تمرین اول

محمد زیاری – ۹۷۲۲۲۰۴۷

در این پروژه به پیاده سازی ۳ مدل نورونی میپردازیم. برای پیاده سازی مدل ها از کلاس استفاده کرده‌ایم. ابتدا مدل LIF را پیاده سازی میکنیم. همانطور که میدانید، معادله‌ای که برای این مدل در نظر گرفته میشود به شرح زیر است:

$$\tau_m \cdot \frac{du}{dt} = -(u - u_{rest}) + R \cdot I(t)$$

که در آن τ_m همان ثابت زمانی است که از حاصل ضرب *R.C* پدید می‌آید. *R* مقاومت مدار و *I* هم جریان ورودی به نورون مورد نظر است. پتانسیل پایدار u_{rest} نیز مقدار اولیه پتانسیل نورون و در حالتی که نورون هیچ کاری انجام نمیدهد را نشان میدهد، همچنین *u* نیز مقدار پتانسیل نورون هست وقتی که شروع به فعالیت میکند.

متغیر هایی را به عنوان ورودی به کلاس میدهیم که بتوانیم به راحتی تغییر دهیم و از آن ها در مدل خود استفاده کنیم. این متغیر ها به شرح زیر هستند:

$Threshold$ همان پتانسیل آستانه ای است که اگر مقدار پتانسیل نورون به آن برسد، *spike* میزدند.

u_{rest} همانطور که در بالا گفته شد پتانسیل اولیه نورون و در حال استراحت آن است.

u_{reset} مقدار پتانسیلی است که نورون پس از *spike* زدن به آن برمیگردد.

R_m , τ_m نیز مقدار مقاومت و ثابت زمانی است که به دلخواه میتوان به مدل داد.

برای استفاده از این معادله بالا کد، نیاز است تا مقدار زمان آن را گسسته سازی کنیم و درواقع این معادله دیفرانسیل را حل کنیم. در ایجا از تعریف مشتق استفاده میکنیم و از رابطه زیر میزان تغییرات *u* را پیدا میکنیم. این رابطه به صورت زیر است:

اگر در نظر بگیریم :

$$\frac{du}{dt} = \frac{u(t + \Delta) - u(t)}{\Delta}$$

آنگاه خواهیم داشت:

$$u(t + \delta) = u(t) - \frac{\delta}{\tau_m} [(u - u_{rest}) - R \cdot I(t)]$$

بدین صورت میتوانیم از معادله بالا استفاده کنیم ودر هر مرحله *u* را آپدیت کنیم.

البته برای استفاده از این رابطه در کد، در قسمت *R.I* تقسیم بر ۱۰۰۰ کردیم. این تقسیم به این خاطر است که مقدار جریان بسیار زیاد است و اگر این تقسیم را انحام ندهیم، مدل ما در هر لحظه *spike*

میزند و باعث میشود که هر لحظه به پتانسیل *rest* برگردد و در نتیجه درست عمل نمیکند و همواره بر روی مقدار *-65* باقی می‌ماند.

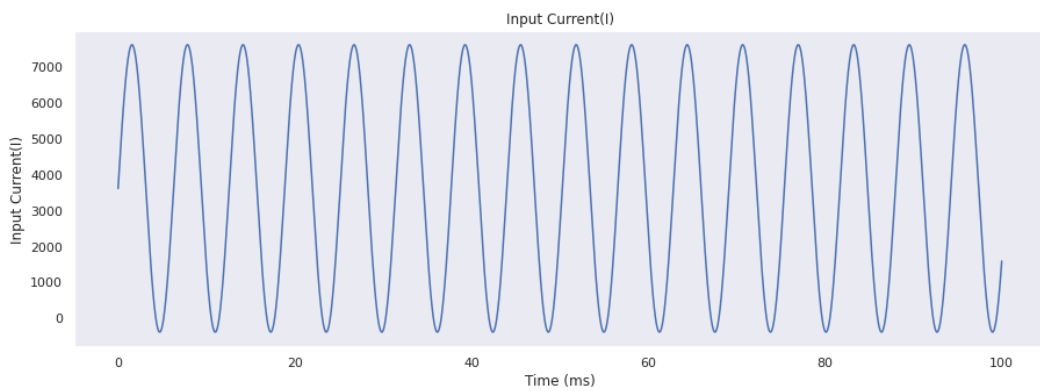
δ همان *time resolution* یا گام زمانی بسیار کوچک است که آن را مقدار ثابتی در ابتدا در نظر گرفته ایم. همچنین کل زمان را نیز *100* واحد در نظر میگیریم.

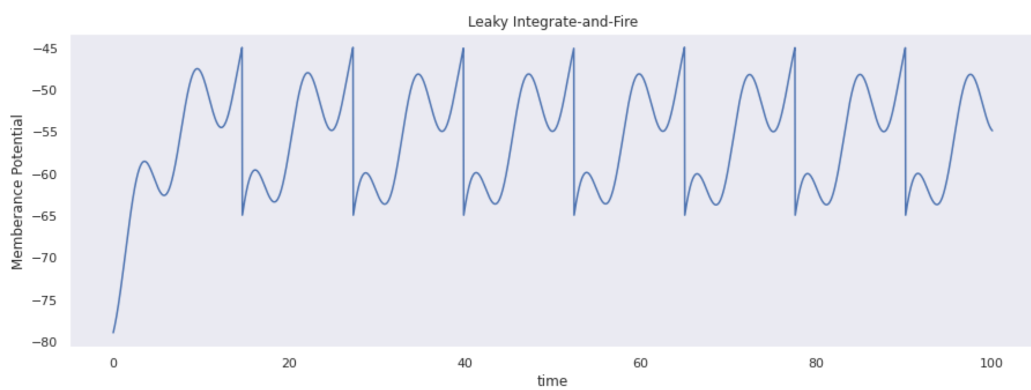
برای پیدا سازی مدل، همانطور که گفته شد معادله مورد نظر را نوشته ودر هر مرحله مقدار *u* بدست آمده را بررسی میکنیم. اگر این مقدار بیشتر از *threshold* باشد، درواقع در اینجا لازم است که *u* را به مقدار u_{reset} برگردانیم و همچنین مقدار پتانسیلی را که نورون در آن *spike* زده است و همچنین زمان این اتفاق را ذخیره میکنیم تا بعدا برای رسم نمودار استفاده کنیم.

پس از اتمام این شبیه سازی، کل زمان صرف شده یعنی *t_time* و همچنین مقادیر پتانسیل ها یعنی *u_list* را در یک دیتا فریم ذخیره میکنیم و با استفاده از آن به رسم نمودار ها میپردازیم.

