Can the Brain Do Backpropagation? — Exact Implementation of Backpropagation in Predictive Coding Networks

Yuhang Song[1], **Thomas Lukasiewicz**[1], **Zhenghua Xu**[2,*], **Rafal Bogacz**[3]
[1]Department of Computer Science, University of Oxford, UK
[2]State Key Laboratory of Reliability and Intelligence of Electrical Equipment,
Hebei University of Technology, Tianjin, China
[3]MRC Brain Network Dynamics Unit, University of Oxford, UK
yuhang.song@some.ox.ac.uk, thomas.lukasiewicz@cs.ox.ac.uk,
zhenghua.xu@hebut.edu.cn, rafal.bogacz@ndcn.ox.ac.uk

A Supplementary Material

A.1 Bias Values as Parameters

We now add bias values as parameters, denoted by v_i^{l+1} and β_i^{l+1} for ANNs and PCNs, respectively. Formally, in ANNs trained with BP, originally recalled in Section 2.1, Eq. (1) becomes:

$$y_i^l = \sum_{j=1}^{n^{l+1}} w_{i,j}^{l+1} f(y_j^{l+1}) + v_i^{l+1}. \tag{10}$$

Accordingly, we have an update rule for bias parameters (bias update) in addition to the one of the weight parameters (weight update):

$$\Delta v_{i,j}^{l+1} = -\alpha \cdot \partial E / \partial v_{i,j}^{l+1} = \alpha \cdot \delta_i^l. \tag{11}$$

Similarly, in PCNs trained with IL, originally recalled in Section 2.2, Eq. (5) becomes:

$$\mu_{i,t}^l = \sum_{j=1}^{n^{l+1}} \theta_{i,j}^{l+1} f(x_{j,t}^{l+1}) + \beta_i^{l+1} \quad \text{and} \quad \varepsilon_{i,t}^l = x_{i,t}^l - \mu_{i,t}^l. \tag{12}$$

Accordingly, we have an update rule for bias parameters in addition to the one of the weight parameters:

$$\Delta \beta_{i,j}^{l+1} = -\alpha \cdot \partial F_t / \partial \beta_{i,j}^{l+1} = \alpha \cdot \varepsilon_{i,t}^l. \tag{13}$$

Otherwise, all equations and conclusions still hold in Section 2.

In Section 3, we only need to add the assumption that the bias parameter for both ANNs and PCNs are also identical initially. To prove the conclusion of zero divergence of the weights update, the procedure remains unchanged. To prove the conclusion of zero divergence of the bias update, we only need Theorem 3.1, as it directly leads to the equivalence of bias update by Eqs. (11) and (13).

A.2 Derivations of Eq. (7)

Before we start, we expand Eq. (6) with the definition of $\varepsilon_{i,t}^l = x_{i,t}^l - \mu_{i,t}^l$:

$$F_t = \sum_{l=0}^{l_{\max}-1} \sum_{i=1}^{n^l} \frac{1}{2} (\varepsilon_{i,t}^l)^2 = \sum_{l=0}^{l_{\max}-1} \sum_{i=1}^{n^l} \frac{1}{2} (x_{i,t}^l - \mu_{i,t}^l)^2 \, . \tag{14}$$

Inference minimizes F_t by modifying $x_{i,t}^l$ proportionally to the gradient of the objective function F_t. To calculate the derivative of F_t over $x_{i,t}^l$, we note that each $x_{i,t}^l$ influences F_t in two ways: (1) it

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

occurs in Eq. (14) explicitly, but (2) it also determines the values of $\mu^{l-1}_{k,t}$ via Eq. (5). Thus, the derivative contains two terms:

$$\Delta x^l_{i,t} = -\gamma \cdot \frac{\partial F_t}{\partial x^l_{i,t}} \tag{15}$$

$$= -\gamma \cdot \Big(\frac{\partial \frac{1}{2}(x^l_{i,t} - \mu^l_{i,t})^2}{\partial x^l_{i,t}} + \frac{\partial \sum_{k=1}^{n^{l-1}} \frac{1}{2}(x^{l-1}_{k,t} - \mu^{l-1}_{k,t})^2}{\partial x^l_{i,t}}\Big) \tag{16}$$

$$= \gamma \cdot (-(x^l_{i,t} - \mu^l_{i,t}) + f'(x^l_{i,t}) \textstyle\sum_{k=1}^{n^{l-1}} (x^{l-1}_{k,t} - \mu^{l-1}_{k,t}) \theta^l_{k,i}) \tag{17}$$

$$= \gamma \cdot (-\varepsilon^l_{i,t} + f'(x^l_{i,t}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,t} \theta^l_{k,i}) \,. \tag{18}$$

Considering also the special cases at $l = l_{\text{max}}$ and $l = 0$, we obtain Eq. (7).

