Appendix for "An Optimal Control Approach to Deep Learning and Applications to Discrete-Weight Neural Networks"

A Full Statement and Sketch of the Proof of Theorem 1

In this section, we give the full statement of Theorem 1 and a sketch of its proof as presented in [Halkin, 1966]. We note that in [Halkin, 1966], more general initial and final conditions are considered. For simplicity, we shall stick to the current formulation in the main text. We note also that the result presented here has been extended (in the sense that the convexity condition has been relaxed to directional convexity) [Holtzman, 1966, Holtzman and Halkin, 1966] and proven in different ways subsequently [Canon et al., 1970].

Before we begin, we simplify the notation by concatenating all the samples x_s into a large vector $x = (x_1, \ldots, x_S)$. The functions f_t are then redefined accordingly in the natural way. Moreover, we define the total loss function $\Phi(x) := \frac{1}{S}\sum_s \Phi_s(x_s)$ and the total regularization $L_t(x,\theta) = \frac{1}{S}\sum_s L_t(x_s,\theta)$. Consequently, we have the reformulated problem

$$\begin{aligned} &\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} J(\boldsymbol{\theta}) := \Phi(x_T) + \sum_{t=0}^{T-1} L_t(x_t,\theta_t) \\ &\text{subject to:} \\ &x_{t+1} = f_t(x_t,\theta_t), \quad t = 0,\ldots,T-1. \end{aligned} \tag{1}$$

We now make the following assumptions:

(B1) Φ is twice continuous differentiable.

(B2) $f_t(\cdot,\theta), L_t(\cdot,\theta)$ are twice continuously differentiable with respect to x, and $f_t(\cdot,\theta), L_t(\cdot,\theta)$ together with their x partial derivatives are uniformly bounded in t and θ.

(B3) The sets $\{f_t(x,\theta) : \theta \in \Theta_t\}$ and $\{L_t(x,\theta) : \theta \in \Theta_t\}$ are convex for every t and $x \in \mathbb{R}^{d_t}$.

The full statement of Theorem 1 is as follows:

Theorem A.1 (Discrete PMP, Full Statement)**.** *Let (B1)-(B3) be satisfied. Suppose that $\boldsymbol{\theta}^* := \{\theta_t^* : t = 0,\ldots,T-1\}$ is an optimal solution of* (1) *and $\boldsymbol{x}^* := \{x_t^* : t = 0,\ldots,T\}$ is the corresponding state process with $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. Then, there exists a co-state (or adjoint) process $\boldsymbol{p}^* := \{p_t^* : t = 0,\ldots,T\}$ and a real number $\beta \geq 0$ (abnormal multiplier) such that $\{\boldsymbol{p}^*, \beta\}$ are not all zero, and the following holds:*

$$x_{t+1}^* = \nabla_p H_t(x_t^*, p_{t+1}^*, \theta_t^*) \qquad x_0^* = x_0 \tag{2}$$

$$p_t^* = \nabla_x H_t(x_t^*, p_{t+1}^*, \theta_t^*) \qquad p_T^* = -\beta\nabla\Phi(x_T^*) \tag{3}$$

$$H_t(x_t^*, p_t^*, \theta_t^*) \geq H_t(x_t^*, p_t^*, \theta) \qquad \textit{for all } \theta \in \Theta_t \tag{4}$$

for $t = 0, 1, \ldots, T-1$, where the Hamiltonian function H is defined as

$$H_t(x,p,\theta) := p \cdot f_t(x,\theta) - \beta L_t(x,\theta).$$

Remark A.1. *Compared with the informal statement, the full statement involves an abnormal multiplier β. It exists to cover degenerate cases. This is related to "normality" in the calculus of variations [Bliss, 1938], or constraint qualification in the language of nonlinear programming [Kuhn and Tucker, 2014]. When it equals 0, the problem is degenerate. In applications we often focus on non-degenerate cases where β is positive, in which case we can normalize $\{p_t^*, \beta\}$ accordingly so that $\beta = 1$. We then obtain the informal statement in the main text.*

Sketch of the proof of Theorem A.1. To begin with, we may assume without loss of generality that $L \equiv 0$. To see why this is so, we define an extra scalar variable w_t with

$$w_{t+1} = w_t + L_t(x_t,\theta_t), \quad w_0 = 0.$$

We then append w to x to form the new $(d_t + 1)$-dimensional state vector (x, w). Accordingly, we modify $f_t(x, \theta)$ to $(f_t(x, \theta), w + L_t(x, \theta))$ and $\Phi(x)$ to $\Phi(x) + w$. It is clear that all assumptions (B1)-(B3) are preserved.

As in the main text, we define the set of reachable states by the original dynamical system

$$W_t := \{x \in \mathbb{R}^{d_t} : \exists \boldsymbol{\theta} \text{ s.t. } x_t^{\boldsymbol{\theta}} = x\} \tag{5}$$

where $x_t^{\boldsymbol{\theta}}$ is the evolution of the dynamical system for x_t under $\boldsymbol{\theta}$. This is basically the set of all states that the system can reach under "some" control at time t. Let $\{\boldsymbol{x}^*, \boldsymbol{\theta}^*\}$ be a pair of optimal solutions of (1). Let us define the set of all final states with lower loss value than the optimum as

$$S := \{x \in \mathbb{R}^{d_T} : \Phi(x) < \Phi(x_T^*)\}. \tag{6}$$

Then, it is clear that W_T and B are disjoint. Otherwise, $\{\boldsymbol{x}^*, \boldsymbol{\theta}^*\}$ would not have been optimal. Now, if W_T and B are convex, then one can then use separation properties of convex sets to prove the theorem. However, in general they are non-convex (even if (B3) is satisfied). The idea is to consider the following linearized problem

$$\begin{aligned} &\psi_{t+1} = f_t(x_t^*, \theta_t) + \nabla_x f_t(x_t^*, \theta_t^*)(\psi_t - x_t^*), \quad t = 0, 1, \dots, T-1 \\ &\psi_0 = x_0 \end{aligned} \tag{7}$$

Then, we can similarly define the counter-parts to W_t and S as

$$W_t^+ := \{x \in \mathbb{R}^{d_t} : \exists \boldsymbol{\theta} \text{ s.t. } \psi_t^{\boldsymbol{\theta}} = x\} \tag{8}$$

and

$$S^+ := \{x \in \mathbb{R}^{d_T} : (x - x_T^*) \cdot \nabla\Phi(x_T^*) < 0\}. \tag{9}$$

It is clear that the sets W_T^+ and S^+ are both convex. In [Halkin, 1966], the author proves an important linearization lemma that says: **if W_T and S are disjoint, then W_T^+ and S^+ are separated**, i.e. there exists a non-zero vector $\pi \in \mathbb{R}^{d_T}$ such that

$$(x - x_T^*) \cdot \pi \leq 0 \qquad x \in W_T^+ \tag{10}$$

$$(x - x_T^*) \cdot \pi \geq 0 \qquad x \in S^+ \tag{11}$$

Here, π is the normal of a separating hyper-plane of the convex sets W_T^+ and S^+. In fact, one can show that $\pi = -\beta\nabla\Phi(x_T^*)$ for some $\beta \geq 0$. We note here that the linearization lemma, i.e. the separation of W_T^+ and S^+, forms the bulk of the proof of the theorem in [Halkin, 1966]. The proof relies on topological properties of non-separated convex sets. We shall omit its proof here and refer the reader to [Halkin, 1966].

Now, we may define $p_T^* = \pi$, and for $t \leq T$, set

$$p_t^* = \nabla_x H_t(x_t^*, p_{t+1}^*, \theta_t^*) = \nabla_x f(x_t^*, \theta_t^*)^T p_{t+1}^*. \tag{12}$$

In other words, p_t^* evolves the normal π of the separating hyper-plane of W_T^+ and S^+ backwards in time. An important property one can check is that p_t^* and ψ_t (defined by Eq. (12) and (7)) are adjoint of each other at the optimum, i.e. if $\theta_t = \theta_t^*$, then we have

$$(\psi_{t+1} - x_{t+1}^*) \cdot p_{t+1}^* = (\psi_t - x_t^*) \cdot p_t^*. \tag{13}$$

This fact allows one to prove the Hamiltonian maximization condition (4). Indeed, suppose that for some $t \in \{0, \dots, T-1\}$ the condition is violated, i.e. there exists $\tilde{\theta} \in \Theta_t$ such that

$$H_t(x_t^*, p_{t+1}^*, \tilde{\theta}) = H_t(x_t^*, p_{t+1}^*, \theta_t^*) + \epsilon$$

for some $\epsilon > 0$. This means

$$p_{t+1}^* \cdot f_t(x_t^*, \tilde{\theta}) = p_{t+1}^* \cdot f_t(x_t^*, \theta_t^*) + \epsilon$$

i.e.,

$$p_{t+1}^* \cdot (f_t(x_t^*, \tilde{\theta}) - x_{t+1}^*) = \epsilon$$

Now, we simply evolve ψ_s, $s \geq t+1$ with $\theta_s = \theta_s^*$ but the initial condition $\psi_{t+1} = f_t(x_t^*, \tilde{\theta})$. Then, Eq. (13) implies that $\pi \cdot (\psi_T - x_T^*) \cdot = \epsilon > 0$, but this contradicts (10). □

Remark A.2. *Note that in the original proof [Halkin, 1966], it is also assumed that $\nabla_x f_t$ is non-singular, which also forces $d_t = d$ to be constant for all t. This is obviously not satisfied naturally by most neural networks that have changing dimensions. However, one can check that this condition only serves to ensure that if $p_T^* \neq 0$, then $p_t^* \neq 0$ for all $t = 0, \dots, T-1$. Hence, without this assumption, we can only be sure that not all $\{\boldsymbol{p}^*, \beta\}$ are 0.*

A.1 The Convexity Condition for Neural Networks

As also discussed in the main text, the most stringent condition in Theorem A.1 is the convexity condition for f_t, i.e. the set $\{f_t(x, \theta) : \theta \in \Theta\}$ must be convex. It is easy to see that for the usual feed-forward neural networks, one can decompose it in such a way that the convexity constraint is satisfied as long as the parameter sets Θ_t are convex. Indeed, we have

$$x_{t+1} = \sigma(g_t(x_t, \theta_t))$$

where σ is some non-trainable nonlinear activation function and g_t is affine in θ. We can simply decompose this into two steps

$$\begin{aligned} x'_{t+1} &= g_t(x_t, \theta_t), \\ x'_{t+2} &= \sigma(x'_{t+1}). \end{aligned}$$

Then, $x'_{t+2} = x_{t+1}$ but each of these two steps now satisfy the convexity constraint.

