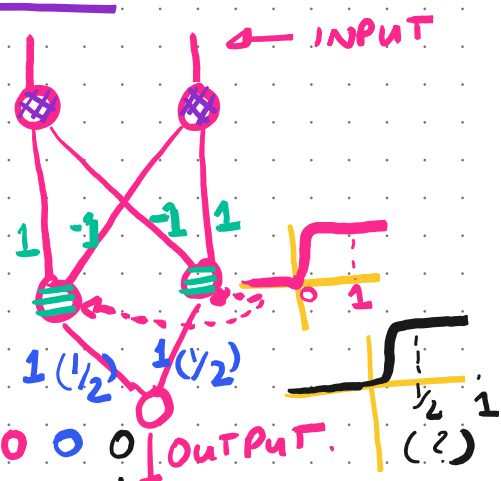
We are writing our first new page!!

purple and thicker

NOTES ON x-OR

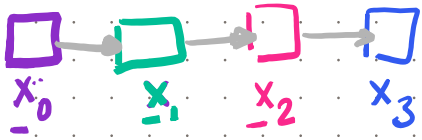
0	0	0
0	1	1
1	0	1
1	1	0



0	0	0 | 0	0 | 0	0
0	1	-1 | 1	0 | 1	1
1	0	1 | -1	1 | 0	1
1	1	0 | 0	0 | 0	0

How To Design the Training algorithm?

Multistage Two-Point Boundary Value Problems



$$x_0 = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$ ⟸ Two "bits" (values are continuous!)

$$x_1 = W_0' x_0$$

$$x_2 = g(x_1)$$

$$x_3 = W_2' x_2$$

INPUT

x_0, y_3 ⟸ Training sample

OBJECTIVE

$$J := \frac{1}{2}(y_3 - x_3)^2 + \lambda_3(x_3 - W_2' x_2) + \lambda_2'(x_2 - g(x_1)) + \lambda_1'(x_1 - W_0' x_0)$$

What are the parameters to J?

$$\underline{\lambda} = \begin{bmatrix} \lambda_3 \\ \lambda_2 \\ \lambda_1 \end{bmatrix} \quad \underline{X} = \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

$$\underline{w} = \begin{bmatrix} w_{0(11)} \\ w_{0(12)} \\ w_{0(21)} \\ w_{0(22)} \\ w_{2(1)} \\ w_{2(2)} \end{bmatrix}$$

$\Rightarrow$ We applied the variational principle $\Rightarrow$

The "System dynamics" provides the nominal solution, the variation around which is our true solution

$\underline{\lambda} \Rightarrow$ Lagrange Multiplier

How to solve?

① $\frac{dJ}{dx_3} = -(y_3 - x_3) + \lambda_3 = 0$

$\Rightarrow \boxed{\lambda_3 = y_3 - x_3}$

② $\frac{\partial J}{\partial w_2} = -\lambda_3 x_2$

③ $\frac{\partial J}{\partial x_2} = \lambda_2 - \lambda_3 w_2 = 0$

$\boxed{\lambda_2 = w_2 \lambda_3}$

④ $\frac{\partial J}{\partial x_1} = \lambda_1 - \left(\frac{\partial}{\partial x_1}[g(x_1)]\right)'\lambda$

$= 0$

$$g(\underline{x}_1) = \begin{bmatrix} g_1(\underline{x}_1) \\ g_2(\underline{x}_1) \end{bmatrix}$$

$$\underline{\underline{\nabla g}} = \begin{bmatrix} \partial g_1/\partial x_{1,1} & \partial g_1/\partial x_{1,2} \\ \partial g_2/\partial x_{1,1} & \partial g_2/\partial x_{1,2} \end{bmatrix}$$

$$\Rightarrow \boxed{\underline{\lambda}_1 = (\underline{\underline{\nabla g}})' \underline{\lambda}_2}$$

(5) $$\frac{\partial J}{\partial \underline{\underline{\omega}}_0} = \left[\frac{\partial J}{\partial \underline{\underline{\omega}}_0}\right]_{i,i} = \begin{bmatrix} \partial J/\partial \omega_{0,1\to 1} & \partial J/\partial \omega_{0,1\to 2} \\ \partial J/\partial \omega_{0,2\to 1} & \partial J/\partial \omega_{0,2\to 2} \end{bmatrix}$$

$$= -\begin{bmatrix} \lambda_{1,1} x_{0,1} & \lambda_{1,2} x_{0,1} \\ \lambda_{1,1} x_{0,2} & \lambda_{1,2} x_{0,2} \end{bmatrix}$$

$$= \boxed{-\underline{x}_0 \underline{\lambda}_1'}$$

We don't (yet) care about $\partial J / \partial \underline{x}_0$

Putting things together

(A) Weight gradients:

$$\frac{\partial J}{\partial \underline{\underline{w}}_2} = -\lambda_3 \underline{x}_2$$

$$\frac{\partial J}{\partial \underline{\underline{w}}_0} = -\underline{x}_0 \underline{\lambda}_1'$$

(B) Normal Equation from Variational Solution -

$$\lambda_3 = (y_3 - x_3)$$

$$\underline{\lambda}_2 = \underline{\underline{w}}_2 \lambda_3$$

Back Propagation

$$\underline{\lambda}_1 = (\underline{\underline{\nabla g}})' \underline{\lambda}_2$$

© System Dynamics

$$\frac{\partial J}{\partial \underline{\lambda}_1} = 0 \Rightarrow$$

$$\underline{x}_1 = \underline{W}_0' \underline{x}_0$$

$$\frac{\partial J}{\partial x_2} = 0 \Rightarrow$$

$$\underline{x}_2 = g(\underline{x}_1)$$

$$\frac{\partial J}{\partial x_3} = 0 \Rightarrow$$

$$x_3 = w_2' x_2$$

Forward Propagation

Algorithm (FORWARD-BACKWARD)

INPUT: x_0, y_3

1 → System Dynamics

$$x_1 = w_0' x_0$$
$$x_2 = g(x_1)$$
$$x_3 = w_2' x_2$$

2 → Back Propagation

$$\lambda_3 = (y_3 - x_3)$$

$$\lambda_2 = w_2 \lambda_3$$

$$\lambda_1 = (\sigma g)' \lambda_2$$

③ → Calc. Parameter Gradients (Weights)

$$\frac{\partial J}{\partial w_2} = -\lambda_3 x_2$$

$$\frac{\partial J}{\partial \underline{w}_0} = -x_0 \lambda_1'$$

④ → Apply gradients to weight update (e.g.)

$$\underline{w}_0^{new} = \underline{w}_0^{old} - \alpha \frac{\partial J}{\partial \underline{w}_0}$$

$$w_2^{new} = w_2^{old} - \alpha \frac{\partial J}{\partial w_2}$$

(5) → Repeat (1) until Convergence

We will need a more Sophisticated Optimization Strategy, however, the Forward-Backward algorithm is classical.

Consider the Dynamical System.

$$X_{n+1} = F(X_n)$$

$$y_n = H X_n + \vartheta$$

The process of assimilating data for state $(x_{0:n})$ or parameter (α) is identical.

⇒ The Adjoint Method or 4D VAR.