1) a) Missing entries in (A)

Matrix B → [POSITIVE SEMIDEFINITE]

Frobenius Norm = $\|A-B\|_F^2 = \sum_{i,j}^{n} (A_{ij}-B_{ij})^2$ is as small as possible

BACKGROUND:

1) B → Positive semi definite (Symmetric matrix B such that $x^TBx \geq 0 \;\forall\; x \in \mathbb{R}^n$)

All of B's eigen values are non-negative

Eigen decomposition → PDP^T [P → Orthonormal basis of eigen vectors of B]

Quadratic real function : $f(x) = x^TBx$ (x column vector in variables)

for some Q, Q^TBQ is positive semidefinite

2) Frobenius norm :

For every nxn square matrix $A \in \mathbb{R}^{n\times n}$

$$\|A\|_F = \left(\sum_{i,j}^{n} |a_{ij}|^2\right)^{1/2} = \left[tr\left(A^TA\right)\right]^{1/2}$$

CHECK:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = 1^2+2^2+3^2+4^2$$

$$A^TA = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1^2+3^2 & \\ & 2^2+4^2 \end{pmatrix}$$

PROPERTIES:

a) $\|AB\|_F \leq \|A\|_F \|B\|_F$

b) $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\,\|A\|_2$ (upper bound on spectral norm)

c) For orthogonal matrices U,V (Not changed by pre- or post-orthogonal transformation)

$$\|UA\|_F = \|AV\|_F = \|A\|_F$$

d) $\|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}$ where σ_i → SINGULAR VALUES OF A

if A is square and symmetric [σ_i → Eigen values.]

(a) $A = \begin{pmatrix} 3 & 1 & ? \\ 1 & 1 & -2 \\ ? & -2 & 6 \end{pmatrix}$ → Matrix B ?

1) Positive semidefinite

2) Minimize Frobenius Norm.

If matrices are written as vectors

then frobenius norm here computes distance between vectors

To minimize frobenius norm $||A-B||_F$

if A is a symmetric positive semidefinite matrix

then B = A thus minimizing frobenius norm

else we need to find the matrix B (POSITIVE SEMIDEFINITE)

that is closest to (A)

Let us check can we turn A into a psd matrix by filling any value

For a psd all of the eigen values ≥ 0

By observing rows of matrix A

Last row can be expressed as sum of first and second rows

So, if missing value is $\boxed{-4}$ this can be verified.

Now rank (A) < 3 → If we can prove A is psd

then $\boxed{B=A}$

a psd matrix that minimizes frobenius norm is $\boxed{A}$

considering filled $\hat{A}$ matrix $\begin{pmatrix} 3 & 1 & -4 \\ 1 & 1 & -2 \\ -4 & -2 & 6 \end{pmatrix} \leftarrow \hat{A}$

By computing eigen values of $\hat{A}$ we get $0,\ 5-\sqrt{19},\ 5+\sqrt{19}$

$\hat{A}$ symmetric matrix with all non negative eigen values $\Rightarrow \hat{A}$ psd.

So, $B = \hat{A} \longrightarrow \begin{pmatrix} 3 & 1 & -4 \\ 1 & 1 & -2 \\ -4 & -2 & 6 \end{pmatrix}$ s.t $\|\hat{A} - B\|_F = 0$

We can also find range of values for which A remains semi definite by writing it in the form.

$|A - \lambda I| = 0 \rightarrow$ Eigen values ≥ 0

$\hookrightarrow f(\lambda, w) = 0$

for cubic in λ to have all positive roots we can evaluate condition on (w)

that might also give us a range of w's for which B can be psd

OBSERVATION

Incomplete matrix A can be looked in as an R^{n^2} vector

Now we are looking for some B (R^{n^2} vector) that has closest euclidean distance

There can be region (R^d) in R^{n^2} which is formed when A is filled with random values

APPROACH - I :

(b) Here we write the optimization problem as follows :

$$\| A - B \|_F^2 \rightarrow \sum_{i,j} (A_{ij} - B_{ij})^2$$

[NON MISSING SET] ↓

Considering only the non missing values. $\longrightarrow$ $i,j \in \Omega$

In an effort to minimize frobenius norm by representing A in terms of B

$$f(V) = \sum_{i,j \in \Omega} \left(A_{ij} - V^{(i)T} V^{(j)} \right)^2$$

(c) Block coordinate descent for the above optimization can be applied as follows

i) Initialize $V_{ij} = A_{ij}$ $\forall (i,j) \in \Omega$ and 0 otherwise

ii) Now considering $V^{(i)}$'s as blocks

repeat

consider $V^{(i)}$ as variable

fix all the other $V^{(j)}$'s $[j \neq i]$

Minimize objective of $f(V)$

Process in step - 2 goes in a round robin fashion

considering n blocks corresponding to $V^{(i)}$

In step-2 :

Consider the optimization problem of minimizing over one $v^{(i)}$ fixing all the other $v^{(i)}$'s.

if $v^{(i)}$ is current variable then

NON MISSING VALUES

$$f(v_\cdot) = \min_{i \neq j} \sum^{+} (A_{ij} - v^{(i)T} v^{(j)})^2 \quad , \quad i,j \in \Omega$$

in our case $v^{(i)} \in \mathbb{R}^3$, $A_{ij} \in \mathbb{R}^{3\times 3}$

$$\|A-B\|_F^2 = \left[3 - v^{(1)T} v^{(1)}\right]^2 + 2\left[1 - v^{(1)T} v^{(2)}\right]^2 + \left[1 - v^{(2)T} v^{(2)}\right]^2$$
$$+ 2\left[-2 - v^{(1)T} v^{(3)}\right]^2 + \left[6 - v^{(3)T} v^{(3)}\right]^2$$

Consider for vector $[v_{11}, v_{12}, v_{13}]$ that gives.

$$f(\vec{v}^{(1)}) = \left[3 - v^{(1)T} v^{(1)}\right]^2 + 2\left[1 - v^{(1)T} v^{(2)}\right]^2$$

$$\frac{\partial}{\partial v_{11}} f = 2\left(3 - v^{(1)T} v^{(1)}\right)\left[-2v_{11}\right] + 4\left(1 - v^{(1)T} v^{(2)}\right)\left[v_{21}\right]$$

$$\frac{\partial}{\partial v_{12}} f = 2\left(3 - v^{(1)T} v^{(1)}\right)\left[-2v_{12}\right] + 4\left(1 - v^{(1)T} v^{(2)}\right]\left[v_{22}\right]$$

$$\frac{\partial f}{\partial v_{13}} = 2\left(3 - v^{(1)T} v^{(1)}\right)\left[-2v_{13}\right] + 4\left(1 - v^{(1)T} v^{(2)}\right)\left[v_{23}\right]$$

$$\Rightarrow \nabla_{\vec{v}^{(1)}} f = -4\left(3 - v^{(1)T} v^{(1)}\right)\vec{v}^{(1)} + 4\left(1 - v^{(1)T} v^{(2)}\right)\vec{v}^{(2)} = 0$$

$$f(V) = \min \sum_{i,j \in \Omega} \left(A_{ij} - V^{(i)T} V^{(j)}\right)^2 , \left(i,j \in \Omega\right)$$

$$\nabla_{\vec{V}^{(i)}} f(V) = \left[-4 \left(A_{ii} - V^{(i)T} V^{(i)}\right) \vec{V}^{(i)}\right] \cdot 1\{A_{ii} \neq ?\}$$

$$+ \sum_{j \neq i} \left[2 \left(A_{ij} - V^{(i)T} V^{(j)}\right) V^{(j)}\right] \cdot 1\{A_{ij} \neq ?\}.$$

Closed form solution might be very hard as the no. of variables that are missing in the matrix A_{ij} is unknown

A better approach of finding appropriate choice $V^{(i)}$ can be by using gradient descent where

$$V^{(i)} = V^{(i)} - \alpha \nabla_{\vec{V}^{(i)}} f(V)$$

(d) Its necessary to include an L2-penalty to the optimization problem in order to discourage large values of (V_{ij})

Optimization problem considered in step (b)

$$\min \sum_{i,j \in \Omega} \left(A_{ij} - V^{(i)T} V^{(j)}\right)^2 + \underbrace{\|V\|_F^2}_{\text{L2 - PENALTY}}$$

[ALTERNATE APPROACH]

APPROACH-1 :

(c) A minimization strategy like K-Means can be applied here

① Complete matrix A with random values to get $\hat{A}$

② Using this matrix $\hat{A}$ compute [B]

③ Use the matrix [B] to compute best $\hat{A}$ that minimizes $\|\hat{A} - B\|_F$