A.3 Derivations of Eq. (9)

The weights update minimizes F_t by modifying $\theta^{l+1}_{i,j}$ proportionally to the gradient of the objective function F_t. To compute the derivative of the objective function F_t over $\theta^{l+1}_{i,j}$, we note that $\theta^{l+1}_{i,j}$ affects the value of the function F_t of Eq. (14) by influencing $\mu^l_{i,t}$ via Eq. (5), hence,

$$\Delta \theta^{l+1}_{i,j} = -\alpha \cdot \partial F_t / \partial \theta^{l+1}_{i,j} \tag{19}$$

$$= -\alpha \cdot \frac{\partial \frac{1}{2}(x^l_{i,t} - \mu^l_{i,t})^2}{\partial \theta^{l+1}_{i,j}} \tag{20}$$

$$= \alpha \cdot \varepsilon^l_{i,t} f(x^{l+1}_{j,t}) \,. \tag{21}$$

A.4 Proof of Theorems 3.1 and 3.2

Theorem 3.1. *Let M be a PCN, M' be its corresponding ANN (with the same initial weights as M), and let s be a datapoint. Then, every prediction error $\varepsilon^l_{i,t}$ at $t = l$, $l \in \{0, \dots, l_{max}-1\}$, in M trained with IL on s under C1 and C3 is equal to the error term δ^l_i in M' trained with BP on s.*

Proof. We first note that $\varepsilon^l_{i,t}$ under C2 is $\varepsilon^l_{i,l}$. We give a proof by induction on the depth l of the PCNs and ANNs. For PCNs, as $t = l$, it is also inducing on the inference moments.

◇ *Base Case*: If $l = 0$,

putting C1 $\mu^l_{i,0} = y^l_i$ into Eq. (8), and by comparison with the first case in Eq. (4): $\varepsilon^l_{i,l} = \delta^l_i$.

◇ *Induction Step*: For $l \in \{1, \dots, l_{\text{max}} - 1\}$,

$\varepsilon^l_{i,l} = f'(\mu^l_{i,0}) \sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^l_{k,i}$, by Lemma A.4;

$\delta^l_i = f'(y^l_i) \sum_{k=1}^{n^{l-1}} \delta^{l-1}_k w^l_{k,i}$, by the second case in Eq. (4);

$\mu^l_{i,0} = y^l_i$, by C1;

$w^l_{i,j} = \theta^l_{i,j}$, as corresponding initial weights in both models are assumed to be identical;

$\varepsilon^l_{i,l} = \delta^l_i$, if $\varepsilon^{l-1}_{k,l-1} = \delta^{l-1}_k$.

□

Theorem 3.2. *Let M be a PCN, M' be its corresponding ANN (with the same initial weights as M), and let s be a datapoint. Then, every update $\Delta \theta^{l+1}_{i,j}$ at $t = l$, $l \in \{0, \dots, l_{max}-1\}$, in M trained with IL on s under C1 and C3 is equal to the update $\Delta w^{l+1}_{i,j}$ in M' trained with BP on s.*

Proof. Looking at how $\partial F_t / \partial \theta^{l+1}_{i,j}$ and $\partial E / \partial w^{l+1}_{i,j}$ can be computed via Eqs. (9) and (3), and putting C2 $t = l$ into the equation,

$$\Delta \theta^{l+1}_{i,j} = \alpha \cdot \varepsilon^{l}_{i,l} f(x^{l+1}_{j,l}), \tag{22}$$

$$\Delta w^{l+1}_{i,j} = \alpha \cdot \delta^{l}_{i} f(y^{l+1}_{j}). \tag{23}$$

We notice that one of the terms in both equations are equivalent according to Theorem 3.1: $\varepsilon^{l}_{i,l} = \delta^{l}_{i}$. Thus, in the following, we focus on proving that the other terms in both equations are identical: $f(x^{l+1}_{j,l}) = f(y^{l+1}_{j})$. First, C1 provides the base to start, which is the equivalence of the initial state between IL and BP under C1, C2, and C3: $x^{l+1}_{j,0} = \mu^{l+1}_{j,0} = y^{l+1}_{j}$. Then, Lemma A.3 links later inference moments of IL under C1, C2, and C3 to its initial state: $x^{l+1}_{j,l} = x^{l+1}_{j,0}$. Thus, we have $f(x^{l+1}_{j,l}) = f(y^{l+1}_{j})$. □

Lemma A.3. *Under C1, so that a variable at a specific layer may diverge from its corresponding initial stable states, it needs specific inference steps related to the layer that the variable belongs to. Formally,*

$$\overline{x}^{l}_{t<l} = \overline{x}^{l}_{0}, \overline{\varepsilon}^{l}_{t<l} = \overline{\varepsilon}^{l}_{0} = 0, \overline{\mu}^{l-1}_{t<l} = \overline{\mu}^{l-1}_{0}, \textit{ for } l \in \{1, \ldots, l_{\textit{max}} - 1\}, \textit{ i.e.,}$$
$$\Delta\overline{x}^{l}_{t<l-1} = \overline{0}, \Delta\overline{\varepsilon}^{l}_{t<l-1} = \overline{0}, \Delta\overline{\mu}^{l-1}_{t<l-1} = \overline{0}, \textit{ for } l \in \{1, \ldots, l_{\textit{max}} - 1\}.$$

