

Τεχνητή Νοημοσύνη

15η διάλεξη (2025-26)

Ίων Ανδρουτσόπουλος

<http://www.aueb.gr/users/ion/>

Τι θα ακούσετε σήμερα

- Ανατροφοδοτούμενα νευρωνικά δίκτυα (RNNs).
- Στοιβαγμένα (stacked) και διπλής κατεύθυνσης (bidirectional) RNNs.
- RNNs με κελιά GRU ή LSTM.
- RNNs με αυτο-προσοχή (self-attention).
- Global max pooling.
- Κατηγοριοποίηση λέξεων και κειμένων με RNNs.
- Νευρωνικά γλωσσικά μοντέλα.

Επιστροφή στην εξαγωγή πληροφοριών

THIS AGREEMENT is made the 15th day of October 2009
(The “Effective Date”) BETWEEN:

- (1) **Sugar 13 Inc.**, a corporation whose office is at James House, 42-50 Bond Street, London, EW2H TL (“Sugar”);
- (2) **E2 UK Limited**, a limited company whose registered office is at 260 Bathurst Road, Yorkshire, SL3 4SA (“Provider”).

RECITALS:

- A. The Parties wish to enter into a framework agreement which will enable Sugar, from time to time, to [...]
- B. [...]

NO THEREFORE IT IS AGREED AS FOLLOWS:

ARTICLE I - DEFINITIONS

- “Sugar” shall mean: Sugar 13 Inc.
“Provider” shall mean: E2 UK Limited
“1933 Act” shall mean: **Securities Act of 1933**

ARTICLE II - TERMINATION

The Service Period will be for **five (5) years** from the Effective Date (The “Initial Term”). The agreement is considered to be terminated in **October 16, 2014**.

ARTICLE III - PAYMENT - FEES

During the service period monthly payments should occur. The estimated fees for the Initial Term are **£154,800**.

ARTICLE IV - GOVERNING LAW

This agreement shall be governed and construed in accordance with the **Laws of England & Wales**. Each party hereby irrevocably submits to the exclusive jurisdiction of the courts sitting in **Northern London**.

IN WITNESS WHEREOF, the parties have caused their respective duly authorized officers to execute this Agreement.

BY: George Fake
Authorized Officer
Sugar 13 Inc.

BY: Olivier Giroux
CEO
E2 UK LIMITED

Εντοπίζονται: ημερομηνία
έναρξης/λήξης, διάρκεια,
συμβαλλόμενοι, ποσό,
παραπομπές σε νόμους,
αρμόδια δικαστήρια κ.λπ.

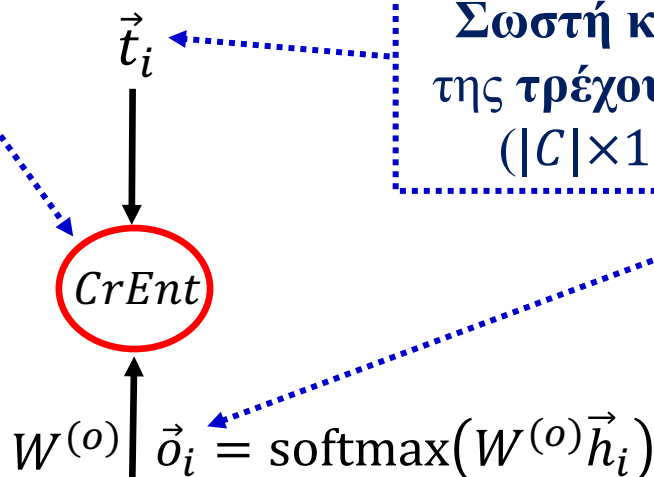
Κατηγοριοποίηση λέξεων με RNN (recurrent NN)

Διασταυρωμένη εντροπία της πρόβλεψης της τρέχουσας λέξης, δηλ. πόσο διαφέρουν τα \vec{o}_i, \vec{t}_i .

Σωστή κατηγορία της τρέχουσας λέξης ($|C| \times 1$, 1-hot).

Κατανομή πιθανότητας ($|C| \times 1$) στις κατηγορίες (π.χ. εταιρεία, πρόσωπο, τοποθεσία, τίποτα) κατά τον ταξινομητή.

Μπορούμε να σκεφτούμε τις γραμμές του $W^{(o)}$ ως ενθέσεις τάξεων.

