Resume:-

1. LSTMs, Bi-directional LSTMs
- why LSTM > RNNs
- cell state vs hidden state
- Difference b/w LSTM & Bi LSTM

2. NLP
- Transformer architecture
- self-attention
- Glove, ELMo, FastText??

3. **Information retrieval
- metrics :- MRR, NDCG
- Traditional methods

4. Classical ML
- Mathematics : Logistic, Linear Regression
SVMs, PCA ✓, SVD ✓,

- Linear algebra revision ✓

-

Chi-square test

- Hypothesis testing (linear regression)
confidence interval
p-value
t-test, f-test etc.
non-parametric testing

coding practice

Deep learning

- Optimisation
  Local minima, saddle points, local maxima

→ Effect of sample sizes on p-value and confidence interval.

→ standard error

→ $\boxed{\text{A/B testing case studies; think about business metrics}}$ } resources around this

→ Using statistics to evaluate choices

Boosting algorithms
XGBoost.
Bagging Random Forest

c

$$(USV^T)^{-1} = (V^T)^{-1} (S)^{-1} (U)^{-1}$$
$$= \boxed{V S^{-1} U^T}$$

$VU^T = I$
$U^T = V^{-1}$
$(V^{-T})^{-1} = V$ $\quad UU^T = I$
$U^T = U^{-1}$

Matrix Rank

→ number of linearly independent rows : row rank
→ number of linearly independent columns : column rank

⇒ row rank = column rank = Rank

$$\text{Rank}(m,n) \leq \min(m,n)$$

For a $\boxed{\text{square matrix}}$ if determinant is non-zero, square matrix is a full rank matrix.

SVD

General setting: $\underset{m \times n}{A} = \underset{m \times m}{U} \underset{m \times n}{S} \underset{n \times n}{V^T}$, $B = USV^T$ is the best k-rank approximation to A

diagonal matrix $S_i = \sigma_i$ ($i = 1 \dots k$ ...) else $s_i = 0$

$\boxed{\text{Compact SVD}}$ $(m \times n)$ $\quad m \times r \quad r \times r \quad n \times r$

U, V : orthogonal matrices $UU^T = I$, $VV^T = I$
U, V : rotate , S : stretching (diagonal matrices)

$$\vec{y} = A\vec{x} = (USV^T)(\vec{x})$$

S : singular matrix with non-zero diagonal elements
→ if S has 0 element calculate pseudo-inverse
→ condition number $= \dfrac{S_1}{S_n}$ $\dfrac{\text{largest}}{\text{smallest}}$

If the second derivative (for all the points) of function exists, it is convex if $\partial^2 f(x) \geq 0 \ \forall x$ i.e. whether all the eigenvalues of hessian are non negative

→ Local minima is global minima for convex functions
→ There can be multiple minima though

Positive definite matrix : Eigenvalues $\boxed{>}$ 0

* If the hessian is everywhere positive semi-definite then function is convex.

$$H = \begin{bmatrix} \frac{d^2 f}{dx_1^2} & \frac{d^2 f}{dx_1 dx_2} & \cdots & \frac{d^2 f}{dx_1 dx_n} \\ \vdots & & & \\ \frac{d^2 f}{dx_n x_1} & \frac{d^2 f}{dx_n x_2} & \cdots & \frac{d^2 f}{dx_n^2} \end{bmatrix}$$

All +ve eigenvalues → local minima

All -ve eigenvalues → local maxima

mix → saddle point

Momentum method

additional ← $V_t = \beta V_{t-1} + g_{t,t-1}$

rate vector

$x_t = x_{t-1} - \eta \cdot V_t$

η → learning rate

Information Theory

Entropy: $H(x) = -\sum_{x} P(x) \log P(x)$

Mutual information:

$I(X,Y) = H(X) + H(Y) - H(X,Y)$

How much does X inform Y

KL Divergence: model — Amount of information loss when 'q' is used to approximate 'p'.

$$D_{KL}(P \| q) = \sum_{x} P(x) \log \frac{P(x)}{q(x)}$$

True distribution

* Maximizing log-likelihood of observing data X with respect to model parameters θ is equivalent to minimizing KL Divergence between the likelihood and true source distribution of the data.

