Phrasing a valid participation loss term for Backprop

The problem when evaluating "participation of single neurons to the computation" is, that it depends on the history of the neuron. I.e., if we want to evaluate that a neuron is 50% of the time active and inactive for the remaining time, we need to sample a huge number of inputs to get an estimate of its trace (which should be around 0.5 in this example). One way to approach this issue, is to maintain the trace $\bar{x}(t)$ outside of Backprop and feed it in as external input to each forward pass $\bar{x}(t-1)$. In this case, we can restate the loss $[\bar{x}(t)-c]^2$ to:

$$L_p = \lambda\Big(\gamma\bar{x}(t-1) + (1-\gamma)x(t) - c\Big)^2 \tag{1}$$

where $x(t)$ is simply the neural activation in the current feedforward pass (the neural net doesn't know about the time parameter t, which is maintained outside).

The root of this loss formulation is:

$$R = -\frac{1}{1-\gamma}\big(\gamma\bar{x}(t-1) - c\big) \tag{2}$$

Even though the actual loss is convex, its root is constantly moving (γ and c are constant, but $\bar{x}(t-1)$ in general not). Hence, Backprop might be very unstable and easily tend to get stuck in local minima. One idea to stabilize this process to some extend, is to make sure that the root of the loss remains within a certain interval, ideally between 0 and 1. I.e., the activation x should tend to 0 if the trace is higher than c and it should tend to 1 if the trace is lower than c. We can achieve this as follows:

$$L_p = \lambda\Bigg(\gamma\bar{x}(t-1) + (1-\gamma)x(t) - c + (1-\gamma)\Big(R - \sigma\big[c - \bar{x}(t-1)\big]\Big)\Bigg)^2 \tag{3}$$

σ would be a gating function such as:

$$\sigma(x) = \frac{1}{1+e^{-\cdot k\cdot(x-x_s)}} \qquad \text{with } x_s = \frac{1}{k}ln\left[\frac{1-c}{c}\right] \tag{4}$$

where x_s ensures that $\sigma(0) = c$ and $k \approx 100$, such that derivations of the trace ($\bar{x}(t-1) \neq c$) quickly lead to an optimum value of 0 resp. 1. This has two advantages: the optimal value of the loss is relative stable and a binary behavior of the neuron is enforced (note, that binomial units have a binary output).

Ideally, we would like to replace the $\bar{x}(t-1)$ within the sigmoid with $\bar{x}(t)$. But this might screw up our convexity of the loss. However, as long as γ is close to 1, we don't need to care that much. The above loss function has the behavior, that if $\bar{x}(t-1) = c$ the root will be at c. Otherwise, the root will either tend to be at 0 or 1, but never leaving this interval, which might be a nice property.

To better understand, how this loss arises, look at the following case:

$$L_p = \lambda\Bigg(\gamma\bar{x}(t-1) + (1-\gamma)x(t) - c + (1-\gamma)\Big(R - c\Big)\Bigg)^2 \tag{5}$$

In this loss, the root will always stay at c, independent of the current trace value (just replace R in this equation).