The Journal of Supercomputing (2023) 79:14827–14847
https://doi.org/10.1007/s11227-023-05255-7

URMP: using reconfigurable multicast path for NoC-based deep neural network accelerators

Yiming Ouyang[1] · Jiaxin Wang[1] · Chenglong Sun[1] · Qi Wang[1] · Huaguo Liang[1]

Accepted: 2 April 2023 / Published online: 12 April 2023

Abstract
Network-on-chip (NoC) exists with the advantages of high communication efficiency, scalability and reliability. In recent years, NoC-based deep neural network (DNN) accelerators have been proposed. Although existing NoC research solutions can solve the problem of the existence of one-to-one traffic in the network and transmit unicast traffic efficiently. However, due to the traffic characteristics of neural networks, there exists a large amount of one-to-many traffic, and if unicast is used to transmit multicast traffic, it may rapidly exhaust the network bandwidth and greatly degrade the performance of the platform. To solve the problem of a large amount of one-to-many multicast traffic existing in the network, we propose a path-based multicast mechanism that greatly exploits the traffic characteristics of neural networks and has excellent scalability. Also a router architecture that can efficiently replicate multicast packets and provide single-cycle per-hop transmission for multicast packets was designed. Detailed simulation results indicate that our proposed scheme can effectively reduce the classification delay, the average packet delay and the number of packets transmitted by the network.

Keywords Network-on-chip · Deep neural network accelerators · Multicast · Single-cycle per-hop transmission

1 Introduction

Deep neural network (DNN) is able to provide highly accurate recognition and classification results, exhibiting enormous advantages in applications such as information processing, pattern recognition and computer vision [1, 2], but this comes at the cost of high computational complexity and high-power consumption.

✉ Jiaxin Wang
wangjiaxin3212@163.com

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

Recent neural network models, such as AlexNet [3] and VGG-16 [4], are unable to run efficiently on existing platforms due to their more complex structures and large number of parameters.

Currently, common DNN accelerators are based on ASCIs, FPGAs, CPUs and GPUs [5]. ASCIs are designed for specific applications and provide some solutions for particular DNN models, but such designs lack flexibility, are not reconfigurable, perform poorly under different applications and incur significant design costs. FPGAs are more flexible than ASCI-based DNN accelerators due to their programmable nature [6], but they are still designed for particular DNN models and application configurations. CPUs are designed for general-purpose computations, such as 48-core Qualcomm Centriq 2400 and 72-core Intel Xeon Phi [7], with high computational flexibility to perform complex computations. Most of the DNN computations are simple operations, such as product and addition operations, which require intensive parallel computation; hence, CPUs are unsuitable as DNN accelerators platforms. GPUs are the most frequently used DNN gas pedal platforms [8], with high computational flexibility and the ability to perform intensive parallel computations, but also impose a significant power consumption overhead [9].

The NoC-based DNN accelerators are a suitable choice. Network-on-chip (NoC) separates the computation function and communication function in traditional bus systems, where the PE (process element, PE) is responsible for the computation of neurons, and the router is responsible for the data transmission, which can decouple the DNN model into data computation and data transmission operations [10]. In the data computation part, the computation of DNN models can be executed independently of the data stream. In the data transmission part, NoC can realize efficient data transmission operations. Meanwhile, the computation results of one PE (process element, PE) can be directly transferred to another PE through the network, which reduces the memory access to off-chip memory and effectively reduces the transmission delay and power consumption. NoC-based DNN accelerators are more flexible in design compared to ASCI- or FPGA-based DNN accelerators, more computational parallelism compared to CPU-based DNN accelerators, and lower power consumption compared to GPU-based DNN accelerators [11].

Although the NoC-based DNN accelerators can well satisfy the demand of intensive parallel computation of DNN models with low-power consumption, the presence of a large amount of one-to-many traffic in NoC is also a performance bottleneck of DNN accelerators. If the traditional unicast packet transmission method is used, a large number of packets carrying the same data will be generated in the network, rapidly exhausting the network bandwidth and causing network congestion and increased packet transmission delay [12]. Therefore, using an effective multicast strategy to determine how packets are efficiently replicated for transmission in the network is an integral part of the NoC-based DNN accelerators. Considering the special traffic characteristics of the DNN model, a path-based multicast mechanism that can effectively reduce the number of packets in the network is proposed. Also, in order to reduce the transmission delay of multicast packets, a router architecture is

designed to provide a dedicated single-cycle per-hop transmission path for multicast packets. The main contributions of this paper are entered under:

- Based on the traffic pattern of DNN, we propose a path-based multicast mechanism to solve the problem of large amount of one-to-many multicast traffic that always exists in the network.
- To effectively reduce the number of hops for multicast packet transmission, an efficient reconfigurable multicast path is designed that can change the length and location based on the traffic in the network, while the multicast path also has a high reuse rate.
- A router architecture is designed to efficiently replicate multicast packets at the destination node and provide single-cycle per-hop transmission for multicast packets. The transmission delay of packets is effectively reduced.
- Detailed simulation results comparing different scale DNN models mapped onto the NoC-based accelerators platform with traditional router architectures and other router architectures demonstrate that our proposed strategy can effectively diminish the classification delay, the average packet delay and the number of packets transmitted by the platform.

2 Related work

2.1 Multicast mechanism

Unicast refers to one-to-one communication in the network, where one PE sends data to another PE; multicast means to one-to-many communication, where one PE sends data to multiple PEs and broadcast ensures one-to-all communication, where one PE sends data packets to all PEs [13]. The results of the neurons in the upper layer of the DNN model will be used as the input of the neurons in the next layer and sent to all neurons in the next layer. Therefore, there will be a large number of multicast packets in the accelerator platform, and a suitable multicast mechanism needs to be designed, and there are two common multicast mechanisms of path-based and tree-based in NoC.

In the path-based multicast mechanism [14, 15], multicast packets choose a path that connects all destination nodes, and packets are replicated only when they reach a destination node. This method is simple to implement in hardware, minimizes the number of packets in the network and is less prone to blocking, but the average hop count of packets is longer. In [16], a path-based link-oriented multicast scheme is proposed. The scheme uses wormhole switching, and the multicast process consists of three phases: establishment, communication and release and multicast packets reach each destination node through the path established in the establishment phase.

In the tree-based multicast mechanism [17, 18], multicast packets are transmitted along the common path as much as possible, after which the packets are replicated at the appropriate nodes based on the location of the destination nodes and the multicast routing algorithm. The tree-based mechanism has a shorter average hop count of packets, but is prone to blocking, where blocking in one branch affects the

transmission in another branch, and also requires solving the deadlock problem. In [19], a tree-based multicast scheme is used, along with up and down point-to-point routing to target a specific subtree and recursive branches to enable multicast packets to be transmitted in that subtree, avoiding deadlock by restricting the branches to the downward direction.