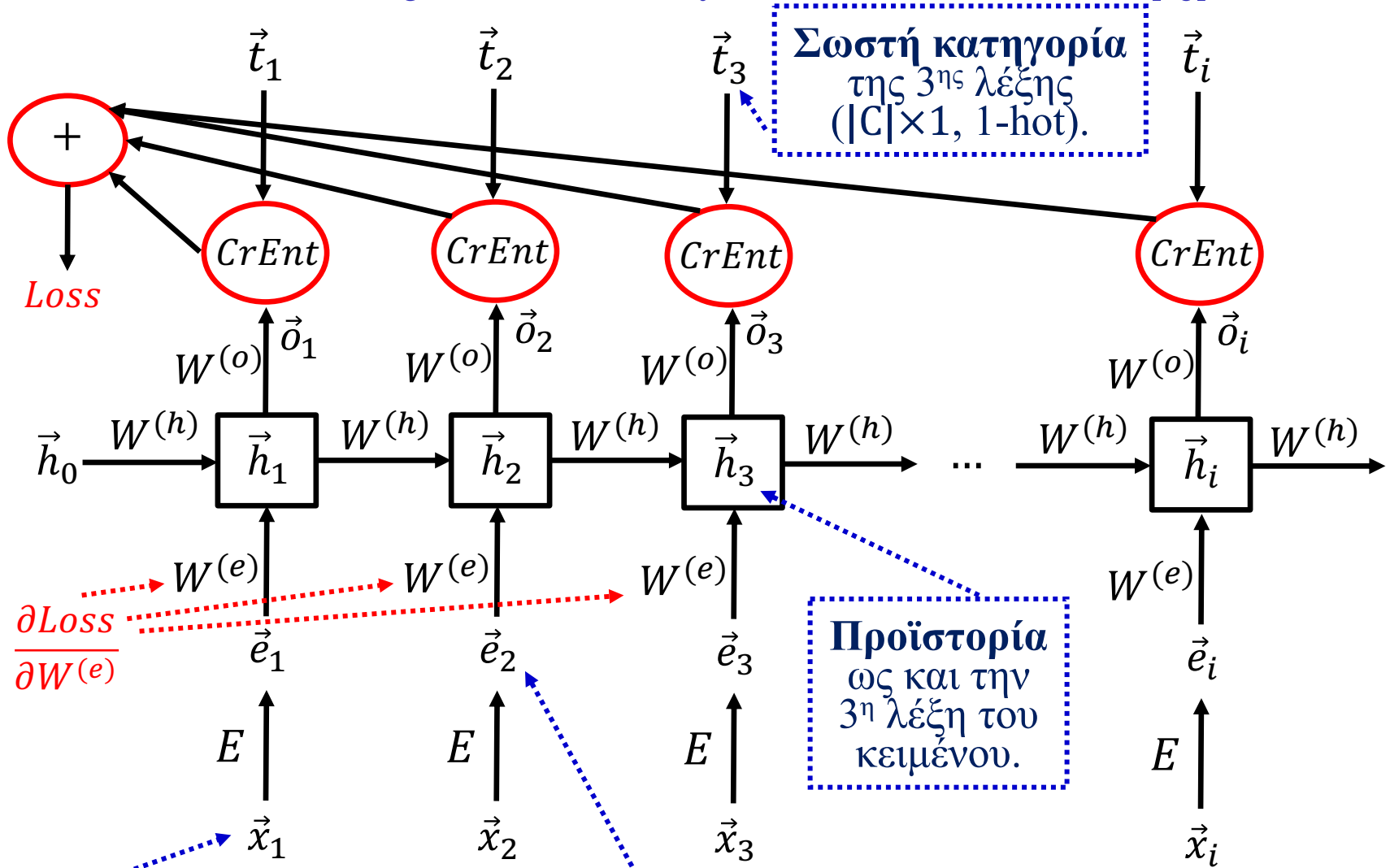


Παριστάνει την προϊστορία (λέξεις του κειμένου ως και την προηγούμενη). Χωρίς «παράθυρο»!

1-hot διάνυσμα της τρέχουσας λέξης.

Ένθεση της τρέχουσας λέξης.

RNN «ξεδιπλωμένο» στον χρόνο



Σωστή κατηγορία της 3ης λέξης ($|C| \times 1, 1\text{-hot}$).

Προϊστορία ως και την 3η λέξη του κειμένου.

1-hot διάνυσμα της 1ης λέξης του κειμένου

Ένθεση της 2ης λέξης του κειμένου

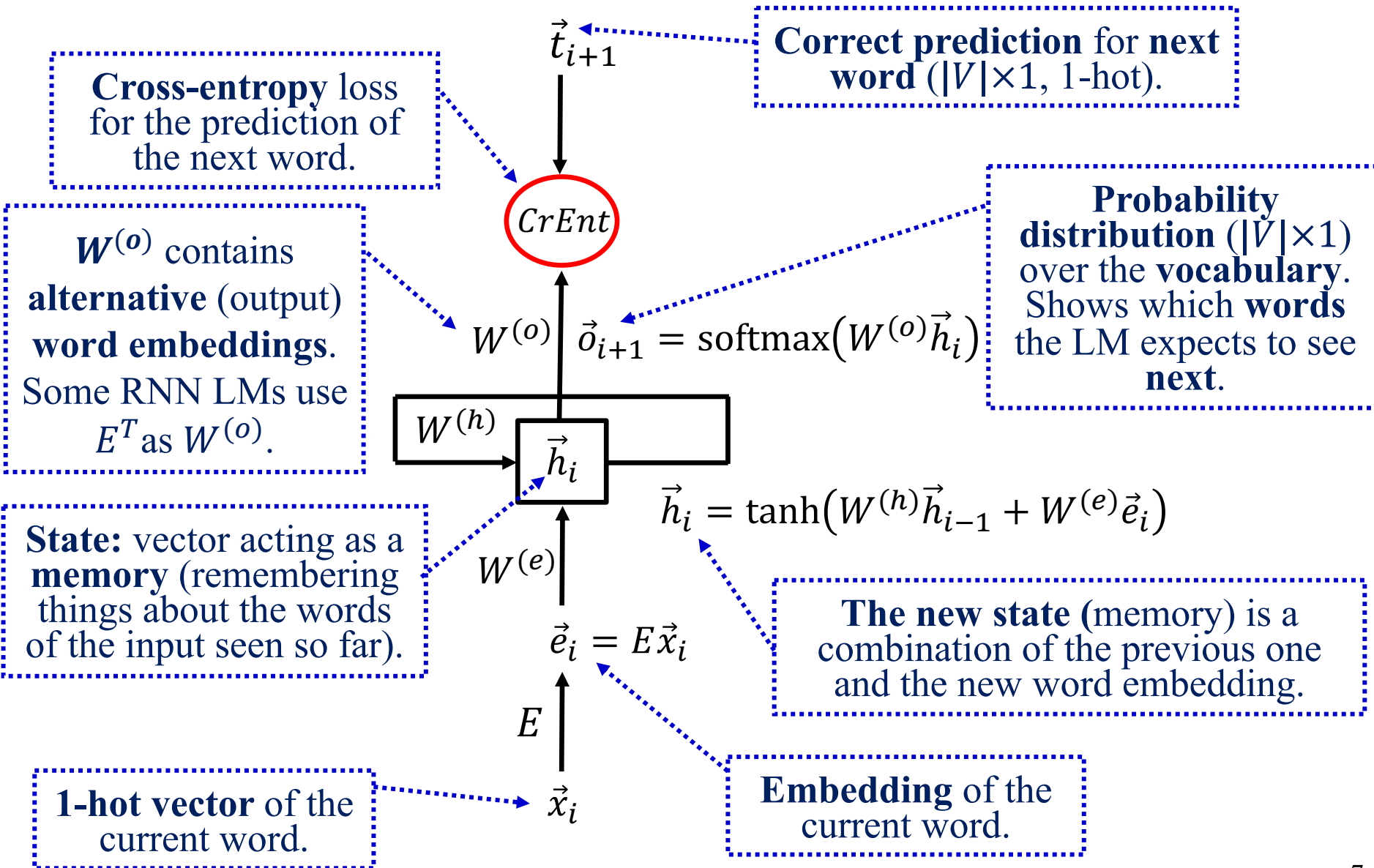
$$\vec{h}_i = \tanh(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i)$$

$$\vec{o}_i = \text{softmax}(W^{(o)}\vec{h}_i)$$

Περισσότερα για τα RNNs

- Εκπαιδεύονται με **ανάστροφη μετάδοση** («ξεδιπλωμένα»)
 - Για **κάθε πρόταση** (ή τμήμα πρότασης ως κάποιο μήκος), την δίνουμε στο **RNN**, υπολογίζουμε τις **προβλέψεις** του RNN, υπολογίζουμε το **συνολικό σφάλμα** (loss) και **ενημερώνουμε** τους **πίνακες βαρών** (π.χ. $W^{(h)}$) με **ανάστροφη μετάδοση**.
 - Συνήθως RNNs με κελιά **GRU** ή **LSTM** (βλ. παρακάτω) δουλεύουν καλύτερα.
 - Μπορούμε να **μάθουμε** και τις **ενθέσεις λέξεων** (πίνακας E) κατά την ανάστροφη μετάδοση, αν έχουμε πάρα πολλά παραδείγματα εκπαίδευσης.
- Το ίδιο μοντέλο μπορεί να χρησιμοποιηθεί για **επισημείωση** **μερών του λόγου**, **αναγνώριση ονομάτων οντοτήτων**, **εξαγωγή στοιχείων συμβολαίων** κ.λπ.
 - Γενικότερα **επισημείωση ακολουθιών** (π.χ. γονιδίων).