Proof. Starting from the inference moment $t = 0$, $\overline{x}^{0}_{0}$ is dragged away from $\overline{\mu}^{0}_{0}$ and fixed to $\overline{s}^{\text{out}}$, i.e., $\overline{\varepsilon}^{0}_{0}$ turns into nonzero from zero. Since $\overline{x}$ in each layer is updated only on the basis of $\overline{\varepsilon}$ in the same and previous adjacent layer, as indicated by Eq. (7), also considering C1, $\overline{\varepsilon}$ is initially zero for all layers but the output layer, it will take l time steps to modify $\overline{x}^{l}_{t}$ at layer l from the initial state. Hence, $\overline{x}^{l}_{t}$ will remain in that initial state $\overline{x}^{l}_{0}$ for all $t < l$, i.e., $\overline{x}^{l}_{t<l} = \overline{x}^{l}_{0}$. Furthermore, any change in $\overline{x}^{l}_{t}$ causes a change in $\overline{\varepsilon}^{l}_{t}$ and $\overline{\mu}^{l-1}_{t}$ instantly via Eq. (5) (otherwise $\overline{\varepsilon}^{l}_{t}$ and $\overline{\mu}^{l-1}_{t}$ remain in their corresponding initial states). Thus, we know $\overline{\varepsilon}^{l}_{t<l} = \overline{\varepsilon}^{l}_{0}$ and $\overline{\mu}^{l-1}_{t<l} = \overline{\mu}^{l-1}_{0}$. Also, according to C1, $\overline{\varepsilon}^{l}_{t<l} = \overline{\varepsilon}^{l}_{0} = 0$. Equivalently, we have $\Delta\overline{x}^{l}_{t<l-1} = \overline{0}$, $\Delta\overline{\varepsilon}^{l}_{t<l-1} = \overline{0}$, and $\Delta\overline{\mu}^{l-1}_{t<l-1} = \overline{0}$. □

Lemma A.4. *The prediction error of IL $\varepsilon^{l}_{i,t}$ at $t = l$ (i.e., $\varepsilon^{l}_{i,l}$) under C1 and C3 can be derived from itself at previous inference moments in the previous layer: $\varepsilon^{l-1}_{k,t}$ at $t = l - 1$ (i.e., $\varepsilon^{l-1}_{i,l-1}$). Formally,*

$$\varepsilon^{l}_{i,l} = f'(\mu^{l}_{i,0}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^{l}_{k,i}, \textit{ for } l \in \{1, \ldots, l_{\textit{max}} - 1\}. \tag{24}$$

Proof. We first write a dynamic version of $\varepsilon^{l}_{i,t} = x^{l}_{i,t} - \mu^{l}_{i,t}$:

$$\varepsilon^{l}_{i,t} = \varepsilon^{l}_{i,t-1} + \left(\Delta x^{l}_{i,t-1} - \Delta \mu^{l}_{i,t-1}\right), \tag{25}$$

where $\Delta \mu^{l}_{i,t-1} = \mu^{l}_{i,t} - \mu^{l}_{i,t-1}$. Then, we expand $\varepsilon^{l}_{i,l}$ with the above equation and simplify it with Lemma A.3, i.e., $\varepsilon^{l}_{i,t<l} = 0$ and $\Delta \mu^{l-1}_{i,t<l-1} = 0$:

$$\varepsilon^{l}_{i,l} = \varepsilon^{l}_{i,l-1} + \left(\Delta x^{l}_{i,l-1} - \Delta \mu^{l}_{i,l-1}\right) = \Delta x^{l}_{i,l-1}, \text{ for } l \in \{1, \ldots, l_{\max} - 1\}. \tag{26}$$

We further investigate $\Delta x^{l}_{i,l-1}$ expanded with the inference dynamic Eq. (7) and simplify it with Lemma A.3, i.e., $\varepsilon^{l}_{i,t<l} = 0$,

$$\Delta x^{l}_{i,l-1} = \gamma(-\varepsilon^{l}_{i,l-1} + f'(x^{l}_{i,l-1})) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^{l}_{k,i} = \gamma f'(x^{l}_{i,l-1}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^{l}_{k,i}, \text{ for } l \in \{1, \ldots, l_{\max} - 1\}. \tag{27}$$

Putting Eq. (27) into Eq. (26), we obtain:

$$\varepsilon^{l}_{i,l} = \gamma f'(x^{l}_{i,l-1}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^{l}_{k,i}, \text{ for } l \in \{1, \ldots, l_{\max} - 1\}. \tag{28}$$

