エッジ AI に向けた無線環境における分散機械学習手法の開発

研究代表者　　杉　村　大　輔　　津田塾大学　学芸学部　情報科学科　准教授

1 はじめに

深層学習の利活用に関する検討が世界的に進められている．深層学習では，数百万を超えるパラメータにより構成される識別器を訓練することでビッグデータの特徴を解析する．これらのパラメータを適切に学習することで高精度な処理を実現できるが，そのためには大量かつ多様性に富んだ学習データの確保が重要である．深層学習応用に関する現行の議論の多くは，あらゆるモノやユーザのデータをクラウド等のサーバへ集約し，それらを解析する形態を前提としたものとなっている．このようなクラウドへのデータの集約を前提とした深層学習では，多くのユーザから大量のデータを回収する必要があるという性質上，以下のような二つの問題が考えられる：

・バックボーン回線の通信負荷増大，

・クラウドからのデータ流出に代表されるデータ提供者のプライバシー問題．

深層学習の産業応用への期待が広がる一方で，上記問題，特にプライバシー面での懸念が実用において障壁の一つとなっている．

これに対し近年，分散機械学習と呼ばれる技術が注目されてきている．各端末においてローカルな学習を行い，学習したモデルパラメータを他の複数の端末と共有および更新を行う．このような手順を繰り返すことにより，機械学習の学習速度向上および構築される識別器の識別性能向上を実現する．従来の機械学習では，計算機一台またはクラウド単独での実行を想定しており，確率的勾配降下法(SGD: Stochastic Gradient Descent)のような数理最適化手法を用いて識別器を構築する．これに対し分散機械学習では，並列・分散型の SGD [1], [2] に基づき，広域ネットワーク上に存在する複数の端末の計算能力を活用することにより，一台の計算機処理に比べ効率的な学習を行うことができる．また各端末の学習データの開示を必要としない．これにより，データ提供者のプライバシーを確保したビッグデータ解析の実現が期待できる．

分散機械学習に関する検討の多くは，Federated Learning [3] に代表されるように，集中制御サーバを備えるシステムを仮定している．一方，このようなサーバを用いず，近隣の端末同士が自律分散的に協調することで識別器を構築することも可能である．完全分散型にすることで，車車間ネットワークでの画像認識や災害環境における環境認識のような，固定インフラの仮定が難しい環境への深層学習応用の活発化が期待できる．このような背景から，本研究では完全分散型に着目した検討を行う(以降，これを分散機械学習と表記する)．

分散機械学習では，端末でのローカル機械学習とその学習結果（モデルパラメータ）の共有を繰り返す．しかしながら，深層学習により構築される識別器が有するモデルパラメータ数は数百万，数千万オーダーである．識別器の軽量化を目指し提案された MobileNetV2 [4] でも 300 万程度のパラメータを必要とし，その情報量は決して少なくない．そのため，無線環境で分散機械学習を実施する場合，距離減衰やフェージングといった通信路の影響に伴い，モデルパラメータ共有時の通信時間が増大する．これにより，学習全体に要する実行時間性能が極端に劣化する恐れがある．

そこで本研究では，無線環境での分散機械学習を高速化するための通信パラメータ設計法を検討する．具体的には，モデルパラメータ共有時の送信レート制御を許容した際，通信時間と学習性能にトレードオフが生じる点に着目する．帯域幅や送信電力が定数で与えられる場合，複数端末へ確実にモデルパラメータを共有するためには，低レートでの送信を行えばよい．これによりモデル共有ごとの学習精度改善の効率を確保できる．一方，通信時間が増大するため，実行時間の観点では学習精度特性が劣化することが考えられる．送信レートを上げればモデルパラメータ共有に要する通信時間を短縮できるが，この場合モデル共有が可能な範囲が狭まる．これにより，モデル共有ごとの学習精度特性は劣化する(図 1)．本研究では，これらの関係を踏まえ，分散機械学習の実行時間高速化手法を検討する．具体的には，ネットワーク密度に関する指標を制約条件とした通信時間最小化問題として定式化する．ネットワークトポロジに対する組合せ最適化および凸最適化に基づく解法により，各端末の送信レートの適応化を行う．これにより，無線環境における分散機械学習の実行時間性能の改善を実現する．

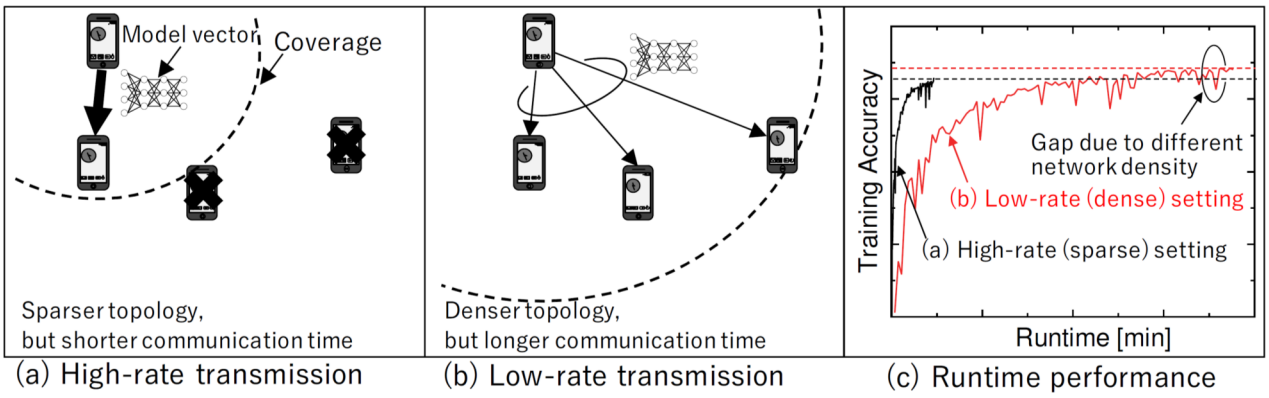

図 1　分散機械学習の学習特性に対する送信レートのトレードオフ：(a) 高レート送信（疎なネットワークトポロジ）；(b) 低レート送信（密なネットワークトポロジ）；(c) ネットワークトポロジの違いによる分散機械学習の実行時間特性の数値計算例.

提案手法による学習の実行時間特性は，深層学習を用いた画像認識タスクを対象とし，計算機シミュレーションにより評価する．距離減衰係数やデータのサンプル条件，制約とするネットワーク密度といったパラメータを変化させた評価を通して，無線環境における送信レート制御により学習を高速化できることを実験的に示す.

2 システムモデル

最初に，本研究で取り扱うシステムモデルについて説明する．通信路モデルを定義した後，分散機械学習における学習方法および通信プロトコルについて説明する.

