Neural Tangent Kernels

Semyon Malamud

EPFL

Table of Contents

What About Neural Networks?

- Neural Networks are Complicated Animals
- Most importantly, they are trained by gradient descent
- Even more importantly, they are trained end-to-end, with forward pass (=evaluating the NN) and backward pass (=computing the full gradient using the superposition formula)
- This is incredibly complicated because, contrary to the random feature model, all NN weights are trained.

The Neural Tangent Kernel i

- Consider a generic NN
$$f(x;\theta) \tag{1}$$
and consider its gradient $\nabla_\theta f(x;\theta) \in \mathbb{R}^{1\times P}$, where P is the number of parameters (weights) of the model.
- What happens when we train it? Consider first the lazy training regime. Namely, suppose we change the weights a little bit and try to match the labels y,
$$\begin{aligned} f(x;\theta+\Delta\theta) &\approx y \\ \Leftrightarrow f(x;\theta) + \underbrace{\Delta\theta^\top}_{optimal\ weight\ change} \nabla_\theta f(x;\theta) &= y \end{aligned} \tag{2}$$
and hence, to ding the optimal weight change $\Delta\theta$, we are running a regression of y on features $S_i = \nabla_\theta f(x_i;\theta) \in \mathbb{R}^P$

The Neural Tangent Kernel ii

- If w is random, these are random features
- Thus,

$$\Delta\theta = \underbrace{(zI + S'S)^{-1}S'}_{\beta}\underbrace{(y - f(x;\theta))}_{residual} \tag{3}$$

- Equivalently, we can define the **Neural Tangent Kernel**

$$\begin{aligned} K(x_i, x_j; \theta) &= \nabla_\theta f(x_i;\theta)^\top \nabla_\theta f(x_j;\theta) \\ &= \sum_k \frac{\partial}{\partial\theta_k} f(x_i;\theta) \frac{\partial}{\partial\theta_k} f(x_j;\theta) \end{aligned} \tag{4}$$

and we get

$$\begin{aligned} f(x;\theta+\Delta\theta) &\approx \hat{f}(x;\theta) \\ &= f(x;\theta) + K(x;X;\theta)^\top (zI + K(X,X;\theta))^{-1}\underbrace{(y - f(x;\theta))}_{residual} \end{aligned} \tag{5}$$

The Neural Tangent Kernel iii

- That is, **a single optimization step = kernel regression with NTK**
- What about multi-step optimization?

Table of Contents

Suppose we have $f(x;\theta)$ and we would like to solve

$$\min_\theta \mathcal{L}(\theta),\ \mathcal{L}(\theta) \ = \ \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i;\theta))\,. \tag{6}$$

We are going to try solving it by gradient descent (but we actually cannot because it is NP-hard).

To this end, we pick a learning rate η and run

$$\theta_{t+1} \ = \ \theta_t \ - \ \eta\,\nabla_\theta\mathcal{L}(\theta_t)\,. \tag{7}$$

This is a dynamical system. It is highly non-linear and complex. Nobody knows how it behaves in general.

Our first simplification will be to assume η is small. This is innocent, right?

No! Catapults in SGD

When the learning rate η is small enough, we get

$$f(x;\theta_{t+1}) \;=\; f(x;\theta_t \;-\; \eta\,\nabla_\theta\mathcal{L}(\theta_t)) \\ \underbrace{\approx}_{first\;\;order\;\;Taylor\;\;approximation} \qquad f(x;\theta_t) \;-\; \eta\nabla_\theta f(x;\theta_t)'\nabla_\theta\mathcal{L}(\theta_t)\,. \tag{8}$$

Now,

$$\begin{aligned} \nabla_\theta\mathcal{L}(\theta) \;&=\; \nabla_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i;\theta))\right) \\ &=\; \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta\ell(y_i, f(x_i;\theta)) \\ &=\; \frac{1}{n}\sum_{i=1}^{n}\ell_{\hat{y}}(y_i, f(x_i;\theta))\,\nabla_\theta f(x_i;\theta) \end{aligned} \tag{9}$$

Substituting, we obtain *gradient descent in the prediction space:*

$$\begin{aligned} &\nabla_\theta f(x;\theta_t)'\nabla_\theta \mathcal{L}(\theta_t) \\ &= \nabla_\theta f(x;\theta_t)' \left(\sum_{i=1}^{n} \ell_{\hat{y}}(y_i, f(x_i;\theta_t))\, \nabla_\theta f(x_i;\theta_t) \right) \\ &= \sum_{i=1}^{n} \ell_{\hat{y}}(y_i, f(x_i;\theta_t))\, \nabla_\theta f(x;\theta_t)'\nabla_\theta f(x_i;\theta_t)\,. \end{aligned} \tag{10}$$

