Deep Neural Network application: 
Higgs boson CP state mixing angle 
in $H \rightarrow \tau\tau$ decay and at LHC

K. Lasocha$^{a,b}$, E. Richter-Was$^a$, M. Sadowski$^c$ and Z. Was$^d$

$^a$ Institute of Physics, Jagellonian University, ul. Lojasiewicza 11, 30-348 Kraków, Poland
$^b$ CERN, Beam Department, 1211 Geneva 23, Switzerland
$^c$ Faculty of Mathematics and Information Technologies, Jagellonian University, ul. Lojasiewicza 6, 30-348 Kraków, Poland
$^d$ Institute of Nuclear Physics, Polish Academy of Sciences, 31-342 Krakow, ul. Radzikowskiego 152, Poland

ABSTRACT

The consecutive steps of cascade decay initiated by $H \rightarrow \tau\tau$ can be useful for the measurement of Higgs couplings and in particular of the Higgs boson parity. In the previous papers we have found, that multi-dimensional signatures of the $\tau^\pm \rightarrow \pi^\pm \pi^0 \nu$ and $\tau^\pm \rightarrow 3\pi^\pm \nu$ decays can be used to distinguish between scalar and pseudoscalar Higgs states. The Machine Learning techniques (ML) of binary classification, offered break-through opportunities to manage such complex multidimensional signatures.

The classification between two possible CP states: scalar and pseudoscalar, is now extended to the measurement of the hypothetical mixing angle of Higgs boson parity states. The functional dependence of $H \rightarrow \tau\tau$ matrix element on the mixing angle is predicted by theory. The potential to determine preferred mixing angle of the Higgs boson events sample including $\tau$-decays is studied using Deep Neural Network. The problem is addressed as classification or regression with the aim to determine the per-event: a) probability distribution (spin weight) of the mixing angle; b) parameters of the functional form of the spin weight; c) the most preferred mixing angle. Performance of methods are evaluated and compared. Numerical results are collected.
1 Introduction

Machine Learning (ML) techniques find increasing number of applications in High Energy Physics phenomenology. With Tevatron and LHC experiments, they became an analysis standards. For the recent reviews see e.g. [1, 2, 3]. Now-days the ML is commonly used for classification, but it is also sometimes explored in the approaches to regression. Let us point to two such examples in LHC experimental analysis. The measurement of polarization fractions in WW pair production using Deep Neural Network (DNN) [4] explores both; the classification [5] and regression [6] potential. The regression technique is also used in [7] for parton distribution functions.

In this paper we present how ML techniques can be helpful to exploit substructure of the hadronically decaying τ leptons in the measurement of the Higgs boson CP-state mixing angle in $H \rightarrow \tau \tau$ decay. This problem of a long history [8, 9], was studied both for electron-positron [10, 11] and for hadron-hadron [12, 13] colliders. Despite these efforts, the Higgs boson CP state was so far not measured at LHC, from $H \rightarrow \tau \tau$ decay. The ML was not even used for the analysis design. Only classical experimental analysis strategies have been prepared and documented, see e.g. [14] for High Luminosity LHC. This is in part, because ML adds complexity to the data analysis. ML solutions need to be investigated in context of their suitability for work on systematic ambiguities.

On the other hand, theoretical basis for the measurement is simple, the cross-section dependence on the parity mixing angle has the form of the first order single angle trigonometric polynomial. It can be implemented in the Monte Carlo simulations as per event spin weight w.r.t. [15] for more details. In [16, 17] we have performed analysis for the three channels of the τ lepton-pair decays, respectively $\rho^0 \nu_\tau \rho^+ \nu_\tau$, $a_1^- \nu_\tau \rho^+ \nu_\tau$, and $a_1^+ \nu_\tau a_1^- \nu_\tau$ but we limited the scope to the scalar-pseudoscalar classification case. We explored the kinematics of outgoing decay products of the τ leptons and geometry of decay vertices.

With these concerns in mind, in the following we extend our previous work on the physics of the Higgs CP parity scalar/pseudoscalar classification, to a measurement of scalar-pseudoscalar mixing angle $\phi^{CP}$ of the $H\tau\tau$ coupling. We do not intend to investigate possible extensions the Standard Model and avoid discussion of the motivations. We constrain ourselves to the measurement of the coupling and the simplest channel of $H \rightarrow \tau^+\tau^- \rightarrow \rho^-\nu_\tau \rho^+\nu_\tau \rightarrow \pi^+\pi^0\nu_\tau \pi^-\pi^0\nu_\tau$ decay. We focus on comparative studies for potential of different ML techniques.

We analyze possible solutions with Deep Neural Network (DNN) algorithms [4] implemented in Tensorflow environment [18] which we have previously found working well for the binary classification [16, 17] between scalar or pseudoscalar Higgs boson variants (which correspond to $\phi^{CP} = 0$ and $\phi^{CP} = \pi/2$). Our goals for the DNN algorithms is to determine per event:

- Spin weight as a function of the mixing angle.
- Decay configuration dependent coefficients, for the known functional form of the spin weight distribution which define the mixing angle dependence.
- The most preferred mixing angle, i.e. where the spin weight is at a maximum.

These goals are complementary or even to large extend redundant, e.g. with functional form of the spin weight we can easily find the mixing angle at which it has a maximum. But the precision of predicting that value would not be necessarily the same for different methods. All three cases are studied as classification and as regression problems. We show quantitative comparison of the performance of DNN learning on the distributions which are relevant for physics analyses and then draw some conclusions.

Paper is organized as follows. In Section 2 we describe basic phenomenology of the problem. Properties of the matrix elements and distributions at the level of final, measurable quantities as well as unmeasurable quantities are presented. In Section 3 we review lists of features (variables) used as an input to DNN and present samples prepared for analyses. As a straightforward extension of [16, 17], still using binary classification, we analyze possibility to distinguish between scalar and arbitrary mix of scalar/pseudoscalar states. This study is covered in Section 4. The multiclass classification approach is covered in Section 5. The regression approach is discussed in Section 6. The comparison of the classification and regression is covered in Section 7. Observations relevant for the future studies of systematic errors are addressed. Summary, Section 8, closes the paper.

