Suppose 2000×1500 Matrix. Decomposing using SVD.

Setting $r=100$ (100 col. of U & 100 col of V). $\Sigma \to 100\times100$

$r \to$ tuning parameter

$$\underset{2000\times1500}{\tilde{X}} = \underset{2000\times100}{\tilde{U}} \; \underset{100\times100}{\tilde{\Sigma}} \; \underset{100\times1500}{\tilde{V}^T}$$

~~applied on other~~ space. dimensionality reduction.

Adv
* For compression ~~applied on other space~~
* Always applied on other space.
* Used for dimensionality reduction.

LU Decomposition (Lower Upper)

2/09

$Ax = b$

$A = LU$

$$\underset{3\times3}{A} = \begin{bmatrix} 1 & 0 & 0 \\ l_1 & 1 & 0 \\ l_2 & l_3 & 1 \end{bmatrix} \begin{bmatrix} u_1 & u_4 & u_5 \\ 0 & u_2 & u_6 \\ 0 & 0 & u_3 \end{bmatrix}$$

$x_1 + x_2 + x_3 = 1$

$3x_1 + x_2 - 3x_3 = 5$

$x_1 - 2x_2 - 5x_3 = 10$

$Ax = b$

$$\begin{bmatrix} 1 & 1 & 1 \\ 3 & 1 & -3 \\ 1 & -2 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 10 \end{bmatrix}$$

$LUx = b$

$Ux = y$ → Backward subst.

$Ly = b$ → Forward substitution.

$A = LU$

$$\begin{pmatrix} 1 & 1 & 1 \\ 3 & 1 & -3 \\ 1 & -2 & -5 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & -2 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -3 \\ 0 & 0 & -5 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 3/2 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & 1 & \\ 0 & -2 & \\ & & \end{bmatrix}$$

$Ly = b$

$$\begin{bmatrix}1&0&0\\3&1&0\\1&-2&1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}=\begin{bmatrix}1\\5\\10\end{bmatrix}$$

$$\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}6\\-7\\2\end{bmatrix}$$

$y_1 = 1$

$3y_1 + y_2 = 5 \qquad \underline{\underline{y_2 = 2}}$

$y_1 - 2y_2 + y_3 = 10$

$1 - 4 + y_3 = 10 \qquad \underline{\underline{y_3 = 13}}$

Sometimes no solution.
zero pivot (...)
PLU solves

$Ux = y$

$$\begin{bmatrix}1&1&1\\0&1&-3\\0&0&-5\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1\\2\\13\end{bmatrix}$$

$x_1 + x_2 + x_3 = 1$

$x_2 - 3x_2 = 2$

$-5x_3 = 13$

$x_3 = \frac{-13}{5}$

$x_2 + 3 \times \frac{13}{5} = 2$

$x_2 = 2 - \frac{39}{5} = \frac{10-39}{5} = \frac{-29}{5}$

$x_1 - \frac{29}{5} - \frac{13}{5} = 1$

$x_1 = 1 + \frac{42}{5} = \frac{5+42}{5} = \underline{\frac{47}{5}}$

$$\begin{array}{r}29+\\13\\\hline 42\end{array}$$

$$A = \begin{bmatrix}1&0&0\\l_{21}&1&0\\l_{31}&l_{32}&1\end{bmatrix}\begin{bmatrix}u_{11}&u_{12}&u_{13}\\0&u_{22}&u_{23}\\0&0&u_{33}\end{bmatrix}$$

$l_{21} = -m_{21}$ (coeff)

$l_{31} = -m_{31} \quad l_{32} = -m_{32}$

$$\begin{bmatrix}1&1&1\\3&1&-3\\1&-2&-5\end{bmatrix} \Rightarrow R_2 \to R_2 - \underbrace{3}_{\text{coeff} = m}R_1 \begin{bmatrix}1&1&1\\0&-2&-6\\1&-2&-5\end{bmatrix} \Rightarrow R_3 \to R_3 - R_1 \begin{bmatrix}1&1&1\\0&-2&-6\\0&-3&-6\end{bmatrix}$$

$R_3 \to R_3 - \frac{3}{2} R_2$.

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & -6 \\ 0 & 0 & 3 \end{bmatrix} \Rightarrow U.$$

$l_{21} = m_{21} = -3$ $\quad l_{21} = 3$ $(-m_{21})$

$m_{31} = -1$ $\quad l_{31} = 1$

$m_{32} = \frac{-3}{2}$. $\quad l_{32} = \frac{3}{2}$.

$$\therefore L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 3/2 & 1 \end{bmatrix}.$$

$\therefore$ L a $Ux = y$

$Ly = b$.

