A Appendix

A.1 Basic definitions

In this study, we first consider the neural network with 2 hidden layers,

A two-layer NN is

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{m} a_j \sigma(\boldsymbol{w}_j \cdot \boldsymbol{x}), \tag{11}$$

where $\sigma(\cdot)$ is the activation function, $\boldsymbol{w}_j = (\bar{\boldsymbol{w}}_j, \boldsymbol{b}_j) \in \mathbb{R}^{d+1}$ is the neuron feature including the input weight and bias terms, and $\boldsymbol{x} = (\bar{\boldsymbol{x}}, 1) \in \mathbb{R}^{d+1}$ is concatenation of the input sample and scalar 1, $\boldsymbol{\theta}$ is the set of all parameters, i.e., $\{a_j, \boldsymbol{w}_j\}_{j=1}^m$. For simplicity, **we call $\boldsymbol{w}_j$ input weight or weight** and $\boldsymbol{x}$ input sample.

Then, we consider the neural network with l hidden layers,

$$\begin{gathered} \boldsymbol{x}^{[0]} = (\boldsymbol{x}, 1), \quad \boldsymbol{x}^{[1]} = (\sigma(\boldsymbol{W}^{[1]}\boldsymbol{x}^{[0]}), 1), \quad \boldsymbol{x}^{[l]} = (\sigma(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l-1]}), 1), \text{ for } l \in \{2, 3, ..., L\} \\ f(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{a}^\intercal \boldsymbol{x}^{[L]} \triangleq f_{\boldsymbol{\theta}}(\boldsymbol{x}), \end{gathered} \tag{12}$$

where $\boldsymbol{W}^{[l]} = (\bar{\boldsymbol{W}}^{[l]}, \boldsymbol{b}^{[l]}) \in \mathbb{R}^{(m_l \times m_{l-1})}$, and m_l represents the dimension of the l-th hidden layer. The initialization of $\boldsymbol{W}_{k,k'}^{[l]}, l \in \{1, 2, 3, ..., L\}$ and $\boldsymbol{a}_k$ obey normal distribution $\boldsymbol{W}_{k,k'}^{[l]} \sim \mathcal{N}(0, \beta_l^2)$ for $l \in \{1, 2, 3, ..., L\}$ and $\boldsymbol{a}_k \sim \mathcal{N}(0, \beta_{L+1}^2)$.

The loss function is mean squared error given below,

$$R(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2. \tag{13}$$

For simplification, we denote $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ as f in following.

A.2 Derivations for concerned quantities

A.2.1 Neural networks with three hidden layers

In order to better understand the gradient of the parameter matrix of the multi-layer neural network, we first consider the case of the three-layer neural network,

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) := \boldsymbol{a}^\intercal \sigma(\boldsymbol{W}^{[2]} \sigma(\boldsymbol{W}^{[1]} \boldsymbol{x})), \tag{14}$$

with the mean squared error as the loss function,

$$R_s(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2. \tag{15}$$

We calculate $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[2]}}$ and $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[1]}}$ respectively, using differential form,

$$\mathrm{d}f = \mathrm{tr}((\frac{\partial f}{\partial \boldsymbol{x}})^\intercal \mathrm{d}f). \tag{16}$$

We consider $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[2]}}$ first,

$$\begin{aligned} \mathrm{d}f &= \mathrm{tr}\{\mathrm{d}(\boldsymbol{a}^\intercal \sigma(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]}))\} \\ &= \mathrm{tr}\{\boldsymbol{a}^\intercal \mathrm{d}(\sigma(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]}))\} \\ &= \mathrm{tr}\{\boldsymbol{a}^\intercal \sigma'(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]}) \odot \mathrm{d}\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]}\} \\ &= \mathrm{tr}\{(\boldsymbol{a} \odot \sigma'(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]})^\intercal \mathrm{d}\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]}\} \\ &= \mathrm{tr}\{\boldsymbol{x}^{[1]} (\boldsymbol{a} \odot \sigma'(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]})^\intercal \mathrm{d}\boldsymbol{W}^{[2]}\} \\ &= \mathrm{tr}\{((\boldsymbol{a} \odot \sigma'(\boldsymbol{W}^{[2]} \boldsymbol{x}^{[1]})) \boldsymbol{x}^{[1]\intercal})^\intercal \mathrm{d}\boldsymbol{W}^{[2]}\}, \end{aligned} \tag{17}$$

where $\odot$ is Hadamard product, and it is the multiplication of matrix elements of the same position. Hence,

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[2]}} &= (\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]}))\boldsymbol{x}^{[1]\intercal}\\ &= \mathrm{diag}\{\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]})\}\boldsymbol{a}\boldsymbol{x}^{[1]\intercal}.\end{aligned} \tag{18}$$

Then, we consider $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[1]}}$,

$$\begin{aligned}\mathrm{d}f &= \mathrm{tr}\{(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]}))^\intercal\boldsymbol{W}^{[2]}\mathrm{d}\sigma(\boldsymbol{W}^{[1]}\boldsymbol{x})\}\\ &= \mathrm{tr}\{(\boldsymbol{W}^{[2]\intercal}(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]})))^\intercal\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x})\odot\mathrm{d}(\boldsymbol{W}^{[1]}\boldsymbol{x})\}\\ &= \mathrm{tr}\{((\boldsymbol{W}^{[2]\intercal}(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]}))\odot\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x}))^\intercal\mathrm{d}(\boldsymbol{W}^{[1]}\boldsymbol{x})\}\\ &= \mathrm{tr}\{[((\boldsymbol{W}^{[2]\intercal}(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]}))\odot\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x}))\boldsymbol{x}^\intercal]^\intercal\mathrm{d}(\boldsymbol{W}^{[1]})\}.\end{aligned} \tag{19}$$

Hence, we have,

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[1]}} &= ((\boldsymbol{W}^{[2]\intercal}(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]})))\odot\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x}))\boldsymbol{x}^\intercal\\ &= \mathrm{diag}\{\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x})\}\boldsymbol{W}^{[2]\intercal}\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}^{[1]})\}\boldsymbol{a}\boldsymbol{x}^\intercal.\end{aligned} \tag{20}$$

Through the chain rule, we can get the evolution equation of $\boldsymbol{W}^{[1]}$ and $\boldsymbol{W}^{[2]}$,

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{W}^{[1]}}{\mathrm{d}t} &= -\frac{\mathrm{d}R_s(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{W}^{[1]}}\\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta},\boldsymbol{x}_i)-y_i)\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[1]}}\\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta},\boldsymbol{x}_i)-y_i)\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[1]}\boldsymbol{x}_i)\}\boldsymbol{W}^{[2]\intercal}\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}_i^{[1]})\}\boldsymbol{a}\boldsymbol{x}_i^\intercal,\end{aligned} \tag{21}$$

and

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{W}^{[2]}}{\mathrm{d}t} &= -\frac{\mathrm{d}R_s(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{W}^{[2]}}\\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta},\boldsymbol{x}_i)-y_i)\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[1]}}\\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta},\boldsymbol{x}_i)-y_i)\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[2]}\boldsymbol{x}_i^{[1]})\}\boldsymbol{a}\boldsymbol{x}_i^{[1]\intercal}.\end{aligned} \tag{22}$$

