Supplementary Material for "Relative Fisher Information and Natural Gradient for Learning Large Modular Models"

Ke Sun and Frank Nielsen

Contents

1 Non-linear Activation Functions

By definition,

$$\mathtt{tanh}(t) \stackrel{\text{def}}{=} \frac{\exp(t)-\exp(-t)}{\exp(t)+\exp(-t)}, \tag{1}$$

and

$$\mathtt{sech}(t) \stackrel{\text{def}}{=} \frac{2}{\exp(t)+\exp(-t)}.$$

It is easy to verify that

$$\mathtt{sech}^2(t) = [1+\mathtt{tanh}(t)]\,[1-\mathtt{tanh}(t)] = 1-\mathtt{tanh}^2(t).$$

By eq. (1),

$$\begin{aligned}
\texttt{tanh}'(t) &= \frac{\exp(t)+\exp(-t)}{\exp(t)+\exp(-t)} - \frac{\exp(t)-\exp(-t)}{[\exp(t)+\exp(-t)]^2}\left[\exp(t)-\exp(-t)\right] \\
&= \frac{[\exp(t)+\exp(-t)]^2-[\exp(t)-\exp(-t)]^2}{[\exp(t)+\exp(-t)]^2} = \frac{4}{[\exp(t)+\exp(-t)]^2} = \texttt{sech}^2(t).
\end{aligned}$$

By definition,

$$\texttt{sigm}(t) \stackrel{\text{def}}{=} \frac{1}{1+\exp(-t)}.$$

Therefore

$$\texttt{sigm}'(t) = -\frac{1}{[1+\exp(-t)]^2}\left(-\exp(-t)\right) = \frac{\exp(-t)}{[1+\exp(-t)]^2} = \texttt{sigm}(t)\left[1-\texttt{sigm}(t)\right].$$

A smoothed version of the `relu` function is given by

$$\texttt{relu}_\omega(t) \stackrel{\text{def}}{=} \omega \ln\left(\exp\left(\frac{\iota t}{\omega}\right)+\exp\left(\frac{t}{\omega}\right)\right),$$

where $\omega > 0$ and $0 \le \iota < 1$. Then,

$$\begin{aligned}
\texttt{relu}'_\omega(t) &= \omega\frac{1}{\exp\left(\frac{\iota t}{\omega}\right)+\exp\left(\frac{t}{\omega}\right)}\left(\frac{\iota}{\omega}\exp\left(\frac{\iota t}{\omega}\right)+\frac{1}{\omega}\exp\left(\frac{t}{\omega}\right)\right) \\
&= \frac{\iota\exp\left(\frac{\iota t}{\omega}\right)+\exp\left(\frac{t}{\omega}\right)}{\exp\left(\frac{\iota t}{\omega}\right)+\exp\left(\frac{t}{\omega}\right)} \\
&= \iota + (1-\iota)\frac{\exp\left(\frac{t}{\omega}\right)}{\exp\left(\frac{\iota t}{\omega}\right)+\exp\left(\frac{t}{\omega}\right)} \\
&= \iota + (1-\iota)\frac{1}{\exp\left((\iota-1)\frac{t}{\omega}\right)+1} \\
&= \iota + (1-\iota)\texttt{sigm}\left(\frac{1-\iota}{\omega}t\right). \qquad (2)
\end{aligned}$$

By definition,

$$\texttt{elu}(t) = \begin{cases} t & \text{if } t \ge 0 \\ \alpha\left(\exp(t)-1\right) & \text{if } t < 0. \end{cases}$$

Therefore

$$\texttt{elu}'(t) = \begin{cases} 1 & \text{if } t \ge 0 \\ \alpha\exp(t) & \text{if } t < 0. \end{cases} \qquad (3)$$

2 Examples of RFIMs

Table 1 shows a list of commonly used RFIMs, with detailed derivations given in the following subsections.

Table 1: Commonly used RFIMs

Subsystem	the RFIM $g^{\boldsymbol{y}}(\boldsymbol{w})$
A `tanh` neuron	$\mathtt{sech}^2(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A `sigm` neuron	$\mathtt{sigm}(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\left[1-\mathtt{sigm}(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\right]\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A `relu` neuron	$\left[\iota+(1-\iota)\mathtt{sigm}\left(\frac{1-\iota}{\omega}\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}\right)\right]^2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal$
A `elu` neuron	$\begin{cases}\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}\ge 0\\ (\alpha\exp(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}))^2\,\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}<0\end{cases}$
A linear layer	$\mathtt{diag}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal,\cdots,\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right]$
A non-linear layer	$\mathtt{diag}\left[\nu_f(\boldsymbol{w}_1,\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal,\cdots,\nu_f(\boldsymbol{w}_m,\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right]$
A soft-max layer	a dense matrix as shown in eq. (10)
Two layers	a dense matrix as shown in eq. (12)

