ON ALIGNMENT IN DEEP LINEAR NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the properties of *alignment*, a form of implicit regularization, in linear neural networks under gradient descent. We define alignment for fully connected networks with multidimensional outputs and show that it is a natural extension of alignment in networks with 1-dimensional outputs as defined by Ji and Telgarsky, 2018. While in fully connected networks, there always exists a global minimum corresponding to an aligned solution, we analyze alignment as it relates to the training process. Namely, we characterize when alignment is an *invariant* of training under gradient descent by providing necessary and sufficient conditions for this invariant to hold. In such settings, the dynamics of gradient descent simplify, thereby allowing us to provide an explicit learning rate under which the network converges linearly to a global minimum. We then analyze networks with layer constraints such as convolutional networks. In this setting, we prove that gradient descent is equivalent to projected gradient descent, and that alignment is impossible with sufficiently large datasets.

1 INTRODUCTION

Although overparameterized deep networks can interpolate randomly labeled training data (Du et al., 2019; Wu et al., 2019), training overparameterized networks with modern optimizers often leads to solutions that generalize well. This suggests that there is a form of *implicit regularization* occurring through training (Zhang et al., 2017).

As an example of implicit regularization, the authors in Ji & Telgarsky (2018) proved that the layers of linear neural networks used for binary classification on linearly separable datasets become aligned in the limit of training. That is, for a linear network parameterized by the matrix product $W_d W_{d-1} \dots W_1$, the top left/right singular vectors u_i and v_i of layer W_i satisfy $|v_{i+1}^T u_i| \to 1$ as the number of gradient descent steps goes to infinity.

Alignment of singular vector spaces between adjacent layers allows for the network representation to be drastically simplified (see Equation 3); namely, the product of all layers becomes a product of diagonal matrices with the exception of the outermost unitary matrices. If alignment is an invariant of training, then optimization over the set of weight matrices reduces to optimization over the set of singular values of weight matrices. Thus, importantly, alignment of singular vector spaces allows for the gradient descent update rule to be simplified significantly, which was used in Ji & Telgarsky (2018) to show convergence to a max-margin solution.

In this work, we generalize the definition of *alignment* to the multidimensional setting. We study when alignment can occur and moreover, under which conditions it is an invariant of training in linear neural networks under gradient descent. Prior works (Gidel et al., 2019; Saxe et al., 2014; 2019) have implicitly relied on invariance of alignment as an assumption on initialization to simplify training dynamics for 2 layer networks. In this work, we provide necessary and sufficient conditions for when alignment is an invariant for networks of arbitrary depth. Our main contributions are as follows:

1. We extend the definition of alignment from the 1-dimensional classification setting to the multi-dimensional setting (Definition 2) and characterize when alignment is an invariant of training in linear fully connected networks with multi-dimensional outputs (Theorem 1).
2. We demonstrate that alignment is an invariant for fully connected networks with multidimensional outputs only in special problem classes including autoencoding, matrix factorization

and matrix sensing. This is in contrast to networks with 1-dimensional outputs, where there exists an initialization such that adjacent layers remain aligned throughout training under *any* real-valued loss function and any training dataset.

3. Alignment largely simplifies the analysis of training linear networks: We provide an explicit learning rate under which gradient descent converges linearly to a global minimum under alignment in the squared loss setting (Proposition 1).
4. We prove that alignment cannot occur, let alone be invariant, in networks with constrained layer structure (such as convolutional networks), when the amount of training data dominates the dimension of the layer structure (Theorem 3).
5. We support our theoretical findings via experiments in Section 6.

As a consequence, our characterization of the invariance properties of alignment provides settings under which the gradient descent dynamics can be simplified and the implicit regularization properties can be fully understood, yet also shows that further results are required to explain implicit regularization in linear neural networks more generally.

2 Related Work

Implicit regularization in overparameterized networks has become a subject of significant interest (Gunasekar et al., 2018a;b; Martin & Mahoney, 2018; Neyshabur et al., 2014). In order to characterize the specific form of implicit regularization, several works have focused on analyzing deep *linear* networks (Arora et al., 2019b; Gunasekar et al., 2018b; 2017; Soudry et al., 2018). Even though such networks can only express linear maps, parameter optimization in linear networks is non-convex and is studied in order to obtain intuition about optimization of deep networks more generally.

