Dataset_creation

December 16, 2023

[1]:
```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score
import xarray as xr
```

[2]:
```
import os
import xarray as xr
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import numpy as np

# Load the xarray dataset (replace 'your_data.nc' with the actual file name)
ds = xr.open_dataset('new.nc')

# Create the Output folder if it doesn't exist
output_folder = 'Output'
os.makedirs(output_folder, exist_ok=True)

# Split the dataset into three parts along the time dimension
num_chunks = 9
chunk_size = len(ds['time']) // num_chunks

# Iterate through each chunk
for chunk_index in range(num_chunks):
    start_index = chunk_index * chunk_size
    end_index = start_index + chunk_size

    # Extract the chunk along the time dimension
    ds_chunk = ds.isel(time=slice(start_index, end_index))

    # Iterate through each time step in the current chunk
    for time_index, time_value in enumerate(ds_chunk['time']):
        # Create a figure with GeoAxes
```

```
        fig = plt.figure(figsize=(16, 8))
        projection = ccrs.PlateCarree()
        ax = plt.subplot(projection=projection)
        ax.coastlines()

        # Iterate through each variable and plot on the GeoAxes
        for var_name in ['z', 'u', 'r', 'pv', 'v', 'cc', 'q']:
            ds_chunk[var_name].sel(time=time_value).plot(ax=ax, transform=ccrs.
↪PlateCarree(), cmap='coolwarm', add_colorbar=False)

        # Customize the title based on the target variable ('t' in this case)
        target_values = ds_chunk['t'].sel(time=time_value).values
        target_value = float(np.mean(target_values))  # Extract the mean value

        # Save the figure with the target value as part of the filename
        output_file = os.path.join(output_folder,␣
↪f'earth_image_chunk{chunk_index}_time{time_index}_target_{target_value:.0f}.
↪png')
        plt.savefig(output_file, bbox_inches='tight')

        # Close the figure to free up resources
        plt.close()

        # Display a message indicating that the figure was saved
        print(f"Figure saved to: {output_file}")
```

```
Unexpected exception formatting exception. Falling back to standard exception

Traceback (most recent call last):
  File "C:\Users\nehag\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py", line 3460, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "C:\Users\nehag\AppData\Local\Temp\ipykernel_2188\2448340106.py", line
44, in <module>
    plt.savefig(output_file, bbox_inches='tight')
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\pyplot.py", line
996, in savefig
    res = fig.savefig(*args, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\figure.py", line
3328, in savefig
    self.canvas.print_figure(fname, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\backend_bases.py",
line 2338, in print_figure
    self.figure.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
95, in draw_wrapper
    result = draw(artist, renderer, *args, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
```

```
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\figure.py", line
3125, in draw
    mimage._draw_list_compositing_images(
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\image.py", line
131, in _draw_list_compositing_images
    a.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\cartopy\mpl\geoaxes.py", line
535, in draw
    return super().draw(renderer=renderer, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\axes\_base.py",
line 3066, in draw
    mimage._draw_list_compositing_images(
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\image.py", line
131, in _draw_list_compositing_images
    a.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\collections.py",
line 2078, in draw
    coordinates = transform.transform(coordinates)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\transforms.py",
line 1490, in transform
    res = self.transform_affine(self.transform_non_affine(values))
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\transforms.py",
line 2422, in transform_non_affine
    return self._a.transform_non_affine(points)
  File "C:\Users\nehag\anaconda3\lib\site-packages\cartopy\mpl\geoaxes.py", line
130, in transform_non_affine
    return prj.transform_points(self.source_projection,
  File "C:\Users\nehag\anaconda3\lib\site-packages\cartopy\crs.py", line 417, in
transform_points
    _safe_pj_transform(src_crs, self, x, y, z, trap=trap)
  File "C:\Users\nehag\anaconda3\lib\site-packages\cartopy\crs.py", line 65, in
_safe_pj_transform
    return transformer.transform(x, y, z, errcheck=trap)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pyproj\transformer.py", line
833, in transform
    iny, y_data_type = _copytobuffer(yy, inplace=inplace)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pyproj\utils.py", line 129,
```

```
in _copytobuffer
    return xxx.astype("d", order="C", copy=not inplace), DataType.ARRAY
numpy.core._exceptions._ArrayMemoryError: Unable to allocate 7.94 MiB for an
array with shape (1040402,) and data type float64

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\nehag\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py", line 2057, in showtraceback
    stb = self.InteractiveTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 1118, in structured_traceback
    return FormattedTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 1012, in structured_traceback
    return VerboseTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 865, in structured_traceback
    formatted_exception = self.format_exception_as_a_whole(etype, evalue, etb,
number_of_lines_of_context,
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 818, in format_exception_as_a_whole
    frames.append(self.format_record(r))
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 736, in format_record
    result += ''.join(_format_traceback_lines(frame_info.lines, Colors,
self.has_colors, lvals))
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 155, in _format_traceback_lines
    line = stack_line.render(pygmented=has_colors).rstrip('\n') + '\n'
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\core.py", line
360, in render
    start_line, lines = self.frame_info._pygmented_scope_lines
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
145, in cached_property_wrapper
    value = obj.__dict__[self.func.__name__] = self.func(obj)
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\core.py", line
780, in _pygmented_scope_lines
    lines = _pygmented_with_ranges(formatter, code, ranges)
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
165, in _pygmented_with_ranges
    return pygments.highlight(code, lexer, formatter).splitlines()