So, step-2 involves finding one amongst nearest psd B that minimizes the frobenius norm.

https://www.sciencedirect.com/science/article/pii/0024379588902236

By reference :

Which says nearest symmetric psd of A is

$$\left[\frac{B+H}{2}\right] \quad \text{where} \quad B = \frac{A+A^T}{2} \quad \text{as} \quad A = A^T \Rightarrow B = A$$

and performing a polar decomposition on B

gives $B = UH \rightarrow (U^TU = I,\ H = H^T \geq 0)$

OBSERVATIONS :

① In practice, as I observed literature there are a lot of interesting things on matrix completion under various assumptions on structure of the matrix and the number of non-missing values in $|\Omega|$

②

As per my intuition, the approach mentioned above and below can do the following :

1) Find $\hat{A}$ (a psd) if possible 2) Find $\hat{A}$ nearest one to any psd minimizing Frobenius Norm

Proof

Let X be psd from facts of frobenius norm $\|S+K\|_F^2 = \|S\|_F^2 + \|K\|_F^2$

if $S = S^T$, $K = -K^T$

As A is symmetric, $B = \frac{A+A^T}{2} = A$

$$\Rightarrow \quad \|A - X\|_F^2 = \|B - X\|_F^2$$

So, it reduces to approximating B. Let $B = Z \Lambda Z^T$

where $Z^T Z = I$ and $\Lambda = \text{diag}(\lambda_i)$ → spectral decomposition

Let $Y = Z^T X Z$ then

$$\|B - X\|_F^2 = \|\Lambda - Y\|_F^2$$

$$= \sum_{i \neq j} y_{ij}^2 + \sum_i (\lambda_i - y_{ii})^2$$

$$\geq \sum_{\lambda_i < 0} (\lambda_i - y_{ii})^2 \geq \sum_{\lambda_i < 0} \lambda_i^2$$

Since $y_{ii} \geq 0$ because Y is psd. This lower bound is attained uniquely for the matrix $Y = \text{diag}(d_i)$ where

$$d_i = \begin{cases} \lambda_i & \text{if } \lambda_i \geq 0 \\ 0 & \lambda_i < 0. \end{cases}$$

$$\Rightarrow \quad X_F = Z \,\text{diag}(d_i) Z^T \Rightarrow \frac{B+H}{2} \text{ as } H = Z \,\text{diag}(|\lambda_i|) Z^T$$

STEP-3 : After finding the nearest psd. (X) from $\hat{A}$

We use the psd to estimate best values for parameters of missing values. minimizing frobenius norm.

Expanding the norm we see

$$(w - X_{13})^2 + [\text{+ve}]$$

↑ Minimum qs attained when $w = X_{13}$ for this (A)

So new matrix $\hat{A}$ is constructed with $w = X_{13}$ from best psd chosen

A sample code to check whether $\begin{bmatrix} 1 & 2 \\ 2 & ? \end{bmatrix}$ Converges to $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ rank 1 psd

Initialized with (1)

```
import pandas as pd
import numpy as np
import scipy.linalg as sp

A = np.array([[1.0,2.0],[2.0,-1.0]])
B = np.array([[1,2],[2,4]])

iterations = 0
while iterations<1000:
    psd = A
    cov_A = np.dot(A.transpose(), A)
    print(cov_A)
    sq_A = sp.sqrtm(cov_A)
    print(sq_A)
    psd = 0.5*(psd + sq_A)
    print(psd)
    A[1][1] = psd[1][1]
    print(A)
    iterations = iterations + 1
    print('----------------------------------------------')
```

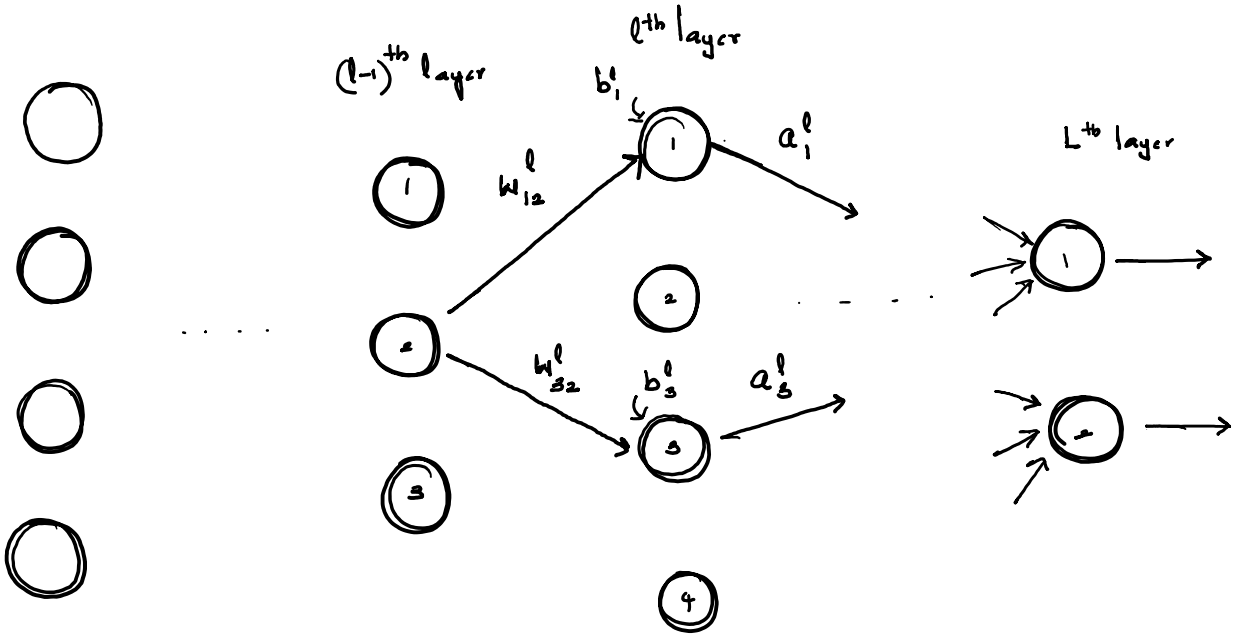
```
ITERATION 999
[[ 5. 10.]
 [10. 20.]]
[[1. 2.]
 [2. 4.]]
[[1. 2.]
 [2. 4.]]
[[1. 2.]
 [2. 4.]]
```

Rank 1 approximation for rank 2 matrix np.array([[1.0,2.0],[2.0,-1.0]])

a) BACKPROPAGATION ALGORITHM → feed forward neural network

① Soft plus activation function $[s(x) = \ln(1+e^x)]$ (SMOOTHES RELU's)

② Squared loss function $s'(x) = \frac{e^x}{1+e^x} = \sigma(x)$



[Weighted input]

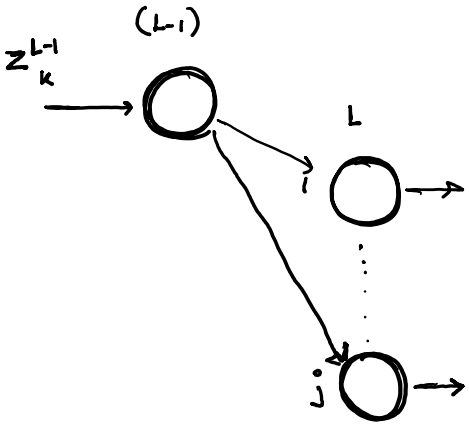
Activation can be given by $a_j^l = s(z_j) \rightarrow$ [SOFT PLUS ACTIVATION]

$$= s\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$$

$C(w,b) = \frac{1}{2}\| y - a(x,w,b)\|^2$ ↙ For a single datapoint

$$\frac{\partial c}{\partial z_j^L} = \frac{\partial c}{\partial a_j^L} \cdot \frac{\partial a_j^L}{\partial z_j^L} = -(y - a_j^L)\frac{\partial}{\partial z_j^L} s(z_j^L) \quad [\text{LAST LAYER}]$$

$$= -(y - a_j^L)\,\sigma(z_j^L) = \delta_j^L$$



$$\frac{\partial C}{\partial z_k^{L-1}} = \sum_j^{\cdot} (a_j^L - y_j) \frac{\partial a_j^L}{\partial z_k^{L-1}}$$

$$= \sum_j (a_j^L - y_j) \frac{\partial}{\partial z_k^{L-1}} \left(\sigma(z_j^L) \right)$$

$$= \sum_j (a_j^L - y_j) \left[\frac{\partial}{\partial z_j^L} \sigma(z_j^L) \right] \left[\frac{\partial z_j^L}{\partial z_k^{L-1}} \right]$$

$$= \sum_j (a_j^L - y_j) \sigma(z_j^L) \frac{\partial}{\partial z_k^{L-1}} \left(\sum_{k'} w_{jk'}^L a_{k'}^{L-1} + b_j^L \right)$$

$$= \sum_j (a_j^L - y_j) \sigma(z_j^L) \left[w_{jk}^L \frac{\partial}{\partial z_k^{L-1}} a_k^{L-1} \right]$$

$$= \sum_j \overbrace{(a_j^L - y_j) \sigma(z_j^L)}^{\delta_j^L} \sigma(z_k^{L-1}) w_{jk}^L$$

$$= \left((\delta^L)^T W_{*k}^L \right) \sigma(z_k^{L-1})$$

$\Rightarrow$ This can be written as.