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 1 & 3/2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 10 \end{bmatrix}$$

$y_1 = 1$

$3y_1 + y_2 = 5$ $\quad 3 + y_2 = 5$ $\quad y_2 = \underline{\underline{2}}$.

$y_1 + \frac{3}{2} y_2 + y_3 = 10$

$1 + \frac{3}{2} \times 2 + y_3 = 10$.

$4 + y_3 = 10$ $\quad y_3 = \underline{\underline{6}}$.

$Ux = y$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -2 & -6 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 6 \end{bmatrix}$$

$x_1 + x_2 + x_3 = 1$

$-2x_2 + -6x_3 = 2$.

$3x_3 = 6$ $\quad x_3 = \underline{\underline{2}}$

$-2x_2 - 6 \times 2 = 2 \Rightarrow -2x_2 = 2 + 12 = 14$

$x_2 = \underline{\underline{-7}}$

$x_1 - 7 + 2 = 1$

$x_1 - 5 = 1$ $\quad x_1 = \underline{\underline{6}}$

$$\therefore \cancel{x_1}\; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -7 \\ 2 \end{bmatrix}.$$

Cholesky Decomposition

* Modified version of LU decomposition.

Conditions:-

1) Matrix should be symmetric
2) Matrix be positive definite.

* If a matrix is symmetric & positive definite then we can decompose into unique lower (a becomes L^T)

$$\therefore \underset{(sy \& P)}{A} = LL^T$$

$L \Rightarrow$ $\cancel{l_{ii}} = \sqrt{}$ $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} (l_{ik})^2}$ $i = 1, 2, \ldots n$

if $(i>j)$ $l_{ij} = \dfrac{a_{ij} - \sum_{k=1}^{j-1} (l_{jk})(l_{ik})}{l_{jj}}$

11	12	13
21	22	23
31	32	33

if $(i<j)$ (upper) $l_{ij} = 0$

$6x + 15y + 55z = 76$
$15x + 55y + 225z = 295$
$55x + 225y + 979z = 1259$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

QR Decomposition

$$A = QR$$

* Any rectangular matrix can be represented as QR
* Q is orthogonal matrix $(Q = Q^T)$
* R is upper triangular matrix.

} Always exists

LSA
Least squares
$\underset{x}{\arg\min} \|Ax - b\|_2$

LSA $\underset{x}{\arg\min} \|Q^T(Ax - b)\|_2$.

$$\Rightarrow \arg\min_x \|Q^T(QRx - b)\|_2.$$

$Q^TQRx - Q^Tb$

$\underset{I}{\ } Rx - Q^Tb$

$$\Rightarrow \arg\min_x \|Rx - Q^Tb\|_2.$$

$$A \begin{pmatrix} 1 & -4 & 4 \\ 1 & 4 & -2 \\ 1 & 4 & 2 \\ 1 & -1 & 0 \end{pmatrix} = A$$

Solve by QR

~~comput~~ General computation for all Decomposition:

$$\alpha n^3 + O(n^2)$$

where $\alpha = \frac{1}{3}$ for Cholesky Decomposition. (less time but condition)

$\alpha = \frac{2}{3}$ for LU and $\alpha = \frac{4}{3}$ for QR

* QR more time, but solution exist for all matrix.

03/09. Least Squares Approximation (fitting values) dist bet x_i & y_i should be min.

For linear equations (x_i, y_i) — x_i → input, y_i → output.

$\therefore f(x) = x\beta + b$, β & b are vectors.

3 → bias (variable) ~~biased basis~~

4 → parameters.

$$\therefore f(x) - y = x\beta + b - y$$

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ 1 & x_{31} & x_{32} & x_{33} \end{bmatrix}_A \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}_x = y$$

$\beta_0 + x_{11}\beta_1 + x_{12}\beta_2 + x_{13}\beta_3$

$x\beta = y$

$y \to$ given, $x_i \to$ given.

$$x\beta - y \Rightarrow \|Ax - b\|_2$$

$\|Ax - b\|_2^2$: $(Ax-b)^T(Ax-b)$

(Not inverse exist state or inconsist)

Diff bet Ax & b should be min $\Rightarrow \arg\min_x \|Ax - b\|_2.$

$A^T A x = A^T b$ after differentiating $(Ax-b)^T(Ax-b)$

$x = (A^T A)^{-1} A^T b$

$\underbrace{(A^T A)^{-1}}$ → Pseudo Inverse

To avoid inverse, to reduce time ∴ we apply QR.

