Prob 1

Let's assume we have a eigenvector v_i of

Matrix $S = \frac{1}{N} XX^T$

$\therefore$ It will satisfy - $Sv = \lambda v$
λ ↳ eigenvalue.

$$\Rightarrow \frac{1}{N}(XX^T)v = \lambda v$$

$$\Rightarrow \frac{1}{N}(X^TX)(X^Tv) = \lambda(X^Tv) \quad \{\text{pre multiply } X^T \text{ both side}\}$$

Substitute $u = X^Tv$

$$\Rightarrow \frac{1}{N}(X^TX)u = \lambda u$$

$\therefore$ $u = X^Tv$ is nothing else but an eigenvector for $\frac{1}{N}(X^TX)$

In normal case to compute K eigenvectors for $\frac{1}{N}(X^TX)$ complexity will be $O(KD^2)$ but in this case order will be $O(KN^2) + O(KND)$

$O(KN^2)$ ↓ for decomp. of $\frac{1}{N}(XX^T)$

$O(KND)$ ↓ for multi matrix

$\therefore$ overall complexity $O(KND)$ ↓ which is less than $O(KD^2)$

as $\boxed{N < D}$

Scanned by CamScanner

Prob 2

$$h(x) = x\sigma(\beta x)$$

if we choose $\beta = 0$

$\Rightarrow \sigma(0) = 1$ ~~a h(x)~~ a linear activation function.

$\Rightarrow h(x) = \frac{x}{2} \Rightarrow$

if we chose $\beta \to \infty$

$\Rightarrow$ for $x < 0 \Rightarrow \sigma(\beta x) = \sigma(-\infty) = 0$

& for $x > 0 \Rightarrow \sigma(\beta x) = \sigma(\infty) = 1$

$$\Rightarrow h(x) = \begin{cases} x & ; x > 0 \\ 0 & ; x < 0 \end{cases}$$

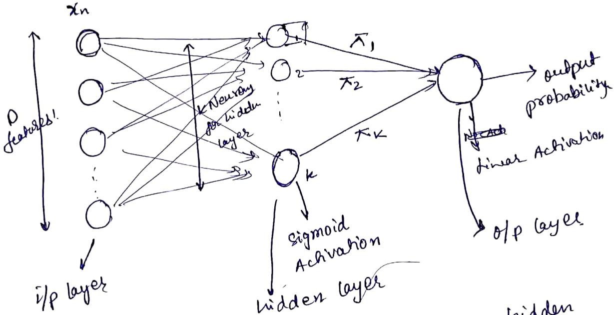
$\rightarrow$ RELU $\downarrow$ Activation f^n

Prob 3

$$P(y_n=1|x_n) = \prod_{k=1}^{K} P(y_n=1|z_n=k, x_n)\, P(z_n=k|x_n)$$

∴ Above equation can be modelled as a multiplication of two probabilistic fn(s).

⇒ 2 layer NN with one hidden layer.



∴ basically $\sigma(w_{zn}^T x_n)$ → acts as an hidden neurons.

Activation is sigmoid.

∴ Weights for first layer will be $(w_{z1}, w_{z2} \dots w_{zK})$ for k^{th} neurons.

∴ Weight for outputs will be $[\pi_1, \pi_2 \dots \pi_K]^T$ multiplying the probability of $p(z_n|x_n)$

Prob 4

$$P(u_n, v_m, \theta_n, \phi_m \mid X)$$

$$= \prod_{n=1, m=1}^{M, N} P(X_{nm} \mid u_n, v_m, \theta_n, \phi_m)\, P(u_n)\, P(v_m)$$

$\Rightarrow$ 0

$$= \prod_{n,m} \sqrt{\frac{\lambda_x}{2\pi}} \exp\left\{-\frac{\lambda_x}{2}\left(X_{nm} - \theta_n - \phi_m - u_n^T v_m\right)^2\right\}$$

$$\times \sqrt{\frac{\lambda_u}{2\pi}} \left\{\exp\left(-\frac{\lambda_u}{2}\|u_n - W_u a_n\|^2\right)\right.$$

$$\times \sqrt{\frac{\lambda_v}{2\pi}} \exp\left(-\frac{\lambda_v}{2}\|v_m - W_v a_m\|^2\right)$$

$$\Rightarrow NLL = \sum_{n,m}\left\{\frac{+\lambda_x}{2}\left(X_{nm} - \theta_n - \phi_m - U_n^T V_m\right)^2\right\}$$

$$+ \sum_{n}\left\{\frac{\lambda_u}{2}\|u_n - W_u a_n\|^2\right\}$$

$$+ \sum_{m}\left\{\frac{\lambda_v}{2}\|v_m - W_v a_m\|^2\right\}$$

NLL $\rightarrow$ cost function

Now, for W_u, W_v → minimization will be similar to linear regression model.

which will give :-

$$W_u^T = (A^T A)^{-1} A^T U$$

where; $A \rightarrow N \times D$ matrix of all a_n's

$U \rightarrow N \times K$ matrix of all u_n's

$$\Rightarrow \quad W_u = U^T A (A^T A)^{-1}$$

Similarly, $W_v = V^T B (B^T B)^{-1}$

for θ_n & θ_m | taking the derivatives yield.

$$\sum_m -\lambda_n (x_{nm} - \theta_n - \phi_m - u_n^T v_m) = 0$$

$$\Rightarrow \quad \theta_n = \frac{1}{M} \sum_m (x_{nm} - \phi_m - u_n^T v_m)$$

Similarly $\phi_m = \frac{1}{N} \sum_n (x_{nm} - \theta_n - u_n^T v_m)$

Now taking the derivative w.r.t. v_m we get.

$$\sum_n -\lambda_x \left(x_{nm} - \theta_n - \phi_m - u_n^T v_m\right) u_n$$
$$+ \lambda_v \left(v_m - W_v b_m\right) = 0$$

$$\Rightarrow \left(\sum \lambda_x u_n u_n^T + \lambda_v I_K\right) v_m = \lambda_v W_v b_m + \sum_n \lambda_x \left(x_{nm} - \theta_n - \phi_m\right) u_n$$

we can write $\sum u_n u_n^T = U^T U$

$$\Rightarrow \boxed{v_m = \left(\lambda_x U^T U + \lambda_v I_K\right)^{-1} \left[\sum_n \lambda_x \left(x_{nm} - \theta_n - \phi_m\right) u_n + \lambda_v W_v b_m\right]}$$

Similarly, for u_m

$$\boxed{u_m = \left(\lambda_x V^T V + \lambda_u I_K\right)^{-1} \left[\sum_m \lambda_x \left(x_{nm} - \theta_n - \phi_m\right) v_m + \lambda_u W_u a_n\right]}$$

∴ Using the equation show above. ALT-OPT algo will be :—

- Initial

PTO

ALT - OPT

→ ① Initialise latent variables $u_n = \hat{u}_n$; $v_m = \hat{v}_m$

→ ② Solve for $\hat{w}_u$, $\hat{w}_v$, $\hat{\theta}_n$, $\hat{\theta}_m$ using the eq^n.

→ ③ Solve $\hat{u}_n$, $\hat{v}_m$ using the eq^n

→ ④ Go to step ② if not converge.