



Τεχνητή Νοημοσύνη

14η διάλεξη (2025-26)

Ίων Ανδρουτσόπουλος

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

Τι θα ακούσετε σήμερα

- Ενθέσεις λέξεων (word embeddings).
- Παράσταση κειμένων με κεντροειδή ενθέσεων λέξεων.
- Κατηγοριοποίηση κειμένων με MLPs και διανύσματα TF-IDF ή κεντροειδή ενθέσεων λέξεων.
- Κατηγοριοποίηση λέξεων με MLPs και κυλιόμενο παράθυρο.
- Ομαλοποίηση με μείωση βαρών (weight decay), απόρριψη (dropout), ομαλοποίηση δέσμης/επιπέδου (batch/layer normalization).

Word embeddings of business terms

(produced with word2vec, here projected to 2D using UMAP)

Vectors (points)
in 2D:


◦ $\langle 2,4 \rangle$

◦ $\langle 3,2 \rangle$

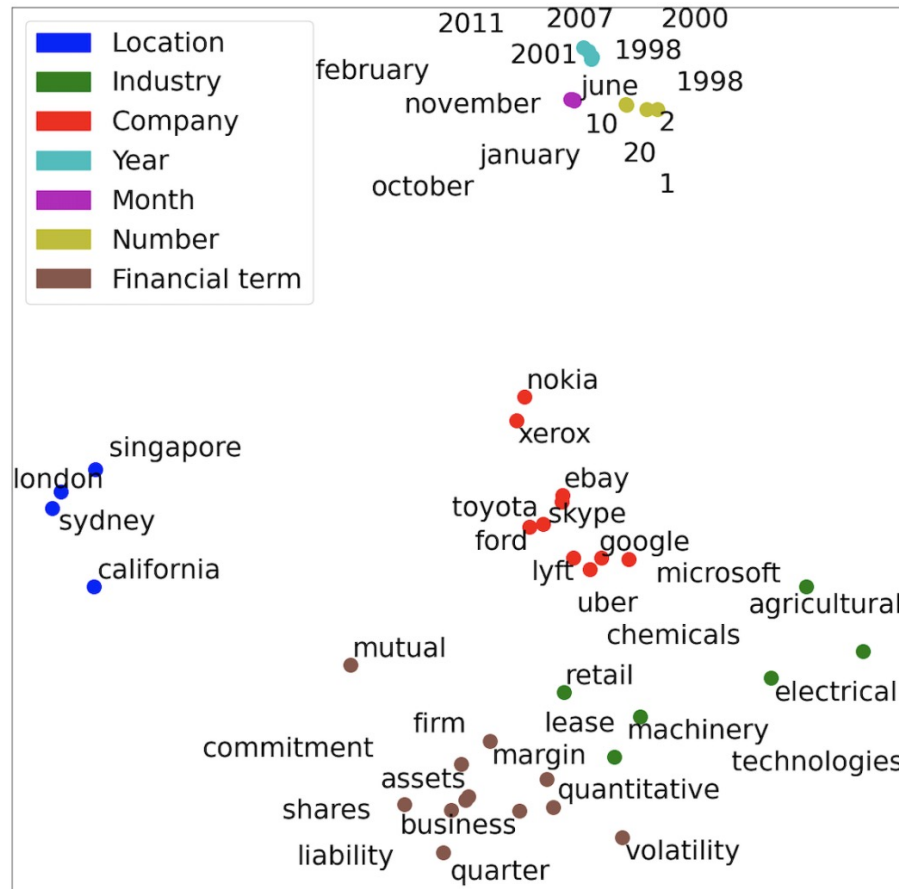
y

x

2D vector β

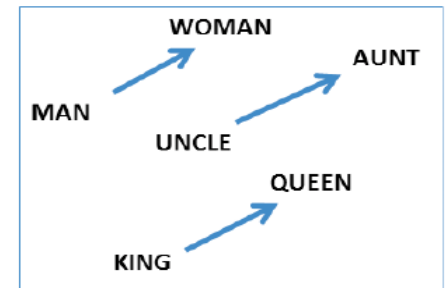
“dense layer”  $\beta = W\alpha$

300D vector α



Word embeddings are vectors (points), e.g., in a 300D space.

They capture relatedness, analogy, ...



Large image from Loukas et al., “EDGAR-CORPUS: Billions of Tokens Make The World Go Round”, EcoNLP workshop, EMNLP 2021 (<https://aclanthology.org/2021.econlp-1.2/>). Small image from Mikolov et al., “Linguistic Regularities in Continuous Space Word Representations”. NAACL 2013 (<https://aclanthology.org/N13-1090/>). For a quick intro to UMAP (and t-SNE) check: <https://www.youtube.com/watch?v=6BPI81wGGP8>.

Embeddings of biomedical terms

Table 1 Closest words to the 30 most frequent words of the BioASQ question answering task, using the cosine similarity of the dense vectors to measure proximity. Relevant (closely related) words are shown in bold, possibly relevant in normal font, and irrelevant (or misspelled) words in ~~strikeout~~.

protein	proteins	a-anchoring	pka-anchoring
thyroid	thyroidal	nonthyroid	hyperfunctioning
associated	correlated	related	correlates
hormone	gh	luteinizing	fshluteinizing
human	murine	mouse	immortalized
used	utilized	employed	applied
genes	gene	paralogs	operons
treatment	therapy	treatments	treating
disease	diseases	disease-like	mmrn1rs6532197
gene	genes	pseudogene	gene-encoding
heart	cardiac	chf	congestive
role	roles	plays	play
affect	alter	modify	impair
dna	dnas	bisulfite-treated	polymerase-mediated
histone	histones	h4k16	h4
involved	implicated	participates	regulating
list	lists	listing	to-do
proteins	protein	polypeptides	hsp70s
known	yet	presently	well-known
patients	outpatients	subjects	whom
present	this	aimed	our
cancer	cancers	crc	caner
receptor	receptors	hmc5	5-nonyloxytryptamine
regulate	modulate	regulates	orchestrate
cell	cells	cancer-cell	sw1710
coding	5-noncoding	5-untranslated	3-noncoding
inhibitors	inhibitor	small-molecule	atp-competing
many	several	some	numerous
related	linked	associated	relate
cardiomyopathy	cardiomyopathies	myocardiopathy	dcm

See <http://bioasq.org/news/bioasq-releases-continuous-space-word-vectors-obtained-applying-word2vec-pubmed-abstracts>

