ANN + Introduction to Deep Learning

Rita P. Ribeiro

Machine Learning - 2021/2022

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO
DEPARTAMENTO DE CIÊNCIA DE COMPUTADORES
FACULDADE DE CIÊNCIAS DA UNIVERSIDADE DO PORTO

Summary

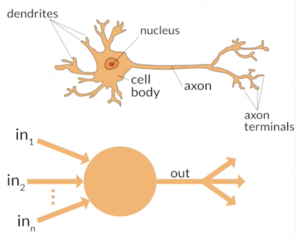
- Artificial Neural Networks
- (Very Short) Introduction to Deep Learning

Artificial Neural Networks

Artificial Neural Networks (ANN)

- Models with a strong biological inspiration. The brain is a highly complex structure, non linear and highly parallel.
- McCulloch e Pitts (1943) proposed the first artificial model of a neuron.

- Neuron: many-inputs / one-output unit
- Synapses: electrochemical contact between neurons

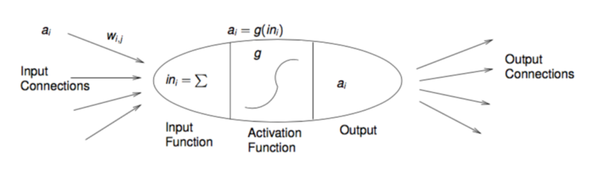


- Output of a neuron: excited or not excited
- Incoming signals from other neurons determine if the neuron shall excite ("fire")
- Output subject to attenuation in the synapses

Artificial Neural Networks (ANN) (cont.)

- An artificial neural network is composed by a set of units (neurons) that are connected. These connections have an associated weight.
- Each unit has an activation level as well as means to update this level.
- Some units are connected to the outside world. We have input and output neurons.
- Learning within ANNs consists of updating the weights of the network connections.

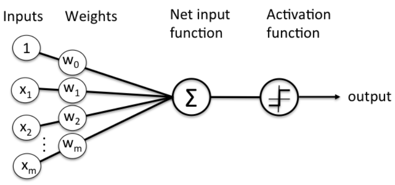
Artificial Neural Networks: Artificial Neuron



- Each unit has a very simple function:
  - receive the input impulses and calculate its ouput as a function of these impulses.
- This calculation is divided in two parts:
  - a linear combination of the inputs: $\in_i = \sum_j w_{ji} a_j + b$
  - a (typically) non-linear activation function: $a_i = g(in_i)$

- Rosenblatt (1958) introduced the notion of perceptron networks. This work was then further extended by Minsky and Papert (1969).
- Perceptrons are networks with an input layer and an output layer.

Simplest Perceptron



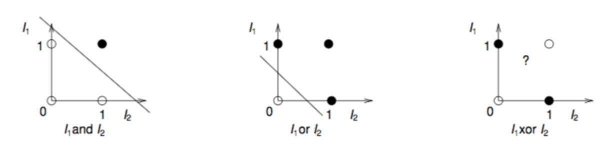
Schematic of Rosenblatt's perceptron.

- A linear classifier for binary classification problems

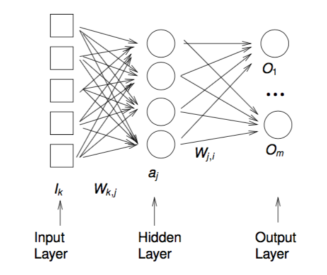
$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + w_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

- It learns by updating the weights through delta rule with learning rate η
- $w_i(t+1) = w_i(t) + \eta(true - predicted)x_i$

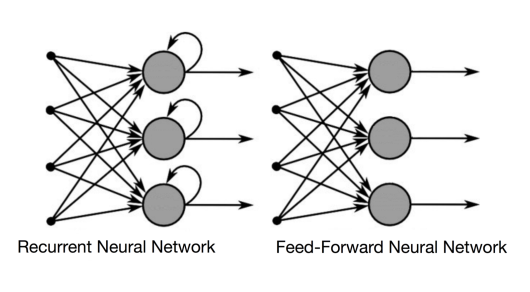
Perceptrons are limited to linearly separable functions.



