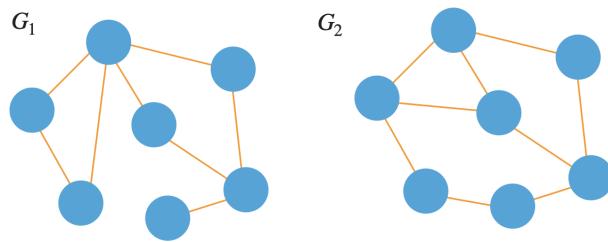
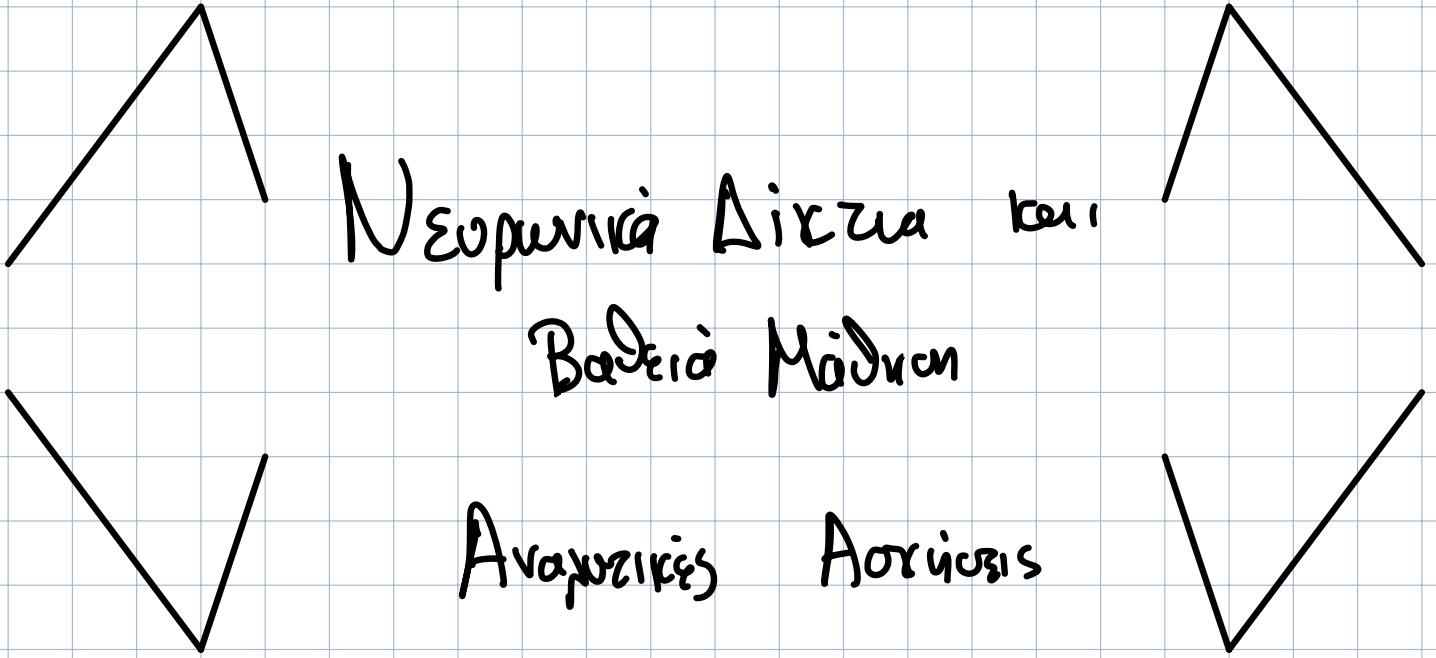


Άσκηση 6 (Graph Neural Networks)

Δίνονται οι γράφοι G_1 και G_2 του παρακάτω σχήματος. Να υπολογιστεί η ομοιότητά τους με βάση τον πυρήνα Weisfeiler-Lehman (WL), δίνοντας όλα τα βήματα του αλγόριθμου.





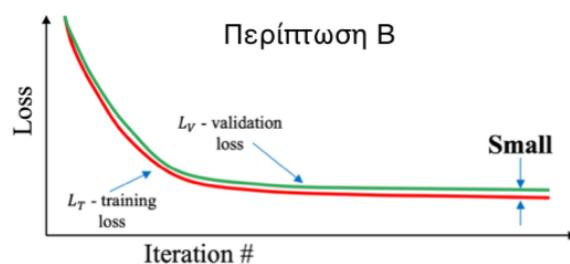
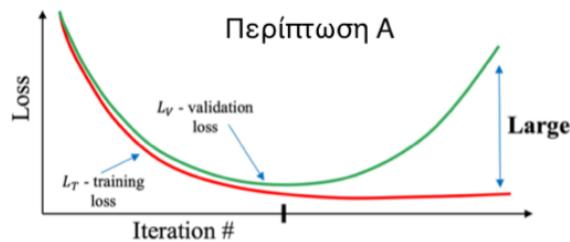
Χριστόδοχος Σπυρίδων

