**Artificial Intelligence & Machine Learning**

-By Shubham Krishna

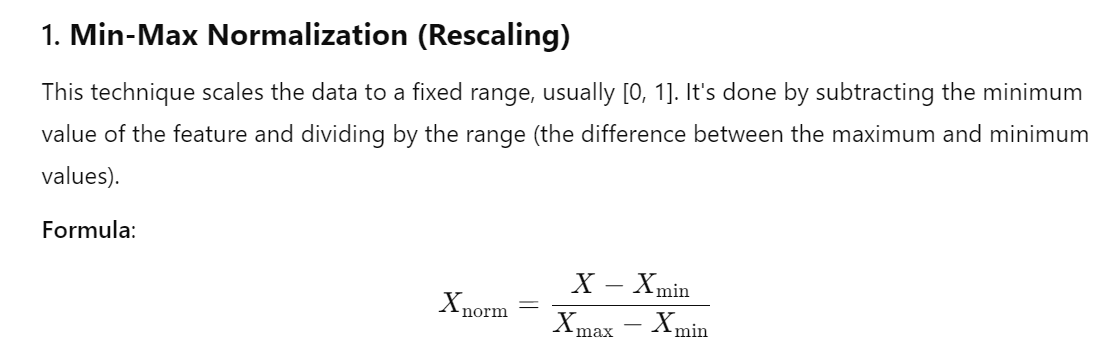
# **1. Machine Learning**

## 1.1 Data Preparation

Prepare the data so that ml algorithms can be used effectively.

Normalization

* Min max normalization

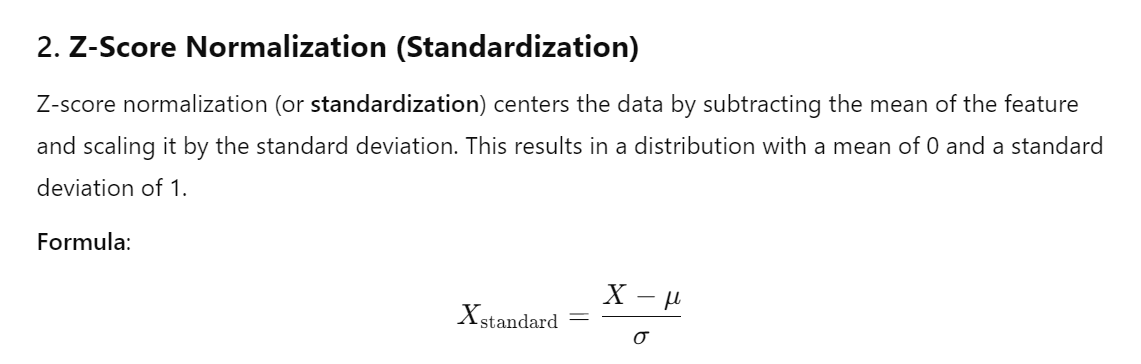


Pros:

* It's simple to apply.
* It works well if the data has a known, fixed range.

Cons:

* Sensitive to outliers, as extreme values can affect the scaling.
* Z-score normalization



Pros:

* Not sensitive to outliers as much as Min-Max normalization.
* Makes the data follow a Gaussian distribution with a mean of 0 and a standard deviation of 1.

Cons:

* If the data is heavily skewed or contains outliers, it may still affect the normalization.
* Log transformation
* Robust scaling

Data Sampling

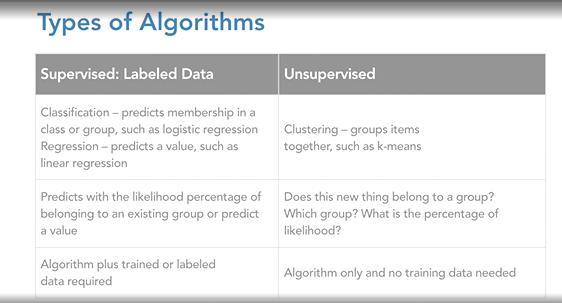
* Random sampling with/without replacement
* Stratified sampling

Dimensionality reduction

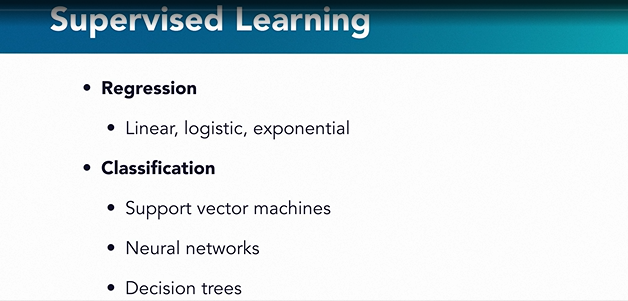
Reduce the number of features in a dataset prior to data modeling.

