

# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

## ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

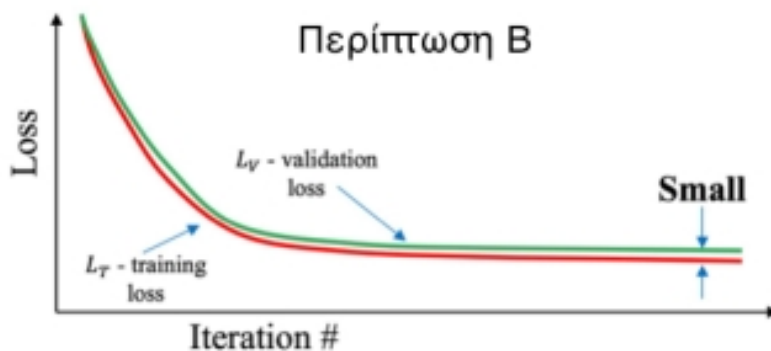
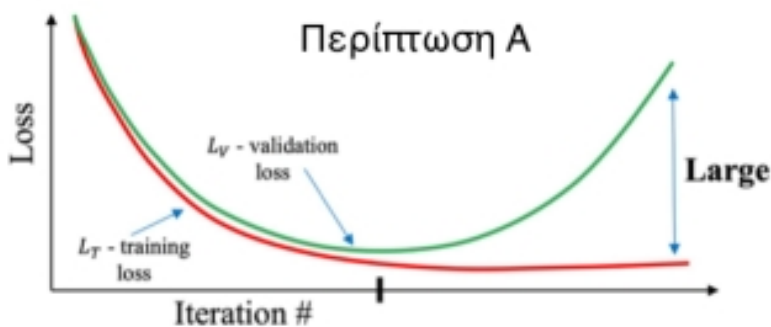
### ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΒΑΘΙΑ ΜΑΘΗΣΗ

#### Σειρά Αναλυτικών Ασκήσεων

Λουκάς Άγγελος (03119877)

Email: lksaggelos@gmail.com

### Άσκηση 1 (Multi Layer Perceptron - Regularization)



#### Ερώτημα 1

Βλέπουμε ότι το underfitting είναι συνήθως η αιτία των αποκλίσεων γύρω από το πρώτο μισό. Στην ουσία, το underfitting συμβαίνει όταν το μοντέλο μας δεν προσαρμόζεται καλά. Δεδομένου ότι το μοντέλο μας πρέπει πρώτα να εκπαιδευτεί, είναι λογικό ότι η απώλεια είναι υψηλή στην αρχή. Ωστόσο, καθώς ολοκληρώνονται περισσότερες επαναλήψεις, με σωστή προσαρμογή, η απώλεια θα πρέπει να μειώνεται. Καθώς κάθε δείγμα αρχίζει να γίνεται μοναδικό, αντιμετωπίζουμε το πρόβλημα του overfitting, όπου το μοντέλο μας έχει

υπερεκπαιδευτεί και χάνει την κατάλληλη γενίκευση, μόλις επιτύχουμε τη βέλτιστη χωρητικότητα. Βλέπουμε αυτό το πρόβλημα συγκεκριμένα στο σενάριο A. Παρόλο που το μοντέλο εκπαίδευσης φαίνεται να έχει μια καλή απώλεια εκπαίδευσης, το μοντέλο παρουσιάζει τελικά μια σημαντική αύξηση των απωλειών στο validation loss, γεγονός που οδηγεί σε ένα μη ικανοποιητικό αποτέλεσμα. Το overfitting είναι η κύρια αιτία της σοβαρής υστέρησης γενίκευσης του μοντέλου. Μια ελάχιστη απόκλιση στις απώλειες validation και training είναι αυτό που επιδιώκουμε. Αντίθετα, το μοντέλο B παρουσιάζει ισχυρή γενίκευση, που σημαίνει ότι μαθαίνει τα underlying patterns πιο καλά και αποφεύγει το overfit.

## Ερώτημα 2

Για το μοντέλο A, θα έχουμε το καλύτερο δυνατό μοντέλο την στιγμή πριν αρχίσει να αυξάνεται το training loss λόγω κακού generalization που θα αρχίσει να γίνεται μέσω overfitting.

Για το μοντέλο B, δεδομένου ότι το μοντέλο έχει ήδη κάνει μια εξαιρετική γενίκευση και δεν φαίνεται να ωφελείται από περισσότερες εποχές, ο αριθμός των εποχών που θα οδηγήσει στο καλύτερο εφικτό μοντέλο μάθησης είναι το σημείο στο οποίο παρατηρούμε ένα plateau στην απόσταση των training και validation loss.

## Ερώτημα 3

Η προσέγγιση Early Stopping είναι ένας τρόπος βελτίωσης της απόδοσης του μοντέλου. Με αυτή τη μέθοδο, σταματάμε τη διαδικασία εκπαίδευσης μόλις παρατηρήσουμε ότι το validation loss χειροτερεύει, καθώς αυτό υποδηλώνει ότι έχει συμβεί overfitting, το οποίο είναι ανεπιθύμητο. Το dropout είναι μια άλλη μέθοδος βελτίωσης που χρησιμοποιείται στα νευρωνικά δίκτυα για την αποφυγή του overfitting. Λειτουργεί με την τυχαία «απόρριψη» ενός υποσυνόλου νευρώνων του δικτύου κατά τη διάρκεια της εκπαίδευσης.

## Ερώτημα 4

Το testing set είναι απαραίτητο για την αξιολόγηση του μοντέλου μας ώστε να μπορούμε να έχουμε μια αντικειμενική εικόνα για αυτό, καθώς το μοντέλο τώρα δοκιμάζεται σε νέα δεδομένα. Με τον τρόπο αυτό, μπορούμε να αξιολογήσουμε την πραγματική ικανότητα γενίκευσης του μοντέλου και να αποφύγουμε την υπερεκτίμηση της ακρίβειας ως αποτέλεσμα του overfitting στα σύνολα training και validation.

## Άσκηση 2 (Representation Learning - Autoencoders)

### Ερώτημα α

Κάθε είσοδος στο autoencoder έρχεται από τα skipgram vectors και αφού οι διαστάσεις τους είναι  $256 \times 1$ , οι εισοδοί  $x_i$  θα έχουν και αυτοί διαστάσεις  $256 \times 1$ .

### Ερώτημα β

Η είσοδός μας λαμβάνει ένα διάνυσμα εισόδου 256 διαστάσεων τα οποία στέλνονται στα στρώματα κωδικοποίησης (Downsampling). Το διάνυσμα εισόδου διέρχεται από διαδοχικά κρυφά στρώματα για την συμπίεση των δεδομένων που μετατρέπουν τις διαστάσεις με τον εξής τρόπο:  $256 \rightarrow 500 \rightarrow 250 \rightarrow 50$ . Ύστερα ξεκινάει η διαδικασία

αποσυμπίεσης(Upsampling) που μετατρέπει τις διαστάσεις με τον εξής τρόπο:  
50 → 250 → 500 → 256.

Ο σκοπός του στρώματος εξόδου είναι να παράγει ένα διάνυσμα που ταιριάζει με την αρχική είσοδο από άποψη διαστάσεων. Αυτό είναι ζωτικής σημασίας για την ικανότητα του αυτόματου κωδικοποιητή να ανακατασκευάζει τα δεδομένα εισόδου.

### Ερώτημα γ

Η λανθάνουσα αναπαράσταση του αυτόματου κωδικοποιητή αντιστοιχεί στο μικρότερο στρώμα της κρυφής δομής. Δεδομένου ότι οι διαστάσεις των κρυφών στρωμάτων είναι [500,250,50,250,500], το μικρότερο στρώμα είναι το μεσαίο στρώμα, το οποίο έχει διάσταση 50.

## Άσκηση 3 (Recurrent Neural Networks)

### Ερώτημα α

RNN Παράμετροι:

- Input to hidden layer matrix  $W_{hx}$
- Hidden to hidden layer matrix  $W_{hh}$
- Hidden to output layer matrix  $W_{yx}$
- ReLU activation function
- Zero bias vectors
- Initial hidden state  $h_0 = [0, 0]^T$

Διαδικασία:

Σε κάθε χρονικό βήμα  $t$ , το RNN λαμβάνει μια είσοδο  $x_t$  η οποία είναι μια συνένωση της προηγούμενης ενσωμάτωσης εξόδου  $y_{t-1}$  και της αναπαράστασης εικόνας  $c$ .

