Práctica 3 - Estadística de trenes de spikes

Zablotsky, Amir Nicolás

Redes Neuronales 2022

Instituto Balseiro, UNCuyo

EJERCICIO 1

Se analizó, mediante un tratamiento estadístico, la actividad de una neurona perteneciente a un saltamontes al estimularlo con una onda sonora. Estos datos experimentales fueron obtenidos por Ariel Rokem.

Se repitió 128 veces el mismo estímulo al saltamontes, de un segundo de duración, registrando en cada realización la actividad de la neurona. Se discretizaron las mediciones en intervalos de 0,1 ms, y en cada uno de estos se indica si se registró un spike o no. En la Fig. 1 se ilustran la envolvente de la onda sonora emitida al grillo, y debajo de esta se observan los trenes de spikes para cada una de las 128 realizaciones.

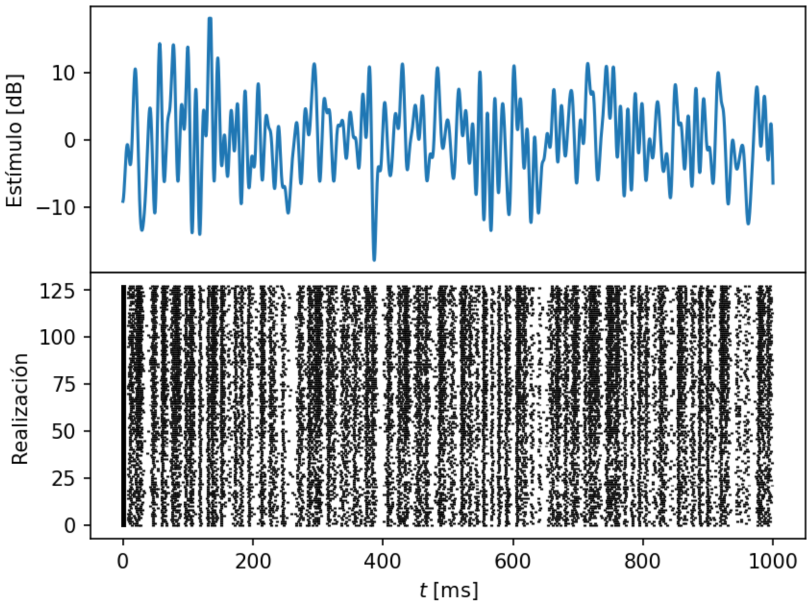

Figura 1: (Arriba) Envolvente del estímulo sonoro presentado al grillo. (Abajo) Representación de los trenes de spikes registrados en las 128 realizaciones del experimento.

En primer lugar se estimó la distribución de probabilidad de intervalos entre spikes $P(\text{ISI})$ a partir de todas las realizaciones. En la Fig. 2 se observa el histograma normalizado del intervalos entre spikes consecutivos, registrados en todas las realizaciones. Este aproxima la distribución de probabilidad $P(\text{ISI})$, a partir de la cual obtenemos un intervalo inter-spike medio de $\langle \text{ISI} \rangle = 8{,}57\,\text{ms}$ con desviación estandar $\sigma_{\text{ISI}} = 5{,}63\,\text{ms}$, y a partir de estos valores obtenemos el coeficiente de variabilidad $CV = 0{,}66$. Esto refleja el hecho de que la emisión de spikes por parte de esta neurona no se comporta como un proceso de Poisson, ya que en este caso el coeficiente de variabilidad daría 1.

