Neural network notes

Marco Marini

April 25, 2016

Contents

Abstract

This document contains notes about neural network regarding the Temporal Differential reinforcement alghorithms.

1 General

Let us have a neural network composed by N layers. Each layer $i = 1 \dots N$ is composed by s_i neurons.

The first layer

$$H_{1i} = x_i$$

rappresents the input values of the network. We identify the input bias signals with

$$H_{i0} = 1$$

Each neuron processes the input signals from its layer and produces the output signal to the next layer.

Let us exam a NLR (Non Linear regression) network.

The transfer function of each hidden neuron $i < N$ is

$$\begin{array}{l} H_{(i+1)j} = \frac{1}{1+e^{-Z_{ij}}} \\ Z_{ij} = \sum_{k=0}^{s_i} w_{ijk} H_{ik} \end{array} \tag{1}$$

the transfer function of each output neuron N is

$$H_{Nj} = \sum_{k=0}^{s_{N-1}} w_{(N-1)jk} H_{(N-1)k} \tag{2}$$

2 Cost Function

Let y_j be the expected output for the neuron j, the error of neuron is

$$\delta_{(N-1)j} = y_j - H_{Nj} \tag{3}$$

The cost function of the network is

$$J = \left| \begin{array}{l} \frac{1-\alpha}{2} \sum_{i=1}^{s_N} \delta_{(N-1)i}^2 + \frac{\alpha}{2} \sum_{i,j,k} w_{ijk}^2,\ k \geq 1 \\ \frac{1-\alpha}{2} \sum_{i=1}^{s_N} \delta_{(N-1)i}^2,\ k = 0 \end{array} \right.$$

where α is the regularization factor.

Let us extract ∇J

$$\frac{\partial}{\partial w_{ijk}} J = \left| \begin{array}{l} (1-\alpha) \sum_{r=1}^{s_N} \delta_{(N-1)r} \frac{\partial}{\partial w_{ijk}} \delta_{(N-1)r} + \alpha w_{ijk},\ k \geq 1 \\ (1-\alpha) \sum_{r=1}^{s_N} \delta_{(N-1)r} \frac{\partial}{\partial w_{ijk}} \delta_{(N-1)r},\ k = 0 \end{array} \right.$$

By (3) we have

$$\frac{\partial}{\partial w_{ijk}} \delta_{(N-1)r} = -\frac{\partial}{\partial w_{ijk}} H_{Nr}$$

therefore

$$\frac{\partial}{\partial w_{ijk}} J = \left| \begin{array}{l} -(1-\alpha) \sum_{r=1}^{s_N} \delta_{(N-1)r} \frac{\partial}{\partial w_{ijk}} H_{Nr} + \alpha w_{ijk},\ k \geq 1 \\ -(1-\alpha) \sum_{r=1}^{s_N} \delta_{(N-1)r} \frac{\partial}{\partial w_{ijk}} H_{Nr},\ k = 0 \end{array} \right.$$

By now we disregard the regularization effects.

By (2) we have

$$\frac{\partial}{\partial w_{(N-1)jk}} H_{Nj} = H_{(N-1)k}$$

$$\frac{\partial}{\partial w_{(N-1)jk}} H_{Nr} = 0,\ r \neq j$$

therefore

$$\frac{\partial}{\partial w_{(N-1)jk}} J = -\delta_{(N-1)j} H_{(N-1)k}$$

By (2) and $i \leq N-2$ we have

$$\frac{\partial J}{\partial w_{ijk}} = -\sum_r \delta_{(N-1)r} \frac{\partial}{\partial w_{ijk}} \left[\sum_s w_{(N-1)rs} H_{(N-1)s} \right] =$$

$$= -\sum_{r,s} \delta_{(N-1)r} w_{(N-1)rs} \frac{\partial}{\partial w_{ijk}} H_{(N-1)s}$$

By (1) we have

$$\frac{\partial}{\partial w_{ijk}} H_{(i+1)j} = \frac{\partial}{\partial Z_{ij}} H_{(i+1)j} \frac{\partial}{\partial w_{ijk}} Z_{ij} = H_{(i+1)j}(1 - H_{(i+1)j}) Hik$$

therefore

$$\frac{\partial}{\partial w_{(N-2)jk}} J = -\sum_r \delta_{(N-1)r} w_{(N-1)rj} H_{(N-1)j}(1 - H_{(N-1)j}) H_{(N-2)k}$$

3 Back-propagation

Let us define

$$\delta_{(N-2)j} = \sum_r \delta_{(N-1)r} w_{(N-1)rj} H_{(N-1)j}(1 - H_{(N-1)j})$$

