Protein Location Classification

Laura Cruz-Albrecht (lcruzalb@stanford.edu)

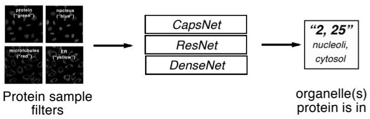
CS230: Deep Learning, Fall 2018

CS230

Problem Statement

Proteins are an integral part of cellular processes; identifying in which organelles the protein is present in can shed better insight into the both the role of the protein in the cell and into cellular mechanisms as a whole.

Goal:	**Train a model that takes in grayscale microscope images for a protein sample and classifies which organelles it is present in.** *=> multi-label, multi-class (28) classification task*



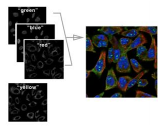
Data & Evaluation

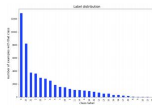
Dataset:
Kaggle Human Protein Atlas Image Classification Challenge Dataset

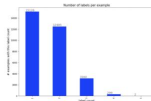
Statistics:
train: 24.9 K, val: 6.2 K, test: 11.7 K (unlabeled)
each sample => 4 512x512 grayscale images ("filters")

Preprocessing:
stacked 3 filters into RGB image
rescaled: baseline (64x64), ResNet/DenseNet (224x224), CapsNet (32x32)

Evaluation: Macro F1-score

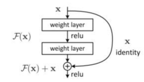




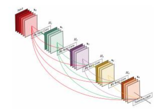


Methods

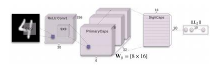
ResNet
Architecture featuring residual connections; applied sigmoid at the end to allow for multi-label classification.



DenseNet
Architecture featuring feed-forward connections between layers; applied sigmoid at the end to allow for multi-label classification.



CapsNet
Good for modeling hierarchical relationships; consists of "capsules" that output vectors; norm ~ probability of class



Loss function
binary cross entropy with logits
experimented with per-class weights

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_n \left[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n)) \right]$$

Loss function
Margin loss, summed over each "digit" (organelle) capsule

$$L_k = T_k \ \max(0, m^+ - ||\mathbf{v}_k||)^2 + \lambda \, (1 - T_k) \ \max(0, ||\mathbf{v}_k|| - m^-)^2$$

Experiments and Results

ResNet

- Used ResNet18 pretrained on ImageNet, froze all layers except final FC
- Hyperparameter search over learning rate and batch size
- Experimented with using: no weights, full weights, and natural log (ln) weights for BCE loss function; found ln weights best

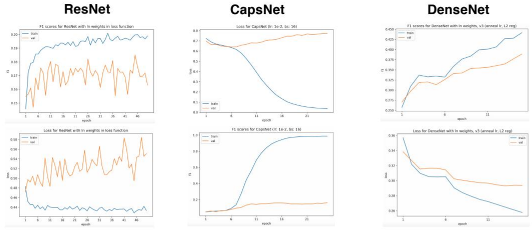
CapsNet

- Used CapsNet model designed for SVHN as starting point, modified to fit task at hand
- Hyperparameter search over learning rate and batch size
- Heavy overfitting

DenseNet

- Used DenseNet161 (4 dense blocks) pretrained on ImageNet
- Experimented with retraining only final layer, and final dense block + final
- Experimented with using: no weights and natural log (ln) weights for BCE loss function; found ln weights best
- Experimented with lr annealing & F1 threshold

ResNet | **CapsNet** | **DenseNet**



Results

Model	Train F1	Train loss	Val F1	Val loss	Test F1
ResNet (no weights)	0.101	0.139	0.092	0.149	0.089
ResNet (full weights)	0.169	1.544	0.153	10.477	0.155
ResNet (ln weights)	0.199	0.433	0.163	0.551	0.156
CapsNet	0.406/0.985*	0.035	0.084/0.163*	0.774	0.067
DenseNet (ln weights)	0.656/0.956*	0.022	0.299/0.395*	0.727	0.249
DenseNet (ln weights, anneal, l2)	0.355/0.441*	0.258	0.306/0.389*	0.294	0.260/0.309+

Table 1: F1 score and loss results. Note: $*$: F1 score calculated over entire set, rather than by batch. $+$: F1 with custom threshold.

Conclusions

Best model: DenseNet
DenseNet proved to be the most promising, obtaining the highest macro F1 score on both the val set and test set

Adding weights to BCE loss function helps
Due to the very unbalanced nature of the classes, adding per-class weights to the loss function helped; we found that ln(#neg/#pos) examples of a class was most effective

CapsNet needs further work to tackle task
The architecture may need more tuning / modification for this multi-class, multi-label problem. CapsNet strongly overfit on the dataset (the train F1 over the whole dataset was 0.985).

Future Work

- **Deeper tuning for CapsNet**. This architecture may still be promising for the task at hand, but would need to be modified to tackle the task. Incorporating weighting to account for inbalanced classes might help.
- **F1 threshold.** Tweaking the F1 threshold boosted DenseNet test F1 from 0.260 to 0.309. Further exploration may further help performance.
- **Use fourth filter**. Since most architectures expected 3 channel images, the 4th filter was omitted. Incorporating it could boost performance.

References

[1] https://www.kaggle.com/c/human-protein-atlas-image-classification/data (Kaggle)
[2] https://www.proteinatlas.org/humancell/organelle (The Human Protein Atlas)
[3] https://pdfs.semanticscholar.org/8015/5ab5da4c739541a4d6b97c0189355ca7d476.pdf (Liimatainen et. al., "Cell organelle classification with fully convolutional neural networks")
[4] https://arxiv.org/pdf/1710.09829.pdf (Hinton et. al. "Dynamic Routing Between Capsules")
[5] https://openreview.net/pdf?id=HJWLfGWRb (Hinton et. al., "Matrix Capsules with EM Routing")
[6] https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks- part-i-intuition-b4b559d1159b ("Understanding Hinton's Capsule Networks")
[7] https://arxiv.org/pdf/1805.08090.pdf (Verma and Zhang, "Graph Capsule Convolutional Neural Networks")
[8] https://arxiv.org/pdf/1712.03480.pdf (Xi et. al., "Capsule Network Performance on Complex Data")