

Multiple Choice Questions

1. Ποια από τις παρακάτω προτάσεις για τα επίπεδα pooling ενός CNN (Convolutional Neural Network) είναι λανθασμένη;
- Η έξοδος των συνελκτικών σταδίων τροφοδοτεί τα στάδια pooling.
 - Η χρήση ενός σταδίου pooling διατηρεί σταθερό το βάθος του χάρτη (τένσορα) χαρακτηριστικών.
 - Μειώνουν τον χρόνο εκπαίδευσης.
 - Καμία από τις παραπάνω.

Τα συνελκτικά επίπεδα (convolutional layers) εξάγουν τοπικά χαρακτηριστικά από την είσοδο και η έξοδός τους τροφοδοτεί τα pooling layers. Τα pooling layers μειώνουν τις διαστάσεις (ύψος × πλάτος) των χαρακτηριστικών χωρίς να αλλάζουν το βάθος (depth). Η μείωση των διαστάσεων οδηγεί σε λιγότερους υπολογισμούς, άρα ταχύτερη εκπαίδευση του δικτύου.

2. Ποιο από τα παρακάτω προεκπαιδευμένα μοντέλα είναι πιο κατάλληλο για μια ένα downstream task που απαιτεί παραγωγή κειμένου με αυτόματη παλινδρόμηση;
- Ένα encoder-only μοντέλο όπως το BERT.
 - Είτε ένα μοντέλο encoder-only όπως το BERT είτε ένα μοντέλο encoder-decoder όπως το T5.
 - Είτε ένα μοντέλο decoder-only όπως το GPT-3 είτε ένα μοντέλο encoder-decoder όπως το T5.
 - Ένα μοντέλο decoder-only όπως το GPT-3.

Η παραγωγή κειμένου με αυτόματη παλινδρόμηση σημαίνει πρόβλεψη της επόμενης λέξης με βάση τις προηγούμενες. Τα decoder-only μοντέλα, όπως το GPT-3, είναι σχεδιασμένα ακριβώς για αυτό. Αντίθετα, το BERT (encoder-only) δεν κάνει παραγωγή, και το T5 (encoder-decoder) είναι πιο γενικό.

3. Ποια από τις παρακάτω προτάσεις σχετικά με τα autoregressive RNN μοντέλα είναι σωστή;
- RNN έχουν απεριόριστη μνήμη, ωστόσο έχουν την τάση να θυμούνται με λιγότερη ακρίβεια τις πιο πρόσφατες λέξεις που παράγουν καθώς το βάρος των αρχικών λέξεων μεγαλώνει όσο ανατροφοδοτείται στα RNN Units.
 - Ακριβώς όπως τα n-gram μοντέλα, τα RNN μπορούν μόνο να θυμηθούν τις τελευταίες n λέξεις που παράγουν, για ένα δεδομένο $n \in \mathbb{N}$.
 - Τα RNN έχουν απεριόριστη μνήμη, ωστόσο για μεγάλες ακολουθίες έχουν μια τάση να «θυμούνται» με μικρότερη ακρίβεια τις αρχικές λέξεις που παράγουν.
 - Τα RNN υποφέρουν από vanishing gradients, αλλά αυτό μπορεί να μετριαστεί με gradient clipping.

Τα autoregressive RNNs έχουν θεωρητικά απεριόριστη μνήμη, σε αντίθεση με τα n-gram μοντέλα, τα οποία θυμούνται μόνο τις τελευταίες n λέξεις. Παρά την απεριόριστη θεωρητική μνήμη, τα RNNs στην πράξη έχουν δυσκολία με μακρινές εξαρτήσεις, λόγω του φαινομένου των vanishing gradients, που μειώνει την επίδραση των παλιών λέξεων. Επιπλέον, τα RNNs μπορεί να εμφανίσουν exploding gradients, όπου οι παράγωγοι μεγαλώνουν υπερβολικά και για την αντιμετώπισή τους χρησιμοποιείται η τεχνική gradient clipping, που περιορίζει το μέγεθος των παραγώγων σε κάθε βήμα οπισθοδιάδοσης (backpropagation).

4. Υποθέτουμε transformer με δύο κεφαλές προσοχής, με τους εξής projection matrices. Έστω P οι πίνακες πιθανότητας προσοχής και Z τα context representations που λαμβάνονται από τις δύο κεφαλές προσοχής. Γενικά ισχύει:

$$\begin{aligned} \text{a) } P^{(1)} &= P^{(2)}, Z^{(1)} = Z^{(2)} & W_Q^{(2)} &= 2W_Q^{(1)}, \quad W_K^{(2)} = W_K^{(1)}, \quad W_V^{(2)} = \frac{1}{2}W_V^{(1)} \\ \text{b) } P^{(1)} &= P^{(2)}, Z^{(1)} \neq Z^{(2)} & & \\ \text{c) } P^{(1)} &\neq P^{(2)}, Z^{(1)} = Z^{(2)} & & \\ \text{d) } P^{(1)} &\neq P^{(2)}, Z^{(1)} \neq Z^{(2)} & & \end{aligned}$$

