

提案(グラフプーリング
とグラフ畳み込みLSTM)

グラフ畳み込みLSTM

グラフ畳み込み(数式)

<https://arxiv.org/pdf/1801.07455.pdf>の(9)

使用するグラフ畳み込みLSTM

$$\mathbf{f} = \sigma(\Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{x}_t W_{xf} + \Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{h}_{t-1} W_{hf} + b_f)$$

$$\mathbf{i} = \sigma(\Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{x}_t W_{xi} + \Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{h}_{t-1} W_{hi} + b_i)$$

$$\mathbf{g} = \tanh(\Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{x}_t W_{xg} + \Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{h}_{t-1} W_{hg} + b_g)$$

$$\mathbf{o} = \sigma(\Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{x}_t W_{xo} + \Lambda^{\frac{-1}{2}} \mathbf{A} \circ M \Lambda^{\frac{-1}{2}} \mathbf{h}_{t-1} W_{ho} + b_o)$$

各式の演算をグラフ畳み込みとした。

グラフプーリング

おそらくグラフ上の特徴量(行列)は

$x(0)$	$y(0)$
$x(1)$	$y(1)$
$x(2)$	$y(2)$
$x(3)$	$y(3)$
$x(4)$	$y(4)$
$x(5)$	$y(5)$
$x(6)$	$y(6)$
$x(7)$	$y(7)$
$x(8)$	$y(8)$

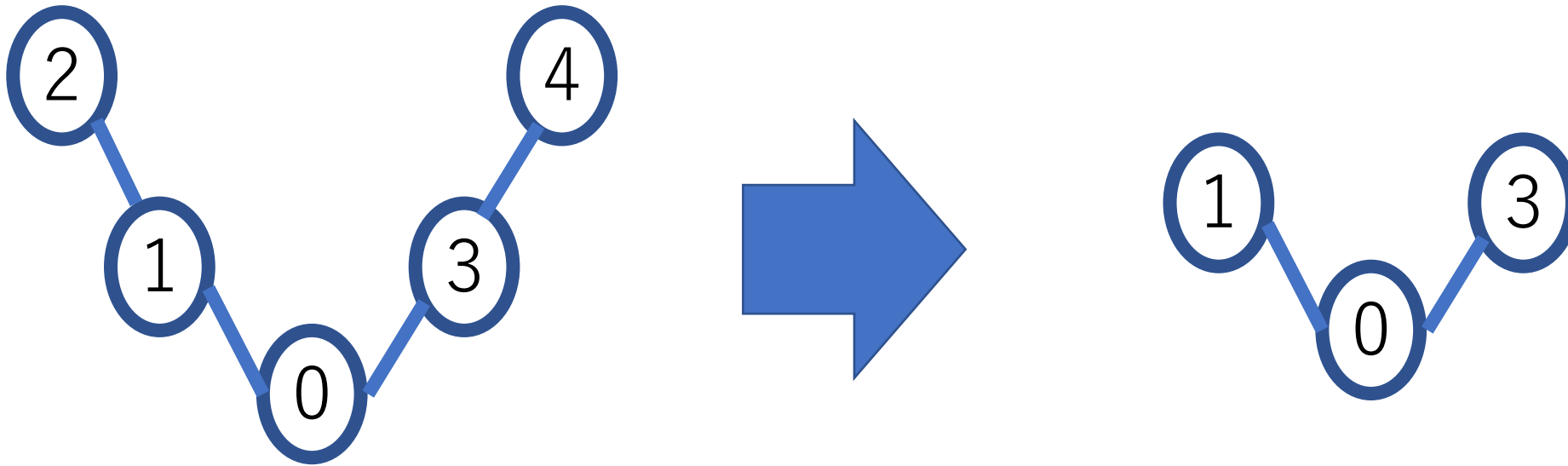
下のURLのチュートリアル of 僕なりの解釈
(<https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780>)

