1. Introduction and definitions

We build upon a paper by Emmanuel Abbé and Colin Sandon proving some limitations of deep learning at learning very simple functions like parity functions. More formally, we define $\mathcal{F} = \{p_s \mid s \subseteq [n]\}$ where $p_s : \{-1, +1\}^n \rightarrow \{-1, +1\}$ is defined by:

$$p_s(x) = \prod_{i \in s} x_i$$

Nature picks $S$ uniformly at random in $2^{[n]}$, and with access to $\mathcal{F}$ but not to $S$, the problem is to run a descent algorithm for a polynomial number of steps $t$ (in $n$) to obtain $w(t)$ (e.g., coordinate, gradient or stochastic gradient descent using labeled samples $(X_i, P_S(X_i))$ where the $X_i's$ are independently and uniformly drawn in $\{+1, 1\}^n$).

The goal is to have the neural network output a label $eval_{w(t)}(X)$ on a uniformly random input $X$ that (at least) correlates with the true label $p_S(X)$, such as $I(eval_{w(t)}(X); p_S(X)) = \Omega_n(1)$ for some notion of mutual information (e.g., TV, KL or Chi-squared mutual information), or

$$P(eval_{w(t)}(X) = p_S(X)) = 1/2 + \Omega_n(1)$$

if the output is made binary.

1.1. Definition. A neural net is a pair of a function $f : \mathbb{R} \rightarrow \mathbb{R}$ and a weighted directed graph $G$ with some special vertices and the following properties. First of all, $G$ does not contain any cycle. Secondly, there exists $n > 0$ such that $G$ has exactly $(n + 1)$ vertices that have no edges ending at them $v_0, v_1, ..., v_n$. We will refer to $n$ as the input size, $v_0$ as the constant vertex and $v_1, v_2, ..., v_n$ as the input vertices. Finally, there exists a vertex $v_{out}$ such that for any other vertex $v$, there is a path from $v$ to $v_{out}$ in $G$. We also use denote by $w(G)$ the weights on the edges of $G$.

1.2. Definition. We will say that a class of binary functions $\mathcal{F}$ together with a distribution $\mathcal{P}_X$ over $\mathcal{F}$ and a distribution $\mathcal{P}_X$ on the input is learnable if there exists a neural net of polynomial size and weights of polynomial (in $n$) value such that, after $T$ steps of gradient descent ($T$ is polynomial in $n$),

$$P(eval_{w(T)}(X) = p_S(X)) = 1/2 + \Omega_n(1)$$

As a shorthand, we will say that $(\mathcal{F}, \mathcal{P}_F, \mathcal{P}_X)$ is learnable.

It was shown in [1] that for $\mathcal{F} = \{p_s \mid s \subseteq [n]\}$, $\mathcal{P}_X$ uniform on $\{-1, +1\}^n$ and $\mathcal{P}_F$ uniform on $\mathcal{F}$, $(\mathcal{F}, \mathcal{P}_F, \mathcal{P}_X)$ is not learnable. As the parity functions form an orthonormal basis of the set of functions from $\{-1, +1\}^n$ to $\mathbb{R}$, it seems interesting to to study what other type of of subset $\mathcal{F}$ is learnable under uniform laws for $X$ and $F$. In the following, we define $\mathcal{M}_k = \{p_s(x) \mid s \subseteq [n], |s| = k\}$, that is the set of all monomials of degree $k$. We show the following:
Theorem 1.1. $M_k$ is learnable if and only if $k$ is constant. Moreover, if $k$ is constant, $M_k$ is learnable in only 1 step of gradient descent, with accuracy $1 - o(1)$.

Proof. If $k$ grows with $n$, one can simply compute the cross predictability of this set of functions and use as a blackbox the theorem from [1].

If $k$ is a constant, we need to build a neural net that learns correctly this set of function. Assume first that $k = 0 \pmod{4}$, then it is clear that for any $X \in \{-1,+1\}^n$ and any set $s \subset S$,

$$p_s(X) = \cos \left( \frac{\pi}{2} \sum_{i \in s} X_i \right)$$

Too see this write $||X||_1 = |\{i, X_i = +1\}|$ and $||X||_{-1} = |\{i, X_i = -1\}|$. Note that $||X||_1 + ||X||_{-1} = k$ is a multiple of 4 by assumption hence $||X||_1$ is even and $p_s(X) = 1$ as well.

We now have two cases, if $||X||_1 - ||X||_{-1}$ is a multiple of 4 then $\cos \left( \frac{\pi}{2} \sum_{i \in s} X_i \right) = \cos \left( \frac{\pi}{2} (||X||_1 - ||X||_{-1}) \right) = 1$ and we can see that $2||X||_{-1} = ||X||_1 + ||X||_{-1} - (||X||_1 - ||X||_{-1})$ is a multiple of 4 as well hence $||X||_{-1}$ is even and $p_s(X) = -1$ as well.