برای نمودار ها نیاز است تا تابع جریان های متفاوتی را به مدل بدهیم و بررسی کنیم تا عملکرد مدل را ببینیم:

ابتدا همان نموداری که به عنوان مثال در سوال آمده را در نظر گرفتیم:





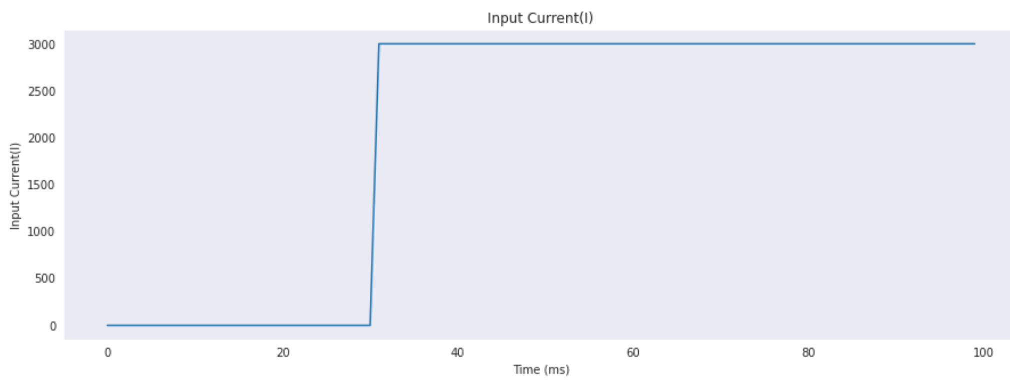
مقادیر داده شده:

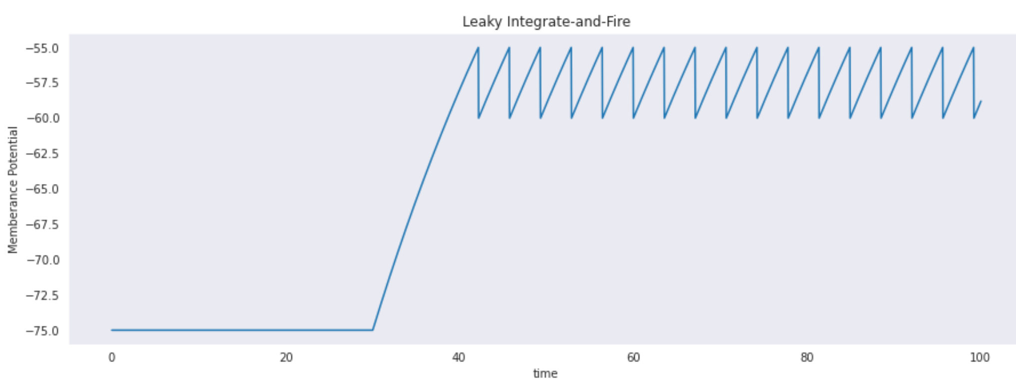
$u_{threshold} = -54, u_{rest} = -79, u_{reset} = -65, R_m = 10, \tau_m = 8,$

$Time = 100 , delta = 0.03125$

$I(t) = 4000 * (\sin(t) + 0.9)$

برای نمودار بعدی، تابع ثابت $t = 3000$ را در نظر گرفته‌ایم اما برای $t>30$. یعنی تا $t = 30$ هیچ عملکردی نداریم و همانطور که میبینید پتانسیل ثابت است، اما از آن به بعد تغییر میکند.





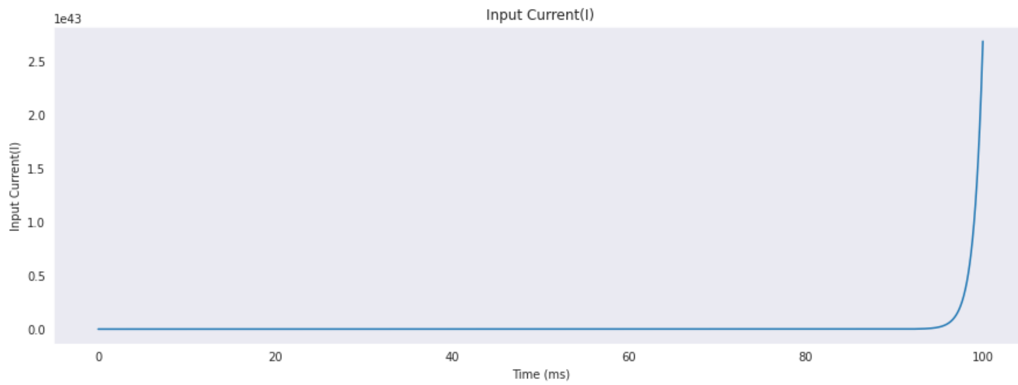
مقادیر داده شده:

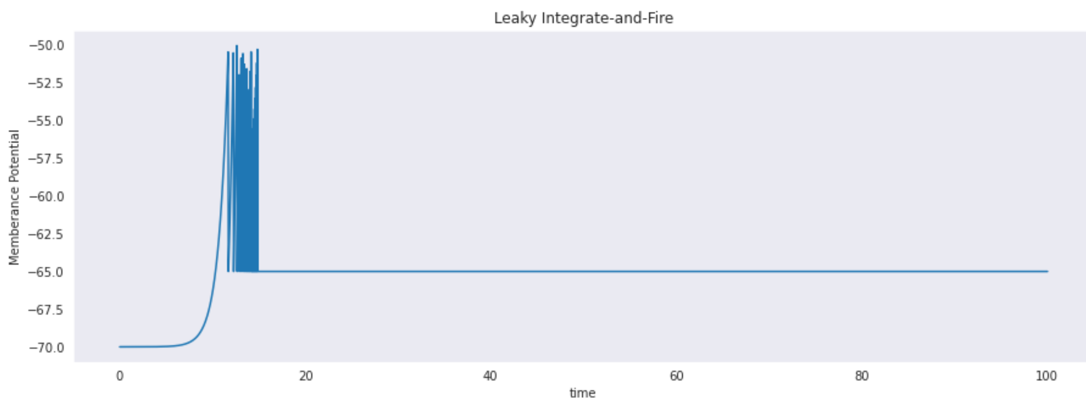
$u_{threshold} = -55, u_{rest} = -75, u_{reset} = -60, R_m = 20, \tau_m = 30,$

$Time = 100 , delta = 0.03125$

$I(t) = 3000 \quad t > 3$

نمودار بعدی تابع نمایی است:





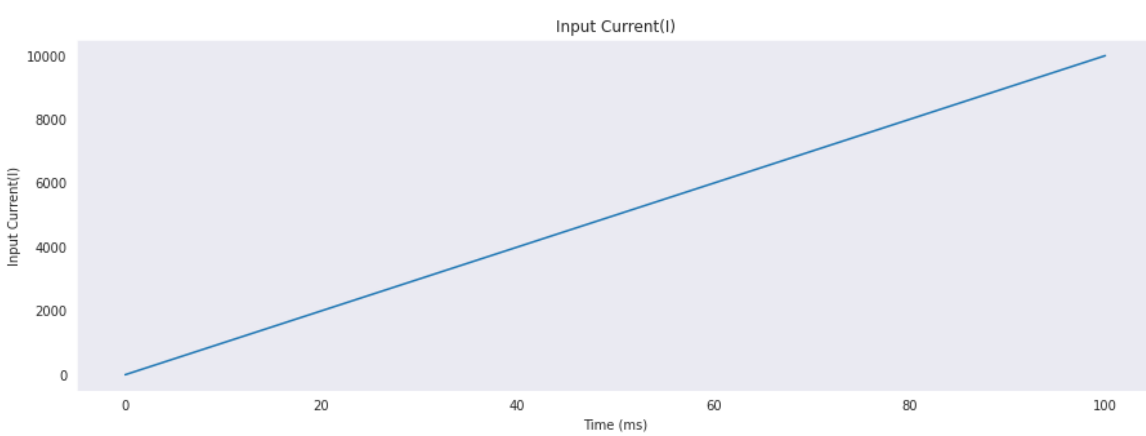
مقادیر داده شده:

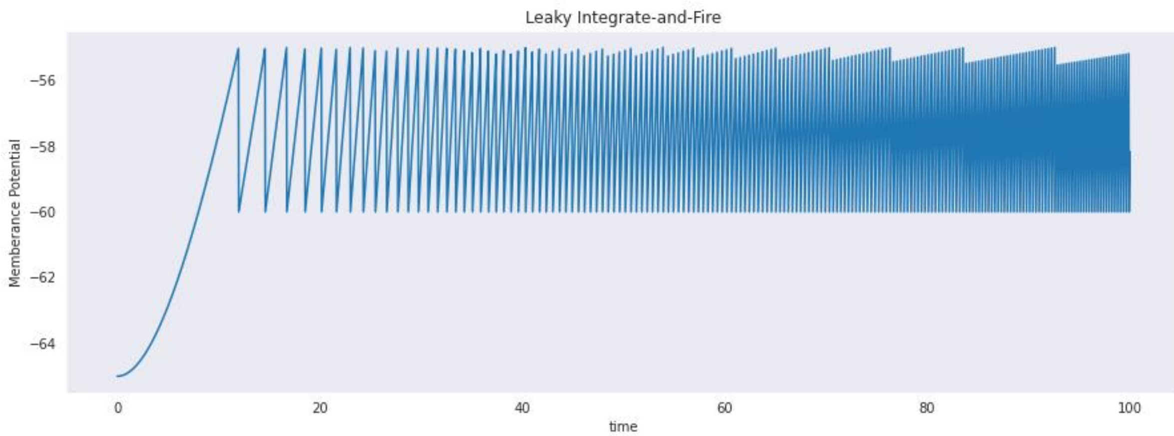
$$u_{threshold} = -50, u_{rest} = -70, u_{reset} = -65, R_m = 4, \tau_m = 23,$$

$$Time = 100, delta = 0.03125$$

$$I(t) = e^t$$

نمودار بعدی مربوط به تابع خطی است





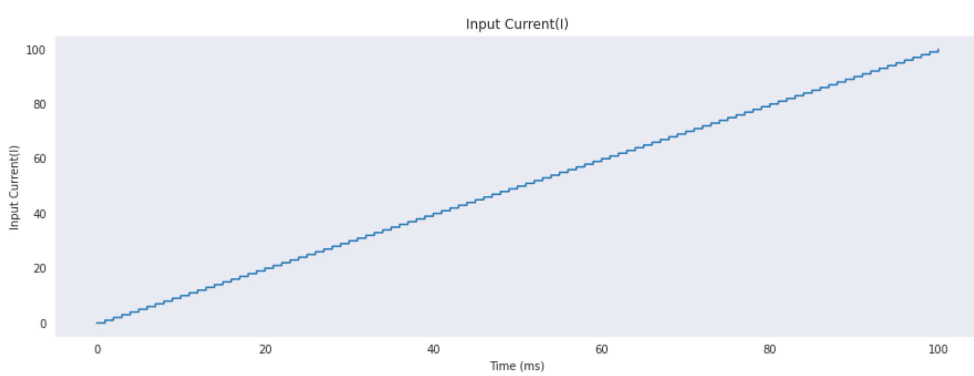
مقادیر داده شده:

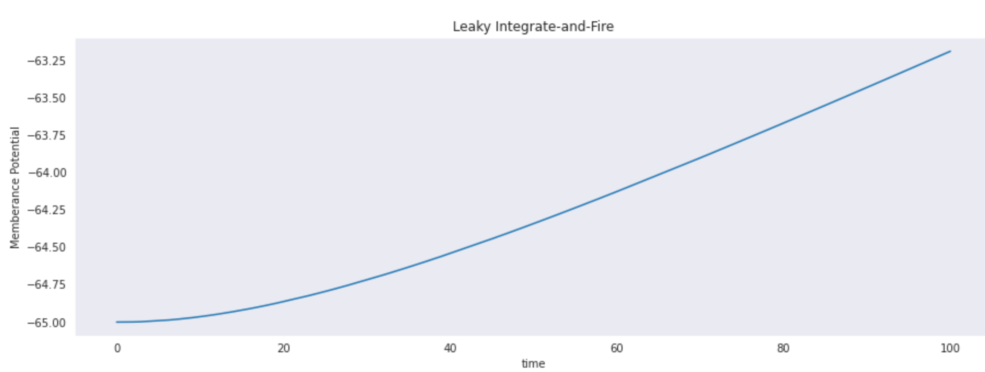
$u_{threshold} = -55, u_{rest} = -65, u_{reset} = -60, R_m = 20, \tau_m = 10,$

$Time = 100 , delta = 0.03125$

$I(t) = 100 \cdot t$

معادله بعدی جریان پلکانی است، تابع براکت x را داده ایم.





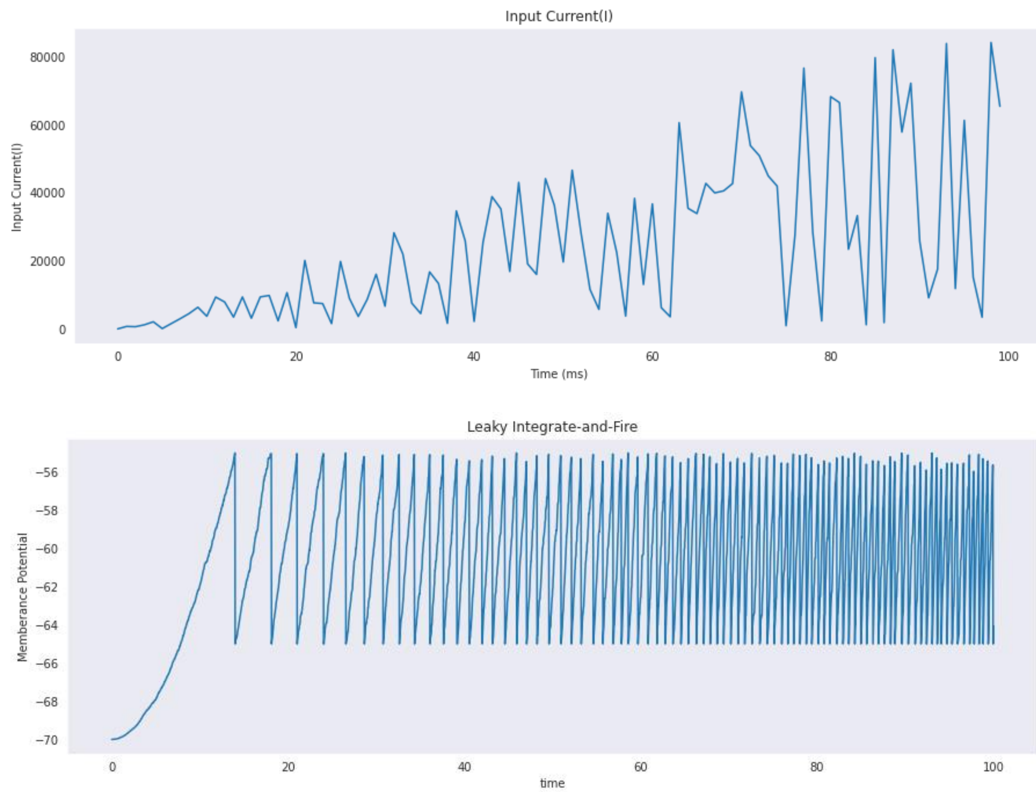
مقادیر داده شده:

$u_{threshold} = -50, u_{rest} = -65, u_{reset} = -65, R_m = 25, \tau_m = 28,$

$Time = 100, delta = 0.03125$

$I(t) = \lfloor t \rfloor$

برای نمودار آخر هم از اعداد رندم استفاده کرده ایم.