Ενθέσεις λέξεων (word embeddings)

- **Ενθέσεις λέξεων:**
 - Σχετικές λέξεις απεικονίζονται σε **κοντινά διανύσματα**.
 - Τα διανύσματα συνήθως είναι **πυκνά** (ελάχιστα μηδενικά), με **100-300 διαστάσεις**.
 - **Ενώ σε 1-hot διανύσματα λέξεων, έχουμε τόσες διαστάσεις όσο το μέγεθος του λεξιλογίου και μόνο μία συνιστώσα (αυτή που αντιστοιχεί στη συγκεκριμένη λέξη) είναι μη μηδενική.**
- **Κατασκευή ενθέσεων λέξεων.**
 - Υπάρχουν **εργαλεία** που παράγουν ενθέσεις λέξεων από **μεγάλα σώματα κειμένων** (π.χ. Wikipedia).
 - Π.χ. **Word2vec** (<https://code.google.com/archive/p/word2vec/>), **GloVe** (<https://nlp.stanford.edu/projects/glove/>).
 - **Εναλλακτικά μπορούμε να μάθουμε (ή να τροποποιήσουμε) ενθέσεις λέξεων με δικά μας νευρωνικά δίκτυα** (βλ. παρακάτω).

Κεντροειδή ενθέσεων λέξεων

- Μπορούμε να παραστήσουμε **κάθε κείμενο T** (ακολουθία λέξεων) w_1, \dots, w_d ως το **κεντροειδές των ενθέσεων λέξεων** του κειμένου:

$$\vec{T} = \frac{1}{d} \sum_{i=1}^d \vec{w}_i = \frac{\sum_{j=1}^{|V|} \vec{w}_j \cdot TF(w_j, T)}{\sum_{j=1}^{|V|} TF(w_j, T)}$$

- Η λαμβάνοντας υπόψη και τις **τιμές IDF** των λέξεων:

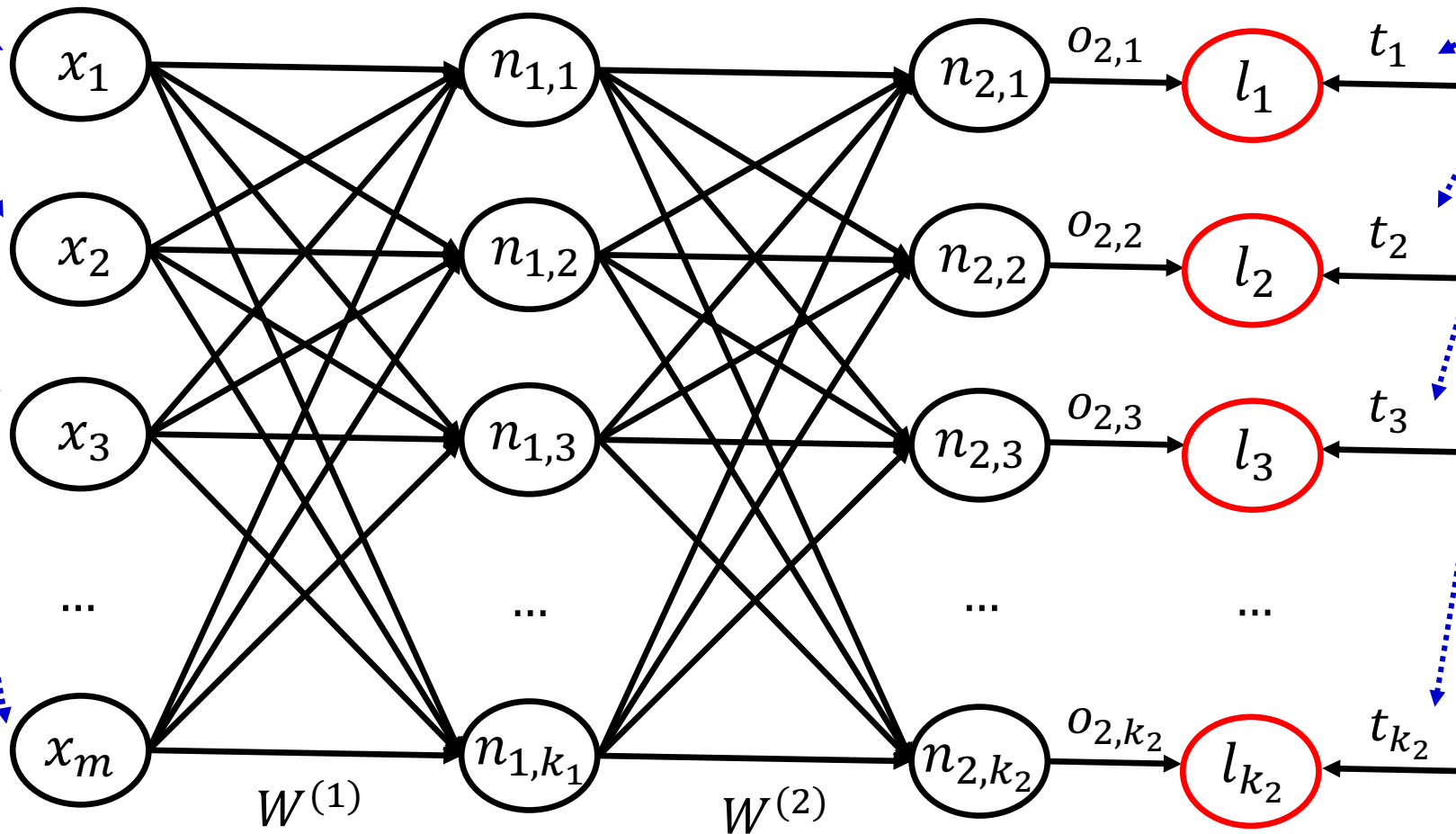
$$\vec{T} = \frac{\sum_{j=1}^{|V|} \vec{w}_j \cdot TF(w_j, T) \cdot IDF(w_j)}{\sum_{j=1}^{|V|} TF(w_j, T) \cdot IDF(w_j)}$$

- Μπορούμε να **χρησιμοποιήσουμε τα κεντροειδή ως διανύσματα χαρακτηριστικών** των κειμένων.
- Θα δούμε καλύτερους τρόπους (π.χ. με RNNs) αργότερα...

