社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－ARC－165（4）
2005／11／30

パーセプトロン分岐予測における冗長入力付加の効果

澁川　誠[†]　二ノ宮　康之[†]
阿部公輝[†]　小林　聡[†]

優れた予測精度を示すとして注目されているパーセプトロン分岐予測器は，線形分離不可能なパターンを学習することができないという欠点を持つ．この欠点を克服するために分岐履歴の相関を冗長入力として加えたパーセプトロン分岐予測法が考案されているが，予測精度の向上に最適な冗長入力の生成方法については研究が十分とはいえない．そこで，冗長の生成に使用する分岐履歴の長さと，履歴どうしの相関を形成する範囲が，予測精度にどんな影響を与えるかについて，シミュレーションにより調べた．その結果，冗長の生成に使用する分岐履歴が短いとかえって予測精度が悪化すること，精度を向上させるには十分な長さの分岐履歴から冗長入力を生成する必要があること，さらに履歴どうしの相関を形成する範囲が大きいほうが予測精度が向上することが分かった．また，64 ビットの GHR を用いて冗長入力を付加する場合，冗長を用いないパーセプトロン分岐予測器の平均予測ミス率を最大約 10%低減することが分かった．

Effects of Introducing Redundant History in Perceptron Branch Prediction

Akira Shibukawa,[†] Yasuyuki Ninomiya,[†] Kôki Abe[†]
and Satoshi Kobayashi[†]

Perceptron branch predictors have been attracting attentions because of their high prediction accuracy. However, perceptron predictors have a disadvantage that linearly inseparable patterns can not be learned. To overcome the drawback, ideas of introducing correlations among past branch behaviors to perceptrons as redundant inputs have been proposed. But studies on what factors affect on branch accuracy in introducing redundant inputs are insufficient. This paper investigates by simulation effects of two factors on prediction accuracy of perceptron branch predictors: length of branch history used for generating correlations and range within which correlations are formed. Results reveal that when branch history used for generating correlations is short, prediction accuracy rather decreases. For improving the prediction accuracy, long enough part of branch history needs to be used and correlations need to be formed within wide enough range. In case of 64 bit GHR, redundant inputs reduce the miss prediction rate of original perceptron predictor by 10 %.

1. はじめに

近年，学習理論を適用した分岐予測法が注目されている．中でもパーセプトロンによる分岐予測法は，単純な構造ながら優れた予測精度を示すものとして注目されている[1)2)3)4)5)]．しかしパーセプトロンはその構造上，線形分離不可能なパターンを学習することができないという欠点を持つ[6)]．そのため，パーセプトロン分岐予測器の予測精度の向上には，線形分離不可能なパターンを学習する手段が必要となる．

パーセプトロンが線形分離不可能なパターンを学習する手法のひとつに，冗長な入力を付加するものがある．Seznec が提案した MAC-RHSP(Multiply-Add Contribution Redundant History Skewed Perceptron Predictor)[3)] には冗長入力が適用されており，パーセプトロン分岐予測器と比較して予測精度が向上していることが示されている．しかし，冗長性をどのように用いるかという点に関しては研究が不十分であり，冗長性の用い方によっては予測精度がより向上する可能性がある．

本稿でははじめにエイリアシングの効果が除かれたパーセプトロン分岐予測器の性能として予測精度の限界を求める．次に，冗長入力の生成において，使用する分岐履歴の長さと履歴どうしの相関を形成する範囲が，予測精度にどんな影響を与えるかに着目し，これらのパラメータの下で生成された冗長入力を付加したパーセプトロン分岐予測器の予測精度をシミュレー

† 電気通信大学 情報工学科
Department of Computer Science, The University of Electro-Communications

ションにより測定する．本稿では冗長入力の付加による効果を系統的に調べることを目的とするので，ハードウェア量や予測処理時間に関しては考慮しない．

以下，2 章でパーセプトロン分岐予測の概要と関連研究を述べ，3 章で冗長入力の生成方法について述べる．4 章で予測精度の評価結果を説明し，5 章で考察を述べ，6 章でまとめる．

