

7/22成果発表会

第 10期「ソフトウェア品質保証部長の会」

AIシステムの品質保証

～AIモデルを組み込んだAIシステム開発の
プロセスと品質観点～

2019 ソフトウェア品質保証部長の会 第3グループ

梯 雅人	(株)日立製作所
小林 依光	KLab (株)
佐藤 孝司	日本電気(株)
永田 哲	(株)メタテクノ
廣石 高	三菱電機(株)
二川 勇樹	(株)モバイルインターネットテクノロジー
(五十音順)	

「AIシステム*の品質保証」目次

*: 本資料の「AIシステム」とは、AIモデルを組み込んだシステムとします

1. 活動の目的と方針
2. AIモデル(ディープラーニング)の特徴
3. AIの品質保証の難しさ
4. AIシステムの品質保証

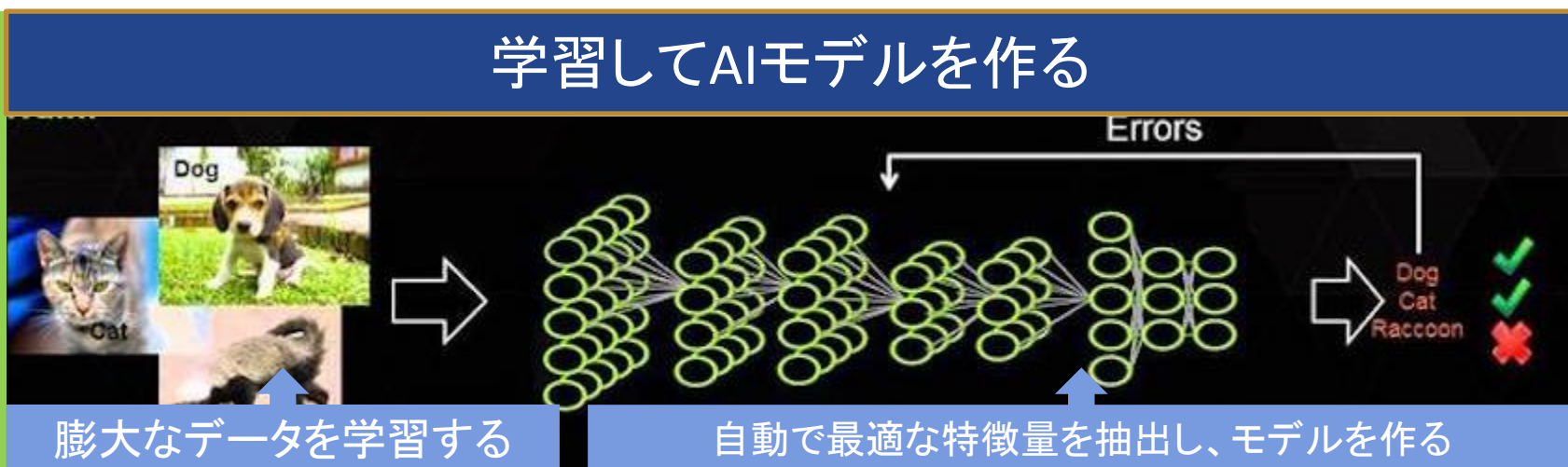
1.活動の目的と方針

目的	AIシステム開発の現場で使える品質保証の観点を抽出し、AIシステムの品質保証活動を支援可能とする
方針	AIシステムの品質保証活動のメタ観点として下記の二面で考える - AIシステムの品質保証をするために最低限やるべきこと - AIモデルが最大限の性能を得るためにやるべきこと
	第3次AIブームを牽引している「教師あり機械学習・ディープラーニング」を対象とする
手段	AIシステムの品質保証観点を抽出する - 実際に簡単なAIモデルを作成し、性能の改善を検証する過程で、具体的な品質観点を抽出する - AIモデルは、「手書き数字識別モデル」を利用する
活動の成果物	AIモデルを組み込んだAIシステム開発プロセスの品質観点チェックシート

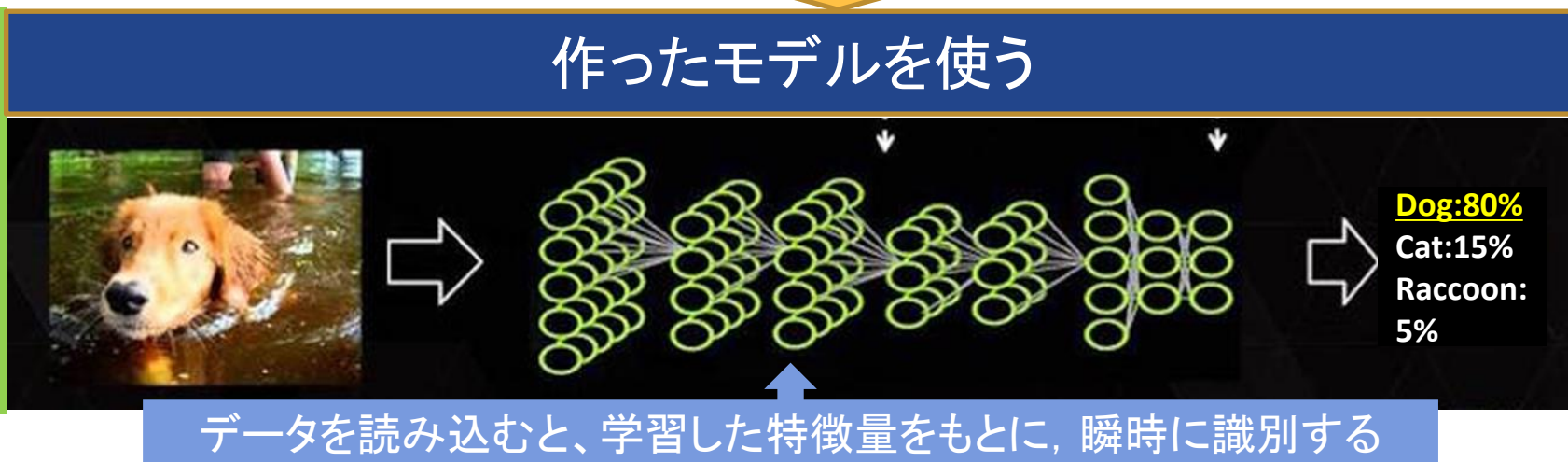
AIモデルの学習と推論

データを学習してAIモデルを開発【学習】、生成したモデルを実行【推論】

学習フェーズ (開発)

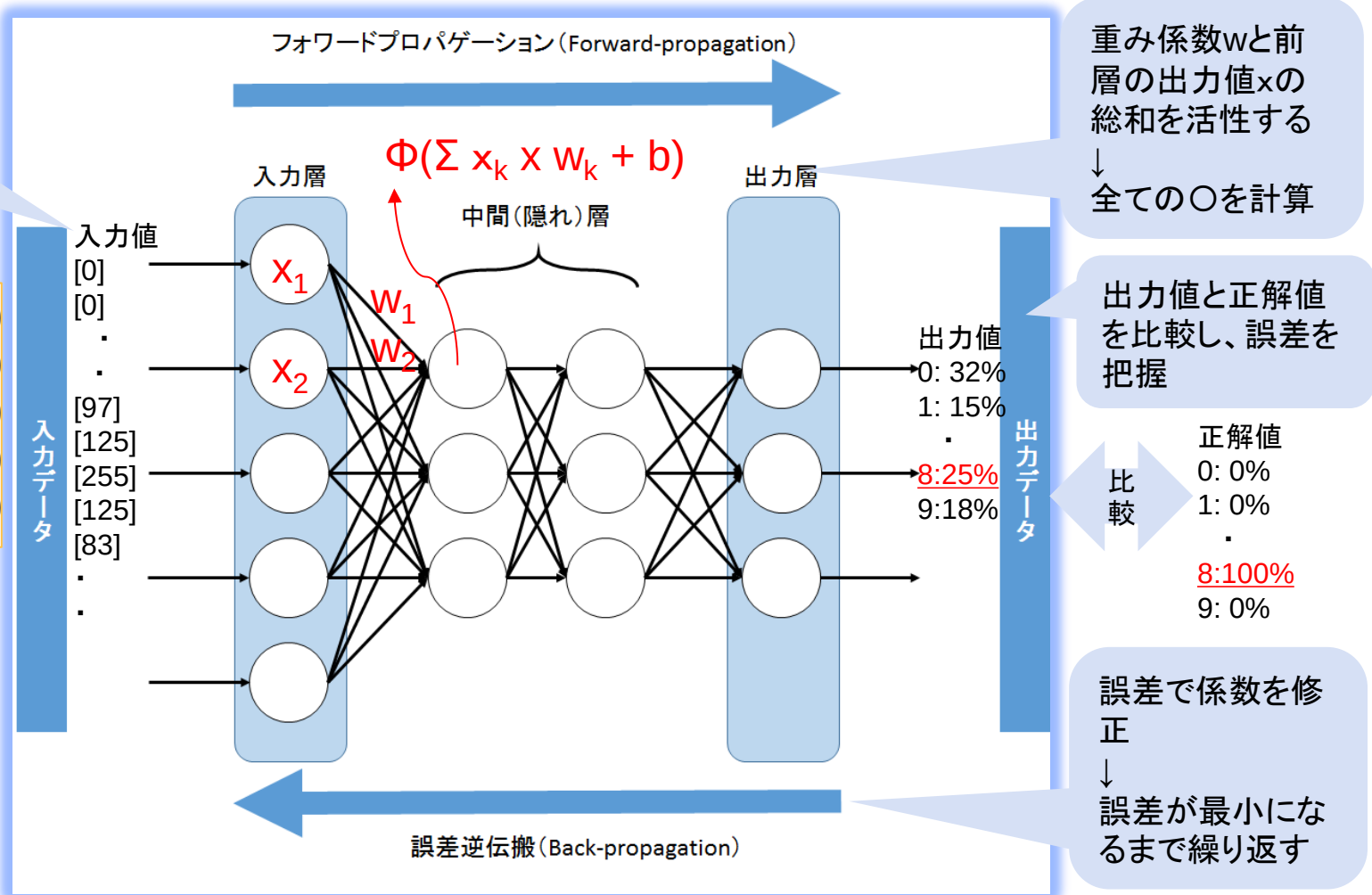


推論フェーズ (実行)



学習フェーズでは、何が生成されるのか？(1)

⇒特徴量を表す重み係数(Wn) (AIモデルが自動算出する)



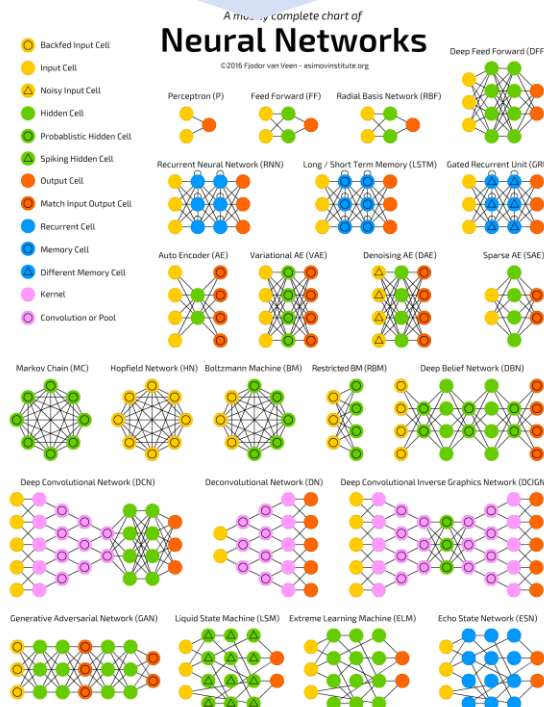
学習には確率的な手法(初期化,確率的勾配降下法,ドロップアウト等)が使われるため、
同じ結果が得られないという不確実さがある

学習フェーズでは何が生成されるのか？(2)

⇒モデル構造やチューニング値（人が決める）

①ネットワークアーキテクチャ選択

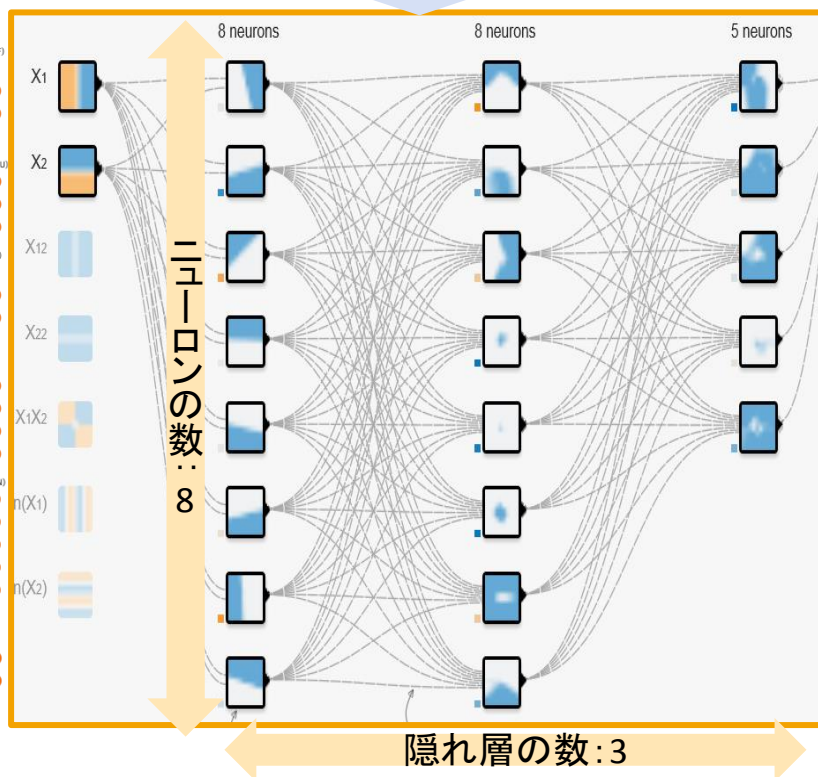
用途に最適なアーキテクチャを選ぶ



<https://postd.cc/neural-network-zoo/>

②モデルの構造を決める

ニューロンの数や隠れ層の数など



③ハイパーパラメータのチューニング

学習率、バッチサイズ、活性化関数の種類、正則化係数、学習イテレーション数、 옵ティマイザーの種類など

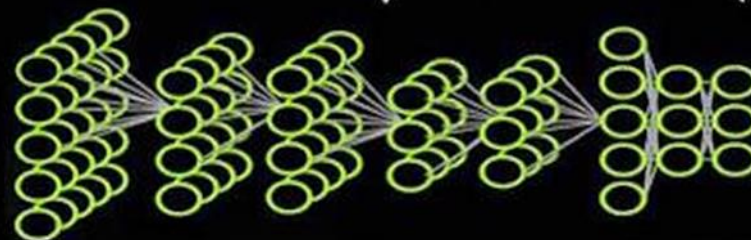
Pythonコーディング例(抜粋)

```
tf.keras.layers.Dense(512,activation=tf.nn.relu),  
tf.keras.layers.Dropout(0.2),  
tf.keras.layers.Dense(10,activation=tf.nn.softmax)])  
model.compile(optimizer=tf.keras.  
optimizers.SGD(lr=0.1)  
model.fit(x_train, y_train,  
epochs=5)
```

いろいろ試して最大の性能を得られる最適解を探す(帰納的開発)

確率的や帰納的な開発の品質保証はできるか？

“正解率が80%”の回答の根拠を説明できない

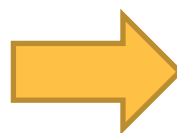


Dog:80%
Cat:15%
Raccoon:
5%

<https://fossbytes.com/hardware-architecture-deep-learning/>

運用時には、学習時の正解率が保証されるわけではない

学習時に正解率を最適にチューニング



推論(運用)時に、学習したデータとは
想定外のデータにより正解率は下がる



“絶対的正解”が無いので、正解との比較で品質保証をする考えは通用しない
⇒確率的・帰納的な動きが、“最適な結果であるか”を問う品質保証を考える

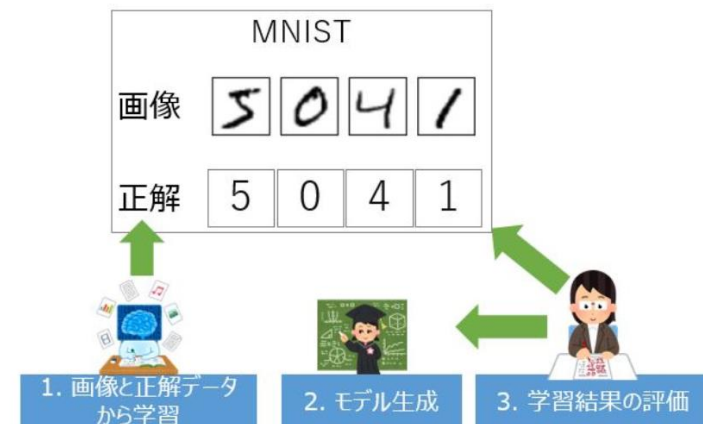
容易に開発できるので、開発者の裾野が広がる 複雑なロジックより、豊富なライブラリを容易に活用できる

容易に開発できるAIモデル開発環境の充実

8行の手書き数字識別モデル

<https://www.tensorflow.org/tutorials/>

- ① import tensorflow as tf
- ② mnist = tf.keras.datasets.mnist
- ③ (x_train, y_train), (x_test, y_test) = mnist.load_data()
- ④ x_train, x_test = x_train / 255.0, x_test / 255.0
- ⑤ model = tf.keras.models.Sequential([tf.keras.layers.Flatten(), tf.keras.layers.Dense(512, activation = tf.nn.relu), tf.keras.layers.Dropout(0.2), tf.keras.layers.Dense(10, activation = tf.nn.softmax)])
- ⑥ model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
- ⑦ model.fit(x_train, y_train, epochs = 5)
- ⑧ model.evaluate(x_test, y_test)



品質保証の観点では、

AI開発スキル差が拡大し、粗悪品が流出するリスクがあるため、
AI開発の受け入れ側にもAI技術のスキルが必要になる

AI品質保証の難しさと対策ポイント

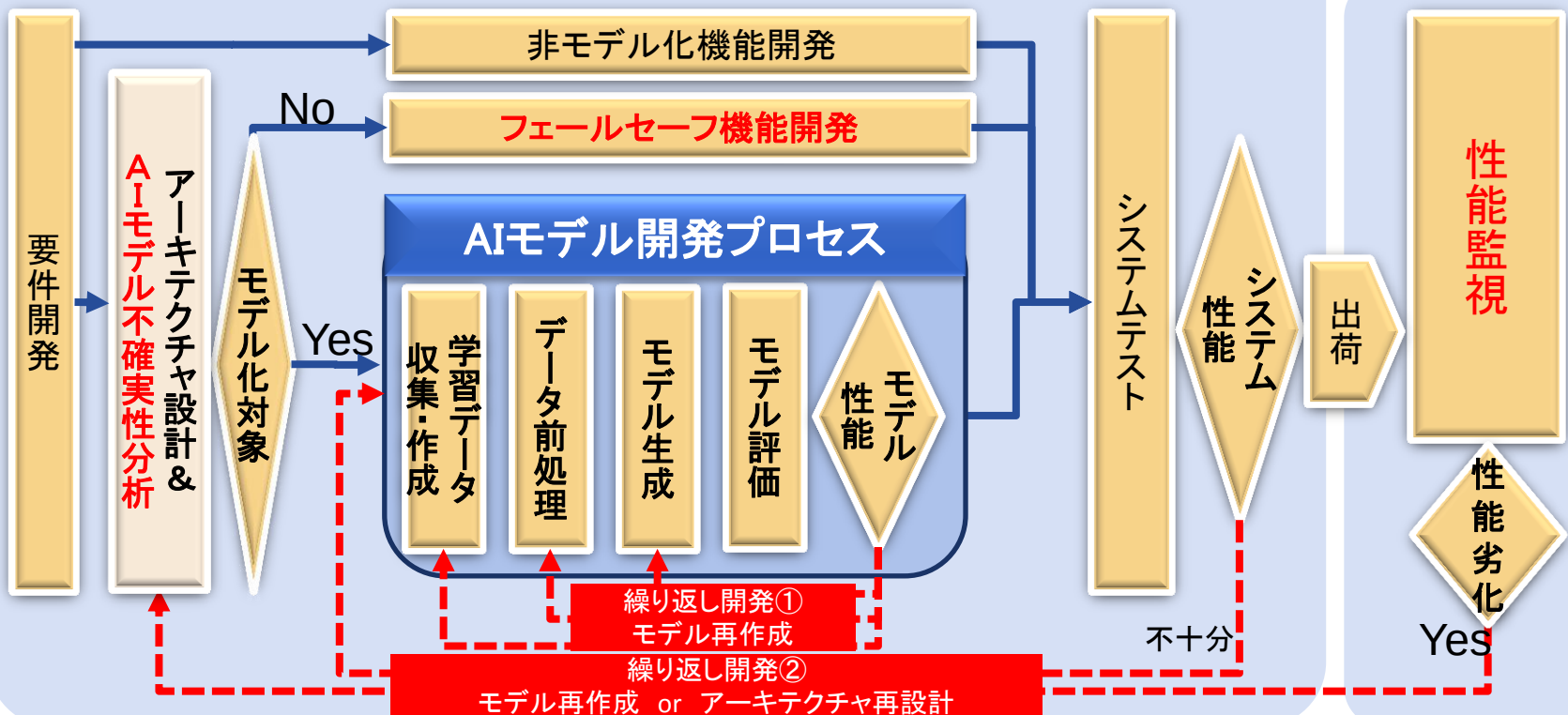
	AI品質保証の難しさ	対策ポイント	品質観点
1	学習データは 不完全である	学習データの網羅性を十分に設計したか	①データの品質観点
2	モデルの性能は、 試してみないと分からない	モデルの性能を最大限に引き出せたか	②AIモデルの品質観点
3	学習データは不完全である モデルは 誤判定する	データの不完全性やモデルの不確実性に対するシステム上の設計が考慮されているか	③AIシステムの品質観点
4	運用時の性能は、 実際に運用してみないと分からない	運用時の性能を監視し、モデルを再作成するプロセスが明確か	④運用時の品質観点

AIシステム開発プロセスと品質観点

AIモデルを組み込んだAIシステム開発プロセス

運用プロセス

開発プロセス



品質観点

①データの
品質

②AIモデルの品質

③AIシステムの品質

④運用
の品質

AIシステムの品質観点チェックシートのポイント

領域	ポイント
①データの品質	学習データの網羅性、モデル作成に十分な学習データの質と量を確認にする
②AIモデルの品質	モデルが最適な解を得るまで繰り返し開発したことを確認にする
③AIシステムの品質	データ網羅性の不完全さや、モデルの誤判定に対して、フェールセーフや障害解析性の機能により、システム全体でAI不確実性を補完できていることを確認にする
④運用時の品質	データの経年劣化やモデルの誤判定に対して、運用監視と障害対策により、運用時のAI不確実性を補完できていることを確認にする

①データの品質観点チェックシート

開発プロセス	品質観点	
要件開発	要件の不確実性	AI化の要件や実現可能性に対する不確実な要素を洗い出し、 お客様との合意 がとれているか？
アーキテクチャ設計 & AIモデル不確実性分析	学習データの不完全性	学習データの ケース を 網羅的 に洗い出したか？
		学習データの 対象外のケース や 準備不能なケースの対処 (システムのフェールセーフ設計や制約条件の明確化等)を設計しているか？
	モデルの不確実性	モデルの 誤判定時の対処 (システムのフェールセーフ設計や制約条件の明確化等)を設計しているか？

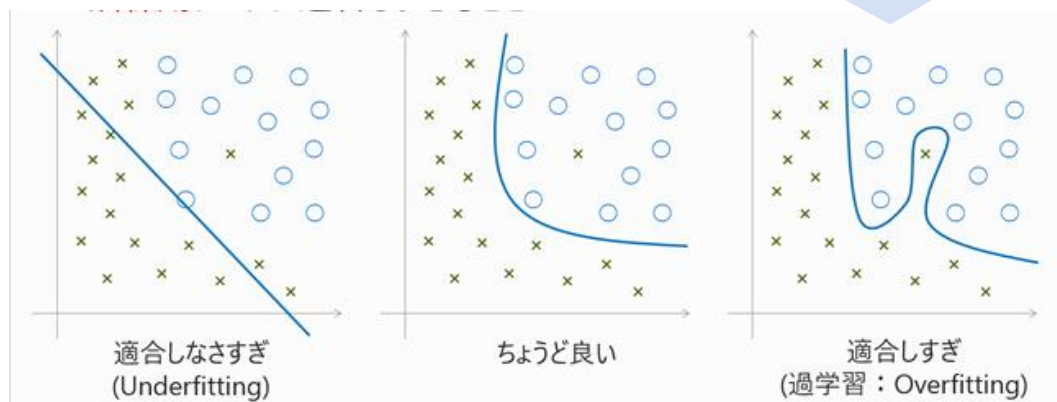
②AIモデルの品質観点チェックシート

開発プロセス		品質観点
学習データ 収集・作成	データ量	サンプリング率は妥当か？抽出領域の偏りはないか？
	データ質	- 不正解や異常値や欠損の割合は多すぎないか？ - 誤データがない、平均値や分散が異常でないか？ - データのラベルは正しいか？
	データ期間	日、月、季節、10年後の傾向等が影響を受ける場合に網羅しているか？
	データバリエーション	データの想定されるバリエーションの網羅性は十分か？
データ前処理	データオーグメンテーション	性能向上のためのデータ増幅をしているか？（画像データの場合では、反転・回転、色・明度・彩度の変更、拡大・縮小、合成、部分抽出など）
	異常データ前処理	異常データを処理をしているか？（画像データの場合では、異常なサイズ、異常な解像度、意味のない画像など）
モデル生成	アルゴリズム選定	- アルゴリズムの選定は、目的やデータ特性に合っているか？ - 複数のアルゴリズムを試した結果から、最適なアルゴリズムを選択しているか？
	パラメータの最適化	- 訓練データの割合、ノイズ、バッチサイズは最適か？ - モデルの構造（層やニューロンの数、構成（CNN, Dropout,）、正則化、正則化項）は最適か？ - 活性化関数（ReLU/tanh/sigmoid等）は最適か？ - 重み初期値の設定方法は最適か？ - オプティマイザー、学習係数は最適か？ - 検証データ率、学習回数（エポック数）は最適か？
モデル評価	テストデータ独立性	テストデータは学習データとは独立しているか？
	異常系/境界・限界系テストの実施	異常系や偏向のデータでも期待する結果が出ているか？（画像データの場合、サイズが異常に大きい／小さい、色がカラー／モノクロ、縦横比／解像度が異なる等）
	性能評価	正解率、適合率、再現率、F値、ROC曲線の各値は、期待する目標を達成しているか？
		適合率と再現率のバランスは要件特性と整合しているか？
		- 結果が期待と大きく外れているデータはあるか？ - 期待値の高い種類のデータの精度は高いか？
		繰り返しテストをしても性能は安定しているか？
	過学習の予防	交差検証、正則化、学習曲線の結果は妥当か？

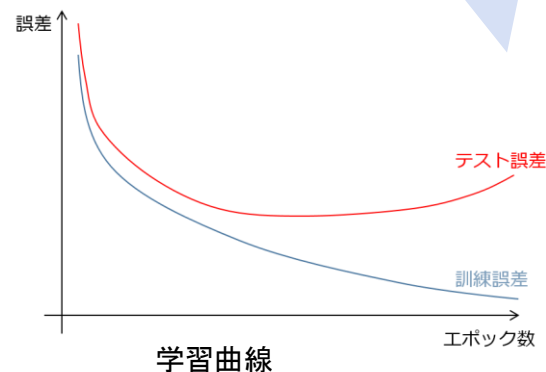
②AIモデルの品質観点の補足(1/2) ～過学習～

- 性能を求めすぎて、学習データに適合しすぎてしまう
→ 過学習（運用時のデータは訓練時とは別物）
- 過学習の予防 → 交差検証、正則化、学習曲線の確認など

学習データに適合しすぎ ⇒ 過学習



訓練とテストが乖離 ⇒ 過学習



<http://gihyo.jp/news/report/01/ptcon2017/0001>

③AIシステムの品質観点チェックシート

開発プロセス	品質観点	
非モデル化 機能開発	※モデルを組み込むシステム 部分の開発(モデルの入力・出 力・画面など)	従来のソフトウェア開発における品質を確保で きているか？
フェールセーフ 機能開発	※学習データの想定外や誤判 定を補完する部分の開発(不 確実性分析の結果、追加され る機能)	システムの安全を確保する機能(モデル代替機 能、性能監視、障害解析ログ設計等)が十分に 設計されているか？
繰り返し開発①	モデル性能の未達成リスク	モデル性能達成のための繰り返し開発を必要 以上に長期化させないために、 達成の見込みの 適切な判断プロセス があるか？
繰り返し開発②	システム性能の未達成リスク	システム性能達成のための繰り返し開発による 達成見込みの適切な判断プロセス があるか？
構成管理	プロセス公正性の担保	プロセス品質確保、係争時などの説明責任、運 用時バージョン切り戻しのため、 モデルを再現で きる環境や確証 (データ、モデル等の構成品や 評価結果)を残しているか？

③AIシステムの品質観点の補足(1/2) ～AIの誤認識と未認識～

■ AIの誤判定は、誤認識と未認識がある

- 誤認識: AIが認識すべきでないものを認識してしまう間違い⇒適合率を測定する
- 未認識: AIが認識すべきものを認識できない間違い⇒再現率を測定する

→これらを要件に適したバランスにすること

～未認識を許しても、誤認識は避けたい⇒疑わしきものは漏れなく認識したい～

混同行列		AIの判定	
		陽性Positive	陰性Negative
実際	陽性 Positive	真陽性 True Positive	偽陰性 False Negative (未認識)
	陰性 Negative	偽陽性 False Positive (誤認識)	真陰性 True Negative

適合率=真陽性/(真陽性+偽陽性)

再現率=真陽性/(真陽性+偽陰性)

③AIシステムの品質観点の補足(2/2) ～AIの誤認識と未認識～



ビン工場のAI検品

【不良品を漏れなくAIで識別し、さらに目視で検品するシステム】

正常品が混じってもよいが、不良品の見逃しはゼロにしたい

(誤認識は許すが、未認識は厳しく)

→再現率 > 適合率



高級イチゴのAI選別

【高級品をAIで判別し自動パックするシステム】

高級品を見逃すことよりも、傷ものを高級品と誤認識することはゼロにしたい
(未認識は許すが、誤認識は厳しく)

→適合率 > 再現率

※ただし、未認識が多いとムダが発生する別の問題になる

④運用時の品質観点チェックシート

開発プロセス	品質観点	
監視	性能劣化の監視	運用時の性能が劣化の判断とモデル再作成のプロセスは明確か？
		運用データは、想定した学習データの範囲内であるか？
		時間経過により最新の運用データに未追従が発生して性能が劣化していないか？
	オンライン学習の監視	継続的に入力データを学習して、モデルを更新・評価しながら運用を続ける場合、良い状態(性能向上、対象拡大)を維持できているか？
障害対策	誤判定時対策	・誤判定時の影響は想定内か？ ・誤判定時のフェールセーフ機能は働いているか？
	バージョン切り戻し対策	障害時や性能劣化に対して、古いバージョンモデルに切り戻しができるか？
	モデル切り換え対策	障害時や性能劣化に対して、別のモデルが準備しており、モデルの切り替えができるか？

さらに・・・⑤提供形態の違いによる品質観点

提供形態	品質観点
SI等の特定顧客 (1to1)	【開発の特徴】お客様と繰り返し開発結果を擦り合わせながら進める 【品質観点】 ・PoCでビジネス価値や実現可能性を判断できているか ・学習時と運用時のデータ差を分析できているか ・お客様の期待が過剰で、目標達成しないリスク対策を共有できているか 【品質領域】データの品質、運用時の品質
クラウドサービス (1toMany)	【開発の特徴】運用監視しながら、価値向上にむけて物件を置換 【品質観点】 ・PoCでビジネス価値や実現可能性を判断できているか ・運用時の問題発生時の切り戻し等のサービス継続性のリスク対策ができているか 【品質領域】運用時の品質
組み込み製品 (1toMany)	【開発の特徴】出荷時点で高い完成度が必要、あらゆる条件での利用 【品質観点】 ・対象とするケースの網羅性の分析ができているか ・対象外とするケースや、AIモデル誤判定に対するフェールセーフ設計などがシステム視点で考慮されているか 【品質領域】データの品質、システムの品質

■ AIの品質保証の難しさ

- 確率的や帰納的な開発であり、最適解かの見極めが難しい
- 開発者の裾野拡大により、最適でない開発物の納入リスク

■ AIの品質保証のポイント

- AIシステム開発の**プロセス定義**と**品質観点**の明確化
- データやAIモデルの不確実性を、**AIシステムとしての機能と運用プロセス**でいかにカバーできているか
- AI技術習得による**直接成果物**のチェックスキル向上

その他
には...

直接成果物の確認
・学習データの網羅性根拠
・性能やパラメータの推移記録

繰り返し開発の終
了判断は適切か

第三者による実運
用に近いテスト

メタモルフィック
テスト等
新しい技術適
用

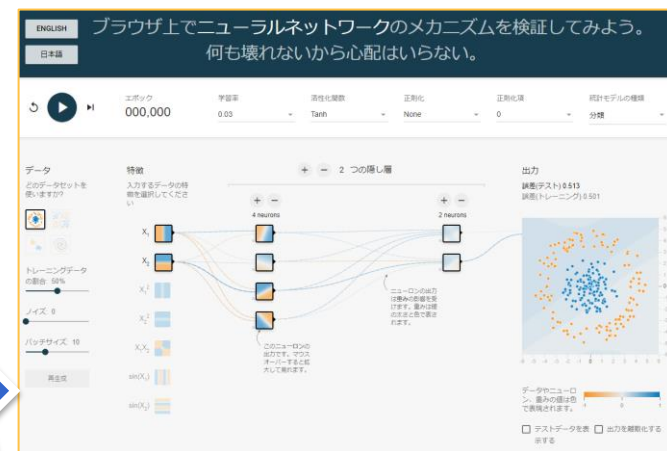
AIを触ってみよう！

“ブラウザ上でニューラルネットワークのメカニズムを検証してみよう。何も壊れないから心配はいらない”

<https://hinaser.github.io/Machine-Learning/raw/learntensorflowplayground/>

“8行の手書き数字識別モデルを実行してみよう”

<https://www.tensorflow.org/tutorials/>



Get Started with TensorFlow

TensorFlow is an open-source machine learning library for research and production. TensorFlow offers APIs for beginners and experts to develop for desktop, mobile, web, and cloud. See the sections below to get started.

Learn and use ML

The high-level Keras API provides building blocks to create and train deep learning models. Start with these beginner-friendly notebook examples, then read the TensorFlow Keras guide.

1. Basic classification
2. Text classification
3. Regression
4. Overfitting and underfitting
5. Save and load

[Read the Keras guide](#)

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(512, activation=tf.nn.relu),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

[Run code now](#)

[Try in Google's interactive notebook](#)

ご清聴ありがとうございました