

# 説明可能なAI



2021年12月13日(月)

第13回 自動チューニング技術の現状と応用に関するシンポジウム (ATTA2021)

【招待講演】

ソニーグループ (株) R&Dセンター  
シニアマシンラーニングリサーチャー

鈴木 健二

# 自己紹介

## 鈴木 健二

博士（工学）， 学士（法学）

シニアマシンラーニングリサーチャー



## 職歴

1999年 東京大学 生産技術研究所

2000年 フランス Institut d'Électronique et de Microélectronique du Nord (IEMN)

2001年 ソニー（株）

現在 ソニーグループ（株） R&Dセンター

## 研究開発テーマ

説明可能なAI， 機械学習における公平性， データ流通

# Agenda

AI倫理と技術

判断根拠の可視化手法

モデルへのデータ影響度  
データクレンジング, ミスラベル検知

データの不確実性

近年の動向や課題

説明可能なAIツール



# AI倫理と技術



# AI倫理の問題

(社会的課題)  
**(技術的課題)**

AIの社会的受容性の向上  
**AIの説明性の解明と公平性**

女性に不利な採用

人間をゴリラとしてタグ付け

自動運転での死亡事故

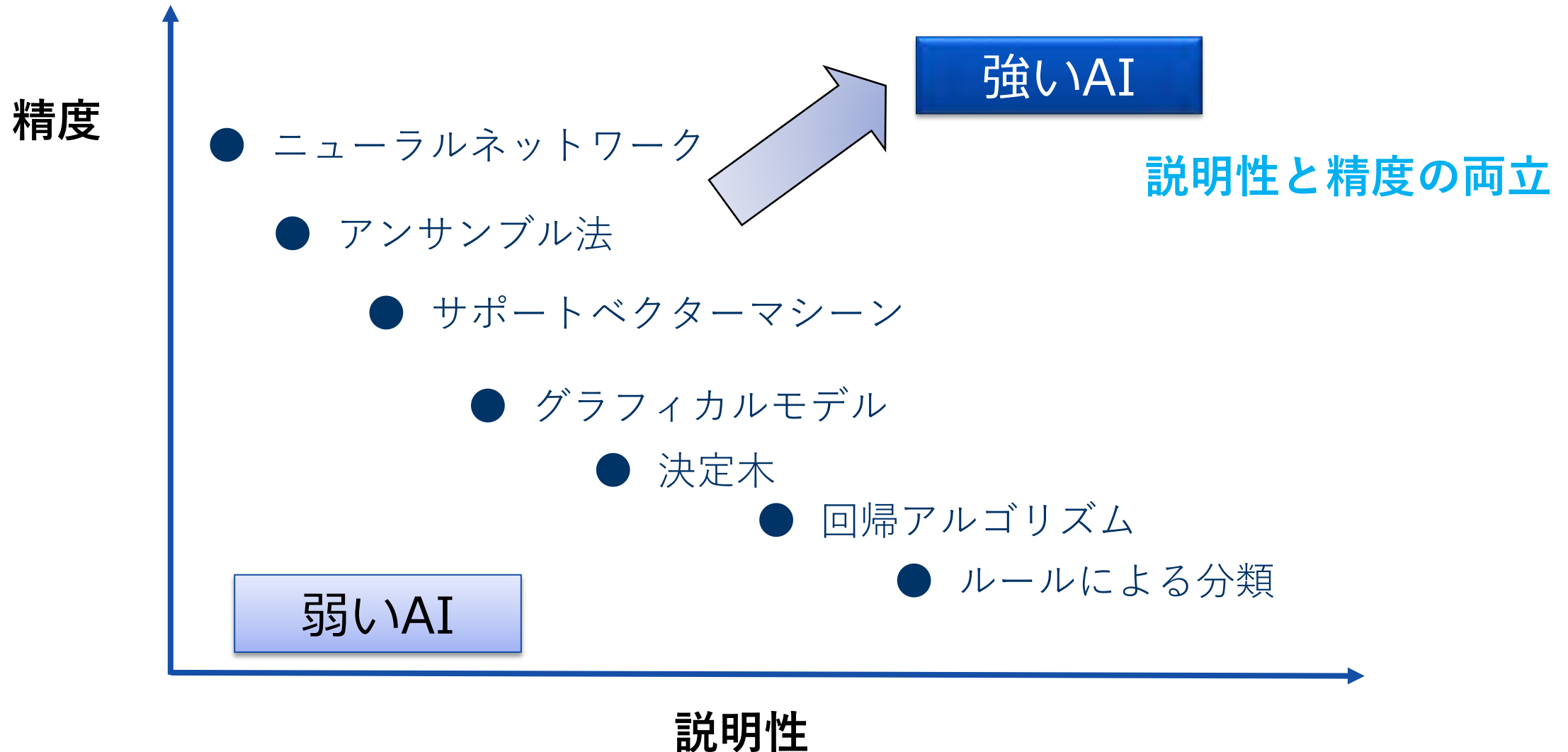
人種間での誤認識率の違い

差別的な発言

# AIに関する法制度・ガイドラインの動向

GDPR 一般データ保護規則	欧州	2018	決定に含まれているロジックに関する意味のある情報等を提供しなければならない（13～15条）． 完全自動意思決定の原則禁止（22条）．
AI利活用ガイドライン	日本	2019	透明性の原則， アカウンタビリティの原則
欧州AI規制法案	欧州	2021	AIの包括的な規制法案． 許容できないAI，ハイリスクAI，特定のAIシステム，最小リスクAIにカテゴライズされ，その義務や要件を規定．

# 機械学習における説明性と精度



# 説明可能なAI



AIはブラックボックス

判断根拠がわからない

説明できないのか？

AIは人間を超える性能を持つ。その判断の根拠はどうなっているのだろうか？



# AIの判断根拠が必要となる例



## AI利活用

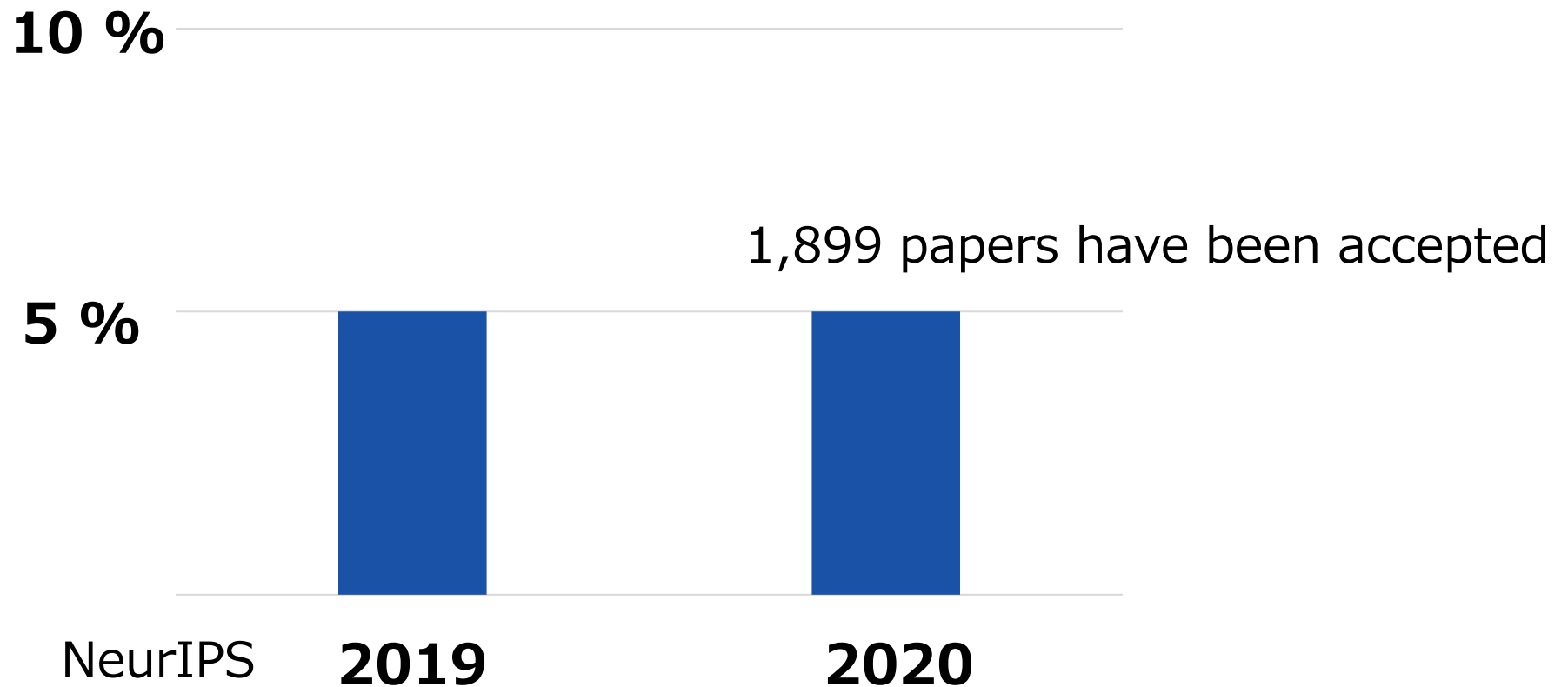
優れている点	課題
高い精度	判断理由の説明

医師による診断でのAI利活用のケース

AIの判断理由が分からないと, 原因の解明ができない.

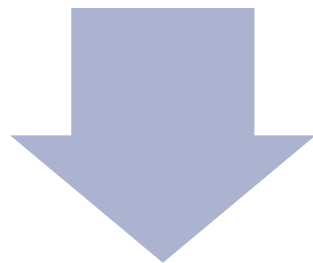
# AI トップカンファレンス NeurIPS2020 における AI 倫理技術関連論文 *Fairness, Accountability, Transparency, Explainability, Privacy etc.*

*Social aspects of Machine Learning* share of accepted paper



# AI トップカンファレンス NeurIPS2020 投稿論文のAI倫理レビュー

Machine learning has real-world impact

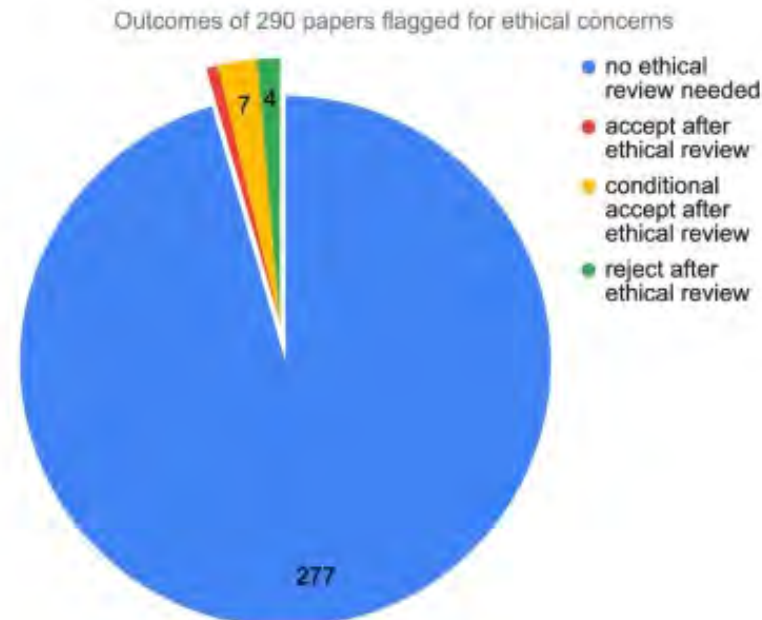


Ethical review

Statement added to the Call for Papers :  
*"Regardless of scientific quality or contribution, a submission may be rejected for ethical considerations, including methods, applications, or data that create or reinforce unfair bias or that have a primary purpose of harm or injury."*

## Results of ethical review

- Outcomes of 290 papers flagged for ethical concerns
- 13 papers were ethically reviewed.
- 2 papers were accepted.
- 7 were conditionally accepted.
- **4 paper were rejected.**



Refer from NeurIPS2020 official Website

# AI トップカンファレンス NeurIPS2020 からみる AI 倫理技術のトレンド

Explainability beyond  
classification

More reliable and  
robustness  
explanation methods

Rigorous evaluation  
of explanation

Theoretical analysis  
of explanation

Bias mitigation with  
fairness

# システム1からシステム2ディープラーニングへ

人間的な思考へ

## 現在のディープラーニング

- 直感的
- 高速
- 無意識
- 非言語的
- 習慣的



## 将来のディープラーニング

- 論理的
- 低速
- 意識
- 言語
- 計画
- 論拠

ハイレベルの概念を操作

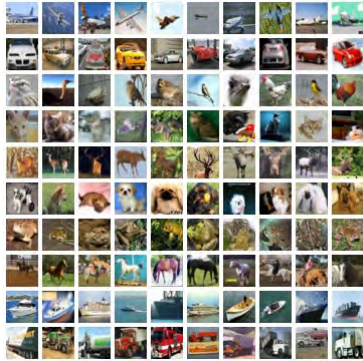
Yoshua Bengio “From System1 Deep Learning to System2 Deep Learning” NeurIPS 2019 <https://youtu.be/T3sxeTgT4qc>



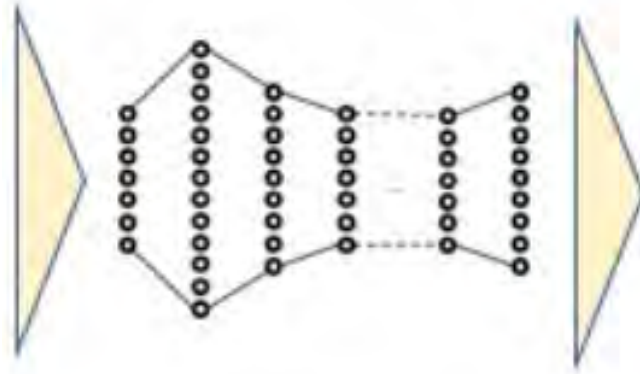
# 説明可能なAIとは

# 説明可能なAI

eXplainable AI (XAI)



データ  
Data



ディープニューラルネットワーク  
DNN

説明モデル

説明インターフェイス



CIFAR-10 dataset : Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

説明可能なAIは、AIの判断根拠を可視化することができる技術。

# Responsible AIと説明可能なAI

## Responsible AI

**Fairness**  
**Accountability**  
**Transparency**

Defining Sony's stance in relation to AI initiatives and promoting dialogue

**Sony Group AI Ethics Guidelines**

SDGs

**SUSTAINABLE DEVELOPMENT GOALS**  
17 GOALS TO TRANSFORM OUR WORLD



5. Gender Equality

10. Reduced Inequalities

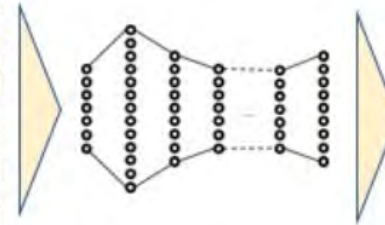
12. Responsible Consumption  
and production

16. Peace, Justice,  
and Strong Institutions

## eXplainable AI (XAI)



Data



DNN

Explanation  
model

Explanation  
Interface

- AIの判断の理由を理解
- 人間の意図にあったAIの制御

# AI倫理技術の実装



# 説明可能なAIの利用価値

AI倫理、法令遵守 AI ethics, Compliance

画像分類, 顔識別, 人物体検出など

画像生成など

金融, マーケティング

自然言語処理

製造, 検査

マテリアル・インフォマティクス

医療

自動運転, 行動学習

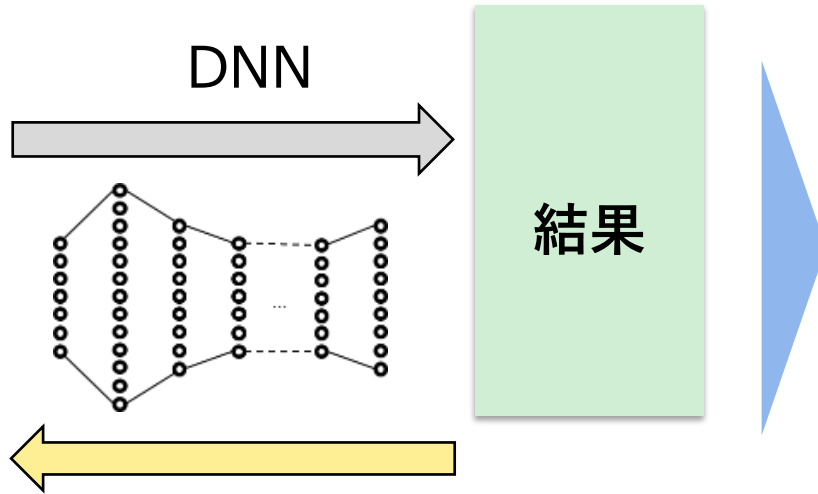
解析全般, 最適化問題

データの質向上

データ



原因



CIFAR-10 dataset : Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

# XAI

DNNの潜在的な予測精度を引き出す。

XAI技術は、AI倫理のみならず、多方面へ応用展開できる。



# 説明可能なAIのスコープ

## Fairness Accountability Transparency (FAT)

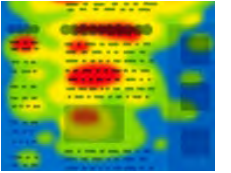
Improving accuracy is an approach to the FAT issues

- \* Data cleansing
- \* Fairness check/definition/solution
- \* Data bias mitigation
- \* Learning Mechanism
- \* Model debugging
- \* Adversarial defense
- \* Robustness



### Visualization

XAI shows the reason of judgement



### Understanding

Humans understand the reason



### Feedback

XAI corrects AI



説明可能なAIは、AI 倫理だけでなく、ディプラーニングの可能性を更に引き出す。

# AIの判断根拠の可視化手法



## 判断根拠

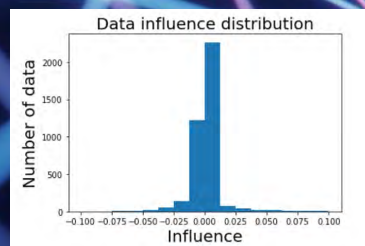
Basis for judge

- Grad-CAM
- LIME
- SHAP
- Smooth Grad
- Attention Branch Network

## データの影響度

Data influence

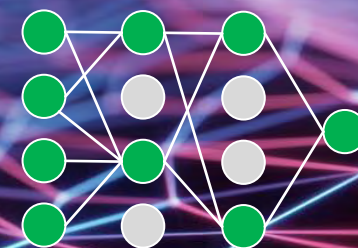
- SGD influence
- Influence functions
- TracIn



## データの不確実性

Data uncertainty

- MC dropout

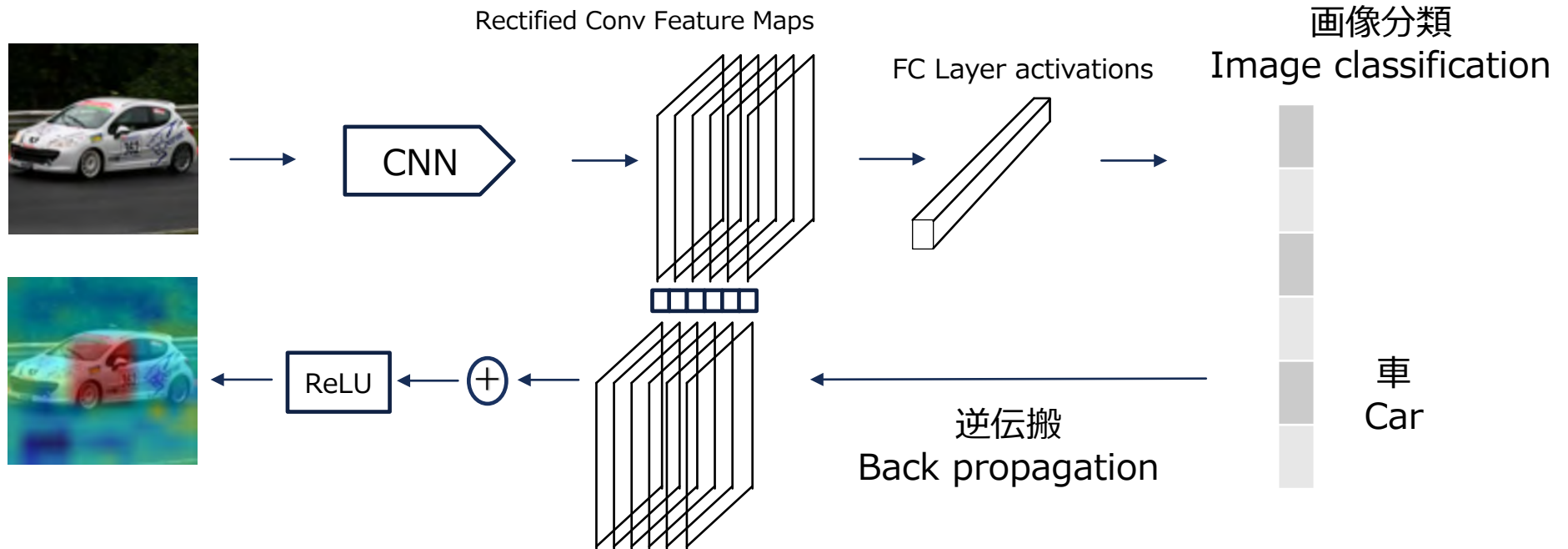


# 代表的な説明可能なAI

XAI技術	特徴	画像	表データ	テキスト
Grad-CAM	CNNでの畳み込み層の勾配を利用し、画像中の判断根拠となる箇所をヒートマップで表示する技術である。	○	×	×
LIME	モデルを局所的に線形モデルで近似することによって判断根拠を示す方法である。	○	○	○
SHAP	ゲーム理論のShapley 値を求める手法を使って各特徴量の寄与度を計算する手法である。	○	○	○
Smooth Grad	入力画像にガウシアンノイズを載せ、複数回の勾配計算をした後に平均を取ることによって、判断根拠となる箇所の可視化画像を生成する手法である。	○	×	×

# Grad-CAM

## Deep learning はどこをみているのだろうか？



STL-10 dataset : Adam Coates, Honglak Lee, Andrew Y. Ng An Analysis of Single Layer Networks in Unsupervised Feature Learning AISTATS, 2011.

CNNでの畳み込み層の勾配を利用し、画像中の判断根拠となる箇所をヒートマップで表示する技術である。

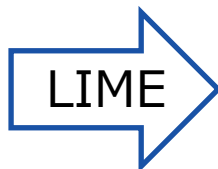
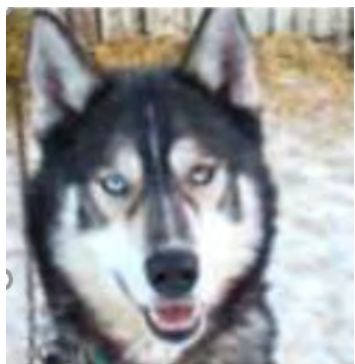
Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra.  
Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International conference on computer vision, pages 618 - 626, 2017.



# LIME

## LIME (Local Interpretable Model-agnostic Explanation)

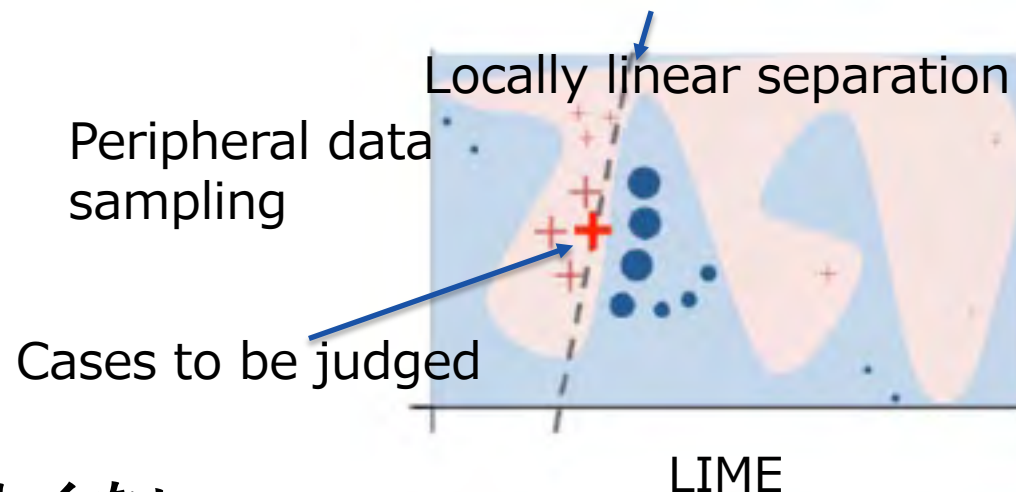
局所的な説明の代表的な方法



ハスキー犬が狼と判断

背景の雪が判断根拠

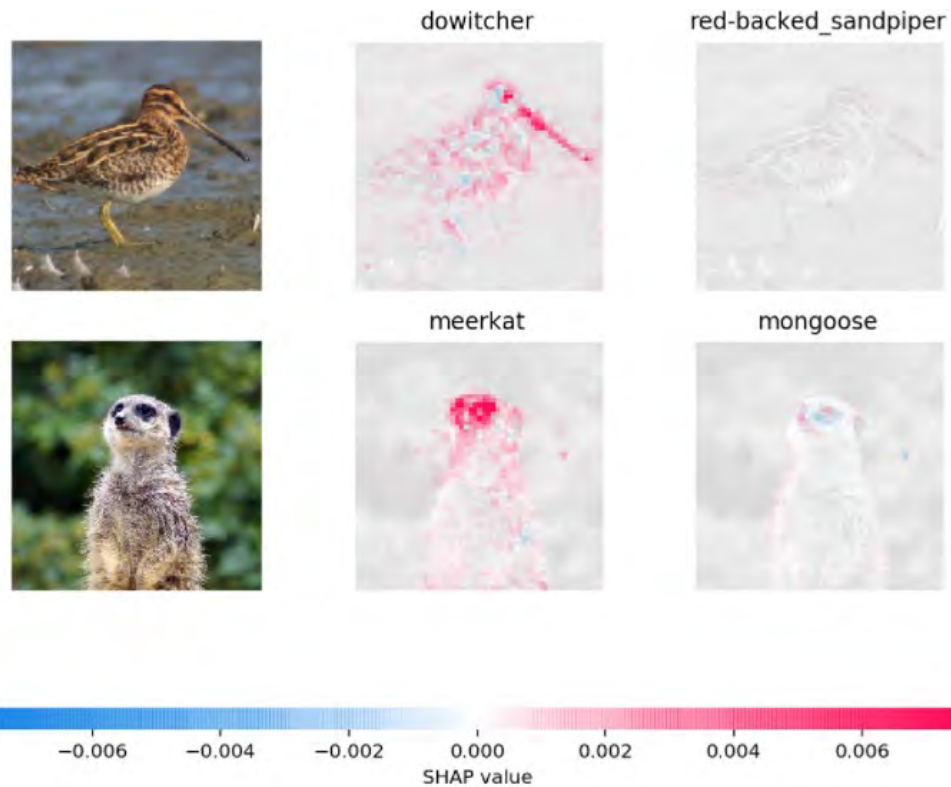
このモデルは、判断根拠が正しくない



Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining, pages 1135 - 1144, 2016.

LIMEは、モデルを局所的に線形モデルで近似することによって判断根拠を示す方法である。

# SHAP



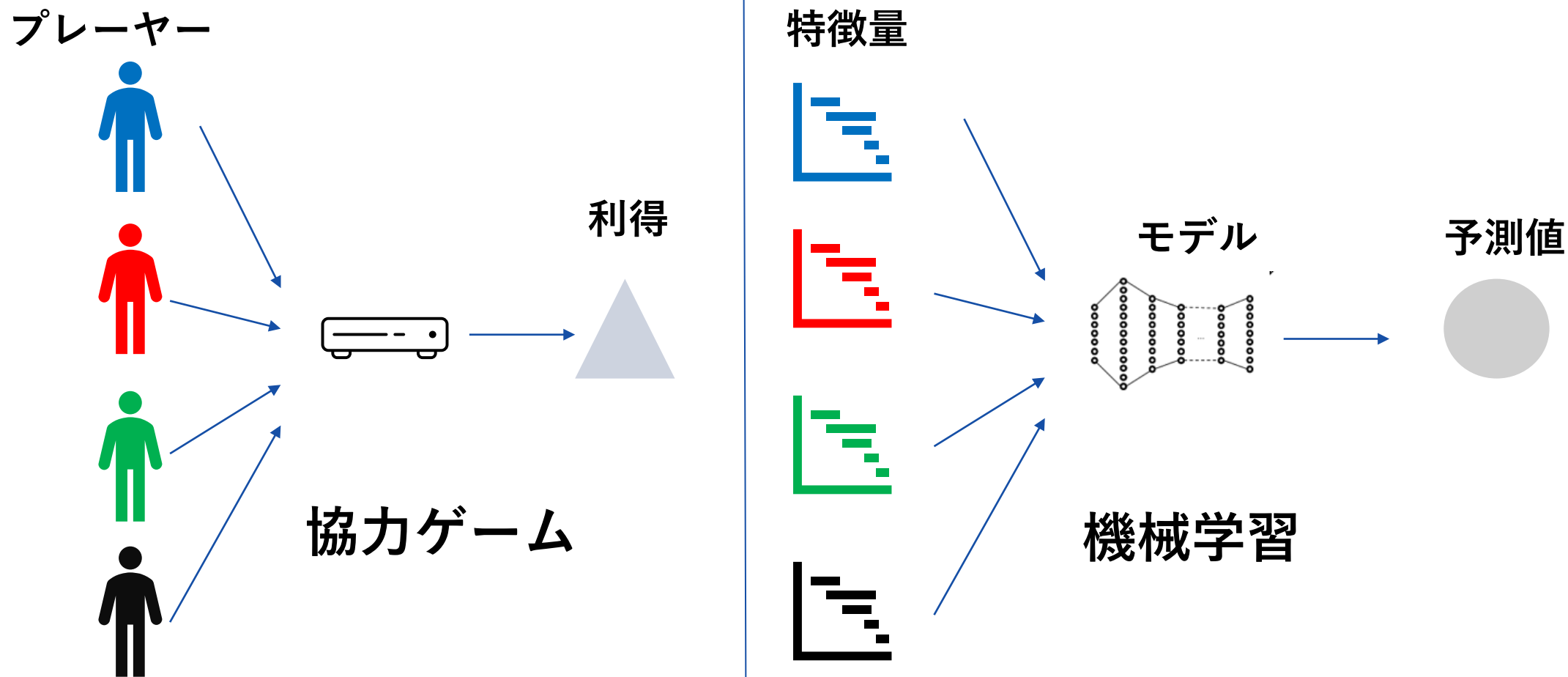
赤： プラスに寄与

青： マイナスに寄与

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30, pages 4768–4. 2017.

SHAPは、ゲーム理論のShapley 値を求める手法を使って  
各特徴量の寄与度を計算する手法

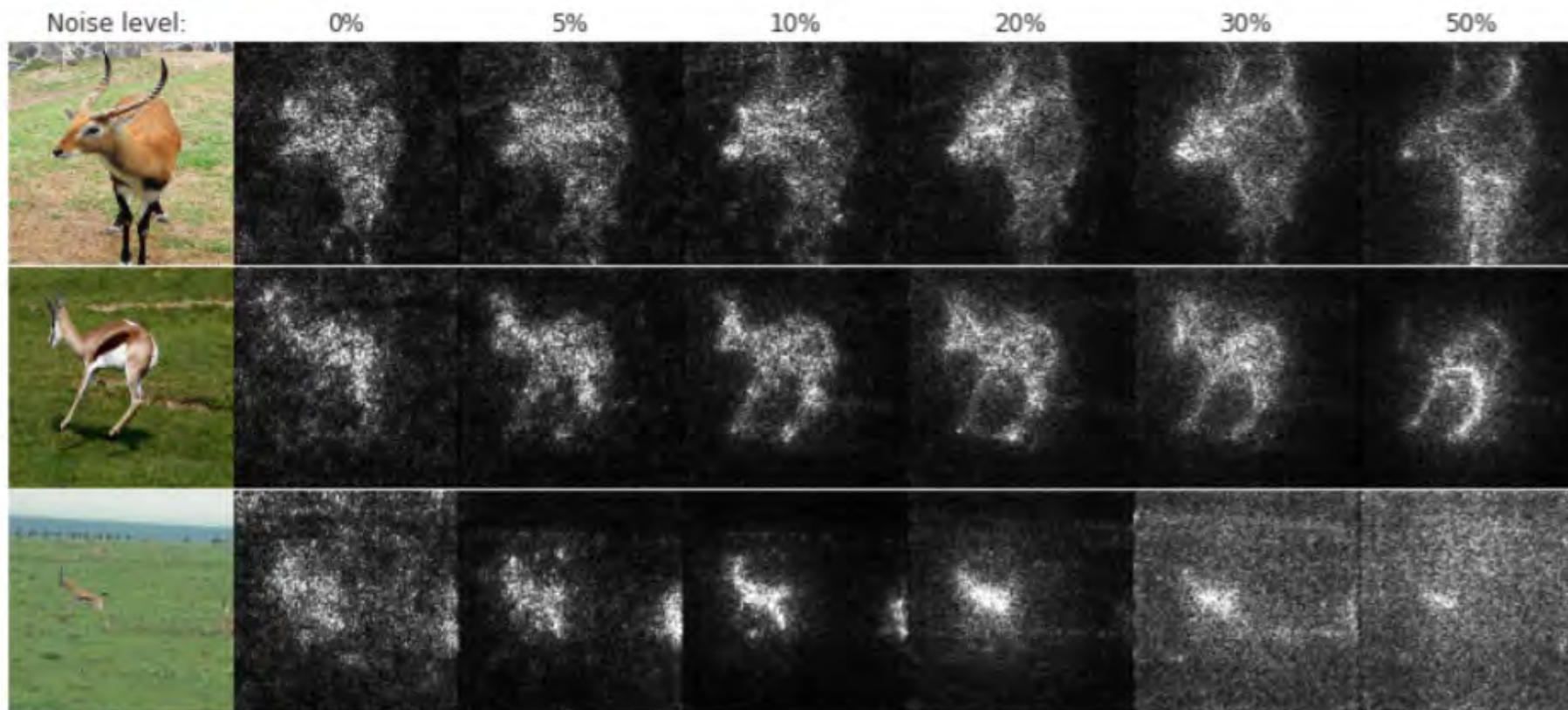
# 協力ゲームと機械学習



協力ゲーム理論のShapley値は、プレイヤー間での報酬を公平に分配する方法。

この考え方を機械学習に適用し、各特徴量の寄与度を算出する。

# Smooth Grad



Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. arXiv:1706.03825, 2017.

SmoothGradは、入力画像にガウシアンノイズを載せ、複数回の勾配計算をした後に平均を取ることによって、判断根拠となる箇所の可視化画像を生成する手法。





## 説明可能なAI 対応

● プログラミング不要

● 簡単なGUI操作

● 一貫して実行可能

# ディープラーニング・説明可能なAIの統合開発フレームワーク



The screenshot displays the Neural Network Console interface. On the left, a menu is open with 'XAI' selected, showing options like 'Grad-CAM (batch)'. In the center, a table shows results for 'y'\_9' with values like 0.99995637. On the right, a 'gradcam' section shows three heatmaps labeled 'gradcam\_0000\_0.png', 'gradcam\_0000\_1.png', and 'gradcam\_0000\_2.png'. Further right, an 'Overview: Main' section shows a layer-wise breakdown of the model's architecture and statistics.

Layer	Time
Input	1:25.28
Convolution	16:22.22
ReLU	18:22.21
MaxPooling	14:21.11
Convolution2	20:2.1
MaxPooling_2	20:4.4
Tanh_2	20:4.4
Attnet	1:50
ReLU_2	1:50
Attnet_2	1:10
Softmax	1:10
LogSoftmax	1:10

Statistic	Value
Output	21,919
CostParameter	78,810
CostAdd	11,304

GUIによる簡単操作

Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf. arXiv:1708.07747

<https://dl.sony.com/ja/>



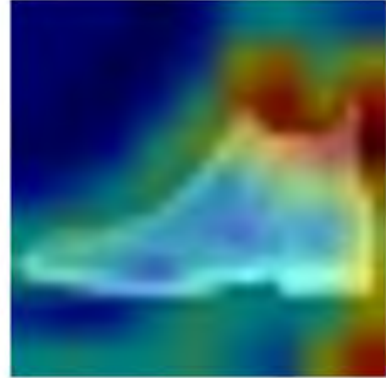
モデルの構築、学習、説明可能なAIまで、  
Neural Network Consoleにて一貫して開発できる。



# 画像分類における判断の根拠の可視化



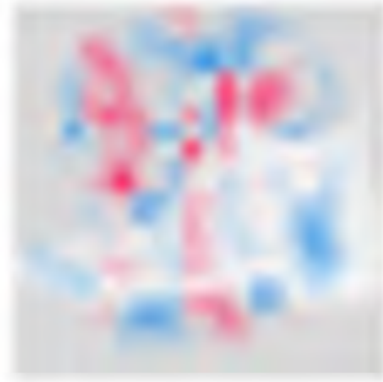
(a) Original



(b) Grad-CAM



(c) LIME



(d) SHAP



(e) SmoothGrad

靴の上部にて判断していることが分かる



(a) Ankle boot













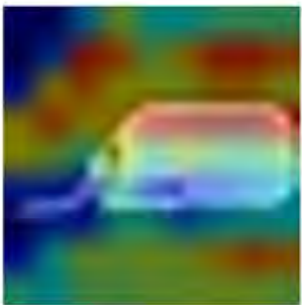
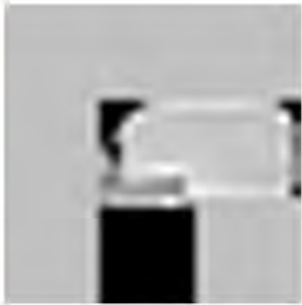

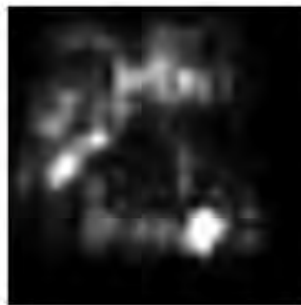
(b) Sandal


↑ ↓ 違いは、靴の上部

それぞれの手法による判断根拠の可視化は、必ずしも一致しない。  
このように説明手法が異なれば、得られる説明も異なる。

データセット内での似たような画像

# 画像分類における誤判断の根拠の可視化

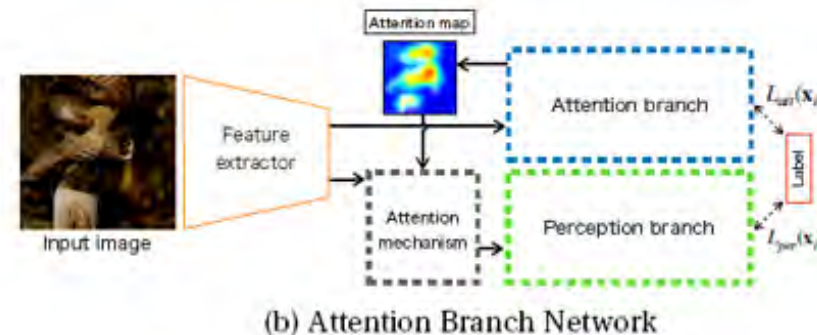
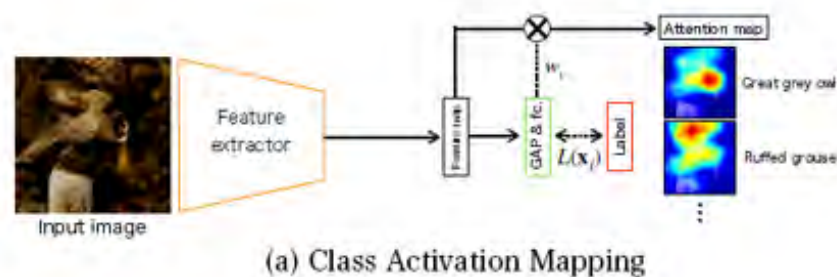
真のラベル	判断	(a) Original	(b) Grad-CAM	(c) LIME	(d) SHAP	(e) Smooth grad
<b>Ankle boot</b> 	<b>Pullover</b> 					
<b>Bag</b> 	<b>Ankle boot</b> 					

An aerial, top-down view of a modern office space. The office is brightly lit with a greenish-white glow. Several white rectangular tables are arranged in a grid-like pattern. People are seated at these tables, working on laptops and tablets. The chairs are colorful, including yellow, green, and orange. The overall atmosphere is clean, professional, and collaborative.

# AIの判断根拠を活用する技術

判断根拠の可視化だけでなく、精度の向上

# Attention Branch Network



H. Fukui, T. Hiraoka, T. Yamashita, and H. Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

- 判断根拠を用いて**CNNの性能を向上させる初めての試み**
- より注目すべき場所を示すことで効率よく学習しモデルの精度を向上
- ベースのモデルにAttention Branchを導入するため様々なモデルに適用可能



# 機械学習モデルへのデータの影響度

- データクレンジング
- ミスラベル検知

## 判断根拠

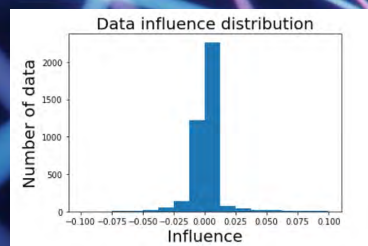
Basis for judge

- Grad-CAM
- LIME
- SHAP
- Smooth Grad
- Attention Branch Network

## データの影響度

Data influence

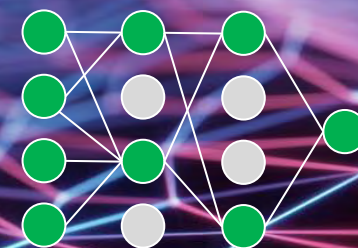
- SGD influence
- Influence functions
- TracIn



## データの不確実性

Data uncertainty

- MC dropout





# データクレンジング

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with SGD. In *Advances in Neural Information Processing Systems*, pages 4215-4224, 2019



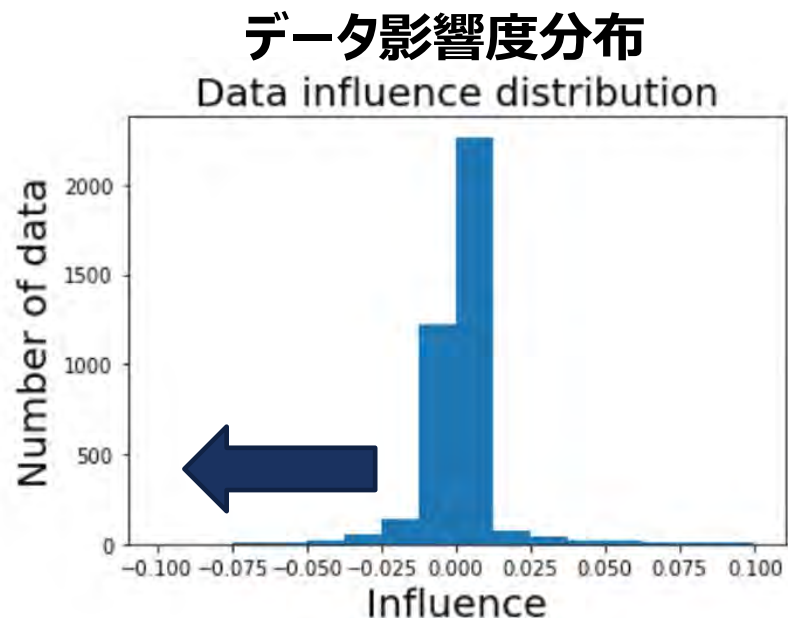
## DNNへ悪影響を及ぼすデータを削除



データを追加することなく精度向上

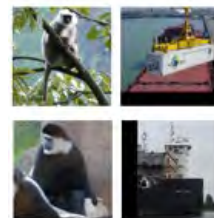


# データの影響度 Data influence SGD influence



悪影響データ

Negative influence data



悪影響データの除去  
Remove bad influence data



DNNの精度向上  
Accuracy up of DNN

SGD influenceは、悪影響データを抽出することができる。

# データの質をどのように向上させるか

Deep learning requires a large amount of data and labeling

## Current issues



- Require a large amount of data
- Uniformly correct of high-quality data
- Need to label data
- Require domain knowledge for labeling

## Goal

Automatically clean data of DNN

## Method

$$\langle u, \theta_{-k}^{[T]} - \theta^{[T]} \rangle$$

## Results



Use data cleansing on Neural Network Console

# DNNへのデータ影響度の計算

Which data should be removed to improve accuracy ?  
How can we know the influence of a piece of data on DNN ?

$z_k$

Removed Data	Loss function
Positively influential	increase
Negatively influential	decrease

$\theta$  : model parameter  
 $D, z$  : data  
 $l$  : loss function

Small loss  $\approx$  Linear influence

Identify **negatively influential** data

Remove **negatively influential** data

Retrain

**Accuracy up**

influence  $\tilde{r}_k = \langle u, \theta_{-k}^{[T]} - \hat{\theta}^{[T]} \rangle$

$\hat{\theta}_{-k} = \arg \min_{\theta} \sum_{z \in D \setminus \{z_k\}} l(z; \theta)$      $\hat{\theta} = \arg \min_{\theta} \sum_{z \in D} l(z; \theta)$

Issue

It takes an enormous amount of time to remove the data **one by one and retrain.**

Approach

Estimate  $\hat{\theta}_{-k} - \hat{\theta}$  **without retraining**

# データ影響度計算のアルゴリズム

---

## Algorithm 1 Training phase

---

Initialize parameter  $\theta$  [1]

Initialize sequence as null :  $A \leftarrow 0$

**for**  $t = 1, 2, \dots, T - 1$  **do**

$$\theta^{[t+1]} \leftarrow \theta^{[t]} - \frac{\eta_t}{|S_t|} \sum_{i \in S_t} g(z_i; \theta^{[t]})$$

**end for**

$A \leftarrow \theta$  // store parameters

---

---

## Algorithm 2 Inference phase

---

**Require:**  $u^{[T-1]} \in \mathbb{R}^p$

Initialize influence:  $\hat{L}_{-j}^{[T]}(u) \leftarrow 0, \forall j$

$\theta \leftarrow A$  // load parameters

**for**  $t = T - 1, T - 2, \dots, 1$  **do**

// update linear influence of  $z_j$

$$\hat{L}_{-j}^{[T]}(u^{[t]})_+ = \langle u^{[t]}, \frac{\eta_t}{|S_t|} g(z_j; \theta) \rangle, \forall j \in S_t$$

$$u^{[t-1]} = u^{[t]} - \eta_t H^{[t]} u^{[t]} // \text{update } u$$

**end for**

---



# ストレージに対して効果的なデータ影響度計算手法

従来手法 [Hara+NeurIPS2019]

**全ての**イテレーションのパラメータを利用

$\theta^{[1]} = \theta_0$     $\theta^{[2]}$     $\theta^{[3]}$     $\theta^{[4]}$     $\theta^{[5]}$     $\theta^{[6]}$     $\theta^{[7]}$    ...  
→ → → → → → →

最終エポック

我々の提案手法

**最後の**パラメータのみを利用

$\theta$

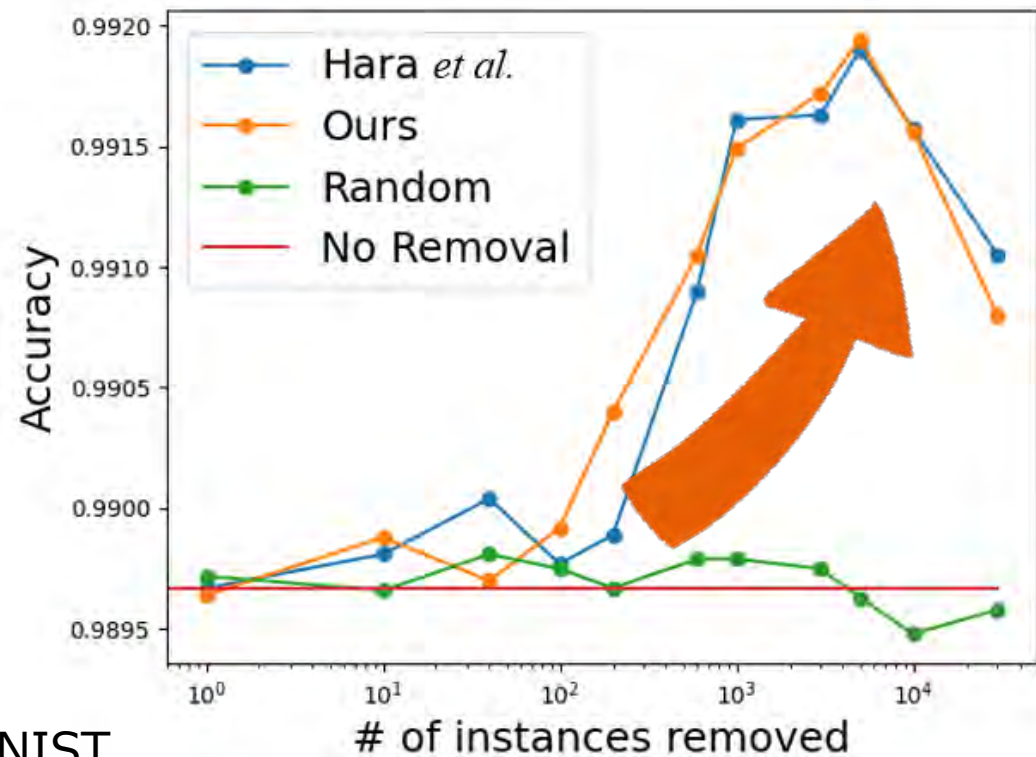
# キャッシュサイズを1/1,563へ削減

Table 1 : Cache size of the parameters in training

Methods	Cache	Cache size (GB)	
		MNIST	CIFAR10
All parameters	$\theta \times T \times k$	38.64	30.90
Hara <i>et al.</i>	$\theta \times T$	1.932	1.545
Ours	$\theta$	0.001236	0.001236



データセット MNIST






従来手法と同等のデータクレンジング性能

# Neural Network Console 実装



<https://dl.sony.com/ja/>

## Neural Network Console

x:image	y:label	influence
	8	-0.012459796509163769
	0	-0.006328233351282367
	0	-0.0063282314659545745

**Input**  
Dataset: x 3, 32, 32

- RandomShuffle 3, 32, 32
- RandomFlip 3, 32, 32
- MulScalar\_2 Value: 0.01735 3, 32, 32
- AddScalar\_2 Value: -1.99 3, 32, 32
- Convolution KernelShape: 3, 3 16, 32, 32
- RepeatStart\_3 x18 16, 32, 32

**BatchNormalization\_2** x18 16, 32, 32

- ReLU\_2 x18 16, 32, 32
- Convolution\_2 KernelShape: 1, 1 x18 16, 32, 32
- BatchNormalization\_3 x18 16, 32, 32
- ReLU\_6 x18 16, 32, 32
- Convolution\_3 KernelShape: 3, 3 x18 16, 32, 32

**Add2** x18 16, 32, 32

- RepeatEnd\_3 16, 32, 32
- BatchNormalization\_3 16, 32, 32
- ReLU\_3 16, 32, 32

**Convolution\_15** KernelShape: 3, 3 32, 16, 16

**Convolution\_4** KernelShape: 3, 3 32, 16, 16

**BatchNormalization\_4** 32, 16, 16

# データクレンジングについての論文



We gratefully acknowledge support from the Simons Foundation and member institutions.

arXiv.org > cs > arXiv:2103.11807v2

Search... All fields Search

Help | Advanced Search

Computer Science > Machine Learning

[Submitted on 22 Mar 2021 (v1), last revised 1 Jun 2021 (this version, v2)]

## Data Cleansing for Deep Neural Networks with Storage-efficient Approximation of Influence Functions

Kenji Suzuki, Yoshiyuki Kobayashi, Takuya Narihira

Identifying the influence of training data for data cleansing can improve the accuracy of deep learning. An approach with stochastic gradient descent (SGD) called SGD-influence to calculate the influence scores was proposed, but, the calculation costs are expensive. It is necessary to temporally store the parameters of the model during training phase for inference phase to calculate influence scores. In close connection with the previous method, we propose a method to reduce cache files to store the parameters in training phase for calculating inference score. We only adopt the final parameters in last epoch for influence functions calculation. In our experiments on classification, the cache size of training using MNIST dataset with our approach is 1.236 MB. On the other hand, the previous method used cache size of 1.932 GB in last epoch. It means that cache size has been reduced to 1/1,563. We also observed the accuracy improvement by data cleansing with removal of negatively influential data using our approach as well as the previous method. Moreover, our simple and general proposed method to calculate influence scores is available on our auto ML tool without programming, Neural Network Console. The source code is also available.

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI)

Cite as: [arXiv:2103.11807](https://arxiv.org/abs/2103.11807) [cs.LG]

(or [arXiv:2103.11807v2](https://arxiv.org/abs/2103.11807v2) [cs.LG] for this version)

### Submission history

From: Kenji Suzuki [[view email](#)]

[v1] Mon, 22 Mar 2021 13:08:46 UTC (921 KB)

[v2] Tue, 1 Jun 2021 08:23:25 UTC (960 KB)

<https://arxiv.org/abs/2103.11807v2>

詳しくは, 論文をご覧ください.

### Download:

- PDF
- Other formats



Current browse context:

cs.LG

< prev | next >

[new](#) | [recent](#) | [2103](#)

Change to browse by:

CS

[cs.AI](#)

### References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar



### DBLP

[listing](#) | [bibtex](#)

Kenji Suzuki  
Takuya Narihira

### Export Bibtext Citation

### Bookmark





# GitHubへコード公開

GitHub navigation bar with search bar, navigation links (Pull requests, Issues, Marketplace, Explore), and user profile icons.

Repository header for `sony / ai-research-code`. Includes Watch (26), Star (85), and Fork (10) buttons. Navigation tabs for Code, Issues (1), Pull requests, Actions, Projects, Wiki, Security, and Insights.

Repository controls: `master` branch selector, 6 branches, 0 tags, and buttons for `Go to file`, `Add file`, and `Code`.

Recent commit history table:

	TakuyaNarihira Merge pull request #21 from sony/feature/20210611-Add-Colab...	32F2698 18 hours ago	🕒 37 commits
📁	d3net	Add note: CityScapes-like input images for best results	3 days ago
📁	data-cleansing	Link Chnaged	2 months ago
📁	mixed-precision-dnns	Mixed Precision DNNs	14 months ago
📁	out-of-core-training	Add an explanation for OoC	4 months ago
📁	x-umx	Remove cudnn heuristic flag from test.py file	2 months ago
📄	.gitianore	Minor fix for release	6 months ago

## About

No description, website, or topics provided.

📖 Readme

📄 Apache-2.0 License



# データクレンジング まとめ

## ✓提案

- ストレージに対して効果的なデータ影響度計算手法を提案

## ✓結果

- キャッシュサイズを1/1,563へ削減(MNISTでの実験)
- 従来手法と同等のデータクレンジング効果あり

## ✓実装

- Neural Network Consoleへ実装
- GitHubへコードを公開

# ミスラベル検知

## 課題

- 実社会のデータの教師あり学習では、ミスラベルが起きやすい。

## 解決策

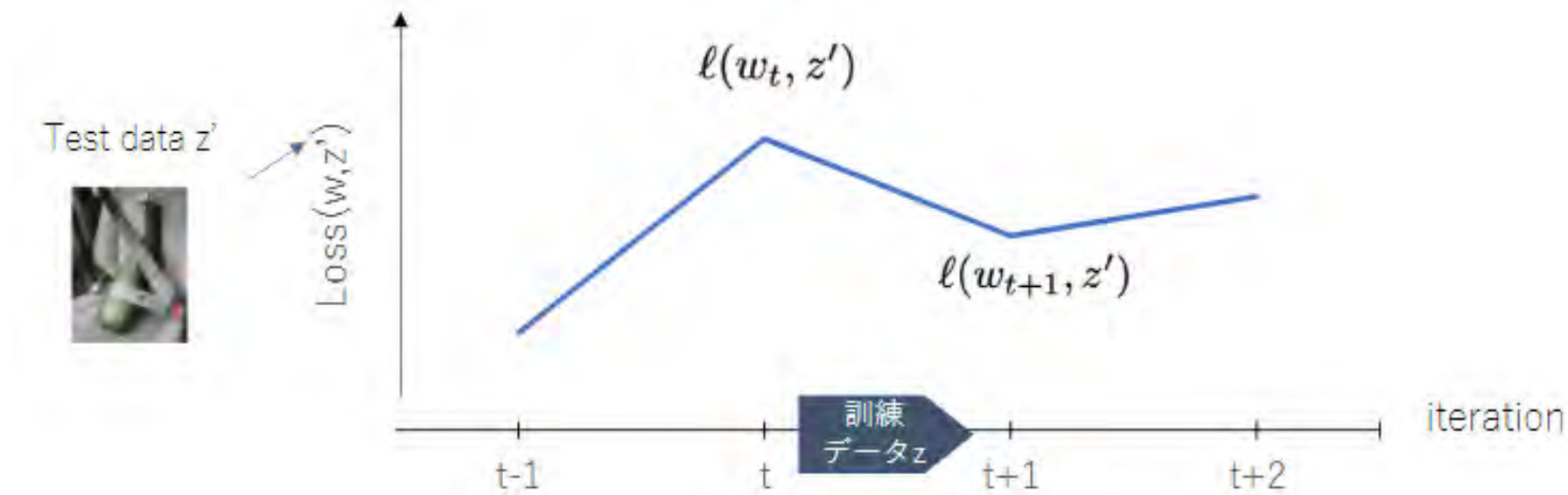
- 訓練データのミスラベルを検知する。

## 手法

- パラメータ更新による損失の変化を追跡し、訓練データの影響を定式化。

Garima Pruthi, Frederick Liu, Mukund Sundararajan, Satyen Kale  
[Estimating Training Data Influence by Tracing Gradient Descent](#), NeurIPS2020.

# 訓練データの影響の定式化(TracIn)



訓練過程でのTest dataのLossの変化をみる

Lossが減少

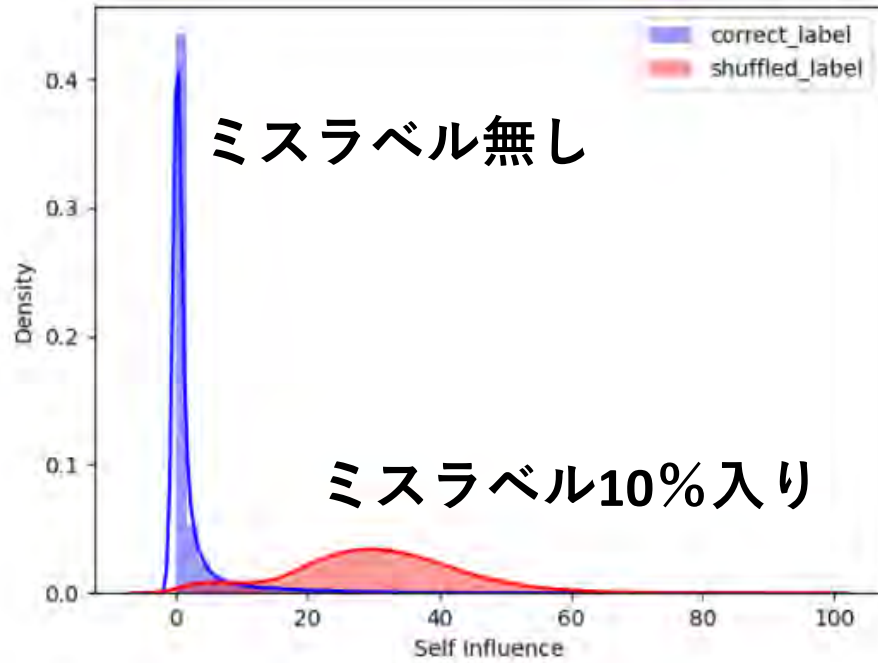


訓練に寄与したデータは良い影響を持っている

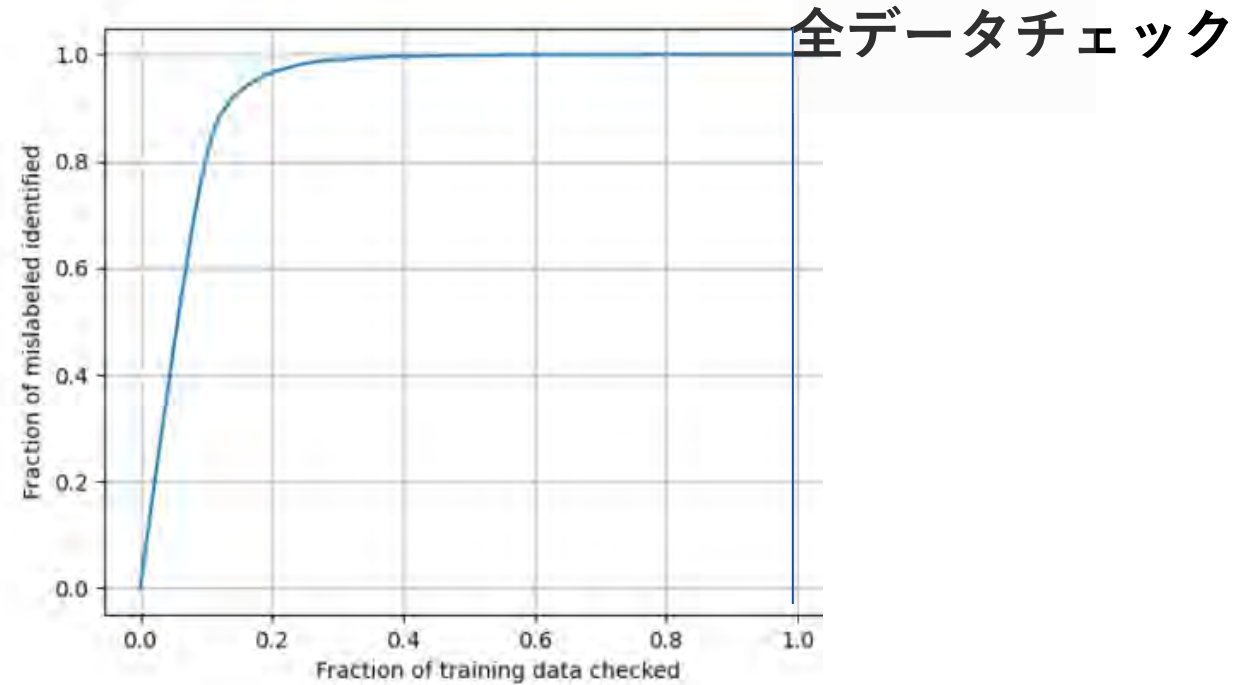


# ミスラベル検出の結果

## 影響度の分布



## ミスラベルの検出



[https://github.com/sony/nnabla-examples/tree/master/responsible\\_ai/tracin](https://github.com/sony/nnabla-examples/tree/master/responsible_ai/tracin)

# データの不確実性



## 判断根拠

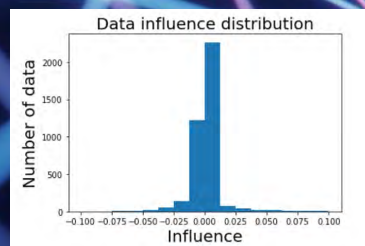
Basis for judge

- Grad-CAM
- LIME
- SHAP
- Smooth Grad
- Attention Branch Network

## データの影響度

Data influence

- SGD influence
- Influence functions
- TracIn



## データの不確実性

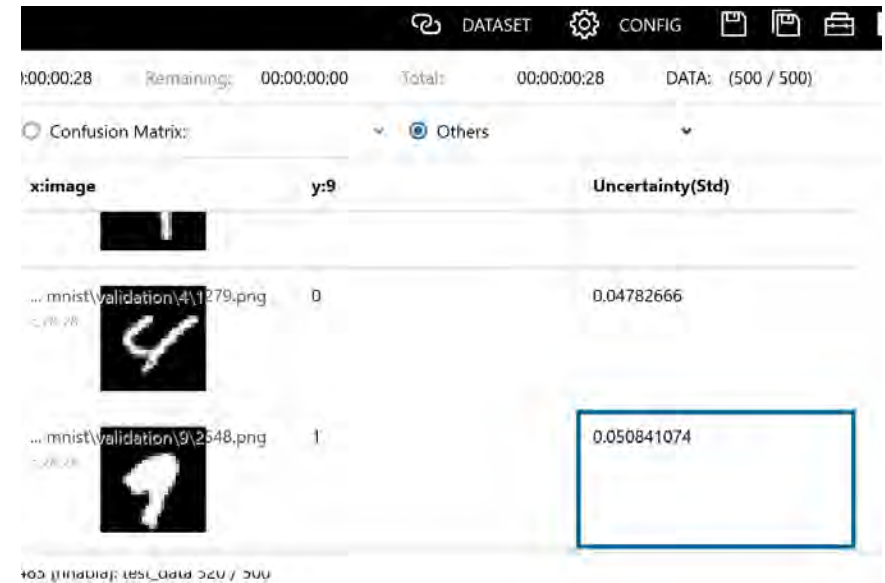
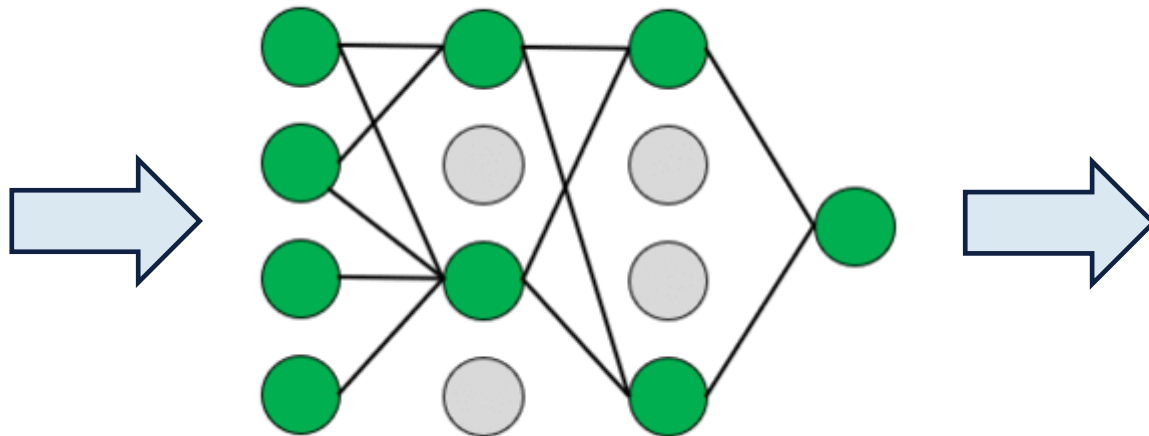
Data uncertainty

- MC dropout



# データの不確実性 Data uncertainty

# MC dropout



# 近似ベイズ推論 Approximate Bayesian inference

MC dropoutは、データの不確実性を示すことができる。



# データの不確実性

近似ベイズ推論による不確実性を伴った予測

Deep learning

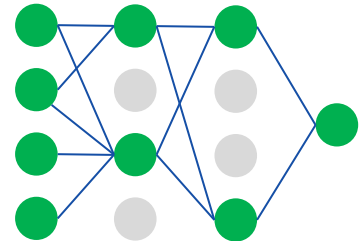
$$y = f(w, x)$$

パラメーターwは定数. 入力xによりyが決まる.

Bayesian Deep Learning

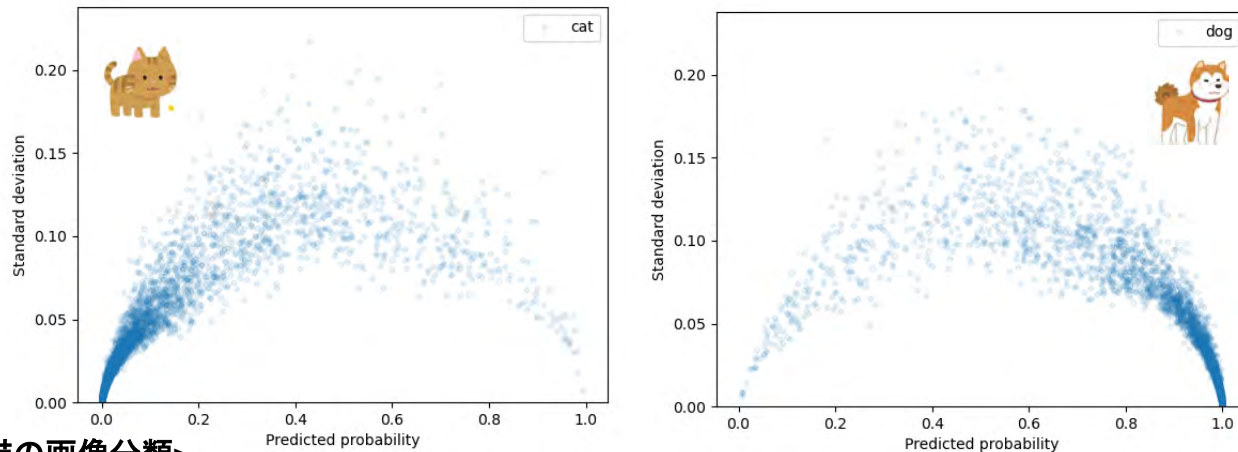
$$q(y^* | x^*) = \int p(y^* | x^*, \omega) q(\omega) d\omega$$

確定な点推定パラメーターではなく, 全ての推定確率分布を利用



推定値と不確実性(標準偏差)の関係

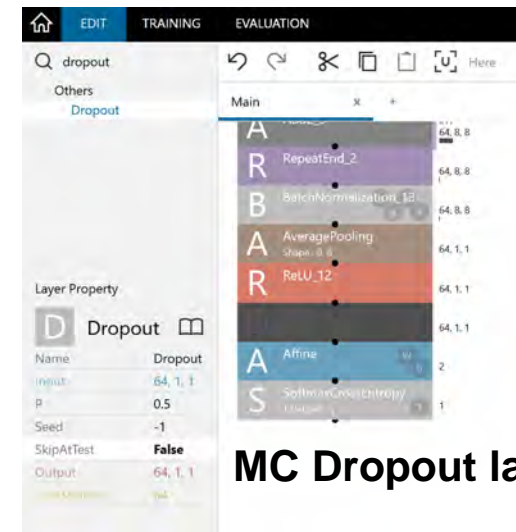
Relationship between estimates and uncertainties (standard deviation)



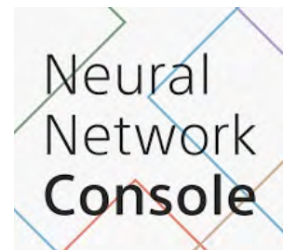
<犬と猫の画像分類>

<Dog and cat image classification>

不確実性は、推定値が0や1では無く、0.5に近いほど大きい



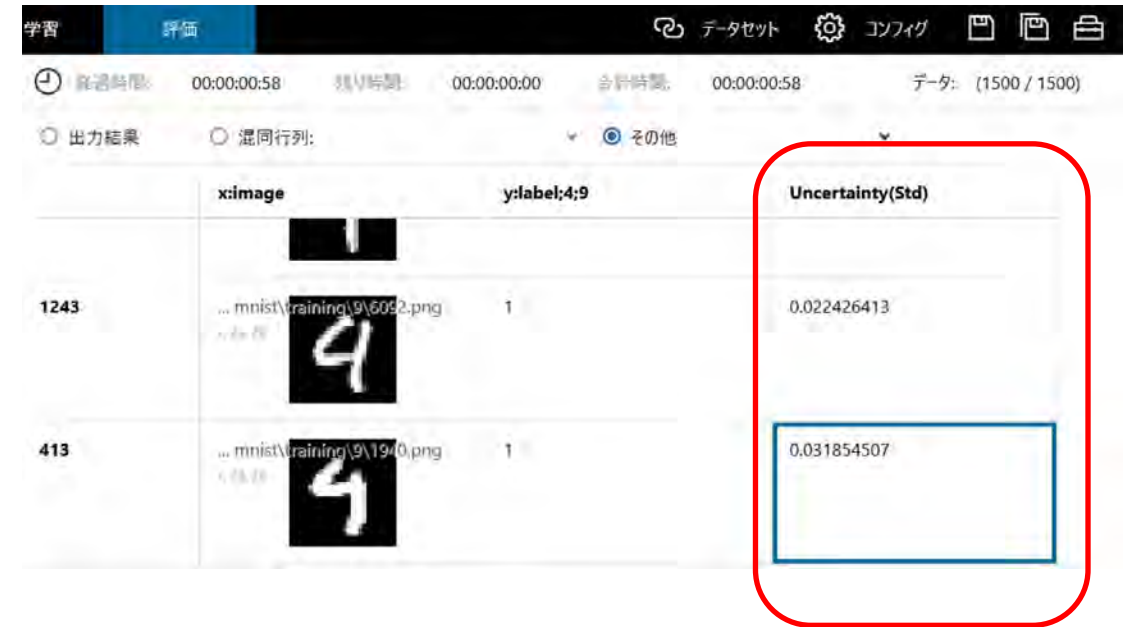
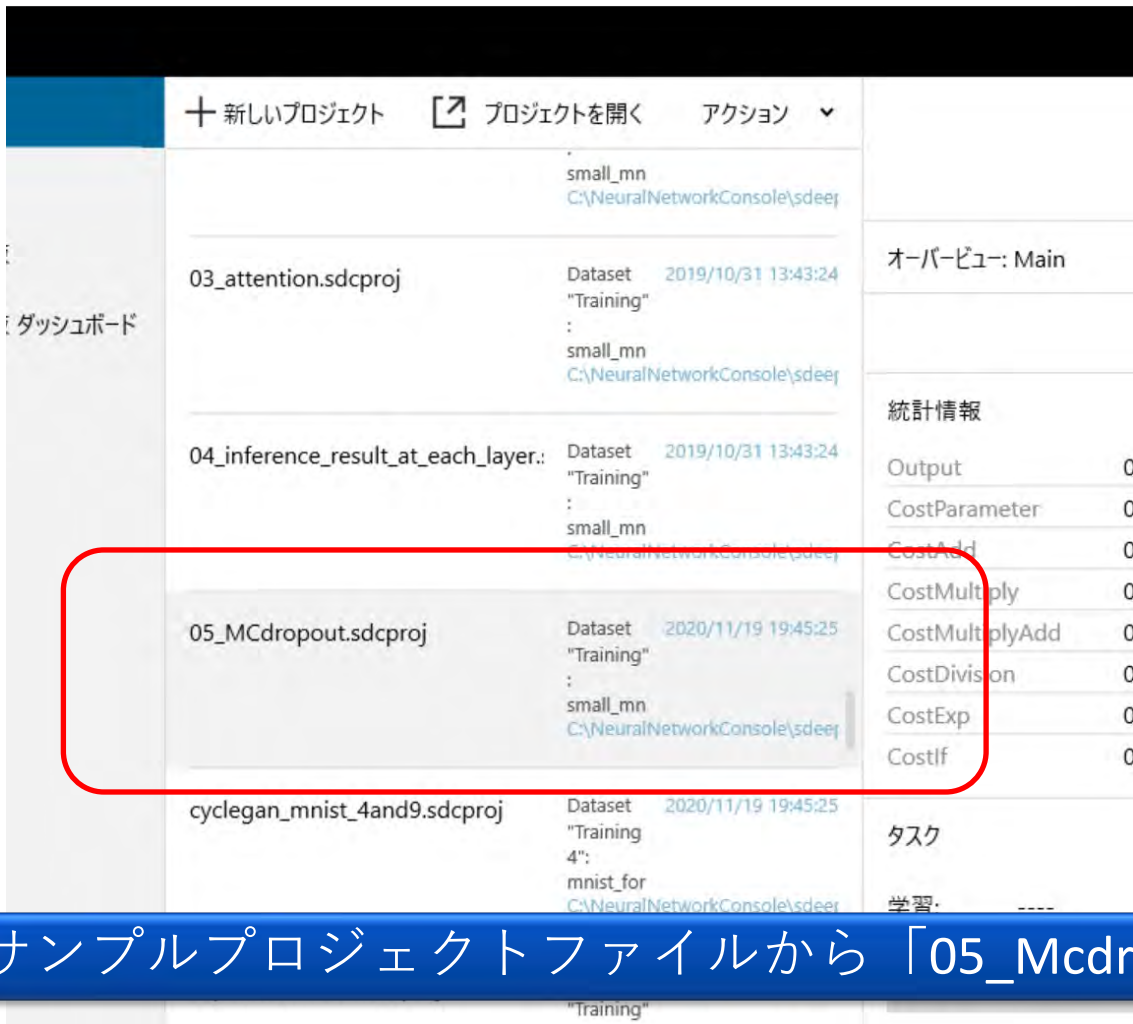
MC Dropout layer



Neural Network Consoleへの実装

# Neural Network Consoleでのデータの不確実性の使い方

- 学習と評価を実行



各データの不確実性(Uncertainty)が表示される

サンプルプロジェクトファイルから「05\_Mcdropout.sdcproj」を選択

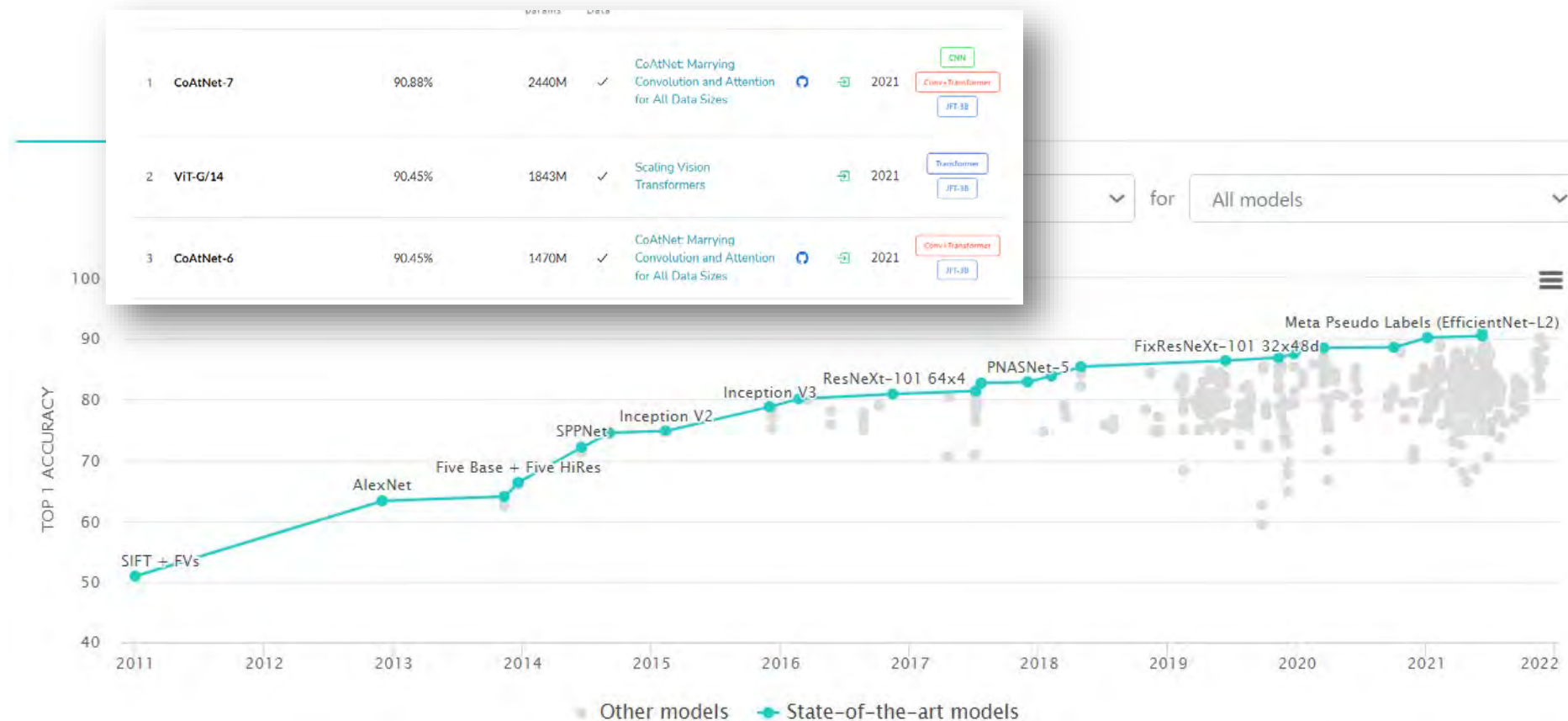


An aerial night view of a city, likely Tokyo, showing a dense network of lights and roads. The city is illuminated with warm yellow and orange lights, contrasting with the dark blue and purple tones of the sky and clouds. The perspective is from a high altitude, looking down on the city's layout. The text "近年の動向や課題" is overlaid in the center in a white, bold font.

# 近年の動向や課題



# Visual Transformer (ViT)の出現

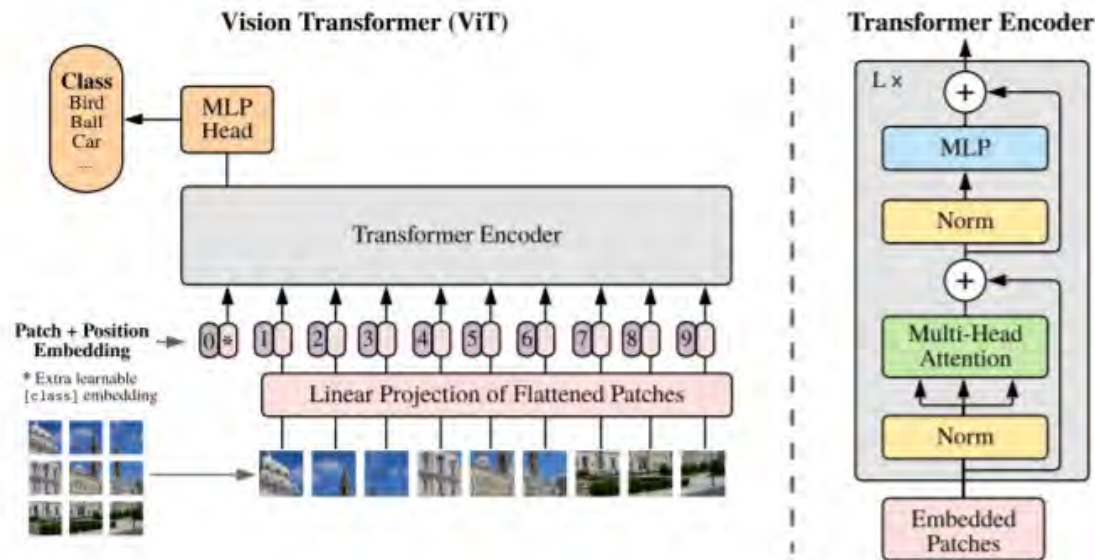


Visual Transformerは、ImageNetの画像認識精度を90%台へ

出典 <https://paperswithcode.com/sota/image-classification-on-imagenet>

# Vision Transformer(ViT)

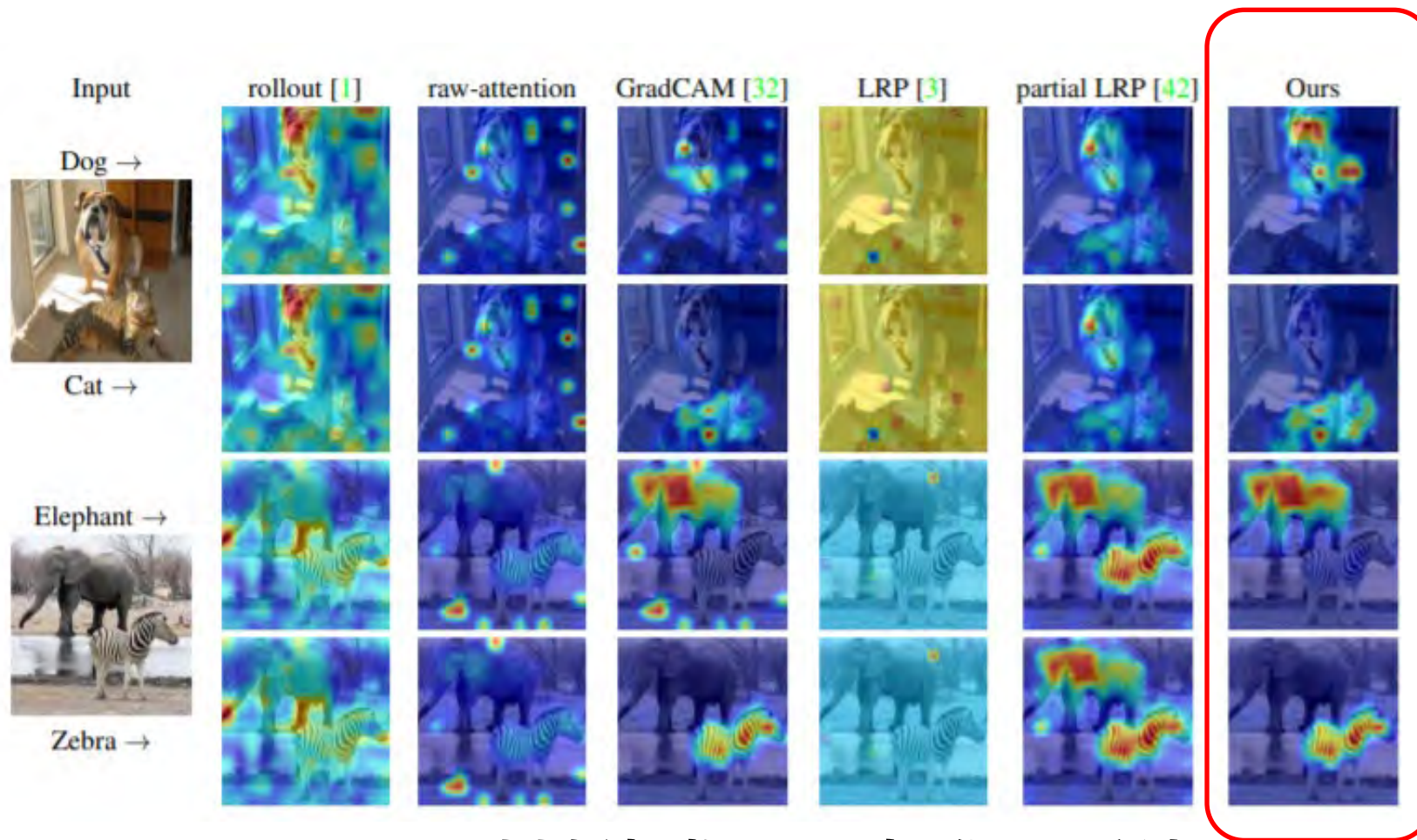
- Transformerは，大域的な空間情報を捉えることができる。
- 画像パッチを単語のように取り扱う。
- 巨大なデータセット(300M)で事前学習をする。



Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby  
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR 2021



# Vision Transformer (ViT) Explainability



## ViTの判断根拠の可視化の手法

Hila Chefer, Shir Gur, Lior Wolf, Transformer Interpretability Beyond Attention Visualization, CVPR 2021

# 説明可能なAIの課題

- ✓ 人間が理解できても、納得がいく説明になっているか。
- ✓ 様々な説明手法により結果が異なり、人間が理解できない。
- ✓ 様々なタスクに対して、どの説明可能なAIの手法を使うのが妥当か。
- ✓ 乱数パラメーターを利用する手法での再現性に注意。
- ✓ 計算コストが大きいケースもある。

**説明可能なAIへの過度な期待は禁物**



# 説明可能なAI ツール

# Neural Network Console による説明可能なAI



The screenshot shows the Neural Network Console interface. On the left, a menu is open with 'XAI' selected under the 'Visualization' category. The main area displays a table with columns for 'y' and 'gradcam'. The 'gradcam' column shows four visualizations labeled 'gradcam\_0000\_0.png' through 'gradcam\_0000\_3.png'. On the right, an 'Overview: Main' panel shows a list of layers and their corresponding statistics.

Layer	Statistics
Input	1,28,28
Convolution	16,28,28
ReLU	16,28,28
MaxPooling	14,28,14
Convolution2	32,14,14
MaxPooling_2	20,14,14
Tanh_2	20,14,14
Attnet	140
ReLU_2	150
Attnet_2	10
Softmax	10
Categorical Crossentropy	1

Statistics

Output	21,919
CostParameter	78,810
CostAdd	11,304

GUIによる簡単操作

Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf. arXiv:1708.07747

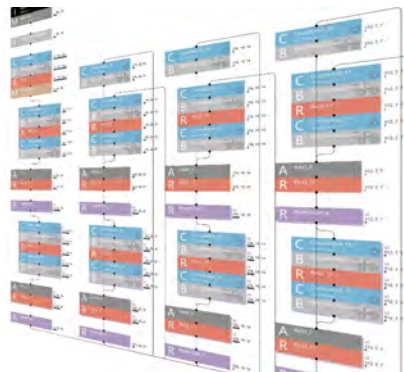
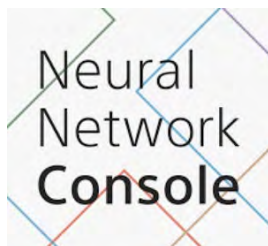
<https://dl.sony.com/ja/>

モデルの構築、学習、説明可能なAIまで、  
Neural Network Consoleにて一貫して開発できる。

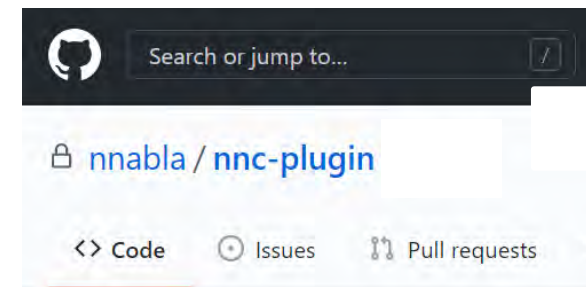


# 説明可能なAIのリリース形態

GUI



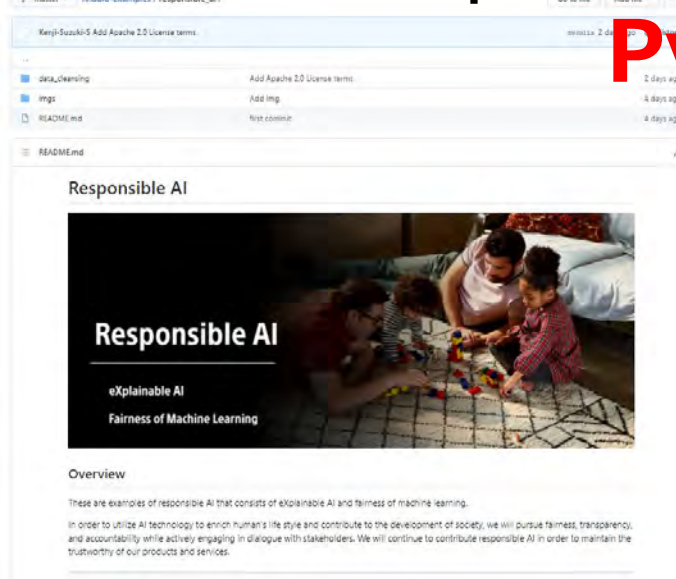
オープンソースソフトウェア



**説明可能なAI**

Neural Network Libraries

nnabla-examples

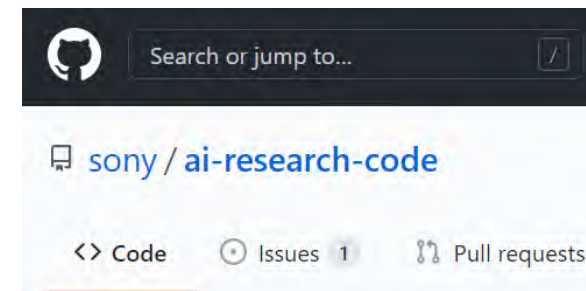


Python

eXplainable AI

Colab

Name	Notebook	Task	Example
Grad-CAM	<a href="#">Open in Colab</a>	Grad-CAM	



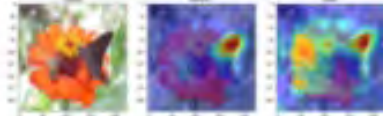


# Responsible AI ライブラリ

## 1. Visualization

### Grad-CAM code

Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization Ramprasaath R. Srinivasan, Michael Cogswell, Abhinav Dhall, Ramakrishna Velupillai, David Farnik, Dhruv Batra, arXiv technical report (arXiv:1610.02644)



### SHAP code

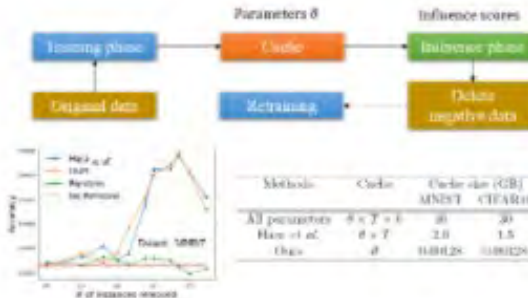
A Unified Approach to Interpreting Model Predictions Scott Lundberg, Su-In Lee, arXiv technical report (arXiv:1705.07875)



## 2. Influence

### Data cleansing with Storage-efficient Approximation of Influence Functions code

Data Cleansing for Deep Neural Networks with Storage-efficient Approximation of Influence Functions Koji Suzuki, Yoshiyuki Kobayashi, Takaya Naritoku, arXiv technical report (arXiv:2102.11607)



### Understanding Black-box Predictions via Influence Functions code

Understanding Black-box Predictions via Influence Functions Peng Wei Koh, Terry Yang, arXiv technical report (arXiv:1703.04743)

### TracIn code

Estimating Training Data Influence by Tracing Gradient Descent Galina Prutkin, Friedrich Liu, Mikael Sundström, Sergey Kalin, arXiv technical report (arXiv:2002.08180)

```
!pip install nnabla
!git clone https://github.com/sony/nnabla-example
%cd nnabla-example/responsible_ai/gradcam

import os
import cv2
import urllib.request
import numpy as np
import matplotlib.pyplot as plt
import nnabla as nn
from nnabla.utils.image_utils import imread
from nnabla.models.imagenet import VGG16
```



[https://github.com/sony/nnabla-examples/tree/master/responsible\\_ai](https://github.com/sony/nnabla-examples/tree/master/responsible_ai)

可視化、データ影響度、Colab形式のサンプルが充実

# 分かりやすいノートブック形式 Colabチュートリアル

gradcam.ipynb  
ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

+ コード + テキスト ドライブにコピー

Rectified Conv Feature Maps

CNN

FC Layer Activations

image classification

Car

Backward

ReLU

+

▼ Preparation

Let's start by installing nnabla and accessing [nnabla-examples repository](https://github.com/sony/nnabla-examples).

```
[ ] !pip install nnabla
[ ] !git clone https://github.com/sony/nnabla-examples.git
[ ] %cd nnabla-examples/responsible_ai/gradcam
```

Import dependencies

```
[ ] import os
[ ] import cv2
[ ] import urllib.request
```

eXplainable AI

Name	Notebook	Task	Example
Grad-CAM	Open in Colab	Visualization	
SHAP	Open in Colab	Visualization	

<https://github.com/sony/nnabla-examples>

# Neural Network Console 最新Pluginコード OSS公開

The screenshot shows the GitHub repository page for 'Plugins for Neural Network Console'. The repository is owned by 'TE-YoshiyukiKobayashi' and has 7 branches and 0 tags. The file list includes: .github/workflows, img, manuals, plugins, tools/scripts, .gitignore, LICENSE, and README.md. The 'About' section provides information about the repository, including the URL (https://dl.sony.com/), a README link, and the Apache-2.0 License. The 'Releases' section shows no published releases. The 'Packages' section shows no published packages. The 'Contributors' section lists Kenji-Suzuki-S, Kenji Suzuki, YukioObuchi, and Yukio Obuchi. The 'Languages' section shows Python at 99.4% and Shell at 0.6%. The main content area displays 'Plugins of Neural Network Console' with a 'Plugin' button and a list of available plugins such as SGD Influence (image), face evaluation, Grad-CAM, Grad-CAM (batch), LIME (image), LIME (image batch), LIME (tabular), LIME (tabular batch), SHAP, SHAP(batch), SmoothGrad, and SmoothGrad(batch).

<https://github.com/sony/nnc-plugin>

## How to use the latest plugins

The plugins run on Neural Network Console. If you do not have Neural Network Console, please download from here (<https://dl.sony.com/>).

1. Download the zip files from this repository.
2. Extract the zip files on your PC.
3. Delete the existing plugins folder. You can find it from `neural_network_console > libs > plugins`.

• NOTE If you do not want to turn off some plugins, please leave them.

4. Put the downloaded plugins folders in the same place, `neural_network_console > libs > plugins`.

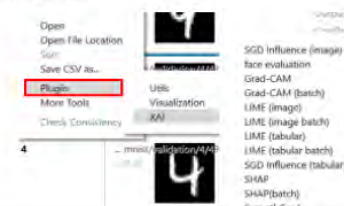
## Pre-processing

- To execute the plugins of the pre-processing, select the "DATASET" on the left of the top screen. Then click "Create Dataset", you can select the plugins of the pre-processing.



## Post-processing

- To execute the plugins of the post-processing, right-click the evaluation results on the Evaluate shortcut menu and select the plugins.

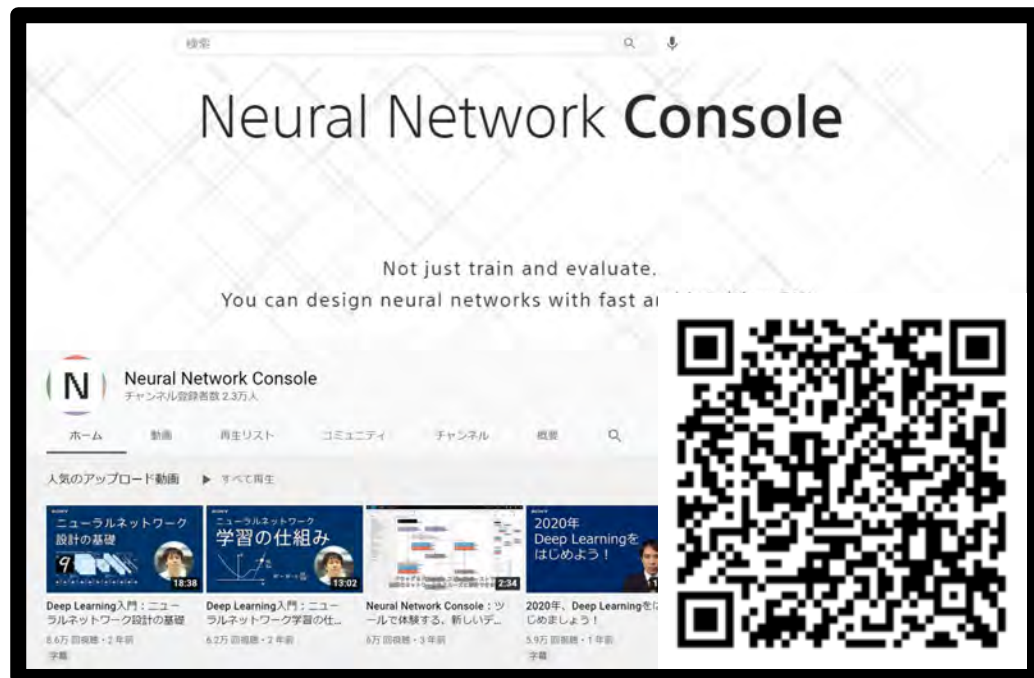


最新版のプラグインをOSS公開サイトからダウンロードして  
Neural Network Consoleで使える。



# YouTube 動画公開

New!!



- Neural Network Consoleのチュートリアル
- ディープラーニング技術の入門的解説
- 約**25,000人**のチャンネル登録者

<https://www.youtube.com/c/NeuralNetworkConsole>



- Neural Network Librariesのチュートリアル
- ディープラーニング技術の先端論文・技術解説
- **2021年3月に開設**後、約**2,500人**のチャンネル登録

<https://www.youtube.com/channel/UCOELxR-yS2EbjBxQ0hx4yBw>

## まとめ 「説明可能なAI」

説明可能なAIとは、人間がAIの判断理由を理解することができるようにする技術である。

社会的な要請により、説明可能なAIは、AI利活用に必要な技術である。

AI倫理のためだけではなく、AIの判断理由を明らかにすることにより、潜在的なAIの能力を引き出すことができる。

Neural Network Consoleにより、専門的な知識を必要とせず、説明可能なAIを手軽に利用することができる。



# SONY

SONYはソニー株式会社の登録商標または商標です。

各ソニー製品の商品名・サービス名はソニー株式会社またはグループ各社の登録商標または商標です。その他の製品および会社名は、各社の商号、登録商標または商標です。