PyTorch:
An introduction

PyTorch

One of the (two more) popular ML frameworks for building neural network

- originally developed by Meta

PyTorch defines a class (i.e. customed data structure) called Tensor

- store n-dimensional rectangular arrays of numbers

PyTorch also provides automatic calculation of partial derivatives

- i.e. gradient (Autograd) for building and training neural networks

Tensors

We have earlier used Scikit-learn to perform ML based model training

- built on NumPy arrays and designed to run efficiently on CPUs

But training of big neural networks using CPU alone will be very time consuming

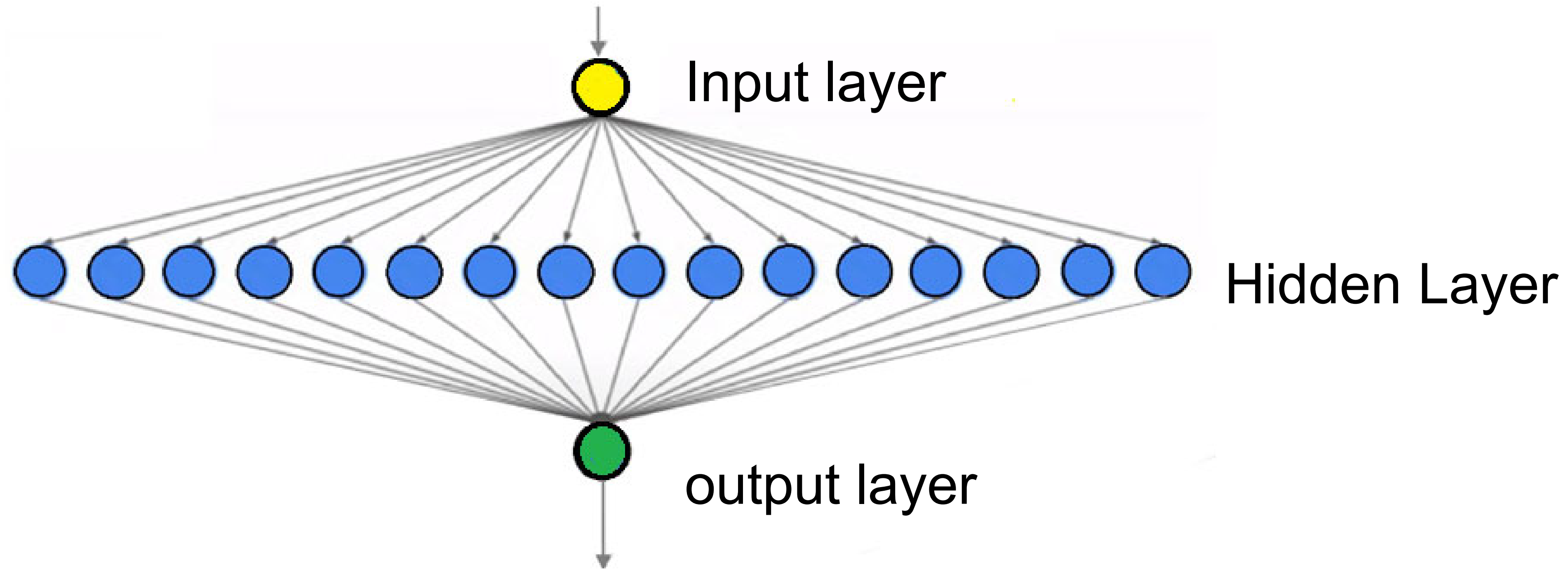
In machine learning

- special data type known as Tensor is used
- all the parameters (weights and biases) in neural networks are stored as Tensors

