Reliable and Trustworthy Artificial Intelligence

Lecture 4: Differentiable refinement of DeepPoly, Branch and Bound

Martin Vechev

ETH Zurich

Fall 2022

Lecture Outline (Part I)

- Arbitrary input norms for DeepPoly

- Differentiable version of the DeepPoly refinement

Lecture Outline (Part I)

- **Arbitrary input norms for DeepPoly**

- Differentiable version of the DeepPoly refinement

Reminder: backsubstitution

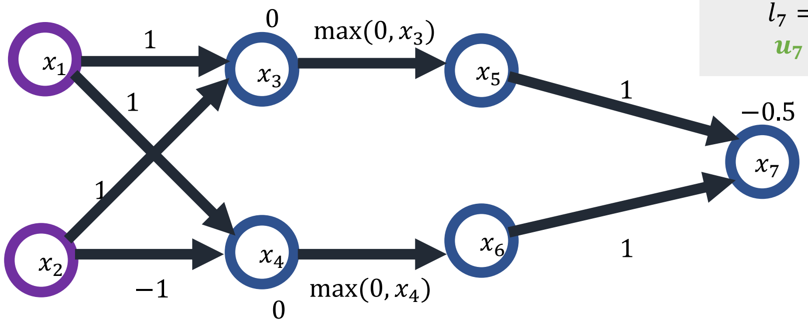
$x_1 \geq -1,$
$x_1 \leq 1,$
$l_1 = -1,$
$u_1 = 1$

$x_3 \geq x_1 + x_2,$
$x_3 \leq x_1 + x_2,$
$l_3 = -2,$
$u_3 = 2$

$x_5 \geq 0,$
$x_5 \leq 0.5 \cdot x_3 + 1,$
$l_5 = 0,$
$u_5 = 2$

$x_7 \geq -0.5,$
$x_7 \leq x_1 + 1.5,$
$l_7 = -0.5,$
$\mathbf{u_7 = 2.5}$

We obtained 2.5 via backsubstitution



$x_7 \leq 0.5x_3 + 0.5x_4 + 1.5$

$x_7 \leq 0.5(x_1 + x_2) + 0.5(x_1 - x_2) + 1.5$

$x_7 \leq \boxed{x_1 + 1.5}$

$x_7 \leq 2.5$

Finding the maximum of the expression $x_1 + 1.5$ was direct as we were dealing with a very specific input region, each neuron in [-1,1], but there could be more complex input regions with multiple x's.

How can we deal with any norm?

Holder inequality (special case)

For vectors x and y, where $\frac{1}{p} + \frac{1}{q} = 1$

Note: p, q need not be integers

$$\left|\left|x * y\right|\right|_1 \leq \left|\left|x\right|\right|_q \left|\left|y\right|\right|_p$$

For the special case where $p = \infty$ we have that $q = 1$ and the above holds.

Let us now use this inequality to obtain lower/upper bounds for more general input regions.

$$\max_{x_0 \in \{x \in D \;|\; ||x - x'||_p \le \epsilon\}} ax^0 + c$$

via syntactic re-write

$$= \max_{||\eta||_p \le \epsilon} a(x' + \eta) + c$$

via expanding brackets

$$= \max_{||\eta||_p \le \epsilon} a\eta + ax' + c$$

via $a\eta \le |a\eta|$

$$\le \max_{||\eta||_p \le \epsilon} |a\eta| + ax' + c$$

via $|a\eta| = ||a\eta||_1$

$$= \max_{||\eta||_p \le \epsilon} ||a\eta||_1 + ax' + c$$

via Holder inequality

$$\le \max_{||\eta||_p \le \epsilon} ||a||_q ||\eta||_p + ax' + c$$

via $\max x * y = x * \epsilon$
where $x \ge 0$, $y \in [0, \epsilon]$

$$= ||a||_q \epsilon + ax' + c$$

We also need to have a way to compute the **minimum of an expression**, for instance, when we try to prove the final property or the lower bounds. The derivation is similar to the maximum.

$$\min_{x_0 \in \{x \in D \;|\; ||x - x'||_p \leq \epsilon\}} ax^0 + c$$

via syntactic re-write

$$= \min_{||\eta||_p \leq \epsilon} a(x' + \eta) + c$$

via expanding brackets

$$= \min_{||\eta||_p \leq \epsilon} a\eta + ax' + c$$

via $a\eta \geq -|a\eta|$

$$\geq \min_{||\eta||_p \leq \epsilon} -|a\eta| + ax' + c$$

via $|a\eta| = ||a\eta||_1$

$$= \min_{||\eta||_p \leq \epsilon} -||a\eta||_1 + ax' + c$$

via Holder inequality

$$\geq \min_{||\eta||_p \leq \epsilon} -||a||_q ||\eta||_p + ax' + c$$

via $\min -x * y = -x * \epsilon$
where $x \geq 0$, $y \in [0, \epsilon]$

$$= -||a||_q \epsilon + ax' + c$$

Back to our example

$$\frac{1}{p} + \frac{1}{q} = 1$$

In the special case where we have the l-inf norm, that is $p = \infty$ then $q = 1$

We need to find an upper bound to the expression: $x_1 + 1.5$

Here we have that $\boldsymbol{x}' = (x_1, x_2) = (0,0)$, $\epsilon = 1$, $\boldsymbol{a} = (1,0)$, $c = 1.5$, $q = 1$

Hence the expression: $||\boldsymbol{a}||_q \epsilon + \boldsymbol{a}\boldsymbol{x}'^{\boldsymbol{T}} + c$

becomes: $||(1,0)||_1 * 1 + (1,0) * (0,0)^T + 1.5 = 2.5$

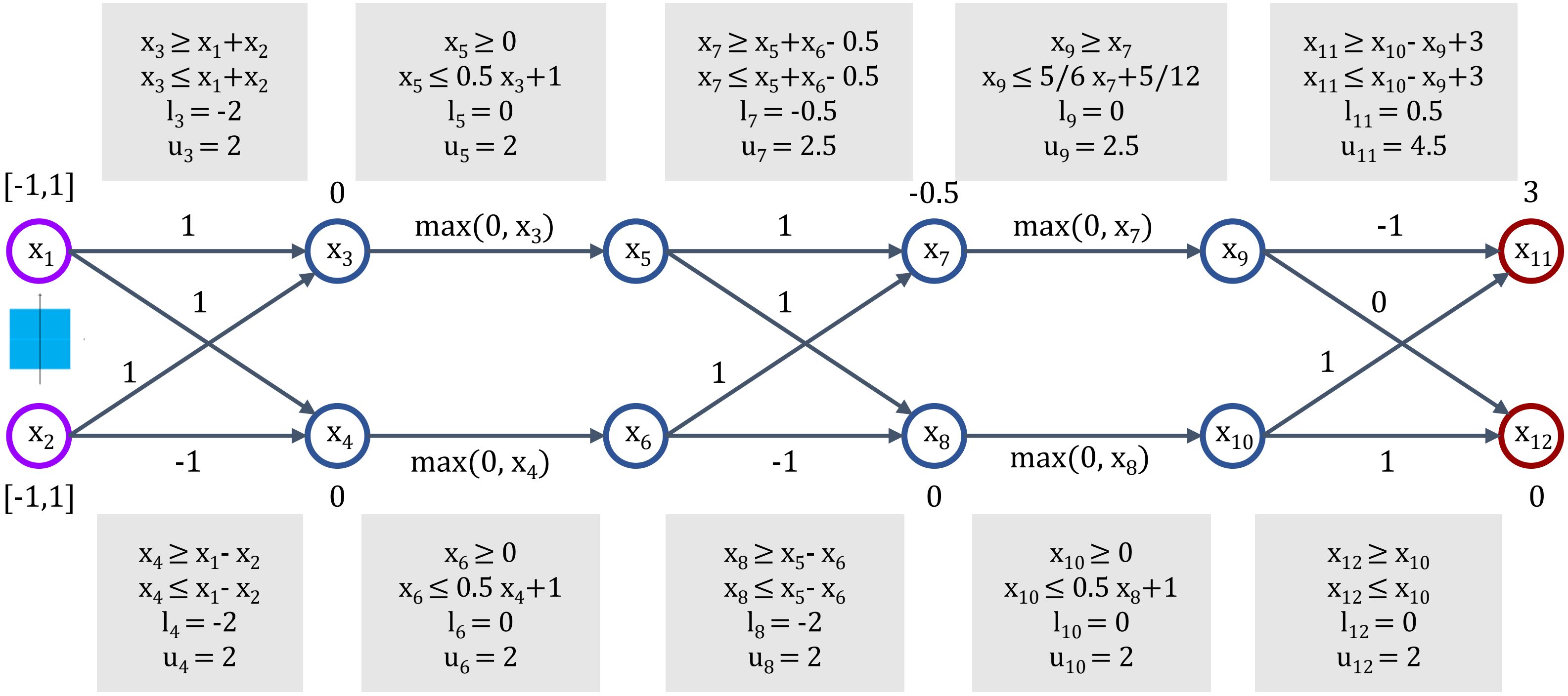
so the upper bound is **2.5**

Lecture Outline (Part I)

