Kaggle Human Protein Atlas

Pascal Pompey (papompey@stanford.edu), Iason Solomos (isolomos@stanford.edu)

project poster, Cs230, Stanford University

Youtube Video Link

Introduction

Proteins are the lego blocks based on which the human is built. Understanding their distribution and role is therefore critical to apprehending how our body functions. Recent advances in medical imagery make it possible to gain further insights by collecting large amount of cell data annotated with their proteins content in an attempt to use machine-learning to automate the annotation process.

Data-Set

Kaggle Data-Set:
- 32,000 images
- 4 Channels: RGB + Yellow
- 512 * 512 resolution
- 28 protein types
- 27 cell types
- multilabel classification problem

Experimental setup:
- split: 90% train, 5% validation, 5% test
- Model: Resnet18 (faster to train)
- Data augment: Horizontal and Vertical flip
- Loss: Binary cross entropy
- Target metric: Mean of F1 scores
- Adam optimizer with vanilla parameters

Your mission if you accept it: Given a microscope image of human tissues, can you predict what protein types are present in it?

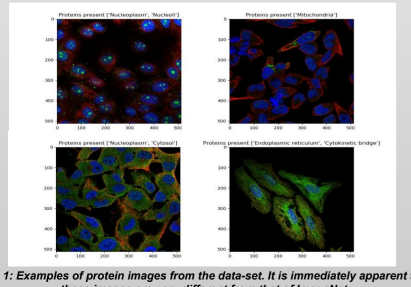

Fig. 1: Examples of protein images from the data-set. It is immediately apparent that these images are very different from that of ImageNet.

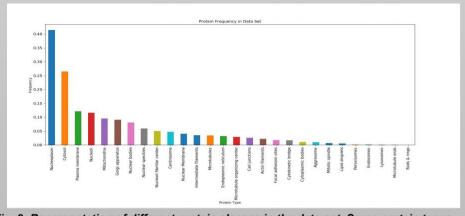

Fig. 2: Representation of different protein classes in the data-set. Some protein types are ubiquitous while others are very rare. We observe an exponential decay in the frequency of proteins.

Challenges:
- adapting resnet to 512 resolution RGBY protein images
- Handling class imbalance
- Finding patterns of mistakes in protein images

Architectural experiments

Core model architecture questions:
- Retrain or freeze deep parameters?
- How to adapt to 512 resolution?
- Shall we add the yellow channel?

Adapting to 512 resolution:
- Downsize the image and use vanilla model
- Average pooling before fully connected layer
- Appending 2 further resnet blocks

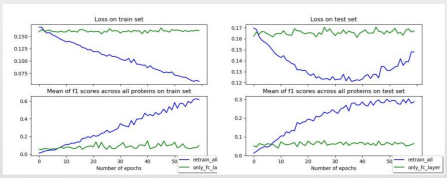

Fig. 3: Retraining yields much better results. Protein images are sufficiently different from ImageNet that retraining deep features in the network is necessary

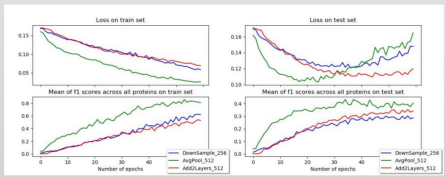

Fig. 4:Using average pooling before the fully connected layer seems to outperform the two other methods for adapting to 512 resolution

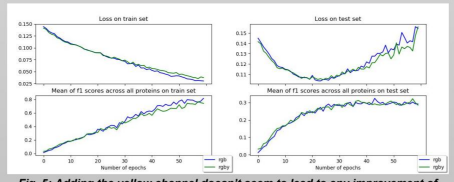

Fig. 5: Adding the yellow channel doesn't seem to lead to any improvement of model performance

Handling class imbalance

Modify the loss to advantage rare proteins
- Vanilla Binary Cross Entropy (BCE) loss
- Weighted BCE Loss
- Focal Loss

$$BCE(y^*, \hat{y}) = y^* \log(\hat{y}) + (1 - y^*) \log(1 - \hat{y})$$

$$weighted\_BCE(y^*, \hat{y}) = (1 - f) BCE(y^*, \hat{y})$$

$$Focal\_Loss(y^*, \hat{y}) = [y^*.\hat{y} + (1 - y^*).(1 - \hat{y})]^\gamma \; BCE(y^*, \hat{y})$$

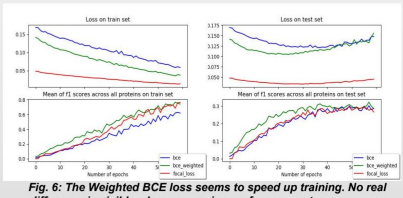

Fig. 6: The Weighted BCE loss seems to speed up training. No real difference is visible when comparing performance at convergence

$$p_t = \begin{cases} p & \text{if } y^* = 1 \\ 1 - p & \text{if } y^* = 0 \end{cases}$$

$$FL(p_t) = -(1 - p_t)^\gamma bce(p_t)$$

Focal Loss formula

Hyper-parameter tuning

Threshold parameter:
- Sigmoid outputs values in [0, 1]
- In most cases .5 is not the optimal threshold
- We fix the optimal threshold on the validation set
- This yielded a .1 increase of f_score from .43 to .53

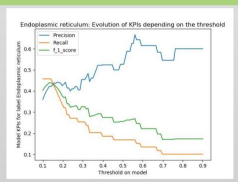

Fig. 7: The optimal threshold for the endoplasmic reticulum is 0.15

Multi-label classification:
- the higher the frequency the better the f_score
- some rare labels benefit from multitask learning
- others don't

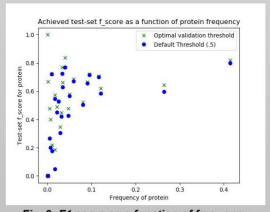

Fig. 9: F1 score as a function of frequency

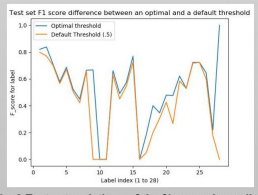

Fig. 8 Test set variations of the f1 score depending on the threshold selected on the validation set

Architectural Conclusions

Core model architecture decisions:
- Retrain all layers -> enables model to adapt to protein images
- Do not downsize, use average pooling before fully connected layer
- Do not use the 4th (Yellow) channel

Having clarified the architecture, the main source of errors was the model predicting 0 for very rare protein types. The team focused on handling class imbalance.

Results and future work

Results
- Final average F1 Score on test: **0.53**
- Optimal architecture is: use Average Pooling with a weighted BCE loss on 512 resolution RGB images
- Some very rare protein types are still not handled properly
- Changing loss doesn't help much to fight class imbalance on multi-label classification problems

Next Steps:
- Use biased sampling and more agressive data-augmentation to handle rare protein types
- No multitask learning: Specialize a model to focus on a single rare protein type
- Try more involved network architectures now that the core architectural questions are answered; e.g. Resnet31, DenseNets, InceptionNets