EECS 598-005: Theoretical Foundations of Machine Learning **Fall 2015**

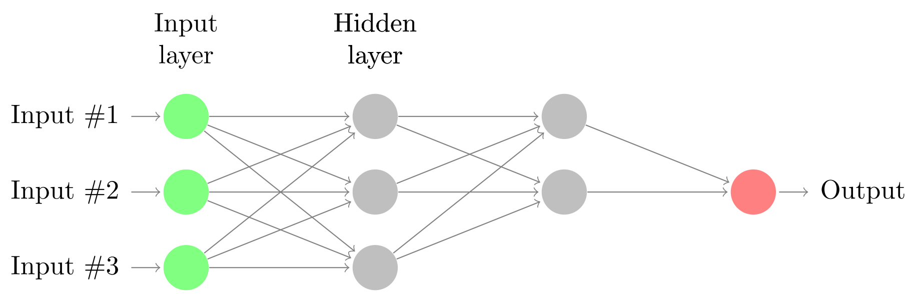
Lecture 15: Neural Networks Theory

Lecturer: Matus Telgarsky *Scribes: Xintong Wang, Editors: Yuan Zhuang*

15.1 Neural Networks Definition and Overview

15.1.1 Neural Networks Definition

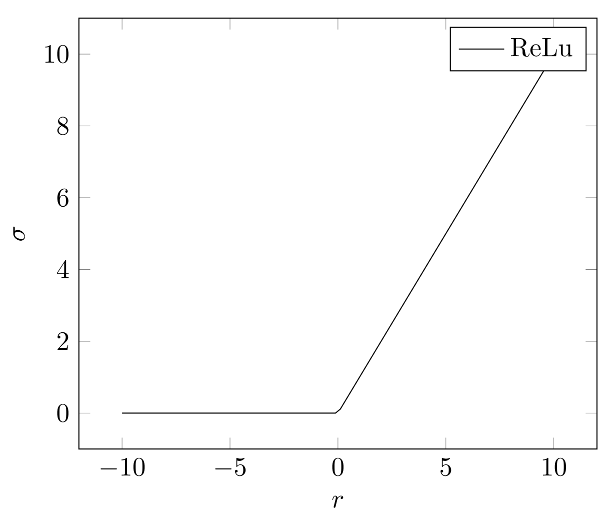
Definition 15.1 (Neural Network). *A* ***neural network*** *is a function class defined by a DAG as follows:*



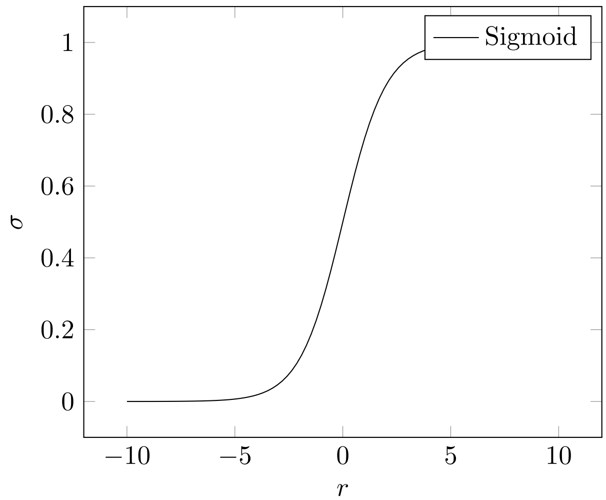
Note: A neural network is not necessarily fully connected.

- Input layer nodes: $g_i(x) = x_i$.
- Internal nodes: $g_i(x) = \sigma(\sum_{g \in \text{parents of } g_i} w_g^i g(x) + w_0^i)$.
- Typical choices of activation function σ:

  ReLu: $\sigma_R(r) = \max\{0, r\}$.



  Sigmoid: $\sigma_S(r) = \frac{1}{1+\exp(-r)}$.



