Infinitely wide networks. Neural Tangent Kernel

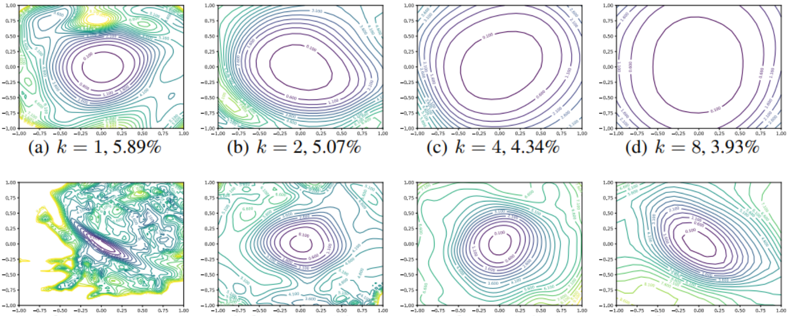
Maksim Velikanov

Wide neural networks

Observation: as width increases, various quantities become more regular.
Examples:

1) output probability distribution[1]

2) Loss surface[2]:



(a) $k = 1$, 5.89% (b) $k = 2$, 5.07% (c) $k = 4$, 4.34% (d) $k = 8$, 3.93%

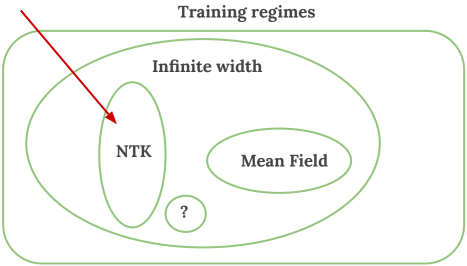
(e) $k = 1$, 13.31% (f) $k = 2$, 10.26% (g) $k = 4$, 9.69% (h) $k = 8$, 8.70%

[1] Google AI blog

[2] Hao Li et al., Visualizing the Loss Landscape of Neural Nets, arXiv:1712.09913

Infinite width limits

There are several (meaningful) infinite width limits depending on **initialisation** and **parametrization** of the network.



Network calculation from layer $l-1$ to l:

Mean Field regime

$$z_j^l = \frac{1}{n}\sum_{i=1}^{n} w_{ij}^l \phi(z_i^{l-1})$$
$$w_{ij}^l \sim O(1)$$

NTK regime

$$z_j^l = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} w_{ij}^l \phi(z_i^{l-1})$$
$$w_{ij}^l \sim \mathcal{N}(0,1)$$

NTK regime[3]

Main properties:

1. *Lazy training* - weights move very little during GD.
2. Linearization around weight initialization.
3. Can be defined naturally for deep networks of almost any architecture.
4. Network output is a Gaussian Process.
5. GD training can be fully described in the network output space.

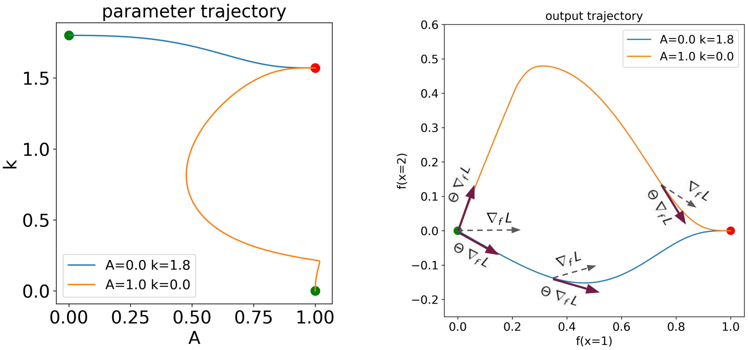
[3] A. Jacot et al., Neural Tangent Kernel: Convergence and Generalization in Neural Networks(2018), arXiv:1806.07572

Tangent Kernel (TK)

What does GD flow look like in model output space?

We will show that it is the velocity field of "naive" output GD $\nabla_f L$ multiplied by a matrix Θ (NTK), resulting in velocity $\Theta\nabla_f L$.

Toy example: sin model $f(\mathbf{W}, x) = A\sin(kx)$, with parameters $\mathbf{W} = \{A, k\}$ trained on dataset $\{(x_a, y_a)\}_{a=1}^2 = \{(1,1),(2,0)\}$



Gradient Descent: parameter perspective

Consider a parametric ML model $f(\mathbf{W}, \mathbf{x})$ with parameters $\mathbf{W}$ trained by GD on the empirical loss $L(\mathbf{W}) = \frac{1}{M}\sum_a l(f(\mathbf{W}, \mathbf{x}_a), y_a)$, where $l(\hat{y}, y)$ is the loss for a single prediction.

Discrete GD:
$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \eta \nabla_{\mathbf{W}} L(\mathbf{W})$

Continuous GD:
$\partial_t \mathbf{W}(t) = -\nabla_{\mathbf{W}} L(\mathbf{W})$

Loss gradient:

$$\nabla_{\mathbf{W}} L(\mathbf{W}) = \frac{1}{M}\sum_{a=1}^{M} \nabla_{\mathbf{W}} f(\mathbf{W}, \mathbf{x}_a) \cdot \nabla_{\hat{y}} l(f(\mathbf{W}, \mathbf{x}_a), y_a)$$

Functional perspective

Loss as a functional $\mathcal{L}$ of model outputs f:

$$\mathcal{L}[f] := \mathbb{E}_\mu[l(f(x), y)] = \int dxdy\mu(x,y)l(f(x), y)$$

$$\mu(x,y) = \frac{1}{M}\sum_{a=1}^{M}\delta(x - x_a)\delta(y - y_a)$$

"Functional" gradient descent:

$$\partial_t f(x,t) = -\frac{\delta\mathcal{L}[f]}{\delta f(x)}$$

Exercise. For a quadratic loss $l(\hat{y}, y) = \frac{1}{2}|\hat{y} - y|^2$ the solution of functional GD dynamics is

$$f(x_a, t) = y_a + e^{-t/M}\big(f(x_a, 0) - y_a\big)$$

Tangent Kernel

Key computation: Evolution of model output under parameter GD

$$\frac{df(\mathbf{W},x)}{dt} = \nabla_{\mathbf{W}} f(\mathbf{W},x) \cdot \frac{d\mathbf{W}}{dt} = -\nabla_{\mathbf{W}} f(\mathbf{W},x) \cdot \nabla_{\mathbf{W}} L(\mathbf{W})$$

$$= -\frac{1}{M}\sum_{a=1}^{M} \nabla_{\mathbf{W}} f(\mathbf{W},x) \cdot \nabla_{\mathbf{W}} f(\mathbf{W},x_a) \nabla_{\hat{y}} l(f(\mathbf{W},x_a),y_a)$$

$$= -\frac{1}{M}\sum_{a=1}^{M} \Theta(\mathbf{W},x,x_a) \nabla_{\hat{y}} l(f(\mathbf{W},x_a),y_a) = -\int dx' \Theta(\mathbf{W},x,x') \frac{\delta \mathcal{L}[f(x')]}{\delta f(x')}$$

Result: Parameter GD viewed in the output space is equivalent to "functional" GD transformed by Tangent Kernel $\Theta(x,x')$:

$$\Theta(x,x') = \sum_p \frac{\partial f(\mathbf{W},x)}{\partial W_p} \frac{\partial f(\mathbf{W},x')}{\partial W_p}$$

Neural Tangent Kernel (NTK) - Tangent Kernel of neural network

Generalization to vector-valued functions

Exercise. For vector-valued function $\mathbf{f} = (f_i)$, if $\mathcal{L}(\mathbf{f}) = \sum_i \mathcal{L}(f_i)$, then

$$\frac{df_i(\mathbf{W}, x)}{dt} = -\frac{1}{M} \sum_{a=1}^{M} \Theta_{ij}(\mathbf{W}, x, x_a) \nabla_{\hat{y}} l(f_j(\mathbf{W}, x_a), y_a),$$

where

$$\Theta_{ij}(x, x') = \sum_p \frac{\partial f_i(\mathbf{W}, x)}{\partial W_p} \frac{\partial f_j(\mathbf{W}, x')}{\partial W_p}$$