RNN language model



Νευρωνικά γλωσσικά μοντέλα

- Μπορούμε να **εκπαιδύσουμε** νευρωνικά γλωσσικά μοντέλα σε **τεράστια σώματα κειμένων** (Wikipedia, βιβλία, ...).
 - Τα γλωσσικά μοντέλα εκπαιδούνται να **προβλέπουν πόσο πιθανή είναι η κάθε λέξη του λεξιλογίου ως επόμενη λέξη**, δοθέντος ενός προηγούμενου (ημιτελούς) κειμένου.
 - **Δεν απαιτείται χειρωνακτική επισημείωση** παραδειγμάτων εκπαίδευσης. Γνωρίζουμε πάντα τη (σωστή) επόμενη λέξη...
 - Μπορούμε να μάθουμε και τις **ενθέσεις λέξεων** (πίνακας E).
- Σε γενικές γραμμές έτσι έχουν εκπαιδευτεί μεγάλα γλωσσικά μοντέλα (**LLMs**), π.χ. Chat-GPT, Claude, Gemini, Llama...
 - Χρησιμοποιούν όμως **Transformers**, μια άλλη μορφή νευρωνικών δικτύων, **πρόσθετη εκπαίδευση με παραδείγματα ερωτήσεων-επιθυμητών αποκρίσεων** κ.λπ.
 - Βλ. επόμενες διαλέξεις.

Νευρωνικά γλωσσικά μοντέλα

- Ένα εκπαιδευμένο γλωσσικό μοντέλο μπορεί να εκτιμήσει πόσο πιθανή είναι μια ακολουθία λέξεων/γραμμάτων.
 - Π.χ. ποια η **πιθανότερη μετάφραση** μεταξύ εναλλακτικών που παράγει ένα σύστημα μηχανικής μετάφρασης.
 - «*Η κυβέρνηση γνώρισε από τις ενάρξεις του 2009 ότι η κατάσταση αντιμετώπισε το φάσμα της χρεοκοπίας.*»
 - «*Η κυβέρνηση γνώριζε από τις ενάρξεις του 2009 ότι η χώρα αντιμετώπιζε το φάσμα της χρεοκοπίας.*»
 - «*Η κυβέρνηση γνώριζε από τις αρχές του 2009 ότι η χώρα αντιμετώπιζε το φάσμα της χρεοκοπίας.*»
- **Πολλαπλασιάζουμε τις πιθανότητες** (προσθέτουμε λογαρίθμους) που δίνει το γλωσσικό μοντέλο στις λέξεις της υπόθεσης.

$$P(w_1, w_2, \dots, w_n) = P(w_1|start) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot P(w_n|w_1, \dots, w_{n-1}) \cong P(w_1|\vec{h}_0) \cdot P(w_2|\vec{h}_1) \cdot P(w_3|\vec{h}_2) \cdot P(w_n|\vec{h}_{n-1})$$

Νευρωνικά γλωσσικά μοντέλα

- Ένα εκπαιδευμένο ΓΜ μπορεί επίσης να συνεχίσει ένα ημιτελές κείμενο (π.χ. ημιτελή πρόταση ή διάλογο).
- Διαλέγοντας κάθε φορά την πιο πιθανή επόμενη λέξη (ή μια από τις πιο πιθανές ή beam search).
- Μια πιθανή συνέχιση ενός ερωτήματος είναι η σωστή απάντηση.
 - Χρειάζεται συνήθως και πρόσθετη εκπαίδευση με ερωτήματα-απαντήσεις.

Default (GPT-3.5)



I teach a 12 week long AI course. Can you suggest what I should cover per week?



Designing a 12-week AI course is a great endeavor. Below is a week-by-week breakdown of topics and concepts you could cover. Keep in mind that the course can be adjusted based on the prior knowledge of your students and the depth you want to go into each topic. Here's a general outline:

Week 1: Introduction to AI

- What is AI? History and milestones.
- Types of AI: Narrow (Weak) vs. General (Strong) AI.
- AI in popular culture and misconceptions.

Week 2: Machine Learning Fundamentals

- Introduction to Machine Learning (ML).
- Supervised, Unsupervised, and Reinforcement Learning.
- Data preprocessing and feature engineering.

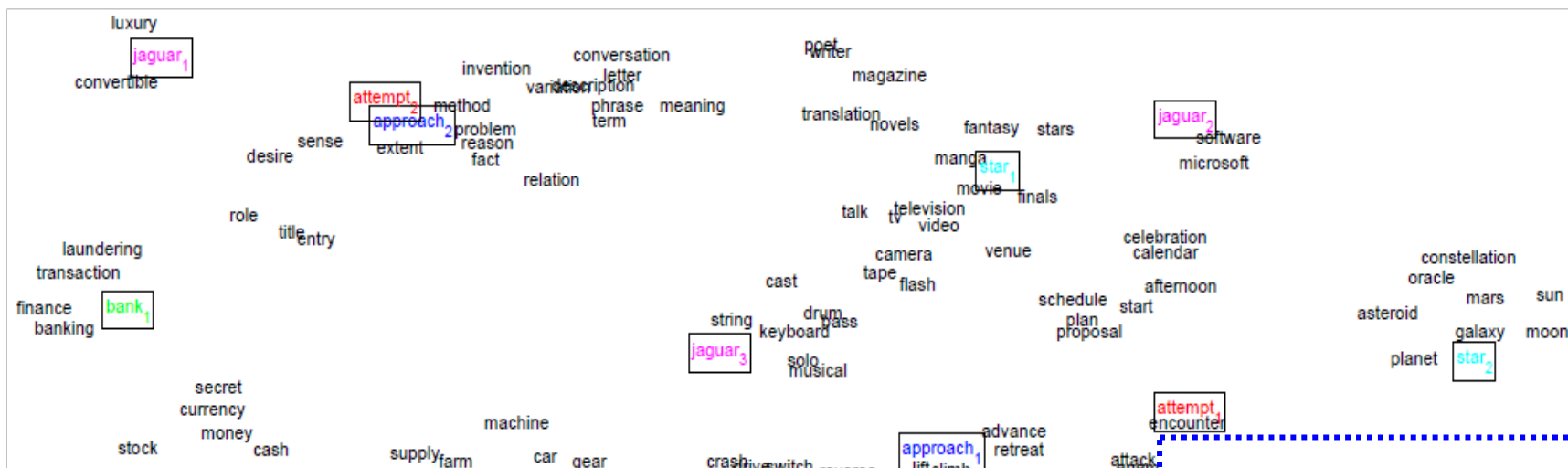
Week 3: Supervised Learning

- Linear Regression.
- Logistic Regression.

Παράδειγμα χρήσης του **ChatGPT** (<https://chat.openai.com/>). Το ChatGPT και παρόμοια LLMs χρησιμοποιούν **Transformers**, αντί για RNNs.

