1. Multi-Norm Zonotopes

1.1. Classical Zonotope

A classical Zonotope [1] abstracts a set of $N \in \mathbb{N}$ variables and associates the k-th variable with an affine expression x_k using $\mathcal{E} \in \mathbb{N}$ noise symbols defined by:

$$x_k = c_k + \sum_{i=1}^{\mathcal{E}} \beta_k^i \varepsilon_i = c_k + \vec{\beta}_k \cdot \vec{\varepsilon} \tag{1}$$

where $c_k, \beta_k^i \in \mathbb{R}$ and $\varepsilon_i \in [-1, 1]$. The value x_k can deviate from its center coefficient c_k through a series of noise symbols ε_i scaled by the coefficients β_k^i. The set of noise symbols $\vec{\varepsilon}$ is shared among different variables, thus encoding dependencies between N values abstracted by the zonotope.

1.2. Multi-Norm Zonotope Definition

The Multi-norm Zonotope domain [2] extends the classical Zonotope by adding noise symbols ϕ_j that fulfill the constraint $\|\vec{\phi}\|_p \leq 1$, where $\vec{\phi} := \left(\phi_1, ..., \phi_{\mathcal{E}_p}\right)^T$. If $p = \infty$, we recover the classical Zonotope. This new domain allows us to easily express ℓ_p-norm bound balls in terms of the new noise symbols ϕ:

$$x_k = c_k + \sum_{i=1}^{\mathcal{E}_p} \alpha_k^i \phi_i + \sum_{j=1}^{\mathcal{E}_\infty} \beta_k^j \varepsilon_j = c_k + \vec{\alpha}_k \cdot \vec{\phi} + \vec{\beta}_k \cdot \vec{\varepsilon} \tag{2}$$

where $c_k, \alpha_k^i, \beta_k^j \in \mathbb{R}$, $\|\vec{\phi}\|_p \leq 1$, and $\varepsilon_j \in [-1, 1]$.

In the implementation, constants are stored as properties:

- `z.p: int` - Special norm, p
- `z.q: int` - Dual norm of p (see Section 1.4), q
- `z.Ei: int` - Number of infinity norm error terms, $\mathcal{E}_p$
- `z.Es: int` - Number of special norm error terms, $\mathcal{E}_\infty$
- `z.N: int` - Number of zonotope variables, N

The properties are dynamic, for instance `z.Ei` will give the current $\mathcal{E}_\infty$, as it may change during operations.

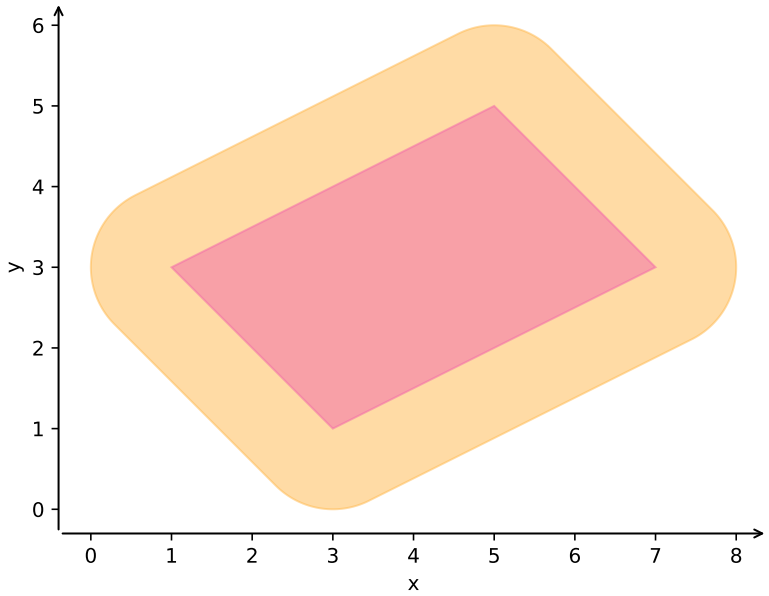


Figure 1: A multi-norm Zonotope with two variables $x = 4 + \phi_1 - \varepsilon_1 + 2\varepsilon_2$, and $y = 3 + \phi_2 + \varepsilon_1 + \varepsilon_2$, where $\|\vec{\phi}\|_2 \leq 1$ and $\varepsilon_1, \varepsilon_2 \in [-1, 1]$. The green region indicates the classical Zonotope obtained by removing the $\vec{\phi}$ noise symbols.

1.3. Matrix Representation

To concisely represent the N zonotope output variables $x_1, ..., x_N$, we write $x = (x_1, ..., x_N)^T$. Therefore, the Multi-norm Zonotope x can be simplified to:

$$x = c + A\vec{\phi} + B\vec{\varepsilon} \tag{3.1}$$

$$c \in \mathbb{R}^N, A \in \mathbb{R}^{N \times \mathcal{E}_p}, B \in \mathbb{R}^{N \times \mathcal{E}_\infty} \tag{3.2}$$

$$\|\vec{\phi}\|_p \leq 1, \varepsilon_j \in [-1, 1] \tag{3.3}$$

where $A_{k,i} = \alpha_k^i$ and $B_{k,j} = \beta_k^j$.

In the implementation, error terms are stored in different tensors:

- `z.W_Es: Float[Tensor, "N Es"]` - Special error terms, $A \in \mathbb{R}^{N \times \mathcal{E}_p}$
- `z.W_Ei: Float[Tensor, "N Ei"]` - Infinity error terms, $B \in \mathbb{R}^{N \times \mathcal{E}_\infty}$
- `z.W_C: Float[Tensor, "N"]` - Bias or center terms, $c \in \mathbb{R}^N$

1.4. Dual Norm

For a given vector $z \in \mathbb{R}^N$, the dual norm $\|z\|_p^*$ of the ℓ_p norm is defined as:

$$\|z\|_p^* = \sup\{z \cdot x \mid x \in \mathbb{R}^N, \|x\|_p \leq 1\} \tag{4}$$

The dual norm $\|z\|_p^*$ is the ℓ_q norm where q satisfies the relationship $\frac{1}{p} + \frac{1}{q} = 1$.

1.5. Computing Concrete Bounds - `z.concretize()`

The tight lower and upper bounds of $z \cdot x$ where $x \in \mathbb{R}^N$ s.t. $\|x\|_p \leq 1$ are given by:

$$l_k^q = -\|z\|_q \tag{5.1}$$

$$u_k^q = \|z\|_q \tag{5.2}$$

Thus the lower and upper interval bounds of the special terms of the zonotopes can be computed as:

$$-\|\vec{\alpha}_k\|_q \leq \vec{\alpha}_k \cdot \vec{\phi} \leq \|\vec{\alpha}_k\|_q \tag{6}$$

Given this, the full lower and upper bounds of the multi-norm Zonotope, l_k and u_k of x_k are:

$$l_k = c_k - \|\vec{\alpha}_k\|_q + \min\left(\vec{\beta}_k \cdot \vec{\varepsilon}\right) = c_k - \|\vec{\alpha}_k\|_q - \|\vec{\beta}_k\|_1 \tag{7.1}$$

$$u_k = c_k + \|\vec{\alpha}_k\|_q + \max\left(\vec{\beta}_k \cdot \vec{\varepsilon}\right) = c_k + \|\vec{\alpha}_k\|_q + \|\vec{\beta}_k\|_1 \tag{7.2}$$

1.6. Sampling a Point from Multi-norm Zonotope - `z.sample_point()`

The sampling procedure consists of two parts. We first sampling the ℓ_p-norm noise symbols by generating points within the ℓ_p-norm unit ball. We then sample the infinity norm noise symbols by generating values in $[-1, 1]$.

1.6.1. Sampling the ℓ_p-norm Noise Symbols

To generate a point within the ℓ_p-norm unit ball:

1. Generate a random vector $v \in \mathbb{R}^{\mathcal{E}_p}$ following a standard normal distribution
2. Normalize v by its p-norm: $v' = \frac{v}{\|v\|_p}$
3. Scale v' by a random factor $r \in [0, 1]$ to ensure coverage of the interior of the ball: $\vec{\phi} = r \cdot v'$

This procedure gives us a random point $\vec{\phi}$ such that $\|\vec{\phi}\|_p \leq 1$.

1.6.2. Sampling the ℓ_∞-norm Noise Symbols

For each ℓ_∞-norm noise symbol ε_j, we simply sample a uniform random value in $[-1, 1]$:

$$\vec{\varepsilon}_j \sim \text{Uniform}(-1, 1) \tag{8}$$

1.7. Noise Symbol Reduction - `z.remove_infinity_errors()`

We follow the DecorrelateMin$_k$ heuristic method [3], that reduces the number of ℓ_∞ noise symbols in a Multi-norm Zonotope to k. The method works as follows:

1. **Score Calculation**: For each ℓ_∞ noise symbol ε_j, we calculate a score m_j representing its significance:

$$m_j = \sum_{i=1}^{N} |B_{i,j}| = \sum_{i=1}^{N} |\beta_i^j| \tag{9}$$

   where $B_{i,j} = \beta_i^j$ is the coefficient of the j-th noise symbol for the i-th variable.

2. **Ranking**: We rank the noise symbols based on their scores and select the top k noise symbols to keep.

3. **Reduction**: We combine the effects of the eliminated noise symbols into a single new noise symbol for each zonotope variable.

Let I denote the indices of the eliminated ℓ_∞ noise symbols and P the indices of the top k ℓ_∞ noise symbols. Then, the new Multi-norm Zonotope is:

$$x = c + A\vec{\phi} + B_P\vec{\varepsilon}_P + \begin{pmatrix} \sum_{j\in I} |\beta_1^j| \\ \dots \\ \sum_{j\in I} |\beta_N^j| \end{pmatrix} \vec{\varepsilon}_{\text{new}} \tag{10}$$

Where:
- $\vec{\varepsilon}_P$ represents the kept noise symbols with indices in P
- B_P contains only the corresponding columns of B
- $\vec{\varepsilon}_{\text{new}} \in [-1, 1]^N$ is the new noise symbol

2. Abstract Transformers - `functional`

2.1. General abstract transformer construction

[4] provide a general method to find sound and minimal area abstract transformers for zonotopes. Sound neuron-wise transformers for the zonotope domaine can be described as:

$$y = \lambda x + \mu + \beta\varepsilon_{\text{new}} \tag{11}$$

For convex C^1 continuous functions, all tangents to the curve of the function yield viable transformers. The resulting parallelogram can be parametrized by the abscissa of the contact point t with $l \leq t \leq u$. Using the mean value theorem and convexity, it follows that there will be a point t_{crit} where the upper edge of the parallelogram will connect the lower and upper endpoints of the graph. For $t < t_{\text{crit}}$ it will make contact on the upper endpoint and for $t > t_{\text{crit}}$ on the lower endpoint. This allows to describe the parameters λ, μ and β of a zonotope transformer for a function $f(x) : \mathbb{R} \to \mathbb{R}$ on the interval $[l, u]$ as:

$$\lambda = f'(t) \tag{12.1}$$

$$\mu = \frac{1}{2}\left(f(t) - \lambda t + \begin{cases} f(l) - \lambda l, & \text{if } t \geq t_{\text{crit}} \\ f(u) - \lambda u, & \text{if } t < t_{\text{crit}} \end{cases}\right) \tag{12.2}$$

$$\beta = \frac{1}{2}\left(\lambda t - f(t) + \begin{cases} f(l) - \lambda l, & \text{if } t \geq t_{\text{crit}} \\ f(u) - \lambda u, & \text{if } t < t_{\text{crit}} \end{cases}\right) \tag{12.3}$$

$$\nabla_x f(x)|_{x=t_{\text{crit}}} = \frac{f(u) - f(l)}{u - l} \tag{12.4}$$

A minimum area transformer can now be derived by minimizing the looseness μ for $l \leq t \leq t_{\text{crit}}$ and $t_{\text{crit}} \leq t \leq u$. This yields the constrained optimization problems:

$$\min_t \frac{1}{2}(f'(t)(t-u) - f(t) + f(u)), \quad s.t,\ l \leq t \leq t_{\text{crit}} \tag{13.1}$$

$$\min_t \frac{1}{2}(f'(t)(t-l) - f(t) + f(l)), \quad s.t,\ t_{\text{crit}} \leq t \leq u \tag{13.2}$$

These can be solved using the method of Lagrange multipliers. Equation 13.1 leads to the following equations:

$$\mathcal{L} = \frac{1}{2}(f'(t)(t-u) - f(t) + f(u)) + \gamma_1(l - t) + \gamma_2(t - t_{\text{crit}}) \tag{14.1}$$

$$\nabla_t \mathcal{L} = \frac{1}{2}f''(t)(t-u) - \gamma_1 + \gamma_2 = 0 \tag{14.2}$$

$$\nabla_{\gamma_1} \mathcal{L} = t - l \tag{14.3}$$

$$\nabla_{\gamma_2} \mathcal{L} = t - t_{\text{crit}} \tag{14.4}$$

$$\gamma_1 \geq 0 \tag{14.5}$$

$$\gamma_2 \geq 0 \tag{14.6}$$

$$\gamma_1(t - l) = 0 \tag{14.7}$$

$$\gamma_2(t - t_{\text{crit}}) = 0 \tag{14.8}$$

Case 1: Neither constraint is active, $\gamma_1 = \gamma_2 = 0$, $\nabla_t \mathcal{L} = f''(t)(t-u) = 0$. Hence, either $t^* = u = t_{\text{crit}}$, or t^* verifies $f''(t^*) = 0$.

Case 2: $\gamma_1 \neq 0, \gamma_2 = 0$, thus $t^* = l$. In this case, $\gamma_1 = \frac{1}{2} f''(l)(l-u)$. However, as f is convex, $f''(x) \geq 0$, so if $u \neq l$, this leads to $\gamma_1 < 0$ which is not possible.

Case 3: $\gamma_1 = 0, \gamma_2 \neq 0$, thus $t^* = t_{\text{crit}}$ and $\gamma_2 = \frac{1}{2} f''(l)(l-u) \geq 0$.

Case 4: $\gamma_1 \neq 0, \gamma_2 \neq 0$. In this case, $t^* = l = t_{\text{crit}}$.

Analogously, equation Equation 13.1 yields a boundary minimum at $t = t_{\text{crit}}$. Consequently $t = t_{\text{crit}}$ yields the minimum area transformer for convex functions. t_{crit} can be computed either analytically or numerically by solving Equation 12.4 as the point where the local gradient is equal to the mean gradient over the whole interval.

2.2. Exponential Transformer

The exponential function has the feature that its output is always strictly positive, which is important when used as input to the logarithmic function to compute the entropy. Therefore, a guarantee of positivity for the output zonotope is desirable. A constraint yielding such a guarantee can be obtained by inserting $\hat{x}_i = l, \varepsilon_{p+1} = -\operatorname{sign}(\mu)$ and $\hat{y}_i \geq 0$ with $\lambda(t) = e^t$ into Equation 11:

$$0 \leq \lambda l + \frac{1}{2}(f(t) - \lambda t + f(u - \lambda u)) - \frac{1}{2}(\lambda t - f(t) + f(u - \lambda u)) \tag{15.1}$$

$$0 \leq \lambda(l - t) + f(t) \tag{15.2}$$

$$0 \leq e^t(l - t + 1) \tag{15.3}$$

$$t \leq 1 + l \equiv t_{\text{crit},2} \tag{15.4}$$

This constitutes the additional upper limit $t_{\text{crit},2}$ on t. Therefore it is sufficient to reevaluate 16 as it will either be inactive in equation 17 if $t_{\text{crit}} \leq t_{\text{crit},2}$ for the solutions computed previously or the constraints will be insatiable ensuring that 17 will have no solutions. If a strictly positive output is required a small delta can simply be subtracted from the upper limit $t_{\text{crit},2}$. It is easy to see that t is now constrained to $\left[l, \min\left(u, t_{\text{crit},2}\right)\right]$ and that the minimum area solution will be obtained with $t_{\text{opt}} = \min\left(t_{\text{crit}}, t_{\text{crit},2}\right)$. The critical points can be computed explicitly to $t_{\text{crit}} = \log\left(e^u - e^l\right)$ and $t_{\text{crit},2} = l + 1$. This can be inserted into equations 11 to 14 to obtain a positive, sound and viable transformer.

2.3. Logarithmic Transformer

The logarithmic transformer can be constructed by plugging $f(t) = -\log(t)$ and $f'(t) = -\frac{1}{x}$ into equations 12 to 14 and their results into equation 11. Equation 15 can be solved to $t_{\text{crit}} = \frac{l-u}{\ln(l) - \ln(u)}$.

2.4. Affine Abstract Transformer

The abstract transformer for an affine combination $z = ax_1 + bx_2 + c$ of two Multi-norm Zonotope variables $x_1 = c_1 + \vec{\alpha}_1 \cdot \vec{\phi} + \vec{\beta}_1 \cdot \vec{\varepsilon}$ and $x_2 = c_2 + \vec{\alpha}_2 \cdot \vec{\phi} + \vec{\beta}_2 \cdot \vec{\varepsilon}$, is:

$$z = ax_1 + bx_2 + c \tag{16.1}$$

$$= a\left(c_1 + \vec{\alpha}_1 \cdot \vec{\phi} + \vec{\beta}_1 \cdot \vec{\varepsilon}\right) + b\left(c_2 + \vec{\alpha}_2 \cdot \vec{\phi} + \vec{\beta}_2 \cdot \vec{\varepsilon}\right) + c \tag{16.2}$$

$$= (ac_1 + bc_2 + c) + (a\vec{\alpha}_1 + b\vec{\alpha}_2) \cdot \vec{\phi} + \left(a\vec{\beta}_1 + b\vec{\beta}_2\right) \cdot \vec{\varepsilon} \tag{16.3}$$

This transformer is exact, as it simply applies the affine operation directly to the Multi-norm Zonotope representation without introducing any over-approximation.

2.5. ReLU Abstract Transformer

The ReLU abstract transformer defined for the classical Zonotope [5] can be extended naturally to the multi-norm setting [2] since it relies only on the lower and upper bounds of the variables, which are computed using the method described for the Multi-norm Zonotope.

For a zonotope variable x with lower bound l and upper bound u, the Multi-norm Zonotope abstract transformer for $\mathrm{ReLU}(x) = \max(0, x)$ is:

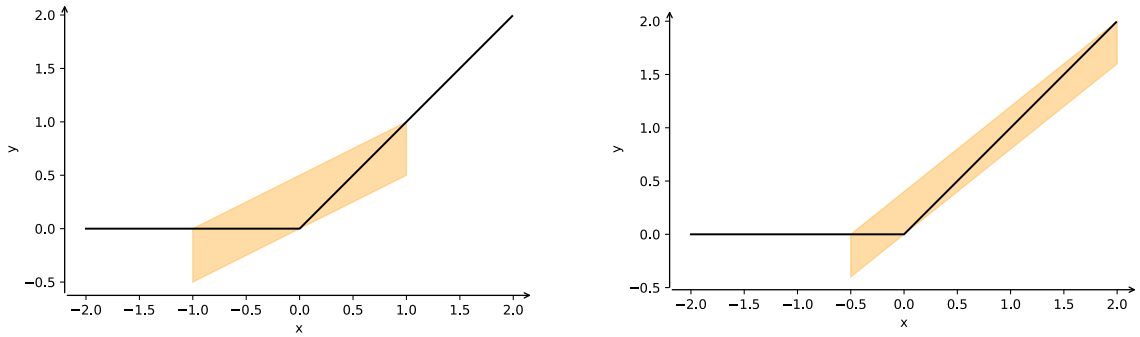
$$y = \begin{cases} 0, & \text{if } u < 0 \\ x, & \text{if } l > 0 \\ \lambda x + \mu + \beta_{\text{new}} \varepsilon_{\text{new}}, & \text{otherwise} \end{cases} \tag{17}$$

where $\varepsilon_{\text{new}} \in [-1, 1]$ denotes a new noise symbol, and:

$$\lambda = \frac{u}{u - l} \tag{18.1}$$

$$\beta_{\text{new}} = \mu = 0.5 \max(-\lambda l, (1 - \lambda) u) \tag{18.2}$$

We note that the newly introduced noise symbol ε_{new} is an ℓ_∞ noise symbol. This holds for all ε_{new} in the following transformers as well.



2.6. Tanh Abstract Transformer

The abstract transformer for the operation $y = \tanh(x)$ is:

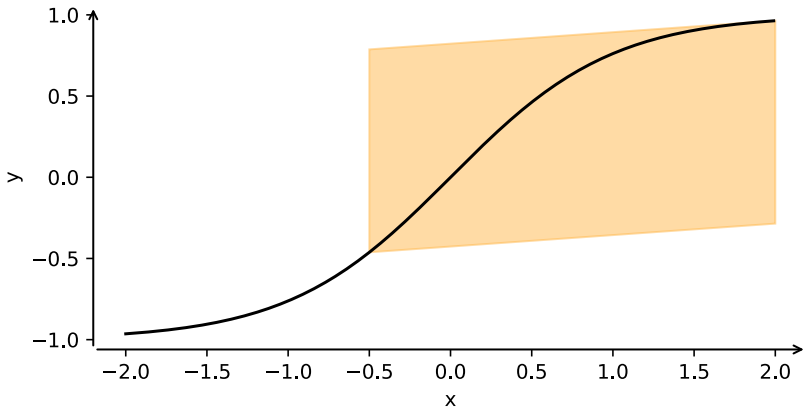
$$y = \lambda x + \mu + \beta_{\text{new}} \varepsilon_{\text{new}} \tag{19}$$

where:

$$\lambda = \min(1 - \tanh^2(l), 1 - \tanh^2(u)) \tag{20.1}$$

$$\mu = \frac{1}{2}(\tanh(u) + \tanh(l) - \lambda(u + l)) \tag{20.2}$$

$$\beta_{\text{new}} = \frac{1}{2}(\tanh(u) - \tanh(l) - \lambda(u - l)) \tag{20.3}$$



2.7. Exponential Abstract Transformer

The operation $y = e^x$ can be modeled through the element-wise abstract transformer:

$$y = \lambda x + \mu + \beta_{\text{new}} \varepsilon_{\text{new}} \tag{21}$$

where:

$$\lambda = e^{t_{\text{opt}}} \tag{22.1}$$

$$\mu = 0.5\big(e^{t_{\text{opt}}} - \lambda t_{\text{opt}} + e^u - \lambda u\big) \tag{22.2}$$

$$\beta_{\text{new}} = 0.5\big(\lambda t_{\text{opt}} - e^{t_{\text{opt}}} + e^u - \lambda u\big) \tag{22.3}$$

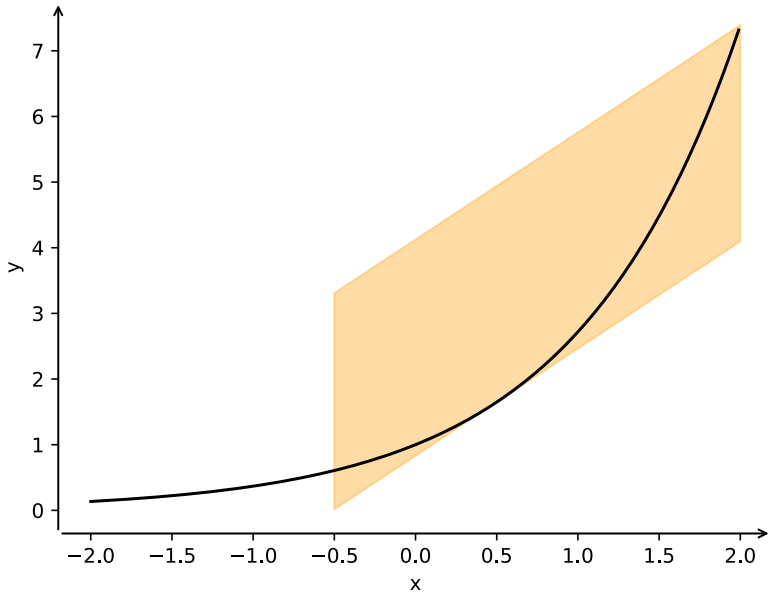
and

$$t_{\text{opt}} = \min\big(t_{\text{crit}}, t_{\text{crit},2}\big) \tag{23.1}$$

$$t_{\text{crit}} = \log\left(\frac{e^u - e^l}{u - l}\right) \tag{23.2}$$

$$t_{\text{crit},2} = l + 1 - \hat{\varepsilon} \tag{23.3}$$

Here, $\hat{\varepsilon}$ is a small positive constant value, such as 0.01. The choice $t_{\text{opt}} = \min\big(t_{\text{crit}}, t_{\text{crit},2}\big)$ ensures that y is positive.



2.8. Reciprocal Abstract Transformer

The abstract transformer for $y = \frac{1}{x}$ with $x > 0$ is given by:

$$y = \lambda x + \mu + \beta_{\text{new}} \varepsilon_{\text{new}} \tag{24}$$

where:

$$\lambda = -\frac{1}{t_{\text{opt}}^2} \tag{25.1}$$

$$\mu = 0.5 \left(\frac{1}{t_{\text{opt}}} - \lambda \cdot t_{\text{opt}} + \frac{1}{l} - \lambda l \right) \tag{25.2}$$

$$\beta_{\text{new}} = 0.5 \left(\lambda \cdot t_{\text{opt}} - \frac{1}{t_{\text{opt}}} + \frac{1}{l} - \lambda l \right) \tag{25.3}$$

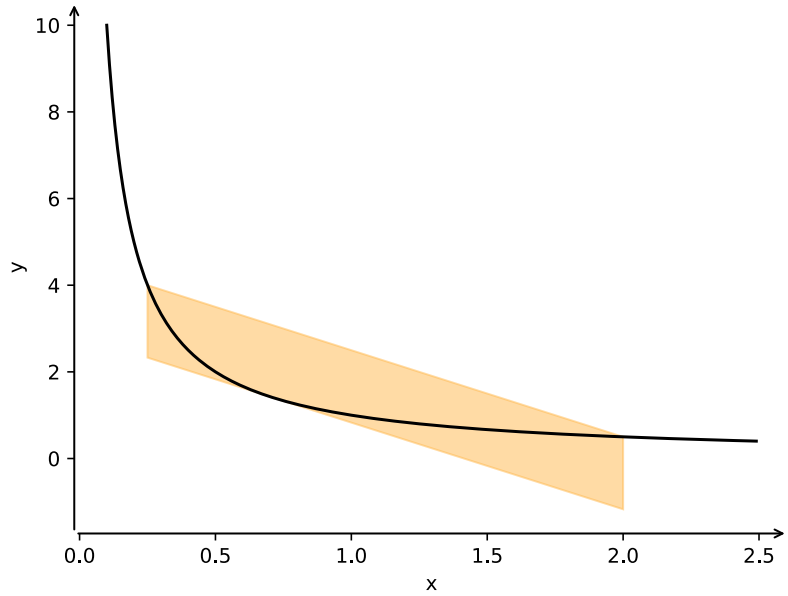
and

$$t_{\text{opt}} = \min\left(t_{\text{crit}}, t_{\text{crit},2}\right) \tag{26.1}$$

$$t_{\text{crit}} = \sqrt{ul} \tag{26.2}$$

$$t_{\text{crit},2} = 0.5u + \hat{\varepsilon} \tag{26.3}$$

Similarly to the exponential transformer, $\hat{\varepsilon}$ is a small positive constant and $t_{\text{opt}} = \min\left(t_{\text{crit}}, t_{\text{crit},2}\right)$ ensures that y is positive.



2.9. Dot Product Abstract Transformer

Next, we define the abstract transformer for the dot product between pairs of vectors of variables of a Multi-norm Zonotope. The transformer is used in the multi-head self-attention, specifically in the matrix multiplications between Q and K and between the result of the softmax and V.

For two Multi-norm Zonotope vectors $\vec{v}_1 = \vec{c}_1 + A_1\vec{\phi} + B_1\vec{\varepsilon}$ and $\vec{v}_2 = \vec{c}_2 + A_2\vec{\phi} + B_2\vec{\varepsilon}$, computing the dot product produces the output variable y:

$$y = \vec{v}_1 \cdot \vec{v}_2 = \left(\vec{c}_1 + A_1\vec{\phi} + B_1\vec{\varepsilon}\right) \cdot \left(\vec{c}_2 + A_2\vec{\phi} + B_2\vec{\varepsilon}\right) \tag{27.1}$$

$$= \vec{c}_1 \cdot \vec{c}_2 + (\vec{c}_1^T A_2 + \vec{c}_2^T A_1)\vec{\phi} + (\vec{c}_1^T B_2 + \vec{c}_2^T B_1)\vec{\varepsilon} + \left(A_1\vec{\phi} + B_1\vec{\varepsilon}\right) \cdot \left(A_2\vec{\phi} + B_2\vec{\varepsilon}\right) \tag{27.2}$$

The last term represents interactions between noise symbols and is not in the functional form of a Multi-norm Zonotope. We first expand it:

$$\left(A_1\vec{\phi} + B_1\vec{\varepsilon}\right) \cdot \left(A_2\vec{\phi} + B_2\vec{\varepsilon}\right) = \left(A_1\vec{\phi}\right) \cdot \left(A_2\vec{\phi}\right) + \left(A_1\vec{\phi}\right) \cdot (B_2\vec{\varepsilon}) + (B_1\vec{\varepsilon}) \cdot \left(A_2\vec{\phi}\right) + (B_1\vec{\varepsilon}) \cdot (B_2\vec{\varepsilon}) \tag{28}$$

Each of these 4 terms contains a different combination of noise symbols and coefficients. We calculate interval bounds for each combination, e.g., $l_{\vec{\phi},\vec{\varepsilon}}, u_{\vec{\phi},\vec{\varepsilon}}$ for $\left(A_1\vec{\phi}\right) \cdot (B_2\vec{\varepsilon})$. Then the sum of the lower and upper bounds:

$$l = l_{\vec{\phi},\vec{\phi}} + l_{\vec{\phi},\vec{\varepsilon}} + l_{\vec{\varepsilon},\vec{\phi}} + l_{\vec{\varepsilon},\vec{\varepsilon}} \tag{29.1}$$

$$u = u_{\vec{\phi},\vec{\phi}} + u_{\vec{\phi},\vec{\varepsilon}} + u_{\vec{\varepsilon},\vec{\phi}} + u_{\vec{\varepsilon},\vec{\varepsilon}} \tag{29.2}$$

bounds the whole term $l \leq \left(A_1\vec{\phi} + B_1\vec{\varepsilon}\right) \cdot \left(A_2\vec{\phi} + B_2\vec{\varepsilon}\right) \leq u$.

2.9.1. Fast Bounds $l_{\gamma,\delta}, u_{\gamma,\delta}$ (DeepT-Fast [2])

To compute bounds for a generic expression $\left(V\xi_{p_1}\right) \cdot \left(W\xi_{p_2}\right)$, where V and W are matrices such that $V\xi_{p_1}$ and $W\xi_{p_2}$ have the same dimension and $\|\xi_{p_1}\|_{p_1} \leq 1$ and $\|\xi_{p_2}\|_{p_2} \leq 1$, we first compute an upper bound for the absolute value:

$$\left|\left(V\xi_{p_1}\right) \cdot \left(W\xi_{p_2}\right)\right| = |\xi_{p_1}^T V^T W\xi_{p_2}| \leq |\xi_{p_1}^T V^T| \; |W\xi_{p_2}| \tag{30}$$

Using Lemma 1, we can bound the elements $|\vec{w}_j \cdot \xi_{p_2}|$ of the vector $|W\xi_{p_2}|$, where $\vec{w}_j$ denotes the j-th row of W and ℓ_{q_2} is the dual norm of ℓ_{p_2}:

$$|\big(V\xi_{p_1}\big) \cdot \big(W\xi_{p_2}\big)| \leq |\xi_{p_1}^T V^T| \begin{pmatrix} |\vec{w}_1 \cdot \xi_{p_2}| \\ ... \\ |\vec{w}_N \cdot \xi_{p_2}| \end{pmatrix} \tag{31.1}$$

$$\leq |\xi_{p_1}^T V^T| \begin{pmatrix} \|\vec{w}_1\|_{q_2} & ... & \|\vec{w}_N\|_{q_2} \end{pmatrix} \tag{31.2}$$

$$= \begin{pmatrix} \|\vec{w}_1\|_{q_2} \\ ... \\ \|\vec{w}_N\|_{q_2} \end{pmatrix}^T |V\xi_{p_1}| \leq \begin{pmatrix} \|\vec{w}_1\|_{q_2} \\ ... \\ \|\vec{w}_N\|_{q_2} \end{pmatrix}^T |V| \, |\xi_{p_1}| \leq \left\| \begin{pmatrix} \|\vec{w}_1\|_{q_2} \\ ... \\ \|\vec{w}_N\|_{q_2} \end{pmatrix}^T |V| \right\|_{q_1} \tag{31.3}$$

where ℓ_{q_1} is the dual norm of ℓ_{p_1}.

The complexity to compute this bound is $O\big(N(\mathcal{E}_p + \mathcal{E}_\infty)\big)$.

2.9.2. More Precise Bounds $l_{\vec{\varepsilon},\vec{\varepsilon}}, u_{\vec{\varepsilon},\vec{\varepsilon}}$ (DeepT-Precise [2])

For the infinity noise interaction, a tighter approximation using interval analysis can be achieved at the cost of increasing the computational complexity to $O(N\mathcal{E}_\infty^2)$.

We begin by summing coefficients related to each pair of noise symbols:

$$(V\vec{\varepsilon}) \cdot (W\vec{\varepsilon}) = \sum_{i=1}^{\mathcal{E}_\infty} \sum_{j=1}^{\mathcal{E}_\infty} (\vec{v}_i \cdot \vec{w}_j) \varepsilon_i \varepsilon_j \tag{32}$$

where $\vec{v}_i$ and $\vec{w}_j$ denote the i-th and j-th column of V and W, respectively. We separate ε_i^2 and $\varepsilon_i \varepsilon_j$ to arrive at:

$$(V\vec{\varepsilon}) \cdot (W\vec{\varepsilon}) = \sum_{i=1}^{\mathcal{E}_\infty} (\vec{v}_i \cdot \vec{w}_i) \varepsilon_i^2 + \sum_{i \neq j} (\vec{v}_i \cdot \vec{w}_j) \varepsilon_i \varepsilon_j \tag{33}$$

Since $\varepsilon_i^2 \in [0, 1]$ and $\varepsilon_i \varepsilon_j \in [-1, 1]$, we have:

$$(V\vec{\varepsilon}) \cdot (W\vec{\varepsilon}) \in \sum_{i=1}^{\mathcal{E}_\infty} (\vec{v}_i \cdot \vec{w}_i)[0, 1] + \sum_{i \neq j} (\vec{v}_i \cdot \vec{w}_j)[-1, 1] \tag{34}$$

Using interval analysis, we can calculate the lower and upper interval bounds $l_{\vec{\varepsilon},\vec{\varepsilon}}$ and $u_{\vec{\varepsilon},\vec{\varepsilon}}$.

2.10. Softmax Abstract Transformer

The softmax can be computed with:

$$\sigma_i(x_1, ..., x_N) = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}} = \frac{1}{\sum_{j=1}^N e^{x_j - x_i}} \tag{35}$$

The latter formula being more numerically stable.

2.10.1. Softmax Sum Zonotope Refinement [6]

By construction, the outputs $y_1, ..., y_N$ of the softmax function σ when applied to inputs $x_1, ..., x_N$ satisfy $\sum_{i=1}^N y_i = 1$, meaning they form a probability distribution. Thus, in the multi-head self-attention, the role of the softmax is to pick some convex combination of the values V, according to the similarity between the query and the keys.

However, this property is not always satisfied for the Multi-norm Zonotope obtained for Z produced by the softmax abstract transformer Equation 35. By abuse of notation, we call this Zonotope Z. There are many valid instantiations of the noise symbols such that the Zonotope variables do not sum to 1, causing non-convex combinations of values to be picked. To address this, we enforce the constraint that the variables must sum to 1, to ensure that a convex combination is selected and to preserve the semantics of the network in our abstract domain. This is achieved by excluding from the Multi-norm Zonotope Z all invalid instantiations of values, obtaining a refined Multi-norm Zonotope Z' with lower volume, that helps to increase verification precision.

We leverage Zonotope constraint methods, which produce refined Zonotopes given some equality constraints. A three-step process is used to refine all Zonotope variables $y_1, ..., y_N$ by:

1. Computing a refined variable y'_1 by imposing the equality constraint $y_1 = 1 - (y_2 + ... + y_N)$,
2. Refining all other variables $y_2, ..., y_N$ to $y'_2, ..., y'_N$,
3. Tightening the bounds of the ε_i's to a subset of $[-1, 1]$.

Note that we arbitrarily select y_1 as the variable to be refined first, but any other variable could have been chosen.

We now detail these three steps that lead to a refined Zonotope Z' with variables $y'_1, ..., y'_n$ that always sum to 1 and have tighter error bounds.

2.10.2. Step 1. Refining y_1

We illustrate the process of obtaining a refined Zonotope with variable z'_1, given the equality constraint $z_1 = z_2$ for a Zonotope with two variables z_1 and z_2. The final result can then be obtained by instantiating $z_2 = 1 - (y_2 + ... + y_N)$ and $z_1 = y_1$ and finally $y'_1 = z'_1$.

While we know that $z_1 = z_2$ needs to hold, not all instantiations of the noise symbols satisfy this constraint. We can compute a new Multi-norm Zonotope variable $z'_1 = c' + \vec{\alpha}' \cdot \vec{\phi} + \vec{\beta}' \cdot \vec{\varepsilon}$, such that for all instantiations of noise symbols of z'_1, we have $z_1 = z_2$ and $z_1 = z'_1$, thereby enforcing the equality constraints $z_1 = z_2$. We have:

$$z_1 := c_1 + \vec{\alpha}_1 \cdot \vec{\phi} + \vec{\beta}_1 \cdot \vec{\varepsilon} = c_2 + \vec{\alpha}_2 \cdot \vec{\phi} + \vec{\beta}_2 \cdot \vec{\varepsilon} =: z_2 \tag{36}$$

If we solve for ε_k (any k such that $\beta_1^k - \beta_2^k \neq 0$ works) in the equation above and substitute it in the equation $z_2 = z'_1$, we obtain the following constraints for the coefficients of z'_1:

$$c' = c_2 + (c_2 - c_1)\frac{\beta'^k - \beta_2^k}{\beta_2^k - \beta_1^k} \tag{37}$$

$$\vec{\alpha}' = \vec{\alpha}_2 + (\vec{\alpha}_2 - \vec{\alpha}_1)\frac{\beta'^k - \beta_2^k}{\beta_2^k - \beta_1^k} \tag{38}$$

$$\vec{\beta}'^I = \vec{\beta}_2^I + \left(\vec{\beta}_2^I - \vec{\beta}_1^I\right)\frac{\beta'^k - \beta_2^k}{\beta_2^k - \beta_1^k} \tag{39}$$

where I are the indices of the other ε terms (i.e., without ε_k).

2.10.3. Choosing a Value for β'^k

In the equations above, we have one degree of freedom, namely β'^k. Any value v for β'^k is valid and leads to a valid affine expression z'_v, with the other coefficients of z'_v being deduced through the equations above.

