MATLAB Simulation for Gradient - Based Neural Networks for Online Matrix Inversion ...

To Solve the Matrix Inverse, the neural system design is based on the Equation

$$\boxed{Ax - I = 0}$$

where..,

$A \in \mathbb{R}^{n\times n}$ (matrix)

$X \in$ Unknown matrix

$I \in$ ~~Inverse~~ ~~Identity~~ ~~matrix~~. Identity Matrix

→ adding a Scalar-Valued energy function.

Takes one (or) more values but returns a single value

$$\mathcal{E}(t) = \|Ax(t) - I\|^2/2$$

(Time) → (time varying Matrix)

→ Taking Negative of the "gradient" as the descent direction

$$\boxed{\partial \mathcal{E}/\partial x = A^T(Ax(t) - I)}$$

Thus, the classic linear equation becomes

$$\boxed{\dot{x}(t) = -\gamma \frac{\partial \mathcal{E}}{\partial x} = -\gamma A^T(Ax(t) - I)}, \quad x(0) = x_0. \quad (1)$$

where γ = design parameter > 0

By adding an non-linear Activation function array "F" we can change the (1) into an "general Neural Model"

$$\boxed{\dot{x}(t) = -\gamma A^T F(Ax(t) - I)} \rightarrow (2)$$

HYBRID ZNN...

In Robotics Matrix Inversion is considered one of t
most important Aspect. It takes considerably
more time to inverse a matrix (i.e) Thrice the
Original Size of the matrix. Our Aspect is to minimi
the time to get effective Matrix Inversion.

Zhang Neural Network (ZNN) which is a part of RNN
Created by Zhang in 2001 deals with this problem of
time-varying Matrices. The GNN which is also a part of
RNN deals with time invarying Matrix. ZNN is morely
focused on dealing with Error function rather than
Scalar Valued Norm. There are lots of hybrid NN
Created by researchers based on ZNN & GNN.

Our motive is to Create a NN which doesn't restrict
the use of Activation Function. A proper Non linear AF
can increase convergence rate.

Problem Statement :-

The General form of a time - Invarying Matrix inversion is

$$B(t)\,X(t) = I \qquad \text{where } t \in (0, +\infty) \rightarrow (1)$$

where
$$\boxed{\begin{aligned} B(t) &\rightarrow \text{Know Time Varying Matrix} \\ X(t) &\rightarrow \text{Unknown Time Varying Matrix.} \\ I &\rightarrow \text{~~Inverse Mat~~ Identity Matrix.} \end{aligned}}$$

GNN modelling :-

$$\varepsilon(t) = \frac{1}{2}\,\|B(t)\,X(t) - I\|_F^2 \rightarrow (2) \rightarrow \text{Forbenius Norm.}$$

The $\varepsilon(t) \rightarrow 0$ when $x(t)$ converge to "0" exact solution (i.e) $X(t) = \dot{x}(t) = B^{-1}$

where $\boxed{B^{-1} = \text{Theoretical Inverse of the Matrix } B}$

To make $\varepsilon(t) \rightarrow 0$, we use Negative Gradient $-\frac{\partial \varepsilon(t)}{\partial X(t)}$

Thus...

$$\frac{\partial \varepsilon(t)}{\partial X(t)} = B^T(t)\,(B(t)\,X(t) - I) \rightarrow (3)$$

The Designed Formula for GNN is

$$\dot{x}(t) = -\gamma_1 \frac{\partial \varepsilon(t)}{\partial X(t)}$$

$$= -\gamma_1 B^T(t)\,(B(t)\,X(t) - I) \rightarrow (4)$$

adding AF(ϕ)

$$\dot{x}(t) = -\gamma_1 B^T(t)\,\phi\,(B(t)\,X(t) - I) \rightarrow (5)$$

where $\gamma_1 > 0$ (Non Negative Integer)
ϕ = AF (odd Monotically increasing)

ZNN Modelling :-

In this the Error of a time-Varying Matrix Inversion is

where $t \in (0, +\infty)$

$$E(t) = B(t)\,X(t) - I \quad \text{(6)}$$

The Derivative of Error is.

$$E'(t) = -\gamma_2\, \phi(E(t)) \longrightarrow \text{(7)}$$

Sub/. eqn (6) in (7).

$$\Rightarrow \quad B(t)\,\dot{X}(t) = -\dot{B}(t)\,X(t) - \gamma_2\, \phi(B(t)X(t) - I) \quad \text{(8)}.$$

Proposed Model :- (HZNN)

$$HZNN = GNN + ZNN.$$

(ie (5) & (9))

Multiply $B(t)$ in (5).

$$B(t)\,\dot{X}(t) = -\gamma_1\, B(t)\, B^T(t)\, \phi\,(B(t)\,X(t) - I) \quad \text{(9)}$$

Adding (8) & (9) $\boxed{\gamma_1 = \gamma_2}$ Assuming.

$$2(B(t)\,\dot{X}(t)) = -\dot{B}(t)\,X(t) - \gamma\,\phi\,(B(t)\,X(t) - I) - \gamma\, B(t)\, B^T(t)\, \phi\,(B(t)X(t) - I)$$

$$\boxed{= -\dot{B}(t)\,X(t) - \gamma\,(B(t)\,B^T(t) + I)\,\phi\,(B(t)\,X(t) - I)} \quad \text{(10)}.$$

Activation Function :-

We are Using Power Sigmoidal Function.

$$\phi_{ps}(x) = \begin{cases} \frac{1+\exp(-\varsigma)}{1-\exp(-\varsigma)} \cdot \frac{1-\exp(-\varsigma)}{1+\exp(-\varsigma)}, & \text{if } |x|<1 \\ x^{p}, & \text{if } |x| \geq 1. \end{cases}$$

Simulink Model :-

Now, we are going to implement the (10) in the Simulink. A Constant block with a value of B Matrix is Considered to be the input & the Constant Block I Carries the Value of Identity Matrix. To get the transpose of the B, we use transpose Function block & to get the time derivated Value we use derivative Function block. There are Several Operation Such as Summation, integration, Multiplication, differentiation. were Used to design this following Block.

Activation Function :-

We are Using Power Sigmoidal AF where the Value of ς is '4'.