A.2.2 L hidden layers condition

And, we consider the neural network with L hidden layers,

$$\begin{aligned}\mathrm{d}f &= \mathrm{tr}\{\mathrm{d}\boldsymbol{a}^\intercal\mathrm{d}\sigma(\boldsymbol{W}^{[L]}\boldsymbol{x}^{[L-1]})\}\\ &= \mathrm{tr}\{(\boldsymbol{a}\odot\sigma'(\boldsymbol{W}^{[L]}\boldsymbol{x}^{[L-1]}))^\intercal\mathrm{d}\boldsymbol{W}^{[L]}\sigma(\boldsymbol{W}^{[L-1]}\boldsymbol{x}^{[L-2]})\}\\ &= \mathrm{tr}\{(\boldsymbol{W}^{[L]\intercal}\Lambda_L)^\intercal\sigma'(\boldsymbol{W}^{[L-1]}\boldsymbol{x}^{[L-2]})\odot\mathrm{d}\boldsymbol{W}^{[L-1]}\sigma(\boldsymbol{W}^{[L-2]}\boldsymbol{x}^{[L-3]})\}\\ &= \mathrm{tr}\{((\boldsymbol{W}^{[L]\intercal}\Lambda_L)\odot\sigma'(\boldsymbol{W}^{[L-1]}\boldsymbol{x}^{[L-2]}))^\intercal\boldsymbol{W}^{[L-1]}\mathrm{d}\sigma(\boldsymbol{W}^{[L-2]}\boldsymbol{x}^{[L-3]})\}\\ &= (\boldsymbol{W}^{[L-1]\intercal}\Lambda_{L-1})^\intercal\mathrm{d}\sigma(\boldsymbol{W}^{[L-2]}\boldsymbol{x}^{[L-3]})\\ &= \dots\\ &= \mathrm{tr}\{\Lambda_k^\intercal\mathrm{d}\boldsymbol{W}^{[k]}\boldsymbol{x}^{[k-1]}\}\\ &= \mathrm{tr}\{(\Lambda_k\boldsymbol{x}^{[k-1]\intercal})^\intercal\mathrm{d}\boldsymbol{W}^{[k]}\},\end{aligned} \tag{23}$$

where $\Lambda_l := (\boldsymbol{W}^{[l+1]\intercal}\Lambda_{l+1}) \odot \sigma'(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l-1]})$ for $l = k, k+1 \dots L-1$ and $\Lambda_L := \boldsymbol{a} \odot \sigma'(\boldsymbol{W}^{[L]}\boldsymbol{x}^{[L-1]})$.

Hence, we get,

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[k]}} = \Lambda_k \boldsymbol{x}^{[k-1]\intercal}. \tag{24}$$

Through the chain rule, we can get the evolution equation of $\boldsymbol{W}^{[k]}$,

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{W}^{[k]}}{\mathrm{d}t} &= -\frac{\mathrm{d}R_s(\boldsymbol{\theta})}{\mathrm{d}\boldsymbol{W}^{[k]}} \\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i)\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{W}^{[k]}} \\ &= -\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i)\Lambda_k \boldsymbol{x}_i^{[k-1]\intercal}.\end{aligned} \tag{25}$$

Through $\boldsymbol{a} \odot \sigma'(\boldsymbol{W}\boldsymbol{x}) = \mathrm{diag}\{\sigma'(\boldsymbol{W}\boldsymbol{x})\}\boldsymbol{a}$,

Finally, the dynamic system can be obtained:

$$\begin{aligned}\dot{\boldsymbol{a}} = \frac{\mathrm{d}\boldsymbol{a}}{\mathrm{d}t} &= -\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^{[L]}\left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right), \\ \dot{\boldsymbol{W}}^{[L]} = \frac{\mathrm{d}\boldsymbol{W}^{[L]}}{\mathrm{d}t} &= -\frac{1}{n}\sum_{i=1}^{n}\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[L]}\boldsymbol{x}_i^{[L-1]})\}\boldsymbol{a}\boldsymbol{x}_i^{[L-1]\intercal}\left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right), \\ \dot{\boldsymbol{W}}^{[k]} = \frac{\mathrm{d}\boldsymbol{W}^{[k]}}{\mathrm{d}t} &= -\frac{1}{n}\sum_{i=1}^{n}\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})\}\boldsymbol{E}^{[k+1:L]}\boldsymbol{a}\boldsymbol{x}_i^{[k-1]\intercal}\left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right)\ \forall i \in [1 : L-1],\end{aligned} \tag{26}$$

where we use $\boldsymbol{E}^l(\boldsymbol{x}) = \boldsymbol{W}^{[l]\intercal}\,\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l-1]})\}$. And $\boldsymbol{E}^{[q:p]} = \boldsymbol{E}^q\boldsymbol{E}^{q+1}...\boldsymbol{E}^p$.

Let $r_{k,j} = \|\boldsymbol{W}_j^{[k]}\|_2$. We have

$$\frac{\mathrm{d}}{\mathrm{d}t}|r_{k,j}|^2 = \frac{\mathrm{d}}{\mathrm{d}t}\|\boldsymbol{W}_j^{[k]}\|^2. \tag{27}$$

Then we obtain

$$\dot{r}_{k,j}r_{k,j} = \dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{W}_j^{[k]}. \tag{28}$$

Finally, we get

$$\begin{aligned}\dot{r}_{k,j} = \frac{\mathrm{d}r_{k,j}}{\mathrm{d}t} &= \dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{W}_j^{[k]}/r_{k,j} \\ &= \dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{u}_{k,j},\end{aligned} \tag{29}$$

where $\boldsymbol{u}_{k,j} = \frac{\boldsymbol{W}_j^{[k]}}{r_{k,j}}$ is a unit vector. Then we have,

$$\begin{aligned}\dot{\boldsymbol{u}}_{k,j} = \frac{\mathrm{d}\boldsymbol{u}_{k,j}}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\boldsymbol{W}_j^{[k]}}{r_{k,j}}\right) \\ &= \frac{\dot{\boldsymbol{W}}_j^{[k]}r_{k,j} - \boldsymbol{W}_j^{[k]}\dot{r}_{k,j}}{r_{k,j}^2} \\ &= \frac{\dot{\boldsymbol{W}}_j^{[k]}r_{k,j} - \boldsymbol{W}_j^{[k]}(\dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{u}_{k,j})}{r_{k,j}^2} \\ &= \frac{\dot{\boldsymbol{W}}_j^{[k]} - \boldsymbol{u}_{k,j}(\dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{u}_{k,j})}{r_{k,j}}.\end{aligned} \tag{30}$$

To conclude, the quantities we concern are summarized as follows,

$$
\begin{cases}
\dot{\boldsymbol{a}} = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i^{[L]} \left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right) & (31)\\
\dot{\boldsymbol{W}}^{[L]} = -\frac{1}{n}\sum_{i=1}^{n} \mathrm{diag}\{\sigma'(\boldsymbol{W}^{[L]}\boldsymbol{x}_i^{[L-1]})\}\boldsymbol{a}\boldsymbol{x}_i^{[L-1]\intercal} \left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right), & (32)\\
\dot{\boldsymbol{W}}^{[k]} = -\frac{1}{n}\sum_{i=1}^{n} \mathrm{diag}\{\sigma'(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})\}\boldsymbol{E}^{[k+1:L]}\boldsymbol{a}\boldsymbol{x}_i^{[k-1]\intercal} \left(f(\boldsymbol{\theta}, \boldsymbol{x}_i) - y_i\right)\ \forall k \in [1:L-1] & (33)\\
\dot{r}_{k,j} = \dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{u}_{k,j} & (34)\\
\dot{\boldsymbol{u}}_{k,j} = \dfrac{\dot{\boldsymbol{W}}_j^{[k]} - \boldsymbol{u}_{k,j}(\dot{\boldsymbol{W}}_j^{[k]} \cdot \boldsymbol{u}_{k,j})}{r_{k,j}}, & (35)
\end{cases}
$$

where we use $\boldsymbol{E}^l(\boldsymbol{x}) = \boldsymbol{W}^{[l]\intercal}\,\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[l]}\boldsymbol{x}^{[l-1]})\}$. And $\boldsymbol{E}^{[q:p]} = \boldsymbol{E}^q\boldsymbol{E}^{q+1}...\boldsymbol{E}^p$.

