ReLu and Maxout Networks and Their Possible Connections to Tropical Methods

Jörg Zimmermann, AG Weber

Institute of Computer Science
University of Bonn, Germany

Overview

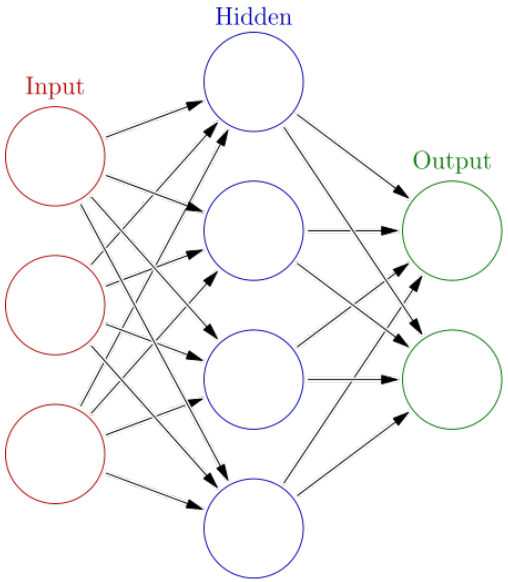
1. Artificial Neural Networks
2. Activation Functions
3. Maxout Networks: are they tropical?

Artificial Neural Networks

- Network of functions
- If the network graph is acyclical (DAG), it is called a Feedforward Neural Networks (FNNs)
- If the network graph contains loops, it is called a Recurrent Neural Network (RNNs).

Feed Forward Networks

- FNNs can be organised in layers: input layer, hidden layer(s), output layer



- Networks having several hidden layers are called deep networks.

Forward Neural Networks

- A node computes:

$$f(\mathbf{x}) = \sigma(\sum_i w_i x_i + b)$$

- σ is a nonlinear function and is called activation function.
- Typically σ was the logistic function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Forward Neural Networks: Basic theorems

- Neural Networks are universal approximator for continuous functions within a bounded domain.
- Even one hidden layer networks are universal approximators.
- But the number of neurons can grow exponentially when one flattens a deep network.

Forward Neural Networks: Training

- Weights (and offsets) are parameters which can be learned (wrt. to a training data set)
- Classical training algorithm: Backpropagation (BP)
- Propagating error signals from the output layer back to the input layer via the hidden layer(s) (gradient descent).

Forward Neural Networks: Problems

- Problem of BP (using logistic activation function): backpropagated error signals grow or vanish exponentially from hidden layer to hidden layer.
- This was one of the main road blocks preventing the training of deep networks, and one of the main reasons interest in FNNs dropped in the mid 90s.

Forward Neural Networks: approaches for training deep networks

Approaches to circumvent the vanishing gradient problem:

- Introduce amplifier neurons $\Rightarrow$ Recurrent Networks, LSTM-Networks (Jürgen Schmidhuber, Lugano, Switzerland)
- transform supervised learning of all layers simultaneously into a sequence of unsupervised learning task, hidden layer by hidden layer. (Geoffrey Hinton, Toronto, Canada)
- Introduction of new activation functions.

Activation Functions

The surge in interest in deep learning has led to the investigation of many different activation functions.

ReLU (Rectified Linear Unit):

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

smooth ReLU:

$$sReLU(x) = \log(1 + e^x)$$

Activation Functions

ELU (Exponential Linear Unit):

$$ELU(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & otherwise \end{cases}$$

Activation Functions

SELU (Scaled Exponential Linear Unit):

$$SELU(x) = \lambda \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & otherwise \end{cases}$$

This activation function is introduced in the article "Self-normalizing Neural Networks" (2017, Sepp Hochreiter). One of the few examples where they prove properties of their activation function. They can show that on average their is no vanishing (or exploding) gradient problem (for magic numbers for α and λ).

Activation Functions

Maxout:

$$maxout(z_1, ..z_k) = max(\sum_i w_{1i}x_i + b_1, ..., \sum_i w_{ki}x_i + b_k)$$

$\mathbf{x} = (x_1, .., x_n) =$ output vector of the previous layer

The maxout-node applies k different scalar products to $\mathbf{x}$ plus k offsets $(b_1, .., b_k)$ and finally takes the maximum of these k values.

$z_j = \sum_i w_{ji}x_i + b_j$

Maxout Networks

In 2013 an article titled "Maxout Networks" (Ian Goodfellow) was published introducing a max-based activation function.

- Maxout networks are neural networks using the maxout-function as activation function.
- ReLu is a special case.
- If k scalar products are provided for a node, effectively this node can learn a local nonlinear activation function by approximating it with a piecewise linear function consisting of k intervals.

Maxout Networks

- Maxout Networks are universal approximators, too.
- Analogically, they can be flattened to one max-layer, but again by blowing up the network exponentially.
- A maxout network tessellates the input space into polytopes, and computes a linear function on each polytope.
- The authors claim that maxout networks are able to generalize from smaller data samples, but only empirical evidence, no proofs.

Maxout Networks: are they tropical?

If one accepts non-integer coefficients, then maxout-Networks are tropical!

So, tropical geometry may be useful to prove properties of maxout networks.