

Multiple Choice Questions

1. Ποια από τις παρακάτω προτάσεις για τα recurrent neural networks (RNN) είναι εσφαλμένη;
- Τα απλά Recurrent Neural Networks κατορθώνουν στην πράξη να μοντελοποιούν αποτελεσματικά τις μακρινές εξαρτήσεις επειδή μέσω της επαναληπτικής δομής τους αντιμετωπίζουν επιτυχώς το πρόβλημα των vanishing gradients.
 - Τα Gated Recurrent Unit (GRU) είναι απλούστερα μοντέλα από τα Long Short-Term Memory (LSTM) και ως εκ τούτου τείνουν να έχουν μικρότερους χρόνους εκπαίδευσης.
 - Τα RNNs μπορούν να χειριστούν ακολουθίες αυθαίρετου μήκους, ενώ τα feedforward neural networks δεν μπορούν.
 - Η εκπαίδευση των RNNs είναι δύσκολη λόγω των προβλημάτων των vanishing & exploding gradients.

Recurrent Neural Networks RNNs: μοντελοποιούν σειρές με εσωτερική μνήμη (κρατάνε προηγούμενη κατάσταση). Δέχονται είσοδο αυθαίρετου μήκους → πλεονέκτημα έναντι feedforward networks.

Προβλήματα εκπαίδευσης RNNs:

- Vanishing gradients → δύσκολη εκμάθηση μακρινών εξαρτήσεων.
- Exploding gradients → αριθμητική αστάθεια
- Η εκπαίδευση γίνεται με Backpropagation Through Time (BPTT).

LSTM & GRU

- Αντιμετωπίζουν τα παραπάνω προβλήματα μέσω πυλών ελέγχου πληροφορίας.
- GRU: πιο απλό από LSTM (λιγότερες πύλες), συνήθως πιο γρήγορο στην εκπαίδευση.
- Και τα δύο μοντελοποιούν μακρινές εξαρτήσεις καλύτερα από απλό RNN.

2. Υποθέστε πως θέλουμε να εφαρμόσουμε τον αλγόριθμο BPE στις ακόλουθες λέξεις με τη συχνότητα που υποδεικνύεται στον πίνακα 1. Ποιος συνδυασμός χαρακτήρων θα συγχωνευτεί (merge) στο 2ο πέρασμα (iteration) του αλγορίθμου;

- Οι χαρακτήρες l, o → lo
- Οι χαρακτήρες s, t → st
- Οι χαρακτήρες a, s → as
- Οι χαρακτήρες a, st → ast

word	frequency
lock	1
stat	1
fast	3
last	3
stop	4
as	5
lost	6

i=1:

lo: lock(1), lost(6) → 7	ta: stat(1) → 1	la: last(3) → 3	n/o συχνό bigram: st(17) ~ merge st
oc: lock(1) → 1	at: stat(1) → 1	to: stop(4) → 4	
ck: lock(1) → 1	fa: fast(3) → 3	op: stop(4) → 4	
st: stat(1), fast(3), last(3), stop(4), lost(6) → 17	as: fast(3), last(3), as(5) → 11	os: lost(6) → 6	

i=2:

lo: lock(1), lost(6) → 7	at: stat(1) → 1	sto: stop(4) → 4	n/o συχνό bigram: lo(7) ~ merge lo
oc: lock(1) → 1	fa: fast(3) → 3	op: stop(4) → 4	
ck: lock(1) → 1	ast: fast(3), last(3) → 6	as: as(5) → 5	
sta: stat(1) → 1	la: last(3) → 3	ost: lost(6) → 6	

3. Σε τι από τα παρακάτω αφορά το generative learning trilemma;
- Στην πρόκληση της ταυτόχρονης επίτευξης υψηλής ποιότητας παραγωγής εικόνων, ποικίλων εξόδων και γρήγορης ταχύτητας δειγματοληψίας στα παραγωγικά μοντέλα.
 - Στον συμβιβασμό μεταξύ ακρίβειας, ερμηνευσιμότητας και υπολογιστικής αποδοτικότητας στη μηχανική μάθηση.
 - Στη δυσκολία εκπαίδευσης βαθιών νευρωνικών δικτύων με εξαφανιζόμενες κλίσεις (vanishing gradients).
 - Στην αδυναμία των μοντέλων να μάθουν από δεδομένα υψηλών διαστάσεων.