Tensors can be used with

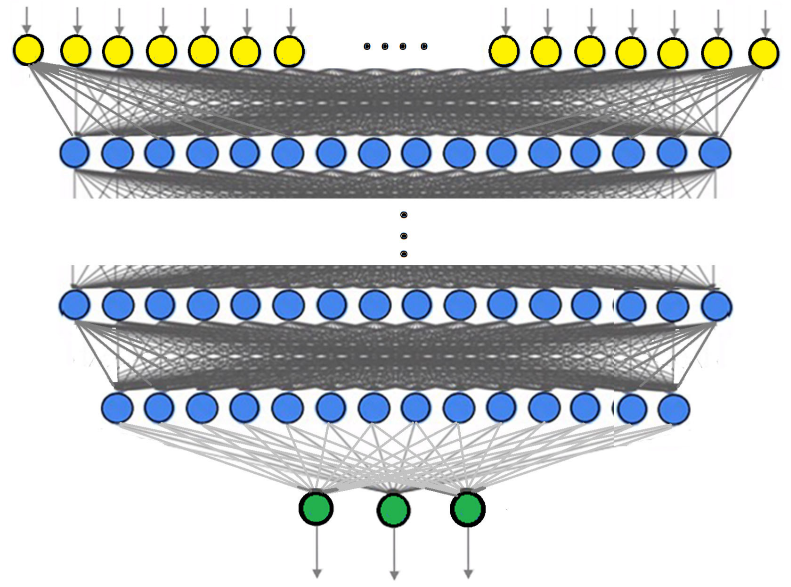
- hardware accelerators (e.g. GPU, TPU) to vastly reduce the computation time

Neurons and Neural Network



Hidden Layer

- consists of learnable parameters - the neurons
- the ‘algorithm’ that can learn and improve by itself



