

MAI4CAREU

Master programmes in Artificial
Intelligence 4 Careers in Europe



Πανεπιστήμιο Κύπρου - Τεχνητή Νοημοσύνη

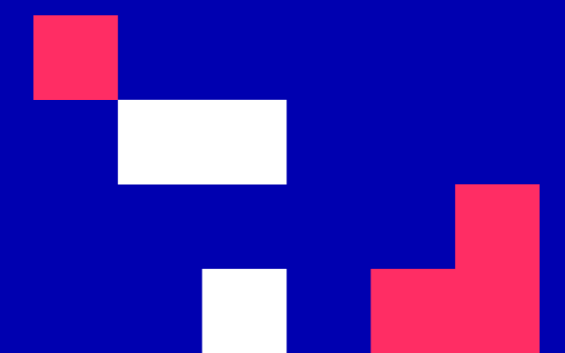
MAI612 - ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Διάλεξη 11: Νευρωνικά Δίκτυα 3: Εισαγωγή στη
Βαθιά Μάθηση

Βασίλης Βασιλειάδης, PhD
Χειμερινό Εξάμηνο 2022/23



CYENS
CENTRE OF EXCELLENCE



MAI4CAREU

Master programmes in Artificial
Intelligence 4 Careers in Europe



Επανάληψη





Νευρωνικά δίκτυα

- Τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ) είναι μοντέλα εμπνευσμένα από τον ανθρώπινο εγκέφαλο
- Αποτελείται από πολλά απλά στοιχεία επεξεργασίας που ονομάζονται τεχνητοί νευρώνες
- Οι τεχνητοί νευρώνες υπολογίζουν το σταθμισμένο άθροισμα των εισόδων τους και το τροφοδοτούν μέσω μιας λειτουργίας ενεργοποίησης η οποία στη συνέχεια γίνεται η έξοδος τους.
- Λειτουργίες ενεργοποίησης:
 - Heaviside βήμα (μη διαφοροποιήσιμο) → Μοντέλο Perceptron
 - Γραμμική → γραμμική παλινδρόμηση
 - Sigmoid → Logistic regression
- Τα Perceptrons είναι γραμμικοί ταξινομητές
 - Με το συνδυασμό πολλαπλών perceptrons σε στρώματα μπορούμε να ταξινομήσουμε μη γραμμικά διαχωριζόμενα προβλήματα
 - Π.χ., το XOR μπορεί να λυθεί χρησιμοποιώντας 2 κρυμμένους νευρώνες και 1 νευρώνα εξόδου
 - Ωστόσο, δεν μπορούμε να τους εκπαιδεύσουμε χρησιμοποιώντας gradient descent όταν χρησιμοποιούν τη συνάρτηση βήματος Heaviside, καθώς είναι μη διαφοροποιήσιμο





Νευρωνικά δίκτυα

- Τα Multilayer Perceptrons είναι συνώνυμα των feedforward ΤΝΔ (συνήθως με διαφορίσιμες λειτουργίες ενεργοποίησης)
- Ως ταξινομητής, ένα MLP με:
 - 1 κρυφές φόρμες ανοιχτές ή κυρτές περιοχές απόφασης
 - 2 κρυμμένα στρώματα δημιουργούν αυθαίρετες περιοχές λήψης αποφάσεων
- Η προωθημένη διάδοση είναι η διαδικασία που τροφοδοτεί ένα παράδειγμα δεδομένων στο επίπεδο εισόδου ενός NN και αυτό μετατρέπεται σταδιακά στην πρόβλεψη εξόδου (αναστροφή ή ταξινόμηση) μέσω κάποιου μη γραμμικού μετασχηματισμού.
 - Αυτός ο μη γραμμικός μετασχηματισμός υπολογίζει τα χαρακτηριστικά της εισόδου που μαθαίνονται
 - Ένα δεύτερο κρυφό στρώμα υπολογίζει τα χαρακτηριστικά γνωρίσματα ως λειτουργίες των υπαρχόντων χαρακτηριστικών
- Η μάθηση σε NNs μπορεί να γίνει χρησιμοποιώντας οπισθοδιάδοση και gradient descent





Νευρωνικά δίκτυα

- Οπίσθια διάδοση: ένας αποτελεσματικός τρόπος υπολογισμού μερικών παραγώγων της συνάρτησης σφάλματος σε σχέση με κάθε παράμετρο χρησιμοποιώντας τον κανόνα της αλυσίδας (δεδομένου ότι ένα NN είναι μια σύνθεση συναρτήσεων)
- Λειτουργία λάθους: ΠΣΕ για παλινδρόμηση, διασταυρούμενη εντροπία για ταξινόμηση
- Η προωθημένη διάδοση υπολογίζει την ενεργοποίηση (εξόδου) κάθε κόμβου, ενώ η οπίσθια διάδοση υπολογίζει το σφάλμα (δέλτα) κάθε κόμβου
- Δέλτα (σφάλμα) κάθε κόμβου: $A \times B$
 - A = παράγωγος της λειτουργίας ενεργοποίησης του κόμβου
 - B = παράγωγος του σφάλματος σε σχέση με την παραγωγή του κόμβου
 - Για έναν κόμβο εξόδου: αυτό είναι το παράγωγο της συνάρτησης σφάλματος wrt στην ενεργοποίηση του κόμβου εξόδου
 - Για έναν κρυφό κόμβο i στο στρώμα k : αυτό είναι το άθροισμα όλων των δέλτα σε κόμβους στο στρώμα $k + 1$ (που συνδέονται με τον κόμβο i) πολλαπλασιασμένο επί το βάρος σύνδεσης τους
- Gradient του σφάλματος wrt σε βάρος = (δέλτα μετασυναπτικού κόμβου) \times (εκροή προσυναπτικού κόμβου)





Νευρωνικά δίκτυα

- Στοχαστικό GD: ενημέρωση βάρους μετά την παρουσίαση κάθε μοτίβου
- Batch GD: ενημέρωση βάρους μετά την παρουσίαση όλων των μοτίβων στο σύνολο εκπαίδευσης
- Mini-batch GD: ενημέρωση βάρους μετά την παρουσίαση υποσυνόλων προτύπων στο σύνολο κατάρτισης
- Ορισμός ορμής: η μνήμη της προηγούμενης κατεύθυνσης, επιταχύνει τη μάθηση
- Πρόωρη διακοπή: τρόπος αποφυγής υπερπροσαρμογής διακόπτοντας την εκπαίδευση όταν το σφάλμα επικύρωσης αρχίσει να αυξάνεται
- Μπορούμε να βελτιώσουμε την απόδοση των μοντέλων ΝΔ χρησιμοποιώντας ρύθμιση, υπερπαραμέτρων και ensembles
- Η εκμάθηση της τοπολογίας ΝΔ μπορεί να γίνει χρησιμοποιώντας μεθόδους χωρίς gradient, όπως εξελικτικούς αλγόριθμους.





Διάλεξη 11: Νευρωνικά Δίκτυα 3: Εισαγωγή στη βαθιά μάθηση

Μαθησιακά αποτελέσματα

Θα μάθετε για:

1. Τι είναι η βαθιά μάθηση και γιατί να την χρησιμοποιήσετε
2. Δίκτυα συνέλιξης για δεδομένα εικόνας
3. Χειρισμός διαδοχικών δεδομένων
4. Επαναλαμβανόμενα νευρωνικά δίκτυα
5. Οπισθοδιάδοση μέσα στο χρόνο
6. Το πρόβλημα της εξαφάνισης και της έκρηξης του gradient
7. Δίκτυα Echo State
8. Δίκτυα Long short-term memory (LSTM)
9. Word embeddings για το χειρισμό κειμένου





Από τη μηχανική μάθηση στη βαθιά μάθηση

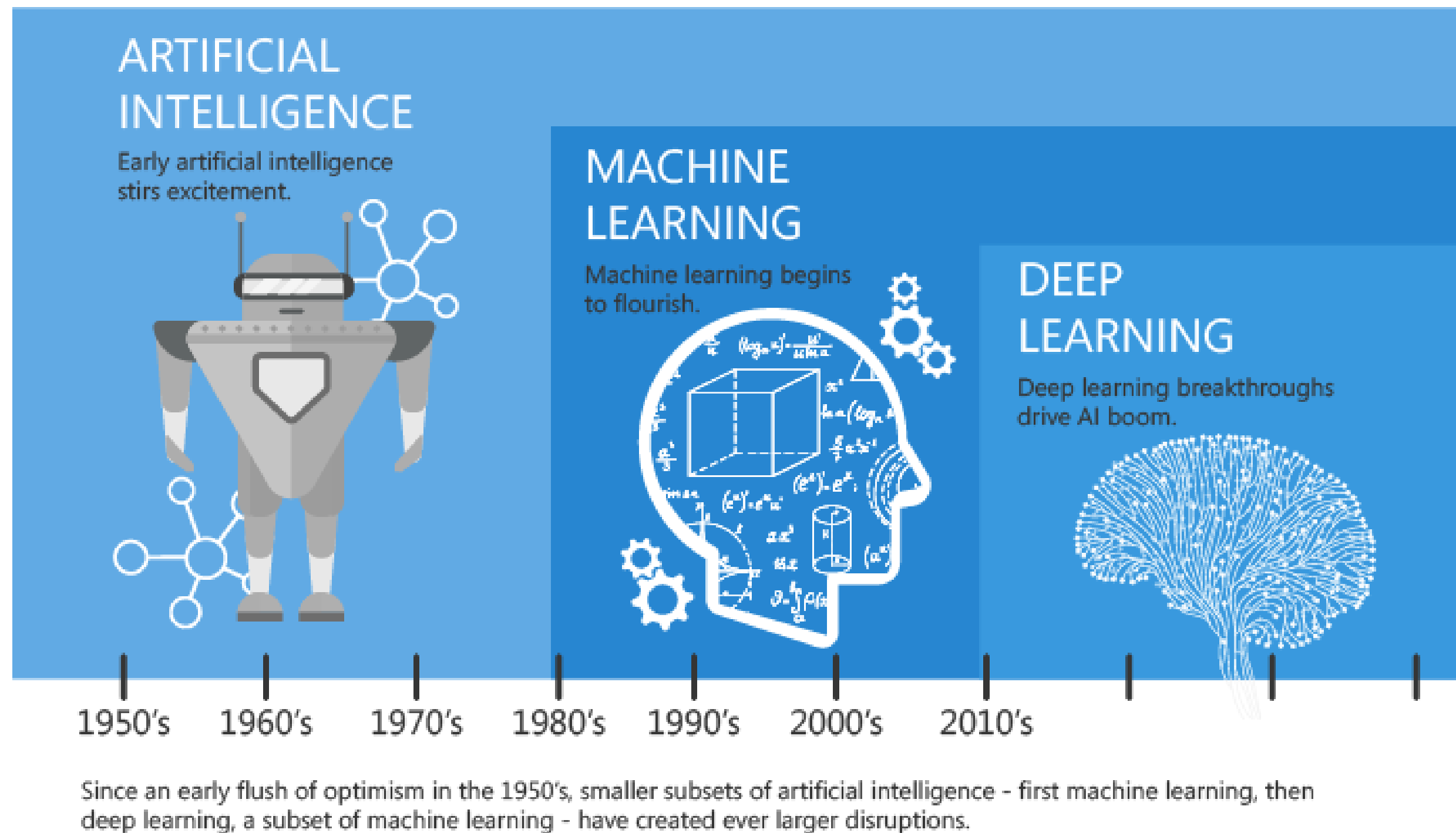
- Το μοντέλο MM μετατρέπει τα δεδομένα εισόδου σε ουσιαστικές εξόδους
 - Μαθαίνοντας χρησιμοποιώντας παραδείγματα
- Πώς να μεταμορφώσετε σημαντικά τα δεδομένα;
 - Πώς να μάθετε **αναπαραστάσεις** των δεδομένων εισόδου που μας φέρνουν πιο κοντά στην αναμενόμενη έξοδο
- **Αναπαράσταση**: διαφορετικός τρόπος για να εξετάσετε τα δεδομένα - για να αναπαραστήσετε ή να κωδικοποιήσετε τα δεδομένα
- Για παράδειγμα:
 - Η έγχρωμη εικόνα μπορεί να κωδικοποιηθεί σε μορφή RGB (κόκκινο-πράσινο-μπλε) ή HSV (τιμή κορεσμού)
 - Τα σημεία σε ένα επίπεδο μπορούν να αναπαρασταθούν με τη χρήση συντεταγμένων καρτεσιανών (x,y) ή πολικών (r,θ)
- Ορισμένες εργασίες μπορεί να είναι δύσκολες με τη μία εκπροσώπηση και ευκολότερη με μια άλλη. Για παράδειγμα:
 - Η εργασία: «επιλέξτε όλα τα κόκκινα εικονοστοιχεία στην εικόνα» απλούστερη με μορφή RGB
 - Η εργασία: «κάντε την εικόνα λιγότερο κορεσμένη» απλούστερη σε μορφή HSV
- Τα μοντέλα MM είναι όλα σχετικά με την **εύρεση κατάλληλων αναπαραστάσεων** για τα δεδομένα εισόδου τους - μετασχηματισμοί των δεδομένων που τα καθιστούν πιο διαχειρίσιμα για το συγκεκριμένο έργο (π.χ. ταξινόμηση)

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Τι είναι η βαθιά μάθηση



[ΠΗΓΗ](#)

- Η βαθιά μάθηση είναι ένα υποπεδίο της ML που επικεντρώνεται στην εκμάθηση διαδοχικών στρωμάτων σημαντικών αναπαραστάσεων χρησιμοποιώντας νευρωνικά δίκτυα.
- «Βαθιά» = Διαδοχικά στρώματα αναπαράστασης
- Άλλα πιθανά ονόματα: πολυεπίπεδες αναπαραστάσεις μάθησης, ιεραρχικές αναπαραστάσεις μάθησης
- Πόσα στρώματα θεωρούνται «βαθιά»;
 - 2 επίπεδα αναπαραστάσεων: ρηχή μάθηση
 - > 2 στρώματα: βαθιά μάθηση
 - Συχνά: δεκάδες ή εκατοντάδες διαδοχικές αναπαραστάσεις

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Γιατί βαθιά μάθηση; Γιατί τώρα;

1. Υλικό

- 1990-2010: Οι επεξεργαστές έγιναν γρηγορότεροι με έναν συντελεστή ~5000
- Καθ' όλη τη δεκαετία του 2000, εταιρείες όπως η NVIDIA και η AMD επενδύουν σε μεγάλο βαθμό στην ανάπτυξη γρήγορων, μαζικά παράλληλων chips (μονάδες επεξεργασίας γραφικών [GPUs]) για να τροφοδοτήσουν τα γραφικά των ολοένα και πιο φωτορεαλιστικών βιντεοπαιχνιδιών
- 2007: Η NVIDIA εγκαινίασε τη διεπαφή προγραμματισμού CUDA για τη σειρά GPUs της
- Μικρός αριθμός GPU αντικαθιστούν μαζικές συστάδες CPU σε διάφορες πολύ παράλληλες εφαρμογές (π.χ. μοντελοποίηση φυσικής)
- 2011-2012: Πρώτες υλοποιήσεις CUDA των ΝΔ που κέρδισαν διαγωνισμούς (π.χ. ImageNet)
- Η βιομηχανία βαθιάς μάθησης αρχίζει να υπερβαίνει τις GPU (π.χ. Tensor Processing Units [TPUs])

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Γιατί βαθιά μάθηση; Γιατί τώρα;

2. Σύνολα δεδομένων και δείκτες αναφοράς

- Η τεχνητή νοημοσύνη είναι η νέα βιομηχανική επανάσταση
 - Αν η τεχνητή νοημοσύνη είναι η ατμομηχανή αυτής της επανάστασης, τότε τα δεδομένα είναι το κάρβουνο της
- Εκθετική πρόοδος στο υλικό αποθήκευσης τα τελευταία 20 χρόνια
- Αλλαγή δεδομένων: η άνοδος του Διαδικτύου, καθιστά εφικτή τη συλλογή και διανομή πολύ μεγάλων συνόλων δεδομένων για ML
- Σήμερα: οι εταιρείες συνεργάζονται με σύνολα δεδομένων εικόνας, σύνολα δεδομένων βίντεο και σύνολα δεδομένων φυσικής γλώσσας που δεν θα μπορούσαν να συλλεχθούν χωρίς το διαδίκτυο
- ImageNet σύνολο δεδομένων: Καταλύτης για την άνοδο της βαθιάς μάθησης
 - 1,4 εκατομμύρια εικόνες σχολιάζονται με 1000 κατηγορίες εικόνων
 - Ετήσιος διαγωνισμός
- Δημόσιοι διαγωνισμοί (π.χ., Kaggle): εξαιρετικός τρόπος για να παρακινηθούν οι ερευνητές να προωθήσουν την εξέλιξη της τεχνολογίας

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Γιατί βαθιά μάθηση; Γιατί τώρα;

3. Αλγοριθμικές εξελίξεις

- Μέχρι τη δεκαετία του 2000, μας έλειπαν αξιόπιστοι τρόποι για να εκπαιδεύσουμε βαθιά ΝΔ
- Τα ΝΔ ήταν ρηχά (1-2 κρυμμένα στρώματα), αδυνατώντας να ανταγωνιστούν τα SVM ή τα τυχαία δάση
- Το βασικό θέμα: το gradient ξεπεράστηκε καθώς ο αριθμός των στρωμάτων αυξήθηκε
- 2009-2016: Οι αλγοριθμικές βελτιώσεις επιτρέπουν καλύτερη διάδοση του gradient:
 - Καλύτερες **λειτουργίες ενεργοποίησης** για νευρικά στρώματα
 - Καλύτερα **συστήματα αρχικοποίησης βάρους**, ξεκινώντας από την προκατάρτιση σε επίπεδο στρώματος, η οποία εγκαταλείφθηκε
 - Καλύτερα **συστήματα βελτιστοποίησης** (π.χ., RMSProp, Adam)
 - **Ομαλοποίηση παρτίδας**
 - **Υπολειπόμενες συνδέσεις**
- Ωστόσο, παλαιότερες τεχνικές όπως τα LSTM ήταν πλέον εφικτές λόγω υλικού και δεδομένων

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Γιατί βαθιά μάθηση; Γιατί τώρα;

4. Ένα νέο κύμα επενδύσεων

- Η βαθιά μάθηση (BM) έγινε η νέα αιχμή της τεχνολογίας για την υπολογιστική όραση το 2012-2013 και τελικά για όλες τις αντιληπτικές εργασίες - οι ηγέτες της βιομηχανίας το παρατήρησαν
- Το σταδιακό κύμα επενδύσεων της βιομηχανίας στη συνέχεια, πολύ πέρα από οτιδήποτε είχε δει προηγουμένως στην ιστορία της τεχνητής νοημοσύνης. Για παράδειγμα, η συνολική επένδυση επιχειρηματικού κεφαλαίου στην τεχνητή νοημοσύνη:
 - 2011 (πριν από τη δημοφιλία της βαθιάς μάθησης) = ~ \$19 εκατομμύρια
 - 2014: ~ \$394M
- 2013: Η Google εξαγόρασε την DeepMind για 500 εκατομμύρια δολάρια (μεγαλύτερη εξαγορά μιας startup AI)
- 2014: Η Baidu ξεκίνησε ένα ερευνητικό κέντρο DL στη Silicon Valley (επενδύοντας 300 εκατομμύρια δολάρια)
- Το Deep Learning έγινε κεντρικό στοιχείο της στρατηγικής προϊόντων των τεχνολογικών κολοσσών
- Η έρευνα και ο αριθμός των ατόμων που εργάζονται/με βαθιά μάθηση αυξήθηκαν εκθετικά τα τελευταία χρόνια

Προσαρμοσμένο από: Chollet, F. (2018). Deep Learning with Python. Manning Publications.