A.2.3 Prove for $\mathcal{P}w$ in 5

We calculate $\mathcal{P}\boldsymbol{w} \overset{\text{leading order}}{\approx} \mathcal{Q}\boldsymbol{w}$ as following,

$$
\begin{aligned}
\mathcal{P}\boldsymbol{w}\approx\mathcal{Q}\boldsymbol{w} :=& -\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j\\
&+(\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{\sigma'(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j \cdot \boldsymbol{u})\boldsymbol{u}\\
=& -\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!} \odot (\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j\\
&+(\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!} \odot (\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j \cdot \boldsymbol{u})\boldsymbol{u}\\
=& -\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}\,\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j\\
&+(\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}\,\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j \cdot \boldsymbol{u})\boldsymbol{u}\\
=& -\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}]_j\,\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})\\
&+(\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}[\mathrm{diag}\{(\boldsymbol{W}^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1}\}]_j\,\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a}) \cdot \boldsymbol{u})\boldsymbol{u}\\
=& -(\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}(\boldsymbol{W}_j^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1})[\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j\\
&+((\frac{1}{n}\sum_{i=1}^{n} e_i\boldsymbol{x}_i^{[k-1]}(\boldsymbol{W}_j^{[k]}\boldsymbol{x}_i^{[k-1]})^{p-1})[\mathrm{diag}\{\frac{\sigma^{(p)}(\boldsymbol{0})}{(p-1)!}\}(\boldsymbol{E}^{[k+1:L]}\boldsymbol{a})]_j \cdot \boldsymbol{u})\boldsymbol{u},
\end{aligned}
\tag{36}
$$

where $(\cdot)^{p-1}$ and $\sigma^p(\cdot)$ operate on component here.

A.3 The Verification of the initial stage

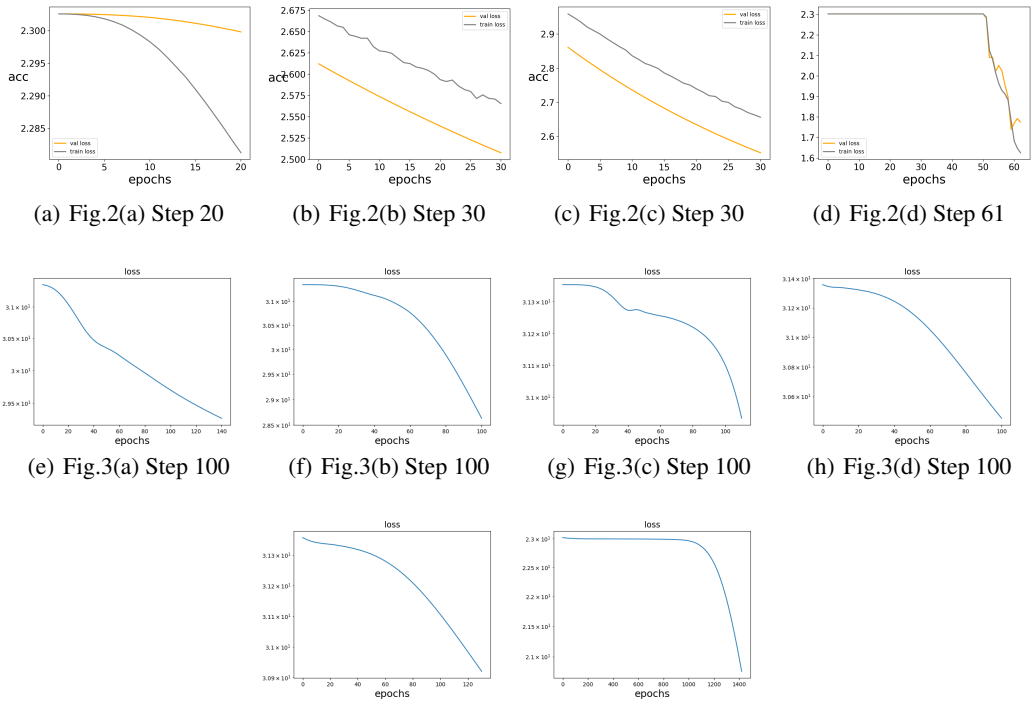
We put the loss of the experiments in the main text here to show that they are indeed in the initial stage of training by the definition.

As is shown in Fig.7 and Fig.8, at the steps demonstrated in the article, loss satisfies the definition of the initial stage, so we consider that they are in the initial stage of training.

Learning rate is not a sensitive to the appearance of condensation. However, a small learning rate could enable us to observe the condensation process in the initial stage under a gradient flow training, more clearly. For example, when the learning rate is relatively small, the initial stage of training may be relatively long, while when the learning rate is relatively large, the initial stage of training may be relatively small.

We empirically find that to ensure the training process follows a gradient follow, where the loss decays monotonically, we have to select a smaller learning rate for large multiplicity p. Therefore, it looks like we have a longer training in our experiments with large p. Note that for a small learning rate in the experiments of small p, we can observe similar phenomena.

In all subsequent experiments in the Appendix, we will no longer show the loss graph of each experiment one by one, but we make sure that they are indeed in the initial stage of training.

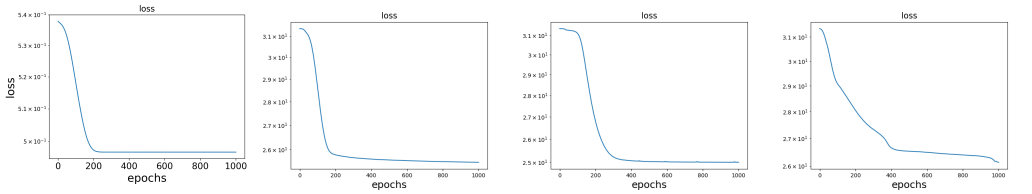


(a) Fig.2(a) Step 20 (b) Fig.2(b) Step 30 (c) Fig.2(c) Step 30 (d) Fig.2(d) Step 61

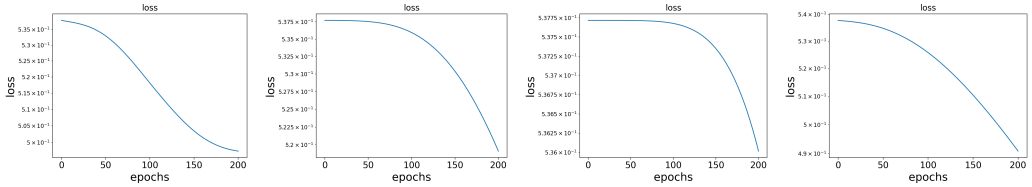
(e) Fig.3(a) Step 100 (f) Fig.3(b) Step 100 (g) Fig.3(c) Step 100 (h) Fig.3(d) Step 100

(i) Fig.3(e) Step 100 (j) Fig.4 Step 900 to 1400

Figure 7: Losses from Fig. 2 to Fig.4. The original figures and the numbers of steps corresponding to each sub-picture are written in the sub-captions.



