Greetings, these are my notes for week 1 of the PadhAI course, covering the topics of Expert Systems and the 6 Jar framework.

[**1.1: Expert Systems**](#_4xhmggzh0tgo) **1**

[1.1.1: Limitations of Expert Systems](#_8l9ljbbamauk) 1

[**1.2: Machine Learning**](#_1grc5fdfift0) **1**

[1.2.1: Why has ML become so successful?](#_p6e6f4qsye5q) 1

[1.2.2: Six Elements(Jars) of ML](#_hbc25zkbcrf4) 2

[1.2.2a: Data](#_w7st8hsbqk50) 2

[1.2.2b: Tasks](#_x2x26lpkouvd) 2

[1.2.2c: Models](#_o09c9mq6savl) 3

[1.2.2d: Loss Function](#_yohul3fap49g) 3

[1.2.2e: Learning Algorithm](#_uwh6t27q9xmg) 3

[1.2.2d: Evaluation](#_e22df76gikcj) 4

[**1.3: Six Jars Summary**](#_usv04qh3jhbo) **4**

[1.3.1: Part 1](#_fjdve2lwov3d) 4

[1.3.2: Part 2](#_5brttvyhri8z) 6

# 1.1: Expert Systems

1. Expert systems are a set of rules which yield true/false values
2. Based on the values obtained by training examples under these rules, the output is determined appropriately
3. Similar to multiple if/else statement
4. This is the old form of AI, Popular from 1965 all the way to the 90s
   1. 1965: Used in organic chemistry to find out structures based on rules.
   2. 1990: Mission control for space programs.
      1. Checks for fuel, pressure, electricals etc before giving the go ahead
   3. 1990: Mortgage/Loan eligibility systems.
   4. 1995: Alice, a chatbot.
   5. 2018: Arabic Speech synthesis system, Hanane et al, May 2018.

## 1.1.1: Limitations of Expert Systems

1. Too many rules to write down
2. Certain rules are very hard to define
3. Some rules cannot be coded, too abstract for computers
4. Rules can sometimes be unknowns, ie task of predicting Ebola

# 1.2: Machine Learning

1. Machine Learning outsources the rule defining to the computer, by allowing it to find relationships between the input conditions and the output values
2. The Best fitting line/curve/boundary is found by iteratively trying algorithms to arrive at the best one
3. Deep learning is a subset of Machine Learning

## 1.2.1: Why has ML become so successful?

1. Abundance of Data, multimodal data (text of various languages, images, videos etc)
2. Democratized model and Learning Algorithms(Strong community)
3. Relatively fast and cheap cloud/computing

## 1.2.2: Six Elements(Jars) of ML

### 1.2.2a: Data

Data Prerequisites

1. For ML, training data with input and output is required for supervised learning
2. All data encoded as numbers
3. Typically High dimensional

Data Curation

1. Open Source: Data available online (Google AI, data.gov.in, UCI Machine Learning Repository)
2. Crowdsourcing: If you are rich, outsource data-gathering online (amazon mechanical turk, dataturks, figure eight)
3. Self work(You are smart): You can create your own data for training.
4. Wikidata: Has a lot of hindi-english translations

### 1.2.2b: Tasks

What do you do with the data

1. For eg: Take unstructured text and use it to populate a specs table in an ecommerce product page
2. For eg: Use review, spec data and faqs from an ecommerce product page and use it to answer customer questions
   1. Can also use personal data
3. For eg: From images, identify people like on Facebook
4. For eg: From images, identify location
5. For eg: From text posts, relate it to relevant advertisements
6. Two major types of supervised tasks (99% of economic value in AI)
   1. Regression:
      1. Predict continuous numerical values based on training data
   2. Classification
      1. Predict whether data falls into one category or other (numerous categories possible)
7. Two major types of unsupervised learning
   1. Clustering
      1. Organizing data into unlabeled classes
   2. Generation
      1. Given a number of examples of a particular type of data, create a new example in the trend of the previously given training examples
      2. GANS for image recreation
      3. AI bot that recreated a script for an episode of the office
      4. Tweet recreation in style of training data

### 1.2.2c: Models

How to choose the best function to predict our test set data

1. Choose a function that best fits the training data
2. The aim is to minimize the cost, ie to minimize error between predicted results and actual results
3. Various options for cost functions, ie mean-sq-error etc
4. Come up with the function that has lowest cost without having high bias or variance
5. In this course, we’ll be dealing with the Neural Network family of functions
6. Overly complex functions results in overfitting of the model, which leads to bad results in the test data

