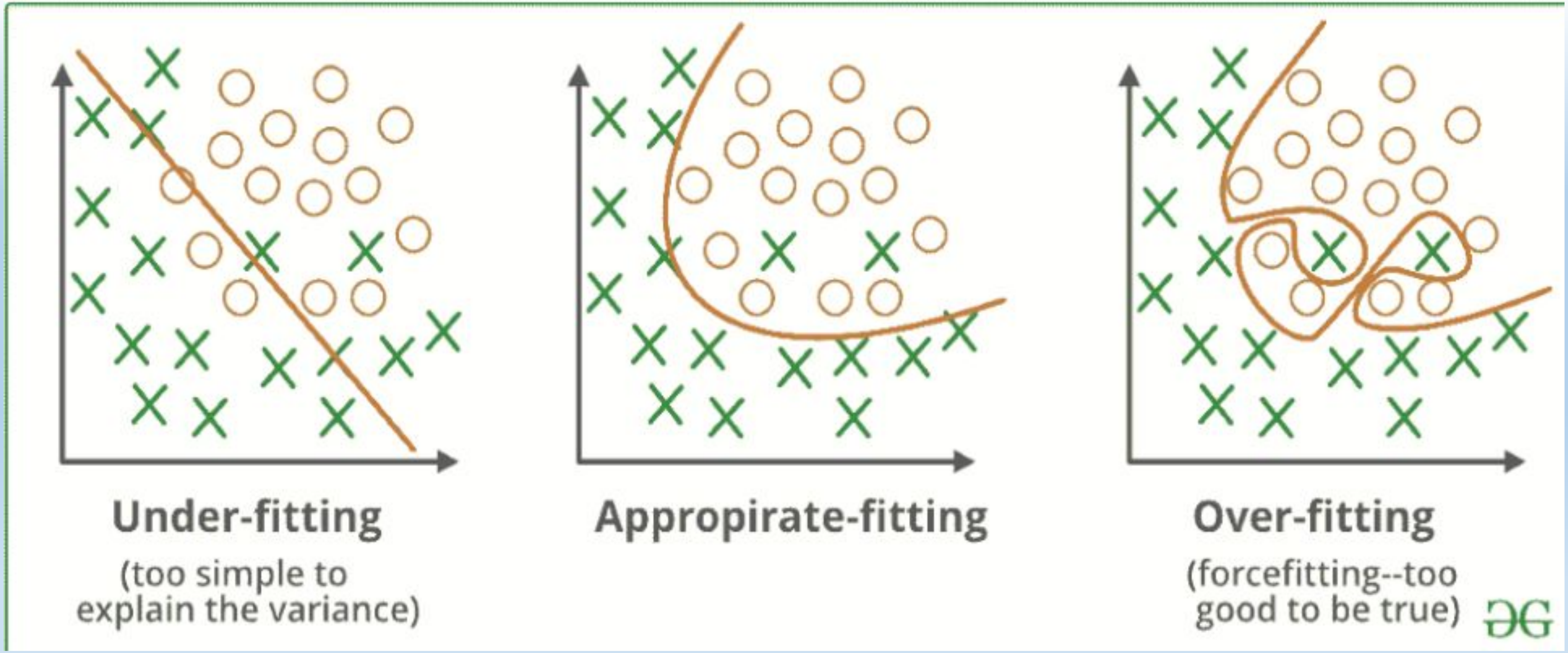
Deep Learning

Regularization

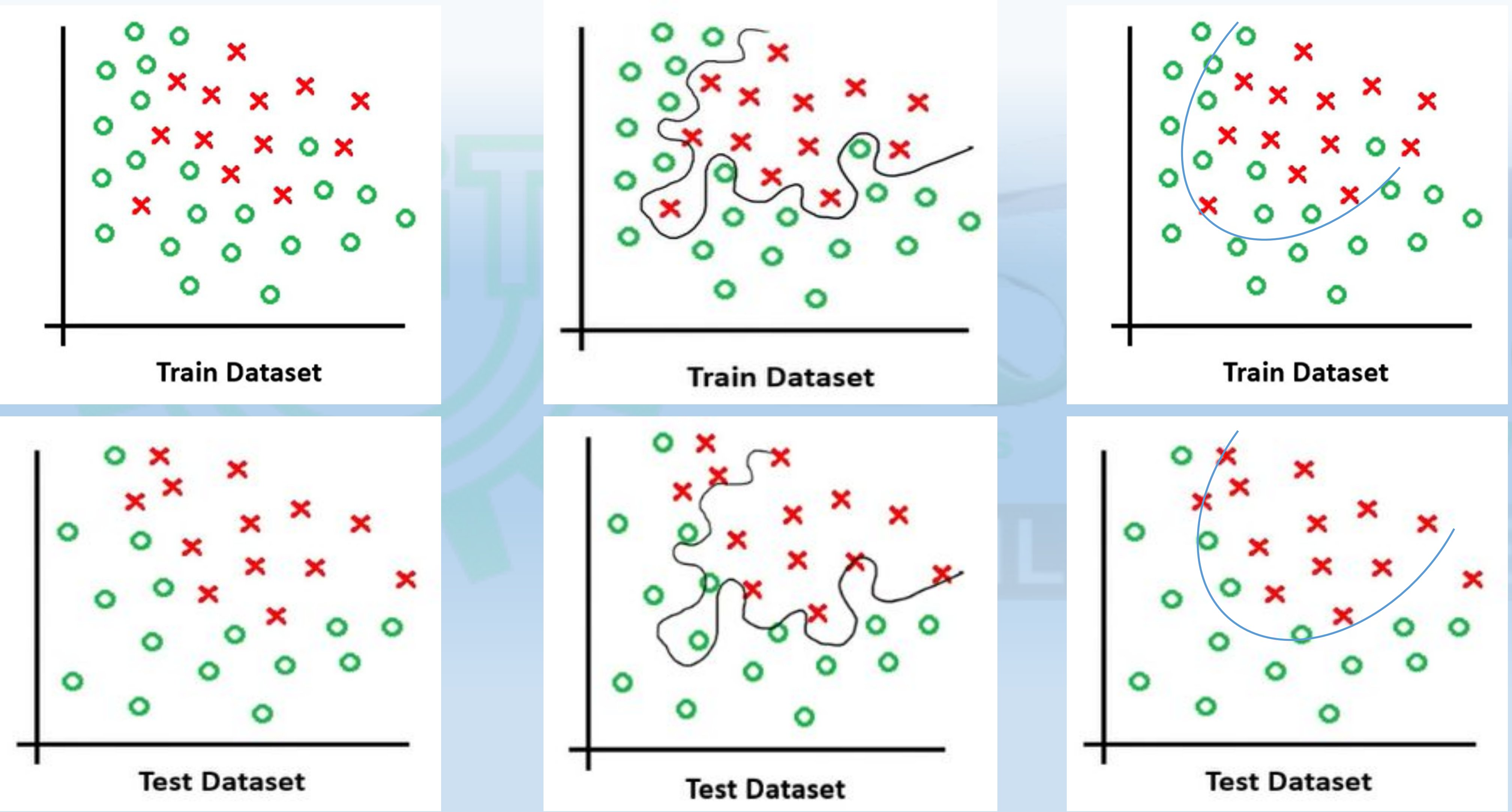
Gananath Bhuyan
Assistant Professor
School of Computer Engineering
KIIT DU, Bhubaneswar