$A = QR$

$(QR)^T QRx = (QR)^T b$

$R^T Q^T Q R x = R^T Q^T b$

$R^T R x = R^T Q^T b$

$((QR)^T QR)^{-1} (QR)^T b$

$(R^T Q^T Q R)^{-1} R^T Q^T b$

$\cancel{(R^T R)^{-1} \cdot R^T Q^T} b$

$(QR)^T QR x = (QR)^T b$

$R^T Q^T Q R x = R^T Q^T b$

$R^T R x = R^T Q^T b$

$R x = Q^T b$

$R x = V$ (back substitution)

Partial Differentiation

* $\frac{df}{dx}$ (diff) → $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ (part Deri) — derivate → gives no based on tensor

* ML deals with multivariable fn.

gradient → vector (value & direction) either ascend or descend (single vector)

$$\text{gradient}, \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

1 vector → gradient

Jacobian Matrix

* Multiple functions we use jacobian matrix.
* Jacobian becomes gradient, if its a single function.

$\nabla f, \nabla g \Rightarrow$ Jacobian matrix

* Partial derivaties of vectors functions

$$J(f(x,y), g(x,y)) = \begin{array}{cc} \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial g}{\partial x} \\ \frac{\partial f}{\partial y} & \frac{\partial g}{\partial y} \end{bmatrix} \\ \nabla f \quad \nabla g \end{array}$$

General form :

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Q* Find the Jacobian matrix at point (1,2) of the following f^{ns}

$$f(x,y) = \underbrace{x^4 + 3y^2x}_{(f_1)}, \underbrace{5y^2 - 2xy + 1}_{(f_2)}$$

$\frac{\partial f_1}{\partial x_1} = 4x^3 + 3y^2$ $\qquad$ $\frac{\partial f_2}{\partial x_1} = \cancel{5y^2 - 2y} -2y$

-2×2 $\qquad$ $20-2$

$\frac{\partial f_1}{\partial y} = 6yx$ $\qquad$ $\frac{\partial f_2}{\partial y} = 10y - 2x$

$6\times1\times2$

$4+12$

$$J = \begin{bmatrix} 4x^3+3y^2 & 6yx \\ -2y & 10y-2x \end{bmatrix} = J_{(1,2)} = \begin{bmatrix} 16 & 12 \\ -4 & 18 \end{bmatrix}$$

Compute the Jacobian $\quad (0,-2)$

$f(x,y) = e^{xy} + y, \; y^2x$

$(2,-2,2)$ $\quad f(x,y,z) = z\tan(x^2-y^2), \; zy\ln\frac{z}{2}$

(π,π) $\quad f(x,y) = \frac{\cos(x-y)}{x}, \; e^{x^2-y^2}, \; x^3\sin 2y$

I Pt $(0,2)$, $f(x,y) = e^{xy} + y \ , \ y^2 x$

$$\frac{\partial f_1}{\partial x} = e^{xy} y \qquad \frac{\partial f_1}{\partial y} = e^{xy} * x + 1$$

$$\frac{\partial f_2}{\partial x} = y^2 \qquad \frac{\partial f_2}{\partial y} = 2yx$$

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} e^{xy} y & x e^{xy} + 1 \\ y^2 & 2yx \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 4 & 0 \end{bmatrix}$$

II Pt $(2,-2,2)$, $f(x,y,z) = z\tan(x^2 - y^2) \ , \ zy \ln \frac{z}{2}$

$$\frac{\partial f_1}{\partial x} = z\sec^2(x^2-y^2) \times 2x \qquad \frac{\partial f_2}{\partial x} =$$

$$\frac{\partial f_1}{\partial y} = z\sec^2(x^2-y^2) \times -2y$$

$$\frac{\partial f_1}{\partial z} = \tan(x^2-y^2)$$

Applications of Jacobian

I Neural Network

* A mapping from $R^m \to R^n$ (m dim mapped to n dim)

$m \to$ no. of features, $n \to$ no. of contexts / classes.

(details given → flower name)

* $x \to net(x)$
(R^m) (R^n)

Generally, $\overbrace{g\left(w^{(1)} \cdot f\left(w^{(0)} x + b^{(0)}\right) + b^{(1)}\right)}^{\hat{y}}$ $b \to$ bias (parameters more converging diff)

↑ (non-linear) mapping

* The layer output is y_i is a parameter.
* x_i input (x_i, y_i) plays imp in Neural network

$\underset{in}{x_i}$ $\underset{out}{y_i}$

$$J_{ij} = \frac{\partial \hat{y}_i}{\partial x_j}$$ $\frac{\partial y_1}{\partial x_1}$ $\frac{\partial y_1}{\partial x_2}$ . . . . .

II * To find out critical points in multivariable function (max, min, saddle) in optimization problems.

III * Used in Function approximation.

Linear approximation at a :- $f(a) + f'(a)(x-a)$

Using Jacobian: $f(x) \approx f(p) + J(p)(x-p)$.

Hessian Matrix

* Second order derivate ~~matrix~~ operator.