- Arbitrary input norms for DeepPoly

- **Differentiable version of the DeepPoly refinement**

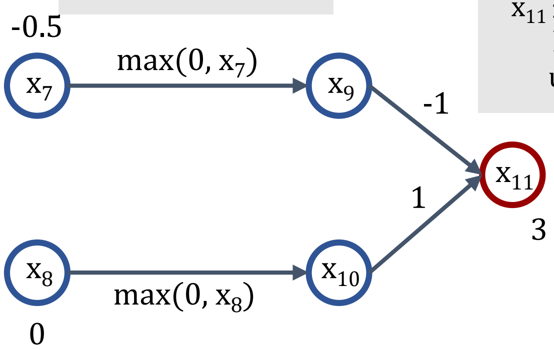
Standard Backsubstitution



Standard Backsubstitution

Depending on sign, we substitute lower or upper bound

$\mathbf{x_9 \geq x_7}$
$x_9 \leq 5/6\, x_7 + 5/12$
$l_9 = 0$
$u_9 = 2.5$

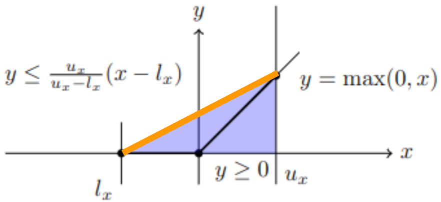


$x_{11} \geq x_{10} - x_9 + 3$
$x_{11} \leq x_{10} - x_9 + 3$
$l_{11} = 0.5$
$u_{11} = 4.5$

$x_{10} \geq 0$
$\mathbf{x_{10} \leq 0.5\, x_8 + 1}$
$l_{10} = 0$
$u_{10} = 2$

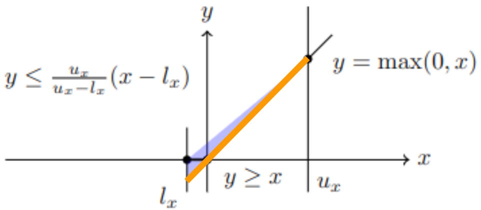
$x_{11} \leq x_{10} - x_9 + 3$

pos. coeff. ⇒ upper bound



Substitute
$\mathbf{x_{10} \leq 0.5\, x_8 + 1}$

neg. coeff. ⇒ lower bound



Substitute
$\mathbf{x_9 \geq x_7}$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\, x_8 - x_7 + 4$

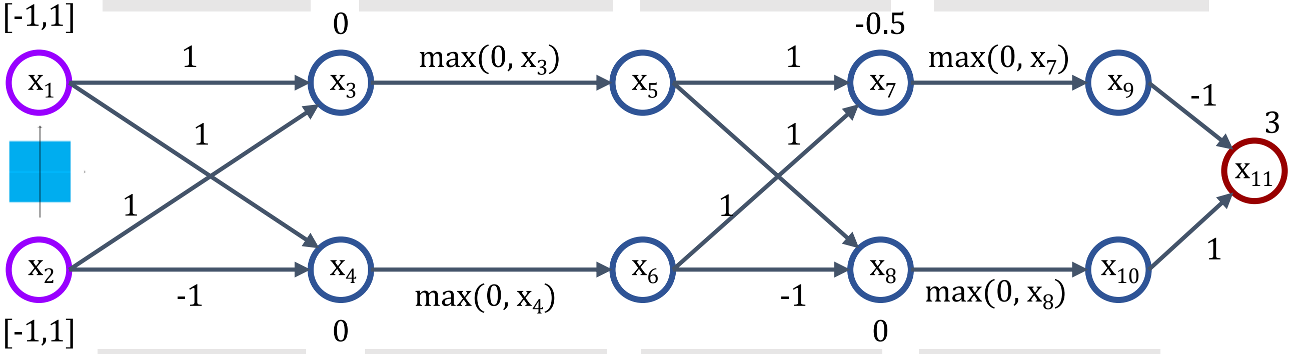
Standard Backsubstitution

$x_3 \geq x_1 + x_2$
$x_3 \leq x_1 + x_2$
$l_3 = -2$
$u_3 = 2$

$\mathbf{x_5 \geq 0}$
$x_5 \leq 0.5\, x_3 + 1$
$l_5 = 0$
$u_5 = 2$

$\mathbf{x_7 \geq x_5 + x_6 - 0.5}$
$x_7 \leq x_5 + x_6 - 0.5$
$l_7 = -0.5$
$u_7 = 2.5$

$\mathbf{x_9 \geq x_7}$
$x_9 \leq 5/6\, x_7 + 5/12$
$l_9 = 0$
$u_9 = 2.5$



$x_4 \geq x_1 - x_2$
$x_4 \leq x_1 - x_2$
$l_4 = -2$
$u_4 = 2$

$\mathbf{x_6 \geq 0}$
$x_6 \leq 0.5\, x_4 + 1$
$l_6 = 0$
$u_6 = 2$

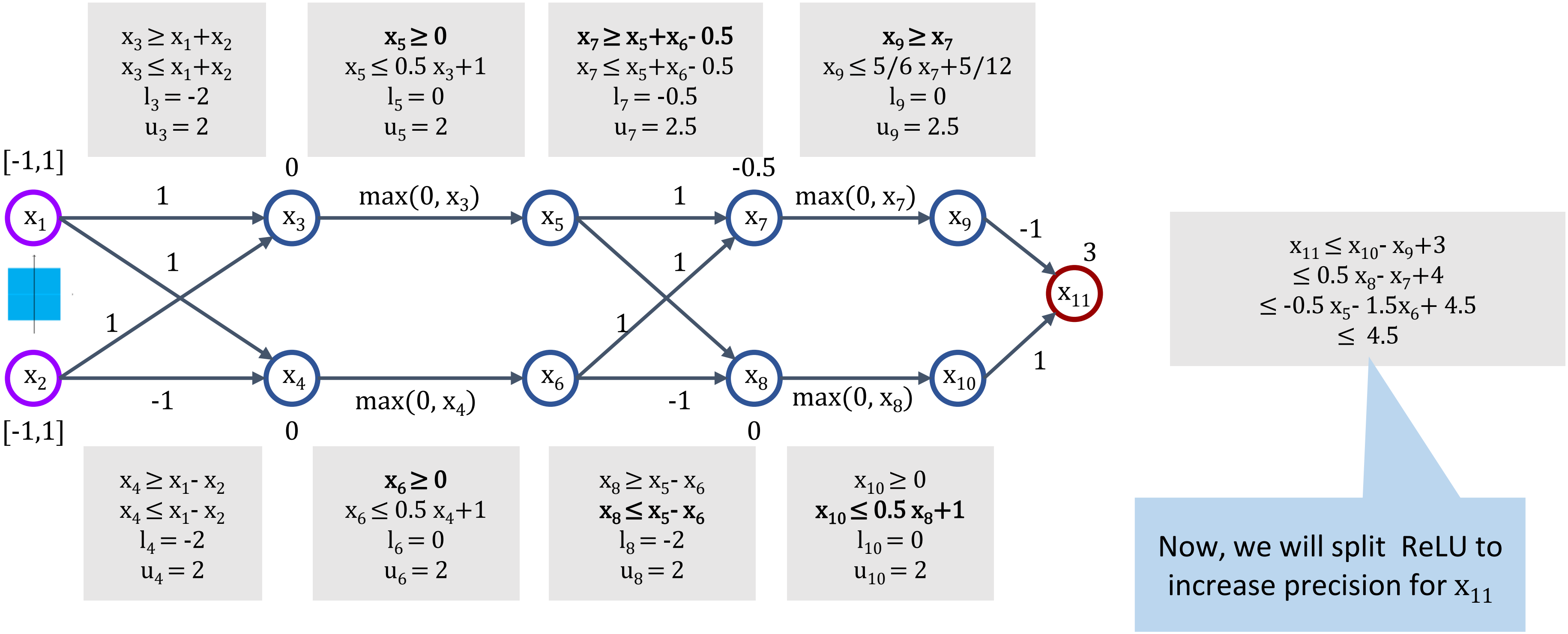
$x_8 \geq x_5 - x_6$
$\mathbf{x_8 \leq x_5 - x_6}$
$l_8 = -2$
$u_8 = 2$

