



Original Article

# Lipschitzness effect of a loss function on generalization performance of deep neural networks trained by Adam and AdamW optimizers

Mohammad Lashkari<sup>\*a</sup>, Amin Gheibi<sup>a</sup>

<sup>a</sup>Department of Mathematics and Computer Science, Amirkabir University of Technology (Tehran Polytechnic), Iran

**ABSTRACT:** The generalization performance of deep neural networks with regard to the optimization algorithm is one of the major concerns in machine learning. This performance can be affected by various factors. In this paper, we theoretically prove that the Lipschitz constant of a loss function is an important factor to diminish the generalization error of the output model obtained by Adam or AdamW. The results can be used as a guideline for choosing the loss function when the optimization algorithm is Adam or AdamW. In addition, to evaluate the theoretical bound in a practical setting, we choose the human age estimation problem in computer vision. For assessing the generalization better, the training and test datasets are drawn from different distributions. Our experimental evaluation shows that the loss function with a lower Lipschitz constant and maximum value improves the generalization of the model trained by Adam or AdamW.

**Review History:**

Received:08 February 2023  
Revised:12 July 2023  
Accepted:20 August 2023  
Available Online:01 October 2024

**Keywords:**

Generalization error  
Adam algorithm  
Lipschitz constant

**MSC (2020):**

68T05; 68T45

## 1. Introduction

The adaptive moment estimation (Adam) algorithm is one of the most widely used optimizers for training deep learning models. Adam is an efficient algorithm for stochastic optimization, based on adaptive estimates of first-order and second-order moments of gradient [17]. The method is computationally efficient and has little memory usage. Adam is much more stable than stochastic gradient descent (SGD) and the experiments of work [17] show that it is faster than previous stabilized versions of SGD, such as SGD Nesterov [14], RMSProp [28] and AdaGrad [8] to minimize the loss function in the training phase. It is recently used in several machine learning problems and performs well. Thus, any improvement in the generalization performance of a model trained by Adam is essential.

One of the main concerns in machine learning is the generalization performance of deep neural networks (DNNs). A generalization measurement criterion is the generalization error which is defined as the difference between the true risk and the empirical risk of the output model [1]. One established way to address the generalization error of machine learning models in order to derive an upper bound for it, is the notion of uniform stability [1, 4, 12]. Roughly speaking, the uniform stability measures the difference in the error of the output model caused by a slight change in the training set. The pioneering work of [4], shows that if a deterministic learning algorithm is more

<sup>\*</sup>Corresponding author.

E-mail addresses: mohammad.l@aut.ac.ir, amin.gheibi@aut.ac.ir



stable, then the generalization error of the ultimate model achieves a tighter upper bound. In the following work of [12], Hardt *et al.* extend the notion of uniform stability to randomized learning algorithms to drive an upper bound for the expected generalization error of a DNN trained by SGD. They prove that SGD is more stable, provided that the number of iterations is sufficiently small. In the recent work of [1], Ali Akbari *et al.* derive a high probability generalization error bound instead of an expected generalization error bound. They demonstrate that if SGD is more uniformly stable, then the generalization error bound is tighter. They also proved the direct relationship between the uniform stability of SGD and loss function properties i.e. its Lipschitzness, resulting in the generalization error connection to the Lipschitz constant and the maximum value of a loss function.

In this paper, we make the following contributions. First, we distinguish the relationship between the uniform stability of Adam and Lipschitzness of a loss function. Second, we derive an upper bound for the generalization error of a DNN trained by Adam which is directly related to the Lipschitz constant and the maximum value of a loss function. Third, we assess the uniform stability of AdamW optimizer which decouples weight decay from estimates of moments to make the regularization technique more effective [18]. Fourth, we connect the generalization error of a DNN trained by AdamW to Lipschitzness and the maximum value of a loss function. In Experiments, we evaluate our theoretical results in the human age estimation problem.

Human age estimation is one of the most significant topics in a wide variety of applications such as age-specific advertising, customer profiling, or recommending new things. However, we are facing many challenges to solve this problem. Face makeup, insufficient light, skin color, and unique features of each person are the factors that can affect the accuracy of the model. Based on these reasons, collecting more data cannot necessarily reduce the generalization error of the final models in this problem. Hence, it is a de facto example of label distribution learning where we can evaluate our theoretical bounds precisely as the authors of [1] did because it is a challenging problem, and if our theoretical analyzes are correct and advantageous, then we can enhance age estimation models. Furthermore, there are available age estimation datasets drawn from different distributions which gives us the ability to assess the generalization error of the models well. The practical results show that choosing a stable loss function based on our theoretical bounds, can improve the accuracy and enhance the generalization performance of a model trained by Adam or AdamW.

## 2. Related Work

There is a variety of approaches to derive upper bounds for the generalization error including algorithmic stability [1, 2, 4, 12, 16, 26], robustness [23, 29], PAC-Bayesian Theory [10, 21], and Vapnik-Chervonenkis (VC) dimension [3, 13, 25]. Each of these approaches theoretically analyzes some effective factors and gives the researchers some information which can enhance the generalization performance of deep learning models.

In the robustness theory, the generalization performance of models is measured according to the structure of models and how to use training data. In the work of [23], a meta-learning algorithm was designed to learn what weights assign to training samples based on their gradient direction to improve the generalization ability. The algorithm can be applied to any type of neural network; it does not need hyper-parameter tuning and has good accuracy on unbalanced data. In the work of [29], Zahavy *et al.* defined a new measure called ensemble robustness, which measures the robustness of several hypotheses. Leveraging it, they demonstrated that a deep learning model can generalize well if its variance is controlled.

The PAC-Bayesian theory is an approach to analyze Bayesian learning algorithms in which a prior distribution is considered on the hypothesis space and the output of the algorithm is also a distribution over this space. In the work of [21], an upper bound in terms of the product of the spectral norm of the layers and the Frobenius norm of the weights was derived for the generalization error of a neural network. Guan and Zhiwu in [11] generalized two generalization bounds i.e. kl-bound and Catoni-bound to meta-learning framework and proposed two classification algorithms with fast convergence rates by minimizing the aforementioned upper bounds.

The VC-dimension is a measure of complexity, flexibility, and generalization of classification models created based on feed-forward neural networks and recurrent neural networks (RNNs). In the work of [25], this concept was extended to graph neural networks (GNNs) and recursive neural networks (RecNNs). The main finding of the work is that the upper bound for the VC-dimension of GNNs and RecNNs is comparable with the VC-dimension upper bound of RNNs. Also, the generalization ability of these models is directly related to the number of connected nodes [25]. The authors of [3], derived an upper bound for the VC-dimension of convolutional neural networks, resulting in reducing the error of text classification models and increasing their generalization ability. In the work of [13], an upper bound and a lower bound for the VC-dimension of neural networks with activation function *ReLU* are derived that both are polynomial in terms of the number of layers and the number of weights. All the results of the work can be generalized to any model with arbitrary piece-wise linear activation functions.

We follow the notion of uniform stability in this paper. This notion was firstly introduced in [4] for deterministic

algorithms. It was extended to randomized algorithms in the work of [12] to derive an expected upper bound for the generalization error which is directly related to the number of training epochs of SGD.

Recently, based on the uniform stability definition of SGD, the generalization error of a DNN trained by it, with high probability, is upper-bounded by a vanishing function which is directly related to the Lipschitz constant and the maximum value of a loss function [1]. In our work, we analyze the uniform stability for Adam and AdamW and its relationship with the Lipschitz constant of a loss function. Then, we demonstrate how the characteristics of a loss function i.e. its Lipschitz constant and maximum value can be effective on the generalization performance of a DNN. We show that the loss function proposed in [1] to stabilize the training process when the optimizer is SGD, can also stabilize the training process and reduces the generalization error when the optimizer is Adam or AdamW. Other researchers can use our theoretical results to find new loss functions for training DNNs by Adam or AdamW.