Similarly, in residual networks, we can usually write the layer transformation as

$$x_{t+1} = x_t + h_t(\sigma(g_t(x_t, \theta_t)), \phi_t)$$

where g_t, h_t are maps affine in θ and ϕ respectively, and σ is a non-trainable non-linearity. The above cannot be straightforwardly split into two layers as there is a shortcut connection from x_n. However, we can introduce auxiliary variables y_t and consider the 3-step decomposition

$$\begin{aligned} x'_{t+1} &= g_t(x_t, \theta_t) & y'_{t+1} &= x_t, \\ x'_{t+2} &= \sigma(x'_{t+1}) & y'_{t+2} &= y'_{t+1}, \\ x'_{t+3} &= y'_{t+2} + h_t(x'_{t+2}, \phi_t) & y'_{t+3} &= y'_{t+2}. \end{aligned}$$

It is clear then that x'_{t+3} is equal to x_{t+1} in the residual network layer. Furthermore, this new decomposed system satisfy the convexity assumption as long as Θ_t is a convex set.

B Proof of Theorem 2

In this section, we prove Theorem 2 in the main text using some elementary estimates. Let us first prove a useful result.

Lemma B.1 (Discrete Gronwall's Lemma)**.** *Let $K \geq 0$ and u_t, w_t, be non-negative real valued sequences satisfying*

$$u_{t+1} \leq K u_t + w_t,$$

for $t = 0, \ldots, T-1$. Then, we have for all $t = 0, \ldots, T$,

$$u_t \leq \max(1, K^T) \left(u_0 + \sum_{s=0}^{T-1} w_s \right).$$

Proof. We prove by induction the inequality

$$u_t \leq \max(1, K^t) \left(u_0 + \sum_{s=0}^{t-1} w_s \right), \tag{14}$$

from which the lemma follows immediately. The case $t = 0$ is trivial. Suppose the above is true for some t, we have

$$\begin{aligned} u_{t+1} &\leq K u_t + w_t \\ &\leq K \max(1, K^t) \left(u_0 + \sum_{s=0}^{t-1} w_s \right) + w_t \\ &\leq \max(1, K^{t+1}) \left(u_0 + \sum_{s=0}^{t-1} w_s \right) + \max(1, K^{t+1}) w_t \\ &= \max(1, K^{t+1}) \left(u_0 + \sum_{s=0}^{t} w_s \right). \end{aligned}$$

This proves (14) and hence the lemma. □

Let us now commence the proof of a preliminary lemma that estimates the magnitude of $\boldsymbol{p}_s^{\boldsymbol{\theta}}$ for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Hereafter, C will be stand for any generic constant that does not depend on $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and S (batch size), but may depend on other fixed quantities such as T and the Lipschitz constants K in (A1)-(A2). Also, the value of C is allowed to change to another constant value with the same dependencies from line to line in order to reduce notational clutter.

Lemma B.2. *There exists a constant $C > 0$ such that for each $t = 0, \dots, T$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, we have*

$$\|p_{s,t}^{\boldsymbol{\theta}}\| \leq \frac{C}{S}.$$

for all $s = 1, \dots, S$.

Proof. First, notice that $p_{s,T}^{\boldsymbol{\theta}} = -\frac{1}{S}\nabla\Phi_s(x_{s,T}^{\boldsymbol{\theta}})$ and so by assumption (A1), we have

$$\|p_{s,T}^{\boldsymbol{\theta}}\| = \frac{1}{S}\|\nabla\Phi_s(x_{s,T}^{\boldsymbol{\theta}})\| \leq \frac{K}{S}.$$

Now, for each $0 \leq t < T$, we have by Eq. (8) and assumption (A2) in the main text,

$$\begin{aligned}\|p_{s,t}^{\boldsymbol{\theta}}\| =& \|\nabla_x H_t(x_{s,t}^{\boldsymbol{\theta}}, p_{s,t+1}^{\boldsymbol{\theta}}, \theta_t)\| \\ \leq& \|\nabla_x f_t(x_{s,t}^{\boldsymbol{\theta}}, \theta_t)^T p_{s,t+1}^{\boldsymbol{\theta}}\| + \frac{1}{S}\nabla_x\|L_t(x_{s,t}^{\boldsymbol{\theta}}, \theta_t)\| \\ \leq& K\|p_{s,t+1}^{\boldsymbol{\theta}}\| + \frac{K}{S}\end{aligned}$$

Using Lemma B.1 with $t \to T - t$, we get

$$\|p_{s,t}^{\boldsymbol{\theta}}\| \leq \max(1, K^T)(\frac{K}{S} + \frac{TK}{S}) = \frac{C}{S}.$$

□

We are now ready to prove Theorem 2.

Proof of Theorem 2. Recall the definition

$$H_t(x, p, \theta) = p \cdot f_t(x, \theta) - \frac{1}{S}L_t(x, \theta).$$

Let us define the quantity

$$I(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{\theta}) := \sum_{t=0}^{T-1} p_{t+1} \cdot x_{t+1} - H_t(x_t, p_{t+1}, \theta_t) - L_t(x_t, \theta_t)$$

Then, from Eq. (7) from the main text, we know that $I(\boldsymbol{x}_s^{\boldsymbol{\theta}}, \boldsymbol{p}_s^{\boldsymbol{\theta}}, \boldsymbol{\theta}) = 0$ for any $s = 1, \dots, S$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Let us now fix some sample s and obtain corresponding estimates. We have

$$\begin{aligned}0 =& I(\boldsymbol{x}_s^{\boldsymbol{\phi}}, \boldsymbol{p}_s^{\boldsymbol{\phi}}, \boldsymbol{\phi}) - I(\boldsymbol{x}_s^{\boldsymbol{\theta}}, \boldsymbol{p}_s^{\boldsymbol{\theta}}, \boldsymbol{\theta}) \\ =& \sum_{t=0}^{T-1} p_{s,t+1}^{\boldsymbol{\phi}} \cdot x_{s,t+1}^{\boldsymbol{\phi}} - p_{s,t+1}^{\boldsymbol{\theta}} \cdot x_{s,t+1}^{\boldsymbol{\theta}} \\ & - \frac{1}{S}\sum_{t=0}^{T-1} L_t(x_{s,t}^{\boldsymbol{\phi}}, \phi_t) - L_t(x_{s,t}^{\boldsymbol{\theta}}, \theta_t) \\ & - \sum_{t=0}^{T-1} H_t(x_{s,t}^{\boldsymbol{\phi}}, p_{s,t+1}^{\boldsymbol{\phi}}, \phi_t) - H_t(x_{s,t}^{\boldsymbol{\theta}}, p_{s,t+1}^{\boldsymbol{\theta}}, \phi_t)\end{aligned} \tag{15}$$