If $||X||_1 - ||X||_{-1}$ is not a multiple of 4 then we can see that $\cos \left( \frac{\pi}{2} \sum_{i \in s} X_i \right) = -1$ and that $||X||_{-1}$ is uneven and $p_s(X) = -1$ as well.

Now consider the following neural net:

![Neural Net Diagram](image)

The first layer is the input $X$. In the second layer we put one node for each set $s \subset S$ of size $k$. This node is linked to the input layer with an edge to every $X_i$ such that $i \in s$. Finally we put an edge between every node of the second layer to the output. We name the weights $r_{i,j}$ between first and second layer and $t_k$ between second and output layer.

We set initially $r_{i,j}^0 = 1$ for all $i,j$ and $t_k^0 = 0$ for all $k$. There is only a need to specify the loss function that we will choose as $L(x,y) = (x - y)^2$ and the learning rate that we will set to $\gamma = 1/2$.

We know that

$$r_{i,j}^1 = r_{i,j}^0 - 1/2 \frac{\partial}{\partial r_{i,j}} \mathbb{E}_X \left( \left( \sum_k t_k \cos \left( \frac{\pi}{2} \sum_i r_{i,k} X_i \right) - p_s(X) \right)^2 \right)$$

By linearity of $\mathbb{E}$ we can rewrite:
\[ r_{i,j}^1 = r_{i,j}^0 - 1/2E_X \left( \frac{\partial}{\partial r_{i,j}} \left( \sum_k t_k \cos \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) - p_s(X) \right)^2 \right) \]

and by simple computations:

\[ r_{i,j}^1 = r_{i,j}^0 - 1/2E_X \left( 2 \left( \sum_k t_k \cos \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) - p_s(X) \right) \cdot t_j \cdot \frac{\pi}{2} \cdot X_i \cdot \left( -\sin \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) \right) \right) \]

Now recall that we set \( r_{i,j}^0 = 1 \) hence \( \frac{\pi}{2} \sum_i r_{i,k}X_i = \frac{\pi}{2} \sum_i X_i = \frac{\pi}{2} (||X||_1 - ||X||_{-1}) = m\pi \) for some integer \( m \). Hence \( \sin \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) = 0 \) for any input hence

\[ r_{i,j}^1 = r_{i,j}^0 \]

(we could have simply used the fact that \( t_{j}^0 = 0 \) but this is a more general statement that might be useful later).

Similarly,

\[ t_k^1 = t_k^0 - 1/2E_X \left( \frac{\partial}{\partial t_k} \left( \sum_k t_k \cos \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) - p_s(X) \right)^2 \right) \]

is rewritten as:

\[ t_k^1 = t_k^0 - 1/2E_X \left( 2 \cdot \left( \sum_k t_k \cos \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) - p_s(X) \right) \cdot \cos \left( \frac{\pi}{2} \sum_i r_{i,k}X_i \right) \right) \]

and since \( t_k^0 = 0 \) for any \( k \) and \( r_{i,j}^0 = 1 \) for any \( i,j \) we get:

\[ t_k^1 = E_X (p_s(X) \cdot p_{sk}(X)) = \delta(s, sk) \]

where \( \delta(x, y) = 1 \) if \( x = y \).

Hence after 1 round we get that \( \text{eval}_{w^1}(X) = p_s(X) \) for any input \( X \). The neural net is indeed of polynomial size since there are \( O(n^k) \) subsets of cardinality \( k \). Note that the noise added at each step is not a problem since the derivative of \( \cos \) is bounded by 1. Finally, we deal similarly with cases where \( k = 1, 2 \) or 3 by replacing cosine by sine of adding a \(-\) sign.

Note that a similar result can be proven for a sum of a constant number of monomials.

1.3. A more general positive result. In [1], it is proven that if the cross predictability of a family of functions is exponentially small then it is impossible to learn this class of functions with a polynomial sized neural net and polynomially many learning steps. However no converse result is shown, we first prove with a simple argument that it is possible to learn with a polynomial sized neural net with accuracy \( \frac{1}{2} + \Omega(CP) \), assuming that all functions in the considered family are computable in polynomial time with a Turing machine.