2.1 A Single `tanh` Neuron

Consider a neuron with parameters $\boldsymbol{w}$ and a Bernoulli output $y\in\{+,-\}$, $p(y=+)=p^+$, $p(y=-)=p^-$, and $p^++p^-=1$. By the definition of RFIM, we have

$$\begin{aligned}g^y(\boldsymbol{w})&=p^+\frac{\partial\ln p^+}{\partial\boldsymbol{w}}\frac{\partial\ln p^+}{\partial\boldsymbol{w}^\intercal}+p^-\frac{\partial\ln p^-}{\partial\boldsymbol{w}}\frac{\partial\ln p^-}{\partial\boldsymbol{w}^\intercal}\\&=\frac{1}{p^+}\frac{\partial p^+}{\partial\boldsymbol{w}}\frac{\partial p^+}{\partial\boldsymbol{w}^\intercal}+\frac{1}{p^-}\frac{\partial p^-}{\partial\boldsymbol{w}}\frac{\partial p^-}{\partial\boldsymbol{w}^\intercal}.\end{aligned}$$

Since $p^++p^-=1$,

$$\frac{\partial p^+}{\partial\boldsymbol{w}}+\frac{\partial p^-}{\partial\boldsymbol{w}}=0.$$

Therefore, the RFIM of a Bernoulli neuron has the general form

$$g^y(\boldsymbol{w})=\left(\frac{1}{p^+}+\frac{1}{p^-}\right)\frac{\partial p^+}{\partial\boldsymbol{w}}\frac{\partial p^+}{\partial\boldsymbol{w}^\intercal}=\frac{1}{p^+p^-}\frac{\partial p^+}{\partial\boldsymbol{w}}\frac{\partial p^+}{\partial\boldsymbol{w}^\intercal}.\tag{4}$$

A single `tanh` neuron with stochastic output $y\in\{-1,1\}$ is given by

$$p(y=-1)=\frac{1-\mu(\boldsymbol{x})}{2},\tag{5}$$
$$p(y=1)=\frac{1+\mu(\boldsymbol{x})}{2},\tag{6}$$
$$\mu(\boldsymbol{x})=\mathtt{tanh}(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}}).\tag{7}$$

By eq. (4),

$$\begin{aligned}g^y(\boldsymbol{w})&=\frac{1}{\frac{1-\mu(\boldsymbol{x})}{2}\frac{1+\mu(\boldsymbol{x})}{2}}\left(\frac{1}{2}\frac{\partial\mu}{\partial\boldsymbol{w}}\right)\left(\frac{1}{2}\frac{\partial\mu}{\partial\boldsymbol{w}^\intercal}\right)\\&=\frac{1}{(1-\mu(\boldsymbol{x}))\,(1+\mu(\boldsymbol{x}))}\left[1-\mu^2(\boldsymbol{x})\right]^2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\\&=\left[1-\mu^2(\boldsymbol{x})\right]\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\\&=\left[1-\mathtt{tanh}^2(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\right]\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\\&=\mathtt{sech}^2(\boldsymbol{w}^\intercal\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal.\end{aligned}$$

An alternative analysis is given as follows. By eqs. (5) to (7),

$$
\begin{aligned}
p(y=-1) &= \frac{\exp(-\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}{\exp(\boldsymbol{w}^\intercal \boldsymbol{x}) + \exp(-\boldsymbol{w}^\intercal \boldsymbol{x})}, \\
p(y=1) &= \frac{\exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}{\exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}) + \exp(-\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}.
\end{aligned}
$$

Then,

$$
\begin{aligned}
g^y(\boldsymbol{w}) &= E_{y\sim p(y\,|\,\boldsymbol{x})}\left(-\frac{\partial^2 \ln p(y)}{\partial \boldsymbol{w}\partial \boldsymbol{w}^\intercal}\right) \\
&= \frac{\partial^2}{\partial \boldsymbol{w}\partial \boldsymbol{w}^\intercal} \ln\left[\exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}) + \exp(-\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})\right] \quad \text{(first linear term vanishes)} \\
&= \frac{\partial}{\partial \boldsymbol{w}^\intercal}\left[\frac{\exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}) - \exp(-\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}{\exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}) + \exp(-\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})}\right]\tilde{\boldsymbol{x}} \\
&= \frac{\partial}{\partial \boldsymbol{w}^\intercal}\mathtt{tanh}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}} \\
&= \mathtt{sech}^2(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal.
\end{aligned}
$$

The intuitive meaning of $g^y(\boldsymbol{w})$ is a weighted covariance to emphasize such "informative" $\boldsymbol{x}$'s that

- are in the linear region of `tanh`
- contain "ambiguous" samples

We will need at least $\dim(\boldsymbol{w})$ samples to make $g^y(\boldsymbol{w})$ full rank.

2.2 A Single `sigm` Neuron

A single `sigm` neuron is given by

$$
\begin{aligned}
p(y=0) &= 1-\mu(\boldsymbol{x}), \\
p(y=1) &= \mu(\boldsymbol{x}), \\
\mu(\boldsymbol{x}) &= \mathtt{sigm}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}).
\end{aligned}
$$

By eq. (4),

$$
\begin{aligned}
g^y(\boldsymbol{w}) &= \frac{1}{p(y=0)p(y=1)}\frac{\partial p(y=1)}{\partial \boldsymbol{w}}\frac{\partial p(y=1)}{\partial \boldsymbol{w}^\intercal} \\
&= \frac{1}{\mu(\boldsymbol{x})(1-\mu(\boldsymbol{x}))}\frac{\partial \mu}{\partial \boldsymbol{w}}\frac{\partial \mu}{\partial \boldsymbol{w}^\intercal} \\
&= \frac{1}{\mu(\boldsymbol{x})(1-\mu(\boldsymbol{x}))}\mu^2(\boldsymbol{x})(1-\mu(\boldsymbol{x}))^2\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \\
&= \mu(\boldsymbol{x})(1-\mu(\boldsymbol{x}))\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal \\
&= \mathtt{sigm}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})\left[1-\mathtt{sigm}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}})\right]\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal.
\end{aligned}
$$