$$\delta^{L-1} = \left((\delta^L)^T W^L \right)^T \sigma(z^{L-1})$$

for some $l < L$ we get

$$\delta^l = \left((\delta^{l+1})^T W^{l+1} \right)^T \sigma(z^{L-1})$$

As $C(w,b) \rightarrow \frac{\partial C}{\partial w}, \frac{\partial C}{\partial b}$ can be written as:

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial b_j^l}$$

$$= \frac{\partial C}{\partial z_j^l} \left(\frac{\partial}{\partial b_j^l} \left(\sum_1 w_{jk}^l a_k^{l-1} + b_j^l \right) \right)$$

$$= \delta_j^l \quad \text{(As assumed earlier)}$$

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \cdot \frac{\partial}{\partial w_{jk}^l} \left(\sum_1 w_{jk}^l a_k^{l-1} + b_j^l \right)$$

$$= \left(\delta_j^l \right) \left(a_k^{l-1} \right)$$

BACKPROPAGATION ALGORITHM:

① $\delta_j^L = -(y_j - a_j^L)\, \sigma(z_j^L)$ for the last layer

② $\delta^l = \left((\delta^{l+1})^T w^{l+1} \right)^T \sigma(z^l)$ for layers from $l = L-1$ to 1

③ Gradient descent

$$b_j^l = b_j^l - \gamma \frac{\partial C}{\partial b_j^l} = b_j^l - \gamma \delta_j^l$$

$$w_{jk}^l = w_{jk}^l - \gamma \frac{\partial C}{\partial w_{jk}^l} = w_{jk}^l - \gamma \delta_j^l a_k^{l-1}$$

(b) Conditional mixture model for regression task

i.e for data points $x^{(1)}, x^{(2)}, \ldots x^{(n)} \in \mathbb{R}^n$

corresponding labels $y^{(1)}, y^{(2)} \ldots y^{(n)} \in \mathbb{R}$

$$p(y|x, \lambda, \theta_1, \ldots \theta_k) = \sum_{k=1}^{K} \lambda_k \, p(y|x, \theta_k) \cdot$$

Here $p(y|x, \theta_k) = \mathcal{N}(y; NN_k(x|\theta_k))$ — $NN_k(x)$ parameter are given by θ_k

Mean and variance of k^{th} normal — returns a pair of outputs (μ_k, σ_k^2)

OBSERVATIONS:

1) Function $f(x)$ to determine label y is expressed as

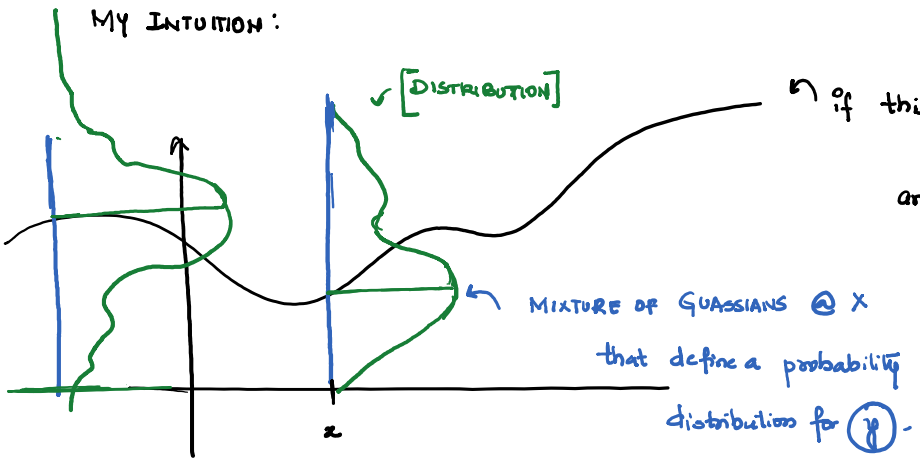
[sum of K-normal distributions]

2) Corresponding normal distributions change for each value of x

i.e (means, variances) of normal distributions are different for each x

and are evaluated from k corresponding neural networks learnt.

MY INTUITION:



if this true function for f(x)

and if we observe

$y = f(x) + \epsilon$ (error)

Here as μ_k, σ_k^2 come from $NN_k(x)$ different distributions of labels y

can be approximated locally.

Conditional mixtures : $\Theta = \{\lambda_1, \lambda_2, \lambda_3 \cdots \lambda_K, \theta_1, \theta_2, \theta_3, \cdots \theta_K\}$

$$P(y|x,\theta) = \sum_{k=1}^{K} \lambda_k \, P(y|x,\theta_k), \quad P(y|x,\theta_k) = \mathcal{N}\left(y; NN_k(x|\theta_k)\right)$$

$q \quad \sum \lambda_k = 1$

Considering data likelihood.

$$= \prod_{i=1}^{n} P(y^{(i)}|x^{(i)},\theta) = \prod_{i=1}^{n} \sum_{k} \lambda_k P(y^{(i)}|x^{(i)};\theta_k)$$

Considering log likelihood.

$$= \sum_{i=1}^{n} \log \sum_{k} \lambda_k P(y^{(i)}|x^{(i)},\theta_k)$$

CONCAVE → [Applying JENSEN'S INEQUALITY]

$$= \sum_{i=1}^{n} \sum_{k} q_i(k) \log\left(\frac{\lambda_k P(y^{(i)}|x^{(i)},\theta_k)}{q_i(k)}\right)$$

As mentioned in Andrew Ng's EM notes we can solve this by repeatedly constructing lower bound on l [E-STEP] and then optimize that lower bound [M-step].

So, to make the inequality tight for a current guess θ of params

↳ So JENSEN'S INEQUALITY should hold with equality:

only when expectation is taken over a constant valued RV

$$\Rightarrow \quad \frac{P\left(y^{(i)} \mid x^{(i)}, k=k, \theta_k\right)}{q_i(k)} = c$$

$$\Rightarrow \quad q_i(k) \propto P\left(y^{(i)} \mid x^{(i)}, k=k, \theta_k\right) \quad \text{as} \quad \sum_k q_i(k) = 1$$

We can write: $q_i(k) = \dfrac{P\left(y^{(i)} \mid x^{(i)}, k=k, \theta_k\right)}{\sum_k P\left(y^{(i)} \mid x^{(i)}, \theta_k\right)}$

[POSTERIORS]

M-STEP:

Here we perform the following update with q's fixed and maximizing the objective over θ

$$\theta^{t+1} \in \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \sum_k q_i^{t+1}(k) \log\left(\frac{P\left(y^{(i)} \mid x^{(i)}, k=k \mid \theta_k\right)}{q_i^{t+1}(k)}\right)$$

As $P\left(y^{(i)} \mid x^{(i)}, k=k \mid \theta_k\right) = \lambda_k \mathcal{N}\left(y_i; NN_k(x \mid \theta_k)\right)$

θ_k are parameters corresponding to neural network

Inorder to update the parameters θ_k of neural networks in the M step

We follow a two step procedure

1) In the first step, we assume μ_k, σ_k^2 for each guassian of the mixture.

Follow the EM procedure to learn the best μ_k, σ_k^2 from the M-step.

2) Use these values of μ_k, σ_k^2 as the required outputs of K neural networks and perform a backpropagation algorithm to update weights and biases of neural network.

