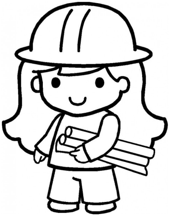


Architecture design

Vincent Barra
LIMOS, UMR 6158 CNRS, Université Clermont Auvergne

Architecture

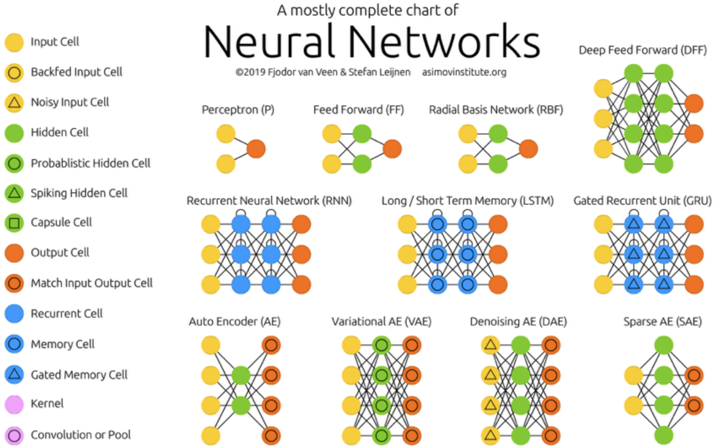
Architecture

- Number of layers (depth)
- Number of neurons per layer
- Type of neurons
- Type of connections between neurons / layers

Classical Network architectures

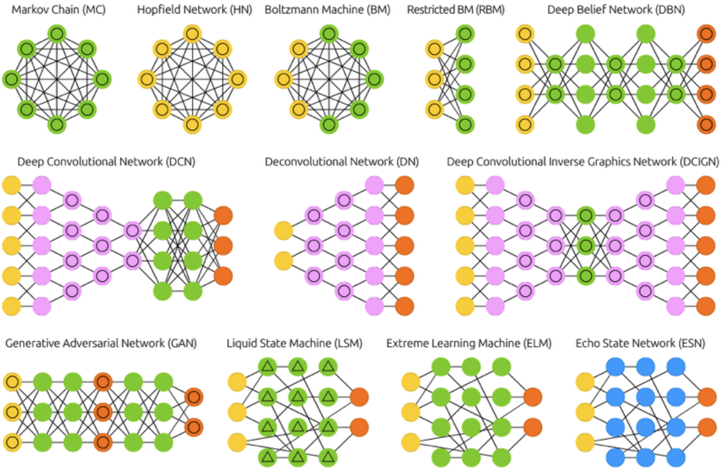
- Most networks are organized into groups of layers, arranged in a chain structure
- Each layer is a function of the previous one

A (MOSTLY) COMPLETE ZOO[1]



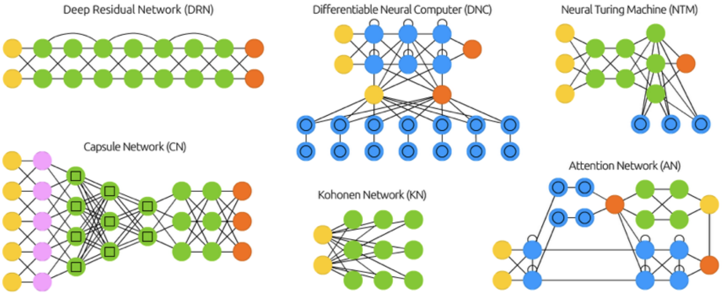
[1] source: The Asimov Institute

A (mostly) complete Zoo



LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

A (MOSTLY) COMPLETE ZOO



Universal approximation

Theorem (Cybenko 1989; Hornik et al, 1991)

σ: bounded, non-constant continuous function, - I_d: d-dimensional hypercube - $C(I_d)$ space of continuous functions on I_d.
$(\forall f \in C(I_d))(\forall \epsilon > 0)(\exists q > 0, v_i, \mathbf{w_i}, b_i, i \in [\![1 \ldots q]\!])$ such that

$$F(\mathbf{x}) = \sum_{i=1}^{q} v_i \sigma(\mathbf{w}^{\mathbf{T}}\mathbf{x} + b)$$

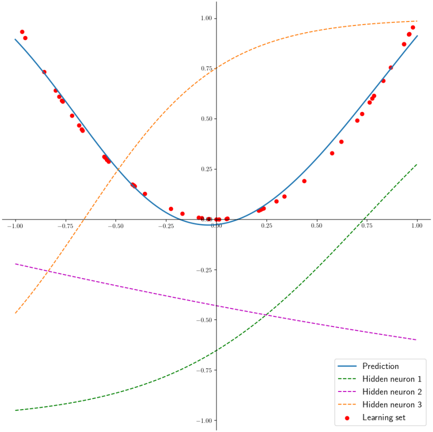
satisfies $\sup_{\mathbf{x} \in I_d} \mid f(\mathbf{x}) - F(\mathbf{x}) \mid < \epsilon$

And so..

A feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of $\mathbb{R}^d$, under mild assumptions on the activation function

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

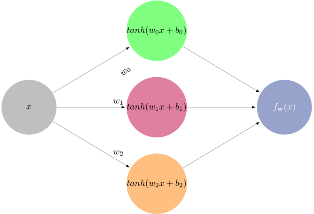
Universal approximation



$f(x) = x^2, |Z| = 50$

A simple example

- $|Z|$ points uniformly sampled (red) over the definition set
- 1 hidden layer MLP, 3 neurons.
- $tanh$ activation function, and linear output neurons
- network output : blue curve
- hidden neurons outputs: dashed curves



LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

cnrs

UNIVERSITÉ Clermont Auvergne

Universal approximation

Good news but...

- Does not inform about good/bad architectures, the number of neurons q nor how they relate to the optimization procedure
- Bounds on size of the single-layer network exist for a broad class of functions....
- But worst case is exponential / q

Bad news :-(: No Free Lunch theorem

There is no universal procedure for examining a training set of samples and choosing a function that will generalize to points not in training set

Effect of depth

Up to now

- A feedforward network with a single layer is sufficient to represent "any" function
- But the layer may be infeasibly large and may fail to generalize well
- Using deeper models can reduce number of units required and reduce generalization error

Theorem (Montúfar et al, 2014)

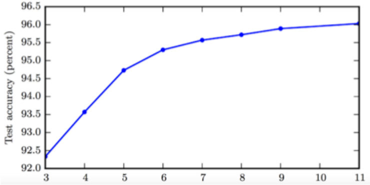
A MLP with ReLU as activation functions, p inputs, L hidden layers with $q \geq p$ neurons can compute functions having $\Omega\left(\left(\frac{q}{p}\right)^{(L-1)p} q^p\right)$ linear regions.

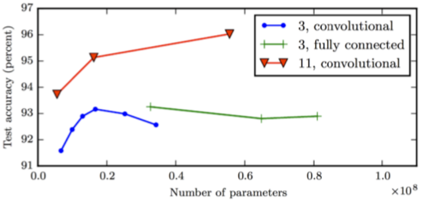
Properties

- The number of linear regions of deep models grows exponentially in L and polynomially in q.
- Even for small values of L and q, deep rectifier models are able to produce substantially more linear regions than shallow models.

Effect of depth

- Test accuracy consistently increases with depth
- Increasing parameters without increasing depth is not as effective





Deep architectures express a useful prior over the space of functions the model learns

Specialized architectures

CNN, RNN,...Discussed in future lectures

LIMOS LABORATOIRE D'INFORMATIQUE, DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

cnrs

UCA UNIVERSITÉ Clermont Auvergne