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\__init__.py", line
82, in highlight
    return format(lex(code, lexer), formatter, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\__init__.py", line
61, in format
```

```
    formatter.format(tokens, realoutfile)
  File "C:\Users\nehag\anaconda3\lib\site-
packages\pygments\formatters\terminal256.py", line 250, in format
    return Formatter.format(self, tokensource, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\formatter.py", line
94, in format
    return self.format_unencoded(tokensource, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-
packages\pygments\formatters\terminal256.py", line 256, in format_unencoded
    for ttype, value in tokensource:
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
158, in get_tokens
    for ttype, value in super().get_tokens(text):
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\lexer.py", line 187,
in streamer
    for _, t, v in self.get_tokens_unprocessed(text):
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\lexer.py", line 629,
in get_tokens_unprocessed
    m = rexmatch(text, pos)
MemoryError

Unexpected exception formatting exception. Falling back to standard exception

Traceback (most recent call last):
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\formatters.py",
line 221, in catch_format_error
    r = method(self, *args, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\formatters.py",
line 338, in __call__
    return printer(obj)
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\pylabtools.py",
line 152, in print_figure
    fig.canvas.print_figure(bytes_io, **kw)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\backend_bases.py",
line 2338, in print_figure
    self.figure.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
95, in draw_wrapper
    result = draw(artist, renderer, *args, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\figure.py", line
3125, in draw
    mimage._draw_list_compositing_images(
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\image.py", line
131, in _draw_list_compositing_images
    a.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
```

```
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\cartopy\mpl\geoaxes.py", line
535, in draw
    return super().draw(renderer=renderer, **kwargs)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\axes\_base.py",
line 3066, in draw
    mimage._draw_list_compositing_images(
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\image.py", line
131, in _draw_list_compositing_images
    a.draw(renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\artist.py", line
72, in draw_wrapper
    return draw(artist, renderer)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\collections.py",
line 2074, in draw
    self.update_scalarmappable()
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\collections.py",
line 887, in update_scalarmappable
    self._mapped_colors = self.to_rgba(self._A, self._alpha)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\cm.py", line 493,
in to_rgba
    x = self.norm(x)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\colors.py", line
1336, in __call__
    result, is_scalar = self.process_value(value)
  File "C:\Users\nehag\anaconda3\lib\site-packages\matplotlib\colors.py", line
1312, in process_value
    result = np.ma.array(data, mask=mask, dtype=dtype, copy=True)
  File "C:\Users\nehag\anaconda3\lib\site-packages\numpy\ma\core.py", line 6610,
in array
    return MaskedArray(data, mask=mask, dtype=dtype, copy=copy,
  File "C:\Users\nehag\anaconda3\lib\site-packages\numpy\ma\core.py", line 2826,
in __new__
    _data = np.array(data, dtype=dtype, copy=copy,
numpy.core._exceptions._ArrayMemoryError: Unable to allocate 3.96 MiB for an
array with shape (721, 1440) and data type float32

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "C:\Users\nehag\anaconda3\lib\site-
packages\IPython\core\interactiveshell.py", line 2057, in showtraceback
    stb = self.InteractiveTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
```

```
line 1118, in structured_traceback
    return FormattedTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 1012, in structured_traceback
    return VerboseTB.structured_traceback(
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 865, in structured_traceback
    formatted_exception = self.format_exception_as_a_whole(etype, evalue, etb,
number_of_lines_of_context,
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 818, in format_exception_as_a_whole
    frames.append(self.format_record(r))
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 736, in format_record
    result += ''.join(_format_traceback_lines(frame_info.lines, Colors,
self.has_colors, lvals))
  File "C:\Users\nehag\anaconda3\lib\site-packages\IPython\core\ultratb.py",
line 155, in _format_traceback_lines
    line = stack_line.render(pygmented=has_colors).rstrip('\n') + '\n'
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\core.py", line
360, in render
    start_line, lines = self.frame_info._pygmented_scope_lines
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
145, in cached_property_wrapper
    value = obj.__dict__[self.func.__name__] = self.func(obj)
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\core.py", line
780, in _pygmented_scope_lines
    lines = _pygmented_with_ranges(formatter, code, ranges)
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
165, in _pygmented_with_ranges
    return pygments.highlight(code, lexer, formatter).splitlines()
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\__init__.py", line
82, in highlight
    return format(lex(code, lexer), formatter, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\__init__.py", line
61, in format
    formatter.format(tokens, realoutfile)
  File "C:\Users\nehag\anaconda3\lib\site-
packages\pygments\formatters\terminal256.py", line 250, in format
    return Formatter.format(self, tokensource, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\formatter.py", line
94, in format
    return self.format_unencoded(tokensource, outfile)
  File "C:\Users\nehag\anaconda3\lib\site-
packages\pygments\formatters\terminal256.py", line 256, in format_unencoded
    for ttype, value in tokensource:
  File "C:\Users\nehag\anaconda3\lib\site-packages\stack_data\utils.py", line
158, in get_tokens
```

```
    for ttype, value in super().get_tokens(text):
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\lexer.py", line 187,
in streamer
    for _, t, v in self.get_tokens_unprocessed(text):
  File "C:\Users\nehag\anaconda3\lib\site-packages\pygments\lexer.py", line 629,
in get_tokens_unprocessed
    m = rexmatch(text, pos)
MemoryError
```

```
<Figure size 1600x800 with 1 Axes>
```

[ ]: