Non-Gaussian Tensor Programs

NeurIPS'2022

Eugene Golikov[*]   Greg Yang[†]

[*] École Polytechnique Fédérale de Lausanne, Switzerland
[†] Microsoft research, USA

We are given:

- L matrices $A^1, \dots, A^L \in \mathbb{R}^{n \times n}$;
- M_0 initial vectors $g^1, \dots, g^{M_0} \in \mathbb{R}^n$;
- M_0 initial scalars $c^1, \dots, c^{M_0} \in \mathbb{R}$.

A **tensor program** generates new vectors and scalars iteratively:

$$g^i_\alpha \leftarrow \sum_{\beta=1}^{n} W^i_{\alpha\beta} x^i_\beta, \quad c^i \leftarrow \frac{1}{n} \sum_{\beta=1}^{n} x^i_\beta, \quad \text{where } x^i_\alpha = \phi^i(g^1_\alpha, \dots, g^{i-1}_\alpha; c^1, \dots, c^{i-1}), \tag{1}$$

and

- ϕ^i is a scalar function;
- $W^i = A^j$ or $W^i = A^{j\top}$ for some $j \in [L]$.

Tensor programs can express

1. A **forward pass** for a neural network of any architecture [Yang, 2019];

Tensor programs can express

1. A **forward pass** for a neural network of any architecture [Yang, 2019];
2. A **backward pass** for a neural network of any architecture [Yang, 2020a];

Tensor programs can express

1. A **forward pass** for a neural network of any architecture [Yang, 2019];
2. A **backward pass** for a neural network of any architecture [Yang, 2020a];
3. **Any number of gradient descent steps** [Yang and Littwin, 2021, Yang and Hu, 2021]:

Tensor programs can express

1. A **forward pass** for a neural network of any architecture [Yang, 2019];
2. A **backward pass** for a neural network of any architecture [Yang, 2020a];
3. **Any number of gradient descent steps** [Yang and Littwin, 2021, Yang and Hu, 2021]:

Proposition

Let the network have k outputs. For each t and input ξ, the network output after t GD steps $f_t(\xi)$ can be expressed as a set of k scalars c in some tensor program.

Theorem (Gaussian Master Theorem, [Yang, 2020b])
Consider a Tensor Program with M vectors $g^1, \dots, g^M \in \mathbb{R}^n$ and scalars $c^1, \dots, c^M$. Suppose

1. *All initial vectors $g^1, \dots, g^{M_0}$ have iid entries from $\mathcal{N}(0, 1)$;*
2. *All matrices A^i have iid entries from $\mathcal{N}(0, n^{-1})$;*
3. *All the nonlinearities ϕ^i are pseudo-Lipschitz[1];*
4. *All initial scalars $c^1, \dots, c^{M_0}$ have almost sure limits as $n \to \infty$.*

Then, as $n \to \infty$, for any $i \in [M]$,

$$c^i \xrightarrow{a.s.} \mathring{c}^i, \tag{2}$$

where $\mathring{c}^i$ is a deterministic scalar given by a certain recurrent formula.

[1] A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called pseudo-Lipschitz if there exist $C, d > 0$ such that for any $x, y \in \mathbb{R}^n$, $\|f(x) - f(y)\| \leq C\|x - y\|(1 + \|x\|^d + \|y\|^d)$.

Theorem (Non-Gaussian Master Theorem, ours)
Consider a Tensor Program with M vectors $g^1, \dots, g^M \in \mathbb{R}^n$ and scalars $c^1, \dots, c^M$. Suppose

1. *All initial vectors $g^1, \dots, g^{M_0}$ have iid entries from $\mathcal{N}(0, 1)$;*
2. *All matrices A^i have iid entries with zero mean, variance n^{-1}, and each k-th moment bounded by $\nu_k n^{-k/2}$;*
3. *All the nonlinearities ϕ^i are polynomially smooth*[2]*;*
4. *All initial scalars $c^1, \dots, c^{M_0}$ have almost sure limits as $n \to \infty$ and all moments.*

Then, as $n \to \infty$, for any $i \in [M]$,

$$c^i \xrightarrow{\text{a.s. \& } L^p} \mathring{c}^i \quad \forall p \in [1, \infty) \tag{3}$$

for the same $\mathring{c}^i$ as in the Gaussian theorem.

[2] We call f polynomially smooth if it is smooth and each derivative of order $k \geq 0$ is polynomially bounded.

Applications of Master theorem:

1. **NNGP correspondence**: Each pre-activation output of a neural network converges to a Gaussian process as width tends to infinity.[3]
2. **Convergence to a kernel method**: Under certain parameterization, SGD training dynamics converges to the training dynamics of a kernel method as width tends to infinity.[4]
3. **Random matrix theory**: Semi-circle and Marchenko-Pastur laws.
4. **Free Independence Principle**: at initialization, neural network's weights become freely independent from its hidden representations as width goes to infinity.[5]
5. **Hyperparameter transfer**: optimal training hyperparameters can be transfered from thin to wide nets under certain parameterization.[6]

[3] [Neal, 1995, Lee et al., 2017, Garriga-Alonso et al., 2018, Novak et al., 2018, Yang, 2019]
[4] [Jacot et al., 2018, Lee et al., 2019, Yang, 2020a, Yang and Littwin, 2021]
[5] [Yang, 2020b]
[6] [Yang and Hu, 2021, Yang et al., 2022]

Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. (2018).
Deep convolutional networks as shallow gaussian processes.
arXiv preprint arXiv:1808.05587.

Jacot, A., Gabriel, F., and Hongler, C. (2018).
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in neural information processing systems*, pages 8571–8580.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. (2017).
Deep neural networks as gaussian processes.
arXiv preprint arXiv:1711.00165.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019).
Wide neural networks of any depth evolve as linear models under gradient descent.
In *Advances in neural information processing systems*, pages 8572–8583.

Neal, R. M. (1995).

BAYESIAN LEARNING FOR NEURAL NETWORKS.
PhD thesis, University of Toronto.

Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018).
Bayesian deep convolutional networks with many channels are gaussian processes.
arXiv preprint arXiv:1810.05148.

Yang, G. (2019).
Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are gaussian processes.
arXiv preprint arXiv:1910.12478.

Yang, G. (2020a).
Tensor programs ii: Neural tangent kernel for any architecture.
arXiv preprint arXiv:2006.14548.

Yang, G. (2020b).
Tensor programs iii: Neural matrix laws.

arXiv preprint arXiv:2009.10685.

Yang, G. and Hu, E. J. (2021).
Tensor programs iv: Feature learning in infinite-width neural networks.
In *International Conference on Machine Learning*, pages 11727–11737. PMLR.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. (2022).
Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer.
arXiv preprint arXiv:2203.03466.

Yang, G. and Littwin, E. (2021).
Tensor programs iib: Architectural universality of neural tangent kernel training dynamics.
In *International Conference on Machine Learning*, pages 11762–11772. PMLR.