université PARIS-SACLAY

TIME SERIES :

DEEP LEARNING ARCHITECTURES

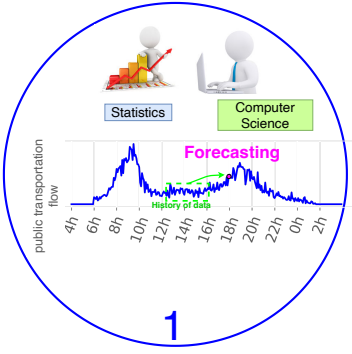
Vincent Guigue

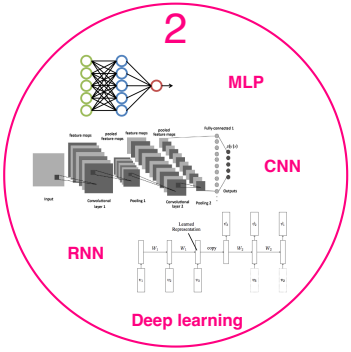
MIA PARIS-SACLAY
EKINOCS

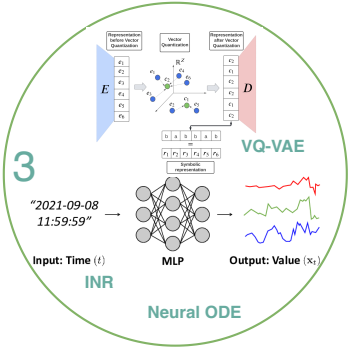
AGROPARISTECH
Institut des Sciences et Industries du Vivant et de l'Environnement

Introduction

Courses organization



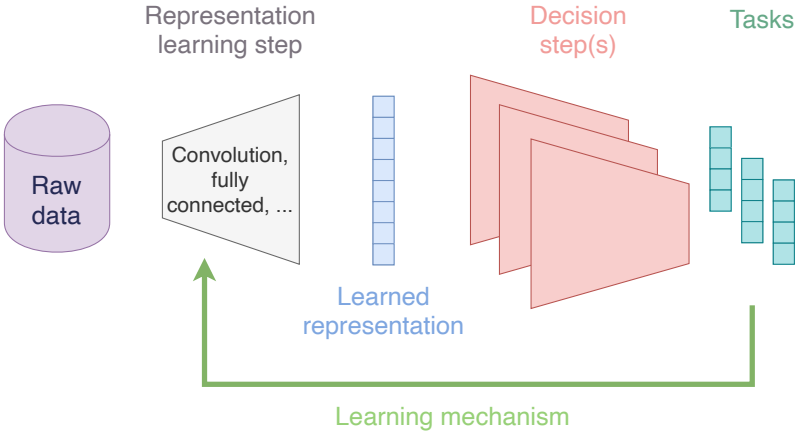




General idea of deep learning for time series

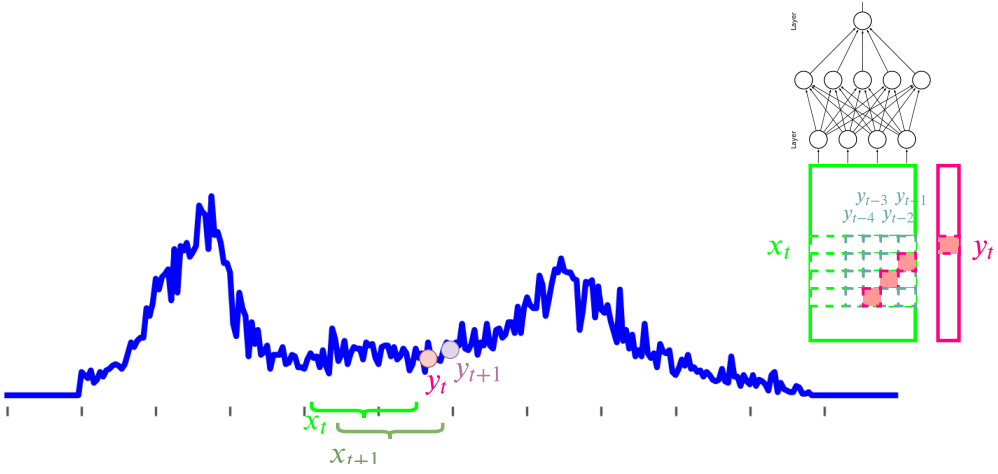
Issues:

- Extracting relevant features
- Representation learning
- Multivariate time series
- Multi-task learning



Historical neural architectures

- TDNN: Time Delay Neural Networks



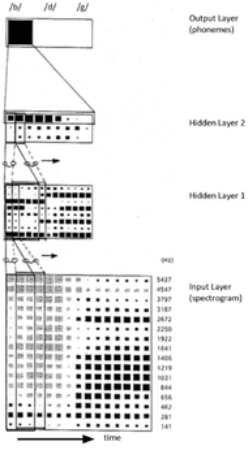
Applications

TDNN: Time Delay Neural Networks

- Originally: Multi Layer Perceptron on lag variables
- By extension: Any neural architecture on a temporal sliding window

Applications:

- Pattern classification:
  - Phoneme classification (speech recognition)
  - Handwriting recognition
- Signal processing
  - Echo and reverberation elimination



A. Waibel et al., IEEE Trans. ASSP, 1989
Phoneme Recognition Using Time-Delay Neural Networks

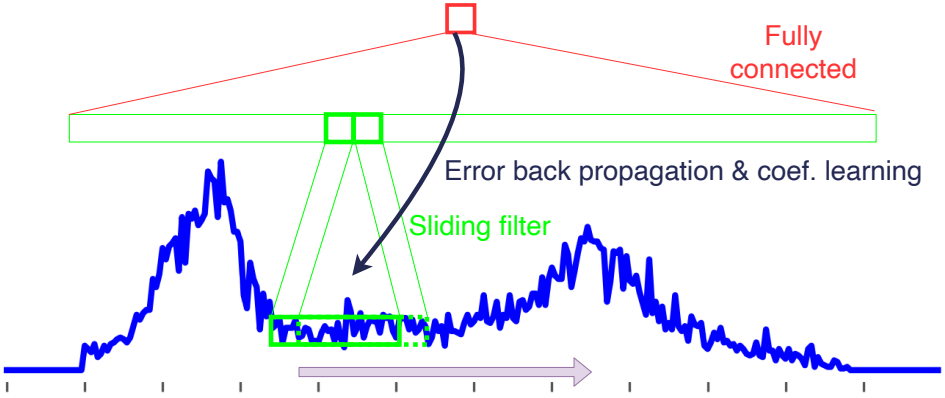
Convolutional Neural Networks

CNN History

One of the first breakthrough in machine learning: zip code recognition in 1989

- In 1989, 50 people in the world were able to set a CNN...
  the others were waiting for SVM !
- In 2010, 5000 people were able to set CNNs... That leads to AlexNet!

Definition of a Convolutional Neural Networks

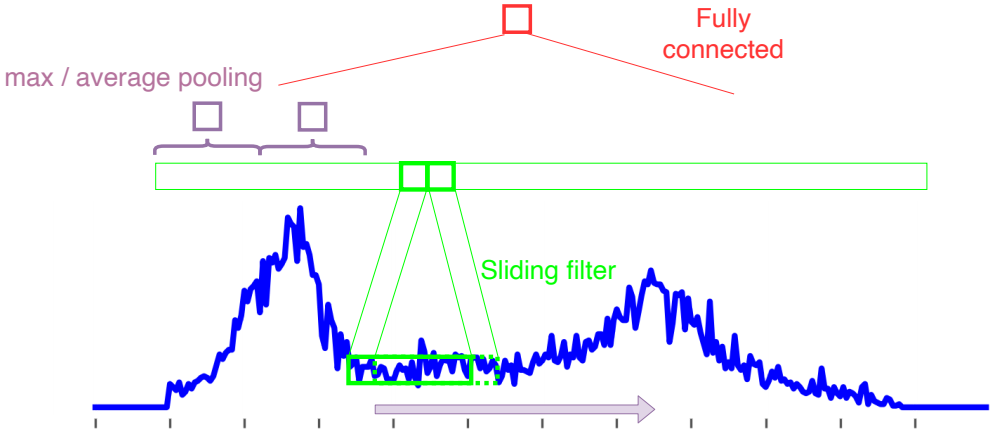


- Convolutional filter (few parameters)
- Signal representation + decision layer (more expensive)

  $\Rightarrow$ Signal classification / pattern detection

Idea: learning to extract relevant features wrt a given task

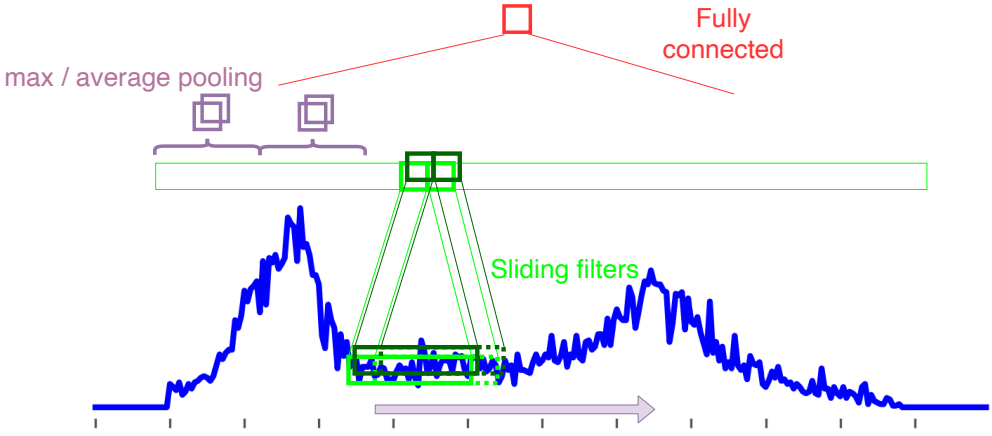
CNN, pooling & multiplication of the filters



Adding a pooling layer:

1. Reduce the cost of the fully connected layer
2. Add (slight but efficient) translation invariance

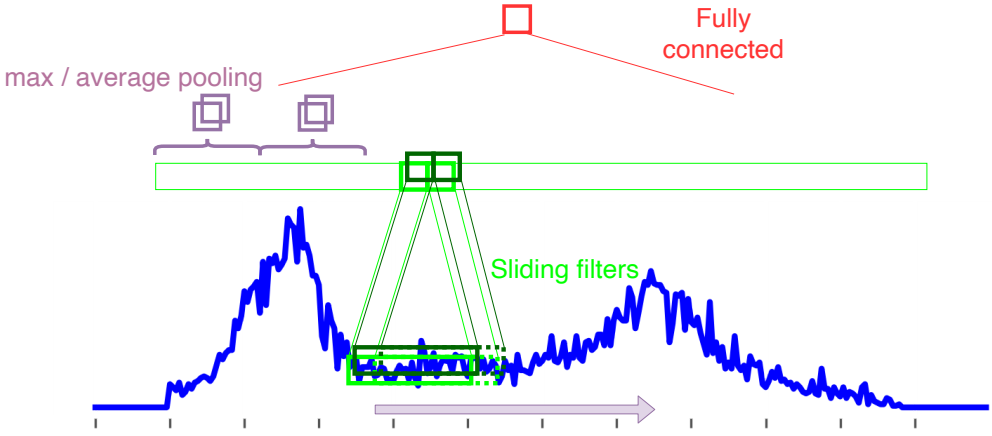
CNN, pooling & multiplication of the filters



Adding a pooling layer:

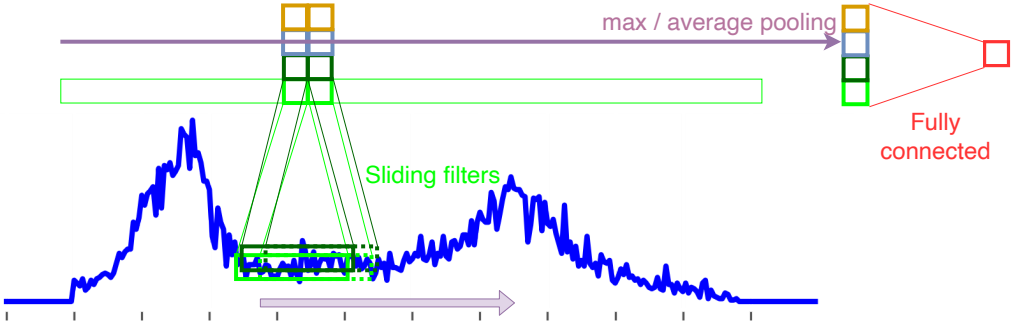
1. Reduce the cost of the fully connected layer
2. Add (slight but efficient) translation invariance

CNN, pooling & multiplication of the filters



Even in 2009 = SVM golden age, CNN > SVM in handwriting reco.
⇒ Investigate variations of the signal in input
⇒ + Translation invariance

CNN & variable length time series

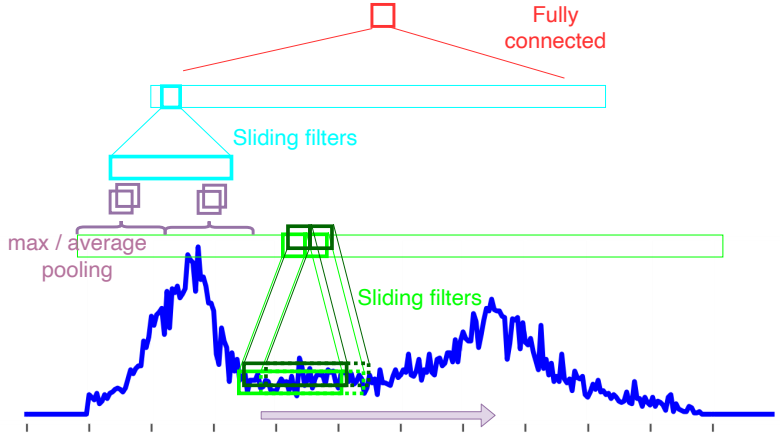


Enlarging pooling operation:

1. Reduce the cost of the fully connected layer
2. Each filter acts as a pattern detector
3. Fixed size signal representation
4. No more temporal descriptors in the representation

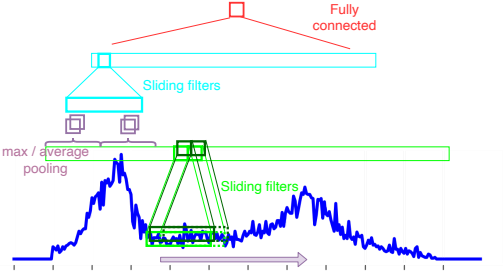
Multi-layer CNN

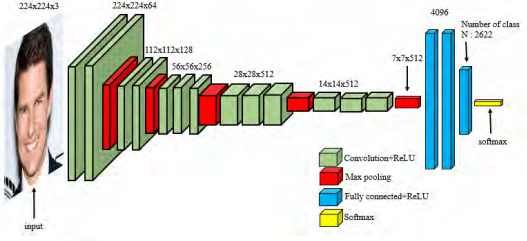
- State of the art in vision architecture
- An in depth explanation of classical internet illustrations

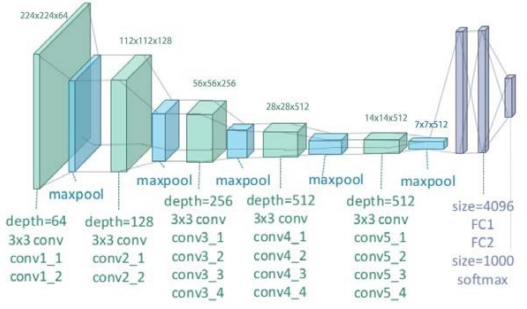


Multi-layer CNN

- State of the art in vision architecture
- An in depth explanation of classical internet illustrations

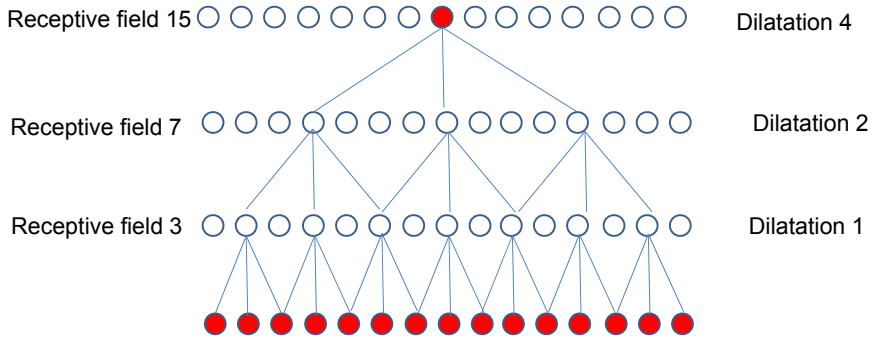






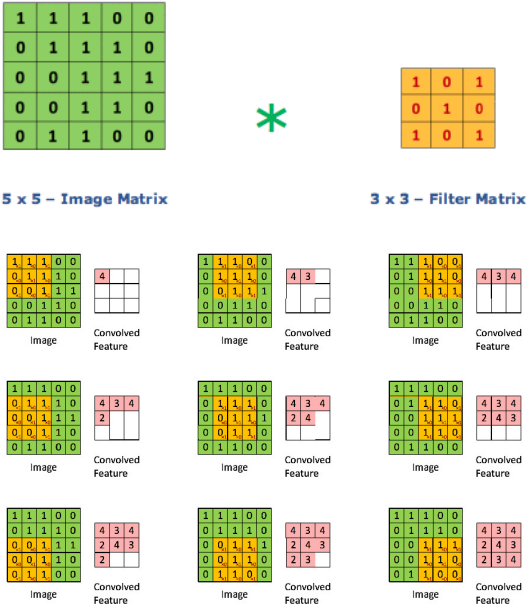
Dilated CNN [Yu 2016]

To analyse efficiently multi-scale aspects of a signal:



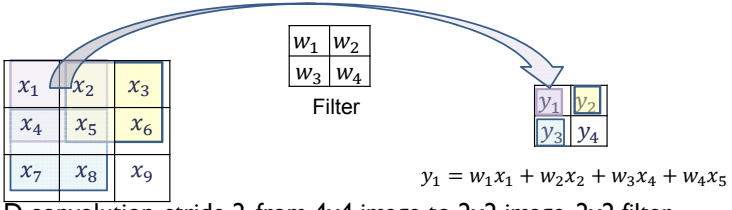
- Less computation
- More efficiency

2D Convolution



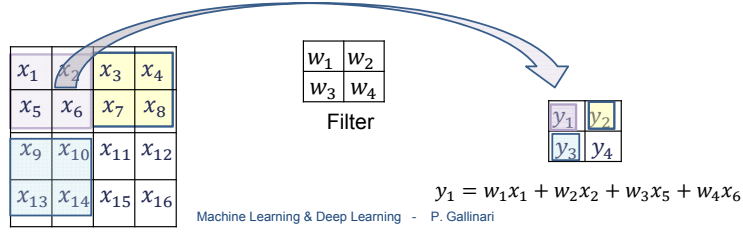
2D CNN: stride

- 2D convolution, stride 1, from 3x3 image to 2x2 image, 2x2 filter



$$y_1 = w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5$$

- 2 D convolution, stride 2, from 4x4 image to 2x2 image, 2x2 filter



$$y_1 = w_1x_1 + w_2x_2 + w_3x_5 + w_4x_6$$

+ pooling on spatial 2D windows

Pooling Layer

Pooling (or subsampling) : Dimensionality reduction of the output

- Max Pooling : takes the *max* over a window
- Average Pooling : takes the mean
- Sum Pooling : sum ...

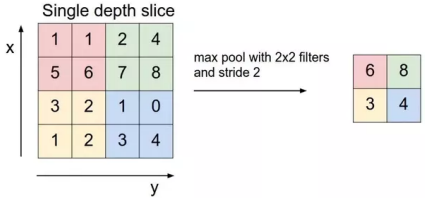
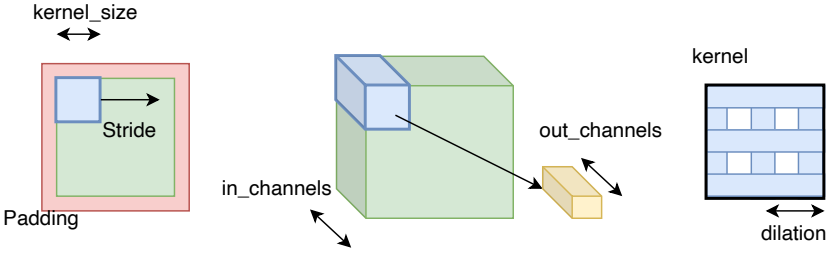


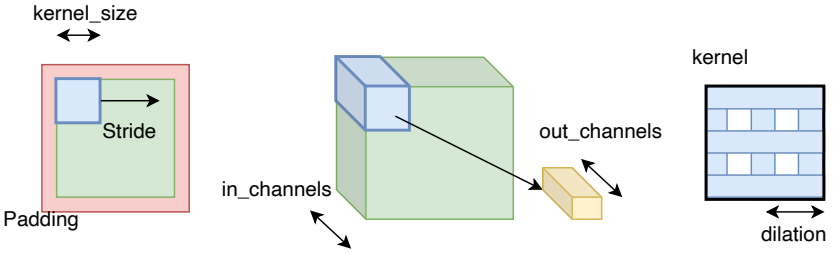
illustration : https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

Parametrization summary




```
1     class ConvNet(nn.Module):
2     # argument sur le choix du pooling
3     def __init__(self,pooling = nn.MaxPool2d):
4         super().__init__()
5         # Convolution 5*5, 16 filtres
6         self.conv1 = nn.Conv2d(in_channels=1, out_channels=16,
7             kernel_size=5, stride=1, padding=0, dilation=1)
8         # Max pooling 3x3
9         self.pool1 = pooling(kernel_size=3, stride=1)
10        ...
```

Input/Output dimensions



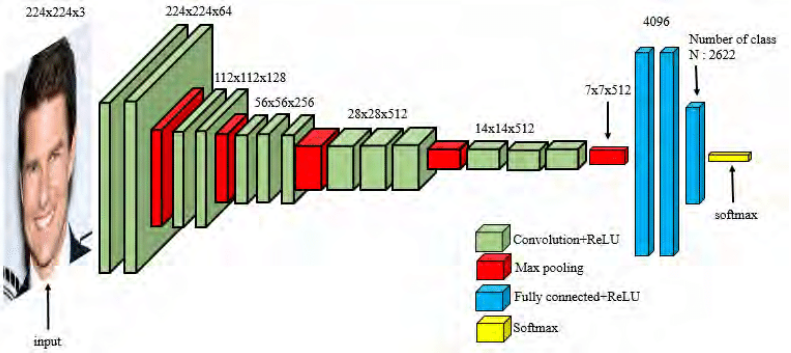
This formula should give you the output size per channel

$$\frac{(W - K + 2P)}{S} + 1$$

- W is the input volume
- K is the Kernel size
- P is the padding
- S is the stride

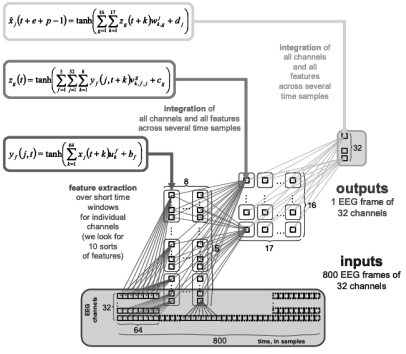
Deep CNN

Image analysis / object recognition: AlexNet, VGG, ...



Use case: Seizure detection

An application in signal classification: detecting seizure in EEG



CNN : a very elegant (& efficient) way to deal with multivariate time-series

P.W. Mirowski et al., IEEE, 2008
Comparing SVM and Convolutional Networks for Epileptic Seizure Prediction from Intracranial EEG

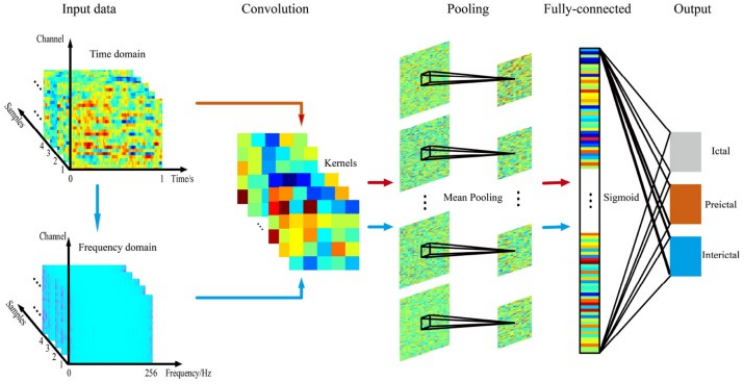
M Zhou et al., F. in Neuroinformatics, 2018
Epileptic Seizure Detection Based on EEG Signals and CNN

Use case: Seizure detection

An application in signal classification: detecting seizure in EEG

Recent architectures are mainly based on

- Time frequency decomposition
- Image analysis



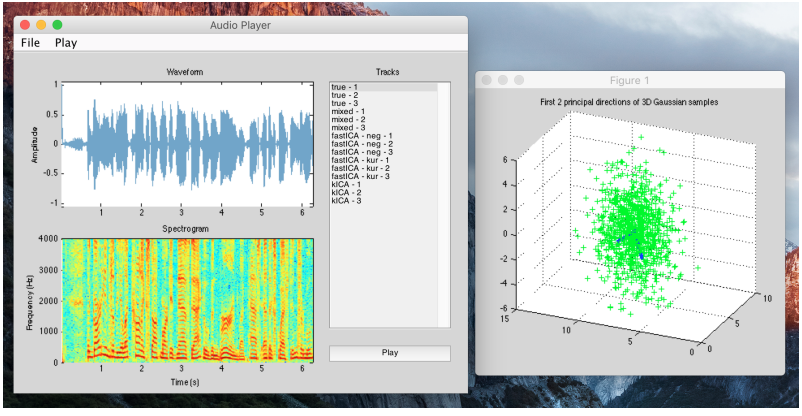
P.W. Mirowski et al., IEEE, 2008
Comparing SVM and Convolutional Networks for Epileptic Seizure Prediction from Intracranial EEG

M Zhou et al., F. in Neuroinformatics, 2018
Epileptic Seizure Detection Based on EEG Signals and CNN

CNN & Time-Frequency representation

The example of source separation (that makes great progress over the last 5 years)

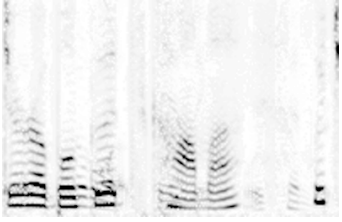
Original problem: ICA (independant componant analysis)
SVD algorithm (unsupervised) in time or time frequency domain:

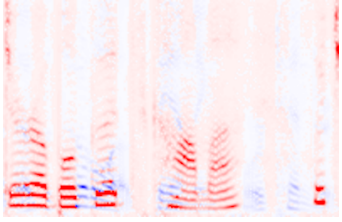


CNN & Time-Frequency representation

The example of source separation (that makes great progress over the last 5 years)

New Problem:
A supervised classification problem in the time frequency domain





CNN & Time-Frequency representation

The example of source separation (that makes great progress over the last 5 years)

Typical architecture: dilated CNN

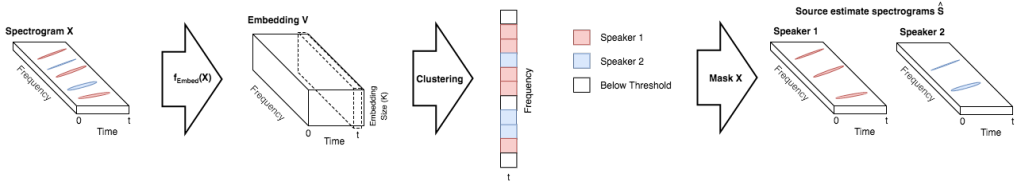
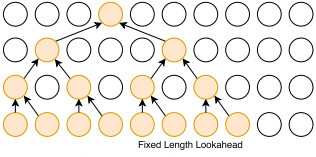
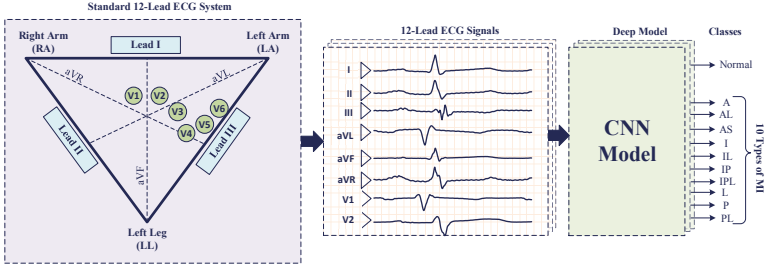


Figure 2: Overview of the source separation process.



Other applications with CNN on signals

- Acoustic scene classification
- Music style classification
- EEG / ECG analysis
  - But only on long signals ($\approx$ 5 minutes)
  - ... Brain computer interface relies on Riemaniann geometry



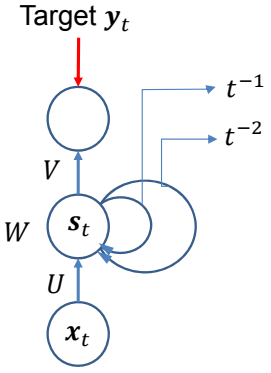
Recurrent Neural Networks

RNN History

- Appears in the 1990'
- Beautiful architecture... But hard to train

$\Rightarrow$ no real world application until 2006 (A. Graves)

- Today state of the art in:
  - Speech / handwriting transcription
  - Machine translation
  - NLP: language understanding / generation



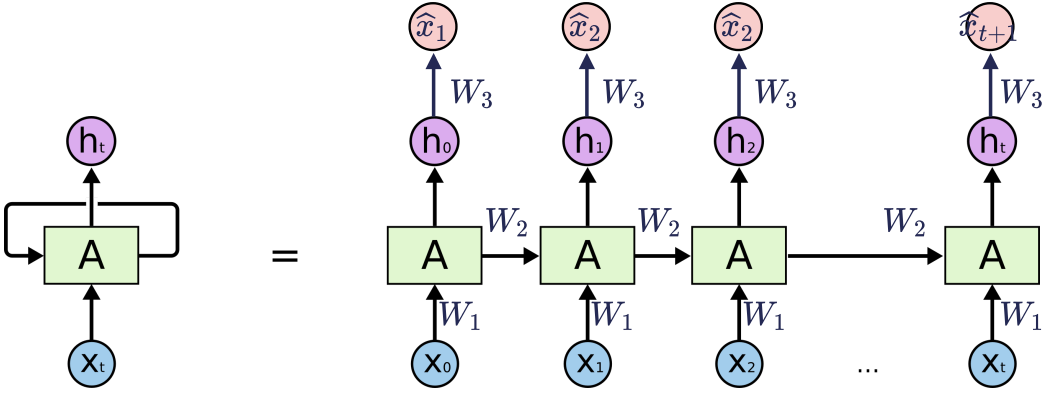
All weights learned

$$s_t = f(W s_{t-1}) + U x_t$$

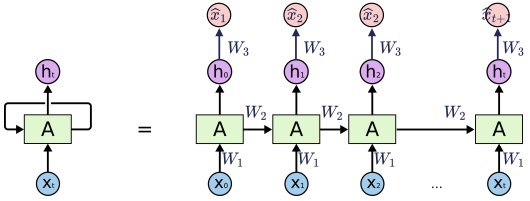
RNN... For which purpose?

- Predicting the next step (from h_T)
- Generation = prediction (in loop)
- Classifying a sequence (from h_T)
- Detecting an event (at every h_t)

Bi-RNN: use h_t & h_t^R
(h_T & h_1^R = caract. whole sequence)



Learn to generate



- Teacher forcing:

$$\hat{x}_{t+1} = g(h_t), \qquad h_{t+1} = f_W(h_t, x_{t+1})$$

- Free Generation

$$\hat{x}_{t+1} = g(h_t), \qquad h_{t+1} = f_W(h_t, \hat{x}_{t+1})$$

In practice:

- Start with teacher forcing (more stable) $\Rightarrow$ then switch to free generation
- Sequential learning also = more complicated solution

$\Rightarrow$ **Reinforcement Learning**

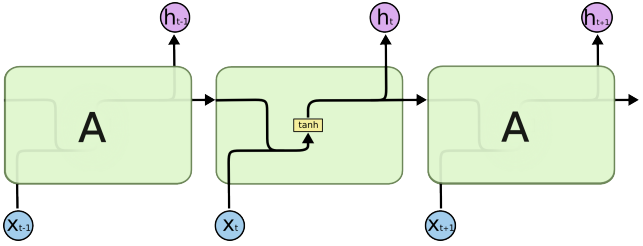
RNN In practice

```
# Define the model
model = torch.nn.RNN(input_size=100, hidden_size=50,
                     num_layers=1,
                     nonlinearity='tanh', bias=True)

# data (x_1,...,x_T)
# Seq. length T=23 & batch of size 10
# Input x_t \in R^n dimension 100
seq_x = torch.rand(23, 10, 100)

# seq_h is a tensor (23, 10, 50)
# For classification purpose, we use last_h
seq_h, last_h = model(seq_x)
```

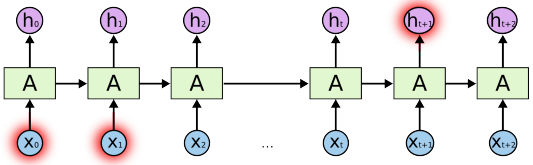
Convention

Tensor of size **time x sample x representation**

Overcoming gradient issues: Gated unit & LSTM



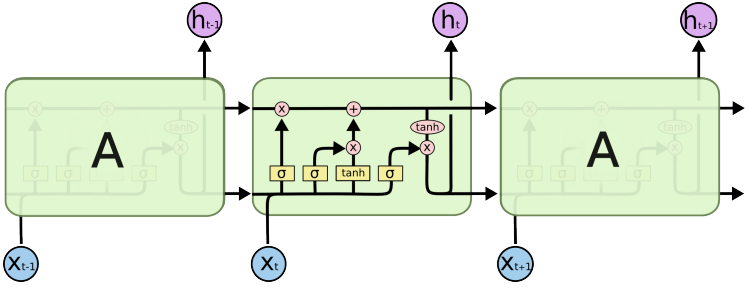
Gradient vanishes & long term dependancies are not modeled....



Chris Olah's blog `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

Overcoming gradient issues: Gated unit & LSTM

The phenomenon has been understood & (partially) overcome:
Neurons **learn** what should be **kept in memory** and what should be **forgotten**

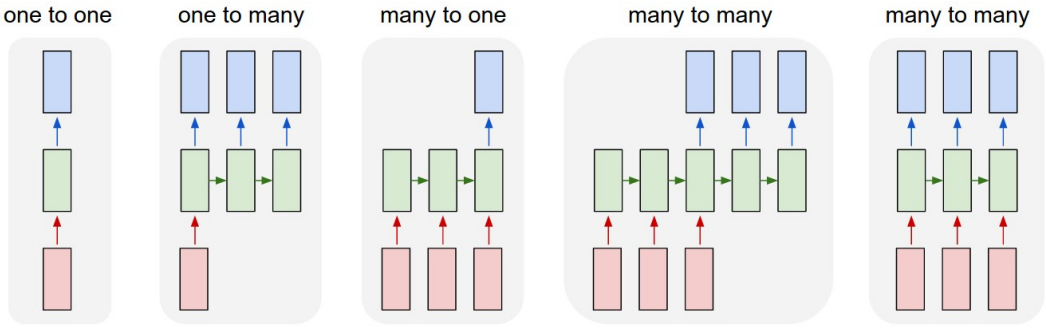


Gated architecture

S. Hochreiter, J. Schmidhuber, Neural computation 1997
Long short-term memory

Chris Olah's blog `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

RNN architecture : different settings

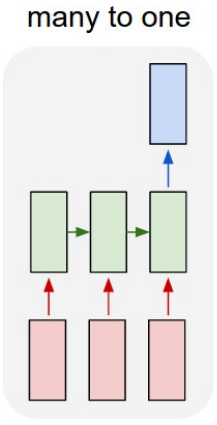


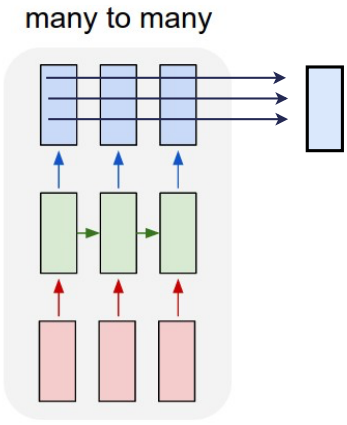
- One to many : image annotation
- many to one : signal classification
- many to many : POS/NER tagging, sequence annotation
- seq to seq : machine translation

Karpathy's blog `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`

Signal classification / many to one architecture

Architecture variation

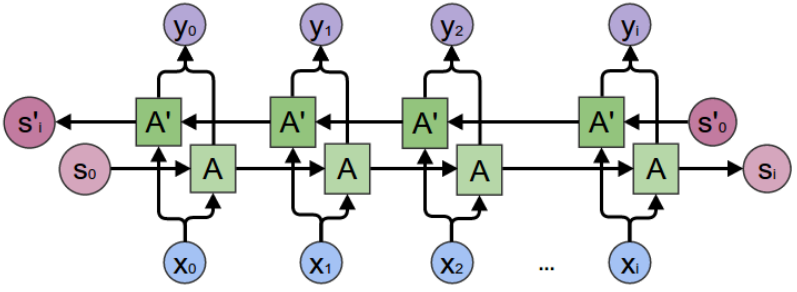




State Of The Art to represent a sequence : Bi-LSTM

LSTM

+ Sequential modeling
− Sequential dependencies ! = partial modeling

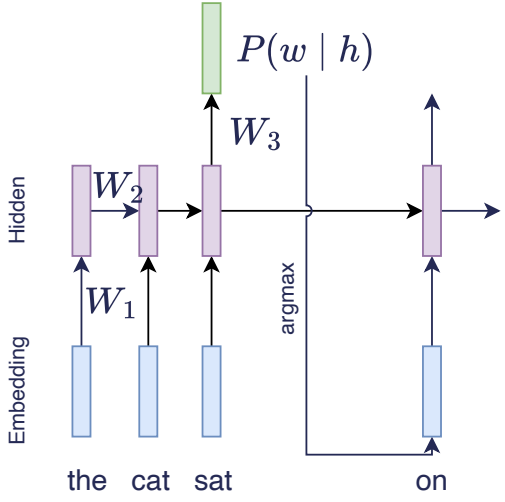


Bi-dimensional representation $[S_1, S'_1]$ is more powerful representation of the sentence S than each single representation.

Classical notation: $\mathbf{s} = [\overrightarrow{\mathbf{s}}, \overleftarrow{\mathbf{s}}]$

Generative Architectures

Basic data generation



- $W_3 \approx$ vocabulary classification
- Several variation: argmax / sampling...

