Gradient descent

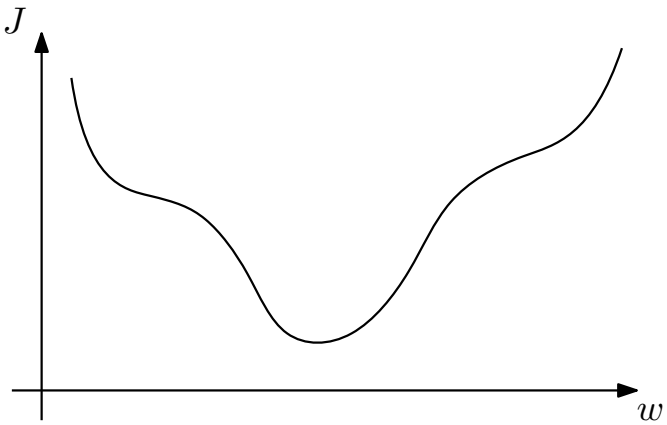
Herman Kamper

Gradient descent

We have some function $J(\mathbf{w})$ that we want to minimise with respect to parameters $\mathbf{w}$.

Idea: Start with a random $\mathbf{w}$ and then keep moving it in a direction that reduces $J(\mathbf{w})$.

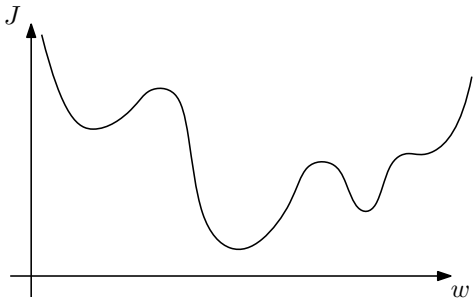
In one dimension:



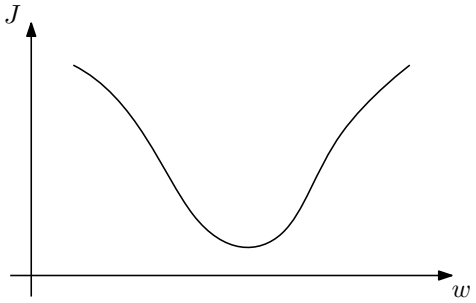
$$w \leftarrow w - \eta \frac{\mathrm{d}J}{\mathrm{d}w}$$

Potential problems

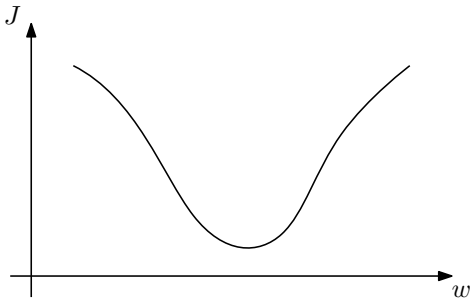
Could get stuck in a local minimum:



If η too small:



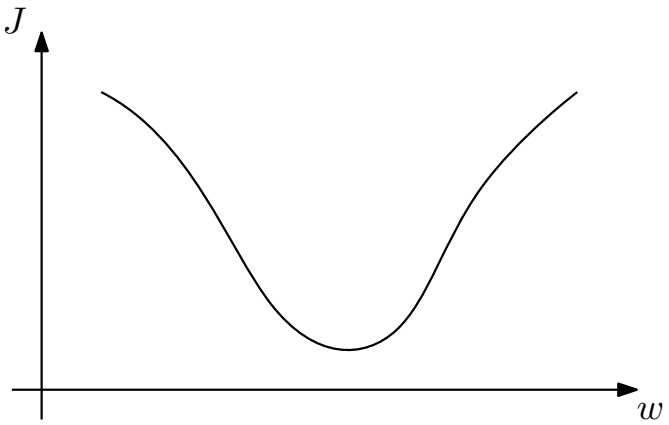
If η too big:



Other aspects

Step sizes

As we get closer to the minimum, the step sizes automatically get smaller:



In D dimensions

$$
\begin{aligned}
w_0 &\leftarrow w_0 - \eta \frac{\partial J}{\partial w_0} \\
w_1 &\leftarrow w_1 - \eta \frac{\partial J}{\partial w_1} \\
&\vdots \\
w_D &\leftarrow w_D - \eta \frac{\partial J}{\partial w_D}
\end{aligned}
$$

Or directly in vector form:

$$
\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial J}{\partial \mathbf{w}}
$$

Demo: Fitting a Gaussian with gradient descent

NLL loss: $J(\mu, \sigma^2)$

Mean update:

$$\hat{\mu} \leftarrow \hat{\mu} - \eta \frac{\partial J}{\partial \hat{\mu}}$$

Variance update:

$$\hat{\sigma}^2 \leftarrow \hat{\sigma}^2 - \eta \frac{\partial J}{\partial \hat{\sigma}^2}$$

(It is a bit silly to use gradient descent for this problem since we have a closed-form solution that directly gives us the best NLL. So this is just for illustrative purposes. But, it is interesting to ask whether for some machine learning models in some cases it might be better to use gradient descent rather than the closed form solution. Can you think of an example?)

Videos covered in this note

- Gradient descent 1: Fundamentals (11 min)