When we use μ_k, σ_k^2 as parameters, M-step results similar to the Guassian Mixture Model used in the class

$$\text{i.e } \sum_{i=1}^{n} \sum_{k} q_i(k) \log \left(\frac{\lambda_k \mathcal{N}(y^{(i)}; \mu_k, \sigma_k^2)}{q_i(k)} \right)$$

Taking derivative wrt to μ_k gives us.

$$\frac{\partial l}{\partial \mu_k} = \sum_{i=1}^{n} q_i(k) \frac{\frac{\partial}{\partial \mu_k} \mathcal{N}(y^{(i)}; \mu_k, \sigma_k^2) \cdot \frac{\lambda_k}{q_i(k)}}{\frac{\lambda_k \mathcal{N}(y^{(i)}; \mu_k, \sigma_k^2)}{q_i(k)}}$$

$$\mathcal{N}\left(y^{(i)};\ \mu_k, \sigma_k^2\right) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y^{(i)}-\mu_k}{\sigma_k}\right)^2\right)$$

$$\frac{\partial}{\partial \mu_k}\mathcal{N}\left(y^{(i)};\ \mu_k, \sigma_k^2\right) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{y^{(i)}-\mu_k}{\sigma_k}\right)^2\right) \frac{\partial}{\partial \mu_k}\left(-\frac{1}{2}\left(\frac{y^{(i)}-\mu_k}{\sigma_k}\right)^2\right)$$

$$= \mathcal{N}\left(y^{(i)};\ \mu_k, \sigma_k^2\right)\left(\frac{y^{(i)}-\mu_k}{\sigma_k}\right)$$

So, $$\frac{\partial l}{\partial \mu_k} = \sum_{i=1}^{n} q_i^{t+1}(k) \frac{\mathcal{N}\left(y^{(i)};\ \mu_k, \sigma_k^2\right)\left(\frac{y^{(i)}-\mu_k}{\sigma_k}\right)}{\mathcal{N}\left(y^{(i)};\ \mu_k, \sigma_k^2\right)} = 0$$

$$\Rightarrow \mu_k^{t+1} = \frac{\sum_{i=1}^{n} q_i^{t+1}(k)\, y^{(i)}}{\sum_{i=1}^{n} q_i^{t+1}(k).}$$ similar to GMM in class

Similarly for σ_k^2 shall be and λ_k which can be obtained by maximizing with the use of lagrange multiplier

$$\sigma_k^{2\,t+1} = \frac{\sum_{i=1}^{n} q_i(k)\left(y^{(i)}-\mu_k^{t+1}\right)^2}{\sum_{i=1}^{n} q_i(k).}$$

$$\lambda_k = \frac{1}{N}\sum_{i=1}^{N} q_i^t(k).$$

After each iteration of M step we take these μ_k, σ_k^2 values as corresponding y's for Neural networks

Using the backpropagation algorithm we compute changes in

$\Theta_k = \{ w^l_{jk}, b^l_j \}$ of the neural network.

b) Here in a way we are trying approximate locally.

i.e $P(y|x, \Theta)$ depends on the current context

↳ DIRECTION TOWARDS [MIXTURES OF EXPERTS]

Based on the above observations, I looked up literature about CONDITIONAL MIXTURES and MIXTURE OF EXPERTS.

These are currently used in various scenarios of solving classification / regression problems

i) Style in Handwritten Character recognition

ii) Dialect / Accent in speech recognition.

They allocate tasks to experts in an INPUT DEPENDENT way.

As they compose each problem into smaller subproblems where each subproblem is solved by a specific expert

STRUCTURE OF NEURAL NETWORK :

OBSERVATIONS :

1) We are converting R^n input to R^2 output

n - inputs $\rightarrow [x^{(i)}]$ and 2 - outputs $\rightarrow [\mu_k, \sigma_k^2]$

2) Assuming equivalent structure for all the K - neural networks

3) σ_k^2 should always be (+ve) [SELECTION OF NODES APPROPRIATELY]

4) Output unit for σ_k^2 should be a SIGMOID unit

5) Output for μ_k should be a simple function on inputs.

approximating to $\boxed{W^T x + b}$

↳ No thresholding should be applied.

for this output

3)

PROBLEM 3 - ROLLING DICE

(a) ROLLING DICE :

Generated by picking one of two dice (A or B)

[Defined by some probability distribution]

4-sided die rolls (1-4)

4 sides of a die

It depends on two things

① Type of die

② Outcome of die

Consider joint probability space of type of die and outcome of die given by parameters $\theta_1, \theta_2, \ldots, \theta_8$. i.e

	O=1	O=2	O=3	O=4	
D=A	θ_1	θ_2	θ_3	θ_4	s.t $\sum_i \theta_i = 1$
D=B	θ_5	θ_6	θ_7	θ_8	

Considering data likelihood in this probability space

$$P(D|\theta) = \theta_1^{\#(O=1, D=A)} \; \theta_2^{\#(O=2, D=A)} \; \ldots \; \theta_8^{\#(O=4, D=B)}$$

$$= \left(\theta_1 \, \theta_2^2 \, \theta_3 \, \theta_4\right)\left(\theta_5^2 \, \theta_6 \, \theta_7^2 \, \theta_8^4\right)$$

$$\text{Log likelihood} = \left[\log\theta_1 + 2\log\theta_2 + \log\theta_3 + \log\theta_4 + 2\log\theta_5 + \log\theta_6 + 2\log\theta_7 + 4\log\theta_8\right]$$

Maximizing LL s.t $\sum_1^8 \theta_i = 1$ gives us. as θ_i's are always positive

$$L(\boldsymbol{\theta}, \lambda) = \left[\log\theta_1 + 2\log\theta_2 + \log\theta_3 + \log\theta_4 + 2\log\theta_5 + \log\theta_6 + 2\log\theta_7 + 4\log\theta_8\right] - \lambda\left[\sum_1^8 \theta_i - 1\right] = 0.$$

$$\frac{\partial L}{\partial \theta_1} = \frac{1}{\theta_1} - \lambda, \quad \frac{\partial L}{\partial \theta_2} = \frac{2}{\theta_2} - \lambda, \quad \frac{\partial L}{\partial \theta_3} = \frac{1}{\theta_3} - \lambda, \quad \frac{\partial L}{\partial \theta_8} = \frac{4}{\theta_8} - \lambda$$

$$\frac{\partial L}{\partial \theta_4} = \frac{1}{\theta_4} - \lambda, \quad \frac{\partial L}{\partial \theta_5} = \frac{2}{\theta_5} - \lambda, \quad \frac{\partial L}{\partial \theta_6} = \frac{1}{\theta_6} - \lambda, \quad \frac{\partial L}{\partial \theta_7} = \frac{2}{\theta_7} - \lambda$$

Forming Lagrangian dual form gives:

$$\theta_1 = \frac{1}{\lambda}, \quad \theta_2 = \frac{2}{\lambda}, \quad \theta_3 = \frac{1}{\lambda}, \quad \theta_4 = \frac{1}{\lambda}, \quad \theta_5 = \frac{2}{\lambda}, \quad \theta_6 = \frac{1}{\lambda}, \quad \theta_7 = \frac{2}{\lambda}, \quad \theta_8 = \frac{4}{\lambda}$$

$$g(\lambda) = \lambda - (N) + \log\left(\frac{2^2 \cdot 2^2 \cdot 4^4}{\lambda^N}\right)$$

$$= \lambda - N - N\log\lambda + \log C$$

$$\frac{dg}{d\lambda} = 1 - \frac{N}{\lambda} = 0 \Rightarrow \lambda = N$$

So, $\theta_1 = \frac{1}{N}, \quad \theta_2 = \frac{2}{N}, \quad \theta_3 = \frac{1}{N}, \quad \theta_4 = \frac{1}{N}$

$\theta_5 = \frac{2}{N}, \quad \theta_6 = \frac{1}{N}, \quad \theta_7 = \frac{2}{N}, \quad \theta_8 = \frac{4}{N}$

} MLE estimators for parameters where $N = 14$

(b) Observations generated by rolling k 6-sided dice for some $k \in \{1, 2, 3\}$

$k=2 \rightarrow$ die one is 5, die two is 6 $\rightarrow$ (OBSERVATION = 11)

Probability of observing n pips denoted by θ_n for $n \in \{1, 2, \cdots 6\}$ is same

k is unknown and must be estimated from data.

OBSERVATIONS :

$\rightarrow \sum \theta_n = 1$ for $n = \{1, 2, 3, \cdots 6\}$

For binary variable in case of coin flips

we considered $P(X=H) = \theta$ and $P(X=T) = 1-\theta$ and estimated

Data likelihood $\rightarrow$ Log likelihood.

Here as we are considering random variable X with 6 outcomes

$P(X=1) = \theta_1$, $P(X=2) = \theta_2$, $\cdots$ $P(X=3) = \theta_6$.

This probability distribution can be expressed as follows

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{n=1}^{6} \theta_n^{x_n} \quad \text{s.t} \quad \sum \theta_n = 1$$

$\boldsymbol{\theta} \rightarrow \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 \rangle$, $\mathbf{x} \rightarrow \langle 0, 0, 1, 0, 0, 0 \rangle$

Let us consider three different models M_1, M_2, M_3 corresponding to $k = 1, 2, 3$

Q find data likelihood for all the models and pick the one with maximum likelihood for the observed data. [DECISION ON k]

Considering model M_1 where only a single dice is rolled.

