A solution to Owkin's data challenge

Wissam KARROUCHA

June 2023

1 Introduction

The goal of this data challenge, proposed by the French healthcare AI unicorn Owkin, is to develop an automated method to detect PIK3CA mutation in breast cancer. This mutation is observed in 30 to 40% of breast cancers. The patients having this mutation are receptive to a class of targeted therapies called PI3Ka inhibitor. It is therefore of great clinical interest to detect PIK3CA mutation.

The current gold standard for PIK3CA mutation detection is the DNA sequencing. However, it requires a technical and bioinformatics expertise which is not affordable for all laboratories. It would therefore be of great interest to develop an automated solution to detect PIK3CA mutation. The goal of the data challenge is to develop a model predicting whether a patient has a PIK3CA mutation from her digitalized histology slides.

2 Dataset

Since histology slides are of very high dimension (more than 100 000 x 100 000 pixels), 1000 smaller tiles (224 x 244) have been extracted from each of these slides. These have been randomly picked from the slide. Each tile contains some tissue.

From each tile of each slide, 2048 features have been extracted by the organizers of the challenge with a Wide ResNet-50-w2 pretrained on TCGA-COAD, a large histology dataset. This enables us to train our models on these 2048 extracted features rather than directly on the images, therefore to save computational cost and train our models on our computers.

The available dataset contains, for 344 slides extracted from 305 patients:

- the label of the patient (does she have the PIK3CA mutation ?)
- for each of the 1000 tiles of the slide:
  - the 2048 features extracted from this tile
  - the zoom level of the tile

– the coordinates of the tile within the slide

The training data come from three different medical centers and the test set comes from a fourth medical center. There is data variability from one medical center to another (for instance, the colour palettes of the images aren't the same).

3 Methodology

3.1 A weakly supervised learning problem

This problem is a binary classification problem since the goal is to output a binary label (with or without mutation). It is weakly supervised since we don't have a label for each tile, but only one label per slide. Therefore, we can see each slide as a bag of tiles and we just have a label per bag and not a label per slide. We know if the mutation is present in the whole slide, but not whether it is present in a single tile of the slide.

3.2 Data preprocessing

I normalized the data using the Z-Score method. Instead of normalizing all patients' tiles together, I performed a separate Z-Score normalization for each medical center. This aims at accounting for the variability of the data and the value ranges of the features from one medical center to another.

I allso tried to do normalization by zoom level in order to account for the difference between the zoom levels. However, It didn't improve the model's results.

3.2.1 Tackling class imbalance

I also tried approaches in order to counter class-imbalance. Indeed, whereas the dataset contains 344 slides of patients without PIK3CA mutation, it only contains 128 slides of patients with PIK3CA mutation.

The first approach was hard negative mining, that is to say adding to the dataset new negative samples which are chosen among the false positive examples. By doing so, the model can learn to better distinguish between positive and negative examples, as it is forced to learn more discriminative features that can help it differentiate between the two.

A further approach consists in using a weighted loss : for each sample belonging to a certain class (with mutation or without mutation), I divide the loss function by the total number of samples belonging to the same class. That enables to give more weight to samples belonging to the under-represented class.

3.2.2 Tackling medical center heterogeneity

In order to tackle medical center hetorogeneity, I tried to use a weighted loss : for each sample belonging to a certain medical center, I divide the loss function by

the total number of samples belonging to the same medical center. That enables to give more weight to samples belonging to the under-represented medical centers. However, it didn't improve the model's results.

I also tried to constitute batches which are well-balanced between the different medical centers (by doing some oversampling or undersampling because we don't have exactly the same number of samples for each medical center).

3.3 Architectures

Since this problem is a weakly supervised learning problem, I decided to use a multiple-instance learning approach. I got inspired by [1] and [2]. In [1], only the minimum and maximum scores are kept and combined into a final prediction. DeepMIL model in [2] uses a weighted sum of the tile features using the tile attention scores and computes the prediction using that average representation.