We use the following a classic result as a black box. That is that if a function is computable by a Turing Machine in polynomial time then there is a circuit of polynomial size computing the same function (see for instance Introduction to the theory of computation by Sipser (2006)). Finally, before building our neural nets we define our own non linear functions with the following claim:
Claim 1.2. For any \( \epsilon > 0 \), there exists \( \sigma \in C^\infty(\mathbb{R}, \mathbb{R}) \) such that:

- \( \sigma(x) = 1 \) for any \( x \geq \epsilon \)
- \( \sigma(x) = -1 \) for any \( x \leq -\epsilon \)
- \( \sigma(0) = 0 \) and \( \sigma'(0) = 1 \)

Proof. Consider \( f_\alpha(x) = \frac{2}{\pi} \arctan \left( \exp(1/(x - \alpha)^2) - \exp(1/(x + \alpha)^2) \right) \) for some \( \alpha > 0 \). We define \( f_\alpha \) on the interval \((-\alpha, \alpha)\) and claim the following for any \( x \in (-\alpha, \alpha) \) and any \( k > 0 \):

\[
f_\alpha^{(k)}(x) = \frac{Q_k \left( \exp \left( \frac{1}{(x-\alpha)^2} \right), \exp \left( \frac{1}{(x+\alpha)^2} \right) \cdot \frac{1}{x-\alpha}, \frac{1}{x+\alpha} \right)}{1 + \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) - \exp \left( \frac{1}{(x+\alpha)^2} \right) \right)^2}
\]

where for any \( k > 0 \), \( Q_k(X, Y, Z, T) \) is a rational fraction and the degree of \( Q \) in the first and second variables \((X \text{ and } Y)\) is lower or equal to 1. We prove this claim by induction, first note that

\[
f_\alpha(x) = \frac{\exp \left( \frac{1}{(x-\alpha)^2} \right) \cdot \frac{-2}{(x-\alpha)^3} + \exp \left( \frac{1}{(x+\alpha)^2} \right) \cdot \frac{2}{(x+\alpha)^3}}{1 + \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) - \exp \left( \frac{1}{(x+\alpha)^2} \right) \right)^2}
\]

which is of the expected form. Now assume this is true for some \( k > 0 \) then:

\[
f_\alpha^{(k+1)}(x) = \frac{\partial}{\partial x} Q_k \left( \exp \left( \frac{1}{(x-\alpha)^2} \right), \exp \left( \frac{1}{(x+\alpha)^2} \right) \cdot \frac{1}{x-\alpha}, \frac{1}{x+\alpha} \right) \cdot \left( 1 + \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) - \exp \left( \frac{1}{(x+\alpha)^2} \right) \right)^2 \right)
\]

\[
\left( 1 + \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) - \exp \left( \frac{1}{(x+\alpha)^2} \right) \right)^2 \right)^2
\]

\[
- Q_k \left( \exp \left( \frac{1}{(x-\alpha)^2} \right), \exp \left( \frac{1}{(x+\alpha)^2} \right) \cdot \frac{1}{x-\alpha}, \frac{1}{x+\alpha} \right) \cdot \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) \cdot \frac{-2}{(x-\alpha)^3} + \exp \left( \frac{1}{(x+\alpha)^2} \right) \cdot \frac{2}{(x+\alpha)^3} \right)
\]

\[
\left( 1 + \left( \exp \left( \frac{1}{(x-\alpha)^2} \right) - \exp \left( \frac{1}{(x+\alpha)^2} \right) \right)^2 \right)^2
\]

Obviously the second term is of the wanted form and for the first one notice that:

\[
\frac{\partial}{\partial x} \left( \frac{ax + b}{cx^2 + dx + e} \right) = \frac{a(cx^2 + dx + e) - (ax + b)(2cx + d)}{(cx^2 + dx + e)^2}
\]

decreases the degree by at least 1. Hence, knowing that \( \frac{\partial}{\partial x} \exp \left( \frac{1}{(x-\alpha)^2} \right) = \exp \left( \frac{1}{(x-\alpha)^2} \right) \cdot \frac{-2}{(x-\alpha)^3} \) and similarly for \( \exp \left( \frac{1}{(x-\alpha)^2} \right) \) finishes the proof.

In particular we can deduce that for any \( k > 0 \),

\[
\lim_{x \to -\alpha^-} f_\alpha^{(k)}(x) = 0
\]

and

\[
\lim_{x \to -\alpha^+} f_\alpha^{(k)}(x) = 0
\]
Now consider $\alpha, \beta$ such that:

$$
\frac{\partial}{\partial x} f_\alpha(\beta x) = \beta f'_\alpha(0) = \frac{4\beta \exp(-1/\alpha^2)}{\alpha^3} = 1
$$

and

$$
\beta \epsilon = \alpha
$$

To prove a solution always exists simply replace $\beta = \alpha/\epsilon$ in the first equation and note that

$$
\lim_{x \to 0^+} \frac{\exp(1/\alpha^2)}{x^2} = +\infty \quad \text{and} \quad \lim_{x \to +\infty} \frac{\exp(1/\alpha^2)}{x^2} = 0.
$$

Finally consider the following function:

$$
\sigma_\epsilon(x) = \begin{cases} 
  f_\alpha(\beta x) & \text{if } -\epsilon < x < \epsilon \\
  -1 & \text{if } x \leq -\epsilon \\
  +1 & \text{otherwise}
\end{cases}
$$