Let us introduce the *Tangent Kernel*

$$K(x,\tilde{x};\theta_t) \;=\; \nabla_\theta f(x;\theta_t)'\nabla_\theta f(\tilde{x};\theta_t)\,. \tag{11}$$

Clearly, this is a positive-definite kernel. This kernel is random because it depends on θ in subtle ways, and *evolves through training.* When the parametric family $f(x;\theta)$ is a Neural Network, it is called a *Neural*

Tangent Kernel (NTK). Using this kernel, replacing η with ηdt, and taking the limit as $dt \to 0$, we can rewrite

$$\begin{aligned} f(x;\theta_{t+1}) &= f(x;\theta_t) - \eta dt \nabla_\theta f(x;\theta_t)' \nabla_\theta \mathcal{L}(\theta_t) \\ \frac{f(x;\theta_{t+1}) - f(x;\theta_t)}{dt} &= -\eta \nabla_\theta f(x;\theta_t)' \nabla_\theta \mathcal{L}(\theta_t) \end{aligned} \tag{12}$$

Theorem (Predictions Dynamics for Gradient Flow)

$$\frac{d}{dt} f(x;\theta_t) = -\eta K(x, X;\theta_t) \frac{1}{n} \ell_{\hat{y}}(y; f(X;\theta_t)), \tag{13}$$

where $X = (x_i)_{i=1}^n$ *and*
$K(x, X;\theta_t) \in \mathbb{R}^{1\times n},\ \ell_{\hat{y}}(y; f(X;\theta_t)) = (\ell_{\hat{y}}(y_i, f(x_i;\theta_t)))_{i=1}^n \in \mathbb{R}^{n\times 1}$.

MSE Dynamics i

- When $\ell(y, \hat{y}) = 0.5(y - \hat{y})^2$, we get

$$\frac{d}{dt} f(x; \theta_t) = -\eta K(x, X; \theta_t) \frac{1}{n}(f(X; \theta_t) - y) . \tag{14}$$

- Thus, for $x = X$, we get

$$\frac{d}{dt} f(X; \theta_t) = -\eta K(X, X; \theta_t) \frac{1}{n}(f(X; \theta_t) - y) . \tag{15}$$

- Let

$$\hat{K}_t = n^{-1} K(X, X; \theta_t), \ u_t = (f(X; \theta_t) - y) .$$

  Then,

$$u_t' = -\eta A_t \, u_t . \tag{16}$$

  Since A_t is moving, there is no closed form solution.

MSE Dynamics ii

- The remarkable discovery of Neural Tangent Kernel implies that $K(x_i, x_j; \theta)$ is independent of θ for very (infinitely!) wide NNs. See also Conditions for constant NTK
- if $A_t = A$, we get

$$u_t' = -\eta A\, u_t \Leftrightarrow u_t = e^{-\eta t A} u_0 \tag{17}$$

  that is the in-sample predictions are

$$f(X;\theta_t) - y = e^{-\eta t A}(\underbrace{f(X;\theta_0)}_{initial\ \ seed} - y) \tag{18}$$

- convergence to interpolation when A is positive definite.

MSE Dynamics iii

- The OOS predictions are then

$$
\begin{aligned}
\frac{d}{dt} f(x;\theta_t) &= -\eta K(x,X) n^{-1} (f(X;\theta_t) - y) \\
\frac{d}{dt} f(x;\theta_t) &= -\eta K(x,X) n^{-1} e^{-\eta t A} (f(X;\theta_0) - y)
\end{aligned} \tag{19}
$$

and the solution is

$$
\begin{aligned}
f(x;\theta_t) &= \underbrace{f(x;\theta_0)}_{initial\ \ random\ \ seed} \\
&+ K(x;X)\, K(X;X)^{-1} (I - e^{-\eta t n^{-1} K(X;X)})(y - f(X;\theta_0))
\end{aligned} \tag{20}
$$

- In the infinite epoch limit, we get Kernel Regression:

$$
f(x;\theta_t) = \underbrace{f(x;\theta_0)}_{initial\ \ random\ \ seed} + K(x;X)\, K(X;X)^{-1} (y - f(X;\theta_0)) \tag{21}
$$

MSE Dynamics iv

- This is a family of interpolators indexed by the random seed θ_0; the final output always depends on θ_0. Thus, contrary to the linear regression, we do not converge to the unique minimal norm interpolator!
- For finite t, this is **spectral shrinkage**:

$$
\begin{aligned}
A &= K(X;X) = UDU' \\
(zI + A)^{-1} &= A^{-1}\underbrace{\left(A(zI + A)^{-1}\right)}_{\leq 1} \leq z^{-1} I \\
A^{-1}\underbrace{\left(I - e^{-\eta t n^{-1} A}\right)}_{\leq 1} &\leq \eta t n^{-1} I
\end{aligned}
\tag{22}
$$