In Appendix A more technical details on the DNN architecture are given, input data formulation for classification and regression cases is explained. We also collect figures monitoring performance of DNN training, supporting choices made for the final configuration with which results presented in the main part of the paper were obtained.
2 Physics content of the problem

The most general Higgs boson Yukawa coupling to τ lepton pair, expressed with the help of the scalar–pseudoscalar parity mixing angle $\phi^{CP}$ reads as

$$\mathcal{L}_Y = N \bar{\tau}i\gamma_5\Phi^{CP} + i\sin\phi^{CP}\bar{\tau}\Phi^{CP}\gamma_5\tau,$$

where $N$ denotes normalization, $h$ Higgs field and $\bar{\tau}$, $\tau$ spinors of the $\tau^+$ and $\tau^-$. As we will see later, this simple analytic form translates itself into useful properties of observable distributions convenient for our goal, determination of the $\phi^{CP}$. Recall of the definitions is thus justifiable, and helpful to systematize properties and correlations of the observable quantities (features).

The matrix element squared for the scalar / pseudoscalar / mix parity Higgs, with decay into $\tau^+\tau^-$ pairs can be expressed as

$$|M|^2 \sim 1 + h_{\perp}^2 \cdot R_{i,j}; \quad i, j = \{x, y, z\}$$

(2)

where $h_{\perp}$ denote polarimetric vectors of $\tau$ decays (solely defined by $\tau$ decay matrix elements) and $R_{i,j}$ density matrix of the $\tau$ lepton pair spin state. In Ref. [19] details of the frames used for $R_{i,j}$ and $h_{\perp}$ definition are given. The corresponding CP sensitive spin weight $w_t$ has the form:

$$w_t = 1 - h_{\perp}^2 \cdot h_{\perp}^2 + h_{\perp}^2 \cdot R(2\phi^{CP}) \cdot h_{\perp}^2.$$  

(3)

The formula is valid for $h_{\perp}$ defined in $\tau^\pm$ rest-frames, $h_{\perp}^2$ stands for longitudinal and $h_{\perp}^2$ for transverse components of $h$. The $R(2\phi^{CP})$ denotes the $2\phi^{CP}$ angle rotation matrix around the $z$ direction: $R_{xx} = R_{yy} = \cos 2\phi^{CP}$, $R_{xy} = -R_{yx} = \sin 2\phi^{CP}$. The $\tau^\pm$ decay polarimetric vectors $h_{\perp}^i$, $h_{\perp}^j$, in the simplest case of $\tau^\pm \rightarrow \pi^\pm\pi^0\nu$ decay, read

$$h_{\perp}^i = \mathcal{N}\left(2(q \cdot p_{\nu})q' - q^2 p_{\nu}^2\right),$$

(4)

where $\tau$ decay products $\pi^\pm$, $\pi^0$ and $\nu_{\tau}$ 4-momenta are denoted respectively as $p_{\pi^\pm}$, $p_{\pi^0}$, $p_{\nu}$ and $q = p_{\pi^\pm} - p_{\pi^0}$. Obviously, complete CP sensitivity can be extracted only if $p_{\nu}$ is known (for $\tau^\pm \rightarrow \pi^\pm\pi^\mp\pi^0\nu$ formula is longer, dependence on modeling of the decay appear too [20], but is of no principle differences).

Note that the spin weight $w_t$ is a simple first order trigonometric polynomial in a $2\phi^{CP}$ angle. This observation is valid for all $\tau$ decay channels. For the clarity of the discussion on the DNN results, we introduce $\alpha^{CP} = \phi^{CP}$, which spans over $(0, 2\pi)$ range. The $\alpha^{CP} = 0, 2\pi$ corresponds to scalar state, the $\alpha^{CP} = \pi$ to pseudoscalar one.

Spin weight can be expressed as

$$w_t = C_0 + C_1 \cdot \cos(\alpha^{CP}) + C_2 \cdot \sin(\alpha^{CP}),$$

(5)

where

$$\begin{align*}
C_0 & = 1 - h_{\perp}^2 \cdot h_{\perp}^2, \\
C_1 & = -h_{\perp}^x h_{\perp}^y - h_{\perp}^y h_{\perp}^x, \\
C_2 & = -h_{\perp}^x h_{\perp}^y - h_{\perp}^y h_{\perp}^x,
\end{align*}$$

(6)

depend on the $\tau$ decays only.

Distribution of the $C_0, C_1, C_2$ coefficients, for the sample of $H \rightarrow \tau\tau$ events used for our numerical results is shown in Fig. [1]. The $C_0$ spans $(0, 2)$ range, while $C_1$ and $C_2$ of $(-1, 1)$ range have a similar shape, quite different than the one of $C_0$.

The amplitude of the $w_t$ as function of $\alpha^{CP}$ depends on the multiplication of the length of the transverse components of the polarimetric vectors. The longitudinal component $h_{\perp}^x h_{\perp}^y$ is defining shift with respect to zero of the $w_t$ mean value over a full $(0, 2\pi)$ range. The maximum of the $w_t$ distribution is reached for $\alpha^{CP} = \angle(h_{\perp}^x, h_{\perp}^y)$, the opening angle of transverse components of the polarimetric vectors.

The spin weight of formula (5) can be used to introduce transverse spin effects into the event sample when for the generation transverse spin effects were not taken into account at all. The above statement is true, independently if longitudinal spin effects were included and which $\tau$ decay channels complete cascade of $H \rightarrow \tau\tau$ decay. The shape of weight dependence on the Higgs coupling to $\tau$ parity mixing angle is preserved.

In Fig. [2] we show distribution of spin weight $w_t$ for five example $H \rightarrow \tau\tau$ events collected in Table [1]. For each event, depending on the polarimetric vectors, single value of $\alpha^{CP}$ is preferred (by the largest weight). For a physics
Table 1: Polarimetric vectors, resulting $C_i$ coefficients of formulas \( i \) and angle $\angle(h^T_+,h^T_-)$ between transverse components of polarimetric vectors for five example events of $H \rightarrow \tau^+\tau^-,\tau^\pm \rightarrow p^\pm v_\tau$. In brackets, angle of only hadronic part of polarimetric vector is given.