2-1 通信路モデル

通信路モデルを定義する．通信距離d［m］における瞬時受信電力を$P(d) = P_{\mathrm{Tx}} G_A G_f \left(\frac{d}{d_0}\right)^{-\epsilon}$ [mW]と表す．ここで，P_{Tx}［mW］は送信電力，G_Aはアンテナ利得，G_fはマルチパスフェージングによる利得である．G_fは独立同分布（i.i.d.: independent and identically distributed)のレイリーフェージングに伴う期待値 1 の指数分布に従い，1 回の通信の間で一定とする．また，d_0［m］は参照距離，ϵは距離減衰係数である．送信電力P_{Tx}［mW］，帯域幅B［Hz］，および平均雑音電力N_0［mW/Hz］は端末間で一定であるとし，$P_{\mathrm{Tx}}, B, d_0, \epsilon$ およびN_0は端末間で既知とする．以上の仮定のもと，通信距離dにおける瞬時通信路容量$C(d)$［bps］は，他者からの信号干渉がない場合，$C(d) = B \log_2\left(1 + \frac{P(d)}{N_0 B}\right)$ [bps] と表すことができる．また，各端末は適応変調によって自身の送信レートを制御できるものとする．ある端末の送信レートをR［bps］とした際，$C(d) \geq R$であるとき，通信成功であるとする.

2-2 学習プロトコル

本研究で取り扱う無線環境における分散機械学習のプロトコルについて説明する．本研究では，協調確率的勾配降下法（C-SGD: Cooperative SGD）［2］に基づいた学習方法の設計を行う．C-SGD は，単独での SGD や並列 SGD，分散 SGD といった確率的勾配降下法に関する種々のアルゴリズムを統合した SGD アルゴリズムである．C-SGD は，分散機械学習におけるネットワークトポロジや 1 回のローカル学習における学習回数といったパラメータを一般化して表現可能であるため，扱うパラメータを調整することにより，集中制御サーバの有無によらず適用可能である.

学習を行う端末の台数をnとする．i番目の端末は学習データ数$|\mathcal{D}_i|$のデータセット$\mathcal{D}_i$，およびN次元のモデルパラメータベクトルを持つ．ここで，$\boldsymbol{x}_i$のデータサイズはM［bit］であり，端末間で一定である．加え

て，各端末が有するモデルパラメータベクトルは，同一の初期ベクトル$\boldsymbol{x}_0$で初期化されているものとする．

分散機械学習における目標は，相互のデータセットに対する損失関数が最小となるよう，各端末のモデルパラメータベクトルを最適化することである．この問題は次式のように定式化することができる:

$$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n} \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{\xi\sim\mathcal{U}(\mathcal{D}_i)}[F(\boldsymbol{x}_i;\xi)] \quad \cdots (1)$$

ここでξはあるデータサンプルであり，$\xi \sim \mathcal{U}(\mathcal{D}_i)$はデータセット$\mathcal{D}_i$から一様分布に従ってデータ 1 つをサンプリングする操作を表す．また，Fは損失関数を意味する．

C-SGD では，次の操作を損失関数の値が収束するまで繰り返すことで式(1)の問題を解く：(i) 端末ごと独立に SGD をτ回実行し，自身の識別器を更新する．(ii) 更新したモデルパラメータベクトルを，通信路を介して周辺端末と共有する．(iii) 周辺端末から受信したモデルと自身のモデルを合成し，最新のモデルパラメータベクトルとする．

分散型 SGD に関する議論(例えば [2], [5])では，学習におけるモデル合成の処理を平均化行列$\boldsymbol{W}$を用いて表現している（ここで，平均化行列$\boldsymbol{W}$は，条件$\boldsymbol{W}\mathbf{1}=\mathbf{1}$($\mathbf{1}$は，全ての要素が 1 である$n$次元ベクトルである)を満たすものとする)．先行研究によると，学習精度の理論的特性は$\boldsymbol{W}$の固有値を用いて評価できることが示されている．このような知見に基づき，送信レートの影響を評価するために，学習プロセスをネットワークトポロジの隣接行列および平均化行列によりモデル化する．

前述の通信路モデルに基づき，$\boldsymbol{W}$の各要素W_{ij}を次式で表す:

$$W_{ij}=\frac{A_{ij}}{\sum_{j=1}^{n}A_{ij}}, \qquad A_{ij}=\begin{cases}1 & if\ C_{ij}\geq R_i \text{ or } i=j\\ 0 & \text{(otherwise)}\end{cases} \quad \cdots (2)$$

ここで，R_iはi番目の端末の送信レート，ηは学習率である．$A_{ij}(\in\{0,1\})$はi番目の端末が送信した情報がj番目の端末で受信されたか否かを表す．さらに，各要素がA_{ij}で与えられる行列$\boldsymbol{A}$を隣接行列として定義する．C_{ij}はこれらの端末間の瞬時通信路容量である．

学習回数kが$k \bmod \tau = 0$を満たすとき，端末間でモデル共有を行う．平均化行列$\boldsymbol{W}$を用いると，$k+1$回目に得られるパラメータベクトルは，$\boldsymbol{X}_{\mathrm{k+1}}=\boldsymbol{W}(\boldsymbol{X}_{\mathrm{k}}-\eta\boldsymbol{G}_{\mathrm{k}})$とモデル化できる．ここで，$\boldsymbol{X}_k=\left(\boldsymbol{x}_{k,1},\boldsymbol{x}_{k,2},\cdots,\boldsymbol{x}_{k,n}\right)^{\mathsf{T}}$，$\boldsymbol{G}_{\mathrm{k}}=\left(\nabla F\left(\boldsymbol{x}_{k,1};\xi_{k,1}\right),\nabla F\left(\boldsymbol{x}_{k,2};\xi_{k,2}\right),\cdots,\nabla F\left(\boldsymbol{x}_{k,n};\xi_{k,n}\right)\right)^{\mathsf{T}}$である．

(1) モデルパラメータ共有時の通信プロトコル

モデルパラメータ共有時の通信プロトコルについて述べる．各端末は，互いの位置関係を予めビーコン信号および GPS によって共有済みであるものとし，かつ時間同期が可能であるとする．学習開始に先立ち，送信レートベクトル$\boldsymbol{R}=[R_1,R_2,\cdots,R_n]$を最適化する．

学習プロセス開始後，モデルパラメータを共有する通信フェーズにおいて，各端末は送信レートR_i[bps]で自身の最新のモデルパラメータベクトルを周辺端末へマルチキャストする．この際，一度にマルチキャストできる端末台数を 1 台とし，時分割に従って端末番号順にマルチキャストを行うものとする．パケット中のデータ情報のサイズMがヘッダ情報と比較して十分大きい場合，1 回の通信フェーズで全端末が通信を終えるまでに要する時間は$t_{\mathrm{com}}=\sum_{i=1}^{n}\left(\frac{M}{R_i}+\Delta t_{\mathrm{com}}\right)$ [s]と表すことができる．ここで，Δt_{com}は定数であり，各通信後の待機時間を表す．この値を適切に設定することで，時間同期のずれや上位レイヤの影響に起因する通信遅延といった実装上の誤差要因に対応することができる．なお，端末間で相互に通信開始及び終了時間の把握を可能とするため，フェージング等により通信が失敗した場合においても再送は行わないものとする．また ACK/NACK による受信結果のフィードバックも設けないものとする．

