二値重み空間での Binary Neural Networksの学習

澁谷 辰吉[1], 井上中順[1], 川上 玲[1], 佐藤育郎[1,2]
[1]東京工業大学, [2]デンソーITラボラトリ

背景と本研究の貢献

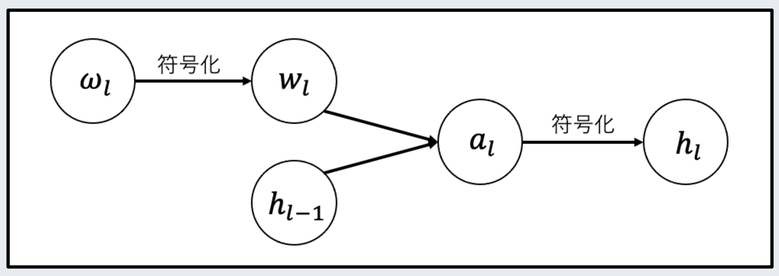
背景 Binary Neural Networksは推論時のメモリ削減に成功
➡ 一方，学習時は**実数重み**を持つため**メモリ消費大**
➡ 学習時メモリ削減はエッジAIのローカル学習に重要

貢献 **BNNsの二値重みを直接更新する学習手法を提案**
- MNISTでの実数重みなしBNNsの学習に初めて成功
- 学習時のメモリ消費量は解析的に最大33倍削減
- 従来の学習手法からの性能劣化は2.0%以内に留まる

従来手法 (実数重み空間での学習)

■ Binary Neural Networks (BNNs) [Courbariaux+, 2016]
- 活性化値$\boldsymbol{h}_l$および実数重み$\boldsymbol{\omega}_l$を符号化関数sgnで二値化

$$\boldsymbol{a}_l = \mathrm{sgn}(\boldsymbol{\omega}_l)\boldsymbol{h}_{l-1}$$
$$\boldsymbol{h}_l = \mathrm{sgn}(\boldsymbol{a}_l)$$



■ Straight Through Estimator (STE) **による勾配近似**
- 逆伝播時にsgn関数をHardtanh関数で近似し勾配消失を回避

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_l} \approx \frac{\partial \mathcal{L}}{\partial \boldsymbol{h}_l} \circ \mathbf{1}_{|\boldsymbol{a}_l| \le 1}, \qquad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\omega}_l} \approx \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_l}$$

■ **実数重み空間での勾配降下法**
- 時刻$t-1$での実数重み$\boldsymbol{\omega}_{t-1}$を目標値$\boldsymbol{\omega}_t^*$との内分点で更新

学習率$(0 < \eta < 1)$　　連続重みに関する勾配

$$\boldsymbol{\omega}_t = (1-\eta)\boldsymbol{\omega}_{t-1} + \eta\boldsymbol{\omega}_t^*, \qquad \boldsymbol{\omega}_t^* = \boldsymbol{\omega}_{t-1} - \boldsymbol{g}_t$$

$$\Leftrightarrow \quad \boldsymbol{\omega}_t \in \mathrm{argmin}_{\boldsymbol{\omega}\in\mathbb{R}^N}\left(\|\boldsymbol{\omega}_{t-1} - \boldsymbol{\omega}\|_2 + \|\boldsymbol{\omega} - \boldsymbol{\omega}_t^*\|_2\right)$$

提案：二値重み空間での学習

1. **勾配降下法の二値重み空間への拡張**
   - 二値重み$\boldsymbol{w}_{t-1}$を目標値$\boldsymbol{w}_t^* = \mathrm{sgn}(-\boldsymbol{g}_t)$との内分点で更新

ハミング距離（異なる成分の数）

$$\boldsymbol{w}_t \in \mathrm{argmin}_{\boldsymbol{w}\in\{-1,1\}^N}\left(d(\boldsymbol{w}_{t-1}, \boldsymbol{w}) + d(\boldsymbol{w}, \boldsymbol{w}_t^*)\right) \quad \Leftrightarrow$$

$$\boldsymbol{w}_t = (\mathbf{1} - \boldsymbol{m}_t) \odot \boldsymbol{w}_{t-1} + \boldsymbol{m}_t \odot \boldsymbol{w}_t^*, \qquad \boldsymbol{m}_t \in \{0,1\}^N$$

マスキングを行うベクトル（ハイパーマスク）

2. **ハイパーマスクの確率的サンプリング**
   - 現在の二値重みw_{t-1}と連続値の勾配g_tからマスクを確率的にサンプリング ➡ $\boldsymbol{m}_t \sim P(\boldsymbol{M}_t | \boldsymbol{w}_{t-1}, \boldsymbol{g}_t)$

