Introduction to Big Data and Data Science (CSCE 5300 Section 005)*

Yunhe Feng

Assistant Professor, Department of Computer Science and Engineering

25th September, 2024

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

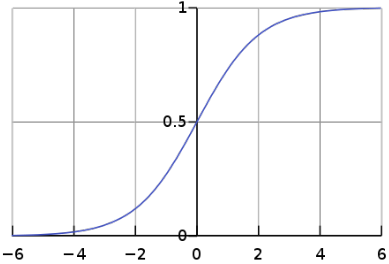
Quiz 2

- Closed-book in-person Quiz
- 5 Questions: 1 point for each question
- Quiz time: 2:35 pm - 3:00 pm

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

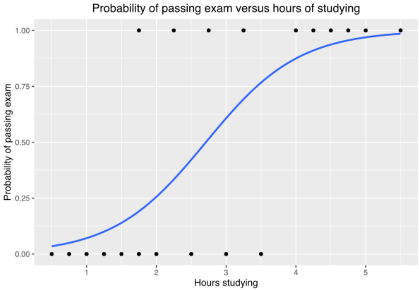
1. Logistic Regression
2. Performance Measurement for ML Classification
3. Deep Learning with PyTorch
4. Assignment

What is Logistic Regression



- Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.
- The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

Logistic Function



$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

- μ is a location parameter (the midpoint of the curve, where $p(\mu) = 1/2$)
- s is a scale parameter

Types of Logistic Regression

- Binary logistic regression (Sigmoid)
  - Response variable can only belong to one of two categories
  - Spam detection
- Multinomial logistic regression (softmax regression)
  - Response variable can belong to one of three or more categories and there is no natural ordering among the categories
  - Sports preference
- Ordinal logistic regression (proportional odds model)
  - Response variable can belong to one of three or more categories and there is a natural ordering among the categories
  - Movie ratings

Types of Logistic Regression

- Binary logistic regression (Sigmoid)

$$P(Y = 1 \mid \mathrm{x}) = \sigma(\mathrm{x}^\top \boldsymbol{\beta}) = \frac{1}{1 + e^{-(\mathrm{x}^\top \boldsymbol{\beta})}}$$

- Multinomial logistic regression (softmax regression)

$$P(Y = k \mid \mathrm{x}) = \frac{e^{\mathrm{x}^\top \boldsymbol{\beta}_k}}{\sum_{j=1}^{K} e^{\mathrm{x}^\top \boldsymbol{\beta}_j}}$$

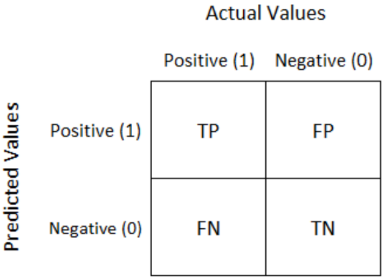
- Ordinal logistic regression (proportional odds model)

$$\log\left(\frac{P(Y > k)}{P(Y \leq k)}\right) = \alpha_k - \mathrm{x}^\top \boldsymbol{\beta}$$

1. Logistic Regression
2. Performance Measurement for ML Classification
3. Deep Learning with PyTorch
4. Assignment

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Confusion Matrix[1]

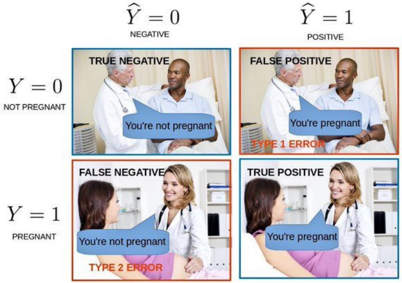


		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes.
- It is a table with combinations of predicted and actual values.

[1] shorturl.at/aJS24

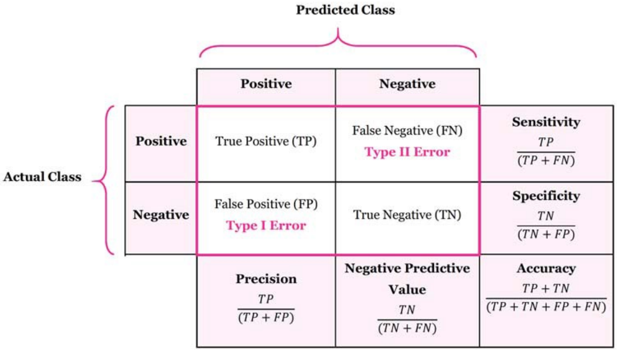
Confusion Matrix[2]



- True Positive: We predicted positive and it's true.
- True Negative: We predicted negative and it's true.
- False Positive (Type 1 Error)- We predicted positive and it's false.
- False Negative (Type 2 Error)- We predicted negative and it's false.