Single-label multi-class classification revisited

Centroid of word embeddings of the text
($\vec{x} = \vec{T}$) or TF-IDF vector

Correct output ($t_j = 1$ means the
single correct class is the j -th one)



$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

$$\vec{o}^{(2)} = \text{softmax}(W^{(2)}\vec{o}^{(1)})$$

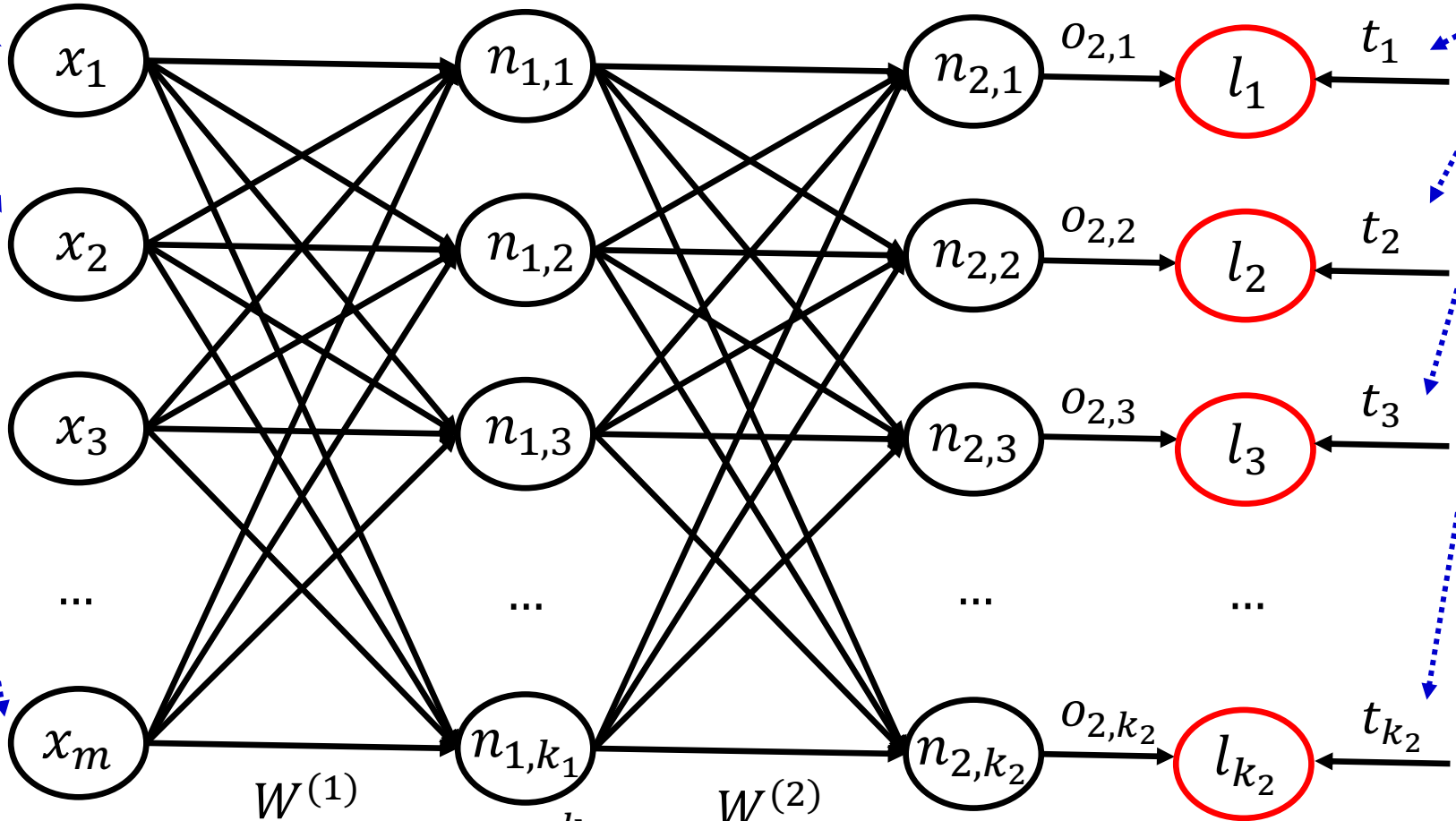
“Categorical” cross
entropy loss at the
current training instance.

$$l = - \sum_{j=1}^{k_2} t_j \log(o_{2,j})$$

Multi-label multi-class classification revisited

Centroid of word embeddings of the text ($\vec{x} = \vec{T}$) or TF-IDF vector

Correct output ($t_j = 1$ means the j -th class is **one** of the correct ones)



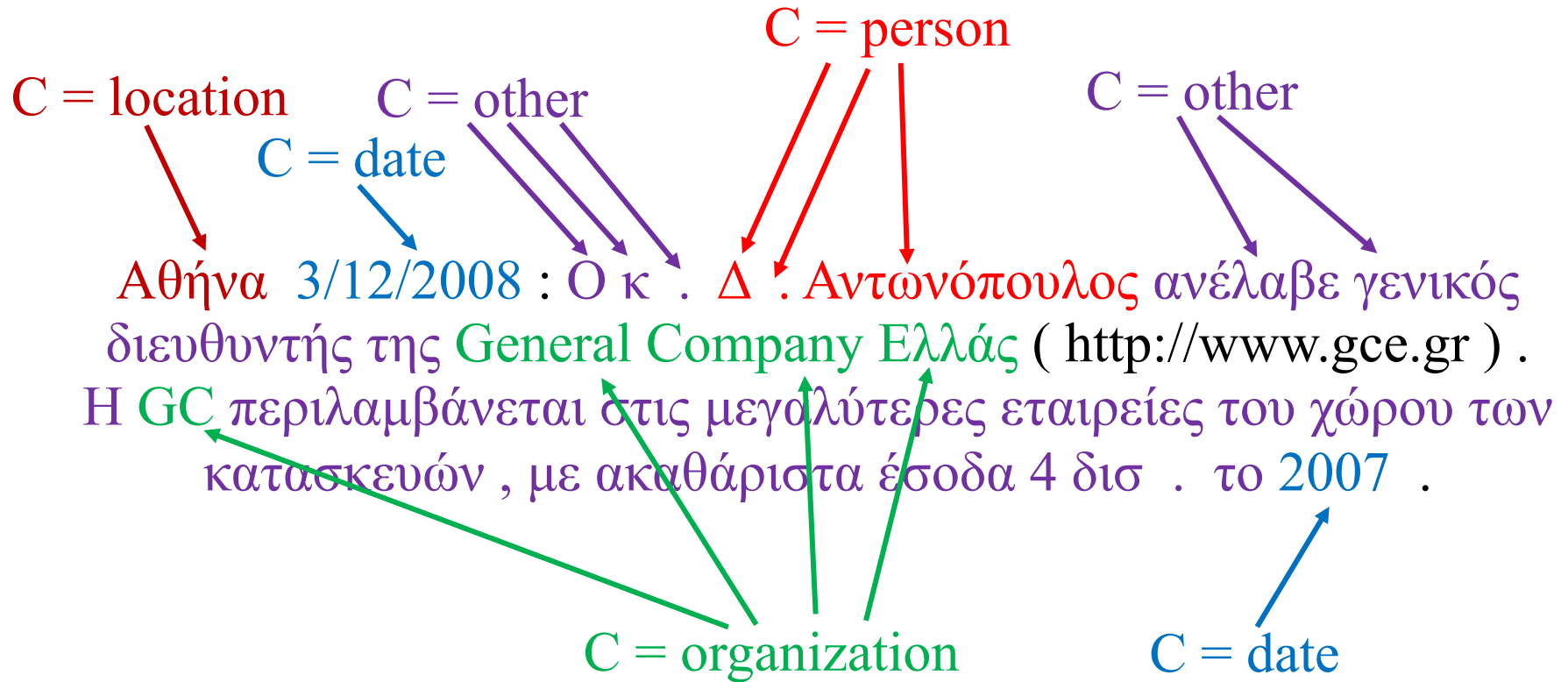
$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