Gradient, $\nabla E(w) = \frac{\partial E(w)}{\partial w} = \begin{bmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \end{bmatrix}$

$\nabla^2 E(w) = \frac{\partial^2 E(w)}{\partial w^2}$, it will be a matrix (not vector)

w_1 & w_2

$$= \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} \end{bmatrix}$$

3 variables: 3 × 3 mat

* Hessian matrix is symmetric.
* Cholesky can be directly applied.

ie $\frac{\partial^2 E^a}{\partial w_1 \partial w_2} = \frac{\partial^2 E^a}{\partial w_2 \partial w_1}$

Generally form of Hessian matrix of function f with n variables $\partial^2 f$

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \frac{\partial^2 f}{} \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Calculate the Hessian matrix at point (1, 0) of the following multivariable f^n.

$$f(x,y) = y^4 + x^3 + 3x^2 + 4y^2 - 4xy - 5y + 8$$

$\frac{\partial f}{\partial x} = 3x^2 + 6x - 4y$ $\qquad$ $\frac{\partial f}{\partial y} = 4y^3 + 8y - 4x - 5$

$\frac{\partial^2 f}{\partial x^2} = 6x + 6$ $\qquad$ $\frac{\partial^2 f}{\partial y^2} = 12y^2 + 8$

$\frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x \partial y} = -4$ $\qquad$ $\frac{\partial^2 f}{\partial y \partial x} = -4$

$$H(f) = \begin{bmatrix} 12 & -4 \\ -4 & 8 \end{bmatrix}$$

Pt (1, 1) $f(x,y) = e^{y \ln(x)}$

Pt $(0, 1, \pi)$ $f(x,y,z) = e^{-x} \sin yz$

24/09 Applications of Hessian Matrix

Jacobian matrix → gives critical points (derivative)

* Gives insight to critical points
* Gradient and is zero at a particular point x, then f has point at x $x \rightarrow$ can be min point, max, saddle.

To calculate x (min, max, saddle)

Then we will find eigenvalue of Hessian matrix,

λ is +ve definite $\rightarrow$ min point

λ is -ve definite $\rightarrow$ max point

λ is mix of +ve & -ve defi $\rightarrow$ indefinite or saddle.

$\lambda = 0$ $\rightarrow$ mix of min, max, saddle point

Q. $f(x,y) = x^4 - 32x^2 + y^4 - 18y^2$. Find the maximum & minimum value of function if exists or point

$f(x,y) = x^4 - 32x^2 + y^4 - 18y^2$.

$\frac{\partial f}{\partial y} = 4y^3 - 36y$

$\frac{\partial f}{\partial x} = 4x^3 - 32 \times 2x = 4x^3 - 64x$.

$\frac{\partial^2 f}{\partial y^2} = 12y^2 - 36$.

$\frac{\partial^2 f}{\partial x^2} = 12x^2 - 64$.

$\frac{\partial^2 f}{\partial x \partial y} = 0$ $\frac{\partial^2 f}{\partial y \partial x} = 0$.

$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Find gradient $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} = 0$.

$4x^3 - 64x = 0$. $4x^3 = 64x$.

~~$2y - 36y = 0$~~

$x^2 = 16$. $x = \pm 4$.

$(0,0), (\pm 4, \pm 3)$

$(\pm 4, 0), (0, \pm 3)$

$4y^3 = 36y$ ~~$2y = 36y$~~ $y^2 = 9$ $y = \pm 3$.

Critical points are: $(0,0)(0,\pm 3)(\pm 4, 0), (\pm 4, \pm 3)$.

$\begin{bmatrix} 12x^2 - 64 & 0 \\ 0 & 12y^2 - 36 \end{bmatrix}$

Even function: $f(-x) = f(x)$

Since Even f^n : we reduce $\therefore (0,0)\,(4,0)\,(0,3)\,(4,3)$

$(0,0)$ $\lambda I - A$

$$\begin{bmatrix} -64 & 0 \\ 0 & -36 \end{bmatrix}$$

$\therefore (0,0)$ is max point

λ is $-$ve definite.

$(4,0)$

$$\begin{bmatrix} 12\times16-64 & 0 \\ 0 & 12\times-36 \end{bmatrix} \rightarrow \begin{bmatrix} 128 & 0 \\ 0 & -36 \end{bmatrix}$$

mix of +ve & −ve

$(0,3)$

$$\begin{bmatrix} -64 & 0 \\ 0 & 12\times9-36 \end{bmatrix}$$

$(4,3)$

$$\begin{bmatrix} 12\times16-64 & 0 \\ 0 & 12\times9-36 \end{bmatrix}$$

$(4,3)$ is min point

λ is + ve definite.

* Assume a diagonal matrix corresponding to Hessian matrix.

$$D = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}$$

Positive definite: $d_1 > 0 \ldots d_n > 0$.