We can rewrite the first term on the right hand side as

$$\begin{aligned}&\sum_{t=0}^{T-1} p_{s,t+1}^{\boldsymbol{\phi}} \cdot x_{s,t+1}^{\boldsymbol{\phi}} - p_{s,t+1}^{\boldsymbol{\theta}} \cdot x_{s,t+1}^{\boldsymbol{\theta}} \\ =& \sum_{t=0}^{T-1} p_{s,t+1}^{\boldsymbol{\theta}} \cdot \delta x_{s,t+1} + x_{s,t+1}^{\boldsymbol{\theta}} \cdot \delta p_{s,t+1} + \delta x_{s,t+1} \cdot \delta p_{s,t+1},\end{aligned} \tag{16}$$

where we have defined $\delta x_{s,t} := x^{\boldsymbol{\phi}}_{s,t} - x^{\boldsymbol{\theta}}_{s,t}$ and $\delta p_{s,t} := p^{\boldsymbol{\phi}}_{s,t} - p^{\boldsymbol{\theta}}_{s,t}$. We may simplify further by observing that $\delta x_{s,0} = 0$, and so

$$\begin{aligned}\sum_{t=0}^{T-1} p^{\boldsymbol{\theta}}_{s,t+1} \cdot \delta x_{s,t+1} + x^{\boldsymbol{\theta}}_{s,t+1} \cdot \delta p_{s,t+1} =& p^{\boldsymbol{\theta}}_{s,T} \cdot \delta x_{s,T} + \sum_{t=0}^{T-1} p^{\boldsymbol{\theta}}_{s,t} \cdot \delta x_{s,t} + x^{\boldsymbol{\theta}}_{s,t+1} \cdot \delta p_{s,t+1} \\ =& p^{\boldsymbol{\theta}}_{s,T} \cdot \delta x_{s,T} + \sum_{t=0}^{T-1} \nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t) \cdot \delta x_{s,t} \\ &+ \sum_{t=0}^{T-1} \nabla_p H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t) \cdot \delta p_{s,t+1}\end{aligned}$$

By defining the extended vector $z^{\boldsymbol{\theta}}_{s,t} := (x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1})$, we can rewrite this as

$$\sum_{t=0}^{T-1} p^{\boldsymbol{\theta}}_{s,t+1} \cdot \delta x_{s,t+1} + x^{\boldsymbol{\theta}}_{s,t+1} \cdot \delta p_{s,t+1} = p^{\boldsymbol{\theta}}_{s,T} \cdot \delta x_{s,T} + \sum_{t=0}^{T-1} \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t) \cdot \delta z_{s,t} \tag{17}$$

Similarly, we also have

$$\begin{aligned}\sum_{t=0}^{T-1} \delta x_{s,t+1} \cdot \delta p_{s,t+1} =& \frac{1}{2} \sum_{t=0}^{T-1} \delta x_{s,t+1} \cdot \delta p_{s,t+1} + \frac{1}{2} \sum_{t=0}^{T-1} \delta x_{s,t+1} \cdot \delta p_{s,t+1} \\ =& \frac{1}{2} \delta x_{s,T} \cdot \delta p_{s,T} \\ &+ \frac{1}{2} \sum_{t=0}^{T-1} (\nabla_z H_t(z^{\boldsymbol{\phi}}_{s,t}, \phi_t) - \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t)) \cdot \delta z_{s,t} \\ =& \frac{1}{2} \delta x_{s,T} \cdot \delta p_{s,T} \\ &+ \frac{1}{2} \sum_{t=0}^{T-1} (\nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t)) \cdot \delta z_{s,t} \\ &+ \frac{1}{2} \sum_{t=0}^{T-1} \delta z_{s,t} \cdot \nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_1(t) \delta z_{s,t}, \phi_t) \delta z_{s,t}\end{aligned} \tag{18}$$

where in the last line we used Taylor's theorem with $r_1(t) \in [0, 1]$ for each t. Now, we can rewrite the terminal terms (i.e. T terms) in (17) and (18) as follows:

$$\begin{aligned}&(p^{\boldsymbol{\theta}}_{s,T} + \frac{1}{2} \delta p_{s,T}) \cdot \delta x_{s,T} \\ =& -\frac{1}{S} \nabla \Phi_s(x^{\boldsymbol{\theta}}_{s,T}) \cdot \delta x_{s,T} - \frac{1}{2S} (\nabla \Phi_s(x^{\boldsymbol{\phi}}_{s,T}) - \nabla \Phi_s(x^{\boldsymbol{\theta}}_{s,T})) \cdot \delta x_{s,T} \\ =& -\frac{1}{S} \nabla \Phi_s(x^{\boldsymbol{\theta}}_{s,T}) \cdot \delta x_{s,T} - \frac{1}{2S} \delta x_{s,T} \cdot \nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_2 \delta x_{s,T}) \delta x_{s,T} \\ =& -\frac{1}{S} (\Phi_s(x^{\boldsymbol{\phi}}_{T}) - \Phi_s(x^{\boldsymbol{\theta}}_{T})) - \frac{1}{2S} \delta x_{s,T} \cdot [\nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_2 \delta x_{s,T}) + \nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_3 \delta x_{s,T})] \delta x_{s,T}\end{aligned} \tag{19}$$

