Contents

Class was not offline (watch Recordings).

Linear Decision Boundaries

- Line in 2D plane $y = mx + c$.
- Plane in 3D Plane.

Generalized line equation in d dimensions:

$$f(x) = w_0 + w_1 x_1 + \cdots + w_d x_d = 0$$

- For points on line $f(x) = 0$
- On either side
- Class ω_1: $f(x)$ +ve
- Class ω_2: f(x) -ve

Goal

Learn the parameters w_i so that we can separate training samples.

Example

Take a 3D Plane. The boundary is a plane $y = f(x)$.

Now the intersection of this plane with the $x_1 x_2$ plane ($y = 0$: ground plane) gives us the decision boundary $f(x) = 0$.

Representing $f(x)$

$$f(x) = w_0 + w_1 x_1 + w_2 x_2$$

Let $\mathbf{w} = [w_0\ w_1\ w_2]^T$ and $\mathbf{x} = [1\ x_1\ x_2]^T$.

x known as augmented feature vector (since 1 extra value for mathematical purpose).

Then $f(x) = wx$ or $w^T x$.

Decision boundary is given by $w^T x = 0$.

Learning the Linear Boundary

- Randomly initialize w_i
- Iteratively modify w_i

Minimizing a Loss Function

Define a Loss Function $J(w)$ Plot Parameter space $J(w)$ vs w_i

How to know which direction to move in (for minima of $J(w)$)?

Use $-\nabla J$ to move towards the minima.

Gradient Vector:

$$\nabla J = \frac{\partial J}{\partial w}$$

Gradient Descent Algo

- Randomly initialize w : w^0
- $w^{t+1} = w^t - \eta \nabla J$
- repeat until $\eta \nabla J \to 0$

What should be the Loss Function?

1. Number of misclassified samples

- Problem: this is an integer. So changing w by a little will not really affect $J(w)$.

2. Take the idea $w^T x > 0$ for +ve class and negative for -ve class

- We can put $+1$ and -1 for the two classes. so $y_i \times w^T x_i > 0$ for all samples
- $J(x) = -y_i w^T x_i$ (Take negative because the abs value indicates correct classification)
- $\nabla J = -y_i x_i$
- Called Perceptron Update Rule (Perceptron Loss Function)

We can compute loss function - per training sample - single sample / stochastic GD - over the whole batch - Batch GD - something in between - Mini batch GD (in NNs we do this one)

Batch Perceptron Rule

only take the samples that are misclassified and update w for them.

Consider Perceptron Loss function

If Loss Function does not converge: - When the samples are not linearly separable (some misclassification will always happen). - Even if it's linearly separable, is our hyperplane the best separating hyperplane.

Perceptron Loss function converges to some kw (where k is constant value and w is final parameter vector). Proof: refer DHS.

Variants: Relaxation Algorithm

$$J_q(w) = \sum_{x_i in X} (w^T x_i)^2$$

- function becomes smooth (too smooth for practical use: converges at the boundary of separating space itself. Also updates in w are very slow.)
- dominated by longest vectors (idea: normalize the vectors)
  - while normalizing we also add a margin. We add margin so that we don't stop at the boundary separating the points but move to the best separating hyperplane

$$J_q(w) = 1/2 \sum_{x_i in X} (w^T x_i - b)^2 / \|x_i\|^2$$

Note: X is the set of samples for which $w^T x_i < b$. Here b is a constant.

Linear Regression Problem

Find hyperplane that approximates the output value y in terms of inputs. (we look at linear regression rn but generalize it later).

Define a loss function and use GD

Possible loss functions: Distance of samples from line (or squared distance)

$$J_{linear}(w) = 1/2 \sum_i (w^T x_i - y_i)^2$$

Interesting point that we don't need GD but can compute the hyperplane (closed solution exists) without using GD.

Fitting a curve can also be linear regression (as long as the Loss function is Linear in w).

Example $f(x, w) = \sum_{j=0}^{M} w_j x^j$.

then $J(w) = 1/2 \sum_i (f(x_i, w) - y_i)^2$.

Genric form

$$f(x, w) = \sum_{j=0}^{M} w_j \phi_j(x)$$

ϕ function known as basis function.

Function is still linear in the unknowns w_j.

Logistic Regression

Non-linear regression.

we use logistic sigmoid function

$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

$d\sigma/da = \sigma(1 - \sigma)$

Thus, we can use logistic regression to solve (binary) classification problems.

Here,

$$J(w) = 1/2 \sum_i (\sigma(w^T x_i) - y_i)^2$$

Logistic R not affected by outliers (since it converges nicely to 0 and 1).

Multi-linear regression: when we want to determine output (which is a vector). Regressors for each output value in the vector.