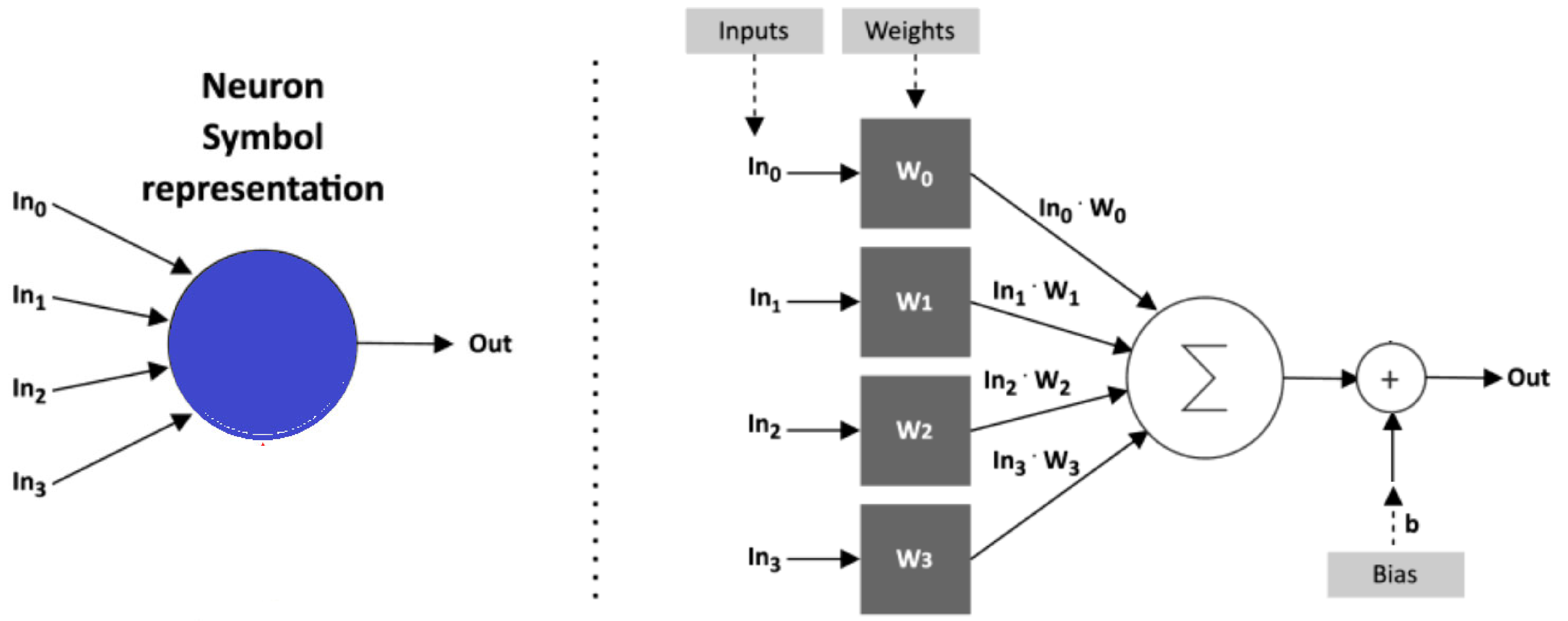
Deep Neural Network

- multiple layers of hidden layers
- much more sophisticated algorithms can be learnt

