Machine Learning: Science and Technology

Machine learning reveals complex behaviours in optically trapped particles

MACHINE
LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
24 May 2020

REVISED
9 August 2020

ACCEPTED FOR PUBLICATION
12 August 2020

PUBLISHED
8 October 2020

PAPER

Machine learning reveals complex behaviours in optically trapped particles

Isaac C D Lenton[1], Giovanni Volpe[2], Alexander B Stilgoe[1], Timo A Nieminen[1] and Halina Rubinsztein-Dunlop[1]

[1] School of Mathematics and Physics, The University of Queensland, Brisbane, 4072, Australia
[2] Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden

E-mail: uqilento@uq.edu.au

Keywords: optical tweezers, machine learning, deep learning, dynamics simulation, swelling microparticles, complex optical fields

Supplementary material for this article is available online

Abstract

Since their invention in the 1980s, optical tweezers have found a wide range of applications, from biophotonics and mechanobiology to microscopy and optomechanics. Simulations of the motion of microscopic particles held by optical tweezers are often required to explore complex phenomena and to interpret experimental data. For the sake of computational efficiency, these simulations usually model the optical tweezers as an harmonic potential. However, more physically-accurate optical-scattering models are required to accurately model more onerous systems; this is especially true for optical traps generated with complex fields. Although accurate, these models tend to be prohibitively slow for problems with more than one or two degrees of freedom (DoF), which has limited their broad adoption. Here, we demonstrate that machine learning permits one to combine the speed of the harmonic model with the accuracy of optical-scattering models. Specifically, we show that a neural network can be trained to rapidly and accurately predict the optical forces acting on a microscopic particle. We demonstrate the utility of this approach on two phenomena that are prohibitively slow to accurately simulate otherwise: the escape dynamics of swelling microparticles in an optical trap, and the rotation rates of particles in a superposition of beams with opposite orbital angular momenta. Thanks to its high speed and accuracy, this method can greatly enhance the range of phenomena that can be efficiently simulated and studied.

Since their invention [1], optical tweezers have been employed to discover and study new phenomena, including molecular motors [2, 3], thermodynamics of small systems [4, 5], and microscopic organisms [6, 7]. In the study of these systems, numerical simulations are frequently used both to plan and to interpret experiments. The particle trajectory is often obtained from Brownian dynamics simulations, where it is simulated as a series of sequential Brownian and deterministic steps [8, 9], which respectively account for the thermal motion and the deterministic optical forces acting on the particle. Typically, the most computationally expensive part of the calculation is determining the optical forces; furthermore, due to the randomness introduced by the thermal motion, these forces need to be calculated sequentially. For complex optical phenomena, multiple trajectories with a large number of time steps may be needed in order to have sufficient statistics. For example, when the inertia contributes to the particle dynamics, such as for the motion of particles in vacuum, the particle trajectories need to be simulated with sub-nanosecond resolution, so that even the simulation of a 1-minute trajectory requires on the order of 10^{12} time steps [10].

In simple situations, the optical trap can be approximated with a harmonic potential, so that the optical force can be easily estimated as $F_{\mathrm{ot}} = -kx$, where k is the stiffness of the optical trap and x is the displacement of the particle from the equilibrium position [9, 11]. However, there are many scenarios where this model is insufficient, for example, when modelling a particle escaping an optical trap [12] or the motion of a particle in a beam with orbital angular momentum [13, 14]. While ad hoc phenomenological force-field models are often employed to simulate these phenomena [15, 16], it is necessary to employ semi-analytical methods

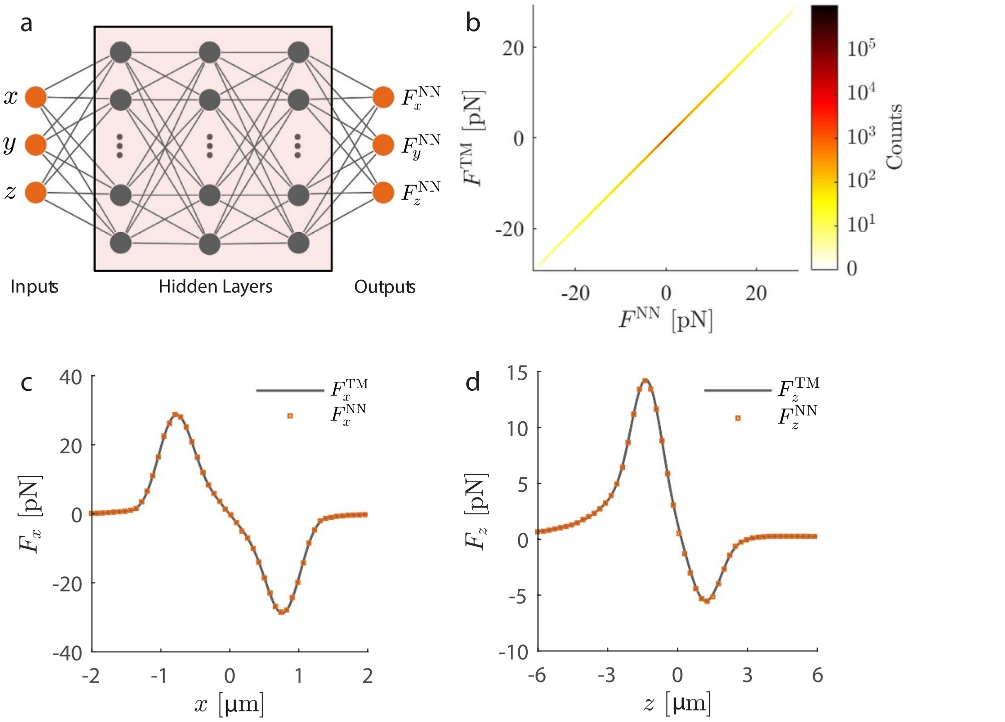

Figure 1. Estimation of optical forces using a neural network. (a) Illustration of a fully connected neural network with three inputs (x, y, z), three outputs (f_x, f_y, f_z), and three hidden layers. (b) Log-density plot comparing neural network force estimates (F^{NN}) with the ground-truth forces calculated with the T-matrix method (F^{TM}) featuring an excellent agreement between the two methods for 10^6 randomly sampled force vectors (viz. 3×10^6 unique counts for the force components). (c) Axial and (d) radial force–position curves calculated using the neural network (orange symbols, F_x^{NN} and F_z^{NN}) featuring an excellent agreement with those calculated with the T-matrix method (gray line, F_x^{TM} and F_z^{TM}).

such as the T-matrix method to obtain a physically accurate simulation [17–20]. However, these semi-analytical methods are prohibitively slow for running extensive dynamics simulations [21].

