Large scale analysis of calcium imaging data

Yidi Zhang,Yang Sun, Shangying Jiang, Yiran Xu, Zhaopeng Liu

1

1. Research Question

Our major task would be improving an existing convolutional neural network based approach to classify spatial components extracted from the CNMF algorithm into four classes: neurons, doubtful neurons, processes and noise artifacts, based initially only on spatial features, but with possible extension to temporal features, like shape of calcium transient. We would apply deep convolution network on our classification task, based on the following two papers.

- Deep Residual Learning for Image Recognition [He et al. 2016]
- Very Deep Convolutional Networks For Large-Scale Image Recognition [Simonyan and Zisserman 2014]

If time permitted, our minor task would be implementing optimized version of the online algorithm in C++ or Java, possibly incorporating the classifier with the Fiji package.

2. Data

Our data is generated from calcium imaging videos that monitored large neuron populations. By identifying the locations of the neurons, demixing spatially overlapping components, and denoising and deconvolving the spiking activity from the slow dynamics of the calcium indicator, we extracted 9028 50×50 pixels pictures, and each contains exactly one component. Each component will have one 50×50 numpy array that contains pixel values as mask, one label that indicates which class the component belongs to and one trace that describes the temporal feature.

2.1. Manual Annotation

The original task was to recognize only neuron and does not focus on anything else. However we are expected to classify not only neurons but also processes and noise. After taking closer look at the data, we realize we need another class that specifically describes doubtful neuron since we don't want to confuse our classifier. Through rigorous training on recognizing these components, we are able to label our data as four classes: clear neuron, process, noise and doubtful neuron.

2.2. Target Variable

Our goal is to build a classifier that can successfully classify four different kinds of components. So our target variable is "label" which can take four categorical value: 1 stands for class neuron, 2 stands for class process, 3 stands for class noise and 4 stands for class doubtful neuron. Clear neurons should be bright and must have clear donut shape and ideally have process attached to it. Processes are dendrites which usually appear as bright dot, line segment or V-structure in calcium imaging. Noises do not have clear contour and are relatively dim. Doubtful neuron might be bright and have certain shape but not clearly donut-like. We currently have 2004 clear neurons, 3253 noises, 1487 processes and 2237 doubtful neurons. Our labeling is subject to change after we train our model and decide whether our labeling will confuse the classifier.

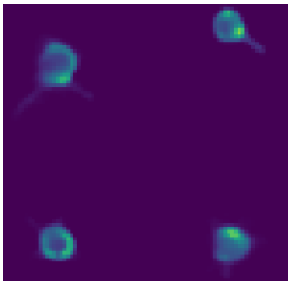

Figure 1. class neuron

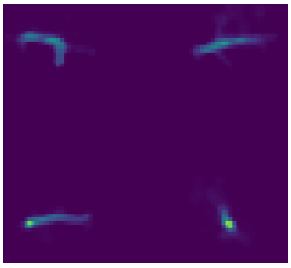

Figure 2. class process

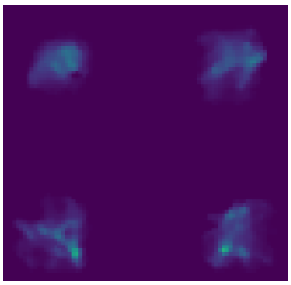

Figure 3. class noise

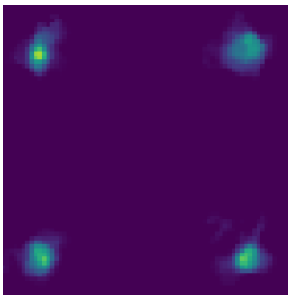

Figure 4. class doubtful neuron

3. Methodology

3.1. Train Data Creation and Prepossessing

We first classify our raw data into four classes: clear neuron, doubtful neuron, process, and noise, with dropping out some very unclear components.As mentioned in the target variables section, the criterions for labeling are: clear neurons are usually donut-shape, or very round (see Figure 1); processes are usually line-shape, V-shape, one to multiple small dots (see Figure 2); noises are usually vague, blurred cloud (see Figure 3); doubtful neurons are others between clear neuron and noise (see Figure 4).

To train neural nets for image classification, a large amount of image data is required. Therefore, after semi-automatic labeling work, we apply the data augmentation, using shears, rotations, shifts, flips, zooms, etc., to create new data and balance training data set.

3.2. CNN based approach

The initial model we use to do classification automatically is Convolutional neural network (CNN) based architecture. It has four convolutional layers, two pooling layers and two fully connected layers. The model is trained with batches of 128 images, which is driven from the augmented data. The loss function used is categorical cross-entropy, and the classification is made based on log-softmax function. And the evaluation method is accuracy.

Both dropout and data augmentation are effective approaches to reduce overfitting and improve validation/testing performance. We have included three dropout layers in the CNN model: two after the max pooling layers and the other one between the two

fully connected layers. Thus, no additional dropout techniques were applied to the model for accuracy improvement.

