[0008] FIG. 33A-33B illustrate compute architectures 3300, 3320 configured to enable compressed transmission of neural network data to processing resources on a parallel compute processor or general purpose graphics processing unit, according to an embodiment. FIG. 33A illustrates a compute architecture 3300 in which data decompression is performed within a compute block 3 302. The illustrated compute block 3 302 may represent one of the compute blocks 2724A-2724N as in FIG. 27. FIG. 33B illustrates a compute architecture 3320 in which data compression and decompression is performed within a processing resource.

[0009] As shown in FIG. 33A, compute architecture 3300 includes a compute block 3302 and hardware scratch buffer 3304 that is coupled to memory 3308 via a DMA controller 3306. The memory 3308 can be main memory or system memory of a data processing system as described herein. The compute block 3302 includes a set of processing resources as described herein and can be similar to any of the compute blocks 2724A-2724N as in FIG. 27. The scratch buffer 3304 can be a high-speed on-chip memory, such as on-chip static random access memory (SRAM). In one embodiment the scratch buffer 3304 is optimized to store feature block units or kernel block units for neural network operations performed by the compute block 3302.

[0010] In one embodiment the decoder 3312 can be hardware decoder logic that is integrated into the compute block 3302 to enable compressed transmission of neural network data across the compute architecture. For example, when processing a CNN, the compute block 3302 can generate output feature map (OFM) data in the scratch buffer 3304 in an uncompressed format. An encoder 3316 integrated into the DMA controller 3306 to enable the writing the output feature maps data to the memory 3308 in a compressed format. When the OFM of one layer become the input feature map (IFM) of the next layer, those IFMs are read from memory 3308 as compressed data 3314 and stored in the scratch buffer 3304. The decoder 3312 can enable the compute block 3302 to read in the compressed data 3314 without requiring the data to be decoded. Alternatively, a codec unit having both encode and decode logic can be integrated into the DMA controller 3306, enabling compressed data to be transmitted and read by the DMA controller 3 306. The feature map data can then be decompressed by the DMA controller 3306 and written to the scratch buffer 3304 in an uncompressed format to be read by the compute block 3302.

[0011] In the embodiments described herein, the specific encoding format for kernel and feature data can be varied based on the statistics of the data to be encoded. Analysis of neural network feature map data indicates that many feature maps may be highly sparse. Analysis of neural network kernel data indicates that while the kernel data is not as sparse as the feature map data, many values in the kernel data are repeated. The dynamic range of kernel data is relatively low, which indicates that raw data allocate more bits than required to store the coefficients.