



Project 3
Time to First Spike Encoding Report

عرشیا ربیعی
۶۱۰۳۹۸۱۲۵

مرداد ۱۴۰۲

۱ مقدمه

در این گزارش میخواهیم تشریح کنیم که روش کدگذاری TTFS بر روی یک عکس چگونه پیاده سازی شده و نمودار Raster plot آن را نمایش میدهیم که هر یک در چه زمانی اسپایک زده اند.

۲ عملیات بر روی عکس

ابتدا یک عکس rgb که در تصویر پایین مشاهده میکنید را دریافت کردیم. این عکس 128×128 میباشد.



Figure 1: RGB Image

Time to First Spike ۳

تیکه کد هایی که در تصویر پایین آمده است، مسیر تبدیل عکس RGB به عکس سیاه و سفید می باشد



Figure 2: Gray Image

پس از آن مقادیر پیکسل ها که بین ۰ تا ۲۵۵ می باشند را نرمال می کنیم و به بازه ۰ تا ۱ با ۳ رقم اعشار می رسانی. حال میدانیم هر چه مقدار پیکسل بیشتر باشد باید آن پیکسل زودتر اسپایک بزند و بدیهی است که اگر مقدار دو پیکسل برابر باشد در یک زمان اسپایک خواهند زد.

Time to First Spike ۴

برای انجام این کد گذاری یک تنسور ۳ بعدی خواهیم داشت که بعد اول طول زمانی است که تمام پیکسل ها بتوانند در آن بازه اسپایک بزنند. و بعد دوم و سوم به اندازه ی ابعاد تصویر که در مثال ما 128×128 می باشد. محتویات این تنسور به صفر و یک می باشد به اینگونه که اگر در زمان مشخص آن پیکسل معین اسپایک زده باشد ، مقدار متناظر آن یک و در غیر این صورت صفر می باشد.
نحوه ی انجام این کد گذاری را در سلول زیر مشاهده می کنید:

```

In [22]: img_copy = img_normalized.copy()
         ttfs = []
         i = 0
         ttfs.append(np.zeros((img_normalized.shape[0] , img_normalized.shape[1])))
         while img_copy.max() != -INFINITY:
             tmp_max = img_copy.max()
             flat_max_index = img_copy.argmax()
             j = flat_max_index // img_copy.shape[1]
             k = flat_max_index % img_copy.shape[0]
             ttfs[i][j , k] = 1
             img_copy[j,k] = -INFINITY
             if tmp_max != img_copy.max():
                 ttfs.append(np.zeros((img_copy.shape[0] , img_copy.shape[1])))
                 i += 1
         ttfs = np.array(ttfs)
         ttfs

```

```

Out[22]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]],

               [[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]])

```

Figure 3: TTFS Encoding

Raster Plot ۵

در این بخش نشان می‌دهیم که هر پیکسل در چه زمانی اسپایک زده. این نکته هم در نظر بگیریم که شماره ی هر پیکسل در نمودار زیر به صورت flat یک ماتریس 128×128 میباشد.

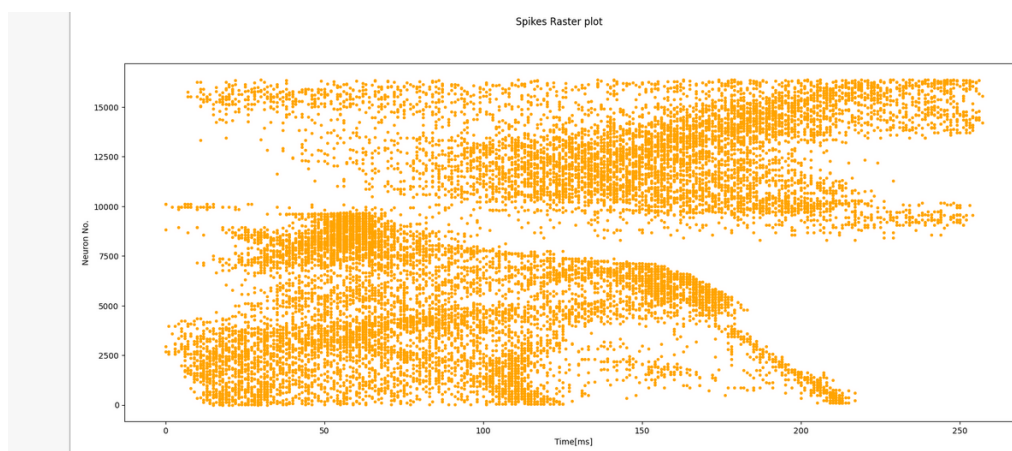


Figure 4: TTFS Raster Plot