2.2 NoC-based DNN accelerators

As more and more cores are integrated into the chip, the chip becomes more sensitive to communication delays [20]. NoC is a widely used on-chip communication subsystem with high communication efficiency, great scalability and high reliability. By decoupling the communication operation and computation operation of neural networks, the intermediate results of the computation of the previous PE in the NoC-based DNN accelerators can be sent directly to the next PE through the router, reducing the access to off-chip memory. And multiple PEs can perform computation simultaneously, which is well satisfied with the intensive parallel computation required by neural networks. NoC-based DNN accelerators have been extensively studied in recent years [21–23].

A design paradigm for neural network accelerators based on reconfigurable microswitch arrays was proposed by Kwon et al. Accelerators provide dedicated paths for traffic in the network by reconfiguring the microswitch arrays [24]. This scheme provides a scalable solution for NoC-based neural network accelerators in four aspects: latency, area, throughput and area.

Tang F et al. proposed a DNN accelerators with a tree-based multicast mechanism that effectively reduces the number of packets in the network [25]. A single-flit router architecture was also designed, which does not have the problem of flit head blocking. The scheme has good performance in terms of classification delay, packet delay, number of packets transmitted by the network and power consumption.

Other researchers have also proposed a series of neural network accelerator simulators based on Noxim [26, 27]. NN-Noxim [28] is a cycle-accurate NoC simulator that can support artificial neural networks (ANNs) and proposes various mapping schemes to map ANN models to NoCs to perform computational tasks. CNN-Noxim [29] is designed for convolutional neural network (CNN) models, and the scheme flattens the CNN model into an ANN model and then maps the neuron clusters onto the NoC. DNNoC-sim [27] can dynamically map neurons onto NoCs and is suitable for large-scale DNN models on resource-constrained NoC platforms. All these studies provide inspiration for our work.

3 NoC-based DNN accelerators architecture

3.1 Overall architecture

Figure 1 exhibits the architecture of the NoC-based DNN accelerators, where the processing element (PE) and the on-chip memory module are interconnected by

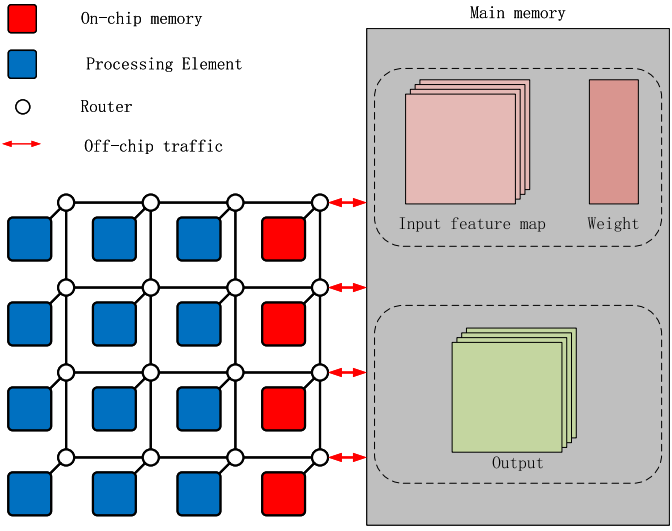

Fig. 1 The accelerator architecture

a mesh network. The rightmost router of the network is not only used for on-chip traffic communication, but also for exchanging information with the off-chip main memory. The DNN model contains a large number of parameters, and the traditional mesh network's on-chip memory is insufficient to store a large number of parameters, which requires frequent exchange of information with the off-chip main memory and affects the performance of the accelerator. Therefore, we set up a column of on-chip memory module connected to the rightmost router of the network, which is responsible for temporarily storing some copies of parameters in the main memory and some intermediate results of PE calculations. Parameters such as pictures and weights in main memory are stored in the on-chip storage module. Then, the router sends these data to the PE mapped by the first layer of DNN neuron clusters, and the PE sends the calculation results to the next layer of PE through the router. Through the calculation of each PE layer, the final calculation result is output through the router.

3.2 Neural network reshape

The time for PEs to complete data computation in NoC is called computation time, and the time for routers to transmit data is called communication time. If each PE computes only one neuron at a time, then the PEs will complete the data computation quickly, while the communication between multiple PEs requires the routers to transfer data frequently, resulting in the communication time of the accelerator will be much longer than the computation time. However, if a PE computes too many neurons, the computation time of the accelerator will be much longer than the communication time. The difference between computation time and communication time is too large to reduce the efficiency of the accelerator, so it is necessary

to reasonably divide the number of neurons to PE so that the computation time and communication time are closer. Meanwhile, in order to fully exploit the forward propagation property of the neural network model, i.e., the output of the neuron in the upper layer will be used as the input of the neuron in the lower layer. We adopt the mapping method of mapping neurons of the same layer of the neural network on the same row PE of NoC. It makes the multicast traffic pattern in the network more fixed, which, in turn, makes the construction of multicast paths in the multicast mechanism simpler.

Figure 2a shows a four-layer artificial neural network (ANN), where the output results of the neurons in the upper layer of the ANN are used as inputs to the neurons in the next layer and sent to all neurons in the next layer. Figure 2b shows the result of dividing the neurons for Fig. 2a. To make the computation time and communication time closer, we divide multiple nerves into a cluster of neurons. Here, we divide at most two neurons into one neuron cluster, and each neuron cluster can only contain neurons from the same layer. After dividing all neurons of the neural network, the communication of neurons becomes the communication between neuron clusters, and then, neuron clusters are mapped to the mesh network. Figure 2c represents the mapping of Fig. 2b to the mesh network, assuming that each router contains only one PE and each PE handles one neuron cluster. Each row of the NoC maps only the neuron clusters at the same layer on the ANN, and the communication between different neuron clusters exchanges information through the NoC.

3.3 Multicast path selection

It is worth noting that DNNs tend to have a fixed communication pattern for multicast traffic in the workload. The output results of the neurons in the upper layer of the DNN are sent to all neurons in the lower layer as inputs to the neurons in the lower layer. Considering the specificity of DNN traffic patterns and the mapping of neuron clusters to NoCs, when the PEs of neuron clusters in the upper layer of DNN send packets to the PEs of neuron clusters in the next layer, the locations of the destination nodes of their packets are adjacent to each other and multiple nodes in the upper layer send packets to the destination nodes in the next layer. Therefore, we propose a path-based multicast mechanism, in which all nodes containing neuron

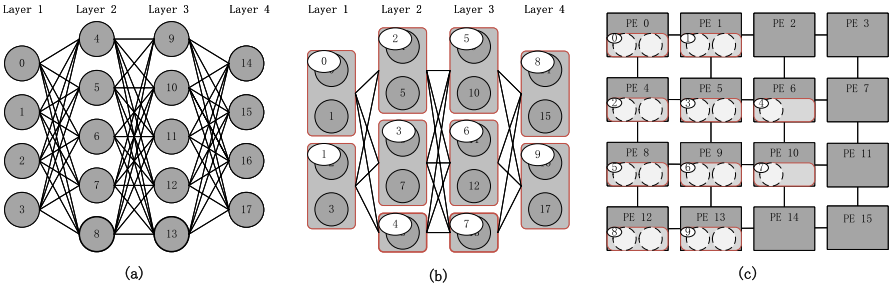

Fig. 2 **a** Artificial neural network, **b** divide neurons into neuron clusters and **c** map neuron clusters to NoC

clusters in the adjacent two layers are connected in series on the same path, and multicast packets are transmitted along this path. In order to reduce the average hop count of multicast packets, a router architecture that can dynamically reconfigure the length and location of multicast paths based on the mapping of neuron clusters to mesh networks and provide a single-cycle per-hop transmission path for multicast packets on the path is designed. The clusters of neurons in both layers are contained on a single multicast path, which also ensures a high reuse rate of the path.