Tangent Kernel: properties

Exercise. Calculate TK for linear regression $f(w,b,x) = w^T x + b$

If only values of $f(x)$ on training dataset $\{x_a\}_{a=1}^M$ are concerned, it is sufficient to consider TK $\Theta(x,x')$ for $x, x' \in \{x_a\}_{a=1}^M$. In that case TK is a $Md \times Md$ matrix, where d is dimension of output vectors.

TK properties:

1. Tangent Kernel is a positive semi-definite matrix.
2. GD dynamics converges to global minimum if $l(\hat{y}, y)$ is convex and spectrum of Θ is separated from zero during training.
3. If $l(\hat{y}, y)$ is convex, then TK has a non-trivial null space at all spurious minima of loss function $L(\mathbf{W})$. (That's why it is possible to have spurious minima in the parameter space.)

Exercise: prove properties $1, 3$

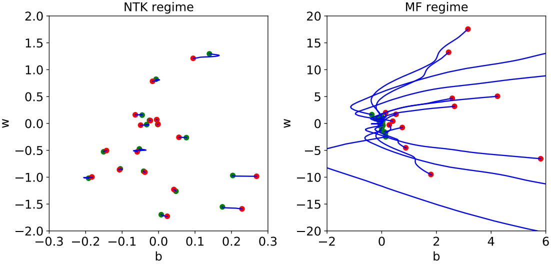
Network parametrization: NTK vs. MF

	forward pass	initialisation	limit law	depth
NTK	$z_j^l = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \phi(z_i^{l-1}) w_{ij}^l$	Gaussian	CLT	any
MF	$z_j^l = \frac{1}{n} \sum_{i=1}^{n} \phi(z_i^{l-1}) w_{ij}^l$	any	LLN	shallow

MF vs NTK parameter trajectories for a network

$$f(x) = \frac{1}{n^a} \sum_i^n (w_i x + b_i)_+$$

$$a = 1,\ \frac{1}{2}$$



In the NTK regime, weights need a smaller adjustment due to smaller a

Network parametrization: NTK vs. Standard

Standard parametrization

$$z_j^l = \sum_{i=1}^{n_{l-1}} h_i^{l-1} W_{ij}^l, \; h_i^l = \phi(z_i^l)$$

initialisation: $W_{ij}^l \sim \mathcal{N}\big(0, \frac{1}{n_{l-1}}\big)$

NTK parametrization

$$z_j^l = \frac{1}{\sqrt{n_{l-1}}} \sum_{i=1}^{n_{l-1}} h_i^{l-1} w_{ij}^l, \; h_i^l = \phi(z_i^l)$$

initialisation: $w_{ij}^l \sim \mathcal{N}\big(0, 1\big)$

In particular, $h_i^0 \equiv x_i$ - network input, and $z^L(x) \equiv f(x)$ - network output.

Exercise.
1) Two parametrizations have equivalent forward pass.
2) Two parametrizations with learning rates η (NTK), $\widetilde{\eta}$ (Standard) have equivalent backward pass if layer dependent learning rate is used in standard parametrization:

$$\widetilde{\eta}(W^l) = \frac{\eta}{n_{l-1}}$$

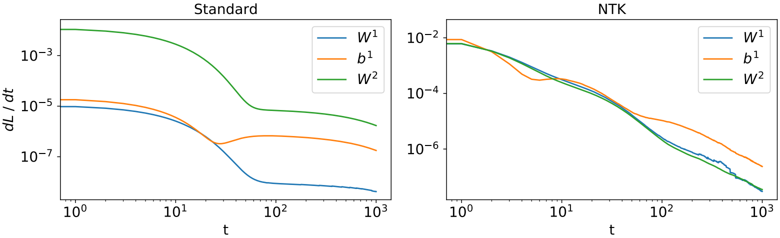
Normalizing backward pass

Exercise. For a parametric model f trained on $\{(x_a, y_a)\}_{a=1}^{M}$, show that

$$\frac{dL}{dt} = -\frac{1}{M^2}\sum_{a,b=1}^{M} \nabla_{\hat{y}} l(f_a, y_a)\Theta(x_a, x_b)\nabla_{\hat{y}} l(f_b, y_b)$$

To prove later: NTK parametrization normalizes backward pass in the limit $n_l \to \infty$, so that NTK's of all layers have the same magnitude.