Word sense embeddings

(produced by a method that produces **dense, sense-specific** word embeddings, then **projected to 2 dimensions**)



Most words have multiple senses. Word embeddings (a single point per word) end up being in the **middle of the points that would correspond to their multiple **senses**.**

In a context of financial discussion, we would like the embedding of “bank**” to move **closer** to its **financial** sense, away from its **river** sense.**

Image source: <http://www.socher.org/uploads/Main/MultipleVectorWordEmbedding.png>

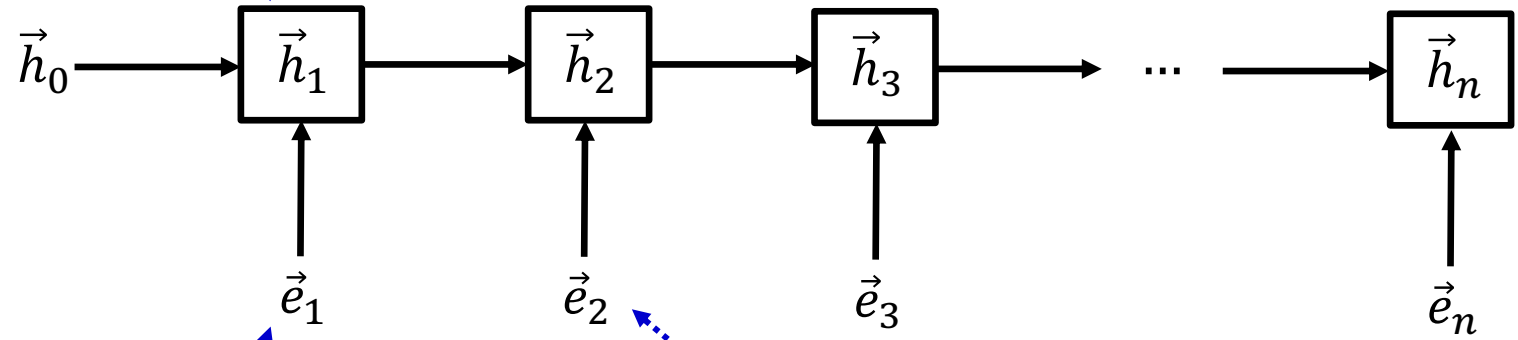
Huan et al. 2012, “Improving Word Representations via Global Context and Multiple Word Prototypes”.

RNNs states as left-context-aware word embeddings

Revised embedding of the 1st word. Knows we are at the beginning of a sentence.

Revised embedding of the 2nd word. Knows the left-context.

We can also treat the \vec{h}_i vectors as the **memory** of the RNN, but in recent NLP work, it's easier to think of them as **revised word embeddings**.



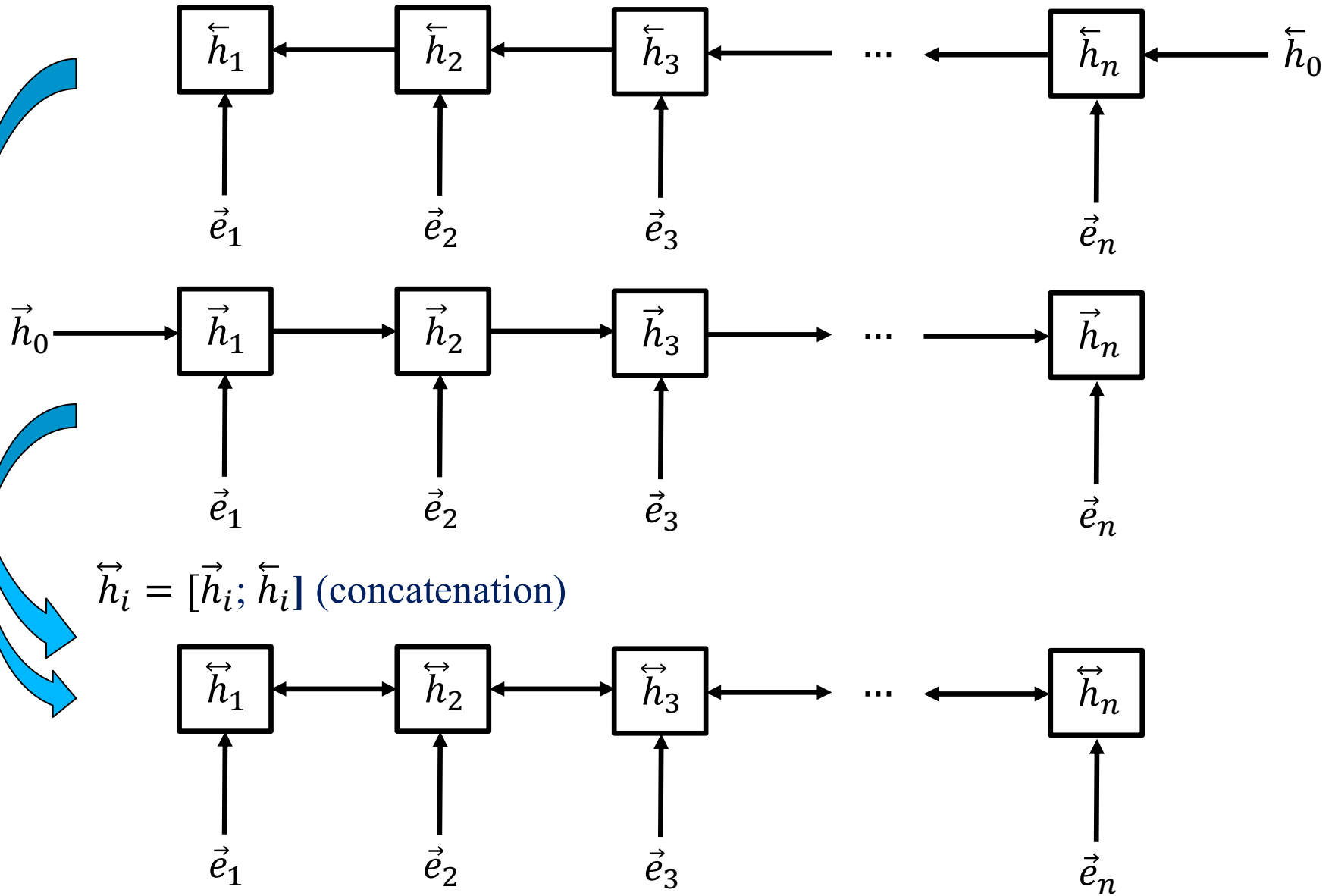
Embedding of the 1st word of the sentence

