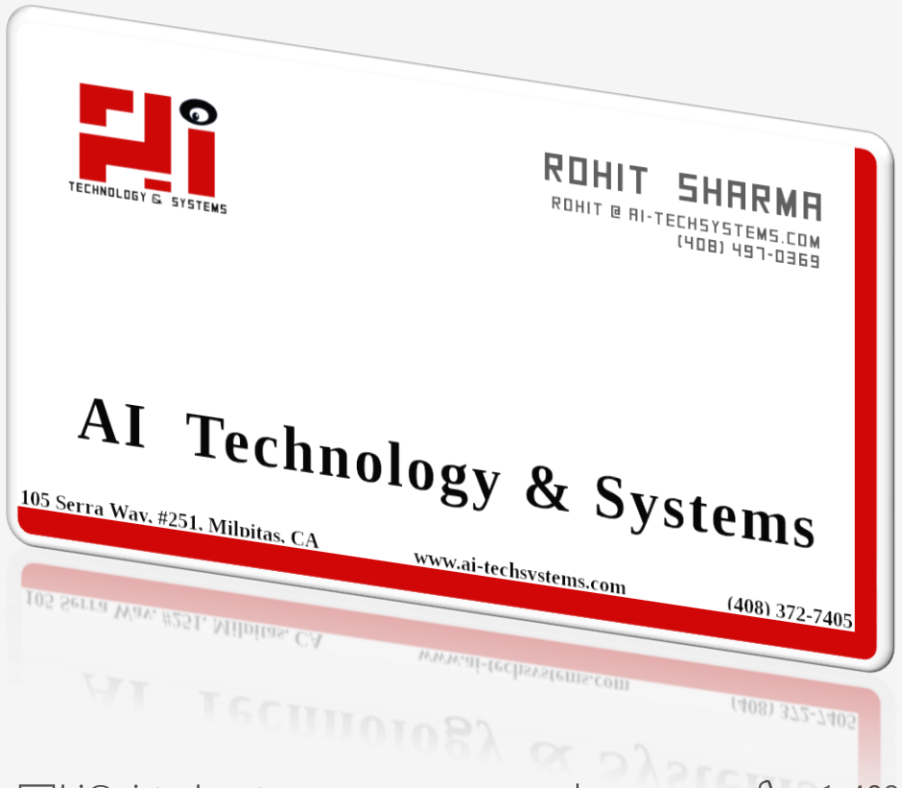
Designing & Training FFNN

California Science and Technology UNIVERSITY

About Me

- Author and Technologist
- Worked for TI, Magma, Apache, Cadence, Paripath and now AITS.
- 20 years n EDA/CAD/ML industry

- www.linkedin.com/in/srohit0/
- medium.com/@srohit0
- qr.ae/TWGSt9
- github.com/srohit0
- twitter.com/srohit



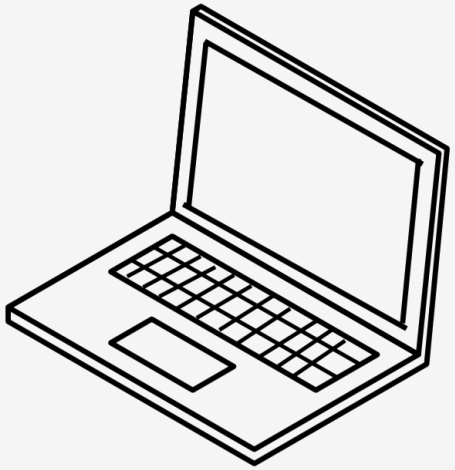
Course Overview

- Pre-requisites
  - basic computer science principles and skills
  - Probability theory
  - Multivariable calculus and Linear algebra
- Applied course with emphasis on real life projects
- Math and programming makes it fun and challenging
- Make friends for study groups for projects.
- Reference Book :
  - Machine Intelligence, Rohit Sharma, 2018.
  - srohit0.github.io/mida/

Homework	Quiz	Midterm Project	Final Project	Final Exam	Participation	Total
5%	15%	20%	25%	30%	5%	100%

