Task 3

- optimization and gradient descent
- What is the difference between overfitting and underfitting?
- precision and recall and the trade of between them
- feature selection
- What do you know about Ensemble learning?
- What is Random Forest and how does it work?
- What do you know about backward propagation?

optimization:

optimization is the process of finding the best solution for a mathematical function given certain constrains.

It refers to minimize the loss function by adjusting model parameters that is weights and biases.

for example, you have a model for predicting house prices, then loss function measures how far predicting prices from actual prices.

and optimization finds parameters that make prediction as accurate as possible.

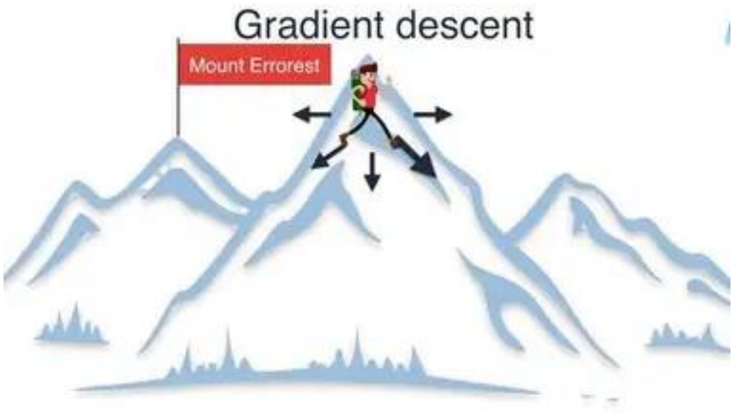
Gradient descent:

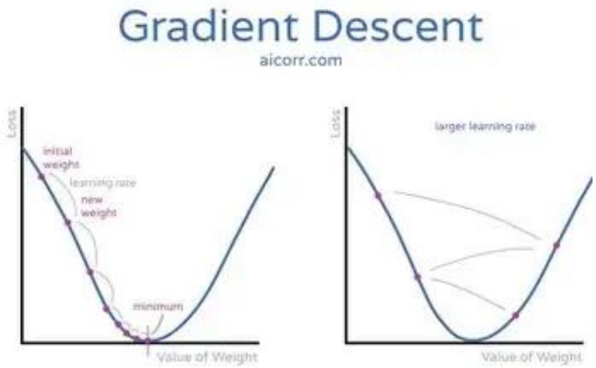
Gradient Descent is an iterative optimization algorithm used to minimize a function by moving in the direction of the steepest descent, as indicated by the negative gradient.

for example, Imagine standing on a hill and You want to get to the lowest point.

At each step, you look around to see which direction is downhill the steepest point and take a step in that direction.

Repeat until you reach the bottom.





What is the difference between overfitting and underfitting?

Overfitting:

This happen when the model is too complex and learns the noise in the training data and it is very sensitive to the training data as it learns the training data very good and it doing best performance on training data, but low performance on test data.

It make her best to memorize the training data, but not to know the generalized patterns in the data.

Underfitting:

This happen when the model is too simple to capture the generalized pattern in the data.

It has poor performance on both training data and test data.

This happen when the model is too simple like liner for too complex data.

precision and recall and the trade of between them:

Percision: “How many selected items are relevant?

Precision = (True positive/True Positive + False Positive)

Example in spam detection, High precision means when we flag an email as spam,

It’s very likely actually spam.

Recall(sensitivity): “How many relevant items are selected?”

Recall=(True positive / True positive + False Negative)

Example in disease screening, high recall means we catch most actual cases.

My way to specify if the problem needs high precision or high Recall is:

1- **Ask your self if it usual to have some of those actions that you don’t need like spam or disease, if it’s ease to say about a not spam email, a spam email, then you want high precision.**

   But if you say yes, it may a doctor say that you have that disease but in actual you don’t, it’s better than that you have a disease and the doctor say that you have a disease.

The Trade-off(Precision-Recall Curve):

Inverse Relationship: As precision increases, recall typically decreases, and vice versa.

- feature selection:

It’s when you selecting the most relevant subset of features for model building.

This is to reduce overfitting and improve model performance and reduces training time and also enhance interpretability, like L1/Lasso, tree importance.

Ensemble Learning:

It’s when you combine multiple simple models to produce a stronger predictive model than any individual model.

AND this to reduce variance, reduce bias, improve generalization for this it gives us more stable predictions.

The types are Bagging:

A parallel training of multiple models on random subsets, thus it reduces variance, like Random Forest.

Boosting:

A sequential training where each model corrects previous errors, thus it reduces bias, like XGBoost and Gradient Boosting.

Stacking:

Combine predictions from multiple models using a meta-model and it leverages strengths of different algorithms.

HERE WE CAN SAY Diversity is crucial that mean that there fault not the same.

Wisdom of the crowd- here we can say if we get the best from each one, then we can say that our decision is correct.

- What is Random Forest and how does it work?

It’s an ensemble learning method that combines multiple decision trees to create a more accurate and stable model. It belongs to the bagging family of ensemble techniques.

How it works?

It create

1-Bootstrap samples as it create multiple datasets by sampling with replacement.

2-Random Subspace: For each tree, select random subset of features at each split

3. Grow Trees: Build decision trees fully (no pruning) on each sample

4.Aggregation: For prediction:

- **Classification: Majority vote**
- **Regression: Average prediction**

- What do you know about backward propagation?

An algorithm for efficiently computing gradients in neural networks using the chain rule of calculus.

The training loop is consists of 3 step:

1- **Forward pass: and it's as you can say like calculating the first output and so on the second and the third till reaching the loss near to zero.**

Y_predicted = w * a + b

A = *σ(Y-predicted)*

2- ***Loss Function: and it's calculating the difference between the desired output and the predicted output.***

L = (Y_actual – Y_predicted)/2 and this for MSE

3- ***Backward Pass using Chain Rule:***

a. ***You calculating the Output layer gradient***

b. ***Then calculating thehidden layer gradients.***

c. ***Then Weight gradiens***

And then updating the weight.