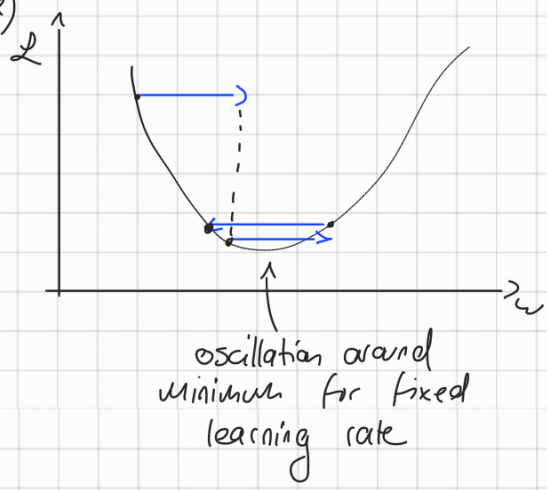
Exercise 1: Optimization and Training Tricks

1)

- Batch Gradient Descent (BGD): The Loss gradient is computed on the whole Training set before an update is made. While this is ideal for convex problems because the gradient is noiseless, pointing to the true minimum, the computational cost of just doing one update per epoch is very high. Thus BGD is rather slow and not apt for very large data sets. Additionally, BGD might get stuck in a local minimum due to the absence of noise.
- Stochastic Gradient Descent (SGD): The loss gradient is computed and the update to the model's parameters is computed for each instance. This reduces computational cost making it ideal for large datasets. The thus computed gradient is very noise, which helps to escape local minima, but can also cause oscillation around a mining.
- Minibatch SGD: The Loss gradient is computed and the update is performed on a (small) subset of the training set. This is a compromise between BGD and SGD. Consequently it benefits from the advantages of both, ie. reduced noise and computational cost, while mitigating the disadvantages, ie. no noise or too much noise and high computational cost. Minibatch is still able to escape local minima, due to the noise introduced by subsampling the training set.

	BGD	SGD	Minibatch SGD
Computational cost	high	low	medium
Speed of convergence: #epochs	few	many	moderately many
Speed of convergence: time	slow (esp. for large Datasets)	fast	often fastest (=> allows vectorized operations)

2) a)



Getting closer to a Loss minimum while maintaining the same large learning rate makes it likely to 'jump' over that minimum and eventually oscillate around it.

Thus it is common practice to reduce the learning rate going further into training to be able to go deeper into a minimum, while still approaching these minima fast by the initially large learning rate. and jumping over sub-optimal local minima.

b)

Exponential decay: $\tau(t) = \tau_0 \exp(-\lambda t)$

with decay rate λ and training step t.

Reduces the learning rate exponentially which yields the above mentioned advantages.

3)

a)

The training set is used to train model.

The validation set is used to judge the model performance and optimise its hyperparameters accordingly

The test set is not involved in training or hyperparameter optimisation, but merely serves to measure the models

generalisation capability.

b) This would implicitly involve the test set in the training process, which contradicts the principle of generalisation to unseen data and thus undermines the purpose of the test set.

c) In grid search the hyperparameter space is scanned over and for each point the model is trained and test using the validation set. That way the best hyperparameters are determined.

4) a)

i) RMSprop: The gradient is normalised using the root mean square of the previous gradients:

$$\Delta\theta_{t-1}^{RMS} = \sqrt{\frac{1}{t-1}\sum_{\tau=1}^{t-1}\Delta\theta_\tau^2} \qquad (1D)$$

where this RMS is updated using a decay $\beta < 1$:

$$\Delta\theta_t^{RMS} = \beta\,\Delta\theta_{t-1}^{RMS} + (1-\beta)\,\Delta\theta_t$$

Leading to a training step of the form:

$$\theta_t = \theta_{t-1} + \tau\frac{\Delta\theta_t}{\sqrt{\Delta\theta_{t-1}^2+\varepsilon}} \qquad \text{with } \varepsilon \ll 1 \text{ preventing division by zero.}$$

This leads to a reduced learning relatively fast because the gradients commonly decrease faster than the accumulated RMS if $\beta \gg 0$.

However, this method is able to adapt too large or too small learning rates through the RMS.

ii) Momentum: The gradient is combined with the gained "momentum" of past gradients via:

$$p_t = \gamma\, p_{t-1} + (1-\gamma)\,\delta\theta_t$$

with momentum p_i and momentum decay γ.

That way even if the current gradient would be zero, there would still be some momentum left, "pushing" the model out of eg. a local minimum.

b)

Adam: $$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \,, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$m_{t-1} = 0{,}5$, $v_{t-1} = 0{,}2$, $g_t = 2$, $\beta_1 = 0.9$, $\beta_2 = 0{,}99$, $\alpha = 0.01$, $\varepsilon = 10^{-8}$

i)

$m_t = 0{,}65$, $v_t = 0{,}238$

ii)

$\Delta w = -0{,}0133$

iii)

- $v_t' \gg v_t$: the history factors in much more than the current gradient with $\beta_2 = 0{,}99$
- $|\Delta w_t'| \ll |\Delta w_t|$: the effective learning rate decreases accounting for the previous very large gradients
- Adam seeks to keep the learning rate on an appropriate level since the history of second moments counters the development of too large/small training steps.