ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΒΑΘΙΑ ΜΑΘΗΣΗ

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2024 ΣΕΙΡΑ ΑΝΑΛΥΤΙΚΩΝ ΑΣΚΗΣΕΩΝ

ΚΥΡΙΑΚΗ ΚΑΡΑΤΖΟΥΝΗ 03120634

Email: el20634@mail.ntua.gr

Άσκηση 1

1. Περίπτωση Α:

Παρατηρούμε ότι η καμπύλη της συνάρτησης κόστους για το σύνολο επικύρωσης (Validation loss) αρχικά μειώνεται αλλά στη συνέχεια αρχίζει να αυξάνεται ξανά, ενώ η καμπύλη της συνάρτησης κόστους για το σύνολο εκπαίδευσης (Training loss) συνεχίζει να μειώνεται.

Έχουμε overfitting, καθώς για μεγάλο πλήθος εποχών έχουμε μεγάλη απόκλιση μεταξύ των validation και training loss. Το μοντέλο έχει καλή απόδοση στα δεδομένα εκπαίδευσης αλλά χειροτερεύει στα δεδομένα επικύρωσης καθώς συνεχίζεται η εκπαίδευση.

Περίπτωση Β:

Στην Περίπτωση Β, παρατηρούμε ότι τόσο η συνάρτηση κόστους για το σύνολο εκπαίδευσης (Training loss) όσο και για το σύνολο επικύρωσης ((Validation loss) μειώνονται συνεχώς κατά τη διάρκεια των επαναλήψεων (iterations).

Το μοντέλο γενικεύει καλά, αποδίδοντας παρόμοια τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα επικύρωσης, γεγονός που υποδηλώνει ότι το μοντέλο δεν υπερπροσαρμόζεται. Υπάρχει μικρή απόκλιση για όλες τις εποχές.

2. Περίπτωση Α:

Για την περίπτωση Α θέλουμε μια τιμή κοντά στο μέσο των εποχών που φαίνεται στο διάγραμμα, καθώς μετά από αυτήν γίνεται το overfitting. Αυτό το σημείο υποδεικνύει ότι το μοντέλο έχει μάθει καλά από τα δεδομένα εκπαίδευσης χωρίς να υπερπροσαρμοστεί.

Περίπτωση Β:

Για την περίπτωση Β, θέλουμε επίσης να επιλέξουμε μια τιμή κοντά στον μέσο όρο των εποχών, γιατί το loss μειώνεται με πολύ αργό ρυθμό. Αυτό σημαίνει ότι μπορούμε να εξοικονομήσουμε υπολογιστικούς πόρους χωρίς να έχουμε σημαντική διαφορά στο loss.

3. α. Κανονικοποίηση (Regularization):

Η κανονικοποίηση προσθέτει ένα επιπλέον όρο στο κόστος που ενθαρρύνει το μοντέλο να έχει μικρότερες τιμές παραμέτρων, μειώνοντας την πολυπλοκότητά του και αποτρέποντας την υπερπροσαρμογή.

β. Dropout:

Το dropout είναι μια τεχνική που εισάγει τυχαία "απόσβεση" σε ορισμένους νευρώνες κατά τη διάρκεια της εκπαίδευσης. Αυτό σημαίνει ότι κατά τη διάρκεια κάθε επανάληψης, κάποιο ποσοστό των

νευρώνων δεν θα είναι ενεργοί. Αυτό αποτρέπει το μοντέλο από το να βασίζεται υπερβολικά σε συγκεκριμένους νευρώνες και ενθαρρύνει τη μάθηση πιο γενικών χαρακτηριστικών.

4. Το σύνολο ελέγχου (testing set) είναι απαραίτητο πέρα από τα σύνολα εκπαίδευσης και επικύρωσης για την Αντικειμενική Αξιολόγηση σε Άγνωστα Δεδομένα.Το σύνολο ελέγχου βοηθά στον αντικειμενικό έλεγχο του μοντέλου σε άγνωστα δεδομένα, εξασφαλίζοντας ότι η απόδοση που παρατηρούμε είναι αντιπροσωπευτική της πραγματικής ικανότητας του μοντέλου να χειριστεί νέα δεδομένα που δεν έχουν χρησιμοποιηθεί στη διαδικασία εκπαίδευσης και επικύρωσης.

