16.82-Neurociencia Computacional
1C 2024

Trabajo Práctico 3:

Adaptado de ***Neural Computation*** del Dr. Xaq Pitkow

Ejercicio 1: Red recurrente lineal y determinista

Dada una red neuronal lineal que cuya dinámica satisface la siguiente ecuación:

$$\dot{\boldsymbol{r}} = -\boldsymbol{r} + W\boldsymbol{r} + \boldsymbol{h}$$

Donde ***r*** es el vector de respuestas neuronales (tasas de disparo), ***W*** es la matriz de pesos sinápticos y ***h(t)*** es una entrada a la red (ruído, estímulos externos, entradas de otras redes, etc.)

Considere una red con dos neuronas, una excitatoria y una inhibitoria[1], que están conectadas recíprocamente con peso de la misma magnitud ***a***.

a. Dibuje la red de dos neuronas (con sus conexiones y respectivos pesos). Usen Paint, no se gasten mucho. (***5 puntos***)
b. ¿Cómo es la matriz de pesos ***W***? ¿Cuáles son sus autovalores? (***5 puntos***)
c. Resuelva para conseguir la dinámica del sistema de manera analítica, partiendo de condición inicial con las dos neuronas activas ($\boldsymbol{r}=(1,1)^T$) y sin ninguna entrada (es decir, ***h=0***). Para esto, calcule los autovalores y autovectores y haga el desarrollo diagonalizando la matriz como vimos en clase[2]. Puede usar Wolfram Alpha para invertir matrices. (***10 puntos***).
d. Grafique la trayectoria del sistema en 2D (con $\boldsymbol{r}_1$ en el eje horizontal y $\boldsymbol{r}_2$ en el eje vertical). (***5 puntos***)
e. Grafique la trayectoria de cada neurona vs tiempo. (***5 puntos***)

Ejercicio 2:

a. Convierta la ecuación diferencial de arriba a su versión en tiempo discreto de la forma:

$$\boldsymbol{r}_{t+dt} = B\boldsymbol{r}_t + \boldsymbol{h}_t$$

relacionando ***B*** a ***W*** (asumiendo que ***dt<<1*****).** (***5 puntos***)

b. Simule el sistema, muestreando ruido de $\boldsymbol{h}_t$ de una distribución normal bivariada con covarianza proporcional a la matriz identidad (es decir, ruido independiente para cada neurona con desvío estándar **σ**). Con numpy, pueden muestrear usando ***h = np.random.normal(scale=0.01, size=(n_timesteps, 2))***. Utilice ***a=10***, ***dt=0.01*** y empiece con las mismas condiciones iniciales que el Ejercicio 1c. Haga los mismos gráficos que en el Ejercicio 1d y 1e. (***10 puntos***)

---

[1] Recordatorio para los Bion't: las neuronas (en general) se pueden clasificar como excitatorias o inhibitorias si su salida incrementa o decrementa la salida de las neuronas con las que hace sinapsis.
[2] Recuerden que $\sin(X) = (e^{i}-e^{-i})/2i$, $\cos(x)=(e^{i}+e^{-i})/2$ e $-i=i^{-1}$

c. Corra esta simulación *K=50* veces, ploteando los puntos como puntos (es decir, como gráfico de dispersión en vez de traza) y coloreando todos los puntos correspondientes a cierto tiempo del mismo color (es decir, 50 puntos de un color para t=0, 50 puntos de otro color para t=1, etc.). ¿Qué observa en la varianza de $\boldsymbol{r_t}$ en el tiempo? (***5 puntos***)

Ejercicio 3:

a. Considere una red con dinámica "Winner-Take-All" definida por las ecuaciones diferenciales

$$\tau E_1' = -E_1 + S(K_1 - 5E_2 - 5E_3)$$
$$\tau E_2' = -E_2 + S(K_2 - 5E_1 - 5E_3)$$
$$\tau E_3' = -E_3 + S(K_3 - 5E_1 - 5E_2)$$

Con $\boldsymbol{\tau=10}$, $\boldsymbol{K_1=K_2=K_3=80}$ y la función ***S*** siendo la función Naka-Rushton (un tipo de versión sigmoidea, pueden graficarla).

$$S(x) = \begin{cases} \frac{100x^2}{40^2+x^2} \text{ for } x \geq 0 \\ 0 \text{ for } x < 0 \end{cases}$$

Esencialmente es un sistema donde cada neurona (con actividad $\boldsymbol{E_i}$) inhibe a las otras con peso sináptico de -5. Calcule los puntos fijos. Son cuatro puntos: tres simétricos (donde $\boldsymbol{E_i=a}$ y $\boldsymbol{E_{!=1}=0}$, es decir, 2 de las 3 neuronas valen 0) y otro donde las tres neuronas valen lo mismo ($\boldsymbol{E_1= E_2= E_3}$). En cualquiera de los casos las ecuaciones se simplifican. Para el caso simétrico, puede ser conveniente usar un solver numérico (*from scipy.optimize import fsolve*). (***10 puntos***)

b. Simule trayectorias del sistema empezando de distintos puntos aleatorios (recuerden que $\boldsymbol{E_i>0}$) y coloreando por el atractor (asimétrico) hacia el que converge. ¿Cómo depende el resultado al que converge de la inicialización? Puede usar un integrador de Euler, Runge-Kutta 4, o el que lo haga feliz. (***10 puntos***)

c. ¿Qué pasa si inicializan el sistema *justo* sobre puntos simétricos (distintos al fijo)? ($\boldsymbol{E_1= E_2= E_3 \neq E^*}$) (grafique no solo en 3D si no $\boldsymbol{E_1}$ vs tiempo) (***5 puntos***)

d. Linealice el sistema en el punto fijo simétrico y en alguno de los puntos fijos asimétricos. Linealizar implica calcular el Jacobiano (la matriz de cada derivada parcial respecto a cada variable[3]), ver su valor numérico *en* ***E=E*** (con ***E**** un punto fijo), y luego analizar sus autovalores. ¿Concuerdan los autovalores con lo observado en las simulaciones en ***b*** y ***c***? ¿Qué tipo de dinámica presentan los puntos fijos?(***15 puntos***)

e. Ahora que tiene el Jacobiano, simular el sistema con una aproximación lineal a primer orden ($\boldsymbol{E' \approx J(E^*)(E-E^*)}$). Inicializando de estados cercanos a los puntos fijos, corra

[3] Va a ser un poquito engorroso, no se gasten por simplificar un montón, mientras que puedan después calcular ***J(E*)*** y les quede la matriz de 3x3 para analizar.

simulaciones usando el sistema linealizado y el sistema no-lineal, viendo cómo difieren las trayectorias. Haga esto partiendo desde el punto fijo simétrico y alguno de los puntos fijos asimétricos. (***10 puntos***)