Implementing Custom Layers and Models

Janani Ravi
CO-FOUNDER, LOONYCORN

www.loonycorn.com

Overview

Defining custom layers using the Layer base class

Layers with trainable and non-trainable weights

Inferring the input shape

Recursively composing custom layers

Building and training custom models

Saving and loading custom models

Keras Building Blocks

Sequential Models

Functional APIs

Model Subclassing

Custom Layers

Keras Building Blocks

Sequential Models

Functional APIs

Model Subclassing

Custom Layers

Keras Building Blocks

Sequential Models

Functional APIs

Model Subclassing

Custom Layers

Model

tf.keras.Model

Can be trained

Can encapsulate multiple layers

Can be subclassed

- Model subclassing

Model Subclassing

Subclass tf.keras.Model and only define your own forward pass imperatively - particularly useful with eager execution.

Eager Execution

Evaluate models immediately (“build-and-run”) rather than first building graphs that are run later (“build-then-run”)

Keras Building Blocks

Sequential Models

Functional APIs

Model Subclassing

Custom Layers

tf.keras.layers.Layer

Contains a call method which defines the transformation applied to input to obtain the output. Also contains a set of weights.

Layers

The call method defines forward pass

Weights can be

- Trainable
- Non-trainable

Non-trainable weights will be unchanged during back-propagation

Best practice: Defer weight creation until shape of inputs is known

Layers

Layers can be recursively combined

- One layer as part of state of another
- Outer layer will depend on weights of inner layer

Layers

Layers recursively collect losses created during the forward pass

- Regularization losses

Custom Layers

Subclass tf.keras.layers.Layer

Implement specific methods

_ _init_ _

build

call

get_config/from_config

Demo

Building custom layers and models in Keras

Demo

Performing regression using a custom model with custom layers

Summary

Defining custom layers using the Layer base class

Layers with trainable and non-trainable weights

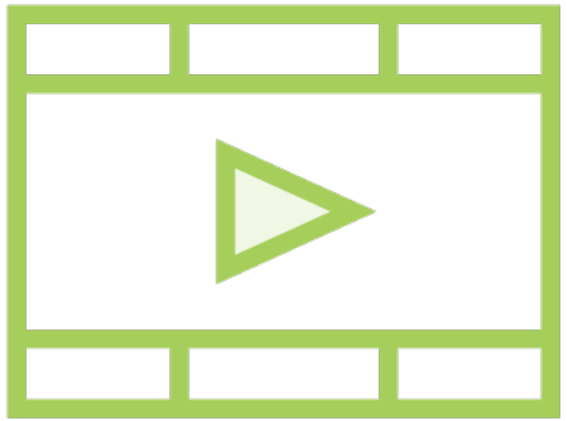
Inferring the input shape

Recursively composing custom layers

Building and training custom models

Saving and loading custom models

Related Courses



Expediting Deep Learning with Transfer Learning: PyTorch Playbook

Natural Language Processing with PyTorch