2.3 A Single `relu` Neuron

Consider a single neuron with Gaussian output $p(y \mid \boldsymbol{w}, \boldsymbol{x}) = G(y \mid \mu(\boldsymbol{w}, \boldsymbol{x}), \sigma^2)$. Then

$$\begin{aligned}
g^y(\boldsymbol{w} \mid \boldsymbol{x}) &= E_{p(y \mid \boldsymbol{w}, \boldsymbol{x})} \left[\frac{\partial \ln G(y \mid \mu, \sigma^2)}{\partial \boldsymbol{w}} \frac{\partial \ln G(y \mid \mu, \sigma^2)}{\partial \boldsymbol{w}^\intercal} \right] \\
&= E_{p(y \mid \boldsymbol{w}, \boldsymbol{x})} \left[\frac{\partial}{\partial \boldsymbol{w}} \left(-\frac{1}{2\sigma^2} (y-\mu)^2 \right) \frac{\partial}{\partial \boldsymbol{w}^\intercal} \left(-\frac{1}{2\sigma^2} (y-\mu)^2 \right) \right] \\
&= E_{p(y \mid \boldsymbol{w}, \boldsymbol{x})} \left[\left(-\frac{1}{\sigma^2} (\mu - y) \right)^2 \frac{\partial \mu}{\partial \boldsymbol{w}} \frac{\partial \mu}{\partial \boldsymbol{w}^\intercal} \right] \\
&= \frac{1}{\sigma^4} E_{p(y \mid \boldsymbol{w}, \boldsymbol{x})} (\mu - y)^2 \frac{\partial \mu}{\partial \boldsymbol{w}} \frac{\partial \mu}{\partial \boldsymbol{w}^\intercal} \\
&= \frac{1}{\sigma^2} \frac{\partial \mu}{\partial \boldsymbol{w}} \frac{\partial \mu}{\partial \boldsymbol{w}^\intercal}.
\end{aligned}$$

We set $\sigma = 1$ to get rid of a scale parameter of the RFIM. We get

$$g^y(\boldsymbol{w} \mid \boldsymbol{x}) = \frac{\partial \mu}{\partial \boldsymbol{w}} \frac{\partial \mu}{\partial \boldsymbol{w}^\intercal}. \tag{8}$$

A single `relu` neuron is given by

$$\mu(\boldsymbol{w}, \boldsymbol{x}) = \mathtt{relu}_\omega(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}).$$

By eqs. (2) and (8),

$$g^y(\boldsymbol{w}) = \left[\iota + (1-\iota) \mathtt{sigm} \left(\frac{1-\iota}{\omega} \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \right) \right]^2 \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal.$$

2.4 A Single `elu` Neuron

Similar to the analysis in Subsec. 2.3, a single `elu` neuron is given by

$$\mu(\boldsymbol{w}, \boldsymbol{x}) = \mathtt{elu}(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}).$$

By eq. (3),

$$\frac{\partial \mu}{\partial \boldsymbol{w}} = \begin{cases} \tilde{\boldsymbol{x}} & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \geq 0 \\ \alpha \exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}) \tilde{\boldsymbol{x}} & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} < 0. \end{cases}$$

By eq. (8),

$$g^y(\boldsymbol{w}) = \begin{cases} \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} \geq 0 \\ (\alpha \exp(\boldsymbol{w}^\intercal \tilde{\boldsymbol{x}}))^2 \, \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & \text{if } \boldsymbol{w}^\intercal \tilde{\boldsymbol{x}} < 0. \end{cases}$$

2.5 RFIM of a Linear Layer

Consider a linear layer

$$p(\boldsymbol{y}) = G\left(\boldsymbol{y} \mid \boldsymbol{W}^\intercal \tilde{\boldsymbol{x}}, \sigma^2 \boldsymbol{I} \right),$$

where $\boldsymbol{W} = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{D_y})$. By the definition of the multivariate Gaussian distribution,

$$\ln p(\boldsymbol{y}) = -\frac{1}{2} \ln 2\pi - \frac{D_y}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{D_y} \left(y_i - \boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}} \right)^2.$$

Therefore,

$$\forall i, \quad \frac{\partial}{\partial \boldsymbol{w}_i} \ln p(\boldsymbol{y}) = -\frac{1}{\sigma^2} \left(\boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}} - y_i\right) \tilde{\boldsymbol{x}}.$$

Therefore,

$$\forall i, \forall j \quad \frac{\partial}{\partial \boldsymbol{w}_i} \ln p(\boldsymbol{y}) \frac{\partial}{\partial \boldsymbol{w}_j^\intercal} \ln p(\boldsymbol{y}) = \frac{1}{\sigma^4} \left(y_i - \boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}}\right) \left(y_j - \boldsymbol{w}_j^\intercal \tilde{\boldsymbol{x}}\right) \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal.$$

