ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Τεχνητή Νοημοσύνη

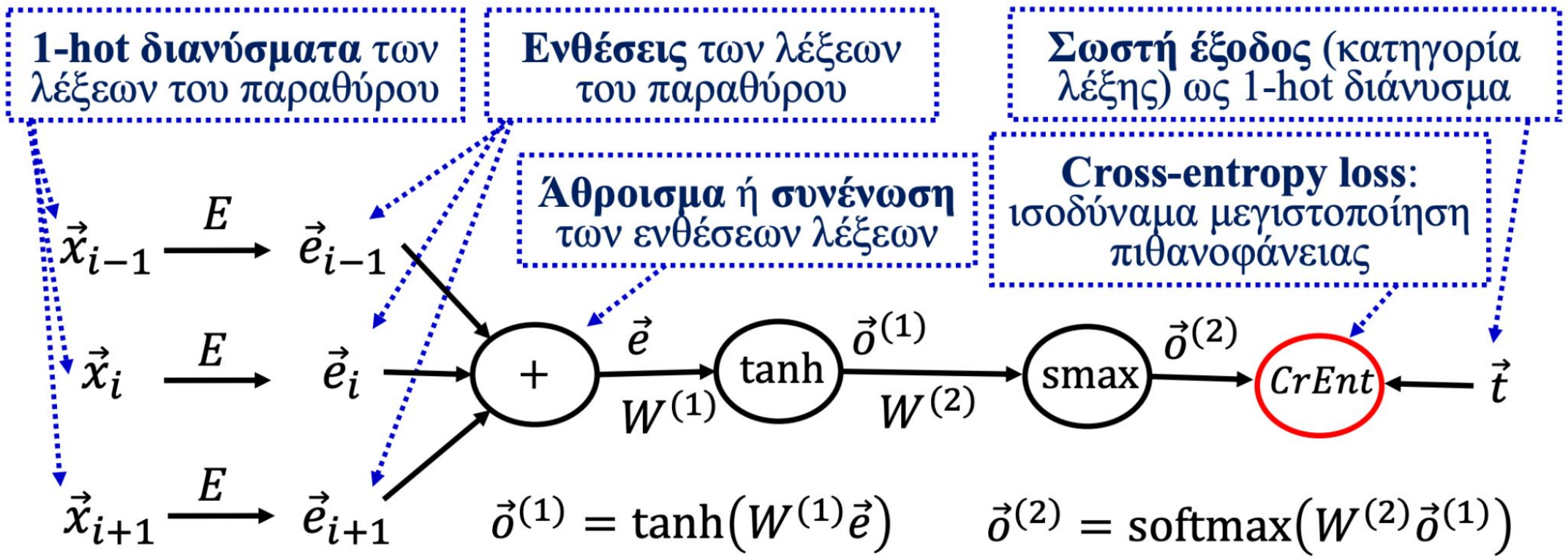
10ο φροντιστήριο (2023-24)

Επιμέλεια: Σοφία Ελευθερίου,
Φοίβος Χαραλαμπάκος

Άσκηση

21.1. Χρησιμοποιούμε το νευρωνικό δίκτυο κυλιόμενου παραθύρου της διαφάνειας 31, για να αναγνωρίζουμε ονόματα προσώπων, οργανισμών και τοποθεσιών. Χρησιμοποιούμε 7 ετικέτες (κατηγορίες) B-I-O, δηλαδή BPerson (beginning of person name, π.χ. «Δημήτριος» στο «Δημήτριος Παπαδόπουλος»), IPerson (inside person name, π.χ. «Παπαδόπουλος» στο «Δημήτριος Παπαδόπουλος»), BOrganization (πρώτη λέξη ονόματος οργανισμού), IOrganization (μη αρχική λέξη ονόματος οργανισμού), BLocation, ILocation και O (other, λέξη που δεν ανήκει σε καμία από τις άλλες κατηγορίες). Το κυλιόμενο παράθυρο καλύπτει 3 λέξεις, την τρέχουσα, την προηγούμενη και την επόμενη. Το μέγεθος του λεξιλογίου είναι $|V| = 100.000$. Κάθε ένθεση λέξης (word embedding) είναι ένα διάνυσμα 300 διαστάσεων. Ο κόμβος "+" συνενώνει (concatenates) τις ενθέσεις των τριών λέξεων του παραθύρου. Το κρυφό επίπεδο αποτελείται από 500 νευρώνες με συνάρτηση ενεργοποίησης *tanh*. Ο πίνακας $W^{(1)}$ περιέχει τα βάρη των νευρώνων του κρυφού επιπέδου, ενώ ο πίνακας $W^{(2)}$ τα βάρη των νευρώνων του επιπέδου εξόδου Ποιες είναι οι διαστάσεις των $E, \vec{e}, W^{(1)}, \vec{o}^{(1)}, W^{(2)}, \vec{o}^{(2)}$; Αιτιολογήστε πλήρως τις απαντήσεις σας.