More specifically, DeepMIL uses a gated-attention layer. I decided to use gated attention layer in my work too. The interest of such a layer is that it enables to learn from the features of a tile whether this tile is important for determining the label of the slide it is part of. This can be of great help since it enables the model to know which tiles it should pay attention to and which it shouldn't. Indeed, we can imagine that certain shapes of tiles, geometries or dispositions make them not informative whereas some others are more.

$$a_k = \frac{\exp\{\mathbf{w}^\top \left(\tanh\left(\mathbf{V}\mathbf{h}_k^\top\right) \odot \mathrm{sigm}\left(\mathbf{U}\mathbf{h}_k^\top\right)\right)\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^\top \left(\tanh\left(\mathbf{V}\mathbf{h}_j^\top\right) \odot \mathrm{sigm}\left(\mathbf{U}\mathbf{h}_j^\top\right)\right)\}}$$

Figure 1: Mathematical formula of the Gated attention layer

I trained the models with and without dropout and when training with dropout, I made the dropout rate vary between 0.2 and 0.5.

3.4 Optimization

I tested two optimizers: the Stochastic gradient descent (SGD) and Adam. I tried each of these optimizers with a wide range of learning rates, from 10^{-1} to 10^{-6} and with different numbers of epochs.

I tested various batch sizes ranging from 1 to 64. I generally used smaller batch sizes using SGD than Adam.

The loss function which is being used is the binary cross-entropy which is a classical loss function for classification tasks.

3.5 Adversarial domain adaptation module

In order to tackle the data variability among medical centers and because the normalization by medical center isn't sufficient to make up for this variability, I decided to add an adversarial domain adaptation module. I tested models with and without this additional module.

The principle is the following. I add inside the model a module introduced by a gradient reversal layer that is trained to predict the medical center of the slide being processed by the model. There is an adversarial training: whereas this module is trained to predict the most accurately the medical centre the patient comes from, the previous layers of the model are trained to fool this classification module. The goal of this principle is to enforce the model to learn features that are as medical center-invariant as possible. It is further detailed in papers such as [3] and [4] and in the schema below.

In this framework, the final loss function is a linear combination of the loss of the global classifier (how good it is at predicting whether a cell has a mutation), the loss of the mdeical centre classifier (how good it is at discriminating among medical centres) and the loss of the part of the model until the penultimate layer (how bad is the medical centre classifier at discriminating among medical centres).

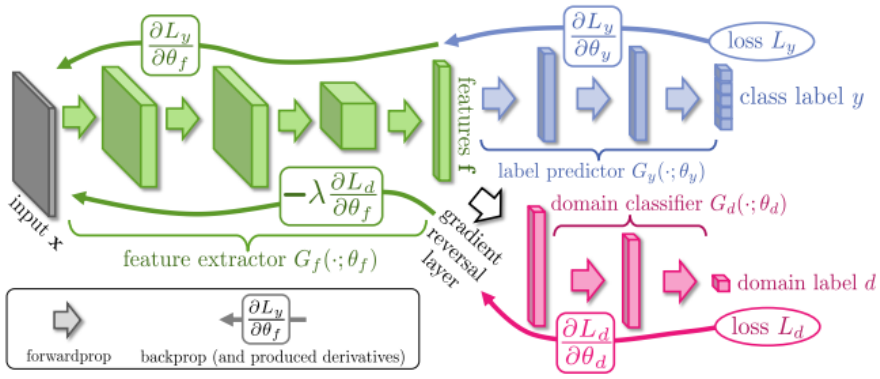

Figure 2: Architecture of a deep neural network with domain adaptation module

3.6 Other approaches tested

I also tried other approaches even if I didn't use them in my final model since they didn't manage to improve the performance of the model when I tested them.

One approach is to take the zoom level of the microscope as input of the model, as if it was an additional feature.

Another approach I tried was to do undersampling and oversampling in order to have exactly the same number of positive and negative samples in my dataset.

3.7 Evaluation metrics

The evaluation metrics which is used is the ROC AUC.

3.8 Validation method

I chose to use a three-fold cross-validation using each medical center as a fold. This aims at assessing the ability of the model to generalize to medical centers that were unseen during the training.