To select v, we opt to minimize the absolute value of the noise symbol coefficients, which acts as a heuristic for the tightness of the zonotope variable:

$$v^* = \min_v S = \min_v \left[\|\vec{\alpha}'\|_1 + \|\vec{\beta}'\|_1 \right] \tag{40}$$

The minimization problem above can be efficiently solved with $O\big((\mathcal{E}_p + \mathcal{E}_\infty)\log(\mathcal{E}_p + \mathcal{E}_\infty)\big)$ complexity, using a method that relies on two observations:

1. Since all coefficients in the minimization can be written in form $r + s\beta_z^k$ with $r, s \in \mathbb{R}$ (see Eqs. above), the expression to be minimized is of the form $S = \sum_t |r_t + s_t \beta_z^k|$. The optimal value v^* for β_z^k will cause one of the $|r_t + s_t\beta_z^k|$ terms to be 0 and therefore v^* must equal $-\frac{r_i}{s_i}$ for some $i \in \left[1, \mathcal{E}_p + \mathcal{E}_\infty\right]$. The values $-\frac{r_i}{s_i}$ are the candidate solutions for the minimization problem.
2. Each term $|r_t + s_t\beta_z^k|$ of S has a constant negative slope before $-\frac{r_t}{s_t}$ and a constant positive slope after it. Therefore, as β_z^k increases, the slope of more and more $|r_t + s_t\beta_z^k|$ terms becomes positive, showing that the slope of S increases monotonically with β_z^k. The minimum value of S will happen at the value of β_z^k where the slope of S changes from negative to positive.

Since the slope of S increases monotonically, we can run a binary search on β_z^k to efficiently find the value at which the slope of S changes sign. We note that to maintain precision, we disallow solutions that lead to the elimination of one of the ℓ_p-norm noise symbols ϕ.

2.10.4. Step 2. Refining $y_2, ..., y_n$

We substitute the expression for ε_k computed in Step 1 in the affine expressions of the variables $y_2, ..., y_N$, to obtain the refined Multi-norm Zonotope variables $y_2', ..., y_N'$.

Specifically, from Step 1, we derived:

$$\varepsilon_k = \frac{c_1 - c_2 + (\vec{\alpha}_1 - \vec{\alpha}_2)\cdot\vec{\phi} + \sum_{j\neq k}\left(\beta_1^j - \beta_2^j\right)\varepsilon_j}{\beta_2^k - \beta_1^k} \tag{41}$$

For each variable y_i with $i \in \{2, ..., N\}$, defined as $y_i = c_i + \vec{\alpha}_i \cdot \vec{\phi} + \vec{\beta}_i \cdot \vec{\varepsilon}$, we substitute the expression for ε_k to get:

$$y_i' = c_i + \vec{\alpha}_i \cdot \vec{\phi} + \sum_{j\neq k} \beta_i^j \varepsilon_j + \beta_i^k \varepsilon_k \tag{42}$$

$$y_i' = c_i + \vec{\alpha}_i \cdot \vec{\phi} + \sum_{j\neq k} \beta_i^j \varepsilon_j + \beta_i^k \left(\frac{c_1 - c_2 + (\vec{\alpha}_1 - \vec{\alpha}_2)\cdot\vec{\phi} + \sum_{j\neq k}\left(\beta_1^j - \beta_2^j\right)\varepsilon_j}{\beta_2^k - \beta_1^k} \right) \tag{43}$$

After simplification, this gives us the coefficients for our refined variables y_i'.

2.10.5. Step 3. Tightening the Bounds of $\vec{\varepsilon}$

The refined sum constraint $S = 1 - \sum_{i=1}^N y_i' = c_S + \vec{\alpha}_S \cdot \vec{\phi} + \vec{\beta}_S \cdot \vec{\varepsilon} = 0$ can be further leveraged to tighten the bounds of the ℓ_∞ noise symbols $\vec{\varepsilon}$, with non-zero coefficient, by solving for ε_m:

$$\varepsilon_m = \frac{1}{\beta_S^m}\left[-c_S - \vec{\alpha}_S \cdot \vec{\phi} - \vec{\beta}_S^I \cdot \vec{\varepsilon}^I\right] \tag{44}$$

Which implies that the range of ε_m is restricted to $[a_m, b_m] \cap [-1, 1]$ where:

$$a_m = \frac{1}{|\beta_S^m|}\left(c_S - \|\vec{\alpha}_S\|_q - \|\vec{\beta}_S\|_1\right) \tag{45}$$

$$b_m = \frac{1}{|\beta_S^m|}\Big(c_S + \|\vec{\alpha}_S\|_q + \|\vec{\beta}_S\|_1\Big) \tag{46}$$

Note that because the noise symbol reduction process assumes all noise symbols $\vec{\varepsilon}$ have range $[-1, 1]$, prior to it a pre-processing step occurs where all noise symbols ε_m with tightened bounds $[a_m, b_m] \subset [-1, 1]$ are re-written as:

$$\varepsilon_m = \frac{a_m + b_m}{2} + \frac{b_m - a_m}{2}\varepsilon_{\text{new},m} \tag{47}$$

with $\varepsilon_{\text{new},m} \in [-1, 1]$.

This three-step refinement process ensures that our Multi-norm Zonotope respects the semantics of the softmax function, improving the precision of our verification procedure.

References

[1] A. Albarghouthi, "Introduction to Neural Network Verification," 2021.

[2] G. Bonaert, D. I. Dimitrov, M. Baader, and M. Vechev, "Fast and precise certification of transformers," pp. 466–481, 2021, doi: 10.1145/3453483.3454056.

[3] M. Mirman, G. Singh, and M. Vechev, "A Provable Defense for Deep Residual Networks," 2020.

[4] M. V. Mark Niklas Müller Mislav Balunović, "Certify or Predict: Boosting Certified Robustness with Compositional Architectures," 2021.

[5] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, "Fast and effective robustness certification," pp. 10825–10836, 2018.

[6] K. Ghorbal, E. Goubault, and S. Putot, "A Logical Product Approach to Zonotope Intersection," pp. 212–226, 2010, doi: 10.1007/978-3-642-14295-6_22.