Extracting Parton Distribution Functions from data: the Neural Network approach

Andrea Piccione

Edinburgh - Jan, 11th 2006

The NNPDF Collaboration

Luigi Del Debbio[1], Stefano Forte[2],
José I. Latorre[3], A. P.[4] and Joan Rojo[3]

[1] Particle Physics Theory Group, School of Physics, University of Edinburgh
[2] Dipartimento di Fisica, Università di Milano and INFN, Sezione di Milano
[3] Departament d'Estructura i Constituents de la Matèria, Universitat de Barcelona
[4] Dipartimento di Fisica Teorica, Università di Torino and INFN, Sezione di Torino

What do we need for the LHC?

What do we need for the LHC?

- SPS: hadronic collider, *strong* signals and low precision
- LEP: leptonic collider, *low* signals and high precision
- LHC: hadronic collider, *low* signals and high precision

What do we need for the LHC?

- Good reconstruction of final states
- Precise partonic cross-sections calculations
- Accurate description of incoming hadrons

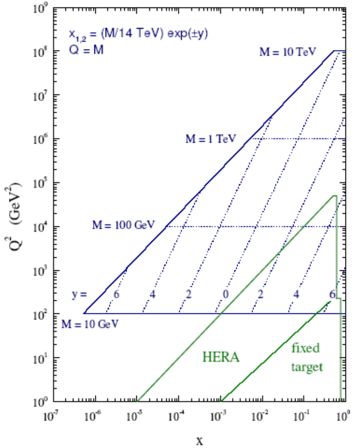
How do we describe hadrons?

- QCD describes interactions between quarks and gluons. Experimentally we observe only hadrons → Confinement
- Perturbative QCD is not trustable at low energies ($\sim$ GeV). We can not solve QCD in the non-perturbative region, but on a lattice . . .
- We can extract information on the proton structure from a process with only one initial proton (DIS at HERA). Then we can use these as an input for a process where two initial protons are involved (DY at LHC) → Factorization

How do we describe hadrons?

- QCD describes interactions between quarks and gluons. Experimentally we observe only hadrons → Confinement
- Perturbative QCD is not trustable at low energies ($\sim$ GeV). We can not solve QCD in the non-perturbative region, but on a lattice . . .
- We can extract information on the proton structure from a process with only one initial proton (DIS at HERA). Then we can use these as an input for a process where two initial protons are involved (DY at LHC) → Factorization

LHC and DIS kinematics



Deep Inelastic Scattering

- The cross section

$$\frac{d^2\sigma}{dxdQ^2} = \frac{4\pi\alpha^2}{Q^4}\left[[1+(1-y)^2]F_1 + \frac{1-y}{x}\left(F_2 - 2xF_1\right)\right]$$

- The structure function

$$F_2(x,\,Q^2) = x\left[\sum_{q=1}^{n_f} e_q^2\, C^q \otimes q_q(x,\,Q^2) + 2n_f\, C^g \otimes g(x,\,Q^2)\right]$$

- Parton distribution evlution is described by DGLAP equations

$$Q^2\frac{d}{dQ^2}q(x,\,Q^2) = \frac{\alpha_s(Q^2)}{2\pi}(P\otimes q)(x,\,Q^2)$$

Deep Inelastic Scattering and QCD

- The cross section

$$\frac{d^2\sigma}{dxdQ^2} = \frac{4\pi\alpha^2}{Q^4}\left[[1+(1-y)^2]F_1 + \frac{1-y}{x}(F_2 - 2xF_1)\right]$$

- The structure function

$$F_2(x,\ Q^2) = x\left[\sum_{q=1}^{n_f} e_q^2\, \mathcal{C}^q \otimes q_q(x,\ Q^2) + 2n_f\, \mathcal{C}^g \otimes g(x,\ Q^2)\right]$$

- Parton distribution evlution is described by DGLAP equations

$$Q^2\frac{d}{dQ^2}q(x,Q^2) = \frac{\alpha_s(Q^2)}{2\pi}(P\otimes q)(x,Q^2)$$

Deep Inelastic Scattering and QCD

- The cross section

$$\frac{d^2\sigma}{dxdQ^2} = \frac{4\pi\alpha^2}{Q^4}\left[[1+(1-y)^2]F_1 + \frac{1-y}{x}\left(F_2 - 2xF_1\right)\right]$$

- The structure function

$$F_2(x,\, Q^2) = x\left[\sum_{q=1}^{n_f} e_q^2\, \mathcal{C}^q \otimes q_q(x,\, Q^2) + 2n_f\, \mathcal{C}^g \otimes g(x,\, Q^2)\right]$$

