Identity Matters in Deep Learning

Moritz Hardt
Google Brain
1600 Amphitheatre Parkway,
Mountain View, CA, 94043
m@mrtz.org

Tengyu Ma
Department of Computer Sciene
Princeton University
35 Olden Street, Princeton, 08540
tengyu@cs.princeton.edu

Abstract

An emerging design principle in deep learning is that each layer of a deep artificial neural network should be able to easily express the identity transformation. This idea not only motivated various normalization techniques, such as *batch normalization*, but was also key to the immense success of *residual networks*.

In this work, we put the principle of *identity parameterization* on a more solid theoretical footing alongside further empirical progress. We first give a strikingly simple proof that arbitrarily deep linear residual networks have no spurious local optima. The same result for feed-forward networks in their standard parameterization is substantially more delicate. Second, we show that residual networks with ReLu activations have universal finite-sample expressivity in the sense that the network can represent any function of its sample provided that the model has more parameters than the sample size.

Directly inspired by our theory, we experiment with a radically simple residual architecture consisting of only residual convolutional layers and ReLu activations, but no batch normalization, dropout, or max pool. Our model improves significantly on previous all-convolutional networks on the CIFAR10, CIFAR100, and ImageNet classification benchmarks.

1 Introduction

Traditional convolutional neural networks for image classification, such as AlexNet (Krizhevsky et al. (2012)), are parameterized in such a way that when all trainable weights are 0, a convolutional layer represents the 0-mapping. Moreover, the weights are initialized symmetrically around 0. This standard parameterization makes it non-trivial for a convolutional layer trained with stochastic gradient methods to preserve features that were already good. Put differently, such convolutional layers cannot easily converge to the identity transformation at training time.

This shortcoming was observed and partially addressed by Ioffe & Szegedy (2015) through *batch normalization*, i.e., layer-wise whitening of the input with a learned mean and covariance. But the idea remained somewhat implicit until *residual networks* (He et al. (2015); He et al. (2016)) explicitly introduced a reparameterization of the convolutional layers such that when all trainable weights are 0, the layer represents the identity function. Formally, for an input x, each residual layer has the form $x + h(x)$, rather than $h(x)$. This simple reparameterization allows for much deeper architectures largely avoiding the problem of vanishing (or exploding) gradients. Residual networks, and subsequent architectures that use the same parameterization, have since then consistently achieved state-of-the-art results on various computer vision benchmarks such as CIFAR10 and ImageNet.

1.1 Our contributions

In this work, we consider identity parameterizations from a theoretical perspective, while translating some of our theoretical insight back into experiments. Loosely speaking, our first result underlines how identity parameterizations make optimization easier, while our second result shows the same is true for representation.

Linear residual networks. Since general non-linear neural networks, are beyond the reach of current theoretical methods in optimization, we consider the case of deep *linear* networks as a simplified model. A linear network represents an arbitrary linear map as a sequence of matrices $A_\ell \cdots A_2 A_1$. The objective function is $\mathbb{E}\|y - A_\ell \cdots A_1 x\|^2$, where $y = Rx$ for some unknown linear transformation R and x is drawn from a distribution. Such linear networks have been studied actively in recent years as a stepping stone toward the general non-linear case (see Section 1.2). Even though $A_\ell \cdots A_1$ is just a linear map, the optimization problem over the factored variables $(A_\ell, \dots, A_1)$ is non-convex.

In analogy with residual networks, we will instead parameterize the objective function as

$$\min_{A_1,\dots,A_\ell} \mathbb{E}\|y - (I + A_\ell)\cdots(I + A_1)x\|^2 \,. \tag{1.1}$$

To give some intuition, when the depth ℓ is large enough, we can hope that the target function R has a factored representation in which each matrix A_i has small norm. Any symmetric positive semidefinite matrix O can, for example, be written as a product $O = O_\ell \cdots O_1$, where each $O_i = O^{1/\ell}$ is very close to the identity for large ℓ so that $A_i = O_i - I$ has small spectral norm. We first prove that an analogous claim is true for all linear transformations R. Specifically, we prove that for every linear transformation R, there exists a global optimizer $(A_1, \dots, A_\ell)$ of (1.1) such that for large enough depth ℓ,

$$\max_{1\le i\le \ell} \|A_i\| \le O(1/\ell). \tag{1.2}$$

Here, $\|A\|$ denotes the spectral norm of A. The constant factor depends on the conditioning of R. We give the formal statement in Theorem 2.1. The theorem has the interesting consequence that as the depth increases, smaller norm solutions exist and hence regularization may offset the increase in parameters.

Having established the existence of small norm solutions, our main result on linear residual networks shows that the objective function (1.1) is, in fact, easy to optimize when all matrices have sufficiently small norm. More formally, letting $A = (A_1, \dots, A_\ell)$ and $f(A)$ denote the objective function in (1.1), we can show that the gradients of vanish only when $f(A) = 0$ provided that $\max_i \|A_i\| \le O(1/\ell)$. See Theorem 2.2. This result implies that linear residual networks have no *critical points* other than the global optimum. In contrast, for standard linear neural networks we only know, by work of Kawaguchi (2016) that these networks don't have local optima except the global optimum, but it doesn't rule out other critical points. In fact, setting $A_i = 0$ will always lead to a bad critical point in the standard parameterization.

Universal finite sample expressivity. Going back to non-linear residual networks with ReLU activations, we can ask: How expressive are deep neural networks that are solely based on residual layers with ReLU activations? To answer this question, we give a very simple construction showing that such residual networks have perfect finite sample expressivity. In other words, a residual network with ReLU activations can easily express any functions of a sample of size n, provided that it has sufficiently more than n parameters. Note that this requirement is easily met in practice. On CIFAR 10 ($n = 50000$), for example, successful residual networks often have more than 10^6 parameters. More formally, for a data set of size n with r classes, our construction requires $O(n \log n + r^2)$ parameters. Theorem 3.2 gives the formal statement.

Each residual layer in our construction is of the form $x + V\mathrm{ReLU}(Ux)$, where U and V are linear transformations. These layers are significantly simpler than standard residual layers, which typically have two ReLU activations as well as two instances of batch normalization.

The power of all-convolutional residual networks. Directly inspired by the simplicity of our expressivity result, we experiment with a very similar architecture on the CIFAR10, CIFAR100, and ImageNet data sets. Our architecture is merely a chain of convolutional residual layers each with a single ReLU activation, but without batch normalization, dropout, or max pooling as are common in standard architectures. The last layer is a fixed random projection that is not trained. In line with our theory, the convolutional weights are initialized near 0, using Gaussian noise mainly as a symmetry breaker. The only regularizer is standard weight decay (ℓ_2-regularization) and there is no need for dropout. Despite its simplicity, our architecture reaches 6.38% top-1 classification error on the CIFAR10 benchmark (with standard data augmentation). This is competitive with the best residual network reported in He et al. (2015), which achieved 6.43%. Moreover, it improves upon the performance of the previous best *all-convolutional* network, 7.25%, achieved by Springenberg et al. (2014). Unlike ours, this previous all-convolutional architecture additionally required dropout and a non-standard preprocessing (ZCA) of the entire data set. Our architecture also improves significantly upon Springenberg et al. (2014) on both Cifar100 and ImageNet.

1.2 Related Work

Since the advent of residual networks (He et al. (2015); He et al. (2016)), most state-of-the-art networks for image classification have adopted a residual parameterization of the convolutional layers. Further impressive improvements were reported by Huang et al. (2016) with a variant of residual networks, called *dense nets*. Rather than adding the original input to the output of a convolutional layer, these networks preserve the original features directly by concatenation. In doing so, dense nets are also able to easily encode an identity embedding in a higher-dimensional space. It would be interesting to see if our theoretical results also apply to this variant of residual networks.

