Sorry for some docstyle, the markdown can't render into pdf perfectly

GoogLeNet - Evgeny Sorokin

Key contribution of the paper

New proposed solution for deep convolutional networks (later DCN).
Key problems of DCN: - More layers - more parameters ~ overtfitting
- Amount of computations: `... uniform increase in the number of their filters results in a quadratic increase of computation.`
- Gradient vanishing

How to solve it: - `Auxiliary classifiers` in order -- reduce gradient vanishing
- Usage of 1x1 non-linear convolutions after each CONV layer (`Network in Network` paper) -- reduce compuitations
- `Inception architecture` reduces locality of solution because of using `1x1, 3x3, 5x5, MP` CONVs outputs vector instead of only one `3x3` CONV (for example)

Key differences with the state of the art

State of the art: VGGNet

- VGG:
  - 19 layers
  - Classic FC layers
  - Fixed conv at each layer
- GoogleNet:
  - 22 layers
  - Usage of `Global Average Pooling`
  - Usage of `Inception modules`

Method explanation

Network consists of 22 layers which moves from classic CONV sequence into new one.
1) In the beginning it convolves the initial image the usual way (before the first Inception).

2) In order to get rid of locality of solution it uses `Inception architecture` which combines results of convolutions of different filter sizes (1x1, 3x3, 5x5 and MP)
- In order to reduce the amount of computations the 1x1 non-linear CONV sublayer is used (presented in NiN paper)
- Filter concatenation block combines the results of those convolutions into a new vector

3) Repeat the process several times and hope everything would be fine
- As it is a Deep Conv Network, the problem of gradient vanishing arises: further layers are not

learning at some point anymore. The solution is to use `Auxiliary classifiers` during the training. The result error would be to next layers.
There are 2 `Auxiliary classifiers` in the network
- **Architecture**
- 1x1 non-linear CONV
- 2 FC
- Softmax

4) End layers
- **Architecture**
- Global Average Pool
- FC
- Softmax
- **Justification**
- The GAP allows to reduce the amount of calculations and (!!!) avoid overfitting (as there will be no weights). The Average Pooling directly connects feature maps into next layer. *(in NiN paper it was connected directly to Softmax)*

Key results

- **first place**: top-5 error 6.67% (ILSVRC 2014 classification challenge)
- **first place** mAP: 43.9% (ILSVRC 2014 detection challenge)

GoogleNet introduced a new idea of CNN: grow in width, not only in depth.
The idea of intermediate evaluations (*auxiliary classifiers*) and non-fixed convolutions(*Inception modules*) provided better results than usual.

ResNet - Evgeny Sorokin

Key contribution of the paper

Microsoft team produced a solution for very deep neural networks.
As we already know - the biggest challenge is gradient vanishing/exploding. The proposed solution for such cases is new such formula, that will allow to overwhelm the gradient problem.

Key differences with the state of the art

State of the art: VGGNet & GoogleNet

- VGG:
    - 19 layers
    - Classic FC layers

    - Fixed conv at each layer
- GoogleNet:
    - 22 layers
    - Usage of `Global Average Pooling`
    - Usage of `Inception modules`
- ResNet:
    - 152(!) layers
    - New solution for gradient vanishing (shortcuts)

Method explanation

The idea is next: we met a problem of gradient vanishing on a **plain networks**, let's refactor the formula a new way.
- The update value on a next layer `L(x) = F(x)` *F is an activation function over input x*
- Becomes `L(x) = F(x) + Ws*x` *Ws is a weight vector (allows to keep dimensions the same)*

Such a way allows to get rid of vanishing gradient problem and train the model even up to 1k layers. This soltion originally called in a paper a `shortcut`

Key results

- ResNet-34 baseline had only 3.6 billion FLOPs (18% of VGG-19 - 19.6 billions)
- ResNet-152: 11.3 billion FLOPs
- 3.57 top-5 err. on ImageNet (ResNet-152)
- ResNet-110 on CIFAR-10 showed 6.43% error.

As for me, the idea kept to make it still simple (plain). The GoogleNet introduced a new solution - wide networks instead of deep. But ResNet prooved, that deeper networks move beneath linearity better, it just needs some help to deal with gradient.