$\boldsymbol{W}$ is vectorized by stacking its columns $\{\boldsymbol{w}_i\}_{i=1}^{D_y}$. In the following $\boldsymbol{W}$ will be used interchangeably to denote either the matrix or its vector form. Correspondingly, the RFIM $g^{\boldsymbol{y}}(\boldsymbol{W})$ has $D_y \times D_y$ blocks, where the off-diagonal blocks are

$$\forall i \neq j, \quad E_{p(\boldsymbol{y})} \left(\frac{\partial}{\partial \boldsymbol{w}_i} \ln p(\boldsymbol{y}) \frac{\partial}{\partial \boldsymbol{w}_j^\intercal} \ln p(\boldsymbol{y}) \right) = \frac{1}{\sigma^4} E_{p(\boldsymbol{y})} \left[\left(y_i - \boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}}\right) \left(y_j - \boldsymbol{w}_j^\intercal \tilde{\boldsymbol{x}}\right) \right] \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal = \boldsymbol{0},$$

and the diagonal blocks are

$$\forall i, \quad E_{p(\boldsymbol{y})} \left(\frac{\partial}{\partial \boldsymbol{w}_i} \ln p(\boldsymbol{y}) \frac{\partial}{\partial \boldsymbol{w}_i^\intercal} \ln p(\boldsymbol{y}) \right) = \frac{1}{\sigma^4} E_{p(\boldsymbol{y})} \left(y_i - \boldsymbol{w}_i^\intercal \tilde{\boldsymbol{x}}\right)^2 \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal = \frac{1}{\sigma^2} \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal.$$

In summary,

$$g^{\boldsymbol{y}}(\boldsymbol{W}) = \frac{1}{\sigma^2} \mathtt{diag}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal, \cdots, \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right].$$

By setting $\sigma = 1$ we get

$$g^{\boldsymbol{y}}(\boldsymbol{W}) = \mathtt{diag}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal, \cdots, \tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\intercal\right].$$

2.6 RFIM of a Non-Linear Layer

The statistical model of a non-linear layer with independent output units is

$$p(\boldsymbol{y} \mid \boldsymbol{W}, \boldsymbol{x}) = \prod_{i=1}^{D_y} p(y_i \mid \boldsymbol{w}_i, \boldsymbol{x}).$$

Then,

$$\ln p(\boldsymbol{y} \mid \boldsymbol{W}, \boldsymbol{x}) = \sum_{i=1}^{D_y} \ln p(y_i \mid \boldsymbol{w}_i, \boldsymbol{x}).$$

Therefore,

$$\frac{\partial^2}{\partial \boldsymbol{W} \partial \boldsymbol{W}^\intercal} \ln p(\boldsymbol{y} \mid \boldsymbol{W}, \boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2}{\partial \boldsymbol{w}_1 \partial \boldsymbol{w}_1^\intercal} \ln p(y_1 \mid \boldsymbol{w}_1, \boldsymbol{x}) & & \\ & \ddots & \\ & & \frac{\partial^2}{\partial \boldsymbol{w}_{D_y} \partial \boldsymbol{w}_{D_y}^\intercal} \ln p(y_{D_y} \mid \boldsymbol{w}_{D_y}, \boldsymbol{x}) \end{bmatrix}.$$

Therefore the RFIM $g^{\boldsymbol{y}}(\boldsymbol{W})$ is a block-diagonal matrix, with the i'th block given by

$$-E_{p(\boldsymbol{y} \mid \boldsymbol{W}, \boldsymbol{x})} \left[\frac{\partial^2}{\partial \boldsymbol{w}_i \partial \boldsymbol{w}_i^\intercal} \ln p(y_i \mid \boldsymbol{w}_i, \boldsymbol{x}) \right] = -E_{p(y_i \mid \boldsymbol{w}_i, \boldsymbol{x})} \left[\frac{\partial^2}{\partial \boldsymbol{w}_i \partial \boldsymbol{w}_i^\intercal} \ln p(y_i \mid \boldsymbol{w}_i, \boldsymbol{x}) \right],$$

which is simply the single neuron RFIM of the i'th neuron.

2.7 RFIM of a Softmax Layer

Recall that

$$\forall i \in \{1, \cdots, m\}, \quad p(y=i) = \frac{\exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})}{\sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})}.$$

Then

$$\forall i, \quad \ln p(y=i) = \boldsymbol{w}_i \tilde{\boldsymbol{x}} - \ln \sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}}).$$

Hence

$$\forall i,\ \forall j, \quad \frac{\partial \ln p(y=i)}{\partial \boldsymbol{w}_j} = \delta_{ij} \tilde{\boldsymbol{x}} - \frac{\exp(\boldsymbol{w}_j \tilde{\boldsymbol{x}})}{\sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})} \tilde{\boldsymbol{x}},$$

where $\delta_{ij} = 1$ if and only if $i = j$ and $\delta_{ij} = 0$ otherwise. Then

$$\begin{aligned}
\forall i,\ \forall j,\ \forall k, \quad \frac{\partial^2 \ln p(y=i)}{\partial \boldsymbol{w}_j \partial \boldsymbol{w}_k^\intercal} &= -\delta_{jk} \frac{\exp(\boldsymbol{w}_j \tilde{\boldsymbol{x}})}{\sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})} \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal + \frac{\exp(\boldsymbol{w}_j \tilde{\boldsymbol{x}})}{\left(\sum_{i=1}^m \exp(\boldsymbol{w}_i \tilde{\boldsymbol{x}})\right)^2} \exp(\boldsymbol{w}_k \tilde{\boldsymbol{x}}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal \\
&= \left(-\delta_{jk} \eta_j + \eta_j \eta_k\right) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal. \qquad (9)
\end{aligned}$$