Άσκηση 2

α. Η διάσταση των χαρακτηριστικών εισόδου xi στον auto-encoder είναι η διάσταση των διανυσμάτων που παράγονται από το skipgram μοντέλο. Δεδομένου ότι η διάσταση των διανυσμάτων uo και uc είναι 256×1, αυτό σημαίνει ότι η διάσταση των χαρακτηριστικών εισόδου xi στον auto-encoder είναι 256.

β. Εφόσον η διάσταση των χαρακτηριστικών εισόδου είναι 256, η διάσταση των χαρακτηριστικών εξόδου θα είναι επίσης 256.

γ. Η λανθάνουσα αναπαράσταση του auto-encoder βρίσκεται στο κεντρικό στρώμα. Έχουμε τα παρακάτω στρώματα διαστάσεων:

[500, 250, 50, 250, 500]

Επομένως, η διάσταση της λανθάνουσας αναπαράστασης του αυτοκωδικοποιητή είναι 50.

Άσκηση 3

α. Υπολογισμός του c:

$$c = \frac{1}{4}\sum_{j=1}^{4} i_j = \frac{1}{4}\left([4,0]^T + [0,4]^T + [0,0]^T + [0,0]^T\right) = \frac{1}{4}\left([4,4]^T\right) = [1,1]^T$$

Αρχικοποίηση της κρυφής κατάστασης h0 και της εισόδου x1:

$$h_0 = [0,0]^T$$

$$y_0 = y_{<start>} = [0,0,0]^T$$

$$x_1 = [y_0; c] = \left[[0,0,0]^T; [1,1]^T\right] = [0,0,0,1,1]^T$$

X1 = concat(<start>, c1)

Υπολογισμός του h1:

$$h_1 = \text{ReLU}(Whx \cdot x_1 + Whh \cdot h_0) = \text{ReLU}\left(\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right)$$

$$h_1 = \text{ReLU}\left(\begin{bmatrix} 0+0+0+1+0 \\ 0+0+0+0+1 \end{bmatrix} \right) = \text{ReLU}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Υπολογισμός της εξόδου y1:

$$y_1 = \text{argmax}(Wyh \cdot h_1)$$

$$Wyh \cdot h_1 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ 3 \\ 3 \\ 4 \\ 2 \\ 2.9 \end{bmatrix}$$

Η μέγιστη τιμή είναι το 4 και από το λεξιλόγιο είναι η λέξη "cat."

Ορισμός της πρώτης λέξης y1 σε "cat" και υπολογισμός του x2:

$$y_1 = y_{cat} = [1, -2, 0]^T$$

$$x_2 = [y_1; c] = \left[[1, -2, 0]^T; [1, 1]^T \right] = [1, -2, 0, 1, 1]^T$$

Υπολογισμός του h2:

$$h_2 = \text{ReLU}(Whx \cdot x_2 + Whh \cdot h_1) = \text{ReLU}\left(\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)$$

$$Whx \cdot x_2 = \begin{bmatrix} 1+0+0+1+0 \\ 0+(-2)+0+0+1 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$$Whh \cdot h_1 = \begin{bmatrix} 0+1 \\ 1+0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$h_2 = \text{ReLU}\left(\begin{bmatrix} 2 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \text{ReLU}\left(\begin{bmatrix} 3 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Υπολογισμός της εξόδου y2:

$$y_2 = \text{argmax}(Wyh \cdot h_2)$$

$$Wyh \cdot h_2 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -15 \\ 0 \\ 3 \\ 6 \\ 9 \\ 8.7 \end{bmatrix}$$

Η μέγιστη τιμή είναι το 9 και από το λεξιλόγιο είναι η λέξη "staring."