Neuron in Neural Network

A neuron produces a single output through a linear transformation

- weighted sum of the inputs - Weights
- plus a constant value called Bias

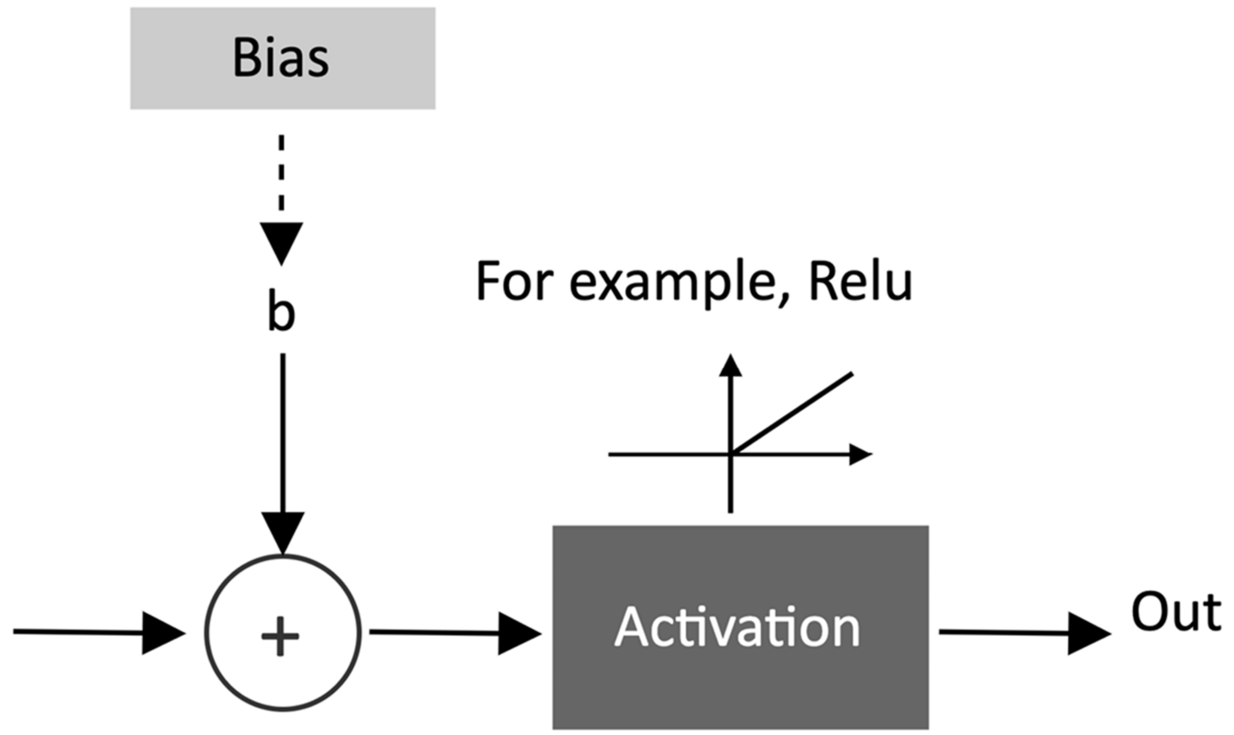


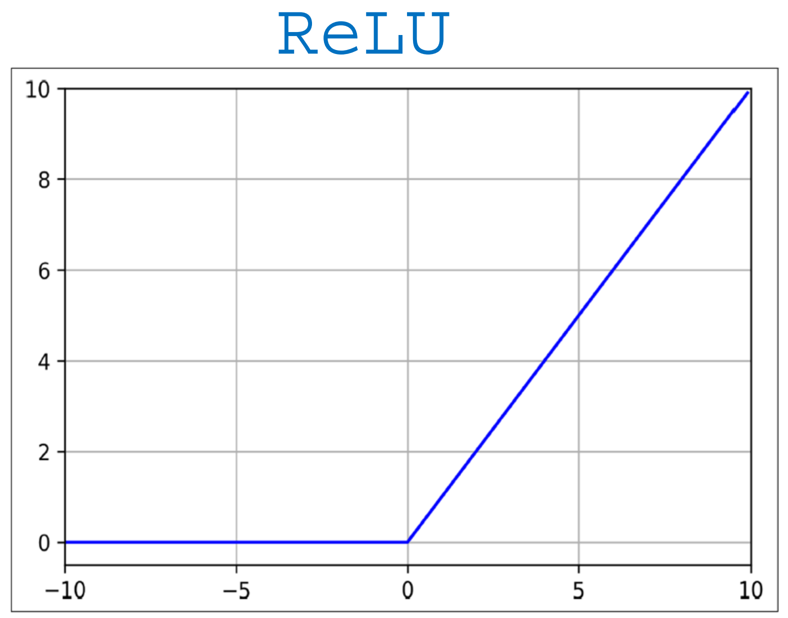
But can only solve simple linear problems with linear transformations

Activation Functions for Neuron

Add a non-linear function on the neuron's output

- to help the network able to learns complex pattern
- Example: Rectified Linear Unit (ReLU)

