(Machine) Learning the production cross sections in the IDM.

Humberto Reyes-González.
LPSC Grenoble.

In collaboration with
S. Kraml, A. Lessa and S. Otten.

IN2P3/IRFU ML Workshop, Lyon CC-IN2P3 22-23 January 2020

Introduction.

MISSION:

To build neural networks that can **precisely** predict, with an **uncertainty estimation,** the cross sections of the production processes in the Inert Doublet Model.

MOTIVATION:

In phenomenological studies of BSM theories, the computation of production cross sections over large parameter spaces usually takes a large amount of time (in the order of minutes per cross section). Training DNNs to predict these cross sections could substantially save computational costs.

CHALLENGES:

- For these predictions to be useful, a high level of precision is required, ideally with very small uncertainties of the predictions.
- The values of the cross sections range over several order of magnitudes.
- The relation between the input parameters and the cross sections is not always linear. For instance, regions with resonances are hard to learn.

The Inert Doublet Model.

The potential:

$$V = \mu_1^2|\mathbf{H_1}|^2 + \mu_2^2|\mathbf{H_2}|^2 + \lambda_1|\mathbf{H_1}|^4 + \lambda_2|\mathbf{H_2}|^4 + \lambda_3|\mathbf{H_1}|^2|\mathbf{H_2}|^2 + \lambda_4|\mathbf{H_1}^\dagger\mathbf{H_2}|^2 + \frac{\lambda_5}{2}[(\mathbf{H_1}^\dagger\mathbf{H_2})^2 + \text{h.c.}]\,.$$

Five free parameters:

$$M_{H^0}^2 = \mu_2^2 + \frac{1}{2}(\lambda_3 + \lambda_4 + \lambda_5)v^2\,,$$
$$M_{A^0}^2 = \mu_2^2 + \frac{1}{2}(\lambda_3 + \lambda_4 - \lambda_5)v^2\,,$$
$$M_{H^\pm}^2 = \mu_2^2 + \frac{1}{2}\lambda_3 v^2\,,$$

λ_2 and $\lambda_L \equiv \lambda_3 + \lambda_4 + \lambda_5$.

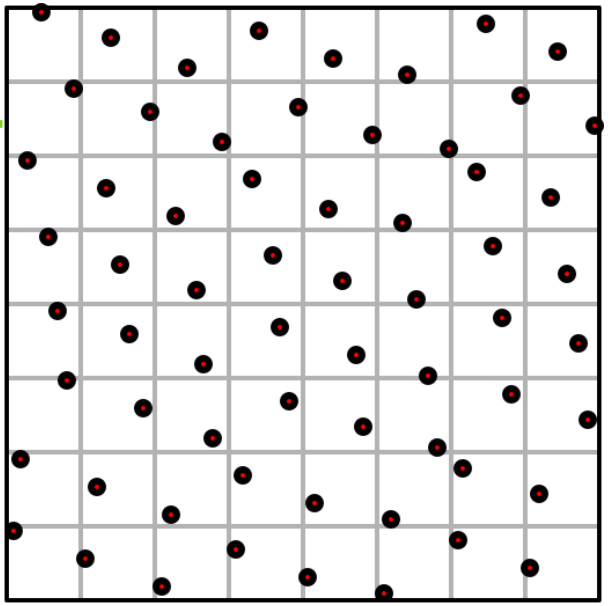
8 production processes to learn:

$$pp \to H^0H^0,\ A^0A^0,\ H^0A^0,\ H^0H^+,\ A^0H^+,\ H^+H^-,\ H^0H^- \text{ and } A^0H^-.$$

Acquiring the training data.

→ All the cross sections were computed at leading order using MADGRAPH 2.6.4 with the IDM UFO implementation from the FeynRules data base.

→ 50,000 samples were generated using the **Jittered Sampling Method,** to evenly cover the full parameter space.



$$50 < M_{H^0},\, M_{A^0},\, M_{H^\pm} < 3000\ \text{GeV}; \quad -2\pi < \lambda_2,\, \lambda_L < 2\pi.$$

→ Since the expected luminosity at HL-LHC is about 3 pb^-1, we imposed a lower limit on the cross section of our dataset of

$$\sigma_{\text{min}} = 10^{-7}$$

→ The remaining data was divided as training and test data set in a 70:30 split.

Data preprocessing and loss function.