2. パーセプトロン分岐予測器

本章ではパーセプトロン分岐予測法の概要と，冗長入力および関連研究を述べる．

2.1 パーセプトロン分岐予測器

Jimenez らはパーセプトロンを導入した分岐予測法[1] を考案した．パーセプトロンはニューラルネットワークの一種であり，ニューロンモデルによる学習機能を備えている．ニューロンモデルには与えられた複数の入力に対応する重みが一つずつ用意されており，入力と重みの積の総和が閾値以上であれば発火し，閾値未満であれば発火しない仕組みになっている．ある入力に対する発火の有無が正しかったかどうかを判定し，正しい結果に近づくように重みを変更することによって学習する．

パーセプトロン分岐予測器では，分岐の成立を 1，不成立を -1 として，過去 N 回分の分岐結果を保持するグローバル・ヒストリ・レジスタ (GHR) を入力として用いる．それぞれの分岐結果が入力として与えられ，これらに分岐命令ごとに用意された重みがかけられる．これらの重みは，分岐命令のアドレスの下位ビットをインデックスとした表に保持されている．GHR に保持された i 番目の分岐結果を X_i，対応する重みを W_i とするとき，総和 y は

$$y = W_0 + \sum_{i=1}^{N} W_i X_i$$

と表される．なお 0 番目の入力は偏向入力として 1 に固定されている．ニューロンモデルの閾値は 0 とされ，総和の値が正ならば分岐成立と予測し，1 を出力する．総和の値が負であれば分岐不成立と予測し，-1 を出力する．図 1 にパーセプトロン分岐予測器の予測機構を示す．

学習の結果が反映されるのは重みである．重み W_i の更新規則は，真の分岐結果 $t \in \{1, -1\}$ と対応する GHR の 1 ビット $X_i \in \{1, -1\}$ を用いて，

$$W_i = W_i + \alpha t X_i$$

と表される．α は学習係数と呼ばれ，通常 1 とされる．パーセプトロン分岐予測器は，予測と学習の過程を繰り返すことで，動的な分岐予測を実現している．

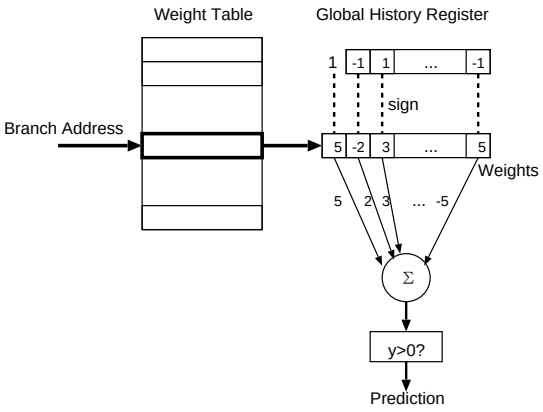


図 1 パーセプトロン分岐予測器
Fig. 1 Perceptron Branch Predictor.

2.2 冗長入力と MAC-RHSP

パーセプトロンはその構造上，線形分離不可能なパターンを学習することができない．例えば $(x_1, x_2) \in \{0, 1\}^2$ の排他的論理和は，線形分離不可能なパターンを持つため，これらを 2 入力とするパーセプトロンでは正しく学習することができない．しかし，$(x_1, x_2, x_1 \oplus x_2)$ の 3 入力のパーセプトロンならば，排他的論理和を正しく学習できるだけでなく，学習パターンのクラスが広がる．$x_1 \oplus x_2$ のような冗長入力を付加することにより，パーセプトロンでも線形分離不可能なパターンを学習することができるようになる場合がある．

Seznec はこの点に着目し，冗長入力を導入した MAC-RHSP[3] を提案した．MAC-RHSP では，4 ビットごとの履歴の相関を冗長入力として使用する．具体的には，GHR を 4 ビットずつのブロックに分割し，各ブロックにおいて 4 ビットの全ての組み合わせについて排他的論理和を求めたものを求める．あるブロックの 4 ビットを $I0, I1, I2, I3$ とすると，そのブロックからは冗長を含めて次の 16 個の入力が生成される: $0, I0, I1, I2, I3, I0 \oplus I1, I0 \oplus I2, I0 \oplus I3, I1 \oplus I2, I1 \oplus I3, I2 \oplus I3, I0 \oplus I1 \oplus I2, I0 \oplus I1 \oplus I3, I0 \oplus I2 \oplus I3, I1 \oplus I2 \oplus I3, I0 \oplus I1 \oplus I2 \oplus I3$.