The output of the neurons on the upper layer of the neural network will be sent to all the neurons on the lower layer as input, and the calculation results shown as PE on the mesh network will be sent to all the PEs on the lower layer through the router. A portion of the neural network model mapped onto the mesh network is illustrated in Fig. 3. The gray nodes in each row of the network indicate the presence of neuron cluster mapping on the PE of that node, the neuron clusters in each PE row are from the same layer of the neural network, the blank nodes indicate that no neuron cluster mapping exists for that node and the red nodes represent that the router is connected to an on-chip memory module. Nodes 1–3 in the first layer need to send multicast packets as source nodes to destination nodes 6–9 in the second layer, respectively. If multicast paths are established separately for nodes 1, 2 and 3, it takes a lot of time to establish the path in the network, and the reuse rate of the links is very low. Therefore, all source nodes in the same layer and all destination nodes in the next layer will be included in the same multicast path to ensure that the multicast link has a high reuse rate, as shown by the blue dashed line in Fig. 3. Node 5, which is connected to the on-chip memory module, needs to send parameters such as weights to nodes 11 and 12 of the layer 3 neuron mapping, and the acceleration platform similarly establishes a multicast path to provide single-cycle per-hop transmission for multicast packets. Since we did not add other bypasses to the crossbar, the same router cannot be involved in the construction of multicast path in two different directions at the same time in order to ensure the correct transmission of unicast packets.

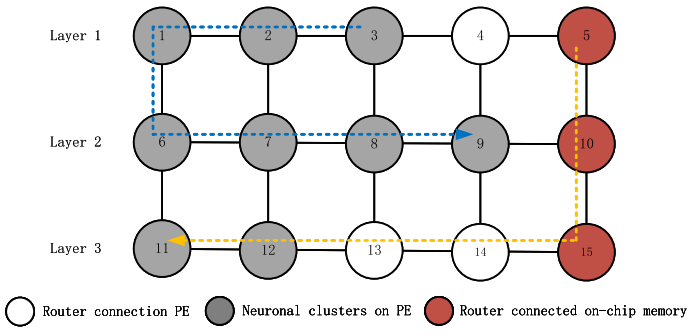

Fig. 3 Multicast path

4 Router architecture

4.1 Microarchitecture

The router architecture is depicted in Fig. 4 and consists of buffers, a packet processing unit, a routing computation unit, a link reconfiguration unit, a reconfigurable crossbar, input and output ports and several multiplexers and demultiplexers. The packet processing unit is mainly responsible for generating multiplexer strobe signals and modifying the address information of multicast packets. The routing computation unit finds the appropriate output port for unicast packets. The link reconfiguration unit configures the reconfigurable crossbar so that the reconfigurable cross switch can maintain a unidirectional link. The reconfigurable crossbar is not only responsible for serially sending unicast packets to each output port, but also for reconfiguring a fast Channel for multicast packets to form a dedicated multicast path between multiple routers. In order to simplify the structure of the reconfigurable crossbar, we did not add other bypasses to the reconfigurable crossbar. When the multicast packet reaches the destination node, the multiplexer of the local output port can directly send the packet to the local output port. The multicast packet can be copied without going through the crossover switch.

Packets enter the packet processing unit from the input port, and the packet processing unit extracts the packet type. For multicast packets, this is shown in Fig. 5a. If the packet does not reach the destination node, the packet is bounced by the packet processing unit and bypasses the buffer to reach the crossbar, after which it is sent directly to the output port through the path in the reconfigurable crossbar to reach the next router. If a multicast packet arrives at a destination node, the packet

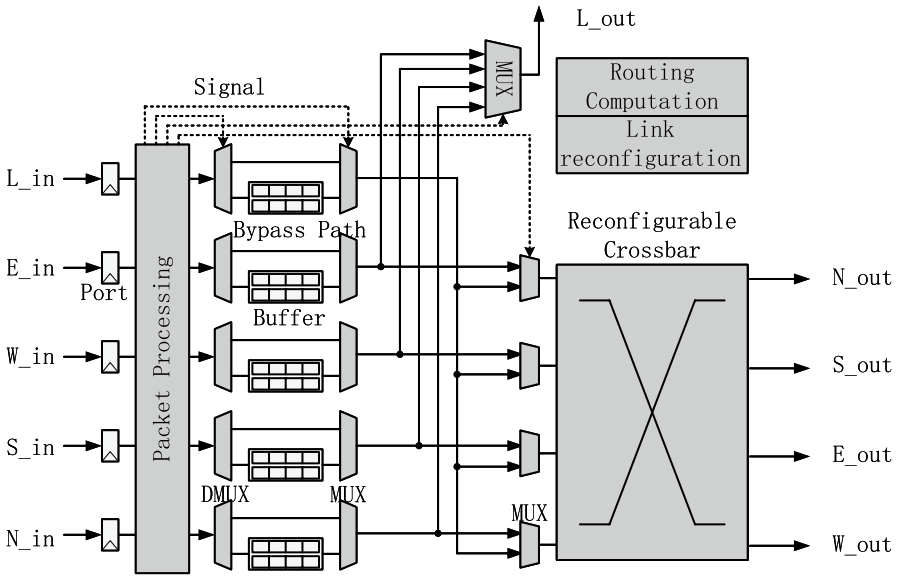

Fig. 4 Router architecture

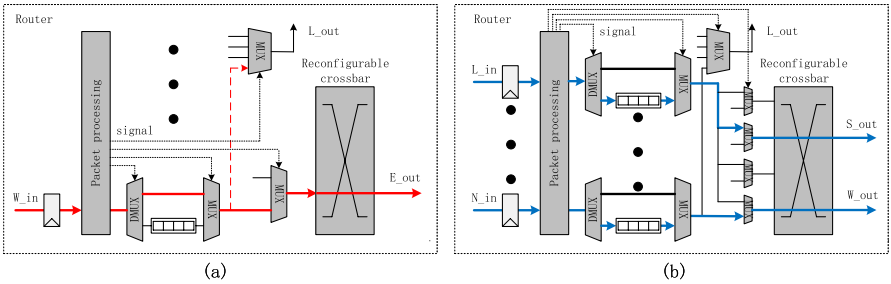

Fig. 5 **a** Multicast packets pass through the router, and **b** unicast packets pass through the router

processing unit also needs to modify the packet's header information, and the multicast packet crosses a 4:1 multiplexer to reach the local output port while crossing the crossbar. Multicast packets do not go through buffer write (BW), route computation (RC), VC allocation (VA), switch allocation (SA) and switch traversal (ST), except the first packet needs to be multicast path construction, the other packets can be sent directly to the output port after only one packet processing stage, thus ensuring the single-cycle per-hop transmission of multicast packets.