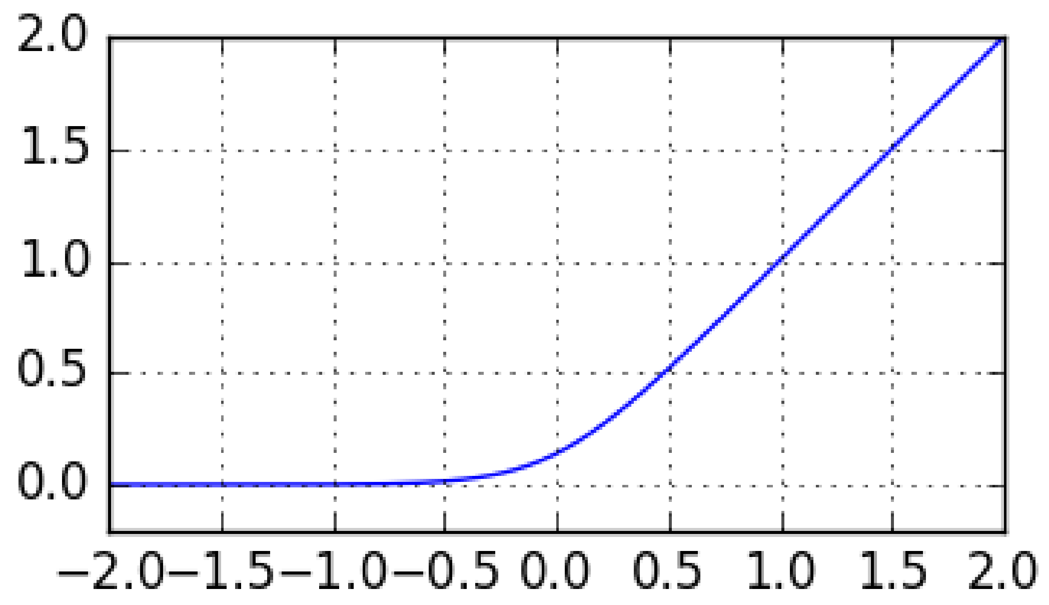




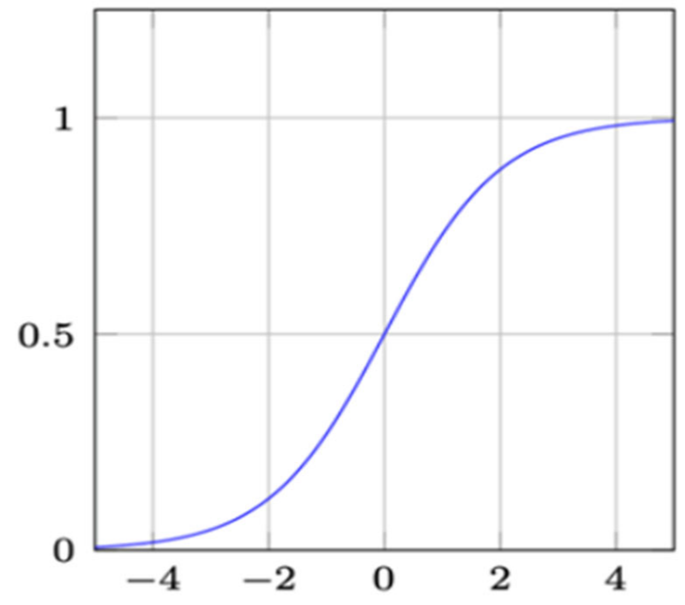
$$f(x) = \max(0, x)$$

Aside: Activation Functions

SoftPlus

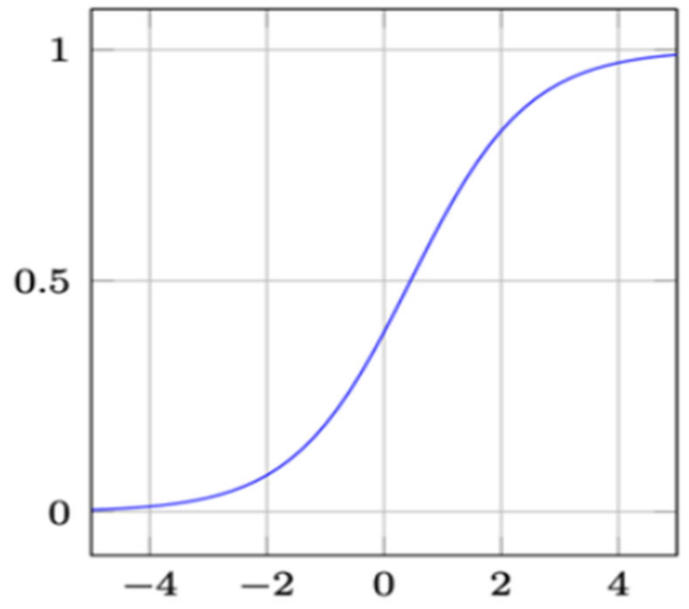


SigMoid



for multi classification at output

Softmax

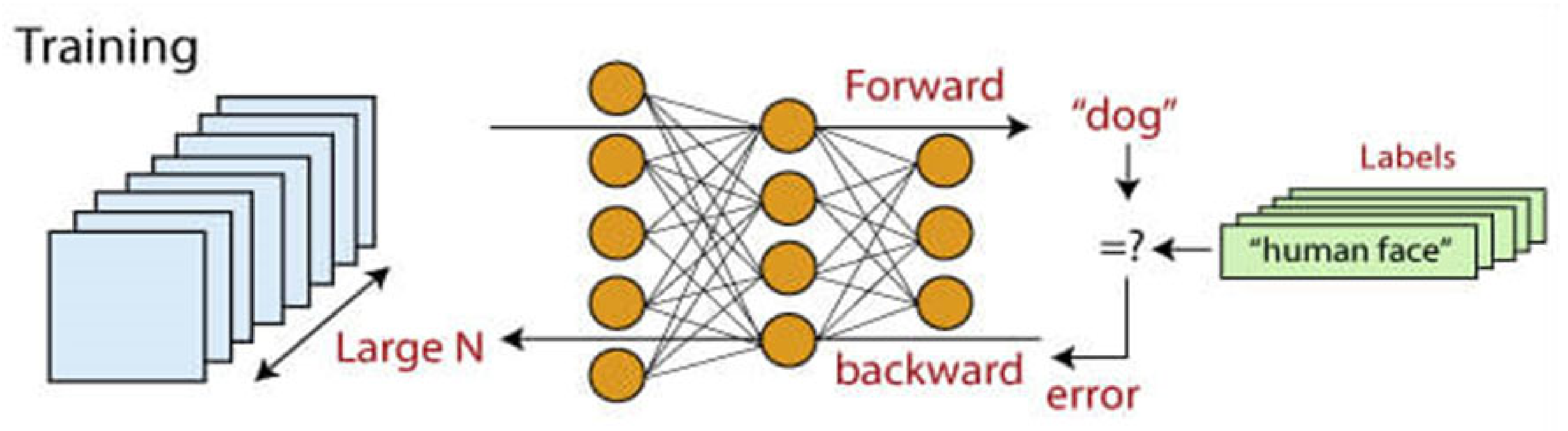


for binary classification at output

Gradient Descent and Autograd

PyTorch provides an Autograd feature

- perform automatic differentiation that is used in neural network training



During neural network training

