SAMPLE COMPLEXITY OF DEEP ACTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many machine learning algorithms require large numbers of labeled training data to deliver state-of-the-art results. However, in many domains of AI, there are abundant unlabeled data but it is costly to get data labeled by experts, such as medical diagnosis and fraud detection. In these domains, active learning, where an algorithm maximizes model accuracy while requiring the least number of labeled data, is appealing. Active learning uses both labeled and unlabeled data to train models, and the learning algorithm decides which subset of data should acquire labels. Due to the costly label acquisition, it is interesting to know whether it is possible from a theoretical perspective to understand how many labeled data are actually needed to train a machine learning model. This question is known as the sample complexity problem, and it has been extensively explored for training linear machine learning models (e.g., linear regression). Today, deep learning has become the de facto method for machine learning, but the sample complexity problem for deep active learning remains unsolved. This problem is challenging due to the non-linear nature of neural networks. In this paper, we present the first deep active learning algorithm which has a provable sample complexity. Using this algorithm, we have derived the first upper bound on the number of required labeled data for training neural networks. Our upper bound shows that the minimum number of labeled data a neural net needs does not depend on the data distribution or the width of the neural network but is determined by the smoothness of non-linear activation and the dimension of the input data.

1 INTRODUCTION

Deep learning has revolutionized our society by achieving unprecedented breakthroughs in various challenging AI tasks, including face detection, text- and image-based search, personal assistants, and autonomous driving. Deep learning models can have millions to billions of parameters, and training deep learning models usually require abundant labeled training data for these models to deliver state-of-art results (Brown et al., 2020; Devlin et al., 2018; Radford et al., 2019). There have been many impactful efforts to bringing more high-quality labeled datasets into our research community (Bengio et al., 2007; Krizhevsky et al., 2012). Unfortunately, in many important areas of AI, there are abundant unlabeled data but acquiring the correct label for them is costly. For example, in healthcare, asking expert radiologists to manually diagnose patients' medical images is more expensive than taking the medical images using increasingly cheaper imaging devices. Another standard example is natural language processing: plenty of texts have already existed on the Internet, but labeling them requires additional human effort.

In these domains, *active learning*, where the learning algorithms can use the least number of labeled data to achieve high model accuracy, is appealing (Balcan et al., 2009). In active learning, the learner has access to a set of unlabeled training data. The learner can select a subset of the unlabeled data for an *oracle* to label. The choice of the subset is usually based on the characteristics of the unlabeled data, for example, uncertainty (Beluch et al., 2018; Joshi et al., 2009; Ranganathan et al., 2017; Lewis & Gale, 1994; Seung et al., 1992; Tong & Koller, 2001), diversity (Bilgic & Getoor, 2009; Gal et al., 2017; Guo, 2010; Nguyen & Smeulders, 2004), and expected model change (Freytag et al., 2014; Roy & McCallum, 2001; Settles et al., 2007). The oracle returns the correct labels of the selected unlabeled data. After this, the learner uses the labels on the selected data to train an accurate model.

Due to the strong performance of deep learning and the active learning's potential to substantially reduce the labeling costs, the oblivious approach is to combine them. This is known as *deep active learning*, and it has recently received significant attention in AI research. Researchers have combined supervised and semi-supervised learning on labeled and unlabeled data to enable deep learning with fewer labeled training data (Hossain & Roy, 2019; Siméoni et al., 2021). Other works target improving existing active learning's sampling strategies for deep neural networks (Sener & Savarese, 2017; Settles, 2009; Ash et al., 2019; Gissin & Shalev-Shwartz, 2019; Kirsch et al., 2019; Zhdanov, 2019). Today, deep active learning has been widely used in various domains, including object recognition and text classification (Du et al., 2019c; Gal et al., 2017; Gudovskiy et al., 2020; Huang et al., 2019; Shen et al., 2017; Zhang et al., 2017; Zhou et al., 2013; Aghdam et al., 2019; Feng et al., 2019; Qu et al., 2020).

One of the most important theoretical questions in active learning is to understand how many labeled data are necessary to train a model that has a high accuracy. This problem is known as the sample complexity problem in active learning, and it has been well studied in the context of linear models, e.g., linear regression (Chen & Price, 2019; Boutsidis et al., 2013; Song et al., 2019).

This raises an important *theoretical* question:

What's the minimal number of labeled data needed for deep active learning?

This question is important because deep active learning has the potential to substantially reduce the cost of AI by reducing the number of labeled data required. At the same time, this question is challenging, because a neural network is a non-linear model and traditional techniques to estimate the sample complexity for linear models do not apply.

In this paper, we present the first active deep learning algorithm with a provable sample complexity. Our algorithm is based on former research in the spectral sparsification for graphs (Spielman & Teng, 2004; Spielman & Srivastava, 2011; Batson et al., 2012; Lee & Sun, 2018). Our main intuition is that similar to spectral sparsifiction for graphs, we can also view active learning as a sparsification problem: we want to get a sparse representation for a large dataset. Our algorithm is based on the randomized BSS algorithm (Lee & Sun, 2018), and we expand it to active learning on non-linear functions.

Using this algorithm, we prove the first upper bound on the number of required labeled data for training one-hidden layer neural networks. Our results show that the minimum number of labeled data a neural net needs does not depend on the data distribution or the width of the network but is determined by the smoothness of non-linear activation and the dimension of the input data. Besides, our results also show that more labels in a dataset do not necessarily improve the prediction result.

