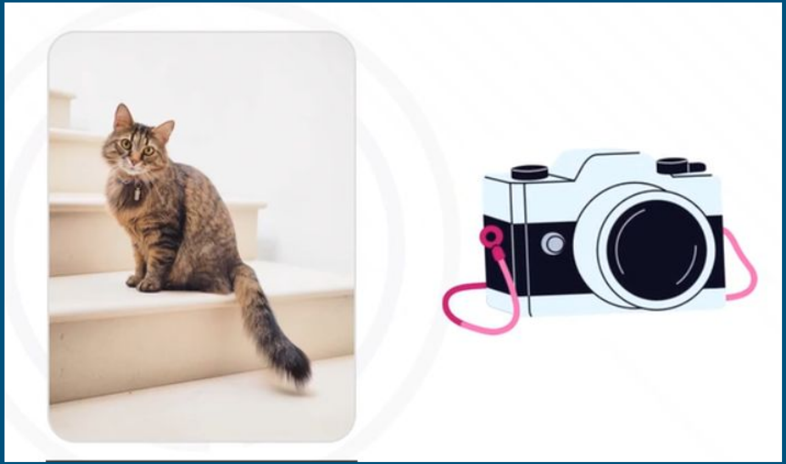
IMAGE PROCESSING

WEEK-1

4. ML to classify an image

- **What is image classification :**

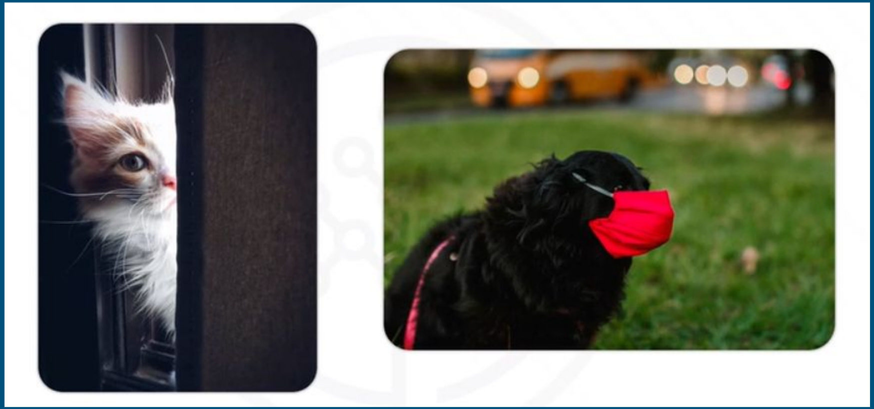
- Image classification is the process of taking an image or picture and getting a computer to automatically classify it ,or try providing the probability of the class of the image.
- **Use cases :**
    - Organize the photos on phones gallery by friends,family,etc.
    - In radiology it is used to find anomalies in x-rays.
    - In self-driving cars to identify object around us.
- Challenges:
    - Change in viewport.



- Change in illumination .

- Change in deformation .

- Change in occlusion .



- Background clutter .

- **Image classification with KNN :**

- KNN :
    - KNN stands for k-nearest neighbours.
    - It is the simplest classification algorithm
    - It classifies the unknown element by most common class and find the nearest points to the class.
    - It calculates the common points by calculating the euclidean distance between 2 points.
- Working :
    - It works by calculating the distance between each point present in the classes and the unknown element
    - The unknown element gets the label of the class with which it had the smallest distance.

- KNN is not used for multiclass classification , knn works slows while dealing with multiple classes.

- **Training and testing set:**
- Separation of the dataset into training and testing set is an important part of model evaluation.
- **Out of 100% of dataset :**
    - 70%,60%,or 80% of the data is used as training set to train the model
    - 20%,30% or 40% of the data is used as testing set to test the model on new data.

