

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΒΑΘΙΑ ΜΑΘΗΣΗ

8^ο εξάμηνο

Ακαδημαϊκό έτος 2023-2024

Σειρά Αναλυτικών Ασκήσεων

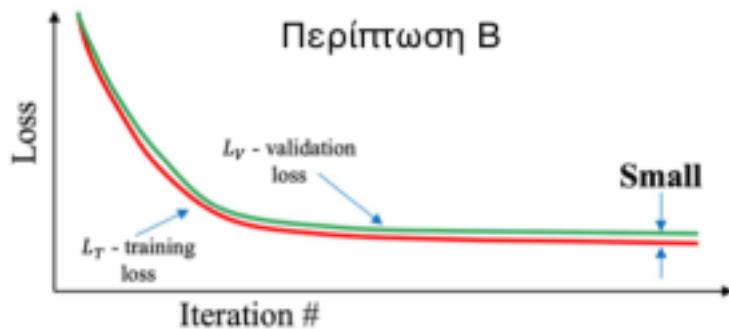
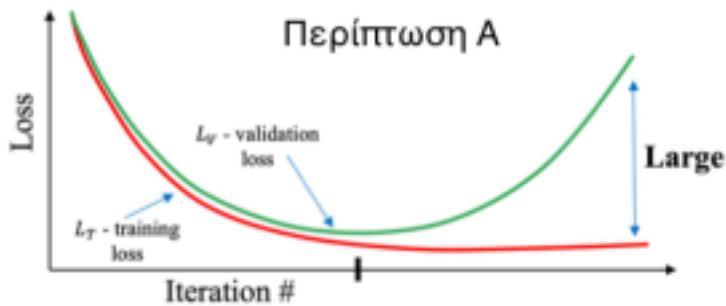
Ημερ. Παράδ.: 21.06.2024

Ονοματεπώνυμο: Κούτρας Ιωάννης- Παναγιώτης

AM: 03120130

Email: ioanniskoutras2002@gmail.com

Ασκηση 1 (Multi Layer Perceptron - Regularization)



1) Για την αρχιτεκτονική των μοντέλων στις περιπτώσεις Α και Β μπορούμε να συνάγουμε τα εξής:

Περίπτωση Α: Το μοντέλο στην περίπτωση Α είναι αρκετά περίπλοκο, το οποίο σημαίνει συνήθως ότι έχει μεγαλύτερο αριθμό παραμέτρων- πιο βαθιά αρχιτεκτονική. Το δίκτυο έχει την ικανότητα να κάνει πολύ καλό fit στα training data , εξ'ού και το χαμηλό training loss.

Από την άλλη η διαφορά validation loss- training loss ανδένεται το οποίο σημαίνει ότι υπάρχει overfitting, γεγονός που αποδεικνύει επίσης την ύπαρξη πολλών παραμέτρων.

Περίπτωση Β: Το μοντέλο στην περίπτωση Β ούτε υπεραπλουστεύει τα δεδομένα, ούτε είναι πολύ περίπλοκο όπως στην περίπτωση Α. Πραγματοποιεί καλύτερη γενίκευση σε (καινούρια) δεδομένα, αφού παρατηρούμε ότι το κενό μεταξύ validation loss- training loss είναι σταθερά μικρό, άρα έχει σημαντικά πιο επιθυμητή απόδοση από το μοντέλο Α.

2) Περίπτωση Α: Ο βέλτιστος αριθμός επαναλήψεων n (epochs) είναι το σημείο, όπου το validation loss φτάνει στη minimum τιμή του και μετά το οποίο ξεκινάει να αυξάνεται. Σε εκείνο το σημείο ελαχιστοποιείται το overfitting.

Περίπτωση Β: Εδώ θα διαλέγαμε μια μεγάλη τιμή n, στην οποία η διαφορά validation loss- training loss είναι ήδη μικρή και έχει πρακτικά σταθεροποιηθεί. Σε εκείνο το σημείο έχουμε πολύ καλή γενίκευση.

3) Τεχνική #1- Early stopping:

Όπως αναλύσαμε παραπάνω, θα μπορούσαμε να προσαρμόσουμε τον αριθμό των επαναλήψεων (epochs), έτσι ώστε να ελαχιστοποιήσουμε το overfitting.

Τεχνική #2- Dropout:

Μία μορφή regularization που εφαρμόζουμε, απενεργοποιώντας «τυχαία» (δηλαδή με βάση μία παράμετρο που είναι η πιθανότητα να απενεργοποιηθεί) ορισμένους νευρώνες, ώστε να αναγκάσουμε το δίκτυο να γενικεύει καλύτερα. Με αυτόν τον τρόπο μειώνουμε την εξάρτηση από συγκεκριμένους νευρώνες και έτσι βελτιώνεται η απόδοση του μοντέλου σε νέα δεδομένα.

Σχόλιο: Γενικά η επιλογή της κατάλληλης τεχνικής είναι συνάρτηση του δικτύου και της αρχιτεκτονικής εκείνου. Μιας και δεν μας δίδονται περαιτέρω πληροφορίες για το μοντέλο Α, μπορούμε απλώς να αναφέρουμε μερικές πιθανές τεχνικές που βελτιώνουν την απόδοση του.

Άλλες υποψήφιες τεχνικές είναι το **Data Augmentation, L1/L2 Regularization, Feature Selection** κλπ.

4) To testing set είναι ένα ξεχωριστό σύνολο δεδομένων, το οποίο είναι νέο για το μοντέλο που εξετάζουμε (δεν έχει εκπαιδευτεί σε αυτό) και χρησιμοποιείται για την αξιολόγηση της επίδοσης του δικτύου μετά τη φάση εκπαίδευσης.

Ο λόγος που το χρειαζόμαστε είναι για να έχουμε μια «αμερόληπτη» αξιολόγηση του μοντέλου σε νέα δεδομένα, δηλαδή να έχουμε μια εικόνα για την πραγματική του ικανότητα να γενικεύει.

Άσκηση 2 (Representation Learning - Autoencoders)

Έχουμε εκπαιδεύσει ένα μοντέλο Skipgram για ένα μεγαλύτερο λεξιλόγιο V' , το οποίο περιέχει representations 1500 λέξεων (με special tokens). Οι διαστάσεις των διανυσμάτων u_o και u_c είναι (256 x 1).

Έπειτα χρησιμοποιούμε αυτά τα Skipgram vectors και εκπαιδεύουμε έναν auto-encoder με 5 κρυμμένα στρώματα διαστάσεων [500, 250, 50, 250, 500] αντιστοίχως.

α) Διάσταση των χαρακτηριστικών εισόδου x_i στον auto-encoder:

Η είσοδος του autoencoder είναι το Skipgram vector (δηλαδή vector representations) που έχει μάθει το μοντέλο Skipgram. Αφού τα u_o και u_c έχουν διάσταση (256 x 1), έχουμε ότι:

Διάσταση των χαρακτηριστικών εισόδου x_i στον auto-encoder = 256.

β) Διάσταση των χαρακτηριστικών εισόδου y_i στον auto-encoder:

Ο autoencoder κατασκευάζει εκ νέου το input του στο output layer. Συνεπώς, η διάσταση των χαρακτηριστικών εισόδου y_i πρέπει να είναι ίδια με εκείνη των χαρακτηριστικών εισόδου x_i .
Άρα,

Διάσταση των χαρακτηριστικών εξόδου y_i στον auto-encoder = 256.

γ) Διάσταση της λανθάνουσας αναπαράστασης (latent representation) του auto-encoder:

Το latent representation είναι το output του μεσαίου layer στον autoencoder. Μας έχουν δοθεί οι εξής διαστάσεις για τα layers:

[500,250,50,250,500]

Το τρίτο layer είναι το μεσαίο, και η διάστασή του είναι ίση με 50. Οπότε:

Διάσταση του latent representation = 50.

Άσκηση 3 (Recurrent Neural Networks)

Παρατίθεται η χειρόγραφη λύση:

Άσκηση 3 (Recurrent Neural Networks)

Λεξιλόγιο: $V = \{\text{start}, \text{stop}, \text{mouse}, \text{cat}, \text{staring}, \text{hunting}\}$
 embedding vectors:

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

Διάφορες ειδος οι χάρτες χαρακτηριστικών εικόνας:

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

Kai οι γιγάντες:

$$W_{yh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2,9 & 0 \end{bmatrix}, W_{hx} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}, W_{hh} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\mu \text{ initial state } h_0 = [0 \ 0]^T$$

$$\text{a) simi: } c = \frac{1}{4} \sum_{j=1}^4 i_j = \frac{1}{4} ([4 \ 0] + [0 \ 4] + [0 \ 0] + [0 \ 0]) = [1 \ 1]^T$$

$$x_1 = [y_{\text{start}} \ c]^T = [0 \ 0 \ 0 \ 1 \ 1]^T$$

$$\begin{aligned} h_1 &= \text{ReLU}(W_{hx}x_1 + W_{hh}h_0) = \text{ReLU} \left[\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \right] \\ &= \text{ReLU} \left[\begin{bmatrix} 0+0+0+1+0 \\ 0+0+0+0+1 \end{bmatrix} \right] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned}$$

$$y_1 = W_{yh}h_1 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2,9 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ 3 \\ 3 \\ 4 \\ \frac{1}{2} \\ 2,9 \end{bmatrix}$$

Έχουμε greedy decoding, επομένως
 επιλέγεται το index της max της
 που αντιστοιχεί στο «cat».

$$x_2 = [y_{cat} \ c]^T = [1 -2 0 1 1]^T$$

$$h_2 = \text{ReLU}(W_{hx}x_2 + W_{hb}h_1) = \text{ReLU} \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \text{ReLU} \begin{pmatrix} 3 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

$$y_2 = W_{yh}h_2 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2,9 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -15 \\ 0 \\ 3 \\ 6 \\ 9 \\ 8,7 \end{bmatrix} \rightarrow \text{enlidjevan zo index nou aansoekjel}\text{ en de} \tilde{\text{j}} \text{n aasting}.$$

$$x_3 = [y_{staring} \ c]^T = [0 -1 -1 1 1]^T$$

$$h_3 = \text{ReLU}(W_{hx}x_3 + W_{hb}h_2) = \text{ReLU} \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$= \text{ReLU} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$y_3 = W_{yh}h_3 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2,9 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -15 \\ 3 \\ 4 \\ 6 \\ 5 \\ 5,8 \end{bmatrix} \rightarrow \text{enlidjevan zo index nou aansoekjel}\text{ en de} \tilde{\text{j}} \text{n aasting}$$

Onder oor ypeis nulles dejsus tips dejsus eina:
as cat staring cat

$$b) \text{ Εξουπερ: } q_1 = h_0 = [0 \ 0]^T$$

$$k_j = v_j = i_j \quad , j \in \{1, 2, 3, 4\}$$

και υπολογίζουμε τα attention scores ως:

$$\text{score}(q_1, k_j) = q_1^T k_j \quad , j \in \{1, 2, 3, 4\}$$

$$\text{Όπως } q_1 = [0 \ 0]^T = O_2 \Rightarrow \text{score}(q_1, k_j) = 0 \quad \forall j \in \{1, 2, 3, 4\} \quad (1)$$

Για τις ιδιες τιμεις j , τα attention probabilities είναι:

$$a_j = \frac{\exp(\text{score}(q_1, k_j))}{\sum_{l=1}^4 \exp(\text{score}(q_1, k_l))} \stackrel{(1)}{=} \frac{\exp(0)}{4 \cdot \exp(0)} = \frac{1}{4}$$

$$c_1 = \sum_{j=1}^4 a_j v_j = \frac{1}{4} \sum_{j=1}^4 i_j \stackrel{(a)}{=} [1 \ 1] \quad \text{όπως υπολογίστηκε και ουτο (a)}$$

ερώτηση

Η διαδικαστική δεν διαφέρει τύπων από το (a) ερώτηση και παρέχει το ίδιο διάνομα y_1 , συντοός παραγόμενης και ίσης τη λέξη "cat".
Αρα η πρώτη λέξη είναι ίδια με αυτή της προηγούμενης ερώτησης.

Άσκηση 4 (Convolutional Neural Networks)

Παρατίθεται η χειρόγραφη επίλυση:

'Άσκηση 4 (Convolutional Neural Networks)

- AlexNet - like network

- Εικόνες διαστάσεων $227 \times 227 \times 3$

- Πρώτο Convolutional Layer: - Filter size: $11 \times 11 \times 3$
- #of filters: 96
- Stride: 4
- Padding: 0

a) Διαστάσεις στην έξοδο του πρώτου Convolutional Layer:

Γενικά έχουμε:

$$\text{Output Dimension} = \left\lfloor \frac{(\text{Input Dimension} - \text{Filter Size} + 2 \times \text{Padding})}{\text{Stride}} \right\rfloor + 1$$

Εφαρμόζουμε για Height, Width παραπομπές:

$$\text{Output Height} = \text{Output Width} = \left\lfloor \frac{(227 - 11 + 2 \times 0)}{4} \right\rfloor + 1 = \left\lfloor \frac{216}{4} \right\rfloor + 1 = 55$$

Για το depth του output έχουμε, output depth = #of filters = 96

Δηλαδή τελικά οι γηγενής διαστάσεις είναι $55 \times 55 \times 96$.

b) Αριθμός units στο πρώτο Convolutional Layer

$$\text{Έχουμε } 55 \times 55 \times 96 = 290.400 \text{ units}$$

γ) Αριθμός ανό trainable parameters στο νέυρο Convolutional Layer με shared weights:

- Trainable Parameters:

Κάθε φίλτρο έχει διαστάσεις $11 \times 11 \times 3$

Κάθε φίλτρο έχει $11 \times 11 \times 3 = 363$, weights και 1 bias term

\Rightarrow Κάθε φίλτρο έχει $363 + 1 = 364$ παραμέτρους

Με 96 φίλτρα, ο συνολικός αριθμός παραμέτρων είναι.

$$96 \times 364 = 34,944 \text{ παραμέτροι}$$

δ) Αριθμός ανό trainable parameters με FeedForward Layer με 256 units ανά για CNN.

Τύπος Εισού: $227 \times 227 \times 3 = 154,587$ input units

Κάθεται από τα 256 units στο feedforward layer συνέπεια με κάθεται ανό τα 154,587 input units.

Κάθε σύνδεση έχει ένα weight $\Rightarrow 154,587 \times 256 = 39,574,912$ weights

Κάθε unit έχει ένα bias term $\Rightarrow 256$ biases

Οντότε συνολική έκπτωση: $39,574,912 + 256 = 39,575,168$ παραμέτροι

Άσκηση 5 (Generative models)

Με βάση τα 3 παρακάτω Generative AI Models:

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

Θα πραγματοποιήσουμε μια αναλυτική σύγκριση στα θέματα που προτείνονται στην εκφώνηση.

Αυτά είναι:

- i) οι θεμελιώδεις αρχές κάθε μοντέλου,
- ii) η διαδικασία εκπαίδευσης,
- iii) η αποτελεσματικότητα,
- iv) οι απαιτήσεις σε μνήμη / χρόνο / δεδομένα εκπαίδευσης,
- v) οι ομοιότητες και οι διαφορές,
- vi) τα πλεονεκτήματα και τα μειονεκτήματα του κάθε τύπου μοντέλου

- i) οι θεμελιώδεις αρχές κάθε μοντέλου:

Αναφέρουμε συνοπτικά τα χαρακτηριστικά των 3 μοντέλων:

Variational Autoencoders (VAEs):

- Όπως και ένας απλός Autoencoder αποτελείται από έναν encoder και έναν decoder με τη διαφορά όμως ότι το input είναι encoded ως μια κατανομή στο latent space.



Eικόνα από σελ. 10 των σχετικών slides.

- Ο encoder κάνει map τα input data σε μια κατανομή πιθανότητας (probability distribution) στο latent space, και ο decoder κάνει map τα latent variables πίσω στον χώρο δεδομένων (data space).
- Σύμφωνα με τη σελ. 12 των σχετικών slides στη σελ. του μαθήματος έχουμε τις εξής αρχές, οι οποίες μας βοηθάνε να αποτρέψουμε το overfitting:
«Regularise covariance matrix and mean of encoder's distributions.
» Enforce distributions to be close to a standard normal distribution (centred and reduced).»

Generative Adversarial Networks (GANs):

- Τα GANs αποτελούνται από δύο νευρωνικά δίκτυα: έναν generator και έναν discriminator.
- Ο generator δημιουργεί false αλλά «plausible» data, ενώ ο discriminator προσπαθεί να διακρίνει τα true από τα false data.
- Ο generator επιδιώκει να ξεγελάσει τον discriminator και ο discriminator επιβάλλει ποινή στον generator σε περιπτώσεις που παράγει implausible δεδομένα.

Diffusion models:

- Βασίζονται σε αρχές της μη-ισορροπημένης θερμοδυναμικής.
- Σταδιακά προσθέτουν θόρυβο στα δεδομένα, με στόχο την εκμάθηση μιας reverse diffusion διαδικασίας, αφαιρώντας τον θόρυβο.
- Σκοπός είναι να μάθει το μοντέλο να παράγει ρεαλιστικά δείγματα από θόρυβο.

ii) η διαδικασία εκπαίδευσης:

Variational Autoencoders (VAEs):

- Εκπαιδεύουμε τον encoder να κάνει map τα εισερχόμενα data σε μια latent distribution.
- Εκπαιδεύουμε τον decoder να ανασυνθέτει τα δεδομένα από sampled latent variables.
- Σελ. 10 από τα slides: « both (encoder and decoder) are trained to minimise the reconstruction error between the encoded-decoded data and the original data.»

Generative Adversarial Networks (GANs):

- Έχουμε alternating training: τόσο ο generator όσο και ο discriminator εκπαιδεύονται για 1 ή παραπάνω epochs. Ένα GAN μπορεί να έχει 2 loss functions: 1 για το generator training και 1 για τον discriminator.
- Ο generator παραμένει σταθερός κατά τη διάρκεια εκπαίδευσης του discriminator. Προσπαθώντας να διακρίνει τα true από τα false data, ο discriminator μαθαίνει τα flaws από τις παραγωγές του generator.
- Ομοίως, ο discriminator παραμένει σταθερός κατά τη διάρκεια εκπαίδευσης του generator και αυτή η διαδικασία εναλλάσσεται μεταξύ των 2 εκπαιδεύσεων.

Diffusion models:

- Εκπαιδεύουμε το μοντέλο να μαθαίνει να αντιστρέψει το diffusion process.
- Ανασυνθέτουμε data από θορυβώδη samples.

- Η εκπαίδευση περιλαμβάνει μια ακολουθία βημάτων όπου ο θόρυβος αφαιρείται σταδιακά.

iii) η αποτελεσματικότητα:

Variational Autoencoders (VAEs):

- Αρκετά efficient λόγω απλών loss functions.
- Γρήγορο inferencing, αρκεί ένα pass από encoder και decoder.

Generative Adversarial Networks (GANs):

- Συχνά απαιτούν πολλές επαναλήψεις εκπαίδευσης για να επιτευχθεί καλή ισορροπία μεταξύ generator και discriminator.
- Η εκπαίδευση περιλαμβάνει alternating updates, τα οποία μπορεί να είναι υπολογιστικά κοστοβόρα.
- Η αποτελεσματικότητα ενδέχεται να μην είναι η επιθυμητή, καθώς με την εκπαίδευση ο generator βελτιώνεται και παράγει πολύ ρεαλιστικά data, οπότε ο discriminator αποφασίζει με 50% πιθανότητα («the discriminator flips a coin to make its prediction»). Οπότε εάν η εκπαίδευση συνεχίσει περαιτέρω, ο discriminator ενδέχεται να μειώσει την επίδοση του μοντέλου, αφού δεν δίνει πλέον χρήσιμο feedback.

Diffusion models:

- Υπολογιστικά κοστοβόρο λόγω της επαναληπτικής διαδικασίας αφαίρεσης θορύβου.
- Η εκπαίδευση περιλαμβάνει πολλά diffusions και reverse diffusions για την προσομοίωση της διάχυσης και της αφαίρεσης θορύβου.

iv) οι απαιτήσεις:

Variational Autoencoders (VAEs):

- Μέτρια χρήση μνήμης.
- Μέτριος χρόνος εκπαίδευσης που αυξάνεται με περίπλοκα data.
- Απαιτείται μεγάλος αριθμός από training data για καλή αναπαράσταση της κατανομής τους.

Generative Adversarial Networks (GANs):

- Υψηλή χρήση μνήμης λόγω πιο περίπλοκης αρχιτεκτονικής δικτύων.
- Ο χρόνος εκπαίδευσης μπορεί να είναι μεγάλος, λόγω της ανάγκης για πολλές επαναλήψεις και πιθανής αστάθειας.

Diffusion models:

- Πολύ υψηλή χρήση μνήμης λόγω αποθήκευσης πολλών βημάτων diffusion.
- Μακροχρόνιοι χρόνοι εκπαίδευσης λόγω της επαναληπτικής φύσης της διαδικασίας.
- Απαιτούν εκτεταμένα training data για να μάθουν με ακρίβεια τη διαδικασία reverse diffusion.

v) οι ομοιότητες και οι διαφορές:

Ομοιότητες:

- Και τα 3 μοντέλα είναι Generative AI Models που εκπαιδεύονται σε data distribution.
- Καθένα περιέχει κάποια μορφή για latent space ή κάποιο ενδιάμεσο representation στη δομή του.
- Μπορούν να χρησιμοποιηθούν για παρόμοια tasks, όπως image generation, data augmentation, anomaly detection, κλπ.

Διαφορές:

- Τα VAEs χρησιμοποιούν πιθανοτικό framework, τα GANs χρησιμοποιούν adversarial framework, και τα Diffusion Models χρησιμοποιούν θερμοδυναμικό framework.
- Τα VAEs αποσκοπούν κυρίως σε reconstruction και regularization, τα GANs σε adversarial training, και τα Diffusion Models σε επαναληπτικό denoising.
- Διαφέρουν σε Training stability and πολυπλοκότητα.

vi) πλεονεκτήματα και μειονεκτήματα:

Variational Autoencoders (VAEs):

1. Πλεονεκτήματα:

- Πιο εύκολη φάση εκπαίδευσης από τα GANs.
- Χρήσιμα για representation learning και data compression.

2. Μειονεκτήματα

- Συνήθως παράγουν πιο θολές εικόνες σε σχέση με τα GANs.

Generative Adversarial Networks (GANs):

1. Πλεονεκτήματα:

- Αποτελεσματικά για data augmentation και synthetic data generation.
- Πολλά variations (conditional GANs, style GANs) άρα και περισσότερο flexibility.

2. Μειονεκτήματα:

- Δύσκολη και unstable φάση εκπαίδευσης.
- Απαίτηση για hyperparameter tuning.

