Loss and Optimizer Functions

Mass Street University

PER EDUCATIONEM PROGRESSUS

Common Optimizer functions

- RMSprop
- Adagrad
- Adadelta
- Adam
- Adamax
- Nadam
- AMSGrad

Mass Street University

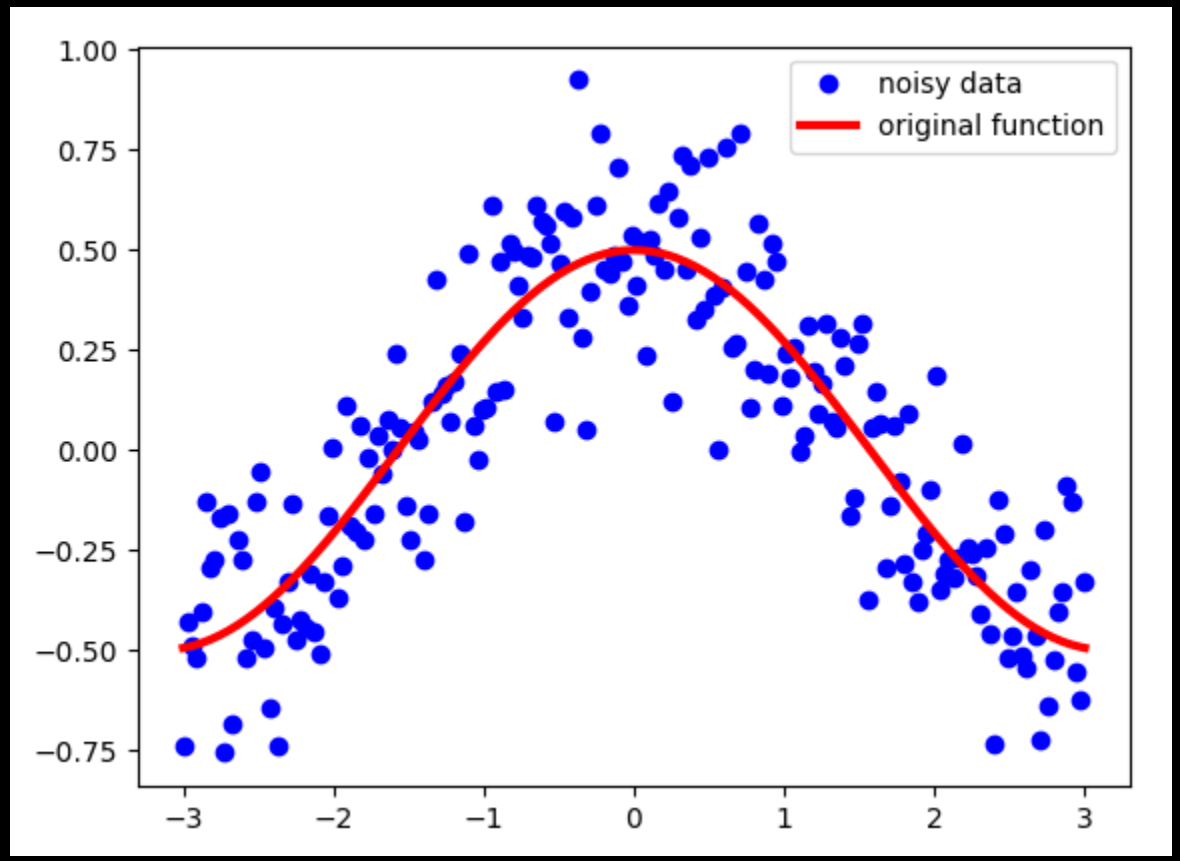
PER EDUCATIONEM PROGRESSUS