For unicast packets, this is shown in Fig. 5b. The packet is ejected from the packet processing unit and temporarily stored in the buffer. If the input port of the packet is not involved in the construction of the multicast path, the packet is ejected from the buffer and sent to the corresponding output port via the reconfigurable crossbar. Otherwise, since the reconfigurable crossbar has no other bypass, unicast packets in that input direction can only be ejected from the buffer and transmitted along the multicast path when the path is idle. Packets from the local input port, unicast packets are sent directly from the idle input port of the reconfigurable crossbar. And multicast packets are sent to the next node of the path through the fast channel configured by the reconfigurable crossbar.

4.2 Packet format

The accelerator platform supports two types of packets, unicast and multicast, and the packet formats are shown in Fig. 6a and b, respectively. The fields are explained as follows.

Type: 1 bit, 0 means the packet is a unicast packet, and 1 means the packet is a multicast packet.

Destination address: Unicast packets occupy a total of 12 bits, using the coordinates of the destination node in the mesh network to indicate the location of the destination node, the coordinates x and y occupy 6 bits, respectively; multicast packets use an n-bit vector, n indicates the number of nodes in the network, 1 on each bit means the node is the destination node and 0 means not the destination node. When the packet reaches the destination node, the packet processing unit in the router sets the 1 in the corresponding position of the vector to 0 and deletes the packet when the vector is all 0 s.

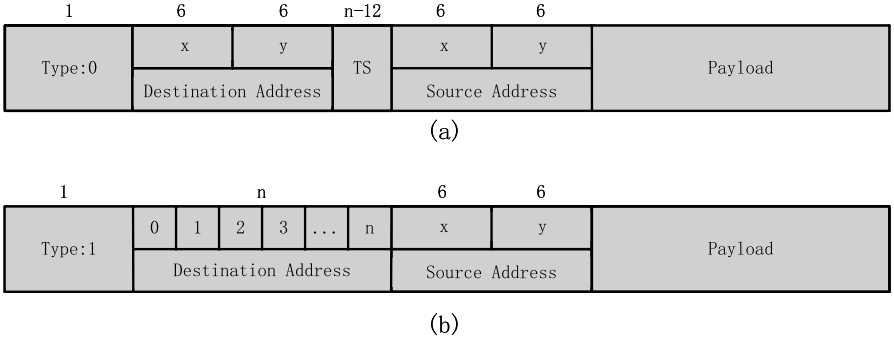

Fig. 6 **a** Unicast packet format and **b** multicast packet format

TS indicates the timestamp. When multicast packets and unicast packets contend, multicast packets have higher priority. To prevent unicast packet starvation, a timestamp field is set for unicast packets, and the number of bits of the timestamp is (n-12) bits. When the value of the timestamp is greater than the threshold value we set, the unicast packet is sent first.

Source address: Unicast and multicast packets are the same, both use coordinates to indicate the location of the source node, with coordinates x and y taking up 6 bits each.

Payload: The valid data carried by the packet, the length of this field changes with the specific application.

5 Multicast path configuration

5.1 Crossbar reconfiguration

Each layer of ANN and DNN is divided into neuron clusters that are mapped to mesh networks, and the neural network to mesh network mapping is determined. The output of the neurons in the upper layer of ANN and DNN is sent as input to all neurons in the next layer, and the model parameters are generally sent to one or several layers of neurons. Due to this traffic characteristic of the neural network model, the multicast packet destination node will be in one or several consecutive layers of the mesh network. Multicast packets enter the packet processing unit from the input port and later cross the bypass to reach the crossbar. Figure 7 indicates the configuration process of the multicast path in the accelerator platform when the computation results of the neurons in layer i are sent to all neurons in layer i+1. By configuring reconfigurable crossbar in the router, a link is formed between the input and output of the crossbar, and multicast packets are transmitted directly from this link to the output port of the router without going through the switch allocation (SA) and switch traversal (ST) stages. We reconfigure all routers with neuron cluster mapping at layer i and layer i+1 to connect all source and destination nodes in a single path, enabling single-cycle per-hop transmission of multicast packets between source and

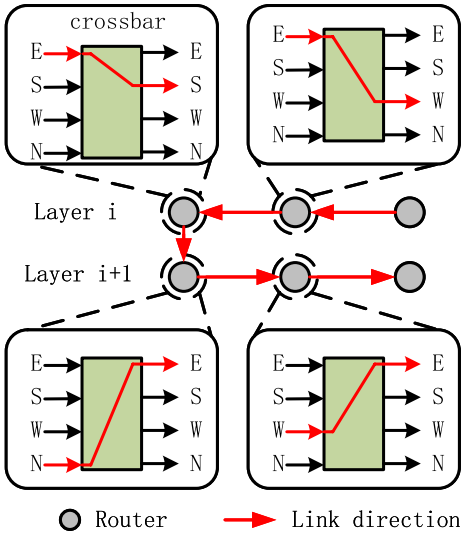

Fig. 7 Example of multicast path configuration

destination nodes and also enabling a high reuse rate of multicast paths. A source node will send a path release signal after the transmission is completed. When all source nodes complete the transmission, the multicast path will be released.

In the case that the current router does not participate in the multicast path configuration, after the unicast data packets arrive at the reconfigurable crossbar from the buffer, the unicast data packets use the YX routing algorithm to reach the appropriate output port through the crossbar. In the situation where the current router is involved in multicast path configuration, assuming that the path direction is from east to west, in the absence of multicast packet contention, except for the unicast packets in the east input direction can still be output to the corresponding output port normally. As for the input from the east direction, there is no redundant bypass inside the crossover switch in order to simplify the structure of the reconfigurable crossbar. Therefore, the input from the eastern direction can only be transmitted along the link. Also, to prevent unicast packet Livelock, we set a timestamp within the unicast packet, and when the timestamp is greater than the threshold we set, the unicast packet will be forced to be sent to the appropriate output port. The threshold is half of the maximum delay that the unicast packet can tolerate while the program is running.

5.2 Multicast path configuration algorithm

According to the special traffic characteristics of ANN and DNN models, we design a simple reconfiguration algorithm for multicast path configuration in mesh networks. By the mapping algorithm in Sect. 3.2 is adopted, only neuron clusters from one layer on one row of the mesh network, and neuron clusters from the same layer will be included in the same multicast path. The configuration direction of a default multicast path is specified, and in cases where no path needs

to be constructed, the reconfigurable crossover switch does not build the link and is primarily responsible for sending unicast packets to the appropriate output port. When a router needs to configure a path, and no reconfigurable crossbar link direction is specified the reconfigurable crossbar uses the default configuration direction as depicted in Fig. 8a. However, due to the different mapping of different neural network models to the NoC, some paths may be too long if the initial method is used for all multicast paths. Therefore, we define the node of the reconfigurable crossbar link direction that needs to be modified as an inflection point. In other words, by modifying the link direction of the reconfigurable crossbar in the inflection point, the length and location of the multicast path can be changed.