ΑΜ: 03120614

email: Chrisstles318@gmail.com

Άσκηση 1 (Multi Layer Perceptron - Regularization)

Για να εκπαιδεύσουμε ένα βαθύ νευρωνικό δίκτυο, χωρίζουμε τα δεδομένα που έχουμε στη διάθεσή μας σε τρία σύνολα: το σύνολο εκπαίδευσης (training set), το σύνολο επικύρωσης (validation set) και το σύνολο ελέγχου (testing set). Σε κάθε επανάληψη (εποχή) καταγράφουμε την τιμή της συνάρτησης κόστους (loss function) και έτσι λαμβάνουμε, για δύο διαφορετικές περιπτώσεις A και B, τα αντίστοιχα διαγράμματα που φαίνονται παρακάτω.



- Τι συμπέρασμα βγάζετε από το διάγραμμα για την αρχιτεκτονική του μοντέλου σας για καθεμία από τις περιπτώσεις A και B;
- Ποια τιμή επαναλήψεων (εποχών) n είναι πιο πιθανό να οδηγεί στο καλύτερο δυνατό μοντέλο μάθησης για καθεμία από τις περιπτώσεις A και B;
- Για την περίπτωση A, προτείνετε δύο τεχνικές που θα μπορούσαν να βελτιώσουν την επίδοση του μοντέλου.
- Εξηγήστε για ποιο λόγο είναι απαραίτητο το σύνολο ελέγχου (testing set) πέρα από τα σύνολα εκπαίδευσης και επικύρωσης.

Αρχήν 1 :

- Στην περίπτωση A παρατηρούμε ότι ήταν από συναφές υπόχρεο, μεγάλο validation loss, θεωρούμε ότι το μοντέλο ήταν overfitting. Στην περίπτωση B η αποκλίνη μεταξύ του training loss και του validation loss είναι πολύ μερική διαδοχή το μοντέλο καταχύπει σε γενικευσης επιρροχήν του δεδομένου του, αποφεύγοντας το overfitting.

2. Στην περιπτώσει A, η επίλογη των επόχων πέραν τα γινεται
τίποι την αποχήν του training loss και validation loss
καπου είναι που είναι η γραφή Συχάδη.

Στην περιπτώσει B, βρεπούμε ότι οι δύο γραφής L_r, L_t τείνουν
να ταυτίζονται, αφού όσο μεγαλύτερο αριθμό επόχων, τόσο το καλύτερο
αρκει να γεμοψεύτει ωστόμι το computation

3. Πλούτης οι λύσεις για να λύσετε το overfitting όπις

Data augmentation:

Η επαύξηση δεδομένων στα νευρωνικά δίκτυα περιλαμβάνει την τεχνητή αύξηση της ποικιλομορφίας των δεδομένων εκπαίδευσης με τυχαίες τροποποιήσεις στα υπάρχοντα δείγματα. Αυτή η τεχνική βελτιώνει την ευρωστία του μοντέλου και βοηθά στην αποφυγή της υπερπροσαρμογής προσομοιώνοντας ένα ευρύτερο φάσμα σεναρίων του πραγματικού κόσμου.

Mixup:

Η ανάμειξη είναι μια τεχνική αύξησης δεδομένων που δημιουργεί νέα παραδείγματα εκπαίδευσης αναμειγνύοντας εικόνες και τις ετικέτες τους από το σύνολο εκπαίδευσης. Αυτή η προσέγγιση ενθαρρύνει το μοντέλο να μάθει πιο γενικευμένα χαρακτηριστικά με γραμμική παρεμβολή μεταξύ πολλαπλών σημείων δεδομένων.

Dropout:

To Dropout είναι μια μέθοδος κανονικοποίησης που χρησιμοποιείται στα νευρωνικά δίκτυα, όπου τυχαία επιλεγμένοι νευρώνες αγνοούνται κατά τη διάρκεια της εκπαίδευσης, ουσιαστικά "πέφτοντας" έξω. Αυτό βοηθά στην αποφυγή υπερβολικής εξάρτησης του μοντέλου από κάθε νευρώνα, προωθώντας μια πιο ισχυρή αναπαράσταση χαρακτηριστικών.

4. Το συνόριο Σλέγχου (sliding) είναι απαραίτητο γιατί σεν
αυτό, αντί ότι νας δειχνεί πώς θα λειτουργεί το φυσικό μας
συν πληρωματικόντα. Το συνόριο εκπαιδεύει χρησιμοποιούσαι
τύπωνα για να φέρει, σε νεαρώντα άνθρωπων σεν το
συνόριο Σλέγχου χρησιμοποιούσαι για να αξιοχορίσει
να Σιορδάνη ταχού ασφαλείς του Σιερίζου, πχ βεργίσιαν
υπερπλανητικές ή/και Σιορδάνην βασίν

Άσκηση 2 (Representation Learning - Autoencoders)

Έστω ότι έχουμε πρόσβαση σε αυξημένες υπολογιστικές υποδομές και εκπαιδεύουμε ένα skipgram μοντέλο για ένα μεγαλύτερο λεξιλόγιο \mathcal{V}' . Το \mathcal{V}' περιέχει αναπαραστάσεις 1500 λέξεων (μαζί με τα special tokens) και η διάσταση των διανυσμάτων u_o και u_c είναι (256×1) . Στη συνέχεια χρησιμοποιούμε τα skipgram vectors και εκπαιδεύουμε έναν αυτοκωδικοποιητή (auto-encoder) με 5 κρυμμένο στρώμα διαστάσεων $[500, 250, 50, 250, 500]$ αντιστοίχως.