To bound the sample complexity, we first project the non-linear neural network into a space with a ρ-nearly orthonormal basis. Decomposing a non-linear neural network into an orthonormal basis is hard, but using the nearly orthonormal basis is easier. We then define a class of procedure called importance sampling procedure. Any sampling procedure that is an importance sampling procedure can select sufficient labels to train an optimal predictor. So the data selected by an importance sampling procedure have the ability to recover the entire distribution. We finally find two useful importance sampling procedures to complete our proof. The first one samples our unlabeled dataset by i.i.d. sampling from an unknown distribution. We show that when the dataset is sufficiently large, it is an importance sampling procedure and it has the potential to approximate the original unknown distribution. Then, we introduce the second importance sampling procedure which is based on randomized BSS. We apply this algorithm with a uniform distribution on the unlabeled dataset. We show that it only requires a small number labels. Correspondingly, we need to change our training objective into a weighted one which can be achieved by using a novel data sampler.

This paper makes the following contributions:

- We provide the first formulation of the sample complexity problem for deep active learning.
- We propose a ρ-nearly orthonormal basis for one-hidden layer neural network.
- We introduce a new active learning algorithm for one-hidden layer neural network.
- We present a novel data sampler in the optimization of neural networks.

2 Related Work

Active learning Active learning aims to select a few most useful data to acquire labels (Balcan et al., 2009). This allows training an accurate machine learning model and at the same time minimizes labeling costs. There are three types of active learning: membership query synthesis, stream-based sampling, and pool-based sampling. In membership query synthesis, the learner can query a label for any data in the input space (Angluin, 1988; King et al., 2004). The data does not have to be in the set of unlabeled data. Stream-based sampling means the active learning algorithm has to decide whether to query for label based on each individual data in a stream (Dagan & Engelson, 1995; Krishnamurthy, 2002). Pool-based sampling means that the active learning algorithm can access the entire set of unlabeled data and then decide which subset to query the oracle (Lewis & Gale, 1994). Our results are based on pool-based sampling, which is more common in practice (Ren et al., 2020). To decide whether to query label for a given dataset, there has been many works that show strategies based on uncertainty (Beluch et al., 2018; Joshi et al., 2009; Ranganathan et al., 2017; Lewis & Gale, 1994; Seung et al., 1992; Tong & Koller, 2001), diversity (Bilgic & Getoor, 2009; Gal et al., 2017; Guo, 2010; Nguyen & Smeulders, 2004), or expected model change (Freytag et al.,

Algorithm 1 Framework of Our Active Learning Algorithm

procedure REGRESSIONUNKNOWNDISTRIBUTION($\varepsilon, \mathcal{F}_{\mathsf{nn}}, \mathcal{X}, P$)
  ▷ $|\mathcal{X}| = O(K_D \log(\overline{d}) + K_D/\varepsilon)$
  Let D_0 be the uniform distribution over $\mathcal{X} = (x_1, \ldots, x_{k_0})$.
  Generate weight u_i and samples $x_i,\ i \in [k]$ from a good output of P with parameters $\mathcal{F}_{\mathsf{nn}}, D_0, \Theta(\epsilon)$.
  Label x_i with y_i with $y_i \sim Y(x_i),\ i \in [k]$.
  Output $\widetilde{W} \leftarrow \arg\min_{W \in \mathbb{R}^{d \times m}} \sum_{i=1}^{k} u_i \cdot |f_{\mathsf{nn}}(W, x_i) - y_i|^2$.
end procedure

2014; Roy & McCallum, 2001; Settles et al., 2007) are effective. Many researchers (Ash et al., 2019; Shui et al., 2020; Yin et al., 2017; Zhdanov, 2019) have also explored combinations of these strategies.

Deep active learning Deep learning has become the de facto method for machine learning, and thus it is the obvious area to apply active learning. This is known as deep active learning, and it has recently received much attention because it is difficult to acquire labels in many domains of deep learning. One challenge of deep active learning is that deep learning inherently requires abundant labeled training data, because deep neural networks have millions to billions of parameters (Brown et al., 2020; Devlin et al., 2018; Radford et al., 2019). However, traditional active learning algorithms often rely on a small number of labeled data to learn. There has been a lot of effort in the research community to integrate active learning into deep learning. Some works focuses on improving sampling strategies (Sener & Savarese, 2017; Settles, 2009; Ash et al., 2019; Gissin & Shalev-Shwartz, 2019; Kirsch et al., 2019; Zhdanov, 2019), and other works target improving neural network training methods (Hossain & Roy, 2019; Siméoni et al., 2021). Our paper focuses on an important theoretical aspect of deep active learning: the sample complexity, i.e., the number of labeled data required to achieve high model accuracy.

Theoretical active learning Many theoretical works have considered ways to perform active learning in linear regression tasks. They consider subsample linear regression problems in which the solution to the subsampled problem approximates the overall solution. A trivial approach is uniform sampling (Cohen et al., 2013; Hsu & Sabato, 2016). However, It can be significantly improved by the most common approach which uses leverage score sampling (Drineas et al., 2008; Magdon-Ismail, 2010; Mahoney, 2011; Woodruff, 2014). To go beyond the leverage score sampling complexity, some works (Boutsidis et al., 2013; Song et al., 2019) apply the deterministic linear-sample spectral sparsification method proposed by (Batson et al., 2012). Other works (Dereziński & Warmuth, 2017; Derezinski et al., 2018) utilize volume sampling to improve the sample complexity. There are also works (Sabato & Munos, 2014; Chaudhuri et al., 2015) with additional assumptions or simplifications. Our paper is the first theoretical work on the sample complexity of deep active learning. Our main challenges are to formulate the sample complexity problem and deal with the non-linear nature of neural networks.

3 PRELIMINARY

In this paper, we consider a regression task. We first define some notations here.