Algorithm 1 Multicast path configuration algorithm

Input: (X_S,Y_S),(X_L,Y_L),Des_L
Output: $Container$

if The source node connects to the PE **then**
  $Y \leftarrow Y_s$
  while $Y + 1 <= Des\_L$ **do**
    if $X_L < (X + 1)_L$&&Link direction from west to east **then**
      Add $((X + 1)_L,Y)$,(W,S) and $((X + 1)_L,Y + 1)$,)(N,W) to $Container$;
    else if $X_L >= (X + 1)_L$&&Link direction from west to east **then**
      Add (X_L,Y),(W,S) and $(X_L,Y + 1)$,)(N,W) to $Container$;
    end if
  end while
else if The source node connects to the on-chip memory unit **then**
  if $Y_S < Des\_L$ **then**
    Add $(X_S,\ Des\_L)$,(N,W) and $(X_S - 1,\ Des\_L)$,(E,W) to the $Container$;
  else
    Adding $(X_S,\ Des\_L)$,(S,W) and $(X_S - 1,\ Des\_L)$,(E,W) to the $Container$;
  end if
end if
if Inflection point turning direction conflicts with the direction of other links **then**
  Waiting for other links to be released;
else
  Configure multicast links based on the inflection points and directions in the $Container$;
end if

Depending on the mapping of neuron clusters onto the mesh network, the multicast paths in the mesh network have to be changed accordingly. As shown in Fig. 8b, gray nodes indicate the presence of neuron clusters mapped to the node, blank nodes mean that no neuron clusters are mapped to the node and red nodes represent the router connected to the on-chip memory unit. The clusters of neurons in layer 1 of the mapped neural network on nodes 1–7 in the network need to send packets to nodes 11–13 where the clusters of neurons in layer 2 are located. The routers where the first layer neuron clusters are located act as source nodes to send

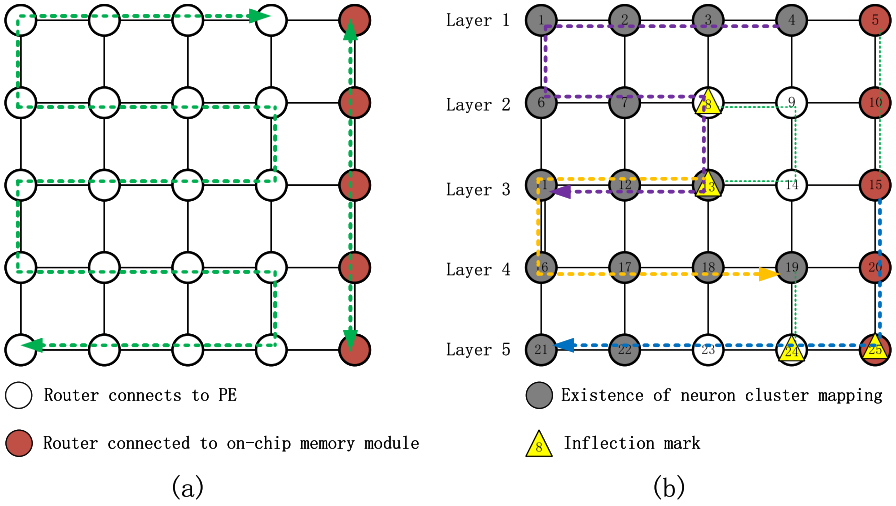

Fig. 8 **a** default multicast path and **b** configure multicast links based on neural network to NoC mapping

multicast packets, respectively, and multiple routers where the second layer neuron clusters are located act as destination nodes for multicast packets. The multicast path is established between the source and destination nodes, and the length of the multicast path can be minimized by changing the direction of the reconfigurable crossbar of nodes 8 and 13 on the base path. Furthermore, nodes 11–13, respectively, act as source nodes to send packets to nodes 16–19, with the multicast path configuration shown as the orange dashed line. Since nodes 11 and 13 have the same link direction as the source and destination nodes, there is no need to wait for the link release at layer 1 and layer 2. Similarly, node 15 needs to send parameters and other information to nodes 21 and 22 of the layer 4 neuron cluster mapping, and a single-cycle per-hop multicast path can be constructed by modifying the direction of the reconfigurable crossbar path of the routers in nodes 24 and 25.

For simplicity, we consider the neurons of each layer mapped to the mesh network from top to bottom (and vice versa from bottom to top). In the mapping method, we adopted, the leftmost node of the mesh network always has a mapping of neuron clusters. To simplify the path length, we only need to consider whether the nodes on the right side of the mesh network need to modify the direction of the path. Therefore, for the communication between the neural network layers, it is only necessary to consider the path from the west to the east and compare it with the abscissa of the router with the neuron cluster mapping on the right side of the next layer. The larger one is defined as the abscissa of the inflection point, and the ordinate is the row of the mesh network where the path is located. For the communication between the on-chip memory unit and the neural network, the last two nodes in the row where the destination node is located can be used as the inflection point. The specific algorithm for finding the inflection point is shown in Algorithm 1. (X_S,Y_S) denotes the source node coordinates, (X_L,Y_L) denotes the coordinates of the last router in the yth row where the neuron clusters mapping exists and Des_L denotes the row where the

destination node is located. The output container denotes the save path inflection point container.

6 Evaluation

6.1 Simulation setup

We use the cycle-accurate NoC simulator improved by Noxim to evaluate our proposed DNN accelerators URMP. And run traces are collected from the SPLASH-2 benchmark. The trace is generated using gem5 in system call simulation mode. We compare URMP with two existing router architectures: (1) Base: The traditional router architecture in NoC, where the input port contains multiple virtual channels, and the crossbar sends packets to the output port according to the routing algorithm, can efficiently handle one-to-one traffic in the network. (2) MRSB: A router architecture that supports tree-based multicast transmission mechanism, storing the destination address information in multicast packets in a shared buffer and combining part of the destination address in the shared buffer and the stored valid data in the input buffer into a new packet transmission method when the packet is output [17].

We simulated different sizes of neural network models, as indicated in Table 1. According to the size of the neural network, we set up different sizes of 2D Mesh network, the router in the rightmost column of the network is not only responsible for the transmission of intra-chip traffic, but also for exchanging information with off-chip memory. As shown in Table 2, we assume that the router has 3 virtual channels (VCs) per input buffer, each VC has a buffer depth of 32 flits and each packet consists of 16 flits. All three router architectures use the YX routing algorithm for unicast packets, the Base router for multicast packets uses unicast to handle multicast traffic, MRSB uses tree-based multicast transmission and URMP uses multicast links to transmit multicast packets. Packet transmission on a dedicated link is a single cycle per hop, and packets not using a dedicated link are every three cycles per hop. The peak performance of PE in the paper is 86.4 GOPS with a clock frequency of 1 GHz and a channel bandwidth of 128 bit. To measure the impact of multicast mechanism on the NoC-based DNN accelerators under different schemes, we made a comparison in three aspects: classification delay, average packet delay and the number of packets transmitted by the network. To demonstrate the effectiveness of adding on-chip memory units to the accelerator, we replaced the on-chip memory units with PEs in Base and MRSB and compared the difference in the number of off-chip memory accesses.