One such form of implicit regularization is *alignment*, identified by Ji & Telgarsky (2018) to analyze linear fully connected networks with 1-dimensional outputs trained on linearly separable data. They proved that in the limit of training, each layer, after normalization, approaches a rank 1 matrix, i.e.

$$\lim_{t\to\infty} \frac{W_i^{(t)}}{\|W_i^{(t)}\|_F} = u_i v_i^T$$

and that adjacent layers, W_{i+1} and W_i become *aligned*, i.e. $|v_{i+1}^T u_i| \to 1$.

In addition, Ji & Telgarsky (2018) proved that alignment in this setting occurs concurrently with convergence to the max-margin solution. Follow-up work mainly focused on this convergence phenomenon and gave explicit convergence rates for overparameterized networks trained with gradient descent (Arora et al., 2019c; Zou et al., 2018).

Our definition of invariance of alignment extends assumptions on initialization appearing in various prior works (Gidel et al., 2019; Saxe et al., 2014; 2019). While the connection to alignment was not mentioned in their work, the authors in Gidel et al. (2019) begin to generalize alignment to multidimensional outputs by considering two-layer networks initialized so that layers are aligned with each other and to the data. We generalize this to networks of any depth, showing that our definition of alignment corresponds to the initialization considered in Gidel et al. (2019). Moreover, we establish necessary and sufficient conditions for when alignment is an invariant of training in Theorem 1 instead of assuming these conditions. Furthermore, their result on sequential learning of components can be derived via our singular value update rule in Corollary 1.

Balancedness is another closely related form of implicit regularization in linear neural networks. It was introduced in Arora et al. (2018) and defined as the property that if $W_i^T W_i = W_{i+1} W_{i+1}^T$ for all i at initialization, then this property is invariant under gradient flow. Du et al. (2018) present a more general form, that $W_i^T W_i - W_{i+1} W_{i+1}^T$ is constant under gradient flow. In practice, analyses rely on this quantity being close to or exactly zero. In this exact setting, balancedness indeed implies alignment of singular vector spaces between consecutive layers. To study gradient descent, slightly more general notions such as approximate balancedness (Arora et al., 2019a) and ϵ-balancedness have been introduced. Du et al. (2018) also defined balancedness with respect to convolutional networks, showing that under gradient flow, the difference in the norm of the weights of consecutive layers is an invariant. Generally, the goal of identifying invariants of training such as balancedness or alignment is to help understand both the dynamics of training and properties of solutions at the end of training.

3 DEFINITION OF ALIGNMENT IN THE MULTI-DIMENSIONAL SETTING

In this section, we first define alignment for linear neural networks with multi-dimensional outputs. We then define when alignment is an invariant of training.

We consider *linear* neural networks. Let $f : \mathbb{R}^{k_0} \to \mathbb{R}^{k_d}$ denote such a d-layer network, i.e.

$$f(x) = W_d W_{d-1} \dots W_1 x, \tag{1}$$

where $W_i \in \mathbb{R}^{k_i \times k_{i-1}}$ for $i \in [d]$, where we follow the convention that $[d] = \{1, 2, \dots d\}$. Let $(X, Y) \in \mathbb{R}^{k_0 \times n} \times \mathbb{R}^{k_d \times n}$ denote the set of training data pairs $\{(x^{(i)}, y^{(i)})\}$ for $i \in [n]$. Gradient descent with learning rate γ is used to find a solution to the following optimization problem:

$$\arg\min_{f \in \mathcal{F}} \frac{1}{2n} \sum_{i=1}^{n} \ell(f(x^{(i)}), y^{(i)}), \tag{2}$$

where $\mathcal{F}$ is the set of linear functions represented by f and ℓ is a real-valued loss function. When not stated otherwise, we assume $\ell(f(x^{(i)}), y^{(i)}) = \|y^{(i)} - f(x^{(i)})\|_2^2$, which is the squared loss (MSE). In addition, we denote by $W_i^{(t)}$ for $t \in \mathbb{Z}_{\geq 0}$ the weight matrix W_i after t steps of gradient descent. When there are no additional constraints on the matrices W_i, then f is a fully connected network.

