Data Science Bowl 2017

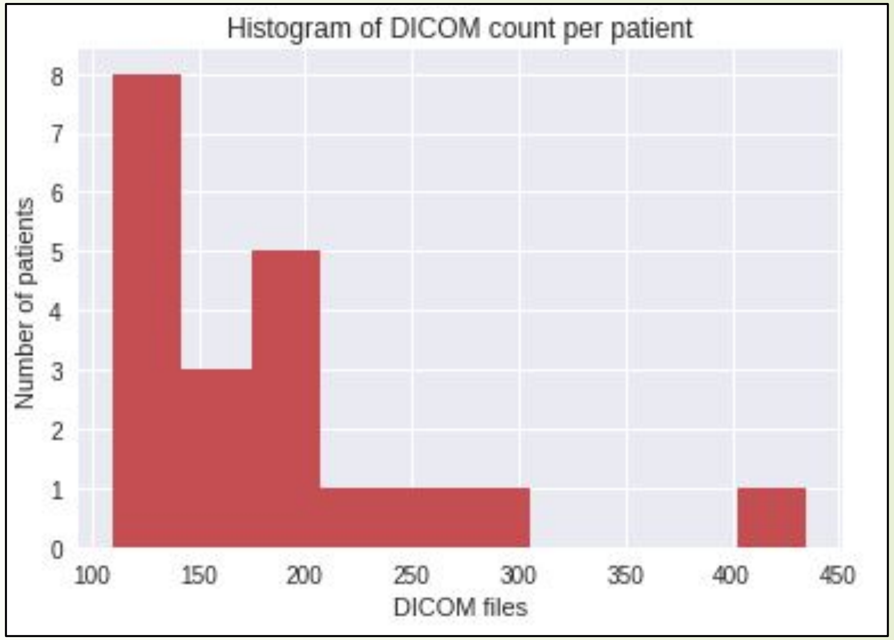
Introduction

- The Data Science Bowl 2017 aims at convening the data science and medical communities to develop lung cancer detection algorithms.
- The data set consists of thousands of high-resolution lung scans provided by the National Cancer Institute with the objective to accurately determine when lesions in the lungs are cancerous.
- We are given DICOM files, which is a format that is often used for medical scans. Using CT scans from 1400 patients in the training set, we have to build a model which can predict on the patients in the test set.

Preprocessing

- Since each image has a variable number of 2D slices, there are Nx512x512 in a 3D rendering for one patient where N varies based on the machine taking the scan and the patient themselves.

- We definitely need to resize the images to apply machine learning/deep learning algorithms, so as not to lose too much information as well as keep within the computational restraints.



- For now, using HPRC's Terra cluster I have been using an input size of 20x50x50. So we map each of the set of CT scans for a patient from:

$$N * 512 * 512 \rightarrow 20 * 50 * 50$$

Preprocessing pipelines

Process 1

Bring all the patients' scans to a standard 3-dimensional representation

Process 2

Using DICOM metadata,

- We convert the pixel values to Hounsfield Units (HU),
- Resample to an isomorphic resolution to remove variance in scanner resolution

Bring all the patients' scans to a standard 3-dimensional representation

Modeling

- 3D Convolutional Neural Network

This is a popular model for Computer Vision problems. Properties like translation invariance and being able to preserve the spatial structure of images makes it well suited for this application.

- XGBoost

With this and most other Machine Learning models we are required to flatten out the pixel grid (1-D representation).

Evaluation

The models are evaluated on the test set provided in the competition. Submissions were scored on the log loss:

