Published by Institute of Physics Publishing for SISSA/ISAS
Received: *April 24, 2002*
Revised: *May 30, 2002*
Accepted: *May 31, 2002*

Neural network parametrization of deep-inelastic structure functions

Stefano Forte,[a] **Lluís Garrido,**[b] **José I. Latorre**[b] **and Andrea Piccione**[c]

[a] *INFN, Sezione di Roma Tre*
Via della Vasca Navale 84, I-00146 Rome, Italy
[b] *Departament d'Estructura i Constituents de la Matèria, Universitat de Barcelona,*
Diagonal 647, E-08028 Barcelona, Spain
[c] *INFN sezione di Genova and*
Dipartimento di Fisica, Università di Genova,
via Dodecaneso 33, I-16146 Genova, Italy
E-mail: forte@roma3.infn.it, garrido@ifae.es, latorre@sophia.ecm.ub.es, Andrea.Piccione@ge.infn.it

Abstract: We construct a parametrization of deep-inelastic structure functions which retains information on experimental errors and correlations, and which does not introduce any theoretical bias while interpolating between existing data points. We generate a Monte Carlo sample of pseudo-data configurations and we train an ensemble of neural networks on them. This effectively provides us with a probability measure in the space of structure functions, within the whole kinematic region where data are available. This measure can then be used to determine the value of the structure function, its error, point-to-point correlations and generally the value and uncertainty of any function of the structure function itself. We apply this technique to the determination of the structure function F_2 of the proton and deuteron, and a precision determination of the isotriplet combination $F_2[p-d]$. We discuss in detail these results, check their stability and accuracy, and make them available in various formats for applications.

Keywords: QCD, Deep Inelastic Scattering, Hadronic Colliders.

http://jhep.sissa.it/archive/papers/jhep052002062/jhep052002062.pdf

JHEP05(2002)062

Contents

JHEP05(2002)062

1. Determining structure functions

Recently, a considerable amount of theoretical and experimental effort has been invested in the accurate determination of the parton distributions of the nucleon, and in particular their associate errors, in view of the accurate computation of collider processes and determination of QCD parameters [1]–[5]. The determination of parton distributions requires unfolding the experimentally measurable structure functions in terms of their parton content, by resolution of the QCD evolution equations. However, for many applications, it is desirable and sufficient to deal directly with the experimentally measurable structure functions themselves. For instance, the nonsinglet structure function F_2 coincides with the (DIS-scheme) quark distribution, and thus in particular the strong coupling can be extracted directly from its scaling violations [6]. Also, knowledge of the structure function F_2 is needed for data analysis: for instance, when extracting polarized structure functions from cross-section asymmetries [7].

For all these applications, one needs to know the structure function F_2 as a function of x and Q^2, and not only for the particular values where it has been measured. Therefore, even though the laborious task of disentangling the parton content of the target and solving the evolution equations is not necessary if one directly determines the structure function, the central problem is the same as in the determination of parton distributions. Namely, one has to determine a function, and associate error, from a finite set of measurements. Because a function is an infinite-dimensional object, this is an a priori ill-posed problem, unless one makes some extra assumptions. The simplest way of implementing suitable extra assumption is to assume that the given function has a certain functional form, parametrized by a finite number of parameters, which can then be fitted from the data. However, the choice of a specific functional form is in general quite restrictive: therefore, it is thus a source of bias, i.e. systematic error which is very difficult to control. Furthermore, when fitting a fixed functional form to the data, it is very hard to obtain a determination not only of the best-fit parameters, but also of their errors (and correlations): systematic uncertainties are difficult to take into account [2, 5], and conventional error propagation and determination of the covariance matrix in a fixed parameter space is unwieldy and may fail because of linearization problems.

A way out of these difficulties, based on the direct determination of the measure in the space of parton distributions by Monte Carlo sampling has been proposed [3, 4]. However, it is computationally quite intensive, and it has not yet led to a full set of parton distributions with errors. Here, we propose a new approach to the determination of structure functions which also aims at the determination of the measure in the space of structure functions. Our approach is based on the use of neural networks as basic interpolating tools. The peculiar feature of neural networks which we will exploit is that they can provide an interpolation without requiring any assumption (other than continuity) on the functional form of the underlying law fulfilled by the given observable. Hence, given a sampling of the measure in the space of function over a finite set of points, the neural networks can provide an unbiased interpolation which gives us the measure for all points, at least within a range of x and Q^2 where the sampling provided by the data is fine enough.

We will use this procedure to construct parametrizations of the structure function F_2 for the proton, the deuteron, and an independent precision determination of the nonsinglet combination $F_2^p - F_2^d$. We will analyze in detail the properties of our results, and show that they reproduce correctly the information contained in the data, while interpolating smoothly between data points. We will also see that when the separation of data points in x and Q^2 is smaller than a certain correlation length, the neural net manages to combine the corresponding experimental information, thus leading to a determination of the structure function which is more accurate than that of individual experiments, and we will devise statistical tests to ascertain that this is achieved in an unbiased way.

JHEP05(2002)062

In section 2 we will introduce the problem of constructing a parametrization of structure functions, the data that we will use, discuss previous solutions of this problem, and motivate our alternative approach. In section 3 we will then provide some general background on neural networks, and in section 4 we will describe our method for the construction of a neural parametrization of structure functions. Finally, in section 5 we will turn to a discussion and assessment of our results. First we will discuss the theoretically simpler non-singlet structure function: we will examine the compatibility of different experiments, and the ability of the neural networks to combine different pieces of experimental information. Then, we will tackle the determination of the individual proton and deuteron structure functions, and discuss the accuracy of our results.

2. Parametrization of structure functions

Our aim in this paper is to construct a simultaneous parametrization of the structure functions F_2 for the proton and deuteron: we start from a set of measurements of a structure function, say F_2, for a discrete set of values of x and Q^2, and we wish to determine $F_2(x, Q^2)$ as a function defined in a certain range of x and Q^2. We choose the structure function F_2 as determined in neutral-current DIS of charged leptons, since an especially wide set of experimental determinations of it is available, and also because it is of direct theoretical interest. Studies of charged-current, neutrino, and polarized structure functions will be left to future work.

A simultaneous determination of the structure functions for proton and deuteron is very useful for applications, since it amounts to a simultaneous determination of the isospin singlet and triplet components of the structure function. Also, such a simultaneous determination poses peculiar theoretical problems, given that we aim at a determination of the full set of correlations.

In fact, we will provide both a determination of the correlated pair of proton and deuteron structure functions, as well as a separate determination of the isotriplet component. This is both useful and interesting because, as we shall see, data on the isotriplet combination turn out to be affected by much larger percentage errors, i.e. to have a much less favourable signal-to-noise ratio: the isotriplet is generally a small number obtained as the difference of two numbers which are by an order of magnitude larger. Therefore, on the one hand a more precise determination can be obtained by a dedicated analysis, rather than just taking a difference, on the other hand this determination will turn out to have peculiar features which deserve a specific study.

Parametrizations of structure functions have been constructed before, and used for the applications discussed in the introduction. In this section, we will first describe the specific data set that we will use, and then briefly review existing approaches to the parametrization of structure functions and their shortcomings.

2.1 Experimental data

We will use the data of the New Muon Collaboration (NMC) [8] and the BCDMS (Bologna-CERN-Dubna-Munich-Saclay) Collaboration [9] on the structure function F_2 of proton and deuteron. These data provide a simultaneous determination of the proton and deuteron

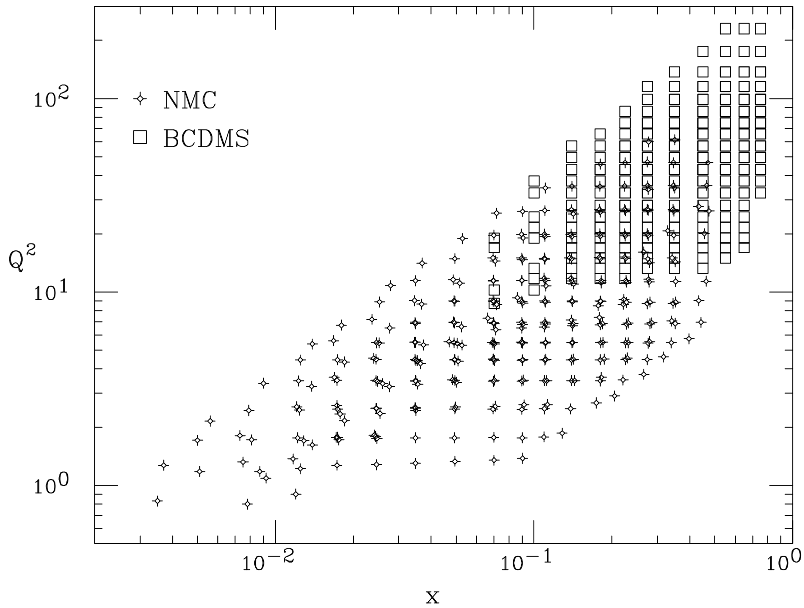

Figure 1: NMC and BCDMS kinematic range.

structure functions in the same kinematic region, and provide the full set of correlated experimental systematics for these measurements. Earlier data from SLAC are not competitive with these in terms of accuracy and kinematic coverage. The more recent HERA data are available in a much wider kinematic region, but only for proton targets. Another set of joint proton and deuteron measurements was performed by the E665 Collaboration [10]. These data are mostly concentrated at low Q^2 (and low x): 80% of the data have $Q^2 < 1\,\mathrm{GeV}^2$. They are therefore somewhat less interesting for perturbative QCD applications, and we will leave them out of the present analysis. We have checked that their inclusion would have a negligible impact on the determination of the structure function in the region covered by NMC and BCDMS. The kinematic coverage of the data which we include in our analysis is displayed in figure 1. A better coverage of the large x region at moderate Q^2 might be achievable in the near future by combining some of the earlier SLAC data [11] with forthcoming JLAB data [12].

2.1.1 NMC

The NMC data consist of four data sets for the proton and the deuteron structure functions corresponding to beam energies of 90 GeV (72 p and 72 d data points), 120 GeV (64 + 64 data points), 200 GeV (74 + 74 data points), and 280 GeV (78 + 78 data points). They cover the kinematic range $0.002 \le x \le 0.60$ and $0.5\,\mathrm{GeV}^2 \le \mathrm{Q}^2 \le 75\,\mathrm{GeV}^2$. The systematic errors, given in ref. [8], are:

- uncertainty on the incoming and outgoing beam energies, fully correlated between proton and deuteron data, but independent for data taken at different beam energies (E, E');

JHEP05(2002)062

- radiative corrections, fully correlated between all energies, but independent for proton and deuteron (RC);
- acceptance (AC) and reconstruction efficiency (RE) , fully correlated for all data sets;
- normalization uncertainty, correlated between the proton and the deuteron data, but independent for data taken at different beam energies (σ_N).

In this experiment, correlation between proton and deuteron data are due to the fact that both targets were exposed to the beam simultaneously.

The uncertainties due to acceptance range from 0.1 to 2.5% and reach at most 5% at large x and Q^2. The uncertainty due to radiative corrections is highest at small x and large Q^2 and is at most 2%. The uncertainty due to reconstruction efficiency is estimated to be 4% at most. The uncertainties due to the incoming and the scattered muon energies contribute to the systematic error by at most 2.5%. The normalization uncertainty is 2%.

2.1.2 BCDMS

The BCDMS data consist of four data sets for the proton structure function, corresponding to beam energies of 100 GeV (97 data points), 120 GeV (99 data points), 200 GeV (79 data points) and 280 GeV (76 data points), and three data sets for the deuteron structure function corresponding to beam energies of 120 GeV (99 data points), 200 GeV (79 data points) and 280 GeV (76 data points). They cover the kinematic range $0.06 \leq x \leq 0.80$ and $7\,\mathrm{GeV}^2 \leq \mathrm{Q}^2 \leq 280\,\mathrm{GeV}^2$.

The systematic errors are:[1]

- calibration of the incoming muon (beam) energy (f_b);
- calibration of the outgoing muon energy (spectrometer magnetic field) (f_s);
- spectrometer resolution (f_r);
- absolute normalization uncertainty (σ_{N_a});
- relative normalization uncertainties (σ_{N_t}, σ_{N_b}).

All these sources of systematics are fully correlated for all targets and for all beam energies, despite the fact that measurements with different targets were done at different times, for the following reasons [13]: the calibration of the incoming beam energy E was dominated by a systematic uncertainty which was more stable in time than the precision of measurements; the calibration of the outgoing muon energy was reproducible to high relative accuracy throughout the experiment, independent of the target or beam energy; the resolution of the spectrometer depended on the constituent material, which did not change during the running of the experiment.

[1] Note that the full set of systematics is listed in the preprint version but not in the published version of ref. [9].

JHEP05(2002)062

The uncertainty due to beam energy is smaller than 5%, and those due to magnetic field and outgoing muon energy are smaller than 10%; all uncertainties are most relevant at large x and small Q^2. The absolute cross-section normalization error is 3%; relative normalization errors are 2% between data taken with different targets, 1% between data taken at different beam energies for the proton, 1% between data taken at 120 GeV and 200 GeV and 1.5% between data taken at 200 GeV and 280 GeV for the deuteron.

2.1.3 Correlation and covariance matrices

The structure of the correlation matrix, expressed in terms of the sources of systematics discussed above is as follows. For NMC

$$\begin{aligned}\rho_{ij} = & \frac{F_i F_j}{\sigma_{i,\mathrm{tot}}\sigma_{j,\mathrm{tot}}} \times \\ & \times \left(E_i\, E_j + E_i'\, E_j' + AC_i\, AC_j + RC_i\, RC_j + RE_i\, RE_j + \sigma_N^2\right)\end{aligned} \tag{2.1}$$

where $\sigma_{\mathrm{tot}} = \sqrt{\sigma_s{}^2 + F(\sigma_c^2 + \sigma_N^2)}$; σ_s is the statistical error, σ_c is the combination of correlated systematic errors, σ_N is the normalization error, and systematic and normalization errors are given in percentage.

For BCDMS

$$\rho_{ij} = \frac{F_i F_j}{\sigma_{i,\mathrm{tot}}\sigma_{j,\mathrm{tot}}}\left(f_{b,i}\, f_{b,j} + f_{s,i}\, f_{s,j} + f_{r,i}\, f_{r,j} + \sigma_N^2\right) \tag{2.2}$$

with $\sigma_N = \sqrt{\sigma_{N_a}^2 + \sigma_{N_t}^2 + \sigma_{N_b}^2}$, where σ_{N_a} is a global normalization error, σ_{N_t} is the relative normalization between different targets and σ_{N_b} is the relative normalization between different beam energies.

The covariance matrix is defined as

$$\mathrm{cov}_{ij} = \rho_{ij}\sigma_i\sigma_j\,. \tag{2.3}$$

In the sequel, we will also be interested in the covariance matrix of data for the nonsinglet structure function $F^{NS} \equiv F^p - F^d$. In general the covariance of two functions of x_i, x_j is related to the covariance of these variables by

$$\mathrm{cov}(f(\vec{x}), g(\vec{x})) = \sum_{i,j} \frac{\partial f}{\partial x_i}\frac{\partial g}{\partial x_j}\mathrm{cov}(x_i, x_j)\,. \tag{2.4}$$

For the nonsinglet structure function we get

$$\mathrm{cov}(F_i^p - F_i^d, F_j^p - F_j^d) = \mathrm{cov}(F_i^p, F_j^p) + \mathrm{cov}(F_i^d, F_j^d) - \mathrm{cov}(F_i^p, F_j^d) - \mathrm{cov}(F_i^d, F_j^p)\,. \tag{2.5}$$

2.2 Structure function fits

Several structure function fits have been presented in the literature [7, 10, 14]. They are all based on the idea of assuming a more or less simple functional from for the structure function F_2 (say), and then fitting its parameters. Once the free parameters of a given functional form are determined by fitting to the data, one can compute any function of

F_2, such as e.g. its Mellin moments. In principle, given the error matrix of the parameters of the fit, one can also determine errors on F_2 and functions thereof by error propagation. Clearly, however, this is both impractical and subject to errors which are difficult to assess, because of the numerous linearizations which may be needed in order to determine the error on a physical observable (such as a Mellin moment) and, more importantly, because of the bias which is imposed by the choice of a specific functional form.

In order to overcome at least the practical difficulty of determining errors by propagation from the covariance matrix, a frequently adopted shortcut consists of giving a band of acceptable results, i.e. a pair of 'upper' and 'lower' values of the structure function, for each (x, Q^2). This error band can be again calculated either by means of the covariance matrix itself [7, 14], or by some other recipe which involves assessing the compatibility of a certain result with the data [5]. Whereas this may be sufficient in order to estimate the size of the error related to uncertainties on structure functions, it is clearly inadequate for a quantitative analysis, since there is no way to use these error bands to uniquely determine the error on a given quantity which depends on the structure function. For example, suppose one wants to determine the error on the integral over x of the structure function: one might be tempted to identify this error with the spread of values of the integral computed from the upper and lower curves. This is however not correct, because it neglects the correlation between the structure function at different values of x, and it can in fact lead to a substantial overestimate of the error.