(a) Fig.5(a) Step 1000 (b) Fig.5(b) Step 1000 (c) Fig.5(c) Step 1000 (d) Fig.5(d) Step 1000

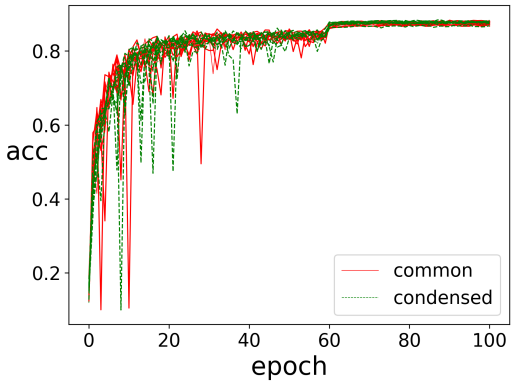


(e) Fig.6(a) Step 200 (f) Fig.6(b) Step 200 (g) Fig.6(c) Step 200 (h) Fig.6(d) Step 200

Figure 8: Losses from Fig. 5 to Fig.6. The original figures and the numbers of steps corresponding to each sub-picture are written in the sub-captions.

A.4 Performance of tanh activation function in condensed regime

For practical networks, such as resnet18-like (He et al., 2016) in learning CIFAR10, as shown in Fig. 9 and Table 1, we find that the performance of networks with initialization in the condensed regime is vary similar to the common initialization methods. For both initialization methods, the test accuracy is about 86.5% to 88.5%, where the highest test accuracy and lowest test accuracy for common methods are 88.07% and 86.73%, respectively, while the highest one and lowest one for condensed methods are 88.26% and 86.88%, respectively. This implies that performance of common and condensed initialization is similar.



(a) test accuracy

Figure 9: The test accuracy of Resnet18-like networks with different initialization methods. Each network consists of the convolution part of resnet18 and fully-connected (FC) layers with size 1024-1024-10 and softmax. The convolution part is equipped with ReLU activation and initialized by Glorot normal distribution (Glorot and Bengio, 2010). For FC layers, the activation is $\tanh(x)$ and they are initialized by three common methods (red) and three condensed ones (green) as indicated in Table 1. The learning rate is 10^{-3} for epoch 1-60 and 10^{-4} for epoch 61-100. Adam optimizer with cross-entropy loss and batch size 128 are used for all experiments.

Table 1: Comparison of test accuracy of resnet18 in learning CIFAR10 with common (Glorot and Bengio, 2010) and condensed Gaussian initializations. $\bar{m} = (m_{\rm in} + m_{\rm out})/2$. $m_{\rm in}$: in-layer width. $m_{\rm out}$: out-layer width. Each line is a trial.

	common			condensed		
	Glorot_uniform	Glorot_normal	$N(0, \frac{1}{\bar{m}})$	$N(0, \frac{1}{m_{\rm out}^4})$	$N(0, \frac{1}{m_{\rm out}^3})$	$N(0, (\frac{1}{\bar{m}})^2)$
Test 1	0.8747	0.8759	0.8807	0.8749	0.8744	0.8765
Test 2	0.8715	0.8673	0.8733	0.8763	0.8799	**0.8826**
Test 3	0.8772	0.8794	0.8788	0.8688	0.8780	0.8771

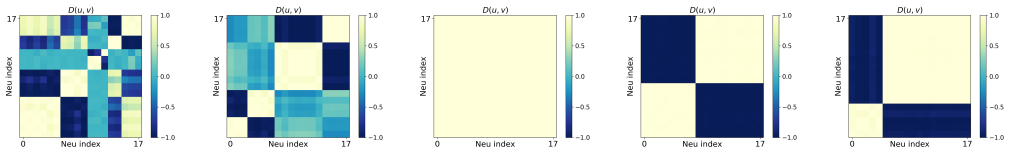
A.5 multi-layer experimental

The condensation of the six layer without residual connections is shown in 10, whose activation functions for hidden layer 1 to hidden layer 5 are $x^2\tanh(x)$, $x\tanh(x)$, $\mathrm{sigmoid}(x)$, $\tanh(x)$ and $\mathrm{softplus}(x)$, respectively.

The condensation of the three layer without residual connections is shown in 11, whose activation functions are same for each layer indicated by the corresponding sub-captions.

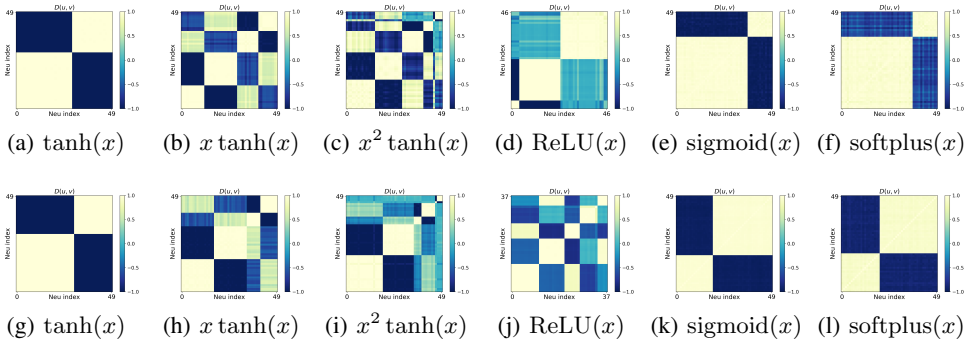
The condensation of the five layer without residual connections is shown in 12, whose activation functions are same for each layer indicated by the corresponding sub-captions.

The condensation of the five layer with residual connections is shown in 13, whose activation functions are same for each layer indicated by the corresponding sub-captions.



