Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εαρινό Εξάμηνο 2023-2024

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΒΑΘΙΑ ΜΑΘΗΣΗ

Λύσεις Θεμάτων Κ23

Ιωάννης (Χουάν) Τσαντήλας
03120883

Contents

Θέμα Πολλαπλής Επιλογής

Ερώτημα 1[1]

Ποιες από τις παρακάτω τεχνικές μπορούν να χρησιμοποιηθούν για τη μείωση της υπερπροσαρμογής (overfitting);

a) Dropout και Χρήση Adam αντί Stochastic Gradient Descent.
b) Χρήση Adam αντί Stochastic Gradient Descent και Επαύξηση Δεδομένων.
c) Επαύξηση Δεδομένων και Dropout.
d) Κανένα από τα παραπάνω.

Απάντηση: Γ. *Δεν το βρήκα κάπου συγκεκριμένα.*

Ωστόσο, από το εργαστήριο, γνωρίζουμε πως τα dropout και data augmentation βοηθούν στη μείωση του overfitting. Για τα Adam και Stochastic βρήκα τα εξής:

- https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/: Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.
- https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/: SGD is a variant of the Gradient Descent algorithm that is used for optimizing machine learning models. It addresses the computational inefficiency of traditional Gradient Descent methods when dealing with large datasets in machine learning projects.

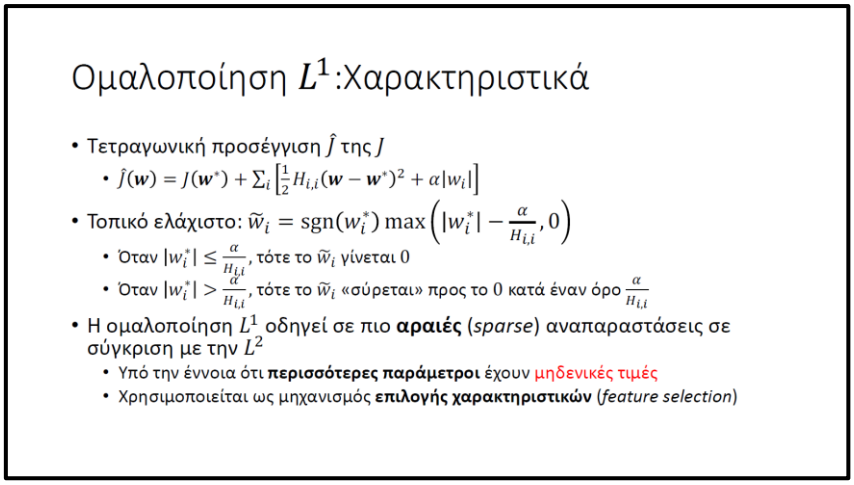
Ερώτημα 2

Ένα μοντέλο βαθιάς μάθησης που αναπτύξατε πετυχαίνει εξαιρετικά υψηλή ορθότητα (accuracy) (90%) στο σύνολο εκπαίδευσης (training set), αλλά σημαντικά χαμηλότερη (75%) στο σύνολο δοκιμής (test set). Για να αντιμετωπίσετε το πρόβλημα, αποφασίζετε να εφαρμόσετε ομαλοποίηση (regularization) τύπου L^1 ή L^2. Έχετε όμως βάσιμες υποψίες ότι κάποια από τα δεδομένα σας έχουν επισημειωθεί με εσφαλμένες ετικέτες (labels). Ποιον τύπο ομαλοποίησης θα προτιμήσετε να χρησιμοποιήσετε;

a) L^1
b) L^2
c) Δεν έχει σημασία, τόσο η L^1 όσο και η L^2 είναι εξίσου κατάλληλες για τη συγκεκριμένη περίπτωση.
d) Δεν μπορεί να αντιμετωπιστεί μέσω ομαλοποίησης το εν λόγω πρόβλημα.

Απάντηση: Α. Σετ 4, Διαφάνεια 23 (;). Αυτό που καταλαβαίνω είναι πως οδηγεί τα βάρη των λιγότερο σημαντικών χαρακτηριστικών στο 0, άρα μπορεί να μειώσει το αντίκτυπο που έχουν αυτά με λάθος ετικέτες.

[1] *Προσπάθησα γενικά να βρω την αντίστοιχη διαφάνεια που να εξηγεί την απάντηση μου. Thing is, έχουν τελείως γτπ διαφάνειες που μπορεί να αναφέρονται στο ερώτημα, π.χ. Ερώτημα 2, αλλά να μην εξηγούν explicitly κάποια από τις επιλογές, επομένως με βοήθεια ChatGPT-4o (June 2024) και του ίντερνετ προσπάθησα να τα απαντήσω. Εάν βρείτε κάπου ότι αναφέρεται πιο συγκεκριμένα, επικοινωνήστε μαζί μου ώστε να ανεβάσω διόρθωση.*



Ερώτημα 3

Κάνετε benchmarking στους χρόνους εκτέλεσης συχνά χρησιμοποιούμενων επιπέδων σε αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων (CNN). Ποιο από τα παρακάτω επίπεδα αναμένετε να έχει τον μικρότερο χρόνο εκτέλεσης (σε floating point operations) για δεδομένη είσοδο;

a) CONV layer (πράξη συνέλιξης και πρόσθεση όρου bias).
b) MAX pooling layer.
c) Average pooling layer.
d) Δεν έχουμε αρκετά δεδομένα για να απαντήσουμε.

Απάντηση: B. *Δεν το βρήκα κάπου συγκεκριμένα*.

Σε αυτό το paper: https://proceedings.mlr.press/v51/lee16a.html, σελίδα 6, figure 2, φαίνεται ότι τα max και ave pooling δεν έχουν σημαντικές διαφορές όσο αφορά το performance τους (το αναφέρουν και οι ίδιοι στη σελίδα 3: *The results indicate that, on the evaluation dataset, there are regimes in which either max pooling or average pooling demonstrates better performance than the other*).

Παρόλα αυτά, η πράξη συνέλιξης είναι σίγουρα πιο χρονοβόρα από ό,τι το max ή το average. Εν συνέχεια, το max κρατάει κάθε φορά ένα στοιχείο και (λογικά) είναι λιγότερο απαιτητικό (υπολογιστικά) από ό,τι το average.

Ερώτημα 4

Τι από τα παρακάτω ισχύει για τους auto-encoders:

a) Οι auto-encoders με ReLU activations και squared loss είναι ισοδύναμοι με PCA.
b) Οι auto-encoders με linear activations και squared loss είναι ισοδύναμοι με PCA.
c) Οι auto-encoders με linear activations είναι ισοδύναμοι με PCA.
d) Οι auto-encoders με ReLU activations είναι ισοδύναμοι με PCA.

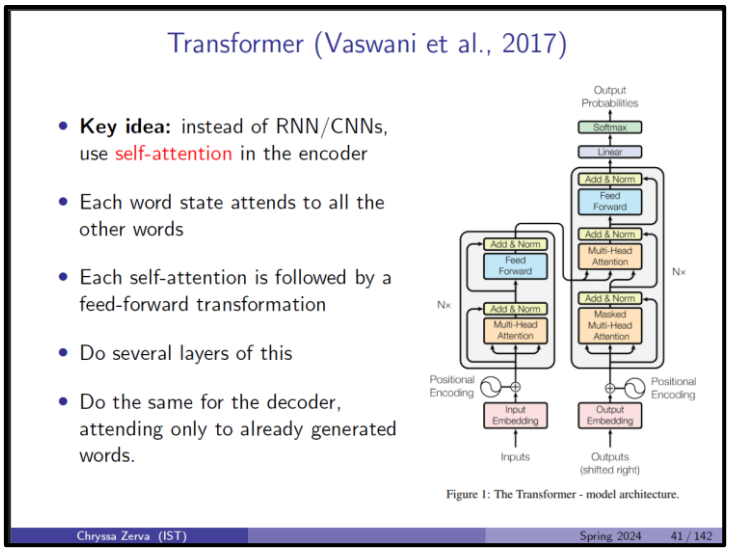
B. Οι αυτό-κωδικοποιητές με γραμμικές συναρτήσεις ενεργοποίησης και τετραγωνικές απώλειες ελαχιστοποιούν τον ίδιο στόχο με την PCA, την καταγραφή των κύριων συνιστωσών των δεδομένων εισόδου.

Ερώτημα 5

Ο κωδικοποιητής (encoder) ενός transformer χρησιμοποιεί:

a) Context-attention.
b) Multi-head cross-attention.
c) Multi-head self-attention.
d) Τίποτα από τα παραπάνω.

Απάντηση: Γ. Σετ 7, Διαφάνεια 41.

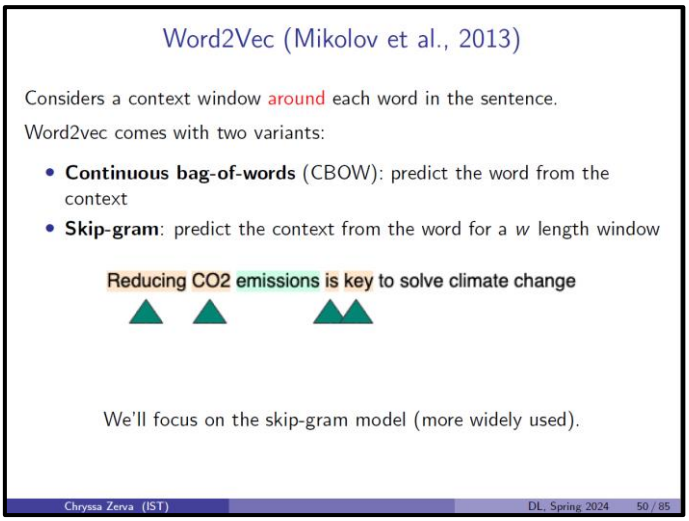


Ερώτημα 6

Ένα μοντέλο word2vec skip-gram εκπαιδεύεται με στόχο (target objective) να:

a) Προβλέπει μία λέξη με βάση ένα παράθυρο n λέξεων γύρω από αυτή.
b) Προβλέπει με βάση μία λέξη τα συμφραζόμενά της (γειτονικές λέξεις) σε παράθυρο n λέξεων.
c) Προβλέπει την απόσταση των διανυσμάτων μεταξύ γειτονικών λέξεων σε παράθυρο n.
d) Τίποτα από τα παραπάνω.