4 Results

I obtained the best results in terms of mean cross-validation ROC AUC with the model having the following architecture AND a domain adaptation module (see domain adaptation subsection in Section 3):

Layer	Layer type	Number of units	Activation
1	Fully-connected	256	Tanh
2	Gated Attention	256	Tanh
3	Fully-connected	128	Tanh
4	Fully-connected	64	Tanh
5	Fully-connected	1	Sigmoid

Table 1: Architecture of the model giving the highest validation ROC AUC

The best results were obtained by training the above described model with Adam optimizer using a batch size of 8, a learning rate of 10^{-5} during 24 epochs.

Below is a schema of this model:

Below are the results of this model on each of the cross-validation folds, both in terms of performance metrics (ROC AUC in this case) and in terms of loss function.

C1 means the model which has been trained on medical centres C2 and C3 and is validated on C1. C2 means the model which has been trained on medical centres C1 and C3 and is validated on C2. C3 means the model which has been trained on medical centres C1 and C2 and is validated on C3.

We chose to train the model for 24 epochs since it is the number of epochs yielding the maximal ROC AUC before the model starts over-fitting (which can be seen because the los starts increasing on two folds).

This model reaches the following performance metrics: 64.55% validation ROC AUC on fold 1, 71.34% validation ROC AUC on fold 2, 69.69 % validation ROC AUC on fold 3 and 69.23% ROC AUC on the test set. This enabled me to be ranked at the 3rd position of the challenge.

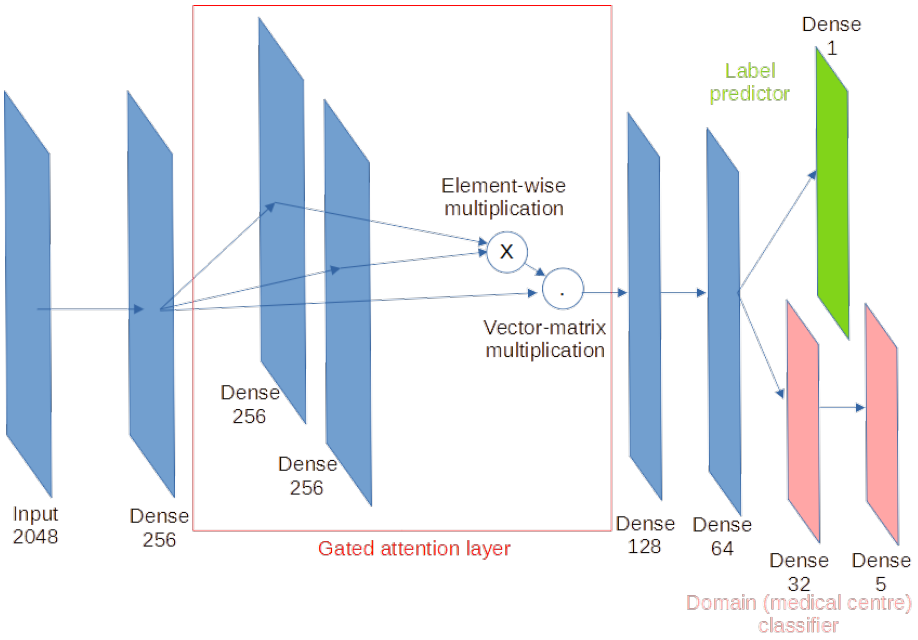

Figure 3: Schema of the best performing model

5 Conclusion

The problem of PIK3CA mutation detection using anatomopathology slides is challenging for at least two reasons. First, it is a weakly supervised problem therefore it is necessary to find an efficient way to use the slide-level information we have even if it isn't as precise as a tile-level information. Second, the data come from heterogeneous medical centers which all have their own imaging standards which raises the issue of domain adpatation.

I proposed to leverage deep multiple-instance learning with attention layers in order to tackle the weakly supervision problem and adversarial domain adaptation in order to tackle the heterogeneity of data stemming from different medical centers. Combining these two approaches enabled me to reach a 69.23% ROC AUC on the test set.

Further research might entail finding more efficient ways to tackle class imbalance between positive and negative samples. Another idea I had would be to develop a model taking into account the relative position of the different tiles of a slide. This could for instance be done by using a Graph neural network. The features of a tile would be a vertex of the graph and the edges of the graph would be the distances between the tiles (which can be inferred from the tiles' coordinates).