→

各列が画像でいうマップ
各行が各ノードの特徴ベクトル

提案手法(グラフプーリング)

メモリ(RAM, VRAM)の節約のために、
ノード数の削減を行う手法を提案する。



従来のプーリングについて

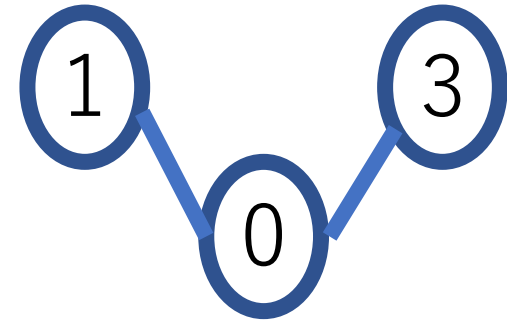
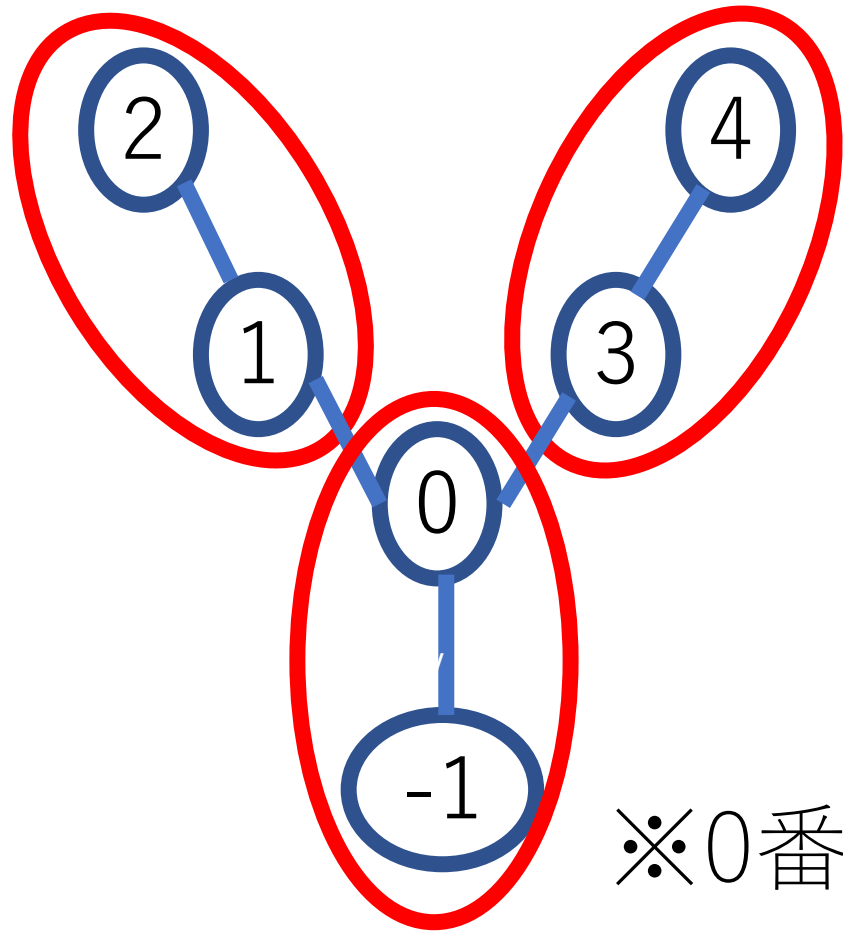
従来のプーリングは、画像のサイズを小さくするための演算

