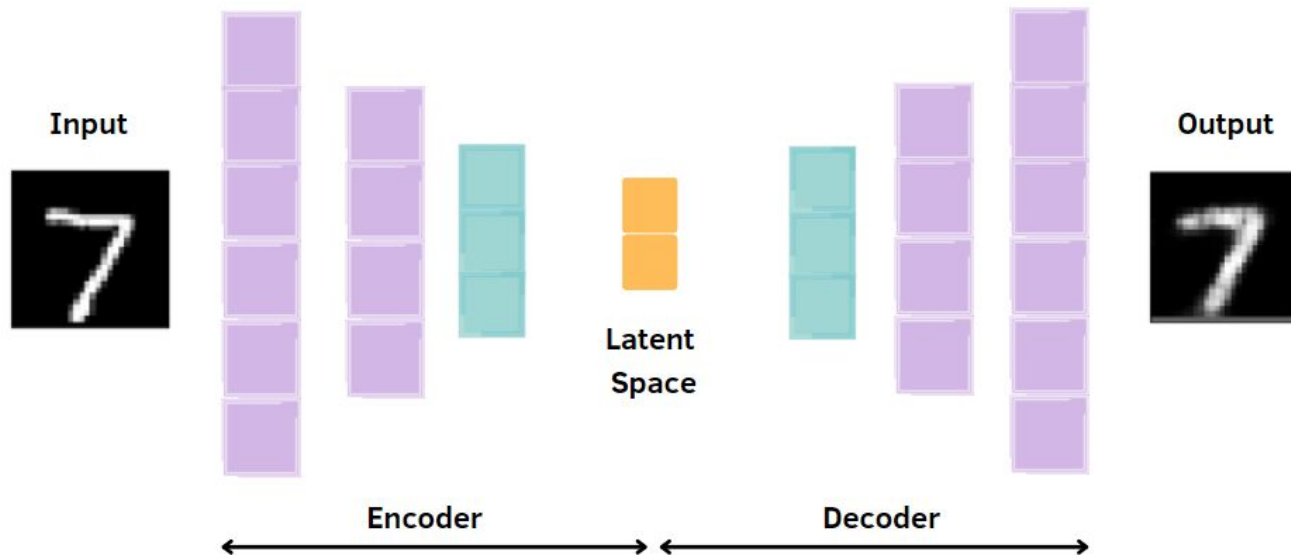


# Autoencoders and CNN

Συμπίεση πληροφορίας και συνέλιξη σε 2Δ και 1Δ δεδομένα

# Autoencoders

Μαθαίνουν να φτιάχνουν την είσοδο, συμπιέζοντας την πληροφορία

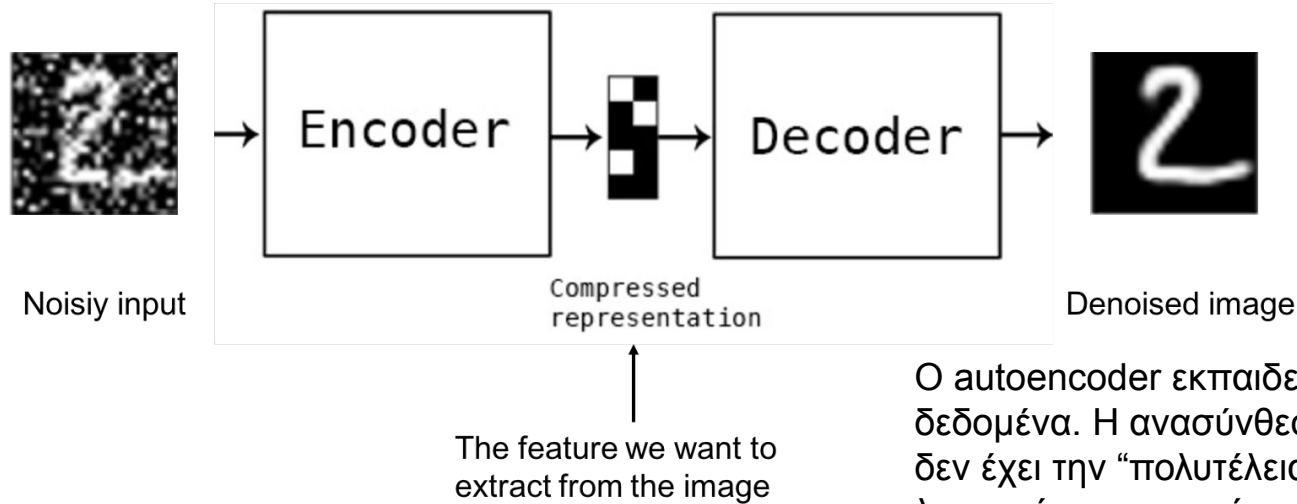


# Σημειώσεις

Κατά την εκπαίδευση:

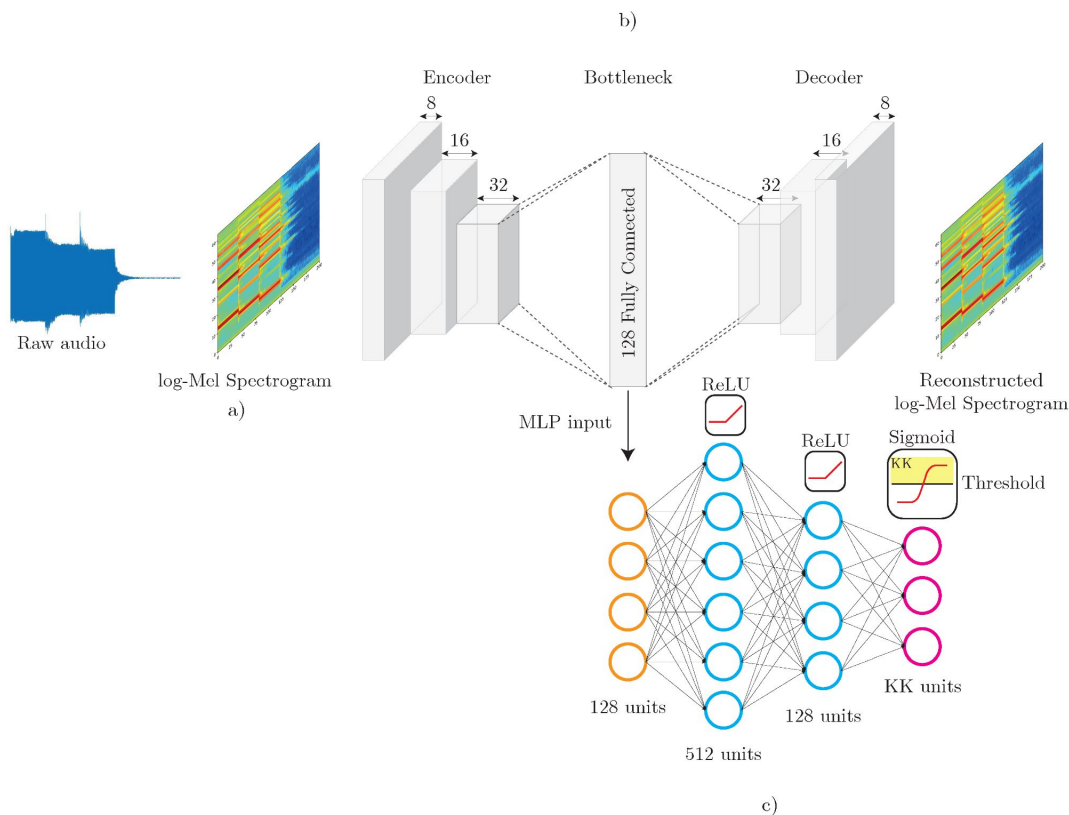
- **Encoder**: μαθαίνει να **συμπιέζει** την πληροφορία με κατάλληλο τρόπο για να χωρέσει βέλτιστα σε λίγες διαστάσεις (θυμίζει PCA, MDS, κτλ).
- **Decoder**: μαθαίνει να **συνθέτει/δημιουργεί** κατάλληλα δεδομένα βάσει εισόδου μικρότερων διαστάσεων.
- Το στρώμα ελάχιστης διάστασης (στο κέντρο) καλείται **latent space** ή **bottleneck**.
- Αν οι ενεργοποιήσεις είναι **γραμμικές** (που δεν έχει νόημα για autoencoders, άρα πρέπει να χρησιμοποιούμε μη γραμμικές), τότε το **bottleneck** συγκλίνει στα αποτελέσματα του **PCA**. Π.χ., βάλουμε 2 νευρώνες στο bottleneck και γραμμικές ενεργοποιήσεις παντού, τότε οι τιμές στους 2 νευρώνες θα συγκλίνουν στις 2 πρώτες διαστάσεις του PCA.

