

機械学習研修 Day2

(ハンズオン・「再帰化と機械学習」座学)

渡辺 龍二 / ペパボ研究所

2022.08.03

GMOペパボ

1. 研修についてのおさらい（数分）
2. ハンズオンの続きをやろう！（120 min）
3. 再帰化と機械学習（座学）（30 min）
4. 学んだことを各自まとめる（15 min）

機械学習研修の目的はこれだ！

サービス価値の向上のために「再帰化」を実現する上で、実装手段として機械学習を検討できるようになる。

- 目的の背景:ペパボのエンジニアとして機械学習を学ぶ動機は、
機械学習を「再帰化」の実現アプローチとして利用することにある
 - サービス価値の向上において「再帰化」は重要な考え方^[栗林2022]
 - 「再帰化 ≠ 機械学習の利用」であることを理解して
再帰化を実現するために機械学習の利用するか否かを判断できることを目指す

研修目的を達成するために以下のカリキュラムを設定しています！

研修の目的

サービス価値の向上のために「再帰化」を実現する上で、実装手段として機械学習を検討できるようになる。

| | ゴール | 実施内容 |
|------|--|--|
| Day1 | 機械学習とは何か、 自分の言葉で説明できる | <ul style="list-style-type: none"> - 機械学習イントロダクション(座学) - Machine Learning Crash Course (ハンズオン) |
| Day2 | 再帰化を実現するために、 機械学習をどう活用できるか イメージできる | <ul style="list-style-type: none"> - Machine Learning Crash Course (ハンズオン) - 再帰化と機械学習(座学) |
| Day3 | サービスの実課題を 機械学習で解決する 一連の流れの例を挙げられる | <ul style="list-style-type: none"> - Machine Learning Crash Course (ハンズオン) - 機械学習のサービス連携事例紹介(座学) |
| | | |

1. 研修についてのおさらい（数分）
2. **ハンズオンの続きをやろう！（120 min）**
3. 再帰化と機械学習（座学）（30 min）
4. 学んだことを各自まとめる（15 min）

ハンズオンを引き続きやっていきましょう！

- [Machine Learning Crash Course](#)
- 今日進めて欲しい単元
 - Generalization (15min)
 - Training and test sets (25 min)
 - Validation set (35 min)
 - Representation (40 min)

Machine Learning Crash Course with TensorFlow APIs

Google's fast-paced, practical introduction to machine learning, featuring a series of lessons with video lectures, real-world case studies, and hands-on practice exercises.

[Start Crash Course](#)

[View prerequisites](#)



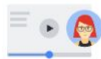
30+ exercises



25 lessons



15 hours



Lectures from Google
researchers



Real-world case studies



Interactive visualizations

進める上での注意点

- ハンズオン資料は英語です
- はあ？どゆこと？となったらすぐに聞いてください！
- 動画は飛ばしてください！
- 休憩を適宜取ってください
- 早く終わった場合はどんどん先に進んで大丈夫です！

1. 研修についてのおさらい（数分）
2. ハンズオンの続きをやろう！（120 min）
- 3. 再帰化と機械学習（座学）（30 min）**
4. 学んだことを各自まとめる（15 min）

再帰化と機械学習

渡辺 龍二 / ペパボ研究所
2022.08.03 機械学習研修Day2

GMOペパボ

趣旨

- ・ 再帰化と機械学習の関係について、一緒に考えていきましょう
 - ・ そのために「再帰化ってなんぞや…？」から確認していく
 - ・ 機械学習は再帰化を実現するアプローチの1つ、であることを確認する

アジェンダ

1. 再帰化とは何か？
2. 再帰化と機械学習の関係
3. 再帰化と機械学習の発展的な話題

1. 再帰化とは何か？

サービス価値の向上、という目的を達成する手段として「再帰化」がある

- ペパボは様々なユーザに対してwebサービスを提供している
- サービスを通して1人1人のユーザが得る体験の価値を高めたい
- Webサービスの主要な構成要素である情報システムが
様々なユーザに価値を感じてもらえる・満足してもらえるように振る舞ってほしい
そのための手段の1つが「再帰化」だと言える

1. 再帰化とは何か？ | 定義

再帰化の定義^[栗林2022]

構造化・自動化されたサービスが、
ユーザとのインタラクションの結果を取り込んで、
システムを自らより良いものへと改修していくプロセス

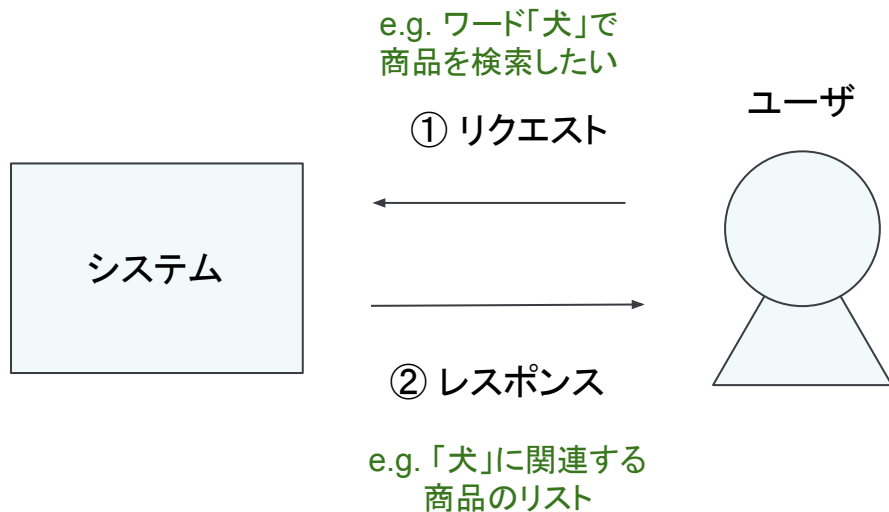
1. 再帰化とは何か？ | このセクションの進め方

再帰化の定義^[栗林2022]

構造化・自動化されたサービスが、
ユーザとのインタラクションの結果を取り込んで、
システムを自らより良いものへと改修していくプロセス

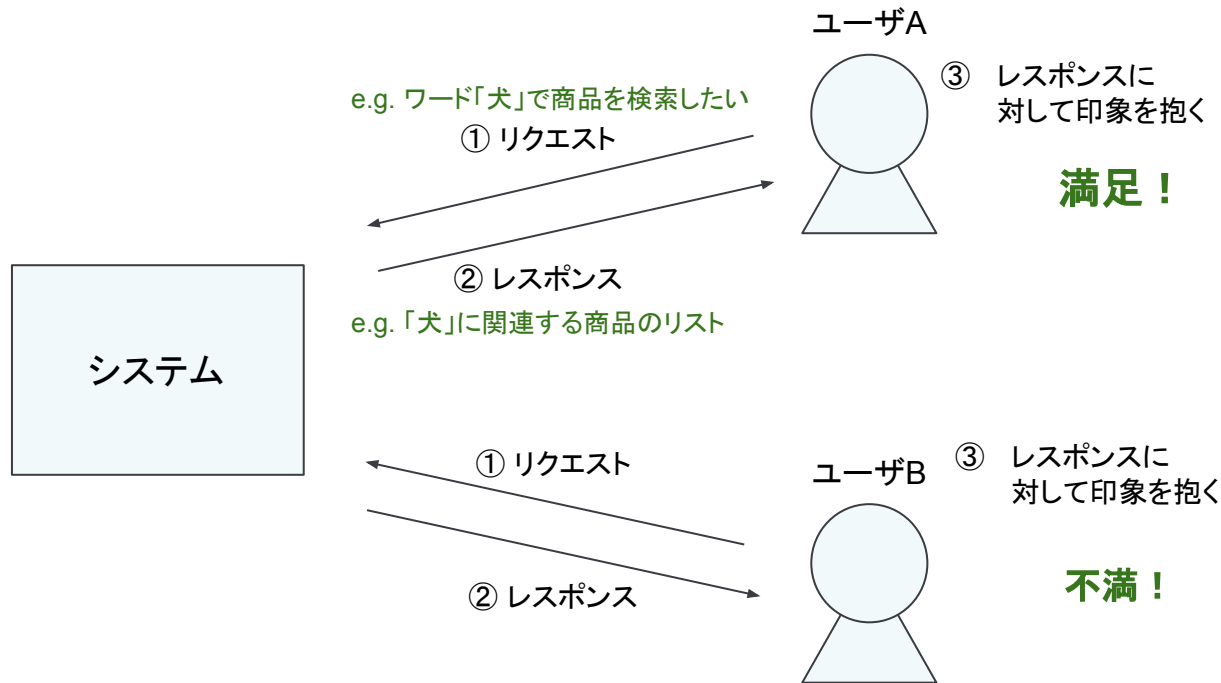
以降、この定義について理解を試みます

我々の運営するwebサービスは
ユーザからの要求に対してレスポンスするシステムによって構成されている



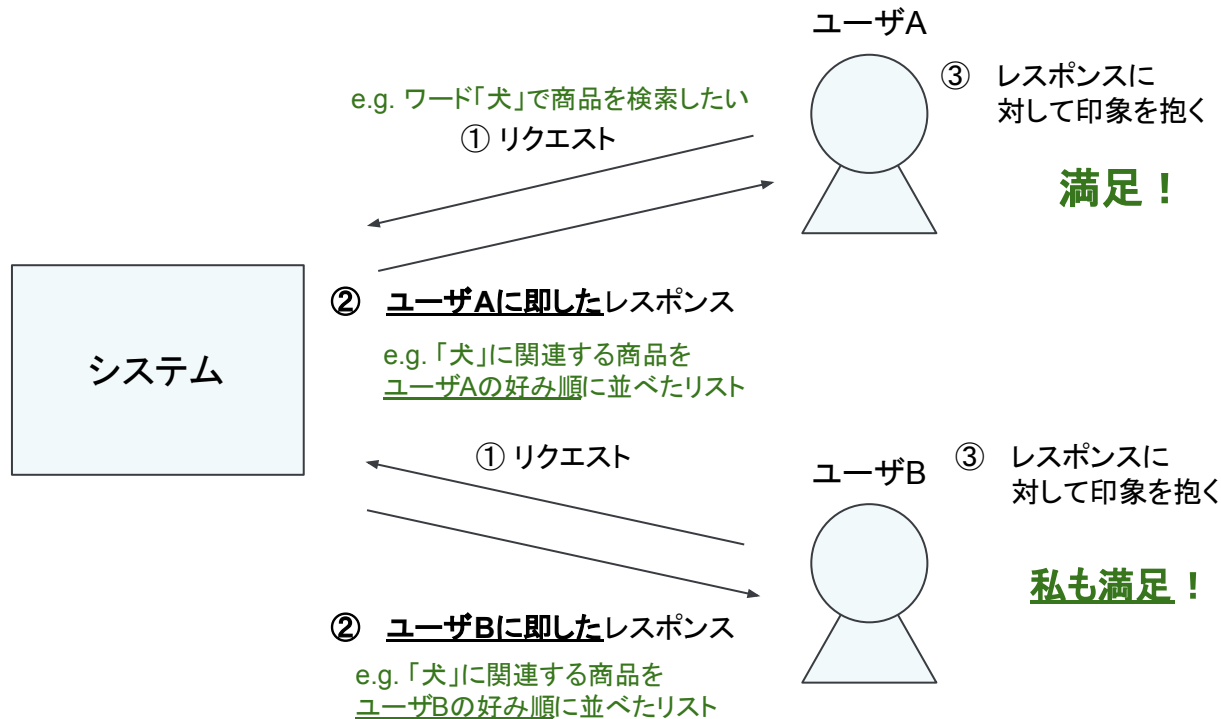
ユーザが変われば、ユーザが満足するレスポンスも変わる

e.g. 複数のユーザが同じリクエストを送り、同じレスポンスを返したとき

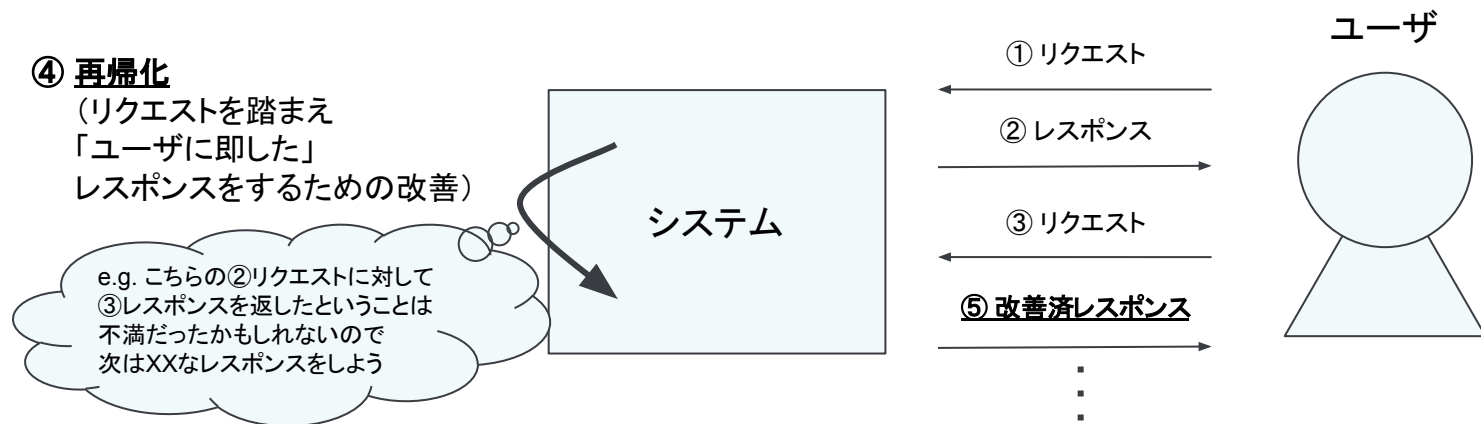


各ユーザに即して適切なレスポンスをするようなシステムを実現したい

e.g. 複数のユーザが同じリクエストを送るが、各ユーザに即したレスポンスをする場合



ユーザとのインタラクションの中でユーザの満足度合いを推し測りながらレスポンスを自ら改善させていく(再帰化する)システムを作りたい

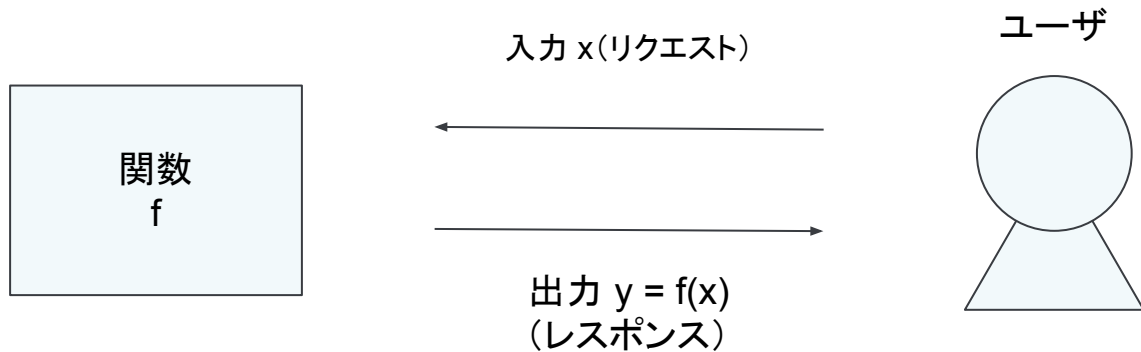


- このようなシステムを再帰化されたシステムと呼ぶ
- 再帰化したい理由は、理想のシステムを人手で作るのが大変かつ有効性が低い場合があるため(詳しくは後述)

2. 再帰化と機械学習の関係

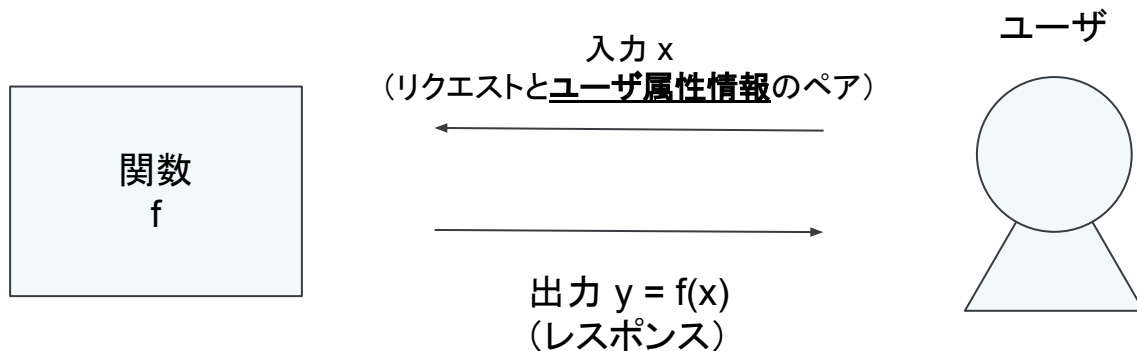
システムは「関数」と捉えることができる

- システムは、リクエスト(入力)に対してレスポンス(出力)を返す仕組みである
- これを抽象的に捉えると関数と見なせる



各ユーザが満足するレスポンスを返す関数を設計したい

- 例えば、この関数の入力として、リクエストだけでなく、ユーザの属性情報も加える
 - こうすると、ユーザの属性に応じて出力を変化させることができる



- このような関数fをあらゆるユーザが満足してくれるように設計したい

関数の設計を運用者が経験を元に行うのは有効性やスピードに劣る(可能性大)

運用者の経験に基づく設計の問題

- ユーザの要望を正確に反映することが難しい^[三宅2022]
 - 様々なユーザの多様性を認知し、
それぞれにどう対応するか設計することには限界がある
 - そのため、設計の際に最大公約数的な振る舞いを定義しがち

人手でやる設計の問題

- 時間がかかる
 - 属人化しやすい
- } スケールしにくい

人手での設計の問題に対しては、自動的・データ駆動な設計が有望である

運用者の経験ベースな設計の問題

- ユーザの要望を正確に反映することが難しい

有望な
打ち手
←

データ駆動な設計への期待

- ユーザの要望を正確に反映できる(かも)
データという客観的な事実を元に設計するため

人手でやる設計の問題

- 時間がかかる
 - 属人化しやすい
- } スケールしにくい

有望な
打ち手
←

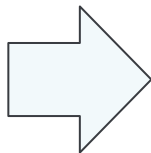
自動的な設計への期待

- 高速化
 - 属人化の抑制
- } スケールしやすい

データ駆動・自動的な関数の設計の実現方法として機械学習がある

機械学習

実現



データ駆動な設計への期待

- ユーザの要望を正確に反映できる(かも)
データという客観的な事実を元に設計するため

自動的な設計への期待

- 高速化
 - 属人化の抑制
- } スケールしやすい

※ あくまで実現方法の1つの選択肢

※ 機械学習を適当に導入すればAll OK! ではない
(ここまでの研修でよくご承知だと思いますが…)

機械学習(教師あり学習)はデータ駆動・自動的な関数の設計に利用できる

- ここまでの数学的な「関数」をプログラムにおける「関数」として見てみたとき…

通常のプログラミングによる関数の設計

関数の入出力関係を
人間(プログラマ)が
全て書き下す
(人間が書いたようにしか
動かない)



機械学習による関数の設計

データからアルゴリズムに
よって自動的に決定する^[丸山2021]
(人間が書き下さなくても
自動的に挙動が決まる)

3. 再帰化と機械学習の発展的な話題

関数の設計を継続的・自動的に更新していく仕組みが必要なのでは？

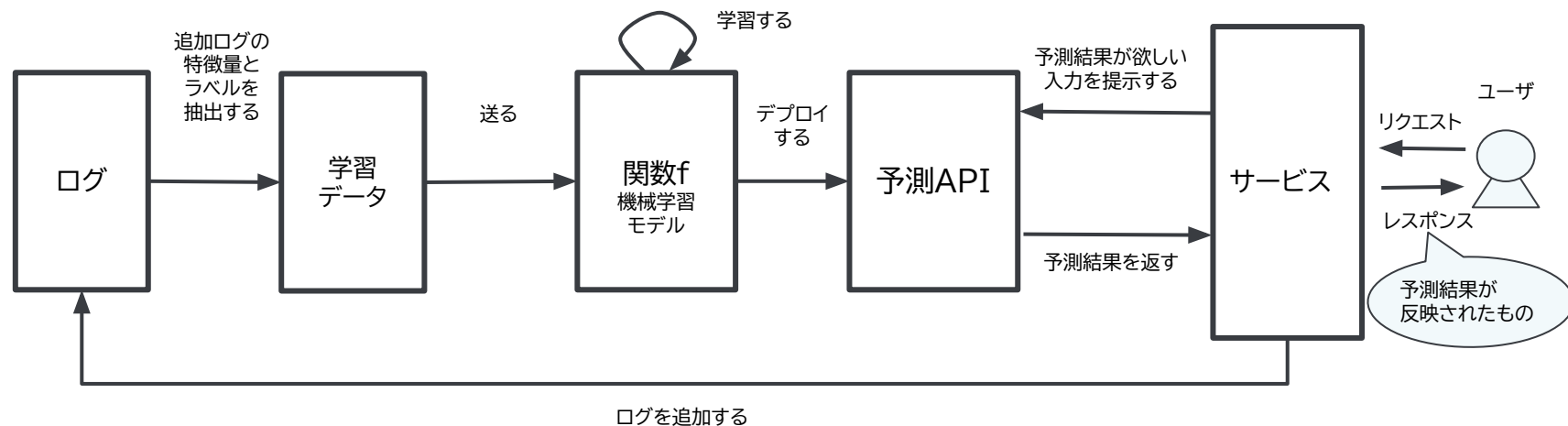
- ある時点で機械学習によって設計した関数がその後も有効とは限らない
 - システムの外側にある、利用者を含む環境は時間と共に変化する
 - 学習時の環境に最適化された関数を使い続けると、環境の変化と共に有効性が落ちる
 - 古典的な機械学習は、環境の変化を想定していないため
 - (イメージとしては内挿はうまくできるが外挿がうまくいかない、という感じ)
- よって、関数を継続的に設計しなおす必要がある
- この継続的な設計も自動化できることが望ましい
 - この仕組みをどう作るか？ということが

MLOps^[Mäkinen2021]領域や機械学習研究の領域^[Lange2021]で盛んに取り組まれている

[Mäkinen2021] Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?
<https://ieeexplore.ieee.org/abstract/document/9474355>

[Lange2021] A Continual Learning Survey: Defying Forgetting in Classification Tasks
<https://ieeexplore.ieee.org/abstract/document/9349197>

「関数の設計を継続的・自動的に更新していく仕組み」のイメージ



システムの改善のループはさらに多重に考えることができる

- 異なるタイムスケールでの改善のループを多重に考えることができる
 - 一重目: システムとユーザのインタラクションによる改善
(e.g. 機械学習の適用による自動的な関数の設計)
 - 二重目: 関数の設計を継続的に更新していく仕組み
 - 三重目: 二重目の仕組みを見直す、という改善のループ
- 再帰化とは、各レイヤーでの自動化に取り組んでいく試みと言える

1. 再帰化とはサービスを構成するシステムが
ユーザとのインタラクションの中で自らを改善していくプロセス
2. 再帰化にはデータ駆動・自動的な関数(システム)の設計が有望であり
そのアプローチの1つとして機械学習がある
3. 機械学習によって設計された関数の継続的・自動的な改善も必要であり
このような多重の改善もまた再帰化である

1. 研修についてのおさらい（数分）
2. ハンズオンの続きをやろう！（120 min）
3. 再帰化と機械学習（座学）（30 min）
4. **学んだことを各自まとめる（15 min）**

今日学んだことのまとめ・アウトプットの時間にします～

- Day2のゴール「再帰化を実現するために、機械学習をどう活用できるかイメージできる」は達成できそうでしょうか…？
- ぜひ以下のことをまとめてみてください～！
 - 「再帰化と機械学習の関係」について
 - 学んだこと、難しかったこと、考えたこと、思ったことなど