Statistical Review on Neural Network Models

Min Hun Lee
mhlee@cmu.edu

1 Introduction

Function approximation is to find the underlying relationship between finite input and output of data. Learning such a mapping between an input and output is the fundamental problem in diverse real world applications, such as prediction, pattern recognition, and classification. Various methods have been developed to address this problem, in which one of them is to use a neural network model. A neural network model has great expressive power [9]. Recently, it has become increasing popular technique for machine learning tasks (e.g. image classification, speech recognition) [11, 7]. However, a deep neural network works mostly as a black-box solution. The theoretical understanding on how it works remain limited.

This paper aims to survey and review some existing works that analyze a shallow neural network in a statistical perspective. Given a neural network with significant representation capability, this work will elaborate the constraints for the success of a neural network. Based on the results of [3, 4], this paper will describe the estimation error between the target function and the estimated network. Furthermore, this paper will review the results of [5, 9] to describe the ability of feedfoward neural networks to learn and achieve a universal approximation. This implies that there exists an estimated neural network that approximates any measurable function. In addition, this paper describe a convergence analysis for Stochastic Gradient Descent on a neural network with identity mapping structure, rich subset of two-layer feedforward network with ReLU activations [10]. Overall, this paper will include only limited statistical analysis on shallow neural networks (i.e. neural network models with one or two hidden layers). More a rigorous statistical analysis of neural networks with multiple hidden layers is an open and important problem. Still, a brief statistical review of this paper might provide another perspectives for researchers to understand the properties of neural networks.

2 Notation and Assumption

2.1 Single Layer Neural Network Models

Let assume that we have n pairs of input output data in the form of (x_i, y_i) for $i = 1, ..., N$. Let $x_i \in R^d$ and $y_i \in R$ for $i = 1, ..., N$. We assume that both input and output are bounded. We seek to find an unknown function $f(x) : R^d \to R$. In addition, this paper assumes that (x_i, y_i) for $i = 1, ..., N$ are independently drawn from a distribution $P_{x,y}$. It also assumes that the response variable is subject to an error that is $y_i = f(x_i) + e_i$, where $E(e_i|x_i) = 0$ that the errors, e_i are bounded and independent from the inputs x_i for $i = 1, ..., N$.

Functions f(x) with bounded domain in R^d can be approximated using feed-forward neural network models. The network models with one layer of sigmoid nonlinear activation functions is defined as follows:

$$f_h(x) = f_h(x, \theta) = \sum_{k=1}^{h} c_k \phi(a_k^T x + b_k) + c_0 \tag{1}$$

The approximation function, f_h is parameterized by the vector θ that consists of weight vector $a_k \in R^d$ and bias terms $b_k, c_k \in R$ for $k = 1, ..., h$ and $c_0 \in R$. h describes the number of nonlinear terms, which are also known as nodes or hidden units. $h \geq 1$.

10702 Class Project (2018).

ϕ is an activation function. The function $x \to \phi(a_k^T x + b_k)$ describes the k-th hidden units. In this paper, we assume the function $\phi(z)$ to be a sigmoid function. The function is assumed to be a sigmoid function if it is a bounded function that satisfies following conditions: $\phi(z) \to 1$ as $z \to \infty$ and $\phi(z) \to -1$ as $z \to -\infty$, where z = $a_k^T x + b_k$.

Let define a Fourier representation of the form $f(x) = \int_{R_d} e^{iw^T x}\tilde{f}(w)dw$. We define $C_f = \int |w|_1 F(dw)$, where $F = |\tilde{F}|$ is the Fourier magnitude of distribution of f and $|w|_1 = \sum_{j=1}^{d} |w_j|$, the l_1 norm of w in R^d. C_f is finite.

2.2 Empirical Risks and Complexity of Estimator

The empirical risk of an estimated network $\hat{f}_{h,N}$ is defined as $\frac{1}{N}\sum_{i=1}^{N}(y_i - f_h(x_i, \theta))^2$, where the square loss is applied given N data samples. For each number of nodes h and sample size N, let define $\Omega_h = \Omega_{h,N}$ be a discrete set of parameter vectors θ and $L_{h,N}(\theta)$ be nonnegative numbers satisfying $L_{h,N}(\theta) \geq l$ for some constant $l > 0$, and $\sum_{\theta \in \Omega_h} e^{-L_{n,N}(\theta)} \leq 1$.