Query projections: $Q = X \cdot W_Q \Rightarrow Q^2 = 2 \cdot Q^1$ } attention scores: $\frac{Q \cdot K^T}{\sqrt{d_k}} \Rightarrow \text{scores}^2 = 2 \cdot \text{scores}^1$
 Key projections: $K = X \cdot W_K \Rightarrow K^1 = K^2$
 Value projections: $V = X \cdot W_V \Rightarrow V^2 = \frac{1}{2} \cdot V^1$

$P_1 = \text{softmax}(\text{scores}^1)$ } $P^1 \neq P^2$ } $Z_1 = P_1 \cdot V_1$ } $Z_2 \neq Z_1$
 $P_2 = \text{softmax}(2 \cdot \text{scores}^1)$ } softmax ~ μη γραμ } $Z_2 = P_2 \cdot \frac{1}{2} \cdot V_1$

5. Πώς σχετίζονται τα Langevin dynamics με τα Score-Based Generative Models;
- Χρησιμοποιούνται για την άμεση αξιολόγηση της πιθανότητας κατανομής δεδομένων ενός δείγματος.
 - Ορίζουν τη συνάρτηση θορύβου που χρησιμοποιείται για την αλλοίωση των δεδομένων κατά τη διάρκεια της εκπαίδευσης.
 - Καθοδηγούν τη διαδικασία δειγματοληψίας με επαναληπτική αποθρομβοποίηση του τυχαίου θορύβου.
 - Χρησιμοποιούνται για την προ-εκπαίδευση του score network.

Τα Score-Based Generative Models δεν μαθαίνουν άμεσα την πιθανότητα $p(x)$, αλλά το score $\nabla_x \log p(x)$, δηλαδή το gradient του log-πιθανοτήτων. Τα Langevin dynamics είναι στοχαστικές διαφορικές εξισώσεις που χρησιμοποιούνται για δειγματοληψία από κατανομές, όταν γνωρίζουμε το score function.

Στα SBGM, τα Langevin dynamics καθοδηγούν τη διαδικασία δειγματοληψίας: ξεκινώντας από τυχαίο θόρυβο, εφαρμόζονται επαναληπτικά βήματα αποθρομβοποίησης ώστε να παραχθεί ένα δείγμα δεδομένων με βάση το score function.

6. Ποια είναι η βασική διαφορά μεταξύ των loss functions που χρησιμοποιούνται για τον generator και τον discriminator σε ένα GAN που εκπαιδεύεται με minimax loss;

a) Το discriminator loss χρησιμοποιεί cross-entropy, ενώ το generator loss χρησιμοποιεί το μέσο τετραγωνικό σφάλμα.

b) Και οι δύο συναρτήσεις προέρχονται από την ίδια αρχική διατύπωση minimax, αλλά το generator loss είναι απλοποιημένο λόγω της αδυναμίας του generator να επηρεάσει την πιθανότητα των πραγματικών δεδομένων.

c) Το discriminator loss ενθαρρύνει την ταξινόμηση των πραγματικών δεδομένων ως πραγματικών ενώ το generator loss στοχεύει στην ταξινόμηση των παραγόμενων δεδομένων ως ψεύτικων.

d) Το discriminator loss ελαχιστοποιεί τη διαφορά μεταξύ πραγματικών και παραγόμενων δεδομένων, ενώ το generator loss τη μεγιστοποιεί.

Το GAN δεν κάνει σύγκριση δεδομένων σε επίπεδο περιεχομένου, όλα σχετίζονται με πιθανότητες ταξινόμησης

discriminator D: binary classifier που προβλέπει πιθανότητες. $\begin{cases} D(x): \text{πιθανότητα δείγμα } x \text{ να είναι πραγματικό} \\ D(G(z)): \text{πιθανότητα παραγόμενο δείγμα } z \text{ να είναι πραγματικό} \end{cases}$

cross-entropy loss: $\begin{cases} \triangleright \text{discriminator loss: } \mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}}(\log D(x)) - \mathbb{E}_{z \sim p_z}(\log(1 - D(G(z)))) \\ \triangleright \text{generator loss: } \mathcal{L}_G = -\mathbb{E}_{z \sim p_z}(\log D(G(z))) \end{cases}$

ο generator G προσπαθεί να παραπλανήσει τον discriminator D (παραγόμενα δεδομένα να ταξινομηθούν ως πραγματικά)

ο discriminator D προσπαθεί να ταξινομήσει $\begin{cases} \text{πραγματικά δείγματα ως πραγματικά} \sim \log D(x) \\ \text{παραγόμενα δείγματα ως ψεύτικα} \sim \log(1 - D(G(z))) \end{cases}$