0	3	1	5
1	6	2	4
1	2	1	3
7	1	1	2



6	5
7	3

赤い丸の中で、プーリング



※0番のノードは、ダミーノードと囲む

先程のプーリングを愚直に実装すると

1つのマップにつきプーリングを実行する。

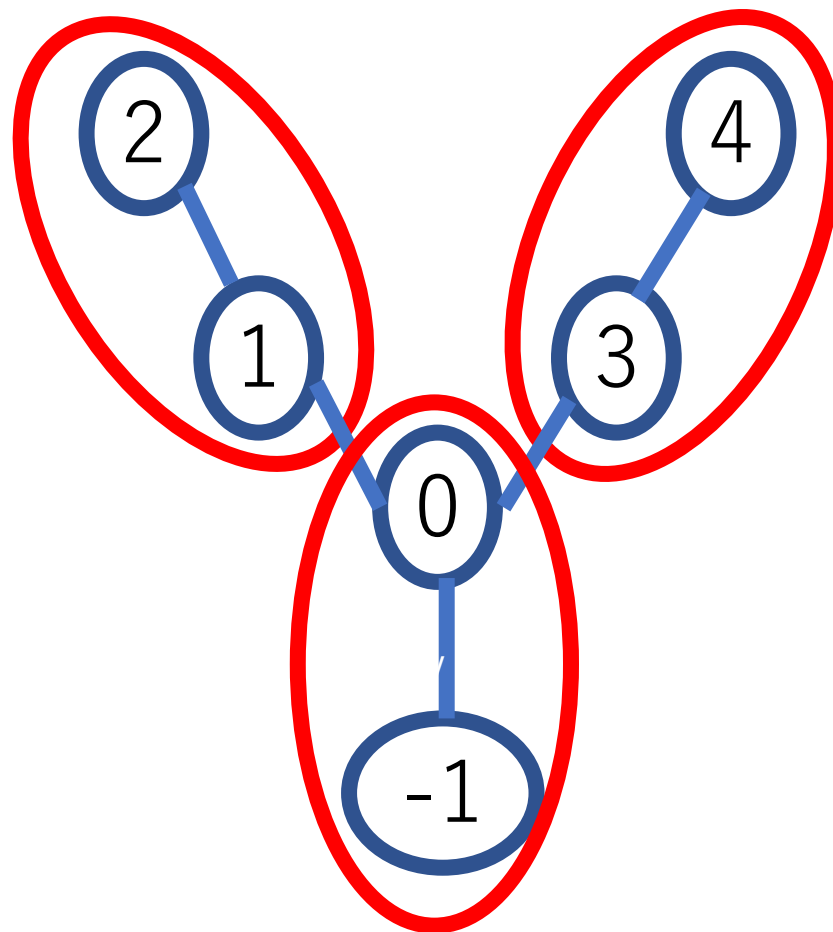
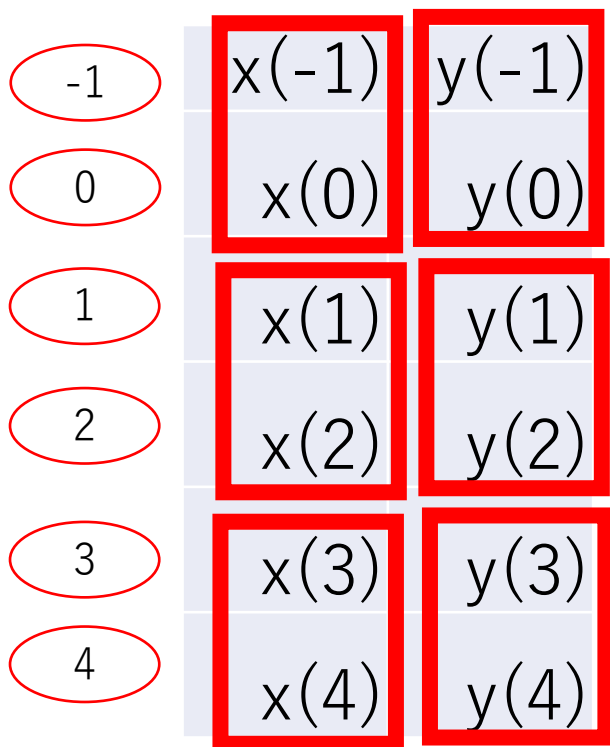
→マップの数 N に比例した計算がかかる。 $O(N)$