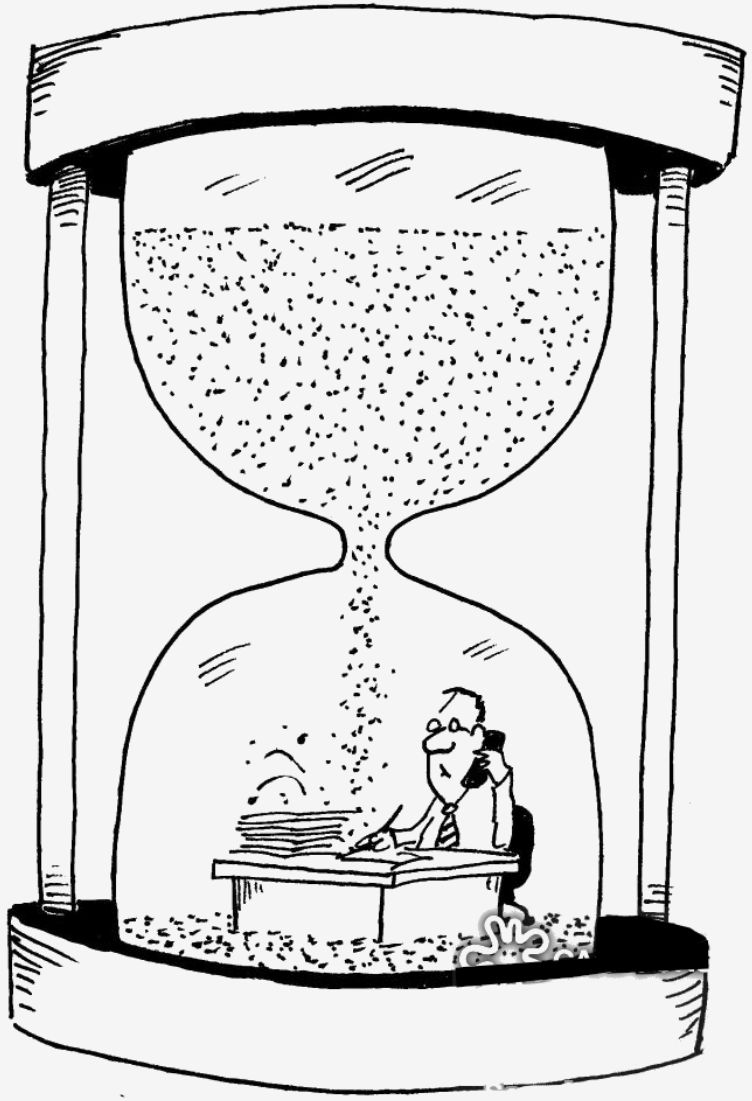
Material

- Text Book:
  - Machine Intelligence by Rohit Sharma
- References Material:
  - Python Machine Learning, 2nd Edition, by Sebastian Raschka
  - Deep Learning, by Ian Goodfellow
- Software
  - Python
  - Google Colab or Jupyter notebook



