Spatial Transformer Networks

1 Citation

Jaderberg, Max, Karen Simonyan, and Andrew Zisserman. "Spatial transformer networks." Advances in neural information processing systems. 2015.

http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf

2 Abstract

The Spatial Transformer is a module allows the neural network to learn how to transform its input spatially to be invariant to rotation, scale, and more.

3 Introduction

Max-pooling helps networks be somewhat spatially invariant, but relies on having many convolutional layers before hand. It is not general purpose.

The spatial transformer aims to fix this and can select a transformation based on the input. It can help image classification by cropping and scaling the relevant part of the image. It can also localize multiple objects in an image. You can also use it as an attention mechanism - no reinforcement learning required!

4 Related Work

There has been some other work around learning spatial transformations and attention mechanisms, but no module that does this in a differentiable way.

5 Spatial Transformers

We take an input feature map and warps it to make an output feature map. Each channel is warped the same way. There are three pieces. The first is the localization network, which uses some layers to compute the parameters of the transformation. The second is the grid generator, which creates a sampling grid that shows where to sample from the input image in order to produce the output image. The third is the sampler, which actually does the sampling and makes the output feature map. Let's consider each in detail.

The localization network is $\theta = f_{loc}(U)$ where $U \in \mathbb{R}^{H \times W \times C}$ and θ is the parameter vector for the spatial transformation.

To make an output pixel, you apply a sampling kernel centered at some input pixel. The output pixels lie on grid $G = \{G_i\}$ for $G_i = (x_i^t, y_i^t)$ to form output map $V \in \mathbb{H}' \times \mathbb{W}' \times \mathbb{C}$. Supposing our

transformation, T_θ is affine, then the transformation is (for source and transformed coordinates, each of which lie on $[-1, 1]$):

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{12} & \theta_{13} \end{pmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

If you combine $T_\theta(G)$ with an input map U, you get an output map V.

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y)$$

$\forall i \in \{1...HW\}$ and $\forall c \in \{1...C\}$ where k is the sampling kernel (and Φ_x and Φ_y parametrize the kernel). For example, bilinear sampling has: $k(x) = \max(0, 1 - |x|)$. The partial derivatives are:

$$\frac{\partial V_i^c}{\partial U_{nm}^c} = \sum_n^H \sum_m^W \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|)$$

$$\frac{\partial V_i^c}{\partial x_i^s} = \sum_n^H \sum_m^W U_{nm}^c \max(0, 1 - |y_i^s - n|) \begin{cases} 0 & |m - x_i^s| \geq 1 \\ 1 & m \geq x_i^s \\ -1 & m < x_i^s \end{cases}$$

$\frac{\partial V_i^c}{\partial y_i^s}$ is similarly defined. There are discontinuities, so you must use sub-gradients, but this works pretty well on a GPU.

If you take the localization network, grid generator, and sampler together, you get a spatial transformer that you can put anywhere in your neural network. In addition to feeding the output map to the rest of the network, you can also feed the θ parameters. You can also set $H' \times W'$ to be smaller than $H \times W$ to downsample an image. You can use multiple spatial transformers in parallel to extract multiple objects in the image.

6 Experiments

We evaluated on distorted MNIST, distorted Street View House Numbers, and bird classification.

We distort MNIST with rotation, scale, elastic warping, etc. Our baseline is a FCN and CNN (with two max pooling layers). We test spatial transformer with FCN and CNN (bilinear sampling and three transformation types - affine, projective, and thin plate spline). We optimize with SGD with scheduled learning rate drops on multinomial cross entropy loss. The spatial transformer always improves the network and the thin plate spline is the best.

Street View House Numbers has 200K house number images from Google Street View. We take a base CNN and put a spatial transformer right after the image and some deeper spatial transformer. Spatial transformers do the best.

For the birds dataset, we use an inception model as the baseline. We use 4 parallel spatial transformers (each identifies a part of the bird) and pass that into another sub-network, which then feeds into a softmax layer. Each transformer learns to detect something different (e.g. head, body).

7 Conclusion

The spatial transformer is a differentiable module you can throw into your CNN to learn how to spatially transform an input map to focus on the parts that matter.