A Linear Lower Bound for the Perceptron for Input Sets of Constant Cardinality.

Research Note RN/08/03

(Updated 18 May, 2008)

Mark Herbster
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, England, UK
m.herbster@cs.ucl.ac.uk

May 18, 2008

Abstract

The perceptron may incur mistakes linear in the dimension of the input for an input set of cardinality three for the target concept of a single literal.

1 A lower bound

In this research note we show that the Perceptron with an input set $X := \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ with inputs,

$$\mathbf{x}_1 := (1, \underbrace{1, \ldots, 1}_{n-1}),\ \mathbf{x}_2 := (1, \underbrace{1, \ldots, 1}_{\frac{3(n-1)}{4}}, \underbrace{-1, \ldots, -1}_{\frac{n-1}{4}}),$$
$$\mathbf{x}_3 := (1, \underbrace{-1, \ldots, -1}_{\frac{3(n-1)}{4}}, \underbrace{1, \ldots, 1}_{\frac{n-1}{4}}), \qquad (1)$$

may incur $\Theta(n)$ mistakes for the target concept of a single literal. Very general linear lower bounds for a wide class of algorithms (with $|X| = n$) including the perceptron and online linear least squares for this problem have been given in [KWA97, WV05]. Novikoff's Theorem [Nov63] for the perceptron gives an upper bound of n mistakes for this problem as the squared radius of the input space is n and the margin of the optimal classifier is one. The Winnow [Lit88] algorithm gives an an upper bound of $\Theta(\log n)$ for this problem. Any algorithm that "memorizes" the input can obtain a constant upper bound for this problem such as online linear least squares. In Figure 1 we recall the primal Perceptron algorithm.

```
**Input:** ${(\mathbf{x}_t, y_t)\}_{t=1}^{\ell} \subseteq \mathbb{R}^n \times \{-1,1\}$.
**Initialization:** $\mathbf{w}_1 = \mathbf{0}$; mistakes $= 0$.
**for** $t = 1, \dots$ **do**
  **Predict:** receive $\mathbf{x}_t$
  $\hat{y}_t = \text{sign}(\mathbf{x}_t^\top \mathbf{w}_t)$
  **Update:** receive $y_t$
  **if** $\hat{y}_t = y_t$ **then**
    $\mathbf{w}_{t+1} = \mathbf{w}_t$
  **else**
    $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t \mathbf{x}_t$
    mistakes $=$ mistakes $+ 1$
**end**
```

Figure 1: Perceptron Algorithm (Primal)

Theorem 1. *The Perceptron algorithm incurs $\frac{n+1}{2}$ mistakes if the inputs are selected uniformly at random from the set X (see equation (1)) with labels $y_1 = y_2 = y_3 = 1$ and if $\mathbf{x}_1$ is the first input selected in an unbounded trial sequence.*

Proof. As the hypothesis vector $\mathbf{w}$ of the perceptron is a linear combination of inputs; we will summarize through a simplifying state diagram. The state $\mathbf{w}$ is determined by three quantities (a, b, c) where

$$\mathbf{w} = (a, \underbrace{b, \dots, b}_{\frac{3(n-1)}{4}}, \underbrace{c, \dots, c}_{\frac{n-1}{4}}) . \tag{2}$$

The state diagram (Figure 2) shows the dynamics of the perceptron with input-label pairs $\{(\mathbf{x}_1, 1), (\mathbf{x}_2, 1), (\mathbf{x}_3, 1)\}$ given the followings simplifications,

1. The first input is assumed to be $\mathbf{x}_1$.

2. The pair in each node are the values of (b, c).

3. The perceptron converges at mistake $\frac{n+1}{2}$.

Every transition in the diagram between distinct states corresponds to a mistake. For every mistake a is incremented by one. Furthermore the transitions $(1,1) \Rightarrow (0,2)$ and $(1,1) \Leftarrow (0,2)$ can only occur if $a \leq \frac{n-1}{2}$ thus the exact bound of $\frac{n+1}{2}$ mistakes. $\square$

2 A connection to graph label prediction

We observe that the inputs corresponding implicitly to a *signed graph Laplacian* [Her08]. First, set $r = n - 1$ and compute the dual (kernel) matrix,

$$\mathbf{K} = \begin{pmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_2 \rangle & \langle \mathbf{x}_1, \mathbf{x}_3 \rangle \\ \langle \mathbf{x}_2, \mathbf{x}_1 \rangle & \langle \mathbf{x}_2, \mathbf{x}_2 \rangle & \langle \mathbf{x}_2, \mathbf{x}_3 \rangle \\ \langle \mathbf{x}_3, \mathbf{x}_1 \rangle & \langle \mathbf{x}_3, \mathbf{x}_2 \rangle & \langle \mathbf{x}_3, \mathbf{x}_3 \rangle \end{pmatrix} = \begin{pmatrix} 1+r & 1+\frac{r}{2} & 1-\frac{r}{2} \\ 1+\frac{r}{2} & 1+r & 1-r \\ 1-\frac{r}{2} & 1-r & 1+r \end{pmatrix} . \tag{3}$$

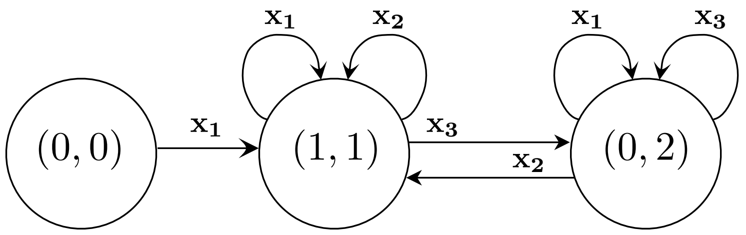

Figure 2: State Diagram

Now we compute the primal matrix $\mathbf{M} = \mathbf{K}^{-1}$

$$\mathbf{M} = \begin{pmatrix} \frac{4}{3r} & -\frac{1}{r} & -\frac{1}{3r} \\ -\frac{1}{r} & \frac{1}{4} + \frac{1}{r} & \frac{1}{4} \\ -\frac{1}{3r} & \frac{1}{4} & \frac{1}{4} + \frac{1}{3r} \end{pmatrix} . \tag{4}$$

Thus $\mathbf{M}$ is a symmetrically diagonally dominant matrix with a positive diagonal hence a signed Laplacian. Therefore the Pounce algorithm [Her08, Theorem 1] will incur four mistakes on this example. Finally we observe although Pounce is described in dual form the algorithm may be equally implemented in the primal form with no need to know the inputs in advance (neither $\mathbf{M}$ or $\mathbf{K}$ is needed). In the primal case the Pounce bound will be obtained if the inputs *after the fact* determine a primal matrix $\mathbf{M}$ which is either a irreducible Laplacian or a positive definite signed Laplacian.

References

[Her08] Mark Herbster. Exploiting cluster-structure to predict the labeling of a graph. Submitted for review. Available on request., 2008.

[KWA97] J. Kivinen, M. K. Warmuth, and P. Auer. The perceptron algorithm vs. winnow: linear vs. logarithmic mistake bounds when few input variables are relevant. *Artificial Intelligence*, 97:325–343, December 1997.

[Lit88] N. Littlestone. Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[Nov63] A. Novikoff. On convergence proofs for perceptrons. In *Proc. Sympos. Math. Theory of Automata (New York, 1962)*, pages 615–622. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963.

[WV05] Manfred K. Warmuth and S. V. N. Vishwanathan. Leaving the span. In Peter Auer and Ron Meir, editors, *COLT*, volume 3559 of *Lecture Notes in Computer Science*, pages 366–381. Springer, 2005.