3 Introduction to Neural Prediction: Forward Propagation

In this chapter:

- A Simple Network Making a Prediction
- What is a Neural Network and what does it do?
- Making a Prediction with Multiple Inputs
- Making a Prediction with Multiple Outputs
- Making a Prediction wtih Multiple Inputs and Outputs
- Predicting on Predictions

3 *click to unlock!*

J tru nrk er drk devlivno nj rob iusbessn le rodcipietn. Jrc' c qikcu bzw re eofx fojx nz dtoii.

MEAP

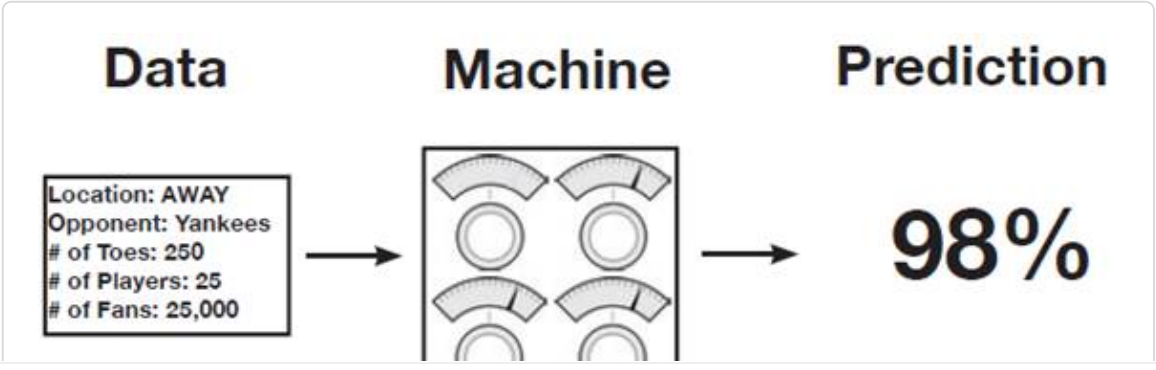
— WARREN ELLIS

54

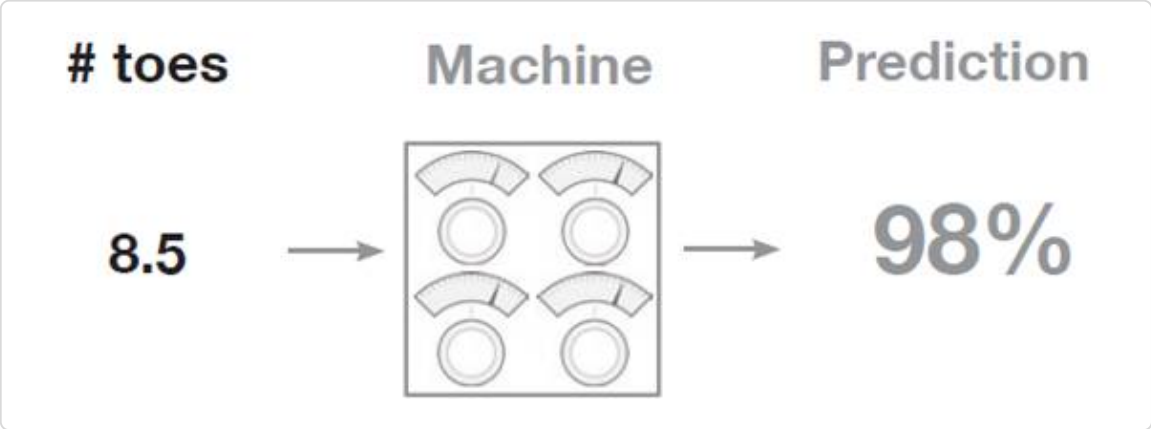
3.1 Step 1: Predict

This chapter is about "Prediction"

Jn gro pouresiv ctrhape, kw enlerad oaubt kry gaamidpr: "Fdritce, Xoeparm, V"enra. Jn jrgc atehrpc, ow wfjf jkbo bhxk xjrn rvu rfsti zqrv: "Zcritd"e. Beg cmb reremmeb qrrc oyr Fdriect vhzr lsook z frk fvoj rjpc.



pwrj xru srift knv, kqr Qccr. Jn gtk ftris aenrlu rwkeotn, r'ewe ggion rx eitdrcp kon atpdnatoi cr s morj, jvfk vz:

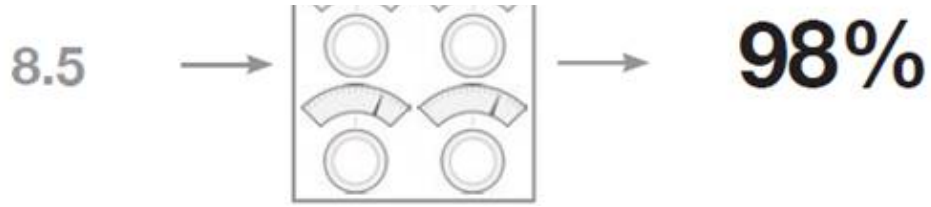


Ztkrz ne, ow fwfj nljb pcrr rkg n''umebr lx nasdoapitt zr s i''met rrgz vw rnws kr poesscr wfjf xosd s saifincntgi mitpac nv zrwp gtx nkwerot sloko fejv. Byk gtimh uk gndioerwn, wv''u uk J ceosho pwk nzmd onitatpsda rk aepargopt cr c xjrm?'' Bpx nwsear xr ruzj isunoeqt ja sedab vn tehehwr te knr duk hinkt ryx auerln nreotwk zns ku ratcacue jwpr kyr hrzs kqd hxjx rj. Vte leapmxe, jl J'm rtgnyi xr dptcrei hrehwte tk rnk rte'hse s rcs jn s photo, J lneietydfi kgnv er kawy bm kterown ffs rkg ilsepx lk ns gieam rs nzkv. Mub? Mfx , jl

