Neural Network Learning: Theoretical Foundations Chapter 24 $\sim$ 26

Martin Anthony and Peter L. Bartlett

Presenter: Yongchan Kwon

Department of Statistics, Seoul National University, Seoul, Korea

ykwon0407@snu.ac.kr

Reviews; pp.312-313

Definition (RP)

The class of decision problems that can be solved by a polynomial-time randomized algorithm is denoted by RP.

Definition (H-FIT)

Instance: $z \in (\mathbf{R}^n \times \{0,1\})^m$ and an integer k between 1 and m.
Question: Is there $h \in H_n$ such that $\hat{er}_z(h) \leq k/m$?
where H_n is a class of a binary function on n-dimensional inputs.

Reviews; pp.312-313

Theorem (23.7)

Let $H = \cup_n H_n$ be a graded binary function class. If there is an efficient learning algorithm for H, then there is a polynomial time randomized algorithm for H-FIT; in other words, H-FIT is in RP.

Theorem (23.8)

Suppose $RP \neq NP$ and that H is a graded class of binary functions. If H-FIT is NP-hard, then there is no efficient learning algorithm for H.

Contents

Ch. 24: The Boolean Perceptron

Learning is Hard for the Simple Perceptron

Definition (BP-FIT)

Instance: $z \in (\{0,1\}^n \times \{0,1\})^m$ and an integer k between 1 and m.
Question: Is there $h \in BP_n$ such that $\hat{er}_z(h) \leq k/m$?
where BP_n is the set of boolean function from $\{0,1\}^n$ to $\{0,1\}$ computed by the boolean perceptron, and $BP = \cup_n BP_n$.

Definition (Simple perceptron)

A simple perceptron is a function $f : \mathbf{R}^n \to \{0,1\}$ of the form

$$f(x) = \begin{cases} 0, & \text{if } w^T x - \theta < 0 \\ 1, & \text{if } w^T x - \theta \geq 0 \end{cases}$$

for input vector $x \in \mathbf{R}^n$, $w \in \mathbf{R}^n$, and $\theta \in \mathbf{R}$.

Learning is Hard for the Simple Perceptron

Theorem (24.2)

BP-FIT is NP-hard.

Key idea: The problem is at least as hard as a well-known NP-hard problem in the field of graph theory.

Vertex cover problem [NP-hard]

A vertex cover of the graph is a set U of vertices such that for each edge (i, j) of the graph, at least one of the vertices i, j belongs to U.
Instance: A graph $G = (V, E)$ and an integer $k \leq |V|$
Question: Is there a vertex cover $U \subset V$ such that $|U| \leq k$?

Corollary (24.3)

If RP $\neq$ NP, then there is no efficient learning algorithm for BP.

Learning is Easy for Fixed Fan-In Perceptrons

- The previous theorem shows that learning the simple perceptron is difficult. We consider simpler perceptrons in which the number of non-zero weights is constrained.

Definition (fan-in)

A simple perceptron with weights $w \in \mathbf{R}^n$ and threshold $\theta \in \mathbf{R}$ has fan-in k if the number of non-zero components of w is no more than k.

Pseudocode for the *Splitting* procedure

```
argument:  Training set, S = {x_1,...,x_m} ⊂ R^n
returns:   Set of weights and thresholds, W = {(w,θ)}

function Splitting(S)
    W := ∅
    P := ∅
    for all t_1 < ··· < t_k from {1,...,n}
        for all l from {1,...,k+1}
            for all r_1 < ··· < r_l from {1,...,m}
                for all α_1,···,α_l from {±1}
                    if there is a solution (w,θ) to the system
                        of linear equations
                        x_{r_i} · w + θ = α_i          i = 1,...,l
                        satisfying
                        {i : w_i ≠ 0} = {t_1,t_2,...,t_k}
                    then
                        S' := {x ∈ S : w·x − θ < 0}
                        S'' := {x ∈ S : w·x − θ ≥ 0}
                        if {S',S''} ∉ P
                        then
                            W := W ∪ {(w,θ)}
                            P := P ∪ {S',S''}
                        endif
                    endif
                endfor
            endfor
        endfor
    endfor
    return W
end
```

