

Description of supporting information for “A machine learning approach to classifying MESSENGER FIPS proton spectra”

Matthew K. James¹, Suzanne M. Imber^{1,2}, Jim M. Raines², Tim K. Yeoman¹, and Emma J. Bunce¹

¹Department of Physics and Astronomy, University of Leicester, Leicester LE1 7RH, UK

²Department of Atmospheric, Oceanic and Space Sciences, University of Michigan, Ann Arbor, Michigan, USA

January 13, 2020

1 Introduction

This document contains information about data files associated with the paper titled “A machine learning approach to classifying MESSENGER FIPS proton spectra” in the Journal of Geophysical Research: Space Physics (*James et al.*, under review). Section 2 described the order of each parameter and their associated scaling factors used to create the input feature matrices for artificial neural networks (ANNs) 0–7. Section 3 describes the parameter order and scaling factors used in creating the input feature matrices for ANN 8. Section 4 describes the data files in this repository which contain the weight and bias matrices learned for each ANN. Finally, section 5 contains a description of the data file which contains the new fitted κ -distribution parameters and the outputs for all 9 ANNs.

2 Feature matrix parameter order and scaling for NN 0–7

This section describes the order in which the neural network parameters listed in Table A.1 of the paper are organized in the input feature matrices for neural networks 0–7 in Figure 2. The shape of the feature matrices for each of these eight networks is $X \in \mathbb{R}^{(m \times 31)}$, where m is the number of samples and there are 31 features in each sample. Table 1 lists the parameters used to provide the feature matrix. The “Features” column shows the number of features provided by each parameter, and the “Indices” column shows which index (starting at 0) these features correspond to in the second dimension of the feature matrix. The columns λ , δ , μ and σ correspond to the Box-Cox transformation parameters used to rescale each input feature and are explained in Appendix A. NOTE: with the exception of n_κ , T_κ and κ which describe the whole spectrum, each of these parameters is calculated using an eighth of the spectrum - each neural network utilizes data from 8 of the 64 (E/q) bins, e.g. NN 0 uses data from bins 0–7, NN 1 uses bins 8–15, etc.

3 Feature matrix parameter order and scaling for NN 8

This section lists the input parameters used to create the feature matrix for the final neural network (NN 8 in Figure 2). The shape of the feature matrix for this network is $X \in \mathbb{R}^{(m \times 239)}$, where m is the number of samples and there are 239 features in each sample. Table 2 lists the parameters used to provide the feature matrix. The “Features” column shows the

Parameter	Features	Indices	λ	δ	μ	σ
C_s	8	0–7	0.01	0.0714	1.2	1.6
C_V	1	8	0.417	0.0	601.0	172.0
Δf	8	9–16	-1.09	3.838	0.678	0.049
n_κ	1	17	-0.1133	0.0	0.8	1.37
T_κ	1	18	0.1514	0.0	2.52	1.66
κ	1	19	-0.0176	0.0	1.32	0.5
$\frac{\sqrt{C_i}}{C_i}$	8	20–27	0.0	1.0	1.0	1.0
D_{split}	1	28	0.0	1.0	1.0	1.0
Δf	1	29	-1.2	3.4	0.63	0.06
G	1	30	0.0	1.0	1.0	1.0

Table 1: Table of parameters which form the feature matrices of the first eight neural networks (NN 0–7 in Figure 2).

Parameter	Features	Indices	λ	δ	μ	σ
C_s	64	0-63	0.01	0.0714	1.2	1.6
C_V	1	64	0.417	0.0	601.0	172.0
Δf	64	65-128	-1.09	3.838	0.678	0.049
n_κ	1	129	-0.1133	0.0	0.8	1.37
T_κ	1	130	0.1514	0.0	2.52	1.66
κ	1	131	-0.0176	0.0	1.32	0.5
Δf_{RMS}	1	132	-0.6719	0.0	-1.32	0.92
χ^2	1	133	0.12	0.0000557	3.46	2.94
$\frac{\sqrt{C_i}}{C_i}$	64	134-197	0.0	1.0	1.0	1.0
D	1	198	0.0	1.0	1.0	1.0
D_{split}	8	199-206	0.0	1.0	1.0	1.0
Δf	8	207-214	-1.2	3.4	0.63	0.06
G	8	215-222	0.0	1.0	1.0	1.0
y_p	8	223-230	0.0	1.0	1.0	1.0
y_c	8	231-238	0.0	1.0	1.0	1.0

Table 2: Table of parameters which form the feature matrix of the final neural network (NN 8 in Figure 2).

Layer	NN 0-7		NN 8	
	w	b	w	b
$s_0 \rightarrow s_1$	(31,39)	(1,39)	(239,68)	(1,68)
$s_1 \rightarrow s_2$	(39,13)	(1,13)	(68,23)	(1,23)
$s_2 \rightarrow s_3$	(13,2)	(1,2)	(23,2)	(1,2)

Table 3: Table showing the shapes of the weight and bias matrices that map between neural network layers.

number of features provided by each parameter, and the “Indices” column shows which index (starting at 0) these features correspond to in the second dimension of the feature matrix. The columns λ , δ , μ and σ correspond to the Box-Cox transformation parameters used to rescale each input feature and are explained in Appendix A.

4 Weight and bias matrices

In this data repository there are 9 files which are binary files containing the exact weight and bias matrices for each of the neural networks trained in this paper; another 9 files contain the weight and bias matrices stored as ASCII if preferred; and another file contains Python 3 code which is capable of reading both file formats.

The binary files are named “Matrix-ANN-X.bin”, where X is an integer in the range 0-8 and corresponds to the 9 neural networks trained for this study. The matrices are stored in the order which they appear in the neural networks, starting with $w^{(0)}$, $b^{(0)}$, $w^{(1)}$, $b^{(1)}$, etc., alternating between weight and bias matrices between each layer. Each matrix is stored contiguously as $s_l \times s_{l+1}$ (or $1 \times s_{l+1}$ for the biases) 32-bit floating points (little-endian). After reading each contiguous block, the values can be reshaped into a 2D array using row-major ordering (row-major ordering is default for C, C++ and Python, but **not** FORTRAN). The shapes of the matrices are listed in table 3.

The same matrices are also stored in ASCII files names “Matrix-ANN-X.dat”, where X corresponds to the neural network number, as with the binary files. The matrices stored in each file are in the same order as those in the binary files, and are separated by a line containing a description of the matrix type and shape, e.g. “WeightMatrix0:(31,39)” or “BiasMatrix1:(1,13)”. Each matrix is stored as s_l lines (or just one line for the bias matrices), each containing s_{l+1} floating points formatted using scientific notation.

The file named “ReadMatrices.py” contains Python 3 code which is capable of reading both file types. In Python, the weights and biases of network I can be read using either

```
w, b = ReadBinaryMatrices(I, MatrixPath)
or
w, b = ReadASCIIMatrices(I, MatrixPath)
```

where `MatrixPath` is a string containing the full path to the directory where the matrix files are stored. `w` and `b` are Python list objects, which contain 2D numpy arrays of weights and biases, respectively.

5 κ distribution fits and ANN outputs

This dataset contains the three parameters fitted to each proton spectrum (density, temperature and κ). Each spectral fit is analysed using 9 artificial neural networks (ANNs), where ANNs 0-7 each analyse an eight of a spectrum and the

final network (ANN 8) provides an overall assessment of the quality of fit. Each network outputs a probability of the fit being good and a corresponding class label. A spectral fit is determined to be “good” when the probability ≥ 0.5 . The accuracy of the final neural network is 96%, so in some cases a few spectra may be misclassified - manual verification is recommended when using these data. Please read the aforementioned paper for more information.

The file “FIPSProtonClass.dat” is organised into 22 columns, each of which is described in table 4.

Column	Description
1	Date and time of spectrum formatted as yyyy-mm-ddTHH:MM:SS.sss, where yyyy is the year, mm is the month, dd is the day, HH is the hour, MM is minutes and SS.sss is seconds.
2	Proton density, n_k , in cm^{-3} .
3	Proton temperature, T_k , in MK.
4	Kappa value - note that any value over ~ 10 may be approximated using a Maxwellian distribution, and any value less than 1.5 is below the limit of anti-equilibrium and is not physical.
5–12	Probabilities, P0-P7, that the kappa distribution fit to each spectrum is good (output from ANN 0-7 in the paper).
13	Probability that the overall fit is good (output from ANN 8 in the paper).
14–21	Class of each split section, where 0 is bad and 1 is good. The class is 1 when $P \geq 0.5$.
22	Overall spectral class predicted by ANN 8, where 1 is good and 0 is bad.

Table 4: Table showing the shapes of the weight and bias matrices that map between neural network layers.

References

James, M. K., S. M. Imber, J. M. Raines, T. K. Yeoman, and E. J. Bunce (under review), A machine learning approach to classifying MESSENGER FIPS proton spectra, *J. Geophys. Res. Sp. Phys.*