MAC-RHSP は冗長入力を用いないパーセプトロン分岐予測器よりも予測精度が優れることが報告されている．

3. 冗長入力の生成

冗長入力により予測精度が向上することは示されているが，MAC-RHSP のような生成方法は根拠が不確かであり，最適な冗長入力の生成方法については研究

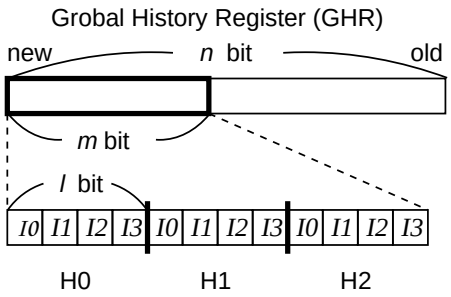


図 2 GHR の範囲選択と区切りの例
Fig. 2 Part of GHR and its range used for forming redundancy.

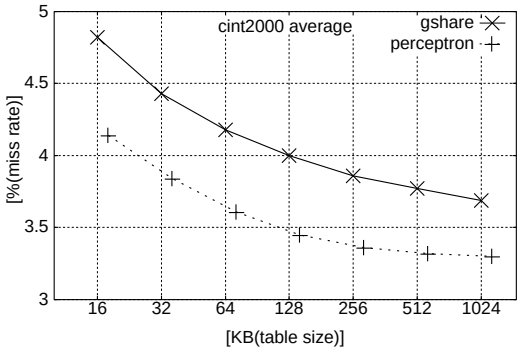


図 3 冗長入力を用いないパーセプトロン分岐予測と Gshare との比較
Fig. 3 Miss Rates of Perceptron Branch Prediction without redundant inputs and Gshare.

が不十分である．ここでは冗長入力の生成において，使用する分岐履歴の長さと履歴どうしの相関を形成する範囲が，予測精度にどんな影響を与えるかに着目する．

3.1 冗長入力の生成方法

付加される冗長入力は，MAC-RHSP と同様に GHR から生成される．生成方法を以下に示す．

(1) GHR のビット長を n，冗長の生成に用いるビットを最新の m ビットとする $(m \leq n)$
(2) GHR の部分履歴 m ビットを l ビットごとのブロックに分割する $(m \bmod l = 0)$
(3) 各ブロックにおいて l ビットの全ての組み合わせについて排他的論理和を求めたものを冗長とする
(4) 冗長生成に用いなかった GHR の残りのビットには，そのまま重みを対応づける

図 2 は，冗長の生成に GHR の最新の 12 ビットを用いる $(m = 12)$ とし，これを 4 ビットごとに区切る $(l = 4)$ とした場合の例である．このときブロックは H0，H1，H2 の 3 つとなる．各ブロックの 4 ビットを $I0, I1, I2, I3$ とすると，1 つのブロックからは MAC-RHSP と同様に計 16 個の冗長入力が生成される．

例ではブロックが 3 つあるため，冗長を含め 48 個の入力が用意されることになる．GHR の全体の長さ n が 40 ビットであった場合，冗長の生成に用いられなかった残りの 28 ビットが入力として加えられ，これらに重みが対応づけられる．

3.2 MAC-RHSP との相違点

MAC-RHSP と異なるのは，GHR の部分履歴から冗長を生成する点と，1 ブロック内のビット数が 4 ビットに限られないという点である．このような方法にするのは，MAC-RHSP の予測精度が優れているという報告を前提とした上で，冗長の生成に必要な GHR の長さがどれほどであるか，またどんな範囲の履歴の相関を用いるのが効果的なのかを調査するためである．