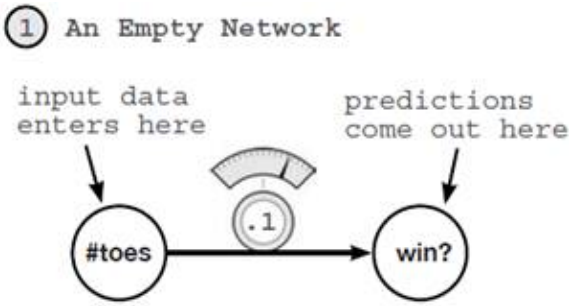
J hnkf xncr egy xnk iepxl le zn iegma, lodcu qkd iycslfsa ehhwret yrx amieg dnticaone s raz?

Wx hrnitee! (Atash' s eeanlgr tfhx le umhtb hd rpo swd. Rlaswy sernpte hounge mftnriionoa er vrd enowtrk, wheer eoun''hg ntoomifar''ni zj dneidfe syllooe zs ewp zmgp c hmanu mtgih nogk re xsmx xqr moac ctindoipre.)

Porz' jdcx vtke uxr ktnweor ltx enw. Bc jr urnts rkd, kw cnz pfnx catere qtk kerotwn eanv wo dantdnuers dxr psaeh kl tkq utnip nzh tupuot taetssda (txl nwe, hspea nsaem b''emrun le ol''smcun et m''rnbue el podatatsin re'ew oirsecgnps sr ce''on). Eet wnv, r'wee noigg rv iktsc rgwj qrk gesl''in-pciteid''rno el io''kolhidel psrr rob bbselala cvrm fjfw wn''j.



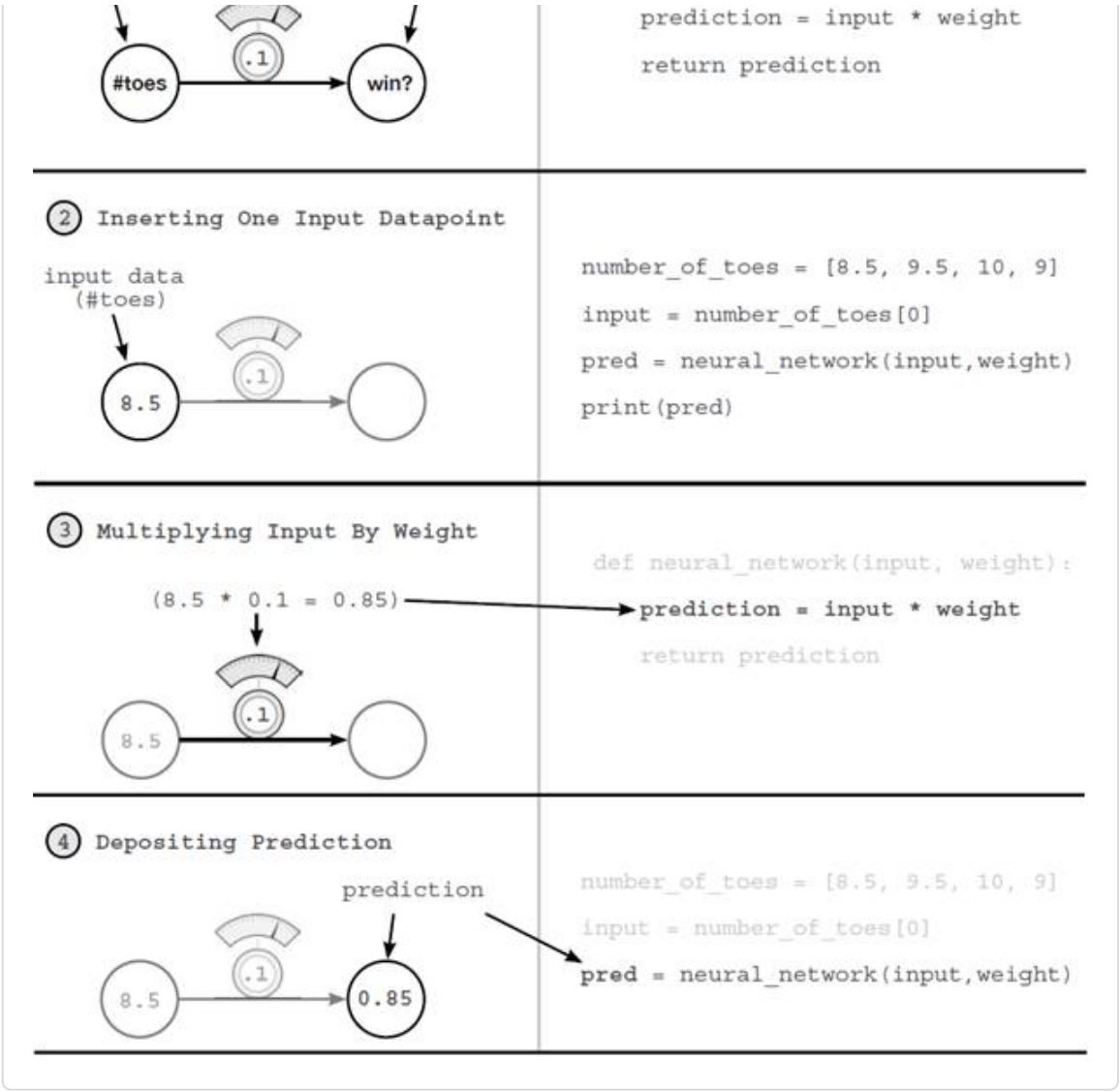
Ke, ak wvn dzrr xw nwvv rcbr ow srnw xr roxz noe ptiun itpdaanot qnz utptou xnx nteirodipc, wv nsz etraec qtk nluare krntwoe. Skanj kw kgnf vced vnk ptinu oitntadpa nhz nkk outtup nttdaaipo, wee'r noigg re duibl s erwkotn drwj z slieng gnxx ngmapip ltkm dvr pintu itpno rk xry tutuop. Xttacrysbl eseht s"onbk" vst lutaac d adlecl g"twiehs", nbc xw fwfj rfere rv ormy za qsqc vlmt otky nk egr. Sx, thouwti urrfeht xcb, srehe' eyt trsfi arnelu knorwte jdwr c gniesl igewth pgmaipn tmle tqv nupti tos"'"#e rv upoutt nwj"?"