Κατηγοριοποίηση λέξεων με κυλιόμενο παράθυρο



Απάντηση: Ο πίνακας E περιέχει (ως στήλες) τις ενθέσεις των 100.000 λέξεων του λεξιλογίου. Κάθε ένθεση είναι διάνυσμα (στήλη) 300 διαστάσεων. Άρα ο E έχει διαστάσεις 300×100.000.

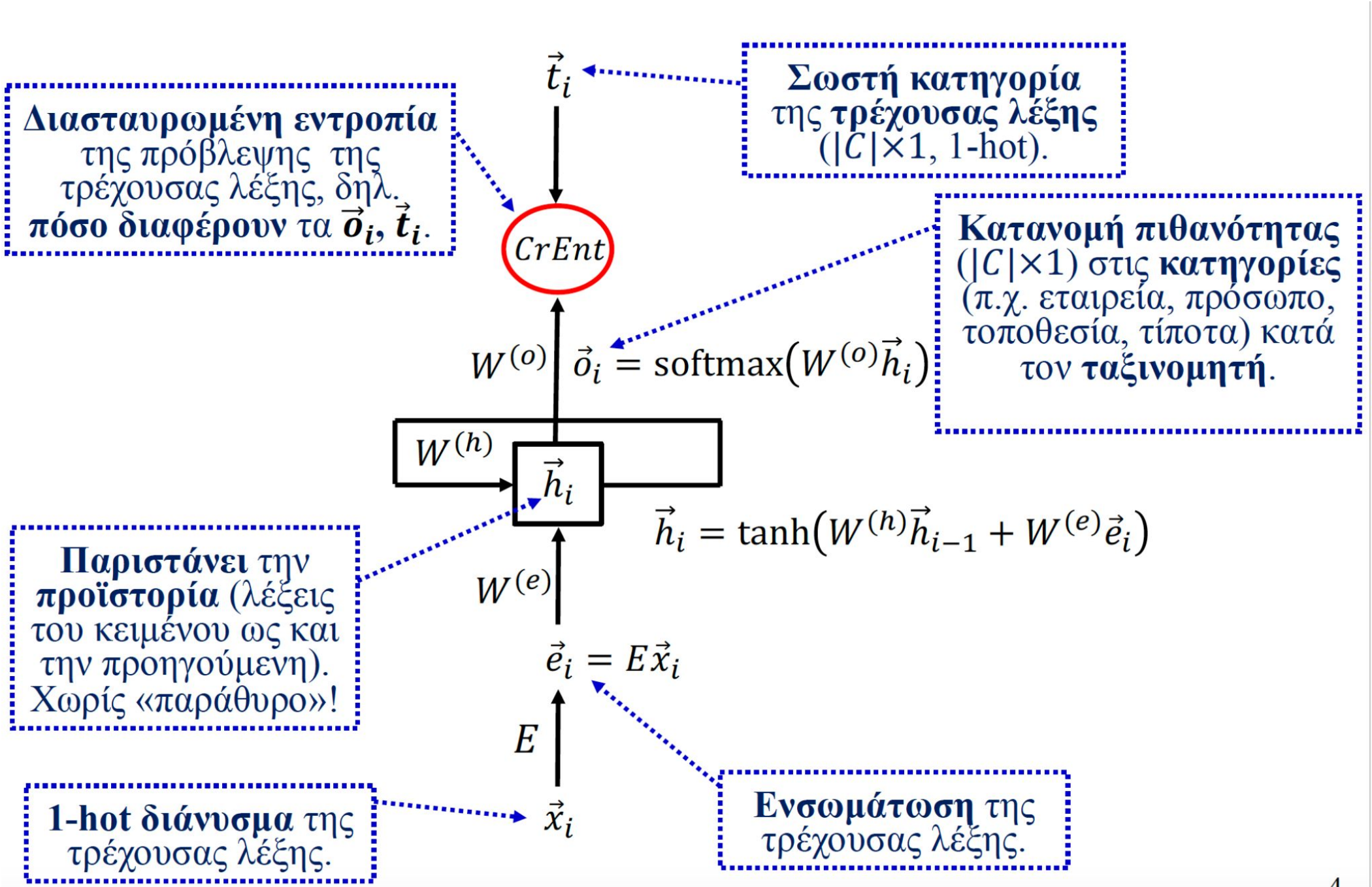
Το διάνυσμα $\vec{e}$ είναι η συνένωση (concatenation) τριών ενθέσεων λέξεων (τρέχουσα, προηγούμενη, επόμενη), κάθε μία από τις οποίες είναι ένα διάνυσμα-στήλη 300×1, επομένως το $\vec{e}$ είναι διαστάσεων 900×1.

