Problem 1

1. True, this is the basic idea of Reconstruction-Base Approach.
2. True, the definition of the diameter is the length of the shortest path between the most distanced node. And in fully connect graph, any pairs of nodes connect to each other, which means the shortest path between any pairs of nodes must be 1.
3. The proximity of an outlier object to its nearest neighbors significantly deviates from the proximity of most other objects to their neighbors.
   In the other word, the outliers are usually far away from other points.
4. Sparse interactions, parameter sharing, translation equivalence.
5. The translational equivalence means if we adopt a translation to the input data, the new output should be equal to the old output adopted with the equal translation.
   The translation invariance means the translation to the input data should not affect the output.
6. In dropout, we randomly drop some non-output unit. Such operation force model to make predictions based on unmasked units/features and prevent the model rely on certain units/features too much when it makes prediction. Thus, dropout can prevent overfitting. On the other hand, dropout is like a regulation process. It prevents certain parameters/weights are dominant in network by randomly choose some units each iteration and not update their parameters.
7. CC(A) = 0
   CC(B) = (2* 1) / (3 * 2) = 1 / 3
   CC(C) = 1 / 3
   CC(D) = (2 * 2) / (3 * 2) = 2 / 3
   CC(E) = (2 * 4) / (6 * 5) = 8 / 30 = 4 / 15
   CC(G) = (2 * 1) / (2 * 1) = 1
   CC(H) = (2 * 2) / (3 * 2) = 2 / 3
   CC(F) = 1
   CC = (0 + 1 / 3 + 1 / 3 + 2 / 3 + 4 / 15 + 1 + 2 / 3 + 1) / 8 = 8 / 15

Problem 2

a) For the numerator, we can explain it as:
   Number of points that have less r distance with target point O. In other words, we are counting the number of O's neighbors.
   And we normalize it with the size of datasets for comparability between different datasets.
   Obviously, using this formula, we are measuring the proximity between target point with its neighbors.
   If a point has a small value on this equation, the proximity of this point to its nearest neighbors is small, which according to the core idea of the proximity-based approach, is less likely to be in a cluster and more likely to be an outlier.

b) Consider the transformation on given formula:

$$|| o' \mid dist(o, o') \leq r || \leq ceil(pi * ||D||)$$

$$|| o' \mid dist(o, o') \leq r || \leq k$$

The above equation can be read as:
A non-outlier point o should have at least k neighbors that have less r distance with o.
Obviously, let's set o_k is k^{th} nearest neighbor of O.
Obviously, if

$$dist(o, o_k) > r$$

Then

$$dist(o, o_{k+1}) > r$$
$$dist(o, o_{k+2}) > r$$
$$dist(o, o_{k+3}) > r$$

…

$$for\ k \leq x \leq ||D||, dist(o, o_x) > r$$

Then,
$|| o' \mid dist(o, o') \leq r ||$ must be $\leq$ k.
Since $|| o' \mid dist(o, o') > r ||$ must be $>$ ||D|| - K, and
$|| o' \mid dist(o, o') \leq r || + || o' \mid dist(o, o') > r|| = ||D||$

Thus, for any point o with

$$dist(o, o_k) > r$$

$|| o' \mid dist(o, o') \leq r || \leq k$ must be true, and it will be counted as an outlier.

c) Point -4.5 has 3 neighbors
Point -4 has 3 neighbors
Point -3 has 3 neighbors
Point -2.5 has 3 neighbors

Thus we have $\frac{3}{10} = 0.3 > 0.2$, and points -4.5, -4, -3, -2.5 are not outliers.

Point 0 has 0 neighbors

Thus we have $\frac{0}{10} = 0 < 0.2$, and point 0 is an outlier.

Point 3 has 4 neighbors
Point 3.5 has 4 neighbors
Point 4 has 4 neighbors
Point 4.5 has 4 neighbors
Point 5 has 4 neighbors

Thus we have $\frac{4}{10} = 0.4 > 0.2$, and points 3, 3.5, 4, 4.5, 5 are not outliers

Problem 3

a) The density-based outlier has a dynamic threshold to determine whether two points are neighbors.
However, distance-based approach has a fix threshold r, if r is not appropriately set. A point in a sparser cluster would be considered as an outlier, while density-based approach won't have such issue.

