**introduction**

「2048」は G. Cirulli によって考案された確率的一人ゲームであり、これまでに数多くのコンピュータプレイヤが提案されてきた。現在もっとも成功しているアプローチは、強化学習によってチューニングした N タプルネットワーク評価関数と Expectimax 探索を組み合わせる手法であり、これを基盤として様々な改良が試みられている。たとえば Guei による最先端プレイヤでは、6-タプルからなる大規模なネットワークを複数用い、Optimistic Initialization やタイル固有の改良手法を組み合わせることで高得点を達成している。

一般に、N タプルネットワークやニューラルネットワークといったパラメトリックな評価関数では、パラメータ数の増加とともに表現力が向上し、性能の改善が期待される。しかしながら、パラメータ数が大きくなりすぎると、学習データやメモリ容量が膨大となり、実用上の制約が生じる。特に 2048 における N タプルネットワークでは、タプルのサイズを大きくするとパラメータ数が指数的に増加するため、7-タプル以上の利用は現実的でない。

この制約に対処するために、本研究では新たに **Vertical Split Encoding (VSE)** を提案する。2048 では値の大きく離れたタイルが同時に盤上に存在する確率は低いため、それらを同一タプルで扱うことは必ずしも効率的でない。そこで本手法では、低い値を担当するタプルと高い値を担当するタプルに分けることで、不要な結合を避けつつ大きなタプルを実現可能とする。

本研究では 2048 を題材として、N タプルネットワークに対して VSE を適用した場合の効果を評価し、VSE を導入したネットワークと従来型ネットワークの性能を比較し、2048 におけるパラメータ効率の改善可能性を検証する。

**2048**

ルール説明

2048は，4×4の盤面でプレイされる確率的一人ゲームである。  
初期局面では，盤面上に2つのタイルがランダムに配置される。新しく置かれるタイルの値は，2が確率0.9，4が確率0.1で選ばれる。

各ターンにおいてプレイヤは上下左右のいずれかの方向を選択する。すると盤面上の全てのタイルがその方向へ可能な限り移動する。移動の結果，同じ値のタイルが衝突した場合，それらは合体して合計値のタイルとなり，その値がスコアに加算される。合体したタイルは，同じターン中には別のタイルと再び合体することはない。例えば，盤面の一行が「\_\_\_\_」「2\_\_ \_」「\_\_22」「2224」であるとき，右を選択するとそれぞれ「\_\_\_\_」「\_\_\_2」「\_\_\_4」「\_\_244」に変化する。その後，空いているマスの中からランダムに1マスが選ばれ，そこに2（確率0.9）か4（確率0.1）のタイルが新たに置かれる。

プレイヤは，少なくとも1つのタイルが移動あるいは合体するような方向しか選択できない。どの方向にも動かせなくなった時点でゲームは終了する。ゲームの目的は，終了するまでにできるだけ高い得点を獲得することである。

用語の導入

2048における1ターンは，「移動・合体ステップ」と「新規タイルステップ」の2つから構成される。これらのステップの前後を区別するために，次の用語を導入する。

* **state**: プレイヤが手を選択する時点での盤面状態（およびスコア）。
* **afterstate**: プレイヤが手を選択し，タイルが移動・合体した直後の盤面状態（およびスコア）。新しいタイルが出現する前の盤面を指す。

また，2048では新しいタイルが2か4のいずれかでランダムに出現するため，単純にターン数をゲームの進行度の指標とするのは適切でない。そこで，本研究では以下の指標を用いる。

* **progress**: 盤面上のタイルの値の合計の半分を progress と呼ぶ。progress は1ターンごとに，新しいタイルが2なら1，新しいタイルが4なら2だけ増加する。

このようにして，state，afterstate，progress を導入することで，2048の進行状況をより適切に表現できる。

ミニ2048で試した話？

事前実験としてミニ2048を用いて**VSEの実験を行い….**

**Player**

Nタプルネットワーク

2048 の強いコンピュータプレイヤを作る最も成功した方法は，N タプルネットワークによる評価関数を強化学習によりチューニングするものである。  
本研究では，先行研究で設計・検討されたタプルに加えて，新たに大きさ 6，7，8，9 のタプルを用いた N タプルネットワークを構築した。表に本研究で用いたタプル設計を示す。

2048 の盤面の対称性を考慮し，各タプルについて回転・反転による対称な 8 通りのサンプリングを行うことで，実際に必要となるタプル数を削減している。NT6 は，先行研究でも広く用いられている大きさ 6 のタプルから構成され，標準的な性能を発揮することが期待される。

一方で，NT7，NT8，NT9 はタプルのサイズが大きいため，そのまま構築するとパラメータ数が爆発的に増加し，学習コストやメモリ使用量の観点から実現が困難である。そこで本研究では，大規模タプルを扱うための新しい工夫として，**Vertical Split Encoding (VSE)** を導入し，パラメータ数を削減しながら大規模タプルの表現力を活用できるようにした。

N タプルネットワークの重みは，afterstate 間の評価値の差に基づく TD 学習法の改良手法によって調整した。本研究における N タプルネットワークの学習では，Optimistic Initialization をはじめとする複数の技術を導入している。

今回のプレイヤで用いた技術の説明

* **Vertical Split Encoding (VSE)**

本研究では，大規模タプルを効率的に扱うための新手法 **Vertical Split Encoding（VSE）** を導入する。2048 では，値が同じ値のタイルが結合して合計の値のタイルになるため，値の大きく離れたタイルは重要性が異なる。実践的な戦略として「大きなタイルを盤面端に固定し，極力動かさない」方針が広く採用される。すなわち，**小さな値のタイル群と，大きな値のタイル群では担う役割が異なる**。この事実に基づき，VSE は盤面の値空間を**担当箇所ごとに分割**し，各帯域に特化したフィルタを用いてタプルの学習を行う。これにより，大きなタプルの表現力を維持しながらパラメータ数と学習コストを削減する。

本研究では表のフィルタを用いてそれ用のタプルの学習を行った．これによりパラメータ数は

* **Multistaging**  
  ゲームの進行に応じて、重みを参照するテーブルを切り替える手法である。本研究では 2 ステージとし、タイル 32768 が生成される前後でネットワークを切り替えるように設計した。
* **Temporal coherence 学習（TC 学習）**  
  学習率を自動的に調整する機能を備えた TD 学習であり、Jaśkowski によって初めて 2048 に導入された手法である。本研究における N タプルネットワークの学習では、効率的かつ安定的な収束のために TC 学習を採用した。
* **Optimistic initialization（OI）**  
  学習初期段階での探索を広く行うために、重みをゼロではなく大きな値で初期化する手法である。本研究では、すべての afterstate の初期値が 320,000 となるように重みを初期化した。

**Experiments**

本研究では実験として6,7,8,9のそれぞれのタプルと既存研究で用いたタプルの5種類についてそれぞれ2.0e12 step分学習を行い，200分の1ごとにパラメータを出力しGreedyプレイとExpectimaxプレイでそれぞれ評価を行った．

**Consideration**

**Conclusion**