11-785

06/05/2020

(15) - convergence

BR: continue with training NN → optimisation

- L4.5: key parts
- BR: goes over lecture 4.5 key parts again (due to lack of clarity)
- NN require training to rep. function (given architecture)
  - ↳ to minimise loss on training set
    - ↳ via gradient descent
      - ↳ backpropagation $\frac{\partial E_n}{\partial w_{ij}}$

- Gradient descent algo
- to minimise $f(x)$ wrt x

(*) initial guess
(*) At each guess, find point where function is decreasing w: (A1) ✓
If derivative is positive → take a step in -ve direction (??)
negative → take a step in +ve dir.

Neural nets:

$$\text{Training error} = \frac{1}{T}\sum_t Div(\underline{Y}_t, \underline{d}_t; \underline{W}_1, \underline{W}_2, \ldots, \underline{W}_K)$$

L4.5: vector formulation w (A2): need to review or not proceed until reviewed. ✓

⇒ Trivial form of forward propagation

Backward propagation

BR: Gradient descent update rule

$$\underline{x}^{k+1} = \underline{x}^k - \eta \nabla_{\underline{x}} f^T$$

(*) This dimensionality check is a code-sanity check

w (A3): check

dimensionality

$y = f(\underline{x})$

scaler ↑ vector or mat

$\nabla_{\underline{x}} y = \underline{x}^T$

$\nabla_{y_N} Div = \nabla_Y Div$

- Div = $\nabla_{y_N} Div$ 

y_N

- Gradient (div.) of divergence wrt output y_N

Div

- Introducti

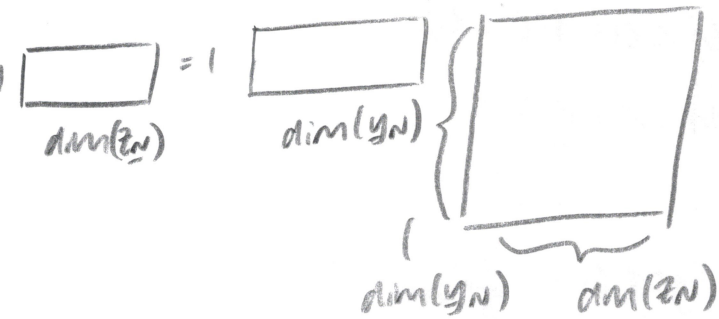
- Then take a step backward

- Compute gradient/deriv of Div wrt affine vector $\underline{z}_N$

$$\nabla_{\underline{z}_N} Div = \nabla_{\underline{y}_N} Div \cdot \underline{J}_{\underline{y}_N}(\underline{z}_N)$$

(*) Take previously compute $\nabla_{\underline{y}_N} Div$, postmultiply by Jacobian

$\underline{J}_{\underline{y}_N}(\underline{z}_N) \in \mathbb{R}^{dim(\underline{y}_N) \times dim(\underline{z}_N)}$



(*) dimensionality illus.

- $\nabla_{\underline{y}_{N-1}} Div = \nabla_{\underline{y}_N} Div \cdot \underline{J}_{\underline{y}_N}(\underline{z}_N) \cdot \underline{W}_N$

(*) Take previously computed $\nabla_{\underline{z}_N}(Div)$ and postmultiply by weights matrix $\underline{W}_N$ (W)(14) :- check

Hence

$$\nabla_{\underline{y}_{N-1}} Div = 1 \quad \underline{\qquad} \quad dim(\underline{y}_N)$$

(*) Each step back is one matrix multiplic.

$\underline{W}_N \in \mathbb{R}^{dim(\underline{z}_N) \times dim(\underline{y}_N)}$

- Each backward step :-

i) Past an activation

→ postmultiply by Jacobian

ii) Past an ....

→ postmultiply by weights matrix (*) Review + check

- In between, compute :- (of most mt.)

$$\nabla_{\underline{W}_R} Div = \underline{y}_{R-1} \nabla_{\underline{z}_R} Div$$

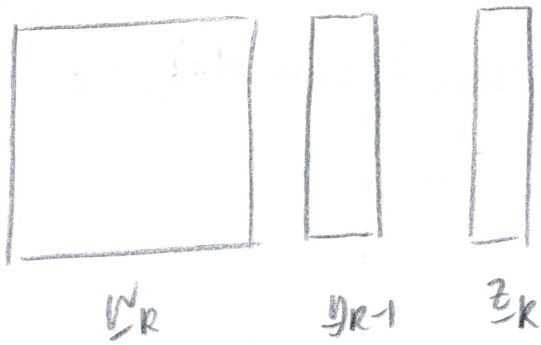
$$\nabla_{\underline{b}_R} Div = \nabla_{\underline{z}_R} Div$$

BR: $dim(\underline{b}_R) = dim(\underline{z}_R)$ as $\underline{z}_R = \underline{W}_R \underline{y}_{R-1} + \underline{b}_R$ ✓

but how does that $\Rightarrow \nabla_{\underline{b}_R} Div = \nabla_{\underline{z}_R} Div$ (??) W(15) : → review

- vector $\underline{y}_{k-1}$
- matrix $\underline{W}_k$

(?)