There has been recent progress on understanding the optimization landscape of neural networks, though a comprehensive answer remains elusive. Experiments in Goodfellow et al. (2014) and Dauphin et al. (2014) suggest that the training objectives have a limited number of bad local minima with large function values. Work by Choromanska et al. (2015) draws an analogy between the optimization landscape of neural nets and that of the spin glass model in physics (Auffinger et al. (2013)). Soudry & Carmon (2016) showed that 2-layer neural networks have no bad *differentiable* local minima, but they didn't prove that a good differentiable local minimum does exist. Baldi & Hornik (1989) and Kawaguchi (2016) show that linear neural networks have no bad local minima. In contrast, we show that the optimization landscape of deep linear residual networks has no bad *critical* point, which is a stronger and more desirable property. Our proof is also notably simpler illustrating the power of re-parametrization for optimization. Our results also indicate that deeper networks may have more desirable optimization landscapes compared with shallower ones.

2 Optimization landscape of linear residual networks

Consider the problem of learning a linear transformation $R\colon \mathbb{R}^d \to \mathbb{R}^d$ from noisy measurements $y = Rx + \xi$, where $\xi \in \mathcal{N}(0, \mathrm{Id}_d)$ is a d-dimensional spherical Gaussian vector. Denoting by $\mathcal{D}$ the distribution of the input data x, let $\Sigma = \mathbb{E}_{x\sim\mathcal{D}}[xx^\top]$ be its covariance matrix.

There are, of course, many ways to solve this classical problem, but our goal is to gain insights into the optimization landscape of neural nets, and in particular, residual networks. We therefore parameterize our learned model by a sequence of weight matrices $A_1, \dots, A_\ell \in \mathbb{R}^{d\times d}$,

$$h_0 = x\,, \qquad h_j = h_{j-1} + A_j h_{j-1}\,, \qquad \hat{y} = h_\ell\,. \tag{2.1}$$

Here $h_1, \dots, h_{\ell-1}$ are the $\ell - 1$ hidden layers and $\hat{y} = h_\ell$ are the predictions of the learned model on input x. More succinctly, we have

$$\hat{y} = (\mathrm{Id}_d + A_\ell)\dots(\mathrm{Id} + A_1)x\,.$$

It is easy to see that this model can express any linear transformation R. We will use A as a shorthand for all of the weight matrices, that is, the $\ell \times d \times d$-dimensional tensor the contains $A_1, \dots, A_\ell$ as slices. Our objective function is the maximum likelihood estimator,

$$f(A,(x,y)) = \|\hat{y} - y\|^2 = \|(\mathrm{Id} + A_\ell)\dots(\mathrm{Id} + A_1)x - Rx - \xi\|^2\,. \tag{2.2}$$

We will analyze the landscape of the *population risk*, defined as,

$$f(A) := \mathbb{E}\left[f(A,(x,y))\right]\,.$$

Recall that $\|A_i\|$ is the spectral norm of A_i. We define the norm $|\!|\!|\cdot|\!|\!|$ for the tensor A as the maximum of the spectral norms of its slices,

$$|\!|\!|A|\!|\!| := \max_{1\le i\le \ell} \|A_i\|\,.$$

The first theorem of this section states that the objective function f has an optimal solution with small $|\!|\!|\cdot|\!|\!|$-norm, which is *inversely* proportional to the number of layers ℓ. Thus, when the architecture is deep, we can shoot for fairly small norm solutions. We define $\gamma := \max\{|\log \sigma_{\max}(R)|, |\log \sigma_{\min}(R)|\}$. Here $\sigma_{\min}(\cdot), \sigma_{\max}(\cdot)$ denote the least and largest singular values of R respectively.

Theorem 2.1. *Suppose $\ell \geq 3\gamma$. Then, there exists a global optimum solution $A^\star$ of the population risk $f(\cdot)$ with norm*

$$|||A^\star||| \leq 2(\sqrt{\pi} + \sqrt{3\gamma})^2/\ell\,.$$

Here γ should be thought of as a constant since if R is too large (or too small), we can scale the data properly so that $\sigma_{\min}(R) \leq 1 \leq \sigma_{\max}(R)$. Concretely, if $\sigma_{\max}(R)/\sigma_{\min}(R) = \kappa$, then we can scaling for the outputs properly so that $\sigma_{\min}(R) = 1/\sqrt{\kappa}$ and $\sigma_{\max}(R) = \sqrt{\kappa}$. In this case, we have $\gamma = \log \sqrt{\kappa}$, which will remain a small constant for fairly large condition number κ. We also point out that we made no attempt to optimize the constant factors here in the analysis. The proof of Theorem 2.1 is rather involved and is deferred to Section A.

Given the observation of Theorem 2.1, we restrict our attention to analyzing the landscape of $f(\cdot)$ in the set of A with $|||\cdot|||$-norm less than τ,

$$\mathcal{B}_\tau = \{A \in \mathbb{R}^{\ell\times d\times d} : |||A||| \leq \tau\}\,.$$

Here using Theorem 2.1, the radius τ should be thought of as on the order of $1/\ell$. Our main theorem in this section claims that there is no bad critical point in the domain $\mathcal{B}_\tau$ for any $\tau < 1$. Recall that a critical point has vanishing gradient.

Theorem 2.2. *For any $\tau < 1$, we have that any critical point A of the objective function $f(\cdot)$ inside the domain $\mathcal{B}_\tau$ must also be a global minimum.*

Theorem 2.2 suggests that it is sufficient for the optimizer to converge to critical points of the population risk, since all the critical points are also global minima.

Moreover, in addition to Theorem 2.2, we also have that any A inside the domain $\mathcal{B}_\tau$ satisfies that

$$\|\nabla f(A)\|_F^2 \geq 4\ell(1-\tau)^{\ell-1}\sigma_{\min}(\Sigma)^2(f(A) - C_{\text{opt}})\,. \tag{2.3}$$

Here C_{opt} is the global minimal value of $f(\cdot)$ and $\|\nabla f(A)\|_F$ denotes the euclidean norm[1] of the $\ell \times d \times d$-dimensional tensor $\nabla f(A)$. Note that $\sigma_{\min}(\Sigma)$ denote the minimum singular value of Σ.

Equation (2.3) says that the gradient has fairly large norm compared to the error, which guarantees convergence of the gradient descent to a global minimum (Karimi et al. (2016)) if the iterates stay inside the domain $\mathcal{B}_\tau$, which is not guaranteed by Theorem 2.2 by itself.

Towards proving Theorem 2.2, we start off with a simple claim that simplifies the population risk. We also use $\|\cdot\|_F$ to denote the Frobenius norm of a matrix.

Claim 2.3. *In the setting of this section, we have,*

$$f(A) = \left\|((\text{Id} + A_\ell)\dots(\text{Id} + A_1) - R)\Sigma^{1/2}\right\|_F^2 + C\,. \tag{2.4}$$

Here C is a constant that doesn't depend on A, and $\Sigma^{1/2}$ denote the square root of Σ, that is, the unique symmetric matrix B that satisfies $B^2 = \Sigma$.

Proof of Claim 2.3. Let $\text{tr}(A)$ denotes the trace of the matrix A. Let $E = (\text{Id}+A_\ell)\dots(\text{Id}+A_1)-R$. Recalling the definition of $f(A)$ and using equation (2.2), we have

$$\begin{aligned}
f(A) &= \mathbb{E}\left[\|Ex - \xi\|^2\right] && \text{(by equation (2.2))}\\
&= \mathbb{E}\left[\|Ex\|^2 + \|\xi\|^2 - 2\langle Ex, \xi\rangle\right]\\
&= \mathbb{E}\left[\text{tr}(Exx^\top E^\top)\right] + \mathbb{E}\left[\|\xi\|^2\right] && \text{(since } \mathbb{E}\left[\langle Ex, \xi\rangle\right] = \mathbb{E}\left[\langle Ex, \mathbb{E}\left[\xi|x\right]\rangle\right] = 0)\\
&= \text{tr}\left(E\,\mathbb{E}\left[xx^\top\right]E^\top\right) + C && \text{(where } C = \mathbb{E}[xx^\top])\\
&= \text{tr}(E\Sigma E^\top) + C = \|E\Sigma^{1/2}\|_F^2 + C\,. && \text{(since } \mathbb{E}\left[xx^\top\right] = \Sigma)
\end{aligned}$$

□

[1] That is, $\|T\|_F := \sqrt{\sum_{ijk} T_{ijk}^2}$.

Next we compute the gradients of the objective function $f(\cdot)$ from straightforward matrix calculus. We defer the full proof to Section A.

Lemma 2.4. *The gradients of $f(\cdot)$ can be written as,*

$$\frac{\partial f}{\partial A_i} = 2(\mathrm{Id} + A_\ell^\top)\dots(\mathrm{Id} + A_{i+1}^\top)E\Sigma(\mathrm{Id} + A_{i-1}^\top)\dots(\mathrm{Id} + A_1^\top)\,, \tag{2.5}$$

where $E = (\mathrm{Id} + A_\ell)\dots(\mathrm{Id} + A_1) - R$.

Now we are ready to prove Theorem 2.2. The key observation is that each matric A_j has small norm and cannot cancel the identity matrix. Therefore, the gradients in equation (2.5) is a product of non-zero matrices, except for the error matrix E. Therefore, if the gradient vanishes, then the only possibility is that the matrix E vanishes, which in turns implies A is an optimal solution.

Proof of Theorem 2.2. Using Lemma 2.4, we have,

$$\begin{aligned}\left\|\frac{\partial f}{\partial A_i}\right\|_F &= 2\left\|(\mathrm{Id} + A_\ell^\top)\dots(\mathrm{Id} + A_{i+1}^\top)E\Sigma(\mathrm{Id} + A_{i-1}^\top)\dots(\mathrm{Id} + A_1^\top)\right\|_F && \text{(by Lemma 2.4)}\\ &\ge 2\prod_{j\neq i}\sigma_{\min}(\mathrm{Id} + A_i^\top)\cdot\sigma_{\min}(\Sigma)\|E\|_F && \text{(by Claim C.2)}\\ &\ge 2(1-\tau)^{\ell-1}\sigma_{\min}(\Sigma)\|E\|\,. && \text{(since } \sigma_{\min}(\mathrm{Id}+A)\ge 1-\|A\|)\end{aligned}$$

It follows that

$$\begin{aligned}\|\nabla f(A)\|_F^2 = \sum_{i=1}^{\ell}\left\|\frac{\partial f}{\partial A_i}\right\|_F^2 &\ge 4\ell(1-\tau)^{\ell-1}\sigma_{\min}(\Sigma)^2\|E\|^2\\ &\ge 4\ell(1-\tau)^{\ell-1}\sigma_{\min}(\Sigma)^2(f(A)-C) && \text{(by the definition of } E \text{ and Claim 2.3)}\\ &\ge 4\ell(1-\tau)^{\ell-1}\sigma_{\min}(\Sigma)^2(f(A)-C_{\mathrm{opt}})\,.\\ & && \text{(since } C_{\mathrm{opt}} = \min_A f(A) \ge C \text{ by Claim 2.3)}\end{aligned}$$

Therefore we complete the proof of equation (2.3). Finally, if A is a critical point, namely, $\nabla f(A) = 0$, then by equation (2.3) we have that $f(A) = C_{\mathrm{opt}}$. That is, A is a global minimum. □

3 Representational Power of the Residual Networks

In this section we characterize the finite-sample expressivity of residual networks. We consider a residual layers with a single ReLU activation and no batch normalization. The basic residual building block is a function $\mathcal{T}_{U,V,s}(\cdot) : \mathbb{R}^k \to \mathbb{R}^k$ that is parameterized by two weight matrices $U \in \mathbb{R}^{\times k}, V \in \mathbb{R}^{k\times k}$ and a bias vector $s \in \mathbb{R}^k$,

$$\mathcal{T}_{U,V,s}(h) = V\mathrm{ReLu}(Uh + s)\,. \tag{3.1}$$

A residual network is composed of a sequence of such residual blocks. In comparison with the full pre-activation architecture in He et al. (2016), we remove two batch normalization layers and one ReLU layer in each building block.

We assume the data has r labels, encoded as r standard basis vectors in $\mathbb{R}^r$, denoted by $e_1,\dots,e_r$. We have n training examples $(x^{(1)}, y^{(1)}),\dots,(x^{(n)}, y^{(n)})$, where $x^{(i)} \in \mathbb{R}^d$ denotes the i-th data and $y^{(i)} \in \{e_1,\dots,e_r\}$ denotes the i-th label. Without loss of generality we assume the data are normalized so that $x^{(i)} = 1$. We also make the mild assumption that no two data points are very close to each other.

Assumption 3.1. *We assume that for every $1 \le i < j \le n$, we have $\|x^{(i)} - x^{(j)}\|^2 \ge \rho$ for some absolute constant $\rho > 0$.*

Images, for example, can always be imperceptibly perturbed in pixel space so as to satisfy this assumption for a small but constant ρ.

Under this mild assumption, we prove that residual networks have the power to express any possible labeling of the data as long as the number of parameters is a logarithmic factor larger than n.

Theorem 3.2. *Suppose the training examples satisfy Assumption 3.1. Then, there exists a residual network N (specified below) with $O(n \log n + r^2)$ parameters that perfectly expresses the training data, i.e., for all $i \in \{1, \dots, n\}$, the network N maps $x^{(i)}$ to $y^{(i)}$.*

It is common in practice that $n > r^2$, as is for example the case for the Imagenet data set where $n > 10^6$ and $r = 1000$.

We construct the following residual net using the building blocks of the form $\mathcal{T}_{U,V,s}$ as defined in equation (3.1). The network consists of $\ell + 1$ hidden layers $h_0, \dots, h_\ell$, and the output is denoted by $\hat{y} \in \mathbb{R}^r$. The first layer of weights matrices A_0 maps the d-dimensional input to a k-dimensional hidden variable h_0. Then we apply ℓ layers of building block $\mathcal{T}$ with weight matrices $A_j, B_j \in \mathbb{R}^{k \times k}$. Finally, we apply another layer to map the hidden variable h_ℓ to the label $\hat{y}$ in $\mathbb{R}^k$. Mathematically, we have

$$\begin{aligned} h_0 &= A_0 x \,, \\ h_j &= h_{j-1} + \mathcal{T}_{A_j, B_j, b_j}(h_{j-1}), \quad \forall j \in \{1, \dots, \ell\} \\ \hat{y} &= h_\ell + \mathcal{T}_{A_{\ell+1}, B_{\ell+1}, s_{\ell+1}}(h_\ell) \,. \end{aligned}$$

We note that here $A_{\ell+1} \in \mathbb{R}^{k \times r}$ and $B_{\ell+1} \in \mathbb{R}^{r \times r}$ so that the dimension is compatible. We assume the number of labels r and the input dimension d are both smaller than n, which is safely true in practical applications.[2] The hyperparameter k will be chosen to be $O(\log n)$ and the number of layers is chosen to be $\ell = \lceil n/k \rceil$. Thus, the first layer has dk parameters, and each of the middle ℓ building blocks contains $2k^2$ parameters and the final building block has $kr + r^2$ parameters. Hence, the total number of parameters is $O(kd + \ell k^2 + rk + r^2) = O(n \log n + r^2)$.

Towards constructing the network N of the form above that fits the data, we first take a random matrix $A_0 \in \mathbb{R}^{k \times d}$ that maps all the data points $x^{(i)}$ to vectors $h_0^{(i)} := A_0 x^{(i)}$. Here we will use $h_j^{(i)}$ to denote the j-th layer of hidden variable of the i-th example. By Johnson-Lindenstrauss Theorem (Johnson & Lindenstrauss (1984), or see Wikipedia (2016)), with good probability, the resulting vectors $h_0^{(i)}$'s remain to satisfy Assumption 3.1 (with slightly different scaling and larger constant ρ), that is, any two vectors $h_0^{(i)}$ and $h_0^{(j)}$ are not very correlated.

Then we construct ℓ middle layers that maps $h_0^{(i)}$ to $h_\ell^{(i)}$ for every $i \in \{1, \dots, n\}$. These vectors $h_\ell^{(i)}$ will clustered into r groups according to the labels, though they are in the $\mathbb{R}^k$ instead of in $\mathbb{R}^r$ as desired. Concretely, we design this cluster centers by picking r random unit vectors $q_1, \dots, q_r$ in $\mathbb{R}^k$. We view them as the surrogate label vectors in dimension k (note that k is potentially much smaller than r). In high dimensions (technically, if $k > 4 \log r$) random unit vectors $q_1, \dots, q_r$ are pair-wise uncorrelated with inner product less than < 0.5. We associate the i-th example with the target surrogate label vector $v^{(i)}$ defined as follows,

$$\text{if } y^{(i)} = e_j, \text{ then } v^{(i)} = q_j \,. \tag{3.2}$$

Then we will construct the matrices $(A_1, B_1), \dots, (A_\ell, B_\ell)$ such that the first ℓ layers of the network maps vector $h_0^{(i)}$ to the surrogate label vector $v^{(i)}$. Mathematically, we will construct $(A_1, B_1), \dots, (A_\ell, B_\ell)$ such that

$$\forall i \in \{1, \dots, n\}, h_\ell^{(i)} = v^{(i)} \,. \tag{3.3}$$

Finally we will construct the last layer $\mathcal{T}_{A_{\ell+1}, B_{\ell+1}, b_{\ell+1}}$ so that it maps the vectors $q_1, \dots, q_r \in \mathbb{R}^k$ to $e_1, \dots, e_r \in \mathbb{R}^r$,

$$\forall j \in \{1, \dots, r\}, q_j + \mathcal{T}_{A_{\ell+1}, B_{\ell+1}, b_{\ell+1}}(q_j) = e_j \,. \tag{3.4}$$

Putting these together, we have that by the definition (3.2) and equation (3.3), for every i, if the label is $y^{(i)}$ is e_j, then $h_\ell^{(i)}$ will be q_j. Then by equation (3.4), we have that $\hat{y}^{(i)} = q_j + \mathcal{T}_{A_{\ell+1}, B_{\ell+1}, b_{\ell+1}}(q_j) = e_j$. Hence we obtain that $\hat{y}^{(i)} = y^{(i)}$.

The key part of this plan is the construction of the middle ℓ layers of weight matrices so that $h_\ell^{(i)} = v^{(i)}$. We encapsulate this into the following informal lemma. The formal statement and the full proof is deferred to Section B.

[2] In computer vision, typically r is less than 10^3 and d is less than 10^5 while n is larger than 10^6

Lemma 3.3 (Informal version of Lemma B.2)**.** *In the setting above, for (almost) arbitrary vectors* $h_0^{(1)}, \dots, h_0^{(n)}$ *and* $v^{(1)}, \dots, v^{(n)} \in \{q_1, \dots, q_r\}$*, there exists weights matrices* $(A_1, B_1), \dots, (A_\ell, B_\ell)$*, such that,*

$$\forall i \in \{1, \dots, n\}, \quad h_\ell^{(i)} = v^{(i)} \,.$$

We briefly sketch the proof of the Lemma to provide intuitions, and defer the full proof to Section B. The operation that each residual block applies to the hidden variable can be abstractly written as,

$$\hat{h} \to h + \mathcal{T}_{U,V,s}(h) \,. \tag{3.5}$$

where h corresponds to the hidden variable before the block and $\hat{h}$ corresponds to that after. We claim that for an (almost) arbitrary sequence of vectors $h^{(1)}, \dots, h^{(n)}$, there exist $\mathcal{T}_{U,V,s}(\cdot)$ such that operation (3.5) transforms k vectors of $h^{(i)}$'s to an arbitrary set of other k vectors that we can freely choose, and maintain the value of the rest of $n - k$ vectors. Concretely, for any subset S of size k, and any desired vector $v^{(i)} (i \in S)$, there exist U, V, s such that

$$\begin{aligned} v^{(i)} &= h^{(i)} + \mathcal{T}_{U,V,s}(h^{(i)}) \;\; \forall i \in S \\ h^{(i)} &= h^{(i)} + \mathcal{T}_{U,V,s}(h^{(i)}) \;\; \forall i \notin S \end{aligned} \tag{3.6}$$

This claim is formalized in Lemma B.1. We can use it repeatedly to construct ℓ layers of building blocks, each of which transforms a subset of k vectors in $\{h_0^{(1)}, \dots, h_0^{(n)}\}$ to the corresponding vectors in $\{v^{(1)}, \dots, v^{(n)}\}$, and maintains the values of the others. Recall that we have $\ell = \lceil n/k \rceil$ layers and therefore after ℓ layers, all the vectors $h_0^{(i)}$'s are transformed to $v^{(i)}$'s, which complete the proof sketch. □

4 Power of all-convolutional residual networks

Inspired by our theory, we experimented with all-convolutional residual networks on standard image classification benchmarks.

4.1 CIFAR10 and CIFAR100

Our architectures for CIFAR10 and CIFAR100 are identical except for the final dimension corresponding to the number of classes 10 and 100, respectively. In Table 1, we outline our architecture. Each *residual block* has the form $x + C_2(\mathrm{ReLU}(C_1 x))$, where C_1, C_2 are convolutions of the specified dimension (kernel width, kernel height, number of input channels, number of output channels). The second convolution in each block always has stride 1, while the first may have stride 2 where indicated. In cases where transformation is not dimensionality-preserving, the original input x is adjusted using averaging pooling and padding as is standard in residual layers.

We trained our models with the Tensorflow framework, using a momentum optimizer with momentum 0.9, and batch size is 128. All convolutional weights are trained with weight decay 0.0001. The initial learning rate is 0.05, which drops by a factor 10 and 30000 and 50000 steps. The model reaches peak performance at around $50k$ steps, which takes about $24h$ on a single NVIDIA Tesla K40 GPU. Our code can be easily derived from an open source implementation[3] by removing batch normalization, adjusting the residual components and model architecture. An important departure from the code is that we initialize a residual convolutional layer of kernel size $k \times k$ and c output channels using a random normal initializer of standard deviation $\sigma = 1/k^2 c$, rather than $1/k\sqrt{c}$ used for standard convolutional layers. This substantially smaller weight initialization helped training, while not affecting representation.

A notable difference from standard models is that the last layer is not trained, but simply a fixed random projection. On the one hand, this slightly improved test error (perhaps due to a regularizing effect). On the other hand, it means that the only trainable weights in our model are those of the convolutions, making our architecture "all-convolutional".

[3] `https://github.com/tensorflow/models/tree/master/resnet`

Table 1: Architecture for CIFAR10/100 (55 convolutions, 13.5M parameters)

variable dimensions	initial stride	description
$3 \times 3 \times 3 \times 16$	1	1 standard conv
$3 \times 3 \times 16 \times 64$	1	9 residual blocks
$3 \times 3 \times 64 \times 128$	2	9 residual blocks
$3 \times 3 \times 128 \times 256$	2	9 residual blocks
–	–	8×8 global average pool
$256 \times$ `num_classes`	–	random projection (not trained)

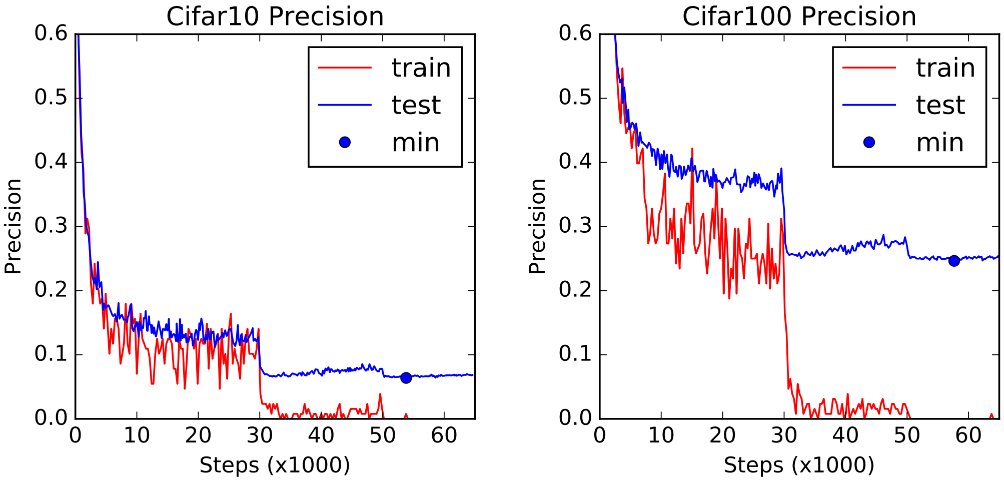

Figure 1: Convergence plots of best model for CIFAR10 (left) and CIFAR (100) right. One step is a gradient update with batch size 128.

An interesting aspect of our model is that despite its massive size of 13.59 million trainable parameters, the model does not seem to overfit too quickly even though the data set size is 50000. In contrast, we found it difficult to train a model with batch normalization of this size without significant overfitting on CIFAR10.

Table 2 summarizes the top-1 classification error of our models compared with a non-exhaustive list of previous works, restricted to the best previous all-convolutional result by Springenberg et al. (2014), the first residual results He et al. (2015), and state-of-the-art results on CIFAR by Huang et al. (2016). All results are with standard data augmentation.

Table 2: Comparison of top-1 classification error on different benchmarks

Method	CIFAR10	CIFAR100	ImageNet	remarks
All-CNN	7.25	32.39	41.2	all-convolutional, dropout, extra data processing
Ours	6.38	24.64	35.29	all-convolutional
ResNet	6.43	25.16	19.38	
DenseNet	3.74	19.25	N/A	

4.2 ImageNet

The ImageNet ILSVRC 2012 data set has $1,281,167$ data points with 1000 classes. Each image is resized to 224×224 pixels with 3 channels. We experimented with an all-convolutional variant of the 34-layer network in He et al. (2015). The original model achieved 25.03% classification error. Our derived model has $35.7M$ trainable parameters. We trained the model with a momentum optimizer (with momentum 0.9) and a learning rate schedule that decays by a factor of 0.94 every two epochs, starting from the initial learning rate 0.1. Training was distributed across 6 machines updating asynchronously. Each machine was equipped with 8 GPUs (NVIDIA Tesla K40) and used batch size 256 split across the 8 GPUs so that each GPU updated with batches of size 32.

In contrast to the situation with CIFAR10 and CIFAR100, on ImageNet our all-convolutional model performed significantly worse than its original counterpart. Specifically, we experienced a significant amount of *underfitting* suggesting that a larger model would likely perform better.

Despite this issue, our model still reached 35.29% top-1 classification error on the test set (50000 data points), and 14.17% top-5 test error after $700,000$ steps (about one week of training). While no longer state-of-the-art, this performance is significantly better than the 40.7% reported by Krizhevsky et al. (2012), as well as the best all-convolutional architecture by Springenberg et al. (2014). We believe it is quite likely that a better learning rate schedule and hyperparameter settings of our model could substantially improve on the preliminary performance reported here.

5 Conclusion

Our theory underlines the importance of identity parameterizations when training deep artificial neural networks. An outstanding open problem is to extend our optimization result to the non-linear case where each residual has a single ReLU activiation as in our expressivity result. We conjecture that a result analogous to Theorem 2.2 is true for the general non-linear case. Unlike with the standard parameterization, we see no fundamental obstacle for such a result.

We hope our theory and experiments together help simplify the state of deep learning by aiming to explain its success with a few fundamental principles, rather than a multitude of tricks that need to be delicately combined. We believe that much of the advances in image recognition can be achieved with residual convolutional layers and ReLU activations alone. This could lead to extremely simple (albeit deep) architectures that match the state-of-the-art on all image classification benchmarks.

References

Antonio Auffinger, Gérard Ben Arous, and Jiří Černý. Random matrices and complexity of spin glasses. *Communications on Pure and Applied Mathematics*, 66(2):165–201, 2013.

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Netw.*, 2(1):53–58, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90014-2. URL http://dx.doi.org/10.1016/0893-6080(89)90014-2.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *ArXiv e-prints*, December 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *arXiv prepring arXiv:1506.01497*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pp. 630–645, 2016. doi: 10.1007/978-3-319-46493-0_38. URL http://dx.doi.org/10.1007/978-3-319-46493-0_38.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 448–456, 2015. URL `http://jmlr.org/proceedings/papers/v37/ioffe15.html`.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

H. Karimi, J. Nutini, and M. Schmidt. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak-\L{}ojasiewicz Condition. *ArXiv e-prints*, August 2016.

K. Kawaguchi. Deep Learning without Poor Local Minima. *ArXiv e-prints*, May 2016.

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

D. Soudry and Y. Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *ArXiv e-prints*, May 2016.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. *ArXiv e-prints*, December 2014.

Eric W. Weisstein. Normal matrix, from mathworld–a wolfram web resource., 2016. URL `http://mathworld.wolfram.com/NormalMatrix.html`.

Wikipedia. Johnsonlindenstrauss lemma — wikipedia, the free encyclopedia, 2016. URL `https://en.wikipedia.org/w/index.php?title=Johnson%E2%80%93Lindenstrauss_lemma&oldid=743553642`.

A Missing Proofs in Section 2

In this section, we give the complete proofs for Theorem 2.1 and Lemma 2.4, which are omitted in Section 2.

A.1 Proof of Theorem 2.1

It turns out the proof will be significantly easier if R is assumed to be a *symmetric positive semidefinite* (PSD) matrix, or if we allow the variables to be complex matrices. Here we first give a proof sketch for the first special case. The readers can skip it and jumps to the full proof below. We will also prove stronger results, namely, $|||A^\star||| \le 3\gamma/\ell$, for the special case.

When R is PSD, it can be diagonalized by orthonormal matrix U in the sense that $R = UZU^\top$, where $Z = \operatorname{diag}(z_1, \dots, z_d)$ is a diagonal matrix with non-negative diagonal entries $z_1, \dots, z_d$. Let $A_1^\star = \cdots = A_\ell^\star = U \operatorname{diag}(z_i^{1/\ell}) U^\top - \mathrm{Id}$, then we have

$$\begin{aligned}(\mathrm{Id} + A_\ell^\star) \cdots (\mathrm{Id} + A_1^\star) &= (U \operatorname{diag}(z_i^{1/\ell}) U^\top)^\ell = U \operatorname{diag}(z_i^{1/\ell})^\ell U && \text{(since } U^\top U = \mathrm{Id}) \\ &= UZU^\top = R\,.\end{aligned}$$

We see that the network defined by $A^\star$ reconstruct the transformation R, and therefore it's a global minimum of the population risk (formally see Claim 2.3 below). Next, we verify that each of the $A_j^\star$ has small spectral norm:

$$\begin{aligned}\|A_j^\star\| &= \|\mathrm{Id} - U \operatorname{diag}(z_i^{1/\ell}) U^\top)\| = \|U(\mathrm{Id} - \operatorname{diag}(z_i)^{1/\ell}) U^\top\| = \|\mathrm{Id} - \operatorname{diag}(z_i)^{1/\ell}\| \\ &\qquad \text{(since } U \text{ is orthonormal)} \\ &= \max_i |z_i^{1/\ell} - 1|\,. \end{aligned} \tag{A.1}$$

Since $\sigma_{\min}(R) \le z_i \le \sigma_{\max}(R)$, we have $\ell \ge 3\gamma \ge |\log z_i|$. It follows that

$$|z_i^{1/\ell} - 1| = |e^{(\log z_i)/\ell} - 1| \le 3|(\log z_i)/\ell| \le 3\gamma/\ell\,. \qquad \text{(since } |e^x - 1| \le 3|x| \text{ for all } |x| \le 1)$$

Then using equation (A.1) and the equation above, we have that $|||A||| \le \max_j \|A_j^\star\| \le 3\gamma/\ell$, which completes the proof for the special case.

Next we give the formal full proof of Theorem 2.1.

Proof of Theorem 2.1. We assume the dimension d is an even number. The odd case has very similar proof and is left to the readers. Let $R = UKV^\top$ be its singular value decomposition, where U,V are two orthonormal matrices and K is a diagonal matrix. Since U is a normal matrix (that is, U satisfies that $UU^\top = U^\top U$), by Claim C.1, we have that U can be block-diagnolaized by orthonormal matrix S into $U = SDS^{-1}$, where $D = \mathrm{diag}(D_1, \dots, D_{d/2})$ is a real block diagonal matrix with each block D_i being of size 2×2.

Since U is orthonormal, U has all its eigenvalues lying on the unit circle (in complex plane). Since D and U are unitarily similar to each other, D also has eigenvalues lying on the unit circle, and so does each of the block D_i. This means that each D_i is a 2×2 dimensional rotation matrix. Each rotation matrix can be written as $T(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$. Suppose $D_i = T(\theta_i)$ where $\theta_i \in [-\pi, \pi]$. Then we have that $D_i = T(\theta_i/q)^q$ for any integer q (that is chosen later). Let $W = \mathrm{diag}(T(\theta_i/q))$. Therefore, it follows that $D = \mathrm{diag}(D_i) = W^q$. Moreover, we have $U = SDS^{-1} = (SWS^{-1})^q$. Therefore, let $B_1 = B_2 = \cdots = B_q = \mathrm{Id} - SWS^{-1}$, then we have $U = (\mathrm{Id} + B_q) \dots (\mathrm{Id} + B_1)$. We verify the spectral norm of these matrices are indeed small,

$$\begin{aligned}
\|B_j\| &= \left\|\mathrm{Id} - SWS^{-1}\right\| = \left\|S(\mathrm{Id} - W)S^{-1}\right\| \\
&= \|\mathrm{Id} - W\| && \text{(since S is unitary)} \\
&= \max_{i \in [d/2]} \|T(0) - T(\theta_i/q)\| && \text{(since $W = \mathrm{diag}(T(\theta_i/q))$ is block diagonal)} \\
&= \max |\sin(\theta_i/q)| \le \pi/q\,.
\end{aligned}$$

Similarly, we can choose $B_1', \dots, B_q'$ with $\|C_j\| \le \pi/q$ so that $V^\top = (\mathrm{Id} + B_q') \dots (\mathrm{Id} + B_1')$.

Last, we deal with the diagonal matrix K. Let $K = \mathrm{diag}(k_i)$. We have $\min k_i = \sigma_{\min}(R), \max k_i = \sigma_{\max}(R)$. Then, we can write $K = (K')^p$ where $K' = \mathrm{diag}(k_i^{1/p})$ and p is an integer to be chosen later. We have that $\|K' - \mathrm{Id}\| \le \max |k_i^{1/p} - 1| \le \max |e^{\log k_i \cdot 1/p} - 1|$. When $p \ge \gamma = \max\{\log \max k_i, -\log \min k_i\} = \max\{\log \sigma_{\max}(R), -\log \sigma_{\min}(R)\}$, we have that

$$\|K' - \mathrm{Id}\| \le \max |e^{\log k_i \cdot 1/p} - 1| \le 3 \max |\log k_i \cdot 1/p| = 3\gamma/p\,.$$
$$\text{(since } |e^x - 1| \le 3|x| \text{ for } |x| \le 1)$$

Let $B_1'' = \cdots = B_p'' = K' - \mathrm{Id}$ and then we have $K = (\mathrm{Id} + B_p'') \cdots (\mathrm{Id} + B_1'')$. Finally, we choose $p = \frac{\ell\sqrt{3\gamma}}{2(\sqrt{\pi}+\sqrt{3\gamma})}$ and $q = \frac{\ell\sqrt{\pi}}{\sqrt{\pi}+\sqrt{3\gamma}}$, [4]and let $A_{2p+q} = B_q, \cdots = A_{p+q+1} = B_1, A_{p+q} = B_p'', \dots, A_{q+1} = B_1'', A_q = B_q', \dots, A_1 = B_1'$. We have that $2q + \ell = 1$ and

$$R = UKV^\top = (\mathrm{Id} + A_\ell) \dots (\mathrm{Id} + A_1)\,.$$

Moreover, we have $|||A||| \le \max\{\|B_j\|, \|B_j'\|.\|B_j''\|\} \le \pi/q + 3\gamma/p \le 2(\sqrt{\pi} + \sqrt{3\gamma})^2/\ell$, as desired. □

[4] Here for notational convenience, p, q are not chosen to be integers. But rounding them to closest integer will change final bound of the norm by small constant factor.

A.2 PROOF OF LEMMA 2.4

We compute the partial gradients by definition. Let $\Delta_j \in \mathbb{R}^{d\times d}$ be an infinitesimal change to A_j. Using Claim 2.3, consider the Taylor expansion of $f(A_1,\dots,A_\ell+\Delta_j,\dots,A_\ell)$

$$\begin{aligned}
&f(A_1,\dots,A_\ell+\Delta_j,\dots,A_\ell)\\
&=\left\|((\mathrm{Id}+A_\ell)\cdots(\mathrm{Id}+A_j+\Delta_j)\dots(\mathrm{Id}+A_1)-R)\Sigma^{1/2}\right\|_F^2\\
&=\left\|((\mathrm{Id}+A_\ell)\cdots(\mathrm{Id}+A_1)-R)\Sigma^{1/2}+(\mathrm{Id}+A_\ell)\cdots\Delta_j\dots(\mathrm{Id}+A_1)\Sigma^{1/2}\right\|_F^2\\
&=\left\|(\mathrm{Id}+A_\ell)\cdots(\mathrm{Id}+A_1)-R)\Sigma^{1/2}\right\|_F^2+\\
&\quad 2\langle((\mathrm{Id}+A_\ell)\cdots(\mathrm{Id}+A_1)-R)\Sigma^{1/2},(\mathrm{Id}+A_\ell)\cdots\Delta_j\dots(\mathrm{Id}+A_1)\Sigma^{1/2}\rangle+O(\|\Delta_j\|_F^2)\\
&=f(A)+2\langle(\mathrm{Id}+A_\ell^\top)\dots(\mathrm{Id}+A_{j+1}^\top)E\Sigma(\mathrm{Id}+A_{j-1}^\top)\dots(\mathrm{Id}+A_1^\top),\Delta_j\rangle+O(\|\Delta_j\|_F^2)\,.
\end{aligned}$$

By definition, this means that the $\frac{\partial f}{\partial A_j}=2(\mathrm{Id}+A_\ell^\top)\dots(\mathrm{Id}+A_{j+1}^\top)E\Sigma(\mathrm{Id}+A_{j-1}^\top)\dots(\mathrm{Id}+A_1^\top)$. □

B MISSING PROOFS IN SECTION 3

In this section, we provide the full proof of Theorem 3.2. We start with the following Lemma that constructs a building block $\mathcal{T}$ that transform k vectors of an arbitrary sequence of n vectors to any arbitrary set of vectors, and main the value of the others. For better abstraction we use $\alpha^{(i)}$,$\beta^{(i)}$ to denote the sequence of vectors.

Lemma B.1. *Let $S\subset[n]$ be of size k. Suppose $\alpha^{(1)},\dots,\alpha^{(n)}$ is a sequences of n vectors satisfying a) for every $1\le i\le n$, we have $1-\rho'\le\|\alpha_i\|^2\le 1+\rho'$, and b) if $i\neq j$ and S contains at least one of i,j, then $\|\alpha^{(i)}-\beta^{(j)}\|\ge 3\rho'$. Let $\beta^{(1)},\dots,\beta^{(n)}$ be an arbitrary sequence of vectors. Then, there exists $U,V\in\mathbb{R}^{k\times k}$, s such that for every $i\in S$, we have $\mathcal{T}_{U,V,s}(\alpha^{(i)})=\beta^{(i)}-\alpha^{(i)}$, and moreover, for every $i\in[n]\backslash S$ we have $\mathcal{T}_{U,V,s}(\alpha^{(i)})=0$.*

We can see that the conclusion implies

$$\begin{aligned}
\beta^{(i)}&=\alpha^{(i)}+\mathcal{T}_{U,V,s}(\alpha^{(i)})\ \ \forall i\in S\\
\alpha^{(i)}&=\alpha^{(i)}+\mathcal{T}_{U,V,s}(\alpha^{(i)})\ \ \forall i\notin S
\end{aligned}$$

which is a different way of writing equation (3.6).

Proof of Lemma B.1. Without loss of generality, suppose $S=\{1,\dots,k\}$. We construct U,V,s as follows. Let the i-th row of U be $\alpha^{(i)}$ for $i\in[k]$, and let $s=-(1-2\rho')\cdot\mathbf{1}$ where $\mathbf{1}$ denotes the all 1's vector. Let the i-column of V be $\frac{1}{\|\alpha^{(i)}\|^2-(1-2\rho')}(\beta^{(i)}-\alpha^{(i)})$ for $i\in[k]$.

Next we verify that the correctness of the construction. We first consider $1\le i\le k$. We have that $U\alpha^{(i)}$ is a a vector with i-th coordinate equal to $\|\alpha^{(i)}\|^2\ge 1-\rho'$. The j-th coordinate of $U\alpha^{(i)}$ is equal to $\langle\alpha^{(j)},\alpha^{(i)}\rangle$, which can be upperbounded using the assumption of the Lemma by

$$\langle\alpha^{(j)},\alpha^{(i)}\rangle=\frac{1}{2}\left(\|\alpha^{(i)}\|^2+\|\alpha^{(j)}\|^2\right)-\|\alpha^{(i)}-\alpha^{(j)}\|^2\le 1+\rho'-3\rho'\le 1-2\rho'\,. \tag{B.1}$$

Therefore, this means $U\alpha^{(i)}-(1-2\rho')\cdot\mathbf{1}$contains a single positive entry (with value at least $\|\alpha^{(i)}\|^2-(1-2\rho')\ge\rho'$), and all other entries being non-positive. This means that $\mathrm{ReLu}(U\alpha^{(i)}+b)=\left(\|\alpha^{(i)}\|^2-(1-2\rho')\right)e_i$ where e_i is the i-th natural basis vector. It follows that $V\mathrm{ReLu}(U\alpha^{(i)}+b)=(\|\alpha^{(i)}\|^2-(1-2\rho'))Ve_i=\beta^{(i)}-\alpha^{(i)}$.

Finally, consider $n\ge i>k$. Then similarly to the computation in equation (B.1), $U\alpha^{(i)}$ is a vector with all coordinates less than $1-2\rho'$. Therefore $U\alpha^{(i)}+b$ is a vector with negative entries. Hence we have $\mathrm{ReLu}(U\alpha^{(i)}+b)=0$, which implies $V\mathrm{ReLu}(U\alpha^{(i)}+b)=0$. □

Now we are ready to state the formal version of Lemma 3.3.

Lemma B.2. *Suppose a sequence of n vectors $z^{(1)}, \dots, z^{(n)}$ satisfies a relaxed version of Assumption 3.1: a) for every i, $1 - \rho' \le \|z^{(i)}\|^2 \le 1 + \rho'$ b) for every $i \neq j$, we have $\|z^{(i)} - z^{(j)}\|^2 \ge \rho'$;. Let $v^{(1)}, \dots, v^{(n)}$ be defined above. Then there exists weigh matrices $(A_1, B_1), \dots, (A_\ell, B_\ell)$, such that given $\forall i, h_0^{(i)} = z^{(i)}$, we have,*

$$\forall i \in \{1, \dots, n\}, \quad h_\ell^{(i)} = v^{(i)} \,.$$

We will use Lemma B.1 repeatedly to construct building blocks $\mathcal{T}_{A_j, B_k, s_j}(\cdot)$, and thus prove Lemma B.2. Each building block $\mathcal{T}_{A_j, B_k, s_j}(\cdot)$ takes a subset of k vectors among $\{z^{(1)}, \dots, z^{(n)}\}$ and convert them to $v^{(i)}$'s, while maintaining all other vectors as fixed. Since they are totally n/k layers, we finally maps all the $z^{(i)}$'s to the target vectors $v^{(i)}$'s.

Proof of Lemma B.2. We use Lemma B.1 repeatedly. Let $S_1 = [1, \dots, k]$. Then using Lemma B.1 with $\alpha^{(i)} = z^{(i)}$ and $\beta^{(i)} = v^{(i)}$ for $i \in [n]$, we obtain that there exists A_1, B_1, b_1 such that for $i \le k$, it holds that $h_1^{(i)} = z^{(i)} + \mathcal{T}_{A_1, B_1, b_1}(z^{(i)}) = v^{(i)}$, and for $i \ge k$, it holds that $h_1^{(i)} = z^{(i)} + \mathcal{T}_{A_1, B_1, b_1}(z^{(i)}) = z^{(i)}$.

Now we construct the other layers inductively. We will construct the layers such that the hidden variable at layer j satisfies $h_j^{(i)} = v^{(i)}$ for every $1 \le i \le jk$, and $h_j^{(i)} = z^{(i)}$ for every $n \ge i > jk$. Assume that we have constructed the first j layer and next we use Lemma B.1 to construct the $j+1$ layer. Then we argue that the choice of $\alpha^{(1)} = v^{(1)}, \dots, \alpha^{(jk)} = v^{(jk)}$, $\alpha^{(jk+1)} = z^{(jk+1)}, \dots, \alpha^{(n)} = z^{(n)}$, and $S = \{jk+1, \dots, (j+1)k\}$ satisfies the assumption of Lemma B.1. Indeed, because q_i's are chosen uniformly randomly, we have w.h.p for every s and i, $\langle q_s, z^{(i)} \rangle \le 1 - \rho'$. Thus, since $v^{(i)} \in \{q_1, \dots, q_r\}$, we have that $v^{(i)}$ also doesn't correlate with any of the $z^{(i)}$. Then we apply Lemma B.1 and conclude that there exists $A_{j+1} = U, B_{j+1} = V, b_{j+1} = s$ such that $\mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(v^{(i)}) = 0$ for $i \le jk$, $\mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(z^{(i)}) = v^{(i)} - z^{(i)}$ for $jk < i \le (j+1)k$, and $\mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(z^{(i)}) = 0$ for $n \ge i > (j+1)k$. These imply that

$$\begin{aligned}
h_{j+1}^{(i)} &= h_j^{(i)} + \mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(v^{(i)}) = v^{(i)} \quad \forall 1 \le i \le jk \\
h_{j+1}^{(i)} &= h_j^{(i)} + \mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(z^{(i)}) = v^{(i)} \quad \forall jk + 1 \le i \le (j+1)k \\
h_{j+1}^{(i)} &= h_j^{(i)} + \mathcal{T}_{A_{j+1}, b_{j+1}, b_{j+1}}(z^{(i)}) = z^{(i)} \quad \forall (j+1)k < i \le n
\end{aligned}$$

Therefore we constructed the $j+1$ layers that meets the inductive hypothesis for layer $j+1$. Therefore, by induction we get all the layers, and the last layer satisfies that $h_\ell^{(i)} = v^{(i)}$ for every example i. □

Now we ready to prove Theorem 3.2, following the general plan sketched in Section 3.

Proof of Theorem 3.2. We use formalize the intuition discussed below Theorem 3.2. First, take $k = c(\log n)/\rho^2$ for sufficiently large absolute constant c (for example, $c = 10$ works), by Johnson-Lindenstrauss Theorem (Johnson & Lindenstrauss (1984), or see Wikipedia (2016)) we have that when A_0 is a random matrix with standard normal entires, with high probability, all the pairwise distance between the the set of vectors $\{0, x^{(1)}, \dots, x^{(n)}\}$ are preserved up to $1 \pm \rho/3$ factor. That is, we have that for every i, $1 - \rho/3 \le \|A_0 x^{(i)}\| \le 1 + \rho/3$, and for every $i \neq j$, $\|A_0 x^{(i)} - A_0 x^{(j)}\| \ge \rho(1 - \rho/3) \ge 2\rho/3$. Let $z^{(i)} = A_0 x^{(i)}$ and $\rho' = \rho/3$. Then we have $z^{(i)}$'s satisfy the condition of Lemam B.2. We pick r random vectors $q_1, \dots, q_r$ in $\mathbb{R}^k$. Let $v^{(1)}, \dots, v^{(n)}$ be defined as in equation (3.2). Then by Lemma B.2, we can construct matrices $(A_1, B_1), \dots, (A_\ell, B_\ell)$ such that

$$h_\ell^{(i)} = v^{(i)} \,. \tag{B.2}$$

Note that $v^{(i)} \in \{q_1, \dots, q_r\}$, and q_i's are random unit vector. Therefore, the choice of $\alpha^{(1)} = q_1, \dots, \alpha^{(r)} = q_r$, $\beta^{(1)} = e_1, \dots, \beta^{(r)} = e_r$, and satisfies the condition of Lemma B.1, and using Lemma B.1 we conclude that there exists $A_{\ell+1}, B_{\ell+1}, s_{\ell+1}$ such that

$$e_j = v_j + \mathcal{T}_{A_{\ell+1}, B_{\ell+1}, b_{\ell+1}}(v_j), \text{ for every } j \in \{1, \dots, r\} \,.. \tag{B.3}$$

By the definition of $v^{(i)}$ in equation (3.2) and equation (B.2), we conclude that $\hat{y}^{(i)} = h_\ell^{(i)} + \mathcal{T}_{A_{\ell+1},B_{\ell+1},b_{\ell+1}}(h_\ell^{(i)}) = y^{(i)}$., which complete the proof. □

C Toolbox

In this section, we state two folklore linear algebra statements. The following Claim should be known, but we can't find it in the literature. We provide the proof here for completeness.

Claim C.1. *Let $U \in \mathbb{R}^{d\times d}$ be a real normal matrix (that is, it satisfies $UU^\top = U^\top U$). Then, there exists an orthonormal matrix $S \in \mathbb{R}^{d\times d}$ such that*

$$U = SDS^\top ,$$

where D is a real block diagonal matrix that consists of blocks with size at most 2×2. Moreover, if d is even, then D consists of blocks with size exactly 2×2.

Proof. Since U is a normal matrix, it is unitarily diagonalizable (see Weisstein (2016) for backgrounds). Therefore, there exists unitary matrix V in $\mathbb{C}^{d\times d}$ and diagonal matrix in $\mathbb{C}^{d\times d}$ such that U has eigen-decomposition $U = V\Lambda V^*$. Since U itself is a real matrix, we have that the eigenvalues (the diagonal entries of Λ) come as conjugate pairs, and so do the eigenvectors (which are the columns of V). That is, we can group the columns of V into pairs $(v_1, \bar{v}_1), \dots, (v_s, \bar{v}_s), v_{s+1}, \dots, v_t$, and let the corresponding eigenvalues be $\lambda_1, \bar{\lambda}_1, \dots, \lambda_{\lambda_s}, \bar{\lambda}_s, \lambda_{s+1}, \dots, \lambda_t$. Here $\lambda_{s+1}, \dots, \lambda_t \in \mathbb{R}$. Then we get that $U = \sum_{i=1}^s 2\Re(v_i\lambda_i v_i^*) + \sum_{i=s+1}^t v_i\lambda_i v_i^\top$. Let $Q_i = \Re(v_i\lambda_i v_i^*)$, then we have that Q_i is a real matrix of rank-2. Let $S_i \in \mathbb{R}^{d\times 2}$ be a orthonormal basis of the column span of Q_i and then we have that Q_i can be written as $Q_i = S_i D_i S_i^\top$ where D_i is a 2×2 matrix. Finally, let $S = [S_1, \dots, S_s, v_{s+1}, \dots, v_t]$, and $D = \mathrm{diag}(D_1, \dots, D_s, \lambda_{s+1}, \dots, \lambda_t)$ we complete the proof. □

The following Claim is used in the proof of Theorem 2.2. We provide a proof here for completeness.

Claim C.2 (folklore)**.** *For any two matrices $A, B \in \mathbb{R}^{d\times d}$, we have that*

$$\|AB\|_F \geq \sigma_{\min}(A)\|B\|_F .$$

Proof. Since $\sigma_{\min}(A)^2$ is the smallest eigenvalue of $A^\top A$, we have that

$$B^\top A^\top AB \succeq B^\top \cdot \sigma_{\min}(A)^2 \mathrm{Id} \cdot B .$$

Therefore, it follows that

$$\begin{aligned} \|AB\|_F^2 &= \mathrm{tr}(B^\top A^\top AB) \geq \mathrm{tr}(B^\top \cdot \sigma_{\min}(A)^2 \mathrm{Id} \cdot B) \\ &= \sigma_{\min}(A)^2 \mathrm{tr}(B^\top B) = \sigma_{\min}(A)^2 \|B\|_F^2 . \end{aligned}$$

Taking square root of both sides completes the proof. □