CNN and unrolling algorithms

Laure Blanc-Féraud

DR CNRS
Laboratoire I3S
Morpheme group (I3S - INRIA Sophia Antipolis – iBV)

Introduction: Deep learning revolution

- Availability of large-scale training data sets (ex. from internet content)
- Accessibility of powerful computational resources (major advances in microelectronics)
- Advance in Neural network research:
  - Effective network architectures
  - Efficient training algorithms

- Unprecedented success of deep learning in computer vision, pattern recognition, speech processing.
- In image processing: classification, image recognition

Introduction

- Learning-based approaches provide an interesting alternative to traditional model-based analytic methods.

- Model-based approach: designed by analysing the physical processes, building mathematical models.
- Learning-based approach: automatically discover information from the data by optimizing network parameters from real-word training samples.

Introduction: DNN advantages

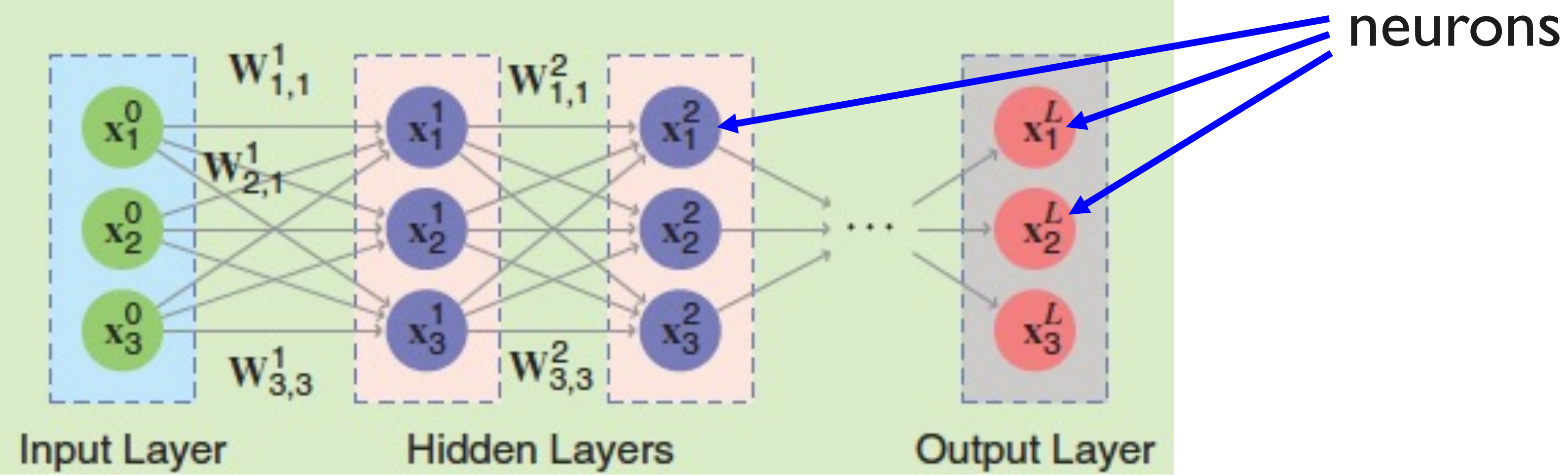
- deep learning approaches attempt to automatically discover model information

- deep neural networks (with many layers) include a large number of parameters (possibly millions) and are thus capable of learning complicated mappings, which are difficult to design explicitly.

- during inference, processing through the network layers can be executed very fast.

Neural Networks review: the Multilayer Perceptron

- Architecture mimicks the human recognition system, generalises the perceptron algorithm to multiple layers.


neurons

- Fully connected layer architecture: all neurons of a layer are connected to all neurons of the next layer except for the output layer).

Neural Networks review: the Multilayer Perceptron

- Analytically, in the *l*th layer, the relationship between the neurons is expressed by

$$\mathrm{x}_i^{l+1} = \sigma\left(\sum_j \mathrm{W}_{ij}^{l+1}\mathrm{x}_j^i + \mathrm{b}_i^{l+1}\right)$$

where W^{l+1} are the weights; b^{l+1} are the biaises

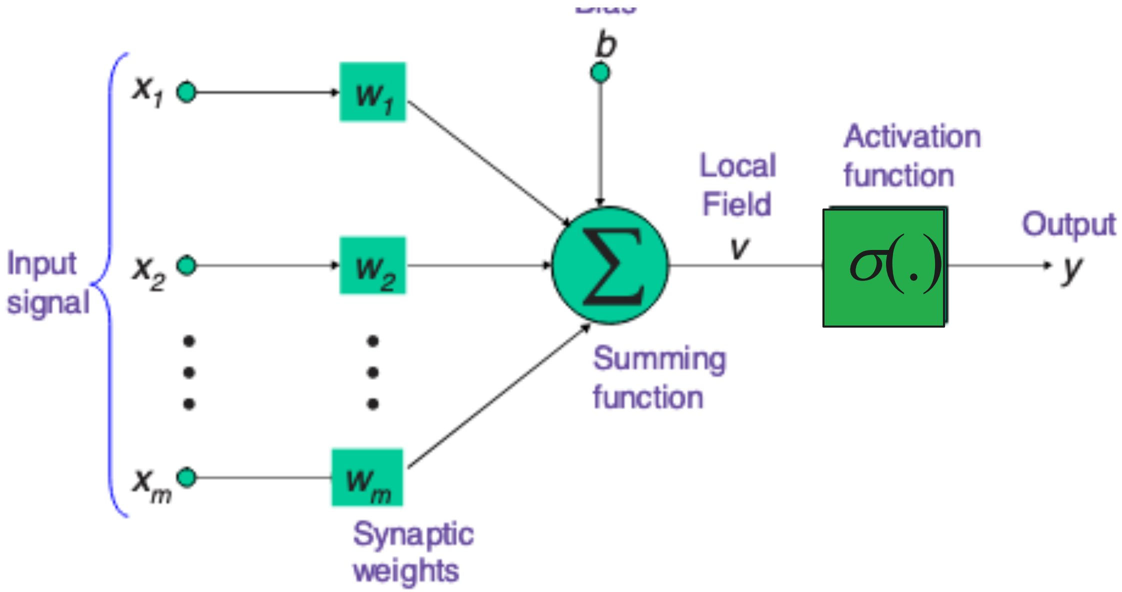
σ is a nonlinear activation function. Most popular function:

$$ReLU(x) = \max\{x, 0\}$$

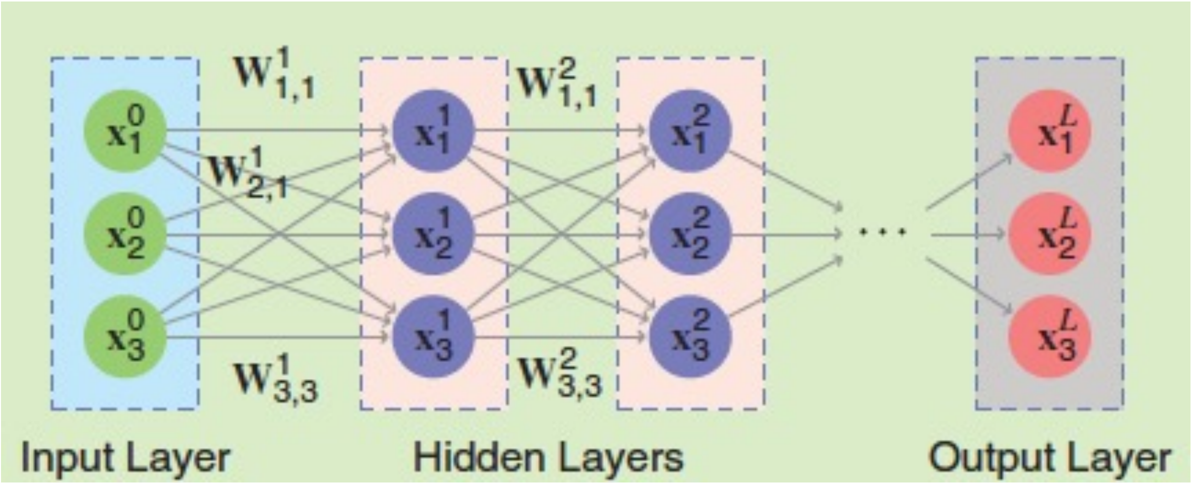
Neural Networks review: the Multilayer Perceptron

Biais and ReLU are not in the general schema,
Computation of one neuron y is:

$$y = \sigma\left(\sum_{j} W_i x_j + b\right)$$

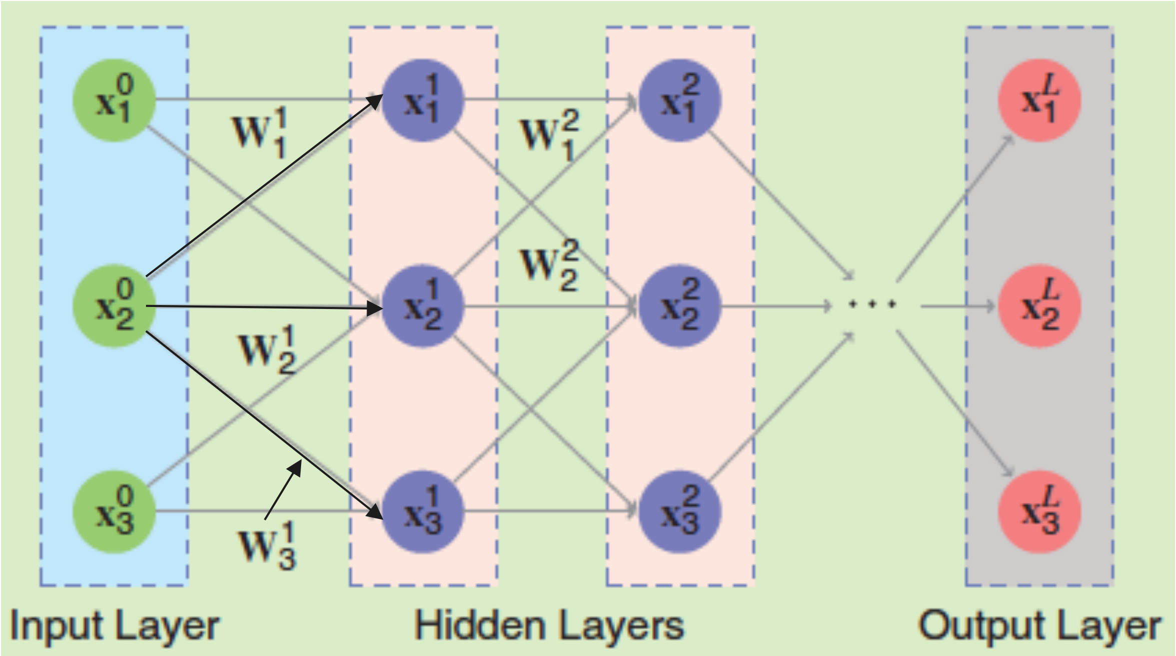


Neural Networks review: the Multilayer Perceptron



- Due to fully connected layer architecture which implies a very big number of parameters making the training difficult, Mutlilayers perceptron are not used in image processing .
- CNN: convolutional neural networks: neuron connections are restricted to local neighbors and weights are shared across different spatial local.

Neural Networks review: Convolutional neural networks

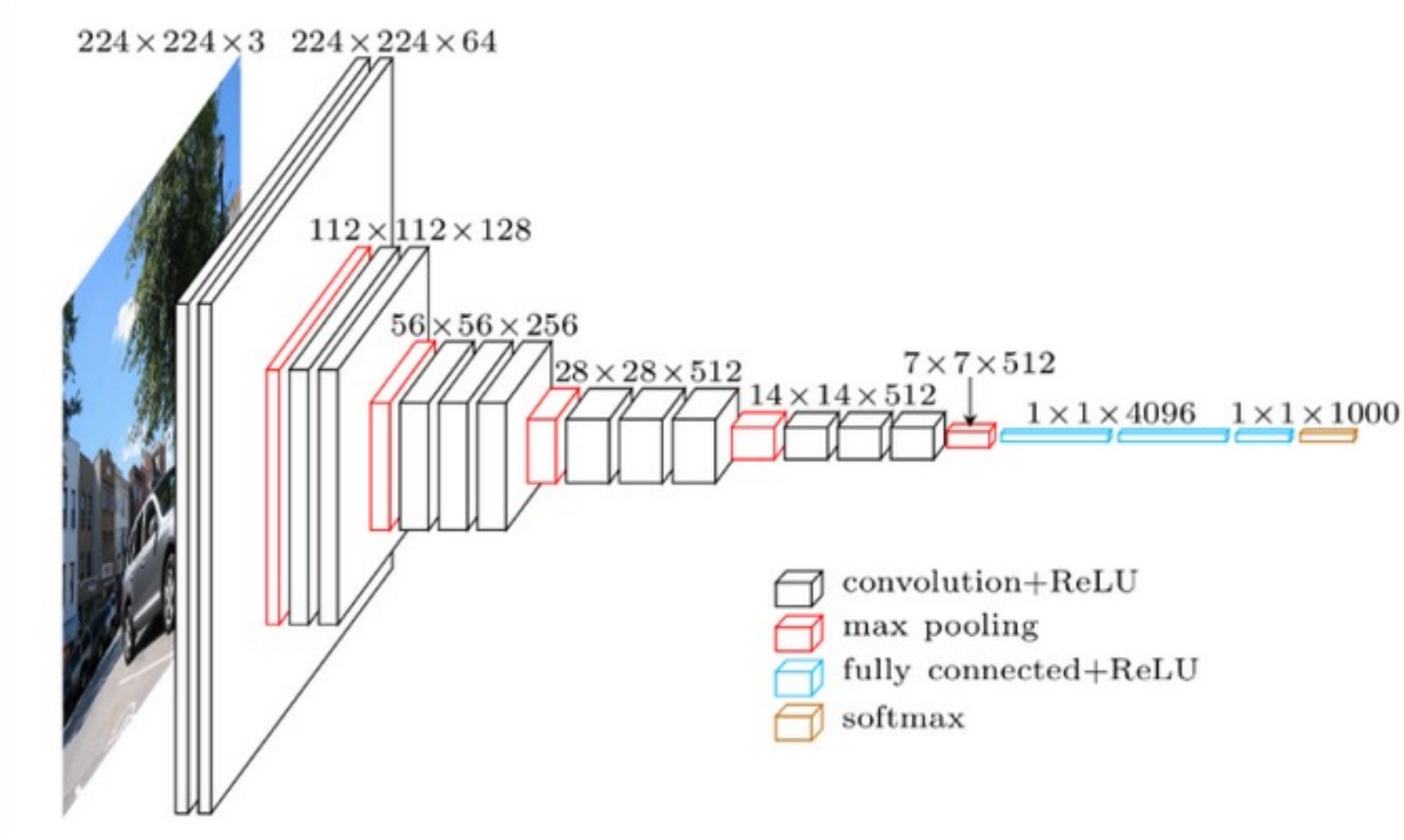


- Convolutional neural network (CNN): sparsely connected neurons, same weights in a layer for every neurons.
- Number of parameters reduced, number of layers can increase: deep neural network.

Neural Networks review: Convolutional neural networks

- Firstly applied to digit recognition, the translation invariance of CNN is a desirable property in many computer vision tasks.
- CNN outperform traditional approaches in many applications, for example in classification.
- Different architectures, most well-known are
  - LeNet-5, (Yann LeCun 98) widely used for written digits recognition (MNIST).
  - AlexNet, VGG, and ResNet won ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

CNN: typical global architecture



Input layer: initial size of the image 224x224x3=100352 neurons

Convolution layers: convolution + ReLU (to have positive values) extract features

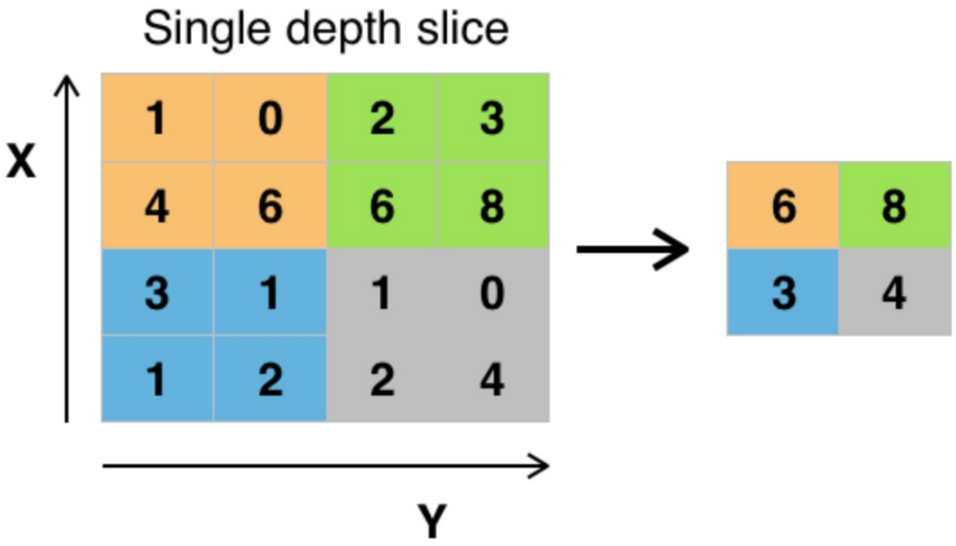
Max pooling: to reduce the dimension by taking the max on a given neighborhood

Fully connected + ReLU to do the classification

Softmax: give the result of the classification (multi-class)

https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

Pooling layer



1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4

Nonlinear down-sampling: Max of the numbers in subregions
The idea is that the exact location of a feature is less important than its rough location relative to other features.

- It reduces the spatial size of the representation: the number of parameters, memory and the amount of computations
- hence to control overfitting.

Very common form of max pooling is a layer with filters of size 2 × 2, applied with a stride of 2.
Pooling is an important component of CNN for object detection.

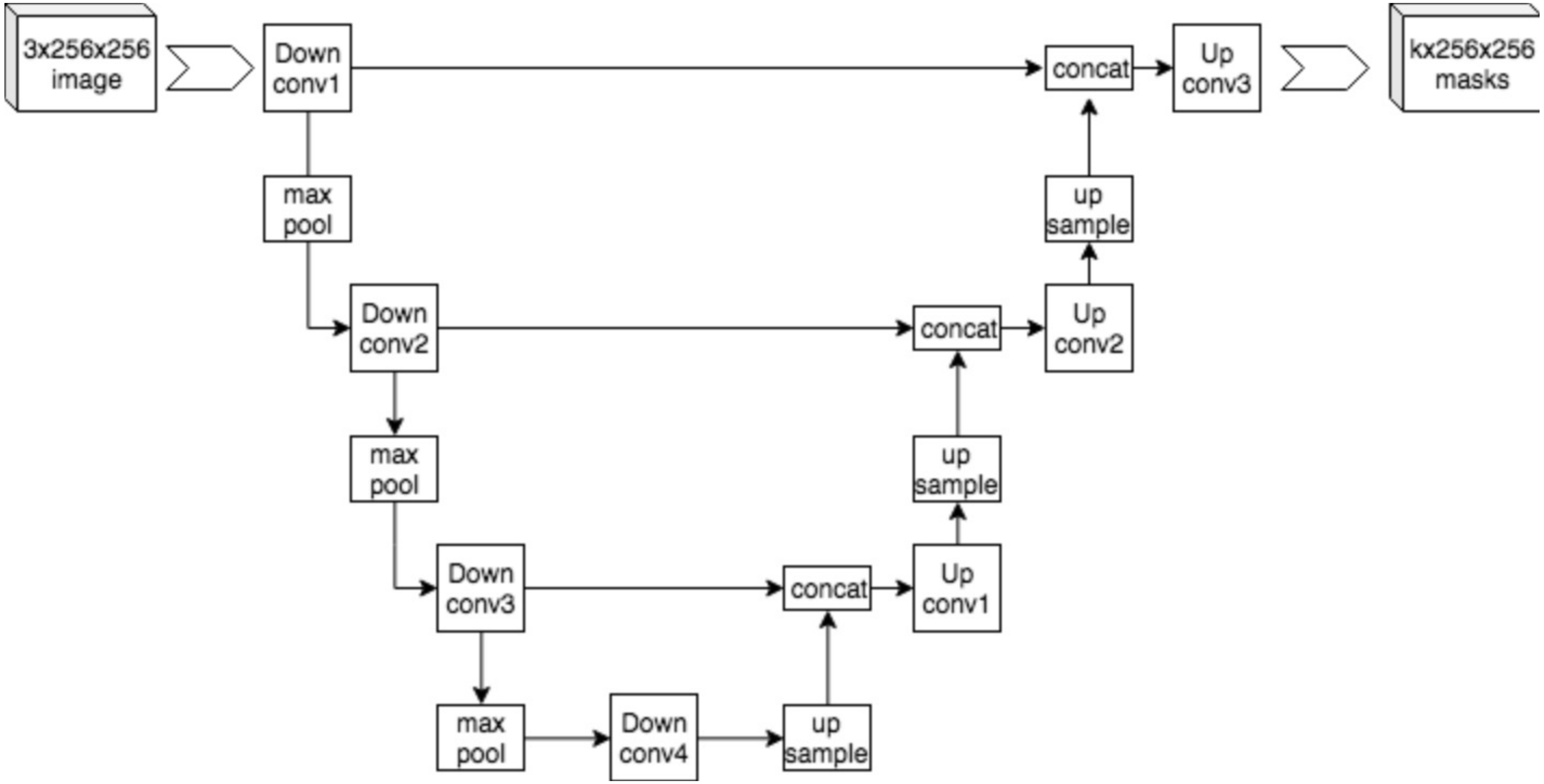
U-Net

- U-Net is a convolutional neural network that was developed for biomedical image segmentation

- The network is optimized to work with fewer training images and to yield more precise segmentations

- Up-sampling steps are added symmetrically to the down-sampling ones to allow the network to propagate context information to higher resolution layers

U-net



- Contracting (convolution, ReLU, maxpooling) and expansive path (up-convolutions and concatenations with high-resolution features from the contraction path)
- During the contraction, the spatial information is reduced while feature information is increased
- The spatial information of the high-resolution features are re-introduced at the expansion step.

NN and inverse problems

- End-to-end NN: training of a CNN from couples of input/output images (degraded,observed/ground truth)

    - Need simulations, data not available from physical acquisition systems as ground truth not always observable
    - Too much training to do for different operators (ex convolution with different filters) and noise (type and level of noise)

We want to:

- Keep the physical knowledge of the acquisition system
- Use NN for image modelling only (a priori model for regularization).

➡ Which architecture to use?

Regularization (reminder)

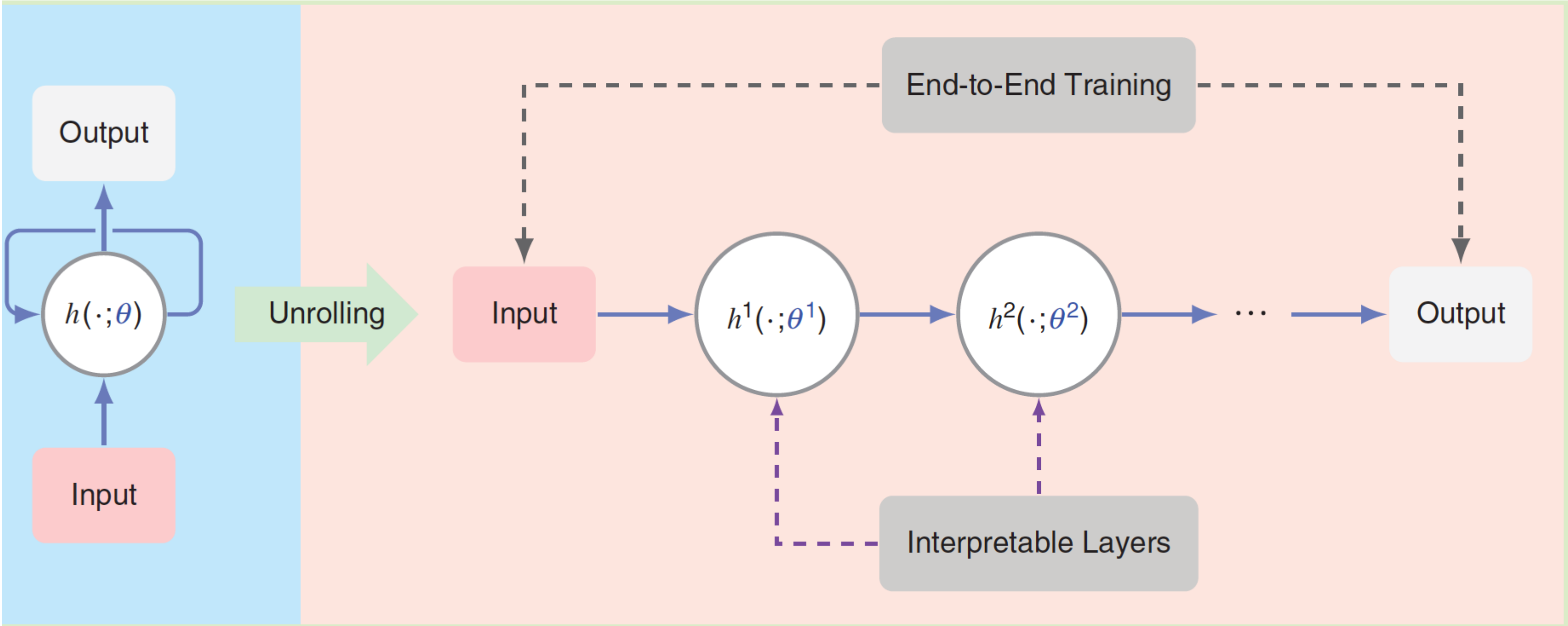
$$\|H\mathrm{u} - g\|_2^2 + J_{reg}(\mathrm{u})$$

- l_2 Regularisation smoothes contours $\|H\mathrm{u} - g\|_2^2 + \lambda\|\nabla\mathrm{u}\|_2^2$
- l_1 sparse regularisation $\|H\mathrm{u} - g\|_2^2 + \lambda\|\mathrm{u}\|_1$
- l_1 regularisation on gradient (Total Variation) $\|H\mathrm{u} - g\|_2^2 + \lambda\|\nabla\mathrm{u}\|_1$
- l_1 Regularisation with wavelet coefficients $\|H\mathrm{u} - g\|_2^2 + \lambda\|W\mathrm{u}\|_1$ (here W = wavelet transform)

- These are model-based regularization.

- What's about data-driven regularization by NN/CNN?
  - Several approaches, we will look at algorithm unrolling

Algorithm unrolling: general idea



Iterative algorithm

corresponding deep network generated by cascading the algorithm's iterations h. One layer = one iteration

Each iteration h depends on algorithm parameters θ, which are transferred into network parameters θ^1, θ^2, .. θ^i. These parameters are learnt from training data sets through end-to-end training. It results in

- Better performance
- Interpretability of the network

Reminder: ISTA algorithm

$$\frac{1}{2}\|H\mathrm{u}-g\|_2^2+\lambda\|\mathrm{u}\|_1$$

- We want to minimize this criterion for $\mathbf{u}\in\mathbb{R}^n$
- We can use the Forward-Backward Splitting algorithm ISTA:

$$\mathrm{u}^{k+1}=\mathrm{prox}_{\frac{\lambda}{\mu}\|.\|_1}\left(\mathrm{u}^k-\frac{1}{\mu}\nabla_{\mathrm{u}}\left(\frac{1}{2}\|H\mathrm{u}^k-g\|_2^2\right)\right)$$

where $\nabla_{\mathrm{u}}\left(\frac{1}{2}\|H\mathrm{u}^k-g\|_2^2\right)=H^T(H\mathrm{u}-d)$ and $\mathrm{prox}_{\lambda\|.\|_1}(\mathrm{x})=S_\lambda(\mathrm{x})$

which is the soft-threshold function applied on each component x_i :

$$S_\lambda(\mathrm{x}_i)=\begin{cases}\mathrm{x}_i-\lambda & \text{if } \mathrm{x}_i>\lambda\\ \mathrm{x}_i+\lambda & \text{if } \mathrm{x}_i<-\lambda\\ 0 & \text{if } |\mathrm{x}_i|\leq\lambda\end{cases}$$

μ controls the iteration step size, $\mu >$ largest eigenvalue of H^TH (Lipschitz constant of the gradient of the least-square term).

Reminder: ISTA alsorithm

- The Forward-Backward Splitting algorithm ISTA can be rewritten as:

$$\mathrm{u}^{k+1} = S_\lambda \left\{ \left(I - \frac{1}{\mu} H^T H \right) \mathrm{u}^k + \frac{1}{\mu} H^T g \right\}$$

- One iteration can be recast into a single network layer, as it includes matrix-vector multiplication, summation and soft-thresholding which is of the same nature as neural network.
- Parameters of the unrolled networks are

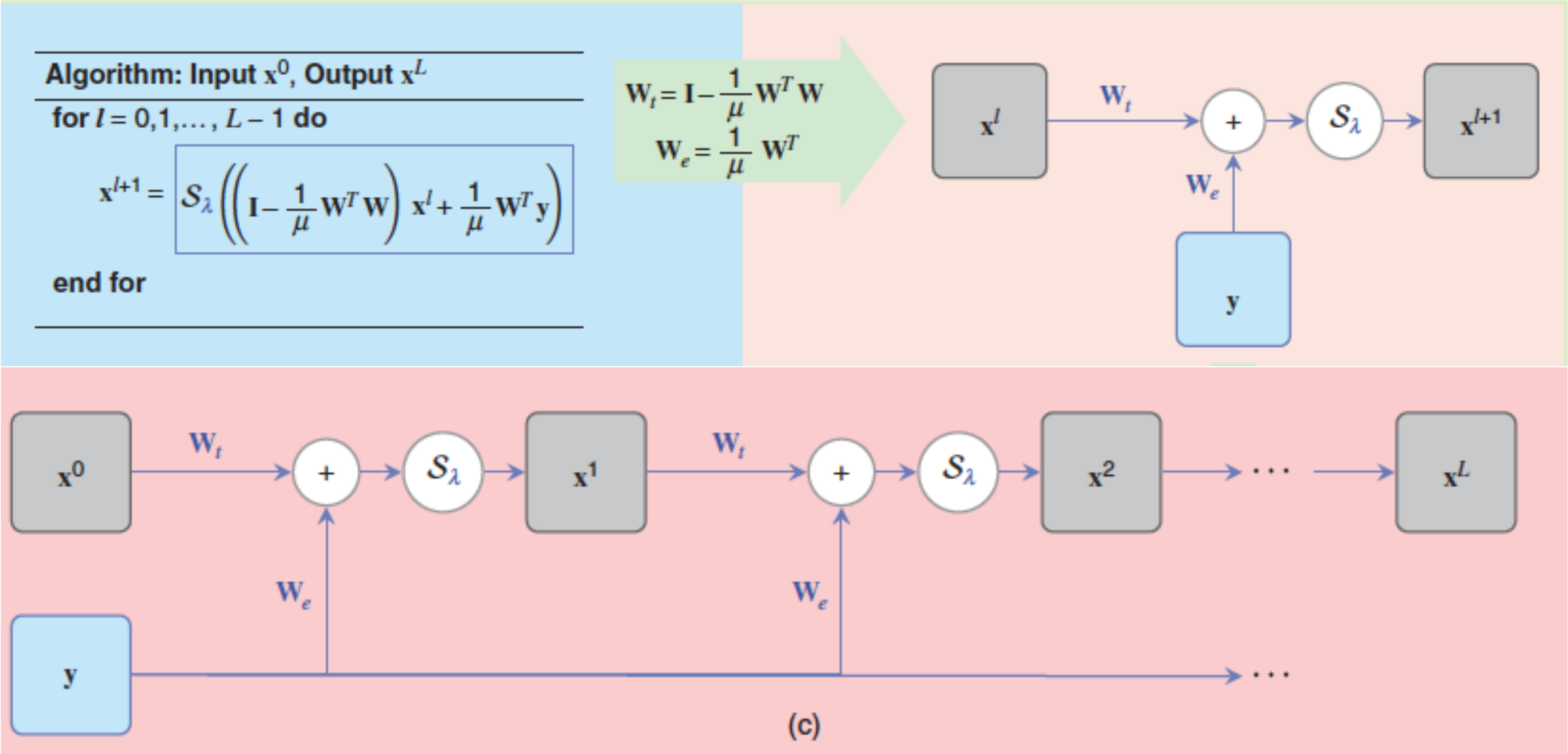
$$W_t = I - \frac{1}{\mu} H^T H$$

$$W_e = \frac{1}{\mu} H^T$$

and the activation function is the soft-threshold S_λ.

Learned ISTA: LISTA

- LISTA applied on $\frac{1}{2}\|W\mathbf{x} - y\|_2^2 + \lambda\|\mathbf{x}\|_1$ with $\begin{cases} W & \leftarrow H \\ \mathbf{x} & \leftarrow \mathbf{u} \\ y & \leftarrow g \end{cases}$



```
Algorithm: Input $\mathbf{x}^0$, Output $\mathbf{x}^L$
for $l = 0,1,\dots, L-1$ do
    $\mathbf{x}^{l+1} = \boxed{\mathcal{S}_\lambda\left(\left(\mathbf{I} - \frac{1}{\mu}\mathbf{W}^T\mathbf{W}\right)\mathbf{x}^l + \frac{1}{\mu}\mathbf{W}^T\mathbf{y}\right)}$
end for
```

$$\mathbf{W}_t = \mathbf{I} - \frac{1}{\mu}\mathbf{W}^T\mathbf{W}$$

$$\mathbf{W}_e = \frac{1}{\mu}\mathbf{W}^T$$

(c)

LISTA algorithm

- The parameters of the network are W_t, W_e , λ
- They are learned through a sequence of training vectors $y^1, y^2, \ldots y^N$ in R^n and their corresponding ground-truth sparse vectors $x^{*1}, x^{*2}, \ldots, x^{*N}$.
- By feeding each y^l, $l=1,..N$, into the network, we retrieve its output $\hat{\mathbf{x}}^l(\mathbf{y}^l; W_t, W_e, \lambda)$

  as predicted sparse vector corresponding to y^l.
- The network training loss-function is formed as

$$\ell(W_t, W_e, \lambda) = \frac{1}{N} \sum_{l=1}^{N} \|\hat{\mathbf{x}}^l(\mathbf{y}^l; W_t, W_e, \lambda) - \mathbf{x}^{*l}\|_2^2$$

LISTA algorithm

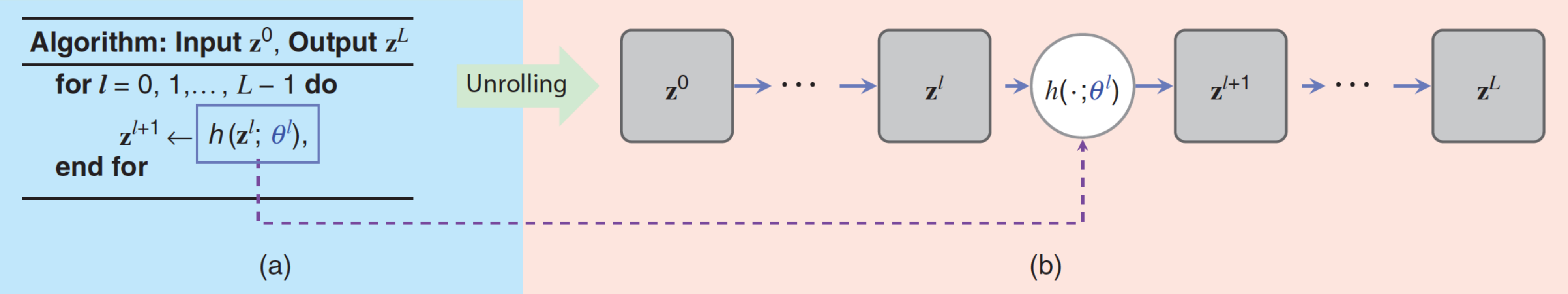
$$\ell(W_t, W_e, \lambda) = \frac{1}{N} \sum_{l=1}^{N} \|\hat{\mathrm{x}}^l(\mathrm{y}^l; W_t, W_e, \lambda) - \mathrm{x}^{*l}\|_2^2$$

- The loss function is minimized using popular gradient-based learning techniques, such as stochastic gradient-descent, to learn W_t, W_e, λ

- It has been empirically shown that the number of layers L in the LISTA can be an order of magnitude smaller than the number of iterations required for the ISTA.

algorithm unrolling

- Generate interpretable networks from the iteration of optimization algorithms



```
Algorithm: Input $\mathbf{z}^0$, Output $\mathbf{z}^L$
for $l = 0, 1, \ldots, L-1$ do
    $\mathbf{z}^{l+1} \leftarrow \boxed{h(\mathbf{z}^l; \theta^l)}$,
end for
```

(a) An iterative algorithm. (b) An unrolled deep network

One iteration is described by function h parametrized by θ^l.
One iteration is mapped into a single network layer, and a finite number of layers are stacked together to form a deep network.

Feeding the data forward through this L-layer network is equivalent to executing the iteration L times (finite truncation).

The parameters θ^l , $l=1,\ldots L$, are learned from real data sets by training the network end to end to optimize the performance. The parameters can either be shared across different layers or vary from layer to layer.