### 3. Preliminaries

Let  $X$  and  $Y \subseteq \mathbb{R}^M$  be the input and output spaces of a problem respectively, and  $F$  be the set of all mappings from  $X$  to  $Y$ . A learning problem is to find  $f^\theta : X \rightarrow Y$ , parameterized by  $\theta \in H$  where  $H \subset \mathbb{R}^K$  is a bounded set, containing all possible values for the neural network parameters <sup>1</sup>. Assume  $\ell : Y \times Y \rightarrow \mathbb{R}^+$  denotes the loss function of the problem. The goal of a learning algorithm is to minimize the true risk  $R_{true}(f^\theta) := \mathbb{E}_{(x,y)} [\ell(f^\theta(x), y)]$  where  $(x, y) \in X \times Y$ :

$$f_{true}^\theta = \underset{f^\theta \in F}{\operatorname{argmin}} R_{true}(f^\theta). \tag{1}$$

Since the distribution of  $X \times Y$  is unknown;  $f_{true}^\theta$  cannot be found in the equation (1). Hence, we have to estimate the true risk. Let  $S \in (X \times Y)^N$  be the training set. The true risk is estimated by the empirical risk  $R_{emp}(f^\theta) := \frac{1}{N} \sum_{i=1}^N \ell(f^\theta(x_i), y_i)$  in which,  $N = |S|$  and  $(x_i, y_i) \in S$ . In current deep learning algorithms, training the model means minimizing  $R_{emp}(f^\theta)$ . In the rest of this paper, in the theorems and proofs, the loss function is denoted by  $\ell(\hat{y}, y)$  where  $\hat{y}$  is the predicted vector and  $y$  is the target vector.

**Definition 3.1 (Partition).** Suppose that  $S$  is a training set of size  $N$ . Let  $1 < k < N$  be a number that  $N$  is divisible by  $k$  (if it is not possible, we repeat a sample enough to make divisibility possible). A partition of  $S$ , which we denote by  $B_S = \{B_1, B_2, \dots, B_k\}$ , is a set of  $k$  subsets of  $S$  such that every sample is in exactly one of these subsets and the size of each subset is  $\frac{N}{k}$ .

We use Definition 3.1 to formalize the training process of deep learning models mathematically. Assume  $S$  is the training set and  $B_S = \{B_1, B_2, \dots, B_k\}$  is a partition of it. Each element of  $B_S$  represents a mini-batch of  $S$ . Without loss of generality we suppose that in each iteration of the optimization algorithm, a mini-batch  $B_i \in B_S$  is randomly selected to the parameters be updated <sup>2</sup>. This is done by the algorithm using a random sequence  $R = (r_1, r_2, \dots, r_T)$  of indices of elements in  $B_S$ , where  $T$  is the number of iterations. We use  $f_{B_S, R}^\theta$  to denote the output model of the optimization algorithm, applied to a partition  $B_S$  and a random sequence  $R$ .

**Definition 3.2 (Generalization Error).** Given a partition  $B_S$  of a training set  $S$  and a sequence  $R$  of random indices of  $B_S$  elements, the generalization error of  $f_{B_S, R}^\theta$  trained by an arbitrary optimization algorithm, is defined as  $E(f_{B_S, R}^\theta) = R_{true}(f_{B_S, R}^\theta) - R_{emp}(f_{B_S, R}^\theta)$ .

**Definition 3.3 (Lipschitzness).** Let  $Y \subseteq \mathbb{R}^M$  be the output space of a problem. A loss function  $\ell(\hat{y}, y)$  is  $\gamma$ -Lipschitz with regard to its first argument, if  $\forall y_1, y_2 \in Y$ , we have:

$$|\ell(y_1, y) - \ell(y_2, y)| \leq \gamma \|y_1 - y_2\|,$$

where  $\|\cdot\|$  is the  $L_2$  norm.

As mentioned before, uniform stability of the optimization algorithm is effective on the generalization performance of the ultimate model  $f_{B_S, R}^\theta$  [1]. We follow the uniform stability definition of work [12] to link Lipschitzness of the loss function to the generalization error of  $f_{B_S, R}^\theta$ . For simplicity, moving forward, we denote  $f_{B_S, R}^\theta$  by  $f_{B_S, R}$  and  $E(f_{B_S, R}^\theta)$  by  $E(f_{B_S, R})$ .

<sup>1</sup>We know the number of iterations in the training phase is finite. Therefore the set of visible values for parameters in the training stage is finite. So we can assume the set of all possible values is an infinite bounded superset of visible values. We will the reason in Subsection 5.2

<sup>2</sup>The index of the mini-batch is randomly selected from  $\{1, 2, \dots, k\}$ .

Along with the notion of uniform stability which we define in Section 5, another concept called bounded difference condition (BDC) affects the generalization error [1]:

**Definition 3.4 (BDC).** Consider two numbers  $k, T \in \mathbb{N}$ . If  $G : \{1, 2, \dots, k\}^T \rightarrow \mathbb{R}^+$ , is a measurable function and for  $R, R' \in \text{Dom}(G)$  which are different only in two elements, constant  $\rho$  exists such that

$$\sup_{R, R'} |G(R') - G(R)| \leq \rho,$$

then,  $G(\cdot)$  holds bounded difference condition (BDC) with the constant  $\rho$ . We use the  $\rho$ -BDC expression to denote that a function holds this condition with the constant  $\rho$ .

In Definition 3.4, we assumed the slight change in the input to be the difference in two elements, which we will see its reason in the proof of the theorems. Intuitively, if a function satisfies the above condition, its value does not differ much due to a slight change in the input. Such functions are dense around their expectation with respect to the input random sequence  $R$  [19].

#### 4. Formulation of Age Estimation Problem

Our problem in the experimental part is human age estimation. Let  $(x, y)$  be a training sample where  $x$  is the input image of a person’s face and  $y \in \mathbb{N}$  is the corresponding age label. Due to the correlation of the neighboring ages, classification methods based on single-label learning [24] are not efficient because these methods ignore this correlation. Also, regression-based models are not stable to solve this problem [1].

According to the aforementioned reasons, another method based on label distribution learning (LDL) framework which was firstly introduced in the work of [9], is used for this problem [1]. In this method  $y$  is replaced by  $y = [y_1, y_2, \dots, y_M] \in \mathbb{R}^M$  where  $y_i$  is the probability of facial image  $x$  belongs to class  $i$ . As usual,  $y$  is assumed to be a normal distribution, centering at  $y$  and standard deviation  $\sigma$  which controls the spread of the distribution [9]. Therefore, the output space,  $Y$  is a subset of  $\mathbb{R}^M$  and our objective is to find  $f^\theta$  which maps  $x$  to  $y \in Y$ .

##### 4.1. Loss Functions for Age Estimation Problem

Let  $(x, y) \in S$  be a training instance where  $x$  represents the facial image and  $y \in \mathbb{R}^M$  is the corresponding label distribution. Consider  $\hat{y} = f^\theta(x)$ , representing the estimated label distribution by  $f^\theta$ . To obtain  $f^\theta$ , a convex loss function named Kullback-Leibler (KL) divergence has been widely utilized. The KL loss function is defined as below:

$$\ell_{KL}(\hat{y}, y) = \sum_{m=1}^M y_m \log\left(\frac{y_m}{\hat{y}_m}\right).$$

As an alternative to KL, another convex loss function called Generalized Jeffries-Matusita (GJM) distance has been proposed in [1] under the LDL framework, defined as

$$\ell_{GJM}(\hat{y}, y) = \sum_{m=1}^M y_m \left| 1 - \left(\frac{\hat{y}_m}{y_m}\right)^\alpha \right|^{\frac{1}{\alpha}},$$

where  $\alpha \in (0, 1]$ . According to the experiments of [1], the best value of  $\alpha$  for good generalization is 0.5. It has been proved that if  $\alpha = 0.5$ , then the Lipschitz constant and the maximum value of GJM are less than the Lipschitz constant and the maximum value of KL respectively <sup>3</sup> [1].

#### 5. Uniform Stability and Generalization Error Analysis

The notion of uniform stability was firstly introduced in [4] for deterministic learning algorithms. They demonstrate that smaller stability measure of the learning algorithm, the tighter generalization error is. However, their stability measure is limited to deterministic algorithms and is not appropriate for randomized learning algorithms such as Adam. Therefore, we follow [1, 12] to define the uniform stability measure for randomized optimization algorithms generally:

---

<sup>3</sup>It should be mentioned here that KL is not Lipschitz because when  $x \rightarrow 0$  the derivative of  $\log(x)$  tends to infinity. So we have to bound its domain from left e.g.  $(10^{-10}, +\infty)$  to make it Lipschitz. In contrast to KL, GJM does not have this issue.

