Running times of DLRawTriggerReader

All the times are measured for the triggerless (>20 Cherenkov p.e.) file of gammas: "/gamma3/users/jbuces/software/mc_data/trigerless_gammas/reduced_ctapipe/corsika_run1.dl1b.h5" (11044 events, only 1 tel). When the output option implies more than a row (multiple patches) for each event, the example identifiers file is modified, i.e. not done in the append features function. The time for batch generation is for multiples of 49 (number of patches) in the output.

RawTriggerReader inherits all the settings from the WaveformReader, with some extra settings as:
- output_settings, to select which type of output we want in the batch generator: `["waveform", "random_patch", "balanced_patches", "hot_patch", "all_patches", "double_random"]`

-trigger_settings, dictionary to set the number of trigger patches (7 if we want 7x7), the cpe_threshold: threshold in Cherenkov p.e. above which patches are labeled cosmic, and where other trigger settings as the characteristics of trigger patches are saved.

-”hot_pixel_from_simulation”

The principal output settings are:

- “balanced_patches”: in order to get the maximum amount of data to train. Introduces the same and maximum amount of nsb and cosmic patches per event. If one event has no nsb or cosmic patches the value of true Cherenkov and patch index in the table is -1, appends an all -1 waveform.

- “all_patches”: generate in the ex identifiers file number of trigger patches **2 of new lines per event with the patch index in each one. Maybe useful for inference.

- If balanced options are too long for initialising (7 seconds for 11000 events) and need an equilibrated dataset, there is the “double_random” option where we create 2 extra lines for each event, one for each patch type, cosmic and nsb, and append a random of the corresponding type for each.

output_setting	identifiers_length	init_time	batch_4	batch_5	batch_6
		s	s	s	s
str16	int64	float64	float64	float64	float64
waveform	11044	1.142	0.639	0.799	0.987
random_patch	11044	1.216	0.734	0.93	1.139
balanced_patches	266209	7.176	0.303	0.381	0.461
hot_patch	11044	1.355	0.778	0.976	1.172
all_patches	541156	1.393	0.306	0.384	0.464
double_random	22088	1.39	0.577	0.729	0.856

Initialisation time for **waveform** is:
1.13 s ± 1.29 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for waveform, for a 4 * 49 batch size is:
638 ms ± 4.9 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for waveform, for a 5 * 49 batch size is:
797 ms ± 4.02 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for waveform, for a 6 * 49 batch size is:
982 ms ± 12.5 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Initialisation time for **random_patch** is:
1.21 s ± 6.41 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for random_patch, for a 4 * 49 batch size is:
727 ms ± 3.22 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for random_patch, for a 5 * 49 batch size is:
921 ms ± 2.73 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for random_patch, for a 6 * 49 batch size is:
1.13 s ± 10.3 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Initialisation time for **balanced_patches** is:
7.19 s ± 70.6 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for balanced_patches, for a 4 * 49 batch size is:
304 ms ± 1.18 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for balanced_patches, for a 5 * 49 batch size is:
389 ms ± 642 µs per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for balanced_patches, for a 6 * 49 batch size is:
468 ms ± 1.67 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Initialisation time for **hot_patch** is:
1.35 s ± 10.3 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for hot_patch, for a 4 * 49 batch size is:
771 ms ± 2.48 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for hot_patch, for a 5 * 49 batch size is:
974 ms ± 6.13 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

Batch generation time for hot_patch, for a 6 * 49 batch size is:
1.17 s ± 7.13 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Initialisation time for **all_patches** is:
1.38 s ± 4.32 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for all_patches, for a 4 * 49 batch size is:
311 ms ± 1.11 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for all_patches, for a 5 * 49 batch size is:
391 ms ± 808 µs per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for all_patches, for a 6 * 49 batch size is:
473 ms ± 893 µs per loop (mean ± std. dev. of 7 runs, 1 loop each)
Initialisation time for **double_random** is:
1.36 s ± 9.16 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for double_random, for a 4 * 49 batch size is:
565 ms ± 4.33 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for double_random, for a 5 * 49 batch size is:
719 ms ± 7.78 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Batch generation time for double_random, for a 6 * 49 batch size is:
872 ms ± 4.53 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