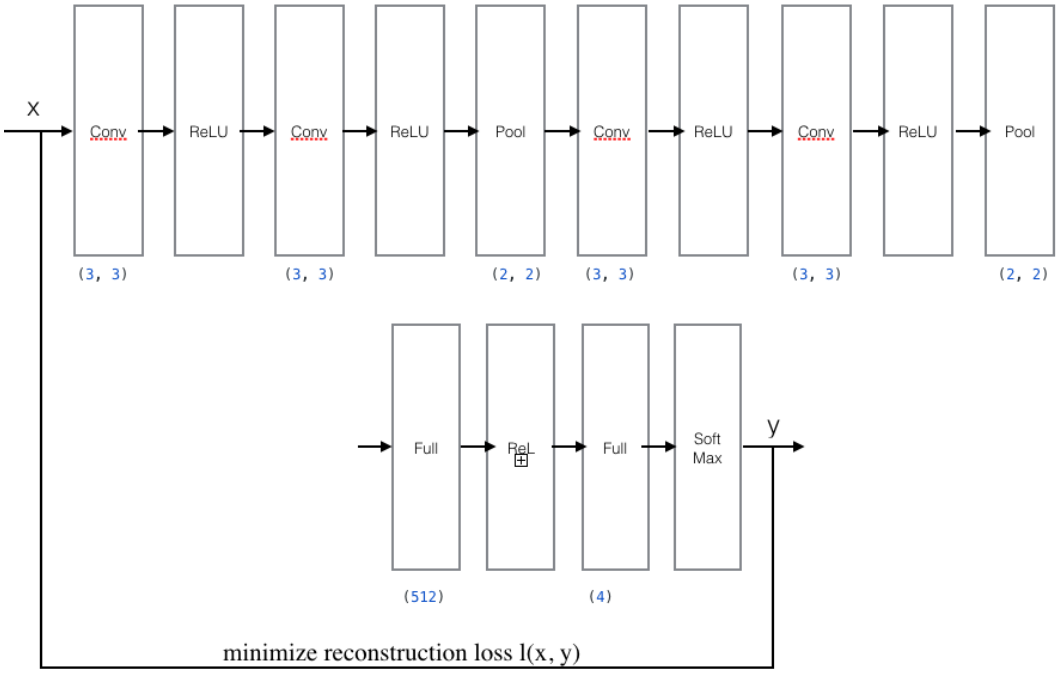

Figure 5. Basic Model Architecture

3.3. VGG-like architectures and ResNet

Convolutional neural network has been proved very useful in image processing. Networks of increasing depth using an architecture with very small (3 3) convolution filters shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers.[Simonyan and Zisserman 2014]

By adding depth of our convolutional neural networks, our CNN model can lead to a lower training error, which has been proved that such architecture can achieve state-of-the-art results in large scale image recognition.

Although convolutional neural networks can produce good results for image classification, they are hard to train with more and more layers. What make things worse is that when we add more layers, a *degradation* problem appears. That is when the networks depth increases, accuracy gets saturated and then decreases rapidly. It is not caused by overfitting and adding more layers to a suitably layers model may lead to higher training error rate. [He et al. 2016]

This probelm can be addressed using a deep residual learning framework (ResNet). We can let layers fit a residual mapping, instead of stacking layers to fit an underlying mapping. It is shown in the ResNet paper [He et al. 2016] that optimizing the residual mapping is easier than optimize the original mapping and gains higher accuracy

with increased depth. Thus, ResNet gains much better performance than deep CNN [He et al. 2016].

4. Results

After careful exploration, labeling and discussion with our mentors, we labeld our data into four classes: clear neuron, doubtful neuron, process and noise.The numbers of components for each class are:

- Number of clear neuron: 2004
- Number of doubtful neuron: 2237
- Number of process: 1487
- Number of noises: 3253

Note that these labels, especially for doubtful neuron, are tentative since it's hard to keep consistent labeling patterns for some borderline components. We will analyze the results of our classifiers carefully to identify components that confuses our classifier, then modify lables based on error analysis.

We are still in the process of refining our labels, getting farmiliar with Convolutional Neural Network models. Thus, we do not have results for classification tasks. We will begin training classifiers and start parameter tunings next week.

5. Future Work

5.1. Semi-automatic labeling

Even though human annotations are accurate for small dataset, going through it multiple times is still time consuming with minor consistency error. Among our four classes, processes are the most distinguishable from other three classes, so we plan to write a script to capture the strongest signal from components in the processes class. The returned signal might be included in the parameters, which can help improve our efficiency and reduce misclassification in the future work.

5.2. Labeling change

In the original dataset, some components have borderline features, which can easily be categorized into two classes during human labling. After training the classifier, we will be able to identify which components will confuse classifier, and changing human labels of corresponding components will reduce categorical cross-entropy error.

5.3. Paramerter tuning

Selecting the best model and seeking for the accurate model parameters are essential step towards model caliberating success. In the below table (Table 1), we show models we plan to train and the parameters and their ranges we plan to tune in this project.

Table 1. Parameter and range

Parameter	Initial Value
Layers	4, 6, 16, 19
Kernel Size	$3\times3, 5 \times 5$
Activation function	RELU
Max Pooling layers	$2\times2, 3 \times 3$
Dropout Size	0.25, 0.5
Number of classes	3, 4
Batch size	32, 64, 128, 256
Epoch	5000
Learning rate	$10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$
Model Selection	CNN, VGG, ResNet

References

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.