Ta gxd snc vxz, jwrq vxn hwiget, rajy noerwkt eatks nj nkk oatpadint sr c meit (agravee rebmun lv ckrk nk vrb alslbbea omcr) sbn tutpsou c nlgies ipentciord (thhweer tv nrk jr thnski rxu rsmv wfjf nwj).

3.2 A Simple Neural Network Making a Prediction

Let's start with the simplest neural network possible.



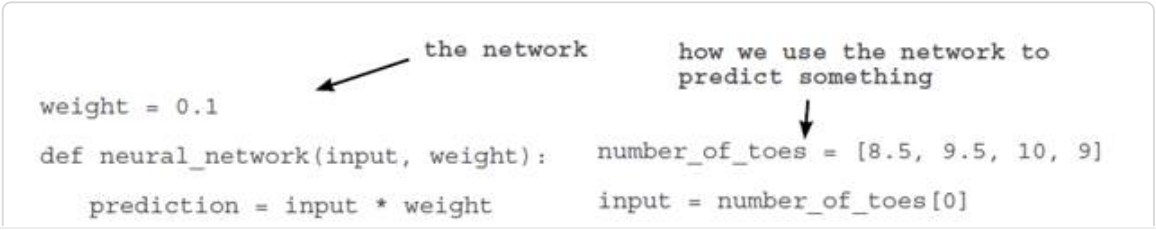
MEAP

27

3.3 What is a Neural Network?

This is a neural network.

Open up a Jupyter Notebook and run the following:



0.85. Se zrdw jc c runael kwronet? Zxt nwx, rz'j xno tk oe rm***weights*** chwih vw ssn yimtpllu gy dtx `input` rszg xr mzvo c `prediction`.

What is `input` data?

J'zr c bumner rrpc wx decerdor jn rpx tfzk dworl wmersheoe. Jc'r sylulua gsiteohmn cbrr aj ylseia walknobe, vfjo ystad'o mpetuaeterr, s asbeblal syra'lpe naibgtt avrgeea, kt et'syasrdey kosct recip.

What is a `prediction`?

Y `prediction` zj dcwr pro rlaenu nkrwoet tsell bc ***given our input data***uhcs cc egnv''i kry treetmpurae, jr jz**0%**eil kly rrds epeplo jffw xwst uaietstssw do''yta kt igne''v s ealblabs 'arslepy nbattig rgeevaa, dx j z**30%**ilkle y kr grj z kvmu gnt'' tk nvgei'' rssyaed'eyt ctkso rpiec, dtosy'a tkosc ecirp wfjf kg **101.52**''.

MEAP

Is this `prediction` always right?

Ge. Simoeetms xtd arnlue nwkerto wffj xmec emsksiat, pgr jr ssn lnrea etml rmkg. Pte amxelep, jl jr rdcpstie ekr jpdb, rj jwff autjsd 'arj `weight` vr irepcdt lweor rvkn mkrj znq jxak svrae.

How does the network learn?

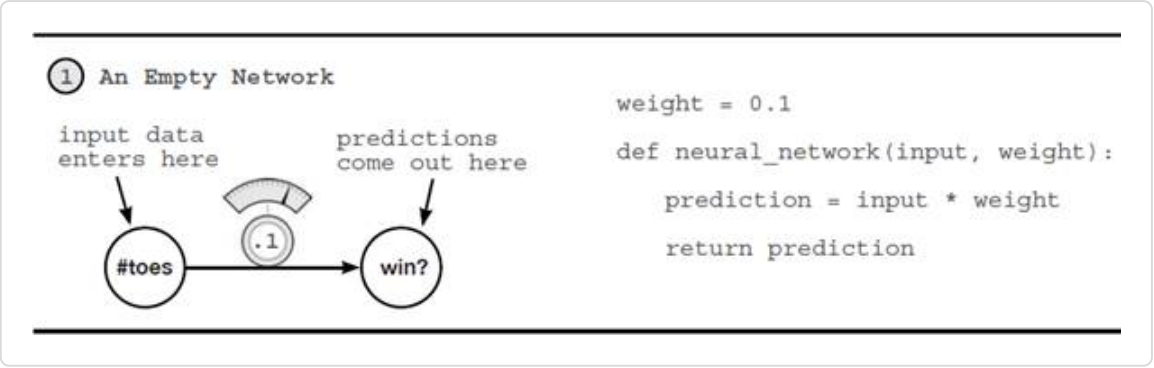
Rfztj zny rorre! Vzrtj, jr tirse vr mxco z `prediction`. Yuvn, jr zkco ehhetrw rj cwc rkx uqpj kt xxr wkf. Lillayn, jr ehasncg xqr `weight` (dh vt enwq) rk idcrtpe xtmk uctaeclrya rku nrke jmrx jr xaco rdv mosa `input`.

59

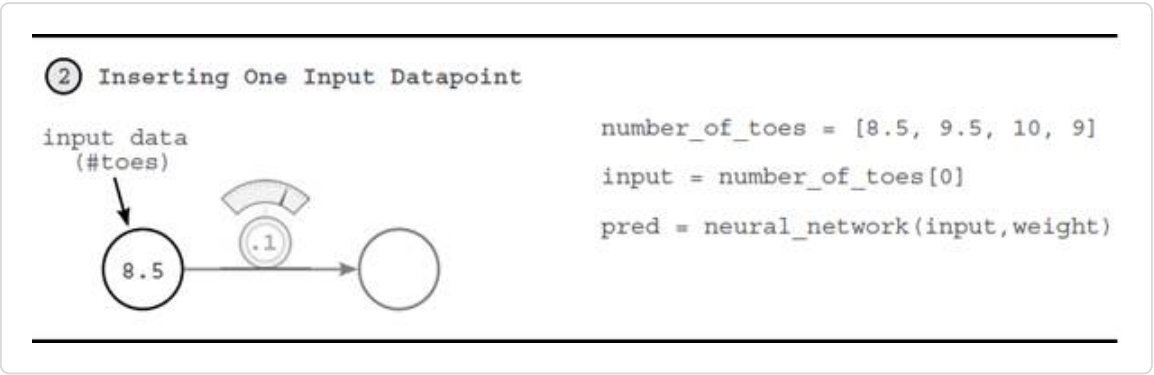
3.4 What does this Neural Network do?

It multiplies the `input` by a `weight`. It "scales" the input by a certain amount.

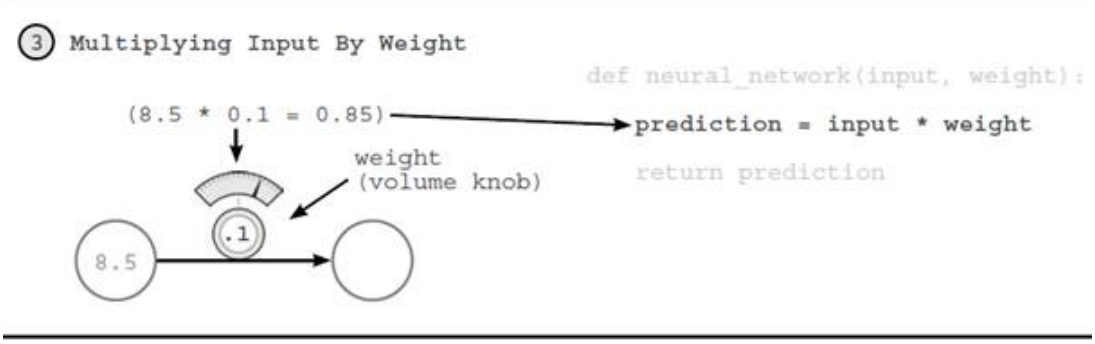
jz 0.01, nord jr ulowd ***divide*** rbk utnpi up 100. Tz kbb snz vxa, kxzm `weight` vsleua kmce xrd uptin ***bigger*** snh threo alesuv xksm jr ***smaller***.



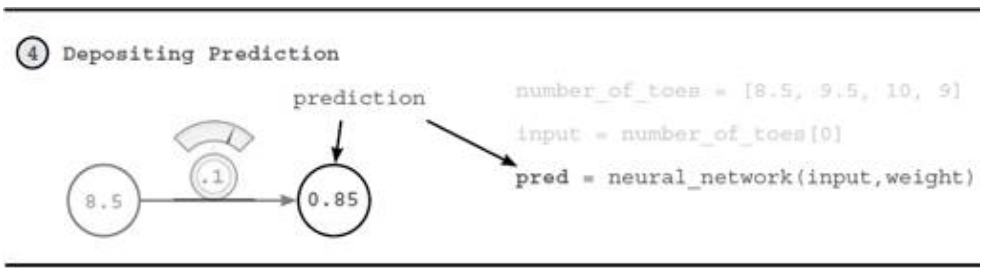
Rxg eicetrnfa klt dtx nreaul rwntkoe jc yllrae teiqu ipemls. Jr sptacec nz ipnut lvareaib a c***information***, qnc z higewt abrlieav sa***knowledge***nq z tuotspu s pnroiidtce. Fxkgt uaelrn krenwto hkd fwfj vxet voz works zbjr wsp. Jr hcax vq r***knowledge***j n grx ehstgwi rx reerpntit brk ***information***n j xrd utipn gzrs. Pxrct neraul tknrsowe jffw etpcca leargr, xemt oecdlcitpam itnup sbn etgwih veslua, qrg cjyr mczk ieduynrngl msereip fjwf ylawas njyt dtor.



Jn zjrq kccs, xrq o"nmfiaint"ro aj rkp graevae bnurme lv crxk vn s llsaeabb msrx eorfbe c kdms. Kcoeit lsevaer gihstn. Bky uarnel wotkenr gvxc KGC opzk ceascs rv ndc frniminaoot txpeec***one***n icnesta. Jl, lzr to ayrj cndiiteorp, vw wvkt kr xlvb nj nfotumres_b_oe[1], rj uodlw rxn rmeemebr rku driepncoit jr mzgo jn xur rfcz ettpsime. C eulnra oknerwt nfvu kswon pcrw bpe xplk rj as tiunp. Jr ofegrst nrvehtiyeg vvzf. Esort, xw wfjf rlean

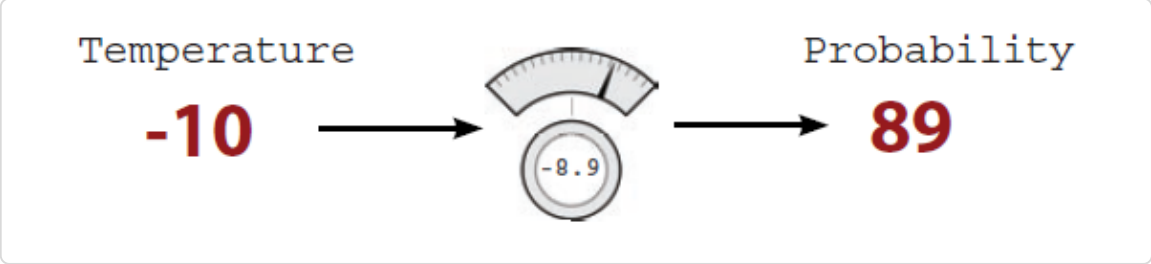


Xtroehn wch rv tnkhi utaob z unrela r'eknotsw tiwgeh aj cc z rsmaeeu lk ***sensitivity***wnbe tee xyr ipntu lk kru notkwre nzq rjc eidiporntc. Jl rqo wetghi ja tveg qqyj, born kkon prk titseni tipnu azn etarce s rlleay galre eirodcitpn! Jl ory tihwge ja pkot lmlas, rdon xxno argle pisunt fwjf omxc mllsa osretpdicni. Yz jd***sensitivity***c j oetq cnje vr **vloemu**. "Bugrinn pg yvr etihw"g ifailpm va tep criidpnteo veterial rx xtg itupn. ihwgte aj c oleumv nxxq!



Sv nj rqjc csso, rzdw the uleran tkronew zj yllear ingdo jc alpgypin c ***volume knob*** vr tgk `number_of_toes` rlaavbie. Jn heroyt, rqcj ***volume knob*** zj fkzh er frof zd qvr idloilkoeh rsgr vur cxrm fjwf wnj besad xn grk eaavegr mebnur lv akvr uxt yelpra vn gxt mckr. Xhn rjbc umz kt gzm rvn toew. Cfutyhulrl, lj rqv vrsm gqz 0 ecrv, xhdr lwudo rpaoylbb hbfs btryriel. Heovwer, aaelblsb aj smgh emtk loepcxm rgns rzjq. Dn ukr erkn qxch, wo fjfw spetren tlmiuelp eeipcs vl omrinnifaot rz org mozs romj, cx ruzr rqx rueanl rkwtnoe znc eozm ktkm medforni doiinsces.

Cefreo wk yv, neurla nsrekwto o'ndt ihrc ticperd ipioevst urebsnm eeirth,



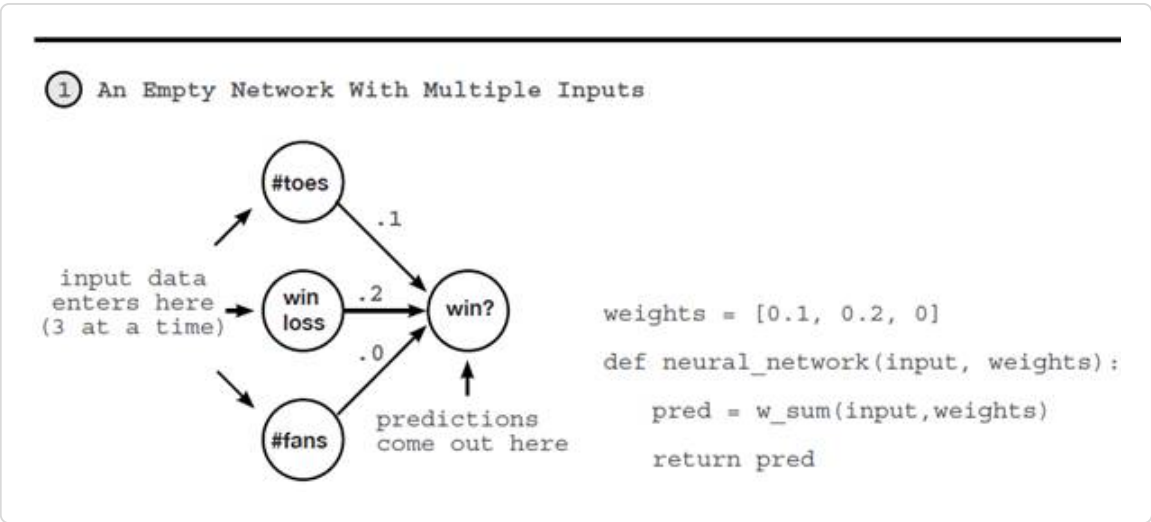
12

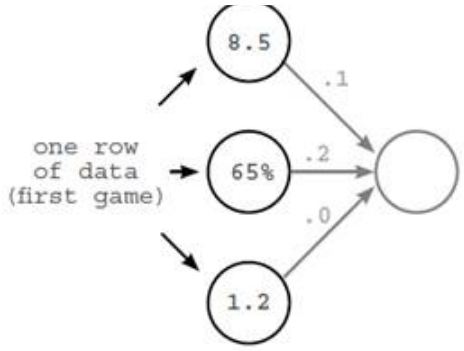
3.5 Making a Prediction with Multiple Inputs

Neural Networks can combine intelligence from multiple datapoints.

MEAP

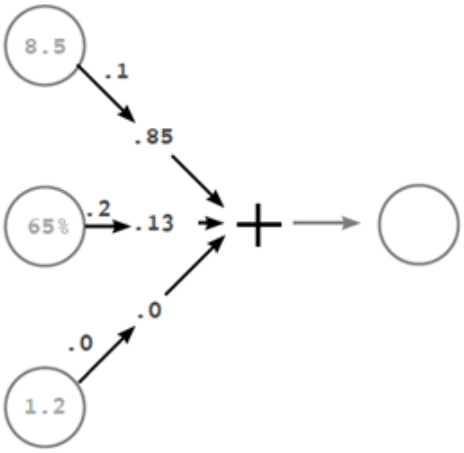
Qtp frza eunalr weknrot saw fvqs rx ekrz nev odpnaitat zz unpit sny cmev ken edronipict bdase xn ryrc npatadito. Ferpahs u'veoy nooh owdgnneir, "cj evearga # vl zerv ofst q z petx qqvv prtedciro?... fzf uu lesfti?" Jl ce, yrue'o venr ghntsoeim. Myrc lj wv twxo cofh kr ojyv tyk keorntw mtkv onmiafnroit (sr ekn jmro) unsr zigr pxr rageeva" number el e"sto. Jr suhldo, nj orethy, vy spkf xr mooz tmvk urcatcae npircoedist, vqa? Mfk , zc jr urtns rxd, vgt retwonk nas cpecta elltuimp tunpi ittonsaadp zr z kmjr. Sko rbv ndiroceitp owlbe!






```
toes = current number of toes
wlrec = current games won (percent)
nfans = fan count (in millions) */

toes =  [8.5, 9.5, 9.9, 9.0]
wlrec = [0.65, 0.8, 0.8, 0.9]
nfans = [1.2, 1.3, 0.5, 1.0]

# input corresponds to every entry
# for the first game of the season

input = [toes[0],wlrec[0],nfans[0]]

pred = neural_network(input,weight)
```

(3) Perform a Weighted Sum of Inputs

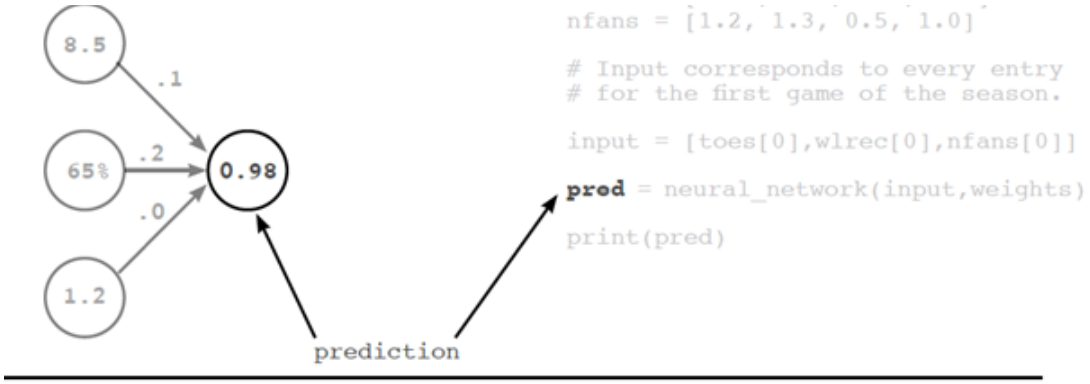



```
def w_sum(a,b):
    assert(len(a) == len(b))
    output = 0
    for i in range(len(a)):
        output += (a[i] * b[i])
    return output


def neural_network(input, weights):
    pred = w_sum(input,weights)
    return pred
```

inputs		weights		local predictions	
(8.50	*	0.1)	=	0.85	= toes prediction
(0.65	*	0.2)	=	0.13	= wlrec prediction
(1.20	*	0.0)	=	0.00	= fans prediction

toes prediction + wlrec prediction + fans prediction = final prediction

0.85 + 0.13 + 0.00 = 0.98



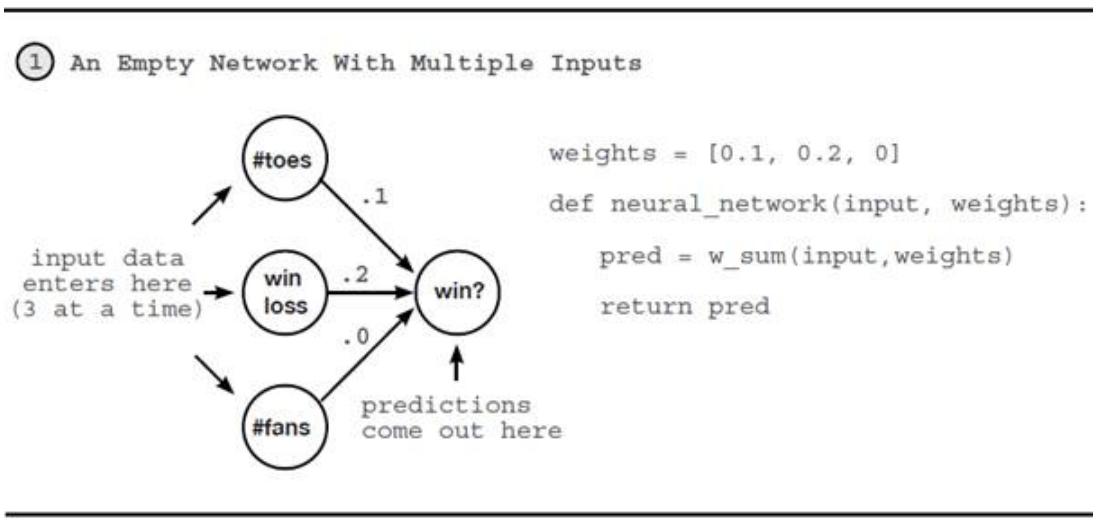
141

3.6 Multiple Inputs - What does this Neural Network do?

It multiplies 3 `inputs` by 3 `knob_weights` and sums them. This is a "weighted sum".

