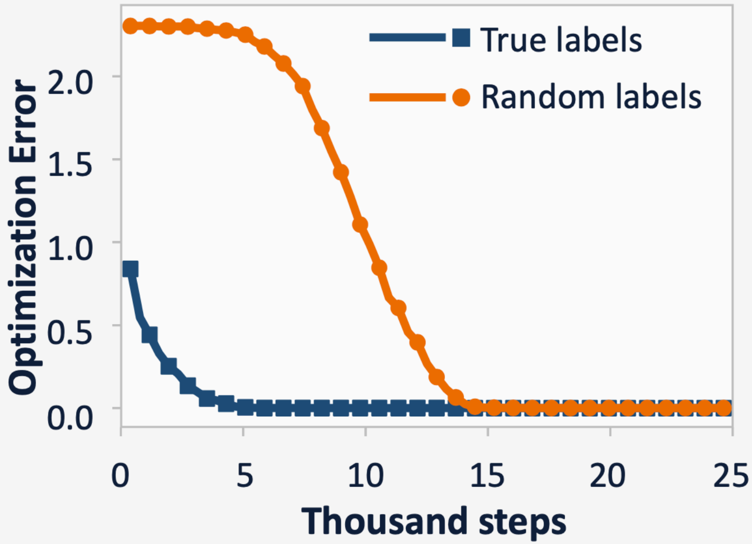
Non-convex Optimization Landscape

Gradient descent finds global minima

Practice: gradient descent

$$\theta(t+1) \leftarrow \theta(t) - \eta \frac{\partial L(\theta(t))}{\partial \theta(t)}$$



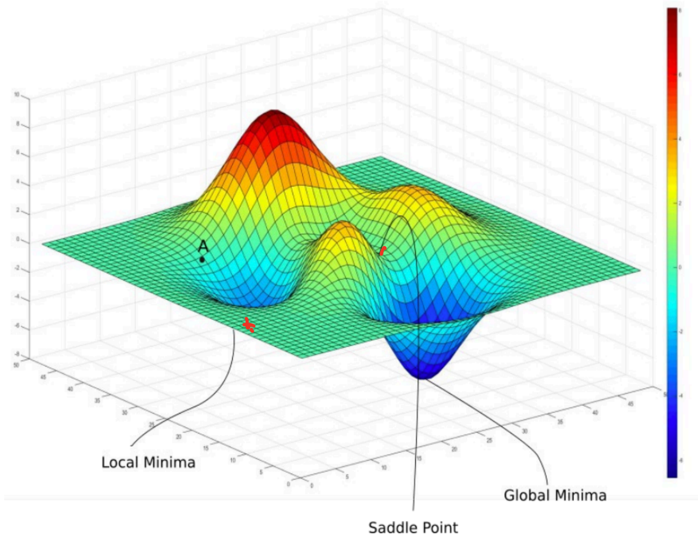
Optimization error **-> 0** for both ***true labels*** and ***random labels*** !

Zhang Bengio Hardt Recht Vinyals 2017

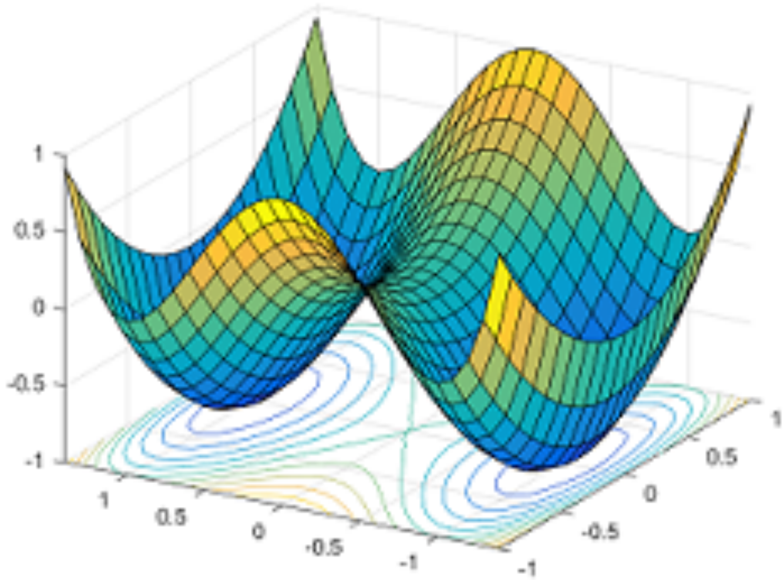
Understanding DL Requires Rethinking Generalization