$$\vec{o}^{(2)} = \sigma(W^{(2)}\vec{o}^{(1)})$$

$$l = - \sum_{j=1}^{k_2} t_j \log(o_{2,j}) + (1 - t_j) \log(1 - o_{2,j})$$

Sum of the **binary cross entropy loss** for each class.

Αναγνώριση ονομάτων οντοτήτων



Εργαλειοθήκη Python για ελληνικά κείμενα (αναγνώριση ονομάτων οντοτήτων, μερών του λόγου, μορφολογική και συντακτική ανάλυση, μετατροπή Greeklish σε Ελληνικά): <https://aclanthology.org/2025.coling-demos.17/>
Πολύγλωσσες εργαλειοθήκες ΕΦΓ: βλ. π.χ. <https://spacy.io/>,
<http://www.nltk.org/>, <https://stanfordnlp.github.io/stanza/>.

Κατηγοριοποίηση λέξεων (token classification)

- Αναγνώριση **ονομάτων οντοτήτων**.
 - C = other (άλλη λεκτική μονάδα)
 - C = person (λεκτική μονάδα ονόματος προσώπου)
 - C = organization (λεκτική μονάδα ονόματος οργανισμού)
 - C = location (λεκτική μονάδα ονόματος τοποθεσίας)
 - C = date (λεκτική μονάδα ημερομηνίας)
 - ...
- Αναγνώριση **μερών του λόγου**. Για κάθε λέξη:
 - Κατηγορίες: C = άρθρο, C = ρήμα, C = επίθετο, ...
 - Παραδείγματα εκπαίδευσης: περιπτώσεις εμφάνισης άρθρων, ρημάτων, επιθέτων κ.λπ. με τις τιμές των ιδιοτήτων.
- Πολλές άλλες εφαρμογές σε επισημείωση ακολουθιών:
 - Π.χ. γονιδίων, ενδείξεων αισθητήρων.

Εξαγωγή στοιχείων συμφωνητικών

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 κυλιόμενο παράθυρο

Λέξη που θέλουμε να κατατάξουμε.

Παράθυρο 3 λέξεων (συνήθως μεγαλύτερο).

yesterday language **tech** announced that...

$$\vec{x}_{i-1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad \vec{x}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad \vec{x}_{i+1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}$$

1-hot διανύσματα ($|V| \times 1$) των λέξεων του παραθύρου. $|V|$ το μέγεθος του λεξιλογίου (π.χ. $|V| = 200.000$).

Ενθέσεις ($d \times 1$) των λέξεων του παραθύρου. d το πλήθος διαστάσεων των ενθέσεων (π.χ. $d = 300$).

$$\vec{e}_{i-1} = \begin{bmatrix} 1.8 \\ 2.3 \\ -1.4 \\ 3.7 \\ \dots \\ -1.1 \end{bmatrix} \quad \vec{e}_i = \begin{bmatrix} 2.4 \\ -3 \\ 9.3 \\ 5.1 \\ \dots \\ 3.9 \end{bmatrix} \quad \vec{e}_{i+1} = \begin{bmatrix} 2.2 \\ 3.8 \\ 1.2 \\ -6.4 \\ \dots \\ 7.1 \end{bmatrix}$$

E : πίνακας ($d \times |V|$) που περιέχει τις ενθέσεις όλων των λέξεων του λεξιλογίου ως στήλες. Τότε:
 $\vec{e}_{i-1} = E\vec{x}_{i-1}$, $\vec{e}_i = E\vec{x}_i$, ...

Κατηγοριοποίηση λέξεων με κυλιόμενο παράθυρο

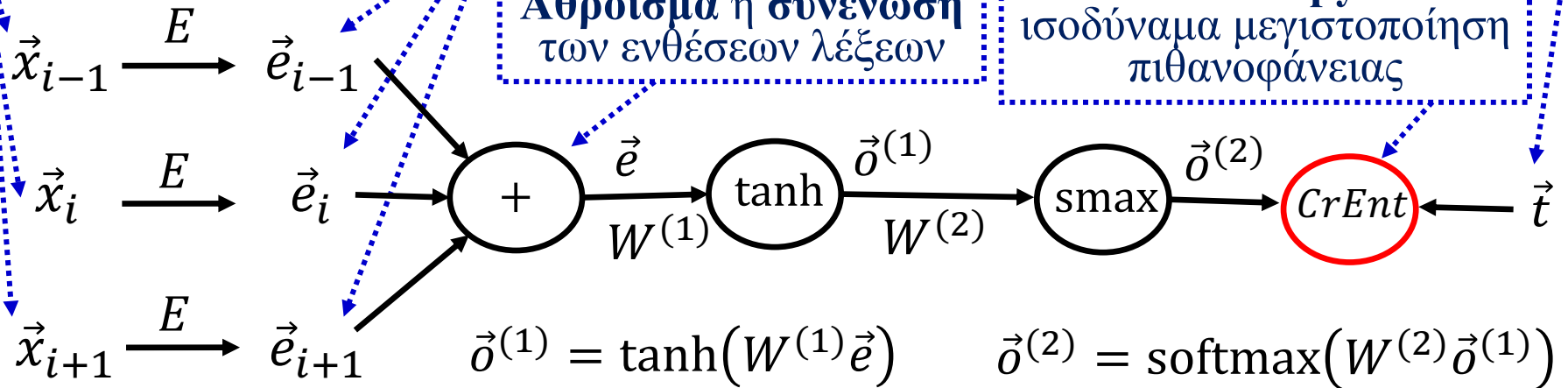
1-hot διανύσματα των λέξεων του παραθύρου