→従来のプーリングと同じように操作したい。
(既存のインターフェースで操作できるようにしたい)

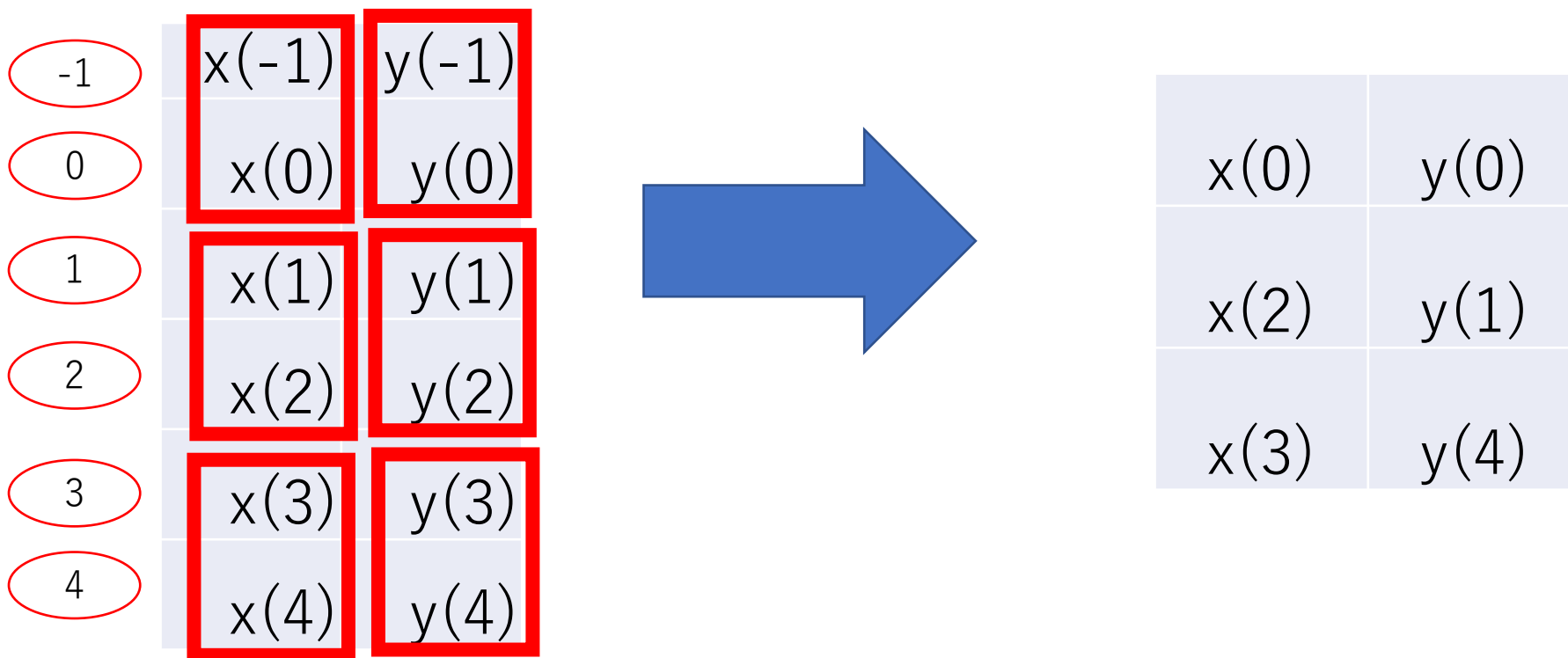
→ノード番号を適切に割り振れば、
従来のプーリングと同じ操作が可能

提案するプーリング法

2 × 1 の領域の最大値を取り出す



プーリング結果



メモリ (RAM, VRAM) の削減

