3.1.6

DL.21 Exercises

Date
Page

① $x_1, \dots, x_n \in \mathbb{R}$, $\sum_i (x_i - b)^2$ ← minimized (like Loss funct)

(i) Analytical Solution
(Solving it by rigorous mathematical treatment to get exact an solution)

$X = \begin{bmatrix} x_1 - b \\ x_2 - b \\ \vdots \\ x_n - b \end{bmatrix}$, Total = $X^T X$ loss

$\frac{\partial X^T X}{\partial b} = 2X^T \frac{\partial X}{\partial b}$ } General result in Matrix Calculus for any n×m matrix

Important Notations

$\frac{\partial \vec{a}}{\partial x}$ → vector, $\vec{a}$ → scalars. $= \left[\frac{\partial a_1}{\partial x}, \frac{\partial a_2}{\partial x}, \frac{\partial a_3}{\partial x}\right]^T$, $\frac{\partial (x)}{\partial \vec{a}}$ → scalar, $\vec{a}$ → vectors $= \left[\frac{\partial x}{\partial a_1}, \frac{\partial x}{\partial a_2}, \frac{\partial x}{\partial a_3}\right] = \nabla x$ → divergence

$\left(\frac{\partial \vec{a}}{\partial \vec{b}}\right) = \left(\frac{\partial a_i}{\partial b_j}\right)$ → 2nd order tensors ,

$\frac{\partial}{\partial \vec{x}}(\vec{x} \cdot \vec{a}) = \frac{\partial}{\partial \vec{x}}(x^T a) = \frac{\partial}{\partial \vec{x}}(a^T x) = a$

$\frac{\partial}{\partial \vec{x}}(\underset{\text{matrices (tensors)}}{AB}) = \frac{\partial A}{\partial \vec{x}} B + A \frac{\partial B}{\partial x}$ OR $\frac{\partial A}{\partial x} B + \frac{\partial B}{\partial x} A$

$\frac{\partial}{\partial \vec{x}}(x^T A x) = (A + A^T) x$

$\therefore \frac{\partial X^T X}{\partial b} = 0$ } condition for minimizing loss

$2X^T \frac{\partial X}{\partial b_i} = 0$, $\frac{\partial X}{\partial b_i} = [-1, -1, -1, \dots, -1]^T$

$2\sum_i (x_i - b) = 0$

$\frac{\sum x_i}{n} = \frac{nb}{n} \Rightarrow \boxed{b = \frac{\sum x_i}{n}}$

(ii) $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(x-\mu)^2\right)$

(Normal Dist^n)

$y = w^T x + b + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$

$y - w^T x - b = \epsilon$, $y \sim w^T x \sim X$; $b \to \mu$

$P(\epsilon) = P(y/x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(y - w^T x - b)^2\right)$

~~(2)~~

↳ Given x variable = 'x', what is the prob y variable = 'y' $\Rightarrow P(\epsilon)$

$P(Y/X) = \prod_{i=1}^{n} P(y^{(i)}/x^{(i)})$

According to the principle of maximum likelihood, (w, b) are best choices when it maximizes the likelihood over entire dataset.

(prob of y, x → linear relation → $y = w^T x + b$, $\epsilon = 0$)

(Likelihood is $P(Y/X)$)

$-\log P(Y/X) =$

$\sum_{i=1}^{n} \left[\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(y^{(i)} - w^T x^{(i)} - b\right)^2\right]$

For the assumption that σ is fixed,

When minimizing $-\log P(Y/X)$, w.r.t (w, b),

(1) the first term has no effect.

(2) The second term is equivalent to minimizing the squared sum error, except for the multiplicative constant.

Minimizing $\sum[\bar{x}_i - b]^2$, $\bar{x}_i = y^{(i)} - w^T x^{(i)}$

(iii) $Loss = \sum_i |(x_i - b)^2| = \sum_i \sqrt{(x_i - b)^2}$.

The same approach like (1 i) can't be applied here, because the derivative of mod is not defined.
But I don't know, if it can be reasonably solved to get an optimal solution.

