Section 5: Optimization.

- Batch Gradient Descent: Take all the samples in one iteration. In this case, iteration = epoch. Compute the gradient of the loss with respect to the network's weights for the entire training dataset, and then perform one update.

- Stochastic gradient Descent: Take in one sample at each iteration
  Number of iteration per epoch = number of samples.

  For each samples $x^{(i)}$ and $y^{(i)}$ $\{\theta = \theta - \eta . \nabla_\theta J(\theta, x^{(i)}, y^{(i)})\}$.

- Mini-batch Gradient Descent: Takes the best from both the batch and Schochastic Gradient Descent
  Number of iterations per epoch = samples / batch size.

  For each batch $x^{(i;i+n)}$ and $y^{(i;i+n)}$ {
  $\theta = \theta - \eta . \nabla_\theta J(\theta; x^{(i;i+n)}; y^{(i;i+n)})$
  }.

- Exponentially weighted average intuition - Bias correction - Momentum

  - EWA: $v_t = \beta v_{t-1} + (1-\beta)\theta_t$

    If $\beta = 0,9$ then $v_t = 0,9\, v_{t-1} + 0,1\, \theta_t$.

    ($0,9$: More weights to previous values; $0,1$: Less weighs to current values)

    The general idea

- Bias Correction of EWA

  $v_t = \beta v_{t-1} + (1-\beta)\theta_t$.

  If we start with $v_0 = 0$, (assum $\beta = 0,98$) then $v_2 = 0,0196\,\theta_1 + 0,02\,\theta_2$

  ($0,0196$: keep smaller)

  Solution: Divide v_t by $1-\beta^t$

  Example: At $t=2$: $1-\beta^t = 1-0,98^2 = 0,0396$.

  $\frac{v_2}{0,0396} \simeq 0,494\,\theta_1 + 0,505\,\theta_2$.

  (Okay!)

KI.ONG

- Momentum.

Compute dw and db on your minibatch, then.

$$v_{dw} = \beta v_{dw} + (1-\beta) dw$$

$$v_{db} = \beta v_{db} + (1-\beta) db$$

Then, $w = w - \alpha v_{dw}$; $b = b - \alpha v_{db}$

- RMSProp: We know dw is small, db is large (abt variation)

$$s_{dw} = \beta s_{dw} + \underbrace{(1-\beta)}_{\text{small}} dw^2 \; ; \; s_{db} = \beta s_{db} + (1-\beta) db^2 \leftarrow \text{large}$$

(dw^2, db^2 are element-wise squared)

$$w_{new} = w_{old} - \frac{\alpha}{\sqrt{s_{dw}} + \varepsilon} dw \; ; \; b_{new} = b_{old} - \frac{\alpha}{\sqrt{s_{db}} + \varepsilon} db$$

- Adam Optimization:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad \text{(momentum)}$$

$$v_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2 \quad \text{(RMSProp)}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \; , \; \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Update: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t$

- SWATS (Switching from ADam to SGD):

- Weight decay and Regularization:

- Rule: $w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$.

- Adding terms to the cost: $\tilde{E}(w) = E(w) + \frac{\lambda}{2} w^2$.

Then: $w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} - \underline{\eta \lambda w_i}$

- Decoupling weight decay: Adding Weight Decay to Momentum and Adam

- SGD w/ momentum and weight decay update:

$$v_t = \gamma v_{t-1} + \eta g_t$$

$$\theta_{t+1} = \theta_t - v_t - \underline{\eta w_t \theta_t} \leftarrow \text{decoupled weight decay}$$

- Similarly for Adam, (AdamW)

- AmsGrad

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} m_t$$