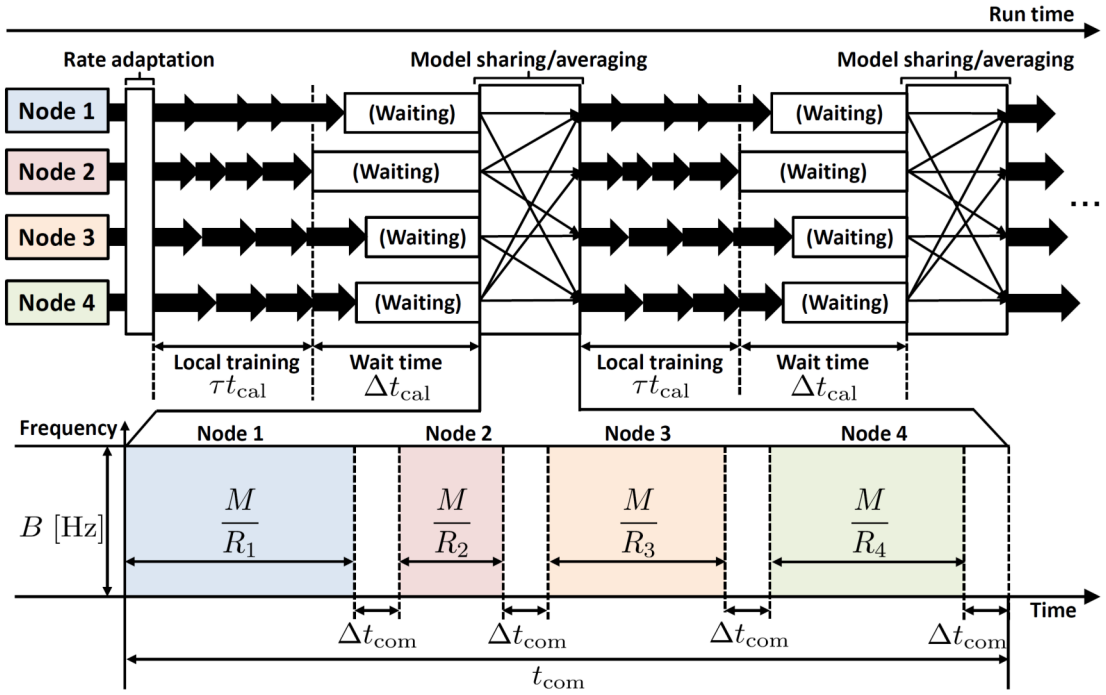

図 2　分散機械学習のプロトコル（$\tau = 4$，$n = 4$）．
各端末は対象となる無線環境に適応するように自身の送信レートを設定する．設定した送信レートに基づき，各ノードは，損失関数の値が最小になるまで次の二つの手順を繰り返す：(1) 端末ごとでモデルパラメータ更新（ローカル機械学習），(2) 更新されたモデルパラメータを端末間で共有・平均化．

(2) 学習全体のプロトコルおよび実行時間

上記システムモデルを踏まえた分散機械学習のプロトコルの例を図 2 に示す．なお，t_{cal}［s/loop］をローカル学習 1 ループに要する計算時間とする．実際は，計算時間は端末ごと，および実行回ごと不均一である．そのため，全ノードがτt_{cal}［s］で 1 フェーズ分のローカル学習を終えると仮定した場合，ネットワーク全体での同期にずれが生じる恐れがある．そこで，時間同期確保のため，計算時間に対しても待機時間に関する定数Δt_{cal}［s］を導入する．各端末はτループ分のローカル学習を行った後，1 回の学習フェーズの時間が$\tau t_{\mathrm{cal}} + \Delta t_{\mathrm{cal}}$となるよう待機する．これにより，端末間での時間動機のずれといった計算における実装上の不確定性にも対応することができる．これより，τ回の学習と 1 回の通信に要する時間は$t_{\mathrm{com}} + (\tau t_{\mathrm{cal}} + \Delta t_{\mathrm{cal}})$とモデル化する．

3 送信レートと学習精度の関係

本章では，無線環境における送信レートの違いが学習精度へ与える影響について議論する．C-SGD の学習繰り返し回数に対する勾配ベクトルの二乗ノルムの期待値の収束特性に関する理論結果が，Wang らにより示されている［2］．この結果は，本特性が平均化行列$\boldsymbol{W}$の固有値に依存することを理論的に示している．無線環境においては，送信レートによってネットワークトポロジの隣接行列が変化するため，本学習特性が送信レートに依存すると考えることができる．そこで，Wang らの解析結果を無線環境に適用することで，送信レートの学習特性への影響を観察する．

Wang らの解析では以下の仮定を置いている：
(1) $\left|\left|\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{y})\right|\right| \leq L\left|\left|\boldsymbol{x}-\boldsymbol{y}\right|\right|$（$L$：$F$のリプシッツ定数），
(2) $F(\boldsymbol{x}) \geq F_{\mathrm{inf}}$，
(3) $\mathrm{E}_{\xi|\boldsymbol{x}}[g(\boldsymbol{x})]=\nabla F(\boldsymbol{x})$（$g(\boldsymbol{x})$：$\boldsymbol{x}$の勾配），
(4) $\mathrm{E}_{\xi|\boldsymbol{x}}\left[\left|\left|g(\boldsymbol{x})-\nabla F(\boldsymbol{x})\right|\right|^2\right] \leq \beta\left|\left|\nabla F(\boldsymbol{x})\right|\right|^2+\sigma^2$（$\beta, \sigma^2$：ミニバッチサイズに反比例する非負の定数），
(5) $\lambda=\max\{|\lambda_2(\boldsymbol{W})|, |\lambda_n(\boldsymbol{W})|\}<\lambda_1(\boldsymbol{W})=1$（$\lambda_n(\boldsymbol{W})$：$\boldsymbol{W}$の固有値のうち$n$番目に大きな値），
(6) $\eta L+5\eta^2L^2\left(\frac{\tau}{1-\lambda}\right)^2 \leq 1$.
これらの仮定に加え，端末間でデータサイズは等しく，全てのデータは i. i. d. に従うものとする．
以上の条件のもと，各端末のモデルパラメータベクトルの初期値が同一のベクトルで与えられる場合，K回学習した後の勾配の二乗ノルムの期待値の上界は次式で与えられる：

$$\mathrm{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left|\left|\nabla F(\boldsymbol{X}_k)\right|\right|^2\right] \leq \underbrace{\frac{2[F(\boldsymbol{X}_1)-F_{\mathrm{inf}}]}{\eta K}+\frac{\eta L\sigma^2}{n}}_{\text{(1) fully synchronized SGD}}+\underbrace{\eta^2L^2\sigma^2\left(\frac{1+\lambda^2}{1-\lambda^2}\tau-1\right)}_{\text{(2) network error}} \quad \cdots (3)$$