| Events | Polarimetric vectors | $|h^T_+||h^T_-|$ | $C_0$ | $C_1$ | $C_2$ | $\angle(h^T_+,h^T_-)$ [rad] (hadronic part only) |
|--------|----------------------|----------------|------|------|------|-------------------------------|
| Event 1 | $h^{\text{CP}}_{+,-} = (0.7547 -0.2232 -0.6167)$
$h^{\text{H}}_{+,-} = (-0.9093 -0.2931 -0.2953)$ | 0.7519 | 0.8179 | 0.7517 | 0.0183 | 6.2586 |
| Event 2 | $h^{\text{CP}}_{+,-} = (0.8617 0.0485 0.5050)$
$h^{\text{H}}_{+,-} = (-0.5959 0.7892 -0.1487)$ | 0.8535 | 1.0751 | 0.5518 | -0.6511 | 5.4134 |
| Event 3 | $h^{\text{CP}}_{+,-} = (0.3402 0.9377 -0.0682)$
$h^{\text{H}}_{+,-} = (0.8262 0.1272 -0.5487)$ | 0.8339 | 0.9626 | -0.1619 | -0.8180 | 5.2130 |
| Event 4 | $h^{\text{CP}}_{+,-} = (-0.6964 0.6204 -0.3605)$
$h^{\text{H}}_{+,-} = (0.2142 -0.3885 -0.8962)$ | 0.4138 | 0.6769 | -0.0919 | -0.4035 | 4.4883 |
| Event 5 | $h^{\text{CP}}_{+,-} = (0.1115 -0.4989 -0.8595)$
$h^{\text{H}}_{+,-} = (-0.2347 -0.01108 0.9720)$ | 0.1201 | 1.8354 | 0.0317 | -0.1158 | 4.9793 |

For compatibility with our previous publications \([16,17]\), we use the same generated event samples, namely Monte Carlo events of the Standard Model, 125 GeV Higgs boson, produced in pp collision at 13 TeV centre-of-mass energy, generated with \texttt{Pythia 8.2} \([21]\) and with spin correlations introduced with \texttt{TauSpinner} \([15]\) package. For $\tau$ leptons we use \texttt{Tauolapp} library \([22]\). All spin and parity effects are implemented with the help of weight $wt$ \([23,24]\). The sample is generated without spin effects, and the spin weights $wt_i$ for few different values of CP mixing angle $\alpha^{\text{CP}}$ are stored. Spin weight, formula \([3]\), is calculated using $R_{i,j}$ density matrix and polarimetric vectors $h^\pm$.

Later, for a given event it is possible to calculate coefficients $C_0,C_1,C_2$, using three $\alpha^{\text{CP}}$ and linear equation \([5]\). Fig.\([4]\) shows the cross-check how well this procedure works. The functional form (orange line) and evaluated spin weights (blue dots) for two example events are shown. The $C_0,C_1,C_2$ coefficients for the functional form are
calculated solving Eq. 5 for $\omega$ stored in the generated event samples at three values of $\alpha^{CP}$.

In this paper we present results for the case when both $\tau$'s decay $\tau^\pm \rightarrow \rho^\pm \nu_\tau$ and about $10^6$ simulated Higgs events are used. To partly emulate detector conditions, a minimal set of cuts is used. We require that the transverse momenta of the visible decay products combined, for each $\tau$, are larger than 20 GeV. It is also required that the transverse momentum of each $\pi^\pm$ is larger than 1 GeV.

The emphasis of the paper is to explore different ML approaches to the problem, and we discuss only the case of the Variant-All feature list from paper [17]. The four-momenta of all decay products of $\tau$ leptons defined in the rest frame of intermediate resonance pairs, and with sum of hadronic decay products aligned with $z$-axis are used. This represents an ideal benchmark case scenario, for performance monitoring.

### 4 Binary classification

The use of the DNN for binary classification have been discussed in our previous papers [16, 17]. The focus was on discriminating between CP-scalar ($\mathcal{H}_0$ hypothesis) and CP-pseudoscalar ($\mathcal{H}_1$ hypothesis).

Now we apply the same procedure but with alternative hypothesis ($\mathcal{H}_{\alpha^{CP}}$) representing the scalar-pseudoscalar mixed state of mixing parameter $\alpha^{CP}$. To quantify performance for Higgs CP state classification the weighted Area Under Curve (AUC) [25, 26] is used again. For each simulated event we know also Bayes optimal probability that it is sampled from $\mathcal{H}_0$ or $\mathcal{H}_{\alpha^{CP}}$ hypothesis, see more detailed description in Appendix A. This forms the so called
Table 2: The AUC scores for discriminating between Higgs CP states. Results from oracle predictions and binary classification for discriminating between $H_0$ hypothesis that Higgs CP is a scalar (CP-mixing angle $\alpha_{CP} = 0.0$ or $2\pi$) and $H_{\alpha_{CP}}$ hypothesis, when Higgs CP is of a parity mixed state, are shown. CP-mixing angle $\alpha_{CP} = \pi$ corresponds to pseudo-scalar case.

<table>
<thead>
<tr>
<th>CP-mixing angle $\alpha_{CP}$ (units of $\pi$)</th>
<th>Oracle predictions</th>
<th>Binary classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.528</td>
<td>0.525</td>
</tr>
<tr>
<td>0.4</td>
<td>0.605</td>
<td>0.595</td>
</tr>
<tr>
<td>0.6</td>
<td>0.699</td>
<td>0.684</td>
</tr>
<tr>
<td>0.8</td>
<td>0.775</td>
<td>0.756</td>
</tr>
<tr>
<td>1.0</td>
<td>0.804</td>
<td>0.784</td>
</tr>
</tbody>
</table>

*oracle predictions*, i.e. ultimate discrimination for this problem. We calculate oracle predictions and evaluate the results of DNN. This is a straightforward extension of the method used in [16][17]. That is why, simple attempt on future discussion of systematic error may follow that suggested in [17]: variations within expected range of detector response can be easily introduced and biases studied.

The oracle predictions for discriminating between $H_0$ and $H_{\alpha_{CP}}$ hypotheses is increasing with $\alpha_{CP}$ and reach AUC=0.78 for $\alpha_{CP} = \pi$. The performance of DNN is following similar pattern, reaching maximum at $\alpha_{CP} = \pi$ (pure pseudo-scalar case). It decreases for smaller or larger $\alpha_{CP}$, where admixture of the scalar component appear. In case of complete feature list, it is almost achieving the performance of oracle predictions. In Fig. 4 the AUC values are plotted for full $\alpha_{CP}$ range. The distributions are (almost) symmetric around $\alpha_{CP} = \pi$. Note that the functional form of spin weight $w_t$, Eq. (5), encapsulating sensitivity to $\alpha_{CP}$ is not symmetric, see Fig. 3 In Table 2 we show numerical results for few $\alpha_{CP}$.

5 Multiclass classification

The binary classification discussed in previous Section is easy to generalize to the multiclass case. The DNN is learning to provide per-event probabilities to associate with each class. Single class represents either discrete point or a specific range in 1-dimensional parameter space. We explore three approaches, each providing complementary physics information, but all allowing to quantify, on the per-event basis, which is the preferred mixing angle of the studied Higgs sample:

- The DNN classifier is learning per-event spin weight as a function of mixing angle $\alpha_{CP}$. The range of mixing angle $(0, 2\pi)$ is discretised into equally spaced points called classes. This approach is described in
Section 5.1 and used for the figures labeled with: Classification: wt.

- The DNN classifier is learning per-event coefficients $C_0, C_1, C_2$. The allowed range of coefficients is split into several equal size ranges (classes), single class represents a range for a coefficient value. The DNN is trained for each coefficient separately. This approach is described in Section 5.2 and used for the figures labeled with: Classification: $C_0, C_1, C_2$.

- The DNN classifier is learning per-event most probable mixing angle $\alpha_{CP}^{\text{max}}$, i.e. value of $\alpha^{CP}$ at which spin weight is maximal. The range of mixing angle $(0, 2\pi)$ is split into several equally spaced points (classes). This approach is described in Section 5.3 and used for the figures labeled with: Classification: $\alpha_{CP}^{\text{max}}$.

We monitor performance of the learning process in a standard manner, with the loss function on the training and validation sets. Respective distributions are shown in Fig. 21 of Appendix A. Note that the loss function, the tf.nn.softmax_cross_entropy_with_logits of the Tensorflow, allows to predict probabilities of the class labels, and not the actual value of the observable at a given class. In case of predicting spin weight distribution, only the normalized to unity shape is predicted. In case of predicting values of $C_i$ coefficients or $\alpha_{CP}^{\text{max}}$, vector of probabilities is returned, and the one-hot encoding transformation selecting most probable class is then applied to retrieve actual predicted value of the parameter.

### 5.1 Learning spin weight $wt$

The DNN classifier is trained with per-event feature list and as a label normalized to unity $N_{\text{class}}$-dimensional vector of spin weight

$$wt_i^{\text{norm}} = wt_i / \sum_{i=1}^{N_{\text{class}}} wt_i$$

is given, each component of $wt^{\text{norm}}(\alpha^{CP})$ vector corresponds to the i-th discrete value of mixing angle $\alpha_i^{CP}$. $N_{\text{class}}$ denotes number of points to which range $(0, 2\pi)$ was discretised. The number of classes is kept odd, to assure that $\alpha^{CP} = 0, \pi, 2\pi$, corresponding respectively to scalar/pseudoscalar/scalar cases, are always represented as a class. Training of DNN is performed with $N_{\text{class}}$ varying from 3 to 51, to monitor how performance depends on number of classes used.

We quantify the DNN performance for classification problem in the context of physics relevant criteria. The first is how well DNN is able to reproduce per-event shape of the spin weight $wt^{\text{norm}}$. For two example events, true and predicted spin weight $wt^{\text{norm}}$ distribution is shown in Fig. 5 as a function of class index i (representing discretised mixing parameter $\alpha_i^{CP}$). Multiclass classification with $N_{\text{class}} = 21$ was performed. Blue dots denote true weights while orange diamonds denote weights predicted by DNN classifier. In general, for some events agreement is very good, while for others DNN fails to classify correctly probability (likeness) with which an event belongs to a given class. In overall, predicted weights follow smoothly, expected shape of linear $\cos(\alpha^{CP})$ and $\sin(\alpha^{CP})$ combination. This is encouraging, because the loss function is not correlating explicitly nearby classes. The DNN is discovering this pattern in the process of learning.

To quantify those observations, performance of DNN is monitored on the statistical basis with $I_2$ norm. The $I_2$ norm is defined as a mean squared difference between predicted $p_{i,k}$ and true $wt_{i,k}^{\text{norm}}$, summed over each class $i$ and event $k$. It is averaged over the number of events $N_{\text{evt}}$ and number of classes $N_{\text{class}}$.

$$I_2 = \frac{1}{N_{\text{class}}} \sum_{k=1}^{N_{\text{evt}}} \sum_{i=1}^{N_{\text{class}}} \left( \frac{wt_{i,k}^{\text{norm}} - p_{i,k}}{N_{\text{class}}} \right)^2$$

(7)

The $p_{i,k}$ is returned by k-th event i-th output of neural network. The $N_{\text{class}}$ factor on the right side of formula (7), scales it back i.e. normalization of $I_2$ becomes $N_{\text{class}}$ independent, which allows for comparison between distinct $N_{\text{class}}$. Distribution of $I_2$ norm is shown in Figure 6 as a function of class multiplicity $N_{\text{class}}$. With increasing number of classes, $I_2$ is flat at the value of about 0.2, indicating that with available statistics DNN is learning well independently of class granularity i.e. $N_{\text{class}}$. We have no explanation yet why $I_2$ remains close to 0.2.

From physics perspectives, learning the shape of $wt$ distribution as function of $\alpha^{CP}$, is equivalent to learning components of the polarimetric vectors. But, because only the shape, not the normalization, is available the $C_i$ coefficients cannot be fully retrieved from formula (5). It is not necessary the aim anyway. The physics interest is to learn $\alpha^{CP}$ which is preferred by events of the analyzed sample, i.e. value at which $wt$ distribution has its maximum. This corresponds to determining CP mixing of the analyzed sample.

\footnote{The $wt_i$ remains in the (0,4) range, as explained in [24].}
Figure 5: Normalized to probability spin weight $w_{\text{norm}}$, predicted (orange diamonds) and true (blue dots), as a function of the class index for two example events (left and right plots). DNN was trained with $N_{\text{class}} = 21$ spanning range $(0, 2\pi)$.

Figure 6: The $l_2$ norm, quantifying difference between true and predicted spin weight $w_{\text{norm}}$, as a function of class multiplicity $N_{\text{class}}$. 
The second criterium is the difference between most probable predicted class and most probable true class, denoted as $\Delta_{\text{class}}$. When calculating difference between class indices, periodicity of the functional form is taken into account. Class indices represent discrete values of $\alpha_{\text{CP}}$, in range $(0, 2\pi)$. The distance between the first and the last class is zero. We take the distance which corresponds to the smaller angle difference and we take the sign according to clock-wise orientation vs class index at which true $wt$ has its maximum.

Let’s $idp_{\text{max}}$ denote the index of most probable predicted class, $idc_{\text{max}}$ be index of true most probable class. The distance $|\Delta_{\text{class}}|$ is defined as:

$$|\Delta_{\text{class}}| = \min ((|idp_{\text{max}} - idc_{\text{max}}|), ((N_{\text{class}} - 1) - |idp_{\text{max}} - idc_{\text{max}}|)), \quad (8)$$

and the sign is attributed

$$\Delta_{\text{class}} = \text{sign}(idp_{\text{max}} - idc_{\text{max}}) |\Delta_{\text{class}}|. \quad (9)$$

In Fig. 7 distributions of $\Delta_{\text{class}}$ for $N_{\text{class}} = 21$ and 51 respectively are shown. The shapes are Gaussian-like and centered around zero. The mean $<\Delta_{\text{class}}>$ = 0.03 [rad] and this we can interpret as the bias of the method. The standard deviation of per-event distribution is $\sigma_{\Delta_{\text{class}}} = 0.5$ [rad]. Those parameters are quite stable with $N_{\text{class}}$ while it exceeds 21. Discretising into even more classes is not improving precision of the $\alpha_{\text{CP}}$ determination. Further optimization of the DNN performance should focus how to reduce both the bias and standard deviation of the $\Delta_{\text{class}}$ distributions.

