Can We Use Ensemble Uncertainty in the Infinite Width Limit?

Lisa Schut[1], Edward Hu[2], Greg Yang[2*], Yarin Gal[1*]

[1]OATML, University of Oxford, [2]Microsoft Research, *Equal Supervision

schut@robots.ox.ac.uk

TL ; DR

Table 1: Do the parametrizations allow for feature learning and uncertainty (via ensembling) in the **infinite width limit?**

Parametrization	Feature Learning	Uncertainty
SP (aka *Kaiming normal* [1])	×	√
μP [2]	√	×
OURS	√	√

Key Definitions

The parametrization (incl. initialization scheme and learning rate) determines whether we can achieve feature learning and our learnt functions is deterministic. We define **feature learning** as

Definition 1. *Let x_l be the network features, i.e. pre-activation layer outputs. Then, feature learning occurs if x_l have an update of $\Theta(1)$,*

where a vector v is $O(n^a)$ iff $\sqrt{||v||^2/n}$ fluctuates on the order of $O(n^a)$, where n is the number of units in a hidden layer.

A function f is **deterministic** iff

Definition 2. $\lim_{n\to\infty}$ *var*$(f_t) \to 0$, *where n is the number of units in a hidden layer.*

Further, abc-parametrization [2] allows us to create an effective per-layer learning rate. We adapt the definition from [2], and define **abc-parametrization** as

Definition 3. *Let W^l be a weight matrix in a L-layer network. Then, $W^l := n^{-a_l} w_l$, where $w_l \sim N(0, n^{-2b_l})$ is a trainable parameter. The third parameter c_l is the learning rate, defined as γn^{-c_l}, where γ is a constant.*

Methods

Table 2: abc-parametrization of standard parametrization (SP), maximal update parametrization (μP) and our parametrization.

	a_l	b_l	c_l
SP	0	$\begin{cases} 0, l=1 \\ \frac{1}{2}, l\geq 2 \end{cases}$	1
μP	$\begin{cases} -\frac{1}{2}, l=1, \\ 0, 2\leq l\leq L, \\ \frac{1}{2}, l=L+1 \end{cases}$	$\frac{1}{2}$	0
Ours	$\begin{cases} -\frac{1}{2}, l=1 \\ 0, l\geq 2 \end{cases}$	$\frac{1}{2}$	$\begin{cases} 0, l\leq L, \\ 1, l=L+1 \end{cases}$

To prevent a layer output from blowing up, parametrizations downscale weights (as in μP) or learning rates (as in SP). Consequentially, **we either do not permit feature learning or learn a deterministic function** (and thereby forego uncertainty via ensembling), as summarized in Table 1.

We propose an alternative parametrization that is able to capture feature learning and avoids learning a deterministic function. Specifically,

- in general, use μP to ensure **maximal feature and function updates** during training,
- contrary to μP, do not downscale the weights in the final layer (i.e., use $a_{L+1} = 1/2$), to **avoid learning a deterministic function**,
- modify the backward pass: set W_t by $\Delta W_t = W_t - W_0$, and
- use a learning rate of γn^{-1} for the final layer.

The last two alterations prevent the network from blowing up during training.

Results

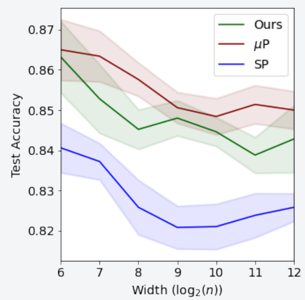

Figure 1: **Feature learning**: logistic regression performance using the top 15 principal components of features. 95% confidence intervals is over 10 seeds.

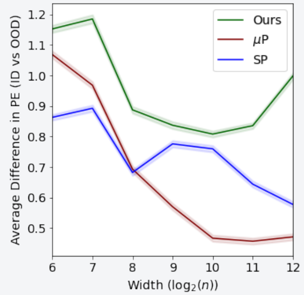

Figure 2: **Uncertainty Estimation**. Difference in predictive entropy for in-distribution data (MNIST) vs out-of-distribution data (FashionMNIST).

Our preliminary results suggest our parametrization

- **permits feature learning** as width increases, comparatively well to μPand contrary to SP which observes a dip in performance.
- **is able to obtain better uncertainty** estimation via ensembling than the other parametrizations.

References

[1] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision.* 2015.

[2] Greg Yang and Edward J. Hu. “Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks”. In: *Proceedings of the 38th International Conference on Machine Learning.* 2021.