Name: QIU Yaowen
ID : 20784389

Question1:

(a)
For two distribution P and Q, the cross-entropy is defined as:

$$H(P,Q) = \sum_X P(X)\log\frac{1}{Q(X)} = -E_P[\log\ Q(X)]$$

Because in machine learning, we always want to minimize the difference between two distribution, measure by KL-divergence, defined as :

$$KL(P \parallel Q) = \sum_X P(X)\log\frac{P(X)}{Q(X)} = E_P[\log\ P(X)] - E_P[\log\ Q(X)]$$
$$= H(P,Q) - H(P)$$

In this formula, KL-divergence equals to cross entropy minus entropy of P (which is fixed). Therefore, minimize cross-entropy is minimize KL-divergence.

(b)
Take supervised learning in machine learning as example:

In supervised learning, we wish to minimize cross-entropy:

$$H(P,Q) = -\int P(\mathbf{x},y)\log\ Q(y \mid \mathbf{x})d\mathbf{x}dy$$
$$\approx -\frac{1}{N}\sum_{i=1}^{N}\log\ Q(y_i \mid \mathbf{x}_i)$$

That is same as maximize the log likelihood:

$$\sum_{i=1}^{N}\log\ Q(y_i \mid \mathbf{x}_i)$$

It is similar in unsupervised learning problem.

Question2

(a)
The training error will decrease with the increase of model capacity. Because a higher model capacity means a more complex model, which can fit the data better in large hypothesis space. This led to less training error

(b)
When model capacity is low, the generalization error will decrease with the increase of model capacity. Since model at this stage cannot learn the distribution of data so well, lead to worse performance in both training data & testing data.

When model capacity is high, the generalization error will increase with the increase of model capacity. Since higher capacity will lead to overfitting problem. The model learn the training data distribution so well and loss generalization ability.

(c)
No, directly minimizing the training error may reduce the generalization error when model is simple but definitely will lead to overfitting problem.

(d)
The surrogate losses are convex functions that are easy to minimize. And the property that " the surrogate losses need to be upper bounds on the true loss function" guarantees the process of minimizing the surrogate loss can push down the real loss.

Logistic loss is one surrogate loss function:

$$L_{\log}(y, z) = \frac{1}{\log 2} \log\left(1 + \exp\left(-yz\right)\right)$$

Question3

(a)
The expected generalization error is the overall performance of the algorithm over all training set, defined as:

$$\epsilon = E_S[\epsilon(h_S)] = E_S\left[E_{(x,y)\sim\mathcal{D}}[(y - h_S(x))^2]\right]$$

Because the less the expected generalization error means the algorithm will work well for every possible dataset.

(b)
The first term is called bias[2], because the choice of hypothesis class affects this term. That is not dependent on training data

The second term is call variance. The randomness of the choice the training data affects this term.

(c)
The variance is an error due to the sensitivity of the fluctuations in the training dataset. A large variance can make a model fit the random noise in the training data, thus lead to overfitting problem.

Question4
(a)
The regularization increase bias and decrease variance. Because it put penalties on weights to prevent model to fit the random noise in training data so sell.

(b)
The L1 regularization makes the total loss achieve minimum at some corners, which lie on the axis. It means that some weights are set to be zero. Therefore, a sparse model is made.
And L2 regularization does not make the total loss achieve its minimum value at the corners (usually), it only shrinks the size of coefficients.

Question5

(a)
Reduce variance.
Because it associates a binary mask on each variable in the network and prevents the co-adaptions of training parameters on the data, similar to the idea to bagging.

(b)
It has three key ideas:

Use momentum to accelerate learning.
The adaption of learning rate to slow down changes on parameters.
The correction of bias in moment estimates.

(c)
Because the deep neural network can be considered that there are lots of sequential models. We want to normalize at each layer to avoid the covariate shift and make different layers more independent. In addition, the batch normalization can decrease sensitivity to initialization and improve regularization.

(d)
In plain net, larger training & testing errors with more layers in deep learning, because the gradients are hard to be propagated to lower layers. The residual connection can back propagate the gradient to lower layers due to the identity connections. This results smaller training & testing errors with more layers.

