SELECTION DYNAMICS FOR DEEP NEURAL NETWORKS

HAILIANG LIU AND PETER MARKOWICH

Abstract. This paper introduces the mathematical formulation of deep residual neural networks as a PDE optimal control problem. We study the wellposedness, the large time solution behavior, and the characterization of the steady states for the forward problem. Several useful time-uniform estimates and stability/instability conditions are presented. We state and prove optimality conditions for the inverse deep learning problem, using the Hamilton-Jacobi-Bellmann equation and the Pontryagin maximum principle. This serves to establish a mathematical foundation for investigating the algorithmic and theoretical connections between optimal control and deep learning.

1. Introduction

Deep learning is machine learning using neural networks with many hidden layers, and it [8, 30, 21] has become a primary tool in a wide variety of practical learning tasks, such as image classification, speech recognition, driverless cars, or game intelligence. As such, there is a pressing need to provide a solid mathematical framework to analyze various aspects of deep neural networks.

Deep Neural Networks (DNN) have been successful in supervised learning, particularly when the relationship between the data and the labels is highly nonlinear. Their depths allow DNNs to express complex data-label relationships since each layer nonlinearly transforms the features and therefore effectively filters the information content.

Linear algebra was appropriate in the age of shallow networks, but is inadequate to explain why deep networks perform better than shallow networks. The continuum limit is an effective method for modeling complex discrete structures to facilitate their interpretability. The depth continuum limit made a breakthrough by introducing a dynamical system viewpoint and going beyond what discrete networks can actually do.

Most prior works on the dynamical systems viewpoint of deep learning have focused on algorithm design, architecture improvement using ODEs to model residual neural networks. However, the ODE description does not reveal any structure for hidden nodes with respect to width. To fill in this gap, we propose a simple PDE model for DNNs that represents the continuum limits of deep neural networks with respect to two directions: width and depth. One main advantage of the PDE model over the ODE model in [24] is its ability to capture the intrinsic selection dynamics among hidden units involved.

The main purpose of this paper is to focus on the study of the fundamental mathematical aspects of the PDE formulation. We seek to gain new insight into the dynamics

Date: Thursday 23rd May, 2019.

2000 *Mathematics Subject Classification.* 49K20, 49L20.

Key words and phrases. Deep Learning, Residual Neural Networks, Optimal control, Stability.

of the forward propagation and the well-posedness of the learning problem, through a study of the PDE that represents the forward propagation dynamics.

We point out that the link of deep learning to dynamical system and optimal control has attracted increasing attention [13, 14, 15, 24, 27, 31, 32, 33, 43]. An appealing feature of this approach is that the compositional structure is explicitly taken into account in the time evolution of the dynamical systems, from which novel algorithms and network structures can be designed.

1.1. **Discrete neural networks.** A neural network can be seen as a recursively defined function Φ on (a compact domain of) $\mathbb{R}^d$ into $\mathbb{R}^{N_L}$:

$$\Phi = L_L \circ F_{L-1} \cdots L_2 \circ F_1 \circ L_1.$$

Here L_k is an affine linear map from $\mathbb{R}^{N_{k-1}}$ to $\mathbb{R}^{N_k}$:

$$L_k(x) = a_k - B_k x,$$

where B_k are $N_k \times N_{k-1}$ matrices (network weights) and a_k are N_k-vectors (network biases). Obviously we have used $N_0 = d$. F_k is a nonlinear mapping from $\mathbb{R}^{N_k}$ into itself, which in the case of a residual neuron network has the form

$$F_k(y) = y + \sigma(y),$$

with $\sigma : \mathbb{R} \to \mathbb{R}$ being the so called network activation function acting component-wise, i.e.,

$$\sigma(y) = \mathrm{diag}(\sigma(y_1) \cdots \sigma(y_{N_k})).$$

For details in the setup of neural network functions Φ and their approximation qualities we refer to [10].

Thus, one layer of the residual network is given by

$$z^{l+1} = z^l + \sigma(a_l - B_l z_l),$$

which after rescaling σ with an artificial layer width $\tau << 1$ can be seen as an explicit Euler step of the system of ordinary differential equations:

$$\dot{z}(t) = \sigma(a(t) - B(t)z(t)), t_{l-1} \le t \le t_l,$$

where $t_l = \tau l$, and $t > 0$ corresponds here to an artificially introduced time-like variable representing the depth of the network. By now this is a fairly common procedure in DNNs, we refer to [13, 14, 24, 33, 32, 43, 15] and references therein.

In practical applications the dimensions $N_0 \cdots N_l$ vary significantly from one network layer to the next, so in order not to have a-priori dimensional restrictions it makes sense to pose the above ODE system on an infinitely dimensional space of continuously defined functions. This is the approach which we shall take in this paper.

Obvious advantages arise. In the space-time continuous case we gain a lot of modeling freedom and highly developed PDE theory and numerics can be applied to analyze and compute geometric aspects of the problem like attractors, sharp fronts etc. Also the associated inverse problem, namely to determine the weight and bias functions based on given data, can be rephrased easily as a classical optimization and/or control problem.

1.2. **Organization.** The paper is organized as follows. We discuss main ingredients of deep learning for the classification problem and introduce the basic PDE model for the forward propagation and the optimal control formulation of deep learning in Sect. 2. In Sect. 3, we study the wellposedness, the large time solution behavior, and the characterization of the steady states for the forward problem. Several useful a priori estimates and stability/instability conditions are presented. Sect. 4 is devoted to the back propagation problem and to show how optimal control theory can be applied. We compute the gradient of the final network loss in terms of the network parameter functions, which involves solving both forward and backward problems. We further develop a control theory based on the Pontryagin maximum principle (PMP) [39], which provides necessary conditions for optimal controls. We finally show that the value function solves an infinite-dimensional Hamilton–Jacobi–Bellman (HJB) partial differential equation. This dynamic programming approach provides the third way to find the optimal control parameters. Hence in this work we establish the link between training deep residual neural networks and PDE parameter estimation. The relation provides a general framework for designing, analyzing and training CNNs. Finally, two numerical algorithms for the learning problem, one is gradient based and another is PMP based, are presented in Sect. 5.

1.3. **Related work.** The approximation properties of deep neural network models are fundamental in machine learning. For shallow networks, there has been a long history of proving the so-called universal approximation theorem, going back to the 1980s [12, 23]. Such universal approximation theorems can also be proved for wide networks, see [9] for a single layer with sufficient number of hidden neurons, or deep networks, see [29, 16] for networks of finite width with sufficient number of layers. A systematic study on the network approximation theory has been recently made available [19].

Continuous time recurrent networks have been known in 1980s like the one proposed by Almeida [1] and Pineda [37], and analyzed by LeCun [28]. Recently, the interpretation of residual networks by He et al. [22] as approximate ODE solvers spurred research in the use of ODEs to deep learning. Based on differential equations, there are studies on the continuum-in-depth limit of neural networks [32, 43] and on designing network architectures for deep learning [13, 14, 24, 33].

The dynamical systems approach has also been explored in the direction of training algorithms based on the PMP and the method of successive approximations [27, 31]. The connection between back-propagation and optimal control of dynamical systems is known since the earlier works on control and deep learning [4, 6, 28]. For a rigorous analysis on formulations based on ODEs with random data we refer to [15]

The present paper proposes a PDE model which represents a continuum limit of neural networks in both depth and width. Instead of the analysis of algorithms or architectures, we focus on the mathematical aspects of the formulation itself and develop a wellposedness theory for the forward and backward problems, and further characterize the optimality conditions and value functions using both ODE (PMP) and PDE (HJB) approaches.

2. Mathematical formulation

There are three main ingredients of deep learning for the classification problem: (i) forward propagation transforms the input features in a nonlinear way to filter their information; (ii) Classication is described to predict the class label probabilities using the features at the final output layer (i.e., the output of the forward propagation); and (iii) the learning problem is formulated to estimate parameters of the forward propagation and classification to approximate the data-label relation.

We consider the following classification problem: Assume we are given training data consisting of a network input function $f_I(y)$, and label function $C(y)$, we want to learn a function that approximates the data-label relation on the training data and generalizes well to similar unlabeled data.

2.1. **The forward problem.**

The forward problem amounts to modeling and simulating the propagation of data. With complex and huge sets of data, fast and accurate forward modeling is a significant step in deep learning. It should be noted that typically the more parameters the model has, the less well-posed the inverse problem is.

We first formulate a forward PDE to model the data propagation using residual neural networks [22]. Let $y \in Y$ denote the neuron identifier variable. Here we assume that Y is a domain in $\mathbb{R}^n$. In order to construct a PDE-type model to describe the forward propagation in deep learning, we introduce an artificial time $t \in [0, T]$. The depth of the network is represented by the final time T. Let $f(y, t)$ be a function describing the residual neural network at time t with neuron identifier y, its propagation is governed by the following PDE:

$$\partial_t f(y,t) = \sigma\left(a(y,t) - \int_{z\in Y} b(y,z,t) f(z,t) dz\right), \tag{2.1}$$

where σ is the nonlinear activation function. Here $b = b(y, z, t)$ is the selection weight function, and $a = a(y, t)$ is the bias function. The input learning data set $f(y, t = 0) = f_I(y)$ then serves as the initial data for the above differential equation. One of our objectives in this work is to highlight the relation of the learning problem to this PDE model.

