Fast regulated network over variable sets of features with loss annealing

DiXi Yao, JiuZahng Wang,

Computer Science, {Jimmyyao18, wangjiuzhang}@sjtu.edu.cn

***Abstract*—Annotation of gene expression images of Drosophila embryos is a meaningful and intersting task. However, since the complexity and variety of gene expressions, it is also a difficult problem. In this paper, we propose a novel model architecture combining CNN and RNN models. We use CNN model to extract features and use RNN model to learn knowledge in sequences. Apart from that, we also propose a novel optimization methods including applying different loss functions and multiple tempering methods. We then evaluate our method on the open dataset FlyExpress, the experiments show our model can reach 95.9%AUC, 64.77% macro F1 score and 65.93% micro F1 score.The code of our project is opensource on https://github.com/daxixi/Flyexpress-pytorch**

***Index Terms*—Drosophila enbryos, Residual Network, multi-label classification, Recurrent Network**

I. Introduction

THE recent progress of deep learning and machine learning has witnessed a variety of machine learning applications especially in bioinformatics. In situ hybridization (ISH) is a family of imaging methods that use microscopic imaging technology to measure and localize gene expression within tissues or cells.

For ISH problems, multi-label classification is a common and critical problem, since a gene image can represent various features of a particular gene stage. FlyExpress [1] is a dataset collecting large amounts of drosophila gene stages. Another ciritcal problem is the relationship between the image and the gene stage is not a one-to-one relationship. There may exist several images to contribute for one stage. As a result, we need to use a bag of images to learn a presentation of a gene stage. Previous works use some trivial models to solve such Multi-instance Multi-label problem (MIML) [2]. Howeever, there exsits minor exploration on such dataset. As a result, we propose a novel model archirecture and training methods to achieve state-of-art results on the dataset. We are able to achieve relatively high AUC and F1 score to give a good identification on various attributes.

After analyzing our task, we use a hybrid architecture called **Fast Regulated Network** which consists of a CNN to extract features of images and a RNN to deal with the sequence information. Besides, we also propose some optimization methods instead of training large amount of DNN and alleviate the workloads of excessively tuning since we only have 2 submission chances on kaggle everyday. Our overall architecture is shown as figure 1.

Three authors are with the Department of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

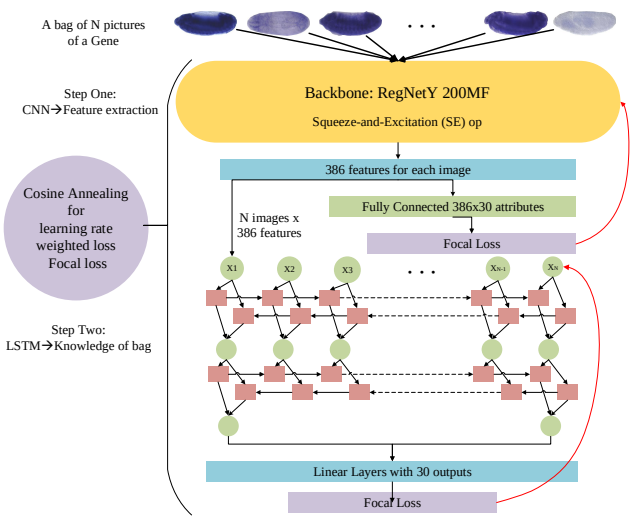

Fig. 1. The overall architecture

In the remainder of the paper. We introduce some related works and preliminaries of our work in Section III. In Section III and Section IV we introduce our main method. Then we give the evaluation in Section V and conclude our work in Section VI.

II. Preliminaries

In this section we will introduce some techiniques and architectures may help understanding our methods.

A. Multi-label tasks

Different from single label tasks, multi-label tasks needs to learn to identify multi labels while the distribution over samples and attributes may be quite uneven. A trivial method is to decouple multi-label tasks into several single label tasks. However, such method can fail to dig the latent correlation between different labels [3], [4].