Το Generative Learning Trilemma αναφέρεται σε ένα θεμελιώδες πρόβλημα που αντιμετωπίζουν τα παραγωγικά μοντέλα (π.χ. VAEs, GANs, Diffusion Models, κ.ά.):

- High sample quality – η παραγόμενη έξοδος να είναι ρεαλιστική και καθαρή.
- High diversity – να παράγονται ποικίλες εξόδους (π.χ. όχι πάντα παρόμοια πρόσωπα).
- Fast sampling – η δειγματοληψία να είναι γρήγορη και πρακτική για εφαρμογές.

4. Ποια από τις παρακάτω προτάσεις για τους Transformers είναι αληθής;
- Ένας Transformer μπορεί να έχει είτε απλό attention είτε self-attention, αλλά όχι και τα δύο.
 - Δεδομένου ότι οι Transformers είναι εγγενώς ευαίσθητοι στη σειρά των λέξεων, η προσθήκη των positional embeddings μπορεί να μειώσει αυτή τη ευαισθησία.
 - Το multi-head attention μπορεί να εμφανίζεται είτε στην πλευρά του encoder είτε στην πλευρά του decoder ενός Transformer, αλλά όχι και στις δύο.
 - Καμία από τις παραπάνω.

Οι Transformers χρησιμοποιούν και self-attention και απλό attention:

- Self-attention: σε encoder και decoder (για τις ίδιες θέσεις).
- Encoder-decoder attention: στο decoder (για σύνδεση με τις εξόδους του encoder).
- Multi-head attention: σε encoder και decoder πολλαπλές φορές

Από μόνοι τους ΔΕΝ είναι ευαίσθητοι στη σειρά (λόγω του ότι δεν έχουν recurrence ή convolution). ➤ χρειάζονται positional embeddings για να αποκτήσουν αίσθηση της θέσης.

5. Τι από τα παρακάτω στοχεύει να διασφαλίσει η διαδικασία regularisation κατά την εκπαίδευση των variational autoencoders;
- Δύο κοντινά σημεία στο λανθάνοντα χώρο (latent space) δεν θα παράγουν τελείως διαφορετικές εξόδους, όταν αποκωδικοποιηθούν.
 - Κάθε σημείο του λανθάνοντα χώρου (latent space) θα αποδίδει "ουσιαστικό" περιεχόμενο όταν αποκωδικοποιείται.
 - Και το A και το B.
 - Ούτε το A ούτε το B.

Οι VAEs επιχειρούν να μάθουν έναν συνεχή και κανονικοποιημένο λανθάνοντα χώρο (latent space) μέσω στοχαστικής δειγματοληψίας και regularization.

- Διασφαλίζει ομαλότητα: κοντινά σημεία στο latent space οδηγούν σε παρόμοιες εξόδους → αυτό κάνει τον χώρο συνεχή.
- Εξαναγκάζει την κατανομή στο latent space να πλησιάζει μια κανονική κατανομή (συνήθως $N(0,1)$) → έτσι όλα τα σημεία στο χώρο μπορούν να παράγουν νόημα κατά την αποκωδικοποίηση, όχι μόνο τα "εκπαιδευμένα".

~ λειτουργικότητα
~ ομαλότητα

6. Η σχέση των Langevin dynamics με τα Score-Based Generative Models έγκειται στο ότι:

- a) Ορίζουν τη συνάρτηση θορύβου που χρησιμοποιείται για την αλλοίωση των δεδομένων κατά τη διάρκεια της εκπαίδευσης.
- b) Καθοδηγούν τη διαδικασία δειγματοληψίας με επαναληπτική αποθρομβοποίηση του τυχαίου θορύβου.
- c) Χρησιμοποιούνται για την προ-εκπαίδευση του score network.
- d) Χρησιμοποιούνται για την άμεση αξιολόγηση της πιθανότητας κατανομής δεδομένων ενός δείγματος.

Score-Based Generative Models (SBGMs): Αντί να μαθαίνουν κατανομή πιθανότητας απευθείας, μαθαίνουν το score function, δηλαδή το gradient της log-πιθανότητας.

Langevin Dynamics: Είναι στοχαστική διαδικασία δειγματοληψίας. Συνδυάζει θόρυβο με το score function για να φτάσει από random noise → σε πραγματικά δεδομένα. Χρησιμοποιείται για να μετατρέψει λευκό θόρυβο σε δείγμα μέσω σταδιακής αποθρομβοποίησης.

score function: $\nabla_x \log p(x)$

στοχαστική διαδικασία δειγματοληψίας: $x_{t+1} = x_t + \frac{\epsilon}{2} \nabla_x \log p(x_t) + \epsilon \cdot \eta_t$

7. Έστω αρχική εικόνα ενός καναλιού 5×5 pixels που συνελίσσεται με πυρήνα διαστάσεων 3×3 με padding = 0 και stride = 1. Οι διαστάσεις του χάρτη χαρακτηριστικών εξόδου είναι:

- a) 3×3
- b) 4×4
- c) 5×5
- d) Δεν μπορούμε να απαντήσουμε.

Input size: $W_{in} \times H_{in} \times C_{in} = 5 \times 5 \times 1$

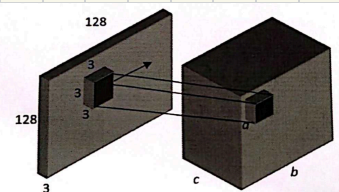
Convolutional layer, kernel $K \times K = 3 \times 3$, stride $S=1$, padding $P=0$:

$$W_{out} = \left\lfloor \frac{W_{in} - K + 2P}{S} \right\rfloor + 1 = \left\lfloor \frac{5 - 3 + 2 \cdot 0}{1} \right\rfloor + 1 = 2 + 1 = 3, \quad H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1 = \left\lfloor \frac{5 - 3 + 2 \cdot 0}{1} \right\rfloor + 1 = 2 + 1 = 3$$

Output size: $W_{out} \times H_{out} = 3 \times 3$

8. Έστω συνελκτικό νευρωνικό δίκτυο (CNN) που παίρνει ως είσοδο εικόνες διαστάσεων $(128, 128, 3)$. Το πρώτο συνελκτικό στρώμα αποτελείται από 32 φίλτρα διαστάσεων $(3, 3, 3)$ το καθένα με padding = 1 και stride = 2. Στο σχήμα φαίνεται ένα από αυτά τα φίλτρα σε μια συγκεκριμένη περιοχή της εισόδου και στην αντίστοιχη περιοχή της εξόδου. Ποιες θα είναι οι τιμές των a, b, c μετά την εφαρμογή του συνελκτικού στρώματος; (Το σχήμα δεν είναι υπό κλίμακα.)

- a) $a=1, b=64, c=32$
- b) $a=3, b=64, c=32$
- c) $a=1, b=32, c=64$
- d) $a=3, b=32, c=64$



Input size: $W_{in} \times H_{in} \times C_{in} = 128 \times 128 \times 3$

Convolutional layer, $N=32$ kernels $K \times K \times D = 3 \times 3 \times 3$, depth=3, stride $S=2$, padding $P=1$:

$$W_{out} = \left\lfloor \frac{W_{in} - K + 2P}{S} \right\rfloor + 1 = \left\lfloor \frac{128 - 3 + 2 \cdot 1}{2} \right\rfloor + 1 = 63 + 1 = 64, \quad H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1 = \left\lfloor \frac{128 - 3 + 2 \cdot 1}{2} \right\rfloor + 1 = 63 + 1 = 64, \quad C_{out} = N \Rightarrow C_{out} = 32$$

