1 Training process

Using neural networks for training of our bomberman agent, we decided to implement Rainbow (without distributional). For this technique, a number of improvements in deep learning is combined to form an integrated agent. First of all we started with Q-learning. The Q_π value is computed by choosing an action a for a given state s and calculating all of the future rewards R_n under a given policy π. Hereby future rewards are scaled by a discount factor γ to increase the weight of the immediate rewards.

$$Q_\pi(s,a) = E\left[\sum_{n=1}^{\infty} \gamma^{n-1} R_n^n\right]\Bigg|_{a,s,\pi} \tag{1}$$

The optimal Q-values are then determined by taking the maximum of all Q_π-values. To estimate these, the neural network learns a Q-function $Q(a, s, \theta_t)$ where θ_t are the weights calculated by the network. These are updated towards the target

$$Y_t^{\mathrm{Q}} \equiv R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \theta_t\right) \tag{2}$$

according to

$$\theta_{t+1} = \theta_t + \alpha\left(Y_t^{\mathrm{Q}} - Q\left(S_t, A_t; \theta_t\right)\right) \nabla_{\theta_t} Q\left(S_t, A_t; \theta_t\right) \tag{3}$$

.