**Tissue Slide Images Classification**

Lior Mayorov

I chose to submit a python notebook since I think it’s the best way to present flow of work, analysis and conclusions.

**Step 1: Create your dataset**

I download 10 digital pathology slides from the suggested website, using the filter to choose 7 “normal” slides and 3 “tumor” slide – that will be our target. I downloaded four more slides (two - normal and two - tumor) for evaluation. Which is still very little data for insights.

“open\_slide\_as\_array” opens the svs file, reduces size and converts to np array.

“create\_csv\_w\_filenames” creates the dataset of file names and their target.

I then displayed the images.

Important note: the training data is the tiles, and the target is the **whole tissue’s** **tag**. I don’t think this is the best way to do it, but it is the simplest. In the end there are suggestions for improvements.

“create\_tiles\_dataset” goes through every image and:

* divides it to square tiles of the same size TILES\_SIZE=32
* drops tiles with more than 95% of over 245 pixel value (supposedly the background, should check it more thoroughly with visualization)
* counts amounts of tiles per image for later evaluation

**Step 2: Re-train ResNet50**

I imported ResNet50 with imagenet pretrained weights, without the final layer, since the original task wasn’t binary classification like ours.

I froze all layers and then added a pooling layer, a fully-connected layer and a final sigmoid layer for a single value prediction (binary). Those are the only layers to be trained, all the other weights will stay the same and save us training time.

I used the standard Adam optimizer with all the default values, used the binary cross entropy loss (BCE) and fine tuned the net for 10 epochs.

**Step 3: Present and analyze the results**

I plotted the losses, both train and evaluation throughout the training. You can see the training loss decreases and the validation loss increases – seems like a classic case of overfitting, the net most likely memorizes the training images.

However, surprisingly the model does classify the validation data pretty well, only got one “normal” slide wrong out of the four.

I plotted the predictions histogram to make sure there isn’t a strong bias, and it seemed to be fine, maybe a bit biased torwards 1, which makes sense given 70% tumor data..

How to improve results:

1. Change the target / loss function. The model can't seem to understand what inidicates that a tile containes a tumor which makes perfect sense, since most of the in a tumorous slide image don't contain a tumor, but they are all tagged as they contain a tumor, since we don't have better tags (segmentation). A few possible fixes:
   1. change loss function to use BCE on the mean of all tiles of the same slide image to that image's target
   2. change data to contain only image data and divide into tiles inside the net
   3. change the target to contain only images tags and average the last layer by tiles and compare
2. ADD DATA. and GPUs :)
3. Add data augmentations to increase dataset - rotations, movements, color manipulations …
4. Process data: grey-scale, HSV dimensions, protain-nuclei dimensions
5. Hyper parameter tuning - change loss, optimizer, lr, epochs...
6. Multi-modality model - add more data on the person in addition to the tissue slide image (age, smoking...)
7. Split data into tissue types and train seperately. I saw some very different-looking tissues, the model might have difficulty generalizing to all kinds and have more success training and predicting for each tissue type alone.