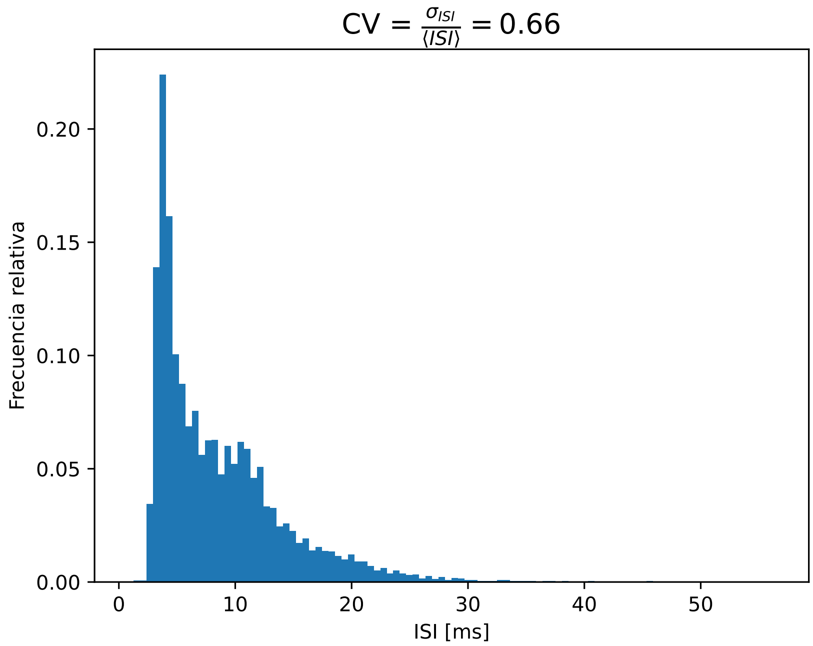

Figura 2: Aproximación de la distribución $P(\mathrm{ISI})$ de intervalor inter-spike, obtenida a partir de las 128 realizaciones.

Luego se realizó un histograma de la cantidad de spikes N ocurridos por realización, el cual es observa en la Fig. 3. A partir de esta distribución, se obtuvo la cantidad de spikes media $\langle N \rangle = 117{,}01$ con varianza $\sigma_N^2 = 183{,}20$, y a partir de estas cantidades se obtiene el factor de Fano $F = 1{,}57$. Esto concuerda con el coeficiente de variabilidad en el hecho de que el proceso no es de Poisson, ya que en ese caso tanto CV como F serían iguales a 1. En caso de tratarse de un proceso de renewal, el factor de Fano tiende a CV^2, sin embargo en este caso vemos que F es mucho mayor a $CV^2 = 0{,}44$, lo cual implica que este proceso tampoco es de renewal (al menos en el intervalo de tiempo medido).

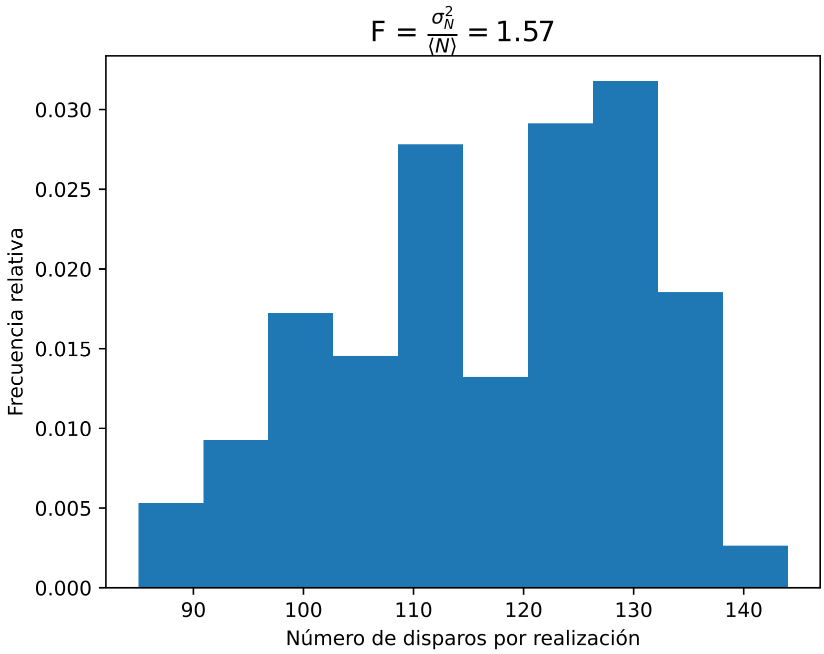

Figura 3: Aproximación de la distribución $P(N)$ de cantidad de spikes en el tren, obtenida a partir de las 128 realizaciones.

Lo siguiente que se hizo fue un histograma de la tasa de disparo en función del tiempo ($r(t)$) a partir del promedio (sobre realizaciones) de spikes en cada intervalo de 0,1 ms, es decir la resolución de las mediciones, el cual puede observarse en la Fig. 4. A partir del histograma de $r(t)$ se obtuvo una tasa de disparo media de $\langle r(t) \rangle = 116{,}02\,\mathrm{Hz}$.

