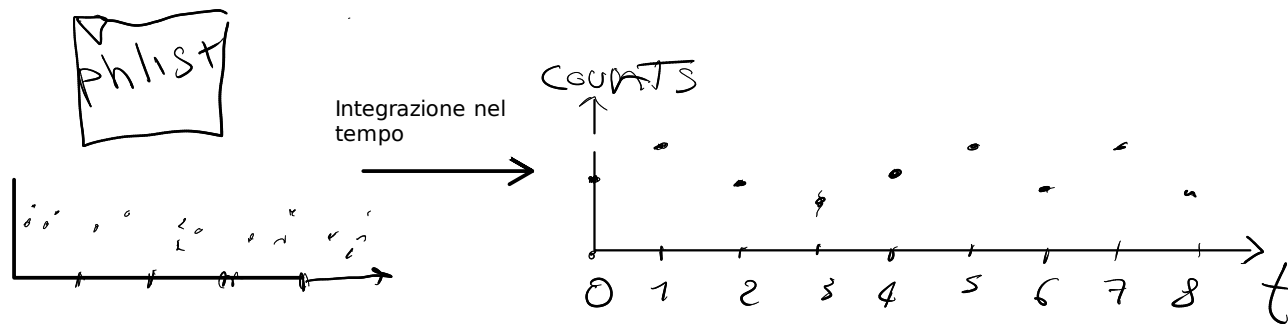


# Rate di input dei dati di training

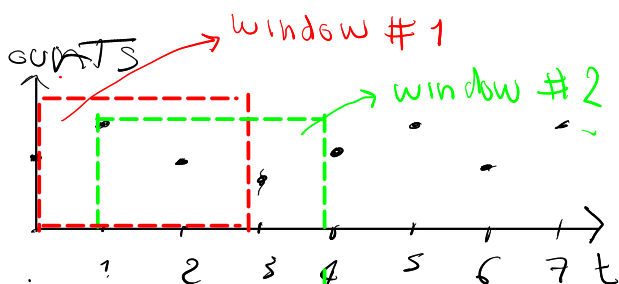
## PARAMETER 1: T



La photon-list viene integrata nel tempo e nell'energia. Attuando un'integrazione temporale, si somma il numero di fotoni in finestre temporali di T secondi, ottenendo così dei "conteggi" i cui valori saranno più o meno grandi a seconda del rate di arrivo dei fotoni, che cambia a seconda di diversi parametri\*.  
=> cosa succede se uso valori di integrazione temporali troppo piccoli? Quanto spesso ottengo valori di conteggi pari a zero? Questo dipende dal rate di input, ovvero il numero di fotoni per secondo: se il telescopio punta una regione con una sorgente che emette, allora il rate sarà alto e il valore di integrazione temporale può scendere, viceversa se la regione è vuota e faccio scendere il valore di integrazione temporale potrei ottenere conteggi pari a zero (specialmente nel caso in cui si effettui anche un'integrazione energetica, infatti in questo caso, i conteggi vengono divisi per bin energetico. Per il momento settiamo T pari a 1 secondo MA:

questo valore di 1 secondo può funzionare in tutte le situazioni?

## PARAMETER 2: WS



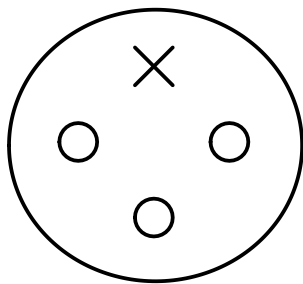
Ogni T secondi si ottiene quindi un nuovo conteggio. L'evoluzione dei conteggi nel tempo va a creare una timeseries che sarà l'input della rete neurale. In questo contesto si deve scegliere la dimensione della timeseries, ovvero il numero di timesteps da cui è composta. Questo valore va a fissare la dimensione del layer di input dell'autoencoder. Come estrarre le timeseries da questo streaming di conteggi? Per esempio con un approccio "sliding window". Supponiamo che la dimensione della timeseries sia pari a WS window size. Posiziono una finestra che va ad estrarre WS conteggi. Per ogni nuovo conteggio aggiunto al plot, faccio scorrere la finestra di 1 unità, estraendo la timeseries successiva.

Più grande è WS più grande è il "tempo di bootstrap di integrazione" ovvero il numero di secondi che devo aspettare prima di avere il primo sample che può essere usato come training. Il tempo di bootstrap per timeseries si definisce:  $TBTS = WS * T$  dove T=tempo di integrazione temporale e WS=dimensione finestra/timeseries.

Una volta che ho superato la fase di bootstrap, si avrà una timeseries ogni T secondi:

$TSR = \text{timeseries rate} = T \text{ secondi}$

## Wobble optimization



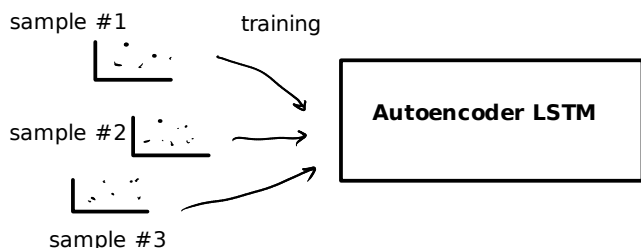
Utilizzando una strategia di osservazione wobble, è possibile aumentare il rate di ottenimento degli esempi con cui poter allenare l'autoencoder.