(a) layer 1 (b) layer 2 (c) layer 3 (d) layer 4 (e) layer 5

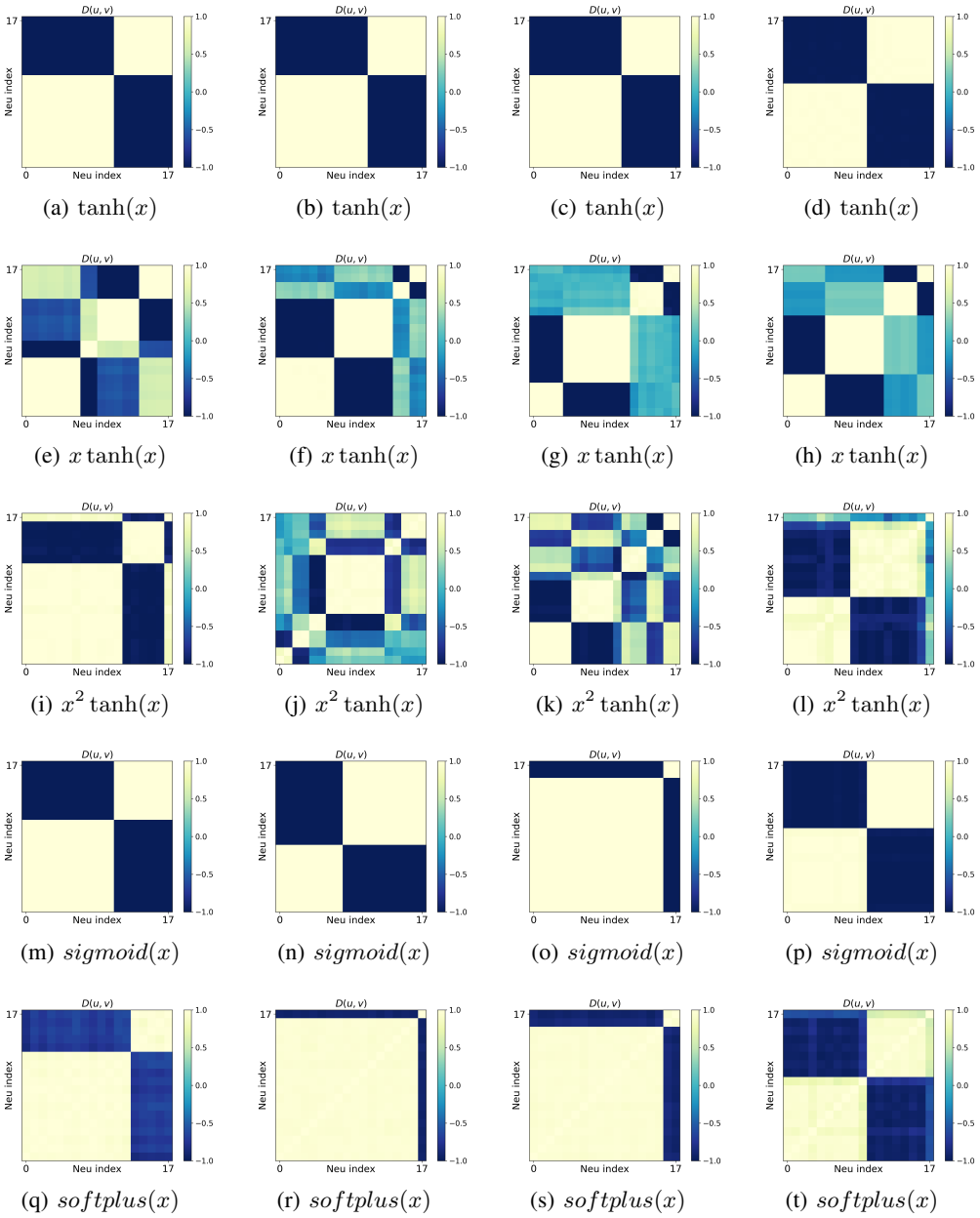
Figure 10: Condensation of six-layer NNs without residual connections. The activation functions for hidden layer 1 to hidden layer 5 are $x^2\tanh(x)$, $x\tanh(x)$, $\mathrm{sigmoid}(x)$, $\tanh(x)$ and $\mathrm{softplus}(x)$, respectively.The numbers of steps selected in the sub-pictures are epoch 6800, epoch 6800, epoch 6800, epoch 6800 and epoch 6300, respectively, while the NN is only trained once. The color indicates $D(\boldsymbol{u},\boldsymbol{v})$ of two hidden neurons' input weights, whose indexes are indicated by the abscissa and the ordinate, respectively. The training data is 80 points sampled from a 3-dimensional function $\sum_{k=1}^{3} 4\sin(12x_k+1)$, where each x_k is uniformly sampled from $[-4,2]$. $n=80$, $d=3$, $m=18$, $d_{\mathrm{out}}=1$, $\mathrm{var}=0.008^2$, $\mathrm{lr}=5\times10^{-5}$.



(a) $\tanh(x)$ (b) $x\tanh(x)$ (c) $x^2\tanh(x)$ (d) $\mathrm{ReLU}(x)$ (e) $\mathrm{sigmoid}(x)$ (f) $\mathrm{softplus}(x)$

(g) $\tanh(x)$ (h) $x\tanh(x)$ (i) $x^2\tanh(x)$ (j) $\mathrm{ReLU}(x)$ (k) $\mathrm{sigmoid}(x)$ (l) $\mathrm{softplus}(x)$

Figure 11: Three-layer NN at epoch 700. (a-f) are for the input weights of the first hidden layer and (g-l) are for the input weights of the second hidden layer. The color indicates $D(\boldsymbol{u},\boldsymbol{v})$ of two hidden neurons' input weights, whose indexes are indicated by the abscissa and the ordinate, respectively. The training data is 80 points sampled from a 5-dimensional function $\sum_{k=1}^{5} 3\sin(8x_k+1)$, where each x_k is uniformly sampled from $[-4,2]$. $n=80$, $d=5$, $m=50$, $d_{\mathrm{out}}=1$, $\mathrm{var}=0.005^2$. $\mathrm{lr}=10^{-4}, 2\times10^{-5}, 1.4\times10^{-5}$ for (a-d), (e) and (f), respectively. For (d) and (j), we discard hidden neurons, whose L_2-norm of its input weight is smaller than 0.1.



(a) $\tanh(x)$ (b) $\tanh(x)$ (c) $\tanh(x)$ (d) $\tanh(x)$

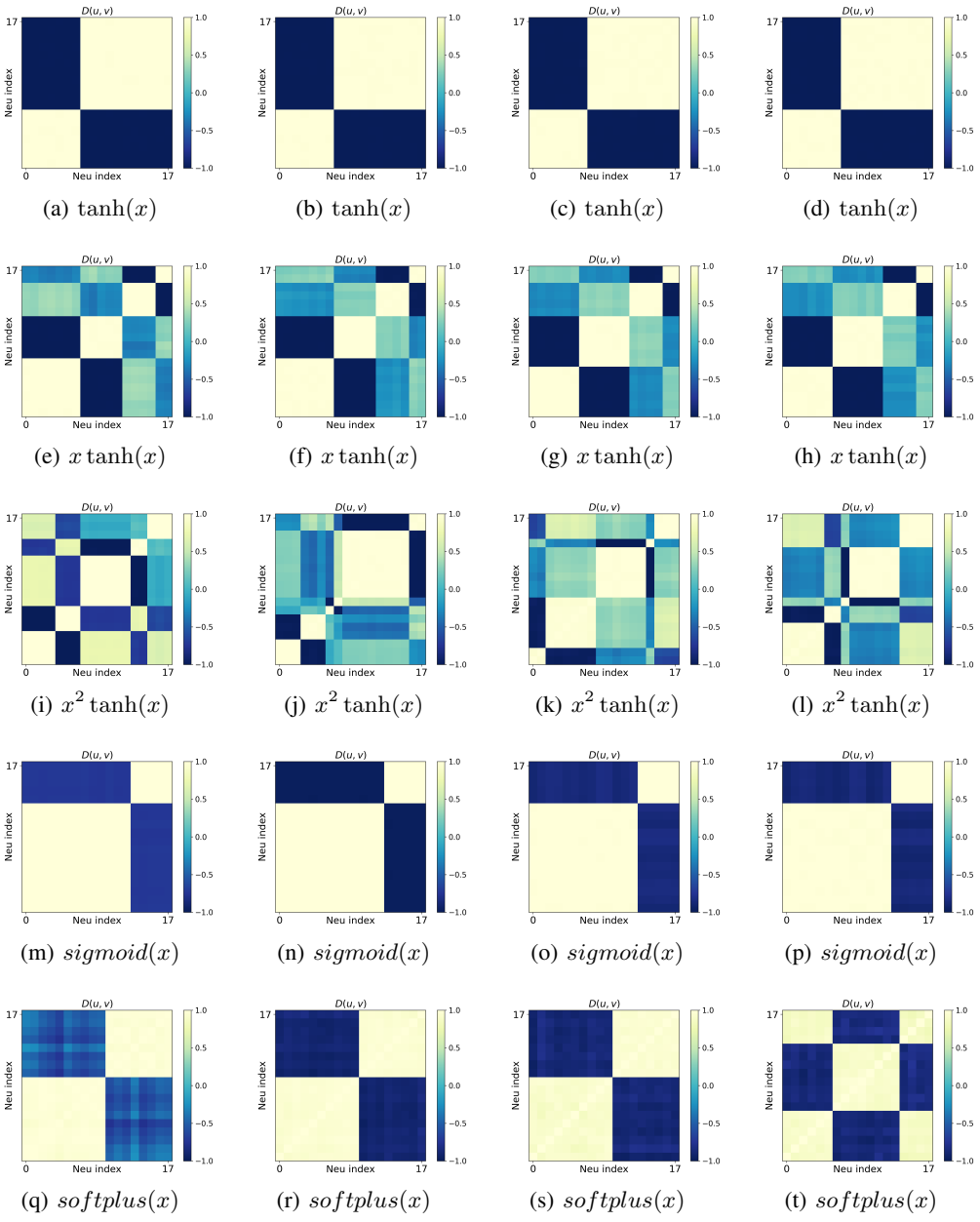
(e) $x\tanh(x)$ (f) $x\tanh(x)$ (g) $x\tanh(x)$ (h) $x\tanh(x)$

