Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **439,** 2102–2121 (2014) doi:10.1093/mnras/stu090
Advance Access publication 2014 February 10

PkANN – II. A non-linear matter power spectrum interpolator developed using artificial neural networks

Shankar Agarwal,[1]★ Filipe B. Abdalla,[2]★ Hume A. Feldman,[3]★ Ofer Lahav[2]★ and Shaun A. Thomas[2]★

[1] *CNRS, Laboratoire Univers et Théories (LUTh), UMR 8102 CNRS, Observatoire de Paris, Université Paris Diderot, 5 Place Jules Janssen, F-92190 Meudon, France*
[2] *Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK*
[3] *Department of Physics and Astronomy, University of Kansas, Lawrence, KS 66045, USA*

Accepted 2014 January 14. Received 2014 January 14; in original form 2013 April 19

ABSTRACT

In this paper we introduce PkANN, a freely available software package for interpolating the non-linear matter power spectrum, constructed using artificial neural networks (ANNs). Previously, using HALOFIT to calculate matter power spectrum, we demonstrated that ANNs can make extremely quick and accurate predictions of the power spectrum. Now, using a suite of 6380 *N*-body simulations spanning 580 cosmologies, we train ANNs to predict the power spectrum over the cosmological parameter space spanning 3σ confidence level around the concordance cosmology. When presented with a set of cosmological parameters ($\Omega_{\rm m}h^2$, $\Omega_{\rm b}h^2$, $n_{\rm s}$, w, σ_8, $\sum m_\nu$ and redshift z), the trained ANN interpolates the power spectrum for $z \le 2$ at sub-per cent accuracy for modes up to $k \le 0.9\,h\,{\rm Mpc}^{-1}$. PkANN is faster than computationally expensive *N*-body simulations, yet provides a worst-case error <1 per cent fit to the non-linear matter power spectrum deduced through *N*-body simulations. The overall precision of PkANN is set by the accuracy of our *N*-body simulations, at 5 per cent level for cosmological models with $\sum m_\nu < 0.5$ eV for all redshifts $z \le 2$. For models with $\sum m_\nu > 0.5$ eV, predictions are expected to be at 5 (10) per cent level for redshifts $z > 1$ ($z \le 1$). The PkANN interpolator may be freely downloaded from http://zuserver2.star.ucl.ac.uk/~fba/PkANN.

Key words: cosmological parameters – cosmology: theory – large-scale structure of Universe.

1 INTRODUCTION

With the upcoming surveys promising to breach the per cent level of precision, any efforts to further improve the constraints on cosmological parameters will be predominantly theory limited. The Baryon Oscillation Spectroscopic Survey (BOSS; Eisenstein et al. 2011) aims to determine the angular diameter distance with a precision of 1 per cent at redshifts $z = 0.3$ and 0.55, and the cosmic expansion rate $H(z)$ with 1–2 per cent precision at the same redshifts. The Dark Energy Survey (DES; The Dark Energy Survey Collaboration 2005) will probe the nature of dark energy through both the growth of structure in the Universe as a function of time and the dependence of distances on the expansion rate. The Dark Energy Spectroscopic Instrument (DESI; Levi et al. 2013), through redshift measurements of millions of galaxies and quasars, will enable baryon acoustic oscillation (BAO) and redshift space distortion measurements. The Large Synoptic Survey Telescope (LSST; Ivezic et al. 2008) will measure the comoving distance in the redshift range $z = 0.3$–3.0 with an accuracy of 1–2 per cent. These studies will shed more light and possibly solve some of the unanswered questions in cosmology including the nature of dark energy, and the absolute mass scale, the hierarchy and the effective number of neutrino species $N_{\rm eff}$. Using BAO and the cosmic microwave background (CMB) data, *Planck* (Planck Collaboration et al. 2013) constrains the dark energy constant equation of state parameter at $w = -1.13 \pm 0.13$ with no evidence for dynamical dark energy. This is consistent with a cosmological constant ($w = -1$) dominated flat universe. In order to distinguish between various models of dark energy, such as $w \neq -1$ and/or a time-varying equation of state parameter, one needs more precise and accurate measurements of the matter power spectrum.

Neutrino oscillation experiments (SNO 2004; Adamson et al. 2008; KamLAND 2008) indicate that at least two neutrino eigentstates have non-zero masses. Massive neutrinos thus qualify

★ E-mail: shankar.agarwal@obspm.fr (SA); fba@star.ucl.ac.uk (FBA); feldman@ku.edu (HAF); lahav@star.ucl.ac.uk (OL); sat@star.ucl.ac.uk (SAT) as a hot dark matter component and contribute to the total energy density of the Universe. Free-streaming of massive neutrinos damps small-scale density perturbations, thereby suppressing the growth of cosmological structure. Accurate measurements of the matter power spectrum offer a powerful tool to constrain the absolute mass-scale of neutrinos, and complement the oscillation experiments which, being sensitive to the mass squared differences between the neutrino eigentstates, only provide a lower bound on the total neutrino mass. Specifically, mass splittings of $|\Delta m^2_{32}| = (2.43 \pm 0.13) \times 10^{-3}\,\mathrm{eV}^2$ and $\Delta m^2_{21} = (7.59 \pm 0.21) \times 10^{-5}\,\mathrm{eV}^2$ (Adamson et al. 2008; KamLAND 2008) imply a lower limit for the sum of the neutrino masses to be 0.06 and 0.1 eV for the normal and inverted mass hierarchies (Otten & Weinheimer 2008), respectively. Assuming a minimal-mass ($\sum m_\nu = 0.06$ eV) normal hierarchy for the neutrino masses, the *Planck* survey find $\sum m_\nu < 0.23$ eV (95 per cent CL). *Wilkinson Microwave Anisotropy Probe* (*WMAP*) 9-year (Hinshaw et al. 2013) analysis find $\sum m_\nu < 0.44$ eV (95 per cent CL). Lahav et al. (2010) obtained an upper limit of 0.11 eV (95 per cent CL). Numerical studies of the scale-dependent suppression of matter power spectrum has been performed by various groups: Brandbyge & Hannestad (2010), Viel, Haehnelt & Springel (2010), Agarwal & Feldman (2011), Bird, Viel & Haehnelt (2012) and Wagner, Verde & Jimenez (2012). Agarwal & Feldman (2011, hereafter Paper I) and Wagner et al. (2012) show that resolving the neutrino mass hierarchy may require the power spectrum to be measured at better than 0.5 per cent accuracy, which may be possible with the next generation of experiments.

Currently, there are four popular approaches to estimate the non-linear matter power spectrum: (i) HALOFIT (Smith et al. 2003); (ii) higher order perturbation theory (PT; e.g. Saito, Takada & Taruya 2008, 2009; Nishimichi et al. 2009; Upadhye et al. 2013); (iii) N-body simulations (e.g. ENZO, O'Shea et al. 2010; GADGET, Springel 2005); (iv) spectrum interpolators (e.g. Heitmann et al. 2006, 2014; Habib et al. 2007; Lawrence et al. 2010). While HALOFIT performs well on large scales ($k \lesssim 0.1\,h\,\mathrm{Mpc}^{-1}$), its performance degrades rapidly on smaller scales. Takahashi et al. (2012) recalibrated the original HALOFIT (Smith et al. 2003) extending it to include dark energy models with constant equation of state $w \neq -1$. The accuracy of HALOFIT predictions is model dependent and may be as low as 5–10 per cent at $k \sim 1\,h\,\mathrm{Mpc}^{-1}$ (Takahashi et al. 2012; Heitmann et al. 2014). Likewise, PT improves upon linear theory predictions on large scales but fails on smaller ($k \gtrsim 0.09\,h\,\mathrm{Mpc}^{-1}$) scales. At higher redshifts when the perturbations are less evolved, the accuracy for both HALOFIT and PT improves. However, since dark energy is a late-time phenomenon ($z \lesssim 2$), one cannot rely on fitting functions like HALOFIT and PT at low redshifts if one aims to develop a theoretical framework capable of predicting the non-linear matter power spectrum at per cent level. This leaves N-body simulations as the only method capable of controlling the accuracy levels as desired. Heitmann et al. (2010) show that gravity-only simulations can be used to calculate the matter power spectrum at sub-per cent accuracy up to $k \lesssim 1\,h\,\mathrm{Mpc}^{-1}$. On smaller scales, baryonic physics affects the power spectrum and needs to be included in numerical simulations to maintain per cent accuracy.

A typical high-resolution dark-matter only simulation intended to probe $k \lesssim 1\,h\,\mathrm{Mpc}^{-1}$ scales can cost ~10 000 CPU hours. Including hydrodynamics in simulations to probe smaller scales can take prohibitively long, especially when running multiple simulations spread across the cosmological parameter space. As discussed earlier in Heitmann et al. (2006) and Habib et al. (2007), parameter estimation and model building typically involves sampling the parameter space and evaluating the power spectrum for each cosmology. As we mentioned in Agarwal et al. (2012, hereafter Paper II), given the multidimensionality of the cosmological parameter space, a brute force application of N-body simulations is beyond our current state of the art computing capabilities.

A novel alternative to running numerical simulations to determine the non-linear response from varying parameter settings is to use machine learning techniques. Machine learning has found use in a variety of applications such as brain–machine interfaces (Jenatton et al. 2011; Pedregosa et al. 2012), analyses of stock market (Ghosh 2011; Hurwitz & Marwala 2012), fitting of cosmological functions (Auld et al. 2007; Fendt & Wandelt 2007; Auld, Bridges & Hobson 2008) and estimating photometric redshifts (Collister & Lahav 2004).

Using machine learning in the form of Gaussian processes, Heitmann et al. (2009, 2014) and Lawrence et al. (2010) have developed a matter power spectrum calculator – COSMIC EMULATOR, that is an order of magnitude improvement over the popular HALOFIT prescription. The COSMIC EMULATOR, based on gravity-only N-body simulations, comes in two versions: h-fixed (Lawrence et al. 2010) and h-free (Heitmann et al. 2014). The h-fixed version computes the Hubble parameter h using the CMB constraint on the acoustic scale and predicts the non-linear matter power spectrum up to $z \leq 1$ for modes $k \lesssim 1\,h\,\mathrm{Mpc}^{-1}$. The h-free version has h as a free parameter that can be set by the user. The range of validity of the h-free version is up to $z \leq 4$ for modes $k \lesssim 15\,h\,\mathrm{Mpc}^{-1}$. Both versions are restricted to cosmological models with massless neutrinos. Since the current understanding is that at least two neutrino eigentstates have non-zero masses, it is reasonable to develop a power spectrum interpolator that is suitable for cosmological models with/without massive neutrinos.

In Paper II, we developed the formalism for estimating the non-linear matter power spectrum using artificial neural networks (ANNs). Using HALOFIT spectra as mock N-body spectra, we showed that the ANN formalism enables a remarkable fit with a manageable number of simulations. In this paper, we use a suite of 6380 N-body simulations spanning 580 cosmologies around the *WMAP* 7-year central values, and train ANNs to predict the power spectrum accurate at 5–10 per cent level for $k \leq 0.9\,h\,\mathrm{Mpc}^{-1}$ up to redshifts $z \leq 2$. The PkANN package, along with instructions to use, is available at http://zuserver2.star.ucl.ac.uk/~fba/PkANN.

We trained PkANN for a range of cosmologies including $w \neq -1$ and $m_\nu \neq 0$. However, the training can be easily extended to include other parameters such as time-varying dark energy, modified gravity as well as probing small-scale baryonic effects. This will require (i) running a few N-body simulations around the cosmological parameter(s) being probed, (ii) calculating the matter power spectra from numerical simulations, (iii) randomly dividing these power spectra into two sets, namely, the training and validation sets (explained in Paper II, and here in Appendix A1) and (iv) training PkANN using the training and validation sets. Once training is over, the trained network can be used to predict the matter power spectrum at new parameter settings.

The outline of this paper is as follows. We discuss our numerical simulations in Section 2. We develop the PkANN interpolator in Section 3. We present our results in Section 4 starting with the performance of the PkANN interpolator against spectra computed using N-body simulations (Section 4.1). The estimate of errors in PkANN's predictions is summarized in Section 4.2. In Section 4.3, we use PkANN to study the response of matter power spectrum to variations in cosmological parameters. PkANN's performance is compared with the h-fixed COSMIC EMULATOR as well. We conclude in Section 5. In Appendix A, we detail the formulae used in developing PkANN.

Table 1. Parameter space used in generating the ANN training and validation sets. The last column shows the corresponding *WMAP* 7-year+BAO+H_0 constraints at 68 per cent CL. Inside parentheses is the range for the ANN testing set. The range of the parameters for the testing set is designed to avoid the boundaries of the parameter space. Neutrino mass being physically bound ($\sum m_\nu \gtrsim 0$), the lower bound on neutrino mass is set at zero.

Cosmological parameters	Lower value	Upper value	*WMAP* 7-year+BAO+H_0^a
$\Omega_m h^2$	0.110 (0.120)	0.165 (0.150)	0.1352 ± 0.0036
$\Omega_b h^2$	0.021 (0.022)	0.024 (0.023)	0.02255 ± 0.00054
n_s	0.85 (0.90)	1.05 (1.00)	0.968 ± 0.012
w	−1.35 (−1.15)	−0.65 (−0.85)	−1.1 ± 0.14
σ_8	0.60 (0.70)	0.95 (0.85)	0.816 ± 0.024
$\sum m_\nu$ (eV)	0 (0)	1.1 (0.5)	<0.58[b]

[a] Komatsu et al. (2011).
[b] 95 per cent CL for $w = -1$.

2 NUMERICAL SIMULATIONS

We run *N*-body simulations over a range of cosmological parameters with the publicly available adaptive mesh refinement (AMR), grid-based hybrid (hydro+gravity) code ENZO[1] (Norman et al. 2007; O'Shea et al. 2010). All our simulations are hydro+gravity and run in unigrid (AMR switched off) mode. For the hydrodynamical simulations, we include radiative cooling of baryons using an analytical approximation (Sarazin & White 1987) for a fully ionized gas with a metallicity of 0.5 $M_\odot$. The cooling approximation is valid over the temperature range from 10^4 to 10^9 K. Below 10^4 K, the cooling rate is effectively zero. We do not account for metal-line cooling, supernova (SN) feedback or active galactic nucleus (AGN) feedback. The parameters we consider are $\boldsymbol{I} \equiv (\Omega_m h^2,\ \Omega_b h^2,\ n_s,\ w,\ \sigma_8,\ \sum m_\nu)$, where h, Ω_m, Ω_b, n_s, w, σ_8 and $\sum m_\nu$ are the present-day normalized Hubble parameter in units of 100 km s^{-1} Mpc^{-1}, the present-day matter and baryonic normalized energy densities, the primordial spectral index, the constant equation of state parameter for dark energy, the amplitude of fluctuation on an 8 h^{-1} Mpc scale and the total neutrino mass, respectively. The limits (see Table 1) on this six-dimensional parameter space includes the *WMAP* 7-year+BAO+H_0 (Komatsu et al. 2011) constraints.

For details on generating the initial conditions for simulations, and treating massive neutrinos, refer Paper I. Our *N*-body simulations do not explicitly account for the presence of neutrino perturbations and implement neutrinos only through its effects on the background evolution. Specifically, we modified the cosmological routines of the ENZO code to include the effects of massive neutrinos on the homogeneous Hubble expansion $h(a)$ (for details, see Paper II) and the linear growth factor. Our modifications to the growth factor neglect any scale dependence in the presence of massive neutrinos. We sample 70 ($\sum m_\nu = 0$) + 130 ($\sum m_\nu \neq 0$) = 200 (training set), 18 + 32 = 50 (validation set) and 150 + 180 = 330 (testing set) cosmologies from the parameter space (see Table 1) using an improved Latin hypercube technique (for details, see Paper II). The training set guides the neural network training, the validation set prevents the ANN from overfitting to the training set and the testing set is used to evaluate the performance of the trained network. The testing set has no effect on training and provides an independent measure of network performance. For each cosmology $\boldsymbol{I} \equiv (\Omega_m h^2,\ \Omega_b h^2,\ n_s,\ w,\ \sigma_8,\ \sum m_\nu)$, we compute the Hubble parameter h using the *WMAP* 7-year+BAO constraint on the acoustic scale $\pi d_{ls}/r_s = 302.54$, where d_{ls} is the distance to the surface of last scattering and r_s is the comoving size of the sound horizon at the redshift of last scattering. The procedure to compute h is outlined in Paper II. This h value, together with the chosen $\Omega_m h^2$ and $\Omega_b h^2$, is used to derive Ω_m and Ω_b. The present-day normalized energy density of dark energy is fixed as $\Omega_{de} = 1 - \Omega_m$. Starting at redshift $z = 99$, all simulations are run in a comoving box of length 200 h^{-1} Mpc, with 256^3 cold dark matter (CDM) particles evolved on a 512^3 grid. We take 111 snapshots of the CDM and baryon positions between redshifts $z = 2$ and 0; specifically 100 snapshots ($\Delta z = 0.01$ apart) in the range $0 \leq z \leq 0.99$, and 11 snapshots ($\Delta z = 0.1$ apart) in the range $1 \leq z \leq 2$.

Using a cloud-in-cell (CIC) interpolation scheme, we transform the CDM and baryon positions into their respective mass density fields. The densities are fast Fourier transformed to obtain the CDM and baryon non-linear power spectra, namely P_{nl}^{c} and P_{nl}^{b}, respectively. Together with the neutrino linear spectrum P_{lin}^{ν}, and the weights $f^i \equiv \Omega_i/\Omega_m$, the non-linear matter power spectrum P_{nl} is then calculated (for details, see Paper I) as

$$P_{nl}(k) = \left[(f^c + f^b)\sqrt{P_{nl}^{cb}(k)} + f^\nu\sqrt{P_{lin}^{\nu}(k)}\right]^2, \qquad (1)$$

where

$$P_{nl}^{cb}(k) = (f^c + f^b)^{-2}\left[f^c\sqrt{P_{nl}^{c}(k)} + f^b\sqrt{P_{nl}^{b}(k)}\right]^2. \qquad (2)$$

The subscripts 'lin' and 'nl' indicate quantities in the linear and non-linear regimes, respectively. Throughout our analyses, we work with flat cosmological models: $\Omega_m (= \Omega_b + \Omega_c + \Omega_\nu) + \Omega_{de} = 1$, where Ω_c and Ω_ν are the present-day normalized energy densities of CDM and neutrino, respectively. To suppress statistical scatter in the matter power spectrum, we average the power spectra for 11 realizations per cosmology. In Fig. 1, we show P_{nl}^{c}, P_{nl}^{b} and P_{nl} spectra (long-dashed, short-dashed and solid lines, respectively) for one of the cosmological models $\boldsymbol{I} \equiv (0.1196, 0.0232, 0.992, -0.72, 0.8587, 0)$ with $h = 0.6496$. The linear matter power spectrum is shown by dot–dashed line. At $k = 1\,h$ Mpc^{-1}, baryons suppress the CDM spectrum at 1–2 per cent level. At low redshifts ($z \lesssim 2$), as the gas component cools and condenses, it collapses to the centre of CDM haloes, thereby enhancing the gas power spectrum above the CDM spectrum on smaller scales ($k \gtrsim 10\,h$ Mpc^{-1}). This is consistent with previous studies (Rudd, Zentner & Kravtsov 2008; Casarini et al. 2011) that investigated the effect of baryonic physics on the matter power spectrum through simulations including gas cooling, star formation and SN feedback. We note that although all our simulations in this work are hydro+gravity, on large scales

[1] http://lca.ucsd.edu/projects/enzo

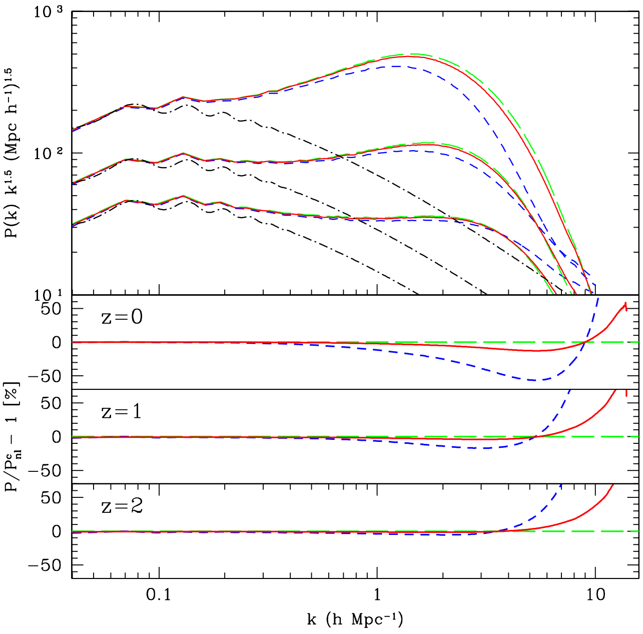

Figure 1. Top panel: matter power spectrum evaluated at redshifts $z = 0$, 1, 2 (top to bottom sets, respectively) for the cosmological model $\boldsymbol{I} \equiv (0.1196, 0.0232, 0.992, -0.72, 0.8587, 0)$ with $h = 0.6496$. At each redshift, the various lines are the non-linear spectra computed using hydro+gravity simulations: (i) $P^{\rm c}_{\rm nl}$ (long-dashed), (ii) $P^{\rm b}_{\rm nl}$ (short-dashed) and (iii) $P_{\rm nl}$ (solid). The linear matter power spectrum is shown by dot–dashed line. $P_{\rm nl}$ is constructed using $P^{\rm c}_{\rm nl}$ and $P^{\rm b}_{\rm nl}$, as discussed in the text (see equations 1 and 2). Lower panels: the ratio of the non-linear spectra ($P^{\rm c}_{\rm nl}$, $P^{\rm b}_{\rm nl}$ and $P_{\rm nl}$) to the CDM spectrum $P^{\rm c}_{\rm nl}$.

($k \lesssim 1\,h\,{\rm Mpc}^{-1}$) the matter power spectrum is minimally affected by baryonic dynamics and one can rely on gravity-only simulations.

We use the one-loop standard PT as implemented by Saito et al. (2008) for estimating the matter power spectrum up to $k \le 0.085\,h\,{\rm Mpc}^{-1}$ and stitch it with the non-linear power spectrum from numerical simulations. Finally, the stitched spectrum is sampled at 50 k-values in the range $0.006 \le k \le 1\,h\,{\rm Mpc}^{-1}$. The stitched-and-sampled non-linear power spectrum is used as $P_{\rm nl}(k, z)$ for ANN training. This stitch-and-sample procedure is repeated for each cosmology $\boldsymbol{I}$ in the training set to complete the training set $P_{\rm nl}(k, z|\boldsymbol{I})$.

3 ARTIFICIAL NEURAL NETWORKS

Fig. 2 shows a skeleton of a machine learning network. Using a suitable training set (input parameters for which data is available), the machine learning algorithm is trained to learn a parametrization. With this parametrization the network is capable of reproducing (as closely as possible) the output, when queried with input parameter settings that are part of the training set. The trained network can now be presented with new settings of the input parameters (for which one does not have any prior data) and by using the same parametrization learnt during the training process, the network makes predictions.

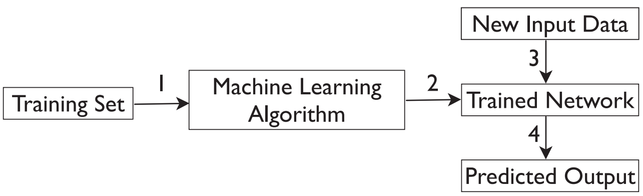

Figure 2. Steps 1 and 2: a machine learning network learns to parametrize the output, for the input patterns that form the training set. Steps 3 and 4: the trained network is capable of making predictions when presented with input parameter settings. The queried input settings must lie within the parameter ranges of the patterns in the training set.

ANN – a form of machine learning – is a collection of *nodes* arranged in a series of layers, with each node in a layer connected to all other nodes in adjacent layers. A network's architecture is specified by the number of nodes from input to output as $N_{\rm in} : N_1 : N_2 : \ldots : N_n : N_{\rm out}$. That is a network with an architecture $4 : 9 : 5 : 7$ has 4 inputs, two hidden layers with 9 and 5 nodes, respectively, and finally 7 outputs. An extra node (called the *bias* node) is added to the input layer as well as to each of the hidden layers. The bias nodes are added in order to compensate for the difference between the network's mean prediction and the mean of the outputs of training set patterns (for details, refer Bishop 1995). Each bias node connects to all the nodes in the next layer. Note that the counts $N_{\rm in}, N_1, N_2, \ldots, N_n$ do not include the bias nodes. The output layer has no bias node. The total number of connections (also called the *weights*) N_W for a generic architecture $N_{\rm in} : N_1 : N_2 : \ldots : N_n : N_{\rm out}$ can be calculated using the formula

$$N_W = N_{\rm in}N_1 + \sum_{l=2}^{n} N_{l-1}N_l + N_n N_{\rm out} + \sum_{l=1}^{n} N_l + N_{\rm out}, \qquad (3)$$

where the summation index l is over the hidden layers only. Throughout this paper, we will use the vector notation $\boldsymbol{w}$ to collectively refer to all the network weights.

In Fig. 3, we show a typical ANN architecture (left-hand panel) and the formulae to calculate the node activations (right-hand panels). In the network configuration depicted, there are $N_{\rm in}$ input parameters/features $(x_1, \ldots, x_i)$, a single hidden layer with N_1 nodes $(z_1, \ldots, z_j)$ and $N_{\rm out}$ output parameters/features $(y_1, \ldots, y_k)$. The bias nodes in the input and hidden layers are x_0 and z_0, respectively.

Each node in the lth hidden layer is a neuron with an *activation*, $z_j \equiv g(a_j)$, taking as its argument

$$a_j = \sum_{i=0} w_{ji} z_i, \qquad (4)$$

where the sum is over all nodes i (including the bias node) of the previous layer sending connections to the jth node (barring the bias

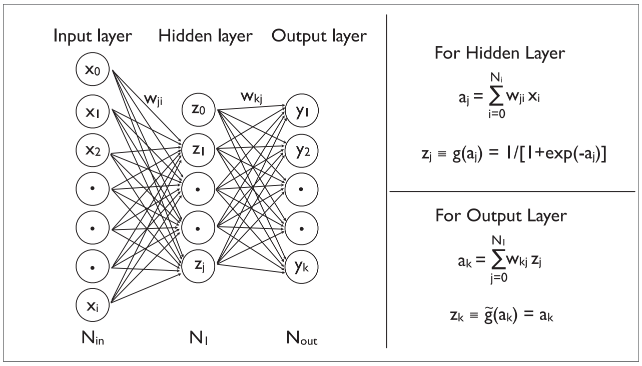
Input layer
Hidden layer
Output layer
x0
x1
x2
xi
wji
z0
z1
zj
wkj
y1
y2
yk
Nin
N1
Nout
For Hidden Layer
$a_j = \sum_{i=0}^{N_i} w_{ji} x_i$
$z_j \equiv g(a_j) = 1/[1+\exp(-a_j)]$
For Output Layer
$a_k = \sum_{j=0}^{N_1} w_{kj} z_j$
$z_k \equiv \tilde{g}(a_k) = a_k$

Figure 3. A typical ANN architecture (left-hand panel) with node activation formulae for the hidden and output layers (right-hand panels). There can be more than one hidden layers. Throughout our PkANN analysis, we work with a single hidden layer.

node) of the current layer. Note that for networks with a single hidden layer (as in Fig. 3), z_i in equation (4) would correspond to the input parameters x_i. The activation functions are typically taken to be sigmoid functions such as $g(a_j) = 1/[1 + \exp(-a_j)]$. Since the range of $g(a_j)$ is from 0 to 1, it allows the output of the neurons to be interpreted as the probability that any specific neuron will 'fire' when presented with an input parameters setting. The sigmoid functions impart some degree of non-linearity to the neural network models. A network becomes overly non-linear if the weights $\boldsymbol{w}$ deviate significantly from zero. This drives the activation $g(a_j)$ of the nodes to saturation. The number and size of the hidden layers add to the complexity of ANNs. The activation of all bias nodes is permanently set to a value of 1 and during network training the bias parameters (namely, w_{j0} and w_{k0} in Fig. 3 left-hand panel) are adjusted so as to minimize the difference between the mean prediction for the network and the mean of the outputs of the training set patterns.

The activation $y_k \equiv \tilde{g}(a_k)$ for neurons in the output layer is usually taken to be a_k, i.e. $\tilde{g}(a_k) = a_k$, with a_k being the weighted sum of all nodes in the final hidden layer,

$$a_k = \sum_{j=0} w_{kj} z_j . \tag{5}$$

For a particular input vector $(x_1, \ldots, x_i)$, the output vector $(y_1, \ldots, y_k)$ of the network is determined by progressing sequentially through the network layers, from inputs to outputs, calculating the activation of each node.

Adjusting the weights $\boldsymbol{w}$ to get the desired mapping is called the *training* of the network. For matter power spectrum estimation, we use a training set of N-body simulations with known cosmological parameters:

$$\boldsymbol{I} \equiv \left(\Omega_\mathrm{m}h^2,\ \Omega_\mathrm{b}h^2,\ n_\mathrm{s},\ w,\ \sigma_8,\ \sum m_\nu\right).$$

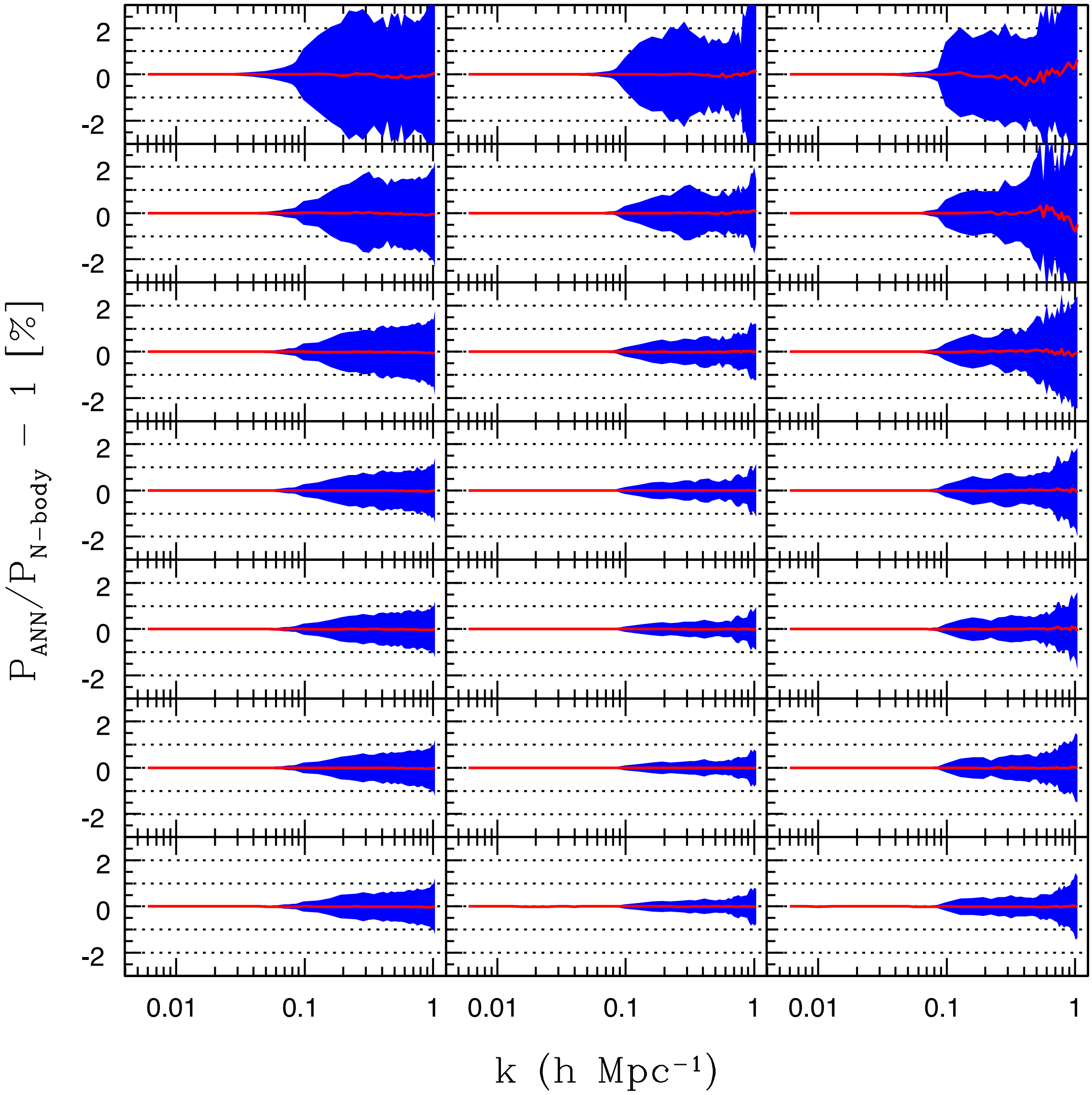

Figure 4. Percentage error at redshift $z = 0$ (left-hand panel), $z = 1$ (middle panel) and $z = 2$ (right-hand panel) between the predicted non-linear power spectrum (using PkANN) and the true underlying spectrum (using N-body simulations) for 200 training set cosmologies. The shaded region contains the middle 99.73 per cent (3σ) of the residuals. The rows (from top to bottom) correspond to N_hidden = 14–98 in increments of 14. The mean error over all 200 cosmologies is shown by a solid line – an indicator of any bias in the ANN training scheme.

It has been shown (see Hornik 1991; Ito 1991; Bishop 1995) that networks with a single hidden layer are capable of making arbitrarily accurate approximation to a function and its derivatives. As such, for PkANN's architecture, we only consider networks having single-hidden layer with sigmoidal activations and output nodes with linear ($\tilde{g}(a_k) = a_k$) activations, as depicted in Fig. 3.

In Appendix A1, we develop the PkANN cost function $\chi^2_{\rm C}(\boldsymbol{w})$. Minimizing this cost function with respect to the weights $\boldsymbol{w}$ generates a trained ANN that can be used for non-linear matter power spectrum interpolation. To minimize $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) with respect to the weights $\boldsymbol{w}$, we use an iterative quasi-Newton algorithm (Appendix A2) that involves evaluating the first-order derivative (gradient) of the cost function. See Appendix A3 for the derivation of the gradient. The quasi-Newton algorithm also involves information about the inverse of the Hessian (second-order derivative) matrix which we approximate using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (see Appendix A4; for details, see Bishop 1995).

Starting with randomly assigned weights $\boldsymbol{w}$, their values are re-estimated iteratively, making sure that each iteration proceeds in a direction that lowers the cost function $\chi^2_{\rm C}(\boldsymbol{w})$. In order to avoid overfitting to the training set, after each iteration to the weights, equation (A11) is also calculated for what is known in neural network parlance as a validation set. The validation set for our application of neural networks is a small set of simulations with known $\boldsymbol{I} \equiv (\Omega_{\rm m}h^2,\ \Omega_{\rm b}h^2,\ n_{\rm s},\ w,\ \sigma_8,\ \sum m_\nu)$ and $P_{\rm nl}(k, z)$. The final weights $\boldsymbol{w}_{\rm f}$ are chosen so as to give the best fit (minimum $\chi^2_{\rm C}(\boldsymbol{w})$) to the validation set. The network training is considered finished once $\chi^2_{\rm C}(\boldsymbol{w})$ is minimized with respect to the validation set. The trained network can now be used to predict $P_{\rm nl}(k, z)$ for new cosmologies. It is important to note that starting with a different (but still random) configuration of weights, may lead to a trained network with a different set of final weights $\boldsymbol{w}_{\rm f}$. As such, we train a number of networks that start with an alternative random configuration of weights. The trained networks are collectively called a *committee* of networks and subsequently give rise to better performance than any single ANN in isolation. For the final output, we average over the outputs of the committee members.

4 RESULTS

4.1 Comparing PkANN against numerical simulations

In Paper II, we compared PkANN's performance against HALOFIT spectra to demonstrate that a suitably trained network is capable of reproducing the HALOFIT spectra at sub-per cent accuracy. Here, we repeat the procedure, this time using spectra calculated using N-body simulations. We selected the combination $7 : N_{\rm hidden} : 50$ as our PkANN architecture, where $N_{\rm hidden}$ (number of nodes in the hidden layer) was varied from 7 to 98, in steps of 7. The number of inputs were fixed at 7, corresponding to $\boldsymbol{I} \equiv (\Omega_{\rm m}h^2,\ \Omega_{\rm b}h^2,\ n_{\rm s},\ w,\ \sigma_8,\ \sum m_\nu)$ including redshift z. We use the CAMB (Lewis, Challinor & Lasenby 2000) code to calculate the CDM, baryon and neutrino transfer functions. The initial conditions for CDM particles and baryons are then generated from their transfer functions using ENZO. The non-linear matter power spectrum $P_{\rm nl}(k)$ is constructed using equations (1) and (2).

As in Paper II, we do not sample the redshift in the Latin hypercube but instead evaluate $P_{\rm nl}(k, z)$ at 111 redshifts between $z = 0$ and 2 from numerical simulations, using equations (1) and (2). As we discussed in Section 2, we extend the range of our spectra to $k = 0.006\,h\,{\rm Mpc}^{-1}$ by using the one-loop standard PT (Saito et al. 2008). We estimate the matter power spectrum up to $k \le 0.085\,h\,{\rm Mpc}^{-1}$ using the one-loop standard PT and stitch it with $P_{\rm nl}(k, z)$. The stitched spectrum is then sampled at 50 k-modes in the range $0.006 \le k \le 1\,h\,{\rm Mpc}^{-1}$. Since our training and validation sets have (130 + 70) and (32 + 18) cosmologies, respectively (see Paper II), we calculated $P_{\rm nl}(k, z)$ for each cosmology, at 111 redshifts. These $P_{\rm nl}(k, z)$ are scaled by their respective linear spectra $P_{\rm lin}(k, z)$ (see equation A9), before being fed to the neural network. Thus, the overall size $N_{\rm T}$ of the training set that we train our ANN with is $N_{\rm T} = 200 \times 111 = 22\,200$. Likewise, we have $50 \times 111 = 5550$ patterns in the validation set. For each $N_{\rm hidden}$ setting, we trained a committee of 16 ANNs. The weights $\boldsymbol{w}$ for each ANN were randomly initialized (the random configuration being different for each ANN). The weights are allowed to evolve until $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) is minimized with respect to the cosmologies in the validation set.

In Fig. 4, we show the percentage error in the ANN predictions with respect to the N-body results when presented with the 200 cosmologies in the training set. We average the $P^{\rm ANN}_{\rm nl}(k, z)$ predictions over the 16 ANN committee members. The rows correspond to $N_{\rm hidden} = 14$–98 (from top to bottom) in increments of 14. The columns (from left to right) correspond to $z = 0$, 1, 2. The mean error over all 200 cosmologies in the training set is shown by a solid line in each panel, to get an idea about any systematics in our ANN training scheme. With $N_{\rm hidden} = 70$ and higher, the ANN predictions are within ±1 per cent of the N-body power spectra for $k \le 0.9\,h\,{\rm Mpc}^{-1}$, after which the performance degrades marginally to ±1.5 per cent. The worst-performing cosmologies correspond to the parameter settings with at least four of the six cosmological parameters at their boundary values.

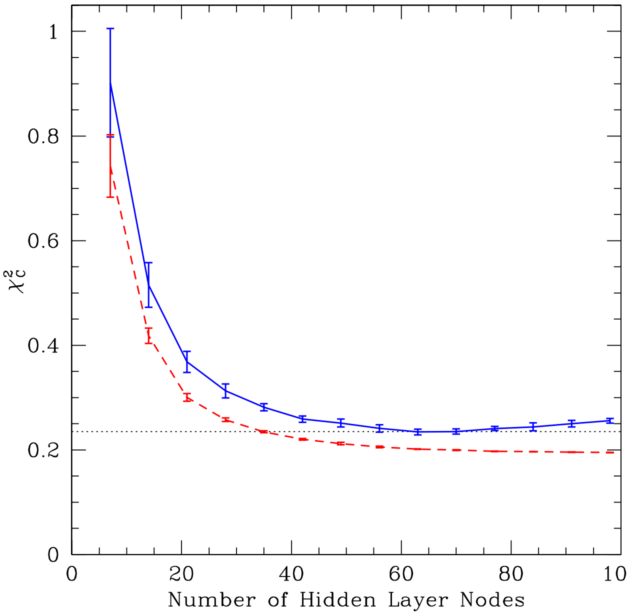

Figure 5. The residual error $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) evaluated as a function of the number of nodes in the hidden layer, $N_{\rm hidden}$. The error is a monotonically decreasing function for the training set (dashed line) while for the validation set (solid line), it starts increasing beyond $N_{\rm hidden} = 70$ indicating that the generalizing ability of the neural network is best with $N_{\rm hidden} = 70$. The error bars correspond to the spread in $\chi^2_{\rm C}(\boldsymbol{w})$ for the 16 ANN committee members.

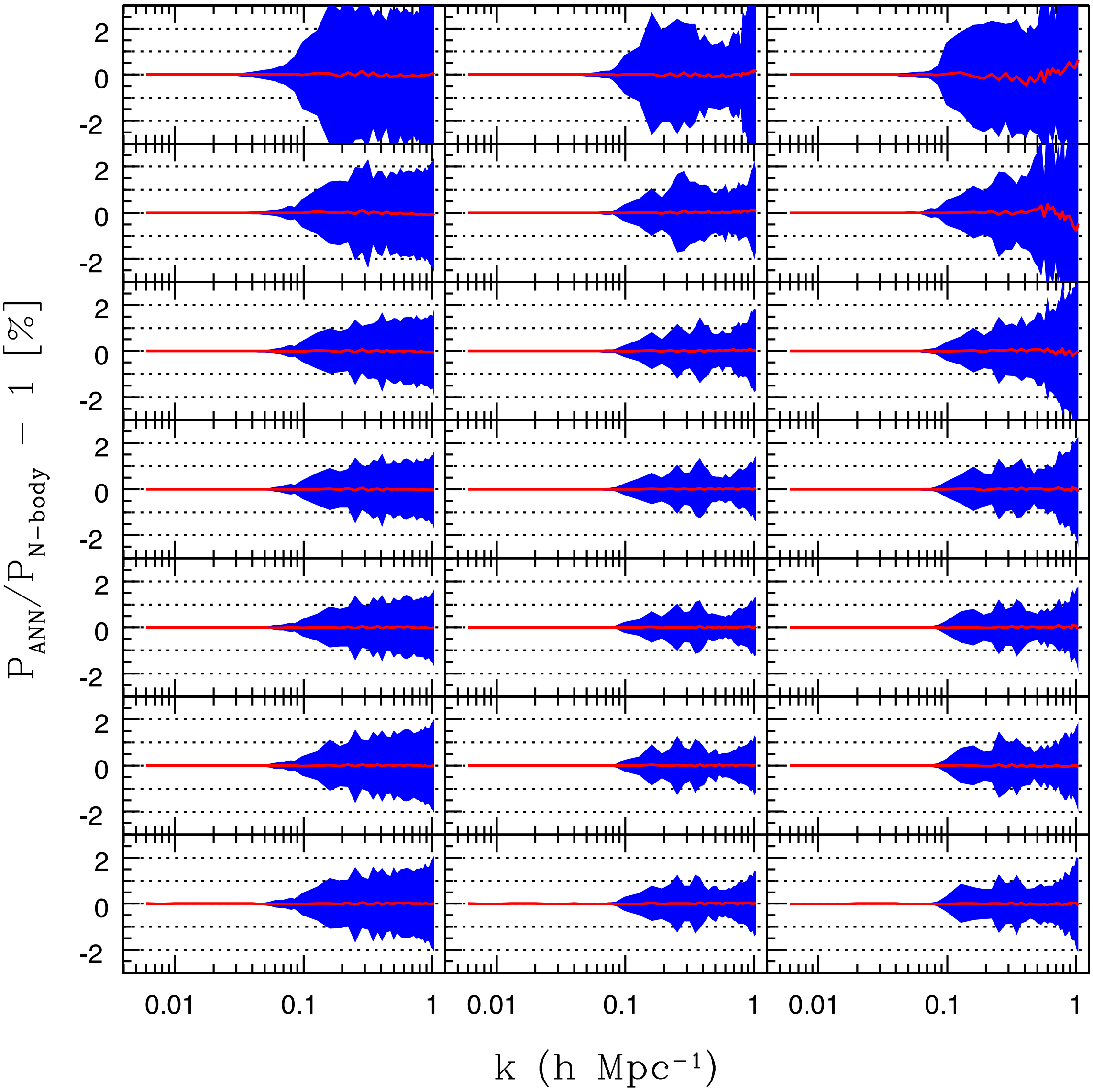

Figure 6. Similar to Fig. 4, using 50 validation set cosmologies.

Increasing the number of nodes in the hidden layer increases the flexibility of a neural network. An increasingly complex network can make extremely accurate predictions on the training set. This is evident from Fig. 4, where the prediction over the training set becomes progressively better (from top to bottom) with increasing $N_{\rm hidden}$ units. However, such complex networks can adversely affect their generalizing ability when presented with a new data set. The validation set helps in controlling the complexity of a network, as we discussed earlier in Section 3. In Fig. 5, we show the residual cost function $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) evaluated as a function of the number of nodes in the hidden layer, $N_{\rm hidden}$. The residual error is a monotonically decreasing function for the training set (dashed line) while for the validation set (solid line), it increases beyond $N_{\rm hidden} = 70$. The performance of the trained ANNs as a function of $N_{\rm hidden}$ units, over the cosmologies in the validation set, is shown in Fig. 6. Increasing $N_{\rm hidden}$ beyond 70 increases the error marginally, indicating that $N_{\rm hidden} = 70$ saturates the generalizing ability of our network.

The performance of the trained ANNs for cosmological models in the testing set is shown in Fig. 7. Increasing $N_{\rm hidden}$ beyond 70 does not contribute to a significant error reduction on the testing set, confirming our assessment that $N_{\rm hidden} = 70$ saturates the generalizing ability of the network. With $N_{\rm hidden} = 70$, the ANN prediction for *every* cosmology, at *all* redshifts $z \le 2$, is within ±0.5 per cent of the *N*-body power spectra up to $k \le 0.9\,h\,{\rm Mpc}^{-1}$. The PkANN performs exceedingly well within the boundaries of the restricted parameter space.

Next, we assess the accuracy of the PkANN network across the range for each of the six parameters, namely, $\Omega_{\rm m}h^2$, $\Omega_{\rm b}h^2$, $n_{\rm s}$, w, σ_8 and $\sum m_\nu$. We vary each parameter between its minimum and maximum values and bin the 200 cosmologies of the training set in 10 intervals across the parameter range. We calculate the prediction error for each bin. We repeat this for all six parameters and show the results for the $\Omega_{\rm m}h^2$ case in Fig. 8. The rows correspond to the 10 linearly spaced bins between $\Omega_{\rm m}h^2 = 0.11$–0.165. The columns are redshift $z = 0$ (left-hand

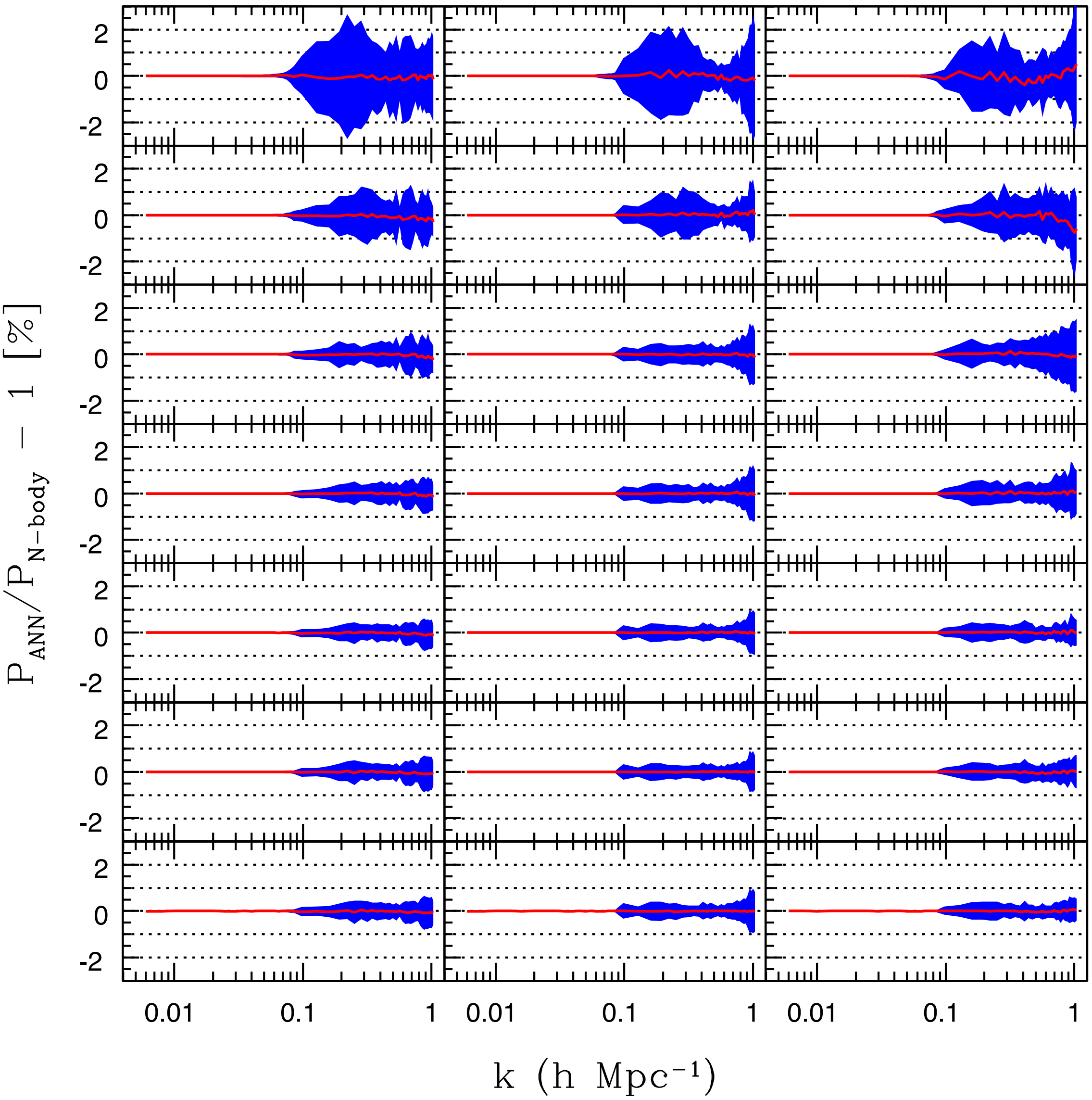

Figure 7. Similar to Fig. 4, using 330 testing set cosmologies.

panel), $z = 1$ (middle panel) and $z = 2$ (right-hand panel). As discussed above, we fix $N_{\rm hidden} = 70$. As expected, PkANN's performance degrades near the edges of the range $\Omega_{\rm m}h^2 = 0.11$–0.165 (compare the middle rows against the outer rows). Overall, the prediction errors remain within ±1 per cent of the N-body power spectra for $k \le 0.9\,h\,{\rm Mpc}^{-1}$. Results with the other five parameters are similar to Fig. 8. We summarize the prediction errors for all six parameters in Table 2.

4.2 PkANN error estimates

Our ANN framework successfully recreates the input power spectrum at sub-per cent level up to $k \le 0.9\,h\,{\rm Mpc}^{-1}$, and the overall accuracy of the PkANN interpolator is set by the force resolution and statistical variance from our N-body simulations. Running ENZO in a 200 h^{-1} Mpc box with 512^3 unigrid results in a matter power spectrum that is progressively suppressed from 1 per cent level at $k = 0.5\,h\,{\rm Mpc}^{-1}$ to 5 per cent level at $k = 0.9\,h\,{\rm Mpc}^{-1}$, when compared to spectrum calculated from high-resolution runs. Limited computing resources prohibited us from running higher resolution simulations. Since PkANN is built using conservative simulation settings described above, we expect all PkANN predictions to be suppressed at 1–5 per cent level between $k = 0.5$ and $0.9\,h\,{\rm Mpc}^{-1}$.

We follow the approach outlined in Jeong & Komatsu (2009) (see their appendix A) to roughly estimate the statistical error on our non-linear power spectrum from numerical simulations. A simulation box of length 200 h^{-1} Mpc corresponds to a fundamental wavenumber of $\delta k = 2\pi/200 = 0.0314\,h\,{\rm Mpc}^{-1}$. The number of independent k-modes available in a spherical shell at $k = 0.1\,h\,{\rm Mpc}^{-1}$ is $N_k = 2\pi(k/\delta k)^2 \approx 64$. With our 11 realizations per cosmology, this gives a relative error of $\sigma_{P(k)/P(k)} = 1/\sqrt{11N_k} \approx 4$ per cent at $k = 0.1\,h\,{\rm Mpc}^{-1}$. Higher k-modes are sampled more frequently and the corresponding sampling errors become progressively smaller, to $\sim$0.4 per cent at $k = 0.9\,h\,{\rm Mpc}^{-1}$.

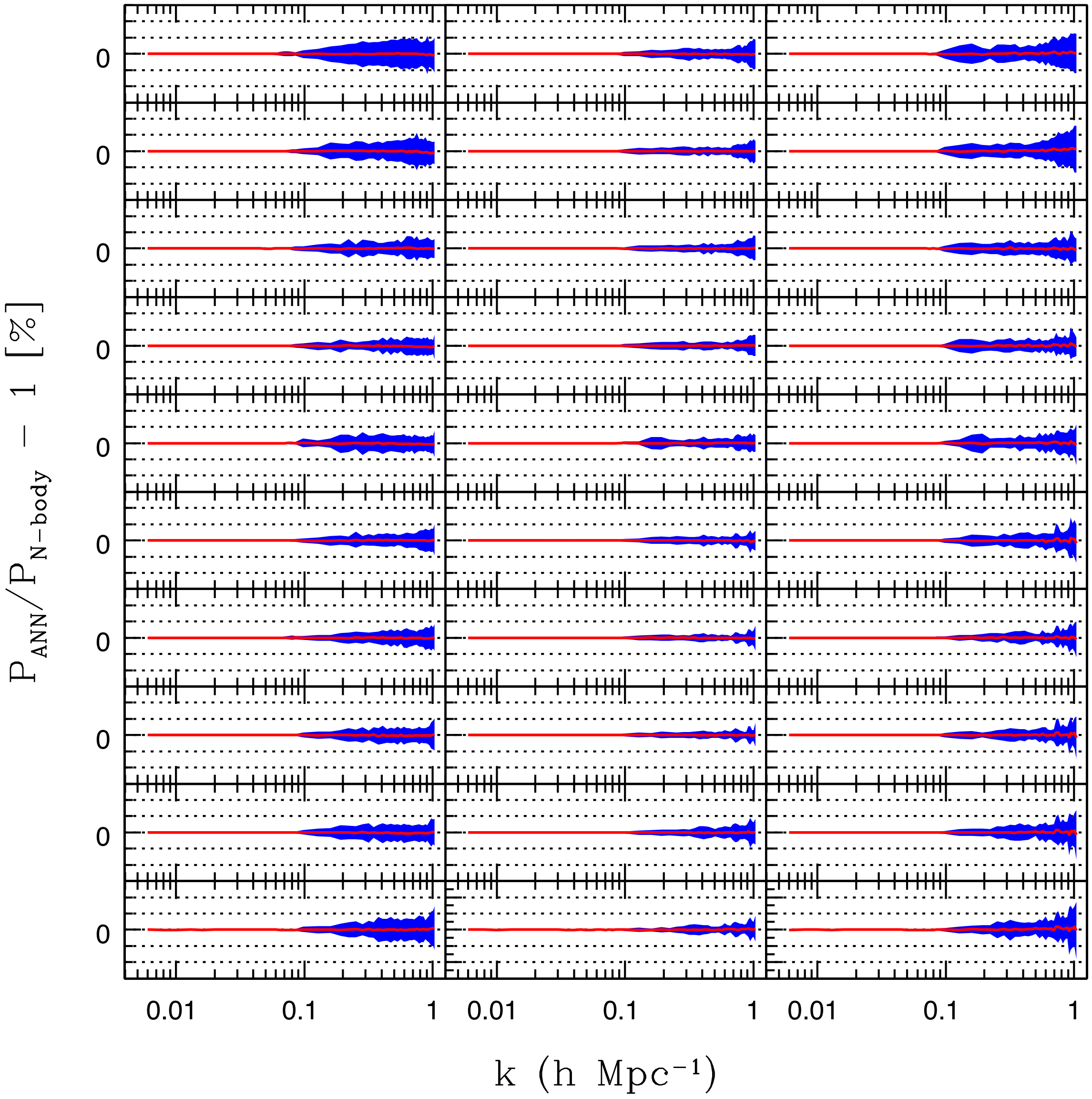

Figure 8. The 200 cosmologies of the training set are binned in 10 equal intervals between $\Omega_{\mathrm{m}}h^2 = 0.11$–$0.165$ (from top to bottom, in increasing order). The columns are redshift $z = 0$, 1, 2 (from left to right, respectively). $N_{\mathrm{hidden}} = 70$ for all panels. For each bin, PKANN's predictions are compared to the *N*-body power spectra and the residual errors (3σ CL) are plotted. Closer to the middle of the range $\Omega_{\mathrm{m}}h^2 = 0.11$–$0.165$ (middle rows), the prediction errors get smaller. Even near the edges (outer rows), the errors are within ±1 per cent of the *N*-body power spectra for $k \leq 0.9\,h\,\mathrm{Mpc}^{-1}$.

Table 2. Performance of the PKANN network as a function of the range of the six parameters, namely, $\Omega_{\mathrm{m}}h^2$, $\Omega_{\mathrm{b}}h^2$, n_{s}, w, σ_8 and $\sum m_\nu$. Each parameter range is subdivided into 10 equal intervals and the training set cosmologies are binned accordingly. The 3σ bounds on the PKANN prediction errors (in per cent) are mentioned for each bin, at redshifts $z = 0$, 1 and 2. The $\Omega_{\mathrm{m}}h^2$ case is shown in Fig. 8.

Bins		$\Omega_{\mathrm{m}}h^2$			$\Omega_{\mathrm{b}}h^2$			n_{s}			w			σ_8			$\sum m_\nu$	
	$z=0$	$z=1$	$z=2$	$z=0$	$z=1$	$z=2$	$z=0$	$z=1$	$z=2$	$z=0$	$z=1$	$z=2$	$z=0$	$z=1$	$z=2$	$z=0$	$z=1$	$z=2$
1	1.0	1.0	1.0	0.9	0.7	1.1	1.0	0.9	1.2	0.9	0.9	0.9	0.8	0.5	1.0	0.8	0.8	1.1
2	1.0	0.8	1.1	0.6	0.7	1.1	0.9	0.8	0.9	0.8	0.8	0.9	0.8	0.5	1.0	0.8	0.7	1.0
3	0.9	0.9	0.9	0.9	0.7	1.0	0.9	0.8	0.9	0.7	0.7	0.9	0.6	0.5	1.0	0.5	0.4	1.0
4	0.7	0.8	0.9	0.8	0.6	0.9	0.8	0.6	0.9	0.6	0.7	0.9	0.6	0.5	1.0	0.5	0.6	0.9
5	0.7	0.5	0.9	0.7	0.6	0.9	0.6	0.5	1.0	0.6	0.6	0.9	0.5	0.5	0.9	0.7	0.5	0.9
6	0.9	0.5	1.0	0.4	0.5	0.8	0.7	0.5	1.0	0.7	0.5	0.9	0.4	0.5	0.8	0.8	0.7	1.0
7	0.8	0.6	0.9	0.7	0.6	1.0	0.7	0.8	1.0	0.7	0.7	0.9	0.5	0.4	0.9	0.9	0.5	0.9
8	1.0	0.6	0.9	0.8	0.6	0.9	0.6	0.7	1.0	0.8	0.5	0.8	0.6	0.4	1.0	0.9	0.8	0.9
9	0.9	0.8	1.0	0.8	0.6	1.0	0.8	0.8	1.0	0.9	0.7	0.9	0.8	0.5	0.9	0.9	0.8	0.9
10	1.0	0.8	1.1	0.9	0.7	1.1	0.8	0.8	1.0	0.9	0.6	0.9	1.0	0.7	0.9	0.8	0.9	1.2

As mentioned earlier, we match the matter power spectra from one-loop standard PT with numerical simulations at $k = 0.085\,h\,\mathrm{Mpc}^{-1}$. Heitmann et al. (2010, their fig. 6) showed that small simulation volumes fail to capture linear evolution on the largest scales probed by the simulation box as well as miss the onset of non-linearity, resulting in the suppression of the matter power spectrum at ∼2–3 per cent level. As such, for a simulation box of length $200\,h^{-1}$ Mpc, we expect our spectra amplitudes to be in error at ∼3 per cent level around $k \approx 1\,h\,\mathrm{Mpc}^{-1}$.

PkANN can be used for spatially flat cosmological models with three species of degenerate massive neutrinos up to $\sum m_\nu = 1.1$ eV. Since our implementation of neutrinos in numerical simulations does not take into account the non-linear evolution of neutrino perturbations, this is expected to introduce errors in the estimated matter power spectrum. In Paper I, we discussed the expected errors by comparing our results with Brandbyge et al. (2008) and Brandbyge & Hannestad (2009). At redshift $z = 0$, our neutrino spectra for $\sum m_\nu$ up to 0.1, 0.475 and 0.95 eV are expected to be in error by $\lesssim$0.1, 4 and 10 per cent, respectively. The respective errors at $z = 1$ and 2 are $\lesssim$0.1, 3, 6 and $\lesssim$0.1, 3, 5 per cent. These error estimates are large for $\sum m_\nu > 0.475$ eV; however, it is important to note that the current constraints on the total neutrino mass are around 0.3 eV.

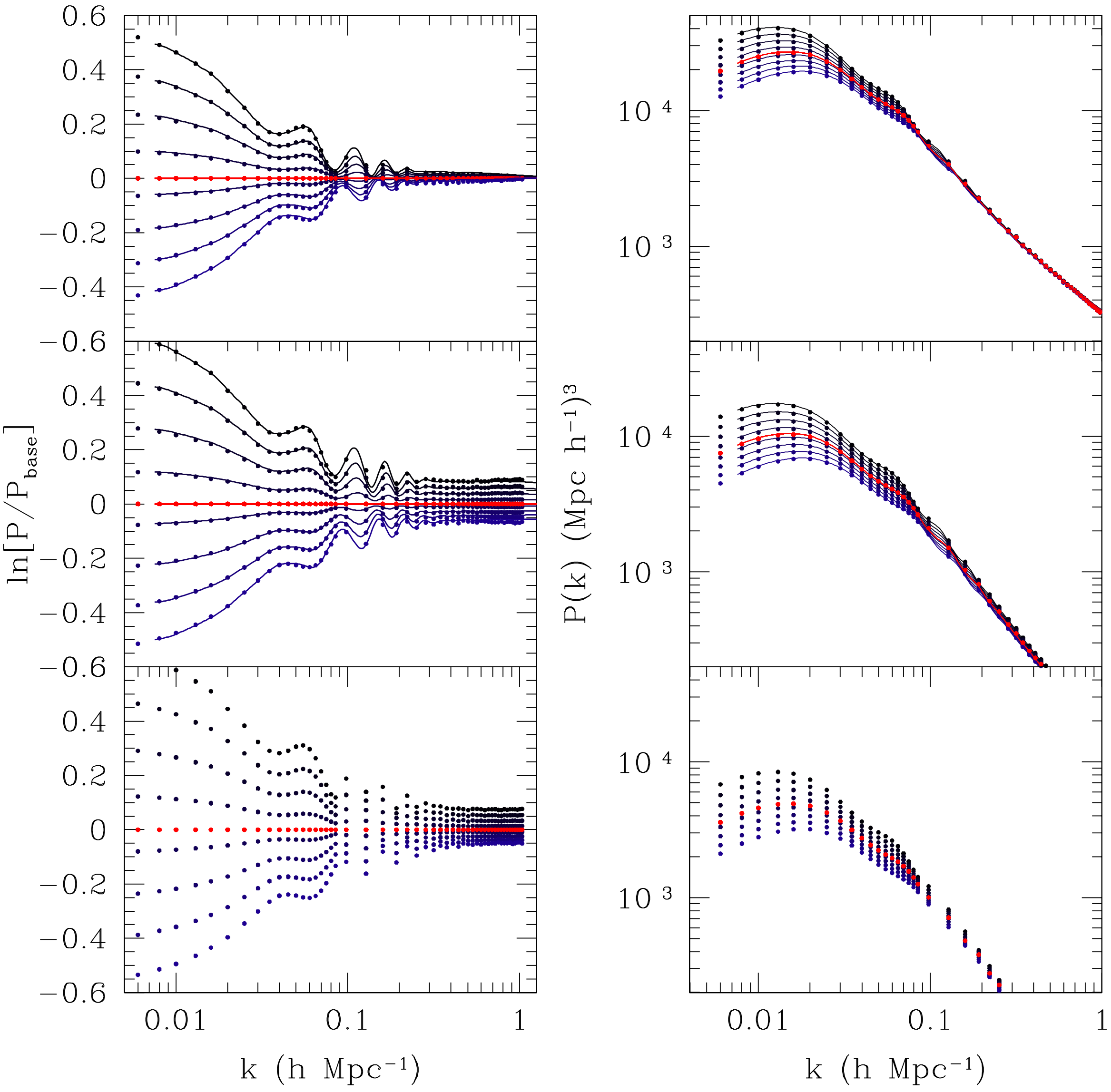

Figure 9. Variations in the power spectrum at redshift $z = 0$ (top row), $z = 1$ (middle row) and $z = 2$ (bottom row). Parameter $\Omega_\mathrm{m}h^2$ is varied between its minimum and maximum value (see testing set range, Table 1) while $\Omega_\mathrm{b}h^2$, n_s, w, σ_8 are fixed at their central values. $\sum m_\nu = 0$ to facilitate comparison with the h-fixed version of the COSMIC EMULATOR (Lawrence et al. 2010). The left-hand panels show natural logarithm of the ratio of the power spectra with different $\Omega_\mathrm{m}h^2$ to the base power spectrum. The cosmological parameters for the base power spectrum are $\boldsymbol{I} \equiv (0.135, 0.0225, 0.95, -1, 0.775, 0)$. The absolute power spectra are shown in the right-hand panels. Within each panel, the power spectra (from top to bottom) correspond to increasing values of $\Omega_\mathrm{m}h^2$. The predicted ratios using PkANN (dotted) are within 0.2 per cent of the COSMIC EMULATOR's predictions (solid lines). At $z = 2$, only PkANN predictions are shown since the h-fixed COSMIC EMULATOR is valid only for $z \leq 1$.

Using photometric redshifts measured from Sloan Digital Sky Survey III Data Release 8 (SDSS DR8; Aihara et al. 2011), de Putter et al. (2012) obtained constraints of $\sum m_\nu < 0.26$–0.36 eV. Using BAO and CMB data, the *Planck* survey (Planck Collaboration et al. 2013) finds an upper limit of 0.23 eV. Using numerical simulations, Wagner et al. (2012) studied the effect of neutrinos on the non-linear matter power spectrum for $\sum m_\nu \leq 0.3$ eV and found very similar results as ours in Paper I. For such low neutrino masses ($\sum m_\nu \leq 0.3$ eV), Brandbyge & Hannestad (2009, their fig. 1) show that at $z = 0$ non-linear neutrino corrections are at 0.3 per cent level, and negligible at higher redshifts. Overall, for $\sum m_\nu \leq 1.2$ eV, corrections are at 1.5 per cent level for $z \geq 1$.

To summarize, across all cosmological models (see Table 1) with $\sum m_\nu < 0.5$ eV, the PkANN interpolator is expected to be accurate at 5 per cent level for all redshifts $z \leq 2$. For models with $\sum m_\nu > 0.5$ eV, the spectra predictions are expected to be accurate at 5 per cent level only for $z > 1$ and degrade to ~10 per cent for $z \leq 1$.

4.3 Exploring cosmological parameter space with PkANN

Having built the power spectrum interpolator, we now study the behaviour of the power spectrum as a function of the cosmological parameters. Similar tests were performed by Heitmann et al. (2014). In Fig. 9, we show variations in the power spectrum at redshift $z = 0$ (top row), $z = 1$ (middle row) and $z = 2$ (bottom row). At each redshift, $\Omega_{\rm m}h^2$ is varied between its minimum and maximum value (see parameter ranges for the testing set, in Table 1) while $\Omega_{\rm b}h^2$, $n_{\rm s}$, w, σ_8 are fixed at their central values. We fix $\sum m_\nu = 0$ since we want to compare our PkANN predictions with the h-fixed version of the COSMIC EMULATOR, which is not trained for cosmological models with massive neutrinos. The left-hand panels show natural logarithm of the ratio of the power spectra with different $\Omega_{\rm m}h^2$ to the base power spectrum. The base power spectrum corresponds to the central values: $\Omega_{\rm m}h^2 = 0.135$, $\Omega_{\rm b}h^2 = 0.0225$, $n_{\rm s} = 0.95$, $w = -1$, $\sigma_8 = 0.775$, with $\sum m_\nu = 0$. The absolute power spectra are shown in the right-hand panels. Within each panel,

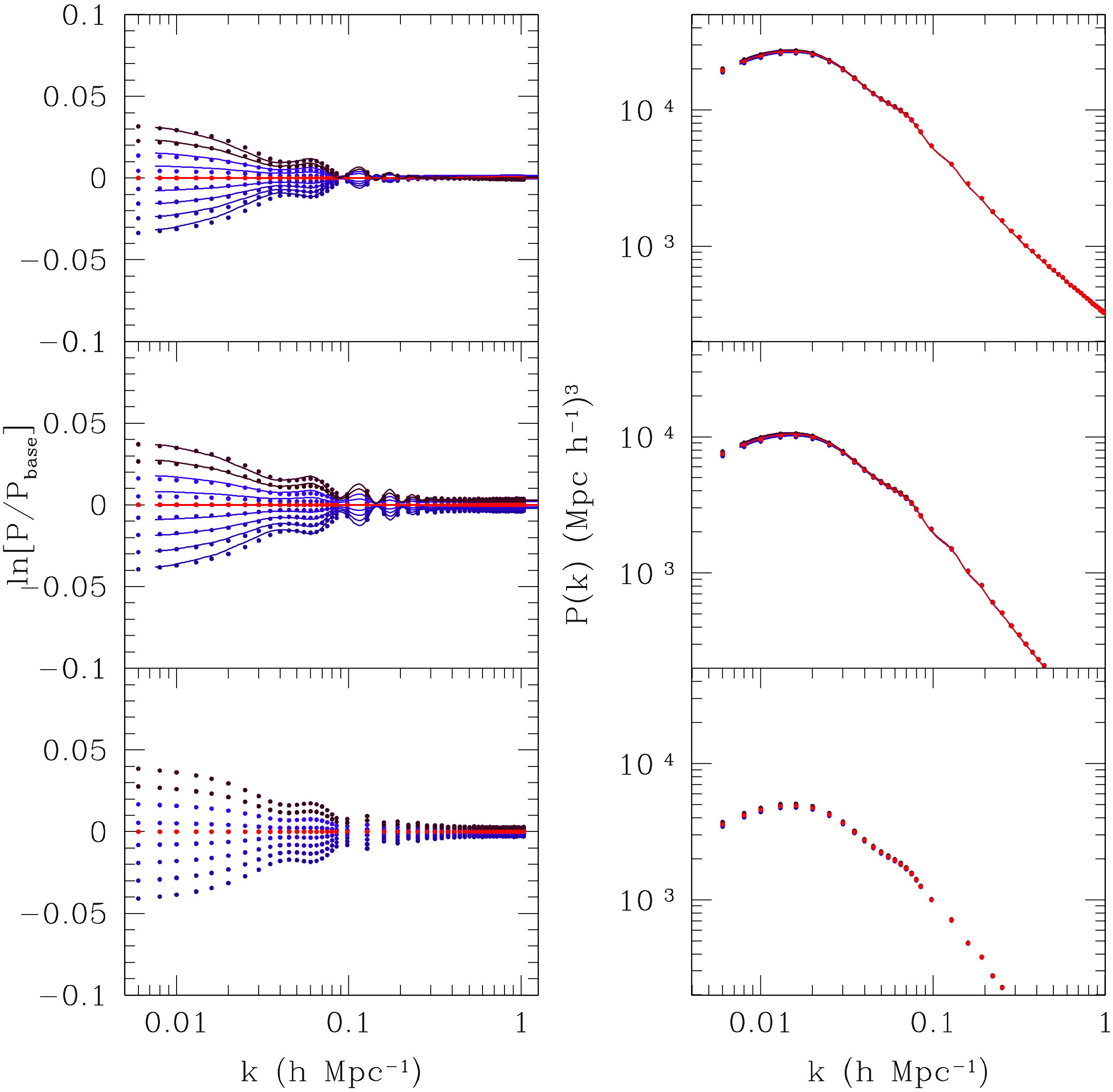

Figure 10. Similar to Fig. 9, but for a range of $\Omega_{\rm b}h^2$ values. Within each panel, the power spectra from bottom to top correspond to increasing $\Omega_{\rm b}h^2$ values.

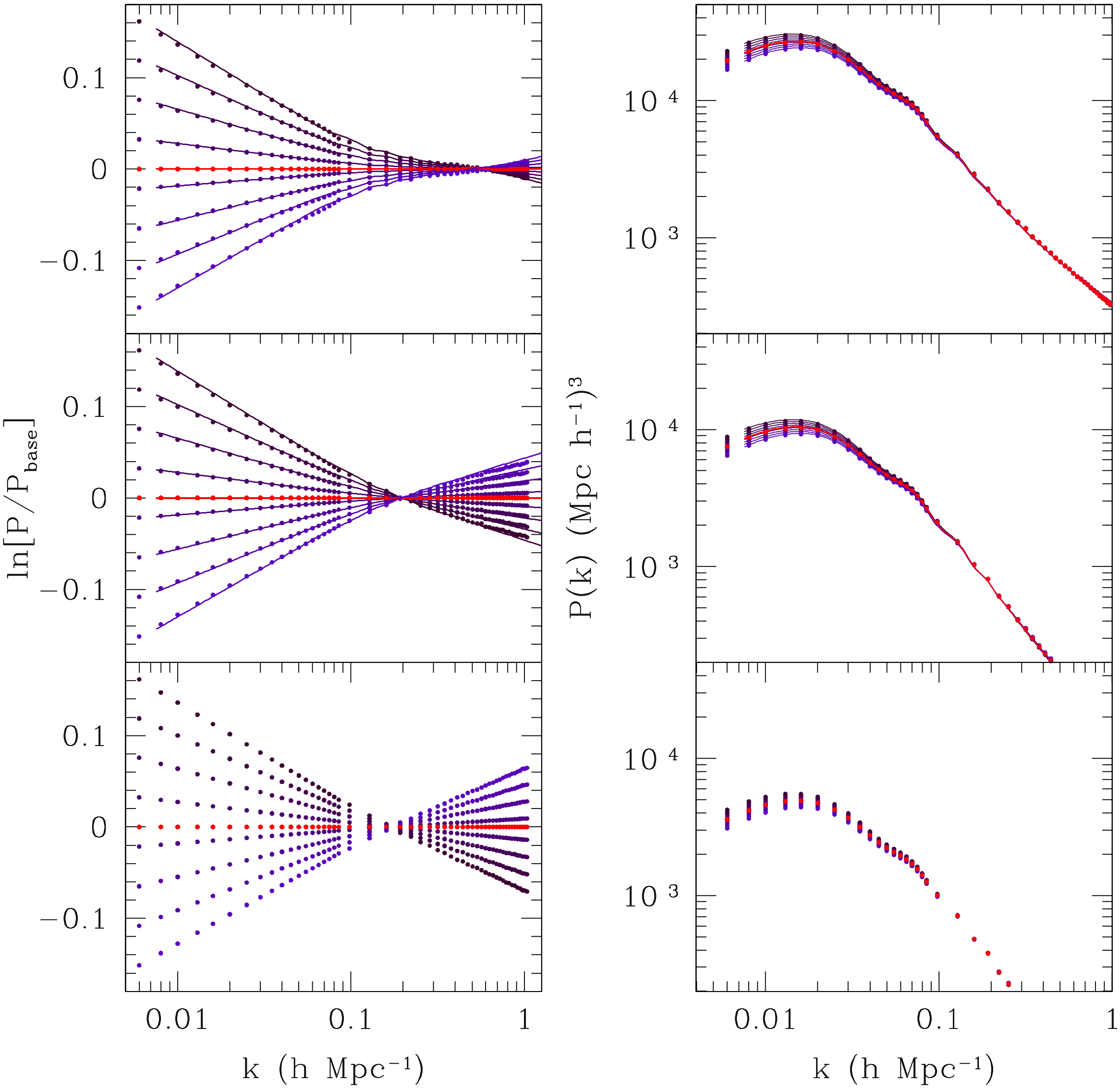

Figure 11. Similar to Fig. 9, but for a range of n_s values. Within each panel, the power spectra from top to bottom correspond to increasing n_s values.

the power spectra (from top to bottom) correspond to increasing $\Omega_m h^2$. Higher $\Omega_m h^2$ reduces the large-scale normalization of the power spectrum significantly. Accurate measurements of the power spectrum amplitude on large scales can help improve the constraints on $\Omega_m h^2$. PkANN predictions (dotted) agree well with the COSMIC EMULATOR (solid lines). Note that for redshift $z = 2$, we only show PkANN predictions since the h-fixed COSMIC EMULATOR can make predictions only up to $z = 1$.

In Figs 10–13, we vary $\Omega_b h^2$, n_s, w and σ_8, respectively. The power spectra trends from minimum to maximum values are as follows: top to bottom (n_s and w) and bottom to top ($\Omega_b h^2$ and σ_8). At $z = 0$, except σ_8, all other parameters affect the power spectrum predominantly on large scales ($\sim k < 0.1\,h\,\mathrm{Mpc}^{-1}$). Reducing uncertainties in the other parameters using small-scale data would be difficult unless one measures the power spectrum at higher redshifts where almost all parameters leave discernible imprints. Note that the power spectra converge around $k \sim 0.1\,h\,\mathrm{Mpc}^{-1}$ in the $\Omega_m h^2$, $\Omega_b h^2$, n_s and w plots. This is a direct consequence of our imposing the CMB constraint on the acoustic scale based on *WMAP* 7-year+BAO data. The acoustic scale is model dependent. Fixing its value to match observations requires adjusting the Hubble parameter h accordingly. As we discussed in Section 2, given a cosmological model $\boldsymbol{I}$, we compute h to satisfy the constraint $\pi d_{\mathrm{ls}}/r_s = 302.54$.

In Fig. 14, we plot the ratio of the non-linear spectra at redshifts $z = 0$ (upper panel) and $z = 1$ (lower panel) computed using PkANN and the h-fixed COSMIC EMULATOR. The cosmologies considered are all models of Section 4.3 as well as the 150 testing set cosmologies with $\sum m_\nu = 0$. The loss of power due to our use of 512^3 unigrid simulations is evident beyond $k = 0.6\,h\,\mathrm{Mpc}^{-1}$. Strictly speaking, a direct comparison of PkANN with COSMIC EMULATOR is not possible for two reasons: (i) to compute the Hubble parameter h, COSMIC EMULATOR uses the constraint equation $\pi d_{\mathrm{ls}}/r_s = 302.4$ for the acoustic scale, while PkANN is built using *WMAP* 7-year+BAO value of $\pi d_{\mathrm{ls}}/r_s = 302.54$, and (ii) the contribution of N_{eff} massless species of neutrinos to the radiation energy density is

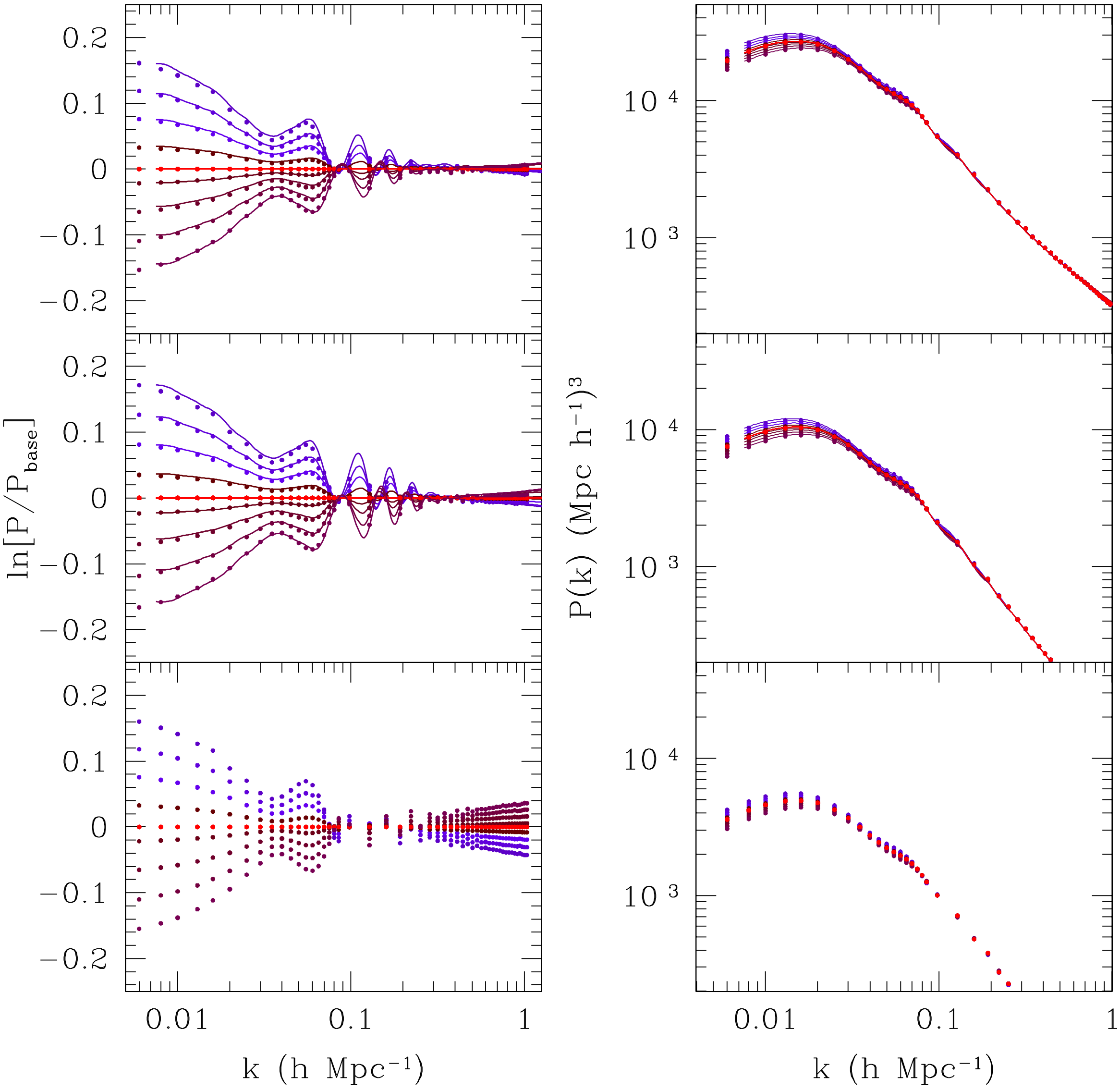

Figure 12. Similar to Fig. 9, but for a range of w values. Within each panel, the power spectra from top to bottom correspond to increasing w values.

given by

$$\rho_\nu = N_{\rm eff}\frac{7}{8}\left(\frac{4}{11}\right)^{4/3}\rho_\gamma, \quad (6)$$

where ρ_ν and ρ_γ are the neutrino and the photon energy densities, respectively. For PKANN (where $N_{\rm eff} = 3.00$), the pre-factor in equation (6) reduces to 0.68132. COSMIC EMULATOR uses 0.6851.

The above two factors result in ∼0.5 per cent variations in the estimates of the Hubble parameter h between PKANN and COSMIC EMULATOR. Running the two codes with identical cosmological parameters ($\Omega_{\rm m}h^2$, $\Omega_{\rm b}h^2$, $n_{\rm s}$, w, σ_8 with $\sum m_\nu = 0$) still corresponds to different cosmologies because the slightly different h values change the normalized densities ($\Omega_{\rm c}, \Omega_{\rm b}, \Omega_{\rm de}$ etc.). This changes the matter power spectra, both linear and therefore, the non-linear. To make this point clear, in Fig. 15, we show the linear power spectra for one of the 150 testing set cosmologies. The model parameters are $\Omega_{\rm m}h^2 = 0.120$, $\Omega_{\rm b}h^2 = 0.02213$, $n_{\rm s} = 0.97584$, $w = -1.0131$, $\sigma_8 = 0.7795$ with $\sum m_\nu = 0$. The upper panel shows the linear spectrum calculated using (i) CAMB (dotted) and (ii) h-fixed COSMIC EMULATOR (solid). Their ratio is shown in the lower panel. The two spectra differ at ∼2 per cent level because the normalized densities (summarized in Table 3) as computed by PKANN and COSMIC EMULATOR are different. This difference in the linear spectrum also correctly reflects in Fig. 14 where the solid triangles show the ratio of the non-linear spectra at $z = 0$ (upper panel) and $z = 1$ (lower panel) corresponding to the two cosmological models of Table 3. Figs 14 and 15 demonstrate the consistency between PKANN and COSMIC EMULATOR, with PKANN valid not only over an extended range of parameter space, but for models with massive neutrinos as well.

5 CONCLUSIONS

Machine learning techniques offer unparalleled advantage in analyses of large data sets of the kind being generated by current and

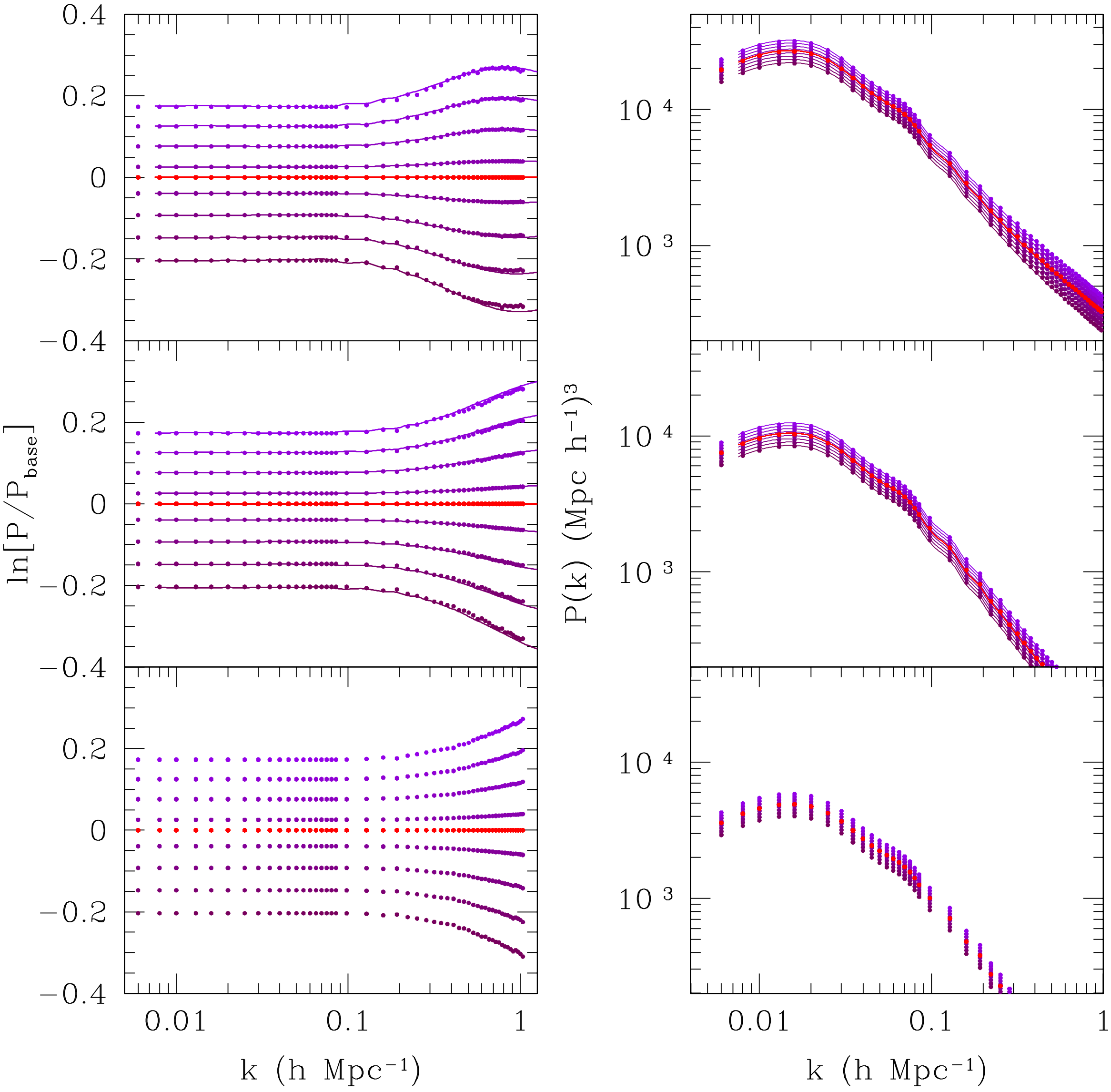

Figure 13. Similar to Fig. 9, but for a range of σ_8 values. Within each panel, the power spectra from bottom to top correspond to increasing σ_8 values.

upcoming surveys. A brute force application of numerical simulations can consume millions of CPU hours and may not be a feasible solution. Instead, by running a limited number of simulations, one can develop machine learning schemes. These schemes can then be used to efficiently handle new and previously unseen data.

In this paper, we have introduced PkANN – the non-linear matter power spectrum interpolator. Using a manageable number of *N*-body simulations, we have successfully developed a neural-network-based interpolating scheme that reconstructs the matter power spectrum over the parameter space spanning 3σ CL around the *WMAP* 7-year central values. Although PkANN reproduces the input power spectrum at sub-per cent level, its overall accuracy is limited by the accuracy of our *N*-body simulations. PkANN (i) predicts matter power spectrum up to redshifts $z \leq 2$, (ii) is capable of handling spatially flat cosmological models with/without massive neutrinos, as well as dark energy models with $w \neq -1$ constant equation of state parameter, (iii) is accurate at 5 per cent level up to $k \leq 0.9\,h\,\mathrm{Mpc}^{-1}$ for models with $\sum m_\nu < 0.5$ eV for all redshifts $z \leq 2$, (iv) is accurate at 5 (10) per cent level for redshifts $z > 1$ ($z \leq 1$) for models with $\sum m_\nu > 0.5$ eV.

The new generation of experiments, such as the DESI redshift maps, will measure matter density fluctuations with precision approaching ~1 per cent level. Such unprecedented precision, while having the potential to refine constraints on various cosmological parameters, poses a tremendous challenge on theoretical predictions of the matter power spectrum. Baryon physics alters the power spectrum at ~20 per cent level at $k \approx 10\,h\,\mathrm{Mpc}^{-1}$. van Daalen et al. (2011) have shown that AGN feedback reduces power relative to CDM-only simulations at per cent level at $k \approx 0.4\,h\,\mathrm{Mpc}^{-1}$. While the dark energy component in numerical simulations is usually assumed smooth and implemented only through its effects on the background evolution, Alimi et al. (2010) find that dark

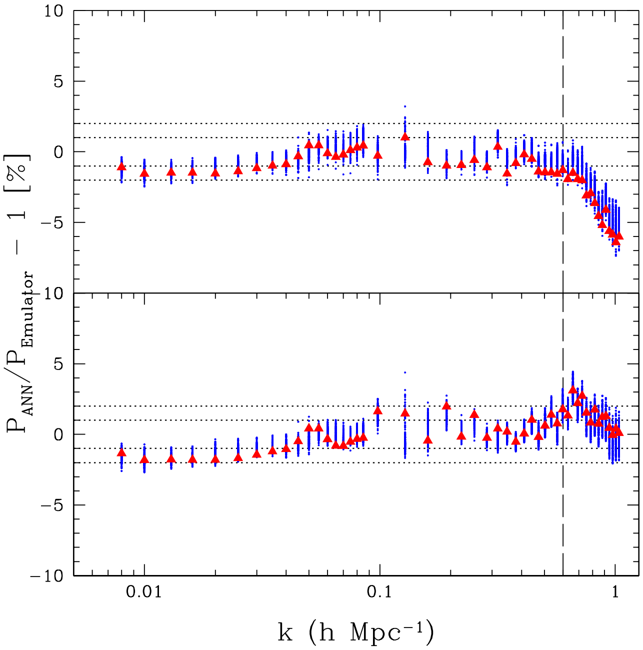

Figure 14. The ratio of the non-linear matter power spectra at $z = 0$ (upper panel) and $z = 1$ (lower panel) computed using PkANN and the h-fixed COSMIC EMULATOR for all models of Section 4.3 as well as the 150 testing set cosmologies with $\sum m_\nu = 0$. For clarity, we show the ratio for one of the 150 testing set cosmologies (mentioned in Table 3) by solid triangles. The two prediction schemes differ at 5 per cent level out to $k \lesssim 0.9\,h\,\mathrm{Mpc}^{-1}$. Beyond $k = 0.6\,h\,\mathrm{Mpc}^{-1}$, PkANN predictions fall off due to our use of unigrid simulations (see Section 4.2 for discussion). See Section 4.3 for a discussion on why the two schemes do not converge on larger scales.

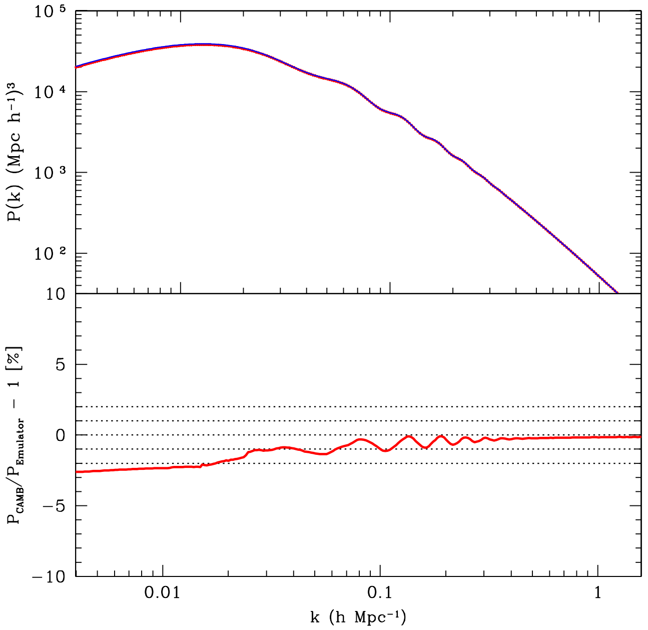

Figure 15. Linear power spectrum for one of the 150 testing set cosmologies. The model parameters are $\Omega_\mathrm{m}h^2 = 0.120$, $\Omega_\mathrm{b}h^2 = 0.02213$, $n_\mathrm{s} = 0.97584$, $w = -1.0131$, $\sigma_8 = 0.7795$ with $\sum m_\nu = 0$. The upper panel shows the linear spectrum calculated using (i) CAMB (dotted) and (ii) h-fixed COSMIC EMULATOR (solid). Their ratio is shown in the lower panel. The two differ at 2 per cent level on large scales due to the differences in the derived parameters (summarized in Table 3) as discussed in Section 4.3.

energy clustering leaves distinct imprints on the non-linear matter power spectrum. To extract any meaningful and unbiased information from current and upcoming data, such effects will need to be incorporated in N-body simulations and any fitting functions thereof. Although we did not consider a wide range of cosmological models such as the ones with non-zero spatial curvature, time-varying equation of state for dark energy or dark energy clustering, our ANN scheme can be readily extended for these cases by running extra simulations. The PkANN package is freely available at http://zuserver2.star.ucl.ac.uk/~fba/PkANN/PkANN.tar.gz.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation through TeraGrid resources provided by the NCSA. The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007–2013 Grant Agreement no. 279954). FBA acknowledges the support of a Royal Society URF, and OL acknowledges a Royal Society Wolfson Research Merit Award and an Advanced ERC Grant. SA would like to thank Pier Stefano Corasaniti and Yann Rasera for useful discussion. We thank the referee for providing constructive comments and help in improving the contents of this paper.

Table 3. Running PkANN and the h-fixed COSMIC EMULATOR with identical values of the six cosmological parameters ($\Omega_\mathrm{m}h^2$, $\Omega_\mathrm{b}h^2$, n_s, w, σ_8 with $\sum m_\nu = 0$) correspond to different normalized densities ($\Omega_\mathrm{c}, \Omega_\mathrm{b}, \Omega_\mathrm{de}$ etc.) due to the slight variations in the Hubble parameter h, as explained in Section 4.3. For one of the cosmological models (columns 2–7), the derived parameters are summarized (columns 8–11) for PkANN and the h-fixed COSMIC EMULATOR. The linear matter power spectra for these two sets of derived parameters are shown in Fig. 15. The ratio of the corresponding non-linear spectra is shown in Fig. 14 by solid triangles.

	Cosmological parameters						Derived parameters			
Scheme	$\Omega_\mathrm{m}h^2$	$\Omega_\mathrm{b}h^2$	n_s	w	σ_8	$\sum m_\nu$	h	Ω_c	Ω_b	Ω_de
PkANN	0.120	0.02213	0.97584	−1.0131	0.7795	0	0.7601	0.1694	0.0383	0.7923
h-fixed COSMIC EMULATOR	0.120	0.02213	0.97584	−1.0131	0.7795	0	0.7642	0.1676	0.0379	0.7945

REFERENCES

Adamson P. et al., 2008, Phys. Rev. Lett., 101, 131802
Agarwal S., Feldman H. A., 2011, MNRAS, 410, 1647 (Paper I)
Agarwal S., Abdalla F. B., Feldman H. A., Lahav O., Thomas S. A., 2012, MNRAS, 424, 1409 (Paper II)
Aihara H. et al., 2011, ApJS, 193, 29
Alimi J.-M., Füzfa A., Boucher V., Rasera Y., Courtin J., Corasaniti P.-S., 2010, MNRAS, 401, 775
Auld T., Bridges M., Hobson M. P., Gull S. F., 2007, MNRAS, 376, L11
Auld T., Bridges M., Hobson M. P., 2008, MNRAS, 387, 1575
Bird S., Viel M., Haehnelt M. G., 2012, MNRAS, 420, 2551
Bishop C. M., 1995, Neural Networks for Pattern Recognition. Oxford Univ. Press, New York
Brandbyge J., Hannestad S., 2009, J. Cosmol. Astropart. Phys., 5, 2
Brandbyge J., Hannestad S., 2010, J. Cosmol. Astropart. Phys., 1, 21
Brandbyge J., Hannestad S., Haugbølle T., Thomsen B., 2008, J. Cosmol. Astropart. Phys., 8, 20
Casarini L., Macciò A. V., Bonometto S. A., Stinson G. S., 2011, MNRAS, 412, 911
Collister A. A., Lahav O., 2004, PASP, 116, 345
de Putter R. et al., 2012, ApJ, 761, 12
Eisenstein D. J. et al., 2011, AJ, 142, 72
Fendt W. A., Wandelt B. D., 2007, ApJ, 654, 2
Ghosh A., 2011, preprint (arXiv:1111.4930)
Habib S., Heitmann K., Higdon D., Nakhleh C., Williams B., 2007, Phys. Rev. D, 76, 083503
Heitmann K., Higdon D., Nakhleh C., Habib S., 2006, ApJ, 646, L1
Heitmann K., Higdon D., White M., Habib S., Williams B. J., Lawrence E., Wagner C., 2009, ApJ, 705, 156
Heitmann K., White M., Wagner C., Habib S., Higdon D., 2010, ApJ, 715, 104
Heitmann K., Lawrence E., Kwan J., Habib S., Higdon D., 2014, ApJ, 780, 111
Hinshaw G. et al., 2013, ApJS, 208, 19
Hornik K., 1991, Neural Networks, 4, 251
Hurwitz E., Marwala T., 2012, preprint (arXiv:1208.4429)
Ito Y., 1991, Neural Networks, 4, 385
Ivezic Z. et al. (for the LSST Collaboration), 2008, preprint (arXiv:0805.2366)
Jenatton R., Gramfort A., Michel V., Obozinski G., Eger E., Bach F., Thirion B., 2011, preprint (arXiv:1105.0363)
Jeong D., Komatsu E., 2009, ApJ, 691, 569
KamLAND, 2008, Phys. Rev. Lett., 100, 221803
Komatsu E. et al., 2011, ApJS, 192, 18
Lahav O., Kiakotou A., Abdalla F. B., Blake C., 2010, MNRAS, 405, 168
Lawrence E., Heitmann K., White M., Higdon D., Wagner C., Habib S., Williams B., 2010, ApJ, 713, 1322
Levi M. et al. (representing the DESI collaboration), 2013, preprint (arXiv:1308.0847)
Lewis A., Challinor A., Lasenby A., 2000, ApJ, 538, 473
Nishimichi T. et al., 2009, PASJ, 61, 321
Norman M. L., Bryan G. L., Harkness R., Bordner J., Reynolds D., O'Shea B., Wagner R., 2007, preprint (arXiv:0705.1556)
O'Shea B. W., Bryan G., Bordner J., Norman M. L., Abel T., Harkness R., Kritsuk A., 2010, Astrophysics Source Code Library, record ascl:1010.072
Otten E. W., Weinheimer C., 2008, Rep. Progress Phys., 71, 086201
Pedregosa F., Gramfort A., Varoquaux G., Thirion B., Pallier C., Cauvet E., 2012, preprint (arXiv:1207.3520)
Planck Collaboration et al., 2013, A&A, preprint (arXiv:1303.5076)
Rudd D. H., Zentner A. R., Kravtsov A. V., 2008, ApJ, 672, 19
Saito S., Takada M., Taruya A., 2008, Phys. Rev. Lett., 100, 191301
Saito S., Takada M., Taruya A., 2009, Phys. Rev. D, 80, 083528
Sarazin C. L., White R. E., III, 1987, ApJ, 320, 32
Smith R. E. et al., 2003, MNRAS, 341, 1311
SNO, 2004, Phys. Rev. Lett., 92, 181301
Springel V., 2005, MNRAS, 364, 1105
Takahashi R., Sato M., Nishimichi T., Taruya A., Oguri M., 2012, ApJ, 761, 152
The Dark Energy Survey Collaboration, 2005, preprint (arXiv:0510346)
Upadhye A., Biswas R., Pope A., Heitmann K., Habib S., Finkel H., Frontiere N., 2013, preprint (arXiv:1309.5872)
van Daalen M. P., Schaye J., Booth C. M., Dalla Vecchia C., 2011, MNRAS, 415, 3649
Viel M., Haehnelt M. G., Springel V., 2010, J. Cosmol. Astropart. Phys., 6, 15
Wagner C., Verde L., Jimenez R., 2012, ApJ, 752, L31

APPENDIX A

The following is based on the treatment presented in Bishop (1995).

A1 PkANN cost function

PkANN is a single hidden-layer feed-forward network with sigmoid hidden nodes and linear output nodes. For training the PkANN neural network to predict the matter power spectrum, we consider a training set consisting of cosmological models for which we have full information about the non-linear matter power spectra $P_{\rm nl}$ (computed from N-body simulations) as a function of scale k and redshift z, as well as the underlying cosmological parameters: $\boldsymbol{I} \equiv (\Omega_{\rm m}h^2,\ \Omega_{\rm b}h^2,\ n_{\rm s},\ w,\ \sigma_8,\ \sum m_\nu)$. The joint likelihood of getting the set of matter power spectra $\{P_{\rm nl}(z;\boldsymbol{I}_t)\}$ for all cosmologies $\boldsymbol{I}_t$ in the training set is

$$\text{\L}\left[\{P_{\rm nl}(z;\boldsymbol{I}_t)\}\right] = \prod_{t=1}^{T} p[P_{\rm nl}(z;\boldsymbol{I}_t)]$$
$$= \prod_{t=1}^{T} p[P_{\rm nl}(z|\boldsymbol{I}_t)]\ p[\boldsymbol{I}_t], \tag{A1}$$

where $p[P_{\rm nl}(z|\boldsymbol{I}_t)]$ is to be interpreted as the conditional probability of getting spectrum $P_{\rm nl}(z)$ *given* cosmology $\boldsymbol{I}_t$, while $p[\boldsymbol{I}_t]$ is the unconditional probability that the cosmological parameters $\boldsymbol{I}$ take a particular setting of $\boldsymbol{I}_t$. The index t runs over all cosmologies $\boldsymbol{I}_t$ in the training set. We can take the product of the individual probabilities since each model $\boldsymbol{I}_t$ is drawn independently from the cosmological parameter space.

The weights $\boldsymbol{w}$ of the PkANN network are chosen (iteratively during network training) so as to minimize the negative logarithm of the likelihood Ł (which is equivalent to maximizing Ł),

$$\chi^2 = -\ln \text{\L} = \sum_{t=1}^{T} \ln p[P_{\rm nl}(z|\boldsymbol{I}_t)] + \sum_{t=1}^{T} \ln p[\boldsymbol{I}_t]. \tag{A2}$$

If the power spectrum is sampled at different values of scale k (the k-modes being represented by the set $\{k\}\,h\,{\rm Mpc}^{-1}$), we can write $p[P_{\rm nl}(z|\boldsymbol{I}_t)]$ as

$$p[P_{\rm nl}(z|\boldsymbol{I}_t)] = \prod_{k_i\in\{k\}} p[P_{\rm nl}(k,z|\boldsymbol{I}_t)], \tag{A3}$$

where the product k_i is over all the scales that form the set $\{k\}\,h\,{\rm Mpc}^{-1}$, and we have assumed that $P_{\rm nl}(k,z|\boldsymbol{I}_t)$ have independent distributions.

To suppress sampling uncertainties in the power spectrum $P_{\rm nl}(k,z|\boldsymbol{I}_t)$, the numerical simulation code is run multiple times with different seeds while keeping the underlying cosmological model $\boldsymbol{I}_t$ fixed. Assuming $P_{\rm nl}(k,z|\boldsymbol{I}_t)$ has Gaussian distribution about the true power spectrum $P_{\rm nl}^{\rm Tr}(k,z|\boldsymbol{I}_t)$ with variance σ^2, we can write the probability that a numerical run would give $P_{\rm nl}(k, z|\boldsymbol{I}_t)$ as

$$p[P_{\rm nl}(k, z|\boldsymbol{I}_t)] = \frac{1}{(2\pi\sigma^2)^{1/2}} \mathrm{e}^{\frac{-\left[P^{\rm Tr}_{\rm nl}(k,z|\boldsymbol{I}_t) - P_{\rm nl}(k,z|\boldsymbol{I}_t)\right]^2}{2\sigma^2}}. \tag{A4}$$

N-body codes give larger variance σ^2 on scales comparable to the simulation volume since the density field on these scales can only be sampled fewer times. However, to simplify the PkANN training algorithm, in equation (A4) we have assumed that the variance σ^2 is independent of the scale k and model $\boldsymbol{I}_t$.

Since the aim of developing PkANN is to model the true spectrum $P^{\rm Tr}_{\rm nl}(k, z|\boldsymbol{I}_t)$ by making an optimal choice for the network weights $\boldsymbol{w}$, we replace $P^{\rm Tr}_{\rm nl}(k, z|\boldsymbol{I}_t)$ in equation (A4) by the ANN prediction $P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I}_t)$ to get

$$p[P_{\rm nl}(k, z|\boldsymbol{I}_t)] = \frac{1}{(2\pi\sigma^2)^{1/2}} \mathrm{e}^{\frac{-\left[P^{\rm ANN}_{\rm nl}(k,z|\boldsymbol{w},\boldsymbol{I}_t) - P_{\rm nl}(k,z|\boldsymbol{I}_t)\right]^2}{2\sigma^2}}. \tag{A5}$$

Inserting equation (A5) into equation (A3), we get

$$p[P_{\rm nl}(z|\boldsymbol{I}_t)] = \frac{1}{(2\pi\sigma^2)^{N_{\rm out}/2}} \mathrm{e}^{\frac{-\sum_{k_i\in\{k\}}\left[P^{\rm ANN}_{\rm nl}(k,z|\boldsymbol{w},\boldsymbol{I}_t) - P_{\rm nl}(k,z|\boldsymbol{I}_t)\right]^2}{2\sigma^2}}, \tag{A6}$$

where $N_{\rm out}$ is the number of k-modes in the set $\{k\}$. Remember that, by construction, $N_{\rm out}$ is also the number of nodes in the output layer of the PkANN network. Using equation (A6), we can now write equation (A2) as

$$\begin{aligned}\chi^2(\boldsymbol{w}) = {} & \frac{1}{2\sigma^2}\sum_{t=1}^{T}\sum_{k_i\in\{k\}}\left[P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - P_{\rm nl}(k, z|\boldsymbol{I}_t)\right]^2 \\ & - T\ln\left[\frac{1}{(2\pi\sigma^2)^{N_{\rm out}/2}}\right] + \sum_{t=1}^{T}\ln p[\boldsymbol{I}_t].\end{aligned} \tag{A7}$$

We can drop the terms that do not depend on the weights $\boldsymbol{w}$, since these terms merely scale $\chi^2(\boldsymbol{w})$ without altering its location in the weight space. Thus, the cost function for the purposes error minimization can be written as

$$\chi^2(\boldsymbol{w}) = \frac{1}{2}\sum_{t=1}^{T}\sum_{k_i\in\{k\}}\left[P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - P_{\rm nl}(k, z|\boldsymbol{I}_t)\right]^2. \tag{A8}$$

Remember that the matter power spectrum is a function of scale k (h Mpc^{-1}). We sample the matter spectrum at discreet values in the range $0.006 \le k \le 1\,h$ Mpc^{-1} and assign the sampled spectrum to the output nodes of the neural network. The discreet values of scale k form the set $\{k\}$ h Mpc^{-1}. In equation (A8), the sum k_i is over all the nodes in the output layer, with each node sampling the matter power spectrum at some specific scale, k (h Mpc^{-1}). $P_{\rm nl}(k, z|\boldsymbol{I})$ is the non-linear matter power spectrum for the specific cosmology $\boldsymbol{I}$, computed using N-body simulations. Given the weights $\boldsymbol{w}$, $P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I})$ is the ANN's predicted power spectrum for the $\boldsymbol{I}$th cosmology. In our fitting procedure, we work with the ratio of the non-linear to linear power spectrum, namely $R(k, z) \equiv P_{\rm nl}(k, z)/P_{\rm lin}(k, z)$, where $P_{\rm lin}(k, z)$ is calculated using CAMB. As such, weighing equation (A8) by $P_{\rm lin}(k, z)$ gives

$$\begin{aligned}\chi^2(\boldsymbol{w}) &= \frac{1}{2}\sum_{t=1}^{T}\sum_{k_i\in\{k\}}\left[\frac{P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - P_{\rm nl}(k, z|\boldsymbol{I}_t)}{P_{\rm lin}(k, z|\boldsymbol{I}_t)}\right]^2 \\ &= \frac{1}{2}\sum_{t=1}^{T}\sum_{k_i\in\{k\}}\left[R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k, z|\boldsymbol{I}_t)\right]^2.\end{aligned} \tag{A9}$$

The ratio $R(k, z)$ is a flatter function and gives better performance, particularly at higher redshifts where the ratio tends to 1. Given the weights $\boldsymbol{w}$, $R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I})$ in equation (A9) is the network's prediction of the ratio $R(k, z|\boldsymbol{I})$ for the specific cosmology $\boldsymbol{I}$. The predicted non-linear spectrum $P^{\rm ANN}_{\rm nl}(k, z|\boldsymbol{w}, \boldsymbol{I})$ is recovered by multiplying $R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I})$ by the corresponding linear spectrum $P_{\rm lin}(k, z|\boldsymbol{I})$.

In equation (A9), optimizing the weights $\boldsymbol{w}$ so as to minimize $\chi^2(\boldsymbol{w})$ generates an ANN that predicts the power spectrum very well for the specific cosmologies in the training set. However, such a network might not make accurate predictions for cosmologies *not* included in the training set. This usually indicates (i) an overly simple network architecture (very few hidden layer nodes), (ii) very sparsely or poorly sampled parameter space and/or (iii) a highly complex non-linear mapping that actually overfits to the noise on the training data set. In order to generate smoother network mappings that generalize better when presented with new cosmologies that are not part of the training set, a penalty term $\chi^2_Q(\boldsymbol{w})$ is added to the cost function $\chi^2(\boldsymbol{w})$,

$$\chi^2_Q(\boldsymbol{w}) = \frac{\xi}{2}||\boldsymbol{w}||^2, \tag{A10}$$

where $||\boldsymbol{w}||^2$ is the quadratic sum of all the weights. The penalty term $\chi^2_Q(\boldsymbol{w})$ prevents the network weights from becoming too large during the training process by penalizing in proportion to their sum. The regularization parameter ξ controls the degree of regularization (smoothing) of a network's predictions. After having initialized ξ, its value is re-estimated during the training process iteratively. For the update formula, see Appendix A5. For its derivation, see Bishop (1995).

Thus, the overall cost function which is presented to the ANN for minimization with respect to the weights $\boldsymbol{w}$ is

$$\begin{aligned}\chi^2_{\rm C}(\boldsymbol{w}) = {} & \frac{1}{2}\sum_{t=1}^{T}\sum_{k_i\in\{k\}}\left[R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k, z|\boldsymbol{I}_t)\right]^2 \\ & + \frac{\xi}{2}||\boldsymbol{w}||^2.\end{aligned} \tag{A11}$$

A2 Quasi-Newton method

Quasi-Newton method, used for finding stationary points (local maxima and minima) of a function, assumes that the function can be approximated by a quadratic in the region around a stationary point. Taylor expanding the PkANN cost function $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) around some point $\boldsymbol{w}_0$ in the weight space and retaining terms up to second order, we get

$$\begin{aligned}\chi^2_{\rm C}(\boldsymbol{w}) = {} & \chi^2_{\rm C}(\boldsymbol{w}_0) + (\boldsymbol{w} - \boldsymbol{w}_0)^{\rm T}\boldsymbol{g}_{\boldsymbol{w}_0} \\ & + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}_0)^{\rm T}\mathbf{H}_{\boldsymbol{w}_0}(\boldsymbol{w} - \boldsymbol{w}_0),\end{aligned} \tag{A12}$$

where the superscript 'T'stands for the transpose and $\boldsymbol{g}_{\boldsymbol{w}_0}$ is defined to be the gradient of $\chi^2_{\rm C}$ evaluated at $\boldsymbol{w}_0$,

$$\boldsymbol{g}_{\boldsymbol{w}_0} \equiv \nabla\chi^2_{\rm C}\Big|_{\boldsymbol{w}_0}. \tag{A13}$$

$\mathbf{H}_{\boldsymbol{w}_0}$ is a symmetric $N_W \times N_W$ Hessian matrix (evaluated at $\boldsymbol{w}_0$) with elements

$$H_{ij}\Big|_{\boldsymbol{w}_0} \equiv \left.\frac{\partial^2\chi^2_{\rm C}}{\partial w_i \partial w_j}\right|_{\boldsymbol{w}_0}, \tag{A14}$$

where N_W (see equation 3) is the total number of nodes in the network. Note that in equation (A14), instead of referencing the weights by the relevant nodes they connect to, for the sake of clarity we refer to the weights with a single subscript running from 1 to N_W.

Taking the gradient of equation (A12) gives the local approximation for the gradient itself,

$$\boldsymbol{g}_w = \boldsymbol{g}_{w_0} + \mathbf{H}_{w_0}(\boldsymbol{w} - \boldsymbol{w}_0). \tag{A15}$$

To find the stationary point around $\boldsymbol{w}_0$, one sets $\boldsymbol{g}_w$ in equation (A15) to zero, thereby giving the *Newton step*,

$$\boldsymbol{w} = \boldsymbol{w}_0 - \mathbf{H}^{-1}_{w_0}\boldsymbol{g}_{w_0}. \tag{A16}$$

Since the cost function $\chi^2_{\rm C}(\boldsymbol{w})$ is not an exact quadratic function, the Newton step of equation (A16) does not point to the local minimum around $\boldsymbol{w}_0$. As such, we apply equation (A16) iteratively, and if the Hessian matrix is positive definite (i.e. all of its eigenvalues are positive), then each successive Newton step moves closer to the local minimum. If the initial choice of the weights $\boldsymbol{w}$ happens to be around a local maximum of $\chi^2_{\rm C}(\boldsymbol{w})$, then the Hessian matrix is not positive definite and the cost function may increase with each Newton step.

One can apply some modifications to the Newton method that guarantee convergence towards a local minimum, irrespective of the initial choice of the weights. Instead of taking a step in the *Newton direction* $(-\mathbf{H}^{-1}\boldsymbol{g})$, one proceeds in a *quasi-Newton direction* $(-\mathbf{G}\boldsymbol{g})$,

$$\boldsymbol{w} = \boldsymbol{w}_0 - \lambda_{w_0}\mathbf{G}_{w_0}\boldsymbol{g}_{w_0}, \tag{A17}$$

where matrix $\mathbf{G}$ represents an approximation to the inverse of the Hessian $\mathbf{H}^{-1}$, and λ is the size of the step taken along the quasi-Newton direction $-\mathbf{G}\boldsymbol{g}$. The step size λ is allowed to vary with each iteration to the weights. Its value is determined by proceeding in the direction $-\mathbf{G}\boldsymbol{g}$ until the minimum of the cost function is found along $-\mathbf{G}\boldsymbol{g}$. Thus, in equation (A17), $\lambda_{\boldsymbol{w}_0}$ is such that the gradient of $\chi^2_{\rm C}$ at $\boldsymbol{w}$ (namely, $\boldsymbol{g}_w$) vanishes along the direction $-\mathbf{G}_{w_0}\boldsymbol{g}_{w_0}$,

$$\left(-\mathbf{G}_{w_0}\boldsymbol{g}_{w_0}\right)^{\rm T}\boldsymbol{g}_w = 0. \tag{A18}$$

The quasi-Newton algorithm involves taking a series of steps τ of equation (A17), which can be written as

$$\boldsymbol{w}_{\tau+1} = \boldsymbol{w}_\tau - \lambda_{w_\tau}\mathbf{G}_{w_\tau}\boldsymbol{g}_{w_\tau}, \tag{A19}$$

with the step size λ_{w_τ} for the τth step being such that

$$\left(-\mathbf{G}_{w_\tau}\boldsymbol{g}_{w_\tau}\right)^{\rm T}\boldsymbol{g}_{w_{\tau+1}} = 0. \tag{A20}$$

At each step of the algorithm, $\mathbf{G}$ is constructed to be positive definite, ensuring that the direction $-\mathbf{G}\boldsymbol{g}$ proceeds towards a local minimum of the cost function. To construct $\mathbf{G}$, we use the BFGS method (see Appendix A4).

A3 PkANN cost function gradient

The overall cost function which is presented to the ANN for minimization with respect to the weights $\boldsymbol{w}$ is given by equation (A11).

We now derive the expression for its derivative with respect to the weights $\boldsymbol{w}$. PkANN's network architecture is $N_{\rm in} : N_1 : N_{\rm out}$ with two layers of adaptive weights. The first layer of weights w_{ji} connect the input layer nodes $(x_0, x_1, \ldots, x_i)$ to the hidden nodes $(z_1, \ldots, z_j)$. Note that the hidden bias node activation z_0 is permanently fixed at 1 and therefore does not receive any connections from the input layer. The activation of each hidden node is $z_j \equiv g(a_j)$, taking as its argument

$$a_j = \sum_{i=0}^{N_{\rm in}} w_{ji}x_i, \tag{A21}$$

where the sum is over all input nodes i (including the input bias) sending connections to the jth hidden node (barring the hidden bias node).

PkANN's hidden nodes have sigmoidal activations $g(a_j) = 1/[1 + \exp(-a_j)]$. The second layer of weights w_{kj} connect the hidden nodes $(z_0, z_1, \ldots, z_j)$ to the network outputs $(y_1, \ldots, y_k)$. The output nodes have linear activations $y_k = a_k$, with a_k being the weighted sum of all hidden nodes,

$$a_k = \sum_{j=0}^{N_1} w_{kj}z_j. \tag{A22}$$

PkANN has two layers of adaptive weights and we will consider the cost function derivatives separately for the two layers.

A3.1 Gradient with respect to first layer weights

Taking the gradient of equation (A11) with respect to a first layer weight w_{ji}, we get

$$\frac{\partial\left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{ji}} = \sum_{t,\{k\}}\left[R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k, z|\boldsymbol{I}_t)\right]\frac{\partial R^{\rm ANN}}{\partial w_{ji}} + \xi w_{ji}. \tag{A23}$$

Since $R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) = a(k, z|\boldsymbol{w}, \boldsymbol{I}_t)$ (see equation A22) for the output nodes, we get

$$\frac{\partial\left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{ji}} = \sum_{t,\{k\}}\left[R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k, z|\boldsymbol{I}_t)\right]\frac{\partial a^t_k}{\partial w_{ji}} + \xi w_{ji}, \tag{A24}$$

where we have introduced the shorthand notation $a^t_k \equiv a(k, z|\boldsymbol{w}, \boldsymbol{I}_t)$. Using equation (A22) for a_k together with sigmoidal activation for z_j, we get

$$\begin{aligned}\frac{\partial a^t_k}{\partial w_{ji}} &= \sum_{j'=0}^{N_1} w_{kj'}\frac{\partial z^t_{j'}}{\partial w_{ji}} \\ &= \sum_{j'=0}^{N_1} w_{kj'}\frac{\partial g\left(a^t_{j'}\right)}{\partial a^t_{j'}}\frac{\partial a^t_{j'}}{\partial w_{ji}}.\end{aligned} \tag{A25}$$

For sigmoidal activation functions, it is straightforward to show that

$$\frac{\partial g(a^t_j)}{\partial a^t_j} = g\left(a^t_j\right)\left(1 - g(a^t_j)\right). \tag{A26}$$

Inserting equation (A26) into equation (A25), we get

$$\frac{\partial a^t_k}{\partial w_{ji}} = \sum_{j'=0}^{N_1} w_{kj'}g^t_{j'}\left(1 - g^t_{j'}\right)\frac{\partial a^t_{j'}}{\partial w_{ji}}. \tag{A27}$$

Differentiating equation (A21) with respect to a first layer weight w_{ji}, we get

$$\frac{\partial a^t_{j'}}{\partial w_{ji}} = \sum_{i'=0}^{N_{\rm in}} x^t_{i'} \frac{\partial w_{j'i'}}{\partial w_{ji}}$$

$$= \sum_{i'=0}^{N_{\rm in}} x^t_{i'} \delta_{ii'} \delta_{jj'} \; = \; x^t_i \delta_{jj'}. \tag{A28}$$

Inserting equation (A28) into equation (A27), we get

$$\frac{\partial a^t_k}{\partial w_{ji}} = \sum_{j'=0}^{N_1} w_{kj'} g_j{}'^t \left(1 - g^t_{j'}\right) x^t_i \delta_{jj'}$$

$$= w_{kj} g^t_j \left(1 - g^t_j\right) x^t_i. \tag{A29}$$

From equations (A24) and (A29), we get our final equation for the derivative of the pkANN cost function with respect to the first layer of adaptive weights w_{ji} to be

$$\frac{\partial \left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{ji}} = \sum_{t,\{k\}} R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t)\, w_{kj} g^t_j \left(1 - g^t_j\right) x^t_i$$

$$- \sum_{t,\{k\}} R(k, z|\boldsymbol{I}_t)\, w_{kj} g^t_j \left(1 - g^t_j\right) x^t_i + \xi w_{ji}. \tag{A30}$$

A3.2 Gradient with respect to second layer weights

Taking the gradient of equation (A11) with respect to a second layer weight w_{kj}, we get

$$\frac{\partial \left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{kj}} = \sum_{t,\{k'\}} \left[R^{\rm ANN}(k', z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k', z|\boldsymbol{I}_t)\right] \frac{\partial R^{\rm ANN}}{\partial w_{kj}}$$

$$+ \xi w_{kj}. \tag{A31}$$

Since $R^{\rm ANN}(k', z|\boldsymbol{w}, \boldsymbol{I}_t) = a(k', z|\boldsymbol{w}, \boldsymbol{I}_t)$ (see equation A22) for the output nodes, we get

$$\frac{\partial \left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{kj}} = \sum_{t,\{k'\}} \left[R^{\rm ANN}(k', z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k', z|\boldsymbol{I}_t)\right] \frac{\partial a^t_{k'}}{\partial w_{kj}} + \xi w_{kj}, \tag{A32}$$

where as before, we use the shorthand notation $a^t_{k'} \equiv a(k', z|\boldsymbol{w}, \boldsymbol{I}_t)$. From equation (A22), we get

$$\frac{\partial a^t_{k'}}{\partial w_{kj}} = \sum_{j'=0}^{N_1} \frac{\partial w_{k'j'}}{\partial w_{kj}} z^t_{j'}$$

$$= \sum_{j'=0}^{N_1} \delta_{kk'} \delta_{jj'} z^t_{j'} \; = \; \delta_{kk'} z^t_j. \tag{A33}$$

Inserting equation (A33) into equation (A32), we get our final equation for the derivative of the pkANN cost function with respect to the second layer of adaptive weights w_{kj} to be

$$\frac{\partial \left[\chi^2_{\rm C}(\boldsymbol{w})\right]}{\partial w_{kj}} = \sum_{t,\{k'\}} \left[R^{\rm ANN}(k', z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k', z|\boldsymbol{I}_t)\right] \delta_{kk'} z^t_j + \; \xi w_{kj}$$

$$= \sum_t \left[R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t) - R(k, z|\boldsymbol{I}_t)\right] z^t_j + \xi w_{kj}. \tag{A34}$$

For any choice of weights $\boldsymbol{w}$, the network output vector $R^{\rm ANN}(k, z|\boldsymbol{w}, \boldsymbol{I}_t)$ is determined for each cosmology $\boldsymbol{I}_t$ in the training set, by progressing sequentially through the network layers, from inputs to outputs, calculating the activation of each node. Having calculated the activations and network outputs for all cosmologies, it is straightforward to compute the derivatives in equations (A30) and (A34).

A4 BFGS approximation for inverse-Hessian matrix

In order to minimize the pkANN cost function $\chi^2_{\rm C}(\boldsymbol{w})$ (see equation A11) with respect to the weights $\boldsymbol{w}$, the weights are first randomly initialized to $\boldsymbol{w}_0$ and then updated iteratively using equation (A19).

Updating the weights involves estimating $\mathbf{G}$ – an approximation to the inverse Hessian matrix $\mathbf{H}^{-1}$. The inverse Hessian $\mathbf{H}^{-1}$ evaluated at $\boldsymbol{w}_0$ is approximated by a $N_W \times N_W$ identity matrix (i.e. $\mathbf{G}_{\boldsymbol{w}_0} = \mathbf{I}$). Following our discussion in Appendix A2, the weight vector is updated to $\boldsymbol{w}_1$ as

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 - \lambda_{\boldsymbol{w}_0} \boldsymbol{g}_{\boldsymbol{w}_0} \tag{A35}$$

by stepping a distance $\lambda_{\boldsymbol{w}_0}$ in the quasi-Newton direction $-\boldsymbol{g}_{\boldsymbol{w}_0}$. Note that the gradient $\boldsymbol{g}_{\boldsymbol{w}_0}$ is computed using equations (A30) and (A34). The step size $\lambda_{\boldsymbol{w}_0}$ is such that the gradient of $\chi^2_{\rm C}$ at $\boldsymbol{w}_1$ (namely, $\boldsymbol{g}_{\boldsymbol{w}_1}$) vanishes along the direction $-\boldsymbol{g}_{\boldsymbol{w}_0}$,

$$-\boldsymbol{g}^{\rm T}_{\boldsymbol{w}_0} \boldsymbol{g}_{\boldsymbol{w}_1} = 0. \tag{A36}$$

To make any further updates in the weight space, one needs to evaluate $\mathbf{H}^{-1}_{\boldsymbol{w}_1}$. The inverse Hessian, being a $N_W \times N_W$ matrix, can be computationally expensive to calculate exactly for networks with $N_W \gtrsim 1000$ connections. We employ the BFGS method to approximate $\mathbf{H}^{-1}_{\boldsymbol{w}_1}$ by $\mathbf{G}_{\boldsymbol{w}_1}$. In general, for the $(\tau + 1)$ step, the approximation $\mathbf{G}_{w_{\tau+1}}$ is

$$\mathbf{G}_{\boldsymbol{w}_{\tau+1}} = \mathbf{G}_{\boldsymbol{w}_\tau} + \frac{1}{S_1}\left[\left(1 + \frac{S_2}{S_1}\right) \boldsymbol{a}\boldsymbol{a}^{\rm T} - \boldsymbol{a}\boldsymbol{b}^{\rm T}\mathbf{G}_{\boldsymbol{w}_\tau} - \mathbf{G}_{\boldsymbol{w}_\tau}\boldsymbol{b}\boldsymbol{a}^{\rm T}\right], \tag{A37}$$

where we use the following definitions for the vectors ($\boldsymbol{a}$ and $\boldsymbol{b}$) and the scalars (S_1 and S_2),

$$\boldsymbol{a} = \boldsymbol{w}_{\tau+1} - \boldsymbol{w}_\tau,$$

$$\boldsymbol{b} = \boldsymbol{g}_{\boldsymbol{w}_{\tau+1}} - \boldsymbol{g}_{\boldsymbol{w}_\tau},$$

$$S_1 = \boldsymbol{a}^{\rm T}\boldsymbol{b},$$

$$S_2 = \boldsymbol{b}^{\rm T}\mathbf{G}\boldsymbol{b}. \tag{A38}$$

At each step, the BFGS method makes increasingly more accurate approximations for $\mathbf{G}$. Moreover, since $\mathbf{G}$ is positive definite (by construction), the $\chi^2_{\rm C}(\boldsymbol{w})$ minimization algorithm is guaranteed to converge to a local minimum.

A5 Regularization parameter ξ

In situations where the training data is noisy, controlling the complexity of a network is crucial to avoid overfitting and underfitting issues. An overly complex network may fit the noise in the training data. On the other hand, a very simple network may not be able to capture the signal in a data set, leading to underfitting. Both overfitting and underfitting lead to models with low predictive performance. One of the methods employed to regularize the complexity of a neural network is to train the network by minimizing a cost function that includes a penalty term $\chi^2_Q(\boldsymbol{w})$ (e.g. see equation A10).

Small (large) values of the regularization parameter ξ lead to complex (simple) networks. Since the optimum value for ξ is not known a priori, its value is initialized randomly, and updated iteratively by the cost minimization algorithm.

Here, we only present the updating rule for ξ. For its derivation, refer Bishop (1995). The PkANN cost function (equation A11) can be written as

$$\chi^2_{\rm C}(\boldsymbol{w})=\beta\left(\frac{1}{2}\sum_{t,\{k\}}\left[R^{\rm ANN}(k,z|\boldsymbol{w},\boldsymbol{I}_t)-R(k,z|\boldsymbol{I}_t)\right]^2+\frac{\alpha}{2\beta}||\boldsymbol{w}||^2\right), \tag{A39}$$

where α and β are the regularization parameters with $\xi \equiv \alpha/\beta$ and $\beta = 1$. For the purposes of cost minimization, the overall scale factor β is irrelevant and the degree of regularization depends only on the ratio $\xi \equiv \alpha/\beta$. For networks where the number of training patterns $N_{\rm T}$ far exceeds the number of weights $N_{\rm W}$, Bishop (1995) derives the following updating rules for α and β:

$$\alpha_{\tau+1} = \frac{N_{\rm W}}{||\boldsymbol{w}_\tau||^2}, \tag{A40}$$

$$\beta_{\tau+1} = \frac{N_{\rm T}}{\chi^2(\boldsymbol{w}_\tau)}, \tag{A41}$$

where $\chi^2(\boldsymbol{w})$ (see equation A9) is the sum of squares of residuals for the training data. Thus, we update ξ as

$$\xi_{\tau+1} = \frac{N_{\rm W}}{N_{\rm T}}\frac{\chi^2(\boldsymbol{w}_\tau)}{||\boldsymbol{w}_\tau||^2}. \tag{A42}$$

From equation (A42), we see that for sufficiently complex networks ($N_{\rm W} \gg 1$) with lots of training data ($N_{\rm T} \gg N_{\rm W}$), the parameter $\xi \ll 1$. It shows that underfitting and overfitting issues can be avoided by simply choosing network architectures that satisfy conditions: (i) $N_{\rm W} \gg 1$ and (ii) $N_{\rm T} \gg N_{\rm W}$. However, both these conditions can put tremendous load on the computing resources. In situations where the computing time is at a premium, a penalty term is used to achieve a balance between computing load and desired prediction accuracy of the neural network.

This paper has been typeset from a TEX/LATEX file prepared by the author.