Generated batch for each “output_settings”:

- ”waveform” (default):

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	waveform	cherenkov_pe	patch_class	true_energy	true_alt
										TeV	deg
int64	int64	int64	int32	int64	int64	int64	float32[117,117,6]	int64	int64	float64	float64
0	0	0	1	0	502	1	0.0 .. 0.0	74	0	0.010171852074563503	70.8581309781398
1	0	1	1	0	1903	1	0.0 .. 0.0	53	0	0.0323750302195549	69.25027029657295
2	0	2	1	0	3101	1	0.0 .. 0.0	43	0	0.3737192451953888	67.79137604080408
3	0	3	1	0	3305	1	0.0 .. 0.0	44	0	0.015845729038119316	69.0996782857525
4	0	4	1	0	3307	1	0.0 .. 0.0	37	0	0.015845729038119316	69.0996782857525

- “random_patch”:

When the random variable hits True takes a random nsb patch, when it hits False takes the nearest patch to the hot pixel (most Cherenkov p.e. if hot_pixel_from_simulation = True, highest integrated charge if hot_pixel_from_simulation = False)

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	patch_index	waveform	cherenkov_pe	patch_class	true_energy	true_alt
											TeV	deg
int64	int64	int64	int32	int64	int64	int64	int64	float32[28,28,75]	int64	int64	float64	float64
0	0	0	1	0	502	1	31	294.0 .. 296.0	74	0	0.010171852074563503	70.8581309781398
1	0	1	1	0	1903	1	12	309.0 .. 296.0	0	1	0.0323750302195549	69.25027029657295
2	0	2	1	0	3101	1	4	0.0 .. 293.0	41	0	0.3737192451953888	67.79137604080408
3	0	3	1	0	3305	1	40	291.0 .. 0.0	0	1	0.015845729038119316	69.0996782857525
4	0	4	1	0	3307	1	4	0.0 .. 293.0	0	1	0.015845729038119316	69.0996782857525
5	0	5	1	0	3700	1	30	294.0 .. 289.0	145	0	0.1309148669242859	69.31930301851428

- “hot_patch”:

Gives the patch which center is the nearest to the hot pixel. Same “hot_pixel_from_simulation” option as “random_patch”.

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	patch_index	waveform	cherenkov_pe	patch_class	true_energy	true_alt
											TeV	deg
int64	int64	int64	int32	int64	int64	int64	int64	float32[28,28,75]	int64	int64	float64	float64
0	0	0	1	0	502	1	31	294.0 .. 296.0	74	0	0.010171852074563503	70.8581309781398
1	0	1	1	0	1903	1	10	295.0 .. 295.0	49	0	0.0323750302195549	69.25027029657295
2	0	2	1	0	3101	1	4	0.0 .. 293.0	41	0	0.3737192451953888	67.79137604080408
3	0	3	1	0	3305	1	12	295.0 .. 305.0	44	0	0.015845729038119316	69.0996782857525
4	0	4	1	0	3307	1	13	0.0 .. 0.0	33	0	0.015845729038119316	69.0996782857525

- “balanced_patches”

Gives min(nsb_patches, cosmic_patches) number of random patches of each class per event. Randomise the selection of patches to avoid selecting always the same patches for nsb, e.g. mostly showers with 10 cosmic patches, so it doesn’t always take the first 10 nsb patches.

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	patch_index	waveform	cherenkov_pe	patch_class	true_energy	true_alt
											TeV	deg
int64	int64	int64	int32	int64	int64	int64	int64	float32[28,28,10]	int64	int64	float64	float64
0	0	0	1	0	502	1	33	290.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
1	0	0	1	0	502	1	25	307.0 .. 307.0	16	0	0.010171852074563503	70.8581309781398
2	0	0	1	0	502	1	5	0.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
3	0	0	1	0	502	1	23	303.0 .. 302.0	1	0	0.010171852074563503	70.8581309781398
4	0	0	1	0	502	1	37	301.0 .. 314.0	1	0	0.010171852074563503	70.8581309781398
5	0	0	1	0	502	1	24	295.0 .. 297.0	18	0	0.010171852074563503	70.8581309781398
6	0	0	1	0	502	1	38	295.0 .. 298.0	51	0	0.010171852074563503	70.8581309781398
7	0	0	1	0	502	1	30	298.0 .. 293.0	2	0	0.010171852074563503	70.8581309781398

- “double_random”:

For each event we generate 2 rows in the “example_identifiers”: 1 for nsb (“patch_class” = 1) another for cosmic (“patch_class” = 0). In the “_append_features” select a random nsb and cosmic patch.