7. Ποια από τις παρακάτω στρατηγικές προσδιορισμού βέλτιστης πολιτικής για έναν πράκτορα αυτοεπισχυόμενης μάθησης (reinforcement learning) συγκλίνει εγγυημένα σε βέλτιστες τιμές;

a) Η επανάληψη τιμών (value iteration)

b) Η επανάληψη πολιτικών (policy iteration)

c) Και οι δύο

d) Καμία από τις δύο

Value Iteration (VI):

υπολογίζει $V^*(s)$ μέσω επαναληπτικής εφαρμογής Bellman optimality equation: $V_{k+1}(s) = \max_a \sum_s P(s'|s,a) (R(s,a,s') + \gamma V_k(s'))$
μετά τη σύγκλιση του V προκύπτει βέλτιστη πολιτική: $\pi^*(s) = \arg \max_a \sum_s P(s'|s,a) (R(s,a,s') + \gamma V^*(s'))$

Policy Iteration (PI): $\sim$ επανάληψη μέχρι σταθεροποίηση

policy evaluation: υπολογισμός $V^\pi(s)$ για τρέχουσα πολιτική

policy improvement: παραγωγή καλύτερης πολιτικής: $\pi_{\text{new}}(s) = \arg \max_a \sum_s P(s'|s,a) (R(s,a,s') + \gamma V^\pi(s'))$

Εγγυημένη σύγκλιση όταν έχουμε: ① πεπεσμένο χώρο καταστάσεων/ενεργειών ② πλήρη γνώση μοντέλου MDP ③ κατάλληλο $\gamma \in (0,1)$

8. Ποιο πρόβλημα του απλού Q-learning αλγορίθμου αντιμετωπίζει ο DQN;

a) Την αργή σύγκλιση στις βέλτιστες τιμές του πίνακα Q.

b) Την αστάθεια του απλού Q-learning αλγορίθμου όταν εφαρμόζεται σε δίκτυα που βασίζονται στην εκτίμηση πολιτικών.

c) Την εμφάνιση φαινομένων ταλάντωσης ή αποκλίσεων όταν εφαρμόζεται στα νευρωνικά δίκτυα.

d) Κανένα από τα παραπάνω.

Το Q-Learning είναι ένας αλγόριθμος ενισχυμένης μάθησης που υπολογίζει τις τιμές δράσης $Q(s,a)$, αποθηκεύοντάς τις σε πίνακα. Όταν όμως χρησιμοποιείται σε περιβάλλοντα με μεγάλο ή συνεχές χώρο καταστάσεων, ο πίνακας αντικαθίσταται από ένα νευρωνικό δίκτυο που προσεγγίζει τη συνάρτηση Q.

Q-learning με νευρωνικά μπορεί να προκαλέσει $\begin{cases} \text{αστάθεια στην εκπαίδευση} \\ \text{φαινόμενα ταλάντωσης ή αποκλίσεις} \end{cases}$

Το Deep Q-Network (DQN) λύνει αυτά τα προβλήματα με δύο βασικές τεχνικές:

1. Experience Replay: Αποθήκευση παλαιών εμπειριών και τυχαία δειγματοληψία, ώστε να σπάσει η χρονική συσχέτιση των δεδομένων.

2. Target Network: Ένα δεύτερο, "παγωμένο" νευρωνικό δίκτυο που παρέχει πιο σταθερούς στόχους κατά την ενημέρωση των Q-τιμών.

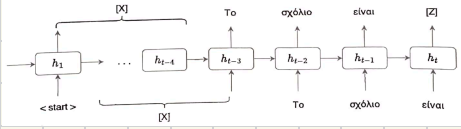
Open-ended Questions

Θέμα 1 (1.8 μονάδες)

Υποθέστε ότι έχουμε ένα μοντέλο ταξινόμησης κειμένου βασισμένο σε autoregressive RNN generation, που ταξινομεί σχόλια ως προς το toxicity ενός κειμένου. Για την ταξινόμηση ενός σχολίου χρησιμοποιούμε prompting, με το template: [X] Το σχόλιο είναι [Y], όπου το [X] γεμίζει με το αρχικό σχόλιο προς ταξινόμηση και το [Y] με την απόφαση ως προς το περιεχόμενο του σχολίου. Έστω t το βήμα που αντιστοιχεί στην παραγωγή του [Y] και $w_1, \dots, w_t$ οι λέξεις που έχουν παραχθεί συνολικά με βάση το παραπάνω prompt template. Η απόφαση για το αν είναι αποδεκτό το σχόλιο δίνεται από:

$$\hat{y} = \begin{cases} +1 & \text{if } P(w_t = \text{"αποδεκτό"} \mid w_1, \dots, w_{t-1}) \geq P(w_t = \text{"τοξικό"} \mid w_1, \dots, w_{t-1}) \\ -1 & \text{otherwise} \end{cases} \quad W_{hx} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix}, \quad W_{hh} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

