Time-Delay Neural Network Synthesis

For Fast Convolution Neuromorphic Processors

Fast Convolution Neural Network Synthesis

For Neuromorphic Processors

1. Introduction
2. Related work
3. Convolutional Neural Network (CNN)
4. Neuromorphic Architecture

The Neuromorphic Architecture has demonstrated the hardware capable of deploying the elements, called neurons and synapses, of brain-inspired computing architectures, and the architecture has shown low-power consumption[ref9].

1. Fast Convolutional Neural Network

We propose Fast CNN architecture with Consolidated Receptive Fields (CRFs) to calculate output pixel values from each partial convolution receptive field of CRFs consistently shown as Fig1.



Fig 1: Consolidated Receptive Field (CRF) which has two partial receptive fields in a row-wise direction. A CRF consists of overlapped partial receptive fields which normal CNN architecture has.

Since CRF is composed of partial receptive fields, the more partial receptive fields are set by hyper-parameter, the larger CRF is needed, but the faster processing speed is achieved.

* Pixel List Expanding(PLE) algorithm

PLE algorithms is to expand the list of pixel nodes as many as CRT needs for convolution process. This list of pixel nodes is expressed as matrix. Let be the number of partial receptive fields in a row-wise direction, and let be the number of partial receptive fields in a column-wise direction. (see Algorithm 1.)

|  |
| --- |
| **Algorithm 1. PLE algorithm** |
| Input: partial receptive field  1: the list of pixel nodes L, expressed as matrix.  2: expand column nodes of L, until the column of L,  3: expand row nodes of L, until the row of L, |

If CRF comprises partial receptive fields, the input nodes is expressed as matrix (see Figure 2 (a)).

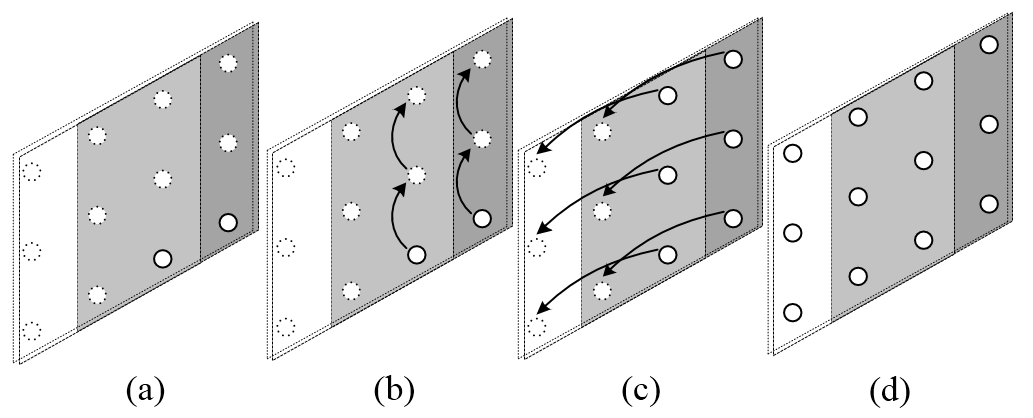


Fig 2: the case of CRF which has two partial receptive fields ( whose size is 3-by-3. By PLE algorithm, CRF is filled with pixel nodes which CRF needs to calculate output pixel values, see (a) to (d)

Note that this algorithm is able to adjust not only to convolution layer, but also max-pooling layer by adding the constraints of max-pooling layer.

1. Modified PLE algorithm: max-pooling

With adjusting to max-pooling layer, we present modified PLE algorithm by adding constraints of max-pooling; that considers stride and other constraints due to mismatching between input nodes and output nodes. In max-pooling window expressed as matrix, let be the number of partial receptive field in a row-wise direction and be the number of partial receptive field in a column-wise direction.

1. Synthesis
2. TDNN conversion

Converting CNN into TDNN has been introduced in [INsight]. When incoming data stream is fed into TDNN reconfigured from CNN, the input nodes receive continuous pixel values and it is necessary to hold pixel values to perform convolution process. In [Insight]’s neuromorphic chip, the receptive field comprises *delay units*, *synapses*, and *neurons* in a TDNN fashion. *Pixel-delay* unit holds pixel value for a clock cycle, and *delay lines* are composed of serialized *pixel-delay*sline by line. Each pixel values delayed by delay units *Synapse units* of the neuromorphic chip consists of multipliers and adder trees. In our scheme, we proposed following algorithm which deploys input nodes and expand them with delay-units to synthesize our Fast-CNN architecture on neuromorphic chip in a same way with [INsight].

Ff

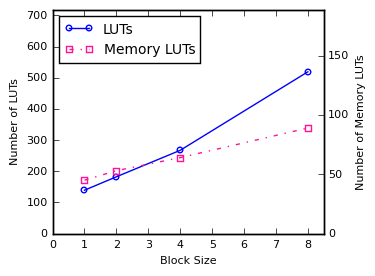
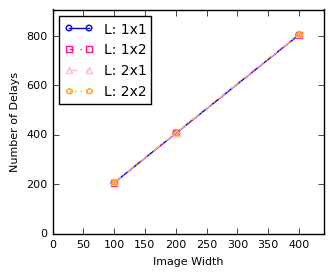
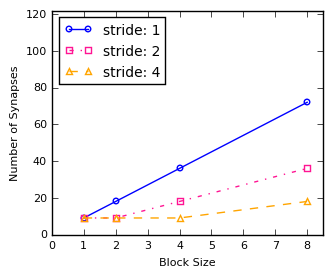
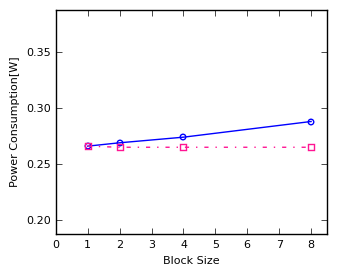
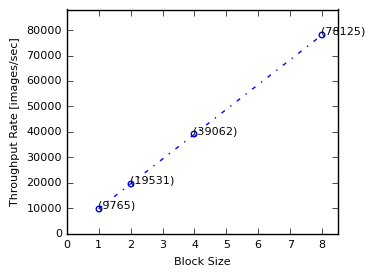
Ff

1. Resource requirement analysis

In convolution layer, required amount of delay units are expressed as

Let be requirements of pixel-delay, and let be requirements of delay lines. Note that total amounts of chained pixel-delays of are , due to decreasing the number of chained pixel-delays of delay lines along .

1. Experimental Results
2. Conclusion



(a)

(b)

(c)

(d)

(e)

Fig 3: The numbers of delay units ― pixel-delays and line-delays ― are shown in (a). Increasing of line-delays units are mainly affected by the width of image, regardless of the way that block size (L) would be chosen by. It shows