Neural network parameterizations of electromagnetic nucleon form-factors

Krzysztof M. Graczyk,[a,*] **Piotr Płonski**[b] **and Robert Sulej**[c]

[a]*Institute of Theoretical Physics, Wrocław University,*
pl. M. Borna 9, 50-204 Wrocław, Poland

[b]*Institute of Radioelectronics, Warsaw University of Technology,*
Nowowiejska 15/19, 00-665 Warsaw, Poland

[c]*A. Soltan Institute for Nuclear Studies,*
Hoza 69, 00-681 Warsaw, Poland

E-mail: kgraczyk@ift.uni.wroc.pl, P.Plonski@stud.elka.pw.edu.pl, Robert.Sulej@cern.ch

Abstract: The electromagnetic nucleon form-factors data are studied with artificial feed forward neural networks. As a result the unbiased model-independent form-factor parametrizations are evaluated together with uncertainties. The Bayesian approach for the neural networks is adapted for χ^2 error-like function and applied to the data analysis. The sequence of the feed forward neural networks with one hidden layer of units is considered. The given neural network represents a particular form-factor parametrization. The so-called *evidence* (the measure of how much the data favor given statistical model) is computed with the Bayesian framework and it is used to determine the best form factor parametrization.

Contents

1 Introduction

The electromagnetic (EM) form-factors (FF) of the nucleon are the quantities which embody the information about the complex electromagnetic structure of the proton and neutron [1]. In practice, the form-factors are introduced in order to model (on effective level) the electromagnetic hadronic current for elastic $ep\,(n)$ scattering. In the one photon exchange approximation it has the following form:

$$J^{\mu}_{e\,p(n)} = \overline{u}(p') \left[\gamma^{\mu} F_1^{p(n)}(Q^2) + \frac{i\sigma^{\mu\nu}q_{\nu}}{2M_{p(n)}} F_2^{p(n)}(Q^2) \right] u(p), \tag{1.1}$$

where $q = p' - p$ denotes the four-momentum transfer; $M_{p(n)}$ is the proton (neutron) mass; p' and p are outgoing and incoming nucleon momenta; $Q^2 \equiv -q^2$; $F_1^{p(n)}$ is the helicity non-flip Dirac proton (neutron) form-factor, while $F_2^{p(n)}$ denotes the helicity-flip Pauli proton (neutron) form-factor. The form-factors are normalized as follows:

$$F_1^p(0) = 1, \quad F_2^p(0) = \mu_p - 1, \quad F_1^n(0) = 0, \quad F_2^n(0) = \mu_n, \tag{1.2}$$

where $\mu_{p,n}$ is anomalous magnetic moment of the proton, neutron.

The nucleon is the many-body system of strongly interacting quarks (three valence quarks and any number of quark-antiquark pairs) and gluons. This complex system is described by the QCD (quantum chromodynamics) in the confinement regime. Study of the

EM form-factors gives an opportunity for testing the models describing the strong interactions. However, computing the EM form-factors from the first principles is an extremely difficult task. Nevertheless, some effort has been done with the effective approaches and the lattice QCD.

A good approximation of the FF is performed within the vector meson dominance models (VMD) [2–6]. There are interesting results obtained with constituent quark models [7–10] as well as with other approaches (see for review [11]). However, the given theoretical description usually works well only on limited Q^2 range. In order to describe the full Q^2 domain various approaches must be combined. Hence a proper prediction of the FF in wide Q^2 range requires to use complex phenomenological models which contain plenty of internal parameters.

On the other hand, the experimental data, which have been collected during the last sixty years, covers a wide Q^2 domain and are accurate enough to provide reasonable information about the nucleon electromagnetic structure [12]. Therefore one can try to represent the nucleon form-factors by the data itself without assuming any model constraints. In this article we follow this philosophy.

Description of the electromagnetic properties of the nucleon is a problem of great interest of modern particle physics. The knowledge of the nucleon form-factors is also important for practical applications. We mention two of them: (i) predicting the cross sections for the quasi-elastic charged current (CC) and elastic neutral current (NC) neutrino scattering off nucleon and nucleus [13]; (ii) investigation of the strange content of the nucleon in elastic lepton scattering off nucleons/nuclei [14, 15].

An accurate modeling of the neutrino-nucleus cross sections plays a crucial role in the analysis of the $\nu_\mu \to \nu_\tau$ neutrino oscillation data, collected in the long-baseline experiments. For instance in the experiments like K2K [16] or T2K [17] the neutrino energy spectrum is reconstructed from the quasi-elastic-like events. Observing the distortion of the energy spectrum in the far detector gives an indication for neutrino oscillation.

The investigation of the quasi-elastic CC neutrino-nucleon interactions gives an opportunity to explore the axial structure of the nucleon. The weak hadronic current is formulated assuming the conserved vector current (CVC) theorem. Then the vector part of the current is expressed in terms of the electromagnetic FF of the proton and neutron, while the axial contribution is described with two axial form factors: G_A and G_P (pseudoscalar axial form-factor). The hadronic weak current for the CC νn quasi-elastic scattering reads [18]

$$J^\mu_{\nu n, CC} = \overline{u}(p') \left[\gamma^\mu F_1^V(Q^2) + \frac{i\sigma^{\mu\nu}q_\nu}{2M} F_2^V(Q^2) + \gamma^\mu \gamma^5 G_A(Q^2) + \frac{q^\mu}{2M} \gamma^5 G_P(Q^2) \right] u(p), \tag{1.3}$$

where $M = (M_p + M_n)/2$. The isovector Dirac, Pauli form-factors are defined as follows:

$$F_{1,2}^V(Q^2) = F_{1,2}^p(Q^2) - F_{1,2}^n(Q^2). \tag{1.4}$$

If the partially conserved vector current hypothesis (PCAC) is assumed then the axial form-factors can be related: $G_P(Q^2) = 4M^2 G_A(Q^2)/(m_\pi^2 + Q^2)$. The G_A is usually pa-

rameterized with dipole functional form:

$$G_A(Q^2) = g_A \left(1 + \frac{Q^2}{M_A^2}\right)^{-2}, \quad g_A = -1.2695 \pm 0.0029. \tag{1.5}$$

M_A denotes the axial mass. Notice that recent studies [19, 20] suggest M_A value larger by about 20% with respect to the old measurements [21–23]. The impact of the electromagnetic form-factors on the axial mass extraction is small, but it can play a role in the future, when more precise measurements of the neutrino-nucleon cross-sections will be performed.

The precise knowledge of the EM form-factors together with uncertainties is more important for predicting the NC elastic νN reaction cross-section. The structure of the weak NC hadronic current is similar to (1.3) [24], namely:

$$J^\mu_{\nu p(n), NC} = \overline{u}(p') \left[\gamma^\mu F_1^{NC,p(n)}(Q^2) + \frac{i\sigma^{\mu\nu}q_\nu}{2M_{p(n)}} F_2^{NC,p(n)}(Q^2) + \gamma^\mu\gamma^5 G_A^{NC,p(n)}(Q^2)\right] u(p), \tag{1.6}$$

where

$$F_{1,2}^{NC,p(n)}(Q^2) = \pm\frac{1}{2}F_{1,2}^V(Q^2) - 2\sin\theta_W F_{1,2}^{p(n)}(Q^2) - \frac{1}{2}F_{1,2}^s(Q^2), \tag{1.7}$$

$$G_A^{NC,p(n)}(Q^2) = \pm\frac{1}{2}G_A(Q^2) - \frac{1}{2}G_A^s(Q^2), \tag{1.8}$$

θ_W is the Weinberg angle. $F_{1,2}^s(Q^2)$ and $G_A^s(Q^2)$ describe the strange content of the nucleon. We see that the investigation of the elastic NC neutrino-nucleon scattering gives the opportunity to explore the nucleon strangeness [24, 25] (mainly the axial strange part). The strangeness of the nucleon is also investigated in the elastic ep scattering [15, 26, 27]. The extraction of this contribution is sensitive to the accuracy of the EM form-factors. Therefore it is necessary to use the well determined FF parametrization together with the uncertainties.

There are many different phenomenological parametrizations of the EM form-factors [6, 28–36]. Some of these are based on the theoretical models, but mostly in practical applications simple functional parametrizations fitted to the data are applied [37]. The functional form is chosen to satisfy some general properties (proper behavior at $Q^2 \to 0$ and $Q^2 \to \infty$, scaling behavior). However, a particular choice of the parametrization determines the final fit and affects also the uncertainty. The form-factors parameterized by the large number of degrees of freedom have a tendency to describe the data too accurately, and the generality of the fit is lost. On the other hand, the model with a small number of the parameters may describe the data imprecisely. Moreover the complexity of the fit has an impact on its uncertainties.

Searching for the proper parametrization, which describes the data well enough without losing the generality of the fit is just solving the problem, known in statistics as *bias-variance trade-off* [38, 39]. Usually the most reasonable solution is chosen with a use of *common sense*, i.e. the fit which leads to the low enough $\chi^2_{\min}$ value is accepted, and more complex models are not considered. The task of this paper is to evaluate a model independent FF parametrizations, which will not be affected by the problems described above i.e. the common sense will be replaced by the objective Bayesian procedure.

One of the possible fitting techniques is to apply artificial neural networks (ANN). The ANN has already been used in the high energy physics for decades [40] and it has been shown to be a powerful tool in the field. The pattern recognition tasks like particle or interaction identification are efficiently addressed with the ANN based methods also in present experiments [41, 42]. The ANN are also applied to the function approximation and parameter estimation problems [43, 44].

The ANN techniques have already been applied by NNPDF collaboration [45] to represent the nucleon and deuteron EM structure functions [43, 46–50]. The method is based on the large collection of networks [48] of the same architecture prepared on the artificial data sets generated from original experimental measurements. Obtained fits are claimed to be unbiased due to networks being intentionally oversized — the number of free parameters, the network *weights*, is larger than required to solve the problem. To avoid potential over-fitting (representing the statistical fluctuations of the experimental data), that may arise under these conditions, the optimization of the network weights (so-called *training*) is stopped before reaching the minimum of the figure of merit (*error function*) calculated on training data. Stopping condition is based on the cross-validation technique, where the portion of available data is excluded from training. Such created subset is then used to calculate the test error function which starts to increase when the network becomes fitted to training data more than to the testing data. This observation is used to break the training. The best fit values and the uncertainties given by the NNPDF are computed by taking the average and standard deviation respectively, over the set of solutions obtained from the whole collection of the networks.

In the case of the present analysis the number of experimental points varies from 26 to 57, and we do not generate the Monte Carlo data. Therefore the cross-validation technique is unsuitable because constructing the testing data set can significantly restrict the information about the underlying data model used in training. Additionally our intention is to compare statistical models which are represented by the networks of various architectures and among them choose the most appropriate parametrization. It motivated us to consider another idea for finding the best fit and the choice of the neural network architecture. We apply Bayesian framework (BF) for the ANN. It is a different philosophy of building the statistical model than the NNPDF approach. However, both techniques are complementary and face with the same *bias-variance trade-off*. A pedagogical description of the main ingredients of both methodologies can be found in Ch. Bishop's book [38] (chapters 9 and 10 respectively).

In the BF approach the sequence of neural networks characterized by different number of hidden units is considered. A given network of a particular size has its specific ability to adjust to training data i.e. small networks give smooth approximation, large networks can over-fit the data. One can think that the network of a particular architecture represents the particular statistical model. With the help of the Bayesian technique we compare the models and choose the most appropriate one. This method has been developed for the ANN [38, 51–53] in nineties of last century. We adapted this approach for the purpose of χ^2 minimization. In practice, the so-called *evidence* is computed for every network type in order to select the most appropriate parametrization for given data set. The evidence is a probabilistic measure which indicates the best solution.

The network of particular architecture has weights that need to be optimized i.e. the global minimum of the error function is searched for. In order to get the solution we consider various gradient algorithms. However, the training done with these algorithms can stick in local minimum. Therefore for a given network architecture the sample of networks with randomized initial weights is trained to find a single configuration at the global minimum (this procedure is described in section 2.2). The error function is modified with so-called *regularization* term to improve generalization ability (to control the over-fitting); the extent of regularization is controlled in the statistically optimal way, also as a part of the Bayesian algorithm.

The main results of our studies are unbiased proton and neutron FF parametrizations, available in the numerical form at [54] as well as in the analytical ones (see appendix A). The proposed statistical method also allows to compute the form-factor uncertainties (from the covariance matrix). One of the strengths of this methodology is its ability of studying the deviations of the form-factors from the dipole form.