Diffusion models:

1. Πλεονεκτήματα:

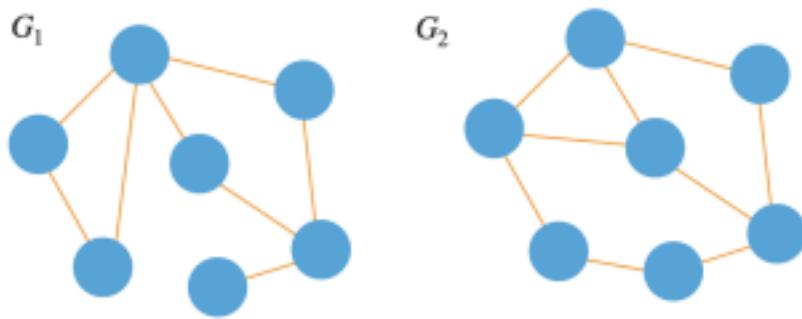
- Το θεωρητικό υπόβαθρο στη θερμοδυναμική παρέχει ένα robust framework.
- Πολύ μικρότερη πιθανότητα για mode collapse (σε σχέση GANs).

2. Μειονεκτήματα:

- Αργή φάση εκπαίδευσης και υπολογιστικά κοστοβόρο.
- Απαιτείται δύσκολος και λεπτομερής σχεδιασμός του diffusion process.
- Υψηλές απαιτήσεις μνήμης και υπολογιστικών πόρων.

Άσκηση 6 (Graph Neural Networks)

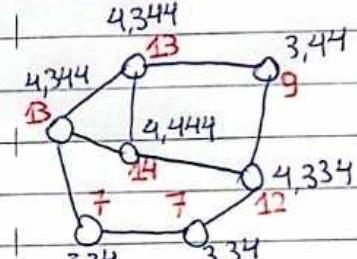
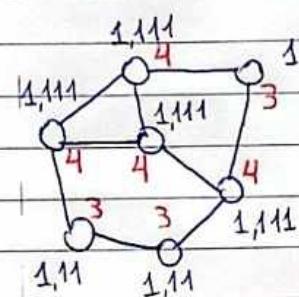
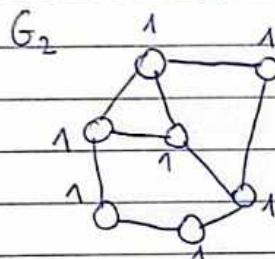
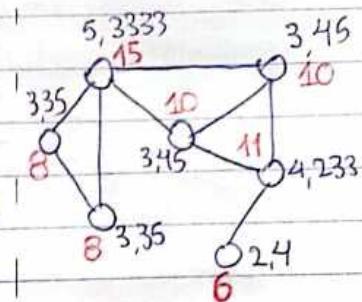
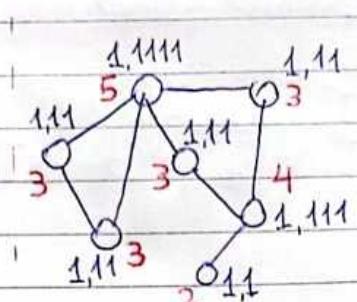
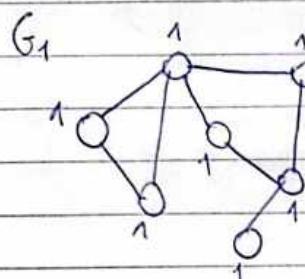
Δίνονται τα παρακάτω γράφοι:



Θα υπολογίσουμε την ομοιότητα τους με βάση τον πυρήνα Weisfeiler-Lehman (WL) με γνώμονα το παράδειγμα του slide 18 του αντίστοιχου αρχείου της σελίδας του μαθήματος.

Η λύση είναι χειρόγραφη και παρατίθεται στην επόμενη σελίδα, ώστε να είναι όσο το δυνατόν πιο ευκρινής:

Άσκηση 6 (Graph Neural Networks)



HASH
1,1→2
1,11→3
1,111→4
1,1111→5

HASH
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

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

$$4,344 \rightarrow 13$$

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

$$5,3333 \rightarrow 15$$

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

$$K(G_1, G_2)_{\max} = 49 + 1 + 16 + 1 + 1 + 1 + 4 + 4 + 1 + 1 = 79$$

Η ορολογία των με βάση των πυρήνων Weisfeiler-Lehman είναι απλά το τέλος μέρους, στις οποίες οι δύο γράφοι εμφανίζουν υψηλό βαθμό συμβένεται.