Name: Zong Fan, NetID: zongfan2

Q1: (a) $I_4 = \cancel{\sigma(I_1)+\sigma(I_2)}\; w_{14}\sigma(I_1) + w_{24}\sigma(I_2) + w_{34}\sigma(I_3)$.

(b)

$$\delta_6 = \frac{\partial L}{\partial I_6} = \frac{\partial L}{\partial O_6} \cdot \frac{\partial O_6}{\partial I_6}$$

$$= \frac{\partial \frac{1}{2}(O_6 - T)^2}{\partial O_6} \cdot \frac{\partial \frac{1}{1+e^{-I_6}}}{\partial I_6}$$

$$= (O_6 - T) \cdot \sigma(I_6) \cdot (1 - \sigma(I_6))$$

$$= O_6 (O_6 - T)(1 - O_6).$$

(c).

$$\delta_4 = \frac{\partial L}{\partial I_4} = \cancel{\frac{\partial L}{\partial I_6} \cdot \frac{\partial O_6}{\partial I_4} = \frac{\partial L}{\partial O_6} \cdot \frac{\partial(\sigma(I_6))}{\partial I_4}}$$

$$= \cancel{\frac{\partial L}{\partial O_6}}$$

$$= \frac{\partial L}{\partial I_6} \cdot \frac{\partial I_6}{\partial I_4} = \frac{\partial L}{\partial I_6} \cdot \frac{\partial(\sigma(I_4) \cdot w_{46} + \sigma(I_5) \cdot w_{56})}{\partial I_4}$$

$$= \delta_6 \cdot w_{46} \cdot \frac{\partial \sigma(I_4)}{\partial I_4}$$

$$= \delta_6 \cdot w_{46} \cdot \sigma(I_4) \cdot (1 - \sigma(I_4))$$

$$= w_{46} \cdot \delta_6 \cdot O_4 \cdot (1 - O_4)$$

(d). For the input layer, the combination of neurons can be:

1. All neurons are activated: $n=1$
2. 1 neuron is inactive: $n=3$. $\Rightarrow N_{in} = 1+3+3 = 7$.
3. 2 neurons are inactive: $n=3$.

For the hidden layer, similarly, $N_{hidden} = 2+1 = 3$. So the total number of distinct network is $7 \times 3 = 21$

Q2: (a):

| | | |
|---|---|---|
| 2 | 1 | 3 |
| 2 | 3 | 0 |

(b) After 1st pooling, there is one conv + 1 pooling (sigmoid activation doesn't change shape)

After conv, $S_1 = \text{Int}(\frac{512-3}{2})+1 = 255$.

After pooling, $S = \text{Int}(\frac{255-2}{2})+1 = 127$

similarly, after 2nd pooling, $S = \text{Int}\left(\frac{\text{Int}(\frac{127-3}{2})+1-2}{2}\right)+1$

$= 31$

If the input size is $1\times1$, Conv cannot be used without padding.

So $N=0$.

(c). Since sigmoid activation and maxpooling don't have trainable parameters, all parameters are from the conv layers.

For conv with kernel $3\times3$, the trainable parameters for a complex layer is $3\times3\times1\times1=9$.

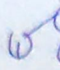
So the total parameters are $9N$.

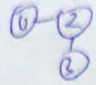
(d). For MLP, $N = 5\times4 = 20$.

For CNN. kernel size 3, $N = 3\times1 = 3$.

For RNN, because a, b, c, d are shared for all hidden state

$N = 4$

Q3 : (a) 0 +  . cutsize = $\frac{1}{4}(4+4) = 2$.

1 +  : cut size = $\frac{1}{4}(4+4) = 2$.

3 +  = CutSize = $\frac{1}{4}(4) = 1$.

 +  : cut size = $\frac{1}{4}(4+4) = 2$.

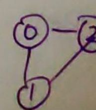
(b): $A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \Rightarrow D = \sum_j A_{ij} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

graph laplacian $L = D - A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$

(c) The second minimum eigenvalue is 1.

According to its eigenvector, the best partition way is

0, 1, 2 $\in \{+1\}$.