The activation function is typically (piecewise) smooth and monotonically non-decreasing. As commonly used examples, we consider the arctan, the hyperbolic tangent, the sigmoid of form $\frac{1}{1+e^{-s}}$, and the Rectified Linear Unit (ReLU) activations given by

$$\sigma(s) = s^+,$$

the positive part of s. Our results also apply to other choices such as the leaky ReLu defined by $\sigma(s) = \max\{0.1s, s\}$, and the Elu given by

$$\sigma(s) = \begin{cases} s & s > 0, \\ \alpha(e^s - 1) & s \le 0. \end{cases}$$

The performance of these activation functions varies on different tasks and data sets [40] and it typically requires a parameter to be turned. Thus, the ReLU remains one of the popular activation functions due to its simplicity and reliability [20, 30, 36].

The network output function at final time T is given by

$$O_T(y) := \int W(y,z) f(z,T) dz + \mu(y), \tag{2.2}$$

where W and μ are weight and bias functions to be determined later.

2.2. **The learning problem.** In order to complete the learning problem, we need to define a prediction function by

$$C^{\text{pre}} = h\left(O_T(y)\right).$$

One of the popular choices for h is the logistic regression function,

$$h(\xi) = e^{\xi}/(1+e^{\xi}).$$

The goal of the learning problem is to estimate the parameters of the forward propagation (i.e., a and b and the classifier W and μ) from an observed label function $C = C(y)$, so that the DNN accurately approximates the data-label relation for the training data and generalizes to new unlabeled data. The forward operator is highly nonlinear, and the learning problem most often does not fulfill Hadamard's postulate of well-posedness.

As we show below, the learning problem can be cast as a dynamic control problem, which provides new opportunities for applying theoretical and computational techniques from parameter estimation to deep learning problems.

We phrase learning as an optimization problem

$$\min J(C^{\text{pre}}, C) \tag{2.3a}$$

$$\text{such that } \partial_t f(y,t) = \sigma\left(a(y,t) - \int_Y b(y,z,t) f(z,t) dz\right), \quad t \in (0,T], \tag{2.3b}$$

where J is a suitable choice of objective/loss function characterizing the difference between the synthetic data C^{pre} generated by the current (and inaccurate) model parameter $m = (a, b, W, \mu)$ and the observable true label C. This is a data-fitting approach, similar to many other inverse problems that are formulated as PDE-constrained optimization.

The optimization problem in (2.3) is challenging for several reasons. Firstly, it is a high-dimensional non-convex optimization problem, and therefore one has to be content with local minima. Secondly, the computational costs per example are high, and the number of examples is large. Thirdly, very deep architectures are prone to problems such as vanishing and exploding gradients that may occur when the discrete forward (or backward) propagation is unstable.

2.3. **The choice of the objective function and regularization.** Typically the loss function J is chosen to be convex in its first argument and measures the quality of the predicted class label probabilities. A typical choice is

$$J(C^{\text{pre}}, C) = \frac{1}{2} \int_Y |C^{\text{pre}}(y) - C(y)|^2 dy.$$

For classification the cross entropy loss is often used to measure the model performance [42, 34].

To control noise and other undesirable effects occurring in inverse problems, one often adds a regularization term so that

$$\min J(C^{\text{pre}}, C) + \lambda R(m), \tag{2.4a}$$

where the regularizer R is a convex penalty functional, and the parameter $\lambda > 0$ balances between minimizing the data fit and noise control. Choosing an "optimal" regularizer, R, and regularization parameter λ is both crucial and nontrivial; see, e.g., [5, 21].

3. Wellposedness of the forward problem

In order to identify useful structures of the deep learning problem, we first make some assumptions on the parameters with which stability of the forward problem can be studied.

3.1. A general existence result.

Most of our results will be obtained under the following:

Assumption 1. Y is a domain in $\mathbb{R}^n$ and $0 < T < \infty$. The propagation operator

$$\sigma(S[f]) \text{ with } S[f] = a(y,t) - \int_{z\in Y} b(y,z,t) f(z,t) dz$$

satisfies:

- σ is globally Lipschitz continuous with $\sigma(0) = 0$ or $|Y| < \infty$.
- $a \in L^2(Y; L^1(0,T))$.
- $b \in L^2(Y \times Y; L^1(0,T))$

These conditions are sufficient to prove the following theorem of existence and uniqueness by Picard's iteration.

Theorem 3.1. *We suppose that Assumption 1 holds. Then,*
1. for any initial function $f_I \in L^2(Y)$ there exists a solution in $L^2(Y; C[0,\infty))$ which solves (2.1) with $f(0,\cdot) = f_I$. Furthermore,
2. for any two solutions f_1, f_2 of (2.1) in $L^2(Y; C[0,\infty))$, one has the following stability property:

$$\forall t \in [0,T],\ \|f_1(t,\cdot) - f_2(t,\cdot)\|_{L^2(Y)} \le e^{Lt} \|f_1(0,\cdot) - f_2(0,\cdot)\|_{L^2(Y)}, \tag{3.1}$$

for some $L > 0$. In particular, if $f_1(\cdot,0) = f_2(\cdot,0)$, then $f_1(\cdot,t) = f_2(\cdot,t)$ for all $t > 0$, so that uniqueness holds.

Proof. Existence follows from the recursive scheme

$$\begin{aligned} &f^0(y,t) = f_I(y), \\ &\partial_t f^{n+1} = \sigma(S[f^n]), \quad f^{n+1}(0) = f_I. \end{aligned}$$

For $f^n \in L^2(Y; C[0,\infty)) := Q$, f^{n+1} is well-defined in the same space by

$$f^{n+1}(t) = f_0 + \int_0^t \sigma(S[f^n])(\tau) d\tau,$$

which in the Q norm is bounded by

$$\|f_I\|_{L^2} + \sup|\sigma'(\cdot)| \left(\int_0^T (\|a(\cdot,\tau)\|_{L^2(Y)} d\tau + \int_0^T \|b(\cdot,\cdot,\tau)\|_{L^2(Y\times Y)} d\tau \|f^n\|_Q + C_0 T \right)$$
$$\leq \|f_I\|_{L^2} + C_0 T + C_1 + C_2(T)\|f^n\|_Q$$

where $C_0 = |\sigma(0)||Y|$ if $\sigma(0) \neq 0$, $C_1 = \sup|\sigma'(\cdot)|\|a\|_{L^1((0,T),L^2(Y))}$, and

$$C_2(T) = \sup|\sigma'(\cdot)| \int_0^T \|b(\cdot,\cdot,\tau)\|_{L^2(Y\times Y)} d\tau.$$

Note that we used the well-known fact that the operator norm in L^2 of the integral operator with the kernel b equals the norm of b in L^2, that is $\|b\|_{L^2(Y\times Y)}$. By using the fact that $C_2(s) \to 0$ if $s \to 0$, we get an upper bound for $\{f^n\}$ uniformly in n:

$$\|f^n\|_Q \leq \frac{\|f_I\|_{L^2} + C_0 T + C_1}{1 - C_2(T)}$$

if T is suitably small such that $C_2(T) < 1$. Then, by studying $f^{n+1} - f^n$ via

$$f^{n+1} - f^n = \int_0^t \sigma'(\cdot) \int b(y,z,\tau)(f^n - f^{n-1}) dz d\tau,$$

we have

$$\|f^{n+1} - f^n\|_Q \leq C_2(T)\|f^n - f^{n-1}\|_Q.$$

Thus one can conclude that $\{f^n\}_{n\in\mathbb{N}}$ is a Cauchy sequence in Q, which converges towards a solution f of the equation (2.1) for $C_2(T) < 1$. Global existence for any T then follows from a continuity argument by extending the local solution, proceeding as for the uniform estimate on $\{f^n\}_{n\in\mathbb{N}}$.

□

3.2. **Large time asymptotics, stability of steady states.** Clearly, it is important to study the forward dynamics of the residual neural network problem. Here we begin the discussion with the analysis of steady states and stability. For simplicity, we first assume the forward propagation operator to be autonomous. That is,

$$a = a(y), \quad b = b(y,z).$$

We make two basic assumptions here:
(A_1) σ is globally Lipschitz on $\mathbb{R}$, but $\sigma'(0) > 0$ and $\sigma(0) = 0$,
(A_2) $a \in L^2(Y)$, $b \in L^2(Y \times Y)$.

Then the forward problem becomes

$$f_t = \sigma(a - Bf), \tag{3.2a}$$
$$f(y,0) = f_I(y). \tag{3.2b}$$

Here the operator defined by

$$(Bf)(y) = \int_Y b(y,z) f(z) dz$$

from $L^2(Y)$ into $L^2(Y)$ is Hilbert-Schmidt and consequently compact.

