Error-Driven Incremental Learning in Deep Convolutional Neural Network for Larg-Scale Image Classification

Xiao, Tianjun, et al. (2014)

Proceedings of the 22nd ACM international conference on Multimedia.

Presenter: Sarah Kim

2018.09.21

Introduction

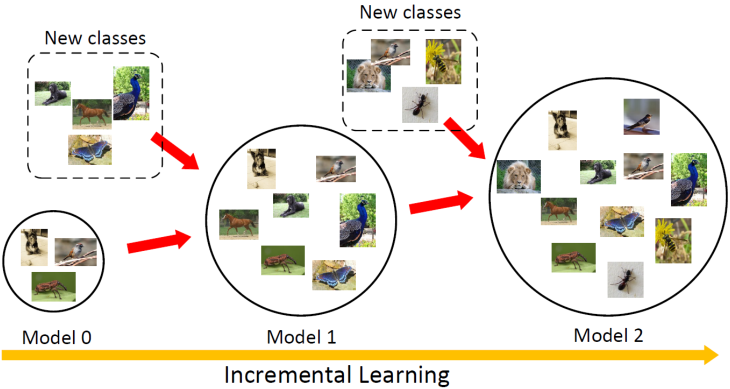

Figure 1 : Incremental learning in multiclass classfication

- We developed a training algorithm that grows a network not only incrementally but also hierarchically.

Incremental Learning Model

- Assume there is a model M_0 that is already trained on N_0 classes.
- Goal: evolve from M_{i-1} to M_i to train N_i classes, in which $N_i - N_{i-1}$ are new classes for $i = 1, \ldots, T$.
- The model must increase it capacity to accomodate more classes:
  1. Flat increment: the output units is increased to hold more classes.
  2. Clone increment: the total classes are partitioned into superclasses, and consider a hierachy of models.

Incremental Learning Model

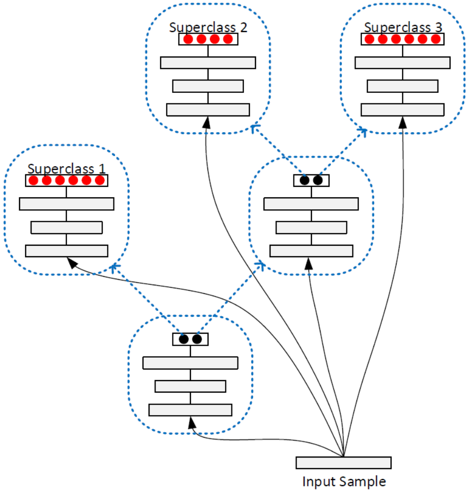

Figure 2 : A hierachy of models: branch models predict superclasses, leaf models return final predictions.

Starting from a Single Superclass

- Here, we discuss the details of this incremental learning.
- In the starting point of the training, all N_0 classes are in one single superclass and predicted by one model L_0.

1. Flat increment: extend L_0 to L_0' by inserting more output units, which increase a small amount of capacity.
2. Clone increment:
   - Partition the superclass into K superclasses;
   - Clone L_0 into several new leaf models $L_1, \ldots, L_K$ to predict final outputs.
   - A branch model B with K final output units is also cloned from L_0 to predict a correct leaf model on a given input sample.

Starting from a Single Superclass

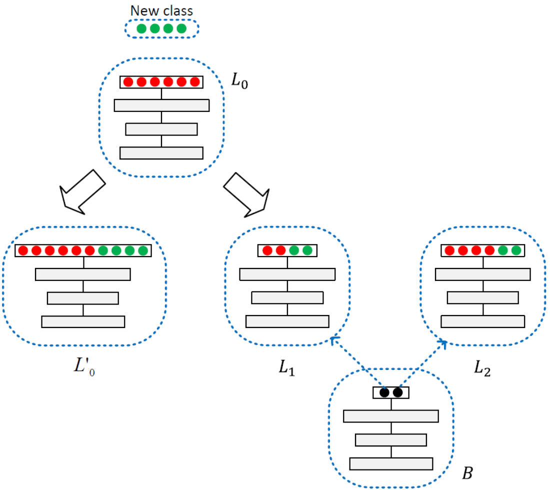

Figure 3 : Two choices of capacity increment. Left: Flat increment, Right: Clone increment.

Starting from a Single Superclass

How to partition a superclass

1. A validation set of N_0 are tested through L_0, and calculating a confusion matrix $C \in \mathbb{R}^{N_0 \times N_0}$ from the output.
   Then, C_{ij} denotes the probability that the i-th class is predicted to j-th class, which also measure the similarity between class i and j.
2. Use spectral clustering partition to split N_0 classes into K clusters based on the confusion matrix.
3. $N_1 - N_0$ new classes are assigned to superclasses based on their confusion rates among the superclasses.

Main algorithm

Algorithm 2: IncrementalLearning

input $(\mathcal{S}, \mathcal{L}, \mathcal{B}, \mathcal{S}_{\text{new}})$: superclass set $\mathcal{S}$, leaf model set $\mathcal{L}$ (each $l \in \mathcal{L}$ is corresponding to a $s \in \mathcal{S}$), branch model set $\mathcal{B}$, new class set $\mathcal{S}_{new}$

output $(\mathcal{S}, \mathcal{L}, \mathcal{B})$: updated superclass set $\mathcal{S}$, leaf model set $\mathcal{L}$, branch model set $\mathcal{B}$

```
/* ditribute new classes to superclasses* /
calculate the confusion matrix Φ with entry Φ(c, s) for
probability of predicting c ∈ S_new to s ∈ S
for all c ∈ S_new do
   select s ∈ S with maximum Φ(c, s)
   s = s ∪ {c}
end for
/* incremental training */
for all s ∈ S and the corresponding l ∈ L do
   (l', b, l_1, l_2, ..., l_K) = ExtendLeafModel(s, l)
   if b ≠ ∅ then
      insert b to B, replace l by {l_1, l_2, ..., l_K} in L
   else
      replace l by l' in L
   end if
end for
/* refine brach models (optional) */
for all b ∈ B do
   incrementally train b according to updated subtrees
end for
return (S, L, B)
```

Experiments

- Dataset: In ImageNet_1K, the dataset include all the 398 animal classes (training set: 501K images, validation set: 18K images).
- To create an incremental training process, the dataset is incremented from 195 randomly drawn classes to 398 classes.

Training	Epochs	Error Rate
from-scratch	41	38.6%
incremental	10	41.6%
incremental	20	39.2%
incremental	30	37.9%
incremental	40	**36.8%**