Assuming $d_1 \ldots d_n$ are not biased towards +ve or −ve, then

$P(d_i) = 1/2$ (to be +ve or −ve)

one d_i is not depending on d_j, (Indep)

$$P(d_1>0, d_2>0 \ldots d_n>0) = P(d_1>0)\,P(d_2>0) \ldots P(d_n>0)$$

(Joint probability to +ve) $= \dfrac{1}{2^n}$

* The no. of parameters increases $P(d_i)$ will decrease exponentially. (same $1/2^n$ for max & min)

$P(\text{saddle}) = \cancel{B}1 - P(\max) - P(\min) = 1 - \frac{1}{2^n} - \frac{1}{2^n}$

$P(\text{saddle}) = 1 - \frac{1}{2^{n-1}}$ (close to 1) (no. of P ↑) Many points will be saddle as no. of parameter increases.

* Hessian matrix also used in Neural network. (matrix operations)
* CNN ~~used~~ uses second order derivative values (for weight)
* ~~Basean~~ Bayesian Neural Network uses Hessian matrix.

25/09 Automatic Differentiation : To reduce the complexity of finding derivative (Chain rule is used)

$x \to \underbrace{a \to b}_{\text{intermediate}} \to y$ (x is input & y is output) to find y's change.

$\frac{dy}{dx} = \frac{dy}{db} \cdot \frac{db}{da} \cdot \frac{da}{dx}$. (Chain rule) backward ↑ pass

*Computation can be saved by: using the value of $f(x)$ for $f'(x)$ & value obtained in forward pass

foward

$\frac{dy}{dx} = \frac{dy}{db} \cdot \left[\frac{db}{da} \cdot \frac{da}{dx}\right]$

backward

$\frac{dy}{dx} = \left[\frac{dy}{db} \cdot \frac{db}{da}\right] \cdot \left\{\frac{da}{dx}\right.$

* Backpropogation Algorithm : Input layer has more parameters than last layer (towards output layer) so we select ~~back~~ backward initially to solve. computation is faster. (outp → Input) → convergence occur.

Q. $f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2))$

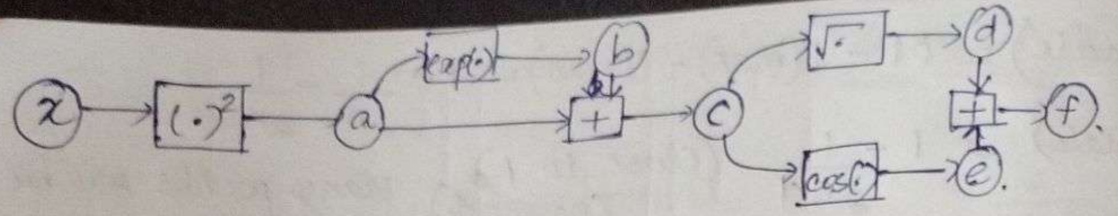
$= \sqrt{x^2 + e^{x^2}} + \cos(x^2 + e^{x^2})$

Let $a = x^2$, $b = \exp(a)$

~~$f(x) =$~~ $c = a + b$ $(x^2 + \exp(x^2))$

$d = \sqrt{c}$

$e = \cos(c)$ $f = d + e$.



$$\frac{\partial a}{\partial x} = 2x \qquad \frac{\partial c}{\partial a} = \frac{\partial c}{\partial b} = 1 \qquad \text{Find: } \frac{\partial f}{\partial x}$$

$$\frac{\partial b}{\partial a} = exp(a)$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \qquad \frac{\partial e}{\partial c} = -\sin(c) \qquad \frac{\partial f}{\partial d} = \frac{\partial f}{\partial e} = 1$$

$f = d + e$ (written in terms of c)

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d} \cdot \frac{\partial d}{\partial c} + \frac{\partial f}{\partial e} \cdot \frac{\partial e}{\partial c}$$

$$= 1 \cdot \frac{1}{2\sqrt{c}} - \sin c = \frac{1}{2\sqrt{c}} - \sin c$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot \frac{\partial c}{\partial b} = \frac{1}{2\sqrt{c}} - \sin c \times 1 = \frac{1}{2\sqrt{c}} - \sin c = \frac{\partial f}{\partial c}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \cdot \frac{\partial b}{\partial a} + \frac{\partial f}{\partial c} \frac{\partial c}{\partial a}$$

$$= \frac{\partial f}{\partial b} exp(a) + \frac{\partial f}{\partial c} \times 1$$

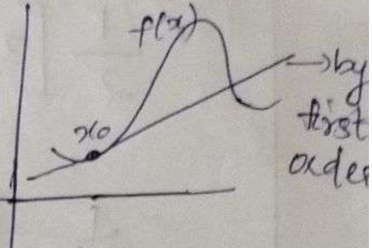
$$= \left(\frac{1}{2\sqrt{c}} - \sin c\right) exp(a) + \frac{1}{2\sqrt{c}} - \sin c = \left(\frac{1}{2\sqrt{c}} - \sin c\right)(exp(a) + 1)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial a} \cdot \frac{\partial a}{\partial x} = 2x\left(\frac{1}{2\sqrt{c}} - \sin c\right)(exp(a) + 1)$$

Multivariate Taylor Series

$e^{x} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$

In taylor series after finding many order it may be approximated / converged towards curve.



First order.

$f(x) \approx f(x_0) + \nabla_x f(x_0)(x - x_0)$

(First order)

General form

$$f(x) = \sum_{k=0}^{\infty} \frac{D_x^k f(x_0)}{k!} \delta^k$$

$D \to \nabla$

$\delta \to (x - x_0)$

$x_0 \to$ vector.

Taylor series expansion

Q: $f(x,y) = x^2 + 2xy + y^3$. compute the taylor expansion at $(x_0, y_0) = (1, 2)$.

$x_0 \to x_{01}, x_{02} \to (1, 2)$.

~~$f(x) = f(xy) = f(1,2) + $~~ ~~$k=0$~~

$k=0$

$f(x) = \frac{D^k x f(x_0)}{0!} \cdot \delta^0 = 1$.

$f(1,2) = 1 + 4 + 8 = \underline{\underline{13}}$

$k=1$ $\frac{D_{xy}^1 f(1,2)}{1!} \delta^1 = \begin{bmatrix} \frac{\partial f}{\partial x}_{(1,2)} & \frac{\partial f}{\partial y}_{(1,2)} \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix}$.

(row vc) (colvc)

$\delta^1 = \begin{bmatrix} x-1 \\ y-2 \end{bmatrix}$.

$= \begin{bmatrix} 6 & 14 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix}$

$2x + 2y$

$2 + 4 = 6$.

$= \cancel{13}(6(x-1) + 14(y-2))$

$2y + 3y^2$

$2 + 3 \times \cancel{6}4$

$= \cancel{13}(6x - 6 + 14y - 28)$

$\frac{28f}{6}$

$= \cancel{13}(6x + 14y - 34)$

$\frac{\partial}{\partial x}\ 2x + 2y$.

$\frac{\partial^2 f}{\partial x} = 2$ $\frac{\partial f}{\partial x \partial y} = $

$\frac{\partial f}{\partial y} = \cancel{2}\ 6y$ $\frac{\partial f}{\partial y \partial x} = 2$

$\begin{bmatrix} 2 & 2 \\ 2 & 6y \end{bmatrix}$

$k=2$ $\frac{D_{xy}^2 f(1,2)}{2!} \delta^2 = \frac{1}{2} \delta^T H(1,2) \delta$

$= \cancel{\tfrac{1}{2}}\ \cancel{\delta}\ \frac{1}{2} \begin{bmatrix} x-1 & y-2 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix}$.

2 2
$\cancel{2}2$ 12

$$= \frac{1}{2}\begin{bmatrix} x-1 & y-2 \end{bmatrix}\begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix}\begin{bmatrix} x-1 \\ y-2 \end{bmatrix}$$

$$= \cancel{\frac{1}{2}\begin{bmatrix} 2(x-1)+2(y-2) \end{bmatrix}} \;\; \frac{1}{2}\begin{bmatrix} x-1 & y-2 \end{bmatrix}\begin{bmatrix} 2(x-1)+2(y-2) \\ 2(x-1)+12(y-2) \end{bmatrix}$$

$$= \frac{1}{2}\left[(x-1)\left(2(x-1)+2(y-2)\right) + (y-2)\left(2(x-1)+12(y-2)\right)\right]$$

$$= \frac{1}{2}\left(2(x-1)^2 + 2(y-2)(x-1) + 2(x-1)(y-2) + 12(y-2)^2\right]$$

$$= \frac{1}{2}\left(2(x^2-2x+1) + 2(xy-y-2x+2) + 2(xy-2x-y+2)\right.$$
$$\left. 12(y^2-4x+4)\right)$$

$\frac{1}{2} \times 2$

$$= \left\{x^2-2x+1+xy-y-2x+2+xy-2x-y+2+6(y^2-4x+4)\right\}$$

$k=3$.
$$\frac{D^3_{xy} f(1,2)}{3!}\delta^3 = \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k}^{D} D^3_x f(x_0)[i,j,k]\,\delta(i)\,\delta(j)\,\delta(k)$$

(Generalised)

$H \to$ Hessian matrix

$$D^3_x f(x_0)[i,j,k] \longrightarrow \begin{bmatrix} \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} \end{bmatrix}$$

Dim $\rightarrow$ $2\times2\times2$.

$$D^3_{xy} f[:,:,1] = \frac{\partial H}{\partial x} = \begin{bmatrix} & \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$ $(y-2)^3$

$$D^3_{xy} f[:,:,2] = \frac{\partial H}{\partial y} = \begin{bmatrix} & \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 6 \end{bmatrix}$$

$$\frac{D^3_{(x,y)} f(1,2)}{3!}\delta^3 = \frac{1}{6}\begin{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 6 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix}$$

1/10. **Gradient Descent Algorithm**

* Numerical estimate where function output is min/max.