B. RegNet

RegNet was first proposed in [5] which is an improvement over commonly used architecture ResNet [6] but with more specific advantages , which support us to choose it. Neural architecture search (NAS) is an outstanding method to search for optimal model architecture for particular tasks. Even we

Model	Flops (Billion)	Parameters (M)
RegNetY 200MF	0.2	2.7
ResNet18	2	33.16
MobileNet V2	0.59	6.9

TABLE I
THE COMPARISON OF REGNET AND OTHER COMMON MODELS IN EFFICIENCY

can use NAS method to search a state-of-art method on non-i.i.d. distributions [7]. While a severe problem is the choice of search space, in [5], the authors well design a search space and find outstanding model architectures. RegNet is one of them. Since it is searched on NAS space, the RegNet has powerful fit ability. Besides, the parameters of it is also very small shown as table I which is very efficient for us to optimize since we only have 500 RMB budget on HUAWEI servers.

Squeeze and Excitation. We also implement the SENet [8] to help solving the problem. The squeeze and excitation enables the network to give attention to extract both global information and local information. Different channels in the model can have different attention to different features. In our problem some attributes are related to the global feature such as *ubiquitous, faint ubiquitous*, while some attributes are highly related to only a small part of the gene images such as *amnioserosa,yolk nuclei*. We finally choose the **RegNetY 200MF** as our backbone network.

C. LSTM

Apart from multi-label tasks, our task is also a multi-instance task, which means several instances contribute to only one gene stage. However, the numbers of images in bags are various, as a result, since RNN has comparable ability to understand information of sequences and deal with various length problems. We use the Bi-directinal LSTM [9] to understand the bags.

D. Focal Loss

Focal Loss [10] was first introduced to solve the uneven samples in object detection tasks. In such tasks, there exist much backgrounds which are negative samples while may only one positive samples. As a result, the strong unbalance may greatly effect the performance of one-stage object detection. The formulation of focal loss is

$$L = -y\alpha(1-y')^{\gamma}\log y' - (1-y)(1-\alpha)y'^{\gamma}\log(1-y') \quad (1)$$

where y is the labels and y' is the prediction probability. In such way, the network can lay more efforts on solving hard samples while ignoring those simple samples.

Following such idea, we extend the method to our architecture and solve the problem of unbalance intra attributes.

E. Semi-Supervised Pretraining

Semi-Supervised pretraining has already achieved much success in helping model having better ability to extract and capture the latent features in the image. Microsoft pretrained the model on large wild dataset with semi-supervised method and transfered it to the imagenet and achieved good performance [11]. The core method is SWAV [12]. Such method uses different data augmentation methods to preprocess the same image and minimized the MSELoss between the feature representation of one same network. We also apply such method to give our model a warmup and let it have some prior knowledge of drosophlia gene images. Such prior knowledge can better help to optimize the network.

III. MODEL CONSTRUCTION

This part introduces how we build our architecture and the pipeline of training process. We first seperate the bags and treat each image equally. The preprocess of images will be introduce in section V. Then we use our backbone network to do the job of feature extraction. We set the last layer of backbone network as a fully connected layer with 30 outputs. After 30 outputs, we directly use the sigmoid function on each activation and use them as the probability corresponding to the attributes. After trainng the backbone network to convergence, we will get a 386 dimensions of features for each images.

After turn the high dimension data (images) into low dimension features (386 vectors), we then use the Bi-directional LSTM to obtain knowledge of the bags. We treat a bag of data as one sample. So each sample is $N \times 386$ where N denotes the images in one bag. We use a two-layer wise Bi-LSTM, the procedure of LSTM dealing with a bag of data is similar to the process where we look images one by one from the first to the last and then from last back to the first. We can use such equation to represent one layer of LSTM

$$h_n = f(h_{n-1}, X_n) \quad (2)$$

, where $n \in (1, \cdots, N)$. Through this way the final output of hidden layer h_N is our result, and the hidden dimension of it is 1024. Because we use the bi-directional LSTM, so h_1 also contains the useful knowledge. Then we put the $[h_N, h_0]$ into the fully connected layers, average pooling layers and fully connected layers to get 30 activated cells which represent the probability of each attributes.

