Lecture 22 -- More NNs

Thursday, November 21, 2024 3:32 PM

SGD:

Replace: $\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{n=1}^{N} \frac{\partial L_n}{\partial \theta}$

with: $\frac{\partial \mathcal{L}_S}{\partial \theta} = \sum_{n \in S} \frac{\partial L_n}{\partial \theta}$.

Can choose S randomly at iteration

or split traing data into K subsets each approx size N/K and then systematically step through (like CV)

$K = |S|$ = (mini) batch size

After N/K iterations we've seen entire data – called one epoch.

Upside to SGD: learning via many small steps

Downside to SGD:

- noisy

Downsides of GD/SGD:

- may want different step sizes in different directions
  (adaptive learning rates)

- Can get stuck in local min / saddle points

  (momentum)

Popular approach: ADAM

May also want to regularize the loss we minimize

Ways to do this:

1. penalize the loss:

(1)' penalize the loss:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \; \mathcal{L}(\theta) + \lambda \|\theta\|_2^2$$

(like ridge)

(2) early stopping

i.e. stop SGD when val. perf. stops decreasing

(3) Dropout: during traing we randomly set some weights to zero (temporarily)

→ forces fitting to basically an average model over space of models w/ only some connections

→ making any single connection unreliable and therefore stopping overfitting by fine tuning

overfitting by fine tuning

How do we fit in practice?

<u>Need</u>: $\frac{\partial \mathcal{L}_S}{\partial \theta} = \sum_{n \in S} \frac{\partial L_n}{\partial \theta}$

<u>Model</u>:

$$\hat{f}_\theta(\underline{x}) = \hat{f}_L(\hat{f}_{L-1}(\cdots \hat{f}_2(\hat{f}_1(\underline{x}))))$$

$$L_n = L(y_n, \hat{f}_\theta(\underline{x}_n))$$

<u>Theoretically</u>: can do this by hand

- Calc 3 problem
- can just use chain rule

<u>Problem!</u>
- very tedious and error prone
- also not modular (may want to switch out parts of arch)

parts of arch)

Solution: Backpropagation

(automatic derivative calc. using chain rule)

idea: $f(x) = g(h(x))$

$f'(x) = g'(h(x))\, h'(x)$

$\frac{\partial f}{\partial x} = \frac{\partial g}{\partial h} \frac{\partial h}{\partial x}$

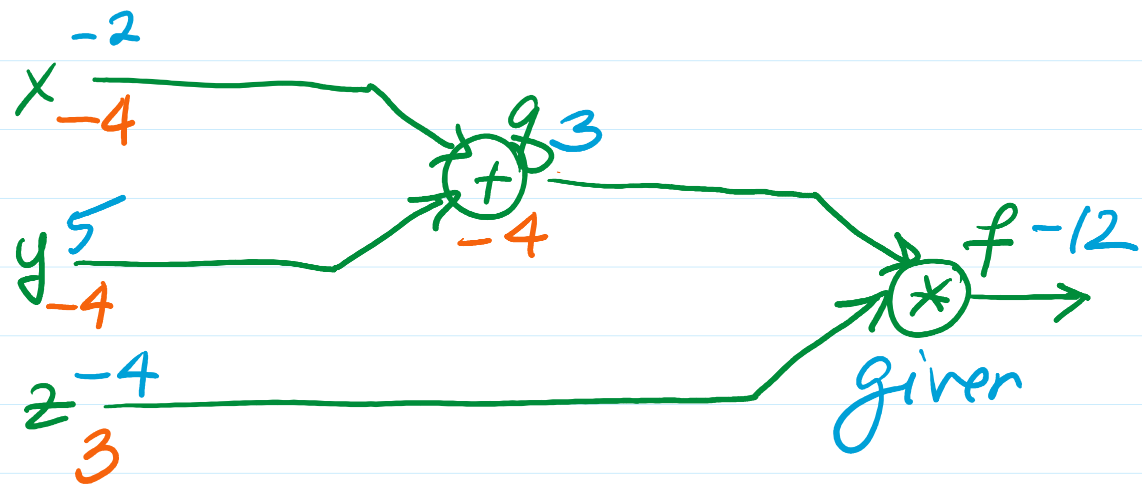
Computational Graphs

$f(x, y, z) = (x+y)z$

input: x, y, z

$$\boxed{\begin{array}{l} (1)\ q = x + y \\ (2)\ f = qz \end{array}}$$

-2



Forward pass: calc f x, y, z

Backward pass: want $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

$$\frac{\partial f}{\partial z} = q = 3$$

$$\frac{\partial f}{\partial q} = z = -4$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y} = (-4)(1) = -4$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x} = (-4)(1) = -4$$

local gradient

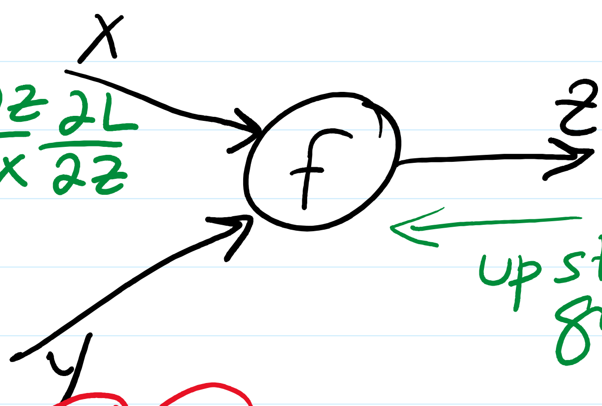
downstream gradient

upstream gradient

downstream gradient   upstream gradient

Zoom in on calculation

$$z = f(x, y)$$



$$\frac{\partial L}{\partial x} = \frac{\partial z}{\partial x}\frac{\partial L}{\partial z}$$

- - -> L

upstream gradient $\frac{\partial L}{\partial z}$

$$\frac{\partial L}{\partial y} = \frac{\partial z}{\partial y}\frac{\partial L}{\partial z}$$

local gradient : $\frac{\partial f}{\partial y} = \frac{\partial z}{\partial y}$

$$z = f(x, y) = xy$$