この式は，勾配の二乗ノルム特性が，次の 2 つの要因の和に依存することを意味する：(1)全端末が相互に接続されており($\boldsymbol{W}=(\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}})/\ (\boldsymbol{1}^{\mathrm{T}}\boldsymbol{1})$)，かつ$\tau=1$であるSGDの特性，(2)ネットワークトポロジ起因の誤差特性．このうち要素(1)は，Kとnが十分大きい場合に 0 と近似できることから，達成可能な勾配の二乗ノルムの最小値はλに依存する．ここで，λは，全ての端末が相互に接続されている際に 0，接続が途切れるほど 1 に近づく性質を示すことに注意する．すなわち，λは，ネットワークトポロジの疎密を表していると考えることができる．このことから，式(3)に示される学習特性は，ネットワークトポロジの密度に依存しているといえる．
送信レートの影響を確認するため，無線環境において送信レートを調整した際のトポロジ密度と，式(3)より評価した学習特性の評価を行った．ここでは$n=6$の端末を 500m 四方のエリアにランダムに配置し，位置を固定した．送信レートは全端末一律でRとし，簡単のため$G_f=1$とした．その他のパラメータは，$G_A=1$，$P_{\mathrm{Tx}}=1$［mW］，$N_0=10^{-\frac{172}{10}}$［mW/Hz］，$B=1.4\times10^6$［Hz］，$\epsilon=4$とした．図 3 左図に，λに対する送信レートRの影響を示す．なお，λとネットワーク密度の関係を確認するため，$|\boldsymbol{A}|/\{n(n-1)\}$により求められるネットワーク密度も同時に評価した（$|\boldsymbol{A}|$は隣接行列$\boldsymbol{A}$中の全要素の総和である）．通信距離はRを大きくとるほど短くなる．そのため，ネットワーク密度はRに依存して小さくなる．一方で，λはRが大きくなるほど 1 に近づいていることがわかる．このことから，λの値を用いてネットワーク密度の低さを測ることができる．

次に学習特性に対する送信レートの影響を評価した．図 3 右図に，学習回数を$K=1,100$，$K\to\infty$と変化させた際の式(3)の数値結果をそれぞれ示す．この図において，y軸の値は図 3 左図に示した数値シミュレーション結果で得られたλを式(3)に代入することで得た．この数値結果より，Kが大きくなるほどネットワークトポロジの影響が大きくなることがわかる．例えば$K\to\infty$において，送信レートが大きくなるほど学習特性が劣化することが確認できる．しかしながら，その影響は顕著なものではなく，Rを 6Mbps から 3Mbps へと半減させてもy軸の値の変化は10^{-3}オーダー程度であり，非常に小さいことがわかる．一方で，Rを小さくとるほどモデル共有に要する通信時間が増大することが見て取れる．このことから，学習の実行時間特性が劣化する恐れがあると考える．

4 送信レート適応化による分散学習の高速化

前章での議論より，学習特性改善に対してネットワークトポロジを密に形成することは必ずしも効果的ではないことがわかった．一方で，送信レートは通信時間特性に直接影響を及ぼすため適切に設定することが重要である．これを解決するために，分散機械学習における送信レート設計を，ネットワーク密度を制約条

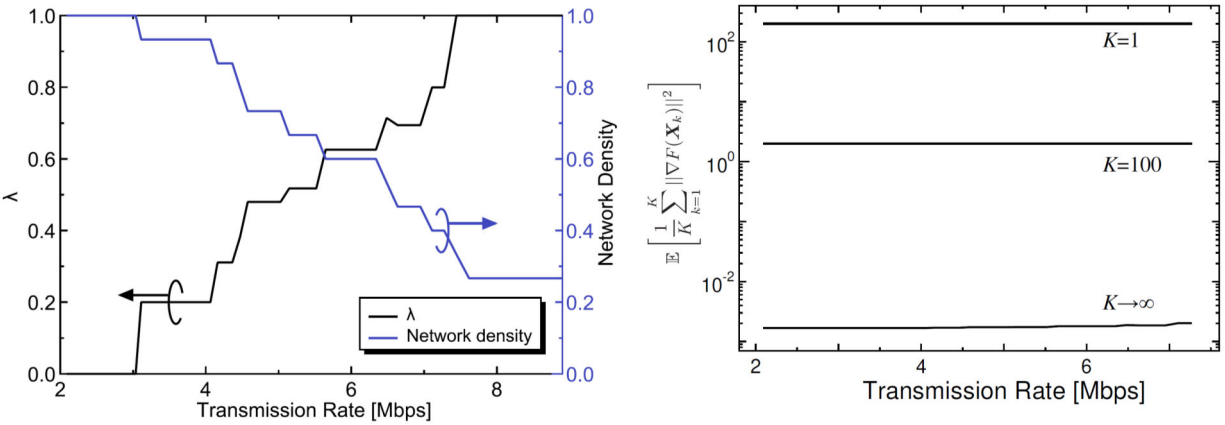

図 3　設定する送信レートの影響に関する数値シミュレーション結果例
（左図：ネットワーク密度（青線）とλ（黒線）に対する影響；右図：学習特性に対する影響）

件とした通信時間最小化問題として定式化する．これにより，学習特性を担保しつつ通信時間特性の向上を実現する．

4-1 最適化問題の定式化

前章で示したように，λの値からネットワーク密度の低さを測ることができる．加えて，λは隣接行列$\boldsymbol{A}$より一意に定まるため，各端末の位置関係および送信レートベクトルの候補から計算することができる．そこで，無線環境における分散機械学習のための送信レート最適化を次のように定式化する：

$$\min_{\boldsymbol{R}} \ t_{\mathrm{com}} \qquad \cdots (4a)$$

$$\text{s.t. } \Pr\left[\lambda \le \lambda_{\mathrm{target}} \middle| \bigcap_{i=1}^{n} S(G, v_i) = V\right] \ge 1 - p_{\mathrm{out}} \qquad \cdots (4b)$$

$$R_i \ge 0 \ \forall i \qquad \cdots (4c)$$

ここで，$\lambda_{\mathrm{target}}$は区間(0，1)の定数であり，達成可能な学習精度の所望値や学習条件によって定めるパラメータとする．また，$G = (V, E)$はネットワークトポロジを表現する有向グラフであり，$V(= \{v_1, v_2, \cdots, v_n\})$は端末の集合を表す．$E$は辺の集合であり，2 つの端末$v_i, v_j$が$C_{ij} \ge R_i$を満たすとき，$G$はこれらの端末間の辺を含む．また，$S(G, v_i)$は有向グラフ$G$において端末$v_i$からの有向路が存在する端末の集合を出力する操作を意味する．即ち，制約条件(4b)における$\bigcap_{i=1}^{n} S(G, v_i) = V$とは，対象のネットワークトポロジが強連結であり，分散機械学習時，各端末の学習結果が相互に行き渡ることを意味する．またp_{out}はトポロジの所望特性に対するアウテージ確率を意味する．フェージングによるトポロジ変動の影響を制限するため，本パラメータを導入する．