$\longrightarrow$ scalar $Div()$ from $\underline{z}_k$

$$\nabla_{\underline{W}_k} Div() = \underline{y}_{k-1} \nabla_{\underline{z}_k} Div()$$

- $\nabla_{\underline{z}_k} Div()$ already computed. (row vector $\in \mathbb{R}^{1 \times dim(\underline{z}_k)}$)
- $\underline{y}_k$ - column vector $\in \mathbb{R}^{dim(\underline{y}_k) \times 1}$
- Hence $\nabla_{\underline{W}_k} Div = \underline{y}_{k-1} \nabla_{\underline{z}_k} Div() \in \mathbb{R}^{dim(\underline{y}_k) \times dim(\underline{z}_k)}$

$\underline{W}_k \in \mathbb{R}^{dim(\underline{z}_k) \times dim(\underline{y}_k)}$

- Recursive formulation. (*) then this

(*) Entire backprop pseudocode:- (BETTER VERSION)

(*) supplement/summarise/check notes/scan

Init: $\underline{y}_N = \underline{Y}$ $\underline{y}_0 = \underline{x}$ $\nabla_{\underline{y}_N} Div = \nabla_{\underline{Y}} Div$

For $k = N, \ldots, 1$:- $\longrightarrow$ compute $J_{\underline{y}_k}(\underline{z}_k)$

$\nabla_{\underline{z}_k} Div = \nabla_{\underline{y}_k} Div \, \underline{J}_{\underline{y}_k}(\underline{z}_k)$ $\quad \nabla_{\underline{y}_{k-1}} Div = \nabla_{\underline{z}_k} Div \, \underline{W}_k$ (recursion)

and $\nabla_{\underline{W}_k} Div = \underline{y}_{k-1} \nabla_{\underline{z}_k} Div$ (gradient comp.)

$\nabla_{\underline{b}_k} Div = \nabla_{\underline{z}_k} Div$

(*) - entire NN training algo $\longrightarrow$ check your understood logic.

- KL divergence rather than L_2 divergence $\rightarrow$ faster optim.

convergence: - will we get to location
- How long will it take
} → focus

- Network of MLP with thresholds → not poss. to train
∴ Replace with diff. activation; replace error 'count' with divergence
- Allows differentiability, parameter tuning
- Does this work?
- Questions about convergence of grad.

Q: does backprop yield desirable results?

- 2 issues

1) changed definition of error
- loss is no longer error (in percptr. sense); but average of divergences
- see slides
- minimising loss does not necessarily mean minimising classification error

(*) - Quiz: missed slides (W) (A6)

---

backprop fails to sep...

(*) red line is global minimum; where loss for $\frac{dw}{dw}$ has minimum

(*) Add a furthr training instance; will perceptron find linear class.

(*) consider add. of one more point and influence on loss for

i) perceptron

ii) backpropagation

(*) uncomfortable with hand-wavy exposition here → (?) source?

(*) see slides

(*) Consider sensitivity of estimated decision boundary to add. training data.

- Perceptron: low bias, high variance
- Backprop: low variance
} → (W): an Immense overly simplified presentation of bias-variance?

- not a bug, it's a feature

- for MLPs rather than single perceptron → same issue.

(W) ~~Backp~~ BR: Backprop will often not find a separating solution even though the soln. IS within the class of functions learnable by the network.

(*) As separating soln. is not a feasible optimum for the loss function

- Backprop trained NN class. has a lower variance than an optimal classifier for training data

(W) (A8): - review how this claim is formalised

- true error/loss surface (assumed)
- find minimum point of this surface : single global optimum
- what does loss surface actually look like?

BR: - open question → contentious.

- Saddle point → see diagram

(*) Surface increases in some dirs, increases in others.

(SM) - some Hessian eigenvalues +ve; some -ve

Notably: ~~Bol~~ see literature (W) (A9):

Story so far:

- not guaranteed to find 'true soln' (??) (W) (A10)
- why/how have gone from assuming global minima → local minima

---

- convergence → local minima
  1. converge?
  2. How long?

(*) Hard to analyse for MLP with complex loss fn (surface?)

