The Provable Unsupervised Learning Rule Known as Batch Normalization

Rudolf Riedi RUDOLF.RIEDI@HEFR.CH *Mathematics Department of the HES-SO, University of Applied Sciences of Western Switzerland, Fribourg, Switzerland.*

Randall Balestriero RANDALL_BALESTRIERO@BROWN.EDU *Department Computer Science, Brown University, Providence, RI, 02906 USA*

Richard Baraniuk RICHB@RICE.EDU *Department of Electrical and Computer Engineering, Rice University, Houston, TX, 77005 USA*

Abstract

Batch normalization (BN) accelerates training and improves generalization, yet why it works remains unsettled. Prior explanations focus on the loss landscape but cannot explain why BN still helps in architectures that already optimize well. We offer a complementary geometric account: BN is an unsupervised learning rule. For networks with ReLU-like activations, BN's centering forces every neuron's hyperplane through the mini-batch mean, anchoring the network's spline partition to the data, independently of the labels or loss. We prove that this anchoring makes random hyperplanes separate low-dimensional data clusters at initialization with high probability and that BN's scaling refines partition boundaries near the data manifold. BN thus acts as a label-free mechanism for geometric adaptation and smart initialization.

Keywords: Batch normalization, unsupervised learning, spline partition, linear regions

1. Introduction

Batch normalization (BN) is a standard component in modern deep networks (DNs), but still not completely understood Ioffe and Szegedy (2015). BN centers and rescales intermediate features over a mini-batch and is empirically known to accelerate training, stabilize optimization, and improve generalization across a wide range of architectures. Most prior explanations have focused on optimization-centric viewpoints, such as a smoothing or reconditioning of the loss landscape and gradients LeCun et al. (2002); Salimans and Kingma (2016); Bjorck et al. (2018); Santurkar et al. (2018); Kohler et al. (2019); Yang et al. (2019). These perspectives, while valuable, do not fully account for BN's persistent benefits even in architectures that already enjoy favorable optimization properties, such as residual and mollifying networks Gulcehre et al. (2016) and when combined with advanced optimizers such as Adam Kingma and Ba (2014).

In this paper, we take a complementary geometric perspective to discover certain key properties of BN that relate to unsupervised learning and the manner in which a DN partitions its input space. In particular, for DNs based on continuous piecewise linear activation functions (e.g., ReLU), we analyze how BN's centering and scaling operations act directly on the affine hyperplane underlying each neuron and thereby on the induced partition of the input space.

Since we wish to study the influence of the BN parameters (centering and normalization) separately we show the parameters w (weights, normal vector), c (bias, offset) and b (normalization, scaling, $b \neq 0$) of the pre-activations of layer ℓ and neuron k explicitly:

$$h_{\ell,k}(z; w, c, b) := h_k(z; w, c, b) := \frac{w^\top z - c}{b} = \frac{\|w\|}{b}\, d(z; w, c). \tag{1}$$

Whenever no confusion is possible we don't show the indices ℓ and k for simplicity of presentation. In (1), e.g., the parameters w, c and b should be interpreted as variables with values that will be individually chosen for each layer ℓ and neuron k. The sign of the term $d(z; w, c)$ in (1) identifies the two hyper-halfspaces separated by the hyper-plane

$$H_{\ell,k}(w, c) = H(w, c) = \{z \ : \ h(z; w, c, b) = 0\}$$

of points where the pre-activation vanishes. Also, it is well known that $|d(z; w, c)|$ equals the Euclidean distance of z from the hyperplane $H(w, c)$. Both, H and d are independent of the actual value of b used in h.

Our starting point is the simple observation that, in a BN layer, the offset c is chosen so that the hyperplane $H(w, c)$ passes through the center of gravity g

$$g \ = \ \frac{1}{|\mathcal{B}|} \sum_{z \in \mathcal{B}} z = \text{mean}_{z \in \mathcal{B}}\, z \tag{2}$$

of the mini-batch $\mathcal{B}$. To see this note that in BN one uses offset $c = \mu$ and factor $b = \sigma$ where

$$\mu \ = \ \text{mean}_{z \in \mathcal{B}}\, w^\top z = w^\top g \qquad \sigma^2 \ = \ \text{var}_{z \in \mathcal{B}}\, w^\top z \tag{3}$$

are computed as the element-wise mean and variance of the pre-activations over each mini-batch $\mathcal{B}$ during training and over the entire training set during testing. As a consequence, $h(g; w, \mu, b) = \frac{1}{b}(w^\top g - \mu) = \frac{1}{b}(w^\top g - w^\top g) = 0$ and the hyperplane passes through g.

Short, in BN one uses pre-activations of the form[1] $h(z; w, w^\top g, \sigma)$ with corresponding hyperplanes $H(w, w^\top g)$

Since an entire DN can be represented as a continuous piecewise affine spline operator (cf. section 2) whenever the activation functions are ReLU-like, such constraints on the per-layer hyperplanes have far reaching consequences. In particular, we show provable benefits of the BN rule for both (i) the so-called linear regions Montufar et al. (2014); Balestriero and Baraniuk (2018, 2021) into which the DN partitions its input space, and (ii) the ability of the network, even at random weight initialization, to separate structured data sets by hyperplanes.

Our analysis proceeds at the level of one and two layers. The collection of signs of these pre-activations across neurons defines a spline partition of the input space into regions separated by hyperplanes; see Figure 1 for an illustration with a small random network. In a randomly placed layer (random offsets c, factors $b = 1$), the hyperplanes are essentially unconstrained with respect to the data and induce a partition that is largely agnostic to the mini-batch. In contrast, in a BN layer (offset $c = w^\top g$, scale $b = \sigma$), all hyperplanes contain g, and the resulting regions are systematically drawn towards the data cloud.

1. The original definition of BN includes an additional centering and scaling parameter that are directly through the DN's objective function (e.g., squared error or cross-entropy). However, it has been shown that these parameters have minimal to no effect on the final performance of the DN Balestriero and Baraniuk (2022).

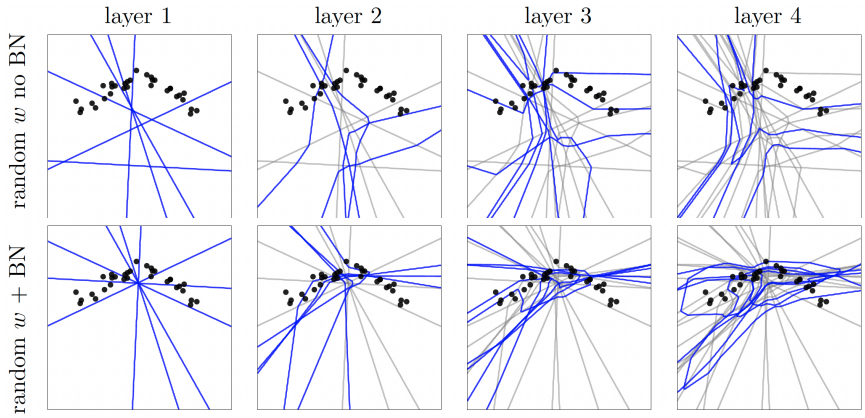

Figure 1: Visualization of the input-space spline partition (linear regions Montufar et al. (2014); Balestriero and Baraniuk (2018, 2021)) of a four-layer DN with a 2D input space, 6 units per layer, leaky-ReLU activation function, and random weights. The training data samples are denoted with black dots. In each plot, blue lines correspond to folded hyperplanes introduced by the units of the corresponding layer, while gray lines correspond to (folded) hyperplanes introduced by previous layers. Top row: Without BN (i.e., using random offset c and scaling factor $b = 1$), the folded hyperplanes are spread throughout the input space, resulting in a spline partition that is agnostic to the data. Bottom row: With BN (i.e., using offset $c = \mu$ and scaling factor $b = \sigma$)), the folded hyperplanes are drawn towards the data, resulting in an adaptive spline partition that — even with random weights — minimizes the distance between the partition boundaries and the data and thus increases the density of partition regions around the data.