- Feed-forward networks (Multilayer perceptrons)
  - networks with uni-directional connections (from input to output), and without cycles
  - each unit is connected only to units in the following layer
  - there are not connections from units on a certain layer and units on previous layers

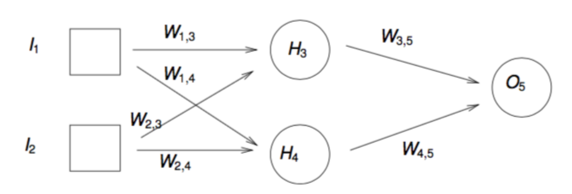


- Recurrent networks
  - networks with arbitrary connections
  - due to the possible feedback effects, recurrent networks are potentially more instable, possibly exhibiting caotic behaviors
  - usually they take longer to converge



Artificial Neural Networks: Types of ANNs (cont.)

- Example of a feed-forward network with one input layer (I), one hidden layer (H) and one output layer (O) with one output variable.



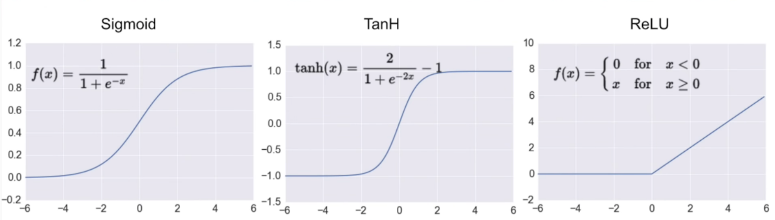
- The output can be represented as follows:

$$
\begin{aligned}
a_5 &= g(W_{3,5}a_3 + W_{4,5}a_4) = \\
&= g(W_{3,5}g(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}g(W_{1,4}a_1 + W_{2,4}a_2))
\end{aligned}
$$

- where $g()$ is the activation function

Artificial Neural Networks: Activation Functions

- Activation functions are used to determine the output of each node of the neural network
  - linear
  - non-linear: most commonly used as it allows the model to generalize or adapt with variety of data
- Examples



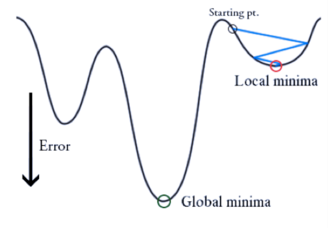
- This is the most popular algorithm for learning ANNs.
- It has similarities with the learning algorithm used in perceptron networks
- Intuition:
  - each unit is responsible for a certain fraction of the error in the output nodes to which it is connected
  - thus, the error is divided according to the weight of the connection between the respective hidden and output units, thus propagating the errors backwards
- Backpropagation computes the gradient in weight space of a feedforward neural network, with respect to a loss function.

Artificial Neural Networks: Backpropagation Algorithm (cont.)

The Algorithm (for one hidden layer)

- Initialize network weights (often small random values)
- Do
  - For each example in training set
    - predict the output
    - calculate the prediction error by a loss function
    - compute δ_h for all the weights from hidden layer to output layer
    - compute δ_i for all the weights from input layer to hidden layer
    - update network weights
- Until it converges
  - all examples are classified correctly or stopping criterion is satistified
- Return the network

Gradient Descendent



- Stochastic Gradient Descent: instead of calculating the gradient of the full error function (which involves using the full training set), we update the weights one example at a time.
- Batch Gradient Descent: the batch size is the number of sub samples given to the network after which weights update happens.
- Both are more effective to escape from local minima.

When to stop training?