* Feature selection
* Feature extraction using mathematical functions

## 1.2 Data Modeling and Evaluation

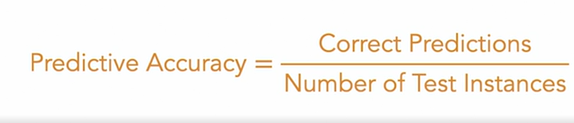


Supervised machine learning



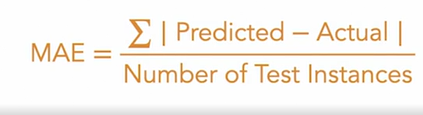
* Classification model for categorical problems.

Evaluation methods- Predictive accuracy



* Logistic regression- yes/no problem
* Decision tree- classification problem. Eg. Customer will or will not pay the loan.
* Regression model for continuous dependent variables.

Evaluation methods- Mean absolute error

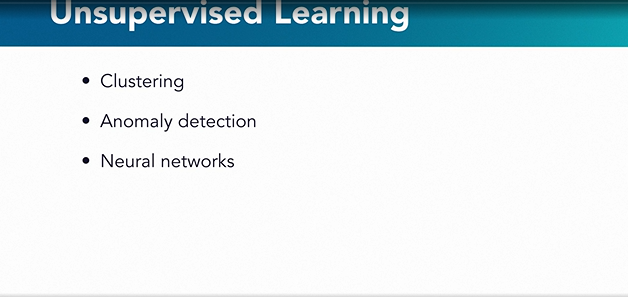


* Linear regression- used to predict the outcomes if there is linear correlation between dependent and independent variables.

Eg. Predict the number of bike rentals based on temp, humidity and windspeed.

* Random forest
* Gradient boosting technique

Unsupervised learning



Used when the data lacks labels. Used for grouping/clustering where data in specific clusters have similarity and different from items in other clusters.

* K-Means Clustering used for customer segmentation, document labeling, anomaly detection.
* Absolute rule – if/else statements.

# **2. Deep Learning**

Deep learning is complex and iterative process.

It starts with random initialization of model parameters and then works towards right values by trial and error.

Applications of deep learning:

* Natural language processing
* Speech recognition and synthesis
* Image recognition
* Self driving cars
* Customer experience, healthcare and robotics.

Linear Regression:

y(dependent variable)= ax(independent variable)+b

a = slope

b = intercept

Logistic Regression:

A binary model which determines the relationship between 2 or more variables. Output is 0 or 1.

y = f(ax+b)d

a = slope

b = intercept

f = activation function

Perceptron

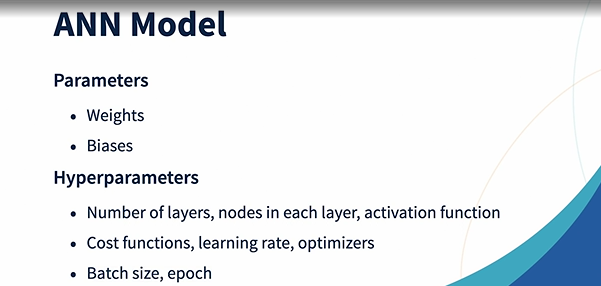
It is a single node in neural network. Multiple inputs are fed and then it does computation and returns a Boolean value.

It is build based on logistic regression.

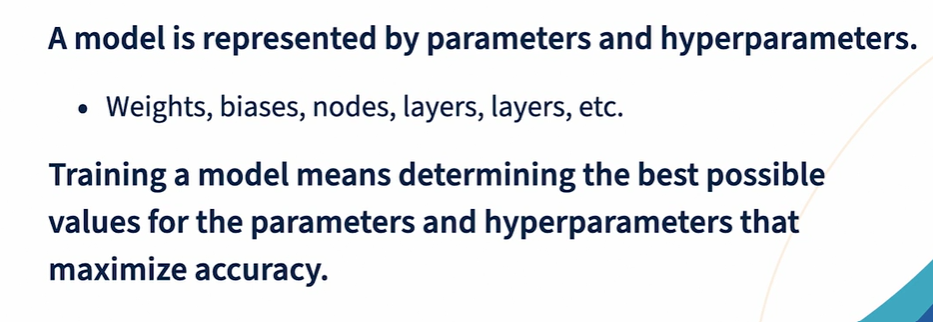
y = f(W1X1+W2X2..+WnXn+b); weights and biases.