# Αφαίρεση θορύβου



Ο autoencoder εκπαιδεύεται με μη θορυβώδη δεδομένα. Η ανασύνθεση που κάνει ο decoder, δεν έχει την “πολυτέλεια” να κρατάει λεπτομέρειες, κρατάει απλά τη γενική εικόνα. Οπότε ο θόρυβος στην είσοδο, είναι μια λεπτομέρεια που δεν μπορεί να αναπαρασταθεί στη γενική εικόνα και έτσι προκύπτει ανασύνθεση της γενικής εικόνας, χωρίς θόρυβο.

# Κατηγοριοποίηση, με λίγα επισημειωμένα δεδομένα

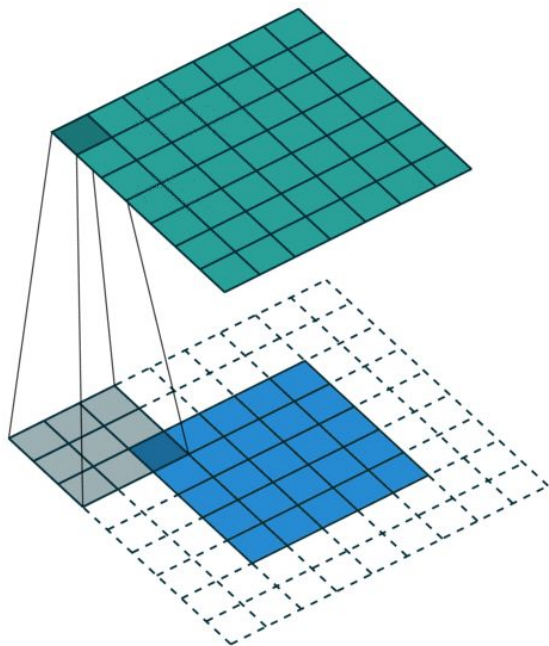


Το κομμάτι του autoencoder (πάνω) εκπαιδεύεται με πολλά δεδομένα. Το κομμάτι του κατηγοριοποιητή (κάτω) εκπαιδεύεται μετά και μόνο με όσα έχουμε επισημειωμένα.

Στη βιβλιογραφία αυτή η κατάσταση καλείται *few-shot learning*. Είναι σύνηθες πλέον να έχουμε πολλά δεδομένα (πχ κομμάτια μουσικής) αλλά λίγα με επισημειώσεις (πχ κομμάτια με *fagotto*). Με το *few-shot learning*, σκοπός είναι να αξιοποιήσουμε πολλά δεδομένα για να βγάλουμε βέλτιστο αποτέλεσμα με τα λίγα επισημειωμένα.

# Συνέλιξη 2Δ: κατάλληλα για εικόνα (πχ spectrogram)

Παράδειγμα φίλτρου 2Δ:  
3x3, βήμα 1, full padding



Άλλο Παράδειγμα φίλτρου 2Δ:  
3x3, βήμα 2, same padding

0 <sub>2</sub>	0 <sub>0</sub>	0 <sub>1</sub>	0	0	0	0
0 <sub>1</sub>	2 <sub>0</sub>	2 <sub>0</sub>	3	3	3	0
0 <sub>0</sub>	0 <sub>1</sub>	1 <sub>1</sub>	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

