Segmentation of Medical Ultrasound Images Using Convolutional Neural Networks with Noisy Activating Functions

You Li LEOYOULI@STANFORD.EDU
Department of Bioengineering, Stanford University, Palo Alto, CA 94305 USA

Abstract

The attempts to segment medical ultrasound images have had limited success than the attempts to segment images from other medical imaging modalities. In this project, we attempt to segment medical ultrasound images using convolutional neural networks (CNNs) with a group of noisy activation functions which have recently been demonstrated to improve the performance of neural networks. We report on the segmentation results using a U-Net-like CNN with noisy rectified linear unit (NReLU) functions, noisy hard sigmoid (NHSigmoid) functions, and noisy hard tanh (NHTanh) function on a small data set.

1. Introduction

Medical ultrasound imaging is one of the mostly widely used medical imaging modalities. Compared with other modalities, including CT, MRI, and PET, ultrasound imaging has the lowest cost and is non-radioactive. However, the analysis of medical ultrasound images is more challenging than that with other modalities due to a few reasons. Firstly, ultrasound images are affected by speckle, which is an intrinsic noise associated with ultrasound imaging. Speckle gives ultrasound images a granular texture, limiting the ideal image SNR to be merely 1.91, and reduces image contrast as well as perceived resolution. In addition, compared to CT and MRI, the data collected by ultrasound scanners usually represents two-dimensional (2D) cross-sections of anatomy, rather than 3D. Therefore, the information of ultrasound image data is limited.

In this project, we segment medical ultrasound images. The inputs are medical ultrasound images. We train CNNs using these images and human segmentation results in the training set. The CNNs are then used to to segment ultrasound images in the test set, and the outputs are segmentation masks. The predicted masks are then compared with ground truth masks for the images in the test set. Quantitative measurement of the segmentation quality was conducted to explore the performance of the algorithm.

Submitted as the final project report of CS 229 Machine Learning, Autumn 2016.

2. Related Work

Previous effort in the automatic segmentation of medical ultrasound images has been focused on prostate, abdominal, or cardiac images, in which there is either strong contrast between region of interest and background, or there is a strong feature. Recently, Yang et al. (Yang et al., 2016) demonstrated the segmentation of prostate images using a fine-grained recurrent neural networks with very high Dice coefficients. Chen et al. (Chen et al., 2016) proposed an iterative multi-domain regularized deep learning method to segment cardiac ultrasound images, which has very high contrast and clear boundaries. The method leverages the transfer learning from cross domains and enhances the feature representations. Yu et al. (Yu et al., 2016) tackle a similar problem with dynamic convolutional neural networks. Cheng et al. (Cheng & Malhi, 2016) explored the use of transfer learning in segmentation of abdominal ultrasound images. Mechon-Lara et al (Menchón-Lara & Sancho-Gómez, 2015) utilized deep learning in the segmentation of carotid artery.

All these research share one strong assumption. In all images, it is assumed that the regions of interest (ROIs) (e.g. cardiac chambers or prostates) exist. This assumption may be valid in their desired applications, but is not necessarily valid in medical ultrasound images in general. One challenge in ultrasound scans is to find the target. In clinical settings, recorded images may not have the ROIs. This brings additional challenges to the training and prediction, which is not discussed in the previous research.

3. Dataset and Features

In this project, we used one publicly available dataset from one of the Kaggle challenges (Kaggle, 2016). It contains ultrasound images acquired on human necks, and the aim is

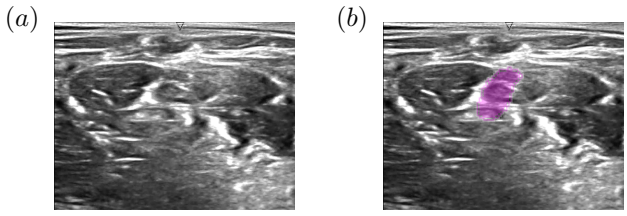


Figure 1. One example of (a) the medical ultrasound images in the dataset, and (b) segmentation of the image by trained human volunteers. The segmented nerves are represented in red.

to segment a collection of nerves called the Brachial Plexus (BP). The data set contains a training set that has been segmented by trained volunteers, and a test set. One example of the original image and its segmentation by human volunteers are shown in Fig. 1. Each image has 420×580 pixels.

