**Organic Network for Hierarchical Classification**

*Abstract*

Convolutional Neural Networks (CNNs) achieves the state-of-the-art accuracy in image classification. CNNs are able to extract features that make the classification problem linearly separable. We propose an architecture which exploits the hierarchical structure by sharing the parameters of CNNs among different subclasses.

*Introduction*

A very large amount of research machine learning and deep learning has focused on flat classification problems. Flat classification problem refers to standard binary or multi-class classification problems which assumes the target distribution as Bernoulli or Categorical Distribution. However, we human beings classify objects based on naturally structured taxonomy. In fact, flat classes can be organized hierarchically into a Tree or DAG (Directed Acyclic Graph). We aim to explore and exploit the hierarchical structure of labels in order to create better features with CNNs for later classification.

*Proposed method*

We explore the parameters of CNNs that can be shared among data from different subclasses. If all classes share the same parameters, then it reduces to the normal usage in deep learning. We simply pass all data into CNNs, extract features and then classify with fully connected layers. On the other hand, we can use separate CNNs for each subclass classification. In this case, each subclass will be associated with its own feature extractor independently. The situation we are interested, however, is in the middle. The parameters associated with different subclasses can be shared. We believe there exists a balance between the number of parameters and the rate of sharing.

*Application*

We conduct experiments with EMNIST (balanced) dataset. EMNIST (balanced) contains 47 categories. EMNIST dataset contains two superclasses, digit and letter, and digit contains 10 classes while letter contains 37 classes. We also resize EMNIST data from 28x28 to 32x32. We train totally 350 epochs. The SGD (Stochastic Gradient Descent) optimizer is used with scheduled decaying learning rate at 150th, 250th epochs. All experiments are run on NVIDIA TITAN Xp with batch size 128. We first forward data into non-shared convolution layer with kernel size 1 in order to enlarge the channel size. Then three standard residual blocks, parameters can be dynamically shared, are used to extract features. Finally, a non-shared fully connected layer is used to classify labels. The details of the residual block and how the parameters are shared are shown in Figure 1. We first allocate twice the parameters used in all sharing convolution layer. Then by increasing the sharing rate, we gradually grant free parameters to the feature extractor of letter classification. As our result shown in Figure 2, by adding a small number of free parameters for letter feature extractor, we can achieve a 5% accuracy gain. Meanwhile, training subclasses separately perform worse than sharing the majority of parameters. This indicates that there exists a substantial amount of space for compressing models that target on different objectives.

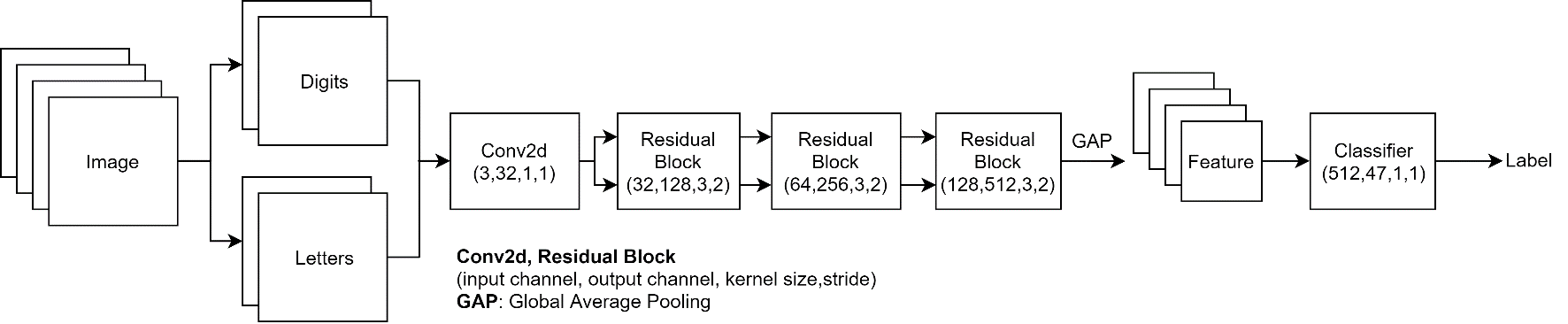
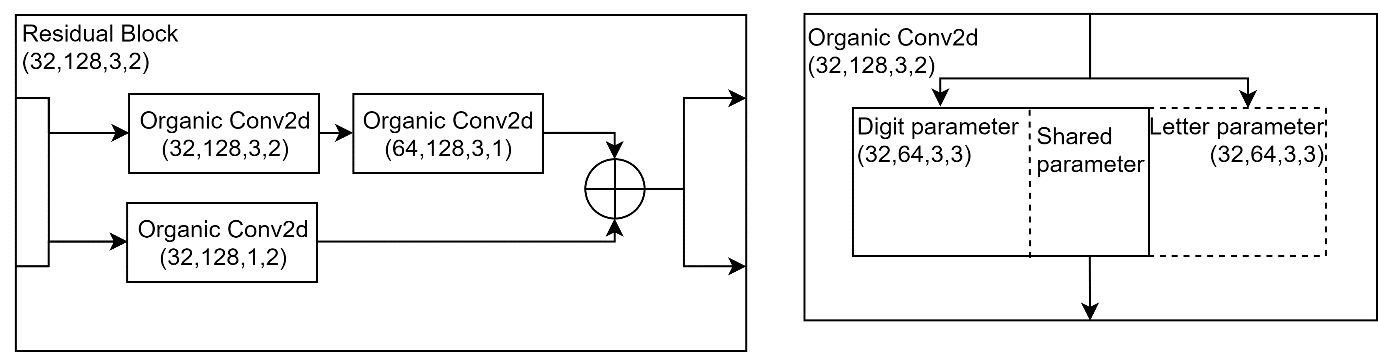
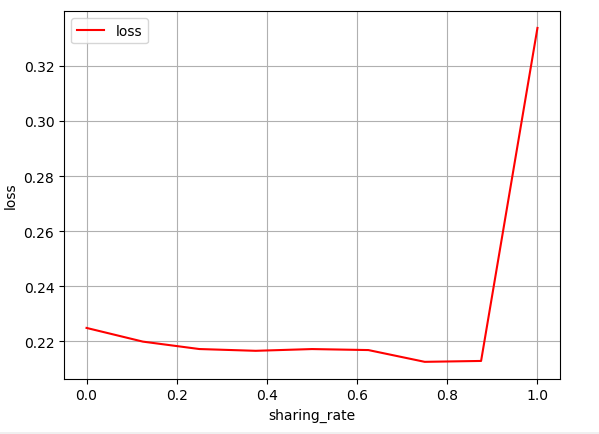


Figure 1(a). Overall model architecture used in our experiment.



Figure 1(b). Residual Block and Organic Convolution layer.

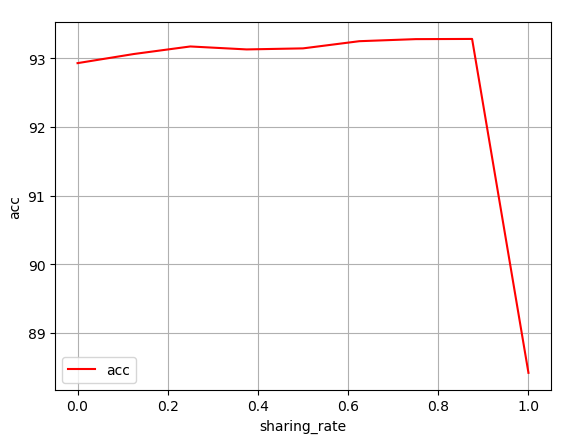


Figure 2. Loss and accuracy vs rate of parameter sharing.

*Future work*

We plan to experiment more on the balancing between the number of parameters and the rate of parameter sharing. By fixing the rate of sharing, we can enlarge the number of parameters and the result can be more straightforward. We will also explore adding new classification targets on top of existing trained models by enforcing them sharing some of existing parameters. Therefore, we do not need to store separate feature extractors for every specific target. Not only we transfer and share existing knowledge but we also want to use the least possible space to achieve that.

*Conclusion*

We explore and exploit the hierarchical structure of labels and show that there exists a substantial space for combing feature extractors aiming different targets. We also show that there exists a balance between the number of parameters and the rate of parameter sharing.