Let define the index of resolvability given h and N as follows:

$$R_{h,N}(f) = \min_{\theta \in \Omega_h}(\left\|f - \bar{f}_n(\grave{},\theta)\right\|^2 + \lambda\frac{L_{h,N}(\theta)}{N}) \tag{2}$$

where λ is a given positive constant from Theorem 3.2. This equation gives the resolvability for a neural network family with a given number of nodes h. For the collection of networks, the index of resolvability is as follows:

$$R_N(f) = \min_{n \geq 1}(R_{h,N}(f) + \lambda\frac{L(h)}{N}) \tag{3}$$

This minimization determines the h that leads to the best resolvability.
The minimum complexity estimator of neural network with h hidden nodes is as follows:

$$\hat{f}_{h,N}(x) = \bar{f}_h(x, \theta_{\hat{h,N}})$$

where

$$\theta_{\hat{h,N}} = arg\min_{\theta \in \Omega_h}(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{f}_h(x_i, \theta))^2 + \lambda\frac{L_{h,N}(\theta)}{N})) \tag{4}$$

Thus, $\hat{f}_{h,N}$ denotes the least square estimator with a complexity penalty. The minimum complexity estimator with both h and θ, $\hat{f_N} = \hat{f}_{\hat{n},N}$ can be found similarly.

2.3 Function Spaces

2.3.1 Continuous functions, Borel measurable functions

Let C^r be the set of continuous function from R^r to R, where $r \in N$ and let M^r be the set of all Borel measurable functions from R^r to R. We denote the Borel σ-field as B^r.

For any Borel measurable ϕ, the class of functions in the form of the Equation (1) belong to M^r. If ϕ is continuous, the class of functions in the form of the Equation (1) belong to C^r.

Closeness of functions f and g belonging to C^r or M^r is measured by a metric, ρ. Closeness of one class of functions to another class is described by the concept of denseness.

2.3.2 Denseness

A subset S of a metric space (X, ρ) is ρ - dense in a subset T if for every $\epsilon > 0$ and for every $t \in T$, there is an $s \in S$ such that $\rho(s,t) < \epsilon$. A subset S of C^r is said to be uniformly dense on compacta in C^r if for every compact subset $K \subset R^r$. S is ρ_K-dense in C^r, where $f, g \in C^r$ $\rho_K(f,g) \equiv sup_{x \in K}|f(x) - g(x)|$. A sequence of functions f_n converges to a function f uniformly on compacta if for all compact $K \subset R^r$ $\rho_k(f_n, f) \to 0$ as $n \to \infty$.

An element of S can approximate an element of T to any desired degree of accuracy. In the theorems of this paper, T and X correspond to C^r or M^r, S corresponds to a neural network with the form of the Equation (1) for specific choice of ϕ, and ρ is chosen appropriately.

2.3.3 Probability Measure

Let μ be a probability measure on (R^r, B^r). If f and g belong to M^r, we say that they are μ-equivalent if $\mu\{x \in R^r : f(x) = g(x)\} = 1$.

The measure μ describes the relative frequency of occurrence of input patterns x. Taking μ to be a probability measure (i.e. $\mu(R^r) = 1$) is a matter of convenience. The Theorems in the results hold for arbitrary finite measures.

Given a probability measure μ on (R^r, B^r), we define the metric ρ_μ from $M^r \times M^r$ to R^+ by $\rho_\mu(f, g) = inf\{\epsilon > 0 : \mu\{x : |f(x) - g(x)| > \epsilon\} < \epsilon\}$. Two functions are close in this metric if and only if there is only a small probability that they differ significantly.

2.3.4 Two-layer Neural Networks with ReLU Activation

This paper utilizes a subset of two-layer feed-forward neural networks with ReLU activation functions to analyze convergence for SGD. This subset is characterized by a special structure called identity mapping. Specifically, we consider the following function:

$$f(x, W) = \left\| ReLU((I + W)^T x) \right\|_1 \tag{5}$$

where ReLU(x) = max(v, 0) describes the ReLU activation function. $x \in R^d$ is the input vector sampled from a Gaussian distribution, N(0, I), $W \in R^{d \times d}$ is the weight matrix (i.e. $(w_1, ..., w_n)$), where d is the number of input units. Note that I adds e_i to column i of W, which makes f asymmetric. We get different functions by switching any two columns in W.

We assume that there exists a two-layer teacher network with weight $W^* = (w_1^*, ..., w_n^*)$ following the standard setting in [13, 15]. We train the student network using l_2 loss as follows:

$$L(W) = E_x[(f(x, W) - f(x, W*))^2] \tag{6}$$

,where f(x, W*) and f(x, W) represent the teacher and student network respectively.

