

深層学習アルゴリズムの 関手的構成の圏論的定式化

東北大学情報科学研究科 中村卓武

電気通信研究所 浅田和之, 中野圭介

背景

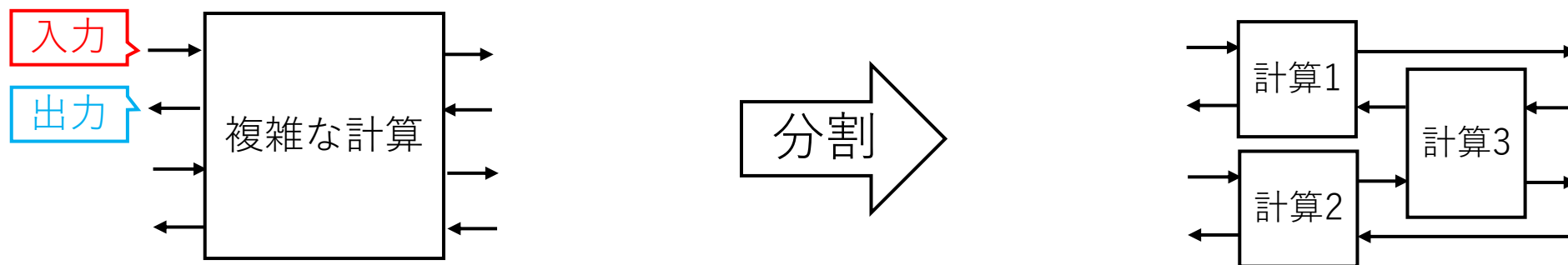
- 機械学習の学習手法は複雑で理論化が追い付いていない
 - 学習手法は既存手法の組み合わせによって提案されることも多い
 - 例：二つの独立したタスクとその学習手法を組み合わせる場合
 - 自然言語処理：単語をベクトルに変換＋ニューラルネットワークで処理
 - 二つのタスクで推論を分担
 - 画像生成：画像を生成＋画像が本物かを判定
 - 二つのタスクを（ゲーム理論的に）競わせる

目的と手法

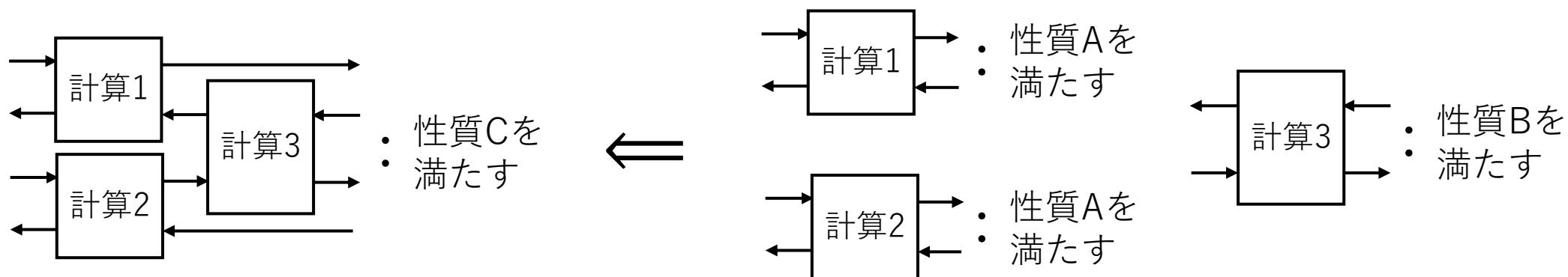
- **目的**：学習手法の**計算をモジュール化**
 - ⇒新規手法の**構成**がより簡単に
 - 計算の性質をモジュールに帰着
 - ⇒組み合わせさせた手法の**性質**を既存手法に帰着
- **方針**：既存手法やその組合せの圏論的定式化
 - **対象とする学習手法**：勾配に基づく学習（深層学習）

計算のモジュール化と性質の帰着

• 計算のモジュール化



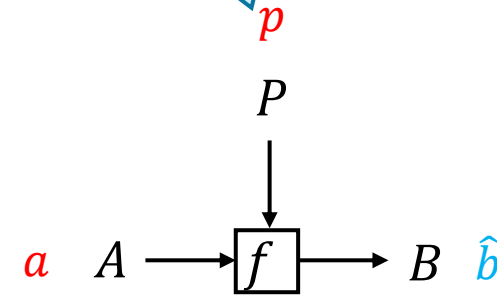
• 計算の性質のモジュールへの帰着



(勾配に基づく) 学習アルゴリズム


- 学習 (訓練) に必要な要素と用途
 - **訓練データ** : 入力 $a \in A$, 正解 $b \in B$, ($A = \mathbb{R}^n, B = \mathbb{R}^m$)
 - **アーキテクチャ** : 微分可能な実数値関数 $f: P \times A \rightarrow B$
 - $p \in P$: パラメータ
 - パラメータ p を用いて入力 a から **予測 $\hat{b} = f(p, a)$** を計算
 - **誤差関数** : $e: B \times B \rightarrow \mathbb{R}$ ($B = \mathbb{R}^m$)
 - 予測 \hat{b} と正解 b の間の **誤差 $L := e(\hat{b}, b)$** を計算
 - 例 : 二乗和誤差 $e(\hat{b}, b) := \frac{1}{2} \sum_i (\hat{b}_i - b_i)^2$
 - **訓練** : 誤差 L を減らすようなパラメータ p の探索
- 訓練方法 (勾配降下法) : **$p' := p - \nabla_p L$** と更新し探索
 - 勾配 $\nabla_p L := \left(\frac{\partial L}{\partial p_1}, \dots, \frac{\partial L}{\partial p_n} \right)$: 誤差 L を減らすための **p の移動量**

パラメータ関係の計算は上下方向に描画



これらの計算を中心にモジュール化したい

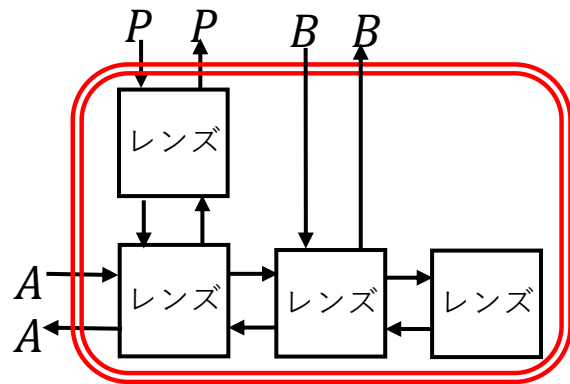
学習アルゴリズムの定式化

 : 学習アルゴリズム

- 学習アルゴリズムの圏論的定式化の二つの研究
 - いずれも**レンズ**を基本的な代数構造として使用
 - 学習アルゴリズムへの解釈方法が異なる

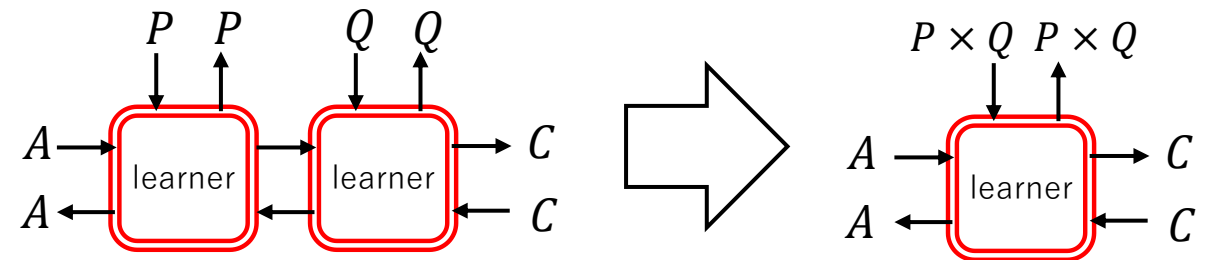
レンズ：順方向の関数と逆方向の関数の組

- **Cruttwellらの構成** [Cruttwell+ '23]
 - 学習アルゴリズムを**モジュール化**



- **Fongらの構成** [Fong+ '19]