(By intuition, it seems $b = \sum x_i / n$ will give the minimum value, as the task is still to minimize $\sum_i \sqrt{(x_i - b)^2}$,
→ for an unbiased term like $(x_i - b)^2$, b is center (mean value) should give the minimized loss)

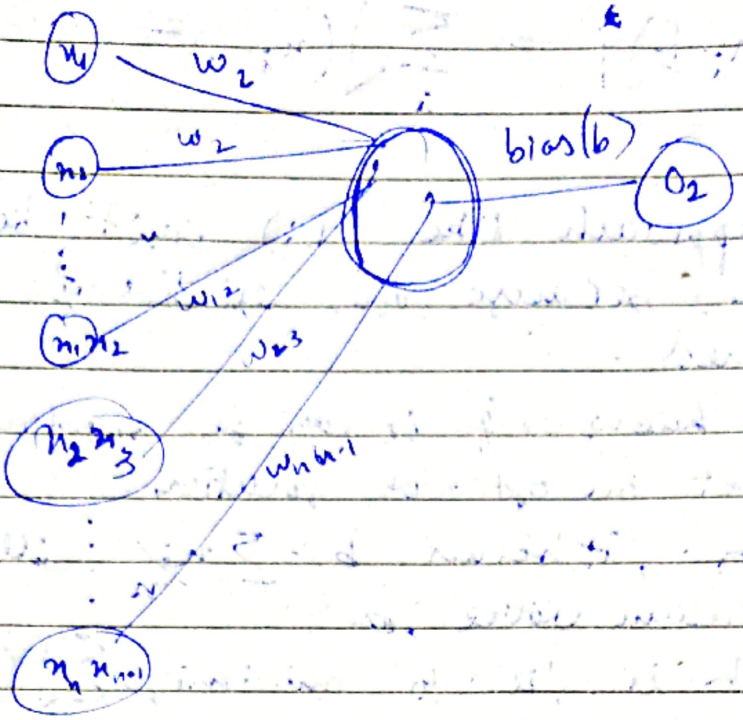
(iv)
(2) Affine functions : linear function + constant
(eg. $x^T w + b$)

Linear functions : $\tilde{x} = \begin{pmatrix} x \\ 1 \end{pmatrix}$ { higher dimensional space
on $(x, 1)$

$f(x) = x^T w + b$; $x^T \in (x_1, x_2, x_3 \ldots x_n)$
(affine function) $w \in (w_1, w_2, w_3 \ldots)$
weights / slopes

$f(x) = \tilde{x}^T w'$; $w' \in (w, b)^T$
(linear function on $(x, 2)$) $\tilde{x}^T \in (x, 1)$

(3)



(4) Analytical solution... (on mean squared loss → for function

$$\partial_w \|y - Xw\|_2^2 \quad \frac{\partial}{\partial w} |y - x^T w^* - b|$$

$$\partial_w \|y - Xw\|^2 = 0 \quad \therefore 2X^T(Xw - y)$$

$$X^T y = X^T X w$$

$$w^* = (X^T X)^{-1} X^T y$$

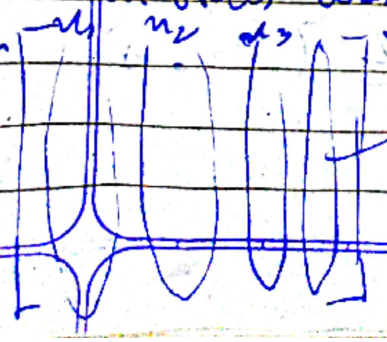
(optimized solution)

∴ unique w^* only for $X^T X$ is invertible.