This geometric anchoring leads to our main technical contribution:

Main result: Separation with known probability. We first consider a layer ℓ with randomly oriented hyperplanes that all pass through a common center g and study their ability to separate given sets in its input space $\mathbb{R}^{D_\ell}$. We first show that, for two points u, v with midpoint m, the choice $g = m$ maximizes the probability that a random hyperplane through g separates u and v; in fact, separation occurs with probability 1 in this case and decays as g moves away from m. Building on this, we derive explicit, favorable bounds on the probability that a BN-style random hyperplane through the mini-batch center separates two clusters with spread δ and separation r, when each class cluster lies on a low-dimensional parsimonious subset (of dimension $K \ll r/\delta$). In particular, under mild geometric assumptions (theorem 2) the failure probability admits an explicit bound of order

$$P[U \not\parallel V] \leq (2/\pi)\frac{2K-1}{r/\delta - 1},$$

which holds also when replacing U and V by their convex hulls. In sharp contrast, for two K-dimensional balls of radius δ (theorem 3), the probability that a random hyperplane through the midpoint of their centers separates them is exponentially small as $K \to \infty$, unless their diameter is extremely small relative to their separation r.

These results demonstrate that, even with random weights at initialization, BN makes separation likely whenever the data are low-dimensional and sufficiently well separated. This formalizes, in a simple probabilistic model, how BN's centering mechanism leverages data parsimony.

Bonus result: Effect of normalization on higher-order boundaries. Going further, in Appendix A, we consider two consecutive DN layers and study how the BN scaling factors shape the "second-order" boundaries of the current input space induced by the next layer. We show there that, while the first-order boundaries (the hyperplanes of layer 1) depend only on the offsets and not on the scales, the set of points that are mapped to the feature-space center of gravity g_1 is invariant under any positive scaling. All second-order boundaries (i.e., folds introduced by layer 2) necessarily pass through this set. For a simple two-cluster model we demonstrate analytically that BN scaling equalizes the coordinates of g_1 and thereby disperses the folding locations more evenly around the data, leading to finer partitions and boundaries that are closer to the data, even with random weights. This effect persists (in a quantified sense) for small but nonzero cluster spread.

Taken together, our results support a view of BN as an unsupervised geometric learning mechanism that (i) aligns separating hyperplanes with the local center of mass of the data, making separation of low-dimensional clusters at random initialization probabilistically likely, and (ii) shapes the multi-layer spline partition so that more regions and boundaries are concentrated near the data manifold. This perspective complements existing optimization-based explanations and helps clarify why BN remains beneficial even in architectures whose loss surfaces are already well behaved.

2. Background: Splines and Hyperplane Offsets

We focus on DNs that employ continuous piecewise-linear activation functions a that are applied component-wise to vectors. To streamline notation, and without loss of generality, we assume that the activation a consists of exactly two linear pieces that meet at the origin, such as the ubiquitous ReLU ($a(u) = \max(0, u)$), leaky-ReLU ($a(u) = \max(\eta u, u)$, with $1 > \eta > 0$), or absolute value ($a(u) = \max(-u, u)$). The extension to more general continuous piecewise-linear activation functions is straightforward.

A DN layer ℓ partitions its input space $\mathbb{R}^{D_\ell}$ into a collection Ω_ℓ of open, convex polytopal regions (the so-called *linear regions* Montufar et al. (2014)) formed by the *hyperplane arrangement* Zaslavsky (1975) $\partial\Omega_\ell = \bigcup_{k=1}^{D_\ell} H_{\ell,k}$. See Figure 1 for an illustration with a small random network. Each linear region can be identified by the signs which the pre-activations $h_{\ell,k}$ attain at the points in such a region.

We start by stating some useful facts regarding these regions. Since they are valid for any layer ℓ we choose $\ell = 0$ and suppress the index ℓ for simplicity. We call a layer *centered*, if the boundaries of all linear regions intersect in at least one common point q.

Some facts. (A) In a centered layer all linear regions are unbounded.
(B) For a *condensing* layer, meaning that $D_1 \leq D_0$, with i.i.d. continuous random weights $w_k[j]$ there is a linear region where all pre-activations attain strictly positive values, and one where they all attain strictly negative values, almost surely.
(C) For a layer as in (B), almost surely, the layer is centered, and consequently, all regions are unbounded. If $D_1 = D_0$ then the common point is almost surely unique.
(D) For an *expanding* layer, meaning that $D_1 > D_0$, where the weights $w_k[j]$ and offsets c_k are i.i.d. continuous random variables, the layer is not centered, almost surely.
(E) For a layer as in (D) there is a non-empty, bounded region, almost surely.

Relevance of the facts. Property (B) is relevant since the layer preserves as much information as possible regarding points in the "positive" region, and looses all differences between points in the "negative" region. Also, (B) implies that there are points in the input space that are mapped into the interior of first hyper-octant of the output space.

The distinction between condensing and expanding layers (cpre. Lemma 5) will be important when we analyze separation properties and the effect of BN on higher-order boundaries in the appendix.

Arguments for the facts. Here, we provide a short argumentation which emphasizes the geometric aspects of a partition and which may be sufficiently detailed at first reading. Full arguments, however, can be found in the appendix (see Lemma 4 and 5).

Property (A) can be seen most easily by picking any point y in the interior of a linear region and considering the straight line through q and y. This line will intersect the hyperplanes forming the boundary of the region only in q. Therefore, the region is unbounded.

Now, consider the first D_0 neurons and their hyperplanes; for a condensing layer add some "auxiliary neurons" if needed. Choose new coordinates $\tilde{x}[k] = w_k^\top x - c_k$ $(k = 1 \dots D_0)$ which are, up to a positive constant, equal to the signed distance of the point x from these hyperplanes (cpre. (1)). Almost surely, the matrix with these D_0 rows $w_k^\top$ is invertible and the change of coordinates is linear, invertible and continuous both ways, therefore preserving relevant properties such as boundedness, interior, boundary, connectedness, affinity etc.

In these new coordinates, the hyperplanes $(k = 1 \dots D_0)$ are the principal coordinate planes intersecting at the new origin, implying (B) and (C). For (D) and (E) notice that in new coordinates the hyperplane $\tilde{H}_{D_0+1}$ will almost surely not pass through the origin, leaving the layer non-centered, and cut a finite region out of one of the new hyper-octants.

Consequences for BN-layers. In BN, the offsets are constrained so that each associated hyperplane $H_{\ell,k}$ contains the center of gravity g of the mini-batch $\mathcal{B}$ defined in (2), see also (3), forming a centered hyperplane arrangement where all regions are unbounded and their boundaries all contain the mini-batch center g.

It is well known that g is closest to the mini-batch data in the sense that $\sum_{z\in\mathcal{B}} \|x - z\|^2$ is minimized at $x = g$. Short, all regions are, in this precise sense, anchored near the data cloud, whereas for a layer with randomly placed offsets the region boundaries almost surely do not pass through g.

The hyperplanes of a *condensing* layer also admits a common intersection point q, almost surely, and its regions possess the same shape and orientation as those of a BN layer with the same weight vectors but BN offsets; in fact, they are simply a random translation (moving q to g) of each other (see Figure 3, top).

3. Single Hidden Layer: BN Separation Properties

We first consider only one neuron of one arbitrary layer, say $\ell = 0$, and drop the indices ℓ and k. We study randomly oriented hyperplanes $h(z; w, \mu, b)$ (recall (1)) that all pass through a common pole g. Concretely, the weight vector $w \in \mathbb{R}^{D_0}$ of the neuron consists of *i.i.d. zero-mean Gaussian* entries $w[k]$ with arbitrary variance, and the offset is chosen as $c = \mu = w^\top g$.

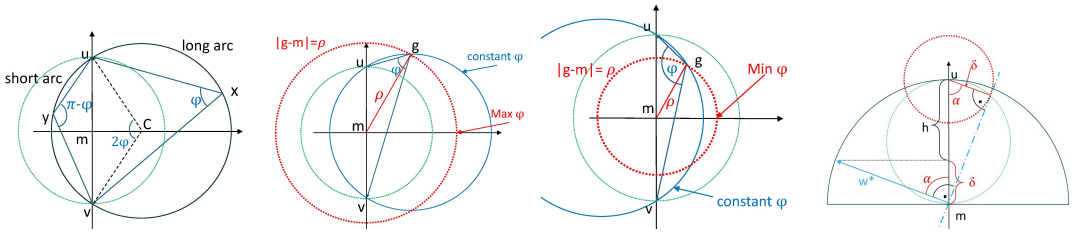

Figure 2: Reminder of the theorem of the inscribed angle (far left). Visualizations for the proofs of Lemma 1 (left and right) and Theorem 3 (far right).

Separating two points. Consider the task of separating two given points u and v at half distance $r = \|u - v\|/2$ and with midpoint $m = (u + v)/2$ using a randomly oriented hyperplane $h(x; w, c, b)$ (see (1)) passing through a common center g. This means that the weight vector w consists of D_0 i.i.d. zero-mean Gaussian variables $w[k]$ with a given arbitrary variance and that $c = \mu = w^\top g$. At this point, no assumption is made on the position of g relative to u and v. Denote the probability of successful separation by $P_g[u\|v]$.

Lemma 1 *In the setting of Section 3 we have* $P_g[u\|v] = 1$ *for* $g = m$ *and*

$$P_g[u\|v] \begin{cases} \geq (2/\pi)\arctan\left(r/\|m-g\|\right) \geq 1/2 & \text{if } 0 < \|m-g\| \leq r, \\ \leq (2/\pi)\arctan\left(r/\|m-g\|\right) < 1/2 & \text{if } r < \|m-g\|. \end{cases} \tag{4}$$

Proof Consider the plane passing through the three points u, v and g.

More concretely, we may move the coordinate center into m and rotate the coordinate system so that the point u lies on the new positive 2nd semi-axis and that the center g lies in the new $(x[1], x[2])$-plane. By exchanging u and v and/or map $x[1]$ to $-x[1]$ if necessary we may further assume that the center g lies in the first quadrant of the new $(x[1], x[2])$-plane.

Note that this translation, rotation and, if necessary, reflections do not change distances between points, nor angles between vectors, nor the distribution of w. In particular, in new or old coordinates, w consists of i.i.d. Gaussian coordinates with hyper-spherical symmetry.

In these coordinates whether or not a hyperplane through the center g separates u and $v = -u$ depends only on $w[1]$ and $w[2]$, since only the first two coordinates of u, v and g are non-zero. But, the vectors $(w[1], w[2])$ with the desired separation property and with $w[2] > 0$ fill an angular section with angle equal to the angle $\varphi = \angle[(u - g), (v - g)]$ that g encloses with u and v. Since $w[1]$ and $w[2]$ are i.i.d. normal variables, the distribution of $(w[1], w[2])$ is invariant under rotation around the origin. Therefore, the probability that $(w[1], w[2])$ with $w[2] > 0$ falls in this sector is exactly φ/π. A symmetrical argument applies for the case $w[2] < 0$. Thus, $P_g[u\|v] = \varphi/\pi$.

Now, fix $\|m - g\| = \rho$ and move g along this circle at constant distance ρ from m. Doing so, φ assumes its extreme value on the $w[1]$-axis which amounts to $2\arctan(r/\|g - m\|)$ (see Figure 2). There, φ is maximal in the case $\|m - g\| \geq r$ and minimal in the case $\|m - g\| \leq r$. ■

Separating two parsimonious (low-dimensional) sets. We now consider a mini-batch $\mathcal{B}$ consisting of two clusters of points U and V centered at u and v, respectively, in D_0-dimensional space. Denote the half distance of the centers by $r = \|u - v\|/2$. Each

cluster contains K points, and all points in U (resp. V) lie at Euclidean distance at most δ from u (resp. v). We refer to δ as the *spread* of the cluster. Clearly, the dimension of each cluster is at most K. Under the mild assumption that the difference vectors from the points to their respective center are i.i.d. and that $K < D_0$, the dimension of each cluster is almost surely equal to K.

We seek to separate the two clusters using a random hyperplane passing through the center of gravity g of the mini-batch (cf. (2)), which is precisely the choice imposed by BN. Unlike the previous subsection, the position of g has now a given relation to the cluster points and satisfies $g = (u+v)/2$. Then, the probability $P[U \not\parallel V]$ of *failing* to separate U and V can be bound as follows, in the setting of Section 3:

Theorem 2 **[Separating two clusters]** *Assume $\kappa := r/\delta - 1 \geq 1$, i.e., $r > 2\delta$. Then,*

$$P[U \not\parallel V] \leq \frac{2}{\pi} \cdot \frac{2K-1}{r/\delta - 1}. \tag{5}$$

Due to the affinity of the hyperplane we may replace U and V by their *convex hull* with the same result. This inequality shows that a nontrivial spread δ can be tolerated relative to the inter-center distance r (e.g., $\delta < r/(2K)$), provided the intrinsic dimension K of the clusters remains small (note that $r < \sqrt{D_0}$ in typical applications in image processing). In other words, high-probability separation is compatible with parsimonious, low-dimensional structure.

Proof Denote the points of the cluster U by u_k $(k = 1, 2, \ldots, K)$ and those of V by v_k. Separating U and V means to separate all pairs of points $u_k,\ v_k$ $(k = 1, 2, \ldots, K)$ from each other, but not the pairs $u_1,\ u_k$ $(k = 2, 3, \ldots, K)$. Indeed, in this situation the signs of $h(u_k; w, w^\top g, b)$ are all equal and at the same time different from $h(v_k; w, w^\top g, b)$. Consequently, failure to separate U and V means to fail in at least one of these $2K-1$ requirements:

$$U \not\parallel V = \left(\bigcup_{k=1}^{K} \{u_k \not\parallel v_k\} \right) \cup \left(\bigcup_{k=2}^{K} \{u_1 \parallel u_k\} \right).$$

Denote the midpoint of u_k and v_k by $m_k = (u_k + v_k)/2$ and their half distance by $r_k = \|u_k - v_k\|/2$. Note that $r_k \geq r - \delta$ and

$$\|m_k - g\| = \|(u_k + v_k)/2 - (u+v)/2\| \leq \|(u_k - u) + (v_k - v)\|/2 \leq \delta$$

Then, using $\delta = r/(\kappa+1)$ we get $r_k/\|m_k - g\| \geq (r-\delta)/\delta = \kappa \geq 1$. Using Lemma 1

$$P_g[u_k \not\parallel v_k] \leq 1 - \frac{2}{\pi} \arctan\left(r_k/\|m_k - g\|\right) \leq 1 - \frac{2}{\pi} \arctan\left(\kappa\right) = \frac{2}{\pi} \arctan\left(\frac{1}{\kappa}\right) \leq \frac{2}{\pi} \cdot \frac{1}{\kappa}.$$

Denote the midpoint of u_1 and u_k by $\tilde{m} = (u_1 + u_k)/2$ and their half distance by $\tilde{r} = \|u_1 - u_k\|/2$. Note that $\|\tilde{m} - u\| = \|(u_1 - u)/2 + (u_k - u)/2\| \leq \delta/2 + \delta/2 = \delta$ so that

$$\|\tilde{m} - g\| = \|(u - g) + (\tilde{m} - u)\| \geq \|u - g\| - \|\tilde{m} - u\| \geq r - \delta > 0.$$

Then, using $\tilde{r} = \|(u_1 - u) + (u - u_k)\|/2 \leq \delta$ and $\delta = r/(\kappa+1)$ we obtain

$$\frac{\tilde{r}}{\|\tilde{m} - g\|} \leq \frac{\delta}{r - \delta} = \frac{\delta}{(\kappa+1)\delta - \delta} = \frac{1}{\kappa} \leq 1.$$

Using Lemma 1,

$$P_g[u_1 \parallel u_k] \leq \frac{2}{\pi} \arctan\left(\frac{\tilde{r}}{\|\tilde{m} - g\|}\right) \leq \frac{2}{\pi} \arctan\left(\frac{1}{\kappa}\right) \leq \frac{2}{\pi} \cdot \frac{1}{\kappa}.$$

Using $P[U \not\parallel V] \leq \sum_{k=1}^{K} P[u_k \not\parallel v_k] + \sum_{k=2}^{K} P[u_1 \parallel u_k]$ completes the proof. ■

For larger clusters, the bound (5) becomes inefficient and a different approach is required.

Separating two balls. With the same notation as above note that each of the two clusters is contained in the intersection of a ball of radius at most δ with a linear subspace of dimension at most K. In slight abuse of notation we denote these intersections, i.e., these "K-dimensional balls", by U and V and the probability of separating them with a hyperplane through the midpoint m of their centers u and v by $P[U\|V]$.

Theorem 3 [Separating two large but still parsimonious clusters] *Assume that $\delta \leq r$. Set $t_{2K} = (\delta/r) \cdot \sqrt{2K}$ and $s_{2K} = t_{2K}\sqrt{(2K-1)/(2K - t_{2K}^2)}$. Moreover, for any fixed $0 < \varepsilon < 1$, we write $\theta_{2K} = 2\exp[-\frac{2K-1}{2}(\varepsilon^2/2 - \varepsilon^3/3)]$. Then*

$$P\left[\ |Z| > s_{2K}\sqrt{1+\varepsilon}\ \right] \cdot (1 - \theta_{2K}) \ \leq\ P[U\|V]\ \leq\ P[\ |Z| > s_{2K}\sqrt{1-\varepsilon}\ ] + \theta_{2K} \tag{6}$$

where Z denotes a standard Gaussian random variable.

Asymptotically for $K \to \infty$, assuming $(\delta/r) \cdot \sqrt{2K} \to t$ we have $s_{2K} \to t$ and $\theta_{2K} \to 0$. Letting $\varepsilon \to 0$ in (6) we get $P[U\|V] \to P[\ |Z| > t]$.

As simple computations show, the transition from reasonable, positive (for $\delta \leq \frac{1}{8} \cdot r/\sqrt{2K}$ and $K \geq 25$) to practically vanishing (for $\delta \geq 8 \cdot r/\sqrt{2K}$ and $K \geq 125$) probabilities $P[U\|V]$ is quite sharp in terms of the size of the spread δ as compared to the ratio $r/\sqrt{2K}$. However, the half-distance r may grow as large as $\simeq \sqrt{D_0}$ allowing for large deviations δ from the centers u and v of the clusters all while keeping reasonable, positive probability of separation.

In short, in a parsimonious setting (with moderate $K \ll D_0$), larger clusters can be separated even at random initialization with reasonable probability.

Proof Similar to the proof of Lemma 1 move and rotate coordinates in such as way that all points of the two clusters U and V come to lie in the subspace spanned by the first $2K$ coordinate axes and u and v lie on the first coordinate axis. Then, $P[U\|V]$ depends only on the first $2K$ coordinates of the normal (weight) vector w.

Now consider the set S of rescaled normals $w^* = (r/\|w\|)w$ in the *upper* half-space, i.e. with positive $x[1]$-coordinate, for which $H(w, w^\top m)$ separates U and V. Obviously, S possesses a rotational symmetry about the $x[1]$-coordinate axis and forms, therefore, a hyper-spherical cap. The probability of a rescaled normal w^* to lie in the set S is proportional to the area of S. A moment's thought and a sketch (see Figure 2 far right) reveal that the height of the cap S amounts to $h = r - \delta$. Therefore, the rescaled normal w^* lies in S exactly when its first coordinate $w^*[1]$ amounts to at least δ. With a symmetrical argument for the case $w[1] < 0$ we have, in summary,

$$P[U\|V] = P[w^* \in S] = P[w[1] \geq \tfrac{\delta}{r}\|w\|].$$

Consider the standard chi-squared random variable $Y_{2K} = \sum_{k=2}^{2K}(w[k]/s)^2$. It is well known that it is well-centered around its mean $2K - 1$. More precisely, considering the "typical" event E_{2K}

$$E_{2K} = \{|Y_{2K} - (2K-1)| < (2K-1)\varepsilon\}$$

we have $P[\overline{E_{2K}}] \le \theta_{2K}$ where $0 < \varepsilon < 1$ is arbitrary. Conditioning on E_{2K} we have

$$\begin{aligned} P[w[1] \ge \tfrac{\delta}{r}\|w\| \;\Big|\; E_{2K}] &= P[w[1]^2 \ge \tfrac{\delta^2}{r^2}(w[1]^2 + s^2 Y_{2K}) \;\Big|\; E_{2K}] \\ &\le P\left[w[1]^2\left(1 - \tfrac{\delta^2}{r^2}\right) \ge \tfrac{\delta^2}{r^2} s^2 (2K-1)(1-\varepsilon) \;\Big|\; E_{2K}\right]. \end{aligned}$$

Note that the event of the probability on the right hand side is independent on E_{2K} since it involves only $w[1]$, but none of the $w[k]$ ($k \ge 2$). Therefore, we may drop the conditioning on E_{2K}. Also, $w[1]/s$ is a standard Gaussian variable. Therefore, with $(\delta/r) = t_{2K}/\sqrt{2K}$

$$P[w[1] \ge \tfrac{\delta}{r}\|w\| \;\Big|\; E_{2K}] \le P\left[|Z| \ge t_{2K}\sqrt{\tfrac{2K-1}{2K}}\left(1 - \tfrac{t_{2K}^2}{2K}\right)^{-1/2}\sqrt{1-\varepsilon}\right] = P\left[|Z| \ge s_{2K}\sqrt{1-\varepsilon}\right].$$

Using $P[E_{2K}] \le 1$ and $P[w[1] \ge \tfrac{\delta}{r}\|w\| \;\big|\; \overline{E_{2K}}] \cdot P[\overline{E_{2K}}] \le P[\overline{E_{2K}}] \le \theta_{2K}$, the law of total probability implies the upper bound of (6). The lower bound is obtained in an analogous way, using $P[E_{2K}] \ge 1 - \theta_{2K}$. ■

Separation with random offsets. In the discussion above, separation was always attempted using hyperplanes passing through a center of gravity. In contrast, if the hyperplane offset is chosen at random (instead of forcing it to pass through g), the resulting hyperplane will, with non-negligible probability, lie at distance at least r from the mini-batch center. By (4), such a hyperplane separates any given pair of points with probability strictly below $\frac{1}{2}$, making the simultaneous separation of all cross-cluster pairs very unlikely.

These results formalize the following principle: if real-world data (such as natural images) are concentrated near low-dimensional manifolds (parsimony), then hyperplanes constrained to pass through the mini-batch center — as in BN — achieve a substantial probability of separating such clusters already at random initialization, as quantified by (5) for small K and by (6) for large K.

4. Conclusions

We have argued that BN is not merely an optimization aid but an unsupervised learning mechanism that adapts a network's geometry to the data. Four conclusions follow.

Geometric anchoring. BN's centering forces each neuron's hyperplane through the mini-batch mean, so decision boundaries pass through the data's center of mass rather than ignoring it. **Separation at initialization**. This anchoring matters before any training. For low-dimensional (parsimonious) data, BN makes a random hyperplane likely to separate clusters—whereas without BN, separation is exponentially unlikely in high dimensions. **Finer resolution near the data**. In deeper layers, BN's scaling disperses folding boundaries more evenly around the data, concentrating partition regions where they matter (see appendix A). **A label-free rule**. All of these effects depend only on the input statistics (mean and variance), not on labels or the loss. BN is therefore a built-in unsupervised rule that supplies a smart initialization and preserves favorable geometry throughout training.

In short, BN succeeds by aligning a network's internal geometry with the data distribution — an explanation orthogonal to, and complementary with, optimization-based accounts.

References

R. Balestriero and R. G. Baraniuk. A spline theory of deep networks. In *Proc. Int. Conf. Mach. Learn.*, volume 80, pages 374–383, Jul. 2018.

Randall Balestriero and Richard Baraniuk. Batch normalization explained. *arXiv preprint arXiv:2209.14778*, 2022.

Randall Balestriero and Richard G. Baraniuk. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109(5):704–727, 2021. doi: 10.1109/JPROC.2020.3042100.

Randall Balestriero, Romain Cosentino, Behnaam Aazhang, and Richard Baraniuk. The geometry of deep networks: Power diagram subdivision. In *Advances in Neural Information Processing Systems*, 2019.

Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. In *Advances in Neural Information Processing Systems*, pages 7694–7705, 2018.

Caglar Gulcehre, Marcin Moczulski, Francesco Visin, and Yoshua Bengio. Mollifying networks. *arXiv preprint arXiv:1608.04980*, 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Thomas Hofmann, Ming Zhou, and Klaus Neymeyr. Exponential convergence rates for batch normalization: The power of length-direction decoupling in non-convex optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 806–815, 2019.

Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493, 2018.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

Thomas Zaslavsky. *Facing up to arrangements: Face-count formulas for partitions of space by hyperplanes: Face-count formulas for partitions of space by hyperplanes*, volume 154. American Mathematical Soc., 1975.

Appendix A. Folding of the Boundaries in BN by Layer 2

A.1. Condensing and expanding layers

Recall from Section 2 that a layer partitions its input space into so-called *linear regions* which can be identified by the signs which the pre-activations $h_{\ell,k}$ attain at the points in such a region. We start by establishing the useful facts regarding these regions from section 2, there labelled as (A) - (E). Since they are valid for any layer ℓ we choose $\ell = 0$ and suppress the index ℓ for simplicity.

Recall that we call a layer *centered*, if the boundaries of all linear regions intersect in at least one common point q. Also, we call a region *all-positive* if all pre-activations attain strictly positive values. Similar for *all-negative*.

Lemma 4 *Consider layer $\ell = 0$ with D_1 pre-activations $h_k(z; w_k, c_k, b_k)$ and input $z \in \mathbb{R}^{D_0}$. Assume that this layer is centered. Then, (A) all linear regions are unbounded.*

Proof We write $h_k(z)$ for $h_k(z; w_k, c_k, b_k)$ when the parameters are clear from the context.

(A) Choose one point q common to all boundaries and consider an arbitrary point y in the interior of any region. Pick any hyperplane H_k and drop the index k for the rest of this proof. Since q lies on H but y does not, the line through q and y intersects H only in q.

While clear by intuitive geometry, this can also be seen by computation as follows: The line through q and y is formed by the points $y_t = q + t(y - q)$ for $t \in \mathbb{R}$. We find that $b \cdot h(y_t) = w^\top y_t - c = w^\top q - c + tw^\top(y - q) - tc + tc = b(h(q) + th(y) - th(q))$. In other words, $h((1-t)q + ty) = (1-t)h(q) + th(y)$, a property which is related to the concept of homotopy and should not be confused with linearity. Since $h(q) = 0 \neq h(y)$ we find that $h(y_t)$ vanishes only if $t = 0$. Consequently, all regions are unbounded. ■

Lemma 5 *In the notation of Lemma 4 assume that all weights $w_k[j]$ are i.i.d. variables drawn from some continuous distribution.*

- *Condensing layer: If $D_1 \leq D_0$, then, almost surely*
  (B) there is an all-positive and an all-negative linear region in the input space,
  (C) the layer is centered, and if $D_1 = D_0$ then the common point q is unique.
- *Expanding layer: If $D_1 > D_0$ and if in addition all offsets c_k are continuous i.i.d. variables, then almost surely,*
  (D) the layer is not centered, and
  (E) there exists a non-empty, bounded linear region.

The relevance of (B) becomes apparent when noticing that in the linear regions where all pre-activations are positive, none of the activations of the neurons alter their input, meaning that the layer preserves in this region as much information as is possible with D_1 neurons. To the contrary, in the linear regions where all pre-activations are negative, the layer maps all points to 0 and any distinction between them is lost to the DN.

The distinction between condensing and expanding layers as stated in Lemma 5 will be important when we analyze separation properties and the effect of BN on higher-order boundaries in the sequel.

Proof We write $h_k(z)$ for $h_k(z; w_k, c_k, b_k)$ when the parameters are clear from the context.

(C): By induction using intuitive geometry: We claim that, almost surely, the intersection $\zeta_j = \bigcap_{k=1}^{j} H_k$ is a non-empty affine subspace of dimension $D_0 - j$ for $j = 1 \dots D_1$. This is obvious for $j = 1$. Assuming by induction that the claim holds for $j \leq D_1 - 1$, the hyperplane H_{j+1} will intersect ζ_j in at least one point and, in fact, $\zeta_{j+1} = \zeta_j \cap H_{j+1}$ will be a non-empty affine subspace of dimension $D_0 - j - 1 \geq 0$ unless H_{j+1} is parallel to ζ_j, which occurs with zero probability. This completes the inductive argument. In particular, ζ_{D_1} is non-empty.

By rigorous computation: Any point q common to all boundaries satisfies $h_k(q) = 0$, or equivalently $w_k^\top q - c_k = 0$, $(k = 1 \dots D_1)$. This constitutes a linear system for the D_0 unknown coordinates of q with coefficient matrix W consisting of the D_1 rows $w_k^\top$, i.e., $Wq = c$ with $c[k] = c_k$. Almost surely, W is full rank and the claim follows since $D_1 \leq D_0$.

(B): Reusing notation of (C), the linear system $Wx = c + v$ possesses solutions for any vector v, implying that any combination of signs of $h_k(x)$ (matching the signs of the coordinates $v[k]$) occurs in the input space, especially all-positive and all-negative.

(D): We could establish the claim considering the linear system with coefficient matrix W as in (C). We choose a different path, which also allows to address (E).

Consider the first D_0 hyperplanes H_k $(k = 1 \dots D_0)$. According to (C), they intersect in one unique common point q. In fact, denoting the truncated coefficient matrix consisting of the rows w_k $(k = 1 \dots D_0)$ by W_{tr}, which is almost surely invertible, and the truncated offset vector by $c_{\text{tr}} = (c[1] \dots c[D_0])^\top$, we have $q = W_{\text{tr}}^{-1} c_{\text{tr}}$.

Now, we change the coordinates x to new coordinates $\tilde{x} = W_{\text{tr}} x - c_{\text{tr}}$. In these new coordinates, the first D_0 neurons read as $\tilde{h}_k(\tilde{x}) = \tilde{x}_k / b_k$. Up to a factor, the new coordinate $\tilde{x}_k$ of a point corresponds to the signed distance (in original coordinates) of this point from the hyperplane H_k (cpre. (1)). In particular, in new coordinates the first D_0 hyperplanes form the principal coordinate-hyperplanes intersecting at $\tilde{q} = 0$.

Also, the next neuron with index $m = D_0 + 1$ reads as $h_m(x) = (w_m^\top x - c_m)/b_m$ in original coordinates, which translates into new coordinates as $\tilde{h}_m(\tilde{x}) = (w_m^\top x - c_m)/b_m = (w_m^\top W_{\text{tr}}^{-1}(\tilde{x} + c_{\text{tr}}) - c_m)/b_m = (\tilde{w}_m^\top \tilde{x} - \tilde{c}_m)/b_m$. As linear combinations of i.i.d. continuous random variables, its weights forming the vector $\tilde{w}_m = w_m^\top W_{\text{tr}}^{-1}$ and its offset $\tilde{c}_m = -w_m^\top W_{\text{tr}}^{-1} c_{\text{tr}} + c_m$ are all non-zero, almost surely.

Note that this coordinate change may alter length and angles, as well as shapes of regions. Being linear, invertible and continuous in both directions, however, it does not change topological properties such as boundaries, interiors, intersections, connectivity, nor certain other properties such as affinity or boundedness of a set.

Let's consider first the most simple case where $\tilde{w}_m$ and $\tilde{c}_m$ are all strictly positive. Then, the point $\tilde{s}$ on $\tilde{H}_m$ closest to the (new) origin, i.e., the orthogonal projection of the (new) origin onto $\tilde{H}_m$, is $\tilde{s} = \tilde{w}_m \cdot (\tilde{c}_m / \|\tilde{w}_m\|^2)$, with all its coordinates strictly positive.

Since the (new) origin does not lie on $\tilde{H}_m$ the layer is not centered, establishing (D).

Consider the interior $\tilde{A}$ of the region of points $\tilde{x}$ defined by $\tilde{h}_k(\tilde{x}) > 0$ $(k = 1 \dots D_0)$ and $\tilde{h}_m(\tilde{x}) < 0$. We claim that $\tilde{A}$ is bounded and non-empty, and the same will hold for the corresponding set A in original coordinates.

To this end, consider the points $\tilde{s}_t = t \cdot \tilde{s}$ $(0 < t < 1)$ on the line segment joining the origin with $\tilde{s}$. They all lie in $\tilde{A}$ since $\tilde{h}_k(\tilde{s}_t) = b_k \tilde{s}[k] > 0$ $(k = 1 \dots D_0)$ and

$$b_m \tilde{h}_m(\tilde{s}_t) = \tilde{w}_m^\top \tilde{s}_t - \tilde{c}_m = t \cdot \tilde{w}_m^\top (\tilde{w}_m \cdot \tilde{c}_m / \|\tilde{w}_m\|^2) - \tilde{c}_m = (t-1)\tilde{c}_m < 0$$

for $0 < t < 1$. Consequently, $\tilde{A}$ is non-empty. Also, for any $\tilde{x}$ in $\tilde{A}$ and $1 \leq n \leq D_0$ we have

$$\tilde{w}_m[n]\tilde{x}[n] < \sum_{k=1}^{D_0} \tilde{w}_m[k]\tilde{x}[k] = \tilde{w}_m^\top \tilde{x} = b_m \tilde{h}_m(\tilde{x}) + \tilde{c}_m < \tilde{c}_m.$$

With all coordinates of $\tilde{x}$ being bounded, $\tilde{A}$ is bounded and so is A. Further hyperplanes H_j ($j \geq D_0 + 2$), if they exist, can only further reduce the size of A, thus establishing (E) in this simple case.

Recall that due to the distortion caused by the change of coordinates, one should not expect that s, the point $\tilde{s}$ in original coordinates, is the one on H_m closest to the common point q, nor should one expect the boundary of the finite region to contain the point of H_m that is actually closest to the original origin.

Finally, if $\tilde{c}_m$ and/or some of the coordinates of $\tilde{w}_m$ should be strictly negative (instead of positive) replace first $\tilde{h}_m$ by $-\tilde{h}_m$ to render $\tilde{c}_m > 0$, if needed. Next, flip the signs of all $\tilde{x}_k$ and corresponding $\tilde{w}_k$, if needed, to render the weight vector of the neuron m in the renewed coordinates strictly positive. Then, proceed as above to establish (E) in general.

(B): At long last, a geometric argument establishing (B) leverages the approach of (E). To this end, add auxiliary neurons to the layer, if needed, so that $D_0 = D_1$. Change coordinates as before to $\tilde{x} = W_{\text{tr}}x - c_{\text{tr}}$ where the new coordinates $\tilde{x}[k]$ of a point are up to a positive factor equal to its signed distances to the hyperplanes, including the auxiliary ones. In new coordinates, the all-positive linear region is the first hyper-octant. In this linear region, even the auxiliary neurons and certainly all actual neurons have positive pre-activation. Similar for "negative". ■

A.2. Interaction between two BN-centered layers

We turn now to the interaction between two consecutive DN layers where we assume that the neurons of both layers are centered with common point g, but not necessarily normalized. We assume further that the weights are i.i.d. continuous random variables.

Summarizing the results of appendix A.1: a BN layer never produces a *finite* linear region in its input space because all hyperplanes pass through the common point g and, provided it is *condensing*, it produces almost surely an all-positive and an all-negative linear region. The same holds, actually, for any condensing layer with same kind of weights but unconstrained offsets, almost surely. In contrast, if one were to replace the constrained offsets of an *expanding* BN-layer with i.i.d. continuous offsets that layer would produce at least one finite region and it would not be centered, almost surely.

Notation We provide our arguments for the first two layers but they apply to any two consecutive layers. We *drop the layer-index* $\ell = 0$ of the hyperplanes and their parameters for improved readability since the arguments of this section actually apply to any layer and since the parameters of the subsequent layer do not come explicitly into play.

To be precise, we denote the input to the first layer by $z_0 \in \mathbb{R}^{D_0}$ and its output by $z_1 \in \mathbb{R}^{D_1}$, which in turn serves as the input to the second layer. Each output-coordinate $z_1[k]$ is computed via the k-th neuron via $z_1[k] = a(h_k(z_0; w_k, w_k^\top g, b_k))$. The BN normalization factors of layer 1 are $b_k = \sigma_k$ where

$$\sigma_k^2 = \text{mean}_{z_0 \in \mathcal{B}} \left(w_k^\top z_0 - w_k^\top g\right)^2 = \|w_k\|^2 \cdot \text{mean}_{z_0 \in \mathcal{B}} \left(d(z_0; w_k, w_k^\top g)\right)^2. \tag{7}$$

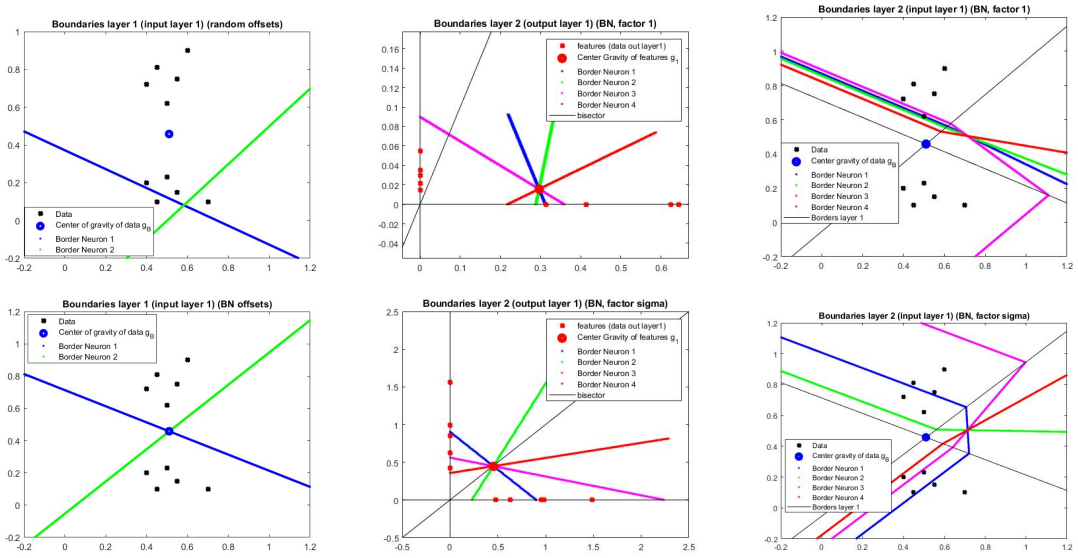

Figure 3: Effect of BN's ingredients for ReLU activation, demonstrated with a toy-example (two clusters consisting of but a handful of points, $D_0 = D_1 = 2$, $D_2 = 4$. Left: Proximity to data: BN (bottom) will place the common center closer to the data (see Section 1) than random offsets (top). This leads to strong probabilities of separation (see Section 2). Middle: When using the scaling factor σ_k (bottom), data will be more evenly placed in all coordinates of the output space of the layer as compared to using a factor 1, with same weights w_k. Thus, their center of gravity will be closer to the bisector (shown in black), resulting in a more even spread of folding locations. Also folding tends to be closer to the data. Note: While the distances from the data to the centered hyperplanes are roughly equal (see image on the left, bottom), here we have $\|w_1\| \simeq 10\|w_2\|$. Right: In the input space to the layer, the more even spread of the folding locations when using the scaling factor σ_k (bottom) tends to result in regions with better resolution and boundaries closer to the data as compared to using a factor 1 (top).

BN factor and weight-vector length. A first, elementary consequence of the normalization by σ_k is that the pre-activation of neuron k becomes independent of the length of its weight vector w_k. We may write:

$$h_k(z_0; w_k, w_k^\top g, \sigma_k) = \frac{w_k^\top z_0 - w_k^\top g}{\sigma_k} = \frac{d(z_0; w_k, w_k^\top g)}{\sqrt{\text{mean}_{z_0 \in \mathcal{B}} \big(d(z_0; w_k, w_k^\top g)\big)^2}}. \tag{8}$$

Note that the norm $\|w_k\|$ is factored out by the BN scaling $b_k = \sigma_k$ and that the signed distance d depends only on the direction of the normal on the hyperplane $H(w_k, w_k^\top g)$, not on its length.

An invariance of the second-order boundaries. A second, less obvious effect of the choice of scaling factors b_k concerns the regions created by the subsequent (second) layer. We begin by recalling a property that is *independent* of the choice of the factors b_k.

The DN input space is partitioned by the neurons of layer 1, with weights w_k, offsets $w_k^\top g$ and factors b_k $(k = 1, \ldots, D_1)$, into regions separated by the *first-order boundaries*

introduced in Section 2; cf. the affine spline perspective of Balestriero and Baraniuk (2021); Balestriero et al. (2019). These first-order boundaries depend only on the hyperplanes $H_k(w_k, w_k^\top g)$ and are therefore clearly independent of the specific values of the scaling factors b_k in the pre-activations. The neurons of layer 2 further refine this partition, creating *second-order boundaries* in the input space of layer 1.

To understand the influence of the BN parameters on these second-order boundaries, we allow the neurons of layers 1 and 2 to use arbitrary positive factors b_k, while still enforcing BN offsets. To be precise, all hyperplanes of layer 1 pass through the center of gravity g of the batch while all hyperplanes of layer 2 pass through the center g_1 of the features z_1 in the input space of layer 2, whose coordinates $g_1[k]$ are given in (9) below.

Since we consider continuous piecewise-linear activations a, the second-order boundaries are themselves continuous, piecewise linear surfaces that *fold* along the first-order boundaries (see again Balestriero and Baraniuk (2021); Balestriero et al. (2019)). Moreover, all second-order boundaries contain the set G_1 of points that are mapped by the first layer exactly to g_1, i.e.,

$$x \in G_1 \quad \Longleftrightarrow \quad a\big((w_k^\top x - w_k^\top g)/b_k\big) = g_1[k] = \frac{1}{|\mathcal{B}|}\sum_{z_0 \in \mathcal{B}} a\big((w_k^\top z_0 - w_k^\top g)/b_k\big) \tag{9}$$

for all neurons k of the first layer.

While g_1 itself depends on the choice of the factors b_k (cf. (9) and Figure 3, middle), the set G_1 does not, as long as $b_k \geq 0$ for all k (see Figure 3, bottom). The reason is that for our class of activations

$$a(\lambda t) = \max(\eta\lambda t, \lambda t) = \lambda \max(\eta t, t) = \lambda a(t), \qquad \lambda \geq 0, \tag{10}$$

so that the factors b_k cancel out in the defining equations of G_1.

In summary, almost surely, *all second-order boundaries pass through the set G_1, which is independent of the choice of the (positive) scaling factors b_k in the first layer.*

Structure of G_1. With the appropriate choice of y the following equation defines G_1, provided $b_k > 0$.

$$(w_k^\top x - w_k^\top g) = y[k] \quad (k = 1 \dots D_1) \qquad \text{short:} \qquad W \cdot (x - g) = y \tag{11}$$

where the $D_0 \times D_1$-matrix W consists of the row vectors $w_k^\top$. Note that W is full rank, almost surely, since its entries are i.i.d. continuous random variables with the well-known implications on the solutions x: unique solution if $D_0 = D_1$, $D_0 - D_1$ free variables $x[k]$ if $D_0 > D_1$ (condensing layer), and typically no solution if $D_0 < D_1$ (expanding layer) except if y takes on a specific form.

The appropriate choices of y are as follows, depending on the choice of activation:

[ReLU] Choose $y[k] = b_k g_1[k]$ which are all non-negative.

[leaky ReLU] Since this activation a is invertible, choose $y[k] = a^{-1}(b_k g_1[k])$.

[Absolute Value] Choose any combination of $y[k] = (\pm b_k) g_1[k]$ and obtain 2^{D_1} symmetrical solution sets (see Figure 4 right two plots).

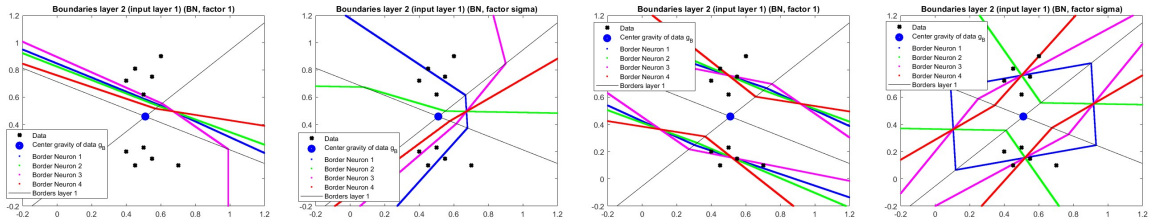

Figure 4: Effect of BN's ingredients for leaky ReLU and Absolute Value activations, demonstrated with the same toy example as in Figure 3. In color, the secondary boundaries in the input space of the network, as created by layer 2, all with centered hyperplanes but with different scaling factors. From left to right: leaky ReLu with scaling factor $b_k = 1$; leaky ReLu with $b_k = \sigma_k$ (BN); Absolute Value with $b_k = 1$; Absolute Value with $b_k = \sigma_k$ (BN).

BN-factor and the shape of second order regions (CPA). The influence of the normalization factor used in BN lies in the dispersion of folding locations as we are about to explain in a simple setting (see Figure 3 right). To this end, assume for a moment the most simple of settings where the mini-batch consists of two clusters as in Section 3, $\mathcal{B} = U \cup V$, and vanishing spread $\delta = 0$. In other words, all points of the cluster U are identical to u and analog for V. Consider a neuron $h_k(z_0; w_k, w_k^\top g, \sigma_k)$ of the first layer and denote the (signed) distance of u from its hyperplane by $\lambda_k = d(u; w_k, w_k^\top g)$. Note that $\lambda_k \neq 0$, almost surely. By symmetry, and since the center g lies in their midpoint $g = m = (u + v)/2$, the distance from v amounts to $-\lambda_k$. Thus, we find that $\text{mean}_{z_0 \in \mathcal{B}} \, (d(z_0; w_k, w_k^\top g))^2$ equals λ_k^2, and that the coordinates of the output of the neuron (feature) on u amount to (compare to (8))

$$u_1[k] = a(h(u; w_k, w_k^\top g, \sigma_k)) = a(\lambda_k/|\lambda_k|) \in \{1, -\eta\} \qquad \text{and} \qquad u_1[k] + v_1[k] = 1 - \eta \quad (12)$$

Recall (10) and that $\eta = 0$ for ReLU, $0 < \eta < 1$ for leaky ReLU and $\eta = -1$ for Absolute Value.

With the same number of points coinciding with u and with v, respectively, we find that all coordinates of g_1 are *equal* when using first layer neurons with the factors $b_k = \sigma_k$; in fact $g_1[k] = (1 - \eta)/2$ for all k. However, when using the factors $b_k = 1$ (no normalization) then the coordinates $g_1[k] = |\lambda_k| \cdot \|w_k\|(1 - \eta)/2$ will depend on the parameters of the neuron, i.e., on their orientation via λ_k as well as on the length of the normal vector w_k.

A similar situation persists, with small deviations of the coordinates $g_1[k]$ from $(1-\eta)/2$, and with large probability on the separation if the spread δ is allowed to be positive, but small (compare to Theorem 2). As illustrated in Figure 3 (middle and bottom), a center g_1 with roughly equal coordinates (as is the case with BN-normalization and continuous piecewise-linear activation) leads to *better resolution* due to a larger spread of the folding locations and tends to create boundaries *closer to the mini-batch data*.

Arbitrary layer. The argumentation of this section applies to any layer ℓ, mutatis mutandis, replacing iteratively z_0 by z_ℓ which are computed iteratively from layer to layer, g by the average output of the batch-data of layer ℓ, i.e., $g_\ell = \text{mean}_{z_0 \in \mathcal{B}} \, (z_\ell)$, and G_1 by G_ℓ, the set of points which are mapped by layer $\ell + 1$ to $g_{\ell+1}$.