Momentum

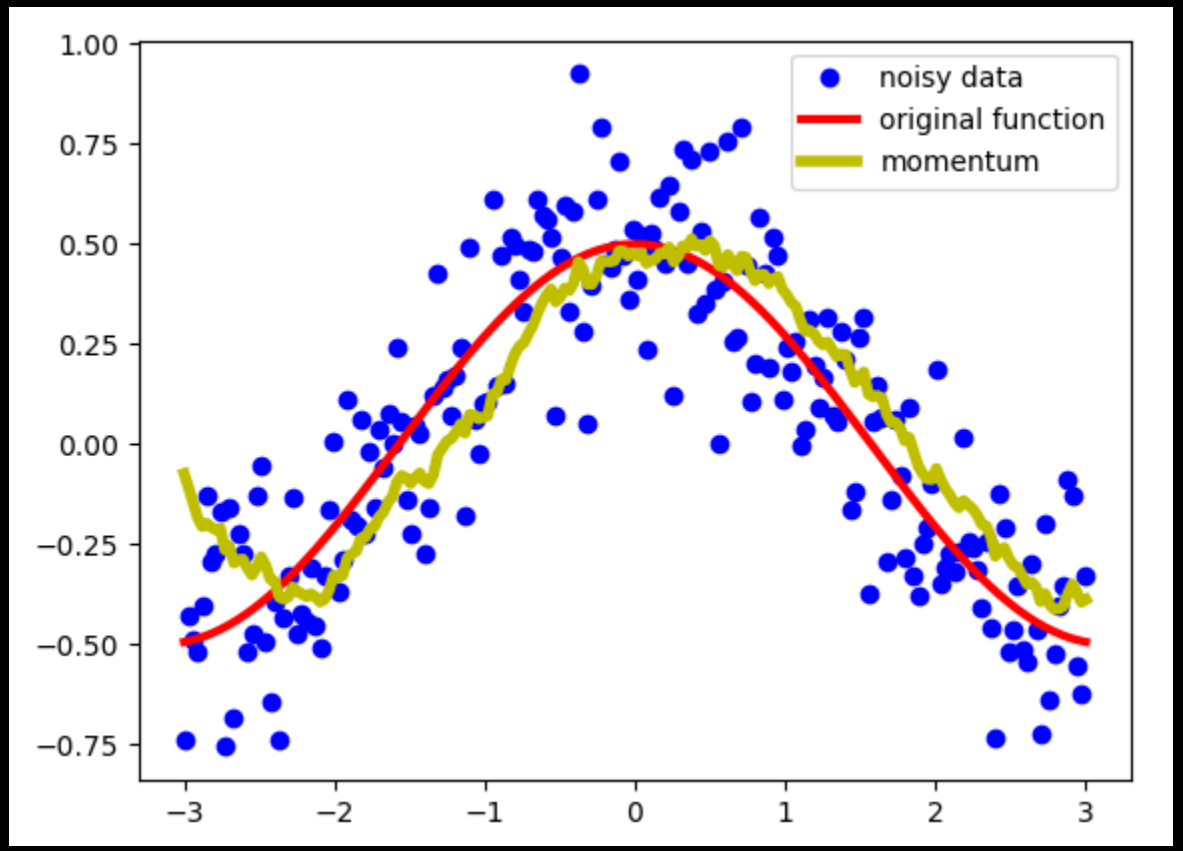
- Is an addition to SGD that makes it work better.
- Accelerates gradient vectors towards the global minima.
- Works by exponentially weighting averages of the gradients over time.
- Makes the noisy gradient of SGD manageable.