3 $\in$ ~~[illegible]~~ $\{-1\}$. $\Rightarrow$ 3 +  with the min CutSize value.

(d). Because the graph has two subgraphs without any connections between them, assuming the left subgraph is A and the right subgraph is B; Cutsize(A,B) = 0.

In this case, it eigenvalues is also the minimum, which is 0.

Q4 = (a). Distance-based, because it checks if the ~~Samp~~ sample has enough number of neighbors

(b). O is the outlier. because it has no neighbor whose distance is within 2. The fraction is $\frac{0}{5} = 0 < 0.1$

Except for O, all other nodes have 1 ~~neighbors~~,

The fraction is $\frac{1}{5} = 0.2 > 0.1$. Not outliers.

(c). ~~No~~ Doesn't work.

For the samples on the right side, they are also sparsely distributed. With ~~the~~ the same distance threshold,

~~It's~~ It's highly possible that these sample don't have enough number of neighbors. They can be treated as outliers.

(d). 1. If data is generated from one gaussian distribution,

$O_1$ is the detected outlier. ~~It's because~~.

This is because the middle points $O_2, O_3, O_4$ should come from the same distribution of the left and right sample cluster. They can not be distinguished as outliers.

2. If they come from 2 Gaussian distributions, both $O_1, O_2, O_3, O_4$ ~~can~~ can be detected., because the left and right sample clusters can be approximated by two gaussian distributions. For $O_2, O_3, O_4$, their probabilities of coming from any gaussian distribution can be low.

# 1 Spectral clustering

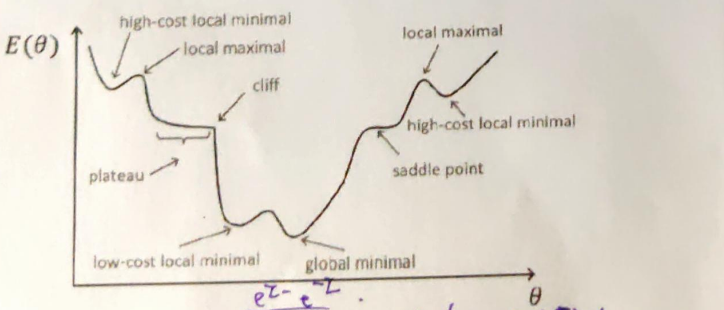
**MinCut**: $q = \underset{q\in\{-1,1\}^n}{argmin}\ CutSize; CutSize = \frac{1}{4}\sum_{i,j}(q_i - q_j)^2 w_{i,j}$
**Relaxation**: 1. relax q to be real number $J = q^T(D-W)q; d_i = \sum_j w_{i,j}, D = [d_i\delta_{i,j}] \to q^* = argmin q^T(D-W)q, s.t. \sum_k q_k^2 = n$. The solution is the second minimum eigenvector for $D-W$.
**Graph Laplacian**: $L = D - W; w = [w_{i,j}], D = [\delta_{i,j}(\sum_j w_{i,j})]$. L is semi-positive definitive matrix ($x^T Lx = x^T Dx - x^T Wx = \sum_{i=1}^n d_i x_i^2 - \sum_{i,j=1}^n w_{i,j} f_i f_j) = 0.5(\sum_{i=1}^n d_i x_i^2 - 2\sum_{i,j=1}^n w_{i,j}x_i x_j + \sum_{j=1}^n d-jx_j^2) = 0.5(\sum_{i,j=1}^n w_{i,j}(f_i - f_j)^2) \geq 0$ and min eigenvalue is 0 (eigenvector is $[1, ..., 1]^T$. For $\mathbf{Dv}$, the value at $i$th row is $\sum_j w_{i,j}$, which picks the degree of node i from the diagonal degree matrix $\mathbf{D}$. For $\mathbf{Av}$, the value at $i$th row is also $\sum_j w_{i,j}$. Therefore, $(\mathbf{D} - \mathbf{A})\mathbf{v} = 0$ is always satisfied and $\mathbf{v}$ is the eigenvector). Parition based on the eigenvector: $A = \{i | q_i < 0\}$
**Spectral clustering**: mincut doesn't balance the size of bi-partite graph. computational complement $Cut(A, B) = \sum_{i\in A; j\in B} w_{i,j}$ and $Vol(A) = \sum_{i\in A}\sum_{j=1}^n w_{i,j}$ Obj1: min inter-cluster connection (min cut(A,B)); Obj2: max intra-cluster connection: max vol(A,A) and vol(B,B). $J = Cut(A, B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})$. Solution: 2nd smallest eigenvector of $(D - W)y = \lambda Dy$