1.1 Ας υποθέσουμε ότι το σχόλιο προς ταξινόμηση είναι "All men must die.". Όλα τα biases είναι 0. Η αρχική κατάσταση του RNN είναι $h_0 = [0, 0]^T$ και η συνάρτηση ενεργοποίησης για το RNN είναι η ReLU. Ποια θα είναι η τελική πρόβλεψη; Δείξτε όλους τους υπολογισμούς.



word embeddings:

$$\begin{aligned} x_{\text{start}} &= [0, 0, 0]^T, & x_{\text{All}} &= [0, 0, -1]^T, & x_{\text{men}} &= [1, 0, 0]^T, & x_{\text{must}} &= [0, 0, 0]^T, \\ x_{\text{die.}} &= [0, -5, 0]^T, & x_{\text{To}} &= [0, 0, 0]^T, & x_{\text{σχόλιο}} &= [-2, 0, 0]^T, & x_{\text{είναι}} &= [0, 0, 0]^T, \\ x_{\text{αποδεκτό}} &= [1, 2, 3]^T, & x_{\text{τοξικό}} &= [-1, -2, -3]^T \end{aligned}$$

$$W_{yh} = \begin{bmatrix} 0 & 0 & \#(\text{start}) \\ 1 & 2 & \#(\text{All}) \\ 3 & 4 & \#(\text{men}) \\ 5 & 6 & \#(\text{must}) \\ 7 & 8 & \#(\text{die.}) \\ 9 & 10 & \#(\text{To}) \\ 11 & 12 & \#(\text{σχόλιο}) \\ 13 & 14 & \#(\text{είναι}) \\ 1 & -1 & \#(\text{αποδεκτό}) \\ -1 & 1 & \#(\text{τοξικό}) \end{bmatrix}$$

input: <start> All men must die. Το σχόλιο είναι

activation function: ReLU $\Rightarrow h_t = \text{ReLU}(a_t)$, $a_t = W_{hx} \cdot x_t + W_{hh} \cdot h_{t-1} = W_{hx} \cdot x_t + h_{t-1}$

► $t=1$: <start> $x_1 = [0, 0, 0]^T$, $h_0 = [0, 0]^T$

$$a_1 = W_{hx} \cdot x_1 + h_0 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot 0 + (-1) \cdot 0 + (-1) \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow h_1 = \text{ReLU}(a_1) = [0, 0]^T$$

► $t=2$: All $x_2 = [0, 0, -1]^T$, $h_1 = [0, 0]^T$

$$a_2 = W_{hx} \cdot x_2 + h_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot 0 + 0 \cdot (-1) \\ 0 \cdot 0 + (-1) \cdot 0 + (-1) \cdot (-1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow h_2 = \text{ReLU}(a_2) = [0, 1]^T$$

► $t=3$: men: $x_3 = [1, 0, 0]^T$, $h_2 = [0, 1]^T$

$$a_3 = W_{hx} \cdot x_3 + h_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot 1 + (-1) \cdot 0 + (-1) \cdot 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow h_3 = \text{ReLU}(a_3) = [1, 1]^T$$

► $t=4$: must: $x_4 = [0, 0, 0]^T$, $h_3 = [1, 1]^T$

$$a_4 = W_{hx} \cdot x_4 + h_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot 0 + (-1) \cdot 0 + (-1) \cdot 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow h_4 = \text{ReLU}(a_4) = [1, 1]^T$$

► $t=5$: die.: $x_5 = [0, -5, 0]^T$, $h_4 = [1, 1]^T$

$$a_5 = W_{hx} \cdot x_5 + h_4 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot (-5) + 0 \cdot 0 \\ 0 \cdot 0 + (-1) \cdot (-5) + (-1) \cdot 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 5 \end{bmatrix} \Rightarrow h_5 = \text{ReLU}(a_5) = [0, 6]^T$$

► $t=6$: Το: $x_6 = [0, 0, 0]^T$, $h_5 = [0, 6]^T$

$$a_6 = W_{hx} \cdot x_6 + h_5 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot 0 + (-1) \cdot 0 + (-1) \cdot 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix} \Rightarrow h_6 = \text{ReLU}(a_6) = [0, 6]^T$$

► $t=7$: σχόλιο: $x_7 = [-2, 0, 0]^T$, $h_6 = [0, 6]^T$

$$a_7 = W_{hx} \cdot x_7 + h_6 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \cdot (-2) + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot (-2) + (-1) \cdot 0 + (-1) \cdot 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} -2 \\ 6 \end{bmatrix} \Rightarrow h_7 = \text{ReLU}(a_7) = [0, 6]^T$$