Principle of maximum entropy

→ Max entropy distribution 'agrees with what is known, but expresses minimum uncertainty with respect to all other matters'.

$\underset{P(x)}{\text{Maximize}} \; -\sum P(x) \log P(x)$

constraints:- $\sum P(x) - 1 = 0$ $P(x)$ must integrate (sum) = 1

Max entropy distribution when:-

① over a finite discrete range {0, 1, .. N}

Uniform distribution

② continuous r.v. X with mean ~~by~~ μ

$$\int_x dx\,(x \cdot p(x)) - \mu = 0$$

⇒ Exponential distribution

③ Maximum entropy distribution with a variance σ

→ variance constraint: $\int dx\,(x-\mu)^2 \cdot p(x) - \sigma^2$

&

Also implicit $\boxed{\text{mean}}$ constraint

⇒ Normal distribution.

KL divergence measures the difference between two probability distributions over the same variable 'x'.

Logistic regression

MLE estimate:

Bernoulli

$$P(Y=1 \mid X) = h_\theta(x) = P$$

$$P(Y=0 \mid X) = 1 - h_\theta(x) = 1-P$$

$$\text{Likelihood} = (P)^{y} (1-P)^{1-y}$$

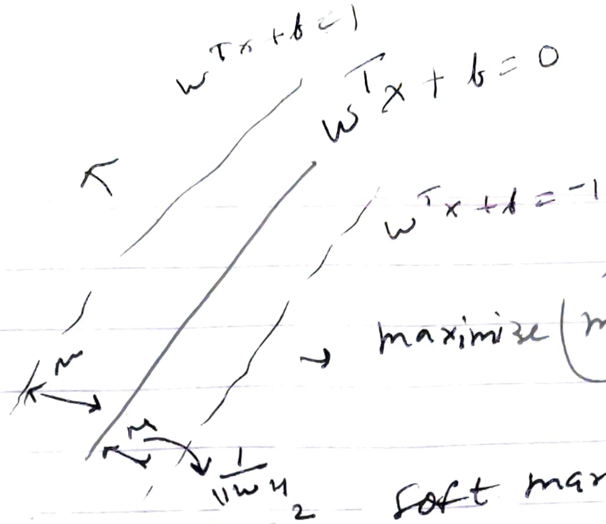
$$\frac{(W-K+2P)}{S} + 1 : \text{output shape}$$

$$Pr(|X-\mu| \geq K\sigma) \leq \frac{1}{K^2}$$

or equivalently

$$Pr(|X-\mu| \geq K) \leq \frac{\sigma^2}{K^2}$$

SUM



hard margin

$$\boxed{y(w^T x + b) \geq 1}$$

→ maximize (minimum distance of points from hyperplane)

soft margin

min. $\frac{1}{2}\|w\|^2 + C\,\xi^2$

such that $y_i(w_i x_i + b_i) \geq 1 - \xi_i$

Self-attention

sequence to sequence operation

$X_1, X_2, \ldots X_n$ $\boxed{K \quad Q \quad V}$

$(x_1 \cdot x_1)$ $(x_1 \cdot x_2)$ $(x_1 \cdot x_3)$

⇓

$Y_1, Y_2, \ldots Y_n$

$Y_1 = w_{11} x_1 + w_{12} x_2 + w_{13} x_3$

$Y_2 = w_{21} x_1 + w_{22} x_2 + w_{23} x_3$

Basic operation: $Y_i = \sum_j (W_{ij}) X_j$ weighted average of inputs

sum over j

$W_{ij} = (X_i) \cdot (X_j)$ → softmax operation to normalize

$(X X^T) X$

key, query, value

$K = W_K^T X$, $Q = W_Q^T X$, $V = W_V^T X$

$$\boxed{\frac{\text{softmax}(K \cdot Q)}{\sqrt{\text{dim}}}(V)}$$ self-attention

SGD or GD

$$\theta = \theta - \alpha \nabla L_\theta$$

~~Ad~~ Momentum

$$v = \beta v_{t-1} + \nabla L_\theta$$
$$\theta = \theta - \alpha v$$

Adagrad

$$r = r + \nabla L_\theta \odot \nabla L_\theta$$
$$\theta = \theta - \frac{\alpha}{\epsilon + \sqrt{r}} \cdot \nabla L_\theta$$

$\nabla L_\theta \rightarrow g$: gradient

decay factor

RMS Prop with momentum

$$r = (1-\rho) \nabla L_\theta \odot \nabla L_\theta + \rho(r)$$ exponential average
$$v = \beta v + \alpha \frac{1}{\sqrt{r}} \odot g$$
$$\theta = \theta - \alpha v$$

decay factors

Adam

$$s = \rho_1 s + (1-\rho_1) g$$
$$r = \rho_2 r + (1-\rho_2) g \odot g$$

$$s = \frac{s}{1-\rho_1}$$

$$r = \frac{r}{1-\rho_2}$$

$$\theta = \theta - \alpha \frac{s}{r}$$

SVD relation to Eigen-decomposition

SVD:

$$A = U \Sigma V^T$$

Eigen decomposition

$$A = X \Lambda X^T \quad *$$

A needs to be symmetric
U, V, X are orthonormal
Λ, Σ are diagonal

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^T U^T)$$

$$= U \Sigma\Sigma^T U^T$$

$$X \quad \Lambda \quad X^T$$

$$A^T A = V \Sigma\Sigma^T V^T$$

$$X \quad \Lambda \quad X^T$$

Shows how to SVD using Eigenvalue decomposition

$$\lambda_i = \sigma_i^2$$

SVD:-

1. Optimal low-rank approximation
2. Interpretability problem
3. Lack of sparsity

LSTM:

3 gates: Input, forget and Output

$$I = \sigma(\quad)$$
$$f = \sigma(\quad)$$
$$O = \sigma(\quad)$$

Additional memory cell C_t : using H_{t-1}

Candidate memory cell $\tilde{C}_t = \tanh(X_t, H_{t-1}, W)$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$$

→ forget from the previous cell state

→ read from the new cell state

$$\boxed{H_t} = O_t \odot \tanh(C_t)$$

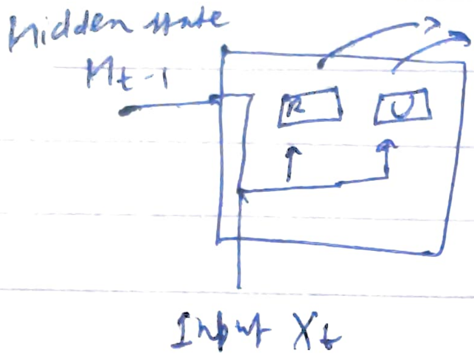
Gated Recurrent Unit :-

RNNs :-

$$H_t = f(W_h, X_t, H_{t-1})$$
$$O_t = g(H_t, W_o)$$

GRUs introduce two designs (gates):

1. Reset gate 2. Update gate



Sigmoid gates

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)$$
$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

Reset gate controls how much previous state we might want to remember

Update gate controls how much of the new state is just a copy of the old state

Candidate hidden states :-

① $$\tilde{H}_t = \tanh(X_t W_{xh} + \underbrace{(R_t \odot H_{t-1})}_{\text{element wise multiplication}} W_{hh} + b_h)$$

② $$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t$$

helps with vanishing gradient problem in RNNs and better captures the dependencies for sequences with large time step distances