- If stopping too early: risk of getting a network not yet trained.
- If stopping too late: danger of overfiting (adjustment to noise in the data)
- Stopping criteria:
  - maximum number of iterations
  - error based on the training set
    - when the error in the training set is below a certain limit.
  - error based on a validation set (independent from the training set)
    - when the error on the validation set has reached a minimum.

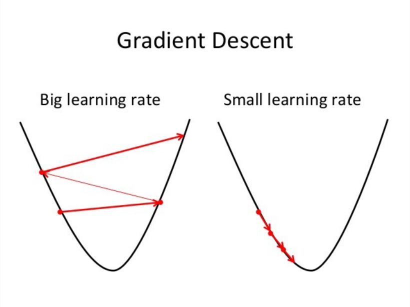
Artificial Neural Networks: Issues

Network topology

- The number of nodes in the hidden layer
  - few nodes: underfitting
  - many nodes: overfitting
  - there are no criteria for defining the number of nodes in the hidden layer
- Effect of learning rate
  - a small learning rate has the effect of learning times higher
  - a high learning rate may lead to non-convergence

Artificial Neural Networks: Issues (cont.)

- The learning rate sets the size of the steps to obtain the direction of maximum descendent.



Artificial Neural Networks: Issues (cont.)

Generalization vs Specialization trade-off

- Optimal number of hidden neurons
  - too many hidden neurons: you get an overfit, training set is memorized, thus making the network useless on new data sets
  - not enough hidden neurons: network is unable to learn problem concept
- Overtraining
  - too much examples, the ANN memorizes the examples instead of the general idea

Artificial Neural Networks: Issues (cont.)

Some relevant hyperparameters

- Network Structure
  - number of layers
  - number of neurons in each layer
  - weights initialization
  - activation function
- Training Algorithm
  - learning rate
  - number of epochs
  - early stopping criterion
  - weight decay (a regularization on the network weights)

Artificial Neural Networks: Issues (cont.)

Some Tips

- Features with very different distributions of values are not convenient, given the typical activation functions.
  - Data should be standarized.
- Missing values in input features may be represented as zeros, which do not influence the neural net training process.
- Output in Multiclass Setting
  - Use one-hot encoding, there are M output neurons (1 per class),
  - For each case, the class with the highest probability value.

Artificial Neural Networks: Issues (cont.)

Some Tips (cont.)

- Initialize the weights with small random values $[-0.05, 0.05]$
- Shuffle the training set between epochs, i.e. change the sequence of the examples
- The learning rate must start with a high value that decreases progressively
- Train the network several times using different initialization of the weights

Artificial Neural Networks: Wrap-Up

Use ANNs when

- Input is high-dimensional discrete or real-valued (e.g. raw sensor input)
- Output is discrete or real valued
  - Classification: use Softmax function as activation function in output layer to compute the probabilites for the classes
  - Regression: use a linear function as activation function in output layer
- Output is a vector of values
- Possibly noisy data
- Form of target function is unknown
- Human readability of result is unimportant

Artificial Neural Networks: Wrap-Up (cont.)

Pros

- Tolerance of noisy data
- Ability to classify patterns on which they have not been trained
- Successful on a wide range of real-world problems
- Algorithms are inherently parallel

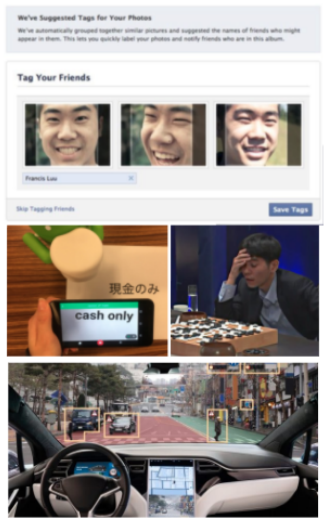
Cons

- Long training times
- Resulting models are essentially black boxes

(Very Short) Introduction to Deep Learning

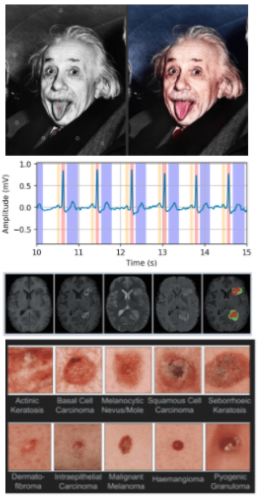
Deep Learning: where?

- Image recognition (e.g. Google, Facebook)
- Automatic text translation (e.g. Google Translator)
- Answers in natural language / digital assistants
- Games (e.g. DeepMind AlphaGo)
- Transcript of handwritten text
- Self-driving cars



Deep Learning: where?

- Image colorization, caption generation
- Classification of protein and DNA sequences
- Heart sound: classification and segmentation
- Tumor images detection from MRI, CT, X-rays
- Skin lesion classification from clinical and dermoscopic images
- Parkinson's disease detection from voice recording



A (Very Short) Introduction to Deep Learning (cont.)

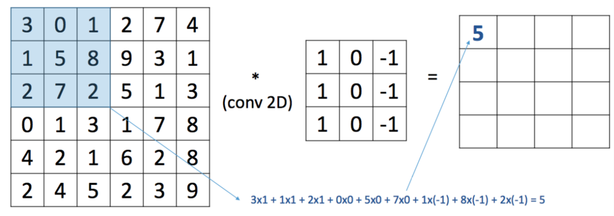
- Deep learning = Deep neural networks
  - Deep = high number of hidden layers
  - Learn a larger number of parameters!
- It was made possible recently (~ in the last 6 years) since we have:
  - Access to big amounts of (training) data
  - Increased computational capabilities (e.g., GPUs)

Convolution Neural Networks (CNN)

- Feedforward neural networks
- Neurons typically use the ReLU or sigmoid activation functions
- Weight multiplications are replaced by convolutions (filters)
- Change of paradigm: can be directly applied to the raw signal, without computing first ad hoc features
- Features are learnt automatically!!

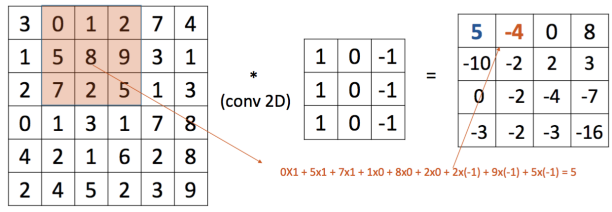
Convolution

- mathematical operation between two matrices;
- the 2nd matrix is a filter that is overlapped to each position of the 1st matrix.

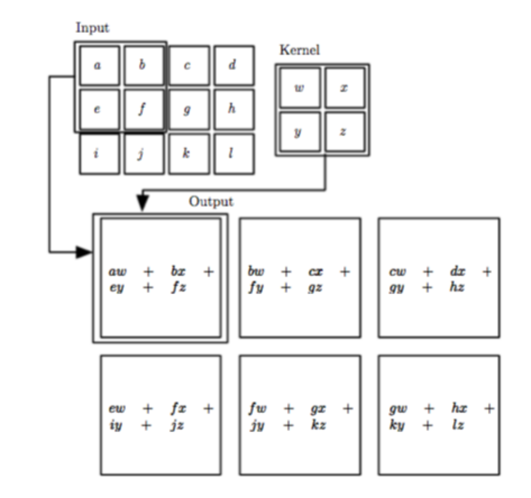


Convolution

- mathematical operation between two matrices;
- the 2nd matrix is a filter that is overlapped to each position of the 1st matrix.

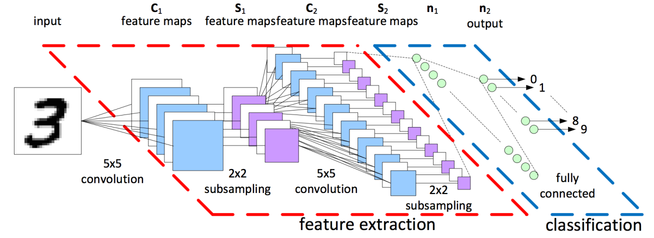


Convolution



Convolutional neural networks (CNNs) (cont.)

