

Feedback and error collection form

Machine:

MacBook Pro
MacOS Sequoia 15.2
Chip: Apple M2
Memory 16GB

Duration to go through the tutorial: 21hrs (I read everything and did all the exercises)

Feedback:

Overall, great! A very valuable resource.

Here are some errors I ran into:

In Setup <https://carpentries-incubator.github.io/deep-learning-intro/index.html#software-setup>

I would add a statement that the jupyter lab needs to be run in the activated venv?

“Machine learning: algorithms that learn ... without human programming”: I think I would rephrase that.

In <https://carpentries-incubator.github.io/deep-learning-intro/1-introduction.html#neural-networks>

Introduction of the neuron: It might be useful to actually show the sketch of a neuron here – not everyone might know what a neuron looks like.

In: <https://carpentries-incubator.github.io/deep-learning-intro/1-introduction.html#what-makes-deep-learning-deep-learning>

Neural networks aren't... would stick with the standard: *are not*

In <https://carpentries-incubator.github.io/deep-learning-intro/1-introduction.html#compute-the-mean-squared-error>

Compute the Mean Squared Error: the hat is over the bracket instead of the y, in the MSE equation

In: <https://carpentries-incubator.github.io/deep-learning-intro/2-keras.html#split-data-into-training-and-test-set> – should render into a list, but does not. Instead is displayed the following:

This function takes a number of parameters which are extensively explained in [the scikit-learn documentation](#) : - The first two parameters are the dataset (in our case features) and the corresponding targets (i.e. defined as target). - Next is the named parameter `test_size` this is the fraction of the dataset that is used for testing, in this case `0.2` means 20% of the data will be used for testing. - `random_state` controls the shuffling of the dataset, setting this value will reproduce the same results (assuming you give the same integer) every time it is called. - `shuffle` which can be either `True` or `False`, it controls whether the order of the rows of the dataset is shuffled before splitting. It defaults to `True`. - `stratify` is a more advanced parameter that controls how the split is done. By setting it to `target` the train and test sets the function will return will have roughly the same proportions (with regards to the number of penguins of a certain species) as the dataset.

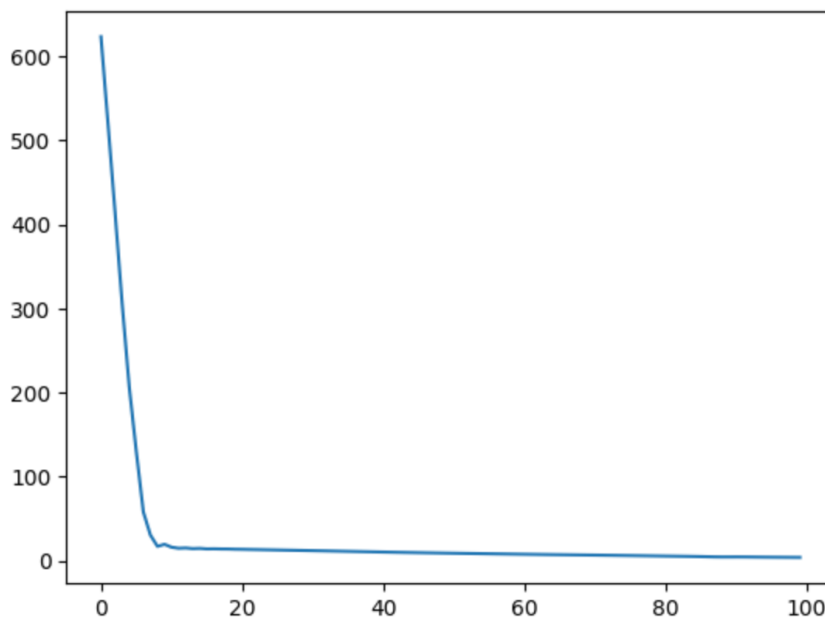
In:

<https://carpentries-incubator.github.io/deep-learning-intro/2-keras.html#build-a-neural-network-from-scratch>

.. “which is surprisingly straightforward” would replace this in line with the Carpentries Ethos not to use language that such as “easy” etc.