Ενθέσεις των λέξεων του παραθύρου

Σωστή έξοδος (κατηγορία λέξης) ως 1-hot διάνυσμα

Άθροισμα ή συνένωση των ενθέσεων λέξεων

Cross-entropy loss: ισοδύναμα μεγιστοποίηση πιθανοφάνειας



Κατηγοριοποίηση λέξεων με κυλιόμενο παράθυρο

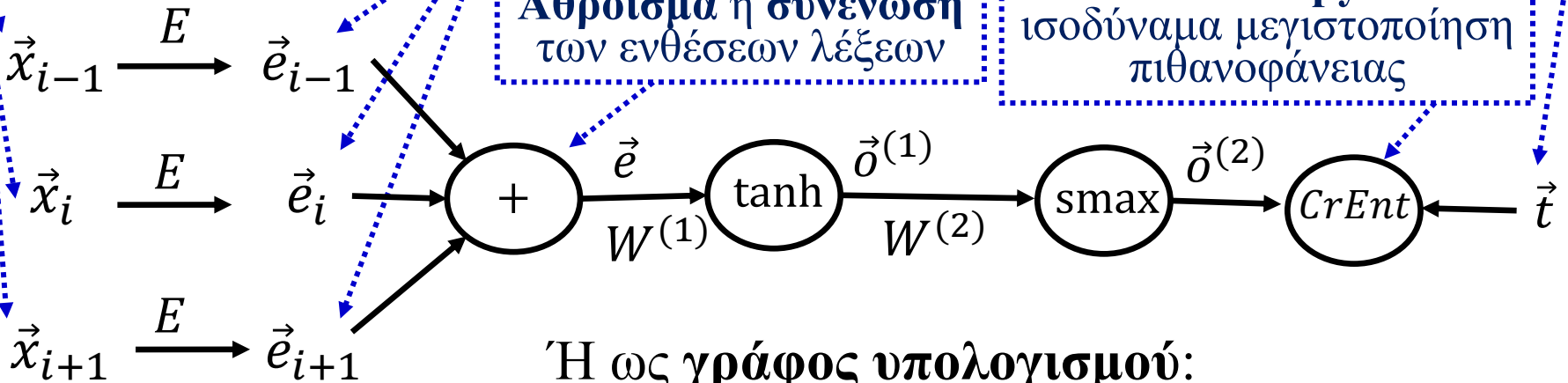
1-hot διανύσματα των λέξεων του παραθύρου

Ενθέσεις των λέξεων του παραθύρου

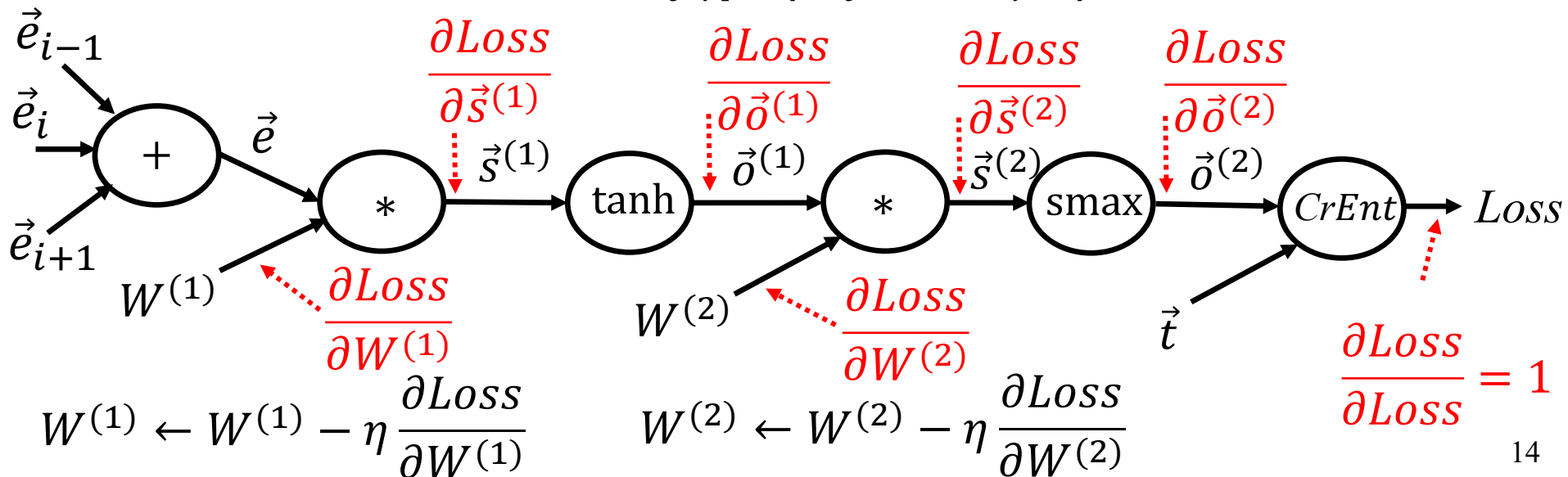
Σωστή έξοδος (κατηγορία λέξης) ως 1-hot διάνυσμα

Άθροισμα ή συνένωση των ενθέσεων λέξεων

Cross-entropy loss: ισοδύναμα μεγιστοποίηση πιθανοφάνειας



Ή ως γράφος υπολογισμού:



Κατηγοριοποίηση λέξεων με κυλιόμενο παράθυρο

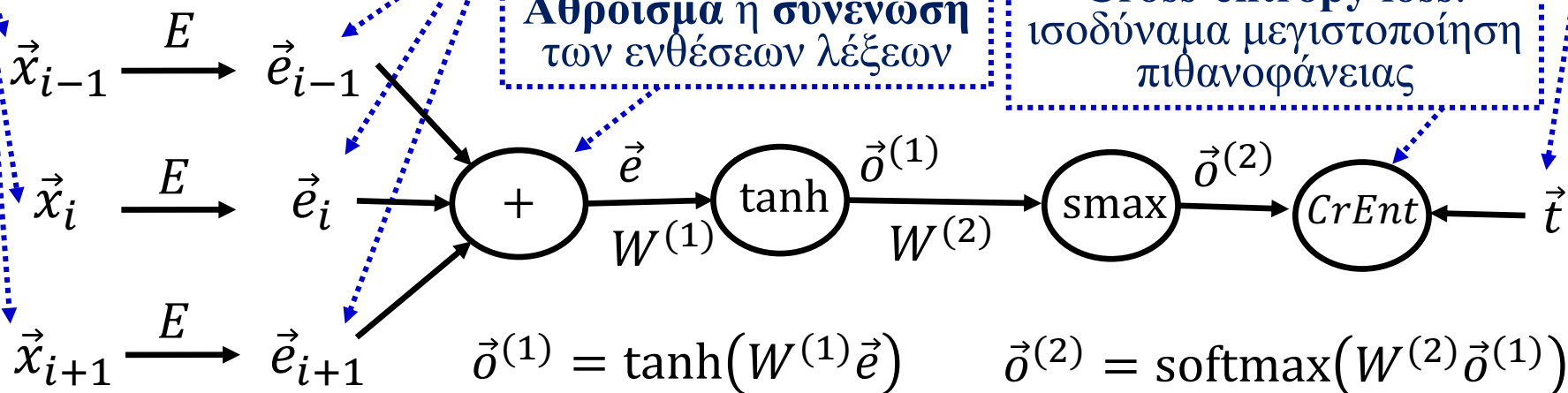
1-hot διανύσματα των λέξεων του παραθύρου

Ενθέσεις των λέξεων του παραθύρου

Σωστή έξοδος (κατηγορία λέξης) ως 1-hot διάνυσμα

Άθροισμα ή σύνενωση των ενθέσεων λέξεων

Cross-entropy loss: ισοδύναμα μεγιστοποίηση πιθανοφάνειας



Μαθαίνουμε τα βάρη $W^{(1)}, W^{(2)}$ του νευρωνικού δικτύου με **ανάστροφη μετάδοση**. Αν έχουμε πάρα πολλά παραδείγματα εκπαίδευσης, μπορούμε να μάθουμε και τις **ενθέσεις λέξεων**, δηλ. τον **πίνακα E** με **ανάστροφη μετάδοση**! Αλλά συχνά δεν έχουμε τόσα πολλά παραδείγματα, οπότε μαθαίνουμε τις ενθέσεις λέξεων με ξεχωριστό εργαλείο (π.χ. Word2vec).

Μπορούμε να χρησιμοποιήσουμε και **απλούστερο ταξινομητή** (π.χ. λογιστικής παλινδρόμησης) με **διάνυσμα χαρακτηριστικών το \vec{e}** .

Ομαλοποίηση βαρών (weight decay)

- Όπως στη λογιστική παλινδρόμηση, αντί για το $l(\vec{w})$ μπορούμε να **ελαχιστοποιήσουμε** το:

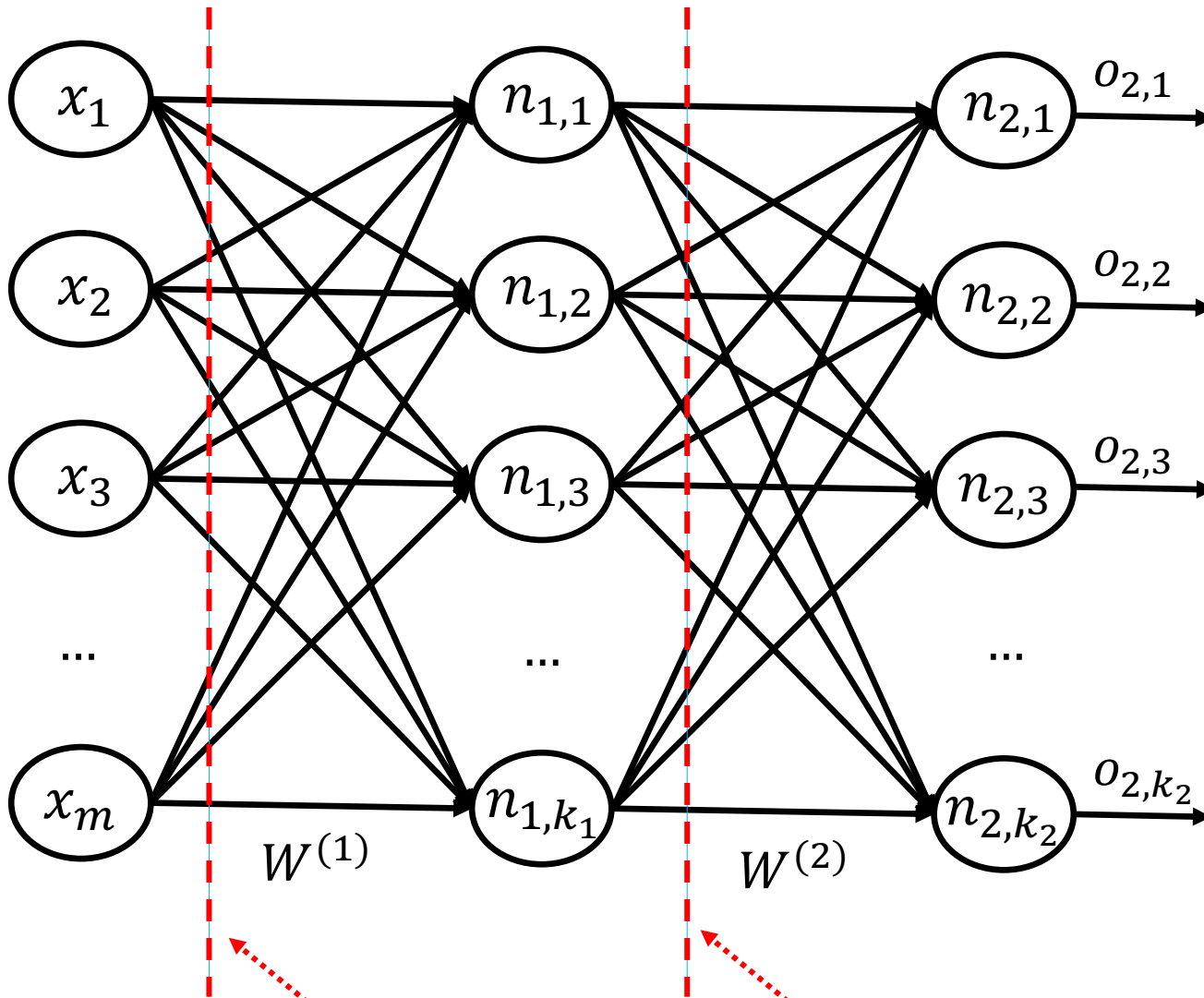
$$l(\vec{w}) + \lambda \|\vec{w}\|^2 = l(\vec{w}) + \lambda \sum_{l=0}^n w_l^2$$

