Deep Learning Project Presentation on CS 732

Yanan Yang

CS 732 Project Presentation
New Jersey Institute of Technology
Newark, NJ, USA

PhD Presentation, 2019
Deep Learning Architecture:
Unet (for segmentation)
CheXnet (for detection and classification).

Data set:
Ultrasound image of the neck (from Kaggle) for Unet
ChestXray-14 (from NIH) for Chexnet

Result
1. Unet (Reproduced)

Layers of left side for downsampling:

- $l_{11} = \text{conv}(3, 1, 3)$
- $l_{12} = \text{conv}(3, 3, 3)$
- $l_{13} = \text{conv}(3, 3, 3)$
- $l_{21} = \text{conv}(3, 3, 6)$
- $l_{22} = \text{conv}(3, 6, 6)$
- $l_{23} = \text{conv}(3, 6, 6)$
- $l_{31} = \text{conv}(3, 6, 12)$
- $l_{32} = \text{conv}(3, 12, 12)$
- $l_{33} = \text{conv}(3, 12, 12)$
- $l_{41} = \text{conv}(3, 12, 24)$
- $l_{42} = \text{conv}(3, 24, 24)$
- $l_{43} = \text{conv}(3, 24, 24)$
- $l_{51} = \text{conv}(3, 24, 48)$
- $l_{52} = \text{conv}(3, 48, 48)$
- $l_{53} = \text{conv}(3, 48, 24)$

(input image tile)
1. Unet (Reproduced)

Layers of right side for upsampling:

- \( r_{11} = \text{conv}(3, 24, 48) \)
- \( r_{12} = \text{conv}(3, 24, 24), (\text{left } l_{42}) \)
- \( r_{13} = \text{conv}(3, 24, 12) \)
- \( r_{21} = \text{conv}(3, 12, 24) \)
- \( r_{22} = \text{conv}(3, 24, 24), (\text{left } l_{32}) \)
- \( r_{23} = \text{conv}(3, 12, 6) \)
- \( r_{31} = \text{conv}(3, 6, 12) \)
- \( r_{32} = \text{conv}(3, 6, 6), (\text{left } l_{22}) \)
- \( r_{33} = \text{conv}(3, 6, 3) \)
- \( r_{41} = \text{conv}(3, 3, 6) \)
- \( r_{42} = \text{conv}(3, 3, 3), (\text{left } l_{12}) \)
- \( r_{43} = \text{conv}(3, 3, 3) \)
- \( \text{finalOutput} = \text{conv}(3, 3, 1) \)
2. Implementation

- This is a 2D CNN implementation
- Smaller feature map size: 1, 3, 6, 12, 24, 48. Adv: more quickly to get output; Limitation: accuracy
- Train model using default dice loss, but only 5 epoch
3. data set

- Original set: ultrasound of the neck, from kaggle.com
- Train Set: 5,653 images from 47 patients
- What I use: 200 for training, 100 for testing
4. result

- Compared with ground truth binary mask, the network seems to correctly segment the area around it.
4. Result

- Unet Training Loss
- Unet Testing Loss
5. Future work

1. More data.

2. Have implemented Unet3D, still waiting for setting environment on my gup-station (continue on summer).
1. ChexNet Network

- Based on original paper: 121 layers DesnseNet\(^\frac{1}{4}\) default model by keras

- With the loss function: the sum of unweighted binary cross entropy losses

\[
L(X, y) = \sum_{c=1}^{14} [-y_c \log p(Y_c = 1|X) - (1 - y_c) \log p(Y_c = 0|X)]
\]
2. some initial value

1. pretrained model weights file is from DenseNet-Keras. epoch=100, batch size=32, learning rate=.001
2. Add 50% horizontal flip augmentation
3. Data set: Chest X-ray 14


2. 14 different thoracic pathology have been labeled

3. .csv files of default split can be downloaded on github.com/brucechou1983/CheXNet-Keras/tree/master/data/defaultsplit.
4 Experiments

- Randomly pick up the small size dataset training (40 images), validation (20 images), test (20 images)
- Change epoch = 6 (init epoch take too long)
- Downscale the images to 96*96 (224 is too large for my gpu)
some can get similar bounding box (left one), but some not (right one)
5. future work

1. since no source code can be found, I will try to use initial weights and input next phases to see if there’s still generate wrong boundary box

2. Train and test on the original data set (ChestXray-14). Due to the workstation I used need to reset environment these days, to be continued

3. No comparison and AUROC/Loss curve analysis
References

2. https://github.com/zhixuhao/unet
3. https://medium.com/coinmonks/learn-how-to-train-u-net-on-your-dataset-8e3f89fbd623

Thank You