Momentum in action

Without Momentum



With Momentum

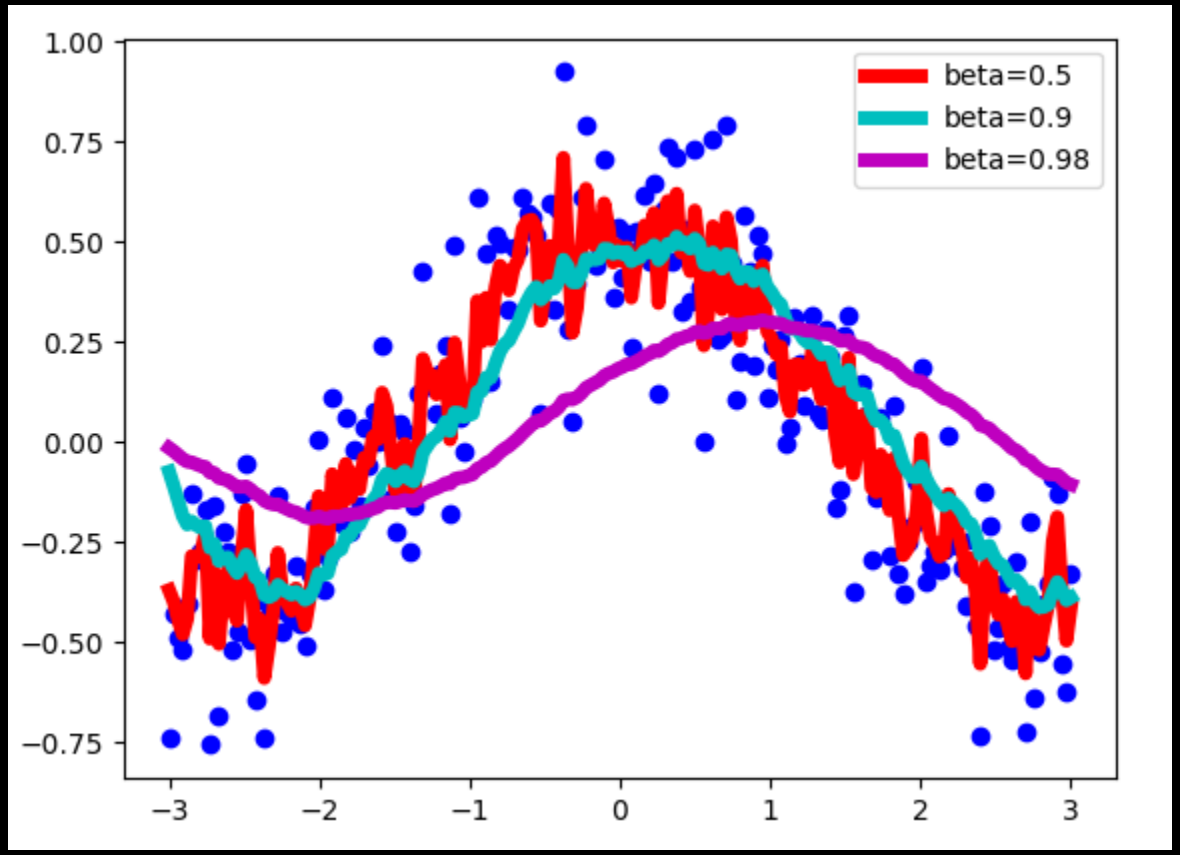


Momentum Math and effects

$$V_t = \beta V_{t-1} + (1-\beta) S_t$$
$$\beta \in [0,1]$$

Early Gradient Fix

$$V_t = \frac{V_t}{1-b^t}$$

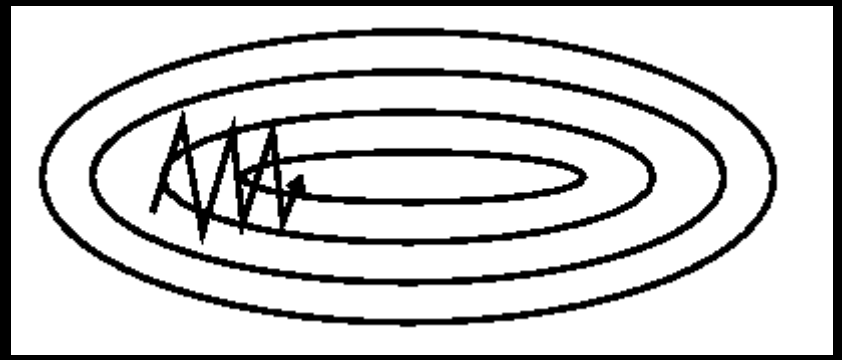


Mass Street University
PER EDUCATIONEM PROGRESSUS

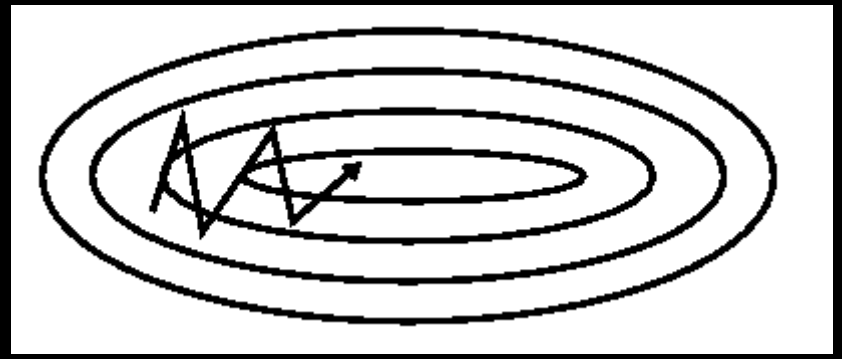
Why it works

- It always tends to have a straighter path and needs to correct itself less. Thus improving computation time and cost.
- It can deal with difficult gradients such as Ravines and saddle points.

Ravines without Momentum



Ravines with Momentum



RMSprop

- Is an adaptive learning rate method.
- Each weight has its own learning value.
- Neurons can be individually tuned.
- Can increase and decrease learning rate and step size down the gradient.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\delta C}{\delta w}\right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\delta C}{\delta w}$$

Mass Street University

PER EDUCATIONEM PROGRESSUS

AdaGrad

- It also has different learning rates for each weight.
- It adapts the weights to each dimension of the gradient.
- Can no longer train after a certain point.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}.$$

Adadelta

- Takes only a subset of past gradients.
- It uses a decaying average of past gradients.
- Was created independently the same time RMSprop was.
- Adadelta seeks to update parameters of the network instead of learning rates.

$$\mathbb{E}[g^2]_t = \gamma \mathbb{E}[g^2]_{t-1} + (1-\gamma) g_t^2,$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} \cdot g_t.$$

Adam

- Combines AdaGrad and RMSProp
- It calculates two different decay rates: the exponential moving gradient and the squared gradient.
- Can find a worse solution than SGD in some situations.

