Upper Confidence Bound

UCB is the key component of MuFasa, determining the empirical performance. $UCB(\mathbf{X}_t)$ can be formulated as

$$UCB(\mathbf{X}_t) = \lambda g(\mathbf{x}; \boldsymbol{\theta}_0) + (1 - \lambda) g(\mathbf{x}; \boldsymbol{\theta}_t)$$

where $g(\mathbf{x}; \boldsymbol{\theta}_0)$ is the gradient at initialization, $g(\mathbf{x}; \boldsymbol{\theta}_t)$ the gradient after k iterations of gradient descent, and λ is a tunable parameter to trade off between them. Intuitively, $g(\mathbf{x}; \boldsymbol{\theta}_0)$ has more bias as the weights of neural network function f are randomness initialized, which brings more exploration portion in decision making of each round. In contrast, $g(\mathbf{x}; \boldsymbol{\theta}_t)$ has more variance as the weights should be nearer to the optimum after gradient descent.

In the setting where the set of arms are fixed, given an arm $\mathbf{x}_i$, let m_i be the number of rounds that $\mathbf{x}_i$ has been played before. When m_i is small, the learner should explore $\mathbf{x}_i$ more (λ is expected to be large). Instead, when m_i is large, the leaner does not need to more exploration on it. Therefore, λ can be defined as a decreasing function with respect to m_i, such as $\frac{1}{\sqrt{m_i+1}}$ and $\frac{1}{\log m_i + 1}$. In the setting without this condition, we can set λ as a decreasing function with respect to the number of rounds t, such as $\frac{1}{\sqrt{t+1}}$ and $\frac{1}{\log t+1}$.

Unfortunately, we did not have enough time to tune the parameters. In the experiments, we simply set $\lambda = 0$.