4-2 問題(4)の解法

制約条件(4b)を閉形式で表すことができないことから，問題(4)の解を一度に得ることは困難である．そこで，本問題を次の 2 ステップに分割して解く：(a)フェージングの影響を無視した上でのトポロジ設計，(b)設計したトポロジの確率的保護を制約とした通信時間最小化．ステップ(a)は端末間の通信路容量行列に基づく組み合わせ最適化，(b)は凸最適化による解を得ることができる．以下，各ステップの詳細について説明する．

(1) フェージングの影響を無視したトポロジ設計

ここでは$G_f = 1$とし，トポロジの確率的変動要素を除いた条件での最適なトポロジ設計を行う．この場合，問題(4)は次式のように表すことができる：

$$\min_{\boldsymbol{R}_{\text{mean}}} \sum_{i=1}^{n} \left(\frac{M}{R_i} + \Delta t_{\text{com}} \right) \quad \cdots (5a)$$

$$\text{s.t.} \qquad \lambda \leq \lambda_{\text{target}}, \quad \cdots (5b)$$

$$\bigcap_{i=1}^{n} S(G, v_i) = V, \quad \cdots (5c)$$

$$R_i \geq 0 \ \forall i \quad \cdots (5d)$$

ここで$\boldsymbol{R}_{\text{mean}} = [R_1, R_2, \cdots, R_n]$は$G_f = 1$のときの送信レートベクトルである．

この問題を解くために，$G_f = 1$のときの端末間の通信路容量行列$\boldsymbol{C}_{\text{mean}}$を定義する．この行列は，要素$C_{ij}$が端末$i$–$j$ 間の通信路容量であり，端末の位置関係に応じて一意に定まる．$G_f = 1$のときi番目の端末がj番目の端末へモデル共有を行うためには，$R_i = C_{ij}$とすればよい．このことから，$\boldsymbol{R}_{\text{mean}}$の候補は，各要素を$\boldsymbol{C}_{\text{mean}}$の各行の対角成分以外から 1 つを選ぶことで構成することができる．すなわち，問題(5)は最大候補ベクトル数が$(n-1)^n$となる組み合わせ最適化問題とみなすことができる．本研究では，$\boldsymbol{C}_{\text{mean}}$に対する全数探索により本問題を解き，最適な隣接行列$\boldsymbol{A}_{\text{target}}$を得る．

具体的な手順は以下の通りである．各候補ベクトルに対し隣接行列$\boldsymbol{A}$を計算し，それが上記制約条件を満たすかどうか評価する．条件を満たす場合，通信時間t_{com}を計算し，それが最小となる候補ベクトルを解とする．なお，グラフが強連結であるかの判定には深さ優先探索に基づいたアルゴリズム［10］を用いる．

(2) フェージングを考慮した通信時間最小化

次に，$\boldsymbol{A}_{\text{target}}$で得たトポロジ中の全辺の通信が同時に成功する確率が$1 - p_{\text{out}}$以上となることを制約に，通信時間最小化問題を解く．これにより得られた結果を最適な送信レートベクトルとする．この問題は次のように表すことができる：

$$\min_{\boldsymbol{R}} \sum_{i=1}^{n} \left(\frac{M}{R_i} + \Delta t_{\text{com}} \right) \quad \cdots (6a)$$

$$\text{s.t.} \quad \prod_{i=1}^{n} \prod_{l=1}^{n_i} p_S\left(R_i, \overline{\gamma}_{il}\right) \geq 1 - p_{\text{out}}, \quad \cdots (6b)$$

$$R_i \geq 0 \ \forall i \quad \cdots (6c)$$

ここでn_iはi番目の送信端末に接続している受信端末の台数であり，$n_i = \sum_{j=1}^{n} A_{ij} - 1$である（$A_{ij}$は$\boldsymbol{A}_{\text{target}}$中の各要素を意味する）．また，$\overline{\gamma}_{il}$は$i$番目の送信端末とそれに接続する$l$番目の受信端末間の単位周波数あたりの平均信号対雑音電力比(SNR: Signal-to-Noise power Ratio)であり，$\overline{\gamma}_{il} = \frac{P_{\text{Tx}} G_A}{N_0} \left(\frac{d_{il}}{d_0} \right)^{-\epsilon}$と表現することができる（$d_{il}$はこれら端末間の通信距離を意味する）．また，$p_S\left(R_i, \overline{\gamma}_{il}\right)$は，$i$番目の送信端末がレート$R_i$で送信した際，それに接続する$l$番目の受信端末が復調に成功する確率を意味する．周波数非選択性レイリーフェージング環境において，瞬時 SNR$\left(= G_f \overline{\gamma}\right)$は平均値$\overline{\gamma}$の指数分布に従う．この点に注意すると，$p_S\left(R_i, \overline{\gamma}_{il}\right)$は次式によ

り求められる：

$$p_S(R,\overline{\gamma}) = \Pr\left[R \le B\log_2\left(1+\frac{G_f\overline{\gamma}}{B}\right)\right] = \exp\left[\frac{B}{\overline{\gamma}}\left(1-\exp\left(\frac{R\ln 2}{B}\right)\right)\right] \quad \cdots(7)$$

これを制約条件(6b)に代入し，両辺の対数を取る．各制約条件の両辺を整理することで，問題(6)を次のように書き直すことができる：

$$\min_{\boldsymbol{R}} \sum_{i=1}^{n}\left(\frac{M}{R_i}+\Delta t_{\mathrm{com}}\right) \quad \cdots(8a)$$

$$\text{s.t.} \quad \ln(1-p_{\mathrm{out}}) - \sum_{i=1}^{n}\sum_{l=1}^{n_i}\left[\frac{B}{\overline{\gamma}_{il}}\left(1-\exp\left(\frac{R_i\ln 2}{B}\right)\right)\right] \le 0, \quad \cdots(8b)$$

$$-R_i \le 0 \ \ \forall i \quad \cdots(8c)$$

本問題は，$R_i > 0$において凸最適化問題となる．本研究では，本問題を凸最適化ライブラリである CVXPY 1.1.1 [6]を用いて解く．なお，ソルバは Splitting Conic Solver [7] を採用する．