Μέγεθος block στο βάθος $\Rightarrow$ depth φίλτρου $\Rightarrow a=3$

Output size: $W_{out} \times H_{out} \times C_{out} = 64 \times 64 \times 32 \Rightarrow b=64, c=32$

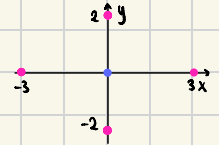
Open-ended Questions

Θέμα 1 (1.7 μονάδες)

Δίνονται τα παρακάτω δεδομένα εισόδου για δύο κλάσεις. Να σχεδιάσετε, για την επίλυση του προβλήματος, ένα Multi Layer Perceptron, δίνοντας την αρχιτεκτονική και τα βάρη του (χωρίς να τα υπολογίσετε από κάποιον αλγόριθμο εκμάθησης).

Κλάση A: $[-3, 0]$, $[0, 2]$, $[0, -2]$, $[3, 0]$

Κλάση B: $[0, 0]$



μη-γραμμικά διαχωρισίμο πρόβλημα $\sim$ MLP με 1 hidden layer

input: 2 neurons $x = (x_1, x_2)^T$

hidden layer: 4 neurons with step-function: $h_i = \text{step}(w_i \cdot x + b_i) \in \{0, 1\}^{4 \times 1}$

output: 1 neuron with step function: $y = \text{step}(w_o \cdot h + b_o) \in \{0, 1\}$

$$\text{step}(z) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

$$y = \begin{cases} 1, & \text{κλάση A} \\ 0, & \text{κλάση B} \end{cases}$$

Θέλουμε να ενεργοποιούνται όταν ένα σημείο βρίσκεται μακριά από το 0 κατά μήκος κάποιου άξονα $\sim$ θέτουμε κατωφλί = 1

νευρώνας h_1 : ενεργοποίηση για $x_1 > 1 \sim$ συνθήκη: $x_1 - 1 > 0 \Rightarrow w_1 = (1, 0), b_1 = -1$

νευρώνας h_2 : ενεργοποίηση για $x_1 < -1 \sim$ συνθήκη: $-x_1 - 1 > 0 \Rightarrow w_2 = (-1, 0), b_2 = -1$

νευρώνας h_3 : ενεργοποίηση για $x_2 > 1 \sim$ συνθήκη: $x_2 - 1 > 0 \Rightarrow w_3 = (0, 1), b_3 = -1$

νευρώνας h_4 : ενεργοποίηση για $x_2 < -1 \sim$ συνθήκη: $-x_2 - 1 > 0 \Rightarrow w_4 = (0, -1), b_4 = -1$

$$\left. \begin{matrix} w_1 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \in \mathbb{R}^{4 \times 2}, \quad b_1 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \in \mathbb{R}^{4 \times 1} \end{matrix} \right\}$$

Θέλουμε να ενεργοποιείται όταν έστω ένα νευρώνας $h_i = 1 \sim$ θέτουμε κατωφλί 0.5

νευρώνας y : ενεργοποίηση για $h_1 + h_2 + h_3 + h_4 > 0.5 \sim$ συνθήκη: $h_1 + h_2 + h_3 + h_4 - 0.5 > 0 \Rightarrow w_o = (1, 1, 1, 1) \in \mathbb{R}^{1 \times 4}, b_o = -0.5$

$x = (0, 0)$	$x = (3, 0)$	$x = (-3, 0)$	$x = (0, 2)$	$x = (0, -2)$	$\sim$ ο MLP ταξινομεί όλα τα δεδομένα σωστά.
$h_1: \text{step}(-1) = 0$	$h_1: \text{step}(2) = 1$	$h_1: \text{step}(-2) = 0$	$h_1: \text{step}(-1) = 0$	$h_1: \text{step}(-1) = 0$	
$h_2: \text{step}(-1) = 0$	$h_2: \text{step}(-4) = 0$	$h_2: \text{step}(4) = 1$	$h_2: \text{step}(1) = 0$	$h_2: \text{step}(1) = 0$	
$h_3: \text{step}(-1) = 0$	$h_3: \text{step}(-1) = 0$	$h_3: \text{step}(-1) = 0$	$h_3: \text{step}(1) = 1$	$h_3: \text{step}(-1) = 0$	
$h_4: \text{step}(-1) = 0$	$h_4: \text{step}(-1) = 0$	$h_4: \text{step}(-1) = 0$	$h_4: \text{step}(-3) = 0$	$h_4: \text{step}(1) = 1$	
$\Rightarrow y = \text{step}(-0.5) = 0$	$\Rightarrow y = \text{step}(0.5) = 1$	$\Rightarrow y = \text{step}(0.5) = 1$	$\Rightarrow y = \text{step}(0.5) = 1$	$\Rightarrow y = \text{step}(0.5) = 1$	
$\Rightarrow$ Κλάση B	$\Rightarrow$ Κλάση A	$\Rightarrow$ Κλάση A	$\Rightarrow$ Κλάση A	$\Rightarrow$ Κλάση A	

Θέμα 2 (1.0 μονάδα)

Έστω ότι έχουμε έναν πράκτορα αυτοενισχυόμενης μάθησης (reinforcement learning), ο οποίος μπορεί να επιλέξει μεταξύ τεσσάρων ενεργειών και να βρεθεί σε μία εκ τεσσάρων καταστάσεων. Παρακάτω δίνεται ο πίνακας ανταμοιβής R του εν λόγω πράκτορα, με τις γραμμές να συμβολίζουν τις καταστάσεις (1 ως 4) και τις στήλες τις ενέργειες (1 ως 4), ενώ η τιμή -1 συμβολίζει ένα μη-αποδεκτό ζεύγος κατάστασης-δράσης. Ας υποθέσουμε ότι τη χρονική στιγμή t ο πράκτορας βρίσκεται στην κατάσταση 2, ενώ ο πίνακας Q έχει ως εξής. Να εξηγήσετε ποια ενέργεια θα επιλέξει ο πράκτορας και πώς θα διαμορφωθεί ο πίνακας Q σύμφωνα με τον αλγόριθμο Q-learning, μετά την επιλογή του αυτή.

Σημειώνεται πως όποια δράση επιλέξει, τον οδηγεί στην αντίστοιχη κατάσταση. Θεωρήστε ρυθμό μάθησης $\alpha = 0.6$ και συντελεστή ελάττωσης ανταμοιβής $\gamma = 0.8$.

$$R = \begin{bmatrix} -1 & -1 & 0 & 50 \\ 10 & -1 & 80 & -1 \\ 20 & -1 & -1 & 40 \\ -1 & 60 & 30 & -1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 50 \\ 30 & 0 & 100 & 0 \\ 40 & 0 & 0 & 100 \\ 0 & 80 & 20 & 0 \end{bmatrix}$$

χρονική στιγμή t : ο agent βρίσκεται στην κατάσταση $s = 2$:

① προσδιορισμός επιτρεπτικών ενεργειών: τιμές $R(2, \cdot) = [10, -1, 80, -1] \Rightarrow \begin{cases} \text{επιτρεπτικές: action 1, 3} \\ \text{μη επιτρεπτικές: action 2, 4} \end{cases}$

② επιλογή ενέργειας: αμοιβές $Q(2, \cdot) = [30, 0, 100, 0]$, για επιτρεπτικές $\Rightarrow \begin{cases} Q(2, 1) = 30 \\ Q(2, 3) = 100 \end{cases} \Rightarrow \max(30, 100) = 100 \Rightarrow$ ο agent επιλέγει action $a = 3$

③ υπολογισμός νέου $Q(2, 3)$ με Q-learning: $Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha(R(s, a) + \gamma \max_{a'} Q(s, a'))$

