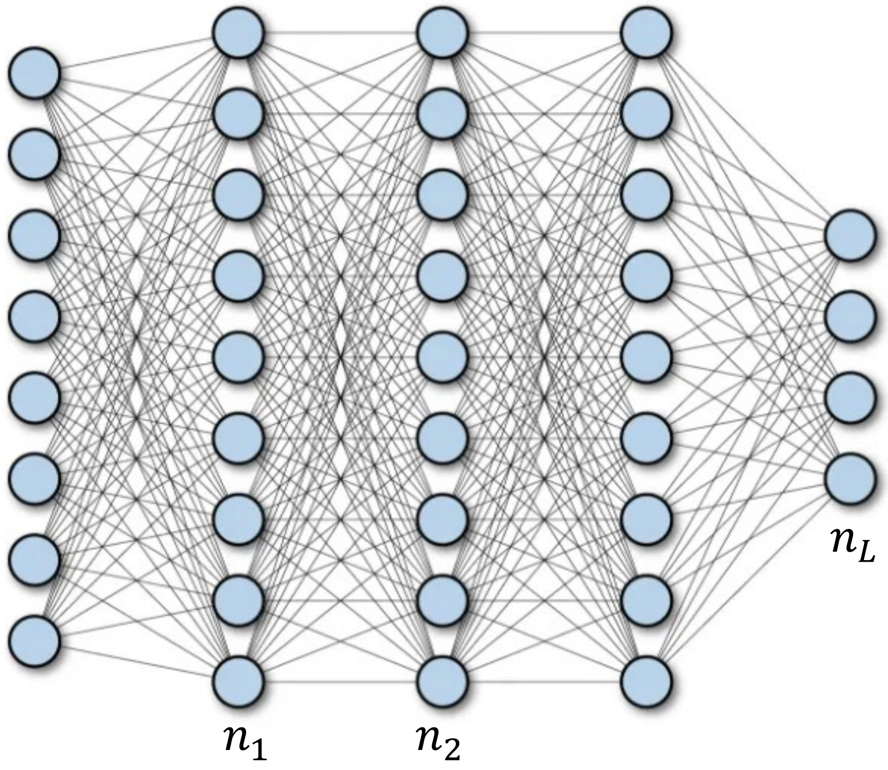
Neural Tangent Kernel:
Convergence and Generalization in Neural Networks

Abstract

- At initialization, ANNs (artificial neural networks) = Gaussian processes in the infinite-width limit
- But, the evolution of an ANN during training = described by a kernel



- Network function f_θ follows the kernel gradient of the functional cost w.r.t. a new kernel -> (NTK)
  - Describes generalization features of ANNs
  - Random at initialization
  - During training varies
    - In infinite width limit -> converges to explicit limiting kernel
- Study ANNs in function space instead of parameter space
- Convergence -> fastest along largest kernel principal components
  - Of the input data respect to the NTK

Introduction

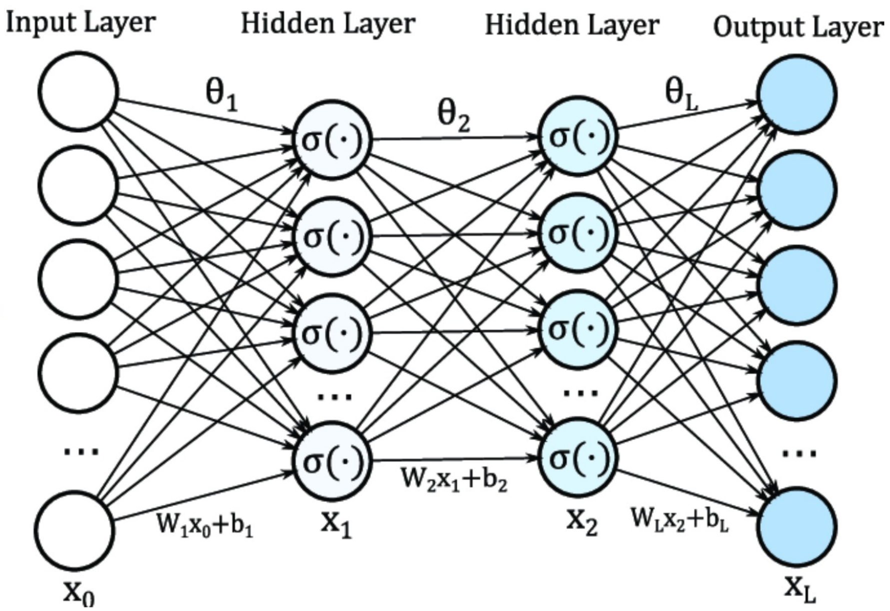
- ANNs are powerful!
  - Can approximate any function with sufficiently many hidden layers
- But, what the optimization of ANNs converges to?
- Although in wide enough networks:
  - Very few bad local minima