4. 評 価 結 果

本章では，パーセプトロン分岐予測器に冗長入力を付加しない場合と，冗長入力を付加する場合の予測精度を評価する方法と評価の結果について述べる．予測精度の測定にはプロセッサシミュレータとして SimAlpha[7] を用い，ベンチマークプログラムとして SPEC Cint2000 を使用する．

4.1 冗長入力を用いない場合

4.1.1 Gshare との比較

一般的なパーセプトロン分岐予測器の性質を見るため，はじめに冗長入力を用いないパーセプトロン分岐予測器と従来法との比較を行った．図 3 のグラフは，従来の 2 ビットカウンタによる Gshare 分岐予測器[8] と，64 入力のパーセプトロン分岐予測器について，テーブルサイズごとに Cint2000 の 11 個のプログラムの平均予測ミス率を測定した結果を示している．両者を比較すると，パーセプトロン分岐予測器のほうが予測精度が優れていることが分かる．

4.1.2 エイリアシングについて

冗長入力の効果のみが現れるようにするためには，エイリアシングによる影響を極力少なくする必要がある．そこで，どのくらいの大きさの重み表を用意すればエイリアシングが発生しなくなるかを調べた．図 4 は GHR の長さを 16，32，48，64 ビットとした場合のパーセプトロン分岐予測器について，重み表のエントリ数を変化させて予測精度を測定した結果を示す．縦軸が平均予測ミス率，横軸がエントリ数の逆数となっている．グラフの左端はエントリ数が 16384 の場合の予測ミス率を示しており，エントリ数が無限に存在

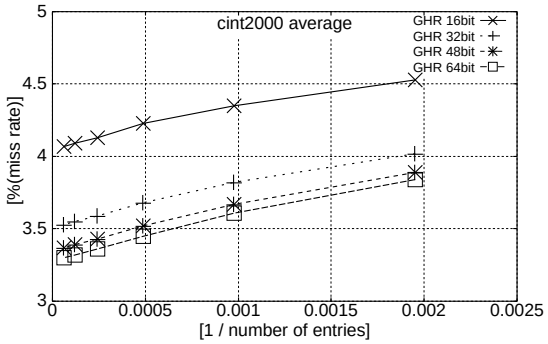


図 4 重み表のエントリ数と平均予測精度 (横軸はエントリ数の逆数)

Fig. 4 Number of entries of weight table and miss prediction. (Inverse of number of entries is taken as x axis.)

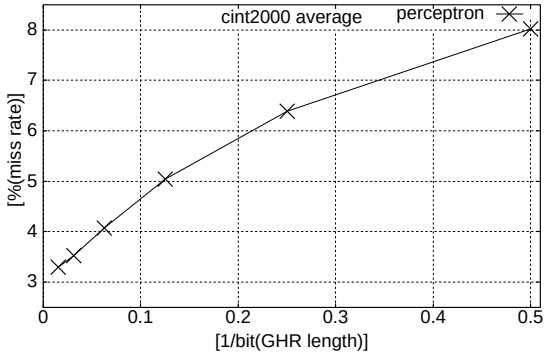


図 5 GHR の長さと予測精度 (横軸は GHR の長さの逆数)

Fig. 5 GHR length and miss prediction rate.(Inverse of GHR length is taken as x axis.)

する場合に漸近する値に近い．このことから，16384 エントリの重み表を持つパーセプトロン分岐予測器ではエイリアシングがほとんど発生していないと判断できる．

4.1.3 パーセプトロン分岐予測器の限界

エントリ数を 16384 としたときのパーセプトロン分岐予測器について予測精度を測定した結果を図 5 に示す．縦軸が平均予測ミス率，横軸が GHR のビット長の逆数である．図から，無限長の GHR に対し，予測ミス率はおよそ 3.1%に漸近することが分かる．パーセプトロン分岐予測器では予測ミスを 3.1%以下にすることは難しいと言える．

