(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7529159号
(P7529159)

(45)発行日　令和6年8月6日(2024.8.6)

(24)登録日　令和6年7月29日(2024.7.29)

(51)国際特許分類

Ｇ０６Ｎ　20/00　(2019.01)	Ｇ０６Ｎ	20/00	１３０
Ｇ０６Ｎ　3/094(2023.01)	Ｇ０６Ｎ	3/094	

ＦＩ

請求項の数　5　（全12頁）

(21)出願番号　特願2023-528902(P2023-528902)
(86)(22)出願日　令和3年6月17日(2021.6.17)
(86)国際出願番号　PCT/JP2021/023123
(87)国際公開番号　WO2022/264387
(87)国際公開日　令和4年12月22日(2022.12.22)
審査請求日　令和5年10月24日(2023.10.24)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　110002147
弁理士法人酒井国際特許事務所
(72)発明者　山田　真徳
東京都千代田区大手町一丁目５番１号
日本電信電話株式会社内
審査官　渡辺　一帆

最終頁に続く

(54)【発明の名称】　学習装置、学習方法、および、学習プログラム

(57)【特許請求の範囲】

【請求項１】

Adversarial Exampleを含む入力データのラベルを予測するためのモデルの学習データを取得するデータ取得部と、

前記モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、前記モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、前記パラメータに対するloss landscapeを平らにしたloss関数と、前記Adversarial Exampleを含む学習データとを用いて、前記モデルの学習を行う学習部と

を備えることを特徴とする学習装置。

【請求項２】

前記学習部は、

前記学習データを用いて、前記loss関数により算出されるlossを最小化する前記モデルのパラメータを求める

ことを特徴とする請求項１に記載の学習装置。

【請求項３】

学習された前記モデルを用いて、入力データのラベルを予測する予測部

をさらに備えることを特徴とする請求項１に記載の学習装置。

【請求項４】

学習装置により実行される学習方法であって、

Adversarial Exampleを含む入力データのラベルを予測するためのモデルの学習デー

タを取得する工程と、
前記モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、前記モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、前記パラメータに対するloss landscapeを平らにしたloss関数と、前記Adversarial Exampleを含む学習データとを用いて、前記モデルの学習を行う工程と
を含むことを特徴とする学習方法。
【請求項５】
Adversarial Exampleを含む入力データのラベルを予測するためのモデルの学習データを取得する工程と、
前記モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、前記モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、前記パラメータに対するloss landscapeを平らにしたloss関数と、前記Adversarial Exampleを含む学習データとを用いて、前記モデルの学習を行う工程と
をコンピュータに実行させるための学習プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、モデルの学習装置、学習方法、および、学習プログラムに関する。
【背景技術】
【０００２】
従来、分類対象のデータにノイズをのせることで、分類器に誤判定をさせるAdversarial Exampleという攻撃がある。このAdversarial Exampleに対する対策として、例えば、Adversarial Exampleを用いてモデル（分類器）の学習を行うAdversarial Trainingがある。
【０００３】
しかし、Adversarial Trainingで学習したモデルは汎化性能が低いという問題がある。これは、Adversarial Trainingで学習したモデルのweightに対するloss landscape（loss関数の形）が尖っていることに起因する。そこで、上記のloss landscapeを平らにするため、モデルのlossを最大化する方向にweightにノイズ（摂動）を加える技術がある。
【先行技術文献】
【非特許文献】
【０００４】
【文献】Diederik P. Kingma, Max Welling, ”Auto-Encoding Variational Bayes”、[2021年6月4日検索]、インターネット＜URL:https://arxiv.org/pdf/1312.6114.pdf＞
【文献】Dongxian Wu, Shu-Tao Xia, Yisen Wang, ”Adversarial Weight Perturbation Helps Robust Generalization”、[2021年6月4日検索]、インターネット＜URL：https://arxiv.org/pdf/2004.05884＞
【発明の概要】
【発明が解決しようとする課題】
【０００５】
しかし、上記の技術は、ノイズがのっていないデータに対する予測性能が低下するという問題がある。そこで、本発明は、前記した問題を解決し、Adversarial exampleに対する頑健さを確保しつつ、ノイズがのっていないデータに対しても精度よく予測できるモデルを学習することを課題とする。
【０００６】
前記した課題を解決するため、本発明は、Adversarial Exampleを含む入力データのラベルを予測するためのモデルの学習データを取得するデータ取得部と、前記モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、前記モデルにおけるlos

s値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、前記パラメータに対するloss landscapeを平らにしたloss関数と、前記Adversarial Exampleを含む学習データとを用いて、前記モデルの学習を行う学習部とを備えることを特徴とする。

【発明の効果】

【０００７】

本発明によれば、Adversarial Exampleに対する頑健さを確保しつつ、ノイズがのっていないデータに対しても精度よく予測できるモデルを学習することができる。

【図面の簡単な説明】

【０００８】

【図１】図１は、学習装置の構成例を示す図である。

【図２】図２は、式（１０）におけるMAX vを求めるためには、Fisher情報行列Gの最大固有値λに対応する固有ベクトルhを求めればよい理由を説明するための式である。

【図３】図３は、学習装置の処理手順の例を示すフローチャートである。

【図４】図４は、学習装置の処理手順の例を示すフローチャートである。

【図５】図５は、学習装置の適用例を説明するための図である。

【図６】図６は、学習装置により学習されたモデルに対する実験結果を示す図である。

【図７】図７は、学習プログラムを実行するコンピュータの構成例を示す図である。

【発明を実施するための形態】

【０００９】

以下、図面を参照して、本発明の実施の形態（実施形態）を説明する。なお、本発明は以下に説明する実施形態に限定されない。

【００１０】

[学習装置の概要]

本実施形態の学習装置は、Adversarial Example（ノイズが付加されたデータ）を含むデータを用いて、入力されたデータのラベルを予測するモデルの学習を行う。ここで、学習装置は、モデルの学習に用いる損失関数（loss関数）として、モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、パラメータに対するloss landscapeを平らにしたloss関数を用いる。

【００１１】

これにより、学習装置は、Adversarial Exampleに対する頑健さを確保しつつ、ノイズがのっていないデータに対しても精度よくラベルを予測できるモデルを学習することができる。

【００１２】

[学習装置の構成例]

図１を用いて、学習装置１０の構成例を説明する。学習装置１０は、例えば、入力部１１、出力部１２、通信制御部１３、記憶部１４、および、制御部１５を備える。

【００１３】

入力部１１は、各種データの入力を受け付けるインタフェースである。例えば、入力部１１は、後述する学習処理および予測処理に用いるデータの入力を受け付ける。出力部１２は、各種データの出力を行うインタフェースである。例えば、出力部１２は、制御部１５により予測されたデータのラベルを出力する。

【００１４】

通信制御部１３は、ＮＩＣ（Network Interface Card）等で実現され、ネットワークを介したサーバ等の外部の装置と制御部１５との通信を制御する。例えば、通信制御部１３は、学習対象のデータを管理する管理装置等と制御部１５との通信を制御する。

【００１５】

記憶部１４は、ＲＡＭ（Random Access Memory）、フラッシュメモリ（Flash Memory）等の半導体メモリ素子、または、ハードディスク、光ディスク等の記憶装置によって実現され、後述する学習処理により学習されたモデルのパラメータ等が記憶される。

【００１６】
制御部１５は、例えば、ＣＰＵ（Central Processing Unit）等を用いて実現され、記憶部１４に記憶された処理プログラムを実行する。これにより、制御部１５は、図１に例示するように、取得部１５ａ、学習部１５ｂおよび予測部１５ｃとして機能する。
【００１７】
取得部１５ａは、後述する学習処理および予測処理に用いるデータを、入力部１１あるいは通信制御部１３を介して取得する。
【００１８】
学習部１５ｂは、Adversarial Exampleを含むデータを学習データとして用いて、入力されたデータのラベルを予測するモデルの学習を行う。ここで、学習部１５ｂは、モデルの学習に用いるloss関数として、モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加えて、パラメータに対するloss landscapeを平らにしたloss関数を用いる。
【００１９】
ここで、学習部１５ｂによる、モデルの学習方法の基本的な考え方を説明する。例えば、学習対象のモデルは、データxのラベルyの確率分布を表すモデルであり、パラメータθを用いて、式（１）により表される。なお、式（１）におけるｆは、モデルが出力するラベルを表すベクトルである。
【００２０】
【数１】

$$p_\theta\left(y_k|x\right)=\frac{\exp f_k(x;\theta)}{\sum_i \exp f_i(x;\theta)} \quad \cdots(1)$$

【００２１】
そして、学習部１５ｂは、式（２）で表されるloss関数の値が小さくなるように、モデルのパラメータθを決定することにより、モデルの学習を行う。ここで、p(y｜x)は、真の確率を表す。
【００２２】
【数２】

$$l\left(x,y;\theta\right)=p\left(y|x\right)\log p_\theta\left(y|x\right) \quad \cdots(2)$$

【００２３】
ここで、学習部１５ｂは、データxにノイズηが乗せられたAdversarial Example（式（３）参照）に対しても正しくラベルを予測できるようにモデルの学習を行う。つまり、学習部１５ｂは、式（４）に示すAdversarial Trainingを行う。
【００２４】
【数３】

$$\max_\eta E_{x,y\sim p(x,y)}\left[l\left(x+\eta,y;\theta\right)\right] \quad \cdots(3)$$

【数４】

$$\min_\theta\left(\max_\eta E_{x,y\sim p(x,y)}\left[l\left(x+\eta,y;\theta\right)\right]\right) \quad \cdots(4)$$

【００２５】
ここで、従来、上記のモデルのAdversarial Training（AT）において、モデルの汎化性能を上げるためにweight（モデルのパラメータ）にノイズ（摂動）を加えることにより、weightに対するloss landscapeを平らにする方法がある。この方法（Adversarial Weight Perturbation、AWP）における、loss関数は、式（５）および式（６）により表される。なお、w（weight）は、学習対象のモデルのパラメータであり、前記したθに相当する。αは、ノイズ（v）の大きさを調節する係数で、wのフロベニウスノルムではかったときのスケールに合うような値が設定される。つまり、パラメータはスケール不変性があるため、αはそのスケールの変化を吸収する役割を担う。

【００２６】
【数５】

$$\rho(\boldsymbol{w}) = \frac{1}{N}\sum_{n}^{N} \max_{\|\boldsymbol{x}_n' - \boldsymbol{x}_n\|_p \leq \varepsilon} l(\boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{w}) \quad \cdots (5)$$

【数６】

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{v}} \{\rho(\boldsymbol{w}) + \rho(\boldsymbol{w} + \boldsymbol{\alpha} \odot \boldsymbol{v}) - \rho(\boldsymbol{w})\} = \min_{\boldsymbol{w}} \max_{\boldsymbol{v}} \rho(\boldsymbol{w} + \boldsymbol{\alpha} \odot \boldsymbol{v}) \quad \cdots (6)$$

α:ノイズの大きさを調整する係数

w:モデルのパラメータ

v:モデルのパラメータに対するノイズ

⊙:アダマール積

【００２７】
ここで、filter normalizationで可視化されたweight loss landscapeを平らにしたいので、αは、フィルターごとのwのスケールにあったノイズ（摂動）になるよう以下の式（７）のように定義される。なお、kはフィルターのインデックスである。

【００２８】
【数７】

$$\boldsymbol{\alpha}^k = \frac{\left\|\boldsymbol{w}^k\right\|}{\left\|\nabla_{\boldsymbol{v}} \rho\left(\boldsymbol{w}^k + \boldsymbol{v}^k\right)\right\|} \quad \cdots (7)$$

【００２９】
よって、vを最大にするための更新式は、式（８）のように表される。

【００３０】
【数８】

$$\boldsymbol{v} \leftarrow \prod_{\gamma}\left(\boldsymbol{v} + \eta_2 \boldsymbol{\alpha} \odot \nabla_{\boldsymbol{v}} \frac{1}{N}\sum_{n} l\left(\boldsymbol{x}_n', \boldsymbol{y}_n, \boldsymbol{w} + \boldsymbol{v}\right)\right) \quad \cdots (8)$$

【００３１】
ちなみに先行研究では、上記のvを最大にするための更新は１回で充分であることが確認されている。また、wの更新式は、以下の式（９）のように表される。

【００３２】

【数９】

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{v} - \eta_3 \nabla_{\boldsymbol{v}+\boldsymbol{w}} \frac{1}{N} \sum_n l\left(\boldsymbol{x}'_n, \boldsymbol{y}_n, \boldsymbol{w} + \boldsymbol{v}\right) - \boldsymbol{v} \qquad \cdots (9)$$

【００３３】

ここで、上記のAWPでは、loss値を最大化するようにwにノイズを加えていたが、学習部１５ｂは、loss値のKLダイバージェンスを最大化するように、wにノイズを加える。このloss関数は、以下の式（１０）により表される。なお、式（１０）におけるρ(w)は、式（５）に示すρ(w)に相当する。

【００３４】

【数１０】

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{v}} \left\{ \rho\left(\boldsymbol{w}\right) + D_{\mathrm{KL}}\left(\rho\left(\boldsymbol{w}\right) || \rho\left(\boldsymbol{w} + \boldsymbol{\alpha} \odot \boldsymbol{v}\right)\right) \right\}$$

$$\boldsymbol{\alpha}^k = \frac{\left\|\boldsymbol{w}^k\right\|}{\left\|\boldsymbol{v}_{\mathrm{KL}}\right\|} = \left\|\boldsymbol{w}^k\right\| \qquad \cdots (10)$$

v_{KL}:KLダイバージェンスを最大化するよう作ったノイズ

【００３５】

式（１０）におけるMAX vを求めるためには、Fisher情報行列Gの最大固有値λに対応する固有ベクトルhを求めればよい。その理由となる式を図２に示す。

【００３６】

よって、vを最大にするための更新式は、式（１１）のようになる。

【００３７】

【数１１】

$$\boldsymbol{v} \leftarrow \prod_{\gamma} \left(\boldsymbol{v} + \eta_2 \boldsymbol{h}_1\right) \qquad \cdots (11)$$

【００３８】

なお、上記のFisher情報行列は巨大なので、負数に固有値分割を行うと時間がかかりすぎる。そのため、例えば、power iterationを使い最大固有値を計算する。また、Fisher情報行列を計算するときに、

【数１２】

$$\frac{\partial}{\partial w} \log p_\theta\left(y|x\right)$$

を計算する必要があるが、これは入力より出力の次元が大きいため、通常のDeep Learningで用いられるback propagationを使うと計算効率がよくない。そのためforward propagationで勾配を計算したいが、Pytorch等の既存の深層学習ライブラリにforward propagationのモードは用意されていない。そのため、以下の文献１に記載のROP trickを使ってforward propagationを実現する。

【００３９】

（文献１）[Adding functionality] Hessian and Fisher Information vector products、https://discuss.pytorch.org/t/adding-functionality-hessian-and-fisher-information-vector-products/23295/2

【００４０】

学習部１５ｂは、Adversarial Exampleを含む学習データと、上記のloss関数とを用いて、入力されたデータのラベルを予測するモデルの学習を行う。つまり、学習部１５ｂは、学習データを用いて、上記のloss関数により算出されるlossを最小化するようなモデルのパラメータθを求める。

【００４１】

予測部１５ｃは、学習されたモデルを用いて、入力データのラベルを予測する。例えば、予測部１５ｃは、学習されたパラメータθを上記式（１）に適用することより、新たに取得されたデータの各ラベルの確率を算出し、最も確率が高いラベルを出力する。これにより、学習装置１０は、例えば、入力データがAdversarial Exampleであった場合にも、正しいラベルを出力することができる。

【００４２】

[学習処理]

次に、図３を参照して、学習装置１０による学習処理手順の例について説明する。図３に示す処理は、例えば、学習処理の開始を指示する操作入力があったタイミングで開始される。

【００４３】

まず、取得部１５ａが、Adversarial Exampleを含む学習データを取得する（Ｓ１）。次に、学習部１５ｂが、学習データと、loss関数とを用いて、入力データのラベルの確率分布を表すモデルを学習する（Ｓ２）。なお、このloss関数は、上記の通り、モデルのパラメータにノイズを加えた場合とノイズを加えなかった場合とで、当該モデルにおけるloss値のKLダイバージェンスが最大になるようなノイズをパラメータに加え、パラメータに対するloss landscapeを平らにしたloss関数である。学習部１５ｂは、Ｓ２で学習されたモデルのパラメータを記憶部１４に記憶する。

【００４４】

[予測処理]

次に、図４を参照して、学習装置１０による入力データのラベルの予測処理の例について説明する。図４に示す処理は、例えば、予測処理の開始を指示する操作入力があったタイミングで開始される。

【００４５】

まず、取得部１５ａは、ラベルの予測対象のデータを取得する（Ｓ１１）。次に、予測部１５ｃは、学習部１５ｂにより学習されたモデルを用いて、Ｓ１１で取得されたデータのラベルを予測する（Ｓ１２）。例えば、予測部１５ｃは、学習されたパラメータθを上記の式（１）に適用することより、Ｓ１１で取得されたデータx’のp(x’)を算出し、最も確率が高いラベルを出力する。これにより、例えば、データx’がAdversarial Exampleであった場合でも、学習装置１０は、正しいラベルを出力することができる。

【００４６】

[学習装置の適用例]

上記の学習装置１０を、データの異常検知に適用してもよい。この場合の適用例を、図５を参照しながら説明する。ここでは、前記した予測部１５ｃの機能が、検知装置２０に装備される場合を例に説明する。

【００４７】

例えば、学習装置１０は、データ取得装置から取得した教師データ（学習データ）と、前記したloss関数とを用いて、モデルの学習（Adversarial Training）を行う。その後、検知装置２０は、データ取得装置から新たなデータx’を取得すると、学習済みモデルを用いて、データx’のp(x’)を算出する。そして、検知装置２０は、確率が最も高いラベルに基づき、データx’が異常なデータか否かのレポートを出力する。

【００４８】

[実験結果]

次に、本実施形態の学習装置１０により学習されたモデルによる、ラベルの予測精度の評価実験の結果を図６に示す。本実験では、本実施形態の学習装置１０により学習された

モデルについて、robust accとnatural accを評価した。
【００４９】
robust accは、Adversarial Exampleがのったデータの分類精度（データのラベルの予測精度）を示す値である。また、natural accは、ノイズなしのデータの分類精度を示す値である。robust accもnatural accも、0～100の値をとる。比較対象としたのは、ATにより学習したモデルと、AWPにより学習したモデルである。実験条件は以下の通りである。
【００５０】
画像のデータセット:Cifar10
Deep learning model: Resnet18
Adversarial Example: PGD
PGDのパラメータ: eps=8/255, train_iter=7, eval_iter=20, eps_iter=0.01, rand_init=True, clip_min=0.0, clip_max=1.0
【００５１】
図６に示すように、学習装置１０により学習されたモデルは、ATにより学習されたモデルと比べ、robust acc、natural accともに高い値になっている。また、本実施形態の学習装置１０により学習されたモデルは、AWPにより学習されたモデルと比べ、robust accは少し低い値となっているが、natural accは大幅に高い値になっている。
【００５２】
したがって、学習装置１０により学習されたモデルは、Adversarial exampleに対する頑健さを確保しつつ、ノイズがのっていないデータに対しても精度よく予測できるモデルであることが確認できた。
【００５３】
［システム構成等］
また、図示した各部の各構成要素は機能概念的なものであり、必ずしも物理的に図示のように構成されていることを要しない。すなわち、各装置の分散・統合の具体的形態は図示のものに限られず、その全部又は一部を、各種の負荷や使用状況等に応じて、任意の単位で機能的又は物理的に分散・統合して構成することができる。さらに、各装置にて行われる各処理機能は、その全部又は任意の一部が、ＣＰＵ及び当該ＣＰＵにて実行されるプログラムにて実現され、あるいは、ワイヤードロジックによるハードウェアとして実現され得る。
【００５４】
また、前記した実施形態において説明した処理のうち、自動的に行われるものとして説明した処理の全部又は一部を手動的に行うこともでき、あるいは、手動的に行われるものとして説明した処理の全部又は一部を公知の方法で自動的に行うこともできる。この他、上記文書中や図面中で示した処理手順、制御手順、具体的名称、各種のデータやパラメータを含む情報については、特記する場合を除いて任意に変更することができる。
【００５５】
［プログラム］
前記した学習装置１０は、パッケージソフトウェアやオンラインソフトウェアとしてプログラムを所望のコンピュータにインストールさせることによって実装できる。例えば、上記のプログラムを情報処理装置に実行させることにより、情報処理装置を学習装置１０として機能させることができる。ここで言う情報処理装置には、デスクトップ型又はノート型のパーソナルコンピュータが含まれる。また、その他にも、情報処理装置にはスマートフォン、携帯電話機やPHS（Personal Handyphone System）等の移動体通信端末、さらには、PDA（Personal Digital Assistant）等の端末等がその範疇に含まれる。
【００５６】
また、学習装置１０は、ユーザが使用する端末装置をクライアントとし、当該クライアントに上記の処理に関するサービスを提供するサーバ装置として実装することもできる。この場合、サーバ装置は、Webサーバとして実装することとしてもよいし、アウトソーシ

ングによって上記の処理に関するサービスを提供するクラウドとして実装することとしてもかまわない。

【００５７】

図７は、学習プログラムを実行するコンピュータの一例を示す図である。コンピュータ１０００は、例えば、メモリ１０１０、ＣＰＵ１０２０を有する。また、コンピュータ１０００は、ハードディスクドライブインタフェース１０３０、ディスクドライブインタフェース１０４０、シリアルポートインタフェース１０５０、ビデオアダプタ１０６０、ネットワークインタフェース１０７０を有する。これらの各部は、バス１０８０によって接続される。

【００５８】

メモリ１０１０は、ROM（Read Only Memory）１０１１及びRAM（Random Access Memory）１０１２を含む。ROM１０１１は、例えば、BIOS（Basic Input Output System）等のブートプログラムを記憶する。ハードディスクドライブインタフェース１０３０は、ハードディスクドライブ１０９０に接続される。ディスクドライブインタフェース１０４０は、ディスクドライブ１１００に接続される。例えば磁気ディスクや光ディスク等の着脱可能な記憶媒体が、ディスクドライブ１１００に挿入される。シリアルポートインタフェース１０５０は、例えばマウス１１１０、キーボード１１２０に接続される。ビデオアダプタ１０６０は、例えばディスプレイ１１３０に接続される。

【００５９】

ハードディスクドライブ１０９０は、例えば、OS１０９１、アプリケーションプログラム１０９２、プログラムモジュール１０９３、プログラムデータ１０９４を記憶する。すなわち、上記の学習装置１０が実行する各処理を規定するプログラムは、コンピュータにより実行可能なコードが記述されたプログラムモジュール１０９３として実装される。プログラムモジュール１０９３は、例えばハードディスクドライブ１０９０に記憶される。例えば、学習装置１０における機能構成と同様の処理を実行するためのプログラムモジュール１０９３が、ハードディスクドライブ１０９０に記憶される。なお、ハードディスクドライブ１０９０は、SSD（Solid State Drive）により代替されてもよい。

【００６０】

また、上述した実施形態の処理で用いられるデータは、プログラムデータ１０９４として、例えばメモリ１０１０やハードディスクドライブ１０９０に記憶される。そして、ＣＰＵ１０２０が、メモリ１０１０やハードディスクドライブ１０９０に記憶されたプログラムモジュール１０９３やプログラムデータ１０９４を必要に応じてRAM１０１２に読み出して実行する。

【００６１】

なお、プログラムモジュール１０９３やプログラムデータ１０９４は、ハードディスクドライブ１０９０に記憶される場合に限らず、例えば着脱可能な記憶媒体に記憶され、ディスクドライブ１１００等を介してＣＰＵ１０２０によって読み出されてもよい。あるいは、プログラムモジュール１０９３及びプログラムデータ１０９４は、ネットワーク（LAN（Local Area Network）、WAN（Wide Area Network）等）を介して接続される他のコンピュータに記憶されてもよい。そして、プログラムモジュール１０９３及びプログラムデータ１０９４は、他のコンピュータから、ネットワークインタフェース１０７０を介してＣＰＵ１０２０によって読み出されてもよい。

【符号の説明】

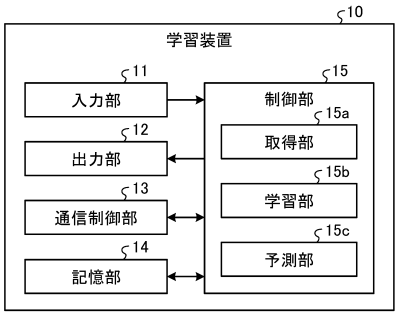
【００６２】

１０　学習装置
１１　入力部
１２　出力部
１３　通信制御部
１４　記憶部
１５　制御部

１５ａ　取得部
１５ｂ　学習部
１５ｃ　予測部
２０　検知装置

【図面】

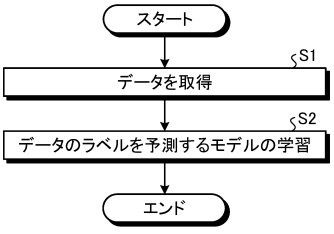
【図１】



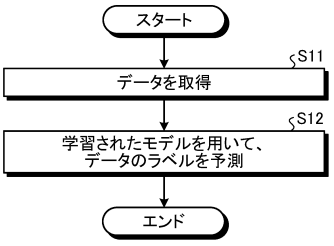
【図２】

$$
\begin{aligned}
D_{\mathrm{KL}}\left(p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\|p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}+\boldsymbol{\alpha}\odot\boldsymbol{v}\right)\right) &= \int p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)p\left(\boldsymbol{x}\right)\left(\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)-\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}+\boldsymbol{\alpha}\odot\boldsymbol{v}\right)\right)d\boldsymbol{y}d\boldsymbol{x} \\
&= \int p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)p\left(\boldsymbol{x}\right)\left(\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)-\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)-\sum_i\frac{\partial}{\partial w_i}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\left(\alpha^k v\right)_i-\sum_{ij}\frac{\partial^2}{\partial w_i\partial w_j}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\left(\alpha^k v\right)_i\left(\alpha^k v\right)_j+\mathcal{O}\left(\alpha^{3}\right)\right)d\boldsymbol{x}d\boldsymbol{y} \\
&\simeq \int p\left(\boldsymbol{x}\right)p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\left(-\sum_{ij}\frac{\partial^2}{\partial w_i\partial w_j}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)_c\left(\alpha^k v\right)_i\left(\alpha^k v\right)_j\right)d\boldsymbol{x}d\boldsymbol{y} \\
&= \int p\left(\boldsymbol{x}\right)p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\left(\sum_{ij}\frac{\partial}{\partial w_i}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\frac{\partial}{\partial w_j}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\left(\alpha^k v\right)_i\left(\alpha^k v\right)_j\right)d\boldsymbol{x}d\boldsymbol{y} \\
&= \int p\left(\boldsymbol{x}\right)\sum_{ij}\left(\alpha^k v\right)_i G_{ij}\left(\alpha^k v\right)_j d\boldsymbol{x} \\
&= \int p\left(\boldsymbol{x}\right)\sum_{ij}\gamma_i G'_{ij}\gamma_j d\boldsymbol{x} \qquad \because \lambda_i \boldsymbol{h}_i = G'_{ij}\boldsymbol{h}_i \\
&\simeq \int p\left(\boldsymbol{x}\right)\lambda_1\left\|\boldsymbol{h}_1\right\|^2 d\boldsymbol{x} \qquad \because \lambda_1 \boldsymbol{h}_1 = G'_{ij}\boldsymbol{h}_1 \\
G_{ij} &\equiv \int p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\frac{\partial}{\partial w_i}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\frac{\partial}{\partial w_j}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)d\boldsymbol{y} \\
G'_{ij} &\equiv \int p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\alpha_i^k\frac{\partial}{\partial w_i}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)\alpha_j^k\frac{\partial}{\partial w_j}\log p\left(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}\right)d\boldsymbol{y}
\end{aligned}
$$

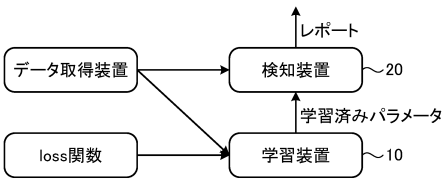
【図３】



【図４】



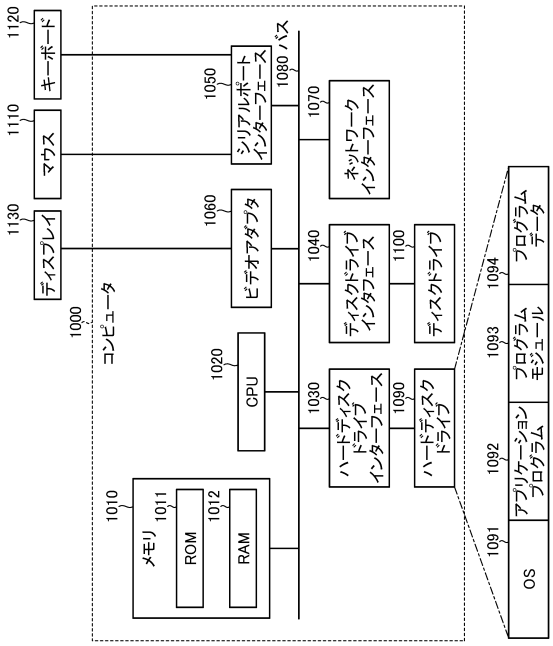
【図５】



【図６】

	robust acc	natural acc
AT	52.79	85.57
AWP	55.39	82.00
本実施形態	54.80	86.57

【図７】



フロントページの続き

(56)参考文献 WU, Dongxian et al.，"Adversarial Weight Perturbation Helps Robust Generalization"，arXiv.org [online]，2020年，pp. 1-20，[retrieved on 2021.08.16], Retrieved from the Internet: 〈URL: https://arxiv.org/abs/2004.05884v2〉
MIYATO, Takeru et al.，"Distributional Smoothing with Virtual Adversarial Training"，arXiv.org [online]，2016年，pp. 1-12，[retrieved on 2021.08.16], Retrieved from the Internet: 〈URL: https://arxiv.org/abs/1507.00677v9〉

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ０６Ｎ　２０／００－２０／２０
Ｇ０６Ｎ　３／０２－３／１０