Γιατί βαθιά μάθηση; Γιατί τώρα;

5. Ο εκδημοκρατισμός της βαθιάς μάθησης

- Οι πρώτες ημέρες απαιτούσαν σημαντική τεχνογνωσία C++ και CUDA, την οποία λίγοι άνθρωποι κατείχαν
- Σήμερα, οι βασικές δεξιότητες σεναρίου Python αρκούν για να κάνουν προηγμένη έρευνα DL
- Αυτό οφείλεται κυρίως στην ανάπτυξη των Theano, [TensorFlow](#) και [PyTorch](#): συμβολικός tensor¹—πλαίσια χειραγώγησης για την Python που υποστηρίζουν την αυτοδιαφοροποίηση
- Αύξηση των φιλικών προς το χρήστη βιβλιοθηκών, όπως η [Keras](#), η οποία καθιστά την DL εύκολη
- Πολιτισμός σχετικά με τη μεταφόρτωση εγγράφων στο arXiv και την απελευθέρωση του κώδικα ως ανοικτού κώδικα (βλ. [paperswithcode](#))

¹Tensor: Generalization of a matrix to N-dimensional space

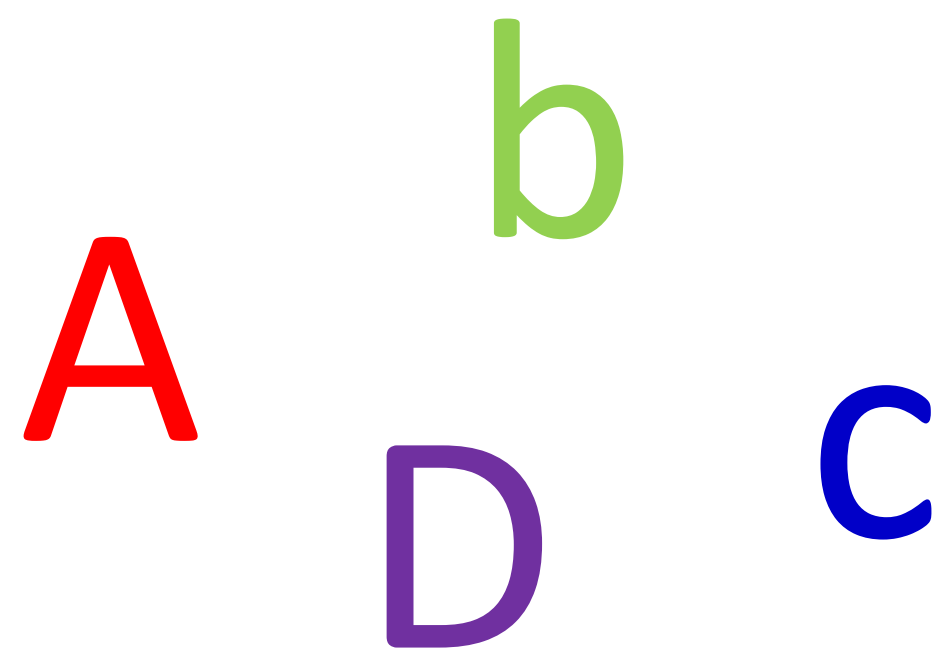




Η δομή βοηθά στη μάθηση

Εάν τα δεδομένα έχουν κάποια δομή και το NN δεν χρειάζεται να μάθει τη δομή από το μηδέν, τότε το NN θα αποδώσει καλύτερα

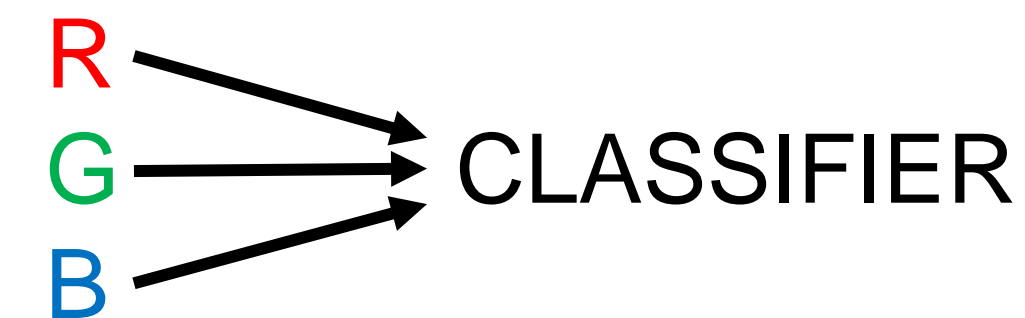
- **Παράδειγμα:** Ταξινόμηση γραμμάτων



Το χρώμα
δεν έχει
σημασία!

Τι νομίζετε ότι θα ήταν πιο εύκολο για έναν ταξινομητή να μάθει;

1. Ένα μοντέλο που χρησιμοποιεί την έγχρωμη εικόνα;



2. Ένα μοντέλο που βλέπει μόνο την κλίμακα του γκρι;

$$\frac{R+G+B}{3} \rightarrow \text{CLASSIFIER}$$





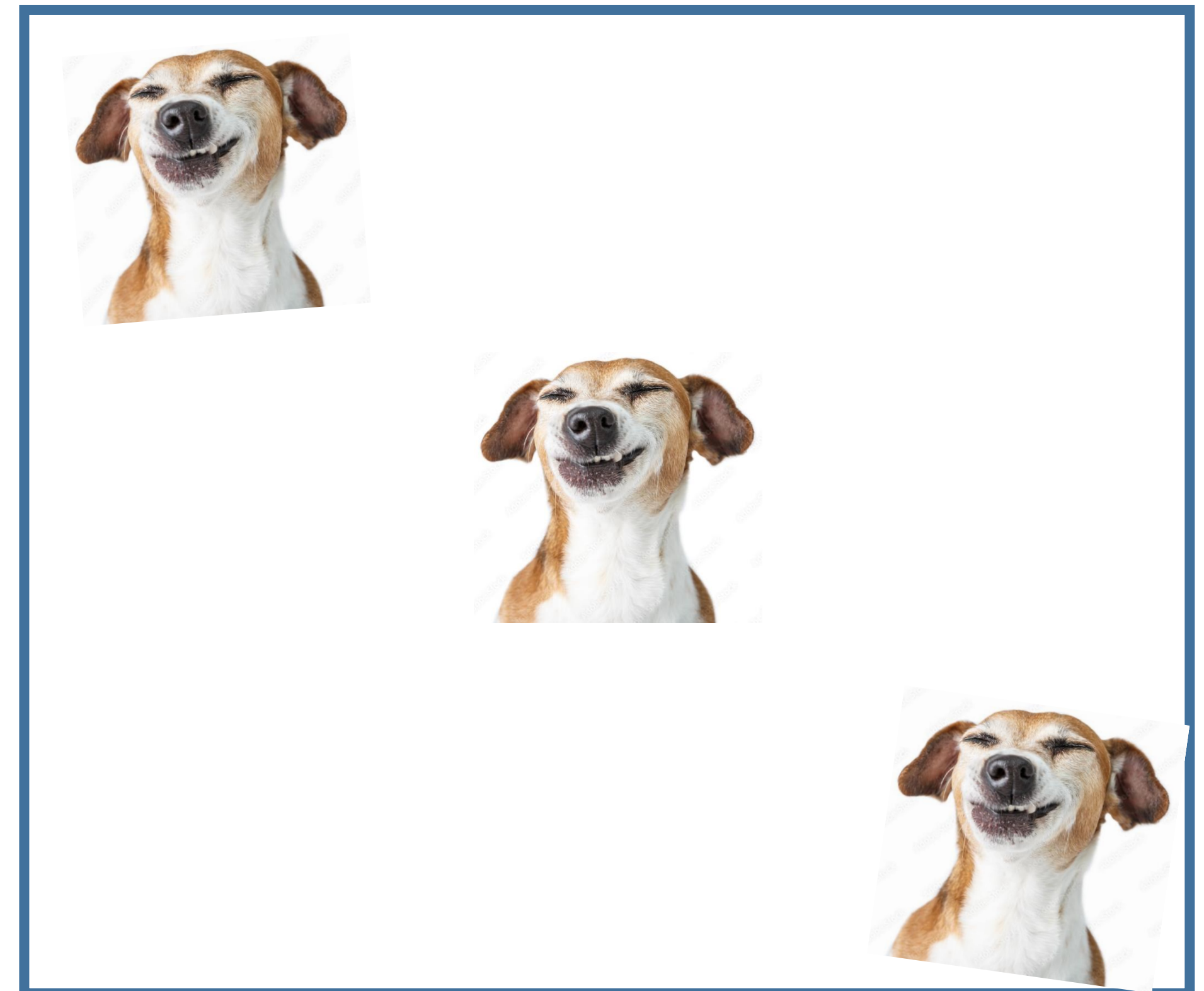
Statistical Invariance

Παράδειγμα: Ταξινόμηση εικόνας σκύλου

- Δεν έχει σημασία πού είναι ο σκύλος, είναι ακόμα μια εικόνα με ένα σκυλί

Θα πρέπει να διδάξουμε ρητά ένα ΝΔ ότι τα αντικείμενα στις εικόνες είναι τα ίδια ανεξάρτητα από το πού βρίσκονται στην εικόνα.

Αυτό ονομάζεται **Translation Invariance**





Statistical Invariance

Παράδειγμα: Κείμενο που μιλάει για σκύλους

Ο γρήγορος καφέ σκύλος πήδηξε πάνω από τον τεμπέλη λευκό σκύλο.

Μια φορά κι έναν καιρό, υπήρχε ένας σκύλος με μυτερά αυτιά και μια μικρή ουρά.

Αλλάζει η έννοια του σκύλου ανάλογα με το αν είναι στην 1^η πρόταση ή στη 2^η;

➤ Όχι

Θέλουμε το τμήμα του ΝΔ που μαθαίνει τι είναι ένας σκύλος να επαναχρησιμοποιείται κάθε φορά που το ΝΔ βλέπει τη λέξη σκύλος, και όχι να το ξαναμάθει κάθε φορά.

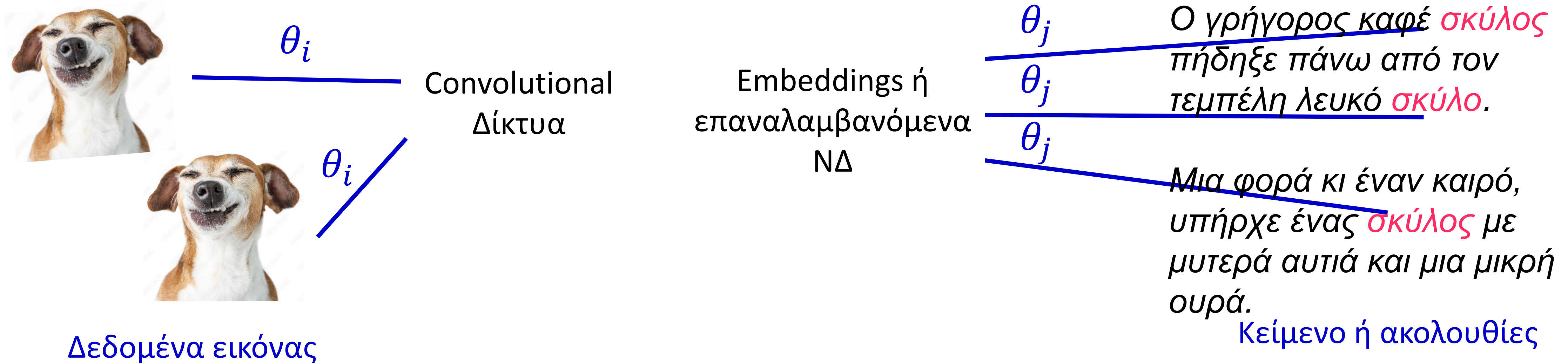




Statistical Invariance: Κατανομή βάρους

Όταν γνωρίζουμε ότι 2 ή περισσότερες εισροές περιέχουν τις ίδιες πληροφορίες, τότε μοιραζόμαστε τα βάρη τους και εκπαιδεύουμε τα βάρη από κοινού για αυτές τις εισροές

Statistical invariants: πράγματα που δεν αλλάζουν κατά μέσο όρο σε χρόνο ή χώρο





Δεδομένα εικόνας





Δεδομένα εικόνας

Παράδειγμα: Εργασία ταξινόμησης

- Είσοδος: 256x256x3 (RGB) → tensor
- Έξοδος: Γάτα ή όχι γάτα

Ας υποθέσουμε ότι χρησιμοποιούμε ένα MLP με 1 πλήρως συνδεδεμένο (ΠΣ) κρυμμένο στρώμα 100 νευρώνων

- 196.608×100 (εισαγωγή σε κρυφό) + 100 (κρυμμένο στην έξοδο) = 19.660,900 παράμετροι!

Μειονεκτήματα:

- Τα MLPs απαιτούν ένα τεράστιο ποσό των παραμέτρων (επιρρεπής σε υπερπροσαρμογή)
- Τα MLPs δεν προσμετρούν το translation invariance





Convolutional Networks (ConvNets)

Τα Convolutional NNs **είναι** τακτοποιημένες εκδόσεις των MLPs, δηλαδή μειώνουν τον αριθμό των παραμέτρων και προσπαθούν να λογοδοτήσουν για την αμετάβλητη μετάφραση, με βάση 3 ιδέες:

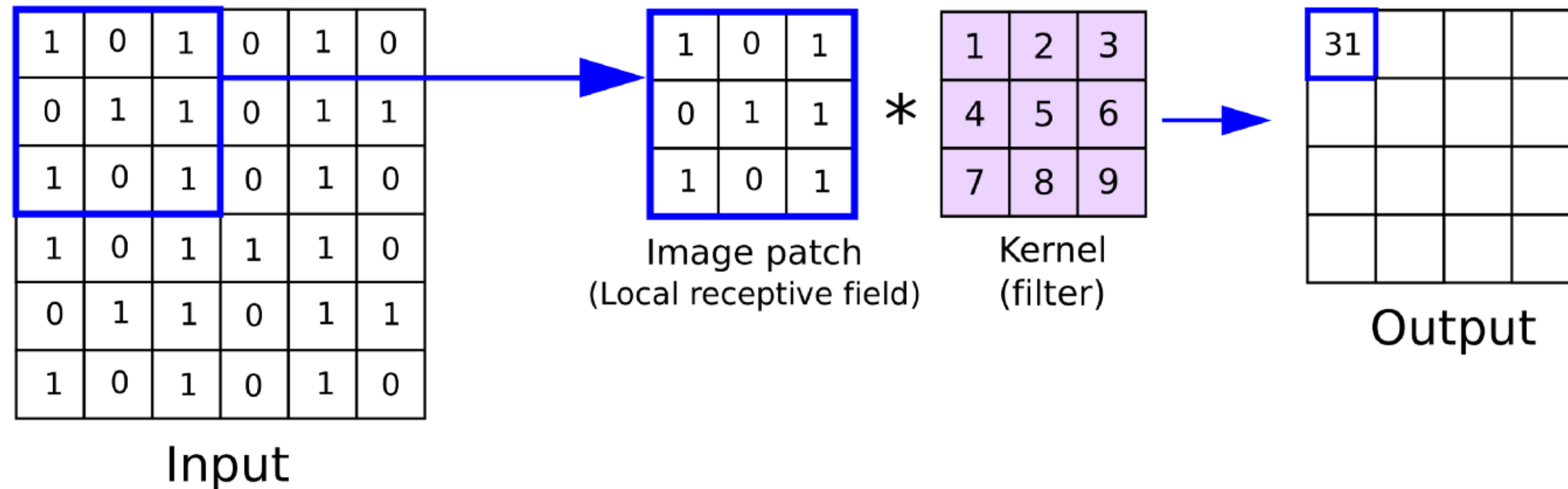
- **Local receptive fields**
 - Κάθε νευρώνας λαμβάνει είσοδο μόνο από μια περιορισμένη περιοχή του προηγούμενου στρώματος (συνήθως ένα τετράγωνο, π.χ., 5x5)
 - Οι μαθησιακές παράμετροι κάθε νευρώνα ονομάζονται **φίλτρο** ή **πυρήνας**.
 - Αντίθετα, το δεκτικό πεδίο των στρωμάτων FC είναι ολόκληρο το προηγούμενο στρώμα
- **Κατανομή βάρους**
 - Κάθε φίλτρο **συνελίσσεται** με το προηγούμενο στρώμα, δηλαδή φέτες πάνω από τον προηγούμενο ολόκληρο χάρτη ενεργοποίησης (π.χ. την αρχική εικόνα) και υπολογίζει το εσωτερικό γινόμενο μεταξύ των καταχωρήσεων φίλτρου και του patch εισόδου
 - Η έξοδος αυτής της λειτουργίας είναι ένας χάρτης ενεργοποίησης για κάθε φίλτρο
- **Spatial sub-sampling (pooling)[προαιρετικό]**
 - Μειώνει τις διαστάσεις των δεδομένων συνδυάζοντας τις εξόδους των νευρωνικών συστάδων σε έναν μόνο αριθμό
 - Τυπικά, μέγιστη συγκέντρωση ή μέση συγκέντρωση





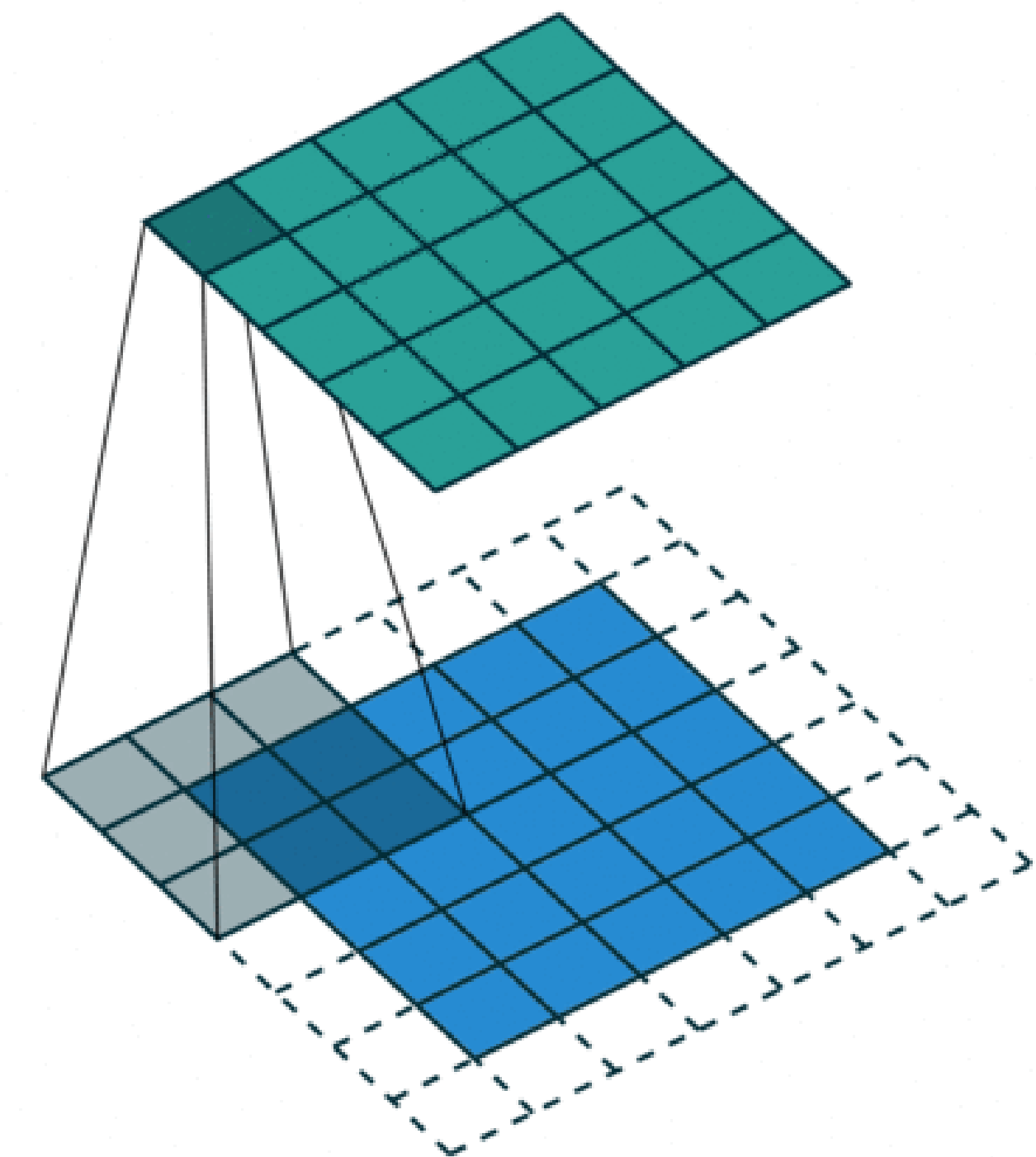
Convolutional Networks (ConvNets)