4.2 冗長入力を用いる場合

冗長入力を付加したパーセプトロン分岐予測器の構成は次のとおりである．

- 重み表のエントリ数は 16384
- 重みのビット数は 9
- GHR の履歴ビット長 n は 64
- 冗長の生成に用いる GHR の部分履歴 m は最新履歴から 16，32，48，64 ビットの 4 種類
- 冗長の生成に用いる GHR の部分履歴を分割するブロックのビット数 l は 3，4，5 の 3 種類

$l = 3, 5$ の場合，冗長の生成に使用する GHR のビット長を割り切れないため，条件にできるだけ近くなるよう，$l = 3$ では $m = 15, 30, 48, 63$ を，$l = 5$ では $m = 15, 30, 45, 60$ を冗長の生成に使用する GHR のビット長とする．また，比較のために冗長入力を全く用いないパーセプトロン分岐予測器についても予測精度を測定する．この場合，エントリ数，重みのビット数，GHR の履歴ビット長は冗長入力を付加したものと同じ条件であるとする．

図 6 に，各ベンチマークプログラムによる予測精度の測定結果を示す．横軸は冗長の生成に用いた GHR のビット長 m，縦軸は予測制度のミス率を表す．グラフは各 m に対し，1 ブロックのビット長 l を変えて測定した結果を表している．$m = 0$ は冗長入力を用いない場合である．

どのベンチマークプログラムにおいても，m が 48，64 ビットの場合に，冗長入力を用いないパーセプトロン分岐予測器よりも優れた精度を示す．予測精度が最も向上するのは crafty における $m = 64, l = 5$ の場合で，冗長入力を用いない場合と比較して予測ミス率がおよそ 30%低くなった．

部分履歴 m が 48 ビットと 64 ビットの場合を比較すると，64 ビットのほうが 0〜0.15%予測ミス率が低い．部分履歴 m が 16，32 ビットの場合では，どのベンチマークプログラムにおいてもパーセプトロンよりも予測精度が劣る結果となった．部分履歴が 16 ビットの場合は予測精度が特に悪く，gap での 1 ブロック 3 ビットの場合ではパーセプトロンの場合と比較して予測ミス率がおよそ 6 倍になった．

予測ミス率が最も小さくなるのは，どのベンチマークプログラムにおいても部分履歴 64 ビット・1 ブロック 5 ビットの場合であった．逆に予測ミス率が最も大きくなるのは，部分履歴 16 ビット・1 ブロック 3 ビットの場合であった．

以上より，冗長入力の付加によって予測精度に以下のような効果が得られることが分かった．

- 冗長の生成に用いる GHR の部分履歴の長さが 32 ビット未満では，予測精度は悪化する
- 冗長の生成に用いる GHR の部分履歴の長さが 48 ビット以上であれば，予測精度は向上する
- ブロックのサイズは小さいよりも大きいほうが予