Then we have

$$\frac{\partial}{\partial w_{(N-2)jk}} J = -\delta_{(N-2)j} H_{(N-2)k}$$

We can induce just for $i \leq N-2$

$$\delta_{ij} = \sum_k \delta_{(i+1)k} w_{(i+1)kj} H_{(i+1)j}(1 - H_{(i+1)j})$$

while in general

$$\frac{\partial}{\partial w_{ijk}} J = -\delta_{ij} H_{ik}$$

Let us now define the back-propagation tensor as

$$B_{ijk} = w_{(i+1)kj} H_{(i+1)j}(1 - H_{(i+1)j}),\ i \leq N-2$$

The errors on intermediate layers are

$$\delta_{ij} = \sum_{k=1}^{s_i} \delta_{(i+1)k} B_{ijk},\ i = 1 \ldots N-2$$

Now reintroducing the regularization effetcs we have

$$\frac{\partial}{\partial w_{ijk}} J = \left| \begin{array}{l} -(1-\alpha)\delta_{ij} H_{ik} + \alpha\, w_{ijk},\ k \geq 1 \\ -(1-\alpha)\delta_{ij} H_{ik},\ k = 0 \end{array} \right.$$

The weight changes to reduce the error (gradient descent) are

$$\Delta w_{ijk} = -\eta \frac{\partial}{\partial w_{ijk}} J = \left| \begin{array}{l} \eta \left[(1-\alpha)\delta_{ij} H_{ik} - \alpha w_{ijk}\right],\ k \geq 1 \\ \eta(1-\alpha)\delta_{ij} H_{ik},\ k = 0 \end{array} \right.$$

where η is the learning rate.

4 Returns

Let us have a process where the best policy generates a never end episode (infinite length) and let the positive reward be $r_+ > 0$ every n steps, while a wrong choice ad step n generates the end of episode with a negative reward of $r_- < 0$.

The return at step n is then

$$R_+ = \frac{r_+}{1 - \lambda^{2(n-1)}}$$

if the best strategy is applied otherwise

$$R_- = r_-$$

Suppose $R_+ \gg -R_-$ we have

$$\frac{r_+}{1 - \lambda^{2(n-1)}} \gg -r_-$$

$$r_+ \gg -(1 - \lambda^{2(n-1)})r_-$$

For example if $\lambda^{2(n-1)} = \frac{1}{2}, r_- = -1$ we have

$$r_+ \gg \frac{1}{2}$$

such as $r_+ = 5$

5 TDLayer

A layer is a list of neurons that process a set of input signal.

The output signal of this process is

$$h_i = f_i(z_i)$$

$$z_i = \sum_{k=0}^{m} w_{ik} x_k$$

The cost function is

$$J = \frac{1}{2}\left[(1-\alpha)\sum_{i=0}^{n-1}(y_i - h_i)^2 + \alpha \sum_{i=0}^{n-1}\sum_{k=1}^{m-1} w_{ik}^2\right] = \frac{1}{2}\left[(1-\alpha)\sum_{i=0}^{n-1}\delta_i^2 + \alpha \sum_{i=0}^{n-1}\sum_{k=1}^{m-1} w_{ik}^2\right]$$

$f(x)$ is the activation function and may be
$f(x) = x$ for NLR layer,
$f(x) = sigmoid(x) = \frac{1}{1+e^{-x}}$ for logistic layer
$f(x) = \tanh(x)$

The gradient of J is

$$\frac{\partial J}{\partial w_{ij}} = -(1-\alpha)\sum_{k=0}^{n-1}\delta_k \frac{\partial h_k}{\partial z_k}\frac{\partial z_k}{\partial w_{ij}} + \alpha\sum_{i=0}^{n-1}\sum_{k=1}^{m-1} w_{ik} = -(1-\alpha)\delta_i f_i' x_j + \alpha w_{ij}$$

the term αw_{ij} is valid only for $j \neq 0$
$f'(x) = 1$ for $f(x) = x$,
$f'(x) = f(1-f)$ for $f = sigmoid(x)$,
$f'(x) = (1+f)(1-f)$ for $f = \tanh(x)$

The backpropagation errors are

$$\delta_i' = \sum_j \delta_j f_j' w_{ji} = \sum_j w_{ij}^T \delta_j f_j'$$

The updated eligility traces are

$$e_{ij}' = \gamma\lambda e_{ij} - g(\nabla J)$$

$g(x)$ is the backtrace function and may be identity x or $sign(x)$

The updated weights are

$$w_{ik}' = w_{ij} + \eta e_{ij}'$$