The dataset imposes additional challenges to the segmentation task. As shown by the example, the region of interest (ROI) does not have a clear boundary against surroundings, and shares very similar texture with the rest of the image. No obvious feature is present. In addition, as described by the Kaggle challenge website, human mislabeling are to be expected. Participants of the challenge reported obvious mistakes by volunteers that segmented the images. False positives and false negatives in the identification of the nerves were reported as well. These factors contribute to the challenges of the task.

This is the only large and publicly available ultrasound image data set we could find that has been segmented and is allowed to be used. Unfortunately, the Kaggle challenge has already ended, and the ground-truth segmentation for the original test set is not released. Therefore, a new training set and a new test set have to be produced by splitting the original training set. The original training set contains 5635 images. Because of the split, a training set of 3606 (64% of the original training set), a validation set of 906 (16%) and a test set of 1127 (20%) were used.

4. Methods

A CNN with structure similar to the U-Net (Ronneberger et al., 2015) is utilized in this project. The architecture of the U-Net is illustrated in Fig. 2. This figure is reproduced from (Ronneberger et al., 2015). Each blue box in this figure represents a multi-channel feature map. The data pass through the horizontal lines simultaneously. The deep blue arrows represents activation functions. Originally, ReLU were used. In the current study, noisy activation functions were used to replace them. The implementation was based on a publicly available modification (Tyantov, 2016), using Theano and Keras. The package is used as a starting point of the project, because it provides the preprocessing

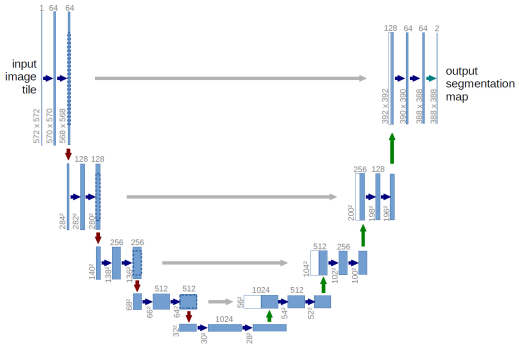


Figure 2. The architecture of U-Net. Illustration only. Numbers may be different from actual implementation in the project. This figure is reproduced from (Ronneberger et al., 2015). Each blue box in this figure represents a multi-channel feature map. The data pass through the horizontal lines simultaneously. The deep blue arrows represents activation functions. Originally, ReLU were used. In this study, noisy activation functions were used to replace them. In addition, the inception blocks were used to replace the original blocks of U-Net.

code to read data and the evaluation of the results using the Dice coefficients. In this implementation, batch normalization, Inception block, and the use of Dice coefficient loss function were used.

U-Net is chosen for this project for a number of reasons. Most importantly, the unique architecture of U-Net makes it suitable for biomedical image segmentation tasks. The architecture of of modified U-Net used in this project has two major components: the contracting path shown in the left half of Fig. 2, and an expansive path shown in the right half of Fig. 2. The contracting path is similar to other CNNs, which contains the repeated applications of two 3×3 convolutions followed by the noisy activation functions, and a convolution with stride 2 for downsampling. The expansive path is almost the reverse of the contracting path. It starts with an upsampling of the feature maps, and a 2×2 convolution that reduces the number of the feature maps, a concatenation with the cropped feature maps from the contracting path, and two 3×3 convolutions followed by the noisy activation functions. An additional 1×1 convolution layer is used as the last step to map feature vectors to the desired number of classes. This architecture produces high precision segmentation, because the expansive path has a large number of feature channels, which facilitates the passing of information to higher resolution layers. Thus, the high resolution features in the contracting path can be combined with the upsampled output and provide better localization of segmentation.

In addition, U-Net has been shown to have superior performance in biomedical image segmentation tasks where the number of training samples is limited (ISBI, 2012). However, during the trial experiment, U-Net has been less suc-

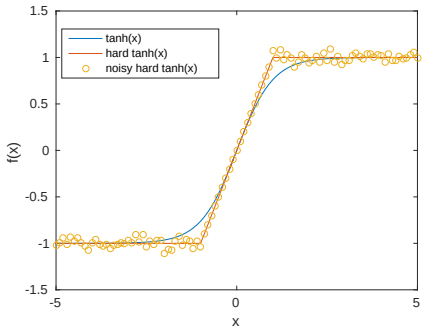


Figure 3. One example of noisy activation functions. The noisy tanh function, as well as tanh function and hard tanh functions are shown in this figure. Only zero-mean noise is shown.

cessful in this project than in previous reported projects. The Dice coefficients are approximately 0.4-0.5. We noticed that the challenges presented by the medical ultrasound images were novel to this network. The texture, the lack of boundaries or features, and the human segmentation errors in the training set make the task difficult. In order to improve the performance of it, we explore the possibility to use the noisy activation functions to push the algorithms out of local minima and improve its segmentation accuracy. Noisy activation functions (Gulcehre et al., 2016) were recently proposed and demonstrated to improve the performance the neural networks. Briefly, they can be constructed by cropping activation functions and adding noise to the saturated zone of the cropped functions. The noise serve to push the algorithm out of local minima and make the algorithm explore a larger area.

The details of noisy activation functions can be found in (Gulcehre et al., 2016). In this report, we only present a summary that is relevant to the implementation of the project. The noisy activation functions were produced by two steps: 1) clipping the original activation function (e.g. tanh function) to be a piecewise linear function using first-order Talor series expansion (termed hard functions), and 2) adding noise to the saturated zone of the activation functions, where the first order derivative is 0 (Gulcehre et al., 2016). As an example, Fig. 3 shows a comparison of tanh, hard tanh, and noisy hard tanh functions. The tanh function is clipped into a hard tanh function with the following form:

$$\max(\min(\tanh(x), 1) - 1). \tag{1}$$

Noisy hard tanh function is obtained by adding noisy to the saturated part of the hard tanh function

More complicated development on the noisy activation function include the use of hyper-parameters to influence the mean of the added noise:

$$\phi(x, \xi) = \alpha h(x) + (1 - \alpha)u(x) + d(x)\sigma(x)\xi, \tag{2}$$

where, $u(x)$ is the original activation function, $h(x)$ is the clipped version of $u(x)$, α is the hyper-parameter that adjusts the mean of the added term, $d(x) = -\text{sgn}(x)\text{sgn}(1 - \alpha)$, and $\sigma(x) = c(\text{sigmoid}(u(x) - h(x)) - 0.5)^2$. An explanation on the rationale of this form is beyond the scope of this report. Please refer to (Gulcehre et al., 2016) for details.

In order to study the impact of noisy activation function (Gulcehre et al., 2016) on the training and performance of the CNN, noisy activation functions were added to the implementation using zero-mean Gaussian noise. Noisy hard sigmoid (NHSigmoid), hard noisy tanh (NHTanh), and noisy ReLU (NReLU) functions have been added. Gaussian noise with a mean (α) of 0 and various power (σ^2) was added. NReLU has been proposed and analyzed in previous research (Nair & Hinton, 2010) and serves as a comparison for further analysis of the results produced using NTanh and NSigmoid. The number of epochs to achieve convergence and the Dice coefficients measured on the test set after each epoch are recorded. In addition, 5-fold cross-validation was used. The final segmentation result of each image is a mask, which has the value of 1 for the ROI and 0 otherwise.

