STAT441 Final Project

Hudson Ash Alexandre Xiao Yuchi Zhang Ziqiu Zhu

December 12, 2018

1 Introduction

Capsule networks is a new type of convolutional neural networks (CNNs) introduced by Sabour et al. (2017). They are motivated to address the drawbacks of CNNs, mainly the lack of spatial awareness of the neurons. Capsules solve this problem by storing neuron information as vectors instead of scalars.

Recently, the application of capsule nets are limited to image classification. LaLonde Bagci (2018) devised an encoder-decoder architecture for capsule nets to perform semantic segmentation called SegCaps. Their network produces two outputs, the segmentation estimate, and an image reconstruction. The reconstruction component is used to regularize the learned embedding of the input space, but is tossed away during prediction phase. We propose to incorporate an supervised conditional random field (CRF) loss as a new regularizer for the network, that is based on pixel affinity.

2 Related Works

2.1 Original Capsule Networks

2.2 Capsule Net for Semantic Segmentation

In 2015, SegNet was developed by researchers at the University of Cambridge to perform this semantic segmentation using a convolutional encoder-decoder architecture. The model architecture can be summarized as follows: the input image would be passed through a CNN known as the "encoder" network, then upsampled through another CNN known as the "decoder" network and then finally fed into soft-max classifier for pixel-wide classification. In the original paper, SegNet used the first 13 convolutional layers of the pretrained VGG16 network as the encoder, where every encoding layer had a corresponding decoding layer. Additionally, the use of max-pooling in other networks led to rougher segmentation results, since these layers would throw out valuable feature map data for boundary identification. To address this, in each step of the encoder network of SegNet, the index location of the maximum feature value in the max-pooling layers were saved and then used by the decoder network to perform a non-linear upsampling to produce a sparse feature map.

The proposed convolutional-deconvolutional capsule network, SegCaps, uses a similar encoder-decoder architecture as described above with a modified dynamic routing algorithm to reduce the computation cost by performing:

- Locally-Constraint Routing: Dynamic routing was only performed for children within a local region of the parent capsules
- Transformation Matrix Sharing: Transformation matrices were only shared between members of the same capsule type

To make up for the resulting loss of connectivity between global capsules, deconvolutional capsules are inserted to further extend the capsule network. The resulting model had very competitive results when compared to other state-of-the-art models, demonstrating the strong potential of capsule networks in image segmentation tasks.

3 Our Approach

3.1 SegCaps Loss

In the SegCaps network, the loss function is formulated as

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{BCE}}(\theta) + \lambda_{\text{Recon}}\mathcal{L}_{\text{Recon}}(\theta)$$

where the binary cross entropy loss is

$$\mathcal{L}_{\text{BCE}}(\theta) = -\sum_{p \in fg} \log P(s_p = 1; \theta) - \sum_{q \in bg} \log P(s_q = 0; \theta)$$

and the reconstruction loss is

$$\mathcal{L}_{\text{Recon}}(\theta) = ||X - \hat{X}||_F$$

We have that the set of all pixels is $\Omega = fg \cup bg$, where fg and bg stands for foreground and background, respectively. θ is the learned network weights. $s_p \in \{0, 1\}$ is the label of pixel p. X is the entire input data matrix; $\hat{X}$ is the corresponding reconstruction. According to LaLonde and Bagci, the reconstruction loss is added as a regularizer to ensure that the learned features in the encoder structure are reasonable, and promotes a better embedding of the input space.

3.2 Our Regularizer

We maintain the binary cross entropy loss, common to binary classification problems. However, we propose to incorporate a regularizer that is more closely related to the task of semantic segmentation, instead of reconstruction. In Tang et. al. (2018), they proposed a conditional random field (CRF) loss to convnets. They have that

$$\mathcal{L}_{\text{CRF}}(\theta) = \sum_{p \in \Omega} \sum_{q \in \Omega} W_{pq}(s_p - s_q)^2$$

where W_{pq} is the association or affinity of pixels p and q. In their work, W_{pq} is the Gaussian kernel distance of the RGBXY channels of pixels p and q. The XY channels specify the pixels' locations.

The CRF loss is completely unsupervised. It is motivated by the idea that similar local pixels should have similar predicted posterior probabilities. If the pixels are drastically different, then the loss does not "care" about the predicted posteriors. This is achieved by the negative exponential (kernel distance).