- Η νέα κρυφή κατάσταση  $h_t$  υπολογίζεται χρησιμοποιώντας την είσοδο  $x_t$  και την προηγούμενη κρυφή κατάσταση  $h_{t-1}$ .
- Τα logits εξόδου υπολογίζονται χρησιμοποιώντας τον πίνακα  $W_{hx}$  του κρυφού στρώματος προς το στρώμα εξόδου.
- Η άπληστη αποκωδικοποίηση χρησιμοποιείται για την επιλογή της λέξης με την υψηλότερη πιθανότητα (logit) σε κάθε βήμα.

$$\text{Image Features Average: } C = \frac{1}{4}(i_1 + i_2 + i_3 + i_4) = [1 \ 1]^T$$

$$\text{Initial hidden state : } h_0 = [0 \ 0]^T$$

$x_1 = [0 \ 0 \ 0 \ 1 \ 1]^T$  Η είσοδος για  $t=1$  είναι μία σύνδεση των  $y_{\text{start}}$  και  $c$ :

$$h_i = \text{ReLU}(W_{hx} \cdot x_i + W_{hh} \cdot h_{i-1} + b_h)$$

$$\text{output logits: } z_i = W_{yh} \cdot h_i + b_y$$

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

$$z_1 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -10 \\ 3 \\ 3 \\ 2 \\ 4 \\ 2.9 \end{bmatrix}$$

Η πιο μεγάλη τιμή είναι το 4 άρα το πρώτο logit θα είναι cat.

$$y_1 = y_{cat} = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$$

Επαναλαμβάνουμε την διαδικασία για t=2 και t=3:

$$x_2 = \begin{bmatrix} y_1 \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

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

$$z_2 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} -15 \\ 0 \\ 3 \\ 6 \\ 9 \\ 8.7 \end{bmatrix}$$

Η πιο μεγάλη τιμή είναι το 9 άρα το δεύτερο logit θα είναι staring.

$$y_2 = y_{staring} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$$

$$x_3 = \begin{bmatrix} y_2 \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

$$h_3 = \text{ReLU} \left( \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 2 \\ 7 \end{bmatrix}$$

$$z_3 = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 7 \end{bmatrix} = \begin{bmatrix} -45 \\ 21 \\ 16 \\ 18 \\ -1 \\ 5.8 \end{bmatrix}$$

Η πιο μεγάλη τιμή είναι το 21 άρα το τρίτο logit θα είναι stop.

$$y_3 = y_{\text{stop}} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

### Ερώτημα β

Initial hidden state :  $h_0 = [00]^T$

$$K = V = \begin{bmatrix} i_1 & i_2 & i_3 & i_4 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix} \text{Θα ισχύει πως:}$$

$$\text{AttentionScore} = \text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d_k}} \right) \cdot V$$

Όπου :

- Q is the matrix of query vectors.
- K is the matrix of key vectors.
- V is the matrix of value vectors.
- $d_k=2$  is the dimensionality of the key vectors.
- the key vectors are  $i_1, i_2, i_3, i_4$ .

$$K = V = \begin{bmatrix} i_1 & i_2 & i_3 & i_4 \end{bmatrix}^T = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{bmatrix}^T \text{Για } t=1 \text{ θα έχουμε:}$$

$$q = h_0^T = [0 \ 0]$$

$$AttentionScore_1 = softmax\left(\frac{q \cdot K^T}{\sqrt{d_k}}\right) \cdot V = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Και τώρα μπορούμε να βρούμε τα attention weights:

$$AttentionWeights_{1,1} = softmax(AttentionScore_{1,1}) = \frac{e^0}{\sum_{k=1}^4 e^0} = 0.25$$

$$AttentionWeights_{1,1} = AttentionWeights_{1,2} = AttentionWeights_{1,3} = AttentionWeights_{1,4}$$

Θα ορίσουμε

$AttentionWeights_{i,j} = a_{i,j}$  Για το υπολογισμό του  $c_1$  :

$$c_1 = \sum_{j=1}^4 a_{1,j} \cdot i_j = \frac{1}{4} \cdot \left( \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Έχουμε ίδιο  $c_1$  με το προηγούμενο ερώτημα και ίδιο  $h_0$  άρα θα βγάλουμε ίδιο αποτέλεσμα στην πρώτη επανάληψη.

