データマージ：平均化データを用いた学習効率化

大竹 啓永 [1,2] 平岡 達也 [1,3] 宮尾 祐介 [2,4] 大関 洋平 [2,4] 磯沼 大 [2,5,6]
[1] 奈良先端科学技術大学院大学 [2] 国立情報学研究所 大規模言語モデル研究開発センター
[3]MBZUAI [4] 東京大学 [5] 東北大学 [6] 理化学研究所
otake.hiroto.od2@naist.ac.jp

概要

機械学習では一般に各データごとに損失を計算し、その平均を最小化するように学習を行うため、データ数の増加に伴う計算コストが課題となる。本研究では、事前に平均化されたデータを用いて学習を行うデータマージを検討する。データの平均化は計算量削減に有効である一方、入力のばらつきやそれに起因する損失の変動が失われ、性能低下を招く恐れがある。そこで、入力の分散とその分散に対する損失の感度を二次近似に基づいて明示的に考慮することで、平均データを用いながらも入力集合全体を反映した学習を試みる。VLM における敵対的学習の実験では、マージするデータ数が少ない場合には通常の学習と同等の頑健性を示す一方、データ数が増加すると頑健性が低下し、大規模データ適用に向けた課題が明らかとなった。

1 はじめに

近年の基盤モデルの成功は、大規模データセットを用いた事前学習に基づく大規模機械学習モデル [1] の発展によるところが大きい。基盤モデルは、事前学習時に含まれていなかった下流タスクに対しても高い転移性能を示している。一方で、これらのモデルは数億から数十億規模の画像やテキストデータを用いて学習されており [2, 3]、モデルとデータのいずれも急速に肥大化している [4]。データセットの増大に伴う学習コストの増加は、物理的な計算資源の不足や、学習完了までのサイクルの長期化を招き [5]、モデルの改良における試行錯誤の回数を制限し、モデル開発の大きな障壁となる。

本研究では、学習時の計算コストを削減しつつ、性能低下を抑制するデータマージを検討する。図 1 に示すように、データマージは複数のデータを平均化して得られる代表データを用いることで、損失計算の回数を削減し、学習効率を向上させるという単純な着想に基づいている。データの平均化は計算量削減の観点から有効である一方で、個々のデータが持つばらつきや、それに起因する損失の変動が失われる可能性があり、そのまま適用すると性能低下を招く恐れがある。本研究では、この問題に対処するため、平均データを用いた学習において失われる情報を補うように損失関数を改良する。平均データの周辺でデータがどの程度ばらついているかという統計的性質と、そのばらつきに対する損失の感度を二次近似に基づいて明示的に考慮することで、代表データのみを用いた単純化された損失を補正する。これにより、平均データを用いながらも、データ集合全体を反映した学習を可能にすることを目指す。

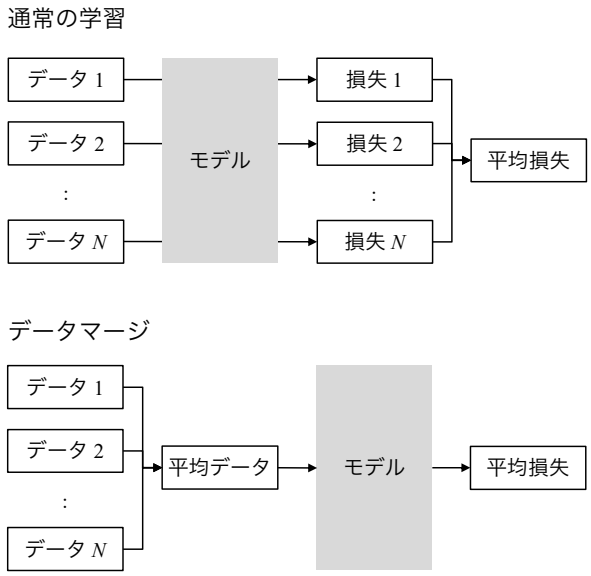

図 1 通常の学習では学習データごとに損失を計算する一方、データマージでは平均化した学習データを用いて損失を効率的に計算する。

まず予備実験として、多層パーセプトロン (MLP) を用いた単純な画像分類にて、データマージが適用可能なデータについて検討した。実験の結果、MNIST のようなクラス内のばらつきが比較的小さいデータセットでは、平均化によって有効な代表