- Συνέλιξη



Local receptive field: 3x3 φίλτρο

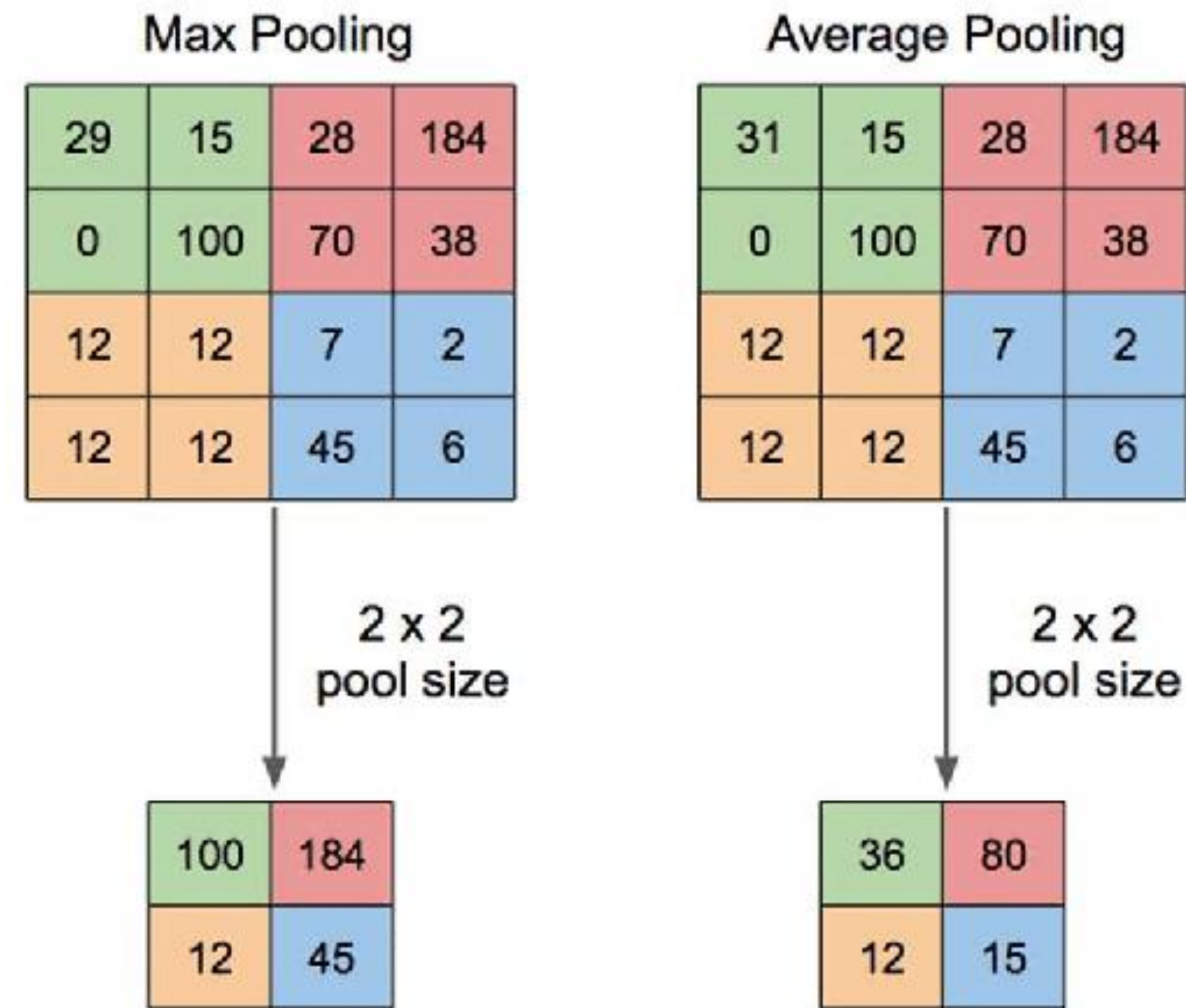
Κατανομή βάρους: μόνο 9 παράμετροι έχουν μάθει να παράγουν το χάρτη ενεργοποίησης εξόδου (από αυτό το φίλτρο)





Convolutional Networks (ConvNets)

- Pooling

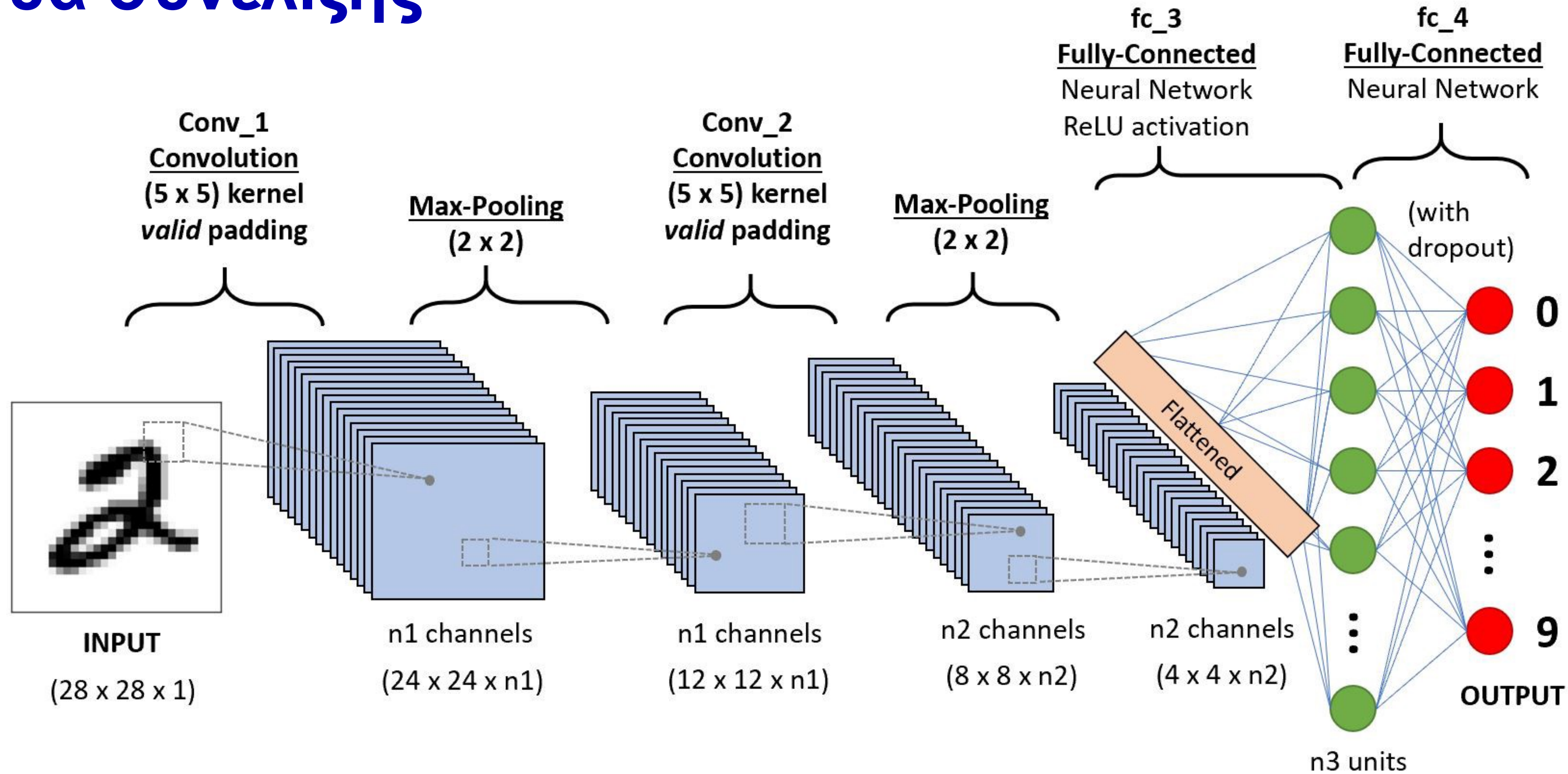


[ΠΗΓΗ](#)





Δίκτυα συνέλιξης





Τι μαθαίνουν τα ConvNets

Κάθε επίπεδο στην ιεραρχία μαθαίνει σταδιακά καλύτερες αναπαραστάσεις

- **Στρώμα εισαγωγής:** αρχικές τιμές εικονοστοιχείων
- **Στρώμα 1:** παρουσία/απουσία άκρων ιδιαίτερου προσανατολισμού σε συγκεκριμένη θέση
- **Στρώμα 2:** τοπικές ρυθμίσεις των άκρων
- **Στρώμα 3:** συνελεύσεις τοπικών ρυθμίσεων που μπορεί να αντιστοιχούν σε μέρη αντικειμένων
- **Στρώμα 4:** συνδυασμοί μερών αντικειμένων που μπορεί να αντιπροσωπεύουν πλήρη αντικείμενα
- **Στρώμα παραγωγής:** ταξινόμηση





Εφαρμογές ConvNets

Κάθε εργασία που απαιτεί δεδομένα εικόνας:

- Αναγνώριση εικόνας
- Αναγνώριση βίντεο
- Ταξινόμηση εικόνας
- Τμηματοποίηση εικόνας
- Ιατρική ανάλυση εικόνας
- ...

Μπορεί επίσης να χρησιμοποιηθεί για άλλους τύπους δεδομένων:

- Επεξεργασία φυσικής γλώσσας
- Πρόβλεψη χρονοσειρών
- ...





Διαδοχικά δεδομένα





Διαδοχικά δεδομένα

- Τα διαδοχικά δεδομένα αναφέρονται σε δεδομένα που περιέχουν στοιχεία που **ταξινομούνται σε ακολουθίες**
- Η σειρά διαφοροποιεί αυτά τα δεδομένα από άλλες περιπτώσεις που έχουμε δει μέχρι τώρα
- Για παράδειγμα:
 - Η τιμή ενός αποθέματος με την πάροδο του χρόνου
 - Περιβαλλοντικά δεδομένα (πίεση, θερμοκρασία, καθίζηση κ.λπ.) με την πάροδο του χρόνου
 - Ακολουθία ερωτημάτων σε μια μηχανή αναζήτησης ή η συχνότητα ενός ερωτήματος με την πάροδο του χρόνου
 - Σήματα EEG
 - Μια αλληλουχία νουκλεοτιδίων DNA
 - Οι λέξεις σε ένα έγγραφο όπως εμφανίζονται με τη σειρά
 - Συμβάντα σε αρχείο καταγραφής
 - Βιντεοαναγνώριση δράσης





Δεδομένα χρονοσειρών

- Χρονοσειρά:
 - Τυπική περίπτωση διαδοχικών δεδομένων
 - Διάνυσμα αριθμητικών τιμών που αλλάζουν με την πάροδο του χρόνου
 - Συνήθως, συλλέγονται σε τακτά χρονικά διαστήματα
- **Συστατικά:**
 - **Οι τάσεις:** ομαλή μακροπρόθεσμη κατεύθυνση
 - **Εποχικότητα:** αλλαγές μέσα σε ένα χρόνο που τείνουν να επαναλαμβάνονται κάθε χρόνο
 - **Κυκλική διακύμανση:** αύξηση και πτώση σε περιόδους συνήθως μεγαλύτερες από ένα έτος
 - **Θόρυβος:** τυχαία μεταβολή/απρόβλεπτο συστατικό





Κουίζ

Τα διαδοχικά δεδομένα και τα δεδομένα χρονοσειρών σημαίνουν ακριβώς το ίδιο πράγμα.
Σωστό ή Λάθος;

Λάθος. Υπάρχουν δεδομένα που περιέχουν στοιχεία των οποίων η σειρά δεν ορίζεται από έναν άξονα χρόνου. Για παράδειγμα, μια ακολουθία DNA, ή λέξεις σε ένα έγγραφο.

Ωστόσο, στις περισσότερες εφαρμογές, τα διαδοχικά δεδομένα είναι χρονικά δεδομένα.

Στις παρακάτω διαφάνειες, θα χρησιμοποιούμε τις λέξεις «σειρά» και «χρόνος» συνώνυμα.





Χειρισμός διαδοχικών δεδομένων

Γιατί ασχολείστε με τα διαδοχικά δεδομένα;

- Οι τιμές των μετοχών δεν έχουν νόημα χωρίς τις πληροφορίες του χρόνου
- Οι μεμονωμένες λέξεις σε μια πρόταση δεν έχουν νόημα χωρίς τα συμφραζόμενα τους

Πώς να χειριστεί τα διαδοχικά δεδομένα;

- Χρειάζεται κάποια μορφή μνήμης.
- Δύο προσεγγίσεις:
 - Η σειρά υποβάλλεται σε εξωτερική επεξεργασία
 - Η σειρά διεκπεραιώνεται εσωτερικά





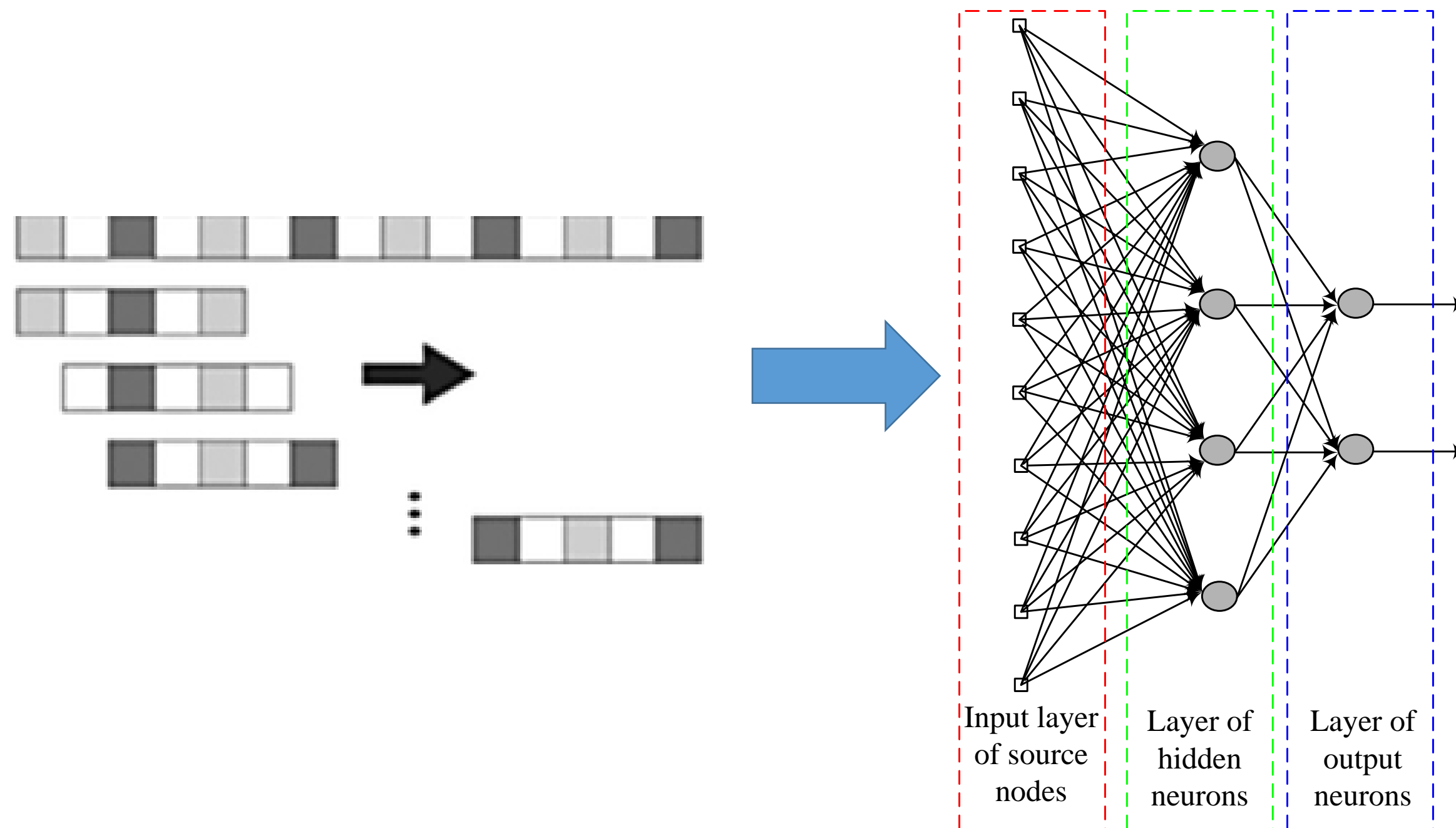
Εξωτερική επεξεργασία της σειράς

- Η σειρά (ή ο χρόνος) υποβάλλεται σε προεπεξεργασία σε ένα μετασχηματισμό τάξης σε διάστημα. Μόνο ο χωρικός μετασχηματισμός είναι προσβάσιμος από το δίκτυο, το οποίο περιέχει μια διάσταση της οποίας η σημασιολογία σχετίζεται με τη σειρά.
- **Τεχνική προσέγγιση** χαρακτηριστικών: εισαγωγή νέων χαρακτηριστικών (εισόδου)
 - Μετατοπισμένες εκδόσεις του σήματος εισόδου, π.χ., $x_t, x_{t-1}, x_{t-2}, \dots, x_{t-k}$
 - Κινούμενοι μέσοι όροι, π.χ., $x_{MA,t} = \frac{\sum_{i=t-k}^t x_i}{k}$
 - Ανάγκη καθορισμού της διάρκειας του **παραθύρου πλαισίου (χρόνος)**, k
 - Το συρόμενο παράθυρο στα διαδοχικά δεδομένα μετατρέπει την ακολουθία μέσα στο παράθυρο σε στατικά δεδομένα





Εξωτερική επεξεργασία της σειράς: MLP



Άκαμπτη προσέγγιση:

- Άκαμπτο όριο στη διάρκεια των μοτίβων
- Δεν συλλαμβάνει τη μεταφραστική αμετάβλητη του ίδιου χρονικού προτύπου σε διαφορετικούς απόλυτους χρόνους

[0,0,1,1,1,0,0,1,0,0]

T

[0,0,0,0,1,1,1,0,0,1]

T+ 2





Εσωτερική επεξεργασία της σειράς

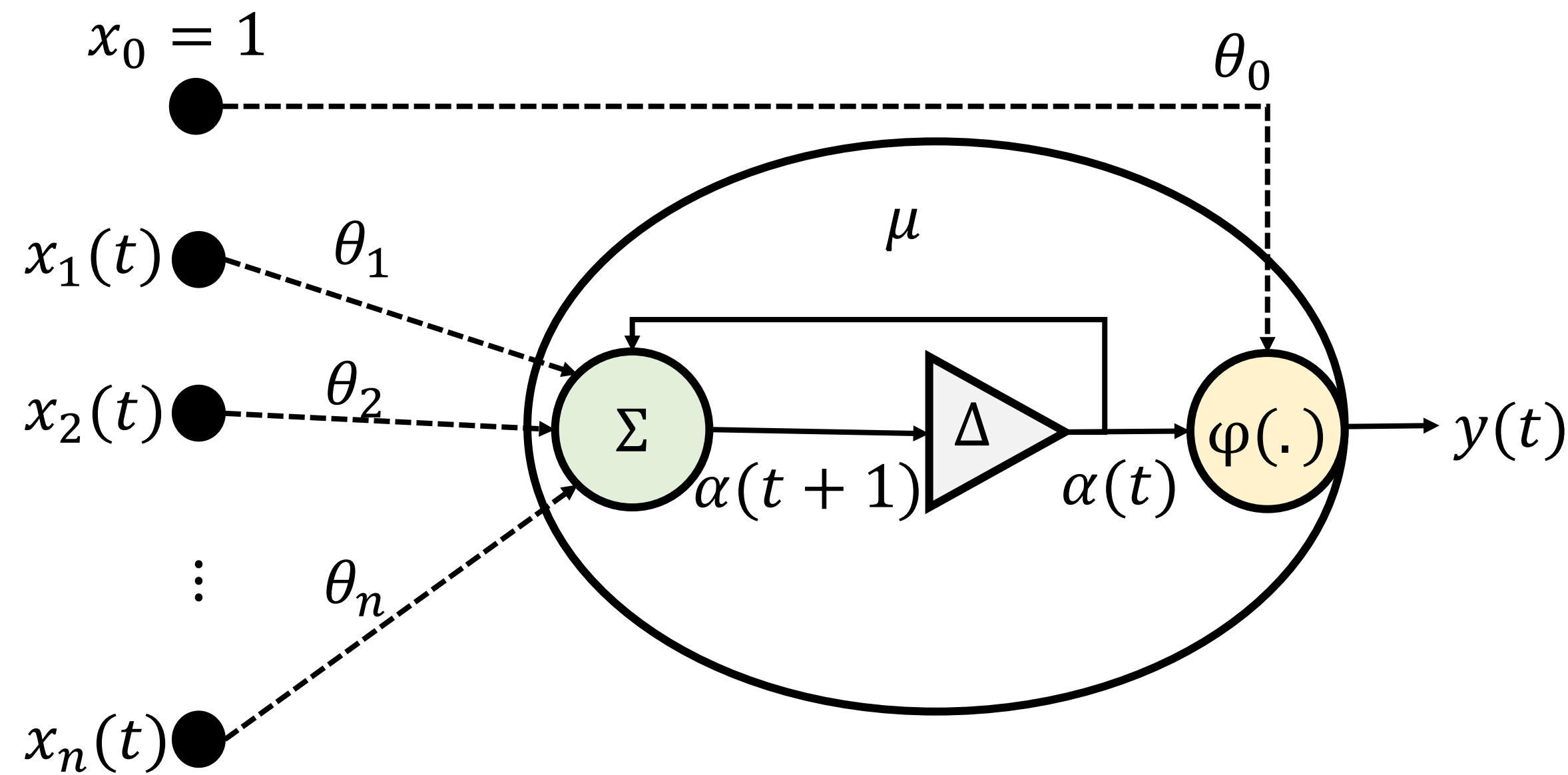
- Το μοντέλο χρησιμοποιεί μια εσωτερική κατάσταση που ενημερώνεται συνεχώς σύμφωνα με μια ροή εισόδου και διατηρεί την προηγούμενη κατάσταση των κόμβων που επανεισάγονται στο επόμενο βήμα.
- Συνήθως γίνεται:
 - Σε **επίπεδο νευρώνων**: το μοντέλο νευρώνων εισάγει κάποια καθυστέρηση κατά την ενσωμάτωση της εισόδου και τον υπολογισμό της απόκρισης.
 - Σε επίπεδο **σύνδεσης**: το σήμα διαδίδεται από τον ένα νευρώνα στον άλλο με κάποια καθυστέρηση