Given the loss function, Equation. (6), we take derivative with respect to w_j, we get the gradient as follows:

$$\nabla L(W)_j = 2E_x[(\sum_i ReLU(< e_i + w_i, x >) - \sum_i ReLU(< e_i + w_i^*, x >))x\mathbb{1}_{<e_j+w_j,x>\geq 0}] \tag{7}$$

where $\mathbb{1}_e$ is the indicator function that equal 1 if the event e is true and 0 otherwise.

Denote $\theta_{i,j}$ as the angle between $e_i + w_i$ and $e_i + w_i$ and $\theta_{i^*,j}$ as the angle between $e_i + w_i^*$ and $e_i + w_i$. Denote $\bar{v} = \frac{v}{\|v\|_2}$. Denote $\overline{I + W^*}$ and $\overline{I + W}$ as the column-normalized version of $I + W^*$ and $I + W$ such that every column has unit norm. As the input is from a normal distribution, we can compute the expectation inside the gradient [15] as follows:

If $x \sim N(0, I)$, then $-\nabla L(W)_j = \sum_{i=1}^d (\frac{\pi}{2}(w_i^* - w_i) + (\frac{\pi}{2} - \theta_{i^*,j})(e_i + w_i^*) - (\frac{\pi}{2} - \theta_{i^*,j})(e_i + w_i) + (\|e_i + w_i^*\|_2 \sin\theta_{i^*,j} - \|e_i + w_i\|_2 \sin\theta_{i,j})\overline{e_j + w_j})$

If we assume input x is from the Gaussian distribution, even if the gradient of ReLU is not well defined at the point of zero, the loss function becomes smooth, and the gradient is well defined everywhere.

Denote $u \in R^d$ as the all one vector. Denote Diag(W) as the diagonal matrix of matrix W, Diag(v) as a diagonal matrix whose main diagonal equals to the vector v. Denote Off-Diag$(W) \equiv W-$Diag(W). Denote $[d]$ as the set $\{1, ..., d\}$. We use the notation of inner product between matrices $W, W^*, \nabla L(W)$, such that $< \nabla L(W), W >$ means the summation of the entry-wise products. $\|W\|_2$ is the spectral norm of W, and $\|W\|_F$ is the Forbenius norm of W.

We define the potential function $g \equiv \sum_{i=1}^d (\|e_i + w_i^*\|_2 - \|e_i + w_i\|_2)$ and variable $g_j \equiv \sum_{i \neq j} (\|e_i + w_i^*\|_2 - \|e_i + w_i\|_2)$

We also define additional variables, A_j and A as follows: $A_j \equiv \sum_{i \neq j}((e_i + w_i^*)\overline{e_i + w_i^*}^T - (e_i + w_i)\overline{e_i + w_i}^T)$ and $A \equiv \sum_{i=1}^d ((e_i + w_i^*)\overline{e_i + w_i^*}^T - (e_i + w_i)\overline{e_i + w_i}^T) = (I + W^*)\overline{I + W^*}^T - (I + W)\overline{I + W}^T$.

This paper considers the standard SGD with mini-batch method for training a neural network. Assume W_0 is the initial point, and in step $t > 0$, we have the following updaing rule:

$$W_{t+1} = W_t - \eta_t G_t$$

where the stochastic gradient $G_t = \nabla L(W_t) + E_t$ with $E[E_t] = 0$ and $\|E_t\|_F \leq \epsilon$. Let $G_2 \equiv 6d\gamma + \epsilon, G_F \equiv 6d^{1.5}\gamma + \epsilon$, which are the upper bound of $\|G_t\|_2$ and $\|G_t\|_F$ respectively. γ is the upper bound of $\|W^*\|_2$ and $\|W_0\|_2$.

L is not convex, so we need a weaker condition to get convergence guarantees, which is called one-point strongly convexity. A function $f(x)$ is δ-one point strongly convex in domain D with respect to point x^*, if $\forall x \in D, < -\nabla f(x), x^* - x >> \delta \|x^* - x\|_2^2$ By the definition, if a function is strongly convex, it is also one-point strongly convex in the entire space with respect to the global minimum. As long as the step size is small enough, we will finally arrive the optimal point by a winding path.