データが得られる一方、CIFAR10 のようにばらつきが大きいデータには有効でないことを確認した。この結果は、データマージが、データ集合のばらつきが小さい条件下で有効であることを示唆している。

この知見に基づき、本研究では、クラス内のデータの分散が小さい設定として、CLIP における敵対的学習に着目する。敵対的学習では、元の画像に小さな摂動を加えた複数の画像を生成し、これらを学習に用いる。これらの画像のばらつきは小さいことから、データマージは敵対的学習に好適である。実験の結果、マージするデータ数が少ない場合、データマージは通常の学習と同等の頑健性を示す一方、マージするデータ数が増加するにつれて頑健性が低下することが確認され、大規模データへのデータマージ適用に向けた課題が明らかとなった。

2 データマージ

本研究では、バッチ内の全データに対する損失計算を効率化するため、入力データの平均 $\bar{\boldsymbol{x}}$ を用いた目的関数の近似手法データマージを提案する。モデルのパラメータを $\boldsymbol{\theta}$ とし、バッチサイズを N とする。バッチ内の i 番目の入力データを $\boldsymbol{x}_i$、対応する正解ラベルを全ての入力に対して共通の y とする。ここで、入力データの平均を $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$ と定義する。通常の学習における目的関数は以下の通りである。

$$\mathscr{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{\theta}, \boldsymbol{x}_i, y) \tag{1}$$

1 次近似による検討とその限界　まず、入力データ $\boldsymbol{x}_i$ とその平均 $\bar{\boldsymbol{x}}$ の距離が十分小さいと仮定し、損失関数 L を $\bar{\boldsymbol{x}}$ 周りで 1 次テイラー展開することを考える：

$$L(\boldsymbol{\theta}, \boldsymbol{x}_i, y) \approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) \tag{2}$$

式 (2) をバッチ全体で平均すると、$\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) = \boldsymbol{0}$ であることから、1 次の項は消失する：

$$\frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{\theta}, \boldsymbol{x}_i, y) \approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) \tag{3}$$

式 (3) は、1 次近似の範囲では、バッチ学習が単に入力平均 $\bar{\boldsymbol{x}}$ に対する学習と等価になることを示唆している。しかし、この近似ではデータのばらつき（分散）に関する情報が完全に失われており、各データ点 $\boldsymbol{x}_i$ の分布を考慮した学習が行えない。

2 次近似によるデータの分散の考慮　データのばらつきを損失関数に反映させるため、2 次の項まで考慮したテイラー展開を行う。

$$\begin{aligned} &L(\boldsymbol{\theta}, \boldsymbol{x}_i, y) \\ &\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}} L + \frac{1}{2}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}}^2 L (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \end{aligned} \tag{4}$$

ただし、表記の簡略化のため、$\nabla_{\boldsymbol{x}} L = \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y)$、$\nabla_{\boldsymbol{x}}^2 L = \nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y)$ とした。式 (4) をバッチ全体で平均すると、以下の近似式が得られる。

$$\begin{aligned} &\frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{\theta}, \boldsymbol{x}_i, y) \\ &\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + \frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}}^2 L (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \end{aligned} \tag{5}$$

ここで、第 2 項はデータの分散と損失関数の曲率（ヘッセ行列）との相互作用を表しており、これによりデータの分布を考慮した近似が可能になる。

計算効率化のための対角近似と半正定値化　パラメータ更新ごとに全データについて 2 次形式を計算し、かつヘッセ行列を保持することは計算コストの観点から現実的ではない。そこで、ヘッセ行列 $\nabla_{\boldsymbol{x}}^2 L$ の非対角成分を無視し、対角成分のみを用いる近似を行う。$\mathrm{diag}[\cdot]$ を対角成分をベクトルとして取り出す操作とすると、式 (5) は以下のように変形できる。

$$\begin{aligned} &L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + \frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}}^2 L (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \\ &\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + \frac{1}{2N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\circ 2\top} \mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L] \\ &= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y) + \frac{1}{2}(\overline{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)^\top \mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L] \end{aligned} \tag{6}$$

ここで、$(\cdot)^{\circ 2}$ は要素ごとの 2 乗を表し、$\overline{\boldsymbol{x}^2} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i^{\circ 2}$ は入力の 2 乗平均ベクトルである。すなわち、$(\overline{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)$ は入力データの要素ごとの分散に相当する。

入力に関するヘッセ行列 $\nabla_{\boldsymbol{x}}^2 L$ は必ずしも半正定値とは限らないため、損失が負の方向に発散するリスクがある。そこで、半正定値であることが保証される一般化ガウスニュートン行列でヘッセ行列を近似する。モデルのロジット出力を $\boldsymbol{t} = f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})$、損失関数 $S(y, \boldsymbol{t})$ をクロスエントロピー誤差とすると、連鎖律よりヘッセ行列は以下のように展開される。

$$\nabla_{\boldsymbol{x}}^2 L \approx \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{t}}^2 S(y, \boldsymbol{t}) \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \tag{7}$$

なお、モデルの 2 階微分を含む項 $\sum_i (\nabla_{\boldsymbol{x}}^2 f)_i (\nabla_{\boldsymbol{t}} S)_i$ は微小であると仮定し無視した。クロスエントロピー誤差の性質 $\nabla_{\boldsymbol{t}}^2 S(y, \boldsymbol{t}) = \mathbb{E}_{y' \sim \sigma(\boldsymbol{t})}[\nabla_{\boldsymbol{t}} S(y', \boldsymbol{t}) \nabla_{\boldsymbol{t}} S(y', \boldsymbol{t})^\top]$

を利用すると、データに関するヘッセ行列は勾配の自己相関行列の期待値として記述できる：

$$\nabla_{\boldsymbol{x}}^2 L \approx \mathbb{E}_{y'\sim\sigma(t)}[\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\bar{\boldsymbol{x}},y')\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\bar{\boldsymbol{x}},y')^\top] \quad (8)$$

上記式を、現在のモデル予測に基づくサンプリング $\hat{y}\sim\sigma(f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}))$ によって近似すると、ヘッセ行列の対角成分は勾配の 2 乗で置き換えられる。

$$\mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L] \approx \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\bar{\boldsymbol{x}},\hat{y})^{\circ 2} \quad (9)$$

最終的な目的関数　以上より、データマージで近似した目的関数は以下の通りとなる。

$$\tilde{\mathscr{L}}(\boldsymbol{\theta}) \approx L(\boldsymbol{\theta},\bar{\boldsymbol{x}},y) + \frac{1}{2}(\overline{\boldsymbol{x}^2}-\bar{\boldsymbol{x}}^2)^\top \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta},\bar{\boldsymbol{x}},\hat{y})^{\circ 2} \quad (10)$$

この目的関数を用いることで、バッチ平均 $\bar{\boldsymbol{x}}$ を用いつつ、データの分散 $(\overline{\boldsymbol{x}^2}-\bar{\boldsymbol{x}}^2)$ とモデルの感度（入力勾配の 2 乗）を考慮した学習が可能となる。$\overline{\boldsymbol{x}^2}-\bar{\boldsymbol{x}}^2$ は事前に計算できるため、パラメータ更新ごとに全データにアクセスする必要がなく、全データを学習する場合に比べて極めて計算コストは小さくなる。本節では、クロスエントロピー誤差を損失関数として用いる場合について説明したが、2 乗誤差についてもほぼ同様の議論にて近似できる（付録 A）。

3　予備実験

まず予備実験として、3 層のパーセプトロン（MLP）を用いた単純な画像分類にて、データマージが適用可能なデータについて検討した。実験には、MNIST[6] および CIFAR10[7] を使用し、10 クラスへの分類を試みた。データマージによる学習では、各クラスに属する画像を平均した代表画像 1 枚ずつを用いて学習を行った。これに対応するベースラインとして、各クラスからランダムに 1 枚ずつサンプルした画像を用いた学習を行い性能を比較することで、データマージの有効性を調べた。