for some $r_2, r_3 \in [0, 1]$. Lastly, for each $t = 0, 1, \ldots, T-1$ we have

$$\begin{aligned}H_t(z^{\boldsymbol{\phi}}_{s,t}, \phi_t) - H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t) =& H_t(z^{\boldsymbol{\theta}}_{s,t}, \phi_t) - H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t) \\ &+ \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \phi_t) \cdot \delta z_{s,t} \\ &+ \frac{1}{2} \delta z_{s,t} \cdot \nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_4(t) \delta z_{s,t}, \phi_t) \delta z_{s,t}\end{aligned} \tag{20}$$

where $r_4(t) \in [0, 1]$.

Substituting Eq. (16, 17, 18, 19, 20) into Eq. (15) yields

$$
\begin{aligned}
&\frac{1}{S}\left[\Phi_s(x^{\boldsymbol{\phi}}_{s,T}) + \sum_{t=0}^{T-1} L_t(x^{\boldsymbol{\phi}}_{s,t}, \phi_t)\right] - \frac{1}{S}\left[\Phi_s(x^{\boldsymbol{\theta}}_{s,T}) + \sum_{t=0}^{T-1} L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\right] \\
=& -\sum_{t=0}^{T-1} H_t(x^{\boldsymbol{\theta}}_t, p^{\boldsymbol{\theta}}_{t+1}, \phi_t) - H_t(x^{\boldsymbol{\theta}}_t, p^{\boldsymbol{\theta}}_{t+1}, \theta_t) \\
&+ \frac{1}{2S}\delta x_{s,T} \cdot (\nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_2 \delta x_{s,T}) + \nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_3 \delta x_{s,T}))\delta x_{s,T} \\
&+ \frac{1}{2}\sum_{t=0}^{T-1} (\nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t)) \cdot \delta z_{s,t} \\
&+ \frac{1}{2}\sum_{t=0}^{T-1} \delta z_{s,t} \cdot (\nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_1(t)\delta z_{s,t}, \phi_t) - \nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_4(t)\delta z_{s,t}, \phi_t))\delta z_{s,t}
\end{aligned}
\tag{21}
$$

Note that by summing over all s, the left hand side is simply $J(\boldsymbol{\phi}) - J(\boldsymbol{\theta})$. Let us further simplify the right hand side. First, by (A1), we have

$$
\delta x_{s,T} \cdot (\nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_2 \delta x_{s,T}) + \nabla^2 \Phi_s(x^{\boldsymbol{\theta}}_{s,T} + r_3 \delta x_{s,T}))\delta x_{s,T} \leq K\|\delta x_{s,T}\|^2. \tag{22}
$$

Next,

$$
\begin{aligned}
&(\nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_z H_t(z^{\boldsymbol{\theta}}_{s,t}, \theta_t)) \cdot \delta z_{s,t} \\
\leq& \|\nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \phi_t) - \nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t)\|\|\delta x_{s,t}\| \\
&+ \|\nabla_p H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \phi_t) - \nabla_p H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t)\|\|\delta p_{s,t+1}\| \\
\leq& \frac{1}{2S}\|\delta x_{s,t}\|^2 + \frac{S}{2}\|\nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \phi_t) - \nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t)\|^2 \\
&+ \frac{S}{2}\|\delta p_{s,t}\|^2 + \frac{1}{2S}\|\nabla_p H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \phi_t) - \nabla_p H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t)\|^2 \\
\leq& \frac{1}{2S}\|\delta x_{s,t}\|^2 + \frac{C^2}{2S}\|\nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2 \\
&+ \frac{1}{2S}\|\nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2 \\
&+ \frac{S}{2}\|\delta p_{s,t}\|^2 + \frac{1}{2S}\|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2,
\end{aligned}
\tag{23}
$$

where in the last line we have used Lemma B.2. Similarly, we can simplify the last term in (21). Notice that the second derivative of H_t with respect to p vanishes since it is linear. Hence, as in Eq. (22) and using Lemma B.2, we have

$$
\begin{aligned}
&\delta z_{s,t} \cdot (\nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_1(t)\delta z_{s,t}, \phi_t) - \nabla^2_z H_t(z^{\boldsymbol{\theta}}_{s,t} + r_4(t)\delta z_{s,t}, \phi_t))\delta z_{s,t} \\
\leq& \frac{2KC}{S}\|\delta x_{s,t}\|^2 + 4K\|\delta x_{s,t}\|\|\delta p_{s,t+1}\| \\
\leq& \frac{2KC}{S}\|\delta x_{s,t}\|^2 + \frac{2K}{S}\|\delta x_{s,t}\|^2 + 2KS\|\delta p_{s,t+1}\|^2
\end{aligned}
\tag{24}
$$

