Training shallow ReLU networks on noisy data using hinge loss: when do we overfit and is it benign?

Erin George
UCLA

Joint work with

Michael Murray   William Swartworth   Deanna Needell

ML Introduction

Supervised learning

A common problem in machine learning takes the following form:

We have a set $X = \{x_i\}_{i=1}^n$ of points in $\mathbb{R}^d$ and corresponding set $Y = \{y_i\}_{i=1}^n$ of labels in $\mathbb{R}^o$. We wish to learn how to identify the label y_i from the point x_i by finding some function $f : \mathbb{R}^d \to \mathbb{R}^o$ such that $y_i \approx f(x_i)$.

How do we define the approximation? We assume there is a distribution $\mathcal{D}$ on $\mathbb{R}^d \times \mathbb{R}^o$ so that each (x_i, y_i) is an i.i.d. (independent and identical distributed) sample from $\mathcal{D}$. We want to find f such that error (or *loss*) is small on average.

If y_i takes values in a discrete set, this problem is called *classification*. A simple notion of error here is counting the number of mistakes, meaning we wish to *maximize* accuracy: $\mathrm{Pr}_{(x,y)\sim\mathcal{D}}[y = f(x)]$.

Training models

We parameterize a class of "learnable" functions as f_θ. Denoting our loss function ℓ, we wish to find

$$\theta_* = \operatorname*{argmin}_\theta \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)].$$

This is difficult to find in practice, so we approximate the inner expectation with the empirical expectation:

$$\theta_* \approx \operatorname*{argmin}_\theta \frac{1}{n}\sum_{i=1}^{n}[\ell(f_\theta(x_i), y_i)].$$

This may still difficult to find, depending on exactly what f_θ and ℓ are! However, a general iterative technique works called *gradient descent*. We start with $\theta^{(0)}$ chosen arbitrarily and update

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla_\theta \frac{1}{n}\sum_{i=1}^{n}[\ell(f_\theta(x_i), y_i)]$$

for some step-size η. This converges to a local (but not global) minimum under various regularity assumptions.

Notes on loss functions

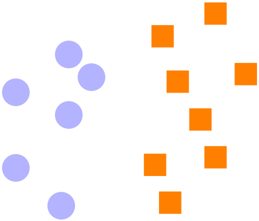
From before, we wish to find

$$\theta_* = \operatorname*{argmin}_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)].$$

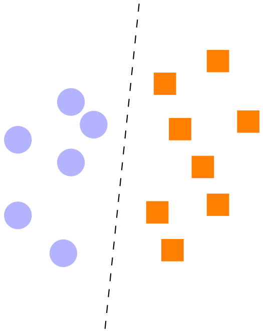
While intuitively, our loss function ℓ is some notion of accuracy, we are free to interpret $f_\theta(x)$ however we wish. In the case of classification, it's common to interpret $f_\theta(x)$ instead as a probability (or more vaguely "confidence") that a particular label is correct.

Gradient descent does not require the problem to be differentiable everywhere! However, it does need to be at least continuous and differentiable almost everywhere, so (in)accuracy does not work as a loss for gradient descent.

