

A More General Convolutional Neural Networks

Author Name

Affiliation

email@example.com

1 Appendix

We first consider function $\mathbf{f} : \mathbb{R}^{\times_{i=1}^d h_i \times n_0} \rightarrow \mathbb{R}^{\times_{i=1}^d h_i \times n_{L-1}}$ without head, as in practice, we suppose that $s_i = s$ for any i . For entries $f_{\mathbf{p}}^{(l)}$ ($1 \leq l \leq L-1$, and note this symbol omits a subscript j that denotes the $f_{\mathbf{p}}^{(l)} = f_{j,\mathbf{p}}^{(l)}$. As can be seen in the paper, this notation can actually be omitted. in which $\mathbf{p} = \{p_1, p_2, \dots, p_d\}$. Furthermore, we define neighbourhood of \mathbf{p} as $N(\mathbf{p}) = \{\cap_{i=1}^d (p_i - s/2, p_i + s/2) \cap \mathbb{Z} | 1 \leq i \leq d\}$. Kernel of samples x, x' as entry is induced by:

$$\begin{aligned}
K_{\mathbf{p},\mathbf{q}}^{(l)}(x, x') &= \nabla_{\theta}^T f_{\mathbf{p}}^{(l)}(x) \nabla_{\theta} f_{\mathbf{q}}^{(l)}(x') \\
&= \nabla_{\theta}^T \sum_{i=1}^{n_l} \left(\mathcal{W}_{ij}^{(l)} * \phi_{N(\mathbf{p})}(f_i^{(l-1)}(x)) + b_{j,N(\mathbf{p})}^{(l)} \right) \nabla_{\theta} \sum_{i=1}^{n_l} \left(\mathcal{W}_{ij}^{(l+1)} * \phi_{N(\mathbf{q})}(f_i^{(l-1)}(x')) + b_{j,N(\mathbf{q})}^{(l)} \right) \\
&= \nabla_{\theta}^T \sum_{i=1}^{n_l} \sum_{\mathbf{p}' \in N(\mathbf{p})} \mathcal{W}_{ij,\mathbf{p}'-\mathbf{p}}^{(l)} \times \phi_{\mathbf{p}'}(f_i^{(l-1)}(x)) \nabla_{\theta} \sum_{i=1}^{n_l} \sum_{\mathbf{q}' \in N(\mathbf{q})} \mathcal{W}_{ij,\mathbf{q}'-\mathbf{q}}^{(l)} \times \phi_{\mathbf{q}'}(f_i^{(l-1)}(x')) + \sigma_b^2 \\
&= \frac{1}{n_l} \sum_{i=1}^{n_l} (w_{ij}^{(l+1)})^2 \sum_{\mathbf{p}' \in N(\mathbf{p})} \sum_{\mathbf{q}' \in N(\mathbf{q})} \nabla_{\theta}^T f_{i,\mathbf{p}'}^{(l-1)}(x) \nabla_{\theta} f_{i,\mathbf{q}'}^{(l-1)}(x') \dot{\phi}_{\mathbf{p}'}(f_i^{(l-1)}(x)) \dot{\phi}_{\mathbf{q}'}(f_i^{(l-1)}(x')) \\
&\quad + \frac{1}{n_l} \sum_{i=1}^{n_l} \sum_{\mathbf{p}' \in N(\mathbf{p})} \sum_{\mathbf{q}' \in N(\mathbf{q})} \phi_{\mathbf{p}'}(f_i^{(l-1)}(x)) \phi_{\mathbf{q}'}(f_i^{(l-1)}(x')) + \sigma_b^2
\end{aligned} \tag{A-1}$$