Embedding of the 2nd word of the sentence

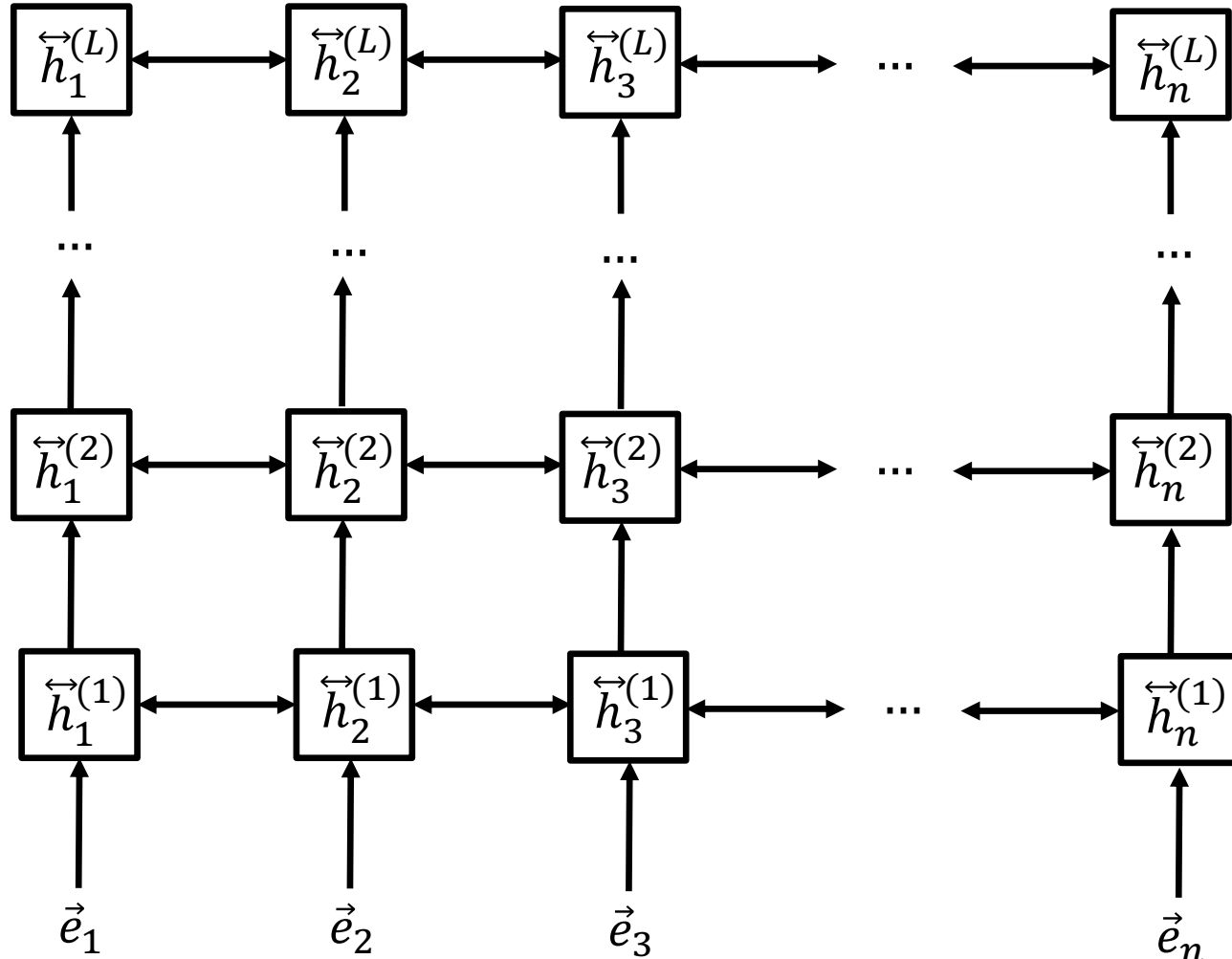
g is an **activation function** (e.g., sigmoid). More complex update mechanisms in practice: **LSTM** or **GRU** cells.

$$\vec{h}_i = g(W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i + \vec{b}^{(h)})$$

Bidirectional RNN (biRNN)



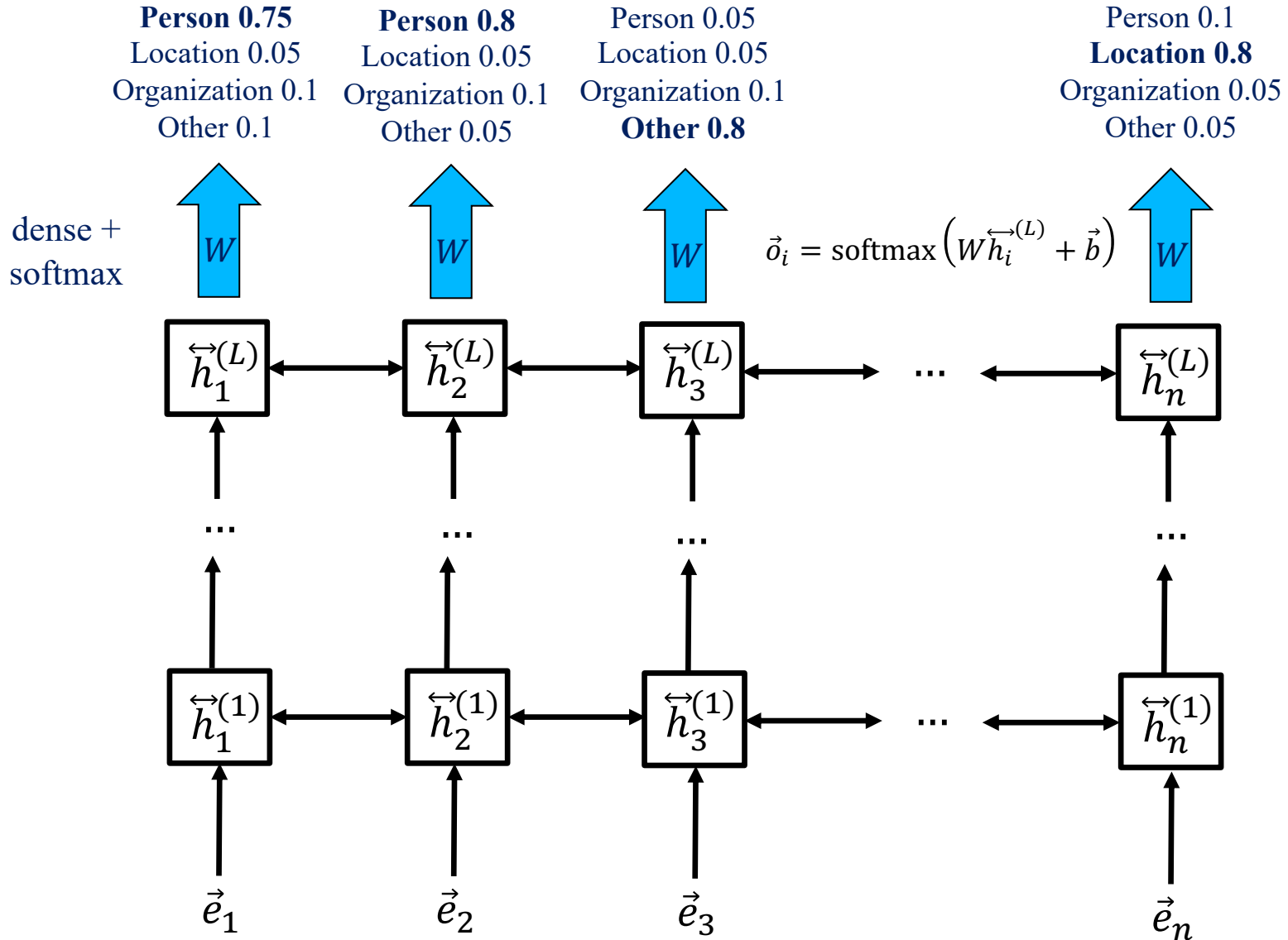
Stacked bidirectional RNN



Each layer revises the word embeddings of the previous (lower) layer. The embeddings become increasingly more context-aware and also increasingly more appropriate for the particular task we address...