Ο πίνακας $W^{(1)}$ έχει διαστάσεις 500×900, ώστε ο πολλαπλασιασμός $W^{(1)}\vec{e}$ να παράγει διάνυσμα 500×1, το οποίο περιέχει τις τιμές των 500 νευρώνων του κρυφού επιπέδου πριν τη συνάρτηση ενεργοποίησης *tanh*. Μετά την εφαρμογή της *tanh*, το διάνυσμα αυτό γίνεται το $\vec{o}^{(1)}$ του σχήματος, δηλαδή περιέχει τις εξόδους των νευρώνων του κρυφού επιπέδου (μετά και την εφαρμογή της συνάρτησης ενεργοποίησης σε κάθε νευρώνα), επομένως και το $\vec{o}^{(1)}$ είναι διαστάσεων 500×1.

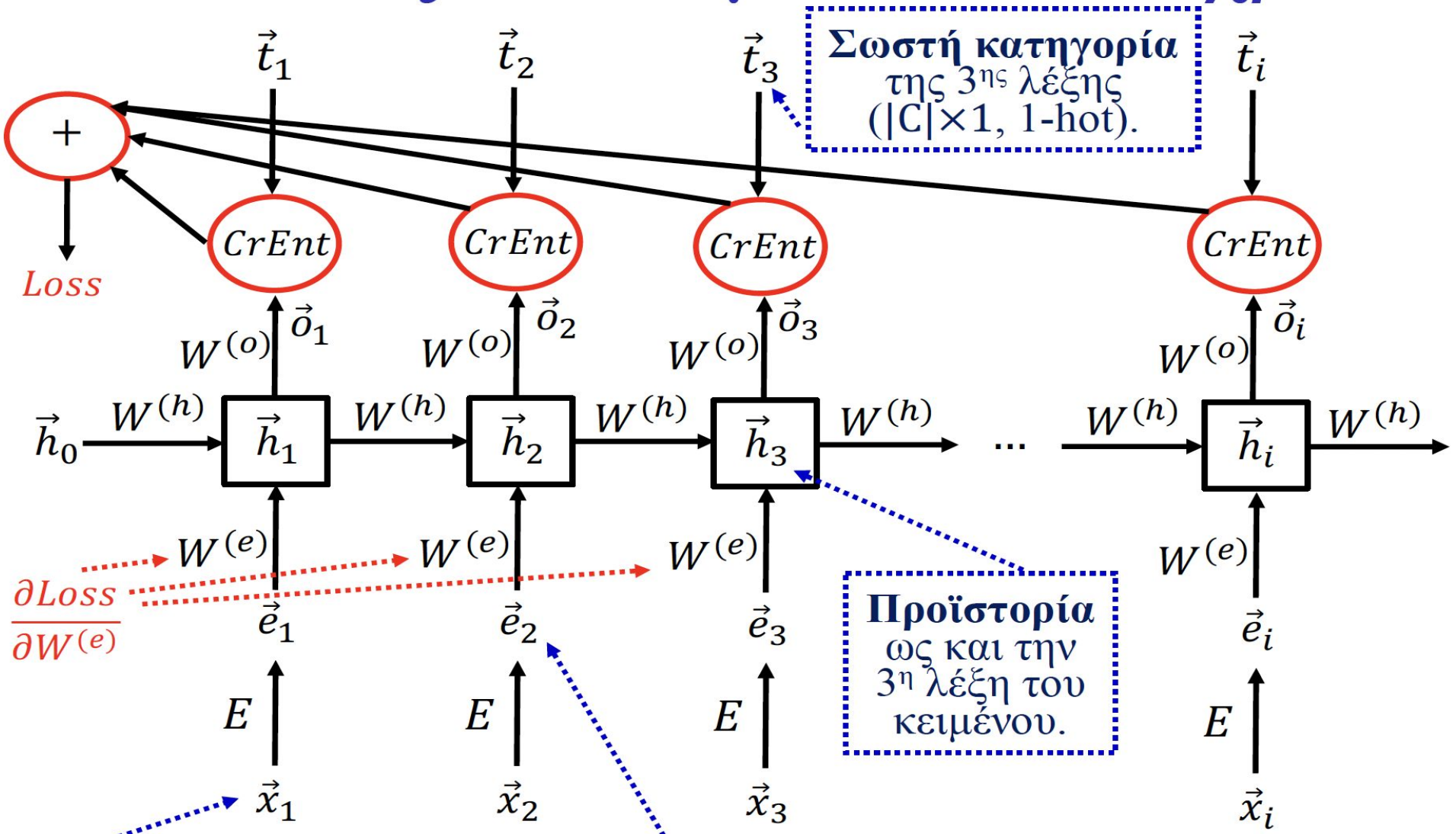
Ο πίνακας $W^{(2)}$ έχει διαστάσεις 7×500, ώστε ο πολλαπλασιασμός $W^{(2)}\vec{o}^{(1)}$ να παράγει διάνυσμα 7×1, το οποίο περιέχει τις τιμές των 7 νευρώνων του επιπέδου εξόδου πριν τη συνάρτηση ενεργοποίησης *softmax*. Μετά την εφαρμογή της *softmax*, το διάνυσμα αυτό γίνεται το $\vec{o}^{(2)}$ του σχήματος, δηλαδή περιέχει τις τιμές των νευρώνων του επιπέδου εξόδου, δηλαδή 7 πιθανότητες, μία για κάθε μία δυνατή ετικέτα (κατηγορία) της τρέχουσας λέξης. Επομένως το $\vec{o}^{(2)}$ είναι διαστάσεων 7×1.

Άσκηση

22.1. Θέλουμε να χρησιμοποιήσουμε το ανατροφοδοτούμενο νευρωνικό δίκτυο (RNN) των διαφανειών 4 και 4, για να αναγνωρίζουμε ονόματα προσώπων, οργανισμών και τοποθεσιών. Χρησιμοποιούμε ετικέτες (κατηγορίες) B-I-O, όπως στην άσκηση 21.1, άρα 7 κατηγορίες. Το μέγεθος του λεξιλογίου είναι $|V| = 100.000$. Κάθε ένθεση λέξης (word embedding) είναι ένα διάνυσμα 300 διαστάσεων. Το κρυφό επίπεδο (η κατάσταση του RNN) αποτελείται από 500 νευρώνες, δηλαδή το $\vec{h}_i$ είναι διάνυσμα 500×1. Ποιες είναι οι διαστάσεις των $E, \vec{e}_i, W^{(h)}, W^{(e)}, W^{(o)}, \vec{o}_i$; Αιτιολογήστε τις απαντήσεις σας.