► $t=8$: είναι: $x_8 = [0, 0, 0]^T$, $h_7 = [0, 6]^T$

$$a_8 = W_{hx} \cdot x_8 + h_7 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 \\ 0 \cdot 0 + (-1) \cdot 0 + (-1) \cdot 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix} \Rightarrow h_8 = \text{ReLU}(a_8) = [0, 6]^T$$

► $t=9$: output layer: $h_8 = [0, 6]^T$

$$\hat{y} = W_{yh} \cdot h_8 = \begin{bmatrix} 0 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 1 & -1 \\ 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 6 \end{bmatrix} = \begin{bmatrix} 0 \cdot 12 + 1 \cdot 24 + 3 \cdot 36 + 5 \cdot 48 + 7 \cdot 60 + 9 \cdot 84 - 6 \cdot 6 \\ 0 \cdot 2 + 2 \cdot 4 + 4 \cdot 6 + 6 \cdot 8 + 8 \cdot 10 + 10 \cdot 12 - 1 \cdot 6 + 1 \cdot 6 \end{bmatrix} = \begin{bmatrix} 600 - 36 \\ 120 - 6 + 6 \end{bmatrix} = \begin{bmatrix} 564 \\ 120 \end{bmatrix}$$

$$\Rightarrow \begin{cases} P(\text{αποδεκτό}) = 0 \\ P(\text{τοξικό}) > P(\text{αποδεκτό}) \end{cases} \Rightarrow \hat{y} = -1 \sim \text{Το σχόλιο είναι τοξικό}$$

$$\text{ή κατευθείαν: } \bullet \text{αποδεκτό} \sim W_{\text{tox}} = [1, -1] \Rightarrow a_{\text{tox}} = [1, -1] \cdot [0, 6]^T = -6 \Rightarrow P(\text{αποδεκτό}) = -6$$

$$\bullet \text{τοξικό} \sim W_{\text{tox}} = [-1, 1] \Rightarrow a_{\text{tox}} = [-1, 1] \cdot [0, 6]^T = 6 \Rightarrow P(\text{τοξικό}) = 6$$

1.2 Τώρα ας υποθέσουμε ότι αντικαθιστούμε το RNN και χρησιμοποιούμε ως γλωσσικό μοντέλο έναν (πολύ απλό) transformer (στην πραγματικότητα, μόνο ένα self-attention layer με μία μόνο κεφαλή προσοχής, χωρίς feed-forward layers και χωρίς residual συνδέσεις). Χρησιμοποιούμε scaled dot product attention για τον υπολογισμό των πιθανοτήτων προσοχής. Το linear layer εξόδου χρησιμοποιεί τον ίδιο πίνακα W_{yh} όπως παραπάνω. Ποια θα είναι η πρόβλεψη; Υποδείξτε όλους τους υπολογισμούς. (Υπόδειξη: παρατηρήστε ότι δεν χρειάζεται να υπολογίσετε τον πλήρη πίνακα πιθανοτήτων προσοχής, μόνο τις πιθανότητες προσοχής που σχετίζονται με την τελευταία λέξη.)

projection matrices:

$$W_Q = W_K = W_V = \begin{bmatrix} 0.1 & 0 \\ 0 & -0.1 \\ 0 & 0.1 \end{bmatrix}$$

position embedding: $p_1 = [0, 0, 0]^T$, $p_2 = [0, 0, 1]^T$, $p_3 = [0, 0, 2]^T$, ..., $p_t = [0, 0, t-1]^T$

input tokens: $z_t = x_t + p_t$

$$z_1 = x_1 + p_1 = [0 \ 0 \ 0]^T + [0 \ 0 \ 0]^T = [0 \ 0 \ 0]^T$$

$$z_2 = x_2 + p_2 = [0 \ 0 \ -1]^T + [0 \ 0 \ 1]^T = [0 \ 0 \ 0]^T$$

$$z_3 = x_3 + p_3 = [1 \ 0 \ 0]^T + [0 \ 0 \ 2]^T = [1 \ 0 \ 2]^T$$

$$z_4 = x_4 + p_4 = [0 \ 0 \ 0]^T + [0 \ 0 \ 3]^T = [0 \ 0 \ 3]^T$$

$$z_5 = x_5 + p_5 = [0 \ -5 \ 0]^T + [0 \ 0 \ 4]^T = [0 \ -5 \ 4]^T$$

$$z_6 = x_6 + p_6 = [0 \ 0 \ 0]^T + [0 \ 0 \ 5]^T = [0 \ 0 \ 5]^T$$

$$z_7 = x_7 + p_7 = [-2 \ 0 \ 0]^T + [0 \ 0 \ 6]^T = [-2 \ 0 \ 6]^T$$

$$z_8 = x_8 + p_8 = [0 \ 0 \ 0]^T + [0 \ 0 \ 7]^T = [0 \ 0 \ 7]^T$$

$$\Rightarrow z = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}^T$$

$$\text{Query vector: } Q = z \cdot W_Q$$

$$\text{Key vector: } K = z \cdot W_K$$

$$\text{Value vector: } V = z \cdot W_V$$

$$W_Q = W_K = W_V$$

$$Q = K = V = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}^T \cdot \begin{bmatrix} 0.1 & 0 \\ 0 & -0.1 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.1 & 0 & 0 & 0 & -0.2 & 0 \\ 0 & 0 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 \end{bmatrix}^T$$

