Measurements of Three-Level Hierarchical Structure in the Outliers in the Spectrum of Deepnet Hessians

Vardan Papyan [1]

Abstract

We expose a structure in deep classifying neural networks in the derivative of the logits with respect to the parameters of the model, which is used to explain the existence of outliers in the spectrum of the Hessian. Previous works decomposed the Hessian into two components, attributing the outliers to one of them, the so-called Covariance of gradients. We show this term is not a Covariance but a second moment matrix, i.e., it is influenced by means of gradients. These means possess an additive two-way structure that is the source of the outliers in the spectrum. This structure can be used to approximate the principal subspace of the Hessian using certain "averaging" operations, avoiding the need for high-dimensional eigenanalysis. We corroborate this claim across different datasets, architectures and sample sizes.

1. Introduction

We consider a C-class classification problem. We are given a sample of n training examples, n_c in each class, $\bigcup_{c=1}^{C}\{(x_{i,c}, y_c)\}_{i=1}^{n_c}$, where $x_{i,c}$ is the i-th example in the c-th class and y_c is its corresponding one-hot vector. The goal is to predict the labels of unseen data based on the limited examples provided for training. State-of-the-art methods fit a deep neural network, parameterized by a vector of parameters $\theta \in \mathbb{R}^p$, to the training data by minimizing the empirical loss

$$\mathcal{L}(\theta) = \underset{i,c}{\text{Ave}}\{\ell(f(x_{i,c};\theta), y_c)\}, \tag{1}$$

averaged across the training data through the operator $\text{Ave}_{i,c}$. Here, $f(x_{i,c};\theta) \in \mathbb{R}^C$ are the logits – the output of the classifier prior to the softmax layer – while

[1]Department of Statistics, Stanford University, Stanford, CA 94305. Correspondence to: Vardan Papyan <papyan@stanford.edu>.

$\ell(f(x_{i,c};\theta), y_c) \in \mathbb{R}^+$ is the cross-entropy loss between the softmax of $f(x_{i,c};\theta)$ and the one-hot vector y_c.

In this work, we investigate the Hessian of the training loss, given by

$$\text{Hess}(\theta) = \underset{i,c}{\text{Ave}}\left\{\frac{\partial^2 \ell(f(x_{i,c};\theta), y_c)}{\partial \theta^2}\right\}. \tag{2}$$

Using the Gauss-Newton decomposition, the above can be written as a summation of two components

$$\text{Hess}(\theta) \tag{3}$$

$$= \underset{i,c}{\text{Ave}}\left\{\underbrace{\sum_{c'=1}^{C} \frac{\partial \ell(z, y_c)}{\partial z_{c'}}\bigg|_{z=f(x_{i,c};\theta)} \frac{\partial^2 f_{c'}(x_{i,c};\theta)}{\partial \theta^2}}_{H}\right\}$$

$$+ \underset{i,c}{\text{Ave}}\left\{\underbrace{\frac{\partial f(x_{i,c};\theta)}{\partial \theta}^T \frac{\partial^2 \ell(z, y_c)}{\partial z^2}\bigg|_{z=f(x_{i,c};\theta)} \frac{\partial f(x_{i,c};\theta)}{\partial \theta}}_{G}\right\},$$

where $f_{c'}(x_{i,c};\theta)$ is the value in the c'-th coordinate of the logits $f(x_{i,c};\theta)$ (similarly for $z_{c'}$). In what follows, we refer to c' as a *logit coordinate*.

Many works studied the Hessian over the years, both from the theoretical and practical point of view (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Chaudhari et al., 2016; Dinh et al., 2017; Hoffer et al., 2017; Pennington & Bahri, 2017; Pennington & Worah, 2018; Jastrzebski et al., 2018; Yaida, 2018; Geiger et al., 2018; Spigler et al., 2018). Of particular relevance to us are two recent works that studied the spectrum of the Hessian. In (Sagun et al., 2016; 2017) the authors showed on small-scale networks that the spectrum exhibits a 'spiked' behavior, with C outliers isolated from a continuous bulk. In (Papyan, 2018), the authors corroborated these findings on modern deepnets with tens of millions of parameters, by applying state-of-the-art tools in modern high-dimensional numerical linear algebra to approximate the full spectrum of the Hessian. They showed that the the outliers can be attributed to the G component, while the majority of the energy in the bulk can be attributed to the H component.

In this work, our goal is to shed light on what is the origin of the outliers observed in G. We provide two motivations for this question:

1. In (Gur-Ari et al., 2018) the authors analyzed the dynamics of stochastic gradient descent (SGD) *as a function of epochs*. They observed that the gradients of SGD live in a small subspace of rank C, spanned by the top eigenvectors of the Hessian, and remarked that utilizing this low-dimensional eigenspace could lead to optimization benefits. These same top eigenvectors of the Hessian were attributed to the G term in (Papyan, 2018).

   In this paper we show that these outliers are caused by a certain structure in the data underlying G. Once this underlying structure is known, we can efficiently compute approximations to the principal subspace. The necessary computations are much simpler even than the power method.

2. In (Papyan, 2018) the authors initiated the investigation of the separation of the top outliers from the bulk *as a function of sample size*.

   In this work we make progress on this question directly by investigating the dynamics of the outliers as a function of sample size. We explain the structure causing the outliers and predict their size without performing eigenanalysis, but rather averaging certain quantities. This provides an alternative to eigenanalysis, which might be easier to analyze and might have better theoretical properties.

3. Previous works alluded to a relation between the generalization of deepnets and the curvature of the loss (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016; Chaudhari et al., 2016; Jastrzebski et al., 2017), usually tied to the spectral norm of the Hessian. In (Papyan, 2018) the authors showed that the spectrum of modern deepnet Hessians consists of outliers and a bulk. This implies: 1) the spectral norm is equal to the maximal outlier; 2) when previous works suggested the spectral norm is related to generalization, they were implicitly claiming the outliers are related to generalization; 3) predicting generalization using the spectral norm requires predicting the location of the outliers, which is the main goal of our work.

1.1. Contributions

We commence this work with the observation that G is a second moment matrix and not a Covariance – the difference between the two being that in the latter a mean is not subtracted from each sample. The aforementioned outliers are a direct sequence of this lack of centering operation and can be computed from the means not being subtracted.
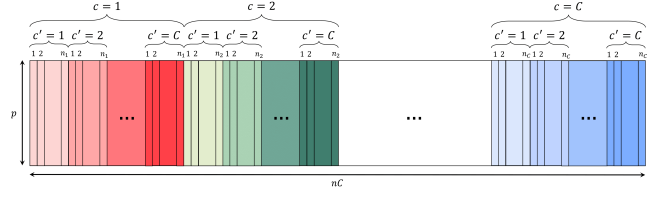

Figure 1. Partitioning of the columns in $\Delta \in \mathbb{R}^{p \times nC}$. The columns can be indexed by three indices, (i, c, c'): i corresponds to the index of a sample in a certain class, c corresponds to the class, and c' corresponds to a logit coordinate. Given this indexing, each column is denoted by $\delta_{i,c,c'}$.

We then show that $G = \frac{1}{n}\Delta\Delta^T$. The rows of $\Delta \in \mathbb{R}^{p \times nC}$ correspond to the coordinates in the space of model parameters and the columns of Δ can be indexed by three indices, (i, c, c'): i corresponds to the index of a sample in a certain class, c corresponds to the class, and c' corresponds to a logit coordinate. Given this indexing, each column in Δ can be denoted by $\delta_{i,c,c'}$. The i-th sample in the c-th class has C columns in Δ associated with it, $\{\delta_{i,c,c'}\}_{c'}$. These correspond to the C logit coordinates. We depict the matrix Δ and its partitioning in Figure 1.

This indexing naturally partitions the columns in Δ into C^2 groups – one for each combination of class c and logit coordinate c'. Each of these groups can be characterized by a *group* mean $\delta_{c,c'}$ and a Covariance $\Sigma_{c,c'}$, which are computed from of all the columns that fall into it, $\{\delta_{i,c,c'}\}_i$. The collection of all group means $\{\delta_{c,c'}\}_{c' \neq c}$ associated with the same class c, but different logit coordinates c', can be considered a *cluster*, characterized by its mean δ_c and Covariance Σ_c.

Intuitively, we think of $\{\delta_{i,c,c'}\}_i$ as members of a *group* with a group mean $\delta_{c,c'}$ and Covariance $\Sigma_{c,c'}$. Moreover, we think of the group means $\{\delta_{c,c'}\}_{c' \neq c}$ as being themselves members of a *cluster* with a cluster center δ_c and Covariance Σ_c. Figure 2 illustrates this intuition while summarizing the above-mentioned definitions. This figure also defines other objects $(G_0, \ldots, G_3)$ that will be introduced in the next sections.

Our main finding in this work is that the top-C outliers in the spectrum of G can be approximated from the eigenvalues of the matrix $G_1 = \text{Ave}_c\{\delta_c\delta_c^T\}$. Equally, these could be approximated from the Gram of cluster centers $\{\delta_c\}_c$.

We show that the cluster centers $\{\delta_c\}_c$ are far apart and the cluster members $\{\delta_{c,c'}\}_{c' \neq c}$ are tightly scattered around the cluster center. In other words, the *within-cluster variation* is small compared to the *between-cluster variation*. This configuration makes the outliers in G attributable to the Gram of the cluster centers. We illustrate this in Figures 3 and 4, showing t-SNE (Maaten & Hinton, 2008) plots of the
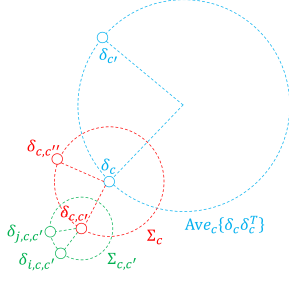
*Figure 2. Three-level hierarchical decomposition of the second moment matrix $G = G_0 + G_1 + G_2 + G_3$. The coarsest level, depicted in **blue**, is comprised of $\{\delta_c\}_c$, whose second moment is given by $G_1 = \text{Ave}_c\{\delta_c\delta_c^T\}$. The top-$C$ outliers observed in the spectrum of G are due to eigenvalues of this matrix. The middle level, depicted in **red**, is composed of $\{\delta_{c,c'}\}_{c,c'}$. For a certain c, $\{\delta_{c,c'}\}_{c'\neq c}$ are centered around δ_c and their Covariance is Σ_c. Averaging this Covariance over all classes results in $G_2 = \text{Ave}_c\{\Sigma_c\}$. The finest level, depicted in **green**, includes all the logit derivatives $\{\delta_{i,c,c'}\}_i$. For a given pair of c and c', $\{\delta_{i,c,c'}\}_i$ are centered around $\delta_{c,c'}$ and their Covariance is $\Sigma_{c,c'}$. Averaging this Covariance over all pairs of classes gives $G_3 = \frac{1}{C}\sum_{c,c'}\Sigma_{c,c'}$. The elements $\{\delta_{c,c}\}_c$ are not plotted above, however their location is at the point zero.*

cluster members $\{\delta_{c,c'}\}_{c,c'}$ and the cluster centers $\{\delta_c\}_c$.

We also investigate this phenomenon throughout the epochs of SGD. Figure 6 shows t-SNE plots of the cluster members $\{\delta_{c,c'}\}_{c,c'}$ in different epochs. We observe that the cluster members $\{\delta_{c,c'}\}_{c,c'}$ cluster around the cluster centers $\{\delta_c\}_c$ only after a certain number of epochs. Prior to that the cluster members $\{\delta_{c,c'}\}_{c,c'}$ cluster according to the logit coordinate c' and not the true class c.

We substantiate our claims, regarding a connection between the outliers in G and the corresponding eigenvalues of the Gram of cluster centers, by testing them empirically across different canonical datasets, contest winning architectures and various training sample sizes. We observe that the top-C outliers in G deviate from their predicted value by a small margin. This phenomenon is well known in the context of Random Matrix Theory (RMT), where the magnitude of such deviations can be computed using dedicated tools.

We summarize below our main deliverables:

1. We show that the outliers in the spectrum of the Hessian, previously attributed to G, are due to G being a second moment matrix and not a Covariance.

2. We show the columns of Δ, the matrix of logit derivatives that goes to form $G = \frac{1}{n}\Delta\Delta^T$, can be grouped into C^2 groups, which can then be grouped into C clusters.

3. We show how to approximate the top-C outliers in the

spectrum of G from the Gram of cluster centers.

4. We show that, empirically, the variation within each cluster is small compared to the variations between the clusters. That quantitative observation is responsible for the fact that the outliers in G are attributable to the Gram of the cluster centers.

5. We investigate the hierarchical structure throughout the epochs of SGD, showing that initially the group means are clustered according to the logit coordinate c' and only afterwards according to the true class c.

6. We verify empirically our claims across various datasets, networks and sample sizes.

7. We observe a deviation between the C outliers and our approximations and draw connections to RMT.

2. G is a second moment matrix

We begin by observing the dimensions of the components constituting G,

$$\text{Ave}_{i,c}\left\{\underbrace{\frac{\partial f(x_{i,c};\theta)^T}{\partial\theta}}_{p\times C}\underbrace{\left.\frac{\partial^2\ell(z,y_c)}{\partial z^2}\right|_{z=f(x_{i,c};\theta)}}_{C\times C}\underbrace{\frac{\partial f(x_{i,c};\theta)}{\partial\theta}}_{C\times p}\right\}.$$
(4)

In the following steps, we will decompose the $C \times C$ matrix $\frac{\partial^2\ell(z,y_c)}{\partial z^2}$ into an outer product of length C vectors. Notice that $\frac{\partial^2\ell(z,y_c)}{\partial z^2}$ is the Hessian of multinomial logistic regression. In (Böhning, 1992) it was shown to be equal to

$$\frac{\partial^2\ell(z,y_c)}{\partial z^2} = \text{diag}(p(x_{i,c};\theta)) - p(x_{i,c};\theta)p(x_{i,c};\theta)^T, \quad (5)$$

where $p(x_{i,c};\theta)$ are the probabilities obtained from applying softmax to the logits of $x_{i,c}$.

Lemma 2.1. *The Hessian of multinomial logistic regression can be equivalently written as follows:*

$$\frac{\partial^2\ell(z,y_c)}{\partial z^2} = (I - \mathbb{1}p(x_{i,c};\theta)^T)^T \text{diag}(p(x_{i,c};\theta))$$
$$\times (I - \mathbb{1}p(x_{i,c};\theta)^T). \quad (6)$$

Proof. Denote $p_i = p(x_{i,c};\theta)$ and $\sqrt{p_i}$ an element-wise square root of p_i. Then,

$$(I - \mathbb{1}p_i^T)^T \text{diag}(p_i)(I - \mathbb{1}p_i^T) \quad (7)$$
$$= (I - \mathbb{1}p_i^T)^T \sqrt{\text{diag}(p_i)}\sqrt{\text{diag}(p_i)}(I - \mathbb{1}p_i^T) \quad (8)$$
$$= (\sqrt{\text{diag}(p_i)} - p_i\sqrt{p_i}^T)(\sqrt{\text{diag}(p_i)} - \sqrt{p_i}p_i^T) \quad (9)$$
$$= \text{diag}(p_i) - p_ip_i^T - p_ip_i^T + p_i\sqrt{p_i}^T\sqrt{p_i}p_i^T \quad (10)$$
$$= \text{diag}(p_i) - p_ip_i^T = \frac{\partial^2\ell(z,y_c)}{\partial z^2}, \quad (11)$$

proving our desired claim. □

Plugging the above into the G term, we obtain:

$$G = \underset{i,c}{\mathrm{Ave}} \left\{ \frac{\partial f(x_{i,c};\theta)}{\partial \theta}^T (I - p(x_{i,c};\theta)\mathbb{1}^T) \right. \tag{12}$$

$$\left. \times \operatorname{diag}(p(x_{i,c};\theta))(I - \mathbb{1}p(x_{i,c};\theta)^T)\frac{\partial f(x_{i,c};\theta)}{\partial \theta} \right\}.$$

Let $\Delta_{i,c}$ denote the matrix associated with fixed i and c and varying c',

$$\Delta_{i,c}^T = \operatorname{diag}\left(\sqrt{p(x_{i,c};\theta)}\right)(I - \mathbb{1}p(x_{i,c};\theta)^T)\frac{\partial f(x_{i,c};\theta)}{\partial \theta}. \tag{13}$$

The above is a product of three matrices. The first is a matrix of logit derivatives $\frac{\partial f(x_{i,c};\theta)}{\partial \theta}$, which contains in its c'-th row the c'-th logit derivative. The second is a centering matrix; the term $p(x_{i,c};\theta)^T \frac{\partial f(x_{i,c};\theta)}{\partial \theta}$ is a weighted average of the C logit derivatives and the vector $\mathbb{1}$ duplicates this mean C times. The third is a diagonal matrix with the square root of the probabilities. The whole expression can be interpreted as centering the logit derivatives by subtracting their mean and then weighting the result by $\sqrt{p(x_{i,c};\theta)}$.

Returning to the derivation, using the definition of $\Delta_{i,c}$,

$$G = \underset{i,c}{\mathrm{Ave}} \left\{ \Delta_{i,c}\Delta_{i,c}^T \right\}. \tag{14}$$

Concatenating the matrices $\Delta_{i,c} \in \mathbb{R}^{p \times C}$ into a single matrix $\Delta \in \mathbb{R}^{p \times Cn}$, we get

$$G = \frac{1}{n}\Delta\Delta^T. \tag{15}$$

Note that no mean term has been subtracted; hence, G is a second moment matrix of logit derivatives, defined in Equation 13.

2.1. G is a second moment of logit derivatives, indexed by three integers

Note that $\Delta_{i,c} \in \mathbb{R}^{p \times C}$ and denote by $\delta_{i,c,c'}$ its c'-th column. Using this definition, we can decompose $\Delta_{i,c}\Delta_{i,c}^T$ into a summation over C elements, obtaining

$$G = \underset{i,c}{\mathrm{Ave}} \left\{ \sum_{c'=1}^{C} \delta_{i,c,c'}\delta_{i,c,c'}^T \right\}. \tag{16}$$

Hence, G is a second moment matrix of logit derivatives, which can be indexed by three integers, (i, c, c').

2.2. Relation to the gradients of the loss.

We now consider the relation between G (equivalently $\Delta_{i,c}$) and the gradients of the loss. Recalling Equation (1), the gradient of the i-th example can be written as follows,

$$\frac{\partial \ell(f(x_{i,c};\theta), y_c)}{\partial \theta}^T = \frac{\partial \ell(z, y_c)}{\partial z}^T \bigg|_{z=f(x_{i,c};\theta)} \frac{\partial f(x_{i,c};\theta)}{\partial \theta}. \tag{17}$$

In (Böhning, 1992) it was shown that the gradient of multinomial logistic regression is given by,

$$\frac{\partial \ell(z, y_c)}{\partial z} = y_c - p(x_{i,c};\theta). \tag{18}$$

Hence,

$$\frac{\partial \ell(f(x_{i,c};\theta), y_c)}{\partial \theta}^T = (y_c - p(x_{i,c};\theta))^T \frac{\partial f(x_{i,c};\theta)}{\partial \theta}. \tag{19}$$

Using Equation (13) and the definition of $\delta_{i,c,c'}$, we get

$$\delta_{i,c,c}^T = \sqrt{p_c(x_{i,c};\theta)}(y_c - p(x_{i,c};\theta))^T \frac{\partial f(x_{i,c};\theta)}{\partial \theta}, \tag{20}$$

where $p_c(x_{i,c};\theta)$ is the c-th element of $p(x_{i,c};\theta)$. Comparing Equations (19) and (20), we observe that $\delta_{i,c,c}$ and the gradient of the loss are equal up to a scalar. Note, however, that $G = \frac{1}{n}\Delta\Delta^T$ contains in addition to the outer products of $\delta_{i,c,c}$, outer products of $\delta_{i,c,c'}$ for $c' \neq c$. Hence, G is not a second moment of the *gradients of the loss*; instead it is a second moment of the *logit derivatives*.

3. Decomposing G into C^2 populations

Having established that G is a second moment matrix, our goal in this section is to decompose it into two components: one associated with its mean and the other with its variance. Denoting

$$\delta_{c,c'} = \underset{i}{\mathrm{Ave}} \left\{ \delta_{i,c,c'} \right\} \tag{21}$$

$$\Sigma_{c,c'} = \underset{i}{\mathrm{Ave}} \left\{ (\delta_{i,c,c'} - \delta_{c,c'})(\delta_{i,c,c'} - \delta_{c,c'})^T \right\}, \tag{22}$$

we can decompose G in Equation (16) as follows[1]:

$$G = \underbrace{\frac{1}{C}\sum_{c,c'} \delta_{c,c'}\delta_{c,c'}^T}_{G_{1+2}} + \underbrace{\frac{1}{C}\sum_{c,c'} \Sigma_{c,c'}}_{G_3}. \tag{23}$$

In the context of Figure 2, note that G_{1+2} corresponds to the aggregate of both the red circle and the blue one, while G_3 corresponds to the green circle.

Our original motivation for decomposing G into its mean and variance terms was to isolate the component that was creating the outliers in the spectrum. Previous observations (Sagun et al., 2016; 2017; Papyan, 2018) suggest the existence of C dominant outliers in the spectrum of G. On the other hand, the first summation in the above expression, being the outer product of C^2 elements, could be of rank C^2. This, in turn, would lead to C^2 outliers in the spectrum.

---

[1]We assume here the classes are balanced. Otherwise, G_{1+2} would be a weighted sum, with weights proportional to the number of examples in each class.

(a) MNIST, 13 examples per class.

(b) Fashion MNIST, 13 examples per class.

(c) CIFAR10, 13 examples per class.

(d) MNIST, 702 examples per class.

(e) Fashion MNIST, 702 examples per class.

(f) CIFAR10, 702 examples per class.

(g) MNIST, 5000 examples per class.

(h) Fashion MNIST, 5000 examples per class.
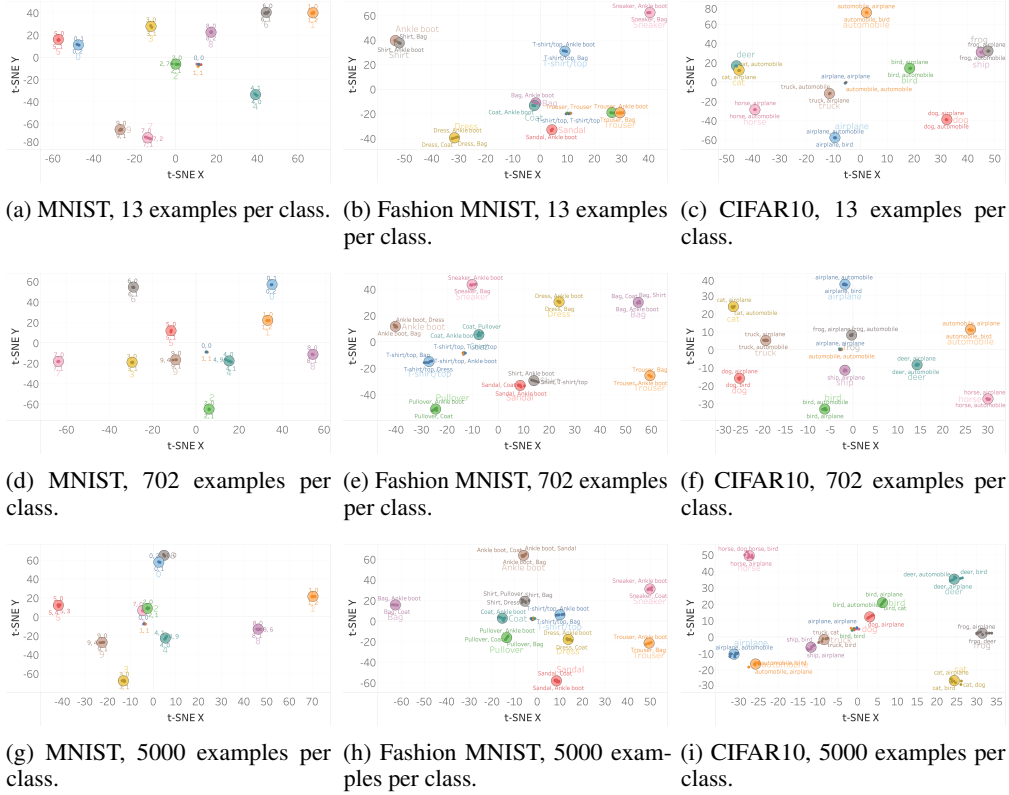
(i) CIFAR10, 5000 examples per class.

Figure 3. t-SNE visualization of the hierarchical structure for DenseNet40. Each column of panels corresponds to a different dataset, and each row to a different sample size. Each panel depicts the two-dimensional t-SNE embedding of the cluster members $\{\delta_{c,c'}\}_{c,c'}$ and the cluster centers $\{\delta_c\}_c$. All circles are colored according to the class c. The δ_c are marked with large circles and have a label written in large font attached to them. The $\delta_{c,c'}$ are marked with small circles and a subset of them also have a label, written in smaller font, attached to them. The label is a concatenation of the two class names corresponding to c and c'. This plot asserts the three level hierarchy. At level one we have the cluster centers $\{\delta_c\}_c$. At level two, next to each cluster center δ_c, we find cluster members $\{\delta_{c,c'}\}_{c'\neq c}$. Although not plotted, at level three, next to each $\delta_{c,c'}$ we would find $\{\delta_{i,c,c'}\}_i$. We also observe a cluster which contains all the $\{\delta_{c,c}\}_c$ ($c = c'$). These points are clustered together because their norms are close to zero, when compared to the other points.
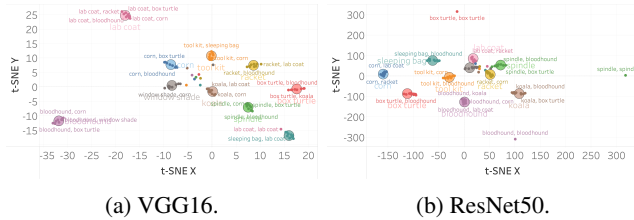

(a) VGG16.

(b) ResNet50.

Figure 4. t-SNE visualization of the hierarchical structure in ImageNet. Each panel depicts the two-dimensional t-SNE embedding of the cluster members and the cluster centers for a different architecture. For further details, see caption of Figure 3. VGG16 was trained on 600 examples per class and ResNet50 on the full dataset. For visualization purposes, we subset randomly ten classes. By and large, $\delta_{c,c'}$ with fixed c and varying c' cluster around δ_c, except for "oddballs" – which turn out to correspond to $c = c'$. Unlike Figure 3, these are not tightly clustered together. This is likely due to their norm not being close to zero, caused by a lack of convergence of SGD.

(a) 51 examples per class.

(b) 1351 examples per class.

Figure 5. t-SNE visualization of the hierarchical structure for VGG11 trained on FashionMNIST with fully corrupted labels. Each panel depicts the two-dimensional t-SNE embedding of the cluster members and the cluster centers for a different training sample size. For further details, see caption of Figure 3. The points are colored according to the logit coordinate c' and not the true class c. Due to the model being trained with scrambled labels, the cluster members cluster according to the logit coordinate c' instead of the true class c. This is akin to Figure 6, where a similar phenomenon occurs in the early epochs of SGD.

(a) ResNet50, epoch 1, colored according to c'.

(b) ResNet50, epoch 10, colored according to c'.

(c) ResNet50, epoch 18, colored according to c.

(d) ResNet50, epoch 33, colored according to c.

(e) ResNet50, epoch 350, colored according to c.

(f) VGG16, epoch 1, colored according to c'.

(g) VGG16, epoch 10, colored according to c.

(h) VGG16, epoch 18, colored according to c.

(i) VGG16, epoch 33, colored according to c.

(j) VGG16, epoch 350, colored according to c.

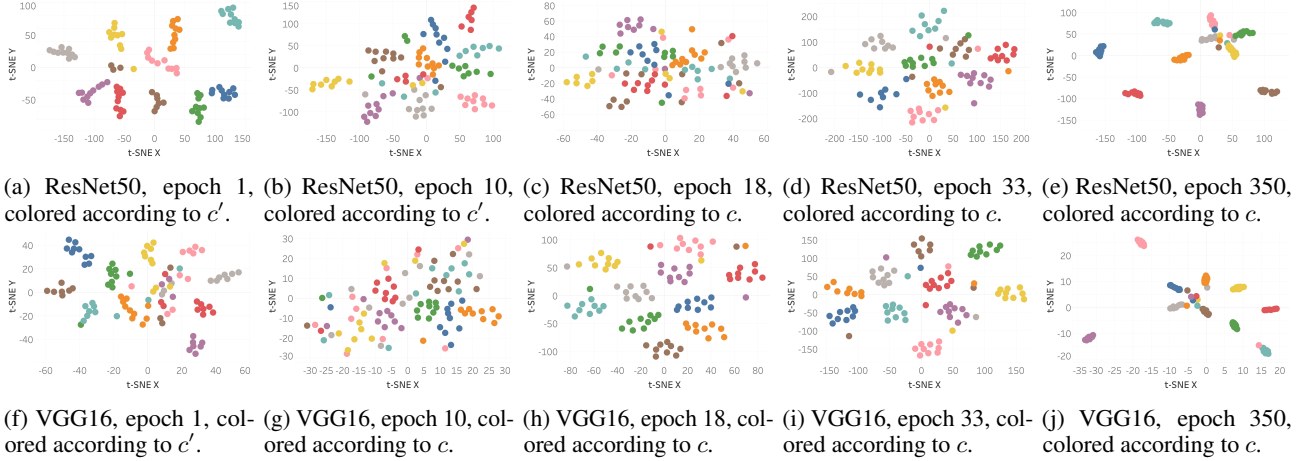*Figure 6. t-SNE visualization of the hierarchical structure in ImageNet as a function of epoch.* Each row of panels corresponds to a different architecture, and each column to a different epoch. Each panel depicts the two-dimensional t-SNE embedding of the cluster members $\{\delta_{c,c'}\}_{c,c'}$. ResNet50 was trained on 600 examples per class, while VGG16 on the full dataset. For computational reasons, we subset randomly ten classes. The smaller epochs are colored according to the logit coordinate c', while the larger according to the true class c. A transition occurs around epoch 18 for ResNet50 and epoch 10 for VGG16 – from cluster members clustering according to the logit coordinate c', to cluster members clustering according to the true class c.

We explain this purported contradiction by noting that while there exist C^2 outliers, C of them are significantly more dominant than the others. In the next section, we show how to extract the C dominant outliers.

4. The means themselves have structure

In this section we focus on further decomposing the G_{1+2} term. For reasons that will become clear later, we separate the elements that correspond to $c = c'$ from the rest,

$$G_{1+2} = \underset{c}{\mathrm{Ave}}\{\delta_{c,c}\delta_{c,c}^T\} + \frac{1}{C}\sum_c\sum_{c'\neq c}\delta_{c,c'}\delta_{c,c'}^T. \quad (24)$$

Denoting

$$\delta_c = \underset{c'\neq c}{\mathrm{Ave}}\{\delta_{c,c'}\} \quad (25)$$

$$\Sigma_c = \underset{c'\neq c}{\mathrm{Ave}}\{(\delta_{c,c'} - \delta_c)(\delta_{c,c'} - \delta_c)^T\}, \quad (26)$$

we can further decompose G_{1+2} into:

$$G_{1+2} = \underset{c}{\mathrm{Ave}}\{\delta_{c,c}\delta_{c,c}^T\} + (C-1)\underset{c}{\mathrm{Ave}}\{\delta_c\delta_c^T\}$$
$$+ (C-1)\underset{c}{\mathrm{Ave}}\{\Sigma_c\}. \quad (27)$$

Plugging the above expression into Equation 23, we obtain

$$G = \underbrace{\underset{c}{\mathrm{Ave}}\{\delta_{c,c}\delta_{c,c}^T\}}_{G_0} + \underbrace{(C-1)\underset{c}{\mathrm{Ave}}\{\delta_c\delta_c^T\}}_{G_1}$$
$$+ \underbrace{(C-1)\underset{c}{\mathrm{Ave}}\{\Sigma_c\}}_{G_2} + \underbrace{\frac{1}{C}\sum_{c,c'}\Sigma_{c,c'}}_{G_3}. \quad (28)$$

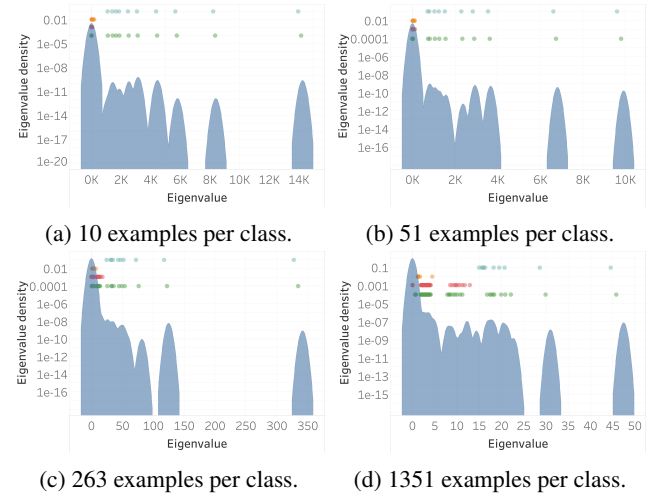Compare Equation (28) with Figure 2. The expressions given here implement the structure depicted in Figure 2.

(a) 10 examples per class.

(b) 51 examples per class.

(c) 263 examples per class.

(d) 1351 examples per class.

Figure 7. Decomposing G into its constituent components for VGG11 trained on MNIST with MSE loss. Each panel corresponds to a different sample size. Each panel depicts the density of the spectrum of G in **blue**, where the y-axis is on a logarithmic scale. The density was approximated using the LANCZOSAPPROXSPEC method presented in (Papyan, 2018). Each panel also plots the eigenvalues of G_0 in **cyan**, G_1 in **orange**, G_2 in **red** and G_{1+2} in **green**. We observe that once the network is trained to minimize the MSE loss, the outliers can be predicted from the eigenvalues of the matrix G_0 and not G_1, as was the case for the cross-entropy loss. This suggests that an alternative decomposition should be proposed for the MSE loss.
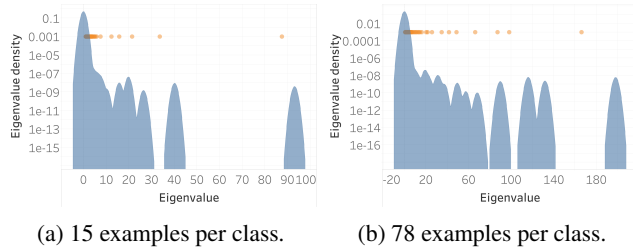
(a) 15 examples per class. (b) 78 examples per class.

Figure 8. Decomposing G into its constituent components for VGG11 trained on CIFAR100. Each panel corresponds to a different sample size. Each panel depicts the density of the spectrum of G in **blue**, where the y-axis is on a logarithmic scale. The density was approximated using the LANCZOSAPPROXSPEC method presented in (Papyan, 2018). Each panel also plots the eigenvalues of G_1 in **orange**. The results show that the outliers in the second moment matrix G, which are separated from the bulk, align with the eigenvalues of G_1 up to a shift.
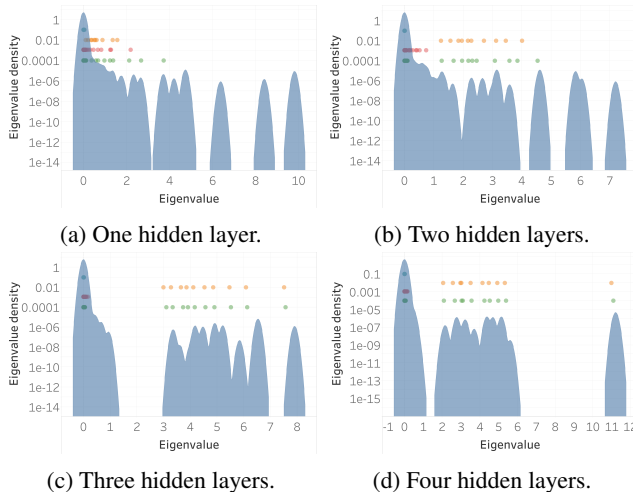

(a) One hidden layer. (b) Two hidden layers.

(c) Three hidden layers. (d) Four hidden layers.

Figure 9. Decomposing G into its constituent components for multilayer perceptron with varying depths. The network is composed of fully connected layers of width 1024 neurons, followed by batch normalization and then ReLU. Each panel depicts for a different number of hidden layers the density of the spectrum of G in **blue**, where the y-axis is on a logarithmic scale. The density was approximated using the LANCZOSAPPROXSPEC method presented in (Papyan, 2018). Each panel also plots the eigenvalues of G_0 in **cyan**, G_1 in **orange**, G_2 in **red** and G_{1+2} in **green**. Note how the outliers align with the eigenvalues of G_1 once the number of hidden layers is larger than two.

5. Experiments

We train VGG11 (Simonyan & Zisserman, 2014), ResNet18 (He et al., 2016) and DenseNet40 (Huang et al., 2017) on the MNIST (LeCun et al., 2010), Fashion MNIST (Xiao et al., 2017) and CIFAR10 (Krizhevsky & Hinton, 2009) datasets. We use stochastic gradient descent with 0.9 momentum, 5×10^{-4} weight decay and 128 batch size. The initial learning rate is annealed by a factor of 10 at $1/3$ and $2/3$ of the

number of epochs. We train for 200 epochs on MNIST and Fashion MNIST and 350 for CIFAR10. For each dataset and network, we sweep over 100 logarithmically spaced initial learning rates in the range $[0.25, 0.0001]$ and pick the one that results in the best test error in the last epoch. For each dataset and network, we repeat the previous experiments on 20 training sample sizes logarithmically spaced in the range $[10, 5000]$. The total number of experiments ran was 18000. The massive amount of computational experiments was run painlessly using ClusterJob and ElastiCluster (Monajemi & Donoho, 2015; Monajemi et al., 2017; 2019).

We also train VGG16 and ResNet50 on ImageNet (Deng et al., 2009) using the same parameters described above, except we use a batch size of 512 with an initial learning rate of 0.01 and train for 350 epochs. We train ResNet50 on 600 examples per class and VGG16 on the full dataset.

We compute the eigenvalues of G_1, G_2 and G_{1+2} using the EIG function available in modern standard libraries such as SciPy. However, instead of computing the eigenavlues of $G_1 = (C-1) \sum_c \delta_c \delta_c^T$, for example, we compute the eigenvalues of the corresponding $C \times C$ Gram matrix.

We summarize our results in Figures 3, 4, 5, 6, 7, 8, 9, 10 and Figure 1 in the Appendix, and discuss their implications in the captions. The code used for generating these figures is available at https://github.com/deep-lab/DeepnetHessian.

5.1. A note on stochasticity

Deep learning practitioners often insert randomness into their architectures. The most common examples are preprocessing the input data using random flips and crops or using dropout (Srivastava et al., 2014) layers. These sources of randomness complicate the analysis of the Hessian and its components in that they turn them into random variables. This complicates the usage of the methods we employ in this paper – such as Lanczos, subspace iteration and SVD – all of which assume deterministic linear operators. To circumvent these nuisances, we do not employ any preprocessing on the input data and we replace the dropout layers in the VGG architecture with batch normalization layers.

6. Conclusion

Outliers sticking beyond the bulk edge were previously observed in the spectrum of the Hessian of deep networks. This paper described an organization of the ingredients of the Hessian which explains the outliers. The structuring we introduce here offers a novel three-level hierarchical decomposition. This provides an approximation for the outliers, which was proven empirically across many scenarios. Moreover, deviations between the two were found to exist, as might have been predicted by RMT.

(a) MNIST, 10 examples per class.

(b) Fashion MNIST, 10 examples per class.

(c) CIFAR10, 10 examples per class.

(d) MNIST, 136 examples per class.

(e) Fashion MNIST, 136 examples per class.

(f) CIFAR10, 136 examples per class.

(g) MNIST, 365 examples per class.

(h) Fashion MNIST, 365 examples per class.

(i) CIFAR10, 365 examples per class.

(j) MNIST, 702 examples per class.

(k) Fashion MNIST, 702 examples per class.

(l) CIFAR10, 702 examples per class.

(m) MNIST, 2599 examples per class.

(n) Fashion MNIST, 2599 examples per class.

(o) CIFAR10, 1351 examples per class.

Figure 10. Decomposing G into its constituent components for the ResNet18 architecture. Each column of panels corresponds to a different dataset, and each row to a different sample size. Each panel depicts the density of the spectrum of G in **blue**, where the y-axis is on a logarithmic scale. The density was approximated using the LANCZOSAPPROXSPEC method presented in (Papyan, 2018). Each panel also plots the eigenvalues of G_0 in **cyan**, G_1 in **orange**, G_2 in **red** and G_{1+2} in **green**. The obtained results corroborate certain predictions made throughout our analysis. Specifically, the outliers in the second moment matrix G can be attributed to the eigenvalues of G_{1+2}. The top-C eigenvalues of G_{1+2} are dominant compared to the others and they match those of G_1. The eigenvalues of G_0 are negligible compared to the spread of the others. They correspond to a single cyan point in the main lobe of each plot. For sample size 365 all three datasets exhibit worse alignment between the outliers of G and the eigenvalues of G_1. Theoretically predicting the deviations between the outliers and the eigenvalues can potentially be done using tools developed in RMT.

Acknowledgements

References

Böhning, D. Multinomial logistic regression algorithm. *Annals of the institute of Statistical Mathematics*, 44(1): 197–200, 1992.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. Ieee, 2009.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*, 2017.

Geiger, M., Spigler, S., d'Ascoli, S., Sagun, L., Baity-Jesi, M., Biroli, G., and Wyart, M. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. *arXiv preprint arXiv:1809.09349*, 2018.

Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pp. 1731–1741, 2017.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *CVPR*, volume 1, pp. 3, 2017.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. On the relation between the sharpest directions of dnn loss and the sgd step length. 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Monajemi, H. and Donoho, D. L. Clusterjob: An automated system for painless and reproducible massive computational experiments. https://github.com/monajemi/clusterjob, 2015.

Monajemi, H., Donoho, D. L., and Stodden, V. Making massive computational experiments painless. *Big Data (Big Data), 2016 IEEE International Conference on*, February 2017.

Monajemi, H., Murri, R., Yonas, E., Liang, P., Stodden, V., and Donoho, D. Ambitious data science can be painless. *arXiv:1901.08705*, 2019.

Papyan, V. The full spectrum of deep net hessians at scale: Dynamics with sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning*, pp. 2798–2806, 2017.

Pennington, J. and Worah, P. The spectrum of the fisher information matrix of a single-hidden-layer neural network. In *Advances in Neural Information Processing Systems*, pp. 5415–5424, 2018.

Sagun, L., Bottou, L., and LeCun, Y. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., and Wyart, M. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yaida, S. Fluctuation-dissipation relations for stochastic gradient descent. *arXiv preprint arXiv:1810.00004*, 2018.