Problem 1 → RNN

(a)

$$\phi(z) = \sigma(z) - 0.5$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\boxed{\nabla h_t = \nabla h_{t+1} \, \sigma'(z_t) \, w}$$

(b)

If $x = 0$, $h_t = 0$ for all t

$\sigma'(0) = 1/4$

$$\nabla h_t = \nabla h_{t+1} \, \frac{1}{4} \, w$$

Only when $\alpha = 4$, $\nabla h_t = \nabla h_{t+1}$

Thus $\alpha = 4$

Problem 2 (LSTM)

(a) False. Because, if x_t = 0 vector we have,

$f_t = \sigma(U_f h_{t-1} + b_f)$

$i_t = \sigma(U_i h_{t-1} + b_i)$

$\tilde{c}_t = \tanh(U_c h_{t-1} + b_c)$

$c_t = \sigma(U_f h_{t-1} + b_f) \odot c_{t-1} + \sigma(U_i h_{t-1} + b_i) \odot \tilde{c}_t$

$o_t = \sigma(U_o h_{t-1} + b_o)$

$\therefore\ h_t = \sigma(U_o h_{t-1} + b_o) \odot \tanh((\sigma(U_f h_{t-1} + b_f) \odot c_{t-1}) + (\sigma(U_i h_{t-1} + b_i) \odot \tilde{c}_t))$

$h_t \neq h_{t-1}$

(b) False. Because, from LSTM equations we can see that i_t & $\tilde{c}_t$ depends on h_{t-1} not on f_t value.

(c) True. Because, the equation of f_t, i_t, o_t is a sigmoid with range (0,1).

(d) False. Because, sigmoids are applied element-wise. Each sigmoid gives values in the range (0,1) which need not sum to 1.

Problem 3

(a) It is given that the dimension of $x_t = [2 \times 1]$

Dimension of $W_f = [1 \times 2]$

The dimension of $W_f x_t$ will therefore be $[1 \times 1]$

This makes dimension of f_t to be $[1 \times 1] = [1]$ due to the property of addition operator to have similar dimension among its elements.

Similarly we can show that the dimension of i_t & o_t to be $1 \times 1 = 1$ as W_i & W_o dimensions are $[1 \times 2]$

$\tilde{c}_t$ dimension is 1

c_t is the output of element-wise multiplication of f_t & c_{t-1}, i_t & $\tilde{c}_t$

The dimension of c_t is 1

The dimension of h_t also is 1 due to c_t & o_t being 1

(b) $c_0 = 0$ vector

$h_0 = 0$ vector

$$f_1 = \sigma\left([1\ 2]\begin{bmatrix}1\\0\end{bmatrix} + [0.5] \times 0 + [0.2]\right) = \sigma(1.2)$$

$$= 0.7685$$

$$i_1 = \sigma\left([-1\ 0]\begin{bmatrix}1\\0\end{bmatrix} + [2] \times 0 + [-0.1]\right) = \sigma(-1.1)$$

$$= 0.24974$$

$$\tilde{c}_1 = \tanh\left([1\ 2]\begin{bmatrix}1\\0\end{bmatrix} + [1.5]\times 0 + [0.5]\right)$$

$$= \tanh(1+0.5) = 0.905$$

$$c_1 = f_1 \times c_0 + i_1 \times \tilde{c}_1 = 0 + 0.226052 = 0.226052$$

$$o_1 = \sigma\left([3\ 0]\begin{bmatrix}1\\0\end{bmatrix} + [-1]\times 0 + [0.8]\right) = \sigma(3.8)$$

$$o_1 = 0.978119$$

$$\boxed{h_1 = 0.217415}$$

$$f_2 = \sigma\left([1\ 2]\begin{bmatrix}0.5\\-1\end{bmatrix} + [0.5]\times 0.217415 + [0.2]\right)$$

$$= \sigma(-1.19129) = 0.233028$$

$$i_2 = \sigma\left([-1\ 0]\begin{bmatrix}0.5\\-1\end{bmatrix} + [2]\times 0.217415 + [-0.1]\right)$$