- use/invoke convex functions (best-case for quadratic loss)

BR: informal, intuitive def of convex: connecting line intersects ✓
convex function/set ✓

(*) see contour plots of surface for illustration of convergence

i) convergence q(iv): asymptotically → ?

ii) jittering

iii) diverging

---

convergence rate: (theoretical const.)

$$R = \frac{|f(x^{(k+1)}) - f(x^*)|}{|f(x^{(k)}) - f(x^*)|}$$

- $x^{(k+1)}$ - k^{th} iteration
- x^* - optimal value of x

- If R is constant

(**) Review slides, supp here

---

quadratic surfaces (convergence)

- a determines shape of quadratic (and ∴ min or max)
- 2nd derivative also
- computing optimal step size. BR claims it is one step/line via Taylor about x_0

$$f(x) = f(x) + f'(x)(x - x_0) + f''(x)(x - x_0)^2$$

✓ BR: check

(*) First R derivatives of resulting expansion are equivalent to first R derivatives of function at that point

(*) How many terms for Taylor series of quadratic?
0th, 1st, 2nd order term → exact for quadratic

(*) Final Taylor exp. at $w^{(k)}$

$$E(w^{(k)}) + E'(w^{(k)})(w - w^{(k)}) + \frac{1}{2}E''(w^{(k)})(w - w^{(k)})^2$$

minimise wrt w:-

$$\frac{dE(w^{(k)})}{dw} = E'(w^{(k)}) + \frac{2}{2}E''(w^{(k)})(w - w^{(k)}) = 0$$

Hence $E''(w^{(k)})^{-1} = \underline{a}^{-1}$ (?) ⓐ ⓐⅱ - check ✓ (*)

Set $\eta_{opt} = E''(w^{(k)})^{-1}$

- set stepsize to be inverse of 2nd derivative of loss (?)
- get to optimum in 1 step
- new and interesting analysis

(*) consider $\eta < \eta_{opt}$ (monotonic)

$2\eta_{opt} > \eta > \eta_{opt}$ (oscillating convergence)

Ⓠ: Bound on step size before it gets bad?

$\eta > 2\eta_{opt}$ (divergence)

- Above results for quadratic
- For generic diff. convex ~~opti~~ objective functions
- same logic as above:-

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

- functions of multivariate inputs

Obj.

$E = g(\underline{w})$ $\quad \underline{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}$ E is a scalar g() is scalar fn of vector input.

$\underline{A}$ - diagonal

- quadratic convex paraboloid:-

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

$$= \frac{1}{2}[w_1 \cdots w_N] \begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} + [w_1 \cdots w_N] \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix} + c$$

If A diagonal:- - sum of indep. quadratics

$$E = \frac{1}{2}\sum_i a_{ii} w_i^2 + \sum_i w_i b_i + c = \sum_i \left(\frac{1}{2} a_{ii} w_i^2 + w_i b_i \right) + c$$

For 2d: A - diagonal

$E = \left(\frac{1}{2}a_{11}w_1^2 + b_1 w_1 + c_1\right) + \left(\frac{1}{2}a_{22}w_2^2 + b_2 w_2 + c_2\right)$

- This is a bowl in 3 dimensions i.e. $z = E(w_1, w_2)$
- Take level sets / slices of bowl
- concentric ellipses when viewed from top
- what is shape of horizontal slices e.g.
  - Fix w_1 and vary w_2
  - Vary w_1 and fix w_2

(A11): Review argument here.

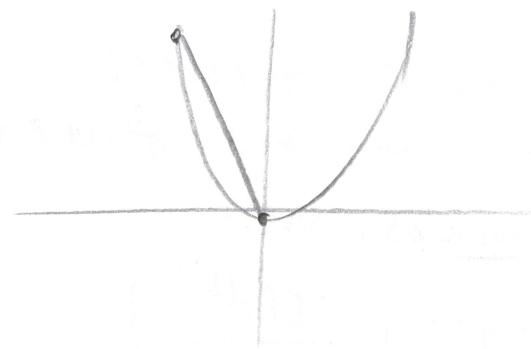
- for each quadratic, check step size prescription

$\eta_{1,opt} = a_{11}^{-1}$ $\eta_{2opt} = a_{22}^{-1}$

---

vector update rule (W)(A12) review

- gradient orthogonal to level set
- Apply to quadratics you have decoupled



---

problem with vector update rule

$$\underline{w}^{(k+1)} \leftarrow \underline{w}^{(k)} - \eta \nabla_{\underline{w}} E$$

$$w_i^{(k+1)} = w_i^{(k)} - \eta \frac{dE(w_i^{(k)})}{dw}$$

- optimal step-sizes for each of independently uncoupled quadratics (W)(A13)

intuition

(*) - conflict between optimal size solns. makes learning slow.

and $\eta_{i,opt} = \left(\frac{d^2E(w_i^{(k)})}{dw_i^2}\right)^{-1} = a_{ii}^{-1}$

- $\eta < 2 \min_i \eta_{i,opt}$ $\eta_{i,opt} < \eta < 2\eta_{i,opt}$

(A14): review the panels in light of above presentation.