For function $f(W)$, consider the SGD update $W_{t+1} = W_t - \eta G_t$, where $E[G_t] = \nabla f(W_t), E[\|G_t\|_F^2] \leq G^2$. Suppose for all t, W_t is always inside the δ-one point strongly convex region with diameter D (i.e. $\|W_t - W^*\|_F \leq D$). Then for any $\alpha > 0$ and any T such that $T^\alpha logT \geq \frac{D^2\delta^2}{(1+\alpha)G^2}$, if $\eta = \frac{(1+\alpha)logT}{\delta T}$, we have $E\|W_T - W^*\|_F^2 \leq \frac{(1+\alpha)logTG^2}{\delta^2 T}$.

3 Key Results

3.1 Approximation Error Bounds for Neural Networks

Theorem 3.1 *Given an arbitrary sigmoid function ϕ, a target function f with finite C_f, and a probability measure μ on a domain in $[-1,1]^d$, there exists a neural network of the form Equation (1), such that*

$$\|f - f_h\| \leq \delta = \frac{C_f}{\sqrt{h}} \tag{8}$$

The parameters of Equation (1) may be restricted to satisfy $\sum_{k=1}^{h} |c_k| \leq C, |c_0 - f(0)| \leq C$ and $|b_k| \leq |a_k|_1$ for functions f with $C_f \leq C$.

Theorem 3.2 *Let a neural network be estimated by least square with a complexity penalty as in Equation (4), where the range of y and each candidate function is restricted to a know interval of length b, then for $\lambda > \frac{5b^2}{3}$, for all $h \geq 1$, and all $N \geq 1$.*
$E\left\|f - \hat{f}_{h,N}\right\|^2 \leq \gamma R_{h,N}(f) + \frac{2\gamma\lambda}{N}$ and $E\left\|f - \hat{f}_N\right\|^2 \leq \gamma R_{h,N}(f) + \frac{2\gamma\lambda}{N}$, where $\gamma = \frac{(3\lambda+b^2)}{(3\lambda-5b^2)}$. Thus,

$$E\left\|f - \hat{f}_N\right\|^2 \leq O(R_N(f)) \tag{9}$$

The index of resolvability captures the effect of the approximation error $\|f - f_h\|^2$ and the estimation error $E\left\|f_h - \hat{f}_{h,N}\right\|^2$.

Theorem 3.3 *Let $\hat{f}_{h,N,C}(x)$ be the least square neural network estimator that minimizes the mean square error estimator subject to the constraints of parameter vector, satisfying that $\|f - f_h\| \leq \frac{C_f}{\sqrt{h}}$, where C_f denotes a bounded absolute magnitude of a Fourier transform of true regression function f. $C_f < C$. Then, for an estimated neural network model with h hidden units, the global accuracy of the estimator is bounded as follows:*

$$E\left\|f - \hat{f}_{h,N,C_f}\right\|^2 \leq O(\frac{C^2}{h}) + O(\frac{hd}{N}logN) \tag{10}$$

where d is the dimension of inputs and N is the total number of training samples.
Furthermore, we can choose h = $O(\frac{d}{N}logN)^{\frac{1}{2}}$ and get the upper bound as follows:

$$E\left\|f - \hat{f}_{h,N,C_f}\right\|^2 \leq O(\frac{d}{N}logN)^{\frac{1}{2}} \tag{11}$$

This result states that the rate of convergence as a function of the sample size N is of order $(\frac{1}{N})^{\frac{1}{2}}$ (times a logarithmic factor), where the exponent $\frac{1}{2}$ is independent of the dimension d in terms of the behavior of the risk. Note that although the rate $(\frac{1}{N})^{\frac{1}{2}}$ as a function of N is independent of the dimension d, it is possible for the constant to be exponentially large in d for sequences of functions f of increasing dimensionality.

3.2 Universal Approximation

The universal approximation theorem states that a feed-forward neural network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact of R^d under assumptions on the activation function. It describes that a simple neural networks in the form of Equation (1) can represent a wide variety of interesting functions when given appropriate parameters.

First, Cybenko, George [5] described this universal approximation theorem for sigmoid functions with the following theorem.

Theorem 3.4 *Let ϕ be bounded measurable sigmoid activation function. Let I_d denote the d-dimensional unit hypercube $[0, 1]^d$. The space of continuous function on I_d is denoted by $C(I_d)$. Then, for any $\epsilon > 0$ and any function $f \in C(I_d)$, there exists an integer N, real constants $c_k, b_k \in R$ and real vectors $a_k \in R^d$, where k = 1, ..., h, such that we define a neural network of the form Equation (1) as an approximate realization of the function f. Then, there is a f_h for which*

$$|f - f_h| \leq \epsilon \tag{12}$$

for all $x \in I_d$. In other words, functions of the form f_h are dense in $C(I_d)$.