Eventually, let us mention that the previous (non-neural) form-factor data analysis (with ah-hoc parametrizations) have been done in the non-Bayesian spirit i.e. authors do not compare the possible FF parametrizations in order to choose the most suitable. Usually the one particular functional form was discussed and analyzed with the χ^2 framework.

The paper is organized as follows. In section 2 the feed forward neural networks are shortly reviewed. Section 3 describes the Bayesian approach to neural networks. The last section contains the numerical results and discussion. We supplement the article with the appendix, which presents the fits in the analytical form.

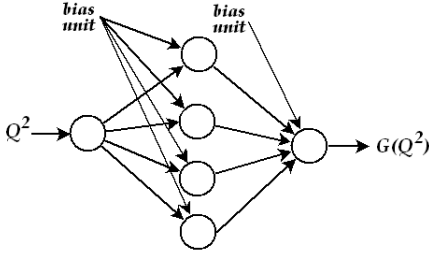## 2 Feed forward neural networks

2.1 Multi-layer perceptron

We consider the feed-forward neural network in the so-called multi-layer perceptron (MLP) configuration. The network structure (shown in figure 1) contains: the input layer, the layer of M hidden neurons and a single neuron in the output layer. We will say that the network of type 1-M-1 is considered. Each neuron (see figure 2) calculates the output value as an activation function f_{act} of the weighted sum of its inputs:

$$f_{act}\left(\sum_i w_i \mu_i\right), \tag{2.1}$$

where w_i denotes the weight parameter, while μ_i represents the output value of the unit from previous layer. Neurons in the hidden layer are usually non-linear, with the sigmoid or hyperbolic tangent functions denoted as f_{act}; in this analysis the output neuron is linear function. In general, the ANN gives a map $(\vec{y})$ of the input into the output vector spaces. The overall network response is then a deterministic function of the input variable (vector $\vec{in}$), and the weight parameters:

$$\vec{y}(\vec{in}, \vec{w}) : \ \mathcal{R}^{D_{\text{input}}} \to \mathcal{R}^{D_{\text{output}}}. \tag{2.2}$$

Figure 1. The feed forward neural network (of type 1-4-1) with one hidden layer, one input and output unit and 4 hidden units, representing the form-factor $G(Q^2)$.
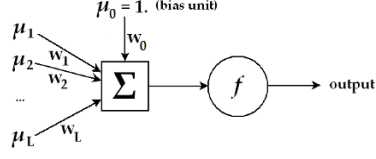
Figure 2. Single neuron.

In our analysis the ANN is expected to approximate the given form-factor G depending on the input variable Q^2:

$$y(Q^2, \vec{w}) = G(Q^2). \tag{2.3}$$

Let $\mathcal{D}$ denotes the training data set of N points:

$$\mathcal{D} = \{(x_1, t_1, \Delta t_1), \ldots, (x_i, t_i, \Delta t_i), \ldots, (x_N, t_N, \Delta t_N)\}, \tag{2.4}$$

where t_i is the measured value of the nucleon form-factor at the point $x_i = Q_i^2$, while the Δt_i denotes the total experimental error. The network training goal is to find $\vec{w}$ that minimizes an error function defined here as:

$$S(\vec{w}, \mathcal{D}) = \chi^2(\vec{w}, \mathcal{D}) + \alpha E_w(\vec{w}). \tag{2.5}$$

χ^2 term is the error on data:

$$\chi^2(\vec{w}, \mathcal{D}) = \sum_{i=1}^{N} \left(\frac{y(x_i, \vec{w}) - t_i}{\Delta t_i} \right)^2. \tag{2.6}$$

α parameter is the factor for the regularization term E_w. In this work we apply the weight decay formula [56]:

$$E_w(\vec{w}) = \frac{1}{2} \sum_{i=1}^{W} w_i^2, \tag{2.7}$$

where W denotes the total number of weights in the network (including bias weights).

In general, the output of the MLP with M hidden neurons and the linear output neuron can be written in the form:

$$y(\mu_0, \ldots, \mu_L) = \sum_{m=0}^{M} \left[w_m^{\text{out}} f_{act} \left(\sum_{l=0}^{L} w_l^{hid_m} \mu_l \right) \right]. \tag{2.8}$$

In this paper we consider the neural networks with ($L = 1$): one input unit $\mu_1 = Q^2$, and one bias unit $\mu_0 = 1$ in the first layer. The bias of the output neuron in the above formula
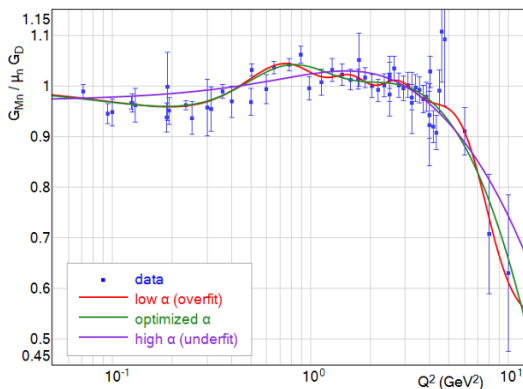
Figure 3. Fits of the $G_{Mn}/\mu_n G_D$ data parametrized with the network of large size. The results were obtained with: fixed, underestimated value of α (red line); fixed, overestimated value of α (violet line); online optimized value of α (green line).
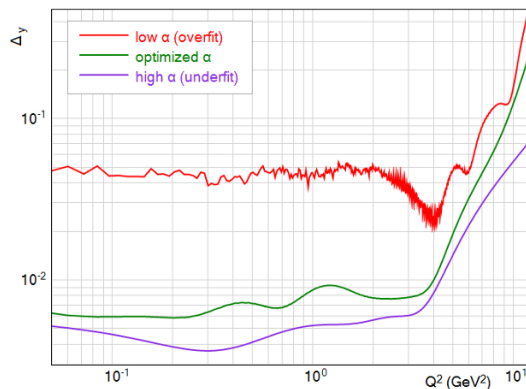
**Figure 4**. The $G_{Mn}/\mu_n G_D$ uncertainties (of the fits presented in figure 3) computed with (3.16).

is considered as the hidden neuron with the constant output, $f_{act} = 1$. Such representation closely corresponds to the Kolmogorov function superposition theorem [55]. Basing on this relation it was shown [57, 58] that the MLP can approximate any continuous function of its inputs, to the extent that depends on the number of the hidden neurons. However, in the practical problem we are faced, the desired function is not known and only the limited number of experimental points is available instead. It leads to the mentioned earlier *bias-variance* problem. The output of the oversized network tends to approach closely to the training data points if weights are not constrained during the training. Usually this means that statistical fluctuations are captured. The weight regularizing term (eq. (2.7)) penalizes the large weight values and smooths out the network output, but on the other hand, applying the regularization with overestimated value of the factor α leads to the fit which does not reproduce significant features of the training data. The effect of applying regularization is illustrated in figures 3 and 4, where the relatively large network was trained with various values of the factor α. Similarly, the network with the low number of the hidden neurons may be not capable to represent the desired function. Section 3 presents the statistical approach to determine the network size appropriate to the given data set and to predict the optimal value of α.

2.2 Training of network

It has been already mentioned that the training of the network is the process of establishing $\vec{w}$ which minimizes the error function (2.5). We denote the minimal error by $S(\vec{w}_{MP}, \mathcal{D})$ (the notation will become clear latter).

The first algorithm for the MLP weights optimization, the *back-prop*, was proposed by D. E. Rumelhart *et al.* in [59]. Currently there is a wide range of gradient descent and stochastic algorithms available for the network training. We use mainly the *Levenberg-*
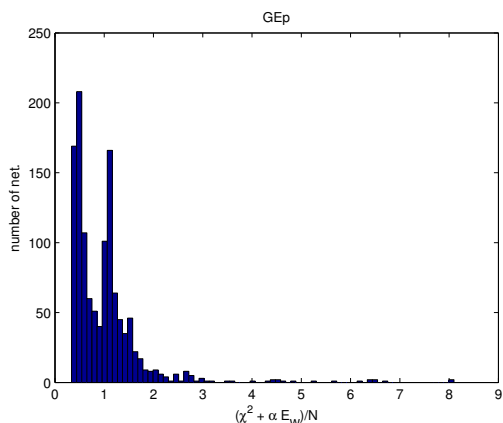
Figure 5. $S(\vec{w}_{MP}, \mathcal{D})/N$ distribution obtained for the network sample trained with the G_{Ep}/G_D data. The 1-3-1 network type was applied.
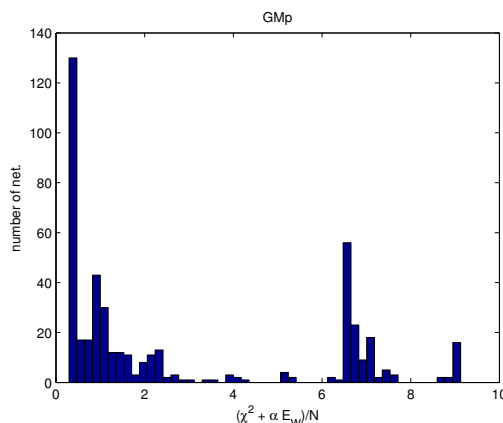
Figure 6. $S(\vec{w}_{MP}, \mathcal{D})/N$ distribution obtained for the network sample trained with the $G_{Mp}/\mu_p G_D$ data. The 1-3-1 network type was applied.

Marquardt algorithm [60, 61], since it converges efficiently and does not require precise parameters tuning. However, we trained the networks also with *quick-prop* [62], and *rprop* [63] algorithms. The obtained results were very similar.

The algorithms we use, as all gradient based optimization patterns, may suffer from local minima. Therefore for given network type 1-M-1 we consider a large sample of networks with different (randomized) initial weights. We use a limited range of initial weight values according to the properties of the neuron activation function.[1]

After the training of the sample of networks of the same type the distribution of the total error value $S(\vec{w}_{MP}, \mathcal{D})$ is obtained (see figures 5 and 6). Notice that the distribution sharply starts at particular S_{cut} value. Such clear cut on the error value gives us an indication that the global minimum is well approximated. The number of networks in the sample required to determine the clear S_{cut} value depends on the complexity of the data. The typical number we obtained were as follows: 150 (G_{En} data), 250 (G_{Mp} data), 700 (G_{Mn} data), and 1300 (G_{Ep} data).

The Bayesian framework allows to choose from the sample *the best model*. It is the solution characterized by the highest evidence (as it is described in section 3). In practice, if the total error is too big then the evidence is too low and the given network can be discarded from further analysis. Hence to simplify the numerical procedure we take into consideration ten fits (neural networks) with the lowest total error values. They are also characterized by the low χ^2 value, namely $\chi^2(\vec{w}_{MP}, \mathcal{D})/(N - W) < 1$; $N - W$ is the number of degrees of freedom. Among them the one with the maximal evidence is selected for further comparison with the network of other types. It was interesting to observe that

---

[1]High weight values make the sigmoid activation function very steep. Then the neuron input values have a very narrow range, where the neuron output is not saturated — this would efficiently block the training, where the output derivative is used extensively. Hence we restrict the initial weight range to $|w_{initial}| = f_{act}^{sat}/(L\overline{\mu})$, where f_{act}^{sat} is the value for which activation function saturates, L is the number of neuron inputs, $\overline{\mu}$ is the mean neuron input value.

the fit parametrizations given by the average over the fits selected by lowest error value were found to be very similar to those indicated by the highest evidence in each sample. This observation confirms that all solutions we select from the sample are localized in close neighborhood of the global minimum and are very similar to the one indicated by the highest evidence.

3 Bayesian approach to neural networks

The Bayesian framework (BF) for the model comparison [38, 51–53, 64] is taken into consideration. We adapt this framework for χ^2 minimization purpose. The data is analyzed with the set of various neural networks types $\mathcal{A}_M$: 1-M-1. Given neural network of architecture $\mathcal{A}_i$ corresponds to a particular statistical model (hypothesis) describing data. The BF allows to:

- quantitatively classify the hypothesis;

- choose objectively the best model (neural network) for representing a given data set;

- establish objectively the weight decay parameter α (see eq. (2.7));

- compute the uncertainty for the neural network response (output), and uncertainties for other network parameters.

The approach in natural way embodies the so-called Occam's razor criterium which penalizes more complex models and prefers simpler solutions.

3.1 Bayesian algorithm

At the beginning of the fitting procedure every neural network architecture $\mathcal{A}_M$ is classified by the prior probability $\mathcal{P}(\mathcal{A}_M)$. After the training of the network with the data $\mathcal{D}$, the posterior probability is evaluated $\mathcal{P}(\mathcal{A}_M|\,\mathcal{D})$ i.e. a probability of the model $\mathcal{A}_M$ given data $\mathcal{D}$. It classifies quantitatively considered hypothesis.