In the calculation process of the tangent kernel, the presence of the bias term introduces a shift in each recursion step. Therefore, in many tangent kernel computations, the bias term is excluded. This article follows the same approach that $\sigma_b = 0$. For convenience, we need some symbols to record intermediate quantities. For $l = 0$:

$$\begin{aligned}
\Sigma^{*(0)}(x, x') &= \sum_{i=0}^{n_0} x_i \otimes x'_i \\
\Sigma_{\mathbf{p},\mathbf{q}}^{(0)}(x, x') &= \text{tr} \left(\Sigma_{N(\mathbf{p}),N(\mathbf{q})}^{*(0)}(x, x') \right).
\end{aligned} \tag{A-2}$$

in which \otimes denotes tensor product. For $1 \leq l \leq L-1$, define the covariance matrix, the 0th-order gradient term, and the 1st-order gradient term in sequence according to the definition order in NTK.

$$\begin{aligned}
\text{covariance matrix : } \Lambda_{\mathbf{p},\mathbf{q}}^{(l)}(x, x') &= \begin{pmatrix} \Sigma_{\mathbf{p},\mathbf{q}}^{(l-1)}(x, x) & \Sigma_{\mathbf{p},\mathbf{q}}^{(l-1)}(x, x') \\ \Sigma_{\mathbf{p},\mathbf{q}}^{(l-1)}(x', x) & \Sigma_{\mathbf{p},\mathbf{q}}^{(l-1)}(x', x') \end{pmatrix} \in \mathbb{R}^{2 \times 2}. \\
0\text{th-order term : } \Sigma_{\mathbf{p},\mathbf{q}}^{*(l)}(x, x') &= \mathbb{E}_{(u,v) \sim \mathcal{N}(\mathbf{0}, \Lambda_{\mathbf{p},\mathbf{q}}^{(l)}(x, x'))} [\phi(u)\phi(v)]. \\
1\text{th-order term : } \dot{\Sigma}_{\mathbf{p},\mathbf{q}}^{*(l)}(x, x') &= \mathbb{E}_{(u,v) \sim \mathcal{N}(\mathbf{0}, \Lambda_{\mathbf{p},\mathbf{q}}^{(l)}(x, x'))} [\dot{\phi}(u)\dot{\phi}(v)].
\end{aligned} \tag{A-3}$$

And the induction:

$$\Sigma_{\mathbf{p},\mathbf{q}}^{(l)}(x, x') = \text{tr} \left(\Sigma_{N(\mathbf{p}),N(\mathbf{q})}^{*(l)}(x, x') \right). \tag{A-4}$$

Now we can present the recursive formula for the kernel as follows:

$$\begin{aligned}
K_{\mathbf{p},\mathbf{q}}^{(l)}(x, x') &= \frac{1}{n_l} \sum_{i=1}^{n_l} (w_{ij}^{(l+1)})^2 \sum_{\mathbf{p}' \in N(\mathbf{p})} \sum_{\mathbf{q}' \in N(\mathbf{q})} \nabla_{\theta}^T f_{i,\mathbf{p}'}^{(l-1)}(x) \nabla_{\theta} f_{i,\mathbf{q}'}^{(l-1)}(x') \dot{\phi}_{\mathbf{p}'}(f_i^{(l-1)}(x)) \dot{\phi}_{\mathbf{q}'}(f_i^{(l-1)}(x')) \\
&\quad + \frac{1}{n_l} \sum_{i=1}^{n_l} \sum_{\mathbf{p}' \in N(\mathbf{p})} \sum_{\mathbf{q}' \in N(\mathbf{q})} \phi_{\mathbf{p}'}(f_i^{(l-1)}(x)) \phi_{\mathbf{q}'}(f_i^{(l-1)}(x')) + \sigma_b^2 \\
&= K_{N(\mathbf{p}),N(\mathbf{q})}^{(l-1)} * \dot{\Sigma}_{N(\mathbf{p}),N(\mathbf{q})}^{*(l)}(x, x') + \Sigma_{\mathbf{p},\mathbf{q}}^{(l)}(x, x').
\end{aligned} \tag{A-5}$$

Now, we consider a neural network f_{wop}, f_{wp} with a head (Assume that the final layer has only one output; the case with

multiple outputs is analogous) as follows:

14

$$\begin{aligned}
K_{wop}(x, x') &= \nabla_{\theta}^T f_{wop}(x) \nabla_{\theta} f_{wop}(x') \\
&= \nabla_{\theta}^T \mathcal{P}_{av} \left(\sum_{i=1}^{n_{L-1}} \mathcal{W}_i^{(L)} * \mathbf{f}(x) \right) \nabla_{\theta} \mathcal{P}_{av} \left(\sum_{i=1}^{n_{L-1}} \mathcal{W}_i^{(L)} * \mathbf{f}(x') \right) \\
&= \frac{1}{(\prod_{i=1}^d h_i)^2} \sum_{\mathbf{p} \in [h_1, h_2, \dots, h_d]} K_{\mathbf{p}}^{(L)}(x, x') \\
&= \text{tr} \left(K^{(L)}(x, x') \right)
\end{aligned} \tag{A-6}$$

15

$$\begin{aligned}
K_{wp}(x, x') &= \nabla_{\theta}^T f_{wp}(x) \nabla_{\theta} f_{wp}(x') \\
&= \nabla_{\theta}^T \sum_{i=1}^{n_{L-1}} \mathcal{W}_i^{(L)} \times \mathcal{P}_{av}(\mathbf{f}(x)) \nabla_{\theta} \sum_{i=1}^{n_{L-1}} \mathcal{W}_i^{(L)} \times \mathcal{P}_{av}(\mathbf{f}(x')) \\
&= \frac{1}{n_{L-1}} \sum_{i=1}^{n_{L-1}} \mathcal{P}_{av}(\mathbf{f}(x)) \times \mathcal{P}_{av}(\mathbf{f}(x')) + \\
&\quad \frac{1}{n_{L-1}} \sum_{i=1}^{n_{L-1}} (w_i^{(L)})^2 \left(\frac{1}{\prod_{i=1}^d h_i} \sum_{\mathbf{p} \in [h_1, h_2, \dots, h_d]} \sum_{\mathbf{q} \in [h_1, h_2, \dots, h_d]} \nabla_{\theta}^T \mathbf{f}_{i,\mathbf{p}}(x) \nabla_{\theta} \mathbf{f}_{i,\mathbf{q}}(x') \right) \\
&= \mathcal{P}_{av} \left(K^{(L)}(x, x') \right) + \mathcal{P}_{av} \left(\Sigma_{\mathbf{p},\mathbf{q}}^{*(L)}(x, x') \right)
\end{aligned} \tag{A-7}$$

If we fix the last layer, the upper equation can be reduced to our proposition:

16

$$K_{wp}(x, x') = \mathcal{P}_{av} \left(K^{(L)}(x, x') \right) \tag{A-8}$$

The calculation of the composite-NTK only requires knowledge of the propagation of NTK at the pooling layer, K_{wp} , and the propagation of covariance, which necessitates computing $\text{Cov}(f(x), f(x'))$:

17

18

$$\begin{aligned}
\text{Cov} \left(P_{av}^{d_1 \rightarrow d_2}(\mathbf{f}(x)), P_{av}^{d_1 \rightarrow d_2}(\mathbf{f}(x')) \right) &= \frac{1}{\prod_{i=d_2+1}^{d_1} h_i} \sum_{i_{d_2+1}, i_{d_2+2}, \dots, i_{d_1}} \text{Cov} \left(\mathbf{f}_{\dots, i_{d_2+1}, i_{d_2+2}, \dots, i_{d_1}}(x), \mathbf{f}_{\dots, i_{d_2+1}, i_{d_2+2}, \dots, i_{d_1}}(x') \right) \\
&= P_{av}^{d_1 \rightarrow d_2}(\Sigma(x, x'))
\end{aligned} \tag{A-9}$$

When $d_2 = 0$, it corresponds to the common average pooling layer, and the covariance tensor is:

19

$$\Sigma^{(L+1)}(x, x') = \mathcal{P}_{av} \left(\Sigma^{*(L)}(x, x') \right) \tag{A-10}$$