First note that $\sigma(0) = f_\alpha(0) = 0$, $\sigma'(0) = \beta f'_\alpha(0) = 1$ (by choice of $\alpha, \beta$) and that

$$
\lim_{x \to -\epsilon^-} \frac{\partial}{\partial x} f_\alpha^{(k)}(\beta x) = \beta^k \lim_{x \to -\epsilon^-} \frac{\partial}{\partial x} f_\alpha^{(k)}(x) = 0 \quad \text{(by choice of } \alpha, \beta) \quad \text{and} \quad \lim_{x \to -\epsilon^+} \frac{\partial}{\partial x} f_\alpha^{(k)}(\beta x) = \beta^k \lim_{x \to -\epsilon^+} \frac{\partial}{\partial x} f_\alpha^{(k)}(x) = 0 \quad \text{(by choice of } \alpha, \beta).$$

Hence the two last conditions give that $\sigma_\epsilon$ is $C^\infty$. 

Now our circuit can be turned into a neural net that is robust to noise in a sense that with high probability the noise does not change the output at each node. For this we use the following implementation of logical gates:

**Claim 1.3.** Let $(x_i)_{i \leq n}$ be a family of boolean inputs ($1$ means true and $-1$ false) then:

$$
\text{OR}(x_1, \ldots, x_n) = \sigma_{1/2} \left( 2n - 1 + 2 \sum_{i=1}^{n} x_i \right)
$$

$$
\text{AND}(x_1, \ldots, x_n) = \sigma_{1/2} \left( 1 - 2n + 2 \sum_{i=1}^{n} x_i \right)
$$

$$
\text{NOT}(x_i) = \sigma_{1/2}(-x_i)
$$

Now we can prove the result with a very simple argument:

**Theorem 1.4.** If all functions in $\mathcal{F}$ are computable in polynomial time then $(\mathcal{F}, \mathcal{P}_\mathcal{F}, \mathcal{P}_X)$ is learnable with accuracy $\frac{1}{2} + \Omega(\text{Pred}(\mathcal{P}_\mathcal{F}, \mathcal{P}_X)) - 2^{-n/4}$.

**Proof.** Draw $\tilde{f}$ at random from $\mathcal{P}_\mathcal{F}$. Consider now the following neural net:
where the weight \( w \) is set as 0 initially and the box \( X \to \tilde{f}(X) \) is the circuit corresponding to \( \tilde{f} \) made in a robust to noise way (see claim 1.3). We note \( W = (w_1, w_2, \ldots, w_N, w) \) the weights of the neural net (where \( N \) is polynomial in \( n \)). We set the loss to be the quadratic loss (i.e. \( L(x, y) = (x - y)^2 \)) and the learning rate \( \gamma = 1/2 \). Finally we set, \( \alpha \) such that if \( z \sim \mathcal{N}(0, 2^{-n}) \) then \( \mathbb{P}(z \in (-\alpha, \alpha)) \leq 2^{-n} \). We claim the following:

**Claim 1.5.** For any \( 1 \leq i \leq N \),

\[
\frac{\partial}{\partial w_i} \mathbb{E}_X (L(\text{output}_{W(0)}(X), f(X)) = 0
\]

\[
\frac{\partial}{\partial w} \mathbb{E}_X (L(\text{output}_{W(0)}(X), f(X)) = -2\langle \tilde{f}, f \rangle
\]

**Proof.** First note that by linearity of expectation and derivative operator we get that:

\[
\frac{\partial}{\partial w_i} \mathbb{E}_X (L(\text{output}_{W(0)}(X), f(X)) = \mathbb{E}_X \frac{\partial}{\partial w_i} (L(\text{output}_{W(0)}(X), f(X))
\]

then

\[
\frac{\partial}{\partial w_i} (L(\text{output}_{W(0)}(X), f(X)) = 2(\text{output}_{W(0)}(X) - f(X)) \cdot \frac{\partial}{\partial w_i} \text{output}_{W(0)}(X)
\]

As we set \( w^{(0)} = 0 \) initially we get that \( \frac{\partial}{\partial w_i} \text{output}_{W(0)}(X) = w^{(0)} \sigma'_\alpha(0) \frac{\partial}{\partial w_i} \sigma_{1/2}(\tilde{f}(X)) = 0 \) which proves the first part of the claim.

Second,

\[
\frac{\partial}{\partial w} (L(\text{output}_{W(0)}(X), f(X)) = 2(\text{output}_{W(0)}(X) - f(X)) \cdot \frac{\partial}{\partial w} \text{output}_{W(0)}(X)
\]

which can be continued as:

\[
\frac{\partial}{\partial w} \text{output}_{W(0)}(X) = \sigma'_\alpha(0) \sigma_{1/2}(\tilde{f}(X)) = \tilde{f}(X)
\]

which ends the second part of the claim as \( \sigma_{1/2}(0) = 0 \) and since \( w^{(0)} = 0 \) hence \( \text{output}_{W(0)}(X) = 0 \) for any input \( X \).

