11-785 14/05/2020

US convergence - review

(A1): see DL book (Q): moving in opposite direction of gradient

- $y = f(x)$; $\frac{dy}{dx} = f'(x)$
- $f(x+\epsilon) \approx f(x) + \epsilon f'(x)$ (from definition of derivative as $\lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$)
- $f(x - \epsilon \, \text{sign}(f'(x))) < f(x)$ for small enough ϵ (?)
- Reduce $f(x)$ by moving x in small steps with opposite sign of derivative
- Best understood graphically ; formal results (?)

(A1) ✓: The directional derivative in $\underline{u}$ (a unit vector) is the slope of function f in direction u.

- Formally ; it is the derivative of $f(\underline{x} + \alpha \underline{u})$ with respect to α, evaluated at $\alpha = 0$.
- via chain rule :- $\frac{\partial}{\partial \alpha} f(\underline{x} + \alpha \underline{u}) \Big|_{\alpha=0} = \underline{u}^T \nabla_{\underline{x}} f(\underline{x})$ ($f(\cdot)$ - scalar fn)

(*)

- minimise f ; find direction in which f decreases fastest
- use the directional derivative

$$\min_{\underline{u}, \underline{u}^T\underline{u}=1} \underline{u}^T \nabla_{\underline{x}} f(\underline{x}) = \min_{\underline{u}, \underline{u}^T\underline{u}=1} \|\underline{u}\|_2 \|\nabla_{\underline{x}} f(\underline{x})\|_2 \cos\theta$$

$$= \min_{\underline{u}} \cos\theta$$

- Note $\underline{u}^T\underline{u} = 1 \rightarrow$ basis vectors
- using dot product cosine rule

- minimised when $\underline{u}$ points in opposite direction of gradient i.e.

$\cos\theta = 180° = \pi$

- gradient points uphill; negative gradient points downhill
- decrease f by moving in direction of negative gradient
- gradient descent / steepest descent:

$$\underline{x}' = \underline{x} - \epsilon \nabla_{\underline{x}} f(\underline{x})$$

ϵ - learning rate

- ϵ selected in different ways
  - Fix constant
  - Evaluate $f(\underline{x} - \epsilon \nabla_{\underline{x}} f(\underline{x}))$ for several values of ϵ, select one with smallest obj fn value $\longrightarrow$ line search

---

(A3): Dimensionality checks

- introduction of transpose seems like a convention
- Recall $\nabla_{\underline{y}_N} Div$ is the gradient of the per instance loss/divergence with respect to the post output activation vector from the N^{th} layer, $\underline{y}_N$.
- $\nabla_{\underline{z}_N} Div = \nabla_{\underline{y}_N} Div \cdot J_{\underline{y}_N}(\underline{z}_N)$
- $\nabla_{\underline{z}_N} Div$ is the gradient of Div wrt pre-activation affine combo vector from N^{th} layer, $\underline{z}_N$.
- Note gradient of a scalar fn wrt vector is itself a vector

(*) In vector formulation;

i) Each backward step past an activation

i.e. $\nabla_{\underline{z}_N} Div = \nabla_{\underline{y}_N} Div \cdot J_{\underline{y}_N}(\underline{z}_N)$

involves post multiplication by a Jacobian

ii) Each backward step past an affine combo

i.e. $\nabla_{\underline{y}_{N-1}} = \nabla_{\underline{z}_N} Div \cdot J_{\underline{y}_{N-1}}(\underline{z}_N) = \underbrace{\nabla_{\underline{y}_N} Div \cdot J_{\underline{y}_N}(\underline{z}_N)}_{\text{from earlier (cached)}} \cdot \underbrace{\underline{W}_N}_{\text{yields a weights matrix } \underline{W}_N}$

involves post multiplication by a Jacobian (which for an affine function)

$\cdot \nabla_{\underline{D}R} Div = \nabla_{\underline{z}R} Div \cdot "\nabla_{\underline{D}R} \underline{z}R" = \nabla_{\underline{z}R} Div \cdot \underline{1} = \nabla_{\underline{z}R} Div$

(Q): dimensionality → revisit

(*) on comparison of perceptron and backprop in terms of bias-variance

- OK with the intuitive sense in which the perceptron algo is more 'sensitive' in its estimate of $\hat{w}$ to new training examples.
- AND the sense in which backpropagation for a linear classifier is not as 'sensitive' to addition of new training examples.
- But I am unsure as to how to formulate this formally
- N intuitively variance is related to error from small sensitivity to fluctuations in training

→ (Q) - revisit

(A8): claims regarding backprop generally ⟹ see paper 'backprop fails to sep...' - Brady et al. (1989)

(*) Structure of the error surface (A9)

- lots of excellent papers
- At this stage, jury's out

(A10) - Pedagogical

(*) convergence rate → analysis definition of convergence.

(*) definition of R is OK

(*) But explanation of linear convergence, and exponentially fast is a little unclear.

(*) There are some more rigorous definitions in Nocedal & Wright (1999) and Tibshirani's 10-725 notes

(*) Quadratic surfaces

- (*) Yes; Taylor series can be seen as encoding all of derivatives of a function f about the point at which it is being expanded i.e. at ~~$x = x_0$~~ x_0.

(*) (A11) - $E'(w^{(k)}) + E''(w^{(k)})(w - w^{(k)}) = 0$

$\Rightarrow E'(w^{(k)}) + wE''(w^{(k)}) - w^{(k)}E''(w^{(k)}) = 0$

$\Rightarrow w = \dfrac{w^{(k)}E''(w^{(k)}) - E'(w^{(k)})}{E''(w^{(k)})}$ → much clearer

$\Rightarrow w_{min} = w^{(k)} - \dfrac{E'(w^{(k)})}{E''(w^{(k)})}$ as required

(*) Key step is to compare

i) Analytic optimisation of w^* via Taylor approx of a quadratic error about an arbitrary w_k; calculus

ii) Gradient descent.

(i) owns in this controlled setting of quadratic error

(*) Compare: gradient descent $w^{(k+1)} = w^{(k)} - \eta \dfrac{dE(w^{(k)})}{dw}$

analytic: $w_{min} = w^{(k)} - \dfrac{E'(w^{(k)})}{E''(w^{(k)})}$

(*) Notationally, $\dfrac{dE(w^{(k)})}{dw} = E'(w^{(k)})$

(*) Hence: setting $\eta_{opt.} = \dfrac{1}{E''(w^{(k)})}$ in gradient descent algo

(*) Essentially showing how learning rate could be set analytically and associated intuition.

(?) - I can't see formal reasons why:-

$\eta < \eta_{opt} \Rightarrow$ monotonic convergence

$\eta_{opt} < \eta < 2\eta_{opt} \Rightarrow$ oscillating convergence

$\eta > 2\eta_{opt} \Rightarrow$ divergence.

- see DL book to do with rel. of gradient and Hessian I suspect

(Q) - add to queries

(*) Generic diff. convex objectives:-

$$E \approx E(w^{(k)}) + (w - w^{(k)}) \frac{dE(w^{(k)})}{dw} + \frac{1}{2}(w - w^{(k)})^2 \frac{d^2E(w^{(k)})}{dw^2} + \dots$$

(*) Taylor expand Error $E(w)$ about $w^{(k)}$

(*) using same logic; set

$$\eta_{opt} = \left(\frac{d^2E(w^{(k)})}{dw^2} \right)^{-1}$$

(Q) - relation between Taylor series and Newton's method → refresh

Multivariate $E(\underline{w})$

$$E(\underline{w}) = \frac{1}{2}\underline{w}^T \underline{A} \underline{w} + \underline{w}^T \underline{b} + c$$

- For $\underline{w} \in \mathbb{R}^2$ i.e $\underline{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, diagonal $\underline{A}$

- Plot contours $z = E(w_1, w_2)$

- (*) - point is that you can decouple $E(\underline{w})$ into 2 independent quadratics in w_1 and w_2 (i.e. components of $\underline{w}$); each with their own coefficients.

- can use earlier expressions η_{opt} to find a learning rate for each quadratic

(Q) Is η actually a vector? in this case?

- currently assuming that η is a scalar; but wouldn't issue BR is noting; i.e learning rate is same for all components of $\frac{dE(w_i^{(k)})}{dw}$ or $\eta \nabla_{\underline{w}} E^T$ just an artefact of his notation.

- not is what prevents us from using:-

$$\underline{w}^{(k+1)} \longleftarrow \underline{w}^{(k)} - \eta \otimes \nabla_{\underline{w}} E^T \quad ?$$

(W): Terrible clarity would be gained by better ind. directional deriv.

(*) For now note that for MV $E(\underline{w})$ with decoupled quadratics; we have $\eta_{1,opt}$ and $\eta_{2,opt}$ and that it will not in general be the case that $\eta_{1,opt} = \eta_{2,opt}$; while η is scalar → issues

---

(*) Note the above is heuristic illustration and that conveg. beh. gets unpredictable

---

(*) Formal results on convergence of gradient descent in language of convex optimisation; in part:-

i) quadratic strongly convex fns.

ii) lipschitz-smooth convex fns.

(*) Axis rescaling

- (*) Formally; may be a form of weight normalisation (see paper by Salimans, Kingma (2016))

(*) BR shows this heuristically;

(*) essentially, amounts to rescaling weights $\hat{\underline{w}} = S\underline{w}$

- Rewriting $E(\underline{w})$ as $E(\hat{\underline{w}})$; solving for diagonal scaling mat S; relating $\nabla_{\underline{w}} E$ to $\nabla_{\hat{\underline{w}}} E$. (OIS)

(Q) revisit this → involved calcul. (not involved, but I need to move on)

(OT):- key eq. in notes

(*) Generic differentiable multivariate convex fns

$$E(\underline{w}) \approx E(\underline{w}^{(k)}) + \nabla_{\underline{w}} E(\underline{w}^{(k)})(\underline{w} - \underline{w}^{(k)}) + \frac{1}{2}(\underline{w} - \underline{w}^{(k)})^T H_E(\underline{w}^{(k)})(\underline{w} - \underline{w}^{(k)}) + \dots$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \eta H_E(\underline{w}^{(k)})^{-1} \nabla_{\underline{w}} E(\underline{w}^{(k)})^T \longrightarrow (Q) \text{ -OIS}$$

(*) ILO with Q.A. — (015)

(?) - confused by distinction between applying normalisation → O/S.
and use of Hessian

uses

DL - 1st order opt - gradient
2nd order opt - Hessian
and Nocedal & Wright (1999)

(*) some issues regarding Hessian:

i) computationally infeasible → matrix inves.

ii) Non convex. fn ⇒ Hessian not PSD or PD : divergence.

(*) cluster of methods aimed at approximating (i) → i.e. approx Hessian.

(*) learning rate/step size η

- previous discussion ⟶ η < 2ηopt to prevent divergence for convex fns
(key)

(*) Looking forward, there are 2 main methods other than fixing learning rate η:

i) use a decay schedule

ii) Adaptive methods

(*) Decay schedules for learning rates often used with validation set

- escaping local minima vs convergence

(?) - unclear whether the sufficient conditions specified in this lecture for convergence hold over which assumptns.

- Results directly lifted from SGD Ch 8.3.1. DL.

(*) In any case, covered in L6 in more detail → defer.