Human Protein Atlas Image Classification with Convolutional Neural Network

YUEYANG LIN

012486123

Task, Metrics and Goals

- **Task:**
  - Identify a protein’s location from a high resolution microscopy image of protein patterns.
- **Metrics:**
  - Accuracy: The percentage of correctly predicted samples.
  - Turns out to be a bad choice (more on this later).
- **Goal:**
  - Achieve accuracy over 90%.

Dataset Description

- **Input**
  - Over 30,000 samples of 512x512 grayscale image.
  - 4 images per sample (green, red, yellow, blue filtered).
  - Green is used for prediction.
  - The sample is fairly large (>10GB with just green).
- **Label**
  - Multi-class, multi-label.
  - 28 classes.
  - Each sample can have more than one label.

Workflow and Baseline Model

1. Read the image as arrays.
2. Normalize and separate into training and validation sets.
3. Multi-hot encode the labels (28 binary classification).
4. Feed the array in to a convolutional neural network.
   - 4 Convolutional Layers, 2 Fully Connected Layers.
   - RELU for hidden layers.
   - Sigmoid for output layer.
   - Loss: binary_crossentropy
5. Determine each class label with a threshold of sigmoid output.
6. Predict labels of the test dataset with the trained model.

Preliminary Results

- Accuracy over 94% with just a few epochs of training.
- Kaggle leaderboard score of 0.202 (F1 score).

Problems and Proposed Solutions

- **Problems:**
  - Loss is minimized very fast and achieved desired accuracy.
  - But F1 score is very low.
  - By examining the dataset distribution, the classes are highly unbalanced.
  - Some classes has less than 10 samples.
- **Proposed Solution:**
  - Change the loss function in order to minimize F1 score.
  - Evaluate with different metrics (F1 score, precision, etc.) instead of accuracy.
  - Apply data augmentation to classes with fewer samples.

Future Works

1. Modify training schemes including loss functions, performance metrics, etc.
2. Tuning network structures and hyper-parameters to achieve better performance.
3. Apply data augmentation and other regularization techniques to improve test performance.
4. Including more data in the training
   - Augmented data
   - Images from other three filters
   - Try 2048x2048 resolution images to see if it help with performance.