∴ $X^T X$ is invertible → X has full rank,

$X \in \mathbb{R}^{n \times d}$ (design matrix)

Rank of a matrix: describes the dimension of the vector space generated by its rows or columns. In other words, no. of independent columns.



basis vectors of the generated space. (each is n-dim)

∴ maximum dim generated (Rank) = max (n, m)

(i) if $X^T X$ is not invertible ($\det(X^T X) = 0$), i.e., then no unique solution for w (as no. of solutions) which means, this algorithm would fail to provide the optimum solution for $w = (\omega, b)$

(ii) ① ~~Mistake~~, we can drop the dependent coordinate (as it is not helping in prediction of y),

② If we add a small amount of coordinate-wise (n_1, n_2 ...) independent Gaussian Noise to all the entries of X. Then $X^T X$ will be invertible with very high probability.

(iii) It is necessary to add Gaussian Noise over all the coordinates rather than just to a single datapoint ($y \to y + \epsilon$).

$$X = \begin{bmatrix} x_1 + \epsilon_1 & x_2 + \epsilon_2 & \cdots \\ x_1 + \epsilon_1 & x_2 + \epsilon_2 & \cdots \\ x_1 + \epsilon_1 & x_2 + \epsilon_2 & \cdots \\ \vdots & \vdots & \end{bmatrix}_{n}$$

ϵ in each element would be different

$$\therefore E(X^T X) = E\left[(x_i + \epsilon_i)(x_j + \epsilon_j)\right]$$

$$= E\left[x_i x_j + \epsilon_i x_j + \epsilon_j x_i + \epsilon_i \epsilon_j\right]$$

$$\Rightarrow x_i x_j + 0 \quad 0 \quad 0,$$

$E(\epsilon_i) = 0$

$E(x_i) = x_i$ deterministic

(iv) minibatch Stochastic Descent

$$(w,b) \leftarrow (w,b) - \frac{\eta}{|B|} \sum_{i \in B_t} \partial_{(w,b)} l^{(i)}(w,b)$$

Loss function

The Stochastic Gradient Descent doesn't rely on the invertibility of X, but instead iteratively updates the regression coefficients based on randomly selected subset of the data. The presence of linearly dependent columns (rows) can cause the loss surface to have multiple minima. (because two/more columns are dependent)

Does it make sense to search the saddle points? → paper - qualitatively characterizing neural networks optimization problems — GoodFellow.

(5i) $p(\epsilon) = \frac{1}{2} \exp.(-|\epsilon|)$, $y = x^T w + b + \epsilon$

$p(y|x) = \frac{1}{2} \exp.\left(-|y - x^T w - b|\right)$. $y - x^T w - b = \epsilon$

$-\log\left(p(y|x)\right) = -\log\left(\frac{1}{2}\right) + |y - x^T w - b|$

(ii) May be, it solvable for w, b, but since the derivative of MOD is undefined, therefore a closed-form solution is not possible.

(iii) Minibatch - SGD algorithm:

→ initialize (w,b) with random variable

→ choose hyperparameters $(\eta, |B|)$

$$\rightarrow (w,b) \leftarrow (w,b) - \frac{\eta}{|B|} \sum_{i \in B_t} \partial_{(w,b)} l^{(i)}(w,b),$$

$$l(w,b) = -\log\left(P(y|x)\right)$$

→ Stopping criterion (eg. $\delta(w,b) \leq \delta$)

Let's assume for computation, we smoothened $|y - x^T w - b|$ near stationary points, we would result in the slope near stationary point tending to infinity → This could cause the algorithm to diverge.