On the other hand applying the Bayes' theorem allows to express the posterior probability in the following way:

$$\mathcal{P}(\mathcal{A}_M|\,\mathcal{D}) = \frac{\mathcal{P}(\mathcal{D}|\,\mathcal{A}_M)\,\mathcal{P}(\mathcal{A}_M)}{\mathcal{P}(\mathcal{D})}, \tag{3.1}$$

where:

$$\mathcal{P}(\mathcal{D}|\,\mathcal{A}_M) \tag{3.2}$$

is called evidence [51] (probability of the data $\mathcal{D}$ given $\mathcal{A}_M$).

There is no reason to prefer some particular model before starting data analysis, hence:

$$\mathcal{P}(\mathcal{A}_1) = \mathcal{P}(\mathcal{A}_2) = \ldots = \mathcal{P}(\mathcal{A}_M) = \ldots \tag{3.3}$$

Then if one neglects the normalization factor $\mathcal{P}(\mathcal{D})$ the evidence (3.2) is the probability distribution which quantitatively classifies hypothesis.

The evidence is constructed in so called hierarchical approach. It is a three level procedure. Applying Bayes' theorem the probability distribution for the weights parameters is constructed, then the probability distribution of the decay parameter α, and eventually the evidence are evaluated.

$$\mathcal{P}\left(\vec{w}\mid \mathcal{D}, \alpha, \mathcal{A}_M\right) = \frac{\mathcal{P}\left(\mathcal{D}\mid \vec{w}, \alpha, \mathcal{A}_M\right)\mathcal{P}\left(\vec{w}\mid \alpha, \mathcal{A}_M\right)}{\mathcal{P}\left(\mathcal{D}\mid \alpha, \mathcal{A}_M\right)} \rightarrow \tag{3.4}$$

$$\mathcal{P}\left(\alpha\mid \mathcal{D}, \mathcal{A}_M\right) = \frac{\mathcal{P}\left(\mathcal{D}\mid \alpha, \mathcal{A}_M\right)\mathcal{P}\left(\alpha\mid \mathcal{A}_M\right)}{\mathcal{P}\left(\mathcal{D}\mid \mathcal{A}_M\right)} \rightarrow \tag{3.5}$$

$$\mathcal{P}\left(\mathcal{A}_M\mid \mathcal{D}\right) = \frac{\mathcal{P}\left(\mathcal{D}\mid \mathcal{A}_M\right)\mathcal{P}(\mathcal{A}_M)}{\mathcal{P}(\mathcal{D})}. \tag{3.6}$$

Below the short description of the Bayesian approach is presented.

Constructing the weight parameter distribution. The probability distribution for the neural network weights is built, assuming that regularization parameter α is fixed:

$$\mathcal{P}\left(\vec{w}\mid \mathcal{D}, \alpha, \mathcal{A}_M\right) = \frac{\mathcal{P}\left(\mathcal{D}\mid \vec{w}, \alpha, \mathcal{A}_M\right)\mathcal{P}\left(\vec{w}\mid \alpha, \mathcal{A}_M\right)}{\mathcal{P}\left(\mathcal{D}\mid \alpha, \mathcal{A}_M\right)}, \tag{3.7}$$

where $\mathcal{P}\left(\vec{w}\mid \alpha, \mathcal{A}_M\right)$ is a prior probability distribution of weights, while $\mathcal{P}\left(\mathcal{D}\mid \vec{w}, \alpha, \mathcal{A}_M\right)$ is the likelihood function. In the case of present analysis the likelihood function is given by the χ^2 function, namely:

$$\mathcal{P}\left(\mathcal{D}\mid \vec{w}, \alpha, \mathcal{A}_M\right) = \frac{1}{Z_\chi}\exp[-\chi^2(\vec{w}, \mathcal{D})], \quad Z_\chi = \int d^N t \exp[-\chi^2(\vec{w}, \mathcal{D})] = \pi^{\frac{N}{2}}\prod_{i=1}^{N}\Delta t_i. \tag{3.8}$$

The prior probability should be as general as possible. Indeed, there are plenty of possibilities (e.g. Laplacian or entropy-based priors see discussion in ref. [65]). We assume that every weight parameter is equally distributed according to a Gaussian distribution (with the zero mean and the variance of $1/\sqrt{\alpha}$)

$$\mathcal{P}\left(\vec{w}\mid \alpha, \mathcal{A}_M\right) = \frac{1}{Z_w(\alpha)}\exp[-\alpha E_w], \quad Z_w(\alpha) = \int d^W w \exp[-\alpha E_w] = \left(\frac{2\pi}{\alpha}\right)^{\frac{W}{2}} \tag{3.9}$$

(the arguments supporting above choice of the prior are presented in section 3.2). It gives the probabilistic interpretation for the regularization function E_w defined in the previous section (see eq. (2.7)). Then we see that:

$$\mathcal{P}\left(\mathcal{D}\mid \alpha, \mathcal{A}_M\right) = \int d^W w \mathcal{P}\left(\mathcal{D}\mid \vec{w}, \alpha, \mathcal{A}_M\right)\mathcal{P}\left(\vec{w}\mid \alpha, \mathcal{A}_M\right) = \frac{Z_M(\alpha)}{Z_\chi Z_w(\alpha)}, \tag{3.10}$$

$$Z_M(\alpha) = \frac{(2\pi)^{\frac{W}{2}}}{\sqrt{|A|}}\exp\left[-\chi(\vec{w}_{MP}) - \alpha E_w(\vec{w}_{MP})\right]. \tag{3.11}$$

The last integral was computed by expanding the error function up to the Hessian term:

$$S(\vec{w}, \mathcal{D}) = S(\vec{w}_{MP}, \mathcal{D}) + \frac{1}{2}(\vec{w} - \vec{w}_{MP})^T A(\vec{w} - \vec{w}_{MP}), \tag{3.12}$$

where $\vec{w}_{MP}$ is the vector of weights which minimizes $S(\vec{w}, \mathcal{D})$ (maximizes the posterior probability (3.7)).

The Hessian matrix reads

$$A_{ij} = \nabla_i \nabla_j \, S|_{\vec{w}=\vec{w}_{MP}} = \nabla_i \nabla_j \chi^2(\vec{w}, \mathcal{D}) + \alpha \delta_{ij} \qquad (3.13)$$

$$= 2 \sum_{k=1}^N \left[\frac{\nabla_i y(x_k, \vec{w}_{MP}) \nabla_j y(x_k, \vec{w}_{MP})}{\Delta t_k^2} + \frac{(y(x_k, \vec{w}_{MP}) - t_k)}{\Delta t_k^2} \nabla_i \nabla_j y(x_k, \vec{w}_{MP}) \right] + \alpha \delta_{ij}. \qquad (3.14)$$

We compute the full Hessian matrix [66]. Usually the double differential term in (3.14) is neglected, which is a good approximation only at the minimum. Taking into account full Hessian plays a crucial role in optimizing α parameter, as it will become clear below.

The network response uncertainty Δy is defined by the variance:

$$(\Delta y(x))^2 = \int d^W w \, [y(x, \vec{w}) - \langle y(x) \rangle]^2 \, \mathcal{P}(\vec{w}| \, \alpha, \mathcal{D}, \mathcal{A}_M). \qquad (3.15)$$

In the first approximation it is expressed by the covariance matrix, i.e. inverse of the Hessian matrix:

$$(\Delta y(x))^2 = (\nabla y(x, \vec{w}_{MP}))^T A^{-1} \nabla y(x, \vec{w}_{MP}). \qquad (3.16)$$

In appendix A the covariance matrices obtained for every considered problem are presented.

Constructing α the distribution of the parameter α. The α parameter is established by applying the so-called *evidence approximation* [51, 52, 67, 68], the method, which is equivalent to *type II maximum likelihood* in conventional statistics.

The Bayes' rule leads to:

$$\mathcal{P}(\alpha| \, \mathcal{D}, \mathcal{A}_M) = \frac{\mathcal{P}(\mathcal{D}| \, \alpha, \mathcal{A}_M) \, \mathcal{P}(\alpha| \, \mathcal{A}_M)}{\mathcal{P}(\mathcal{D}| \, \mathcal{A}_M)}, \qquad (3.17)$$

where the $\mathcal{P}(D| \, \alpha, \mathcal{A}_M)$ has been obtained in the previous section (see eq. (3.10)).

We are searching for the α_{MP} parameter, i.e. the one which maximizes the prior probability (3.17). It can be shown that in the *Hessian approximation* it is given by the solution of the equation:

$$2\alpha_{MP} E_w(\vec{w}_{MP}) = \sum_{i=1}^W \frac{\lambda_i}{\lambda_i + \alpha_{MP}} \equiv \gamma, \qquad (3.18)$$

where λ_i's are eigenvalues of the matrix $\nabla_n \nabla_m \chi^2|_{\vec{w}=\vec{w}_{MP}}$. In practice, the eigenvalues depend on α, therefore to get a proper α_{MP} the α parameter is iteratively changed during the training process i.e.:

$$\alpha_{k+1} = \gamma(\alpha_k)/2E_w(\vec{w}). \qquad (3.19)$$

The iteration procedure fixes in the optimal way the α parameter. The typical dependence of α_k on the iteration step is presented in figure 8. In section 3.2 it is shown that the choice of the initial α value has a small impact on the final results.

At the end of the training procedure one can approximate (3.10) as follows:

$$\mathcal{P}\left(\mathcal{D}|\ln\alpha,\mathcal{A}_M\right) = \mathcal{P}\left(\mathcal{D}|\ln\alpha_{MP},\mathcal{A}_M\right)\exp\left[-\frac{(\ln\alpha - \ln\alpha_{MP})^2}{2\sigma_{\ln\alpha}^2}\right], \qquad (3.20)$$

where in the *Hessian approximation* $\sigma_{\ln\alpha} \approx 2/\gamma$.

Constructing the evidence. The evidence for given model is defined by denominator of (3.17). If one assumes the uniform prior distribution of $\ln\alpha$ parameter[2] on some large $\ln\Omega$ region then the evidence can be approximated by:

$$\mathcal{P}\left(\mathcal{D}|\,\mathcal{A}_M\right) \approx \mathcal{P}\left(\mathcal{D}|\,\alpha_{MP},\mathcal{A}_i\right)\frac{2\pi\sigma_\alpha}{\ln\Omega}. \qquad (3.21)$$

The $\ln\Omega$ is a constant which is the same for the all hypotheses.

The ln of evidence (we show only model independent terms) reads

$$\ln\mathcal{P}\left(\mathcal{D}|\,\mathcal{A}_M\right) \approx -\chi^2(\vec{w}_{MP}) - \alpha_{MP}E_w(\vec{w}_{MP}) - \frac{1}{2}\ln|A| + \frac{W}{2}\ln\alpha_{MP} - \frac{1}{2}\ln\frac{\gamma}{2}. \quad (3.22)$$

The first term in the above expression, $-\chi^2(\vec{w}_{MP})$, (usually of low-value for simple models) is the misfit of the approximated data, while the next four terms constitute the so called Occam factor, which penalizes the complex models. Since in this work we consider only the networks of type 1-M-1 (only one hidden layer) in the rest of the paper we will denote the evidence $\mathcal{P}\left(\mathcal{D}|\,\mathcal{A}_M\right)$ by $\mathcal{P}\left(\mathcal{D}|\,M\right)$.

3.2 Prior function

We have already mentioned that the various possible prior distributions are considered in the literature [69]. In this analysis the likelihood function is given by χ^2 distribution, which has a Gaussian probabilistic interpretation. Therefore it seems to be reasonable to assume that the weight parameters distribution should also be described by the Gaussian-like prior function. Additionally we assume, without losing the generality, that:

- negative, and positive values of the weight parameters are equally likely;

- at the beginning of the learning procedure the weight parameters are independent;

- small[3] weight values are more likely than the large values.

Then the Gaussian-like prior distribution can have a form:

$$\mathcal{P}\left(\vec{w}|\,\alpha,\mathcal{A}_M\right) \sim \exp\left[-\frac{1}{2}\sum_{i=1}^{W}\alpha_i w_i^2\right]. \qquad (3.23)$$

Notice that every w_i parameter has its own α_i regularization parameter. As it was mentioned in the previous section the α is the so-called scale parameter. The number of the

---

[2] It is the consequence of the fact that α is the scale parameter.