- Parton distribution evlution is described by DGLAP equations

$$Q^2\frac{d}{dQ^2}q(x,Q^2) = \frac{\alpha_s(Q^2)}{2\pi}(P\otimes q)(x,Q^2)$$

The problem

- For a single quantity → 1 sigma error
- For a pair of numbers → 1 sigma ellipse
- For a function → We need an "error band" in the space of functions (*i.e.* the probability density $\mathcal{P}[f]$ in the space of functions $f(x)$)

Expectation values → Functional integrals

$$\langle \mathcal{F}[f(x)] \rangle = \int \mathcal{D}f \, \mathcal{F}[f(x)] \, \mathcal{P}[f(x)]$$

Determine an infinite-dimensional object (a function) from finite set of data points → Mathematically ill-posed problem

The problem

- For a single quantity $\rightarrow$ 1 sigma error
- For a pair of numbers $\rightarrow$ 1 sigma ellipse
- For a function $\rightarrow$ We need an "error band" in the space of functions (*i.e.* the probability density $\mathcal{P}[f]$ in the space of functions $f(x)$)

Expectation values $\rightarrow$ Functional integrals

$$\langle \mathcal{F}[f(x)] \rangle = \int \mathcal{D}f \mathcal{F}[f(x)] \mathcal{P}[f(x)]$$

Determine an infinite-dimensional object (a function) from finite set of data points $\rightarrow$ Mathematically ill-posed problem

The problem

- For a single quantity → 1 sigma error
- For a pair of numbers → 1 sigma ellipse
- For a function → We need an "error band" in the space of functions (*i.e.* the probability density $\mathcal{P}[f]$ in the space of functions $f(x)$)

Expectation values → Functional integrals

$$\langle \mathcal{F}[f(x)] \rangle = \int \mathcal{D}f \mathcal{F}[f(x)] \mathcal{P}[f(x)]$$

Determine an infinite-dimensional object (a function) from finite set of data points → Mathematically ill-posed problem

A solution

1. Choose a basis of functions, and project the PDFs on it
2. Fit coefficients of basis elements by minimizing χ^2

Some possible basis:

- $q(x, Q_0^2) = x^\alpha (1-x)^\beta P(x; \lambda_1, \ldots, \lambda_n)$

  [Les Houches Accord PDF sets: Alekhin, Botje, CTEQ, Fermi (GKK), GRV, H1, MRST, ZEUS]
- Orthogonal Polynomials

  [F. J. Yndurain (1978), G. Parisi and N. Sourlas (1979), W. Furmanski and R. Petronzio (1982)]
- Truncated Moments

  [S. Forte and L. Magnea (1999), S. Forte, L. Magnea, A. P. and G. Ridolfi (2001)]
- Neural Networks

  ["Any continuous function can be uniformly approximated by a continuous neural network having only one internal layer, and with an arbitrary continuous sigmoid non-linearity", G. Cybenko (1989)]

A solution

1. Choose a basis of functions, and project the PDFs on it
2. Fit coefficients of basis elements by minimizing χ^2

Some possible basis:

- $q(x, Q_0^2) = x^{\alpha}(1-x)^{\beta} P(x; \lambda_1, \ldots, \lambda_n)$

  [Les Houches Accord PDF sets: Alekhin, Botje, CTEQ, Fermi (GKK), GRV, H1, MRST, ZEUS]

- Orthogonal Polynomials

  [F. J. Yndurain (1978), G. Parisi and N. Sourlas (1979), W. Furmanski and R. Petronzio (1982)]

- Truncated Moments

  [S. Forte and L. Magnea (1999), S. Forte, L. Magnea, A. P. and G. Ridolfi (2001)]

- Neural Networks

  ["Any continuous function can be uniformly approximated by a continuous neural network having only one internal layer, and with an arbitrary continuous sigmoid non-linearity", G. Cybenko (1989)]

Still some problems

- Error propagation from data to parameters and from parameters to observables is not trivial
- Theoretical bias due to the choice of a parametrization is difficult to assess (effects can be large if data are not precise or hardly compatible)

Still some problems

- Error propagation from data to parameters and from parameters to observables is not trivial
- Theoretical bias due to the choice of a parametrization is difficult to assess (effects can be large if data are not precise or hardly compatible)

The standard approach

- MRST: 15 parms. - $\Delta\chi^2 = 50$ - NC and CC DIS, DY, W-asym, jets

$$xq(x,Q_0^2) = A(1-x)^{\eta}(1+\epsilon x^{0.5}+\gamma x)x^{\delta}, \quad x[\bar{u}-\bar{d}](x,Q_0^2) = A(1-x)^{\eta}(1+\gamma x+\delta x^2)x^{\delta}.$$

$$xg(x,Q_0^2) = A_g(1-x)^{\eta_g}(1+\epsilon_g x^{0.5}+\gamma_g x)x^{\delta_g} - A_-(1-x)^{\eta_-}x^{-\delta_-},$$

- CTEQ: 20 parms. - $\Delta\chi^2 = 100$ - NC and CC DIS, DY, W-asym, jets

$$x\,f(x,Q_0) = A_0\,x^{A_1}\,(1-x)^{A_2}\,e^{A_3 x}\,(1+e^{A_4}\,x)^{A_5}$$

with independent params for combinations $u_v \equiv u-\bar{u}$, $d_v \equiv d-\bar{d}$, g, and $\bar{u}+\bar{d}$, $s=\bar{s}=0.2\,(\bar{u}+\bar{d})$ at Q_0; norm. fixed by sum rules

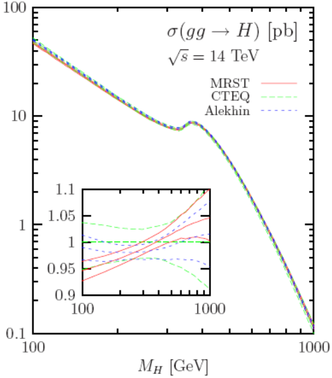
- Alekhin: 17 parms. - $\Delta\chi^2 = 1$ - NC DIS (+ DY)

$$xu_V(x,Q_0) = \frac{2}{N_u^V}x^{a_u}(1-x)^{b_u}(1+\gamma_2^u x); \qquad xu_S(x,Q_0) = \frac{A_S}{N_S}\eta_u x^{a_s}(1-x)^{b_{su}}$$

$$xd_V(x,Q_0) = \frac{1}{N_d^V}x^{a_d}(1-x)^{b_d}; \qquad xd_S(x,Q_0) = \frac{A_S}{N^S}x^{a_s}(1-x)^{b_{sd}},$$

$$xs_S(x,Q_0) = \frac{A_S}{N^S}\eta_s x^{a_s}(1-x)^{(b_{su}+b_{sd})/2}; \qquad xG(x,Q_0) = A_G x^{a_G}(1-x)^{b_G}(1+\gamma_1^G\sqrt{x}+\gamma_2^G x),$$

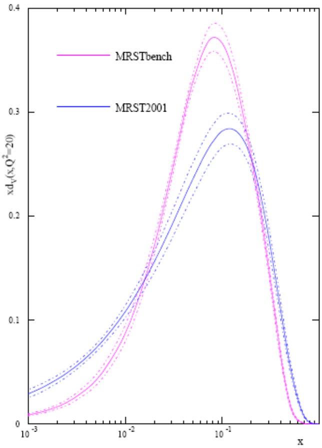
The standard approach - Higgs cross section



"Within a given set of PDFs, the deviations of the cross sections from the values obtained with the reference PDF sets can reach the level of 10% at the LHC in the case of the gluon-gluon fusion process for large enough Higgs boson masses, $M_H \sim 1\,TeV$".

[A. Djouadi and S. Ferrag, hep-ph/0310209]

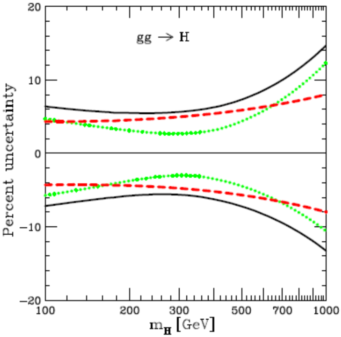
The standard approach - HERA-LHC WKS benchmark



"[...] the inclusion of more data from a variety of different experiments moves the central values of the partons in a manner indicating either that the different experimental data are inconsistent with each other, or that the theoretical framework is inadequate for correctly describing the full range of data. To a certain extent both explanations are probably true."

[R. S. Thorne, hep-ph/0511119]

The standard approach - Dependence on α_s



"the previous fits with $\alpha_s(m_Z) = 0.118$ are adequate for most processes, because the uncertainty associated with α_s is smaller than the other sources of PDF uncertainty. However, α_s uncertainty is important for inclusive jet production at relatively small p_T and Higgs boson production by the $gg \to H$ process in the standard model."

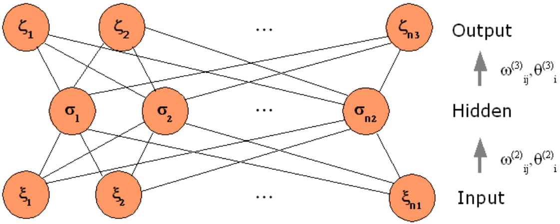
[CTEQ, hep-ph/0512167]

The NNPDF approach

- Determination of the Structure Functions:
  this is the easiest case, since no evolution is required, but only data fitting. A good application to test the technique → Done
- Determination of the Parton Distributions:
  the real stuff → Working on it ...

What are Neural Networks?

Neural networks are a class of algorithms very suitable to fit incomplete or noisy data



Some details on their structure

- Building blocks: neurons, *i. e.* input/output units characterized by sigmoid activation

$$\xi_i^{(l)} = g\left(\sum_{j=1}^{n_l-1} \omega_{ij}^{(l-1)} \xi_j^{(l-1)} - \theta_i^{(l)}\right) \qquad g(x) = \frac{1}{1+e^{-x}}$$

- Parameters: weights $\omega_{ij}^{(l)}$ and thresholds $\theta_i^{(l)}$.
- Architecture: multilayer feed-forward NN. Each neuron receives input from neurons in preceding layer and feeds output to neurons in successive layer
- Assumption: smooth function

Back Propagation

1. Set the parameters randomly.
2. Present an input and calculate the output.
3. Evaluate χ^2.
4. Modify the weights to reinforce correct decisions and discourage incorrect ones:

$$\omega_{ij} \rightarrow \omega_{ij} - \eta \frac{\partial \chi^2}{\partial \omega_{ij}}$$

   where η is the learning rate.
5. Back to 2, till the stability of χ^2 is reached

Genetic Algorithm

1. Set the parameters randomly.
2. Make clones of the set of parameters.
3. Mutate each clone.
4. Evaluate χ^2 for all the clones.
5. Select the clone that has the lowest χ^2.
6. Back to 2, till the stability of χ^2 is reached.

General strategy

- Monte Carlo sampling of data (generation of replicas of experimental data) → Faithful error propagation

$$F_i^{(art)(k)} = \left(1 + r_N^{(k)}\sigma_N\right)\left[F_i^{(exp)} + r_i^s \sigma_i^{stat} + \sum_{l=1}^{N_{sys}} r^{l,(k)} \sigma_i^{sys,l}\right]$$

- NN training over MC replicas → Unbiased parametrization

Expectation values → Sum over the Nets

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

General strategy

- Monte Carlo sampling of data (generation of replicas of experimental data) → Faithful error propagation

$$F_i^{(art)(k)} = \left(1 + r_N^{(k)}\sigma_N\right)\left[F_i^{(exp)} + r_i^s \sigma_i^{stat} + \sum_{l=1}^{N_{sys}} r^{l,(k)}\sigma_i^{sys,l}\right]$$

- NN training over MC replicas → Unbiased parametrization

Expectation values → Sum over the Nets

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

General strategy

- Monte Carlo sampling of data (generation of replicas of experimental data) → Faithful error propagation

$$F_i^{(art)(k)} = \left(1 + r_N^{(k)}\sigma_N\right)\left[F_i^{(exp)} + r_i^s\sigma_i^{stat} + \sum_{l=1}^{N_{sys}} r^{l,(k)}\sigma_i^{sys,l}\right]$$

- NN training over MC replicas → Unbiased parametrization

Expectation values → Sum over the Nets

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

General strategy

- Monte Carlo sampling of data (generation of replicas of experimental data) → Faithful error propagation

$$F_i^{(art)(k)} = \left(1 + r_N^{(k)}\sigma_N\right)\left[F_i^{(exp)} + r_i^s \sigma_i^{stat} + \sum_{l=1}^{N_{sys}} r^{l,(k)}\sigma_i^{sys,l}\right]$$

- NN training over MC replicas → Unbiased parametrization

Expectation values → Sum over the Nets

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

Details

Architecture: 4-5-3-1

- Inputs: x, $\log x$, Q^2, $\log Q^2$
- Output: $F_2(x, Q^2)$

Minimization strategy:

- Back Propagation ($\sim 10^8$ training cycles):

$$\chi^{2\,(k)}_{\mathrm{diag}} = \frac{1}{N_{\mathrm{dat}}} \sum_{i=1}^{N_{\mathrm{dat}}} \frac{\left(F_i^{(\mathrm{art})(k)} - F_i^{(\mathrm{net})(k)}\right)^2}{\sigma_{i,t}^{(\mathrm{exp})2}}$$

- Genetic Algorithm ($\sim 10^4$ generations):

$$\chi^{2\,(k)} = \frac{1}{N_{\mathrm{dat}}} \sum_{i,j=1}^{N_{\mathrm{dat}}} \left(F_i^{(\mathrm{art})(k)} - F_i^{(\mathrm{net})(k)}\right) \mathrm{cov}_{ij}^{-1} \left(F_j^{(\mathrm{art})(k)} - F_j^{(\mathrm{net})(k)}\right)$$

Details

Architecture: 4-5-3-1

- Inputs: x, $\log x$, Q^2, $\log Q^2$
- Output: $F_2(x, Q^2)$

Minimization strategy:

- Back Propagation ($\sim 10^8$ training cycles):

$$\chi^{2\,(k)}_{\mathrm{diag}} = \frac{1}{N_{\mathrm{dat}}} \sum_{i=1}^{N_{\mathrm{dat}}} \frac{\left(F_i^{(\mathrm{art})(k)} - F_i^{(\mathrm{net})(k)}\right)^2}{\sigma_{i,t}^{(\mathrm{exp})2}}$$

- Genetic Algorithm ($\sim 10^4$ generations):

$$\chi^{2\,(k)} = \frac{1}{N_{\mathrm{dat}}} \sum_{i,j=1}^{N_{\mathrm{dat}}} \left(F_i^{(\mathrm{art})(k)} - F_i^{(\mathrm{net})(k)}\right) \mathrm{cov}_{ij}^{-1} \left(F_j^{(\mathrm{art})(k)} - F_j^{(\mathrm{net})(k)}\right)$$

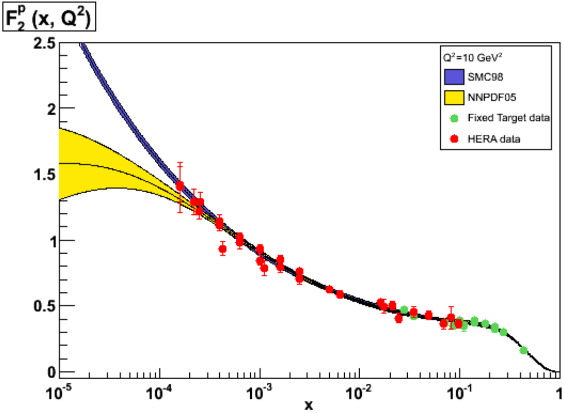
Credits

- S. Forte, L. Garrido, J. I. Latorre and A. P., *"Neural network parametrization of deep-inelastic structure functions,"* JHEP05 (2002) 062 [arXiv:hep-ph/0204232]
- L. Del Debbio, S. Forte, J. I. Latorre, A. P. and J. Rojo [NNPDF Collaboration], *"Unbiased determination of the proton structure function F_2^p with faithful uncertainty estimation"*, JHEP03 (2005) 080 [arXiv:hep-ph/0501067]

Source code, driver program and graphical web interface for F_2 plots and numerical computations available @

http://sophia.ecm.ub.es/f2neural

Fit of $F_2^p(x, Q^2)$



Same strategy

- Monte Carlo sampling of data
- Parametrization of parton distributions with neural networks
- DGLAP evolution of parton distributions to experimental data scale and training over Monte Carlo replica sample

Same strategy, but much more complex!

- Monte Carlo sampling of data
- Parametrization of parton distributions with neural networks
- DGLAP evolution of parton distributions to experimental data scale and training over Monte Carlo replica sample

Examples

- Expectation values:

$$\langle \mathcal{F}\left[g(x)\right]\rangle = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} \mathcal{F}\left(g^{(net)(k)}(x)\right)$$

- Errors:

$$\sigma_{\mathcal{F}[g(x)]} = \sqrt{\langle \mathcal{F}\left[g(x)\right]^2\rangle - \langle \mathcal{F}\left[g(x)\right]\rangle^2}$$

- Correlations between pairs of different parton distributions at different points:

$$\langle u(x_1)d(x_2)\rangle = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} u^{(net)(k)}(x_1, Q_0^2)d^{(net)(k)}(x_2, Q_0^2)$$

PDF Evolution (*very technical!*)