In addition, Kurt [9] showed that the choice the activation function is not specifically limited to a sigmoid function. Instead, Kurt described the universal approximate theorem is applicable regardless of the activation function. Thus, the multilayer feedforward architecture itself gives neural networks the potential of being universal approximators. The output units are always assumed to be linear.

Theorem 3.5 *Let ϕ be any continuous non-constant function from R to R. Then, a neural network of the form Equation (1) with an activation function ϕ is uniformly dense on compacta in C^r.*

Theorem 3.5 implies that a feed-forward neural network in the form of the Equation (1) is capable of arbitrarily accurate approximation to any real-valued continuous function over a compact set. This result indicates that the activation function ϕ is not required to be a squashing, sigmoid function even if it is allowed. The activation function may be any continuous non-constant function.

Theorem 3.6 *For every continuous non-constant function ϕ, every r, and every probability measure μ on (R^r, B^r), a feed-forward neural network in the form of the Equation (1) is ρ_μ-dense in M^r*

Theorem 3.6 implies that a feed-forward neural network with a single hidden layer in the form of the Equation (1) can approximate any measurable function arbitrarily well, regardless of the continuous non-constant function ϕ used, regardless of r and μ. It implies that feed-forward neural networks in the form of the Equation (1) is an universal approximation.

Theorem 3.7 *For every squashing function ϕ, every r, and every probability measure μ on (R^r, B^r), a feed-forward neural network in the form of the Equation (1) is uniformly dense on compacta in C^r and ρ_μ-dense in M^r.*

Theorem 3.7 implies that feed-forward neural networks in the form of the Equation (1) is an universal approximation again regardless of the squashing, sigmoid function ϕ (continuous or not).

Corollary 3.1 *For every function g in M^r, there is a compact subset K of R^r and an f in the form of the Equation (1) such that for any $\epsilon > 0$, we have $\mu(K) < 1 - \epsilon$ and for every $x \in K$ we have $|f(x) - g(x)| < \epsilon$, regardless of $\phi, r, or \mu$*

Corollary 3.1 implies that there is a feed-forward neural network with a single hidden layer in the form of the Equation (1) that approximates any measurable function to any desired degree of accuracy on some compact set K of input patterns that to the same degree of accuracy has measure, probability of occurrence 1. The difference between the Theorem 3.5 and the Corollary 3.1 is that in the Theorem 3.5, g is continuous and K is an arbitrary compact set. In the Corollary 3.1, g is measurable and K must be chosen specifically.

Theorem 3.8 *Let $\{x_1, ..., x_h\}$ be a set of distinct points in R^r and let $g : R^r \to R$ be an arbitrary function. If ϕ achieves 0 and 1, then there is a feed-forward neural network in the form of the Equation (1) with h hidden units such that $f(x_i) = g(x_i), i \in \{1, ..., h\}$.*

Theorem 3.8 implies that functions with finite support can be approximated exactly with a single hidden layer.

Corollary 3.2 *Theorem 3.7 and Corollary 3.1 remain valid for multilayer feed-forward neural networks with l layers mapping $R^r toR^s$ using squashing, sigmoid functions in $C^{r,s} andM^{r,s}$ with $\rho_\mu^s, \rho_\mu^s(f, g) \equiv \sum_{i=1}^{s} \rho_\mu(f_i, g_i)$*

Corollary 3.2 implies that multi-output multilayer feed-forward neural networks are universal approximators of vector-valued functions. It describes the approximation capabilities of multi-output neural networks with multiple hidden layers.

3.3 Convergence Analysis of Two-layer Neural Networks with ReLU Activation

Theorem 3.9 *There exists constants $\gamma > \gamma_0 > 0$ such that if $x \sim N(0, I)$, $\|W_0\|_2, \|W^*\|_2 \leq \gamma_0, d \geq 100, \epsilon\gamma^2, thenSGDforL(W)willfindthegroundtruthW^*bytwophases.$ (Phase 1) when we set $\eta \leq \frac{\gamma^2}{G_2^2}$, the potential function will keep decreasing until it is smaller than $197\gamma^2$, which takes at most $\frac{1}{16}$ steps. (Phase 2) for any $\alpha > 0$ and any T such that $T^\alpha logT \geq \frac{36d}{100^4(1+\alpha)G_F^2}$, if we set $\eta = \frac{(1+\alpha)logT}{\delta T}$, we have $E\|W_T - W^*\|_F^2 \leq \frac{100^2(1+\alpha)logTG_F^2}{9T}$.*

Theorem 3.9 implies that W_T will be sufficiently close to W^* with small step size η.

4 Proof Outlines

4.1 Approximation Error Bounds for Neural networks

Theorem 3.1: Let constrain $|a_k|_1$ to be no larger than τ_h, where τ_h indicates the rate at which $\phi(z)$ approaches its limits. It is bounded by a polynomial function of h i.e., $\tau_h \leq r_0 h^{r_1}$ for some $r_0, r_1 > 0$. Let denote dist(ϕ_τ, sgn) the distance between the scaled sigmoid function and the signum function as follows: $\inf_{0<\epsilon<\frac{1}{2}}(2\epsilon + \sup_{|z|\geq\epsilon} |\phi(\tau z) - sgn(z)|)$.

Then, under the same restrictions in Theorem 3.1, there exists a neural network model of the Equation (1) such that $\|f - f_h\| \leq \delta + C_f$ dist(ϕ_τ, sgn). If we assume that τ is chosen accordingly such that we have dist(ϕ_τ, sgn) $\leq \frac{1}{\sqrt{h}}$. Then, $\|f - f_h\| \leq \frac{2C_f}{\sqrt{h}}$.

Theorem 3.2: The detailed proof can be found in [2]. The key idea is based on that it is possible to obtain bounds on the total mean square eror by controlling these sources of error. By the triangle inequality, we have the following statement: $\left\|f - \hat{f}_{h,N}\right\| \leq \|f - f_h\| + \left\|f_h - \hat{f}_{h,N}\right\|$.

Theorem 3.3: Let Ω be a discrete set of parameter points. For every $\theta \in \Omega$, there is a θ^*, such that $|a_k - a_k^*|_1 \leq \epsilon$, $|b_k - b_k^*| \leq \epsilon$, $\sum_{k=1}^{h} |c_k - c_k^*| \leq C\epsilon$ and $|c_0 - c_0^*| \leq C\epsilon$ for $\epsilon > 0$ and $C \geq 1$ uniformly for x. Then, we have $|f_h(x, \theta) - f_h(x, \theta*)| \leq 4vC\epsilon$, where $f_h(x, \theta)$ is the family of sigmoid networks of the Equation (1).

For function f with $C_f \leq C$, there exists a neural network approximation f_h with parameter restriction such that $\|f - f_h\| \leq \frac{2C_f}{\sqrt{h}} + 4vC\epsilon$. If a ϵ_h is selected to be order $O(1/\sqrt{h})$, then the approximation error remains the same order as Theorem 3.1. $\|f - f_h\| = O(C/\sqrt{h})$.

We can also find the bound of $log\#\Omega_{h,\epsilon,\tau_h,C}$ as $m_h log(\frac{2e(1+\tau_h}{\epsilon})$, where $m_h = h(d+2)+1$. Then, we can bound the index of resolvability in Equation (2) as follows: $R_{h,N}(f) \leq \|f - f_h\|^2 + \frac{\lambda}{N} log\#\Omega_{h,\epsilon,\tau_h,C} \leq O(\frac{C^2}{h} + O(\frac{hd}{N} log\frac{\tau_h^2 N}{hd})$. Thus, by Theorem 3.2, we achieve $E\left\|f - \hat{f}_{h,N,C_f}\right\|^2 \leq O(\frac{C^2}{h}) + O(\frac{hd}{N} logN)$.

4.2 Universal Approximation

Theorem 3.4: Let $S \subset C(I_d)$ be the set of functions of the form $f_h(x)$ in the Equation (1). S is a linear subspace of $C(I_d)$, so that the closure of S is all of $C(I_d)$.

If we assume that the closure of S is not all all of $C(I_d)$, the closure of S is a closed proper subspace of $C(I_d)$. By Hahn-Banach theorem [12], there is a bounded linear functional on $C(I_d)$, L. By the Riesz Representation Theorem [8], this bounded linear functional, L has the form of L(h) for some $\mu \in M(I_d)$. This condition implies that $\mu = 0$. However, this condition contradicts our assumption that ϕ is discriminatory. Thus, the subspace S must be dense in $C(I_d)$. $f_h(x)$ is dense in $C(I_d)$, providing that ϕ is continuous and discriminatory.

Then, we can show that any continuous sigmoid function is discriminatory. By the Lesbegue Bounded Convergence Theorem [14], we can derive the measure of all half-planes being 0. This implies that the measure μ itself must be 0. We can check that the Fourier Transform of μ is 0 and so μ must be zero as well. Hence, any continuous sigmoid function is discriminatory.

Theorem 3.5: Let A be an algebra of real continuous functions on a compact set K. If A separates points on K and if A vanishes at no point of K, then the uniform closure B of A consists of all real continuous functions on K. This implies that A is ρ_K-dense in the space of real continuous functions on K from Stone-Weierstrass Theorem.

Let $K \subset R^r$ be any compact set. For any ϕ, a neural network with the form of the Equation (1) is an algebra on K. We can ensure that it is separating on K. The Stone-Weierstrass Theorem implies that a neural network is ρ_K-dense in the space of real continuous functions on K. The results follows as K is arbitrary.

Theorem 3.6: Given any continuous non-constant function, from the Theorem 3.5 and the fact that if $\{f_n\}$ is a sequence of functions in M^r that converges uniformly on compacta to the function f, then $\rho_\mu(f_n, f) \to 0$, a feed-forward neural network in the form of the Equation (1) is ρ_μ-dense in C^r.

For any finite measure μ, C^r is ρ_μ-dense in M^r. As C^r is ρ_μ-dense in M^r, it follows that a neural network in the form of the Equation (1) is ρ_μ-dense in M^r by applying triangle inequality.

The extension from continuous to arbitrary squashing function can be achieved with the following statement. Let F be a continuous squashing function and an arbitrary squashing function. For every $\epsilon > 0$, there is an element H_e of a neural network in the form of the Equation (1) such that $sup_{\lambda \in R}|F(\lambda) - H_e(\lambda)| < \epsilon$.

Theorem 3.7: For every squashing function, a neural network in the form of the Equation (1) is uniformly dense on compacta in C^r. if $\{f_n\}$ is a sequence of functions in M^r that converges uniformly on compacta to the function f, then $\rho_\mu(f_n, f) \to 0$.

It implies that a neural network in the form of the Equation (1) is ρ_μ-dense in C^r. For any finite measure μ, C^r is ρ_μ-dense in M^r. With the triangle inequality, we can find that a neural network in the form of the Equation (1) is ρ_μ-dense in M^r

Corollary 3.1: Let fix $\epsilon > 0$. By Lusin's theorem [1], there is a compact set K^1 such that $\mu(K^1) > 1 - \frac{\epsilon}{2}$ and $g|K^1(g$ restricted to K^1 is continuous on K^1. By the Tietz extension theorem [6], there is a continuous function $g' \in C^r$ such that $g'|K' = g|K^1$ and $sup_{x \in K^1} g|K^1(x)$.

For every k the class of J_k that indicates a neural network is uniformly dense on compacta in C^r. If we take f such that $sup_{x \in K^2}|f(x) - g'(x)| < \epsilon$. Then, $sup_{x \in K^1 \cap K^2}|f(x) - g(x)| < \epsilon$ and $\mu(K^1 \cap K^2) > 1 - \epsilon$

Theorem 3.8: We can first prove its validity when $\{x_1, ..., x_h\} \subset R^1$. Let order x so that $x_1 < x_2 < ... < x_{h-1}, < x_h$. Let pick $M > 0$ such that $\phi(-M) = 1 - \phi(M)$. Let define A_1 as the constant affine function $A_1 = M_0$, set $c_1 = g(x_1)$, and set $f^1(x) = c_1\phi(A_1(x_1))$.

Since $f^1(x) \equiv g(x_1)$, we have $f^1(x_1) = g^1(x_1)$. Inductively, we define A_h by $A_h(x_{h-1}) = -M$ and $A_h(x_h) = M$ and define $c_h = g(x_h) - g(x_{h-1})$. And set $f^h(x) = \sum_{k=1}^{h} c_k \phi(A_k(x))$. For $k \leq l, f^h(x_k) = g(x_k)$. Then, f(x) is the desired function. We can also extend the results to R^r.

Corollary 3.2: We can approximate each g_i to become within $\frac{\epsilon}{\delta}$, using vectors c_i that is 0 except in the i-th position. Adding together δ approximations keeps us within the classes of approximate functions in $C^{r,s}$ and $M^{r,s}$.

Let F (resp. G) be a class of functions from R to R (resp. R^r to R) that is uniformly dense on compacta in C^r (resp. C^r). The class of functions $G \circ F = \{f \circ g : g \in Gandf \in F\}$ is uniformly dense on compacta in C^r. Then, for every k the class of J_k that indicates multioutput multilayer neural networks is uniformly dense on compacta in C^r. For every squashing function, a neural network is uniformly dense on compacta in C^r. We can complete the proof with the induction hypothesis.

4.3 Convergence Analysis of Two-layer Neural Networks with ReLU Activation

Theorem 3.9: First, we would like to check whether L is one-point convex. When we check the negative gradient in the Section 5, the first two terms have positive inner products. The last term can point to arbitrary direction. If the last term is small, it can be covered by the first two terms. We can define a potential function closely related to the last term. Thus, L becomes one-point strongly convex.

(Phase 1) We then show the potential function actually decreases O(1) after polynomial number of iterations by getting joint update rules. After solving this dynamics, we can show that g_t will approach to (and stay around) O(γ). There exists a constant $\gamma > \gamma_0 > 0$ such that if $\|W_0\|_2$, $\|W^*\|_2 \leq \gamma_0$, $d \geq 100, \eta_l e \frac{\gamma^2}{G_2^2}, \epsilon \leq \gamma^2$, then g_t will keep decreasing by a factor of $1 - 0.5\eta d$ for every step until $g_{t_1} \leq 197\gamma^2$ for step $t_1 \leq \frac{1}{16\eta}$.

(Phase 2) We can use the Taylor expansion and control the higher order terms (i.e. write $\theta_{i^*,j}$ = argcos and expand arccos at point 0. Then, we consider joint Taylor expansion. As W approaches W^*, the second term of joint Taylor expansion tends to be zero. There exists a constant γ such that if $\|W\|_2$, $\|W^*\|_2 \leq \gamma$, $g \leq 0.1$, then, $<-\nabla L(W), W^* - W> > 0.03 \|W^* - W\|_F^2$. Thus, our approximation has bounded error. By setting $\delta = 0.03$, $D = \frac{\sqrt{d}}{50}, G = G_F$ with our definitions, we can get the convergence guarantee.

5 Conclusions

This paper describes statistical analysis on shallow neural networks. First, approximation error bound of neural network with single hidden layers and sigmoid functions is described, which is O($N^{-\frac{1}{2}}$). It implies that a single hidden layer neural network does not require exponentially large sample numbers to achieve the same approximate errors. In addition, we describe the expressive power of neural network by showing universal approximation. Specifically, single-layer neural network with a sigmoid function can represent a various functions under the mild assumptions of activation functions. We also show that this result is applicable to any continuous non-constant activation function. Lastly, we describe convergence analysis for SGD on a subset of two-layer neural networks with ReLU activation. Assuming that input is sampled from Gaussian distribution with standard O($\frac{1}{\sqrt{d}}$) initialization of the weights, we describe SGD converges to global minimum in polynomial number of steps.

Although the presented results are not applicable to the deep neural networks, the results on shallow neural network provide an insight why the application of neural networks to high-dimensional settings can lead to meaningful representation/analysis. For the understanding of the current popularity of deep learning in various domains (e.g. speech, vision, NLP), more rigorous analysis on neural networks is necessary.

References

[1] S. Alpern and R. D. Edwards. Lusin's theorem for measure preserving homeomorphisms. *Mathematika*, 26(1):33–43, 1979.

[2] A. R. Barron. Complexity regularization with application to artificial neural networks. In *Nonparametric functional estimation and related topics*, pages 561–576. Springer, 1991.

[3] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[4] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1):115–133, 1994.

[5] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[6] J. Dugundji. An extension of tietze's theorem. *Pacific Journal of Mathematics*, 1(3):353–367, 1951.

[7] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning.(2016). *Book in preparation for MIT Press. URL: http://www. deeplearningbook. org*, 2016.

[8] R. K. Goodrich. A riesz representation theorem. *Proceedings of the American Mathematical Society*, 24(3):629–636, 1970.

[9] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[10] Y. Li and Y. Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[11] J. MARKOFF. Scientists see promise in deep-learning programs, ny times. *http://nyti. ms/sgcVec*, 2012.

[12] B. Nowak and A. Trybulec. Hahn-banach theorem. *Journal of Formalized Mathematics*, 5(199):3, 1993.

[13] D. Saad and S. A. Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. In *Advances in neural information processing systems*, pages 302–308, 1996.

[14] S. Simons. A convergence theorem with boundary. *Pacific Journal of Mathematics*, 40(3):703–708, 1972.

[15] Y. Tian. Symmetry-breaking convergence analysis of certain two-layered neural networks with relu nonlinearity. 2016.