The right-hand-side of eq. (9) does not depend on i. Therefore

$$g^{\boldsymbol{y}}(\boldsymbol{W}) = \begin{bmatrix} (\eta_1 - \eta_1^2) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & -\eta_1 \eta_2 \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_1 \eta_m \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal \\ -\eta_2 \eta_1 \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & (\eta_2 - \eta_2^2) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & \cdots & -\eta_2 \eta_m \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal \\ \vdots & \vdots & \ddots & \vdots \\ -\eta_m \eta_1 \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & -\eta_m \eta_2 \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal & \cdots & (\eta_m - \eta_m^2) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal \end{bmatrix}. \qquad (10)$$

2.8 RFIM of Two layers

Consider a two layer structure, where the output $\boldsymbol{y}$ satisfies a multivariate Bernoulli distribution with independent dimensions. By a similar analysis to Subsec. 2.1, we have

$$g^{\boldsymbol{y}}(\boldsymbol{W}) = \sum_{l=1}^{D_y} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) \frac{\partial \boldsymbol{c}_l^\intercal \boldsymbol{h}}{\partial \boldsymbol{W}} \frac{\partial \boldsymbol{c}_l^\intercal \boldsymbol{h}}{\partial \boldsymbol{W}^\intercal}. \qquad (11)$$

It can be written block by block as $g^{\boldsymbol{y}}(\boldsymbol{W}) = [\boldsymbol{G}_{ij}]_{D_h \times D_h}$, where each block $\boldsymbol{G}_{ij}$ means the correlation between the i'th hidden neuron with weights $\boldsymbol{w}_i$ and the j'th hidden neuron with weights $\boldsymbol{w}_j$. By eq. (11),

$$\begin{aligned}
\boldsymbol{G}_{ij} &= \sum_{l=1}^{D_y} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) \frac{\partial \boldsymbol{c}_l^\intercal \boldsymbol{h}}{\partial \boldsymbol{w}_i} \frac{\partial \boldsymbol{c}_l^\intercal \boldsymbol{h}}{\partial \boldsymbol{w}_j^\intercal} = \sum_{l=1}^{D_y} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) \frac{\partial c_{il} h_i}{\partial \boldsymbol{w}_i} \frac{\partial c_{jl} h_j}{\partial \boldsymbol{w}_j^\intercal} \\
&= \sum_{l=1}^{D_y} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) c_{il} c_{jl} \frac{\partial h_i}{\partial \boldsymbol{w}_i} \frac{\partial h_j}{\partial \boldsymbol{w}_j^\intercal} = \sum_{l=1}^{D_y} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) c_{il} c_{jl} \left(\nu_f(\boldsymbol{w}_i, \boldsymbol{x}) \tilde{\boldsymbol{x}}\right) \left(\nu_f(\boldsymbol{w}_j, \boldsymbol{x}) \tilde{\boldsymbol{x}}^\intercal\right) \\
&= \sum_{l=1}^{D_y} c_{il} c_{jl} \nu_f(\boldsymbol{c}_l, \boldsymbol{h}) \nu_f(\boldsymbol{w}_i, \boldsymbol{x}) \nu_f(\boldsymbol{w}_j, \boldsymbol{x}) \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^\intercal. \qquad (12)
\end{aligned}$$

The proof of the other case, where two `relu` layers have stochastic output $\boldsymbol{y}$ satisfying a multivariate Gaussian distribution with independent dimensions, is very similar and is omitted.

3 Proof of Theorem 3

Proof. By assumption, the joint distribution $p(\boldsymbol{x},\, \boldsymbol{h})$ is in a factorable form. Therefore

$$\log p(\boldsymbol{x},\, \boldsymbol{h}) = \sum_{l=1}^{L} \log p(\boldsymbol{h}_l \,|\, \boldsymbol{\theta}_l, \boldsymbol{r}_l), \tag{13}$$

where $l = 1, \cdots, L$ is the index of subsystems, $\boldsymbol{h}_l$ is the subsystem output, and $\boldsymbol{r}_l$ is the reference of the subsystem. We have $\biguplus_{l=1}^{L}\{\boldsymbol{h}_l\} = \{\boldsymbol{x}, \boldsymbol{h}\}$ and $\biguplus_{l=1}^{L}\{\boldsymbol{\theta}_l\} = \{\boldsymbol{\Theta}\}$. Therefore

$$\begin{aligned} E_p\left(-\frac{\partial^2}{\partial\boldsymbol{\theta}_l\partial\boldsymbol{\theta}_l^{\intercal}} \log p(\boldsymbol{x},\, \boldsymbol{h})\right) &= E_p\left(-\frac{\partial^2}{\partial\boldsymbol{\theta}_l\partial\boldsymbol{\theta}_l^{\intercal}} \log p(\boldsymbol{h}_l \,|\, \boldsymbol{\theta}_l, \boldsymbol{r}_l)\right) \\ &= E_{p(\boldsymbol{r}_l)}\left(E_{p(\boldsymbol{h}_l \,|\, \boldsymbol{r}_l)}\left(-\frac{\partial^2}{\partial\boldsymbol{\theta}_l\partial\boldsymbol{\theta}_l^{\intercal}} \log p(\boldsymbol{h}_l \,|\, \boldsymbol{\theta}_l, \boldsymbol{r}_l)\right)\right) \\ &= E_p\left(g^{\boldsymbol{h}_l}(\boldsymbol{\theta}_l)\right), \end{aligned}$$

and

$$E_p\left(-\frac{\partial^2}{\partial\boldsymbol{\theta}_{l_1}\partial\boldsymbol{\theta}_{l_2}^{\intercal}} \log p(\boldsymbol{x},\, \boldsymbol{h})\right) = \mathbf{0} \quad (\forall l_1 \neq l_2).$$

Based on the Hessian expression of RFIM, $\mathcal{J}(\boldsymbol{\Theta})$ is in a block-diagonal form, with each block given by $E_p\left(g^{\boldsymbol{h}_l}(\boldsymbol{\theta}_l)\right)$. $\square$

4 Experimental Settings & Zoomed Learning Curves

The training/validation/testing sets have 50,000/10,000/10,000 images, respectively. Each sample is a gray scale image of size 28×28 (784 dimensional feature space) and is labeled as one of ten different classes. For all methods, the mini-batch size is fixed to 50 and the L_2 regularization strength is fixed to 10^{-3}. For each optimizer, we try to find the best learning rate in the range $\{\cdots, 10^{-1}, 5\times10^{-2}, 10^{-2}, 5\times10^{-3}, 10^{-3}, \cdots\}$. On the tested architectures, a good learning rate configuration for RNGD is usually around 10^{-2} or 5×10^{-3}. The optimizers are in their default settings in TensorFlow 1.0. For the Adam optimizer, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. For RNGD, we set empirically $T = 100$, $\lambda = 0.005$ and $\omega = 1$. We use the Glorot uniform initializer to set the initial weights.

For each method and each learning rate configuration, we try 40 independent runs with different random seeds. Then, we select the best configuration based on the validation accuracy. Then, we plot the 40 learning curves as well as the average validation curve. The learning curves are obtained by evaluating the training error and validation accuracy after each epoch (one pass over all available training data).

See the following figs. (1–4) for the learning curves on four different architectures with `relu` activation units and L_2 regularization. Only the training curves and validation curves are shown for a clear presentation. The testing accuracy is close to the validation accuracy (run our codes to see the detailed results).

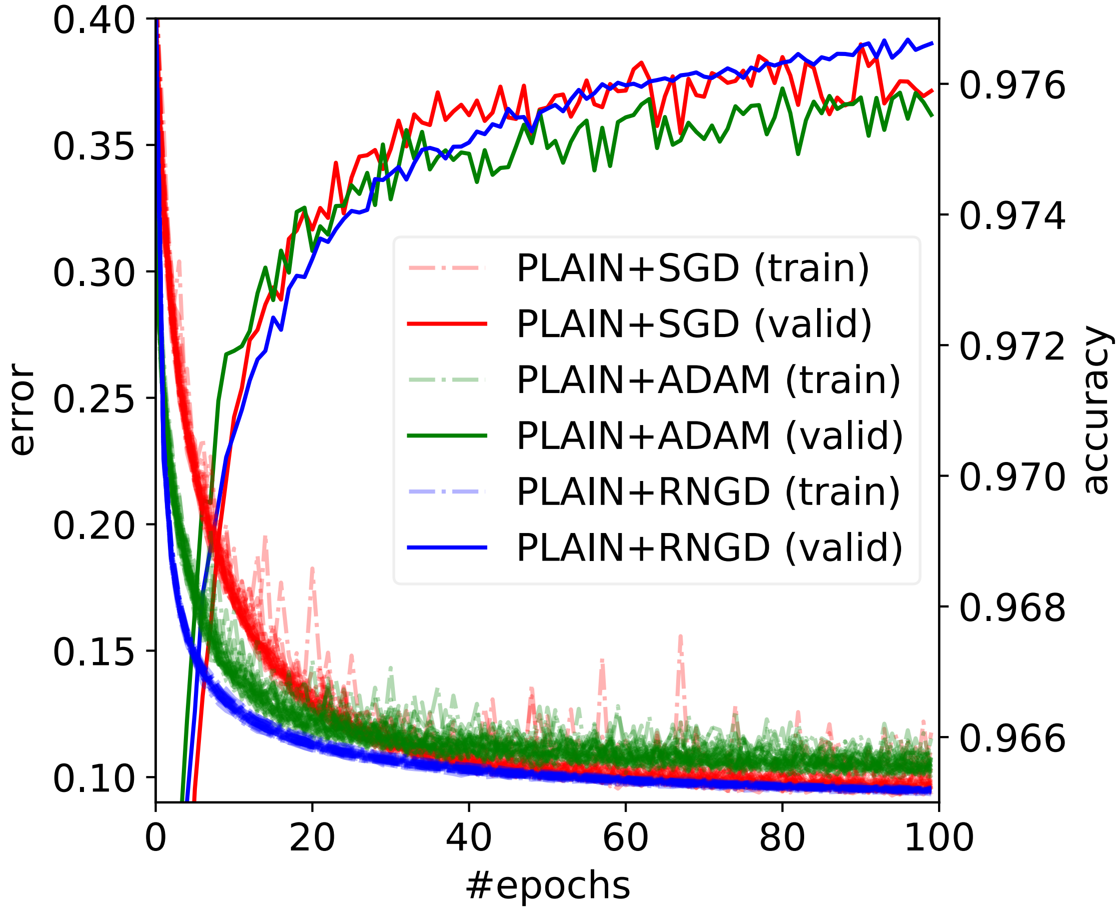

Figure 1: A MLP with shape 784–80–80–80–10.