IV. MODEL OPTIMIZATION

A trivial and exhausting way to optimize the model is grid search on all possible learing rate and train multiple networks and generate the result with feature reconcile. But such methods cannot well solve many problems and considering our computation limits (budget limit actually) and time cost, we propose novel and more efficient optimization methods to solve our ptoblem which achieve better performance (seciton V). The loss we choose to train our CNN and RNN is BCELoss

$$Loss = -\frac{1}{M}\sum_{m} Y\log Y' + (1-Y)\log(1-Y') \quad (3)$$

where M is the num of attributes and Y is the batch of labels and Y' is the prediction probabilities of the batch of data.

Atr	Stage	Atr	Stage	Atr	Stage	Atr	Stage
2	6	9	6	17	5	24	6
3	4,5	10	6	18	4	25	3
4	5	12	6	19	4	26	3
5	5	13	5	20	4	27	3
6	6	14	6	21	6	28	6
7	6	15	6	22	3	29	4
8	2	16	5	23	4,5		

TABLE II
THE RELATIONSHIP OF SOME ATTRIBUTES AND GENE STAGES. THESE ATTRIBUTES ONLY EXIST IN SUCH STAGES

A. Balance

The first problem we solved is balancing intra and inter attributes. Intra one attribute there exists much unbalanced samples. In one attribute there may exist much more negative samples than positive samples. To solve such problem we use the focal loss over BCELoss.

To solve the problem of unbalance inter attributes. We use weighted BCELoss methods. Some attributes may be easier comparing to other attributes, as a result the loss of each attributes may various much and some attributes may finally fail to converge to optimal solution since the network negelects them too early. As a result, we give different weights of different attributes:

$$Loss = -\frac{1}{M}\sum_{m} w_m \left(Y \log Y' + (1-Y)\log(1-Y')\right) \quad (4)$$

where w_m is weights of different attributes.

We also find that for some attributes, it only exists in particular stage in the gene instead of all posible stage shown as table II. As a result, if we find a gene stage is not the corresponding stage for the attributes, we set the value of loss weight of that sample to 0 directly.

B. Annealing and Reverse-Annealing

Anealing algorithm is a common method in the area of machine learning. The cosine annealing learning rate:

$$\eta = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + cos(\frac{T_{cur}}{T}\pi)) \quad (5)$$

is a common scheduling method in training nerual networks. We also use it. Besides that the promise of such method give us the spirit to the schduler of other hyper parameters.

γ in Focal Loss. One problem of focal loss is that at the early stage of training a neural network, it is difficult to clarify which sample is easier sample because the losses are all relatively big. A more reasonable way is giving all samples equal efforts first and gradually change the value of γ to better learn on different samples. As a result, in the first 10 epochs of training CNN, we gradually increase the γ from 1 to 2 by the scheduler of cosine annealing.

Weights in BCELoss. Weights in the BCELoss is a hyper parameters highly reliable on human setting. Apart from that, we find whether continually using the weight BCELoss or not can both lead to overfit. While not using the weights can lead to underfit. Following this idea, we propose a method called **multiple tempering** which means applying multiple times annealing and reverse annealing. First set the weights from w_m to one then from one back to w_m and repeat such processes several times to find a good balance point between overfitting and underfitting ,and that may possibly be a good solution. As shown in figure 2, though the problem of overfitting still exists, implementing multiple tempering can alleviate it.

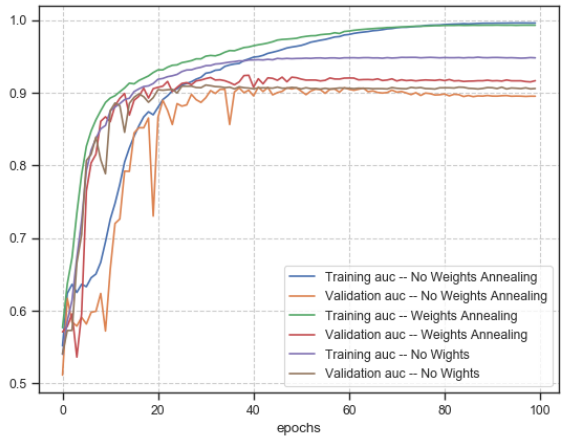

Fig. 2. The comparison of not applying weights BCEloss, applying weights BCELoss and applying weights BCELoss with multiple tempering

V. EVALUATION

We choose FlyExpress [1] as our dataset which consists of more than 4500 gene stages and 15000 images. For each gene, it has 30 attributes to identify. We use 45% of the dataset as the training dataset, 5% as the validation dataset, and the rest as the test dataset.

A. Hyper parameter setting

The first stage is warmup for pretraining. We use batch size 32 and run for 2 epochs. The learning rate is 4.8 and we use Adam optimizer to optimize model. Different random data augmentation methods include random horizontal, random vertical, GaussianBlur, GaussianNoise, and random rotation with maximum 15 degrees.

The second stage is training CNN. We use batch size 128 and run for 100 epochs. The initial learning rate is 0.005 and decay to 1e-8 under cosine scheduler with 90 epochs. The optimizer is Adam. The initial γ of focal loss is 1 and set to 2 in first 10 epochs. The weights in BCELoss is initially set at the propotion of negative samples and positives in each attribute. Then from 40 to 85 epochs the weights will go through multiple tempering following 30 half period and freeze at the 85 epochs.

The third stage is traning RNN. We use batch size 256 and run for 60 epochs.. The initial learning rate is 0.01 and the decay to 1e-8 under cosine scheduler with 60 epochs. The optimizwr is Adam. The scheduler of γ is the same as the second stage. No modifications on weights. The dropout rate of linear layers are 0.5.

TABLE III
THE THRESHOLD OF SETTING PREDICTION, WITH ONE ROW OF ATTRIBUTE AND THE NEXT ROW OF THRESHOLD

0	1	2	3	4
0.53	0.74	0.9	0.26	0.62
5	6	7		9
0.19	0.64	0.95	0.89	0.91
10	11	12	13	14
0.74	0.61	0.66	0.97	0.82
15	16	17	18	19
0.93	0.11	0.21	0.87	0.66
20	21	22	23	25
0.84	0.75	0.79	0.88	0.57
25	26	27	28	29
0.35	0.99	0.99	0.97	0.98

TABLE IV
THE EVALUATION ON VALIDATION DATASET WITH DIFFERENT METHODS

	AUC	macro F1 score	sample F1 score
ResNet	0.9188	0.5419	0.5304
RegNetY	0.9118	0.5491	0.5342
Our CNN	0.9240	0.5425	0.5671
Our CNN+RNN	**0.9590**	**0.6477**	**0.6593**

B. Preprocess

The major preprocess method we adopt is data augmentation. We first resize the image into 116×340 for better fitting the nerual networks. Using the random crop to 112×336 with four padding during the training phase. Random flip and rotation with 15 degrees are applied. The image is added gauussian noise and transformed into grayscale images. The image tensor is normalized with $\mathcal{N} \sim (0.61, 0.12)$.

C. Postprocess

After getting the result, we evaluate the performance with F1-score. We need to convert the propability into 0-1. Since the severe unbalance of samples, the standard to decide the prediction result is not strictly 0.5. Because of the unbalance, the model may prone to increase the probablity if there exist much positive samples. As a result, threshold refine is necessary. So we set the different threshold t for each attributes. If p is over t, then the predicted class is 1 otherwise 0. The threshold is decided on validation set and as table III shows.

D. Feature representation

To better display the generation ability of our model, we also use some common visual evaluation methods to better evaluate the feature extraction ability of CNN models. Guided backpropagation [13] trace the graidient through the network and evaluate whether the network can capture the edge information of the image. As shown in figure 3, we can see the network can precisely capture the edge including the detailed edges of some particles in the gene, not only the large boundary of the gene. So the network can really capture the features.

Also, we evaluate the performance of semi-supervised warmup shown as figure 4. Though,only by semi-supervise training, the network prone to lose some important features. The network can also lay some attetion on different practicles in the gene. We can think of this step as providing good initiaization parameters for the neural network.

