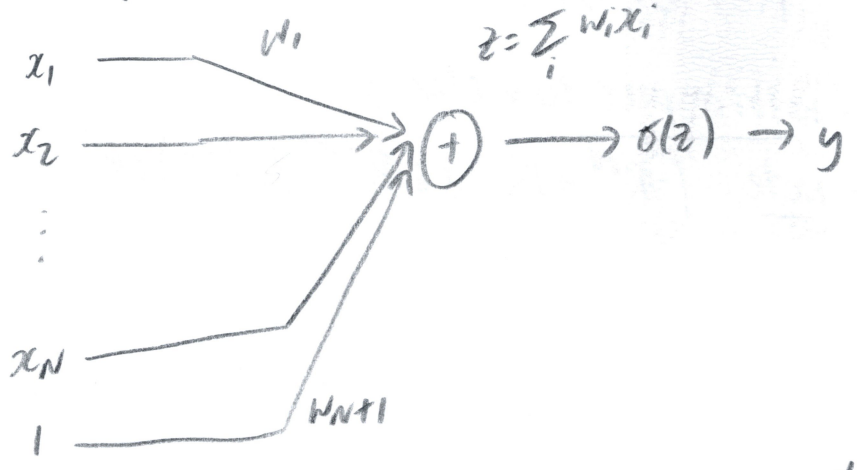
11-785 09/05/2020

L3 - Training NN (review)

(*) scalar specification of MLP with diff activations. (sigmoid)



✓

- $\frac{dy}{dz} = \sigma'(z)$ $\frac{dy}{dw_i} = \frac{dy}{dz}\cdot\frac{dz}{dw_i} = \sigma'(z)x_i$ $\frac{dy}{dx_i} = \frac{dy}{dz}\cdot\frac{dz}{dx_i} = \sigma'(z)w_i$

- okay → happy with this

(*) A little uncomfortable with the way the distinction between expected and empirical error, and the graphical presentation.

- Whilst the presentation (BR) is consistent within presentation, trying to figure out discomfort

- In particular, the way in which the probabilistic interpretation is offered.

- The general presentation of generalisation error, training error in (expected) (empirical)

Bishop, Hastie, Salakhutdinov:-

- Is that expectation of error tends to be taken over a joint distribution of inputs $\underline{x}$ and target outputs t

i.e. $\mathbb{E}_{\underline{x},t\sim p(\underline{x},t)}[L] = \iint L(t, y(\underline{x}))\, p(\underline{x},t)\, d\underline{x}\, dt$

- AND a distinction made between classification and regression (via choice of loss function)

(*) comfortable with derivatives; review gradient operator

(*) Gradient operator

- for a scalar valued diff. function of several variables

$f: \mathbb{R}^N \mapsto \mathbb{R}$

$f(\underline{x}): \underline{x} \longmapsto f(\underline{x})$

$\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$

$$\nabla_{\underline{x}} f(\underline{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_N} \end{bmatrix}$$

- with respect to vector $\underline{x}$
- Note: $f: \mathbb{R}^N \to \mathbb{R}$ $\underline{x} \in \mathbb{R}^{N \times 1}$ $\nabla_{\underline{x}} f(\underline{x}) \in \mathbb{R}^{N \times 1}$

(*) Note: understand that:-

i) gradient of scalar fn of a vector points in rate of greatest increase of a function

ii) - if it is -ve; rate of fastest decrease of function

} an incomplete spec

(*) see OL - direct. derivatives

(*) (W) (AI) → Formal, mathematical justification for this; it's hand-wavy

- As this is not crucial; but would be helpful; leave it here; find resources → add to O/S list

- Req will require additional machinery (vector calculus) - Di

(*) directional derivative (*) see MIT OCW on multivariable calculus OR "deep learning"

(*) level sets

(*): later → gradient operator and level sets (*) Also clarifies the approx. not. BR uses

(*) Hessian matrix — i.e. for scalar valued functions of several variables

- similar to gradient → matrix of 2nd (setup)

$$\nabla^2 f(x_1, \ldots, x_N) = \nabla^2 f(\underline{x}) = \begin{bmatrix} \frac{\partial f^2}{\partial x_1^2} & \frac{\partial f^2}{\partial x_1 x_2} & \cdots & \frac{\partial f^2}{\partial x_1 x_n} \\ \frac{\partial f^2}{\partial x_2 x_1} & \frac{\partial f^2}{\partial x_2^2} & & \frac{\partial f^2}{\partial x_2 x_n} \\ \vdots & & \ddots & \\ \frac{\partial f^2}{\partial x_N x_1} & & \cdots & \frac{\partial f^2}{\partial x_n^2} \end{bmatrix}$$

(*) can use properties of Hessian matrix i.e. S_n^{++}, S_n^{+}, S_n^{--}, S_n^{-}
to identify local minima/maxima
and values of its eigenvalues (all of them)

- All eigenvalues of matrix $\lambda_i > 0 \ \forall i \Rightarrow S_n^{++}$ (positive def) → minimum
- " " $\lambda_i < 0 \ \forall i \Rightarrow S_n^{--}$ (neg. def) → maximum

---

(*) Deep learning book really helps here. (Goodfellow et al.)

- Ch 4.3 - Gradient-based optimisation
  4.3.1 - Jacobian, Hessians

(*) The relation between expected error and empirical error and also empirical risk minimisation is given in DL (Goodfellow), Ch 8 (unregularised) (empirical estimate of train exp. error)

(8.1)

- $J(\underline{\theta}) = \mathbb{E}_{(\underline{x},y)\sim \hat{p}_{data}} L(f(\underline{x};\underline{\theta}), y)$

L - per example loss fn.
$f(\underline{x};\underline{\theta})$ - predicted output when input is $\underline{x}$
$\hat{p}_{data}$ - empirical distri.

(8.1) - Obj. function over training set

- we also have :-

$$J^*(\underline{\theta}) = \mathbb{E}_{(\underline{x},y)\sim p_{data}} L(f(\underline{x};\underline{\theta}), y)$$

- This is the expected generalisation error or risk, where the expectation is taken over the data generating distribution p_{data} rather than just training set
- we do not know $p_{data}(\underline{x},y)$, so replace with $\hat{p}_{data}(\underline{x},y)$ defined (empirical distri) (true distri)

By training set.

- minimise $\mathbb{E}_{\underline{x},y\sim\hat{p}_{data}(\underline{x},y)} L(f(\underline{x};\underline{\theta}),y) = \frac{1}{m}\sum_{i=1}^{m} L(f(\underline{x}^{(i)};\underline{\theta}), y^{(i)})$

m - no. training examples

(*)