$\nabla f = 0$ (find gradient & equate to zero & find value for parameter)

* Only approximate value.
* Iterative form.

Let $f(x)$, start at random point x_0, take small steps towards gradient; (start at point and converge stops when converge)

Gradient has magnitude and direction.

Going to the descent (-ve value) (Move towards it).

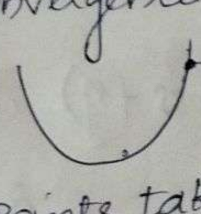
$x_1 = x_0 - \alpha \nabla f(x_0)$ (α - parameters)

General: $x_{n+1} = x_n - \alpha \nabla f(x_n)$ (Gradient Descent Algo) (May be take many iterations)

Two types of Problems:

1. Convex

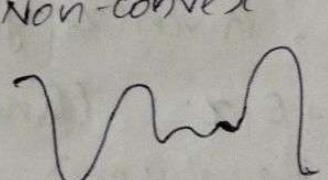
* Convergence is assured.



→ p lies here

* Points taken, solution lies in the

2) Non-convex.



* Many saddle points are possible.
* If we consider 2 points, the solution lies outside

Ex: How can we best approximate $\sin x$ with degree 5 polynomial? Value range between -3 to 3 (x). $x \in [-3, 3]$

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5$$ close to sin(x)

[continuous fn]

$$f(a_0, a_1, a_2, a_3, a_4, a_5) = \int_{-3}^{3} (P(x) - \sin(x))^2 \, dx$$

α ~~$a_0 = a_0 - \alpha \nabla f$~~ $a_0 = a_0 - \alpha \frac{\partial f}{\partial a_0}$

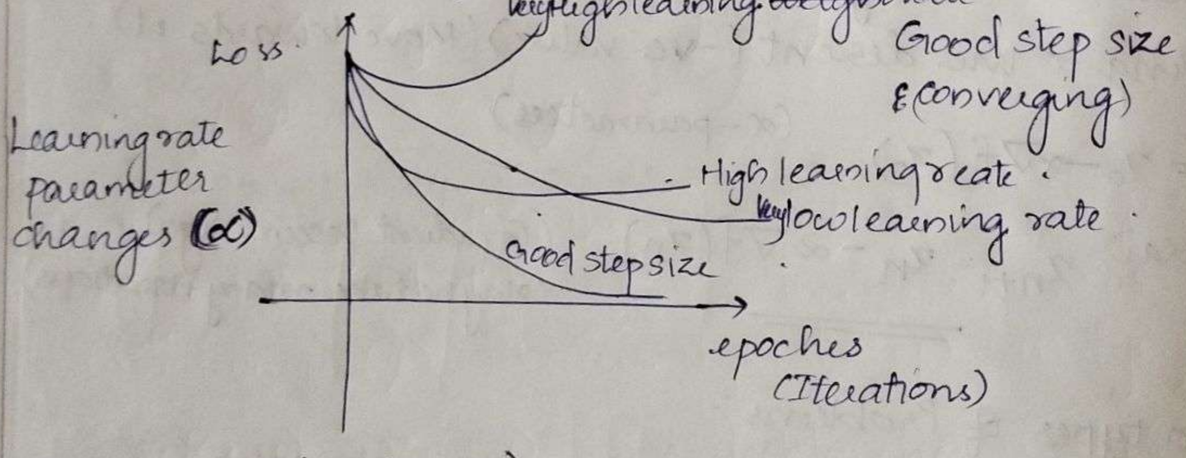
$a_1 = a_1 - \alpha \frac{\partial f}{\partial a_1}$

$\underline{\alpha = 0.1}$ (best value)

$\alpha = 0.1, 0.01, 0.001$

Problems with Algorithm
* May be arrive at localminima (if α not taken properly)
* Step size (Good stepsize convergence is assured for both convex & non-convex)
* Large (diverge)
* To small (slow convergence / or not)

Learning rate parameter changes (α)



Ex: Fit a line, Given (x_i, y_i)

(x_i, y_i) pairs discrete pairs

$y = \theta_0 + \theta_1 x$ (line eq)