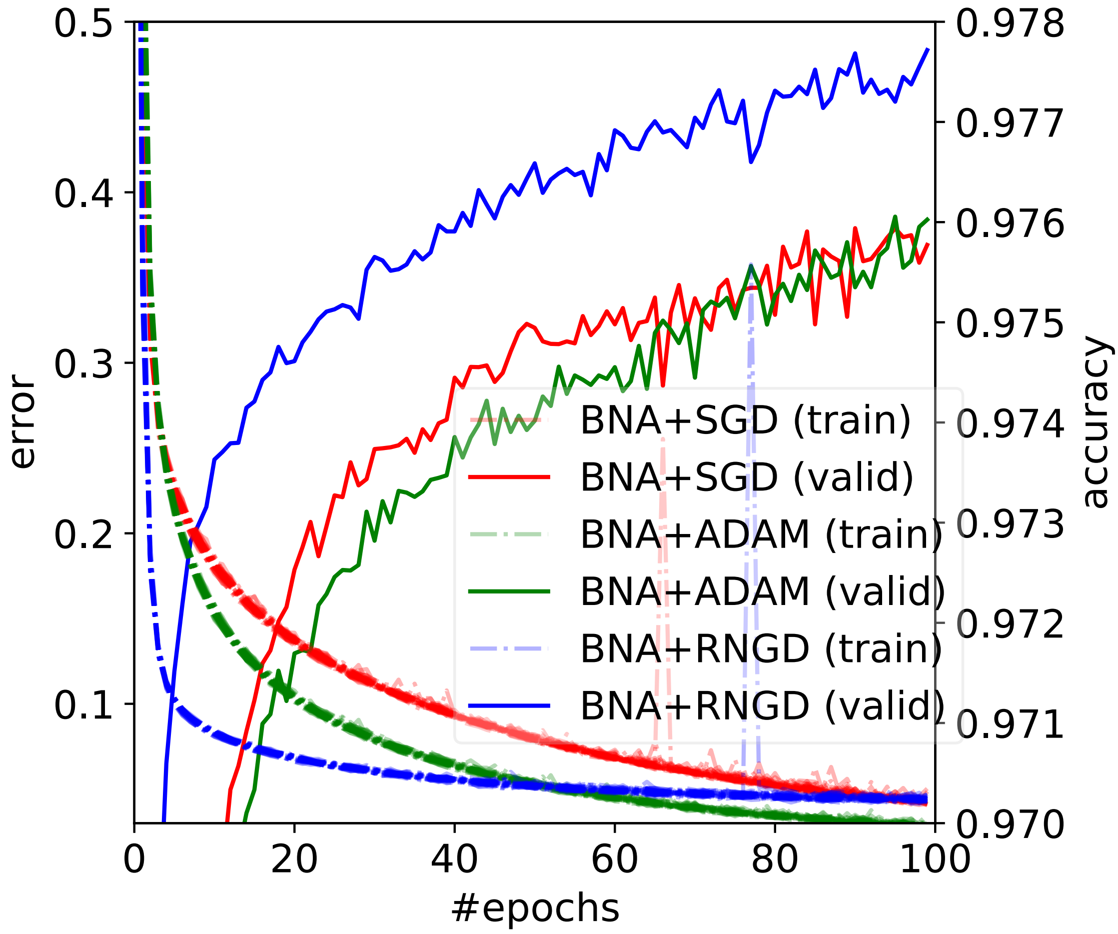

Figure 2: A MLP with shape 784–80–80–80–10 and batch normalization after each hidden layer.