Απαντήστε στα παρακάτω ερωτήματα:

- Ποια είναι η διάσταση των χαρακτηριστικών εισόδου x_i στον auto-encoder?
- Ποια είναι η διάσταση των χαρακτηριστικών εξόδου y_i του auto-encoder?
- Ποιά είναι η διάσταση της λανθάνουσας αναπαράστασης (latent representation) του auto-encoder?

Άσκηση 2 :

Τούχισι ότι το λεξιλόγιο μας \mathcal{V} περιέχει 1500 λέξεις με διάνυσμα u_o, u_c ίση.

Διάσταση 256×1 την ο αυτοencoder έχει 5 κρυφά στρώματα με διάσταση $[500, 250, 50, 250, 500]$

(a) Η διάσταση των χαρακτηριστικών εισόδου x_i στον auto-encoder είναι ίδια με την διάσταση των skipgram vectors. Δηλαδή, αφού τα u_i και u_c έχουν διάσταση 256×1 ταυτόχρονα έχουν 1500 λέξεις άρα, η διάσταση x_i είναι $1500 \times 256 \times 1 \times 2$

επειδή εχουμε (u_i, u_c)

$$\text{Άρα τελικά } x_i : 768000$$

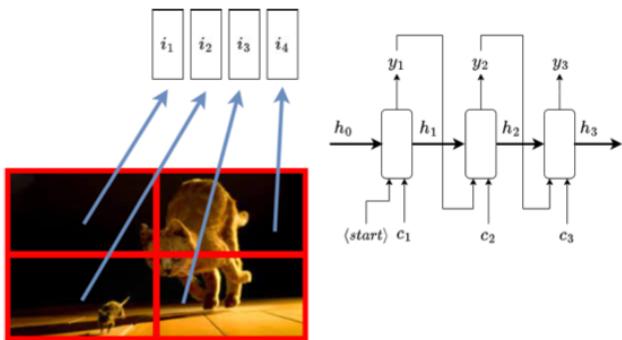
(b) Η διάσταση χαρακτηριστικών εξόδου y_i : πρέπει να έχει ίδια διάσταση με αυτή των εισόδων. Αφού ο auto-encoder αναπαράγει τα αρχικά διανύσματα μετα την συμπλήρωση του το decoding. Επομένως, η διάσταση χαρακτηριστικών εξόδου y_i είναι 768000.

(c) Η λανθάνουσα αναπαράσταση (latent representation) αναρριχείται με μικρότερο κρυφό στρώμα του auto-encoder. Δηλαδή, 50.

[latent representation: συμπληρωμένη μορφή των διεύθυνσηών]

Άσκηση 3 (Recurrent Neural Networks)

Εξετάστε το πρόβλημα που φαίνεται στο σχήμα, όπου δίνεται μια εικόνα και το αντικείμενο είναι να δημιουργηθεί μια περιγραφική λεξάντα φυσικής γλώσσας για την εικόνα.



Την εικόνα επεξεργάζεται ένα CNN (δεν αναπαρίσταται και δε χρειάζεται για την επίλυση), με αποτέλεσμα 4 αναπαραστάσεις χαρακτηριστικών i_1, i_2, i_3, i_4 , όπου κάθε $i_j \in \mathbb{R}^2$ όπως φαίνεται στο σχήμα. Στη συνέχεια, η λεξάντα δημιουργείται αυτόματα με παλινδρόμηση από έναν αποκωδικοποιητή που βασίζεται σε RNN, που εξαρτάται από τις αναπαραστάσεις των χαρακτηριστικών της εικόνας. Το λεξιλόγιο εξόδου περιέχει μόνο 6 λέξεις, συμπεριλαμβανομένων των συμβόλων $\langle \text{start} \rangle$ και $\langle \text{stop} \rangle$ με το ακόλουθο indexing: $V = [\langle \text{start} \rangle, \langle \text{stop} \rangle, \text{mouse}, \text{cat}, \text{staring}, \text{hunting}]$ και με embedding vectors:

$$\begin{aligned} y_{\langle \text{start} \rangle} &= [0, 0, 0]^T, & y_{\langle \text{stop} \rangle} &= [1, 1, 1]^T, & y_{\text{mouse}} &= [-1, 2, 0]^T, \\ y_{\text{cat}} &= [1, -2, 0]^T, & y_{\text{staring}} &= [0, -1, -1]^T, & y_{\text{hunting}} &= [0, 2, 1]^T. \end{aligned}$$