L2 regularization (“ridge regression”)

δηλ. επιβραβεύουμε υποψήφια \vec{w} με **πολλά μικρά βάρη**.

- Υπάρχει έτσι **μικρότερος κίνδυνος υπερ-εφαρμογής**.
 - Αν πολλά βάρη $w_{i,j}$ είναι πολύ μικρά, οι αντίστοιχες τιμές (ιδιότητες) από προηγούμενους νευρώνες i στους νευρώνες j ουσιαστικά δεν χρησιμοποιούνται. Με λιγότερες ιδιότητες έχουμε **μικρότερο κίνδυνο υπερ-εφαρμογής**.
 - $\lambda > 0$. Η τιμή επιλέγεται με δοκιμές σε δεδομένα επικύρωσης.
 - Παραλλαγή: το **L1 regularization** προσθέτει $-\lambda \sum_{l=0}^n |w_l|$. **Οδηγεί σε πιο αραιά \vec{w}** (με πολλά μηδενικά).

Dropout



Dropout at the input layer.
E.g. $p_{drop} = 0.2$.

Dropout at the output of a hidden
layer. E.g., $p_{drop} = 0.5$.

Dropout

- **For each training instance** (or mini-batch), we **drop** (remove) **each neuron** of the layer where dropout is applied with **probability** $p_{drop} = 1 - p_{keep}$.
 - Helps the neural net **avoid relying too much on particular neurons** (or inputs). A form of **regularization**. Works well!
 - **Gradients** also **do not flow** through dropped neurons.
 - Alternative explanation: we train an **ensemble** of networks, containing **all the pruned network versions** dropout creates.
- **We don't drop neurons during testing** (only during training).
 - One of the reasons we need to **specify in our code** if the network is in **training or inference (testing) mode**.

Dropout – continued

- **During testing, we multiply the output** of each neuron (of the layer where dropout was applied) by p_{keep} , so that the neuron's **expected output value** will be **as in training**.

- **Expected output during training with dropout** of a neuron that would output y without dropout:

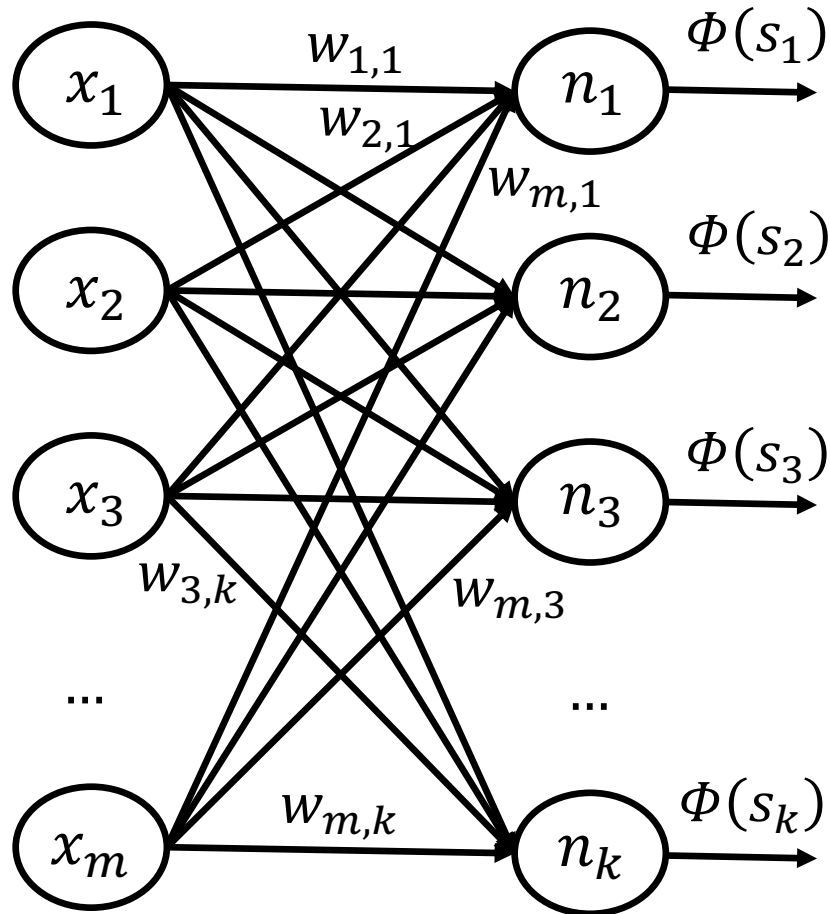
$$p_{drop} \cdot 0 + p_{keep} \cdot y = p_{keep} \cdot y$$

- **Adjusted output during testing** (inference):

$$p_{keep} \cdot y$$

- **Or we divide** the output by p_{keep} **during training**, and we do nothing during testing.
- **Done automatically** by most dropout implementations.

