Ongoing Work on Deep Learning for Lung Cancer Prediction

Oier Echaniz and Manuel Graña(✉)

Grupo de Inteligencia Computacional (GIC),
Universidad Del País Vasco (UPV/EHU), San Sebastián, Spain
manuel.grana@ehu.es

Abstract. Deep learning is one of the breakthrough technologies that have emergent in the last few years. It has been applied to a wide variety of problems, most of them related with image processing. It is also being considered for 3D data in medical image processing. This paper is a report of ongoing work about the development of deep learning architectures for lung cancer prediction. Data has been extracted from an ongoing Kaggel challenge, involving multi-center CTA data. First we have normalized in intensity the images. Then we have devised an auto encoder architecture with convolutional layers to obtain a compressed representation of the lung images. These representations are fed as features to a random forest classifier.

1 Introduction

In the United States, lung cancer strikes 225,000 people every year, and accounts for $12 billion in health care costs. Early detection is critical to give patients the best chance at recovery and survival. Realization of this urgent need has sparkled initiatives of the american institutions directed to improve the availability of data to researchers in order to advance on the detection and prediction issues, the so called Cancer Moonshot initiative[1].

As part of the activities under this initiative, a large dataset of CTA chest images from many hospitals and health institutions has been released and a computational challenge has been proposed in the Kaggle Data Science Bowl convening the data science and medical communities to develop lung cancer detection algorithms. The dataset offers thousands of high-resolution lung scans provided by the National Cancer Institute. The goal is set to develop algorithms that accurately determine when lesions in the lungs are cancerous. The aim is to reduce the false positive rate, which is very high for the current detection technology. Therefore, patients may get earlier access to life-saving interventions, while radiologists have more time to improve attention to their patients.

Deep learning is everywhere. Several articles [4] and works had already probe that deep learning is working really well in image based problems. In the last years, Convolutional Neural Networks (CNNs) [2,3] have achieved excellent

[1] https://www.cancer.gov/research/key-initiatives/moonshot-cancer-initiative.

J.M. Ferrández Vicente et al. (Eds.): IWINAC 2017, Part II, LNCS 10338, pp. 42–48, 2017.
DOI: 10.1007/978-3-319-59773-7_5

performance in many computer vision tasks. Several advances have solved convergence issues, and the advent of easy to exploit powerful Graphics Processing Units (GPUs) has speed up the training times by several orders of magnitude [1]. A CNN is a shared-weight neural network: all the neurons in a hidden layer share the same weights and bias. In fact, each layer implements a linear convolution filter whose kernel is learnt by gradient descent. Therefore, the output of the successive layers is a series of filtered/subsampled images which are interpreted as progressively higher level abstract features. Most CNN are applied to 2D signals, i.e. images, however in the medical image domain they are increasingly applied to 3D signals, i.e. volumetric imaging information. Autoencoders [6] are deep architectures that can be trained unsupervisedly, because their training error is the reconstruction error of the input after being processed by the entire auto encoder. The typical architecture has a middle hidden layer of small dimension, which is supposed to provide the features for further processing. Autoencoders have been used for soft organ segmentation [5].

The main objective of this work was to develop and compare existing deep learning methods capable of determining whether or not the patient will be diagnosed with lung cancer within one year of the date the scan was taken. When making this predictions we need to take in account that giving a wrong diagnosis is never equal, diagnosis as a non cancer patient into a cancer patient has less live cost than predicting a cancer patient to a non cancer patient, since no having treatment because of a wrong diagnosis will lead to death easily. Prediction method has to be accurate, reproducible and, above all, comparable to pathologists diagnosis.

2 Materials and Methods

Data. The dataset comes from a kaggle competition[2]. The dataset, provides over a thousand low-dose CT images from high-risk patients in DICOM format, coming from several institutions across the states. Each DICOM image sequence contains a series with multiple axial slices of the chest cavity which put together provide a 3D image of the chest of the patient. The number of 2D slices may vary between patients due to differences in the machines taking the scan. The ground truth labels (i.e. developing cancer or not) were confirmed by pathology diagnosis and were provided in the challenge dataset.

Server. We are using for this work a server with 2 connected nvidia 1080 GPU cards. The deep architectures have been implemented in Python using Keras[3] with Tensorflow as backend. For the explained methods the time considered for preprocessing is for about 10 h and to train the network for less than a day.

[2] https://www.kaggle.com/c/data-science-bowl-2017.
[3] https://keras.io.

Data Preprocessing. The image data were provided in Digital Imaging and Communication in Medicine (DICOM) format. For easier processing, we transform the images to a unique HDF file using Python scientific libraries. The individual image data had wide differences in intensity range, and resolution. Therefore, we need to carry out several preprocessing steps:

1. We have to correct the geometry of the image to a standard square capture layout. Some of the CAT systems have a circular filed of view.
2. We have to resample the images to obtain the same voxel size for all the images.
3. We have to correct the intensity in order to have the same correspondence of signal values to materials (air, fat, muscle, etc.).
4. We reduce the image size to $50 \times 50 \times 20$ by subsampling in order to be able to process the entire volume.