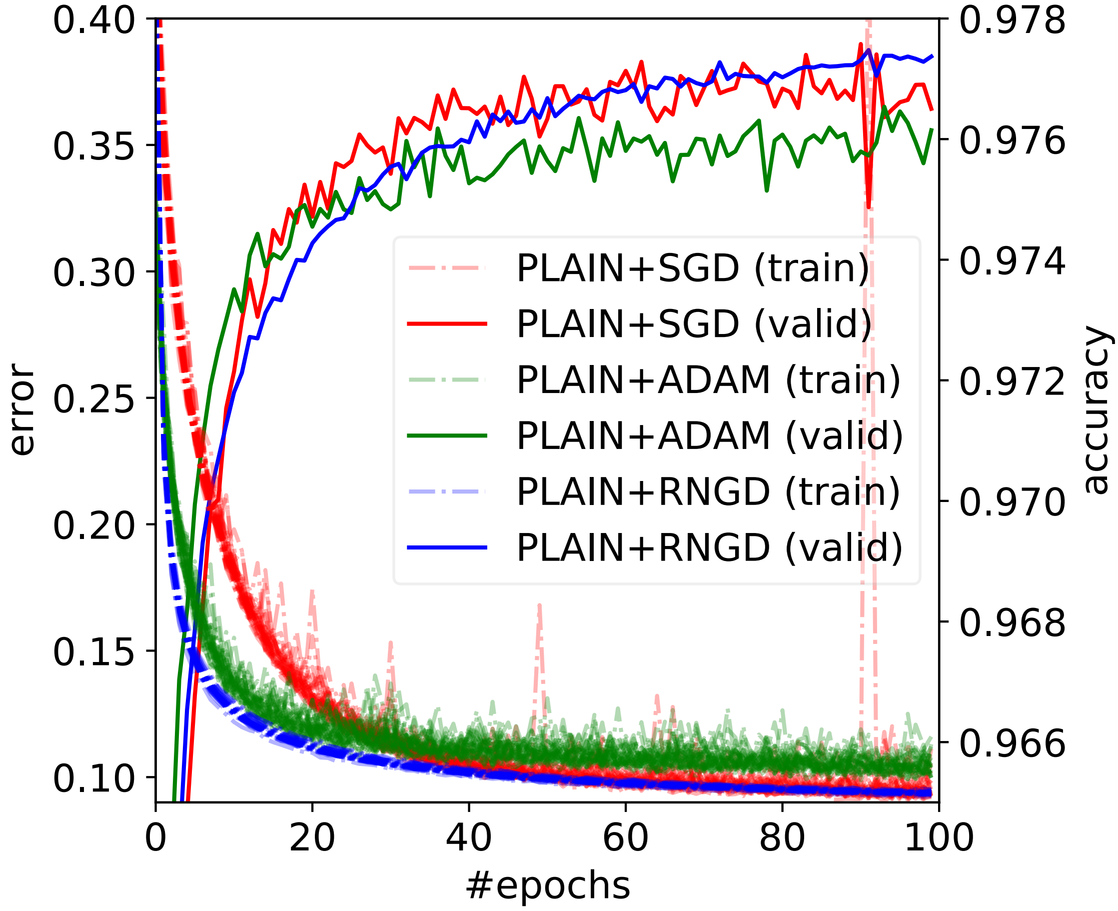

Figure 3: A MLP with shape 784–100–100–100–10.

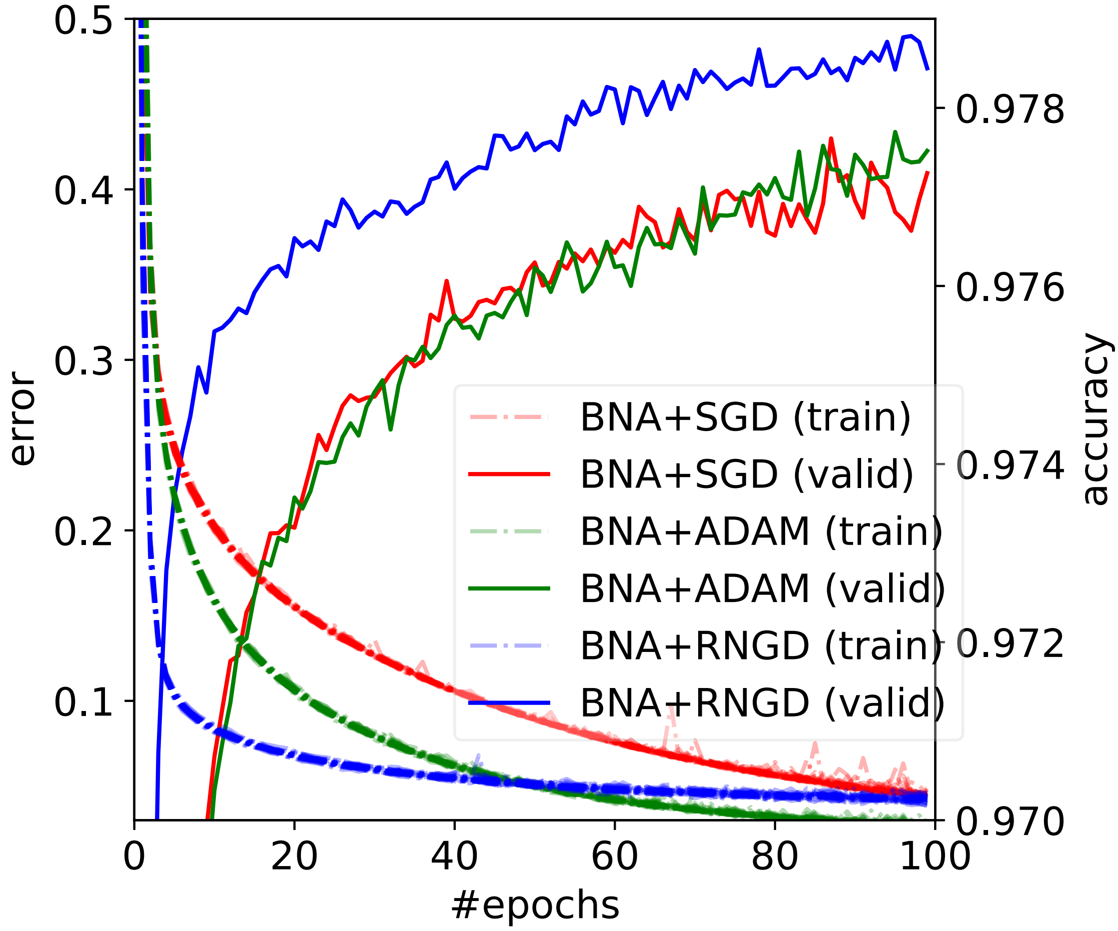

Figure 4: A MLP with shape 784–100–100–100–10 and batch normalization after each hidden layer.