Mysterious Features of ANNs

- Good generalization properties in spite of usual overparameterization
- Can fit random labels
- Still obtaining good test accuracy on real data
- Same as kernel methods

Neural Network's Setting

- Fully-connected ANN
- Layers containing $n_0, \ldots, n_L$ neurons
- $\sigma: \mathbb{R} \to \mathbb{R}$:
  - Lipschitz, twice differentiable, nonlinearity function, bounded second derivatives



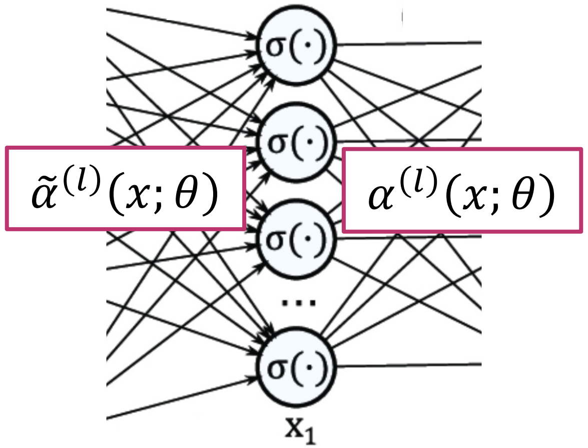
Notations

- Realization function $F^{(L)}: \mathbb{R}^P \to \mathcal{F}$
  - $P = \sum_{l=0}^{L-1}(n_l + 1)n_{l+1}$
  - $\mathcal{F} = \{f(.;\theta) | \, \theta \in \mathbb{R}^P\}$
- Parameters are initialized as iid Gaussian $\mathcal{N}(0,1)$
- p^{in}: a fixed distribution on the input space
  - The empirical distribution on a finite dataset
  - Semi norm:
    - $< f, g >_{p^{in}} = \mathbb{E}_{x \sim p^{in}}[f(x)^T g(x)]$

Gradient Flow

- minimize $F(\theta)$ over parameter θ
- By GD:
  - $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta F(\theta)|_{\theta=\theta^{(t)}}$
- Differential Equation:
  - $\frac{d\theta^{(t)}}{dt} = -\nabla_\theta F(\theta)|_{\theta=\theta^{(t)}}$

Network Function



- $f_\theta(x) = \tilde{\alpha}^{(L)}(x;\theta)$

- $\alpha^{(0)}(x;\theta) = x$
- $\tilde{\alpha}^{(l+1)}(x;\theta) = \frac{1}{\sqrt{n_l}} W^{(l)} \alpha^{(l)}(x;\theta) + \beta b^{(l)}$ (like Xavier initialization)
  - Infinite width limit -> consistent asymptotic behavior
- $\alpha^{(l)}(x;\theta) = \sigma(\tilde{\alpha}^{(l)}(x;\theta))$

Nonconvexity problem

- $\mathcal{F} = \{f(.;\theta) | \theta \in \mathbb{R}^P\}$
- Cost function: $C: \mathcal{F} \to \mathbb{R}$
- But, $C \circ \mathcal{F}: \mathbb{R}^P \to \mathbb{R}$ is in general highly nonconvex

Using Gradient Flow

- L is fixed & $n_0, \dots, n_{L-1} \to \infty$
- $\theta_1, \dots, \theta_P \sim \mathcal{N}(0,1)$
- $\frac{\partial \theta^{(t)}}{\partial t} = -\nabla_\theta C(f(.;\theta))|_{\theta=\theta^{(t)}}$
- If C is a least square function and $\Delta_i^{(t)} = f(x_i;\theta^{(t)}) - y_i$
  - $\frac{\partial \theta^{(t)}}{\partial t} = -\sum \Delta_i^{(t)} \nabla_\theta (f(x_i;\theta))|_{\theta=\theta^{(t)}}$