$\vec{t}_i$
Σωστή κατηγορία της **τρέχουσας λέξης** ($|C|\times 1$, 1-hot).
Διασταυρωμένη εντροπία της πρόβλεψης της τρέχουσας λέξης, δηλ. **πόσο διαφέρουν** τα $\vec{o}_i, \vec{t}_i$.
$CrEnt$
Κατανομή πιθανότητας ($|C|\times 1$) στις **κατηγορίες** (π.χ. εταιρεία, πρόσωπο, τοποθεσία, τίποτα) κατά τον **ταξινομητή**.
$W^{(o)}$
$\vec{o}_i = \mathrm{softmax}\left(W^{(o)}\vec{h}_i\right)$
$W^{(h)}$
$\vec{h}_i$
$\vec{h}_i = \tanh\left(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i\right)$
Παριστάνει την **προϊστορία** (λέξεις του κειμένου ως και την προηγούμενη). Χωρίς «παράθυρο»!
$W^{(e)}$
$\vec{e}_i = E\vec{x}_i$
E
1-hot διάνυσμα της τρέχουσας λέξης.
$\vec{x}_i$
Ενσωμάτωση της τρέχουσας λέξης.

RNN «ξεδιπλωμένο» στον χρόνο



Σωστή κατηγορία της 3ης λέξης (|C|×1, 1-hot).

Προϊστορία ως και την 3η λέξη του κειμένου.

1-hot διάνυσμα της 1ης λέξης του κειμένου

Ενσωμάτωση της 2ης λέξης του κειμένου

$$\vec{h}_i = \tanh\left(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i\right)$$

$$\vec{o}_i = \mathrm{softmax}\left(W^{(o)}\vec{h}_i\right)$$

Απάντηση: Ο πίνακας E περιέχει (ως στήλες) τις ενθέσεις των 100.000 λέξεων του λεξιλογίου. Κάθε ένθεση λέξης είναι διάνυσμα (στήλη) 300 διαστάσεων. Άρα ο E έχει διαστάσεις 300×100.000.

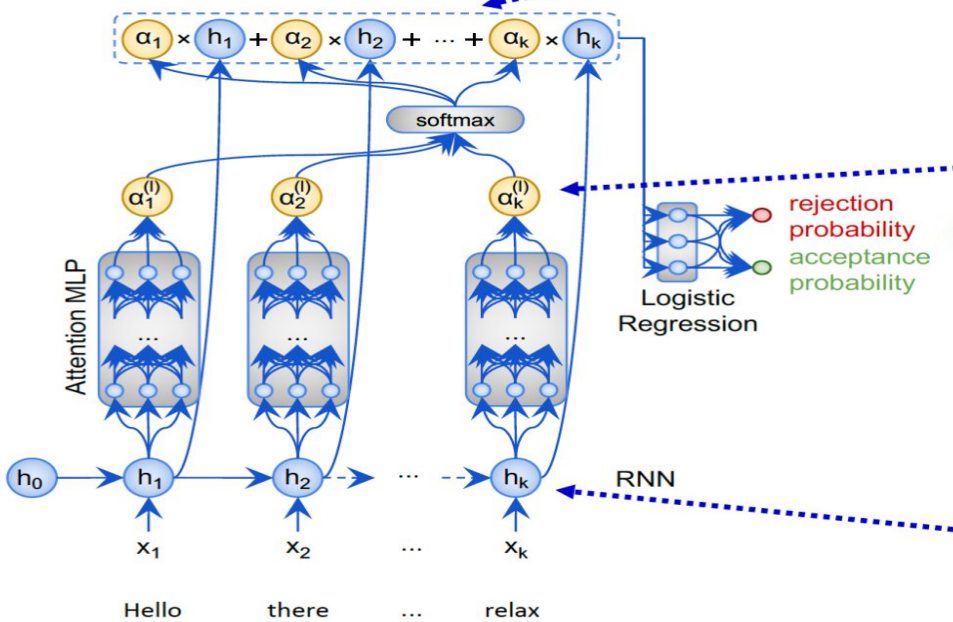
Το διάνυσμα $\vec{e}_i$ είναι η ένθεση (embedding) της i-στής λέξης της εισόδου (π.χ. μιας πρότασης), άρα είναι διαστάσεων 300×1. Το ίδιο συμπέρασμα προκύπτει και από την παρατήρηση ότι ο πολλαπλασιασμός $E\vec{x}_i$ επιστρέφει την i-στή στήλη του πίνακα E.