(a) Έστω εικόνα εισόδου αυτή που φαίνεται στο σχήμα, με τους ακόλουθους χάρτες χαρακτηριστικών εικόνας:

$$i_1 = [4, 0]^T, \quad i_2 = [0, 4]^T, \quad i_3 = [0, 0]^T, \quad i_4 = [0, 0]^T.$$

Σε κάθε χρονικό βήμα t , ο decoder που βασίζεται σε RNN λαμβάνει ως είσοδο $x_t \in \mathbb{R}^5$, η οποία είναι ένα concatenation της προηγούμενης εξόδου που ενσωματώνει $y_{t-1} \in \mathbb{R}^3$ και μια αναπαράσταση εικόνας $c_t \in \mathbb{R}^2$ (με αυτή τη σειρά) και χρησιμοποιεί αυτήν την είσοδο και την προηγούμενη κρυφή κατάσταση h_{t-1} για να υπολογίσει τη νέα κατάσταση h_t . Αυτό ακολουθείται από ένα γραμμικό επίπεδο εξόδου (linear output layer) με πίνακα

hidden to output:

$$W_{yh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \quad \begin{array}{l} \# \langle \text{start} \rangle \\ \# \langle \text{stop} \rangle \\ \# \text{mouse} \\ \# \text{cat} \\ \# \text{staring} \\ \# \text{hunting} \end{array}$$

όπου κάθε σειρά αυτού του πίνακα αντιστοιχεί στις λέξεις που αναφέρονται παραπάνω. Ο πίνακας input to hidden και ο πίνακας recurrence δίνονται αντίστοιχα από

$$\checkmark \quad W_{hx} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad W_{hh} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad \text{Πίνακας recurrence}$$

Ας υποθέσουμε ότι το RNN χρησιμοποιεί ενεργοποιήσεις ReLU, ότι όλα τα bias vectors είναι μηδενικά διανύσματα και ότι η αρχική κρυφή κατάσταση (hidden state) h_0 είναι ένα μηδενικό διάνυσμα.

Σε αυτήν την ερώτηση, υποθέστε ότι το $c_t := c$ είναι σταθερό για όλα τα χρονικά βήματα και προκύπτει από τη μέση συγκέντρωση των αναπαραστάσεων των χαρακτηριστικών εικόνας, $c = \frac{1}{4} \sum_{j=1}^4 i_j$, χωρίς μηχανισμό προσοχής. Υπολογίστε τις τρεις πρώτες λέξεις της λεξάντας χρησιμοποιώντας greedy decoding (επιλογή της πιο πιθανής λέξης σε κάθε βήμα).

(β) Ας υποθέσουμε τώρα ότι, αντί να χρησιμοποιεί ένα σταθερό c_t για όλα τα χρονικά βήματα, ο αποκωδικοποιητής που βασίζεται σε RNN έχει έναν μηχανισμό προσοχής scaled dot-product που παρακολουθεί τις αναπαραστάσεις των χαρακτηριστικών της εικόνας. Για κάθε χρονικό βήμα, το διάνυσμα ερωτήματος είναι h_{t-1} και οι αναπαραστάσεις των χαρακτηριστικών εικόνας i_1, i_2, i_3, i_4 χρησιμοποιούνται τόσο ως κλειδιά (key) όσο και ως τιμές (value).

Υποθέστε ξανά ότι το h_0 είναι ένα μηδενικό διάνυσμα. Για το πρώτο χρονικό βήμα ($t = 1$), υπολογίστε τις πιθανότητες προσοχής (attention probabilities) και το διάνυσμα εικόνας που προκύπτει c_1 . Η πρώτη λέξη θα είναι ίδια ή διαφορετική από αυτήν στην προηγούμενη ερώτηση;

Aσκηση 3:

$$(a) \text{ Τίπιντα υπολογίζω το } C, \quad C = \frac{1}{4} \sum_{j=1}^4 c_j = \frac{1}{4} \left[[4, 0]^T + [0, 4]^T + [0, 0]^T + [0, 0]^T \right] \\ = \frac{1}{4} \begin{bmatrix} 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Είσοδος δίνεται από την εκφρώνων ότι είναι $[0, 0, 0]^T$ και το C , Από:

$$X_t = [0, 0, 0, 1, 1]^T$$

$t=1$:

bias
input

$$h_1 = \text{ReLU}\left(V^* X_t + U^* h_{t-1} + C\right) = \text{ReLU}\left(\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot 0 + 0\right) \\ = \text{ReLU}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$y_1 = (W_{yh} \cdot h_1 + b) = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ 3 \\ 3 \\ 4 \\ 2 \\ 2.9 \end{bmatrix} \Rightarrow \max \text{ σημ} \text{ 4^η item from}$$

$$W_{yh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \quad \begin{array}{l} \# \langle \text{start} \rangle \\ \# \langle \text{stop} \rangle \\ \# \text{mouse} \\ \# \text{cat} \\ \# \text{staring} \\ \# \text{hunting} \end{array}$$