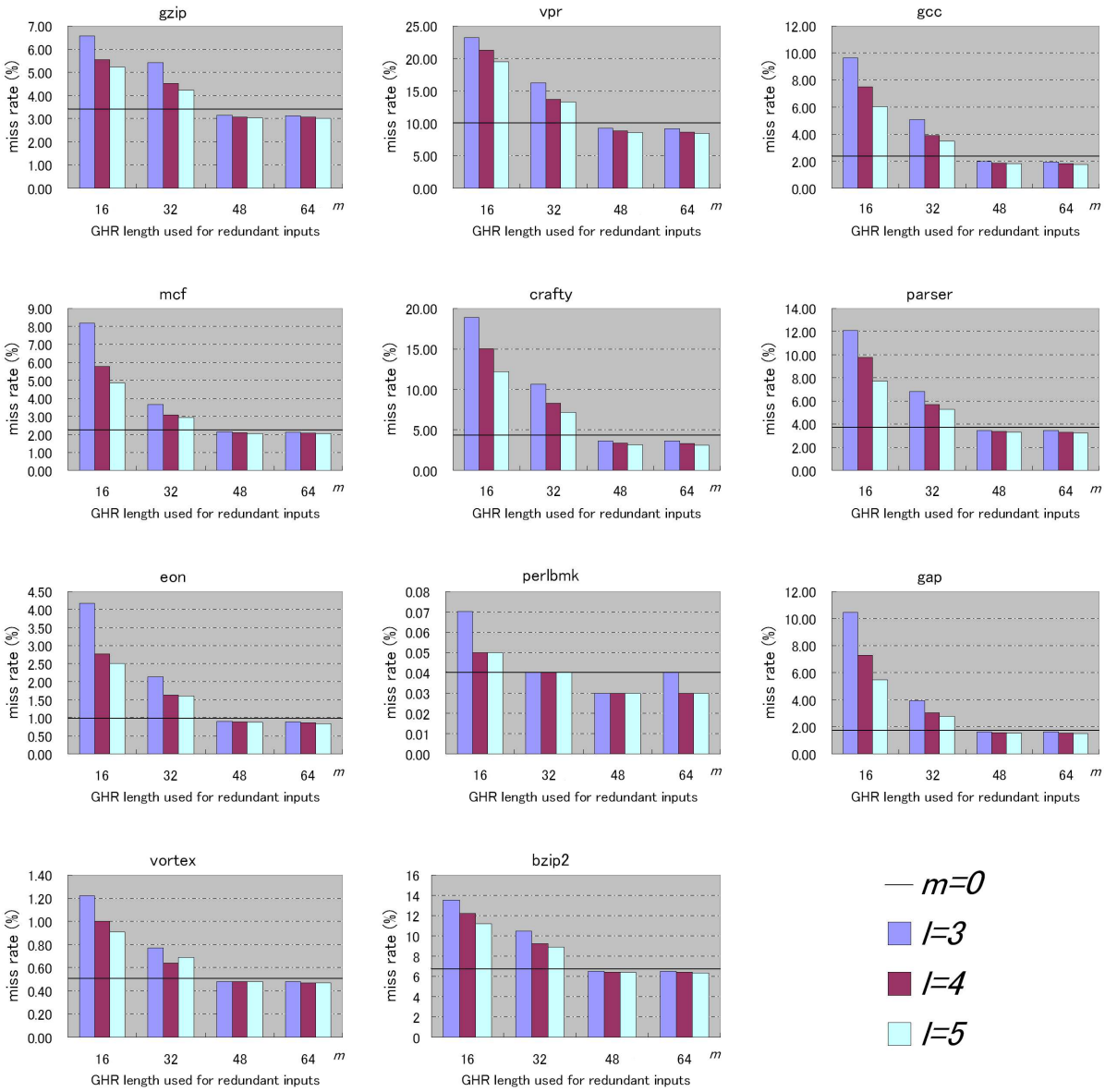


図 6　冗長入力を付加したパーセプトロン分岐予測器による予測精度
Fig. 6　Miss rate of perceptron branch predictor with various redundant inputs.

測精度が良い

5. 考　　察

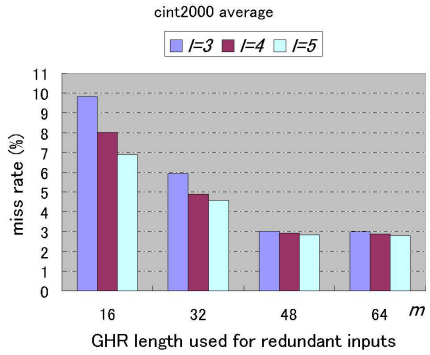
5.1　冗長入力の付加による効果

図 6 において，$m = 64, l = 5$ における平均ミス率は 2.8%である．パーセプトロン分岐予測法では，履歴長をどれだけ長く使用しても予測ミス率はおよそ 3.1%である (図 5)．以上より，パーセプトロン分岐予測法ではどうしても予測できないパターンの平均約 10%を，冗長入力の付加によって予測できるようになることが分かる．

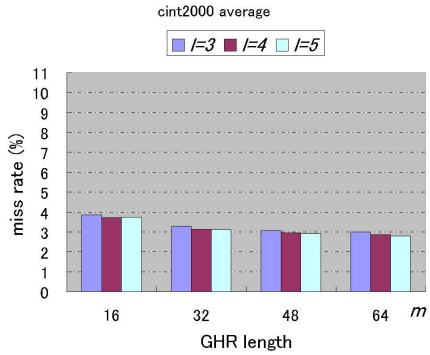
また，ブロックのサイズ l が大きいほうが予測精度が良いことから，履歴の相関を多く取ることが予測精度の向上に貢献すると考えられる．