実験結果を表 1 に示す。いずれのデータセットにおいても、各クラスの平均化画像を用いた場合の方が、ランダムに選択した単一画像を用いた場合よりも高い判別性能を示したものの、CIFAR10 は 10 クラス分類のチャンスレート（10%）とほぼ同等の性能に留まっている。図 2 に、各データセットにおけるクラスごとの平均化画像を示す。CIFAR10 では、同一クラス内でも画像間のばらつきが大きいため、平均化を行うと画像の細部が失われ、クラス間の違いが不明瞭になっていることが分かる。一方で、MNIST は手書き数字を対象としたデータセットであり、クラス内の構造が比較的一貫しているため、

表 1　MLP での CIFAR10、MNIST での全体の正解率。データマージにおいては、6000 枚の画像データの平均を用いている。

データ	データマージ	通常の学習（1 枚）
MNIST	65.18	50.70
CIFAR10	16.27	12.60

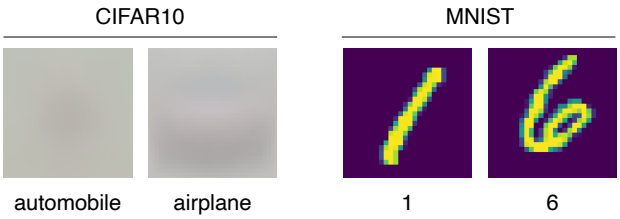

図 2　各クラスごとに平均化した画像データ。

平均化後の画像においても各クラスの特徴が保持されており、クラス間の違いを視覚的に確認することができる。これらの観察結果から、データマージの有効性は、元の画像データのばらつきの大きさに強く依存していることが示唆される。クラス内のばらつきが小さいデータに対しては、平均化によって有用な代表データが得られる一方で、ばらつきが大きいデータでは情報の欠落が生じやすい。以下の VLM への応用実験では、平均化する画像データのばらつきが比較的小さいデータへの適用を議論する。

4　VLM の敵対的学習への適用

本節では，データマージが VLM の敵対的学習にて有効に機能するかを検証する。敵対的学習は元データに対してモデルを撹乱する微小な摂動を加えたデータを用いて学習を行い、モデルの頑健性を向上させる方法である。予備実験ではクラス内のデータの分散が小さい場合に、データマージが比較的有効に機能することが示唆された。敵対的学習にて用いられる学習データのばらつきは小さく、データマージに好適であると考えられることから、先行研究 [8] での学習・評価設定を参考に、データマージを CLIP[2] の敵対的学習に適用した。

敵対的学習　本実験では ImageNet [9] 中の画像データに対し、10 ステップの PGD [10] によって摂動を加え、元の画像 1 枚に対して 10 枚の敵対的画像を生成した。敵対的学習に関する先行研究 FARE[8] と同様に、元の画像データと敵対的画像データを CLIP によりそれぞれエンコードした潜在表現について、その 2 乗誤差を最小化するように敵対的学習

表 2 VLM の敵対的学習における実験結果。元画像を用いた場合のスコアを **Clean**、敵対的画像を用いた場合のスコアのうち最も低い値を **Adversarial** として報告している。すべての評価指標は高い値ほど良い性能を表す。

	COCO		VQAv2	
	Clean	Adversarial	Clean	Adversarial
w/o AT	126.0	4.9	74.3	2.3
AT-1-Ind	112.8	26.0	67.9	27.6
AT-2-Ind	114.6	27.2	68.7	32.2
AT-5-Ind	119.0	24.0	68.4	29.3
AT-10-Ind	118.9	25.1	68.8	28.4
AT-2-Avg	119.5	17.9	69.4	25.9
AT-5-Avg	104.2	7.0	67.3	13.8
AT-10-Avg	115.2	8.7	69.3	13.8

を行う。

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} ||f_{\boldsymbol{\theta}}(x) - f_{\boldsymbol{\theta}}(x_i')||^2 \tag{11}$$

ただし、$f_{\boldsymbol{\theta}}$ は CLIP、x は元の画像データ、x_i' は i 番目の敵対的画像データである。本実験では、通常の目的関数 (11) にて学習した場合と、データマージにより近似した目的関数にて学習した場合を比較した。敵対的学習の詳細な設定は付録 B に記載した。