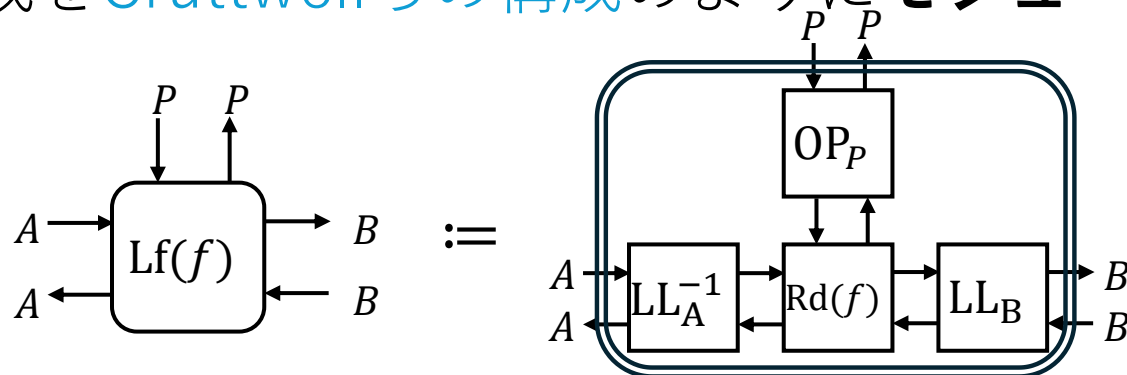
- **learner**：学習アルゴリズムの解釈を持つレンズ
- learner同士の合成でlearnerが得られる



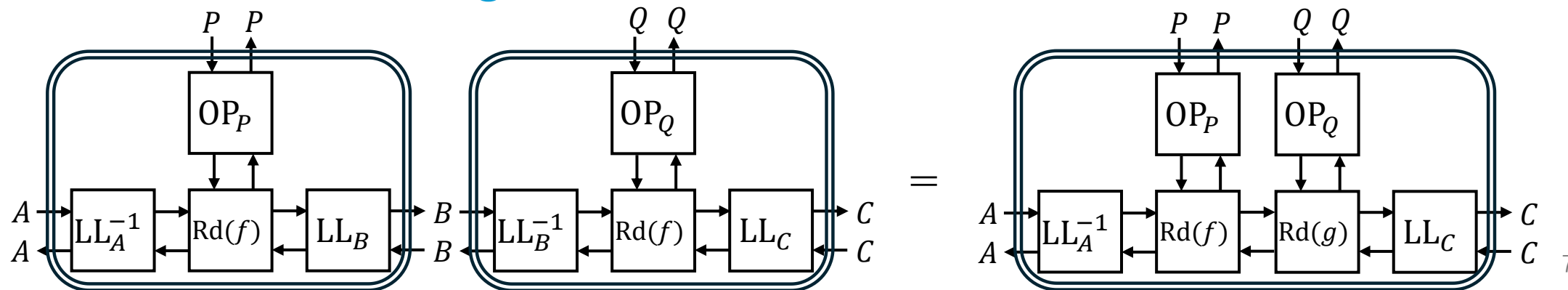
- **Cruttwellらの手法でFongらのlearnerをモジュール化する**

学習アルゴリズムの新規構成法

- learnerの新規構成法を提案
 - learner構成をCruttwellらの構成のように**モジュール化**



- learner同士をFongらの構成と同様に**合成**できる



新規構成法の応用

- 訓練データの正解が**ベクトル**ではなく**確率分布**である場合がある

- **Fongらの構成では学習が困難**

• モジュール化の応用

- 正解が確率分布のタスクでの learner 構成法を定義
- learner 構成の性質を **モジュールの(別の)性質へ帰着**
 - 関手性
 - GetPut 則
 - well-defined 性

Cruttwellらの
構成の長所

Fongらの
構成の長所

正解が ベクトル	モジュール 化の種類	学習の構成	微分の 一般化	最適化手法 の一般化	関手性	GetPut 則
Cruttwell+	学習構成	○	○	○	N/A	N/A
Fong+	learner	○	×	×	○	○
新規	両方	○	○	○	○	○

正解が 確率分布	モジュール 化の種類	学習の構成	微分の 一般化	最適化手法 の一般化	関手性	GetPut 則
Cruttwell+	学習構成	○	○	○	N/A	N/A
Fong+	learner	△	×	×	○	×
新規	両方	○	○	○	△	○

アウトライン

1. 既存研究
 - a. 圏論的な準備
 - b. Cruttwellらの構成の紹介
 - c. Fongらの構成の紹介
 - l. Fongらの構成の性質
2. 新規結果
3. 結果の応用

アーキテクチャの圏 [Gavranović '19, Gavranović '24]

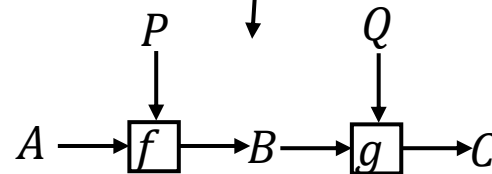
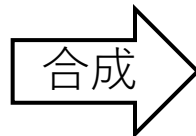
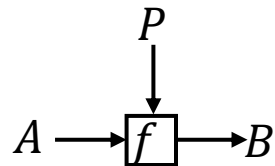
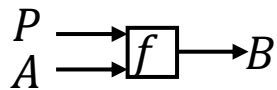
- **圏 Smooth** : $\text{Obj}(\mathbf{Smooth}) = \{\mathbb{R}^n \mid n \in \mathbb{N}\}$
 - $\mathbf{Smooth}(\mathbb{R}^n, \mathbb{R}^m)$: \mathbb{R}^n から \mathbb{R}^m への滑らかな関数の集合
 - 本発表では基本的に $\mathbf{C} = \mathbf{Smooth}$
- **Para構成** : SMC \mathbf{C} から SMC $\mathbf{Para}(\mathbf{C})$ を構成
 - $\text{Obj}(\mathbf{Para}(\mathbf{C})) = \text{Obj}(\mathbf{C})$ $\mathbf{Para}(\mathbf{C})(A, B) = \text{colim}(\mathbf{C}(\iota(-) \otimes A, B))$
 - 包含関手 $\iota: \mathbf{D} \rightarrow \mathbf{C}$ (\mathbf{D} は部分圏)

本研究の微分計算は Cartesian reversed differential category で一般化できる [Cockett+ '20]

アーキテクチャを $\mathbf{Para}(\mathbf{C})$ の射とみなす

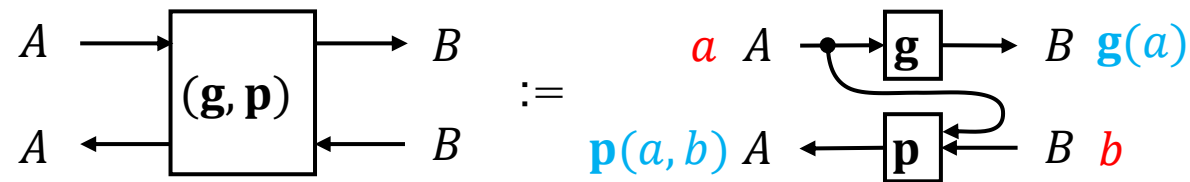
圏 \mathbf{C} $\mathbf{Para}(\mathbf{C})$
 射 $f: P \otimes A \rightarrow B$ $(P, f): A \rightarrow B$

射の図示



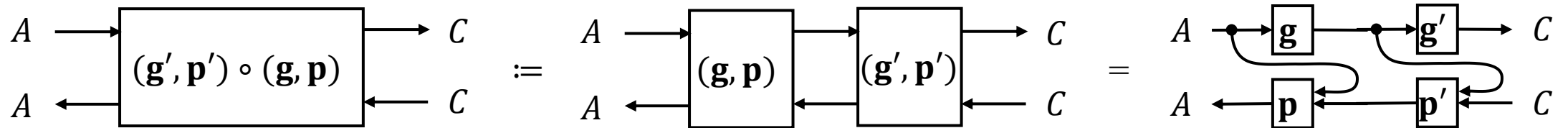
レンズの巻 [Foster+'07]

- レンズの巻 **Lens(C)** : **C**はcartesian **Obj(Lens(C)) = Obj(C)**
 - 射 **(g, p) : A → B** : レンズ (**C** の二つの射 **g : A → B**, **p : A × B → A** の組)
 - FongらとCruttwellらの構成で使用される基本的な代数構造
 - 次のように図示 [Boisseau+'23]



使用される理由：
誤差逆伝播法と相性が良い

- レンズの合成 :



⇒ 順方向・逆方向の計算の合成を同時に表現できる

アウトライン

1. 既存研究
 - a. 圏論的な準備
 - b. Cruttwellらの構成の紹介
 - c. Fongらの構成の紹介
 - l. Fongらの構成の性質
2. 新規結果
3. 結果の応用

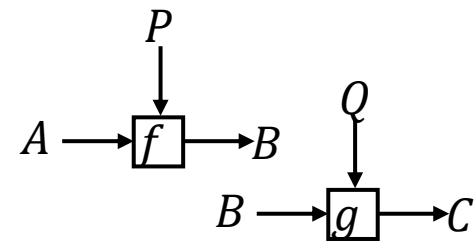
Cruttwellらの学習アルゴリズム構成法

- Cruttwellらの学習アルゴリズムの構成法
 - 勾配に基づく学習で慣用されている構成の圏論的一般化
 - アーキテクチャを構成→学習アルゴリズム化

• モジュールによって学習アルゴリズムを構成

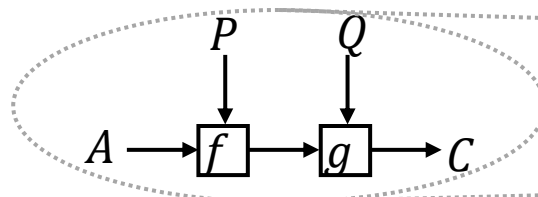
- **モジュール**：レンズで表される機械学習アルゴリズムの各計算
- **モジュールの結合**：レンズの合成

• アルゴリズム同士を合成できない



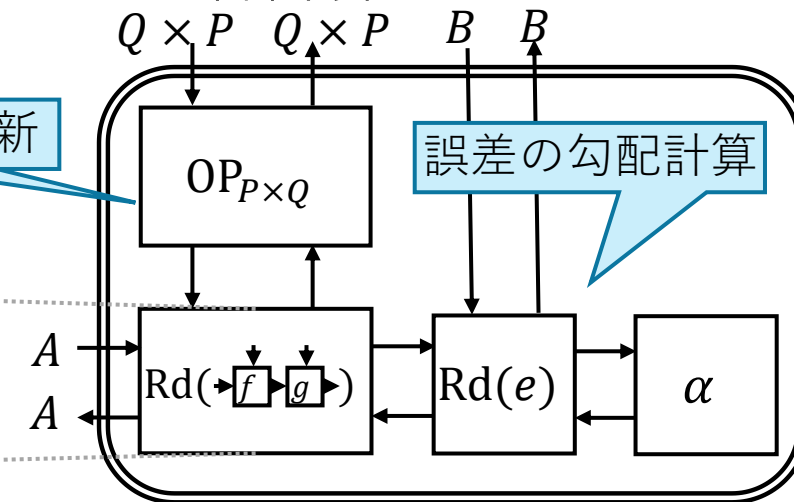
アーキテクチャ

$$(P, f) : A \rightarrow B, (Q, g) : B \rightarrow C$$



アーキテクチャ

パラメータ更新



学習アルゴリズム

アウトライン

1. 既存研究
 - a. 圏論的な準備
 - b. Cruttwellらの構成の紹介
 - c. Fongらの構成の紹介
 - l. Fongらの構成の性質
2. 新規結果
3. 結果の応用

learner [Fong+ '19]

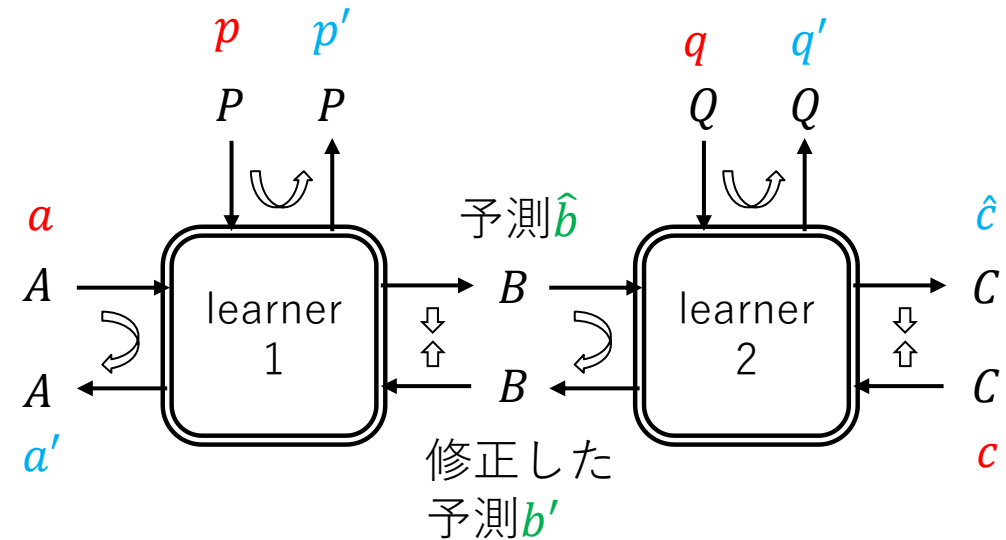
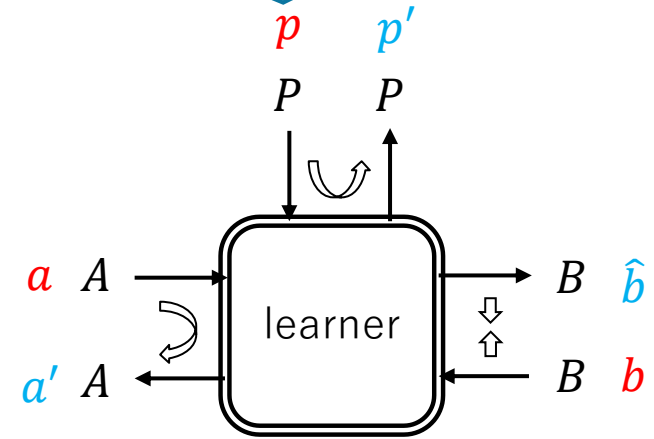
- learner : **Para(Lens(C))** の射 $(P, (\mathbf{g}, \mathbf{p})) : A \rightarrow B$
 - Fongらの構成における抽象化された学習アルゴリズム
 - $\mathbf{g} : P \times A \rightarrow B$: パラメータ p と入力 a から予測 \hat{b} を計算
 - $\mathbf{p} : P \times A \times B \rightarrow P \times A$:
正解 b を基にパラメータ p と入力 a を修正 (更新)

入力 a の修正はFongらの構成特有

- learnerの合成 : **Para(Lens(C))** の合成
 - 訓練 (put関数) では次の順序で計算する
 1. learner1 : \hat{c} を c に近づけるように q, \hat{b} を修正
 2. learner2 : \hat{b} を b' に近づけるように p, a を修正

learner2の主張 : \hat{b} ではなく b' を使えば \hat{c} が c に近づく

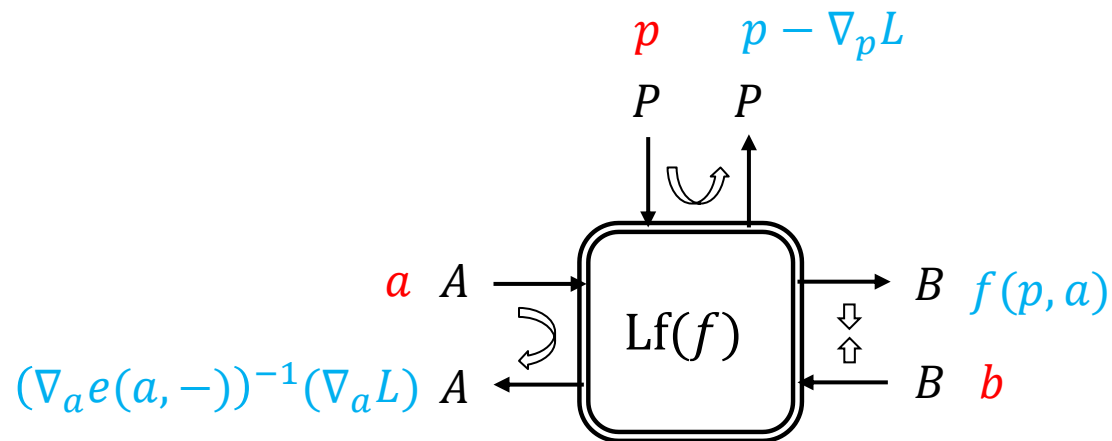
レンズのパラメータも
上下方向に描画



Fongらのlearner構成 [Fong+ '19]

- アーキテクチャ $(P, f): A \rightarrow B$ から learner $\text{Lf}(P, f): A \rightarrow B$ を構成

- **get関数** : パラメータ p と入力 a から予測 $\hat{b} = f(p, a)$ を計算
- **put関数** : 正解 b を基にパラメータ p と入力 a を修正 (更新)
 - p の修正 : 勾配降下法 $p' := p - \nabla_p L$ $L := e(\hat{b}, b)$
 - a の修正 : $a' := (\nabla_a e(a, -))^{-1}(\nabla_a L)$
 - 例(誤差関数が二乗和誤差) : $a' = a - \nabla_a L$

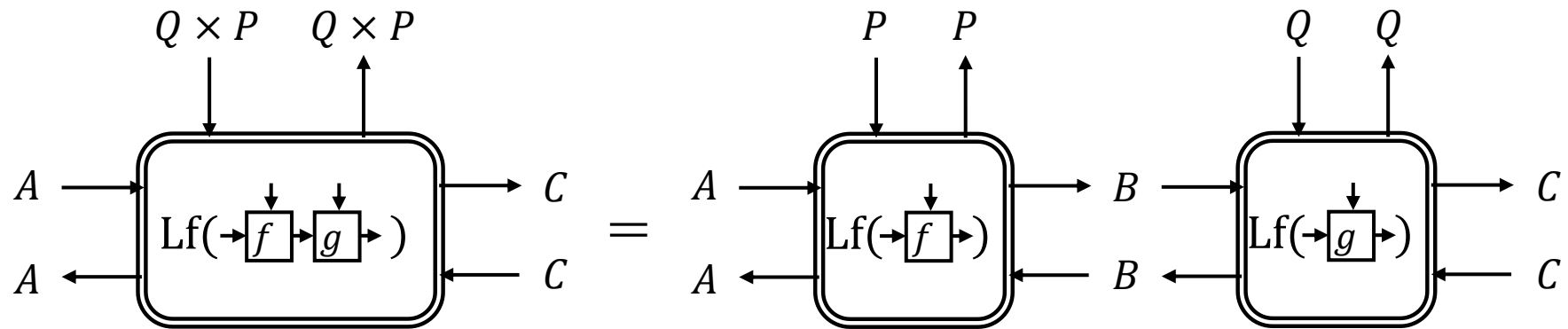


アウトライン

1. 既存研究
 - a. 圏論的な準備
 - b. Cruttwellらの構成の紹介
 - c. Fongらの構成の紹介
 - l. Fongらの構成の性質
2. 新規結果
3. 結果の応用

Fongらの構成の性質：関手性 [Fong+ '19]

- Lfは関手性 ($Lf: \mathbf{Para}(\mathbf{C}) \rightarrow \mathbf{Para}(\mathbf{Lens}(\mathbf{C}))$)を持つ

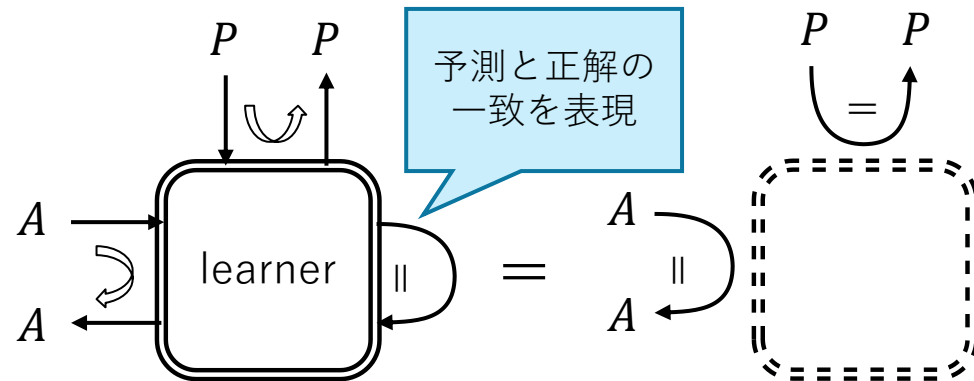


⇒アーキテクチャの結合よりもlearnerの合成の方が汎用的

- 例：全く異なるメカニズムで動作するlearner同士の合成が可能

Fongらの構成の性質：GetPut則 [Fong & Johnson '19]

- Lfで構成したlearnerは**GetPut則**を満たす
 - レンズにおける**代数的な**性質
 - 例 (learner) : 予測 \hat{b} と正解 b が一致 \Rightarrow 入力 a とパラメータ p を修正しない
 - **パラメータ更新の収束性**に関して望ましい性質



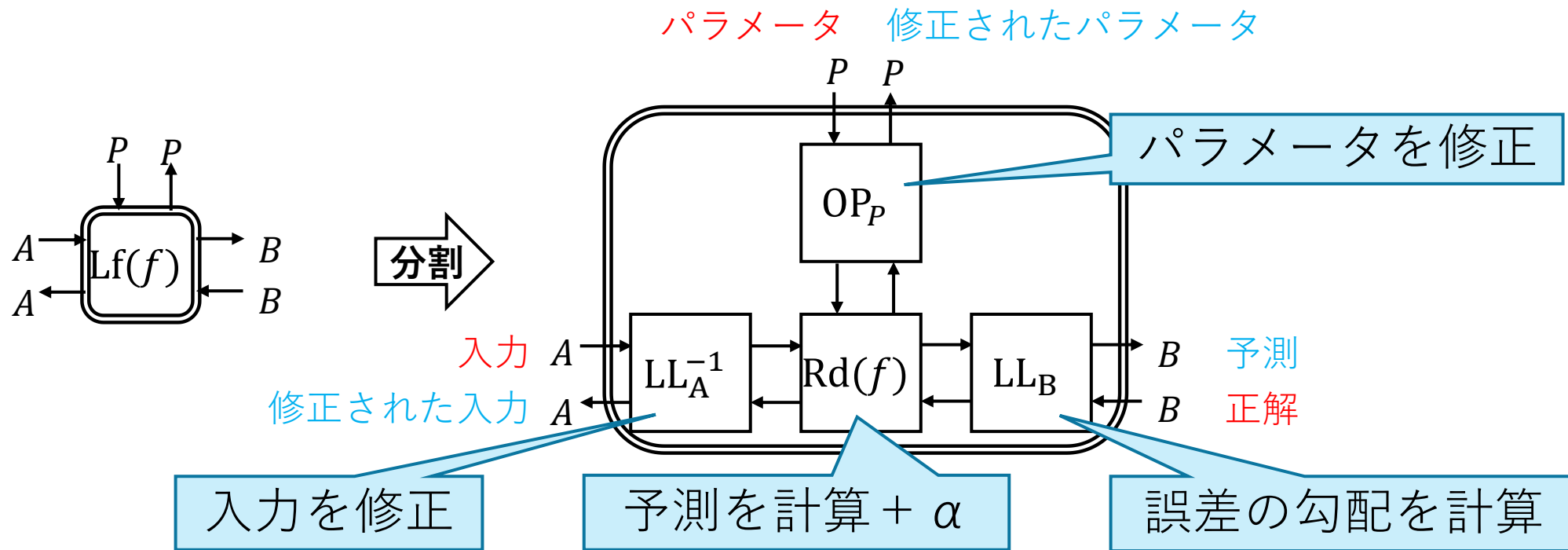
- ただし適切な誤差関数が必要 (例：二乗和誤差)

アウトライン

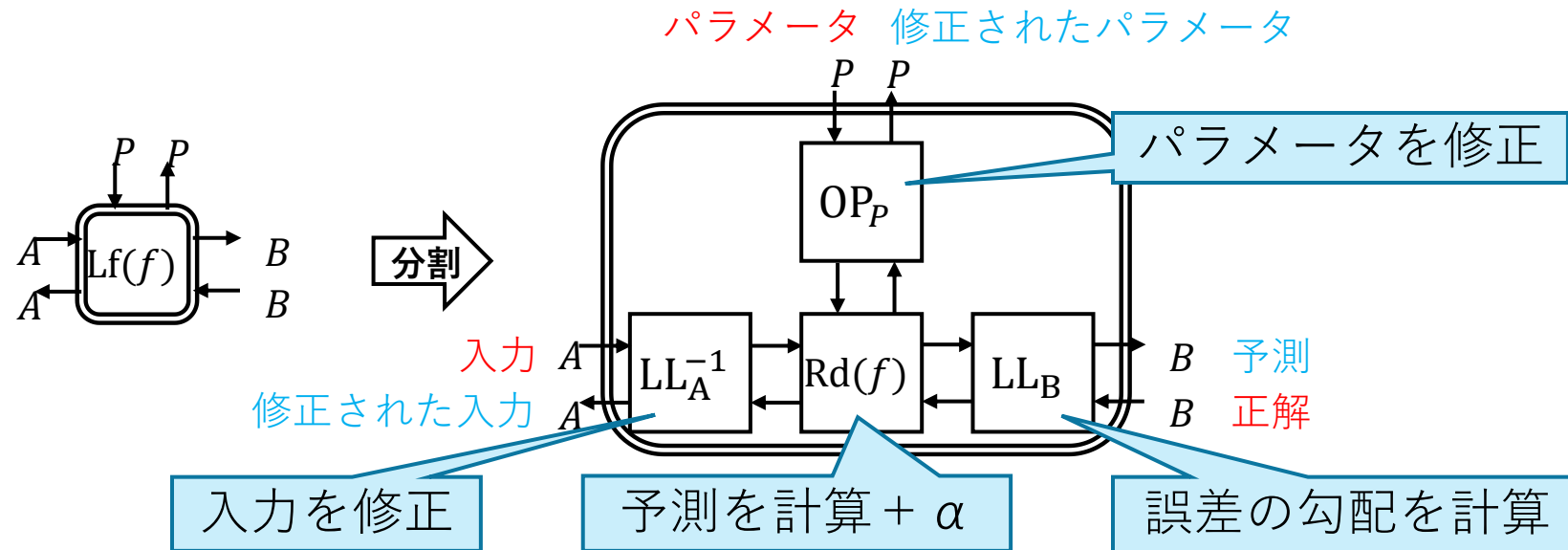
1. 既存研究
2. 新規結果
 - a. **Lfのモジュール化による新規構成の定義**
 - b. Lfの性質のモジュールへの帰着
 - l. Fongらの構成のwell-defined性
3. 結果の応用

learnerの構成のモジュール化

- learner $Lf(f)$ を (Crutwellらに倣って) モジュール化
⇒ 4つのレンズ $LL_B, Rd(f), LL_A^{-1}, OP_P$ から learner を構成できる



learner構成の一般化



- learner構成のモジュール化 \Rightarrow 各モジュールを一般化

- 最適化計算 OP : 勾配降下法のみ \rightarrow Momentum, Adagrad [Cruttwell+ '23] など
- 誤差の計算 $LL(+LL^{-1})$: 二乗和誤差のみ \rightarrow 平均二乗誤差・ソフトマックス交差エントロピー (SCE) [Cruttwell+ '23] など

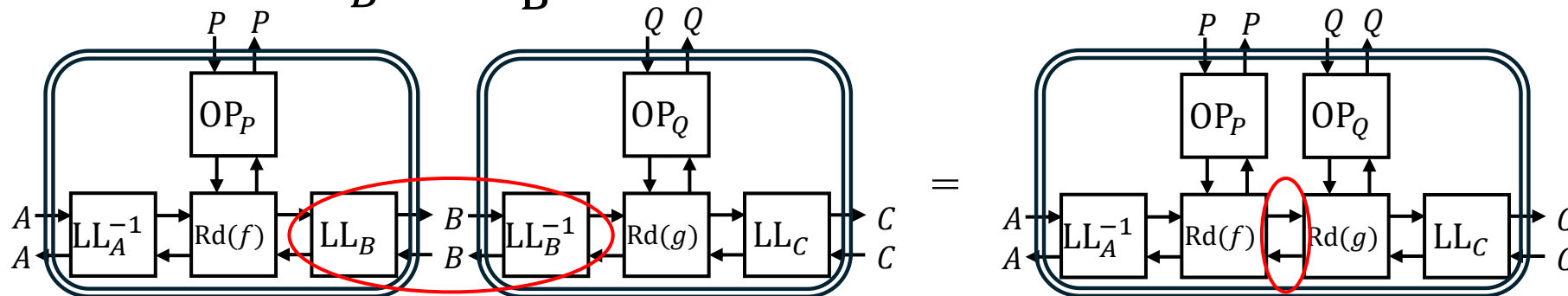
アウトライン

1. 既存研究
2. 新規結果
 - a. Lf のモジュール化による新規構成の定義
 - b. Lf の性質のモジュールへの帰着
 - l. Fongらの構成のwell-defined性
3. 結果の応用

learner構成の性質の構成モジュールへの帰着

入力の修正: $a' := (\nabla_a e(a, -))^{-1} (\nabla_a L)$

- 関手性: LL_B と LL_B^{-1} が互いに同型であることに帰着

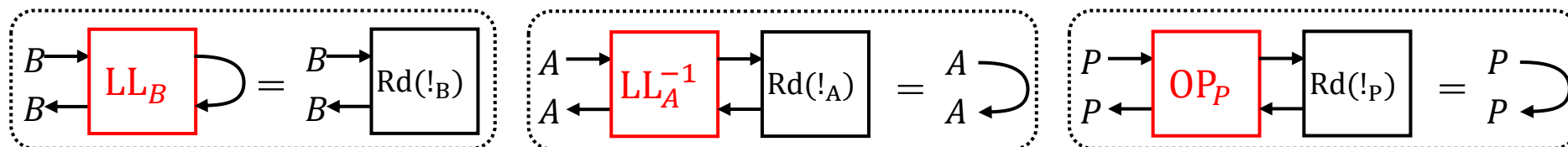


LL_B と LL_B^{-1} が相殺

アーキテクチャ + α 同士が隣接
 \Rightarrow アーキテクチャの結合を保つ

- GetPut則: LL, LL^{-1}, OP の以下の性質に帰着

- 次の三つの等式が成り立つ \Rightarrow 構成した全learnerがGetPut則を満たす



アウトライン

1. 既存研究
2. 新規結果
 - a. Lf のモジュール化による新規構成の定義
 - b. Lf の性質のモジュールへの帰着
 - l. Fongらの構成のwell-defined性
3. 結果の応用

アーキテクチャ（やlearner）の同一視

- **Para(C)**（や**Para(Lens(C))**）の射は同値関係 \sim で割られる
 - 適切な同値類によってbicategoryからcategoryになる
 - **C**の（対称モノイダル）部分圏**D**を取る
 - 以下のように同値関係を定義する

Cruttwellらの構成では
strictificationを使用
同値類は不使用

$$g \sim f \iff A \xrightarrow{\quad} \begin{array}{c} Q \\ \downarrow \\ \boxed{g} \end{array} \xrightarrow{\quad} B = A \xrightarrow{\quad} \begin{array}{c} \boxed{i} \\ \downarrow P \\ \boxed{f} \end{array} \xrightarrow{\quad} B$$

$\exists i \in \mathbf{D}_{\text{iso}}(Q, P)$

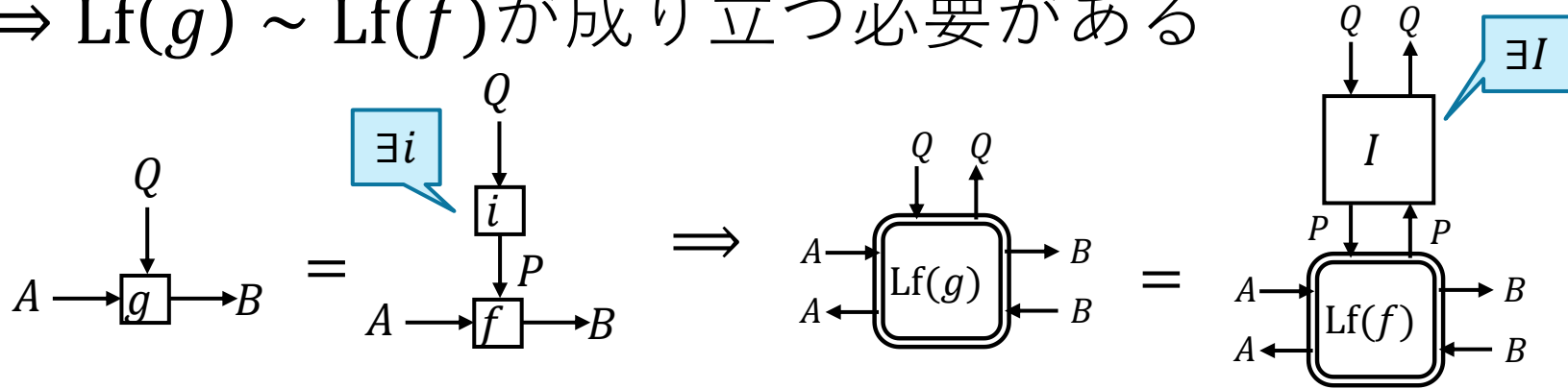
- ⇒アーキテクチャはパラメータ側における圏**D**の操作で不変
- **D**の一般例 **Iso(C)**：**C**における同型射のみからなる部分圏
 - **D**の新規例 **C_S**：**C**における自明な同型射のみからなる部分圏
 - 自明な同型射：identity, associator, unitor, symmetryから構成

関手LfのWell-defined性

Para(C)

Para(Lens(C))

- $g \sim f \Rightarrow \text{Lf}(g) \sim \text{Lf}(f)$ が成り立つ必要がある



- Fongらによる同値関係：部分圏 $\mathbf{Iso}(\mathbf{C})$ と特殊な部分圏を用いて定義

- $i = \phi, I = \begin{matrix} \downarrow \\ \boxed{\psi} \\ \downarrow \end{matrix} \begin{matrix} \uparrow \\ \boxed{\psi} \\ \uparrow \end{matrix}$ (ϕ, ψ は \mathbf{C} の任意の同型射) \Rightarrow 成り立たない

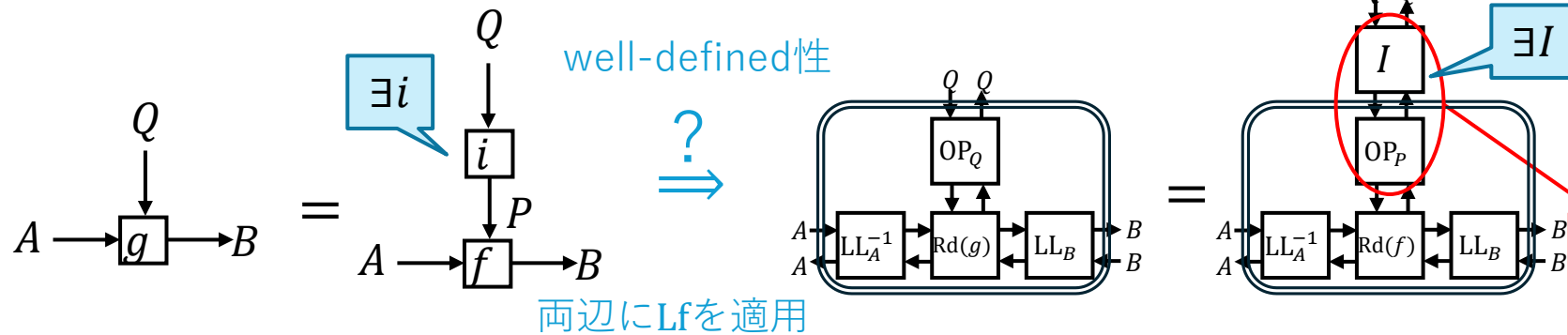
- 新規同値関係：部分圏 $\mathbf{C}_S, \mathbf{Lens}(\mathbf{C})_S$ によって定義

- $i = \phi, I = \begin{matrix} \downarrow \\ \boxed{\psi} \\ \downarrow \end{matrix} \begin{matrix} \uparrow \\ \boxed{\psi^{-1}} \\ \uparrow \end{matrix}$ (ϕ, ψ は \mathbf{C} の任意の自明な同型射) \Rightarrow 成り立つ

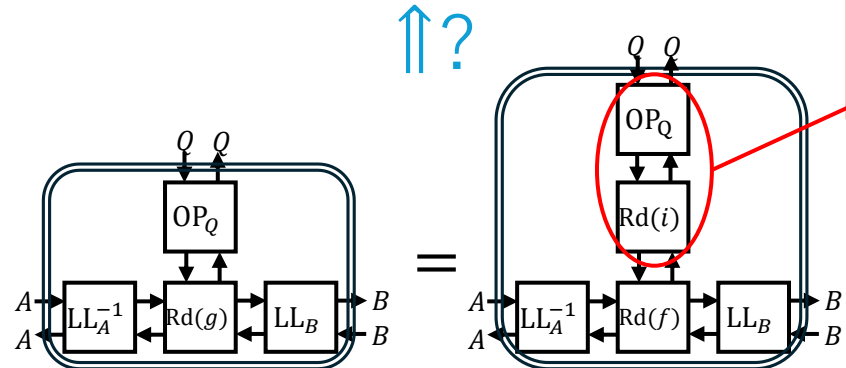
関手Lfのwell-defined性のモジュールへの帰着

- well-defined性：OPの**自然性**に帰着

- レンズOPはパラメータ側に合成 \Rightarrow **同値関係 \sim に干渉**

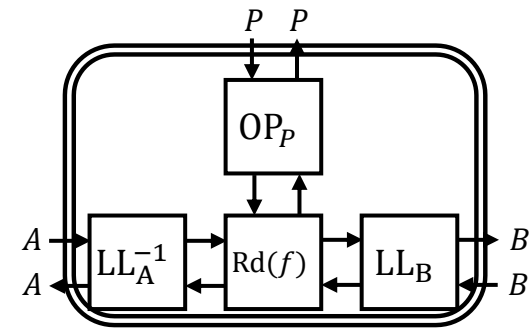


- 部分圏 \mathbf{C}_S , $\mathbf{Lens}(\mathbf{C})_S$ を用いて i, I を自明な同型射に制限する \Rightarrow **勾配降下法のOPは自然**

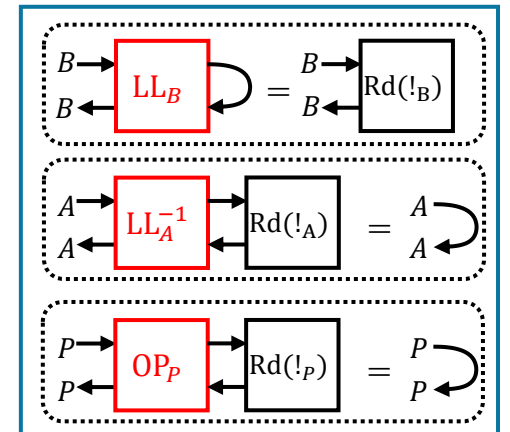


一致(OPが自然)ならば well-defined

モジュールの具体例の評価



- 性質のモジュールへの帰着 \Rightarrow 各モジュールの"健全性"が評価可能に
 - LLと LL^{-1} が互いに同型 \rightarrow 関手性
 - LL, LL^{-1} , OPが左の性質を満たす \rightarrow GetPut則
 - OPが自然 \rightarrow well-defined性
- LL, LL^{-1} , OPの具体例を調査
 - モジュールの性質からlearner構成の性質が導出できる



OP	GetPut	well-defined
勾配降下法	○	○
Momentum, Adagrad	×	○

LL^{-1}	LL	GetPut	関手性
二乗和誤差	二乗和誤差	○	○
平均二乗誤差	平均二乗誤差	○	○
二乗和誤差	SCE	○	×

詳細は後に

アウトライン

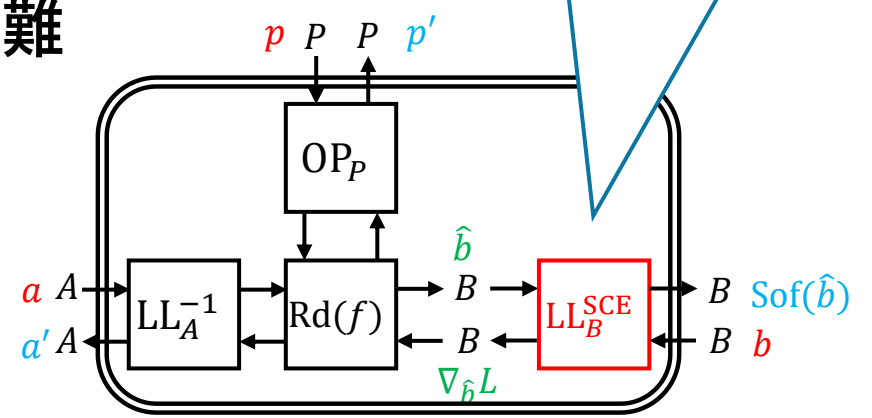
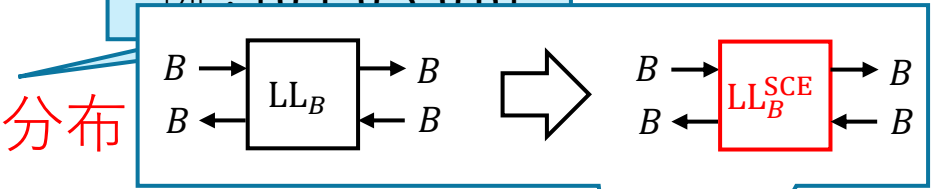
1. 既存研究
2. 新規結果
3. 結果の応用
 - a. 確率分布に対応したlearner構成の定義
 - b. モジュールの性質からの構成の性質の導出

分類タスクを学習するlearner

- タスクによって訓練データの形式が異なる
 - 回帰タスク → 入力：ベクトル 正解：ベクトル
 - 回帰learner：回帰タスクを学習するlearner
 - 例：二乗和誤差を用いる $L(f)$
 - 分類タスク → 入力：ベクトル 正解：有限確率分布
 - 分類learner：分類タスクを学習するlearner
 - Fongらの手法では分類learnerの構成が困難
- 新規構成による分類learner
 - LL_B の代わりに特殊なレンズ LL_B^{SCE} を使用
 - 確率分布の比較に対応
 - モジュール化により容易に置換可能

例：(1.5, -8, 40)

例：(0.1, 0.3, 0.6)

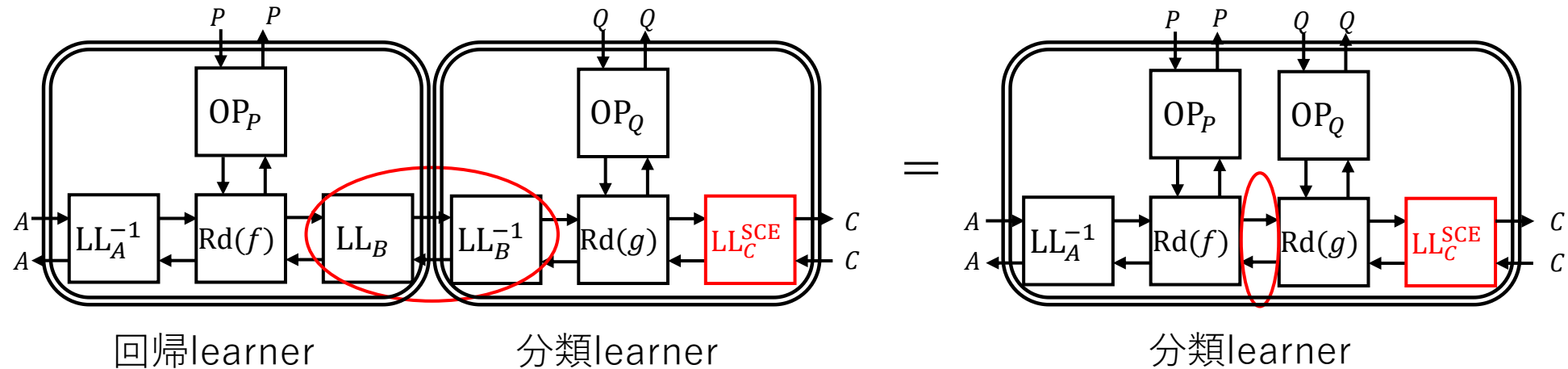


アウトライン

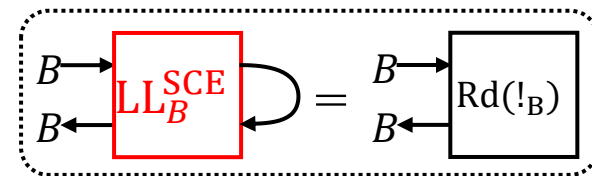
1. 既存研究
2. 新規結果
3. 結果の応用
 - a. 確率分布に対応したlearner構成の定義
 - b. モジュールの性質からの構成の性質の導出

分類learner構成の性質の導出

- **関手性**：満たさないが、現実的な範囲では成り立つ
 - 実用上、分類learnerが現れるのは「**回帰learner ; 分類learner**」の形のみ



- **GetPut則**： OP, LL^{-1} が帰着された性質を満たせば成り立つ
 - LL_B^{SCE} は帰着された性質を満たす
- **well-defined性**：回帰learnerと同様



まとめ・既存研究との比較

Cruttwellらの
構成の長所

Fongらの
構成の長所

- Fongらの**learner構成**をCruttwellらの手法で**モジュール化**し一般化
- その**具体例**で以下を評価
 - 関手性
 - GetPut則
 - well-defined性
- **分類タスク**を学習するlearner構成を提案

正解が ベクトル	モジュール 化の種類	学習の構成	微分の 一般化	最適化手法 の一般化	関手性	GetPut 則
Cruttwell+	学習構成	○	○	○	N/A	N/A
Fong+	learner	○	×	×	○	○
新規	両方	○	○	○	○	○

正解が 確率分布	モジュール 化の種類	学習の構成	微分の 一般化	最適化手法 の一般化	関手性	GetPut 則
Cruttwell+	学習構成	○	○	○	N/A	N/A
Fong+	learner	△	×	×	○	×
新規	両方	○	○	○	△	○

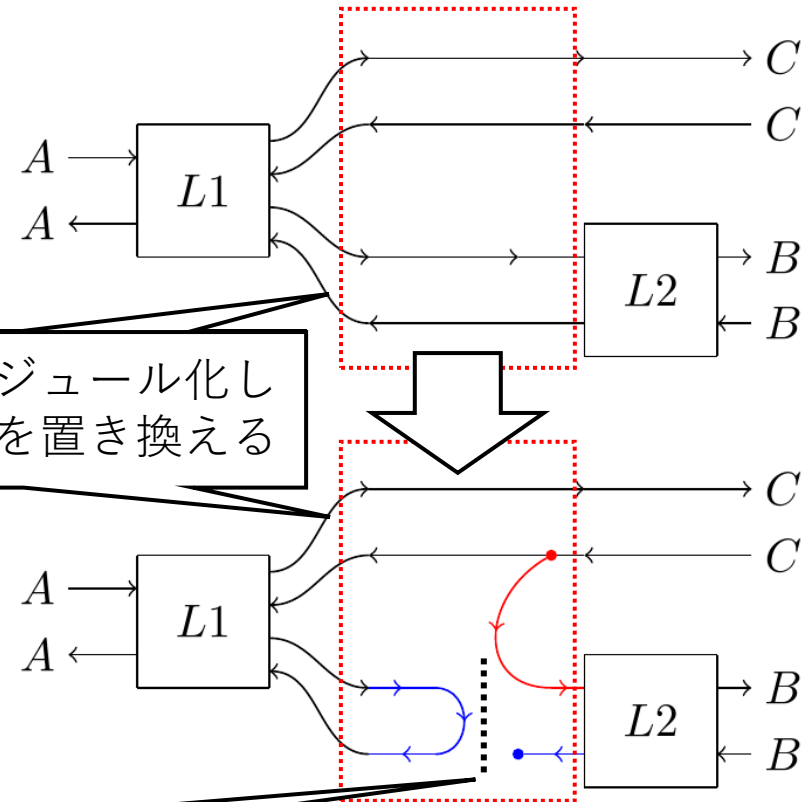
付録A：学習手法のモジュール化の例

- 教師強制：訓練に近似を導入する手法の一つ
 - 学習の効率が上がる
- learner自身によるモジュール化
 - 教師強制もモジュール化可能
 - 教師強制の適用はモジュールの置換
- 利点：レンズがlearnerでなくとも適用できる
 - 例：ベイズ学習, 線形回帰など

$L1, L2$: learner

計算の流れをモジュール化しそのモジュールを置き換える

直感： $L1$ と $L2$ の依存関係を近似によって分断



付録A：教師強制の拡張

- 新発見：左上に $L3$ が存在しても、教師強制が適用できる
⇒ 拡張された教師強制を ignoring と呼ぶことにする
- 実際の計算の変化：赤線に流れる値が変化
訓練データ $\in C$
→ 修正された入力 $\in C$

Fongらの構成特有の操作で得られる値

• これで本当に学習ができる？

- 現状：学習は可能だがほとんどの場合で性能は落ちる