With Lemma A.3, $x^{l}_{i,l-1}$ can be replaced with $x^{l}_{i,0}$. With C1, we can further replace $x^{l}_{i,0}$ with $\mu^{l}_{i,0}$. Thus, the above equation becomes:

$$\varepsilon^{l}_{i,l} = \gamma f'(\mu^{l}_{i,0}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^{l}_{k,i}, \text{ for } l \in \{1, \ldots, l_{\max} - 1\}. \tag{29}$$

Then, put C3, $\gamma = 1$, into the above equation. □

A.5 Solely Relaxing C3

Solely relaxing C3 will result in the conclusion of Lemma A.4, i.e., Eq. (24) changing to:

$$\varepsilon^l_{i,l} = \gamma f'(\mu^l_{i,0}) \textstyle\sum_{k=1}^{n^{l-1}} \varepsilon^{l-1}_{k,l-1} \theta^l_{k,i}, \text{ for } l \in \{1, \dots, l_{\text{max}} - 1\}, \tag{30}$$

since the derivation of Lemma A.4 terminates at Eq. (29). It further causes the conclusion of Theorem 3.1 changing from $\varepsilon^l_{i,t} = \delta^l_i$ to $\varepsilon^l_{i,t} = \gamma^l \delta^l_i$ at $t = l$, the proof of which is the same as that of the original Theorem 3.1 but using the new Lemma A.4. The change in the conclusion of Theorem 3.1 immediately changes the conclusion of Theorem 3.2 from $\partial F_t / \partial \theta^{l+1}_{i,j} = \partial E / \partial w^{l+1}_{i,j}$ to $\partial F_t / \partial \theta^{l+1}_{i,j} = \gamma^l \partial E / \partial w^{l+1}_{i,j}$, where $t = l$. Thus, solely relaxing the condition $\gamma = 1$, i.e., relaxing C3 while keeping C1 and C2 satisfied, results in BP with a different learning rate for different layers, where γ is the decay factor of this learning rate along layers.

A.6 Schematic Algorithms

Algorithm 1 Learning one training pair ($\overline{s}^{\text{in}}, \overline{s}^{\text{out}}$) (presented for the duration T) with IL

```
Require: $\overline{x}_0^{l_{\max}}$ is fixed to $\overline{s}^{\text{in}}$, $\overline{x}_0^0$ is fixed to $\overline{s}^{\text{out}}$.
for $t = 0$ to $T$ do                                      // presenting a training pair
    for each neuron $i$ in each level $l$ do               // in parallel in the brain
        Update $x^l_{i,t}$ to minimize $F_t$ via Eq. (7)   // inference
        if $t = t_c$ then                                  // external control signal
            Update each $\theta^{l+1}_{i,j}$ to minimize $F_t$ via Eq. (9)
            return                                         // the brain rests
        end if
    end for
end for
```

Algorithm 2 Learning one training pair ($\overline{s}^{\text{in}}, \overline{s}^{\text{out}}$) (presented for the duration T) with Z-IL

```
Require: $\overline{x}_0^{l_{\max}}$ is fixed to $\overline{s}^{\text{in}}$, $\overline{x}_0^0$ is fixed to $\overline{s}^{\text{out}}$.
Require: $x^l_{i,0} = \mu^l_{i,0}$ for $l \in \{1, \dots, l_{\text{max}} - 1\}$ (C1), and $\gamma = 1$ (C3).
for $t = 0$ to $T$ do                                      // presenting a training pair
    for each neuron $i$ in each level $l$ do               // in parallel in the brain
        Update $x^l_{i,t}$ to minimize $F_t$ via Eq. (7)   // inference
        if $t = l$ then                                    // external control signal
            Update each $\theta^{l+1}_{i,j}$ to minimize $F_t$ via Eq. (9)
        end if
        if $t = l_{\text{max}}$ then
            return                                         // the brain rests
        end if
    end for
end for
```

Algorithm 3 Learning one training pair ($\overline{s}^{\text{in}}, \overline{s}^{\text{out}}$) (presented for the duration T) with Fa-Z-IL

```
Require: $\overline{x}_0^{l_{\max}}$ is fixed to $\overline{s}^{\text{in}}$, $\overline{x}_0^0$ is fixed to $\overline{s}^{\text{out}}$.
Require: $x^l_{i,0} = \mu^l_{i,0}$ for $l \in \{1, \dots, l_{\text{max}} - 1\}$ (C1), and $\gamma = 1$ (C3).
for $t = 0$ to $T$ do                                      // presenting a training pair
    for each neuron $i$ in each level $l$ do               // in parallel in the brain
        Update $x^l_{i,t}$ to minimize $F_t$ via Eq. (7)   // inference
        Update each $\theta^{l+1}_{i,j}$ to minimize $F_t$ via Eq. (9) with learning rate $\alpha \cdot \phi(\varepsilon^l_{i,t})$
    end for
end for
```

Schematic algorithms of IL, Z-IL, and Fa-Z-IL are provided in Algorithms 1, 2, and 3 for comparison, respectively, the two of which, IL and Fa-Z-IL are already provided in the paper.