### 1.2.2d: Loss Function

How do we know which model is better

1. (Mean squared error, Cross entropy, KL divergence) function to determine the model
2. Our aim is to get the cost function for any particular model as close to zero as possible
3. The model with the lowest cost/loss is the model with the best chance of predicting the test data accurately

### 1.2.2e: Learning Algorithm

How do we identify the parameters of the model

1. Plug in values of Theta into the cost function and see which produces the lowest cost (Brute force computation)
2. Perform it iteratively till we get the values of Theta that yield the minimum cost
3. Brute force computation will prove to be expensive and time consuming for larger ranges of Theta in real world scenarios
4. (Gradient Descent ++, Adagrad, RMSProp, Adam) are optimization algorithms that we can use
5. Backpropogation for Deep NNs and BPTT for RNNs

### 1.2.2d: Evaluation

How do we compute a score for our ML model

1. Use numerical values to assess a model’s performance on training and test data
2. Accuracy: Number of correct predictions/Total number of predictions
3. Top 3 Accuracy: Essentially a best of 3 accuracy,
   1. Number of correct predictions in top 3/ Total number of predictions
4. Why can’t we just use our cost function score to determine model peformance? Simply put, the cost function score is not an easily interpretable metric of our model’s correctness, it is more for the foundational level. Other numerical metrics like accuracy, precision and recall are numbers which directly give us metrics of the model’s performance in terms of number of correct predictions etc
5. Precision = True positives/ True Positives + False Positives.
6. Recall = True Positives/ True Positives + False Negatives.
7. Common numerical metrics are (Top-k accuracy, prediction, recall, F1 score etc)

# 1.3: Six Jars Summary

## 1.3.1: Part 1

Concepts of importance

1. Loss Function: hinge loss, max-margin, lasso, square error, kl divergence, cross validation
2. Model: LSTM, GRU, RNN, CNN, FFNN, MP Neuron, Sigmoid Neuron, Perceptron, AlexNet, ZF Net
3. Data: Find open source datasets, from amazon, google, own organization etc
4. Tasks: Focus on supervised learning - Classification, Regression, transliteration, Object detection, character recognition, multiclass classification
5. Learning ALgorithm: Stochastic GD, Backprop, Adagrad, Adam, Nestrov accelerated GD, RMSprop, Momentum based GD
6. Evaluation: Precision, Top-K-Accuracy, Recall, F1 score

Mathematical Concepts

1. Linear Algebra:
   1. Used in model formation. For eg: *f(Wx + b)*
   2. W is mxn matrix, x is nx1 vector
2. Probability:
   1. Likelihood, cross-entropy, KL-divergence, distribution(discrete etc)
3. Calculus:
   1. Learning algorithms are based off of calculus
   2. Taylor Series, maxima, chain-rule, differentiable function, gradient, minima etc

Why is ML successful

1. Standardised Evaluation
   1. IMAGENET has a standardised training set for you to test your model performance
   2. Pascal2 is a standardised dataset for object detection
2. Improvised Learning/Loss function
   1. Largely improvised and standardised over the years
   2. Tensorflow and pytorch
   3. These frameworks have very good solutions for almost all conventional ML problems
3. Democratized Model
   1. People have openly published their models
   2. Lot of community strength in democratization of model information
   3. We know exactly what equations go into these functions
4. Abundance of Data
   1. Abundance of data present

Connecting to the capstone

1. Data
   1. Two sets of training data
   2. Signboard with text and the exact bounding box around the text
   3. Text in Hindi and the transliteration of the test
2. Task
   1. Binary classification to see if image has text or no text
   2. Character-recognition/Multiclass-classification to identify individual characters
   3. Object detection (regression) finding the bounding box
   4. Transliteration (Classification, Regression and a bit of Generation)
3. Model
   1. Sigmoid model
   2. Deep Neural Networks
   3. Recurrent Neural Networks
   4. Convolution Neural Networks
   5. Class of models combining the above models.
4. Loss
   1. Mean square error
   2. Cross entropy loss
5. Learning
   1. Various gradient descent
6. Evaluation
   1. Accuracy, precision, recall, F1

## 1.3.2: Part 2

How to distribute your work through the six jars

1. In most cases, your job:
   1. *Data and task identification*
   2. For eg: Helmet detection of motorists for identification of violations, locate number plate and identify characters of the number plate.
   3. For eg: Analysing crop leaves to identify if the plant is diseased and classify which disease.
2. Mix & Match:
   1. *Model, Loss, Learning, Evaluation*
   2. These are pre-engineered tools, we should learn how to effectively apply them to accomplish our tasks
   3. They can be tweaked to yield results in our favour, this is called Hyperparameter Tuning.