Token classification with stacked biRNN

Compare to the correct predictions (sum the cross-entropy loss for all token positions) and backpropagate to adjust all the weights, including the weights of the stacked biRNN.



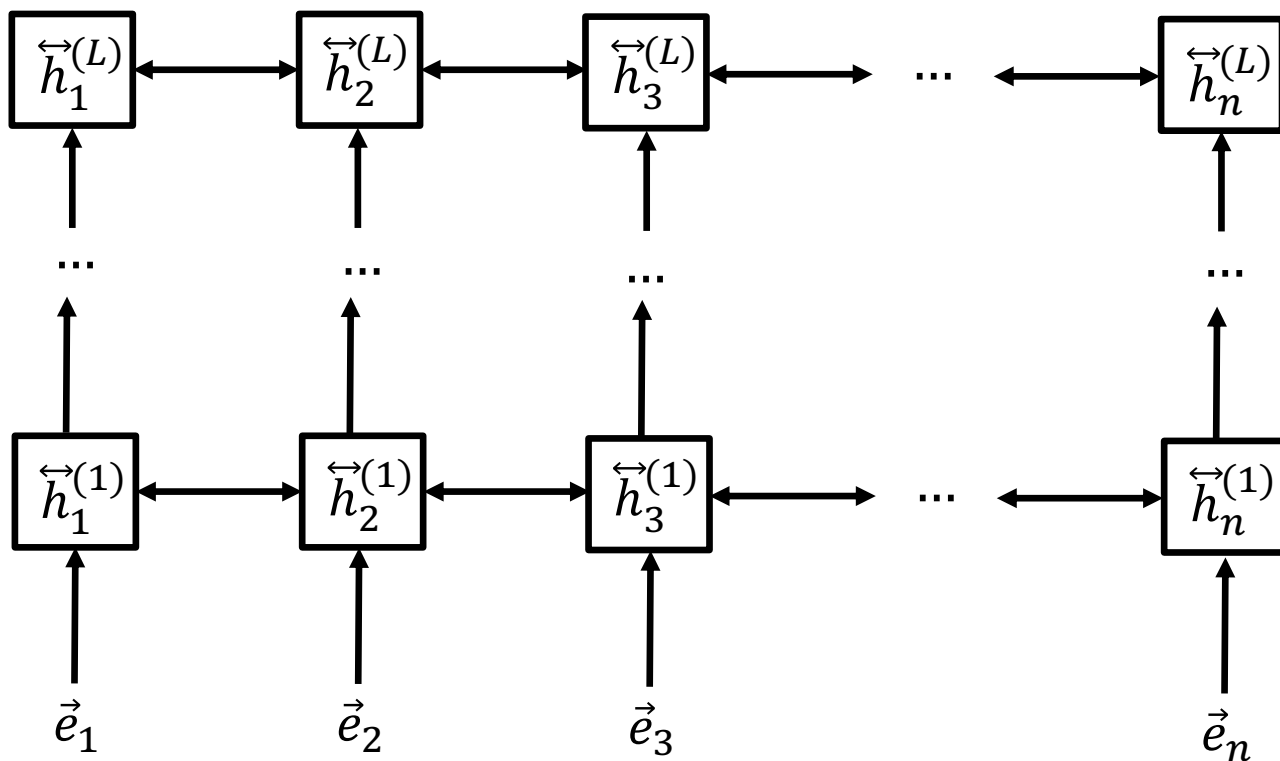
Text classification with stacked biRNN

Compare (via **categorical cross entropy**) the **predicted** \vec{o} to the **correct 1-hot distribution** and **backpropagate** to adjust all the weights, including the weights of the stacked biRNN.

$$\vec{o} = \text{softmax}(W\vec{h}_{max} + \vec{b})$$

$$\vec{h}_{max} = \left\langle \max(\vec{h}_{*,1}^{(L)}), \max(\vec{h}_{*,2}^{(L)}), \dots, \max(\vec{h}_{*,n}^{(L)}) \right\rangle^T$$

Global max-pooling creates a **single vector** containing the **max per dimension** of all the $\vec{h}_i^{(L)}$. We pass it through a **dense layer and softmax (or MLP)** to obtain a **probability per class**.

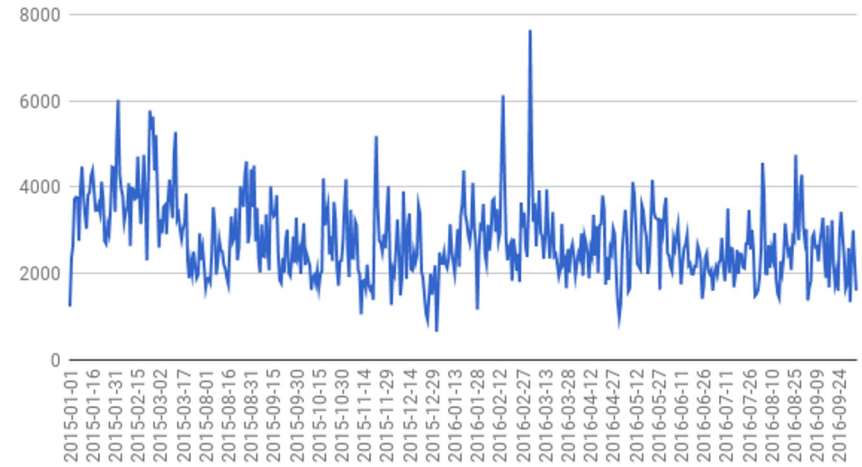


Text classification example: comment moderation

A moderation panel assists the moderators to detect abusive comments, and leads to quicker publication of non-abusive comments.

Highlighting suspicious words using an RNN with self-attention.