$x_{10} \geq 0$
$\mathbf{x_{10} \leq 0.5\, x_8 + 1}$
$l_{10} = 0$
$u_{10} = 2$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\, x_8 - x_7 + 4$
$\leq -0.5\, x_5 - 1.5 x_6 + 4.5$
≤ 4.5

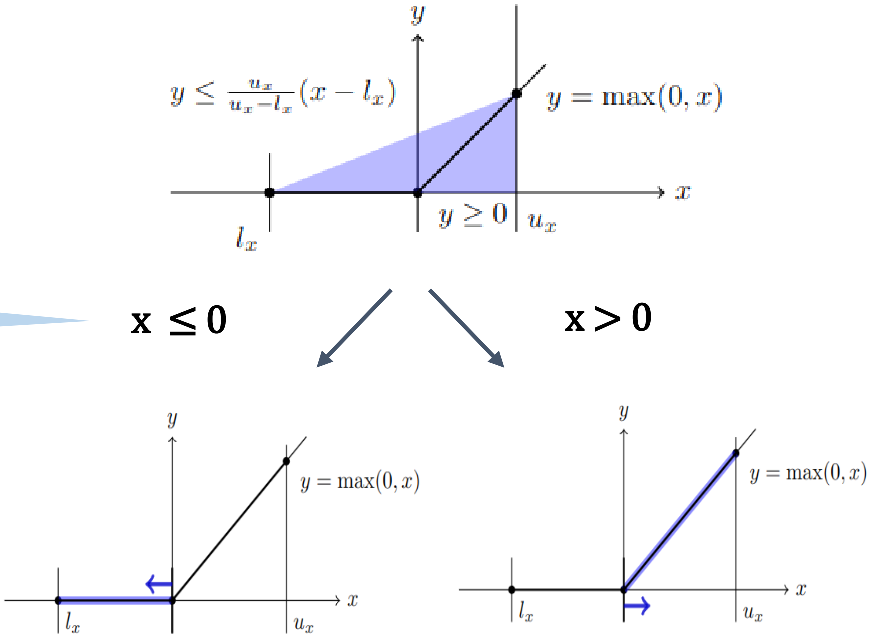
Both x_5 and x_6 are substituted with lower bound 0 (negative sign)

Standard Backsubstitution



Branch-and-Bound

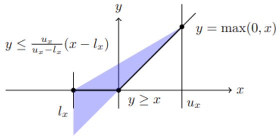
How to enforce with backsubstitution?



Branch-and-Bound (BaB)
[Bunel et al. 2020]

Bunel et al., "Branch and bound for piecewise linear neural network verification", Journal of Machine Learning Research, 2020

Replace ReLU with linear function

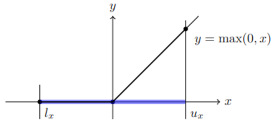


$x_9 \geq x_7$
$x_9 \leq 5/6\ x_7 + 5/12$
$l_9 = 0$
$u_9 = 2.5$

$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\ x_8 - x_7 + 4$
$\leq -0.5\ x_5 - 1.5x_6 + 4.5$
$\leq\ 4.5$

Split on x_7



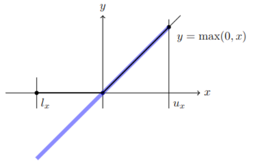
$\mathbf{x_9 = 0}$

$\mathbf{x_9 \geq 0}$
$\mathbf{x_9 \leq 0}$
$l_9 = 0$
$\mathbf{u_9 = 0}$

$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

$\mathbf{x_7 \leq 0}$
$\mathbf{\Rightarrow x_9 = 0}$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\ x_8 + 4$
$\leq 0.5\ x_5 - 0.5x_6 + 4$
$\leq 0.25\ x_3 + 4.5$
$\leq 0.25\ x_1 + 0.25\ x_2 + 4.5$
≤ 5



$\mathbf{x_9 = x_7}$

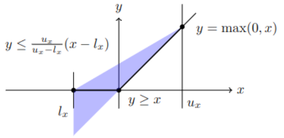
$x_9 \geq x_7$
$\mathbf{x_9 \leq x_7}$
$l_9 = 0$
$u_9 = 2.5$

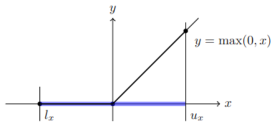
$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

$\mathbf{x_7 > 0}$
$\mathbf{\Rightarrow x_9 = x_7}$

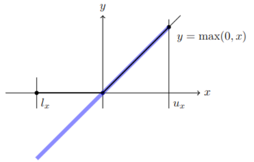
$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\ x_8 - x_7 + 4$
$\leq -0.5\ x_5 - 1.5x_6 + 4.5$
$\leq\ 4.5$

Naive splitting is not strictly better!





$\mathbf{x_9 = 0}$



$\mathbf{x_9 = x_7}$

$x_9 \geq x_7$
$x_9 \leq 5/6\, x_7 + 5/12$
$l_9 = 0$
$u_9 = 2.5$

$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\, x_8 - x_7 + 4$
$\leq -0.5\, x_5 - 1.5 x_6 + 4.5$
≤ 4.5

Split on x_7

$\mathbf{x_9 \geq 0}$
$\mathbf{x_9 \leq 0}$
$l_9 = 0$
$\mathbf{u_9 = 0}$

$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

$\mathbf{x_7 \leq 0}$
$\mathbf{\Rightarrow x_9 = 0}$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\, x_8 + 4$
$\leq 0.5\, x_5 - 0.5 x_6 + 4$
$\leq 0.25\, x_3 + 4.5$
$\leq 0.25\, x_1 + 0.25\, x_2 + 4.5$
≤ 5

$x_9 \geq x_7$
$\mathbf{x_9 \leq x_7}$
$l_9 = 0$
$u_9 = 2.5$

$x_7 \xrightarrow{\max(0,\, x_7)} x_9$

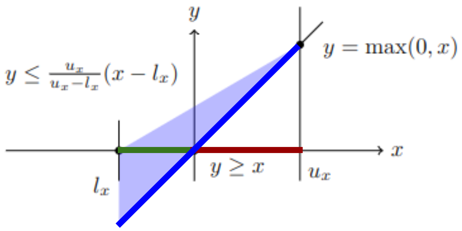
$\mathbf{x_7 > 0}$
$\mathbf{\Rightarrow x_9 = x_7}$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq 0.5\, x_8 - x_7 + 4$
$\leq -0.5\, x_5 - 1.5 x_6 + 4.5$
≤ 4.5

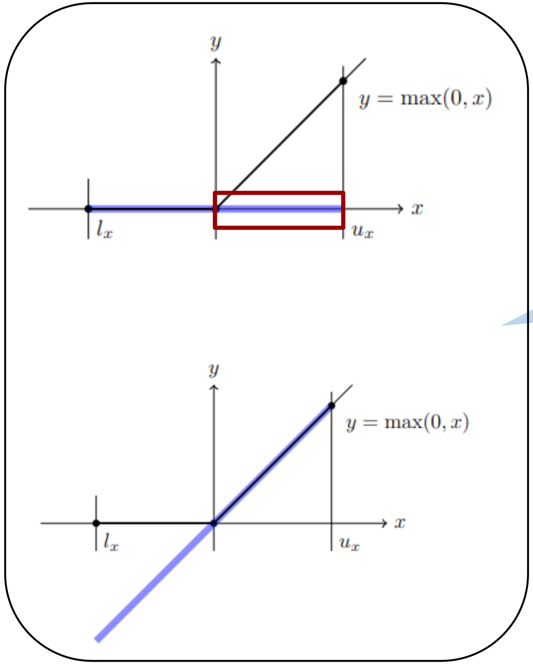
Bounds actually worse in this example

Naive splitting is not strictly better!