Table 1: Unscaled data in Fig. 2.

σ	0	0.0001	0.001	0.01	0.1	1	10	100
Divergence of test error	$\mathbf{0}$	4.23×10^{-3}	1.77×10^{-2}	6.05×10^{-2}	6.15×10^{-1}	3.73×10^{1}	4.17×10^{1}	4.26×10^{1}
Divergence of final weights	$\mathbf{2.37\times10^{-13}}$	2.40×10^{-2}	9.94×10^{-1}	3.36×10^{0}	1.43×10^{1}	1.08×10^{2}	1.12×10^{3}	1.22×10^{4}

Table 2: Test error of different σ.

σ	0	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}
Test error	1.0546×10^{-1}	1.0548×10^{-1}	1.0557×10^{-1}	1.0596×10^{-1}	1.1507×10^{-1}	6.8915×10^{-1}	7.5753×10^{-1}	7.7116×10^{-1}

A.7 Large Figures

Larger versions (with numbers as annotations) of the subfigures in Fig. 3 are given in Figs. 1 and 2. Note that the divergence of final weights is not exactly zero but very small ($< 10^{-12}$), due to rounding errors.

A.8 Why Put $t = l$ Between $t = l_{\max} - 2$ and $t = l_{\max} - 1$

We put Z-IL ($t = l$) between $t = l_{\max} - 2$ and $t = l_{\max} - 1$ for models with different $l_{\max}$. This is due to the fact that Z-IL ($t = l$) is doing inference at a degree between $t = l_{\max} - 2$ and $t = l_{\max} - 1$. Specifically, Z-IL ($t = l$) uses $F_{l_{\max}-1}$ to update $\theta_{i,j}^{l_{\max}}$, which is a result from doing inference at $t = l_{\max} - 2$, however, Z-IL did not use $F_{l_{\max}-1}$ for weights update in all layers, instead it uses $F_{l<l_{\max}-1}$ at previous inference moments to update weights $\theta_{i,j}^{l}$ in other layers $l < l_{\max}$. Thus, we consider that Z-IL ($t = l$) lies between $t = l_{\max} - 2$ and $t = l_{\max} - 1$ from the perspective how much inference is conducted. For example, for $l_{\max} = 3$, we present the result as $t = \{0, 1, l, 2, \ldots\}$.

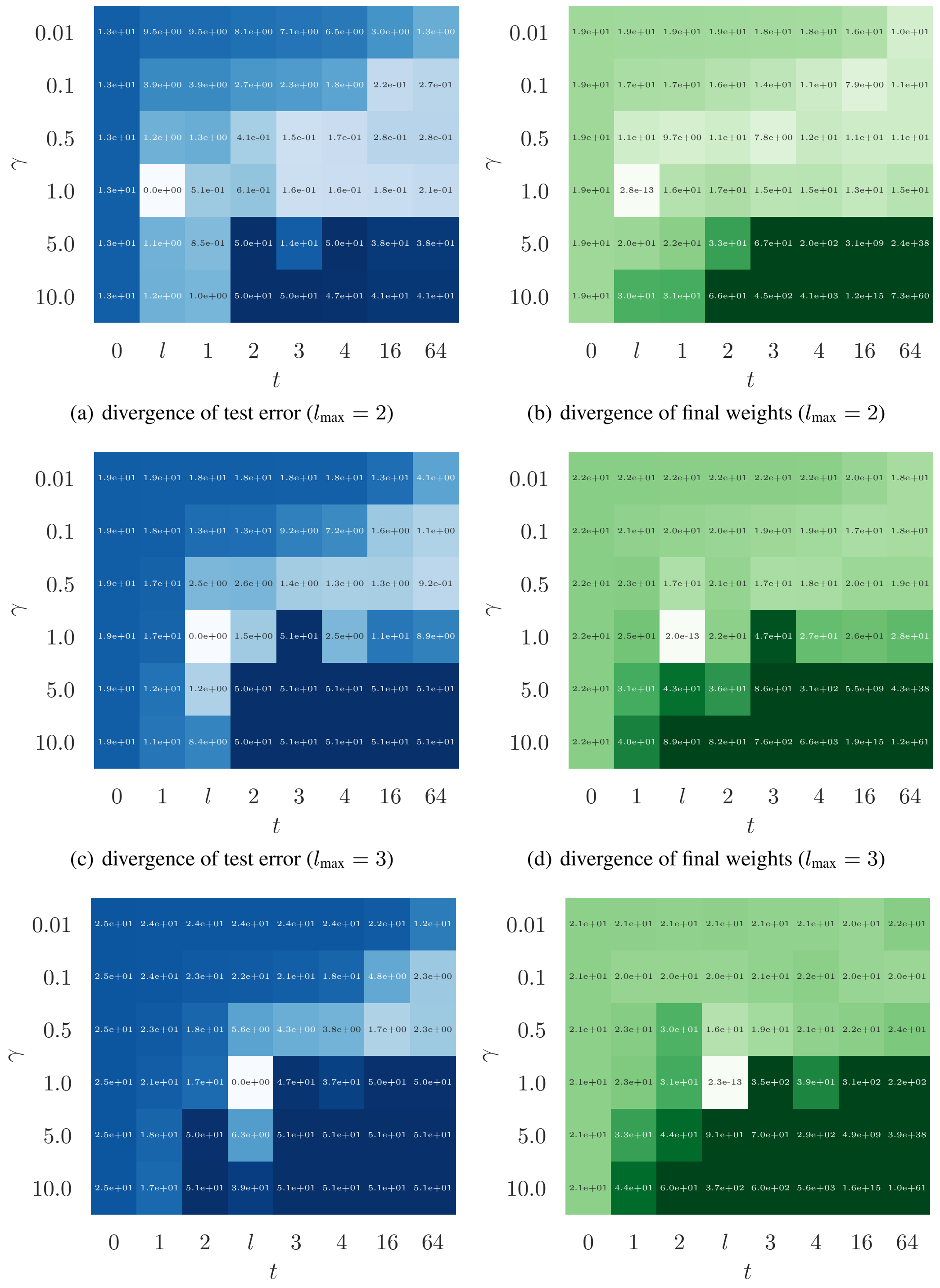
A.9 Detailed Setup of Experiments on ImageNet