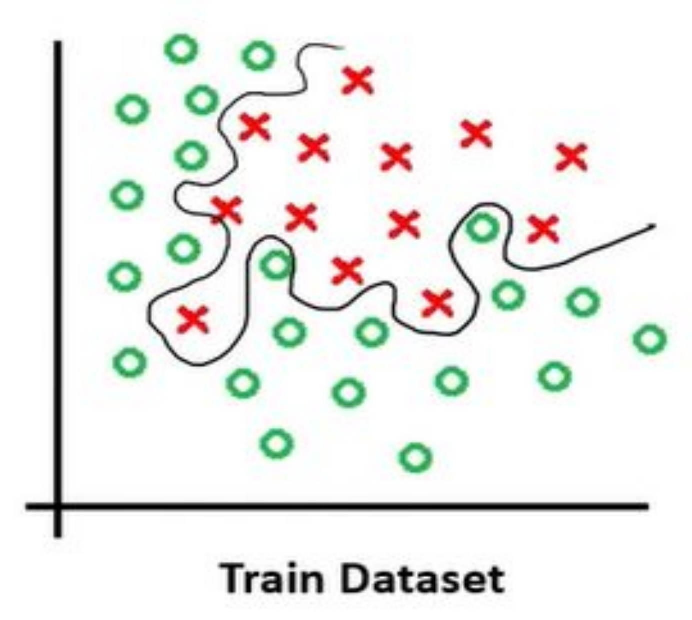
Overfitting vs Underfitting

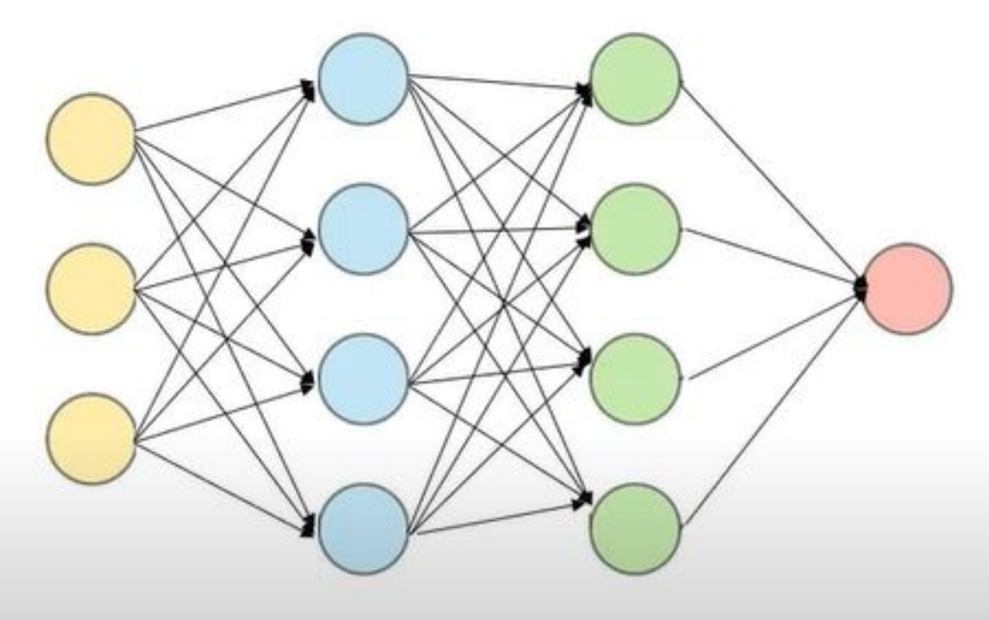


Overfitting vs Underfitting

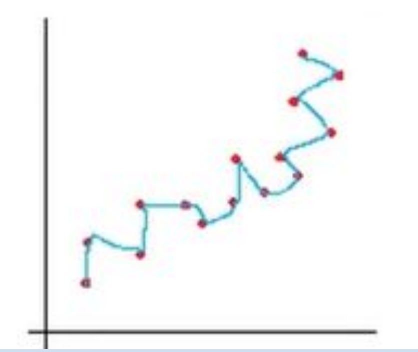
- Overfitting happens when a model learns the training data too well including the noise and random details. This makes the model to perform poorly on new, unseen data because it memorizes the training data instead of understanding the general patterns.
- Eg: Only last week’s weather is used to predict tomorrow’s.
- Underfitting is the opposite problem which happens when the model is too simple to learn even the basic patterns in the data. An underfitted model performs poorly on both training and new data. To fix this we need to make the model more complex or add more features.
- Eg: only the average temperature of the year to predict tomorrow’s weather


Train Dataset
Train Dataset
Train Dataset
Test Dataset
Test Dataset
Test Dataset

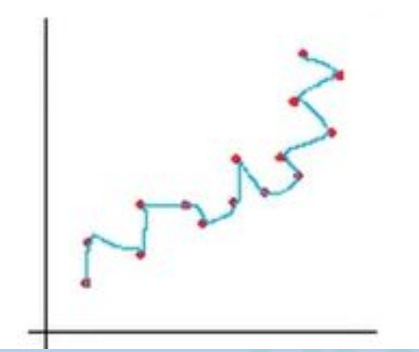




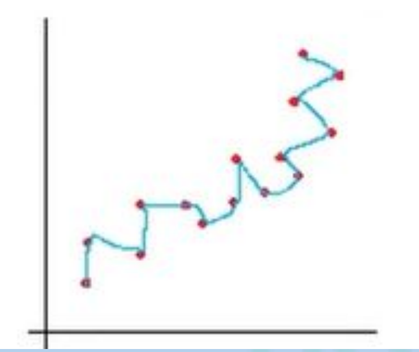
Too many neurons compared to the dataset