Learning is Easy for Fixed Fan-In Perceptrons

Theorem (24.4)

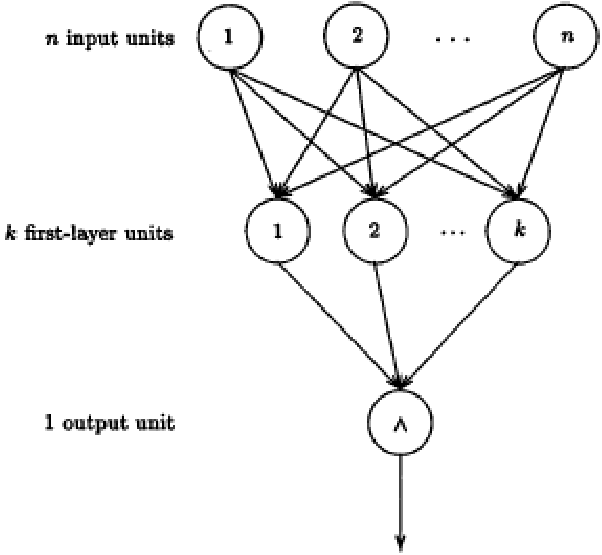
The procedure **Splitting** *returns all dichotomies of its arguments* $S \subset \mathbf{R}^n$ *that can be computed by some simple perceptron with fan-in no more than k. For* $|S| = m$*, it takes time* $O(n^{2k}2^k m^{2k+3})$

Corollary (24.5)

For fixed k*, define the graded class* $H^k = \cup_n H_n^k$*, where* H_n^k *is the class of simple perceptrons defined on* $\mathbf{R}^n$ *with fan-in no more than* k*. The class* H^k *is efficiently learnable.*

Ch. 25: Hardness Results for Feed-Forward Networks

Linear Threshold Networks with Binary Inputs



Linear Threshold Networks with Binary Inputs

- Let $N^k_{\wedge,n}$ be a neural network on n binary inputs and $k+1$ linear threshold units. Further, we only consider $N^k_{\wedge,n}$ has two layers of computation units, the first consisting of k linear threshold units.
- The output unit is also a linear threshold unit, with a connection of fixed weight 1 from each of the other k threshold units.
- Consider the graded space $N^k_{\wedge} = \cup_n N^k_{\wedge,n}$.

$N^k_{\wedge} - CONSISTENCY$

Instance: $z \in (\{0,1\}^n \times \{0,1\})^m$
Question: Is there $h \in N^k_{\wedge,n}$ such that $\hat{er}_z(h) = 0$?

Linear Threshold Networks with Binary Inputs

Corollary (25.2)

Let $k \leq 3$ be any fixed integer. Then, $N^k_\wedge - CONSISTENCY$ is NP-hard.

Key idea: Again, the problem is at least as hard as a well-known NP-hard problem in the field of graph theory.

$k - colouring$ [NP-hard]

A $k - colouring$ of G is a function $\chi : V \to \{1, 2, ..., k\}$ with the property that whenever $(i, j) \in E$, then $\chi(i) \neq \chi(j)$.
Instance: A graph G
Question: Does G have a $k - colouring$?

Corollary (25.3)

Unless $RP = NP$, there is no efficient learning algorithm for the graded class $H = \cup_n H_n$, where H_n is the set of functions computable by $N^k_{\wedge,n}$.

Linear Threshold Networks with Real Inputs

- The result of the previous section is limited, since it shows that learning is difficult for a rather unusual network class. But....

Theorem (25.4)

Unless $RP = NP$, there is no efficient learning algorithm for the graded class $H = \cup_n H_n$, where H_n is the set of functions computable by N_n^k, a network with n real inputs.

- Similar results are obtained for sigmoid networks. (chapter 25.4).

