Machine Learning

Definitions

- Basic Definitions
  - **Artificial Intelligence** A branch of Computer Science that aims to enable machines to *behave and think like humans*.
  - **Machine Learning** A branch of Artificial Intelligence that aims to enable machines to *simulate human learning*.
- Performance Measures
  - **Accuracy** The ratio of the correctly classified observations to the total observations.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

  - **Precision** The ratio of the observations correctly classified as positive to the total positive observations.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

  - **Recall/Sensitivity** The ratio of the correctly classified positive observations to the total observations in the actual class.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  - **Specificity** The proportion of actual negatives which were classified as negative.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

  - **F1 Score** The weighted average of Precision and Recall.

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Other Definitions
  - **Cross Validation** A model evaluation technique where the data set is divided into *k* partitions, the model is then trained on each of these partitions.
  - **Linear regression** One of the Machine Learning algorithms. It is a statistical method that is used for *predictive analysis*.

Model Evaluation Techniques

- Mean Absolute Error (MAE)
- Root Mean Square Error (RMSE)
- Coefficient of determination or R2

- Adjusted R2

Machine Learning Cycle

1. Data Gathering
2. Data Preparation
3. Data Wrangling
4. Data Analysis
5. Model Training
6. Model Testing and Validation
7. Deployment

FPGA

Ch1

Classification of Device Technologies

- Discrete logic
- Programmable devices (FPGAs)
- ASIC (Application specific IC)

System Representation

- Behavioral level
- Structural level
- Physical level

System Abstraction Levels

- Transistor level
- Gate level
- Register transfer level (RTL)
- Processor Level

FPGA Definition

An array of gates in addition to some other elements such as memory controllers and communication interfaces that can be programmed and reconfigured more than once.

FPGA Applications

Cryptography, ASIC prototyping, Industrial, medical and Scientific Instruments, Audio/Video and Image processing and broadcasting, high-

performance computing, AI, and Deep Learning, Military and Space applications, Networking, packet processing, and other communications

FPGA Design Flow

1. **Architecture design** involves the analysis of the project requirements, problem decomposition.
2. **HDL design entry** using HDLs to describe the design.
3. **Test environment design** writing test environments and behavioral models to ensure the correctness of the HDL description.
4. **Behavioral simulation** comparing the HDL outputs and the behavioral model to ensure the correctness of the HDL description
5. **Synthesis** involves the conversion of an HDL description to a netlist
6. **Implementation** a synthesizer-generated netlist is mapped onto particular device's internal structure
7. **Timing analysis** to ensure the implemented design satisfies the timing constraints

Ch3

Unfamiliar Verilog Operators

- Exponentiation: `**`
- Concatenation: `{}{}` → `{2'b01}{2'b10}` yields `4'b0110`
- Replication: {{}} → `{3{2'b01}}` yields `6'b010101`

Coding Styles

- Structural modeling: gate level → `and(x, a, b);`
- Dataflow modeling: C-style assignments → `assign z = x && !y;`
- Behavioral modeling: `initial` and `always` blocks →

```
always @ (x or y)
   begin
     z=0;
     if (x ==y)
     z=1;
   end
```

More Notes

Uses

- `wire` VS `reg`
  - `wire` is used for continuous assignments and must be continuously driven.
  - `reg` is used for procedural assignments and can store values.
- Module port directions: input, output, and inout.
- Compiler directives are used to control the compilation of a Verilog description.

Passing Parameters to Modules

- `module-name instance-name(.formal(actual), …);`