NPTEL ONLINE CERTIFICATION COURSES

Course Name: Deep Learning

Faculty Name: Prof. P. K. Biswas

Department : E & ECE, IIT Kharagpur

Topic

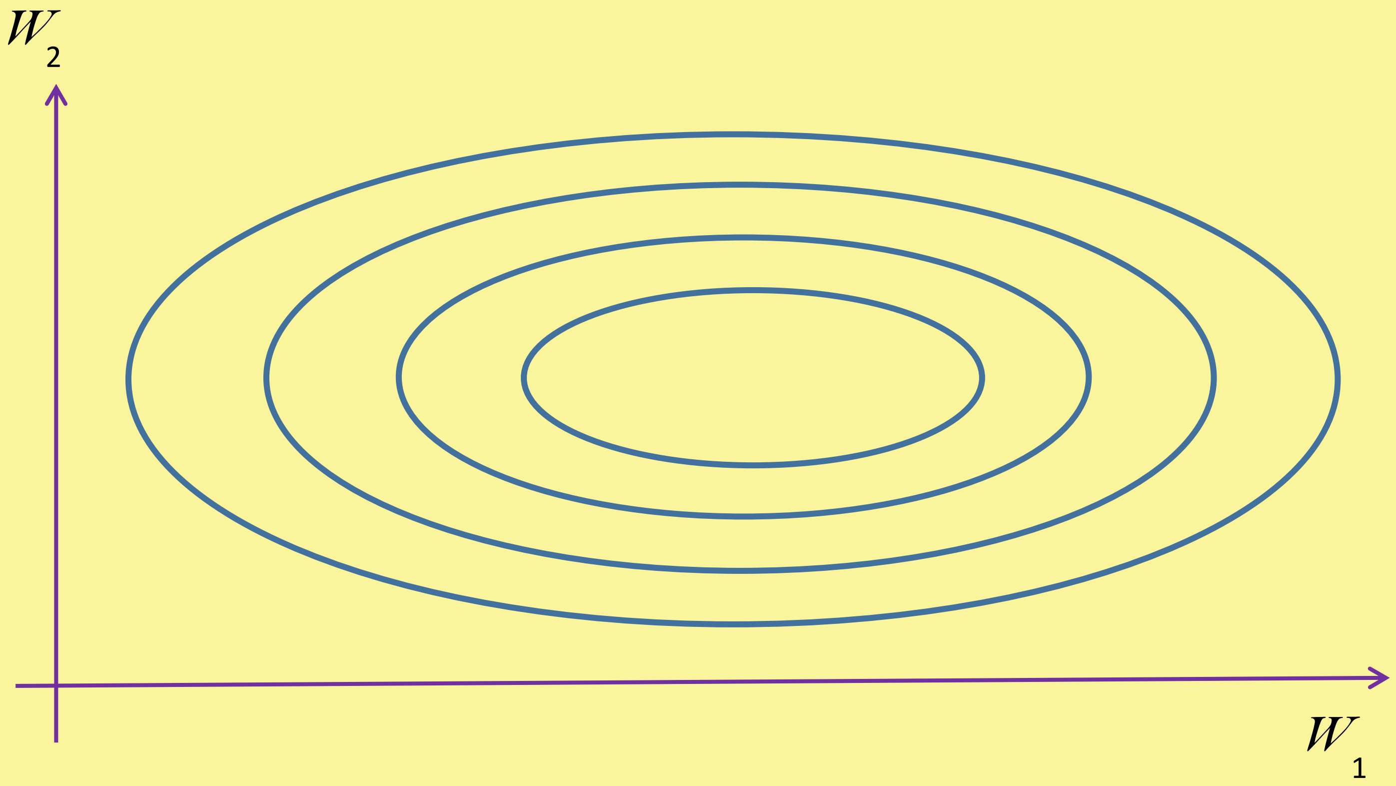
Lecture 44: Optimizing Gradient Descent II