## 2.1 Aritificial Neural Networks

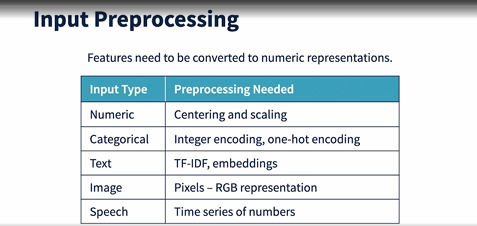
A network of perceptrons modeled after the human brain.



Training an ANN:

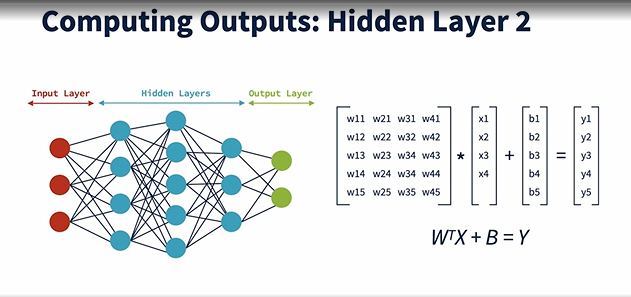


### 2.1.1. Input processing



* The primary goal of StandardScaler is to **transform** the data so that the features have a **mean of 0** and a **standard deviation of 1**.
* This ensures that each feature is on a comparable scale, which is important for many machine learning algorithms that rely on the magnitude of the input data (e.g., gradient-based methods like Logistic Regression, SVM, and neural networks).
* **One-hot encoding** is a technique used to convert categorical data into a binary format.
* This is particularly useful when working with machine learning algorithms that require numerical input.
* One-hot encoding transforms categorical variables (e.g., colors, categories, or labels) into vectors where each category is represented as a binary vector with a single high (1) value at the index corresponding to the category, and all other positions are 0.
* One-hot encoding turns each categorical value into a vector with one 1 and the rest as 0s.
* It helps machine learning algorithms interpret categorical data.
* Commonly used in classification tasks, especially when dealing with non-numeric data.

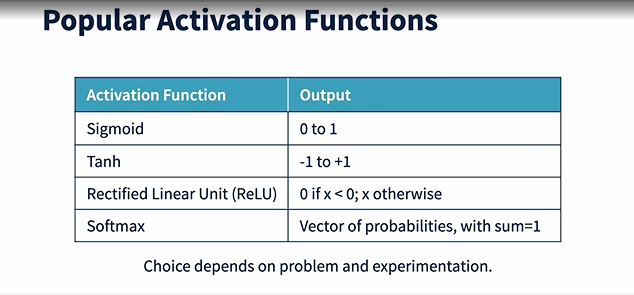
### 2.1.2. Hidden layer processing



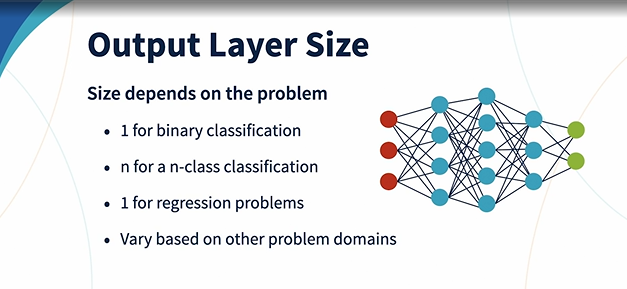
### 2.1.3. Activation function

- Binary step function:

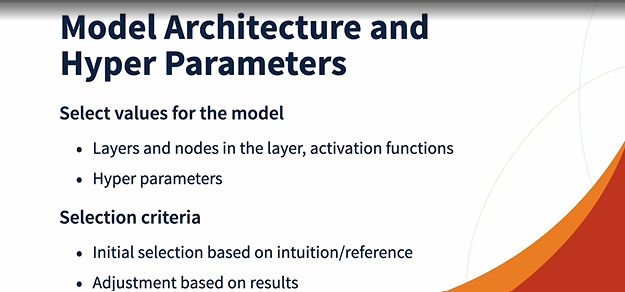
- Outputs are 0 or 1.



