Keras layers API

Layers are the basic building blocks of neural networks in Keras. A layer consists of a tensor-in tensor-out computation function (the layer's `call` method) and some state, held in TensorFlow variables (the layer's *weights*).

A Layer instance is callable, much like a function:

```
from tensorflow.keras import layers

layer = layers.Dense(32, activation='relu')
inputs = tf.random.uniform(shape=(10, 20))
outputs = layer(inputs)
```

Unlike a function, though, layers maintain a state, updated when the layer receives data during training, and stored in `layer.weights`:

```
>>> layer.weights
[<tf.Variable 'dense/kernel:0' shape=(20, 32) dtype=float32>,
 <tf.Variable 'dense/bias:0' shape=(32,) dtype=float32>]
```

Creating custom layers

While Keras offers a wide range of built-in layers, they don't cover ever possible use case. Creating custom layers is very common, and very easy.

See the guide Making new layers and models via subclassing for an extensive overview, and refer to the documentation for the base `Layer` class.

Layers API overview

The base Layer class

- Layer class
- weights property
- trainable_weights property
- non_trainable_weights property
- add_weight method
- trainable property
- get_weights method
- set_weights method
- get_config method
- add_loss method
- losses property

Layer activations

- relu function
- sigmoid function
- softmax function
- softplus function
- softsign function
- tanh function
- selu function
- elu function
- exponential function
- leaky_relu function
- relu6 function

- silu function
- gelu function
- hard_sigmoid function
- linear function
- mish function
- log_softmax function

Layer weight initializers

- RandomNormal class
- RandomUniform class
- TruncatedNormal class
- Zeros class
- Ones class
- GlorotNormal class
- GlorotUniform class
- HeNormal class
- HeUniform class
- Orthogonal class
- Constant class
- VarianceScaling class

Layer weight regularizers

- L1 class
- L2 class
- L1L2 class
- OrthogonalRegularizer class

Layer weight constraints

- MaxNorm class
- MinMaxNorm class
- NonNeg class
- UnitNorm class

Core layers

- Input object
- Dense layer
- Activation layer
- Embedding layer
- Masking layer
- Lambda layer

Convolution layers

- Conv1D layer
- Conv2D layer
- Conv3D layer
- SeparableConv1D layer
- SeparableConv2D layer
- DepthwiseConv2D layer
- Conv1DTranspose layer
- Conv2DTranspose layer
- Conv3DTranspose layer

Pooling layers

- MaxPooling1D layer
- MaxPooling2D layer
- MaxPooling3D layer
- AveragePooling1D layer
- AveragePooling2D layer
- AveragePooling3D layer
- GlobalMaxPooling1D layer
- GlobalMaxPooling2D layer
- GlobalMaxPooling3D layer
- GlobalAveragePooling1D layer
- GlobalAveragePooling2D layer
- GlobalAveragePooling3D layer

Recurrent layers

- LSTM layer
- GRU layer
- SimpleRNN layer
- TimeDistributed layer
- Bidirectional layer
- ConvLSTM1D layer
- ConvLSTM2D layer
- ConvLSTM3D layer
- Base RNN layer

Preprocessing layers

- Text preprocessing
- Numerical features preprocessing layers
- Categorical features preprocessing layers
- Image preprocessing layers
- Image augmentation layers

Normalization layers

- BatchNormalization layer
- LayerNormalization layer
- UnitNormalization layer
- GroupNormalization layer

Regularization layers

- Dropout layer
- SpatialDropout1D layer
- SpatialDropout2D layer
- SpatialDropout3D layer
- GaussianDropout layer
- GaussianNoise layer
- ActivityRegularization layer

Attention layers

- MultiHeadAttention layer
- Attention layer
- AdditiveAttention layer

Reshaping layers

- Reshape layer
- Flatten layer
- RepeatVector layer
- Permute layer
- Cropping1D layer
- Cropping2D layer
- Cropping3D layer
- UpSampling1D layer
- UpSampling2D layer
- UpSampling3D layer
- ZeroPadding1D layer
- ZeroPadding2D layer
- ZeroPadding3D layer

Merging layers

- Concatenate layer
- Average layer
- Maximum layer
- Minimum layer
- Add layer
- Subtract layer
- Multiply layer
- Dot layer

Activation layers

- ReLU layer

- Softmax layer
- LeakyReLU layer
- PReLU layer
- ELU layer

Backend-specific layers

- TorchModuleWrapper layer