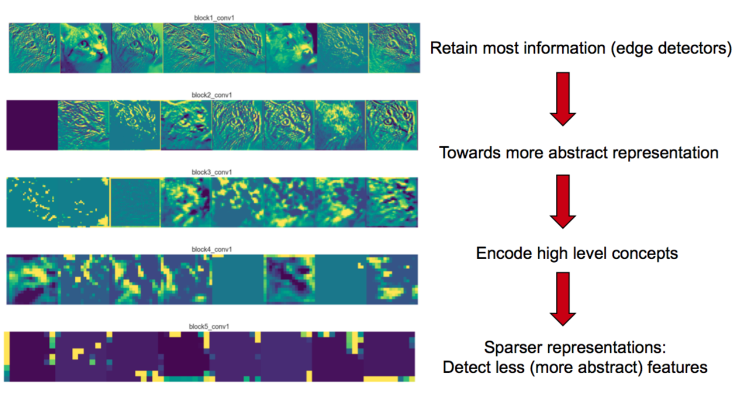
Example for Image Processing



- convolutional layers, followed by nonlinear activation and subsampling (pooling)
- output of hidden layers (feature maps) are features learnt by the CNN
- flatten fully connected layers for classification (as in “standard” NN)

Convolutional neural networks (CNNs) (cont.)

Example for Image Processing: feature extraction

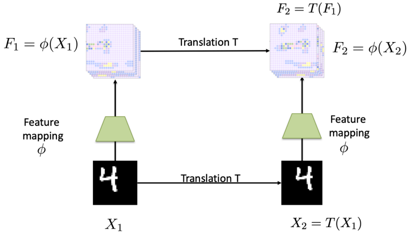


- the convolutions, applied to various zones of the image, act as filters that can detect certain patterns

Convolutional neural networks (CNNs) (cont.)

Properties

- Reduced amount of parameters to learn (local features)
- More efficient than dense multiplication
- Specifically thought for images or data with grid-like topology
- Convolutional layers are equivariant to translation



- if image input is translated by a certain amount,
- the feature map is also translated
- useful for classification

- Currently state-of-the-art in several tasks

Great results! But...

- Like any other technique, DL does not solve all problems and will not always be the best option for any learning task.
- Difficult to select best architecture for a problem
- Require new training for each task/configuration
- (Most commonly) require a large training dataset to generalize well
    - Data augmentation, weight regularization, dropout, transfer learning, etc.
- Still not fully understood why it works so well
    - Unstable against adversarial examples

To know more

- Book – I.Goodfellow,Y.Bengio,and A.Courville. Deep learning. Vol.1. Cambridge: MIT press, 2016.
- Tutorial – Oxford Visual Geometry Group: VGG Convolutional Neural Networks Practical

References

References

Aggarwal, Charu C. 2015. *Data Mining, the Texbook*. Springer.

Gama, João, André Carlos Ponce de Leon Ferreira de Carvalho, Katti Faceli, Ana Carolina Lorena, and Márcia Oliveira. 2015. *Extração de Conhecimento de Dados: Data Mining -3rd Edition*. Edições Sílabo.

Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Renna, Francesco. 2019. "Introduction to Deep Learning." Slides.

Rocha, Miguel. 2019. "Foundations and Applications of Machine Learning Course." Slides.

Smola, Alex J., and Bernhard Schölkopf. 2004. "A Tutorial on Support Vector Regression." *Statistics and Computing* 14 (3): 199–222.

Tan, Pang-Ning, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining*. 2nd ed. Pearson.

Torgo, Luís. 2017. "Data Mining I Course." Slides.