Way to NTK

- $\frac{\partial\theta^{(t)}}{\partial t} = -\sum \Delta_i^{(t)} \nabla_\theta(f(x_i;\theta))|_{\theta=\theta^{(t)}}$
- $\frac{\partial\Delta_i^{(t)}}{\partial t} = \nabla_\theta(f(x_i;\theta))^T|_{\theta=\theta^{(t)}} \frac{\partial\theta^{(t)}}{\partial t}$
  - $\Delta_i^{(t)} = f(x_i;\theta^{(t)}) - y_i$
- Thus, $\frac{\partial\Delta_i^{(t)}}{\partial t} = -\sum \Delta_j^{(t)} \nabla_\theta(f(x_i;\theta))^T|_{\theta=\theta^{(t)}} \nabla_\theta(f(x_j;\theta))|_{\theta=\theta^{(t)}}$
- So, we can say that:
  - $\frac{\partial}{\partial t}\vec{\Delta}^{(t)} = -K^{(t)}\vec{\Delta}^{(t)}$
  - Where $K_{ij}^{(t)} = < \nabla_\theta\left(f(x_i;\theta)\right)|_{\theta=\theta^{(t)}} \middle| \nabla_\theta\left(f\left(x_j;\theta\right)\right)|_{\theta=\theta^{(t)}} >$

NTK

- NTK: $\Theta(x, x'|\theta) =< \nabla_\theta(f(x;\theta)) \mid \nabla_\theta(f(x';\theta)) >$
- Remember: $\frac{\partial}{\partial t}\vec{\Delta}^{(t)} = -K^{(t)}\vec{\Delta}^{(t)}$
- If $K^{(t)}$ is constant w.r.t. t -> $\vec{\Delta}^{(t)} = e^{-tK}\vec{\Delta}^{(0)}$
  - Where $e^{-tK} = \sum_{i=0}^{\infty}\frac{(-tK)^i}{i!}$
  - If λ is eigenvalue of K -> $e^{-t\lambda}$ is eigenvalue of e^{-tK}

Convergence of NTK

- L and T are fixed
- Would like to show that $K^{(t)}$ converges to a constant in $[0, T]$ in infinite width limit
  - Uniform & in probability

Gaussian Process in ANN at initialization!

- $\theta_1, \ldots, \theta_P \sim \mathcal{N}(0,1)$
- For any x, $f(x;\theta)$ is random
- $f(.;\theta)$ is a centered Gaussian Process in initialization
  - First layer ✅
  - Other layers: by induction ✅

$$\boxed{\tilde{\alpha}^{(l+1)}(x;\theta) = \frac{1}{\sqrt{n_l}} W^{(l)} \alpha^{(l)}(x;\theta) + \beta b^{(l)}}$$

$$\Sigma^{(1)}(x,x') = \frac{1}{n_0} x^T x' + \beta^2$$

$$\tilde{\Sigma}^{(L+1)}(x,x') = \frac{1}{n_L} \alpha^{(L)}(x;\theta)^T \alpha^{(L)}(x';\theta) + \beta^2.$$

$$\Sigma^{(L+1)}(x,x') = \mathbb{E}_{f \sim \mathcal{N}(0,\Sigma^{(L)})}[\sigma(f(x))\sigma(f(x'))] + \beta^2$$

Limiting Kernel!

- Review: $\Theta_t^{(L)}(x, x') =< \nabla_\theta\big(f\big(x; \theta^{(t)}\big)\big) \mid \nabla_\theta\big(f\big(x'; \theta^{(t)}\big)\big) >$
- Define it by induction:

$$\Theta_\infty^{(1)}(x, x') = \Sigma^{(1)}(x, x')$$

$$\Theta_\infty^{(L+1)}(x, x') = \Theta_\infty^{(L)}(x, x')\dot{\Sigma}^{(L+1)}(x, x') + \Sigma^{(L+1)}(x, x')$$

$$\dot{\Sigma}^{(L+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}\left(0, \Sigma^{(L)}\right)}\left[\dot{\sigma}\left(f(x)\right)\dot{\sigma}\left(f(x')\right)\right]$$

- The limiting $\Theta_\infty^{(L)}$ only depends on:
  - The choice of σ
  - Depth of network
  - Variance of parameters at initialization

Convergence of NTK during training

- L and T are fixed
- Uniformly for $[0, T]$, in probability we have:
  - $\lim_{n_1,\ldots,n_{L-1}} \Theta_t^{(L)}(x, x') \rightarrow \Theta_\infty^{(L)}(x, x')$
- The variation during training of the individual activations in the hidden layers shrinks as their width grows

Convergence and Early Stopping

- Remember: $\frac{\partial}{\partial t}\vec{\Delta}^{(t)} = -\Theta_{\infty}^{(L)}\vec{\Delta}^{(t)}$
- Thus, $\vec{\Delta}^{(t)} = e^{-t\Theta_{\infty}^{(L)}}\vec{\Delta}^{(0)}$
- The convergence is indeed faster along the eigenspaces corresponding to larger eigenvalues

- Early stopping:
  - convergence on the most relevant kernel principal components
  - avoiding to fit the ones in eigenspaces with lower eigenvalues

Kernel Gradient General Case

- Multi dimensional kernel $K: \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$
  - $K(x, x') = K(x', x)^T$
- $\langle f, g \rangle_K := \mathbb{E}_{x,x' \sim p^{in}} \left[f(x)^T K(x, x') g(x') \right]$

- Dual $\mathcal{F}$: $\mathcal{F}^*$ set of linear forms $\mu: \mathcal{F} \to \mathbb{R}$ -> $\mu = < d, . >_{p^{in}}$ for $d \in \mathcal{F}$

Introducing of Kernel Gradient

- Mapping $\phi_K : \mathcal{F}^* \to \mathcal{F}$
  - $\mu = < d, . >_{p^{in}}$
  - $f_\mu = \phi_K(\mu)$
    - $f_{\mu,i}(x) = < d, K_{i,.}(x, .) >_{p^{in}}$
- The (functional) derivative of the cost C -> as an element of $\mathcal{F}^*$
  - At point f_0: $\partial_f^{in} C|_{f_0}$
  - Corresponding dual element: $\mathrm{d}|_{f_0}$ -> $\partial_f^{in} C|_{f_0} = < \mathrm{d}|_{f_0}, . >_{p^{in}}$
- Kernel Gradient:
  - $\nabla_K C|_{f_0} = \phi_K(\partial_f^{in} C|_{f_0})$

Kernel Gradient Descent

- $\partial_f^{in} C|_{f_0}$ only defined on the dataset
  - C only depends on the values of f at the data points
- Kernel gradient generalizes to all values
  - $\nabla_K C|_{f_0}(x) = \frac{1}{N}\sum_{j=1}^{N} K(x, x_j)\mathrm{d}|_{f_0}(x_j)$
- $f(t)$ follows kernel gradient descent w.r.t. K if
  - $\partial_t f(t) = -\nabla_K C|_{f(t)}$

Approximating the kernel

- Kernel K can be approximated by a choice of P random functions:
  - $\mathbb{E}[f_k^{(p)}(x) f_{k'}^{(p)}(x')] = K_{kk'}(x, x')$
- Random linear parametrization $F^{lin} : \mathbb{R}^P \to \mathcal{F}$

$$\theta \mapsto f_\theta^{lin} = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \theta_p f^{(p)}$$

- Partial derivatives

$$\partial_{\theta_p} F^{lin}(\theta) = \frac{1}{\sqrt{P}} f^{(p)}$$

Catching the Approximation of the Kernel

- Gradient descent on $C \circ F^{lin}$
  - $\partial_t \theta_p(t) = -\partial_{\theta_p}(C \circ F^{lin})(\theta(t)) = -\frac{1}{\sqrt{P}} \partial_f^{in} C|_{f_{\theta(t)}^{lin}} f^{(p)} = -\frac{1}{\sqrt{P}} \left\langle d|_{f_{\theta(t)}^{lin}}, f^{(p)} \right\rangle_{p^{in}}$
- As a result,
  - $\partial_t f_{\theta(t)}^{lin} = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \partial_t \theta_p(t) f^{(p)} = -\frac{1}{P} \sum_{p=1}^{P} \left\langle d|_{f_{\theta(t)}^{lin}}, f^{(p)} \right\rangle_{p^{in}} f^{(p)}$
- The R.H.S is the kernel gradient w.r.t.
  - $\tilde{K} = \sum_{p=1}^{P} \partial_{\theta_p} F^{lin}(\theta) \otimes \partial_{\theta_p} F^{lin}(\theta) = \frac{1}{P} \sum_{p=1}^{P} f^{(p)} \otimes f^{(p)}$