Τίπων δίγνον Cat

$t=2$:

Output του το cat, από input του επιθέτου σινι το output του cat της δίνεται

$$X_2 = [1, -2, 0, 1, 1]^T$$

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

$$y_2 = (W_{yh} h_2 + b) = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -15 \\ 0 \\ 3 \\ 6 \\ 9 \\ 8.7 \end{bmatrix}$$

$$W_{yh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix}$$

(start)
(stop)
mouse
cat
staring
hunting

max from 5

Diagram sign **Staring**

$t=3$:

Output from staring after input from step 2 will be output from staring. Since

$$x_3 = [0, -1, -1, 1, 1]^T$$

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

$$y_3 = (W_{yh} h_3 + b) = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} -10 \\ 6 \\ 4 \\ 4 \\ -2 \\ 0 \end{bmatrix}$$

$$W_{yh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix}$$

(start)
(stop)
mouse
cat
staring
hunting

max from 2

Diagram sign for
the previous sign **<Stop>**

$$(b) z = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) \cdot V$$

$$\text{Score} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$$

$d_k = 2$ (outputs)

$$\text{Scale} = \frac{s}{\sqrt{d_k}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, P = \text{softmax}(s)$$

$$\text{softmax} = \frac{e^{z_i}}{\sum_{j=1}^4 e^{z_j}} \Rightarrow$$

$$\text{softmax} = \begin{bmatrix} 0,25 \\ 0,25 \\ 0,25 \\ 0,25 \end{bmatrix}^T$$

$$z = [0,25 \quad 0,25 \quad 0,25 \quad 0,25] \quad \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = [1 \quad 1]$$

Apa artus $C = [\quad]$

Αρχαίο σήμερα με πρώτη
τοποθετημένη την πρώτη σειρά
στην δεύτερη σειρά

Άσκηση 4 (Convolutional Neural Networks)

Ας υποθέσουμε οτι έχουμε ένα δίκτυο σαν το AlexNet (Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NeurIPS 2012). Θεωρήστε πως έχουμε εικόκες διαστάσεων $227 \times 227 \times 3$ (έγχρωμες με RGB channels) και φίλτρο $11 \times 11 \times 3$ στο πρώτο convolution layer. Το δίκτυο έχει συνολικά 96 φίλτρα, stride ίσο με 4 και μηδενικό padding.

(α) Υπολογίστε τις διαστάσεις στην έξοδο του πρώτου convolutional layer.

(β) Υπολογίστε τον αριθμό των units στο πρώτο convolutional layer.

(γ) Υπολογίστε τον αριθμό των εκπαιδεύσιμων παραμέτρων του πρώτου convolutional layer με διαμοιρασμό βαρών.

(δ) Αν αντικαθιστούσαμε το CNN με ένα FeedForward layer με 256 units πόσες εκπαιδεύσιμες παραμέτρους θα είχαμε;

Aσκηση 4:

$$W_1 \times H_1 \times D_1$$

$$\text{Εικόνες} \quad 227 \times 227 \times 3$$

$$\text{φίλτρο} \quad F = 11 \times 11 \times 3 \quad 1^{\text{st}} \text{ layer}$$

$$K = 96 \text{ filters}$$

$$S \text{ stride} = 4$$

$$P \text{ padding} = 0$$

$$(a) \text{ Μέτρα του πρώτου convolutional layer} \quad W_2 = \frac{(W_1 - F + 2P)}{S} + 1 = 55$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1 = 55$$

$$D_2 = K = 96$$

Σ Συστόματα
μετατόπιση
1st cl:

$$55 \times 55 \times 96$$

$$(b) \text{ Units} \Rightarrow 55 \times 55 \times 96 = 290400 \text{ units}$$

(γ) # εκπαιδεύσιμων παραμέτρων του 1st conv. layer με διαμοιρασμό βαρών :

$$K \times F = 96 \times (11 \times 11 \times 3 + 1) = 34846 + 96 = 34.944$$

↑
bias

(δ) CNN \rightarrow FeedForward layer με 256 units, για τις εκπαιδεύσιμες παραμέτρους αριθμούς τα βραχιά το πρώτο του αριθμού εισόδων ($227 \times 227 \times 3 + 1$) με την αριθμό των units 256

↑
bias

$$\text{Άρα: } (227 \times 227 \times 3 + 1) \times 256 = 39.574.528$$

Άσκηση 5 (Generative models)

Έχοντας μελετήσει τα βασικά χαρακτηριστικά των ακόλουθων τριών τύπων παραγωγικών μοντέλων:

- (α) Variational Autoencoders
- (β) Generative Adversarial Networks
- (γ) Diffusion models

να προβείτε σε μια αναλυτική σύγκριση των τριών τύπων μοντέλων. Μπορείτε να εξετάσετε θέματα όπως οι θεμελιώδεις αρχές κάθε μοντέλου, η διαδικασία εκπαίδευσης, η αποτελεσματικότητα, οι απαιτήσεις σε μνήμη / χρόνο / δεδομένα εκπαίδευσης, κ.α., και να αναφερθείτε σε ομοιότητες και διαφορές, πλεονεκτήματα και μειονεκτήματα του κάθε τύπου μοντέλου.