5 実験

5-1 評価内容

画像認識タスクにおける計算機シミュレーションにより，提案手法の効果を検証した．本評価では，$n = 6$ の端末を 500 [m] 四方内にランダムに配置した際の，学習精度の実行時間特性を評価した．学習対象には Fashion-MNIST データセット [8] を用いた．これは，衣類に関する 10 種類の画像全 7 万枚を収録した画像認識ベンチマーク用のデータセットである（このうち学習データ 6 万枚，検証データ 1 万枚）．各画像は単一チャネルであり，画素数は28×28 [pixel] である．また，1 画素あたり 8 ビットの情報量を有する．

本実験では，学習データ全 60000 枚をn分割して各端末へ分配した．識別器には，文献 [9] で公開されている畳み込みニューラルネットワークを用いた．本識別器は，畳込み層とプーリング層のペア 2 層と 4 層の全結合層からなる．モデルパラメータの大きさは，32 ビット浮動小数点表現の場合$M = 47210816$ [bit] である．

学習の実行時間を評価するためには，端末上でのモデルパラメータベクトルの更新に要する計算時間と，モデル共有に有する通信時間の双方が必要である．そこで本実験では，計算時間は$t_{\mathrm{cal}} = 0.01$ [s/loop]，$\Delta t_{\mathrm{cal}} = 0$とし，通信時間は$\Delta t_{\mathrm{com}} = 0$とした．学習はバッチサイズ 1，$\tau = 1000$とした上で合計 200000 回繰り返した．学習 5000 回ごとに，検証データ 1 万枚を用いて各端末が所有するモデルの検証精度を評価した．得られた検証精度を端末間で平均化し，その値を，その時点での検証精度とした．その他のパラメータは，$P_{\mathrm{Tx}} = 1.0$，$G_A = 1.0, B = 1.4 \times 10^6$ [Hz], $N_0 = 10^{-\frac{-172}{10}}$ [mW/Hz]，$p_{\mathrm{out}} = 0.50$とした．

以上の評価を独立に 10 回試行し，その結果を繰り返し回数ごとに平均化することで，実行時間および学習回数ごとの検証精度の推移を検証した．

提案手法は Python 3.7.6 を用いて実装した．機械学習処理は PyTorch 1.5.1 を，送信レート最適化処理は凸最適化ライブラリ CVXPY 1.1.1 を用いた．使用した計算機の OS は Ubuntu 18.04 LTS である．

表 1　200000 回学習後の平均検証精度特性

	i.i.d.	non-i.i.d.		
		7 labels	5 labels	3 labels
$\lambda_{\text{target}} = 0.10$	0.910	0.885	0.857	0.718
$\lambda_{\text{target}} = 0.30$	0.905	0.871	0.823	0.656
$\lambda_{\text{target}} = 0.50$	0.905	0.866	0.806	0.622
$\lambda_{\text{target}} = 0.70$	0.904	0.862	0.796	0.609
$\lambda_{\text{target}} = 0.90$	0.900	0.845	0.764	0.543

5-2 評価結果

はじめに，学習データを重複のない無作為抽出(i.i.d.)に従って各端末へ分配した際の学習特性を評価した．

表 1 に，提案手法による学習を 200000 回繰り返した後の平均検証精度特性を示す．なお，この表には併せて後述の non-i.i.d.環境における特性もまとめた．表 1 より，学習回数の観点ではλ_{target}を小さく取り，ネットワークを密にするほど精度が得られることがわかる．

次に，$\epsilon = 4.0, 3.0$における実行時間特性を，それぞれ図 4(a)，(b)に示す．なお，参考のため次の 2 つも評価した：(i) 全端末が相互に接続されており，通信遅延がない場合(“Ideal cooperation”)，(ii) 全て端末が独立に学習する場合(“Individual training”)．学習回数特性と異なり，ある目標精度z(例えば 0.88)を達成するまでに要する実行時間は，λ_{target}を大きく取った方が短く済むことがある結果となった．これは，λ_{target}を小さく取った際，トポロジを密に取るために各端末の送信レートを下げ，通信時間が増大したためである．

一方，図 4(b)に示す$\epsilon = 3.0$における実行時間特性は，学習回数特性と同様，λ_{target}を小さくとるほど性能が良いことがわかる．これは，ネットワークトポロジを密に取る際の送信レートの低下の影響が大きくないためと考えられる．

次に，所望精度zの達成に要する実行時間特性を評価した．$\epsilon = 4.0$における特性を図 5(a)に，$\epsilon = 3.0$における特性を図 5(b)に示す．図 5(a)より，$\epsilon = 4.0$においては，ほぼ全てのzについて，λ_{target}を大きく取るほど学習を高速化できることがわかる．一方$\epsilon = 3.0$においては，図 4(b)と同様，λ_{target}を 0.1 から 0.3 程度と小さく取ることで学習を高速化できた．

以上より，適切なλ_{target}の値は距離減衰係数ϵに依存することがわかる．具体的には，$\epsilon = 4.0$のような減衰係数が比較的高い環境においては，達成可能な学習精度の多少の劣化を許容しても，λ_{target}を高く設定し通信時間短縮を図ることが有効である．一方，距離減衰係数が小さい場合は，λ_{target}を小さく設定し，学習回数あたりの精度改善の効率を向上させることが有効である．

端末間で学習データに偏りがある環境(non-i.i.d.)について議論する．第 3 章で記した Wang らの C-SGD の解析結果は，i.i.d.環境を想定したものである．そのため，non-i.i.d.環境において適切なλ_{target}の設計指針が異なる可能性があることに注意されたい．本評価において，各端末は全 10 ラベルのうち一部のラベル内の画像(例えば 3 ラベル)のみを学習に用いる．ここで，サンプリング対象とするラベルは端末ごと無作為に選択する．選択ラベルに対応する学習用画像から$60000/n$枚を無作為抽出し，それらをその端末が持つ学習データセットとする．

表 1 より，サンプリングラベル数が少なくデータの偏りが強くなるほど，達成精度が低下することが確認できる．また，同ラベル数内では，i.i.d.環境と同様にλ_{target}を小さく取るほど精度が改善されるが，その効果はデータの偏りが強いほど顕著であることがわかる．これは，周辺端末と協調することで未観測ラベルに対する知見を共有することができたため学習精度が改善したと考えられる．

次に，ラベル数 3 における$\epsilon = 4.0$および3.0での実行時間特性を，それぞれ図 6(a)(b)に示す．i.i.d.環境と異なり，いずれのϵにおいても，λ_{target}を小さく取り学習回数辺りの精度を改善するほど高速に学習できた．

