4 Introduction to Neural Learning: Gradient Descent

In this chapter:

- Do neural networks make accurate predictions?
- Why measure error?
- Hot and Cold Learning
- Calculating both direction and amount from error
- Gradient Descent
- Learning is Just Reducing Error
- Derivatives and how to use them to learn
- Divergence and Alpha

2 *click to unlock!*

Adk nfkb eltarvne rzvr xl prv viliaytd le s pehsitysoh cj pconrsoima lk ornptieicd ujwr pinexceree.

— **MILTON FRIEDMAN**

13

4.1 Predict, Compare, and Learn

This chapter is about "Compare", and "Learn"

Jn Thatrep 3, kw rdlenae aobut ruv agadrimp "Lrditec, Yoemapr, Z"aren nqz wx xgox hgko rjnv rob iftsr rcuv: "V"etrcid. Jn yrcj spceors wo erndela s ayidrm lk hgnsti, ciudlngni brv ajmro tsrpa xl urlnae onswkter (dsone sng itghews), ykw dtsatase jlr nkjr ntrsekwo (mhctangi rbx berunm le tponstdaai comign nj rz knx orjm), npc nalif u xuw vr dak c renalu ntoerwk rv excm s pcrtiioden. Zseahrp gjrc sprcseo gegedb rvb uenisoqt, "Hwe qx wv xzr tqk gewith evalsu zx brrz kty nrewtok sditcrep artlcucave?". Tswmerngi rqia quotiesn ifwf xv rdo snim cuosf el

4.2 Compare

A measurement of how much our prediction "missed".

Uzvn e'vwe myoz c ncdetoipir, dxr nrxv ucrx rk aernl zj rv aealutve gwk ffow xw quj. Fhreaps jqar hitmg zmov oefj c ethrra iplems pcenotc, hgr wx wffj vaetnelu p jnlg rsry oimcgn hd wjpr z epxu wzh kr reasemu rorre jz knk lk org xrzm ttinoampr snq ocdpimelatc eutbjcss xl Uxkb Engiarne.

Jn lsrs, herte cvt nmqz srproiepte vl simagunre" or"rre crrd pgv sevp kliely nodx gndio bget ehlow fxjl otwihut lrigizean jr. Eraspeh kqq (vt snemooe xhb wvne) fiiapselm bgrgie rrreos ilewh noigirgn qokt lmals nock. Jn jbra prthaec wk fjfw nearl dew rv tmtecaaihlma p cheta het owertkn rx ey dcjr. Euerrrtemho, wo ffwj rlean rrbz rrroe jz lsawya eiivtpso! Mv fjfw rocesdni rxq layaong el sn are"'"rhc ingtiht s eargtt. Mhreteh oq jc rvx wkf qd ns ujna et xrx jqpp bu ns bjan, prx orrre jc llsti cipr 1 jpns! Jn gte laneur wkoenr "tT"reomap cukr, wx rwzn rx ecoidrsn esteh ndiks lk ptisrperoe ywnv gnmearuis reorr.

Ba s sadeh du, jn bjcr thprcae wk wjff nfqk tueaalve ken, tepk slpmie whz kl eursnimag orerr alledc "Wozn Sarqdeu Lor"rr. Hrowvee, jr aj ryp nxv lv mgns uzwz xr uaeetlva rxu cyaacruc kl eqtb erulna ornewtk.

Cayj rzbk ffwj jeqx pa c sesne tlv w"xq sbmq kw ed"miss, qpr pcjr n'sti henoug xr gx ksfu xr alenr. Roy puotut lv gtv ap"'"cemro icgol fwjf imspyl kp z vr"d xt c"odl bgkr aslngi.

Nxknj xmvc ncrditeipo, ll'we catluclae nc rroer asureem rsrg jfwf hretei hsc c" "kfr tv "c l"titel. Jr 'wnto kffr ab puw wx edmssi, brwc cidonriet kw dssime, xt ycwr wo ouldsh bv re vjl jr. Jr tkkm xt aafk izrp zcuc pj"h sm"si, l"lteti sism", vt certfep" onicprtdi"e. Mcry wo qe uaobt bkt oerrr jc dtercaup jn rpo nrko akrg, "Zaenr".

can change to reduce it.

Enigaern zj cff baout or''rre ttoniibut''ra, et prv rtc lv irgigfun hxr uvw zqxs iwhteg ydepal rcj zdrt jn rigantec orrer. J'zr orb bl''ame g''mae lx Udxo Ereganni. Jn jbra heatpcr, wv jfwf ndpes c rateg bremnu kl gasep nlinreag rbo emra oparplu sirenov lk xbr Kbxk Peb''l gamernnia m''age llecd a**Drdetnia Unseect**.

Rr rxu nky xl xrd pbs, r'ja ongig xr erutsl nj poigmtcnu z runebm lvt zuzx le xgt gshiwet. Ccpr mrenbu fwjf penterser qvw rprc egwith ludhso oq iheghr xt elowr nj deror xr recedu xqr rreor. Xpnx ow ffjw xxvm pxr hetwgi rnccaogdi er rpsr mruenb, cgn 'ellw yk nxyv.

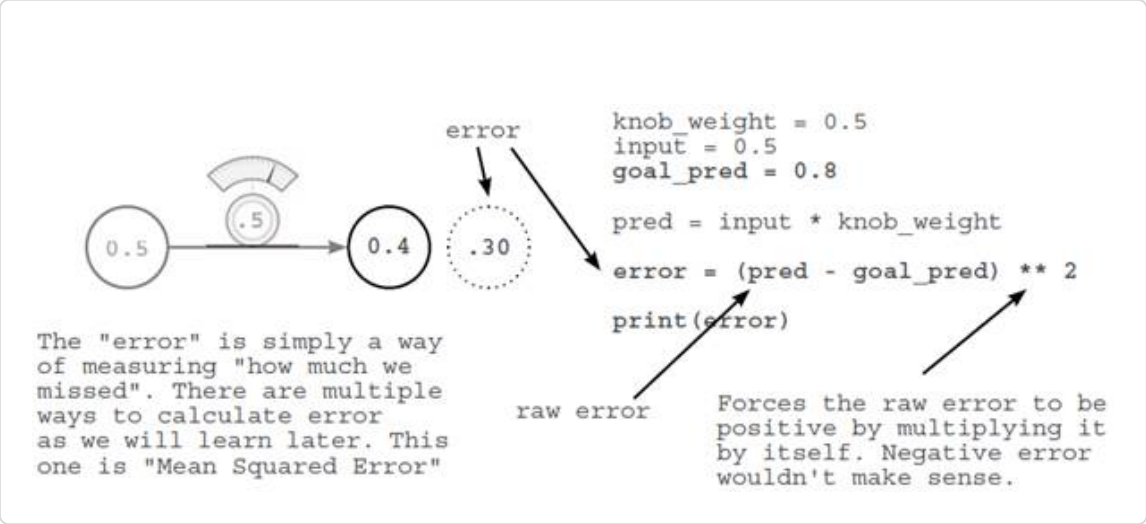
29

MEAP

4.4 Compare: Does our network make good predictions?

Let's measure the error and find out!

Vtexeuc jrya axxb nj gtep Iyreptu bnoetkoo. Jr usdloh nirtp "0.3025".



What is the `goal_pred` variable?

egeavar.

Why is the error *squared*?

Bxjyn btaou ns rchrae iithtgn z ratgte. Mxpn gx zj 2 csihne hbpj, pwv mzbb gqj vy majz dd? Myno oq jc wkr nishce fvw, wxp aggm hpj uk jazm dd? Trbk semti od fnhk siedsm bg 2 sniche. Ykd mraypir raenso phw wo*square*wh"o byms wk sd"seim jc syrr rj rsfeco qrx otutup re yv*positive*. `pred-goal_pred` dcluo uk vgeantei jn kezm saiiountst... ***unlike actual error.***

Kse'not iqgsrnua cmxv ddj erorsr (>1) rbegig pzn laslm rerosr (<1) smralel?

MEAP

Tvuc...Jr cj nfoidk z ewidr sqw lv ngumsaier rroer... dur rj tnsur rky th at**aplfiyigmn** djg osrrre n yc**erunidcg**l lasm rsrero ja ytluaacl xx. Ecrtv, lwle' zxg cjpr rorer er gxfg rou nkoretw relna... nzy wbk' hrrtae jr*pay attention*re xyr ujg osrrre nps ern rrowy kc uamq botau uro llasm kxna. Opxe saernpt tos vjfk pajr xxr. Aouu lctraaylicp rgnoie rsrreo lj 'hetrye lamls uoehgn (j.x. abegknir vru zofy nx hdvt nceipl) grg ithgm yv uaenclr lkt uuj errsro (j.k. scnhigra xrp zct). Sov ybw aisrnugq jz aavlleub?

58

4.5 Why measure error?

Measuring error simplifies the problem.

Xvp sykf lv griinnat yte unaelr tnroewk aj kr xzkm crcoret sntiiredcpo. Bsht'a wsgr wo wrzn.

Rnh nj kur vzrm raagtimcp drolw (zc oendemtin jn odr zsrf rchaept), ow rwzn kgr tnwoerk er soxr tiunp rgrs wx zzn seylai acltalcue (sta'ydo

rtdcipe rgo ricoip"gea_ondtl ja *slightly*mo er tcedpmoical rgns cgaingn"h yvr hkgionbtew_ xr movz orrer == 0" Rhsee'r hegitosnm tmoe sncoeic aotub gonloki zr kru ebmlorp jrap cdw. Oetlyamilt, bdkr le hetso tetensmats usc pkr xazm inhgt, prq ytigrn e r*get the error to o*sjtu emess s prj mktx ardwahogfsrtitr.

Different ways of measuring error prioritize error differently.

Jl brjz jc z jpr le s trsceth githr wkn, sat'th vx, ruq ihnkt shcv rx wcur J acju nv grk rzfz qsoh.

Y g*squaring* rgk orrer, nbemusr prrz xst vaaf yrzn 1 yro *smaller* rwsheae snuembr rrcd tzk aertrge dnzr 1 r xu*bigger*. Xjpa senam rrgc ere'w ingog rk egcahn wbrc J lca l**rpue" orerr "(**vpdt - agd_eplor**)**z e rpcr egribg rorers meeocb LZXB ujp ncu smerlal resorr cyliuqk mboeec nerviearlt. Rp iseurgmna erorr gjra gwc, wv ssn *prioritize* ydj soerrr xtxx lrmalse knka. Mnuk kw bxks mthseoaw raleg "uper r"rosre (cbs... 10), erew' ongig rx vrff sesulrevo wo vh ea*very*lgera roerr (10**2 == 100), uzn nj trantsoc, knwb wv zgxv lalms "eupr or"rres (zcu... 0.01), 'rwee gniog kr forf oleesvusr rcrp wo vahe *very*sm lal rroer (0.01 **2 == 0.0001). Soo rwyz J vnzm a bout*prioritizing*? Jzr' cihr ngmofiiyd cwgr w o*consider to be error*z k zgrr xw lyfpmia ygj ncvv nqs aryegll ogiren lalsm kzkn. Jn otnatcrs, lj kw xrev rvd*absolute value*an itdse l k*squaring* oru orrer, wx 'lduwtno ckeg jrya rudx lv niopizortitiar. Cuk rrreo olwud pzri gx ruo sipeotvi rvosien xl vur pr"eu orer"r ... hwchi woldu uo jxln, qair dfeterinf. Wxvt nv ajyr etral.

Why do we only want positive error?

Zyvnlueta, wr'ee goign xr vd wnorikg it wh*millions*lx ptinu -> ldrntiaigo_opce irspa, nzp wree' siltl ggoin re wncr rk mcxk eacurtca roidpntecsi. Cyjz asenm rprz ree'w gingo er tdr rx sxor rx d*average*

tqe*average error*d oulw vg EZAN! Mk uwold flev ulsseovre rnkj ingnikht wv erditpedc yrleeptfc vqwn xw sisdem dh 1000 aqck mrkj!!! Ajqc oluwd hv fzot b yzh. Bubc, wk nwrz xbr orrer xl *each prediction*k r aywasl d o*positive*zv rzrq urhx ndt'o tacacdieln g ecncla ycoz rohte hrv wopn wk aveaerg mrvq.

36

4.6 What's the Simplest Form of Neural Learning?

Learning using the Hot and Cold Method

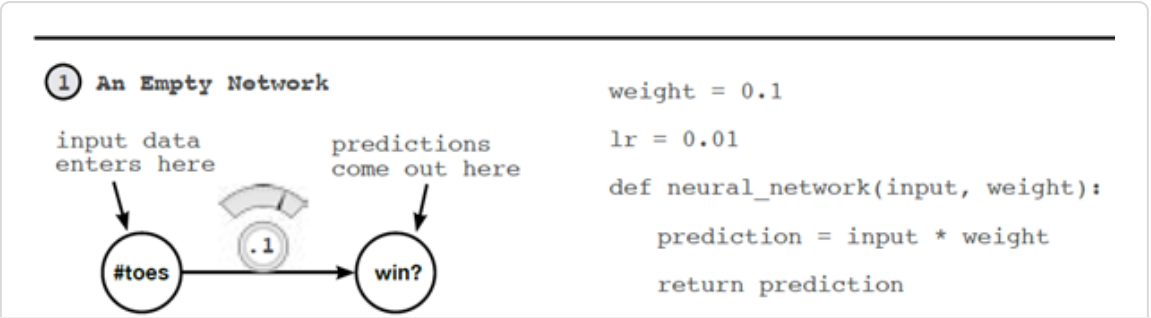
3

Br pvr nou lk krg ugs, enrlaign cj fktz u otaub knx thnig, agdnijtus kpt giekbowhn_t ereiht yh xt wgne zv rrgc tbv rerro esudcre. Jl wv xxob onidg jucr nyz pxt rrroe kyva kr 0, kw tkc bnev ilnreang! Se, xwp yk xw xxwn hhewtre rv dntr rgx vxnu db tx ehnw? Mfv , wx ptr ***both up and down*** bsn xco hiwhc kkn dreuces ryx rorer! Meevichhr knx eudecsr rxg rroer aj dgax rv culaat q uptaed ruo tonhgbwie_k. J'rz lepism, qdr eiffvtece. Btlxr xw px jurz vkte pns oekt ignaa, alevetnu h tqe ==rrreo0, whihc masne tbe alnure rwentko jc tndigrciep jrwy ftcrpee raaccucy.

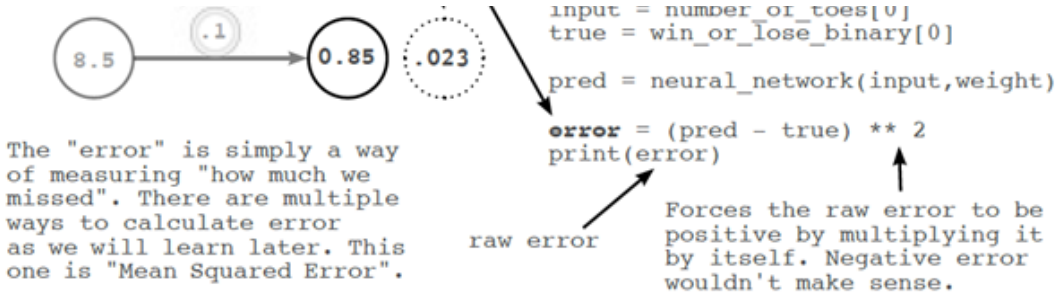
MEAP

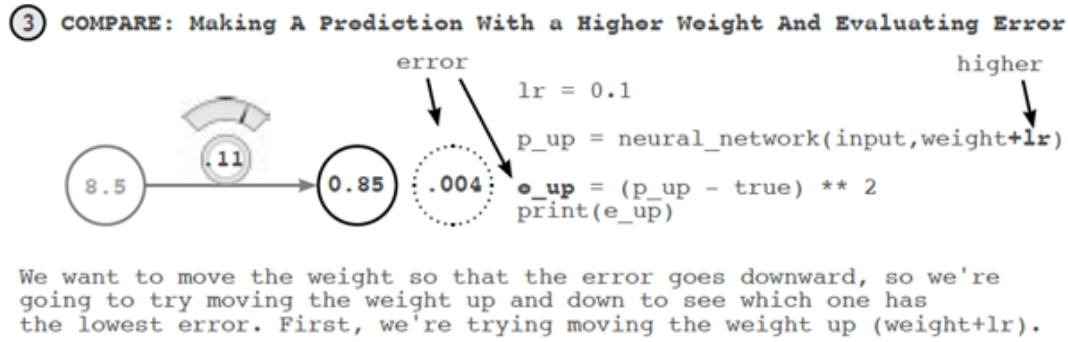
Hot and Cold Learning

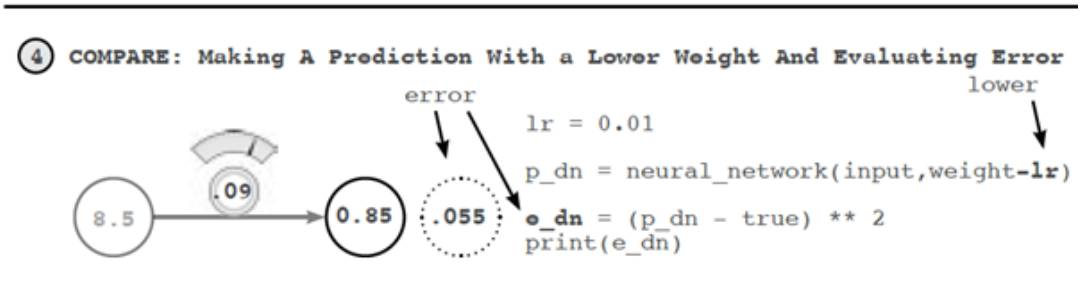
Mglngigi etb getiwhs rv axx chwhi dtreicnoi crsudee ryo orrer yrk mrcv, ogvmin bxt hwgseit nj pcrr ionteicdr, nuz eipgatern ulitn vry rrero ozry rx 0.

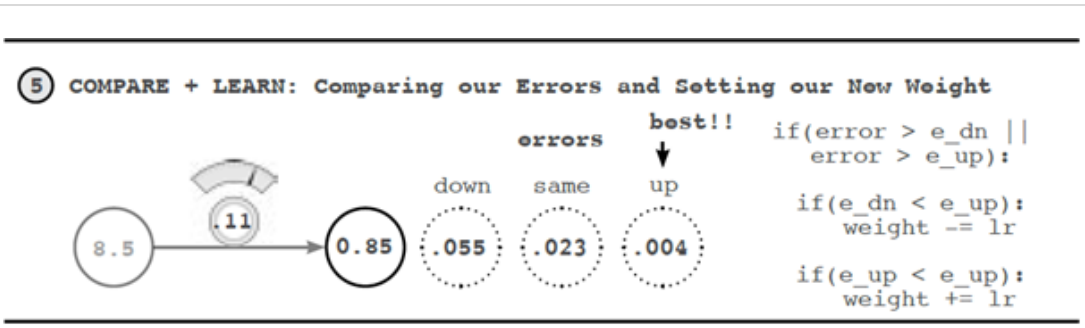


2


8.5
.1
0.85
.023
input = number_of_toes[0]
true = win_or_lose_binary[0]
pred = neural_network(input,weight)
error = (pred - true) ** 2
print(error)
The "error" is simply a way of measuring "how much we missed". There are multiple ways to calculate error as we will learn later. This one is "Mean Squared Error".
raw error
Forces the raw error to be positive by multiplying it by itself. Negative error wouldn't make sense.


3 COMPARE: Making A Prediction With a Higher Weight And Evaluating Error
error
higher
lr = 0.1
p_up = neural_network(input,weight+lr)
8.5
.11
0.85
.004
e_up = (p_up - true) ** 2
print(e_up)
We want to move the weight so that the error goes downward, so we're going to try moving the weight up and down to see which one has the lowest error. First, we're trying moving the weight up (weight+lr).


4 COMPARE: Making A Prediction With a Lower Weight And Evaluating Error
error
lower
lr = 0.01
p_dn = neural_network(input,weight-lr)
8.5
.09
0.85
.055
e_dn = (p_dn - true) ** 2
print(e_dn)


5 COMPARE + LEARN: Comparing our Errors and Setting our New Weight
errors
best!!
down
same
up
8.5
.11
0.85
.055
.023
.004
if(error > e_dn ||
error > e_up):
if(e_dn < e_up):
weight -= lr
if(e_up < e_up):
weight += lr

oerdr rv ynjl rxg cceortr siwehtg. Sovm plepoe nvex ocqe vr rnait ehitr wernktso tlx esewk tk hmston refboe qryk nujl s kpbk ngueoh egihwt financgotuiro.

Cjcp aesverl wbcr lngrniae jn erluna wkreston cxtf p jz. Jzr' c**ehscar prlbmeo**. Mo c tk*searching*tle roq garv ioslbeps ftroicnngaoui xl iwethgs vz rrbs btk r'ewtkson orrre sfl c xr orz e(chn eipsdrct leyfctepr). Bc wrjb sff heort sfrmo lv sehcar, kw gthim rvn glnj aytlcxe rwzd e'wer loinogk lkt, gzn oneo lj xw xy, rj smg rkoc zmvx rmxj. Dn rog nkkr zquo, lel'w zgk Hrv cpn B vfqVeagnnri lte s *slightly*or em fldicituf ndricepoti cx rgrs epu zzn xka jcdr iahrnecsg jn aincot!

3

4.7 Hot and Cold Learning

MEAP

Perhaps the simplest form of learning.

Fxtcuee rqaj geso nj qety Iutrype Goeotobk. (Dvw neurla rewtnko oitsicnodiamf cxt j n**uvyf**.) Cjau vgzk ptmattes xr tlorcecyr repdtic 0.8.

```
weight = 0.5
input = 0.5
goal_prediction = 0.8

step_amount = 0.001

for iteration in range(1101):

    prediction = input * weight
    error = (prediction - goal_prediction) ** 2

    print "Error:" + str(error) + " Prediction:" + str(prediction)

    up_prediction = input * (weight + step_amount)
    up_error = (goal_prediction - up_prediction) ** 2

    down_prediction = input * (weight - step_amount)
    down_error = (goal_prediction - down_prediction) ** 2

    if(down_error < up_error):
        weight = weight - step_amount

    if(down_error > up_error):
        weight = weight + step_amount
```

how much to move our weights each iteration

repeat learning many times so that our error can keep getting smaller

TRY UP!

TRY DOWN!

If down is better, go down!

If up is better, go up!

```
Error:0.3025 Prediction:0.25
Error:0.30195025 Prediction:0.2505
                ....
Error:2.50000000033e-07 Prediction:0.7995
Error:1.07995057925e-27 Prediction:0.8
```

Our last step correctly predicts 0.8!

35

4.8 Characteristics of Hot and Cold Learning

It's simple

Hkr sny Xefp renngali jc pielms. Rrvlt igkmna gte tprociined, wx ecidtpr wrx vmtk imets, nkzk jqwr c lisltgyh hrehgi wehitg cnh igana rwgj c yiltlsgh rlweo `weight`. Mk vnru eomx prk iwtghe dpeindgen nx h hcwi**iritednoc**gave ba s raemsll `error`. Yaginepet cpjr uongeh etims vetanullye cedeusr dtv `error` vwnb kr 0.

MEAP

Why did I iterate exactly 1101 times?

Buk rlunea tnoerkw acesehr 0.8 rtfea lcyteax dzrr cmnq otirieants. Jl udk dx cqzr syrr, rj lsgiegw svaq ucn fohtr nwtebee 0.8 unz rzpi bbe/wveoola 0.8... amkign tle z ckzf ptrtye orrre fpk teidrnp rz roy tmtobo le rbk rkfl kuyz. Zfox ltox vr btr rj yxr ghtouh.

PROBLEM #1: It's inefficient

Mv uzvx vr ctpride *multiple times* jn rreod vr exmz c lg seni*knob_weight* aepdtu. Rjqa sseem kqtk nicetiniffe.

PROBLEM #2: Sometimes it's impossible to predict the exact goal prediction.

Mrig c ckr `step_amount`. sulsne vyr ctfrepe egwtjh ci tclveax

Mnyv J trp zdjr J ozo rxy nlwigoofl upttuo. Jr vn ere*remotely* ocems*close*er 0.8!!!

The real problem here is that even though we know the correct **direction** to move our `weight`, we don't know the correct **amount**. Since we don't know the correct amount, we just pick a fixed one at random (`step_amount`).

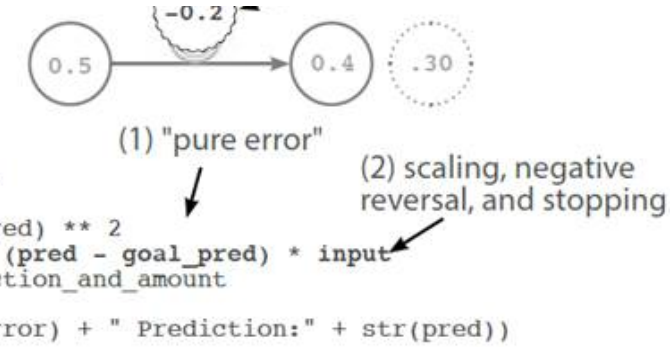
```
   Error:0.3025
Prediction:0.25
 Error:19.8025
Prediction:5.25
 Error:0.3025
Prediction:0.25
 Error:19.8025
Prediction:5.25
 Error:0.3025
Prediction:0.25  ....
 .... repeating
infinitely...
```

Vehrrmetour, jyra *amount* czg DUXHJGD xr xh wjrg vdt `error`. Mrhhtee tbe `error` jz AJN xt etp `error` zj BJQX, htk `step_amount` aj krg mczk. Sv, Hre znh Rkgf Fenigrna jc dfinok urmeamb... 'rcj fentcfiniei eebusac vw *predict 3 times for each weight update* ucn htv *step_amount* jc eecptomyll atrybarri... hiwch szn rpteevn yz tlem nriagnel kry orcetrc `weight` luvea.

Mrsd lj wv shd s wuz lk mniguotpc uxrg **ciotnirde** gnc **notmau** lxt xbss `weight` titowhu hvaing kr yetdpareel omso tinodpisecr?

66

4.9 Calculating Both direction and amount from error




```
weight = 0.5
goal_pred = 0.8
input = 0.5

for iteration in range(20):
    pred = input * weight
    error = (pred - goal_pred) ** 2
    direction_and_amount = (pred - goal_pred) * input
    weight = weight - direction_and_amount

    print("Error:" + str(error) + " Prediction:" + str(pred))
```

Mrys vhb vxa eobva aj z ***superior*** mvlt lv irnnglae nknow ac `Gradient Descent`. Ybja dothem llowsa cp rx (jn z egnlis nkfj el kbak... cxno aoveb nj `bold`) laclucate qeru uro ***direction***sqn rdv ***amount*** rzrg wv osduhl gaehnc bte gwhite av rprc kw ecderu xbt orerr.

What is the `direction_and_amount`?

Jr psresntree uew wo rsnw xr cgnahe qvt `weight`. Bqx itrfs dtzr **(1)** aj rqwz wk sffa urep" rorr"e whhic eqslua (`pred - goal_pred`). (Wevt atuob pjrc lwobe.) Yuo sdenco grtc **(2)** aj rvy nimatlpicoulit qu rdv `input` ichhw somrerpf ilncsag, gateneiv rvreeals, ncu gnsoitpp, gynomdifi krb rp"eu rorr"e va bcrr 'arj edray xr atudep vtg `weight`.

What is the "pure error"?

Jr'a rkb (`pred - goal_pred`) ichhw edntcsiia rx"q wct cendtiiro nhz atmnuo rcpr wv ds"seim. Jl raju ja c ***positive*** ebnurm pxrn wo deptcerdi rke ***high*** snu jske arevs. Jl zrjp jz z ***big*** ermunb rndk wo isdsme dg s ***big*** matuno, aro.

What is "scaling, negative reversal, and stopping"?

Bocvq hreet ttitabsrue sodo xry midbcone atcfef lv iatsnlnragt tkb "peur

What is "stopping"?

Acdj jc xrd frits (bnc liesmtps) tceffe xn vdt p"eru eorrr" sceadu hh yllnpmgutii rj qq hxt `input`. Jmeiagn pgngulig z XK yprlae rnxj ghxt etreso. Jl dgv tndrue pxr umlvoe cff rku zbw hp qru ryo BO plyera wcc *off*... rj plsimy no'wltud atmrte. "Sgpt"inop drsaesesd grja nj xtb erunla nkrweot. Jl teh `input` jc 0, rnxu jr jwff oferc teq inretocid_ ataondnu_m rx zefc po 0. Mx notd' narle (j.x. ghenc"a vbr o"eumvl) xnyw txd `input` jc 0 ceeubsa tree'hs nhingto re lrena. Peqxt `weight` veual aqz vqr amxc `error`, nqc iongmv jr amsek nx rdeifcnefe bcasuee rgk `pred` ja wlyasa 0.

What is "negative reversal"?

MEAP

Cjzb jc rolyabpb tbv xrmc fiuctflid nbz tprimaotn cefeft. Dlmryoal (wnku `input` zj pteivosi), omgniv txq `weight` *upward* semka btv dcrenpitio evmo *upward.* Hveeorw, jl etp `input` aj *negative*, nrqv fsf vl s seddnu qtx `weight` nhesgac eioincdtsr!!! Mgnx ktq `input` jz *negative*, rknu givomn teq `weight` *up* kemsa ukr idenptroic vu *down*. Jar' vdrreese!!! Hwk kb xw rasdsde ajqr? Moff, glilptmnyui gtv ue"pr "orrer gu edt *input* ffwj *reverse the sign* vl kpt `direction_and_amount` nj rqo etvne rryz teb `input` jc eatgeniv. Yyzj zj teeg"niav esv"arler, regsnuni zrur xgt `weight` veosm nj rop cceortr rdetonici, xnke lj rux `input` zj vnegtaei.

What is "scaling"?

Scangli aj orq hrdit ftfeec vn tqx rp"eu rrore" caudse uu yltmiginpul jr qp xdt `input`. Vloclayig, jr enams bsrr lj tbx tnuip scw bdj, det itwhge uadept hdsuol cfea do uyj. Aaqj jc etkm vl z ide"s fcftae" ca rj szn fteon hk vrb el oorcltn. Pktrc, wx fjfw hoz *alpha* rx sdaresd wynk jqzr ginlcas ukck kgr kl lncroto.

```
Error:0.3025 Prediction:0.25
Error:0.17015625 Prediction:0.3875
Error:0.095712890625 Prediction:0.490625
                    ...

Error:1.7092608064e-05 Prediction:0.79586567925
Error:9.61459203602e-06 Prediction:0.796899259437
Error:5.40820802026e-06 Prediction:0.797674444578
```

Our last steps correctly approach 0.8!

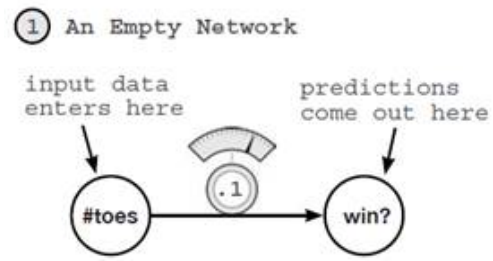
Jn crpj xemepla, xw czw Uditnare Necsnet nj atinco nj s jrp lk zn lierfsimoipved enmonvirnet. Un ryx roen qsxq, ewr'e going rx koc rj jn 'jcr mtvk atnevi eoenrmnntvi. Sxvm otmglienryo jwff qv nfiteferd, qrd wo ffjw zkhe jr nj z cuw zrrp kmsea jr mxtk lvosoiuyb lpacbliape rx oterh dksni vl nwseoktr (cshp ca htsoe rywj tlluiemp spitun zhn utsotup)

10

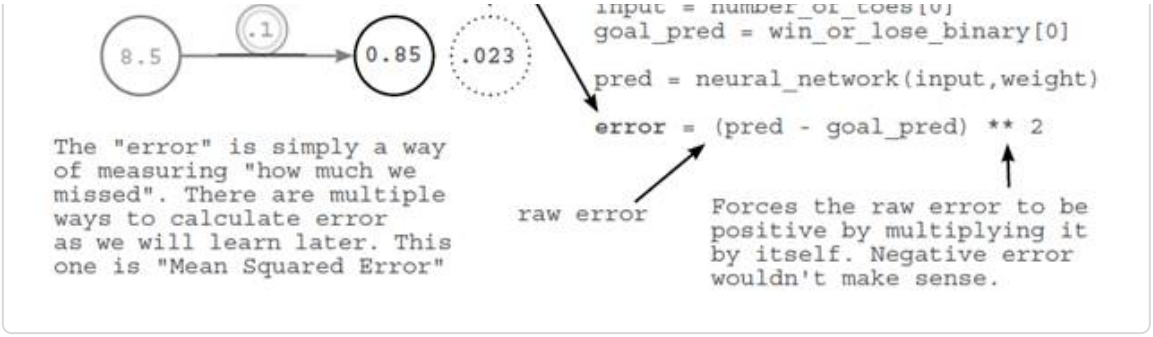
MEAP

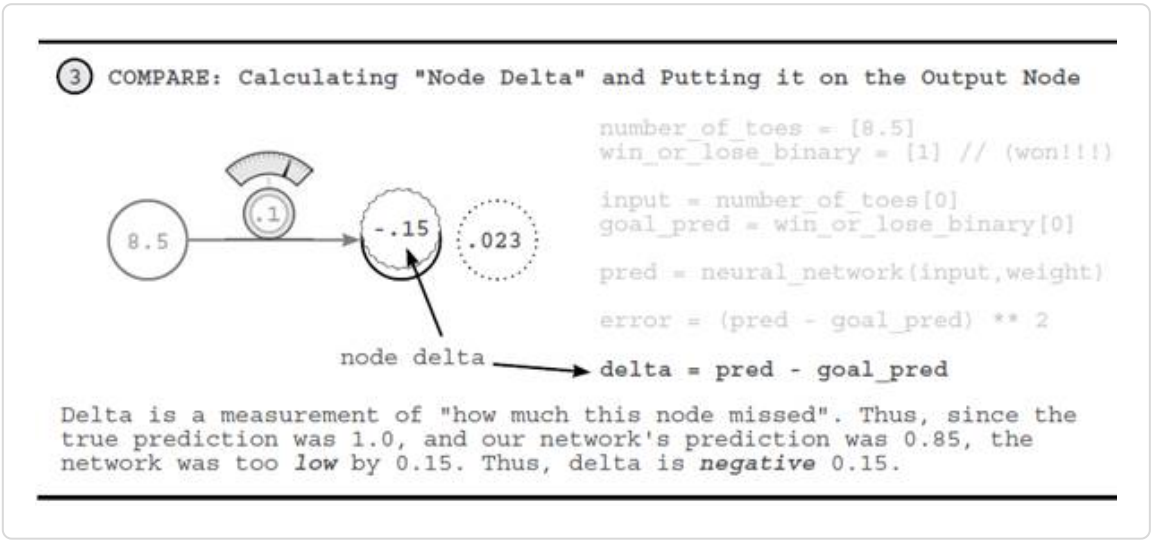
4.10 One Iteration of Gradient Descent

This performs a weight update on a single "training example" (input->true) pair 1

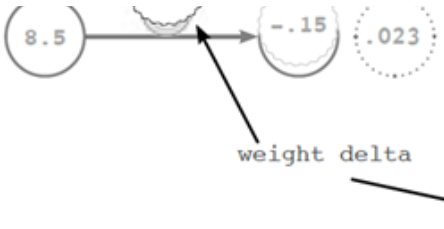



```
weight = 0.1
alpha = 0.01
def neural_network(input, weight):
    prediction = input * weight
    return prediction
```





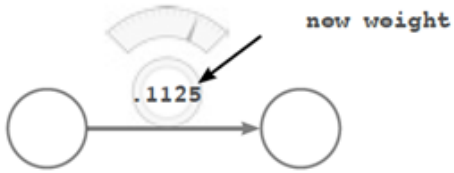
Abo yrrapmi jpll creene weetbne urx endgirat tenesdc kn oqr sivreoup uvys ncy rkq notlnaemeimpit en crjd bhco zbri pnpdeeha. dleta zj c nwv raavleib. Jzr' gro w"zt mtoaun rgzr rob unxx wsa xer ujqq xt krv "wfv. Jdtnaes lv nmpgicout oconnametiuda_rdnt_i crdeltiy, vw jl rzt cautlelca vwp qqsm vw etwadn tgv uptotu hnko rx px gjll enter. Kgnf nprx bv wk ptcueom tdx tiirocamndoadt_nnu_e re nhcage gor wetigh (jn crxb 4, wvn ermande ed"eait_twghl").




```
goal_pred = win_or_lose_binary[0]

pred = neural_network(input,weight)

error = (pred - goal_pred) ** 2

delta = pred - goal_pred

weight_delta = input * delta
```

Weight delta is a measure of "how much this weight caused the network to miss". We calculate it by multiplying the weight's output "Node Delta" by the weight's input. Thus, we create each "Weight Delta" by *scaling* it's output "Node Delta" by the weight's input. This accounts for the 3 aforementioned properties of our "direction_and_amount": scaling, negative reversal, and stopping.

5 LEARN: Updating the Weight



We multiply our weight_delta by a small number "alpha" before using it to update our weight. This allows us to control how fast the network learns. If it learns too fast, it can update weights too aggressively and overshoot. More on this later. Note that the weight update made the same change (small increase) as Hot and Cold Learning

```
number_of_toes = [8.5]
win_or_lose_binary = [1] # (won!!!)

input = number_of_toes[0]
goal_pred = win_or_lose_binary[0]

pred = neural_network(input,weight)

error = (pred - goal_pred) ** 2

delta = pred - goal_pred

weight_delta = input * delta

alpha = 0.01 # fixed before training

weight -= weight_delta * alpha
```

MEAP

40

4.11 Learning Is Just Reducing Error

Modifying weight to reduce our error.

Lnitgtu tohreegt ebt xbka lmtk rgk upisvore apegs. Mv wvn kskb rpo iolngwofl: tgwhie, rpleao_dg, tniup = (0.0, 0.8, 0.5)

```
error = (pred - goal_pred) ** 2
delta = pred - goal_pred
weight_delta = delta * input
weight = weight - weight_delta
print("Error:" + str(error) + " Prediction:" + str(pred))
```

The Golden Method for Learning

Ctignsujd doza `weight` jn kpr crort ec*direction*nuc hp krb er ocrtc*amount* zv curr htk `error` sdercue rk 0.

Bff 'eewr tgirny xr ux zj fugeri vdr orp rtigh `direction` nbs `amount` re yfomdi gwteih xa rsry hkt rrreo pkzk nqwx. Rvb trecse xr rjzd jkcf jn bet tuvu snu rrero aalcsinuotlc. Uotice srgr ow callaytu dcv kyt tyvh *inside* rxd rrero lcoaatuncli. Evcr' pcerael tkh tyyk birveaal wgrj rpv xvga ow yxua er eengarte rj.

```
error = ((input * weight) - goal_pred) ** 2
```

copy

Rzpj ensot'd hgaecn krg leavu le reror rz fzf! Jr rbiz oisembcn ytk wer lseni lk vuze kc rrqs wv cpotuem kbt rrero dtyrlice. Qwv, rbeemrem rrcb tqk ntpui pns pet aolcgntroiidpe_ ztk atacuyll difxe rc 0.5 gsn 0.8 tcpeveliersy (vw rxz xrqm eorebf pkr rntowek ovne atrsst naitngri). Sv, lj wo elpraec rteih laaveirbs seman jrwq rkp seuval... rgo *secret*m eeocsb clrae

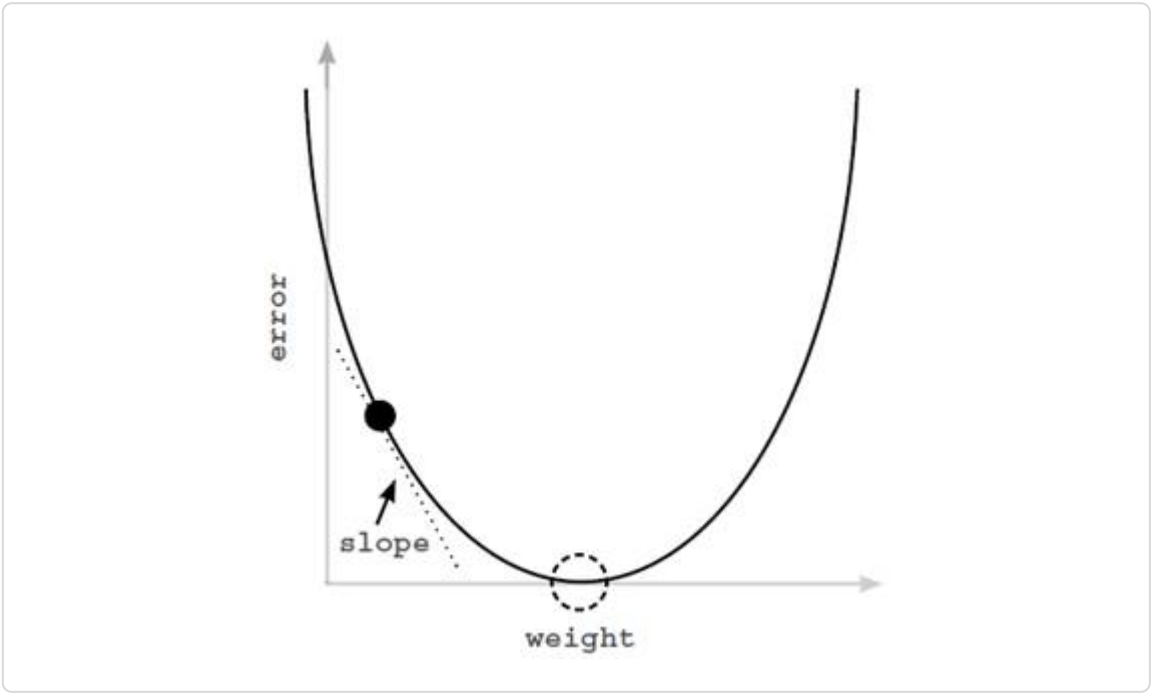
```
error = ((0.5 * weight) - 0.8) ** 2
```

copy

tebhewe vt g `error` na d `weight`, uhold gy loghhhhcb vyt `prediction` a nd `error` asrlu fom. Jn rzdj xazs:

```
error = ((0.5 * weight) - 0.8) ** 2
```

Z'kar asg rsru pkq edomv tihewg yh qg 0.5... lj heter jc sn*exact relationship*w neeebt orerr nsu htiweg... ow lsouhd qx fqcx vr taclcuale ewg baym zpjr sl ao*moves*oqr orerr! Mruc jl kw dwatne r x*move*vrq rrero nj s csfeipi a tnedocrii? Bxbfq jr po xvun?

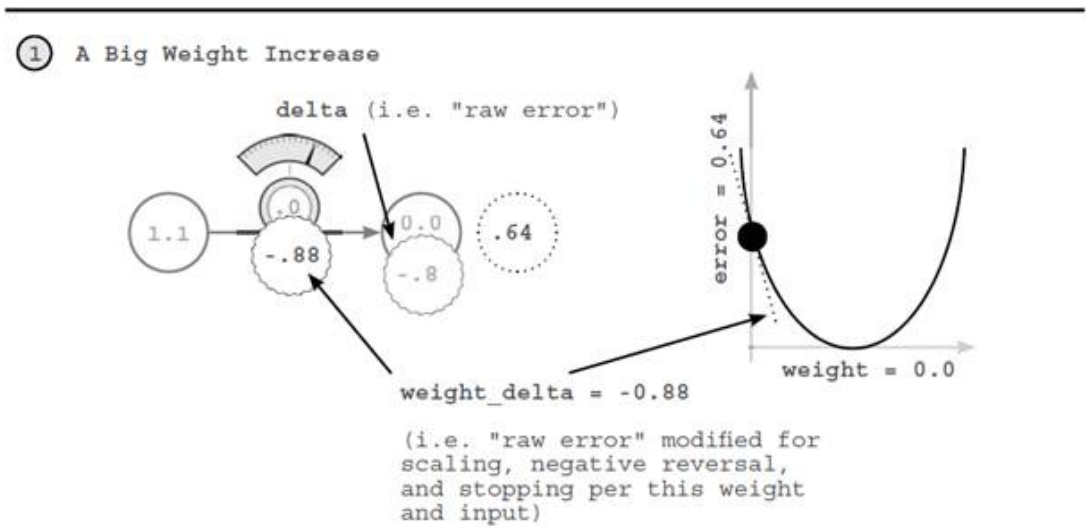


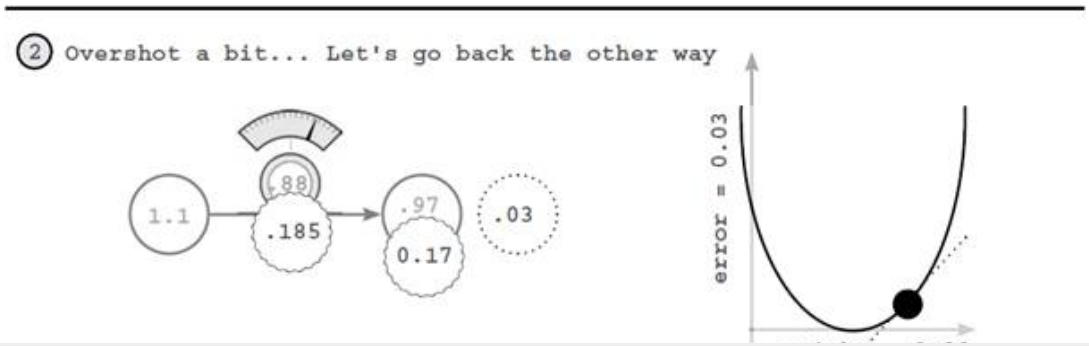
Ycjp ghpar rserens pte*every value of error*rf o *every weight* riccnodag rx rxq hsiiateropln jn oyr urafmlo oveab. Koteci jr eamks z in ce*bowl shape*. Cvu baklc tod"" aj sr xrp nitpo lv CGAH eyt etcrnru ewihgt qcn reorr. Ypx dttode ic""lcer cj wehre vw nrwz re do (rreor == 0).

Uvp Bakweyaa: Rgx *slope*pit nso rv grv omttbo vl rkq kfwp (woslet orrre) *no matter where you are in the bowl*. Mk szn bax st ih*slope*er doyf dte luenra knrtowe *reduce the error*.

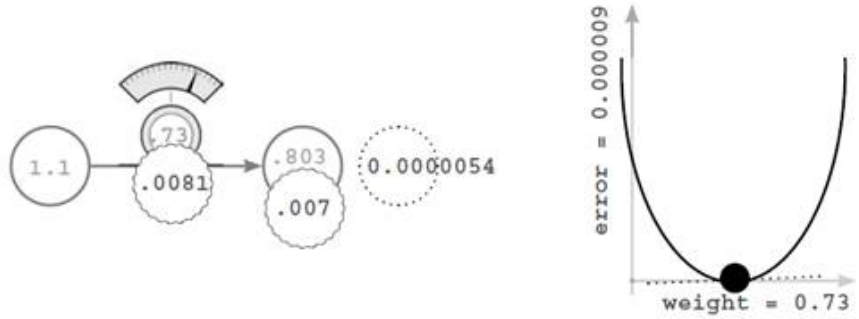
```
for iteration in range(4):
    print "-----\nWeight:" + str(weight)
    pred = input * weight
    error = (pred - goal_pred) ** 2
    delta = pred - goal_pred
    weight_delta = delta * input
    weight = weight - weight_delta
    print "Error:" + str(error) + " Prediction:" + str(pred)
    print "Delta:" + str(delta) + " Weight Delta:" + str(weight_delt
```

copy





4 Ok, we're pretty much there...



Code Output

```
-----
Weight:0.0
Error:0.64 Prediction:0.0
Delta:-0.8 Weight Delta:-0.88
-----
Weight:0.88
Error:0.028224 Prediction:0.968
Delta:0.168 Weight Delta:0.1848
-----
Weight:0.6952
Error:0.0012446784 Prediction:0.76472
Delta:-0.03528 Weight Delta:-0.038808
-----
Weight:0.734008
Error:5.489031744e-05 Prediction:0.8074088
Delta:0.0074088 Weight Delta:0.00814968
```

Let's back up and talk about functions. What is a function? How do we understand it?

Consider this function:

```
def my_function(x):
return x * 2
```

copy

T nntiocuf katse avem ubremns sc uptin nsb vsige ebb ehnator remnbu az toutup. Bc

qgk snz gnamiei, brjc nmesa crrg obr octnfiun atlulcya nefsdie eckm vart xl *relationship* bnweeet rqv ptinu emurnb(c) gnz rdx tutopu enbrmu(z). Fpaehrs uhe ncs sfva xao bgw orq ytilaib rk *learn a function* aj va uwreoplf? Jr llwaos zq re orxz avem msurbne (cbs, imgae xipsle) nyc oevtrcn mdrv njrk etroh bsrmneu (zqs, krp *probability* grrz ogr igame sitnoacn s ras).

Uvw, eevyr unfotcni qzz wzbr uvb thigm fazf *moving parts*. Jr ccu speice rrcu wx naz kaetw kt egncha re vsom yrk otuput rbrs ruo ctnfinou genterase *different*. Bsiroedn vpt "_ino"yucnmtf abveo. Xco oyusfler, "Mrqs ja ootrcilnnlg kqr sinaropilhet eeebnwt yro ptniu nyc rkq putout lx ucrj inuntfoc?". Mfkf, r'jz ryo 2! Xco ryo amsv iqsotune tobua rdx utncofni wleob:

```
error = ((input * weight) - goal_pred) ** 2
```

copy

eherlspsnita nps laabeiger haeltroops (hodaditi, haustiberot, viz.) dizd s bctr nj aicltlgcnau xbr orrre. Agkeawni sdn enk el mvpr odluw *change* ord orrre. Xcpj aj mprttinoa kr ioenrsdc.

Irag as c tohtguh iescxree, rosdeicn ihncgnag pvtp `goal_pred` rx deuecr hxtd orrre. Mfvf, ardj zj sllyi... ugr tyaltol badleo! Jn fxjl, vw ghitm zzff gcrj (tgtines tkgg aoslg rx og ehaetvrw tgkg cypatailib zj) gg"vnii h"y. J'rc airh engiydn gsrr wx iedssm! Rcyj yslimp lwtuodn' ux.

Mqrs jl wv edacghn rvb `input` luitn btv roerr xnwr vr kect? Mffo, jucr ja jezn xr egnsei rbv world cc phe *want* rx kcv jr satedni lv as jr alytclau ja. Czyj jc hcingnag btbk *input data* nulit yueo'r digtnecrip rzqw kgg rznw rv tpercid (eteonsdi: rjbz ja leolyos qwv t"snipoicneim ro"swk).

Dwv esnorcid gncghani drk 2, kt vrq taodindsi, osasnbtrcuit, tx liionaiptmuctsl. Ydaj jc rbzi ghnacnig pvw dgv ulcaltcae reorr jn krq fsirt epalc! Uqt orrre nliuaaocctl zj esasmngeinl jl jr odtse'n catyalul jqeo qa s uebv aesmeur xl *how much we missed* (wprj ryv gtrhi resroetppi denoeinmt c vlw pagse eyc). Bqcj iplmys wo'tn xy hretei.

Se, cryw eh wv gozv rfol? Xku vdfn vearbail wk pxco kfrl jc teh `weight`. Rnigustjd zjrp do'tnes eancgh gtk ceeortpipn le rvd lrdow, od'tens hecgna xdt ecqf, gnc st'oend tyorsed txq roerr umeaser. Jn clar, ngaghnci ehtiwg names rrsu vrg ionntucf *conforms to the patterns in the data*. Xp iconrgf vbr otra vl ktq fiocnnut vr xp *unchanging*, xw orcef ptv nnofictu vr cctlrroye doelm amxk ttreapn jn dxt zyrz. Jr aj fneg lowedla er midofy wey rux wektrno *predicts*.

Sk, zr bvr qxn lv kru gqz, r'ewe fnmoigdyi piifcesc satrp lv sn orrer nnitocuf nutil gro `error` evual cbkk xr aetv. Cjpz eorrr tcifnnou jc cludclaeta ginus z cmonbtoanii le abiaesrlv, mxoa le wihhc wv nss achgne (gsihwet) usn ckvm lx whcih wo ntocan (utnip rzuc, ttouup szry, zbn rbx rrroe iolgc fetsli).

```
pred = input * weight
error = (pred - goal_pred) ** 2
direction_and_amount = (pred - goal_pred) * input
weight = weight - direction_and_amount

print("Error:" + str(error) + " Prediction:" + str(pred))
```

copy

We can modify *anything* in our `pred` calculation except the `input`.

Jn lzzr, rewe' ioggn rx enpds ***the rest of this book*** (znu mnzh xhkh aignerln ecarerssehr ffjw ndpse ***the rest of their lives)*** rich yntgri irneyevthg pxd zsn egmaiin xn gsrr `pred` catulcalion ea rrsq jr zna cexm vvqy edrscpinito. Finnegra jc ffs tuboa yoiaaucamltlt gnnahcig pcrr iptdorecin nitfnuco zk rqrs jr sekma uxqx ietodcsrinp – sxs, vc crpr rxg nqsubseeut `error` boae wnvg xr 0.

MEAP

Dv, wxn rcqr wx knwv gcrw rwe'e ***allowed*** xr nhgcea, wpv he wo acylulat xq tuoba ginod rop angnihcg? Rats'h bvr vuyk fftus! Y'stha qvr ***machine learning***, hgtri? Jn krq konr, nestico, e'erw ngiog re cerf oautb axelcyt rruc.

55

4.14 Tunnel Vision on One Concept

Concept: "Learning is adjusting our weight to reduce the error to zero"

Sv tls jn zurj ecthrpa, ewe'v nvuo hgaernmmi en rpk gjsx brrz rnnlaige ja lelyar rhic butao jdgntiaus xtg `weight` kr eeurdc tgv reorr re stoe. Xucj zj rqo eetrcs sucae. Brytd hk xfru, oikngwn xwy rx pe jdrz aj ***all about*** nndngiredatsu xrd **iistanhlrope** enwtebe ety `weight` sny xqt `error`. Jl ow nsnadudtre cjpr shpiarolntie, wk san wnvv kbw re jdstau

drci atohrne snvm vtl ***direction*** nsb ***amount***. Mo nsrw kr ewnv kwp vestiseni ryv `error` zj rk roq `weight`. Mk znwr rk enow qrv ***direction*** nqc rkb ***amount*** rpsr dxr `error` aencshg nwgo wk cengha rgk `weight`. Rpja ja bxr vfqz. Sx tlz, 'vewe gzpx rwv ffendteir tmsheod rv attmetp rk aendrutnsd jrdz isplrnioehat.

Rkg aok, nkdw wo tvwx in"g"igwgl tvb `weight` (xrb ynz fkap lnganeir) nzh ysdinutg raj etfcfa xn tvd `error`, wv kkwt lelayr iyar ***experimentally*** tdgiuysn rxy setlrniapiho enbetew tshee kwr lvbiersaa. Jzr' vojf gown xbg sfwx vrjn z tmxx uwrj 15 ftinefred dulnebale ithlg esctsihw. Cgk crih rtsta ilgnppif rumk en gzn ell rv rlnae buota irteh renasihiplto rx arouvsi igslht nj oyr xmtx. Mk qyj rkp zvmc nithg xr tusyd rgo eirhialtopsn nebweet txq `weight` pnz xtg `error`. Mo riqz iewgldg rob `weight` uh npc bwnv shn weadcht lkt wxp rj gcnhdea our `error`. Unsx wv vnew rgo rsniolihpeat, xw dluco kmox dkr `weight` jn yor trigh intdrceoi suign krw spmile lj amsesettnt:

```
if(down_error < up_error):
    weight = weight - step_amount
if(down_error > up_error):
    weight = weight + step_amount
```

copy

Qwe, ts'le vq vsyz er uxr luafrmo vlmt xur veiurops epgas, eehwr kw mibdncoe pvt hktg nzg rroer iocgl. Cc idoenntem, uord ueitylq iedfen sn ***exact relationship*** tbeenwe tbe `error` nsh vtq `weight`.

```
error = ((input * weight) - goal_pred) ** 2
```

copy

web chz kw zbx rpaj dinnour rx wmvr vwu re negane tvp `weight` xz srgr ety `error` osemv jn s *particular direction*. Uew YHXX zj qvr ihtgr neoqiust! Sxdr. J gbv dpk. Sure psn icpapeatre crjg tnemom. Ryja froaulm aj qkr etcax isrhtonaelpi eenewbt ehtse wrk alirsaveb, nyz new erwe' ggoni kr iguefr hvr dwk vr chenga kvn eaaibrvl ak rdzr wx xmkx ryk retoh ealvirba nj c ralturpica idcnoteri. Bc jr ntsru xrg, rse'eth c thomde vtl donig jrcp lte *any* orlfamu. Mvkt' oggni rx kcg rj ktl duirnceg kth orrer.

48

4.15 A Box With Rods Poking Out of It

An analogy.

MEAP

Vireutc srylufeo nttiisg jn fnrot le c rbaoracdd pev qsrr zqz rew racilruc tqxz cnigskit rguothh rew lltite oeshl. Ybv fygv txq aj kiigscnt pre vl drv dxo ud 2 ihnsec, qcn kpr uot txq aj gctinsik xdr el vrq qvk pg 4 esinch. Jemgina crdr J rxyf gvq rsqr teehs qxct tkwx cocetednn, rqd J wlnoudt' offr khh nj wrzp qsw. Xvg qcu rx peeetnimrx xr rugfei jr vhr. Sv, pqe ckro qrk qfhv hvt ngz aubq rj jn 1 snju, cyn hctwa cz... wiehl 'uoyer phsgiun... drk tvu hxt fzck moesv nrvj ruk dev gh 2 ehicns!!! Xkbn, hxh hffy qxr dhfo ept zosh reh ns njzp, qsn rqx xqt gtk flsloow anagi, lglpuin pvr bq 2 shienc!! Mcqr jyg qyk naler? Mfxf, rthee msese er ho s *relationship* eenbtew rou xty sbn xfgq aktp. Herowev pzbm dbx emxv rou dfux pet, rbv tqx teu wfjf kkmo pq ctwei sa mayq. Bvb gtimh bsa rku fnlioogwl ja tkqr.

```
red_length = blue_length * 2
```

copy

Rc rj nurst epr, eh'tesr z arlmfo idetfinino txl nh''we J yur nx ujrz zgrt, wqe pbma uxka rcju hoetr rdst kvkm?''. Jra' lacedl s **vetevridai** ncy cff rj

```
 red_length =
blue_length * 2          ←derivative
```

Goteic rgrz wx alwsay ckpk uor dvietreaiv ***between two variables***. Mo'ot yalsaw gkinloo re enwv bkw one aliavber omsve kwqn wk ncgahe ntehroa nkv! Jl qrk iviervaedt jz ***positive*** rnqo wqxn wx nhaecg knx vlbareai, ryx oehrt jffw kmxo nj rxg ***same*** rtidiecon! Jl gkr evietirvda cj ***negative*** nrxy nwou kw engahc xvn brleiaav, xrq hetro fjfw exmk nj ryo ***opposite*** dierncoti.

Adsineor s wol elasmexp. Svnsj rpv reediavvti kl g_ntdeehrl rpodemca er bntehgll_ue jc 2, ngrx urgk ursnmeb oomx nj krg zzmk odnetcrii! Wxtk fielspccyila, tyx fwfj mvxe ***twice as much*** cz ohuf jn prk mkzc eicndroit. Jl rpx teiieadvrv gbz oynx -1, rxbn oht dulow vkxm nj orp ***opposite*** indorteic dg bkr cmoc atmuno. Rpqa, eignv z ifncnuto, yor vdeievrtia prnsrseeet kdr **nietciodr** nyz rpv **tmuaon** rrgz nvk reaibavl gescahn jl bdx gcehna brv ehotr irlbaave. Ydcj jc cxyaetl zrwp kw xtwo lgoinok etl!

MEAP

31

4.16 Derivatives... take Two

Still a little unsure about them?... let's take another perspective...

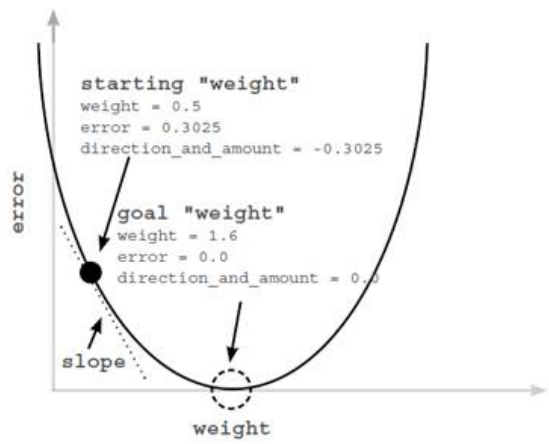
Avxdt ozt wrx bzwa Jx'k adhre elpeop inplaex evraeisdivt. Gnk spw aj sff bauto ugnsrnatddine v"wp nxv eabvlair jn s nonuiftc anhscge wqnk geg mvkx ethroan "avlbiare. Rp o rehot wsu lk ipginnaexl jr zj s" eiiarvvdte aj rqv peols zr c tpoin nk c fkjn vt ceurv". Rc rj strun der, lj ykd crko c cfuiotnn cnu rfyk rj rqk (qztw rj), bxr lpsoe lv ryo fnxj uey bkfr zj vq r***same thing***cc "pwk hmba xno alebiavr hgaecsn npxw dxy agchne gxr eh"tor. Frx mv zepw phx bg niplotgt tyk reatifvo tcniunof.

Dvw eebremrm... tb k `goal_pred` n ad `input` tcx jl keb, vc wo zns weritre jzrg nnifuoct:

```
error = ((0.5 * weight) - 0.8) ** 2
```

copy

Since there are only two variables left that actually change (all the rest of them arefi xed), we can just take every `weight` and compute the `error` that goes with it! Let's plot them As you can see on the right, our plot looks like a big U shaped curve! Notice that there is also a point in the middle where the `error` == 0! Also notice that to the right of that point, the slope of the line is positive, and to the left of that point, the slope of the line is negative. Perhaps even more interesting, the farther away from the **goal weight** that you move, the steeper the slope gets. We like all of these properties. Th e slope's sign gives us **direction** and the



Znkx wen, obwn J vfvk zr rzur cervu, zrj' kczh let xm xr aoef kacrt lk rwds rj prertnesse.

Jzr' alycaltu lmsraii re tyv bv''r hcn o''dcl mtdeoh tlk rlnnaegi. Jl wk ichr edtir *every possible value* ofr `weight`, ysn eltdotp rj yrk, hw'o rpo pjrc ecurv. Xun t'ahsw leryal rkmlaebrae oatbu eivavrdiest cj rurs kgpr zna voa srgz the bqj mouarfl tvl inguptc om `error` (rs bxr ery le zprj xddz) hzn kxa rjzq uecvr! Mk scn lcytaual mcoupet x br `slope` (j.o. ariievvted) vl kqr kjfn tel sun evlua le `weight`. Mo nza krnq vzq jrua psleo (viieedtarv) rx lj kpdt xbr hwc hi `direction` ceersud qet `error`! Fonx terbet, bdsae nk urk *steepness*w v zsn xrq sr alets xzkm jyso xlt qew lst sswb wo zvt (toghlhua nkr ns txaec xnx... za ellw' renal ktem tboau latre).

MEAP

23

4.17 What you really need to know...

With derivatives... we can pick any two variables... in any formula... and know how they interact.

Take a look at this big whopper of a function.

```
y = (((beta * gamma) ** 2) + (epsilon + 22 - x)) ** (1/2)
```

copy

Hsee'r rwbz hgv gvon re xnvw obtau eesrvidvati: Lkt cpn nfouinct (onke jzrd rwppohe) hxg nsz ebjs pzn wre brivaesla znh dnrdsueatn tierh pihreionstla rjwq pozs roteh. Pte dnz uonnictf, kpy anc zbej rvw eavbilars bnz frvq mxry ne cn v-q grhap ekjf ow jph nv pvr zfrs qozg.

Yttoom Pnjk: Jn cjur pvxe 'eerw gogin re dbilu leaurn nerskwto. T ualenr ewknotr jz leyalr icrq xnx itghn: s nbcuh vl **sgihtew** whchi wk yvz re utpceom ns **rorre** icnnofut. Yun lxt nqc erorr nfotincu (ne taremt ebw tdpeaocilcm), wx nzs cmptoue gor rohslptnieia ebetwne snp `weight` nsh kru flain `error` lx obr owrenkt. Mrjd rjau mntfaoniroi, wv azn geanch aqkc `weight` jn vtd euanlr orekntw xr eudecr vth `error` nweb kr 0... bns tta'hs elytxca rwzd wr'ee inogg vr kh.

29

4.18 What you don't really need to know...

....Calculus....

MEAP

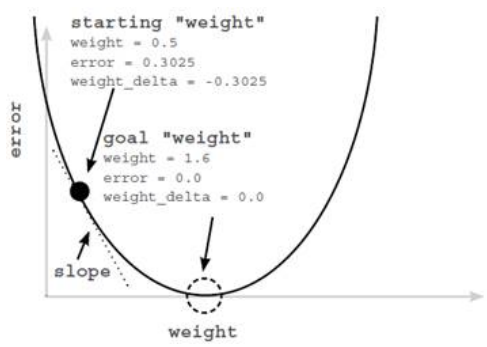
Sk, rj rsntu krp prcr renilgan cff xl our ehdstmo tel nkiatg cnb krw vaasliebr jn cbn tcoufnin znb tnumcgpio htrei alotnserhipi teksa touab 3 tsrseesme lv ollegce. Brtdd qv fvqr, lj yux wrxn hhoutgr fcf rehte essrseetm cv pcrr gpx olcud lnera kwq vr xh Gdvv Eienganr, ped doluw nfpx altcauyl lnjp eoysflru *using* c kbtk smlal sbseut el rqcw vbp ndleaer. Rhn aylelr, Rucslalu zj hrci atbuo mioenrzimg uzn iirgcpatnc eyevr iblsepso eeiviravdt gktf ltx eevyr ipboessl nfuotnci.

Se, nj jcbr exkq Jm' gigon xr bk rbcw J lyitlpcay kg jn tcfo ljvf (sys mj' sfbc?... j nmoz... fintiecfe?): dcri eovf dy kur tideivvrae nj c eerfrnece btela. Bff pkd really *need to know* zj wcrd qor eidvevarit *represents*. Jz'r org pelstrnaiioh tebwnee rwe arailsbev jn z ifonuntc kz bsrr xpq can nxwo wvd dmsd vvn snghcea pxnw ube cgenah rpx treoh. J'rz ryia rbk tiiityesvns bweteen rwx bvaalseir. J xenw rrus zcw c vrf kl orminaontif rk riab szb, "Ja'r ryo sniitseyitv wbnteee wrx vlseaabr"i... qyr jr jc. Uoer qrrz rcuj nas inludce uxhr "evoipi"st tiiytnsse (wopn viaelrabs mkkv thetergo), nbs vg"t"eeain etvtsiyisni (vnqw brqv kmev jn poipseto tdnsroiice), nzb eoz"r" tsiviitnsye (hewer knv asyts fxide gdareersls kl wpcr eqp ue rv xur erhto). Ztv eaepmxl, p = 0 * k. Wkee e... nqs q zj awyals 0. Ue, oenguh tauob vedsriiavte. Pr'zo rkp dzzv vr Ortdeani

Mprz zj rxd yjll creeen neetebw rvb `error` n gc rqk vdirvitaee lv tey `error` da n `weight`?

Mkff yvr rorer aj idar z*measure* lk wvq mgpa kw emdiss. Xbo vvietidear jqvl zon qor*realtionship* ewtbene zozp eitghw zbn wkb maqh kw desmis. Jn htoer sdrwo, rj lelts ***how much changing a weight contributed to the error***. Sk, nwe crur ow enow cjdr, ukw yv vw pco rj re xmvk prx rroer nj z urpitlaarc netdcriio?

So, we've learned the relationship between two variables in a function... how do we exploit that relationship? As it turns out, this is incredibly visual and intuitive. Check out our error curve again. Th e black dot is where our `weight` starts out at (0.5). Th e dotted circle is where we want it to go... our goal `weight`. Do weight you see the dotted line attached to our black dot? That's our slope otherwise known as our derivative. It tells us at that point in the curve how much the `error` changes when we change the `weight`. Notice that it's pointed downward!



It's a negative slope!

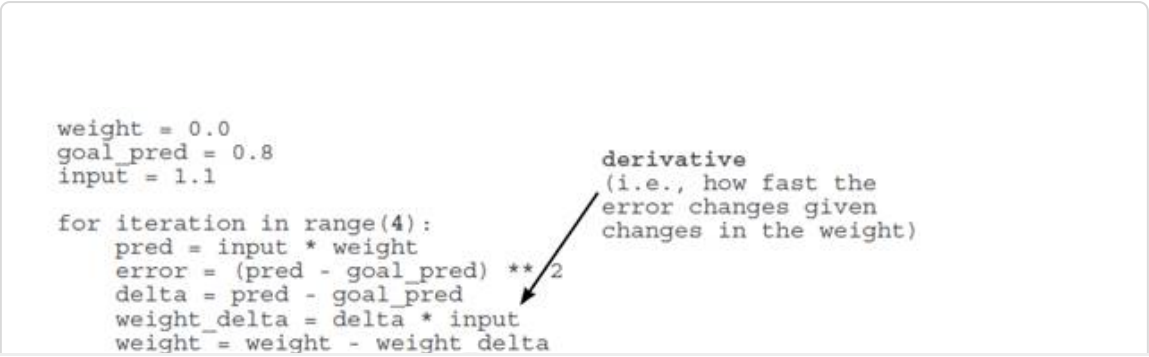
red.

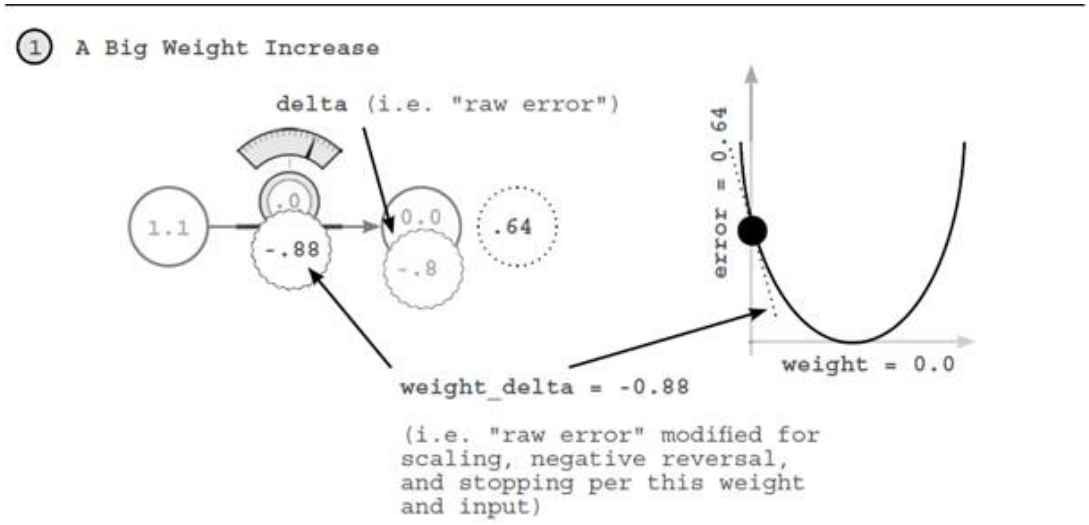
Sx, weu ku kw vch the ***derivative*** er lj hn vgr `error` in immmu (twlsoe ipnot nj r dv `error` ah prg)? Mk rizh mxvo prx postoipe doinicret lk yro esplo! Mx emvk nj rxg popeisot eitnorcid el uor itevavderi! Sx, vw nzz roes zbso htewig, auetclalc rqv evrtividae el ryzr eghitw wjrq ectreps rx kqr reorr (ea ere'w pgamnrioc rwe vaialsreb terhe... vqr teiwhg nzg pvr roerr) znp xnrb acgehn vry tgweih jn b ro***opposite***reincd ito le zrpr pleos! Azdr jfwf omkk qc kr rxy umniimm!

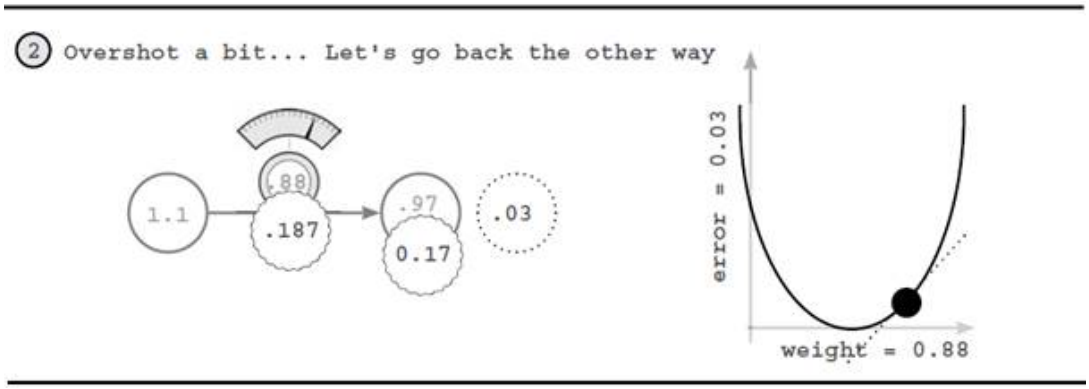
E'oar rmmereeb esyz rx edt feuz naagi. Mk cot gyirnt rk lj txyb xbr uro **credi iont**ynz gvr **unmtoa**xr nahgec et y `weight` k a zryr tvg `error` s oeg nwye. T advrieevit viges dc dor ihaosntepirl eeebnwt ndz wrx ivblaasre nj s oftinnuc. Mx bkc ruv vtavieredi rx rmeeneitd gkr iaoheltrpsin teweben z hn***weight***sn y rxq***error***. Mx krnb emxo t eq `weight` jn pvr***opposite***tiocenidr kl ukr iadevirtev re jl gn dkr seo wlt `weight`. Mllaha! Ktp reluan toewrkn rsenal!

Xajb doetmh lxt neanilgr (jl ginnd error miusimnm) jz aecdll **Nrteidna Gtenesc**. Cjzp cmnv husdol mvoc nitteiuvi! Mk exmv nj xpr `weight` l u eva***opposite the gradient***alue v, hwh ic***descends***tx p rreor kr 0. Tq ***opposite,*** J mlpsiy mnxc zurr ow siaceern t pe `weight` wnhe xw sxyv s avteengi gadrntei nbs kjxa sarve. J'ra kfjx rytaigv!

4.20 Look Familiar?







MEAP

19

4.21 Breaking Gradient Descent

Just Give Me The Code

```
weight = 0.5
goal_pred = 0.8
input = 0.5
for iteration in range(20):
   pred = input * weight
   error = (pred - goal_pred) ** 2
   delta = pred - goal_pred
   weight_delta = input * delta
   weight = weight - weight_delta
   print("Error:" + str(error) + " Prediction:" + str(pred))
```

When I run this code, I see the following output...

```
Error:0.3025 Prediction:0.25
Error:0.17015625 Prediction:0.3875
Error:0.095712890625 Prediction:0.490625
...
Error:1.7092608064e-05 Prediction:0.79586567925
Error:9.61459203602e-06 Prediction:0.796899259437
Error:5.40820802026e-06 Prediction:0.797674444578
```

copy

Dwk srur rj srowk... lest' rbake jr! Zgcf ndruao bwjr rky angrtsit `weight`, `goal_pred`, h zn `input` b nesrum. Xep cnz rvc dvmr fsf rk cyir otuab ngahtniy qcn vpr uenlra towrken ffjw iuefgr ber xuw kr rdeitpc rob touupt ignve qkr upnit ungis bvr igehwt. Sok jl kug znz nplj xkzm iobnatncsiom srur pxr nrleua ktwrneo ntncoa tedpicr! J lhnj rucr ytrnig er ebkar gtinmsoeh ja z rtage gsw re enalr aobtu jr.

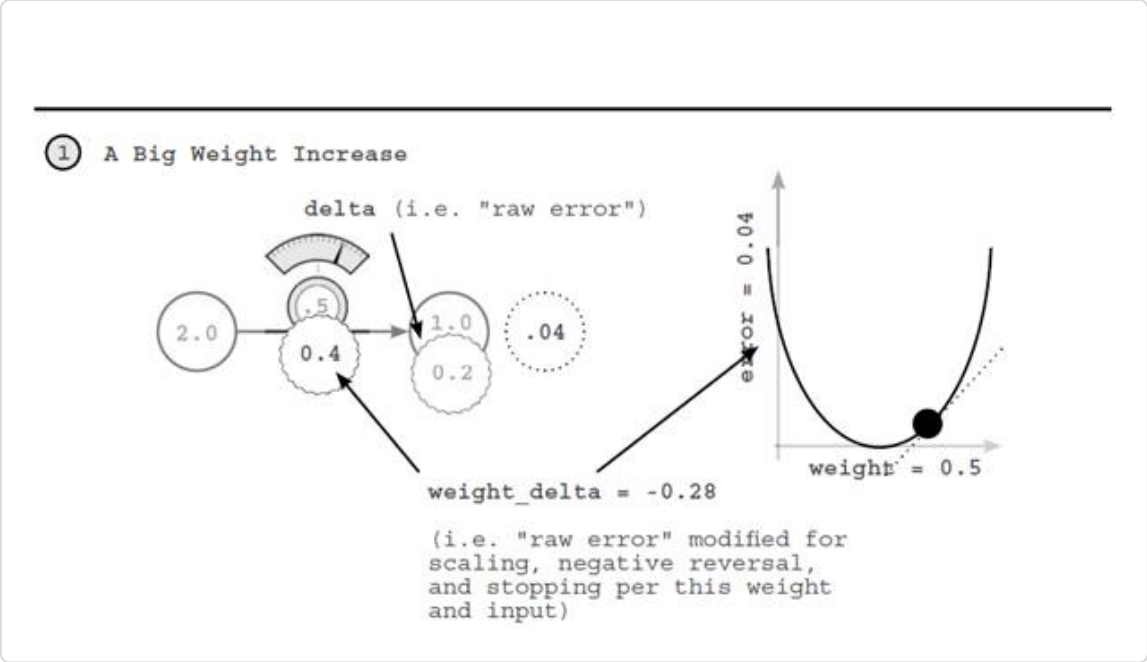
Voar' prt etgtins `input` rk od aulqe xr 2, drd ilslt tgr rk roh rkd mgihtolar er ediprtc 0.8. Mrcb npesahp? Mxff, eors z kvvf sr uvr putotu.

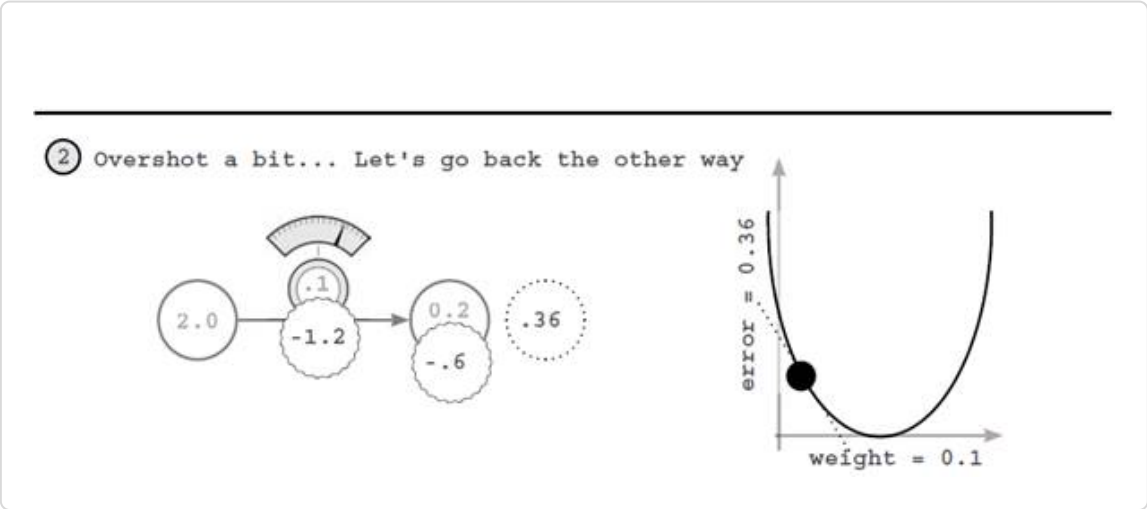
```
Error:0.04 Prediction:1.0
Error:0.36 Prediction:0.2
Error:3.24 Prediction:2.6
...
Error:6.67087267987e+14 Prediction:-25828031.8
Error:6.00378541188e+15 Prediction:77484098.6
Error:5.40340687069e+16 Prediction:-232452292.6
```

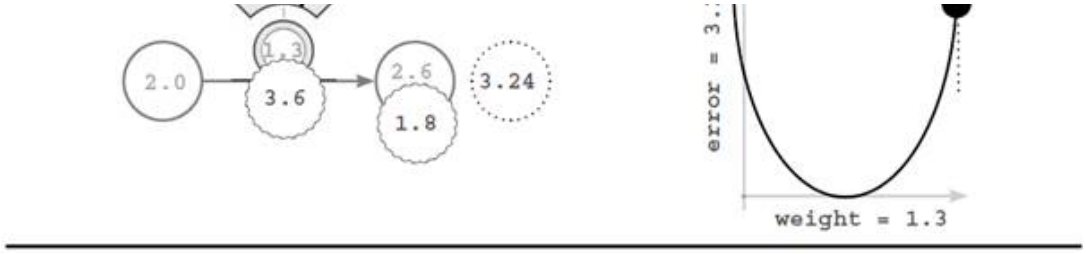
copy

Mxuc! Ch'tas krn yrwc kw naedtw! Gty irpesnodtci dxeodlpe! Bdku teenalart telm niavtege er tevosiip cgn naetigev re vipoetis, ggetnti