## Άσκηση 4 (Convolutional Neural Networks)

Ας υποθέσουμε ότι έχουμε ένα δίκτυο όπως το AlexNet που έχει ένα φίλτρο  $11 \times 11 \times 3$  στο πρώτο επίπεδο συνέλιξης και  $227 \times 227 \times 3$  εικόνες. Το δίκτυο περιλαμβάνει συνολικά 96 φίλτρα, με μηδενική συμπλήρωση και βήμα 4. Συνεπώς, έχουμε:

Πλήθος φίλτρων  $K = 96$  , Μέγεθος φίλτρων  $F = 11$ , Stride  $S = 4$ , Padding  $P = 0$

### Ερώτημα α

Για τον υπολογισμό των διαστάσεων στην έξοδο του πρώτου convolution layer θα ακολουθήσουμε τους εξής τύπους:

$$W_2 = \left( \frac{W_1 - F + 2P}{S} \right) + 1 = 55$$

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

$$D_2 = K = 96$$

Οι διαστάσεις θα είναι  $W_2 \times H_2 \times D_2 = 55 \times 55 \times 96$ .

### Ερώτημα β

Ο αριθμός των units στο πρώτο convolutional layer είναι το γινόμενο των διαστάσεων του άρα  $55 \times 55 \times 96 = 290400$

### Ερώτημα γ

Για τον υπολογισμό των παραμέτρων του πρώτου convolutional layer θα θεωρήσουμε τα παρακάτω.

$$W_c = K^2 \times C \times N = \text{Number of weights of the convolutional layer}$$

$$K = \text{Size of kernels used in the convolutional layer}$$

$$C = \text{Number of channels of the input image}$$

$$N = \text{Number of kernels}$$

$$B_c = N = \text{Number of biases of the convolutional layer}$$

$$P_c = W_c + B_c = \text{Number of parameters of the convolutional layer}$$

Άρα θα βγάλουμε:

$$W_c = 11 \times 11 \times 3 \times 96$$

$$B_c = 96$$

$$P_c = 34944$$

Άρα έχουμε 34944 παραμέτρους.

### Ερώτημα δ

Σε ένα FeedForward layer, κάθε pixel της εικόνας συνδέεται με κάθε unit του layer. Αυτό σημαίνει ότι το input θα έχει συνολικά  $227 \times 227 \times 3 = 154,587$  διαστάσεις. Οι παράμετροι υπολογίζονται ως εξής:

$$N_{\text{param}} = N_{\text{input}} + 1 \times N_{\text{units}} = 154,588 \times 256 = 39,573,568$$

Άρα συνολικά 39,573,568 παραμέτρους.

## Άσκηση 5 (Generative models)

Πληροφορίες λήφθηκαν από τους παρακάτω συνδέσμους:

- [An Introduction to Variational Autoencoders](#)
- [Tutorial on Variational Autoencoders](#)
- [Efficient generative adversarial networks using linear additive-attention Transformers](#)
- [Diffusion Models: A Comprehensive Survey of Methods and Applications](#)

### Ερώτημα α

#### Θεμελιώδεις Αρχές

Οι VAEs συνδυάζουν την αρχιτεκτονική των Autoencoders με στοχαστική μοντελοποίηση. Αποτελούνται από δύο βασικά μέρη:

- Encoder: Μετατρέπει τα δεδομένα εισόδου σε παραμέτρους μιας κατανομής πιθανότητας (συνήθως Gaussian).
- Decoder: Χρησιμοποιεί δείγματα από αυτή την κατανομή για να ανακατασκευάσει τα δεδομένα.