Μας ενδιαφέρει η τελευταία λέξη: $q_8 = [0 \ 0.7]^T$, $d_k = 2 \Rightarrow$ attention score $S_{8,j} = q_8^T k_j / \sqrt{d_k}$ και attention weight $A_{8,j} = \text{softmax}(S_{8,j})$

$$S_{8,1} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0) = 0$$

$$S_{8,2} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0) = 0$$

$$S_{8,3} = \frac{1}{\sqrt{2}} (0 \cdot 0.1 + 0.7 \cdot 0.2) = 0.09899$$

$$S_{8,4} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0.3) = 0.24849$$

$$S_{8,5} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0.4) = 0.44547$$

$$S_{8,6} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0.5) = 0.24748$$

$$S_{8,7} = \frac{1}{\sqrt{2}} (0 \cdot (-0.2) + 0.7 \cdot 0.6) = 0.29698$$

$$S_{8,8} = \frac{1}{\sqrt{2}} (0 \cdot 0 + 0.7 \cdot 0.7) = 0.34648$$

$$\sim \frac{e^{S_{8,j}}}{\sum e^{S_{8,j}}}$$

$$\sim 1$$

$$\sim 1$$

$$\sim 1.104$$

$$\sim 1.160$$

$$\sim 1.561$$

$$\sim 1.281$$

$$\sim 1.346$$

$$\sim 1.414$$

$$Z = 9.866$$

context vector:

$$C = A^T \cdot V =$$

$$= \begin{bmatrix} A_{8,1} \cdot 0 + A_{8,2} \cdot 0 + A_{8,3} \cdot 0.1 + A_{8,4} \cdot 0 + A_{8,5} \cdot 0 + A_{8,6} \cdot 0 + A_{8,7} \cdot (-0.2) + A_{8,8} \cdot 0 \\ A_{8,1} \cdot 0 + A_{8,2} \cdot 0 + A_{8,3} \cdot 0.2 + A_{8,4} \cdot 0.3 + A_{8,5} \cdot 0.4 + A_{8,6} \cdot 0.5 + A_{8,7} \cdot 0.6 + A_{8,8} \cdot 0.7 \end{bmatrix}$$

$$= \begin{bmatrix} -0.01609 \\ 0.44719 \end{bmatrix}$$

$$\triangleright \text{ανοδεκτό} \Rightarrow w_{acc} = [1 \ -1] \Rightarrow \text{logit}_{acc} = w_{acc}^T C = 1 \cdot (-0.01609) + (-1) \cdot 0.44719 = -0.46328 = P(\text{ανοδεκτό})$$

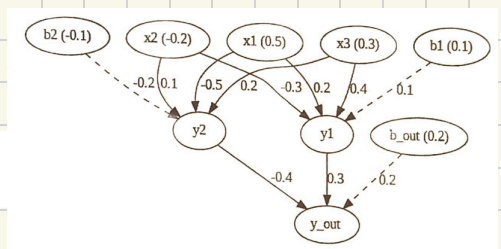
$$\triangleright \text{τοξικό} \Rightarrow w_{tox} = [-1 \ 1] \Rightarrow \text{logit}_{tox} = w_{tox}^T C = (-1) \cdot (-0.01609) + 1 \cdot 0.44719 = 0.46328 = P(\text{τοξικό})$$

$$\Rightarrow P(\text{τοξικό}) > P(\text{ανοδεκτό}) \Rightarrow \hat{y} = -1 \Rightarrow \text{Το σκόπιο είναι τοξικό.}$$

Θέμα 2 (1.6 μονάδες)

Δίνεται το Multi-Layer Perceptron του Σχήματος.

2.1 Θεωρώντας ότι οι συναρτήσεις ενεργοποίησης είναι η sigmoid για το κρυφό επίπεδο και η γραμμική (linear) για το επίπεδο εξόδου, να υπολογίσετε την έξοδο y_{out} .