[2] shorturl.at/aJS24

Confusion Matrix[3]



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP+FN}$$

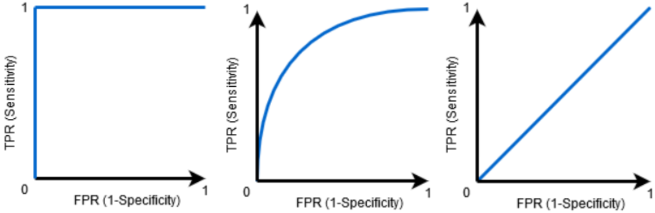
$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP+FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN+FP}$$

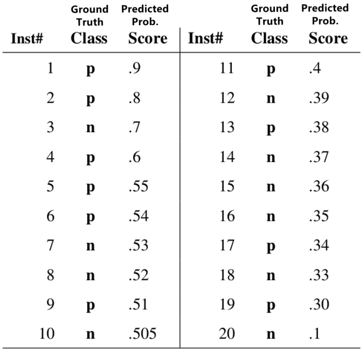
$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN+FP}$$

[3] https://tinyurl.com/4afwpcch

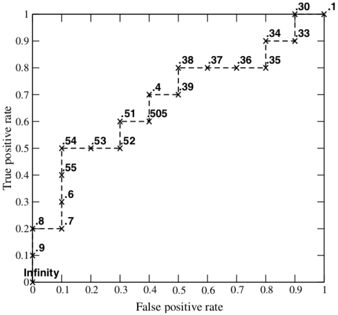
AUC-ROC (Area Under the Curve of Receive Characteristic Operator) Curve



AUC-ROC (Area Under the Curve of Receive Characteristic Operator) Curve



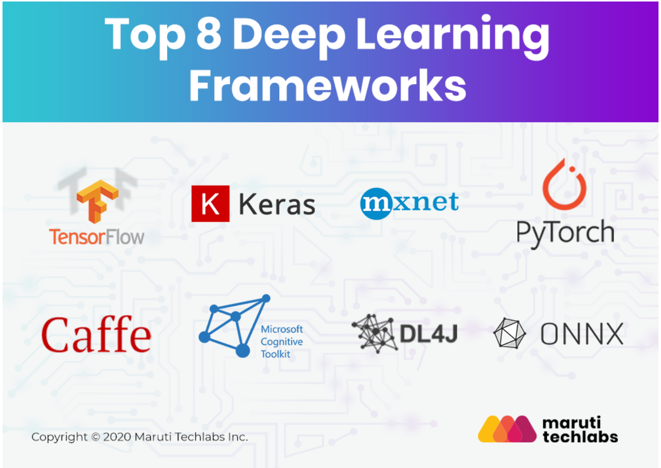
Inst#	Ground Truth Class	Predicted Prob. Score	Inst#	Ground Truth Class	Predicted Prob. Score
1	**p**	.9	11	**p**	.4
2	**p**	.8	12	**n**	.39
3	**n**	.7	13	**p**	.38
4	**p**	.6	14	**n**	.37
5	**p**	.55	15	**n**	.36
6	**p**	.54	16	**n**	.35
7	**n**	.53	17	**p**	.34
8	**n**	.52	18	**n**	.33
9	**p**	.51	19	**p**	.30
10	**n**	.505	20	**n**	.1



1. Logistic Regression
2. Performance Measurement for ML Classification
3. Deep Learning with PyTorch
4. Assignment

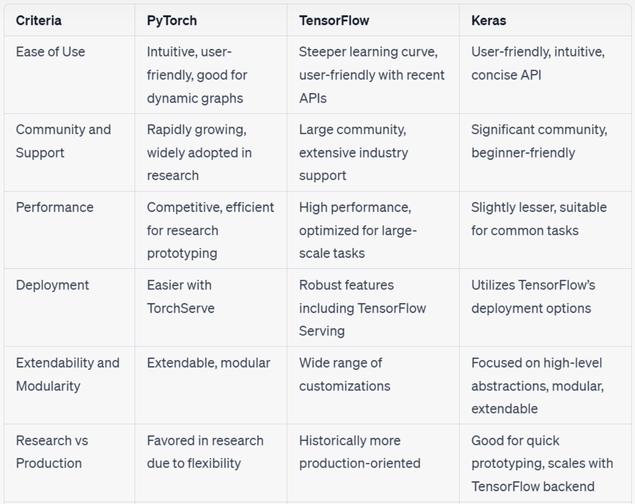
*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Deep Learning Frameworks