### 2.1.4. Output layer



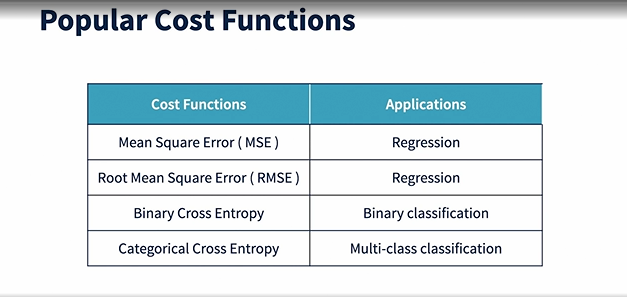
### 2.1.5. Model setup and installation



### 2.1.6. Measuring Accuracy and error

Lost and cost functions

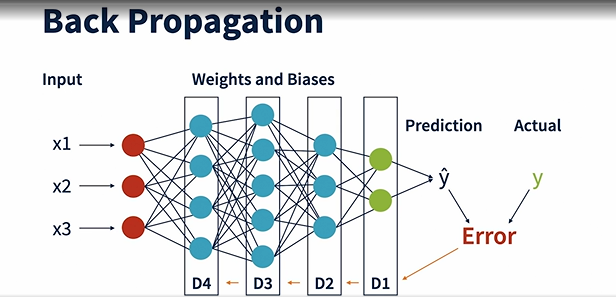
Popular cost functions:



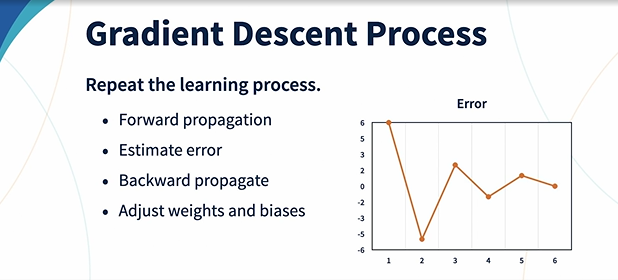
### 2.1.7. Forward and Back Propagation

Forward propagation is used to determine the error in model.

Backward propagation is used to tune the weight and biases to reduce the error in models.

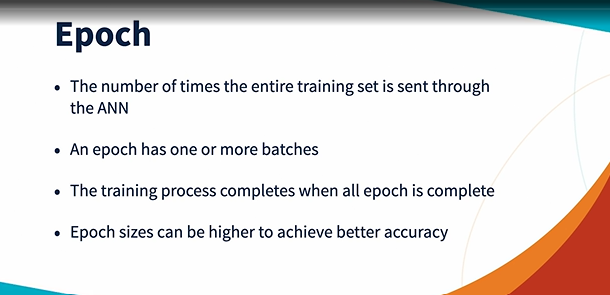


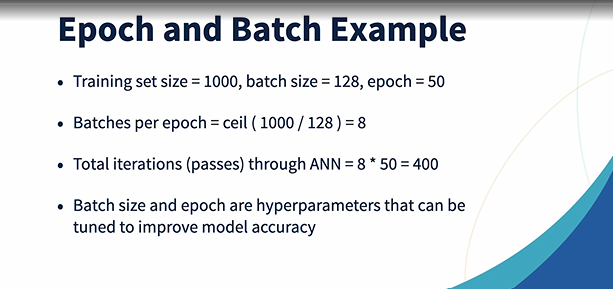
### 2.1.8. Gradient descent



### 2.1.9. Epochs

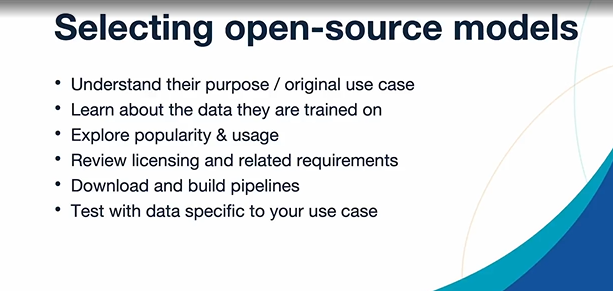
It is total number of times entire training set is passed through the ANN. When all the epochs are completed, training is completed.





Popular network architectures:

* AlexNet
* ResNet
* LeNet5
* VGG
* LSTM
* Transformers



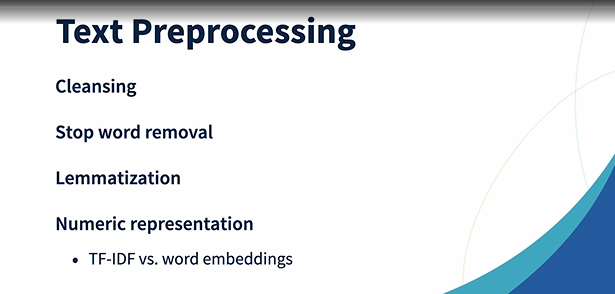
### 2.1.10. Multiclass classification:

