citations

This study developed a novel deep multiple instance learning model predicting tumor purity from H&E stained digital histopathology slides. Also, the researcher obtained spatially resolved tumor purity maps and showed that tumor purity varies spatially within a sample. An accurate tumor purity estimation is crucial for accurate pathologic evaluation. The novel MIL model can bring me the inspiration of machine learning.

1 predicting tumor purity

1. 1 Implications for Biology

- The model can be utilized for high throughput sample selection for genomic analysis
- The MIL models successfully predicted tumor purity from histopathology slides in different TCGA cohorts
- The MIL model successfully classifies samples into tumor vs. normal
- The MIL model's predictions gave a higher correlation and lower mean-absolute-error with genomic tumor purity values
- tumor purity map analysis reveals the probable cause of pathologists' high percent tumor nuclei estimates

1.2 Research Method

- The model accepts a bag of patches cropped from the top and bottom slides of a sample as input and predicts the sample's tumor purity at its output. The *MIL pooling filter* summarizes extracted features into a bag-level representation by estimating marginal feature distributions.
- Researchers obtain a spatial tumor purity map for a slide by inferring tumor purity over each $1mm2$ region of interest within the slide in a sliding window fashion.
- The MIL model learned discriminant features for cancerous vs. normal histology from sample-level genomic tumor purity labels . Researchers used discriminant features to obtain cancerous vs. normal segmentation maps for tumor slides. Trained *feature extractor* module extracts features of patches from tumor and normal slides of a patient. Then, segmentation maps are obtained by hierarchical clustering over the extracted feature vectors.

1.3 Training of Multiple Instance Learning (MIL) Models

Reseachers trained the MIL model on samples of patients in the training set. They used a bag of patches from slides of the tumor sample as the input and tumor purity value obtained from genomic sequencing data by ABSOLUTE as the ground-truth label Genomic tumor purity values were extracted from publicly available data in the Genomic Data Commons.

Researchers initialized the neural networks randomly and trained them end-to-end. They trained models using the ADAM optimizer with a learning rate of lr = 0.0001 and $L2$ regularization on the weights with a weight decay of *weight_decay* = 0.0005. The batch size was 1. They used absolute error as the loss function and employed early-stopping based on loss in the validation set to avoid overfitting. Then, we evaluated our model on the unseen test set.

2 Practice

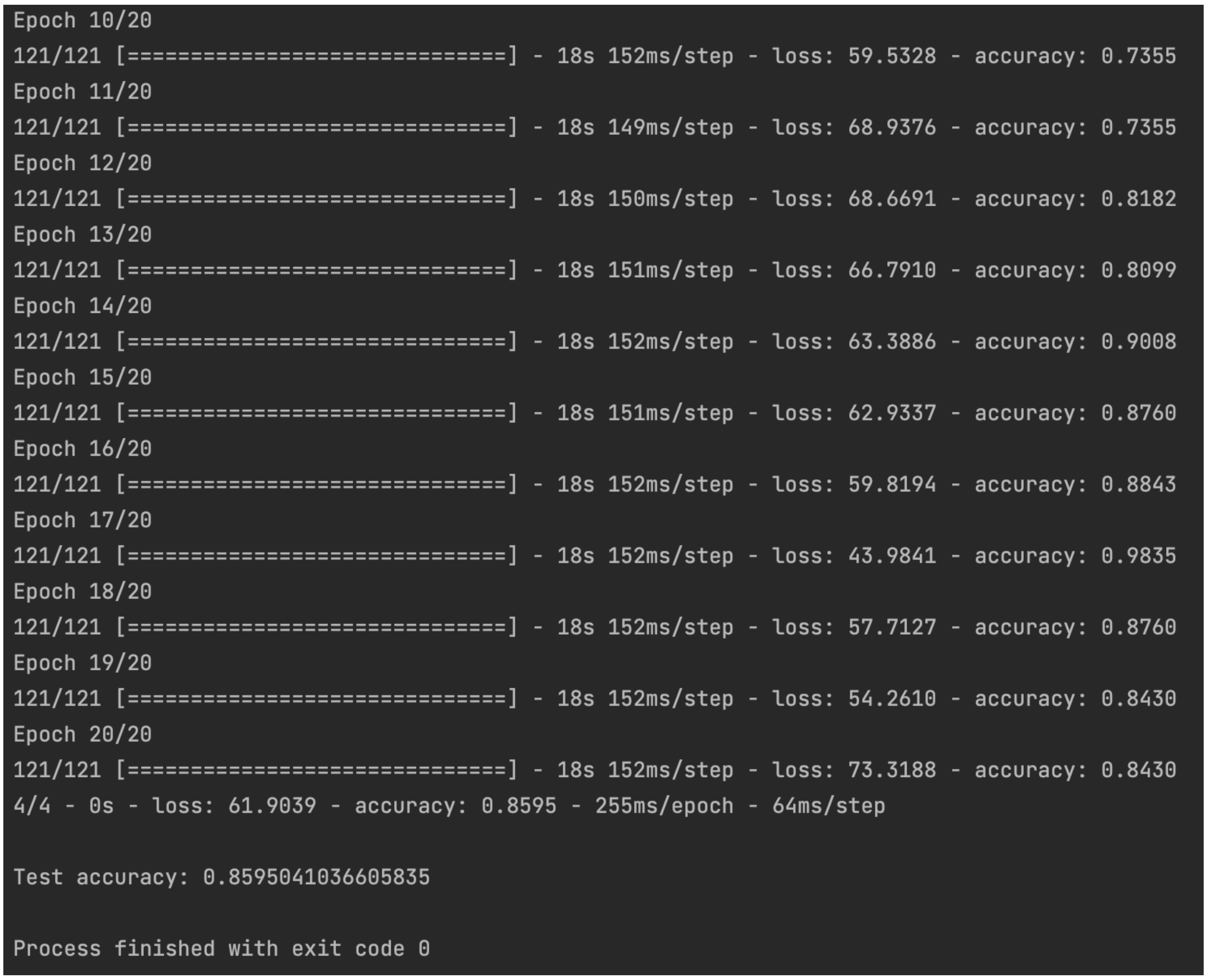
2.1 Implementation Code Main Process

- Induce and deal with data
    - Induce MNIST data set and due with it. Set Each bag consists of 100 images with a fraction x of digit 0 and 1-x of digit 7. Set sum of 0 as tag of the bag. The bags make up the x_train_toy and the tags make up the y_train_toy. Treating 0 as a cancer cell and each graph as a slice, predicting the number of occurrences of 0 is similar to predicting the density of cancer cells.
- Build up the model
    - Convert bags from a three-dimensional array (100 x 28 x 28) to a one-dimensional array. Treat this layer as an unstacked row of pixels in the image and arrange them. This layer has no parameters to learn, it just reformats the data.
    - After flattening the package, the network will include the sequence of four TF.keras. Layers.Dense layer. These are densely connected or fully connected neural layers.
    - **The first Dense layer** has 256 nodes (or neurons) where L2 regularization is set. **The second layer** sets Dropout regularization to make the network less sensitive to changes in the weight of a neuron, increasing generalization and reducing overfitting.**The third Dense layer** has 128 nodes (or neurons). **The fourth (and final) layer** returns a logits array of length 100. Each node contains a score that indicates how many zeros there are in the current package.
- Train neural network using the neural network architecture specified in the given paper
    - Train models using the ADAM optimizer with a learning rate of *lr* = 0.0001 and *L*2 regularization on the weights with a weight decay of *weight_decay* = 0.0005. The batch size was 1. Use absolute error as the loss function and employed early-stopping based on loss in the validation set to avoid overfitting.
- Fit model with training data
    - The model is "fitted" to the training data, and during the model training, loss and accuracy indicators are displayed. After the model is trained, it is used to predict the test package and output the test accuracy.
- Visualization
    - Finally, the results are visualized. I provide an interface for predicting different bags in the test set using matplotlib. You can set the certain bag for visualization.
    - The visualization results will include all the digital images in the bag and the bar chart for predicting the content of 0 in the bag. The result with the highest probability of prediction is shown in blue if it is correct, and red otherwise.

2.2 Restult

More results are uploaded to the Github directory as files

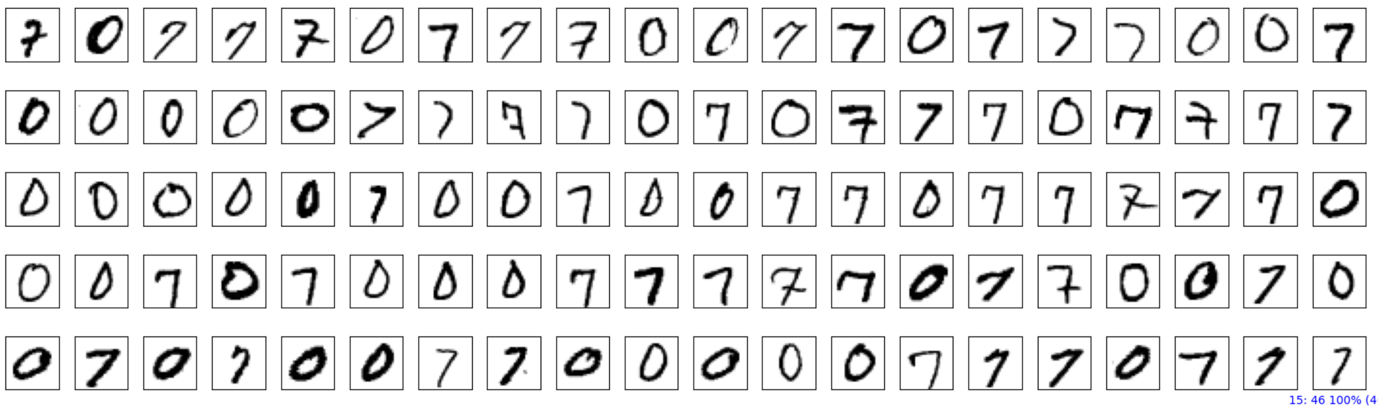
- Training process and accuracy(Omit the first ten epochs)

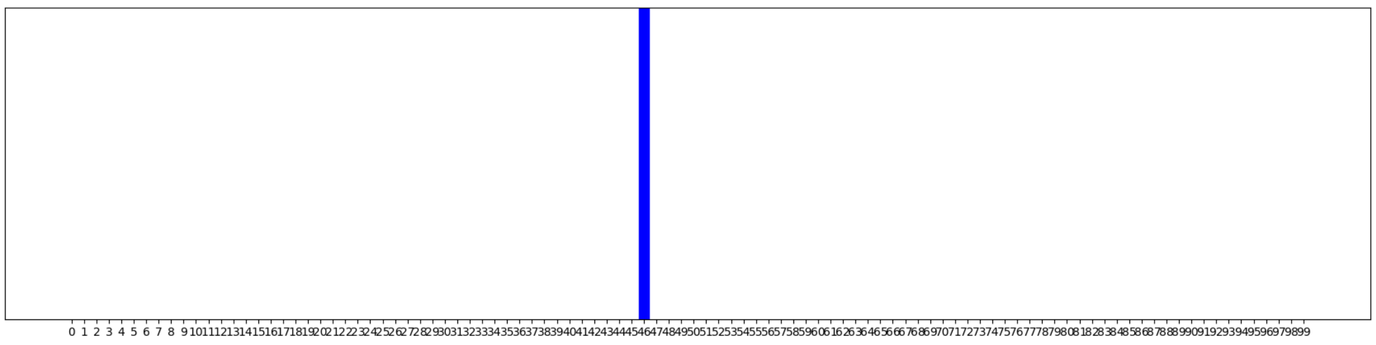

```
Epoch 10/20
121/121 [==============================] - 18s 152ms/step - loss: 59.5328 - accuracy: 0.7355
Epoch 11/20
121/121 [==============================] - 18s 149ms/step - loss: 68.9376 - accuracy: 0.7355
Epoch 12/20
121/121 [==============================] - 18s 150ms/step - loss: 68.6691 - accuracy: 0.8182
Epoch 13/20
121/121 [==============================] - 18s 151ms/step - loss: 66.7910 - accuracy: 0.8099
Epoch 14/20
121/121 [==============================] - 18s 152ms/step - loss: 63.3886 - accuracy: 0.9008
Epoch 15/20
121/121 [==============================] - 18s 151ms/step - loss: 62.9337 - accuracy: 0.8760
Epoch 16/20
121/121 [==============================] - 18s 152ms/step - loss: 59.8194 - accuracy: 0.8843
Epoch 17/20
121/121 [==============================] - 18s 152ms/step - loss: 43.9841 - accuracy: 0.9835
Epoch 18/20
121/121 [==============================] - 18s 152ms/step - loss: 57.7127 - accuracy: 0.8760
Epoch 19/20
121/121 [==============================] - 18s 152ms/step - loss: 54.2610 - accuracy: 0.8430
Epoch 20/20
121/121 [==============================] - 18s 152ms/step - loss: 73.3188 - accuracy: 0.8430
4/4 - 0s - loss: 61.9039 - accuracy: 0.8595 - 255ms/epoch - 64ms/step

Test accuracy: 0.8595041036605835

Process finished with exit code 0
```

- Bag-15 prediction result




0 1 2 3 4 5 6 7 8 9 10 ... 99