Let $f_\infty \in L^2(Y)$ be a steady state. Note that steady states exist if and only if $a \in R(B)$ (= Range of B), and they are non-unique if and only if $N(B)$ (= Nullspace of B) is non-trivial. To study the stability of f_∞, we first linearize (3.2) at f_∞, so that its perturbation w satisfies

$$w_t = -\sigma'(0)Bw,$$
$$w(t=0) = w_I.$$

We obtain

$$w(t) = e^{-\sigma'(0)Bt} w_I.$$

For an eigenvalue-eigenfunction pair (ω, ϕ) of B we obviously have $e^{-\sigma(0)\omega t}\phi$ as a solution of the linearized IVP. Therefore if the spectrum of B contains an eigenvalue with negative real part, exponential instability holds for the linearized problem. If an eigenvalue of B with zero real part exists, asymptotic stability for the linearized problem does not hold.

To obtain stability for the linearized problem, we can impose

$$(Bv, v)_{L^2(Y)} = (B_s v, v)_{L^2(Y)} \geq 0 \quad \forall v \in L^2(Y),$$

where $B_s = \frac{1}{2}(B + B^\top)$ is the symmetric part of B.

Assume now that $B^\top = B$ and that all eigenvalues of B are positive (i.e., 0 is a spectral value but not an eigenvalue!). Then asymptotic stability can be concluded from the solution representation

$$w(t) = \sum_{l=1}^{\infty} e^{-\sigma'(0)\omega_l t}(w_I, \phi_l)_{L^2(Y)}\phi_l,$$

where $\{\phi_l\}$ is the C.O.N.S (complete orthonormal system) of eigenfunctions.

If $\omega = 0$ is an eigenvalue then stability (but not asymptotic stability) holds in the symmetric case.

We now turn to derive estimates for the forward propagation problem using Lyapunov functionals.

Set

$$u := a - Bf,$$

so that u solves

$$u_t = -B\sigma(u),$$
$$u(t=0) = u_I := a - Bf_I.$$

In order to recover f from u we use the equation $f_t = \sigma(u)$ so that

$$f(y,t) = f_I(y) + \int_0^t \sigma(u(y,s))ds. \tag{3.3}$$

Multiply the u-equation by $\sigma(u)$ so that

$$\frac{d}{dt}\int_Y \Sigma(u)dy = -\int_Y (B_s\sigma(u), \sigma(u)) \leq 0,$$

(again assuming that $B_s \geq 0$). This upon integration gives

$$\int_Y \Sigma(u(y,t))dy \leq \int_Y \Sigma(u_I(y))dy,$$

where $\Sigma'(s) = \sigma(s)$:

$$\Sigma(s) = \int_0^s \sigma(\xi)d\xi.$$

In order to obtain estimates for f we distinguish two cases. We assume that

$$\sigma(s)s \geq 0 \quad \text{for } s \neq 0, \tag{3.4}$$

and $|\sigma(s)| \geq C_1|s|$ for $|s| \geq C_2$, then

$$\Sigma(s) \geq C_3 s^2$$

for $|s| \geq C_4$. This allows to estimate Bf the following way:

$$\int_Y |Bf|^2 dy \leq 2\int_Y |a|^2 dy + 2\int_Y |a - Bf|^2 dy \leq C \quad \forall t > 0.$$

If only $|\sigma(s)| \geq C_1$ for $|s| \geq C_2$, then

$$\int_Y |Bf(t)|dy \leq C \quad \forall t > 0$$

follows. Similarly, from the equation $f_t = \sigma(u)$ with $u = a - Bf$ it follows

$$\int_Y f_t Bf dy - \frac{d}{dt}\int_Y af dy = -\int_Y \sigma(u)u dy \leq 0,$$

because of (3.4). Assume now that $B^\top = B$ and B non-negative, we then have

$$\frac{1}{2}\frac{d}{dt}\|B^{1/2}f\|_{L^2(Y)} - \frac{d}{dt}(a,f)_{L^2(Y)} = -\int_Y \sigma(u)u dy.$$

Thus

$$\frac{1}{2}\|B^{1/2}f(t)\|_{L^2(Y)} - (a,f)_{L^2(Y)}$$

is monotonically decreasing and bounded from below, admitting a limit as $t \to \infty$.

Also

$$0 \leq \int_0^\infty \int_Y \sigma(u(y,s))u(y,s)dyds < \infty$$

and assuming $a \in R(B^{1/2})$,

$$\begin{aligned}\|B^{1/2}f(t)\|^2_{L^2(Y)} &\leq K + (a, f(t))_{L^2(Y)} \\ &\leq K + (B_s^{-1/2}a, B_s^{1/2}f(t))_{L^2(Y)} \leq K + C\|B_s^{1/2}f(t)\|_{L^2(Y)}.\end{aligned}$$

Thus

$$\|B^{1/2}f(t)\|^2_{L^2(Y)} \leq K_1 \quad \forall t > 0.$$

Again, the projection of f onto $N(B)$ is not controlled by this estimate.

To collect facts, we have the following time-uniform estimates

Theorem 3.2. *1) If $B_s \geq 0$, then for any $t > 0$, we have*

$$(i) \qquad \int_Y \Sigma(a - Bf(t))(y))dy \leq \int_Y \Sigma(a - Bf_I)(y))dy,$$

where $\Sigma' = \sigma$, and

$$(ii) \qquad \int_0^\infty \|B_s^{1/2}\sigma(a - Bf(s))\|^2_{L^2(Y)}ds < \infty,$$

which implies that $B_s^{1/2} f_t := B_s^{1/2}\sigma(u) \in L^2(Y \times (0, \infty))$.
2) If $s\sigma(s) \geq 0$, $B^\top = B$, $B \geq 0$ and $a \in R(B^{1/2})$, then for all $t > 0$

$$(iii) \qquad \|B^{1/2}f(t)\|_{L^2(Y)} + |(a, f(t))_{L^2(Y)}| \leq K.$$

$$(iv) \qquad \int_0^\infty \int_Y \sigma(a - Bf(s))(y)) \cdot (a - Bf(s))(y))dyds < \infty.$$

3.3. **Characterization of steady states.** Note that the equation

$$\dot{u} = -B\sigma(u)$$

may have other equilibria than $u_e = 0$. In fact every u_e such that $\sigma(u_e) \in N(B)$ is an equilibrium. But $u_e = 0$ is the only one which may correspond to the equilibrium $f = 0$ of the equation (3.2) (it does if only if $a \in R(B)$). Note that

$$u(t) = u_I - B\int_0^t \sigma(u(s))ds \Rightarrow u(t) - u_I \in R(B).$$

Since $u_I = a - Bf_I$ we have $u_I - a \in R(B)$ and $u(t) - a \in R(B)$. Consider $0 \neq u_e \in L^2(Y)$ such that $\sigma(u_e) \in N(B)$. Then the corresponding solution of the $f-$equation is

$$f(y, t) = f_I + t\sigma(u_e), \quad u_e = a - Bf_I,$$

if $u_e - a \in R(B)$. Clearly the linearly increasing component $t\sigma(u_e) \in N(B)$ is not seen by the time-uniform estimates of Theorem 3.2.

To consider an example pick $\phi \in L^2(Y)$, $\|\phi\|_{L^2} = 1$. Define $u_e = \phi$, compute $\sigma_e = \sigma(\phi)$. Now choose $\psi \in \{\sigma_e\}^\perp$ and define the rank one operator

$$(Bf)(y) := \int_Y f(z)\psi(z)dz\phi(y).$$

Clearly $u_e = \phi$ is an equilibrium of $\dot{u} = -B\sigma(u)$. Also $u_e = \phi \in R(B)$. Now let $a = \alpha\phi \in R(B)$ ($\alpha \in \mathbb{R}$ given) and choose $f_I \in L^2(Y)$ such that

$$\int_Y f_I\psi dy = \alpha - 1.$$

Then

$$f(t) = f_I + t\sigma(\phi)$$

solves (3.2).

Lemma 3.3. *If B is symmetric, $\sigma(s)s \geq 0$ for all $s \in \mathbb{R}$ then every equilibrium u_e in $R(B)$ satisfies $u_e\sigma(u_e) = 0$.*

Proof. Since u_e is an equilibrium, $\sigma(u_e) \in N(B)$. The conclusion follows from $\overline{R(B)} = N(B)^\perp$, i.e.,

$$\int_Y u_e \sigma(u_e) dy = 0.$$

Hence $u_e \sigma(u_e) \equiv 0$. □

We shall now consider the stability of $u_e = 0$ for the case of non-symmetric B. To understand the involved complications, we start with the rank one operator with the kernal given by

$$b(y,z) := \mu \psi(z)\phi(y), \quad \mu > 0,$$

where

$$\phi, \psi \in L^2(Y), \quad \int_Y \phi^2 dy = \int_Y \psi^2 dy = 1.$$

Then the equation $\dot{u} = -B\sigma(u)$ reads

$$u_t(y,t) = -\mu \int_Y \sigma(u(z,t))\psi(z)dz\phi(y),$$

assume $u(t=0) = u_I = \beta_0 \phi(y)$. Clearly $u(y,t) = \beta(t)\phi(y)$ solves the ODE:

$$\begin{aligned} \dot{\beta}(t) &= -\mu \int_Y \sigma(\beta(t)\phi(z))\psi(z)dz =: -\mu g(\beta) \\ \beta(0) &= \beta_0. \end{aligned}$$

We now compute the Taylor expansion of g at $\beta = 0$, assuming sufficiently smoothness of σ:

$$\begin{aligned} g(\beta) = &\int_Y \phi(z)\psi(z)dz\sigma'(0)\beta \\ &+ \frac{1}{2}\int_Y \phi(z)^2\psi(z)dz\sigma''(0)\beta^2 \\ &+ \frac{1}{6}\int_Y \phi(z)^3\psi(z)dz\sigma'''(0)\beta^3 + O(\beta^4). \end{aligned}$$

Case 1. $\int_Y \phi(z)\psi(z)dz > 0$.
Then $u = 0$ is a locally isolated asymptotically stable equilibrium.

