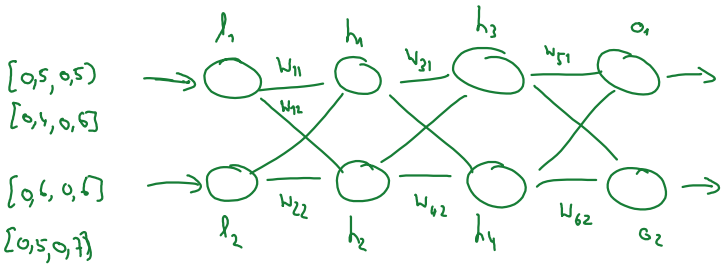
Project Tutorial

Wednesday, November 14, 2018 11:16

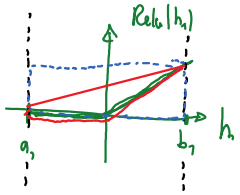


$Score(o_1) > Score(o_2)$

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \begin{matrix} \nearrow [0.4, 0.6] \\ \searrow [0.5, 0.7] \end{matrix} \rightarrow \begin{bmatrix} [a_1, b_1] \\ [a_2, b_2] \end{bmatrix}$$

$h \rightarrow Relu(h)$

$Relu(h) = \max(0, h)$



We'd like to encode

$(h \leq 0 \wedge Relu(h) = 0) \vee (h > 0 \wedge Relu(h) = h)$

↖ Problem for linear programming is this ∨ (disjunction)

$a_1 \leq h_1 \leq b_1$

One approximation:

$-\infty \leq ReLU(h_1) \leq \infty$

Another:

- $0 \leq ReLU(h_1) \leq b_1$ (still quite a crude approx.)

Another:

- $ReLU(h_1) \geq 0$
- $ReLU(h_1) \geq h_1$
- $ReLU(h_1) \leq \lambda_1 h_1 + \mu_1$ $\quad \lambda_1 = \frac{b_1}{b_1 - a_1}$, $\mu_1 = \frac{-a_1 b_1}{b_1 - a_1}$

(best approx. in linear programming) (?)

For one Neuron deeper in: h_3:

$h_3 = w_{31} ReLU(h_1) + w_{41} ReLU(h_2)$

$$\begin{cases} ReLU(h_1) \geq 0 \\ \geq h_1 \\ \leq \lambda_1 h_1 + \mu_1 \end{cases} \qquad ReLU(h_2) \begin{matrix} \geq 0 \\ \geq h_2 \\ \leq \lambda_2 h_2 + \mu_2 \end{matrix}$$

$$\begin{cases} h_1 = \end{cases}$$

1. Run interval analysis
2. Use bounds to formulate lin. programs
3. Obj. function: minimize: $\min o_1 - o_2$ (*) s.t. $o_1 = \sum w_{ij} ReLU(h_j)$, $o_2 = \sum w_{ij} ReLU(h_j)$, $ReLU(h_3) \begin{matrix} \geq 0 \\ \geq h_3 \\ \leq \lambda_3 h_3 + \mu_3 \end{matrix}$ ...

4. → Check sign of $o_1 - o_2$.
   (Search for opt. value of $o_1 - o_2$)

→ At some points do interval analysis & at other goal better bounds

e.g. optimize $\min h_3$, $\max h_3$ ⊛
→ Then call linear solver better

Project goal:

Find out when to (1) Do interval analysis (possibilities: - Split dep. on layers (inter layer)
- Split dep. on neurons in layer (intra layer)
(2) Linear solver
- time bound on linear solver
(e.g. already a slightly better estimate might be sufficient rather than get optimal value)

⇝ Find out adaptively if bounds are good
→ if LB & UB negative → interval exact.

⇝ Try not to use lin. solver in deeper layers, better use it in first few layers

→ ~ 100 Runs
(1 Run: 1 epsilon, 1 net, 1 img. - 7 min)