(i) $x^2\tanh(x)$ (j) $x^2\tanh(x)$ (k) $x^2\tanh(x)$ (l) $x^2\tanh(x)$

(m) $sigmoid(x)$ (n) $sigmoid(x)$ (o) $sigmoid(x)$ (p) $sigmoid(x)$

(q) $softplus(x)$ (r) $softplus(x)$ (s) $softplus(x)$ (t) $softplus(x)$

Figure 12: Five-layer NN. The first to fourth columns of each row are for the input weights of neurons from the first to the fourth hidden layers, respectively. The color indicates $D(\boldsymbol{u}, \boldsymbol{v})$ of two hidden neurons' input weights, whose indexes are indicated by the abscissa and the ordinate, respectively. The training data is 80 points sampled from a 5-dimensional function $\sum_{k=1}^{3} 3\sin(10x_k + 1)$, where each x_k is uniformly sampled from $[-4, 2]$. $n = 80$, $d = 5$, $m = 18$, $d_{\text{out}} = 1$, $\text{var} = 0.008^2$. $\text{lr} = 1.5 \times 10^{-5}$, 1.5×10^{-5}, 1.5×10^{-5}, 1.5×10^{-5}, 1.5×10^{-6} and epoch is 10000, 10000, 26000, 10000, 20000 for $\tanh(x)$, $x\tanh(x)$, $x^2\tanh(x)$, $sigmoid(x)$, $softplus(x)$, respectively.



(a) $\tanh(x)$ (b) $\tanh(x)$ (c) $\tanh(x)$ (d) $\tanh(x)$

(e) $x\tanh(x)$ (f) $x\tanh(x)$ (g) $x\tanh(x)$ (h) $x\tanh(x)$

(i) $x^2\tanh(x)$ (j) $x^2\tanh(x)$ (k) $x^2\tanh(x)$ (l) $x^2\tanh(x)$

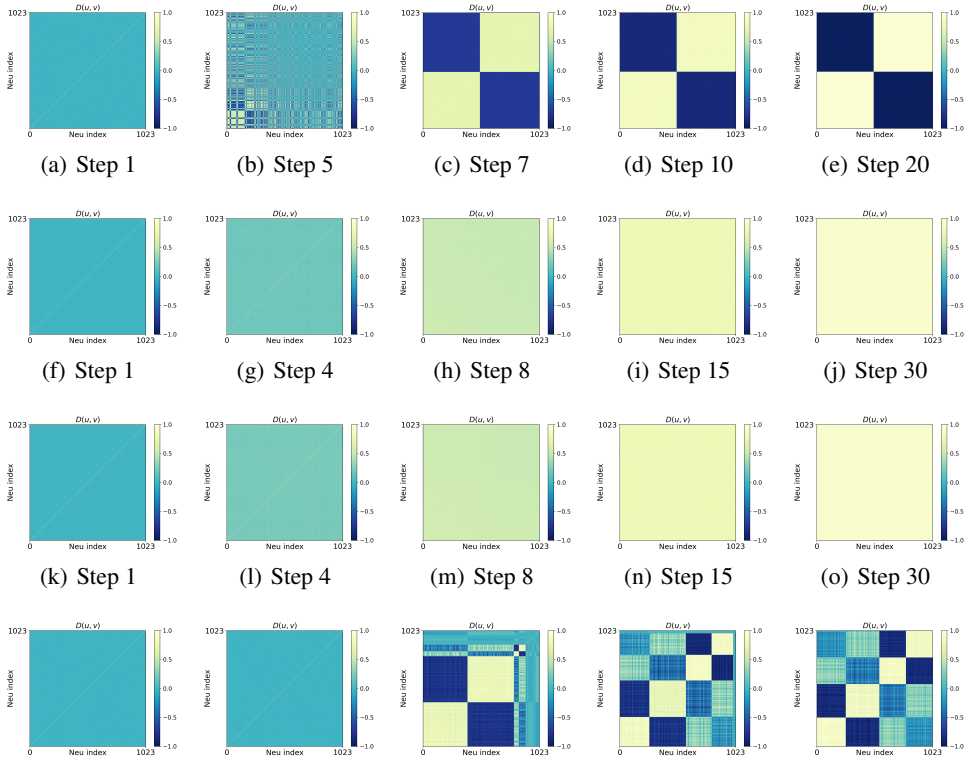
(m) $sigmoid(x)$ (n) $sigmoid(x)$ (o) $sigmoid(x)$ (p) $sigmoid(x)$

(q) $softplus(x)$ (r) $softplus(x)$ (s) $softplus(x)$ (t) $softplus(x)$

Figure 13: Five-layer NN. The first to fourth columns of each row are for the input weights of neurons from the first to the fourth hidden layers, respectively. The color indicates $D(\boldsymbol{u}, \boldsymbol{v})$ of two hidden neurons' input weights, whose indexes are indicated by the abscissa and the ordinate, respectively. The training data is 80 points sampled from a 5-dimensional function $\sum_{k=1}^{3} 3\sin(10x_k + 1)$, where each x_k is uniformly sampled from $[-4, 2]$. $n = 80$, $d = 5$, $m = 18$, $d_{\text{out}} = 1$, var $= 0.008^2$. lr $= 1\times10^{-4}$, 1×10^{-4}, 1×10^{-4}, 5×10^{-5}, 5×10^{-5} and epoch is 400, 400, 400, 3000, 360, 400 for $\tanh(x)$, $x\tanh(x)$, $x^2\tanh(x)$, $x^2 tanh(x)$, $sigmoid(x)$, $softplus(x)$, respectively.

A.6 Several steps during the evolution of condensation at the initial stage

In the article, we only give the results of the last step of each condense, while the details of the evolution of condensation are lacking, which may provide a better understanding. Therefore, we show these details in Fig. 14, Fig. 15, Fig. 16 and Fig. 17, which also further illustrate the rationality of the experimental results and facilitate the understanding of the evolution of condensation in the initial stage.