These difficulties can be traced to the fact that when fitting a structure function one is trying to determine a function, i.e. an infinite-dimensional quantity, from a finite set of measurements [3]. Assuming a functional form of the function projects the problem onto the finite-dimensional subspace of functions of the given form, but at the cost of introducing a bias when choosing how to perform this projection. A full, unbiased solution of the problem requires instead the determination of the probability measure $\mathcal{P}[F_2]$ in the space of functions $F_2(x, Q^2)$, such that the expectation value of any observable $\mathcal{F}\left[F_2(x, Q^2)\right]$ which depends on F_2 can be found by averaging with this measure:

$$\left\langle \mathcal{F}\left[F_2(x, Q^2)\right] \right\rangle = \int \mathcal{D}F_2 \mathcal{F}\left[F_2(x, Q^2)\right] \mathcal{P}[F_2] , \tag{2.6}$$

where the integral is a functional integral over the space of functions F_2 [3]. For example, $\mathcal{F}\left[F_2(x, Q^2)\right]$ could be a Mellin moment of $F_2(x, Q^2)$.

A practical way of determining this probability measure, in the context of determining parton distributions, has been suggested in ref. [4]. The measure is determined by means of a Monte Carlo approach coupled to Bayesian inference: one builds a Monte Carlo sample of the space of functions, starting from an essentially arbitrary assumption on the form of the measure, and then one applies Bayesian inference in order to update this measure based on the available data.

Here, we suggest a different strategy to attain the same goal, in the related context of fitting structure functions: we use the available data to generate a Monte Carlo sample of the functional measure at a discrete set of points, and then use neural networks to interpolate between these points, thus generating the desired representation of the proba-

JHEP05(2002)062

bility measure in the space of functions. After introducing in the next section the general features of neural networks, we will give a detailed explanation of our procedure in section 4.

3. Neural networks

Artificial neural networks provide unbiased robust universal approximants to incomplete or noisy data. Applications of artificial neural networks range from pattern recognition to the prediction of financial markets. In particular, artificial neural networks are now a well established technique in high energy physics, where they are used for event reconstruction in particle detectors [15]. Here we will give a brief introduction to the specific type of artificial neural networks (henceforth simply neural networks) which we shall use for our analysis of structure functions. In particular, we will describe multilayer feed-forward neural networks (or perceptrons), and the algorithm used to train them (back-propagation learning).

3.1 Multilayer neural networks

An artificial neural network consists of a set of interconnected units (neurons). The state or activation of a given i-neuron, ξ_i, is a real number, determined as a function of the activation of the neurons connected to it. Each pair of neurons (i, j) is connected by a synapsis, characterized by a real number ω_{ij} (weight). Note that the weights need not be symmetric. The activation of each neuron is a function g of the difference between a weighted average of input from other neurons and a threshold θ_i:

$$\xi_i = g\left(\sum_j \omega_{ij}\xi_j - \theta_i\right) . \tag{3.1}$$

The activation function g is in general non-linear. The simplest example of activation function $g(x)$ is the step function $g(x) = \Theta(x)$, which produces binary activation only. However, it turns out to be advantageous to use an activation function with two distinct regimes, linear and non-linear, such as the sigmoid

$$g(x) \equiv \frac{1}{1+e^{-\beta x}} . \tag{3.2}$$

This function approaches the step function at large β; without loss of generality we will take $\beta = 1$. The sigmoid activation function has a linear response when $x \approx 0$, and it saturates for large positive or negative arguments. If weights and thresholds are such that the sigmoids work on the crossover between linear and saturation regimes, the neural network behaves in a non-linear way. Thanks to this non-linear behaviour, the neural network is able to reproduce nontrivial functions.

We will in particular consider Rosenblatt's perceptrons, also known as multilayer feed-forward neural networks [16]. These networks are organized in ordered layers whose neurons only receive input from a previous layer. For L layers with $n_1, \ldots, n_L$ units respectively,

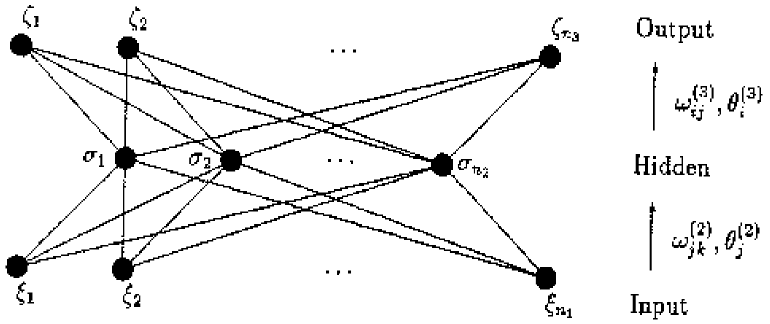

Figure 2: A three-layer feed-forward neural network consisting of input, hidden and output layers.

the state of the multilayer neural network is established by the recursive relations

$$\xi_i^{(l)} = g\left(\sum_{j=1}^{n_{l-1}} \omega_{ij}^{(l-1)} \xi_j^{(l-1)} - \theta_i^{(l)}\right); \qquad i = 1, \dots, n_l, \qquad l = 2, \dots, L, \tag{3.3}$$

where $\boldsymbol{\xi}^{(l)}$ represents the state of the neurons in the l^{th} layer, $\omega_{ij}^{(l)}$ the weights between units in the $(l-1)^{th}$ and the l^{th} layers, and $\theta_i^{(l)}$ the threshold of the i^{th} unit in the l^{th} layer. The input is the vector $\boldsymbol{\xi}^{(1)}$ and the output the vector $\boldsymbol{\xi}^{(L)}$.

Multilayer feed-forward neural networks can be viewed as functions F from $\mathbb{R}^{n_1} \to \mathbb{R}^{n_L}$, parametrized by weights, thresholds and activation function,

$$\boldsymbol{\xi}^{(L)} = F\left[\boldsymbol{\xi}^{(1)}; \{w_{ij}^{(l)}\}, \{\theta_i^{(l)}\}; g\right]. \tag{3.4}$$

For given activation function, the parameters can be tuned in such a way that the neural network reproduces any continuous function. The behaviour of a neural network is determined by the joint behaviour of all its connections and thresholds, and it can thus be built to be redundant, in the sense that modifying, adding or removing a neuron has little impact on the final output. Because of these reasons, neural networks can be considered to be robust, unbiased universal approximants.

3.2 Learning process for neural networks

The usefulness of neural networks is due to the availability of a training algorithm. This algorithm allows one to select the values of weights and thresholds such that the neural network reproduces a given set of input-output data (or patterns). This procedure is called 'learning' since, unlike a standard fitting procedure, there is no need to know in advance the underlying rule which describes the data. Rather, the neural network generalizes the examples used to train it.

The learning algorithm which we shall use is known as supervised training by back-propagation. Consider a set of input-output patterns $(\mathbf{x}, \mathbf{z})$ which we want the neural network to learn. The state of the neural network is generically given by

$$\xi_i^{(l)} = g(h_i^{(l)}), \qquad h_i^{(l)} = \sum_{j=1}^{n_{l-1}} \omega_{ij}^{(l-1)} \xi_j^{(l-1)} - \theta_i^{(l)}; \qquad i = 1, \dots, n_l, \quad l = 1, \dots, L. \tag{3.5}$$

Input-output patterns correspond to pairs of states of the first and last layers, which we shall denote as

$$\begin{aligned} \mathbf{x} &= \boldsymbol{\xi}^{(1)} , \\ \mathbf{o}(\mathbf{x}) &= \boldsymbol{\xi}^{(L)} . \end{aligned} \tag{3.6}$$

The goal of the training is to learn from a given set of data patterns, which consist of associations of a given input with a desired output. For any given values of the weights and thresholds, define the error function

$$E[\omega,\theta] \equiv \frac{1}{2}\sum_{A=1}^{n_p}\sum_{i=1}^{n_L}\left(o_i\left(\mathbf{x}^A\right)-z_i^A\right)^2 , \tag{3.7}$$

where n_p is the number of data, i.e., the number of input-output patterns. The error eq. (3.7) is the quadratic deviation between the actual and the desired output of the network, measured over the training set.

Given a fixed activation function g, eq. (3.4), the error function is a function of the weights and thresholds. It can be minimized by looking for the direction of steepest descent in the space of weights and thresholds, and modifying the parameters in that direction:

$$\begin{aligned} \delta\omega_{ij}^{(l)} &= -\eta\frac{\partial E}{\partial\omega_{ij}^{(l)}} , \\ \delta\theta_i^{(l)} &= -\eta\frac{\partial E}{\partial\theta_i^{(l)}}, \end{aligned} \tag{3.8}$$

where η fixes the rate of descent, i.e. the 'learning rate'.

The nontrivial result on which training is based is that it can be shown that the steepest descent direction is given by the following recursive expression:

$$\begin{aligned} \frac{\partial E}{\partial\omega_{ij}^{(l)}} &= \sum_{A=1}^{n_p}\Delta_i^{(l)A}\xi_j^{(l-1)A} ; \qquad i=1,\ldots,n_l , \qquad j=1,\ldots,n_{l-1} , \\ \frac{\partial E}{\partial\theta_i^{(l)}} &= -\sum_{A=1}^{n_p}\Delta_i^{(l)A} , \qquad i=1,\ldots,n_l , \end{aligned} \tag{3.9}$$

where the information on the goodness of fit is fed to the last layer through

$$\Delta_i^{(L)A} = g'\left(h_i^{(L)A}\right)\left[o_i\left(\mathbf{x}^A\right)-z_i^A\right] \tag{3.10}$$

and then it is back-propagated to the rest of the network by

$$\Delta_j^{(l-1)A} = g'\left(h_j^{(l-1)A}\right)\sum_{i=1}^{n_l}\Delta_i^{(l)A}\omega_{ij}^{(l)} . \tag{3.11}$$

The back-propagation algorithm then consists of the following steps:

1. initialize all the weights and thresholds randomly, and choose a small value for the learning rate η;

2. run a pattern $\mathbf{x}^A$ of the training set and store the activations of all the units;

3. calculate $\Delta_i^{(L)A}$ eq. (3.10), and then back-propagate the error using eq. (3.11);

4. compute the contributions to $\delta\omega_{ij}^{(l)}$ and $\delta\theta_i^{(l)}$ substituting eq. (3.9) in eq. (3.8);

5. update weights and thresholds.

The procedure is iterated until a suitable convergence criterion is fulfilled: for instance, if the value of the error function eq. (3.7) stops decreasing. The back-propagation can be performed in "batched" or "on-line" modes. In the former case, weights and thresholds are only updated after all patterns have been presented to the network, as indicated in eq. (3.9). In the latter case, the update is performed each time a new pattern is processed. Generally, on-line processing leads to faster learning, is less prone to getting trapped into local minima, and it is thus preferable if the number of data is reasonably small (i.e. $\lesssim 10^3$).

It is clear that back-propagation seeks minima of the error function given in eq. (3.7), but it cannot ensure that the global minimum is found. Several modifications have been proposed to improve the algorithm so that local minima are avoided. One of the most successful, simple and commonly used variants is the introduction of a 'momentum' term. This means that eq. (3.8) is replaced by

$$\begin{aligned} \delta\omega_{ij}^{(l)} &= -\eta \frac{\partial E}{\partial \omega_{ij}^{(l)}} + \alpha\ \delta\omega_{ij}^{(l)}(\text{last}) \\ \delta\theta_i^{(l)} &= -\eta \frac{\partial E}{\partial \theta_i^{(l)}} + \alpha\ \delta\theta_i^{(l)}(\text{last}) \end{aligned} \tag{3.12}$$

where "last" denotes the values of the $\delta\omega_{ij}^{(l)}$ and $\delta\theta_i^{(l)}$ used in the previous updating of the weights and thresholds. The parameter α ('momentum') must be a positive number smaller than 1.

Also, it may be convenient to choose a more general form of the error function, such as

$$E[\omega,\theta] \equiv \sum_{A,B=1}^{n_p} \sum_i^{n_L} \left(o_i\left(\mathbf{x}^A\right) - z_i^A\right) V_{AB} \left(o_i\left(\mathbf{x}^B\right) - z_i^B\right), \tag{3.13}$$

where V_{AB} is a $n_p \times n_p$ matrix which can accommodate weights and correlations between data, such as, typically, the inverse of the covariance matrix. If the matrix V_{AB} is diagonal, the back-propagation algorithm is essentially unchanged, up to a rescaling of $\Delta_i^{(L)A}$ eq. (3.10) by the corresponding matrix element V_{AA}. However, if the matrix V_{AB} is not diagonal (i.e. the data are correlated), eq. (3.10) must be substituted by a non-local expression which involves the sum over all correlated data. Clearly, in this case batched training is preferable, so that all correlated patterns are shown to the net before the weights and thresholds are updated.

JHEP05(2002)062

4. Neural structure functions

4.1 General strategy

We now describe in detail the strategy which we shall follow in order to construct a neural parametrization of structure functions (see figure 3). As discussed in section 2.2, the basic idea of our approach consists of constructing a representation of the probability measure in the space of structure functions. We do this in two steps: first, we generate a sampling of this measure based on the available data, and then we interpolate between these points using neural networks. The final set of neural networks is the sought-for representation of the probability measure.

The first step consists of generating a Monte Carlo set of 'pseudo-data', i.e. $N_{\rm rep}$ replicas of the original set of $N_{\rm dat}$ data points:

$$F_i^{(\rm art)(k)}\,; \qquad k=1,\ldots,N_{\rm rep}\,, \qquad i=1,\ldots,N_{\rm dat}\,, \tag{4.1}$$

where the subscript i is a shorthand for (x_i, Q_i^2). The $N_{\rm rep}$ sets of $N_{\rm dat}$ points are distributed according to an $N_{\rm dat}$-dimensional multigaussian distribution about the original points, with expectation values equal to the central experimental values, and error and covariance equal to the corresponding experimental quantities. Because [17, 18] the distribution of the experimental data coincides (for a flat prior) with the probability distribution of the value of the structure function F_2 at the points where it has been measured, this Monte Carlo set gives a sampling of the probability measure at those points. We can then generate arbitrarily many sets of pseudo-data: the number of sets $N_{\rm rep}$ can be chosen to be large enough that all relevant properties of the probability distribution of the data (such as errors and correlations) are correctly reproduced by the given sample. We will perform a number of tests in order to determine the optimal value of $N_{\rm rep}$, which will be discussed in section 4.2.

The second step consists of training $N_{\rm rep}$ neural networks, one for each set of $N_{\rm dat}$ pseudo-data. The key issue here is the choice of a suitable error function eq. (3.13). A possibility would be to use the simplest unweighted form eq. (3.7):

$$E^{(k)}[\omega,\theta] = \sum_{i=1}^{N_{\rm dat}} \left(F_i^{(\rm art)(k)} - F_i^{(\rm net)(k)}\right)^2\,, \tag{4.2}$$

where $F_i^{(\rm net)(k)}$ is the output of the neural network for the input values of (x, Q^2) corresponding to the i^{th} data point. Then, with suitable choices of architecture and training, we can achieve a value of the error such that $E/N_{\rm dat}$ is much smaller than the typical experimental uncertainty on each point. In other words, the trained net will then go on top of each point, on the scale of experimental errors. Therefore, it will reproduce the sampling of the probability distribution at the measured points, and interpolate between them, thus achieving our goal. In fact, this happens even if $E/N_{\rm dat}$ is not significantly smaller than the experimental uncertainty, provided only that for each point i the shifts $\delta_i^{(k)} \equiv F_i^{(\rm net)(k)} - F_i^{(\rm art)(k)}$ are distributed as uncorrelated random variables with zero mean as k runs over replicas. Then, it is easy to see that the distribution of $F_i^{(\rm net)(k)}$ has the same average, variance, and point-to-point correlations as the distribution of $F_i^{(\rm art)(k)}$.

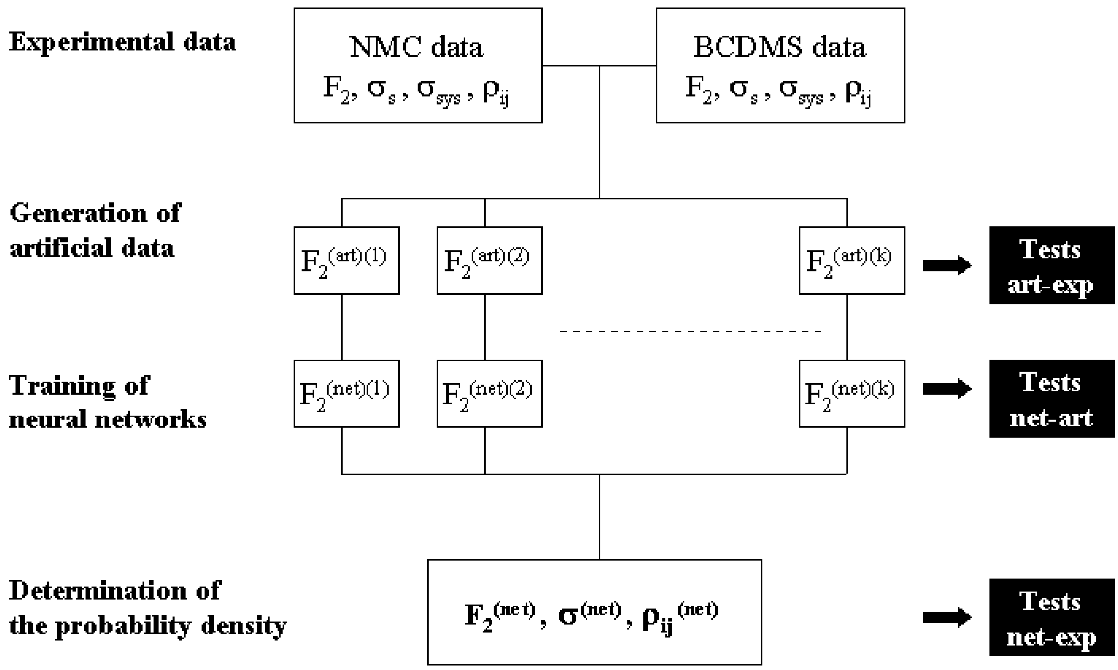

Figure 3: Flow chart for the construction of the parametrization of structure functions

This is however not an optimal solution in that it does not fully exploit the available physical information. To understand this, assume the value of the structure function at the same $(x,\, Q^2)$ point is measured by two different experiments. If the measurements are independent, they are uncorrelated. However, since they determine the same physical observable, they cannot be treated as independent, and they must in fact be combined to give a single determination, with an error that it is smaller than either of the two measurements. Now, it is clear that if the structure function is measured at two points which are very close in $(x,\, Q^2)$, then the value of the structure function at these points are likewise not independent: even though we do not know the functional form of the structure function, obvious QCD arguments imply that such a functional form exists, and it is continuous. Hence, by continuity, two neighbouring points cannot be treated as independent samplings of a probability distribution, and should be combined. This combined distribution has a variance which is in general smaller than its sampling at fixed points.

This reduction in variance is automatically achieved when fitting a given function to the data: if we fit a function to each set of pseudo-data, then as the pseudo-data span their multigaussian space, the best-fit values of the parameters which parametrize the function vary over their own space. As the number of data points is increased, the variance of these parameters decreases. For instance, assuming for the sake of argument a multilinear law with k parameters, with k data points all parameters can be determined with an error which can be found by error propagation, but with $N_{\rm dat} > k$ points this error is reduced by a factor $1/\sqrt{N_{\rm dat} - k}$. More in general, if the number of data exceeds the number of parameters, the error on the parameters will be reduced by a factor which is bounded by the so-called RCF inequality [18]; if there exists an estimate of the parameters which saturates this bound (as in the multilinear case), then it can be found by maximum-likelihood.

We are thus led to think that the optimal strategy consists of choosing as an error function the log-likelihood function for the given data. In this case, the matrix V_{AB} in eq. (3.13) is identified with the experimental covariance matrix, so the error function is

$$E^{(k)}[\omega,\theta] = \sum_{i,j=1}^{N_{\rm dat}} \left(F_i^{({\rm art})(k)} - F_i^{({\rm net})(k)}\right) {\rm cov}^{-1}_{ij} \left(F_j^{({\rm art})(k)} - F_j^{({\rm net})(k)}\right), \qquad (4.3)$$

where ${\rm cov}_{ij}$ is the covariance matrix defined in eq. (2.3). In such case, if we assume that after training the neural network can capture — at least locally — the underlying functional form, then a maximum-likelihood determination of its parameters leads to a distribution of neural networks with a smaller uncertainty than that based on the minimization of the naive error function eq. (4.2).

In practice, however, this choice turns out not to be viable. Indeed, as discussed in section 3.2, the back-propagation algorithm with the error function eq. (4.3) is non-local, and has poor convergence properties, especially when the number of data is large, as in our case. Note that this difficulty is related to the number of correlated data points, and thus it persists even if the number of correlated systematics is actually small, despite the fact that in such case the inversion of the correlation matrix is not a problem [5].

A strategy that is both physically effective and numerically viable consists instead of choosing as an error function the log-likelihood calculated from uncorrelated statistical errors, namely

$$E^{(k)}[\omega,\theta] = \sum_{i=1}^{N_{\rm dat}} \frac{\left(F_i^{({\rm art})(k)} - F_i^{({\rm net})(k)}\right)^2}{\sigma_{i,s}^{({\rm exp})\,2}}, \qquad (4.4)$$

where $\sigma_{i,s}^{({\rm exp})}$ is the statistical error on the i^{th} data point. Because the error function is now diagonal, use of the back-propagation algorithm is no longer problematic. In order to understand the effect of minimizing the error function eq. (4.4), write the probability distribution of the i^{th} data point as k runs over replicas as

$$F_i^{({\rm art})(k)} = F_i^{({\rm exp})} + \sum_{p=1}^{N_{sys}} r_{i,p}^{(k)} \sigma_{i,p} + r_{i,s}^{(k)} \sigma_{i,s}\,, \qquad (4.5)$$

where $F_i^{({\rm exp})}$ is the experimental value, all $r^{(k)}$ are distributed in k as univariate gaussian random numbers with zero mean, σ_s is the statistical error, and σ_p are the N_{sys} systematics. Whereas there are $N_{\rm dat}$ independent $r_{i,s}$, the number of independent variables $r_{i,p}$ for each p is smaller, in that there is one single random variable $r_{i,p}$ for all correlated data points. The maximum-likelihood fit of the error function eq. (4.4) to a set of data distributed according to eq. (4.5) determines the values of $F_i^{(\mathrm{net})}$ which best fit the distribution of

$$F_i^{(\mathrm{sys})(k)} \equiv F_i^{(\mathrm{exp})} + \sum_{p=1}^{N_{sys}} r_{i,p}^{(k)} \sigma_{i,p} \,. \tag{4.6}$$

In this case, the distribution of best-fit functional neural networks as k runs over replicas will have the smaller uncertainty which is obtained by maximum-likelihood combination of the statistical errors. However, it will still reproduce the (unimproved) systematic errors as k runs over the Monte Carlo sample. The distribution of systematic errors will remain unbiased, provided only that for each point i the distribution of best-fit values of $F_i^{(\mathrm{net})(k)}$ about $F_i^{(\mathrm{art})(k)}$ as k runs over replicas is such that $F_i^{(\mathrm{net})(k)}$ is uncorrelated to $r_{i,p}^{(k)}$. In other words, the best-fit nets provides a probability distribution which optimally combines statistical errors, but reproduces the systematic errors of the Monte Carlo sample.

Just as in the first step, also in this second step of the procedure it will be necessary to verify whether the number of replicas is large enough that the statistical properties of the distributions are properly reproduced. Furthermore, it will be necessary to verify that the network fitting procedure did not introduce any bias, and specifically that if the statistical variance is reduced this is due to having combined data points, and not to a bias of the fitting procedure. To this purpose, we will develop a number of statistical tools which will be discussed in section 5.1 when presenting our results.

At the end of our procedure, we obtain a set of N_{rep} trained neural networks which provides us with a representation of the probability measure in the space of structure functions, and thus it allows us to determine any observable by averaging with this measure, eq. (2.6). In particular, estimators for expectation values, errors and correlations are

$$\left\langle F_i^{(\mathrm{net})} \right\rangle_{\mathrm{rep}} = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} F_i^{(\mathrm{net})(k)} \tag{4.7}$$

$$\sigma_i^{(\mathrm{net})} = \sqrt{\left\langle \left(F_i^{(\mathrm{net})} \right)^2 \right\rangle_{\mathrm{rep}} - \left\langle F_i^{(\mathrm{net})} \right\rangle_{\mathrm{rep}}^2} \tag{4.8}$$

$$\rho_{ij}^{(\mathrm{net})} = \frac{\left\langle F_i^{(\mathrm{net})} F_j^{(\mathrm{net})} \right\rangle_{\mathrm{rep}} - \left\langle F_i^{(\mathrm{net})} \right\rangle_{\mathrm{rep}} \left\langle F_j^{(\mathrm{net})} \right\rangle_{\mathrm{rep}}}{\sigma_i^{(\mathrm{net})} \sigma_j^{(\mathrm{net})}} \,, \tag{4.9}$$

where i, j denote two pairs of values (x_i, Q_i^2), (x_j, Q_j^2) (not necessarily in the original data set). More in general, any functional average, defined by eq. (2.6), can be estimated by

$$\left\langle \mathcal{F}\left[F_2(x,Q^2) \right] \right\rangle = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \mathcal{F}\left[F_2{}^{(\mathrm{net})(k)}(x,Q^2) \right] . \tag{4.10}$$

4.2 Generation of artificial data

The Monte Carlo set of pseudo-data can be generated by using equations of the form of (4.5). In particular, for the NMC experiment we have

$$
\begin{aligned}
F_i^{(\mathrm{art})\,(k)} &= \left(1 + r_7^{(k)}\,\sigma_N\right) \times \\
&\times \left[F_i^{(\mathrm{exp})} + \frac{r_{i,1}^{(k)}\,E_i + r_{i,2}^{(k)}\,E_i' + r_{i,3}^{(k)}\,AC_i + r_{i,4}^{(k)}\,RC_i + r_{i,5}^{(k)}\,RE_i}{100} F_i^{(\mathrm{exp})} + r_{i,s}^{(k)}\,\sigma_{i,s} \right],
\end{aligned} \tag{4.11}
$$

where each $r_{i,p}^{(k)}$ is distributed as an univariate gaussian random number with zero mean over the replica sample (i.e., as k varies), $F_i^{(\mathrm{exp})}$ is the central experimental value, and the various sources of systematics were discussed in section 2.1.1. Since statistical errors are uncorrelated, there is an independent $r_{i,s}$ for each data point. Correlated systematics are correctly reproduced by taking the same r_3 and r_5 for all data corresponding to all energies and targets, the same r_4 for all targets, and the same r_1, r_2 and r_7 for all targets.

Likewise, for BCDMS we have

$$
\begin{aligned}
F_i^{(\mathrm{art})\,(k)} &= \left(1 + r_5^{(k)}\,\sigma_N\right) \sqrt{1 + r_{i,6}^{(k)}\,\sigma_{N_t}} \sqrt{1 + r_{i,7}^{(k)}\,\sigma_{N_b}} \times \\
&\times \left[F_i^{(\mathrm{exp})} + \frac{r_{i,1}^{(k)}\,f_b + r_{i,2}^{(k)}\,f_{i,s} + r_{i,3}^{(k)}\,f_{i,r}}{100} F_i^{(\mathrm{exp})} + r_{i,s}^{(k)}\,\sigma_s^i \right],
\end{aligned} \tag{4.12}
$$

where now r_1,r_2,r_3 and r_5 take the same value for all data (all energies and targets), r_6 is fixed for a given target, r_7 is fixed for given beam energy.

In order to determine a suitable value of N_{rep}, we now compare expectation values, variance and correlations of the pseudo-data set with the corresponding input experimental values, as a function of the number of replicas. A graphical comparison is shown in figure 4, where we display scatter plots of the central values, errors and correlation coefficients for samples of 10, 100 and 1000 replicas, for the proton, deuteron, and nonsinglet structure functions.

A more quantitative comparison can be performed defining the following quantities

- Average over the number of replicas for each experimental point i (i.e. a pair of values of x and Q^2)

$$
\left\langle F_i^{(\mathrm{art})} \right\rangle_{\mathrm{rep}} = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} F_i^{(\mathrm{art})(k)} . \tag{4.13}
$$

- Variance of $\langle F_i^{(\mathrm{art})} \rangle_{\mathrm{rep}}$

$$
\sigma_i^{(\mathrm{art})} = \sqrt{\left\langle \left(F_i^{(\mathrm{art})} \right)^2 \right\rangle_{\mathrm{rep}} - \left\langle F_i^{(\mathrm{art})} \right\rangle_{\mathrm{rep}}^2} . \tag{4.14}
$$

- Correlation and covariance of two points

$$
\rho_{ij}^{(\mathrm{art})} = \frac{\left\langle F_i^{(\mathrm{art})}\,F_j^{(\mathrm{art})} \right\rangle_{\mathrm{rep}} - \left\langle F_i^{(\mathrm{art})} \right\rangle_{\mathrm{rep}} \left\langle F_j^{(\mathrm{art})} \right\rangle_{\mathrm{rep}}}{\sigma_i^{(\mathrm{art})}\,\sigma_j^{(\mathrm{art})}} , \tag{4.15}
$$

$$
\mathrm{cov}_{ij}^{(\mathrm{art})} = \rho_{ij}^{(\mathrm{art})} \sigma_i^{(\mathrm{art})} \sigma_j^{(\mathrm{art})} . \tag{4.16}
$$

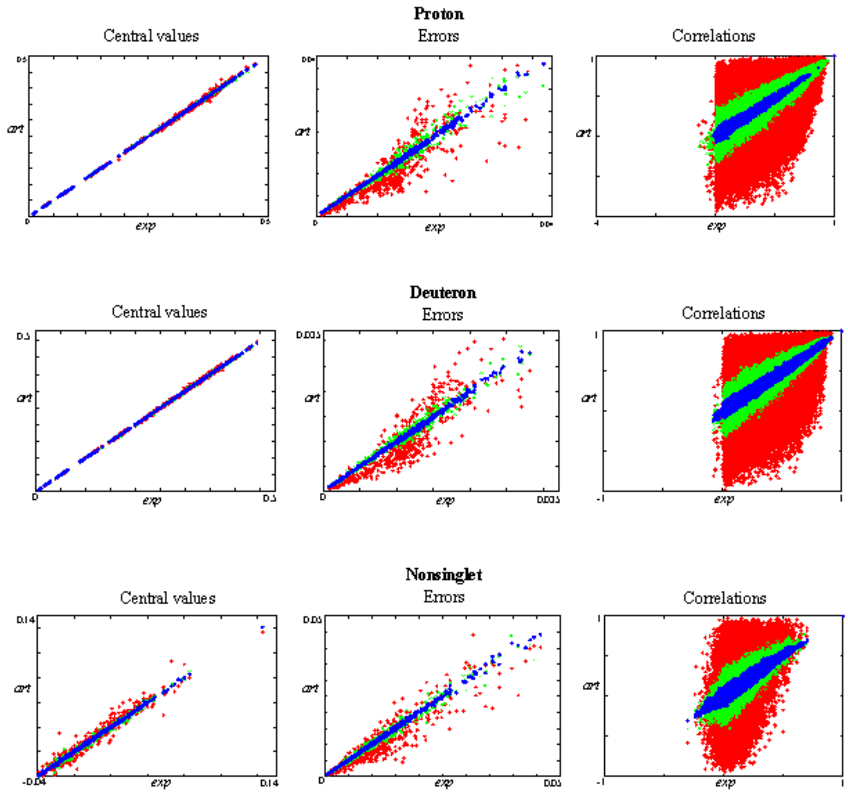

Figure 4: $\langle F_i^{(\mathrm{art})} \rangle_{\mathrm{rep}}$ vs. $\bar{F}_i^{(\mathrm{exp})}$, $\langle \sigma_i^{(\mathrm{art})} \rangle_{\mathrm{rep}}$ vs. $\bar{\sigma}_i^{(\mathrm{exp})}$ and $\langle \rho_{ij}^{(\mathrm{art})} \rangle_{\mathrm{rep}}$ vs. $\rho_{ij}^{(\mathrm{exp})}$ with $N_{\mathrm{rep}} = 10$ (red), 100 (green) and 1000 (blue) replicas.

These quantities provide the estimators of the experimental values, errors, and correlation which one extracts from the sample of artificial data.

- Mean variance and mean percentage error on central values over the number of points N_{dat}

$$\left\langle V\left[\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right]\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\left(\left\langle F_i^{(\mathrm{art})}\right\rangle_{\mathrm{rep}} - F_i^{(\mathrm{exp})}\right)^2 \tag{4.17}$$

$$\left\langle PE\left[\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right]\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\left|\frac{\left\langle F_i^{(\mathrm{art})}\right\rangle_{\mathrm{rep}} - F_i^{(\mathrm{exp})}}{F_i^{(\mathrm{exp})}}\right| . \tag{4.18}$$

We can similarly define

$$\left\langle V[\sigma^{(\mathrm{art})}]\right\rangle_{\mathrm{dat}}, \qquad \left\langle PE[\sigma^{(\mathrm{art})}]\right\rangle_{\mathrm{dat}}, \qquad \left\langle V[\rho^{(\mathrm{art})}]\right\rangle_{\mathrm{dat}} \quad \text{and} \quad \left\langle V[\mathrm{cov}^{(\mathrm{art})}]\right\rangle_{\mathrm{dat}} .$$

These estimators indicate how close the averages over generated data are to the experimental values.

- Scatter correlation:
$$r\left[F^{(\mathrm{art})}\right] = \frac{\left\langle F^{(\mathrm{exp})}\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}} - \left\langle F^{(\mathrm{exp})}\right\rangle_{\mathrm{dat}}\left\langle\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}}}{\sigma_s^{(\mathrm{exp})}\,\sigma_s^{(\mathrm{art})}}. \tag{4.19}$$
where the scatter variances are defined as
$$\begin{aligned}\sigma_s^{(\mathrm{exp})} &= \left\langle (F^{(\mathrm{exp})})^2\right\rangle_{\mathrm{dat}} - \left(\left\langle F^{(\mathrm{exp})}\right\rangle_{\mathrm{dat}}\right)^2 \\ \sigma_s^{(\mathrm{art})} &= \left\langle\left(\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right)^2\right\rangle_{\mathrm{dat}} - \left(\left\langle\left\langle F^{(\mathrm{art})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}}\right)^2.\end{aligned} \tag{4.20}$$
Similarly we define $r[\sigma^{(\mathrm{art})}]$, $r[\rho^{(\mathrm{art})}]$ and $r[\mathrm{cov}^{(\mathrm{art})}]$. The scatter correlation indicates the size of the spread of data around a straight line in the scatter plots of figure 4. Specifically, $r[\sigma^{(\mathrm{net})}] = 1$ implies that $\langle\sigma_i^{(\mathrm{net})}\rangle$ is proportional to $\sigma_i^{(\mathrm{exp})}$ (and similarly for $r[F^{(\mathrm{net})}]$ etc.). Note that the scatter correlation and scatter variance are not related to the variance and correlation eqs. (4.14)–(4.16) or averages thereof.

- Average variance:
$$\left\langle\sigma^{(\mathrm{art})}\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\sigma_i^{(\mathrm{art})}. \tag{4.21}$$
Similarly we can define $\langle\rho^{(\mathrm{art})}\rangle_{\mathrm{dat}}$ and $\langle\mathrm{cov}^{(\mathrm{art})}\rangle_{\mathrm{dat}}$. Analogously, we can define the corresponding experimental quantities
$$\left\langle\sigma^{(\mathrm{exp})}\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\sigma_i^{(\mathrm{exp})} \tag{4.22}$$
where $\sigma_i^{(\mathrm{exp})}$ is the experimental error on the i^{th} data point, an so forth. These quantities are interesting because even if the scatter correlation r eq. (4.19) is very close to one (so all points in the scatter plots figure 4 lie on a straight line), there could still be a systematic bias in the estimators eqs. (4.13)–(4.16). For example, if for all points i the variance $\sigma_i^{(\mathrm{art})} = \lambda\sigma_i^{(\mathrm{exp})}$, then $r[\sigma^{(\mathrm{art})}] = 1$, but $\langle\sigma^{(\mathrm{art})}\rangle_{\mathrm{dat}} = \lambda\langle\sigma^{(\mathrm{exp})}\rangle_{\mathrm{dat}}$.

The values of these quantities for samples of 10, 100 and 1000 replicas and for the three structure functions under consideration are shown in tables 1–3. From these tables, it can be seen that the scaling of the various quantities with N_{rep} follows the standard behaviour of gaussian Monte Carlo samples [18]. For instance, the variance on central values should scale as $1/N_{\mathrm{rep}}$, while the variance on the errors should scale as $1/\sqrt{N_{\mathrm{rep}}}$. This implies that a smaller sample is sufficient in order to achieve a given accuracy on means than it is required to reach the same accuracy on errors. Also, because $V[\rho^{(\mathrm{art})}] = [1-(\rho^{(\mathrm{exp})})^2]^2/N_{\mathrm{rep}}$, the estimated correlation fluctuates more for small values of $\rho^{(\mathrm{exp})}$, and thus the average correlation $\langle\rho^{(\mathrm{art})}\rangle_{\mathrm{dat}}$ tends to be larger than the corresponding experimental value $\langle\rho^{(\mathrm{exp})}\rangle_{\mathrm{dat}}$. Notice finally that a larger sample is necessary in order to achieve a given percentage error on the nonsinglet than on the proton and deuteron. This is a consequence of the fact that the signal-to-noise ratio is smaller for the nonsinglet: the proton and deuteron structure functions are roughly of the same size, so the absolute value of the nonsinglet structure function is typically by about a factor 10 smaller.

JHEP05(2002)062

F_2^p

$N_{\rm rep}$	10	100	1000
$\langle V\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	1.2×10^{-5}	1.4×10^{-6}	2.5×10^{-7}
$\langle PE\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	0.9%	0.3%	0.1%
$r[F^{(\rm art)}]$	0.99966	0.99996	0.99999
$\langle V[\sigma^{(\rm art)}]\rangle_{\rm dat}$	8.0×10^{-5}	2.4×10^{-5}	8.3×10^{-6}
$\langle PE[\sigma^{(\rm art)}]\rangle_{\rm dat}$	41%	12%	4%
$\langle\sigma^{(\rm art)}\rangle_{\rm dat}$	0.0114	0.0122	0.0122
$r[\sigma^{(\rm art)}]$	0.859	0.988	0.999
$\langle V[\rho^{(\rm art)}]\rangle_{\rm dat}$	0.0904	0.0072	0.0007
$\langle\rho^{(\rm art)}\rangle_{\rm dat}$	0.364	0.321	0.319
$r[\rho^{(\rm art)}]$	0.723	0.952	0.995
$\langle V[{\rm cov}^{(\rm art)}]\rangle_{\rm dat}$	5.4×10^{-9}	6.7×10^{-10}	5.6×10^{-11}
$\langle{\rm cov}^{(\rm art)}\rangle_{\rm dat}$	4.0×10^{-5}	4.0×10^{-5}	3.8×10^{-5}
$r[{\rm cov}^{(\rm art)}]$	0.529	0.891	0.987

Table 1: Comparison between experimental and generated artificial data for the proton structure function. The experimental data have $\langle\sigma^{(\rm exp)}\rangle_{\rm dat}=0.0123$, $\langle\rho^{(\rm exp)}\rangle_{\rm dat}=0.323$ and $\langle{\rm cov}^{(\rm exp)}\rangle_{\rm dat}=3.9\times10^{-5}$.

F_2^d

$N_{\rm rep}$	10	100	1000
$\langle V\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	9.2×10^{-6}	9.1×10^{-7}	9.7×10^{-8}
$\langle PE\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	1.1%	0.3%	0.1%
$r[F^{(\rm art)}]$	0.99976	0.99998	0.99999
$\langle V[\sigma^{(\rm art)}]\rangle_{\rm dat}$	5.3×10^{-5}	1.5×10^{-5}	4.0×10^{-6}
$\langle PE[\sigma^{(\rm art)}]\rangle_{\rm dat}$	39%	11%	3%
$\langle\sigma^{(\rm art)}\rangle_{\rm dat}$	0.0095	0.0102	0.0102
$r[\sigma^{(\rm art)}]$	0.857	0.990	0.999
$\langle V[\rho^{(\rm art)}]\rangle_{\rm dat}$	0.0923	0.0075	0.0007
$\langle\rho^{(\rm art)}\rangle_{\rm dat}$	0.374	0.310	0.310
$r[\rho^{(\rm art)}]$	0.641	0.934	0.993
$\langle V[{\rm cov}^{(\rm art)}]\rangle_{\rm dat}$	2.6×10^{-9}	2.5×10^{-10}	2.3×10^{-11}
$\langle{\rm cov}^{(\rm art)}\rangle_{\rm dat}$	3.3×10^{-5}	3.3×10^{-5}	3.2×10^{-5}
$r[{\rm cov}^{(\rm art)}]$	0.568	0.932	0.992

Table 2: Comparison between experimental and generated artificial data for the deuteron structure function. The experimental data have $\langle\sigma^{(\rm exp)}\rangle_{\rm dat}=0.0102$, $\langle\rho^{(\rm exp)}\rangle_{\rm dat}=0.313$ and $\langle{\rm cov}^{(\rm exp)}\rangle_{\rm dat}=3.3\times10^{-5}$.

Inspection of the tables shows that in order to reach average scatter correlations of 99% and percentage accuracies of a few percent on structure functions, errors and correlations a sample of about 1000 replicas is necessary. Notice in particular that with 1000 replicas the estimated correlation fluctuates about the true value, rather than systematically overshooting it.

$F_2^p - F_2^d$

$N_{\rm rep}$	10	100	1000
$\langle V\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	1.4×10^{-5}	1.8×10^{-6}	2.7×10^{-7}
$\langle PE\left[\langle F^{(\rm art)}\rangle_{\rm rep}\right]\rangle_{\rm dat}$	35%	11%	4%
$r[F^{(\rm art)}]$	0.980	0.998	0.999
$\langle V[\sigma^{(\rm art)}]\rangle_{\rm dat}$	6.6×10^{-5}	2.19×10^{-5}	7.8×10^{-6}
$\langle PE[\sigma^{(\rm art)}]\rangle_{\rm dat}$	35%	12%	4%
$\langle\sigma^{(\rm art)}\rangle_{\rm dat}$	0.0101	0.0114	0.0114
$r[\sigma^{(\rm art)}]$	0.927	0.991	0.999
$\langle V[\rho^{(\rm art)}]\rangle_{\rm dat}$	0.1133	0.0094	0.0010
$\langle\rho^{(\rm art)}\rangle_{\rm dat}$	0.1112	0.0990	0.0946
$r[\rho^{(\rm art)}]$	0.405	0.816	0.971
$\langle V[{\rm cov}^{(\rm art)}]\rangle_{\rm dat}$	5.1×10^{-9}	5.8×10^{-10}	6.0×10^{-11}
$\langle{\rm cov}^{(\rm art)}\rangle_{\rm dat}$	7.3×10^{-6}	9.0×10^{-6}	8.7×10^{-6}
$r[{\rm cov}^{(\rm art)}]$	0.346	0.791	0.972

Table 3: Comparison between experimental and generated artificial data for the nonsinglet structure function. The experimental data have $\langle\sigma^{(\rm exp)}\rangle_{\rm dat} = 0.0114$, $\langle\rho^{(\rm exp)}\rangle_{\rm dat} = 0.090$ and $\langle{\rm cov}^{(\rm exp)}\rangle_{\rm dat} = 8.4\times10^{-6}$.

4.3 Building and training neural networks

The choice of an optimal structure of the neural nets and of an optimal learning strategy cannot follow fixed rules and must be tailored to the specific problem. Here we summarize and motivate the specific choices which we made for our fits.

Number of layers. The optimal number of layers in a neural network depends on the complexity of the specific task it should perform. It can be proved [15] that any continuous function, no matter how complex, can be represented by a multilayer neural network with no more than three layers (input layer, a middle hidden layer, and output layer). However, in practice, using a single hidden layer may require a very large number of units in it. Thus, it can be more useful to have two hidden layers and a smaller number of neurons on each. We will only consider networks with two hidden layers.

Number of hidden units. The number of hidden units needed to approximate a given function $\mathcal{F}$ is related to the number of terms which are needed in an expansion of $\mathcal{F}$ over the basis of functions $g(x)$, $g(g(x))$, ... There exist several techniques to determine the optimal number of units. Here we have taken a pragmatic approach. First, we carry out all the learning with a small number of units. Then, we restart the whole learning adding new units one by one, until stability of the error function is reached. In this way, an optimal architecture is found, which is large enough to reproduce faithfully the training patterns, but small enough that training is fast. The final results which we will present are obtained with the architecture $(4-5-3-1)$. The reason why there are four input nodes will be discussed below.

Activation function. We have taken a sigmoid activation function eq. (3.2). The sensitivity of the neural network can be enhanced by substituting the sigmoid activation function in the last layer with the identity (linear response). This avoids saturation of the last layer neurons and leaves space for more sensitive responses. For this reason, we have adopted linear response in the last layer. With these choices, and the architecture discussed above, the behaviour of the network is determined by the values of 47 free parameters (38 weights and 9 thresholds).

Rescaling. If the activation ξ_i of input (output) node i is numerically large, in order for the activation function to be in the nonlinear response regime, the weights ω_{ij} must be very small. However, if the activation is very large, the shifts eq. (3.10) in the first stages of the learning process are also very large, thus leading to an erratic behaviour of the training process, unless the learning rate η eq. (3.9) is initially taken to be very small and then changed along the training. In order to avoid this, it is convenient to rescale both the input and the output data in such a way that that typical activations are $\xi_i \sim \mathcal{O}(1)$ for all i. In practice, we have rescaled all values of (x, Q^2) and the structure function F_2 in such a way that they span the range $0.1 - 0.9$. This ensures that activation functions are in the nonlinear regime when the absolute values of all the weights settle to values close to one.

Input. We have used as input variables x, Q^2, $\log x$ and $\log Q^2$. The choice of taking $\log x$ and $\log Q^2$ along with x and Q^2 is motivated by the expectation that these are the variables upon which F_2 naturally depends. Note that this is not a source of theoretical bias: if this expectation turns out to be incorrect and these variables are useless, after training the neural network will simply disregard them. If, however, the expectation is correct this choice will speed up the training. We have checked that neural networks trained just with x and Q^2 perform as well as the ones we use but need longer training times.

Theoretical assumptions. The only theoretical assumption on the shape of $F_2(x, Q^2)$ produced by the neural nets is that it satisfies the kinematic bound $F_2(1, Q^2) = 0$ for all Q^2. In general, a constraint can be enforced by adding it to the error function eq. (3.7), so that configurations that violate the constraint are unfavourably weighted. In our case, the constraint is local in x and Q^2 so its implementation is straightfoward: it can be enforced by including in the data set a number of artificial data points which satisfy the constraint, with suitably tuned error. More general cosntraints, such as the momentum sum rule in a parton fit, would require a term which makes the error function nonlocal in x and Q^2, and would thus make the training computationally more intensive.

We have thus added to the data set 10 artificial points at $x = 1$ with equally spaced values of Q^2 on a linear scale between 20 and 200 GeV^2 (which ensures that the bound is respected even when extrapolating beyond the data region). The choice of the error on these points is very delicate: if it is too small, the neural networks will spend a significant fraction of their training time in learning these points. In such case, the kinematic constraint $F_2(x = 1, Q^2) = 0$ is enforced with great accuracy, but at the expense of the fit to experimental data. We have thus taken the error on these points at $x = 1$ to be of the same order of the smallest experimental error, namely, 10^{-3} for F_2^p and F_2^d, and $\sqrt{2} \times 10^{-4}$ for $F_2^p - F_2^d$.

JHEP05(2002)062

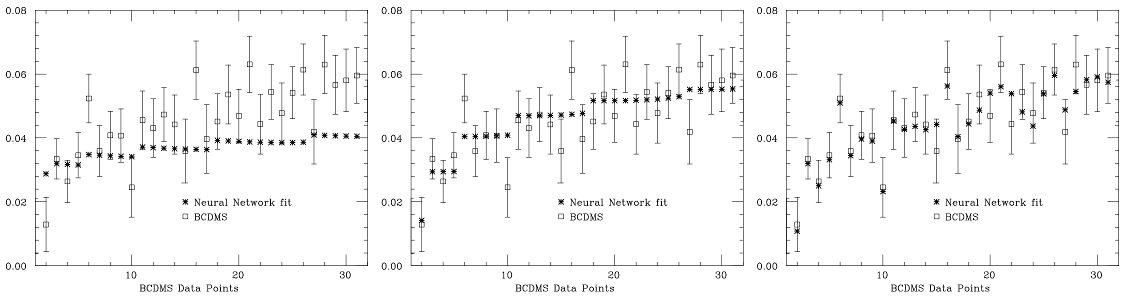

Figure 5: Fit of the nonsinglet structure function $F_2^p - F_2^d$ to a subset of BCDMS data points for increasing training lengths: insufficient training (left); ¡ normal training (middle); overlearning (right). The variable in abscissa is an arbitrary point number.

Training patterns. Since the number of data is reasonably small, we have adopted on-line training (see section 3.2): weights are updated after each pattern is shown to the net. If the experimental data were shown to the net in a fixed order, a bias could arise, e.g. because of the regularity of some input patterns. Therefore, we have shown the data to the network using each time a random permutation of their order. Furthermore, the data are always shown at least 10^4 times, in order to allow compensations of different variations. Because we use a pseudo-random number generator to compute permutations with periodicity 6×10^6 which is smaller than a typical number of training cycles (see section 5), there might still be periodic oscillations in behaviour of the best-fit network. However, we have checked that these oscillations do not significantly affect the fit.

Learning parameters. The choice of the learning rate η eq. (3.8) is particularly important: large values of η lead to an unstable training process, while small values lead to slow training which may get trapped in local minima. In practice, it is convenient to vary the value of η during the training: first, one looks for the region of the global minimum with a larger value of η. Once this region is located, the learning rate is reduced so that the probability of jumping away is small and it is then easier to deepen into the minimum. The momentum term α eq. (3.12) is closely connected to the learning rate: an increase of α implies an increase of the effective learning rate. The optimum value of α depends on the updating procedure used.

For our training, we have chosen to first minimize the error function eq. (4.2), and then switch to the minimization of eq. (4.4): first we look for the rough location of the minimum, and then we refine its search. The first set of training (of the order of 10^6 cycles) is performed with $\eta \sim 10^{-3}$, and the second (of the order of 10^8 cycles) with $\eta \sim 10^{-8}$ (see section 5). We will always take $\alpha = 0.9$.

Convergence of the training procedure and parametrization bias. As the training of the neural network progresses, the neural net reproduces in a more and more detailed way the features of the given data set. It follows that the problem of determining convergence of the training procedure is tangled with the problem of avoiding a possible fitting bias. Indeed, if the net is too small or the training is too short, it will artificially smooth out physical fluctuations in the data, whereas if the training is too long, the net will also fit statistical noise ('overlearning'). Convergence must be therefore assessed on the basis of a criterion which guarantees that the trained nets faithfully reproduce the local functional form of the data, without fitting statistical fluctuations, but also without imposing a smoothing bias. This issue is illustrated in figure 5, where we show the form of a neural network with the architecture and features discussed above, trained to a small, arbitrarily chosen subset of 30 BCDMS data points for the structure function $F_2^p - F_2^d$, with increasingly long training cycles. With this small data set, chosen for the sake of illustration, the architecture of the nets is very redundant, and overlearning is easily achieved. The first plot (very short training) shows that the neural network strongly correlates data points, but it does not quite reproduce the behavior of data. The second plot corresponds to a longer training, which leads to a more or less ideal fit. In the last plot (very long training) the neural network follows the statistical fluctuations of individual data points (overlearning). Note that the abscissa in this plot is an arbitrary point number; jumps in the best-fit function are not significant and simply reflect the fact that points of neighbouring number might correspond to rather different values of (x, Q^2).