Απάντηση: Β. Σετ 2, Διαφάνεια 50.



Ερώτημα 7

Εκπαιδεύετε ένα Generative Adversarial Network (GAN) για να παράγει ρεαλιστικές εικόνες αιλουροειδών. Ωστόσο, υποπτεύεστε ότι ο generator παρουσιάζει το πρόβλημα του mode collapse. Τι από τα παρακάτω μπορεί να αποτελεί ένδειξη για το συγκεκριμένο πρόβλημα;

a) Η τιμή του κόστους (loss) του discriminator είναι υψηλή ενώ αυτή του generator παραμένει χαμηλή.
b) Ο discriminator παρουσιάζει υψηλή ορθότητα (accuracy) στις αληθινές εικόνες, αλλά χαμηλή ορθότητα στις ψεύτικες.
c) Ο generator παράγει διάφορες εικόνες άσχετες με αιλουροειδή.
d) Ο generator παράγει μόνο εικόνες με λευκές τίγρεις.

Απάντηση: Δ. *Το «mode collapse» δεν αναφέρεται πουθενά. Το GAN είναι Σετ 9, Διαφάνεια 27.*

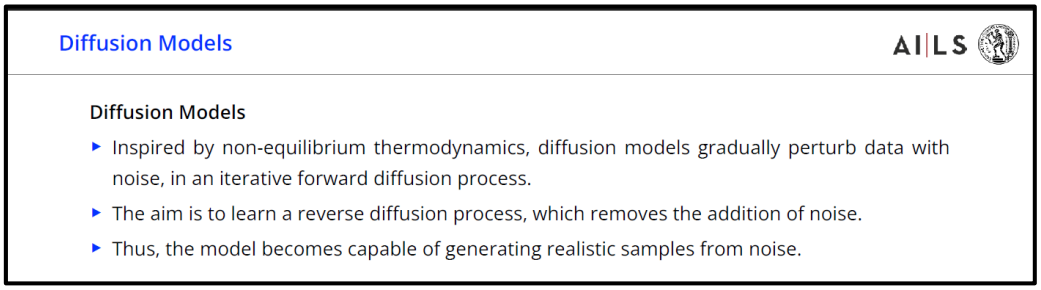
- https://spotintelligence.com/2023/10/11/mode-collapse-in-gans-explained-how-to-detect-it-practical-solutions/, «Mode collapse is a common issue in generative models, particularly in the context of generative adversarial networks (GANs) and some variants of autoencoders. It occurs when the model generates limited or repetitive outputs, failing to capture the full diversity of the data it's trained on. Instead of producing a wide range of unique samples, the generator repeatedly makes similar or identical samples.»

Ερώτημα 8

Ποια από τις παρακάτω προτάσεις για τα μοντέλα διάχυσης (diffusion models) είναι εσφαλμένη;

a) Η βασική αρχή λειτουργίας τους περιλαμβάνει μια προς-τα-εμπρός διαδικασία πρόσθεσης θορύβου και μια προς-τα-πίσω διαδικασία αφαίρεσης θορύβου.
b) Η παραγωγή ρεαλιστικών δειγμάτων πραγματοποιείται κατά κύριο λόγο κατά τη προς-τα-εμπρός διαδικασία.
c) Στην τυπική μορφή των diffusion models, η ανάστροφη (προς-τα-πίσω) διαδικασία αφαιρεί τον θόρυβο κατά κανόνα με χρήση αλυσίδων Markov.
d) Τα diffusion models είναι γενικώς πιο αργά στην παραγωγή ρεαλιστικών δειγμάτων σε σχέση με τα GANs.

Απάντηση: Β. Σετ 9, Διαφάνεια 31 (πρώτα προσθέτουν θόρυβο και έπειτα τον αφαιρούν, με Markov. Στο τέλος δηλαδή παράγεται η σωστή εικόνα). *Για το Δ ωστόσο δεν βρήκα κάτι explicitly, αλλά φαντάζομαι επειδή έχουν αυτή την πρόσθεση/αφαίρεση θορύβου είναι πιο αργά.*



Ερώτημα 9

Ποια από τις παρακάτω εργασίες δεν χρησιμοποιείται ως pretext task σε τεχνικές αυτό-επιβλεπόμενες μάθησης (self-supervised learning);

1. Αναγνώριση γεωμετρικού μ/σ εικόνας.
2. Ταξινόμηση καρέ από βίντεο στη σωστή σειρά.
3. Σημασιολογική κατάτμηση εικόνας σε επίπεδο pixel.
4. Κανένα από τα παραπάνω.

