MLSP Cheat Sheet-170333(karan),170714(soumya)

Digital Signal Processing

- Convolution = $x[n] * h[n] = \Sigma_m x[m] * h[n-m]$
- Correlation = $x[n] \otimes h[n] = \Sigma_m x[m] * h[n+m]$
- Discrete Fourier Transform = $x[n] = (1/N)\Sigma^{N-1}{}_{k=0} X[k] e^{j(2\pi/N)kn}$
- Short time fourier transform = $X[k,m] = \Sigma^{N-1}{}_{n=0} x[n] * w[n-mH] e^{-j(2\pi/N)kn}$; k=0,1,....,N-1

Evaluation Metrics

- Accuracy = $(TP+TN)/(TP+TN+FP+FN)$
- Precision = $TP/(TP+FP)$
- Recall = $TP/(TP+FN)$
- F-score(harmonic mean) = $2PR/(P+R)$

Regression

- Let $y = w_0 + w_1 x + w_2 x^2 + ... + w_D x^D$
  - If $D+1 < N$ minimize $(t-W)^T(t-W) + \lambda w^T w$ mean squared error (analytical soln $w = (\lambda I + {}^T)^{-1} ??^T t$
  - If $D+1 > N$ minimize $||t-y|| + \lambda w^T w$ and get analytically $w = ??^T(??^T)^{-1} t$
- For multiple inputs $Y = W$ solve analytically to get $w = ({}^T)^{-1} ??^T t$

Non-Linear Modelling:

- Perceptron: single neuron :
- Neural Network: Model size = $\Sigma_i N_{i+1}(N_i + 1)$

Activation Functions:

- Sigmoid : $\sigma(x) = 1/(1+e^{-x})$
- Hyperbolic tan : $tan(x) = (e^{-2x} - 1)/(e^{2x} + 1)$
- ReLU : $g(x) = max(0,x)$
- Softmax : $softmax(h) : y_c = e^{h_c}/\Sigma_{c'} e^{h_{c'}}$

Linear Classification :

- Two class classification: $w^T x + w_0$
- Distance from line: $y(x)$ / $||w||$
- Multi class classification use K-Class discriminant : $k^* = argmax_k\, y(x) = argmax_k\, w_k{}^T x + w_{0,k}$

Non-Linear Classification

- Multiclass classification: The output of the model is $\boldsymbol{h}$ or $h_c \in \mathbb{R}$, c = 0,1,2,…, C − 1; y = softmax(h); i.e $y_c = e^{h_c}/(\Sigma_{c'} e^{h_{c'}})$; and the loss function used is categorical cross-entropy $E_{xent} = -\Sigma_c t_c log(y_c)$
- Multilabel classification: The output of the model is $\boldsymbol{h}$ or $h_l \in \mathbb{R}$, c = 0,1,2,…,L − 1; y = softmax(h); i.e $y_l = sigmoid(h_l)$; and the loss function used is binary cross-entropy $E_{binxent} = -\Sigma_l t_l log(y_l) + (1-t_l) log(1-y_l)$

Neural Network Optimization:

- Momentum in gradient descent: $v = \beta v + (1-\beta)\Delta w$; $w = w - \alpha v$
- RMS prop: $S_{dw} = \beta S_{dw} + (1-\beta)(dw)^2$; $w = w - \alpha dw/(\sqrt{S_{dw}} + \varepsilon)$
- Hyperparameter Tuning:
  - Grid search: Try using all possible combination, time-consuming
  - Random Search: Random number of tries on values within a range
- Data Normalisation: $x_{i,s} = (x_{i,s} - \mu_i)/\sigma_i$

- Regularization: Early stopping: *search volume* α ηπ
- Dropout: Masking of neurons randomly from nn

Gaussian mixture modeling:
- K-means clustering:
  - $L = \Sigma_{n,k} r_{n,k} ||x_k - \mu_k||^2$
  - $\mu_{k'} = \Sigma_n r_{nk'} x_{k'} / \Sigma_n r_{nk'}$
  - $r_{nk} = 1,\ only\ when\ k\ = argmin_{k'} ||x_n - \mu_{k'}||$
- Gaussian mixture modelling/ soft k-means clustering:
  - $\gamma_{nk'} = \pi_{k'} N(S_n | \mu_{k'} \Sigma_{k'}) \,/\, \Sigma(\pi_{k'} N(S_n | \mu_{k'} \Sigma_{k'}))$
  - $\mu_{k'} = \Sigma_n \gamma_{nk'} S_n \,/\, N_{k'}$
  - $\Sigma_{k'} = \Sigma_n \gamma_{nk'} (S_n - \mu_{k'})(S_n - \mu_{k'})^T$
  - $\pi_{k'} = N_k / N$

Principal Component Analysis:
- Given data samples s(∈ ℝ^D)
- Normalize: $\boldsymbol{x}\, x_n = s_n - E[s];\ E[s] = (1/N)\Sigma s_n$
- Obtain variance matrix = $S = (1/N)\Sigma x_n x_n{}^T$
- Eigenvalue decomposition of S to get λ5, $\boldsymbol{u}$5; i = 1,…,D with λ5 in decreasing order
- Choose first M eigenvectors as the principal axes
- $= \sum_{i=1}^{M} (x_n{}^T u_i) u_i$

Non-Negative Matrix Factorization:
- $W_{ir} = W_{ir} (VH^T)_{ir} / (WHH^T)_{ir}$
- $H_{rj} = H_{rj} (W^T V)_{rj} / (W^T W H)_{rj}$

Probabilistic Latent Component Analysis:
- Expectation step: $q_t(z|f)\ =\ P_t(z) P(f|z) / \Sigma P_t(z) P(f|z)$
- Maximization step:
  - $P_t(z)\ =\ \Sigma_f V_{ft} q_t(z|f) / \Sigma_{z'} \Sigma_f V_{ft} q_t(z'|f)$
  - $P(f|z)\ =\ \Sigma_t V_{ft} q_t(z|f) / \Sigma_f \Sigma_t V_{ft} q_t(z|f)$

Probability:
- Joint probability = $P(w)\ =\ \Sigma_t P(t, w)$;Conditional Probability = $P(w|t)\ =\ P(w, t)/P(t)$;Bayes Theorem $P(t|w)\ =\ P(w|t)P(t)/P(w)$; condition for independence $P(w|t)\ =\ P(w)$; CDF $P(z) = \int_{-inf}^{z} p(x)dx$;Expectation $E[f(x)] = \int f(x)p(x)dx$;variance $var[f(x)] = E[(f(x) - E[f(x)])^2]$; conditional expectation $E[f(x)|y] = \int f(x)p(x|y)dx$; covariance $cov[x, y] = E[(x - E[x])(y^T - E[y]^T)]$; correlation $corr(x, y) = \frac{cov[x,y]}{\sigma_x \sigma_y}$
- Gaussian distribution $\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)} exp\{-\frac{1}{2\sigma^2}(x - \mu)^2\}$ estimate parameters from MLE $\mu_{ml} = (1/N)\Sigma X_i\ \ \sigma_{ml}{}^2 = (1/N)\Sigma(X_i - \mu_{ml})^2$; MAP $\mu_{map} = (\Sigma s_i/\sigma^2 + \mu_0/\sigma_0{}^2)/(N/\sigma^2 + 1/\sigma_0{}^2)$
- Multivariate Gaussian distribution: $\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\ \}$ $\mu_{ml} = (1/N)\Sigma X_i$; $\Sigma_{ml} = (1/N)\Sigma_{i=1}(s_i - \mu_{ml})\,(s_i - \mu_{ml})^T$