Batch normalization



At each layer, instead of:

$$s_j = \sum_{i=1}^m w_{i,j} x_i$$

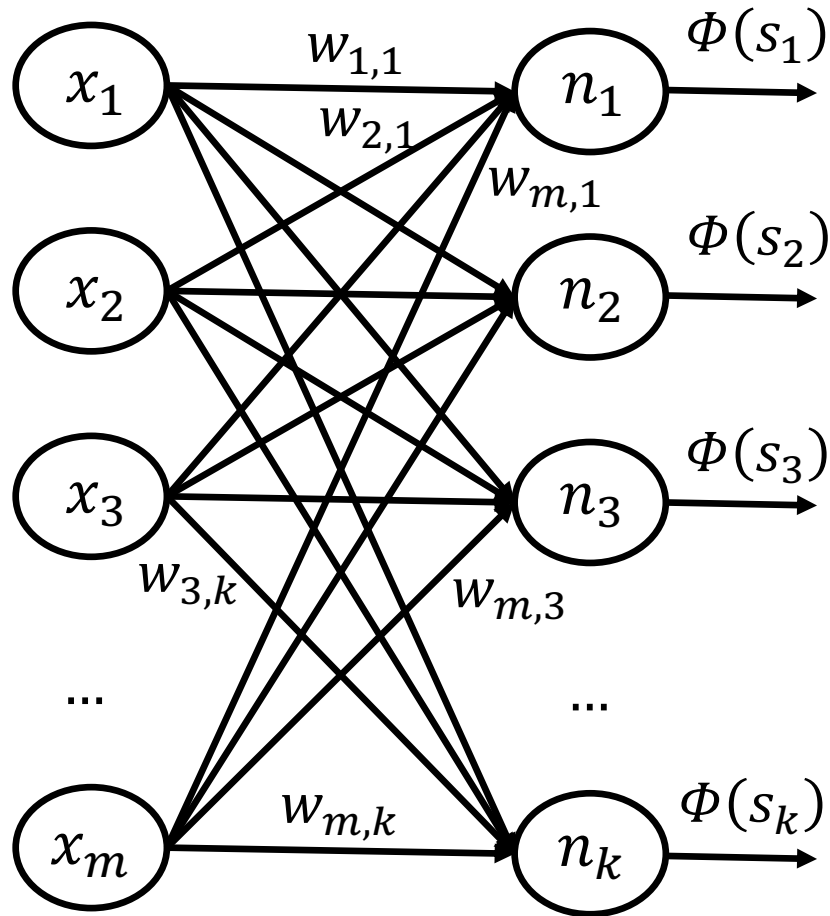
we use:

$$\bar{s}_j = \frac{g_j}{\sigma_j} (s_j - \mu_j) + b_j$$

- μ_j, σ_j are the **mean** and **std. dev. of s_j** in the **mini-batch**.
- g_j, b_j are **learned** parameters (constant after training).
- Φ now applied to \bar{s}_j .

Assuming s_j follows a **normal distribution**, we **shift** the distribution (by subtracting μ_j) so that its mean will be at 0, and **adjust its width** (dividing by σ_j) to make it standard ($\mu = 0, \sigma = 1$). We want $\Phi(\mathbf{s})$ to operate around $\mathbf{s} = \mathbf{0}$, where it has **non-linear behavior**.

Layer normalization



At each layer, instead of:

$$s_j = \sum_{i=1}^m w_{i,j} x_i$$

we use:

$$\bar{s}_j = \frac{g_j}{\sigma} (s_j - \mu) + b_j$$

- μ, σ are the **mean** and **std. dev. of s_1, \dots, s_k** in the layer.
- g_j, b_j are **learned** parameters (constant after training).
- Φ applied to \bar{s}_j .

See <https://arxiv.org/pdf/1607.06450.pdf> for **batch vs. layer normalization**. The latter works better for RNNs and Transformers.

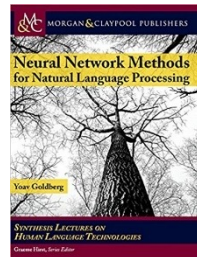
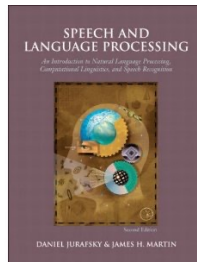
Βιβλιογραφία

- Russel & Norvig (4^η έκδοση, ελληνική μετάφραση): ενότητες 21.4.2, 21.5.3, 21.5.4, 24.1.
- Βλαχάβας κ.ά.: ίδιες ενότητες με την προηγούμενη διάλεξη.
- Δείτε και την πρόσθετη προτεινόμενη βιβλιογραφία της 19ης διάλεξης.
- Δείτε προαιρετικά και τις διαφάνειες των μεταπτυχιακών μαθημάτων του ΟΠΑ «Επεξεργασία Φυσικής Γλώσσας» (ΠΜΣ «Επιστήμη Υπολογιστών» και «Αναλυτική Κειμένων» (ΠΜΣ «Επιστήμη Δεδομένων» (βλ. e-class).

With dropout, batch/layer normalization, residuals (to be discussed) and other additions, strictly speaking we no longer have an “MLP”. Some people prefer “**Feed Forward Neural Network**” (FFNN), but “MLP” still often used as synonym.

Βιβλιογραφία – συνέχεια

- Όσοι ενδιαφέρονται για την επεξεργασία φυσικής γλώσσας (και φωνής) αξίζει να μελετήσουν σταδιακά το βιβλίο «Speech and Language Processing» των D. Jurafsky and J.H. Martin, 2^η έκδοση, Prentice Hall, 2008.
 - Υπάρχει στη βιβλιοθήκη του ΟΠΑ.
 - Διατίθεται ελεύθερα η υπό προετοιμασία 3^η έκδοση. Βλ. <http://web.stanford.edu/~jurafsky/slp3/>.
- Μια καλή εισαγωγή στη χρήση βαθιάς μάθησης για επεξεργασία φυσικής γλώσσας είναι το βιβλίο του Y. Goldberg «Neural Network Models for Natural Language Processing», Morgan & Claypool Publishers, 2017.
 - Υπάρχει στη βιβλιοθήκη του ΟΠΑ.



Other recommended resources

- M. Surdeanu and M.A. Valenzuela-Escarcega, *Deep Learning for Natural Language Processing: A Gentle Introduction*, Cambridge Univ. Press, 2024.
 - Chapters 5–9.
 - <https://clulab.org/gentlenlp/text.html>
 - Also available at AUEB's library.
- Stanford's *NLP with Deep Learning* course.
 - Course material: <http://web.stanford.edu/class/cs224n/>
 - Videos: available on YouTube.
- J. Johnson's (U Michigan) *Deep Learning for Computer Vision* course.
 - See: <https://www.youtube.com/playlist?list=PL5-TkQAfAZFbzxjBHtzdVCWE0Zbhong7r>

