Most Used PyTorch *Functions*

Most Used PyTorch Functions

Tensor Creation

- torch.tensor(): Creates a tensor from a Python list or NumPy array.
- torch.zeros(): Creates a tensor filled with zeros.
- torch.ones(): Creates a tensor filled with ones.
- torch.rand(): Creates a tensor with random values.
- torch.randn(): Creates a tensor with random values from a normal distribution.

Tensor Operations

- torch.add(): Element-wise addition of two tensors.
- torch.sub(): Element-wise subtraction of two tensors.
- torch.mul(): Element-wise multiplication of two tensors.
- torch.div(): Element-wise division of two tensors.
- torch.matmul(): Matrix multiplication of two tensors.
- torch.dot(): Dot product of two tensors.
- torch.sum(): Computes the sum of all elements in a tensor.
- torch.mean(): Computes the mean of all elements in a tensor.
- torch.max(): Computes the maximum value in a tensor.
- torch.min(): Computes the minimum value in a tensor.

Autograd

- torch.autograd.grad(): Computes the gradient of a tensor with respect to another tensor.
- torch.autograd.backward(): Computes the gradients of a tensor with respect to all inputs.

Neural Networks

- torch.nn.Module(): Base class for all neural network modules.
- torch.nn.Sequential(): Creates a sequential neural network.
- torch.nn.Linear(): Creates a fully connected linear layer.
- torch.nn.Conv2d(): Creates a 2D convolutional layer.
- torch.nn.ReLU(): Creates a ReLU activation function.
- torch.nn.Sigmoid(): Creates a sigmoid activation function.
- torch.nn.Tanh(): Creates a tanh activation function.

Optimization

- torch.optim.SGD(): Creates a stochastic gradient descent optimizer.
- torch.optim.Adam(): Creates an Adam optimizer.
- torch.optim.RMSprop(): Creates an RMSprop optimizer.
- torch.optim.lr_scheduler(): Creates a learning rate scheduler.

Data Loaders

- torch.utils.data.DataLoader(): Creates a data loader from a dataset.
- torch.utils.data.Dataset(): Base class for all datasets.

Model Persistence

- torch.save(): Saves a model to a file.
- torch.load(): Loads a model from a file.

Utilities

- torch.device(): Specifies the device (CPU or GPU) for a tensor.
- torch.cuda.is_available(): Checks if a CUDA device is available.
- torch.backends.cudnn.enabled: Enables or disables CuDNN acceleration.