- **Accuracy :**
- Accuracy is determine how good the model works.
- Accuracy is determined by calculating the number of times model predicted data correctly.
- 1- dogs in the dataset
- 0- cats in the dataset
- Y - actual class
- y^ - predicted class

Samples: n	1	2	3	4
y	1	0	1	0
$\hat{y}$	1	0	0	1
Correct	1	1	0	0

• Feature extraction techniques:

• HOG -> Histogram of oriented gradients:

- Captures edges directions and object shape
- HOG captures
    - Shapes
    - Human silhouettes
    - Object boundaries
- Working:
    - Compute pixel gradients
    - Divide image into small cells
    - Build histograms of edge directions in each cell
    - Normalize them in blocks
    - Combine all histogram
- Widely used in pedestrian detection

- **SIFT- Scale invariant feature transform**

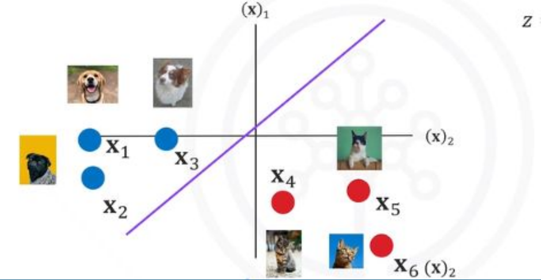
- It is used for :
    - Extracts key points that are stable under : scale changes, rotation , illumination changes.
- **Working :**
    - Detect important points
    - Compute orientation around each point
    - Create a 128-d descriptor
    - Use these descriptors for matching or classification
- **benefits:**
    - Works well even when the images are resized or rotated.

- **SURF - Speeded up robust features**
  - It is an faster version of SIFT
  - **Working :**
    - Uses box filters and integral images for speed
    - Still scale and rotation invariant
    - Produces describes like SIFT but faster
  - **Use cases:**
    - Object matching ,tracking

- **LBP - Local Binary Patterns:**
  - Captures textures information
  - **Working :**
    - Compare it with its 8 neighbours.
    - If neighbor>=center ->then “1”,else”0”
    - From an 8-bit binary number
    - Convert to decimal , and build histogram of all LBP values
  - **Use case:**
    - Face recognition , texture analysis

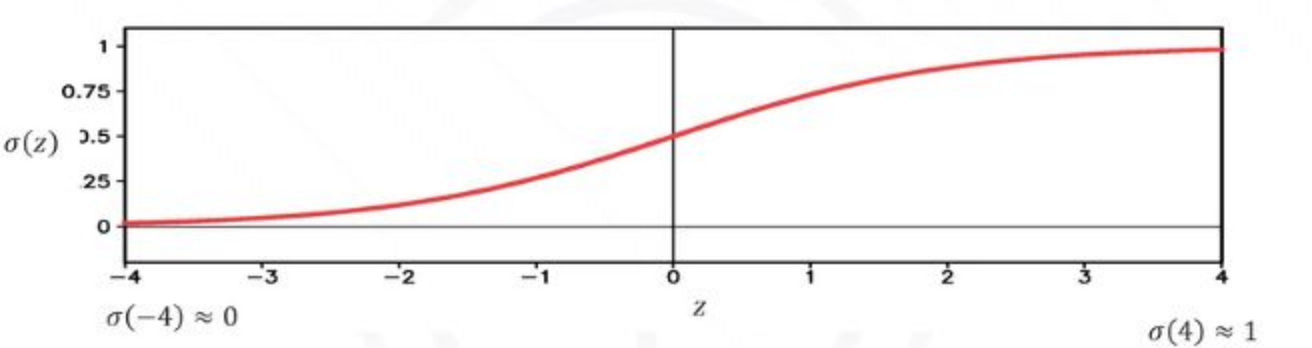
- Linear classifiers :

- Linear classifier is the most simplest and widely used models in machine learning.
- It is used for predicting the class of the data point by computing the weighted sum of its features and applying decision rules
- It makes decision as : if f(x)>0 ->class 1 or else class 0
- Linear classifiers works by separating the classes by using a straight line.
- Advantage :
    - Fast and easy to train
    - Works well with high dimensional data
    - Give great accuracy when the data is linearly separable.
- Disadvantage :
    - Cannot model complex nonlinear patterns
    - Performs poorly if classes overlaps in nonlinearly



- Logistic regression :
  - Logistic regression is type of linear classifier that uses sigmoid function to convert the output to probability.
  - Logistic functions works best for binary classification.
  - Sigmoid function : $\sigma(z) = \frac{1}{1 + e^{-z}}$

  - If the value of z is very large negative then the expression is approximately zero.
  - If the value of z is very large positive then the expression is approximately one.



  - Everything in middle has value between 1 and 0.
  - Can make multiple decision boundary and has to select the best decision boundary for classification

- Loss function:
  - The function determines how good your prediction is .
  - Every time your prediction is correct => 0
  - Every time your prediction is incorrect => 1

$loss(y,\hat{y})$	y	$\hat{y}$
0	1	1
0	0	0
1	1	0
1	0	1

- Cost function: classification error:
  - Cost function yells how good our learnable parameters are doing on the dataset.
  - The loss increase the cost .
  - Cost is the sum of loss and cost is related to decision boundary the better the decision boundary is cost is going to be 0.

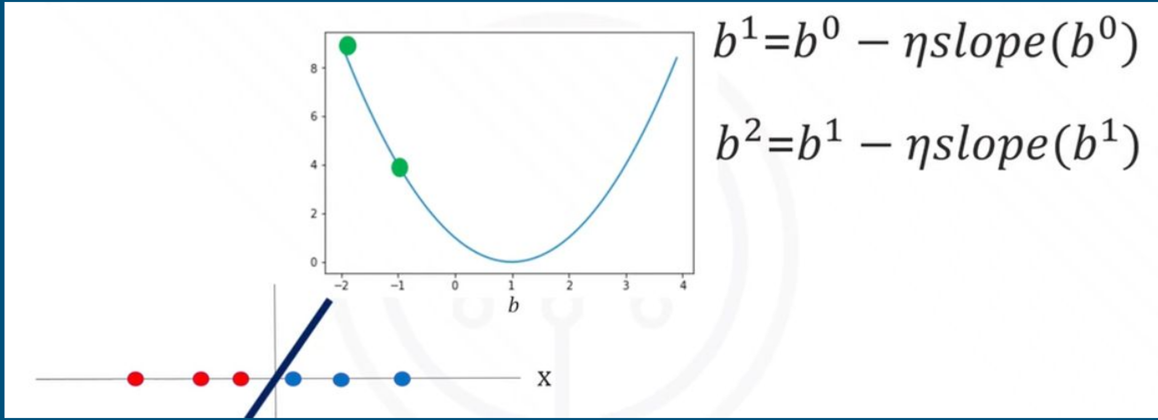
- Cross entropy:
  - Cross-entropy deals with how likely the image is going to be in a specific class.
  - If likelihood of belonging to a particular class incorrect is large then cross entropy will be large.
  - If likelihood of belonging to a particular class correct then cross entropy will be small but not zero.

$$Cost(\mathbf{w}, b) = loss\big(y_1, \underbrace{\sigma(\mathbf{w}\mathbf{x}_1 + \mathrm{b})}_{\text{probability}}\big) + loss\big(y_2, \underbrace{\sigma(\mathbf{w}\mathbf{x}_2 + \mathrm{b})}_{\text{probability}}\big) + ..$$

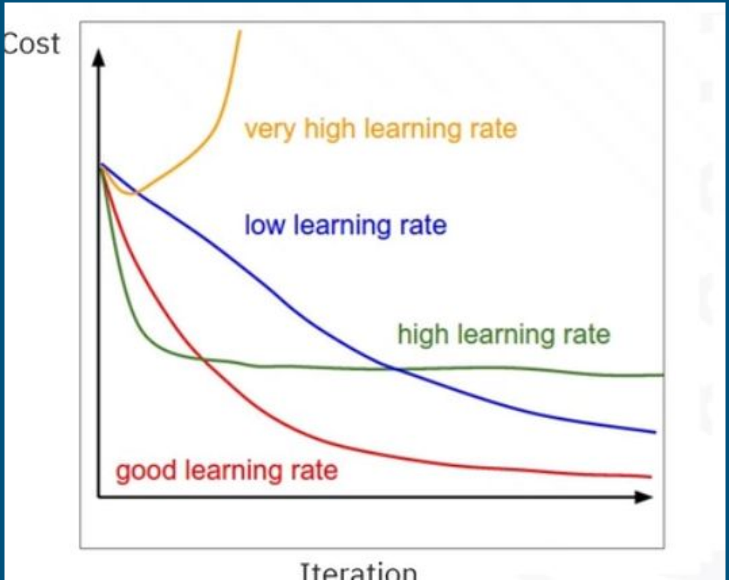
- Gradient descent :
  - Gradient descent is a method to find the minimum of the cost function.
  - Formulae :

