Carnegie Mellon

Deep Neural Network Ensembles for Extreme Classification

Yichen PAN, Yuxuan SUN
Carnegie Mellon University

Overview

Problem:
- Extreme classification on Cdiscount's dataset
- Much noisy data
- Image-wise classification => product-wise classification

- Goal:
  - Classify each product (with 1-4 images) into 5,270 anonymous categories

	Train	Validation	Test
Images	12,118,359	252,934	3,095,080
Products	6,924,452	147, 758	1,768,182

Methodology:

Deep Ensembles:
  Resnet:
  - Residual Learning
  - Identity Mapping by Shortcuts

  Inception
  - Stack of Inception modules
  - Convolution factorization

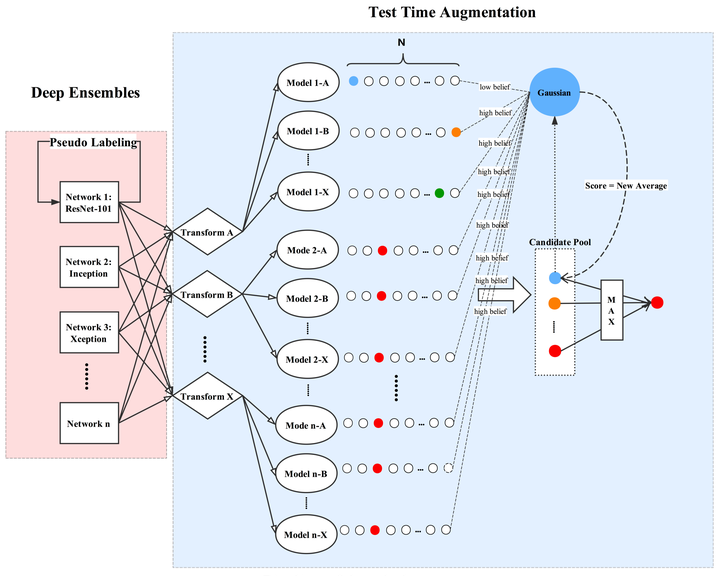
  Xception:
  - Depth-wise separable convolution layer
  - Residual connection

Test Time Augmentation (TTA):
- Augmentation: random resize/flip/shift/crop/rotate

Prediction Aggregation:
- Naïve: average of augmented predictions => vulnerable to noise
- Robust: A Gaussian confidence interval based prediction aggregation algorithm:

Algorithm 1: Belief-based Prediction Aggregation

```
function prediction_aggregator (P, D);
  Input  : Probability distribution over target categories of all images for
           one product P, product D
  Output: Category prediction of the product
for candidate ∈ cadidates_pool do
    for image ∈ imgaes_D do
        calculate exclusive average and standard deviation of probability
        predictions of candidate category among all images except for
        current image;
        if P(candidate|image) outside 95% confidence interval then
            exclude the image;
        else
        end
        update candidate score;
end
find winner with highest score;
```



Pipeline of the whole system

Pseudo-labeling:
- Pseudo-Label in a fine-tuning phase
- Regularizing effect
- Balance between original data and pseudo-labeled data

$$L = \frac{1}{n}\sum_{i=1}^{n} D(y_i, l_i) + \alpha(t)\frac{1}{n'}\sum_{i=1}^{n'} D(y_i', l_i')$$

$$\alpha(t) = \begin{cases} 0 & t < T_1 \\ \dfrac{t - T_1}{T_2 - T_1} & T_1 \le t \le T_2 \\ a_f & T_2 < t \end{cases}$$

Experiments/Results

- Based on Tesla K80 Accelerators with one GPU of12 GiB of memory
- Resume on pre-trained models from Pytorch
- Gradient accumulation
- SGD with scheduled learning rate
- Augmentation: random resize/flip/shift/crop/rotate
- Batch size trick: increase the batch size rather than decay the learning rate

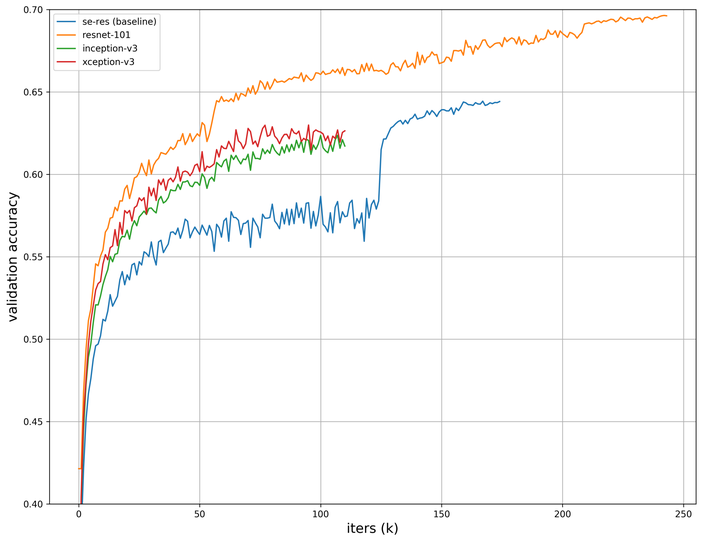
	Validation Accuracy
Resnet-101-baseline	0.638
Resnet-101-TTA (1-2)	0.648
Ensemble(ResNet101+Inception+Xception)-TTA (1-1)	0.653

	Test Accuracy
Resnet-101-baseline	0.683
Resnet-101-TTA (1-2)	0.703
Resnet-101-Pseudo-Labeling	0.712

Experiments Results

	Se-ResNet-50 (Baseline)	Inception	Xception	ResNet-101
Input Size	160*160 crop from 180*180	180*180	180*180	160*160 crop from 180*180
Batch	160*2	64*4	64*4	64*4
Time per epoch	55.54h	31h	31h	64h

Network Settings



Comparison of Model Accuracy of Validation Set

Future Work

- Implement affine transformations as layers in a network
- Further fine-tune the models