Απάντηση: Γ. Σετ 5, Διαφάνειες:

- 9: train a model to predict the rotation degree that was applied **(α)**.
- 12: train a model to predict the relationship between the patches.
- 17: predict the positions of all 9 patches.
- 20: fill in a missing piece in the image.
- 27: predict the colors of the objects in grayscale images.
- 30: predict the missing channel from the other image channels.
- 34: predict a high-resolution image that corresponds to a downsampled low-resolution image.
- 36: predict the cluster to which an image belongs.
- 38: predict whether an input image is synthetic or real, based on predicted depth, surface normal, and instance contour maps.
- 41: predict the order for a sequence of patches using contrastive learning.
- 59: predict if the frames are in the correct temporal order **(β)**.
- 62: predict the location of a patch with a moving object across frames.
- 63: predict the color of moving objects in other frames.

Θέμα 1

Ερώτημα 1

Έστω συνελικτικό νευρωνικό δίκτυο (Convolutional Neural Network – CNN) που αποτελείται από τα επίπεδα που φαίνονται στην αριστερή στήλη του παρακάτω πίνακα. Συμπληρώστε τον πίνακα με τις διαστάσεις κάθε επιπέδου και το πλήθος των παραμέτρων του. Γράψτε τις διαστάσεις στη μορφή $W \times H \times C$, όπου W, H, C είναι το πλάτος (width), ύψος (height) και βάθος (channel). Όπου χρειάζεται, υποθέστε ότι padding και stride είναι ίσα με 1, εκτός αν προσδιορίζεται κάτι διαφορετικό. Επεξήγηση συμβολισμών:

- CONVx-N: συνελικτικό επίπεδο με N φίλτρα (πυρήνες) ύψους και πλάτους ίσων με x.
- POOL-n: max pooling επίπεδο $n \times n$ με $stride = n$ και $padding = 0$.
- FC-N: πλήρως συνδεδεμένο επίπεδο N νευρώνων.

Επίπεδο	**Διαστάσεις**	**Πλήθος Παραμέτρων**
Είσοδος	$32 \times 32 \times 3$	0
CONV3-8	$32 \times 32 \times 8$	$3 \cdot 3 \cdot 3 \cdot 8 + 8 = 224$
ReLU	$32 \times 32 \times 8$	0
POOL-2	$16 \times 16 \times 8$	0
CONV3-16	$16 \times 16 \times 16$	$3 \cdot 3 \cdot 8 \cdot 16 + 16 = 1168$
ReLU	$16 \times 16 \times 16$	0
FLATTEN	4096	0
FC-10	10	$4096 \cdot 10 + 10 = 40{,}970$

Λύση

Παραθέτω έναν πιο λεπτομερή υπολογισμό.

1. *Layer εισόδου*
    a. **Διαστάσεις**: $32 \times 32 \times 3$ (δεδομένο).
    b. **Παράμετροι**: 0 (το layer εισόδου δεν έχει παραμέτρους).
2. *CONV3-8*
    a. **Διαστάσεις**: $32 \times 32 \times 8$. Δεδομένου ότι το padding και το stride είναι 1, το μέγεθος εξόδου παραμένει το ίδιο με το μέγεθος εισόδου.
    b. **Παράμετροι**: $3 \cdot 3 \cdot 3 \cdot 8 + 8 = 224$. Το μέγεθος φίλτρου είναι 3×3, 3 κανάλια εισόδου, 8 φίλτρα, συν 8 όροι πόλωσης.
3. *ReLU*
    a. **Διαστάσεις**: $32 \times 32 \times 8$.
    b. **Παράμετροι**: 0 (η συνάρτηση ενεργοποίησης δεν έχει παραμέτρους).
4. *POOL-2*
    a. **Διαστάσεις**: $32 \times 32 \times 8$. Το pooling layer μειώνει κάθε διάσταση κατά 2 φορές.
    b. **Παράμετροι**: 0 (το pooling layer συγκέντρωσης δεν έχει παραμέτρους)
5. *CONV3-16*
    a. **Διαστάσεις**: $16 \times 16 \times 8$.
    b. **Παράμετροι**: $3 \cdot 3 \cdot 8 \cdot 16 + 16 = 1168$. Το μέγεθος φίλτρου είναι 3×3, 8 κανάλια εισόδου, 16 φίλτρα, συν 16 όροι πόλωσης.
6. *ReLU*
    a. **Διαστάσεις**: $16 \times 16 \times 8$.

b. **Παράμετροι**: 0 (η συνάρτηση ενεργοποίησης δεν έχει παραμέτρους)

7. *FLATTEN*
   a. **Διαστάσεις**: 4096. Flatten της εξόδου $16 \times 16 \times 16$ σε ένα μόνο διάνυσμα.
   b. **Παράμετροι**: 0 (το flattening δεν έχει παραμέτρους)
8. *FC-10*
   a. **Διαστάσεις**: 10
   b. **Παράμετροι**: $4096 \cdot 10 + 10 = 40{,}970$. Πλήρως συνδεδεμένο επίπεδο με 4096 εισόδους και 10 εξόδους, συν 10 όρους πόλωσης.

Ερώτημα 2

Περιγράψτε επιγραμματικά δύο λόγους για τους οποίους συνήθως είναι προτιμότερη η χρήση συνελικτικών επιπέδων αντί πλήρως συνδεδεμένων επιπέδων όταν δουλεύουμε με εικόνες.

Λύση

Πηγές στις οποίες βασίστηκα:

- https://www.geeksforgeeks.org/fully-connected-layer-vs-convolutional-layer/

Πλήθος Παραμέτρων: Τα convolutional layers είναι πιο αποδοτικά ως προς τις παραμέτρους σε σύγκριση με τα fully connected layers, καθώς τα convolutional layers μπορούν να μάθουν τοπικά μοτίβα χρησιμοποιώντας μικρά φίλτρα που εφαρμόζονται στο χώρο εισόδου, ενώ τα fully connected layers μαθαίνουν παγκόσμια μοτίβα που απαιτούν περισσότερες παραμέτρους.

Καταλληλότητα δεδομένων: Τα convolutional layers πλεονεκτούν ιδιαίτερα για χωρικά δεδομένα όπως οι εικόνες, όπου η τοπικότητα και η αμετάβλητη μετάφραση είναι σημαντικές, ενώ τα πλήρως συνδεδεμένα στρώματα είναι πιο ευέλικτα και μπορούν να χρησιμοποιηθούν με οποιαδήποτε μορφή δεδομένων.

Εκμάθηση χαρακτηριστικών: Τα συνεπτυγμένα στρώματα έχουν σχεδιαστεί για να μαθαίνουν και να γενικεύουν αυτόματα χαρακτηριστικά από τα δεδομένα εισόδου, όπως ακμές στα αρχικά στρώματα που ακολουθούνται από πιο σύνθετες δομές σε βαθύτερα στρώματα, ενώ τα πλήρως συνδεδεμένα στρώματα δεν αναγνωρίζουν εγγενώς τέτοια ιεραρχικά μοτίβα χωρίς προηγούμενη αναδιαμόρφωση των δεδομένων εισόδου.

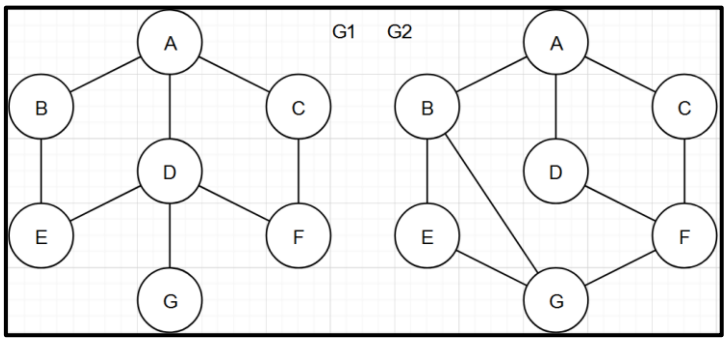
Χρήση σε αρχιτεκτονικές: Στην πράξη, πολλές αρχιτεκτονικές βαθιάς μάθησης χρησιμοποιούν συνδυασμό και των δύο τύπων στρωμάτων. Τα συνεπτυγμένα στρώματα χρησιμοποιούνται συνήθως στα προηγούμενα στάδια για την εξαγωγή και εκμάθηση χαρακτηριστικών, ενώ τα πλήρως συνδεδεμένα στρώματα χρησιμοποιούνται συχνά στο τέλος του δικτύου για να κάνουν προβλέψεις με βάση αυτά τα χαρακτηριστικά.

Features	**Fully Connected Layer**	**Convolutional Layer**
Definition	Every neuron is connected to every neuron in the previous layer.	Neurons are connected only to a local region of the previous layer.
Connectivity	Dense connections; each neuron connects to all neurons in the previous layer.	Sparse connections; each neuron connects only to a local patch of the input.
Parameters	Large number of parameters due to full connectivity.	Fewer parameters due to shared weights and local connectivity.

Weight Sharing	No weight sharing; each connection has its own weight.	Weights are shared across spatial positions, reducing the number of parameters.
Typical Use Cases	Final classification layers in neural networks.	Feature extraction, especially in image and video processing.
Computation Cost	Higher computational cost due to large number of connections.	Lower computational cost per neuron due to local connections.
Overfitting	Higher risk of overfitting due to large number of parameters.	Lower risk of overfitting due to fewer parameters and regularization effects of local connections.
Dimensionality Reduction	Does not inherently reduce dimensionality.	Can reduce dimensionality through pooling layers.
Examples	Multilayer Perceptron (MLP), Dense layers in CNNs.	Convolutional Neural Networks (CNNs), such as layers in AlexNet, VGGNet.