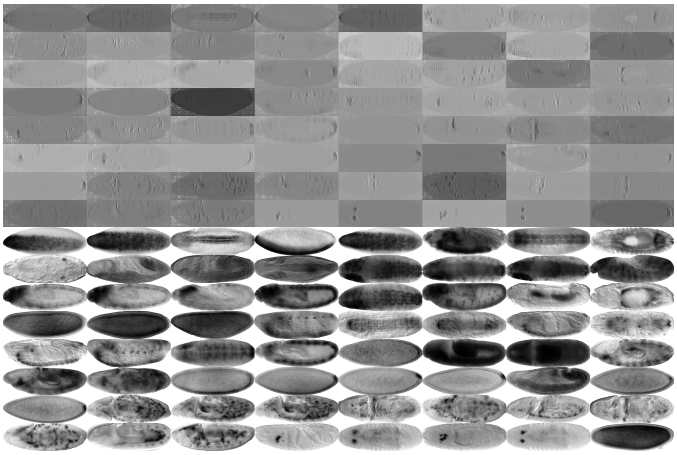

Fig. 3. The visualization of guided backpropagation of samples, the upper is the gradients of images and the lower is the original images.

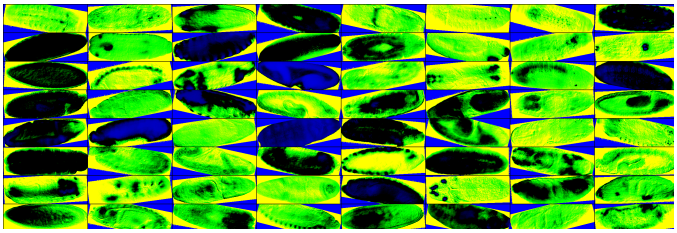

Fig. 4. The visualization of heat map of inter mediate layers of RegNet Y after doing the semi-supervised warmup.

E. Results

Here we choose three different metrics to evaluate our model: AUC, macro F1 scroe and sample F1 score. The detailed definition can refer to sklearn package.

Besides using the RegNetY, we also try the performance of ResNet at very begining of our project. Table IV lists the performace of adopting different network architectures and methods. We can see the RegNetY has little improvement over ResNet. But after applying our methods which is our CNN in the table, the performance can be better. After implementing the whole architecture of our methods, the performance is outstanding.

On the final evaluation over the test dataset, our model can yield AUC **0.95561**, macro F1 score **0.62405**, and sample F1 score **0.61615**.

F. Furture Work

Though our method and architecture has already reached the SOTA performance, there are still several points that could be further improved. First, due to the limitation of time and resources, we haven't done any hyperparameter optimization like adjusting different learning rate. We just use the default learning rate of our backbone model. Though the current results show our model can achieve relatively good

performace, trying other leanring rates and hyper parameters can further imporve the performace.

Another direction of imrovement is adopting feature aggregation, actually we can train several models and use voter or other mechanism to aggregate the feature of these models to yield better performace.

VI. Conclusion

Anotation of gene expression images of drosophila embryos is a very meaningful and interesting task. However, it is not a easy job consedering its a MIML problem. We propose a novel architecture through feature extraction and sequence learning to address such problem. Besides, we also propose novel optimization methods, especially multiple tempering methods to better train the neural network and yield nice performace. The evaluation of our model shows we can achieve start-of-the-art performance and have a good explanation of extracted features. Further works can lay on trying various hyperparameters and using feature aggregation method.

Acknowledgement

We give our sincerest appreciation to Prof.Yang and Dr. Tu for giving us this chance to contribute our efforts in the work of drosophila embryos multilabel classification. We also thank them for their generous help.

References

[1] S. Kumar, C. Konikoff, B. Van Emden, C. Busick, K. T. Davis, S. Ji, L. W. Wu, H. Ramos, T. Brody, S. Panchanathan *et al.*, "Flyexpress: visual mining of spatiotemporal patterns for genes and publications in drosophila embryogenesis," *Bioinformatics*, vol. 27, no. 23, pp. 3319–3320, 2011.

[2] Y.-X. Li, S. Ji, S. Kumar, J. Ye, and Z.-H. Zhou, "Drosophila gene expression pattern annotation through multi-instance multi-label learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 1, pp. 98–112, 2011.

[3] S.-J. Huang and Z.-H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012.

[4] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, 2017.

[5] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 428–10 436.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] D. Yao, L. Wang, J. Xu, L. Xiang, S. Shao, Y. Chen, and Y. Tong, "Federated model search via reinforcement learning."

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[9] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[11] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin *et al.*, "Self-supervised pretraining of visual features in the wild," *arXiv preprint arXiv:2103.01988*, 2021.

[12] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.

[13] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.