The previous claim immediately implies that:

**Claim 1.6.**

\[
w^{(1)} = w^{(0)} + \langle f, \tilde{f} \rangle + z = \langle f, \tilde{f} \rangle + z
\]

and for any \( 1 \leq i \leq N \),

\[
w^{(1)}_i = w^{(0)}_i + z_i
\]

where \( z, z_1, \ldots, z_N \) are iid random variables and \( z \sim \mathcal{N}(0, 2^{-n}) \).

**Proof.** Apply the noisy SGD algorithm with \( \gamma = 1/2 \) and claim 1.5.

Before finishing the proof, we need one last claim:

**Claim 1.7.** Let \( f_1, f_2 \) be two boolean functions, then:

\[
\mathbb{P}_X(f_1(X) = f_2(X)) = \frac{1}{2} + \frac{\langle f_1, f_2 \rangle}{2}
\]
Proof. Write
\[
P_X (X \in \{x, f_1(x) = f_2(x)\}) + P_X (X \in \{x, f_1(x) \neq f_2(x)\}) = 1
\]
and
\[
P_X (X \in \{x, f_1(x) = f_2(x)\}) - P_X (X \in \{x, f_1(x) \neq f_2(x)\}) = \langle f_1, f_2 \rangle
\]
\[
\square
\]
Hence, using claim 1.3, we get that for any \( f \):
\[
P_X (output_{W(1)}(X) = f(X)) = \frac{1}{2} + \frac{|\langle \hat{f}, f \rangle|}{2}
\]
if \( z \geq \alpha - \langle \hat{f}, f \rangle \) (assume \( \langle \hat{f}, f \rangle \geq \alpha \) without loss of generality).
\[
P_X (output_{W(1)}(X) = f(X)) = \frac{1}{2} - \frac{|\langle \hat{f}, f \rangle|}{2}
\]
if \( z \leq -\langle \hat{f}, f \rangle + 2\alpha \).

Recall that we chose \( \alpha \) such that the probability of \( z \) falling into an interval of length \( 2\alpha \) is extremely low so this case essentially never exists. Now we can see that the second case (the bad case) can happen with good probability only if \( |\langle \hat{f}, f \rangle| \) is small. More precisely, using theorem 7.1, we get that the bad case happens with probability at most:
\[
\exp \left( -2^n \cdot |\langle \hat{f}, f \rangle|^2 / 2 \right) / 2^{n/2} \cdot |\langle f, f \rangle|
\]
so if \( |\langle \hat{f}, f \rangle| \geq 2^{-n/4} \) this happens with probability \( o(2^{-n}) \). Hence we can lower bound the probability of success by:
\[
P_{X,Z} (output_{W(1)}(X) = f(X)) \geq \left( \frac{1}{2} + \frac{|\langle \hat{f}, f \rangle|}{2} - 2^{-n/4} \right) (1 - o(2^{-n}))
\]
where now the probability is also taken over the noise \( Z \).

So finally,
\[
P_{F,X,Z} (output(X) = F(X)) \geq E_F \left( \left( \frac{1}{2} + \frac{|\langle \hat{f}, f \rangle|}{2} - 2^{-n/4} \right) (1 - o(2^{-n})) \right)
\]
Now since \( \hat{f} \) is drawn independently from \( P_F \) we get that (since \( |\langle f, \hat{f} \rangle| \geq |\langle f, f \rangle| \)),
\[
E_F (P_{F,X,Z} (output(X) = F(X))) \geq \left( \frac{1}{2} + \frac{\text{Pred}(P_F, P_X)}{2} \right) (1 - o(2^{-n})) - 2^{-n/4}
\]
and by a probabilistic argument, there must be an \( f_0 \) such that
\[
P_{F,X,Z} (output(X) = F(X)) \geq \left( \frac{1}{2} + \frac{\text{Pred}(P_F, P_X)}{2} \right) (1 - o(2^{-n})) - 2^{-n/4}
\]
Finally, assuming that \( \text{Pred}(P_F, P_X) = \Omega(2^{-n/4}) \) finishes the proof.
\[
\square
\]
However we saw that some class of functions as parity of bounded size sets are learnable with accuracy $\frac{1}{2} + \Omega(1)$ even though the cross predictability is polynomially small. This raises the question of a characterization of what type of functions can be learnt with a constant accuracy.

To this end, we define the following quantity, for a vector space of boolean functions $E$:

$$c(\mathcal{F}, E) = \mathbb{E}_F (||\pi_E(F)||^2)$$

where $\pi_E(F)$ is the orthogonal projection of $F$ on subspace $E$ and $|| \cdot ||$ is the norm associated to scalar product.

Following this definition we need the following claim:

**Claim 1.8.** Let $E$ be a vectorial subspace, $f$ a boolean function, then if we write $f_E = \pi_E(f)$ and $f_\perp = f - f_E$ then if $||f_E||^2 = \mathbb{E}_X(f_E^2(X)) \geq \beta$ then:

$$\mathbb{P}_X (|f(X) - f_E(X)| > \alpha) \leq \frac{1 - \beta}{\alpha^2}$$

**Proof.** By Markov’s inequality we get that

$$\mathbb{P}_X (|f(X) - f_E(X)| > \alpha) = \mathbb{P}_X (|f_\perp(X)| > \alpha) \leq \frac{||f_\perp||^2}{\alpha^2}$$

Now by Pythagore's theorem we have that $1 = ||f||^2 = ||f_\perp||^2 + ||f_E||^2$ hence $||f_\perp||^2 \leq 1 - \beta$ and the result follows. □

Next we give the following sufficient condition for learnability:

**Theorem 1.9.** Assume there exists a vector space of boolean functions $E$ of polynomial dimension such that $c(\mathcal{F}, E) = \Omega(1)$ and such that one can build an orthogonal basis of $E$ of computable functions, then $\mathcal{F}$ is learnable with constant accuracy.

**Proof.** Let $f_1, \ldots, f_l$ the orthonormal basis of computable functions. Consider the following neural net:

$$\begin{align*}
\mathbb{E}_F (||\pi_E(F)||^2) & = \mathbb{E}_F (||f||^2) - \mathbb{E}_F (||f_\perp||^2) \\
& \geq \mathbb{E}_F (||f||^2) - \beta.
\end{align*}$$

The result follows.
with the quadratic loss function and weights $w_1, \ldots, w_n$ set to 0 initially. After one learning steps, these weights become

$$w_i^1 = \langle f, f_i \rangle$$

for any $f$ with similar calculations as the previous theorem. Hence this neural net computes in one step $\pi_E(f)$ for any $f$ with some failure probability. Given our choice of $\alpha$, with arbitrarily high probability, the output of this neural net will be either $\pi_E(f)(X)$ or $-\pi_E(f)(X)$ since after one step of gradient descent we get that $\sum_i w_i \sigma_{1/2}(f_i(X)) = f_E(X)$. The only thing left to deal with is the noise that can turn this into $-f_E(X)$ but if $f_E(X) > \frac{1}{n}$ for instance, since the noise is exponentially small with high probability this stays $f_E(X)$. Using claim 1.8 finishes the proof. □

Questions/remarks: can we remove assumption on computability of the basis of the subspace and replace it by computability of $F$? If vectors can be expressed with polynomially many non-zero coefficients in the parity basis then it is computable however the converse is false probably. What if we remove assumption on orthogonality of the basis? Probably several gradient steps are enough in that case, computations to be made.

We give a partial answer to the the second question. The previous theorem extends to the case where pairwise dot products are relatively small instead of perfect orthogonality.

**Theorem 1.10.** Assume there exists a vector space of boolean functions $E$ of polynomial dimension such that $c(F, E) = \Omega(1)$ and such that one can build a basis $(f_1, \ldots, f_d)$ of $E$ of computable functions such that $\max_{i \neq j} |\langle f_i, f_j \rangle| < \frac{1}{d}$, then $F$ is learnable with constant accuracy.

**Proof.** Use the same construction as before. With straightforward calculations one can see that the recursion relation for weights of the last layer is given by (assuming a quadratic loss and learning rate $1/2$ as before):

$$w_i^{t+1} = w_i^t + \langle f, f_i \rangle - \langle \text{output}_{w^t}, f \rangle$$

which can be rewritten as:

$$w_i^{t+1} = w_i^t + \langle f, f_i \rangle - \sum_{j=1}^d w_j^t \langle f_i, f_j \rangle$$

which can be rewritten as:

$$W^{t+1} = (I_d - G)W^t + U$$

where $W^t = (w_1^t, \ldots, w_k^t)^T$ is the vector of current weights, $U = ((f, f_1), \ldots, (f, f_d))^T$, $I_d$ the identity matrix and $G = ((f_i, f_j))_{i,j}$ the Gram matrix of the family of $f_i$s. The Gram matrix is symmetric and its rank is the same of the rank of the family $(f_i)_i$; hence in this case it is invertible. Finally it is definite positive.