Case 2. $\int_Y \phi(z)\psi(z)dz < 0$.
Then $u = 0$ is a locally unstable equilibrium.

Case 3. $\int_Y \phi(z)\psi(z)dz = 0$. Then the local behavior is governed by

$$\dot{\beta} = -\frac{\mu}{2}\sigma''(0)\int_Y \phi^2\psi dz\beta^2 - \frac{\mu}{6}\sigma'''(0)\beta^3 \int_Y \phi^3\psi dz + O(\beta^4).$$

Case 3 (i): $\sigma''(0)\int_Y \phi^2\psi dz > 0$. Then local asymptotic stability holds for $\beta_0 \geq 0$ but not for $\beta_0 \leq 0$.
Case 3 (ii). $\sigma''(0)\int_Y \phi^2\psi dz < 0$. Then local asymptotic stability holds for $\beta_0 \leq 0$ but not for $\beta_0 \geq 0$.

Case 3 (iii). $\sigma''(0)\int_Y \phi^2\psi dz = 0$. Then local asymptotic stability holds for

$$\sigma'''(0)\int_Y \phi^3\psi dz \neq 0.$$

This discussion shows the inherent difficulty of the stability question in the non-symmetric case, arbitrarily high derivatives of σ at 0 can be decisive.

In more generality we consider the singular value decomposition of the operator B [44]:

$$(Bf)(y) = \sum_{l=1}^{\infty} \mu_l (\psi_l, f)_{L^2(Y)} \phi_l(y),$$

where the singlar values $\mu_l \geq 0$ are the eigenvalues of $|B|$, $\{\psi_l\}$ and $\{\phi_l\}$ are orthonomal systems, $\{\psi_l\}$ is complete in $L^2(Y)$ and $\phi_l = U\psi_l$, where $B = U|B|$ is the polar decomposition of B. Here U is a partial isometry so that $N(U) = N(B)$. Set

$$f = \sum_{k=1}^{\infty} f_k \psi_k, \quad f_k = \int_Y f\psi_k dy.$$

Thus

$$\begin{aligned}(Bf, f)_{L^2(Y)} &= \sum_{l=1}^{\infty}\sum_{k=1}^{\infty}\sum_{n=1}^{\infty} \mu_l f_k f_n (\psi_l, \psi_k)(\phi_l, \psi_n) \\ &= \sum_{l=1}^{\infty}\sum_{n=1}^{\infty} \mu_l f_l f_n (\phi_l, \psi_n) \\ &= f^\top D T f,\end{aligned}$$

where $f = (f_1, f_2, \cdots)^\top$, $D = \mathrm{diag}(\mu_1, \mu_2, \cdots)$ and T is the generalized Gram matrix:

$$T = ((\phi_l, \psi_n)).$$

Note that $(Bf, f)_{L^2(Y)} \leq 0$ for all $f \in L^2(Y)$ if and only if DT is non-positive definite (not necessarily symmetric).

Now let B have rank $N < \infty$.

$$(Bf)(y) = \sum_{l=1}^{N} \mu_l (f, \psi_l)_{L^2(Y)} \phi_l(y).$$

Set

$$u(y,t) = \sum_{l=1}^{N} u_l(t)\phi_l(y) + Z(y,t),$$

where $Z \in R(B)^\perp$. Thus

$$B\sigma(u(t)) = \sum_{l=1}^{N} \mu_l (\sigma(u(t)), \psi_l)\phi_l \in R(B).$$

Clearly, $\mu_1, \cdots, \mu_N > 0$. Assuming again $a \in R(B)$ we have $u(\cdot, t) \in R(B)$ for all $t \geq 0$. We conclude $Z \equiv 0$ and

$$u(y,t) = \sum_{l=1}^{N} u_l(t)\phi_l(y).$$

Thus, we find

$$\dot{u}_l = -\mu_l \int_Y \sigma\left(\sum_{k=1}^{N} u_k \phi_k(y)\right) \psi_l(y) dy \tag{3.5a}$$

$$u_l(t=0) = \int_Y u_I \phi_l dy. \tag{3.5b}$$

Let $v = (v_1, \cdots, v_N)^\top \in \mathbb{R}^N$ and denote

$$H(v) = (H(v)_1, \cdots, H(v)_N)^\top,$$

where

$$H(v)_l = -\mu_l \int_Y \sigma\left(\sum_{k=1}^{N} v_k \phi_k(y)\right) \psi_l(y) dy.$$

We have $\sigma(v) = \alpha v + o(v)$ for $v \sim 0$ with $\alpha > 0$. Thus

$$H(v)_l = -\mu_l \alpha \sum_{k=1}^{N} \int_Y \phi_k \psi_l dy v_k = -\alpha \mu_l (T_N v)_l + o(v),$$

where the generalized $N \times N$ Gram matrix T_N is given by

$$T_N = \left((\phi_k, \psi_l)_{L^2(Y)}\right).$$

If T_N is invertible then $u = 0$ is an isolated equilibrium of (3.5). Now we check the Lyapunov functional

$$L(u) = \int_Y \Sigma(u) dy, \quad \Sigma' = \sigma.$$

Clearly, $\Sigma(v) \sim \frac{\alpha}{2} v^2$ for $v \sim 0$ and

$$\int_Y \Sigma\left(\sum_{l=1}^{N} v_l \phi_l\right) dy \sim \frac{\alpha}{2} \int_Y |\sum_{l=1}^{N} v_l \phi_l|^2 dy = \frac{\alpha}{2} |v|^2.$$

Thus $L(u) > 0$ for $u \neq 0$ small.

Differentiating L gives

$$\begin{aligned} \frac{d}{dt} L(u(t)) &= -\int_Y \sigma(u(t)) B \sigma(u(t)) dy \\ &= -\sum_{l=1}^{N} \mu_l (\psi_l, \sigma(u(t))_{L^2(Y)} (\phi_l, \sigma(u(t))_{L^2(Y)} \end{aligned}$$

with

$$\sigma(u) \sim \alpha \sum_{k=1}^{N} u_k \phi_k,$$

we find

$$\begin{aligned}\frac{d}{dt}L(u(t)) &\sim -\alpha^2\sum_{l=1}^{N}\sum_{k=1}^{N}\sum_{n=1}^{N}u_k u_n\mu_l(\psi_l,\phi_k)_{L^2(Y)}(\phi_l,\phi_n)_{L^2(Y)}\\ &= -\alpha^2\sum_{l=1}^{N}\sum_{k=1}^{N}u_k u_l\mu_l(\psi_l,\phi_k)_{L^2(Y)}\\ &= -\alpha^2 u^\top D_N T_N D_N u\end{aligned}$$

with $D_N = \mathrm{diag}(\mu_1,\cdots,\mu_N)$.

We conclude

Theorem 3.4. *Let*
(a) T_N be invertible;
(b) $D_N T_N$ be positive-definite (not necessarily symmetric).
Then, the equilibrium $u = 0$ is locally asymptotically stable. The convergence of $u(t)$ to zero is exponential.

We remark that the local exponential stability of u induces local exponential stability of f. The proof follows standard arguments using Lyapunov functionals for ODE systems [26].

If B is symmetric, then $\phi_l = \psi_l$, $T_N = id$ and $D_N T_N = D_N$ is non-negative definite if only if $B \geq 0$.

3.4. **On solutions for the ReLu activation.** Note that for the arctan, sigmoid, and hyperbolic tangent activation functions, the asymptotic growth rate in time of the solution f can at most be linear, no matter what the properties of the operator B are. In this respect, the ReLu and leaky ReLu activation functions behave worse as we shall show below.

We now consider the activation function $\sigma(s) = s^+$, which is one of the most popular activations used in practical applications. In this case assuming $a \in L^1(Y)$ and $b \in L^\infty(Y\times Y)$, from the equation for f it follows

$$f_t \leq |a - B_b f| \leq |a| + B_{|b|}|f| \leq |a| + \sup_{Y\times Y}|b|\int_Y |f(y,t)|dy.$$

Integration against $\mathrm{sign}(f)$ yields

$$\frac{d}{dt}\int_Y |f(y,t)|dy \leq \int_Y |a(y)|dy + \sup_{Y\times Y}|b|\int_Y |f(y,t)|dy.$$

Thus

$$\int_Y |f(y,t)|dy \leq C_1 e^{\sup_{Y\times Y}|b|t}, \quad \forall t > 0.$$

We shall show below by example that the exponential upper bound is sharp. This tells us that for $\sigma(s) = s^+$, exponential forward instability for f is possible.

Now let $f_e \in L^1(Y)$ be a steady state so that

$$a(y) - (B_b f_e)(y) \leq 0 \quad \forall y \in Y.$$

We consider two cases:
1) $\sup_Y(a - B_b f_e) < 0$. Then there is a ball in $L^1(Y)$ with sufficiently small radius and center f_e which only contains steady states.

2) $\sup_Y(a - B_b f_e) = 0$. Consider B_b as a rank one operator given by

$$(B_b v)(y) = \int_Y v(z)\phi(z)dz\psi(y),$$

with $\phi, \psi \not\equiv 0$. Assume that $a(y) = a_0\psi(y)$ and look for solution of the form

$$u(y,t) = \lambda(t)\psi(y),$$

so that

$$\dot{\lambda}(t) = -\int_Y (\lambda(t)\psi(y))^+\phi(z)dz,$$
$$\lambda(t=0) = \lambda_I := a_0 - \int_Y f_I(z)\phi(z)dz.$$

From the decomposition

$$(\lambda\psi)^+ = \lambda^+\psi^+ + \lambda^-\psi^-,$$

where $\lambda = \lambda^+ - \lambda^-$, it follows

$$\dot{\lambda}(t) = -\alpha\lambda^+ - \beta\lambda^-,$$

where

$$\alpha = \int_Y \psi^+\phi dz, \quad \beta = \int_Y \psi^-\phi dz.$$

(i) If $\lambda_I \geq 0$, then $\lambda^- = 0$, and

$$\lambda(t) = e^{-\alpha t}\lambda_I.$$

(ii) If $\lambda_I < 0$, then $\lambda^+ = 0$ and $\lambda = -\lambda^-$, so that

$$\lambda(t) = -e^{\beta t}|\lambda_I|.$$

Hence we have

$$u^+(y,t) = \begin{cases} e^{-\alpha t}|\lambda_I|\psi^+(y), & \lambda_I \geq 0, \\ e^{\beta t}|\lambda_I||\psi^-(y)|, & \lambda_I < 0. \end{cases}$$

This allows us to recover f from

$$f(y,t) = f_I(y) + \int_0^t u^+(y,s)ds.$$

That is,

$$f(y,t) = \begin{cases} f_I(y) + \frac{|\lambda_I|}{\alpha}\psi^+(y)(1 - e^{-\alpha t}), & \lambda_I \geq 0, \\ f_I(y) + \frac{|\lambda_I|}{\beta}\psi^-(y)(e^{\beta t} - 1), & \lambda_I < 0. \end{cases}$$

Recall that we have assumed $a = a_0\psi(y)$ for some $a_0 \in \mathbb{R}$ and $\lambda_I = a_0 - \int_Y f_I(y)\phi(y)dy$. Hence

$$\lim_{t\to\infty} f(y,t) = \begin{cases} f_I(y) + \frac{|\lambda_I|}{\alpha}\psi^+(y), & \lambda_I \geq 0,\ \alpha > 0, \\ f_I(y) + \frac{|\lambda_I|}{|\beta|}\psi^-(y), & \lambda_I < 0,\ \beta < 0. \end{cases}$$

Let $\psi, \phi \in L^\infty(Y)$. Assume that $a - B_b f_e \equiv 0$, i.e.,

$$a_0 = \int_Y f_I(z)\phi(z)dz.$$

Then in every L^1 neighborhood of f_e there is a function $f_I^{(1)}$ such that $\lambda_I^{(1)} > 0$ and a function $f_I^{(2)}$ such that $\lambda_I^{(2)} < 0$. In both cases it is easy to construct functions ϕ and ψ such that exponential stability and, respectively, instability occurs.

For the rank one operator, it turns out that the steady states with $a - Bf_e \leq 0$ but not identically zero are locally stable, although not asymptotically stable.

If $B_s \geq 0$, we can actually prove that $\|f(\cdot,t)\|_{L^2(Y)}$ has at most linear growth in time.

Proposition 3.5. *Let $\sigma(s) = s^+$ and $B_s = \frac{1}{2}(B + B^\top) \geq 0$. Then there exist $C_1, C_2 > 0$ such that*

$$\|f(t)\|_{L^2(Y)} \leq C_1 + C_2 t \quad \forall t > 0.$$

Proof. From $f_t = u^+$ and $u = a - Bf$ we have

$$u_t = -Bu^+.$$

Using the Lyapunov argument from §3.2 gives

$$\frac{1}{2}\frac{d}{dt}\int_Y (u^+(t))^2(y)dy = -(B_s u^+, u^+)_{L^2(Y)} \leq 0.$$

Thus

$$\int_Y (u^+(t))^2(y)dy \leq C \ \ \forall t > 0.$$

This together with $f_t = u^+$ yields

$$\int_Y (f_t)^2 dy \leq C \ \ \forall t > 0.$$

Using the relation $f(t) = f_I + \int_0^t \partial_s f ds$, we obtain the estimate as claimed. □

Note that the same result holds for the leaky ReLu activation.

3.5. **Local conditioning of the forward problem.** For numerical analysis and computational purpose it is beneficial to understand the conditioning of the forward propagation operator, which means that its linearization of the actual solution, not only the steady state must be looked at. Also, the output of the learning problem will be time-dependent functions $a = a(y,t)$ and $b = b(y,z,t)$ such that the forward propagation and its linearization will be non-autonomous. Consider a solution $u = u(y,t)$ of the forward problem (3.2) and compare the linearization of the solution in direction $w = w(y,t)$, with $u = a - Bf$:

$$\partial_t w(y,t) = -\sigma(u(y,t)) \int_Y b(y,z,t)w(z,t)dz. \tag{3.6}$$

If the residual neural network problem is 'very' deep, and if $u(t)$ is close to the stationary state $u \equiv 0$ (assuming that a and b stabilize sufficiently fast as $t \to \infty$), then the

dynamics for w will be close to the autonomous case considered above. To see this, multiply by $\frac{w}{\sigma'(u)}$ and integrate over Y:

$$\frac{1}{2}\frac{d}{dt}\int_Y \frac{w^2(y,t)}{\sigma'(u)}dy = -(B(t)w,w)_{L^2(Y)} - \frac{1}{2}\int_Y \frac{\sigma''(u)\dot{u}(t)}{(\sigma'(u))^2}w^2(t)dy.$$

Here $-\frac{\sigma''(u)\dot{u}(t)}{(\sigma'(u))^2}$ measures the effect of the non-autonomous coefficients and of the linearization at the local solution (instead of a stationary one). If $f(t)$ is far away from the stationary state, then much less can be said about the operator $-\sigma(u)B(t)$ in general, except that it is bounded by

$$\sup_{\mathbb{R}} |\sigma'| \|b(\cdot,\cdot,t)\|_{L^2(Y\times Y)}$$

as an operator from $L^2(Y)$ into itself. It is generally non self-adjoint, even if $B(t)$ is self-adjoint.

4. Back propagation and optimal control

4.1. **Computing cost gradients.** The main technical difficulty in training continuous-depth networks is preforming reverse-mode differentiation (also known as back propagation). We introduce the following notation:

$$a = a(y,t),\ b = b(y,z,t), \quad u_{a,b} := a - B_b f, \quad f = f_{a,b},$$

where $f_{a,b}$ solves (4.1) below and

$$(B_b v)(y) = \int_Y b(y,z,t)v(z)dz.$$

For the sake of simplicity in the calculation, we consider first optimizing a simple terminal value loss functional

$$J(a,b) = \frac{1}{2}\int_Y (f_{a,b}(y,T) - \tilde{f}(y))^2 dy$$

subject to

$$\partial_t f_{a,b} = \sigma(a - B_b f_{a,b}), \tag{4.1a}$$

$$f_{a,b}(t=0) = f_I. \tag{4.1b}$$

Here $\tilde{f}(y)$ is the target output function. Let the Gateaux differential of f in a along direction α be

$$g = D_a f_{a,b}(\alpha) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(f_{a+\epsilon\alpha,b} - f_{a,b}),$$

then

$$g_t = \sigma'(u_{a,b})(\alpha - B_b g),$$
$$g(t=0) = 0.$$

Let $M_{a,b}(t,s)$ be the evolution system [38] generated by $-\sigma'(u_{a,b})B_b$, i.e. $z(t) := M_{a,b}(t,s)z_0$ solves

$$\dot{z} = -\sigma'(u_{a,b}(t))B_b z,\ t \geq s,$$
$$z(s) = z_0.$$

Then

$$g(t) = \int_0^t M_{a,b}(t,s)(\sigma'(u_{a,b}(s)\alpha(s))ds.$$

Similarly, $h = D_b f(\beta)$ solves

$$\begin{aligned} h_t &= -\sigma'(u_{a,b})(B_b h + B_\beta f), \\ h(t=0) &= 0, \end{aligned}$$

which gives

$$h(t) = -\int_0^t M_{a,b}(t,s)(\sigma'(u_{a,b}(s)B_\beta f(s))ds.$$

We proceed to compute the derivatives of J with respect to a and b as follows:

$$\begin{aligned} D_a J(a,b)(\alpha) &= \int_Y (f_{a,b}(y,T) - \tilde{f}(y))g(y,T)dy \\ &= \int_Y (f_{a,b}(y,T) - \tilde{f}(y)) \int_0^T M_{a,b}(T,s)(\sigma'(u_{a,b}(s)\alpha(s))(y)dsdy \\ &= \int_0^T \int_Y \alpha(y,s)\sigma'(u_{a,b}(y,s))M_{a,b}(T,s)^*(f_{a,b}(y,T) - \tilde{f}(y))dyds. \end{aligned}$$