Types of stationary points

- Stationary points: $x : \nabla f(x) = 0$
- Global minimum: $x : f(x) \leq f(x') \forall x' \in \mathbb{R}^d$
- Local minimum: $x : f(x) \leq f(x') \forall x' : \|x - x'\| \leq \epsilon$
- Local maximum: $x : f(x) \geq f(x') \forall x' : \|x - x'\| \leq \epsilon$
- Saddle points: stationary points that are not a local min/max

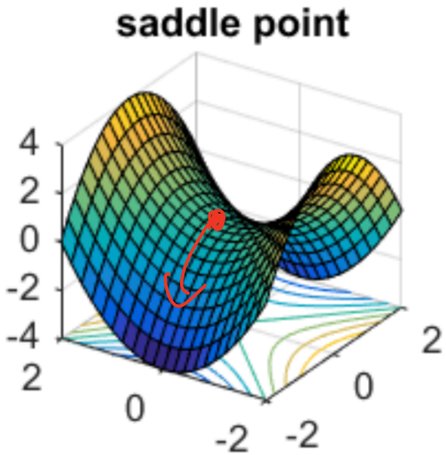


Landscape Analysis



- All local minima are global!
- Gradient descent can escape saddle points.

Strict Saddle Points (Ge et al. '15, Sun et al. '15)



- Strict saddle point: a saddle point and $\lambda_{\min}(\nabla^2 f(x)) < 0$

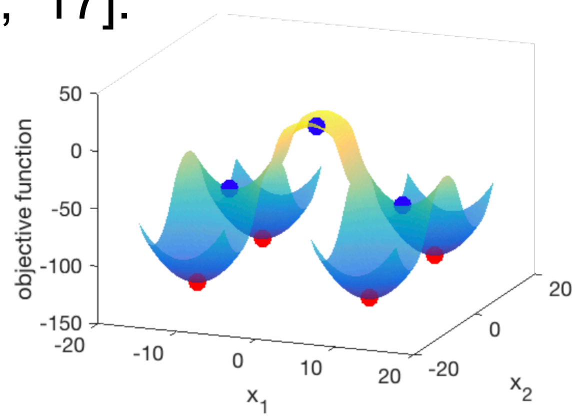
Escaping Strict Saddle Points

- Noise-injected gradient descent can escape strict saddle points in polynomial time [Ge et al., ’15, Jin et al., ’17].

- Randomly initialized gradient descent can escape all strict saddle points asymptotically [Lee et al., ’15].
  - Stable manifold theorem.

- Randomly initialized gradient descent can take exponential time to escape strict saddle points [Du et al., ’17].

If 1) all local minima are global, and 2) are saddle points are strict, then noise-injected (stochastic) gradient descent finds a global minimum in polynomial time

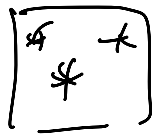


What problems satisfy these two conditions

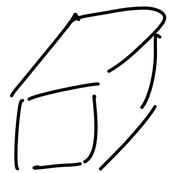
- Matrix factorization $\min_{U,V} \| UV^T - \Sigma \|_F^2$
- Matrix sensing $i=1,\dots,n,$ $\langle A_i, \Sigma \rangle = y_i,$ Σ: low-rank

$$\min_{U,V} \sum_{i=1}^{n} (\langle A_i, UV^T \rangle - y_i)^2$$

- Matrix completion



- Tensor factorization



$$T = u \otimes u \otimes u$$

- Two-layer neural network with quadratic activation

$$y_i = \sum_{j=1}^{m} (\langle x_i, w_j \rangle)^2$$

What about neural networks?