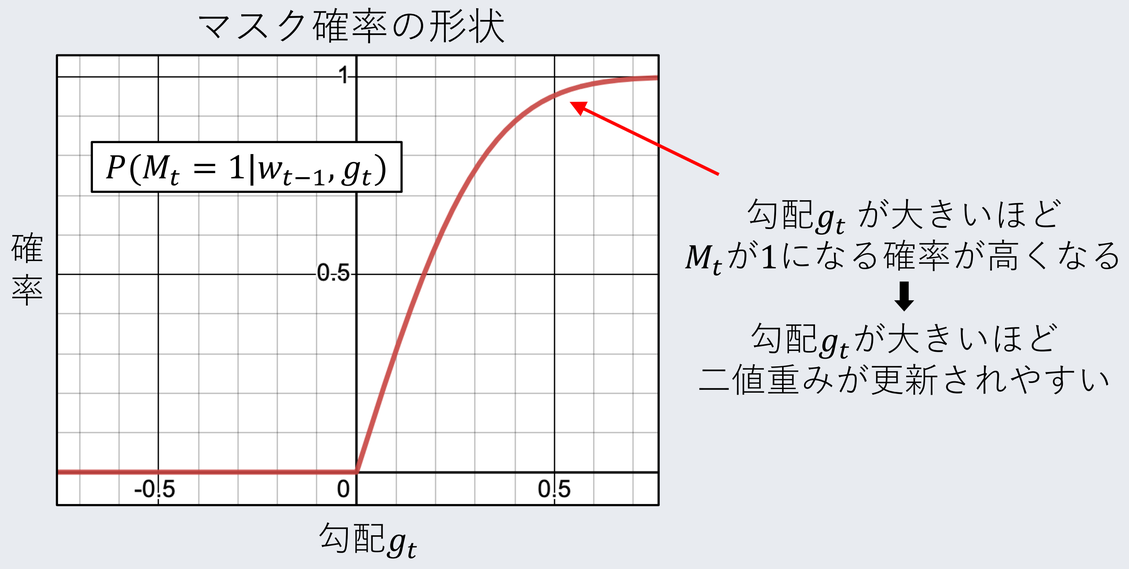
3. **ハイパーマスクの生成分布の定義**

$$P(M_t = 1 | w_{t-1}, g_t) = \begin{cases} \mathrm{erf}(\alpha_{t-1} g_t) - \mathrm{erf}(\min(\alpha_{t-1} g_t, 0)) & (w_{t-1} = 1) \\ \mathrm{erf}(\max(\alpha_{t-1} g_t, 0)) - \mathrm{erf}(\alpha_{t-1} g_t) & (w_{t-1} = -1) \end{cases}$$

$$\alpha_t = \frac{\eta}{\sqrt{2(\sigma_0^2 + \eta^2 \sum_{k=1}^{t} \sigma_{g_k}^2)}}, \quad \eta\text{: 学習率}, \quad \sigma_{g_k}^2\text{: 勾配}g_k\text{の標準偏差}$$

この分布による二値重みの更新は，**勾配g_tと実数重みω_{t-1}の独立性の仮定**のもとで**実数重み空間での更新と期待値的に一致**

マスク確率の形状



評価実験

実験１．汎化性能

モデル 4層の全結合型ニューラルネットワーク
中間層：256次元（Small）or 8192次元（Large）

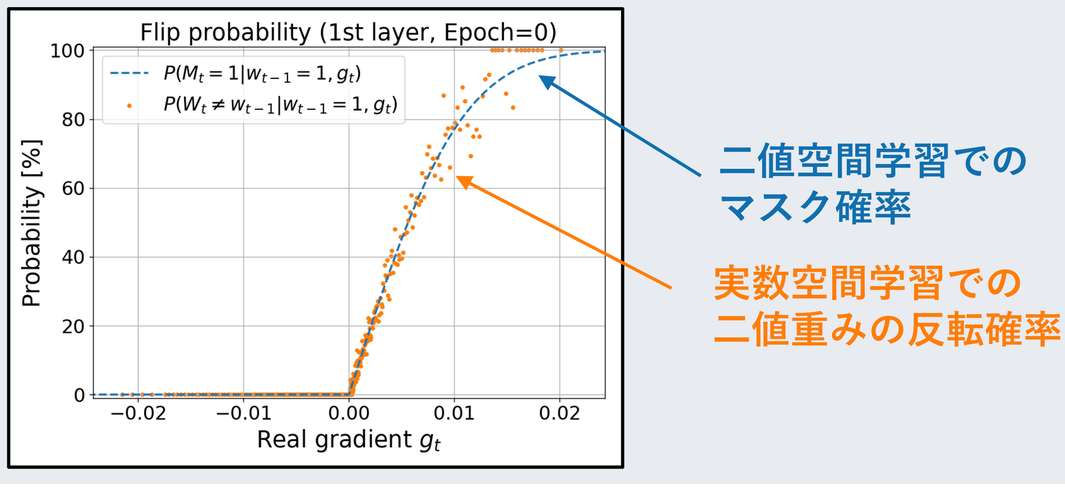
結果
- Small：実数空間での学習から20%弱の性能劣化
- Large：性能劣化は約2%に留まる

	学習時の重み空間	学習アルゴリズム	MNIST Small	MNIST Large
NNs	$\mathbb{R}^N$	SGD	$1.79_{\pm 0.10}$	$2.32_{\pm 0.04}$
BNNs	$\mathbb{R}^N$	SGD + STE	$4.79_{\pm 0.42}$	$3.63_{\pm 0.17}$
	$\mathbb{B}^N$	提案手法	$23.78_{\pm 1.04}$	$5.57_{\pm 0.22}$

実験２．ハイパーマスクの解析

設定 勾配g_tに対応する要素のマスク確率$P(M_t = 1 | w_t, g_t)$と実空間学習での離散重みの反転確率$P(W_t \neq w_t)$を比較

結果 マスク確率と反転確率はほぼ一致している
➡実数重みと関係なく，勾配大の二値重みを更新で学習可



本研究は東京工業大学「DENSO IT LAB 認識・学習アルゴリズム共同研究講座」における研究プロジェクト「機械学習の実応用に向けた品質基盤開発」の成果である.