Example: NN $f(x) = \frac{1}{\sqrt{n}}\sum_i W_i^2 \phi(W_i^1 x + b_i^1)$ approximating $y = x^2$.
Contributions of parameter types W^1, b^1, W^2 to loss change $dL \, / \, dt$



NTK regime

In the infinite width limit $n_l \to \infty$ the network has the following properties

1. Network output is a Gaussian Process (GP) at initialisation.
2. NTK Θ is deterministic.
3. NTK Θ is constant during training.

Consequence: Output dynamics is described by a closed equation with NTK $\Theta(x, x')$ calculated at initialisation

$$\frac{df(x)}{dt} = -\frac{1}{M} \sum_{a=1}^{M} \Theta(x, x_a) \nabla_{\hat{y}} l(f(x_a), y_a)$$

In particular, $f(x,t) - f(x, t=0) = \sum_{a=1}^{M} c_a(t) \Theta(x, x_a)$ with some coefficients $c_a(t)$ ("kernel model")

Exercise. For quadratic loss the GD dynamics is linear in output space.

GP property[4]

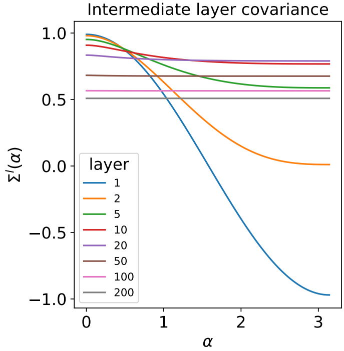
Theorem. In the infinite width limit outputs of all network layers $z_j^l(x)$ are GP with zero mean and covariance $\delta_{jj'}\Sigma^l(x,x')$, which is computed iteratively as

$$\begin{cases} \Sigma^1(x,x') = \frac{x^T x'}{n_0} \\ \Sigma^{l+1}(x,x') = \langle \phi(z^l(x))\phi(z^l(x'))\rangle, \quad z^l(x) \sim \mathcal{GP}(0, \Sigma^l(x,x')) \end{cases}$$

Remark. Limits $n_l \to \infty$ are taken consequently, starting from the first layer.

Exercise. $\Sigma^l(x,x')$ depends only on norms of x, x' and angle α between them.

Figure: $\Sigma^l(x,x')$ for ReLU network



[4] J. Lee et al., Deep neural networks as Gaussian Processes(2018), arXiv:1711.00165

GP property: proof

Induction base. For fixed x, x' output of the first layer $z^1_j(x)$ is a sum of Gaussian random variables. The covariance is

$$\langle z^1_j(x) z^1_{j'}(x') \rangle = \langle \frac{1}{n_0} \sum_{ii'} x_i x'_{i'} w^1_{ij} w^1_{i'j'} \rangle = \delta_{jj'} \frac{x^T x'}{n_0}$$

Induction step. $z^l_i(x)$ is a GP with covariance $\delta_{ii'} \Sigma^l(x, x')$, then for $i \neq i'$ pre-activations are independent. For $z^{l+1}_j(x)$ we apply CLT

$$z^{l+1}_j(x) = \frac{1}{\sqrt{n_l}} \sum_i \phi(z^l_i(x)) w^{l+1}_{ij} \to \mathcal{GP}(0, \Sigma^{l+1}_{jj'}(x, x'))$$

The covariance is

$$\begin{aligned}\langle z^{l+1}_j(x) z^{l+1}_{j'}(x') \rangle &= \frac{1}{n_l} \sum_{ii'} \langle \phi(z^l_i(x)) \phi(z^l_{i'}(x')) \rangle \langle w^{l+1}_{ij} w^{l+1}_{i'j'} \rangle \\ &= \delta_{jj'} \langle \phi(z^l_i(x)) \phi(z^l_i(x')) \rangle\end{aligned}$$