Thus

$$D_a J(a,b)(y,s) = \sigma'(u_{a,b}(y,s))M_{a,b}(T,s)^*(f_{a,b}(T) - \tilde{f})(y).$$

Define

$$r_T(y) := (f_{a,b}(T) - \tilde{f})(y).$$

Clearly, $r(s) := M_{a,b}(T,s)^* r_T$ solves the co-state terminal value problem,

$$\begin{aligned} \dot{r} &= (\sigma'(u_{a,b}(s)B_b)^* r = B_b^*(\sigma'(u_{a,b})(s)r), \\ r(T) &= r_T. \end{aligned}$$

Note that $B_b^* = B_{b^\top}$ with $b^\top(y,z,t) = b(z,y,t)$. Thus, $r(s) = r_{a,b}$, and $r_{a,b}$ solves

$$\dot{r}_{a,b} = B_{b^\top}(\sigma'(u_{a,b})(s)r_{a,b}(s)), \tag{4.2a}$$

$$r_{a,b}(T) = f_{a,b}(T) - \tilde{f}. \tag{4.2b}$$

We conclude

$$D_a J(a,b)(y,s) = \sigma'(u_{a,b}(y,s))r_{a,b}(y,s). \tag{4.3}$$

As for the gradient with respect to b we have

$$\begin{aligned} &D_b J(a,b)(\beta) \\ &= \int_Y (f_{a,b}(y,T) - \tilde{f}(y))h(y,T)dy \\ &= -\int_Y (f_{a,b}(y,T) - \tilde{f}(y)) \int_0^T M_{a,b}(T,s)(\sigma'(u_{a,b})(s)B_\beta f_{a,b}(s))(y)dsdy \\ &= -\int_0^T \int_Y \sigma'(u_{a,b}(s))B_\beta f_{a,b}(s)M_{a,b}(T,s)^*(f_{a,b}(T) - \tilde{f})(y)dyds \end{aligned}$$

$$= -\int_0^T \int_Y \int_Y \beta(y,z,s) f_{a,b}(z,s)\sigma'(u_{a,b}(y,s)) M_{a,b}(T,s)^*(f_{a,b}(T)-\tilde{f})(y)dydzds$$
$$= -\int_Y \int_Y \int_0^T \beta(y,z,s) f_{a,b}(z,s)\sigma'(u_{a,b}(y,s)) r_{a,b}(y,s)dsdydz.$$

This gives

$$D_b J(a,b)(y,z,s) = -f_{a,b}(z,s)\sigma'(u_{a,b}(y,s))r_{a,b}(y,s). \tag{4.4}$$

We collect the results on the gradient of J in the following:

Proposition 4.1. *We have*

$$(i) \qquad D_a J(a,b)(y,s) = \sigma'(u_{a,b}(y,s))r_{a,b}(y,s),$$
$$(ii) \qquad D_b J(a,b)(y,z,s) = -f_{a,b}(z,s)\sigma'(u_{a,b}(y,s))r_{a,b}(y,s).$$

Therefore, conditions (necessary and sufficient) for a stationary point

$$(a,b) \in L^2(Y\times(0,T)) \times L^2(Y\times Y\times(0,T))$$

of the functional $J(a,b)$ are:
(a) solve

$$f_t = \sigma(a - B_b f), 0 < t \le T, \quad f(t=0) = f_I$$

for $f = f_{a,b} = f_{a,b}(y,t)$, $u_{a,b} := a - B_b f_{a,b}$;
(b) solve

$$r_s = B_b{}^\top(\sigma'(u_{a,b})(s)r(s)),\ 0 \le s < T,$$
$$r(s=T) = f_{a,b}(T) - \tilde{f}$$

for $r = r_{a,b} = r_{a,b}(y,s)$. Then the first condition is

$$\sigma'(u_{a,b}(y,s))r_{a,b}(y,s) = 0, \quad a.e.\ y \in Y, \quad s \in (0,T); \tag{4.5}$$

and the second is

$$f_{a,b}(z,s)\sigma'(u_{a,b}(y,s))r_{a,b}(y,s) = 0, \quad a.e.\ (y,z) \in Y\times Y, s \in (0,T). \tag{4.6}$$

Remark 4.1. If $\sigma' > 0$ (this holds for arctan, hyperbolic tangent, and Sigmoid). The above two conditions imply that the optimal (a^*, b^*) exists if only if $\tilde{f}$ is reachable in the sense that the above two derivatives vanish if and only if $f_{a,b}(T,y) = \tilde{f}(y)$ a.e. in Y.

Remark 4.2. Note that the conclusion of Remark 4.1 does not hold if the cost functional is regularized by, say, the Tikhonov regularizer

$$R(m) = \frac{1}{2}\int_Y (|\mu(y)|^2 + \int_0^T |a(y,t)|^2 dt)dy$$
$$+ \frac{1}{2}\int_{Y\times Y} (|W(y,z)|^2 + \int_0^T |b(y,z,t)|^2 dt)dydz,$$

such that $J(a,b)$ is replaced by

$$J_{\mathrm{mod}}(m) := J(a,b) + \lambda R(m). \tag{4.7}$$

Then

$$D_a J_{\rm mod}(m) = (\sigma'(u_{a,b})r_{a,b})(y,s) + \lambda a(y,s), \tag{4.8a}$$

$$D_b J_{\rm mod}(m) = -f_{a,b}(z,s)(\sigma'(u_{a,b})r_{a,b})(y,s) + \lambda b(y,z,s) \tag{4.8b}$$

This is commonly done in the deep learning applications.

The above analysis is well generalizable to the classification problem for which only the final cost needs to be modified by

$$J(a,b) = \frac{1}{2}\int_Y |C^{\rm pre}(y) - C(y)|^2 dy$$

with

$$C^{\rm pre}(y) = h(O_T(y)), \quad O_T(y) = \int_Y W(y,z) f(z,T) dz + \mu(y).$$

For the back propagation, we obtain the same equation

$$r_s = B_{b^\top}(\sigma'(u_{a,b})(s) r(s)),\ 0 \le s < T,$$

but with a different terminal condition

$$r(z,T) = \int_Y (C^{\rm pre}(y) - C(y)) h'(O_T(y)) W(y,z) dy.$$

4.2. **Pontryagin Maximum Principle.** We now view the deep learning problem in the framework of the mathematical control theory using the Pontryagin maximum principle to obtain optimal controls for the network parameter functions a and b, see [17]. This is of particular interest, when controls (a,b) which vary in regions with boundaries, are sought.

Let $a = a(y,t)$ and $b = b(y,z,t)$ be in a measurable set $A \subset \mathbb{R}^2$ pointwise a.e.. Define

$$I(a,b) = -\frac{1}{2}\int_Y (f_{a,b}(y,T) - \tilde{f}(y))^2 dy.$$

Look for

$$\max_{(a,b)\in A} I(a,b) = I(a^*,b^*).$$

Also we define the Hamiltonian

$$H(f,r,a,b) := \int_Y \sigma(a - B_b f) r dy,$$

where r is the co-state variable. Let (a^*,b^*) be optimal for I. Define $f^* = f_{a^*,b^*}$, then

$$\dot{f}^* = \sigma(a^* - B_{b^*} f^*),\ 0 \le t \le T,$$
$$f^*(t=0) = f_I.$$

Also define the optimal co-state r^* by

$$\dot{r}^* = B_{(b^*)^\top}(\sigma'(a^* - B_{b^*} f^*) r^*,\ 0 \le t \le T,$$
$$r^*(t=T) = \tilde{f} - f^*(T),$$

where $(b^*)^\top(y,z,t) = b^*(z,y,t)$. Then (a^*, b^*) satisfies the maximum-principle.

$$\begin{aligned} H(f^*, r^*, a^*, b^*) &= \max_{(a,b)\in A} H(f^*, r^*, a, b) \\ &= \max_{(a,b)\in A} \int_Y \sigma(a - B_b f^*) r^* dy. \end{aligned} \tag{4.9}$$

Note that the Hamiltonian is constant along the coupled dynamics:

$$\frac{d}{dt} H(f^*(t), r^*(t), a^*(t), b^*(t)) = 0.$$

As an example, take $A = [a^-, a^+] \times [b^-, b^+] \subset \mathbb{R}^2$. Let $\sigma'(s) \geq 0$, $\sigma \not\equiv 0$ and assume there is no $t \in [0,T]$ such that $r^*(t) \equiv 0$. Then one concludes immediately defining χ_Ω as the indicator function on the set Ω,

$$\begin{aligned} a^*(y,t) &= a^+ \chi_{\{r^* \geq 0\}}(y,t) + a^- \chi_{\{r^* \leq 0\}}(y,t), \\ b^*(y,z,t) &= b^- \left(\chi_{\{r^* \geq 0\}}(y,t) \chi_{\{f^* \geq 0\}}(z,t) + \chi_{\{r^* < 0\}}(y,t) \chi_{\{f^* < 0\}}(z,t) \right) \\ &\quad + b^+ \left(\chi_{\{r^* > 0\}}(y,t) \chi_{\{f^* < 0\}}(z,t) + \chi_{\{r^* < 0\}}(y,t) \chi_{\{f^* > 0\}}(z,t) \right). \end{aligned}$$