Άσκηση 5

Βασικά στοιχεία κάθε μοντέλου

1. Variational Autoencoders (VAEs)

- Οι VAEs είναι πιθανοτικά παραγωγικά μοντέλα που χρησιμοποιούν το πλαίσιο της Μπεϋζιανής συμπερασματολογίας (Bayesian inference). Βασίζονται σε μια δομή παρόμοια με τους παραδοσιακούς autoencoders, οι οποίοι έχουν μια αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή. Ο κωδικοποιητής αντιστοιχίζει τα δεδομένα εισόδου σε μια αναπαράσταση latent space, και ο αποκωδικοποιητής ανακατασκευάζει τα δεδομένα εισόδου από αυτόν τον λανθάνοντα χώρο (latent space).
- Η βασική ιδέα πίσω από τους VAE είναι η εκμάθηση των παραμέτρων της κατανομής (μέσος όρος και διακύμανση) των λανθάνουσων μεταβλητών, επιτρέποντας τη δημιουργία νέων σημείων δεδομένων με δειγματοληψία από αυτή την κατανομή.

2. Generative Adversarial Networks (GANs)

- Τα GANs αποτελούνται από δύο ανταγωνιστικά μοντέλα νευρωνικών δικτύων: έναν γεννήτορα και έναν διαχωριστή. Η γεννήτρια δημιουργεί δείγματα που προορίζονται να προέρχονται από την ίδια κατανομή με τα δεδομένα εκπαίδευσης, ενώ ο διαχωριστής τα αξιολογεί έναντι των πραγματικών δεδομένων, μαθαίνοντας να διακρίνει τα πραγματικά από τα ψεύτικα.
- Η εκπαίδευση των GANs περιλαμβάνει ένα παιχνίδι min-max όπου η γεννήτρια προσπαθεί να ξεγελάσει τον διαχωριστή και ο διαχωριστής προσπαθεί να ταξινομήσει σωστά τα πραγματικά και τα ψεύτικα δείγματα.

3. Diffusion Models

- Τα μοντέλα διάχυσης μαθαίνουν να παράγουν δεδομένα αντιστρέφοντας μια διαδικασία διάχυσης. Αυτή η διαδικασία μετατρέπει σταδιακά τα δεδομένα από το υψηλό θόρυβο σε μια σειρά βημάτων.
- Για τη δημιουργία νέων δειγμάτων, το μοντέλο μαθαίνει να αντιστρέψει αυτή τη διαδικασία προσθήκης θορύβου, ζεκινώντας από το θόρυβο και σταδιακά αποθρούβωντας τον πίσω σε ένα δείγμα από την κατανομή δεδομένων.

Training Process

VAEs

- Η εκπαίδευση περιλαμβάνει την ελαχιστοποίηση της απώλειας ανακατασκευής (πόσο καλά ο αποκωδικοποιητής μπορεί να αναδημιουργήσει την είσοδο από τις λανθάνουσες μεταβλητές) σε συνδυασμό με την απόκλιση KL, η οποία μετράει πόσο αποκλίνει η μαθημένη κατανομή λανθάνουσας μεταβλητής από την προηγούμενη κατανομή.

GANs

- Η εκπαίδευση ενός GAN είναι συνήθως πιο δύσκολη λόγω της ανάγκης διατήρησης της ισορροπίας μεταξύ της γεννήτριας και του διαχωριστή (generator and discriminator). Εάν το ένα υπερτερεί σημαντικά του άλλου, μπορεί να οδηγήσει σε αποτυχίες εκπαίδευσης, όπως η κατάρρευση λειτουργίας.

Diffusion Models

- Η εκπαίδευση μοντέλων διάχυσης περιλαμβάνει την εκμάθηση της αποθορυβοποίησης δεδομένων σε πολλά βήματα, τα οποία μπορεί να είναι υπολογιστικά εντατικά και να απαιτούν πολλή μνήμη λόγω της ανάγκης επεξεργασίας πολλαπλών βημάτων για κάθε δείγμα εκπαίδευσης.

Efficiency, Memory/Time/Data Requirements

VAEs

- Γενικά αποδοτικά και σταθερά στην εκπαίδευση με μέτριες απαιτήσεις μνήμης και χρόνου. Ωστόσο, ενδέχεται να δυσκολεύονται να συλλάβουν υψηλής ποιότητας, λεπτομερείς μεταβολές στα δεδομένα σε σύγκριση με τα GAN ή τα μοντέλα διάχυσης.