Figure 7: Distribution of $\Delta_{\text{max}}$ between predicted most probable class and true most probable class for $N_{\text{class}} = 21$ and 51 respectively. The mean and std are calculated in units of class index [idx] or units of radians [rad].

Figure 8: Fractions of correctly classified events, i.e. such that $|\Delta_{\text{class}}| < 1$ to 4 respectively (left plot) or $|\Delta_{\text{CP}}^{\text{max}}| < 0.25$ to 1.0 [rad] (right plot), as a function of $N_{\text{class}}$. Note that step-like structure on the right-side plot is of the technical origin, for details see in the text.
The limiting precision with which $\Delta^{CP}_{\text{max}}$ can be determined with DNN classification is the granularity with which range $(0,2\pi)$ was discretised. The distance between individual classes is equal to $2\pi/N_{\text{class}}$ [rad]. For $N_{\text{class}} = 51$ it corresponds to 0.125 [rad]. However, having predicted binned $wt^{\text{norm}}$, one can apply standard fitting procedure to more precisely find its maximum, and to reduce limiting effect from the $N_{\text{class}}$ granularity. This was not tried in the scope of the paper, but certainly is a possible extension to follow.

The third criterium is the fraction of events which are correctly classified, i.e. such that $\Delta_{\text{class}}$ is not outside tolerance $\pm \Delta^{\text{max}}_{\text{class}}$. The fraction of correctly classified events is shown in Fig. 8, as a function of $\alpha$. The fraction of events falling into the range $\pm \Delta^{\text{max}}_{\text{class}}$ decreases steeply as $N_{\text{class}}$ increases, for the thresholds $|\Delta_{\text{class}}| < 1$, 2 but becomes flatter for $|\Delta_{\text{class}}| < 3$. For $|\Delta_{\text{class}}| < 4$ and $N_{\text{class}} = 51$, the fraction is still around 90%. The step-like structure visible on the right-plot is a consequence of the threshold $|\Delta_{\text{class}}|$ in [rad] crossing the $|\Delta_{\text{class}}|$ boundaries in [idx]. For the threshold $|\Delta^{CP}_{\text{max}}| < 0.25$ [rad] and $N_{\text{class}} = 51$ the probability for correctly classified event is about 80%.

The DNN classifier which is predicting normalized spin weight $wt^{\text{norm}}$, provides enough information to identify the most probable mixing angle $\Delta^{CP}_{\text{max}}$. The information is not sufficient to reconstruct complete set of $C_i$ coefficients and the polarimetric vectors though.

### 5.2 Learning $C_0, C_1, C_2$ Coefficients

The second approach is to learn formula $\Delta^{CP}_{\text{max}}$ coefficients $C_0, C_1, C_2$ for the spin weight $wt$. They can be then used to predict $wt$ and $wt^{\text{norm}}$. Coefficients $C_0, C_1, C_2$ represent physical observables, products of longitudinal and transverse components of polarimetric vectors, as shown in formulas $\Delta^{\text{class}}$.

The classification technique using DNN is configured to learn each of the $C_i$ with separate training. The allowed range is well known, the $C_0$ spans the range $(0.0, 2.0)$ and $C_1, C_2$ the range $(-1.0, 1.0)$, see Fig. 1. The allowed range is binned into $N_{\text{class}}$ and as a label, the $N_{\text{class}}$-dimensional vector with one-hot encoded value of the $C_i$ parameter is associated with each event. In this case, single class represents range of the $C_i$ coefficient. During training, the DNN is learning per-event association between feature list and the class labels. The output is probability $N_{\text{class}}$-dimensional vector, it is converted to one-hot encoded representation, i.e. the most probable class is chosen as a predicted value of the $C_i$ coefficient.

Distributions of the difference between true and predicted $C_i$ coefficients are shown in Figs. 9. In that case, as there is no periodicity involved, $\Delta_{\text{class}} = idp - idc$ where $idp, idc$ denote respectively true and predicted class index. The shape is Gaussian-like. Mean of $\Delta C_i$ is close to zero and standard deviation is of 0.10-0.15, which is less than 10% of the range. Precision with which $C_i$ coefficients are predicted is limited by the $N_{\text{class}}$.

We use the true and predicted $C_0, C_1, C_2$ coefficients to calculate $wt$ distribution of $\Delta^{CP}_{\text{max}}$. It is then discretised with $N_{\text{class}}$ points (the $N_{\text{class}}$ could be different than the one used for learning coefficients), and the $\Delta^{CP}_{\text{max}}$ is determined from the class of maximal weight. The difference between true and predicted $\Delta^{CP}_{\text{max}}$ is shown in Fig. 10 for $N_{\text{class}} = 21$ and 51. The Gaussian-like shape of those distributions, centered around zero, clearly demonstrated that method works as expected. The mean and standard deviation of the distributions are close to those obtained with Classification: $wt$ approach, of Fig. 7.
5.3 Learning the $\alpha_{\text{CP}}^{\text{max}}$

The third approach is to directly learn per-event most preferred mixing angle, $\alpha_{\text{CP}}^{\text{max}}$. The allowed range $(0, 2\pi)$ is again binned into $N_{\text{class}}$ classes, where single bin represents discrete $\alpha_{\text{CP}}$. For training, as a label the one-hot encoded vector of $N_{\text{class}}$-dimension is associated with each event. The DNN is returning $N_{\text{class}}$-dimensional vector of probabilities, which is converted to single one, with the class of the highest probability which is selected as a predicted $\alpha_{\text{CP}}^{\text{max}}$. With this approach, neither spin weight nor $C_i$ coefficients are predicted.