- We use $[k]$ to denote the set $\{1, 2, \cdots, k\}$.
- We denote unlabeled training data $x_i \in \mathbb{R}^d,\ i \in [k_0]$, where k_0 denotes the number of unlabeled training data. We also use $\mathcal{X} = (x_1, \ldots, x_{k_0})$ to denote the unlabeled dataset.
- For convenience, we use $x_i \in \mathbb{R}^d,\ i \in [k]$ to denote the labeled training data.
- We denote corresponding label as $y_i \in \mathbb{R}, i \in [k]$.
- We use (D, Y) to denote an unknown joint distribution where the data and label are from.
- We use $\|h(x)\|_D^2$ to denote $\mathbb{E}_{x \sim D}[h^2(x)]$.
- We use $v : \mathbb{R}^d \to \mathbb{R}^{\overline{d}}$ to denote $\big(v_1(x), \ldots, v_{\overline{d}}(x)\big)$ where $x \in \mathbb{R}^d$.

3.1 ACTIVE LEARNING

In the setting of active learning, We assume that our data $x \in \mathbb{R}^d$ and corresponding labels $y \in \mathbb{R}$ are sampled from an unknown joint distribution $(x, y) \sim (D, Y)$. More specifically, we sample data $x \sim D$ from distribution D and obtain

the unlabeled dataset $\mathcal{X} = [x_1, \dots, x_{k_0}] \in \mathbb{R}^{k_0 \times d}$. Since labeling is costly, we only select k samples and label them according to the conditional distribution $y \sim Y(x)$. We use our sampled data and the corresponding labels to train our neural network.

The global framework of our algorithm is shown in Algorithm 1. In Step 4, we perform our proposed query algorithm which selects samples $x_i,\ i \in [k]$ in the large dataset and generate an weight $u_i,\ i \in [k]$. Finally, as shown in Step 6, we use a weighted objective to obtain the optimal parameter of the neural network.

3.2 Neural Networks

For simplicity, we only consider one hidden layer neural network in this paper. Our result holds for a large class of non-linear functions including any polynomial, ReLU, Sigmoid, and Swish. We now provide the definition of a neural network function.

Definition 3.1 (Two layer neural network)**.** *Let $w_r \in \mathbb{R},\ r \in [m]$ be the weight vector of the first layer, $a_r \in \mathbb{R},\ r \in [m]$ be the output weight. We define a two layer neural network*

$$f_{\mathsf{nn}} : \mathbb{R}^{d \times m} \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}$$

as the following form

$$f_{\mathsf{nn}}(W, a, x) := \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \phi(w_r^\top x) \in \mathbb{R}$$

where $x \in \mathbb{R}^d$ is the input and $\phi(\cdot)$ is the non-linear activation function. We will also define $W = [w_1, \cdots, w_m]^\top \in \mathbb{R}^{d \times m}$ and $a = [a_1, \cdots, a_m]^\top \in \mathbb{R}^m$ for convenience.

Similar to other works in theoretical deep learning (Li & Liang, 2018; Allen-Zhu et al., 2019a;b; Du et al., 2019a;b; Song & Yang, 2019; Brand et al., 2021), we use normalization $1/\sqrt{m}$ and consider only training W while fixing $a \in \{-1, +1\}^m$. Note that each entry of W are initialized to be $\mathcal{N}(0, 1)$. So, we can write $f_{\mathsf{nn}}(W, x) = f_{\mathsf{nn}}(W, a, x)$.

To prove the above result, without loss of generality, we make the following assumption which bounds the samples $x \in \mathbb{R}^d$.

Assumption 3.2 (Bounded samples)**.** *For the distribution D, we have that $\max_{x \in \mathsf{supp}(D)} \|x\|_2 \leq 1$.*

The assumption basically says that there is a bound on the maximum value in the input data. We can always achieve this by rescaling the data. Inputs are bounded is a standard assumption in the optimization field (Lee et al., 2020; Li & Liang, 2018; Allen-Zhu et al., 2019a;b).

3.3 Condition Number

We use condition number K to measure the concentration of neural network on distribution D. Intuitively, concentration means that there does not exist $x \in D$ that can lead to a very large value in neural network f_{nn}. We prove that both the number of required unlabeled samples and the number of required labeled data are proportional to the condition number of distribution D. We define condition number as follows:

Definition 3.3 (condition number)**.** *Let D be the marginal distribution over x. We will define "condition number" as follows*

$$K := \sup_{W \in \mathbb{R}^{d \times m} : W \neq 0} \frac{\sup_{x \in G} |f_{\mathsf{nn}}(W, x)|^2}{\|f_{\mathsf{nn}}(W, x)\|_D^2}. \tag{1}$$

Note that this definition for condition number is also used by (Chen & Price, 2019). However, they target linear functions, where this paper focuses on the non-linear case.

4 Our Results

Our main result is the following upper bound of required labeled samples and increasing the number of labeled samples further would not improve the prediction accuracy of a neural network. The theorem states that when the number of unlabeled samples k_0 and the number of required labels k is large enough, the prediction accuracy difference is negligible between the optimal predictor $f_{\mathsf{nn}}(W^*, x)$ trained by unlimited labeled data and the optimal predictor $f_{\mathsf{nn}}(\widetilde{W}, x)$ trained by sampled data with only k labels.