Ο πίνακας $W^{(h)}$ έχει διαστάσεις 500×500, ενώ ο πίνακας $W^{(e)}$ έχει διαστάσεις 500×300, ώστε τα $W^{(h)}\vec{h}_{i-1}$ και $W^{(e)}\vec{e}_i$ να έχουν τις ίδιες διαστάσεις (500×1), να μπορούν να προστεθούν $(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i)$ και η νέα κατάσταση $\vec{h}_i = \tanh\left(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i\right)$ να έχει πάλι διαστάσεις 500×1, όπως η προηγούμενη κατάσταση $\vec{h}_{i-1}$. Η tanh εφαρμόζεται σε κάθε στοιχείο του διανύσματος $W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i$, χωρίς να αλλάζει τις διαστάσεις του.

Ο πίνακας $W^{(o)}$ έχει διαστάσεις 7×500, ώστε ο πολλαπλασιασμός $W^{(0)}\vec{h}_i$ να παράγει διάνυσμα 7×1 με έναν πραγματικό αριθμό για κάθε κατηγορία. Η softmax στον υπολογισμό $\vec{o}_i = \text{softmax}\left(W^{(o)}\vec{h}_i\right)$ μετατρέπει τους αριθμούς αυτούς σε κατανομή πιθανότητας (μία πιθανότητα για κάθε κατηγορία), χωρίς να αλλάζει τις διαστάσεις του $W^{(o)}\vec{h}_i$. Επομένως το $\vec{o}_i$ έχει και αυτό διαστάσεις 7×1.

22.2. Write down the equations for a modified version of the "RNN with deep self-attention" (slides 16–17), where the uni-directional RNN with GRU cells is replaced by a stacked bi-directional RNN with GRU cells. Use the notation $\text{GRU}(h_{t-1}, \tau_t)$ to denote the new state of a GRU cell with previous state h_{t-1} and input τ_t.

RNN with deep self-attention



$$h_{sum} = \sum_{t=1}^{k} a_t h_t$$

$$P_{a-\text{RNN}}(reject|c) = \sigma(W_p h_{sum} + b_p)$$

$$a_t^{(1)} = \text{RELU}(W^{(1)} h_t + b^{(1)})$$

$$\cdots$$

$$a_t^{(l-1)} = \text{RELU}(W^{(l-1)} a_t^{(l-2)} + b^{(l-1)})$$

$$a_t^{(l)} = W^{(l)} a_t^{(l-1)} + b^{(l)}$$

$$a_t = \text{softmax}(a_t^{(l)}; a_1^{(l)}, \ldots, a_k^{(l)})$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

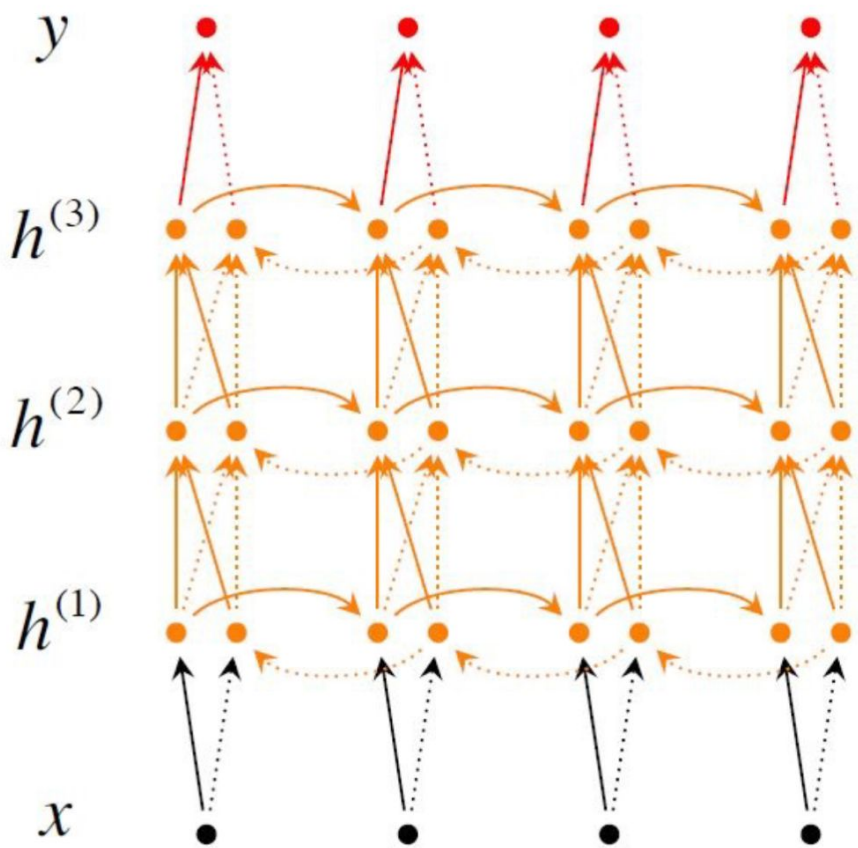
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