As the event sample is generated without any CP mixture favoured, the distribution of the $\alpha_{\text{CP}}^{\text{max}}$ is expected to be uniform, and such sanity check is demonstrated in the left plot of Fig. 11. The DNN is well reproducing this behaviour. The $\Delta \alpha_{\text{max}}$, the difference between true and predicted value of the $\alpha_{\text{CP}}^{\text{max}}$ is shown in the right plot of Fig. 11. It has a Gaussian-like shape with the mean $<\Delta \alpha_{\text{CP}}^{\text{max}}>_0 = 0.005 \pm 0.002$ [rad] and standard deviation 0.447 [rad]. This was obtained with $N_{\text{class}} = 21$. Results are comparable with the ones obtained with the previously discussed approaches.
Figure 12: Example plots with DNN regression results: the spin weight $wt$, predicted (orange diamonds) and generated (blue dots), as a function of the mixing angle $\alpha_{\text{CP}}$ for two example events.

6 Regression

The ML regression is not so commonly used in the high energy physics analyses. For classification, the goal is to produce discrete assignments, of a given event to a given class, or soft assignment i.e. probability distribution of the class assignment. We explore three approaches, defined similarly as in Section 5 but now as a regression problem, that is for continuous variables:

- The DNN is learning to predict per-event spin weight as a function of mixing angle $\alpha_{\text{CP}}$. The range of mixing angle $(0, 2\pi)$ is split into discrete points of $\alpha_{\text{CP}}$ at which value of spin weight is learned. This approach is described in Section 6.1 and used for the figures labeled with: Regression:wt.

- The DNN is learning to predict per-event value of the coefficients $C_0, C_1, C_2$ of the functional form (5). The DNN is trained for all coefficients simultaneously. This approach is described in Section 6.2 and used for the figures labeled with: Regression:$C_0, C_1, C_2$.

- The DNN is learning to predict per-event most probable mixing angle $\alpha_{\text{CP}}^{\text{max}}$, i.e. where $\alpha_{\text{CP}}$ spin weight has maximum. This approach is described in Section 6.3 and used for the figures labeled with: Regression:$\alpha_{\text{CP}}^{\text{max}}$.

For the studies we continue with Tensorflow package, but now with tf.losses.mean_squared_error function to define the loss in the training procedure of Section 6.1, 6.2 and the tf.reduce_mean function in the training procedure of Section 6.3.

6.1 Learning spin weight $wt$

Similarly as in the classification case, the DNN regression is trained on an input information consisting of per-event feature list and as a label a vector of the spin weight $wt_i$ for the discrete values of $\alpha_{\text{CP}}$. Training is performed for different granularities of $\alpha_{\text{CP}}$ discretisation, to monitor performance sensitivity. Also in this case we use odd number of equally spaced points $\alpha_{\text{CP}}^j$, so the $\alpha_{\text{CP}} = 0, \pi, 2\pi$ coincide with the points. In case of regression, both shape and normalization of the $wt$ are learned by the DNN.

For two example events in Fig. 12 spin weight $wt$ distribution is shown as a function of binned mixing parameter $\alpha_{\text{CP}}$. Generally, for some events agreement is very good, while for others, DNN fails to predict correctly the shape, normalization or both for the spin weight $wt$. In overall, predicted weights follow smoothly expected shape of linear $\cos(\alpha_{\text{CP}})$ and $\sin(\alpha_{\text{CP}})$ combination, even if no attempt to regularize for such smooth behaviour was made.

Distributions of $l_2$ norm, defined similarly as in the classification case, as a function of $N_{\text{class}}$ (granularity for discretising $\alpha_{\text{CP}}$) is shown in Figure 13. For more compatibility with the classification case of Section 5.1 it is calculated from $wt$ or normalized to unity $wt_{\text{norm}}$. With increasing number of bins, $l_2(wt)$ and $l_2(wt_{\text{norm}})$ become flat and equal to about 0.20.
Figure 13: The $l_2$ norm for predicted spin weight $wt$ (orange diamonds) and $wt^{\text{norm}}$ (black dots) as a function of $N_{\text{class}}$.

Figure 14: Distribution of $\Delta_{\text{class}}$ between most probable predicted class and true most probable class. The $N_{\text{class}} = 21$ and 51 are used for respectively left and right plot. The mean and std standard deviation are calculated in units of class index [idx] and units of radians [rad].

Figure 15: Fraction as a function of $N_{\text{class}}$ of events with $|\Delta_{\text{class}}| < 1$ to 4 (left plot) and of events with $|\Delta \alpha_{\text{max}}^{CP}| < 0.25$ to 1.0 [rad] (right plot).
Regression: $C_0, C_1, C_2$

mean = $0.019 \pm 0.001$
std = $0.119$

Figure 16: Difference between true and predicted coefficients $C_0, C_1, C_2$ of formula (5).

Figure 17: The $l_2$ norm as a function of $N_{class}$, number of points used for discretising $\alpha^{CP}$ range, for predicted and true $wt$ and $wt^{norm}$.

In Fig. [14] distributions of $\Delta_{class}$ for $N_{class} = 21$ and 51 used to train DNN regression are respectively shown. The shape is Gaussian-like and as expected centered around $\Delta_{class}=0$.

Similarly as in classification case, we monitor probability that true and predicted index of point (class) with maximal spin weight differs by no more than certain tolerance $\pm \Delta_{class}$. This distribution is shown in Fig. [15] respectively for $\Delta_{class} < 1$ to 4. The fraction of events falling into the fixed range $\pm \Delta_{class}$ decreases with $N_{class}$. The optimal performance of DNN should maximize those fractions.

6.2 Learning $C_0, C_1, C_2$ coefficients

Distributions of the true $C_0, C_1, C_2$ coefficients are shown in Fig. [1] The differences between true and predicted ones are shown in Figs. [16] On average, all three coefficients are predicted reasonably well. Consistent are the statistical summaries of $\Delta C_i$: means remain in the range (0.01, 0.02) and standard deviations in (0.12-0.16). Coefficients $C_i$ are then used to calculate predicted spin weight $wt$ of formula (5).

For consistency, we evaluate DNN performance using the same criteria as for classification approaches. This is achieved by using coefficients $C_0, C_1, C_2$ to calculate spin weight $wt$, and then turning it into discrete predictions for $wt$ and $wt^{norm}$ in the $N_{class}$ points. As in Section [5] for classification approach, we use $l_2$ norm, and $\Delta_{class}$, as defined by formulas (7) and (9). The $l_2(wt)$ and $l_2(wt^{norm})$ distributions are shown in Fig. [17] As expected they are flat as a function of granularity $N_{class}$, this is because true and predicted $wt$ are calculated from functional form and the difference is averaged over all events and full range of $\alpha^{CP}$. The $l_2(wt)$ includes also discrepancy due to absolute normalization of $wt$, while $l_2(wt^{norm})$ accounts only for the shape differences.