6.2 Classification delay

The NoC-based DNN accelerators platform reads the data from the main memory and stores it in the on-chip memory unit, and then, the router sends the data to the PE mapped by the first layer of DNN neuron cluster, and the PE sends the

Table 1 Experimental parameters

Neural network model	Input image
$400 \times 400 \times 100$ ANN	28×28
Lenet-5	32×32
AlexNet	$227 \times 227 \times 3$
VGG-16	$224 \times 224 \times 3$
MobileNet	$224 \times 224 \times 3$
EfficientNet	224×224

Table 2 Experimental parameters

Parameters	Settings
Topology	6×6 Mesh ($400 \times 400 \times 100$ ANN)
	8×8 Mesh (Lenet-5)
	10×10 Mesh (AlexNet)
	16×16 Mesh (VGG-16)
	16×16 Mesh (MobileNet)
	12×12 Mesh (EfficientNet)
Routing	*Y-X* Routing (Base)
Buffers per port	12 (3 virtual channels)
PE performance	86.4 GOPS
Frequency of router	1 GHz
Memory bandwidth	2 GB/s
Channel width	128 bits

calculation results to the next layer of PE through the router. Through the calculation of each layer of PE, the calculation result is finally output by the router. The classification delay is the time required for the DNN accelerators platform to complete the above process.

The classification delays with different design parameters are given in Fig. 9. The vertical coordinate indicates the normalized classification delay, and the horizontal coordinate indicates the gradually increasing number of neuron clusters mapped on the PE. The mesh networks used in (a), (b), (c) and (d) are relatively small in size and can take advantage of the path-based single-hop per-cycle transmission of URMP. URMP reduced classification delays by 31%, 30%, 14% and 18% on average compared to MRSB, and by 45%, 43%, 39% and 37% on average compared to Base. Due to the relatively large size of the mesh network used in (d) and (f), multicast packet transmission using path-based transmission may experience more hops than tree-based transmission to reach all destination nodes, but we use single-cycle per-hop transmission to compensate for this disadvantage. URMP reduces classification latency by an average of 14% and 5% over MRSB, and reduces classification latency by an average of 32% and 20% compared to Base. (e), (f) The lack of significant reduction in classification delay as

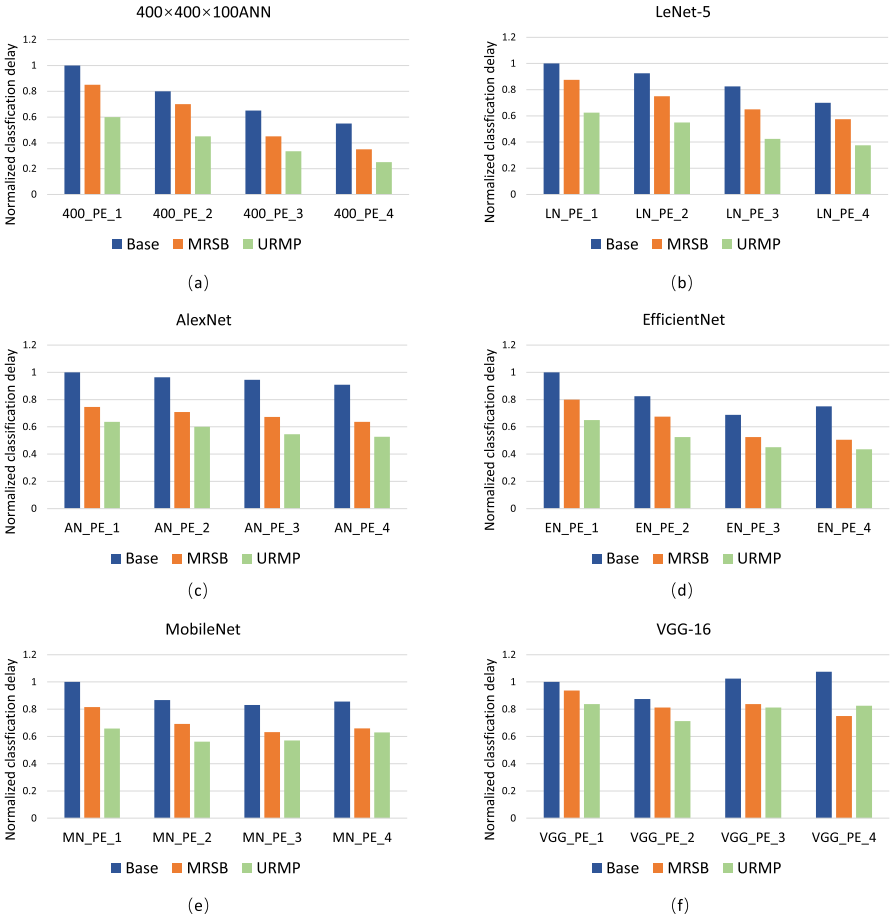

Fig. 9 Normalized classification delay under different neural network models

the number of neuron clusters on PE increases is due to the relatively large size of VGG-16 and MobileNet and the excessive number of neuron clusters on PE, resulting in a computation time that far exceeds the communication time.

6.3 Number of packets transmitted

Unicast packets can be transmitted efficiently on the existing NoC platform, but for multicast packets, if multicast packets are transmitted in a unicast manner, the network will be flooded with a large number of duplicate packets, rapidly exhausting the network bandwidth. Therefore, we believe that the number of packets transmitted during a DNN accelerators running a neural network model is also an important indicator of the performance of platform.