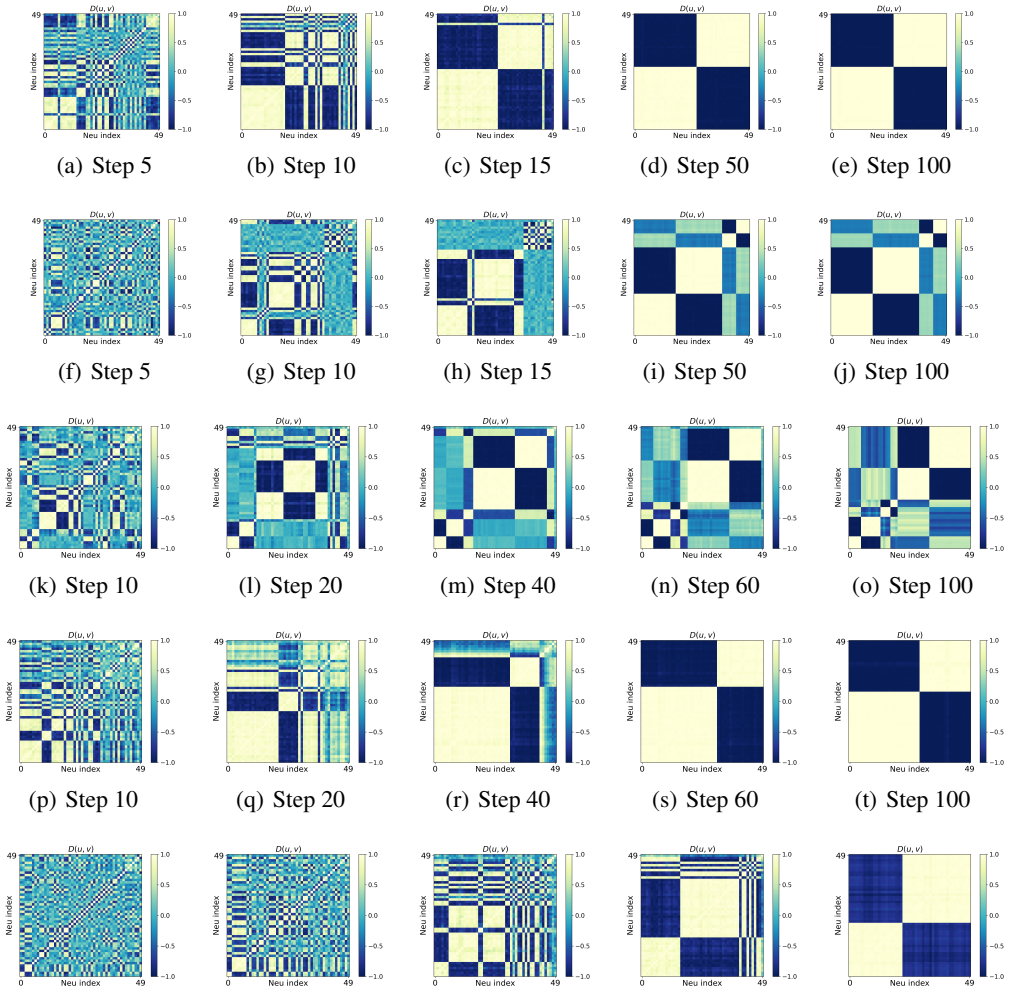
(a) Step 1 (b) Step 5 (c) Step 7 (d) Step 10 (e) Step 20

(f) Step 1 (g) Step 4 (h) Step 8 (i) Step 15 (j) Step 30

(k) Step 1 (l) Step 4 (m) Step 8 (n) Step 15 (o) Step 30

(p) Step 30 (q) Step 51 (r) Step 54 (s) Step 58 (t) Step 61

Figure 14: Evolution of condensation of Fig. 2(a), Fig. 2(b), Fig. 2(c), and Fig. 2(d). The evolution from the first row to the fourth row are corresponding to the Fig. 2(a), Fig. 2(b), Fig. 2(c), and Fig. 2(d). The numbers of evolutionary steps are shown in the sub-captions, where sub-figures in the last row are the epochs in the article.



(a) Step 5 (b) Step 10 (c) Step 15 (d) Step 50 (e) Step 100

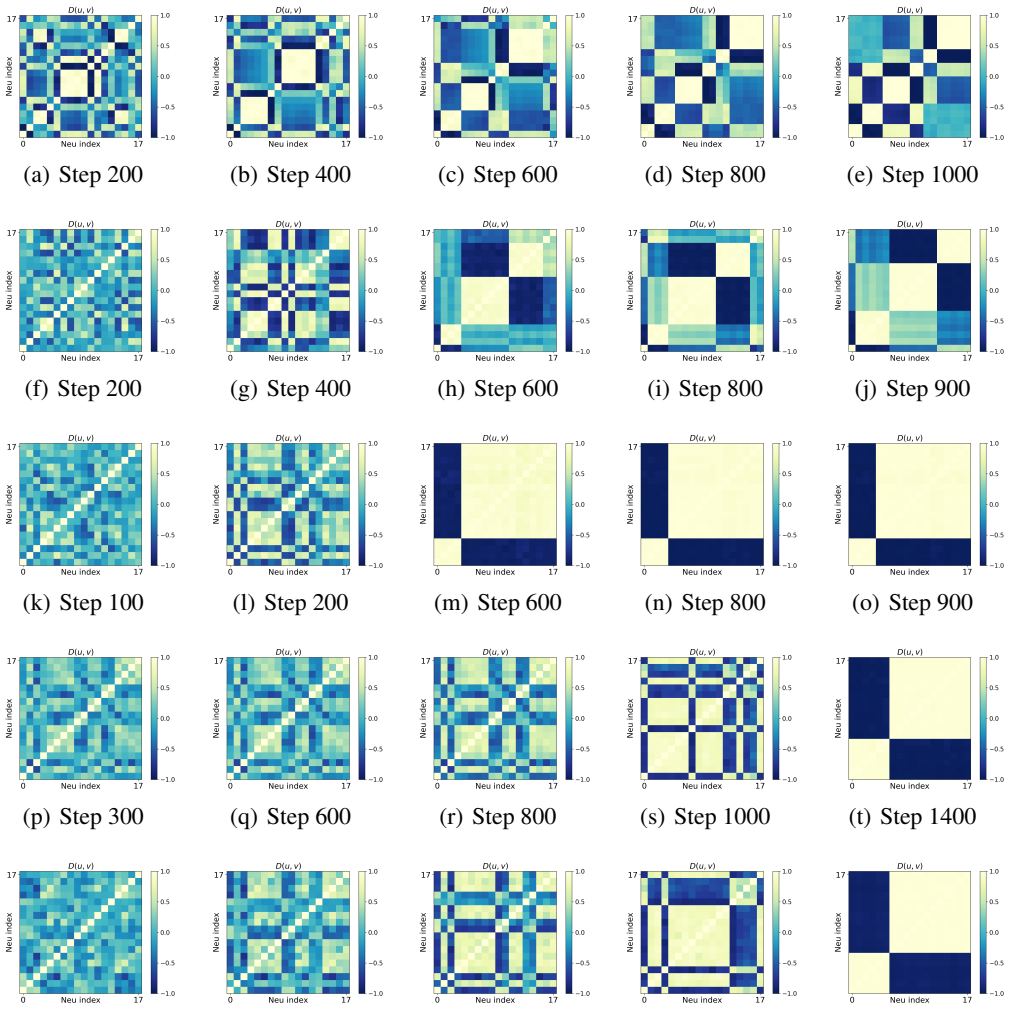
(f) Step 5 (g) Step 10 (h) Step 15 (i) Step 50 (j) Step 100

(k) Step 10 (l) Step 20 (m) Step 40 (n) Step 60 (o) Step 100

(p) Step 10 (q) Step 20 (r) Step 40 (s) Step 60 (t) Step 100

(u) Step 10 (v) Step 20 (w) Step 40 (x) Step 60 (y) Step 100

Figure 15: Evolution of condensation from Fig. 3(a) to 3(e). The evolution from the first row to the fifth row are corresponding to the Fig. 3(a), Fig. 3(b), Fig. 3(c), Fig. 3(d), Fig. 3(e). The numbers of evolutionary steps are shown in the sub-captions, where sub-figures in the last row are the epochs in the article.