Εσωτερική επεξεργασία της σειράς: επίπεδο νευρώνων

Για παράδειγμα: μοντέλο δυναμικού νευρώνα διακριτού χρόνου



$$\alpha(t + 1) = \mu \alpha(t) + \sum_{i=1}^n \theta_i x_i$$

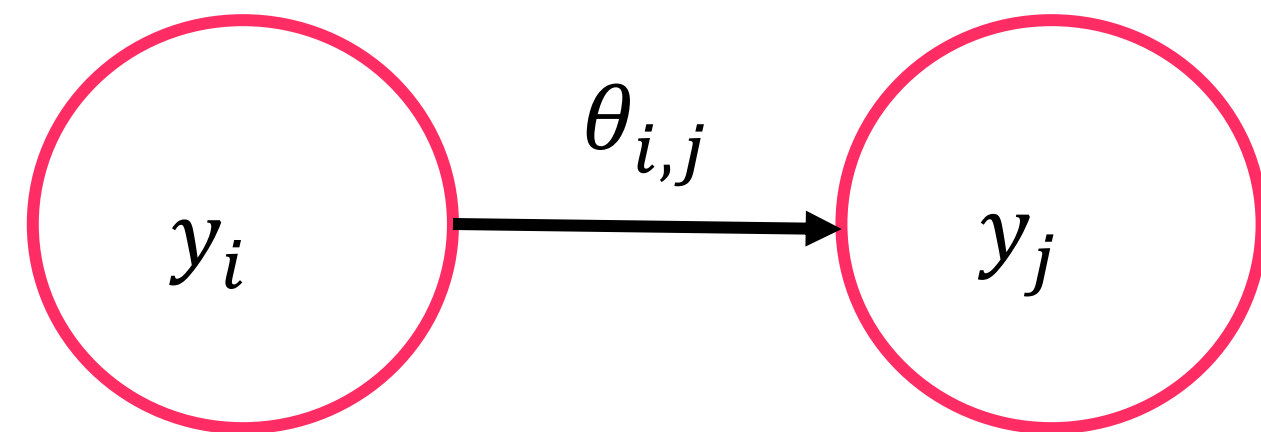
$$y(t) = \varphi(\alpha(t) - \theta_0)$$





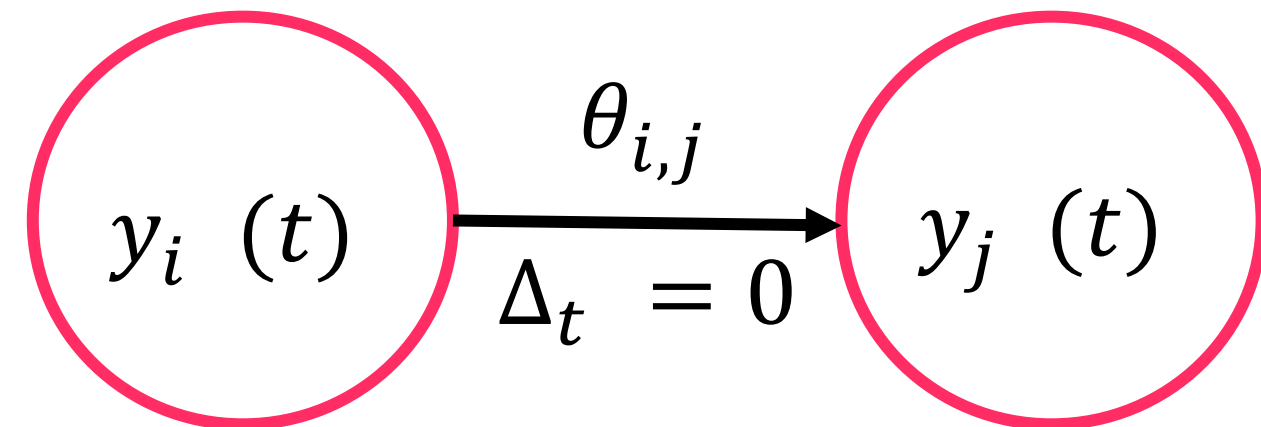
Εσωτερική επεξεργασία της σειράς: επίπεδο σύνδεσης

Στα δίκτυα τροφοδοσίας, θεωρούμε ότι η διάδοση προς τα εμπρός, δηλαδή, η διάδοση ενός σήματος εισόδου σε όλη τη διαδρομή προς το επίπεδο εξόδου γίνεται σε ένα μόνο χρονικό βήμα.



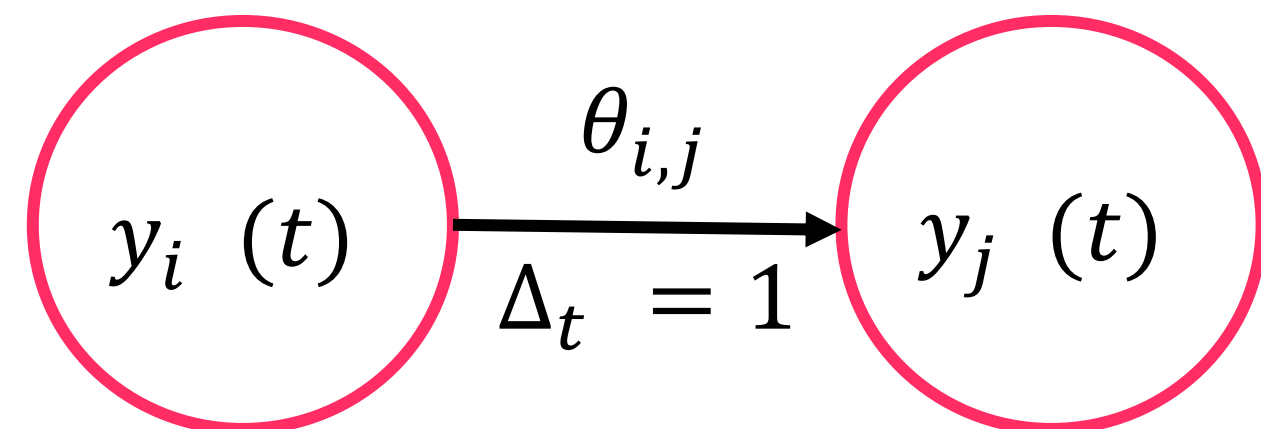
$$y_j = \varphi(\theta_{i,j} y_i)$$

$$y_j(t) = \varphi(\theta_{i,j} y_i(t))$$



$$y_j(t) = \varphi(\theta_{i,j} y_i(t-1))$$

$$y_j(t) = \varphi(\theta_{i,j} y_i(t - \Delta_t))$$



Οι συνδέσεις με καθυστέρηση είναι γνωστές ως **επαναλαμβανόμενες συνδέσεις**.

Τα νευρωνικά δίκτυα που χρησιμοποιούν επαναλαμβανόμενες συνδέσεις είναι γνωστά ως **επαναλαμβανόμενα νευρωνικά δίκτυα**.





Επαναλαμβανόμενα Νευρωνικά Δίκτυα





Επαναλαμβανόμενα νευρωνικά δίκτυα: εμπρόσθια διάδοση

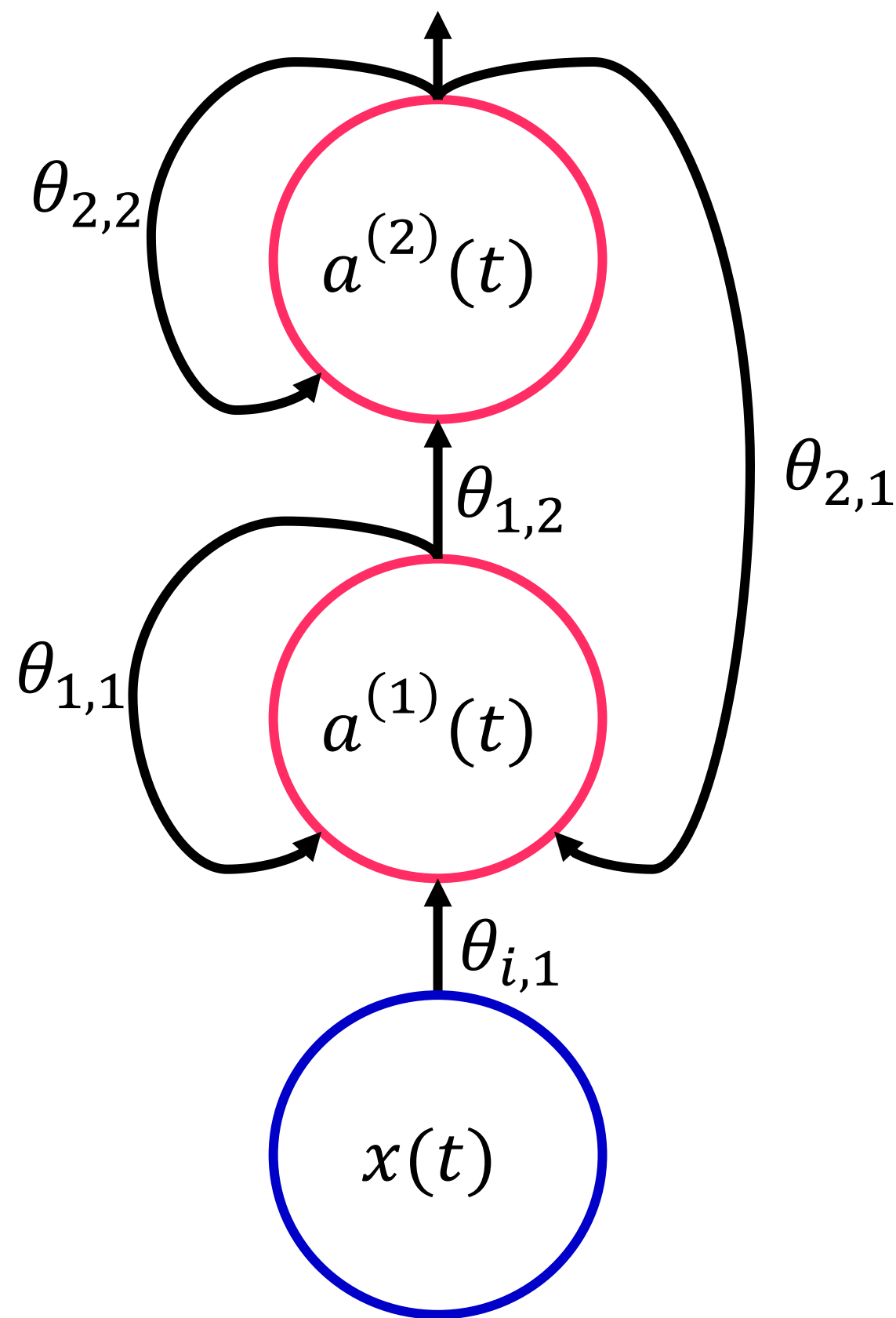
Οι επαναλαμβανόμενες συνδέσεις έχουν συνήθως καθυστέρηση 1

$$a^{(1)}(t) = \theta_{i,1} x(t) + \theta_{1,1} a^{(1)}(t - 1) + \theta_{2,1} a^{(2)}(t - 1)$$

$$a^{(2)}(t) = \theta_{1,2} a^{(1)}(t) + \theta_{2,2} a^{(2)}(t - 1)$$

Κάθε νευρώνας που έχει μια εξερχόμενη επαναλαμβανόμενη σύνδεση λέγεται ότι διατηρεί κάποια **κατάσταση**.

- Αυτή είναι η ενεργοποίησή της που απαιτείται στο επόμενο βήμα





Εκπαίδευση: Οπισθοδιάδοση μέσα στο χρόνο

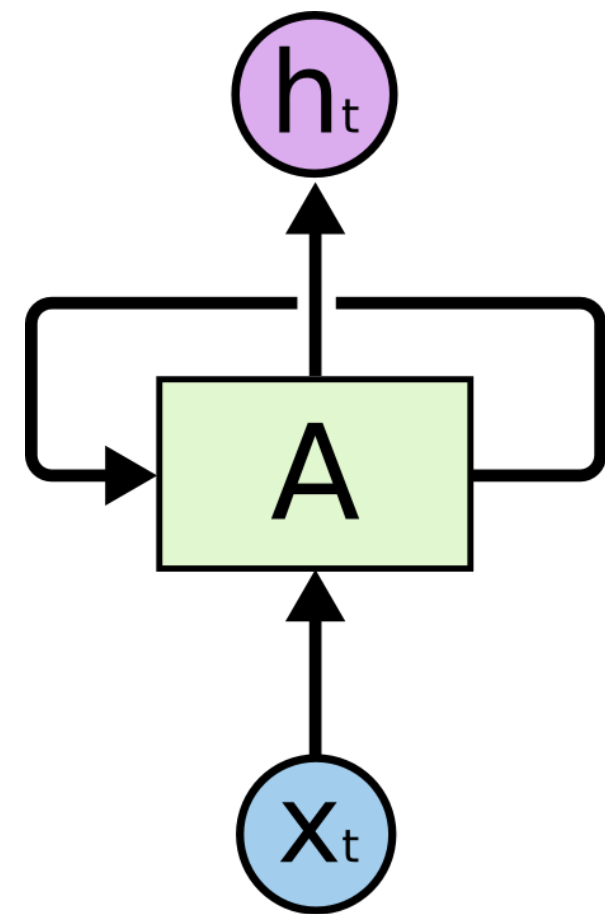
- Η οπισθοδιάδοση μπορεί ακόμα να χρησιμοποιηθεί για την εκπαίδευση επαναλαμβανόμενων νευρωνικών δικτύων
- Τροποποίηση:
 - εμπρόσθια διάδοση: **ξετυλίγετε το υπολογιστικό γράφημα με την πάροδο του χρόνου**, δηλ., διατηρήστε όλες τις καταστάσεις νευρώνων στη μνήμη για το μήκος της ακολουθίας, $t = 1$ έως T
 - σφάλματα οπισθοδιάδοσης ξεκινώντας από το τελευταίο βήμα (T) προς το πρώτο βήμα, ενώ σε κάθε βήμα κάνουμε και **backpropagate** σφάλματα από το επίπεδο εξόδου (αν υπάρχει)
 - συσσωρεύστε όλες τα **gradient** και εφαρμόστε **gradient descent**





Ξετυλίγετε το υπολογιστικό γράφημα

Το δίκτυο γίνεται πολύ βαθύ

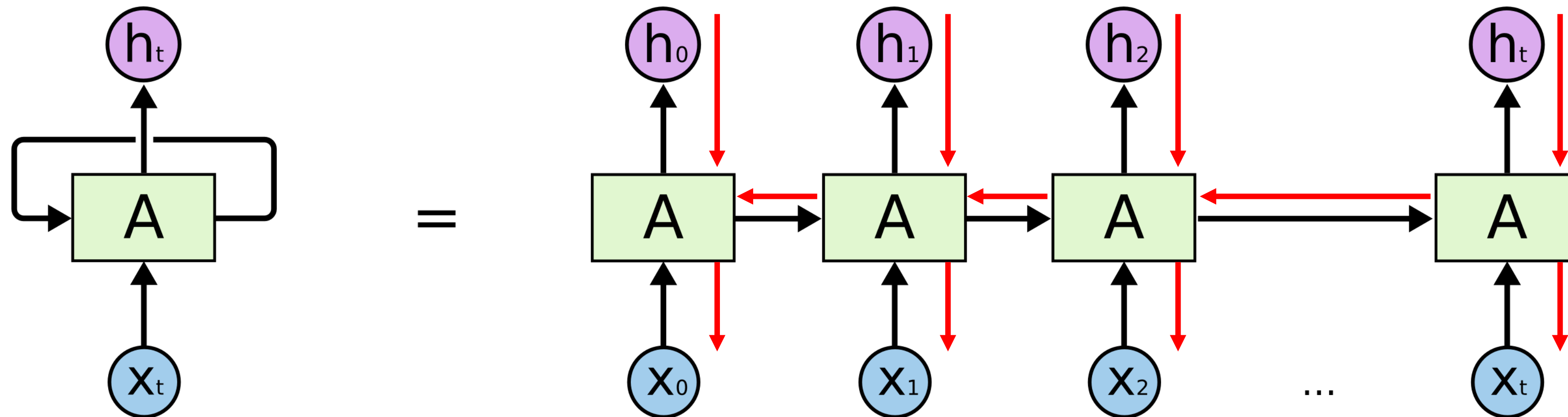


[ΠΗΓΗ](#)





Οπισθοδιάδοση σφαλμάτων



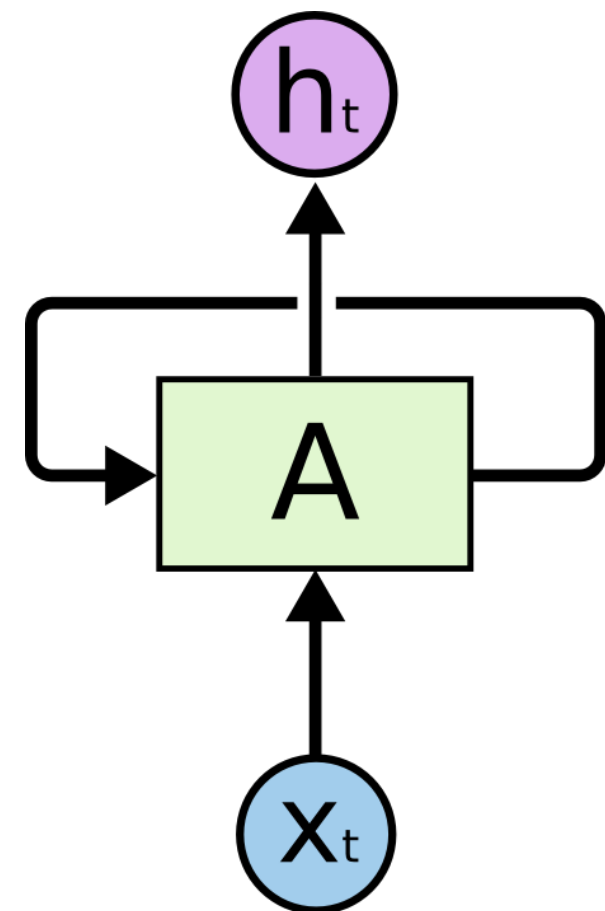
[ΠΗΓΗ](#)





Εφαρμογή ενημέρωσης παραμέτρων

Μόλις συγκεντρωθούν όλες τα gradient, κάνουμε ένα βήμα gradient descent και τροποποιούμε τις παραμέτρους



Παρατηρήστε ότι χρησιμοποιούμε **καταμερισμό βάρους**: το μη τυλιγμένο γράφημα χρησιμοποιεί τα ίδια βάρη για την επεξεργασία διαφορετικών εισόδων στην ακολουθία

- Αν τα βάρη ήταν διαφορετικά τότε θα ήταν ένας τύπος νευρωνικού δικτύου τροφοδοσίας.