NTK at initialisation[5]

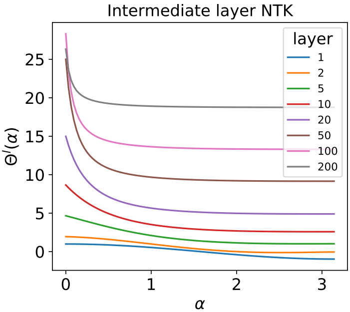
Theorem. In the infinite width limit, NTK's of all layers are deterministic, neuron diagonal $\delta_{jj'}\Theta^L(x,x')$, and computed as

$$\begin{cases} \Theta^1(x,x') = \Sigma^1(x,x') \\ \Theta^{l+1}(x,x') = \Theta^l(x,x')\langle\dot{\phi}(z^l(x))\dot{\phi}(z^l(x'))\rangle + \Sigma^l(x,x') \\ z^l(x) \sim \mathcal{GP}(0,\Sigma^l(x,x')) \end{cases}$$

Remark: Parameter gradients

$$\frac{\partial f(\mathbf{W},x)}{\partial w^l_{ij}}, \frac{\partial f(\mathbf{W},x)}{\partial b^l_j}$$

are dependent random(not deterministic) variables.



[5] A. Jacot et al., Neural Tangent Kernel: Convergence and Generalization in Neural Networks(2018), arXiv:1806.07572

Proof. Step 1 - finite network

Define intermediate layer NTK $\Theta^l_{jj'}(x,x')$

$$\Theta^l_{jj'}(x,x') \equiv \nabla_{w^l} z^l_j(x) \nabla_{w^l} z^l_{j'}(x') + \nabla_{\mathbf{w}^{<l}} z^l_j(x) \nabla_{\mathbf{w}^{<l}} z^l_{j'}(x')$$

First layer: $\Theta^1_{jj'} = \delta_{jj'} \frac{x^T x'}{n_0}$. Then $\Theta^l_{jj'}$ is computed iteratively from $\Theta^{l-1}_{jj'}$.
Contribution from the current layer:

$$\nabla_{w^l} z^l_j \nabla_{w^l} z'^l_{j'} = \frac{1}{n_{l-1}} \sum_{km} \nabla_{w^l_{km}} \left(\sum_i w^l_{ij} h^{l-1}_i \right) \nabla_{w^l_{km}} \left(\sum_{i'} w^l_{i'j'} h'^{l-1}_{i'} \right)$$
$$= \delta_{jj'} \frac{1}{n_{l-1}} \sum_i h^{l-1}_i h'^{l-1}_i$$

Contribution from previous layers:

$$\nabla_{\mathbf{w}^{<l}} z^l_j \nabla_{\mathbf{w}^{<l}} z'^l_{j'} = \sum_{ii'} \frac{\partial z^l_j}{\partial z^{l-1}_i} \frac{\partial z'^l_{j'}}{\partial z'^{l-1}_{i'}} \nabla_{\mathbf{w}^{<l}} z^{l-1}_i \nabla_{\mathbf{w}^{<l}} z'^{l-1}_{i'}$$
$$= \frac{1}{n_{l-1}} \sum_{ii'} w^l_{ij} w^l_{i'j'} \dot{\phi}(z^{l-1}_i) \dot{\phi}(z'^{l-1}_{i'}) \Theta^{l-1}_{ii'}$$

Proof. Step 2 - taking the limit

We take limits $n_l \to \infty$ sequentially, starting from the first layer.
Assume that after $n_{l-2} \to \infty$ NTK is diagonal: $\Theta^{l-1}_{ii'} = \delta_{ii'}\Theta^{l-1}$.
Contribution from the current layer:

$$\delta_{jj'}\frac{1}{n_{l-1}}\sum_i h_i^{l-1}h_i'^{l-1} \underset{n_{l-1}\to\infty}{\longrightarrow} \delta_{jj'}\langle h^{l-1}h'^{l-1}\rangle = \delta_{jj'}\Sigma^l(x,x')$$