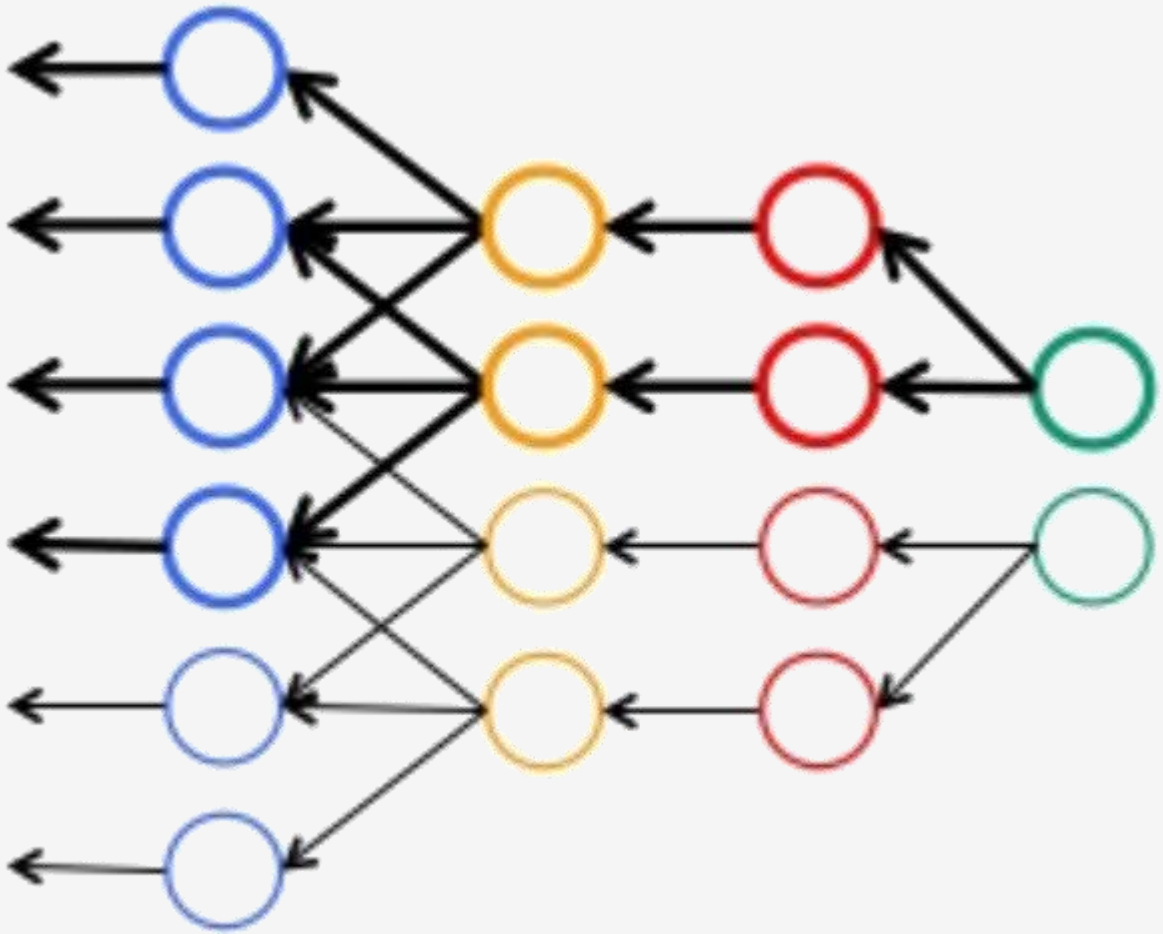
Final Projects Discussion

Mid Tern Project Submission Date – Dec 2nd



Review

- Chain Rule
- Computing Error Gradients
- Backprop algorithm
- Vanishing Gradients
- Exploding Gradients
- Workarounds
  - Gradient Clipping
  - Regularization



Design FFNN – 1

- Define problem and the value
- Select feature set
- Inspect Data
  - Cleansing
  - Augmentation
  - Add/Remove features based on domain expertise
- Use t-sne to visualize.
- Explore Data
  - Statistical properties like
    - Mean,
    - Median
    - Sigma
    - Norms
    - Range
    - Normalize
- Check covariance matrix
  - find overlapping features

Design FFNN – 2

- Randomize dataset
- Decide train-test split ration.
- Use t-sne to visualize and PCA or LDA to reduce dataset if needed.
- Decide number of features based on accuracy/precision (regression) and number of classes (classification)
  - more classes need more features
  - high precision requires more features
- Number of features becomes your input layer.
- Number of classes become output layer.

Design FFNN - 3

- Start with a few (around 5) hidden layers and observe high bias.
- Choice of activation function
  - Relu
  - Sigmoid
  - Tanh
- Tune in hyperparameters to improve stability of training.
  - Learning rate alpha
  - Learning decay
  - Weight Initialization
  - Regularization parameter like λ (L_2 norm) or dropout factor
  - Gradient Clipping
- Add more features in input and/or hidden layers to improve accuracy.

Design FFNN - 4

- Slowly increase number of epochs to few hundreds to overserve accuracy stagnation.
- If accuracy is acceptable, your model is ready for deployment.
- If not, repeat the process.

Over to Handout

California Science and Technology UNIVERSITY