Θέμα 2

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.



Λύση

Graph G_1						
Node	**Neighbors**	**i=0**	**i=1**	**Hash 1**	**i=2**	**Hash 2**
A	B,C,D	1	1; 1,1,1	4	4; 3,3,5	11
B	A,E	1	1; 1,1	3	3; 3,4	7
C	A,F	1	1; 1,1	3	3; 3,4	7
D	A,E,F,G	1	1; 1,1,1,1	5	5; 2,3,3,4	13
E	B,D	1	1; 1,1	3	3; 3,5	8
F	C,D	1	1; 1,1	3	3; 3,5	8
G	D	1	1; 1	2	2; 5	6
Graph G_2						
Node	**Neighbors**	**i=0**	**i=1**	**Hash 1**	**i=2**	**Hash 2**
A	B,C,D	1	1; 1,1,1	4	4; 3,3,4	10
B	A,E,G	1	1; 1,1,1	4	4; 3,4,4	12
C	A,F	1	1; 1,1	3	3; 4,4	9
D	A,F	1	1; 1,1	3	3; 4,4	9
E	B,G	1	1; 1,1	3	3; 4,4	9
F	C,D,G	1	1; 1,1,1	4	4; 3,3,4	10
G	B,E,F	1	1; 1,1,1	4	4; 3,4,4	12

Στο Hash 1, κοιτώ τις unique ετικέτες του Iteration 1 και τις ονοματίζω. Με βάση αυτά προκύπτει το Iteration 2. Οι unique ετικέτες που έχουμε είναι οι:

- (1;-): 1
- (1; 1): 2
- (1; 1,1): 3
- (1; 1,1,1): 4
- (1; 1,1,1,1): 5

Στο Hash 2, κοιτώ τις unique ετικέτες του Iteration 2 και τις ονοματίζω. Με βάση αυτά προκύπτει το Iteration 3. Οι unique ετικέτες που έχουμε είναι οι:

- (2; 5): 6
- (3; 3,4): 7
- (3; 3,5): 8
- (3; 4,4): 9
- (4; 3,3,4): 10
- (4; 3,3,5): 11

- (4; 3,4,4): 12
- (5; 2,3,3,4): 13

Παρατηρώ πως οι γράφοι δεν έχουν καμία ετικέτα κοινή μεταξύ τους, επομένως σταματώ τα iterations.

Κρατώ στον παρακάτω πίνακα τις unique ταμπέλες και τις φορές που εμφανίστηκαν σε κάθε γράφο αντίστοιχα. Στην τελευταία γραμμή έχω το γινόμενο της αντίστοιχης στήλης (παρατηρούμε κιόλας πως για τις ετικέτες 6-12 ότι τα γινόμενα είναι 0, επιβεβαιώνοντας πως δεν χρειάζεται άλλη επανάληψη):

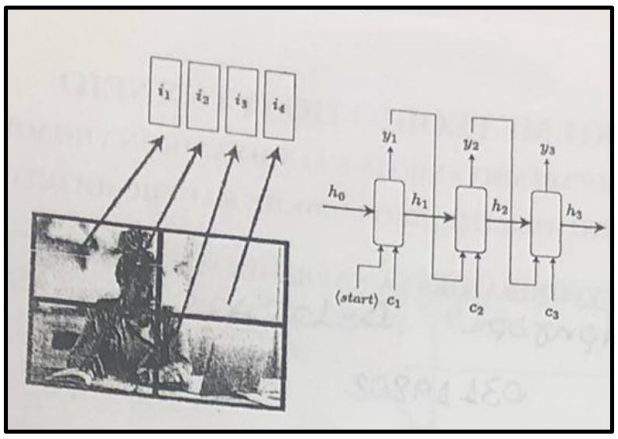
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
$\varphi(G_1)$	7	1	4	1	1	1	2	2	0	0	1	0	1
$\varphi(G_2)$	7	0	3	4	0	0	0	0	3	2	0	2	0
Product	49	0	12	4	0	0	0	0	0	0	0	0	0

Το άθροισμα των γινομένων είναι 65, δηλαδή:

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

Θέμα 3

Εξετάστε το πρόβλημα που φαίνεται στο σχήμα, όπου δίνεται μια εικόνα και το αντικείμενο είναι να δημιουργηθεί μια περιγραφική λεζάντα 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 $i_j \in R^2$ όπως φαίνεται στο σχήμα. Στη συνέχεια, η λεζάντα 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 (start) και (stop) με το ακόλουθο indexing: $V = [(start), (stop), teacher, student, reading, studying]$ και με embedding vectors:

$$y_{<start>} = [0,0,0]^T, y_{<stop>} = [1,1,1]^T, y_{teacher} = [-1,2,0]^T, y_{student} = [1,-2,0]^T,$$

$$y_{reading} = [0,-1,-1]^T, y_{studying} = [0,2,1]^T$$

Ερώτημα 1

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 $\boldsymbol{x_t} \in \boldsymbol{R^5}$, η οποία είναι **concatenation** της προηγούμενης εξόδου που ενσωματώνει $\boldsymbol{y_{t-1}} \in \boldsymbol{R^3}$ και μια αναπαράσταση εικόνας $\boldsymbol{c_t} \in \boldsymbol{R^2}$ (με αυτή τη σειρά) και χρησιμοποιεί αυτήν την είσοδο και την προηγούμενη κρυφή κατάσταση $\boldsymbol{h_{t-1}}$ για να υπολογίσει τη νέα κατάσταση $\mathbf{h_t}$. Αυτό ακολουθείται από ένα γραμμικό επίπεδο εξόδου (linear output layer) με πίνακα hidden to output:

$$W_{gh} = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{matrix} \#<start> \\ \#<stop> \\ \#teacher \\ \#student \\ \#reading \\ \#studying \end{matrix}$$

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

$$W_{hx} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \qquad W_{hh} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

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

Σε αυτήν την ερώτηση, υποθέστε ότι το $\boldsymbol{c_t} \coloneqq \boldsymbol{c}$ 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. Υποθέστε τις 3 πρώτες λέξεις της λεζάντας χρησιμοποιώντας **greedy coding** (επιλογή της πιο πιθανής λέξης σε κάθε βήμα).

Λύση

Υπολογισμός της μέσης αναπαράστασης χαρακτηριστικών της εικόνας c

Για να βρούμε το c , παίρνουμε το μέσο όρο των τεσσάρων αναπαραστάσεων χαρακτηριστικών:

$$c = \frac{1}{4}\sum_{j=1}^{4} i_j = \frac{1}{4}([4,0]^T, [0,4]^T, [0,0]^T, [0,0]^T) = [1,1]^T$$

Υπολογισμός της κρυφής κατάστασης και της εξόδου σε κάθε χρονικό βήμα

- *Χρονικό βήμα t=1*

Είσοδος

$$x_1 = [0,0,0,1,1]^T$$

Hidden state

$$h_1 = ReLU(W_{hx} \cdot x_1 + W_{hh} \cdot h_0 + bias_{in}) = ReLU\left(\begin{bmatrix}1 & 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} + \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \cdot 0 + 0\right) =$$

$$= ReLU = \left(\begin{bmatrix}1\\1\end{bmatrix}\right) = \begin{bmatrix}1\\1\end{bmatrix}$$

Έξοδος

$$y_1 = \left(W_{gh} \cdot h_1 + bias_{out}\right) = \begin{bmatrix}-5 & -5\\0 & 3\\1 & 2\\2 & 2\\3 & -1\\2.9 & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}-10\\3\\3\\4\\2\\2,9\end{bmatrix}$$

Η μεγαλύτερη πιθανότητα είναι η 4, στην 4η γραμμή, η λέξη στην οποία αντιστοιχεί είναι η **student**.

- *Χρονικό βήμα t=2*

Είσοδος

$$x_2 = [1,-2,0,1,1]^T$$

Hidden state

$$h_2 = ReLU(W_{hx} \cdot x_2 + W_{hh} \cdot h_1 + bias_{in}) = ReLU\left(\begin{bmatrix}1 & 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ -2\\ 0\\ 1\\ 1\end{bmatrix} + \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \cdot \begin{bmatrix}1\\ 1\end{bmatrix} + 0\right) =$$

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

Έξοδος

$$y_2 = \left(W_{gh} \cdot h_2 + bias_{out}\right) = \begin{bmatrix}-5 & -5\\ 0 & 3\\ 1 & 2\\ 2 & 2\\ 3 & -1\\ 2.9 & 0\end{bmatrix}\begin{bmatrix}3\\ 0\end{bmatrix} = \begin{bmatrix}-15\\ 0\\ 3\\ 6\\ 9\\ 8{,}7\end{bmatrix}$$

Η μεγαλύτερη πιθανότητα είναι η 9, στην 5η γραμμή, η λέξη στην οποία αντιστοιχεί είναι η **reading**.

- *Χρονικό βήμα t=3*

Είσοδος

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

Hidden state

$$h_3 = ReLU(W_{hx} \cdot x_3 + W_{hh} \cdot h_2 + bias_{in}) = ReLU\left(\begin{bmatrix}1 & 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\ -1\\ -1\\ 1\\ 1\end{bmatrix} + \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \cdot \begin{bmatrix}3\\ 0\end{bmatrix} + 0\right) =$$

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

Έξοδος

$$y_3 = \left(W_{gh} \cdot h_3 + bias_{out}\right) = \begin{bmatrix}-5 & -5\\ 0 & 3\\ 1 & 2\\ 2 & 2\\ 3 & -1\\ 2.9 & 0\end{bmatrix}\begin{bmatrix}0\\ 2\end{bmatrix} = \begin{bmatrix}-10\\ 6\\ 4\\ 4\\ -2\\ 0\end{bmatrix}$$

Η μεγαλύτερη πιθανότητα είναι η 6, στην 2η γραμμή, η λέξη στην οποία αντιστοιχεί είναι η **(stop)**.