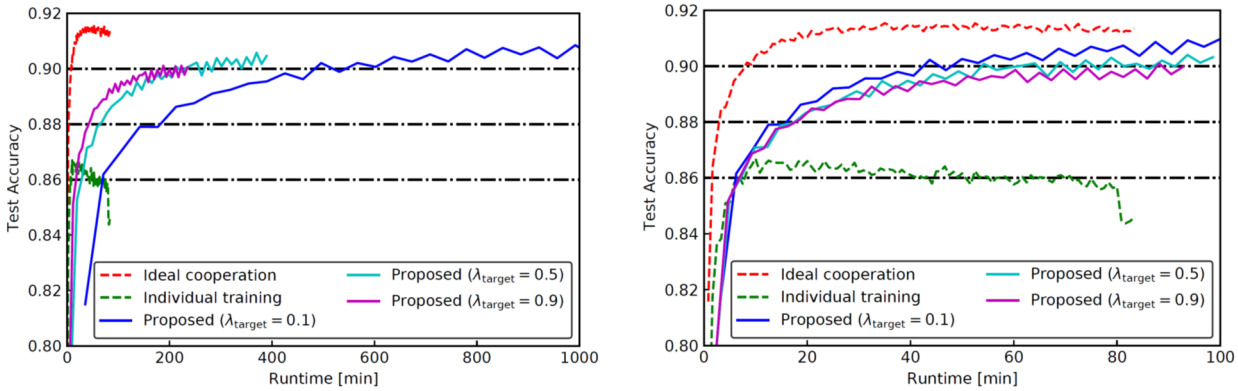

図 4　i. i. d. 環境における平均検証精度特性
（左図：(a) $\epsilon = 4.0$；右図：(b) $\epsilon = 3.0$）

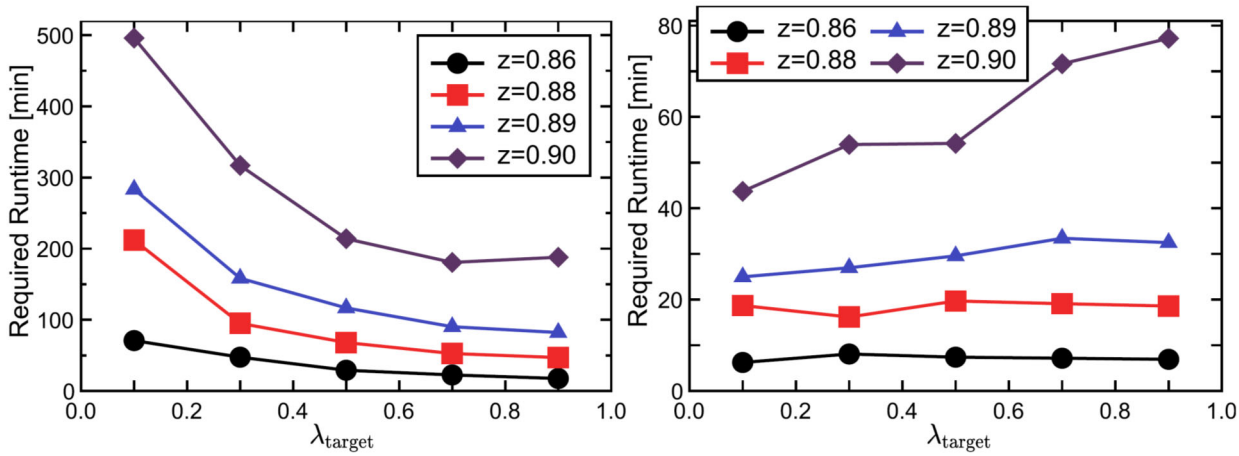

図 5　i. i. d. 環境におけるλ_{target}の影響
（左図：(a) $\epsilon = 4.0$；右図：(b) $\epsilon = 3.0$）

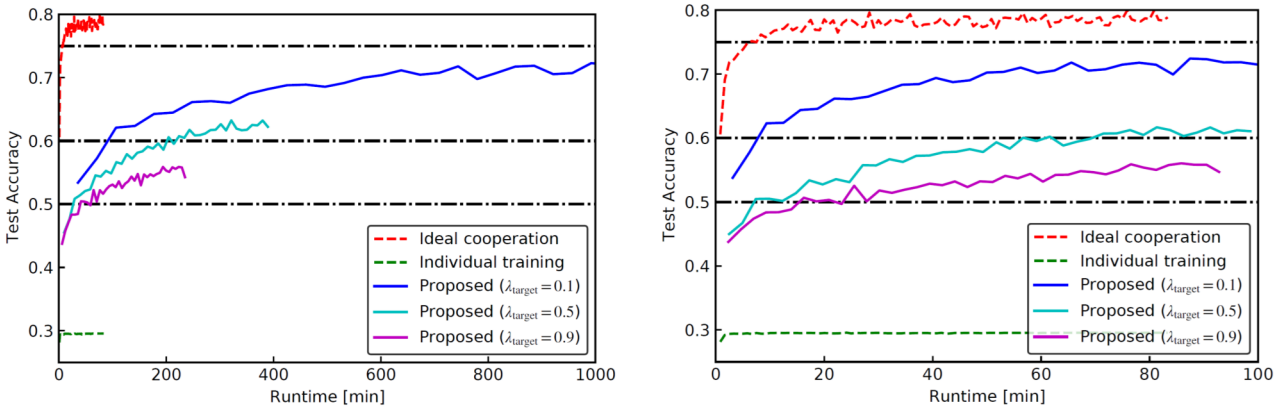

図 6　non-i. i. d. 環境における平均検証精度特性（サンプリングラベル数 3）
（左図：(a) $\epsilon = 4.0$；右図：(b) $\epsilon = 3.0$）