- Output: Neural networks outputs value at output node and it can be multivariate.
- $\mathcal{N} := \{\text{fixed DAG with varying weights}\} = \{\mathbf{x} \to f(\mathbf{x}; \mathbf{w}) : \mathbf{w} \in \mathbb{R}^p\}$, where p is the total number of weights and thresholds.

Remark: Current practical neural networks have some variations.

15.1.2 Theory Overview

We have three aspects of theory (for classification specific setting):

- Approximation/Representation
  How well the function class fits the problem?
  Known: Neural networks can fit "arbitrary" continuous functions.
  Unknown: Are good representations learnable? What should be the size and the number of layers of a good representation?

- Optimization
  How well you optimize risk over the function class (on a finite sample)?
  Little theory in this aspect: training O(1) size neural network is NP-hard; in practice, we use gradient descent (on a nonconvex function).

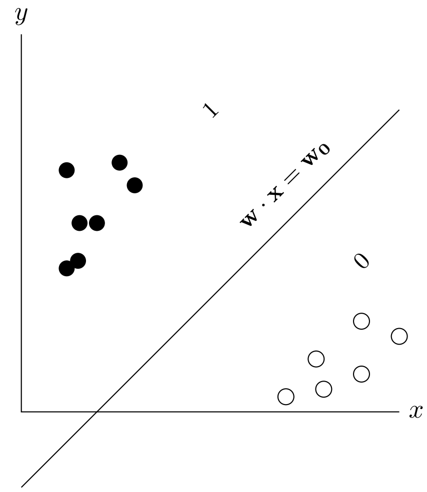
- Estimation
  Difference in risk between sample and distribution.
  There is a nice VC theory, which is very difficult. It is still unclear what this VC theory means in practice.

Remark: In terms of non-classification problems, many successes of neural networks are for unsupervised learning.

15.2 Representation/Approximation

15.2.1 Neural Network with One Internal Node

Suppose we have only one internal(non-input) node in a neural network. With activation function $\sigma(r) = \mathbb{1}[r \geq 0]$, we can exactly represent the indicator function for a halfspace.
Specifically, the following diagram shows the hyperplane $\{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w_0}\}$ and a classification achieved by the corresponding halfspace indicator $\mathbf{x} \mapsto \sigma(\langle \mathbf{w}, \mathbf{x} \rangle - \mathbf{w_0})$.



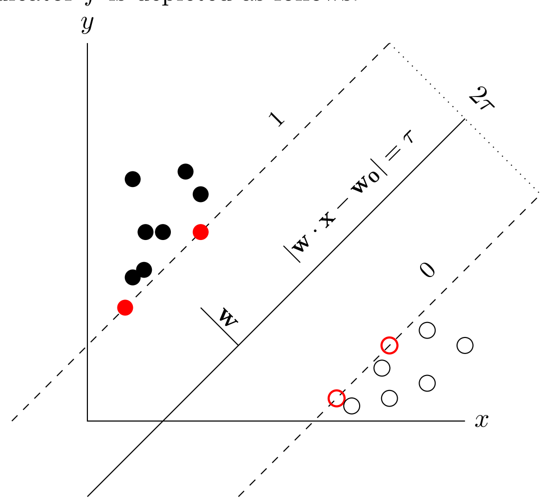
Now suppose σ is bounded and continuous, with $\lim_{r\to\infty} \sigma(r) = 1$ and $\lim_{r\to-\infty} \sigma(r) = 0$. Consequently, there exists M such that

$$\forall r > M,\ \sigma(r) \in [1-\epsilon, 1+\epsilon] \text{ and } \forall r < -M,\ \sigma(r) \in [-\epsilon, \epsilon]$$

Thus, given any $\tau > 0$, the function $f(\mathbf{x}) := \sigma(\frac{M}{\tau}(\langle \mathbf{w}, \mathbf{x} \rangle - \mathbf{w}_0))$ approximates the earlier halfspace indicator in the following sense:

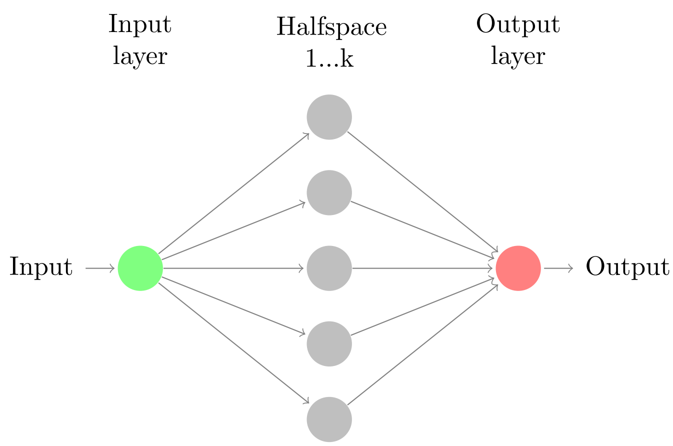
$$f(\mathbf{x}) \in \begin{cases} [-\epsilon, +\epsilon] & \text{when } \langle \mathbf{w}, \mathbf{x} \rangle \leq \mathbf{w}_0 - \tau, \\ [1-\epsilon, 1+\epsilon] & \text{when } \langle \mathbf{w}, \mathbf{x} \rangle \geq \mathbf{w}_0 + \tau. \end{cases}$$

On the other hand, f is not controlled in any way when $|\langle \mathbf{w}, \mathbf{x} \rangle - \mathbf{w}_0| < \tau$. This approximate halfspace indicator f is depicted as follows.



15.2.2 Neural Network with k Halfspaces

Next note how a polyhedron P can be approximated by adding another layer to the preceding construction. Suppose the polyhedron is specified via k halfspaces, and each of these is approximated as before with some $\tau \in (0, \frac{1}{4k})$, giving a function g_i.



Now consider adding a final node defined as $g(\mathbf{x}) := \sigma(\frac{M}{\tau}(\sum_i g_i(\mathbf{x}) - (k - 1/2)))$. To see that this approximates an indicator on P, consider any $\mathbf{x}$ which does not fall within the error region of width 2τ of any of the preceding approximate halfspace indicators.

If this $\mathbf{x}$ also fails to land within at least one of the halfspaces, then $\sum_i g_i(\mathbf{x}) \leq (k-1)(1+\tau) < k - 1/2 - \tau$, thus $g(\mathbf{x}) \in [-\epsilon, +\epsilon]$.On the other hand, if $\mathbf{x}$ is in the intersection of the halfspaces, then $\sum_i g_i(\mathbf{x}) \geq k(1-\tau) > k - 1/2 + \tau$, thus $g(\mathbf{x}) \in [1-\epsilon, 1+\epsilon]$.

Homework Problem: based on what we have said so far, for any continuous $f : [0,1]^d \to [0,1]$ and any $\epsilon > 0$, there exists a 3 layer (+input layer) neural network g with $\int_{[0,1]^d} |f(x) - g(x)| dx \leq \epsilon$.

15.3 Estimation/Statistics

The main results in this section will be VC dimension bounds which are polynomial in the number of layers L. Before giving these bounds, note that it is not clear how to specify a meaningful Rademacher complexity bound which is polynomial in the number of layers.
For instance, consider a fixed DAG representing the layout of a neural network, and suppose the weights $\mathbf{w} \in \mathbb{R}^p$ obey a constraint $\|w\|_1 \leq B$ (which is natural for instance in the case of linear separators). Then by placing weight $\frac{B}{L}$ along each edge of a chain in the network, the constraint $\|w\|_1 \leq B$ is obeyed, and it can be seen as the Lipschitz constant of the function computed by the network and thus its Rademacher complexity are upper bounded by $(\frac{B}{L})^L$.

15.3.1 VC dimension of Linear Threshold Network(LTN)

Notations

- S: a fixed set of n examples.