Theorem 4.1. *Let $f_{\mathsf{nn}}(W, x)$ be a neural network as defined in Definition 3.1, and consider any (unknown) distribution on (x, y) over $\mathbb{R}^d \times \mathbb{R}$. Suppose that $C = (10d + \log(1/\varepsilon_0)/\log(d))$ and the C-th derivative of the activation function ϕ of f_{nn} satisfied that*

- *$\phi^{(C)}(x)$ exists and is continuous.*
- *$\phi^{(C)}(x) \leq 1,\ x \in \mathbb{R}$.*

Let D be the marginal distribution over x, and suppose it has bounded "condition number"

$$K := \sup_{W \in \mathbb{R}^{d\times m}: W \neq 0} \frac{\sup_{x\in G} |f_{\mathsf{nn}}(W, x)|^2}{\|f_{\mathsf{nn}}(W, x)\|_D^2}. \tag{2}$$

Let $W^* \in \mathbb{R}^{d\times m}$ minimizes $\mathbb{E}_{(x,y)\sim(D,Y)}[|f_{\mathsf{nn}}(W, x) - y|^2]$. For any $0 < \varepsilon \leq O(1/\log^3(\overline{d}))$, $\rho \in (1, 1/10)$, and $\varepsilon_0 \in (0, 1/10)$, there exists

$$\overline{d} \leq \binom{10d + \log(1/\varepsilon_0)/\log(d)}{d}$$

and an randomized algorithm P that takes $O((1 + \rho\overline{d})(K \log(\overline{d}) + K/\varepsilon))$ unlabeled samples from D and requires $O(\overline{d}/\varepsilon)$ labels to output $\widetilde{W} \in \mathbb{R}^{d\times m}$ such that

$$\mathop{\mathbb{E}}_{P} \mathop{\mathbb{E}}_{x\sim D}[|f_{\mathsf{nn}}(\widetilde{W}, x) - f_{\mathsf{nn}}(W^*, x)|^2] \leq \varepsilon_0 + \varepsilon(1 + \rho\overline{d}) \cdot \mathop{\mathbb{E}}_{(x,y)\sim(D,Y)}[|y - f_{\mathsf{nn}}(W^*, x)|^2].$$

The size of unlabeled dataset depends on both the distribution D and desirable accuracy ε. Note that the number of size of unlabeled dataset is also proportional to the condition number and $1/\varepsilon$. However, our upper bound of required labeled data does not depend on the data distribution or the width of the neural network but is determined by the smoothness of non-linear activation and the dimension of the input data.

To test our model, we can choose any $(x, y) \sim (D, Y)$ and measure the ℓ_2 distance between the prediction results of the practical optimal model and the theoretical optimal model. To make our result reasonable, we take expectation over the distribution (D, Y). Note that $\mathbb{E}_{x,y}[|y - f_{\mathsf{nn}}(W^*, x)|^2]$ is an internal constant measuring the expressive power of neural network f_{nn}. We prove the perturbation between predictor loaded practical optimal weights $\widetilde{W}$ and predictor loaded theoretical optimal weights W^* within up to a constant scaled with ε. Another way to understand the perturbation bound is equivalently rewrite the bound in Theorem 4.1 as

$$\mathop{\mathbb{E}}_{P} \mathop{\mathbb{E}}_{(x,y)\sim(D,Y)}[|y - f_{\mathsf{nn}}(\widetilde{W}, x)|^2] \leq \varepsilon_0 + (1 + \varepsilon) \cdot \mathop{\mathbb{E}}_{(x,y)\sim(D,Y)}[|y - f_{\mathsf{nn}}(W^*, x)|^2].$$

which shows that training with limited labeled samples can achieve very similar prediction accuracy compared with training with unlimited labeled data.

However, treat each labeled data equally when we train a neural network would not obtain the optimal predictor because some of the data and labels are more important than others. It's natural to include the consideration of importance into the training strategy. To obtain the optimal result, we should combine our new query algorithm with a novel data sampler. Our theorem is as follows:

Theorem 4.2 (Data Sampler)**.** *Let the neural network $f_{\mathsf{nn}}(W, x)$, condition number K, algorithm P, labeled samples x_i, corresponding label y_i, $i \in [k]$ and, practically optimal parameter $\widetilde{W}$ be defined as in Theorem 4.1. Our algorithm P generates weights u_i, $i \in [k]$. We claim that the optimal parameter $\widetilde{W}$ can be obtained by optimizing with weighted objective as follows*

$$\widetilde{W} = \arg\min_{W\in\mathbb{R}^{d\times m}} \left\{ \sum_{i=1}^{k} u_i \cdot |f_{\mathsf{nn}}(\widetilde{W}, x_i) - y_i|^2 \right\}.$$

This theorem claims that a weighted objective is a better choice. Practically, This theorem implies that we should sample our data proportional to the weight w_i if we use stochastic gradient descent to train our network. If we sample with $\Pr[x_i] = u_i / \sum_{j=1}^{k} u_j$, then we can use the weighted objective to train to obtain the optimal result. Under this sampling strategy, we can get the optimal parameter $\widetilde{W}$ claimed in Theorem 4.1.

5 OVERVIEW OF TECHNIQUES

In this section, we first claim how to select x_i and generate u_i as mentioned in Step 4. We then provide a high-level view of our approach. Finally, we introduce our techniques in detail.

5.1 OUR ALGORITHM

Our algorithm is shown as in Definition 5.5. Our algorithm executes in an iterative way. In each iteration, it generates a coefficient β_i, a distribution D_i, sample $x_i \sim D_i$ from the distribution, and calculates the weight u_i. After each iteration, coefficient β_i and distribution D_i change. In Lemma 5.8, we provide an effective procedure that can fit into the framework and we get an upper bound of the required labels as shown in Step 4 in Algorithm 1. Full details about this algorithm can be found in Appendix. In Lemma 5.9, we provide an upper bound on the number of required samples. This shows that the requirement on the size of unlabeled dataset $|\mathcal{X}| = k_0$ in Algorithm 1 is reasonable.

5.2 HIGH-LEVEL APPROACH

Our goal is to find an algorithm that can recover the optimal predictor with finite data and limited labels. However, this problem is hard to tackle directly because a neural network is a non-linear function. Our main approach is that we need to project the non-linear neural network f_{nn} into an orthonormal basis.