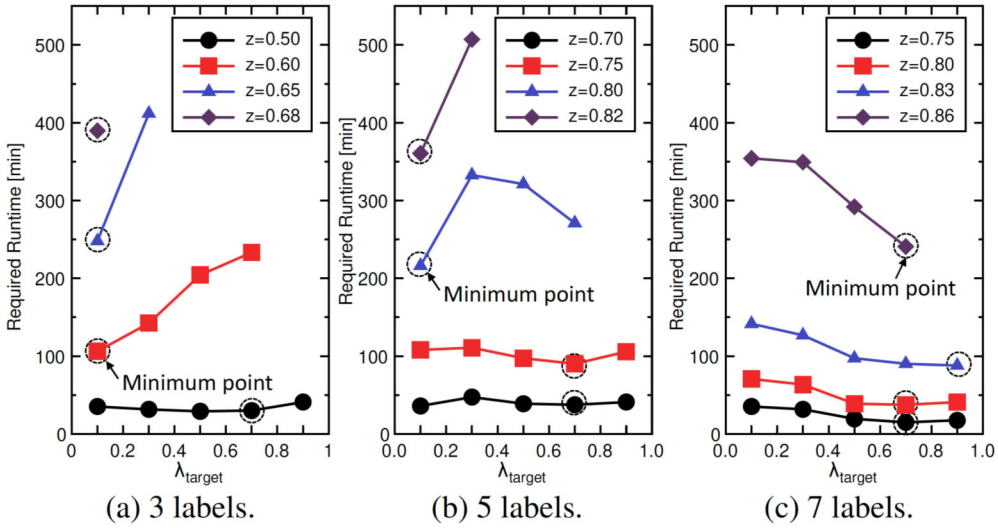

図 7　non-i. i. d. 環境におけるλ_{target}の影響（$\epsilon = 4.0$）

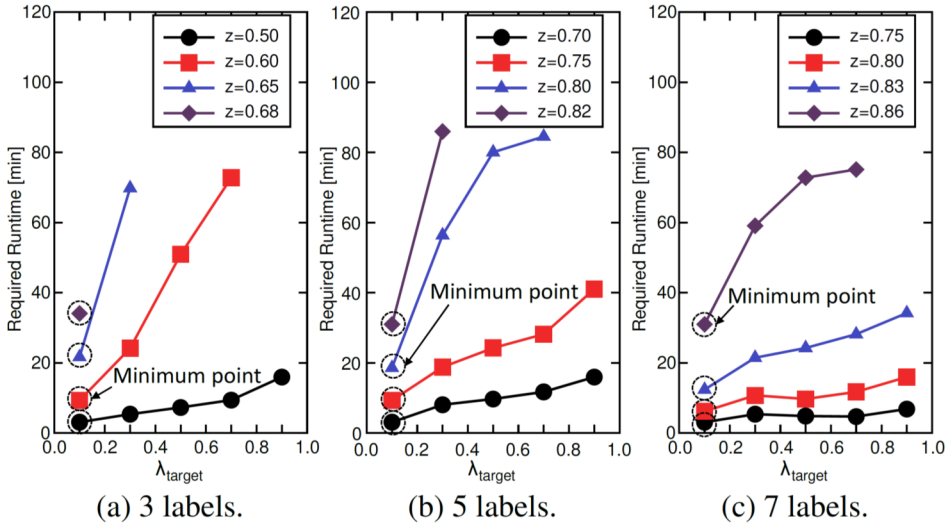

図 8　non-i. i. d. 環境におけるλ_{target}の影響（$\epsilon = 3.0$）

最後に，non-i. i. d. 環境におけるλ_{target}の設計指針を議論するため，所望精度z達成に要する実行時間を評価した．$\epsilon = 4.0$におけるλ_{target}の影響を図 7 に示す．ここでは，サンプリングラベル数を 3，5，7 とし，それぞれ図 7 (a) (b) (c) にまとめた．なお，ラベル数に応じて十分学習させた後に達成可能な精度が大きく異なることから，所望精度の大小はラベル数ごとに異なる．そのため，ラベル数ごとに異なるzを採用した．

zが低い場合，i. i. d. 環境と同様，λ_{target}を 0. 7-0. 9 程度とすることが望ましいことがわかる．一方，ラベル数が少なくなるほど，大きなλ_{target}では達成可能な精度が低く，所望精度を達成できない場合があった．そのため，データの偏りが高く所望精度が低くない場合，$\epsilon = 4.0$においてもλ_{target}を 0. 1 から 0. 3 程度に取ることが望ましいと考える．また，$\epsilon = 3.0$における評価結果を図 8 に記す．i. i. d. 環境と同様，λ_{target}を小さく取ることが望ましい結果となった．

以上の解析結果より，non-i. i. d. 環境においては，λ_{target}の設計指針は距離減衰係数に加え，サンプルの分布の偏りの強さにも依存することがわかった．偏りが弱い場合は，i. i. d. 環境と同様の設計指針でよいと考えられる．一方，偏りが強い場合は，距離減衰係数によらずλ_{target}を小さく設定し，学習回数辺りの学習精度

を向上させることで，高速な学習を実現することができる．

6 おわりに

無線環境における分散機械学習の高速化を目的とした送信レート制御の定式化およびその解法を示した．画像認識を対象とした計算機シミュレーションにより，本研究で新たに提案したネットワークトポロジの密度を制御するパラメータ$\lambda_{\mathrm{target}}$の設定が，学習時間特性に大きく影響を与えることを実験的に評価した．

評価実験を通じて，距離減衰係数や学習データの条件に応じて$\lambda_{\mathrm{target}}$を適切に設定し，ネットワーク密度に制約を与えた上での送信レート制御を行うことで，高速に所望の学習精度を達成できることを確認した．距離減衰係数が小さく遠方の端末とも比較的高速に通信できる場合や，non-i. i. d. 環境のような他端末との協調による繰り返し回数あたりの学習精度改善効果が高い環境では，$\lambda_{\mathrm{target}}$を小さい値に設定し，ネットワークトポロジを密に設計することが望ましい．一方，それ以外の環境，例えば i. i. d. 環境かつ距離減衰係数が大きい環境においては，$\lambda_{\mathrm{target}}$を大きい値に設定し，ネットワークトポロジを疎に設計する．これにより，繰り返し回数あたりの学習精度の多少の精度低下を許容しつつ通信時間を短縮した方が実行時間特性は改善される．

以上の解析結果より，無線環境での分散機械学習においては，ネットワーク密度に着目した通信設計が重要であるといえる．

【参考文献】

[1] M. A. Zinkevich *et al.*, “Parallelized stochastic gradient descent,” in *Proc. NeurIPS*, vol. 2, pp. 2595-2603. 2010.
[2] J. Wang and G. Joshi, “Cooperative SGD: A Unified Framework for the Design and Analysis of Communication-Efficient SGD Algorithms,” in *Proc. ICML Workshop*, 2019.
[3] B. McMahan *et al.*, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proc. AISTATS*, pp. 1273-1282, 2017.
[4] M. Sandler *et al.*, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proc. IEEE CVPR*, pp. 4510-4520, 2018.
[5] X. Lian *et al.*, “Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent,” in *Proc. NeurIPS*, pp. 5330-5340, 2018.
[6] S. Diamond and S. Boyd, “CVXPY: A Python-embedded Modeling Language for Convex Optimization,” *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1-5, 2016.
[7] B. O'Donoghue *et al.*, “Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding,” *J. Optimiz. Theory. App.*, vol. 169, no. 3, pp. 1042-1068, 2016.
[8] H. Xiao *et al.*, “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms,” in *arXiv*, 2017. [Online]. Available: https://arxiv.org/abs/1708.07747
[9] Pankaj, “Fashion MNIST with PyTorch (93% accuracy),” [Online]. Available: https://www.kaggle.com/pankajj/fashion-mnist-with-pytorch-93-accuracy
[10] R. Tarjan, “Depth-First Search and Linear Graph Algorithms,” in *Proc. SWAT*, pp. 114-121, 1971.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
Network-Density-Controlled Decentralized Parallel Stochastic Gradient Descent in Wireless Systems	IEEE International Conference on Communications	2020年6月
Rate-Adapted Decentralized Learning Over Wireless Networks	IEEE Transactions on Cognitive Communications and Networking	2021年4月（早期公開）
無線環境における送信レート適応化に基づく分散機械学習の高速化	電子情報通信学会無線通信研究会	2021年7月（発表予定）