The Dice coefficient is used as the metrics of the segmentation quality. It has been extensively used in the evaluation of medical image segmentation and it is also specified by the Kaggle challenge. It is defined as (Zhang et al., 2015)

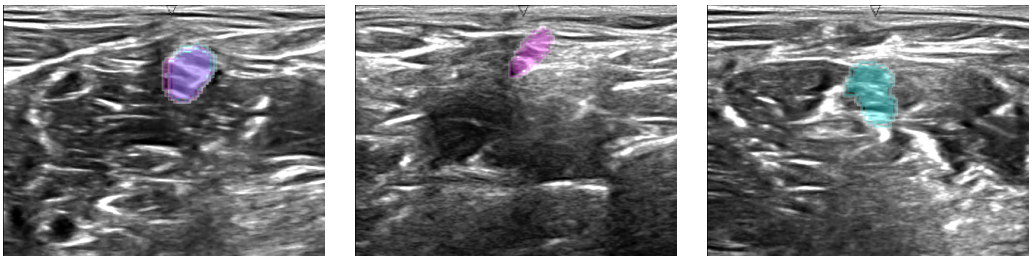
$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|}, \tag{3}$$

in which, $|X|$ and $|Y|$ represent the numbers of positive elements in the segmentation masks generated by the code and from the human volunteer, respectively. $|X \cap Y|$ is the number of the shared positively elements in X and Y. This measures the overlap of the predictions of the algorithm and the ground truth from human volunteers.

The algorithm was executed on a Dell workstation using Nvidia M4000 graphics card with 8 GB graphic memory. Because of the memory limit, a mini-batch size of 32 was used. A maximum of 50 epochs were specified, and only 20-30 were typically used to achieve convergence. Each epoch took approximately 5-8 minutes on the specified machines. The total training time for each set of parameters was 120-240 minutes. The Dice coefficients, false positive rate, and false negative rates were recorded and analyzed.

5. Results and Discussions

Fig. 4 shows examples of three different segmentation cases: (a) true positive case, where the algorithm correctly detects and locate the region of interest (ROI); (b) false negative, where the algorithm does not detect the nerve in the image; and (c) False positive case. The algorithm indicates



(a) True positive (b) False negative (c) False positive

Figure 4. Three examples of segmentation results by human volunteers and the algorithm. Red area:ground truth from human volunteer segmentation. Blue area: segmentation results by the algorithm. (a) True positive case. The algorithm correctly detects the nerves and achieve high agreement with the segmentation by human volunteers. Dice Coefficient = 0.90. (b) False negative case. The algorithm does not detect the nerve in the image. This is the case that should be avoided at all costs, because it may lead to injuries to patients. (c) False positive case. The algorithm indicates the existence of nerves in images where the nerves do not exist. In both (b) and (c), the Dice coefficients are 0.

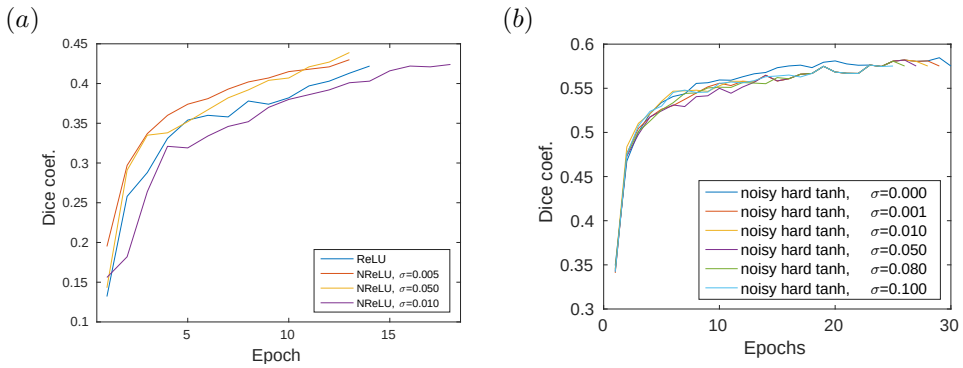


Figure 5. Dice coefficients as a function of epochs for (a) NReLU and (b) NHTanh functions. In general, the Dice coefficients increases with epoches. NHTanh has better performance than NReLU.

the existence of nerves in images where the nerves do not exist. Of all cases, (a) is the ideal one, and (b) is the one that needs to be avoided at all costs, because it may lead to injuries to patients.

