

$$\begin{aligned}
L &:= L(z(t), t, \theta) \\
\dot{z}^\alpha(t) &:= f^\alpha(z(t), t, \theta) \\
\Rightarrow z^\alpha(t + \epsilon) &= z^\alpha(t) + \epsilon f^\alpha(z(t), t, \theta) \\
\Rightarrow \frac{\partial z^\beta(t + \epsilon)}{\partial z^\alpha(t)} &= \delta_\alpha^\beta + \epsilon \frac{\partial f^\beta}{\partial z^\alpha(t)}(z(t), t, \theta) \\
a_\alpha(t) &:= \frac{\partial L}{\partial z^\alpha(t)} \\
\frac{\partial L}{\partial z^\alpha(t)} &= \frac{\partial L}{\partial z^\beta(t + \epsilon)} \frac{\partial z^\beta(t + \epsilon)}{\partial z^\alpha(t)} \\
\Rightarrow a_\alpha(t) &= a_\beta(t + \epsilon) \frac{\partial z^\beta(t + \epsilon)}{\partial z^\alpha(t)} = [a_\beta(t) + \epsilon \dot{a}_\beta(t)] \left[ \delta_\alpha^\beta + \epsilon \frac{\partial f^\beta}{\partial z^\alpha(t)}(z(t), t, \theta) \right] \\
\Rightarrow \dot{a}_\alpha(t) &= -a_\beta(t) \frac{\partial f^\beta}{\partial z^\alpha(t)}(z(t), t, \theta)
\end{aligned}$$

## 1 Calculation of $\partial L / \partial \theta$

Suppose the model is layerized, the loss depends on the variables (inputs and model parameters) on the  $i$ th layer can be regarded as the loss of a **new** model by truncating the original at the  $i$ th layer, which we call  $L_i(z_i)$ . Varing  $\theta$  will vary the  $L_0(z_0)$  from two aspects, the effect from  $dL_1/d\theta$  and the  $\Delta z_1$  caused by  $\Delta\theta$ .

$$\frac{dL_0}{d\theta}(z_0) = \frac{dL_1}{d\theta}(z_1) + \frac{dL_1}{dz_1} \frac{\partial z_1}{\partial \theta}.$$

The same relation holds for any  $i$ , by simply considering a truncated model,

$$\frac{dL_i}{d\theta}(z_0) = \frac{dL_{i+1}}{d\theta}(z_{i+1}) + \frac{dL_{i+1}}{dz_{i+1}} \frac{\partial z_{i+1}}{\partial \theta}.$$

Thus we have, recursively,

$$\begin{aligned}
\frac{dL_0}{d\theta}(z_0) &= \frac{dL_1}{d\theta}(z_1) + \frac{dL_1}{dz_1} \frac{\partial z_1}{\partial \theta} \\
&= \left[ \frac{dL_2}{d\theta}(z_1) + \frac{dL_2}{dz_2} \frac{\partial z_2}{\partial \theta} \right] + \frac{dL_1}{dz_1} \frac{\partial z_1}{\partial \theta} \\
&= \frac{dL_2}{d\theta}(z_1) + \frac{dL_2}{dz_2} \frac{\partial z_2}{\partial \theta} + \frac{dL_1}{dz_1} \frac{\partial z_1}{\partial \theta} \\
&= \dots \\
&= \frac{dL_N}{d\theta}(z_1) + \sum_{i=1}^N \frac{dL_i}{dz_i} \frac{\partial z_i}{\partial \theta}.
\end{aligned}$$

By  $z_{i+1}(t) = z_i(t) + \epsilon f(z_i(t), t; \theta)$ ,  $\partial z_i / \partial \theta = \epsilon \partial f(z_i, t; \theta) / \partial \theta$

## 1 Continuum of Hopfield

### 1.1 Hopfield Network

Consider Hopfield network. Let  $x(t) \in \{-1, +1\}^N$  denotes the state of the network at discrete time  $t = 0, 1, \dots$ ; and  $W$  a matrix on  $\mathbb{R}^N$ , essentially ensuring  $W_{\alpha\beta} = W_{\beta\alpha}$  and  $W_{\alpha\alpha} = 0$ . Define energy  $E_W(x(t)) := -W_{\alpha\beta} x^\alpha(t) x^\beta(t)$ .

**Theorem 1.** Along dynamics  $x_\alpha(t+1) = \text{sign}[W_{\alpha\beta} x^\beta(t)]$ ,  $E_W(x(t+1)) - E_W(x(t)) \leq 0$ .

