1 Convolutional zonotope transformation

We can compute the zonotope approximation of the convolutional layers by considering each ε separately and computing the convolution of the coefficients corresponding to that ε.

Let I be the unknown input tensor of the convolutional layer, K be the filter and A be I's coefficients in the zonotope approximation. K refers to the filter for a single channel. Neurons for a given layer are indexed by $[x, y]$.

$$(I \times K)[x, y] = \sum_{i=0}^{m} \sum_{j=0}^{m} I[x + i - m/2, y + j - m/2] * K[i, j]$$

But $I[x, y] = \sum_{k=1}^{n} A_k[x, y] * \varepsilon_k + A_0[x, y]$. So:

$$(I \times K)[x, y] = \sum_{i=0}^{m} \sum_{j=0}^{m} (\sum_{k=1}^{n} A_k[x+i-m/2, y+j-m/2] * \varepsilon_k + A_0[x+i-m/2, y+j-m/2]) * K[i, j]$$

Now I distribute the multiplication with the filter into the sum and change the order of the summations to get:

$$(I \times K)[x, y] = \sum_{k=1}^{n} \sum_{i=0}^{m} \sum_{j=0}^{m} A_k[x+i-m/2, y+j-m/2] * K[i, j] * \varepsilon_k + A_0[x+i-m/2, y+j-m/2] * K[i, j]$$

And now I take out the ε from the inner summation to get:

$$(I \times K)[x, y] = \sum_{k=1}^{n} (\sum_{i=0}^{m} \sum_{j=0}^{m} A_k[x+i-m/2, y+j-m/2] * K[i, j]) * \varepsilon_k + \sum_{i=0}^{m} \sum_{j=0}^{m} A_0[x+i-m/2, y+j-m/2] * K[i, j]$$

As we can see from the equation, we have a zonotype where the center is given by a convolution over the centers of the input zonotope, A_0. While the coefficients of each ε are just a convolution over the coefficients of that ε in the input:

$$(I \times K)[x, y] = \sum_{k=1}^{n} (A_k \times K)[x, y] * \varepsilon_k + (A_0 \times K)[x, y]$$

It suffices to compute the $k + 1$ convolutions of A and K, which can be done efficiently using pytorch.

2 Loss function

Output layer: $[o_1, o_2, ...o_n]$

Zonotope approximation of verification objective (target t):

$$Z = \sum_{i=1}^{n} \max(o_i - o_t) \tag{1}$$

$Z > 0$ only if one or more of the o_i is greater that o_t.

In particular if we compute the upper bound on Z:

$$L = max_{\varepsilon} Z \tag{2}$$

If $L = 0 => o_t >= o_i \forall i$ which is the property that we want to verify.

Else: $L > 0$ and we could minimize L by gradient descent with respect to lambdas.

In order to do that, we could build the entire Zonotope approximation with pythorch tensors and operators and then use it to compute gradients with respect to the lambdas.