VII. Neural Networks for Amplitudes

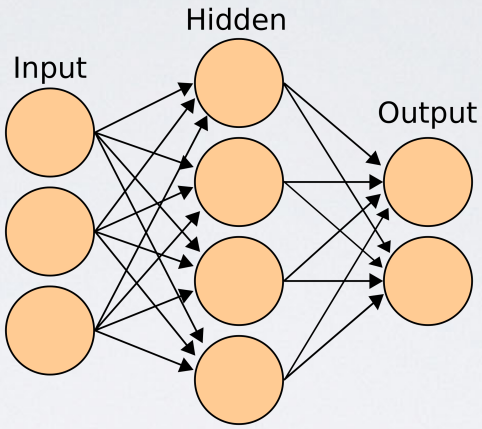
1) First attempt

2) Errors? - Trained network reliability

3) What have we done wrong?

1)

The amplitude Neural Network

momenta → 

→ $\langle |\mathcal{A}|^2 \rangle$

ideal answer

universal interface to amplitude generators → NeuralNetwork → universal interface to event generators

NeuralNetwork ↕ approximation sufficiently robust against changes in cuts, PDFs etc.

phasespace

not sure where this goes...

1.4)

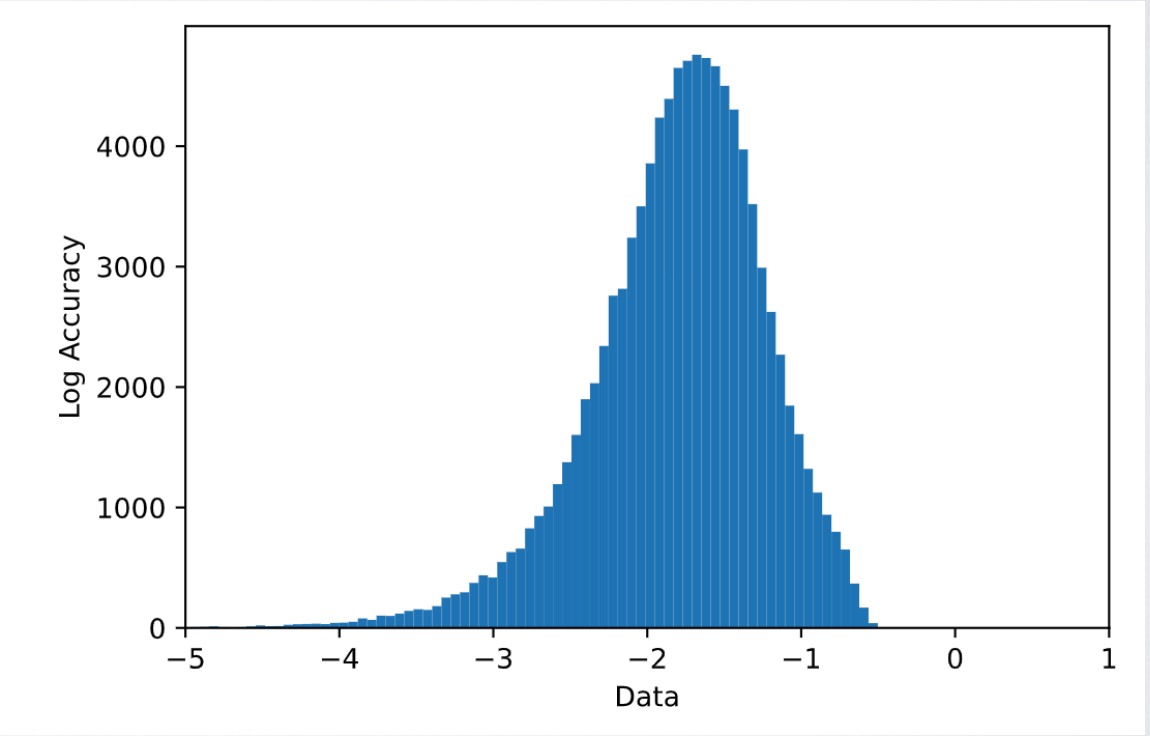
First results

$$\langle |\mathcal{A}|^2 \rangle = 4 \left(\frac{\alpha}{4\pi}\right)^2 \left(\frac{\alpha_s}{4\pi}\right) N_c C_F Q_q^2 \frac{s_{a1}^2 + s_{a2}^2 + s_{b1}^2 + s_{b2}^2}{s_{ab} s_{13} s_{23}}$$

tree-level validation - amplitude evaluation is very fast so no chance of optimisation here…