Ορισμός της δεύτερης λέξης y2 σε "staring" και υπολογισμός του x3:

$$y_2 = y_{staring} = [0, -1, -1]^T$$

$$x_3 = [y_2; c] = \left[[0, -1, -1]^T; [1, 1]^T \right] = [0, -1, -1, 1, 1]^T$$

Υπολογισμός του h3:

$$h_3 = \text{ReLU}(Whx \cdot x_3 + Whh \cdot h_2) = \text{ReLU}\left(\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} \right)$$

$$Whx \cdot x_3 = \begin{bmatrix} 0+0+(-1)+1+0 \\ 0+(-1)+(-1)+0+1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$Whh \cdot h_2 = \begin{bmatrix} 0+0 \\ 3+0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$h_3 = \text{ReLU}\left(\begin{bmatrix} 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix} \right) = \text{ReLU}\left(\begin{bmatrix} 0 \\ 2 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

Υπολογισμός της εξόδου y3:

$$y_3 = \text{argmax}(Wyh \cdot h_3)$$

$$Wyh \cdot h_3 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -10 \\ 6 \\ 4 \\ 4 \\ 1 \\ 0 \end{bmatrix}$$

Η μέγιστη τιμή είναι το 6 και από το λεξιλόγιο είναι το "<stop>".

Άρα έχουμε: <start> cat staring <stop>

β. Αρχικοποίηση του h0:

$$h_0 = [0, 0]^T$$

Υπολογισμός των ακατέργαστων βαθμολογιών προσοχής (attention scores): Οι βαθμολογίες προσοχής υπολογίζονται ως το εσωτερικό γινόμενο του ερωτήματος h_{t-1} (δηλαδή του h0 για το πρώτο χρονικό βήμα) με κάθε αναπαράσταση χαρακτηριστικών i_j:

$$\text{score}_{ij} = h_{t-1}^T \cdot i_j$$

$$\text{score}_{i1} = [0,0]^T \cdot [4,0]^T = 0$$

$$\text{score}_{i2} = [0,0]^T \cdot [0,4]^T = 0$$

$$\text{score}_{i3} = [0,0]^T \cdot [0,0]^T = 0$$

$$\text{score}_{i4} = [0,0]^T \cdot [0,0]^T = 0$$

Υπολογισμός των κανονικοποιημένων πιθανοτήτων προσοχής (attention probabilities):
Χρησιμοποιούμε την softmax συνάρτηση για να κανονικοποιήσουμε τις βαθμολογίες προσοχής:

$$\alpha_j = \frac{\exp(\text{score}_{ij})}{\sum_{k=1}^{4} \exp(\text{score}_{ik})}$$

Επειδή όλες οι βαθμολογίες είναι μηδενικές:

$$\alpha_j = \frac{\exp(0)}{\sum_{k=1}^{4} \exp(0)} = \frac{1}{4} = 0.25$$

Υπολογισμός του διανύσματος εικόνας c1: Το διάνυσμα εικόνας c1 είναι ένας σταθμισμένος μέσος όρος των αναπαραστάσεων των χαρακτηριστικών, χρησιμοποιώντας τις πιθανότητες προσοχής:

$$c_1 = \sum_{j=1}^{4} \alpha_j i_j = 0.25 \cdot [4,0]^T + 0.25 \cdot [0,4]^T + 0.25 \cdot [0,0]^T + 0.25 \cdot [0,0]^T$$

$$c_1 = 0.25 \cdot [4,0]^T + 0.25 \cdot [0,4]^T = [1,0]^T + [0,1]^T = [1,1]^T$$

Η μέγιστη τιμή είναι το 4 και από το λεξιλόγιο είναι η λέξη "cat."

Άρα, η πρώτη λέξη παραμένει ίδια, δηλαδή "cat".