b) 1. The numerator of the equation a) means the sum of reachability distance from sample point o to all the other k nearest neighbor.
If we divide it by size of k-distance neighborhoods,
The whole equation means in o's k-nearest neighbors set, what's the average distance between point o and its neighbors.
If we take the reciprocal, we can explain it as:
In a distance unit area, how many k-nearest neighbors of point o would be inside.
That's exactly the definition of density.
2. That's if sample point o is closer to a cluster, the lrd(o) will get larger.

c) For $\sum_{o' \in N_k(o)} lrd_k(o')$, there no direct connection/comparison between target point p and its neighbors.
This equation can only determine
Whether o's neighborhood are in the dense clusters, while point o doesn't have to be in these clusters.
By taking ratio of their lrd valuek,
Now we can compare o's surrounding density and it's neighbor's surrounding density.
If such ratio is >1. It means o's neighbors are in a denser pattern compared to o, which means the target point o is isolated by it's neighbors, point o will be more likely to be an outlier.

Problem 4

a) a: local maximal
b: saddle point
c: cliff
d: high-cost local minimal

b) 1. $\delta_k = \frac{\partial L}{\partial I_k} = \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial I_k}$

$L = \frac{1}{2}(T - O_k)^2$

$O_k = \frac{1}{1+e^{-I_k}}$

$\delta_k = (O_k - T) * O_k * (1 - O_k)$

2. $\delta_j = \frac{\partial L}{\partial I_j}$

$O_i = \frac{1}{1+e^{-I_i}}$

$\delta_i = \frac{\partial L}{\partial I_i} = \sum_j \frac{\partial L}{\partial I_j} * \frac{\partial I_j}{\partial O_i} * \frac{\partial O_i}{\partial I_i}$

$= \sum_j (\delta_j * w_{ij}) * \frac{\partial O_i}{\partial I_i}$

$= O_i * (1 - O_i) * \sum_j (\delta_j * w_{ij})$

3. $\delta_k = \frac{\partial L}{\partial I_k} = \frac{\partial L}{\partial O_k} * \frac{\partial O_k}{\partial I_k}$

$L = -T\log O_k - (1-T)\log(1-O)$

$O_k = \frac{1}{1+e^{-I_k}}$

$\delta_k = -T_k * \frac{1}{O_k} * \frac{\partial O_k}{\partial I_K}$

$$\delta_k = \left(-T * \frac{1}{O_k} - (T-1) * \frac{1}{1-O_k}\right) * O_k * (1-O_k)$$

$$\delta_k = -T * (1-O_k) - (T-1) * O_k$$

$$\delta_k = -T + O_k T - O_k T + O_k$$

$$\delta_k = O_k - T$$

4. For the neuron k, we have error term $\delta_k = (O_k - T) * O_k * (1-O_k)$, we would find one part in δ_k, $(1-O_k)$, will lead δ_k very close 0 no matter what T is, when output is saturated, that's when O_k very close 1.
Similarly, the middle term, O_k, will lead δ_k very close 0 no matter what T is, when output is saturated, that is when O_k very close 0.
Since the gradient of parameters after neuron k will be a function of δ_k, this situation would lead gradient vanishing.
By applying cross-entropy, we have the error term $\delta_k = O_k - T$ instead. In such case δ_k will be 0 only when $O_k == T$.

Problem 5

a) k_1:
0 2 1 0
1 1 0 2
2 0 2 1
1 2 0 1
k_2:
2 1 1 2
1 2 1 2
1 2 1 2
3 1 1 1

b) Feature map 1, pooling:
1 3/4
5/4 1
Feature map 2, pooling:
6/4 6/4
7/4 5/4

c) $floor\left(\frac{N-K}{S}\right) + 1$
*
$floor\left(\frac{N-K}{S}\right) + 1$
*
L

d) $h_1 = \begin{matrix}0 & \mathbf{1}\\ 1 & 0\end{matrix} @ \begin{matrix}1\\0\end{matrix} + \begin{matrix}1 & \mathbf{1}\\ 0 & 1\end{matrix} @ \begin{matrix}1\\1\end{matrix}$

$= 0 \quad 1 + 2 \quad 1$

$= 2 \quad 2$

$h_2 = \begin{matrix}0 & \mathbf{1}\\ 1 & 0\end{matrix} @ \begin{matrix}1\\1\end{matrix} + \begin{matrix}1 & \mathbf{1}\\ 0 & 1\end{matrix} @ \begin{matrix}2\\2\end{matrix}$