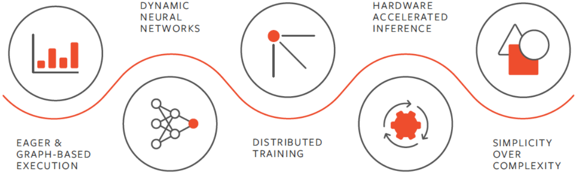
Copyright © 2020 Maruti Techlabs Inc.

PyTorch Vs TensorFlow Vs Keras



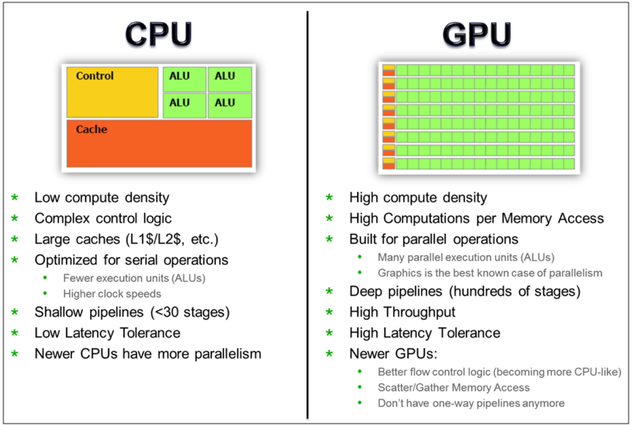
Criteria	PyTorch	TensorFlow	Keras
Ease of Use	Intuitive, user-friendly, good for dynamic graphs	Steeper learning curve, user-friendly with recent APIs	User-friendly, intuitive, concise API
Community and Support	Rapidly growing, widely adopted in research	Large community, extensive industry support	Significant community, beginner-friendly
Performance	Competitive, efficient for research prototyping	High performance, optimized for large-scale tasks	Slightly lesser, suitable for common tasks
Deployment	Easier with TorchServe	Robust features including TensorFlow Serving	Utilizes TensorFlow's deployment options
Extendability and Modularity	Extendable, modular	Wide range of customizations	Focused on high-level abstractions, modular, extendable
Research vs Production	Favored in research due to flexibility	Historically more production-oriented	Good for quick prototyping, scales with TensorFlow backend

PyTorch



- Large and vibrant community at PyTorch.org
- Well supported by major cloud platforms
- Supports CPU, GPU, and parallel processing, as well as distributed training

CPU VS GPU[4]



CPU

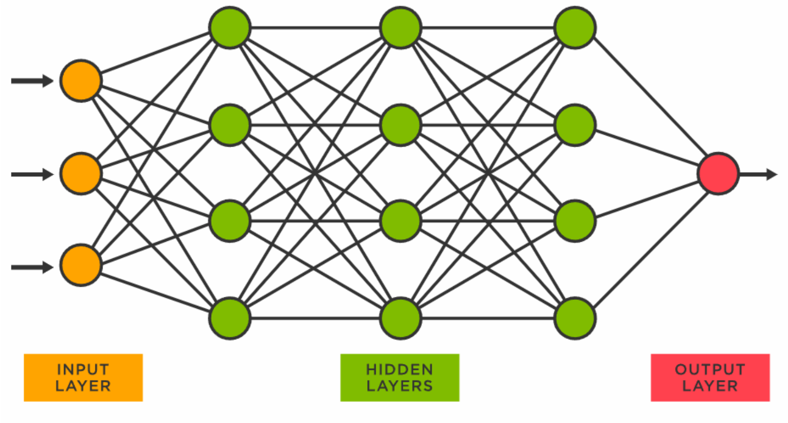
- Low compute density
- Complex control logic
- Large caches (L1$/L2$, etc.)
- Optimized for serial operations
  - Fewer execution units (ALUs)
  - Higher clock speeds
- Shallow pipelines (<30 stages)
- Low Latency Tolerance
- Newer CPUs have more parallelism

GPU

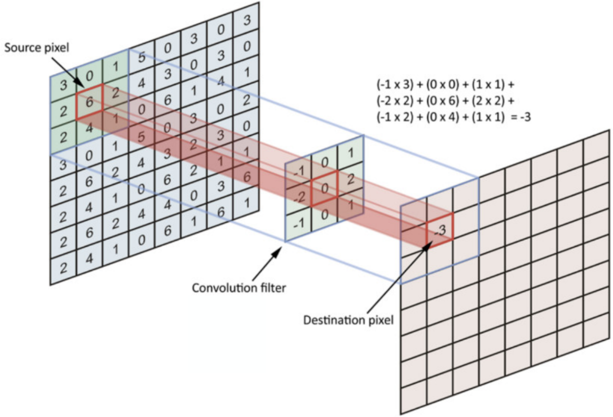
- High compute density
- High Computations per Memory Access
- Built for parallel operations
  - Many parallel execution units (ALUs)
  - Graphics is the best known case of parallelism
- Deep pipelines (hundreds of stages)
- High Throughput
- High Latency Tolerance
- Newer GPUs:
  - Better flow control logic (becoming more CPU-like)
  - Scatter/Gather Memory Access
  - Don't have one-way pipelines anymore

[4] `https://tinyurl.com/yc8b3nts`

What is a Neural Network

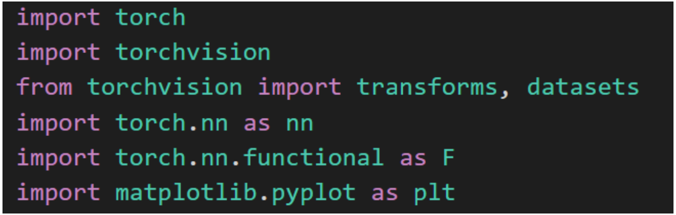


Convolution Neural Network



(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3

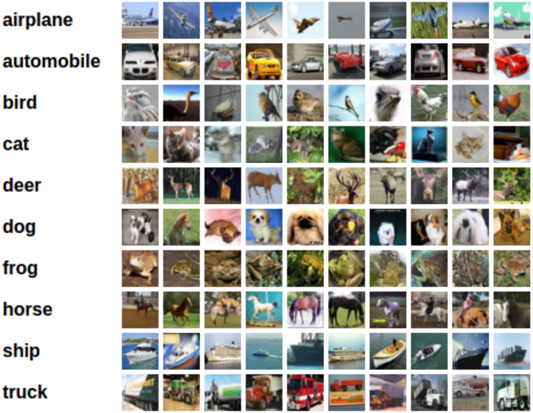
Deep Learning with PyTorch



```
import torch
import torchvision
from torchvision import transforms, datasets
import torch.nn as nn
import torch.nn.functional as F
import matplotlib.pyplot as plt
```

- PyTorch is a Python package that provides two high-level features:
  - Tensor computation (like NumPy) with strong GPU acceleration
  - Deep neural networks built on a tape-based autograd system
- The torchvision package consists of popular datasets, model architectures, and common image transformations for computer vision.

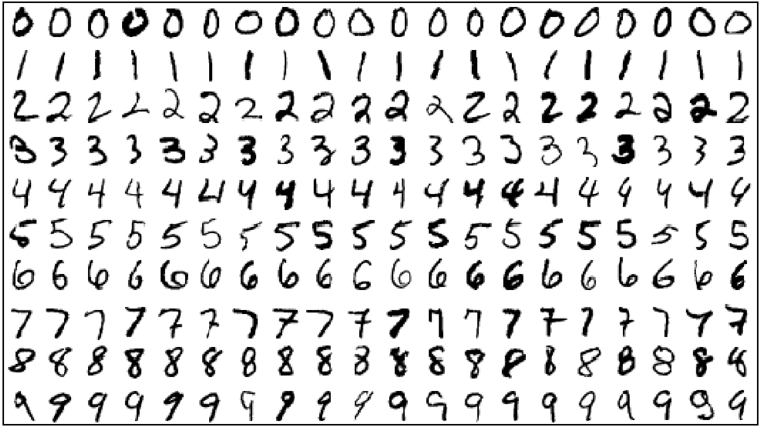
Deep Learning Example with CIFAR10



Tutorial:

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

Deep Learning Example with MNIST



1. Logistic Regression
2. Performance Measurement for ML Classification
3. Deep Learning with PyTorch
4. **Assignment**

*The teaching materials are reorganized and reformed based on Prof. Ravi Vadapalli's slides (Ravi.Vadapalli@unt.edu, UNT & University of Miami)

Assignment-5 (4.0 pts.)

- Plot ROC (2 pts.)
- Solving MNIST using PyTorch (2 pts.)