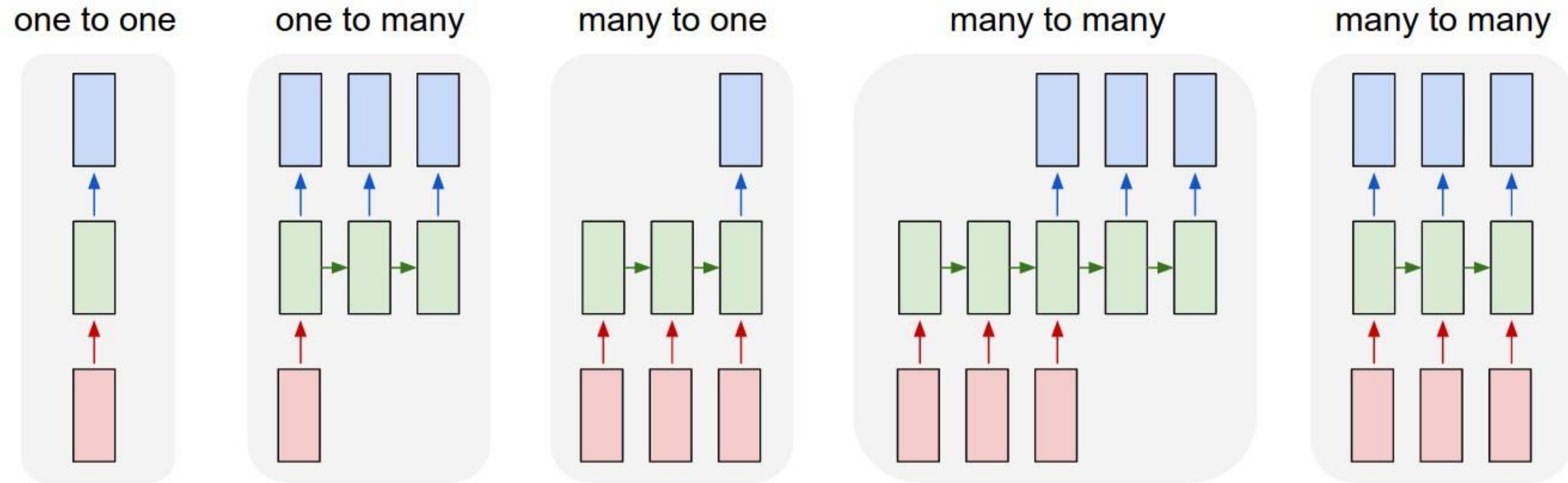
Αυτό επιτρέπει στα ΣΝΔ να περιορίζουν τους υπολογιστικούς πόρους και τις ακολουθίες διεργασιών μεταβλητού μήκους

[ΠΗΓΗ](#)





ΣΝΔ Παραδείγματα



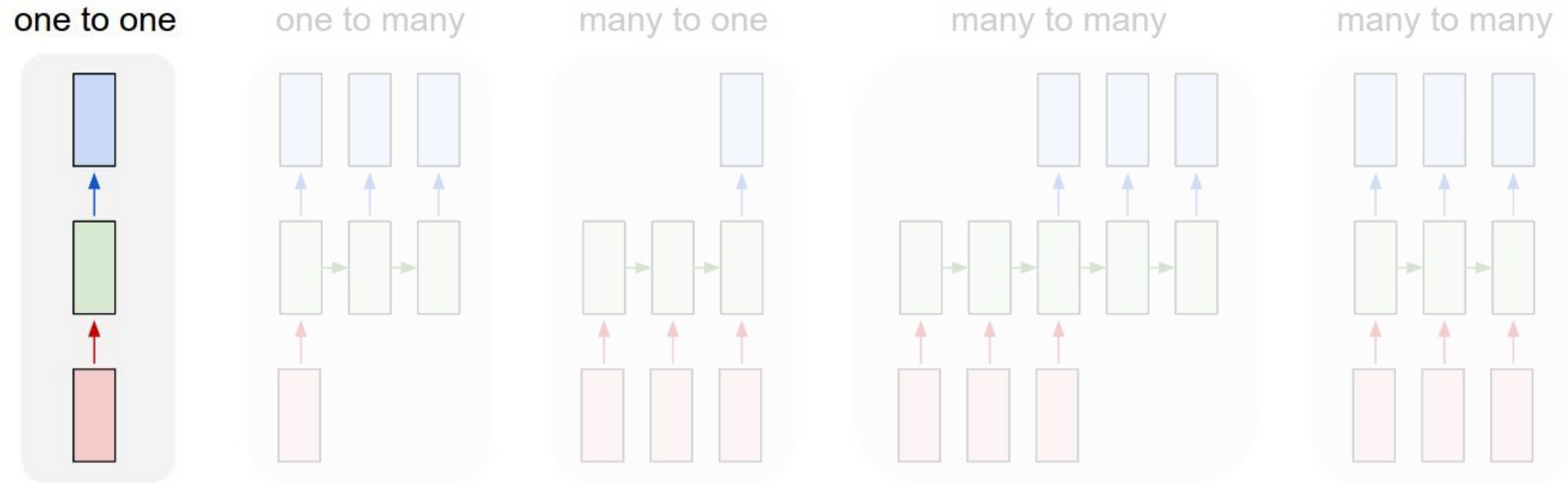
[ΠΗΓΗ](#)

Κάθε ορθογώνιο είναι ένα διάνυσμα και τα βέλη αντιπροσωπεύουν συναρτήσεις (π.χ. πολλαπλασιασμός πινάκων). Τα διανύσματα εισόδου είναι σε κόκκινο χρώμα, τα διανύσματα εξόδου είναι μπλε και τα πράσινα διανύσματα κρατούν την κατάσταση του ΣΝΔ





RNN Παραδείγματα



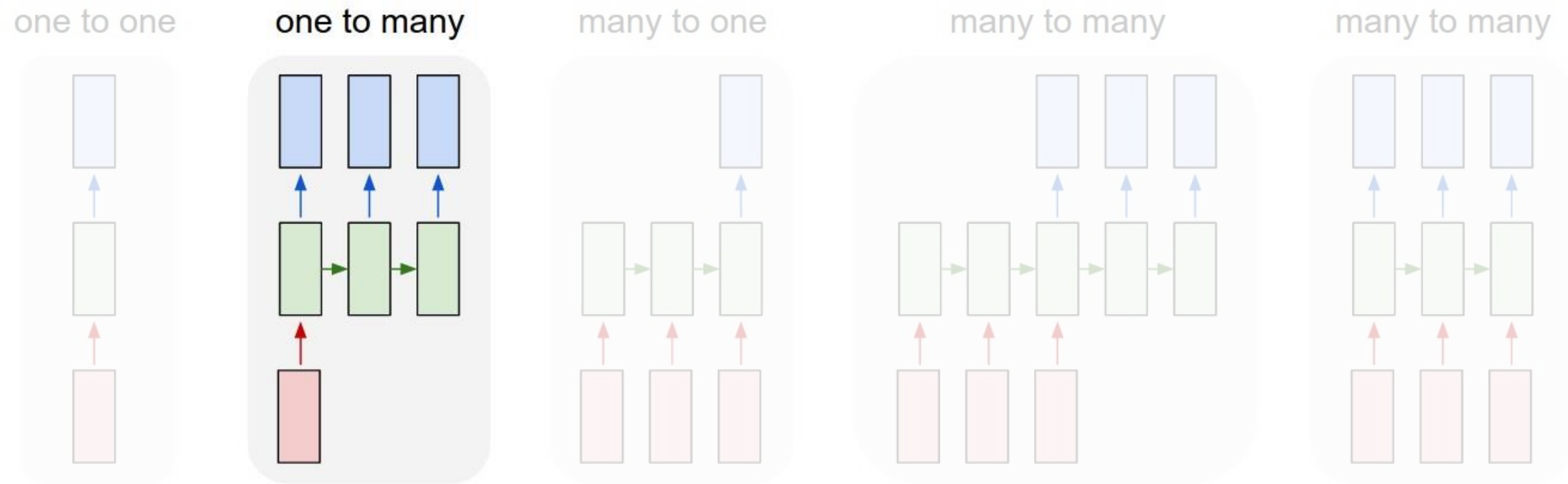
[ΠΗΓΗ](#)

Απλός τρόπος επεξεργασίας χωρίς ΣΝΔ, από είσοδο σταθερού μεγέθους έως έξοδο σταθερού μεγέθους (π.χ. ταξινόμηση εικόνας).





RNN Παραδείγματα



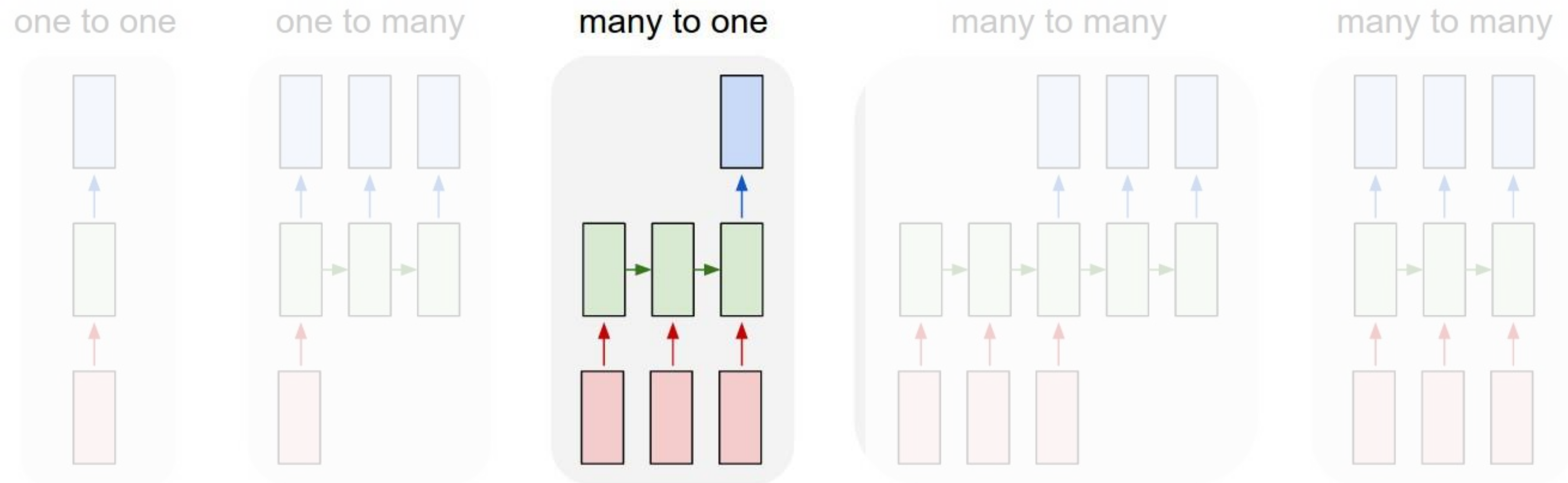
[ΠΗΓΗ](#)

Έξοδος αλληλουχίας (π.χ. η λεζάντα εικόνας λαμβάνει μια εικόνα και εξάγει μια πρόταση λέξεων).





RNN Παραδείγματα

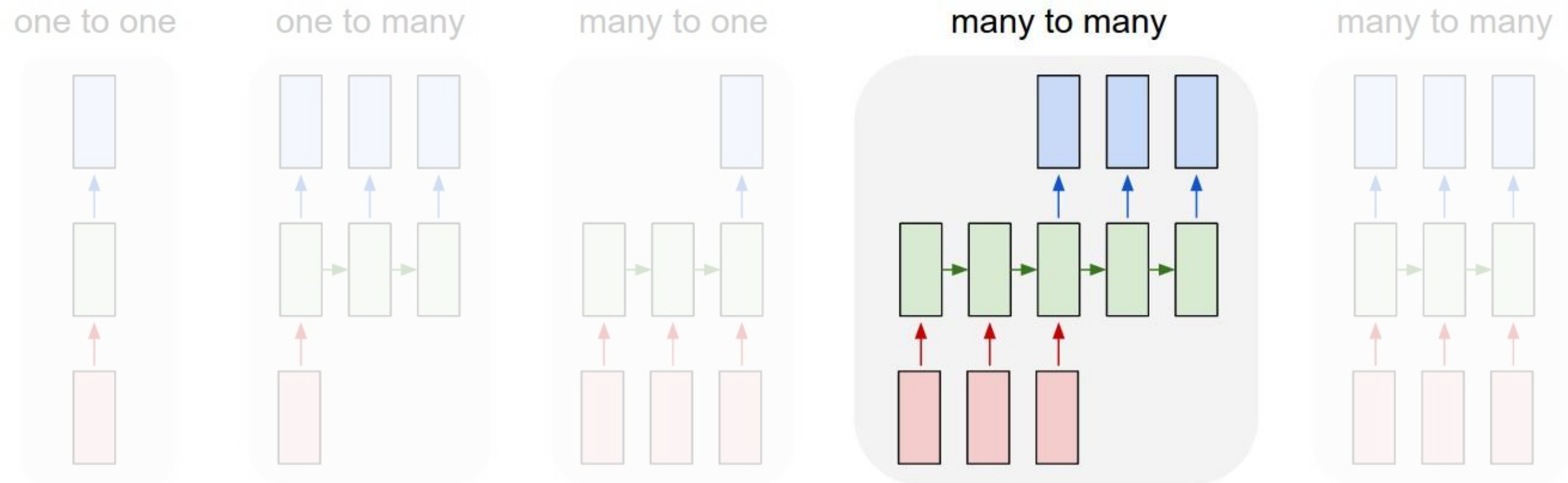


[ΠΗΓΗ](#)

Εισαγωγή αλληλουχίας (π.χ. ανάλυση αισθήματος κατά την οποία μια δεδομένη πρόταση ταξινομείται ως έκφραση θετικού ή αρνητικού συναισθήματος).



RNN Παραδείγματα



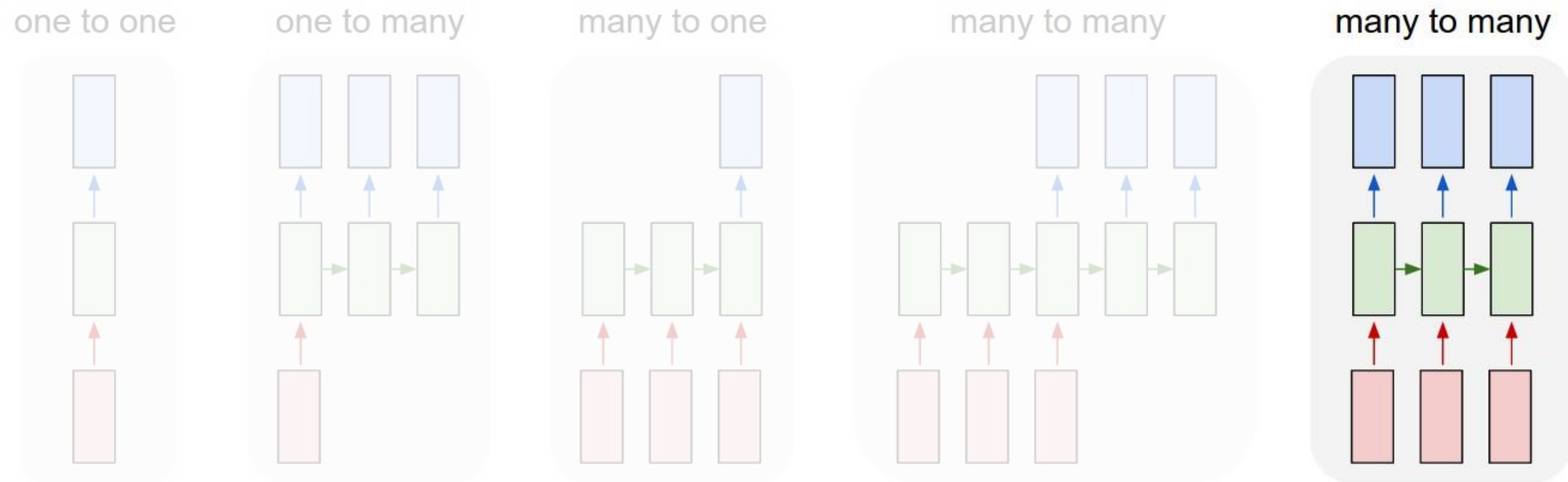
[ΠΗΓΗ](#)

Εισαγωγή ακολουθίας και έξοδος αλληλουχίας (π.χ. μηχανική μετάφραση: ένα ΣΝΔ διαβάζει μια πρόταση στα αγγλικά και στη συνέχεια εξάγει μια πρόταση στα γαλλικά).





RNN Παραδείγματα



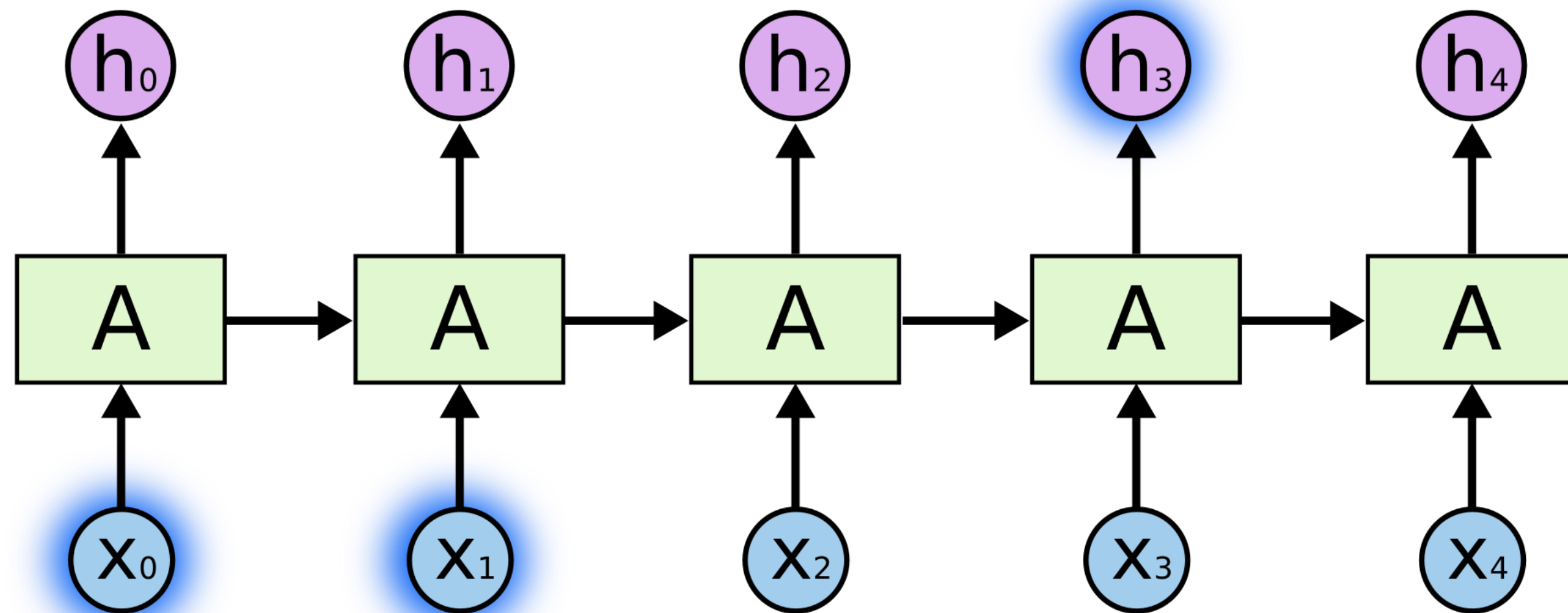
[ΠΗΓΗ](#)

Συγχρονισμένη εισαγωγή και έξοδος αλληλουχίας (π.χ. ταξινόμηση βίντεο όπου θέλουμε να επισημάνουμε κάθε frame του βίντεο).





Το πρόβλημα των μακροπρόθεσμων εξαρτήσεων



[ΠΗΓΗ](#)

Εξετάστε ένα γλωσσικό μοντέλο που προσπαθεί να προβλέψει την επόμενη λέξη με βάση τις προηγούμενες

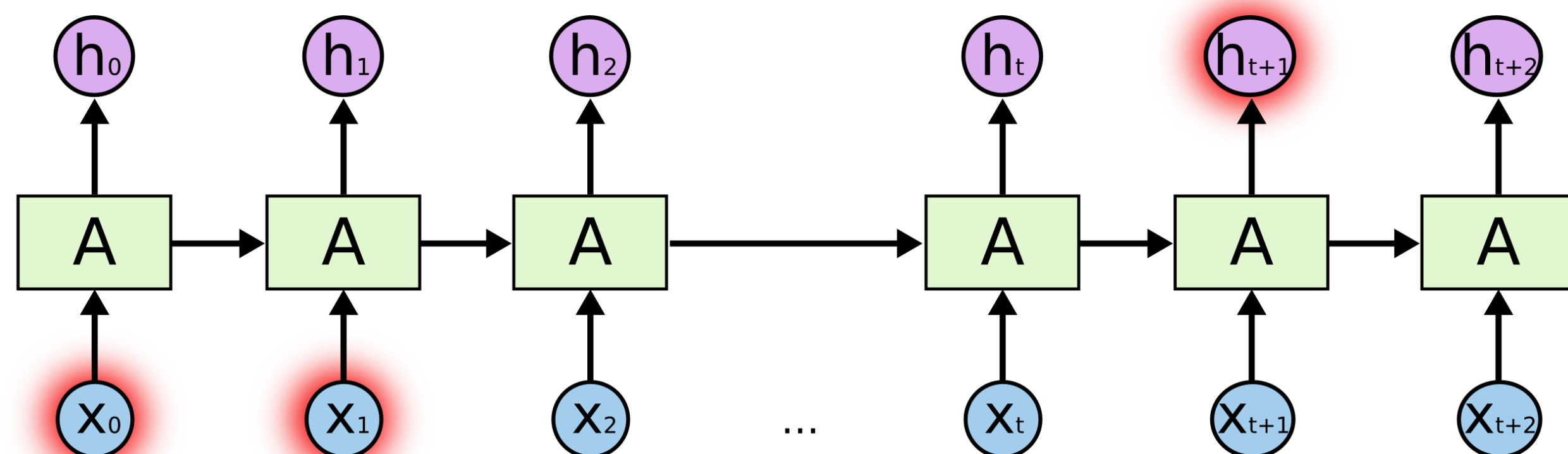
Αν προσπαθούμε να προβλέψουμε την τελευταία λέξη στο «τα σύννεφα είναι στον *ουρανό*», δεν χρειαζόμαστε περαιτέρω πλαίσιο - είναι αρκετά προφανές ότι η επόμενη λέξη θα είναι ο *ουρανός*.