Finalmente se calculó el filtro asociado a la neurona, con el fin de dar la mejor predicción posible de la respuesta $r(t)$. Partiendo de que la respuesta de la neurona a tiempo t corresponde a un funcional del estímulo a todo tiempo $t' < t$, es decir

$$r(t) = \mathcal{F}\left(\{s\,(t')\}\right), \quad t' < t\,,$$

realizamos una expansión en serie de Volterra a primer orden en el lado derecho, obteniendo así una expresión para nuestra predicción dada por la Ec. 1.

$$r_{\mathrm{est}}(t) = r_0 + \int_0^\infty D\,(\tau)\,S\,(t-\tau)\,d\tau \tag{1}$$

Si despreciamos el tiempo de autocorrelación del estímulo, asumimos que $\langle S(t) \rangle = 0$ y que es estacionario, podemos obtener el filtro lineal a partir de la Ec. 2, donde σ_S^2 corresponde a la varianza del estímulo, N es la cantidad de spikes en el tren, y el promedio es realizado sobre realizaciones.

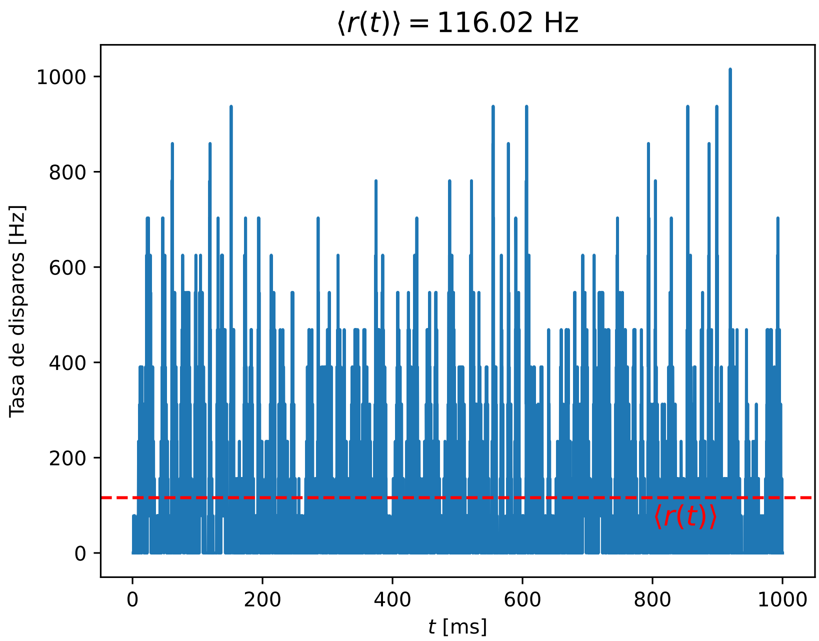

Figura 4: Tasa de disparo en función del tiempo, computada a partir de las 128 realizaciones.

$$D\,(\tau) = \frac{1}{\sigma_S^2}\left\langle \frac{1}{N}\sum_{\{t_{\mathrm{spike}}\}} S\left(t_{\mathrm{spike}} - \tau\right)\right\rangle \tag{2}$$

En la Fig. 5 se observa el filtro lineal $D(\tau)$ en función de τ, y en particular vemos un máximo en 6 ms. Considerando que $D(\tau)$ es proporcional a la correlación entre el estímulo y la respuesta, podemos identificar este pico como el tiempo aproximado de respuesta de la neurona frente al estímulo.

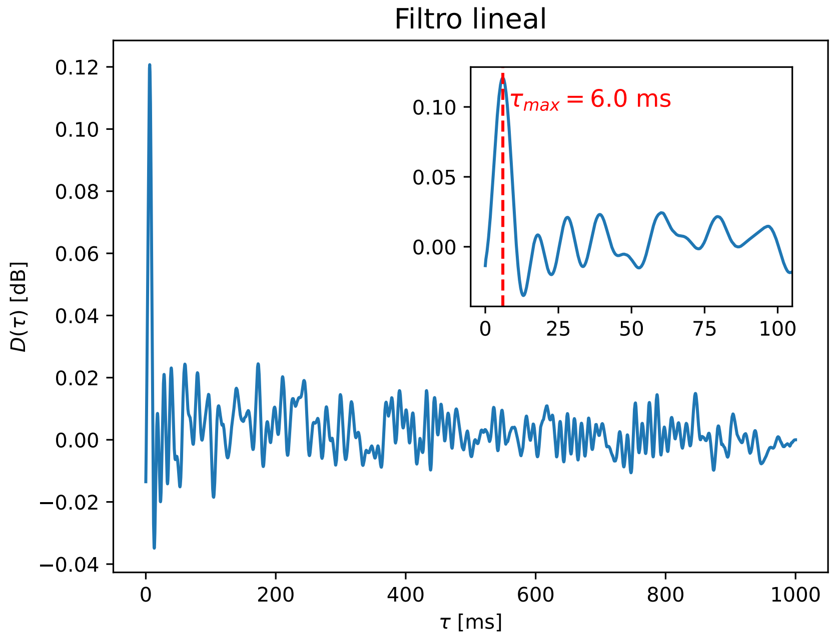

Figura 5: Filtro lineal asociado a la neurona, en función de τ. Se observa un máximo a los 6 ms.

Por último, se empleó el filtro $D(\tau)$ obtenido para calcular r_{est} a partir de la Ec. 1, donde r_0 corresponde a $\langle r(t) \rangle$. En la Fig. 6 se encuentra graficada la predicción de la respuesta de la neurona, junto al histograma de $r(t)$ suavizado, y se puede ver como la predicción es acertada, incluso tratándose de una aproximación a primer orden.

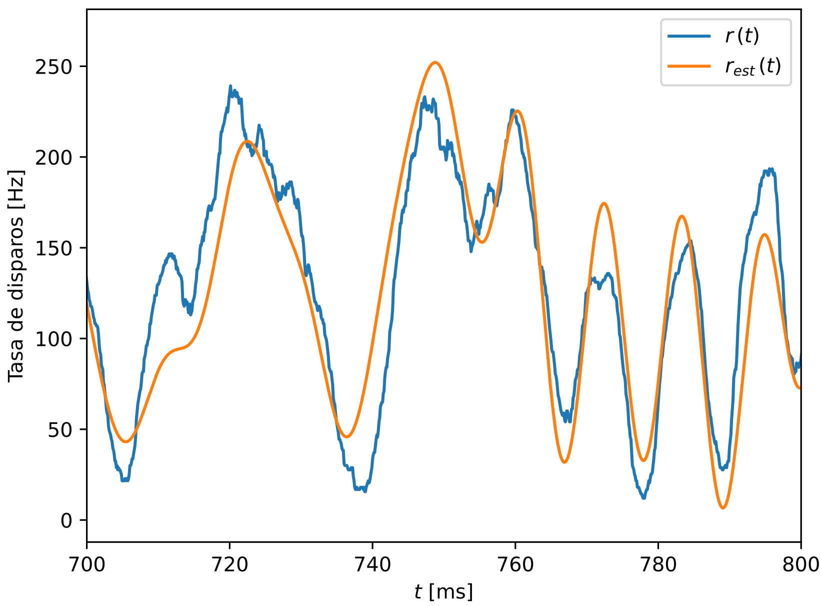

Figura 6: Predicción de la respuesta de la neurona frente al estímulo, en función del tiempo. Se observa como predice de manera correcta la respuesta real.