**Proof.** Consider the async-updation of Hopfield network. Let's change the component at dimension  $\hat{\alpha}$ , i.e.  $x'_{\hat{\alpha}} = \text{sign}[W_{\hat{\alpha}\beta} x^\beta]$ , then

$$\begin{aligned}
E_W(x') - E_W(x) &= -W_{\alpha\beta} x'^\alpha x'^\beta + W_{\alpha\beta} x^\alpha x^\beta \\
&= -2(x'^{\hat{\alpha}} - x^{\hat{\alpha}}) W_{\hat{\alpha}\beta} x^\beta,
\end{aligned}$$

which employs conditions  $W_{\alpha\beta} = W_{\beta\alpha}$  and  $W_{\alpha\alpha} = 0$ . Next, we prove that, combining with  $x'_{\hat{\alpha}} = \text{sign}[W_{\hat{\alpha}\beta}x^{\beta}]$ , this implies  $E_W(x') - E_W(x) \leq 0$ .

If  $(x'^{\hat{\alpha}} - x^{\hat{\alpha}}) > 0$ , then  $x'^{\hat{\alpha}} = 1$  and  $x^{\hat{\alpha}} = -1$ . Since  $x'_{\hat{\alpha}} = \text{sign}[W_{\hat{\alpha}\beta}x^{\beta}]$ ,  $W_{\hat{\alpha}\beta}x^{\beta} > 0$ . Then  $E_W(x') - E_W(x) < 0$ . Contrarily, if  $(x'^{\hat{\alpha}} - x^{\hat{\alpha}}) < 0$ , then  $x'^{\hat{\alpha}} = -1$  and  $x^{\hat{\alpha}} = 1$ , implying  $W_{\hat{\alpha}\beta}x^{\beta} < 0$ . Also  $E_W(x') - E_W(x) < 0$ . Otherwise,  $E_W(x') - E_W(x) = 0$ . So, we conclude  $E_W(x') - E_W(x) \leq 0$ .  $\square$

Since the states of the network are finite, the  $E_W$  is lower bounded. Thus the network converges (relaxes) at finite  $t$ .

## 1.2 Continuum

Let's consider applying the convergence of Hopfield network to neural ODE for generic network architecture. This makes the discrete time  $t$  a continuum.

**Theorem 2.** *Let  $M$  be a Riemann manifold with metric  $g$ . Given any function  $\mathcal{E} \in C^1(M, \mathbb{R})$ . Let  $x(t) \in C^1(\mathbb{R}, M)$  denote trajectory. Then,  $d\mathcal{E}/dt \leq 0$  along  $x(t)$  if*

$$\frac{dx^{\alpha}}{dt}(t) = -\nabla^{\alpha}\mathcal{E}(x(t)).$$

**Proof.** We have

$$\frac{d\mathcal{E}}{dt}(t) = \nabla_{\alpha}\mathcal{E}(x(t))\frac{dx^{\alpha}}{dt}(t) = -\nabla_{\alpha}\mathcal{E}(x(t))\nabla^{\alpha}\mathcal{E}(x(t)) \leq 0. \quad \square$$

Further, if the function  $\mathcal{E}$  is lower bounded, then the trajectory converges (relaxes) at finite  $t$ . We call this dynamic with lower bounded  $\mathcal{E}$  as ‘‘Hopfield dynamic with energy  $\mathcal{E}$ ’’.

This is the continuum analogy to the convergence of Hopfield network. Indeed, let  $M$  be  $\mathbb{R}^N$ , and  $\mathcal{E}(x) = -W_{\alpha\beta}x^{\alpha}x^{\beta}$  with  $W_{\alpha\beta} = W_{\beta\alpha}$ , then dynamics becomes

$$\frac{dx^{\alpha}}{dt}(t) = -\nabla_{\alpha}\mathcal{E}(x(t)) = -2W_{\alpha\beta}x^{\beta}(t),$$

which makes

$$\frac{d\mathcal{E}}{dt}(t) = -2\frac{dx^{\alpha}}{dt}(t)W_{\alpha\beta}x^{\beta}(t).$$

Comparing with the proof of convergence of Hopfield network, i.e.  $\Delta E_W(x) = -2\Delta x^{\alpha}W_{\alpha\beta}x^{\beta}$ , the analogy is obvious. The only differences are that the condition  $W_{\alpha\alpha} = 0$  and the sign-function are absent here.