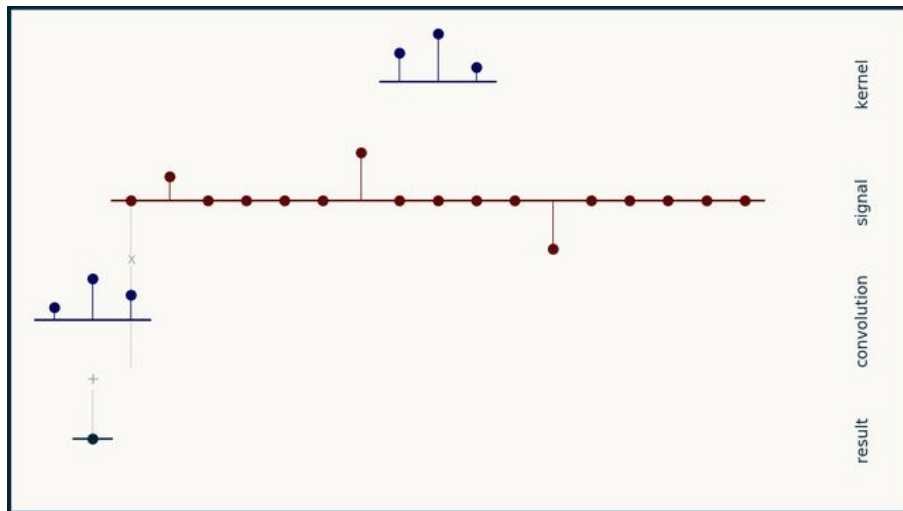
1	6	5
7	10	9
7	10	8

[https://keras.io/api/layers/convolution\\_layers/convolution2d/](https://keras.io/api/layers/convolution_layers/convolution2d/)  
[https://github.com/vdumoulin/conv\\_arithmetic](https://github.com/vdumoulin/conv_arithmetic)

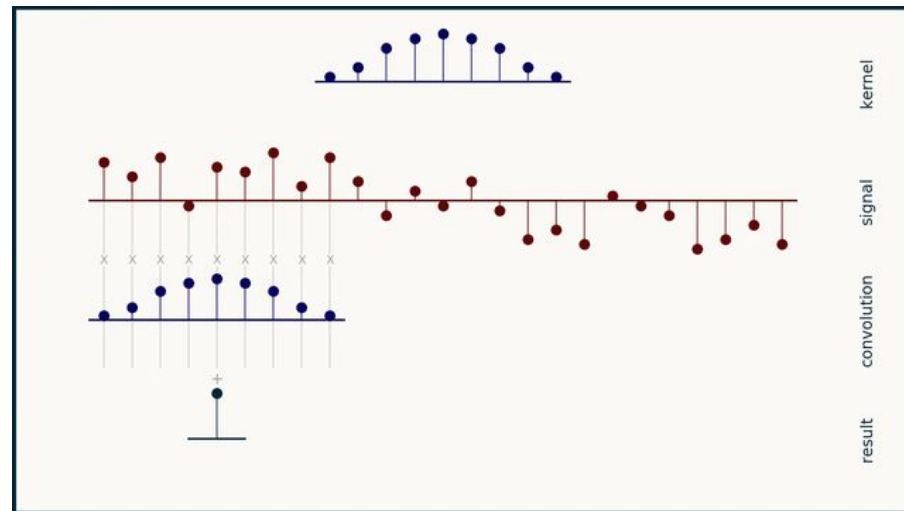
Ένα στρώμα συνελικτικού νευρωνικού δικτύου (CNN) είναι μια συλλογή από διαφορετικά φίλτρα, με τα ίδια χαρακτηριστικά μεγέθους, βηματισμού και padding (ομοίως και στη 1Δ περίπτωση στην επόμενη διαφάνεια).

# Συνέλιξη 1Δ: κατάλληλα για σειρά, πχ για κυματομορφή

Παράδειγμα φίλτρου 1Δ:  
Μέγεθος 3, βήμα 1, full padding



Άλλο παράδειγμα φίλτρου 1Δ:  
Μέγεθος 9, βήμα 2, valid padding



[https://keras.io/api/layers/convolution\\_layers/convolution1d/](https://keras.io/api/layers/convolution_layers/convolution1d/)

[https://e2eml.school/convolution\\_one\\_d.html](https://e2eml.school/convolution_one_d.html)

# Αντίστροφη συνέλιξη - για το decoder κομμάτι

Χρήση φίλτρου για ανασύνθεση σήματος