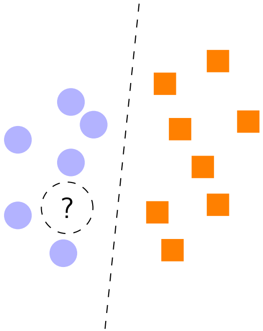
Classification example



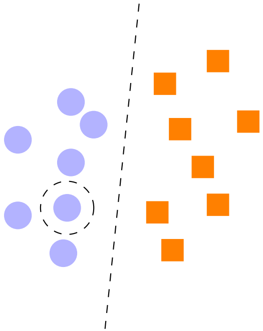
Classification example



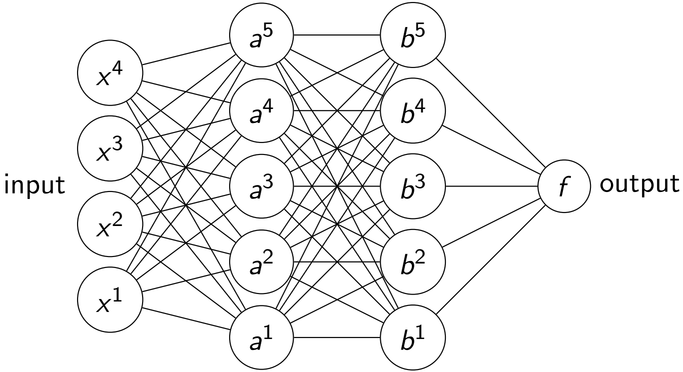
Classification example



Classification example



Neural network overview



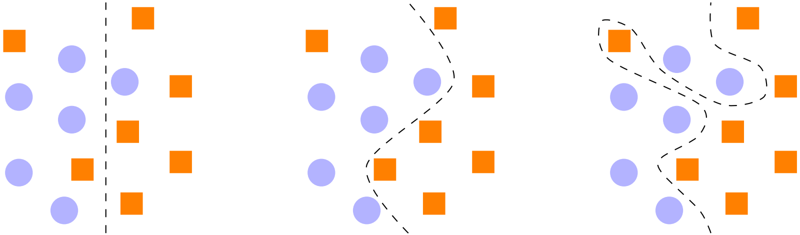
Each non-input node is a linear combination of its input nodes, with a non-linear activation function applied. Ex:

$$b^i = \sum_{j=1}^{5} \max\{0, w_{ij} a^j\}$$

The learnable parameters are the weights w_{ij}. Separate for each layer.

Generalization results in machine learning

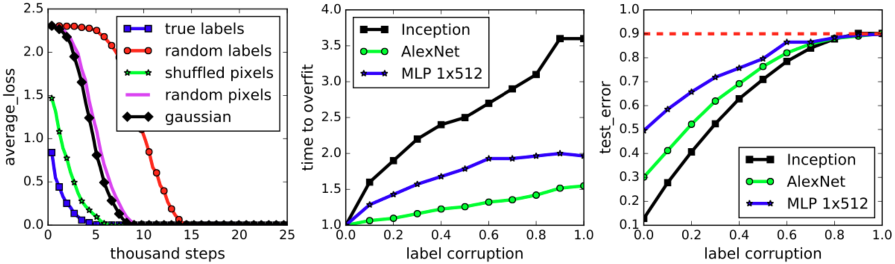
Classical bias–variance tradeoff



less complex → more complex

Modern generalization results

Deep learning models are highly complex and expressive, yet even when trained with no explicit regularization to perfectly interpolate noisy training data, they still generalize well [ZBH+17].



(a) learning curves (b) convergence slowdown (c) generalization error growth

Benign overfitting

- Informally, we say a model exhibits **benign overfitting** if it achieves zero error on noisy training data, but still performs well on test data.
- Significant progress has been made in understanding benign overfitting in linear models, but less is known about non-linear models.
- We seek to study the dynamics of a (shallow) ReLU neural network trained using GD and hinge loss on a noisy binary classification problem.

Key results

Assume inputs $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ have a signal and noise component and let $\text{👻} \in [0, 1]$ control the strength of the signal component:

$$\mathbf{x}_i \approx \sqrt{\text{👻}}\, y_i \mathbf{s}_i + \sqrt{1 - \text{👻}}\, \mathbf{n}_i .$$

We show three distinct training outcomes:

1. **Benign overfitting** (👻 *small but not too small*): zero training loss and generalization error asymptotically (in dimension d) optimal.
2. **Non-benign overfitting** (👻 *very small*): zero training loss and generalization error bounded below by a constant. (note! optimal classifier exists)
3. **No overfitting** (👻 *large*): zero training loss on "clean" points but nonzero loss on "corrupted" points, and asymptotically optimal generalization error.

Comparison with other works

A number of benign/tempered overfitting results have emerged for two layer networks trained with GD + logistic loss on noisy, linearly separable data for binary classification with near-orthogonal inputs.

- [FCB22] consider smoothed leaky ReLU activations and assume the data is drawn from a mixture of well-separated sub-Gaussian distributions.
- [XG23] extends this result to more general activation functions, including ReLU.
- [CCBG22, KCCG23] study convolutional networks where the noise and signal components lie on disjoint patches.
- [FVBS23] considers leaky ReLU and analyzes the KKT points of the max-margin problem.
- [KYS23] demonstrate benign-tempered overfitting transitions in the case of univariate inputs for ReLU networks.

Setup

Problem and data assumptions

- Training sample has $2n$ points $(\mathbf{x}_i, y_i)_{i=1}^{2n} \in (\mathbb{R}^d \times \{-1, 1\})$.
- k positive and k negative points have their output label flipped: denote $\beta(i) = -1$ if i-th point is corrupted otherwise $\beta(i) = 1$.
- Labels: $y_i = (-1)^i \beta(i)$ (clean label is $(-1)^i$)
- Inputs are of the form

$$\mathbf{x}_i = (-1)^i (\sqrt{\text{👻}}\mathbf{v} + \sqrt{1 - \text{👻}}\beta(i)\mathbf{n}_i).$$

- **Noise vectors** $(\mathbf{n}_i)_{i=1}^{2n}$ are mutually independent and identically distributed (i.i.d.) random vectors drawn from the uniform distribution over $\mathbb{S}^{d-1} \cap \text{span}\{\mathbf{v}\}^{\perp}$.
- $\text{👻} \in [0, 1]$ controls the strength of the signal versus the noise.
- Test data has same form but is assumed uncorrupted.

Loss function, model and training

- We study a densely connected, single layer feed-forward ReLU neural network with no bias terms $f : \mathbb{R}^{2m \times d} \times \mathbb{R}^d \to \mathbb{R}$,

$$f(\mathbf{W}, \mathbf{x}) = \sum_{j=1}^{2m} (-1)^j \max\{0, \langle \mathbf{w}_j, \mathbf{x} \rangle\}.$$

- Use the hinge loss $L(t) := \sum_{i=1}^{2n} \max\{0, 1 - y_i f(t, \mathbf{x}_i)\}$.
- Inner weights trained using (sub)gradient descent. Let
  - $\mathcal{F}^{(t)} := \{i \in [2n] : \ell(t, \mathbf{x}_i) < 1\}$
  - $\mathcal{A}_j^{(t)} := \{i \in [2n] : \langle \mathbf{w}_j^{(t)}, \mathbf{x}_i \rangle > 0\}$,

  then update can be written as

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} + (-1)^j \eta \sum_{l=1}^{2n} \mathbb{1}(l \in \mathcal{A}_j^{(t)} \cap \mathcal{F}^{(t)}) y_l \mathbf{x}_l.$$

Hinge versus logistic loss: recap

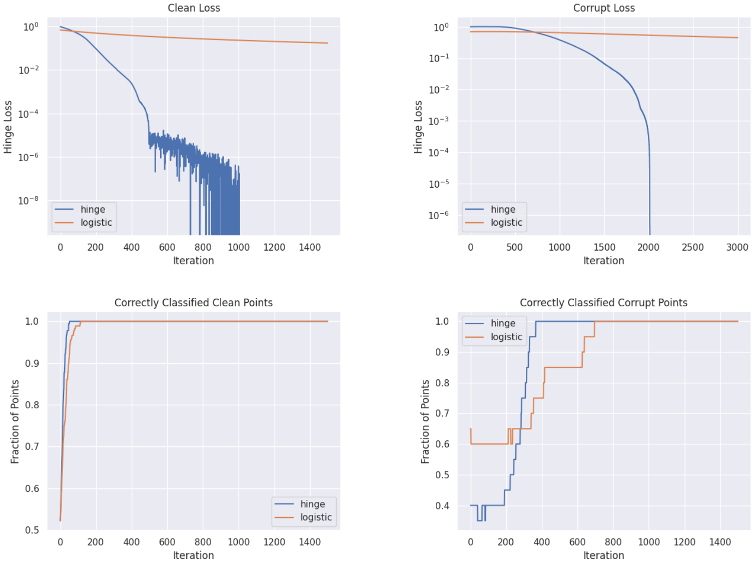
Hinge loss: $\max\{0, 1 - z\}$

- Defines a margin separating classes and penalizes points for lying within or on the incorrect side.
- Contribution of each point to overall loss driven only its network activation.
- When $y_i f(\mathbf{x}_i) \geq 1$, point no longer contributes to dynamics (switches off).

Logistic loss: $\log(1 + \exp(-z))$

- Attempts to learn log odds of point being in positive class.
- Points which are already well fitted, i.e., $y_i f(\mathbf{x}_i)$ is large, have a reduced contribution.
- A point always contributes to the dynamics of the network (never switches off).

Hinge versus logistic loss: dynamics



Hinge versus logistic: analysis

- For the logistic loss one can consider the surrogate $g(z) := -\ell'(z) = 1/(1 + \exp(z))$.
- If one can show that the evaluation of g on each sample at any given time is 'balanced' then provided the fraction of corruptions is not too large the training dynamics are primarily driven by the clean points.
- Strategy: uniformly upper bounded the ratio $\ell'(y_i f(t, \mathbf{x}_i))/\ell'(y_l f(t, \mathbf{x}_l))$ in time for all pairs of inputs.
- For the hinge loss this approach is not feasible as if iteration t some points achieve zero loss while others have not then this ratio is unbounded.
- The key idea we use to characterize the training dynamics is to reduce the analysis of the trajectory of each neuron to that of counting the number of clean versus corrupt updates to it.

Assumptions on model parameters

Let $\delta \in (0, 1/2)$ denote the failure probability, $\rho \in (0, 1)$ bound the magnitude of inner products of the noise and λ_w bound the norm of weight initializations. For sufficiently large and small constants $C \geq 1$ and $c \leq 1$ respectively,

1. $k \leq cn$,
2. $d \geq C\rho^{-2}\log(n/\delta)$
3. $\lambda_w \leq c\eta$
4. $\eta \leq \xi$, where ξ depends on n, m, k, 👻, and d.

Comments on setup

There are a number of notable limitations with the data model studied here that should be addressed.

- **Signal and noise being orthogonal:** this simplifies the analysis but is not necessary, in particular one could extend our techniques to the setting where the inner product between the signal and noise are sufficiently small.
- **Inputs have equal magnitude:** this simplifies our analysis as it means the push each input gives an activated neuron is the same, thereby reducing the problem to that of counting activations. Can be relaxed to all magnitudes within a sufficiently small range.
- **Near orthogonality of the noise:** intuitively, if noise components are nearly orthogonal then correlations between activations of different inputs is due to the signal. Much harder to relax: equivalent restrictions are present in nearly all other works in this space.

Results

Benign overfitting

Theorem 1

Assume $n \geq C \log(1/\delta)$, $m \geq C \log(n/\delta)$, $\rho \leq c \cdot \text{👻}$ *and* $C\sqrt{\log(n/\delta)/d} \leq \text{👻} \leq cn^{-1}$. *Then there exists a sufficiently small step-size* η *such that with probability at least* $1 - \delta$ *over the randomness of the dataset and network initialization the following hold.*

1. *The training process terminates at an iteration* $\mathcal{T}_{end} \leq \frac{Cn}{\eta}$.
2. *For all* $i \in [2n]$ *then* $\ell(\mathcal{T}_{end}, \mathbf{x}_i) = 0$.
3. *The generalization error satisfies*

$$\mathbb{P}(\operatorname{sgn}(f(\mathcal{T}_{end}, \mathbf{x})) \neq y) \leq \exp\left(-cd \cdot \text{👻}^2\right).$$

Main ideas behind Theorem 1

1. There are two phases of training driven by the relative imbalance in the number of clean versus corrupt points. Clean data dominates the dynamics early on but once fitted the corrupt points takeover.
2. In the first phase the network fits the clean data by learning a strong signal component, in particular by the end of this phase for most neurons $(-1)^j \langle \mathbf{w}_j, \mathbf{s} \rangle$ is large. Each corrupt point has some neurons of the correct output sign that activate on it throughout this phase.
3. In the second phase clean points start to switch off. The network fits the corrupt data by learning the noise components, however, only so many updates can occur before these points are fitted and thus the signal component the network has learned is not overly impacted.
4. At test time the noise component of a new input is approximately orthogonal to the noise components the network has learned, therefore it classified based on its signal component.

Harmful overfitting

Theorem 2

Assume $m \geq C \log(n/\delta)$, $\rho \leq cn^{-1}$, and 👻 $\leq \frac{c}{\sqrt{nd}}$. Then with probability at least $1-\delta$ over the randomness of the dataset and network initialization the following hold.

1. *The training process terminates at an iteration $\mathcal{T}_{end} \leq \frac{Cn}{\eta}$.*
2. *For all $i \in [2n]$ then $\ell(\mathcal{T}_{end}, \mathbf{x}_i) = 0$.*
3. *The generalization error satisfies*

$$\mathbb{P}(\mathrm{sgn}(f(\mathcal{T}_{end}, \mathbf{x})) \neq y) \geq \frac{1}{8}.$$

Main ideas behind Theorem 2

1. Training dynamics are dominated by the noise and points are fitted based on their individual and approximately orthogonal noise components.
2. The network fails to learn the signal strongly enough to overcome the noise.
3. The network generalizes poorly!

No overfitting

Theorem 3

Assume $m \geq 2$, $n \geq C \log\left(\frac{m}{\delta}\right)$, $\rho \leq c \cdot \text{👻}$ and $Cn^{-1} \leq \text{👻} \leq ck^{-1}$. Then there exists a sufficiently small step-size η such that with probability at least $1 - \delta$ over the randomness of the dataset and network initialization we have the following.

1. *The training process terminates at an iteration $\mathcal{T}_{end} \leq \frac{Cn}{\eta}$.*
2. *For all i clean $\ell(\mathcal{T}_{end}, \mathbf{x}_i) = 0$ while $\ell(\mathcal{T}_{end}, \mathbf{x}_i) = 1$ for all i corrupt.*
3. *The generalization error satisfies*

$$\mathbb{P}(\mathrm{sgn}(f(\mathcal{T}_{end}, \mathbf{x})) \neq y) \leq \exp\left(-cd \cdot \text{👻}^2\right).$$

Main ideas behind Theorem 3

1. Before any points are fitted training dynamics are dominated by the contributions to the signal component across the training sample.
2. In the first phase of training clean points are fitted based on their signal component while corrupt points quickly cease to activate neurons of the same sign.
3. In the second phase the activation of corrupt points on neurons of the opposite sign gradually decreases before ceasing, eventually each corrupt point is zeroed by the network. Not enough activations occur for the corrupt points to start activating neurons of the correct sign.
4. The number of updates required to zero the corrupt points is small enough that the signal component the network has learned is not overly impacted and remains strong.
5. At test time the noise component of a new input is approximately orthogonal to the noise components the network has learned, therefore it is classified based on its signal component.