[3] For the networks with the sigmoid activation functions the non-trivial smooth functional parametrization are described by the low $|w_i|$ weights.

scale parameters can be reduced if the symmetry property of the given network architecture is taken into account. The network of the type 1-M-1 has: M hidden weights; M corresponding bias weights and $M + 1$ linear weighs (output weights + one bias parameter). The permutation between the hidden units does not change the network functional type. Permuting two hidden units is realized by exchange between the weight parameters of the same type (hidden, bias and linear weights). This symmetry property allows us to reduce the number of α's to three independent scale parameters:

- α_h for the hidden weights;

- α_b for bias weights (in hidden layer);

- α_l for linear weights in the output layer.

Then the prior function reads

$$\mathcal{P}\left(\vec{w}\middle|\,\alpha, \mathcal{A}_M\right) \sim \exp\left[-\frac{1}{2}\left(\alpha_h \sum_{i\in\text{hidden}} w_i^2 + \alpha_b \sum_{i\in\text{bias}} w_i^2 + \alpha_l \sum_{i\in\text{linear}} w_i^2\right)\right]. \qquad (3.24)$$

We made an effort to compare results which are obtained with both (3.9) and (3.24) priors. It was observed that final results are very similar. Analogically as in the case of (3.9) prior the α_h, α_b and α_l parameters were iteratively changed during the training procedure. The typical results, obtained for the $G_{Mn}/\mu_n G_D$ and G_{En} data sets, are shown in figure 7. The differences between the final best fits are negligible. In the left column of the same figure we plot the dependence of the $S(\vec{w}, \mathcal{D})$ on the iteration step. We see that the minimal value of the total error is almost the same for both prior functions. For both cases the training started from the same initial weight configuration.

All above seem to justify the simplest choice of the prior function, namely the one given by eq. (3.9). Nevertheless, it may happen that for more complex data then we discuss, the results will significantly depend on prior assumptions. In such case the Bayesian framework can be used to indicate the best prior function.

Eventually, we discuss the dependence of the final results on the initial α_0 value. We considered several initial values of α_0 (see table 1). After training we noticed that the choice of the initial α_0 had a small impact on the final α_{MP} value (see figure 8) as well as the fits. It is shown in table 1 where the relative distances, in the weight space, between fits are presented. Notice that the only one solution computed for $\alpha_0 = 1$ is out of others.

It is worth to mention that decreasing the α_0 parameter can be understood as enlarging the effective prior domain. For the final analysis we set $\alpha_0 = 0.001$.

In this section we have demonstrated that our results weakly depend on the prior assumptions. It has been also shown that it is relatively easy to construct the prior function if the symmetry properties of network are taken into consideration. Usually, it is not the case in the conventional form-factor data analysis, where the ad-hoc parametrizations are discussed. The typical phenomenological parametrization has no straightforward symmetries. As an example consider the function [32, 70]:

$$G(Q^2) = \frac{a_0 + a_1 Q^2 + a_2 Q^4}{b_0 + b_1 Q^2 + b_2 Q^4 + b_3 Q^6 + b_4 Q^8}. \qquad (3.25)$$
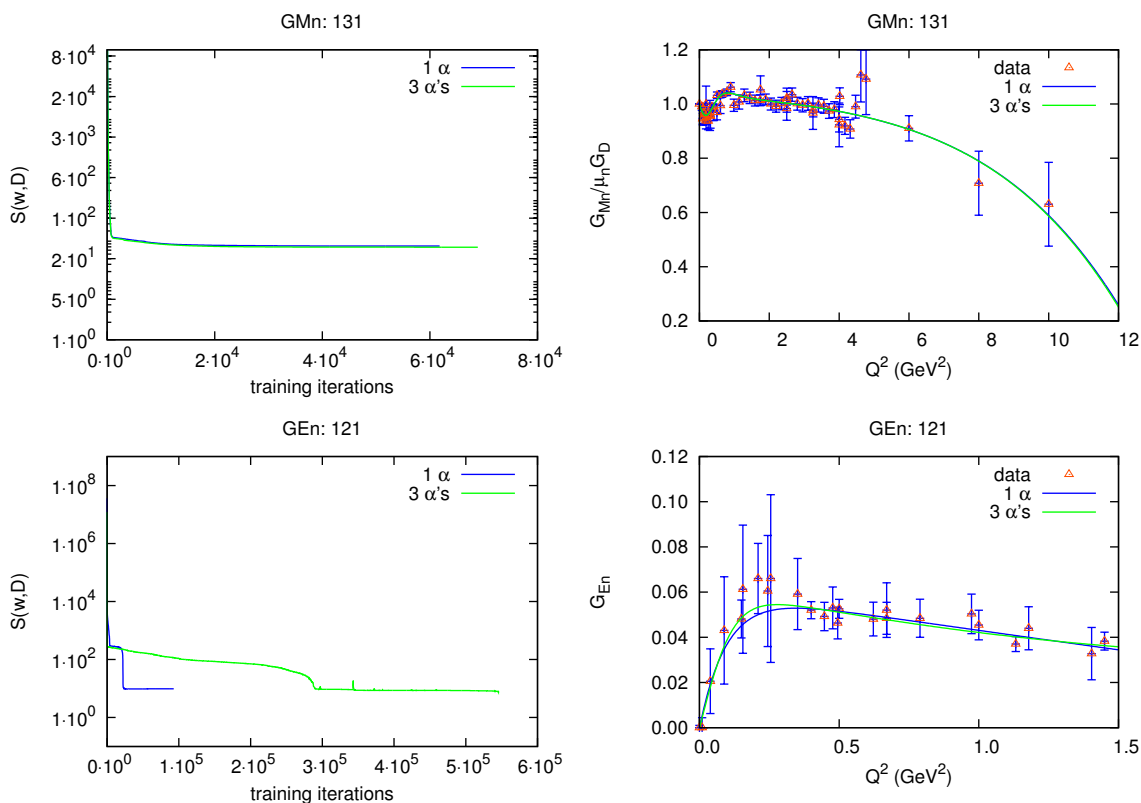
Figure 7. Left panels: $S(\vec{w}, \mathcal{D})$ dependence on the iteration step. Right panels: the best fits obtained for $G_{Mn}/\mu_n G_D$ and G_{En} data. The results obtained with (3.24) prior are denoted by green lines, while the results computed for the (3.9) prior function are plotted with blue lines. For the magnetic neutron data the network of 1-3-1 type was trained. The electric neutron data was analyzed with 1-2-1 network type.

α_0	0.0001	0.001	0.01	0.1	1
0.0001	0	0.0925	0.0196	0.8048	14.8847
0.0010	0.0925	0	0.0748	0.8921	14.9641
0.0100	0.0196	0.0748	0	0.8214	14.8965
0.1000	0.8048	0.8921	0.8214	0	14.2768
1.0000	14.8847	14.9641	14.8965	14.2768	0

Table 1. The distance $d(\vec{w}_1, \vec{w}_2) = \sqrt{\sum_{i=1}^{W}(w_{1i} - w_{2i})^2}$ between fits obtained for various initial α_0 values. The computations are done for the G_{En} data for the network of 1-2-1 type.

Constructing the prior function for above form-factor parametrization seems to be more complicated than in the ANN case. One can postulate the values of the ratios a_0/b_0 and a_2/b_4, which describe the low and high Q^2 behavior of the FF. However, the rest of parameters, which seem to model the intermediate Q^2 region, can have any arbitrary values. Therefore building the prior distribution for above FF would require an extra phenomenological and theoretical knowledge.
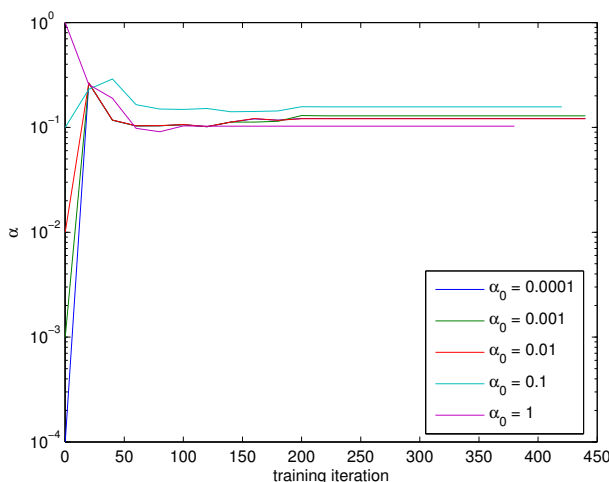
Figure 8. Dependence of iteration of α parameter on the initial α_0 value. The results were obtained for the 1-2-1 network type trained with G_{En} data.

4 Form-factor fits

4.1 Data

We consider the electric and magnetic proton and neutron form-factor data. The electric and magnetic nucleon form-factors are defined as follows:

$$G_{Mp,n}(Q^2) = F_1^{p,n}(Q^2) + F_2^{p,n}(Q^2), \tag{4.1}$$

$$G_{Ep,n}(Q^2) = F_1^{p,n}(Q^2) - \frac{Q^2}{4M^2} F_2^{p,n}(Q^2), \tag{4.2}$$

where:

$$G_{Mp,n} = \mu_{p,n}, \quad G_{Ep} = 1, \quad G_{En} = 0. \tag{4.3}$$

The experimental data is usually normalized to the dipole form-factor $G_D = 1/(1 + Q^2/0.71)^2$.

The electric G_{Ep} and magnetic G_{Mp} proton FF data have been obtained via Rosenbluth separation technique from elastic ep scattering [71]. Additionally since the beginning of nineties of last century the measurement of the form-factor ratio $\mu_p G_{Ep}/G_{Mp}$ in the spin dependent elastic ep scattering have been performed [72].

It turned out that systematic discrepancy exists between so-called *Rosenbluth* and *polarization transfer* $\mu_p G_{Ep}/G_{Mp}$ ratio data. The difference can be explained when the two photon exchange effect (TPE) [73–76] is taken into account (for review see [77]). Hence, a proper fit of the EM form-factors requires to take into account the TPE correction [37]. In this work we consider the re-analyzed (TPE corrected Rosenbluth) $G_{Mp}/\mu_p G_D$ and G_{Ep}/G_D data (tables 2 and 3 of ref. [70]). However, to see the TPE effect we consider also the original, (called here *old Rosenbluth data*) $G_{Mp}/\mu_p G_D$ [71, 78–86] and G_{Ep}/G_D [71, 78–84, 87, 88] data sets.[4] The neutron form-factor data (G_{En} and G_{Mn}) are obtained from

---

[4]We used the JLab data-base [89].

the electron scattering off light nuclei (deuteron [90], helium [91]). Since the complexity of nuclear target, getting nucleon form-factors is more demanding than in the case of the elastic ep scattering. The ground and final states of the nucleon must be properly described. In this analysis we consider the same G_{En} and $G_{Mn}/\mu_n G_D$ data sets as in ref. [37].

Let us mention that to obtain proper fits of the form-factors at $Q^2 = 0$ we added to every data set one artificial point, namely ($Q^2 = 0, t = 1, \Delta t = 0.001$) for $G_{Mn}/\mu_n G_D$, $G_{Mp}/\mu_p G_D$ and G_{Ep}/G_D data sets, and ($Q^2 = 0, t = 0, \Delta t = 0.001$) for G_{En} data set. This constraints have an effect on the final fit value and the uncertainty only in the close surrounding of the added point, as it is shown in table 2, where we present how the best fit values and its uncertainties depend on the artificial point error. We present results for G_{Mn} data but for other considered data sets we got analogical conclusions. The Δt value assigned to the additional point should be comparable to data uncertainties used in the network training. We have found that using $\Delta t = 0.01$ and higher is not sufficient to attract the fit to desired value at constraint point, while $\Delta t = 0.0001$ causes numerical difficulties during the training since the point has dominant contribution to the overall network error value.

error of artificial point (Δt)	$G_{Mn}/\mu_n G_D$	error
Q^2=0		
1.0000	1.01809	0.03258
0.1000	1.01633	0.03097
0.0100	1.00198	0.00947
0.0010	1.00002	0.00075
0.0001	1.00000	0.00009
Q^2=0.1		
1.0000	0.96920	0.00700
0.1000	0.96893	0.00696
0.0100	0.96709	0.00642
0.0010	0.96650	0.00624
0.0001	0.96986	0.00599
Q^2=1.0		
1.0000	1.03657	0.00792
0.1000	1.03669	0.00797
0.0100	1.03720	0.00815
0.0010	1.03778	0.00813
0.0001	1.03575	0.00773

Table 2. Dependence of $G_{Mn}/\mu_n G_D$ and its uncertainty (computed for Q^2 =0, 0.1, and 1) on the Δt of the artificial point added at $Q^2 = 0$.