$$b^{i} = b^{i-1} - \eta\, slope(b^{i-1})$$

    - b^i = updated parameter at current iteration i.
    - b^i-1 = parameter value of the previous iteration
    - Eta is the learning rate-> a hyper parameter that controls the size of steps taken
    - When gradient becomes zero ,the parameter stops updating.



- Learning curves =>



- Mini-batch gradient descent :
- In mini-batch gradient descent we divide the dataset into small batches
- For each batch we can calculate the loss, compute the gradient and update the model weights.
- When we use all the samples in the dataset we call it as an epoch.
- It’s called batch gradient descent where one iteration is equal to one epoch.
- Epoch :\
    - Epoch means one complete pass of the entire training dataset through model
    - Example - reading a book :
        - One epoch => reading a whole book once
        - Multiple epochs means reading the book many times so the model can learn well.
    - Advantages :
        - Reduces loss
        - Adjust weights better
        - Improve accuracy
- In one epoch we can have multiple batchs

- No of iterations :

$$\text{Iterations} = \frac{\text{Training size}}{\text{Batch size}}$$

- Softmax and multi-class classification :
  - Argmax function :
    - Argmax function returns the index of largest value in sequence of numbers.
  - Softmax is a simple linear classifier used for multiclass classification(more than 2 classes)
  -

$$z_0 = w_0\mathbf{x} + b_0$$

$$z_1 = w_1\mathbf{x} + b_1$$

$$z_2 = w_2\mathbf{x} + b_2$$

- Sometimes the softmax function is not the best option .
- Advantages of softmax :
    - It is simple and fast
    - Works well with multi-class classification
    - Used in CNN final layers
    - Used for text,images classification tasks.
- Disadvantage of softmax :
    - Only learn linear values
    - Sensitive to outliers
    - Poor performance in high-dimensional data
    - Assumes classes are mutually exclusive

Classical ML Classifiers:

- **SVM - support vector machine:**

- Helps in finding best separating hyperplane between classes.
- **Need ?:**
    - Works well with high-dimensional feature like HOG
    - Use kernels to classify non-linear patterns
- **Output:**
    - Binary or multi-class prediction

- **Random forest:**

- It is an ensemble of many decision trees
- Each tree give a vote -> majority decides output
- **Strengths:**
    - Handles noisy data
    - Robust
    - Works well when combined with texture-based features.

- **KNN - k-nearest neighbors:**

- It computes the prediction by :
    - Compute distance to all the training samples
    - Pick k nearest ones
    - Majority label becomes output
- **Pros :**
    - Very simple , no real training phase
- **Cons:**
    - Slow during prediction , sensitive to noisy features

- **Why classical ML fails compared to deep learning:**

- Deep learning(CNNs) has replaced classical ML for several reasons:

1. **Manual feature engineering:**

- Classical ml need - HOG,SIFT,LBP,color features,etc and are designs by humans. But CNNs learn features automatically

2. **Not robust to variations:**

- Traditional ML fails with : different lighting,background clutter,rotations,scaling.CNNs handle these naturally.

3. **Shallow representations:**

- classical ML extracts low-level feature only,CNN extracts hierarchical features: low-level->edges,mid-level->corners,textures,high-level -> object concepts

4. **Cannot handle large datasets:**

- traditional ml is memory expensive. Deep prediction scales perfectly with GPUs and large data.

5. **Lower accuracy :**

- Modern accuracy architectures outperform classical ML by huge margins.

- POC -
- Cat vs Dog classification using HOG+SVM
  - https://colab.research.google.com/drive/1nydnGhFaF9PMRkem3e2B0MPwu5TNDbi3#scrollTo=nTZQfyT-yglN
- Real-time webcam classification
  - https://colab.research.google.com/drive/1CMkeK6cf0ZI4MrG0sm0UtXW_Q2bicQBc