Keras notes

|  |
| --- |
| from tensorflow import keras  from tensorflow.keras import layers  New = True  n = int(128 \* 2)  if New:  inputs = keras.Input(shape=(len(x\_cols),), name='time')  xx = layers.Dense(n, activation='relu')(inputs)  #xx = layers.Dense(n, activation='relu')(xx)  #xx = layers.Dense(n, activation='relu')(xx)  xx = layers.Dense(n, activation='relu')(xx)  xx = layers.Dense(n, activation='relu')(xx)  outputs = layers.Dense(len(y\_cols), activation='linear')(xx)  model = keras.Model(inputs=inputs, outputs=outputs, name='simple\_model')  else:  model = keras.models.load\_model('small\_grid\_model.h5') |
| Fully connected layers are defined using the Dense class.  This ensures that all nodes are connected to all nodes in the adjacent layers.  <https://keras.io/layers/core/> |
| A Gentle Introduction to the Rectified Linear Unit (ReLU)  <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/> |
| tf.keras.backend.clear\_session() # For easy reset of notebook state. |
| Inputs are logged to reduce the dynamical range, to make the neural net happier  This uses float32 which are single precision. It’s probably good (accurate to enough descimal places) enough for our purposes, and using single precision makes it faster  x\_cols = ['star\_mass', 'star\_age']  y\_cols = ['luminosity', 'effective\_T']  x = np.log10(df\_big[x\_cols].values).astype('float32')  y = np.log10(df\_big[y\_cols].values).astype('float32') |
|  |
|  |
| inputs=keras.Input(shape=(2,))  xx=keras.layers.Dense(8,activation='relu')(inputs)  xx=keras.layers.Dense(8,activation='relu')(xx)  xx=keras.layers.Dense(8,activation='relu')(xx)  outputs=keras.layers.Dense(1,activation='linear')(xx)  model = keras.Model(inputs=inputs, outputs=outputs)  model.compile(optimizer='Nadam',  loss='MAE',  metrics=['accuracy'])  # shape=(2,) tells the neural net to take an input with 2 parameters i.e. (x,y) but the comma after 2, acts as a place holder so work can be done with the inputs  #{inputs} tells it to use the input layer as the inputs for the 2nd layer  #the subsequent layers are all named xx but they still have (xx) to reference the previous layer  # |
| Optimizers = <https://keras.io/optimizers/> |