4.2 Numerical procedure

The numerical analysis was done with two independent neural network softwares (in order to cross-validate the results). One written by R.S. and P.P. [54] and another, which has been developed by K.M.G. [92].

The procedure for finding the best neural network model for each data set consists of the five major steps:

1. the sequence of networks of different 1-M-1 type (1-1-1, 1-2-1, 1-3-1, ... etc.) is taken into consideration;

2. for each network of 1-M-1 type the sample of networks with randomly initiated weights is trained;

3. among the networks obtained in the previous step, ten networks with the lowest total error are selected for further analysis, see section 2.2 and the $S(\vec{w}_{MP}, \mathcal{D})$ distributions shown in figures 5 and 6;

4. the network (from the step above) with the highest evidence is chosen as the best fit candidate for given network type;

5. the best fits obtained for every network type are compared; the one with the highest evidence is chosen to represent the data.

Let us remind that in the second step the large number (from 150 to 1300) of networks in the sample (as it is explained in section 2.2) is considered in order to find the solutions which maximizes the posterior probability for the given model.

The procedure for the single network training is as follows (see figure 9):

- initialize the network weights as small random values;

- initialize the regularization factor (eq. (2.7)), in this analysis $\alpha_0 = 0.001$;

- perform the network training iterations, according to the *Levenberg-Marquardt*, *quick-prop*, or *rprop* algorithms;

- calculate the updated regularization factor α_{k+1} (eq. (3.19)) every 20 iterations of the training algorithm; eigenvalues of Hessian matrix below 10^{-6} are rejected from the evaluation of $\gamma(\alpha_k)$ (eq. (3.18));

- calculate the network output (eq. (2.3)) and uncertainty (eq. (3.16)) values for the given range of Q^2 values;

- calculate the ln of evidence (eq. (3.22))

Eventually, we will shortly highlight the major differences between the NNPDF approach and the one presented in this article.

In this work we consider the sequence of networks with graded number of hidden units. With the help of the Bayesian framework the best solution is chosen. The NNPDF group considers one particular network architecture (2-5-3-1 type) to fit the data [48]. But some discussion of the dependence of results on the network architecture is presented.

The NNPDF group prepares the sample of the networks. Each network from the sample is trained with the artificial data which is Monte Carlo generated from the original measurements. Then the best fit and its uncertainty are obtained as an average and standard deviations computed over the sample. In this work every network is always trained with the original data set. Nevertheless the large sample of networks of given type is prepared but in order to find the architecture and the weights which maximize the evidence. The network response uncertainty is computed from the covariance matrix (eq. (3.16)).

Both approaches deal with the over-fitting problem but in different ways. The NNPDF applies the early stopping in the training (cross-validation algorithm is imposed). Whereas we consider the regularization penalty term in the error function, which is optimized by the Bayesian procedure. Hence the approach we apply does not require validation of the solutions by comparing with the test data set.
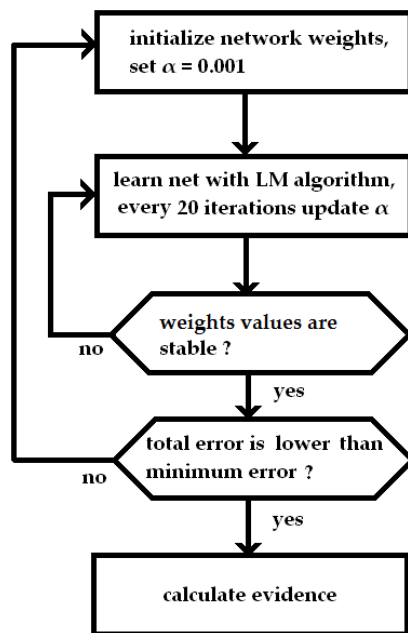
Figure 9. Learning schema.

4.3 Numerical results

The numerical procedures described in the previous section were applied to all (six) the data sets. We consider networks with $M = 1 - 5$ hidden units for G_{Mn}, G_{En}, and G_{Ep} data and with $M = 1 - 6$ for the G_{Mp} data. The evidence quantitatively classifies the networks i.e. the most suitable network architecture for representing the data is indicated by the maximum of the evidence. Notice that the optimal way to deal with these results would be taking an average over all solutions weighted by the evidence. However, in all problems considered here we obtained clear signal (a peak at the evidence) for particular solution. It allowed us to neglect the contribution from networks of other size.

We start the presentation of the numerical results by the discussion of the $G_{Mn}/\mu_n G_D$ FF data. As it was described above, we consider a set of networks, which differ by number of hidden units M. In figure 10 we show the scatter plot presenting the dependence of given network size on error function and log of evidence. One can notice that the networks 1-2-1 and 1-3-1 have the highest evidences, but the networks with $M = 2$ hidden units are not able to reproduce as low total error value as 1-3-1 networks. It is interesting also to mention that for $M \geq 3$ the total error slowly varies, i.e. increasing the number of the hidden units lowers the total error by the minor amount. The clear indication for 1-3-1 network type is seen in figure 11, where only dependence of $\ln \mathcal{P}\left(\mathcal{D}|\,M\right)$ on M is shown. In this figure we plot the maximal evidences obtained for given network type. However, in order to control the stability of numerical procedure we plot also the ln of evidence averaged over the networks around global minimum (solutions selected in step 3, section 4.2), as well as the ln of the minimal values of $\mathcal{P}(\mathcal{D}\,|\mathcal{M})$.
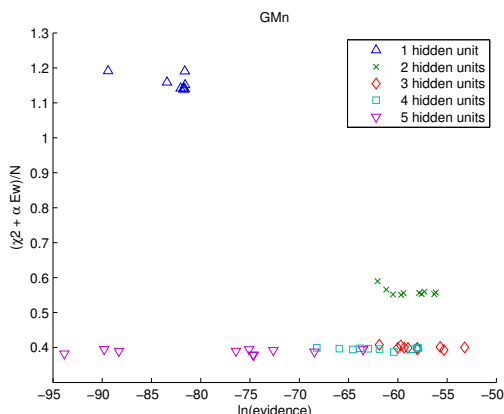
Figure 10. The total error, $S(\vec{w}_{MP})$, as a function of $\ln \mathcal{P}\left(\mathcal{D}\mid M\right)$ (ln evidence). The evidence is computed for networks trained with $G_{Mn}/\mu_n G_D$ data. The results obtained for networks with $M = 1 - 5$ hidden units are shown. Single point represents the fit obtained for given starting weight configuration and particular network type.
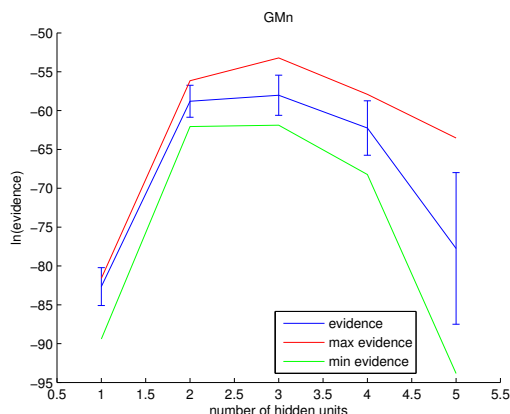
Figure 11. The dependence of $\ln \mathcal{P}\left(\mathcal{D}\mid M\right)$ on the number of hidden units. The evidence is computed for networks trained with $G_{Mn}/\mu_n G_D$ data. The maximal and minimal values of $\ln \mathcal{P}\left(\mathcal{D}\mid M\right)$ (for given network type) are plotted with the red and green lines respectively. The mean of $\ln \mathcal{P}\left(\mathcal{D}\mid M\right)$ over all acceptable solutions is represented by the blue line.

All together suggest the network of type 1-3-1 (with the highest evidence) for the best fit of the G_{Mn} data. The network output is drawn in figure 12 together with the experimental data. The neural network response uncertainty is computed with (3.16) expression and shown in figure 13. In figure 12 we plot also the best fits obtained for networks: 1-1-1, 1-2-1, 1-4-1 and 1-5-1. As could be expected increasing the number of hidden units makes the fit more flexible.

The electric neutron FF data (G_{En}) is analyzed in the same way as the magnetic neutron one. In figures 14, 15 and 16 the plots of evidence and G_{En} form-factor are shown. For $M = 2$ we obtained the peak of the Occam's hill, what indicates 1-2-1 network architecture as the most representative parametrization.

The results for the electric and magnetic FF data are presented in figures 17, 18 and 21, 22 (scatter and evidence plots) and figures 19 and 20 (form-factor plots). The network of type 1-3-1 is preferred by the both electric and magnetic data sets. As it has been mentioned above we analyzed also the old form-factor data, which are not TPE corrected. It was obtained that the old G_{Ep} prefers representation by the network of type 1-1-1. Hence, the old Rosenbluth G_{Ep}/G_D data fit is almost linear constant function in Q^2. But the data seems to be not conclusive enough, so the Bayesian procedure leads to the simplest possible solution. On the other hand, it means that the old proton electric data does not show clear indication for deviation from the dipole form.
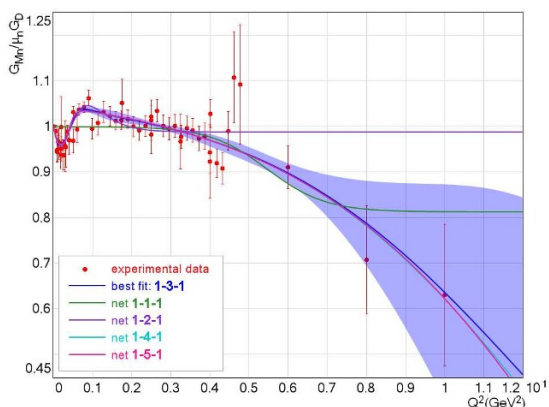
Figure 12. Fits of the $G_{Mn}/\mu_n G_D$ data parametrized with networks of 1-1-1 (green line), 1-2-1 (violet line), 1-3-1 (blue line), 1-4-1 (cyan line) and 1-5-1 (magenta line) types. The best fit (shown with 1σ uncertainty), which was indicated by the maximal evidence, is given by 1-3-1 network. The blue area denotes fit uncertainty computed with (3.16). The experimental data is the same as the one discussed in ref. [37].
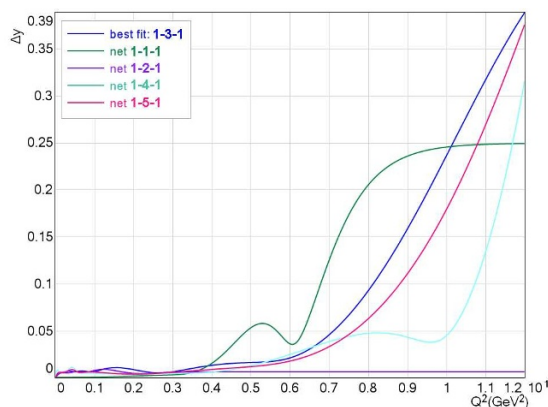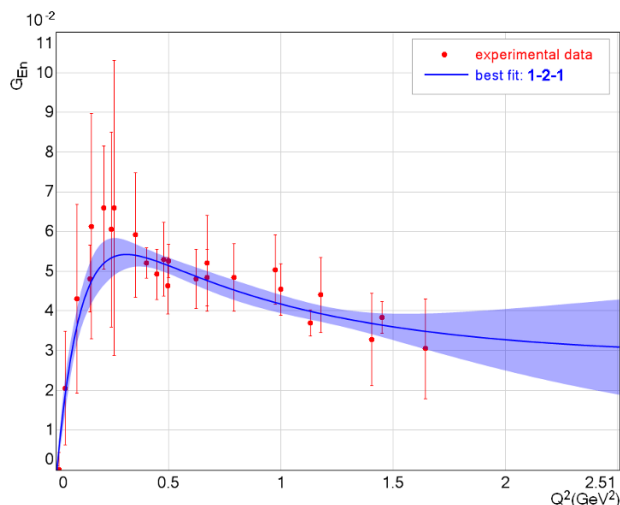

