Lung Nodule False Positive Reduction Challenge

Vismantas Dilys vismantas.d@gmail.com, Saad Masood saad.masood1@gmail.com,

Melissa Mozifian melissafm24@gmail.com, Andrew Murray

Edinburgh, Scotland

06/04/2016

Data Pre-processing and feature extraction:

As the main features for the algorithm we used three patches extracted at perpendicular axis, cantered around the candidate coordinate. We rescaled data so that voxel dimensions in each direction are 0.5mm and the extracted patches are 41x41 voxels size. The three patches were extracted in the axial, coronal and sagittal planes.

Because there were so few positive candidates, we used data augmentation, to generate more positive examples. This was done by using rotations around each of the axis of the axial, coronal and sagittal planes. For each axis we used 5 different angles -20, -10, 0, 10, 20 degrees. This gives 125 different examples per positive candidate. (Figure 1)

For negative candidates we did not use data augmentation. We took first 150 negative candidate locations for each volume from the candidates list and extracted patches aligned with the axial, sagittal and coronal planes.

The intensities in the images were clipped to -1000 - 2000HU range. Also when extracting patches, any parts lying outside the volume boundaries were padded with -1000.

The data pre-processing stage resulted in ~250000 training examples, with both positive and negative cases accounting for ~50% of the data.

Algorithm:

For the algorithm we used deep convolutional neural networks.

We used an ensemble model for which we trained two slightly different architectures of networks:

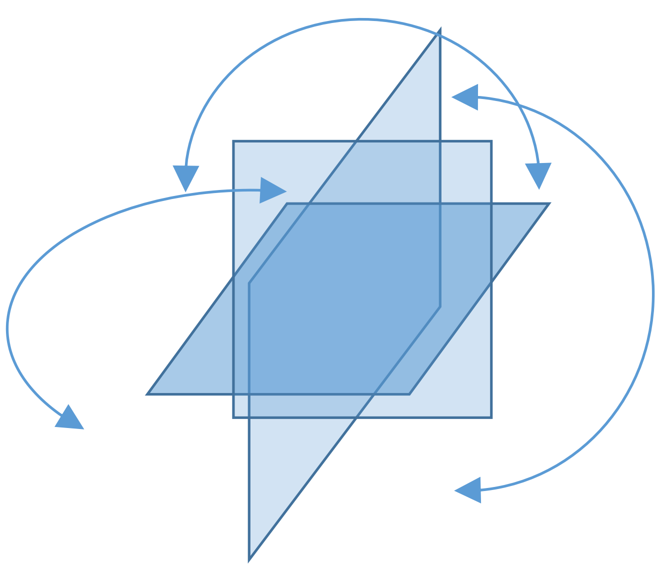

Figure 1 Patch extraction and augmentation

Model A:

First seven layers make up a network that is shared across the three patches – we apply exactly the same network on all three patches. The three resulting outputs are then concatenated and passed on to the 8th layer.

	Layer	Shape	Comments
1	2D conv	36x7x7	Activation: leaky ReLU with alpha 0.02
2	2D conv	48x5x5	Activation: leaky ReLU with alpha 0.02
3	2D conv	64x4x4	Activation: leaky ReLU with alpha 0.02 Step sizes 2,2
4	2D conv	96x4x4	Activation: leaky ReLU with alpha 0.02
5	2D conv	128x3x3	Activation: leaky ReLU with alpha 0.02
6	2D conv	156x3x3	Activation: leaky ReLU with alpha 0.02 Step sizes 2,2
7	2D conv	196x3x3	Activation: leaky ReLU with alpha 0.02
8	2D conv	512x3x3	Activation: leaky ReLU with alpha 0.02
9	Dropout		P=0.5
10	Dense	256	Activation: leaky ReLU with alpha 0.02
11	Dropout		P=0.5
12	Dense	128	Activation: leaky ReLU with alpha 0.02
13	Dropout		P=0.5
14	Dense	1	

Model B:

	Layer	Shape	Comments
1	2D conv	36x7x7	Activation: leaky ReLU with alpha 0.02
2	2D conv	48x5x5	Activation: leaky ReLU with alpha 0.02
3	2D conv	64x4x4	Activation: leaky ReLU with alpha 0.02 Step sizes 2,2
4	2D conv	64x4x4	Activation: leaky ReLU with alpha 0.02
5	2D conv	76x3x3	Activation: leaky ReLU with alpha 0.02
6	2D conv	76x3x3	Activation: leaky ReLU with alpha 0.02 Step sizes 2,2
7	2D conv	84x3x3	Activation: leaky ReLU with alpha 0.02
8	2D conv	256x3x3	Activation: leaky ReLU with alpha 0.02
9	Dropout		P=0.5
10	Dense	256	Activation: leaky ReLU with alpha 0.02
11	Dropout		P=0.5
12	Dense	128	Activation: leaky ReLU with alpha 0.02

13	Dropout		P=0.5
14	Dense	1	

First seven layers make up a network that is shared across the three patches – we apply exactly the same network on all three patches. The three resulting outputs are then concatenated and passed on to the 8th layer.

Both models were trained using stochastic gradient descent with batch size of 32, momentum 0.9, decay 10^-6. Learning rate for each epoch was scheduled as 0.003*0.096^epoch_number. We used binary cross-entropy loss function.

The prediction results of the two models we averages to produce the final prediction

We used 10 fold cross validation as was specified by the rules of the challenge.