Original approximation

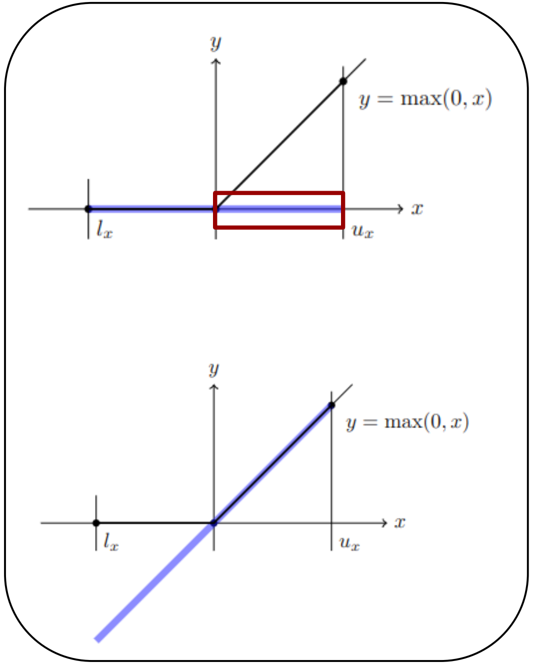


New approximation

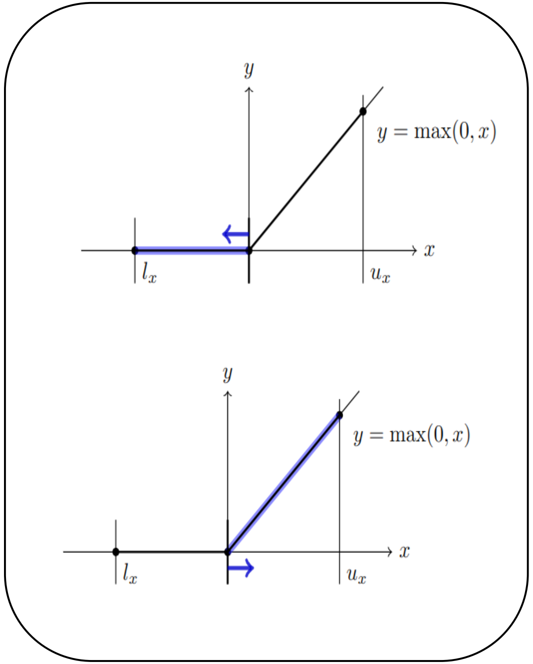


New approximation is non-comparable: it removes certain region and introduces other (red) region, top.

Naive Splitting



Wanted



Strictly better approximation

How to enforce constraints on input?

Enforce Constraints with KKT Condition

$$\begin{aligned} \max_{\mathbf{x}\in\mathcal{X}} \quad & f(\mathbf{x}) \\ s.t. \quad & g(\mathbf{x}) \leq 0 \end{aligned} \qquad \leq \qquad \max_{\mathbf{x}\in\mathcal{X}} \min_{\beta\geq 0} \quad f(\mathbf{x}) - \beta g(\mathbf{x})$$

Intuition:

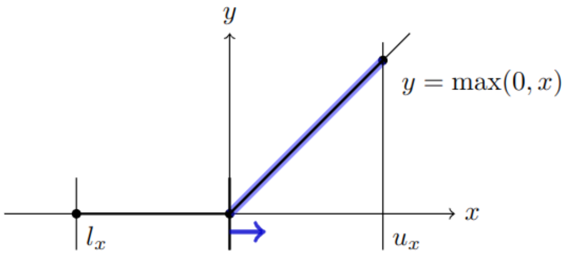
If the constraint $g(x) \leq 0$ is violated for a fixed x, we have

$$g(x) > 0 \Rightarrow f(x) - \beta * g(x) \ \rightarrow -\infty \ \text{ for } \ \beta \rightarrow \infty$$

thus "incentivising" the choice of a different x.

*KKT for Karush–Kuhn–Tucker

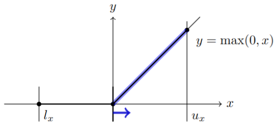
Enforce Split Constraints with KKT



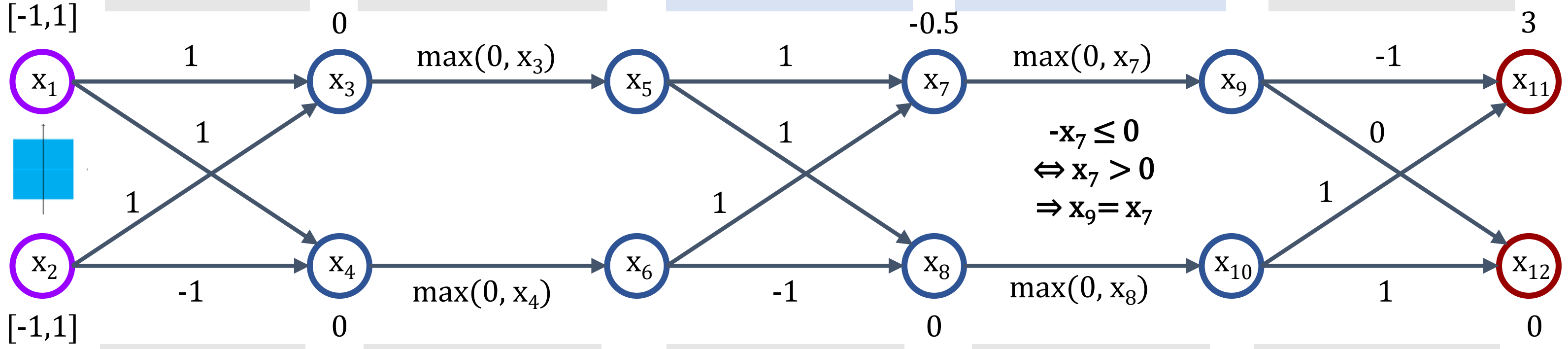
$$\max_{\mathbf{x}\in\mathcal{X}} \quad \mathbf{a}\mathbf{x} + c \qquad \le \qquad \max_{\mathbf{x}\in\mathcal{X}} \min_{\beta\ge 0} \quad \mathbf{a}\mathbf{x} + c + \beta x_i$$

$$s.t. \quad -x_i \le 0$$

Positive Split on x_7



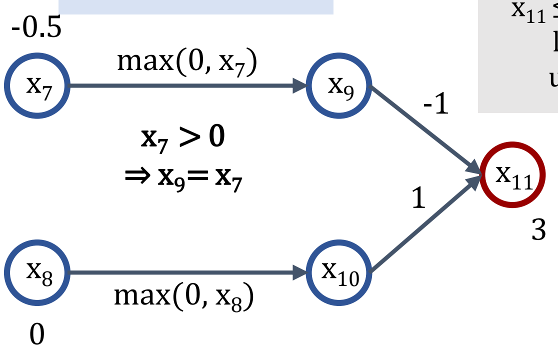
$x_3 \geq x_1+x_2$ $x_3 \leq x_1+x_2$ $l_3 = -2$ $u_3 = 2$	$x_5 \geq 0$ $x_5 \leq 0.5\ x_3+1$ $l_5 = 0$ $u_5 = 2$	$x_7 \geq x_5+x_6 - 0.5$ $x_7 \leq x_5+x_6 - 0.5$ $\mathbf{l_7 = 0}$ $u_7 = 2.5$	$x_9 \geq x_7$ $\mathbf{x_9 \leq x_7}$ $l_9 = 0$ $u_9 = 2.5$	$x_{11} \geq x_{10} - x_9+3$ $x_{11} \leq x_{10} - x_9+3$ $l_{11} = 0.5$ $u_{11} = 4.5$



$x_4 \geq x_1 - x_2$ $x_4 \leq x_1 - x_2$ $l_4 = -2$ $u_4 = 2$	$x_6 \geq 0$ $x_6 \leq 0.5\ x_4+1$ $l_6 = 0$ $u_6 = 2$	$x_8 \geq x_5 - x_6$ $x_8 \leq x_5 - x_6$ $l_8 = -2$ $u_8 = 2$	$x_{10} \geq 0$ $x_{10} \leq 0.5\ x_8+1$ $l_{10} = 0$ $u_{10} = 2$	$x_{12} \geq x_{10}$ $x_{12} \leq x_{10}$ $l_{12} = 0$ $u_{12} = 2$