مقادیر داده شده:

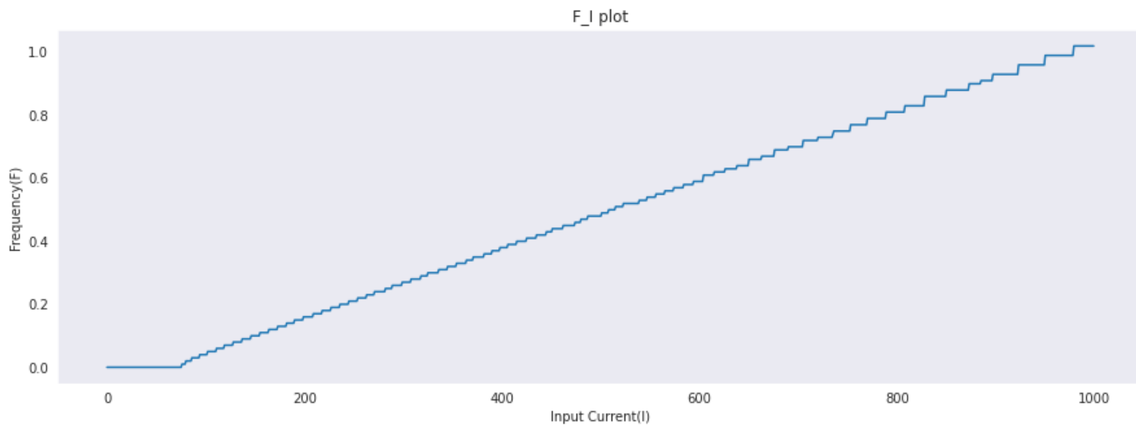
$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 10, \tau_m = 28,$

$Time = 100, delta = 0.03125$

$I(t) = random \cdot 1000 \cdot t$

در انتها هم برای کشیدن نمودار F-I به اینگونه عمل میکنیم که تعدادی جریان های مختلف(در اینجا ۱۰۰۰تا) را در نظر میگیریم و تابع جریان های خطی i * 10 را برای ۱۰۰۰ شبیه سازی تعریف میکنیم. برای هر جریان مدل را پیاده کرده و تعداد زمان هایی که مدل fire کرده است را بر کل زمان تقسیم میکنیم، این همان فرکانس بدست آمده است. در ابتدا که مدل هنوز spike نزده، میزان فرکانس برابر با 0 است.

سپس نمودار F-I را رسم میکنیم:



در بخش بعدی، به بررسی مدل exponential leaky integrate-and-fire میپردازیم.

این مدل بسیار مشابه مدل قبل است، اما همانطور که میدانیم، غیرخطی بودن آن باعث میشود تا عملکرد مدل بسیار نزدیک تر به نورون واقعی باشد، درواقع وقتی به threshold میرسیم، مدل fire میکند و با سرعت بیشتری به threshold میرسد و باعث میشود که نورون با سرعت بیشتری بالا برود و با استفاده از یک متغیر جدید Δ_T باعث میشود تا میزان این سرعت تنظیم شود.

پیاده سازی مدل کاملا مشابه مدل قبلی است، با این تفاوت که یک جملهی نمایی به رابطه قبل اضافه میشود و دو متغیر جدید Δ_T , θ_{rh} را داریم.

در اینجا θ_{rh} نقش firing threshold را دارد و همانطور که گفتیم Δ_T میزان سرعت بالارفتن نورون در لحظه fire شدن را نشان میدهد که به آن sharpness parameter هم گفته میشود.

معادلهای که برای این مدل استفاده میشود به شرح زیر است:

$$\tau \cdot \frac{du}{dt} = -(u - u_{rest}) + \Delta_T \exp\left(\frac{u - \theta_{rh}}{\Delta_T}\right) + R \cdot I(t)$$

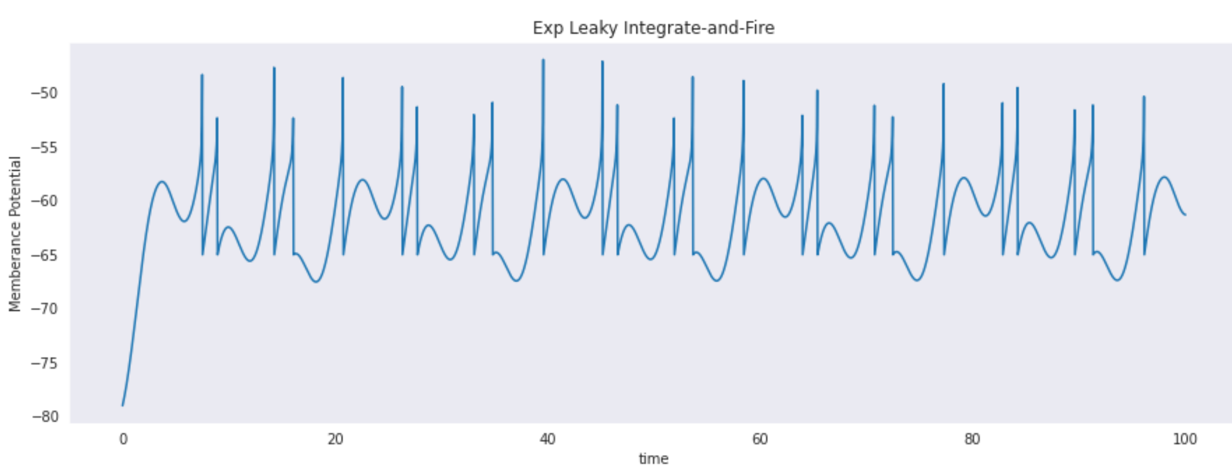
همانند معادله قبل با جایگزاری مشتق، به عبارت زیر میرسیم:

$$u(t + \delta) = u(t) + \frac{\delta}{\tau_m}\left[-(u - u_{rest}) + \Delta_T \exp\left(\frac{u - \theta_{rh}}{\Delta_T}\right) + R \cdot I(t)\right]$$

ازین عبارت در کد خود استفاده کردهایم. ادامه توضیحات مشابه مدل قبلی است،

حال به بررسی نمودار ها میپردازیم:

نمودار اول:



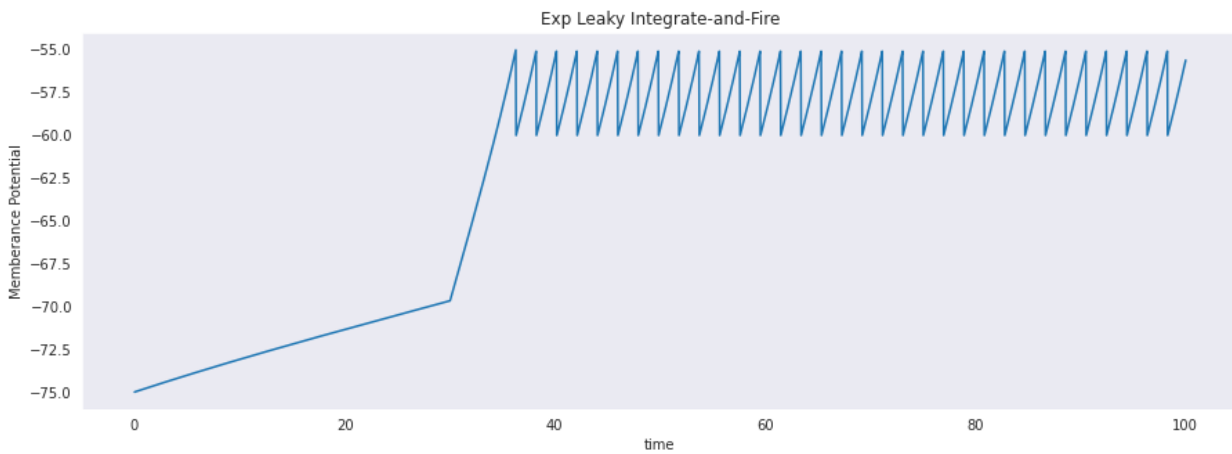
مقادیر داده شده:

$u_{threshold} = -45, u_{rest} = -79, u_{reset} = -65, R_m = 10, \tau_m = 8, \Delta_T = 1,$

$\theta = -60, Time = 100, delta(time) = 0.03125$

$I(t) = 4000 \cdot (\sin(t) + 0.9)$

نمودار دوم:



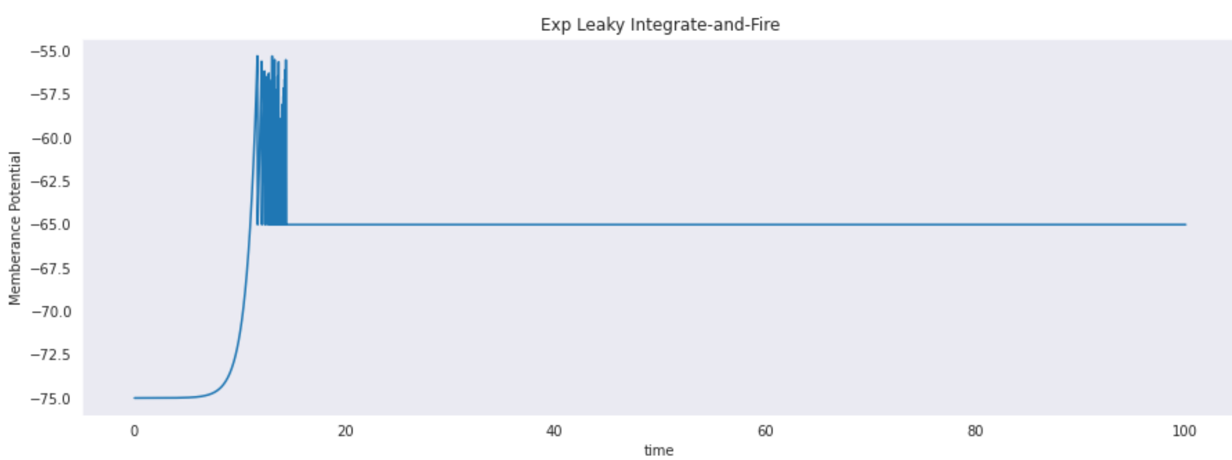
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -75, u_{reset} = -60, R_m = 20, \tau_m = 30, \Delta_T = 10,$

$\theta = -70, Time = 100, delta(time) = 0.03125$

$I(t) = 3000 \quad t > 30$

نمودار سوم:



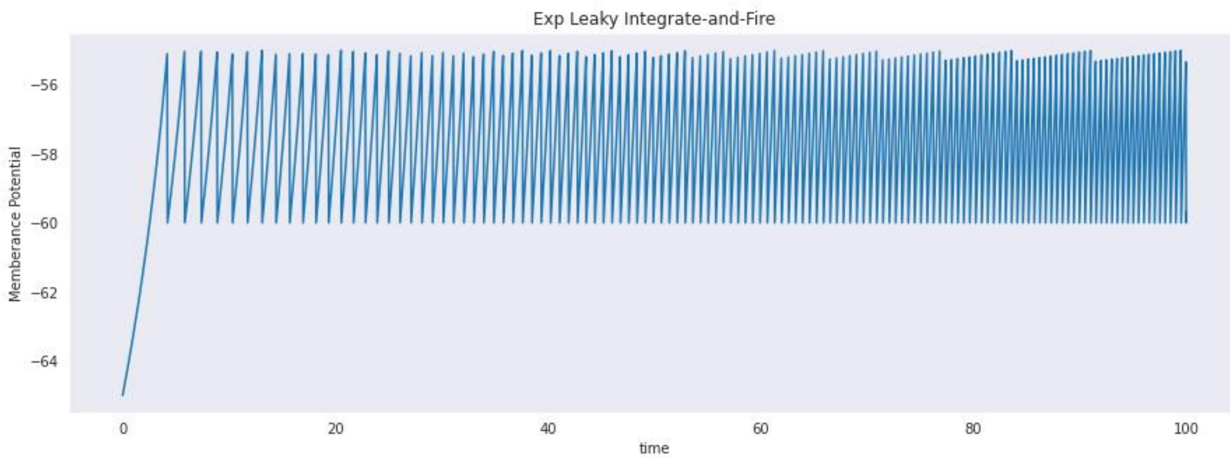
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -75, u_{reset} = -65, R_m = 4, \tau_m = 23, \Delta_T = 1,$

$\theta = -60, Time = 100, delta(time) = 0.03125$

$I(t) = e^t$

نمودار چهارم:



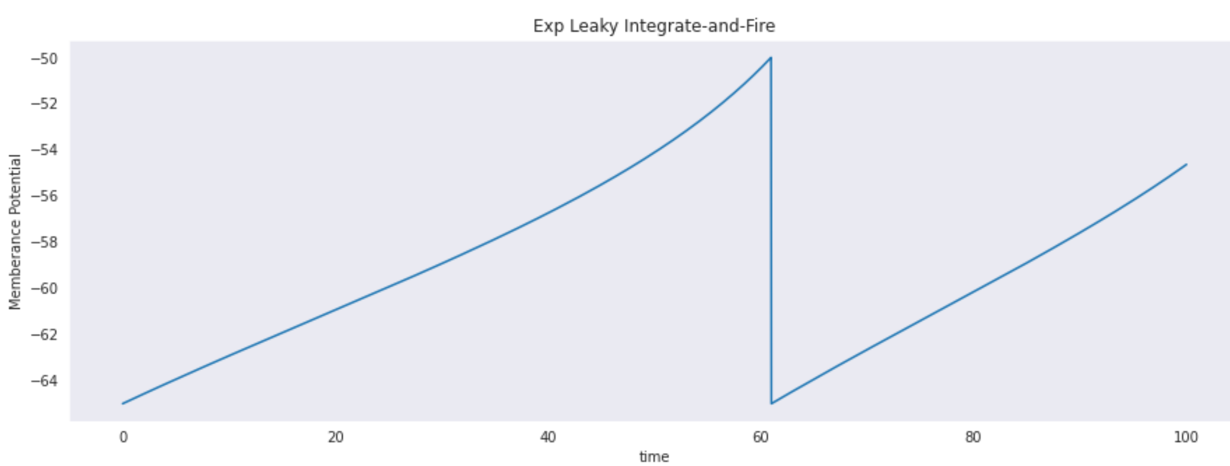
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -65, u_{reset} = -60, R_m = 8, \tau_m = 10, \Delta_T = 10,$

$\theta = -70, Time = 100, delta(time) = 0.03125$

$I(t) = 100 \cdot t$

نمودار پنجم:



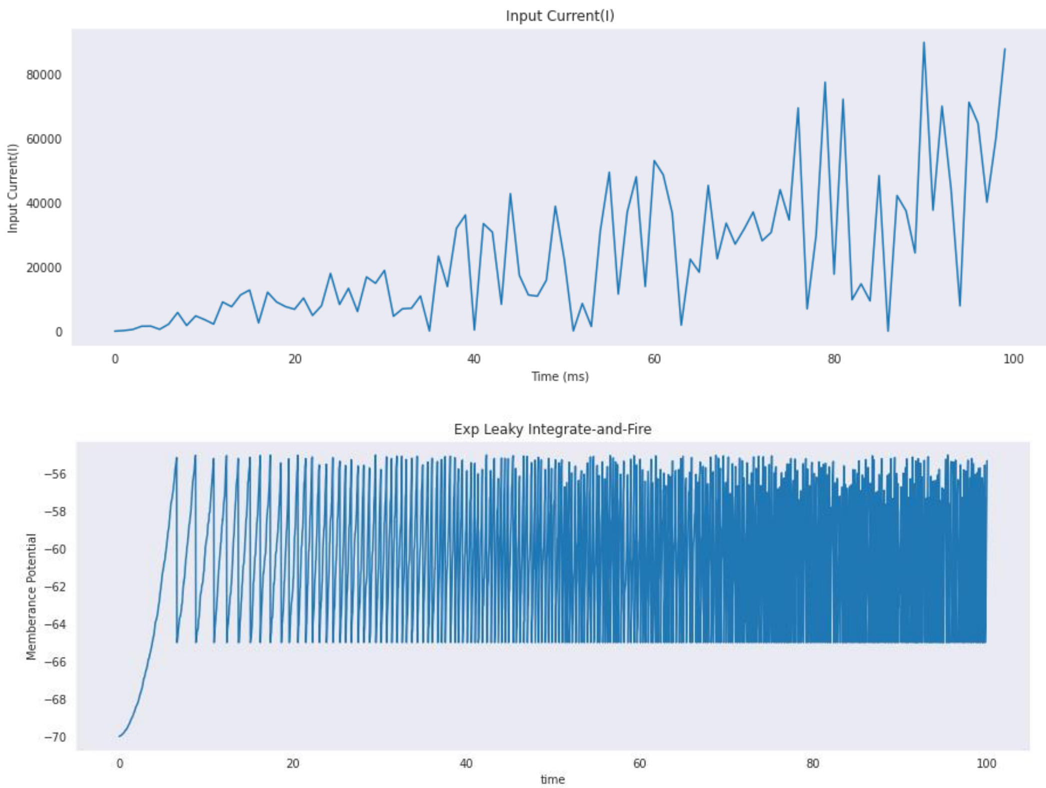
مقادیر داده شده:

$u_{threshold} = -50, u_{rest} = -65, u_{reset} = -65, R_m = 25, \tau_m = 28, \Delta_T = 10,$

$\theta = -60, Time = 100, delta(time) = 0.03125$

$I(t) = \lfloor t \rfloor$

نمودار آخر:



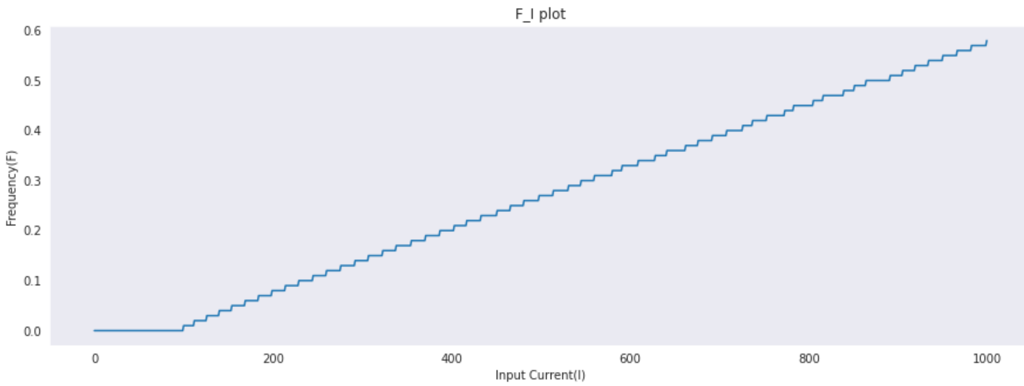
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 20, \tau_m = 18, \Delta_T = 10,$

$\theta = -60, Time = 100, delta(time) = 0.03125$

$I(t) = random \cdot t \cdot 1000$

برای رسم نمودار *F-I* نیز مانند مدل قبلی عمل میکنیم:



مقادیر داده شده:

$u_{threshold} = -45, u_{rest} = -79, u_{reset} = -65, R_m = 10, \tau_m = 8, \Delta_T = 10,$

$\theta = -60, Time = 100, delta(time) = 0.03125$

$I(t) = random \cdot t \cdot 1000$

در بخش بعدی به بررسی مدل *Adaptive exponential leaky integrate-and-fire* می پردازیم.

این مدل فاصله های بین *spike* ها را کنترل میکند، یعنی به محض آنکه یک *spike* زده میشود بلافاصله پس از آن *spike* زده نمیشوند، بلکه یک مقدار این عمل توسط یک متغیر و عبارت جدید کنترل میشود.

متغیر های جدید a, b , τ_w را مشاهده میکنیم. ضرایب *a,b* برای کنترل میزان *spike* زدن ها اضافه شده اند و همچنین τ_w ضریب ثابت جدید است که به واسطه متغیر *w* در فرمول اضافه میشود.

