Input Parametrization and Neural Networks

Alberto Guffanti

Albert-Ludwigs-Universität Freiburg

On behalf of the NNPDF Collaboration:
R. D. Ball, L. Del Debbio, M. Ubiali (Edinburgh), S. Forte, A. Piccione (Milano),
J. I. Latorre (Barcelona), J. Rojo-Chacon (LPTHE - Paris)

PDF4LHC Workshop CERN, February 22 - 23, 2008

Initial scale parametrization of PDFs

Standard Approach

- Introduce a simple functional form with enough (but not too many) free parameters

$$q(x, Q^2) = x^\alpha (1 - x)^\beta P(x; \lambda_1, ..., \lambda_n).$$

- Fit parameters minimizing χ^2.

Initial scale parametrization of PDFs

Standard Approach

- Introduce a simple functional form with enough (but not too many) free parameters

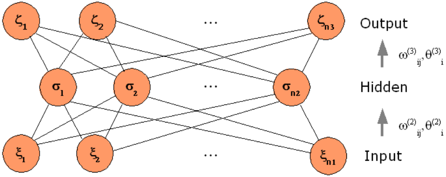
$$q(x, Q^2) = x^\alpha (1 - x)^\beta P(x; \lambda_1, ..., \lambda_n).$$

- Fit parameters minimizing χ^2.

Open problem:

- Theoretical bias due to the chosen parametrization is difficult to assess.

Why use Neural Networks?



- Neural Networks are non-linear statistical tools.
- Any continuous function can be approximated with neural network with one internal layer and non-linear neuron activation function.
- Efficient minimization algorithms for complex parameter spaces.
- They provide a parametrization which is redundant and robust against variations.

Neural Networks

... just another basis of functions

Multilayer feed-forward networks

- Each neuron receives input from neurons in preceding layer and feeds output to neurons in subsequent layer
- Activation determined by weights and thresholds

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

- Sigmoid activation function

$$g(x) = \frac{1}{1 + e^{-\beta x}}$$

Neural Networks

... just another basis of functions

Multilayer feed-forward networks

- Each neuron receives input from neurons in preceding layer and feeds output to neurons in subsequent layer

- Activation determined by weights and thresholds

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

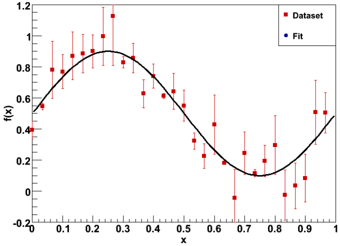
- Sigmoid activation function

$$g(x) = \frac{1}{1 + e^{-\beta x}}$$

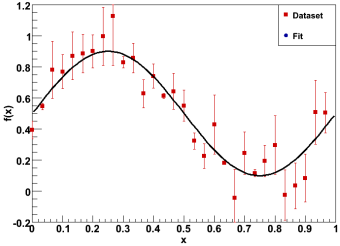
A 1-2-1 NN:

$$\xi_1^{(3)}(\xi_1^{(1)}) = \frac{1}{1 + e^{\theta_1^{(3)} - \frac{\omega_{11}^{(2)}}{1+e^{\theta_1^{(2)} - \xi_1^{(1)}\omega_{11}^{(1)}}} - \frac{\omega_{12}^{(2)}}{1+e^{\theta_2^{(2)} - \xi_1^{(1)}\omega_{21}^{(1)}}}}}$$

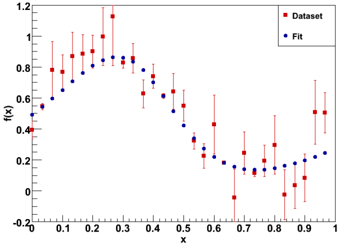
Neural Network vs. Polynomial form

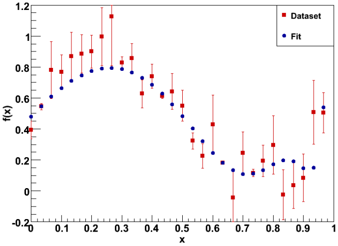


Neural Network vs. Polynomial form



Neural Net Fit ($\chi^2 = 1$) **Polynomial form Fit** ($\chi^2 = 1$)





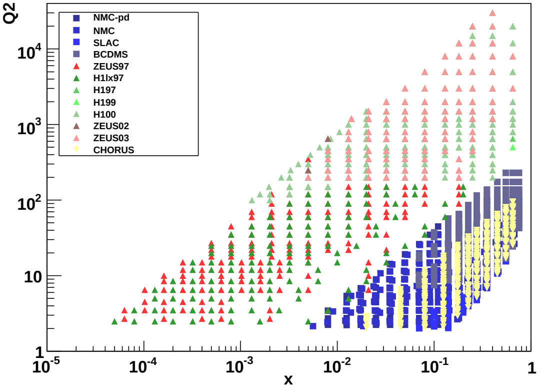
NNPDF Singlet fit: Datasets

Datasets included in the analysis

Name	N_c	N_n	N_N	Data points	Target
NMC_pd	5	0	0	260	F_2^d/F_2^p
NMC	12	0	4	288	F_2^p
SLAC	0	1	1	211 (211)	$F_2^{p(d)}$
BCDMS	3	9	1	351 (254)	$F_2^{p(d)}$
ZEUS97	10	0	1	240	$\tilde{\sigma}^{NC,+}$
H1lx97	5	0	1	133	$\tilde{\sigma}^{NC,+}$
H197	6	0	1	130 (25)	$\tilde{\sigma}^{NC(CC),+}$
H199	6	0	1	126 (27)	$\tilde{\sigma}^{NC(CC),-}$
H100	7	0	1	147 (28)	$\tilde{\sigma}^{NC(CC),+}$
ZEUS02	6	0	1	92	$\tilde{\sigma}^{NC,-}$
ZEUS03	8	0	1	90	$\tilde{\sigma}^{NC,+}$
CHORUS	13	0	2	607 (607)	$\tilde{\sigma}^{\nu(\bar{\nu})}$

NNPDF Singlet fit: Datasets

Kinematical coverage

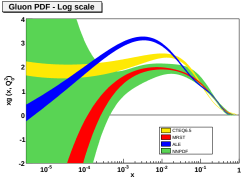


NNPDF Singlet fit: Neural Network set up

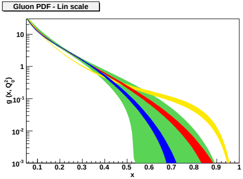
- 5 Neural Networks [g, Σ, Isotriplet, $(u_v + d_v)$, $(\bar{d} - \bar{u})$]
- Network Architecture: 2-3-2-1 (exception Sea Asymmetry: 2-3-1)
- 93 free parameters

Preliminary results: 25 replicas

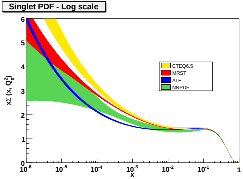
gluon



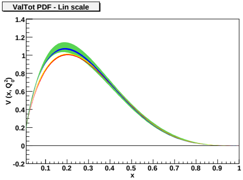
gluon



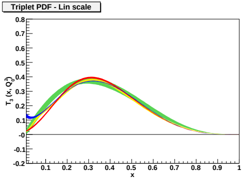
singlet



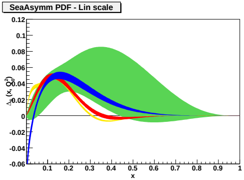
valence



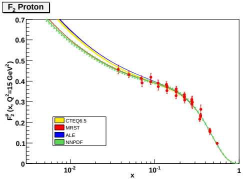
triplet



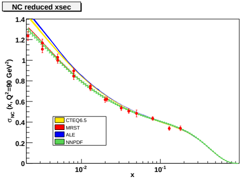
sea asym.



proton F_2



NC xsect.



neutrino xsect.