Backsubstitution with Split

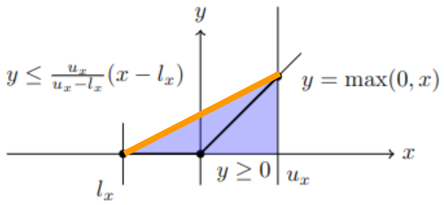
$x_9 \geq x_7$
$x_9 \leq x_7$
$l_9 = 0$
$u_9 = 2.5$



$x_{11} \geq x_{10} - x_9 + 3$
$x_{11} \leq x_{10} - x_9 + 3$
$l_{11} = 0.5$
$u_{11} = 4.5$

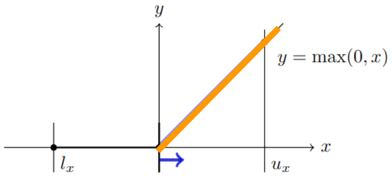
$x_{11} \leq x_{10} - x_9 + 3$

pos. coeff. ⇒ upper bound



Substitute
$\mathbf{x_{10} \leq 0.5\, x_8 + 1}$

neg. coeff. ⇒ lower bound



Substitute
$\mathbf{x_9 \geq x_7}$
Enforce
$\mathbf{x_7 > 0}$

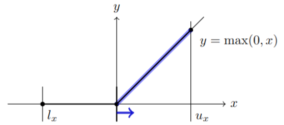
$x_{10} \geq 0$
$\mathbf{x_{10} \leq 0.5\, x_8 + 1}$
$l_{10} = 0$
$u_{10} = 2$

$\max_x x_{11}$ (s.t. $x_7 > 0$) $\leq \max_x x_{10} - x_9 + 3$ (s.t. $x_7 > 0$)
$\leq \max_x 0.5\, x_8 - x_7 + 4$ **(s.t. $x_7 > 0$)**
$\leq \max_x \mathbf{\min_{\beta \geq 0}}\, 0.5\, x_8 - x_7 + 4 + \boldsymbol{\beta x_7}$

We hide $\max_x$ for readability

KKT

Positive Split on x_7

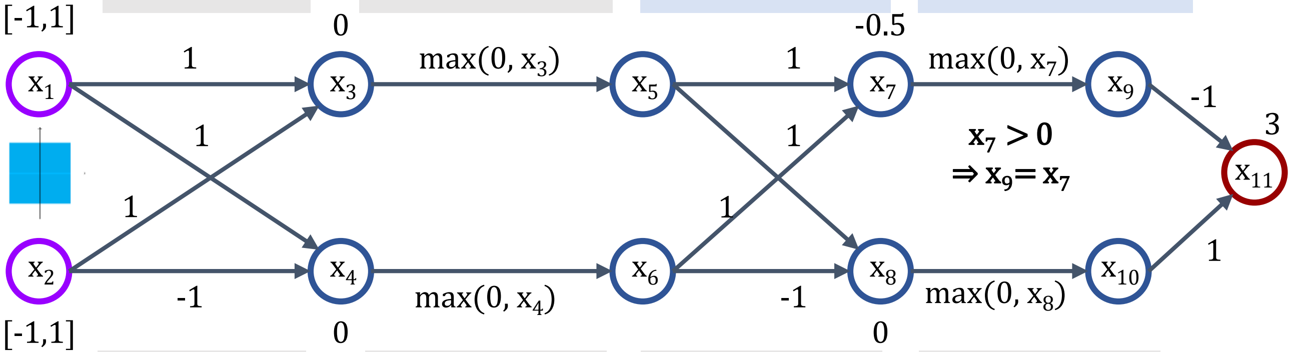


$x_3 \geq x_1 + x_2$
$x_3 \leq x_1 + x_2$
$l_3 = -2$
$u_3 = 2$

$\mathbf{x_5 \geq 0}$
$x_5 \leq 0.5\,x_3 + 1$
$l_5 = 0$
$u_5 = 2$

$x_7 \geq x_5 + x_6 - 0.5$
$\mathbf{x_7 \leq x_5 + x_6 - 0.5}$
$l_7 = 0$
$u_7 = 2.5$

$x_9 \geq x_7$
$\mathbf{x_9 \leq x_7}$
$l_9 = 0$
$u_9 = 2.5$



$x_{11} \leq x_{10} - x_9 + 3$
$\leq \min_\beta 0.5\,x_8 - x_7 + 4 + \boldsymbol{\beta x_7}$
$\leq \min_\beta (\beta - 0.5)\,x_5 + (\beta - 1.5)x_6 + 4.5 - 0.5\beta$

$\boldsymbol{\beta \leq 0.5}$

$\min_\beta (\beta - 0.5)\,x_5 + (\beta - 1.5)x_6 + 4.5 - 0.5\beta$
$\leq \min_\beta 4.5 - 0.5\beta$
$= 4.25$

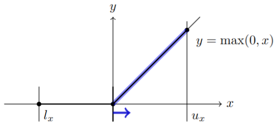
$x_4 \geq x_1 - x_2$
$x_4 \leq x_1 - x_2$
$l_4 = -2$
$u_4 = 2$

$\mathbf{x_6 \geq 0}$
$x_6 \leq 0.5\,x_4 + 1$
$l_6 = 0$
$u_6 = 2$

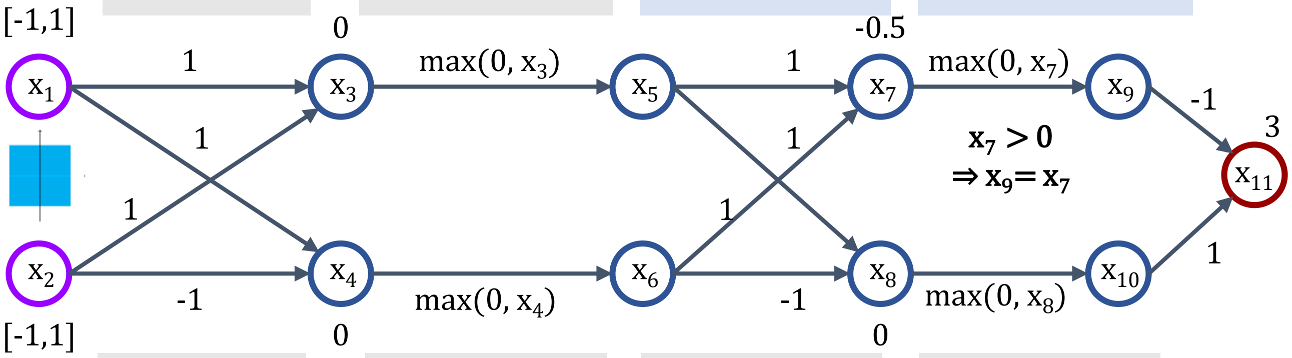
$x_8 \geq x_5 - x_6$
$\mathbf{x_8 \leq x_5 - x_6}$
$l_8 = -2$
$u_8 = 2$

$x_{10} \geq 0$
$\mathbf{x_{10} \leq 0.5\,x_8 + 1}$
$l_{10} = 0$
$u_{10} = 2$

Positive Split on x_7



$x_3 \geq x_1+x_2$ $\mathbf{x_3 \leq x_1+x_2}$ $l_3 = -2$ $u_3 = 2$	$x_5 \geq 0$ $\mathbf{x_5 \leq 0.5\, x_3+1}$ $l_5 = 0$ $u_5 = 2$	$x_7 \geq x_5+x_6 - 0.5$ $\mathbf{x_7 \leq x_5+x_6 - 0.5}$ $l_7 = 0$ $u_7 = 2.5$	$x_9 \geq x_7$ $\mathbf{x_9 \leq x_7}$ $l_9 = 0$ $u_9 = 2.5$



$x_4 \geq x_1 - x_2$ $x_4 \leq x_1 - x_2$ $l_4 = -2$ $u_4 = 2$	$\mathbf{x_6 \geq 0}$ $x_6 \leq 0.5\, x_4+1$ $l_6 = 0$ $u_6 = 2$	$x_8 \geq x_5 - x_6$ $\mathbf{x_8 \leq x_5 - x_6}$ $l_8 = -2$ $u_8 = 2$	$x_{10} \geq 0$ $\mathbf{x_{10} \leq 0.5\, x_8+1}$ $l_{10} = 0$ $u_{10} = 2$

$$x_{11} \leq x_{10} - x_9 + 3$$
$$\leq \min_\beta 0.5\, x_8 - x_7 + 4 + \beta x_7$$
$$\leq \min_\beta (\beta - 0.5)\, x_5 + (\beta - 1.5) x_6 + 4.5 - 0.5\beta$$

$\beta \leq 0.5$

$$\min_\beta (\beta - 0.5)\, x_5 + (\beta - 1.5) x_6 + 4.5 - 0.5\beta$$
$$\leq \min_\beta 4.5 - 0.5\beta$$
$$= 4.25$$

$1.5 \geq \beta \geq 0.5$

$$\min_\beta (\beta - 0.5)\, x_5 + (\beta - 1.5) x_6 + 4.5 - 0.5\beta$$
$$\leq \min_\beta (0.5\,\beta - 0.25)\, x_3 + 4 + 0.5\beta$$
$$\leq \boxed{\min_\beta (0.5\,\beta - 0.25)\,(x_1+x_2) + 4 + 0.5\beta}$$
$$\leq \min_\beta 3.5 + 1.5\beta$$
$$= 4.25$$

Next, lets see how we simplify this expression in more detail

$\max_x \min_\beta (0.5\,\beta - 0.25)\,(x_1+x_2) + 4 + 0.5\beta$

$\leq \min_\beta \max_x (0.5\,\beta - 0.25)\,(x_1+x_2) + 4 + 0.5\beta$ **via** weak duality

$= \min_\beta \max_x (0.5\,\beta - 0.25, 0.5\,\beta - 0.25)\ (x_1, x_2)^T + 4 + 0.5\beta$ **via** rewrite

$\leq \min_\beta ||(0.5\,\beta - 0.25,\ 0.5\,\beta - 0.25)||_1 * 1 + 4 + 0.5\beta$ **via** Holder $q = 1$

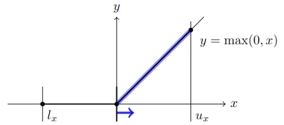
$= \min_\beta 1\,\beta - 0.5 + 4 + 0.5\beta$ **via** $1.5 \geq \beta \geq 0.5$

$= \min_\beta 3.5 + 1.5\beta$

$= 4.25$ **with** $\beta = 0.5$

Here solved symbolically, typically using gradient descent

Positive Split on x_7

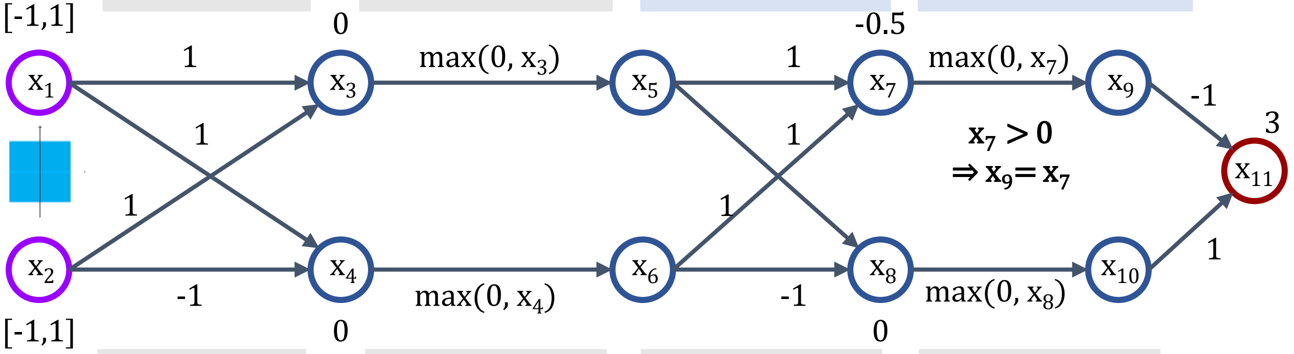


$x_3 \geq x_1+x_2$
$x_3 \leq x_1+x_2$
$l_3 = -2$
$u_3 = 2$

$x_5 \geq 0$
$x_5 \leq 0.5\, x_3+1$
$l_5 = 0$
$u_5 = 2$

$x_7 \geq x_5+x_6 - 0.5$
$x_7 \leq x_5+x_6 - 0.5$
$l_7 = 0$
$u_7 = 2.5$

$x_9 \geq x_7$
$x_9 \leq x_7$
$l_9 = 0$
$u_9 = 2.5$



$x_4 \geq x_1 - x_2$
$x_4 \leq x_1 - x_2$
$l_4 = -2$
$u_4 = 2$

$x_6 \geq 0$
$x_6 \leq 0.5\, x_4+1$
$l_6 = 0$
$u_6 = 2$

$x_8 \geq x_5 - x_6$
$x_8 \leq x_5 - x_6$
$l_8 = -2$
$u_8 = 2$

$x_{10} \geq 0$
$x_{10} \leq 0.5\, x_8+1$
$l_{10} = 0$
$u_{10} = 2$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq \min_\beta 0.5\, x_8 - x_7 + 4 + \beta x_7$
$\leq \min_\beta (\beta-0.5)\, x_5 + (\beta-1.5) x_6 + 4.5 - 0.5\beta$

$\beta \leq 0.5$

$\min_\beta (\beta-0.5)\, x_5 + (\beta-1.5) x_6 + 4.5 - 0.5\beta$
$\leq \min_\beta 4.5 - 0.5\beta$
$= 4.25$

$1.5 \geq \beta \geq 0.5$

$\min_\beta (\beta-0.5)\, x_5 + (\beta-1.5) x_6 + 4.5 - 0.5\beta$
$\leq \min_\beta (0.5\,\beta - 0.25)\, x_3 + 4 + 0.5\beta$
$\leq \min_\beta (0.5\,\beta - 0.25)\, (x_1+x_2) + 4 + 0.5\beta$
$\leq \min_\beta 3.5 + 1.5\beta$
$= 4.25$

$\beta \geq 1.5$

$\min_\beta (\beta-0.5)\, x_5 + (\beta-1.5) x_6 + 4.5 - 0.5\beta$
$\leq \min_\beta (0.5\,\beta - 0.25)\, x_3 + (0.5\,\beta - 0.75)\, x_4 + 2.5 + 1.5\beta$
$\leq \min_\beta (\beta - 1)\, x_1 - 0.5\, x_2 + 2.5 + 1.5\beta$
$\leq \min_\beta 2 + 2.5\beta$
$= 5.75$

Numerical Optimization

Explicit case distinction only for symbolic solve

Actually, we pick numeric value for $\beta_i \Rightarrow$ standard backsubstitution

β_i are concrete numerical values, not variables

Symbolic (Instructive, not actually what runs)

$\beta \leq 0.5$

$\min_\beta (\beta-0.5)\, x_5+(\beta- 1.5)x_6+ 4.5 - 0.5\beta$
$\leq \min_\beta 4.5 - 0.5\beta$

$1.5 \geq \beta \geq 0.5$

$\min_\beta (\beta-0.5)\, x_5+(\beta- 1.5)x_6+ 4.5 - 0.5\beta$
$\leq \min_\beta 3.5 + 1.5\beta$

$\beta \geq 1.5$

$\min_\beta (\beta-0.5)\, x_5+(\beta- 1.5)x_6+ 4.5 - 0.5\beta$
$\leq \min_\beta 2 + 2.5\beta$

Numerical (what runs)

Initialize: $\beta_0 := 0$

$-0.5x_5-1.5x_6+ 4.5$
$\Rightarrow \text{UB} = 4.5 - 0.5\beta_0 = 4.5$

$\nabla_\beta \text{UB} = - 0.5 \quad \Rightarrow \quad \beta_1 := 0.75$

step size: 1.5

$0.25x_5- 0.75x_6+ 4.5 - 0.375$
$\Rightarrow \text{UB} = 3.5 + 1.5\beta_1 = 4.625$