در اینجا متغیر *w* تاثیر منفی در فرمول ما دارد وبا افزایش آن مقدار *u* کم میشود و به *rest* میل میکند

معادله کلی برای این مدل به این صورت است:

$$\tau_m \cdot \frac{du}{dt} = -(u - u_{rest}) + \Delta_T \exp\left(\frac{u - \theta_{rh}}{\Delta_T}\right) - Rw + R.I(t)$$

$$\tau_w \cdot \frac{dw}{dt} = a(u - u_{rest}) - w + b\tau_w \sum_{t^f} \delta(t - t^f)$$

$$if\ firing: u = u_{reset}$$

با توجه به این معادلات و با استفاده از تعریف مشتق، همانند مدل های قبلی عبارت های زیر را برای آپدیت کردن *u* , *w* بدست می‌آوریم:

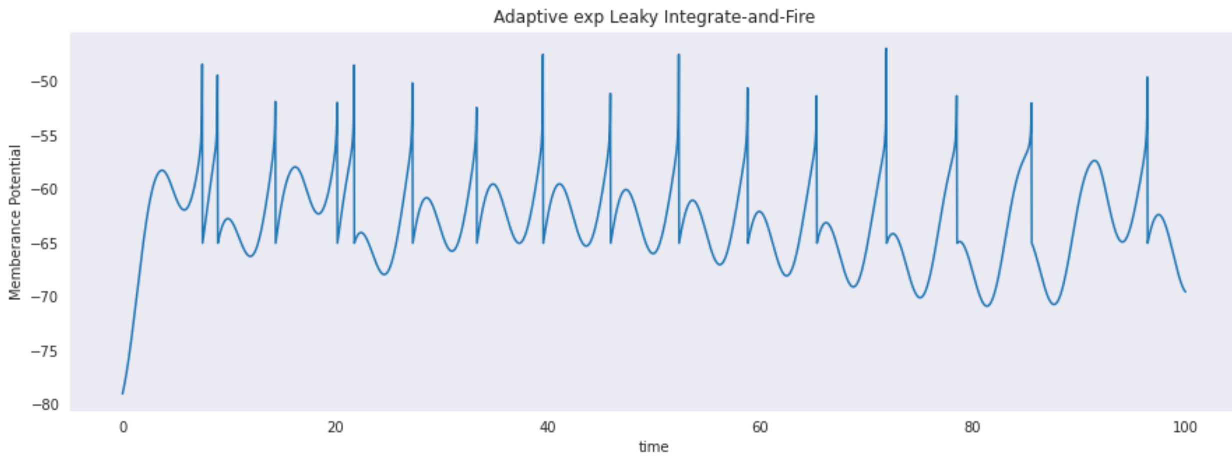
$$w(t + \delta) = w + \frac{\delta}{\tau_k}\left[a(u - u_{rest}) - w + b\tau_w \sum_{t^f} \delta(t - t^f)\right]$$

$$u(t + \delta) = u + \frac{\delta}{\tau_m}\left[-(u - u_{rest}) + \Delta_T \exp\left(\frac{u - \theta_{rh}}{\Delta_T}\right) - Rw + R.I(t)\right]$$

با استفاده از عبارت اول ابتدا *w* را آپدیت میکنیم، سپس مقدار پتانسیل را در هر مرحله بدست می آوریم.

مانند بخش های قبلی نمودار های مختلف را باهم بررسی میکنیم:

نمودار اول:



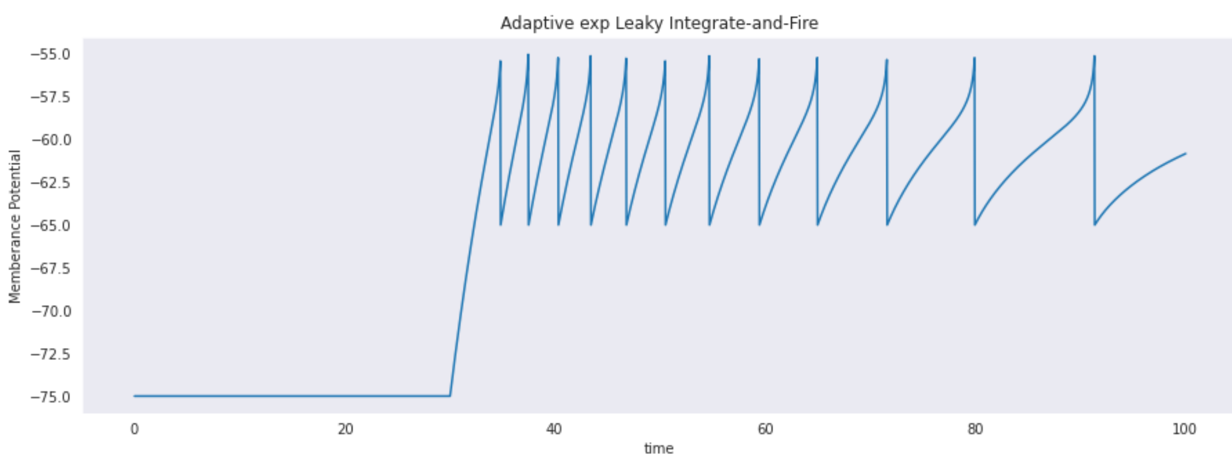
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 20, \tau_m = 18, \Delta_T = 10,$

$\theta = -60, a_k = 0.01, b_k = 100, \tau_w = 1, Time = 100, delta(time) = 0.03125$

$I(t) = 4000 \cdot (\sin(t) + 0.9)$

نمودار دوم:



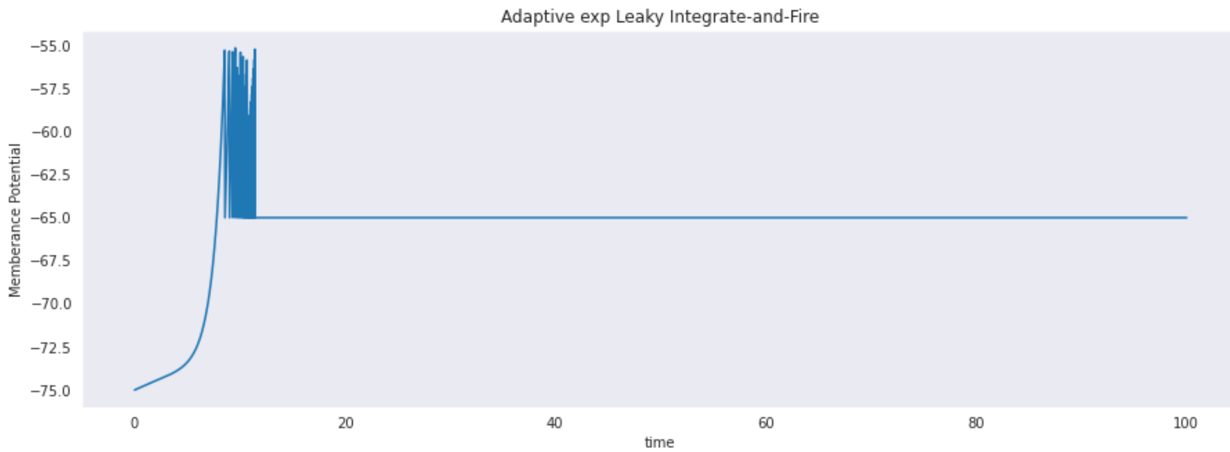
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 10, \tau_m = 8, \Delta_T = 1,$

$\theta = -60, a_k = 0.1, b_k = 200, \tau_w = 1, Time = 100, delta(time) = 0.03125$

$I(t) = 4000 \;\; t > 30$

نمودار سوم:



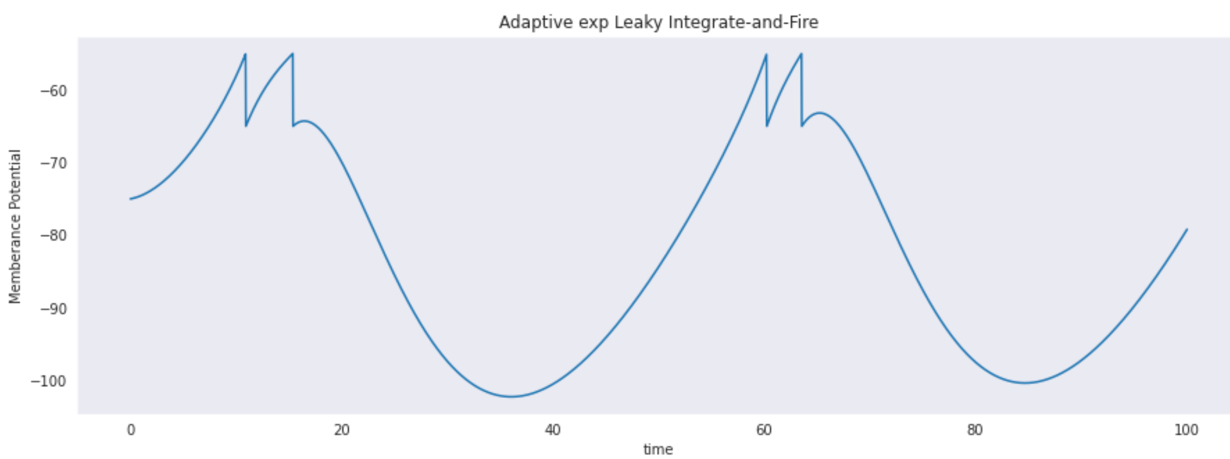
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -75, u_{reset} = -65,\ R_m = 30, \tau_m = 8, \Delta_T = 10,$

$\theta = -60, a_k = 0.01, b_k = 300, \tau_w = 8, Time = 100, delta(time) = 0.03125$

$I(t) = e^t$

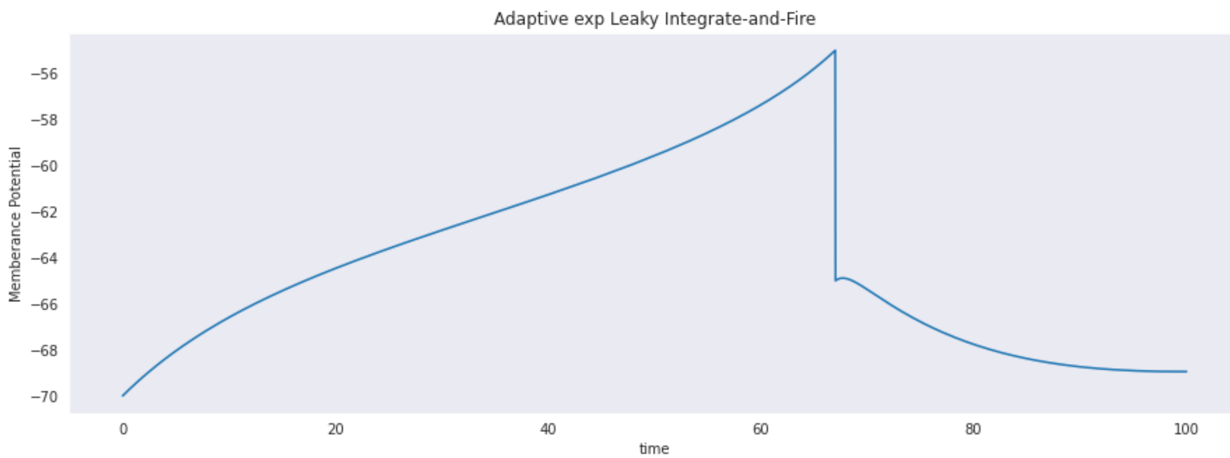
نمودار چهارم:



** مقادیر داده شده همانند نمودار قبلی است.

$I(t) = 100 \cdot t$

نمودار پنجم:



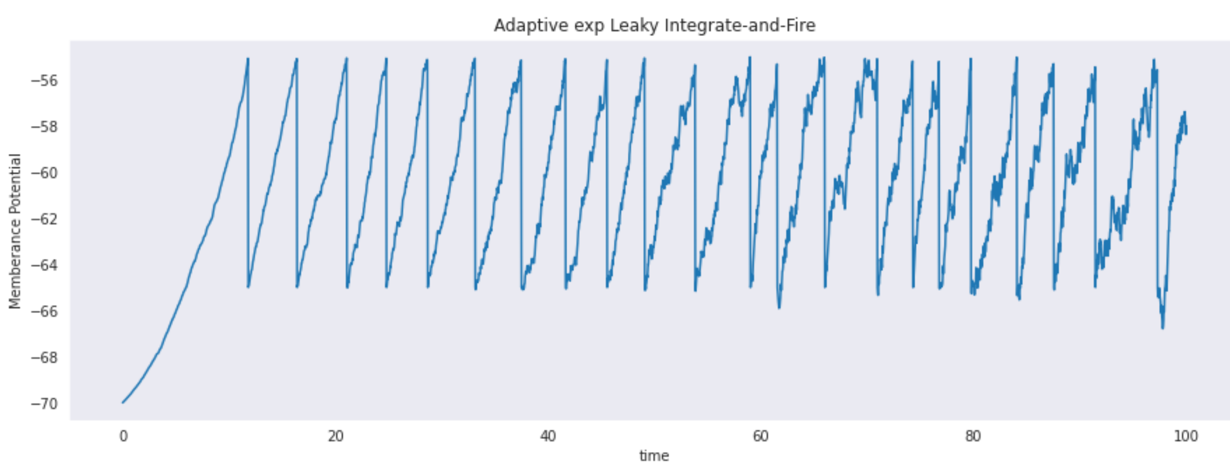
مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 30, \tau_m = 8, \Delta_T = 10,$

$\theta = -60, a_k = 0.01, b_k = 200, \tau_w = 1, Time = 100, delta(time) = 0.03125$

$I(t) = \lfloor t \rfloor$

نمودار آخر:



مقادیر داده شده:

$u_{threshold} = -55, u_{rest} = -70, u_{reset} = -65, R_m = 30, \tau_m = 8, \Delta_T = 10,$

$\theta = -60, a_k = 0.01, b_k = 200, \tau_w = 1, Time = 100, delta(time) = 0.03125$

$I(t) = random \cdot t \cdot 100$

همانطور که میبینید، نمودار های رسم شده در این مدل نسبت به مدل قبل در فاصله *spike* ها تفاوت واضحی دارند، به همین علت هست که این مدل را *adaptive* می نامند و از دو مدل قبلی خیلی بهتر است و به ساختار نورون واقعی نزدیک تر می‌باشد.

در نهایت هم باید نمودار *w-t* را رسم کنیم:

در همان ابتدای کد مقادیر *w* را به یک لیست اضافه کرده بودیم، در نهایت میتوانیم از همان برای *Plot* کردن استفاده کنیم: