

2.12.1 Prefetching of Data

AMD family 19h processors implement data prefetch logic for its L1 data cache and L2 cache. In general, the L1 data prefetchers fetch lines into both the L1 data cache and the L2 cache, while the L2 data prefetchers fetch lines into the L2 cache.

The following prefetchers are included:

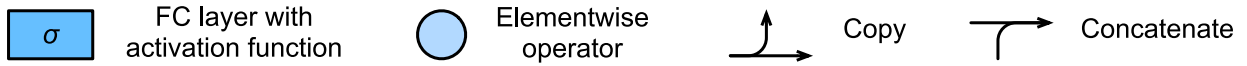
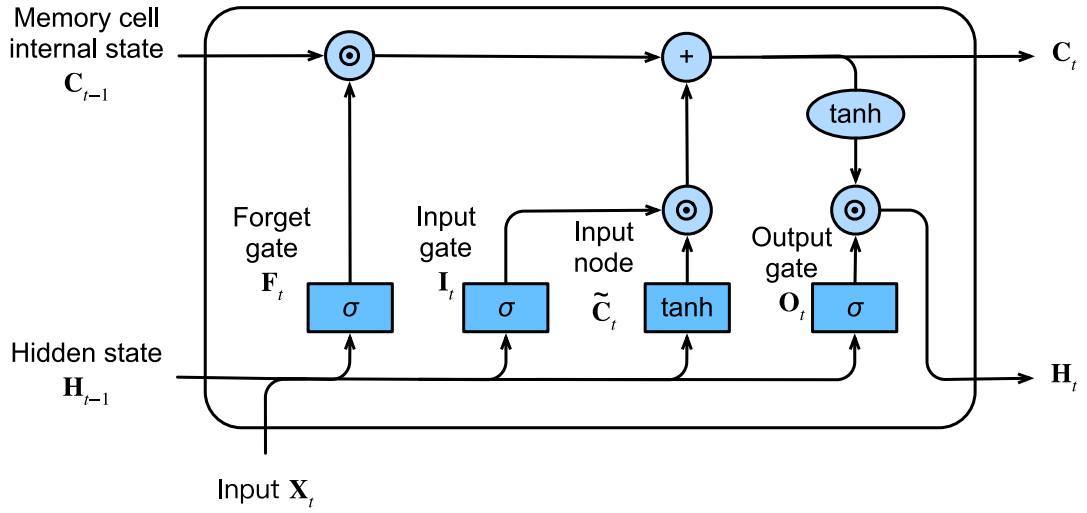
- L1 Stream: Uses history of memory access patterns to fetch additional sequential lines in ascending or descending order.



- L1 Stride: Uses memory access history of individual instructions to fetch additional lines when each access is a constant distance from the previous.
- L1 Region: Uses memory access history to fetch additional lines when the data access for a given instruction tends to be followed by a consistent pattern of other accesses within a localized region.
- L2 Stream: Uses history of memory access patterns to fetch additional sequential lines in ascending or descending order.
- L2 Up/Down: Uses memory access history to determine whether to fetch the next or previous line for all memory accesses.

For workloads that miss in the L1 or L2 caches, software may get improved performance if data structures are designed such that data access patterns match one of the above listed behaviors.

While prefetcher logic has been tuned to improve performance in most cases, for some programs the access patterns may be hard to predict. This can lead to prefetching data that will not eventually be used causing excess cache and memory bandwidth usage. This can be the case for workloads with random access patterns or less regular access patterns such as some database applications, etc. For this reason, some server variants of the family 19h processors support a prefetch control MSR that can individually disable or enable the prefetchers. See Processor Programming Reference for details on CPUID enumeration and MSR details.



Borrowed from D2L, Fig10.1.3 “Computing the memory cell internal state in an LSTM model.”
https://d2l.ai/chapter_recurrent-modern/lstm.html#fig-lstm-3