GANs

- Μπορούν να παράγουν πολύ υψηλής ποιότητας και ευκρινείς εικόνες, καθιστώντας τα δημοφιλή για εργασίες που απαιτούν υψηλή οπτική πιστότητα. Ωστόσο, είναι γνωστό ότι είναι δύσκολο να εκπαιδευτούν και απαιτούν σημαντικούς υπολογιστικούς πόρους.

Diffusion Models

- Συχνά παράγουν αποτελέσματα υψηλής ποιότητας, ξεπερνώντας ενδεχομένως τα GAN σε πιστότητα και ποικιλομορφία. Ωστόσο, απαιτούν σημαντικούς υπολογιστικούς πόρους και χρόνο εκπαίδευσης λόγω της επαναληπτικής φύσης του μοντέλου.

Similarities and Differences

Similarities

- Και τα τρία μοντέλα είναι παραγωγικά και στοχεύουν στην εκμάθηση της υποκείμενης κατανομής των δεδομένων.
- Όλα μπορούν να δημιουργήσουν νέα δείγματα δεδομένων μετά την εκπαίδευση.

Differences

- Τα VAE και τα μοντέλα διάχυσης έχουν μια πιο ρητή αντιμετώπιση της κατανομής των δεδομένων μέσω πιθανοτικών πλαισίων, ενώ τα GAN επικεντρώνονται σε μια προσέγγιση άμεσης αντιθετικής εκπαίδευσης χωρίς ρητή μοντελοποίηση της κατανομής.
- Τα VAEs συνήθως παράγουν πιο ομαλές αλλά λιγότερο ευκρινείς εικόνες, ενώ τα GANs εστιάζουν στην ευκρίνεια και τη λεπτομέρεια. Τα μοντέλα διάχυσης παρέχουν μια καλή ισορροπία μεταξύ ποικιλομορφίας και πιστότητας.

Advantages and Disadvantages

VAEs

- **Advantages:** Ρητή μοντελοποίηση της κατανομής των δεδομένων.
- **Disadvantages:** Συχνά οδηγεί σε πιο θολές ή λιγότερο λεπτομερείς εξόδους.

GANs

- **Advantages:** Ιδανικά για εφαρμογές που απαιτούν υψηλή οπτική λεπτομέρεια.
- **Disadvantages:** Επιρρεπείς σε κατάρρευση λειτουργίας.

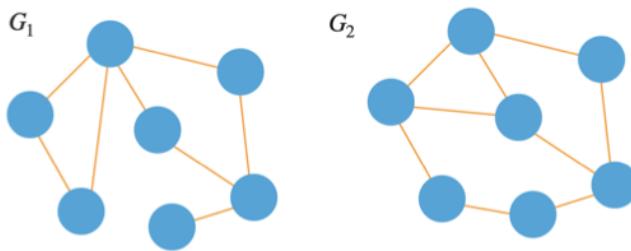
Diffusion Models

- **Advantages:** Υψηλή πιστότητα και ποικιλομορφία των αποτελεσμάτων- Ισχυρή απόδοση σε μια σειρά εργασιών.
- **Disadvantages:** Αργή διαδικασία δημιουργίας σε σύγκριση με άλλα μοντέλα.

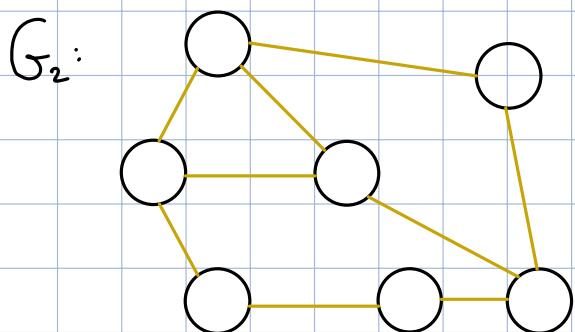
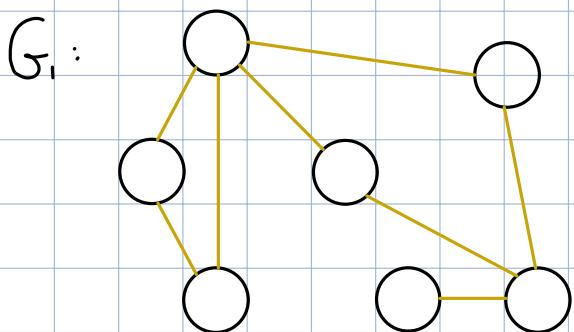
Συνολικά, η επιλογή μεταξύ αυτών των μοντέλων εξαρτάται συχνά από τις συγκεκριμένες απαιτήσεις της εφαρμογής, συμπεριλαμβανομένης της επιθυμητής ποιότητας των αποτελεσμάτων, των διαθέσιμων υπολογιστικών πόρων και των ειδικών χαρακτηριστικών των δεδομένων που μοντελοποιούνται.