hidden layer:

$$\triangleright \text{neuron 1: input: } \tilde{x}_1 = [x_1 \ x_2 \ x_3 \ b_1]^T = [0.5 \ -0.2 \ 0.3 \ 0.1]^T$$

$$\text{weights: } \tilde{w}_1 = [w_{11} \ w_{12} \ w_{13} \ w_{1b}] = [0.2 \ -0.3 \ 0.4 \ 0.1]$$

$$\left. \begin{array}{l} \text{pre-activation: } z_1 = \tilde{x}_1 \cdot \tilde{w}_1 = 0.2 \cdot 0.5 + (-0.3) \cdot (-0.2) + 0.4 \cdot 0.3 + 0.1 \cdot 0.1 = 0.29 \\ \text{post-activation: } y_1 = \sigma(z_1) = \frac{1}{1 + e^{-0.29}} = 0.5720 \end{array} \right\}$$

$$\triangleright \text{neuron 2: input: } \tilde{x}_2 = [x_1 \ x_2 \ x_3 \ b_2]^T = [0.5 \ -0.2 \ 0.3 \ -0.1]^T$$

$$\text{weights: } \tilde{w}_2 = [w_{21} \ w_{22} \ w_{23} \ w_{2b}] = [-0.5 \ 0.1 \ 0.2 \ -0.2]$$

$$\left. \begin{array}{l} \text{pre-activation: } z_2 = \tilde{x}_2 \cdot \tilde{w}_2 = (-0.5) \cdot 0.5 + 0.1 \cdot (-0.2) + 0.2 \cdot 0.3 + (-0.2) \cdot (-0.1) = -0.19 \\ \text{post-activation: } y_2 = \sigma(z_2) = \frac{1}{1 + e^{-0.19}} = 0.4526 \end{array} \right\}$$

output layer:

$$\text{input: } \tilde{x}_{out} = [y_1 \ y_2 \ b_{out}]^T = [0.5720 \ 0.4526 \ 0.2]^T$$

$$\text{weights: } \tilde{w}_{out} = [w_{out1} \ w_{out2} \ w_{outb}] = [0.3 \ -0.4 \ 0.2]$$

$$\left. \begin{array}{l} \text{output: } y_{out} = \tilde{x}_{out} \cdot \tilde{w}_{out} = 0.3 \cdot 0.5720 + (-0.4) \cdot 0.4526 + 0.2 \cdot 0.2 = 0.0306 \end{array} \right\}$$

2.2 Αν η επιθυμητή έξοδος είναι $y_{target} = 0.21$, να υπολογιστεί το loss function χωρίς regularization.

$$MSE: L = \frac{1}{2} \cdot (y_{out} - y_{target})^2 = \frac{1}{2} \cdot (0.0306 - 0.21)^2 = 0.0161$$

2.3 Χρησιμοποιήστε το L2 regularization (Ridge regularization) με $\lambda=0.01$ και υπολογίστε το συνολικό loss function με L2 regularization.

$$L_{MSE} = \frac{1}{2} (y_{out} - y_{target})^2 = 0.0161$$

$$L_{reg} = \frac{\lambda}{2} \sum w_i^2 = 0.01/2 \cdot [0.2^2 + (-0.3)^2 + (0.4)^2 + (-0.5)^2 + 0.1^2 + 0.2^2 + 0.3^2 + (-0.4)^2] = 0.0042$$

$$L_{total} = L_{MSE} + L_{reg} = 0.0203$$

2.4 Πώς επηρεάζει η επιλογή της συνάρτησης ενεργοποίησης τη συμπεριφορά και την απόδοση ενός MLP; Συγκρίνετε τη συνάρτηση σιγμοειδούς (sigmoid), την υπερβολική εφαιπτομένη (tanh) και τη Rectified Linear Unit (ReLU).

Η επιλογή της συνάρτησης ενεργοποίησης επηρεάζει καθοριστικά τη συμπεριφορά και την απόδοση ενός MLP, καθώς επηρεάζει τη μη γραμμικότητα του μοντέλου, τη διάδοση του σφάλματος και την ταχύτητα εκπαίδευσης.

- Η sigmoid δίνει εξόδους στο διάστημα (0, 1) και χρησιμοποιείται συχνά σε προβλήματα ταξινόμησης. Παρουσιάζει όμως πρόβλημα vanishing gradient για ακραίες τιμές, δεν είναι zero-centered και γενικά οδηγεί σε αργή σύγκλιση.
- Η tanh έχει έξοδο στο (-1, 1), είναι zero-centered και συνεπώς επιτρέπει ταχύτερη εκπαίδευση σε σχέση με τη sigmoid. Παρ' όλα αυτά, και αυτή εμφανίζει vanishing gradients σε μεγάλα $|z|$.
- Η ReLU είναι σήμερα η πιο διαδεδομένη στα κρυφά επίπεδα. Είναι υπολογιστικά απλή, επιτρέπει γρήγορη σύγκλιση και περιορίζει το πρόβλημα vanishing gradient. Το βασικό μειονέκτημά της είναι οι "dead neurons" όταν $z < 0$.

Συμπερασματικά, η ReLU αποτελεί την καλύτερη επιλογή για τα κρυφά επίπεδα στα περισσότερα προβλήματα, ενώ sigmoid και tanh χρησιμοποιούνται κυρίως σε εξόδους, ανάλογα με το εύρος τιμών που απαιτεί το πρόβλημα.

Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}} \in (0,1)$, όχι zero-centered, παρουσία vanishing gradient, αργή σύγκλιση

tanh: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \in (-1,1)$, zero-centered $\rightarrow$ καλύτερη από sigmoid, έχει vanishing gradient

ReLU: $\text{ReLU}(z) = \max(0, z) \in [0, +\infty)$, χρήση εκπαίδευση, όχι vanishing gradient για $z > 0$, κίνδυνος για dead neurons για $z < 0$

Θέμα 3 (1.2 μονάδες)

Για τον γράφο του παρακάτω σχήματος, να υπολογιστεί ο συντελεστής συσταδοποίησης και η ενδιάμεσότητα κάθε κόμβου. Με βάση τα μέτρα αυτά, τι συμπεράσματα θα μπορούσατε να βγάλετε σχετικά με τη σημαντικότητα των κόμβων;

κόμβος		#γείτον	ακμές	#ακμές	#δυνατά	
v	γείτον	deg(v)	γείτον	γείτον	γείτον	Cv
A	C	1	-	0	-	0
B	C,D	2	CD	1	1	1/1=1
C	A,B,D	3	BD	1	3	1/3=0.333
D	B,C,E,H	4	BC	1	6	1/6=0.167
E	D,F,G	3	-	0	3	0/3=0
F	E	1	-	0	-	0
G	E	1	-	0	-	0
H	D	1	-	0	-	0

ενδιάμεσότητα betweenness centrality:

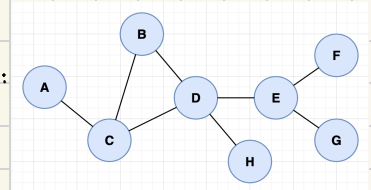
$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

σ_{st} : αριθμός shortest path μεταξύ s-t

$\sigma_{st}(v)$: ποσα από αυτά περνάει από v

συντελεστής συσταδοποίησης clustering coefficient:

$$C_v = \frac{\text{αριθμός ακμών μεταξύ γειτονικών κόμβων}}{\text{συνολικός αριθμός πιθανών ακμών μεταξύ γειτονικών κόμβων}} = \frac{\text{αριθμός ακμών μεταξύ γειτ}}{\text{deg(v)(deg(v)-1)/2}}$$



s-t	shortest paths	σ_{st}	ενδιάμεσοι
A-B	ACB	1	C(1)
A-D	ACD	1	C(1)
A-H	ACDH	1	C(1), D(1)
A-E	ACDE	1	C(1), D(1)
A-F	ACDEF	1	C(1), D(1), E(1)
A-G	ACDEG	1	C(1), D(1), E(1)
C-E	CDE	1	D(1)
C-H	CDH	1	D(1)
C-F	CDEF	1	D(1), E(1)
C-G	CDEG	1	D(1), E(1)
B-E	BDE	1	D(1)
B-H	BDH	1	D(1)
B-F	BDEF	1	D(1), E(1)
B-G	BDEG	1	D(1), E(1)
D-F	DEF	1	E(1)
D-G	DEG	1	E(1)

s-t	shortest paths	σ_{st}	ενδιάμεσοι
H-E	HDE	1	D(1)
H-F	HDEF	1	D(1), E(1)
H-G	HDEG	1	D(1), E(1)
F-G	FEH	1	E(1)

$$\begin{aligned} & \triangleright C_B(A) = 0 \\ & \triangleright C_B(B) = 0 \\ & \triangleright C_B(C) = 6 \\ & \triangleright C_B(D) = 15 \\ & \triangleright C_B(E) = 11 \\ & \triangleright C_B(F) = 0 \\ & \triangleright C_B(G) = 0 \\ & \triangleright C_B(H) = 0 \end{aligned}$$

Κόμβος	Clustering	Ενδιάμεσότητα	Συμπεράσματα για σημασία κόμβου
A	0	0	Περιφερειακός - χωρίς τοπική ή διαμεσολαβητική σημασία
B	1	0	Τοπικά σημαντικός - πλήρης συσχέτιση με γείτονες, αλλά χωρίς ρόλο γέφυρας
C	0.333	6	Μέτρια τοπική συνοχή - σημαντικός ως διάμεσος σε paths A→...
D	0.167	15	Κρίσιμος κόμβος - χαμηλή συσταδοποίηση αλλά μεγάλη διαμεσολάβηση, συνδέει τμήματα
E	0	11	Διαμεσολαβητής μεταξύ τερματικών - σημαντικός σε πολλά μονοπάτια
F	0	0	Τελικός κόμβος - χωρίς συμμετοχή σε clusters ή paths
G	0	0	Όπως και ο F - τελικός περιφερειακός
H	0	0	Περιφερειακός κόμβος - δεν διαμεσολαβεί, δεν ανήκει σε σύνολα