Maxout Networks

Author : Jaeseok Huh @ Hanyang University
Date : Jan 29, 2018

Comment

- Cited 993 times as for now
- This paper assumes that current g. mean approximation is plausible without mathematical explanation and insists their work is valid because the result bears a resemblance to the result of g. mean approximation – *Figure 7*
- It implies that ReLU is specified version of maxout – one of the rectifier is merely 0, which hindes backprops. to be steered to flow to lower layers without diminishing or being deactiviated

Abstract

- Tried to empirically improve **dropout**
- ***maxout*** – natural companion of dropout

1. Introduction

- Dropout is generally viewed as indiscriminately applicable to almost any model with modes improvement in model
- ***maxout*** is beneficial both for optimization and model averaging
- Empirically in the real world, it is far from ideal SGD

2. Review of dropout

- Similar to bagging (Breiman, 1994)
- For the ensemble to make a prediction by averaging together, most prior works used an inexpensive geometric mean – $softmax(v^T W/2 + b)$ – exactly holds in case of a single layer
- Even in multilayer perceptrons, it performed well in practice, though the approximation has not been characterized mathematically

3. Description of maxout

Given an input $x \in \mathbb{R}^d$, a maxout hidden layer implements the function

$$h_i(x) = \max_{j \in [1,k]} z_{ij}$$

where $z_{ij} = x^T W_{...ij} + b_{ij}$ and $W \in \mathbb{R}^{d \times m \times k}$ and $b \in \mathbb{R}^{m \times k}$ are learned parameters

- Maxout networks learn not just the relationship between hidden units, but also the activiation function of each hidden unit.
- Maxout is locally linear almost everywhere, whereas popular activation functions have significant curavture
- Robust, excellent performance, easy to train

4. Maxout is a universal approximator

MLP(Multilayer perception)

3+ layers with nonlinear activation function

Proposition 4.2

Let C be a compact domain $C \in \mathbb{R}^n$, $f : C \to \mathbb{R}$ be a *continuous function*, and $\epsilon > 0$.

$$\exists\, g : C \to \mathbb{R}\ s.t.\ \forall v \in C,\ |f(v) - g(v)| < \epsilon$$

where g is a *continuous PWL function*

In general as $\epsilon \to 0$, we have $k \to \infty$

5. Benchmark results

5.1 MNIST

5.2 CIFAR-10

5.3 CIFAR-100

5.4 Street View House Numbers

- Color images collected by *Google Street View*
- Used 2nd format – 32 x 32 size, the task is to classify the digit in the center of the images

6. Comparison to rectifiers

- Whether pre-processed or larger models? As same as the comparison target
- Rectifier units do best without cross-channel pooling but with the same number of filters, meaning that **the size of the state and # of param must be about k times higher for rectifiers** to obtain generalization performance approaching that of maxout

7. Model averaging

- Dropout training encourages maxout units to have large linear regions around inputs that appear in the training data

KL divergence (Kullback–Leibler divergence)

- a measure of how one probability distribution diverges from a second, expected probability distribution.
- also called relative entropy

8. Optimization

- Compared to max-pooling, maxout does not include a 0 in the max. Including 0 is harmful – MNIST, $1.04\% \rightarrow 1.2\%$

8.1 Optimization experiments

Maxout optimization degrades gracefully with depth but pooled rectifier units worsen at 6 layers and dramatically at 7

8.2 Saturation

- Rectifier units transit frequently from pos. to 0 than the opposite transition, resulting a preponderance of 0 activations
- Maxout units freely move between positive and negative signs at roughly equal rates.

8.3 Lower layer gradients and bagging

- The variance of the gradient on the output weights was 1.4x for maxout, on the first layer it was 3.4x, namely that maxout does propagate well to lower layers parameters

9. Conclusion

- It emulates bagging training. Can go deeper with maxout and dropout