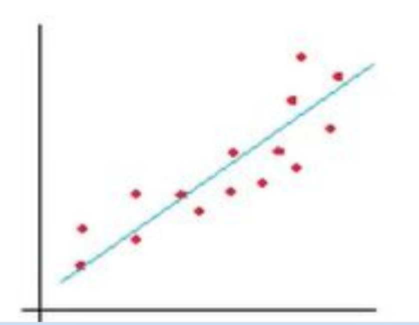
$$Z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4 + \theta_5 X_5 + \ldots.. \theta_{500} X_{500}$$



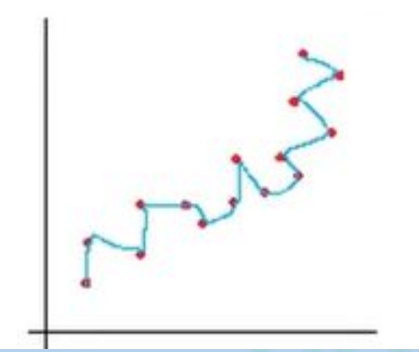
$$Z = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4 + \theta_5 X_5 + \ldots\ldots \theta_{500} X_{500}$$



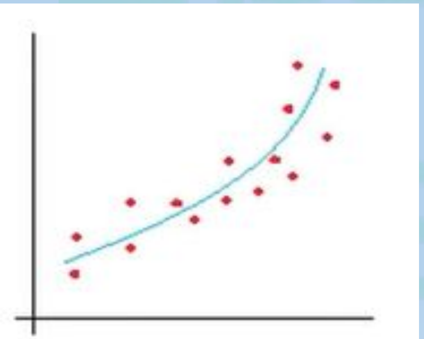
$$Z = \theta_0 + \theta_1X_1 + \theta_2X_2 + \theta_3X_3 + \theta_4X_4 + \theta_5X_5 + \ldots\ldots \theta_{500}X_{500}$$



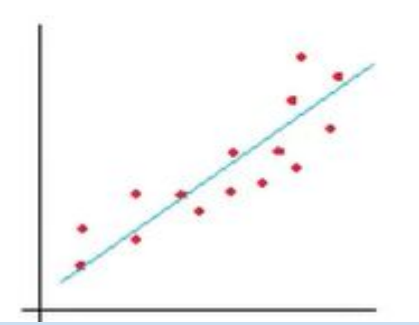
$$Z = \theta_0 + \theta_1X_1$$



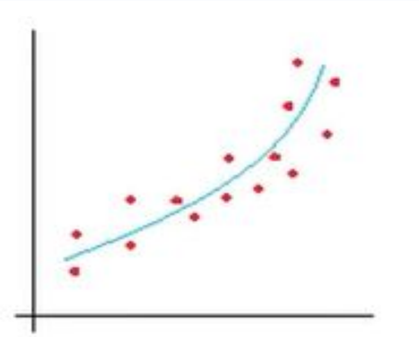
$$Z = \theta_0 + \theta_1X_1 + \theta_2X_2 + \theta_3X_3 + \theta_4X_4 + \theta_5X_5 + \ldots\ldots \theta_{500}X_{500}$$



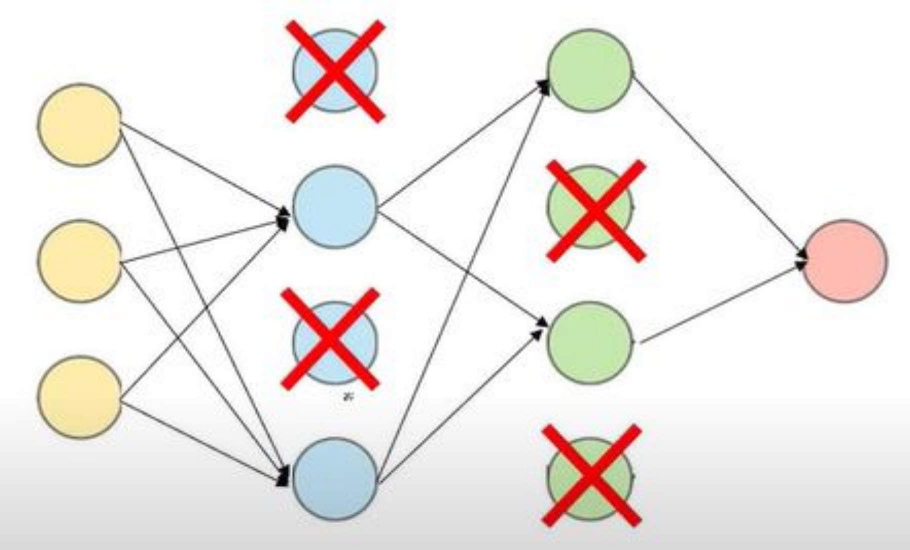
$$Z = \theta_0 + \cancel{\theta_1X_1} + \theta_2X_2 + \cancel{\theta_3X_3} + \theta_4X_4 + \cancel{\theta_5X_5} + \ldots\ldots \theta_{500}X_{500}$$



$$Z = \theta_0 + \theta_1X_1$$



$$Z = \theta_0 + \cancel{\theta_1 X_1} + \theta_2 X_2 + \cancel{\theta_3 X_3} + \theta_4 X_4 + \cancel{\theta_5 X_5} + \ldots\ldots \theta_{500} X_{500}$$



Regularization

- Regularization is a technique used in ML/DL to prevent overfitting by nullifying the effect of a few neurons to produce a smooth curve.
- It is done by adding a penalty term to the model's loss function, thus discouraging overly complex models

- L1 Regularization (Lasso)(Least Absolute Shrinkage and Selection Operator)
- L2 Regularization (Ridge)
- Dropouts

L1 and L2 Regularization

$$Cost = \frac{1}{n}\sum_{i=1}^{n} Loss_i + \frac{\lambda}{2}\sum_{j=1}^{m} \mid W_j \mid \quad \cdots\cdots\cdots\cdots (L_1)$$

$$Cost = \frac{1}{n}\sum_{i=1}^{n} Loss_i + \frac{\lambda}{2}\sum_{j=1}^{m} \mid W_j \mid^2 \quad \cdots\cdots\cdots\cdots (L_2)$$

n : no of data points

m : no of features

$Loss_i = (y_i - y_i')^2$

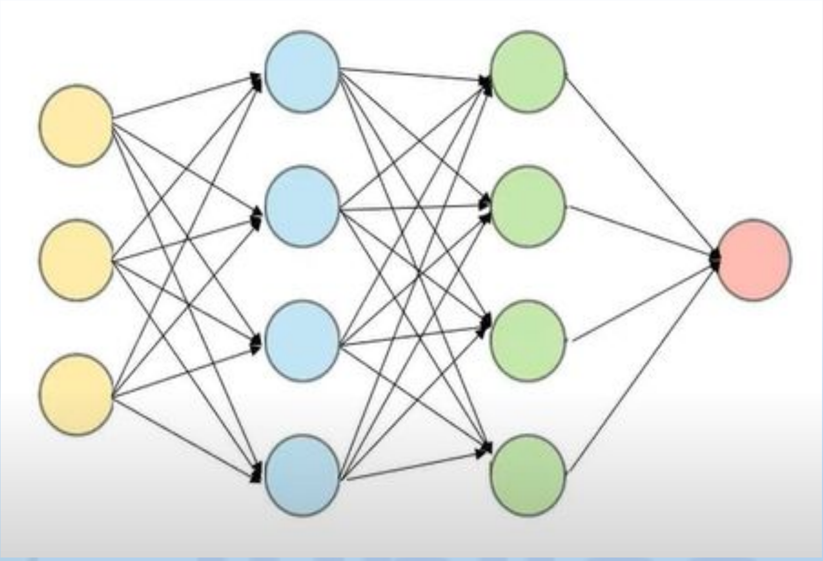
W_j : weight of the j^{th} feature

λ : Tuner or Regularization parameter

L1 and L2 Regularization

$$Cost = \frac{1}{n}\sum_{i=1}^{n} Loss_i + \frac{\lambda}{2}\sum_{j=1}^{m} \mid W_j \mid^2$$

$$Cost = \frac{1}{n}\sum_{i=1}^{n} Loss_i + \frac{\lambda}{2}\sum_{l=1}^{L}\sum_{i=1}\sum_{j=1} \mid W_{ij}^l \mid^2$$



$$\sum \mid W^l \mid^2 = (W_{11}^l)^2 + (W_{12}^l)^2 + (W_{13}^l)^2 \cdots + (W_{21}^l)^2 + (W_{22}^l)^2 + \cdot\cdot + \cdot\cdot + (W_{ij}^l)^2 + \cdot\cdot$$

- Backpropagation:

$$dL = dZ_i \cdot A_i^T + \frac{\lambda}{?} W^i$$