**Definition 5.1 (Uniform Stability).** Let  $S$  and  $S'$  denote two training sets drawn from a distribution  $\mathbb{P}$ . Suppose that  $B_S$  and  $B_{S'}$  of equal size  $k$ , are two partitions of  $S$  and  $S'$  respectively, which are different in only one element (mini-batch). Consider a random sequence  $R$  of  $\{1, 2, \dots, k\}$  to select a mini-batch at each iteration of an optimization algorithm,  $A_{opt}$ . If  $f_{B_S, R}$  and  $f_{B_{S'}, R}$  are output models obtained by  $A_{opt}$  with the same initialization, then  $A_{opt}$  is  $\beta$ -uniformly stable with regard to a loss function  $\ell$ , if

$$\forall S, S' \sup_{(x, y)} \mathbb{E}_R [|\ell(f_{B_{S'}, R}(x), y) - \ell(f_{B_S, R}(x), y)|] \leq \beta.$$

To evaluate the uniform stability of Adam and AdamW in order to prove its link to loss function properties, a lemma named **Growth recursion** which has been stated in [12] for SGD is central to our analysis. In the following, we state this lemma for an arbitrary iterative optimization algorithm, but before stating the lemma, we need some definitions. As we know, gradient-based optimization algorithms are iterative, and in each iteration, the network parameters are updated. Let  $H$  be the set of all possible values for the neural network parameters. Let  $A_{opt}$  be an arbitrary iterative optimization algorithm that runs  $T$  iterations. In the  $t$ -th iteration, the update that is computed in the last command of the loop for the network parameters, is a function  $A^t : H \rightarrow H$  mapping  $\theta_{t-1}$  to  $\theta_t$  for each  $1 \leq t \leq T$ . We call  $A^t$  the **update rule** of  $A_{opt}$ . Let's define two characteristics of an update rule: The update rule,  $A^t(\cdot)$  is  $\sigma$ -bounded if

$$\sup_{\theta \in H} \|\theta - A^t(\theta)\| \leq \sigma, \tag{2}$$

and it is  $\tau$ -expensive if

$$\sup_{\theta, \theta' \in H} \frac{\|A^t(\theta) - A^t(\theta')\|}{\|\theta - \theta'\|} \leq \tau, \tag{3}$$

where  $\|\cdot\|$  is the  $L_2$  norm.

**Lemma 5.2 (Growth recursion).** [12] Given two training set  $S$  and  $S'$ , suppose that  $\theta_0, \theta_1, \dots, \theta_T$  and  $\theta'_0, \theta'_1, \dots, \theta'_T$  are two updates of network parameters with update rules  $A_S^t$  and  $A_{S'}^t$ , running on  $S$  and  $S'$  respectively such that for each  $1 \leq t \leq T$ ,  $\theta_t = A_S^t(\theta_{t-1})$  and  $\theta'_t = A_{S'}^t(\theta'_{t-1})$ . For  $\Delta_t = \|\theta_t - \theta'_t\|$ , we have:

- If  $A_S^t$  and  $A_{S'}^t$  are equal and  $\tau$ -expensive, then  $\Delta_t \leq \tau \Delta_{t-1}$ .
- If  $A_S^t$  and  $A_{S'}^t$  are  $\sigma$ -bounded, then  $\Delta_t \leq \Delta_{t-1} + 2\sigma$  <sup>4</sup>.

We state the proof of Lemma 5.2 in Appendix A. In Subsection 5.1, we discuss the uniform stability of Adam to upper-bound the generalization error of a DNN trained by it. Subsequently, In Subsection 5.2, we state different theorems for the uniform stability of AdamW and the generalization error because AdamW exploits decoupled weight decay, and its update parameters statement is different from Adam.

### 5.1. Adam Optimizer

Let  $\ell(f^\theta; B)$  represents the computation of a loss function on an arbitrary mini-batch,  $B = \{(x_i, y_i)\}_{i=1}^b$ , which we use at each iteration to update parameters in order to minimize  $R_{emp}(f^\theta)$ :

$$\ell(f^\theta; B) = \frac{1}{b} \sum_{i=1}^b \ell(f^\theta(x_i), y_i),$$

in which  $\theta$  are the parameters and  $b$  is the batch size. Let  $g(\theta) = \nabla_\theta \ell(f^\theta; B)$  where  $\nabla_\theta$  is the gradient. For  $t \geq 1$  suppose that  $m_t, v_t$  are estimates of the first and second moments respectively:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g(\theta_{t-1}); m_0 = 0, \tag{4}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g^2(\theta_{t-1}); v_0 = 0, \tag{5}$$

where  $\beta_1, \beta_2 \in (0, 1)$  are exponential decay rates and the multiply operation is element-wise. Let  $\hat{m}_t = m_t / (1 - \beta_1^t)$  and  $\hat{v}_t = v_t / (1 - \beta_2^t)$  be the bias-corrected estimates; Adam computes the parameters update using  $\hat{m}_t$  adapted by  $\hat{v}_t$ :

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{(\sqrt{\hat{v}_t} + \epsilon)},$$

<sup>4</sup>In the work of [12], this inequality has been written as  $\Delta_t \leq \min(1, \tau) \Delta_{t-1} + 2\sigma$  whose right side is less than  $\Delta_{t-1} + 2\sigma$  that we just need in the proofs of the theorems.

where  $\eta$  is the learning rate and  $\epsilon = 10^{-8}$ . Based on what we discussed so far, to evaluate the uniform stability of Adam, we need to formulate its update rule. Given  $\beta_1, \beta_2 \in (0, 1)$  for each  $1 \leq t \leq T$  let

$$\hat{M}(m_{t-1}, \theta) = \frac{\beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g(\theta)}{1 - \beta_1^t}, \tag{6}$$

$$\hat{V}(v_{t-1}, \theta) = \frac{\beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g^2(\theta)}{1 - \beta_2^t}, \tag{7}$$

where  $m_{t-1}$  and  $v_{t-1}$  are the biased estimates for the first and second moments of the gradient at the previous step respectively as we explained in the equations (4) and (5). Adam's update rule is obtained as follows:

$$A^t(\theta) = \theta - \eta \cdot \left( \frac{\hat{M}(m_{t-1}, \theta)}{\sqrt{\hat{V}(v_{t-1}, \theta) + \epsilon}} \right), \tag{8}$$

where  $\eta$  is the learning rate and the division operation is element-wise. We use the following lemma in the proof of Theorem 5.4:

**Lemma 5.3.** *Let  $m_{t-1} = \beta_1 \cdot m_{t-2} + (1 - \beta_1) \cdot g(\theta_{t-2})$  such that  $\beta_1 \in (0, 1)$  is constant and  $m_0 = 0$ . Let  $\ell(\hat{y}, y)$  be  $\gamma$ -Lipschitz. Then for all  $t \geq 1$  and  $\theta \in H$ , we have  $\left\| \hat{M}(m_{t-1}, \theta) \right\| \leq \gamma$ .*

The proof of Lemma 5.3 is available in Appendix A. Now we can state the theorems which link the generalization error with the loss function properties. In Theorem 5.4 we assess the stability measures including the uniform stability and in Theorem 5.5, we drive an upper bound for the generalization error of a DNN trained by Adam.

**Theorem 5.4.** *Assume Adam is executed for  $T$  iterations with a learning rate  $\eta$  and batch size  $b$  to minimize the empirical risk in order to obtain  $f_{B_S, R}$ . Let  $\ell(\hat{y}, y)$  be convex and  $\gamma$ -Lipschitz. Then, Adam is  $\beta$ -uniformly stable with regard to the loss function  $\ell$ , and for each  $(x, y)$ ,  $\ell(f_{B_S, R}(x), y)$  holds the  $\rho$ -BDC with respect to  $R$ . Consequently, we have*

$$\beta \leq \frac{2\eta}{c} \cdot \frac{bT\gamma^2}{N}, \quad \rho \leq \frac{8\eta}{c} \cdot \left( \frac{b\gamma}{N} \right)^2,$$

in which  $c \in (0, 1)$  is a constant number and  $N$  is the size of the training set.

