Highlights: Neural Tangent Kernel

Antônio H. Ribeiro, Dave Zachariah, Per Mattsson

Notes about the paper *Neural Tangent Kernel: Convergence and Generalization in Neural Networks* (Jacot et al., 2018).

1 Setup

Let $f(\cdot;\theta) : \mathbb{R}^d \to \mathbb{R}$ be an almost everywhere diferentiable function parametrized by $\theta \in \mathbb{R}^p$. Assume given a training dataset $\{(x_i, y_i)\}_{i-1}^n$ of input and outputs. We define the cost function

$$V(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)^2. \tag{1}$$

Taking the derivative of V we obtain,

$$\nabla_\theta V(\theta) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)\nabla_\theta f(x_i;\theta). \tag{2}$$

Now assume the parameter θ is estimated using gradient flow, let θ_t be the parameters estimated at the instant t. Then,

$$\frac{d\theta_t}{dt} = -\eta\nabla_\theta V(\theta), \tag{3}$$

and the chain rule yields

$$\frac{df(z,\theta_t)}{dt} = \eta\nabla_\theta V(\theta)^\mathsf{T}\nabla_\theta f(z;\theta) = \frac{\eta}{n}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)\left(\nabla_\theta f(x_i;\theta)^\mathsf{T}\nabla_\theta f(z;\theta)\right). \tag{4}$$

We define the Neural Tangent Kernel, $K(\cdot,\cdot;\theta) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

$$K(x, z;\theta_t) = \nabla_\theta f(x;\theta)^\mathsf{T}\nabla_\theta f(z;\theta). \tag{5}$$

This is the kernel associated with the feature map $x \mapsto \nabla_\theta f(x, ;\theta)$. It then follows that

$$\frac{df(z,\theta_t)}{dt} = \frac{\eta}{n}\sum_{i=1}^{n}(f(x_i;\theta) - y_i)K(x_i, z;\theta). \tag{6}$$

2 Model

Here they consider $f(x;\theta) = \tilde{\alpha}^{(\ell)}(x,\theta)$ is the output of a neural network with L layers. that could be defined recursively as:

$$\begin{aligned} \alpha^{(0)}(x;\theta) &= x \\ \tilde{\alpha}^{(\ell+1)}(x;\theta) &= \frac{1}{\sqrt{n_\ell}}W^{(\ell)}\alpha^{(\ell)}(x;\theta) + \beta b^{(\ell)} \\ \alpha^{(\ell)}(x;\theta) &= \sigma\left(\tilde{\alpha}^{(\ell)}(x;\theta)\right) \end{aligned} \tag{7}$$

where the nonlinearity σ is applied entrywise and β is a scaling factor. Here $\theta = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, \cdots, W^{(L)}, b^{(L)})$. At initialization, each entry of W^ℓ or b^ℓ is sampled from i.i.d Gausians $\mathcal{N}(0,1)$. Hence, θ_0 is a random variable.

3 Challenges

Now there are two elements that make the model hard to deal with using traditional tools.

1. **The kernel is stochastic.** For the models studied θ_0 is a random variable. Hence $K(\cdot, \cdot; \theta_0)$ is not deterministic.

2. **The kernel is parametrized.** The kernel depends on a parameter θ that varies which is itself being updated during training. Hence $K(\cdot, \cdot; \theta)$ is a kernel that evolves with the training. This reflects making the Eq. 6 not linear.

Solutions that are proposed in the paper:

1. (Theorem 1) In probability, $K(x, y; \theta_0) \to K_0(x, y)$, i.e. where K_0 is a deterministic kernel.

2. (Theorem 2) Uniformly on t, $K(x, y; \theta_t) \to K_0(x, y)$ for all $t \in [0, T]$.

4 Relation with other models

We have presented another linear model approximation before, the one that assumed that $\theta = \theta_0 + \beta$ and the number of parameters is so large that training effectively only changes the parameter by a small amount. Them β is small and:

$$f(z; \theta) \approx f(x; \theta_0) + \nabla_\theta f(x; \theta_0)^\intercal \beta.$$

In the case, $f(z; \theta_0)$ this would come down to the map: $x \mapsto \nabla_\theta f(x; \theta_0)$ followed by the estimation of β using a linear parameter. Notice that this nonlinear map coincides with the nonlinear map we are considering here.

References

A. Jacot, F. Gabriel, C. Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks In *Advances in Neural Information Processing Systems 32*, 2018.