Contribution from previous layers:

$$\begin{aligned}
&\frac{1}{n_{l-1}}\sum_{ii'} w^l_{ij}w^l_{i'j'}\dot\phi(z_i^{l-1})\dot\phi(z_{i'}'^{l-1})\Theta^{l-1}_{ii'} \\
&= \Theta^{l-1}\frac{1}{n_{l-1}}\sum_i w^l_{ij}w^l_{ij'}\dot\phi(z_i^{l-1})\dot\phi(z_i'^{l-1}) \\
&\underset{n_{l-1}\to\infty}{\longrightarrow} \Theta^{l-1}\langle w^l_{ij}w^l_{ij'}\rangle\langle\dot\phi(z^{l-1})\dot\phi(z'^{l-1})\rangle = \delta_{jj'}\Theta^{l-1}\langle\dot\phi(z^{l-1})\dot\phi(z'^{l-1})\rangle
\end{aligned}$$

NTK during training

Theorem[6]**.** For any failure probability $\delta > 0$ there exist $C > 0$ and $N \in \mathbb{N}$, such that change of empirical NTK $\hat{\Theta}_t(x, x')$ is bounded, if learning rate $\eta_0 < \frac{1}{\lambda_{\mathsf{max}}(\Theta)}$

$$\sup_t |\hat{\Theta}_t(x, x') - \hat{\Theta}_0(x, x')| \leq \frac{C}{\sqrt{n}}, \quad \text{for any width} \quad n > N$$

Difficulty: Need to work with finite network away from initialisation.

We will show stability of NTK during training for a simple univariate shallow NN $f(x) = \frac{1}{\sqrt{n}} \sum_i v_i \phi(u_i x)$. Expression for NTK:

$$\Theta(x, x') = \frac{1}{n} \sum_i \left[\phi(u_i x)\phi(u_i x') + v_i^2 x x' \dot{\phi}(u_i x)\dot{\phi}(u_i x') \right]$$

[6] J. Lee et al., Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent(2019), arXiv:1902.06720

NTK during training. Illustration

Consider change of parameters at single GD step.

$$\Delta v_i = -\eta \frac{1}{M} \sum_{a=1}^{M} \frac{\partial f(x_a)}{\partial u_i} \frac{\partial l(f(x_a), y_a)}{\partial f(x_a)} = -\eta \frac{1}{\sqrt{n}} \sum_a \phi(u_i x_a) L'_a \sim \frac{1}{\sqrt{n}}$$

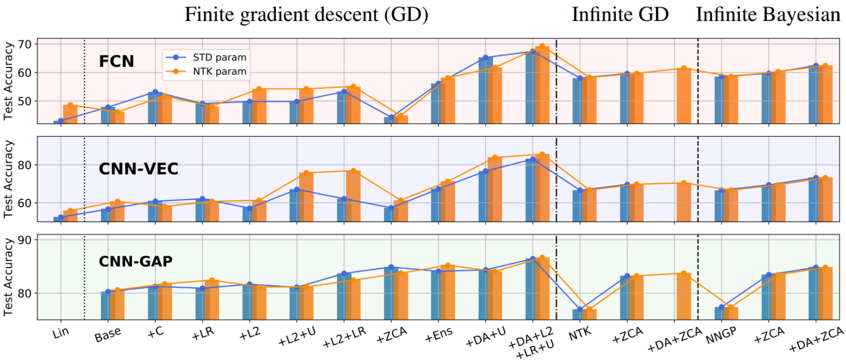
Here $L'_a \equiv \frac{1}{M} \frac{\partial l(f(x_a), y_a)}{\partial f(x,a)}$. For u_i we similarly get

$$\Delta u_i = -\eta \frac{1}{\sqrt{n}} v_i \sum_a \dot{\phi}(u_i x_a) x_a L'_a \sim \frac{1}{\sqrt{n}}$$

