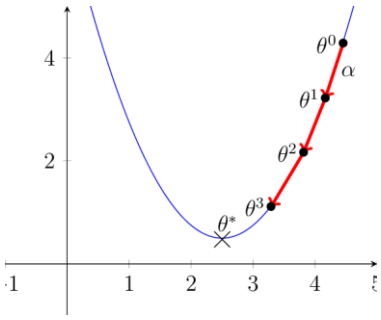
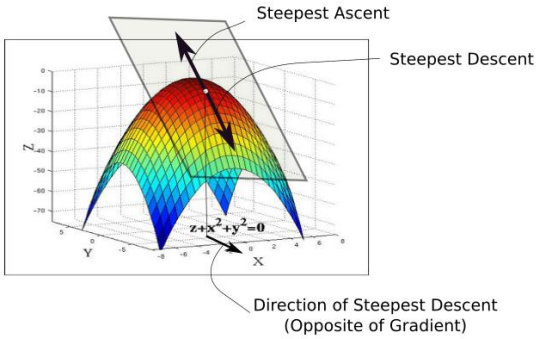
# Math foundations of ML – lesson 10

## Gradient descent

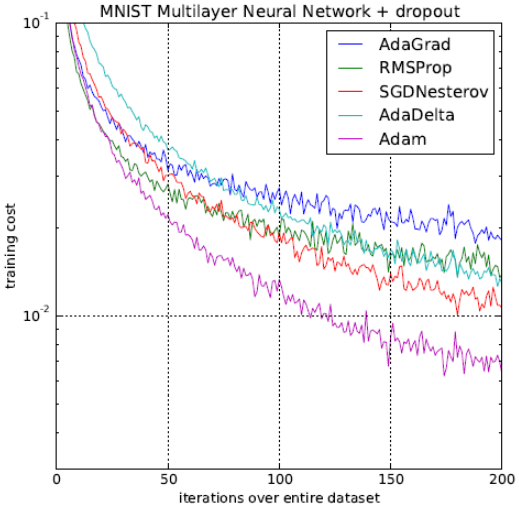



## Momentum SGD

$$
\begin{aligned}
v_t &= \gamma v_{t-1} + \eta \nabla_\theta J\left(\theta^{(t)}\right) \\
&= \gamma\left(\gamma v_{t-2} + \eta \nabla_\theta J\left(\theta^{(t-1)}\right)\right) + \eta \nabla_\theta J\left(\theta^{(t)}\right) \\
&= \gamma^2 v_{t-2} + \gamma\eta \nabla_\theta J\left(\theta^{(t-1)}\right) + \eta \nabla_\theta J\left(\theta^{(t)}\right) \\
&\cdots \\
&= \gamma^n v_{t-n} + \eta\left(\gamma^{n-1} \nabla_\theta J\left(\theta^{(t-n+1)}\right) + \cdots + \gamma \nabla_\theta J\left(\theta^{(t-1)}\right) + \nabla_\theta J\left(\theta^{(t)}\right)\right)
\end{aligned}
$$

## Adam



**Loss functions for segmentation**

Let $p = (p_1, \ldots, p_n)$ be the ground truth labels of the segmentation. This means that $p_i \in \{0,1\}$. Let $\tilde{p} = (\tilde{p}_1, \ldots, \tilde{p}_n)$ be an approximation (e.g. as generated through a deep learning network) of the segmentation, with $0 \le \tilde{p}_i \le 1$. One can ensure that the network outputs 'probability pixels' by applying at the last layer at each pixel the logistic function

$$\sigma(t) := \frac{1}{1+e^{-t}}.$$

**Negative log-likelihood as a loss for image segmentation**

We can define a loss per image

$$-\sum_{i=1}^{n} p_i \log \tilde{p}_i + (1-p_i)\log(1-\tilde{p}_i)$$

**Simple differentiable approximations to the Jaccard index**

It is clear that in unbalanced cases, where there are more ground truth background pixels than ground truth object pixels, the standard negative log-likelihood loss is potentially biased.

One would like to train a DL segmentation network and measure/maximize performance using Intersection Over Union (IoU) loss, which is also called the Jaccard loss

$$0 \le \frac{|A \cap B|}{|A \cup B|} \le 1.$$

The Jaccard index in our special is

$$J(p, \tilde{p}) = \frac{\#\{i : p_i = 1 \text{ and } \tilde{p}_i \ge 0.5\}}{\#\{i : p_i = 1 \text{ or } \tilde{p}_i \ge 0.5\}}.$$

Since the Jaccard index is not differentiable, we look for surrogates for which we can compute gradients and thus optimize. We use an approximation to the Jaccard index which is differentiable and based on the equality

$$\frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|},$$

Let

$$\langle p, \tilde{p} \rangle = \sum_{i=1}^{n} p_i \tilde{p}_i, \qquad |p|_1 = \sum_{i=1}^{n} p_i, \qquad |\tilde{p}|_1 = \sum_{i=1}^{n} \tilde{p}_i.$$

We define a differentiable 'loss' function that we wish to maximize

$$\tilde{J}\left(p,\tilde{p}\right) \coloneqq \frac{\left\langle p,\tilde{p}\right\rangle}{\left|p\right|_1+\left|\tilde{p}\right|_1-\left\langle p,\tilde{p}\right\rangle}.$$

Note that in the special case where $\tilde{p}_i \in \{0,1\}$, $i=1,\ldots,n$, we get

$$\tilde{J}\left(p,\tilde{p}\right) = J\left(p,\tilde{p}\right).$$

The two losses are sometimes combined with a weight. There are other more complicated surrogates (e.g. Lovász).