APÉNDICE

Código Ejercicio 1

```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp

stimulus = np.loadtxt('stimulus.dat')
spikes = np.loadtxt('spikes.dat')

##### Estmulo y trenes de spikes #####

fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True, gridspec_kw={'hspace': 0})

ax1.plot(stimulus[:,0], stimulus[:,1])
ax1.set_ylabel('Estmulo [dB]')

data_raster=np.copy(spikes)
for i in range(len(data_raster)):
    for j in range(len(data_raster[i])):
        if data_raster[i,j]==1:
            data_raster[i,j]=j*0.1

ax2.eventplot(data_raster,orientation='horizontal',colors='k',linelengths=1)
ax2.set_xlabel('$t$ [ms]')
ax2.set_ylabel('Realizacin')

plt.show()

##### P(ISI) #####

isi = [] #ms
for i in range(len(spikes)):
    last_s=-1
    for j in range(len(spikes[i])):
        if spikes[i][j] == 1:
            if last_s == -1:
                last_s = j
            else:
                isi.append((j-last_s)*0.1)
                last_s = j

CV=np.std(isi)/np.mean(isi)

plt.hist(isi, bins=100,density=True)
plt.xlabel('ISI [ms]')
plt.ylabel('Frecuencia relativa')
plt.title('CV = $\\frac{\sigma_{ISI}}{\langle ISI\\rangle}= $'+str(round(CV,2)),fontsize=14)

plt.show()

print(np.std(isi))
print(np.mean(isi))


##### P(N) #####
```

```
N=[]
for i in range(len(spikes)):
    N.append(np.sum(spikes[i]))

F=np.var(N)/np.mean(N)

plt.hist(N, bins=10,density=True)
plt.xlabel('Nmero de disparos por realizacin')
plt.ylabel('Frecuencia relativa')
plt.title('F = $\\frac{\sigma_{N}^2}{\langle N\\rangle}= $'+str(round(F,2)),fontsize=14)

plt.show()

print(np.var(N))
print(np.mean(N))

##### r(t) #####

tasa=(np.sum(spikes, axis=0)/128)*10000 #Hz
plt.plot(np.linspace(1, 1000, len(tasa)-1), tasa[1:],label="$r\,(t)$")
plt.plot()
plt.xlabel('$t$ [ms]')
plt.ylabel('Tasa de disparos [Hz]')
plt.axhline(y=np.mean(tasa[1:]), color='r', linestyle='--')
plt.text(800, np.mean(tasa[1:]-60), '$\langle r(t) \\rangle$', fontsize=14, color='r')
plt.title('$\langle r\left( t \\right) \\rangle= $'+str(round(np.mean(tasa[1:]),2))+'
    Hz',fontsize=14)

plt.show()

##### Filtro lineal #####

def D(spikes, stimulus):
    Cs=[]
    for i in range(len(spikes)):
        C_realizacion=np.zeros(10000)

        n=np.sum(spikes[i])
        spike_times=[] # 0.1 ms
        for j in range(len(spikes[i])):
            if spikes[i][j]==1:
                spike_times.append(j)

        for tau in range(0,10000): # 0.1 ms
            C_realizacion[tau]=0
            for t in spike_times:
                if t-tau>=0:
                    C_realizacion[tau]+=stimulus[t-tau]
            C_realizacion[tau]/=n

        Cs.append(C_realizacion)

    C_tau_mean=np.sum(Cs, axis=0)/len(Cs)

    return(C_tau_mean/(np.std(stimulus)**2))
```

```python
D_tau=D(spikes, stimulus[:,1])

fig, ax1=plt.subplots()
plt.plot(np.linspace(0, 1000, len(D_tau)), D_tau)
plt.xlabel('$\\tau$ [ms]')
plt.ylabel('$D(\\tau)$ [dB]')
plt.title('Filtro lineal',fontsize=14)

axins = ax1.inset_axes([0.5, 0.5, 0.45, 0.45])
axins.plot(np.linspace(0, 1000, len(D_tau)), D_tau)
peak=np.argmax(D_tau)
axins.axvline(x=peak*0.1, color='r', linestyle='--')
axins.text(peak*0.1+2, 0.1, '$\\tau_{max}= $'+str(peak*0.1)+' ms', fontsize=12,color='r')
axins.set_xlim(peak-10, peak+10)
axins.set_xlim(-5, 105)

plt.show()


##### r_est (t) #####

def smooth(x, N):
    return np.convolve(x, np.ones((N,))/N, mode='same')

window=65
r_est=np.mean(smooth(tasa[1:], window))+np.convolve(stimulus[:,1], D_tau,
    mode='full')[:10000]
fig, ax1=plt.subplots()
ax1.plot(np.linspace(1, 1000, len(smooth(tasa[1:], window))), smooth(tasa[1:],
    window),label="$r\,(t)$")
ax1.set_xlabel('$t$ [ms]')
ax1.set_ylabel('Tasa de disparos [Hz]')
ax1.plot(np.linspace(1, 1000, len(r_est)), r_est,label="$r_{est}\,(t)$")
plt.legend()
plt.xlim(700,800)

plt.show()
```