$\nabla_\beta \text{UB} = 1.5 \quad \Rightarrow \quad \beta_2 := 0.375$

step size: 0.25

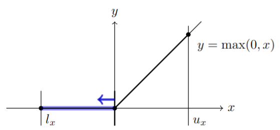
$-0.125x_5- 1.125x_6+ 4.5 - 0.1875$
$\Rightarrow \text{UB} = 4.5 - 0.5\beta_2 = 4.3125$

$\nabla_\beta \text{UB} = - 0.5 \quad \Rightarrow \quad \beta_3 := 0.5$

step size: 0.25

$- x_6+ 4.5 - 0.25$
$\Rightarrow \text{UB} = 4.5 - 0.5\beta_3 = 4.25$

Negative Split on x_7



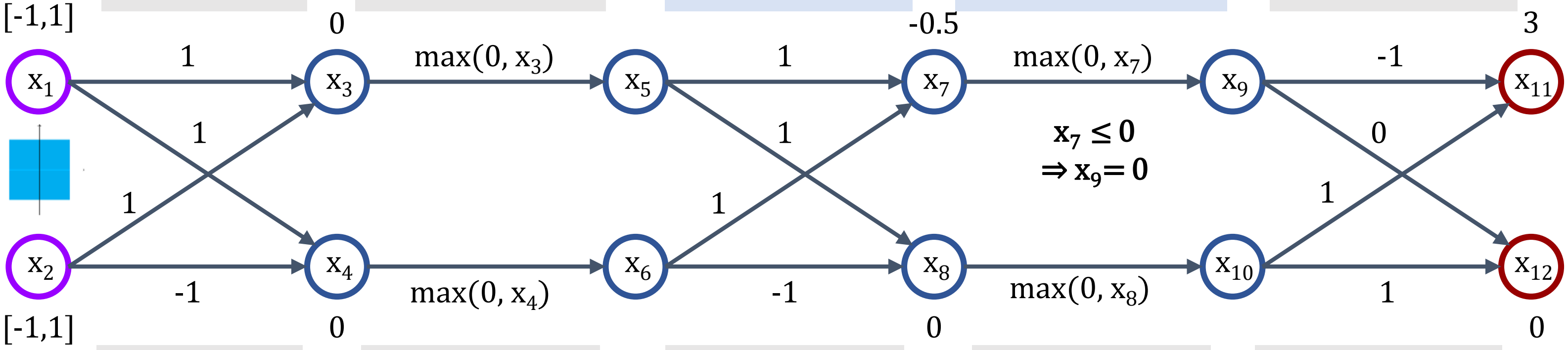
$x_3 \geq x_1+x_2$
$x_3 \leq x_1+x_2$
$l_3 = -2$
$u_3 = 2$

$x_5 \geq 0$
$x_5 \leq 0.5\ x_3+1$
$l_5 = 0$
$u_5 = 2$

$x_7 \geq x_5+x_6- 0.5$
$x_7 \leq x_5+x_6- 0.5$
$l_7 = -0.5$
$\mathbf{u_7 = 0}$

$\mathbf{x_9 \geq 0}$
$\mathbf{x_9 \leq 0}$
$l_9 = 0$
$u_9 = 0$

$x_{11} \geq x_{10}- x_9+3$
$x_{11} \leq x_{10}- x_9+3$
$l_{11} = 0.5$
$u_{11} = 4.5$



$x_4 \geq x_1- x_2$
$x_4 \leq x_1- x_2$
$l_4 = -2$
$u_4 = 2$

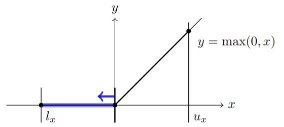
$x_6 \geq 0$
$x_6 \leq 0.5\ x_4+1$
$l_6 = 0$
$u_6 = 2$

$x_8 \geq x_5- x_6$
$x_8 \leq x_5- x_6$
$l_8 = -2$
$u_8 = 2$

$x_{10} \geq 0$
$x_{10} \leq 0.5\ x_8+1$
$l_{10} = 0$
$u_{10} = 2$

$x_{12} \geq x_{10}$
$x_{12} \leq x_{10}$
$l_{12} = 0$
$u_{12} = 2$

Negative Split on x_7



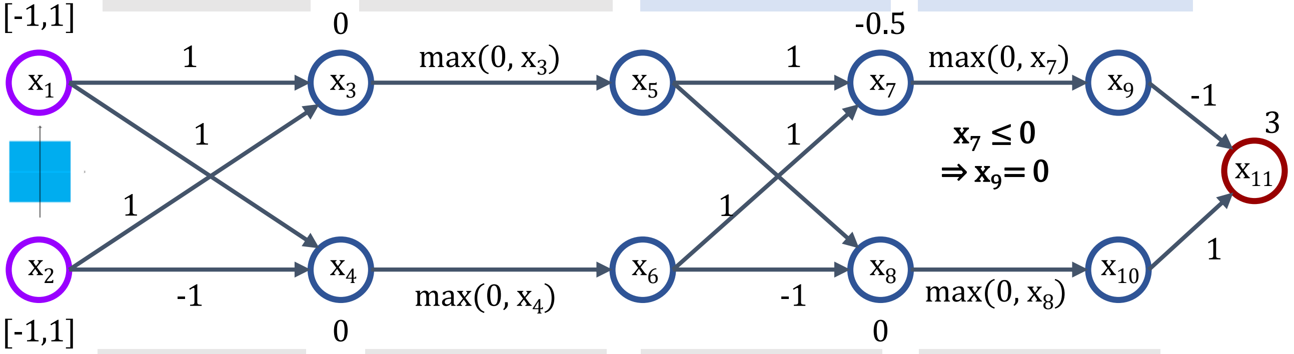
$x_3 \geq x_1 + x_2$
$x_3 \leq x_1 + x_2$
$l_3 = -2$
$u_3 = 2$

$x_5 \geq 0$
$x_5 \leq 0.5\, x_3 + 1$
$l_5 = 0$
$u_5 = 2$

$x_7 \geq x_5 + x_6 - 0.5$
$x_7 \leq x_5 + x_6 - 0.5$
$l_7 = -0.5$
$\mathbf{u_7 = 0}$

$\mathbf{x_9 \geq 0}$
$\mathbf{x_9 \leq 0}$
$l_9 = 0$
$u_9 = 0$

$x_{11} \leq x_{10} - x_9 + 3$
$\leq \min_\beta 0.5\, x_8 + 4 - \beta x_7$
$\leq \min_\beta (0.5-\beta)\, x_5 + (-\beta - 0.5)\, x_6 + 4 + 0.5\beta$



$\beta \leq 0.5$

$\min_\beta (0.5-\beta)\, x_5 + (-\beta - 0.5)\, x_6 + 4 + 0.5\beta$
$\leq \min_\beta (0.5-\beta)\, x_5 + 4 + 0.5\beta$
$\leq \min_\beta (0.25 - 0.5\beta)\, x_3 + 4.5 - 0.5\beta$
$\leq \min_\beta (0.25 - 0.5\beta)\, (x_1 + x_2) + 4.5 - 0.5\beta$
$= 4.25$

$\beta \geq 0.5$

$\min_\beta (0.5-\beta)\, x_5 + (-\beta - 0.5)\, x_6 + 4 + 0.5\beta$
$\leq \min_\beta 4 + 0.5\beta$
$= 4.25$

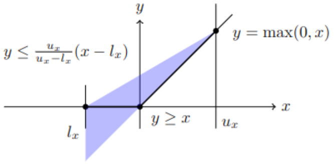
$x_4 \geq x_1 - x_2$
$x_4 \leq x_1 - x_2$
$l_4 = -2$
$u_4 = 2$

$x_6 \geq 0$
$x_6 \leq 0.5\, x_4 + 1$
$l_6 = 0$
$u_6 = 2$

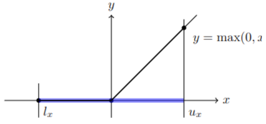
$x_8 \geq x_5 - x_6$
$x_8 \leq x_5 - x_6$
$l_8 = -2$
$u_8 = 2$