Άσκηση 4

α.

$$W_{\text{out}} = \frac{W_{\text{in}} - F + 2P}{S} + 1$$
$$H_{\text{out}} = \frac{H_{\text{in}} - F + 2P}{S} + 1$$

Wout και Hout είναι οι διαστάσεις της εξόδου (πλάτος και ύψος αντίστοιχα),

Win και Hin είναι οι διαστάσεις της εισόδου (πλάτος και ύψος αντίστοιχα),

$$W_{\text{out}} = \frac{227 - 11 + 2 \cdot 0}{4} + 1 = \frac{216}{4} + 1 = 54 + 1 = 55$$
$$H_{\text{out}} = \frac{227 - 11 + 2 \cdot 0}{4} + 1 = \frac{216}{4} + 1 = 54 + 1 = 55$$

Άρα, οι διαστάσεις της εξόδου του πρώτου convolutional layer θα είναι 55×55.

Δεδομένου ότι υπάρχουν 96 φίλτρα, η τελική έξοδος του πρώτου convolutional layer θα έχει διαστάσεις 55×55×96.

β. Ο αριθμός των units υπολογίζεται ως το γινόμενο αυτών των διαστάσεων:

55×55×96 = 290400

γ. Ο αριθμός των παραμέτρων για κάθε φίλτρο υπολογίζεται ως το γινόμενο του μεγέθους του φίλτρου και του αριθμού των καναλιών εισόδου:

Παράμετροι ανά φίλτρο=11×11×3 =363

Κάθε φίλτρο έχει επίσης μία παράμετρο bias, οπότε οι συνολικές παράμετροι ανά φίλτρο είναι:

Παράμετροι ανά φίλτρο με bias = 363+1 = 364

Τέλος, επειδή υπάρχουν 96 φίλτρα, ο συνολικός αριθμός των παραμέτρων υπολογίζεται ως:

Συνολικές παράμετροι = 364×96 = 34944

δ. Η είσοδος του FeedForward layer είναι το επίπεδο της εικόνας, δηλαδή:

227×227×3=154587

Ο αριθμός των εκπαιδεύσιμων παραμέτρων σε ένα FeedForward layer είναι:

input size×units+units

Άρα: 154587×256+256=39574272+256=39574528

Συνεπώς, θα είχαμε 39574528 εκπαιδεύσιμες παραμέτρους.

Άσκηση 5

Variational Autoencoders: Χρησιμοποιούν έναν κωδικοποιητή και έναν αποκωδικοποιητή για τη δημιουργία νέων δεδομένων. Εισάγουν κανονικοποίηση στον λανθάνοντα χώρο για να αποφεύγουν την υπερπροσαρμογή και να εξασφαλίζουν μια συνεχή και πλήρη δομή του λανθάνοντα χώρου. Ο κωδικοποιητής συμπιέζει τα δεδομένα στον λανθάνοντα χώρο, ενώ ο αποκωδικοποιητής ανασυγκροτεί τα αρχικά δεδομένα από αυτόν τον χώρο.

Generative Adversarial Networks: Αποτελούνται από δύο νευρωνικά δίκτυα: μια γεννήτρια και έναν διαχωριστή, που ανταγωνίζονται μεταξύ τους. Η γεννήτρια δημιουργεί ψεύτικα δεδομένα, ενώ ο διαχωριστής προσπαθεί να τα διακρίνει από τα πραγματικά. Η γεννήτρια εκπαιδεύεται να παράγει δεδομένα που μπορούν να ξεγελάσουν τον διαχωριστή, ενώ ο διαχωριστής εκπαιδεύεται να αναγνωρίζει τα ψεύτικα δεδομένα.

Diffusion Models: Βασίζονται σε διαδικασίες διάχυσης και αντιστρόφου διάχυσης θορύβου για τη δημιουργία νέων δειγμάτων. Προσθέτουν θόρυβο σταδιακά στα δεδομένα και μαθαίνουν να αντιστρέφουν αυτή τη διαδικασία. Ο θόρυβος προστίθεται σταδιακά στα δεδομένα σε μια προωθητική διαδικασία και αφαιρείται σε μια αντίστροφη διαδικασία διάχυσης.

Διαδικασία Εκπαίδευσης:

Variational Autoencoders: Η διαδικασία εκπαίδευσης περιλαμβάνει τη βελτιστοποίηση του κωδικοποιητή και του αποκωδικοποιητή για να ελαχιστοποιηθεί το σφάλμα ανακατασκευής. Η κανονικοποίηση εισάγεται στον λανθάνοντα χώρο για να διατηρηθεί η δομή και να αποφευχθεί η υπερπροσαρμογή.

Generative Adversarial Networks: Εκπαιδεύουν τη γεννήτρια και τον διαχωριστή σε εναλλασσόμενες περιόδους εκπαίδευσης. Ο διαχωριστής εκπαιδεύεται για να διακρίνει μεταξύ πραγματικών και ψεύτικων δεδομένων, ενώ η γεννήτρια εκπαιδεύεται για να παραπλανά τον διαχωριστή. Η σύγκλιση είναι δύσκολη και απαιτεί προσεκτική ρύθμιση των υπερπαραμέτρων.

Diffusion Models: Απαιτούν μεγάλο αριθμό βημάτων εκπαίδευσης (συνήθως μερικές χιλιάδες) για να παράγουν δείγματα υψηλής ποιότητας. Χρησιμοποιούν δυναμικές Langevin και προσδιοριστικές διαφορικές εξισώσεις για τη δημιουργία δειγμάτων.

Αποτελεσματικότητα και Απαιτήσεις:

Οι Variational Autoencoders απαιτούν λιγότερους πόρους για εκπαίδευση συγκριτικά με τα GANs. Ωστόσο, ενδέχεται να παράγουν δείγματα χαμηλότερης ποιότητας. Τα Generative Adversarial Networks απαιτούν περισσότερη μνήμη και χρόνο εκπαίδευσης, με τη διαδικασία σύγκλισης να είναι συχνά δύσκολη και ασταθής. Παρά τις προκλήσεις, όταν εκπαιδευτούν σωστά, μπορούν να παράγουν εξαιρετικά ρεαλιστικά δείγματα. Τα Diffusion Models παράγουν δείγματα υψηλής ποιότητας, αλλά η διαδικασία δειγματοληψίας τους είναι αργή και απαιτεί πολλές αξιολογήσεις του νευρωνικού δικτύου, καθιστώντας τα δύσκολα για εφαρμογές σε πραγματικό χρόνο.

Ομοιότητες και Διαφορές:

Όλα αυτά τα μοντέλα χρησιμοποιούν νευρωνικά δίκτυα για τη δημιουργία δεδομένων και στοχεύουν στην εκμάθηση κατανομών δεδομένων για τη δημιουργία ρεαλιστικών δειγμάτων. Οι διαφορές τους

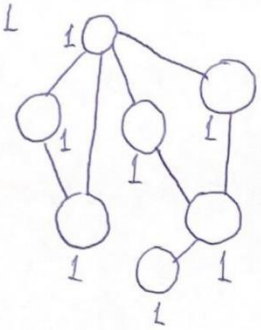
περιλαμβάνουν την προσέγγισή τους: οι Variational Autoencoders χρησιμοποιούν κωδικοποίηση και αποκωδικοποίηση με κανονικοποίηση του λανθάνοντα χώρου, τα Generative Adversarial Networks βασίζονται σε έναν ανταγωνισμό μεταξύ γεννήτριας και διαχωριστή, ενώ τα Diffusion Models προσθέτουν και αφαιρούν θόρυβο στα δεδομένα σε μια διαδικασία διάχυσης.

Πλεονεκτήματα και Μειονεκτήματα:

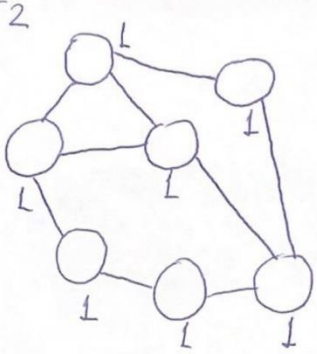
Οι Variational Autoencoders είναι εύκολοι στην εκπαίδευση και προσφέρουν καλή κανονικοποίηση του λανθάνοντα χώρου. Ωστόσο, μπορεί να παράγουν δείγματα χαμηλότερης ποιότητας. Τα Generative Adversarial Networks έχουν την ικανότητα να δημιουργούν ρεαλιστικά δείγματα, αλλά η εκπαίδευσή τους είναι δύσκολη, απαιτεί μεγάλη υπολογιστική ισχύ και προσεκτική ρύθμιση των υπερπαραμέτρων. Τα Diffusion Models παράγουν δείγματα υψηλής ποιότητας και έχουν καλή θεωρητική βάση, αλλά η διαδικασία δειγματοληψίας τους είναι αργή και η εφαρμογή τους σε πραγματικά προβλήματα είναι δύσκολη λόγω του υψηλού υπολογιστικού κόστους.

Άσκηση 6

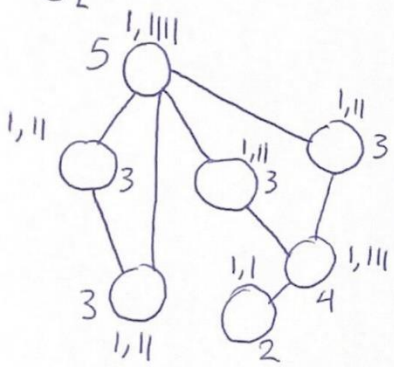
G_1



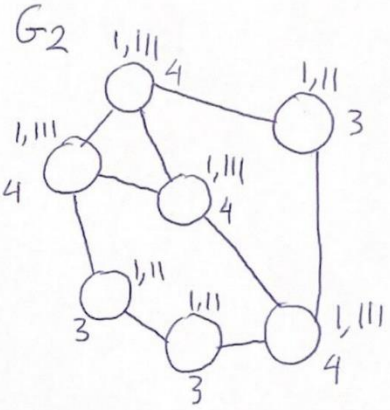
G_2



G_1

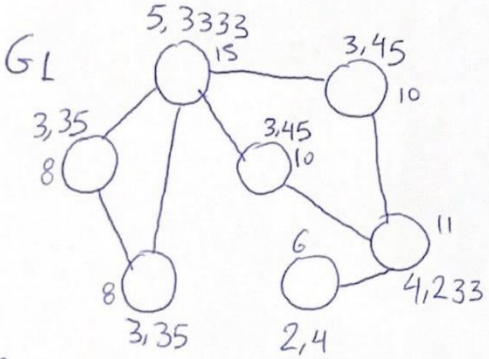


G_2

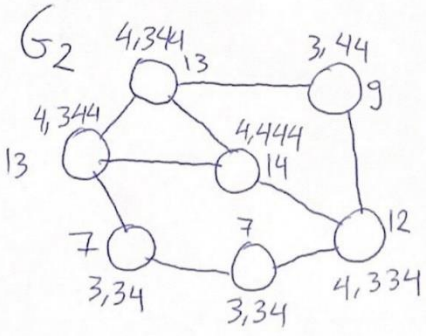


1,1 → 2 1,11 → 3 1,111 → 4 1,1111 → 5

G_1



G_2



2,4 → 6

3,34 → 7

3,35 → 8

3,44 → 9

3,45 → 10

4,233 → 11

4,334 → 12

4,344 → 13

4,444 → 14

5,3333 → 15

$\phi(G_1) = [7, 1, 4, 1, 1, 1, 0, 2, 0, 2, 1, 0, 0, 0, 1]$

$\phi(G_2) = [7, 0, 3, 4, 0, 0, 2, 0, 1, 0, 0, 1, 2, 1, 0]$

$K(G_1, G_2) = \phi(G_1)^T \phi(G_2) = 49 + 0 + 12 + 4 + 0 + 0 +$
$0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 65$