- We want Mellin space evolution (numerically efficient):

$$q(N, Q^2) = q(N, Q_0^2)\Gamma\left(N, \alpha_s\left(Q^2\right), \alpha_s\left(Q_0^2\right)\right)$$

- We do not want complex neural networks:

$$\Gamma\left(x, \alpha_s\left(Q^2\right), \alpha_s\left(Q_0^2\right)\right) \equiv \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} dN\, x^{-N}\Gamma\left(N, \alpha_s\left(Q^2\right), \alpha_s\left(Q_0^2\right)\right)$$

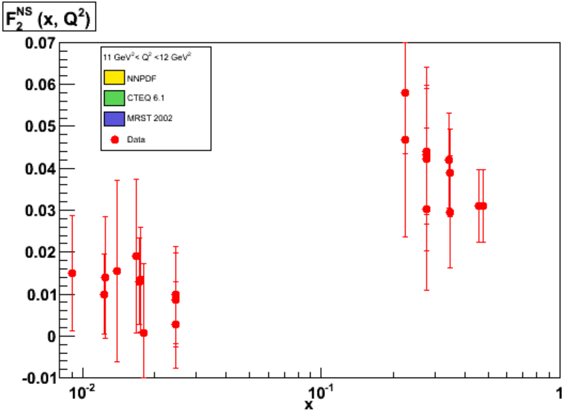
- $\Gamma(x)$ is a distribution $\rightarrow$ must be regularized at $x = 1$:

$$q(x, Q^2) = q(x, Q_0^2)\int_x^1 dy\, \Gamma(y) + \int_x^1 \frac{dy}{y}\Gamma(y)\left(q\left(\frac{x}{y}, Q_0^2\right) - yq(x, Q_0^2)\right)$$

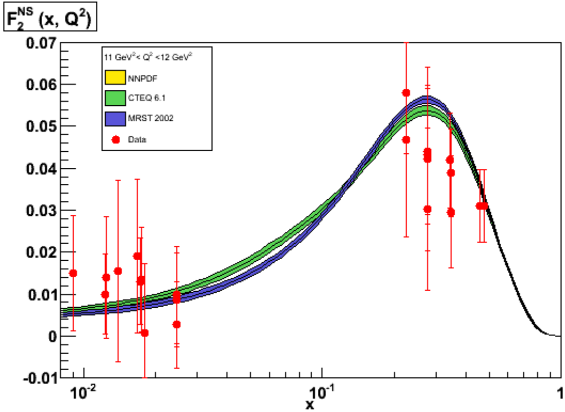
Details (*technical*)

- $q_{NS}(x, Q^2) \equiv \frac{1}{6}\left(u + \bar{u} - d - \bar{d}\right)(x, Q^2)$
- Experimental data: NMC (229 pts) and BCDMS (254 pts)
- Kinematical cuts: $Q^2 \geq 3\ GeV^2$, $W^2 \geq 6.25\ GeV^2$
- Neural network architecture: 2-5-3-1 (37 params.)
- Strong coupling: $\alpha_s\left(M_Z^2\right) = 0.1182$
- Perturbative order: NLO
- VFN: $m_c = 1.5 GeV$, $m_b = 4.5 GeV$, $m_t = 175 GeV$
- TMC: F_2 integral evaluated with NN F_2
- # replica: 100 (should be 1000)

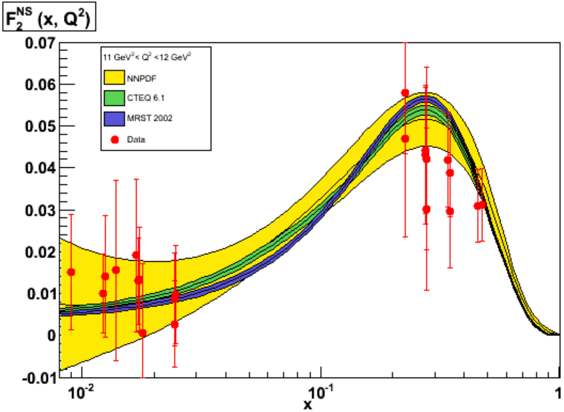
Non-Singlet (preliminary)



Non-Singlet (preliminary)



Non-Singlet (preliminary)



Outlook

- Construct full set of NNPDF parton distributions from all available data
- Assess impact of uncertainties of PDFs for relevant observables at LHC
- Make formalism compatible with standard interfaces (LHAPDF, PDFLIB) $\rightarrow$ NNPDF partons available for use in Monte Carlo generators