Experiments are conducted on 2 GPUs of Nvidia GeForce GTX 1080Ti, and 8 CPUs of Intel Core i7, with 32 GB RAM. The batch size is set to 1 to test the running time. The input size of the image is 224×224 gray scale, we only implemented fully connected layers: the hidden layers are of size 4096, 2048, and 1024, respectively. The size of the output layer is 27, corresponding to the 27 high-level categories in ImageNet. All dependencies and their versions are specified as follows:

```
torchvision==0.4.2
setproctitle==1.1.10
psutil==5.6.3
requests==2.22.0
pillow==6.1
pandas==0.24.2
matplotlib==3.1.0
seaborn==0.9.0
numpy==1.16.4
tensorflow==1.15
pyinquirer==1.0.3
torch==1.3.1
ray==0.7.4
pprint==0.1
tabulate==0.8.7
opencv-python==4.2.0.34
PyInquirer==1.0.3
```

A.10 Test error of all situations

Note that in all our experiments, Fa-Z-IL produces the same test error as Z-IL, thus, no separated tables or figures for Fa-Z-IL are provided. Table 2 shows the test error of different σ.



(a) divergence of test error ($l_{\max} = 2$)

γ \ t	0	l	1	2	3	4	16	64
0.01	1.3e+01	9.5e+00	9.5e+00	8.1e+00	7.1e+00	6.5e+00	3.0e+00	1.3e+00
0.1	1.3e+01	3.9e+00	3.9e+00	2.7e+00	2.3e+00	1.8e+00	2.2e-01	2.7e-01
0.5	1.3e+01	1.2e+00	1.3e+00	4.1e-01	1.5e-01	1.7e-01	2.8e-01	2.8e-01
1.0	1.3e+01	0.0e+00	5.1e-01	6.1e-01	1.6e-01	1.6e-01	1.8e-01	2.1e-01
5.0	1.3e+01	1.1e+00	8.5e-01	5.0e+01	1.4e+01	5.0e+01	3.8e+01	3.8e+01
10.0	1.3e+01	1.2e+00	1.0e+00	5.0e+01	5.0e+01	4.7e+01	4.1e+01	4.1e+01

(b) divergence of final weights ($l_{\max} = 2$)

γ \ t	0	l	1	2	3	4	16	64
0.01	1.9e+01	1.9e+01	1.9e+01	1.9e+01	1.8e+01	1.8e+01	1.6e+01	1.0e+01
0.1	1.9e+01	1.7e+01	1.7e+01	1.6e+01	1.4e+01	1.1e+01	7.9e+00	1.1e+01
0.5	1.9e+01	1.1e+01	9.7e+00	1.1e+01	7.8e+00	1.2e+01	1.1e+01	1.1e+01
1.0	1.9e+01	2.8e-13	1.6e+01	1.7e+01	1.5e+01	1.5e+01	1.3e+01	1.5e+01
5.0	1.9e+01	2.0e+01	2.2e+01	3.3e+01	6.7e+01	2.0e+02	3.1e+09	2.4e+38
10.0	1.9e+01	3.0e+01	3.1e+01	6.6e+01	4.5e+02	4.1e+03	1.2e+15	7.3e+60

(c) divergence of test error ($l_{\max} = 3$)