Figure 13. The fit uncertainty computed (with eq. (3.16)) for the parametrizations shown in figure 12.
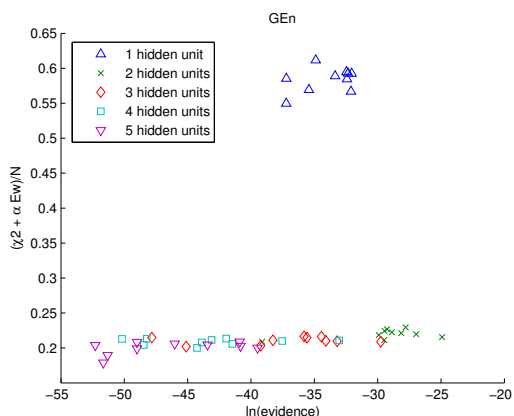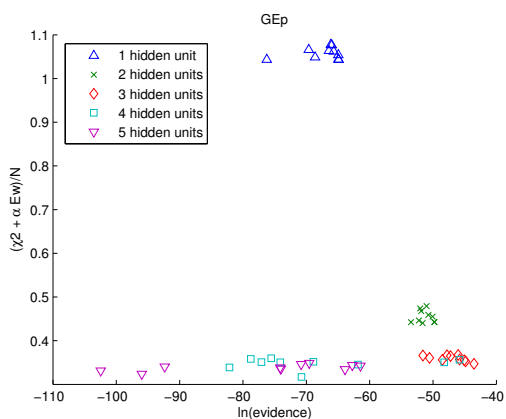

Figure 14. The total error, $S(\vec{w}_{MP})$, as a function of $\ln \mathcal{P}(\mathcal{D}|M)$ (ln evidence). The evidence is computed for networks trained with the G_{En} data. The results obtained for networks with $M = 1 - 5$ hidden units are shown. Single point represents the fit obtained for given starting weight configuration and particular network type.
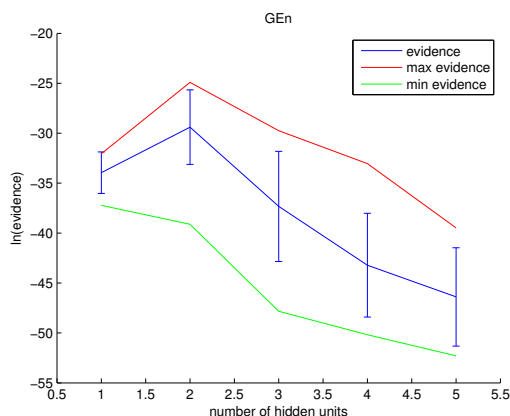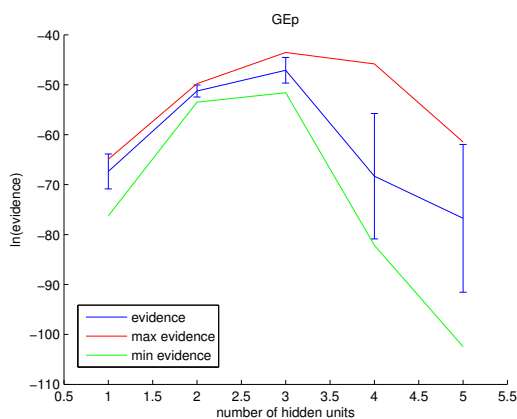

Figure 15. The dependence of $\ln \mathcal{P}(\mathcal{D}|M)$ on the number of hidden units. The evidence is computed for networks trained with the G_{En} data. The maximal and minimal values of $\ln \mathcal{P}(\mathcal{D}|M)$ (for given network type) are plotted with the red and green lines respectively. The mean of $\ln \mathcal{P}(\mathcal{D}|M)$ over all acceptable solutions is represented by the blue line.

Figure 16. The best fit of G_{En} data given by the 1-2-1 network. The blue area denotes fit uncertainty computed with eq. (3.16). The experimental data is the same as the one discussed in ref. [37].

Figure 17. The total error, $S(\vec{w}_{MP})$, as a function of $\ln \mathcal{P}(\mathcal{D}|\,M)$ (ln evidence). The evidence is computed for networks trained with the G_{Ep}/G_D data. The results obtained for networks with $M = 1 - 5$ hidden units are shown. Single point represents the fit obtained for given starting weight configuration and particular network type.

Figure 18. The dependence of $\ln \mathcal{P}(\mathcal{D}|\,M)$ on the number of hidden units. The evidence is computed for networks trained with the G_{Ep}/G_D data. The maximal and minimal values of $\ln \mathcal{P}(\mathcal{D}|\,M)$ (for given network type) are plotted with the red and green lines respectively. The mean of $\ln \mathcal{P}(\mathcal{D}|\,M)$ over all acceptable solutions is represented by the blue line.

Figure 19. The best fit of G_{Ep}/G_D data. The fit to TPE corrected data is given by 1-3-1 network (blue line), the data (red points) is taken from [70]. The fit to "old Rosenbluth data" (green points) is given by 1-1-1 network (violet line), the data is taken from [71, 78–84, 87, 88]. The fit uncertainty is computed with eq. (3.16).
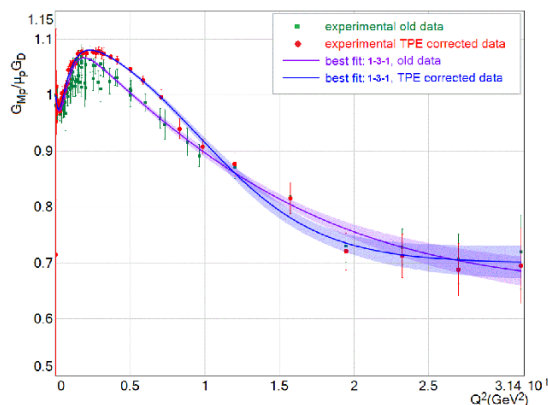
Figure 20. The best fit of $G_{Mp}/\mu_p G_D$ data given by the 1-3-1 network. The fit to TPE corrected data is given by 1-3-1 network (violet line), the data (red points) is taken from [70]. The fit to "old Rosenbluth data" (green points) is given by 1-1-1 network (violet line), the data is taken from [71, 78–86]. The fit uncertainty is computed with eq. (3.16).
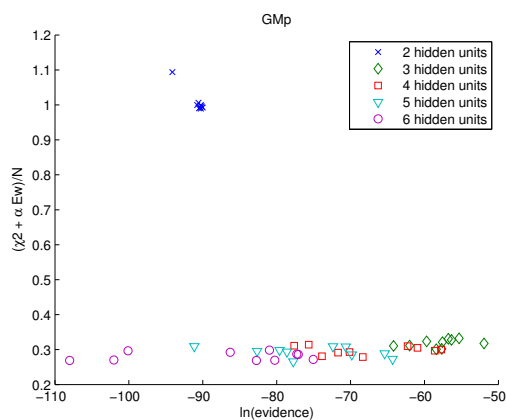
Figure 21. The total error, $S(\vec{w}_{MP})$, as a function of $\ln\mathcal{P}(\mathcal{D}|\,M)$ (ln evidence). The evidence is computed for networks trained with $G_{Mp}/\mu_p G_D$ data. The results obtained for networks with $M = 1 - 6$ hidden units are shown. Single point represents the fit obtained for given starting weight configuration and particular network type.
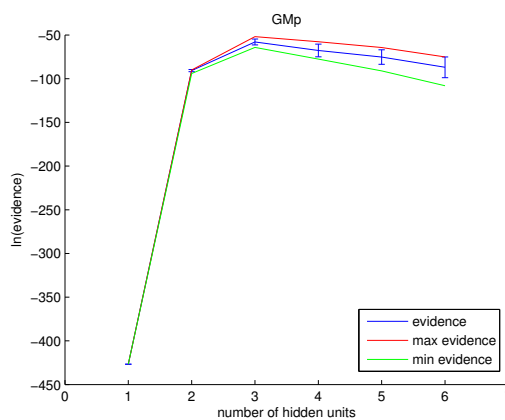
Figure 22. The dependence of $\ln\mathcal{P}(\mathcal{D}|\,M)$ on the number of hidden units. The evidence is computed for networks trained with $G_{Mp}/\mu_p G_D$ data. The maximal and minimal values of $\ln\mathcal{P}(\mathcal{D}|\,M)$ (for given network type) are plotted with the red and green lines respectively. The mean of $\ln\mathcal{P}(\mathcal{D}|\,M)$ over all acceptable solutions is represented by the blue line.

4.4 Summary

We have analyzed the form-factor data by the means of the artificial neural networks. The Bayesian approach has been adapted for the χ^2 minimization and then applied to the data analysis. For every form-factor data set sequence of neural networks have been considered. The Bayesian approach provided us with an objective criteria for choosing the most suitable form-factor parametrization (neural network) with the statistically optimal balance of the fit complexity and its uncertainty. Therefore the resulting fits are unbiased and model independent. It has been demonstrated also that the final results weakly depend on the prior assumptions.

The approach allowed to investigate objectively the non-dipole deviations of the form-factors. It is interesting to mention that the G_{Ep}/G_D, $G_{Mp}/\mu_p G_D$ as well as $G_{Mn}/\mu_n G_D$ form-factor data prefer the same type (size) network 1-3-1. The form-factor parametrizations, obtained in this analysis can be easily applied to any phenomenological and experimental analysis. Additionally, a part of the our software used in the analysis is available at [54, 92].

Presented method seems to be a promising statistical framework for studying and representing the experimental data. Especially, if the theoretical predictions are not able to reproduce measurements with desired accuracy, but the experimental data is sufficiently comprehensive to describe physical quantity by itself.

Acknowledgments

A   Analytical formulae

The two parametrizations of the form-factors have been obtained. The network of the type 1-2-1 representing G_{En}:

$$G_{En}(Q^2) = w_5 f_{act}(Q^2 w_1 + w_2) + w_6 f_{act}(Q^2 w_3 + w_4) + w_7, \tag{A.1}$$

and the network of the type 1-3-1, representing $G_{Mn}/\mu_n G_D$, G_{Ep}/G_D and $G_{Mp}/\mu_p G_D$:

$$G_f(Q^2)/gG_D = w_7 f_{act}(Q^2 w_1 + w_2) + w_8 f_{act}(Q^2 w_3 + w_4) + w_9 f_{act}(Q^2 w_5 + w_6) + w_{10},$$
$$f = Mm, Ep, Mp, \tag{A.2}$$

where $g = 1$ for proton electric form-factor and $g = \mu_{p,n}$ for the proton, neutron magnetic form-factors. The activation function reads

$$f_{act}(x) = \frac{1}{1 + \exp(-x)}. \tag{A.3}$$

The weights obtained for G_{En}:

$$\vec{w}_{MP}^T = (10.19704, 2.36812, -1.144266, -4.274101, 0.8149924, 2.985524, -0.7864434) \tag{A.4}$$

with the covariance matrix:

$$A^{-1} = \begin{pmatrix}
77182.936 & -76674.953 & 11320.149 & -976.911 & -59149.683 & -510.459 & 59023.698 \\
-141838.399 & 158041.683 & -17763.896 & 1808.806 & 121039.907 & 875.737 & -120845.155 \\
1007.74 & 1987.396 & 2153.904 & 94.542 & 1216.369 & 99.514 & -1244.23 \\
-881.971 & 1138.085 & 154.164 & 2325.543 & 841.299 & -6673.131 & -844.274 \\
-106233.935 & 117881.485 & -13555.199 & 1345.44 & 90325.25 & 660.27 & -90176.259 \\
-524.981 & -282.957 & -492.929 & -6713.68 & -132.119 & 19769.326 & 138.851 \\
106231.986 & -117915.687 & 13528.692 & -1347.504 & -90347.707 & -661.274 & 90199.073
\end{pmatrix}. \tag{A.5}$$

The weights obtained for $G_{Mn}/\mu_n G_D$:

$$\vec{w}_{MP}^T = (3.19646, 2.565681, 6.441526, -2.004055, -0.2972361, 3.606737, -3.135199, 0.299523, 1.261638, 2.64747) \tag{A.6}$$

with the covariant matrix:

$$A^{-1} = \begin{pmatrix}
13019.47 & 5437.135 & 1625.832 & 2407.977 & 2421.111 & -9226.711 & -5508.625 & -466.761 & 11858.122 & -5018.992 \\
-110.632 & 2389.64 & -1419.064 & 1007.869 & 68.926 & 748.578 & -8134.262 & 132.414 & 1320.543 & 6692.985 \\
1186.145 & 1096.726 & 5283.129 & -2368.423 & -32.076 & 97.757 & 331.547 & -371.507 & -157.415 & 188.748 \\
2412.026 & 476.941 & -2382.018 & 2014.753 & 486.682 & -1841.165 & -1386.815 & 128.242 & 2393.05 & -961.438 \\
1688.083 & 877.17 & -114.146 & 433.604 & 445.447 & -1374.196 & -1269.078 & -43.863 & 2404.463 & -943.785 \\
-15867.205 & -7087.878 & -406.067 & -3161.81 & -3587.665 & 16599.626 & 7510.982 & 544.902 & -14185.447 & 4857.133 \\
9913.113 & -1376.838 & 7840.949 & -2871.111 & 1670.017 & -9080.441 & 18045.64 & -1025.567 & 5532.985 & -21923.165 \\
-424.63 & -308.244 & -386.653 & 125.67 & -74.709 & 273.379 & 250.767 & 48.032 & -378.848 & 53.352 \\
-824.755 & 638.712 & -1287.206 & 628.491 & 250.047 & 4601.007 & -3463.562 & 142.212 & 6594.3 & -3256.938 \\
-7934.871 & 1405.406 & -6188.958 & 2288.342 & -1669.7 & 3594.294 & -15306.448 & 813.979 & -10852.434 & 24785.914
\end{pmatrix}. \tag{A.7}$$

The weights obtained for G_{Ep}/G_D:

$$\vec{w}_{MP}^T = (3.930227, 0.1108384, -5.325479, -2.846154, -0.2071328, 0.8742101, 0.4283194, 2.568322, 2.577635, -1.185632) \tag{A.8}$$

with the covariance matrix:

$$A^{-1} =$$
$$\begin{pmatrix}
36866.41 & -52184.005 & 17354.195 & -9375.943 & 693.571 & 7949.39 & -16875.298 & -11986.299 & 10541.393 & 4687.308 \\
-68432.176 & 103227.83 & -25251.786 & 22514.051 & -1329.157 & -14150.394 & 34458.11 & 24954.878 & -19958.73 & -11910.498 \\
14227.233 & -16215.276 & 26518.973 & 2749.674 & 160.818 & 2066.422 & -2921.857 & 2228.451 & 2430.186 & -38.278 \\
-35928.337 & 57408.998 & -6198.14 & 18394.036 & -745.922 & -7749.442 & 20171.669 & 8522.062 & -11194.428 & -7642.097 \\
181.767 & -354.001 & 27.284 & -99.739 & 55.716 & -65.584 & -118 & -98.718 & 696.136 & -337.281 \\
6881.763 & -8970.135 & 2549.152 & -1714.537 & 128.027 & 2720.834 & -3012.463 & -2059.728 & 2495.438 & -329.115 \\
-23847.659 & 36615.2 & -6410.826 & 8820.603 & -475.037 & -5101.518 & 12518.641 & 9338.177 & -7159.334 & -4418.53 \\
22789.127 & -35227.846 & 11996.228 & -14431.928 & 441.824 & 4699.586 & -11630.739 & 8794.696 & 6644.126 & 4132.854 \\
3700.817 & -6332.644 & 791.595 & -1652.194 & 760.047 & -46.633 & -2130.248 & -1683.296 & 9852.346 & -4813.821 \\
17126.803 & -26781.151 & 4320.733 & -6631.983 & -137.16 & 3552.551 & -9216.18 & -6918.134 & -1252.964 & 7966.577
\end{pmatrix}. \tag{A.9}$$

The weights obtained for $G_{Mp}/\mu_p G_D$:

$$\vec{w}_{MP}^T = (-2.862682, -1.560675, 2.321148, 0.1283189, -0.2803566, 2.794296, 1.726774, 0.861083, 0.4184286, -0.1526676) \tag{A.10}$$

with the covariant matrix:

$$A^{-1} = \begin{pmatrix}
15709.171 & 6861.227 & 2766.185 & -6126.712 & -121.495 & 1318.866 & 8737.945 & 4008.92 & -94.694 & -3978.556 \\
3284.282 & 2803.079 & 843.705 & -1333.839 & -40.306 & 438.59 & 817.679 & 1156.474 & -31.955 & -1145.984 \\
1993.96 & 1142.807 & 495.778 & -954.548 & -31.697 & 341.671 & 836.303 & 462.66 & -25.013 & -454.17 \\
-3841.3 & -1694.722 & -859.519 & 2450.449 & 47.229 & -515.322 & -1122.054 & -167.405 & 37.338 & 155.266 \\
-88.252 & -54.872 & -31.981 & 50.267 & 8.865 & -76.635 & -33.677 & -16.192 & 6.739 & 12.676 \\
935.495 & 585.08 & 339.546 & -538.917 & -76.046 & 720.701 & 349.457 & 169.075 & -53.896 & -145.686 \\
14256.242 & 5279.888 & 2235.735 & -4498.246 & -93.725 & 1012.932 & 9909.287 & 4366.293 & -72.416 & -4343.104 \\
5227.654 & 2640.608 & 867.323 & -1228.301 & -32.179 & 348.563 & 3580.716 & 2148.384 & -24.917 & -2140.484 \\
-67.875 & -43.08 & -25.232 & 39.496 & 6.797 & -54.859 & -24.96 & -12.175 & 6.271 & 8.358 \\
-5204.745 & -2625.836 & -858.408 & 1214.946 & 28.58 & -324.213 & -3572.214 & -2144.257 & 21.06 & 2139.095
\end{pmatrix}. \tag{A.11}$$

References

[1] F.E. Close, A. Donnachie and G. Shaw, *Electromagnetic Interactions and Hadronic Structure*, Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology, Cambridge (2007).

[2] G. Hohler et al., *Analysis of Electromagnetic Nucleon Form-Factors*, *Nucl. Phys.* **B 114** (1976) 505 [SPIRES].

[3] E.L. Lomon, *Extended Gari-Kruempelmann model fits to nucleon electromagnetic form factors*, *Phys. Rev.* **C 64** (2001) 035204 [nucl-th/0104039] [SPIRES].

[4] E.L. Lomon, *Effect of recent R(p) and R(n) measurements on extended Gari-Kruempelmann model fits to nucleon electromagnetic form factors*, *Phys. Rev.* **C 66** (2002) 045501 [nucl-th/0203081] [SPIRES].

[5] C. Crawford et al., *The Role of Mesons in the Electromagnetic Form Factors of the Nucleon*, arXiv:1003.0903 [SPIRES].

[6] M.A. Belushkin, H.W. Hammer and U.G. Meissner, *Dispersion analysis of the nucleon form factors including meson continua*, *Phys. Rev.* **C 75** (2007) 035202 [hep-ph/0608337] [SPIRES].

[7] G.A. Miller, *Light front cloudy bag model: Nucleon electromagnetic form factors*, *Phys. Rev.* **C 66** (2002) 032201 [nucl-th/0207007] [SPIRES].

[8] F. Cardarelli and S. Simula, *SU(6) breaking effects in the nucleon elastic electromagnetic form factors*, *Phys. Rev.* **C 62** (2000) 065201 [nucl-th/0006023] [SPIRES].

[9] R.F. Wagenbrunn, S. Boffi, W. Klink, W. Plessas and M. Radici, *Covariant nucleon electromagnetic form factors from the Goldstone-boson exchange quark model*, *Phys. Lett.* **B 511** (2001) 33 [nucl-th/0010048] [SPIRES].

[10] M.M. Giannini, E. Santopinto and A. Vassallo, *An overview of the hypercentral constituent quark model*, *Prog. Part. Nucl. Phys.* **50** (2003) 263 [nucl-th/0301017] [SPIRES].

[11] C.F. Perdrisat, V. Punjabi and M. Vanderhaeghen, *Nucleon electromagnetic form factors*, *Prog. Part. Nucl. Phys.* **59** (2007) 694 [hep-ph/0612014] [SPIRES].

[12] G.A. Miller, *Transverse Charge Densities*, arXiv:1002.0355 [SPIRES].

[13] L. Álvarez-Ruso, *Theoretical highlights of neutrino-nucleus interactions*, *AIP Conf. Proc.* **1222** (2010) 42 [arXiv:0911.4112] [SPIRES].

[14] W.M. Alberico, S.M. Bilenky and C. Maieron, *Strangeness in the nucleon: Neutrino nucleon and polarized electron nucleon scattering*, *Phys. Rept.* **358** (2002) 227 [hep-ph/0102269] [SPIRES].

[15] D.H. Beck and B.R. Holstein, *Nucleon structure and parity-violating electron scattering*, *Int. J. Mod. Phys.* **E 10** (2001) 1 [hep-ph/0102053] [SPIRES].

[16] K2K collaboration, M.H. Ahn et al., *Measurement of Neutrino Oscillation by the K2K Experiment*, *Phys. Rev.* **D 74** (2006) 072003 [hep-ex/0606032] [SPIRES].

[17] Y. Hayato et al., *Neutrino Oscillation Experiment at JHF*, letter of intent to the JPARC 50 GeV proton synchrotron, Jan. 21, 2003, http://neutrino.kek.jp/jhfnu/loi/loi_JHFcor.pdf

[18] C.H. Llewellyn Smith, *Neutrino Reactions at Accelerator Energies*, *Phys. Rept.* **3** (1972) 261 [SPIRES].

[19] K2K collaboration, R. Gran et al., *Measurement of the quasi-elastic axial vector mass in neutrino oxygen interactions*, *Phys. Rev.* **D 74** (2006) 052002 [hep-ex/0603034] [SPIRES].

[20] MiniBooNE collaboration, A.A. Aguilar-Arevalo et al., *Measurement of muon neutrino quasi-elastic scattering on carbon*, *Phys. Rev. Lett.* **100** (2008) 032301 [`arXiv:0706.0926`] [SPIRES].

[21] V. Bernard, L. Elouadrhiri and U.G. Meissner, *Axial structure of the nucleon*, *J. Phys.* **G 28** (2002) R1 [`hep-ph/0107088`] [SPIRES].

[22] K.S. Kuzmin, V.V. Lyubushkin and V.A. Naumov, *Quasielastic axial-vector mass from experiments on neutrino-nucleus scattering*, *Eur. Phys. J.* **C 54** (2008) 517 [`arXiv:0712.4384`] [SPIRES].

[23] A. Bodek, S. Avvakumov, R. Bradford and H. Budd, *Extraction of the Axial Nucleon Form Factor from Neutrino Experiments on Deuterium*, *J. Phys. Conf. Ser.* **110** (2008) 082004.

[24] W.M. Alberico et al., *Inelastic nu and anti-nu scattering on nuclei and *strangeness* of the nucleon*, *Nucl. Phys.* **A 623** (1997) 471 [`hep-ph/9703415`] [SPIRES].

[25] K.S. Kim, M.-K. Cheoun and B.G. Yu, *Effect of strangeness for neutrino (anti-neutrino) scattering in the quasi-elastic region*, *Phys. Rev.* **C 77** (2008) 054604 [SPIRES].

[26] J. Liu, R.D. McKeown and M.J. Ramsey-Musolf, *Global Analysis of Nucleon Strange Form Factors at Low Q^2*, *Phys. Rev.* **C 76** (2007) 025202 [`arXiv:0706.0226`] [SPIRES].

[27] R.D. Young, J. Roche, R.D. Carlini and A.W. Thomas, *Extracting nucleon strange and anapole form factors from world data*, *Phys. Rev. Lett.* **97** (2006) 102002 [`nucl-ex/0604010`] [SPIRES].

[28] P.E. Bosted, *An Empirical fit to the nucleon electromagnetic form- factors*, *Phys. Rev.* **C 51** (1995) 409 [SPIRES].

[29] E.J. Brash, A. Kozlov, S. Li and G.M. Huber, *New empirical fits to the proton electromagnetic form factors*, *Phys. Rev.* **C 65** (2002) 051001 [`hep-ex/0111038`] [SPIRES].

[30] H.S. Budd, A. Bodek and J. Arrington, *Modeling quasi-elastic form factors for electron and neutrino scattering*, presented at *2nd International Workshop on Neutrino - Nucleus Interactions in the Few GeV Region* (NUINT 02), Irvine, California, 12-15 Dec. 2002, [`hep-ex/0308005`] [SPIRES].

[31] J. Arrington, *How well do we know the electromagnetic form factors of the proton?*, *Phys. Rev.* **C 68** (2003) 034325 [`nucl-ex/0305009`] [SPIRES].

[32] J.J. Kelly, *Simple parametrization of nucleon form factors*, *Phys. Rev.* **C 70** (2004) 068202 [SPIRES].

[33] J. Arrington and I. Sick, *Precise determination of low-Q nucleon electromagnetic form factors and their impact on parity-violating e p elastic scattering*, *Phys. Rev.* **C 76** (2007) 035201 [`nucl-th/0612079`] [SPIRES].