CONCEPTS COVERED

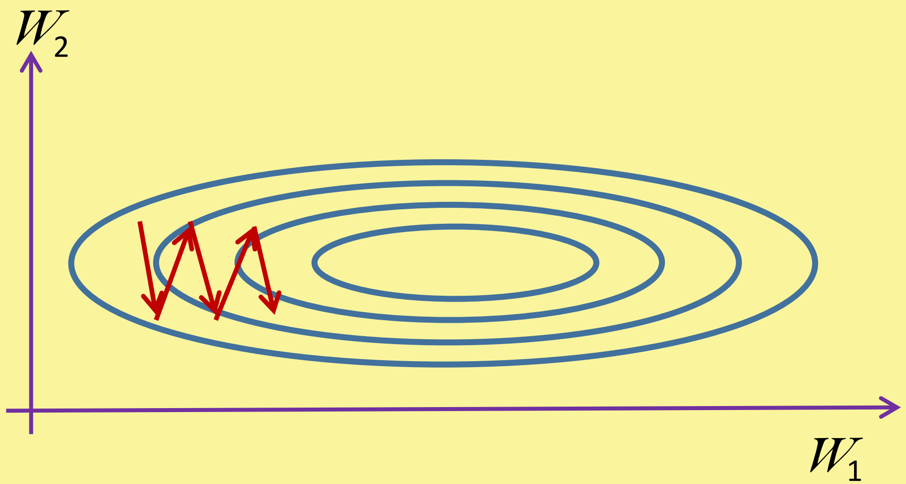
Concepts Covered:

- CNN
  - Gradient Descent Challenges
  - Momentum Optimizer
  - Nesterov Accelerated Gradient
  - Adagrad
  - RMSProp
  - etc.

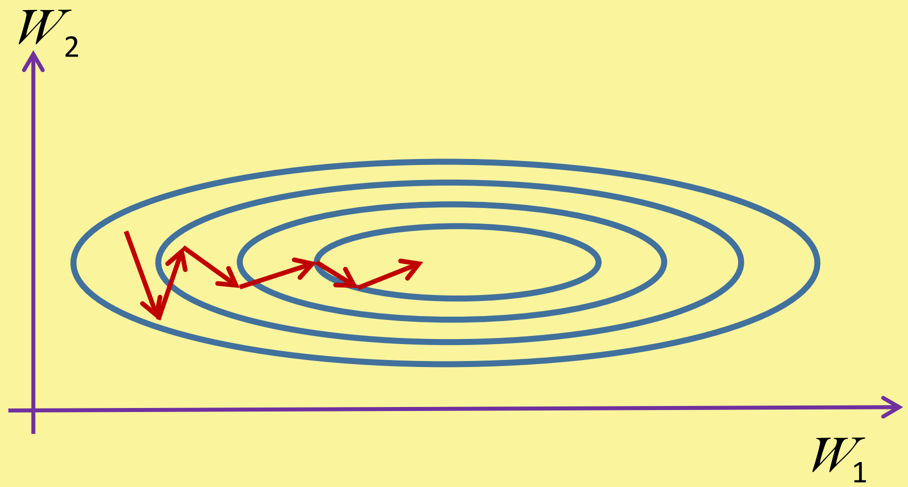
Momentum Optimizer



Momentum Optimizer



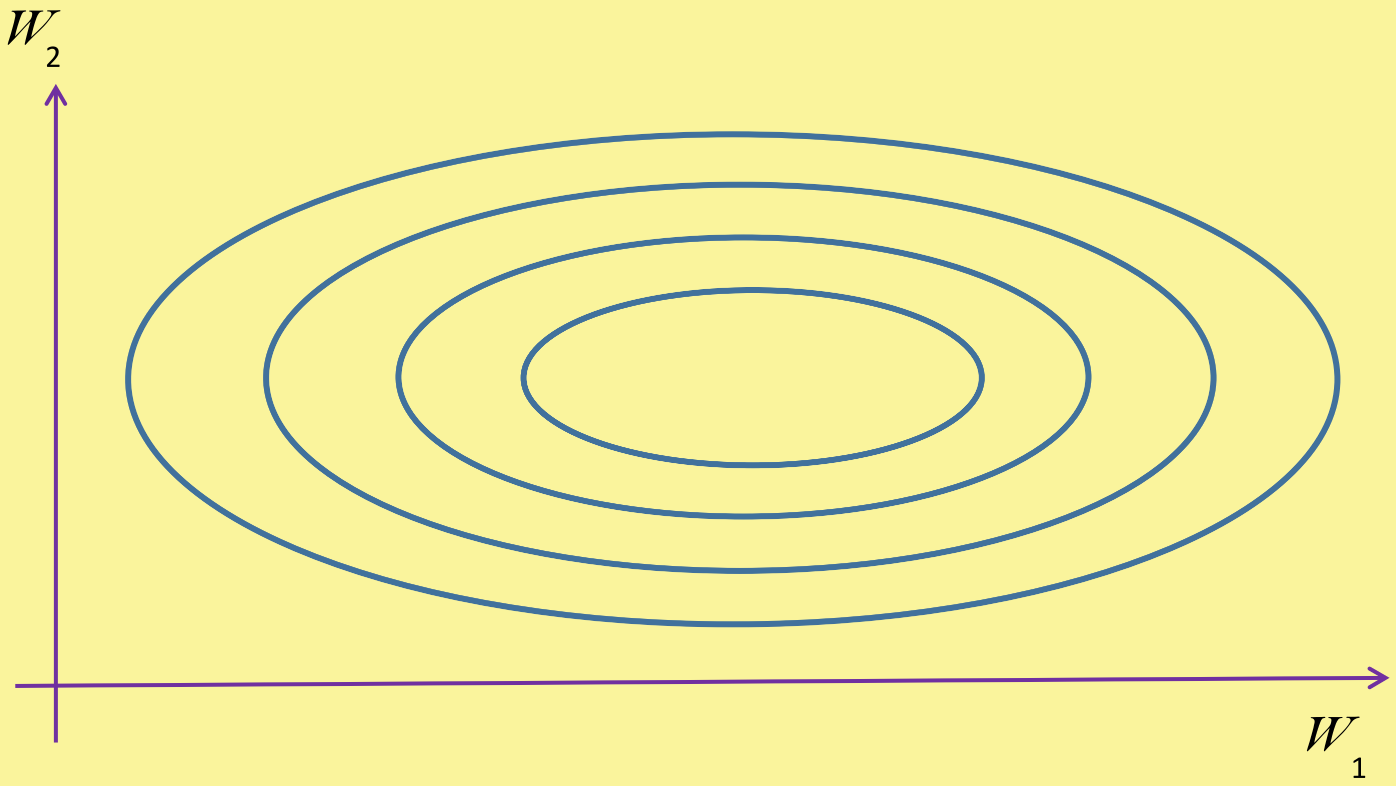
SGD



SGD with Momentum

Nesterov Accelerated Gradient (NAG)

Nesterov Accelerated Gradient (NAG)



Problem with Momentum Optimizer/NAG

- Both the algorithms require the hyper-parameters to be set manually.
- These hyper-parameters decide the learning rate.
- The algorithm uses same learning rate for all dimensions.
- The high dimensional (mostly) non-nonconvex nature of loss function may lead to different sensitivity on different dimension.
- We may require learning rate be small in some dimension and large in another dimension.

Adagrad

Adagrad

- Adagrad adaptively scales the learning rate for different dimensions.
- Scale factor of a parameter is inversely proportional to the square root of sum of historical squared values of the gradient.
- The parameters with the largest partial derivative of the loss will have rapid decrease in their learning rate.
- Parameters with small partial derivatives will have relatively small decrease in learning rate.

Adagrad

$$g_t = \frac{1}{n} \sum_{\forall X \in Minibatch} \nabla_W L(W_t, X) \qquad r_t = \sum_{\tau=1}^{t} g_\tau \circ g_\tau$$

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\in I + r_t}} \circ g_t$$

$$\circ \rightarrow \text{element - wise product}$$

Adagrad

$$\begin{bmatrix} W_{t+1}^{(1)} \\ W_{t+1}^{(2)} \\ \vdots \\ W_{t+1}^{(d)} \end{bmatrix} = \begin{bmatrix} W_{t}^{(1)} \\ W_{t}^{(2)} \\ \vdots \\ W_{t}^{(d)} \end{bmatrix} - \begin{bmatrix} \dfrac{\eta}{\sqrt{\in + r_t^{(1)}}} . g_t^{(1)} \\ \dfrac{\eta}{\sqrt{\in + r_t^{(2)}}} . g_t^{(2)} \\ \vdots \\ \dfrac{\eta}{\sqrt{\in + r_t^{(d)}}} . g_t^{(d)} \end{bmatrix}$$

Adagrad

Positive Side:

- Adagrad adaptively scales the learning rate for different dimensions by normalizing with respect to the gradient magnitude in the corresponding dimension.
- Adagrad eliminates the need to manually tune the learning rate.
- Reduces learning rate faster for parameters showing large slope and slower for parameters giving smaller slope.
- Adagrad converges rapidly when applied to convex functions.

Adagrad

Negative side:

- If the function is non-convex:- trajectory may pass through many complex terrains eventually arriving at a locally region.
- By then learning rate may become too small due to the accumulation of gradients from the beginning of training.
- So at some point the model may stop learning.

NPTEL ONLINE CERTIFICATION COURSES

Thank you