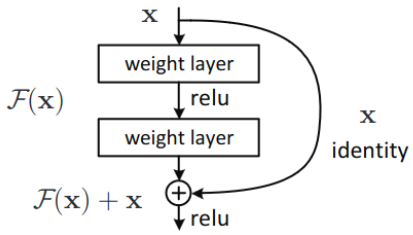

Figure 2. Residual learning: a building block.

Figure 3: Typic resudial network taken from [1]

The typical residual network as in Figure 3 makes training very deep networks easier than previous iterations. The formula typically follows

$$\mathbf{y}_l = h(\mathbf{x}_l) + F(\mathbf{x}_l, W_l) \tag{3}$$
$$\mathbf{x}_{l+1} = f(\mathbf{y}_l) \tag{4}$$

$\mathbf{x}_l$ and $\mathbf{x}_{l+1}$ are input and output of the l-th unit, and F is a residual function. Above, $h(x_l) = x_l$ is an identity funciton.

3.1.2 [5]

3.2 Dense connections

[3]

3.3 Attention Mechanisms

[6]

References

[1] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: (2015). URL: https://arxiv.org/pdf/1512.03385.pdf.

[2] Kaiming He et al. “Identity Mappings in Deep Residual Networks”. In: (2016). URL: https://arxiv.org/pdf/1603.05027.pdf.

[3] Simon Jegou1 et al. “The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation”. In: (2017). URL: https://arxiv.org/pdf/1611.09326.pdf.

[4] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. Tech. rep. Berkley, 2015. URL: https://arxiv.org/pdf/1411.4038.pdf.

[5] Fausto Milletari1, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”. In: (2016). URL: https://arxiv.org/pdf/1606.04797.pdf.

[6] Ozan Oktay, Jo Schlemper, and Loic Le Folgoc. “Attention U-Net: Learning Where to Look for the Pancreas”. In: (2018). URL: https://arxiv.org/pdf/1804.03999.pdf.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.