PREPRINT 500

JANUARY 2020

Kinetic Theory for Residual Neural Networks

Michael Herty, Torsten Trimborn and Giuseppe Visconti

Institut für Geometrie und Praktische Mathematik
Templergraben 55, 52062 Aachen, Germany

Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany, email: herty@igpm.rwth-aachen.de, trimborn@igpm.rwth-aachen.de, visconti@igpm.rwth-aachen.de

Kinetic Theory for Residual Neural Networks

Michael Herty, Torsten Trimborn, Giuseppe Visconti

Institut für Geometrie und Praktische Mathematik (IGPM)
RWTH Aachen University
Templergraben 55, 52062 Aachen, Germany

January 7, 2020

Abstract

Deep residual neural networks (ResNet) are performing very well for many data science applications. We use kinetic theory to improve understanding and existing methods. A microscopic simplified residual neural network (SimResNet) model is studied as the limit of infinitely many inputs. This leads to kinetic formulations of the SimResNet and we analyze those with respect to sensitivities and steady states. Aggregation phenomena in the case of a linear activation function are also studied. In addition the analysis is validated by numerics. In particular, results on a clustering and regression problem are presented.

Mathematics Subject Classification (2010) 35Q20 (Boltzmann equation), 35Q84 (Fokker-Planck equation), 90C31 (Sensitivity, stability, parametric optimization), 92B20 (Neural networks, artificial life and related topics), 68T05 (Learning and adaptive systems)

Keywords Residual neural network, continuous limit, mean field equation, kinetic equation, machine learning application

1 Introduction

The use of machine learning algorithms has gained a lot of interest in the past decade [14, 15, 31]. Besides the data science problems like clustering, regression, image recognition or pattern formation there are novel applications in the field of engineering as e.g. for production processes [21, 29, 32]. In this study we focus on deep residual neural networks (ResNet) which date back to the 1970s and have been heavily influenced by the pioneering work of Werbos [36]. ResNet have been successfully applied to a variety of applications such as image recognition [37], robotics [38] or classification [6]. More recently, also applications to mathematical problems in numerical analysis [25, 26, 34] and optimal control [28] have been studied.

A ResNet can be shortly summarized as follows: Given inputs, which are usually measurements, the ResNet propagates those to a final state. This final state is usually called output and aims to fit a given target. In order to solve this optimization procedure, parameters of the ResNet needs to be optimized and this step is called training. The parameters are distinguished as weights and biases. For the training of ResNets backpropagation algorithms are frequently used [35, 36].

The purpose of this work is to use kinetic theory in order to gain insights on performance of ResNet. There have been made several attempts to describe ResNet by differential equations [2, 19, 27]. For example in [27] the connection of deep convolution neural networks to partial differential equation (PDE) is derived. In [2] the time continuous version of a ResNet is studied and different temporal discretization schemes are discussed. There are also studies on application of kinetic methods to ResNet [1, 20, 30]. For example in [30] the authors consider the limit of infinitely many neurons and gradient steps in the case of one hidden layer. They are able to proof

a central limit theorem and show that the fluctuations of the neural network at the mean field limit are normally distributed.

In this work we do not consider the limit of infinitely many neurons. Instead we fix the number of neurons to the number of input features, and we call the resulting network simplified residual neural network (SimResNet). Then, we derive the time continuous limit of the SimResNet model which leads to a possible large system of ODEs. We consider the mean field limit in the number of inputs (or measurements) deriving a hyperbolic PDE. Throughout this study we assume that the bias and weights are optimized and given. The purpose of this approach is to analyze the forward propagation of the derived mean field neural network model with given weights and bias. We especially focus on aggregation and clustering phenomena, which we study with the help of the corresponding moment model. Furthermore, we compute steady states and perform a sensitivity analysis. The quantity of interest of the sensitivity analysis is an operator called loss function. With the help of the sensitivities we are able to deduce an update formula for the bias and weights in the case of a change in the input or target distribution.

In addition we study Boltzmann type equations with noisy neural network dynamics as extension to the mean field formulation. Long time behavior of such Boltzmann type equations can be conveniently studied in the grazing limit regime. This asymptotic limit naturally leads to Fokker-Planck type equations where it is possible to obtain non trivial steady state distributions [23, 24, 33]. The study of the aggregation phenomena gives us conditions on the shape of the weights and bias. In addition we gain information on the simulation time needed to reach a desired target. The novel update algorithm for the weight and bias seems to give a large performance increase in the case of a shifted target or initial condition. Finally, our Fokker-Planck asymptotics indicate that a stochastic SimResNet model performs well in several applications.

The outline of the paper is as follows. In Section 2 we define the microscopic ResNet model and the time continuous limit. In Section 3 we introduce first the SimResNet and then we derive the mean field neural network model. The corresponding Boltzmann type neural network model is presented in Section 4, with an asymptotic limit leading to a Fokker-Planck equation. We analyze the kinetic neural network formulations with respect to steady states and study qualitative properties with the help of a moment model and a sensitivity analysis. In Section 5 we conduct several numerical test cases which validate our previous analysis. Especially, we conduct two classical machine learning tasks, namely a clustering problem and a regression problem. We conclude the paper in Section 6 with a brief conclusion and an outlook on future research perspectives.

2 Time Continuous ResNet

We assume that the input signal consists of d features. A feature is one type of measured data as e.g. temperature of a tool, length or width of a vehicle, color intensity of pixels of an image. Without loss of generality we assume that the value of each feature is one dimensional and thus the input signals are given by $\boldsymbol{x}_i(0) \in \mathbb{R}^d,\ i = 1, ..., M$. Here, M denotes the number of measurements or input signals. In the following, we assume that the number of neurons is identical in each layer, corresponding to a fully connected ResNet. Namely, we consider L layers and in each layer the number of neurons is given by $\bar{N} := d\ N$, where N is the number of neurons for one feature. The microscopic model which defines the time evolution of the activation energy of each neuron $\boldsymbol{x}_i^k(t) \in \mathbb{R}^d,\ k = 1, ..., N$, fixed input signal $\boldsymbol{x}_i(0) \in \mathbb{R}^d$ and bias $\boldsymbol{b}(t) \in \mathbb{R}^d$ reads [10]:

$$\begin{cases} \boldsymbol{x}_i^k(t + \Delta t) = \boldsymbol{x}_i^k(t) + \Delta t\ \sigma\left(\frac{1}{dN} \sum_{j=1}^{N} \widehat{\boldsymbol{w}}_{kj}(t)\ \boldsymbol{x}_i^j(t) + \boldsymbol{b}(t)\right), \\ \boldsymbol{x}_i^k(0) = \boldsymbol{x}_i(0) \end{cases} \tag{1}$$

for each fixed $i = 1, \dots, M$ and $k = 1, \dots, N$. Here, $\sigma : \mathbb{R} \to \mathbb{R}$ denotes the activation function which is applied component wise. Examples for the activation function are given by the identity function $\sigma_I(x) = x$, the so-called ReLU function $\sigma_R(x) = \max\{0, x\}$, the sigmoid function $\sigma_S(x) = \frac{1}{1+\exp(-x)}$ and the hyperbolic tangent function $\sigma_T(x) = \tanh(x)$. In general (1) can be written in

compact formulation by suitably collecting all the weights $\widehat{\boldsymbol{w}}_{k,j}(t) \in \mathbb{R}^{d\times d},\ k = 1, ..., N, j = 1, ..., N$ in an extend matrix $\boldsymbol{W}(t) \in \mathbb{R}^{dN\times dN}$. In particular, we have that $\widehat{\boldsymbol{w}}_{k,j}(0) = 0$, for each $k \neq j$.

In (1) we formulated a neural network by introducing a parameter Δt. In a classical ResNet $\Delta t = 1$. Here, instead, we introduce a time discrete concept which corresponds to the layer discretization. More precisely the time step $\Delta t > 0$ is defined as $\Delta t := \frac{1}{L}$ and, in this way, we can see (1) as an explicit Euler discretization of an underlying time continuous model. As similar modeling approach with respect to the layers has been introduced in [27].

A crucial part in applying a neural network is the training of the network. By training one aims to minimize the distance of the output of the neural network at some fixed time $T > 0$ to the target $\boldsymbol{h}_i \in \mathbb{R}^d$. Mathematically speaking one aims to solve the minimization problem

$$\min_{\boldsymbol{W},\boldsymbol{b}} \|\boldsymbol{x}_i(T) - \boldsymbol{h}_i\|_2^2,$$

where we use the squared L^2 distance between the target and the output to defined the loss function. Other choices are certainly possible [12]. The procedure can be computationally expensive on the given training set. Most famous examples of such an optimization are so called back propagation algorithms or ensemble Kalman filters [9, 16, 35]. In the following we assume that the bias and weights are given and the neural network is already trained.

In the following we aim to consider the continuum limit which corresponds to $\Delta t \to 0$ and $L \to \infty$. In this limit, (1) is given by:

$$\begin{cases} \dot{\boldsymbol{x}}_i^k(t) = \sigma\left(\frac{1}{dN}\sum\limits_{j=1}^{N} \widehat{\boldsymbol{w}}_{kj}(t)\ \boldsymbol{x}_i^j(t) + \boldsymbol{b}(t)\right), \\ \boldsymbol{x}_i^k(0) = \boldsymbol{x}_i(0), \end{cases} \tag{2}$$

for each fixed $i = 1, \dots, M$ and $k = 1, \dots, N$.

Existence and uniqueness of a solution is guaranteed as long the activation function σ satisfies a Lipschitz condition.

3 Mean Field Formulations of SimResNet

In this section we derive a time continuous PDE model for the forward propagation of residual neural networks. We follow a Liouville type approach for infinitely many measurements or inputs in order to obtain a mean field equation.

3.1 SimResNet

We assume a single neuron $N = 1$ for each feature. Thus equation (2) becomes

$$\begin{cases} \dot{\boldsymbol{x}}_i(t) = \sigma\left(\frac{1}{d}\boldsymbol{w}(t)\ \boldsymbol{x}_i(t) + \boldsymbol{b}(t)\right), \\ \boldsymbol{x}_i(0) = \boldsymbol{x}_i(0), \end{cases} \tag{3}$$

for each fixed $i = 1, \dots, M$, and where $\boldsymbol{w}(t) \in \mathbb{R}^{d\times d}$. This simplification reduces the complexity of neural networks drastically. This special form is not only beneficial for the kinetic formulation of a neural network but especially reduces the costs in the training stage of a neural networks. This novel method has been successfully applied to an engineering application [7]. We refer to this formulation as SimResNet.

3.2 Mean Field Limit

We perform the kinetic limit in the number of measurements M. Since the dimension of $\boldsymbol{x}_i$ is directly related to the dimension of the variable of the kinetic distribution function, for practical

purposes we should consider moderate d. The mean field model corresponding to the dynamic (3) is

$$\partial_t g(t,\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \cdot \left(\sigma \left(\frac{1}{d} \boldsymbol{w}(t)\boldsymbol{x} + \boldsymbol{b}(t) \right) g(t,\boldsymbol{x}) \right) = 0 \tag{4}$$

where $g : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}$ is the compactly supported probability distribution function with known and normalized initial conditions

$$g(0,\boldsymbol{x}) = g_0(\boldsymbol{x}), \int\limits_{\mathbb{R}^d} g_0(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 1.$$

Observe that $g_0(\boldsymbol{x})$ corresponds to the distribution of the measured features and that (4) preserves the mass, i.e. $\int\limits_{\mathbb{R}^d} g(t,\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = 1, \forall\, t > 0$. The derivation is classical and we refer to Golse et al. [8, 11] for details. If $F(t,\boldsymbol{x},u) := u\,\sigma\left(\frac{1}{d}\boldsymbol{w}(t)\boldsymbol{x} + \boldsymbol{b}(t)\right)$ fulfills

$$F \in C^2(\mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}; \mathbb{R}); \ \partial_u F \in L^\infty(\mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}; \mathbb{R}), \ \partial_u div_x(F) \in L^\infty(\mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}; \mathbb{R}) \tag{5}$$

Then the hyperbolic conservation law (4) admits a unique weak entropy solution in the sense of Krûzkov [3] for initial data $g_0 \in L^\infty \cap L^1$. The mean field neural network (4) can be solved pointwise by the method of characteristics.

Proposition 1. *Let $g(t,\boldsymbol{x})$ be a compactly supported weak solution of the mean field equation (4). Consider the case of the identity activation function $\sigma_I(x) = x$ or the L^∞ hyperbolic tangent activation function $\sigma_T(x) = \tanh(x)$. Assume $\boldsymbol{b}^\infty = \lim_{t\to\infty} \boldsymbol{b}(t)$ and $\boldsymbol{w}^\infty = \lim_{t\to\infty} \boldsymbol{w}(t)$. Then*

$$g_\infty(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{y})$$

is a steady state solution of (4) in the sense of distributions provided that $\boldsymbol{y}$ solves $\frac{1}{d}\boldsymbol{w}^\infty\boldsymbol{y} + \boldsymbol{b}^\infty = \boldsymbol{0}$.

Proof. For a test function $\phi(\boldsymbol{x}) \in C_0^\infty(\mathbb{R}^d)$ the steady state equation reads

$$\int \nabla_{\boldsymbol{x}} \phi(\boldsymbol{x}) \sigma \left(\frac{1}{d} \boldsymbol{w}^\infty \boldsymbol{x} + \boldsymbol{b}^\infty \right) g_\infty(\boldsymbol{x}) = 0. \tag{6}$$

If $g_\infty = \delta(\boldsymbol{x} - \boldsymbol{y})$ is a Dirac delta function located at $\boldsymbol{y}$, equation (6) is satisfied only if $\boldsymbol{y}$ is the solution to the system $\frac{1}{d}\boldsymbol{w}^\infty\boldsymbol{y} + \boldsymbol{b}^\infty = \boldsymbol{0}$. □

With the help of the empirical measure it is straightforward to connect the solution of the large particle dynamics to the PDE. The empirical measure defined by the solution vector $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_M)^T \in \mathbb{R}^{dM}$ is given by

$$\mu_{\boldsymbol{X}}^M(t,\boldsymbol{x}) = \frac{1}{M} \sum_{k=1}^{M} \delta(x^1 - x_k^1(t)) \cdot ... \cdot \delta(x^d - x_k^d(t)).$$

A straightforward calculation shows that $\mu_{\boldsymbol{X}}^M(t,\boldsymbol{x})$ is a weak solution of the weak form of the model (4), provided that the initial distribution is given by

$$g_0(\boldsymbol{x}) = \frac{1}{M} \sum_{k=1}^{M} \delta(x^1 - x_k^1(0)) \cdot ... \cdot \delta(x^d - x_k^d(0)).$$

In order to show that the microscopic dynamics converge to the mean field limit we use the Wasserstein distance and the Dobrushin inequality. We follow the presentation in [8]. The convergence is obtained in the space of probability measures $\mathcal{P}(\mathbb{R}^d)$ using the 1-Wasserstein distance, which is defined as follows:

Definition 1. *Let μ and ν two probability measures on $\mathbb{R}^d$. Then the 1-Wasserstein distance is defined by*

$$W(\mu,\nu) := \inf_{\pi\in\mathcal{P}^*(\mu,\nu)} \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} |\xi-\eta| d\pi(\xi,\eta), \tag{7}$$

where $\mathcal{P}^*$ is the space of probability measures on $\mathbb{R}^d\times\mathbb{R}^d$ such that the marginals are μ and ν i.e.

$$\int_{\mathbb{R}^d} d\pi(\cdot,\eta) = d\mu(\cdot), \quad \int_{\mathbb{R}^d} d\pi(\xi,\cdot) = d\nu(\cdot)$$

Theorem 1. *We assume that the activation function σ of the microscopic system* (3) *is Lipschitz continuous with Lipschitz constant $L>0$. Let $g_0(\boldsymbol{x})$ the initial condition of the Cauchy problem* (4) *be a probability measure with finite first moment and*

$$W(\mu^M, g_0) \to 0, \text{ as } M\to\infty,$$

holds. Then the Dobrushin stability estimate

$$W(g(t),\mu^M(t)) \le \exp\{L\ t\}\ W(\mu^M, g_0)$$

is satisfied and

$$W(g(t),\mu^M(t)) \to 0,$$

holds as $M\to\infty$.

Proof. We only sketch the main steps of the proof and refer to [8, 11] for details. As first step we define the characteristic equations of the mean field neural network model (4). These characteristic equations are measure dependent and one usually uses the push-forward operator in order to be able to derive the Dobrushin stability estimate. The existence of the solution of the characteristic equations ban be shown with the help of the Lipschitz constant and the corresponding fixed point operator. As next step one considers the distance of two measures and again uses the fixed point operator and the Lipschitz continuity in order to bound the distance. Then one can apply the Gronwall inequality and obtains the Dobrushin stability estimate. □

3.3 Properties of the One Feature Mean Field Equation

In the case of one feature, i.e. $d=1$, the mean field equation (4) reduces to

$$\partial_t g(t,x) + \partial_x\left(\sigma(w(t)\ x + b(t))\ g(t,x)\right) = 0. \tag{8}$$

Our subsequent analysis is performed in this simple case.

We first define the k-th moment, $k\ge 0$, and variance of our mean field model by

$$m_k(t) := \int_{\mathbb{R}} x^k\ g(t,x)\ \mathrm{d}x, \quad \mathbb{V}(t) = m_2(t) - (m_1(t))^2. \tag{9}$$

Clearly, the possibility to obtain a moment model is solely determined by the shape of the activation function $\sigma(\cdot)$.

In the following we aim to characterize concentration phenomena of our solution g with respect to the functions w and b. Therefore, we study the expected value and energy of our mean field model. In particular, we are interested in the characterization of several concentration phenomena that we define below.

Definition 2. *We say that the solution $g(t,y)\ t\ge 0,\ y\in\mathbb{R}$ to equation* (8) *is characterized by*

i) energy bound if

$$m_2(0) > m_2(t),$$

holds at a fixed time t;

(ii) energy decay if

$$m_2(t_1) > m_2(t_2),$$

holds for any $t_1 < t_2$. This means that the energy is decreasing with respect to time;

(iii) concentration if

$$\mathbb{V}(0) > \mathbb{V}(t)),$$

holds at a fixed time t, where $\mathbb{V}$ denotes the variance;

(iv) aggregation if

$$\mathbb{V}(t_1) > \mathbb{V}(t_2),$$

holds for any $t_1 < t_2$. This means that the variance is decreasing with respect to time.

We observe that if the first moment is conserved in time, then definition of energy bound is equivalent to concentration, and definition of energy decay is equivalent to aggregation.

3.3.1 Identity as activation function

A simple computation reveals that the 0-th moment is conserved and $m_0(t) = 1$ holds for all times $t \geq 0$. For the moments $k \geq 1$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k(t) = k \, \Big(w(t) \; m_k(t) + b(t) \; m_{k-1}(t) \Big), \quad m_k(0) = m_k^0. \tag{10}$$

Notice that the k-th moment only depends on the $k-1$-th moment. It is then possible to solve the moment equations iteratively with the help of the separation of variables formula obtaining

$$m_k(t) = \exp\left\{ k \int_0^t w(s) \; \mathrm{d}s \right\} \left[m_k(0) + \int_0^t \exp\left\{ -k \int_0^s w(x) \; \mathrm{d}x \right\} \; k \; b(s) \; m_{k-1}(s) \; \mathrm{d}s \right]. \tag{11}$$

Let us define

$$\Phi_k(t) := k \int_0^t w(s) \; \mathrm{d}s.$$

Proposition 2. *Assume that the bias is identical to zero, namely $b(t) \equiv 0$, $\forall \, t \geq 0$. Then we obtain*

energy bound if $\Phi_1(t) < 0$ at a fixed time t; and

energy decay if and only if $w(t) < 0$ for all $t > 0$; and

aggregation if and only if $\lim_{t \to \infty} \Phi_1(t) = -\infty$. In particular the steady state is distributed as a Dirac delta centered at $x = 0$.

Proof. If $b(t) \equiv 0$ then (11) simplifies to

$$m_k(t) = m_k(0) \; \exp\{\Phi_k(t)\} \tag{12}$$

and thus the first and the second moment are identical except the given initial conditions. Then we can easily apply the definitions of energy bound, energy decay and aggregation to prove the statement. □

Corollary 1. *Assume that the bias is identical zero, namely $b(t) \equiv 0$, $\forall\, t \geq 0$. Then*

if energy bound exists at a time t we have concentration; and

if energy decay holds we have aggregation.

Proof. We start proving the first statement. First we observe that (12) is still true, since by hypothesis we assume that the bias is zero. Due to the definition of concentration we need to verify that $\mathbb{V}(0) > \mathbb{V}(t)$ for a fixed time t. Using the definition of the variance, concentration is implied by

$$m_2(0)\ (1 - \exp(\Phi_2(t))) > m_1(0)^2(1 - \exp(2\Phi_1(t))).$$

We have aggregation if $\mathbb{V}(t_1) > \mathbb{V}(t_2)$ for any $t_1 < t_2$. This is equivalent to

$$\begin{aligned} & m_2(t_1) - m_1(t_1)^2 > m_2(t_2) - m_1(t_2)^2 \\ \iff & m_2(t_1) - m_2(t_2) > (m_1(t_1) - m_1(t_2))\ (m_1(t_1) + m_1(t_2)) \\ \iff & m_2(t_1) > m_1(t_1)^2. \end{aligned}$$

□

The choice $b(t) \equiv 0$ is suitable for clustering problems at the origin, independently on the initial first moment. Conservation of the first moment is possible by choosing $b(t) := -w(t)m_1(t)$. See the following results.

Proposition 3. *Let the bias be $b(t) := -w(t)m_1(t)$. Then the first moment m_1 is conserved in time and we obtain*

a energy bound if $\Phi_2(t) < 0$ at a fixed time t; and an

aggregation phenomena if $w(t) < 0$ holds for all $t \geq 0$. The steady state is Dirac delta centered at $x = m_1(0)$.

Proof. The solution formula for the second moment is

$$m_2(t) = \exp\{\Phi_2(t)\}(m_2(0) - m_1(0))^2) + m_1(0)^2 = \exp\{\Phi_2(t)\}\mathbb{V}(0) + m_1(0)^2.$$

Then we have energy bound if

$$\mathbb{V}(0)\,(\exp\{\Phi_2(t)\} - 1) < 0$$

which is satisfied assuming that $\Phi_2(t) < 0$ at a fixed time t. For the second statement we observe that delta aggregation is also implied by $\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{V}(g(t)) < 0$, for all times. Or, equivalently, by $\frac{\mathrm{d}}{\mathrm{d}t}m_2(t) < 0$, for all times. We have

$$\frac{\mathrm{d}}{\mathrm{d}t}m_2(t) = w(t)\mathbb{V}(t) < 0$$

if and only if $w(t) < 0$ for all times. □

We aim to discuss the impact of the variance on aggregation and concentration phenomena. This is especially interesting if we are not interested in the long time behavior but rather aim to know if $\mathbb{V}(T) \leq V$ for some tolerance $V > 0$ and time $T > 0$. In applications this level would be determined by the variance of the target distribution.

Corollary 2. *If the bias is identical to zero, namely $b(t) \equiv 0$, $\forall\, t \geq 0$, then the energy at time $t > 0$ is below tolerance $V > 0$ if*

$$\Phi_2(t) < \ln\left(\frac{V}{m_2(0)}\right)$$

is satisfied. Similarly, we have that the variance is below the level $V > 0$ if

$$\Phi_2(t) < \ln\left(\frac{V}{\mathbb{V}(0)}\right)$$

holds.

Similarly, if the bias fulfills $b(t) := -w(t)m_1(t)$, then the energy at time $t > 0$ is below the level $V > 0$ if

$$\Phi_2(t) < \ln\left(\frac{V - m_1(0)^2}{\mathbb{V}(0)}\right)$$

is satisfied provided that $V > m_1(0)^2$ holds. Similarly, the variance at time $t > 0$ is below the level $V > 0$ if

$$\Phi_2(t) < \ln\left(\frac{V}{\mathbb{V}(0)}\right)$$

is satisfied provided that $V > 0$ holds.

Remark 1. *In general it is not possible to obtain a closed moment model in the case of the sigmoid $\sigma_S(x)$ or hyperbolic tangent $\sigma_T(x)$ activation function Nevertheless one might approximate the sigmoid or tanh activation function by the linear part of their series expansion.*

$$\sigma_S(x) \approx \frac{1}{2} + \frac{x}{4}, \quad \sigma_T(x) \approx x.$$

Remark 2. *In the case of the ReLU activation function we decompose the moments $k \geq 1$ in two parts*

$$m_k(t) = \int\limits_{\Omega^+(t)} x^k \; g(t,x) \; \mathrm{d}x + \int\limits_{\Omega^-(t)} x^k \; g(t,x) \; \mathrm{d}x,$$

and we define

$$m_k^+(t) := \int\limits_{\Omega^+(t)} x^k \; g(t,x) \; \mathrm{d}x, \quad m_k^-(t) := \int\limits_{\Omega^-(t)} x^k \; g(t,x) \; \mathrm{d}x$$

where $\Omega^+(t) := \{x \in \mathbb{R} \mid x > -\frac{b(t)}{w(t)}\}$ and $\Omega^-(t) := \mathbb{R} \setminus \Omega^+(t)$.

Let us define $a(t) = -\frac{b(t)}{w(t)}$, then using the Leibniz integration rule we compute

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k^-(t) = a(t)^k g(t, a(t)) \frac{\mathrm{d}}{\mathrm{d}t} a(t) \tag{13}$$

and

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} m_k^+(t) = &- a(t)^k g(t, a(t)) \frac{\mathrm{d}}{\mathrm{d}t} a(t) + k w(t) m_k^+(t) + k b(t) m_{k-1}^+(t) && (14) \\ &+ a(t)^{k+1} \; w(t) \; g(t, a(t)) + a(t)^k \; b(t) \; g(t, a(t)). && (15) \end{aligned}$$

Consequently, the evolution equation for the k-th moment cannot be expressed by a closed formula since it depends on the partial moment on $\Omega^+(t)$ and boundary conditions:

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k(t) = k \left(w(t) m_k^+(t) + b(t) m_{k-1}^+(t) \right) + a(t)^{k+1} \; w(t) \; g(t, a(t)) + a(t)^k \; b(t) \; g(t, a(t)). \tag{16}$$

In the case of constant weights and bias the equality $\dot{m}_k = \dot{m}_k^+$ holds. Notice also that the above discussion simplifies in the case of vanishing bias, and it becomes equivalent to the case when the activation function is the identity function. In fact, if $b(t) \equiv 0$, $\forall\, t \geq 0$, then the set Ω^+ switches to be $(-\infty, 0)$ or $(0, \infty)$, depending on the sign of the weight $w(t)$. Moreover, thanks to (13) we immediately obtain that $\frac{\mathrm{d}}{\mathrm{d}t} m_k^- \equiv 0$ holds true and thus $\frac{\mathrm{d}}{\mathrm{d}t} m_k(t) = \frac{\mathrm{d}}{\mathrm{d}t} m_k^+(t)$ is satisfied for all $t \geq 0$. Hence, the evolution equation (16) reduces to the case (10) and same computations can be performed.

3.4 Sensitivity Analysis

The goal is to compute the sensitivity of a quantity of interest with respect to some parameter. The quantity of interest is the distance of function g at finite time T to the target distribution h. We assume the ResNet has been trained using the loss function D.

$$D(T; w, b, g_0) := \frac{1}{2} \int_{\mathbb{R}} |g(T, x) - h(x))|^2 \, \mathrm{d}x.$$

We may expect that training was expensive and will not necessarily be done again if input or target changes. Therefore, it is of interest if the trained network (namely w and b) can be reused if h or g_0 changes. We propose to compute the corresponding sensitivities of D with respect to the weight and bias. This in turn can be used to apply a gradient step on (w, b). We use adjoint calculus to adjust (w, b) to the modified (h, g_0), i.e. the formal first order optimality condition for Lagrange multiplier $\lambda(t, x)$ reads:

$$\begin{aligned}
&\partial_t \lambda(t,x) + \sigma(w\ x + b)\ \partial_x \lambda(t,x) = 0 \\
&\lambda(T,x) = |g(T,x) - h(x)| \\
&\partial_t g(t,x) + \partial_x(\sigma(w\ x + b)\ g(t,x)) = 0 \\
&g(0,x) = g_0(x) \\
&g(t,x)\ x\ \sigma'(wx + b)\ \partial_x(\lambda) = 0 \\
&g(t,x)\ \sigma'(wx + b)\ \partial_x(\lambda) = 0
\end{aligned}$$

where $\sigma'(x)$ is the derivative of the activation function. It is amed that σ is differentiable, i.e. $\sigma = \sigma_T$ or $\sigma = \sigma_S$. Adjustment of optimal weights and bias can be then obtained via gradient step. After an update of initial data or target maybe summarized by:

- `initially select optimized weights` $w,\ b$
- `update` D by `new initial data and target` (g_0, h)
- `compute`

$$\begin{aligned}
w^{k+1} &= w^k - \gamma\ g\ x\ \sigma'\ \partial_x(\lambda) \\
b^{k+1} &= b^k - \gamma\ g\ \sigma'\ \partial_x(\lambda),
\end{aligned}$$

  `with step size` $\gamma > 0$

- `repeat until`

$$(w^{k+1} - w^k)^2 + (b^{k+1} - b^k)^2 < tol$$

  `for a given tolerance` $tol > 0$ `is reached`

4 Boltzmann type Equations

The mean field models in the previous section can be obtained as suitable asymptotic limit of Boltzmann type space homogeneous integro-differential kinetic equations. In particular, the case of one-dimensional feature can be derived from a linear Boltzmann type equation.

In fact, in the one-dimensional case the system of ODEs (3) can be recast as the following linear interaction rule:

$$x^* = x + \sigma(w(t)\ x + b(t)), \tag{17}$$

where, by kinetic terminology, x^* and x are the post- and pre-collision states, respectively. The corresponding weak form of the Boltzmann type equation reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}} \phi(x)\ g(t,x)\ \mathrm{d}x = \frac{1}{\tau}\int_{\mathbb{R}} [\phi(x^*) - \phi(x)]\ g(t,x)\ \mathrm{d}x \tag{18}$$

where τ represents the interaction rate. In the present homogeneous setting, τ influences only the relaxation speed to equilibrium and thus, without loss of generality, in the following we take $\tau = 1$.

The one-dimensional Boltzmann type equation (18) leads to the one-dimensional mean field equation (8) by suitable scaling [23, 24, 33].

An advantage of the Boltzmann type description (18) in comparison to the mean field equation is the possibility to study different asymptotic scales leading to non-trivial steady states. The choice of weights w, bias b and of the activation function may be obtained by fitting the target distribution.

In order to obtain non-trivial steady states of model (18) consider self-similar solutions [24]:

$$\bar{g}(t,x) = m_1(t) g(t, m_1(t)\ x).$$

In the the case of the identity activation function the first moment can be computed explicitly and a Fokker-Planck type asymptotic equation can be derived. In order to study steady-state profiles for arbitrary activation functions, we choose the following approach: we add noise to the microscopic interaction rule (17) leading in the asymptotic limit to a Fokker-Planck equation.

4.1 Fokker-Planck Neural Network Model

Let ϵ be a small parameter, weighting for the strength of the interactions. Modify the interaction (17) as

$$x' = x + \epsilon\sigma(w(t)x + b(t)) + \sqrt{\epsilon}K(x)\eta, \tag{19}$$

where η is a random variable with mean zero and variance ν^2, and $K(x)$ is a diffusion function. In the classical grazing collision limit $t = \epsilon t$, $\epsilon \to 0$, we recover the following Fokker-Planck equation for the probability distribution g

$$\partial_t g(t,x) + \partial_x \left[\mathcal{B}g(t,x) - \mathcal{D}\partial_x g(t,x)\right] = 0 \tag{20}$$

where we define the interaction operator $\mathcal{B}$ and the diffusive operator $\mathcal{D}$ as

$$\mathcal{B} = \sigma(w(t)x + b(t)) - \frac{\nu^2}{2}\partial_x K^2(x), \quad \mathcal{D} = \frac{\nu^2}{2}K^2(x).$$

The advantage in computing the grazing collision limit is that the classical integral formulation of the Boltzmann collision term is replaced by differential operators. This allows a simple analytical characterization of the steady state solution $g_\infty = g_\infty(x)$ of (20). Provided the target can be well fitted by a steady state distribution of the Fokker-Planck neural network model, the weight, the bias and the activation function are immediately determined. This is a huge computational advantage in comparison the the classical training of neural networks. In the following we present examples of steady states of the Fokker-Planck neural network model.

4.1.1 Steady state characterization

Steady state solution can be easily found as

$$g_\infty(x) = \frac{C}{K^2(x)} \exp\left(\int \frac{2\sigma(w_\infty x + b_\infty)}{\nu^2 K^2(x)}\mathrm{d}x\right) \tag{21}$$

The constant $C \in \mathbb{R}$ is determined by mass conservation, i.e. $\int_{\mathbb{R}} g_\infty(x)\mathrm{d}x = 1$. The existence and explicit shape of the steady state is determined by the specific choice of activation function $\sigma(\cdot)$, diffusion function $K(\cdot)$ and parameters w_∞, b_∞.

If the target $h(x)$ is distributed as a Gaussian, then choosing $\sigma(x) = \sigma_I(x)$ and $K(x) = 1$ yields a suitable approximation since the steady state (21) is given by

$$g_\infty(x) = C \exp\left(\frac{w_\infty}{\nu^2}x^2 + 2\frac{b_\infty}{\nu^2}x\right),$$

provided that $b_\infty = 0$. Condition on mass conservation leads to

$$C = \frac{\sqrt{-\frac{w_\infty}{\nu^2}}\exp\left(\frac{b_\infty^2}{w_\infty \nu^2}\right)}{\sqrt{\pi}}$$

with $w_\infty < 0$ to guarantee converge of $\int_{\mathbb{R}} g_\infty(x)\mathrm{d}x$.

If the target $h(x)$ is distributed as an inverse Gamma, then choosing $\sigma(x) = \sigma_I(x)$ and $K(x) = x$ yields a suitable approximation since the steady state (21) is given by

$$g_\infty(x) = \begin{cases} 0, & x \leq 0, \\ \frac{C}{x^{1+\mu}}\exp\left(\frac{\mu-1}{x}\ \frac{b_\infty}{w_\infty}\right), & x > 0, \end{cases}$$

with $\mu := 1 + \frac{2\ w_\infty}{\nu^2}$ and normalization constant

$$C = \frac{\left((1-\mu)\frac{w_\infty}{b_\infty}\right)^\mu}{\Gamma(\mu)},$$

where $\Gamma(\cdot)$ denotes the Gamma function. Notice that we have to assume that $w_\infty < 0$ and $b_\infty > 0$ hold in order to obtain a distribution.

Let $\sigma(x) = \sigma_R(x)$ and $K(x) = x$, and w. l. o. g. we assume $w_\infty < 0$ so that σ_R is identical zero on the set $\Omega := \{x \in \mathbb{R} | x \geq -\frac{b_\infty}{w_\infty}\}$. The steady state on Ω is given by $g_\infty^\Omega = \frac{c}{x^2}, c > 0$ and can be extended on $\mathbb{R}$ by the Pareto distribution:

$$g_\infty = \begin{cases} -\frac{b_\infty}{w_\infty}\frac{1}{x^2}, & x \geq -\frac{b_\infty}{w_\infty}, \\ 0, & x < -\frac{b_\infty}{w_\infty}. \end{cases}$$

If activation and diffusion function

$$\sigma_N(x) := \left[\frac{1}{\delta}\left(\frac{x}{c}\right)^\delta - 1\right]x,\ 0 < \delta < 1,\ c > 0, \quad K(x) = x,$$

it is possible to obtain a generalized Gamma distribution as steady state. This specific model has been discussed in [5] and the exponential convergence of the solution to the steady state has been proven in [22]. This may motivate to choose the novel activation function $\sigma_N(\cdot)$ provided that the data is given by a generalized Gamma distribution.

5 Numerical Experiments

In this section we present two classical applications of machine learning algorithms, namely a clustering and regression problem. Furthermore, we validate the theoretical observations of the moment model. In addition we test our weight and bias update derived in the sensitivity analysis. Finally, we show that the Fokker-Planck type neural network is able to fit non trivial continuous

probability distributions. The weights and biases are given and constant in time. For the simulations we solve the PDEs models presented in this work by using a third order finite volume scheme [4], briefly reviewed below.

All the cases we consider for the numerical experiments can be recast in the following compact formulation:

$$\partial_t u(t,x) + \partial_x F(u(t,x),t,x) = \frac{\nu^2}{2}\partial_{xx} u(t,x) + kS(u(t,x)), \tag{22}$$

with ν and k given constants. Application of the method of lines to (22) on discrete cells Ω_j leads to the system of ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{U}_j(t) = -\frac{1}{\Delta x}\left[\mathcal{F}_{j+1/2}(t) - \mathcal{F}_{j-1/2}(t)\right] + \frac{\nu^2}{2}K_j(t) + kS_j(t), \tag{23}$$

where $\overline{U}_j(t)$ is the approximation of the cell average of the exact solution in the cell Ω_j at time t.

Here, $\mathcal{F}_{j+1/2}(t)$ approximates $F(u(t,x_{j+1/2}),t,x_{j+1/2})$ with suitable accuracy and is computed as a function of the boundary extrapolated data, i.e.

$$\mathcal{F}_{j+1/2}(t) = \mathcal{F}(U^+_{j+1/2}(t), U^-_{j+1/2}(t))$$

and $\mathcal{F}$ is a consistent and monotone numerical flux, evaluated on two estimates of the solution at the cell interface, i.e $U^{\pm}_{j+1/2}(t)$. We focus on the class of central schemes, in particular we consider a local Lax-Friedrichs flux. In order to construct a third-order scheme the values $U^{\pm}_{j+1/2}(t)$ at the cell boundaries are computed with the third-order CWENO reconstruction [4, 18].

The term $K_j(t)$ is a high-order approximation to the diffusion term in (22). In the examples below we use the explicit fourth-order central differencing employed in [17] for convective-diffusion equations with a general dissipation flux, and which uses point-values reconstructions computed with the CWENO polynomial.

Finally, $\mathcal{S}_j(t)$ is the numerical source term which is typically approximated as $\mathcal{S}_j(t) = \sum_{q=0}^{N_q}\omega_q S(\mathcal{R}_j(t,x_q))$, where x_q and ω_q are the nodes and weights of a quadrature formula on Ω_j. the four point Gaussian quadrature of order seven. We employ three point Gaussian quadrature formula matching the order of the scheme.

System (22) is finally solved by the classical third-order (strong stability preserving) SSP Runge-Kutta with three stages [13]. At each Runge-Kutta stage, the cell averages are used to compute the reconstructions via the CWENO procedure and the boundary extrapolated data are fed into the local Lax-Friedrichs numerical flux. The initial data are computed with the three point Gaussian quadrature. The time step Δt is chosen in an adaptive fashion and all the simulations are run with a CFL of 0.45.

5.1 Moment Model

We have solved the mean field neural network model in order to compute the corresponding moments. In this section we choose the initial condition to be the following Gaussian probability distribution:

$$g_0(x) = \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{(x-1^2)}{2}\right\}.$$

In section 3.3 we extensively discussed several aggregation phenomena of the moment model. As predicted by Proposition 2 we obtain with the choice $w = -1,\ b = 0$ a decay to zero of the energy, expected value and the variance as depicted in Figure 1. Furthermore, we have plotted in Figure 1 the case with $b = -\frac{m_1(t)}{m_0(0)}$ which guarantees the conservation of the first moment as discussed in Proposition 3. Figure 2 illustrates the time needed in order to reach a desired energy or variance level, which has been studied in Corollary 2 and **??**.

5.2 Machine Learning Applications

We present the kinetic approach for a classification and regression problem. In these section the activation function is chosen to be the hyperbolic tangent.

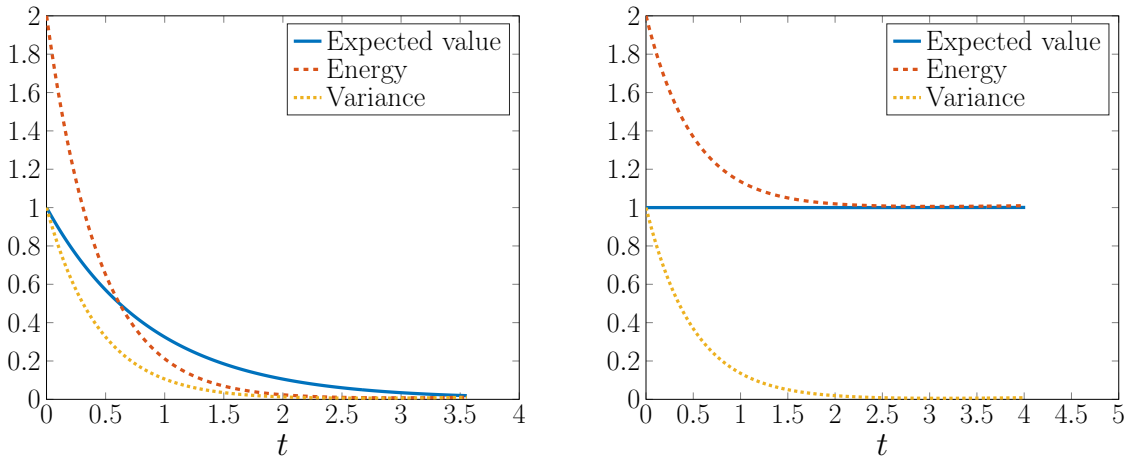

Figure 1: LHS: Moments of our PDE model with $\sigma(x) = x, w = -1, b = 0$. RHS: Moments of our PDE model with $\sigma(x) = x, w = -1, b = -\frac{m_1(t)}{m_0(0)}$.

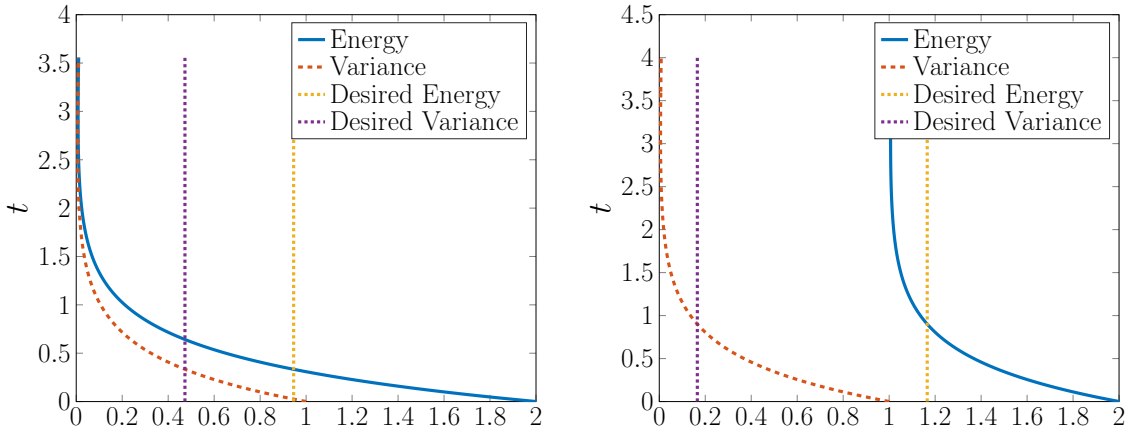

Figure 2: LHS: The energy and variance plotted against the desired values with $\sigma(x) = x, w = -1, b = 0$. RHS: The energy and variance plotted against the desired values with $\sigma(x) = x, w = -1, b = -\frac{m_1(t)}{m_0(0)}$.

5.2.1 Classification Problem

Consider a classification problem as follows. We measure a quantity (e.g. length of a car) and the object must be sorted with respect of the type (e.g. car or truck). The task of the neural network is to determine the type given a measurement. A training set might look like Table 1.

A probabilistic interpretation of the length measure can be obtained by a normalized histogram. Thus, the histogram shows the frequency of a classifier of certain measurement (see Figure 3). The continuous approximation of this histogram is the input of our mean field neural network model. The type could be a binary variable. In terms of our kinetic model the target is characterized by two Dirac delta distributions located at the binary values. Therefore, we introduce zero flux boundary conditions on the numerical scheme. The kinetic variable describes the distribution of the measurements (e.g. the length of vehicles). At final time we interpret this length as the decision of being a car or a truck. Thus, we introduce two measurements that determine the type. On the particle level we obtain the convergence of the neurons two these clusters (see Figure 4). The solution to the mean field neural network model is depicted in Figure 5.

Table 1: Example of input data.

Quantity	Classifier
3	car
3.5	car
5.5	truck
7	truck
4.5	car
8	truck
⋮	⋮

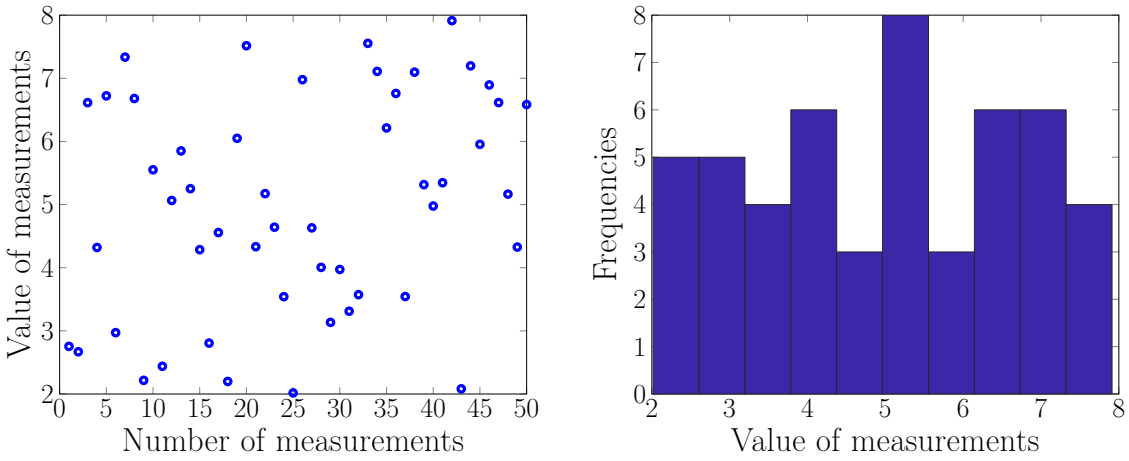

Figure 3: We consider 50 vehicles with measured length between 2 and 8. Mathematically, we sampled 50 realizations of a uniformly distributed random variable between 2 and 8. LHS: Length of vehicles plotted against the number of measurements. RHS: Histogram of the measured length of the vehicles.

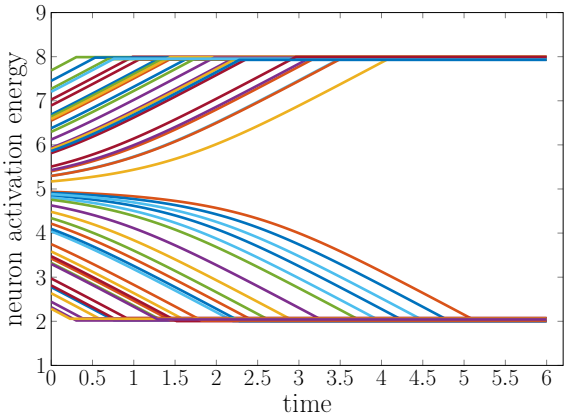

Figure 4: Trajectories of the neuron activation energies in the case of 50 measurements.

5.2.2 Regression Problem

We may have given measurements at fixed locations. These measurements might be disturbed possibly due to measurement errors as in Figure 6. The task of the neural network is to find the fit of those data points. It is not possible to solve this task with our model in one dimension

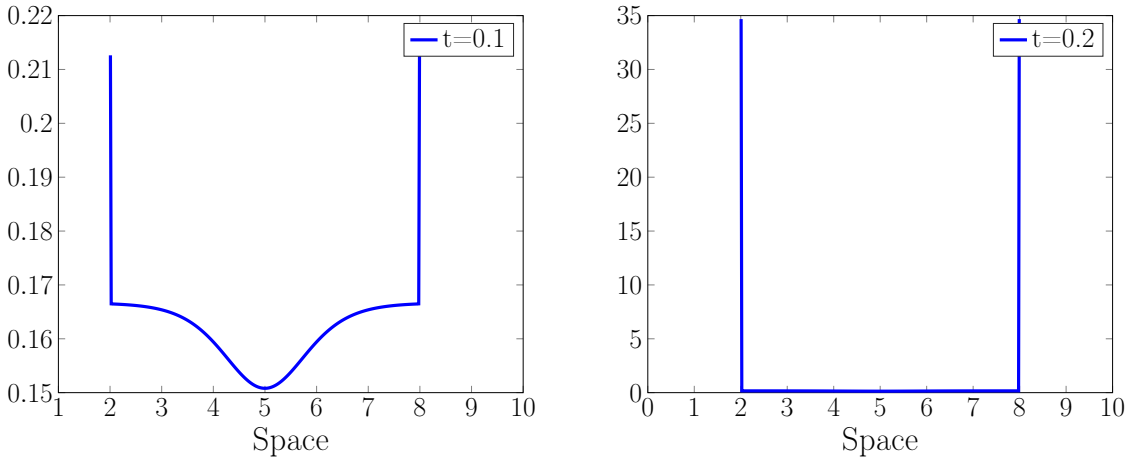

Figure 5: Solution of the mean field neural network model at different time steps. The initial value is a uniform distribution on $[2, 8]$ and the weight and bias is chosen as $w = 1,\ b = -5$.

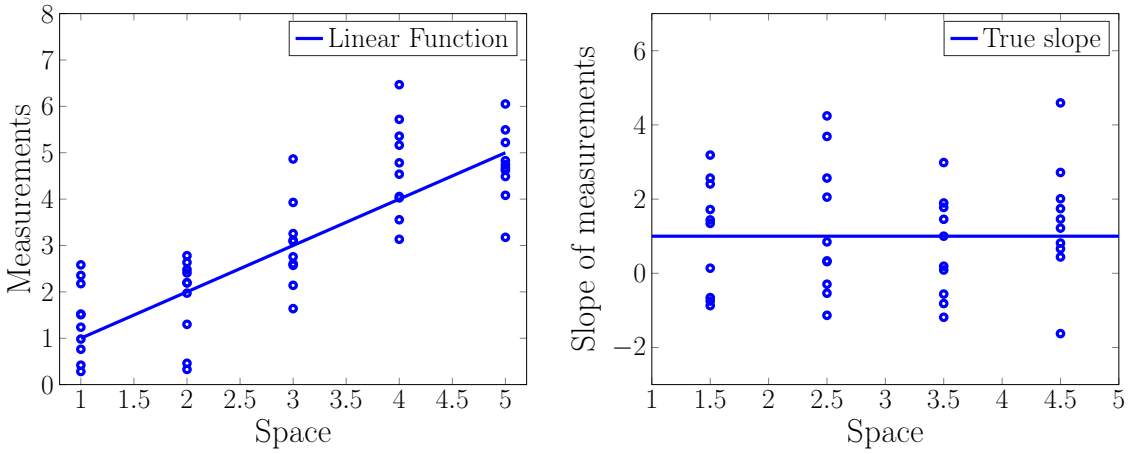

Figure 6: LHS: Regression problem with 50 Measurements at fixed positions around the linear slope. Measurement errors are distributed standard Gaussian. RHS: Numerical slopes computed out of the previous measurements.

since we have proven in Proposition 1 that the mean field neural network model only performs clustering tasks in the case of the identical and hyperbolic tangent activation function.

Therefore we transform the problem and assume a linear fit and aim to learn the slope of this fit by a neural network. The data is used to generate empirical slopes. These slopes are given by a probabilistic interpretation as in the histogram in Figure 7, being the input of the model. The target is a Dirac delta distribution located at $x = 1$. The solution converges to the target, see Figure 8. Thus, the location of the Dirac delta gives the correct slope of the graph in Figure 6.

5.3 Sensitivity Analysis and Update of Weights and Bias

We aim to present the benefit of the sensitivity analysis. We consider the regression problem as introduced in the previous section. The initial data is Gaussian with mean and variance equal to one, the target distribution is a Dirac delta centered at $x = 1$ and the weights and bias are $w = -1, b = 1$.

As Figure 8 shows, $g(t = 5, x)$ is close to the target. Then, we introduce as new target a

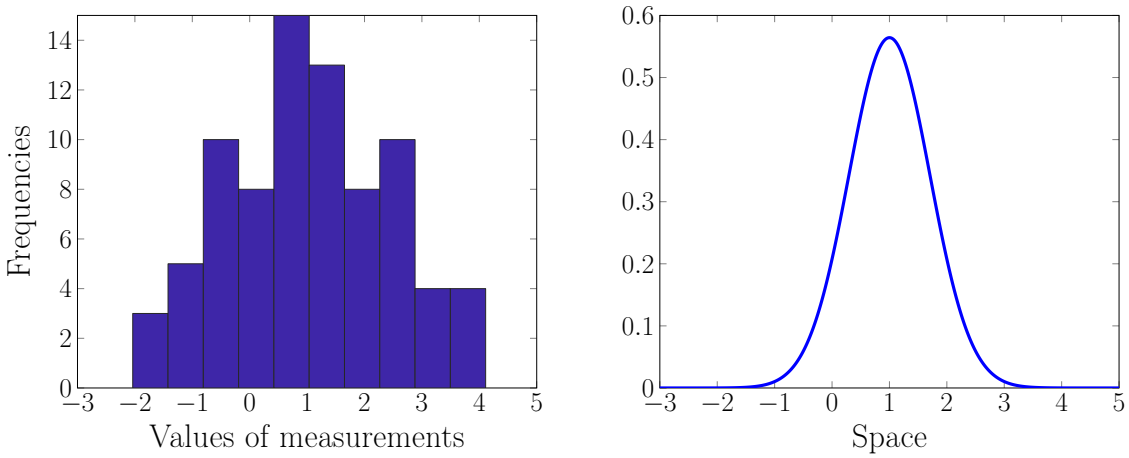

Figure 7: LHS: Histogram of numerical computed slopes with 100 measurements. RHS: Time continuous Gaussian distribution with mean one and variance one.

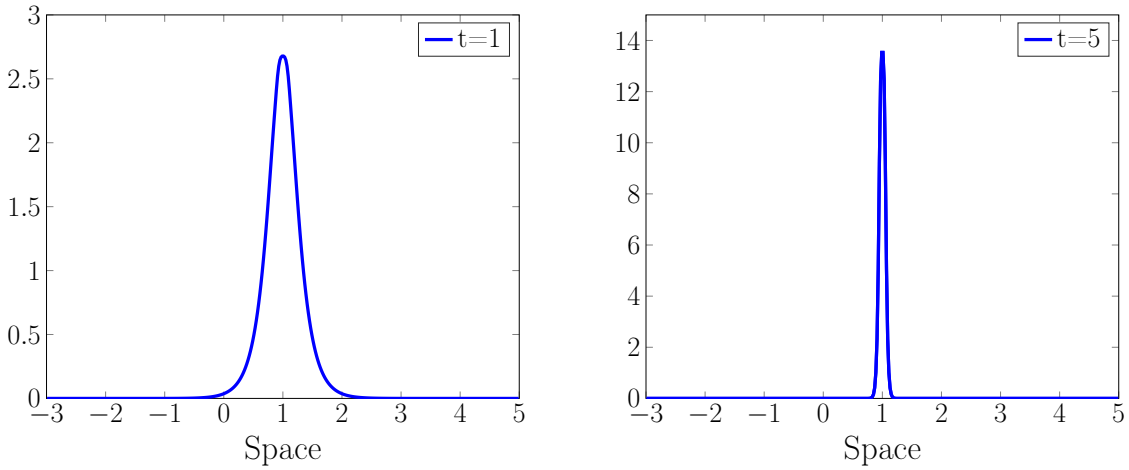

Figure 8: Solution of the mean field neural network model at different time steps with weight $w = 1$ and bias $b = -1$.

Dirac delta centered at $x = 2$ and use adjoint calculus with fixed step size $\gamma := 2$ to update the weights. The result of the mean field neural network model for different number of gradient steps are presented in Figure 9. Thus, the gradient step can be used in order to update weights and bias in case of changing the initial input or the target.

5.4 Fokker-Planck Type Neural Network

In this example we consider a standard normal Gaussian distribution as target and the input is uniformly distributed on $[-1, -\frac{1}{2}]$. As presented in Section 4.1.1 the Fokker-Planck type neural network model is able to obtain the Gaussian distribution as steady state. This directly leads us to the choice of the weight, bias and activation function. We need to choose the identity as activation function. This approach allows us to drive the initial input to the given target by simply fitting the two parameters w_∞ and b_∞.

Recall that, on the contrary, and as proven in Proposition 1, the mean field neural network can perform in the case of a hyperbolic tangent or identity activation function only clustering tasks. This means that for large times the distribution approaches a Dirac delta distribution

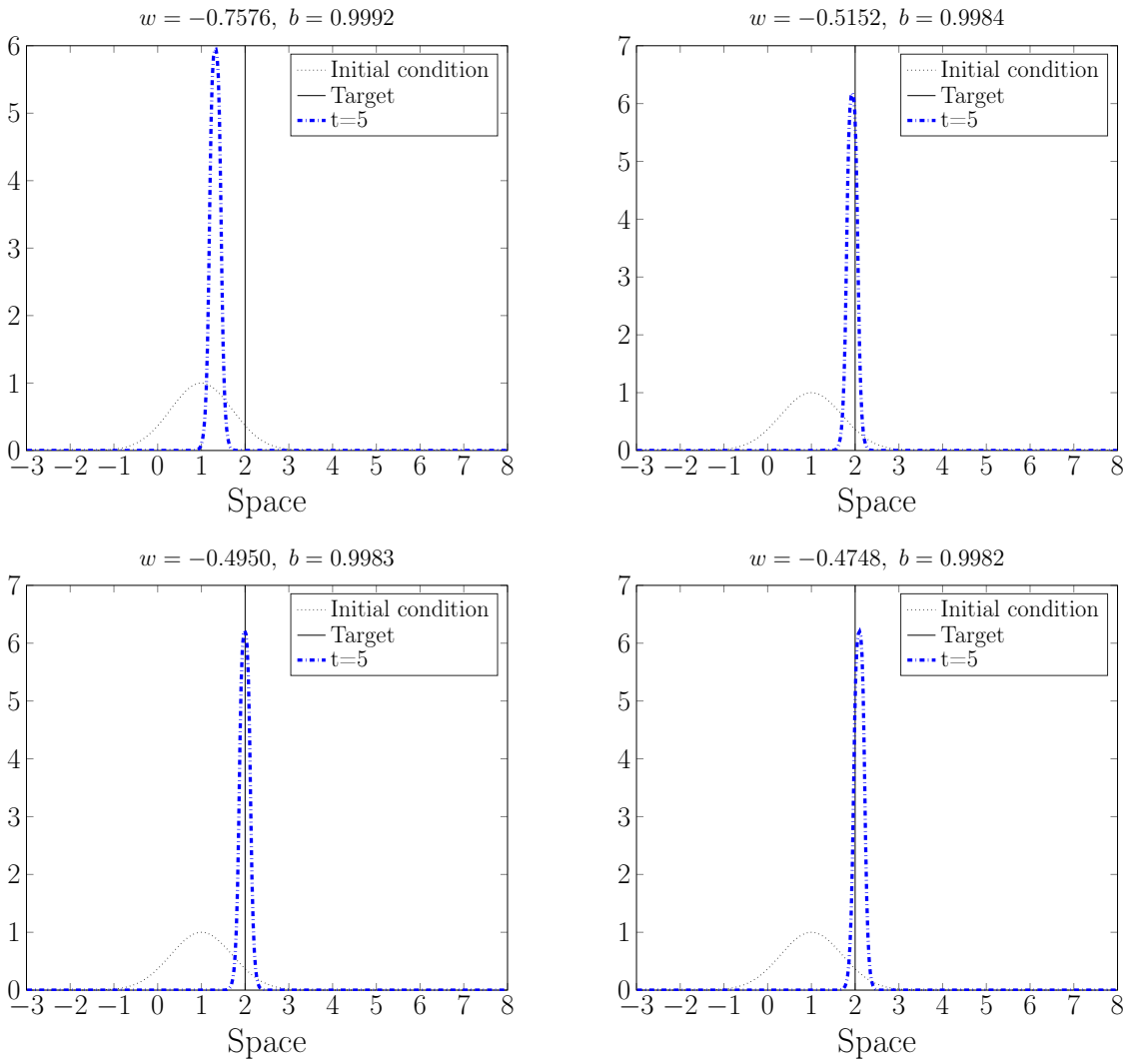

Figure 9: Results of the mean field neural network model with updated weights and bias in the case of a novel target.

and consequently it is not possible to fit a Gaussian distributed target by using the deterministic SimResNet model.

The histogram of sampled data is depicted in Figure 10. The solution of the Fokker-Planck neural network model for different time steps is presented in Figure 11 showing the convergence to the given target.

6 Conclusion

Starting from the classical formulation of a residual neural network, we have derived a simplified residual neural network and the corresponding time continuous limit. Then, we have taken the mean field limit in the number of measurements, thus switching from a microscopic perspective on the level of neurons to a probabilistic interpretation. We have analyzed solutions and steady states of the novel model. Furthermore, we have investigated the sensitivity of the loss function with respect to the weight and bias. Finally, we have derived a Boltzmann description of the simplified residual neural network and extended it to the case of a noisy setting. As consequence, non trivial

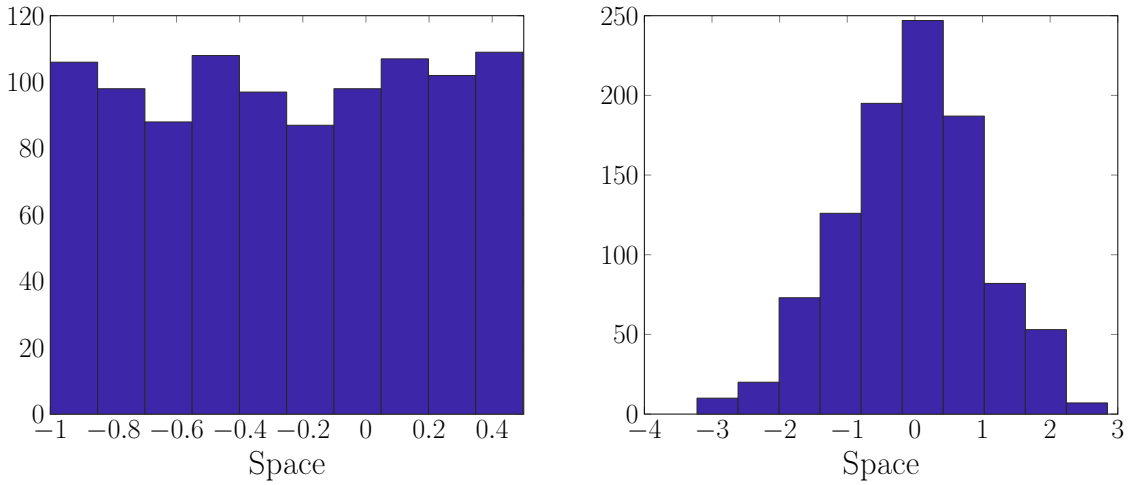

Figure 10: LHS: Histogram of 100 measurements uniformly distributed between $[-1, \frac{1}{2}]$. RHS: Histogram of the standard Gaussian distributed target values.

steady states have been obtained for the limiting Fokker-Planck type model. In the last section we have validated our analysis and have presented simple machine learning applications, namely regression and classification problems.

Our study may yield insights in order to understand the performance of residual neural networks. E.g. the moment analysis gives practical estimates on the simulation time and how to choose bias and weight. The gradient algorithm derived form the sensitivity analysis provides us with an update formula to recompute bias and weight: after changes in initial or target conditions. Probably most interestingly, we have seen that our mean field neural network model has in special situations only a Dirac delta function as unique steady state. In these cases the mean field neural network performs clustering tasks. In comparison to the mean field model, the Fokker-Planck neural network model is able to exhibit non-trivial steady states.

Acknowledgments

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612 and supported also by DFG HE5386/15.

M. Herty and T. Trimborn acknowledge the support by the ERS Prep Fund - Simulation and Data Science. The work was partially funded by the Excellence Initiative of the German federal and state governments.

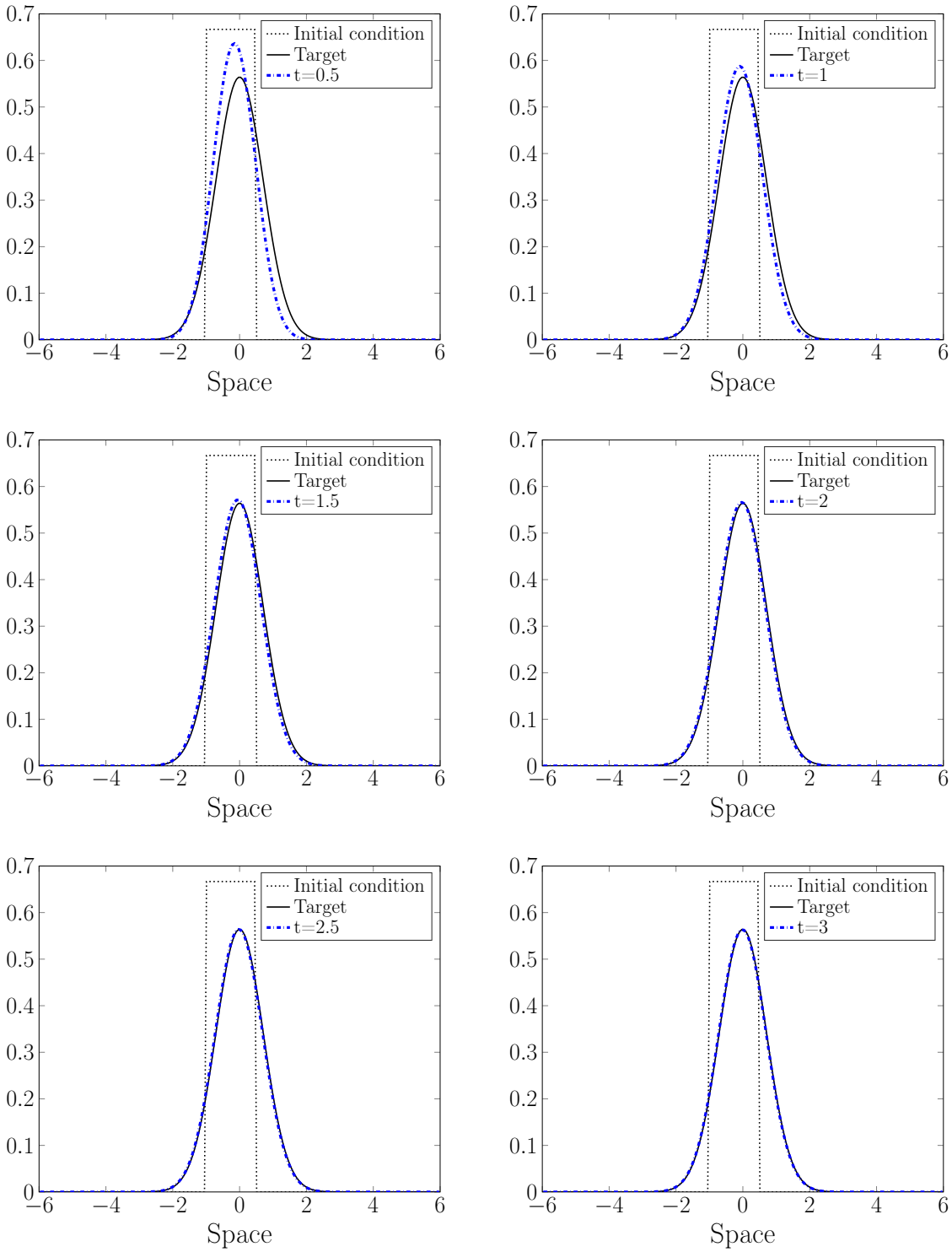

Figure 11: Solution of the Fokker-Planck neural network model at different times. Here, we have chosen the identity as activation function with weight $w = -1$, bias $b = 0$ and diffusion function $K(x) = 1$.

References

[1] D. Araújo, R. I. Oliveira, and D. Yukimura. A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193*, 2019.

[2] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[3] R. M. Colombo, M. Mercier, and M. D. Rosini. Stability and total variation estimates on general scalar balance laws. *Commun. Math. Sci.*, 7(1):37–65, 2009.

[4] I. Cravero, G. Puppo, M. Semplice, and G. Visconti. CWENO: uniformly accurate reconstructions for balance laws. *Math. Comp.*, 87(312):1689–1719, 2018.

[5] G. Dimarco and G. Toscani. Kinetic modeling of alcohol consumption. *arXiv preprint arXiv:1902.08198*, 2019.

[6] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455*, 2018.

[7] C. Gebhardt and T. Trimborn. Simplified ResNet approach for data driven prediction of microstructure-fatigue relationship. In preparation.

[8] F. Golse. On the dynamics of large particle systems in the mean field limit. In *Macroscopic and large scale phenomena: coarse graining, mean field limits and ergodicity*, pages 1–144. Springer, 2016.

[9] E. Haber, F. Lucka, and L. Ruthotto. Never look back - A modified EnKF method and its application to the training of neural networks without back propagation. Preprint arXiv:1805.08034, 2018.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[11] P.-E. Jabin. A review of the mean field limits for vlasov equations. *Kinetic & Related Models*, 7(4):661–711, 2014.

[12] K. Janocha and W. M. Czarnecki. On loss functions for deep neural networks in classifications. Preprint arXiv:1702.05659v1, 2017.

[13] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[14] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[15] A. V. Joshi. Machine learning and artificial intelligence, 2019.

[16] N. B. Kovachki and A. M. Stuart. Ensemble Kalman inversion: a derivative-free technique for machine learning tasks. *Inverse Probl.*, 35(9):095005, 2019.

[17] A. Kurganov and D. Levy. A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations. *SIAM J. Sci. Comput.*, 22(4):1461–1488, 2000.

[18] D. Levy, G. Puppo, and G. Russo. Compact central WENO schemes for multidimensional conservation laws. *SIAM J. Sci. Comput.*, 22(2):656–672, 2000.

[19] 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.

[20] S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[21] S. C. Onar, A. Ustundag, Ç. Kadaifci, and B. Oztaysi. The changing role of engineering education in industry 4.0 era. In *Industry 4.0: Managing The Digital Transformation*, pages 137–151. Springer, 2018.

[22] F. Otto and C. Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000.

[23] L. Pareschi and G. Toscani. Self-Similarity and Power-Like Tails in Nonconservative Kinetic Models. *J. Stat. Phys.*, 124(2-4):747–779, 2006.

[24] L. Pareschi and G. Toscani. *Interacting Multiagent Systems. Kinetic equations and Monte Carlo methods*. Oxford University Press, 2013.

[25] D. Ray and J. S. Hesthaven. An artificial neural network as a troubled-cell indicator. *J. Comput. Phys.*, 367(15):166–191, 2018.

[26] D. Ray and J. S. Hesthaven. Detecting troubled-cells on two-dimensional unstructured grids using a neural network. *J. Comput. Phys.*, 397, 2019. To appear.

[27] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *arXiv preprint arXiv:1804.04272*, 2018.

[28] L. Ruthotto, S. Osher, W. Li, L. Nurbekyan, and S. W. Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. Preprint arXiv:1912.01825, 2019.

[29] R. Schmitt and G. Schuh. *Advances in Production Research: Proceedings of the 8th Congress of the German Academic Association for Production Technology (WGP), Aachen, November 19-20, 2018*. Springer, 2018.

[30] J. Sirignano and K. Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 2019.

[31] R. J. Solomonoff. Machine learning-past and future. *Dartmouth, NH, July*, 2006.

[32] H. Tercan, T. Al Khawli, U. Eppelt, C. Büscher, T. Meisen, and S. Jeschke. Improving the laser cutting process design by machine learning techniques. *Production Engineering*, 11(2):195–203, 2017.

[33] G. Toscani. Kinetic models of opinion formation. *Commun. Math. Sci.*, 3(4):481–496, 2006.

[34] Q. Wang, J. S. Hesthaven, and D. Ray. Non-intrusive reduced order modelling of unsteady flows using artificial neural networks with application to a combustion problem. *J. Comput. Phys.*, 384:289–307, 2019.

[35] K. Watanabe and S. G. Tzafestas. Learning algorithms for neural networks with the Kalman filters. *J. Intell. Robot. Syst.*, 3(4):305–319, 1990.

[36] P. J. Werbos. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*, volume 1. John Wiley & Sons, 1994.

[37] Z. Wu, C. Shen, and A. Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.

[38] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.