- k: number of non-input nodes in the neural network. d is the total number of input nodes.
- p: total number of parameters in the neural network. p_i is the number of parameters involving node i. $p = \sum_i p_i$. Moreover, suppose $p \leq n$.
- L: total number of layers in the neural network.
- $D_i :=$ |All output values possible for non-input nodes up to i|, meaning

  $$D_i = |\{(g_{d+1}(S;w), g_{d+2}(S;w), \ldots, g_{d+i}(S;w)) : w \in \mathbb{R}^p\}| \, .$$

  This definition is the key idea of the proof: rather than keeping track of the possible outputs of the output node of the network, the outputs of *all* nodes are considered.

It will now be shown by induction that

$$D_i \leq \prod_{j=1}^{i} \left(\frac{en}{p_j}\right)^{p_j} .$$

Consider the base case $i = 1$; by Sauer's Lemma, since $n \geq p \geq p_1$,

$$D_1 \leq \left(\frac{en}{p_1}\right)^{p_1} .$$

Now consider some $i > 1$. Even though S is fixed, it is no longer the case that (non-input) node i is receiving a fixed set of inputs, since it is potential taking input from non-input nodes. However, for any *fixed* outputs of the parents to node i, Sauer-Shelah can once again be used (once again using $n \geq p \geq p_i$). Combining this with the inductive hypothesis,

$$\begin{aligned} D_i &\leq \sum_{\text{possible inputs to } i} \left(\frac{en}{p_i}\right)^{p_i} \\ &\leq D_{i-1} \left(\frac{en}{p_i}\right)^{p_i} \\ &\leq \prod_{j=1}^{i} \left(\frac{en}{p_j}\right)^{p_j} . \end{aligned}$$

To complete the VC dimension calculation, first note that every distinct output for the network also increments the value of D_k, $\Pi_{\mathcal{N}}(n) \leq \sup_{|S|=n} D_k$, where $\mathcal{N}$ denotes this class of networks. Secondly, note that D_k does not actually depend on S, the upper bound holds for all S with $|S| \leq n$. As such we have,

$$\Pi_{\mathcal{N}}(n) \leq \sup_{|S|=n} D_k \leq \prod_{i=1}^{k} \left(\frac{en}{p_i}\right)^{p_i} \leq \prod_{i=1}^{k} (en)^{p_i} = (en)^p \, .$$

To finish the calculation, it is possible to use a guess-and-check. In order to show that the VC dimension is at most d, it suffices to show that $\Pi_{\mathcal{N}}(d) < 2^d$. As such, suppose $n \geq cp\ln(p)$ for some $c \geq 0$. By the above calculation,

$$\ln(\Pi_{\mathcal{N}}(n)) \leq p\ln(ecp\ln(p)) = p\ln(p) + p(1 + \ln(c) + \ln(\ln(p))).$$

Since this is strictly less than $cp\ln(p)\ln(2)$ for sufficiently large c, it follows that the VC dimension is $O(p\ln(p))$.

15.3.2 Further VC dimension results (without proof)

The following table summarizes worst case VC dimension bounds given various conditions. There are two key points here: first, the linear threshold network really did not gain much power (whereas other networks do), and secondly the choice of σ is essential.

σ	Worst case VC dimension
$r \mapsto \mathbf{1}[r \geq 0]$	With one non-input node ($k = 1$), this is perceptron, thus VC is $\Theta(p)$. In general, the VC dimension is $\tilde{O}(p \ln(p))$; layers did not matter much in this example!
piecewise polynomial	$\Omega(pL)$ and $\tilde{O}(pL^2)$; this also holds in the piecewise affine case (in particular for the popular choice $\sigma(r) = \max\{0, r\}$).
convex for $x < 0$, concave for $x > 0$, and satisfying limit properties $\lim_{r\to\infty} \sigma(r) = 1$ and $\lim_{r\to-\infty} \sigma(r) = 0$	It's possible for VC dimension to be infinite! (Note however that other members of this class, for instance the sigmoid $\sigma(r) = 1/(1+\exp(-r))$, have VC dimension polynomial in p and L.)