Fig. 5 shows the Dice coefficients as a function of epochs measured on the validation set for (a) NReLU and (b) NHTanh functions. In general, the Dice coefficients increases with epoches. NHTanh has better performance than NReLU. As noise levels increase, the algorithm achieves higher Dice coefficients at convergence at the cost of more epochs.

Fig. 6 shows (a) Dice coefficients as a function of epochs at various noise levels. (b) Dice coefficients at convergence and the total number of epochs at convergences, both as functions of noise levels. Fig. 6(a) show that the algorithm achieves higher Dice coefficients at convergence at the cost of more epochs. In general, NHSigmoid produces higher Dice coefficients than NReLU or NHTanh functions. Fig. 6(b) show that the Dice coefficients increases as we increase the noise from 0. However, there is a saturation point after which adding more noise does not result in higher Dice coefficients. The number of epochs as a function of noise levels follows a similar trend as the Dice coefficients, indicating that the cost of achieving higher Dice coefficients is more epochs. However, even at convergence, the Dice coefficients are approximately 0.61, which is less than ideal. This is 17% lower than the best reported on Kaggle. Note that we have a 20% smaller training set for this project than Kaggle challengers.

Because the results of NHSigmoid is superior to those produced with NReLU or NHTanh, NHSigmoid is chosen for further analysis. Table 1 shows the measurement of Dice coefficients, false negative rate, and false positive rate at various noise levels. The false negative cases are the most detrimental to the project, because it may lead to injuries to patients. As for false positive cases, although we also would like to minimize their number, they are less of a concern. We noticed that the false negative rate of the algorithm is relatively low (2-5% in most cases), which is desirable. However, the false positive rate is high. We hypothesize that one major reason for such a high false positive

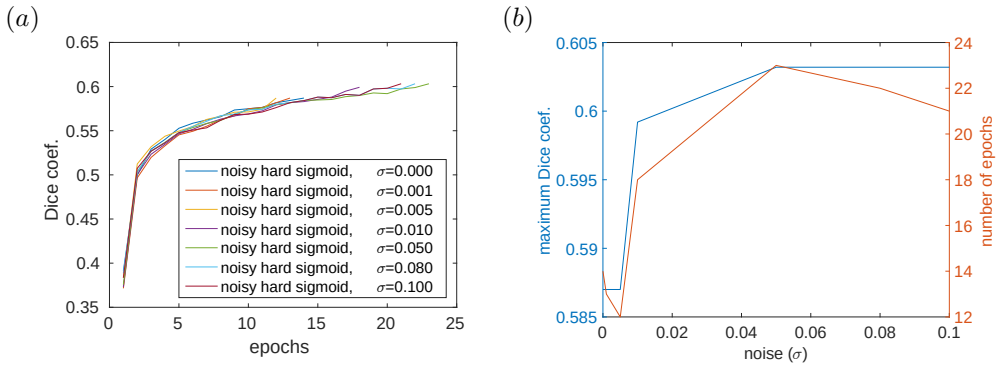


Figure 6. Results produced using NHSigmoid functions. (a) Dice coefficients as a function of epochs at various noise levels. (b) Dice coefficients at convergence and the total number of epochs at convergences, both as functions of noise levels. NHSigmoid produces higher Dice coefficients than either NReLU and NHTanh.

rate is the inconsistency of "ground truth" segmentation in the training set. We noticed that some images have positive identification of the nerves, while very similar ones (suspected to be the following frames from video stream) may have a negative reading by human volunteer. Participants in the Kaggle challenge reported that manual pruning of the data to remove such inconsistency results in significant improvement of segmentation results. However, because of the arbitrary nature of manual data pruning, it was not used in this study.

Table 1. The Dice coefficients, false negative rates, and false positive rates of the algorithms trained with noisy hard sigmoid activation functions at different noise levels. Measurement was done on the test set.

NOISE (δ)	DICE COEFF. (%)	FALSE NEGATIVE (%)	FALSE POSITIVE (%)
0.000	0.5241	2.00	63.77
0.001	46.61	3.30	68.00
0.005	50.31	2.96	63.77
0.010	57.94	5.23	46.96
0.050	55.81	4.44	55.08
0.080	59.06	5.43	48.37
0.100	61.83	9.57	35.34

In order to provide another perspective on the results, we measured the Dice coefficients of the true positive cases. Table 2 shows that in these cases, the Dice coefficients are approximately 0.75. This indicates that in these cases, the algorithm achieves better segmentation results. In addition, noise activation functions results in higher Dice coefficients if the noise level is between 0.01 and 0.1.

6. Conclusion and Future Work

We have segmented medical ultrasound images using convolutional neural networks with noisy activation functions. Compared to none-noisy activation functions, noisy activation functions can achieve better performance in segmentation in terms of Dice coefficients at the cost of computation time. In general, adding noise to the activation function results in an improvement of segmentation and an increase of computation time, until a saturation point is reached.

Table 2. The Dice coefficients in true positive cases for the algorithms trained with noisy hard sigmoid activation functions at different noise levels. Measurement was done on the test set.

NOISE (δ)	DICE COEFF. (%)
0.000	71.60
0.001	69.32
0.005	71.51
0.010	76.41
0.050	74.85
0.080	76.55
0.100	76.27

With U-Net and noisy hard sigmoid activation function, a Dice coefficient of 0.61 can be achieved. The main type of error is false positives. In most settings, the false negative rate of the detection and segmentation is low, which is desirable in the particular medical application. In true positive cases, the Dice coefficients at convergence is above 0.75.

In this project, we did not explore the impact of the mean of the noise on the performance of the algorithm. It can be a future direction. In addition, because of the inconsistency of ground truth segmentation in this project, just as in many medical imaging tasks, a pre-processing algorithm that cross-validate the images and remove such inconsistency will be desirable.

References

Chen, Hao, Zheng, Yefeng, Park, Jin-Hyeong, Heng, Pheng-Ann, and Zhou, S Kevin. Iterative multi-domain regularized deep learning for anatomical structure detection and segmentation from ultrasound images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 487–495. Springer, 2016.

Cheng, Phillip M and Malhi, Harshawn S. Transfer learning with convolutional neural networks for classification of abdominal ultrasound images. *Journal of Digital Imaging*, pp. 1–10, 2016.

Gulcehre, Caglar, Moczulski, Marcin, Denil, Misha, and Bengio, Yoshua. Noisy activation functions. *arXiv preprint arXiv:1603.00391*, 2016.

ISBI. ISBI challenge: Segmentation of neuronal structures in EM stacks, 2012. URL http://brainiac2.mit.edu/isbi_challenge/leaders-board-new.

Kaggle. Ultrasound nerve segmentation, 2016. URL https://www.kaggle.com/c/ultrasound-nerve-segmentation.

Menchón-Lara, Rosa-María and Sancho-Gómez, José-Luis. Fully automatic segmentation of ultrasound common carotid artery images based on machine learning. *Neurocomputing*, 151:161–167, 2015.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807–814, 2010.

Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.

Tyantov, Edward. Ultrasound nerve segmentation using keras, 2016. URL https://github.com/EdwardTyantov/ultrasound-nerve-segmentation.

Yang, Xin, Yu, Lequan, Wu, Lingyun, Wang, Yi, Ni, Dong, Qin, Jing, and Heng, Pheng-Ann. Fine-grained recurrent neural networks for automatic prostate segmentation in ultrasound images. *arXiv preprint arXiv:1612.01655*, 2016.

Yu, Li, Guo, Yi, Wang, Yuanyuan, Yu, Jinhua, and Chen, Ping. Segmentation of fetal left ventricle in echocardiographic sequences based on dynamic convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 2016.

Zhang, Wenlu, Li, Rongjian, Deng, Houtao, Wang, Li, Lin, Weili, Ji, Shuiwang, and Shen, Dinggang. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, 108:214–224, 2015.