Lecture (6)

⇒ Gradient descent (optimization technique to minimize the loss function and thus fine tune the parameters)

Ex) Linear Regression with (SGD)

$y = B_0 + B_1 X$, assume $B_0 = 0$, $B_1 = 0$

$y = 0 + 0 \Rightarrow y = 0$

① Calculate error $Y = 1$, $y = 1$

⇒ $p(i) - y(i)$

(predicted - expected)

* substitute (X) to find y then $(p(y) - 1)$

⇒ $e = 0 - 1 \Rightarrow e = -1$

② update parameter

⇒ $B_0(t+1) = B_0(t) - \text{alpha} * \text{error}$

$B_0(t+1) = 0 - 0,01 * -1$

$B_0(t+1) = 0,01$

⇒ $B_1(t+1) = B_1(t) - \text{alpha} * \text{error}$

$B_1(t+1) = 0 - 0,01 * -1$

$B_1(t+1) = 0.01$

* Repeat this until convergence

Lecture (7) Logistic Regression

$f = B_0 + B_1 X_1 + B_2 X_2$, $B_0 = 0$, $B_1 = 0$, $B_2 = 0$, $y = 0$

$X_1 = 2,7$, $X_2 = 2,5$

~ sigmoid $\frac{1}{1 + e^{-\text{output}}} = \frac{1}{1 + e^{0}} = \frac{1}{2} = 0,5$

use the update equation ($b = b + \text{alpha} * (y - \text{prediction}) * p * (1-p) * X$

$B_0 = B_0 + \alpha * (0 - 0,5) * 0,5 * (1 - 0,5) * 1$

$B_0 = -0,0375$

⋮

$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{Total predictions}} * 100\%$

Lecture (8) classification

⇒ Binary classifiers
- ⇝ SGD classifier
- ⇝ SVM

⇒ Multiclass classifiers
- ⇝ Random Forest
- ⇝ Naive Bayes

⇒ multi label classification
- ⇝ multiple class for each instance

⇒ We can perform multiclass classification using Binary classifiers
- → One versus All (OvA)
  - ⇒ We get a score from each classifier to decide on the correct prediction
- → One versus One (OvO) (SVM)
  - ⇒ train Binary classifier for each pair of (digits) Ex
  - # of needed classifiers = $\frac{N(N-1)}{2}$ (45 for MNIST)
  - # Each classifier needs to train on a subset of training set

* Performance measures

(1) Accuracy
- → $\frac{\text{correct predictions}}{\text{total \# of predictions}}$
- → Measuring accuracy on certain # of folds (cross validation)

(2) confusion matrix
- → recall = $\frac{TP}{TP+FN}$, precision = $\frac{TP}{TP+FP}$

(3) F1 Score $F_1 = \frac{TP}{TP + \frac{FN+FP}{2}}$

* If we increase decision threshold ⇒ (↑ precision) eliminate FP
* // // Decrease // // ⇒ (↑ recall) eliminate FN

Lecture (9) Knn

⇒ Supervised learning, used for regression & classification
(Lazy Learner algorithm)

- How it works?

Select # of neighbors and calculate Euclidean distance between the new point & each point of (K) then take the nearest K's to the point and assigns the new point to that category in which number of neighbors is maximum

⇒ pros & cons

pros: (easy to implement, robust to noisy data, efficient when training data is large)

cons: (slow and lazy, high computational cost)

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \qquad X_1 = 8.0936 \;,\; X_2 = 5.3657$$

⇒

X_1	X_2	Y	$(X_1 - x_1)^2$	$(X_2 - x_2)^2$	Sum	Distance
3.3952	2.3312	0	22,0749 (take X_1 from table − x_1 new)	1,0701 (same as X_1)	23,1450 (Sum the previous two columns)	4,810 (take the Sqrt of the prious column)

then take the last distance according to # of neighbors and make the decision

---

Lecture (10) SVM (Binary classification)

⇒ High accuracy & low computation power, used for regression & classification, the objective is to find hyperplane in N-dimension which classify the data | output = $Y(B_1 X_1) + (B_2 X_2)$

⇒ optimum hyperplane (maximum margin to ensure classifying with more confidence)

⇒ Kernal trick to transform the space into high dimensional space and use dot product without computing new higher space dimensions

→ Linear Kernal

→ Polynomial Kernal (more general)

update if output > 1, $b = b(1 - \frac{1}{t})$ t: iteration
update if output < 1 ⇒ $b = (1 - \frac{1}{t})b + \frac{1}{\text{lambda} * t}(y * x)$

⇒ Output margin (greater than 0 ⇒ label 1, otherwise label -1)

Lecture (11) clustering

Identify similar instances & assign them into one cluster, used for classification, anomaly detection and semi-supervised learning

K-means: * Initialize Centroids & find the Euclidean distance between the centroid and point the assign the point to certain cluster according to (K)

* Optimum # of Clusters (K)

→ Inertia: mean squared distance between instance & closest centroid

→ Silhouette Coefficient: between (1 & -1)

- 1 or close to it means that the instance is inside its own cluster and far from other clusters
- 0 or close means that instance is close to boundary
- -1 or close means that instance is assigned to wrong cluster

* We plot the silhouette score with respect to K to find the optimum # of clusters

⇒ We can use clustering for preprocessing (dimensionality reduction)

Lecture (12) Ann

* perceptron is used for classification and regression, a single layer of perceptrons consists of (TLUs)

* the slides are step function for Activation (either 0 or 1)

* DNN: when it has two or more hidden layers

⇒ Multilayer perceptron steps:-

(1) Forward pass (2) back propagation (3) fine tune (Gradient descent)

* MLP's for Regression: (1 to 5 hidden layers, 10 to 100 neuron per hidden layer MSE for loss)

* MLP's for classifications (uses softmax activation to ensure probabilities between 1 and zero for each class, 1 to 5 hidden layers, 10 to 100 neurons per layer uses logistic function for binary & multilabel classification and soft max for multi-class, cross entropy for loss)

Q1) Advantages of loss function?

① helps in fine tune parameters

② measure the distribution between the true & predicted values (cross entropy)

Q2) What is soft max used for?

Ensure that the probabilities are between 1 and 0 for each class (useful in multiclass classification problems)

Q2.3) Multi independant task need?

→ In context of NN there are many answers for the Question

① input layer, shared layers, task specific layer

② Loss function ③ training process

The correct answer: (Chat Gpt) multi task learning needs Validation data

Q4) The best algorithm for multiclass classification?

Neural networks

Q5) Knn euclidean distance?

Q6) How SVM classify with more confidence?

Choosing the optimum hyper plane which provide maximum margin

Q7) How pca works?

Find correlated features and captare the highest variance (impact on target value) by calculating Eigen values of components and choose the highest value

أي سؤال تجاه المادة أو مساعدة،
تواصل مع الحساب المعطى ⬇️