We next introduce a generalized form of the singular value decomposition:

Definition 1. *An **unsorted, signed singular value decomposition (usSVD)** of a matrix $A \in \mathbb{R}^{m \times n}$ is a triple $U \in \mathbb{R}^{m \times m}, \Sigma \in \mathbb{R}^{m \times n}, V \in \mathbb{R}^{n \times n}$ such that U, V are orthonormal matrices, Σ is diagonal, and $A = U \Sigma V^T$.*

In contrast to the usual definition of singular value decomposition (SVD) of a matrix, the diagonal entries of Σ may be in any order and take negative values. Throughout, we will refer to the entries of Σ in a usSVD as singular values and the vectors in U, V as singular vectors. Using the usSVD, we now generalize the notion of alignment from Ji & Telgarsky (2018) to the multi-dimensional setting.

Definition 2. *Let $f = W_d W_{d-1} \dots W_1$ be a linear network. We say that f is **aligned** if there exists a usSVD $W_i = U_i \Sigma_i V_i^T$ with $U_i = V_{i+1}$ for all $i \in [d-1]$. (We also say that a matrix A is aligned with another matrix B if there exist usSVD's $A = U_A \Sigma_A V_A^T, B = U_B \Sigma_B V_B^T$ such that $V_A = U_B$.)*

Note that if W_i and W_{i+1} are rank 1 matrices in an aligned network f, then the inner product of the first columns of V_{i+1} and U_i is 1 in absolute value. Hence Definition 2 is consistent with alignment in the 1-dimensional setting from Ji & Telgarsky (2018).

We next define when alignment is an invariant of training for deep linear networks. Again, such invariants are of interest since they may provide insights into properties of trained networks and significantly simplify the dynamics of gradient descent.

Definition 3. ***Alignment is an invariant of training** for a linear neural network f if there exists an initialization $\{W_j^{(0)}\}_{j=1}^d$ such that $W_1^{(\infty)}, W_2^{(\infty)}, \dots, W_d^{(\infty)}$ achieves zero training error [1] and for all gradient descent steps $t \in \mathbb{Z}_{\geq 0}$*

(a) the network f is aligned;

(b) $W_i^{(t)} = U_i \Sigma_i^{(t)} V_i^T$ for all $i \in \{2, \dots d-1\}$, that is, U_i, V_i are not updated;

(c) $W_1^{(t)} = U_1 \Sigma_1^{(t)} {V_1^{(t)}}^T$ and $W_d^{(t)} = U_d^{(t)} \Sigma_d^{(t)} V_d^T$, that is, U_1 and V_d are not updated.

*If additionally, V_1 and U_d are not updated for any $t \in \mathbb{Z}_{\geq 0}$, then we say that **strong alignment is an invariant of training**.*

When alignment is an invariant of training, there are important consequences for training. In particular, note that when the network f is aligned with usSVDs $W_i = U_i \Sigma_i V_i^T$ for all $1 \leq i \leq d$, then

$$f(x) = W_d \cdots W_1 x = U_d \left(\prod_{i=0}^{d-1} \Sigma_{d-i} \right) V_1^T x. \tag{3}$$

[1] The interpolation condition in this definition (i.e., achieving zero training error) is important in ruling out several architectures where the layers are trivially aligned. For example, if all layers are constrained to be diagonal matrices throughout training, then the layers are all trivially aligned, but cannot interpolate datasets where the target is not the product of a diagonal matrix with the input.

Hence if alignment is an invariant of training, then the singular vectors of layers 2 through $d-1$ are never updated and the analysis of gradient descent can be limited to the singular values of the layers and the matrices V_1 and U_d.

Remarks. For the remainder of the paper, we assume that the gradient of the loss function at initialization $\{W_i^{(0)}\}_{i=1}^d$ is non-zero. Otherwise, training with gradient descent would not proceed. We also only consider datasets (X, Y) for which there is a linear network that achieves loss zero. This is consistent with the assumptions in Ji & Telgarsky (2018).

4 Alignment in Fully Connected Networks

In this section, we first characterize when alignment is an invariant of training for fully connected networks (Theorem 1). In particular, we show that this is not the case in general. We then present special classes of problems for which alignment is an invariant of training, namely autoencoding, matrix factorization, and matrix sensing. In contrast, for a linear network with 1-dimensional outputs, we demonstrate that there exists an initialization for which the layers remain aligned throughout training given any dataset and any real-valued loss function. Finally, we discuss various consequences of alignment, including a proof of linear convergence of gradient descent to an interpolating solution.

4.1 Characterization of Alignment with Multi-dimensional Outputs

Theorem 1 is one of our main results and characterizes when alignment is an invariant of training in a fully connected network with multi-dimensional outputs. To simplify notation, we consider the case when the layers are square matrices, i.e. $k_i = k_j$ for all $0 \leq i, j \leq d$. The general result for non-square matrices is provided in Appendix D.

Theorem 1. *Let $f : \mathbb{R}^k \to \mathbb{R}^k$ be a linear fully connected network with $d \geq 3$ square layers of size $k > 1$. Alignment is an invariant of training under the squared loss on a dataset $(X, Y) \in \mathbb{R}^{k \times n} \times \mathbb{R}^{k \times n}$ if and only if there exist orthonormal matrices $U, V \in \mathbb{R}^{k \times k}$ such that $U^T Y X^T V$ and $V^T X X^T V$ are diagonal.*

The full proof of this result is presented in Appendices A-E; here, we provide a proof sketch.

Proof Sketch. The proof essentially follows by induction. For the base case, we initialize the layers $\{W_i\}_{i=1}^d$ to satisfy the conditions for alignment given in Definition 3. Assuming that these conditions hold at gradient descent step t, we prove that they hold at step $t+1$.

After substituting the alignment conditions into the gradient descent update equation for the squared loss at step $t+1$ and cancelling terms, we obtain that alignment is an invariant of training if and only if

$$U_d^{(t)^T} \sum_{k=1}^{n} (y^{(k)} - f(x^{(k)})) x^{(k)^T} V_1^{(t)} \tag{4}$$

is a diagonal matrix. By considering the update for $W_1^{(t)}$ and $W_d^{(t)}$, one sees that alignment implies strong alignment and so U_d, V_1 are also invariant across updates. Thus, let $U_d = U$ and $V_1 = V$. By expanding $f(x^{(k)})$ using equation 3, and considering the update across multiple timesteps, we obtain that the matrix in equation 4 is diagonal if and only if $U^T Y X^T V$ and $V^T X X^T V$ are diagonal. To complete the proof, we show in Appendix D that under strong alignment, gradient descent converges to a solution with zero training error. □

Theorem 1 implies that invariance of alignment throughout training holds only for special classes of problems. In particular, the above implies that alignment is an invariant of training when X and Y have the same right singular vectors, a very special condition on the data. Note that this corresponds to the $\epsilon = 0$ data condition with the initialization considered in Gidel et al. (2019). In Section 6, we also provide empirical support showing that alignment is not an invariant of training for important tasks that violate the data condition presented here, such as multi-class classification.

4.2 Classes of Problems with Alignment

We next discuss classes of problems for which alignment is an invariant of training.

Autoencoding: In the case when $X = Y$, it holds that $U^T Y X^T V = U^T X X^T V$. Taking $U = V$ to be the left singular vectors of X satisfies the conditions of Theorem 1.

Matrix Factorization and Inversion: In the case of matrix factorization, we have that $X = I$. Hence taking U and V to be the left and right singular vectors of Y respectively satisfies the conditions of Theorem 1. For matrix inversion, we have that $Y = I$ and we proceed analogously.

Matrix Sensing. Given pairs of observations $\{(M_i, y_i)\}_{i=1}^n$ with $M_i \in \mathbb{R}^{k\times k}$ and $y_i = \text{Tr}(M_i^T X^*)$ for some unobserved matrix $X^* \in \mathbb{R}^{k\times k}$, gradient descent on $\{W_i\}_{i=1}^n$ is used to solve

$$\arg\min_{\{W_i\}} \frac{1}{2n} \sum_{i=1}^{n} \|y_i - \text{Tr}(M_i^T W_d W_{d-1} \dots W_1)\|_2^2.$$

Implicit regularization in the matrix sensing setting has been analyzed extensively (Arora et al., 2019b; Du et al., 2018; Gunasekar et al., 2017; Li et al., 2018). Theorem 1 shows that alignment is an invariant of training in this setting if and only if $M_i = U\Lambda_i V^T$ for all $i \in [n]$, and $U_d = U, V_1 = V$.

1-dimensional Outputs. In Appendix F, we show that alignment is an invariant of training for fully connected networks with 1-dimensional outputs for any real-valued loss function provided that gradient descent converges to zero training error.

4.3 Consequences of Alignment

We next discuss various consequences of the invariance of alignment for the analysis of training. Our explicit characterization of alignment as an invariant is significant as it allows us to greatly simplify the convergence analysis of gradient descent, which is a main goal of defining an invariant of training.

The following corollary (proof in Appendix B) follows from the proof of Theorem 1, and shows that under alignment the gradient descent update rule is simplified significantly.

Corollary 1. *Let $r = \min(k_0, k_1, \dots, k_d) > 1$ and let the top left $r \times r$ submatrix of $U^T Y X^T V$ be Λ' and that of $V^T X X^T V$ be Λ. Under the invariance of strong alignment (i.e., when Λ' and Λ are diagonal), we can express the partial derivative with respect to W_i as follows:*

$$\frac{\partial L}{\partial W_i} = -\frac{1}{n} U_i \left(\prod_{j=i+1}^{d} {\Sigma_j}^T (U^T Y X^T V_1 - \Sigma_d \cdots \Sigma_1 V^T X X^T V) \prod_{j=1}^{i-1} {\Sigma_j}^T \right) V_i^T. \tag{5}$$

As a result, gradient descent only updates the first r values of Σ_i. Let ${\Sigma'}_i^{(t)}$ be the top left $r \times r$ matrix of $\Sigma_i^{(t)}$. The updates are then given by:

$${\Sigma'}_i^{(t+1)} = {\Sigma'}_i^{(t)} + \frac{\gamma}{n} \prod_{j=1}^{d} {\Sigma'}_j^{(t)} (\Lambda' - \prod_{j=1}^{d} {\Sigma'}_j^{(t)} \Lambda). \tag{6}$$

The other entries of $\Sigma_i^{(t+1)}$ are not updated.

We can use this corollary to provide an explicit learning rate under which gradient descent converges linearly to a global minimum. The proof of the following proposition is given in Appendix C.

Proposition 1. *For $k \in [r]$, let $\sigma_k(W_i)$ denote the kth entry of Σ_i in the usSVD of W_i, and let λ_k, λ'_k denote the kth entries of Λ, Λ' respectively. Under the conditions of Corollary 1 and assuming that $\sigma_k(W_i^{(0)}) > 0$ and $\prod_{i=1}^{d} \sigma_k(W_i^{(0)}) < \frac{\lambda'_k}{\lambda_k}$ for all $k \in [r]$, if the learning rate satisfies $\gamma \leq \frac{n \ln 2}{d} \cdot \min_k \frac{\sigma_k(W_i^{(0)})^2 \lambda_k}{\lambda'^2_k}$ then gradient descent only updates the top r singular values of the solution and converges linearly to the global minimum.*

Remarks. Shamir (2018) shows that for linear neural networks with one-dimensional outputs, the rate of convergence can be exponentially slow in the depth. Proposition 1 shows that for a fixed depth, gradient descent converges linearly. Our analysis in Appendix C shows that our upper bound on the rate of convergence can also grow exponentially in the depth.

Alignment in the Limit of Training. We briefly comment on understanding whether alignment will occur in the limit of training. The following proposition, which states that for a 2-layer network, an aligned solution achieves the minimum ℓ_2-norm. The proof is given in Appendix G.

Proposition 2. *Let W_1, W_2 be matrices such that $W_2 W_1 = P$, for a fixed matrix P. Then, $\|W_1\|_F^2 + \|W_2\|_F^2$ achieves a minimum at the solution where W_1 and W_2 are aligned* and *0-balanced, i.e. there exist usSVD's $W_1 = W\Sigma V^T, W_2 = U\Sigma W^T$.*

It has been shown that SGD in the overparameterized setting for a network initialized close to zero will converge to a solution close in ℓ_2-norm to the minimum ℓ_2-norm solution (Azizan et al., 2019). Therefore we expect such networks to converge to a solution which is close to an aligned solution.

5 Alignment Under General Layer Structure

In the previous section, we analyzed fully connected networks, where parameters of each weight matrix are optimized independently. The most commonly used deep learning models, however, rely on convolutional layers or layers with other forms of constraints. In this section, we analyze alignment in the setting of linear networks with layer constraints. In particular, we show that when the dimension of the subspace induced by the layer constraints is small compared to the number of training samples, alignment cannot happen, let alone be an invariant of training.

5.1 Linear Neural Networks with Layer Structure

We start by setting up mathematical terminology to describe different layer structures.

Definition 4. *Let $\mathcal{S} \subset \mathbb{R}^{m\times n}$ be a linear subspace of matrices and let $\{A_i\}_{i=1}^r$ be an orthogonal*[2] *basis for $\mathcal{S}$. Layer W_i has layer structure $\mathcal{S}$ if $W_i \in \mathcal{S}$, i.e., there exist coefficients $\{c_j^i\}_{j=1}^r \subset \mathbb{R}$ such that $W_i = \sum_{j=1}^r c_j^i A_j$, and gradient descent operates on the $\{c_j^i\}_{i,j=1}^r$.*

Definition 4 encompasses layer structures commonly used in practice, such as:

Convolutional layers: Treating a $p \times p$ image as a vector in $\mathbb{R}^{p^2}$, a single $s \times s$ convolutional filter with stride 1 and padding $(s-1)/2$ maps the image to another $p \times p$ image; this linear transformation is a matrix in $\mathbb{R}^{p^2 \times p^2}$ and the set of all such transformations forms an s^2-dimensional subspace. The parameters of the filter are coefficients of an orthogonal basis of this subspace; see Appendix I.

Layers with Sparse Connections: Consider a fixed connection pattern between layers such that the jth hidden unit in layer i depends only on a subset of units in layer $i-1$. In this case, the subspace $\mathcal{S}$ consists of matrices where particular entries are forced to be zero corresponding to missing connections between features in consecutive layers.

The following theorem provides, in closed-form, the gradient descent update rules for linear networks with layer structure. The proof is provided in Appendix H.

Theorem 2. *Performing gradient descent on the basis coefficients $\{c_j^i\}_{j=1}^r$ leads to the following weight matrix updates:*

$$W_i^{(t+1)} = W_i^{(t)} - \eta \cdot \pi_{\mathcal{S}} \left(\frac{\partial l}{\partial W_i^{(t)}} \right),$$

where $\pi_{\mathcal{S}}$ denotes the projection operator onto $\mathcal{S}$.

Theorem 2 shows that gradient descent in networks with layer structure is equivalent to projected gradient descent[3]. Hence alignment is an invariant of training if and only if it holds throughout the projected gradient descent updates and leads to an aligned solution with zero training loss.

[2] Orthogonality is w.r.t the inner product $\langle A, B\rangle = \mathrm{Tr}(A^T B)$, or equivalently the dot product in $\mathbb{R}^{mn}$

[3] $\pi_{\mathcal{S}}$ is a projection in the traditional sense if and only if the A_j form an orthonormal basis; otherwise, $\pi_{\mathcal{S}}$ is a projection onto $\mathcal{S}$ followed by an appropriate scaling in each basis direction.

5.2 Necessary Condition for Alignment

Motivated by the above characterization via projected gradient descent, we now show that for layer structures with constrained dimension, aligned networks generally cannot achieve zero training error under the squared loss, given sufficient data (Proposition 4). This is the case even when there is a solution with the desired layer structure that achieves zero training error. Hence, if loss is minimized to zero, gradient descent must lead to a non-aligned network.

We first show that for an aligned network which interpolates the data, the first and last layer must align with the pseudoinverse. The proof of this result is presented in Appendix J.

Proposition 3. *Let $(X, Y) \in \mathbb{R}^{k_0 \times n} \times \mathbb{R}^{k_d \times n}$ such that $n \geq k_0$ and X is full-rank (ensuring that XX^T is invertible). If an aligned network $f = W_d W_{d-1} \dots W_1$ achieves zero error under squared loss (i.e. if $Y = f(X)$), then W_d^T aligns with $YX^T(XX^T)^{-1}$, which in turn aligns with W_1^T.*

The following result tells us that when a linear space $\mathcal{S}$ of matrices is sufficiently low-dimensional, the set of matrices that align with an element of $\mathcal{S}$ has measure zero. While we are mainly interested in the setting where $n \geq k$, we state it in full generality using $\binom{m}{2} = 0$, when $m < 2$.

Proposition 4. *Let $\mathcal{S}$ be an r-dimensional linear subspace of $k \times k$ matrices. If $r < k - 1 - \binom{k-n}{2}$ then the set of matrices of size $k \times n$ that can align with an element of $\mathcal{S}$, excluding scalar multiples of the identity, has Lebesgue measure zero.*

The proof of Proposition 4 is provided in Appendix K. Taken together, Propositions 3 and 4 imply Theorem 3, which states that alignment does not occur in linear networks with constrained layer structures given enough training samples. We assume $k = k_0 = \dots = k_d$ and that all layers have the same structure, $\mathcal{S}$. The statement can trivially be extended to the general setting.

Theorem 3. *Let $n \geq k$, let $X, Y \in \mathbb{R}^{k \times n}$ be generic, let $\mathcal{S} \subset \mathbb{R}^{k \times k}$ be a linear subspace of dimension $r < k - 1$, and let $W_1, \dots, W_d \in \mathcal{S}$ such that at least one W_i is not a scalar multiple of the identity[4]. If the network $f = W_d \cdots W_1$ satisfies $Y = f(X)$, then f is not aligned.*

Theorem 3 is in contrast to fully connected networks (i.e., no layer constraints), where we showed that alignment is possible for particular classes of problems including autoencoders. An explicit example of a convolutional linear autoencoder, where alignment is ruled out by Theorem 3, is discussed next.

Example. *If $m \geq 4$, then a generic dataset consisting of $n \geq m^2$ $m \times m$ images cannot be aligned by any convolutional linear autoencoder with filter size 3, aside from the trivial case where all layers are scalar multiples of the identity. This follows from letting $k = m^2$, $r = 9$ in Proposition 4.*

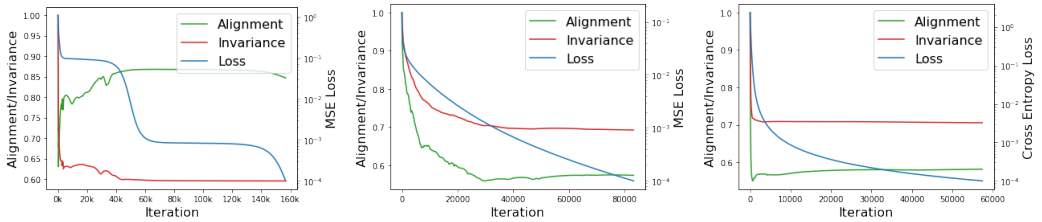
6 Empirical Support

In this section, we provide experimental results to validate our findings[5]. We measure two properties: (1) invariance of alignment from initialization, and (2) alignment between layers. Invariance of alignment at time t is measured by the average dot product between corresponding columns of $U_i^{(t)}$ and $U_i^{(0)}$, as well as $V_i^{(t)}$ and $V_i^{(0)}$. Alignment is measured by the average dot product between corresponding columns of $U_i^{(t)}$ and $V_{i+1}^{(t)}$. For both, a value of 1 is perfect alignment / invariance.

We begin by demonstrating that alignment is not an invariant of training for fully connected networks when the data conditions of Theorem 1 are violated. Figure 1a shows an example where alignment is not an invariant for multi-dimensional regression with random data under squared loss. We used standard normal inputs $X \in \mathbb{R}^{9 \times 9}$ and targets $Y \in \mathbb{R}^{9 \times 9}$, and a 2-hidden layer network initialized so that alignment holds at the start of training. Since X and Y do not have the same right singular vectors, the conditions of Theorem 1 are violated, and hence alignment is not an invariant of training. In Figures 1b and c, we show that alignment is also not an invariant in standard classification settings. We trained a 2-hidden layer fully connected network to classify a linearly separable subset of 256 MNIST examples under MSE loss and cross entropy loss. Figure 1b is consistent with the generalization of

[4]This is not a serious restriction; modulo scalar multiplication, the only case in which such a network could achieve zero loss is autoencoding, in which case the latent space would be a scalar multiple of the data itself.

[5]Hyperparameter settings are detailed in Appendix M

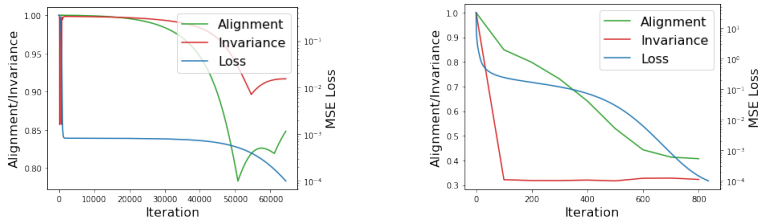


(a) Multi-dimensional regression on random data with squared loss.

(b) Multi-class classification on MNIST with squared loss.

(c) Multi-class classification on MNIST with cross entropy loss.

Figure 1: Examples of fully connected networks with multi-dimensional outputs where alignment is not an invariant of training.



(a) Matrix factorization with layers constrained to be Toeplitz matrices.

(b) Autoencoding a single MNIST example using a convolutional network.

Figure 2: Examples of layer constrained networks, where alignment is not an invariant of training.

Theorem 1 (Appendix D). Interestingly, this result empirically transfers to the case of cross entropy loss, suggesting that our theoretical results may also be relevant for other loss functions.

In networks with constrained layer structure, Theorem 3 shows that given sufficient data alignment cannot occur. We now present empirical evidence that alignment is not an invariant of training, even when the number of training samples is much smaller than the output dimension of the network or the dimensionality of the layer structure is much larger than that of the output. In the setting of matrix factorization ($Y \in \mathbb{R}^{k \times k}$, $X = I$), $k = n$, so Theorem 3 states that alignment is impossible when the linear structure has dimension $r < k - 1$. In Figure 2a, we observe that alignment is not invariant also even when $r \geq k - 1$, by training a 2-hidden layer Toeplitz network to factorize a 4×4 matrix. Our network has 4 hidden units per layer and thus $r = 7, k = 4, n = 4$. Even when $n < r < k$, we observe that alignment is not an invariant. In Figure 2b, we show that alignment is not an invariant of training when autoencoding a single MNIST example using a 2-hidden layer linear convolutional network (i.e. $n = 1, r = 9, k = 784$). In Appendix L, we provide empirical validation that alignment is indeed an invariant of training when the data conditions of Theorem 1 are satisfied.

7 Discussion

We generalized the definition of alignment to linear networks with multi-dimensional outputs. We then analyzed the invariance properties of alignment, showing that under particular data conditions alignment is an invariant for fully connected networks, which allows us to significantly simplify the convergence analysis of gradient descent. We then extended our analysis of alignment to networks with constrained layer structures, such as convolutions, and proved that alignment cannot be an invariant of training in such networks when the dimension of the layer structure r is small compared to the number of training samples n.

While the simplification of gradient descent convergence analysis in the fully connected setting shows that our alignment definition is useful in understanding such networks, the fact that it does not generalize as an invariant to the constrained layer structure setting suggests that other approaches may be necessary to fully understand implicit regularization, such as studying how architecture influences the function classes that can be represented by deep networks (Savarese et al., 2019; Zhang et al., 2020; Radhakrishnan et al., 2019).

REFERENCES

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning (ICML)*, 2018.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A converge analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2019a.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference in Machine Learning (ICML)*, 2019c.

Navid Azizan, Sahin Lale, and Babak Hassibi. Stochastic mirror descent on overparameterized nonlinear models: Convergence, implicit regularization, and generalization. *arXiv preprint arXiv:1906.03830*, 2019.

Rajendra Bhatia. *Matrix Analysis*. Springer-Verlag, 1997.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks . In *International Conference on Learning Representations (ICLR)*, 2019.

Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *Internation Conference on Machine Learning (ICML)*, 2018a.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b.

Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent converges to minimizers. In *Conference on Learning Theory (COLT)*, 2016.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory (COLT)*, 2018.

Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning, 2018. arXiv:1810.01075.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning, 2014. arXiv:1412.6614.

Adityanarayanan Radhakrishnan, Mikhail Belkin, and Caroline Uhler. Memorization in overparameterized autoencoders. In *ICML Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *arXiv preprint arXiv:1902.05040*, 2019.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1820226116. URL `https://www.pnas.org/content/116/23/11537`.

Ohad Shamir. Exponential convergence time of gradient descent for one-dimensional deep linear neural networks. *arXiv preprint arXiv:1809.08587*, 2018.

Daniel Soudry, Elad Hoffer, Mor S. Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research (JMLR)*, 19(1): 2822–2878, 2018.

Xiaoxia Wu, Simon S. Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network? *arXiv preprint arXiv:1902.07111*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, and Yoram Singer. Identity crisis: Memorization and generalization under extreme overparameterization. In *International Conference on Learning Representations (ICLR)*, 2020.

Difan Zou, Yuan Cao Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.