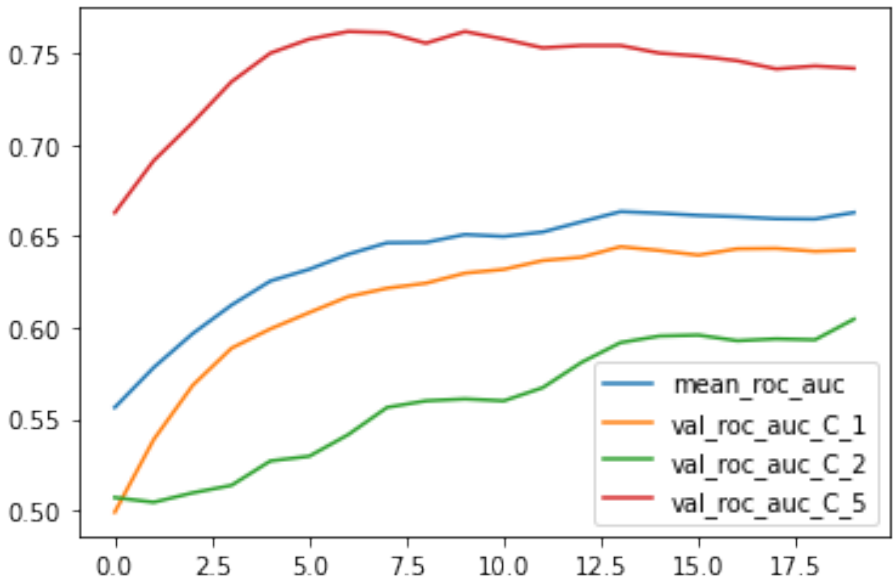

Figure 4: Learning curves representing validation ROC AUC on each cross-validtion fold

References

[1] Courtiol, P., Tramel, E. W., Sanselme, M., & Wainrib, G. (2018). Classification and disease localization in histopathology using only global labels: A weakly-supervised approach. arXiv preprint arXiv:1802.02212

[2] Ilse, M., Tomczak, J., & Welling, M. (2018, July). Attention-based deep multiple instance learning. In International conference on machine learning (pp. 2127-2136). PMLR.

[3] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

[4] Ganin, Y., & Lempitsky, V. (2015, June). Unsupervised domain adaptation by backpropagation. In International conference on machine learning (pp. 1180-1189). PMLR.

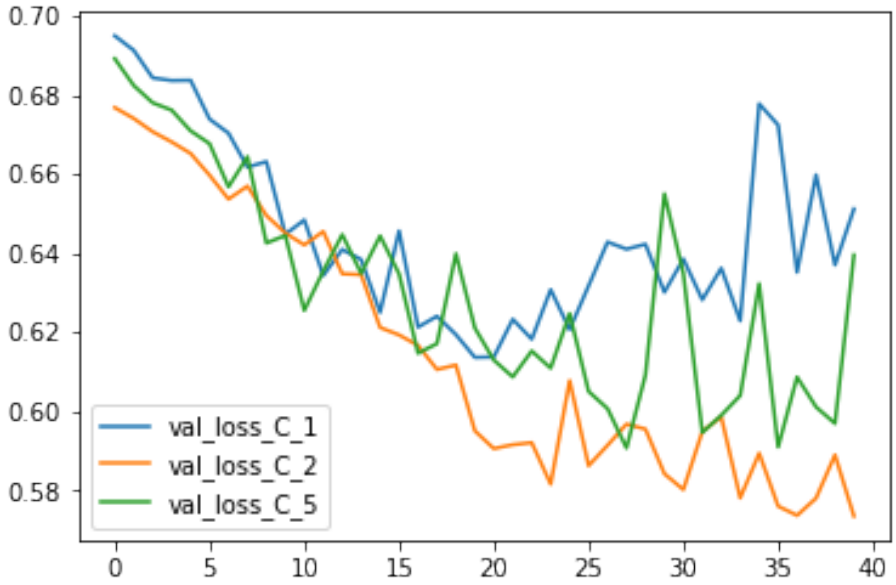

Figure 5: Learning curves representing validation loss on each cross-validtion fold