$\Rightarrow$ reduce vocabulary

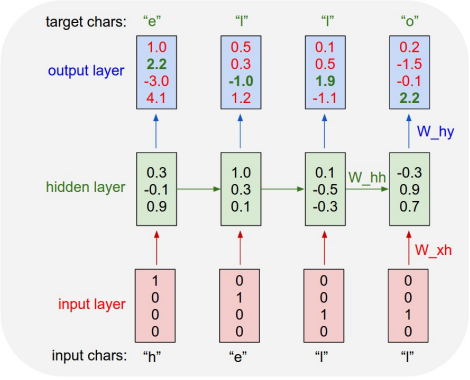
- at the letter scale (dim ≈ 60)
- byte pair encoding $\Rightarrow$ most freq. ngrams of letters

- Iteration at $t+1$ only requires:
  - x_{t+1}
  - h_t (that encode the whole sequence of previous words)

Word prediction

- Also for continuous generation: classifier framework $\Rightarrow$ regressor

Karpathy's demonstration on char2char




```
class RNN:
  # ...
  def step(self, x):
    # update the hidden state
    self.h = np.tanh(np.dot(self.W_hh, se
    # compute the output vector
    y = np.dot(self.W_hy, self.h)
    return y
```

+ multilayer architecture:

```
y1 = rnn1.step(x)
y = rnn2.step(y1)
```

A. Karpathy's blog : The Unreasonable Effectiveness of Recurrent Neural Networks
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Learning a sequence model & generate new data:

Training a model to predict the next character given a sequence:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt    h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

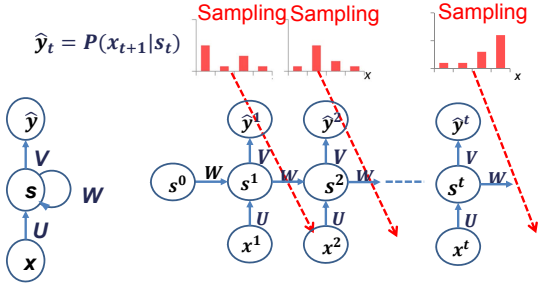
train more

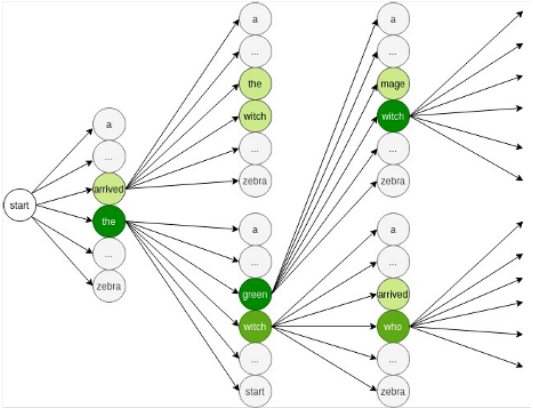
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Beam Search





- Sampling ability $\Rightarrow$ tree exploration
- Keeping the best sequence (likelihood criterion)

Karpathy's demonstration on char2char

Sample of *Shakespeare* generation

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.
```

A. Karpathy's blog : The Unreasonable Effectiveness of Recurrent Neural Networks
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Karpathy's demonstration on char2char

Wikipedia sample

```
Naturalism and decision for the majority of Arab countries' capitalide was
grounded by the Irish language by [[John Clair]], [[An Imperial Japanese
Revolt]], associated with Guangzham's sovereignty.  His generals were the
powerful ruler of the Portugal in the [[Protestant Immineners]], which
could be said to be directly in Cantonese Communication, which followed a
ceremony and set inspired prison, training.  The emperor travelled back to
[[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica,
unsuccessful fashioned the [[Thrales]]
```

A. Karpathy's blog : The Unreasonable Effectiveness of Recurrent Neural Networks
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

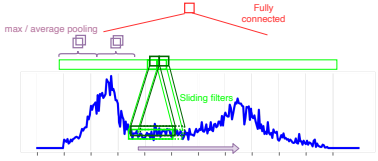
Karpathy's demonstration on char2char

Linux code generation

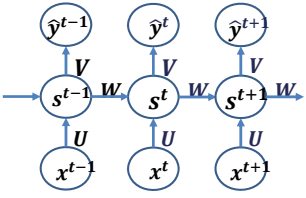
```
*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
```

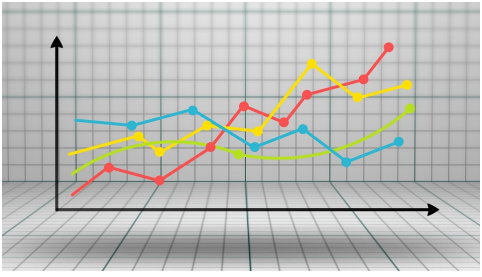
Various neural architectures to deal with multivariate time series

- CNN
  - Different filters for each channel / same filters



- RNN :
  - no problem to give a vector as input at each time step





Conclusion

Benefits of deep learning architecture for time series modeling

- Efficient against noise
- Extract very relevant features
  - & relevant pattern with translation invariance
- Great software framework with GPU abilities
- Plasticity of the architectures
  - Naturally adapted to complex inputs
    - Variable length signals
    - Multivariate signals
  - New opportunities in signal generation / classification / understanding

The question of pre-training

- Pre-training language model is a great advance for many application
  - Application with small corpus
  - Fine tuning
- Pre-training vision model is a great advance for many application
  - Recognizing cats on images improve the performance in detecting default on breaking pads...

$\Rightarrow$ It gives us a common knowledge of the world.

The question of pre-training

- Pre-training language model is a great advance for many application
  - Application with small corpus
  - Fine tuning
- Pre-training vision model is a great advance for many application
  - Recognizing cats on images improve the performance in detecting default on breaking pads...

$\Rightarrow$ It gives us a common knowledge of the world.

$\Rightarrow$ **Is is possible to learn a language model for signals?**