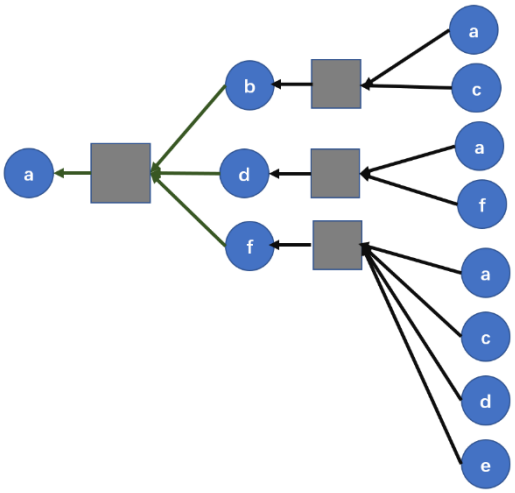
$= 1 \quad 1 + 4 \quad 2$

$= 5 \quad 3$

$\hat{y}_2 = \begin{matrix}1 & \mathbf{1}\\ 1 & 0\end{matrix} @ \begin{matrix}5\\3\end{matrix}$

$= 8 \quad 5$

e)



Problem6

a) Number of features: 27
Number of samples: 49534

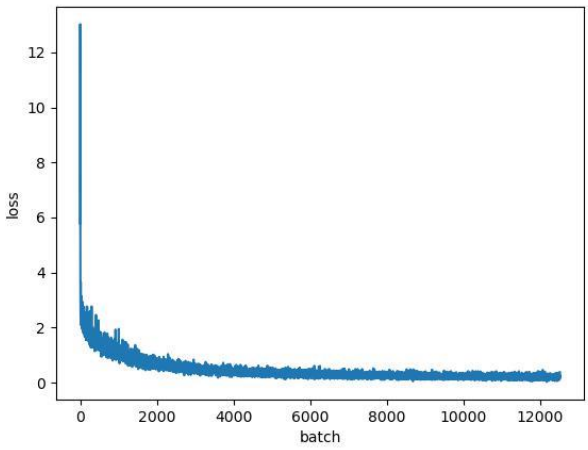
b) 0.92

c) 0.88

Problem7

a) Number of classes: 10
Number of training images: 50048
Number of test images: 10112
Image shape: 3 * 32 * 32

b)



Problem 8

a) First eigenvalue is closely related to the epidemic threshold as: $\tau = \frac{1}{\lambda}$; A larger first eigenvalue/vulnerability score, λ, leads to a smaller epidemic threshold, τ, that's an epidemic is more likely to occur in such network.
On the other hand, first eigenvalue is also closely related to the number of loops and paths of length l(l>=2). Obviously, if a graph has many such loops and paths, then it's well connected, and the virus would be easy to spread through such network.

b) It will be expensive to compute the first eigenvalue when there are too many nodes.
The 'Vulnerability' measure is not comparable between two graphs with different number of nodes.

c) Since according to the author's proof, the error of such approximation would only be: $\boldsymbol{O}(\sum_{j \varepsilon S} ||\boldsymbol{A}(:, \boldsymbol{j})||^2)$ when the eigen gap is not less than a function of number of best 'Shield-value' nodes and the max degree.
From the other hand, by observing the equation of the Shield-value:

$$\sum_{i \in S} 2\lambda u(i)^2 + \sum_{i,j \epsilon S} A(i,j)u(i)u(j)$$

Intuitively, a set of nodes would have a high shield value if a) They are very important in the graph (These nodes have high eigen value). b) They are diverse among themselves. These strategies make sense when we want to disconnect graph into isolated parts.
Finally, the 'Shield-value' is sub-modular (+monotonically non-decreasing), which allows us to find the near-optimal solution using greedy approach(much smaller time complexity!).

d) Time Complexity:

Algorithm 1. *NetShield*

Input: the adjacency matrix $\mathbf{A}$ and an integer k
Output: a set $\mathcal{S}$ with k nodes
1: compute the first eigenvalue λ of $\mathbf{A}$; let $\mathbf{u}$ be the corresponding eigenvector $\mathbf{u}(j)(j = 1, \ldots, n)$;
2: initialize $\mathcal{S}$ to be empty;
3: **for** $j = 1$ to n **do**
4: $\quad \mathbf{v}(j) = (2 \cdot \lambda - \mathbf{A}(j, j)) \cdot u(j)^2$;
5: **end for**
6: **for** iter $= 1$ to k **do**
7: $\quad$ let $\mathbf{B} = \mathbf{A}(:, \mathcal{S})$;
8: $\quad$ let $\mathbf{b} = \mathbf{B} \cdot \mathbf{u}(\mathcal{S})$;
9: $\quad$ **for** $j = 1$ to n **do**
10: $\quad\quad$ **if** $j \in \mathcal{S}$ **then**
11: $\quad\quad\quad$ let score$(j) = -1$;
12: $\quad\quad$ **else**
13: $\quad\quad\quad$ let score$(j) = \mathbf{v}(j) - 2 \cdot \mathbf{b}(j) \cdot \mathbf{u}(j)$;
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: $\quad$ let $i = \text{argmax}_j \text{score}(j)$, add i to set $\mathcal{S}$;
17: **end for**
18: return $\mathcal{S}$.

The time complexity of step 1 would be O(m), m would be number of edges in the graph.
The time complexity of step2 would be O(1).
The time complexity of steps 3-5 would be O(n), n would be number of nodes in the graph.
Under for the for loop in step 6:

The time complexity of step 16 would be O(n * iter).

The time complexity of step steps 9-15 would be O(n).

Thus the time complexity of steps 6-17 would be:

$$\sum_{iter=1}^{k} (n + n * iter) = O(nk^2)$$

Thus the time complexity of the entire algorithm would be:

$$O(nk^2 + m)$$

Space Complexity:
The space complexity of step1 would be O(n+m+1) to do eigen decomposition:

O(m) to store the graph matrix, O(n) to store the eigenvectors, O(1) to store the eigen value.

The space complexity of step2 would be O(1)
The space complexity of step3-5 would be O(n) to store 'Shield-value' score of each node.
The space complexity of step6-17 would be O(n+k):

O(n) to store the search score of each node, O(k) to store the result S.

Thus the space complexity of such algorithm would be O(n+m+k).

e) The difference is that instead of only computing first eigen pairs once and finding all k best 'Shield-value' nodes with this fixed eigen pairs, NetShield+ will update eigen pairs everytime b(<=k) best 'Shield-value' nodes are found and deleted from the graph, and use the updated eigen pairs to do computations.

NetShield+ solved the performance deterioration issue of NetShield when the max degree of the graph is relatively small.

NetShield+ is also more computationally efficient than NetShield.

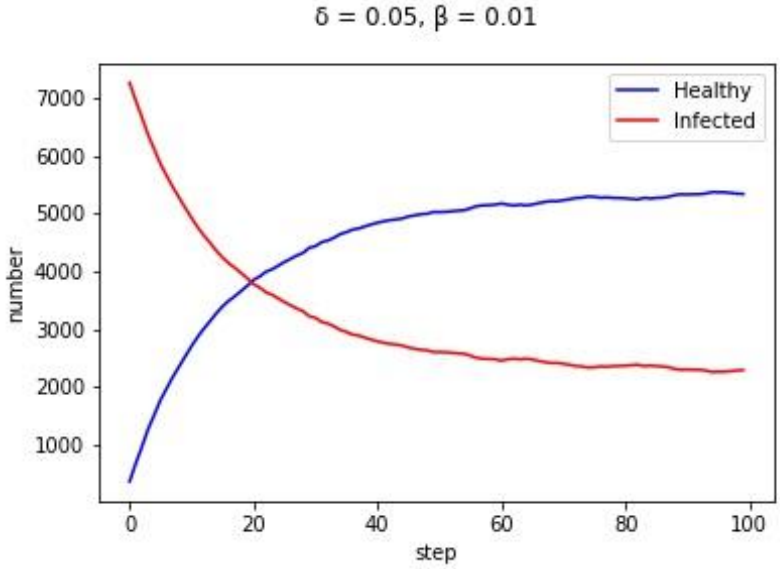
Problem 9

a) $1-(1-\beta)^n$

b) 38.60

c) Yes, the result aligns with the theory.

Since $\frac{\beta\lambda}{\delta}=\frac{0.01*38.60}{0.05}=\mathbf{7.72>1}$, there will be an epidemic.



d) Since $\frac{\beta\lambda}{\delta}=\frac{0.01*38.60}{0.4}=\mathbf{0.96<1}$, there will no epidemic.