[34] A. Bodek, S. Avvakumov, R. Bradford and H.S. Budd, *Vector and Axial Nucleon Form Factors:A Duality Constrained Parameterization*, *Eur. Phys. J.* **C 53** (2008) 349 [`arXiv:0708.1946`] [SPIRES].

[35] S. Galster et al., *Elastic electron - deuteron scattering and the electric neutron form-factor at four momentum transfers $5\,fm^{-2} < q^2 < 14\,fm^{-2}$*, *Nucl. Phys.* **B 32** (1971) 221 [SPIRES].

[36] A.F. Krutov and V.E. Troitsky, *Extraction of the neutron charge form factor from the charge form factor of deuteron*, *Eur. Phys. J.* **A 16** (2003) 285 [`hep-ph/0202183`] [SPIRES].

[37] W.M. Alberico, S.M. Bilenky, C. Giunti and K.M. Graczyk, *Electromagnetic form factors of the nucleon: new fit and analysis of uncertainties*, *Phys. Rev.* **C 79** (2009) 065204 [`arXiv:0812.3539`] [SPIRES].

[38] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press (2008).

[39] S. Geman, E. Bienenstock, and R. Doursat, *Neural networks and the bias/variance dilema*, *Neural Comput.* **4** (1992) 1.

[40] B. Denby, *Neural networks and cellular automata in experimental high energy physics*, *Comput. Phys. Commun.* **49** (1988) 429.

[41] B. Mellado, W. Quayle and S.L. Wu, *Prospects for the observation of a Higgs boson with* $H \to \tau^+\tau^- \to \ell^+\ell^-$ $\not{p}_t$ *associated with one jet at the LHC*, *Phys. Lett.* **B 611** (2005) 60 [hep-ph/0406095] [SPIRES].

[42] K. Kurek, E. Rondio, R. Sulej and K. Zaremba, *Application of the neural networks in events classification in the measurement of spin structure of the deuteron*, *Meas. Sci. Technol.* **18** (2007) 2486.

[43] S. Forte, L. Garrido, J.I. Latorre and A. Piccione, *Neural network parametrization of deep-inelastic structure functions*, *JHEP* **05** (2002) 062 [hep-ph/0204232] [SPIRES].

[44] J. Damgov and L. Litov, *Application of neural networks for energy reconstruction*, *Nucl. Instrum. Meth.* **A 482** (2002) 776 [hep-ex/0012003] [SPIRES].

[45] NNPDF collaboration, *Site of the NNPDFs collaboration* http://sophia.ecm.ub.es/nnpdf/.

[46] NNPDF collaboration, L. Del Debbio, S. Forte, J.I. Latorre, A. Piccione and J. Rojo, *Unbiased determination of the proton structure function F2(p) with faithful uncertainty estimation*, *JHEP* **03** (2005) 080 [hep-ph/0501067] [SPIRES].

[47] NNPDF collaboration, L. Del Debbio, S. Forte, J.I. Latorre, A. Piccione and J. Rojo, *Neural network determination of parton distributions: the nonsinglet case*, *JHEP* **03** (2007) 039 [hep-ph/0701127] [SPIRES].

[48] NNPDF collaboration, R.D. Ball et al., *A determination of parton distributions with faithful uncertainty estimation*, *Nucl. Phys.* **B 809** (2009) 1 [*Erratum ibid.* **B816** (2009) 293] [arXiv:0808.1231] [SPIRES].

[49] NNPDF collaboration, R.D. Ball et al., *Fitting Parton Distribution Data with Multiplicative Normalization Uncertainties*, *JHEP* **05** (2010) 075 [arXiv:0912.2276] [SPIRES].

[50] R.D. Ball et al., *A first unbiased global NLO determination of parton distributions and their uncertainties*, *Nucl. Phys.* **B 838** (2010) 136 [arXiv:1002.4407] [SPIRES].

[51] D.J.C. MacKay, *Bayesian interpolation*, *Neural Comput.* **4** (1992) 415.

[52] D.J.C. MacKay, *A practical Bayesian framework for backpropagation networks*, *Neural Comput.* **4** (1992) 448.

[53] D.J.C. MacKay, *Bayesian methods for backpropagation networks*, in *Models of Neural Networks III*, section 6 E. Domany, J.L. van Hemmen and K. Schulten eds., Springer-Verlag, New York (1994).

[54] R. Sulej, *NetMaker*, http://www.ire.pw.edu.pl/~rsulej/NetMaker/ (written in C#); raw fit results presented in this paper available at http://www.ire.pw.edu.pl/~rsulej/NetMaker/index.php?pg=h33.

[55] V. Kurková, *Kolmogorov's theorem and multilayer neural networks*, *Neural Networks* **5** (1992) 501.

[56] A.S. Weigend, D.E. Rumelhart and B.A. Huberman, *Generalization by Weight-Elimination with Application to Forecasting*, proceedings of the *Conference on Advances in Neural Information Processing Systems*, **3** (1990) 875, Denver, Colorado, U.S.A.

[57] K. Hornik, *Approximation Capabilities of Multilayer Feedforward Networks*, *Neural Networks* **4** (1991) 251.

[58] M. Leshno et al., *Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function*, *Neural Networks* **6** (1993) 861.

[59] D.E. Rumelhart et al., *Learning internal representations by error propagation*, in monograph D.E. Rumelhart and J.A. McClelland, *Parallel Distribuited Processing: Exploration in the Microstructure of Cognition*, **1** (1986) 318, The MIT Press.

[60] K. Levenberg, *A Method for the Solution of Certain Non-Linear Problems in Least Squares*, *Q. Appl. Math.* **2** (1944) 164.

[61] D. Marquardt, *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*, *SIAM J. Appl. Math.* **11** (1963) 431.

[62] S. Fahlman, *An Empirical Study of Learning Speed in Back-Propagation Networks*, CMU-CS-88-162, School of Computer Science, Carnegie Mellon University (1988).

[63] Ch. Igel, *Improving the Rprop Learning Algorithm*, proceedings of the *Second International Symposium on Neural Computation*, NC'2000, ICSC Academic Press (2000), pp. 115-121.

[64] H.H. Thodberg, *Ace of Bayes: application of neural networks with pruning*, Technical Report 1132E, The Danish Meat Research Institute, Maglegaardsvej 2, DK-4000 Roskilde, Denmark (1993).

[65] R.M. Neal, *Bayesian Learning for Neural Networks*, Ph.D thesis, University of Toronto, Canada.

[66] C.M. Bishop, *Exact calculation of the Hessian matrix for the multilayer perceptron*, *Neural Comput.* **4** (1992) 494.

[67] S.F. Gull, *Bayesian inductive inference and maximum entropy.* in *Maximum-Entropy and Bayesian Methods in Science and Engineering*, G.J. Ericson and C.R. Smith eds., vol. 1: Fundations, Dordrecht: Kluwer (1988), pp 53-74.

[68] S.F. Gull, *Development in maximum entropy data analysis*, in *Maximum Entropy and Bayesian Methods*, J. Skilling ed., Dordrecht: Kluwer, Cambridge (1988), pp. 53-71.

[69] P.M. Williams, *Bayesian Regularization and Pruning using a Laplace Prior*, *Neural Comput.* **7** (1995) 117.

[70] J. Arrington, W. Melnitchouk and J.A. Tjon, *Global analysis of proton elastic form factor data with two-photon exchange corrections*, *Phys. Rev.* **C 76** (2007) 035205 [arXiv:0707.1861] [SPIRES].

[71] I.A. Qattan et al., *Precision Rosenbluth measurement of the proton elastic form factors*, *Phys. Rev. Lett.* **94** (2005) 142301 [nucl-ex/0410010] [SPIRES].

[72] O. Gayou et al., *Measurements of the elastic electromagnetic form factor ratio mu pGEp/GMp via polarization transfer*, *Phys. Rev.* **C 64** (2001) 038202 [SPIRES];

[73] P.A.M. Guichon and M. Vanderhaeghen, *How to reconcile the Rosenbluth and the polarization transfer method in the measurement of the proton form factors*, *Phys. Rev. Lett.* **91** (2003) 142303 [hep-ph/0306007] [SPIRES].

[74] P.G. Blunden, W. Melnitchouk and J.A. Tjon, *Two-photon exchange and elastic electron proton scattering*, *Phys. Rev. Lett.* **91** (2003) 142304 [nucl-th/0306076] [SPIRES].

[75] Y.C. Chen, A. Afanasev, S.J. Brodsky, C.E. Carlson and M. Vanderhaeghen, *Partonic calculation of the two-photon exchange contribution to elastic electron proton scattering at large momentum transfer*, *Phys. Rev. Lett.* **93** (2004) 122301 [hep-ph/0403058] [SPIRES].

[76] A.V. Afanasev, S.J. Brodsky, C.E. Carlson, Y.-C. Chen and M. Vanderhaeghen, *The two-photon exchange contribution to elastic electron nucleon scattering at large momentum transfer*, *Phys. Rev.* **D 72** (2005) 013008 [hep-ph/0502013] [SPIRES];

[77] C.E. Carlson and M. Vanderhaeghen, *Two-photon physics in hadronic processes*, *Ann. Rev. Nucl. Part. Sci.* **57** (2007) 171 [hep-ph/0701272] [SPIRES].

[78] L. Andivahis et al., *Measurements of the electric and magnetic form-factors of the proton from Q^2=1.75 to 8.83 $(GeV/c)^2$*, *Phys. Rev.* **D 50** (1994) 5491 [SPIRES].

[79] W. Bartel et al., *Measurement of proton and neutron electromagnetic form- factors at squared four momentum transfers up to $3\,GeV/c^2$*, *Nucl. Phys.* **B 58** (1973) 429 [SPIRES].

[80] C. Berger, V. Burkert, G. Knop, B. Langenbeck and K. Rith, *Electromagnetic form-factors of the proton at squared four momentum transfers between $10\,fm^{-2}$ and $50\,fm^{-2}$*, *Phys. Lett.* **B 35** (1971) 87 [SPIRES].

[81] F. Borkowski, P. Peuser, G.G. Simon, V.H. Walther and R.D. Wendling, *Electromagnetic Form-Factors of the Proton at Low Four-Momentum Transfer*, *Nucl. Phys.* **B 93** (1975) 461 [SPIRES].

[82] K.M. Hanson et al., *Large angle quasielastic electron-deuteron scattering*, *Phys. Rev.* **D 8** (1973) 753 [SPIRES].

[83] L.E. Price et al., *Backward-angle electron-proton elastic scattering and proton electromagnetic form-factors*, *Phys. Rev.* **D 4** (1971) 45 [SPIRES].

[84] R.C. Walker et al., *Measurements of the proton elastic form-factors for $1 \leq Q^2 \leq 3\ (GeV/c)^2$ at SLAC*, *Phys. Rev.* **D 49** (1994) 5671 [SPIRES].

[85] P.E. Bosted et al., *Measurements of the deuteron and proton magnetic form-factors at large momentum transfers*, *Phys. Rev.* **C 42** (1990) 38 [SPIRES].

[86] A.F. Sill et al., *Measurements of elastic electron-proton scattering at large momentum transfer*, *Phys. Rev.* **D 48** (1993) 29 [SPIRES].

[87] G.G. Simon, C. Schmitt, F. Borkowski and V.H. Walther, *Absolute electron Proton Cross-Sections at Low Momentum Transfer Measured with a High Pressure Gas Target System*, *Nucl. Phys.* **A 333** (1980) 381 [SPIRES].

[88] J.J. Murphy, Y.M. Shin and D.M. Skopik, *Proton form factor from 0.15 to 0.79 fm^{-2}*, *Phys. Rev.* **C 9** (1974) 2125 [SPIRES].

[89] *Electron-nucleon scattering in resonance regions*, http://www.jlab.org/resdata

[90] BLAST collaboration, E. Geis et al., *The Charge Form Factor of the Neutron at Low Momentum Transfer from the $^2\vec{\mathrm{H}}(\vec{e}, e'\mathrm{n})\mathrm{p}$ Reaction*, *Phys. Rev. Lett.* **101** (2008) 042501 [arXiv:0803.3827] [SPIRES].

[91] JEFFERSON LAB E95-001 collaboration, W. Xu et al., *PWIA extraction of the neutron magnetic form factor from quasi-elastic $^3\vec{\mathrm{He}}(\vec{e}, e')$ at $Q^2 = 0.3\,(GeV/c)^2$ to 0.6-$(GeV/c)^2$*, *Phys. Rev.* **C 67** (2003) 012201 [nucl-ex/0208007] [SPIRES].

[92] K. Graczyk, *Krzysztof Graczyk Homepage*, http://www.ift.uni.wroc.pl/∼kgraczyk/nn.html.