Spatial Transformer Networks

Kaichun Mo

Shanghai Jiao Tong University

\textit{daerduomkch@sjtu.edu.cn}

July 28, 2015
Overview

1. Spatial Transformer Network

2. Caffe: A popular deep learning framework

3. Future Work
Start from the interesting part:
Spatial Transformer Network!
Spatial Transformer Network: Ideal Images
Spatial Transformer Network: Ideal Images

large
well-centered
similar size
straight/right
Spatial Transformer Network: ideal Images

Small intra-class difference
Spatial Transformer Network: ideal Images

Large inter-class difference
Spatial Transformer Network: Real Images

small

NOT well-centered
different size

rotated
Spatial Transformer Network: Spatial Variance

Large intra-class difference
BAD!!!
Introduction: Spatial Transformer Layer

- A new type of layer that is designed to preserve the spatial invariance.

It is expected to be more powerful than **pooling layer**.
- **Localisation Network**: For each input $i$, output its specific transform matrix $\theta_i$.
- **Grid generator and Sampler**: Compute the transformed result using $\theta_i$. 
Localisation Network: For each input image, try to learn the specific transformation matrix $\theta_i$ by which it was deformed.

After learning that, we can reverse it as possible as we can.
**Spatial Transformer Network: Algorithm**

- **Localisation Network:** For each input $i$, output its specific transform parameter $\theta_i$.

- **Grid generator and Sampler:** Compute the transformed result using parameter $\theta_i$. 

---

 Kaichun Mo (SJTU@Cornell)  
 STN  
 July 28, 2015  13 / 29
Spatial Transformer Network: Grid Generator

- For each input, using **localisation network**, we know the transformation matrix to reverse the deformation.
- Then, we need to perform this matrix on the input image by first generating the grid.
Some projected point on the input image may not be sampled.

We need to use \textbf{interpolation technique} to sample it.
- Ex. bilinear.
**Spatial Transformer Layer: Algorithm**

- **Localisation Network**: For each input $i$, output its specific transform matrix $\theta_i$.
- **Grid generator and Sampler**: Compute the transformed result using $\theta_i$. 
Spatial Transformer Network: Differentiability

Forward: inference $f_W(x)$

Backward: learning $\nabla f_W(x)$

“espresso” + loss
In order to learn the weights in **localisation network** and perform **back propagation** during training, we need the forwarding function to be **differentiable**.

Fortunately, it is, at least in sense of **subgradient**!

More detail in paper.
Spatial Transformer Network: Advantages

- **Efficiency**: highly localized $\rightarrow$ highly parallelizable $\rightarrow$ GPU acceleration.

- **End-to-end training**: can be seamlessly incorporated into neural network $\rightarrow$ no pre-training is required.

- **Spatial Invariance**: Make neural network to be less vulnerable to spatial transformations.
Introduction to Caffe!
Caffe

Deep learning framework
by the BVLC

Created by
Yangqing Jia
Lead Developer
Evan Shelhamer

View On GitHub

Caffe

Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. Yangqing Jia created the project during his PhD at UC Berkeley. Caffe is released under the BSD 2-Clause license.

Check out our web image classification demo!

Why Caffe?

Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices.

Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models.

Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning. We believe that Caffe is the fastest convnet implementation available.

Community: Caffe already powers academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia. Join our community of brewers on the caffe-users group and Github.
Why Caffe? In one sip…

- **Expression**: models + optimizations are plaintext schemas, not code.
- **Speed**: for state-of-the-art models and massive data.
- **Modularity**: to extend to new tasks and settings.
- **Openness**: common code and reference models for reproducibility.
- **Community**: joint discussion and development through BSD-2 licensing.
Why Caffe? In one sip...

- **Expression**: models + optimizations are plaintext schemas, not code.
- **Speed**: for state-of-the-art models and massive data.
- **Modularity**: to extend to new tasks and settings.
- **Openness**: common code and reference models for reproducibility.
- **Community**: joint discussion and development through BSD-2 licensing.
Net

- A network is a set of layers connected as a DAG:

  ```
  name: "dummy-net"
  layers { name: "data" ... }
  layers { name: "conv" ... }
  layers { name: "pool" ... }
  ... more layers ...
  layers { name: "loss" ... }
  ```
Layer

```plaintext
name: "conv1"
type: CONVOLUTION
bottom: "data"
top: "conv1"
convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
        type: "xavier"
    }
}
```

- **name**, **type**, and the connection structure (input blobs and output blobs)

- layer-specific parameters
Solving: Training a Net

Optimization like model definition is configuration.

```plaintext
train_net: "lenet_train.prototxt"
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
max_iter: 10000
snapshot_prefix: "lenet_snapshot"

> caffe train -solver lenet_solver.prototxt -gpu 0
```

All you need to run things on the GPU.
Future Work

- Finish implementing **Spatial Transformer Layer** on **Caffe**.
- Test its performance on different vision tasks.
- This layer should be powerful whenever images are not spatially aligned and attention or localisation is necessary.
References

- Official Website: http://caffe.berkeleyvision.org/
- Official Tutorial: DIY Deep Learning for Vision with Caffe
- Spatial Transformer Networks, by Max Jaderberg, et. al.
Thank you for listening!
Q&A