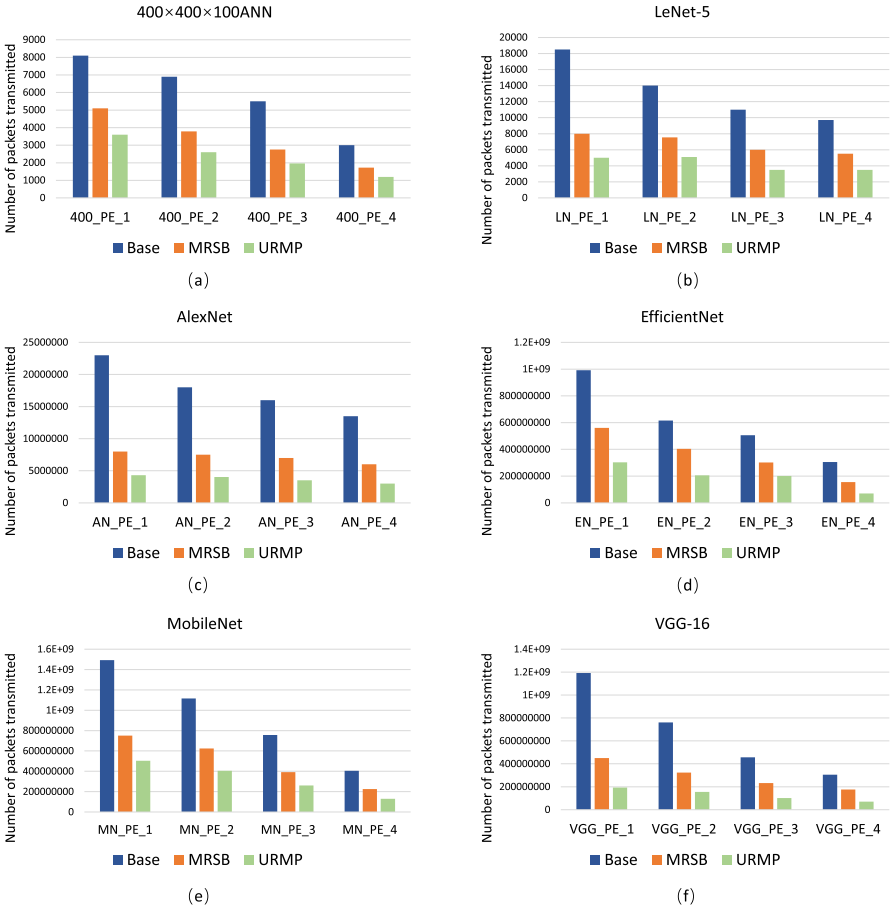

Fig. 10 The number of packets transmitted by the network under different neural network models

The number of packets transmitted by the accelerator platform with different design parameters is given in Fig. 10. The vertical coordinate indicates the number of packets transmitted, and the horizontal coordinate indicates the gradual increase in the number of neuron clusters mapped on the PE. As the number of neuron clusters on PE increases, the number of packets transmitted in the network also decreases gradually. (a–f) show an average reduction of 61%, 68%, 79%, 68%, 66% and 80% in the number of packets transmitted in the URMP compared to Base in the network. The number of packets transmitted compared to MRSB is reduced by 30%, 37%, 48%, 45%, 35% and 55% on average.

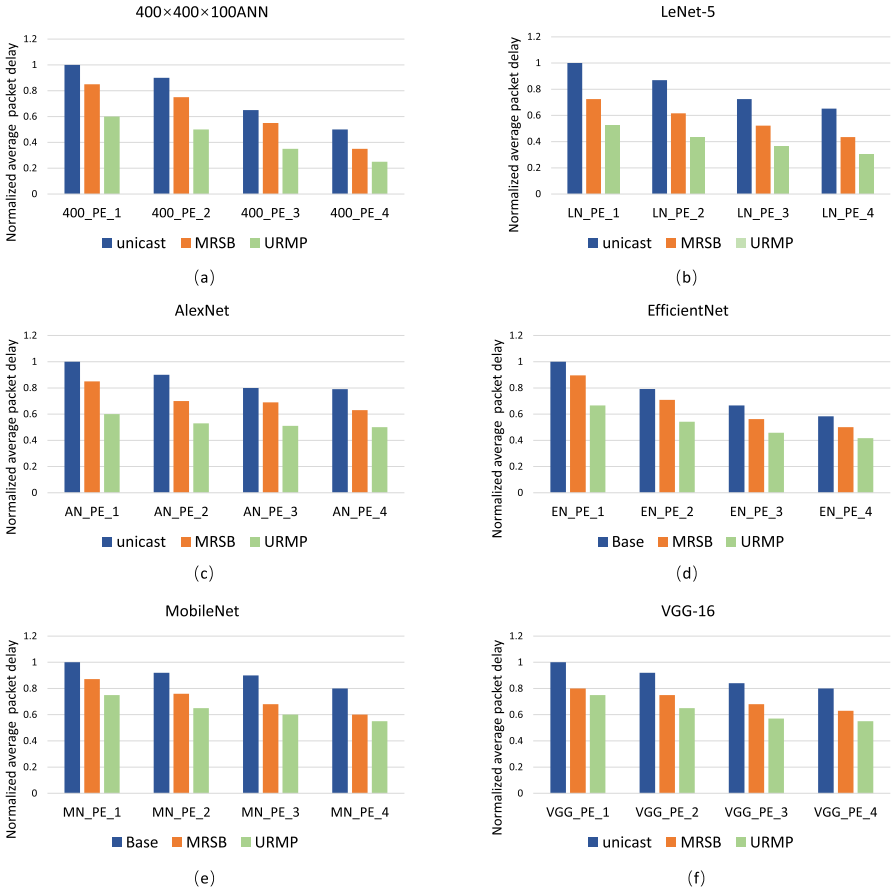

Fig. 11 Average packet delay with different neural network models

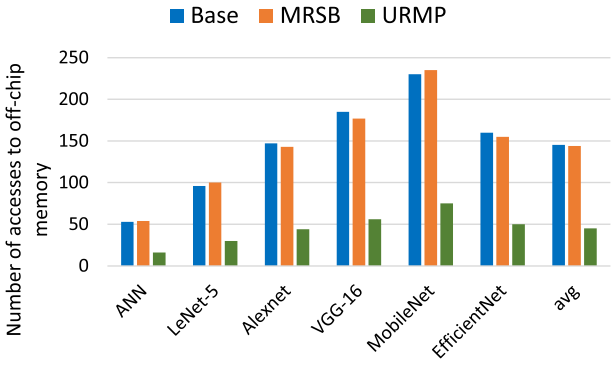

Fig. 12 Number of accesses to off-chip memory

6.4 Average packet delay

The output of the neuron on the upper layer of the neural network will be sent to multiple neurons on the lower layer as the input of the neuron on the lower layer. In NoC, the calculation result of the upper layer neuron cluster mapping to PE will be sent to the PE on the lower layer through the router. Average packet delay is the average delay of packets in the network from the source node to the destination node.

The average packet delay of the accelerator platform with different design parameters is given in Fig. 11. The topology of the mesh in (a–d) is relatively small, and the multicast mechanism of path-based single-cycle per-hop transmission can perform well. The average is packet delay reduction of 34%, 49%, 38% and 31% compared to URMP and Base. The average packet delay is reduced by 20%, 30%, 25% and 22% compared to MRSB. Although (e) and (f) use larger mesh networks and multicast packets using path-based transmission have more hops to reach all destination nodes than tree-based transmission, URMP uses a single-cycle per-hop approach to handle multicast traffic, which still has some performance advantages. The average packet delay reduction between URMP and Base is 29% and 26%. Compared to MRSB, the average packet delay is reduced by 12% and 13%.

6.5 Number of accesses to off-chip memory

Figure 12 illustrates the number of accelerators accesses to off-chip memory in different schemes during different neural network model runs. The horizontal coordinate indicates the type of neural network, and the vertical coordinate indicates the number of accesses to off-chip memory. Since in both Base and MRSB schemes, no on-chip memory units are added and the PE uses the same metrics, the Base and MRSB schemes are essentially the same in terms of the numbers of off-chip memory accessed. In UPMP, we add a column of on-chip memory units, which can store a large number of parameters of the neural network model and some intermediate results, effectively reducing the number of accelerator accessing off-chip memory. URMP reduces the number of accesses to off-chip memory by an average of 69% compared to Base and MRSB.

