Three layer RNN:

* Input feature(request related feature)
  + Request(index)
  + Time
  + Address
  + Size
  + Type
* Ground true label
  + Oracle(optiml)
    - 可以讀取整個trace，所以知道未來的資訊。
    - 透過知道的資訊，算出每個request的benefit value，來決定是否cache
      * 在cache simulator內，oracle會模擬實際情形:
        + 每次access，都將資料放入cache，直到cache滿了。
        + 此時，若又有新的access進來，oracle就會去看，新進的request的Benefit value，和當下cache內的request benefit value。
        + 若cache內存在request的benefit value<新進request的benefit value，則將cache內該request踢出，將新進的request送進cache
      * 模擬結束後，依照已知資訊，新增兩個tag: cache tag, duration tag
        + Cache tag:每個request 是否cache(在cache simulator中跑過一輪，oracle會知道每個request實際上會不會被cache)
        + Duration tag:

每個在cache的request，從被放入cache，到被踢出cache之間，總共有多少個access進入cache(同上，模擬過後，oracle也會知道這項訊息)

依照number of access in cache將duration分成三種類別:soon ,mean ,late，而會位於哪種類別不僅僅取決於number of access更取決於cache size

* + - * 將這兩個tag送入RNN，當成ground true label
  + Benefit function(針對每個request)
    - HDD,SSD response time
    - Number of access
      * 公式中是以【Nacc-1】來表示:意義是，如果某個request被存取的次數只有1次，那此時，benefit function會是0，因為不希望將很少被存取的request放入cache
    - Size
      * 占用cache空間越大的request，對應的權重越小-公式中是以【】表示
    - Read ratio
      * 公式中以【+1】表示，意味著，如果來的request是write，假如沒有+1，那此時benefit function會直接歸零，但對資料進行寫入，不代表該資料應該從cache踢出，因此需要透過+1來維持住benefit function的權重