$$= \sigma(-0.16517) = 0.458801$$

$$\tilde{c}_2 = \tanh\left([1\ 2]\begin{bmatrix}0.5\\-1\end{bmatrix} + 1.5 \times 0.217415 + [0.5]\right)$$

$$= \tanh(-0.67388) = -0.58753$$

$$c_2 = f_2 \times c_1 + i_2 \times \tilde{c}_2 = -0.21688$$

$$o_2 = \sigma\left([3\ 0]\begin{bmatrix}0.5\\-1\end{bmatrix} + [-1]\times 0.217415 + [0.8]\right)$$

$$= \sigma(2.082585) = 0.889199$$

$$h_2 = 0.889199 \times \tan(-0.21688) = -0.18988$$

$$\boxed{h_2 = -0.18988}$$

(c) $MSE = \frac{1}{2}\left[(0.5-h_1)^2 + (0.8-h_2)^2\right]$

$= 0.52986$

$= 52.986\,\%$

Problem 4

(a) To find out KL divergence between $(q(x) \| p(x))$

Let $q(x) = N(\mu_1, \sigma_1)$, $p(x) = N(\mu_2, \sigma_2)$

$$KL(q,p) = -\int q(x) \log p(x)\,dx + \int q(x) \log q(x)\,dx$$

$$\int q(x) \log q(x)\,dx = \frac{-1}{2}(1 + \log 2\pi\sigma_1^2)$$

$$\therefore -\int q(x) \log \frac{1}{(2\pi\sigma_2^2)^{1/2}} e^{-(x-\mu_2)^2/2\sigma_2^2}\,dx = \int -q(x) \log p(x)\,dx$$

$$-\int q(x) \log p(x)\,dx = \frac{1}{2}\log(2\pi\sigma_2^2) - \int q(x) \log e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}\,dx$$

$$= \frac{1}{2}\log(2\pi\sigma_2^2) - \int q(x)\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)dx$$

Separating the sums and taking out σ_2^2 of integral we get.

$$-\int q(x)\log p(x)dx = \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\int q(x)x^2dx - \int q(x)2x\mu_2 dx + \int q(x)\mu_2^2 dx}{2\sigma_2^2}$$

Expectations

$$\int q(x)x^2dx = \sigma_1^2 + \mu_1^2$$

$$\int q(x)\mu_2^2 dx = \mu_2^2$$

$$\int q(x)2x\mu_2 dx = \mu_1\mu_2 \times 2$$

$$= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{(\sigma_1^2 + \mu_1^2 - 2\mu_1\mu_2 + \mu_2^2)}{2\sigma_2^2}$$

$$= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2}$$

$$\therefore KL(q,p) = \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + \log 2\pi\sigma_1^2)$$

$$= \frac{2}{2}\log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

We have $\sigma_2 = 1$, $\mu_2 = 0$, $\sigma_1 = \sigma$, $\mu_1 = \mu$

$$\therefore KL(q,p) = \log\left(\frac{1}{\sigma}\right) + \frac{\sigma^2 + \mu^2}{2} - \frac{1}{2}$$

(b) When α is too large the input will lead to similar encoding. As a result whenever the decoder works it will generate a single kind of output which will not vary with the input.

(c) The aspects where VAE & PCA are different are

(i) PCA is essentially a linear transformation but Auto-encoders are capable of modelling complex non-linear functions

(ii) PCA features are totally linearly uncorrelated with each other since features are projection onto the orthogonal basis. But autoencoded features might have correlations since they are just trained for accurate reconstruction

(iii) PCA is faster and computationally cheaper than autoencoders

Problem 5

(a) False

In GAN both the generator & discriminator are trained simultaneously. With update of one parameter the nature of optimization problem that is being solved changes as it is a dynamic system. Therefore the update of generator k (>1) times for every one time update of discriminator cannot guarantee acceleration of training of GAN

(b) The value of D(G(z)) is closer to 0 because early in the training D is much better than G. One reason is that G's task of generating images that look like real data is a more difficult that D's task of distinguishing fake images from real images.

(c) I would choose non-saturating cost as it leads to much higher gradient early in the training and thus help the generator to learn quicker.

(d) False. At the end of training of G it is able to fool D. So $D(G(z))$ is close to 0.5 which means that D is randomly guessing