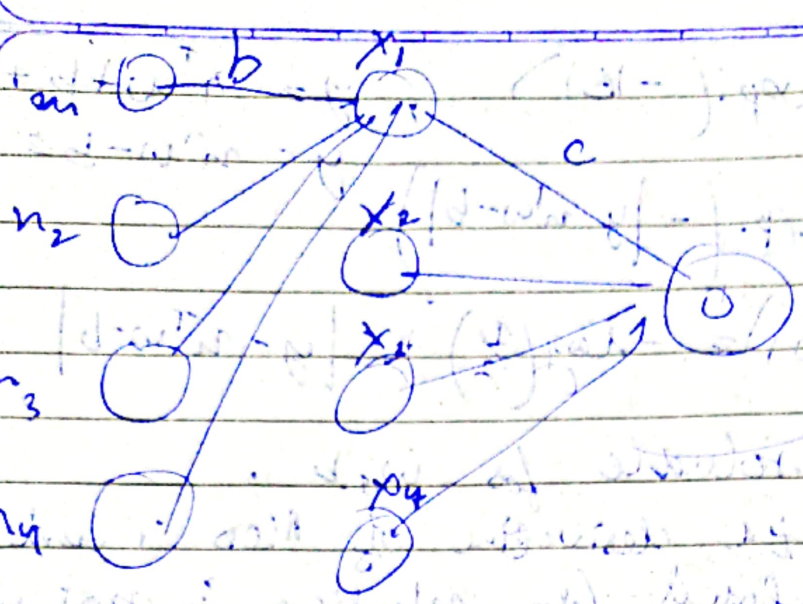
Solutions —

① Use a variant of gradient descent that includes a regularization term, such as gradient descent with momentum or RMSProp.

② By skipping a chunk near the stationary point.

③ ~~Redefining to little accurate but compatible function~~ of stationary

(6)



$$O = \left(w_{11}^{21} C_1 + w_{21}^{2} C_2 + w_{31}^{*} C_3 + w_{41}(C_4)\right) n_1 +$$
$$\left(w_{12} C_1 + w_{22}^{} C_2 + w_{32} C_3 + w_{42}(C_4)\right) n_2 +$$

$$= \vec{C} \cdot \sum_{i=1}^{n} \vec{w}^{i} \cdot n^{i}$$

- This will just like a single-layer only.
- This can't model non-linear relations, to model such relations, more no. of linear variables are needed rather than layers.
- For larger n, variable coefficients may become very small or large, if the ranges of the variable are way apart.

(7) (i) Additive Gaussian Noise: ϵ is Added to the function

The Gaussian Noise varies from $-\infty$ to ∞, which means some house prices might be negative.

But excluding those negative prices (extreme values), still the normal dist., may be used for Prices

The house prices in general increase with time over years but ϵ would have an expectation of '0'. (Another ϵ could be added which would change with time but its expectation value would be also '0' which would mean that we need to modifying it to have a bias to imitate the fluctuations) ⟶ which just means gaussian noise is not appropriate & needs lot of modification & tweaking to fit the "houses price problem" well

(iii) Black-Scholes Model ⟶ option Pricing Pricing

(iii) log price may be better,
because only +ve values are allowed
Houses to increase after certain values

(8) (i) Selling of Apples is a discrete quantity, which Gaussian Noise can't model.

(ii) $p(k/\lambda) = \lambda^k e^{-\lambda}/k!$ { Poisson Distribution function

$\mu = E(k)$ (mean) λ = rate function

$= \sum_{k=0}^{\infty} k\, p(k/\lambda)$ k = no. of events (i.e. captures distribution over counts)

$= \sum_{k=0}^{\infty} \left(k \frac{\lambda^k}{k!}\right) e^{\lambda}$

$= \lambda e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^{k-1}}{(k-1)!}$

$= \lambda e^{-\lambda} e^{\lambda} = \lambda$

(iii) $-\log(p(k|\lambda)) = -\sum_{i=1}^{n} \log(p(k_i|\lambda))$ like the maximum likelihood problem

$= n\lambda - \left(\sum_{i=1}^{n} k_i\right) \log\lambda + \sum_{i=1}^{n} \log k_i!$

$\mathcal{L}(\lambda) = n\lambda - \left(\sum_{i=1}^{n} k_i\right) \log\lambda$ (loss function)

→ minimizing this value maximizes likelihood over entire dataset. (loss function for Poisson Distribution)

(iv) $\lambda \longrightarrow \log \lambda = t$, $\underbrace{\mathcal{L}(\lambda) = \mathcal{L}(t)}$

the function has to remain same.

$$\mathcal{L}(t) = n e^{t} - \left(\sum_{i=1}^{n} b_i\right) t$$