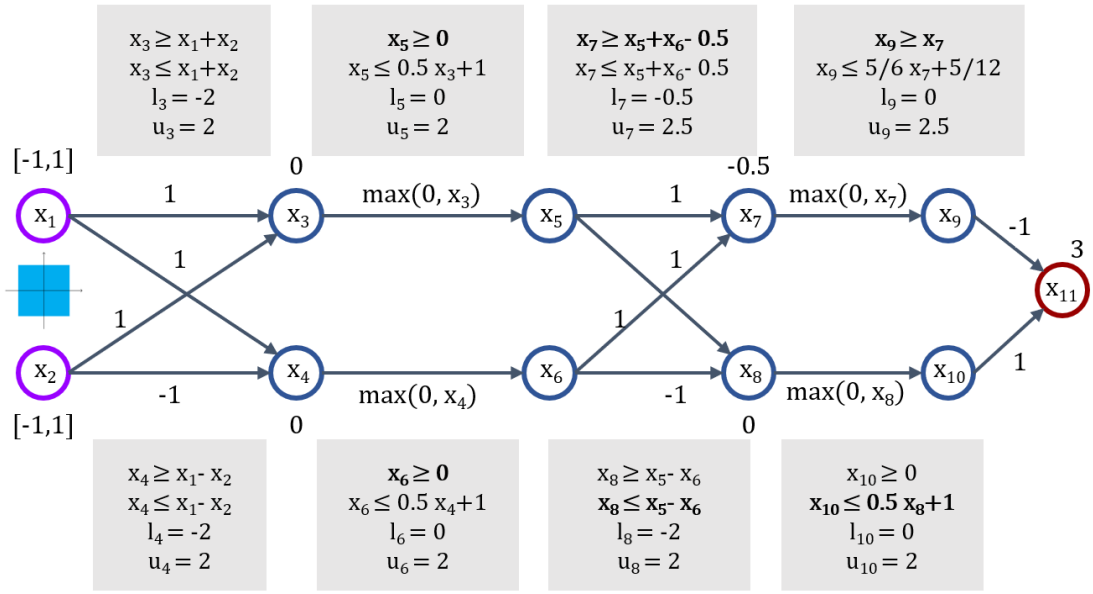
Exercise 04 - Solution

DeepPoly Branch and Bound Certification

Reliable and Trustworthy Artificial Intelligence
ETH Zurich

Problem 1 (DeepPoly Branch and Bound)**.** Consider the neural network below, taken from this week's lecture slides. We show the result of analysing the network using the DeepPoly algorithm on the ℓ_∞ region $\left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_\infty \leq 1$ i.e. ℓ_∞ ball around $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ with size 1.



(a) Recall from the lecture, in the original DeepPoly analysis we computed the upper bound of x_{11} to be 4.5. Apply branching to the ReLU node at x_8. What upper bound for x_{11} do you obtain if you apply symbolic analysis on β (where β is the KKT variable introduced by the split at x_8, as in the lecture)? Is the resulting bound more or less precise than the original bound?

(b) The analysis you performed in (a) was done for the input region represented by an ℓ_∞ ball of size 1 around $\begin{bmatrix}0\\0\end{bmatrix}$. Without changing the intermediate neuron lower and upper bounds, use the Holder inequality to similarly compute an upper bound on x_{11} for two additional input regions — an ℓ_1 and ℓ_2 balls of size 1 around $\begin{bmatrix}0\\0\end{bmatrix}$. Is the resulting upper bound on x_{11} sound? How can you make it more precise?

(c) In (a) and (b), we applied symbolic analysis to obtain the upper bound on x_{11}. This is often infeasible in practice. Next, we find the value of β that produces the best upper bound for x_{11} using numerical optimization for the original ℓ_∞ input region. Assume, β is initialized to 1.2 (for both branches). Perform one gradient step on β with step size 0.3 on both branches. What upper bound do you obtain for x_{11}? Is the produced upper bound sound? How does it compare to the original DeepPoly bound? How does it compare to the bound obtained in (a)?

Solution 1. (a) Using standard backsubstitution we get:

$$x_{11} \leq x_{10} - x_9 + 3 \leq x_{10} - x_7 + 3 \tag{1}$$

Next, we need to branch on x_8. To do this, we look at the two ReLU branches of x_8 — the positive branch where $x_8 \geq 0$ and the negative branch where $x_8 \leq 0$.

For the positive branch we resolve the ReLU exactly to:

$$x_8 \leq x_{10} \leq x_8. \tag{2}$$

Which we can substitute back in Eq. (1):

$$x_{11} \leq x_{10} - x_7 + 3 \leq x_8 - x_7 + 3. \tag{3}$$

In addition to resolving the ReLU, we also need to add the additional positivity constraint:

$$-x_8 \leq 0. \tag{4}$$

Using KKT with $g(x_8) = -x_8$ and $f(x_8) = x_8 - x_7 + 3$, we can incorporate the constraint as follows :

$$x_{11} \leq \max_x \min_{\beta \geq 0} x_8 - x_7 + 3 + \beta x_8 \leq \min_{\beta \geq 0} \max_x x_8 - x_7 + 3 + \beta x_8, \tag{5}$$

where last inequality comes from weak duality. For notational convenience we shorten this to:

$$x_{11} \leq \min_{\beta \geq 0} x_8 - x_7 + 3 + \beta x_8. \tag{6}$$

We can now continue the backsubstitution procedure:

$$\begin{aligned} x_{11} &\leq \min_{\beta \geq 0} x_8 - x_7 + 3 + \beta x_8 = \min_{\beta \geq 0} (\beta + 1) x_8 - x_7 + 3 \leq \\ &\leq \min_{\beta \geq 0} (\beta + 1)(x_5 - x_6) - (x_5 + x_6 - 0.5) + 3 = \\ &= \min_{\beta \geq 0} \beta x_5 - (\beta + 2) x_6 + 3.5. \end{aligned} \tag{7}$$

As $\beta \geq 0$, we know that the coefficient in front of x_5 is non-negative and the coefficient in front of x_6 is non-positive. Therefore, we can continue the backsubstitution:

$$\begin{aligned} x_{11} &\leq \min_{\beta \geq 0} \beta x_5 - (\beta + 2) x_6 + 3.5 \leq \min_{\beta \geq 0} \beta(0.5 x_3 + 1) - (\beta + 2)0 + 3.5 = \\ &= \min_{\beta \geq 0} 0.5\beta x_3 + 3.5 + \beta \leq \min_{\beta \geq 0} 0.5\beta(x_1 + x_2) + 3.5 + \beta = \\ &= \min_{\beta \geq 0} 0.5\beta x_1 + 0.5\beta x_2 + 3.5 + \beta. \end{aligned} \tag{8}$$

Again, as $\beta \geq 0$, we know the coefficients in front of x_1 and x_2 are non-negative, which allows us to continue the backsubstitution:

$$\begin{aligned} x_{11} &\leq \min_{\beta \geq 0} 0.5\beta x_1 + 0.5\beta x_2 + 3.5 + \beta \leq \min_{\beta \geq 0} 0.5\beta + 0.5\beta + 3.5 + \beta = \\ &= \min_{\beta \geq 0} 2\beta + 3.5 = 3.5 \end{aligned} \tag{9}$$

Thus, we refine our upper bound in this branch to $u_{11}^{+} = 3.5$.

For the negative branch we resolve the ReLU exactly to:

$$0 \leq x_{10} \leq 0 \tag{10}$$

Which we can substitute back in Eq. (1):

$$x_{11} \leq x_{10} - x_7 + 3 \leq -x_7 + 3. \tag{11}$$

Like before, we also need to add the additional negativity constraint:

$$x_8 \leq 0, \tag{12}$$

which we incorporate with KKT with $g(x_8) = x_8$ and $f(x_8) = -x_7 + 3$ and weak duality. The result is as follows:

$$x_{11} \leq \max_{x} \min_{\beta \geq 0} \; - x_7 + 3 - \beta x_8 \leq \min_{\beta \geq 0} \max_{x} \; - \beta x_8 - x_7 + 3. \tag{13}$$

We continue with the backsubstitution:

$$x_{11} \leq \min_{\beta \geq 0} \; -\beta x_8 - x_7 + 3 \leq \min_{\beta \geq 0} \; -\beta(x_5 - x_6) - (x_5 + x_6 - 0.5) + 3 = \\ = \min_{\beta \geq 0} \; -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \tag{14}$$

As $\beta \geq 0$, the coefficient in front of x_5 is non-positive. However, depending whether $\beta > 1$ or not the sign in front of x_6 changes.

We look first at the case when $0 \leq \beta \leq 1$. In that case, the coefficient in front of x_6 is non-positive. We back-substitute, resulting in:

$$\min_{0 \leq \beta \leq 1} \; -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \leq \min_{0 \leq \beta \leq 1} \; -(\beta+1)0 + (\beta-1)0 + 3.5 = 3.5. \tag{15}$$

Next, we look at the case when $\beta \geq 1$. In that case, the coefficient in front of x_6 is non-negative. We back-substitute, resulting in:

$$\min_{\beta \geq 1} \; -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \leq \min_{\beta \geq 1} \; -(\beta+1)0 + (\beta-1)(0.5x_4 + 1) + 3.5 = \\ = \min_{\beta \geq 1} 0.5(\beta-1)x_4 + 2.5 + \beta \leq \min_{\beta \geq 1} 0.5(\beta-1)(x_1 - x_2) + 2.5 + \beta \leq \\ \leq \min_{\beta \geq 1} 0.5(\beta-1) + 0.5(\beta-1) + 2.5 + \beta = \min_{\beta \geq 1} 2\beta + 1.5 = 3.5, \tag{16}$$

where transition between the second and third lines is again due to using $\beta \geq 1$. As $\min_{\beta \geq 1} -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \leq 3.5$ and $\min_{0 \leq \beta \leq 1} -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \leq 3.5$, we get:

$$x_{11} \leq \min_{\beta \geq 0} \; -(\beta+1)x_5 + (\beta-1)x_6 + 3.5 \leq \min(3.5, 3.5) = 3.5. \tag{17}$$

Therefore, the refined upper bound in this branch is $u_{11}^- = 3.5$.

The final upper bound on x_{11} is then given by $\max(u_{11}^+, u_{11}^-) = 3.5$.

(b) As we don't want to change the intermediate neuron bounds, the back-substitution in both the positive branch and the two cases of the negative branch above are not affected, except for the final backsubstitution step of x_1 and x_2. For the positive

branch, we have from Eq. (8):

$$x_{11} \leq \min_{\beta \geq 0} \max_{x} 0.5\beta x_1 + 0.5\beta x_2 + 3.5 + \beta = \min_{\beta \geq 0} \max_{x} \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T + 3.5 + \beta \leq$$
$$\leq \min_{\beta \geq 0} \max_{x} \left\| \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \right\| + 3.5 + \beta \leq \min_{\beta \geq 0} \max_{x} \left\| \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \right\|_p \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_q + 3.5 + \beta \tag{18}$$

for any $\frac{1}{p} + \frac{1}{q} = 1$.

For the ℓ_1 ball input region, we set $q = 1$ and $p = \infty$. Therefore, for the ℓ_1 ball input region:

$$x_{11} \leq \min_{\beta \geq 0} \max_{x} \left\| \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \right\|_\infty \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_1 + 3.5 + \beta = \min_{\beta \geq 0} 1.5\beta + 3.5 = 3.5. \tag{19}$$

For the ℓ_2 ball input region, we set $q = 2$ and $p = 2$. Therefore, for the ℓ_2 ball input region:

$$x_{11} \leq \min_{\beta \geq 0} \max_{x} \left\| \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_2 + 3.5 + \beta = \min_{\beta \geq 0} (0.5\sqrt{2} + 1)\beta + 3.5 = 3.5. \tag{20}$$