Substituting Eq. (22,23,24) into (21) and summing over s, we have (renaming constants)

$$
\begin{aligned}
&\frac{1}{S}\left[\Phi_s(x^{\boldsymbol{\phi}}_{s,T}) + \sum_{t=0}^{T-1} L_t(x^{\boldsymbol{\phi}}_{s,t}, \phi_t)\right] - \frac{1}{S}\left[\Phi_s(x^{\boldsymbol{\theta}}_{s,T}) + \sum_{t=0}^{T-1} L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\right] \\
&= - \sum_{t=0}^{T-1} H_t(x^{\boldsymbol{\theta}}_t, p^{\boldsymbol{\theta}}_{t+1}, \phi_t) - H_t(x^{\boldsymbol{\theta}}_t, p^{\boldsymbol{\theta}}_{t+1}, \theta_t) \\
&\quad + \frac{C}{S} \sum_{t=0}^{T} \|\delta x_{s,t}\|^2 + CS \sum_{t=0}^{T-1} \|\delta p_{s,t+1}\|^2 \\
&\quad + \frac{C}{S} \sum_{t=0}^{T-1} \|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2 \\
&\quad + \frac{C}{S} \sum_{t=0}^{T-1} \|\nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2 \\
&\quad + \frac{C}{S} \sum_{t=0}^{T-1} \|\nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|^2
\end{aligned} \tag{25}
$$

It remains to estimate the magnitudes of $\delta x_{s,t}$ and $\delta p_{s,t}$. Observe that $\delta x_{s,0} = 0$, hence we have for each $t = 0, \dots, T-1$

$$
\begin{aligned}
\|\delta x_{s,t+1}\| \leq& \|f_t(x^{\boldsymbol{\phi}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t)\| + \|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\| \\
\leq& K\|\delta x_{s,t}\| + \|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|
\end{aligned}
$$

Using Lemma B.1, we have

$$
\|\delta x_{s,t}\| \leq C \sum_{t=0}^{T-1} \|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\| \tag{26}
$$

Similarly,

$$
\begin{aligned}
\|\delta p_{s,t}\| \leq& \|\nabla_x H_t(x^{\boldsymbol{\phi}}_{s,t}, p^{\boldsymbol{\phi}}_{s,t+1}, \phi_t) - \nabla_x H_t(x^{\boldsymbol{\theta}}_{s,t}, p^{\boldsymbol{\theta}}_{s,t+1}, \theta_t)\| \\
\leq& 2K\|\delta p_{s,t+1}\| + \frac{C}{S}\|\delta x_{s,t}\| \\
&+ \frac{C}{S}\|\nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|_2 \\
&+ \frac{C}{S}\|\nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|,
\end{aligned}
$$

and so by Lemma B.1, Eq. (26) and the fact that $\|\delta p_{T,s}\| \leq \frac{K}{S}\|\delta x_{T,s}\|$ (by (A1)), we have

$$
\begin{aligned}
\|\delta p_{s,t}\| \leq& \frac{C}{S} \sum_{t=0}^{T-1} \|f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\| \\
&+ \frac{C}{S} \sum_{t=0}^{T-1} \|\nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x f_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|_2 \\
&+ \frac{C}{S} \sum_{t=0}^{T-1} \|\nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \phi_t) - \nabla_x L_t(x^{\boldsymbol{\theta}}_{s,t}, \theta_t)\|.
\end{aligned} \tag{27}
$$

Finally, we conclude the proof of Theorem 2 by substituting estimates (26) and (27) into (25) and summing over s. □

C Gradient Descent with Back-propagation as a modification of MSA

Here we show that the classical gradient-descent algorithm where the gradients are computed using back-propagation [LeCun, 1988] is a modification of the MSA. This was originally discussed in [Li et al., 2018]. As discussed in the main paper, the reason MSA may diverge is if the arg-max step is too drastic such that the non-negative penalty terms dominate. One simple way is to make the arg-max step infinitesimal, in the appropriate direction, provided such updates provide feasible solutions. In other words, if we assume differentiability with respect to θ for all f_t and that Θ_t is the whole Euclidean space, we may substitute the arg-max step with a steepest ascent step

$$\theta_t^1 = \theta_t^0 + \eta \nabla_\theta \sum_{s=1}^{S} H_t(x_{s,t}^{\boldsymbol{\theta}^0}, p_{s,t+1}^{\boldsymbol{\theta}^0}, \theta_t^0), \tag{28}$$

for small small learning rate $\eta > 0$. We show the following:

Proposition C.1. *The MSA (Alg. 1 in the main text) with the maximization step replaced by* (28) *is equivalent to gradient-descent with back-propagation on J.*

Proof. As in Appendix A, WLOG we can assume $L \equiv 0$ by redefining coordinates. We have the following form for the Hamiltonian of the sample s

$$H_t(x_{s,t}^{\boldsymbol{\theta}}, p_{s,t+1}^{\boldsymbol{\theta}}, \theta_t) = p_{s,t+1}^{\boldsymbol{\theta}} \cdot f(x_{s,t}^{\boldsymbol{\theta}}, \theta_t),$$

and the total loss function is $J(\boldsymbol{\theta}) = \frac{1}{S}\sum_{s=1}^{S} \Phi_s(x_{s,T}^{\boldsymbol{\theta}})$. It is easy to see that $p_{s,t}^{\boldsymbol{\theta}} = -\frac{1}{S}\nabla_{x_{s,t}^{\boldsymbol{\theta}}} \Phi_s(x_{s,T}^{\boldsymbol{\theta}})$ (here $\nabla_{x_{s,t}^{\boldsymbol{\theta}}}$ is the total derivative) by working backwards from $t = T$ and the fact that $\nabla_{x_{s,t}^{\boldsymbol{\theta}}} x_{s,t+1}^{\boldsymbol{\theta}} = \nabla_x f_t(x_{s,t}^{\boldsymbol{\theta}}, \theta_t)$. Hence,

$$\begin{aligned}
\nabla_{\theta_t} J(\boldsymbol{\theta}) =& \frac{1}{S}\sum_{s=1}^{S} \nabla_{x_{s,t+1}^{\boldsymbol{\theta}}} \Phi_s(x_{s,T}^{\boldsymbol{\theta}}) \cdot \nabla_{\theta_t} x_{s,t+1}^{\boldsymbol{\theta}} \\
=& \sum_{s=1}^{S} -p_{s,t+1}^{\boldsymbol{\theta}} \cdot \nabla_{\theta_t} f_t(x_{s,t}^{\boldsymbol{\theta}}, \theta_t) \\
=& -\nabla_\theta \sum_{s=1}^{S} H_t(x_{s,t}^{\boldsymbol{\theta}}, p_{s,t+1}^{\boldsymbol{\theta}}, \theta_t)
\end{aligned}$$

Hence, (28) is simply the gradient descent step

$$\theta_t^{k+1} = \theta_t^k - \eta \nabla_{\theta_t} J(\boldsymbol{\theta}^k).$$

□

Thus, we have shown that the classical gradient descent algorithm constitute a modification of the MSA where the arg-max step is replaced by a gradient ascent step, so that (10) dominates the penalty terms in Theorem 2 in the main text (one can see this by observing that the penalty terms are now $\mathcal{O}(\eta^2)$ but the gains from steepest ascent is $\mathcal{O}(\eta)$). However, differentiability must be assumed, and moreover, θ_t^{k+1} must also be admissable, i.e. belong to Θ_t. If either condition is violated, the modification is not valid.

D Implementation and Model Details

A Tensorflow implementation of the binary and ternary MSA algorithm, together with code to reproduce our results are found at

`https://github.com/LiQianxiao/discrete-MSA`

D.1 MSA for Binary-weight Neural Networks

We give additional implementation details of our binary network algorithm (Alg. 2 in the main text), which is essentially Alg. 1 with the parameter update step replaced by (16). One extra step is to also keep and update an exponential moving average of $M_t^{\boldsymbol{\theta}_k}$ and use the averaged value to update our parameters. Note that in applications, we may have some floating-point precision layers (e.g. batch normalization layers), in which case the simplest way is to just train them using gradient descent. Also, Alg. 2 assumed that binary layers are fully-connected networks. For convolution networks, to compute $M_t^{\boldsymbol{\theta}}$, we simply have to take gradient of H_t with respect to θ (noting that H is linear in θ) to obtain the corresponding quantity. Before we discuss the choice of hyper-parameters in Sec. D.1.2, we first give an argument for the convergence of the binary MSA algorithm in a simple setting.

D.1.1 Convergence of the Binary MSA for a Simple Problem

Let us show informally that Alg. 2 in the main text converges, with an appropriate choice of regularization parameter, for a simple binary linear regression problem. The motivation here is show the importance of the added regularization terms involving $\rho_{k,t}$.

Consider a simple linear regression problem (i.e. linear network with $T = 1$) in which the unique solution is a Binary matrix. For $s = 1, \dots, S$, let $x_{s,0} \in \mathbb{R}^{d_0}$ be independent and have independent and identically distributed components with mean 0 and variance 1. These are the training samples. We shall consider the full-batch version so no exponential moving averages are applied.

Let $\theta_0^* \in \{-1, +1\}^{d_0 \times d_1}$ be the ground-truth, and so our regression targets are $y_s = \theta_0^* x_{s,0}$. Define the sample loss function

$$\Phi_s(x) := \frac{1}{2}\|y_s - x\|^2.$$

At the k^{th} iteration, let us denote the error vector $\delta\theta_0^k = \theta_0^* - \theta_0^k$. Then, using the update rules in Alg. 2, we have

$$x_{s,1}^{\boldsymbol{\theta}^k} = \theta_0^k x_{s,0} \qquad p_{s,1}^{\boldsymbol{\theta}^k} = \frac{1}{S}\delta\theta_0^k x_{s,0}$$

and so

$$H_0(x_{s,0}^{\boldsymbol{\theta}^k}, p_{s,1}^{\boldsymbol{\theta}^k}, \theta_0) = \frac{1}{S}\delta\theta_0^k x_{s,0} \cdot \theta_0 x_{s,0}$$

The update step is then

$$[\theta_0^{k+1}]_{ij} = \begin{cases} \operatorname{sign}([\delta\theta_0^k G_S]_{ij}) & |[\delta\theta_0^k G_S]_{ij}| \geq 2S\rho_{k,0} \\ [\theta_0^k]_{ij} & \text{otherwise} \end{cases}$$

where $G_S := \frac{1}{S}\sum_{s=1}^{S} x_{s,0} x_{s,0}^T$. For large S, by the central limit theorem G_S is approximately the identity matrix plus a small perturbation that is $\mathcal{O}(1/\sqrt{S})$ (valid for small perturbations only, large deviations will have to be bounded carefully by concentration inequalities or precise asymptotics [Den Hollander, 2008, Boucheron et al., 2013]). Therefore, $\delta\theta_0^k G_S = \delta\theta_0^k + \mathcal{O}(\|\delta\theta_0^k\|_F/\sqrt{S})$. Taking the sign, we see that we get the correct answer (i.e. $\delta\theta_1^{k+1} = 0$) if $\|\delta\theta_0^k\|_F S^{-3/2} \ll \rho_{k,0} \ll \|\delta\theta_0^k\|_F S^{-1}$. Since $\|\delta\theta_0^k\|_F$ decreases as optimization proceeds, this also shows that we need to decrease $\rho_{k,t}$ as k increases.