The distributions of the true and predicted most probable class, $\alpha_{max}^{CP}$ and their difference are shown in Figs. [18] for the $N_{class} = 51$. We expect the distributions to be flat as sample was generated without any polarization correlation (carrier of CP effects) included. The difference between true and predicted $\alpha_{max}^{CP}$ forms a narrow peak with the mean value $< \Delta \alpha_{max}^{CP} >= 0.029 \pm 0.002$ [rad] and standard deviation 0.447 [rad].
6.3 Learning the $\alpha_{\text{CP}}^{\text{max}}$

This implementation of the regression method allows to learn per-event most preferred mixing angle, $\alpha_{\text{CP}}^{\text{max}}$ directly. As the used event sample is generated without any polarization, the distribution of the $\alpha_{\text{CP}}^{\text{max}}$ is expected to be uniform, see the left plot of Fig. 19. The DNN is reproducing well this feature.

The difference between true and predicted $\alpha_{\text{CP}}^{\text{max}}$ forms a narrow peak with the mean $<\Delta\alpha_{\text{CP}}^{\text{max}}>= -0.009 \pm 0.003$ [rad] and standard deviation 0.685 [rad]. This is less precise estimate, than obtained with method of Section 6.2.

7 Classification or regression: comparison and complementarity

In this Section we shortly compare classification and regression approaches. In Table 3 we collect the mean and standard deviation for difference between true and predicted with classification and regression methods $C_i$. There is no clear winner, both methods give predictions of similar precision. In Table 4 we compare the difference between true and predicted $\alpha_{\text{CP}}^{\text{max}}$ obtained with different methods. With the classification method comparable performance is achieved when learning spin weight $w_t$, coefficients $C_0, C_1, C_2$ or directly $\alpha_{\text{CP}}^{\text{max}}$. For the regression method learning directly $\alpha_{\text{CP}}^{\text{max}}$ is less perfomant. Also in this case there is no clear winner of better precision than for other methods.

Fig. 20 shows overlaid distributions for probabilities that $|\Delta_{\text{class}}| < 1$ to 4 (top plot) or $|\Delta\alpha_{\text{CP}}^{\text{max}}| < 0.25$ to
Table 3: The mean and standard deviations of $\Delta C_i$, the difference between generated and predicted $C_i$, obtained from DNN with classification and regression methods.

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Classification</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta C_0$</td>
<td>mean = -0.003 ± 0.001 std = 0.114</td>
<td>mean = 0.019 ± 0.001 std = 0.119</td>
</tr>
<tr>
<td>$\Delta C_1$</td>
<td>mean = -0.005 ± 0.001 std = 0.163</td>
<td>mean = -0.019 ± 0.001 std = 0.151</td>
</tr>
<tr>
<td>$\Delta C_2$</td>
<td>mean = -0.016 ± 0.001 std = 0.156</td>
<td>mean = -0.014 ± 0.001 std = 0.154</td>
</tr>
</tbody>
</table>

Table 4: The mean and standard deviation of $\Delta \alpha_{\text{CP}}^{\text{max}}$, the difference between true and predicted $\alpha_{\text{CP}}^{\text{max}}$, obtained from DNN with classification and regression methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Classification</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using $w \tau$</td>
<td>mean = 0.032 ± 0.002 [rad] std = 0.481 [rad]</td>
<td>mean = 0.027 ± 0.002 [rad] std = 0.478 [rad]</td>
</tr>
<tr>
<td>Using $C_0, C_1, C_2$</td>
<td>mean = 0.006 ± 0.002 [rad] std = 0.461 [rad]</td>
<td>mean = 0.029 [rad] std = 0.457 [rad]</td>
</tr>
<tr>
<td>Direct</td>
<td>mean = 0.005 [rad] std = 0.447 [rad]</td>
<td>mean = -0.009 [rad] std = 0.685 [rad]</td>
</tr>
</tbody>
</table>

1.0 [rad] (bottom plot). Predictions from the classification and regression methods are compared. Again similar performance of both implementations is observed.

8 Summary

We have performed a proof-of-concept for the DNN methods in the measurement of Higgs boson $H \rightarrow \tau \tau$ CP mixing angle dependent coupling. That extends work of refs. [16, 17] of classification between scalar and pseudoscalar Higgs CP state. Several solutions of classification and of regression types were prepared and numerical results were collected. For the measurement we have studied approaches where; (i) spin weights, (ii) coefficients for the functional form of the spin weight (iii) directly the mixing angle at which the weight has its maximum, were targeted. In all three cases the classification approach seemed to be slightly more performant than regression, but the comparisons relied on the discretised quantities might have been thus somewhat biased toward classification which is of the discreet nature by itself too. Indeed, the regression approach is more natural for continuous observables. The result may be thus a consequence of the testing method, rather than of principle. The minor differences did not point to strong argument at this point which choice is better anyway.

For the feature list we have chosen idealistic case, assuming that complete set of $\tau$ decay products 4-momenta is known, including challenging to reconstruct neutrinos. We have exploited then the $\tau \rightarrow \rho \nu$ decay mode. The results are encouraging, the understanding of environment for future discussion of measurement ambiguities was not compromised with respect to what was achieved in previous publications for scalar/pseudoscalar classifications.

The mean value of the preferred mixing angle $\alpha_{\text{CP}}^{\text{max}}$ can be constrained by the trained DNN with the precision better than 0.05 [rad] and with per-event resolution better than 0.5 [rad]. Both classification and regression approaches allow to learn spin weight with uncertainties (average $l_2$ norm) better than 20%. Both approaches allow also to learn coefficients $C_0, C_1, C_2$ of the functional spin weight form. The coefficients are directly related to the polarimetric vectors of decaying $\tau^\pm$ leptons. This provides interesting possibility for the future studies of experimental ambiguities with samples of the $Z \rightarrow \tau \tau$ decays, much more abundant and available for the LHC measurements. Departure from SM predictions on $Z \tau \tau$ coupling can reveal itself in the observables build from polarimetric vectors of decaying $\tau^\pm$ leptons too.

We plan, following [16,17], to extend our studies to more realistic feature lists and other decay modes. Already
Figure 20: Fraction of event with $|\Delta_{\text{class}}| < 1$ to 4 (top plot) and fraction with $|\Delta \alpha_{\text{max}}^C| < 0.25$ to 1.0 [rad] (bottom plot), as a function of $N_{\text{class}}$. Predictions from DNN with classification and regression methods are collected. Note that step-like structure on the bottom plot, is of the technical origin. It is a consequence of the threshold $|\Delta_{\text{class}}|$ in [rad] crossing the $|\Delta_{\text{class}}|$ boundaries in [idx] with which discretisation was performed initially.
now, the variety of ML methods for the determination of most preferred CP state mixing angle, demonstrated potential and robustness for future experimental analyses and measurements with the LHC data.

Acknowledgments

We would like to thank J. Kurek and P. Winkowska for help with technical implementation and testing of the analysis code used for this paper preparation.

References

where for event \( k \) we want to discretise it. For predicting \( C \) initialized with random weights. Learning procedure is optimized using a variant of stochastic gradient descent loss function for sample of \( N \) neural network predictions [4]. It is a common choice in case of binary or multiclass classification models. The last layer is specific to the implementation case, different in dimension of the output vector, activation function space, the more detailed discussion in case of two hypotheses only, scalar and pseudoscalar, can be found in [16].

The structure of the simulated data and the \( DNN \) architecture follows what was published in our previous papers [16, 17]. It is prepared for TensorFlow [27], an open-source machine learning library.

We consider \( H \rightarrow \tau^+ \tau^- \) channel of both \( \tau^+ \rightarrow \rho^+ \nu \) decay. The data point is thus an event of the Higgs boson production and \( \tau \) lepton pair decay products. The structure of the event is represented as follows:

\[
x_i = (f_{i1}, ..., f_{iD}), w_{a_1}, w_{b_1}, ..., w_{m_i}
\]

(10)

The \( f_{i1}, ..., f_{iD} \) represent numerical features and \( w_{a_1}, w_{b_1}, w_{m_i} \) are weights proportional to the likelihoods that an event comes from a class \( A, B, ..., M \), each representing different \( \alpha^{CP} \) mixing angle. The \( \alpha^{CP} = 0, 2\pi \) corresponds to scalar CP state and \( \alpha^{CP} = \pi \) to pseudoscalar CP state. The weights calculated from the quantum field theory matrix elements are available and stored in the simulated data files. This is a convenient situation, which does not happen in many other cases of ML classification. The \( A, B, ... M \) distributions highly overlap in the \( (f_{i1}, ..., f_{iD}) \) space, the more detailed discussion in case of two hypotheses only, scalar and pseudoscalar, can be found in [16].

Thanks to similar \( DNN \) architecture, we have prepared three implementations for measuring Higgs boson CP state: binary classification, multiclass classification and regression:

- For binary classification the aim is to discriminate between two hypotheses, \( \mathcal{H}_0 \) and \( \mathcal{H}_{\alpha^{CP}} \).

- For multiclass classification, the aim is to simultaneously learn weights (probabilities) for several \( \mathcal{H}_{\alpha^{CP}} \) hypotheses; learn coefficients of the weight functional form or directly learn the mixing angle at which spin weight has its maximum, \( \alpha^{CP}_{\text{max}} \). A single class can be either single discretised \( \alpha^{CP} \) or a range for the \( C_i \) parameters. The system is learning probabilities for classes to associate with the event.

- For regression case, the aim is similar as for multiclass classification case, but now problem is defined as a continuous case. The system is learning value to associate with the event. The value can be a vector of spin weights for a set of \( \mathcal{H}_{\alpha^{CP}} \) hypotheses, set of \( C_i \) coefficients or \( \alpha^{CP}_{\text{max}} \).

The network architecture consists of 6 hidden layers, 300 nodes each with ReLU activation functions and is initialized with random weights. Learning procedure is optimized using a variant of stochastic gradient descent algorithm called Adam [28]. We also use Batch Normalization [29] (that has regularization properties) as well. The last layer is specific to the implementation case, different is dimension of the output vector, activation function and a loss function. In the following, we will describe details.

**Classification:** The loss function used in stochastic gradient descent is a cross entropy of valid values and neural network predictions [4]. It is a common choice in case of binary or multiclass classification models. The loss function for sample of \( N_{\text{evt}} \) events and classification for \( N_{\text{class}} \) reads as follows:

\[
\text{Loss} = \sum_{k=1}^{N_{\text{evt}}} \sum_{i=1}^{N_{\text{class}}} y_{i,k} \log(p_{i,k}),
\]

(11)

where \( k \) stands for consecutive event and \( i \) for class index. The \( y_{i,k} \) represents neural-network predicted probability for event \( k \) being of class \( i \) while \( p_{i,k} \) represents true probability used in supervised training.

**Regression:** In case of predicting \( wt \) the last layer of \( DNN \) is \( N \) dimensional output (granularity with which we want to discretise it). For predicting \( C_0, C_1, C_2 \) the last layer of \( DNN \) is \( N=3 \) dimensional output, i.e. values of
Activation of this layer is a linear function. Loss functions is defined as Mean Squared Error (MSE) between true and predicted parameters

\[ \text{Loss} = \sum_{k=1}^{N_{\text{evt}}} \sum_{i=1}^{N} (y_{i,k} - p_{i,k})^2, \]

where \( k \) stands for event index and \( i \) for index of function form parameter. The \( y_{i,k} \) represents predicted value of \( C_i \) parameter for event \( k \) while \( p_{i,k} \) represents true value. For predicting the \( \alpha_{CP}^{\text{max}} \), the last layer of DNN is \( N=1 \) dimensional output, i.e. values of \( \alpha_{CP}^{\text{max}} \).

The `tf.reduce_mean` method of TensorFlow is used, with the loss function

\[ \text{Loss} = \sum_{k=1}^{N_{\text{evt}}} (1 - \cos(y_k - p_k)), \]

where \( y_k, p_k \) denotes respectively predicted and true value of \( \alpha_{CP}^{\text{max}} \).

In Fig. 21 for all problems considered, distributions of the loss functions on the training and validation samples, as a function of number of epochs used for training are shown. Left plots are for the classification and right plots for the corresponding regression. The values of the loss are case specific and should not be directly compared, their shape is monitoring the training process. For all cases the loss is decreasing with number of epochs, both on training and validation samples. It is overlapping for all cases except [Regression: \( \alpha_{CP}^{\text{max}} \)] (bottom right plot), for that single case one small loss in performance is observed for validation sample compared to training sample. Training with 5-10 epochs seems sufficient for both classification and regression, while learning spin weights or \( C_i \) coefficients, top and middle plots. For learning directly \( \alpha_{CP}^{\text{max}} \) with regression method, training with at least 25 epochs is closer to optimal.
Figure 21: The DNN loss for classification (left-side) and regression (right-side), as function of number of epochs used for training. It is shown for learning spin weight (top plots), \( C_i \) coefficients (middle plots) and most likely mixing angle \( \alpha_{CP}^{\text{max}} \) (bottom plots). For the classification, \( N_{\text{class}} = 21 \) was used.