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	patch_index	waveform	cherenkov_pe	patch_class	true_energy	true_alt
											TeV	deg
int64	int64	int64	int32	int64	int64	int64	int64	float32[28,28,10]	int64	int64	float64	float64
0	0	0	1	0	502	1	24	295.0 .. 297.0	18	0	0.010171852074563503	70.8581309781398
1	0	0	1	0	502	1	34	297.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
2	0	1	1	0	1903	1	0	0.0 .. 298.0	2	0	0.0323750302195549	69.25027029657295
3	0	1	1	0	1903	1	44	296.0 .. 0.0	0	1	0.0323750302195549	69.25027029657295
4	0	2	1	0	3101	1	6	0.0 .. 0.0	2	0	0.3737192451953888	67.79137604080408
5	0	2	1	0	3101	1	8	0.0 .. 298.0	0	1	0.3737192451953888	67.79137604080408
6	0	3	1	0	3305	1	16	300.0 .. 309.0	0	1	0.015845729038119316	69.0996782857525
7	0	3	1	0	3305	1	12	299.0 .. 311.0	44	0	0.015845729038119316	69.0996782857525

- “all_patches”:

For each event we generate (number_of_trigger_patches)**2 rows in the example identifiers. Then append for each “patch_index” its corresponding “waveform”, “patch_class”, and number of “cherenkov_pe”

index	file_index	table_index	obs_id	tel_type_id	event_id	tel_id	patch_index	waveform	cherenkov_pe	patch_class	true_energy	true_alt
											TeV	deg
int64	int64	int64	int32	int64	int64	int64	int64	float32[28,28,10]	int64	int64	float64	float64
0	0	0	1	0	502	1	0	0.0 .. 295.0	0	1	0.010171852074563503	70.8581309781398
1	0	0	1	0	502	1	27	287.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
2	0	0	1	0	502	1	28	0.0 .. 291.0	0	1	0.010171852074563503	70.8581309781398
3	0	0	1	0	502	1	29	301.0 .. 299.0	0	1	0.010171852074563503	70.8581309781398
4	0	0	1	0	502	1	30	298.0 .. 293.0	2	0	0.010171852074563503	70.8581309781398
5	0	0	1	0	502	1	31	294.0 .. 298.0	74	0	0.010171852074563503	70.8581309781398
6	0	0	1	0	502	1	32	301.0 .. 302.0	70	0	0.010171852074563503	70.8581309781398
7	0	0	1	0	502	1	33	290.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
8	0	0	1	0	502	1	34	297.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
…	…	…	…	…	…	…	…	…	…	…	…	…
41	0	0	1	0	502	1	22	299.0 .. 296.0	0	1	0.010171852074563503	70.8581309781398
42	0	0	1	0	502	1	16	312.0 .. 306.0	0	1	0.010171852074563503	70.8581309781398
43	0	0	1	0	502	1	17	302.0 .. 296.0	0	1	0.010171852074563503	70.8581309781398
44	0	0	1	0	502	1	18	292.0 .. 306.0	0	1	0.010171852074563503	70.8581309781398
45	0	0	1	0	502	1	19	293.0 .. 302.0	0	1	0.010171852074563503	70.8581309781398
46	0	0	1	0	502	1	20	314.0 .. 0.0	0	1	0.010171852074563503	70.8581309781398
47	0	0	1	0	502	1	21	0.0 .. 300.0	0	1	0.010171852074563503	70.8581309781398
48	0	0	1	0	502	1	12	292.0 .. 307.0	0	1	0.010171852074563503	70.8581309781398
49	0	1	1	0	1903	1	15	0.0 .. 302.0	2	0	0.0323750302195549	69.25027029657295