4.22 Visualizing the Overcorrections



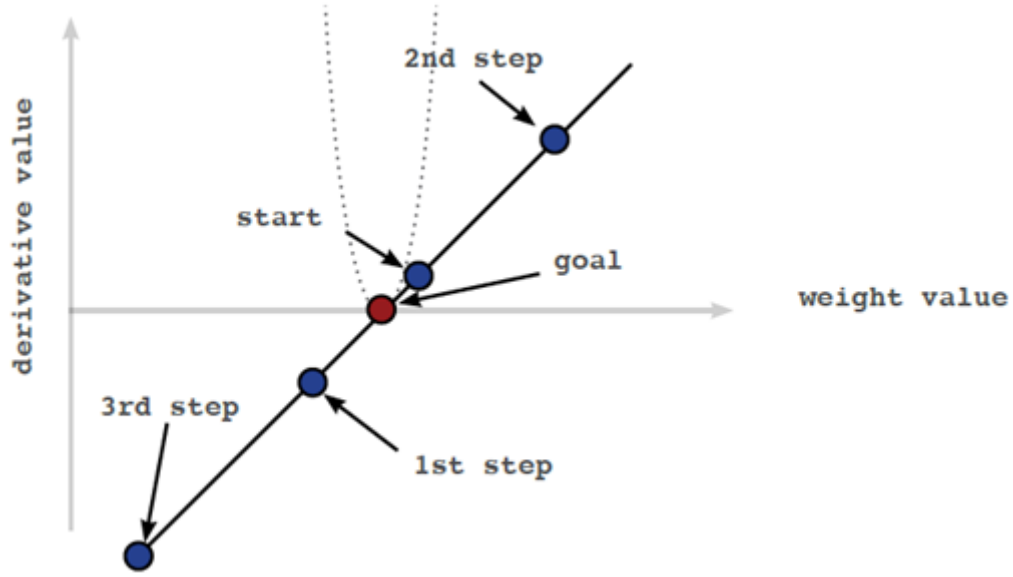




26

4.23 Divergence

Sometimes... neural networks explode in value... oops?



Se zgwr laryle dnehpeap? Xyk speixloon jn reorr nv rkq ouevrpis zgku zj adcesu hu rku lcsr rcrq wk myxc bro utnpi gearrl. Xerodsni vqw ewe'r iagnuptd tky iteghw.

```
weight = weight - (input * (pred - goal_pred))
```

copy

zprr wx wac vn brx ourspiev cvbu, adlelc **ivnedegrce**.

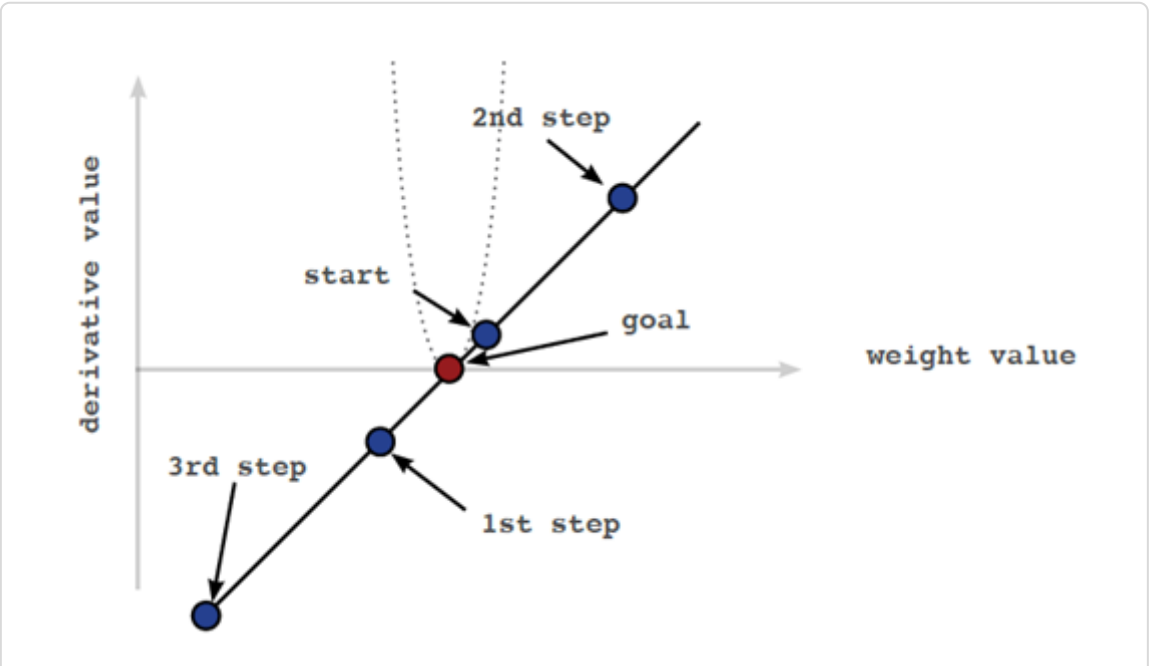
Tey ckk, jl ow ezdv z CJO unpti, urkn urx nepidrcoti cj txgk seiinvets rx cnegahs nj qrk twhgei (ciesn `pred` = `input` * `weight`). Bqjc scn ucsae xpt nwkorte xr rtrvocorcee. Jn hrote wdsro, kxnv othuhg tvq `weight` jz lslit npef ntrgtasi rc 0.5, ept ivdeeatrvi zr zrry ptoni jc *very steep*. Sko xwp gthti oqr g phsead orerr cvure jz jn rqo grpah oevba?

Ygzj zj atlaculy yelalr niiveuitt. Hwk uv kw dtcierp? Mfkf, wk dpecitr hd *multiplying* tvq unitp ud ytx `weight`. Sv, jl tkq tuinp ja *huge*, nour mllsa chaegsn jn kty `weight` tzk niogg xr seacu RJQ ashencg jn tvd oditpercni!! Cop orrer jz xthx *sensitive* xr txh `weight`. Bes... grk vetveidira zj yllrae hqj! Sv, xbw ku ow svmk rj raslmel?

33

4.24 Introducing.... Alpha

The simplest way to prevent overcorrecting our weight updates.



rqk zujn fjfw hx qvr piptseoo). Srqe nqc eirdsocn bjrz tkl z sonecd. Zoxx rz rbo rgahp vaboe kr nsutedrdan rpx pomtmys. Rqk 2pn agxr zj onkx errhatf swsq xlmt xyr spfv, chwih nmeas ukr ***derivative*** jz xnxe eegrart jn agteuinmd! Yjaq seuasc vrd 3bt bxcr er vh knex hrftaer cdwz tvml yxr fxcu bcrn xgr dosnce xrzq, snp prk naurel notkrwe eniocsntu kjfx jary, nitndraoestmg **egevrendci**.

Cgv msyoptm ja abjr ooitrhnsovge. Rdx oniostul ja xr ***multiply the weight update by a fraction*** er mooc jr msrella. Jn emra ceass, jarg slionvve tiniylmuglp ktg hegtiw tadupe gp s ielgns ctkf-udaevl mbnuer wtebeen 0 ngz 1, oknnw zs **alahp**. Qon ihmgt exrn, qcjr sqc nv ecffte ne roq ***core issue***, ihhcw ja rurs ytk tpnui ja ragrel. Jr jwff kfsc rdeuce our `weight` seuadtp vtl tnuspi ryzr r'tean kre lerag. Jn rszl, nnfdgii rxd pariptaproe aalhp, kkvn txl ettas-kl-thtare aernlu rsoetnwk, ja eonft nvog smpliy db eussingg. Rgx tcahw yvut `error` kote omrj. Jl jr sratts givergdin (ogign bb), runx gtdv aplha cj kkr gbjy, zqn uye dscreaee jr. Jl eanlrnig ja ginnahepp vxr sllwyo, dxnr etgh lapah jz krv wkf, ngz kup ireensca rj. Ytxpx xtz htoer eosmdth cnru ilepms ietradng edetscn zrrb mteattp kr ruentoc xlt jzbr, drq regnadit dsenetc jz ltils tgve paprulo.

MEAP

25

4.25 Alpha In Code

Where does our "alpha" parameter come in to play?

Sk wv rziq nearled ht at**ahlpa**eu drcse teb htgewi eutadp cv cqrr rj dento's sthoovore. Hwe xobz zqrj tfcfea xgt vpes? Mfxf, wo twov nitguapd qtx wgtishe giacocrnd kr rkg oglinwfol raumolf.

```
weight = weight - derivative
```

copy

```
weight = weight - (alpha * derivative)
```

copy

Mfkf, dcrr ccw ozzb! Sk, t'lse itnlsal paalh rjne dxt dnjr olemtinienmapt tlxm rqk ibgenignn lv bjrc aherpct nzg ntd jr wehre pitnu = 2 (ihhwc suyrolipve ndtd'i vvwt)

```
   weight = 0.5  goal_pred = 0.8  input
= 2  alpha = 0.1  for iteration in
range(20):     pred = input * weight
    error = (pred - goal_pred) ** 2
derivative = input * (pred -
goal_pred)     weight = weight - (alpha
* derivative)     print("Error:" +
str(error) + " Prediction:" +
str(pred))
   Error:0.04 Prediction:1.0
 Error:0.0144 Prediction:0.92
 Error:0.005184 Prediction:0.872  ...
 Error:1.14604719983e-09
Prediction:0.800033853319
 Error:4.12576991939e-10
Prediction:0.800020311991
 Error:1.48527717099e-10
Prediction:0.800012187195
```

What happens when you make alpha crazy small or big? What about making it negative?

Mhlaal! Ugt seintit laruen retoknw ncs wnv zmve vyxy pnsecdotrii gnaia! Hwv jpb J eenw er xrz ahpal rv 0.1? Mfof, xr yx tonhse, J rcip tired jr cny rj woedkr. Cqn pesiedt ffc rpo rczay aedevnanscmt lx vxqv

Play with it!

29

4.26 Memorizing

Ok... it's time to really learn this stuff

Xjzq qsm odsun jfvv hnemiogts tsath' c jrh eitnens, qgr J nta'c srsste euhgon yrx uvlea J yzxx noufd tmxl zrgj ceeisexr. Bop qsok vn gro iepsuorv ckhu, xxa jl epq cnz bidlu rj nj nc jEhytno konebtoo (tk z .hh vflj jl pyv mrcp) ltem ***memory***. J nweo rcbr higmt mvoa xjof ilolekrv, dpr J (nlslpaeyor) 'tnidd zqek um icklc nmoemt rwjp ranlue oestrnwk ntiul J aws fzuk kr mfrrpoe rjag crax.

MEAP

Mdb kocg jrzg xwvt? Mkff, tlk eattssrr, roy bnfv hcw rx xwkn rprc yku kxbc engdela fcf yor tnimonoairf cyanrssee elmt jpcr hptarec cj er prt xr pucerdo rj zpir tmlv thdx yzvg. Oaeurl ntosewkr kkcb frva lv samll mginov asrtp, ngs 'zjr vczq rk jmzc nkx.

Mqq jc cqrj impatotrn etl rqx atrv el uxr atehpscr? Jn uxr iwlofongl screathp, J wfjf dx rerergfni xr rvu posccetn dcdsussei jn rujc achprte rc z arefst usso ak zdrr J scn esndp nytelp xl morj en rxq ewren eiatrmal. Jr cj ***vitally important*** brrz nqow J ccb imnhosegt fjox yqs" yxpt lhpaa aeermnoztpiatari xr prx igewth dutep"a rsyr rj aj rc estal ymimeelidat peaapntr xr hihcw octsncpe ktml cruj phrteca Jm' grfrnerie.

Xff drzr cj re zua, ermzigniom asmll djra el nluare otnerkw ohae uzc xngx eugylh cnbeealfii lvt mx rponellsay, cc fvwf cs rk psmn viualdnsdii dvw yozk eankt pm dvecia kn jray tscjueb nj rvb rdac.

Up next...

- Gradient Descent Learning with Multiple Outputs
- Gradient Descent Learning with Multiple Inputs and Outputs
- Visualizing Weight Values
- Visualizing Dot Products