Neural Tangent Kernel: Convergence and Generalization in Neural Networks

Arthur Jacot-Guillarmod* Franck Gabriel* Clément Hongler* *EPFL

The training of Artificial Neural Networks (ANNs) involves a priori highly non-convex optimization, yet in practice one can observe impressive convergence and generalization properties.

Question How to mathematically understand the dynamics of the training of an ANN as it becomes large?

Thanks to a new object, the Neural Tangent Kernel, we can precisely describe the evolution, convergence, and generalization of ANNs of large width

Theorem In the infinite-width limit, the ANN function follows a kernel gradient descent with respect to the limiting NTK

1. ANN Training: Architecture and Optimization

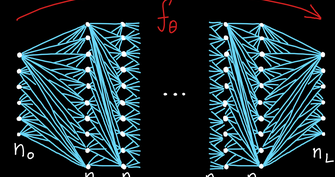
- Fully connected neural net with $L+1$ layers of widths $n_0,\dots,n_L$ and nonlinearity $\sigma:\mathbb{R}\to\mathbb{R}$

$$f_\theta:\mathbb{R}^{n_0}\to\mathbb{R}^{n_1}\to\mathbb{R}^{n_2}\to\cdots\to\mathbb{R}^{n_{L-1}}\to\mathbb{R}^{n_L}$$
$$x\mapsto\tilde\alpha^{(0)}(x)\mapsto\alpha^{(0)}(x)\mapsto\tilde\alpha^{(1)}(x)\overset{\sigma}{\mapsto}\cdots\overset{\sigma}{\mapsto}\alpha^{(L-1)}(x)\mapsto\tilde\alpha^{(L)}(x)=f_\theta(x)$$
$$\tilde\alpha^{(0)}=\tfrac{1}{\sqrt{n_0}}W^{(0)}\alpha^{(0)}+\beta b^{(0)}\qquad\alpha^{(1)}=\sigma(\tilde\alpha^{(1)})\qquad\tilde\alpha^{(1)}=\tfrac{1}{\sqrt{n_1}}W^{(1)}\alpha^{(1)}+\beta b^{(1)}$$
$$\sigma(\tilde\alpha^{(L-1)})\qquad\tilde\alpha^{(L)}=\tfrac{1}{\sqrt{n_{L-1}}}W^{(L-1)}\alpha^{(L-1)}+\beta b^{(L-1)}$$

Network params $\in\mathbb{R}^{n_0\times n_1}$, $\mathbb{R}^{n_1}$; pointwise application

$\theta=(W^{(0)},b^{(0)},\dots,W^{(L-1)},b^{(L-1)})\in\mathbb{R}^P$

- Gaussian Initialization: $\theta_p\overset{iid}{\sim}\mathcal{N}(0,1)$ $\quad\epsilon\in\{1,\dots,P\}$
- Training Set: $(x_i,z_i)_{i=1}^N\in\mathbb{R}^{n_0}\times\mathbb{R}^{n_L}$ (Training Inputs, Training Outputs) $\quad\epsilon\in\{1,\dots,N\}$
- Training Loss: $C(\theta)=\frac{1}{N}\sum_{i=1}^N c(f_\theta(x_i),z_i)$

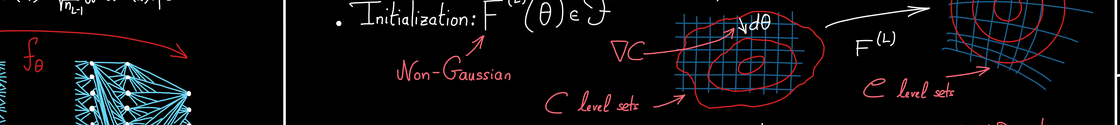


Goal: Minimize $C(\theta)$ with Gradient Descent

- Gradient Descent Step: $\theta\mapsto\theta-\eta\nabla C(\theta)\rightsquigarrow$ Flow: $\partial_t\theta(t)=-\nabla C(\theta)$
- Inference Flow: $f_{\theta(t)}(x)\xrightarrow{t\to\infty}?$ for $x\in\{$training set, test set$\}$