#### Διαδικασία Εκπαίδευσης

Εκπαιδεύονται με βελτιστοποίηση της συνάρτησης κόστους χρησιμοποιώντας αλγορίθμους όπως ο Adam. Η εκπαίδευση περιλαμβάνει μείωση της απώλειας ανακατασκευής, διασφαλίζοντας ότι τα ανακατασκευασμένα δεδομένα είναι όσο το δυνατόν πιο κοντά στα αρχικά και μείωση της KL απόκλισης, διατηρώντας την κατανομή των λανθανουσών μεταβλητών κοντά στην κανονική κατανομή.

#### Αποτελεσματικότητα

Τα δεδομένα που παράγονται από τους VAEs είναι ποιοτικά καλά αλλά όχι τέλεια και έχουν γρήγορη δειγματοληψία.

#### Απαιτήσεις

Μεγάλα σύνολα δεδομένων είναι απαραίτητα για βελτίωση της ποιότητας των παραγόμενων δεδομένων.

### Ερώτημα β

#### Θεμελιώδεις Αρχές

Τα GANs αποτελούνται από δύο νευρωνικά δίκτυα. Τον Generator που δημιουργεί ψεύτικα δεδομένα και τον Discriminator που διακρίνει αν τα δεδομένα είναι πραγματικά ή ψεύτικα.



### Διαδικασία Εκπαίδευσης

Η εκπαίδευσή τους είναι ένα ανταγωνιστικό παιχνίδι όπου ο Generator προσπαθεί να εξαπατήσει τον Discriminator. Ο Discriminator εκπαιδεύεται με πραγματικά και ψεύτικα δεδομένα, ενώ ο Generator προσπαθεί να βελτιώσει την ικανότητά του να εξαπατήσει τον Discriminator.

### Αποτελεσματικότητα

Λόγω της ανταγωνιστικότητας των δύο αυτών μοντέλων παράγονται πολύ ποιοτικά δεδομένα με καλή ταχύτητα δειγματοληψίας.

### Απαιτήσεις

Ο χρόνος εκπαίδευσης είναι μεγάλος με σημαντική υπολογιστική απαίτηση ισχύς και απαιτούν μεγάλα σύνολα δεδομένων.

## **Ερώτημα γ**

### Θεμελιώδεις Αρχές

Τα Diffusion Models είναι μοντέλα που μαθαίνουν να δημιουργούν δεδομένα μέσω μιας διαδικασίας διάχυσης. Αυτά τα μοντέλα σταδιακά προσθέτουν θόρυβο σε μια εικόνα και εκπαιδεύονται για να αντιστρέψουν αυτή τη διαδικασία, αναδημιουργώντας την αρχική εικόνα από θόρυβο.

### Διαδικασία Εκπαίδευσης

Έχει δύο διεργασίες, την forward process που προσθέτει θόρυβο σταδιακά στα δεδομένα και την reverse process που εκπαιδεύεται για να αφαιρέσει τον θόρυβο και να αναδημιουργήσει τα δεδομένα.

### Αποτελεσματικότητα

Μπορεί να παράγει υψηλής ποιότητας δεδομένα αλλά η διαδικασία είναι αργή λόγω των πολλών βημάτων της αντίστροφης διάχυσης.

### Απαιτήσεις

Χρειάζεται πολύ χρόνο στην εκπαίδευση λόγω της πολυπλοκότητας της διαδικασίας και απαιτεί μεγάλα σύνολα δεδομένων.

## **Ομοιότητες**

Όλα χρησιμοποιούνται για τη δημιουργία συνθετικών δεδομένων και έχουν εφαρμογές στη μηχανική μάθηση και τεχνητή νοημοσύνη και όλα απαιτούν μεγάλα σύνολα δεδομένων για την εκπαίδευσή τους.

## **Διαφορές**

VAEs χρησιμοποιούν κωδικοποιητές/αποκωδικοποιητές, τα GANs χρησιμοποιούν ένα ζεύγος δικτύων που ανταγωνίζονται, ενώ τα Diffusion Models χρησιμοποιούν μια διαδικασία θορύβου και αντίστροφης θορύβωσης. Επίσης οι VAEs και τα GANs έχουν διαφορετικές προσεγγίσεις στη συνάρτηση κόστους, ενώ τα Diffusion Models βασίζονται σε διαδικασίες θορύβου.

## **Πλεονεκτήματα**

VAEs: Γρήγορη δειγματοληψία, καλή ποικιλία δεδομένων, μέτριες απαιτήσεις σε μνήμη και χρόνο.

GANs: Πολύ υψηλή ποιότητα δεδομένων, ικανοποιητική ταχύτητα δειγματοληψίας.

Diffusion Models: Πολύ λεπτομερή και υψηλής ποιότητας δεδομένα.

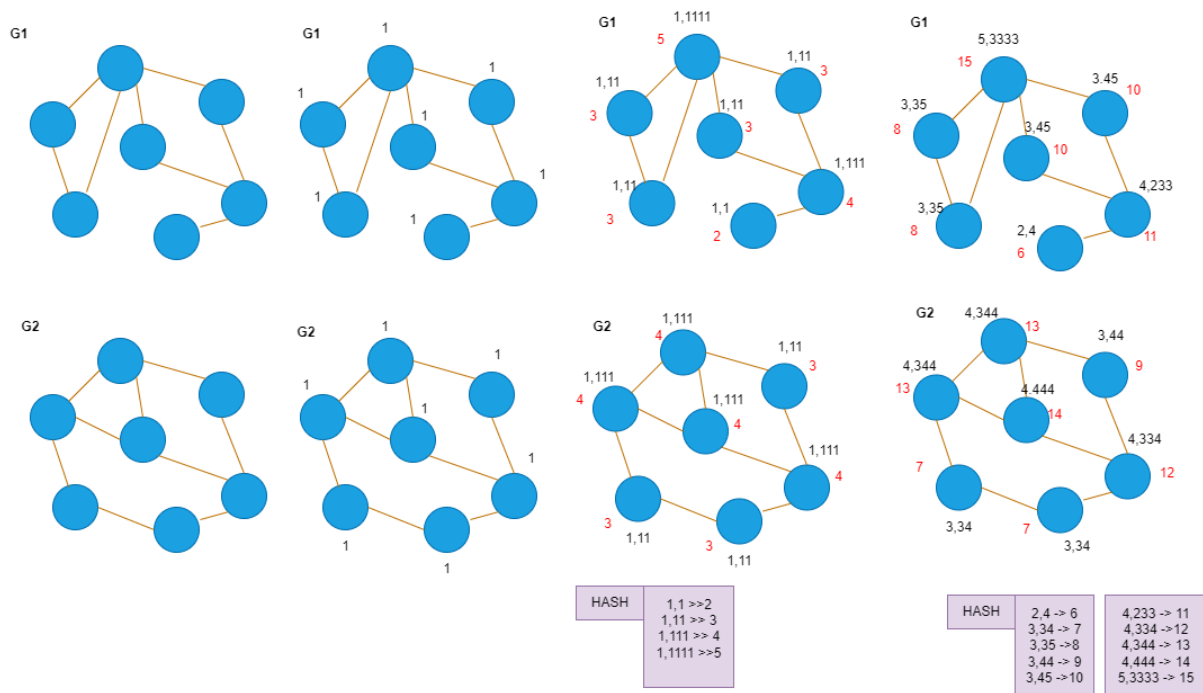
## **Μειονεκτήματα**

VAEs: Ελαφρώς χαμηλότερη ποιότητα δεδομένων σε σύγκριση με τα GANs και τα Diffusion Models.

GANs: Υψηλές απαιτήσεις σε μνήμη και χρόνο, ασταθής εκπαίδευση.

Diffusion Models: Αργή διαδικασία δειγματοληψίας, υψηλές απαιτήσεις σε μνήμη και χρόνο εκπαίδευσης.

## Άσκηση 6 (Graph Neural Networks)



Ακολουθώντας τα βήματα του αλγορίθμου Weisfeiler-Lehman, μπορούμε να υπολογίσουμε αποτελεσματικά την ομοιότητα μεταξύ των δύο γραφημάτων G1 και G2.

Ο αλγόριθμός μας σταμάτησε στον τρίτο υπολογισμό του γράφου καθώς βγαίνουν διαφορετικές οι τιμές των κόμβων.

Μετά από τις επαναλήψεις του αλγορίθμου, τα αποτελέσματα είναι τα εξής:

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

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

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