Σε τέτοιες περιπτώσεις, όπου το χάσμα μεταξύ των σχετικών πληροφοριών και του τόπου που χρειάζονται είναι μικρό, τα ΣΝΔ μπορούν να μάθουν να χρησιμοποιούν τις προηγούμενες πληροφορίες.





Το πρόβλημα των μακροπρόθεσμων εξαρτήσεων



Σκεφτείτε να προσπαθήσετε να προβλέψετε την τελευταία λέξη στο κείμενο «Μεγάλωσα στη Γαλλία [...] Μιλώ άπταιστα *γαλλικά*».

Πρόσφατες πληροφορίες δείχνουν ότι η επόμενη λέξη είναι πιθανώς το όνομα μιας γλώσσας.

Αν θέλουμε να περιορίσουμε ποια γλώσσα, χρειαζόμαστε το πλαίσιο της Γαλλίας, από πιο πίσω.

Το χάσμα μεταξύ των σχετικών πληροφοριών και του σημείου όπου χρειάζεται μπορεί να γίνει πολύ μεγάλο.

Καθώς αυτό το χάσμα μεγαλώνει, τα ΣΝΔ αδυνατούν να μάθουν να συνδέουν τις πληροφορίες.

[ΠΗΓΗ](#)





Το πρόβλημα της εξαφάνισης και της έκρηξης της κλίσης

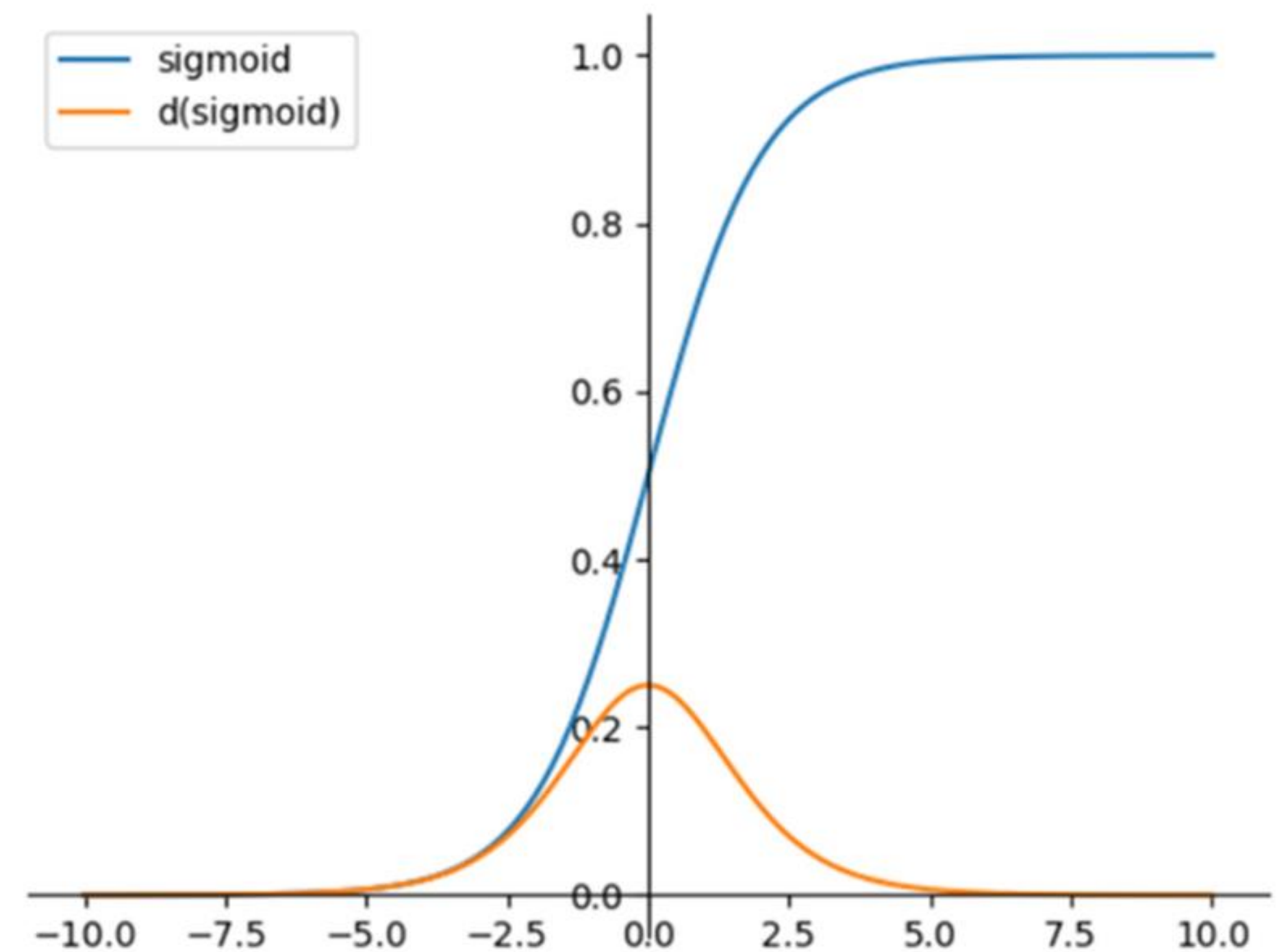
- Κατά τη διάδοση των κλίσεων σε ένα βαθύ δίκτυο (π.χ. ένα επαναλαμβανόμενο νευρωνικό δίκτυο που ξετυλίγεται για πολλά χρονικά βήματα) από το επίπεδο εξόδου στο επίπεδο εισόδου, συχνά οι κλίσεις γίνονται μικρότερες και μικρότερες και πλησιάζουν το μηδέν.
 - Το Gradient descent αφήνει αμετάβλητα τα βάρη των κατώτερων στρωμάτων
 - Ποτέ δεν συγκλίνουν στο βέλτιστο
 - Αυτό είναι γνωστό ως το **πρόβλημα των αποκλίσεων που εξαφανίζονται**
- Σε ορισμένες περιπτώσεις, οι κλίσεις γίνονται μεγαλύτερες και μεγαλύτερες
 - Αυτό προκαλεί μεγάλες ενημερώσεις βάρους που κάνουν το gradient να αποκλίνει
 - Αυτό είναι γνωστό ως το **πρόβλημα των εκρηκτικών gradient**





Γιατί εξαφανίζονται τα gradient;

- Λειτουργίες ενεργοποίησης όπως το σιγμοειδές, διαπερνά ένα μεγάλο χώρο εισόδου σε ένα μικρότερο χώρο εξόδου που βρίσκεται σε $[0,1]$
- Μεγαλύτερες εισροές (θετικές ή αρνητικές) διαποτίζουν τη συνάρτηση που οδηγεί σε παράγωγο κοντά στο μηδέν
- Αυτό σημαίνει ότι πρακτικά δεν υπάρχουν gradient για να διαδοθούν προς τα πίσω, ενώ οποιαδήποτε υπολείμματα συνεχίζουν να αραιώνονται.



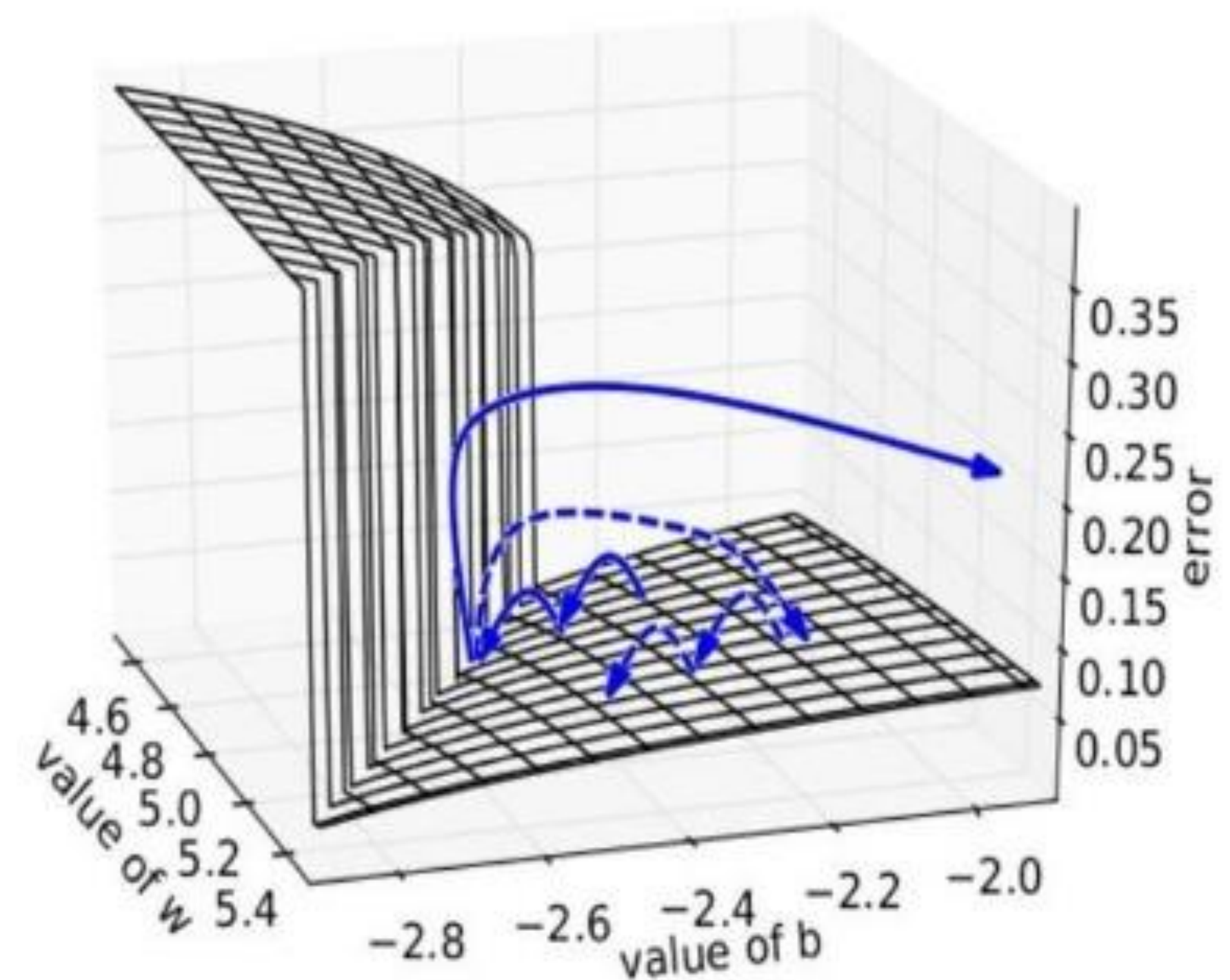
[Πηγή εικόνας](#)





Γιατί τα gradient εκρήγνυνται;

- Ας υποθέσουμε ότι οι αρχικές παράμετροι οδηγούν σε κάποια μεγάλη απώλεια.
- Τα gradient μπορούν να συσσωρευτούν κατά τη διάρκεια μιας ενημέρωσης και να οδηγήσουν σε πολύ μεγάλες gradient οι οποίες τελικά οδηγούν σε πολύ μεγάλες ενημερώσεις των παραμέτρων του δικτύου.
- Μερικές φορές οι παράμετροι υπερχειλίζουν και παράγουν τιμές NaN



[Πηγή εικόνας](#)



Πιθανές λύσεις;

- Καλύτερη αρχικοποίηση του βάρους (π.χ. λαμβάνοντας υπόψη τη διακύμανση των εισροών και των εκροών σε κάθε στρώμα)
- Χρήση μη κορεσμένων λειτουργιών ενεργοποίησης (ReLU, Leaky ReLU, ELU,...)
- Ομαλοποίηση παρτίδας (μηδενικός μέσος όρος ομαλοποίησης της εισόδου σε ένα στρώμα με βάση δεδομένα mini-batch)
- Αποκοπή gradient (οι τιμές gradient πέραν του -1 ή + 1 αποκόπτονται σε αυτό το εύρος)
- Μην μαθαίνετε τα επαναλαμβανόμενα βάρη: αγνοήστε το πρόβλημα (δείτε την επόμενη διαφάνεια)
- Χρησιμοποιήστε μια πιο προηγμένη αρχιτεκτονική (δείτε τις επόμενες διαφάνειες)





Αγνοώντας το πρόβλημα: Echo State Networks

- Ξετυλίγουμε το υπολογιστικό γράφημα όταν θέλουμε να εκπαιδεύσουμε επαναλαμβανόμενες συνδέσεις.
- Με το να **μην** εκπαιδεύουμε τις επαναλαμβανόμενες συνδέσεις, δεν έχουμε το πρόβλημα της εξαφάνισης/εξερεύνησης gradient.

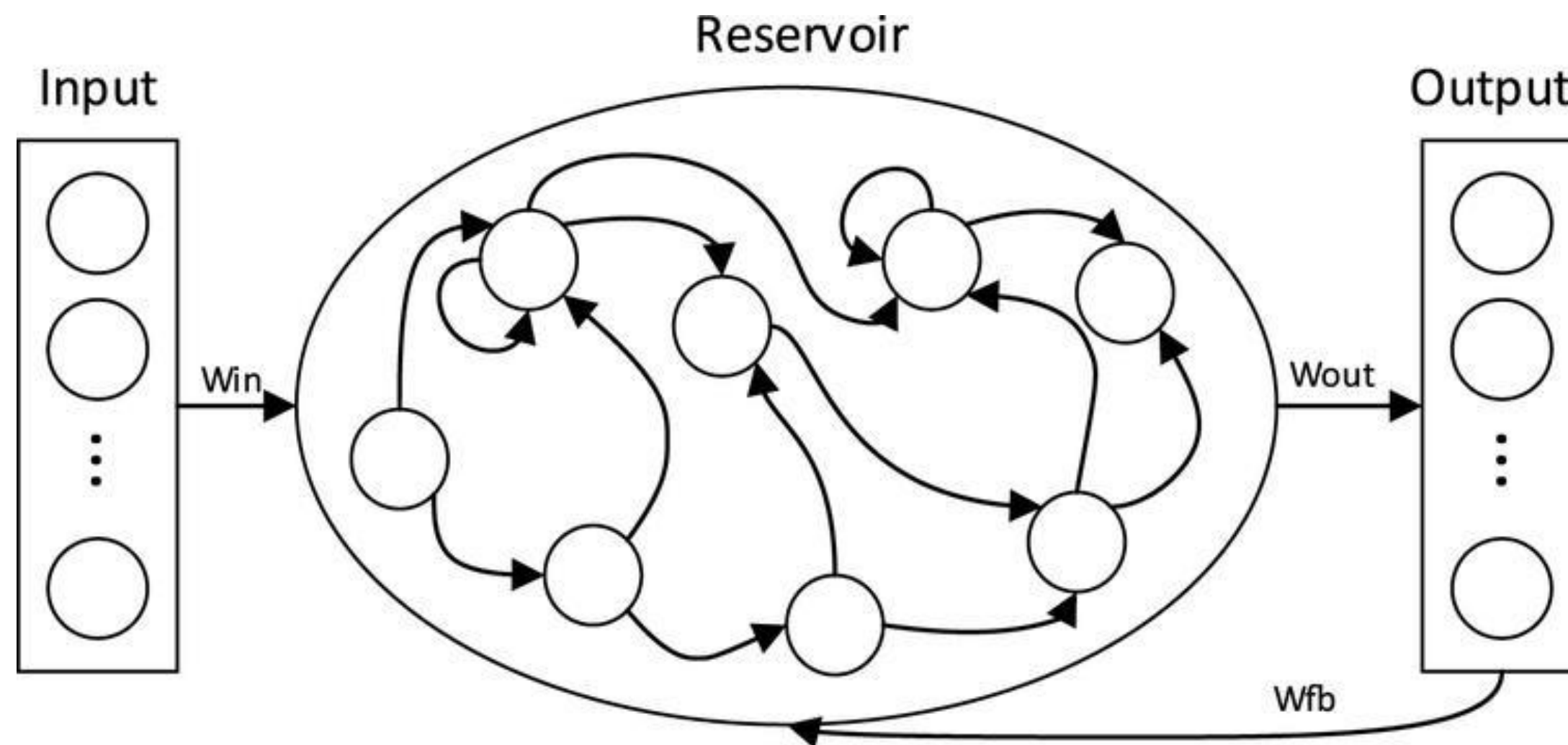
Echo State Networks:

- **Reservoir computing**: μεγάλο **αραιά συνδεδεμένο** κρυφό στρώμα (1% συνδεσιμότητα) με σύνδεση και βάρη **σταθερά** και **τυχαία** εκχωρημένα.
- Είσοδος σε κρυμμένες και συνδέσεις εξόδου σε κρυμμένες επίσης τυχαίες
- Μόνο συνδέσεις από κρυφό στρώμα στην έξοδο είναι εκπαιδεύσιμες
- Προβλήματα οπισθοδρόμησης: η έξοδος είναι ένας **γραμμικός συνδυασμός** της κρυφής κατάστασης





Echo State Networks: Πως λειτουργουν



Η εισαγωγή σε κάθε βήμα τροφοδοτείται στη δυναμική δεξαμενή, η οποία μαζί με τις εσωτερικές επαναλαμβανόμενες συνδέσεις και τις επαναλαμβανόμενες συνδέσεις από το στρώμα εξόδου υπολογίζει την επόμενη κατάστασή της.

Η έξοδος υπολογίζεται ως γραμμικός συνδυασμός της τρέχουσας κατάστασης της δεξαμενής και των βαρών εξόδου.



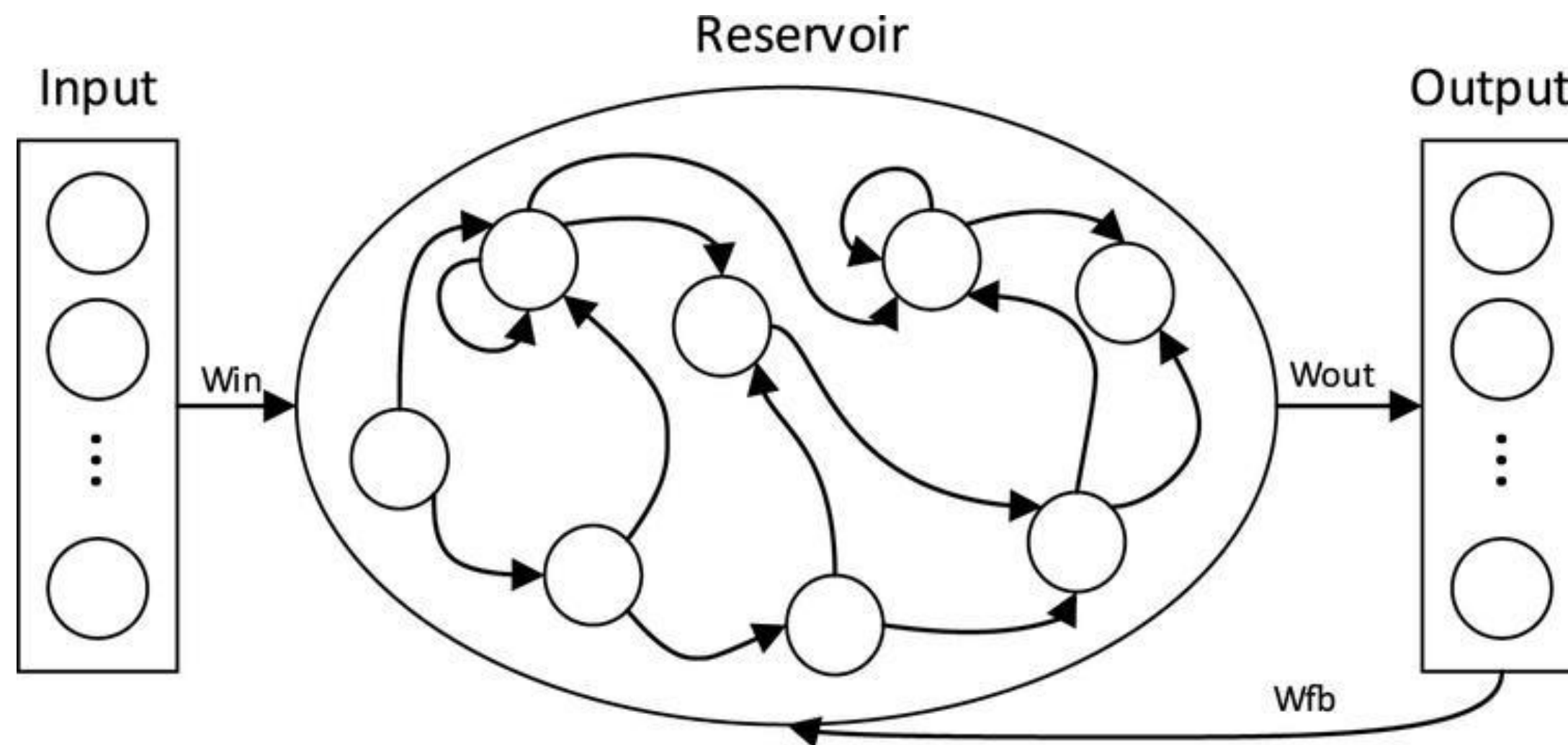


Echo State Networks: πώς να τα εκπαιδεύσετε

Οι κρυφές μονάδες λειτουργούν ως χαρακτηριστικά που υπολογίζονται από την ακολουθία.