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where

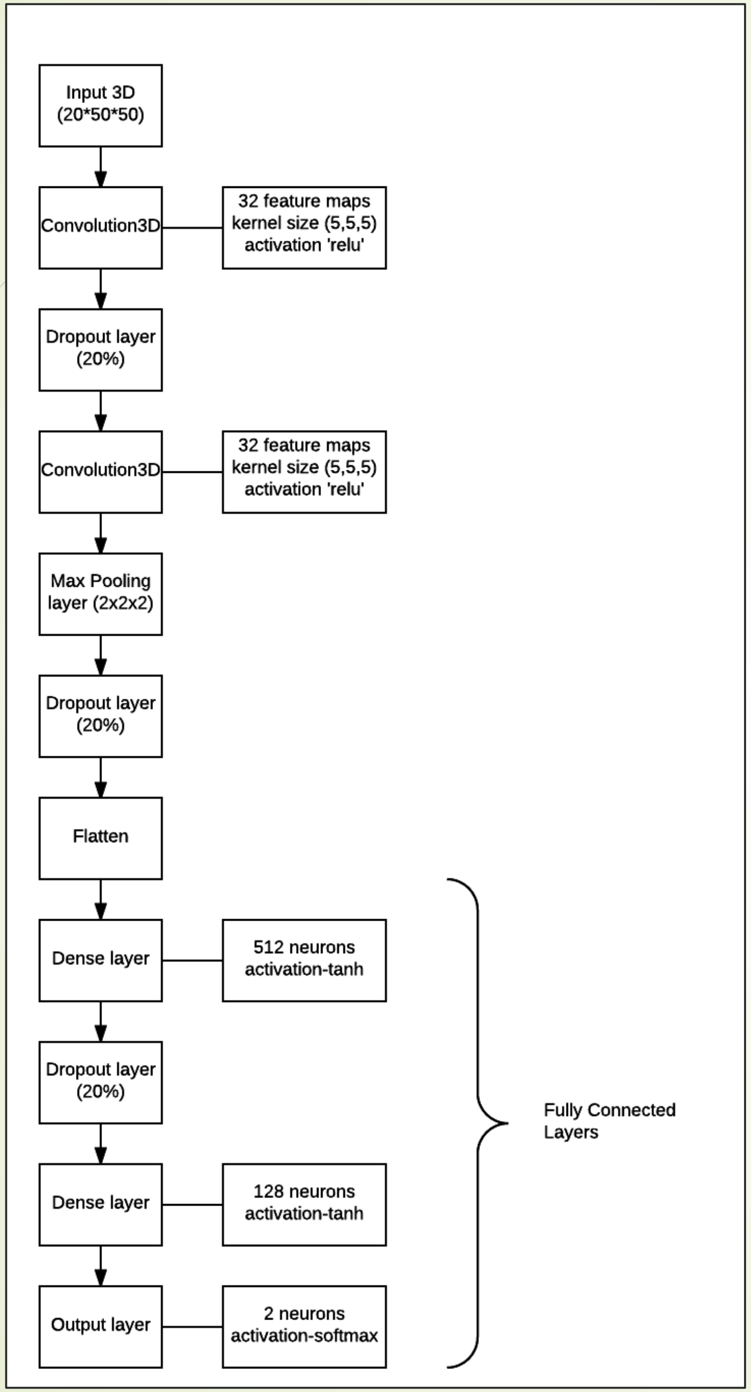
- n is the number of patients in the test set
- $\hat{y}_i$ is the predicted probability of the image belonging to a patient with cancer
- y_i is 1 if the diagnosis is cancer, 0 otherwise

Results

	Input dimension	Model	Logloss metric
Process 1	20x50x50	Convolutional Neural Network	0.6017
	20x50x50	XGBoost	9.594
Process 2	20x50x50	Convolutional Neural Network	0.6015
	20x50x50	XGBoost	9.942

Kaggle Competition:

I finished at 108^{th} in the leaderboard (1972 teams). Could have done better…



Layer (type)	Output Shape	Param #	Connected to
convolution3d_1 (Convolution3D)	(None, 32, 16, 46, 46)	4032	convolution3d_input_1[0][0]
dropout_1 (Dropout)	(None, 32, 16, 46, 46)	0	convolution3d_1[0][0]
convolution3d_2 (Convolution3D)	(None, 32, 12, 42, 42)	128032	dropout_1[0][0]
maxpooling3d_1 (MaxPooling3D)	(None, 32, 6, 21, 21)	0	convolution3d_2[0][0]
flatten_1 (Flatten)	(None, 84672)	0	maxpooling3d_1[0][0]
dense_1 (Dense)	(None, 512)	43352576	flatten_1[0][0]
dropout_2 (Dropout)	(None, 512)	0	dense_1[0][0]
dense_2 (Dense)	(None, 128)	65664	dropout_2[0][0]
dense_3 (Dense)	(None, 2)	258	dense_2[0][0]

Total params: 43550562

Further goals

- Expand dimensional representation
- Deeper ConvNet architectures
- Residual Network
- Convolutional layers + XGB (or other classifier)

Thank you!

References

1. https://www.kaggle.com/gzuidhof/data-science-bowl-2017/full-preprocessing-tutorial
2. https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet