The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

類似学習ケースに対するカリキュラム学習

Curriculum Learning for Similar Learning Cases

太田 悠太 *1　　滝 勇太 *1
Yuta Ota　　Yuta Taki

*1株式会社構造計画研究所
KOZO KEIKAKU ENGINEERING Inc.

In appricationsof machine learning, we often need to generate some models for multiple similar cases. In our study, we propose an efficient learning method exploiting the idea of the curriculum learning. Our idea is that we evaluate the effectiveness of a curriculum learning for one problem, then we use the learned curriculum as feedback for learning other cases. We apply our proposal method to two small problems, addition and subtraction of integers, using Sequence-to-Sequence model. Our experimental results show that the method works efficiently for some problem settings.

1. はじめに

機械学習の活用において，複数の類似する対象について，それぞれの学習モデルを構築したいことがある．

例えば，工場設備に対する異常検知モデルを構築したい場合，同じ種類の設備であっても設備の個体差や配備される工場の環境の差が存在するため，共通のモデルを構築するよりも，各設備に対してそれぞれのモデルを構築する方が望ましいことがある．また，生産ライン上の複数のロボットに動作を学習させたい場合，各ロボットの基本的な動作は類似しているが，担う機能によって必要な動作が部分的に異なるため，各ロボットに対してそれぞれの学習を行いたい場合が考えられる．

本研究では，このような一定の類似性を持つ複数の学習対象についてそれぞれのモデルを構築したい場合を考え，各学習を独立に行うのではなく，学習間で学習の効率性に関する情報を共有することで一連の学習を効率的に行う方法を提案する．より具体的には，カリキュラム学習 [Benjio 2009, Zaremba 2014, Wu 2017, Graves 2017, Matiisen 2017] を活用し，ある学習対象に対して行ったカリキュラム学習の効率性を評価し，類似する別の対象に関する学習を行う際に，その評価結果に基づくフィードバックを行う仕組みを提案する．

2. 関連研究

ここでは，カリキュラム学習に関する研究について説明を行う．カリキュラム学習とは，学習期間の短縮やモデルの高精度化のため，学習する事例の順序を工夫する学習アプローチを指す．

2.1 カリキュラム学習に関する研究

Benjio らは学習において相対的に容易な事例を最初に提示し，徐々に事例の難易度を高くすることで，ランダムに事例を提示するよりもモデルの予測精度が高くなることを画像分類，言語モデルのそれぞれの実験により示している [Benjio 2009]．ここでは基本的に難易度が上がっていくように事例を並び替えることでカリキュラムを生成しており，画像分類タスクでは難易度は目視で判断され，言語モデルでは単語数を難易度として並び替えを行なっている．

連絡先: 太田悠太，株式会社構造計画研究所，東京都中野区中央 5-4-3，03-5342-1125，yuta-ota@kke.co.jp

また，Zaremba らは計算プログラムのコードから計算結果を予測する問題を LSTM(long short-term memory) で学習する方法を提案する中で，従来の確率的勾配降下法による学習が困難な問題に対して，適切なカリキュラムによる学習戦略が重要であることを示している [Zaremba 2014]．カリキュラムの生成は，コードの長さと含まれる文法の種類数の 2 種類の難易度パラメータを考え，単純に難易度を上げていく戦略だけでなく，いずれかの難易度を固定する事例や，部分的にランダムに難易度を選択した事例を混ぜる戦略により，予測精度が高くなることを示している．

2.2 カリキュラムの自動生成に関する研究

Graves らは，Zaremba らのカリキュラムによる学習戦略に限界があることを指摘し，カリキュラムを自動生成する仕組みを構築した [Graves 2017]．より具体的には，敵対的バンディットの報酬として，学習の進捗率と学習対象のニューラルネット構造の複雑性の増加率を与え，ニューラルネットに対する学習において，学習エポックごとに提示する事例を選択するという枠組みを提示した．この枠組みを 3 つの実験で検証し、いくつかの設定で同じ精度になる学習時間を半分にできることを示した．また，Matiisen らは非定常バンディットを拡張した方法により，同様にカリキュラムを自動生成する仕組みを構築した [Matiisen 2017]．

これらの研究は，提示した事例に対する学習の進捗に基づき，次に提示する事例を選択するアルゴリズムを用いることで，カリキュラムの自動生成を可能にしている．

3. 提案手法

前節ではカリキュラム学習に関する研究について説明した．ここでは，類似性を持つ複数の学習対象について，各々のモデルを構築する問題設定を考える．そして学習を行うにあたり，ある学習対象に対してカリキュラム学習を行う過程で観測可能な「学習モデルがどのような状態の時に，どの事例を提供したら，どの程度学習が進捗したか」に関する情報を活用し，別の類似する学習対象の学習をより効率的に行う仕組みを提案する．

3.1 概要

本研究で提案する類似学習ケースに対するカリキュラム学習の仕組みを図 1 に示す．1 ケース目の学習は「事前学習」の位

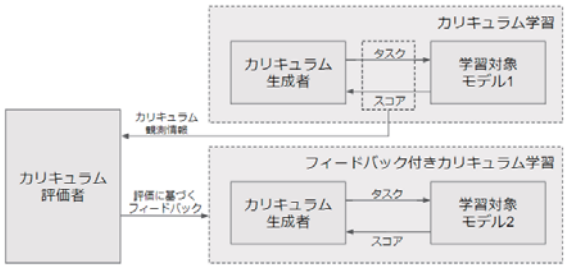

図 1: 類似学習ケースに対するカリキュラム学習の仕組みの概念図

置付けの学習であり，カリキュラムの自動生成 [Graves 2017, Matiisen 2017] により行われる (図 1 のカリキュラム学習)．2 ケース目 (もしあればそれ以降) の学習はターゲットとなる学習であり，カリキュラムの自動生成に加えて，1 ケース目のカリキュラムの効率性の評価に基づくフィードバックを踏まえたカリキュラム学習を行う (図 1 のフィードバック付きカリキュラム学習)．

ここでは，カリキュラムの自動生成機能を提供するクラスを「カリキュラム生成者」と呼び，事前のカリキュラム学習に対する評価機能とフィードバック機能を提供するクラスを「カリキュラム評価者」と呼ぶ．また，1 ケース目のカリキュラムの効率性に関する情報を「カリキュラム観測情報」と呼ぶ．

図 1 では 1 ケースのカリキュラム学習に対するカリキュラム観測情報を，別の類似する 1 ケースの学習に活用する状況を示しているが，3 ケース目以降の学習についても同様に評価とフィードバックを逐次的に行う仕組みを取ることも想定する．また，事前の複数のカリキュラム観測情報を合わせて評価してフィードバックを行う仕組みを取ることも想定する．

次節以降でこの仕組みの詳細を説明する．

3.2 カリキュラム学習

ここでは本研究で用いるカリキュラム学習の仕組みを示す．カリキュラム学習の仕組みは，[Graves 2017, Matiisen 2017] に従い，ここでは教師あり学習を考える．以下にその設定を示す．

目標系列として $Y_1, Y_2, \ldots$ を，入力系列として $X_1, X_2, \ldots$ を考える．系列は N 個のバッチに分割され，これをタスク D_k, $k = 1, \ldots, N$ とする．カリキュラム学習は T 回のバッチ学習により行われ，より少ない学習ステップで学習が収束することを目指して各学習ステップ t におけるタスクの選択を行う．このタスクのアンサンブルをカリキュラムと呼ぶ．

学習の目標として 2 つの設定が考えられる．1 つは全てのタスクが目標タスクである場合であり，この場合は，全てのタスクに対する学習の収束が目標となる．2 つ目は，興味のある任意のタスク $D_{\hat{k}}$ のみが目標タスクである場合である．この場合は $D_{\hat{k}}$ に対する学習の収束が目標となり，通常，$D_{k\neq\hat{k}}$ のタスクは，$D_{\hat{k}}$ に対する効率的な学習を達成するためのより難易度の低いタスクである．

3.3 カリキュラムの自動生成方法

カリキュラム生成者は各学習ステップ t でタスクを選択することで，カリキュラムを生成する．本研究ではカリキュラム生成者のカリキュラム自動生成方法として [Graves 2017] の仕組みを採用する．本節では [Graves 2017] のカリキュラムの自動生成方法を示す．

カリキュラム学習を T 回のタスク選択により総報酬 $\sum_{t=1}^{T} r^{(t)}$ を最大化するバンディット問題として考える．ここで，バンディット問題におけるアームの選択は時点 t におけるタスク D_k の選択，報酬 $r^{(t)}$ はタスク D_k を与えた時の学習の進捗率とする．このとき，各タスクに対する期待報酬は学習の進捗により変化するため，非定常な期待報酬を想定できる敵対的バンディットアルゴリズムの 1 つである Exp3.S アルゴリズム [Auer 2002] を用いる．Exp3.S アルゴリズムでは，報酬 $r^{(t)}$ に基づいて，タスク D_k を採用する確率に関する重みパラメータを学習ステップ t ごとに更新する．

各 t ステップ目の学習における目標タスクに対する損失を $\mathcal{L}^{(t)}$ とした時，報酬 $r^{(t)}$ を以下の式で与える：

$$r^{(t)} = \mathcal{L}^{(t)} - \mathcal{L}^{(t-1)}. \tag{1}$$

式 (1) は学習の進捗により取りうる範囲が変化するため，全期間における共通指標としては不適切である．従って，t ステップまでのすべての $r^{(t)}$ の 20 %四分位点を $q_{lo}^{(t)}$、80 %四分位点を $q_{hi}^{(t)}$ として、正規化を行った以下の指標 $\hat{r}^{(t)}$ を報酬として用いる：

$$\hat{r}^{(t)} = \begin{cases} -1 & \text{if} \quad r^{(t)} < q_{lo}^{(t)} \\ 1 & \text{if} \quad r^{(t)} > q_{hi}^{(t)} \\ \frac{2(r^{(t)} - q_{lo}^{(t)})}{q_{hi}^{(t)} - q_{lo}^{(t)}} - 1 & \text{otherwise.} \end{cases} \tag{2}$$

3.4 フィードバック付きカリキュラム学習

本節ではフィードバック付きカリキュラム学習 (FCL: Feedback-based Curriculum Learning) について説明する．FCL ではカリキュラム生成者によるカリキュラムの自動生成に加えて，カリキュラム観測情報に基づくフィードバックを踏まえた学習を行う．フィードバックの仕組みは以下の通りである．

いま，事前のカリキュラム学習により，カリキュラム観測情報が得られているものとする．式 (2) の報酬 $\hat{r}^{(t)}$ はその定義から $[-1, 1]$ の連続値をとるが，ここでは $\hat{r}^{(t)}$ の値に応じて $[-1, 1]$ を J 分割した離散的なスコア R_j, $j = 1, \ldots, J$ のいずれかをとるものとする．この時，カリキュラム観測情報は一期前に与えたタスク $D_k^{(\tau-1)}$，その時得られたスコア $R_j^{(\tau-1)} (\ni \hat{r}^{(\tau-1)})$，次に与えたタスク $D_l^{(\tau)}$ からなる状態 $S_{k,j,l} \equiv (D_k^{(\tau-1)}, R_j^{(\tau-1)}, D_l^{(\tau)})$, $k = 1, \ldots, N$, $j = 1, \ldots, J$, $l = 1, \ldots, N$ を持つ．カリキュラム学習の各ステップでは，式 (2) で与えられる報酬を得るが，その時の状態は $\{S_{k,j,l}\}$ のいずれかをとる．状態 $S_{k,j,l}$ をとった時の報酬の集合を $\mathcal{R}_{k,j,l} = \{\hat{r}^{(t_1)}, \ldots, \hat{r}^{(t_{T_{k,j,l}})}\}$ と表現する．ただし，$T_{k,j,l}$ は状態 $S_{k,j,l}$ をとった回数を表す．状態 $S_{k,j,l}$ を一度も取らなかった場合には，$\mathcal{R}_{k,j,l} = \phi$ となる．

カリキュラム評価者は，現在状態が $S_{k,j,\hat{k}}$ における価値を報酬集合 $\mathcal{R}_{k,j,\hat{k}}$ を用いて以下の式で計算する：

$$V_{k,j,\hat{k}} \equiv V(\mathcal{R}_{k,j,\hat{k}}) = \frac{1}{|\mathcal{R}_{k,j,\hat{k}}|} \sum_{r \in \mathcal{R}_{k,j,\hat{k}}} r$$

なお，$\mathcal{R}_{k,j,l} = \phi$ の場合は，$\mathcal{I} = \{(k, j, l') | \mathcal{R}_{k,j,l'} \neq \phi\}$ として，$\{V_{k,j,l'} | (k, j, l') \in \mathcal{I}\}$ の平均を用いる．

FCL の各ステップでは，一期前に選択したタスクが D_k，その際の報酬が R_j だった時，次に選択するタスク $D_{\hat{k}}$ の選択確

Algorithm 1 フィードバック付きカリキュラム学習

1: **for** $t = 1, \ldots, T$ **do**
2: カリキュラム生成者によりタスクごとの選択確率 P^G を計算
3: カリキュラム評価者によりタスクごとの選択確率 P^F を計算
4: $P \leftarrow (1-\alpha^{(t)})P^G + \alpha^{(t)}P^F$
5: 選択確率 P に基づいてタスク D_k を選択
6: 選択したタスク D_k により学習対象モデルを訓練
7: 目標タスクに対する損失 $\mathcal{L}^{(t-1)}$, $\mathcal{L}^{(t)}$ から報酬 $\hat{r}^{(t)}$ を計算
8: 報酬 $\hat{r}^{(t)}$ によりカリキュラム生成者の重みパラメータを更新
9: **end for**

率 $P^F_{k,j}(\hat{k})$ をフィードバックとして利用する．$P^F_{k,j}(\hat{k})$ は価値集合 $\{V_{k,j,s}\}_{s=1,\ldots,N}$ を用いて，

$$P^F_{k,j}(\hat{k}) = \frac{\exp V_{k,j,\hat{k}}}{\sum_{s=1}^{N} \exp V_{k,j,s}}, \tag{3}$$

により与える．

最終的にタスク $D_{\hat{k}}$ を選択する確率 $P_{k,j}(\hat{k})$ は，カリキュラム生成者の Exp3.S により得られるタスク選択確率 $P^G_{k,j}(\hat{k})$ および式 (3) により得られるフィードバックを用いて以下により計算する：

$$P_{k,j}(\hat{k}) \quad = \quad (1-\alpha^{(t)})P^G_{k,j}(\hat{k}) + \alpha^{(t)}P^F_{k,j}(\hat{k}).$$

ここで，$\alpha^{(t)}$ はカリキュラム評価者，カリキュラム生成者の各タスクの選択確率の平滑化パラメータである．$\alpha^{(t)}$ が 1 に近いほどカリキュラム評価者を信頼し，0 に近いほどカリキュラム生成者を信頼する．$\alpha^{(t)}$ が常に 0 の時，カリキュラム生成者が算出した確率 $P^G_{k,j}(\hat{k})$ にのみ従い，Graves らが提案した仕組みと一致する．

カリキュラム評価者によるフィードバックは，カリキュラム生成者が報酬に関する情報を充分に受け取っていない学習期間の序盤で，より有効であると考えられる．これを踏まえ，$\alpha^{(t)}$ は以下の式で与える：

$$\alpha^{(t)} = \frac{1}{1+\exp[-a(T_F - t)]},\ T_F > 0,\ a > 0.$$

ここで，T_F はカリキュラム評価者の選択確率をカリキュラム生成者の選択確率よりも重視する期間である．$T_F > t$ の場合は $\alpha^{(t)} > 0.5$ であるため，$P^F_{k,j}(\hat{k})$ が重視される期間となる．この期間をすぎると，徐々に $\alpha^{(t)}$ が減少する．また，a は $\alpha^{(t)}$ の減少の傾きの大きさを決める．a が小さいほど $\alpha^{(t)}$ の減少は緩やかに行われる．

ここまでの議論を踏まえ，FCL のアルゴリズムを Algorithm 1 に示す．

4. 計算機実験

4.1 実験設定

実験は，Sequence-to-Sequence モデル [Sutskever 2014] による加算問題 [Zaremba 2014] を対象とするもの，および，これを応用した減算問題を対象とするものの 2 種類を行う．加算問題は，文字列で与えられる'+' 記号で区分された 2 つの数字列を入力とし，2 つの数字列の和を出力とする．減算問題は'-' を用い，2 つの数字列の差を出力とする．入出力の例を表 1 に示す．

表 1: 入出力データ例

	入力	出力
例 1：加算問題	'29742+315'	'30057'
例 2：減算問題	'15413-8142'	'7271'
例 3：減算問題	'919-2313'	'-1397'

カリキュラムのタスク $\{D_k\}_{k=1}^N$ の設定は [Zaremba 2014, Matiisen 2017] の設定に倣う．加算 (もしくは減算) の入力における 2 つの数字列の最大桁数をタスクの難易度とする．表 1 の例 1, 2 は最大桁数 5 桁，例 3 は最大桁数 4 桁である．最大桁数 1 から 9 の 9 種類のタスクを用いることとし，各タスクにおいて 4,096 サンプルの入出力セットを生成する．各学習ステップでは，この 9 つのタスク $\{D_k\}_{k=1}^9$ から 1 つを選択して，モデルパラメータの更新を行う．タスクとは別に目標タスクとして最大桁数 9 桁の入出力セットを 128 サンプル生成する．モデルの損失 $\mathcal{L}^{(t)}$ はこの目標タスクを対象として算出する．損失は，正解の出力とモデルによる予測値の差として，128 サンプルに対する平均絶対偏差により計算する．

本実験では，提案手法である FCL を含む 4 つの手法，FCL, [Graves 2017] に従うカリキュラム学習 (CL)，各学習ステップにおいてランダムにタスクを選択する方法 (RANDOM)，すべての学習ステップにおいて目標タスクと同じ最大桁数 9 桁のタスクを選択する方法 (MAX)，を適用する．各手法に対して，目標タスクに対する損失の収束に要したステップ数，収束時の損失の大きさを比較する．FCL においてカリキュラム評価者が受け取るカリキュラム観測情報の内容は各実験の節に記載する．

実験において採用した各種パラメータは次の通りである．Sequence-to-Sequence モデルにおいて，encoder，decoder ともに 256 ユニットの LSTM を用い，RMSProp による最適化を行った．カリキュラム生成者が用いる Exp3.S については定式化を [Graves 2017] に従い，各パラメータは $\beta = 0.001$, $e = 0.02$, $\eta = 0.01$ とした．カリキュラム評価者はモデルの学習進捗率の状態を $[-1,\ 1]$ を 0.2 ステップで分割し，平滑化パラメータについては $T_F = 120$, $a = 0.2$ とした．

4.2 加算問題

各学習ステップにおける損失を図 2 に示す．学習はランダムシードを変更して 3 回行った．各学習について 20 ステップ分の移動平均をとった損失を求め，3 回の学習の平均損失の対数値を示している．

FCL におけるカリキュラム観測情報として，減算問題に対するカリキュラム学習をステップ 700 まで行い，損失の減少が大きいステップ 400 までをカリキュラム観測情報として利用した．また，カリキュラム観測情報が多く与えられるほど効率的な学習が達成されることを検証するため，カリキュラム学習 1 回分のカリキュラム観測情報を利用したときの学習を FCL(1)，3 回分のカリキュラム観測情報を利用したときの学習を FCL(3) としてそれぞれを実験対象とした．

結果として，FCL(1), FCL(3) は 100 ステップ目以降で CL と比較して損失減少の傾きが大きくなり，最終的に CL, MAX, RANDOM 手法に比べて，より小さい損失に収束していることが確認できる．このことは，フィードバックが有効であるこ

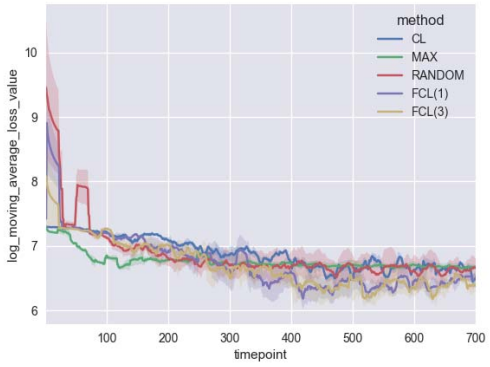

図 2: 加算問題における目標タスクに対する損失

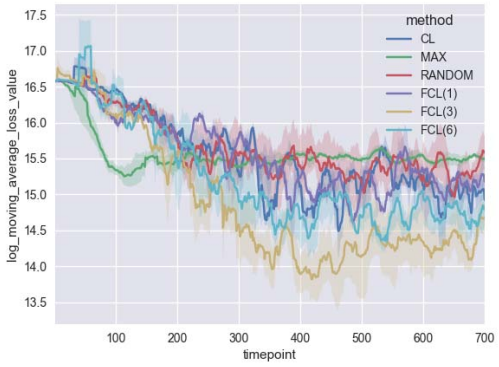

図 3: 減算問題における目標タスクに対する損失

とを示唆している．

なお，FCL(1) と FCL(3) は同程度の値に収束し，その収束にかかるステップ数も同程度であった．本実験からはカリキュラム観測情報の多寡による収束速度および損失への影響は明確に現れなかった．

4.3 減算問題

各学習ステップにおける損失を図 3 に示す．加算問題と同様に，学習はランダムシードを変更して 3 回行った．各学習について 20 ステップ分の移動平均をとった損失を求め，3 回の学習の平均損失の対数値を示している．

FCL におけるカリキュラム観測情報として，加算問題に対するカリキュラム学習をステップ 700 まで行い，損失の減少が大きいステップ 400 までをカリキュラム観測情報として利用した．カリキュラム学習 1 回分のカリキュラム観測情報を利用したときの学習を FCL(1)，3 回分のカリキュラム観測情報を利用したときの学習を FCL(3)，6 回分のカリキュラム観測情報を利用したときの学習を FCL(6) としてそれぞれを実験対象とした．

結果として，FCL(1)，CL は同程度の値に収束し，MAX，RANDOM より小さい損失に収束していることが確認できる．FCL(3) はステップ 150 程度から CL と比較して損失減少の傾きが大きくなり，より小さい損失に収束している．また，FCL(6) は 200 ステップ目以降で CL と比較して損失減少の傾きが大きくなり, 充分な差ではないものの，より小さい損失に収束している．これらはフィードバックが有効であることを示唆している．

この実験では，FCL(3) が最も小さい損失となり，FCL(6)，FCL(1) の順にこれに続く結果となった．前節の加算問題と同様に，カリキュラム観測情報は多く与えれば良いというわけではないことが確認された．従って，対象とするタスクに対して適切な量および質のカリキュラム観測情報を与えることが重要であると考えられる．

5. 結論

本研究では，類似性を持つ複数の学習ケースについてモデルを構築したい場合を考え，カリキュラム学習の仕組みを応用し，カリキュラム観測情報に基づくフィードバックによって類似ケースの学習をより効率的に行う手法を提案した．

実験により，提案手法について一定の効果がある可能性を確認することができた．カリキュラム観測情報の与え方によって収束速度および損失が変化することから，適切なカリキュラム観測情報の選択方法についても検討の必要がある．また，カリキュラム観測情報の結果への影響度合を検証することで,「学習対象の類似性」を明らかにし，フィードバックが有効に作用する条件などについても検証する必要がある．さらに，今回実施した実験設定が充分な規模と言い切れるものでないことから，今後はより大規模な実験設定や異なる問題に対する実験により，提案手法の有効性を検証する必要がある．

参考文献

[Auer 2002] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E.: The nonstochastic multiarmed bandit problem, SIAM Journal on Computing, 32(1):pp. 48-77 (2002)

[Benjio 2009] Bengio, Y., Louradour, J., Collobert, R., and Weston, J.: Curriculum learning, ICML (2009)

[Graves 2017] Graves, A., Bellemare, M., Menick, J., Munos, R., and Kavukcuoglu, K.: Automated curriculum learning for neural networks, arXiv preprint arXiv:1704.03003 (2017)

[Matiisen 2017] Matiisen, T., Oliver, A., Cohen, T., Schulman, J., and Kavukcuoglu, K.: Teacher-Student Curriculum Learning, arXiv preprint arXiv:1707.00183 (2017)

[Sutskever 2014] Sutskever, I., Vinyals, O., and Le, Quoc, V.: Sequence to sequence learning with neural networks, In Advances in neural information processing systems, pp. 3104-3112 (2014)

[Wu 2017] Wu, Y., and Tian, Y.: Training agent for first-person shooter game with actor-critic curriculum learning, In Submitted to Int'l Conference on Learning Representations (2017)

[Zaremba 2014] Zaremba, W., and Sutskever, I.: Learning to execute, arXiv preprint arXiv:1410.4615 (2014)