$x_{10} \geq 0$
$x_{10} \leq 0.5\, x_8 + 1$
$l_{10} = 0$
$u_{10} = 2$

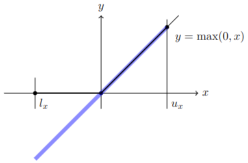
Comparison of Tightness



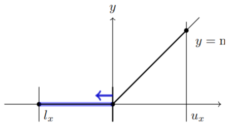
$x_{11} \leq 4.5$



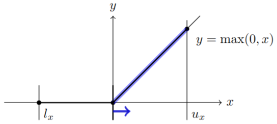
$x_{11} \leq 5.0$



$x_{11} \leq 4.5$



$x_{11} \leq 4.25$



$x_{11} \leq 4.25$

$x_{11} \leq 4.5$

$\boxed{x_{11} \leq 5.0}$

$\boxed{x_{11} \leq 4.25}$

No Split

Naive Split

Full Split

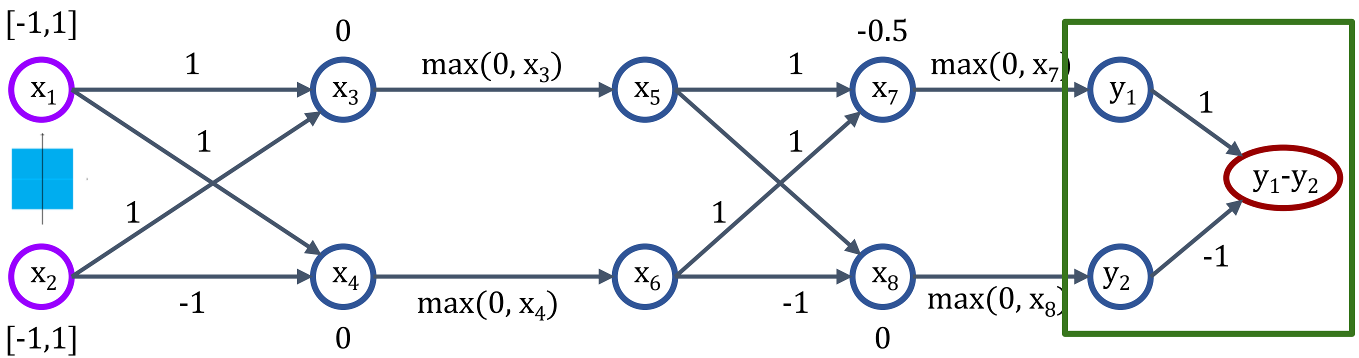
Note: for verification, we need to consider the worst-case over all splits. This implicitly means that for upper bounds, we need to consider the max over all splits, and for lower bounds we need to consider the min over all splits.

Verification as Optimization

Given a neural network $\mathbf{y} = \mathrm{N}(\mathrm{x})$ and target label t, we can show correctness for some input region by proving:

$$y_i - y_t < 0 \quad \forall\, i \neq t$$

We can encode this as an extra layer:



Efficient Optimization

We start backsubstitution with a linear expression, e.g. y_i - y_t to determine whether $y_i < y_t$

$$\max_{\mathbf{x}^0 \in \mathcal{X}} \quad \mathbf{a}\mathbf{x}^L + c$$

$$s.t. \quad x^{i+1} = \sigma(\mathbf{W}\mathbf{x}^i + b) \quad \forall i \in [L]$$

$$\leq \max_{\mathbf{x}^0 \in \mathcal{X}} \min_{\beta \geq 0} \quad \mathbf{a}\mathbf{x}^0 + \mathbf{b}\boldsymbol{\beta} + c$$

Backsubstitution (**a** is function of β)

$$\leq \min_{\beta \geq 0} \max_{\mathbf{x}^0 \in \mathcal{X}} \quad \mathbf{a}\mathbf{x}^0 + \mathbf{b}\boldsymbol{\beta} + c$$

Weak Duality

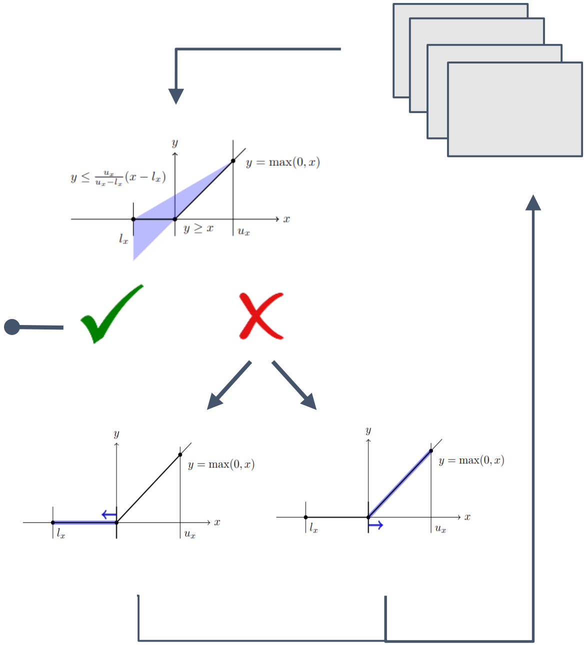
$$\leq \min_{\beta \geq 0} \quad ||\mathbf{a}||_q \epsilon + \mathbf{a}\mathbf{x}_0 + \mathbf{b}\boldsymbol{\beta} + c$$

Hölder's inequality for $\mathcal{X} = \{\mathbf{x} \in \mathcal{D} \mid ||\mathbf{x} - \mathbf{x}'||_p \leq \epsilon\}$ with $\frac{1}{p} + \frac{1}{q} = 1$

Now, we use **standard gradient descent** to optimise β, doing every backsubstitution for a **concrete numerical value**.

End-to-end branch and bound verification

- Initialize queue with full verification problem (no splits)
- While queue is not empty and not timed out do:
    - Get subproblem from queue
    - Compute bound of interest
    - If not verified, pick neuron to split according to heuristic
    - Add both new subproblems to queue



Lecture Summary (Part I)

- Handling arbitrary input norms for convex relaxations

- A method to refine the results of DeepPoly by combining with KKT (obtaining a differentiable version, not possible with standard LP).

Questions after lecture

Can we split on x7, get 2 upper bounds from both splits (+ and -) and instead of max, propagate further and then take max later?

Once you split on x7, you do not explicitly merge (max or min). You just have 2 separate optimization problems and you solve them. If one of them is not verified, we consider the entire problem not verified (so in a sense we are performing an implicit min.). Now, as long as there is a non-verified instance, we will refine that non-verified instance by splitting.

How many optimization problems do we get in the worst case given a decision to split N neurons?

In the worst-case it is 2^N (exponential), but in practice we can rule out many branches early on as the corresponding instances are verified.

If we split on all neurons, that is for N neurons we have 2^N instances, and solve each KKT instance exactly, is this method complete?

Yes, as strong duality holds, see section 5.2.3 here: https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

In a complete split, can we solve KKT exactly?

Yes, subject to finding a learning rate for gradient descent.

How does completeness arise in practice?

In practice, If a particular split is verified, we stop further splitting, and if it doesn't, we keep splitting up to a time out. So we usually time out long before we split all neurons.

Can p and q be non-integers?

Yes, now its clarified in slide 5

Scale of deterministic verification: VNNCOMP'22

	Name	Network Type	# Parms	# Neurons	Input Dim	Domain
Complex	Carvana UNet	Complex U-Net	150k - 330 k	275k - 373k	5828	BaB* with DeepPoly
	VGGNet 16	Conv + ReLU + MaxPool	138M	13.6 M	164k	Box + DeepPoly
CNN / ResNet	Cifar Biasfield	Conv + ReLU	363k	45k	16	BaB* with DeepPoly
	Large ResNet	ResNet (Conv + ReLU)	1.3M - 7.9M	55k - 286k	3k-9k	BaB* with DeepPoly
	Collins Rul CNN	Conv + ReLU	60k - 262k	5.5k - 28k	400-800	BaB* with DeepPoly
	oval21	Conv + ReLU	54k - 214k	3.1k - 6.2k	3072	BaB* with DeepPoly
	ResNet A/B	ResNet (Conv + ReLU)	354k	11k	3072	BaB* with DeepPoly
FC	MNIST FC	FC + ReLU	270k - 530k	512 - 1536	784	BaB* with DeepPoly + MILP refinement

* BaB is implemented via KKT

Note: Before using more expensive methods, we always try cheaper methods. We try *Box, Box(intermediate) + DeepPoly (final bounds), full DeepPoly, DeepPoly with slope optimization, BaB with DeepPoly* in this order.