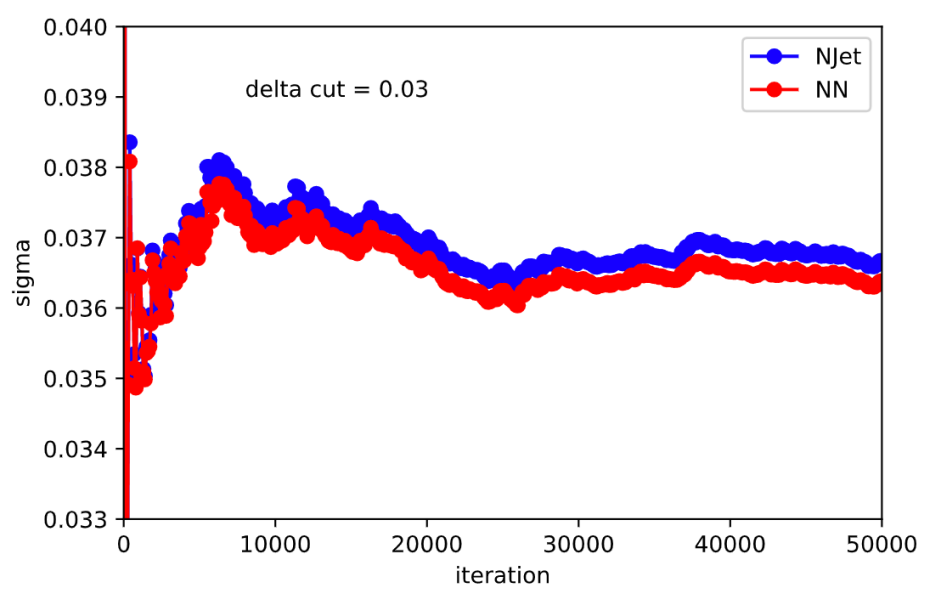
10000 training (80:20 split),
1M interpolation

most points around 10%
(1 digit) accurate

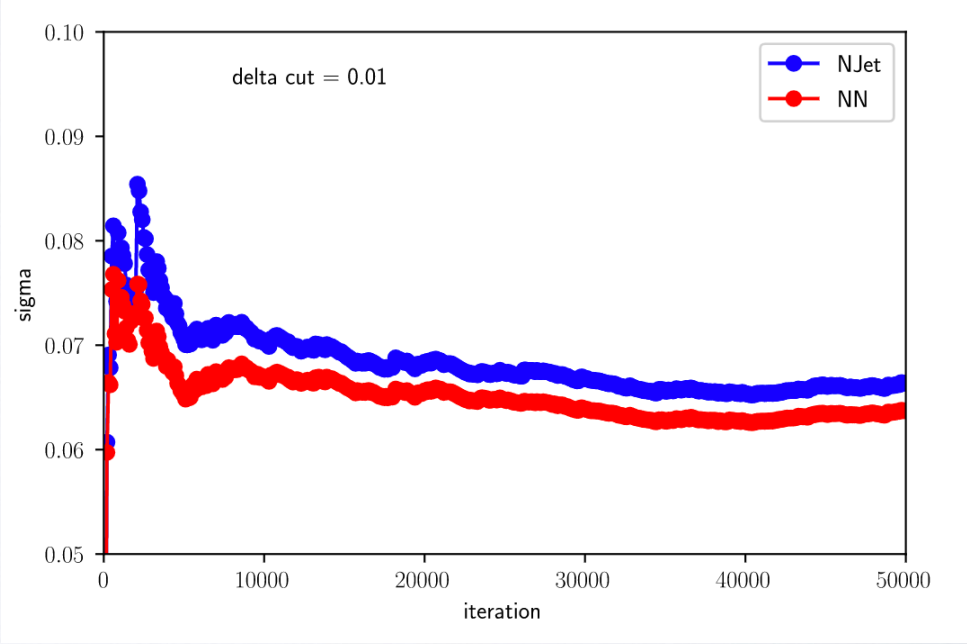


1.4)

First results



NN approximation is less good for small δ



2)

Errors

The MC integration has a well defined error

$$x_i = \langle |\mathcal{A}(p_i)|^2 \rangle$$

$$\sigma = \langle x \rangle \pm \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$$

The dominant error from the NN is the uncertainty in the fit **not** the standard deviation of the interpolated/extrapolated values

Since the NN is fast to call the 'MC' error can be practically zero

2.1)

Neural Network Errors

As far as I know, quantifying errors from a Neural Network is not a standard task.

Especially in this case where the input data is free of experimental noise

Each hyperparameter variation will affect the network -
we can use this to ascertain the reliability of the fit

Shuffle the testing/training splitting (default with `sklearn.model_selection.train_test_split`) while fixing the parameter initialisation (`glorot_uniform(seed = i)`)

train an ensemble of networks (10-20) and take the average and standard deviation for the full prediction