Cartesian Convolution Network

Wenhao Zhang
National Institute of Materials Science, Tsukuba

I. Standard convolution network

Structure of convolution network

Convolution networks are usually networks that contain at least one convolution layer. General convolution network contain stacked components of convolution, activation and pooling. We can summarize three important properties of the convolution neural network (CNN) as follows:

1) Space interaction: The connection between the input and output are sparse, in contrast to a fully connected network.
2) Parameter sharing: The weights in the convolution are shared. Therefore, CNN's are very efficient in storage and reduced amount of parameters
3) Equivariant representation: Due to weight sharing, convolution operations are equivariant to translation.

In the following, we define the components of a standard convolution neural network.

Convolution

We call convolution operation as:

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da \tag{1}$$

the first argument, $x(a)$, to the convolution is known as *input* and the second argument, w, as the *kernel*. The output of the convultion is referred to as the *feature map* In the case of two dimensional image I, we need to use a two-dimensional kernel K:

$$S(i,j) = (I*K)(i,j) = \sum_m \sum_n I(m,n)K(i-m,j-n) \tag{2}$$

Convolution is commutative, so that

$$\begin{aligned} S(i,j) &= \sum_m \sum_n I(m,n)K(i-m,j-n) &(3)\\ &= \sum_m \sum_n I(i-m,j-n)K(m,n) &(4)\end{aligned}$$

from the change of summation $m \to i-m$ and $n \to j-n$. The second form ($\sum_m \sum_n I(i-m,j-n)K(m,n)$) is frequently used in software implementation because the range of m and n for the kernel function $K(m,n)$ can be reduced in the second expression.

We can also define *cross-correlation* as follows:

$$\begin{aligned} S(i,j) &= (I \star K)(i,j)\\ &= \sum_m \sum_n I(i+m,j+n)K(m,n) &(5)\end{aligned}$$

This is often used in practice.

Activation

For standard convolution function, varies activation function $h\colon \mathbb{R} \to \mathbb{R}$ can be used.

Pooling

A pooling layer extract a summary statistic of nearby outputs at a certain location. Different pooling can be used, such as average pooling and max-pooling. In all case, pooling helps to make the feature map approximately invariant to small translation of the input in that region by downsampling the information. Similarly, pooling over inputs from rotated filter will produce results that are invariant to the rotations. They are illustrated by the following figure.1.

Invariance and equivariance of feature map

Equivariance refers to the property of mapping that perserve the transformation properties. We can formally define a convolution layer in neural network as a mapping $\Phi\colon X \to Y$ that map an input vector (feature map) in vector space X to output feature map vector in vector space Y. We further denote some linear operation $T(T')$ that act on vector space $X(Y)$. Φ is said to be strucutre preserving or equivariant, if we have:

$$\Phi(Tx) = T'\Phi(x) \tag{6}$$

The linear operation together with the vector space are called a representation of a group of transformations. For equivariant mapping, group of T and T' are homomorphism. Therefore, both (T,X) and (T',Y) are representation of the group of transformation.

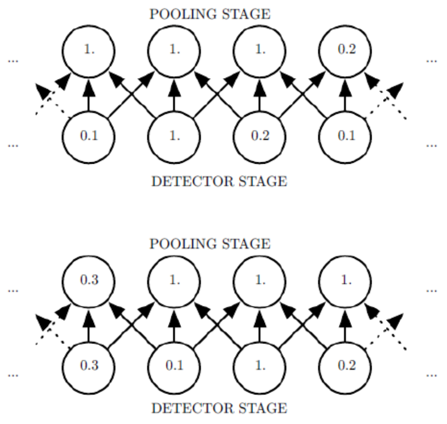
In the case when (T',Y) are identity representation: $T'_g = \mathbf{I}$ for $g \in G$, We have the relationship:

$$\Phi(Tx) = \Phi(x) \tag{7}$$

All linear operation on x corresponding to groups of transformation leave $\Phi(x)$ unmodified.

Is the standard CNN translation invariant or translation equivariant? By constructure, convolution, activation and pooling are translational equivariant so that the convolution neural network is equivariant to translation (complete equivariance is lost when pooling is straded). However, translation invariance is not guaranteed by the convolution neural network, although it can be trained to identifying objects at different locations on the image with good performance. The keypoint to achieve such invariance in the result seems to due to:

1) Effect of pooling with decreasing spatial resolution with higher depth of features (channels) that contain highly extracted information.



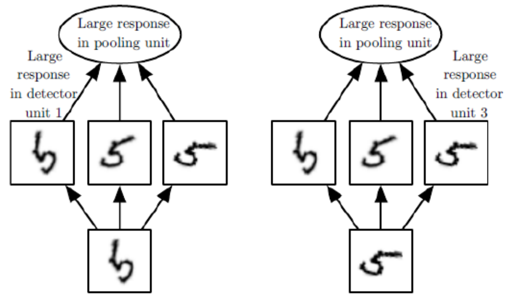

Large response in pooling unit
Large response in detector unit 1
Large response in detector unit 3

Fig. 1. Left: All the inputs of the pooling is shifted left by one. However, the value of two output does not change. This shows that the output of the pooling have a lower resolution than the input and is less sensitive to translation. Furthermore, if we use strades in pooling, more details are lost, i.e., the invariance to small perturbation comes with the loss of spatial resolution (as compensate, depth of feature usually increase). Right: polling over the activation output of transformed filters give resulting feature invariance to these transformations

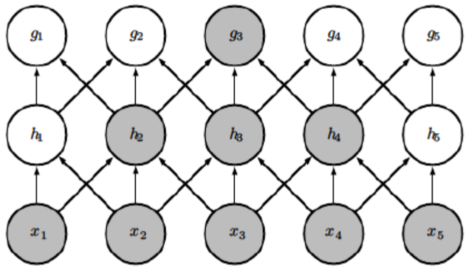


Fig. 2. The unit in deeper networks are indirectly connect to all or most of the image from the gradial convolution, even without pooling

2) The larger field of reception at deeper layer, shown in Figure.2.

But such invariance is not built-in in the CNN, but is likely to be a trained properties[E. K. Abrams, 2017].

II. Group equivariant convolution networks

General consideration

In the standard convolution networks, each components are translational equivariant, meaning that the feature maps transform in the same way as the input image (Equation.(6)). Now, we extand convolution network to deal with general transformation, other than translation[Cohen and Welling, 2016].

We consider the group of transformation G with group elements g that can transform an image. We define an mages and stacks of feature maps as functions $f\colon \mathbb{Z}^2 \to \mathbb{R}^K$, which a a function that map (x, y) position of a figure to K real numbers. For example, for an RGB color image. $K = 3$ and we have three real value at a given pixel.

As a concrete example for transformations, we consider that the input image are subject to transformation of operation in plane group $p4$, which contain operation of 4-fold rotation around the origin and translations. we can parameterize any operation with integer number r, u and v as:

$$g(r,u,v) = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where r specifies four fold rotation and $t = (u, v)$ specifies translations. Operation of group element on a vector $x = (i, j)$ is:

$$gx = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \tag{9}$$

The transformations form a group, i.e., successive combination of transformations in the group yield another transformation belong to the same group. This can be directly verified by matrix multiplication with Equation.(8). We can say that the matrix of group operation and the corresponding vector space (i, j) form a representation of the group $p4$.

Formally, we define the transformation of feature maps as

$$[L_g f](x) = f(g^{-1}x) \tag{10}$$

Where L_g denote transformation operation of the feature map corresponding to transformation g. An illustration of the above definition is Figure. 3, which is a one-dimensional case with feature map given as $f\colon \mathbb{Z} \to \mathbb{R}$. The feature map is translated to the right by $t = 2$, which correspond to the feature map $f(x - 2)$

We use l as index for layers in the network. In each layer, we take an input feature maps (K^l channels) and correlate it with a set of K^{l+1} filters. We define a filter function $\psi\colon \mathbb{Z}^2 \to \mathbb{R}^{K^l}$ (the dimensional of input and output of the filter is the same as that of the feature map). Filters act on feature map as:

$$[f * \psi](x) = \sum_y \sum_{k=1}^{K^l} f_k(y)\psi_k(x - y) \quad \text{convolution} \tag{11}$$

$$[f \star \psi](x) = \sum_y \sum_{k=1}^{K^l} f_k(y)\psi_k(y - x) \quad \text{correlation} \tag{12}$$

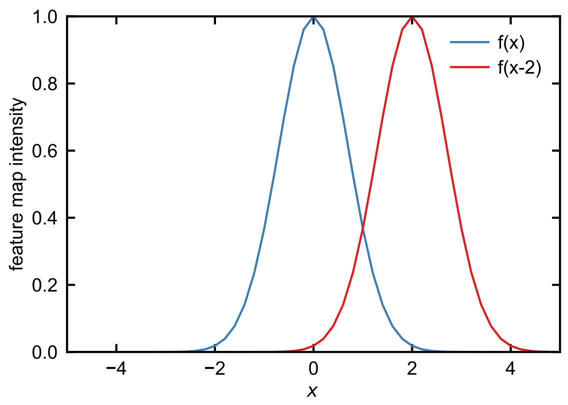


Fig. 3. Apply a shift to resulting feature map is equivalent to a reverse shift in x

again, x and y are tuple denoting positions on the image. We consider the case of correlation in the following.

Proposition 1. *Correlation defined in Equation.*(12) *is equivariant with respect to translation.*

Proof. For a translation $t = (u, v)$, omitting the channel index k, we can show that

$$\begin{aligned}[[L_t f] \star \psi](x) &= \sum_y f(y-t)\psi(y-x) \\ &= \sum_y f(y)\psi(y-x+t) \\ &= \sum_y f(y)\psi(y-(x-t)) \\ &= [f \star \psi](x-t) = [L_t[f \star \psi]](x) \qquad \square\end{aligned}$$

where the symbol $[[L_t f] \star \psi](x)$ means we perform correlation on the transformed feature map $L_t f$ and $[L_t[f \star \psi]](x)$ means that we first perform correlation $[f \star \psi]$, than we perform a translation operation on the resulting feature map. However, such correlation is not equivariant with respect to rotation C_4. This is shown in Figure.4, where the results of $[[L_r f] \star \psi](x)$ and $[L_t[f \star \psi]](x)$ are different.

G-correlation

We define a correlation with respect to a group operation $g \in G$ (G-correlation) as:

$$[f \star \psi](g) = \sum_h \sum_{k=1}^{K^l} f_k(h)\psi_k(g^{-1}h) \tag{13}$$

where $h \in G$. We can see that the standard convolution Eq.(12) is a special case:

$$[f \star \psi](x) = \sum_y \sum_{k=1}^{K^l} f_k(y)\psi_k(y-x) \tag{14}$$

interpreting x, y as a translation operation. Now, the feature map is defined on discrete trasnformation $f: g \to \mathbb{R}^{K^l}$.

Proposition 2. *G-correlation is equivariant to group operation $g \in G$.*

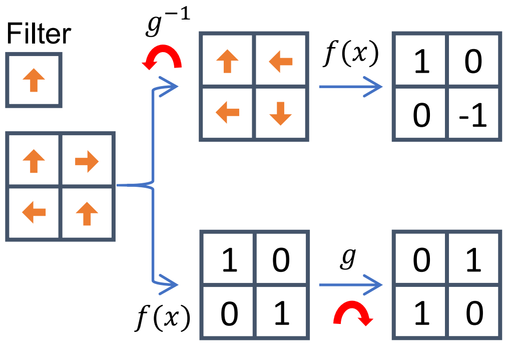


Fig. 4. Upper($[[L_r f] \star \psi](x)$): we rotate the inputs, then perform correlation (rotating the feature map is equivalent to rotating the input with a reversed operation g^{-1}). Lower($[L_t[f \star \psi]](x)$): we first perform correlation, then rotate the resulting feature map

Proof.

$$\begin{aligned}[[L_u f] \star \psi](g) &= \sum_h f(u^{-1}h)\psi(g^{-1}h) \\ &= \sum_h f(h)\psi(g^{-1}uh) \\ &= \sum_h f(h)\psi((u^{-1}g)^{-1}h) \\ &= [f \star \psi](u^{-1}g) = [L_u[f \star \psi]](g) \qquad \square\end{aligned}$$

Nonlinearity

We define a nonlinearity $v: \mathbb{R} \to \mathbb{R}$ and write:

$$C_v f(g) = v(f(g)) \tag{15}$$

i.e. ,the nonlinearity is applied on the feature map. The combined operation satisfy Eq.(6):

$$C_v L_h f(g) = L_h C_v f(g)$$

. Therefore, nonlinear activation is equivariant and does not change the transformation properties of the feature map.

Pooling

In the standard convolution network, a max-pooling is defined by an operation P:

$$Pf(x) = \max_{y \in xU} f(y) \tag{16}$$

where $xU = xu \mid u \in U$ and U is the pooling domain typically a set of small translation (2×2 or 3×3) square around the origin. Similarly, we define a pooling for feature map over G as:

$$Pf(g) = \max_{k \in gU} f(k) \tag{17}$$

with the pooling domain a subgroup $U \subset G$. The pooling domains gU therefore partition the group G into left cosets. It is shown that pooling commutes with group operation and therefore is equivariant. Pooling are reductions, therefore, the feature map after the pooling over cosets will be defined on the quotient groups G/U.

For example, for plane group $p4$, the group of all four rotations around a lattice points R (point group) are a subgroup of G, The quotient group $p4/R$ is isomorphic to the translation of $p4$ and the rotation of the input feature map will have no effect on the output feature map. This is because a the coset is invariant to its elements: $rR = R$ for $r \in R$ and the feature map over r is poolled over. This is similar to the right panel in Figure.1.