The plot I get does not match what is the instructions

: <Axes: >



In:

<https://carpentries-incubator.github.io/deep-learning-intro/2-keras.html#share-model>:
It seems the `.keras` extension is missing in the `model.save('my_first_model')` command.

```

-----
ValueError                                Traceback (most recent call last)
Cell In[30], line 1
----> 1 model.save('my_first_model')

File ~/Documents/dl_workshop/lib/python3.11/site-packages/keras/src/utils/trace
back_utils.py:122, in filter_traceback.<locals>.error_handler(*args, **kwargs)
    119     filtered_tb = _process_traceback_frames(e.__traceback__)
    120     # To get the full stack trace, call:
    121     # `keras.config.disable_traceback_filtering()`
--> 122     raise e.with_traceback(filtered_tb) from None
    123 finally:
    124     del filtered_tb

File ~/Documents/dl_workshop/lib/python3.11/site-packages/keras/src/saving/savi
ng_api.py:114, in save_model(model, filepath, overwrite, zipped, **kwargs)
    110 if str(filepath).endswith((".h5", ".hdf5")):
    111     return legacy_h5_format.save_model_to_hdf5(
    112         model, filepath, overwrite, include_optimizer
    113     )
--> 114 raise ValueError(
    115     "Invalid filepath extension for saving. "
    116     "Please add either a `.keras` extension for the native Keras "
    117     f"format (recommended) or a `.h5` extension. "
    118     "Use `model.export(filepath)` if you want to export a SavedModel "
    119     "for use with TFLite/TF Serving/etc. "
    120     f"Received: filepath={filepath}."
    121 )

ValueError: Invalid filepath extension for saving. Please add either a `.keras`
extension for the native Keras format (recommended) or a `.h5` extension. Use `
model.export(filepath)` if you want to export a SavedModel for use with TFLite/
TF Serving/etc. Received: filepath=my_first_model.

```

Similarly, the .keras extension needs to be added in the load command:
pretrained_model = keras.models.load_model('my_first_model.keras')

I could not run the second example. I get a ssl certification error:

```
data = pd.read_csv("https://zenodo.org/record/5071376/files/weather_prediction_d
```

```
-----  
SSLCertVerificationError                                Traceback (most recent call last)  
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/urllib/r  
equest.py:1348, in AbstractHTTPHandler.do_open(self, http_class, req, **http_co  
nn_args)  
    1347 try:  
-> 1348     h.request(req.get_method(), req.selector, req.data, headers,  
    1349                 encode_chunked=req.has_header('Transfer-encoding'))  
    1350 except OSError as err: # timeout error
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/http/cli  
ent.py:1286, in HTTPConnection.request(self, method, url, body, headers, encode  
_chunked)  
    1285 """Send a complete request to the server."""  
-> 1286 self._send_request(method, url, body, headers, encode_chunked)
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/http/cli  
ent.py:1332, in HTTPConnection._send_request(self, method, url, body, headers,  
encode_chunked)  
    1331     body = _encode(body, 'body')  
-> 1332 self.endheaders(body, encode_chunked=encode_chunked)
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/http/cli  
ent.py:1281, in HTTPConnection.endheaders(self, message_body, encode_chunked)  
    1280     raise CannotSendHeader()  
-> 1281 self._send_output(message_body, encode_chunked=encode_chunked)
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/http/cli  
ent.py:1041, in HTTPConnection._send_output(self, message_body, encode_chunked)  
    1040 del self._buffer[:]  
-> 1041 self.send(msg)
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/urllib/r  
equest.py:1391, in HTTPSHandler.https_open(self, req)  
    1390 def https_open(self, req):  
-> 1391     return self.do_open(http.client.HTTPSConnection, req,  
    1392                        context=self._context, check_hostname=self._check_hostname)
```

```
File /Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/urllib/r  
equest.py:1351, in AbstractHTTPHandler.do_open(self, http_class, req, **http_co  
nn_args)  
    1348     h.request(req.get_method(), req.selector, req.data, headers,  
    1349                 encode_chunked=req.has_header('Transfer-encoding'))  
    1350     except OSError as err: # timeout error  
-> 1351     raise URLError(err)  
    1352     r = h.getresponse()  
    1353 except:
```

```
URLError: <urlopen error [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify fa  
iled: unable to get local issuer certificate (_ssl.c:1002)>
```

But I was able to download it directly from Zenodo.

In:

<https://carpentries-incubator.github.io/deep-learning-intro/3-monitor-the-model.html#measure-performance>;

The below gives me a warning:

```
from sklearn.metrics import mean_squared_error
rmse_baseline = mean_squared_error(y_test, y_baseline_prediction, squared=False)
print('Baseline:', rmse_baseline)
print('Neural network: ', test_metrics['root_mean_squared_error'])
```

```
.....
/Users/johannabayer/Documents/dl_workshop/lib/python3.11/site-packages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean squared error, use the function 'root_mean_squared_error'.
  warnings.warn(
```

<https://carpentries-incubator.github.io/deep-learning-intro/3-monitor-the-model.html#save-model>

save model command requires .keras extension again.
model.save('my_tuned_weather_model.keras')

<https://carpentries-incubator.github.io/deep-learning-intro/4-advanced-layer-types.html#identify-inputs-and-outputs>

train_labels.shape
gives ((878,)
for me, not (878, 1)

<https://carpentries-incubator.github.io/deep-learning-intro/4-advanced-layer-types.html#share-model>
model.save('cnn_model.keras')

In

<https://carpentries-incubator.github.io/deep-learning-intro/5-transfer-learning.html#choose-a-pre-trained-model-or-start-building-architecture-from-scratch>

```
# upscale layer
method = tf.image.ResizeMethod.BILINEAR
upscale = keras.layers.Lambda(
    lambda x: tf.image.resize_with_pad(x, 160, 160, method=method))(inputs)
```

- ⇒ Tf was not defined – assuming tensor flow?
- ⇒ import tensorflow as tf is missing
- base_model = keras.applications.DenseNet121(include_top=False,

```

pooling='max',
weights='imagenet',
input_tensor=upscale,
input_shape=(160,160,3),
)

```

Gives me an error:

```

ifier_activation, name)
292 else:
293     if blocks == [6, 12, 24, 16]:
--> 294         weights_path = file_utils.get_file(
295             "densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5",
296             DENSENET121_WEIGHT_PATH_NO_TOP,
297             cache_subdir="models",
298             file_hash="30ee3e1110167f948a6b9946edeeb738",
299         )
300     elif blocks == [6, 12, 32, 32]:
301         weights_path = file_utils.get_file(
302             "densenet169_weights_tf_dim_ordering_tf_kernels_notop.h5",
303             DENSENET169_WEIGHT_PATH_NO_TOP,
304             cache_subdir="models",
305             file_hash="b8c4d4c20dd625c148057b9ff1c1176b",
306         )

File ~/Documents/dl_workshop/lib/python3.11/site-packages/keras/src/utils/file_utils.py:295, in
get_file(fname, origin, untar, md5_hash, file_hash, cache_subdir, hash_algorithm, extract, arch
ive_format, cache_dir, force_download)
293     raise Exception(error_msg.format(origin, e.code, e.msg))
294     except urllib.error.URLError as e:
--> 295         raise Exception(error_msg.format(origin, e.errno, e.reason))
296 except (Exception, KeyboardInterrupt):
297     if os.path.exists(fpath):

Exception: URL fetch failure on https://storage.googleapis.com/tensorflow/keras-applications/de
nset/densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5: None -- [SSL: CERTIFICATE_VERIF
Y_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1002)

```

Fixed it by downloading the weights manually and importing them from disk and also importing DenseNet121:

```
from tensorflow.keras.applications import DenseNet121
```

```

base_model = keras.applications.DenseNet121(include_top=False,
pooling='max',
weights='densenet121_weights_tf_dim_ordering_tf_kernels_notop.h5',
input_tensor=upscale,
input_shape=(160,160,3),
)

```