$\sigma(z) = z$

- Linear networks (neural networks with linear activations functions): all local minima are global, but there exists saddle points that are not strict [Kawaguchi '16].

$\exists$ saddle x, $\lambda_{min}(\nabla^2 f(x)) = 0$

$x = 0$

$\min_{w_1, \ldots, w_H} (w_H \cdots w_1 x - y)^2$

- Non-linear neural networks with:
  - Virtually any non-linearity,
  - Even with Gaussian inputs,
  - Labels are generated by a neural network of the same architecture,

ReLU, Sigmoid

$x \sim N(0, I)$

$y = NN(x)$

There are many bad local minima [Safran-Shamir '18, Yun-Sra-Jadbaie '19].

$\exists$ global minima

Global convergence of gradient descent

Global convergence of gradient descent

Theorem (Du et al. '18, Allen-Zhu et al. '18, Zou et al 19') If the width of each layer is poly(n) where n is the number of data. Using random initialization with a particular scaling, gradient descent finds an approximate global minimum in polynomial time.

Neural Tangent Kernel

$$|f(\underset{\sim}{w}) - f(w^{\#})| \leq \underset{\sim}{\varepsilon}$$

$$\text{poly}\left(\frac{1}{\varepsilon}, \text{width}, \text{depth}\right)$$

Gradient Flow: a Kernel Point of View

- $L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\theta, x_i), y_i)$

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{1}{n} \sum_{i=1}^{n} \ell'(f(\theta, x_i), y_i) \cdot \frac{\partial f(\theta, x_i)}{\partial \theta}$$

GF: $\frac{d\theta(t)}{dt} = -\frac{\partial L(\theta)}{\partial \theta}$

if strong convex, $\exists\, \theta^*$, $\theta(t) \to \theta^*$

in NN, # of parameters $> n$

we know $t \to \infty$, $f(\theta(t), x_i) \to y_i$

Gradient Flow: a Kernel Point of View

$$u_i(t) = f(\theta(t), x_i), \quad u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{pmatrix}$$

$$\frac{d u_i(t)}{dt} = \left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, \frac{d\theta(t)}{dt} \right\rangle$$

$$= \left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, -\frac{1}{n}\sum_{j=1}^{n} \ell'(u_j(t), y_j) \cdot \frac{\partial u_j(t)}{\partial \theta(t)} \right\rangle$$

$\ell'(u(t), y) \in \mathbb{R}^n$

$[\ell'(u(t), y)]_j = \ell'(u_j(t), y_j)$

$$= -\frac{1}{n}\left[\ell'(u_1(t), y_1), \ldots, \ell'(u_n(t), y_n)\right] \left(\left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, \frac{\partial u_1(t)}{\partial \theta(t)} \right\rangle, \ldots, \left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, \frac{\partial u_n(t)}{\partial \theta(t)} \right\rangle \right)$$

$H(t) \in \mathbb{R}^{n \times n}$

$$[H(t)]_{ij} = \left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, \frac{\partial u_j(t)}{\partial \theta(t)} \right\rangle$$

$$\frac{du(t)}{dt} = -\frac{1}{n} H(t) \cdot \ell'(u(t), y),$$

Gradient Flow: a Kernel Point of View

If ℓ = quadratic loss, $\ell(u(t), y) = \frac{1}{2}(u(t)-y)^2$

$$\ell'(u(t), y) = u(t) - y$$

$$(*) \quad \frac{du(t)}{dt} = -\frac{1}{n} \underbrace{H(t)}_{\text{does not depend on loss}} (u(t) - y)$$