Question6

(a)
$\mathbf{c}_{t-1}$ is query, because it holds the context of step t-1
$\mathbf{h}_j$ is key, because it is the latent state at step j for encoding
$\mathbf{h}_j$ is value, reason is same as above.

(b)
The input is the sentence (or any sequence).
The output is calculated as following: First, each word in the input is embedded. We have queries Q, Keys K, and Values V. Then, computing the Score by $Q*K^T$. The score is divided by $\sqrt{d_k}$ and take the SoftMax. The output is the dot product of score and V, which is the

representation of input.

Question7

It is a lower bound of loglikelihood:

$$\log p_\theta\left(\mathbf{x}^{(i)}\right) \geq \mathcal{L}\left(\mathbf{x}^{(i)}, \boldsymbol{\theta}, \boldsymbol{\phi}\right)$$

It is called the reconstruction error because it measures how well the distribution of the model decoded output consistent with the input $\mathbf{x}^{(i)}$.

The original form makes the backpropagation impossible, but the reparameterization trick can change the first term to make it possible to perform backpropagation:

$$\mathbf{z} = \mu_z(\mathbf{x}, \phi) + \sigma_z(\mathbf{x}, \phi) \odot \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Now $\mathbf{z}$ depends on μ_z, σ_z, and ϵ.

Question8

(a)

GAN minimize the JSD:

$$JS\left(p_r \parallel p_g\right) = \frac{1}{2} KL(p_r \parallel p_a) + \frac{1}{2} KL\left(p_g \parallel p_a\right)$$

Where p_r is the real distribution, p_g is the fake distribution, $p_a = \left(p_r + p_g\right)/2$.

A high cost for generating fake images in JSD.

(b)

EMD is the measure of distance between two distributions. Suppose there are two piles of dirt x and y, which are different shape. The minimum cost of the plan that can let x become y by moving dirt from x is the EMD.

(c)

Compared to JS, the EMD provides a smooth measurement between two probabilistic distribution and reliable measurement of two distribution. It is useful for stable learning using gradient descents in training GAN.

Question9

In GAN, the process of generating fake images is essential and the training gradient is not stable. The EMD can solve this problem better than JS.

But in traditional machine learning classification problem, the KL-divergence is strong enough to handle traditional machine learning classification problem, the computation of KL-divergence is easy, and the time complexity is lower than EMD, which makes the KL-divergence the main-stream of distance metrics.

Question10

(a)

Because the temporal difference target, which is an unbiased estimation of the $Q_{k+1+}(s,a)$, can improve the current estimation of $Q_k(s,a)$.

The condition of convergence is each (s,a) pair updated infinitely often.
It converges by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[\left(r(s,a) + \gamma \max_{a'} Q(s',a')\right) - Q(s,a)\right]$$

Where Q(s,a) is the Q-value, s is the state, a is the action, γ is the discount factor.

The epsilon-policy help convergence because it helps the agent collect more data for learning by exploring new parts of state space and making use of experience.

(b)

The θ is updated using the past experience and their Q-values, each update of θ will make the value network closer to the real Q-function.

The target network that has the same structure as DQN can solve the problem of unstable learning, which means the update of θ is like chasing a moving target.

Because the Q-values used are not accurate, and they are influenced by some random factors.

We can get a robust estimate of the value by taking average over many runs.

Question 11

(a)

First, in the update rule, if $r(\tau^i) < 0$, the θ will be changed in the opposite direction and the update rule makes bad experiences less likely; if $r(\tau^i) > 0$, the update rule makes good experience more likely.

Second, the REINFORCE is an on-policy algorithm which take all of data which are generated by the policy itself to update its parameter, compared to DQN which is an off-policy algorithm using partial data which are not generated by the algorithm itself.

(b)

$V^{\pi}(s)$ is the value functions of a policy π under state s; $Q^{\pi}(s,a)$ is the Q-value under policy π when state is s and action is a.

It is called the advantage function because it tells us the performance of action a is relative to the average.