The change of NTK

$$\begin{aligned}\Delta\Theta(x, x') =& \frac{1}{n} \sum_i \Delta u_i \Big[x \dot{\phi}(u_i x) \phi(u_i x') + \phi(u_i x) x' \dot{\phi}(u_i x') \\ & + v_i^2 x x' \big(x \ddot{\phi}(u_i x) \dot{\phi}(u_i x') + \dot{\phi}(u_i x) x' \ddot{\phi}(u_i x') \big) \Big] \\ & + \frac{1}{n} \sum_i \Delta v_i \left[2 v_i x x' \dot{\phi}(u_i x) \dot{\phi}(u_i x') \right] \sim \frac{1}{\sqrt{n}}\end{aligned}$$

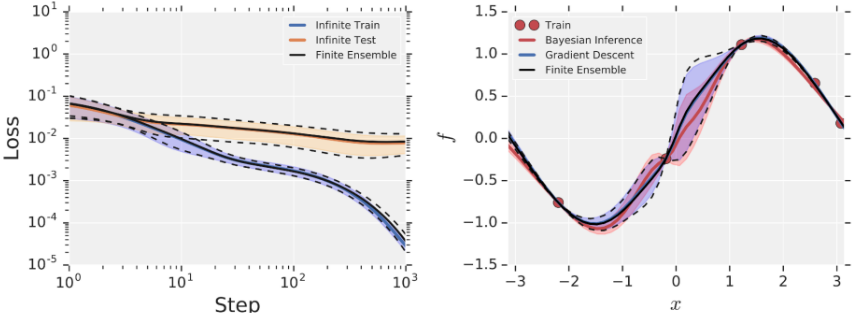
Exact computation with infinite nets or linearized nets[7]



[7] J. Lee et al., Finite Versus Infinite Neural Networks: an Empirical Study(2020), arXiv:2007.15801

Exact Bayesian Inference

- Output of infinite width NN stays Gaussian process throught training
- Mean and covariance are analytically calculable



Neural Tangent Kernel. Exercise solutions.

Maksim Velikanov

Exercise. For a quadratic loss $l(\hat{y}, y) = \frac{1}{2}|\hat{y} - y|^2$ the solution of functional GD dynamics is

$$f(x_a, t) = y_a + e^{-t}\big(f(x_a, 0) - y_a\big)$$

Solution. In the case of finite training dataset variational derivative $\delta/\delta f(x)$ is usual gradient.

$$\big(\nabla_f L\big)_a = \frac{1}{M}(f(x_a) - y_a)$$

Then

$$\frac{df(x_a, t)}{dt} = -\frac{1}{M}(f(x_a, t) - y_a)$$

This is a linear 1d ODE, with exponential solution. Importantly, evolution of different points is independent

Exercise. Calculate TK for linear regression $f(w, b, x) = w^T x + b$

Solution. Using definition

$$\begin{aligned}
\Theta(x, x') &= \sum_p \frac{\partial f(\mathbf{W}, x)}{\partial W_p} \frac{\partial f(\mathbf{W}, x')}{\partial W_p} \\
&= \sum_{i=1}^{d} \frac{\partial f(w, b, x)}{\partial w_i} \frac{\partial f(w, b, x')}{\partial w_i} + \frac{\partial f(w, b, x)}{\partial b} \frac{\partial f(w, b, x')}{\partial b} \\
&= \sum_{i=1}^{d} x_i x_i' + 1 = x^T x' + 1
\end{aligned}$$

Exercise. Tangent Kernel is a positive semi-definite matrix.

Solution. Choose a single parameter W_p. Its contribution to TK is

$$\Theta_{ij,p}(x_a, x_b) = \frac{\partial f_i(\mathbf{W}, x_a)}{\partial W_p} \frac{\partial f_j(\mathbf{W}, x_b)}{\partial W_p} = |v_p\rangle\langle v_p|$$

where $|v_p\rangle$ is a Md dimensional vector(we used bra-ket notation)

$$v_{ia} = \frac{\partial f_i(\mathbf{W}, x_a)}{\partial W_p}$$

Thus, the contribution of single parameter to NTK is an outer product of vector with itself. Such outer products, as well as any sum of them, are positive semi-definite

$$\Theta = \sum_p |v_p\rangle\langle v_p|$$

Exercise. If $l(\hat{y}, y)$ is convex, then TK has a non-trivial null space at all spurious minima of Loss function $L(\mathbf{W})$. (That's why it is possible to have spurious minima in the parameter space.)