Ch. 26: Constructive Learning Algorithms for Two-Layer Networks

Real Estimation with Convex Combinations of Basis Functions

- We consider learning algorithms for classes F of real valued functions that can be expressed as convex combinations of functions from some class G of basis functions.
- Some boosting and neural networks classes are example of F under some constraints

Real Estimation with Convex Combinations of Basis Functions

Theorem (26.1)

Let V be a vector space with an inner product, and let $\|f\| = \sqrt{(f, f)}$ be the induced norm on V. Suppose that $G \subset V$ and that, for some $B > 0$, $\|g\| \leq B$ for all $g \in G$. Fix $f \in V, k \in \mathbf{N}$, and $c \geq B^2$, and define $\hat{f}_0 = 0$. Then for $i = 1, \ldots, k$, choose $g_i \in G$ such that

$$\|f - \hat{f}_i\|^2 \leq \inf_{g \in G} \|f - ((1 - \alpha_i)\hat{f}_{i-1} + \alpha_i g)\|^2 + e_i,$$

where $\alpha_i = 2/(i+1), e_i \leq 4(c - B^2)/(i+1)^2$, and $\hat{f}_i = (1 - \alpha_i)\hat{f}_{i-1} + \alpha_i g$. Then,

$$\|f - \hat{f}_k\|^2 < \inf_{\hat{f} \in co(G)} \|f - \hat{f}\|^2 + \frac{4c}{k}.$$

Real Estimation with Convex Combinations of Basis Functions

Note that $\|f - ((1-\alpha_i)\hat{f}_{i-1} + \alpha_i g)\|^2 = \alpha_i^2 \|\tilde{f} - g\|^2$, where $\tilde{f} = (f - (1-\alpha_i)\hat{f}_{i-1})/\alpha_i$. This suggests using an approximate-SEM algorithm for a class G to approximately minimize sample error over the class $co(G)$.

Corollary (26.2)

Suppose that $G = \cup_n G_n$ is a graded class of real-valued functions that map to some bounded interval, and L is an efficient approximate-SEM algorithm for G, with running time $O(p(m, n, 1/e))$ for some polynomial p. Then, the algorithm $Construct_L$ can be used as the basis of an efficient approximate-SEM algorithm for $co(G) = \cup_n co(G_n)$, and this algorithm has running time $O(p(m, n, 1/e)/e)$.

Pseudocode for the *Construct* procedure

```
arguments: Training set, S = {(x_1, y_1), ..., (x_m, y_m)} ⊂ (X × ℝ)^m
           Number of iterations, k
           Bound, B, on range of functions in G
returns:   Convex combination of functions from G, f̂_k = Σ_{i=1}^k γ_k g_k

function Construct_L(S, k, B)
    f̂_0 := 0
    for i := 1 to k
        α_i := 2/(i + 1)
        for j := 1 to m
            ỹ_j = (1/α_i)(y_j − (1 − α_i) f̂_{i−1}(x_j))
        end for
        S̃ = {(x_1, ỹ_1), ..., (x_m, ỹ_m)}
        g_i := L(S̃, B^2)
        f̂_i := (1 − α_i) f̂_{i−1} + α_i g_i
    endfor
    return f̂_k
end
```

Fig. 26.1. Pseudocode for the $\texttt{Construct}_L$ algorithm. (L is an approximate-SEM algorithm for $G \subset [-B, B]^X$.)

Real Estimation with Convex Combinations of Basis Functions

Let $G = BH \cup -BH$, with $H = \{sgn(w^T x + w_0) : w \in \mathbf{R}^n, w_o \in \mathbf{R}\}$, and $BH = \{B \times h : h \in H\}$. Then, $F = co(G)$ is the class of two-layer networks with linear threshold units in the first layer and a linear output unit.

Theorem (26.6)

Let H_n^k be the set of k fan-in linear threshold functions, and let $F_n^k = co(BH_n^k \cup -BH_n^k)$. Then, the algorithm Construct, based on the algorithm Splitting, is an efficient learning algorithm for the graded class $F^k = \cup_n F_n^k$.