However, it is hard to find an exact orthonormal basis for a non-linear neural network f_{nn}. Instead, we introduce the ρ-nearly orthonormal basis. We can take advantage of the basis and turn the problem of studying the property of a non-linear function into studying the property of a function space with a nearly orthonormal basis. In the active learning problem, we can separate our problem into two levels. We should first provide a bound on the unlabeled dataset, then provide a bound on the minimum number of labels that we need. In another word, we should handle those two problems:

- We provide an upper bound of the size of the training dataset under which the optimal predictor can be recovered.
- We provide an upper bound of the number of required labeled data that can guarantee the recovery.

Although those two problems seem very different, technically we can handle them similarly. To see this, we can consider that the dataset is sampled from the unknown distribution D. If we can find a general condition that can deduce the recovery guarantee, we can check the condition for our algorithm proposed for each of the two parts. We call this condition importance sampling procedure. If a procedure is importance sampling procedure, we can guarantee the recovery.

First, we handle the dataset forming part. Normally, we obtain the dataset by i.i.d sampling in a distribution D. We show that this procedure is an importance sampling procedure when the dataset is sufficiently large. So, if we use this algorithm to train our neural network, it would take too many data labels. This algorithm is not efficient and is not a good choice for active learning. However, if we sample i.i.d. from the dataset, we get a known distribution and thus we can apply our randomized sampling on it.

Second, we handle the query strategy part. We show that our randomized sampling is an importance sampling procedure and only required a limited number of labels. We build our algorithm on the procedure. However, this algorithm required known distribution D_0. We show that a uniform distribution D_0 on our dataset is a good choice.

Now, we can conclude that there are five major steps to prove our main result.

1. By picking a sufficiently large dimension $\overline{d}$, we prove that neural network can be decomposed into a ρ-nearly orthonormal function family $\mathcal{F}_{\mathsf{nn}}$ with perturbation less than ε.
2. We prove that a good output of an importance sampling procedure has the recovery guarantee.
3. We prove that random i.d.d. sampling from an unknown distribution D is an importance sampling procedure. The result says that a large explicit dataset can replace the unknown implicit distribution when we try to obtain the optimal predictor. The result provides the minimum size of the dataset but does not provide the minimum number of labels.
4. We prove that our randomized sampling for the ρ-nearly orthonormal basis is an importance sampling procedure. As a result, the recovery guarantee holds for samples from a known dataset with only a limited of data labeled.

5. Lastly, we combine the results of step 1, step 3, and step 4 using triangle inequality. This finishes the proof of the equivalence between optimal predictor trained by unlimited labeled data and optimal predictor trained by data that select to label by our proposed algorithm.

5.3 ρ-NEARLY ORTHONORMAL BASIS

We define basis of a function family $\mathcal{F}$ to be a set of function $\mathcal{V} = \{v_1(x), \ldots, v_{|\mathcal{V}|}(x)\}$ such that $v_i : \mathbb{R}^d \to \mathbb{R},\ i \in [|\mathcal{V}|]$ and for any function $f \in \mathcal{F}$, there exists $\alpha_1, \ldots, \alpha_{|\mathcal{V}|} \in \mathbb{R}$ holds that

$$f = \alpha_1 v_1 + \ldots + \alpha_{|\mathcal{V}|} v_{|\mathcal{V}|}.$$

An orthonormal basis for distribution D is a basis of function family $\mathcal{F}$ with

$$\mathop{\mathbb{E}}_{x\sim D}[v_i(x) \cdot v_j(x)] = 1_{i=j}.$$

For linear functions such an orthonormal basis always exists for different distributions. However, it's hard to decompose a non-linear function into the summation of such a base. Finding an orthonormal basis for non-linear functions is even harder. This fact motivates us to propose a generalized concept for an orthonormal basis to take the advantage of the orthonormal basis.

Now, we define the ρ-nearly orthonormal basis as follows. We show such a ρ-nearly orthonormal basis exists for neural network f_{nn}. We motivate this definition from Cheap Kabatjanskii-Levenstein bound (Tao, 2019) and Johnson–Lindenstrauss lemma.

Definition 5.1 (ρ-nearly orthonormal basis)**.** *Given the distribution D and desired accuracy ε_0, a set of function $\{v_1(x), \ldots, v_{\overline{d}(x)}\}$ forms a fixed ρ-nearly orthonormal basis of neural network f_{nn} when the inner products taken under the distribution D such that*

$$\mathop{\mathbb{E}}_{x\sim D}[v_i(x) \cdot v_j(x)] = 1, \forall i = j \in [\overline{d}]$$
$$\left|\mathop{\mathbb{E}}_{x\sim D}[v_i(x) \cdot v_j(x)]\right| \leq \rho, \forall i \neq j \in [\overline{d}]$$

Furthermore, let the basis forms function family $\mathcal{F}_{\mathsf{nn}}$, for any weight $W \in \mathbb{R}^{d\times m}$, there exist function $h \in \mathcal{F}_{\mathsf{nn}}$ and $\alpha(h) := (\alpha(h)_1, \ldots, \alpha(h)_{\overline{d}})$ under the basis $(v_1, \ldots, v_{\overline{d}})$ such that

$$h(x) = \sum_{i=1}^{\overline{d}} \alpha(h)_i \cdot v_i(x) \quad \textit{and} \quad |h(x) - f_{\mathsf{nn}}(W, x)| \leq \varepsilon_0.$$

Remark 5.2. *Note that for linear function family $\mathcal{F}$, we know that $\overline{d} = d$ and $\rho = 0$. However, for the neural network function family $\mathcal{F}_{\mathsf{nn}}$, we will have $\overline{d} \gg d$ and ρ is not necessary to be 0.*

Note that this definition relies on the distribution D. So, the distribution should be known when we practically compute orthonormal basis. We would not utilize the orthonormal basis for unknown distribution D in our algorithm.

In Definition 5.1, we state that ρ-nearly orthonormal basis always exist for neural network f_{nn}. We will show the mentioned correctness here.

Claim 5.3. *Let ρ-nearly orthonormal basis be defined as in Definition 5.1. There exists $\{v_1, \ldots, v_{\overline{d}}\}$ which forms a fixed ρ-nearly orthonormal basis for neural network f_{nn}. Furthermore, let $\mathcal{F}_{\mathsf{nn}}$ be the function family formed by the ρ-nearly orthonormal basis, then for any $W \in \mathbb{R}^{d\times m}$, there exists $h \in \mathcal{F}_{\mathsf{nn}}$ such that*

$$\|h(x) - f_{\mathsf{nn}}(W, x)\|_D^2 \leq \varepsilon.$$

Besides, for any $h \in \mathcal{F}_{\mathsf{nn}}$, there exists $W \in \mathbb{R}^{d\times m}$ such that

$$\|h(x) - f_{\mathsf{nn}}(W, x)\|_D^2 \leq \varepsilon.$$

5.4 IMPORTANCE SAMPLING PROCEDURE

In this section, we first propose α-condition number. Then, based on α-condition number, we propose an importance sampling procedure and its corresponding good output. Finally, we prove that good output of the importance sampling procedure is suffice to recover the optimal predictor.

In our algorithm, we utilize important sampling trick and we can estimate the properties of an desired distribution by sampling in an unknown distribution as shown below

$$\mathop{\mathbb{E}}_{x\sim D'}[\frac{D(x)}{D'(x)}h(x)] = \int D'(x)\frac{D(x)}{D'(x)}h(x)\mathrm{d}x = \mathop{\mathbb{E}}_{x\sim D}[h(x)].$$

As a result, we generalize the definition in Definition 3.3 and propose α-condition number.

Definition 5.4 (α-Condition Number)**.** *For any distribution D' over the domain $\mathbb{R}^d$ and any function $h : \mathbb{R}^d \to \mathbb{R}$. Let the ρ-nearly orthonormal basis $\{v_1, \ldots, v_{\overline{d}}\}$ and corresponding decomposition coefficient $\alpha(h)$ be defined as in Definition 5.1. When the distribution D is clear, we use $K_{\alpha,D'}$ to denote the α-condition number of sampling from D', i.e.,*

$$K_{\alpha,D'} = \sup_x \left\{ \frac{D(x)}{D'(x)} \cdot \sup_{h\in\mathcal{F}_{\mathsf{nn}}} \{ \frac{|h(x)|^2}{\|\alpha(h)\|_2^2} \} \right\}.$$

Note that when $\rho = 0$, we have $\|\alpha(h)\|_2^2 = \|h\|_D^2$ and

$$\mathop{\mathbb{E}}_{x\sim D'}[\frac{D(x)}{D'(x)} \cdot \frac{|h(x)|^2}{\|\alpha(h)\|_2^2}] = \mathop{\mathbb{E}}_{x\sim D}[\frac{|h(x)|^2}{\|\alpha(h)\|_2^2}] = \mathop{\mathbb{E}}_{x\sim D}[\frac{|h(x)|^2}{\|h\|_D^2}] = 1,$$

which indicates that $K_{\alpha,D'}$ shows the concentration of the weighted term.

Next, we introduce the importance sampling procedure. The importance sampling procedure helps us find an important property of i.d.d. samples and provide us an effective way to find an available query algorithm. We show our definition as follows:

Definition 5.5 (Importance Sampling Procedure)**.** *Given $\mathcal{F}_{\mathsf{nn}}$ and underlying distribution D, let P be a random sampling procedure depend on D and $\mathcal{F}_{\mathsf{nn}}$. Suppose P terminates in k iterations (k is not necessarily fixed) and generates a coefficient β_i and a distribution D_i to sample $x_i \sim D_i$ in every iteration $i \in [k]$. We say P is an* ε-importance sampling procedure *if it satisfies the following two properties:*

1. *Let $v_1, \ldots, v_{\overline{d}}$ of $\mathcal{F}_{\mathsf{nn}}$ under D be defined as in Definition 5.1. Let weight $w_i = \beta_i \cdot D(x_i)/D_i(x_i)$ for each $i \in [k]$. With probability 0.9, the matrix $A(i,j) = \sqrt{u_i} \cdot v_j(x_i) \in \mathbb{R}^{k\times\overline{d}}$ has $\lambda(A^*A) \in [\frac{3}{4}, \frac{5}{4}]$.*
2. *The coefficients always have $\sum_{i=1}^k \beta_i \le \frac{5}{4}$ and $\beta_i \cdot K_{\alpha,D_i} \le \epsilon/2$, $\forall i \in [k]$.*

The definition of an importance sampling procedure consists two parts. First, it claims a bound for eigenvalue of a specific matrix. To give some intuition for it, we claim that if both ε_0 and ρ defined in Definition 5.1 equals 0, then the first property is equivalent to

$$\sup_{h\in\mathcal{F}} \frac{\sum_{i=1}^k u_i \cdot |h(x_i)|^2}{\|h\|_D^2} \in [\frac{3}{4}, \frac{5}{4}]$$

This indicates that the sampling procedure preserves the mass of the signal. The second property provides the necessary bound on the coefficient β_i. It helps us bound perturbation $\|f_{\mathsf{nn}}(\widetilde{W}, x) - f_{\mathsf{nn}}(W^*, x)\|_D^2$ in Theorem 4.1.