As we see the observed sum is $\boxed{>6}$ which is not possible with single dice

Data likelihood for $M_1 \rightarrow \boxed{0}$

For model M_2 which considers two dice are rolled

combinations of die outcomes possible for each observed sum are

① 5 – (1,4), (2,3) [Since identical die (No permutations are required)]

② 9 – (3,6), (4,5)

② 10 – (4,6), (5,5) So, the data likelihood here shall be.

⑤ 12 – (6,6)

Considering probability distribution for a single dice $\rightarrow (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$

When two dice are rolled $\rightarrow$ Probabilities for each outcome [IDENTICAL DICE]

1 $\rightarrow 0$

2 $\rightarrow \theta_1^2$

3 $\rightarrow 2\theta_1\theta_2$

4 $\rightarrow$ (1,3), (2,2) $\Rightarrow 2\theta_1\theta_3 + \theta_2^2$

5 $\rightarrow$ (1,4), (2,3) $\Rightarrow 2(\theta_1\theta_4 + \theta_2\theta_3)$

6 $\rightarrow$ (1,5), (2,4), (3,3) $\Rightarrow 2(\theta_1\theta_5 + \theta_2\theta_4) + \theta_3^2$

7 $\rightarrow$ (1,6), (2,5), (3,4) $- 2(\theta_1\theta_6 + \theta_2\theta_5 + \theta_3\theta_4)$

8 $\rightarrow$ (2,6), (3,5), (4,4) $- 2(\theta_2\theta_6 + \theta_3\theta_5) + \theta_4^2$

9 $\rightarrow$ (3,6), (5,4) $- 2(\theta_3\theta_6 + \theta_5\theta_4)$

10 $\rightarrow$ (4,6), (5,5) $- 2\theta_4\theta_6 + \theta_5^2$

11 $\rightarrow$ (5,6) $- 2\theta_5\theta_6$

12 $\rightarrow$ (6,6) $\rightarrow \theta_6^2$

Data likelihood for model (M_2)

Observed sums → 5, 9, 10, 12, 12, 12, 9, 12, 12, 10.

$$\max_{\theta_i\text{'s}} \left[2(\theta_1\theta_4+\theta_2\theta_3)\right]\left[2(\theta_3\theta_6+\theta_5\theta_4)\right]^2\left[2\theta_4\theta_6+\theta_5^2\right]^2\left[\theta_6^2\right]^5 \quad \text{s.t} \sum_i \theta_i = 1$$

Considering log likihood of the data

$$\Rightarrow \log\left(2(\theta_1\theta_4+\theta_2\theta_3)\right) + 2\log\left(2(\theta_3\theta_6+\theta_5\theta_4)\right) + 2\log\left(2\theta_4\theta_6+\theta_5^2\right) + 10\log\theta_6$$

$$\text{s.t} \sum \theta_i = 1.$$

$$\max \log(\theta_1\theta_4+\theta_2\theta_3) + 2\log(\theta_3\theta_6+\theta_5\theta_4) + 2\log(2\theta_4\theta_6+\theta_5^2) + 10\log\theta_6 + |C| \text{ st } \sum\theta_i = 1$$

Two approaches — LAGRANGIAN (OR) assume $\theta_1 = 1-(\theta_6+\theta_2+\theta_3+\theta_4+\theta_5)$

$\left(\sum \theta_i = 1\right)$

There does not seem a closed form solution for θ_i's

Next approaches

↳ Apply gradient ascent to find θ_i's maximumizing likelihood

$$\Rightarrow \log\left(\left[1-(\theta_6+\theta_2+\theta_3+\theta_4+\theta_5)\right]\theta_4+\theta_2\theta_3\right) + 2\log(\theta_3\theta_6+\theta_5\theta_4)$$

$$+ 2\log(2\theta_4\theta_6+\theta_5^2) + 10\log\theta_6.$$

$$\Rightarrow \log\left(1-\theta_6\theta_4-\theta_2\theta_4-\theta_3\theta_4-\theta_4^2-\theta_5\theta_4+\theta_2\theta_3\right) + 2\log(\theta_3\theta_6+\theta_5\theta_4)$$

$$+ 2\log(2\theta_4\theta_6+\theta_5^2) + 10\log\theta_6$$

$$\frac{\partial L}{\partial \theta_2} = \frac{\theta_3}{1 - \theta_6\theta_4 - \theta_2\theta_4 - \theta_3\theta_4 - \theta_4^2 - \theta_5\theta_4 + \theta_2\theta_3}, \quad \theta_2 = \theta_2 - \alpha \frac{\partial L}{\partial \theta_2}$$

$$\frac{\partial L}{\partial \theta_3} = \frac{\theta_2 - \theta_4}{(1 - \theta_6\theta_4 - \theta_2\theta_4 - \theta_3\theta_4 - \theta_4^2 - \theta_5\theta_4 + \theta_2\theta_3)} + \frac{2\theta_6}{\theta_3\theta_6 + \theta_5\theta_4}, \quad \theta_3 = \theta_3 - \alpha \frac{\partial L}{\partial \theta_3}$$

$$\frac{\partial L}{\partial \theta_4} = \frac{-\theta_6 - \theta_2 - \theta_3 - 2\theta_4 - \theta_5}{(1 - \theta_6\theta_4 - \theta_2\theta_4 - \theta_3\theta_4 - \theta_4^2 - \theta_5\theta_4 + \theta_2\theta_3)} + \frac{2\theta_5}{\theta_3\theta_6 + \theta_5\theta_4} + \frac{2\theta_6}{2\theta_4\theta_6 + \theta_5^2}$$

$$\left[\theta_4 = \theta_4 - \alpha \frac{\partial L}{\partial \theta_4}\right]$$

$$\frac{\partial L}{\partial \theta_5} = \frac{-\theta_4}{(1 - \theta_6\theta_4 - \theta_2\theta_4 - \theta_3\theta_4 - \theta_4^2 - \theta_5\theta_4 + \theta_2\theta_3)} + \frac{2\theta_4}{\theta_3\theta_6 + \theta_5\theta_4} + \frac{4\theta_5}{2\theta_4\theta_6 + \theta_5^2}$$

$$\left[\theta_5 = \theta_5 - \alpha \frac{\partial L}{\partial \theta_5}\right]$$

$$\frac{\partial L}{\partial \theta_6} = \frac{-\theta_4}{(1 - \theta_6\theta_4 - \theta_2\theta_4 - \theta_3\theta_4 - \theta_4^2 - \theta_5\theta_4 + \theta_2\theta_3)} + \frac{2\theta_3}{\theta_3\theta_6 + \theta_5\theta_4} + \frac{4\theta_4}{\theta_5^2 + 2\theta_4\theta_6} + \frac{1}{\theta_6}$$

$$\left[\theta_6 = \theta_6 - \alpha \frac{\partial L}{\partial \theta_5}\right]$$

For model M_3 we take $k=3$ so possibilities of sums. $[X_1+X_2+X_3]$

$S=1 \rightarrow 0$ [PARTITIONS OF SIZE (3) and corresponding probabilities]

$S=2 \rightarrow 0$

$S=3 \rightarrow \theta_1^3$

$S=4 \rightarrow (2,1,1) = \frac{3!}{2!}\theta_2\theta_1^2$

$S=5 \rightarrow (3,1,1),(2,2,1) = \frac{3!}{2!}\theta_3\theta_1^2 + \frac{3!}{2!}\theta_2^2\theta_1$

$S=6 \rightarrow (4,1,1),(3,2,1),(2,2,2) = \frac{3!}{2!}\theta_4\theta_1^2 + 3!\,\theta_3\theta_2\theta_1 + \theta_2^3$

$S=7 \rightarrow (5,1,1),(4,2,1),(3,3,1),(3,2,2)$

$= \frac{3!}{2!}\theta_5\theta_1^2 + 3!\,\theta_4\theta_2\theta_1 + \frac{3!}{2!}\theta_3^2\theta_1 + \frac{3!}{2!}\theta_3\theta_2^2$

$S=8 \rightarrow (6,1,1),(5,2,1),(4,3,1),(4,2,2),(3,3,2)$

$= \frac{3!}{2!}\theta_6\theta_1^2 + 3!\,\theta_5\theta_2\theta_1 + 3!\,\theta_4\theta_3\theta_1 + \frac{3!}{2!}\theta_4\theta_2^2 + \frac{3!}{2!}\theta_5^2\theta_2$

