Roll No. ____________________ **Nov 2024**

UCS749: Conversational AI: Speech Proc. [...] Time: **2 hours** MM:–

Faculty: RGB

Thapar Institute of Engineering & Technology

Computer Science & Engineering Department

EXERCISE (TTS) (2024-25 ODD)

---

1. If P_θ models a data $\mathscr{D}$ with an intent of generating novel samples, $\mathbf{x} \sim P_\theta$. Is P_θ a posterior distribution? Comment. [2 marks]

2. If $X \equiv \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ is a temporal sequence and each time step is independently and identically distributed,
   a) Prove that $\log P(X) = \sum_{i=1}^{T} \log P(x_i)$;
   b) If X is a Bernoulli process, *i.e.* if each x_i is a coin toss, with hit rate p, the log likelihood of k hits is $k \log p + (T-k)\log(1-p)$.

2. a) $X \equiv \{x_1 \dots x_T\}$

$P(X) = P(x_1 \dots x_T)$

$= P(x_T | x_1 \dots x_{T-1}) \cdot P(x_1 \dots x_{T-1})$

... (by Bayes Rule)

$= \prod_{i=1}^{T} P(x_i | x_1 \dots x_{i-1})$ —— ①

... (expanding similarly)

But $x_i \perp x_j \; \forall i \neq j$ because iid.

hence $P(x_i | x_1 \dots x_{i-1}) = P(x_i)$ —— ②

from ① & ②,

$P(X) = \prod_{i=1}^{T} P(x_i)$

$\log P(X) = \log \prod_{i=1}^{T} P(x_i) = \sum_{i=1}^{T} \log P(x_i)$

2b) For a Bernoulli Process with hit rate 'p'. The probability of 'k' hits in 'T' trials is given as,

$$P(X) = P(k, T; p) = p^k (1-p)^{T-k}$$

or $$\log P(X) = k \log p + (T-k) \log(1-p)$$

$\blacksquare$

3. If $X \equiv \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ is the temporal sequence of a Markov process, prove that $\log P(X) = \sum_{i=1}^{T} \log P(x_i \mid x_{i-1})$.

Speech sample X is a temporal sequence of intensities,

$$X \equiv \{x_1 \ldots x_T\}$$

Given speech samples $\mathscr{D}$ as evidence, estimate $P_\theta \approx \mathscr{D}$ so that $X \sim P_\theta$ is a valid speech sample.

Let $G_{\theta,\mathcal{N}} : \mathbb{X}^K \to \mathbb{X}$ represent the model.

where,

- $\mathbb{X}$ is the field of inputs.
  - For a continuous model, $\mathbb{X} \equiv \mathbb{R}$; whereas,
  - For categorical model, $\mathbb{X} \equiv \mathbb{R}^{256}_{[0,1]}$;
  - It may also be a hybrid model,
    e.g. $G_{\theta,\mathcal{N}} : \mathbb{R}^K \to \mathbb{R}^{256}_{[0,1]}$.
    Can you think how?

Let $G_{\theta,\mathcal{N}} : \mathbb{X}^K \to \mathbb{X}$ represent the model.

where,

- $\mathcal{N}$ is a noise sampler,
  - Generally, implemented as a normal distribution; or
  - Implemented implicitly as dropouts.

Let $G_{\theta,\mathcal{N}} : \mathbb{X}^K \to \mathbb{X}$ represent the model.

where,

- θ are parameters of the model; and
- $\mathbf{x}_t = G_{\theta,\mathcal{N}}\left(\left[\mathbf{x}_{t-K} \ \dots \ \mathbf{x}_{t-1}\right]\right)$ models the conditional distribution $P(x_t \mid x_{t-K} \dots x_{t-1})$

In case of Conditional Generation,

$\mathbf{x}_t = G_{\theta,\mathcal{N}}\left(\left[\mathbf{x}_{t-K} \ \dots \ \mathbf{x}_{t-1}\right], \mathbf{h}\right)$
models the conditional distribution
$P(x_t \mid x_{t-K} \dots x_{t-1}, \mathbf{h})$

$\mathbf{h}$ represents the global conditions, *e.g.*

- Text input;
- Speaker;
- Accent;
- and so forth.

Let
$X \equiv \begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_T\end{bmatrix} \sim G_{\theta,\mathcal{N}}$ be the output of the auto-regressive model, so that
$\mathbf{x}_t = G_{\theta,\mathcal{N}}\left(\begin{bmatrix}\mathbf{x}_{t-K} & \dots & \mathbf{x}_{t-1}\end{bmatrix}\right)$

And,
$Y \equiv \begin{bmatrix}\mathbf{y}_1 & \dots & \mathbf{y}_T\end{bmatrix} \sim \mathscr{D}$ be a speech sample from dataset. Recall that for each time step, the sound intensity is pre-processed so that the values are remapped, quantised, and converted to one-hot vectors, so that $\mathbf{y}_t \in \{0, 1\}^{256}; \|\mathbf{y}_t\|_1 = 1$.