7 Conclusion

NoC-based deep neural network accelerators can decouple computation and communication in neural networks due to their flexible computation and communication characteristics. The computation results of PE can be stored and transmit directly on-chip, reducing the access to off-chip memory. Since there are a large number of one-to-many communications in the network and the destination nodes of the communications are located adjacent to each other in the network, the adoption of a suitable multicast mechanism can significantly improve the performance of the

platform. To this end, we propose a path-based multicast mechanism, called URMP, which has excellent scalability and also effectively exploits the traffic characteristics of neural networks to reduce the packets transmitted in the network and reduce the transmission delay and the classification delay of the accelerator platform. Compared with unicast scheme, URMP reduces the classification delay by 33% on average, the average transmission delay of packets by 31% and the number of packets transmitted in the network by 70%. Therefore, we believe that URMP is an effective multicast mechanism for the DNN accelerators platform.

Authors' Contributions YO helped in funding acquisition. JW worked in writing—original draft and methodology. CS worked in writing—review and editing. QW helped in data curation. HL helped in validation.

Funding This study was funded by the National Natural Science Foundation of China (NSFC) research projects (Grant Number 61874157).

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval Our work does not involve human and animal research.

References

1. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778
2. Xu L, Ren J, Liu C, Jia J (2014) Deep convolutional neural network for image deconvolution. In: International Conference on Neural Information Processing Systems, pp 1790–1798
3. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
4. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv e-prints
5. Chen K, Ebrahimi M, Wang TY, Yang YC (2019) Noc-based dnn accelerator: a future design paradigm. In: the 13th IEEE/ACM International symposium
6. Lian X, Liu Z, Song Z, Dai J, Ji X (2019) High-performance fpga-based cnn accelerator with block-floating-point arithmetic. IEEE Trans Very Large Scale Integr VLSI Syst 27(99):1874–1885
7. Jeffers Jim, Reinders James, Sodani Avinash (2016) Knights Landing overview. Intel Xeon Phi processor high performance programming. Elsevier, pp 15–24. https://doi.org/10.1016/B978-0-12-809194-4.00002-8
8. Baji T (2018) Evolution of the gpu device widely used in ai and massive parallel processing. In: 2018 IEEE 2nd Electron devices technology and manufacturing conference (EDTM)
9. Wang Qiong, Li Ning, Shen Li, Wang Zhiying (2019) A statistic approach for power analysis of integrated GPU. Soft Comput 23(3):827–836. https://doi.org/10.1007/s00500-017-2786-1
10. Goossens KGW, Dielissen J, Radulescu A (2005) Aethereal network on chip: concepts, architectures, and implementations. IEEE Design Test Comput 22:414–421
11. Ascia G, Catania V, Jose J, Monteleone S, Palesi M, Patti D (2020) Improving inference latency and energy of network-on-chip based convolutional neural networks through weights

compression. 2020 IEEE International parallel and distributed processing symposium workshops (IPDPSW), pp 54–63
12. Xiao S, Guo Y, Liao W, Deng H, Luo Y, Zheng H, Wang J, Li C, Li G, Yu Z (2020) Neuronlink: an efficient chip-to-chip interconnect for large-scale neural network accelerators. IEEE Trans Very Large Scale Integr Syst VLSI 28:1966–1978
13. Krichene H, Philippe JM (2021) Analysis of on-chip communication properties in accelerator architectures for deep neural networks. 2021 15th IEEE/ACM International symposium on networks-on-chip (NOCS), pp 9–14
14. Daneshtalab M, Ebrahimi M, Mohammadi S, Afzali-Kusha A (2009) Low-distance path-based multicast routing algorithm for network-on-chips. IET Comput Digit Tech 3:430–442
15. Lin X, McKinley PK, Ni LM (1994) Deadlock-free multicast wormhole routing in 2-d mesh multicomputers. IEEE Trans Parallel Distrib Syst 5:793–804
16. Lu Z, Yin B, Jantsc A (2006) Connection-oriented multicasting in wormhole-switched networks on chip. IEEE Computer society annual symposium on emerging VLSI technologies and architectures (ISVLSI'06), p 6
17. Li Y, Wu M, Li W, Xue R, Fan D, Li D, Ji Y, Ye X (2020) An efficient multicast router using shared-buffer with packet merging for dataflow architecture. 2020 14th IEEE/ACM International symposium on networks-on-chip (NOCS), 1–8
18. Hu W, Lu Z, Jantsch A, Liu H (2011) Power-efficient tree-based multicast support for networks-on-chip. 16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011), pp 363–368
19. Merolla P, Arthur JV, Alvarez-Icaza R, Bussat J-M, Boahen KA (2014) A multicast tree router for multichip neuromorphic systems. IEEE Trans Circuits Syst I Regul Pap 61:820–833
20. Wang L, Liu L, Wang X, Han J, Deng C, Wei S (2020) Cdring: Reconfigurable ring architecture by exploiting cycle decomposition of torus topology. 2020 57th ACM/IEEE Design Automation Conference (DAC), pp 1–6
21. Holanda PC, Reinbrecht CRW, Bontorin G, Bandeira VV, Reis R (2016) Dhyana: a noc-based neural network hardware architecture. 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp 177–180
22. Liu J, Harkin J, Maguire LP, McDaid L, Wade JJ, Martin G (2016) Scalable networks-on-chip interconnected architecture for astrocyte-neuron networks. IEEE Trans Circuits Syst I Regul Pap 63:2290–2303
23. Liu X, Wen W, Qian X, Li HH, Chen Y (2018) Neu-noc: a high-efficient interconnection network for accelerated neuromorphic systems. 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), pp 141–146
24. Kwon H, Samajdar A, Krishna T (2017) Rethinking nocs for spatial neural network accelerators. 2017 Eleventh IEEE/ACM International symposium on networks-on-chip (NOCS), pp 1–8
25. Ouyang Y, Tang F, Hu C, Zhou W, Wang Q (2021) Mmnnn: a tree-based multicast mechanism for noc-based deep neural network accelerators. Microprocess Microsyst 85:104242
26. Catania V, Mineo A, Monteleone S, Palesi M, Patti D (2016) Cycle-accurate network on chip simulation with noxim. ACM Trans Model Comput Simul TOMACS 27:1–25
27. Chen K-CJ, Ebrahimi M, Wang T, Yang Y-C, Liao Y-H (2020) A noc-based simulator for design and evaluation of deep neural networks. Microprocess Microsyst 77:103145
28. Chen KCJ, Wang T (2018) Nn-noxim: High-level cycle-accurate noc-based neural networks simulator. 2018 11th International workshop on network on chip architectures (NoCArc), pp 1–5
29. Chen KCJ, Wang T, Yang YC (2019) Cycle-accurate noc-based convolutional neural network simulator. Proceedings of the International Conference on Omni-Layer Intelligent Systems