2. ANN Training: Functional Formulation

- Neural Realization Function: $F^{(L)}:\mathbb{R}^P\to\mathcal{F}$ (Function Space) $\quad\theta\mapsto\left(f_\theta:\mathbb{R}^{n_0}\to\mathbb{R}^{n_L}\right)$
- Neural Tangent Kernel: $\Theta_{\theta,ij}(x,y)=\sum_{p=1}^P\partial_{\theta_p}f_{\theta,i}(x)\,\partial_{\theta_p}f_{\theta,j}(y)$ $\quad i,j\in\{1,\dots,n_L\}$, $x,y\in\mathbb{R}^{n_0}$, $\in$ Tangent of $F^{(L)}(\mathbb{R}^{n_0})$ at f_θ
- Initialization: $F^{(L)}(\theta)\in\mathcal{F}$ (Non-Gaussian)



- Functional Cost: $\mathcal{C}:\mathcal{F}\to\mathbb{R}$ $\quad f\mapsto\frac{1}{N}\sum_{i=1}^N c(f(x_i),z_i)$ (Functional Derivative of $\mathcal{C}$ at f_θ $\in\mathcal{F}$)
- Training Loss: $C=\mathcal{C}\circ F^{(L)}$ (Convex, Nonlinear) $\rightsquigarrow$ Chain Rule $\partial_{\theta_i}C(\theta)=D\mathcal{C}\big|_{f_\theta}\partial_{\theta_i}F^{(L)}(\theta)$
- Gradient Descent Step: $f_\theta\mapsto f_{\theta-\eta\nabla C(\theta)}\simeq f_\theta-\eta\sum_{p=1}^P\partial_{\theta_p}C(\theta)\,\partial_{\theta_p}F(\theta)$

$\rightsquigarrow$ Flow: $\partial_t f_{\theta(t)}=-\nabla_{\Theta_{\theta(t)}}\mathcal{C}\big|_{f_{\theta(t)}}=-\nabla_{\Theta_\theta}\mathcal{C}\big|_{f_\theta}$

Kernel Gradient: $\nabla_{@_\theta}\mathcal{C}\big|_{f_\theta}(x)=\left(\frac{1}{N}\sum_{k=1}^N\sum_{j=1}^{n_L}\Theta_{\theta,ij}(x,x_k)\frac{\partial}{\partial f_{\theta,j}(x_k)}\mathcal{C}\big|_{f_\theta}\right)_{i=1,\dots,n_L}$

3. Infinite-Width Limit: $n_1,\dots,n_{L-1}\to\infty$

- $\left(f_{\theta(0),i}(x)\right)_{i=1,\dots,n_L,\ x\in\mathbb{R}^{n_0}}\to Z$ centered Gaussian Process with $\mathrm{Cov}(Z_i(x),Z_j(y))=\Sigma(x,y)\delta_{ij}$ (Neal, 1996 [2])
- $\Theta_{\theta(0)}\xrightarrow{n_1,\dots,n_{L-1}\to\infty}\Theta^\infty\otimes\mathrm{Id}_{n_L}$ $\quad$ Explicitly Computable $\quad\left(\Theta^\infty\otimes\mathrm{Id}_{n_L}\right)_{ij}(x,y)=\Theta^\infty(x,y)\delta_{ij}$

} At Initialization

- $\left|\Theta_{\theta(t)}-\Theta_{\theta(0)}\right|\xrightarrow{\text{pointwise}}0\rightsquigarrow\partial_t f_{\theta(t)}=-\nabla_{\Theta^\infty\otimes\mathrm{Id}_{n_L}}\mathcal{C}\big|_{f_{\theta(t)}}$

} During Training

4. Consequences as $n_1,\dots,n_{L-1}\to\infty$

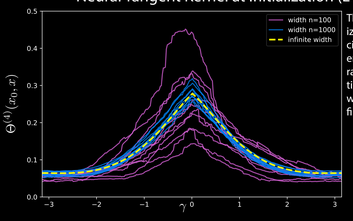
Optimal Emergent Learning

- Individually the weights evolve less and less: $|\theta_i(t)-\theta_i(0)|\xrightarrow{n_{L-1}\to\infty}0$ … yet learning occurs at each layer
- The influence of $(x_i,z_i)\in\mathbb{R}^{n_0+n_L}$ (Training) on the prediction $f_\theta(x)$ for an $x\in\mathbb{R}^{n_0}$ (Test) can be understood using Θ^∞
- Convergence to global min of $\mathcal{C}$ is guaranteed if Θ^∞ is positive-definite (Proven for $\|x_1\|=\dots=\|x_N\|$)

Least-Squares Regression

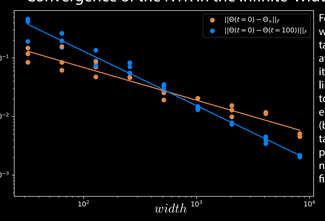
- $f_{\theta(t)}$ is Gaussian for any $t\geq0$
- $\mathbb{E}[f_\theta(t)]\xrightarrow{t\to\infty}$ Bayesian Max A Posteriori with respect to $\mathcal{N}(0,\Theta^\infty\otimes\mathrm{Id}_{n_L})$ prior $=\Theta^\infty\otimes\mathrm{Id}_{n_L}$ Kernel Ridge Regression with $\lambda\to0$ regularization
- Learning governed by the Θ^∞ PCA: conv. rate along $v_i\propto\lambda_i$ (PCA direction, PCA eigenvalue)

Neural Tangent Kernel at Initialization (L=4)



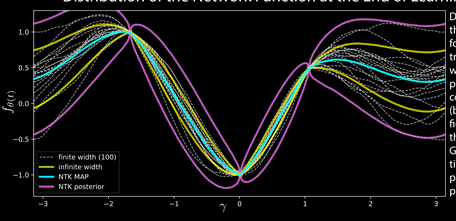
The NTK at initialization on the unit circle for 2 different widths (10 random initializations each), along with its infinite-width limit.

Convergence of the NTK in the Infinite-Width Limit



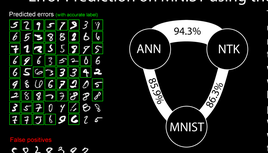
For a range of widths, the distances of the NTK at initialization to its infinite-width limit (orange) and to its limit at the end of training (blue). The distances are computed in Frobenius norm on a fixed batch.

Distribution of the Network Function at the End of Learning (L=3)



Distribution of f_θ at the end of training for an L^2 cost: 20 trials for an ANN of width 100, compared to the common mean (blue) and 80% confidence intervals of the infinite-width Gaussian distribution (yellow) and the posterior for NTK prior (pink).

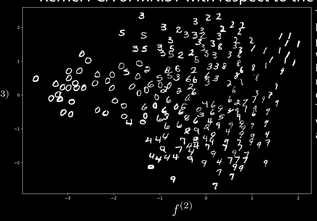
Error Prediction on MNIST using the NTK



The NTK can be used to predict the misclassifications that an ANN of width 500 with L=4 will make on MNIST. It yields a very good prediction on which wrong labels will be chosen by the ANN.

Prediction Coincidences

Kernel PCA of MNIST with respect to the NTK



The 2nd and 3rd kernel principal components of MNIST with respect to the NTK (the first component is essentially constant) with L=4. The three first eigenvalues are 41.03, 1.88, and 1.46.

References

[1] A. Jacot, F. Gabriel, C. Hongler Neural Tangent Kernel: Convergence and Generalization in Neural Networks, NeurIPS 2018, arXiv:1806.07572

[2] R.M. Neal, Bayesian Learning for Neural Networks, 1996

Acknowledgments

ERC CG CRITICAL (Gabriel)
ERC SG CONSTAMIS
Blavatnik Family Foundation (Hongler)
Latsis Family Foundation