A Rotation Invariance w.r.t. the Initialized Weights

In this paper, we analyze neural networks trained on high-dimensional data that lies on a low dimensional linear subspace denoted by P. We assume that the dimension of P is $d-\ell$. Throughout the paper it will be more convenient to analyze data which lies on the subspace $M = \text{span}(\{e_1, \dots, e_{d-\ell}\})$, because then the "off manifold" directions correspond exactly to certain coordinates of the input. In this section we show that we can essentially analyze the data as if it is rotated to lie on M, and it would imply the same consequences as the original data from P.

Theorem A.1. *Let $P \subseteq \mathbb{R}^d$ be a subspace of dimension $d-\ell$, and let $M = span\{e_1, \dots, e_{d-\ell}\}$. Let R be an orthogonal matrix such that $R \cdot P = M$, let $X \subseteq P$ be a training dataset and let $X_R = \{R \cdot x : x \in X\}$. Assume we train a neural network $N(x) = \sum_{i=1}^m u_i \sigma(w_i^\top x)$ as explained in Section 3, and denote by N^X and N^{X_R} the network trained on X and X_R respectively for the same number of iterations. Let $x_0 \in P$, then we have:*

1. *W.p. p (over the initialization) we have $\left\|\Pi_{P^\perp}\left(\frac{\partial N^X(x_0)}{\partial x}\right)\right\| \geq c$ (resp. $\leq c$) for some $c \in \mathbb{R}$, iff w.p. p also $\left\|\Pi_{M^\perp}\left(\frac{\partial N^{X_R}(Rx_0)}{\partial x}\right)\right\| \geq c$ (resp. $\leq c$).*
2. *For any $c, p \geq 0$, w.p. p (over the initialization) there exists $z \in P^\perp$ with $\|z\| = c$ such that $\text{sign}(N^X(x_0 + z)) \neq \text{sign}(N^X(x_0))$, iff w.p. p there exists $z' \in M^\perp$ with $\|z'\| = c$ such that $\text{sign}(N^{X_R}(Rx_0 + z')) \neq \text{sign}(N^{X_R}(Rx_0))$.*

Proof. Denote by $\mathbf{w}_{1:m} := (w_1, \dots, w_m)$ and by $R\mathbf{w}_{1:m} = (Rw_1, \dots, Rw_m)$. Let $\mathbf{w}_{1:m}^{(t)}$ the weights of the network trained on the dataset X where $\mathbf{w}_{1:m}^{(0)}$ is some initialization, and $\tilde{\mathbf{w}}_{1:m}^{(t)} = (\tilde{w}_1^{(t)}, \dots, \tilde{w}_m^{(t)})$ the weights of the network trained on X_R and initialized at $R\mathbf{w}_{1:m}^{(0)}$. In the proof, when taking derivatives w.r.t. the w_i's we will explicitly write $N(x, \mathbf{w}_{1:m})$.

We first show by induction on the number of training steps that $\tilde{\mathbf{w}}_{1:m}^{(t)} = R\mathbf{w}_{1:m}^{(t)}$. For $t = 0$ it is clear by the assumption on the initialization. Assume it is true for t, then we have for some $x \in X$:

$$
\begin{aligned}
\frac{\partial N(Rx, \tilde{\mathbf{w}}_{1:m}^{(t)})}{\partial w_i} &= u_i \sigma'(\langle \tilde{w}_i^{(t)}, Rx \rangle) Rx \\
&= u_i \sigma'(\langle Rw_i^{(t)}, Rx \rangle) Rx \\
&= u_i \sigma'(\langle w_i^{(t)}, x \rangle) Rx \\
&= R \cdot \frac{\partial N(x, \mathbf{w}_{1:m}^{(t)})}{\partial w_i} .
\end{aligned}
$$

This is true for every $i \in [m]$ and for every $x \in X$. Also note that by our induction assumption we have:

$$
N(x, \mathbf{w}_{1:m}^{(t)}) = \sum_{i=1}^m u_i \sigma(\langle w_i^{(t)}, x \rangle) = \sum_{i=1}^m u_i \sigma(\langle Rw_i^{(t)}, Rx \rangle) = N(Rx, \tilde{\mathbf{w}}_{1:m}^{(t)}) . \tag{1}
$$

Finally, the derivative of the loss on a single data point $x \in X$ with label y can be written as:

$$
\frac{\partial L\left(N(x, \mathbf{w}_{1:m}^{(t)}) \cdot y\right)}{\partial w_i} = L'\left(N(x, \mathbf{w}_{1:m}^{(t)}) \cdot y\right) \cdot \frac{\partial N(x, \mathbf{w}_{1:m}^{(t)})}{\partial w_i} ,
$$

where the first term depends only on the value of $N(x, \mathbf{w}_{1:m}^{(t)})$. Hence, taking a single gradient step of N with weights $\mathbf{w}_{1:m}^{(t)}$ and dataset X will change the weights by the same term up to multiplication by R as if taking a gradient step with with weights $\tilde{\mathbf{w}}_{1:m}^{(t)}$ and dataset X_R. This finishes the induction.

Let $\mathbf{w}_{1:m}^{(0)}$ be an initialization for the training of N^X, where there exists $z \in P^\perp$ with $\|z\| = c$ such that $\text{sign}(N^X(x_0 + z)) \neq \text{sign}(N^X(x_0))$. Then, by Eq. (1) the initialization $R\mathbf{w}_{1:m}^{(0)}$ for the training of N^{X_R} is such that for $z' = Rz$ we have $\|z'\| = c$ and $\text{sign}(N^{X_R}(Rx_0 + z')) \neq \text{sign}(N^{X_R}(Rx_0))$. This argument holds also in the opposite direction. Let $A \subseteq \{\mathbf{w}_{1:m} \in \mathbb{R}^{d \cdot m}\}$ be the set of all initializations to N^X where there exists $z \in P^\perp$ with $\|z\| = c$ such that $\text{sign}(N^X(x_0 + z)) \neq \text{sign}(N^X(x_0))$, then by the above the set $R \cdot A = \{R\mathbf{w}_{1:m} : \mathbf{w}_{1:m} \in A\}$ are exactly all the initializations to N^{X_R} where there exists $z' \in M^\perp$ with $\|z'\| = c$ such that $\text{sign}(N^{X_R}(Rx_0 + z')) \neq \text{sign}(N^{X_R}(Rx_0))$. Since we initialize the w_i's using a Gaussian initialization which is spherically symmetric, we have that $\Pr(A) = \Pr(RA)$. This proves item (2). Item (1) follows from similar arguments (which we do not repeat for conciseness). □

Under the assumption that the data lies on $M = \text{span}\{e_1, \dots, e_{d-\ell}\}$, and no regularization is used, we can show that the weights of the first layer projected on $M^\perp$ do not change during training. This is an essential part of the proofs, as it allows us to analyze those weights as random Gaussian vectors, and apply concentration bounds on them.

Theorem A.2. *Let $M = span\{e_1, \dots, e_{d-\ell}\}$. Assume we train a neural network $N(x, \mathbf{w}_{1:m}) := \sum_{i=1}^m u_i \sigma(w_i^\top x)$ as explained in Section 3 (where $\mathbf{w}_{1:m} = (w_1, \dots, w_m)$). Denote by $\hat{w} := \Pi_{M^\perp}(w)$ for $w \in \mathbb{R}^d$, then after training, for each $i \in [m]$, $\hat{w}_i$ did not change from their initial value.*

Proof. Note that for each $i \in [m]$ and $x \in M$ we have:

$$\Pi_{M^\perp}\left(\frac{\partial N(x, \mathbf{w}_{1:m})}{\partial w_i}\right) = \Pi_{M^\perp}\left(u_i \sigma'(w_i^\top x) x\right) = u_i \sigma'(w_i^\top x)\hat{x} = \mathbf{0}\ .$$

Taking the derivative of the loss we have:

$$\begin{aligned}\Pi_{M^\perp}\left(\frac{\partial L\left(N(x, \mathbf{w}_{1:m}) \cdot y\right)}{\partial w_i}\right) &= \Pi_{M^\perp}\left(L'\left(N(x, \mathbf{w}_{1:m}) \cdot y\right) \cdot \frac{\partial N(x, \mathbf{w}_{1:m})}{\partial w_i}\right) \\ &= L'\left(N(x, \mathbf{w}_{1:m}) \cdot y\right) \cdot \Pi_{M^\perp}\left(\frac{\partial N(x, \mathbf{w}_{1:m})}{\partial w_i}\right) = \mathbf{0}\ .\end{aligned}$$

The above calculation did not depend on the specific value of the w_i's. Hence, the value of the $\hat{w}_i$'s for every $i \in [m]$ did not change during training from their initial value. □

B Proofs from Section 4

Before proving the main theorem, we will first need the next two lemmas about the concentration of Gaussian random variables:

Lemma B.1. *Let $w \in \mathbb{R}^n$ such that $w \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$. Then:*

$$\mathbb{P}\left[\|w\|^2 \leq \frac{1}{2}\sigma^2 n\right] \leq e^{-\frac{n}{16}}\ .$$

Proof. Note that $\left\|\frac{w}{\sigma}\right\|^2$ has the Chi-squared distribution. A concentration bound by Laurent and Massart [Laurent and Massart, 2000, Lemma 1] implies that for all $t > 0$ we have

$$\Pr\left[n - \left\|\frac{w}{\sigma}\right\|^2 \geq 2\sqrt{nt}\right] \leq e^{-t}\ .$$

Plugging-in $t = \frac{n}{16}$, we get

$$\Pr\left[n - \left\|\frac{w}{\sigma}\right\|^2 \geq \frac{1}{2}n\right] = \Pr\left[\left\|\frac{w}{\sigma}\right\|^2 \leq \frac{1}{2}n\right] \leq e^{-n/16}\ .$$

Thus, we have

$$\Pr\left[\|w\| \leq \sigma\sqrt{\frac{n}{2}}\right] \leq e^{-n/16}\ .$$

□

Lemma B.2. *Let $w_1, \dots, w_m \in \mathbb{R}^n$ such that for all $i \in [m]$, $w_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$, then we have:*

$$\mathbb{P}\left[\left\|\sum_{i=1}^m w_i\right\|^2 \leq \frac{1}{2}m\sigma^2 n\right] \leq e^{-\frac{n}{16}}\ .$$

Proof. We denote the j-th coordinate of the vector $w_i \in \mathbb{R}^n$ by $w_{i,j}$. Note, for any $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$ we have $w_{i,j} \sim \mathcal{N}(0, \sigma^2)$. We denote by s the sum vector $s := \sum_{i=1}^{m} w_i$, and by s_j the j- th coordinate of s. By this definition, $s_j = \sum_{i=1}^{m} w_{i,j}$ is a sum of m independent Gaussian variables and therefore also a Gaussian variable. Particularly, $s \sim \mathcal{N}(\mathbf{0}, m\sigma^2 I_n)$. We use Lemma B.1 with variance $m\sigma^2$ and get that:

$$\mathbb{P}\left[\left\|\sum_{i=1}^{m} w_i\right\|^2 \leq \frac{1}{2}m\sigma^2 n\right] \leq e^{-\frac{n}{16}} .$$

□

We are now ready to prove the main theorem of this section:

Proof of Theorem 4.1. Let $M = \text{span}\{e_1, \ldots, e_{d-\ell}\}$. By Theorem A.1(1), given a training dataset $X \subseteq P$, it is enough to consider a training set $X_R = \{Rx : x \in X\}$, where R is an orthogonal matrix such that $R \cdot P = M$, and training is done over X_R. From now on, we assume that the training data, as well as x_0 lie on M, and the consequences of this proof would also imply for a dataset X and $x_0 \in P$.

The projection of the gradient on $M^\perp$ is equal to:

$$\Pi_{M^\perp}\left(\frac{\partial N(x_0)}{\partial x}\right) = \Pi_{M^\perp}\left(\sum_{i=1}^{m} u_i w_i \mathbb{1}_{\langle w_i, x_0\rangle \geq 0}\right) = \sum_{i=1}^{m} \Pi_{M^\perp}\left(u_i w_i\right) \mathbb{1}_{i\in S} = \sum_{i\in S} \Pi_{M^\perp}\left(u_i w_i\right) .$$

Denote by $\hat{w}_i = (w_i)_{d-\ell+1:d}$, the last ℓ coordinates of w_i. By Theorem A.2 we get that for every $i \in [m]$, $\hat{w}_i$ did not change from their initial value during training.

Recall that we initialized $\hat{w}_i \sim \mathcal{N}(\mathbf{0}, \frac{1}{\sqrt{d}} I_\ell)$. Note that the set S is independent of the value of the $\hat{w}_i$'s. This is because $\hat{w}_i$ does not effect the training, hence will not effect $w_i - \Pi_{M^\perp}(w_i)$. Also, after choosing x_0 we have $\langle \hat{w}_i, \hat{x}_0\rangle = 0$, since $\hat{x}_0 = \mathbf{0}$, which means that the choice of S is independent of the $\hat{w}_i$'s. We can conclude that the random variables $\hat{w}_i$ for $i \in S$ are sampled independently.

Note, since for all $i \in \{1, \ldots, m\}$, $|u_i| = \frac{1}{\sqrt{m}}$ and they are not trained, we get that $u_i\hat{w}_i$ are also Gaussian random variables with the same mean, and variance multiplied by $\frac{1}{m}$. Therefore, from Lemma B.2 we get that w.p. $\geq 1 - e^{-\ell/16}$:

$$\left\|\sum_{i\in S} u_i \hat{w}_i\right\| \geq \sqrt{\frac{1}{2}}\sqrt{\frac{kl}{dm}} .$$

Combining the above, we get:

$$\left\|\Pi_{M^\perp}\left(\frac{\partial N(x_0)}{\partial x}\right)\right\| \geq \sqrt{\frac{1}{2}}\sqrt{\frac{kl}{dm}} .$$

□

C Proofs from Section 5

Before proving the main theorem, we prove a few lemmas about concentration of Gaussian random variables:

Lemma C.1. *Let $w \in \mathbb{R}^n$ with $w \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$. Then:*

$$\Pr\left[\|w\|^2 \geq 2\sigma^2 n\right] \leq e^{-\frac{n}{16}} .$$

Proof. Note that $\left\|\frac{w}{\sigma}\right\|^2$ has the Chi-squared distribution. A concentration bound by Laurent and Massart [Laurent and Massart, 2000, Lemma 1] implies that for all $t > 0$ we have

$$\Pr\left[\left\|\frac{w}{\sigma}\right\|^2 - n \geq 2\sqrt{nt} + 2t\right] \leq e^{-t} .$$

Plugging-in $t = \frac{n}{16}$, we get

$$\Pr\left[\left\|\frac{w}{\sigma}\right\|^2 \geq 2n\right] \leq \Pr\left[\left\|\frac{w}{\sigma}\right\|^2 - n \geq n/2 + n/8\right] \leq e^{-n/16} .$$

Thus, we have

$$\Pr\left[\|w\| \geq \sigma\sqrt{2n}\right] \leq e^{-n/16} .$$

$\square$

Lemma C.2. *Let $u \in \mathbb{R}^n$, and $v \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$. Then, for every $t > 0$ we have*

$$\Pr\left[|\langle u, v\rangle| \geq \|u\|\, t\right] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right) .$$

Proof. We first consider $\langle \frac{u}{\|u\|}, v\rangle$. As the distribution $\mathcal{N}(\mathbf{0}, \sigma^2 I_n)$ is rotation invariant, one can rotate u and v to get $\tilde{u}$ and $\tilde{v}$ such that $\frac{\tilde{u}}{\|u\|} = e_1$, the first standard basis vector and $\langle \frac{u}{\|u\|}, v\rangle = \langle \frac{\tilde{u}}{\|u\|}, \tilde{v}\rangle$. Note, v and $\tilde{v}$ have the same distribution. We can see that $\langle \frac{\tilde{u}}{\|u\|}, \tilde{v}\rangle \sim \mathcal{N}(0, \sigma^2)$ since it is the first coordinate of $\tilde{v}$. By a standard tail bound, we get that for $t > 0$:

$$\Pr\left[|\langle \frac{u}{\|u\|}, v\rangle| \geq t\right] = \Pr\left[|\langle \frac{\tilde{u}}{\|u\|}, \tilde{v}\rangle| \geq t\right] = \Pr\left[|\tilde{v}_1| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right) .$$

Therefore

$$\Pr\left[|\langle u, v\rangle| \geq \|u\|\, t\right] \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right) .$$

$\square$

Lemma C.3. *Let $u \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 I_n)$, and $v \sim \mathcal{N}(\mathbf{0}, \sigma_2^2 I_n)$. Then, for every $t > 0$ we have*

$$\Pr\left[|\langle u, v\rangle| \geq \sigma_1\sqrt{2n}t\right] \leq e^{-n/16} + 2e^{-t^2/2\sigma_2^2} .$$

Proof. Using Lemma C.1 we get that w.p. $\leq e^{-n/16}$ we have $\|u\| \geq \sigma_1\sqrt{2n}$. Moreover, by Lemma C.2, w.p. $\leq 2\exp\left(-\frac{t^2}{2\sigma_2^2}\right)$ we have $|\langle u, v\rangle| \geq \|u\|\, t$. By the union bound, we get

$$\Pr\left[|\langle u, v\rangle| \geq \sigma_1\sqrt{2n}t\right] \leq \Pr\left[\|u\| \geq \sigma_1\sqrt{2n}\right] + \Pr\left[|\langle u, v\rangle| \geq \|u\|\, t\right] \leq e^{-n/16} + 2\exp\left(-\frac{t^2}{2\sigma_2^2}\right) .$$

$\square$

We are now ready to prove the main theorem of this section:

Theorem 5.1. By Theorem A.1(2), we can assume w.l.o.g. that $P = M = \text{span}\{e_1, \ldots, e_{d-\ell}\}$. We also assume w.l.o.g. that $y_0 = 1$, the case $y_0 = -1$ is proved in a similar manner. Denote by $\bar{w} := (w)_{d-\ell+1:d}$, the last ℓ coordinates of w. By Theorem A.2 we have that $\bar{w}_i$ have not changed after training from their initial value.

We can write $N(x_0 + z)$ as:

$$
\begin{aligned}
N(x_0+z) &= \sum_{i=1}^{m} u_i\sigma(\langle w_i, x_0\rangle + \langle w_i, z\rangle) \\
&= \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle + \langle w_i, z\rangle) + \sum_{i\in I_+} u_i\sigma(\langle w_i, x_0\rangle + \langle w_i, z\rangle) \\
&= \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle + \langle \bar{w}_i, \bar{z}\rangle) + \sum_{i\in I_+} u_i\sigma(\langle w_i, x_0\rangle + \langle \bar{w}_i, \bar{z}\rangle) \qquad (2)
\end{aligned}
$$

where the last equality is since $(z)_{1:d-\ell} = \mathbf{0}$, hence $\langle w, z\rangle = \langle \bar{w}, \bar{z}\rangle$ for every $w \in \mathbb{R}^d$. We will bound each term of the above separately.

For the first term in Eq. (2), where $i \in I_-$ we can write:

$$
\langle \bar{w}_i, \bar{z}\rangle = \alpha \left\|\bar{w}_i\right\|^2 + \alpha\langle \bar{w}_i, \sum_{j\neq i} \text{sign}(u_j)\bar{w}_j\rangle \ .
$$

By our assumptions, $\bar{w}_i \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{d}I_\ell\right)$ and $\sum_{j\neq i}\text{sign}(u_j)\bar{w}_j \sim \mathcal{N}\left(\mathbf{0}, \frac{m-1}{d}I_\ell\right)$, since it is a sum of $m-1$ i.i.d. Gaussian random variables, which are also symmetric hence multiplying them by -1 does not change their distribution. From Lemma B.1 we get w.p. $\geq 1 - e^{-\ell/16}$ that

$$
\alpha \cdot \left\|\bar{w}_i\right\|^2 \geq \alpha \cdot \frac{\ell}{2d} \ .
$$

From Lemma C.3, and using $t = \sqrt{\frac{(m-1)\log(dm^2)}{d}}$ we get w.p. $\geq 1 - e^{-\ell/16} + 2e^{-t^2 d/2(m-1)} = 1 - e^{-\ell/16} + 2m^{-1}d^{-1/2}$ that

$$
\begin{aligned}
\langle \bar{w}_i, \sum_{j\neq i}\text{sign}(u_j)\bar{w}_j\rangle &\leq \frac{1}{\sqrt{d}} t\sqrt{2\ell} \\
&= \frac{1}{d}\cdot\sqrt{2\ell(m-1)\log(m^2 d)} \ . \qquad (3)
\end{aligned}
$$

Applying union bound over the above two events, and for every $i \in I_-$, we get w.p. $\geq 1 - 2\left(me^{-\ell/16} + d^{-1/2}\right)$ that:

$$
\langle \bar{w}_i, \bar{z}\rangle \geq \frac{\alpha\ell}{2d} - \frac{\alpha}{d}\sqrt{2\ell(m-1)\log(m^2 d)} \ .
$$

For the second term in Eq. (2), where $i \in I_+$ we can write in a similar way:

$$
\langle \bar{w}_i, \bar{z}\rangle = -\alpha \left\|\bar{w}_i\right\|^2 + \alpha\langle \bar{w}_i, \sum_{j\neq i} \text{sign}(u_j)\bar{w}_j\rangle \ .
$$

Using the same argument as above, we get w.p $\geq 1 - 2\left(me^{-\ell/16} + d^{-1/2}\right)$ that:

$$
\langle \bar{w}_i, \bar{z}\rangle \leq -\frac{\alpha\ell}{2d} + \frac{\alpha}{d}\sqrt{2\ell(m-1)\log(m^2 d)} \ .
$$

By assuming that $\ell \geq 8(m-1)\log(m^2 d)$ we get that $\langle \bar{w}_i, z\rangle \leq 0$. Denote $C := \frac{\alpha\ell}{2d} - \frac{\alpha}{d}\sqrt{2\ell(m-1)\log(m^2 d)}$, then going back to Eq. (2), using the above bounds and applying union bound, we get w.p. $\geq 1 - 4\left(me^{-\ell/16} + d^{-1/2}\right)$ that:

$$
\begin{aligned}
N(x_0+z) &\leq \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle + C) + \sum_{i\in I_+} u_i\sigma(\langle w_i, x_0\rangle) \\
&= \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle + C) + \sum_{i\in I_+} u_i\sigma(\langle w_i, x_0\rangle) + \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle) - \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle) \\
&= \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle + C) - \sum_{i\in I_-} u_i\sigma(\langle w_i, x_0\rangle) + N(x_0) \\
&= \sum_{i\in I_-} u_i\left(\sigma(\langle w_i, x_0\rangle + C) - \sigma(\langle w_i, x_0\rangle)\right) + N(x_0) \ .
\end{aligned}
$$

Define $F_- := \{i \in I_- : \langle w_i, x_0 \rangle \geq 0\}$, and $k_- = |F_-|$. We have that:

$$\begin{aligned}
\sum_{i \in I_-} u_i \left(\sigma(\langle w_i, x_0 \rangle + C) - \sigma(\langle w_i, x_0 \rangle)\right) &\leq \sum_{i \in F_-} u_i \left(\sigma(\langle w_i, x_0 \rangle + C) - \sigma(\langle w_i, x_0 \rangle)\right) \\
&= \sum_{i \in F_-} u_i C = -\frac{k_- C}{\sqrt{m}} \ ,
\end{aligned}$$

where the first inequality is since we only sum over negative terms, and the second inequality is since both $\langle w_i, x_0 \rangle \geq 0$ (because $i \in F_-$) and $C \geq 0$ (because $\ell \geq 32(m-1)\log(m^2 d)$). Combining all of the above, we get that:

$$N(x_0 + z) \leq -\frac{k_- C}{\sqrt{m}} + N(x_0) \ . \tag{4}$$

By our assumption that $\ell \geq 32(m-1)\log(m^2 d)$ we have that

$$\begin{aligned}
C &= \alpha \left(\frac{1}{2}\frac{\ell}{d} - \sqrt{2}\sqrt{m-1}\frac{\sqrt{\ell}}{d}\sqrt{\log(dm^2)} \right) \\
&= \frac{\alpha\sqrt{\ell}}{d}\left(\frac{\sqrt{\ell}}{2} - \sqrt{2(m-1)\log(m^2 d)} \right) \\
&\geq \frac{\alpha \ell}{4d} \ .
\end{aligned}$$

Plugging in C and $\alpha = \frac{8\sqrt{m} d N(x_0)}{k_- \ell}$ to Eq. (4) we get that:

$$\begin{aligned}
N(x_0 + z) &\leq -\frac{k_- C}{\sqrt{m}} + N(x_0) \\
&\leq -\frac{k_-}{\sqrt{m}} \cdot \frac{\ell}{4d} \cdot \frac{8\sqrt{m} d N(x_0)}{k_- \ell} + N(x_0) = -N(x_0) < 0 \ ,
\end{aligned}$$

and in particular $\text{sign}(N(x_0)) \neq \text{sign}(N(x_0 + z))$.

We are left with calculating the norm of z:

$$\begin{aligned}
\|z\| &= \alpha \cdot \left\| \sum_{i \in I_-} \Pi_{M^\perp}(w_i) - \sum_{i \in I_+} \Pi_{M^\perp}(w_i) \right\| \\
&= \alpha \cdot \left\| \sum_{i=1}^{m} -\text{sign}(u_i)\Pi_{M^\perp}(w_i) \right\| \\
&= \alpha \cdot \left\| \sum_{i=1}^{m} -\text{sign}(u_i)\bar{w}_i \right\| \ .
\end{aligned}$$

Since for each $i \in [m]$, $\bar{w}_i \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{d} I_\ell\right)$, then $-\text{sign}(u_i)\bar{w}_i$ also have the same distribution, because this is a symmetric distribution. Hence, $\sum_{i=1}^{m} -\text{sign}(u_i)\bar{w}_i \sim \mathcal{N}\left(\mathbf{0}, \frac{m}{d} I_\ell\right)$ as a sum of Gaussian random variables. Using Lemma C.1 we get w.p $\geq 1 - e^{-\ell/16}$ that $\left\|\sum_{i=1}^{m} -\text{sign}(u_i)\bar{w}_i\right\|^2 \leq \frac{2m\ell}{d}$. Plugging in α we get that:

$$\|z\| \leq \sqrt{\frac{2m\ell}{d}} \cdot \frac{8\sqrt{m} d N(x_0)}{k_- \ell} = 8\sqrt{2} N(x_0) \cdot \frac{m}{k_-} \cdot \sqrt{\frac{d}{\ell}} \ .$$

□

D Proofs for Section 6

For proving the main theorem, we will use the following lemma that upper bounds the norm of a sum of Gaussian random variables:

Lemma D.1. *Let $w_1, .., w_m \in \mathbb{R}^n$ such that for all $i \in [m]$, $w_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$, then we have:*

$$\mathbb{P}\left[\left\|\sum_{i=1}^{m} w_i\right\|^2 \geq 2m\sigma^2 n\right] \leq e^{-\frac{n}{16}}$$

Proof. We denote the j-th coordinate of the vector $w_i \in \mathbb{R}^n$ by $w_{i,j}$. Note, for any $i \in [m]$ and $j \in [n]$ we have $w_{i,j} \sim \mathcal{N}(0, \sigma^2)$. We denote by s the sum vector $s := \sum_{i=1}^{m} w_i$, and by s_j the j-th coordinate of s. By this definition, $s_j = \sum_{i=1}^{m} w_{i,j}$ is a sum of m independent Gaussian variables and therefore also a Gaussian variable. Therefore, $s \sim \mathcal{N}(\mathbf{0}, m\sigma^2 I_n)$. We use Lemma C.1 with variance $m\sigma^2$ and get that:

$$\mathbb{P}\left[\left\|\sum_{i=1}^{m} w_i\right\|^2 \geq 2m\sigma^2 n\right] \leq e^{-\frac{n}{16}} \; .$$

$\square$

We now prove the main theorem of this section:

Proof of Theorem 6.1. Similar to the lower bound of the norm, let $M = \text{span}\{e_1, \ldots, e_{d-\ell}\}$. By Theorem A.1(1), given a training dataset $X \subseteq P$, it is enough to consider a training set $X_R = \{Rx : x \in X\}$, where R is an orthogonal matrix such that $R \cdot P = M$, and training is done over X_R. From now on, we assume that the training data, as well as x_0 lie on M, and the consequences of this proof would also imply for a dataset X and $x_0 \in P$.

The projection of the gradient on $M^\perp$ is equal to:

$$\Pi_{M^\perp}\left(\frac{\partial N(x_0)}{\partial x}\right) = \Pi_{M^\perp}\left(\sum_{i=1}^{m} u_i w_i \mathbb{1}_{\langle w_i, x_0\rangle \geq 0}\right) = \sum_{i=1}^{m} \Pi_{M^\perp}(u_i w_i)\, \mathbb{1}_{i\in S} = \sum_{i\in S} \Pi_{M^\perp}(u_i w_i) \; .$$

Denote by $\hat{w}_i = (w_i)_{d-\ell+1:d}$, the last ℓ coordinates of w_i. By Theorem A.2 we get that for every $i \in [m]$, $\hat{w}_i$ did not change from their initial value during training.

Recall that we initialized $\hat{w}_i \sim \mathcal{N}(\mathbf{0}, \beta^2 I_\ell)$. Note that the set S is independent of the value of the $\hat{w}_i$'s. This is because $\hat{w}_i$ does not effect the training, hence will not effect $w_i - \Pi_{M^\perp}(w_i)$. Also, after choosing x_0 we have $\langle \hat{w}_i, \hat{x}_0\rangle = 0$, since $\hat{x}_0 = \mathbf{0}$, which means that the choice of S is independent of the $\hat{w}_i$'s. We can conclude that the random variables $\hat{w}_i$ for $i \in S$ are sampled independently.

Therefore, from Lemma B.2 we get that w.p. $\geq 1 - e^{-\ell/16}$:

$$\left\|\sum_{i\in S} \hat{w}_i\right\| \leq \beta\sqrt{2k\ell} \; .$$

Note, since for all $i \in [m]$, $|u_i| = \frac{1}{\sqrt{m}}$ and they are not trained, we get w.p. $\geq 1 - e^{-\ell/16}$ that:

$$\left\|\Pi_{M^\perp}\left(\frac{\partial N(x_0)}{\partial x}\right)\right\| \leq \beta\sqrt{\frac{2k\ell}{m}} \; .$$

$\square$

D.1 Explicit L_2 regularization

Proof of Theorem 6.2. As before, for this proof we rotate the data subspace P to lie on $M = \text{span}\{e_1, \ldots, e_{d-\ell}\}$ and rotate the model's weights accordingly. For a dataset $(x_1, y_1), .., (x_r, y_r)$, we train over the following objective:

$$\sum_{j=1}^{r} L(y_j \cdot N(x_j, \mathbf{w}_{1:m})) + \frac{1}{2}\lambda \left\|\mathbf{w}_{1:m}\right\|^2$$

In Theorem A.2, we showed for all (x_j, y_j) that if we train the model using the loss L we get:

$$\Pi_{M^\perp}\left(\frac{\partial L\left(N(x_j, \mathbf{w}_{1:m}) \cdot y_j\right)}{\partial w_i}\right) = 0$$

Now, we analyze the training process using the new loss which includes the regularization term. We denote by $w_i^{(t)}$ the weight vector w_i after t training steps, and by $\hat{w}_i^{(t)} := \Pi_{M^\perp}\left(w_i^{(t)}\right)$ its projection on the subspace orthogonal to M. We look at the projected gradient of $w_i^{(t)}$ w.r.t. the loss:

$$\begin{aligned}
&\Pi_{M^\perp}\left(\frac{\partial \sum_{j=1}^{r} L\left(N(x_j, \mathbf{w}_{1:m}^{(t)}) \cdot y_j\right)}{\partial w_i} + \frac{\partial \frac{1}{2}\lambda \left\|w_i^{(t)}\right\|^2}{\partial w_i}\right) = \\
&= \sum_{j=1}^{r} \Pi_{M^\perp}\left(\frac{\partial L\left(N(x_j, \mathbf{w}_{1:m}^{(t)}) \cdot y_j\right)}{\partial w_i}\right) + \Pi_{M^\perp}\left(\frac{\partial \frac{1}{2}\lambda \left\|w_i^{(t)}\right\|^2}{\partial w_i}\right) \\
&= \Pi_{M^\perp}\left(\frac{\partial \frac{1}{2}\lambda \left\|w_i^{(t)}\right\|^2}{\partial w_i}\right) \\
&= \Pi_{M^\perp}\left(\lambda w_i^{(t)}\right) \\
&= \lambda \hat{w}_i^{(t)}.
\end{aligned}$$

For a training step of size η, using gradient descent we get that:

$$\hat{w}_i^{(t+1)} = \hat{w}_i^{(t)} - \eta\lambda\hat{w}_i^{(t)}.$$

Thus, after a total of T iteration of training we get that:

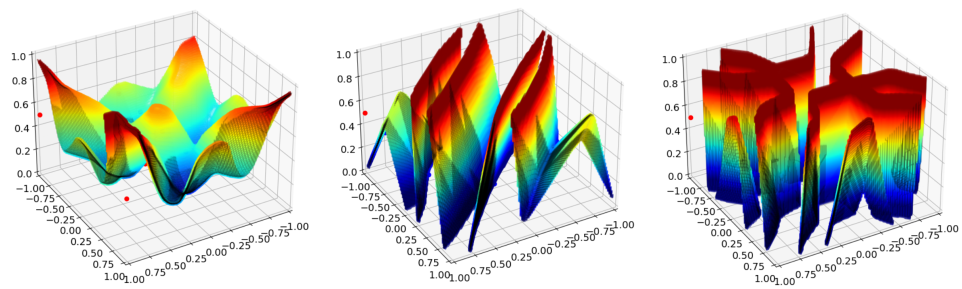
$$\hat{w}_i^{(T)} = (1 - \eta\lambda)^T \hat{w}_i^{(0)} \ .$$

Therefore, the projection of gradients after training onto $P^\perp$ will be the same as if they were initialized to $\sim \mathcal{N}\left(0, \frac{(1-\eta\lambda)^{2T}}{d} I_d\right)$ and trained using logistic loss without regularization. The rest of the proof is the same as Theorem 6.1 for $\beta = \frac{(1-\eta\lambda)^T}{\sqrt{d}}$. $\square$

E Further Experiments and Experimental Details

E.1 Further Experiments

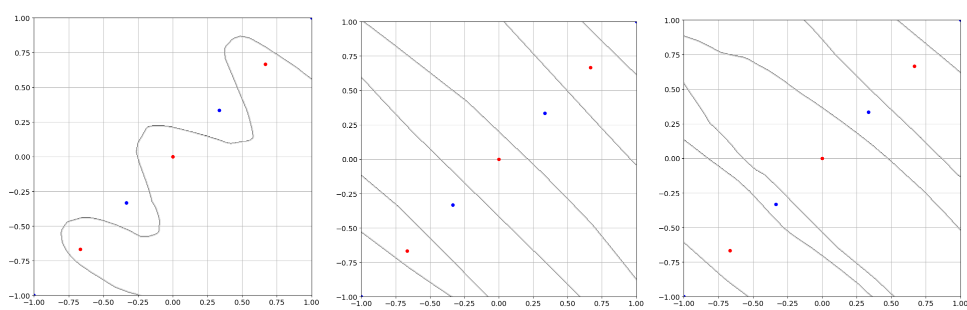
In Figure 3 we present the boundary of a two-layer ReLU network trained over a 25-point dataset on a two-dimensional linear subspace, similar to Figure 2. We train the networks until reaching a constant positive margin. The difference between the figures is that in Figure 3 we initialize the weights using the default PyTorch initialization, while in Figure 2 we initialized using a smaller scale for the robustness effect to be smaller, and visualized more easily. The experiment in Figure 3 is demonstrating an extreme robustness effect, occurring when using the standard settings.



(a) (b) (c)

Figure 3: **Experiments on two-dimensional dataset.** We plot the dataset points and the decision boundary in 3 settings: (a) Vanilla trained network, (b) The network's weights are initialized from a smaller variance distribution, and (c) Training with regularization. Colors are used to emphasise the values in the z axis.

In Figure 4 we go beyond the theory discussed in this paper, and present similar phenomena in all three settings for a five-layer ReLU network. In Figure 4a we can see the boundary of the regularly trained network within a small distance in $P^{\perp}$ from the data points. In Figure 4b we use small initialization for all five layers, and present a boundary almost orthogonal to the data manifold. In Figure 4c, the boundary of a regularized trained network is in a similar form. This experiment suggests that our theoretical results might be extended also to deeper networks, where all layers are trained.



(a) (b) (c)

Figure 4: **Experiments on one-dimensional dataset with deep network.** We plot the dataset points and the decision boundary in 3 settings: (a) Vanilla trained network, (b) The network's weights are initialized from a smaller variance distribution, and (c) Training with regularization.

E.2 One-dimensional dataset experiment - 2 layer network (Figure 1)

Dataset For all the three experiments we used a 7-point data set, spread equally on the two dimensional line $y = x$ from $(-1, -1)$ to $(1, 1)$.

Network For all the three experiments we used two-layer ReLU network of width 100 with biases in both layers. The weights of both layers were initialized using (1+3) default PyTorch initialization for linear layers, (2) default initialization divided by 3.

Training We used train step of size 0.02 for (1+3) and 0.04 for (2). We trained both layers until the margin reached 0.3. The losses we used were (1+2) Logistic loss, (3) Logistic loss with 0.005 L_2 regularization.

E.3 Two-dimensional dataset experiment - smaller effect (Figure 2)

Dataset For all the three experiments we used a 25-point data set, spread equally on a grid which lies on the $z = 0.5$ axis.

Network For all the three experiments we used two-layer ReLU network of width 4000 with biases in both layers. The weights in the first layer were initialized in (1+3) from $\mathcal{N}(\mathbf{0}, 1/3I_3)$, and in (2) from $\mathcal{N}(\mathbf{0}, 1/36I_3)$. The weight of the output layer were initialized to the uniform distribution over the set $\{-1, 1\}$.

Training For all the experiments we trained both layers until the margin reached 0.3 and we used train step of size 0.002. The losses we used were (1+2) Logistic loss, (3) Logistic loss with 0.8 L_2 regularization on the weights of the first layer.

E.4 Two-dimensional dataset experiment (Figure 3)

Dataset For all the three experiments we used a 25-point data set, spread equally on a grid which lies on the $x - y$ axis.

Network For all the three experiments we used two-layer RelU network of width 400 with biases in both layers. The weights in any layer were initialized using (1+3) default PyTorch initialization for linear layers, (2) default initialization divided by 3.

Training For (1) experiments we used train step of size 0.005, and for (2+3) we used step of size 0.05. We trained both layers until the margin reached 0.1. The losses we used were (1+2) Logistic loss, (3) Logistic loss with 0.005 L_2 regularization.

E.5 One-dimensional dataset experiment - 5 layer network (Figure 4)

Dataset For all the three experiments we used a 7-point data set, spread equally on the two dimensional line $y = x$ from $(-1, -1)$ to $(1, 1)$.

Network For all the three experiments we used 5-layer RelU network of width 100 with biases in all layers. The weights in any layer were initialized using (1+3) default PyTorch initialization for linear layers, (2) default initialization divided by 3.

Training For (1+3) experiments we used train step of size 0.02, and for (2) we used step of size 0.06. we trained all layers until the margin reached 0.3. The losses we used were (1+2) Logistic loss, (3) Logistic loss with 0.01 L_2 regularization.