Table of Contents

Proposition

Suppose that f is differentiable, $\ell_{\hat{y}}$ is continuous, and that ℓ is such that, for any $R > 0$, the set $\{v \in \mathbb{R} : \ell(y, v) < R\}$ is bounded and $\ell \geq -A$ for some $A > 0$. Suppose also that NTK stabilizes after T epochs: $\|K(x; X; \theta_t) - K(x; X; \theta_T)\| \leq \varepsilon$ for all $t \geq T$. Then,

$$f(x; \theta_t) \;=\; \underbrace{f(x; \theta_T)}_{trained\ \ DNN} \;+\; \underbrace{K(x, X; \theta_T)\mathcal{U}_t}_{trained\ \ kernel\ \ machine} \;+\; O(\varepsilon) \tag{23}$$

for some vector $\mathcal{U}_t$ that depends on the training data.

- The link between NTK and DNN is particularly clear for the MSE loss $\ell(y, \hat{y}) = (y - \hat{y})^2$. In this case (23) takes the form

$$f(x;\theta_t) \approx f(x;\theta_T) + K(x, X;\theta_T)(zI + K(X, X;\theta_T))^{-1}(y - f(X;\theta_T)) \tag{24}$$

  for some ridge parameter z.
- In other words, residual DNN training (for $t > T$) experiences a form of "gradient boosting" in which DNN residuals $y - f(X;\theta_T)$ are fit via kernel ridge regression (the kernel ridge predictor uses $\mathcal{U}(z) = (n^{-1}K(X, X;\theta_T) + zI)^{-1}y$ rather than the implicit $\mathcal{U}_t$ of (23)). Note that the after-training NTK in (23) is not directly optimized, it is just evaluated at the trained DNN parameters.

- A striking discovery is that replacing $f(x;\theta_T)$ with 0 and changing z with a judiciously chosen $\tilde{z}$ gives approximately the same result,

$$\begin{aligned} & f(x;\theta_T) \ + \ K(x,X;\theta_T)(zI + K(X,X;\theta_T))^{-1}(y - f(X;\theta_T)) \\ & \approx \ K(x,X;\theta_T)(\tilde{z}I + K(X,X;\theta_T))^{-1}y\,. \end{aligned} \tag{25}$$

- Prediction performance of the after-training NTK $K(\cdot,\cdot;\theta_T)$ often matches or surpasses the DNN predictor $f(x;\theta_T)$.
- Evidently, the kernel component is not just a booster, it is the main event.

[**Proof**]We have

$$\frac{d}{dt} f(x;\theta_t) \;=\; -\eta K(x, X;\theta_t)\,\frac{1}{n}\ell_{\hat{y}}(y; f(X;\theta_t))\,, \tag{26}$$

Our first observation is that

$$\frac{d}{dt}\sum_{i=1}^{n} \ell(y_i, f(X_i;\theta_t)) \;=\; -\ell_{\hat{y}}(y; f(X;\theta_t))^\top \eta K(x, X;\theta_t)\,\ell_{\hat{y}}(y; f(X;\theta_t)) \;\leq\; 0 \tag{27}$$

because K is positive semi-definite. From the assumptions made about ℓ, we immediately get that $f(X;\theta_t)$ stays uniformly bounded. Let $\check{f}$ satisfy

$$\frac{d}{dt}\check{f}(x;\theta_t) \;=\; -\eta K(x, X;\theta_T)\,\frac{1}{n}\ell_{\hat{y}}(y; f(X;\theta_t))\;,\;\; \check{f}(x;\theta_T) = f(x;\theta_T)\,. \tag{28}$$

Then,

$$
\begin{aligned}
&\|\check{f}(x;\theta_t) - f(x;\theta_t)\| \\
&= \ \| \int_t^T \eta(K(x,X;\theta_\tau) - K(x,X;\theta_T))\frac{1}{n}\ell_{\hat{y}}(y;f(X;\theta_\tau))d\tau\| \\
&\le \ \int_t^T \eta\|(K(x,X;\theta_\tau) - K(x,X;\theta_T))\frac{1}{n}\ell_{\hat{y}}(y;f(X;\theta_\tau))\|d\tau \\
&\le \ \varepsilon\,\eta(t-T)\sup_{\tau\in[T,t]}\|\frac{1}{n}\ell_{\hat{y}}(y;f(X;\theta_\tau))\|,
\end{aligned}
\tag{29}
$$

where the latter supremum is finite because $\ell_{\hat{y}}$ is continuous and $f(X;\theta_t)$ stays uniformly bounded . The claim now follows because

$$
\check{f}(x;\theta_t) \ = \ f(x;\theta_T) - K(x,X;\theta_T)\eta\int_t^T \frac{1}{n}\ell_{\hat{y}}(y;f(X;\theta_\tau))d\tau \tag{30}
$$