- τρέχουσα κατάσταση $s = 2$, επιλεγμένη ενέργεια $a = 3 \Rightarrow Q(s, a) = Q(2, 3) = 100, R(s, a) = R(2, 3) = 80$

- ρυθμός μάθησης $\alpha = 0.6$, συντελεστής ελάττωσης $\gamma = 0.8$

- επόμενη κατάσταση $s' = 3$: τιμές $R(3, \cdot) = [20, -1, -1, 40] \Rightarrow \begin{cases} \text{επιτρεπτικές: action 1, 4} \Rightarrow a' = 1, 4 \\ \text{μη επιτρεπτικές: action 2, 3} \end{cases}$

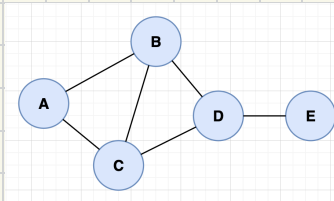
$$\begin{matrix} a' = 1: Q(3, 1) = 40 \\ a' = 4: Q(3, 4) = 100 \end{matrix} \Rightarrow \max_{a'} Q(3, a') = 100$$

$$\Rightarrow Q(2, 3) \leftarrow (1 - 0.6) \cdot 100 + 0.6(80 + 0.8 \cdot 100) = 136 \Rightarrow Q = \begin{bmatrix} 0 & 0 & 0 & 50 \\ 30 & 0 & 136 & 0 \\ 40 & 0 & 0 & 100 \\ 0 & 80 & 20 & 0 \end{bmatrix}$$

Θέμα 3 (1.4 μονάδες)

3.1 Για τον γράφο G_1 του σχήματος να υπολογιστούν οι τιμές ενδιαμεσότητας των κόμβων του.

ενδιαμεσότητα : $C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, σ_{st} : αριθμός shortest path μεταξύ $s-t$
 $\sigma_{st}(v)$: πόσα από αυτά περνάνε από v



s-t	shortest paths	σ_{st}	ενδιαμεσοί
A-B	A-B	1	-
A-C	A-C	1	-
A-D	A-B-D, A-C-D	2	B(1/2), C(1/2)
A-E	A-B-D-E, A-C-D-E	2	B(1/2), C(1/2), D(1)
B-C	B-C	1	-
B-D	B-D	1	-
B-E	B-D-E	1	D(1)
C-D	C-D	1	-
C-E	C-D-E	1	D(1)
D-E	D-E	1	-

$$\triangleright C_B(A) = 0$$

$$\triangleright C_B(B) = \frac{1}{2} + \frac{1}{2} = 1$$

$$\triangleright C_B(C) = \frac{1}{2} + \frac{1}{2} = 1$$

$$\triangleright C_B(D) = 1 + 1 + 1 = 3$$

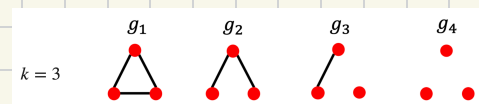
$$\triangleright C_B(E) = 0$$

3.2 Για τον γράφο G_1 του σχήματος να υπολογιστούν το διάνυσμα $\phi(G_1)$ bag of graphlets για $k=3$.

τριάδα	ακμές	# ακμ	είδος	είδος	#
{A,B,C}	AB, AC, BC	3	g_1	g_1	2
{A,B,D}	AB, BD	2	g_2	g_2	4
{A,B,E}	AB	1	g_3	g_3	4
{A,C,D}	AC, CD	2	g_2	g_4	0
{A,C,E}	AC	1	g_3		
{A,D,E}	DE	1	g_3		
{B,C,D}	BC, BD, CD	3	g_1		
{B,C,E}	BC	1	g_3		
{B,D,E}	BD, DE	2	g_2		
{C,D,E}	CD, DE	2	g_2		

$$\Rightarrow \phi(G_1) = [2 \ 4 \ 4 \ 0]$$

από θεωρία:



4 graphlets $\leadsto$ νούμερος 4 διατάξεων