Predicting semantic features from CT images of liver lesions using deep learning

Vibhu Agarwal and David Odgers*

Stanford University, School of Medicine
Department of Biomedical Informatics
vibhua@stanford.edu, djodgers@stanford.edu

Abstract

Driven by the technological advancements in imaging, the usage of medical imaging data is also increasing in research as well as in the clinical setting and is resulting in an increased burden on radiologists who wish to interpret and make use of this data. The wealth of information contained in this data is likely to have novel uses in diagnosis and care, if we are able to structure the information. Machine learning offers some hope of coming up with an efficient way to deal with this rapidly increasing burden as well as for structuring this information based on content. Recently, there has been interest in discovering hierarchical learning models that can simultaneously learn concepts at multiple levels from image data [4]. Such an algorithm could potentially learn useful high-level features from a data set of related, unlabelled images. Key challenges in scaling the technique to handle biomedical images involve addressing the issues of high dimensionality and translation invariance.

We attempt to implement a convolutional deep learning network to predict semantic features from images of liver lesions and gain insights into the conceptual and practical challenges involved in such an approach. We also have an opportunity to compare our results with those obtained from an earlier study that takes a classical machine learning approach to this problem.

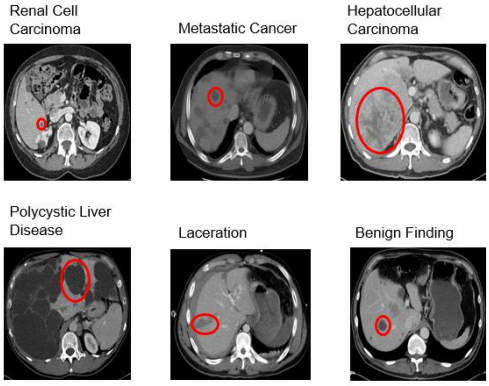
1 Introduction

Lesion types differ in how they are irrigated by blood vessels. As a consequence, they appear different during different phases of the circulatory cycle. Using contrast enhanced CT imaging a trained radiologist is able to differentiate between lesion types that can give rise to different diagnoses and outcomes (figure 1). Traditionally, the practice of radiological observations is based on annotating a region of interest (ROI) using a controlled vocabulary (RadLex) [3]. These annotations (also referred to as semantic features), while of value to clinicians are susceptible to human variability and may not be useful features for carrying out similarity searches on a database of CT images. Researchers in computer aided diagnostics have investigated the approach of using computationally derived features of CT images, as predictors of these semantic annotations[5][7]. Such a technique can improve computer aided diagnoses of lesions and alleviate issues related to human variability. In a recent study by Giminez et al.[1], a classical machine learning approach was taken to train logistic regression models on a 431-dimensional feature vector of computationally derived features. The output variable was a pruned list of the standard semantic annotations for each CT image. Since annotation of image data requires the expertise of a trained radiologist, obtaining large labeled data sets for training machine learning algorithms can be a challenge. In such a situation, unsupervised learning approaches can be an attractive

*Special thank you to Francisco Gimenez for aiding in a follow on study to original work

alternative to sourcing labeled training data.

Figure 1: *Sample CT scan images with associated clinical diagnosis*



2 Data

We were able to obtain 278 de-identified CT images showing liver lesions in the portal venous phase. These were provided by the Rubin lab (Department of Radiology, Stanford University) for the purpose of this study project. Each image represents the axial slice showing the largest area of the lesion. In images with multiple lesions, 2 - 3 additional lesions in addition to the main lesion were marked using the e-PAD software, giving a total of 504 lesion images. As it has been reported that training on image patches leads to better feature extraction in bio-medical images [2], we decided to use randomly sampled image patches for pre-training the image filters. A total of 4491 image patches each 21 pixels by 21 pixels, were sampled randomly from the available lesion images. Out of the total set of images 74 images were available with labels, where each label consists of a list of 86 yes/no values each corresponding to a Radlex descriptor.

2.1 Pre-processing

Since all the feature labels are not equally informative, it is necessary to have a pruned list of features so as to not have degenerate features in the training set. A feature selection technique to pick the most informative features is required; since we wanted to compare our results with the results obtained in the earlier study, we decided to adopt the same feature selection method as used in the earlier study[1]. This was based on selecting features with the highest binary entropy value. The binary entropy of a feature was calculated as:

$$H(X) = -p\log_2(p) - (1-p)\log_2(1-p) \quad (1)$$

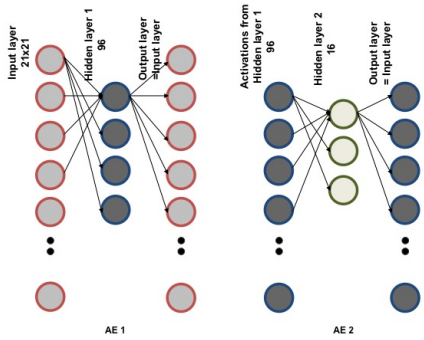
Applying this feature selection criterion, gave us a subset consiting of the top 30 features (figure 5) for training our system. Segmentation of lesion images was done on the basis of the region of interest (ROI) coordinates available in the image meta data. Each segmented image was scaled to a 60 pixels by 60 pixels size using the (default) bicubic interpolation method in matlab's imresize function. Scaling was done by the largest dimension so as to preserve the aspect ratio of the images. Padding was done using a reflection of the rows about the center line for the dimension that scaled to size less than 60. All pixel values were scaled to lie between 0 and 1.

3 Methods

Our strategy was to implement a stack of two auto-encoders to learn features from the randomly sampled image patches (figure 2). These pre-trained filters are then arranged within a neural network consisting of convolution and dense layers and fine tuned using back propagation. Since the optimization function in deep neural networks could be non convex, pre-training the filters greedily allows a way to trick the optimization objective by starting from a point that is likely to be closer to the optima. The selection of network configuration parameters (number of hidden layers, depth of the network, convolution and pooling parameters etc.) seems to be a somewhat ad-hoc process with a scientific approach for selecting these parameters, a subject of active research.

We decided to experiment with a few different network configurations to get a sense for the implementation complexity and performance trade offs associated with different configurations.

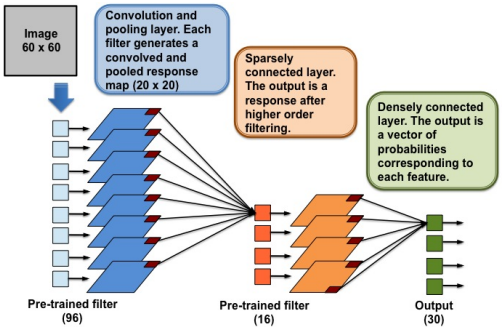
Figure 2: *The input to the auto-encoder 1 consists of the 21x21 randomly sampled patches*



3.1 Configuration 1

Our first configuration consisted of a network with 2 hidden layers comprising of the pre-trained filters and connected to a dense output layer as shown in figure 3. The first layer in the network is a convolution and mean pooling layer. The second hidden layer is also sparsely connected and forward feeds into the output layer. The output layer is dense (all input units connect into all nodes) and uses logistic regression to compute a probability value for each of the semantic labels being associated with the representation of the input image that is produced by the preceding two layers. The network was fine tuned using 67 training images at a time, in each iteration of a 10 fold cross validation loop. The starter code provided in the UFLDL tutorials[6] was helpful in working out the details of writing out the cost and gradient functions and using the LBFGS routine to find the optimal parameters.

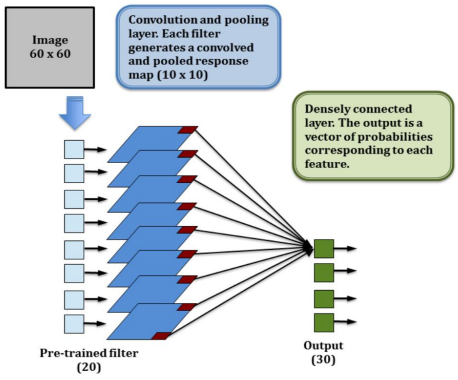
Figure 3: *Convolutional deep learning network - configuration 1*



3.2 Configuration 2

Our second attempt was based on a smaller network, consisting of just a single hidden layer doing pooling and convolution, as shown in figure 4. We also adjusted the value of the pooling dimension to mean pool a larger number of units.

Figure 4: *Convoutional neural network - configuration 2*



4 Results

4.1 Configuration 1

The classifier has very high test error, while it has zero training error. This is on account of the high sample complexity of the algorithm.

From Vapnik's theorem, training and generalization errors will be close provided m (the number of training data points) is O(d). However, in configuration 1 the number of parameters is much larger (240,000) compared to the number of data points (70). The over-fitting that we observed is as one would expect based on learning theory.

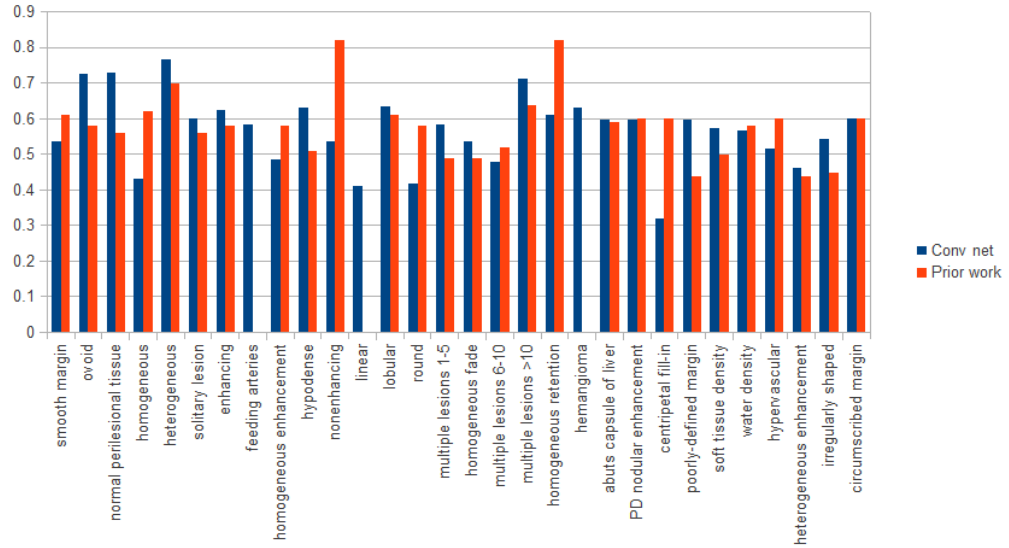
4.2 Configuration 2

The results from a ten fold cross validation on configuration 2 are as shown in figure 5. For comparison, results from the earlier study that used logistic regression on hand crafted features are also indicated alongside.

5 Discussion

The results from configuration 2 show that the classifier was able to learn features after fine tuning of the pre-trained filters. The feature wise AUCs are similar to the results obtained in the prior work (this was validated using a 2 sample t test that gave a p-value of 0.74, rejecting the null hypothesis). We believe that with a sufficiently large unlabeled data set, it should be possible to improve the learning and bring down the generalization error. With sufficient data to support a deep learning architecture, features learnt with larger data sets may be as good or better than hand crafted features.

Figure 5: *Area under the ROC curve values for convolutional neural network and logistic regression on hand crafter features. The 30 semantic features are on the x axis.*



Pooling helps make the system resilient to small linear translations at the expense of some information. However, for features that are related to the image texture, mean pooling may lead to some loss of information required to separate the data. Max-pooling may be a better choice for such features but may have an adverse impact for features that are based on image contrast. While our choice of the pooling dimension and technique was motivated by considerations of simplicity of implementation and achieving a smaller sized network,

the choice of a pooling technique is an important one for designing convolutional neural networks that work with biomedical images and requires further investigation as well as ingenuity.

5.1 Takeaways and future Work

We could obtain a practical appreciation of how convolution and pooling can provide a way to represent relevant features in large images with a relatively smaller number of parameters and thus mitigate the risk of over fitting. We could also understand the details of back propagating errors in a convolution neural network which were not intuitive at a first glance.

We would like to improve our understanding of convolution neural network design and develop at least heuristic guidelines for sizing such neural nets. We also wish to experiment with pooling techniques and observe how these impact the quality of the extracted features.

6 Acknowledgments

We would like to expressly thank Dr Daniel Rubin, Francisco Giminez, Assaf Hoogi, Sameep Tandon, Brody Huval and Andrew Mass for helping us learn about a topic that was challenging and was new to us, and for helping us stay motivated as we worked through the challenges.

We would also like to thank the team of people maintaining the UFLDL tutorials for providing an excellent resource to those interested in learning deep learning and unsupervised feature learning techniques.

References

[1] Giminez, F., et al. "Automatic Anotation of Radiological Observations in Liver CT Images." *AMIA Annu Symp Proc. 2012*. (2012): 257-63.

[2] Jamieson, Andrew R., Karen Drukker, and Maryellen L. Giger. "Breast image feature learning with adaptive deconvolutional networks." *SPIE Medical Imaging*. International Society for Optics and Photonics, 2012.

[3] Langlotz, Curtis P. "RadLex: A New Method for Indexing Online Educational Materials." *Radiographics*. 26.6 (2006): 1595-1597.

[4] Lee, Honglak, et al. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009.

[5] Manay, S., et al. "Integral Invariants for shape matching". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28.10 (2006) 1602-1618.

[6] Ng, Andrew, Jiquan Ngiam, Chuan Y. Foo, Yifan Mai, and Caroline Suen. "UFLDL Tutorial." *Ufldl at Stanford University*. n.d. Web. 12 Dec. 2013.

[7] Zhao, Cheng, et al. "Liver CT-image retrieval based on Gabor texture." *Engineering in medicine and Biology Society, 2004. IEMBS âĂŹ04. 26th Annual International Conference of the IEEE* (2004): 1491-1494.