Estraendo i dati da 3 regioni OFF, triplicherò la sample rate. In generale con  $nrOFF$  = numero di regioni OFF si avrà che

$TBTS = WS * T$  (non cambia, ogni regione ha il suo tempo di bootstrap)

$TSR = \text{timeseries rate} = T / nrOFF$

## PARAMETER 3: BS



Per poter effettuare il training dell'autoencoder si devono scegliere i seguenti parametri:

- \* batch size = BS
- \* epoch = ep

Le reti neurali accettano esempi che hanno la seguente shape: (BS, WS, NF). Dove NF = numero di features.

La fase di training consiste nel dare in input alla rete, uno o più samples alla volta. Quando si sono dati alla rete tutti i samples del training set (divisi in batch) un'epoca è terminata.

Solitamente il valore di epoch è  $> 1$ .

Una strategia di training in questo senso, potrebbe essere effettuata empiricamente con un batch size di 5 elementi. Anche in questo caso c'è un tempo di bootstrap: quello necessario ad ottenere il primo batch di samples.

Questo nuovo tempo di bootstrap si valuta considerando il tempo di bootstrap necessario ad ottenere la prima timeseries (=TBTS), dopodiché si deve aspettare di ottenere un numero di timeseries pari a (BS-1). Il numero di secondi per ottenere una timeseries è pari a TSR.

$TBB = \text{tempo di bootstrap batch} = TBTS + (BS-1) * TSR$

Dopo quanti secondi si ottiene il primo batch?

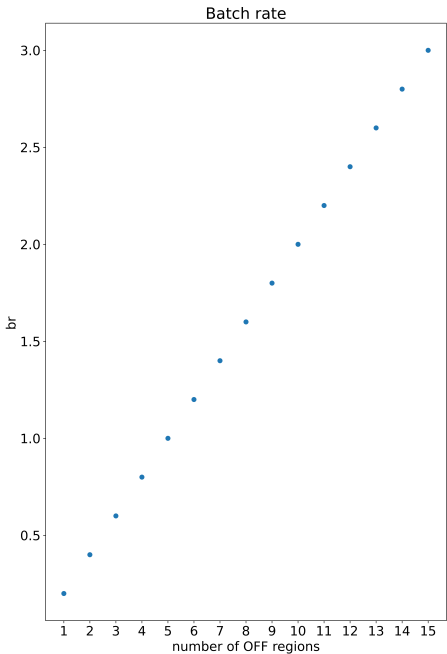
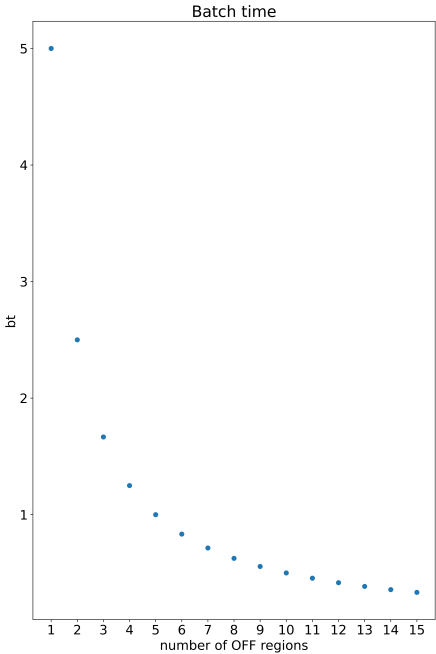
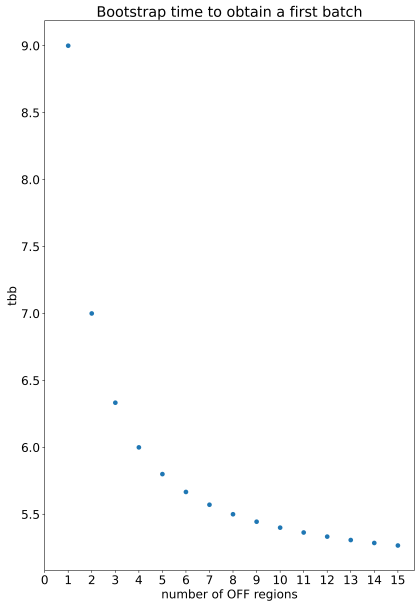
$T=1$   
 $WS=5$   
 $nrOFF=1$   
 $BS=5$   
 $TBB = WS * T + (BS-1) * ( T / nrOFF ) = 5 * 1 + 4 * ( 1 / 1 ) = 5 + 4 = 9 \text{ s}$

$T = 1$   
 $WS = 5$   
 $nrOFF = 2$   
 $BS = 5$   
 $TBB = WS * T + (BS-1) * ( T / nrOFF ) = 5 * 1 + 4 * ( 1 / 2 ) = 5 + 2 = 7 \text{ s}$

$T = 1$   
 $WS = 5$   
 $nrOFF = 4$   
 $BS = 5$   
 $TBB = WS * T + (BS-1) * ( T / nrOFF ) = 5 * 1 + 4 * ( 1 / 4 ) = 5 + 1 = 6 \text{ s}$

Superato il tempo di bootstrap, e ottenuto il primo batch, sarà possibile ottenere i successivi aspettando un tempo pari a:

$BT = \text{batch time} = BS * TSR$   
 $BR = \text{batch rate} = 1 / BT$



Il valore di BT sarà usato per simulare l'arrivo in streaming dei campioni per fare il training online della rete.