Solution. At any local minima of $L(\mathbf{W})$ the model otput is stationary

$$\frac{df_i(x_a, t)}{dt} = 0 = \frac{1}{M} \sum_{j=1}^{d} \sum_{b=1}^{M} \Theta_{ij}(x_a, x_b) \frac{\partial l(f(x_b), y_b)}{\partial f_j(x_b)} = \Theta \nabla_f L$$

If minimum $\mathbf{W}_0$ is spurious, then $L(\mathbf{W_0}) \neq 0$. Due to convexity of $l(\hat{y}, y)$ w.r.t. first argument output gradient is non-zero $\nabla_f L \neq 0$. Thus

$$\dim \ker \Theta > 0, \quad \text{and} \quad \nabla_f L \in \ker \Theta$$

Exercise. Two parametrizations with learning rates $\eta, \widetilde{\eta}$ have equivalent backward pass if layer dependent learning rate is used in standard parametrization:

$$\widetilde{\eta}(W^l) = \frac{\eta}{n_{l-1}}$$

Solution. Each trainable parameter in standard parametrization is obtained from corresponding parameter in NTK parametrization by scaling

$$\widetilde{W}^l_{ij} = \frac{1}{\sqrt{n_{l-1}}} w^l_{ij}$$

For simplicity consider a single parameter p: $\widetilde{W}_p = \lambda_p W_p$. Its change in NTK and Standard parametrizations

$$\Delta W_p = -\eta \frac{\partial L(\mathbf{W})}{\partial W_p}, \qquad \Delta \widetilde{W}_p = -\widetilde{\eta}_p \frac{\partial L(\widetilde{\mathbf{W}})}{\partial \widetilde{W}_p} = -\frac{\widetilde{\eta}_p}{\lambda_p} \frac{\partial L(\mathbf{W})}{\partial W_p}$$

For equivalence, the "actual" change of parameter should be the same $\Delta \widetilde{W}_p = \lambda_p \Delta W_p$. Substituting changes we get $\widetilde{\eta}_p = \lambda_p^2 \eta$.

Exercise. For a paramtric model f trained on $\{(x_a, y_a)\}_{a=1}^M$, show that

$$\frac{dL}{dt} = -\frac{1}{M^2} \sum_{a,b=1}^{M} \nabla_{\hat{y}} l(f_a, y_a) \Theta(x_a, x_b) \nabla_{\hat{y}} l(f_b, y_b)$$

Solution. Just compute time derivative of L

$$\frac{dL(f(\mathcal{X}))}{dt} = \left(\nabla_{f(\mathcal{X})} L\right)^T \frac{df(\mathcal{X})}{dt} = -\left(\nabla_{f(\mathcal{X})} L\right)^T \Theta \nabla_{f(\mathcal{X})} L$$

$$= -\frac{1}{M^2} \sum_{a,b=1}^{M} \nabla_{\hat{y}} l(f_a, y_a) \Theta(x_a, x_b) \nabla_{\hat{y}} l(f_b, y_b) = -\left(\nabla_f L\right)^T \Theta \nabla_f L$$

Exercise. For quadratic loss GD dynamics is linear output spac e.

Solution. For quadratic loss we have $\nabla_f(\mathcal{X})L(f(\mathcal{X})) = \frac{1}{M}(f(\mathcal{X}) - \mathcal{Y})$. Then the dynamics reads

$$\frac{df(\mathcal{X})}{dt} = -\frac{1}{M}\Theta\left(f(\mathcal{X}) - \mathcal{Y}\right)$$

Note that NTK Θ is constant and fixed, thus we obtained multidimensional linear ODE.