Ερώτημα 2

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 1ο χρονικό βήμα (t=1), υπολογίστε τις πιθανότητες προσοχής (attention probabilities) και το διάνυσμα εικόνας που προκύπτει c_1. Η 1η λέξη θα είναι ίδια ή διαφορετική από αυτήν στην προηγούμενη ερώτηση;

Λύση

Attention Scores

$$Score = Q \cdot K^T = h_0 \cdot [i_1, i_2, i_3, i_4]^T = \begin{bmatrix}0\\0\end{bmatrix}^T \begin{bmatrix}4 & 0\\0 & 4\\0 & 0\\0 & 0\end{bmatrix} = [0 \quad 0 \quad 0 \quad 0]$$

Scaled Scores

$$Scaled\ Score = \frac{Score}{\sqrt{d_k}} = \frac{[0 \quad 0 \quad 0 \quad 0]}{\sqrt{2}} = [0 \quad 0 \quad 0 \quad 0]$$

Attention Weights

$$P = softmax(ScaledScore) = softmax([0 \quad 0 \quad 0 \quad 0]) =$$

$$= [e^0/4 \quad e^0/4 \quad e^0/4 \quad e^0/4] = [0.25 \quad 0.25 \quad 0.25 \quad 0.25]$$

Τελικά:

$$z = P \cdot V = [0.25 \quad 0.25 \quad 0.25 \quad 0.25] \cdot \begin{bmatrix}4 & 0\\0 & 4\\0 & 0\\0 & 0\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix}$$

Καταλήξαμε στο ίδιο ακριβώς διάνυσμα. Επομένως, η λέξη παραμένει **student**.

Ερώτημα 3

Υπολογίστε τη 2η λέξη της λεζάντας χρησιμοποιώντας τον αποκωδικοποιητή (decoder) RNN με χρήση προσοχής (attention).

Λύση

Attention Scores

$$Score = Q \cdot K^T = h_1 \cdot [i_1, i_2, i_3, i_4]^T = \begin{bmatrix}1\\1\end{bmatrix}^T \begin{bmatrix}4 & 0\\0 & 4\\0 & 0\\0 & 0\end{bmatrix} = [4 \quad 4 \quad 0 \quad 0]$$

Scaled Scores

$$Scaled\ Score = \frac{Score}{\sqrt{d_k}} = \frac{[4 \quad 4 \quad 0 \quad 0]}{\sqrt{2}} = [2.83 \quad 2.83 \quad 0 \quad 0]$$

Attention Weights

Για ευκολία, θέτω: $e^{2.83} + e^{2.83} + e^{0} + e^{0} = den$

$$P = softmax(ScaledScore) = softmax([2.83 \quad 2.83 \quad 0 \quad 0]) =$$

$$= [e^{2.83}/den \quad e^{2.83}/den \quad e^{0}/den \quad e^{0}/den] = [0.4721 \quad 0.4721 \quad 0.0278 \quad 0.0278]$$

Τελικά:

$$z = P \cdot V = [0.4721 \quad 0.4721 \quad 0.0278 \quad 0.0278] \cdot \begin{bmatrix} 4 & 0 \\ 0 & 4 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1.8884 \\ 1.8884 \end{bmatrix}$$

Δεν καταλήξαμε στο ίδιο διάνυσμα.

Νέο Context Vector c_2

$$c_2 = \sum_{i=1}^{4} p_{2i} \cdot i_i = 0.48 \cdot [4,0]^T + 0.48 \cdot [0,4]^T + 0.02 \cdot [0,0]^T + 0.02 \cdot [0,0]^T = [1.92,1.92]^T$$

Hidden State και Έξοδος για t=2

Είσοδος

$$x_2 = [1, -2,0,1.92,1.92]^T$$

Hidden state

$$h_2 = ReLU(W_{hx} \cdot x_2 + W_{hh} \cdot h_1 + bias_{in}) = ReLU\left(\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 0 \\ 1.92 \\ 1.92 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 \right) =$$

$$= ReLU = \left(\begin{bmatrix} 2.92 \\ -0.08 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 3.92 \\ 0.92 \end{bmatrix}$$

Έξοδος

$$y_2 = \left(W_{gh} \cdot h_2 + bias_{out} \right) = \begin{bmatrix} -5 & -5 \\ 0 & 3 \\ 1 & 2 \\ 2 & 2 \\ 3 & -1 \\ 2.9 & 0 \end{bmatrix} \begin{bmatrix} 3.92 \\ 0.92 \end{bmatrix} = \begin{bmatrix} -24.2 \\ 2.76 \\ 5.76 \\ 9.68 \\ 10.84 \\ 11.368 \end{bmatrix}$$

Η μεγαλύτερη πιθανότητα είναι η 11.368, στην 6η γραμμή, η λέξη στην οποία αντιστοιχεί είναι η **studying**.