5.2　冗長の生成に用いる GHR の長さについて

測定結果から，冗長の生成に用いる GHR の部分履歴の長さが短い場合は，予測精度が悪化することが分かる．



(a)



(b)

図 7 (a) 冗長入力を GHR の部分から生成する場合 ($m \leq n = 64$); (b)GHR の全体から生成する場合 ($n = m$)

Fig. 7 (a)Average miss rates when redundant inputs are formed from part of GHR($m \leq n = 64$); (b)from whole of GHR($m = n$).

冗長の生成に用いない GHR の履歴ビットをパーセプトロンの入力から削除した場合 ($n = m$) について予測精度を測定した．図 6 から求めた平均予測ミス率を図 7(a) に，$n = m$ とした場合の平均予測ミス率を図 7(b) に示す．これらを比較すると，$m = 16, 32$ では $n = m$ の場合のほうが予測精度が良い．

以上より，冗長入力の付加により予測精度を向上させるには，GHR を十分広く使用して冗長入力を生成する必要があることが分かる．

このような結果となった原因として，予測器の分類能力や学習アルゴリズムの収束性が悪化したことが考えられる．冗長入力の付加の仕方は仮説空間に影響を与える．つまり，冗長入力の与え方は，予測器の分類能力を変化させる．GHR の長さも予測器の分類能力を変化させる．重みを整数にしていることも，予測器の分類能力をオリジナルのものよりも劣化させてしまっている可能性が高い．また，学習アルゴリズムの収束性が厳密にオリジナルの場合と同様に成立するかどうかは，検討が必要である．学習係数 $\alpha = 1$ に固定していることは，学習の収束速度に影響すると考えられる．

こうした学習理論の観点による解析が今後の課題である．

6. おわりに

エイリアシングをなくしたパーセプトロン分岐予測期において，GHR の長さを無限長へ外挿すると予測ミス率が平均約 3.1%となるのに対し，64 ビットの GHR を用いて冗長入力を付加することで予測ミス率が最大 10%低減することを示した．また，冗長入力の生成において，GHR のできるだけ多くのビットを冗長入力の生成に用いるのが望ましいこと，さらにブロックのサイズを大きくして履歴ビットの相関を多く使用するほうが予測精度が良いことが分かった．分岐命令が学習するのに必要な実行履歴の長さと重みの数や学習係数との関連については今後の課題である．

謝辞 本研究において，シミュレータ SimAlpha を提供してくださった電気通信大学大学院情報システム学研究科の吉瀬謙二博士に感謝いたします．

参 考 文 献

1) D. A. Jimenez, and C. Lin : Dynamic branch prediction with perceptrons, 7th International Symposium on High Performance Computer Architecture, pp.197-206(2001)
2) S. Kim : Branch prediction using advanced neural methods, Technical Report, University of California, Berkeley(2003)
3) A. Seznec : Revisiting the perceptron predictor, Technical Report 1620, IRISA(2004)
4) D. Tarjan, and K. Skadron : Revisiting the Perceptron Predictor Again, Technical Report CS-2004-28, University of Virginia(2004)
5) H. Gao, and H. Zhou : Adaptive information processing: an effective way to improve perceptron predictors, In 1st Championship Branch Prediction Workshop, in conjunction with the 37th Intl. Symposium on Microarchitecture(2004)
6) M. Minsky, and S. Papert : Perceptrons: an introduction to computational geometry, MIT Press, MA(1969)
7) 吉瀬謙二，本多弘樹，弓場敏嗣 : SimAlpha: シンプルで理解しやすいコード記述を目指した Alpha プロセッサシミュレータ, Tecnical Report UEC-IS-2002-2, 電気通信大学大学院情報システム学研究科 (2002)
8) S. McFarling : Combining branch predictors, Technical Report TN-36, Digital Western Research Laboratory(1993)