(avg value min)

$$f(\theta_0, \theta_1) = \frac{1}{N} \sum_{i=0}^{N} y_i - (\theta_0 + \theta_1 x_i)^2 \quad \text{(loss } f^n\text{)}$$

$$= \frac{1}{2N} \sum_{i=0}^{N} y_i - (\theta_0 + \theta_1 x_i)^2$$

$2(\theta_0 + \theta_1 x_i)^+$

$\frac{1}{N} \cdot (y_i - \quad x_i$

To apply gradient descent:

$$\frac{\partial f}{\partial \theta_0} = \frac{-1}{N} \sum_{i=0}^{N} (y_i - (\theta_0 + \theta_1 x_i)) \qquad \frac{\partial f}{\partial \theta_1} = \frac{-1}{N} \sum_{i=0}^{N} (y_i - (x_i(\theta_0 + \theta_1 x_i$$

Repeat the following steps until convergence:

$\theta_0 = \theta_0 - \alpha \frac{\partial f}{\partial \theta_0}$

$\theta_1 = \theta_1 - \alpha \frac{\partial f}{\partial \theta_1}$

Diff strategies

* Batch Gradient Descent Algorithm → ~~single~~ * batch values set
  - n samples required
  - expensive computationally
  - time consuming
  - Equal weightage
  - * slow & convergence.
  - * To update single value all values to should be computed.
* Stochastic Gradient Descent (SGD) (Popular)

Batch ~~Batch~~

(M) * Faster convergence

* Easy to fit

* No. of iterations high. (DM)

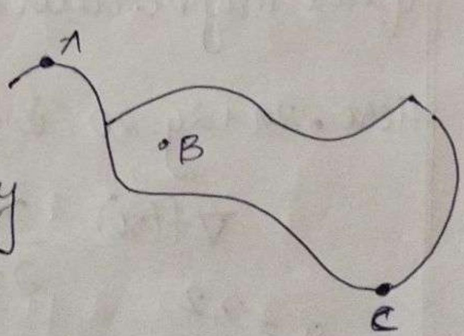
SGD ↓

* At a time single sample is used for computation
* Frequent updates (DM)
* Less Time ~~cony~~ consumption
* (Fluctuation high) (DM)
* Less memory requirement

* Mini batch Gradient Descent.
  * Both Batch Gradient & Stochastic.
  * Batch size is fixed & frequent updates.
  * Faster convergence & less memory requirement.

Gradient Descent with Momentum

* If gradient becomes small value updation does not happen & fall into local parameter.



* If we add momentum, then it may fall into global minimum:
* Also depends on velocity
* $\beta = 0.9$ (velocity)

Let w & b,
$$w = w - \alpha \frac{\partial f}{\partial w}, \quad b = b - \alpha \frac{\partial f}{\partial b}$$
Normal Gradient Descent

$w = w - \alpha\, vdw$ where $vdw = \beta\, vdw + (1-\beta)\frac{\partial f}{\partial w}$

$b = b - \alpha\, vdb$. $vdb = \beta\, vdb + (1-\beta)\frac{\partial f}{\partial b}$

update $= \alpha \nabla f$ ($\alpha = 0.1$, $\beta = 0.9$)

velocity = previous update $\times \beta$.

Parameter = (current) parameter value + velocity − update.

parameters: Adagrad, Adadelta, Adam g., (Second order) RMS optim, Nestro

16/10 Lechlanche Equation

Q1. Maximise an area of rectangle whose perimeter is 20m. (Find dimension)

~~2(l+b) = 20~~

Max $A = xy$ Max $f(x,y) = xy$.

$g(x,y) = P = 2x + 2y = 20$.

The Theorem says that, if the constraint equation $g(x,y) = c$ describes a Bounded set (B) in $\mathbb{R}^2$, then the maximum or minimum of $f(x)$ will occur at point (x,y) satisfying

$\nabla f(x,y) = \lambda \nabla g(x,y)$ $\lambda \to$ lachlancher multiplier. (or a boundary)

if not may be occur at boundary

Here, $2x + 2y = 20$ is bounded as its line ($0 \le x, y \le 10$)

$\nabla f(x)$. $\frac{\partial f}{\partial x} = \lambda \frac{\partial g}{\partial x}$ $\frac{\partial f}{\partial y} = \lambda \frac{\partial g}{\partial y}$ $x = 2\lambda$

$y = \lambda 2$

$y = 2\lambda$ $x = 2\lambda$

$\frac{y}{2} = \lambda = \frac{x}{2} \Rightarrow \underline{x = y}$

$2x + 2y = 20$

~~$2x+$~~

$2x + 2x = 20$ $4y = 20$

$x = 5$ // $y = 5$ //