CS725: OPTIMIZERS

Why do we need more support than what a vanilla SGD learner offers?

Two potential pitfalls:

(A) Complex (non-convex) loss landscapes that could result in gradients that vary a lot (Soln: Adaptive LR that varies across dimensions)

(B) Can get stuck in "flat" or "plateau" regions of the loss function (Soln: Momentum)

① SGD with momentum

first, define a "velocity" vector

$$v_t \leftarrow \beta v_{t-1} + \eta g_t \quad \text{where} \quad \boxed{g_t = \nabla_w L(w_{t-1})}$$

$$(0 \leq \beta < 1)$$

New weight update rule for SGD w/ momentum: $w_t \leftarrow w_{t-1} - v_t$

Expand $V_t = \beta V_{t-1} + \eta g_t$

$$= \eta g_t + \beta(\eta g_{t-1} + \beta V_{t-2})$$

$$= \eta g_t + \beta \eta g_{t-1} + \beta^2 V_{t-2}$$

$$= \eta g_t + \beta \eta g_{t-1} + \dots + \beta^t V_0$$

$\rightarrow$ If $V_0 = 0$, then this term disappears

What happens to V_t if $\beta = 0$? Recover Std/vanilla SGD

If β is a high value ≈ 1, many past gradients will influence w_t

If β is small ≈ 0, then very few past gradients will influence w_t

② RMSProp [for adaptive learning rates]

Consider weights w_1, w_2 where $w_1 \leftarrow w_1 - \eta \frac{\partial L}{\partial w_1}$

$$w_2 \leftarrow w_2 - \eta \frac{\partial L}{\partial w_2}$$

Say $\frac{\partial L}{\partial w_1} \gg \frac{\partial L}{\partial w_2}$

If η is high, then w_1 can diverge

If η is low, then the update to w_2 will be very small

RMSProp defines a new vector s_t to adaptively change the learning rate across the weight dimensions

$$s_t \leftarrow \gamma s_{t-1} + (1-\gamma)\, g_t \odot g_t$$

elementwise product or Hadamard product

$$(0 \leq \gamma < 1)$$

Expand

$$\begin{aligned} s_t &= (1-\gamma)\, g_t \odot g_t + \gamma s_{t-1} \\ &= (1-\gamma)\, g_t \odot g_t + \gamma\left((1-\gamma)\, g_{t-1} \odot g_{t-1} + \gamma s_{t-2}\right) \\ &= (1-\gamma)\, g_t \odot g_t + (1-\gamma)\gamma\, g_{t-1} \odot g_{t-1} + \dots + \gamma^t s_0 \end{aligned}$$

What do the coefficients in s_t add up to? ≈ 1 for large t

S_t is an exponentially weighted moving average

New weight update rule:

$$w_t \leftarrow w_{t-1} - \frac{\eta}{\sqrt{S_t + \epsilon}} \odot g_t$$

epsilon is added to avoid divide-by-zero errors

③ ADAM optimizer combines ideas from SGD w/ momentum and RMS Prop

$$V_t \leftarrow \beta V_{t-1} + (1-\beta)\, g_t$$

$$S_t \leftarrow \gamma S_{t-1} + (1-\gamma)\, g_t \odot g_t$$

$$\hat{S}_t \leftarrow \frac{S_t}{1-\gamma^t}, \quad \hat{V}_t \leftarrow \frac{V_t}{1-\beta^t}$$

// BIAS CORRECTION

Needed to correct bias in S_t and V_t for small values of t

Wt update: $w_t \leftarrow w_{t-1} - \frac{\eta}{\sqrt{\hat{S}_t + \epsilon}} \odot \hat{V}_t$

CONVOLUTIONAL NEURAL NETWORKS

CNNs are motivated by the visual cortex and aim to satisfy the following desired properties;

(A) Locality or modeling spatial information

(B) Translation invariance → (model behaviour should be identical regardless of where an object appears in an image)

(C) Parameter-efficient