III. Spherical convolution network

Previous section extends the definition of convolution(correlation) to transformations operation of finite group. Now we consider extending it to continuous transformation[Cohen et al. 2018] in three dimensional case.

We consider an image on a unit sphere. We define the image on a sphere as function $f\colon S^2 \to \mathbb{R}^K$. S^2 is the set of points $x \in \mathbb{R}^3$ on the unit sphere with position specified by (α, β) with $\alpha \in [0, 2\pi]$ and $\beta \in [0, \pi]$. However, three dimension rotation operation R of the image on the sphere is specified by three Euler angles α, β and γ, $\gamma \in [0, 2\pi]$, as shown in Figure.5. An important difference between the rotation of a sphere and translation or rotation on a lattice is that rotation group SO(3) is a continuous group.

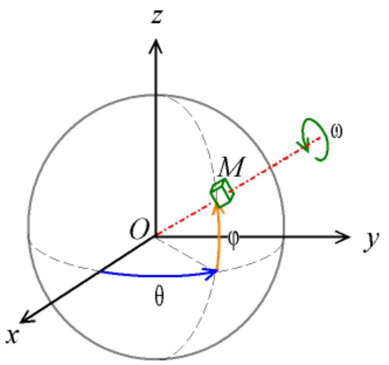


Fig. 5. A point on a unit sphere is specified by two parameter, while an arbitrary rotation in three dimension is specified by three Euler angles

We

In this short note, the structure of the euclidean neural network is presented.

Definition 1. *(Equivariant) A mapping $L\colon \mathcal{X} \to \mathcal{Y}$ between two vector space $\mathcal{X}$ and $\mathcal{Y}$ is called equivariant with respect to a group G and group representation $D^{\mathcal{X}}$ and $D^{\mathcal{Y}}$ if*

$$L \circ D^{\mathcal{X}} = D^{\mathcal{Y}} \circ L$$

If we have $L \circ D^{\mathcal{X}} = L$, then the mapping L is called invariant, i.e., it map vector space $\mathcal{X}$ to a vector space $\mathcal{Y}$, in which the group operations are identity: $D^{\mathcal{Y}}(g) = \mathbf{1}$ for all $g \in G$.

In euclidean neural network, each layer of the neural network perform one of the equivariant mapping. The inputs for the layer are denoted by $(\mathbf{r}_a, \mathbf{s}_a)$ in the vector space $R^3 \oplus \mathcal{S}$ and the output is $(\mathbf{r}_a, \mathbf{s}'_a)$ vector space is $R^3 \oplus \mathcal{S}'$, where $\mathbf{s}_a$ are a feature vector in feature space $\mathcal{S}$, representing a point in the euclidean space R^3 and $\oplus$ indicate direct sum.

We denote the feature vector as a tensor V^l_{acm}, where a is the index of the point, l is the rotational order, $m = -l, \ldots, l$ is the index of the spherical base function. c is the channels index (depth of a convolution). In practical implementation, feature vector are implemented as a dictionary with key l, each item in the dictionary correspond to a multidimensional arrays, each with shape $(|S|, n_l, 2l+1)$ corresponding to the number of points in space, channels and m.

Definition 2. *(Filter) We define a filter to be of the following form:*

$$F^{l_f, l_i}_{cm_f}(\mathbf{r}) = R^{l_f, l_i}_c Y^{l_f}_{m_f}(\mathbf{r})$$

where l_i and l_f is the rotation order of the input and the filter, $m = -l_f, \ldots, l_f$ is the index of the basis function, $Y^{l_f}_m(\mathbf{r})$ is the spherical harmonics and R^{l_f, l_i}_c is learned functions mapping real number to real number and is a learned property.

With the filter function, we define a convolution layer as:

$$V^{l_o}_{acm_o} = L^{l_o}_{acm_o}(\mathbf{r}_a, V^{l_i}_{acm_i}) = \sum_{m_f, m_i} C^{l_o, m_o}_{(l_f, m_f)(l_i, m_i)} \sum_{b \in N} F^{l_f, l_i}_{cm_f}(\mathbf{r}_{ab}) V^{l_i}_{bcm_i} \tag{18}$$

where C is the *Clebsch-Gordan coefficients*

We can also define self-interaction layer as:

$$V^l_{acm} = \sum_{c'} W^l_{cc'} V^l_{ac'm}$$

and nonlinearity as:

$$\begin{aligned} &\eta^0(V^0_{ac} + b^0_c) && \text{for } l = 0 \\ &\eta^(\|V\|^l_{ac} + b^l_c) V^l_{acm} && \text{for } l > 0 \end{aligned}$$

where $\|V\|^l_{ac} = \sqrt{\sum_m |V^l_{acm}|^2}$.