- If $\exists \lambda_0 > 0$, $\forall t$, $\lambda_{min}(H(t)) \geq \lambda_0$

$$\frac{d\left(\underbrace{\frac{1}{2}\|u(t)-y\|_2^2}_{\text{loss}}\right)}{dt} = -\frac{1}{n}(u(t)-y)^T H(t)(u(t)-y)$$

$$\leq -\frac{\lambda_0}{n}\|u(t)-y\|_2^2$$

Gradient Flow: a Kernel Point of View

Consider $\frac{d}{dt}\left(\exp\left(\frac{\lambda_0 t}{n}\right)\cdot\frac{1}{2}\|u(t)-y\|_2^2\right)$

$$= \frac{\lambda_0}{2n}\exp\left(\frac{\lambda_0 t}{n}\right)\|u(t)-y\|_2^2 + \frac{d\left(\frac{1}{2}\|u(t)-y\|_2^2\right)}{dt}\cdot\exp\left(\frac{\lambda_0 t}{n}\right)$$

$$\leq \exp\left(\frac{\lambda_0 t}{n}\right)\|u(t)-y\|_2^2\left(\frac{\lambda_0 t}{2n}-\frac{\lambda_0 t}{n}\right) < 0$$

$\Rightarrow$ $\exp\left(\frac{\lambda_0 t}{n}\right)\cdot\frac{1}{2}\|u(t)-y\|_2^2$ is decreasing

$t=0$, $\underbrace{\frac{1}{2}\|u(0)-y\|_2^2}$ $O(C)$

$\forall t$ $\exp\left(\frac{\lambda_0 t}{n}\right)\cdot\frac{1}{2}\|u(t)-y\|_2^2 \leq C$

$\frac{1}{2}\|u(t)-y\|_2 \leq C\cdot\exp\left(-\frac{\lambda_0 t}{n}\right)$

$\log\left(\frac{1}{\varepsilon}\right)$ rate

$t\to\infty$, loss $\to 0$, $u(t)\to y$

Gradient Flow: a Kernel Point of View

kernel: $[u(t)]_i = \phi(x_i)^T \theta(t)$ ϕ: feature

$$[H(t)]_{ij} = \left\langle \frac{\partial u_i(t)}{\partial \theta(t)}, \frac{\partial u_j(t)}{\partial \theta(t)} \right\rangle$$

$$= \langle \phi(x_i), \phi(x_j) \rangle$$

$$= k(x_i, x_j)$$ does not depend on t

$$H(t) = \begin{pmatrix} & & \\ & k(x_i, x_j) & \\ & & \end{pmatrix}$$ if kernel is full rank

$\lambda_{min}(H(t)) > 0$ if kernel is universal (Gaussian) $\lambda_{min}(H(t)) > 0$

Gradient Flow: a Kernel Point of View

- $f(\theta, x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \, \sigma(w_r^T x)$
- $x \in \mathbb{R}^d$, $w_r \in \mathbb{R}^d$, $a_r \in \mathbb{R}$, $\sigma(\cdot)$: ReLU
- Initialization: $a_r \sim \text{unif}\{1, -1\}$ only for simplicity
  $w_r \sim N(0, I)$
- Training: only train $w_1, \ldots, w_m$

$$\min_{w_1, \ldots, w_m} \frac{1}{n} \sum_{i=1}^{n} (f(x_i; a, w) - y_i)^2$$

$$\frac{d\,u(t)}{dt} = -\frac{1}{n} H(t) \cdot (u(t) - y)$$

Idea: show $H(t) \approx \underbrace{H^*}_{NTK}$, $H^*_{ij} = \lim_{m \to \infty} \mathbb{E}_{\text{init dist}} \left\langle \frac{\partial f(x_i; a, w)}{\partial w}, \frac{\partial f(x_j; \cdot)}{\partial w} \right\rangle$