It's clear that the first property of Definition 5.5 is not necessary to be satisfied. We define good output of an importance sampling procedure when the first property is satisfied.

Definition 5.6 (Good Output)**.** *Given an* importance sampling procedure *P, we say the output of P is* good *only if the samples x_i with weights $u_i = \beta_i \cdot D(x_i)/D_i(x_i)$ satisfy the first property in Definition C.4. Given a joint distribution (D, Y) and an execution of an* importance sampling procedure *P with $x_i \sim D_i$ and $u_i = \beta_i \cdot D(x_i)/D_i(x_i)$ of each $i \in [k]$. Querying $y_i \sim (Y|x_i)$ for each point x_i. We define $\widetilde{f}$ as follows:*

$$\widetilde{f} := \mathop{\arg\min}_{h\in\mathcal{F}_{\mathsf{nn}}} \sum_{i=1}^k u_i \cdot |h(x_i) - y_i|^2. \tag{3}$$

The next theorem shows that a good output of an ε-importance sampling procedure suffice for the recovery of the optimal predictor.

Theorem 5.7 (Importance Sampling Case)**.** *Given a neural network function family $\mathcal{F}_{\mathsf{nn}}$, joint distribution (D, Y), and $\varepsilon \in (0, 1)$, let P be an ε-importance sampling procedure for $\mathcal{F}_{\mathsf{nn}}$ and D, and we define f as follows: $f := \mathop{\arg\min}_{h\in\mathcal{F}_{\mathsf{nn}}} \mathop{\mathbb{E}}_{(x,y)\sim(D,Y)}[|y - h(x)|^2]$. Let P' be a good output of P. Let $\widetilde{f}$ be define as Eq. (3). Then $\widetilde{f}$ of P' satisfies*

$$\mathop{\mathbb{E}}_{P'}[\|f - \widetilde{f}\|_D^2] \le \epsilon \cdot \mathop{\mathbb{E}}_{(x,y)\sim(D,Y)}[|y - f(x)|^2].$$

Note that the inequality in Theorem 5.7 is exactly the inequality in Theorem 4.1. So, if we can check a procedure is an importance sampling produce, we can directly apply the Theorem 5.7 and obtain an bound on the number of required labeled data.

5.5 Sample Algorithm for Known Distribution

Definition 5.5 and Theorem 5.7 shows that we may acquire better result when $D_i,\ i \in [k]$ are not set to D and $\beta_i,\ i \in [k]$ are not equal. The result in this section allows our algorithm to label less number of samples.

Lemma 5.8 (Importance Sampling Procedure for Known Distribution)**.** *Given any dimension $\overline{d}$ linear space $\mathcal{F}_{\mathsf{nn}}$, any distribution D over the domain of $\mathcal{F}_{\mathsf{nn}}$, and any $\varepsilon \in (0,1)$, there exists an* ε-importance sampling procedure *that terminates in $O(\overline{d}/\varepsilon)$ rounds with arbitrarily large constant probability.*

The process claimed in this lemma is based on randomized BSS (Lee & Sun, 2018). BSS (Batson et al., 2012) is first proposed as sparsifiers of arbitrary graphs. Later, (Lee & Sun, 2018) developed randomized BSS to construct linear-sized spectral sparsification for graphs. (Chen & Price, 2019) utilized it for subsample linear regression problem and Fourier-sparse signal recovery. We expand it into our non-linear active learning cases. Randomized BSS is an iterative process that defines a potential function for matrices. During the iteration process, the potential function is non-increasing. We will use this property to construct our distribution D_i and coefficient β_i. This process is too complex and we put the details into Appendix.

5.6 I.I.D. Distributions

A specific case of Definition 5.5 will occur when we set $D_i \leftarrow D$ and $\beta_i \leftarrow 1/k$. It is exactly the process when we obtain the unlabeled dataset $X = [x_1, \ldots, x_{k_0}]$. We will show in this section that it is an importance sampling procedure when k_0 is sufficiently large. As a result, the optimal predictor trained by k_0 labeled samples can recovery the theoretical optimal predictor trained with unlimited samples and labels.

Although results in Lemma 5.8 show that labeling all the data in a dataset is not a clever choice, this result provides us a theoretical tool to bridge the unknown distribution D and the requirement of explicit distribution in Definition 5.5.

Lemma 5.9. *Given any distribution D' with the same support of D and any $\epsilon \in (0,1)$, the random sampling procedure with $k = \Theta(K_{\alpha,D'} \log(\overline{d}) + \varepsilon^{-1} K_{\alpha,D'})$ i.i.d. random samples from D' and coefficients $\beta_i = 1/k$, $\forall i \in [k]$ is an* ε-importance sampling procedure.

The result provides the minimum size of the dataset but does not provide the minimum number of labels. This Lemma states that the recovery guarantee holds for a sufficiently large dataset that all the samples in it are labeled. This indicates that a large explicit dataset can replace the unknown implicit distribution when we try to obtain the optimal predictor. Combining with the original result in Lemma 5.8 where D_i and β_i are different, Lemma 5.9 can indicate that acquiring less label is always possible and labeling all the samples in a dataset will not necessarily improve the prediction result.

6 Conclusion

It is well-known that deep learning requires large numbers of labeled training data to deliver state-of-the-art results. However, in many domains of AI, abundant unlabeled data are available but it is expensive to acquire data labels. This is especially important in domains where only experts have the ability to label data, e.g., diagnosing patients using their x-rays. In these AI domains, active learning can potentially substantially reduce the costs of AI by reducing the labeling costs. An active learning algorithm uses unlabeled data and selects a fraction of them for an oracle to label. The algorithm then uses the labels to generate an accurate model for an AI task. The goal of active learning is to maximize model accuracy while maintaining a low sample complexity, i.e., the number of data for the oracle to label. From a theoretical perspective, it is interesting to know whether it is possible to understand how many labeled data are actually needed to train a machine learning model. This problem has been extensively explored for training linear machine learning models (e.g., linear regression). Today, deep learning has become the de facto method for machine learning, but the sample complexity problem for deep active learning is still unsolved. This problem is challenging because neural networks are inherently non-linear. We present the first deep active learning algorithm which has a provable sample complexity. Using this algorithm, we have derived the first upper bound on the sample complexity of deep active learning. Our upper bound shows that the minimum number of labeled data does not depend on the data distribution or the width of the neural network, but it is determined by the smoothness of non-linear activation and the dimension of the input data.

ETHICS STATEMENT

This work does not raise any ethical issues.

REPRODUCABILITY STATEMENT

Our paper is a theoretical work. We explicitly stated all the assumptions we made and provided complete proofs in supplementary materials.

REFERENCES

Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3672–3680, 2019.

Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 237–245, 2015.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. In *NeurIPS*, 2019a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *ICML*, 2019b.

Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9368–9377, 2018.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pp. 153–160, 2007.

Mustafa Bilgic and Lise Getoor. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, volume 4, 2009.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal coresets for least-squares regression. *IEEE transactions on information theory*, 59(10):6880–6892, 2013.

Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (overparametrized) neural networks in near-linear time. In *ITCS*, 2021.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Kamalika Chaudhuri, Sham Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. *arXiv preprint arXiv:1506.02348*, 2015.

Xue Chen and Eric Price. Active regression via linear-sample sparsification. In *Conference on Learning Theory (COLT)*, pp. 663–695. PMLR, 2019.

Albert Cohen, Mark A Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of computational mathematics*, 13(5):819–834, 2013.

Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pp. 150–157. Elsevier, 1995.

Michał Dereziński and Manfred K Warmuth. Unbiased estimates for linear regression via volume sampling. *arXiv preprint arXiv:1705.06908*, 2017.

Michal Derezinski, Manfred K Warmuth, and Daniel Hsu. Tail bounds for volume sampled linear regression. *arXiv preprint arXiv:1802.06749*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*. arXiv preprint arXiv:1810.02054, 2019b.

Xuefeng Du, Dexing Zhong, and Huikai Shao. Building an active palmprint recognition system. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1685–1689. IEEE, 2019c.

Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. Deep active learning for efficient training of a lidar 3d object detector. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 667–674. IEEE, 2019.

Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision*, pp. 562–577. Springer, 2014.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. Deep active learning for biased datasets via fisher kernel self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9041–9049, 2020.

Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, pp. 802–810, 2010.

HM Sajjad Hossain and Nirmalya Roy. Active deep learning for activity recognition with context aware annotator selection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1862–1870, 2019.

Daniel Hsu and Sivan Sabato. Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582, 2016.

Yue Huang, Zhenwei Liu, Minghui Jiang, Xian Yu, and Xinghao Ding. Cost-effective vehicle type recognition in surveillance images with deep active learning and web data. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):79–86, 2019.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2372–2379. IEEE, 2009.

Ross D King, Kenneth E Whelan, Ffion M Jones, Philip GK Reiser, Christopher H Bryant, Stephen H Muggleton, Douglas B Kell, and Stephen G Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32:7026–7037, 2019.

Vikram Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Jason D Lee, Ruoqi Shen, Zhao Song, Mengdi Wang, and Zheng Yu. Generalized leverage score sampling for neural networks. In *NeurIPS*, 2020.

Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *SIAM Journal on Computing*, 47(6):2315–2336, 2018.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pp. 3–12. Springer, 1994.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NeurIPS*, 2018.

Malik Magdon-Ismail. Row sampling for matrix algorithms via a non-commutative bernstein bound. *arXiv preprint arXiv:1008.0587*, 2010.

Michael W Mahoney. Randomized algorithms for matrices and data. *arXiv preprint arXiv:1104.5557*, 2011.

Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 79, 2004.

Zhenshen Qu, Jingda Du, Yong Cao, Qiuyu Guan, and Pengbo Zhao. Deep active learning for remote sensing object detection. *arXiv preprint arXiv:2003.08793*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep active learning for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3934–3938. IEEE, 2017.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*, 2020.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, 2:441–448, 2001.

Sivan Sabato and Remi Munos. Active regression by stratification. *Advances in Neural Information Processing Systems*, 27:469–477, 2014.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Burr Settles. Active learning literature survey. 2009.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. *Advances in neural information processing systems*, 20:1289–1296, 2007.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017.

Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pp. 1308–1318. PMLR, 2020.

Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 1220–1227. IEEE, 2021.

Zhao Song and Xin Yang. Quadratic suffices for over-parametrization via matrix chernoff bound. In *arXiv preprint*. https://arxiv.org/pdf/1906.03593.pdf, 2019.

Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2772–2789. SIAM, 2019.

Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 81–90. https://arxiv.org/abs/cs/0310051, divided into https://arxiv.org/abs/0809.3232, https://arxiv.org/abs/0808.4134, https://arxiv.org/abs/cs/0607105, 2004.

Terence Tao. A cheap version of the kabatjanskii-levenstein bound for almost orthogonal vectors, 2019.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12 (4):389–434, 2012.

David P Woodruff. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.

Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. Deep similarity-based batch mode active learning with exploration-exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 575–584. IEEE, 2017.

Ye Zhang, Matthew Lease, and Byron Wallace. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Fedor Zhdanov. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

Shusen Zhou, Qingcai Chen, and Xiaolong Wang. Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, 120:536–546, 2013.