Number of comments per day



Moderation Panel

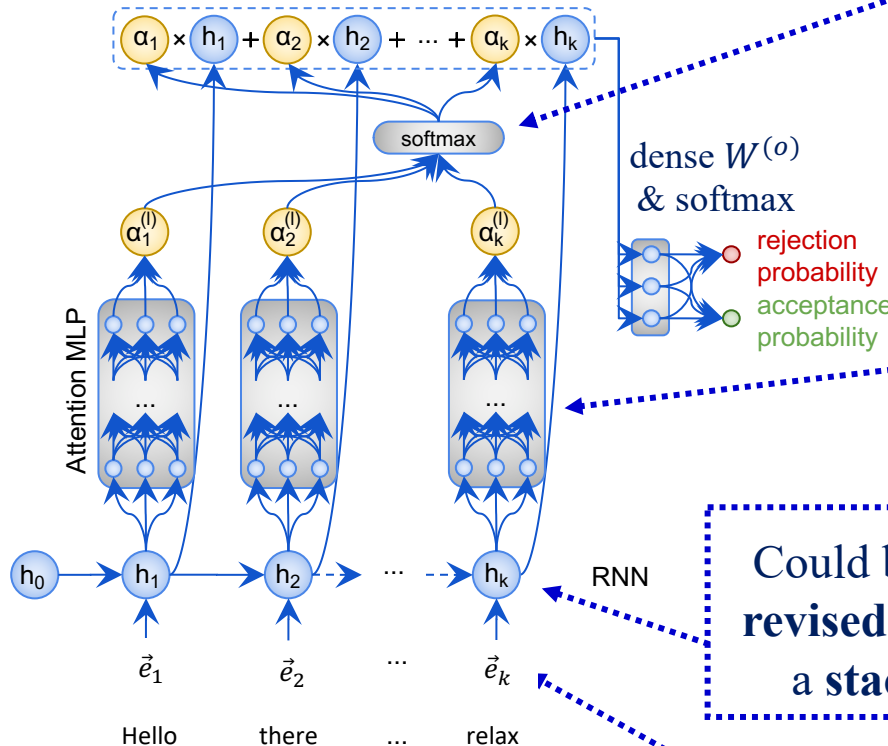
Go	and	hang	yourself	!						85%		
You	are	ignorant	and	vandal	!	Stop	it	!		88%		
Hello	there	try	to	relax						0%		
Thanks	.	Please	go	f#\$@	yourself	.	Ty	!		85%		

RNN with deep self-attention

The **entire input text** is now represented by the **weighted (by a_i scores) sum of the revised embeddings** of its words.

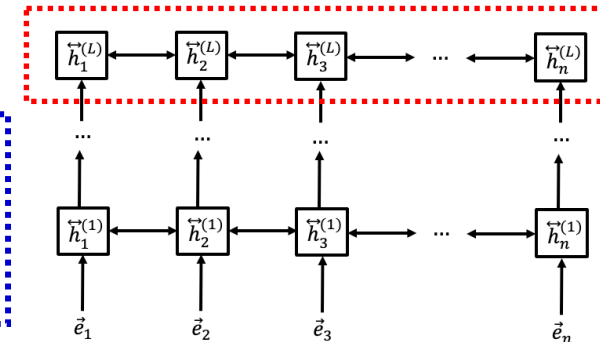
The **softmax** ensures all the a_i scores are between 0 and 1, and that they sum to 1.

We use an **MLP** (the same at all time-steps) to obtain an **attention score** (importance) a_i for each word from its revised embedding h_i . We could also use a **single dense layer**: $a_i = W^{(a)} h_i$.



Could be the **top-level revised embeddings of a stacked biRNN**.

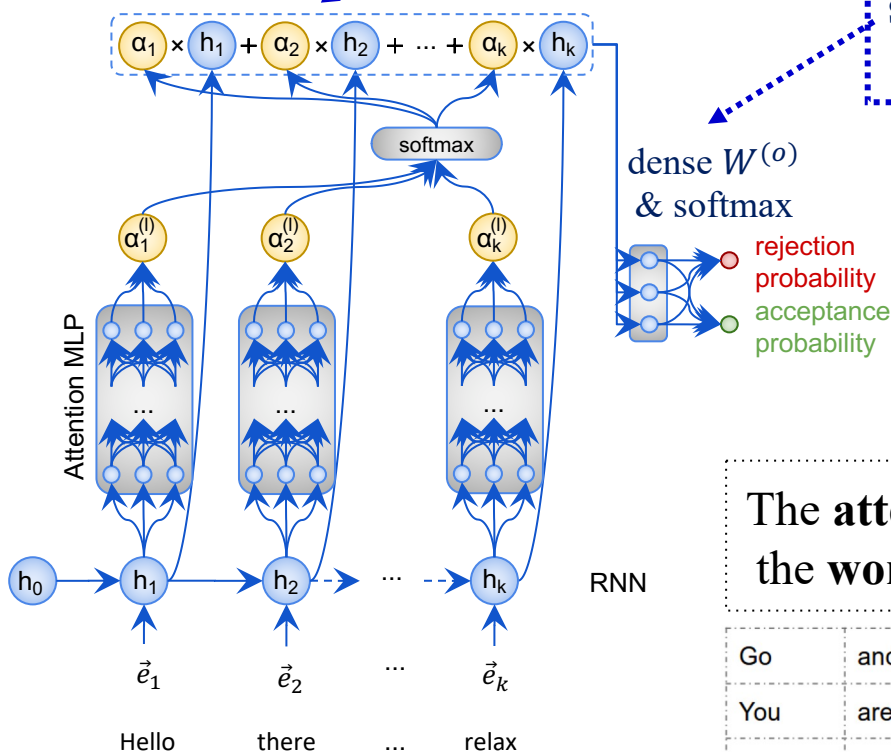
Initial word embeddings (e.g., via Word2Vec).



RNN with deep self-attention

The **entire input text** is now represented by the **weighted (by a_i scores) sum** of the **revised embeddings** of its words.

We pass the **weighted sum vector** (point) through another **dense layer and softmax** to obtain a **probability score** for **each class** (here accept, reject).