- Dependence on learning rate

(X) convergence behaviour becomes increasingly unpredictable as dimensions increase.

(X) Fastest convergence; based on intuition with quadratic surfaces, learning rate η must be close to both largest $\eta_{i,opt}$ and smallest $\eta_{i,opt}$.

- to ensure convergence in every direction/dimension
- BR: generally infeasible

(X) BR: generally slow if $\frac{\max_i \eta_{i,opt}}{\min_i \eta_{i,opt}}$ large

- Gradient descent - step size η is fixed for every direction

---

- (X) (Q): Do you understand the formulation of the problem?
- (S): Rescale axes in response to different optimal learning rates for different directions.

- scale w_1 and w_2 axes $\hat{\underline{w}} = \underline{S}\underline{w}$ $\underline{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ $\underline{S} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix}$ $\hat{w} = \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \end{bmatrix}$

(X) (15) ↘ check

Yielding:- $E = \frac{1}{2}\hat{\underline{w}}^T \hat{\underline{H}} \hat{\underline{w}} + \hat{\underline{b}}^T \underline{w} + c$ (as $\underline{A}$ is (2nd? 'derivative' one in every direction)

optimal step size $\eta = 1$ (check; as 'derivative' one in every direction)

- How to find $\underline{S}$?

(*) Review these derivations

(X) solve for scaling matrix $\underline{S}$ by inspection.

(IV) solve quadratic (?) in scaled space yielding:-

$$\hat{\underline{w}}^{(k+1)} = \underline{w}^{(k)} - \eta \nabla_{\hat{\underline{w}}} E(\hat{\underline{w}}^{(k)})^T \quad \text{(make subst.)}$$

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \eta \underline{A}^{-1} \nabla_{\underline{w}} E(\underline{w}^{(k)})^T$$

- gets you to optimum in single step if $\eta = 1$

(**) Generic differentiable multivariate convex fns:-

- Taylor exp:- check dimensionality (*) $\underline{w}^{(k+1)} = \underline{w}^{(k)} - \eta H_E(\underline{w}^{(k)})^{-1} \nabla_{\underline{w}} E(\underline{w}^{(k)})^T$
- (Q): What justification for using only 1st two terms of Taylor expansion for generic multivariates (quadratic approx.)
- Notice role of Hessian here (derivative of scalar fn wrt 2nd vector input)

(*) Heated localised optim with quad. approx.

(Q) Why $\eta = 1$ by Newton's method.

Issues - Hessian - Normalising by Hessian

(A) 100,000 parameters → difficult to invert / compute

(B) Non convex functions; Hessian may not be positive semi-definite due to convexity?
(?) (surely positive definite)

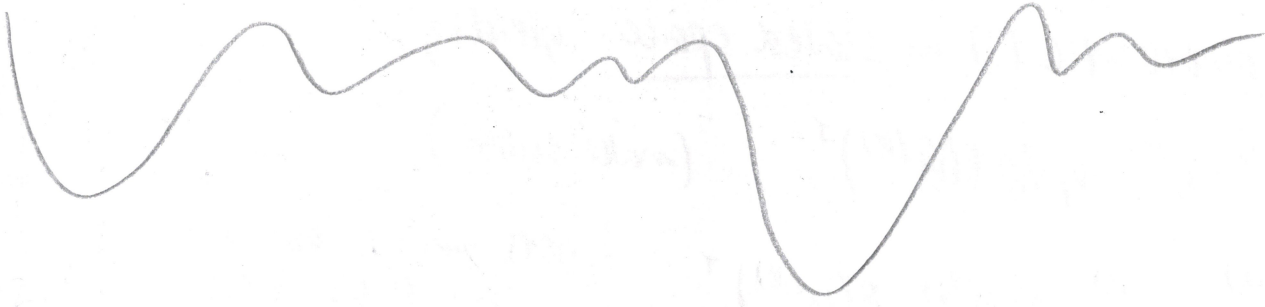
- Algorithm diverges - away from optimum
- Eigenvalues → what are the consequences?

(*) Quantitative techniques for approximating Hessian, improve positive-definiteness

BR: Earlier analysis was focused on ensuring step size was not so large as to cause divergence within a convex region $\eta < 2/\lambda_{max}$

(*) But for non-convex functions, is divergence a bad thing?

(611)



(*) $\eta > 2\eta_{opt}$ may help escape local optima; but always having $\eta > 2\eta_{opt}$ will ensure that we never actually find a sol.

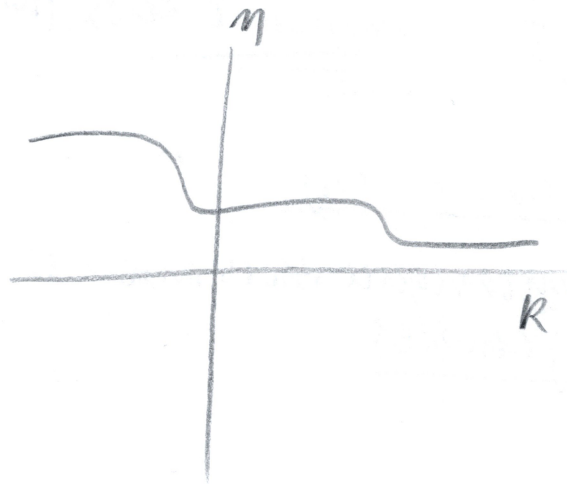
(W): Adaptive step sizes?

- decaying learning rate - post-normalisation $\eta > 2$ (reduce with iterations)

decay schedules

common approach - see slides

linear $\eta_R = \frac{\eta_0}{R+1}$



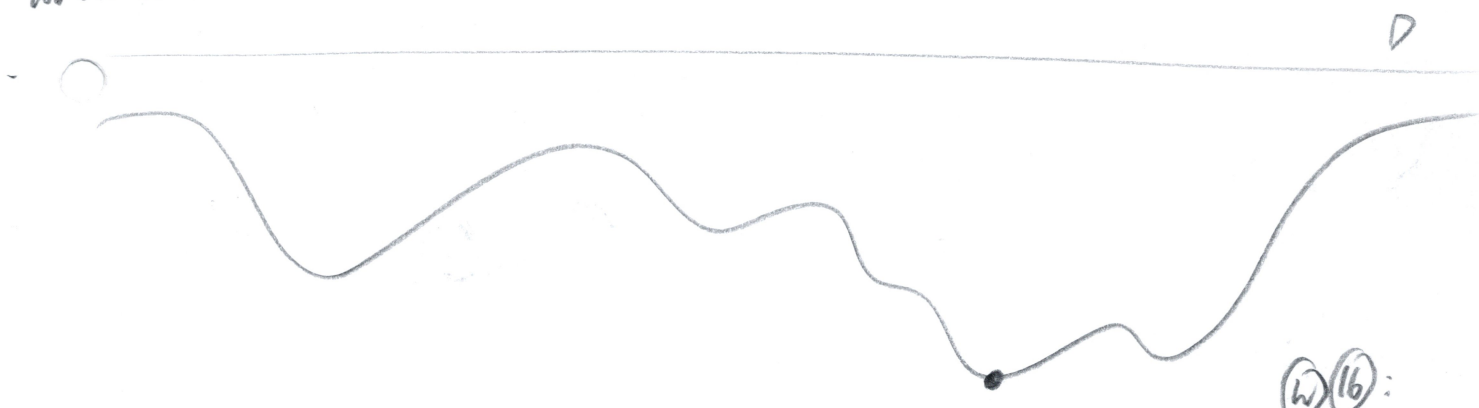
quadratic $\eta_R = \frac{\eta_0}{(R+1)^2}$

exponential $\eta_R = \eta_0 e^{-\beta R}$ $\beta > 0$

- Note in (S11) that function derivatives are bounded

$f'(x) < B$

- what criterion to ensure I can find global minimum?



(W)(16):

- require:-

$$\sum_{R=1}^{K} \eta_R |f'(x)| > D$$

(?)

$$\sum_{R=1}^{K} \eta_R = \infty$$

- make sure step sizes sum to infinity

- otherwise if start in wrong place, never able to reach optimum.
- want stepsize to also shrink a little → another criterion.

- $\sum_{R=1}^{K} \eta_R^2 < C$ (i.e. sum of squared step sizes is bounded).

(*) These 2 criteria allow exploration of entire space; converge to solution.

- candidate function - $\frac{1}{R}$ - boundary where sum of step sizes is infinite
  sum of square step sizes is finite

$\sum_{R=0}^{\infty} \frac{1}{R}$ - harmonic series (wiener)

(W)(*) Review this thinking

(*) Review summary

(SGD) many convergence issues arise because we force learning rate on all parameters