γ \ t	0	1	l	2	3	4	16	64
0.01	1.9e+01	1.9e+01	1.8e+01	1.8e+01	1.8e+01	1.8e+01	1.3e+01	4.1e+00
0.1	1.9e+01	1.8e+01	1.3e+01	1.3e+01	9.2e+00	7.2e+00	1.6e+00	1.1e+00
0.5	1.9e+01	1.7e+01	2.5e+00	2.6e+00	1.4e+00	1.3e+00	1.3e+00	9.2e-01
1.0	1.9e+01	1.7e+01	0.0e+00	1.5e+00	5.1e+01	2.5e+00	1.1e+01	8.9e+00
5.0	1.9e+01	1.2e+01	1.2e+00	5.0e+01	5.1e+01	5.1e+01	5.1e+01	5.1e+01
10.0	1.9e+01	1.1e+01	8.4e+00	5.0e+01	5.1e+01	5.1e+01	5.1e+01	5.1e+01

(d) divergence of final weights ($l_{\max} = 3$)

γ \ t	0	1	l	2	3	4	16	64
0.01	2.2e+01	2.2e+01	2.2e+01	2.2e+01	2.2e+01	2.2e+01	2.0e+01	1.8e+01
0.1	2.2e+01	2.1e+01	2.0e+01	2.0e+01	1.9e+01	1.9e+01	1.7e+01	1.8e+01
0.5	2.2e+01	2.3e+01	1.7e+01	2.1e+01	1.7e+01	1.8e+01	2.0e+01	1.9e+01
1.0	2.2e+01	2.5e+01	2.0e-13	2.2e+01	4.7e+01	2.7e+01	2.6e+01	2.8e+01
5.0	2.2e+01	3.1e+01	4.3e+01	3.6e+01	8.6e+01	3.1e+02	5.5e+09	4.3e+38
10.0	2.2e+01	4.0e+01	8.9e+01	8.2e+01	7.6e+02	6.6e+03	1.9e+15	1.2e+61

(e) divergence of test error ($l_{\max} = 4$)

γ \ t	0	1	2	l	3	4	16	64
0.01	2.5e+01	2.4e+01	2.4e+01	2.4e+01	2.4e+01	2.4e+01	2.2e+01	1.2e+01
0.1	2.5e+01	2.4e+01	2.3e+01	2.2e+01	2.1e+01	1.8e+01	4.8e+00	2.3e+00
0.5	2.5e+01	2.3e+01	1.8e+01	5.6e+00	4.3e+00	3.8e+00	1.7e+00	2.3e+00
1.0	2.5e+01	2.1e+01	1.7e+01	0.0e+00	4.7e+01	3.7e+01	5.0e+01	5.0e+01
5.0	2.5e+01	1.8e+01	5.0e+01	6.3e+00	5.1e+01	5.1e+01	5.1e+01	5.1e+01
10.0	2.5e+01	1.7e+01	5.1e+01	3.9e+01	5.1e+01	5.1e+01	5.1e+01	5.1e+01

(f) divergence of final weights ($l_{\max} = 4$)

γ \ t	0	1	2	l	3	4	16	64
0.01	2.1e+01	2.1e+01	2.1e+01	2.1e+01	2.1e+01	2.1e+01	2.0e+01	2.2e+01
0.1	2.1e+01	2.0e+01	2.0e+01	2.0e+01	2.1e+01	2.2e+01	2.0e+01	2.0e+01
0.5	2.1e+01	2.3e+01	3.0e+01	1.6e+01	1.9e+01	2.1e+01	2.2e+01	2.4e+01
1.0	2.1e+01	2.3e+01	3.1e+01	2.3e-13	3.5e+02	3.9e+01	3.1e+02	2.2e+02
5.0	2.1e+01	3.3e+01	4.4e+01	9.1e+01	7.0e+01	2.9e+02	4.9e+09	3.9e+38
10.0	2.1e+01	4.4e+01	6.0e+01	3.7e+02	6.0e+02	5.6e+03	1.6e+15	1.0e+61

Figure 1: Larger versions (with numbers as annotations) of the subfigures in Fig. 3.