For a general control set A, compact in $\mathbb{R}^2$, set:

$$K_A = \left\{ (a,b) \subset \mathbb{R} \times L^2(Y) \,\middle|\, \quad (a, b(z)) \in A \text{ a.e. in } Y \right\}.$$

K_A is closed in $\mathbb{R} \times L^2(Y)$. For $f \in L^2(Y)$ define the affine linear functional

$$T_f(a,b) = a - \int_Y b(z) f(z) dz. \tag{4.10}$$

Clearly, $T_f : K_A \to \mathbb{R}$ assumes its minimum at (a_f^-, b_f^-) and maximum at (a_f^+, b_f^+) in K_A since T_f is bounded on K_A, weakly continuous and minimizing and maximizing sequences in K_A have weakly converging subsequences in K_A. This gives, again assuming that σ is non-decreasing and that $r^* \not\equiv 0$:

$$a^*(y,t) = a^+_{f^*(t)} \chi_{\{r^* \geq 0\}}(y,t) + a^-_{f^*(t)} \chi_{\{r^* < 0\}}(y,t), \tag{4.11a}$$

$$b^*(y,z,t) = b^+_{f^*(t)}(z) \chi_{\{r^* \geq 0\}}(y,t) + b^-_{f^*(t)}(z) \chi_{\{r^* < 0\}}(y,t). \tag{4.11b}$$

Note that the forward evolution for f^* and the backward evolution for the co-state r^* are now coupled in a highly nonlinear way through the optimal controls (a^*, b^*). Existence and uniqueness issues for this initial-terminal value problem will be the subject of future work.

Also note that the Maximum Principle does not give any information on optimality if the state $\tilde{f}$ is reachable by a bounded control in K_A. In this case $maxI = 0$ and the optimal control has to be computed as in Section 4.1.

Remark 4.3. For the network loss function of the classification problem

$$I(a,b) = -\frac{1}{2} \int_Y |C^{\text{pre}}_{a,b} - C|^2 dy,$$

with

$$C^{\text{pre}}_{a,b}(y) = h\left(\int_Y f_{a,b}(z,T) W(z,y) dz + \mu(y) \right) = h(O_{a,b,T}(y)),$$

the only modification again is the terminal value of the co-state,

$$r^*(T) = -\int_Y (C^{\rm pre}_{a^*,b^*}(y) - C(y))h'(O_{a^*,b^*,T}(y))W(y,z)dy.$$

4.3. **Functional Halmilton-Jacobi-Bellman PDE.** We now present an alternative approach to the control problem base on the dynamic programming principle. Consider

$$\begin{aligned}\partial_s f(y,s) &= \sigma(a(y,s) - (B_b f)(y,s)), \quad t < s \le T,\\ f(y,t) &= v(y)\end{aligned}$$

for general $v(\cdot) \in L^2(Y)$. Let a general cost functional be defined by

$$J_{v,t}(a,b) = \int_t^T \int_Y L(f(y,s),a,b)dyds + \frac{1}{2}\int_Y (f(y,T) - \tilde{f})^2 dy,$$

where the first term denotes the running cost and the second term is a terminal cost. Define a value functional as

$$F(v,t) = \inf_{(a,b)\in A} J_{v,t}(a,b) = J_{v,t}(a^*,b^*).$$

Note that $F(v,T) = \frac{1}{2}\int_Y (v(y) - \tilde{f})^2 dy$. By the dynamic programming principle (see e.g.,[7]) we conclude

Theorem 4.2. *Assume the value functional F is smooth in its arguments (v,t). Then $F(v,t)$ solves the functional Hamilton-Jacobi-Bellman (HJB) equation*

$$\partial_t F(v,t) + \min_{(a,b)\in A}\left\{\int_Y D_v F(v,t)\sigma(a - B_b v)dy + \int_Y L(v,a,b)dy\right\} = 0 \tag{4.12}$$

with the terminal condition

$$F(v,T) = \frac{1}{2}\int_Y (v(y) - \tilde{f}(y))^2 dy. \tag{4.13}$$

Remark 4.4. Note that $D_v F(v,t)$ is the L^2 variational gradient of the functional $F(t) : L^2(Y) \to \mathbb{R}$.
(i) We can express the HJB as

$$\partial_t F(v,t) + H(v, D_v F(v,t)) = 0,$$

where we define the Hamiltonian as

$$H(v,r) = \min_{(a,b)\in A}\left\{\int_Y \sigma(a - B_b v)r dy + \int_Y L(v,a,b)dy\right\}.$$

It is easy to see that the characteristic system of this functional HJB equation in the case $L = 0$ is precisely the coupled optimal control system of the previous section.

Note that the HJB equation 'lives' in the space of functionals on the space $L^2(Y)$.

Next, we show how to design the optimal control (a^*, b^*) using the above dynamic programming approach.

Step 1. Solve the HJB equation

$$\partial_t F(v,t) + H(v, D_v F(v,t)) = 0 \quad 0 \le t \le T,$$

subject to the terminal condition (4.13) to find the value functional $F(v,t)$.

Step 2. Use $F(v,t)$ and the HJB equation to construct an optimal (a^*, b^*):
(i) for each $v \in L^2(Y)$ and each time $t \in [0,T]$, define

$$(\tilde{a}(v(t))(y), \tilde{b}(v(t))(y,z)) = \text{argmin}_{(a,b)\in A}\left\{\int_Y D_v F(v,t)\sigma(a - B_b v)dy + \int_Y L(v,a,b)dy\right\}.$$

(ii) Next we find $\tilde{f}(y,s)$ by solving the following PDE

$$\begin{aligned}&\partial_s \tilde{f} = \sigma(\tilde{a}(v)(y,t) - B_{\tilde{b}(v)(y,z,t)}\tilde{f}), \quad t \le s \le T,\\ &\tilde{f}(t) = v.\end{aligned}$$

(iii) Finally define the feedback control

$$a^*(y,s) := \tilde{a}(\tilde{f}(s))(y),\ b^*(y,z,s) := b(\tilde{f}(s))(y,z), \quad t \le s \le T.$$

Theorem 4.3. *The control (a^*, b^*) is optimal.*

Proof. By standard arguments from dynamic programing, see [7]. □

5. Two iterative algorithms

5.1. **Gradient descent.** We recall that the gradient of the cost functional

$$J(a,b) = \frac{1}{2}\int_Y (f_{a,b}(y,T) - \tilde{f}(y))^2 dy$$

is given by

$$D_a J = \sigma(u_{a,b}(y,s))r_{a,b}(y,s), \quad D_b J = -f_{a,b}(z,s)\sigma(u_{a,b}(y,s))r_{a,b}(y,s),$$

where $u_{a,b} = a - B_b f_{a,b}$, and $r_{a,b}$ is obtained by solving

$$\begin{aligned}&\dot{r}_{a,b} = B_{b^\top}(\sigma'(u_{a,b}(y,s))r_{a,b}(y,s), \quad 0 \le s \le T,\\ &r_{a,b}(\cdot,T) = f_{a,b}(\cdot,T) - \tilde{f}(\cdot).\end{aligned}$$

We remark that the conditioning of the control problem for $r_{a,b}$ is identical to the conditioning of the forward problem. More precisely, with $\tau = T - s$, $0 \le \tau \le T$, $R_{a,b}(\tau) := r_{a,b}(s)$ we obtain

$$\dot{R}_{a,b}(y,\tau) = -B_{b^\top(T-\tau)}(\sigma'(u_{a,b}(y,T-\tau)))R_{a,b}(y,\tau), \quad 0 \le \tau \le T.$$

Note that the generator of the evolution equation for $R_{a,b}$ at τ is precisely the transposed of the generator of the linearized convolution equation for $f_{a,b}$ at time $T - \tau$ and we can estimate

$$\|B_{b^\top(T-\tau)} \circ \sigma'(u_{a,b}(T-\tau))\|_{L^2(Y)\to L^2(Y)} \le \sup_{\mathbb{R}} |\sigma'| \|b(\cdot,\cdot,T-\tau)\|_{L^2(Y\times Y)}.$$

(compare to section 3.5).

Then we present the following algorithm.

Algorithm 1.
Inputs: $\tilde{f}(y)$, $f_I(y)$, a^0, b^0 as initial guess, step size τ.