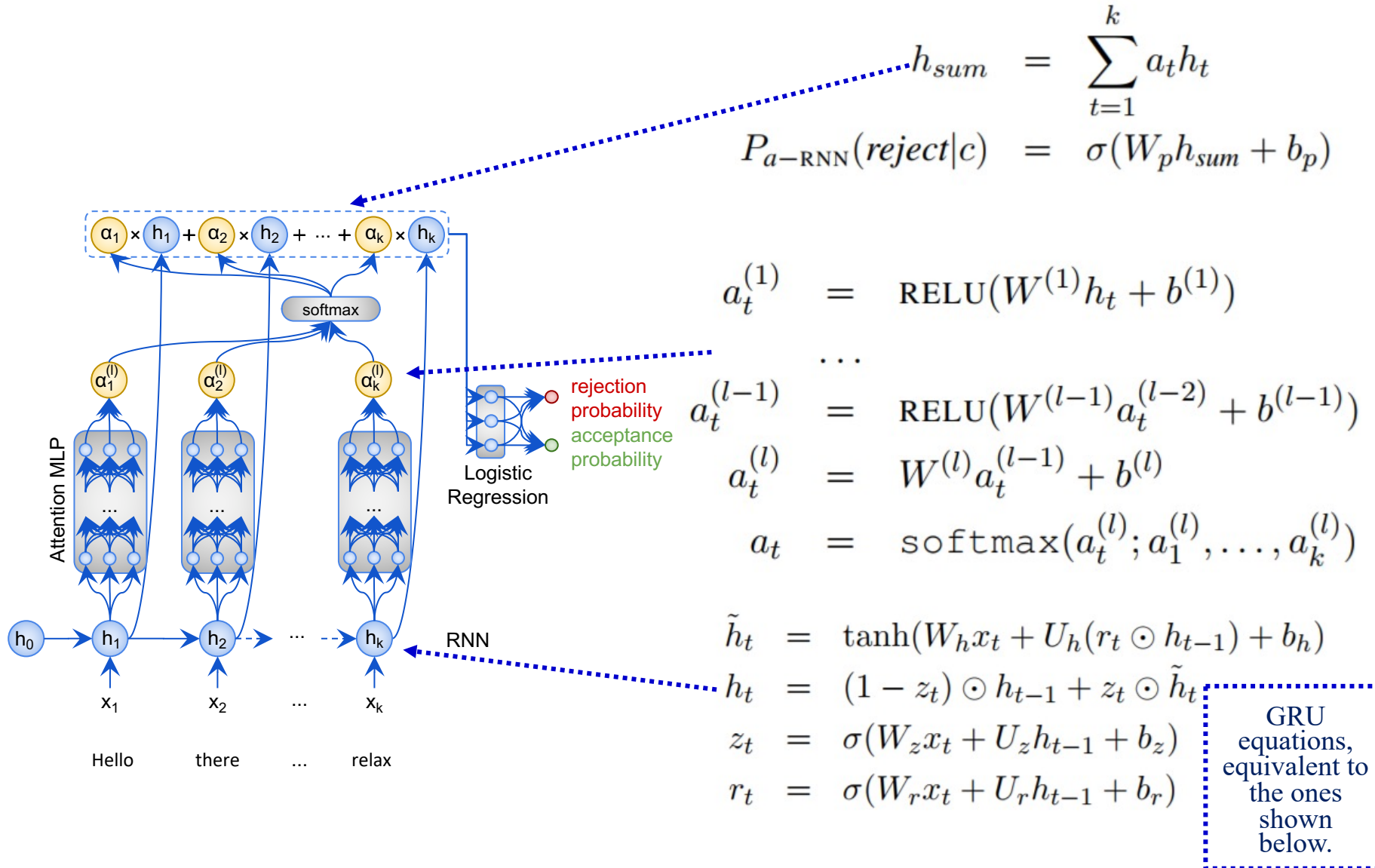


Compare to the correct predictions with a **cross-entropy loss** and **backpropagate** to **adjust the weights** of the **entire neural net**, including the MLP and RNN(s).

The **attention scores a_i** can also be used to **highlight** the **words** that influence the system's decision most.

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

RNN with deep self-attention



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

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

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

$$\dots$$

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

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

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

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

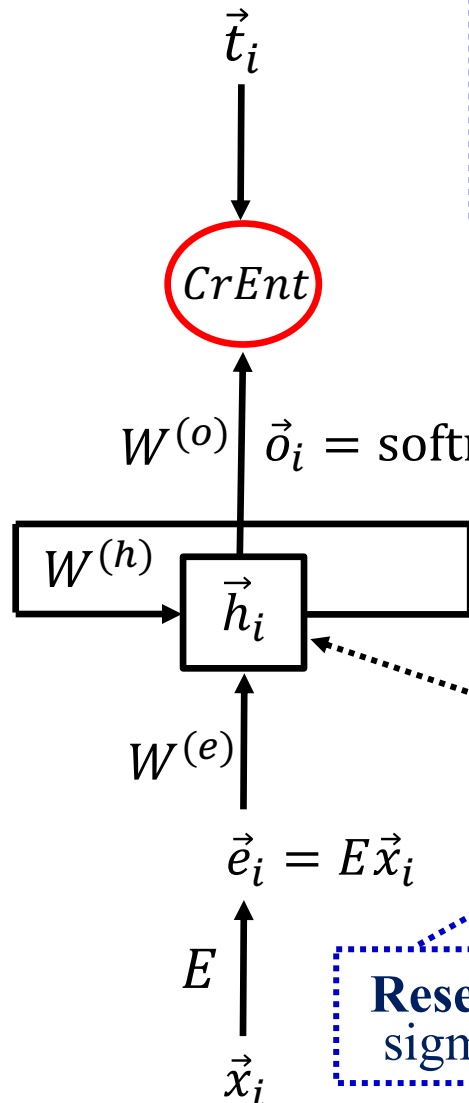
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

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

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

GRU equations, equivalent to the ones shown below.

RNN με κελί GRU (προαιρετική μελέτη)



Candidate new history (\circ denotes pairwise multiplication). For $\vec{r}_i \approx \vec{1}$, same as the \vec{h}_i of a simple RNN cell. For $\vec{r}_i \approx \vec{0}$, **forgets** \vec{h}_{i-1} and considers only the current word embedding.

New history. For $\vec{z}_i \approx \vec{0}$, same as \vec{h}_i . For $\vec{z}_i \approx \vec{1}$, ignores \vec{h}_i and **maintains** \vec{h}_{i-1} as \vec{h}_i .

GRU cell:

$$\tilde{h}_i = \tanh(\vec{r}_i \circ W^{(h)}\vec{h}_{i-1} + W^{(e)}\vec{e}_i)$$

$$\vec{h}_i = \vec{z}_i \circ \vec{h}_{i-1} + (1 - \vec{z}_i) \circ \tilde{h}_i$$

$$\vec{r}_i = \sigma(W^{(r)}\vec{h}_{i-1} + U^{(r)}\vec{e}_i)$$

$$\vec{z}_i = \sigma(W^{(z)}\vec{h}_{i-1} + U^{(z)}\vec{e}_i)$$

Reset gate (σ is the sigmoid function).

Update gate.

LSTM cells are similar, but with **more gates**. LSTM and GRU cells were actually created to address “**vanishing gradients**”. See <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Βιβλιογραφία

- Russel & Norvig (4^η έκδοση, ελληνική μετάφραση): ενότητες 21.6, 21.8.2, 24.1, 24.2.
 - Η ενότητα 24.4.1 περιγράφει μια άλλη μορφή αυτό-προσοχής που χρησιμοποιείται στους Transformers.
 - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά ολόκληρα αυτά τα κεφάλαια.
- Βλαχάβας κ.ά: ενότητα 19.9.2.
 - Δεν υπάρχουν ενότητες για όλη την ύλη αυτής της διάλεξης. Για τις εξετάσεις χρειάζεται να ξέρετε ό,τι αναφέρουν οι διαφάνειες.
 - Όσοι ενδιαφέρονται μπορούν να διαβάσουν προαιρετικά ολόκληρη την ενότητα 19.9.

Recommended reading

- M. Surdeanu and M.A. Valenzuela-Escarcega, *Deep Learning for Natural Language Processing: A Gentle Introduction*, Cambridge Univ. Press, 2024.
 - Chapters 10–13. See <https://clulab.org/gentlenlp/text.html>
 - Also available at AUEB's library.
- Jurafsky and Martin's, *Speech and Language Processing* is being revised (3rd edition) to include deep learning methods for NLP.
 - <http://web.stanford.edu/~jurafsky/slp3/>