**Proof.** Consider Adam's update rule,  $A^t(\cdot)$  in the equation (8). In order to prove that  $A^t(\cdot)$  satisfies the conditions of Lemma 5.2,  $\sigma$ -boundedness and  $\tau$ -expensiveness of  $A^t(\cdot)$  are needed to be evaluated. From the formula (2), we have:

$$\|\theta - A^t(\theta)\| = \left\| \eta \cdot \left( \frac{\hat{M}(m_{t-1}, \theta)}{\sqrt{\hat{V}(v_{t-1}, \theta) + \epsilon}} \right) \right\|,$$

where  $m_{t-1}$  and  $v_{t-1}$  are the biased estimates for  $\mathbb{E}[g]$  and  $\mathbb{E}[g^2] \geq 0$  in the  $t$ -th step respectively. Therefore:

$$\left\| \eta \cdot \left( \frac{\hat{M}(m_{t-1}, \theta)}{\sqrt{\hat{V}(v_{t-1}, \theta) + \epsilon}} \right) \right\| \leq \eta \cdot \left\| \frac{\hat{M}(m_{t-1}, \theta)}{\epsilon} \right\| \tag{9}$$

$$\leq \frac{\eta\gamma}{\epsilon}. \tag{10}$$

Because  $\epsilon > 0$  and  $\hat{V}(v_{t-1}, \theta) \geq 0$ , we deduced the inequality (9). In the inequality (10), Lemma 5.3 has been applied, which implies that,  $A^t(\cdot)$  is  $\sigma$ -bounded such that  $\sigma \leq \frac{\eta\gamma}{\epsilon}$ . Now, we check the  $\tau$ -expensiveness condition: we know that for all  $\theta \in H$ ,  $\frac{\hat{M}(m_{t-1}, \theta)}{\sqrt{\hat{V}(v_{t-1}, \theta)}} \simeq \pm 1$  because  $|\mathbb{E}[g]|/\sqrt{\mathbb{E}[g^2]} \leq 1$ . On the other hand  $\ell(\hat{y}, y)$  is convex. Thus, for two updates of network parameters  $\theta_{t-1}$  and  $\theta'_{t-1}$  in an arbitrary iteration  $t$  with the same initialization, by choosing a sufficiently small learning rate, the two vectors  $\frac{\hat{M}(m_{t-1}, \theta_{t-1})}{\sqrt{\hat{V}(v_{t-1}, \theta_{t-1})}}$  and  $\frac{\hat{M}(m_{t-1}, \theta'_{t-1})}{\sqrt{\hat{V}(v_{t-1}, \theta'_{t-1})}}$  are approximately equal. Thus, by substituting  $A^t(\cdot)$  in the formula (3), it is concluded that,  $A^t(\cdot)$  is 1-expensive. Note that in the training process, we only work with  $\theta_{t-1}, \theta'_{t-1}$  at each timestep  $t$ . Therefore, according to the definition of  $\Delta_{t-1}$  and the proof of the first case of Lemma 5.2, it is enough to prove 1-expensiveness for  $\theta_{t-1}, \theta'_{t-1}$  at each timestep  $t$ .

Let  $B_S$  and  $B_{S'}$  having equal size  $k$ , be two partitions of training sets  $S$  and  $S'$  respectively, such that  $B_S$  and  $B_{S'}$  are different in only one mini-batch. Let  $\theta_0, \theta_1, \dots, \theta_T$  and  $\theta'_0, \theta'_1, \dots, \theta'_T$  be two parameters updates obtained

from training the network by Adam with update rules  $A_S^t$  and  $A_{S'}^t$ , respectively where  $A_S^t$  runs on  $B_S$  and  $A_{S'}$  runs on  $B_{S'}$  with the same random sequence  $R$  such that  $\theta_0 = \theta'_0$ . Let two mini-batches  $B$  and  $B'$  have been selected for updating the parameters in the  $t$ -th iteration. If  $B = B'$ , then  $A_{S'}^t = A_S^t$  else  $A_{S'}^t \neq A_S^t$ .  $B = B'$  occurs with probability  $1 - \frac{1}{k}$  and the opposite occurs with probability  $\frac{1}{k}$ . At the beginning of the proof, we demonstrated that  $A^t(\cdot)$  (for an arbitrary training set) is  $\sigma$ -bounded and 1-expensive. Let  $\Delta_t = \|\theta_t - \theta'_t\|$ , from Lemma 5.2, we have:

$$\begin{aligned} \Delta_t &\leq \left(1 - \frac{1}{k}\right)\Delta_{t-1} + \frac{1}{k} \left(\Delta_{t-1} + \frac{2\eta\gamma}{\epsilon}\right) \\ &= \Delta_{t-1} + \frac{1}{k} \cdot \frac{2\eta\gamma}{\epsilon}. \end{aligned}$$

We know  $k = \frac{N}{b}$ . Therefore, solving the recursive relation gives

$$\Delta_T \leq \Delta_0 + 2T\eta \cdot \frac{\gamma}{k\epsilon} = 2\eta \cdot \frac{bT\gamma}{N\epsilon}.$$

Let  $\theta_{T,i}$  are the effective parameters of  $\theta_T$  on the  $i$ -th neuron of the last layer with  $M$  neurons. notation  $\langle \cdot, \cdot \rangle$  is inner product and  $[f(i)]_{i=1}^M$  for an arbitrary function  $f$ , denotes the vector  $[f(1), f(2), \dots, f(M)]$ . Now we proceed to prove Adam's uniform stability. According to Definition 5.1, we have:

$$\begin{aligned} \mathbb{E}_R (|\ell(f_{B_{S'},R}(x), y) - \ell(f_{B_S,R}(x), y)|) &\leq \mathbb{E}_R (\gamma \|f_{B_{S'},R}(x) - f_{B_S,R}(x)\|) \\ &= \gamma \mathbb{E}_R \left( \left\| [\langle \theta'_{T,i}, x \rangle]_{i=1}^M - [\langle \theta_{T,i}, x \rangle]_{i=1}^M \right\| \right) \\ &\leq \gamma \mathbb{E}_R (\|\theta'_T - \theta_T\|) \tag{11} \\ &= \gamma \mathbb{E}_R [\Delta_T] \end{aligned}$$

$$\leq 2\eta \cdot \frac{bT\gamma^2}{N\epsilon}. \tag{12}$$

In the inequality (11), we assumed  $\|x\| \leq 1$ ; that is the re-scaling technique that is common in deep learning. In the last inequality,  $\epsilon$  is a constant number between 0 and 1.

After showing the relation between the uniform stability of Adam and the Lipschitz constant of the loss function, we evaluate the bounded difference condition for the loss function with respect to the random sequence and a fixed training set. Suppose that  $R$  and  $R'$  are two random sequences of batch indices to update the parameters in which only the location of two indices has been changed; that is if  $R = (\dots, i, \dots, j, \dots)$  then  $R' = (\dots, j, \dots, i, \dots)$ . Without loss of generality, assume  $1 \leq i \leq \frac{k}{2}$  and  $\frac{k}{2} + 1 \leq j \leq k$ . The probability of selecting two identical batches in the  $t$ -th iteration is  $1 - \frac{4}{Tk^2}$ . Thus, two updates of neural network parameters as  $\theta_0^R, \theta_1^R, \dots, \theta_T^R$  and  $\theta_0^{R'}, \theta_1^{R'}, \dots, \theta_T^{R'}$  are made with the same initialization,  $\theta_0^R = \theta_0^{R'}$ . Let  $\Delta_t = \|\theta_t^R - \theta_t^{R'}\|$ . From Lemma 5.2, we have:

$$\Delta_T \leq \frac{8}{Tk^2} \cdot \frac{\eta T\gamma}{\epsilon} = \frac{8}{k^2} \cdot \frac{\eta\gamma}{\epsilon}.$$

According to Definition 3.4, we have:

$$\begin{aligned} |\ell(f_{B_{S'},R'}(x), y) - \ell(f_{B_S,R}(x), y)| &\leq \gamma \|f_{B_{S'},R'}(x) - f_{B_S,R}(x)\| \\ &= \gamma \left\| [\langle \theta_{T,i}^{R'}, x \rangle]_{i=1}^M - [\langle \theta_{T,i}^R, x \rangle]_{i=1}^M \right\| \\ &\leq \gamma \|\theta_T^{R'} - \theta_T^R\| \tag{13} \\ &= \gamma \Delta_T \end{aligned}$$

$$\leq \frac{8}{k^2} \cdot \frac{\eta\gamma^2}{\epsilon}. \tag{14}$$

The inequality (13) has been obtained similar to (12). Replacing  $k$  by  $\frac{N}{b}$  in the inequality (14) leads to the inequality in the proposition.  $\square$

**Theorem 5.5.** Let  $\ell(\hat{y}, y)$  with the maximum value of  $L$  be convex and  $\gamma$ -Lipschitz. Assume Adam is run for  $T$  iterations with a learning rate  $\eta$  and batch size  $b$  to obtain  $f_{B_S,R}$ . Then we have the following upper bound for  $E(f_{B_S,R})$  with probability at least  $1 - \delta$ :

$$E(f_{B_S,R}) \leq \frac{2\eta}{c} \left( 4 \left( \frac{b\gamma}{N} \right)^2 \sqrt{T \log(2/\delta)} + \frac{bT\gamma^2}{N} \left( 1 + \sqrt{2N \log(2/\delta)} \right) \right) + L \sqrt{\frac{\log(2/\delta)}{2N}}, \tag{15}$$

in which  $c \in (0, 1)$  is a constant number and  $N$  is the size of the training set.

**Proof.** In the work of [1], an upper bound for the generalization error of the output model trained by any optimization algorithm  $A_{opt}$  is established with probability at least  $1 - \delta$ , under the condition  $A_{opt}$  satisfies uniform stability measure with bound  $\beta$  and for each  $(x, y)$ ,  $\ell(f_{B_S, R}(x), y)$  holds the  $\rho$ -BDC with regard to  $R$ <sup>5</sup>:

$$E(f_{B_S, R}) \leq \rho\sqrt{T \log(2/\delta)} + \beta(1 + \sqrt{2N \log(2/\delta)}) + L\sqrt{\frac{\log(2/\delta)}{2N}}. \tag{16}$$

By combining Theorem 5.4 and the inequality (16), we have the following upper bound with probability  $1 - \delta$ :

$$E(f_{B_S, R}) \leq \frac{2\eta}{c} \left( 4 \left( \frac{b\gamma}{N} \right)^2 \sqrt{T \log(2/\delta)} + \frac{bT\gamma^2}{N} \left( 1 + \sqrt{2N \log(2/\delta)} \right) \right) + L\sqrt{\frac{\log(2/\delta)}{2N}}, \tag{17}$$

where  $c \in (0, 1)$  is a constant number. □

Theorem 5.5 shows how the generalization error bound of deep learning models trained by Adam depends on the Lipschitz constant  $\gamma$  and the maximum value  $L$ . Furthermore, the inequality (15), implies the sensitivity of the generalization error to the batch size; when the batch size grows,  $E(f_{B_S, R})$  increases. On the other hand, from the basics of machine learning, we know, if the batch size is too small, the parameters update is very noisy. Thus, an appropriate value should be considered for the batch size according to the training set size.

As we mentioned in Section 4 for the KL and GJM losses we have  $\gamma_{KL} \leq \gamma_{GJM}$  and  $L_{KL} \leq L_{GJM}$  [1]. Hence, following Theorem 5.5 we have the following corollary:

**Corollary 5.6.** Let  $f_{B_S, R}^{KL}$  and  $f_{B_S, R}^{GJM}$  be the output models trained by Adam optimizer, under the same settings using the KL and GJM loss functions respectively and the partition  $B_S$  obtained from the training set  $S$ . Then the upper bound of  $E(f_{B_S, R}^{GJM})$  is less than the upper bound of  $E(f_{B_S, R}^{KL})$ .

**Proof.** We know if  $\alpha = 0.5$ , then the Lipschitz constant and the maximum value of GJM are less than the Lipschitz constant and the maximum value of KL respectively [1]. So, under the same settings for hyper-parameters of Adam and the same initialization, from Theorem 5.5, the proposition is concluded. □

### 5.2. AdamW Optimizer

The objective of regularization techniques is to control the network parameters' domain in order to prevent the over-fitting issue.  $L_2$ -regularization which exploits the  $L_2$  norm of the parameters vector, is more practical than  $L_1$  because it keeps the loss function differentiable and convex. In continuation, we study  $L_2$ -regularization and note its effect on SGD and Adam. The lack of significant effect of this technique on Adam led to AdamW<sup>6</sup> [18].

Let  $\ell^{reg}(f^\theta; B)$  be a regularized loss function computed on a mini-batch,  $B = \{(x_i, y_i)\}_{i=1}^b$ :

$$\ell^{reg}(f^\theta; B) = \frac{1}{b} \left( \sum_{i=1}^b \ell(f^\theta(x_i), y_i) + \frac{\lambda}{2} \|\theta\|^2 \right), \tag{18}$$

where  $\|\cdot\|$  is the  $L_2$  norm,  $\lambda \in \mathbb{R}^+$  is the weight decay and  $b$  is the batch size. According to the equation (18), to compute the parameters update in SGD, we have:

$$\theta_t = \left( 1 - \frac{\eta\lambda}{b} \right) \theta_{t-1} - \frac{\eta}{b} \sum_{i=1}^b \nabla_{\theta} \ell(f^\theta(x_i), y_i).$$

In SGD, minimizing the regularized loss function can improve the output model generalization. However, this technique, cannot be effective in Adam because it uses adaptive gradients to update the parameters [18]. In AdamW, the weight decay hyper-parameter was decoupled from optimization steps taking gradient of the loss

<sup>5</sup>In the assumptions of the main theorem in the work of [1], it has been stated that the model is trained by stochastic gradient descent, but by studying the proof, we realize that their argument can be extended to any iterative algorithm that is  $\beta$ -uniformly stable because, in their proof, the upper bound has been derived independently of the update rule of stochastic gradient descent. The proof is available at <http://proceedings.mlr.press/v139/akbari21a/akbari21a-supp.pdf>.

<sup>6</sup>Adam with decoupled weight decay



function. Let  $\widehat{m}_t$  and  $\widehat{v}_t$  denote the bias-corrected estimates illustrated in Subsection 5.1. The parameters update is computed as follows:

$$\theta_t = \theta_{t-1} - \alpha_t \left( \eta \cdot \frac{\widehat{m}_t}{(\sqrt{\widehat{v}_t} + \epsilon)} + \lambda \theta \right), \tag{19}$$

where  $\alpha_t$  is the schedule multiplier. The equation (19) exhibits that AdamW updates the parameters in a different way than Adam. Hence, we need to state theorems specific to AdamW for the stability and the generalization error. Consider  $\widehat{M}(m_{t-1}, \theta)$  and  $\widehat{V}(v_{t-1}, \theta)$  in the equations (6) and (7). According to parameters update statement of AdamW in formula (19), The AdamW's update rule is defined as

$$A_W^t(\theta) = \theta - \alpha_t \left( \eta \cdot \frac{\widehat{M}(m_{t-1}, \theta)}{\sqrt{\widehat{V}(v_{t-1}, \theta)} + \epsilon} + \lambda \theta \right). \tag{20}$$

where  $0 < \alpha_t \lambda < 1$  because otherwise, the update occurs in a wrong direction which means it goes away from the minimum. Consider  $H$  as the set of all possible values for the network parameters. As we noted in Section 3,  $H$  is bounded. Hence, the supremum of  $H$  is well-defined. Let  $\|\theta\|_{\text{sup}} := \sup_{\theta \in H} \|\theta\|$ :

**Theorem 5.7.** Assume AdamW is executed for  $T$  iterations with a learning rate  $\eta$ , batch size  $b$ , weight decay  $\lambda$ , and schedule multiplier  $\alpha_t$  to minimize the empirical risk in order to obtain  $f_{B_S, R}$ . Let  $\ell(\hat{y}, y)$  be convex and  $\gamma$ -Lipschitz. Then, Adam is  $\beta$ -uniformly stable with regard to the loss function  $\ell$ , and for each  $(x, y)$ ,  $\ell(f_{B_S, R}(x), y)$  holds the  $\rho$ -BDC with respect to  $R$ . Consequently, we have

$$\beta \leq \frac{2bT}{N} \sum_{t=1}^T \alpha_t \left( \frac{\eta \gamma^2}{c} + \gamma \lambda \|\theta\|_{\text{sup}} \right), \quad \rho \leq \frac{8b^2}{N^2} \sum_{t=1}^T \alpha_t \left( \frac{\eta \gamma^2}{c} + \gamma \lambda \|\theta\|_{\text{sup}} \right),$$

in which  $c \in (0, 1)$  is a constant number and  $N$  is the size of the training set.

**Proof.** First, we check the  $\sigma$ -boundedness of  $A_W^t(\theta)$ :

$$\begin{aligned} \|\theta - A_W^t(\theta)\| &= \left\| \alpha_t \left( \eta \cdot \frac{\widehat{M}(m_{t-1}, \theta)}{\sqrt{\widehat{V}(v_t, \theta)} + \epsilon} + \lambda \theta \right) \right\| \\ &\leq \left\| \alpha_t \eta \cdot \frac{\widehat{M}(m_{t-1}, \theta)}{\epsilon} \right\| + \alpha_t \lambda \|\theta\| \\ &= \alpha_t \left( \eta \cdot \frac{\|\widehat{M}(m_{t-1}, \theta)\|}{\epsilon} + \lambda \|\theta\| \right) \\ &\leq \alpha_t \left( \frac{\eta \gamma}{\epsilon} + \lambda \|\theta\|_{\text{sup}} \right). \end{aligned} \tag{21}$$

By applying Lemma 5.3, we concluded the inequality (21), which shows that  $A_W^t(\theta)$  is  $\sigma$ -bounded. Now we evaluate the  $\tau$ -expensiveness of AdamW. According to the formula (3), we have

$$\frac{\|A_W^t(\theta) - A_W^t(\theta')\|}{\|\theta - \theta'\|} = \frac{\left\| -\alpha_t \left( \eta \cdot \frac{\widehat{M}(m_{t-1}, \theta)}{\sqrt{\widehat{V}(v_{t-1}, \theta)} + \epsilon} + \lambda \theta \right) + \alpha_t \left( \eta \cdot \frac{\widehat{M}(m_{t-1}, \theta')}{\sqrt{\widehat{V}(v_{t-1}, \theta')} + \epsilon} + \lambda \theta' \right) + \theta - \theta' \right\|}{\|\theta - \theta'\|}. \tag{22}$$

As said in the proof of Theorem 5.4, for every  $\theta \in H$ , we have  $\frac{\widehat{M}(m_{t-1}, \theta)}{\sqrt{\widehat{V}(v_{t-1}, \theta)}} \simeq \pm 1$  because  $|\mathbb{E}[g]|/\sqrt{\mathbb{E}[g^2]} \leq 1$ . Therefore, the equation (22) is written as follows:

$$\begin{aligned} \frac{\|-\alpha_t \lambda \theta + \alpha_t \lambda \theta' + \theta - \theta'\|}{\|\theta - \theta'\|} &= \frac{\|\alpha_t \lambda (\theta' - \theta) + \theta - \theta'\|}{\|\theta - \theta'\|} \\ &= \frac{|1 - \alpha_t \lambda| \|\theta - \theta'\|}{\|\theta - \theta'\|} \\ &= |1 - \alpha_t \lambda| < 1. \end{aligned} \tag{23}$$

AdamW update rule in the equation (20) implies that  $0 < \alpha_t \lambda < 1$  which its consequent is the inequality (23) with an analogous demonstration to what we did in the proof of Theorem 5.4, i.e. considering update sequences

and using Lemma 5.2 in order to evaluate the uniform stability and bounded difference condition according to their definitions, we conclude the following inequalities:

$$\begin{aligned} \beta &\leq \frac{2bT}{N} \sum_{t=1}^T \alpha_t \left( \frac{\eta\gamma^2}{\epsilon} + \gamma\lambda \|\theta\|_{\text{sup}} \right), \\ \rho &\leq \frac{8b^2}{N^2} \sum_{t=1}^T \alpha_t \left( \frac{\eta\gamma^2}{\epsilon} + \gamma\lambda \|\theta\|_{\text{sup}} \right). \end{aligned} \quad \square$$

**Theorem 5.8.** Let  $\ell(\hat{y}, y)$  with the maximum value of  $L$  be convex and  $\gamma$ -Lipschitz. Assume AdamW is run for  $T$  iterations with a learning rate  $\eta$ , batch size  $b$ , weight decay  $\lambda$ , and schedule multiplier  $\alpha_t$  to obtain  $f_{B_S, R}$ . Then we have the following upper bound for  $E(f_{B_S, R})$  with probability at least  $1 - \delta$ :

$$E(f_{B_S, R}) \leq \frac{2b}{N} \sum_{t=1}^T \alpha_t \left( \frac{\eta\gamma^2}{c} + \gamma\lambda \|\theta\|_{\text{sup}} \right) \left( \frac{4b}{N} \sqrt{T \log(2/\delta)} + T \sqrt{2N \log(2/\delta)} \right) + L \sqrt{\frac{\log(2/\delta)}{2N}}. \quad (24)$$

in which  $c \in (0, 1)$  is a constant number and  $N$  is the size of the training set.

**Proof.** By combining the equation (16) and Theorem 5.7 we conclude the proposition. □

The inequality (24) implies that the generalization error growth of a DNN trained by AdamW, is directly related to the Lipschitz constant and the maximum value of a loss function. Following Theorem 5.8 we have the following corollary for the KL and GJM loss functions:

**Corollary 5.9.** Let  $f_{B_S, R}^{KL}$  and  $f_{B_S, R}^{GJM}$  be the output models trained by AdamW optimizer, under the same settings using the KL and GJM loss functions respectively and the partition  $B_S$  obtained from the training set  $S$ . Then the upper bound of  $E(f_{B_S, R}^{GJM})$  is less than the upper bound of  $E(f_{B_S, R}^{KL})$ .

**Proof.** The proposition is concluded by Theorem 5.8 and an analogous argument to Corollary 5.6. □

## 6. Experimental Evaluation

### 6.1. Datasets

We use 4 datasets, including UTKFace [31], AgeDB [20], MegaAge-Asian [30], and FG-NET [6] to evaluate age estimation performance. UTKFace dataset contains 23,708 facial images, providing enough samples of all ages, ranging from 0 to 116 years-old. AgeDB contains 16,488 in-the-wild images in the age range from 0 to 100 years-old. MegaAge-Asian has been already split into MegaAge-Train and Mega-Age-Test datasets, containing 40,000 and 3,945 images respectively, belonging to Asian people with the age label in the range from 1 to 69 years-old. FG-NET dataset contains 1,002 facial images in the age range of 0 to 69 years. This dataset covers variations in pose, age expression, resolution, and lighting conditions. By collecting the samples from UTKFace, MegaAge-Train, and AgeDB datasets whose ages are in the range from 0 to 100 years-old, we create a new dataset called UAM, which includes 80,174 images. We use UTKFace and UAM as the training sets. FG-NET, MegaAge-Test, and 10% randomly selected from AgeDB called AgeDB-Test, are left as the test sets.

### 6.2. Settings

All images are pre-processed by the following procedures: face detection and alignment are done by prepared modules in OpenCV package. All images are reshaped to the size of  $256 \times 256$  and standard data augmentation techniques, including random cropping and horizontal flipping, are carried out during the training phase. We use two neural network architectures VGG16 [22] and ResNet50 [15], pre-trained on ImageNet [7] and VGGFace2 [5] datasets respectively, to estimate human age. VGGFace2 dataset was created with the aim of estimating human pose and age. With the same seed, the last layer of these models is replaced with a M-neurons dense layer with random weights. The last layer of VGG16 is trained on UTKFace in 5 epochs and the last layer of ResNet50 is trained on UAM in 15 epochs. M is set to 116 in VGG16 and 101 in ResNet50 model. We train the models via Adam and AdamW with learning rate  $2 \times 10^{-5}$  for KL and  $10^{-4}$  for GJM<sup>7</sup>. The batch size and AdamW's weight decay are set to 64 and 0.1 respectively. We set  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  for both Adam and AdamW as the authors of [17] and [18] suggested.

<sup>7</sup>In our experiments, when we set the learning rate to  $2 \times 10^{-5}$  for the GJM loss, the ultimate model at the last epoch remained under-fit.

### 6.3. Evaluation Metrics and Results

As the first observation, we measure the generalization error estimate in the training steps of ResNet50 trained by Adam and AdamW which is defined as

$$\hat{E}(f_{B_S,R}) = |R_{train}(f_{B_S,R}) - R_{val}(f_{B_S,R})|,$$

where  $f_{B_S,R}$  is the output model,  $R_{train}(f_{B_S,R})$ ,  $R_{val}(f_{B_S,R})$  are the average of loss values on the training and validation sets respectively. The results of this experiment are shown in Figure 1 and Figure 2. In the first epochs, the models are still under-fit and the loss is far from its minimum; therefore,  $\hat{E}(f_{B_S,R})$  does not give us critical information about the generalization error, but in the rest of epochs, when the experimental loss of the models approaches its minimum,  $\hat{E}(f_{B_S,R})$  can represent the generalization error. As can be seen in Figure 1a and Figure 2a, after epoch 5 or 6 the generalization error estimate of the models trained by Adam and AdamW using the GJM loss function is lower than the models trained using the KL loss.

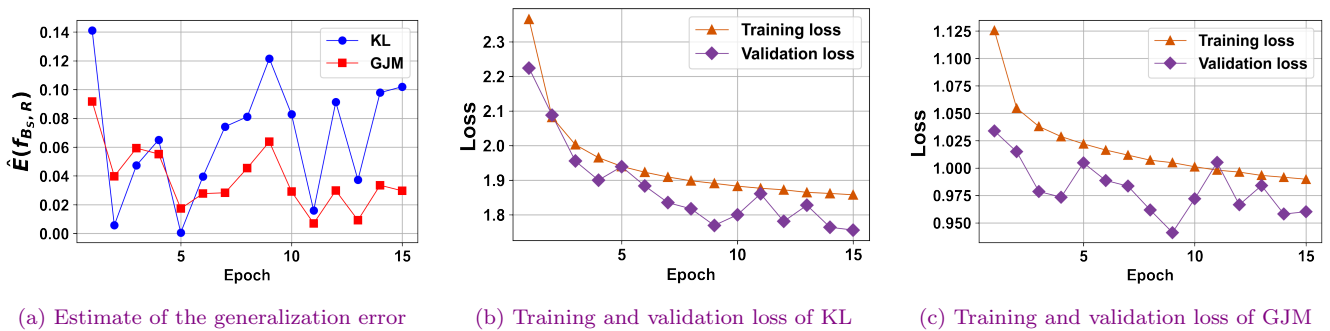


Figure 1: Loss measurement (Model: ResNet50, Optimizer: Adam, Training set: UAM, Validation set: FG-NET)

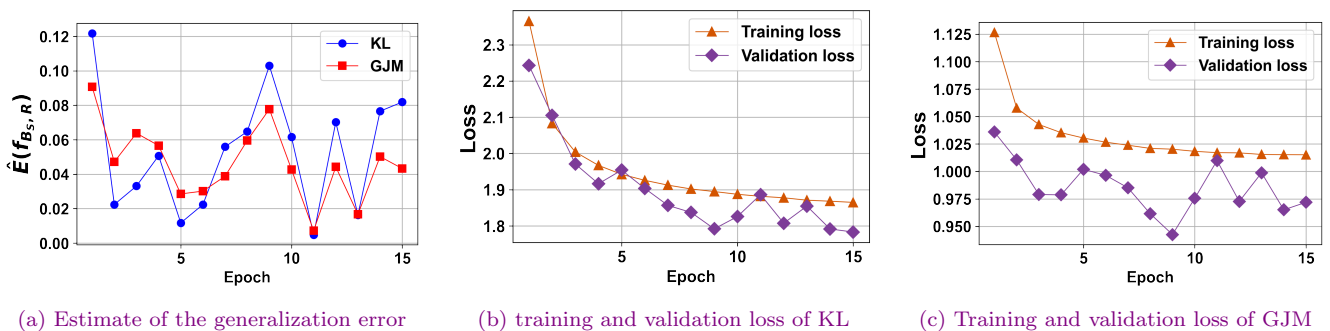


Figure 2: Loss measurement (Model: ResNet50, Optimizer: AdamW, Training set: UAM, Validation set: FG-NET)

In addition, we measure the generalization performance in terms of Mean Absolute Error (MAE) and Cumulative Score (CS). Consider the training set  $S$ , and the test set  $S_{test} \in (X \times Y)^D$ . Let  $(x_k, y_k) \in S_{test}$  represents a test example where  $y_k \in \mathbb{N}$  is the label of  $k$ -th example of the test set. Since we use label distribution learning, for each  $(x, y) \in S$ ,  $y \in \mathbb{R}^M$  is the probability distribution corresponding to  $x$ . Therefore, in the evaluation phase, the output of the model for the test example  $x_k$  is the predicted probability distribution  $\hat{y}_k = [\hat{y}_{k,1}, \hat{y}_{k,2}, \dots, \hat{y}_{k,M}]$ . MAE is defined as  $\frac{1}{D} \sum_{k=1}^D |\hat{l}_k - l_k|$  where  $\hat{l}_k$  is the index of the largest element of  $\hat{y}_k$  and  $l_k$  is the true label. CS is defined as  $\frac{D_I}{D} \times 100\%$  where  $D_I$  is the number of test samples such that  $|\hat{l}_k - l_k| < I$ . Commonly, the value of  $I$  is set to 5 [1, 27].

The results are reported in Tables 1-3. The ResNet50 models are more accurate than the VGG16 models because VGG16 is pre-trained on ImageNet dataset which is not suitable for age estimation. Tables 1-3 show that when we train a DNN by Adam or AdamW, the GJM loss performs better than the KL loss.

Table 1: MAE and CS metrics (Model: VGG16, Optimizer: AdamW, Training set: UTKFace)

Method	FG-NET		AgeDB-Test	
	MAE	CS(%)	MAE	CS(%)
LDL (KL)	14.41	15.67	22.77	16.13
LDL (GJM)	<b>14.11</b>	<b>17.06</b>	<b>22.53</b>	<b>17.04</b>

Table 2: MAE and CS metrics (Model: ResNet50, Optimizer: AdamW, Training set: UAM)

Method	FG-NET		MegaAge-Test	
	MAE	CS(%)	MAE	CS(%)
LDL (KL)	5.55	54.49	6.23	50.59
LDL (GJM)	<b>5.48</b>	<b>54.90</b>	<b>6.21</b>	<b>51.58</b>

Table 3: MAE and CS metrics (Model: ResNet50, Optimizer: Adam, Training set: UAM)

Method	FG-NET		MegaAge-Test	
	MAE	CS(%)	MAE	CS(%)
LDL (KL)	5.43	55.09	6.28	49.96
LDL (GJM)	<b>5.27</b>	<b>56.57</b>	<b>6.16</b>	<b>51.91</b>

## 7. Conclusion

In this paper, we have shown how the properties of a loss function affect the generalization performance of a DNN trained by Adam or AdamW. We theoretically linked the Lipschitz constant and the maximum value of a loss function to the generalization error of the output model obtained by Adam or AdamW. We evaluated our theoretical results in the human age estimation problem, where we trained and tested the models on various datasets. In our future work, we focus on the tasks addressed with single-label and multi-label learning instead of label distribution learning because there are several classification tasks in computer vision, recommender systems, predictive modeling, etc where the models do not need to learn the label distribution, and the training process is done using the cross-entropy loss. Our next step is proposing an alternative loss function to cross-entropy to improve the generalization performance of single-label and multi-label learning models.

## A. Appendix: Proofs of Lemmas

### A.1. Proof of Lemma 5.2

**Proof.**

- The first case:

$$\Delta_t = \|\theta_t - \theta'_t\| = \|A_S^t(\theta_{t-1}) - A_{S'}^t(\theta'_{t-1})\| = \|A_S^t(\theta_{t-1}) - A_S^t(\theta'_{t-1})\| \leq \tau \|\theta_{t-1} - \theta'_{t-1}\|.$$

- The second case:

$$\begin{aligned} \Delta_t &= \|\theta_t - \theta'_t\| \\ &= \|A_S^t(\theta_{t-1}) - A_{S'}^t(\theta'_{t-1})\| \\ &= \|(A_{S'}^t(\theta'_{t-1}) - \theta'_{t-1}) + (\theta_{t-1} - A_S^t(\theta_{t-1})) + \theta'_{t-1} - \theta_{t-1}\| \\ &\leq \|\theta'_{t-1} - \theta_{t-1}\| + \|A_{S'}^t(\theta'_{t-1}) - \theta'_{t-1}\| + \|\theta_{t-1} - A_S^t(\theta_{t-1})\| \\ &\leq \Delta_{t-1} + 2\sigma. \end{aligned}$$

□

### A.2. Proof of Lemma 5.3

**Proof.** We know  $\|\nabla_{\theta}\ell(f^{\theta}(x), y)\| \leq \gamma$  for each  $(x, y)$ . Hence, for every  $\theta \in H$  and a mini-batch  $B = \{(x_i, y_i)\}_{i=1}^b$  we have

$$\|g(\theta)\| = \|\nabla_{\theta}\ell(f^{\theta}; B)\| = \left\| \frac{1}{b} \sum_{i=1}^b \nabla_{\theta}\ell(f^{\theta}(x_i), y_i) \right\| \leq \gamma.$$

The proposition is clear for  $t = 1$ . For  $t > 1$  by solving the recurrence relation of  $m_{t-1}$  we have:

$$\hat{M}(m_{t-1}, \theta) = \frac{(1 - \beta_1) \sum_{j=1}^{t-1} \beta_1^{t-j} g(\theta_{j-1}) + (1 - \beta_1)g(\theta)}{1 - \beta_1^t}.$$

Now we proceed to draw the conclusion:

$$\begin{aligned} \|\hat{M}(m_{t-1}, \theta)\| &= \left\| \frac{(1 - \beta_1) \sum_{j=1}^{t-1} \beta_1^{t-j} g(\theta_{j-1}) + (1 - \beta_1)g(\theta)}{1 - \beta_1^t} \right\| \\ &\leq \frac{(1 - \beta_1) \sum_{j=1}^{t-1} \beta_1^{t-j} \|g(\theta_{j-1})\| + (1 - \beta_1) \|g(\theta)\|}{1 - \beta_1^t} \\ &\leq \frac{(1 - \beta_1) \sum_{j=1}^t \beta_1^{t-j}}{1 - \beta_1^t} \cdot \gamma \\ &= \gamma. \end{aligned}$$

□

### References

- [1] A. AKBARI, M. AWAIS, M. BASHAR, AND J. KITTLER, *How does loss function affect generalization performance of deep learning? application to human age estimation*, in International Conference on Machine Learning, PMLR, 2021, pp. 141–151.
- [2] A. BANERJEE, T. CHEN, X. LI, AND Y. ZHOU, *Stability based generalization bounds for exponential family Langevin dynamics*, in Proceedings of the 39th International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., vol. 162 of Proceedings of Machine Learning Research, PMLR, 17–23 Jul 2022, pp. 1412–1449.
- [3] S. BASU, S. MUKHOPADHYAY, M. KARKI, R. DIBIANO, S. GANGULY, R. NEMANI, AND S. GAYAKA, *Deep neural networks for texture classification—a theoretical analysis*, Neural Networks, 97 (2018), pp. 173–182.
- [4] O. BOUSQUET AND A. ELISSEEFF, *Stability and generalization*, The Journal of Machine Learning Research, 2 (2002), pp. 499–526.
- [5] Q. CAO, L. SHEN, W. XIE, O. M. PARKHI, AND A. ZISSERMAN, *Vggface2: A dataset for recognising faces across pose and age*, in 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018), IEEE, 2018, pp. 67–74.
- [6] K. CHEN, S. GONG, T. XIANG, AND C. CHANGE LOY, *Cumulative attribute space for age and crowd density estimation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2467–2474.
- [7] J. DENG, W. DONG, R. SOCHER, L.-J. LI, K. LI, AND L. FEI-FEI, *Imagenet: A large-scale hierarchical image database*, in 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- [8] J. DUCHI, E. HAZAN, AND Y. SINGER, *Adaptive subgradient methods for online learning and stochastic optimization.*, Journal of machine learning research, 12 (2011).
- [9] X. GENG, *Label distribution learning*, IEEE Transactions on Knowledge and Data Engineering, 28 (2016), pp. 1734–1748.
- [10] J. GUAN AND Z. LU, *Fast-rate PAC-Bayesian generalization bounds for meta-learning*, in Proceedings of the 39th International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds., vol. 162 of Proceedings of Machine Learning Research, PMLR, 17–23 Jul 2022, pp. 7930–7948.

- [11] ———, *Fast-rate pac-bayesian generalization bounds for meta-learning.*, in ICML, 2022, pp. 7930–7948.
- [12] M. HARDT, B. RECHT, AND Y. SINGER, *Train faster, generalize better: Stability of stochastic gradient descent*, in International conference on machine learning, PMLR, 2016, pp. 1225–1234.
- [13] N. HARVEY, C. LIAW, AND A. MEHRABIAN, *Nearly-tight VC-dimension bounds for piecewise linear neural networks*, in Proceedings of the 2017 Conference on Learning Theory, S. Kale and O. Shamir, eds., vol. 65 of Proceedings of Machine Learning Research, PMLR, 07–10 Jul 2017, pp. 1064–1068.
- [14] E. HAZAN, K. LEVY, AND S. SHALEV-SHWARTZ, *Beyond convexity: Stochastic quasi-convex optimization*, Advances in neural information processing systems, 28 (2015).
- [15] J. HU, L. SHEN, AND G. SUN, *Squeeze-and-excitation networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
- [16] D. JAKUBOVITZ, R. GIRYES, AND M. R. RODRIGUES, *Generalization error in deep learning*, arXiv preprint arXiv:1808.01174, (2018).
- [17] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).
- [18] I. LOSHCHILOV AND F. HUTTER, *Decoupled weight decay regularization*, arXiv preprint arXiv:1711.05101, (2017).
- [19] C. MCDIARMID, *On the method of bounded differences*, in “survey in combinatorics,”(j. simons, ed.) london mathematical society lecture notes, vol. 141, 1989.
- [20] S. MOSCHOGLOU, A. PAPAIOANNOU, C. SAGONAS, J. DENG, I. KOTSIA, AND S. ZAFEIRIOU, *Agedb: the first manually collected, in-the-wild age database*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, vol. 2, 2017, p. 5.
- [21] B. NEYSHABUR, S. BHOJANAPALLI, AND N. SREBRO, *A pac-bayesian approach to spectrally-normalized margin bounds for neural networks*, in International Conference on Learning Representations, 2018.
- [22] O. M. PARKHI, A. VEDALDI, AND A. ZISSERMAN, *Deep face recognition*, in Proceedings of the British Machine Vision Conference (BMVC), M. W. J. Xianghua Xie and G. K. L. Tam, eds., BMVA Press, September 2015, pp. 41.1–41.12.
- [23] M. REN, W. ZENG, B. YANG, AND R. URTASUN, *Learning to reweight examples for robust deep learning*, in International conference on machine learning, PMLR, 2018, pp. 4334–4343.
- [24] R. ROTHE, R. TIMOFTE, AND L. VAN GOOL, *Deep expectation of real and apparent age from a single image without facial landmarks*, International Journal of Computer Vision, 126 (2018), pp. 144–158.
- [25] F. SCARSELLI, A. C. TSOI, AND M. HAGENBUCHNER, *The vapnik–chervonenkis dimension of graph and recursive neural networks*, Neural Networks, 108 (2018), pp. 248–259.
- [26] S. SHALEV-SHWARTZ, O. SHAMIR, N. SREBRO, AND K. SRIDHARAN, *Learnability, stability and uniform convergence*, The Journal of Machine Learning Research, 11 (2010), pp. 2635–2670.
- [27] W. SHEN, Y. GUO, Y. WANG, K. ZHAO, B. WANG, AND A. L. YUILLE, *Deep regression forests for age estimation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
- [28] T. TIELEMAN, G. HINTON, ET AL., *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural networks for machine learning, 4 (2012), pp. 26–31.
- [29] T. ZAHAVY, B. KANG, A. SIVAK, J. FENG, H. XU, AND S. MANNOR, *Ensemble robustness and generalization of stochastic deep learning algorithms*, arXiv preprint arXiv:1602.02389, (2016).
- [30] Y. ZHANG, L. LIU, C. LI, AND C. C. LOY, *Quantifying facial age by posterior of age comparisons*, in British Machine Vision Conference (BMVC), 2017.
- [31] Z. ZHANG, Y. SONG, AND H. QI, *Age progression/regression by conditional adversarial autoencoder*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5810–5818.

Please cite this article using:

Mohammad Lashkari, Amin Gheibi, Lipschitzness effect of a loss function on generalization performance of deep neural networks trained by Adam and AdamW optimizers, *AUT J. Math. Comput.*, 5(4) (2024) 361-375

<https://doi.org/10.22060/AJMC.2023.22182.1139>