We want to create a function that maps the free parameters of the IDM to their corresponding production cross sections.

$$g_\phi : x_{\mathrm{IDM}} \equiv \left(m_{H^0}, m_{A^0}, m_{H^\pm}, \lambda_L, \lambda_2\right) \rightarrow \sigma_{\mathrm{IDM}}$$

For the input parameters, we implemented a z-score transformation:

$$x'_{\mathrm{IDM}} = \frac{x_{\mathrm{IDM}} - \mu(x_{\mathrm{IDM}})}{\sigma(x_{\mathrm{IDM}})} \,,$$

For the cross sections we chose a log transformation, to reduce the range of the target values:

$$\sigma'_{\mathrm{IDM}} = \log\left[\frac{\sigma_{\mathrm{IDM}}}{\min(\sigma_{\mathrm{IDM}})}\right] .$$

To take into account the preprocessing of the target values we used a custom loss function that minimizes the MAPE of the original cross sections.

$$L(\sigma'_{\mathrm{true}}, \sigma'_{\mathrm{pred}}) = \frac{1}{N}\sum_{i=1}^{N} \left|1 - \exp(\sigma'_{\mathrm{pred}} - \sigma'_{\mathrm{true}})\right| ,$$

The training algorithm.

Fixed hyperparameters: Initializer → He Normal, Activation function in each hidden layer → LeakyReLu, Optimizer → Adam.

```
model.add(Dense(neurons, kernel_initializer = he_normal(seed=42), kernel_regularizer=l2(L2lambda)))
model.add(LeakyReLU(alpha=0.2))
model.add(PermaDropout(dropout_fraction))
```

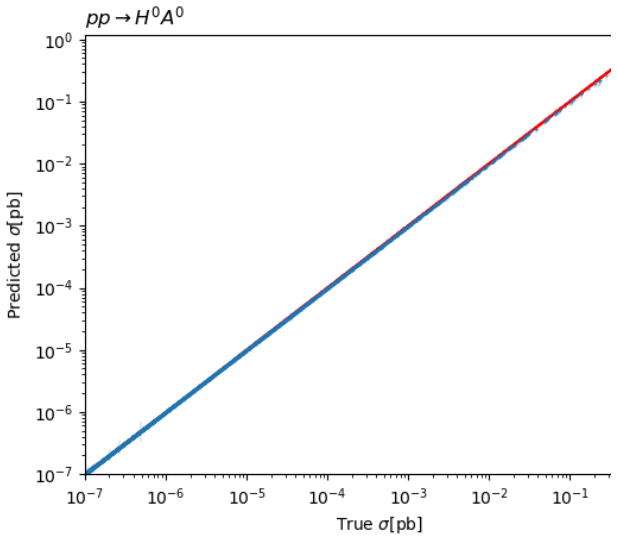
```
from keras import backend as K

from keras.layers.core import Lambda

def PermaDropout(rate):
    return Lambda(lambda x: K.dropout(x, level=rate))
```

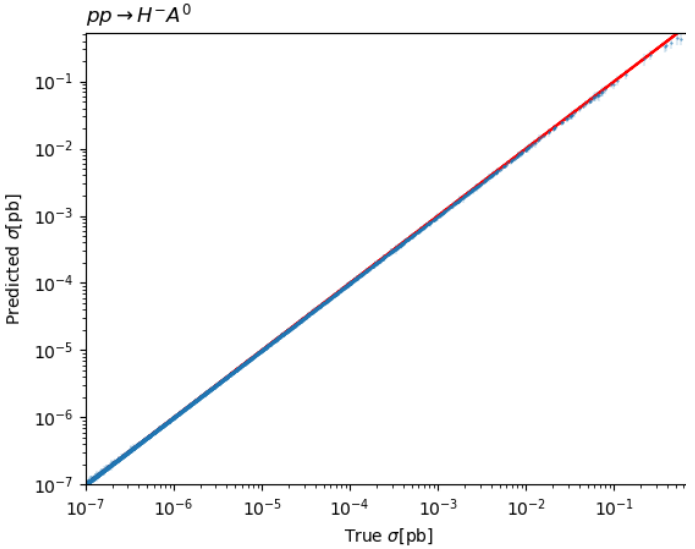
In order to obtain an approximation of the **Bayesian uncertainties** as Monte Carlo dropout a **"permanent" dropout layer** was implemented after each hidden layer, this means that the dropout is present not only during training, but also for inferences.