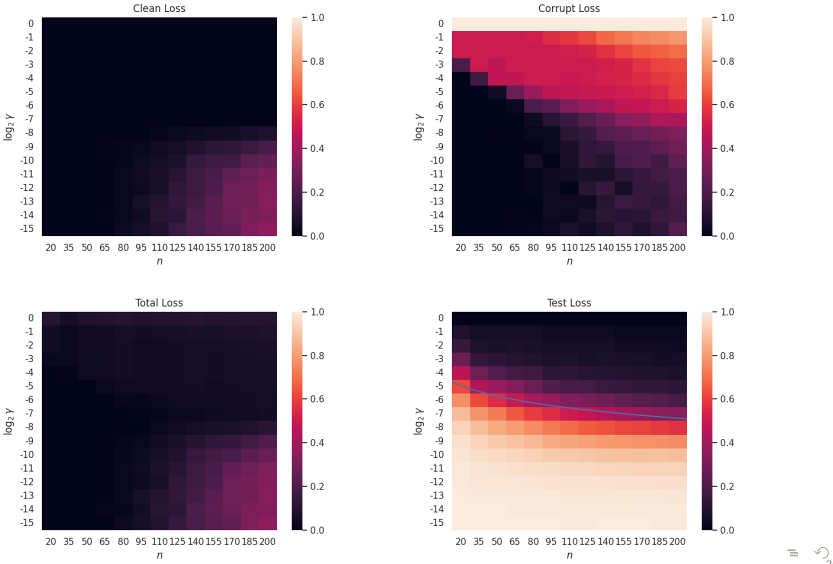
Comparison of results

Table 1: across all results $k \leq cn$ while $d \geq Cn^2 \log(n/\delta)$ for [FCB22], [XG23] and Theorem 1.

	[FCB22]	[XG23]	Theorem 1	Theorem 2	Theorem 3
$n \geq C\cdot$	$\log\left(\frac{1}{\delta}\right)$	$\log\left(\frac{m}{\delta}\right)$	$\log\left(\frac{1}{\delta}\right)$	1	$\log\left(\frac{m}{\delta}\right)$
$m \geq C\cdot$	1	$\log\left(\frac{n}{\delta}\right)$	$\log\left(\frac{n}{\delta}\right)$	$\log\left(\frac{n}{\delta}\right)$	1
👻 $\leq c\cdot$	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{\sqrt{nd}}$	$\frac{1}{k}$
👻 $\geq C\cdot$	$\frac{1}{\sqrt{nd}}$	$\sqrt{\frac{\log(\frac{md}{n\delta})}{nd}}$	$\sqrt{\frac{\log(\frac{n}{\delta})}{d}}$	0	$\frac{1}{n}$
Result	Benign[1]	Benign	Benign	Non-benign	No-overfit

Supporting experiments

In all plots, $\gamma =$ 👻.



Clean Loss

Corrupt Loss

Total Loss

Test Loss

Conclusion

Summary

- Under a simple data model we prove transitions between three different training outcomes based on 👻, which controls the clean margin of the data,
  1. benign overfitting,
  2. harmful overfitting,
  3. no overfitting.
- Unlike prior and concurrent works we study the hinge loss and prove our results in essence by bounding the number of clean versus corrupt updates that occur throughout different phases of training.

Future work and open questions

- Relax data assumptions, trainable outer layer etc.
- Classification problems with a non-linear decision boundary
- Role of depth!
- Working with structured, correlated data as opposed to near-orthogonal data.

Thanks for attending!

References I

Yuan Cao, Zixiang Chen, Misha Belkin, and Quanquan Gu.
Benign overfitting in two-layer convolutional neural networks.
Advances in neural information processing systems, 35:25237–25250, 2022.

Spencer Frei, Niladri S Chatterji, and Peter Bartlett.
Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data.
In *Conference on Learning Theory*, pages 2668–2703. PMLR, 2022.

Spencer Frei, Gal Vardi, Peter L. Bartlett, and Nathan Srebro.
Benign overfitting in linear classifiers and leaky relu networks from kkt conditions for margin maximization, 2023.

Yiwen Kou, Zixiang Chen, Yuanzhou Chen, and Quanquan Gu.
Benign overfitting for two-layer relu networks, 2023.

Guy Kornowski, Gilad Yehudai, and Ohad Shamir.
From tempered to benign overfitting in relu neural networks, 2023.

References II

Xingyu Xu and Yuantao Gu.
Benign overfitting of non-smooth neural networks beyond lazy training.
In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 11094–11117. PMLR, 25–27 Apr 2023.

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