(a) Forward propagation

- neural network runs the input data through its neurons' functions to predict the output

(b) Backward propagation

- neural network adjusts its parameters based on the errors of the prediction
- which use differentiation to calculate the gradient for gradient descent algorithm

Useful PyTorch Modules

torch
- original name before it is ported to Python
- used to create and store data in the form of ‘tensors’

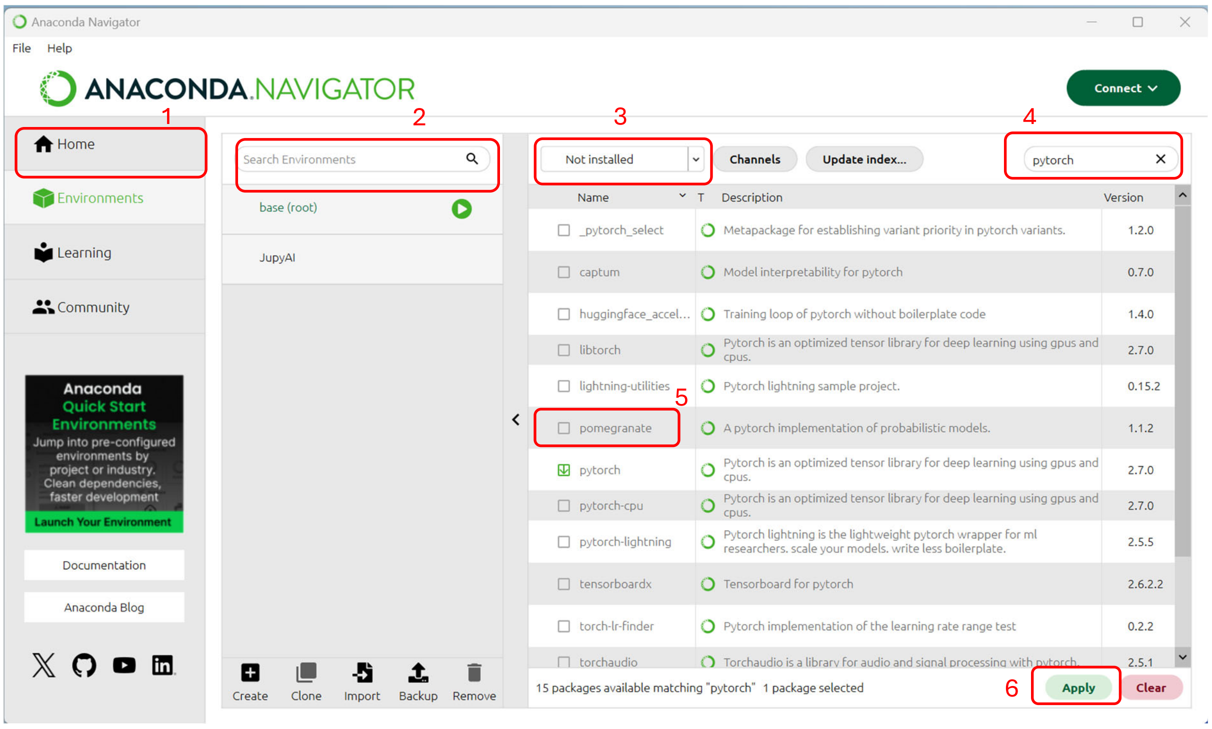
torch.Tensor
- Class for instantiate tensor

