Data Science Bowl project report

Akash Mehta, Shreya Pandit, Xide Xia, Xinye Chen (ASXX2017)

Problem definition:

The project statement: - to use a dataset of thousands of high-resolution lung scans provided by the National Cancer Institute and develop algorithms that accurately determine when lesions in the lungs are cancerous. After designing the algorithm, every team need to reduce the false rate that plagues the current detection technology and get patients earlier access to life-saving interventions, and give radiologists more time to spend with their patients.

Solution:

In our pursuit to learn and be as creative as possible, while maintaining a semblance of positive results, we devised multiple classifiers to try and tackle the problem, and see which ones result in the best solutions.

Finally, the classifier that proved to be the one with the maximum success rate, was the one used with the LUNA16 dataset

Final Result:

The final result has a two way interpretation. Firstly, classification of individual nodes, and classification of entire lungs.

Individual nodes: -

- Confusion matrix:

	NC	C
NC	7646	168
C	374	979

Entire Lung classification

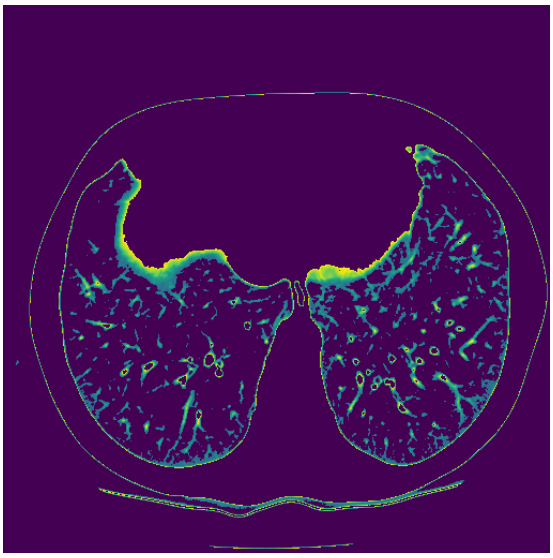
- Logloss of 0.76

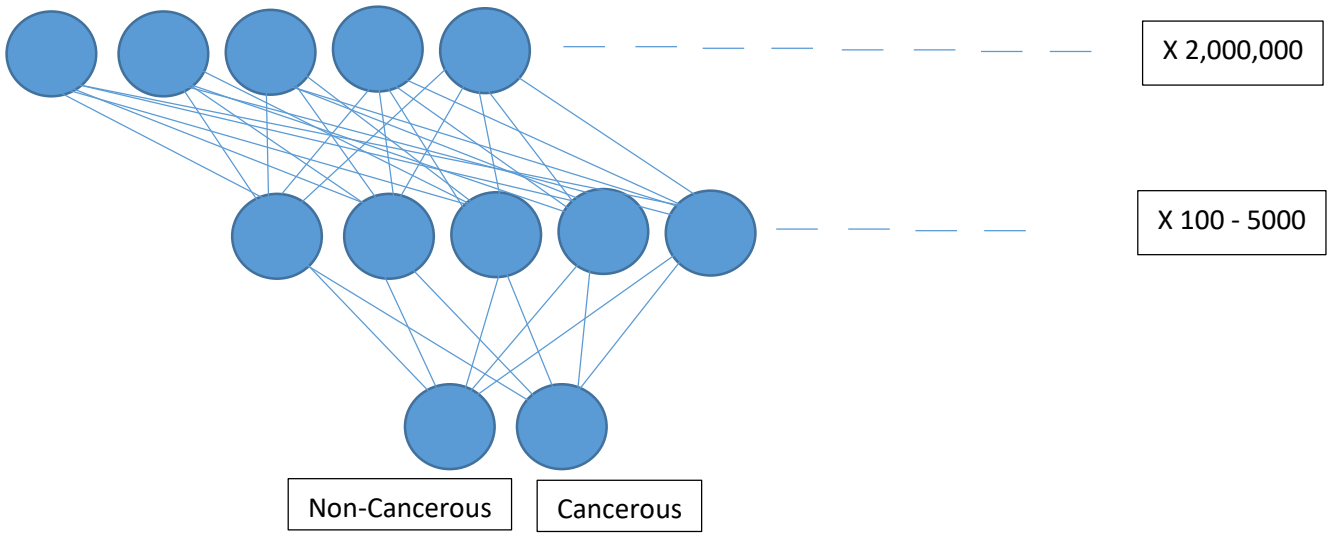
Classifiers:

Classifier 1

Procedure:

Take all the pixels that are relevant after preprocessing and removing bones, non-cancerous - flesh and blood, into a 2,000,000 long array for each patient. After resizing each slice to 200X200 and select about 50 slices





Theory:

One would expect that even though humans are unable to look at the big picture and instantly recognize a small part of it that is out of order, like looking at a page of text, and instantly recognizing a misspelled word, without having to go word by word, a computer might be capable of this. As such, with enough depth, the classifier could potentially recognize a cancerous node instantaneously.

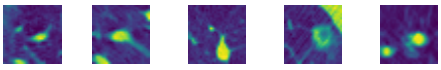
Problem:

- Very high memory requirement.
- Very slow per iteration.
- Very bad classifier as the number of variables were enormous that were not being taken into consideration. i.e. location, size, actual size of the images.
- Results were bordering 50% always, i.e. as good as guessing

Classifier 2

Procedure:

Take all the nodules of a patient into 30x30 images



Generate **5 randomly generated kernels** and pass each nodule for each patient into them. Find the max 5 values from the kernel output for each patient. Take these max 5 output nodules and send the kernel outputs to 5 seperate Neural Network. After a 100 - 1000 cycles of training, calculate the Confusion matrix for each kernel.

Kill of the worst performing Kernel and reinitialize a randomized kernel and continue training.

Continue this for 100 cycles, this should result in the death of 100 kernels.

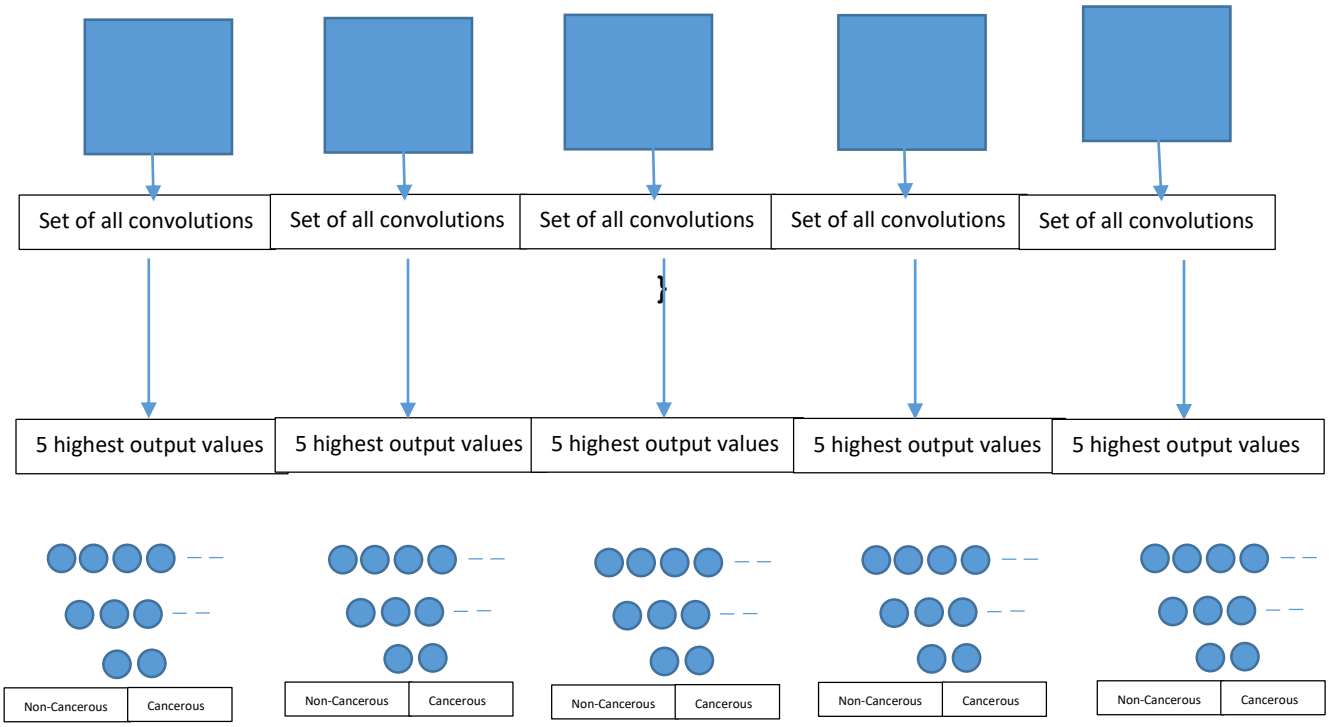
This would use the survival of the fittest to continuously weed out the kernels with the worst performance, and even if the final result is not the best possible outcome, it should result in an acceptable logloss.

For 100 Kill Cycles {

For 100 Training Cycles {

For Each Patient {

For each Nodule {



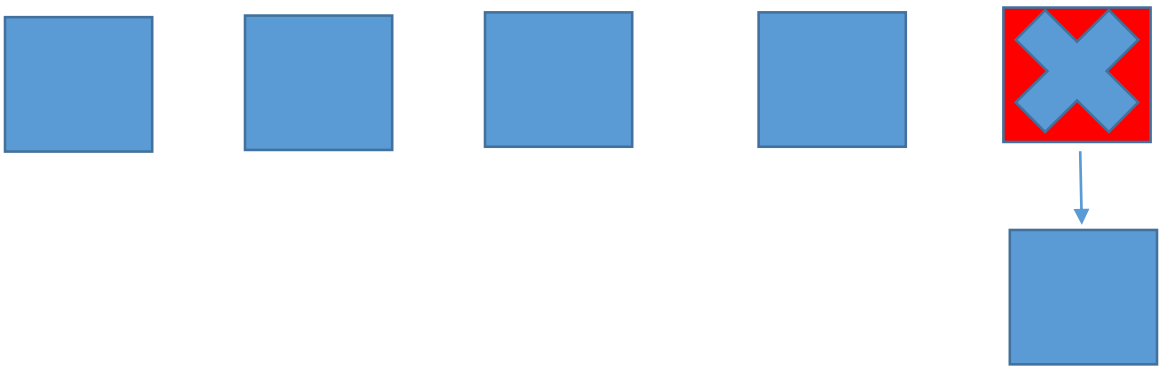
}

}

}

Calculate performance of each kernel to check for valid output

Kill worst performing kernel



Theory:

The basic logic behind this method was that through enough itterations, it might be possible to randomly generate a kernel that fits our requirements exactly. This method given enough time and itterations would have actually worked to give a generally valid output. This woud also be an ideal applicant for running on a GPU as each classifier is running in parallel. Another tangent to this idea is clamping evolution theory into the mix.

As per evolution theory, the further the generations, the better the results will become as survival of the fittest and random mutation occurs. Unfortunately, there were no resources or time to put that into action.

Problems:

- Very low probability of a valid kernel randomly being created.
- 100 itterations very low for a valid training cycle – Higher cycles made weeding out invalid kernels much clower.
- No way to train kernels as nodes are not labeled.

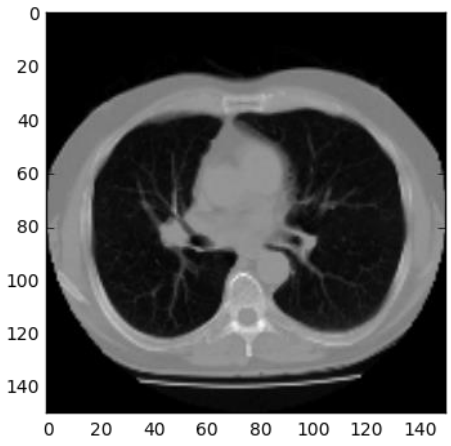
Classifier 3

Procedure:

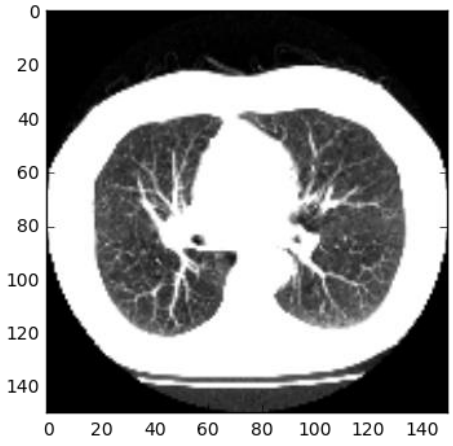
After pre-processing of the raw data, we get 20 slice images for each patient with size of 150*150. And then we do normalization and zero centering on these images.

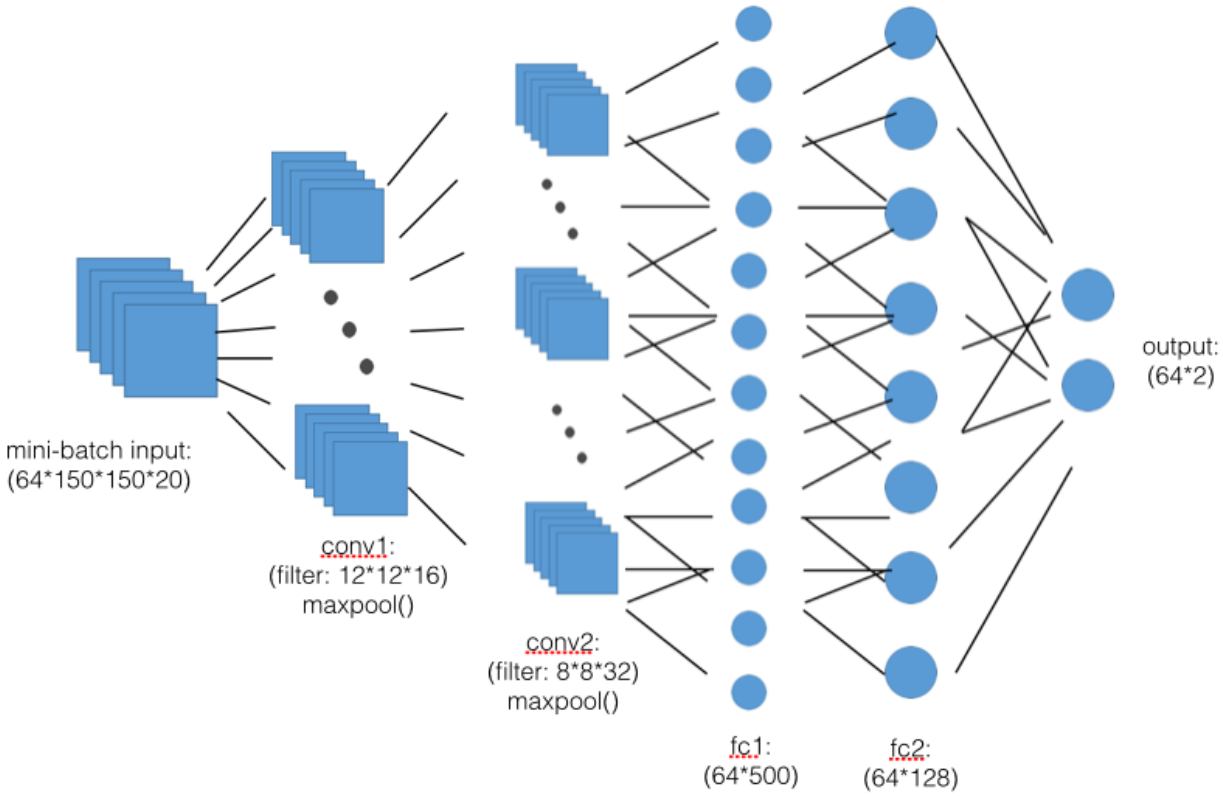
We use 3D-Conv Net train the data with mini-batch SGD optimization.

Original:



After Normalization and Zero center:





Theory:

The crest of today's machine learning algorithm revolve around deeper and deeper systems. Adding more kernels in the layers results in reduction of computational values for the neural nets, and training with multiple layers of Convolutional kernel results in better filtering of the images for the neural network.

Problems:

- Because this system required such high computation, sending all the slices of a lung scan was infeasible
- Only 20 slices per scan increased the chances of skipping the cancerous slice exponentially since each CT scan had anything from 200-500 slices.
- Thus this classifier worked more as a proof of concept rather than a working model

Classifier 4 – Final Submission

Procedure:

Use the LUNA16 dataset to extract all the cancerous nodules as they are labelled. Roughly - 1500

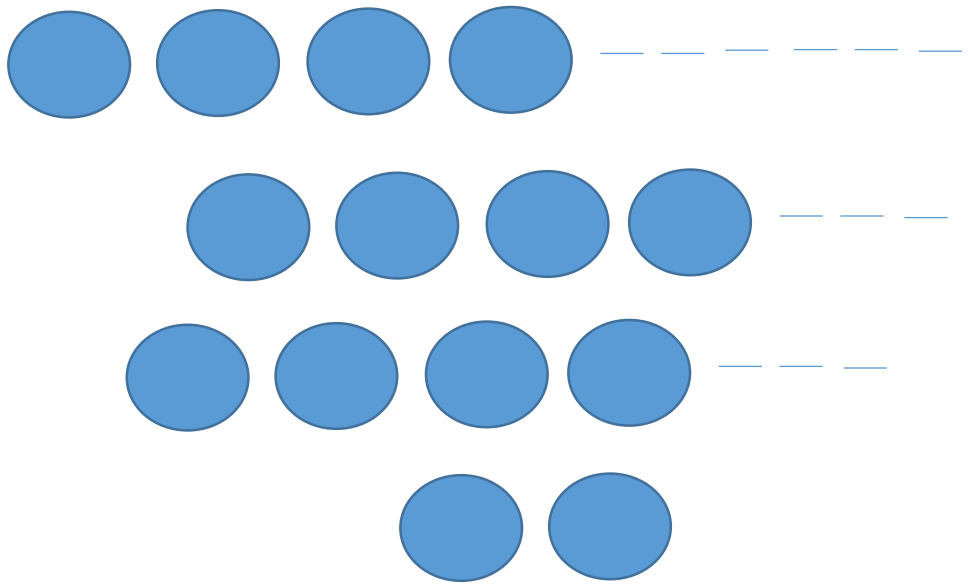
Use the Kaggle dataset to extract non-cancerous nodules. Roughly 8,500

Label this data and send to the classifier.

After training, classify each nodeule in the patients, then consider the node with the highest probability of cancer as the cancer probability for that patient.

For all patients {

For all nodules {



Experimenting with various combinations of Nodes and layers

}

}

Theory:

Training a classifier to differentiate each node seperately, if much more logically similar to a human way of thinking. Doctors, would usually look at scan slice by slice and carefully scrutanize each node to mentally generate the probabilities of each node being cancerous.

They would then assume that the node to most likely look like cancer holds the probability that the lung is cancerous.

Problems

- Even though the classifier is classifying the cancerous vs non-cancerous nodules with a low error, each patient has on an average 700 – 800 nodules, even if one nodule is mislabeled, the probability of a non-cancerous patient becomes cancerous.
- This does result in a low probability of a cancerous patient being labelled non-cancerous, even though it doesn't classify Cancer that well, but it behaves as a very low efficiency filter for non- cancerous as many come up with a cancerous probability.

Result:

- Logloss of 0.76 with 2 hidden layer and 1000 nodes in each with 30,000 iterations of training.
- Considering the average of the 10 highest cancer probable nodes.
- Purely considering node classification: -

Confusion matrix:	NC	C
NC	7646	168
C	374	979