Figure 1 shows an example of a slice before and after preprocessing, the top row shows the histograms of the images in the bottom row, so that it is possible to appreciate the change in distribution made by the intensity correction. After preprocessing we have volumetric images of the same size. Figure 2 shows two example input volumes after preprocessing.

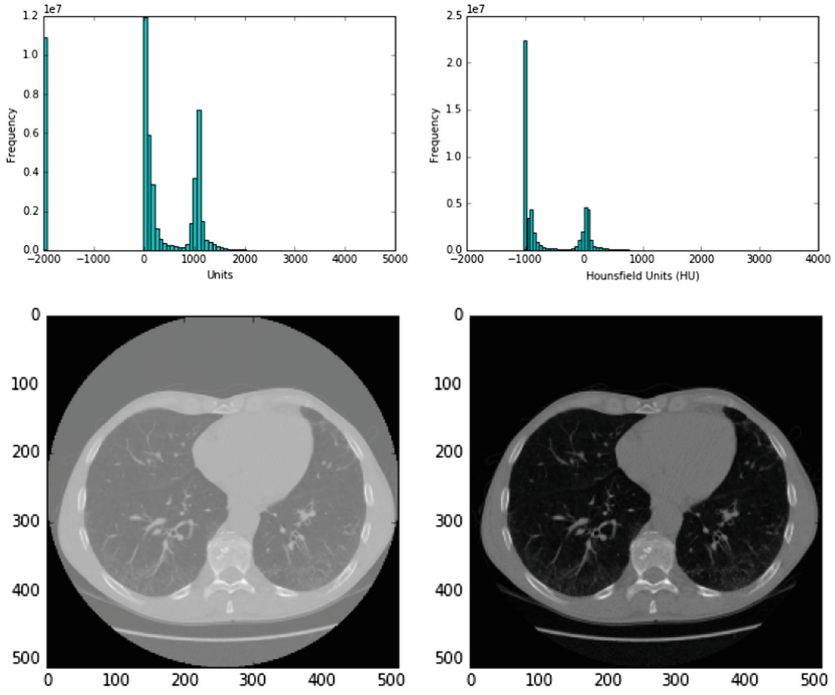
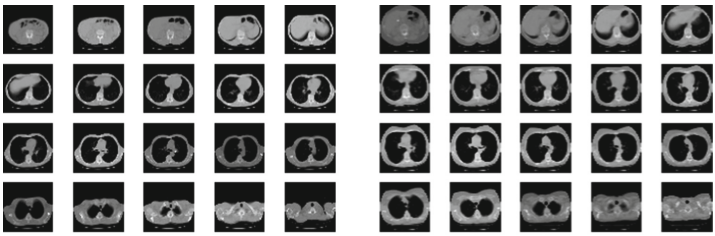


Fig. 1. Example of raw data (left) and preprocessed data (right). Top row: histograms of the images. Bottom row: visualization of the central slice.



(a) (b)

Fig. 2. An example of a input volume to the networks, showing the 20 axial slices. (a) cancer patient (b) no cancer patient.

Table 1. Autoencoder architecture layout

Layer	Output shapes	Params
Input	(None, 1, 50, 50, 20)	
Convolution3D	(None, 32, 50, 50, 20)	896
MaxPooling3D	(None, 32, 10, 10, 4)	
Dropout	(None, 32, 10, 10, 4)	
Convolution3D	(None, 64, 10, 10, 4)	55360
MaxPooling3D	(None, 64, 3, 3, 1)	
Dropout	(None, 64, 3, 3, 1)	
Convolution3D	(None, 64, 3, 3, 1)	110656
(**Code**) Dropout	(None, 64, 3, 3, 1)	
UpSampling3D	(None, 64, 9, 9, 3)	
ZeroPadding3D	(None, 64, 11, 11, 5)	
Convolution3D	(None, 32, 11, 11, 5)	55328
UpSampling3D	(None, 32, 55, 55, 25	
Convolution3D	(None, 1, 53, 53, 23)	865
(Decoded) Cropping3D	(None, 1, 50, 50, 20)	
Total params: 223,105		
Trainable params: 223,105		
Non-trainable params: 0		

Architectures. We have trained two architectures:

1. 3D Convolutional Neural Network (CNN). It is a conventional architecture with 3D input volume corresponding to the CAT volume, the output is the decision units, and we have two 3D convolution layers interspersed by three maxpooling layers that produce the dimension reduction.
2. Autoencoder + classifier: We build an auto encoder whose hidden layers are convolutional networks as specified in Table 1. The middle layer, denoted Code

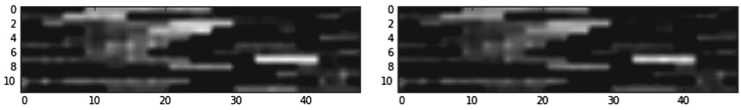


Fig. 3. The code achieved by the auto encoder after training for the input volumes in Fig. 2. Left cancer patient, right no cancer patient. The code has been reshaped into a matrix for visualization.

in the table, provides the features for classification carried out by conventional machine learning classifiers. Figure 3 shows the representation of the code for example cancer and non-cancer subjects, in fact it is not apparent the existence of discriminant features. We have tested Random Forest (RF) and Support Vector Machines (SVM), and k-NN with $k = 5$. The architecture has three convolution layers interspersed by maxpooling and dropout layers, all in 3D, to reduce the input to the Code dimensions. The reconstruction by up-sampling and zero padding interspersed by 3D convolutions.

We have benefitted from the great flexibility of Keras and easy specification of the architecture, as well as its easy interface to the GPUs for training speedup.

3 Results

One of the characteristics of the dataset is its class imbalance, there are much more non-cancer subjects than cancer patients. We have carried out training of the CNN with a small sample of 200 non cancer subjects and 100 cancer subjects, training it for 10 epochs. Results are shown in Fig. 4. The maximum accuracy is low, and there is a clear overfitting effect in the last epochs. The auto encoder architecture has been trained with three different training sets featuring diverse imbalance ratios, and they have been tested with 100 randomly selected subjects, 28 cancer and 72 non-cancer. Table 2 gives the results of our experiments so far. We provide the confusion matrices, whose rows correspond to the actual class,

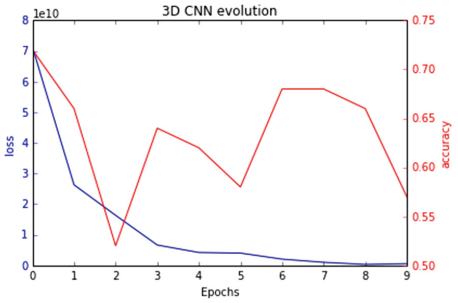


Fig. 4. The evolution of the error function (blue plot) and accuracy (red) of 3D CNN training. (Color figure online)

Table 2. Results given by the confussion matrices of the classifiers obtained with different distributions of imbalance of the test data for the auto encoder

training imbalance	RF			SVM			5-NN		
50 cancer 100 non cancer		nc	c		nc	c		nc	c
	nc	68	4	nc	48	24	nc	50	22
	c	26	2	c	16	12	c	16	12
100 cancer 50 non cancer		nc	c		nc	c		nc	c
	nc	19	53	nc	20	52	nc	12	60
	c	5	23	c	7	21	c	6	22
75 cancer 75 non cancer		nc	c		nc	c		nc	c
	nc	42	30	nc	32	40	nc	29	43
	c	14	14	c	11	17	c	11	17

and columns to the predicted class. Highest specificity (correct classification of cancer) is obtained when training with the imbalanced dataset containing more cancer subjects. These experiments are not according to the orthodox treatment of imbalanced datasets, which consist on one of the following strategies:

- manipulating the dataset adding new instances of the minority class by random interpolation between minority class samples, i.e. the SMOTE algorithm. Obviously, in the case at hand this amounts to generating new images of cancer prone patients, which is not feasible.
- manipulating the dataset removing instances of the majority class. This corresponds to the experiments with balanced datasets, which are not very successful.
- changing the error function to weight more the minority class errors. We are working on that solution as the most promising, but having to deal with technical problems.

The conclusion from Table 2 is that the auto encoder is still very sensitive to the class distribution of the training set, biasing towards the majority class in the training set.

4 Conclusions

Lung cancer is a very dramatic and urgent problem in many countries, specifically the initiative in the USA has brought this kind of cancer to the forefront of the search for innovative technical solutions to its diagnosis. The recent ongoing Kaggle challenge provides thousands of chest images from many medical institutions, which is a very hard testing ground for image based diagnosis tools. We are working with this data applying deep learning architectures. So far we have achieved the normalization of the images, and testing preliminary architectures with modest success. We have found that deep architectures are not immune to problems raised by imbalanced datasets, which are specially difficult to attack

when the input data are complex images where subtle features may induce dramatic change of the output. We are working in the near future to submit some competitive solution to the Kaggle competition.

References

1. Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in neural information processing systems, pp. 2843–2851 (2012)
2. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: a deep convolutional activation feature for generic visual recognition. CoRR, abs/1310.1531 (2013)
3. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
4. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, pp. 1717–1724. IEEE Computer Society Washington, DC, USA (2014)
5. Shin, H.C., Orton, M.R., Collins, D.J., Doran, S.J., Leach, M.O.: Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1930–1943 (2013)
6. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 1096–1103, New York, NY, USA, ACM (2008)