Another approach is to use local interpolation, which involves sampling the force distribution at a number of discrete points on either a grid (structured interpolation) or at random locations (unstructured interpolation) and using an interpolation function (e.g. a linear or cubic polynomial) to estimate intermediate values [22]. Unlike Brownian dynamics simulations, which require sequential force calculations, the force values required for interpolation can be calculated in parallel, significantly reducing run-time. While interpolation works well for low-dimensional problems (1 to 3 DoF) [22], it is difficult to implement efficiently for higher-dimensional problems because of runaway requirements on the memory storage and the lack of ready algorithms implemented in the major software packages.

In this article, we demonstrate an alternative method based on machine learning. We show that a neural network (NN) can be trained to calculate optical forces with greater accuracy than phenomenological methods, in significantly less time than exact methods, and with less memory requirements than interpolation methods. To demonstrate the power of this approach, we employ it to study two phenomena that would be prohibitively slow to accurately simulate otherwise: the escape dynamics of swelling micro-particles in an optical trap, and the hopping rates of particles in a superposition of Laguerre-Gaussian beams.

While standard methods require explicit formulas to determine the optical forces starting from the physical parameters of the particle and trapping beam, in this work we employ machine-learning models that are trained to automatically determine the optimal rules to calculate the optical forces through a large series of known samples, *i.e.* pairs of physical parameters and corresponding optical forces (which can be obtained either from an experiment or an accurate numerical simulation). Specifically, we employ NNs because they have been one of the most successful tools for machine learning in recent years [23, 24]. NNs have outperformed standard approaches to track particles [25, 26], to enhance microscopy [27], to design electromagnetic nano-structures [28, 29], to predict optical scattering [30], and to calculate hydrodynamic forces [31]. Here, we employ dense fully-connected NNs, an example of which is illustrated in figure 1(a). These NNs consist of a series of fully-connected layers of neurons, where the output of each neuron is a non-linear function of the weighted sum of the neuron's inputs. The weights are iteratively adjusted (trained) until the NN learns to associate the correct optical force to each set of physical parameters. In this work, we

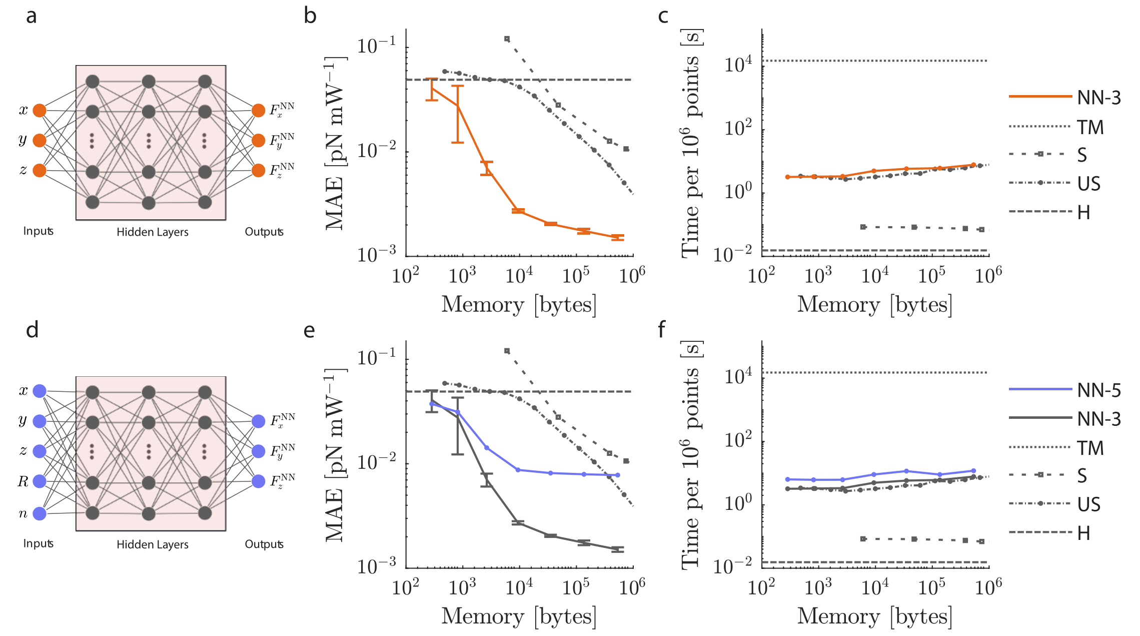

Figure 2. Improved performance and accuracy of neural networks compared to other methods. (a) NN with 3 DoF (position x, y, z): (b) as a function of their memory footprint, this NN (NN-3, orange solid line) outperforms in terms of mean absolute error (MAE) structured interpolation (S) and unstructured interpolation (US) as well as the harmonic model (H); (c) furthermore, this NN (NN-3, orange solid line) is several orders of magnitude faster than the T-matrix (TM) and similarly fast as the unstructured interpolation (US), while the structured interpolation (S) and the harmonic approximation (H) are even faster. (d) NN with 5 DoF (position x, y, z, particle radius R, refractive index n): (e) its MAE and (f) its speed are still comparable to those of the 3 DoF NN, while the other methods cannot be easily employed as discussed in the text; for convenience of comparison, the gray lines in (e) and (f) report the results in (b) and (c), respectively. Error bars in (b) show the variation between 10 NNs trained on the same training data (see Methods for details).

generate and train three different networks using optical forces obtained from exact T-matrix simulations [19, 32]. Networks are trained using Keras with a Tensorflow backend [33] (see Methods).

To evaluate the accuracy of this approach, we first explore the simple scenario of the optical forces acting on a spherical particle in an optical trap generated by a Gaussian optical beam. The NN, shown in figure 1(a), consists of 3 layers of 256 neurons connected to three input and three output neurons. The three inputs are the three-dimensional particle position (x, y, z) and three outputs are the components of the optical force acting on the particle ($F_x^{NN}, F_y^{NN}, F_z^{NN}$). We train this NN using 10^6 samples randomly distributed around the beam focus. Figure 1(b) compares the NN predicted forces with the exact T-matrix forces for 10^6 previously unseen samples, qualitatively demonstrating very good agreement between the NN predictions and the T-matrix method. Figures 1(c) and (d) further demonstrate the excellent agreement between the NN (orange symbols) and the exact T-matrix method (gray lines) both along the transverse x-direction (figure 1(c)) and along the axial z-direction (figure 1(d)), which extends well beyond the linear-force regime around the focal point.

Figure 2 compares more quantitatively the NN and alternative standard methods in terms of both accuracy and speed as a function of their memory footprint. We start by considering the 3 DoF NN discussed in figure 1. By varying the number of neurons in the 3 intermediate layers, we directly control the NN complexity and correspondingly the required memory. This allows for a straightforward comparison between the NN memory footprint and the memory needed to store the interpolation data (see Methods). The resulting mean absolute error (MAE) as a function of the memory footprint is shown by the orange line in figure 2(b) (NN-3). To put this result into context: in a typical optical tweezers experiment we might expect experimental noise to be between 10^{-2} and 10^{-1} pN mW^{-1}; achieving a MAE of almost 10^{-3} pN mW^{-1} for some of the larger networks demonstrates that NNs are more than suitable for modelling these kinds of experiments. The NN accuracy is strongly dependent on the training sample distribution: for optimal training performance, the samples should be clustered around regions where the force varies rapidly. Since in a Gaussian-beam optical trap the forces are largest and vary the most around the beam focus, a convenient choice for the sample distribution is Gaussian-distributed samples around the focus (see Methods). Similar considerations apply for the choice of points for unstructured interpolation (see Methods). For all memory footprints, the NN greatly outperforms both structured interpolation (S, gray squares) and unstructured interpolation (US, gray circles). This improved performance is due to the fact that, while unstructured interpolation requires storing all training samples indefinitely, NN only needs to see the training points during training and is able to encode the same information in a network using significantly less memory. For reference, we show also the MAE of the harmonic model (H, dashed line). We can observe that, while for sparse grids (low memory footprint) both interpolation methods perform worse than the harmonic model, the NN already outperforms it even for the smaller memory footprints.

Figure 2(c) compares the performance of the various methods in terms of their computational speed on a standard laptop computer (3.4 GHz processor, 16 GB RAM, see Methods). The exact T-matrix method is the slowest (TM, gray dotted line, ≈ 10000 s to calculate 10^6 samples), while the harmonic approximation is the fastest (H, gray dashed line, ≈ 0.01 s to calculate 10^6 samples). For both these methods the memory footprint is fixed. The NN (NN-3, orange line) and unstructured interpolation (US, gray circles) perform similarly ($\approx$ 10 s to calculate 10^6 samples) and about three orders of magnitude faster than the T-matrix method almost independently of the memory footprint. Structured interpolation (S, gray squares) is faster than both these methods, but at the expense of a much lower accuracy (figure 2(b)).

The results in figures 2(b) and (c) show that the NN is better in terms of accuracy and similar in terms of speed to unstructured interpolation for systems with ≤ 3 DoF. The NN gains a real edge when increasing the number of DoF. For example, we can consider a problem with 5 DoF, corresponding to the particle position, x, y and z, its radius R, and its refractive index n. In such case, structured interpolation becomes unfeasible because of memory constraints (a 5-dimensional grid with comparable resolution reaches quickly into the Terabytes). Similar memory-management problems emerge when considering unstructured interpolation algorithms. On the other hand, a NN (figure 2(d)) can still be trained with enough samples to learn the 5-dimensional dependence of the force field, achieving good accuracy (figure 2(e)) at a similar speed as in the 3-DoF case (figure 2(f)). The additional input parameters increase the number of multiplications needed to evaluate the network resulting in a slight increase in evaluation time, as demonstrated in Figure 2(f). The only minor difficulty encountered with training these larger networks is generating enough training data; fortunately, this task can easily be parallelised.

We now turn our attention towards scenarios that are almost impossible to accurately simulate without the NN approach. First, in figure 3, we consider a particle whose size and refractive index gradually change while held in an optical trap. This scenario is relevant to problems including modelling hydrogel particles swelling over time in optical tweezers or the study of cells undergoing osmotic stresses. More generally, this is an example of a problem where particles deform as they move through optical fields, such as the study of water-glycerol droplets [34] and of the deformation and growth of cells or microorganisms [35]. Using the 5-DoF NN shown in figure 3(a) (the same as that we employed in figures 3(d)–(f)), we simulate the escape trajectories of a swelling particle held in an optical trap in the presence of a fluid flow (see Methods). As schematically shown in figure 3(b), a swelling particle, whose radius increases and whose refractive index decreases, is gradually pulled out of the trap when the restoring optical force decreases. This scenario is difficult to explore experimentally or computationally due to the large number of repetitions required to determine the statistics of the escape trajectories. For a rapid swelling (figures 3(c)-(e)), the particle escapes the trap almost immediately after the flow is turned on. For a slow swelling instead (figures 3(f)–(h)), the particle initially remains trapped at the edge of the beam before eventually escaping. Without the NN approach, it is practically impossible to collect enough statistics to clearly see how the escape trajectories change. For example, using exact T-matrix calculations it takes us ≈ 50 s to simulate just 5 trajectories (figures 3(d) and (g)), the same amount of time during which we can simulate 10^4 trajectories with the NN (figures 3(e) and (h)). The latter amount of statistics are essential, for example, to determine the final particle distribution with a high degree of accuracy.

The second scenario we consider involves the mixing of two beams with opposite orbital angular momentum (OAM) [36], which generate interesting interference patterns that can be used, for example, to design ratchet-like micro-motors [37, 38]. Figure 4(a) shows a schematic of the experiment. A particle is placed in a mixture of two Laguerre-Gaussian beams of order ± 5 mixed into a single beam with a fixed power, so that the resulting beam is $\alpha\,\mathrm{LG}_{0,+5} + (1-\alpha)\,\mathrm{LG}_{0,-5}$, where $\alpha \in [0, 1]$. Different values of α lead to different orbital rates ω for the optically trapped particle. Figure 4(b) compares the results of an experiment (black symbols), where we trap a particle pushing it against the microscope slide using the laser beam (see details in Methods), with those we obtain employing a 4-DoF NN (shown in the inset of figure 4(b)) to extensively simulate this system (orange line). We obtain excellent agreement with the experimental results. This strong agreement is further illustrated by the time-averaged experimental (figure 4(c)) and simulated (figure 4(d)) trajectories, which represent ≈ 20 s of a single particle's trajectory, revealing three distinct behaviours: When $\omega \approx 0$ ($\alpha = 0.5$), we have a series of confining potential wells resulting from the even mixture of two beams with opposite OAM [38]. When ω is significantly larger than 0 (e.g. $\alpha = 1$), we have a smooth inclined potential around the beam, where the particle slides at an approximately constant rate [13, 36]. Finally, in intermediate regions (e.g. $\alpha = 0.7$), we obtain a washboard potential, where the particle is

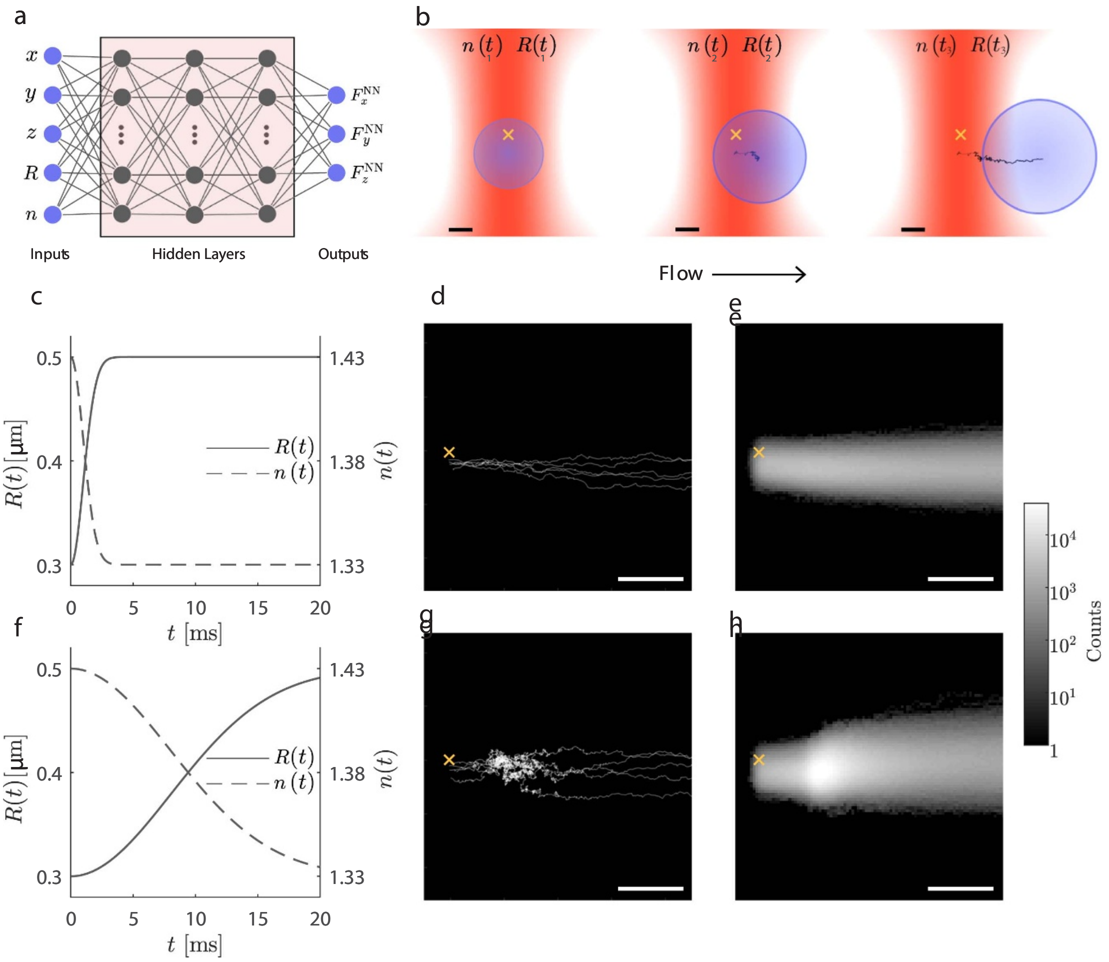

Figure 3. Swelling particle escape from an optical trap simulated with a 5-DoF neural network. (a) 5-DoF NN with inputs for particle position, x, y and z, radius R and refractive index n. (b) A swelling particle is trapped in the presence of a flow; as the particles swells, its radius $R(t)$ increases and its refractive index $n(t)$ decreases, so that the restoring optical force decreases and, eventually, the particle escapes from the trap. The beam is propagating downwards and the trap centre is marked by the orange cross. (c) Time dependence of the particle properties $R(t)$ (solid line) and $n(t)$ (dashed line) for a case in which the particle swells quickly. (d) Example of 5 escape trajectories computed using the T-matrix method and (e) 10^4 trajectories simulated using the 5-DoF NN taking about the same computational time ($\approx$ 50 s). (f)–(h) Corresponding results for a particle swelling more slowly. The scale bars in (b), (d), (e), (g) and (h) represent 0.2 μm.

metastably trapped in local minima, but on average moves around the beam [39]. The NN allows us to rapidly explore the effect of different parameters (including height of the beam relative to the microscope slide, beam power, and viscous drag) on the observed rotation rates and trajectories. Further still, the NN allows us to simulate significantly more accurate probability distributions: For example, figure 4(e) shows the average of 100 simulated trajectories, clearly displaying the steady-state behaviour of this system. This system demonstrates how the NN approach can be used to model and understand the behaviours of particles in complex light fields such as those used in references [40, 41].

Thus far, we have focused on how NNs can empower fast and accurate simulations for designing, analysing and modelling experiments. Another key advantage of the NN approach is its small memory footprint, which allows easy distribution of pre-trained NNs to calculate optical forces. In this way, multiple users can take advantage of the initial training cost and perform sophisticated numerical simulations on readily available hardware. This enables more collaborative workflows and the possibility of integrating numerically accurate simulations into interactive online demonstrations or teaching material.

In conclusion, we have shown NNs to be a valuable tool for simulating optical forces enabling longer, more accurate, and more memory-efficient simulations. Compared to interpolation and other methods, NNs are easier to implement for problems with many DoF and they use significantly less memory while still performing with similar evaluation times to the most efficient interpolation techniques. In particular, NNs can be used to supplement experiments, including to design trapping configurations or to analyse experimental results. While here we considered only spherical particles, our method can be extended to particles with other shapes by adding rotations as inputs and torques as outputs. Further investigation into

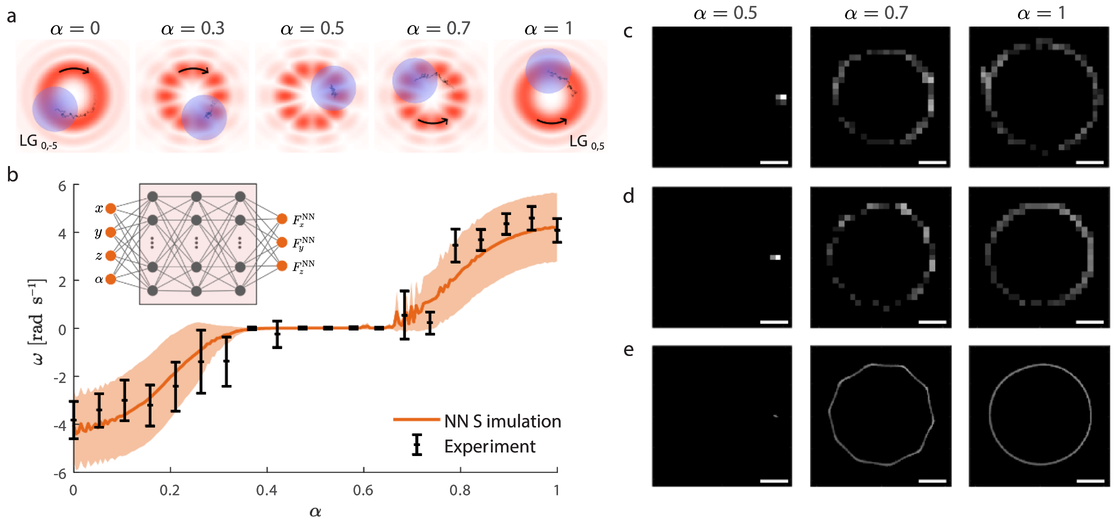

Figure 4. Dynamics of a particle in a superposition of Laguerre-Gaussian beams. (a) Schematic of an experiment with a particle held by a superposition of two Laguerre-Gaussian beams of order ± 5 with opposite orbital angular momentum, weighted by the parameter α so that the total beam is $\alpha\,\mathrm{LG}_{0,+5} + (1-\alpha)\,\mathrm{LG}_{0,-5}$. (b) Rotation rates obtained from experiments (black symbols) and from neural-network-based simulations (orange line). The error bars represent standard errors. Inset: schematic of the employed 4-DoF NN. (c) 20-s experimental trajectory of a particle for various α, leading to a confining potential ($\alpha = 0.5$), a washboard potential ($\alpha = 0.7$), and an inclined potential ($\alpha = 1$). (d) Corresponding 20-s simulated trajectory for a single particle. (e) Average of 100 simulated trajectories clearly showing the steady state behaviour in the various cases. The scale bars in (c)-(e) represent 1 μm.

different network architectures and training could further improve accuracy and decrease the number of points needed for training. More broadly, the method described here is general enough to be applicable to a wide range of fields including simulations of optomechanics, optoelectronics or acoustics. Although we focus on demonstrating this method for simulating optically trapped particles, we believe this technique will be of interest to the wider optics and photonics community and can be used to model other complex optical processes.

1. Methods

1.1. Training of neural networks

NN are trained in Python using Keras (version 2.2.2) [33] with TensorFlow backend (version 1.5.0). An example Jupyter Notebook for the 3-DoF NN is provided in Supplementary Information (stacks.iop.org/MLST/1/045009/mmedia). Training each NN consists of three main parts: loading the data, setting up the NN model, and training the NN on the training data.

Training data for each NN is loaded from a data file containing exemplary inputs and outputs. Values are read from the file, scaled to near unity, randomly shuffled and split into a validation and training set (with a 1/9 split). For our NNs, training data consists of simulation data generated in Matlab (R2018a) using the Optical Tweezers Toolbox (OTT, version 1.5) [19, 32]. Importantly, training data could have been generated with another process, for example, by position and force measurement in an experiment.

Setting up the NN model involves specifying the shape and parameters for each NN layer. In this work we explored dense fully connected NNs with few intermediate layers. Details about the number of layers and neurons per layer can be found in subsequent sections. Intermediate layers use rectified linear unit (ReLU) activation functions. The final layer uses a linear activation function. Single precision values are used for NN weights for compatibility with available training hardware.

The NN is trained by optimising the NN weights to minimise a loss function. We use the Keras implementation of the Adam optimiser [33] with default parameters and mean squared error for the loss function. For training the NN, we use gradually increasing batch sizes. The training set is used for training the NN and the validation set is used to evaluate the accuracy of the model at the end of each training epoch. We train the NNs on NVIDIA Tesla V100 graphics cards.

1.2. Degrees of freedom neural network

These are NN with 3 inputs, 3 layers of hidden neurons and 3 outputs. For the comparison presented in figures 2(b) and (c), we train 7 different NNs with 4, 8, 16, 32, 64, 128, and 256 neurons in each hidden layer. Figure 1 uses 256 neurons for each hidden layer. Training data is simulated with OTT for a spherical particle (radius 818 nm, refractive index 1.5) in a circularly polarised Gaussian beam (numerical aperture 1.02, medium refractive index 1.3, wavelength in medium 818 nm). Training data consists of 10^6 data points randomly distributed according to a normal distribution with 3.19 μm standard deviation and centred on the beam focus. Error bars in figure 2(b) show average results of 10 NNs trained on the same data.

1.3. Mean average error

Mean average error (MAE) is calculated according to

$$\text{MAE} = \frac{1}{3N_{\text{samples}}} \sum_{i}^{N_{\text{samples}}} \sum_{j=\{x,y,z\}} \left| F_{\text{TM}}^{i,j} - F_{\text{NN}}^{i,j} \right|$$

where F_{TM} is the T-matrix optical force calculated with OTT, F_{NN} is the NN estimate, and N_{samples} is the number of samples in the validation data set. In figures 1(b), 2(b) and 2(e), the validation data set consists of 10^6 additional locations not used in the training of the NN.

1.4. Calculation of memory footprint

Memory footprint depends on the method. Optical tweezers harmonic models typically use between 1 and 9 values (*i.e.* between 8 and 72 bytes if using double precision values). Structured interpolation stores values on an evenly spaced grid (*i.e.* a matrix), so that the required memory is directly related to the data type and the number of values stored (*i.e.* an interpolation grid with 100 × 100 × 100 locations, each location containing 3 double precision values for F_x, F_y, and F_z, has an approximate memory footprint of 24×10^6 bytes ≈ 24 MB). Unstructured interpolation requires storing the location and value at each grid point (*i.e.* for 10^6 unstructured double precision position and force values, we need to store 48×10^6 bytes ≈ 48 MB). For the 3 DoF NN, the number of parameters (NN weights and biases) scales as

$$\text{Net}_3 = 2\,\text{N}^2 + 9\,\text{N} + 4$$

where N is the number of neurons per hidden layer. The memory footprint for the NN is approximately the number of parameters multiplied by the memory per parameter (4 bytes for single precision values). The 5 DoF NN has an additional 2 N parameters corresponding to the weights connecting to the two additional inputs, so that the total is

$$\text{Net}_5 = 2\,\text{N}^2 + 11\,\text{N} + 4.$$

1.5. Sampling distribution

For both unstructured interpolation and the NN approach, the choice of sampling locations affects the accuracy of the predicted forces. With infinite samples, interpolation will exactly reproduce the target distribution. With a suitable architecture and training, a NN is capable of emulating interpolation; consequently, the upper bound for the accuracy of a NN is related to the number and distribution of points used for training. For the NN in this work, points are distributed around locations where the particle is likely to be and where the forces vary most rapidly. For Gaussian beam NNs (3 DoF and 5 DoF), this corresponds to a Gaussian distribution centred at the beam focus. For the 4 DoF NN, this corresponds to a toroidal shaped distribution; the shape of the toroid was determined from preliminary simulations which indicated where the particle was likely to be found.

1.6. Calculation of evaluation time

To measure the evaluation time shown in Figures 2(c) and (f), we used a 3.4 GHz Intel Core i7-6700 personal computer with 16 GB RAM running implementations of each algorithm in Matlab. Matlab is chosen to allow simple comparison between performance of OTT (which is a Matlab package [19, 32]) and the other methods. Structured and unstructured interpolation use Matlab's built-in implementations; and a harmonic model with 1 parameter per dimension is implemented. Evaluation locations matching those used for the MAE comparison are used. Methods are not explicitly parallelised. Additional performance may be achievable using specialised implementations or explicit parallelisation.

1.7. Degrees of freedom neural network

NN with 5 inputs, 3 hidden layers and 3 outputs. For figure 2, number of neurons in hidden layers is varied, similar to the 3 DoF NN. For figure 3, each hidden layer has 256 neurons. Training data is simulated with OTT for a spherical particle in a circularly polarised Gaussian beam (numerical aperture 1.02, wavelength in medium 800 nm, medium refractive index 1.33). Training data consists of 10^7 data points with: refractive index and radius values randomly sampled from a uniform distribution with the range 1.33–2.00 and 0.1–1.0 μm, respectively; and positions randomly sampled from a normal distribution centred at the focus and standard deviation of 5 μm.

1.8. Escape trajectory simulation

The particle refractive index and radius are given by

$$n(t,\sigma) = n_{\text{water}} + 0.1e^{-t^2/2\sigma^2}$$

and

$$R(t,\sigma) = 0.5 - 0.2e^{-t^2/2\sigma^2}\ \mu\text{m}$$

respectively, where t is the simulation time, σ is the decay (growth) rate of the refractive index (size) of the particle, and n_{water} is the refractive index of the medium. The particle is initially trapped by the beam. At 1 ms a flow is enabled causing the particle to move towards the edge of the trap and eventually escape.

1.9. Laguerre-Gaussian beam mixture experiment

Particles are confined in two dimensions using mixtures of LG beams generated using a spatial light modulator (SLM). These beams carry orbital angular momentum which is transferred to the particle. Supplementary figure 1 shows an overview of the experimental setup which has been previously described in references [42, 43]. Beams are created with a liquid crystal SLM (Meadowlark Optics, 512x512 HSP512L, high-speed SLM), illuminated by a 1064 nm fibre laser (YLR-10-1064-LP, 10 W, IPG Photonics). The phase pattern displayed on the SLM consists of a superposition of LG+/-5 azimuthal gratings. Although methods exist for generating pure LG modes with SLMs [44], our method can be applied to model any arbitrary incident beam including these impure LG modes (see details below). The hologram is imaged onto the back-aperture of a 1.2 numerical aperture water immersion objective (Olympus UPlanSApo 60 ×) and focused into the sample. The sample consists of 2 μm polystyrene spheres (refractive index ≈ 1.59) in water (refractive index ≈ 1.33) prepared between two microscope coverslips. Particles are axially confined by pushing them against the microscope coverslip with the optical trapping beam. A surfactant is mixed with the water to reduce the chance of particles sticking to the coverslip.

1.10. Degrees of freedom neural network

NN with 4 inputs, 3 hidden layers with 128 neurons per layer, and 3 outputs. Training data is simulated for a spherical particle (radius 1 μm, refractive index of particle 1.59) held in different mixtures of LG beams (refractive index of medium 1.33, wavelength in medium 800 nm). In order to accurately model the experiment, training data is generated using beams modelled on the experimentally realised beams (using the paraxial point-matching code from OTT and an approximation for the phase and intensity of the beam at the objective back aperture, numerical aperture 1.2). The experimentally realised beams are approximately LG+/-5 beams except for variations in the intensity profile due to the incident illumination. 10^7 training points are generated. Due to the computational expense in calculating different beam mixtures, only 21 different alpha values are present in the data set, evenly spaced from 0 to 1. For each data point, alpha is randomly selected with a discrete uniform distribution, and position is uniformly randomly chosen from a toroidal volume with inner radius 0.5 μm and outer radius 2 μm and height 1 μm centred on the beam axis 0.5 μm from the focus. The toroid corresponds to locations where the particle is trapped in the LG ring and allows multiple axial positions to be explored in simulations.

1.11. Laguerre-Gaussian beam mixture simulation

The simulation is designed to model the experimental observations. The beam power is set to match the experiment rotation rate for a pure LG beam. Axial position is chosen to produce a ring with the same radius as the experimentally observed ring. Particle movement is constrained to prevent motion along the beam axis, emulating the effect of the microscope coverslip. Particle motion is modelled using a finite difference simulation including both the deterministic optical forces and random Brownian motions. The particle in the experiment was observed to occasionally stick to the slide; this is modelled with a frictional term introduced after including Brownian motion.

Funding

This research was funded by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP180101002). I.L. acknowledges support from the Australian Government RTP Scholarship.

Acknowledgment

We would like to acknowledge the computing resources provided via the UQ SMP `getafix` cluster.

Contributions

This idea was conceived by G V and developed by G V and I L. I L generated the data sets used to train the NN. G V and I L generated and trained the NNs. I L, T N, H R, and A S planned possible experiments. I L conducted the experiment. All authors contributed to the manuscript including planning figures.

Competing interests

The authors declare no competing interests.

ORCID iDs

Isaac C D Lenton https://orcid.org/0000-0002-5010-6984
Giovanni Volpe https://orcid.org/0000-0001-5057-1846
Alexander B Stilgoe https://orcid.org/0000-0002-9299-5695
Timo A Nieminen https://orcid.org/0000-0003-3055-7275
Halina Rubinsztein-Dunlop https://orcid.org/0000-0002-8332-2309

References

[1] Ashkin A, Dziedzic J M, Bjorkholm J E and Chu S 1986 Observation of a single-beam gradient force optical trap for dielectric particles *Opt. Lett.* **11** 288–90

[2] Svoboda K, Schmidt C F, Schnapp B J and Block S M 1993 Direct observation of kinesin stepping by optical trapping interferometry *Nature* **365** 721–7

[3] Bustamante C, Liphardt J and Ritort F 2005 The nonequilibrium thermodynamics of small systems *Phys. Today* **58** 43

[4] Seifert U 2012 Stochastic thermodynamics, fluctuation theorems and molecular machines *Rep. Prog. Phys.* **75** 126001

[5] Bérut A, Arakelyan A, Petrosyan A, Ciliberto S, Dillenschneider R and Lutz E 2012 Experimental verification of Landauer's principle linking information and thermodynamics *Nature* **483** 187–9

[6] Ashkin A and Dziedzic J M 1987 Optical trapping and manipulation of viruses and bacteria *Science* **235** 1517–20

[7] Zhang H and Liu K-K 2008 Optical tweezers for single cells *J. R. Soc. Interface* **5** 671–90

[8] Volpe G and Volpe G 2013 Simulation of a Brownian particle in an optical trap *Am. J. Phys.* **81** 224–30

[9] Bowman R W and Padgett M J 2013 Optical trapping and binding *Rep. Prog. Phys.* **76** 026401

[10] Gieseler J and Millen J 2018 Levitated nanoparticles for microscopic thermodynamics—a review *Entropy* **20** 326

[11] Jones P H, Maragò O M and Volpe G 2015 *Optical Tweezers: Principles and Applications* (Cambridge: Cambridge University Press)

[12] Bui A A M, Stilgoe A B, Khatibzadeh N, Nieminen T A, Berns M W and Rubinsztein-Dunlop H 2015 Escape forces and trajectories in optical tweezers and their effect on calibration *Opt. Express* **23** 24317–30

[13] Allen L, Beijersbergen M W, Spreeuw R J C and Woerdman J P 1992 Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes *Phys. Rev.* A **45** 8185–9

[14] Simpson S H and Hanna S 2009 Optical angular momentum transfer by Laguerre-Gaussian beams *J. Opt. Soc. Am.* A **26** 625–38

[15] Bui A A M, Stilgoe A B, Nieminen T A and Rubinsztein-Dunlop H 2013 Calibration of nonspherical particles in optical tweezers using only position measurement *Opt. Lett.* **38** 1244–6

[16] Phillips D B, Padgett M J, Hanna S, Ho Y-L D, Carberry D M, Miles M J and Simpson S H 2014 Shape-induced force fields in optical trapping *Nat. Photon.* **8** 400–5

[17] Rohrbach A 2005 Stiffness of optical traps: quantitative agreement between experiment and electromagnetic theory *Phys. Rev. Lett.* **95** 168102

[18] Borghese F, Denti P and Rosalba S 2007 *Scattering From Model Nonspherical Particles* (Berlin: Springer)

[19] Nieminen T A, Loke V L Y, Stilgoe A B, Knöner G, Brańczyk A M, Heckenberg N R and Rubinsztein-Dunlop H 2007 Optical tweezers computational toolbox *J. Opt. A: Pure Appl. Opt.* **9** S196–S203

[20] Nieminen T A, Du Preez-Wilkinson N, Stilgoe A B, Loke V L Y, Bui A A M and Rubinsztein-Dunlop H 2014 Optical tweezers: theory and modelling *J. Quant. Spectrosc. Radiat. Transfer* **146** 59–80

[21] Bui A A M, Stilgoe A B, Lenton I C D, Gibson L J, Kashchuk A V, Zhang S, Rubinsztein-Dunlop H and Nieminen T A 2017 Theory and practice of simulation of optical tweezers *J. Quant. Spectrosc. Radiat. Transfer* **195** 66–75

[22] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd edn: The Art of Scientific Computing* (Cambridge: Cambridge University Press)

[23] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[24] Schmidhuber J 2015 Deep learning in neural networks: An overview *Neural Netw.* **61** 85–117

[25] Hannel M D, Abdulali A, O'Brien M and Grier D G 2018 Machine-learning techniques for fast and accurate feature localization in holograms of colloidal particles *Opt. Express* **26** 15221–31

[26] Helgadottir S, Argun A and Volpe G 2019 Digital video microscopy enhanced by deep learning *Optica* **6** 506–13

[27] Rivenson Y, Göröcs Z, Günaydin H, Zhang Y, Wang H and Ozcan A 2017 Deep learning microscopy *Optica* **4** 1437–43

[28] Kiarashinejad Y, Zandehshahvar M, Abdollahramezani S, Hemmatyar O, Pourabolghasem R and Adibi A 2020 Knowledge Discovery in Nanophotonics Using Geometric Deep Learning *Adv. Intell. Syst.* **2** 1900132

[29] Kiarashinejad Y, Abdollahramezani S and Adibi A 2020 Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures *Npj Comput. Mater.* **6** 1–12

[30] Peurifoy J, Shen Y, Jing L, Yang Y, Cano-Renteria F, DeLacy B G, Joannopoulos J D, Tegmark M and Soljačić M 2018 Nanophotonic particle simulation and inverse design using artificial neural networks *Sci. Adv.* **4** eaar4206

[31] Gibson L J, Zhang S, Stilgoe A B, Nieminen T A and Rubinsztein-Dunlop H 2019 Machine learning wall effects of eccentric spheres for convenient computation *Phys. Rev.* E **99** 043304

[32] Lenton I, Stilgoe A B, Nieminen T A and Rubinsztein-Dunlop H *et al* 2020 Optical Tweezers Toolbox *GitHub* (https://github.com/ilent2/ott 10.5281/zenodo.3945818)

[33] Chollet F *et al* 2015 Keras (https://keras.io)

[34] Bae A J, Hanstorp D and Chang K 2019 Juggling with light *Phys. Rev. Lett.* **122** 043902

[35] Norregaard K, Metzler R, Ritter C M, Berg-Sørensen K and Oddershede L B 2017 Manipulation and motion of organelles and single molecules in living cells *Chem. Rev.* **117** 4342–75

[36] Padgett M and Bowman R 2011 Tweezers with a twist *Nat. Photon.* **5** 343–8

[37] Franke-Arnold S, Leach J, Padgett M J, Lembessis V E, Ellinas D, Wright A J, Girkin J M, Öhberg P and Arnold A S 2007 Optical ferris wheel for ultracold atoms *Opt. Express* **15** 8619–25

[38] Jesacher A, Fürhapter S, Bernet S and Ritsch-Marte M 2004 Size selective trapping with optical "cogwheel" tweezers *Opt. Express* **12** 4129–35

[39] Dholakia K, Reece P and Min G 2007 Optical micromanipulation *Chem. Soc. Rev.* **37** 42–55

[40] Yang D, Chang Li, Yao Z, Huang X, Yan Li, Jin P and Lin J 2020 Multiparameter controllable chiral optical patterns *Phys. Rev. Appl.* **14** 014066

[41] Rodrigo J A, Alieva T, Abramochkin E and Castro I 2013 Shaping of light beams along curves in three dimensions *Opt. Express* **21** 20544–55

[42] Lenton I C D, Armstrong D J, Stilgoe A B, Nieminen T A and Rubinsztein-Dunlop H 2020 Orientation of swimming cells with annular beam optical tweezers *Opt. Commun.* **459** 124864

[43] Kashchuk A V, Nieminen T A, Rubinsztein-Dunlop H and Stilgoe A B 2019 High-speed transverse and axial optical force measurements using amplitude filter masks *Opt. Express* **27** 10034–49

[44] Ando T, Ohtake Y, Matsumoto N, Inoue T and Fukuchi N 2009 Mode purities of Laguerre–Gaussian beams generated via complex-amplitude *Opt. Lett.* **34** 34–6