torch.nn
- use to define the neural network (weights and biases)

torch.nn.functional
- contains the activation functions (e.g. ReLU)

torch.nn.optim
- contains the optimization functions (e.g. SGD)

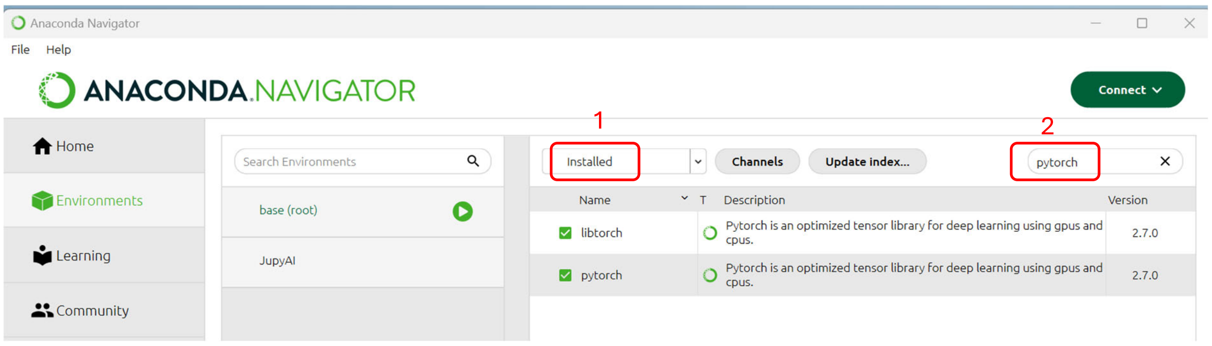
Installation of PyTorch



	Name	T	Description	Version
☐	_pytorch_select		Metapackage for establishing variant priority in pytorch variants.	1.2.0
☐	captum		Model interpretability for pytorch	0.7.0
☐	huggingface_accel...		Training loop of pytorch without boilerplate code	1.4.0
☐	libtorch		Pytorch is an optimized tensor library for deep learning using gpus and cpus.	2.7.0
☐	lightning-utilities		Pytorch lightning sample project.	0.15.2
☐	pomegranate		A pytorch implementation of probabilistic models.	1.1.2
✓	pytorch		Pytorch is an optimized tensor library for deep learning using gpus and cpus.	2.7.0
☐	pytorch-cpu		Pytorch is an optimized tensor library for deep learning using gpus and cpus.	2.7.0
☐	pytorch-lightning		Pytorch lightning is the lightweight pytorch wrapper for ml researchers. scale your models. write less boilerplate.	2.5.5
☐	tensorboardx		Tensorboard for pytorch	2.6.2.2
☐	torch-lr-finder		Pytorch implementation of the learning rate range test	0.2.2
☐	torchaudio		Torchaudio is a library for audio and signal processing with pytorch.	2.5.1

15 packages available matching "pytorch" 1 package selected

After successful installation



	Name	T	Description	Version
✓	libtorch		Pytorch is an optimized tensor library for deep learning using gpus and cpus.	2.7.0
✓	pytorch		Pytorch is an optimized tensor library for deep learning using gpus and cpus.	2.7.0