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

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

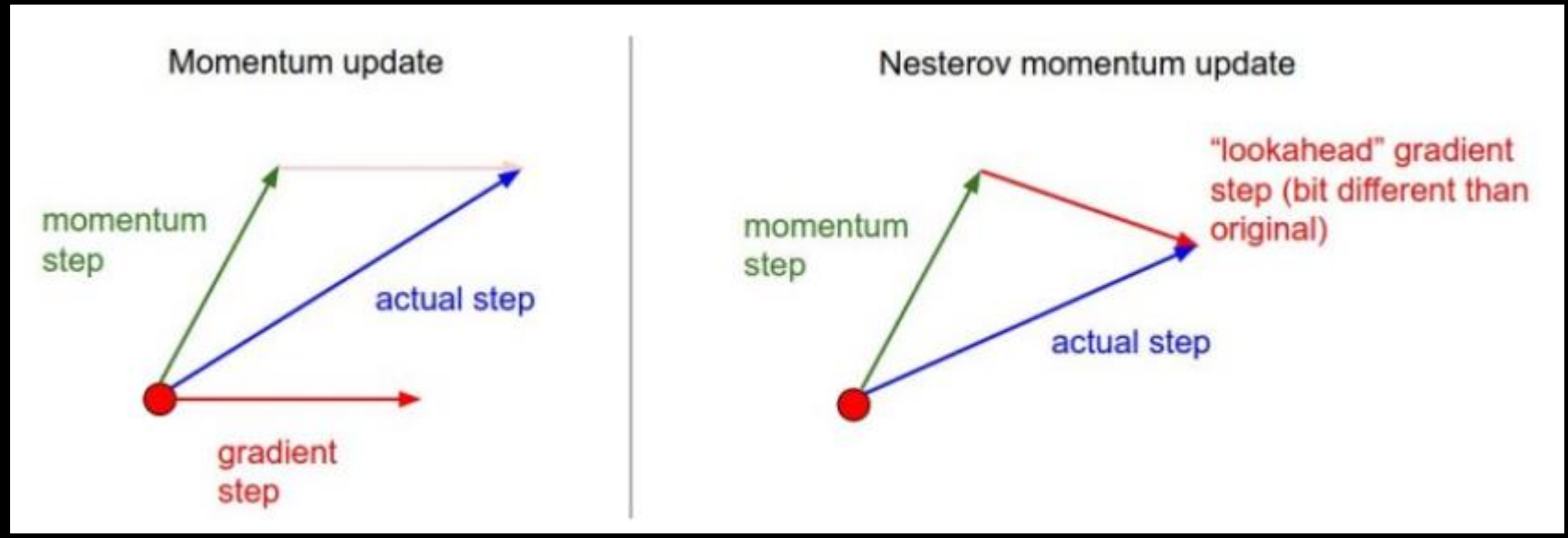
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Adamax

- Replaces L2 Norm in Adam with L_∞.
- No longer uses the bias constraint since the denominator no longer decreases towards zero.
- Makes the network more stable and converge at the global minima.

Nesterov Momentum

- Adds a new term to momentum to correct the course of descent before it gets off track.
- Creates an even straighter path than momentum.
- Makes the model do less back tracking and converge even faster.



Mass Street University
PER EDUCATIONEM PROGRESSUS

Nadam

- Just Adam but with Nesterov added onto the momentum term.
- Is generally faster and more stable than plain Adam.

Mass Street University

PER EDUCATIONEM PROGRESSUS

AMSgrad

- Captures more important information than every other method.
- Is the newest algorithm and isn't super tested.
- Theoretically works the best.

$$
\begin{aligned}
u_t &= \beta_2^{\infty} v_{t-1} + (1 - \beta_2^{\infty}) |g_t|^{\infty} \\
&= \max(\beta_2 \cdot v_{t-1}, |g_t|)
\end{aligned}
$$

Mass Street University

PER EDUCATIONEM PROGRESSUS

Advanced Loss Functions

- Squared Hinged
- Kullback Leibler Divergence

Mass Street University

PER EDUCATIONEM PROGRESSUS

Squared Hinge

- Uses -1 and 1 as target variables instead of 0 and 1.
- Tries to match the same signs of the predicted and actual labels.
- Collects the probabilities that the model gave as well as the actual vs predicted labels.
- It penalizes the model for being unsure about its predictions.

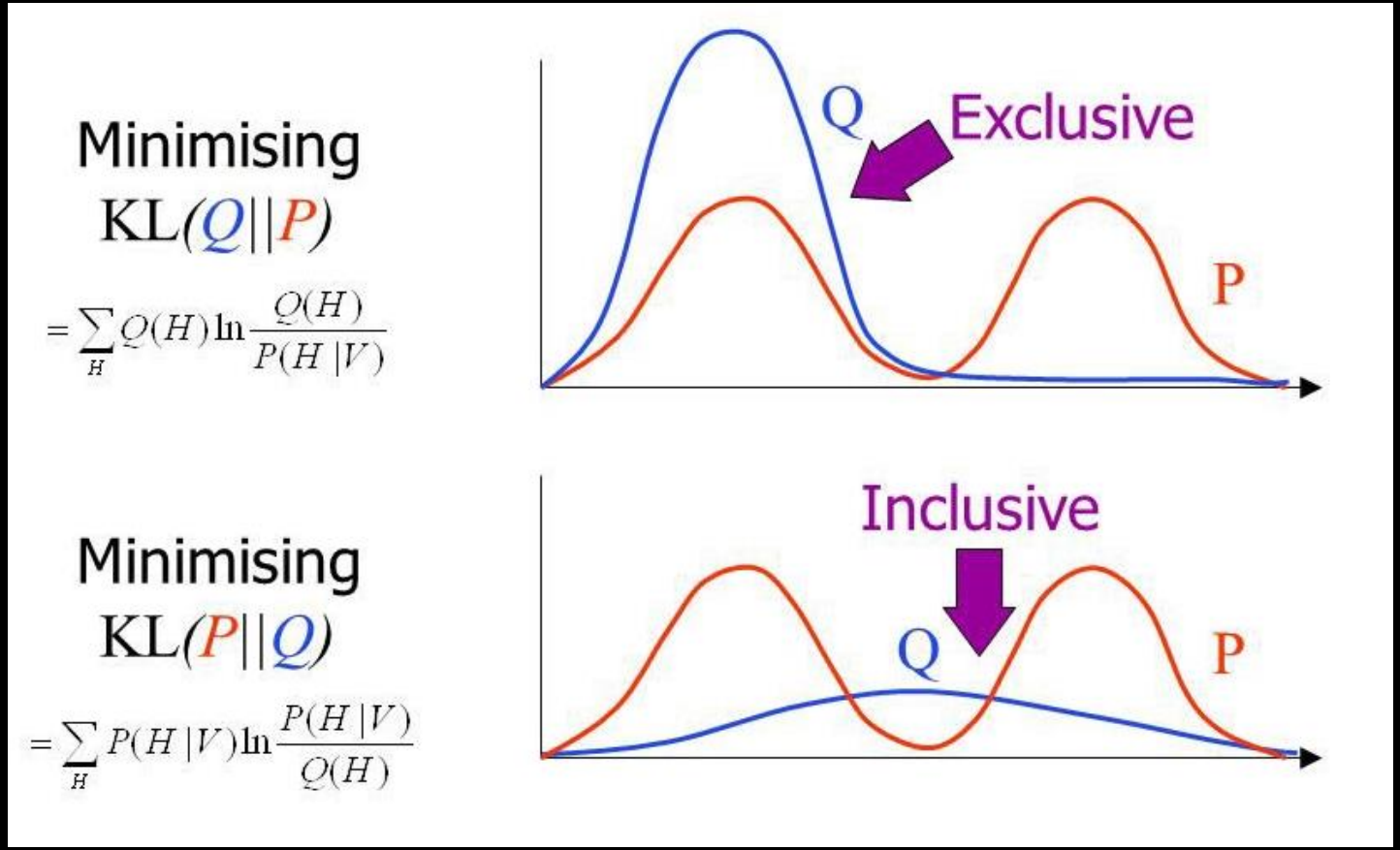
$$L(y, \hat{y}) = \sum_{i=0}^{N} (Max(0, 1 - y_i \cdot \hat{y}_i)^2)$$

Mass Street University

PER EDUCATIONEM PROGRESSUS

KL Divergence

- Measures how different the probability distribution is from the actual distribution of the data.
- Calculates how much information is loss if it tries to use the predicted distribution to approximate the original.



Mass Street University
PER EDUCATIONEM PROGRESSUS

Standard Operating Procedure

- Try Nadam, RMSprop, Adadetla, Nesterov Momentum, and AMSgrad.
- Try cross-entropy and Hinge Loss. If Hinge Loss works well use Squared Hinge loss. If nothing works use KL Divergence or MSE.

Mass Street University

PER EDUCATIONEM PROGRESSUS