CID : 01752815

Q1.

ⓐ x_t : average atmospheric pressure and temperature of today

y_t : average atmospheric pressure and temperature of the next day.

$x_t \Rightarrow 2\times1$ vector $\quad y_t \Rightarrow 2\times1$ vector.

$h_t \Rightarrow d\times1$ vector.

~~So~~

So. $W_{hh} \Rightarrow d\times d$ matrix , $W_{xh} \Rightarrow d\times2$ matrix

$W_{hy} \Rightarrow d\times2$ matrix.

without bias term : $\underbrace{\text{number of parameters}} = d\times d + d\times2 + d\times2 = d^2 + 4d$

with bias term:

bias for h_t : $d\times1$

bias for y_t : 2×1

So. $\underbrace{\text{number of parameters}} = d^2 + 4d + d + 2 = d^2 + 5d + 2$

ⓑ None

ⓒ We will input the first data x_t of the batch into RNN. If this is the first batch, we will set h_{t-1} to be zero and compute the h_t. If this is not the first batch, we will use h_{t-1} and x_t to compute h_t. Then we wil save the value h_t for next data and compute y_t for this data. After all the computation, we finish the one data of the batch. Then, we need to loop all data of this batch by above process. Finally, we need to save the last h_t value from the last data of the batch for next batch.

Because the target is continuous values, I ~~think~~ think the loss function shoul be Mean Square Error.

Q2

① $x_1 = 1$, $x_2 = 1$ $\Rightarrow \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 0.7214 \\ 0.2654 \\ 0.0132 \end{pmatrix} \Rightarrow$ Red

② for $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0.2595 \\ 0.7054 \\ 0.0351 \end{pmatrix}$

for $\begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} \Rightarrow \begin{pmatrix} 0.9362 \\ 0.0466 \\ 0.0171 \end{pmatrix}$

cross-entropy : the base is e

for $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $S_1 = -\log(0.2595) \approx 1.3490$

for $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $S_2 = -\log(0.0466) \approx 3.0659$

for $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ $S_3 = -\log(0.0132) \approx 4.3266$

So. $S = S_1 + S_2 + S_3 = 8.7415$

Q3

for uniform distribution. $U[a,b]$

PDF $f(x) = \begin{cases} \frac{1}{b-a}, & a<x<b \\ 0, & \text{others} \end{cases}$

Mean: $E(x) = \int_{-\infty}^{+\infty} x f(x)\,dx$

$= \frac{1}{b-a}\int_{-\infty}^{+\infty} x\,dx$

$= \frac{a+b}{2}$

Variance: $V(x) = \int_{-\infty}^{+\infty} x^2 f(x)\,dx$

$= \frac{1}{b-a}\int_{-\infty}^{+\infty} x^2\,dx$

$= \frac{(a-b)^2}{12}$

So. for $U[-a,a]$.

Mean $= \frac{(-a)+a}{2} = 0$

Variance $= \frac{(-a-a)^2}{12} = \frac{4a^2}{12} = \frac{1}{n}$

$\therefore \quad n = \frac{3}{a^2}$

Q4

a. and b.

	Number of output neurons	Number of parameters
Conv1	2560	760
MaxPool	640	\
Reshape and FC	200	128200
Softmax	10	2010
~~Total~~		
Total	3410	130970

Q5

(a) $N(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$

$$\text{Log-Likelihood} = -\frac{1}{2}\sum_{i=1}^{m}\left(\frac{x^{(i)}-\mu}{\sigma}\right)^2 - m\log\sigma - \frac{m}{2}\log 2\pi$$

$$= -\frac{1}{2}\left[\left(\frac{2-\mu}{\sigma}\right)^2 + \left(\frac{5-\mu}{\sigma}\right)^2 + \left(\frac{6-\mu}{\sigma}\right)^2\right] - 3\log\sigma - \frac{3}{2}\log 2\pi$$

(b) None.

Q6

ⓐ We can apply Autoencoder method to achieve the goal. We can input the high-dimension input data to encode to be low-dimension latent vectors. Then, we decode the low-dimension latent vectors to be reconstructed high-dimension output data. After training process, we discard the decoder part and only apply encoder to reduce dimensionality of the data.

ⓑ we can apply variational autoencoders (VAEs) method. VAEs associate to each latent vector z the mean and covariance of a multiGaussian in the model space, and then sample x in the model space from this multi-Gaussian pdf. So it generates latent vectors and images that statistically look like but are not identical.

ⓒ We can apply Generative Adversarial Networks (GANs) method. GANs apply a generator neural network G to a latent vector z, which is usually Gaussian. Once $G(z)$ has been trained jointly with discriminator $D(G(z))$, each $x = G(z)$ in the model space, for every random value of z, can be considered a sample of original image.