Outputs: a, b and $J(a, b)$
1. For $k = 1, 2, \cdots$ iterate until convergence.
2. Employ the celebrated proximal point algorithm (PPA) [41] for a and b, respectively,

$$a^{k+1} = \text{argmin}_a \left\{ J(a, b^k) + \frac{1}{2\tau}\|a - a^k\|^2 \right\}. \tag{5.1a}$$

$$b^{k+1} = \text{argmin}_b \left\{ J(a^{k+1}, b) + \frac{1}{2\tau}\|b - b^k\|^2 \right\}. \tag{5.1b}$$

3. Update f as

$$f^{k+1} = f_{a^{k+1}, b^{k+1}}(y, s)$$

by solving

$$\partial_s f = \sigma(a^{k+1} - B_{b^{k+1}} f), \quad f(t = 0) = f_I.$$

Note that this algorithm needs to be modified when the cost functional is regularized. For the Tikhonov regularizer given in Remark 4.2, we replace $J(a, b)$ by $J_{\text{mod}}(a, b)$ defined in (4.7) and use (4.8) for the gradients.

We remark that (5.1) is actually the backward Euler method for gradient flows, also known as the minimizing movement scheme [18]. Here, at each step, the distance of the parameter update acts as a regularization to the original loss function. Note that PPA has the advantage of being monotonically decreasing, which is guaranteed for any step size $\tau > 0$. Indeed, by the definition of (a^{k+1}, b^{k+1}) in (5.1),

$$J(a^{k+1}, b^{k+1}) \leq J(a^k, b^k) - \frac{1}{2\tau}\left(\|a^{k+1} - a^k\|^2 + \|b^{k+1} - b^k\|^2\right).$$

PPA based implicit gradient descent algorithms have been explored in [46] for the classic k-means problem, and in [11] to accelerate the training of Deep Neural Networks.

A second way of obtaining a numerical scheme in using gradients is in terms of the corresponding Riemannian structure. A well known example is the Fisher natural gradient [2].

We should point out that training deep neural networks using gradient-based optimization fall into the noncovex nonsmooth optimization. Many researchers have been working on mathematically understanding the gradient descent method and its ability to solve nonconvex nonsmooth problems (see, e.g., [3, 25, 35, 45]).

5.2. **Hamiltonian maximization.** Recall the Hamiltonian of the form

$$H(v, r, a, b) = \int_Y \sigma(a - B_b v) r dy.$$

We present the following algorithm based on the Pontryagin maximum principle (PMP).

Algorithm 2.
Inputs: $\tilde{f}(y)$, $f_I(\cdot)$, a^0, b^0 as initial guess.
Outputs: a, b and $J(a, b)$

1. For $k = 1, 2, \cdots$ iterate until convergence.
2. find $f^k = f_{a^k,b^k}$ by solving the forward problem

$$\partial_t f = \sigma(a^k - B_{b^k} f) \quad f(t=0) = f_I.$$

3. find $r^k = r_{a^k,b^k}$ by solving the backward problem

$$\partial_t r = B_{b^\top}(\sigma'(a^k - B_{b^k} f^k), \quad r(t=T) = \tilde{f} - f^k(T).$$

4. Update (a, b) by

$$(a^{k+1}, b^{k+1}) = \text{argmax}_{(a,b)\in A} H(f^k, r^k, a, b).$$

Since σ is non-decreasing, the linear programing problem (4.10) may be used to update (a, b).

One advantage of this approach is that it does not rely on gradients with respect to the trainable parameters. For recent works using PMP based algorithms to train neural networks, we refer to [27, 31].

Implementation and convergence analysis of the above two algorithms are left for further work.

Acknowledgement

We are grateful to Michael Herty (RWTH) for his interest, which motivated us to investigate this problem and eventually led to this paper. Liu was partially supported by the National Science Foundation under Grant DMS1812666 and by NSF Grant RNMS (Ki-Net)1107291.

References

[1] L.B. Almeida. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. *In Proceedings ICNN 87*, San Diego, 1987. IEEE, IEEE.
[2] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.
[3] S. Arora, N. Cohen, N. Golowich, and W. Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv:1810.02281*, 2018.
[4] M. Athans and P.L. Falb. Optimal Control: An Introduction to the Theory and Its Applications. *Courier Corporation*, Chelmsford, 2013.
[5] C. M. Bishop Pattern Recognition and Machine Learning. *(Information Science and Statistics)*, Springer, 2006.
[6] A.E. Bryson. Applied Optimal Control: Optimization, Estimation and Control. *CRC Press*, Boca Raton, 1975.
[7] M. Bardi and I. Capuzzo-Dolcetta. Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. *Birkhauser*, 1997.
[8] Y. Bengio. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, 2009.
[9] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.
[10] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal approximation with sparsely connected deep neural networks. *https://arxiv.org/pdf/1705.01714.pdf*, May 17, 2018.
[11] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. *arXiv:1611.01838.*, 2016.
[12] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[13] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert and E. Holtham. Reversible architectures for arbitrarily deep residual neural networks. *In: Proceedings of AAAI Conference on Artificial Intelligence*, 2018.
[14] B. Chang, L. Meng, E. Haber, F. Tung and D. Begert. Multi-level residual networks from dynamical systems view. *In: Proceedings of International Conference on Learning Representations*, 2018.
[15] W. E, J. Han and Q. Li. A mean-field optimal control formulation of deep learning. *Res. Math Sci.*, 6:10, 2019.
[16] W. E and Q. Wang Exponential convergence of the deep neural network approximation for analytic functions. *https://arxiv.org/pdf/1807.00297.pdf*, 2018.
[17] W. H. Fleming and R. W. Rishel. Deterministic and Stochastic Control. *Springer*, 1975.
[18] E. De Giorgi. New problems on minimizing movements. *in Boundary Value Problems for PDEs and their Applications*, eds. Massons, 81–98, 1993.
[19] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei. Deep neural network approximation theory. *https://arxiv.org*, 2019.
[20] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectified neural networks. *In International Conference on Artificial Intelligence and Statistics*, 315–323, 2011.
[21] Goodfellow, I., Bengio, Y. and Courville, A. Deep Learning. *MIT Press*, Cambridge, 2016.
[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778, 2016a.
[23] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
[24] E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Probl.*, 34(1): 014004, 2017.
[25] K. Kawaguchi. Deep learning without poor local minima. *In Advances in Neural Information Processing Systems*, 586–594, 2016.
[26] H. K. Khalil. Nonlinear Systems. *Pearon*, 3rd edition, 2014.
[27] Li, Q., Chen, L., Tai, C., E, W. Maximum principle based algorithms for deep learning. *J. Mach. Learn. Res.*, 18:1–29, 2018.
[28] LeCun, Y. A theoretical framework for back-propagation. *In: The Connectionist Models Summer School*, 21–28, 1988.
[29] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *In Advances in Neural Information Processing Systems*, 6232–6240, 2017.
[30] Y. LeCun, Y. Bengio and G. Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.
[31] Q. Li and S. Hao. An optimal control approach to deep learning and applications to discrete-weight neural networks. *arXiv:1803.01299v2*, June 2018.
[32] Li, Z., Shi, Z. Deep residual learning and PDEs on manifold. *arXiv preprint* arXiv:1708.05115, 2017.
[33] Y. Lu, A. Zhong, Q. Li and B. Dong. Beyond finite layer neural networks: bridging deep architectures and numerical differential equations. *arXiv preprint* arXiv:1710.10121, 2017.
[34] H. Masnadi-Shirazi and N. Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageBoost, *Statistical Visual Computing Laboratory*, University of California, San Diego, retrieved 6 December 2014.
[35] Y. Nesterov. Introductory lectures on convex optimization: A basic course. *Springer Science & Business Media*, Volume 87, 2013.
[36] V. Nair and G. E. Hinton. Rectied linear units improve restricted Boltzmann machines. *In Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814, 2010
[37] F. J. Pineda. Generalization of back propagation to recurrent and higher order neural networks. *In Proceedings of IEEE Conference on Neural Information Processing Systems*, Denver, Colorado, November, 1987. IEEE.
[38] A. Pazy. Semigroups of Linear Operators and Applications to Partial Differential Equations. *(Applied Mathematical Sciences)*, Springer, 1992.

[39] L. S. Pontryagin. Mathematical Theory of Optimal Processes. *CRC Press*, Boca Raton, 1987.
[40] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv:1710.05941*, 2017.
[41] R. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14: 877–898, 1976.
[42] L. Rosasco, E. D. De Vito, A. Caponnetto, M. Piana, A. Verri. Are loss functions all the same? *Neural Computation*, 16(5): 1063–1076, 2004.
[43] Sonoda, S., Murata, N. Double continuum limit of deep neural networks. *In: ICML Workshop on Principled Approaches to Deep Learning*, 2017.
[44] A.S. Sunder. Operators on Hilbert space. *Springer*, 2016.
[45] I. Safran and O. Shamir. Spurious local minima are common in two-layer Relu neural networks. *In International Conference on Machine Learning*, 4430–4438, 2018.
[46] P.Yin, M. Pham, A. Oberman, and S. Osher. Stochastic backward Euler: an implicit gradient descent algorithm for k-means clustering. *J. Sci. Comput.*, 77:1133–1146, 2018.

Iowa State University, Mathematics Department, Ames, IA 50011
E-mail address: hliu@iastate.edu

Computer, Electrical, Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.
E-mail address: Peter.Markowich@kaust.edu.sa