# 2 Feed forward NN

Why multiple layers: Automatic feature learning; Learn non-linear mapping function. Process: feed forward; compute gradient $\frac{\partial}{\partial\theta}J_\theta$: update parameter: $\theta = \theta - \eta\frac{\partial}{\partial\theta}J_\theta$
**BP**: error term $\delta_j^{(l)}$ is a function of (1): all $\delta_k^{(l+1)}$ in the layer l+1, if layer l is hidden layer; (2) the overall loss function value, if layer l is the output layer

# 3 Deep learning

**Challenges**: optimization is non-convex (find high-quality local optima); generalization: min generalization error (reduce overfitting)




$tanh = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ $tanh = 2\sigma(2z) - 1$.

**Responsive activation function:** saturation of sigmoid: $O = \sigma(I) = \frac{1}{1+exp(-I)}$; derivative: $\frac{\partial O}{\partial I} = O(1-O)$; error: $\delta_j = O_j(1-O_j)(O_j - T_j)$. If $O_j$ is close to 0 or 1, bother derivative and error is close 0 (gradient vanishing).
ReLU: $O = I if I > 0, otherwise 0$. No decaying in error, avoid gradient vanishing.
**Adaptive learning rate**: SGD $\theta_{t+1} = \theta_t - \eta g_t$. Potential problems: slow progress, jump over gradient cliff; oscillation. Strategy: 1. $\eta_t = \frac{1}{t}\eta_0$; 2. $\eta_t = (1 - t/T)\eta_0 + t/T\eta_\infty$; 3. AdaGrad: $\eta_t = \frac{1}{\rho + r_t}\eta_0, r_t = \sqrt{\sum_{k=1}^{t} - 1 g_{i,k}^2}$. Intuition: the magnitude of gradient $g_t$ as the indicator of overall progress.
**Dropout**: to prevent overfitting by randomly dropout of some non-output units. Regularization; Force the model to be more robust to noise, and to learn more generalizable features. VS bagging: each model is trained independently, while the model of current dropout network are updated based on previous dropout network.
**Pre-training**: the process of initializing the model in a suitable region. Greedy supervised pretraining; pre-set model parameters layer-by-layer in a greedy way; unsupervised pretraining: auto-encoder; hybrid.
**Cross-entropy**: MSE for regression. CE Loss $-Tlog(O) - (1 - T)log(1 - O)$; error: $O - T$

# 4 CNN

Challenges of MLP: Long training time, slow convergence, local minima. Motivation: Sparse interactions (Units in deeper layers still connect to a wide range of inputs); Parameter sharing (Reduce parameters); Translational equivalence $f(g(x)) = g(f(x))$. CNN layer followed by non-linear activation and pooling. The deeper the better: learn from a larger receptive field.
**Pooling**: Introduces invariance to local translations; Reduces the number of hidden units in hidden layer

# 5 RNN

Handle sequence. $h^t = f(Ux^t, Wh^{t-1} + a)$; $\hat{y}^T = g(Vh^T + b)$. Recurrence to capture long-term dependence: same hidden-to-hidden matrix W; same input-to-hidden matrix U, same bias a. VS CNN: localized dependence.
Challenges: long-term dependence. It needs deep RNN, leading to gradient vanishing or exploding. Solution: Gated RNN (LSTM, GRU) or attention mechanism.
**LSTM**: cell state; accumulate the information from the past; three gate to control info flow (forget; input; output).

control what info to update for hidden state. RNN: gradient vanishing infor from previous disapp