Below code shows how to create a deep learning model to predict the species of flower:

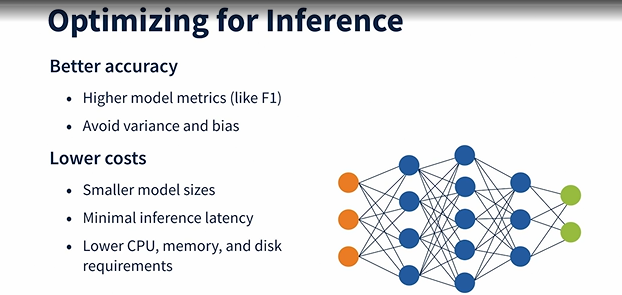


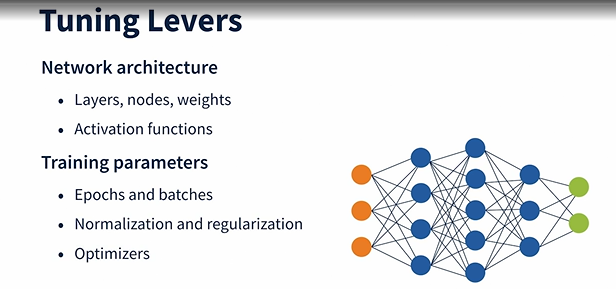
### 2.1.11. Binary classification

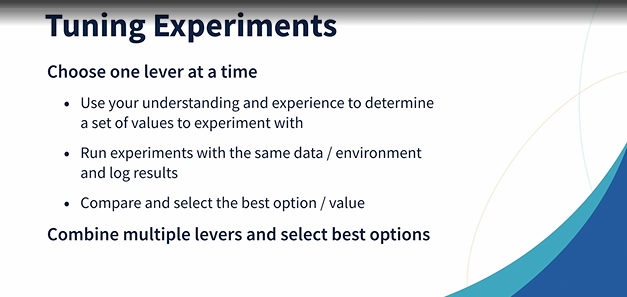
Spam filtering project



## 2.2 Model Optimization and Tuning

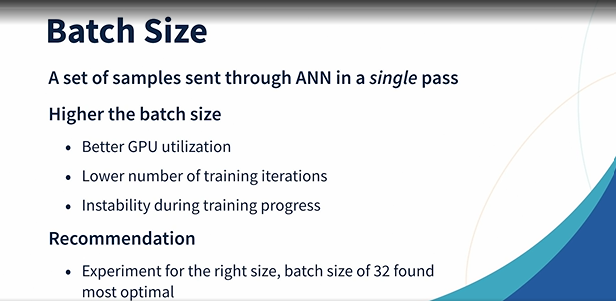




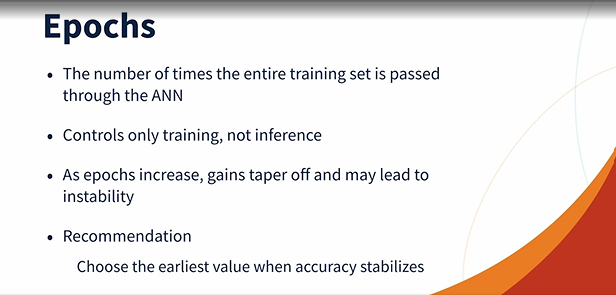


### 2.2.1. Batches and Epoch training

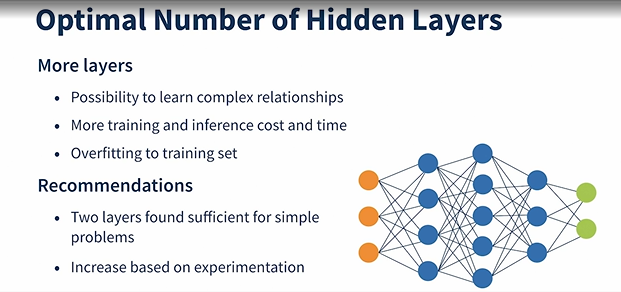
Batches:



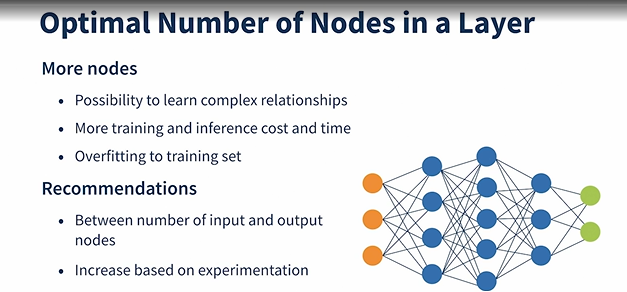
Epochs:



### 2.2.2. Hidden layer tuning

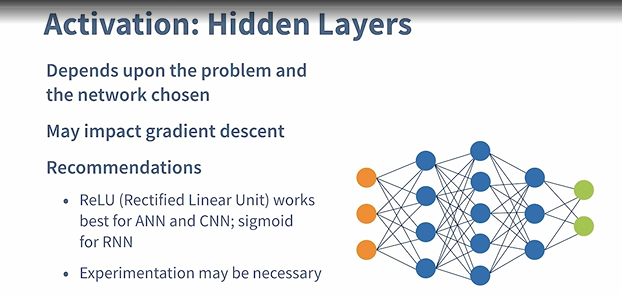


### 2.2.3. Nodes in a layer tuning

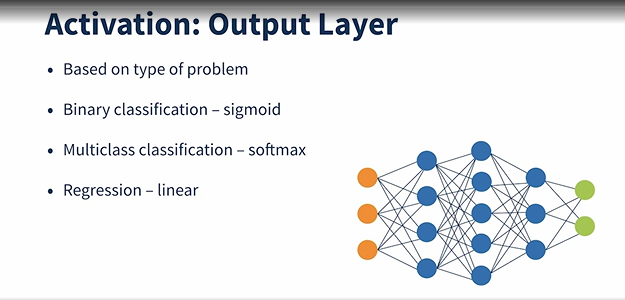


### 2.2.4. Activation function tuning

Activation function- Hidden layer:



Activation function- Output layer:

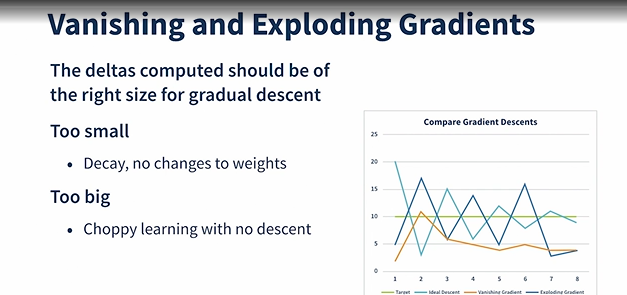


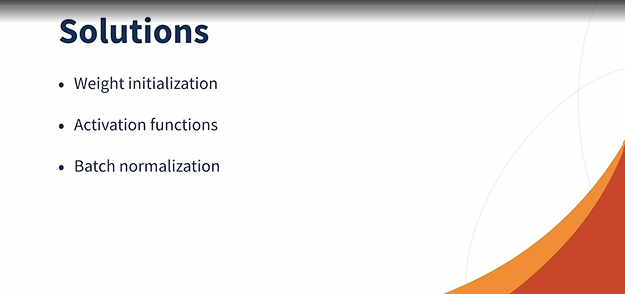
### 2.2.5. Weights initialization

Random normal works best for most cases.