- Μπορεί να θεωρηθεί ως ένα είδος μηχανικής χαρακτηριστικών

Μπορούμε να χρησιμοποιήσουμε γραμμική παλινδρόμηση για να εκπαιδεύσουμε τα βάρη παραγωγής!





Echo State Networks

- Καλή απόδοση σε πολλές εφαρμογές επεξεργασίας σημάτων, και εργασίες πρόβλεψης χρονοσειρών.
- Εύκολος στην εφαρμογή
- Εξετάστε τα όταν το πρόβλημα δεν είναι πολύ περίπλοκο και όταν είναι επιθυμητή η φθηνή, γρήγορη και προσαρμοστική εκπαίδευση

Ωστόσο, για εργασίες που έχουν πολλές μεταβλητές και μακροπρόθεσμη μνήμη (π.χ., στην επεξεργασία φυσικής γλώσσας), θα χρειάζονταν μια δεξαμενή υπερβολικού μεγέθους.

Η έλευση των βιβλιοθηκών αυτόματης διαφοροποίησης και οι πιο σταθερές επαναλαμβανόμενες αρχιτεκτονικές (δείτε την επόμενη διαφάνεια) επιτρέπουν την εκπαίδευση των ΣΝΔ





Άμβλυνση του προβλήματος: Long Short-Term Memory

LSTM: Ειδικός τύπος ΣΝΔ ικανός να χειρίζεται μακροχρόνιες εξαρτήσεις

Εισάγει ειδικούς **μηχανισμούς** που επιτρέπουν τη μάθηση:

- ποιες πληροφορίες πρέπει να **ξεχάσουμε**,
- ποιες νέες πληροφορίες πρέπει να **αποθηκευτούν στη μνήμη** και
- τι πληροφορίες για την **παραγωγή**

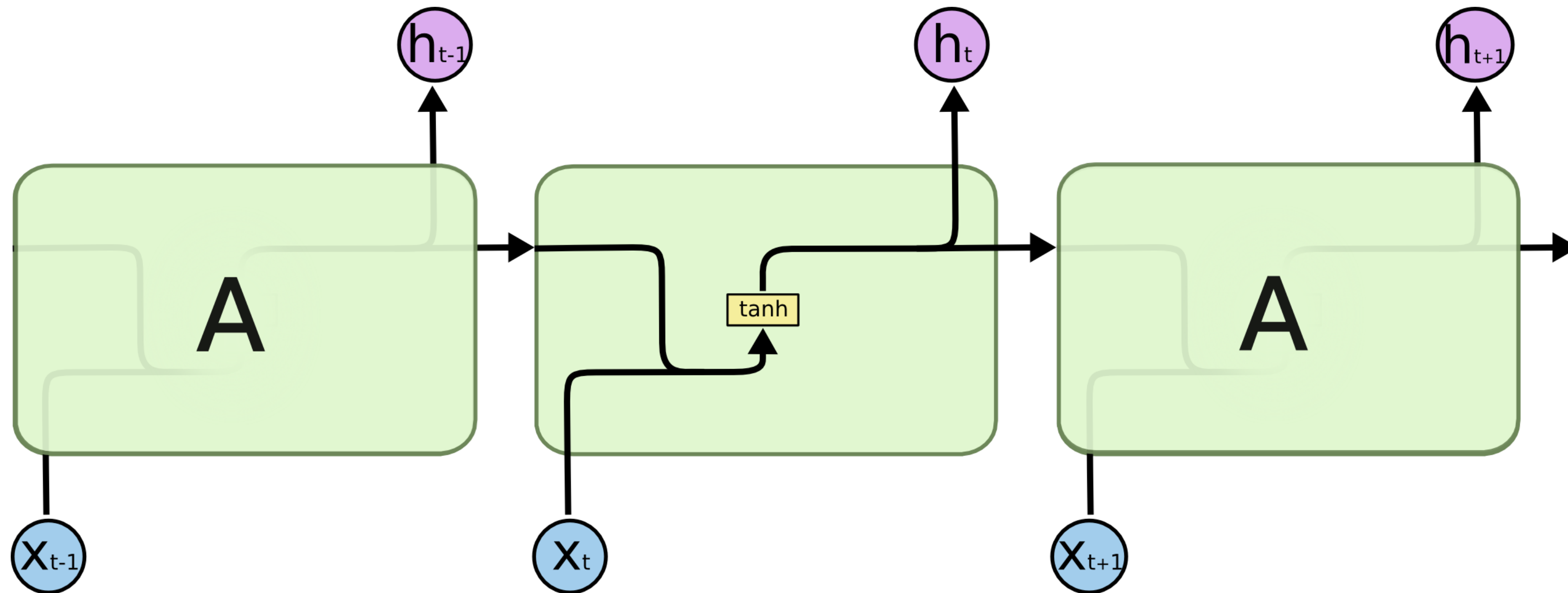
Πώς επιτυγχάνει την περτύλιξη:

- τροφοδοτήστε ένα σήμα μέσω μιας **σιγμοειδούς** λειτουργίας: παράγει μια αξία στο $[0,1]$
- **πολλαπλασιάστε** το αποτέλεσμα με άλλο σήμα x : κάνει το αποτέλεσμα σε $[0, x]$





Ένα επαναλαμβανόμενο στρώμα tanh



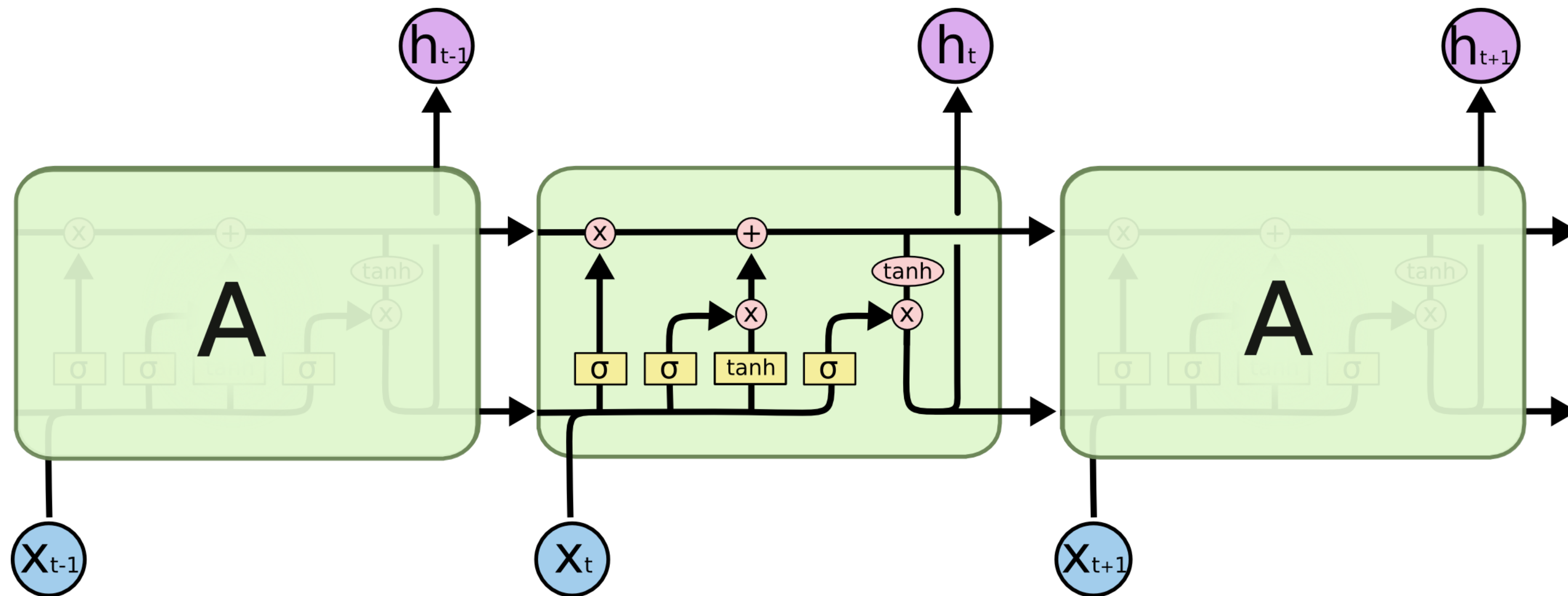
[ΠΗΓΗ](#)

Ένα κρυφό στρώμα με επαναλαμβανόμενες συνδέσεις που χρησιμοποιεί τη συνάρτηση ενεργοποίησης tanh παίρνει ως είσοδο το διάνυσμα εισόδου στο χρόνο t (x_t) και την κρυμμένη κατάσταση κατά το χρόνο $t-1$ (h_{t-1}), και εξάγει την κρυμμένη κατάσταση κατά το χρόνο t h_t που μπορεί να χρησιμοποιηθεί σε άλλο στρώμα (στην κορυφή) ή να διαδοθεί προς τα εμπρός στο χρόνο (στα δεξιά).





LSTM στρώματα



[ΠΗΓΗ](#)

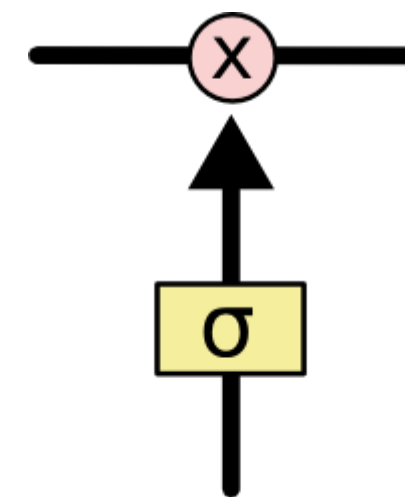
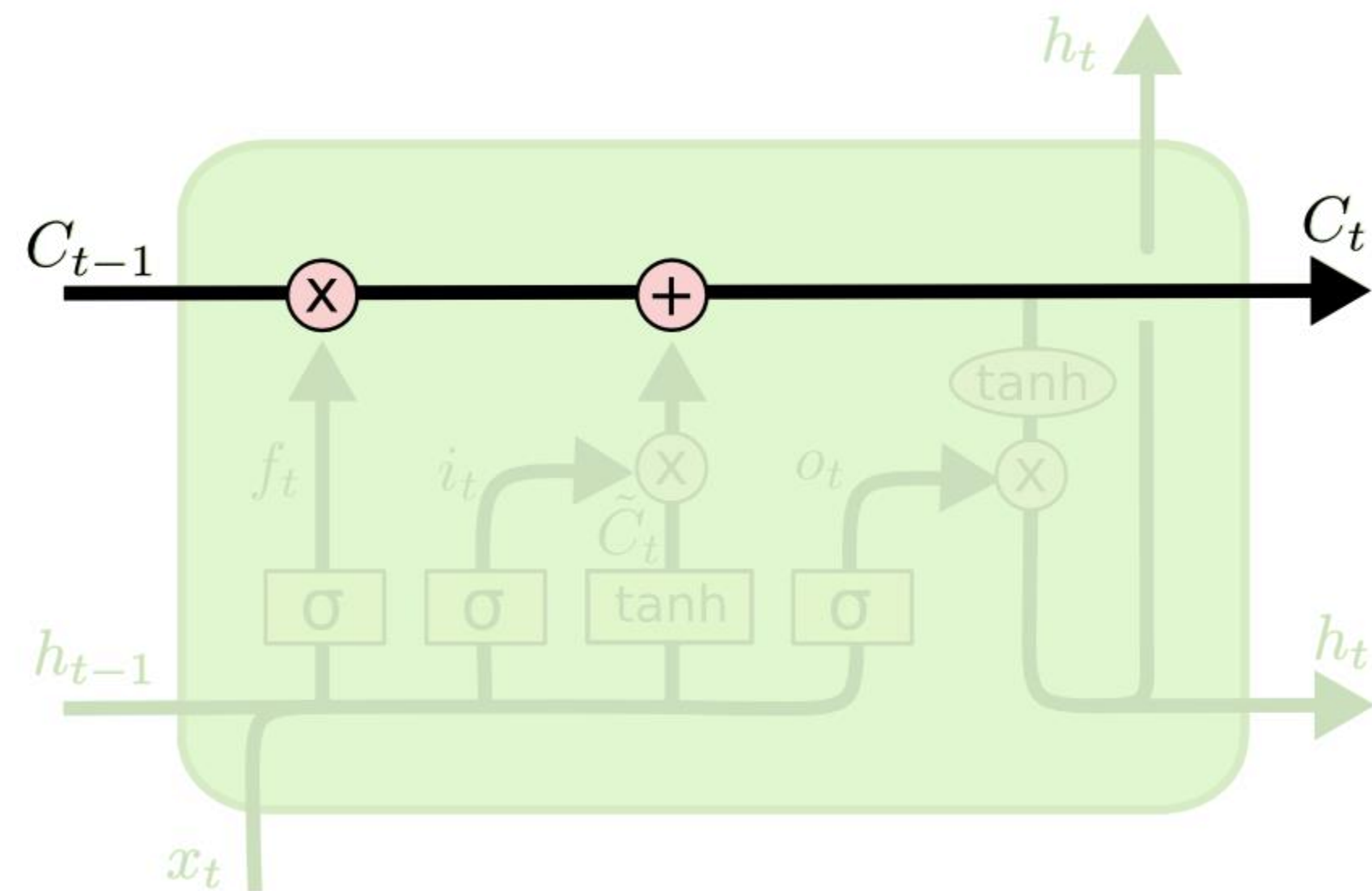
Ένα κύτταρο LSTM εξάγει δύο καταστάσεις: (1) η κατάσταση του κυττάρου που πολλαπλασιάζεται προς τα εμπρός στο χρόνο και (2) η κρυφή κατάσταση που μπορεί να χρησιμοποιηθεί σε άλλα στρώματα στην κορυφή ή να διαδοθεί προς τα εμπρός στο χρόνο.

LSTM: έχει 3 πύλες για την προστασία και τον έλεγχο της κατάστασης κυττάρων





LSTM στρώματα



Η κατάσταση των κυττάρων τρέχει μόνο με κάποιες μικρές **γραμμικές** αλληλεπιδράσεις. Διαδίδεται προς τα εμπρός στο χρόνο.

Πύλες: τρόπος για να αφήσετε προαιρετικά τις πληροφορίες να περάσουν

Αποτελείται από το σιγμοειδές στρώμα και τη σημειακή λειτουργία πολλαπλασιασμού

Παραγωγή σιγμοειδούς: $[0, 1]$

Τιμή 0: μην αφήσεις τίποτα να περάσει

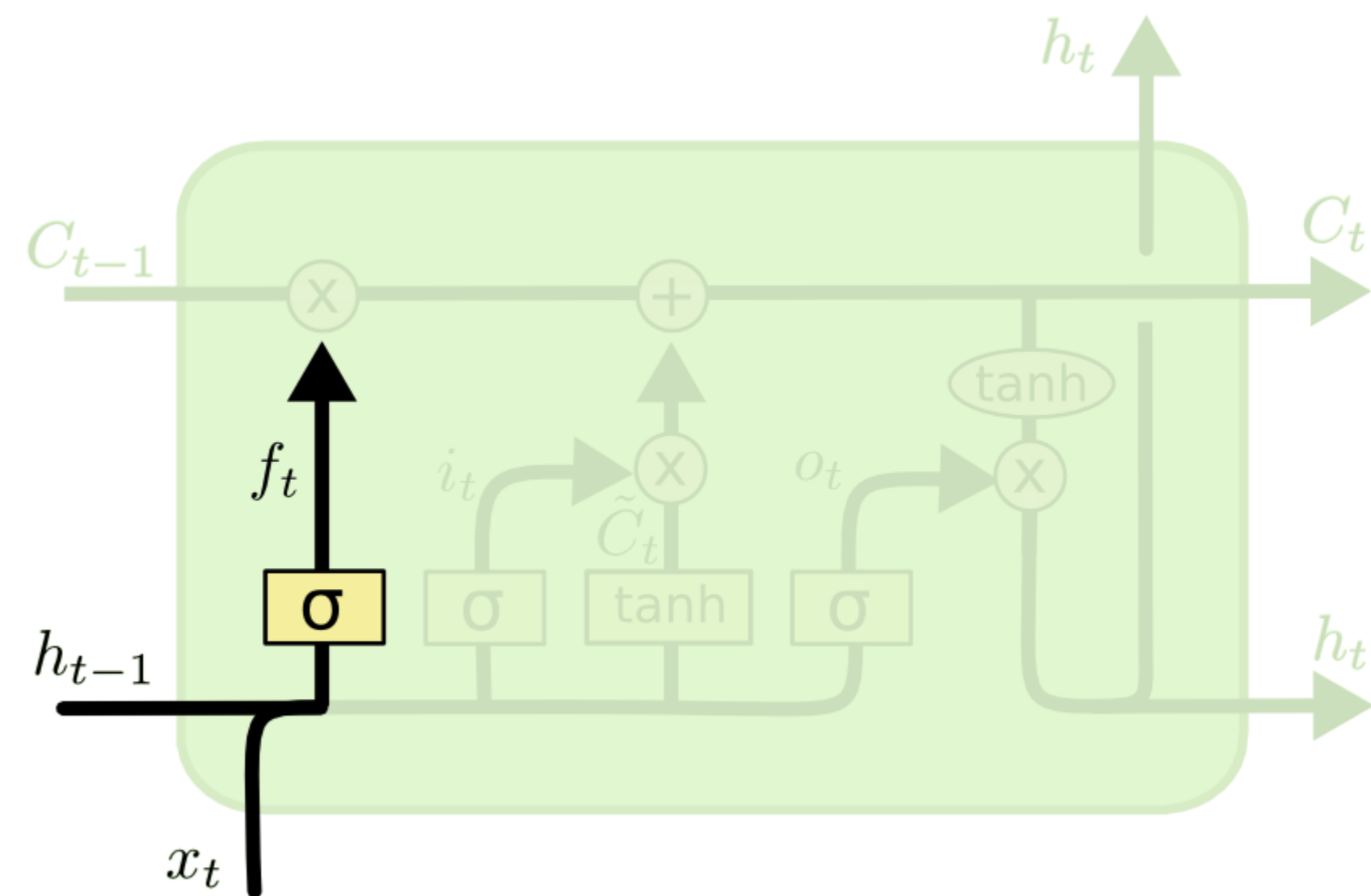
Τιμή 1: αφήστε τα πάντα να περάσουν

[ΠΗΓΗ](#)





LSTM στρώματα



Ξεχάστε την πύλη: τι πληροφορίες να πετάξετε

Έξοδοι [0,1]

1: κράτησε εντελώς αυτό

0: ξέχασε εντελώς αυτό

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

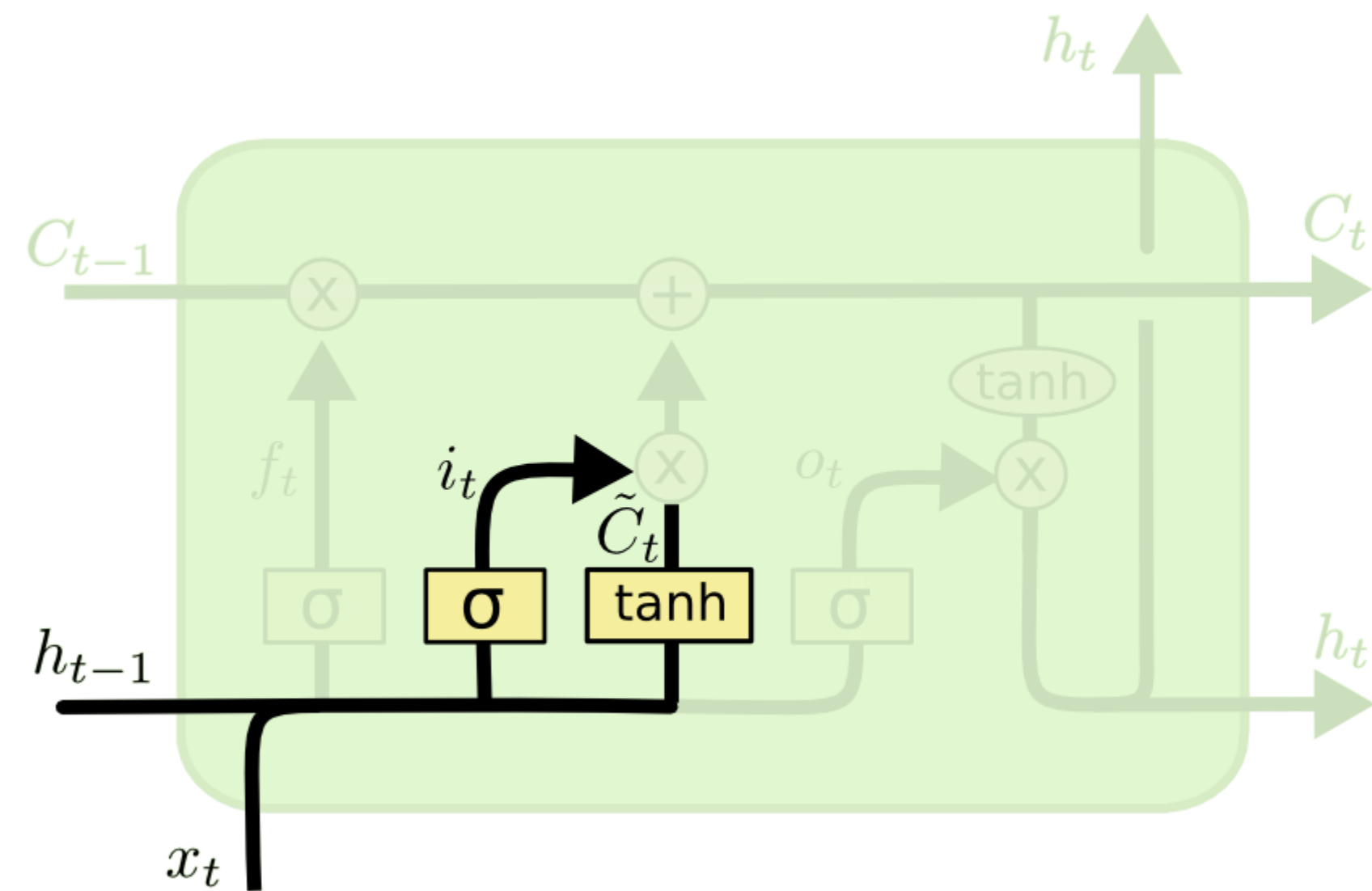
[ΠΗΓΗ](#)





