



DEEP LEARNING TAKEAWAYS



Chapter:

Getting Started with PyTorch

Matrix Fundamentals

- 1** Matrix is a table like arrangement of numbers.
- 2** Matrix arithmetic (Addition, Subtraction, Multiplication etc.) helps in solving many business problems
- 3** In neural networks, weights can be efficiently multiplied with the output from the previous layer using matrix multiplication. If you are using a GPU, this becomes even faster as it will use multiple cores to compute dot products in parallel
- 4** Neural networks require a lot of matrix multiplications and that is the reason why GPUs got very popular for deep learning as it helps in parallel processing
- 5** You can perform matrix multiplication in two ways (1) Element wise multiplication (a.k.a. Hadamard product) (2) Matrix multiplication
- 6** Basic rule for matrix multiplication is that the columns in the first matrix must be equal to rows in the second matrix

PyTorch Tensor Basics 1

- 1** Tensor is a generic term for scalar (0 dimension tensor), vector (1 dimension tensor), matrix (2 dimension tensor) etc.
- 2** Tensor can have any number of dimensions.
- 3** Using `torch.Tensor`, you can create a tensor object. Tensor objects look very much like numpy arrays. Numpy arrays can not be created on GPU directly whereas you can create a tensor object directly in GPU memory

PyTorch Tensor Basics 2

- 1 Tensor has numpy and dataframe like attributes such as **dtype**, **shape**, **device** etc.
- 2 **view()** method allows you to reshape the tensor
- 3 **zeros()**, **ones()**, **rand()** can be used to create a new tensor with specific values
- 4 You don't need to remember these APIs. You can always use ChatGPT and Google to take help on API syntax.

Derivatives, Partial Derivatives

- 1** Slope of a line at a given point is Derivative
- 2** Derivative $\rightarrow X^n = n X^{(n-1)}$
- 3** Slope is used for Linear equations, whereas Derivative is used for non-linear equations.
- 4** Slope is constant, whereas Derivative is a function.
- 5** The purpose of a Partial Derivative is to measure how a function changes as one of its variables is varied while keeping the other variables constant.

Chain Rule

- 1** Chain rule is a technique used to compute the derivative of a function, composed of multiple functions.
- 2** Chain rule will be used in the Gradient Descent Technique.

PyTorch Autograd

- 1 Autograd feature allows to calculate gradients (i.e. partial derivatives) automatically. While training a neural network, we need to calculate gradients during backpropagation step. Automatic gradient calculation helps in this process.
- 2 **torch.no_grad** can be used if you want to temporarily stop calculating gradients.

Numpy Array vs PyTorch Tensor

- 1** PyTorch tensors and numpy arrays have similar functionality but tensor offers 3 key benefits over numpy arrays that are useful in deep learning
- 2** Benefit 1: Tensor come with in built support to leverage GPU acceleration
- 3** Benefit 2: Tensors have autograd features that computes gradients automatically. Numpy arrays do not have this feature
- 4** Benefit 3: Tensors are tightly integrated with PyTorch ecosystem that makes it easier to use with deep learning tasks