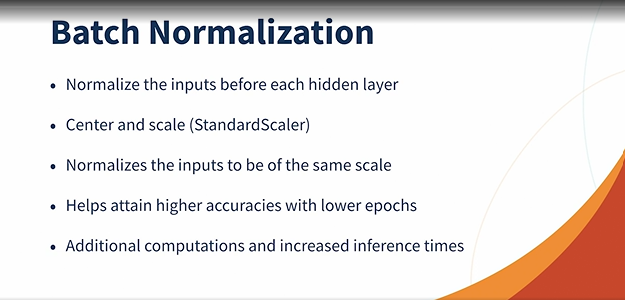


### 2.2.6. Vanishing and Exploding gradients

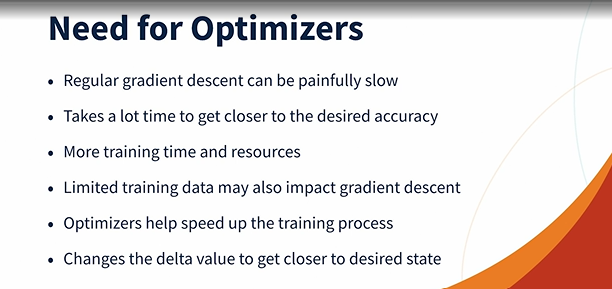


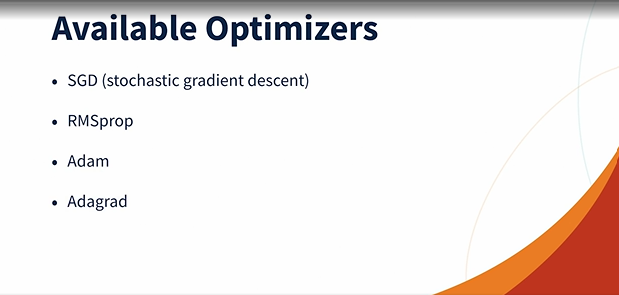


### 2.2.7. Batch normalization

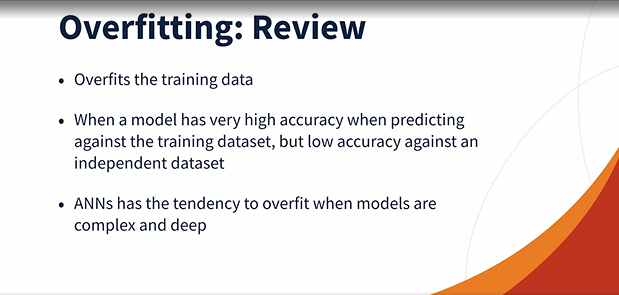


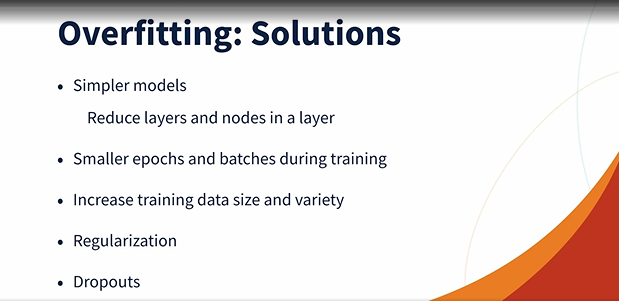
### 2.2.8. Optimizers



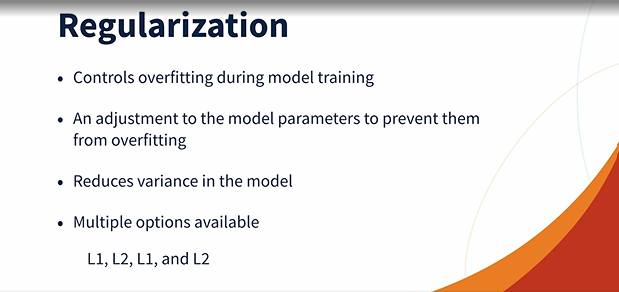


### 2.2.9. Overfitting

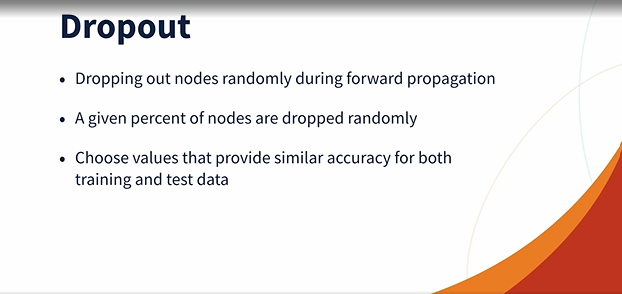




### 2.2.10. Regularization



### 2.2.11. Dropouts



### 2.2.12. Possible optimization and Tuning techniques

1. Number of layers- 1-5

model\_config = {

2."HIDDEN\_NODES" : [8/16/32/40/64],

"HIDDEN\_ACTIVATION" : "relu/sigmoid/tanh",

"OUTPUT\_NODES" : 3,

"OUTPUT\_ACTIVATION" : "softmax",

"WEIGHTS\_INITIALIZER" : "'random\_normal'/'zeros'/'ones'/"random\_uniform"",

"BIAS\_INITIALIZER" : "zeros",

"NORMALIZATION" : "none/batch",

"OPTIMIZER" : "'sgd'/'rmsprop'/'adam'/'adagrad'",

"LEARNING\_RATE" : 0.001/ 0.005/0.01/0.1/0.5,

"REGULARIZER" : None/'l1'/'l2'/'l1\_l2'

"DROPOUT\_RATE" : 0.0/0.1/0.2/0.5

"EPOCHS" : 10,

"BATCH\_SIZE" : 16/32/64/80/96/112/128,

"VALIDATION\_SPLIT" : 0.2,

"VERBOSE" : 0,

"LOSS\_FUNCTION" : "categorical\_crossentropy",

"METRICS" : ["accuracy"]

}

# **3. Reinforcement Learning**

### 3.1. Commonly used terms

Agents: Reinforcement learning agents observe and explore the environment to learn.

State: Position of agent at any given period.

Environment: Agents learning area.

Action: Agents choice of activity in a state. If the action is correct, agent gets positive reward.

