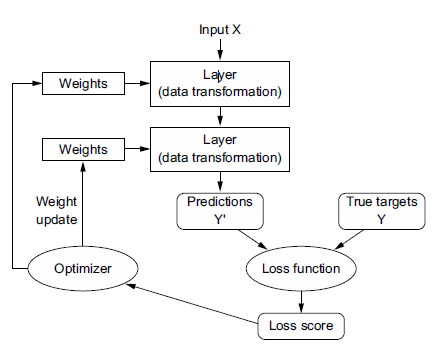
**Deep learning steps explained:**



# (with modal)

**optimizer** = 'adam'

# (with modal)

**loss** = tf.keras.losses.SparseCategoricalCrossentropy

# (with layer)

**activation**="softmax" (each layer)

**some basic layers usually used in deep learning projects.**

* **TextVectorization layer**:

A preprocessing layer which maps text features to integer sequences

* **Embedding** **layer**:

Turns positive integers (indexes) into dense vectors of fixed size,This layer can only be used on positive integer inputs of a fixed range.

input: 2D tensor ,output: 3D tensor

* **Rescaling layer:**

preprocessing layer which rescales input values to a new range. This layer rescales every value of an input (often an image) by multiplying by scale and adding offset

* **Dense layer**: Dense implements the operation: output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer (only applicable if use\_bias is True).
* 2D **convolution** **layer**.

This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs. If use\_bias is True, a bias vector is created and added to the outputs. Finally, if activation is not None, it is applied to the outputs as well.

* **MaxPooling2D:**

Max pooling operation for 2D spatial data.The main purpose of pooling is to reduce the size of feature maps, which in turn makes computation faster because the number of training parameters is reduced. The pooling operation summarizes the features present in a region, the size of which is determined by the pooling filter.

Downsamples the input along its spatial dimensions (height and width) by taking the maximum value over an input window (of size defined by pool\_size) for each channel of the input. The window is shifted by strides along each dimension.

* **Flatten** :

Flattens the input. Does not affect the batch size.

* **Dropout** : is a technique where randomly selected neurons are ignored during training. They are “dropped out” randomly.