Άσκηση 6 (Graph Neural Networks)

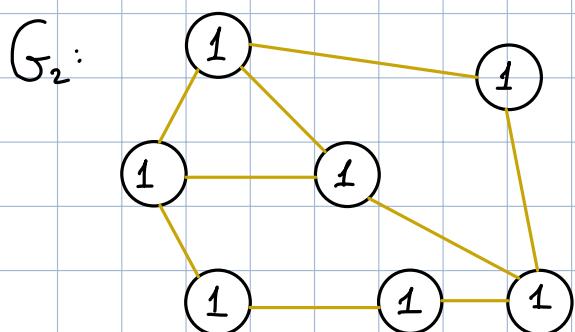
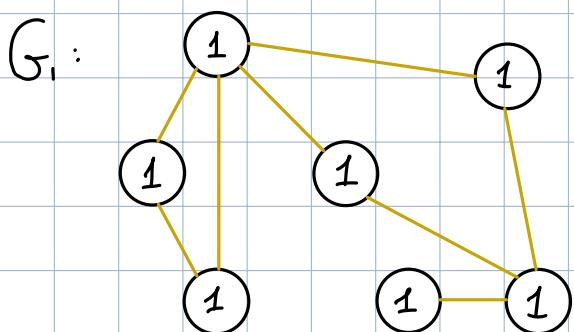
Δίνονται οι γράφοι G_1 και G_2 του παρακάτω σχήματος. Να υπολογιστεί η ομοιότητά τους με βάση τον πυρήνα Weisfeiler-Lehman (WL), δίνοντας όλα τα βήματα του αλγόριθμου.

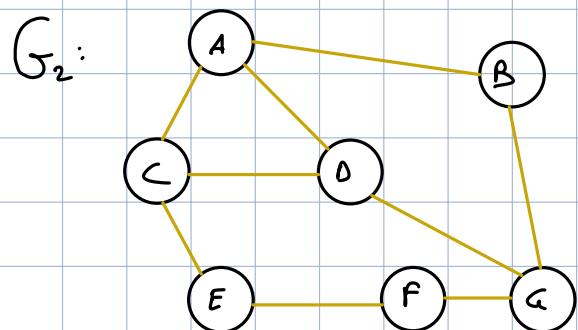
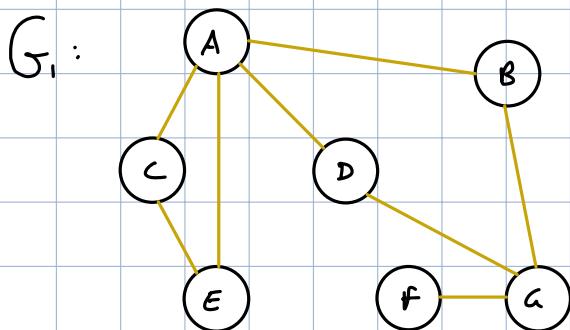


Άσκηση 6:



Βήμα 0: Θέτω στην σήμη "1" σε κάθε κόμβο





Node	Neighbors	$i=0$	$i=1$	Hash 1	$i=2$	hash2
A	B C D E	1	1; 1, 1, 1, 1	5	5; 3, 3, 3, 3	15
B	A G	1	1; 1, 1	3	3; 4, 5	10
C	A E	1	1; 1, 1	3	3; 3, 5	8
D	A G	1	1; 1, 1	3	3; 4, 5	10
E	A C	1	1; 1, 1	3	3; 3, 5	8
F	G	1	1; 1	2	2; 4	6
G	B D F	1	1; 1, 1, 1	4	4; 2, 3, 3	11

Παραγόντων

όπι σεν

υπάρχουν
ιδιοι αριθμοί

Node	Neighbors	$i=0$	$i=1$	Hash 1	$i=2$	hash2
A	B C D	1	1; 1, 1, 1	4	4; 3, 4, 4	12
B	A G	1	1; 1, 1	3	3; 4, 4	9
C	A D E	1	1; 1, 1, 1	4	4; 3, 4, 4	12
D	A C G	1	1; 1, 1, 1	4	4; 4, 4, 4	14
E	C F	1	1; 1, 1	3	3; 3, 4	7
F	E G	1	1; 1, 1	3	3; 3, 4	7
G	B D F	1	1; 1, 1, 1	4	4; 3, 3, 4	13

- 1; 1 → 2 • 2; 4 → 6 • 3; 4, 5 → 10 • 4; 4, 4, 4 → 14
- 1; 1, 1 → 3 • 3; 3, 4 → 7 • 4; 2, 3, 3 → 11 • 5; 3, 3, 3, 3 → 15
- 1; 1, 1, 1 → 4 • 3; 3, 5 → 8 • 4; 3, 4, 4 → 12 • 4; 3, 3, 4 → 13
- 1; 1, 1, 1, 1 → 5 • 3; 4, 4 → 9 • 4; 3, 3, 4 → 13

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	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	2	1	1	0
Product	49	0	12	4	0	0	0	0	0	0	0	0	0	0	0

Μαργαριτάρια στο product είναι μόνο
από τα μεγαλύτερα αριθμούς των φυλαρίδων

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