Reward: Prize agents get for correct or wrong actions. Correct actions leads to positive reward. Wrong actions leads to negative reward.

Policy: Strategy to take best possible action in a state.

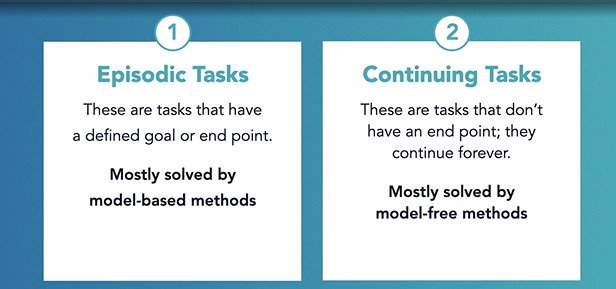
Goal: Agents mission.

Discount Factor: Reward function

### 3.2. Recognize reinforcement learning problem

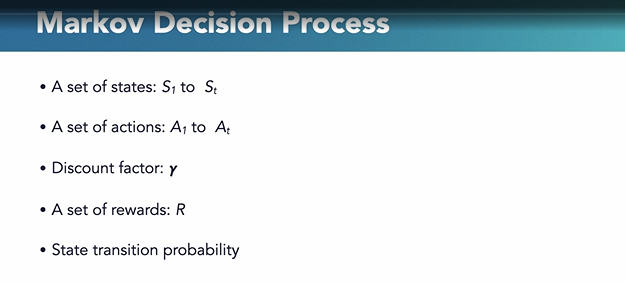
Model free environment: for eg. Relocating to new house, taking any action. Environemnt is unknown.

Model based enviroment: for eg. Getting up from bed and taking new action at home. Enviroment is known.



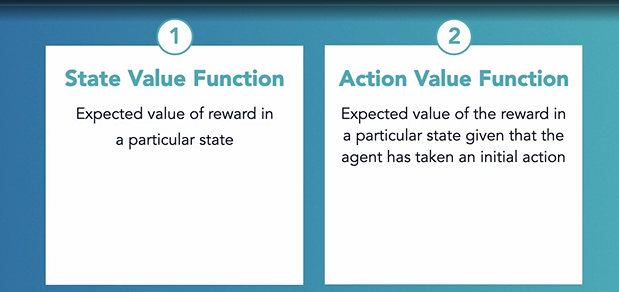
### 3.3 Markov decision process

Way to represent a RL problem mathematically.

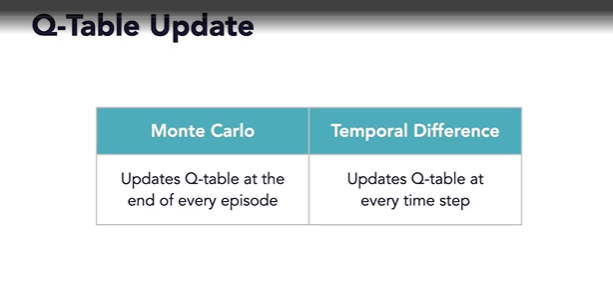


### 3.4. Bellman Equation

It is used to solve the markov decision process.

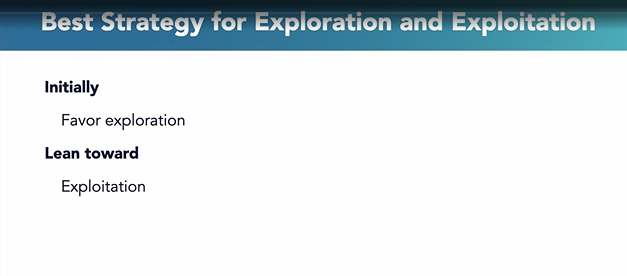


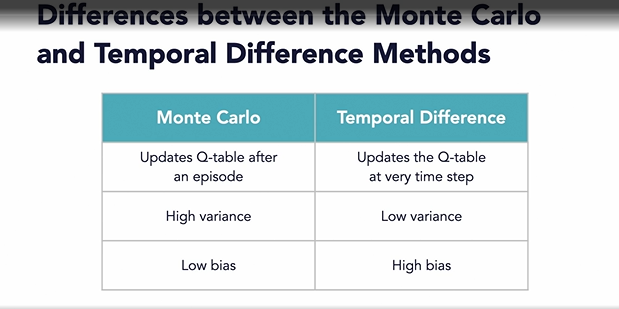
### 3.5. RL Algorithms



### 3.6. Exploration and Exploitation







SARSA:

State->Action->Reward->State->Action