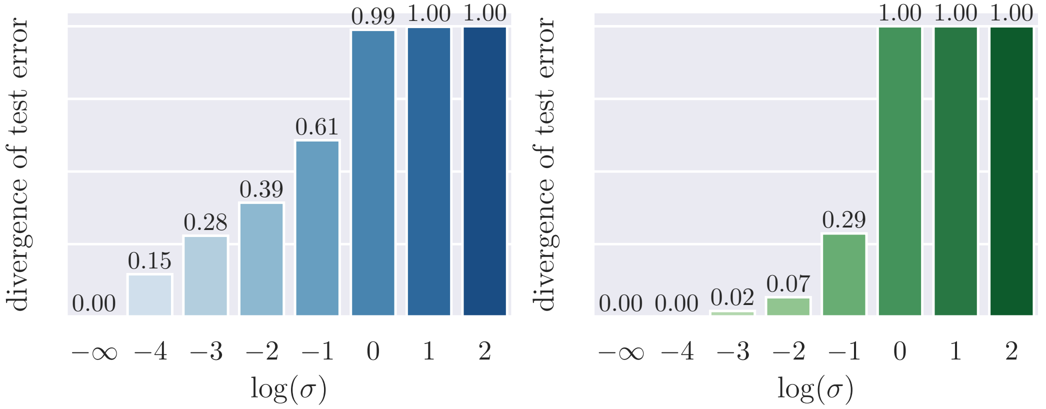
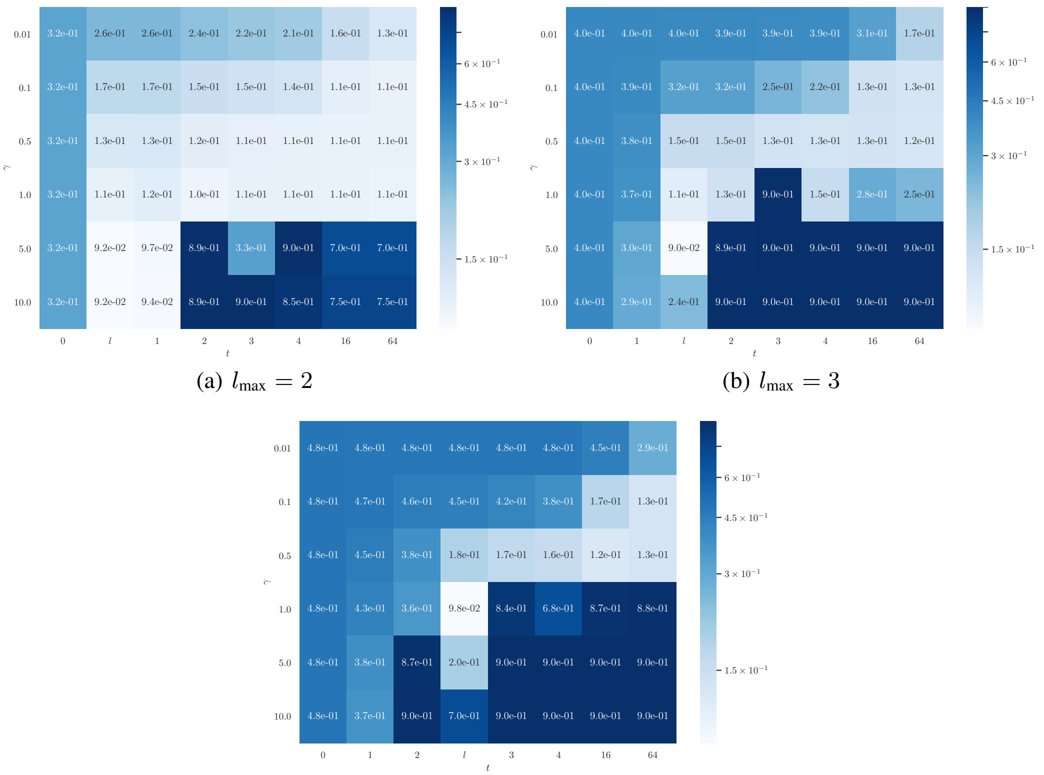


Figure 2: Larger versions (with numbers as annotations) of the subfigures in Fig. 3. For visualization, the divergence of test error has been normalized to $[0, 1]$ by a logarithmic scale and then clipped to $[0, 1]$; the divergence of the final weights has been normalized linearly to $[0, 1]$ from min of 0 and max of 50, values larger than 50 have been clipped to 50. The unscaled data are given in Table 1.



(a) $l_{\max} = 2$

(b) $l_{\max} = 3$

(c) $l_{\max} = 4$

Figure 3: Test error of different t and γ.

A.11 Differences of learning rules between [46] and Z-IL

The energy function of [46] is closely related to the Hopfield net (but with latent variables) or with the Boltzmann machine energy function (but with continuous nonlinearities inserted), best expressed with notations and setup in this paper as:

$$E_t = \textstyle\sum_i \frac{1}{2}(s_i)^2 - \frac{1}{2}\sum_{i \neq j} \eta_{i,j} f(s_i) f(s_j) - \sum_i b_i f(s_i), \tag{31}$$

where the support of s_i is given by all the value nodes in the neural network, $\eta_{i,j}$ is the synapse weight from node j to node i, b_i is the bias term, and f is the nonlinear activation function. While the energy function of Z-IL is simply the energy of the error nodes:

$$F_t = \textstyle\sum_{l=0}^{l_{\max}-1} \sum_{i=1}^{n^l} \frac{1}{2} (\varepsilon_{i,t}^{l})^2 \, . \tag{32}$$

The other learning rules thus diverge significantly from there. We leave the discussion of how the two works could be related regardless of such huge differences as future research.