Extreme overlearning as displayed in figure 5 can be avoided by using a number of parameters in the neural network which, while still being largely redundant so as to avoid parametrization bias, is significantly smaller than the number of points to be fitted. However, an accurate assessment of whether convergence has been reached can only be obtained by means of statistical indicators of the goodness of fit, tailored to the particular problem. These will be discussed in the next section. By means of such indicators, one may determine an optimal training length, and then subject all networks to training of this fixed length. This is more convenient than stopping the training on the basis of a convergence criterion, e.g. on the value of the error function, because this latter choice would tend to artificially damp statistical fluctuations of the pseudo-data sample, and it could thus lead to biased results.

5. Results

We will present results for the proton, deuteron and nonsinglet structure function F_2. We have decided to train a separate neural network for the nonsinglet in view of precision applications. This is because the proton and deuteron structure functions are roughly of the same size, while their difference is typically smaller by a factor 10. Hence, in order to achieve a 1% accuracy on $F_2^p - F_2^d$ we would need approximately a 0.1% accuracy on F_2^p and F_2^d. An individual fit of the proton, deuteron, and nonsinglet allows us to achieve a comparable accuracy on all structure functions. After discussing the general aspects of the assessment of the quality of the fits, we will discuss separately the nonsinglet, proton and deuteron fits.

5.1 Fit assessment

As discussed in section 4.1, the determination of the probability density in the space of structure function is based on the training of a neural network on each of the replicas of the original data set. We must therefore first, assess whether the training of each network is sufficient, and then verify that the set of networks represents the probability density of the data in a faithful and unbiased way.

The goodness of fit provided by each net is measured by the error function $E^{(k)}$ eq. (4.4) which is minimized by the training process; the average goodness of fit is thus

$$\langle E\rangle_{\mathrm{rep}} = \frac{1}{N_{\mathrm{rep}}}\sum_{k=1}^{N_{\mathrm{rep}}}\frac{1}{N_{\mathrm{dat}}}E^{(k)} \tag{5.1}$$

where we have further normalized to the number of data so that eq. (5.1) coincides with the mean square error per data point per replica. We also define the normalized error function computed for the original set of data (rather than one of the replicas)

$$E^{(0)} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\frac{\left(F_i^{(\mathrm{exp})} - F_i^{(\mathrm{net})(0)}\right)^2}{\sigma_{i,s}^{(\mathrm{exp})\,2}}, \tag{5.2}$$

where $F_i^{(\mathrm{net})(0)}$ is the prediction from a neural net trained on the set of central experimental values $F_i^{(\mathrm{exp})}$.

The estimator for the central value of each data point is given by the expectation value $\langle F_i^{(\mathrm{net})}\rangle_{\mathrm{rep}}$ eq. (4.7). The goodness of the fit to the original data provided by this estimator is

$$\chi^2 = \frac{1}{N_{\mathrm{dat}}}\sum_{i,j=1}^{N_{\mathrm{dat}}}\left(\left\langle F_i^{(\mathrm{net})}\right\rangle_{\mathrm{rep}} - F_i^{(\mathrm{exp})}\right)\mathrm{cov}_{ij}^{-1}\left(\left\langle F_j^{(\mathrm{net})}\right\rangle_{\mathrm{rep}} - F_j^{(\mathrm{exp})}\right), \tag{5.3}$$

where cov_{ij} is the covariance matrix eq. (2.3), and we have again normalized to the number of data points. Note that the number of degrees of freedom is not just the difference of N_{dat} and the number of parameters of the neural network, because the neural network is by construction redundant (see section 4.3): a network with a smaller number of parameters could lead to fits of the same quality, though with a longer training. However, the data points are about 500 (see section 2) while the parameters of the net are about 50 (see section 4.3), corresponding to some 10 or 20 truly independent degrees of freedom: hence, the expression eq. (5.3) differs only by a few percent from the χ^2 per degree of freedom.

In order to assess how well experimental errors are reproduced by their neural estimator eq. (4.8) we define the corresponding average variance and percentage errors

$$\left\langle V\left[\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right]\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\left(\left\langle F_i^{(\mathrm{net})}\right\rangle_{\mathrm{rep}} - F_i^{(\mathrm{exp})}\right)^2 \tag{5.4}$$

$$\left\langle PE\left[\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right]\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\left|\frac{\left\langle F_i^{(\mathrm{net})}\right\rangle_{\mathrm{rep}} - F_i^{(\mathrm{exp})}}{F_i^{(\mathrm{exp})}}\right|. \tag{5.5}$$

We can similarly define percentage errors on the correlation and covariance $\langle V[\rho^{(\mathrm{net})}]\rangle_{\mathrm{dat}}$ and $\langle V[\mathrm{cov}^{(\mathrm{net})}]\rangle_{\mathrm{dat}}$.

JHEP05(2002)062

A point-by-point measure of the agreement between network estimators and their experimental counterparts is provided by defining the scatter correlation of central values, in analogy to eq. (4.19)

$$r[F^{(\mathrm{net})}] = \frac{\left\langle F^{(\mathrm{exp})}\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}} - \left\langle F^{(\mathrm{exp})}\right\rangle_{\mathrm{dat}}\left\langle\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}}}{\sigma_s^{(\mathrm{exp})}\,\sigma_s^{(\mathrm{net})}}\,. \tag{5.6}$$

where the scatter variance of the network is

$$\sigma_s^{(\mathrm{net})} = \left\langle\left(\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right)^2\right\rangle_{\mathrm{dat}} - \left(\left\langle\left\langle F^{(\mathrm{net})}\right\rangle_{\mathrm{rep}}\right\rangle_{\mathrm{dat}}\right)^2\,. \tag{5.7}$$

Similarly we define $r[\sigma^{(\mathrm{net})}]$, $r[\rho^{(\mathrm{net})}]$ and $r[\mathrm{cov}^{(\mathrm{net})}]$. As discussed in section 4.2, the scatter correlation $-1 \le r \le 1$ indicates how closely the network estimator follows its experimental counterpart as the index i runs over experimental points. Specifically, $r[\sigma^{(\mathrm{net})}] \approx 1$ implies that $\langle\sigma_i^{(\mathrm{net})}\rangle$ is approximately proportional to $\sigma_i^{(\mathrm{exp})}$ (and similarly for $r[F^{(\mathrm{net})}]$ etc.). The coefficient of proportionality can then be found by comparing the average experimental variance eq. (4.22) with the average network variance

$$\left\langle\sigma^{(\mathrm{net})}\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}}\sum_{i=1}^{N_{\mathrm{dat}}}\sigma_i^{(\mathrm{net})}\,. \tag{5.8}$$

A substantial deviation of the scatter correlation from one indicates that the neural estimator is not following the pattern of experimental variations. This might indicate that the neural network is incapable of reproducing the shape of the data, or else that it is successfully reducing fluctuations in the data.

In practice, we are interested in the case in which the neural networks lead to a successful fit (i.e. with a good value of the χ^2 (5.3)), while leading to a reduction of the variance, as discussed in section 4.1, so that $\langle\sigma^{(\mathrm{net})}\rangle_{\mathrm{dat}} < \langle\sigma^{(\mathrm{exp})}\rangle_{\mathrm{dat}}$. Because we are minimizing the diagonal error function eq. (4.4) and not the full likelihood, we cannot simply rely on the value of the average error $\langle E\rangle$ and the χ^2 to assess the quality of the fit. Rather, we must verify that the distribution of systematic errors remains unbiased, and that if a reduction of the variance is observed it is not due to lack of flexibility of the neural network.

In order to test whether systematics are reproduced in an unbiased way, we notice that experimental correlations are entirely due to systematic errors, according to eqs. (2.1), (2.2). It follows from these equations and from the definition of covariance matrix (2.3) that if the networks reduce the statistical variance but reproduce the experimental systematics (as they ought to), then $\sigma_i^{(\mathrm{net})} < \sigma_i^{(\mathrm{exp})}$ and $\rho_{ij}^{(\mathrm{net})} > \rho_{ij}^{(\mathrm{exp})}$ but $\mathrm{cov}_{ij}^{(\mathrm{net})} \approx \mathrm{cov}_{ij}^{(\mathrm{exp})}$. So if $\mathrm{cov}^{(\mathrm{net})}$ and $\mathrm{cov}^{(\mathrm{exp})}$ are strongly correlated and have approximately the same mean, the systematics are reproduced in an unbiased way. Notice however that this condition is in general sufficient, but not necessary: for instance, more measurements at the same point from independent experiments have $\rho^{(\mathrm{exp})} = 0$ identically, while $\rho^{(\mathrm{net})} = 1$, so in this case $\mathrm{cov}^{(\mathrm{exp})}$ and $\mathrm{cov}^{(\mathrm{net})}$ cannot possibly be equal.

JHEP05(2002)062

In order to test whether the variance reduction is due to the fact that the networks are distributed with smaller variance around the true experimental values, we construct a suitable estimator, assuming for simplicity all errors to be gaussian. Consider a measurement value m_i of F_2, where i represents a pair of values (x, Q^2). For each measurement

$$m_i = t_i + \sigma_i\, s_i \tag{5.9}$$

where s_i is a univariate gaussian number with zero mean, t_i is the true value of F_2, and σ_i its error. The k^{th} replica of generated data is then

$$g_i^{(k)} = m_i + r_i^{(k)}\,\sigma_i = t_i + (s_i + r_i^{(k)})\sigma_i\,, \tag{5.10}$$

where r_i^k is also a univariate zero mean gaussian random number. Assume now that the best-fit neural networks are distributed about the true values t_i with an error $\hat{\sigma}_i$. For the k^{th} neural network we have

$$n_i^{(k)} = t_i + {r'}_i^{(k)}\,\hat{\sigma}_i\,, \tag{5.11}$$

where, assuming that the network is an unbiased estimator, ${r'}_i^{(k)}$ have zero mean, and can be normalized to be univariate. Because the networks are determined by fitting to $g_i^{(k)}$, in general, ${r'}_i^{(k)}$ will be correlated to $r_i^{(k)}$ and t_i.

If we assume that the number of replicas is large enough that $\langle r_i^{(k)}\rangle_{\mathrm{rep}} = \langle {r'}_i^{(k)}\rangle_{\mathrm{rep}} = 0$ and $\langle (r_i^{(k)})^2\rangle_{\mathrm{rep}} = \langle ({r'}_i^{(k)})^2\rangle_{\mathrm{rep}} = 1$, while the number of data points is large enough that $\langle s_i\rangle_{\mathrm{dat}} = 0$ and $\langle s_i^2\rangle_{\mathrm{dat}} = 1$ we get the average error eq. (5.1)

$$\langle E\rangle_{\mathrm{rep}} = 2 + \left\langle \left(\frac{\hat{\sigma}}{\sigma}\right)^2 \right\rangle_{\mathrm{dat}} - 2\left\langle \langle r r'\rangle_{\mathrm{rep}} \left(\frac{\hat{\sigma}}{\sigma}\right) \right\rangle_{\mathrm{dat}}\,, \tag{5.12}$$

where $\langle\rangle_{\mathrm{dat}}$ denotes averaging with respect to data in one replica (index i) and $\langle\rangle_{\mathrm{rep}}$ denotes averaging with respect to replicas (index k). Now, define a modified average error, where the prediction of the k-th network instead of being compared to the k-th replica (as in the average error proper eq. (5.1)) are compared to the experimental data:

$$\widetilde{E}^{(k)} = \sum_{i=1}^{N_{\mathrm{dat}}} \frac{\left(F_i^{(\mathrm{exp})(k)} - F_i^{(\mathrm{net})(k)}\right)^2}{{\sigma_{i,s}^{(\mathrm{exp})}}^2}\,; \tag{5.13}$$

$$\langle \widetilde{E}\rangle_{\mathrm{rep}} = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} \frac{1}{N_{\mathrm{dat}}} \widetilde{E}^{(k)}\,. \tag{5.14}$$

We get immediately

$$\langle \widetilde{E}\rangle_{\mathrm{rep}} = 1 + \left\langle \left(\frac{\hat{\sigma}}{\sigma}\right)^2 \right\rangle_{\mathrm{dat}}. \tag{5.15}$$

We finally define the ratio

$$\mathcal{R} \equiv \frac{\langle \widetilde{E}\rangle_{\mathrm{rep}}}{\langle E\rangle_{\mathrm{rep}}}, \tag{5.16}$$

which is the desired estimator. Indeed, assume that the networks displays significantly smaller fluctuations than the data, $\sigma_i^{(\mathrm{net})} \ll \sigma_i^{(\mathrm{exp})}$. We wish to test whether this is due to the fact that the network has performed substantial error reduction by combining several experimental data points $g_i^{(k)}$ into a determination of its parameters. In such case the correlation between ${r'}_i^{(k)}$ and $r_i^{(k)}$ is very weak, since ${r'}_i^{(k)}$ is determined by the values of several points $g_j^{(k)}$ with $j \neq i$. Hence

$$\mathcal{R} \approx \frac{1 + \langle (\hat{\sigma}/\sigma)^2 \rangle_{\rm dat}}{2 + \langle (\hat{\sigma}/\sigma)^2 \rangle_{\rm dat}} . \tag{5.17}$$

If error reduction is substantial and $\langle (\hat{\sigma}/\sigma^2) \rangle_{\rm dat} \ll 1$, then $\mathcal{R} \approx 1/2$. Physically, this means that by combining many data, the network manages to always be closer to the true values t_i than the replicas. If instead the network was just artificially smoothing the data, we would expect $(\hat{\sigma}_i/\sigma_i) > 1$ and again no correlation $\langle {r'}_i^k r_i^k \rangle_{\rm rep} \approx 0$, so $\mathcal{R} \gtrsim 1$.

It is instructive to consider also a case in which we observe no error reduction, $\sigma_i^{(\rm net)} \approx \sigma_i^{(\rm exp)}$. In this case, we can simply check that the neural network are behaving properly by verifying that the scatter correlation $r[\sigma]$ is high. However, the value of $\mathcal{R}$ provides a cross-check. In such case, we expect $\hat{\sigma}_i \approx \sigma_i$, and a large correlation $\langle {r'}_i^k r_i^k \rangle_{\rm rep} \approx 1$ since n_i^k is determined by g_i^k. Hence, in this case we would expect $\mathcal{R} \approx 2$. If instead the networks were not reproducing the data accurately we would again get $\mathcal{R} \lesssim 1$ as above.

5.2 Nonsinglet

The nonsinglet structure function is defined as

$$F_2^{NS}(x, Q^2) = F_2^p(x, Q^2) - F_2^d(x, Q^2) . \tag{5.18}$$

Data for this quantity are obtained by discarding the BCDMS proton data taken with a 100 GeV beam, which have no deuteron counterpart, and then exploiting the fact that all remaining BCDMS and NMC data come in proton, deuteron pairs, taken at the same values of x and Q^2. For NMC data, there are small difference in the binning in Q^2 between proton and deuteron, which are however negligible on the scale of the typical variation of the structure function and the size of the experimental errors.

In order to establish the length of the training, we study the behaviour of the average error function $E^{(0)}$ eq. (5.2) determined from the original set of data as a function of the number of training cycles (figure 6).[1] It is seen that an extremely long training (of order of 10^9 training cycles) is required in order for the error function to stabilize. However, if we look at the average error function computed from data of either of the two experiments, it is apparent that it is only the contribution to $E^{(0)}$ from the BCDMS data which keeps improving, while $E^{(0)}_{NMC}$ attains very rapidly a small value which is then stable. If we train the nets only to data of one experiment (figures 7, 8), the training required for reaching convergence with BCDMS is still rather longer than with NMC data, but substantially shorter than when both experiments are fitted at once. The greater difficulty in learning the BCDMS data is due to fact that these data have smaller errors. Notice that the net

[1] Cyclic patterns in the behaviour of $E^{(0)}$ as a function of the training length are due to the periodicity of the pseudo-random number generator which is used to determine the permutation of data in the training process, see section 4.3.

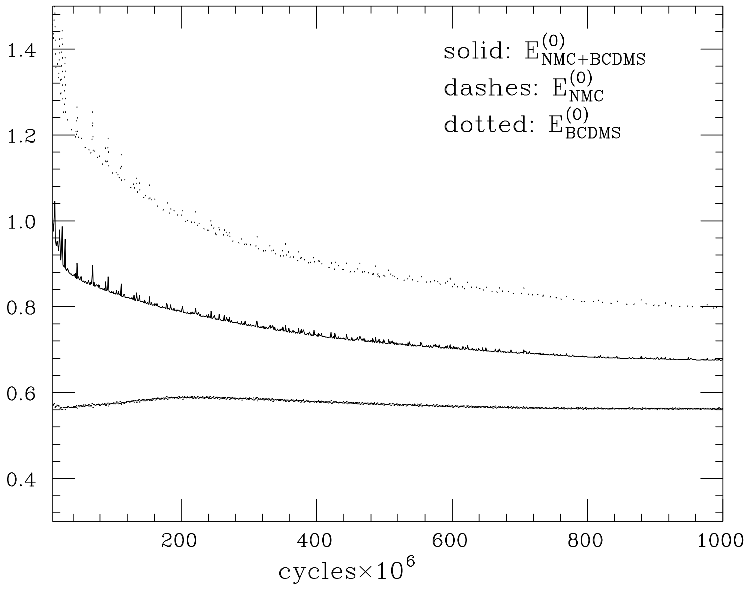

Figure 6: Nonsinglet: dependence of the error on the length of training.

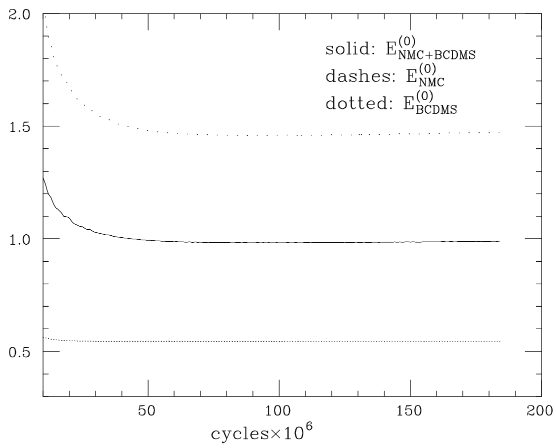

Figure 7: NS training on NMC data.

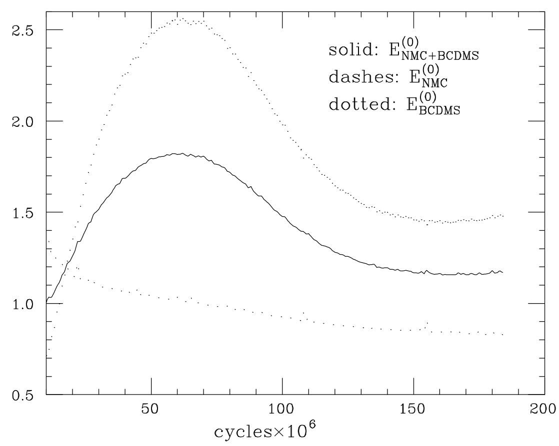

Figure 8: NS training on BCDMS data.

trained on each of the two experiments predicts to a large extent the other experiment once convergence has been achieved. This means that, even though only one experiment is used for training, the contributions to $E^{(0)}$ from both experiments are reduced in the process — though, of course, the experiment which is not used has eventually a worse average $E^{(0)}$.

This suggests that an optimized training can be obtained by showing to the network BCDMS data more often than NMC data. Then, the BCDMS data are learnt faster but the information from the NMC data is not lost. An optimal combination is found (by trial and error) when the BCDMS data are shown 90% of times, and NMC data 10% of times. Then, the average error $E^{(0)}$ for the two experiments both reach convergence at the same value after about 1.8×10^8 training cycles (figure 9). Note that, after convergence has been reached, the average error $E^{(0)}$ of the two experiment oscillate about each other, while slowly improving at the same rate. This training is optimized, in comparison to

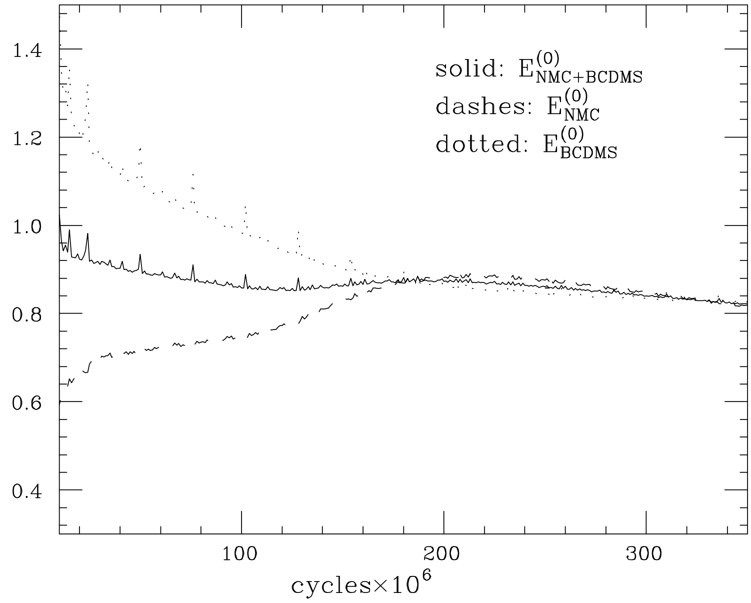

Figure 9: Nonsinglet weighted training (90% BCDMS, 10% NMC)

that of figure 6 both because it is by more than a factor 5 shorter, and, more importantly, because the convergence of the values of $E^{(0)}$ for the two experiments shows that the global minimum has been found.

Indeed, a substantial difference in values of $E^{(0)}$ for the two data sets is not acceptable, because it entails that if on average $E^{(0)} \approx 1$, in actual fact the net is overlearning one of the two data sets, and underlearning the other, i.e., overlearning in one kinematic region and underlearning in another region. This is what happens with $\sim 10^7$ training cycles (origin of the plot of figure 6). Unless there are reasons to believe that errors are not correctly estimated for one of the data sets, this means that the neural network is getting trapped in a local minimum corresponding to the data set which has been learnt faster, and thus unable to find the global minimum. Indeed, when the global minimum is found, the values of $E^{(0)}$ for the two data sets are approximately equal and improve at the same rate, as it happens for the weighted training of figure 9 with $\gtrsim 2 \times 10^8$ training cycles. It is clear from figure 6 that if an unweighted training is adopted, it might still be possible to achieve comparable values of $E^{(0)}$ for the two data sets, but only with a very long training, leading to a value of $E^{(0)} << 1$. In such case, the net would be overlearning throughout the full data region. Hence, weighted training is mandatory if we wish to avoid both local minima and overlearning.

As discussed in section 4.3, it turns out to be convenient to let each network first undergo a preliminary short training to the simple error function eq. (4.2) with large learning rate η eq. (3.9). After some experimentation along the lines discussed above, the training parameters have been chosen to be as follows: 4.6×10^6 cycles with learning rate $\eta = 4 \times 10^{-3}$ and error function eq. (4.2), followed by 1.8×10^8 cycles with $\eta = 4 \times 10^{-8}$ 90% of the times on BCDMS data and 10% of the times on NMC data with error function eq. (4.4). The momentum term eq. (3.12) is always set to $\alpha = 0.9$.

JHEP05(2002)062

N_{net}	NMC+BCDMS	NMC	BCDMS
χ^2	0.79	0.78	0.80
$\langle E\rangle_{\rm rep}$	1.14	1.05	1.25
$\mathcal{R}$	0.58	0.53	0.66
$r\left[F^{(\rm net)}\right]$	0.81	0.69	0.95
$\langle V[\sigma^{(\rm net)}]\rangle_{\rm dat}$	1.7×10^{-4}	2.9×10^{-4}	3.5×10^{-5}
$\langle PE[\sigma^{(\rm net)}]\rangle_{\rm dat}$	80%	82%	77%
$\langle\sigma^{(\rm exp)}\rangle_{\rm dat}$	0.011	0.016	0.006
$\langle\sigma^{(\rm net)}\rangle_{\rm dat}$	0.004	0.005	0.003
$r[\sigma^{(\rm net)}]$	0.50	-0.02	0.92
$\langle V[\rho^{(\rm net)}]\rangle_{\rm dat}$	0.34	0.33	0.35
$\langle\rho^{(\rm exp)}\rangle_{\rm dat}$	0.09	0.04	0.16
$\langle\rho^{(\rm net)}\rangle_{\rm dat}$	0.57	0.45	0.73
$r[\rho^{(\rm net)}]$	0.37	0.12	0.56
$\langle V[{\rm cov}^{(\rm net)}]\rangle_{\rm dat}$	9.7×10^{-10}	1.7×10^{-9}	2.1×10^{-11}
$\langle{\rm cov}^{(\rm exp)}\rangle_{\rm dat}$	8.4×10^{-6}	9.7×10^{-6}	6.8×10^{-6}
$\langle{\rm cov}^{(\rm net)}\rangle_{\rm dat}$	9.0×10^{-6}	1.1×10^{-5}	6.8×10^{-6}
$r[{\rm cov}^{(\rm net)}]$	0.26	0.21	0.86

Table 4: Nonsinglet results

We have thus produced a set of 1000 networks, each trained on a replica of the original data set. The prediction for the structure function and its error computed from them according to eqs. (4.7), (4.8) are compared to the experimental data in figure 10. The features of this set of neural networks are summarized in table 4.

The table shows that central values are very well reproduced. The value of χ^2 of the two data sets is almost the same, and close enough to $\chi^2 \approx 1$, so that there is neither overlearning nor underlearning. The statistical error is substantially reduced by the network, especially for NMC data which have larger experimental error. Correlations correspondingly increase, so that the average covariance is essentially unchanged: this indicates that the systematics is reproduced in an unbiased way. The expectation that $\mathcal{R} \approx 0.5$ is beautifully fulfilled, indicating that the variance reduction is due to the fact that the network is learning an underlying law.

It is interesting to observe that while $\sigma^{(\rm net)}$ is highly correlated to $\sigma^{(\rm exp)}$ for BCDMS data, this does not happen for NMC. The reason for this can be traced to the fact that local information provided by each NMC data point is very weak: a few NMC points on top of the BCDMS data are sufficient to train the neural network and the remaining ones do not provide significant extra information. To illustrate this, in figure 11 we show the average error computed from a network trained on all BCDMS data, but only 20 NMC points (7% of NMC points arbitrarily chosen among those where the systematics is less than the statistical error): this fit is as good as the fit where all NMC data are kept. Note, however, that in this case the value of $E^{(0)}$ computed from the two data sets cross and

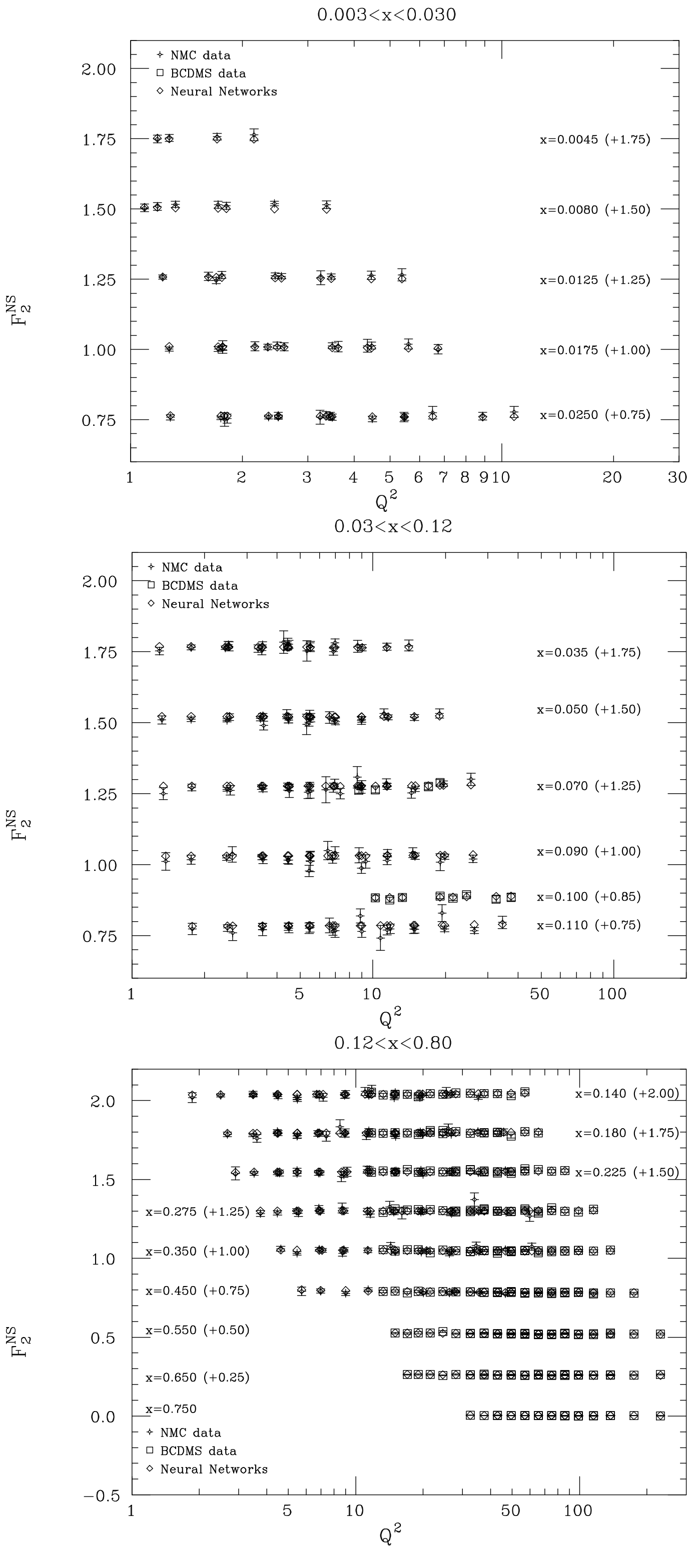

Figure 10: Nonsinglet data and network prediction at the corresponding (x, Q^2) value. For clarity, the points are offset by the amount given in parenthesis.

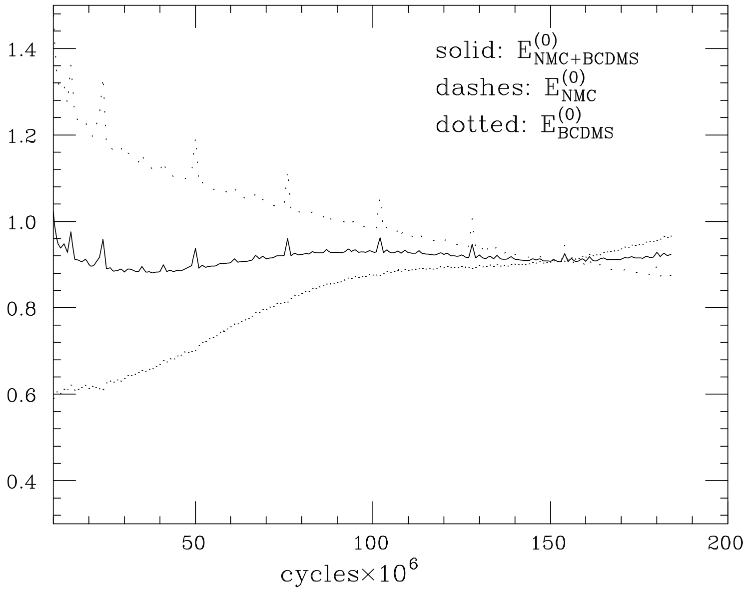

Figure 11: Nonsinglet training on all BCDMS data and 20 NMC data

rapidly diverge as the number of cycles grows above 1.5×10^8. This divergence is not observed when the full set of NMC data is included (figure 9): this shows that the reduced NMC data set does not carry the full information needed for a detailed fit, so the use of the full data set remains necessary if we want to avoid fine-tuning.

It is clear from figure 10 that the coverage of the data is such that, within the data region, the interpolating structure function is very well constrained by the data. In fact, the observed substantial error reduction is due to the fact that each new data point modifies very little the behaviour predicted by neighbouring data. It is therefore interesting to ask how far one can trust the neural networks when extrapolating outside the data region.

At large x, it can be easily verified that the extrapolation is sufficiently constrained by the kinematic bound $F_2(1, Q^2) = 0$ that no increase in the uncertainty is observed when x goes outside the region of the data even at $Q^2 = 3\,\mathrm{GeV}^2$, where there are no data with $x > 0.3$. In the small x region, instead, the uncertainty increases very rapidly: for instance, at $Q^2 = 30\,\mathrm{GeV}^2$ the error is $\sigma \approx 0.004$ at $x = 0.3$ (bulk of the data), $\sigma \approx 0.006$ at $x = 0.1$ (edge of the data region) and $\sigma \approx 0.017$ at $x = 0.01$ (extrapolation by one order of magnitude in $1/x$). Similar behaviour is observed at other scales: hence, extrapolation at small x outside the data region is not predictive.

Coming now to extrapolation in Q^2, it turns out that extrapolation at large Q^2 is somewhat more reliable than extrapolation at low Q^2. For instance, at $x = 0.3$ the error is $\sigma \approx 0.004$ at $Q^2 = 30\,\mathrm{GeV}^2$ (bulk of the data). At low Q^2 it reaches $\sigma \approx 0.006$ at $Q^2 = 3\,\mathrm{GeV}^2$ (edge of the data), and becomes $\sigma \approx 0.016$ at $Q^2 = 1\,\mathrm{GeV}^2$ (extrapolation). At high Q^2 it reaches $\sigma \approx 0.004$ at $Q^2 = 100\,\mathrm{GeV}^2$ (edge of the data), and becomes $\sigma \approx 0.007$ at $Q^2 = 300\,\mathrm{GeV}^2$ (extrapolation). Similar behaviour is observed for other values of x. The fact that the extrapolation is more stable at large Q^2 can be understood as a consequence of the fact that, because of asymptotic freedom, the large Q^2 scaling

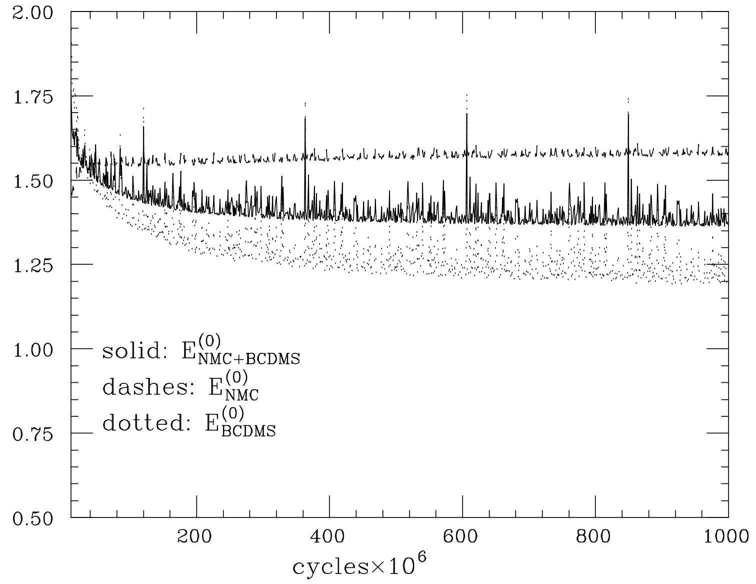

Figure 12: Proton: dependence of the error on the length of training.

violations are smaller and smoother, and thus easier for the neural network to capture. Note however that, in contrast to a parton fit, here the theoretical information on scaling violations encoded in evolution equations is not being used, and thus extrapolation, even at large Q^2, should be used with care.

5.3 Proton and deuteron

Neural networks for proton and deuteron are trained independently, but on a simultaneous set of pseudo-data. Namely, because proton and deuteron experimental data are correlated (recall section 2.1), each pseudo-data replica is generated as a set of correlated proton and deuteron data points. Two independent proton and deuteron neural networks are then trained on the proton and deuteron data of each replica. This guarantees that experimental correlations between proton and deuteron will be reproduced by the network sample.

5.3.1 Proton

The training of neural networks on proton structure function data is rather more difficult than in the nonsinglet case, because these data have a much smaller percentage error. The dependence of the average error on the training length is displayed in figure 12. The noisiness of the training process reflects the high precision of the data, especially for BCDMS. Clearly, there are two main difficulties: first, the BCDMS data are only learnt very slowly, and second, the quality of the fit to NMC data is poor, and does not improve (in fact, it slowly deteriorates) as the BCDMS data are learnt.

Inspection of the fits to data of a single experiment (figures 13, 14) reveals that, similarly to the nonsinglet case, NMC data are learnt very fast whereas BCDMS data require a much longer training. However, unlike in the nonsinglet case, networks trained on data of one experiment are now entirely incapable of predicting data from the other

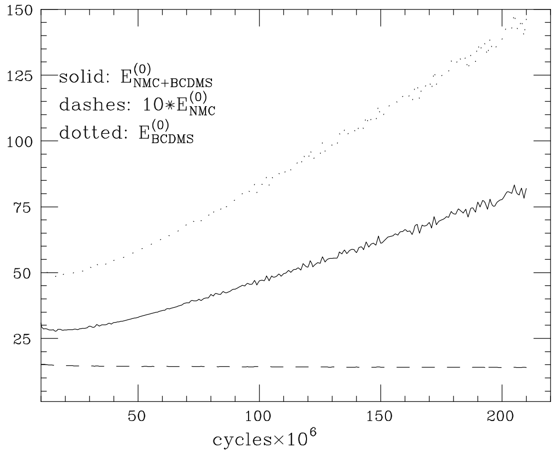

Figure 13: Proton training with NMC data.

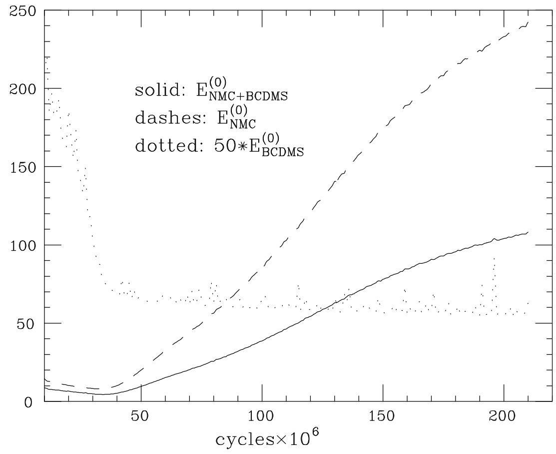

Figure 14: Proton training with BCDMS data.

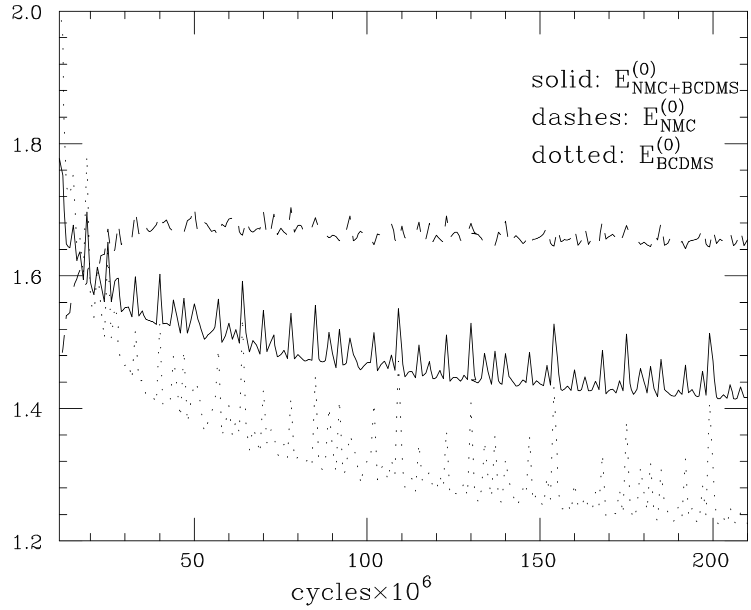

Figure 15: Proton weighted training (60% BCDMS, 40% NMC)

experiment: as one average error slowly improves, the other deteriorates dramatically. This suggests that an optimized training can again be obtained by giving more weight to the BCDMS data, however the imbalance cannot be too large because we cannot rely on one experiment being able to predict the other.

An optimal combination is found when BCDMS data are shown 60% of the time and NMC data 40% of time (figure 15). Both experiments then reach convergence after about 2×10^8 training cycles. However, unlike in the nonsinglet case, the average error for NMC remains significantly worse than BCDMS. In fact, this feature is already present in fits to a single experiment of figures 13, 14: the average error for NMC at convergence, though learnt fast, remains rather worse than BCDMS. We found no way of obtaining better fits

N_{net}	NMC+BCDMS	NMC	BCDMS
χ^2	1.30	1.53	1.11
$\langle E \rangle_{\rm rep}$	3.55	2.69	4.23
$\mathcal{R}$	1.45	0.75	2.00
$r\left[F^{net}\right]$	0.996	0.956	0.999
$\langle V[\sigma^{(\rm net)}]\rangle_{\rm dat}$	1.2×10^{-4}	2.3×10^{-4}	2.5×10^{-5}
$\langle PE[\sigma^{(\rm net)}]\rangle_{\rm dat}$	50%	66%	37%
$\langle \sigma^{(\rm exp)}\rangle_{\rm dat}$	0.012	0.017	0.007
$\langle \sigma^{(\rm net)}\rangle_{\rm dat}$	0.007	0.009	0.006
$r[\sigma^{(\rm net)}]$	0.68	0.17	0.98
$\langle V[\rho^{(\rm net)}]\rangle_{\rm dat}$	0.30	0.27	0.31
$\langle \rho^{(\rm exp)}\rangle_{\rm dat}$	0.27	0.17	0.33
$\langle \rho^{(\rm net)}\rangle_{\rm dat}$	0.69	0.53	0.80
$r[\rho^{(\rm net)}]$	0.33	0.15	0.37
$\langle V[{\rm cov}^{(\rm net)}]\rangle_{\rm dat}$	1.5×10^{-9}	2.7×10^{-9}	7.2×10^{-10}
$\langle {\rm cov}^{(\rm exp)}\rangle_{\rm dat}$	3.1×10^{-5}	4.5×10^{-5}	2.3×10^{-5}
$\langle {\rm cov}^{(\rm net)}\rangle_{\rm dat}$	3.2×10^{-5}	3.7×10^{-5}	2.9×10^{-5}
$r[{\rm cov}^{(\rm net)}]$	0.53	0.36	0.70

Table 5: Proton results

of NMC data even with NMC data only, much longer training, or larger architecture of the network. This appears therefore to be a feature of the data, to which we will come back shortly.

It is interesting to observe that the constraint $F_2(1, Q^2) = 0$ is crucial in order to ensure proper convergence of these fits. On the other hand, it should be noticed that, because this constraint is only imposed with an absolute error of 10^{-3}, the structure function fits in the vicinity of $x = 1$ are only reliable within this accuracy, especially at low Q^2 where no large x data are available.

The final training parameters are chosen as follows: 9.6×10^6 cycles with learning rate $\eta = 9 \times 10^{-3}$ and error function eq. (4.2), followed by 2×10^8 cycles with $\eta = 4 \times 10^{-8}$ 60% of the times on BCDMS data and 40% of the times on NMC data and error function eq. (4.4). The momentum term eq. (3.12) is always set at $\alpha = 0.9$. A sample of 1000 networks trained in this way is used to produce values of the structure function and error which are compared to the data in table 5.

Table 5 shows that all central values are well reproduced, even though the χ^2 for NMC is rather high. Errors and correlations are very well reproduced for the BCDMS data. There does not appear to be any reduction of variance for this experiment: $\sigma^{(\rm net)} \approx \sigma^{(\rm exp)}$ and $\rho^{(\rm net)} \approx \rho^{(\rm exp)}$: the networks simply reproduce the behaviour of the data. The prediction for the ratio eq. (5.16) $\mathcal{R} \approx 2$ is again beautifully borne out by the data. The NMC data instead display significant variance reduction. However, the covariance is not reduced, indicating that systematics are faithfully reproduced, and the value of $\mathcal{R}$ is in good agreement with the

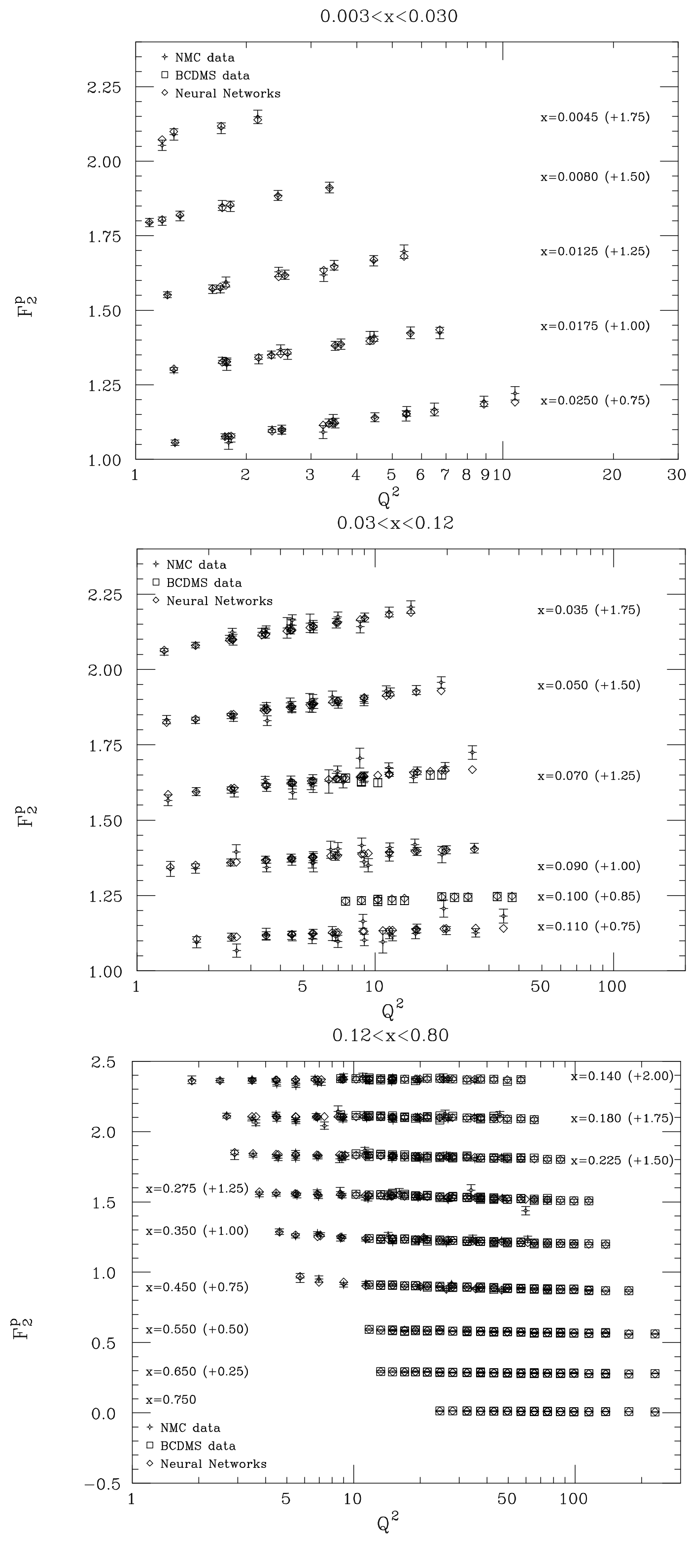

Figure 16: Proton data and network prediction at the corresponding (x, Q^2) value.

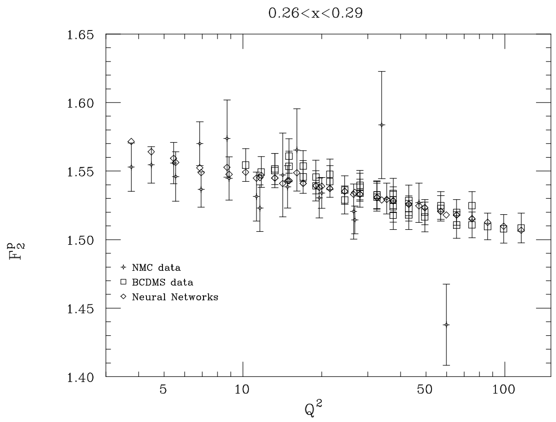

Figure 17: Blow-up of proton data and predictions.

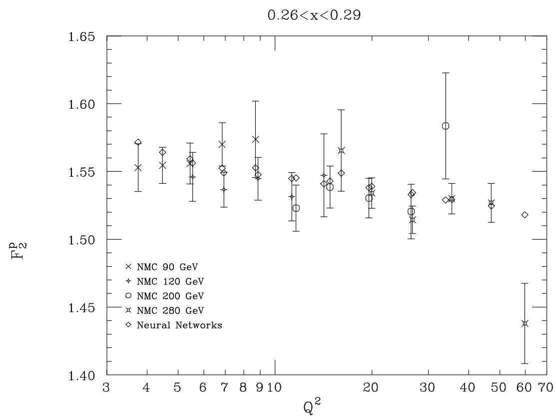

Figure 18: Proton NMC data at different beam energies, and associate predictions.

expectation of eq. (5.17), with $\hat{\sigma} = \langle \sigma^{(\text{net})} \rangle$. The weak scatter correlation of variances can be understood analogously to the nonsinglet case, as a consequence of the fact that the nets can predict most NMC data based on the information provided by a subset of these data.

We are therefore left with the problem of the large value of χ^2 for NMC. The difficulty in obtaining a satisfactory fit of NMC data was already noted by other groups [5, 3] in the context of global parton fits. The origin of the problem can be understood by taking a closer look at the NMC data, especially in the central kinematic region (middle plot in figure 16). It is clear that a sizable fraction of these data are not compatible with the remaining ones: some data points deviate by several standard deviation from other measurements performed at equal or almost equal values of (x, Q^2). This may suggested underestimated systematics. A blow-up of the data, however (figure 17), reveals that NMC data not only disagree with BCDMS data with the same (x, Q^2), but in fact also with other NMC data at neighbouring (x, Q^2). Furthermore (figure 18) the inconsistency persists even within sets of data taken by NMC at fixed beam energy. Therefore, none of the known sources of systematics can cause this discrepancy. Inspection reveals that the data which appear to be display the largest disagreement are always at the edge of a bin (e.g., the largest or smallest Q^2 value for given x). The cumulative effect of these inconsistencies leads to a very large χ^2 which cannot be reasonably attributed to a statistical fluctuation.

Whereas we are unable to pinpoint the precise origin of the effect, we conclude that a subset of the NMC data are not consistent with remaining information on the proton structure function, and that this inconsistency cannot be cured by enlarging any of the experimental systematics. The neural network correctly discards these points, and the relatively large value of χ^2 signals this inconsistency in the data.

A consequence of the relatively poor quality of the fit to NMC proton data is that in this case extrapolation at large Q^2 is as unreliable as extrapolation at low Q^2, because the large Q^2 region is dominated by the NMC data. Interpolation, as well as extrapolation at large x in the region where low x data are available remain quite accurate.

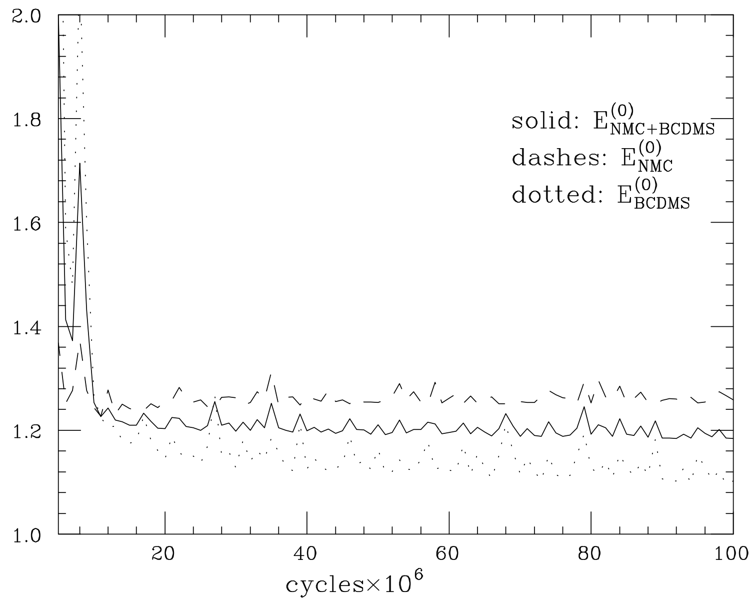

Figure 19: Deuteron: dependence of the error on the length of training.

5.3.2 Deuteron

The training of networks on deuteron data does not present specific problems. The dependence of the average error on the training length is displayed in figure 19, and shows reasonable fast convergence after about 10^8 training cycles. Fits to individual data sets are displayed in figures 20, 21: BCDMS data are learnt more slowly and have a slightly better $\langle E^{(0)} \rangle$ than NMC data. Training on individual data sets shows that the two data sets cannot be used to predict each other, in that the average error of the predicted set is poor, even though it does not deteriorate as the training progresses. No weighted training is necessary.

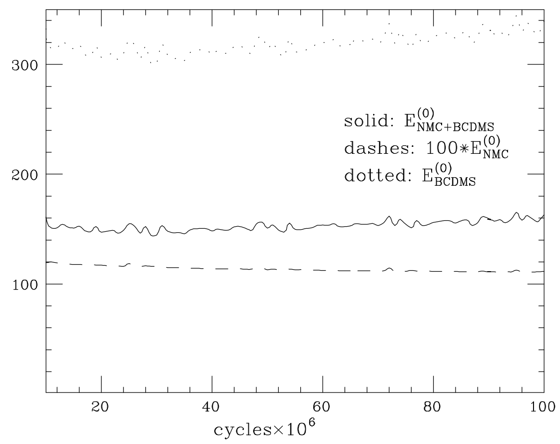

Figure 20: Deuteron training with NMC data.

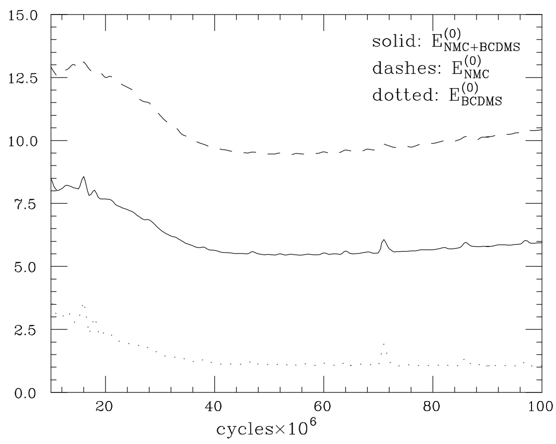

Figure 21: Deuteron training with BCDMS data.

N_{net}	NMC+BCDMS	NMC	BCDMS
χ^2	1.10	1.16	1.03
$\langle E\rangle_{\rm rep}$	2.64	2.74	2.53
$\mathcal{R}$	0.96	0.90	1.04
$r\left[F^{(\rm net)}\right]$	0.998	0.986	0.999
$\langle V[\sigma^{(\rm net)}]\rangle_{\rm dat}$	9.1×10^{-5}	1.4×10^{-4}	3.2×10^{-5}
$\langle PE[\sigma^{(\rm net)}]\rangle_{\rm dat}$	61%	66%	56%
$\langle\sigma^{(\rm exp)}\rangle_{\rm dat}$	0.010	0.014	0.006
$\langle\sigma^{(\rm net)}\rangle_{\rm dat}$	0.006	0.007	0.004
$r[\sigma^{(\rm net)}]$	0.72	0.34	0.94
$\langle V[\rho^{(\rm net)}]\rangle_{\rm dat}$	0.17	0.21	0.11
$\langle\rho^{(\rm exp)}\rangle_{\rm dat}$	0.31	0.22	0.43
$\langle\rho^{(\rm net)}\rangle_{\rm dat}$	0.63	0.57	0.71
$r[\rho^{(\rm net)}]$	0.50	0.27	0.69
$\langle V[{\rm cov}^{(\rm net)}]\rangle_{\rm dat}$	3.4×10^{-10}	1.2×10^{-9}	2.1×10^{-10}
$\langle{\rm cov}^{(\rm exp)}\rangle_{\rm dat}$	3.2×10^{-5}	4.0×10^{-5}	2.2×10^{-5}
$\langle{\rm cov}^{(\rm net)}\rangle_{\rm dat}$	2.6×10^{-5}	3.0×10^{-5}	1.3×10^{-5}
$r[{\rm cov}^{(\rm net)}]$	0.71	0.63	0.90

Table 6: Deuteron results

The final training parameters are chosen as follows: 9.6×10^6 cycles with learning rate $\eta = 9\times 10^{-3}$ and error function eq. (4.2), followed by $2\times 9\times 10^7$ cycles with $\eta = 4\times 10^{-8}$ and error function eq. (4.4). The momentum term eq. (3.12) is always set at $\alpha = 0.9$. A sample of 1000 networks trained in this way is used to produce values of the structure function and error which are compared to the data in figure 22. The features of these networks are summarized in table 6.

The results of table 6 show that central values and errors are well reproduced. There is a moderate amount of variance reduction for BCDMS data, and a somewhat larger reduction for NMC. The covariance is well reproduced, indicating that systematics are reproduced in a faithful way. The features interpolation and extrapolation are similar as in the nonsinglet case: interpolation and extrapolation at large x where small x data are available are quite accurate, extrapolation at large Q^2 by a small amount is tolerable, while extrapolation at small Q^2 or at small x are subject to large uncertainties.

6. Summary

We have presented a determination of the probability density in the space of structure functions for the structure function F_2 for proton, deuteron and nonsinglet, as determined from experimental data of the NMC and BCDMS collaborations. Our results, for each of the three structure functions, take the form of a set of 1000 neural nets, each of which gives a determination of F_2 for given x and Q^2. The distribution of these functions is a

JHEP05(2002)062

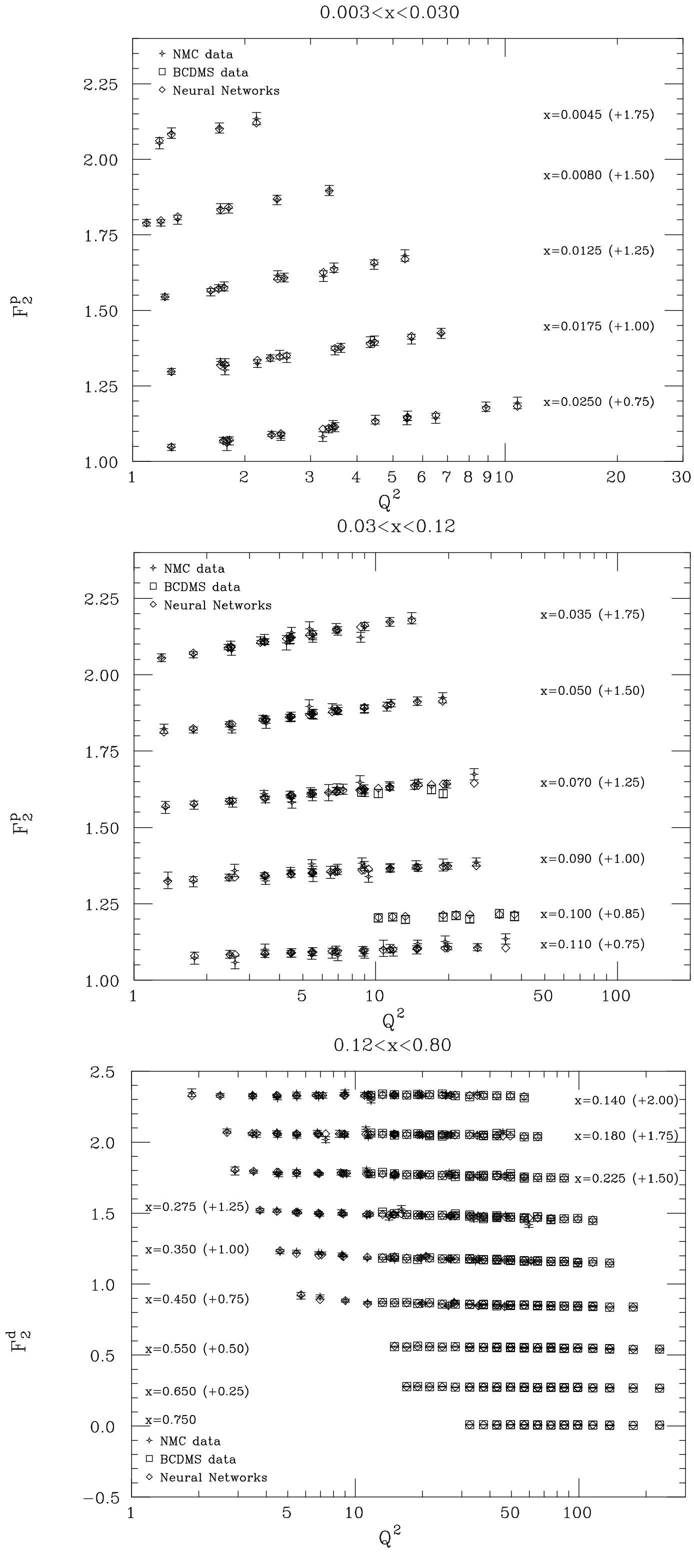

Figure 22: Deuteron data and network prediction at the corresponding (x, Q^2) value.

Monte Carlo sampling of the probability density. This Monte Carlo sampling has been obtained by first, producing a sampling of the space of data points based on the available experimental information through a set of Monte Carlo replicas of the original data, and then, training each neural net to one of these replicas.

In practice, all functions are given by a FORTRAN routine which reproduces a feed-forward neural network (described in section 3) entirely determined by a set of 47 real parameters. Each function is then specified by the set of values for these parameters. Our results are available at the web page `http://sophia.ecm.ub.es/f2neural/`. The full set of FORTRAN routines and parameters can be downloaded from this page. On-line plotting and computation facilities for structure functions, errors and point-to-point correlations are also available through this web interface.

Given the Monte Carlo sample of the probability density, any functional of the structure function, as well as errors and correlations, can be computed by averaging over the sample according to eq. (4.10). The sample has been carefully tested against the available data to produce reliable results wherever data are available. Because the data provide a rather fine scanning of the x, Q^2 plane (figure 1), this representation of the probability density is expected to be very accurate throughout the region of the data. However, care should be adopted in extrapolating results far from the data region. On the other hand, we have verified that as one gets further from the data region, the spread of the probability density rapidly increases: so, errors computed from the Monte Carlo sample should be a good indicator of the point where the extrapolation becomes unreliable.

Our results can be useful for a variety of applications to the precision phenomenology of deep-inelastic scattering. A particularly promising application consists of coupling the bias-free determination of structure functions discussed here with the bias-free technique for the description of QCD scaling violations based on the use of truncated moments [6]. In this approach, QCD evolution equations are directly expressed in terms of the scale dependence of the contribution to moments of structure functions from the experimentally accessible region. A precision determination of the strong coupling α_s based on these techniques will be presented in a companion paper [19].

In conclusion, this is the first complete determination of a probability density in a space of structure functions. As such, it provides an example of a bias-free determination of a function with its associate error from a discrete set of experimental data, which is a problem of foremost importance in the determination of errors on parton distributions.

Acknowledgments

We thank F. Zomer and A. Milsztajn for patiently explaining to us the structure of the data and their errors, and G. d'Agostini, R. Ball, L. Magnea and G. Ridolfi for several discussions along the course of this work. We are very grateful to L. Magnea for a critical reading of the manuscript, to R. Ball for stimulating comments, and to S. Simula for trying out the web interface. This work was supported in part by EU TMR contract FMRX-CT98-0194 (DG 12 - MIHT) and the Spanish and Catalan grants AEN99-0766, AEN99-0483, 1999SGR-00097.

JHEP05(2002)062

References

[1] See S. Forte, *Structure functions in perturbative QCD*, *Nucl. Phys.* **A 666** (2000) 113;
S. Catani et al., *QCD*, `hep-ph/0005025`;
S. Catani et al., *The QCD and standard model working group: summary report*, `hep-ph/0005114`.

[2] S. Alekhin, *Extraction of parton distributions and α_s from DIS data within the bayesian treatment of systematic errors*, *Eur. Phys. J.* **C 10** (1999) 395 [`hep-ph/9611213`].

[3] W.T. Giele and S. Keller, *Implications of hadron collider observables on parton distribution function uncertainties*, *Phys. Rev.* **D 58** (1998) 094023 [`hep-ph/9803393`];
W.T. Giele, S. Keller and D.A. Kosower, *Parton distributions with errors*, in *La Thuile 1999, Results and perspectives in particle physics* Frascati, 1999.

[4] W.T. Giele, S.A. Keller and D.A. Kosower, *Parton distribution function uncertainties*, `hep-ph/0104052`.

[5] J. Pumplin et al., *New generation of parton distributions with uncertainties from global QCD analysis*, `hep-ph/0201195`.

[6] S. Forte and L. Magnea, *Truncated moments of parton distributions*, *Phys. Lett.* **B 448** (1999) 295 [`hep-ph/9812479`];
S. Forte, L. Magnea, A. Piccione and G. Ridolfi, *Evolution of truncated moments of singlet parton distributions*, *Nucl. Phys.* **B 594** (2001) 46 [`hep-ph/0006273`];
A. Piccione, *Solving the altarelli-parisi equations with truncated moments*, *Phys. Lett.* **B 518** (2001) 207 [`hep-ph/0107108`].

[7] Spin Muon collaboration, *Spin asymmetries A(1) and structure functions g1 of the proton and the deuteron from polarized high energy muon scattering*, *Phys. Rev.* **D 58** (1998) 112001.

[8] New Muon collaboration, *Measurement of the proton and deuteron structure functions, F2p and F2d and of the ratio σ_l/σ_t*, *Nucl. Phys.* **B 483** (1997) 3 [`hep-ph/9610231`].

[9] BCDMS collaboration, *A high statistics measurement of the proton structure functions $F2(X,Q^2)$ and R from deep inelastic muon scattering at high Q^2*, *Phys. Lett.* **B 223** (1989) 485;
BCDMS collaboration, *A high statistics measurement of the deuteron structure functions $F2(X,Q^2)$ and R from deep inelastic muon scattering at high Q^2*, *Phys. Lett.* **B 237** (1990) 592.

[10] E665 collaboration, *Proton and deuteron structure functions in muon scattering at 470-GeV*, *Phys. Rev.* **D 54** (1996) 3006.

[11] P. Bosted et al., *Threshold inelastic electron scattering from the proton at high momentum transfer*, *Phys. Rev.* **D 49** (1994) 3091.

[12] CLAS 1999 data analysis, in preparation;
S. Simula, private communication.

[13] A. Milsztajn, private communication.

[14] T. Çuhadar, PhD. Thesis, Amsterdam Free University, 1998;
New Muon collaboration, M. Arneodo et al., *Measurement of the proton and the deuteron structure functions, F2p and F2d*, *Phys. Lett.* **B 364** (1995) 107 [`hep-ph/9509406`];
New Muon collaboration, *Measurement of the proton and the deuteron structure functions, F2(p) and F2(d)*, *Phys. Lett.* **B 364** (1995) 107 [`hep-ph/9509406`].

JHEP05(2002)062

[15] C. Peterson and T. Rögnvaldsson, *Lectures at the 1991 CERN school of computing*, preprint LU-TP-91-23;
B. Müller, J. Reinhardt and M. T. Strickland, *Neural networks: an introduction* Berlin, 1995;
G. Stimpfl-Abele and L. Garrido, *Fast track finding with neural nets*, *Comput. Phys. Commun.* **64** (1991) 46.

[16] F. Rosenblatt, *Principles of neurodynamics* New York, 1962.

[17] G. D'Agostini, *Probability and measurement uncertainty in physics — a bayesian primer*, hep-ph/9512295.

[18] G. Cowan, *Statistical data analysis* Oxford, 1998.

[19] S. Forte, J.I. Latorre, L. Magnea and A. Piccione, *Determination of α_s from scaling violations of truncated moments of structure functions*, hep-ph/0205286.