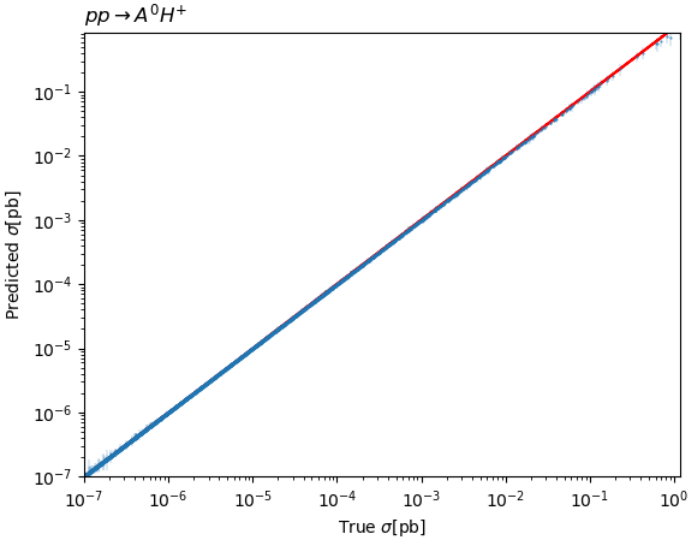
We implemented EarlyStopping callback with a patience of 50. After 500 epochs have ended or EarlyStopping has terminated the iteration, the learning rate was divided by 2 and the training continues until 10 of those iterations were completed.

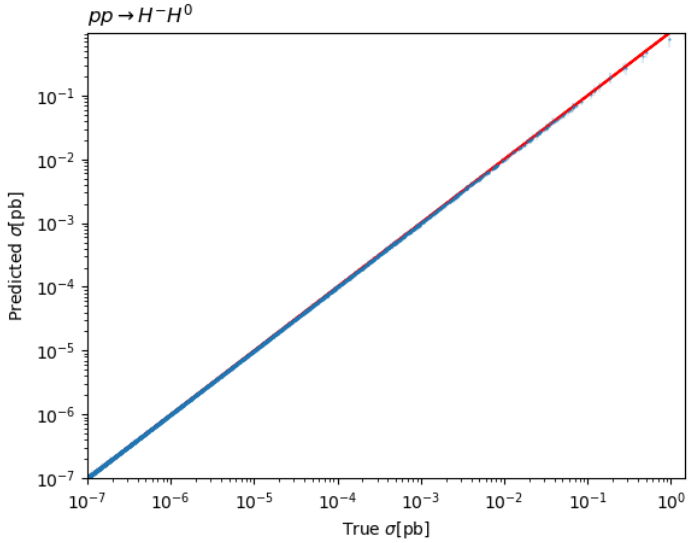
To choose the best configuration, we ran a scan over the rest of the hyperparameter and trained a neural network with each combination for the $pp \to H_0 H_0$. This configuration is formed by 6 hidden layers with 192 artificial neurons, λ of L2 regularization= 10^−5 and a dropout fraction of 1 %

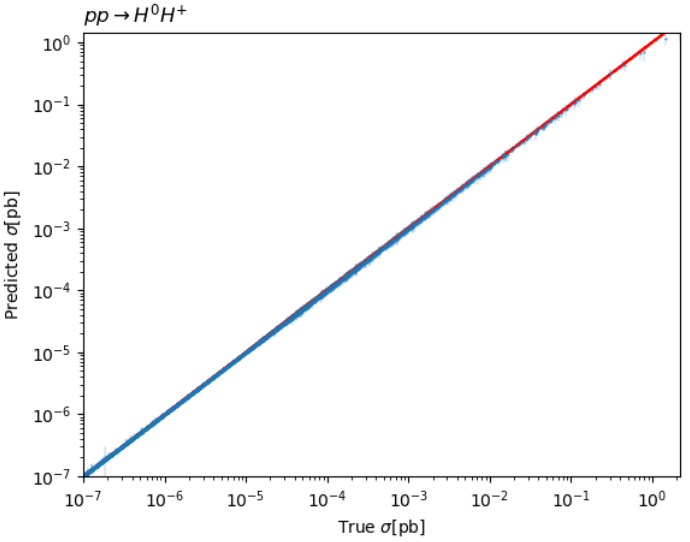
First the best results...