J. Pavlopoulos, P. Malakasiotis and I. Androutsopoulos,, "Deeper Attention to Abusive User Content Moderation", EMNLP 2017, http://nlp.cs.aueb.gr/pubs/emnlp2017.pdf.

Deep Bidirectional RNNs by Irsoy and Cardie



From the slides of R. Socher's course "Deep Learning for NLP", 2015. http://cs224d.stanford.edu/

$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)} h_t^{(i-1)} + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)})$$

$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\overrightarrow{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

E.g., probabilities of B, I, O tags at t-th word of the sequence.

Each memory layer passes an intermediate sequential representation to the next.

Answer: At the first layer of the GRU RNN, we have (for $t = 1, \dots, k$):

$$\vec{h}_t^{(1)} = \text{GRU}\left(\vec{h}_{t-1}^{(1)}, x_t\right)$$

$$\overleftarrow{h}_t^{(1)} = \text{GRU}\left(\overleftarrow{h}_{t+1}^{(1)}, x_t\right)$$

$$h_t^{(1)} = [\vec{h}_t^{(1)}; \overleftarrow{h}_t^{(1)}]$$

where $\vec{h}_0^{(1)}$ is the initial state of the left-to-right GRU RNN of the first layer, $\overleftarrow{h}_{k+1}^{(1)}$ is the initial state of the right-to-left GRU RNN of the first layer, `;' denotes concatenation, and $x_1, \dots, x_k$ are the word embeddings of the input word sequence.

Similarly, at the m-th layer of the GRU RNN:

$$\vec{h}_t^{(m)} = \text{GRU}\left(\vec{h}_{t-1}^{(m)}, h_t^{(m-1)}\right)$$

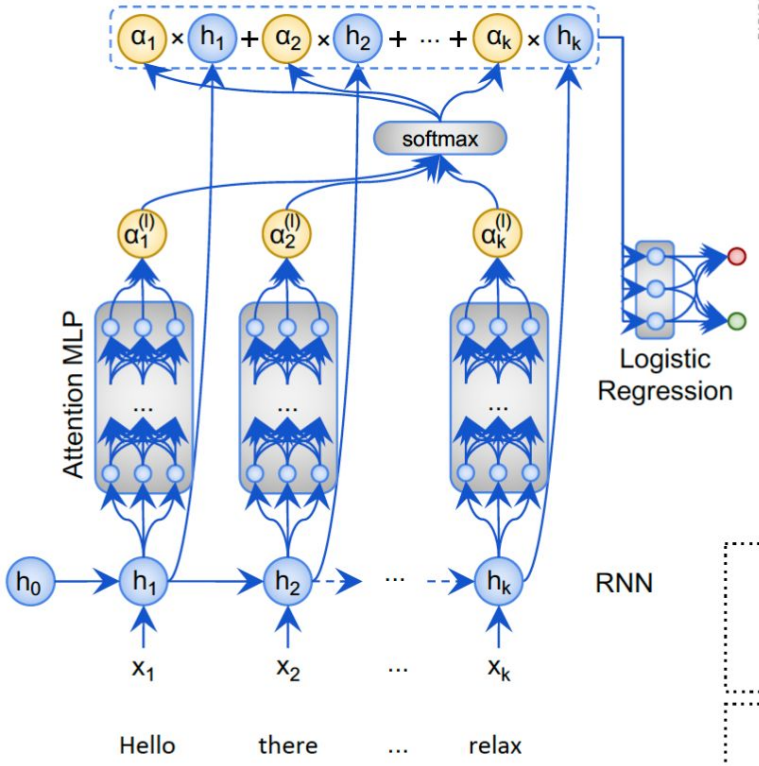
$$\overleftarrow{h}_t^{(m)} = \text{GRU}\left(\overleftarrow{h}_{t+1}^{(m)}, h_t^{(m-1)}\right)$$

$$h_t^{(m)} = [\vec{h}_t^{(m)}; \overleftarrow{h}_t^{(m)}]$$

The other equations remain as on slide 17.

22.3. Modify the equations of the neural network of the previous exercise to support *multi-label classification*, i.e., cases where the same text (e.g., tweet) may belong in multiple classes (labels). Use a separate *label-specific self-attention-head* for each class, which will produce a different distribution of attention scores $a_{c,1}, \ldots, a_{c,k}$ (where k is again the length of the input text, counted in words) and a different $h_{sum,c}$ for each class c. Feed the $h_{sum,c}$ of each class c to a separate (different per class) dense layer with a sigmoid to produce the probability that the input text should be assigned class c.

RNN with deep self-attention



The **attention scores** can also be used to **highlight suspicious words**!

Go	and	hang	yourself	!				
You	are	ignorant	and	vandal	!	Stop	it	!
Thanks	.	Please	go	fuck	yourself	.	ty	!

An **LR layer** uses the **attention-weighted sum** of the **RNN states** as a feature vector.

Each state of the RNN is used as **input** to a **self-attention MLP** (**4 layers**, **same MLP** at each **time step**) that scores the **importance** of the corresponding **word**.

The **RNN states** act as (left-)**context sensitive word embeddings**. We use **GRUs** (128 dims).

Words are mapped to **embeddings** (word2vec, 300 dimensions).

J. Pavlopoulos, P. Malakasiotis and I. Androutsopoulos,, "Deeper Attention to Abusive User Content Moderation", EMNLP 2017, http://nlp.cs.aueb.gr/pubs/emnlp2017.pdf.

Answer: Let C be the set of possible classes (labels). We modify the self-attention MLP of slides 16–17, so that $a_t^{(l)} \in \mathbb{R}^{|C|}$, i.e., $a_t^{(l)}$ is now a vector (not a scalar) containing $|C|$ attention scores $a_{1,t}, \ldots, a_{|C|,t}$ for word position t, one for each possible class. To achieve this, we modify the dimensions of $W^{(l)}$ and $b^{(l)}$ of layer l of the self-attention MLP, to be $|C| \times d$ and $|C|$, respectively, where d is the dimensionality of the previous layer $a_t^{(l-1)}$.

The softmax of slides 16–17 is now applied *label-wise*, on the attention scores of a particular class, i.e., for each possible class c:

$$a_{c,t} = \text{softmax}\left(a_{c,t}^{(l)}; a_{c,1}^{(l)}, \ldots, a_{c,k}^{(l)}\right) = \frac{\exp(a_{c,t}^{(l)})}{\sum_{t'=1}^{k} \exp(a_{c,t'}^{(l)})}$$

We form a separate weighted sum $h_{sum,c}$ for each possible class c:

$$h_{sum,c} = \sum_{t=1}^{k} a_{c,t} h_t^{(M)}$$

where M is the number of stacked GRU RNNs of the previous exercise, and we feed each $h_{sum,c}$ to a separate dense layer $W_{p,c}$ (with bias term $b_{p,c}$) per class c, to compute the probability of the corresponding class:

$$P(c|x_1, \ldots, x_k) = \sigma(W_{p,c} h_{sum,c} + b_{p,c})$$

The other equations of the neural network remain as in the previous exercise.