For the negative branch and the case $0 \leq \beta \leq 1$, as we don't need to propagate back to the input to obtain the bound, the bound remains 3.5 for all input regions. For the negative branch and the case $\beta \geq 1$, we have:

$$x_{11} \leq \min_{\beta \geq 1} \max_{x} 0.5(\beta - 1)(x_1 - x_2) + 2.5 + \beta = \min_{\beta \geq 1} \max_{x} \begin{bmatrix} 0.5\beta \\ 0.5\beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T + 2.5 + \beta \leq$$
$$\leq \min_{\beta \geq 1} \max_{x} \left\| \begin{bmatrix} 0.5\beta - 0.5 \\ -0.5\beta + 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \right\| + 2.5 + \beta \leq \tag{21}$$
$$\leq \min_{\beta \geq 1} \max_{x} \left\| \begin{bmatrix} 0.5\beta - 0.5 \\ -0.5\beta + 0.5 \end{bmatrix} \right\|_p \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_q + 2.5 + \beta$$

for any $\frac{1}{p} + \frac{1}{q} = 1$.

Therefore, for the ℓ_1 ball input region:

$$x_{11} \leq \min_{\beta \geq 1} \max_{x} \left\| \begin{bmatrix} 0.5\beta - 0.5 \\ -0.5\beta + 0.5 \end{bmatrix} \right\|_\infty \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_1 + 2.5 + \beta =$$
$$= \min_{\beta \geq 1} \|0.5\beta - 0.5\| + 2.5 + \beta = \min_{\beta \geq 1} 1.5\beta + 2 = 3.5. \tag{22}$$

and for the ℓ_2 ball input region:

$$x_{11} \leq \min_{\beta \geq 1} \max_{x} \left\| \begin{bmatrix} 0.5\beta - 0.5 \\ -0.5\beta + 0.5 \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_2 + 2.5 + \beta = \\ = \min_{\beta \geq 1} (0.5\beta - 0.5)\sqrt{2} + 2.5 + \beta = \min_{\beta \geq 1} (0.5\sqrt{2} + 1)\beta + (2.5 - 0.5\sqrt{2}) = 3.5. \tag{23}$$

To summarize, none of the bounds changed. As the ℓ_1 and ℓ_2 balls are subregions of the ℓ_∞ ball our approximation is sound but not precise (as the removed volume didn't improve bounds). One way to improve the precision is to apply Holder's inequality to all intermediate bounds in the analysis, resulting in a tighter DeepPoly encoding of the network.

(c) As no decision in the back-substitution in the positive branch depends on the value of β, executing the DeepPoly with $\beta_0 = 1.2$ will exactly result in Eq. (9) with β substituted with β_0:

$$x_{11} \leq 2\beta_0 + 3.5 = 5.9 \tag{24}$$

To find the optimal β, we need to compute the gradient $\nabla_\beta(2\beta + 3.5) = 2$ and apply 1 step of SGD, resulting in:

$$\beta_1 = \beta_0 - \gamma \nabla_\beta (2\beta + 3.5) = 1.2 - 0.3 * 2 = 0.6. \tag{25}$$

As the back-substitution does not depend on the value of β for the positive branch, we obtain the following better upper bound for x_{11} at β_1:

$$x_{11} \leq 2\beta_1 + 3.5 = 4.7. \tag{26}$$

The back-substitution for the initial value $\beta_0 = 1.2 \geq 1$ in the negative branch, results, as demonstrated in (a), in Eq. (16) with β substituted with β_0:

$$x_{11} \leq 2\beta_0 + 1.5 = 3.9 \tag{27}$$

To find the optimal β, we need to compute the gradient $\nabla_\beta(2\beta + 1.5) = 2$ and apply 1 step of SGD, resulting in:

$$\beta_1 = \beta_0 - \gamma \nabla_\beta (2\beta + 1.5) = 1.2 - 0.3 * 2 = 0.6. \tag{28}$$

As the back-substitution for $\beta_1 = 0.6 \leq 1$ in the negative branch, results, as demonstrated in (a), in Eq. (15) with β substituted with β_1:

$$x_{11} \leq 3.5. \tag{29}$$

The final upper bound of x_{11} is given by the maximum of the bounds in both branches, which is 4.7. This bound is sound but less precise than the original DeepPoly bound. This is because any value for β produces a valid upper bound for x_{11}, but our optimization in the positive branch failed to get close to the global optimal value for β. The global optimum value was nevertheless achieved for the negative branch.