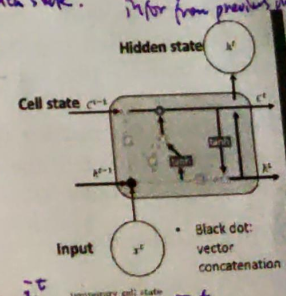
learn what info to keep . to discard

**LSTM Cell**

- Three Gates: control info flow (blown)
  - Forget gate $f^t = \sigma(W_f[x^t, h^{t-1}] + a_f)$
  - Input gate $i^t = \sigma(W_i[x^t, h^{t-1}] + a_i)$
  - Output gate $o^t = \sigma(W_o[x^t, h^{t-1}] + a_o)$
- Temporary Cell State (middle, blue)

$$\tilde{C}^t = \sigma(\mathbf{W}_c[\mathbf{x}^t, \mathbf{h}^{t-1}] + \mathbf{a}_c)$$

- Cell state update (upper) $C^t = \underbrace{f^t}_{\text{forget gate}} \cdot \overbrace{C^{t-1}}^{\text{old cell state}} + \underbrace{i^t}_{\text{input gate}} \cdot \overbrace{\tilde{C}^t}^{\text{temporary cell state}}$ — $c^t$ $f^t$ $C^{t-1}$ $i^t$ $\tilde{c}^t$
- Hidden state update (top right) $h^t = \underbrace{o^t}_{\text{output gate}} \cdot \underbrace{\tilde{h}^t}_{\text{temporary hidden state}}$




**Attention**: Key Idea of Attention Mechanism: context vectors. Augment hidden state of 2nd RNN with context vectors. Introduce an alignment vector a and use linear weighted sum to obtain context vector.

## 6 GNN

Challenges: Irregular graph structure (non-Euclidean): Unfixed size of node neighborhoods; Permutation invariance: Node ordering does not matter; Undefined convolution computation
**Graph convolution in spectral domain**: spectral-based model (GCN): $x *_{\theta} y \approx \theta(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2})x$, $\tilde{A} = A + I_n$ transductive learning

$$Z = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta \longrightarrow \text{Parameters}$$

Output features

Adjacency matrix with self-loops

Input features

A two-layer architecture for node classification: $\tilde{Y} = softmax(\hat{A}\sigma(\hat{A}X\theta_1)\theta_2)$, $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$
**Graph convolution in spatial domain**: $x(i) = w_{i,i}x(i)\sum_{j\in(i,k)} w_{i,j}x(j)$, where N is the k-hop neighborhood. key idea: message passing: how to aggregate node representations.

## 7 Outlier

Global Outliers (=point anomalies); Contextual Outliers (=conditional outliers); Collective Outliers (=group anomaly) Challenge: Difficulty in modeling normality, ambiguity between normal and abnormal. Application-specific outlier detection; noise vs outlier (Noise: unavoidable, less interesting to the users, but make outlier detection more challenge); model interpretability.
**Statistical approaches**: Assume normal data are generated by a stochastic process Data objects in low density regions are flagged as outlier. Parametric Methods: The normal data objects are generated by a parametric distribution with a finite number of parameters: Single Variable Data: Grubb's test; Multi variable Data: Mahalanobis distance; $\chi^2$-statistics; mixture models Non Parametric Methods: Do not assume a priori statistical model with a finite number of parameters: Outlier Detection by Histogram (Construct histogram data objects outside bins are outliers); Outlier Detection by Kernel Density Estimation (Kernel function: influence of a sample within its neighbor)
**Proximity-based approaches**: Intuition: objects that are far from others can be regarded as outliers. Assumption: the proximity of an outlier object to its nearest neighbors significantly deviates from the proximity of most other objects to their nearest neighbors
**Distance-based outlier detection**: Consult the neighborhood of a sample. Outlier: if there are not enough objects in its neighborhood. r: distance threshold; $\pi$: fraction threshold. o is a $DB(r, \pi)$ -outlier if $\frac{\|\{o'|dist(o, o') \leq r\}\|}{\|D\|} \leq \pi$. Equivalent criteria: if $dist(o, o_k) > r$. $o_k$ is the k-nearest neighbor of o; $k = \lceil \pi\|D\| \rceil$