Question1

$W_1 = 27$ $H_1 = 27$ $D_1 = 256$

Number of filters K = 384 Spatial Extent F = 3 Stride S = 1 amount of zero padding P = 0

$W_2 = (27 – 3 + 0)/1 + 1 = 24$

$H_2 = (27 – 3 + 0)/1 + 1 = 24$

$D_2 = K = 384$

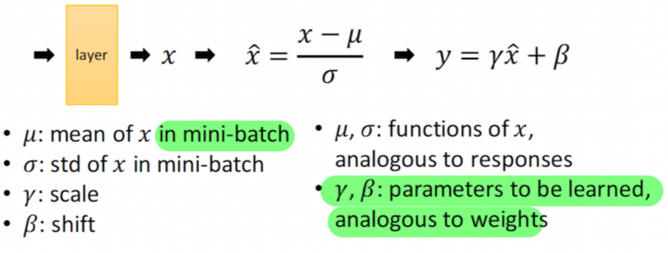
Therefore, the output shape is 24 * 24 * 384

Number of parameters = (3 * 3 * 256 + 1)*384 = 885120

Number of FLOPs = 2 * 3 * 3 * 256 * 24 * 24 * 384 = 1019215872

Question 2

Batch normalization means normalizing each layer for each mini batch. The batch input will subtract the mean and divided by the standard deviation. And then multiply the scale and add the bias, which are to be learned. The advantage is it can reduce the vanishing gradients problem, accelerates training, decreases sensitivity to initialization, and improve regularization. The disadvantage is it is not suitable for Recurrent network (RNN, LSTM) and online learning, and it also slow down the speed of prediction.



$$\rightarrow \text{layer} \rightarrow x \rightarrow \hat{x} = \frac{x-\mu}{\sigma} \rightarrow y = \gamma\hat{x} + \beta$$

- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

Layer normalization is similar to batch normalization, but it is done vertically. It computes the mean and standard deviation for all feature dimensions, respectively. And perform the normalization by subtract the mean and divided by standard deviation. The advantage is it is more suitable in Recurrent Neural Network since it keeps sequential dependency of the data. The disadvantage is it may not as good as batch normalization in CNN.

Question3

(a) The forget gate determines which components of the previous state $\mathbf{c}_{t-1}$ and how much of them to remember/forget. The input gate determines which components of the input from current input $\mathbf{x}^{(t)}$and $\mathbf{h}^{(t-1)}$ and how much of them should go into the current state.

(b) Sigmoid function used for forget gate and input gate is to guarantee the value of gate vector is often close to 0 or 1. While the tanh used for the memory cell and output guarantees they have strong gradient signal during back propagation.

Question4

First, similar to Transformer, it contains no recurrence in BERT, which means information about token positions need to be inject in order to make use of order of sequence. Second, compare to the positional encoding used in Transformer, the positional embedding introduces learnable parameters. It can improve the ability of representing the sequence information better since there are more training examples in BERT.

Question6

z should follow the Gaussian distribution with mean equal to $x^2 + 2x$, and variance equal to $x^4/4$ since ε is a random variable that follows the normal distribution N (0, 1). Therefore, the density function of $p(z|x)$ is:

$$p(z \mid x) = \frac{1}{\sqrt{2\pi} \cdot x^2/2} e^{\frac{-\left(z-(x^2+2x)\right)^2}{\frac{x^4}{2}}}$$

Question7

VAE and GAN both take use of unlabeled data and assume that they are generated from latent space. However, VAE chooses to learn a conditional distribution of x regards to z while GAN choose to learn function x = g(z). In the other hand, VAE tries to minimize the KL divergence between the real data distribution and the generated data distribution, while GAN tries to minimize the JS divergence between the real data distribution and model distribution. Since the objective is intractable, VAE introduces an encoder-decoder pair while GAN uses generator-discriminator pair.

Question8

The objective function of the discriminator in GAN is :

$$V(G, D) = E_{\mathbf{x} \sim p_r(\mathbf{x})}[\log\ D(\mathbf{x})] + E_{\mathbf{x} \sim p_g(\mathbf{x})}[\log\ (1 - D(\mathbf{x}))]$$

$p_r(\mathbf{x})$ is the distribution of real data, $p_g(\mathbf{x})$ is the distribution of generated data, $D(\mathbf{x})$ is the probability that x is a real data.

The generator of GAN hopes to minimize:

$$-E_{\mathbf{x} \sim p_g(\mathbf{x})}[\log\ D(\mathbf{x})]$$

The objective function of critic in WGAN is:

$$L_c = E_{\mathbf{x} \sim p_r}[f(\mathbf{x})] - E_{\mathbf{x} \sim p_g}[f(\mathbf{x})]$$

f(x) is the critic function.

The objective function of WGAN is:

$$-E_{\mathbf{x} \sim p_g(\mathbf{x})}[\log\ f^*(\mathbf{x})]$$