Winning Model Documentation

Name: Elias Vansteenkiste, Matthias Freiberger, Andreas Verleysen, Iryna Korshunova, Lionel Pigou, Frederic Godin, Jonas Degrave

Location: Belgium

Competition: data-science-bowl-2017

1. Background team

If part of a team, please answer these questions for each team member. For larger teams (3+), please give shorter responses.

- What your academic/professional background?
  Elias: Postdoctoral researcher
  Fréderic: Ph.D. student NLP/Deep Learning
  Matthias: Ph.D. student Neuromorphic computing
  Andreas: Ph.D. student deep reinforcement learning in robotics
  Iryna: PhD student
  Lionel: PhD student
- Did you have any prior experience that helped you succeed in this competition?
  Elias: applying deep learning to raw sensor signals
  Matthias: computer vision training
  Andreas: machine learning master dissertation + computer vision projects
  Iryna: previous Kaggle competitions
  Lionel: previous Kaggle competition
  Fréderic: deep learning research for NLP

- What made you decide to enter this competition?
  We were looking for a challenge to participate in to share (deep learning) knowledge and we liked this one particularly because of the noble goal.
- How much time did you spend on the competition?
  Every team member spent a different amount of time on the competition, depending on his/her other research activities, ranging from a couple of weeks up to the full competition length.
- If part of a team, how did you decide to team up?
  We are all part of the same lab and it is becoming a tradition to participate in the annual data science bowl.
- If you competed as part of a team, who did what?
  Elias: designing false positive reduction networks for nodules, lung segmentation, region of interest extraction, nodule property prediction, final cancer prediction classification, inception-resnet v2 architecture
  Iryna: framework, nodule segmentation model
  Fréderic: Lung segmentation, transfer learning approaches, nodule malignancy prediction networks.
  Matthias: data analysis/augmentation, prob. modelling, external data, model exploration
  Andreas: ensembling
  Lionel: ROI extraction, some network architecture, hardware/software support, 3d affine transformations

2. Summary

4-6 sentences summarizing the most important aspects of your model and analysis, such as:

- The training method(s) you used (Convolutional Neural Network, XGBoost)
  We used convolutional networks in the different steps of our approach, but step 2 (see next)
- The most important features
  Predictions are made based on features built during several steps:
  1. Nodule segmentation
  2. Region Of Interest (ROI) extraction: blob detection to find the centers of the nodule
  3. False positive reduction for the nodules: ROI's are ranked following the prediction of this network
  4. Final Cancer prediction: a network trained starting from transferred weights from a malignancy prediction network with as input
- The tool(s) you used
  We built a framework built on top of lasagne and theano.
- How long it takes to train your model
  It takes a few days to train our models and make predictions. The exact amount of time depends on the number and the speed of the available GPUs.

3. Features Selection / Engineering

- What were the most important features? *
  Region of interests (candidate nodules)
  Malignancy prediction network
- How did you select features?
  Trial and error
- Did you make any important feature transformations?
  We trained the networks so we don't have the specific features
- Did you find any interesting interactions between features?
  Not really

- Did you use external data?
  LUNA dataset
  SPIE/AAPM data has been evaluated, but it's incorporation did not result in measureable improvements

4. Training Method(s)

- What training methods did you use?
- Did you ensemble the models? Yes
- If you did ensemble, how did you weight the different models?

Our ensemble merges the predictions of our last stage models. Since Kaggle allowed two submissions, we used two ensembling methods:

1. **Defensive ensemble**: Average the predictions using weights optimized on our internal validation set. The recurring theme we saw during this process was the high reduction of the number of models used in the ensemble. This is caused by the high similarity between the models. It turned out that for our final submission, only one model was selected.
2. **Aggressive ensemble**: Cross-validation is used to select the high-scoring models that will be blended uniformly. The models used in this ensemble are trained on all the data, hence the name 'aggressive ensemble'. We uniformly blend these 'good' models to avoid the risk of ending up with an ensemble with very few models because of the high pruning factor during weight optimization. It also reduces the impact of an overfitted model. Reoptimizing the ensemble per test patient by removing models that disagree strongly with the ensemble was not very effective because many models get pruned anyway during the optimization. Another approach to select final ensemble weights was to average the weights that were chosen during CV. This didn't improve our performance. We also tried stacking the predictions using tree models but because of

the lack of meta-features, it didn’t perform competitively and decreased the stability of the ensemble.

5. Interesting findings

- What was the most important trick you used?

We applied several tricks that helped us progress. One interesting trick was the use of transfer learning to initialize the weights of our final networks with weights learned on other tasks on the LUNA dataset. Another interesting trick was the use of a LogMeanExp function to join the predictions of the individual ROIs, which acted as a “soft” max function.

- What do you think set you apart from others in the competition?

Our solution is a full deep learning approach, mainly consisting of networks trained on CT scans, rather than focussing on specific features, edge cases or a combination thereof.

- Did you find any interesting relationships in the data that don't fit in the sections above?

No

6. Simple Features and Methods

Many customers are happy to trade off model performance for simplicity. With this in mind:

- Is there a subset of features that would get 90-95% of your final performance? Which features? *

Our algorithm ranks the ROIs of which our final models typically use the highest ranked 8-12 ROIs. Our calculations on the LUNA showed that the all nodules of interest are covered when using 8-12 ROIs. Using only the 4 highest ranked ROIs already covered roughly 90% of LUNA nodules of interest. Hence, we can expect that only using 4 top-ranked ROIs per patient should yield very good results.

- What model that was most important?

Based on the individual scores, the most important model for the final prediction was a model which was trained using pretrained weights of the a LUNA (LIDC-IDRI) malignancy prediction network. The individual predictions of that model were joined using a LogMeanExp function.

Appendix

This section is for a technical audience who are trying to run your solution. Please make sure your code is well commented.

A1. Model Execution Time

Many customers care about how long the winning models take to train and generate predictions:

- What software did you use for training and prediction?
  Theano, Lasagne, Scikit-image

- What hardware (CPUS spec, number of CPU cores, memory)?
  7 machines
  GPU: GTX 1080, GTX 980, Titan X and Tesla K40
  Mem: 32GB to 64GB per machine
  CPU: mostly 6 cores, 3GHz to 4GHz
- How long does it take to train your model?
  4 days
- How long does it take to generate predictions using your model?
  3 days
- How long does it take to train the simplified model (referenced in section 4)?
  3 days
- How long does it take to generate predictions from the simplified model?
  2 days

A2. Dependencies

List of all dependencies including:

- programming language/statistical tool
  Python 2.7
- libraries or packages
  Theano 0.9.0b1, Lasagne 0.2.dev1, Scikit-Image 0.12.3, SciPy 0.18.1, Numpy 1.12.0, Scikit-Learn 0.18.1
  Cuda 8.0, CuDNN 5.1
- operating system
  Ubuntu 16.04 LTS
- another other software used to generate your solution.

A3. How To Generate the Solution

See readme file included in the submission of the code

A4. References

Citations to references, websites, blog posts, and external sources of information where appropriate.