Gradient Flow: a Kernel Point of View

$$W \in \mathbb{R}^{m \times d}$$

$$H_{ij}(t) = \left\langle \frac{\partial u_i(t)}{\partial W(t)}, \frac{\partial u_j(t)}{\partial W(t)} \right\rangle,$$

$$= \sum_{r=1}^{m} \left\langle \frac{\partial u_i(t)}{\partial w_r(t)}, \frac{\partial u_j(t)}{\partial w_r(t)} \right\rangle$$

$$\frac{\partial u_i(t)}{\partial w_r(t)} = \frac{1}{\sqrt{m}} a_r \cdot x_i \cdot \mathbb{1}\{w_r^T x_i \geq 0\}$$

$$H_{ij}(t) = \sum_{r=1}^{m} \frac{1}{m} \langle a_r \cdot x_i \cdot \mathbb{1}\{w_r^T x_i \geq 0\}, a_r \cdot x_j \mathbb{1}\{w_r^T x_j \geq 0\} \rangle$$

$$= \frac{1}{m} x_i^T x_j \sum_{r=1}^{m} \mathbb{1}\{w_r^T x_i \geq 0, w_r^T x_j \geq 0\}$$

To show $H(t) \approx H^*$: ① $H(0) \approx H^*$

② $H(t) \approx H(0)$

Gradient Flow: a Kernel Point of View

Initialization:

Hoeffding Inequality:

R.V. $z_1, \ldots, z_n \overset{i.i.d.}{\sim}$, $|z_i| \leq 1$

if $n = \Omega\left(\frac{\log(\frac{1}{\delta})}{\varepsilon^2}\right)$, $0 < \delta < 1$, $0 < \varepsilon < 1$

w.p. $1-\delta$, $\left|\frac{1}{n}\sum_{i=1}^{n} z_i - \mathbb{E}[z_i]\right| \leq \varepsilon$

$$H_{ij}(0) = x_i^T x_j \cdot \underbrace{\frac{1}{m}\sum_{r=1}^{m}}_{\text{average}} \cdot \underbrace{\mathbb{1}\{w_r(0)^T x_i \geq 0, w_r(0)^T x_j \geq 0\}}_{\text{random variable}}$$

$$H^*_{ij} = \mathbb{E}_{w \sim N(0, I)}\, x_i^T x_j \cdot \mathbb{1}\{w^T x_i \geq 0, w^T x_j \geq 0\}$$

when m is large enough, $|H_{ij}(0) - H^*_{ij}| \leq \varepsilon$

Gradient Flow: a Kernel Point of View

$$\|H^* - H(0)\|_F$$

$$\le \sum_{i,j} |H^*_{ij} - H_{ij}(0)|$$

$$\le n^2 \epsilon \qquad \text{choose } \epsilon \text{ small}$$

First : power of over-parameterization

$\rightarrow$ concentration

Gradient Flow: a Kernel Point of View

ε - training error

Want $H(t) \approx H(0)$

for simplicity: 1) just train till time t

2) $y_i = O(1)$

3) $\|x_i\|_2 = 1$

$$H_{ij}^{*} = x_i^T x_j \cdot \frac{\pi - \arccos(x_i^T x_j)}{2\pi}$$

Key idea: every weight vector only moves a little ($\frac{1}{\sqrt{m}}$), lazy training

$$\| w_r(t) - w_r(0) \|_2 = \left\| \int_0^t \frac{d w_r(\tau)}{d\tau} d\tau \right\|_2$$

$$\leq \int_0^t \left\| \frac{d w_r(\tau)}{d\tau} \right\|_2 d\tau$$

$$= \int_0^t \left\| \frac{1}{\sqrt{m}} \frac{1}{n} \sum_{i=1}^{n} (u_i(\tau) - y_i) a_r x_i \cdot \mathbb{1}\{ w_r^T x_i \geq 0 \} \right\|_2 d\tau$$

If $u_i(\tau) = O(1)$

$$\leq C \cdot \int_0^t \frac{1}{\sqrt{m}} 1 \cdot d\tau$$

$$\leq \frac{C \cdot t}{\sqrt{m}}$$