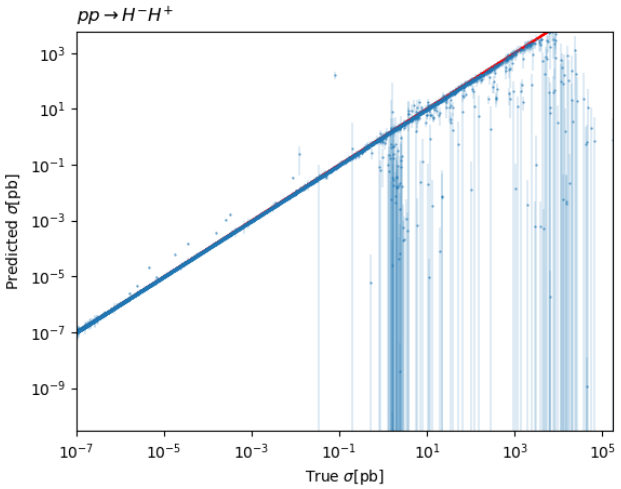


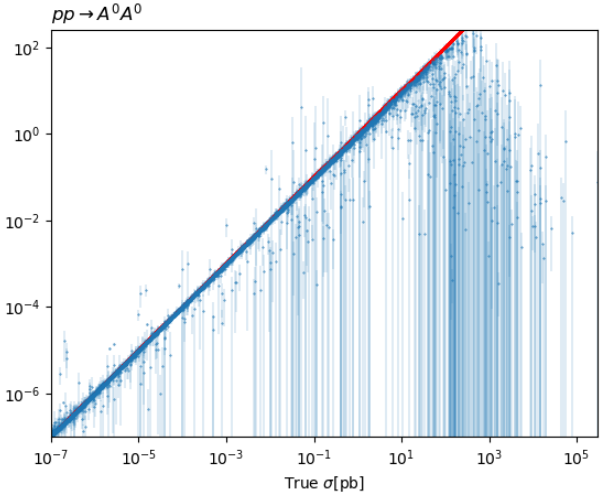


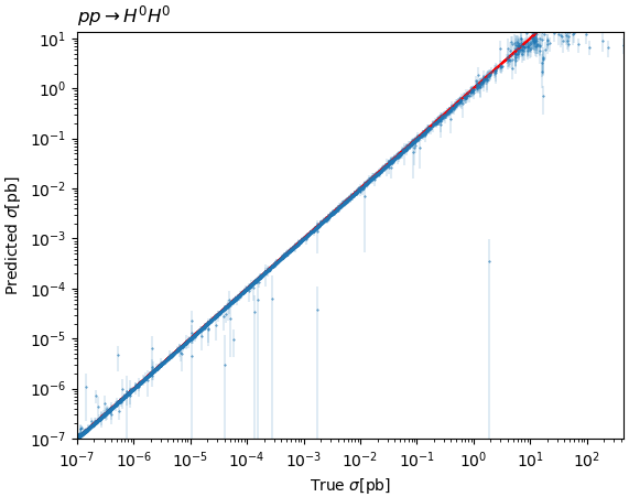


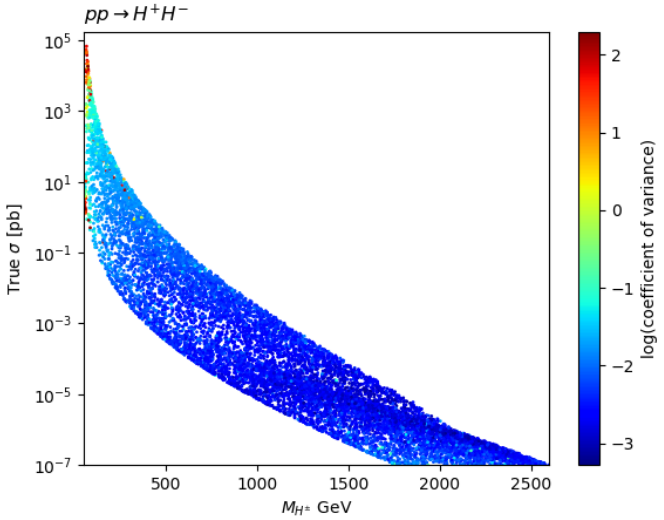


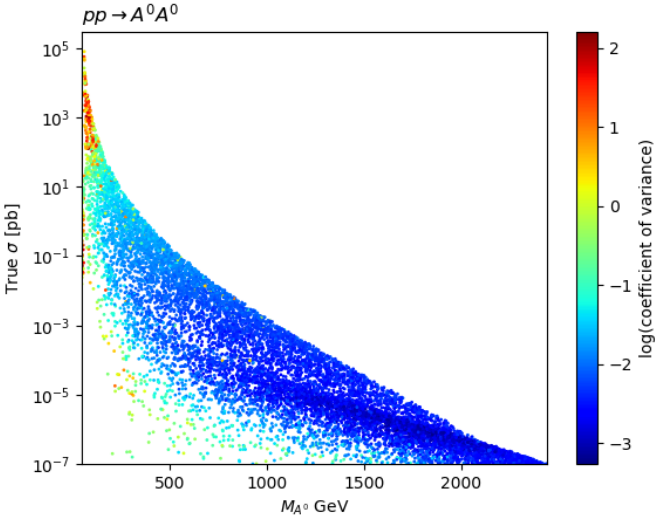
...and now the next to best ones.

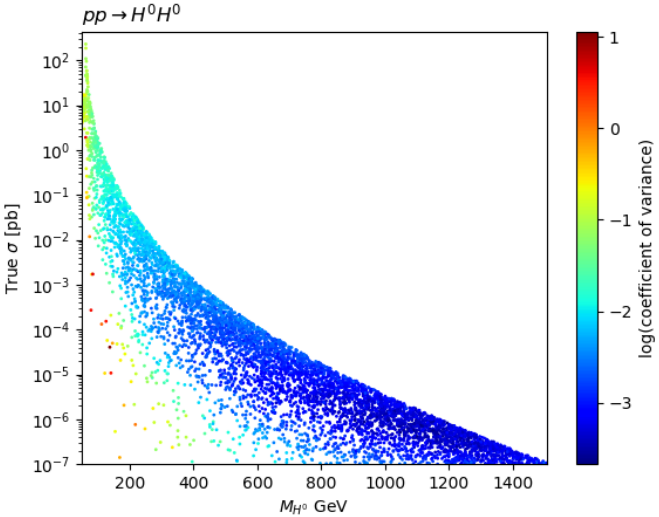












In summary:

- We trained neural networks to predict the cross sections of the production processes of the IDM that include an uncertainty estimation. Results are promising but they can definitely be improved.
- Results suggest that our training data should be more evenly distributed over the target values. Possible solutions: active learning, MCMC,…
- A positive note, is that results show that, in general, the coefficient of variance and relative error are correlated.
- All the material of this project is available at: https://github.com/SydneyOtten/IDM_XS

Pair	H^0H^0	A^0A^0	H^0A^0	H^0H^+	A^0H^+	H^+H^-	H^0H^-	A^0H^-
MAPE	0.03034	0.10488	0.00582	0.00764	0.00479	0.22595	0.00573	0.00716
Mean CV	0.08498	0.18797	0.02717	0.04020	0.02757	0.15080	0.02765	0.02867
within 1σ ratio	0.98328	0.95090	0.99813	0.99950	0.99970	0.98125	0.98861	0.98169

- To be able to apply and thrust these kind of predictors in particle physics, the estimation of the uncertainty is primordial. Furthermore, we need to make sure that these uncertainties are correlated to the relative error of the prediction. The big challenge is to minimize the uncertainty over the full parameter space.

Ah, and one more thing… ->

An open library of classifiers and regressors for HEP phenomenology.

Sascha Caron[IMAPP,Nikhef], Andrea Coccaro[INFN], Sabine Kraml[LPSC], Andre Lessa[UFABC], Sydney Otten[IMAPP,GRAPPA], Humberto Reyes-González[LPSC], Richard Ruiz[CP3], Roberto Ruiz de Austri[IFC], Bob Stienen[IMAPP,Nikhef], Riccardo Torre[INFN]

GOAL: To build a framework in which all material regarding ML applications for particle physics phenomenology can be shared and found for the purpose of education, reproducibility, etc...

Mainly we want:

- A collection of Machine Learning models.
- A collection of Training Data.
- A collection of code to build Machine Learning models.

Thank you!