We can rewrite this equation using $\tilde{W}^t = W^t - G^{-1}U$ as

$$\tilde{W}^{t+1} + G^{-1}U = (I_d - G)(\tilde{W}^t + G^{-1}U) + U$$

which is equivalent to:

$$\tilde{W}^{t+1} = (I_d - G)\tilde{W}^t$$
hence we know that if all the eigenvalues of \((I_d - G)\) are less than 1 in absolute values then 
\[
\lim_{t \to +\infty} ||\tilde{W}^t|| = 0.
\]

To finish the proof we claim that this is the case if \(\max_{i \neq j} |\langle f_i, f_j \rangle| < \frac{1}{d-1}\). Indeed assume there exists \(X\) such that \(GX = (1 - \lambda)X\) then for every \(1 \leq i \leq d\):

\[
(1 - \lambda)X[i] = \sum_{j=1}^{d} G_{ik}X[k] \iff -\lambda X[i] = \sum_{j \neq i} G_{ik}X[k] \implies |\lambda X[i]| < \sum_{j \neq i} \frac{1}{d-1}|X[j]|
\]

Summing all these inequalities we get:

\[
|\lambda| \cdot ||X||_1 < ||X||_1
\]

which yields the result. Hence \(W^t\) converges to \(G^{-1}U\) exponentially fast. What remains to show is that \(G^{-1}U\) is indeed the correct vector to represent the projection on the subspace. Write \(f = \sum_i \alpha_i f_i\) and \(A = (\alpha_1, \ldots, \alpha_d)^T\). We want to prove that:

\[
G^{-1}U = A \iff GA = U
\]

Indeed, for every \(i \leq d\),

\[
(GA)_i = \sum_j (f_i, f_j) \alpha_j = (f_i, \sum_j \alpha_j f_j) = \langle f_i, f \rangle = U_i
\]

\(\square\)
2. A Counterpart Result to Theorem 1.9

In this section we show a kind of counterpart to theorem 1.9. In this section, we define for $S$ a subset of function $\mu(S)$ the measure of the set $S$. Let us write

$$C(n) = \sup_{E, \dim(E) \leq n} \mathbb{E}_F(||\pi_E(F)||^2)$$

and we assume that for any polynomial $P$,

$$\lim_{n \to +\infty} C(P(n)) = 0$$

We are now ready to claim the following:

**Claim 2.1.** Let $P$ be a polynomial and assume $\lim_{n \to +\infty} C(P(n)) = 0$, then for any $\epsilon > 0$, there exists $n_0 > 0$ such that for any $n \geq n_0$ there exists a set $S \subset \mathcal{F}$ of functions such that:

$$|S| = P(n)$$

and

$$\max_{i \neq j} \langle f_i, f_j \rangle^2 \leq \epsilon$$

**Proof.** Let $\epsilon > 0$. By definition there exists $n_0$ such that for all $n \geq n_0$, $C(P(n)) \leq \epsilon$. We build the set $S$ by induction. Set $S_1 = \{f\}$ for an arbitrary $f \in \mathcal{F}$, it satisfies the property $\max_{f \neq f'} \langle f, f' \rangle^2 \leq \epsilon$.

Now assume that $S_i$ is built satisfying the property and such that $|S_i| < P(n)$. Then, setting $E = \text{Vect}(\{f \mid f \in S_i\})$, we know that

$$\mathbb{E}_F(||\pi_E(F)||^2) \leq \epsilon$$

rewriting this with conditional expectation we get:

$$\mathbb{E}_F(||\pi_E(F)||^2) = \mu(S_i) + \mu(\mathcal{F} - S_i) \cdot \mathbb{E}_F(||\pi_E(F)||^2 \mid F \notin S_i)$$

hence

$$\mathbb{E}_F(||\pi_E(F)||^2 \mid F \notin S_i) \leq \frac{\epsilon - \mu(S_i)}{1 - \mu(S_i)} \leq \epsilon$$

therefore there must exist one $f \notin S$ such that $||\pi_E(f)||^2 \leq \epsilon$. Set $S_{i+1} = S_i \cup \{f\}$ and check that for any $f' \in S_i$, $\langle f, f' \rangle^2 = \langle \pi_E(f), f' \rangle^2 \leq ||\pi_E(f)||^2 \cdot ||f'||^2 \leq \epsilon$ hence the property is still satisfied. \qed

We now make the following simple claim that if an algorithm learns well on average a set of function then there must be a subset of these functions of good measure such that the algorithm learns correctly all the functions in the subset:

**Claim 2.2.** Let $A$ be a (randomized) algorithm and $(\mathcal{F}, \mathcal{P}_F, \mathcal{P}_X)$ such that:

$$\mathbb{P}_{F,X,R}(A(x) = F(x)) \geq \alpha > 0$$

Here the randomness is taken over $F, X$ and $R$ the random bits used by the algorithm. Then if we call the set $B$ the set of functions badly learnt by the algorithm, more precisely:

$$B = \{f \in \mathcal{F}, \mathbb{P}_{X,R}(A(x) = f(x)) \leq \frac{\alpha}{2}\}$$

then

$$\mu(B) \leq \frac{1 - \alpha}{1 - \alpha/2}$$
Proof. Assume this is not the case, then
\[
\mathbb{P}_F(\mathcal{X}, \mathcal{R})(A(x) = F(x)) < \mu(B) \frac{\alpha}{2} + (1 - \mu(B)) < \frac{\alpha/2 - \alpha^2/2}{1 - \alpha/2} + \frac{\alpha/2}{1 - \alpha/2} = \frac{\alpha(1 - \alpha/2)}{1 - \alpha/2} = \alpha
\]
which contradicts our hypothesis. \qed

Now if an algorithm learns efficiently a class of functions satisfying [1] the previous two claims show that we can extract from \( \mathcal{F} \) a family of \( P(n) \) functions with low pairwise dot product such that the algorithm learns efficiently all of them:

**Claim 2.3.** Let \( \mathcal{A} \) be a (randomized) algorithm and \(( \mathcal{F}, \mathcal{P}_F, \mathcal{P}_X )\) such that:

\[
\mathbb{P}_F(\mathcal{X}, \mathcal{R})(A(x) = F(x)) \geq \alpha > 0
\]

Assume also that \(( \mathcal{F}, \mathcal{P}_F, \mathcal{P}_X )\) satisfies [1] then for any \( \epsilon > 0 \) there exists \( n_0 > 0 \) such that for any \( n \geq n_0 \) there exists a family \( S \subset \mathcal{F}_n \) of \( P(n) \) functions such that:

(2) \[ \text{For all } f \in S, \mathbb{P}_X(\mathcal{R})(A(x) = f(x)) \geq \frac{\alpha}{2} \]

(3) \[ \max_{f \neq f', f, f' \in S} \langle f, f' \rangle^2 \leq \epsilon \]

Proof. \qed
3. Bias and Learning Complexity

Here we assumed that \( P_{F,X}(F(X) = 1) = \frac{1}{2} \). Otherwise one could argue that without information about the drawn function then one cannot do better that the Bayes optimal predictor. If we write \( \rho = P_{F,X}(F(X) = 1) \) one can redefine the cross predictability of \([\Pi]\) as:

\[
\text{Pred}(\mathcal{P}_F, \mathcal{P}_X) = \mathbb{E}_{F,F'} \left( \mathbb{E}_X \left( -1 + Y \left( (Y + 1) \frac{1 - \rho}{\rho} + (Y - 1) \frac{\rho}{1 - \rho} \right) \right)^2 \right)
\]

with \( Y = \frac{F(X) + F'(X)}{2} \). One can check that setting \( \rho = 1/2 \) is consistent with the previous definition. If this quantity is small one cannot do better than Bayes optimal predictor.

4. Removing noise in gradient descent

5. Learning connectivity

One important question would be to use these results on cross predictability to prove that connectivity is not learnable. However it seems to be difficult to compute cross predictability on class of connectivity functions. Two ideas were:

- Set \( F = \{ c_s(X), s \subset \{1, \ldots, n\}, |s| = n/2 \} \) with \( s \) picked uniformly at random, and \( c_s(X) = 1 \) iff the graph induced by \( s \) is connected.
- Set \( F = \{ c_s(X), s \subset \{1, \ldots, n\} \} \) with \( c_s(X) = 1 \) iff there is a path from every vertex to a node in \( s \).

However it seems not easy to compute the cross predictability. For instance in the first instance we might be tempted to set the input \( X \) to follow an Erdos-Renyi distribution such that the probability of every induced subgraph is connected with probability \( 1/2 \). To compute the CP, we need to consider pairs \( s, s' \subset \{1, \ldots, n\} \) such that \( |s \cap s'| \) is small. In that case each one behaves like an erdos renyi where one node is "cheating" by multiplying its probability of appearance of incident edges. The case \( |s \cap s'| \) big is dealt with by saying that there are few pairs such that this is big. However, computing the CP remains challenging.

6. Statistical Query Learning

There appears to be a link between these neural net results and the classical statistical query model for learning.

References


7. Appendix

**Theorem 7.1.** Let \( z \sim \mathcal{N}(0, \sigma^2) \), then for any \( a > 0 \),

\[
P(z > a) \leq \frac{\exp \left(-\sigma^{-2} \cdot a^2/2\right)}{\sigma^{-1} \cdot a \cdot \sqrt{2\pi}}
\]

**Proof.** We know that \( \sigma^{-1} \cdot z \sim \mathcal{N}(0, 1) \) and that \( \int_a^\infty \frac{e^{-t^2/2}}{\sqrt{2\pi}} \leq \frac{e^{-a^2/2}}{a \sqrt{2\pi}} \) hence:

\[
P(z > a) = \mathbb{P}(\sigma^{-1} \cdot z > \sigma^{-1} \cdot a) \leq \frac{\exp \left(-\sigma^{-2} \cdot a^2/2\right)}{\sigma^{-1} \cdot a \cdot \sqrt{2\pi}}
\]

\[\square\]
School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, Switzerland

Email address: etienne.bamas@epfl.ch