評価方法 LLaVA-1.5 7B[11] の CLIP を、敵対的学習を行った CLIP に置き換え、画像キャプション生成 (IC)、視覚的質問応答 (VQA) タスクにおける敵対的攻撃に対する頑健性を評価した。本研究では、IC タスクとして COCO[12] を、VQA タスクとして VQAv2[13] を用いた。ランダムにサンプリングした 100 枚の画像を用いて、COCO では CIDEr[14]、VQAvs では正解率で評価した。

実験結果 表 2 に評価実験の結果を示す。n 枚の敵対的画像を用いて通常の目的関数にて学習したモデルを **AT-n-Ind**、データマージで近似した目的関数にて学習したモデルを **AT-n-Avg**、敵対的学習を行っていないモデルを **w/o AT** で示した。ノイズを加えていない画像と、敵対的ノイズを加えられた画像の双方に対して評価を行い、それぞれを **Clean** および **Adversarial** として報告する。Clean 設定では、データマージで学習した場合（AT-n-Avg）は通常の目的関数で学習した場合（AT-n-Ind）に比べ遜色ない性能が得られた。敵対的画像はいずれも元画像との差分が小さいため、データマージで複数の敵対的画像を平均化したとしても、元画像に共通する特徴が残存した代表データが得られ、ノイズなし画像に対して高い性能を示したと示唆される。一方、Adversarial 設定では、2 枚の画像を平均化した場合（AT-2-Avg）は有効性を示すものの、それ以上の枚数の画像を平均化した場合（AT-5-Avg、AT-10-Avg）は通常の学習（AT-n-Ind）に比べ性能低下がみられた。これは平均化により、敵対的摂動の成分も弱められるためであると示唆される。マージするデータ数が少ない場合、データマージは通常の学習と同等の頑健性を示す一方、マージするデータ数が増加するにつれて性能が低下することが確認された。

5 関連研究

データセット蒸留 [15, 16, 17] は，大量の学習データを少量の合成データへと集約することで，学習の効率化を図る手法である。本稿で提案するデータマージは、データセット蒸留と異なり圧縮前データでの学習を予め必要とせず、より簡便にデータを集約できる。本研究と類似する研究として、異なるクラスの画像とラベルを混合する Mixup [18] や、画像の一部を切り取り重ね合わせる CutMix [19] といった手法が提案されているが、これらはいずれもデータオーグメンテーションを目的としている。MixTrain [20] では，これらのアイデアをデータ圧縮に応用しているが、単純なデータの線形結合は，データの分散情報を捨象してしまう。データマージは、損失関数の二次近似により分散情報を目的関数に組み込むことで、計算効率を維持しつつデータのばらつきを反映したより精緻な学習を実現する。

6 まとめ

本研究は、機械学習モデルの学習効率化を目的として、複数の学習データを平均化した平均データを用い、損失関数の二次近似により学習データのばらつきを考慮した学習手法を提案した。予備実験の結果から、学習データのばらつきが小さいタスクにおいては提案手法が比較的有効であることが確認された。VLM の敵対的学習での評価実験では、マージするデータ数が少ない場合、データマージは通常の学習と同等の頑健性を示す一方、マージするデータ数が増加するにつれて頑健性が低下することが確認され、大規模データへのデータマージ適用に向けた課題が明らかとなった。これらの結果は、マージする学習データのばらつきが小さければ、データマージが有効に機能することを示唆しており、今後はクラスタリングなどとの組合せなどを検討していく。

謝辞

本研究成果は，データ活用社会創成プラットフォーム mdx を利用して得られたものであり，JST BOOST JPMJBY24A6，JPMJBY24B2 の支援を受けたものです.

参考文献

[1] Rishi Bommasani, et al. On the opportunities and risks of foundation models, 2022.

[2] Alec Radford, et al. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, **Proceedings of the 38th International Conference on Machine Learning**, Vol. 139 of **Proceedings of Machine Learning Research**, pp. 8748–8763. PMLR, 18–24 Jul 2021.

[3] Christoph Schuhmann, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 25278–25294. Curran Associates, Inc., 2022.

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[5] Zanlin Ni, Yulin Wang, Jiangwei Yu, Haojun Jiang, Yue Cao, and Gao Huang. Deep incubation: Training large models by divide-and-conquering. In **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**, pp. 17335–17345, October 2023.

[6] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. **ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist**, Vol. 2, , 2010.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[8] Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. **ICML**, 2024.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In **2009 IEEE Conference on Computer Vision and Pattern Recognition**, pp. 248–255, 2009.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In **International Conference on Learning Representations**, 2018.

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 34892–34916. Curran Associates, Inc., 2023.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, **Computer Vision – ECCV 2014**, pp. 740–755, Cham, 2014. Springer International Publishing.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, July 2017.

[14] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 4566–4575, 2015.

[15] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2020.

[16] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In **International Conference on Learning Representations**, 2021.

[17] Asaf Shul, Eliahu Horwitz, and Yedid Hoshen. Distilling datasets into less than one image, 2024.

[18] H Zhang, M Cisse, Y Dauphin, and D Lopez-Paz. mixup: Beyond empirical risk management. In **6th Int. Conf. learning representations (ICLR)**, pp. 1–13, 2018.

[19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**, October 2019.

[20] Sarada Krithivasan, Sanchari Sen, Swagath Venkataramani, and Anand Raghunathan. Mixtrain: accelerating dnn training via input mixing. **Frontiers in Artificial Intelligence**, Vol. Volume 7 - 2024, , 2024.

[21] James Martens. New insights and perspectives on the natural gradient method. **Journal of Machine Learning Research**, Vol. 21, No. 146, pp. 1–76, 2020.

[22] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. **arXiv preprint arXiv:2305.14342**, 2023.

A 二乗誤差での平均化データへの損失関数の導出

モデルのパラメータを $\boldsymbol{\theta}$、ノイズなしデータを $\boldsymbol{x}$、i 番目のノイズ有りデータを $\boldsymbol{x}_i$ とする。バッチ $[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ について、入力データの平均を $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_i \boldsymbol{x}_i$ とする。モデル (CLIP) の出力を $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^K$、正解ラベル（ノイズなしデータのモデル出力）を $\boldsymbol{y} \in \mathbb{R}^K$ とする。$\boldsymbol{x}_i$ と $\bar{\boldsymbol{x}}$ 間の距離が小さいと仮定し、ある入力データ $\boldsymbol{x}_i$ に関する損失関数を 2 次近似すると

$$
\begin{aligned}
&L(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}) \\
&\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})
\end{aligned}
\tag{12}
$$

したがって、バッチ $[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ の損失関数は入力データの平均 $\bar{\boldsymbol{x}}$ を用いて、2 次近似で以下のように近似できる。ここで、計算の簡略化のために $\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y)$ の対角成分のみを用いる。

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N L(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}) \\
&= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) + \frac{1}{2N}\sum_{i=1}^N (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \\
&\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2N}\sum_{i=1}^N (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top \mathrm{diagonal}[\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})](\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \\
&= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) + \frac{1}{2N}\sum_{i=1}^N (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{2\top} \mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})] \\
&= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2N}\sum_{i=1}^N (\boldsymbol{x}_i^2 - 2\boldsymbol{x}_i\bar{\boldsymbol{x}} + \bar{\boldsymbol{x}}^2)^\top \mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})] \\
&= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) + \frac{1}{2}(\bar{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)^\top \mathrm{diag}[\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y})]
\end{aligned}
\tag{13}
$$

ただし $\bar{\boldsymbol{x}^2} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i^2$ は $\boldsymbol{x}_i$ の 2 乗平均である。ヘッセ行列 $\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y)$ は半正定値とは限らないため、損失関数が負になる可能性がある。そこでヘッセ行列 $\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, y)$ を半正定値行列である一般化ガウスニュートン行列（フィッシャー情報行列）で近似する [21, 22]。損失関数 L として二乗誤差 $S(\boldsymbol{y}, \boldsymbol{t}) = \frac{1}{2}||\boldsymbol{t} - \boldsymbol{y}||^2$ を用いる場合を考え、$\boldsymbol{t} = f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})$ をノイズ入りデータの平均 $\bar{\boldsymbol{x}}$ の潜在表現、$\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ をノイズなしデータ $\boldsymbol{x}$ の潜在表現とする。まずヘッセ行列は以下のように一般化ガウスニュートン行列で近似できる。

$$
\begin{aligned}
\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) &= \nabla_{\boldsymbol{x}} S(\boldsymbol{y}, f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})) \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{t}} S(\boldsymbol{y}, \boldsymbol{t})
\end{aligned}
\tag{14}
$$

$$
\begin{aligned}
\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) &= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{t}}^2 S(\boldsymbol{y}, \boldsymbol{t}) \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \\
&+ \sum_i \nabla_{\boldsymbol{x}}^2 [f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})]_i [\nabla_{\boldsymbol{t}} S(\boldsymbol{y}, \boldsymbol{t})]_i \\
&\approx \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{t}}^2 S(\boldsymbol{y}, \boldsymbol{t}) \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top
\end{aligned}
\tag{15}
$$

ただし $\nabla_{\boldsymbol{t}} S(\boldsymbol{y}, \boldsymbol{t}) = \boldsymbol{t} - \boldsymbol{y}$ より、ノイズ入りデータの平均の潜在表現とノイズなしデータの潜在表現の距離 $\boldsymbol{t} - \boldsymbol{y}$ が小さいと仮定する。$\nabla_{\boldsymbol{t}}^2 S(\boldsymbol{y}, \boldsymbol{t}) = I$ を利用。ここで、擬似ラベル $\hat{\boldsymbol{y}} = \boldsymbol{t} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ を導入すると、$\nabla_{\boldsymbol{t}} S(\hat{\boldsymbol{y}}, \boldsymbol{t}) = \boldsymbol{t} - \hat{\boldsymbol{y}} = \boldsymbol{\epsilon}$ より、

$$
\begin{aligned}
&\mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}})^\top] \\
&= \mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{t}} S(\hat{\boldsymbol{y}}, \boldsymbol{t}) \nabla_{\boldsymbol{t}} S(\hat{\boldsymbol{y}}, \boldsymbol{t})^\top \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top] \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{t}} S(\hat{\boldsymbol{y}}, \boldsymbol{t}) \nabla_{\boldsymbol{t}} S(\hat{\boldsymbol{y}}, \boldsymbol{t})^\top] \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \mathrm{E}_{\boldsymbol{\epsilon}}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \boldsymbol{I} \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top \\
&= \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}})^\top
\end{aligned}
\tag{16}
$$

まとめると、
$\nabla_{\boldsymbol{x}}^2 L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) = \mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}})^\top]$ より、

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N L(\boldsymbol{\theta}, \boldsymbol{x}_i, \boldsymbol{y}) \\
&\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2}(\bar{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)^\top \mathrm{diag}[\mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}})^\top]] \\
&= L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2}(\bar{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)^\top \mathrm{E}_{\hat{\boldsymbol{y}}}[\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \odot \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}})] \\
&\approx L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \boldsymbol{y}) \\
&+ \frac{1}{2}(\bar{\boldsymbol{x}^2} - \bar{\boldsymbol{x}}^2)^\top (\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \odot \nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \bar{\boldsymbol{x}}, \hat{\boldsymbol{y}}))
\end{aligned}
\tag{17}
$$

ただし $\hat{\boldsymbol{y}} = \boldsymbol{t} + \boldsymbol{\epsilon} = f_{\boldsymbol{\theta}}(\bar{\boldsymbol{x}}) + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

B 敵対的学習での詳細な設定

敵対的学習および敵対的データ生成におけるハイパーパラメータなどの設定を表 3 及び表 4 に示す。

表 3 敵対的学習の主要設定

項目	設定内容
最適化手法	AdamW
学習率	1×10^{-5}
Weight Decay	1×10^{-4}
バッチサイズ	1
学習ステップ数	5000（Warmup: 350）
損失関数	二乗誤差（L_2 loss）
出力正規化	無効（output_normalize=False）

表 4 敵対的データ生成の主要設定

項目	設定内容
最適化手法	PGD
ノルム制約	ℓ_∞
摂動強度 ε	2/255
PGD ステップ数	10
ステップサイズ	1
損失関数	二乗誤差（L_2 loss）