(a) Step 200 (b) Step 400 (c) Step 600 (d) Step 800 (e) Step 1000

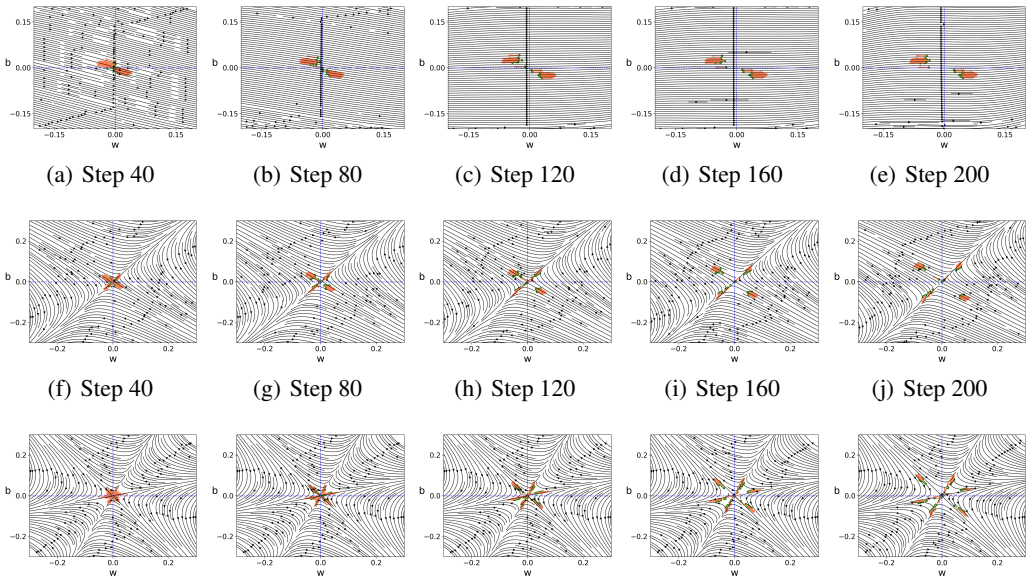
(f) Step 200 (g) Step 400 (h) Step 600 (i) Step 800 (j) Step 900

(k) Step 100 (l) Step 200 (m) Step 600 (n) Step 800 (o) Step 900

(p) Step 300 (q) Step 600 (r) Step 800 (s) Step 1000 (t) Step 1400

(u) Step 100 (v) Step 500 (w) Step 900 (x) Step 1000 (y) Step 1400

Figure 16: Evolution of condensation from Fig. 4(a) to 4(e). The evolution from the first row to the fifth row are corresponding to the Fig. 4(a), Fig. 4(b), Fig. 4(c), Fig. 4(d), Fig. 4(e). The numbers of evolutionary steps are shown in the sub-captions, where sub-figures in the last row are the epochs in the article.



(a) Step 40 (b) Step 80 (c) Step 120 (d) Step 160 (e) Step 200

(f) Step 40 (g) Step 80 (h) Step 120 (i) Step 160 (j) Step 200

(k) Step 40 (l) Step 80 (m) Step 120 (n) Step 160 (o) Step 200

Figure 17: Evolution of condensation from Fig. 6(a) to 6(c). The evolution from the first row to the fifth row are corresponding to the Fig. 6(a), Fig. 6(b), and Fig. 6(c). The numbers of evolutionary steps are shown in the sub-captions, where sub-figures in the last row are the epochs in the article.

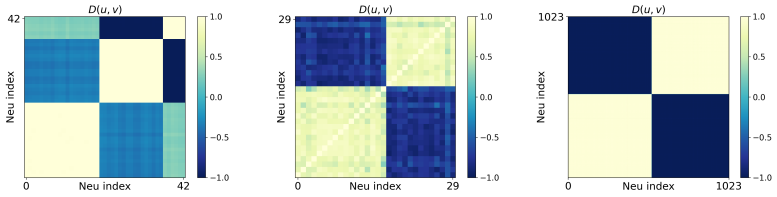
A.7 The influence of training data on condensation

We also find that when the training data is less oscillated, the NN may condense at fewer directions. For example, as shown in Fig. 18(a), compared with the high frequency function in Fig. 3, we only change the target function to be a lower-frequency function, i.e., $\sum_{k=1}^{5} 3.5\sin(2x_k+1)$. In this case, the NN with $x^2\tanh(x)$ only condenses at three directions, in which two are opposite. For MNIST data in Fig. 18(b), we find that, the NN with $x^2\tanh(x)$ condenses at one line, which may suggest that the function for fitting MNIST dataset is a low-frequency function. For CIFAR100 data in Fig. 18(c), we find that input weights of the first FC layer with $x\tanh(x)$ condense at only one line, which implies that features extracted by the convolution part of the NN may own low complexity.

These experiments does not contradict to our results, which claim that the *maximal* number of condensed orientations in the initial training is twice the multiplicity of the activation function used in general NNs.

For CIFAR100 dataset, we use Resnet18-like neural network, which has been described in Fig. 2. Besides, the input dimension is $d = 32 * 32 * 3$, the output dimension is $d_{\rm out} = 100$, and all parameters are initialized by a Gaussian distribution $N(0, {\rm var})$. The total data size is n. The training method is Adam with batch size 128, learning rate ${\rm lr}$ and cross-entropy loss.

For MNIST dataset, we use fully-connected neural network with size, d-m-$\cdots$-m-$d_{\rm out}$. The input dimension is $d = 784$, and the output dimension is $d_{\rm out} = 10$. The number of hidden neurons m is specified in Fig. 18. All parameters are initialized by a Gaussian distribution $N(0, {\rm var})$. The total data size is n. The training method is Adam with full batch, learning rate ${\rm lr}$ and MSE loss.



(a) $\sum_{k=1}^{5} 3.5\sin(2x_k+1)$ (b) MNIST (c) CIFAR100

Figure 18: Condensation of low-frequency functions with two-layer NNs in (a,b) and condensation of the first FC layer of the Resnet18-like network on CIFAR100 in (c). The color indicates $D(\boldsymbol{u}, \boldsymbol{v})$ of two hidden neurons' input weights, whose indexes are indicated by the abscissa and the ordinate. For (a,b), two-layer NN at epoch: 100 with activation function: $x^2\tanh(x)$. For (a), we discard about 15% of hidden neurons, in which the L_2-norm of each input weight is smaller than 0.04, while remaining those bigger than 0.4. The mean magnitude here for each parameter is $(0.4^2/785)^{0.5}$ $\sim$0.01, which should also be quite small. All settings in (a) are the same as Fig. 3, except for the lower frequency target function. Parameters for (b) are $n = 60000$, $d = 784$, $m = 30$, $d_{\rm out} = 10$, ${\rm var} = 0.001^2$. ${\rm lr} = 5 \times 10^{-5}$. The structure and parameters of the Resnet18-like neural network for (c) is the same as Fig. 2, except for the data set CIFAR100 and learning rate ${\rm lr} = 1 \times 10^{-6}$.