$S=9 \rightarrow ,(6,2,1),(5,3,1),(5,2,2),(4,4,1),(4,3,2),(3,3,3)$

$3!\,\theta_6\theta_2\theta_1 + 3!\,\theta_5\theta_3\theta_1 + \frac{3!}{2!}\theta_5\theta_2^2 + \frac{3!}{2!}\theta_4^2\theta_1 + 3!\,\theta_4\theta_3\theta_2 + \theta_3^3$

$S=10 \rightarrow (6,3,1),(6,2,2),(5,4,1),(5,3,2),(4,4,2),(4,3,3)$

$3!\,\theta_6\theta_3\theta_1 + \frac{3!}{2!}\theta_6\theta_2^2 + 3!\,\theta_5\theta_4\theta_1 + 3!\,\theta_5\theta_3\theta_2 + \frac{3!}{2!}\theta_4^2\theta_2 + \frac{3!}{2!}\theta_4\theta_3^2$

$S=11 \rightarrow (6,4,1),(6,3,2),(5,5,1),(5,4,2),(5,3,3),(4,4,3)$

$3!\,\theta_6\theta_4\theta_1 + 3!\,\theta_6\theta_3\theta_2 + \frac{3!}{2!}\theta_5^2\theta_1 + 3!\,\theta_5\theta_4\theta_2 + \frac{3!}{2!}\theta_5\theta_3^2 + \frac{3!}{2!}\theta_4^2\theta_3$

$S=12 \rightarrow (6,5,1),(6,4,2),(6,3,3),(5,5,2),(5,4,3),(4,4,4)$

$3!\,\theta_6\theta_5\theta_1 + 3!\,\theta_6\theta_4\theta_2 + \frac{3!}{2!}\theta_6\theta_3^2 + \frac{3!}{2!}\theta_5^2\theta_2 + 3!\,\theta_5\theta_4\theta_3 + \theta_4^3$

$s = 13 \rightarrow (6,6,1), (6,5,2), (6,4,2), (5,5,3), (5,4,4)$

$$\frac{3!}{2!}\theta_6^2\theta_1 + 3!\,\theta_6\theta_5\theta_2 + 3!\,\theta_6\theta_4\theta_3 + \frac{3!}{2!}\theta_5^2\theta_3 + \frac{3!}{2!}\theta_5\theta_4^2$$

$s = 14 \rightarrow (6,6,2), (6,5,3), (6,4,4), (5,5,4)$

$$\frac{3!}{2!}\theta_6^2\theta_2 + \frac{3!}{2!}\theta_6\theta_3^2 + \frac{3!}{2!}\theta_6\theta_4^2 + \frac{3!}{2!}\theta_5^2\theta_4$$

$s = 15 \rightarrow (6,6,3), (6,5,4), (5,5,5)$

$$= \frac{3!}{2!}\theta_6^2\theta_3 + 3!\,\theta_6\theta_5\theta_4 + \theta_5^3$$

$s = 16 \rightarrow (6,6,4), (6,5,5) = \frac{3!}{2!}\theta_6^2\theta_4 + \frac{3!}{2!}\theta_6\theta_5^2$

$s = 17 \rightarrow (6,6,5) = \frac{3!}{2!}\theta_6^2\theta_5$

$s = 18 \rightarrow (6,6,6) = \theta_6^3$

Data likelihood for observations

1 (5), 2 (9), 2 (10), 5 (12)

$$\max_{\theta_i's} \left(3\theta_3\theta_1^2 + 3\theta_2^2\theta_1\right)\left(6\theta_6\theta_2\theta_1 + 6\theta_5\theta_3\theta_1 + 3\theta_5\theta_2^2 + 3\theta_4^2\theta_1 + 6\theta_4\theta_3\theta_2 + \theta_3^3\right)^2$$

$$\left(6\theta_6\theta_3\theta_1 + 3\theta_6\theta_2^2 + 6\theta_5\theta_4\theta_1 + 6\theta_5\theta_3\theta_2 + 3\theta_4^2\theta_2 + 3\theta_4\theta_3^2\right)^2$$

$$\left(6\theta_6\theta_5\theta_1 + 6\theta_6\theta_4\theta_2 + 3\theta_6\theta_3^2 + 3\theta_5^2\theta_2 + 6\theta_5\theta_4\theta_3 + \theta_4^3\right)^5 \quad \sum_i^4 \theta_i = 1$$

(b)

(2) Here θ's can be assumed to be part of $p(x|\theta) = \prod \theta_k^{x_k}$

$\theta = \langle \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6 \rangle$, $x = \langle 0, 1, 0, 0, 0, 0 \rangle$

Now based on this data likelihood for dice

$$\prod_{k=1}^{K} \theta_k^{m_k} \quad \text{where} \quad m_k = \sum_{n}^{} x_{nk}$$

It can be assumed m_i's jointly form a [Multinomial Distribution]

A good choice of prior for this distribution of the same form as data likelihood. → DIRICHLET DISTRIBUTION

$$P(\theta | D, \alpha) \propto P(D|\theta) P(\theta|\alpha)$$

$$\propto \prod_{k=1}^{K} \theta_k^{\alpha_k + m_k - 1}$$

(c) Data generated by → Flip a coin with bias (b)

(b) HEADS TAILS (1-b)

D1 - TWICE D2 - TWICE

OBSERVATION : Sum of LOADED DIE

ESTIMATE

(1) Bias of the coin

(2) Probability distribution of the outcomes of each of the loaded dice

$P(C=H) = b$, $P(C=T) = (1-b)$

$P(O|C=H)$, $P(O|C=T)$ are the two other distributions to be considered

So, $P(O) = P(O|C=H)\,P(C=H) + P(O|C=T)\,P(C=T)$

Now, For each dice let us define corresponding probability distributions

for $D_1 \rightarrow P(X|D_1) = \prod_{k=1}^{6} \theta_{1k}^{x_k}$ for $D_2 \rightarrow P(X|D_2) = \prod_{k=1}^{6} \theta_{2k}^{x_k}$

such that $\sum_1^6 \theta_{1k} = \sum_1^6 \theta_{2k} = 1$

Probability distributions of outcomes : [d ← dice no]

$1 \rightarrow 0$

$2 \rightarrow \theta_{d1}^2$

$3 \rightarrow 2\theta_{d1}\theta_{d2}$

$4 \rightarrow 2\theta_{d1}\theta_{d3} + \theta_{d2}^2$

$5 \rightarrow 2(\theta_{d1}\theta_{d4} + \theta_{d2}\theta_{d3})$

$6 \rightarrow 2\theta_{d1}\theta_{d5} + 2\theta_{d2}\theta_{d4} + \theta_{d3}^2$

$7 \rightarrow 2(\theta_{d1}\theta_{d6} + \theta_{d2}\theta_{d5} + \theta_{d3}\theta_{d4})$

$8 \rightarrow 2\theta_{d2}\theta_{d6} + 2\theta_{d3}\theta_{d5} + \theta_{d4}^2$

$9 \rightarrow 2(\theta_{d3}\theta_{d6} + \theta_{d5}\theta_{d4})$

$10 \rightarrow 2\theta_{d4}\theta_{d6} + \theta_{d5}^2$

$11 \rightarrow 2\theta_{d5}\theta_{d6}$

$12 \rightarrow \theta_{d6}^2$

Now considering data likelihood for this model. $\theta \rightarrow \{b, \theta_{di}\text{'s}\}$

$$\prod_{i=1}^{n} P(O^{(i)} \mid \theta)$$

$$= \prod_{i=1}^{n} \left[b \left[P(O^{(i)} \mid c=H) \right] + (1-b) \left[P(O^{(i)} \mid c=T) \right] \right]$$

log likelihood of data.

$$\Rightarrow \sum_{i=1}^{n} \log \left(b \left[P(O^{(i)} \mid c=H) \right] + (1-b) \left[P(O^{(i)} \mid c=T) \right] \right)$$

$$\Rightarrow \sum_{i=1}^{n} \log \left(\sum_{c \in \{H,T\}} P(c=c)\, P(O^{(i)} \mid c=c) \right)$$

Concave

↳ Applying JENSEN's inequality gives us

$$\geq \sum_{i=1}^{N} \sum_{c \in \{H,T\}} q_i(c) \log \frac{P(O^{(i)}, c=c \mid \theta)}{q_i(c)}.$$

Applying the block co-ordinate ascent strategy described by EM algorithm

We start with initializations for b and θ_{di}'s

$[\, b = \tfrac{1}{2},\ \theta_{di} \rightarrow \tfrac{1}{6} \,]$ ~ Possible option

E-STEP:

$$q_i^{t+1}(c) = P\left(C=c \mid O^{(i)}, \theta^t\right)$$

M-STEP :

$$\theta^{t+1} \rightarrow \underset{\theta}{\arg\max} \sum_{i=1}^{n} \sum_{c \in \{H,T\}} q_i^{t+1}(c) \log \frac{P\left(O^{(i)}, C=c \mid \theta^t\right)}{q_i^{t+1}(c)}.$$

Parameters in θ are b, θ_{di}'s

[COIN BIAS] [DICE BIAS]

$$= \sum_{i=1}^{n} \sum_{c \in \{H,T\}} q_i^{t+1}(c) \log \left[\frac{P(C=c)\left(P\left(O^{(i)} \mid C=c\right)\right)}{q_i^{t+1}(c)} \right]$$

$$= \sum_{i=1}^{n} q_i^{t+1}(C=H) \log \left(\frac{b\, P\left(O^{(i)} \mid C=H\right)}{q_i^{t+1}(c)} \right) + q_i^{t+1}(C=T) \log \left(\frac{(1-b)\, P\left(O^{(i)} \mid C=T\right)}{q_i^{t+1}(C=T)} \right)$$

$$\frac{\partial l}{\partial b} = \sum_{i=1}^{n} \left[\frac{q_i^{t+1}(C=H)}{b} - \frac{q_i^{t+1}(C=T)}{(1-b)} \right] = 0. \qquad \sum_{c \in \{H,T\}} q_i^{t+1}(C=c) = 1$$

$$= \frac{\sum_i q_i^{t+1}(C=H)}{b} - \frac{\sum_i \left(1-b\right) q_i^{t+1}(C=T)}{1-b} = 0.$$

$$\sum_{i=1}^{n} q_i^{t+1}(c=H) - b\left(\underbrace{\sum_{i=1}^{n} q_i^{t+1}(c=H) + \sum_{i=1}^{n} q_i^{t+1}(c=T)}_{n}\right) = 0$$

$$b = \frac{1}{n} \sum_{i=1}^{n} q_i^{t+1}(c=H)$$

For each parameter θ_{dk} ← Dice 1 → $\{\theta_{11}, \theta_{12}, \theta_{13} \cdots \theta_{1k}\}$

Dice 2 → $\{\theta_{21}, \theta_{22}, \theta_{23} \cdots \theta_{2k}\}$

Each of the parameters $\sum_k \theta_{1k} = 1$, $\sum_k \theta_{2k} = 1$

Can be solved applying lagrange multipliers for θ_{1k}'s and θ_{2k}'s separately

(or) we could write one of θ_{rk}'s and θ_{2k}'s in terms of others

For both the approaches there is NO CLOSED FORM solution

So, GRADIENT ASCENT can be applied to learn parameters corresponding to θ_{dk}'s

$$\Rightarrow \sum_{i=1}^{n} \sum_{c \in \{H,T\}} q_i^{t+1}(c) \log\left(\frac{P(c=c)\, P(o^{(i)}, c=c|\theta)}{q_i^{t+1}(c)}\right)$$

$$\frac{\partial l}{\partial \theta_{1k}} = \sum_{i=1}^{n} q_i^{t+1}(c=H) \frac{\partial}{\partial \theta_{1k}} \log\left(\frac{b\, P(o^{(i)}, c=H)}{q_i^{t+1}(c)}\right)$$

dice 1 ↑

$$\Rightarrow \sum_{i=1}^{n} q_i^{t+1}(c=H) \frac{\frac{\partial}{\partial \theta_{1k}}\left[P(o^{(i)}, c=H)\right] \cdot b}{b\, P(o^{(i)}, c=H)}$$

So, $\theta_{1k} = \theta_{1k} + \alpha \underbrace{\frac{\partial}{\partial \theta_{1k}} f(\theta)}$

Similarly for $\theta_{2k} = \theta_{2k} + \alpha \underbrace{\frac{\partial}{\partial \theta_{2k}} f(\theta)}$

$$\hookrightarrow \sum_{i=1}^{n} q_i^{t+1}(c=T) \frac{\frac{\partial}{\partial \theta_{2k}}\left[P(o^{(i)}, c=T\right] \cdot (1-b)}{(1-b)\, P(o^{(i)}, c=T)}$$

θ^{t+1} shall be the parameters that result after performing gradient ascent

Q-LEARNING : BACKGROUND :

↳ Representation of a problem

↳ Divide problem into situations [STATES]

DOTS & BOXES : A state for each possible grid setup

Different set of lines make up different state

Our problem in 9x9 grid has 144 edges

↳ Actions can be taken in each state

Legal moves in the game.

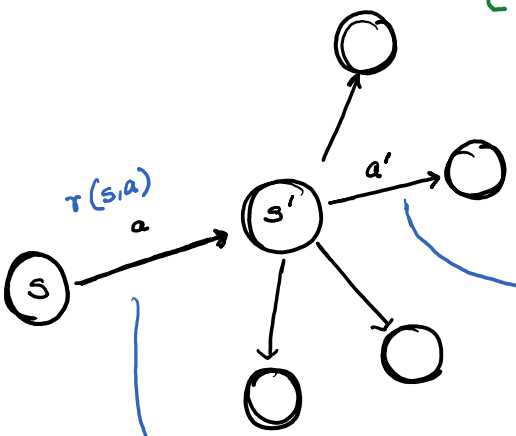
Joining adjacent unconnected dots

↳ Simplest Q-learning algorithms store a (state, action) pair

Our problem in 9x9 grid with 144 edges → Memory → $\omega(2^{144})$

Possible solution : Use a neural network to learn parameters

[DEEP Q LEARNING]



Checks for best possible state it can get to

→ For this $Q(s,a) = Q(s,a)(1-\alpha) + \alpha\left[r(s,a) + \gamma \max_{a'} Q(s',a')\right]$

Only one value is updated

↳ It takes time for the feedback to propagate backwards

So, for feedback to propagate backwards to the beginning path has to be taken same number of times the paths are long.

If there is only one, stable solution — Q-learning shall converge to that soln

But for feedbacks to propagate it requires large no. of visits to every (state, action)

PARAMETERS :

1) LEARNING RATE (α) : Impact of future actions

$\alpha = 0$ → Doesn't learn anything new

$\alpha = 1$ → Doesn't remember past information

2) DISCOUNT FACTOR (γ) : Impact of future feedback.

$\gamma = 0$ → Doesn't care about future feedback

$\gamma = 1$ → Cares about long term reward. [PRIORITIZED]

As DOTS & BOXES is a two player game. Game can be played by two agents

Possible choices for agents

1) Q-learning agent [Uses Q-learning to choose action]

2) Random agent [Used to train Q-learning agent]

MDP FORMULATION :

Markov decision process is defined by set of $\{S, A, T, R\}$

$S \rightarrow$ STATE SPACE , $A \rightarrow$ ACTION SPACE, $T \rightarrow$ TRANSITION DYNAMICS

$R \rightarrow$ REWARD FOR (STATE, ACTION)

STATE SPACE :

State for each possible grid setup [GAME CONFIGURATION]

So, for 9x9 grid it contains 8x8 boxes & 144 edges.

So, game configuration can contain upto 144 edges.

Total number of states : $\omega(2^{144})$

ACTION SPACE :

It involves connecting adjacent dots which are not connected.

So, 144 different actions

↳ As game progresses some of the lines shall be drawn.

So, on average 72 edges can be drawn in a state

$\Rightarrow 2^{144} \times 72$ (state, action) pairs

For the feedback to propagate → As discussed in background

Q-learning has to be run several times.

$\Rightarrow$ Huge amount of games to be run

REWARD POLICY ANALYSIS

Two kinds of reward structures can be performed

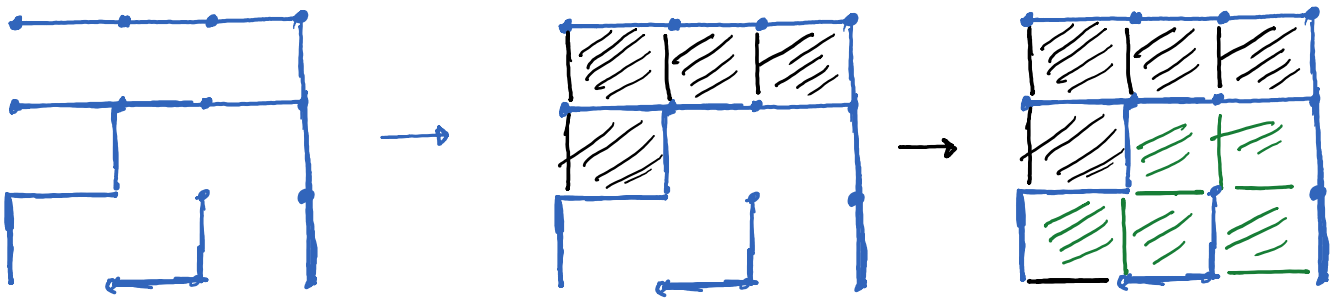
1) Only give reward when game ends

2) Give reward for each box filled and when the game ends.

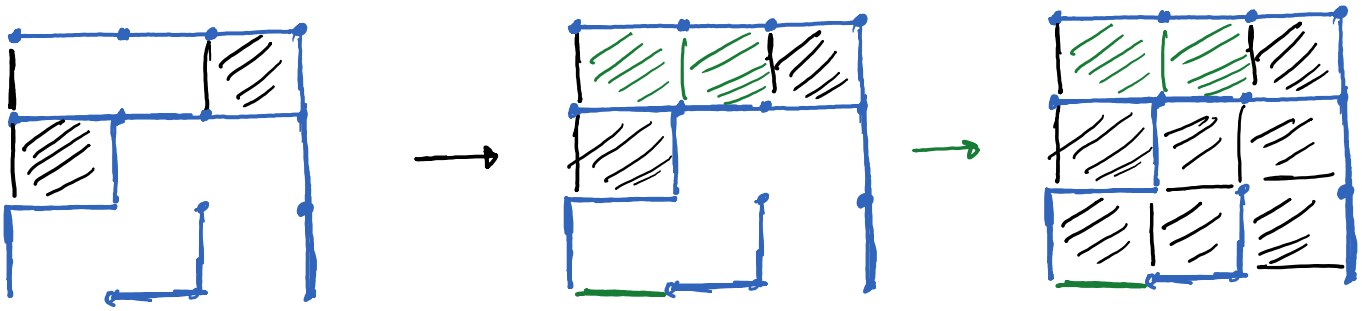
When (2) is followed reward for winning should be very high than the reward for filling up the box. Similarly reward for losing should also be very high.

Reward for each box filled up can lead to a bias of not choosing a path where boxes can be lost. But however game dots and boxes has interesting board configurations where sacrifice pays off

One such possible board configuration :



[Referred example] Thus loosing the game., instead a strategy as follows leads to win



⇒ One more possible such configuration where not maximizing immediate rewards leads to winning state [From wikipedia] (DOUBLE CROSS STRATEGY)

⇒ One more possible drawback for assigning rewards when boxes get filled is when the game ends in a draw. As per rules of games, players that did not start win. As states donot include starting order of the agents.

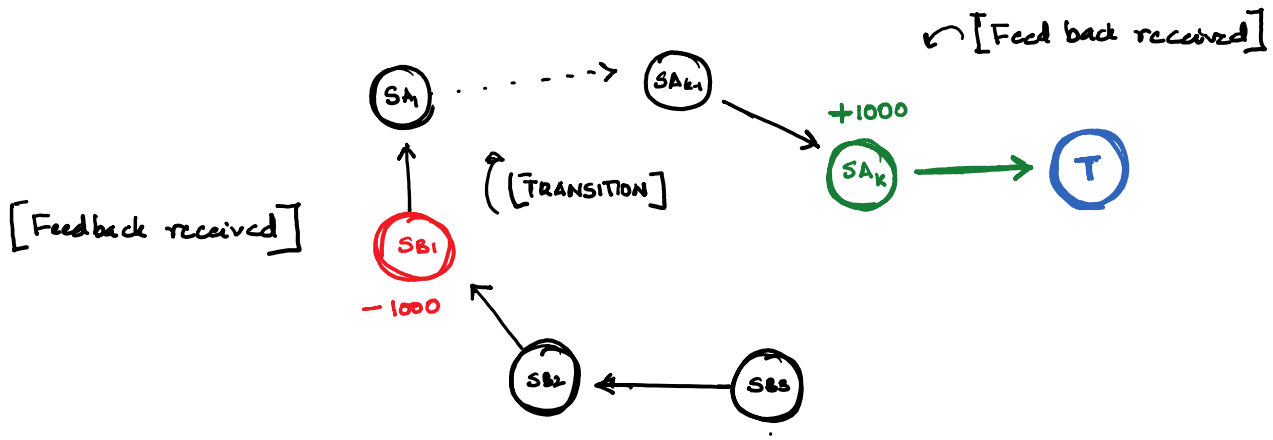
INTUITION FOR REWARD

Going by the drawbacks for giving rewards to boxes created

Lets go with simple reward policy [FEEDBACK]

of winners last state actions gets + 1000

loosers last state actions gets - 1000



(b) For the above MDP in part (a)

As there are multiple terminal states and

each of the terminal state can be reached in deterministic number of steps.

Choice of discount factor in $\gamma \in (0,1)$

DOESN'T MATTER in choosing policy that maximizes reward function

(c) DEEP Q LEARNING with neural network. [MAINTAINS EXPERIENCE REPLAY]

For every state, action pair explored, it is added to replay set

Instead updating one (s,a) we subsample (s,a,r,s') from replay set

and update parameters of neural network to better fit samples of replay

based on current neural network settings.

From notes:

Choose an initial θ for $Q(\cdot|\theta)$ [INPUT: (s,a) OUTPUT: $\mathbb{R}$]

Repeat until convergence

Choose an action a for the current state s based on $Q(s,\cdot|\theta)$

ϵ-greedy strategy

[random action] and [greedy action from s] based on NN

Take action a, observe the reward r

new state s', add (s,a,s',r)

Sample $S \subset R$

For each element in S, set $y_{(s,a,s',r)} = \left(r + \gamma \max_{a'} Q(s',a'|\theta)\right)$ [BELLMAN UPDATE]

Perform gradient descent starting @ θ.

$$\sum_{(s,a,s',r)\in S} \left(Q(s,a|\theta) - y_{(s,a,s',r)}\right)^2$$ to yield θ'

(c) QLEARNING DRAWBACKS :

1) Amount of memory required to save and update the table that increases as number of states increases

2) Amount of time required to explore each state to create the required Q-table would be unrealistic.

Neural network to approximate Q-value function

Here we try to solve a regression problem

Where our target : BELLMAN UPDATE → $[r + \gamma \max_{a'} Q(s', a' | \theta)]$

Predicted value of $Q(s,a)$ by NN → $Q(s,a|\theta)$

Minimize cost : $\sum_{(s,a,s',r)} \left(Q(s,a|\theta) - y(s,a,s',r) \right)^2$

By performing gradient descent

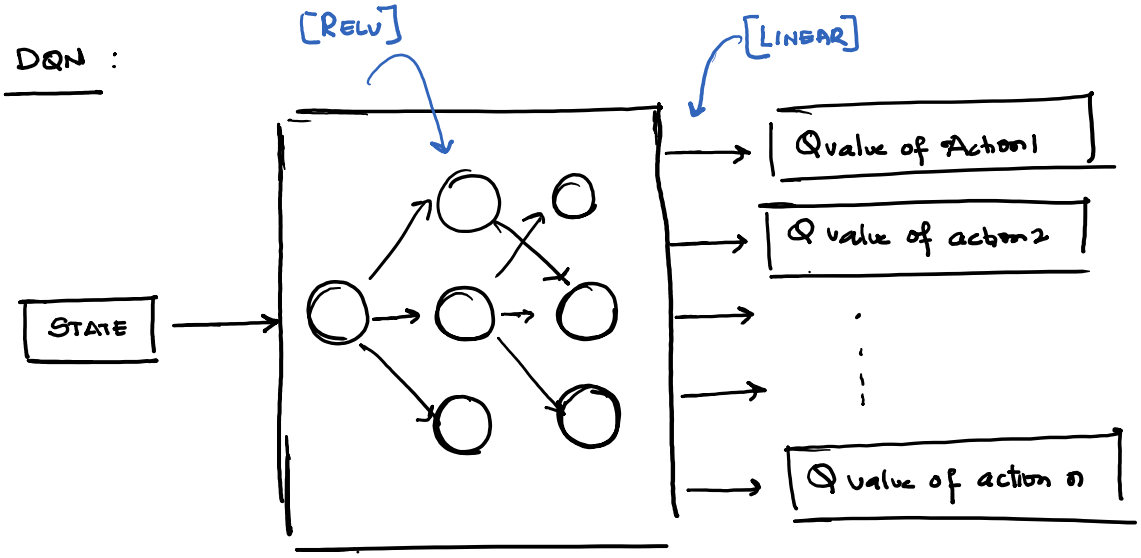
and update neural network parameters $\boxed{\theta}$

NEURAL NETWORK STRUCTURE :

Here as we are solving a regression problem

Hidden RELU units would solve our purpose

DQN :



For any state dqn outputs its predicted Q value for all the actions

As the structure remains same inorder to pick an optimal action