The loss is easily differentiable with respect to the predicted posteriors s_p. We have that

$$\frac{\partial}{\partial s_p}\mathcal{L}(\theta)^{\text{CRF}} = \sum_{q\in\Omega} 2W_{pq}(s_p - s_q)$$

In our approach, we implicitly consider the pixel locations by defining a local neighbourhood $\mathcal{N}_p$, such that

$$\mathcal{L}(\theta)^{\text{CRF}} = \sum_{p\in\Omega}\sum_{q\in\mathcal{N}_p}(s_p - s_q)^2 \exp\Big(-\frac{||I_p - I_q||^2}{\sigma^2}\Big)$$

where I_p is the RGB values of pixel p. In practice, we choose $\sigma^2 := |\hat{\Sigma}_X|$, to be the determinant of the empirical covariance matrix of the images. We also set neighbourhood $\mathcal{N}_p$ to be the 5 pixels to the right of p and 5 pixels to the bottom of p.

3.3 Our Loss

We formulate our network loss as

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{BCE}}(\theta) + \lambda_{\text{CRF}}\mathcal{L}_{\text{CRF}}(\theta)$$

4 Experimental Results

We use a dataset of satellite images to perform road segmentation. The dataset contains 100 images, 80 of which we use for training, and the remaining 20 for testing. Since capsule nets take very long to train, and due to limited computing resources, we are only able to train 100 epochs for each algorithm.

After 100 epochs, the original network obtained a binary cross entropy loss of 0.3529. λ_{Recon} is set to 50, and the learning rate is 0.0001. We see that the reconstruction loss falls very sharply during the first 10 epochs, and then stays flat for the remaining epochs. Only after the reconstruction loss plateaus, the cross entropy loss falls steadily. This may suggest that segmentation and reconstruction are closely related tasks.

After 100 epochs of our method, the cross entropy loss fell to **0.3008**, much lower than the original network. λ_{CRF} is set to 0.02, and the same learning as SegCaps is used. The CRF loss falls sharply at first, similar to the reconstruction loss. However, the loss is formulated based on pixel values and segmentation estimates, and relates closely to the task of interest.

Given the limited number of epochs, both algorithms did not converge to their final values. We have that the original SegCaps network achieved a Jaccard index of 0.61 on the test images, and our network achieved **0.67**. We show some test results from both algorithms below.

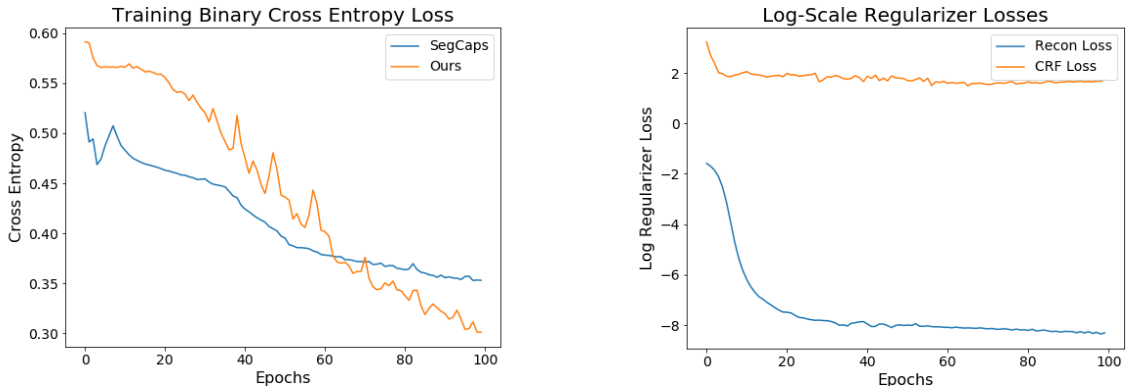

Figure 1: SegCaps Training Loss

References

[1] LaLonde, R. and Bagci, U. (2018). Capsules for Object Segmentation. arXiv preprint arXiv:1804.04241.

[2] Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions On Pattern Analysis And Machine Intelligence, 39(12), 2481-2495.

[3] M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus (2010). "Deconvolutional networks," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, pp. 2528-2535.

[4] Sara Sabour, Nicholas Frosst, Geoffery Hinton.

(a) SegCaps image 1 (b) Our method image 1 (c) SegCaps image 2 (d) Our method image 2

(e) SegCaps image 3 (f) Our method image 3 (g) SegCaps image 4 (h) Our method image 4

Figure 2: Test Examples