The training objective is the cross entropy function, given as,

$$\underset{G}{\text{minimise}} \quad \underset{X\sim G, Y\sim \mathscr{D}}{\mathbb{E}} \left[\sum_{i,t} y_{i,t} \log x_{i,t} \right]$$

Tacotron

Artificial Neuron

$$\mathbf{y} = \mathcal{N}(W\mathbf{x} + \mathbf{b})$$

where,

- $\mathbf{x}, \mathbf{y} \in V$, for some vector space V;
- $W, \mathbf{b}$ are learnable weights; and
- $\mathcal{N}$ is a non-linearity applied element-wise.

without loss of generality

ANN Layer

$$\mathbf{x}_{l+1} = \mathcal{N}(W_l\mathbf{x}_l + \mathbf{b}_l)$$

where,

- $\mathbf{x}_l \in V \;\forall l$, for some vector space V;
- $W, \mathbf{b}$ are learnable weights; and
- $\mathcal{N}$ is a non-linearity applied element-wise.

Sequential Network

e.g. AlexNet, VGG Net etc.

$$
\begin{aligned}
\mathbf{y} &= \mathbf{g} \otimes \mathbf{x} \\
\mathbf{g} &= \sigma(W\mathbf{x} + \mathbf{b})
\end{aligned}
$$

where,

- $\otimes$ represents Hadamard product;
- σ represents the logistic sigmoid function that is applied element-wise; and
- $\mathbf{g} \in \mathbb{R}^n_{[0,1]}$ represents the gate.

Highway Gate

$$
\begin{aligned}
\mathbf{y} &= (\mathbf{1} - \mathbf{g}) \otimes \mathbf{x} + \mathbf{g} \otimes \mathbf{h} \\
\mathbf{g} &= W_g\mathbf{x} + \mathbf{b}_g \\
\mathbf{h} &= W_h\mathbf{x} + \mathbf{b}_h
\end{aligned}
$$

Highway Gate

In practice,

$$
\begin{aligned}
\mathbf{y} &= (\mathbf{1} - \mathbf{g}) \otimes \mathbf{x} + \mathbf{g} \otimes \mathbf{h} \\
(\mathbf{g}, \mathbf{h}) &= (\sigma(\tilde{\mathbf{y}}_{:n}), \mathcal{N}(\tilde{\mathbf{y}}_{n:})) \\
\tilde{\mathbf{y}} &= W\mathbf{x} + \mathbf{b}
\end{aligned}
$$

Constructing a highway network

$\forall\, l \in \{1, \ldots, L\}$

$$
\begin{aligned}
\mathbf{x}_l &= (\mathbf{1} - \mathbf{g}_l) \otimes \mathbf{x}_l + \mathbf{g}_l \otimes \mathbf{h}_l \\
(\mathbf{g}_l, \mathbf{h}_l) &= \big(\sigma(\tilde{\mathbf{x}}_{l,\,:\,n}), \mathcal{N}(\tilde{\mathbf{x}}_{l,n\,:\,})\big) \\
\tilde{\mathbf{x}}_l &= W_l \mathbf{x}_{l-1} + \mathbf{b}_l
\end{aligned}
$$

With $L = 50$ layer deep models, the very-deep network building strategy shown here predates googlenet and resnet.

$$P_\theta\,(Y|X) \quad \text{modelled as} \quad \mathbf{y} = G(\mathbf{x}; \theta)$$

as simple
as possible

$$P(\text{speech} \mid \text{text})$$

$$
\begin{aligned}
X &\equiv \{\mathbf{x}_1 \ldots \mathbf{x}_N\} \\
Y &\equiv \{\mathbf{y}_1 \ldots \mathbf{y}_N\}
\end{aligned}
$$

$$P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T)$$

0.53	0.24	0.01	0.45	0.14	0.42	0.59	0.37	0.32	0.18	0.29
0.56	0.16	0.32	0.44	0.32	0.63	0.52	0.85	0.75	0.75	0.33
0.84	0.93	0.13	0.63	0.06	0.83	0.13	0.65	0.11	0.65	0.17
0.92	0.62	0.08	0.13	0.18	0.72	0.83	0.54	0.83	0.29	0.45
0.41	0.33	0.89	0.71	0.33	0.86	0.15	0.13	0.68	0.99	0.14
0.26	0.17	0.03	0.48	0.29	0.40	0.60	0.27	0.85	0.66	0.11
0.95	0.07	0.17	0.79	0.59	0.41	0.07	0.22	0.60	0.11	0.11
0.27	0.72	0.02	0.97	0.62	0.53	0.49	0.81	0.00	0.29	0.52
0.88	0.17	0.24	0.77	0.59	0.40	0.71	0.88	0.07	0.21	0.17
0.23	0.06	0.48	0.97	0.18	0.85	0.62	0.82	0.73	0.66	0.70
0.67	0.66	0.24	0.46	0.19	0.87	0.83	0.19	0.15	0.21	0.70
0.28	0.72	0.32	0.04	0.48	0.16	0.47	0.34	0.52	0.14	0.75
0.94	0.73	0.20	0.57	0.68	0.47	0.90	0.54	0.67	0.18	0.84