Neural Tangent Kernel

- During training: $\partial_t f_{\theta(t)} = -\nabla_{\Theta^{(L)}} C|_{f_{\theta(t)}}$
- Neural Tangent Kernel:

$$\Theta^{(L)}(\theta) = \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta)$$

- Derivative $\partial_t F^{(L)}(\theta)$ and the NTK depend on the parameters
- The NTK is therefore random at initialization and varies during training

Gaussian Process in ANN at initialization!

- $\theta_1, \dots, \theta_P \sim \mathcal{N}(0,1)$
- For any x, $f(x;\theta)$ is random
- $f(.;\theta)$ is a centered Gaussian Process in initialization
  - First layer ✅
  - Other layers: by induction ✅

$$\Sigma^{(1)}(x,x') = \frac{1}{n_0} x^T x' + \beta^2$$

$$\tilde{\Sigma}^{(L+1)}(x,x') = \frac{1}{n_L} \alpha^{(L)}(x;\theta)^T \alpha^{(L)}(x';\theta) + \beta^2.$$

$$\Sigma^{(L+1)}(x,x') = \mathbb{E}_{f \sim \mathcal{N}(0,\Sigma^{(L)})}[\sigma(f(x))\sigma(f(x'))] + \beta^2$$

Limiting Kernel!

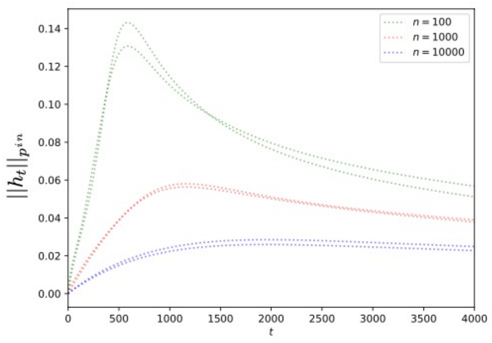
- Define it by induction:

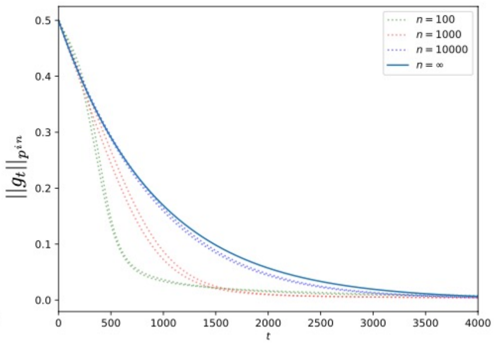
$$\Theta_{\infty}^{(1)}(x,x') = \Sigma^{(1)}(x,x')$$
$$\Theta_{\infty}^{(L+1)}(x,x') = \Theta_{\infty}^{(L)}(x,x')\dot{\Sigma}^{(L+1)}(x,x') + \Sigma^{(L+1)}(x,x')$$
$$\dot{\Sigma}^{(L+1)}(x,x') = \mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma^{(L)}\right)}\left[\dot{\sigma}\left(f\left(x\right)\right)\dot{\sigma}\left(f\left(x'\right)\right)\right]$$

- The limiting $\Theta_{\infty}^{(L)}$ only depends on:
  - The choice of σ
  - Depth of network
  - Variance of parameters at initialization

Convergence of NTK in practice

- A surprising observation is that smaller networks appear to converge faster than wider ones.
- The NTK of large-width network is more stable during training, larger learning rates can in principle be taken.





(b) Deviation of the network function f_θ from the straight line.

(c) Convergence of f_θ along the 2nd principal component.

Conclusion

- NTK provides a powerful framework to understand the behavior of ANNs during training, linking them to kernel methods.
- At initialization, wide ANNs behave like Gaussian Process.
- At initialization and during training, their training dynamics are governed by a fixed kernel (NTK) in the infinite-width limit.
- One can relate convergence of ANN training with early stopping methods.

Thanks for your attention!