LSTM στρώματα

Πύλη εισαγωγής: ποιες πληροφορίες θα αποθηκευτούν στην κατάσταση του κυττάρου



$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

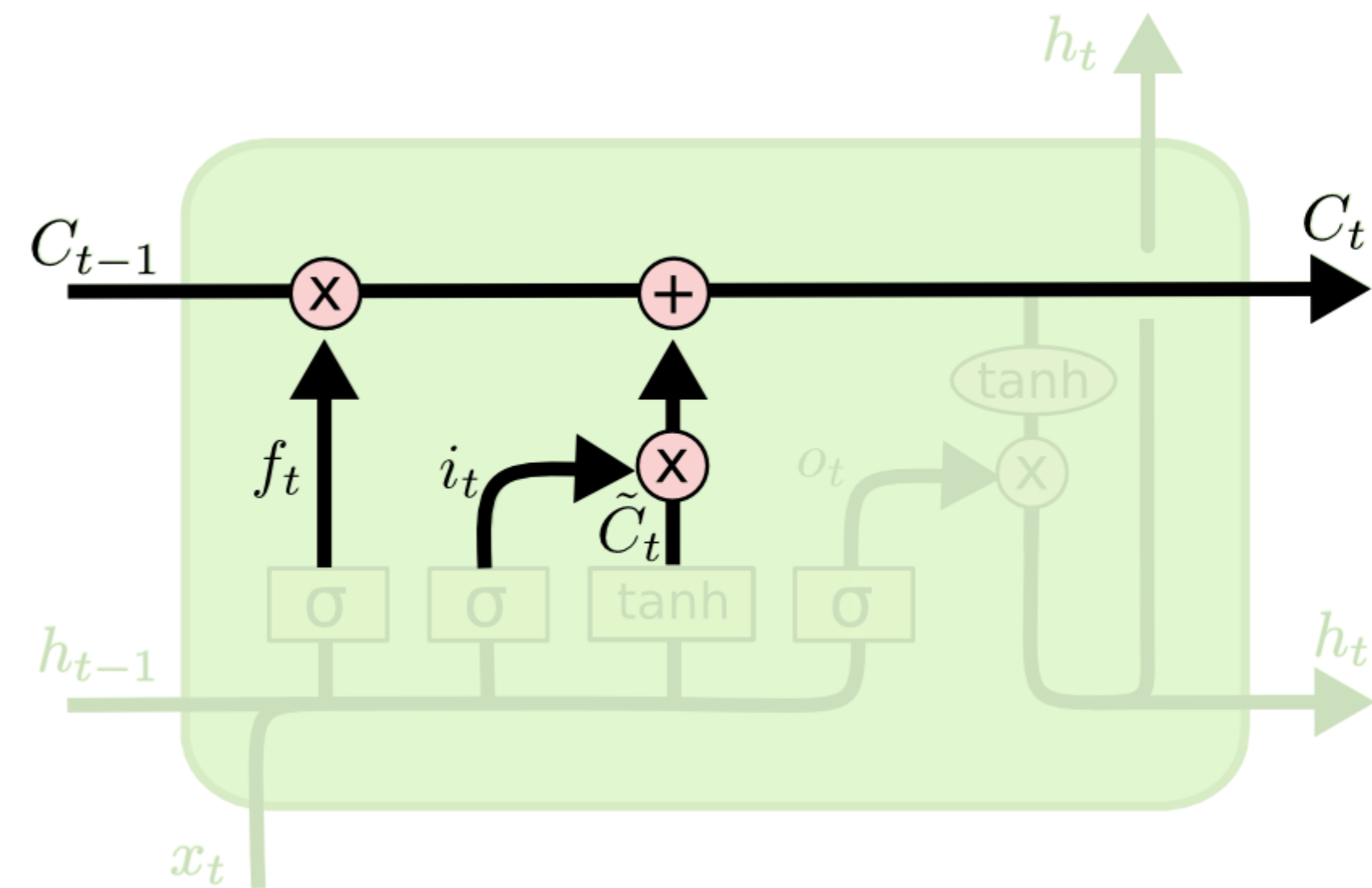
Διάνυσμα νέων υποψήφιων τιμών που θα μπορούσαν να προστεθούν στην κατάσταση: έσπρωξε μέσα από το Tanh για να γίνει [-1,1]

[ΠΗΓΗ](#)





LSTM στρώματα



Ενημέρωση της παλιάς κατάστασης κυττάρων στη νέα κατάσταση

Ελέγχει τι πρέπει να ξεχάσει

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

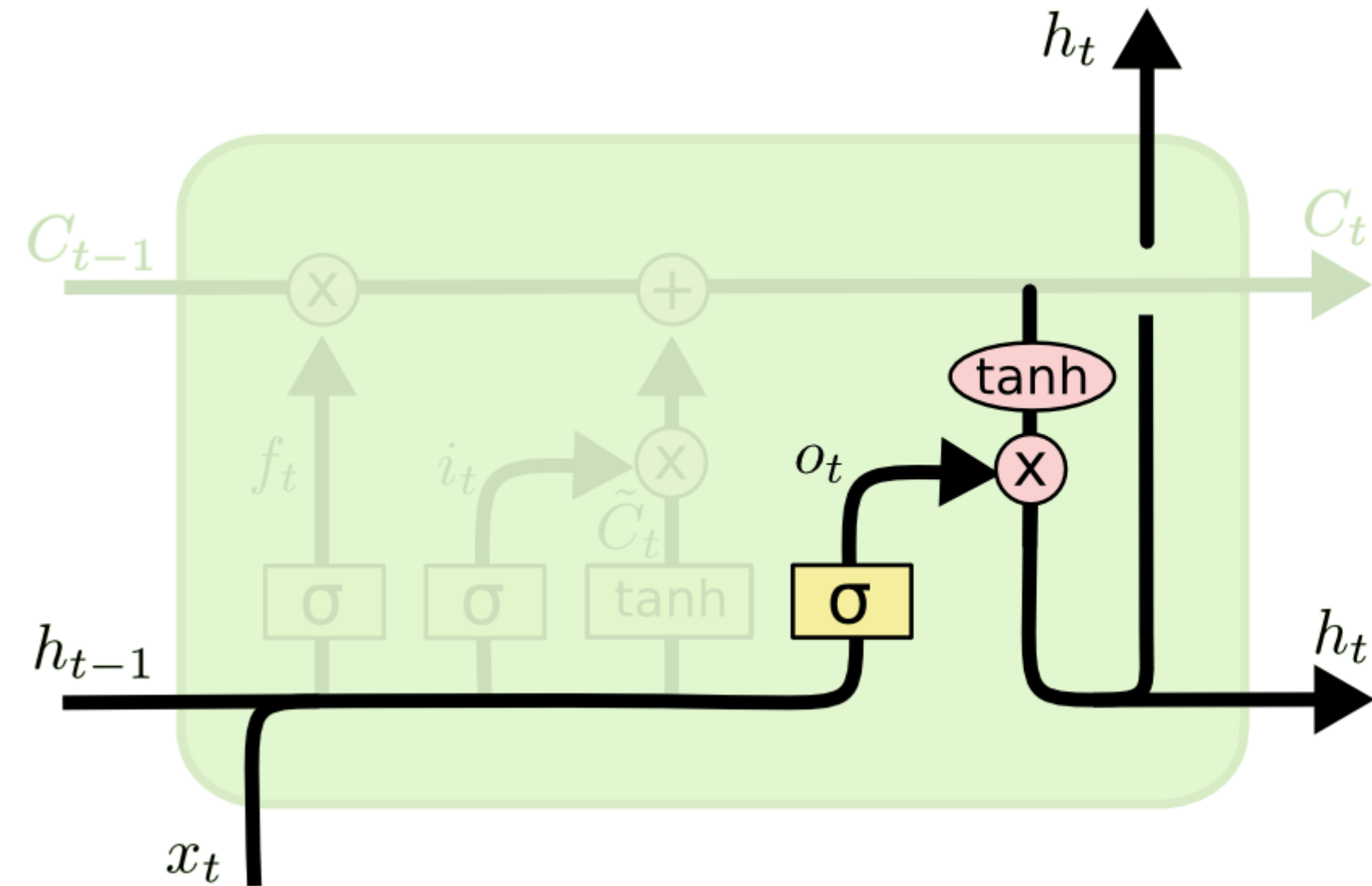
Ελέγχει τι να προσθέσετε

[ΠΗΓΗ](#)





LSTM στρώματα



Πύλη παραγωγής: αποφασίστε ποιο μέρος της κατάστασης του κυττάρου στην έξοδο

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

Κρυφή κατάσταση: κατάσταση κυττάρων μέσω tanh για να γίνει [-1, 1] που περιβάλλεται από την πύλη εξόδου

[ΠΗΓΗ](#)





Εφαρμογές LSTM

- Έλεγχος ρομπότ
- Πρόβλεψη χρονοσειρών
- Αναγνώριση ομιλίας
- Ρυθμική μάθηση
- Μουσική σύνθεση
- Γραμματική μάθηση
- Αναγνώριση γραφής
- Αναγνώριση της ανθρώπινης δράσης
- Σχεδιασμός φαρμάκων
- ...





Δεδομένα κειμένου





Αντιπροσώπηση λέξεων

$V = [a, aaron, \dots, zulu]$

1-hot encoding

Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$

$|V| = 10,000$

Αδυναμία: Δεν γενικεύεται καλά στις λέξεις!

Θέλω ένα ποτήρι πορτοκάλι ____
 Θέλω ένα ποτήρι μήλο ____

Η απόσταση μεταξύ οποιουδήποτε ζεύγους αυτών των λέξεων θα είναι η ίδια





Word embeddings

- Μετατροπή μιας θερμής αναπαράστασης σε μια μητριάσιμένη, δηλαδή, μια αριθμητική διανυσματική αναπαράσταση σε έναν προκαθορισμένο N-διάστατο χώρο
 - Αυτή η μεταμόρφωση μαθαίνεται
 - Κάθε λέξη έχει μια μοναδική διανυσματική αναπαράσταση
- Οι ενσωματωμένες λέξεις προσπαθούν να αποτυπώσουν τη σημασιολογική, συμφραζόμενη και συντακτική σημασία κάθε λέξης με βάση τη χρήση αυτών των λέξεων σε προτάσεις
 - Λέξεις που έχουν παρόμοια σημασιολογική και συμφραζόμενη έννοια έχουν επίσης παρόμοιες διανυσματικές αναπαραστάσεις

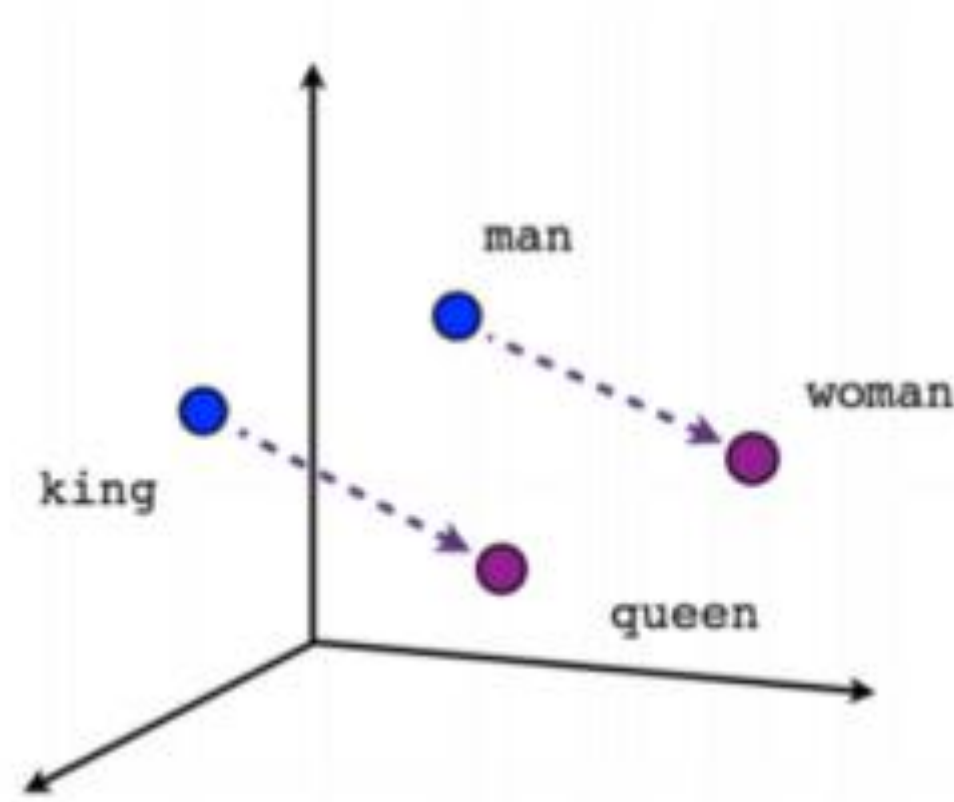
Δημοφιλή μοντέλα: Word2Vec, Glove, ELMo



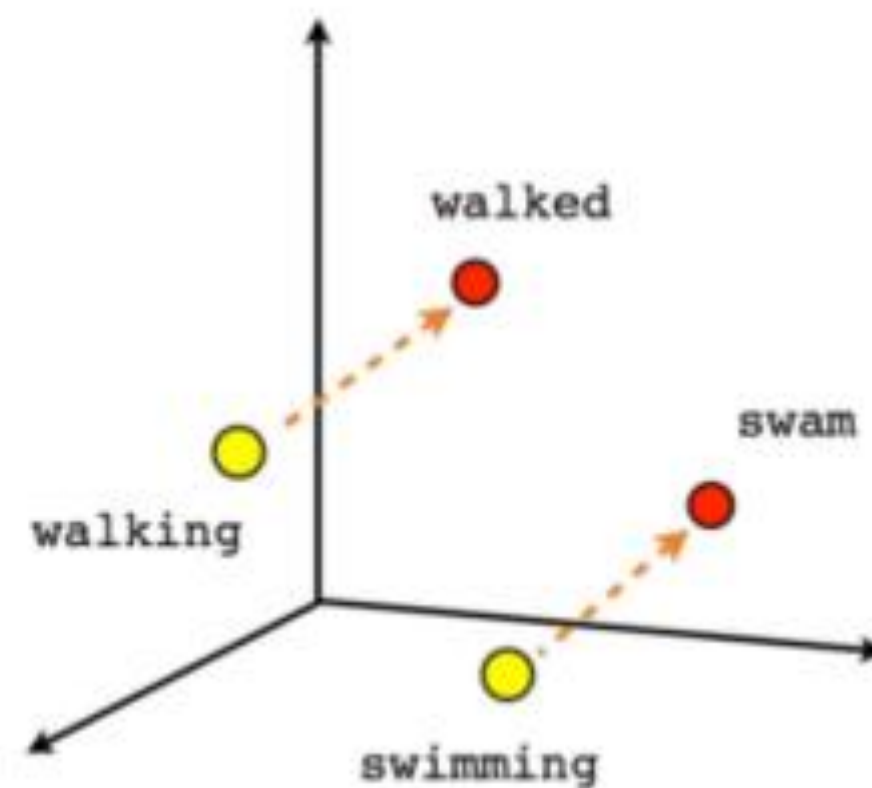


Ιδιότητες των word embeddings

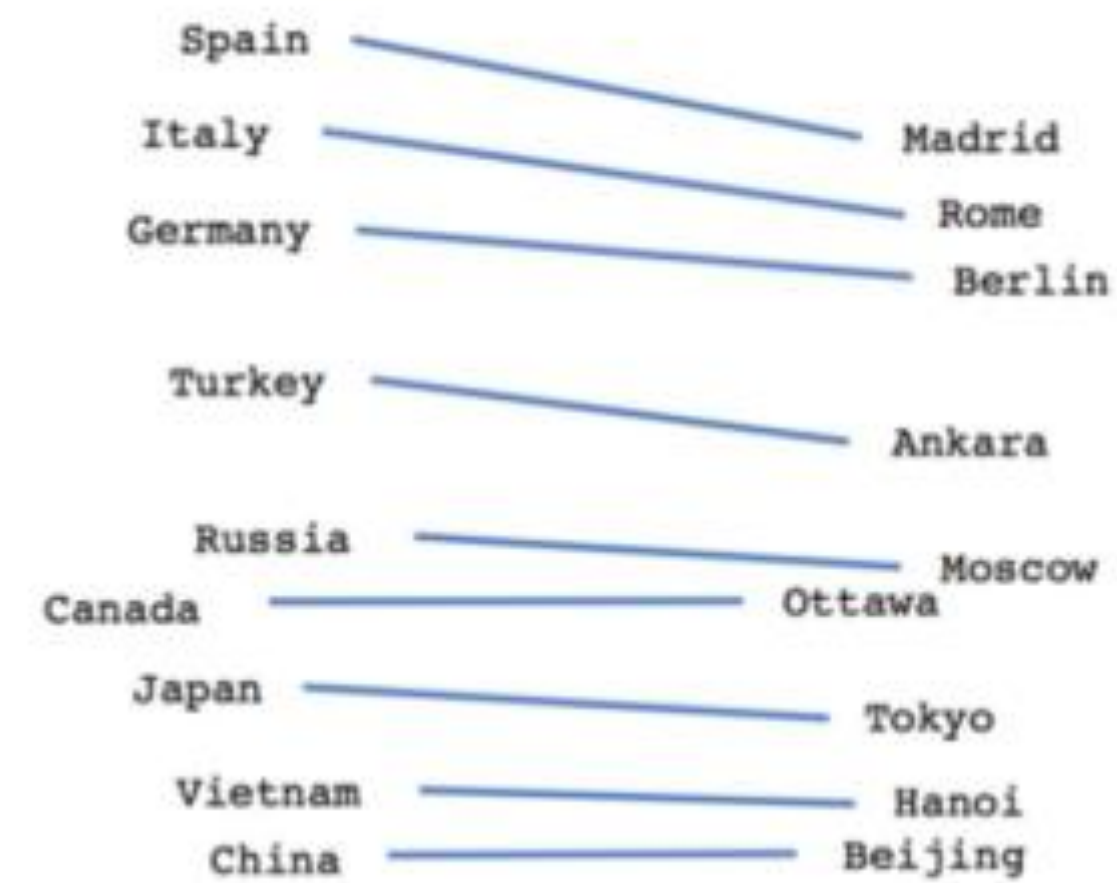
$$\text{Word2Vec}[\text{"King"}] - \text{Word2Vec}[\text{"man"}] + \text{Word2Vec}[\text{"woman"}] = \text{Word2Vec}[\text{"Queen"}]$$



Male-Female



Verb tense



Country-Capital





Εφαρμογές

- Αναγνώριση κατονομαζόμενης οντότητας
- Ταξινόμηση συναισθημάτων
- Ταξινόμηση εγγράφων
- Ανάλυση των απαντήσεων της έρευνας
- Ανάκτηση πληροφοριών
- Μετάφραση γλώσσας
- Ανίχνευση spam
- Συστήματα συστάσεων
- ...



MAI4CAREU

Master programmes in Artificial
Intelligence 4 Careers in Europe



Σας ευχαριστούμε



Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action
No 2020-EU-IA-0087, co-financed by the EU CEF Telecom
under GA nr. INEA/CEF/ICT/A2020/2267423