Note that if we took the naive, unstabilized MSA with $\rho_{k,0} \equiv 0$ (i.e. Alg. 1 in the main text), then it is clear that a coordinate that has the right sign ($[\delta\theta_0^k]_{ij} = 0$) will continue to fluctuate because of the random signs introduced by the $\mathcal{O}(\|\delta\theta_0^k\|_F/\sqrt{S})$ term, and hence will not converge. This shows the importance of the regularization term in our algorithm.

D.1.2 Choice of Hyperparameters

Note that all constant factors multiplied to the hyper-parameters can be absorbed into the hyper-parameters themselves when implementing the algorithms. Hence in the following, $\rho_{k,t}$ represents the value of $2\rho_{k,t}$ in Alg. 2.

The preceding example also shows that the regularization parameter $\rho_{k,t}$ should be suitably decreased as the optimization proceeds. We found a good heuristic is to simply set $\rho_{k,t}$ to be a constant fraction of the maximum absolute value of the components of $M_t^{\boldsymbol{\theta}^k}$ that is not of the same sign as θ_t^k. For the binary experiment, we take this constant fraction to be 0.5 for all layers.

Another hyper-parameter is the exponential moving average parameter, α_t, which we take to be 0.999 in all experiments. We also decay it (i.e. making it closer to 1) as the iterations proceed.

D.1.3 Model Details for Experiments

For ease of comparison, we have used almost identical set-ups as in [Courbariaux et al., 2015]. The only difference is that we ignore the bias terms in all binary layers, resulting in slightly fewer parameters.

For the MNIST experiment, we optimize a (3xFC2048)-FC10 fully connected network. For CIFAR-10, we consider a convolutional neural network with (2xConv128)-2x2maxpool-(2xConv256)-2x2maxpool-(2xConv512)-2x2maxpool-(2xFC1024)-FC10. Lastly, for SVHN, we use the same network as CIFAR-10, but with half the number of channels in the convolution layers. All networks used ReLU activations and square-smoothed hinge loss. Note that the ReLU activation and the square-smoothed hinge loss are not twice differentiable, so technically it does not satisfy the assumptions in Theorem 2. Nevertheless, we observe that the algorithm converges. Also, we tested other activations (e.g. soft-plus, tanh) and losses (soft-max with cross entropy) and the results are similar. Batch-normalization is added after each affine transformation and before the non-linearity. Binary layers are trained according to Alg. 2, but batch-normalization layers have floating-point weights, and hence are trained by Adam optimizer [Kingma and Ba, 2014] for simplicity. In all our experiments, no preprocessing steps are used other than scaling all input values to be between 0 and 1. We have checked that using different set-ups (e.g. cross-entropy loss, different network structures) does not generally require retuning the parameters and the algorithm performs well. Note that however, we found that batch normalization layers are quite necessary for obtaining good performance in our algorithms, as is also the case in [Courbariaux et al., 2015]. In the main text, Theorem 2 justifies this to a certain extent, by requiring the inputs fed to be $\mathcal{O}(1)$.

In our comparisons with BinaryConnect [Courbariaux et al., 2015], we used the original code published at `https://github.com/MatthieuCourbariaux/BinaryConnect` with the only difference being that we changed the inference step to use binary weights (instead of full precision weights). Note that there are quite a number of regularization techniques employed here. To check their effects on the training loss, we ran the BinaryConnect code without stochastic binarization etc., but the training graphs are generally similar, hence we omit them here.

D.2 MSA for Ternary-weight Neural Networks

The model setups for ternary-weight neural networks are identical as the binary ones, except we also have a parameter λ_t for each layer that promotes sparsity. We take λ_t=1e-7 for all t and all experiments. It is expected that large values will lead to sparser solutions, but with worse accuracy. We did not tune this value to find the best sparsity-performance trade-off. This is worthy of future exploration. The other hyperparameter choices are mostly identical as in the Binary case, except we take $\rho_{k,t}$ to be a smaller fraction at 0.25.

References

[Bliss, 1938] Bliss, G. A. (1938). Normality and abnormality in the calculus of variations. *Transactions of the American Mathematical Society*, 43(3):365–376.

[Boucheron et al., 2013] Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

[Canon et al., 1970] Canon, M. D., Cullum Jr, C. D., and Polak, E. (1970). *Theory of optimal control and mathematical programming*. McGraw-Hill Book Company.

[Courbariaux et al., 2015] Courbariaux, M., Bengio, Y., and David, J.-P. (2015). Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131.

[Den Hollander, 2008] Den Hollander, F. (2008). *Large deviations*, volume 14. American Mathematical Soc.

[Halkin, 1966] Halkin, H. (1966). A maximum principle of the pontryagin type for systems described by nonlinear difference equations. *SIAM Journal on control*, 4(1):90–111.

[Holtzman, 1966] Holtzman, J. (1966). Convexity and the maximum principle for discrete systems. *IEEE Transactions on Automatic Control*, 11(1):30–35.

[Holtzman and Halkin, 1966] Holtzman, J. M. and Halkin, H. (1966). Discretional convexity and the maximum principle for discrete systems. *SIAM Journal on Control*, 4(2):263–275.

[Kingma and Ba, 2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kuhn and Tucker, 2014] Kuhn, H. W. and Tucker, A. W. (2014). Nonlinear programming. In *Traces and emergence of nonlinear programming*, pages 247–258. Springer.

[LeCun, 1988] LeCun, Y. (1988). A theoretical framework for back-propagation. In *The Connectionist Models Summer School*, volume 1, pages 21–28.

[Li et al., 2018] Li, Q., Chen, L., Tai, C., and E, W. (2018). Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18:1–29.