It is known that a Riemann manifold can be locally re-coordinated to be Euclidean. Thus, locally  $\exists \hat{x}$  coordinate, s.t.

$$\frac{dx^{\alpha}}{dt}(t) = -\delta^{\alpha\beta}\frac{\partial\mathcal{E}}{\partial x^{\beta}}(x(t)).$$

## 1.3 General Form

In the proof of theorem 2,

$$\frac{d\mathcal{E}}{dt}(t) = \nabla_{\alpha}\mathcal{E}(x(t))\frac{dx^{\alpha}}{dt}(t).$$

We try to find the generic form of  $dx^{\alpha}/dt$  that ensures  $d\mathcal{E}/dt \leq 0$ . To restrict the formation, symmetries are called for. Denote

$$\frac{dx^{\alpha}}{dt} = F^{\alpha}[\mathcal{E}](x).$$

**Axiom 3.** *Locality.*

This implies:

$$F[\mathcal{E}] = F(\mathcal{E}, \nabla \mathcal{E}, \nabla^2 \mathcal{E}, \dots);$$

**Axiom 4.**  $\mathcal{E} \rightarrow \mathcal{E} + C$  for any constant  $C$ .

Combining with the previous, this then implies

$$F[\mathcal{E}] = F(\nabla \mathcal{E}, \nabla^2 \mathcal{E}, \dots).$$

**Axiom 5.** *Co-variance.*

This implies the balance of index. Thus

$$F^\alpha[\mathcal{E}] = c_1 \nabla^\alpha \mathcal{E} + c_3 \nabla^\alpha \mathcal{E} (\nabla^\beta \mathcal{E} \nabla_\beta \mathcal{E}) + c'_3 \nabla^\alpha \mathcal{E} (\nabla^\beta \nabla_\beta \mathcal{E}) + \mathcal{O}(\nabla^5).$$

**Axiom 6.** For  $x \rightarrow \lambda x$ ,  $\exists k < m < M < K$  s.t.  $m < F[\mathcal{E}](x) < M$  for any  $\lambda$ , where  $k$  and  $K$  are numerically finite. E.g.  $k \sim 1$  and  $K \sim 10$ . This is essential for numerical stability, i.e. no under- and over-flow.

First, we have to notice a property of the feed forward neural network with rectified activations (e.g. ReLU, leaky ReLU, and linear).

**Lemma 7.** Let  $f_{\text{nn}}(x; \theta)$  a feed forward neural network with rectified activations, where  $\theta$  represents the parameters (weights and biases). At the initial stage of training,  $f_{\text{nn}}(\cdot; \theta)$  is linearly homogeneous. That is,

$$f_{\text{nn}}(\lambda x; \theta_{\text{ini}}) \approx \lambda f_{\text{nn}}(x; \theta_{\text{ini}}).$$

**Proof.** Notice that  $f_{\text{nn}}(\cdot; \theta)$  is linearly homogeneous when its biases vanish, and that biases are initialized as zeros. So, at the initial stage of training, biases can be approximately omitted. Thus, in this stage,  $f_{\text{nn}}(\cdot; \theta)$  is approximately linearly homogeneous, i.e.  $f_{\text{nn}}(\lambda x; \theta_{\text{ini}}) \approx \lambda f_{\text{nn}}(x; \theta_{\text{ini}})$ .  $\square$

If  $\mathcal{E}$  is constructed by such neural network,  $F[\mathcal{E}]$  can be further simplified. Indeed, if  $\mathcal{E}(x; \theta) := \sqrt{f_\alpha(x; \theta) f^\alpha(x; \theta)}$ , then  $\mathcal{E}(\lambda x; \theta_{\text{ini}}) = \lambda \mathcal{E}(x; \theta_{\text{ini}})$ , implying  $F^\alpha[\mathcal{E}] = c_1 \nabla^\alpha \mathcal{E} + c_3 \nabla^\alpha \mathcal{E} (\nabla^\beta \mathcal{E} \nabla_\beta \mathcal{E}) + \mathcal{O}(\nabla^5)$ , which scales as  $\lambda^0$ .<sup>1</sup>

Alternatively, if  $\mathcal{E}(x; \theta) := f^2(x; \theta)$ , then  $\mathcal{E}(\lambda x; \theta_{\text{ini}}) = \lambda^2 \mathcal{E}(x; \theta_{\text{ini}})$ . In this case, axiom 6 can never be satisfied.

<sup>1</sup>. Numerical experiment on MNIST dataset shows that this configuration indeed out-performs than others, like  $\mathcal{E}(x; \theta) := f_\alpha(x; \theta) f^\alpha(x; \theta)$ ,  $\mathcal{E}(x; \theta) := f^2(x; \theta)$ , and non-Hopfield, e.t.c. In this experiment,  $c_1 = 5$  and  $c_{i>1} \equiv 0$ ; Nadam optimizer is employed, with standard parameters, except for  $\epsilon = 10^{-3}$ ; the dimension of  $x$  is 64. For the details, c.f. the file `note/experiments/Hopfield.ipynb`.