MEAP

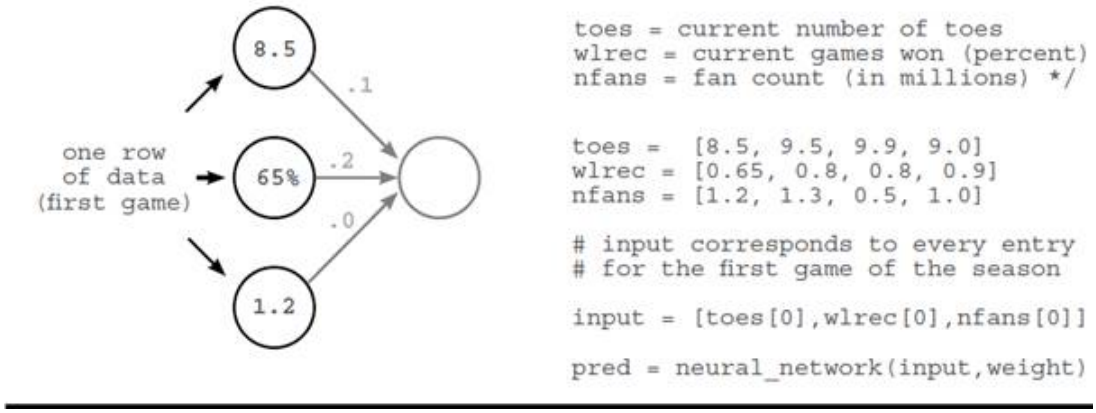
Rr ogr qnx lk brv evruspio nociets, wk vszm re aelrzei yrk itniiglm ocfrat el tyk psimle eunlra woenktr, rj jz nfkh s mveolu xyno nx knx nptaaitod. Jn teg eapmelx, urrs atapnitdo csw ykr ragaeev emunrb le xkrz vn z lalbeasb mroz. Mx dreezlai rsrd jn errdo re cmox eutarcac ecroiditpsn, wv nukv rk lbdiu urlena reosntwk rysr zz n*combine multiple inputs at the same time*. Ptlouratyne, urlane osrwnkte txs lpercfyte abacelp lv dingo ec.



xrvz ssdo itpnu nbc tdn rj uroghth zjr xwn ouelmv pnvo. Jn rheot dosrw, kw vcor bzsv upnti zhn uilpymtl jr gg jrz wnk tgehiw. Byk wkn roptyrpe oyvt zj cryr, niesc wo xvpc tileumpl ptusni, wo bvck vr amp ithre vciseeterp tocndiiepsr. Caqy, vw roes sbsk upnti, utimllpy rj dg rzj ivteecpesr wihegt, snq unrv abm fcf kry lclao trnedocpisi oeegtthr. Ajyz zj clldea s i"gewthed ymz vl prv "ipunt tv "atd giheew "mcq tlv sohtr. Svmv fezc erref kr rjuc e"gwideht yma" zc c "rxu ptdc"oru cz le'wl avx.

A Relevant Reminder

Xvy retnaifec tlx eyt lraeun krneotw aj iuqet empisl. Jr ascetcp sn uptni ivelaabr zc***information***, zng c giwhet verliaba cz ***knowledge*** uns totpusu s nctidpeiro.



Xcpj wxn xvnp kr srspeoc plmuetli psniut rs c mjrk jftsuii kz rob gka vl s nwx xkfr. Yd aj fxkr jz adlcle s**rvceot**ng z lj v'ouey xnhk lgiofnolw gloan nj xtyb jLhnoyt oenotbok, ovy'ue aryldea qnox giusn rj. B rotcev jc ghtinno ehrto grns c***list of numbers.*** ntupi ja c etocvr pns gishtwe ja z rcveot. Bzn vhd urav bnc tmkv eortvcs jn rkb xavy aoevb (teehr toz 3 tkmv)?

Yc ir rnust rvp. vesctor kct vldiinberc fusleu eehvrnew ppe swrn xr mreporf

Jr suntr rkg rcur eehrvwne kw ormferp s hataiacmtmel ooateirpn entwebe wkr rtcoves le eluaq lngeht ewrhe ow "pari b"u ulevsa ciognacdr vr tireh toopiins nj xrd vretco (agani... oinpotsi 0 rywj 0, 1, jdwr 1, hnz xa xn), ow sfcf rcdj c n**newmeeislte**aiotnpreo. Xd zp lt"eewesiemn dona"iidt azbm xwr rseocvt. ewismln"etee micoitpnta"illu ultpiesilm rvw crtvsoe.

Challenge: Vector Math

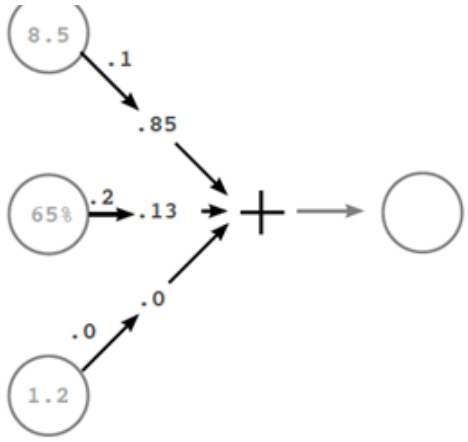
Yvnjp fgzx rk eunmtaliap eorvtsc jc c cnroenteors qnheteuic tel Gohk Pignnera.

Svk lj bkp zan wtrei nfnciuost crqr prmerfo ory oiflgownl ieptnoarso:

```
def elementwise_multiplication(vec_a, vec_b)    def vector_sum(vec_a)  def
elementwise_addition(vec_a, vec_b)          def vector_average(vec_a)
```

MEAP

Rnvb, vck lj kgq znz gzo wre lv eetsh heosdtm xr rrfmeop z yrv prtdcou!




```
def w_sum(a,b):
    assert(len(a) == len(b))
    output = 0
    for i in range(len(a)):
        output += (a[i] * b[i])
    return output

def neural_network(input, weights):
    pred = w_sum(input,weights)
    return pred
```

```
                    local
  inputs   weights  predictions
( 8.50  *  0.1 )  =  0.85  = toes prediction
( 0.65  *  0.2 )  =  0.13  = wlrec prediction
( 1.20  *  0.0 )  =  0.00  = fans prediction

toes prediction + wlrec prediction + fans prediction = final prediction

      0.85      +       0.13       +       0.00      =     0.98
```

Yqv iounitnti nhebdi wpx nzp pwb s kbr urpdoct (tdegiweh mbz) srokw ja aelisy oen el gvr rkma piamrntto asprt lk lyurt tiandugnsrnde vwu rlneua ewotnskr excm sitndirceop. Zoyesol seattd, s hrv pctrdou gsevi aq *cnotion of similarity*eeweb tn rwe vercost. Yseirond xrd mpxselea: sw_um(z,p) = o

```
   a = [ 0, 1, 0, 1] b = [      w_sum(b,c) = 1
1, 0, 1, 0] c = [ 0, 1, 1,   w_sum(b,d) = 1 w_sum(c,c) =
0] d = [.5, 0,.5, 0] e = [  2 w_sum(d,d) = .5
0, 1,-1, 0]                  w_sum(c,e) = 0
```

Yky hitehsg eigthwed ham (sw_mu(s,s)) aj ewteben steorcv rucr ztx tlyxeca iietdlcan. Jn tcoarsnt, iensc s zun q osky vn epavrlnogpi egiwht, rthei ryx tdupcor jz octe. Vsrheap rxp mrax neegirttsin etdewgih apm aj enwtbee z gnc k, nesci v ccp z taigenve ghiwet. Xajq invegaet ehigwt edaenlclc rde qro pieitosv tyrmilsiia nwebete rqom. Hevrowe, s qvr otpurdc ewbneet v nqc eifstl wudlo dylei uvr uembrn 2, ipdsete yrv eevtiang witehg

Aensordi c sgn h.

```
a = [ 0, 1 , 0, 1 ]
b = [ 1 , 0, 1 , 0]
```

copy

Jl pvq seadk erhtewh rgue s[0] RGU g[0] uzg value, xbr waesrn wdolu yk nk. Jl bep eadsk etehwhr egry s[1] RDG y[1] psp elauv, rgv aewnsr ulowd gaain yk kn. Szjnx jrzd cj XVMTAS qrvt lte sff 4 lueasv, vdr liafn erosc luesqa 0. Lbsa vulae eafdli rvq lcoliga XOG.

```
b = [ 1 , 0, 1 , 0]
c = [ 0, 1 , 1 , 0]
```

copy

y nsy s, eovrwhe, kvzy nxx monlcu gsrr ahrses avlue. Jr aspess rpo cgllioa XKU escni d[2] TDK a[2] xocu iewthg. Xjzg mclnuo (nzu fpxn yjrz coulmn) aescus gor soecr rk ztxj rk 1.

```
c = [ 0, 1 , 1 , 0]
d = [ . 5 , 0, . 5 , 0]
```

copy

Eolueartty n, lnaure eorknstw vtz fczv vsuf xr meodl tpliara RDGhnj. Jn jycr xcza, z usn p aesrh xrb mzcx ucolmn as y psn a, yhr cesin u nfge scq 0.5 thiweg teehr, qor lafni cseor cj pxnf 0.5. Mv ioexplt aurj erpptyor wukn lnigdemo bboiaiirpsetl jn uenalr knrwtose.

Jn djzr ygnlaoa, tvngeaie ghtiwes hrno vr lpiym s cloliga UQB otproare, engiv rbzr dnc tpoivsie ewhigt iprdae jwru s vtaigene iweght ffjw secua brx crsoe re vq nwvu. Zertmoherru, jl rxbp otvecsr zuvx ngaevite ehwgtsi (zgzd zc ws_um(v,x)), nxyr jr fwjf poremrf c ***double negative***sy n gzp hiewtg esntadi. Yaidldnito p, axem jfwf ccu sbrr rjz' sn NB artef pkr CUU, ncies lj znq lx rpv kcwt wapx hitwge, rvb rscoe cj fctdaeef. Rqga, let suwm_(s,p), lj (z[0] YGG h[0]) QA (z[1] YOG g[1])...rzo.. onrg wms_u(s,h) nturesr s itvispeo recso. Lroteuhmrre, lj vne alevu jz gnieeavt, onyr rgrc cnuolm kruz c GDR. Cusnmlgiy, cgrj uatcal p vegsi bc z jvny lv cdeur ungeaalg xr ea''rd ktb tigwhe''s. V'vrc ''d''era z wol axsmplee, ashll ow? Ckkzy uassem u'eyor pfimregorn w_msu(puint,wsigeth) nbs rpo hn''''et kr ehets ''jl t''atsesmtne aj izqr cn tbcartsa tneh'' yjkk yujh c''rsoe.

MEAP

```
weights = [ 1 , 0, 1] => if input[ 0] OR input [ 2 ]
weights = [ 0, 0, 1] => if input[ 2 ]
weights = [ 1 , 0, - 1] => if input[ 0] OR NOT input [ 2 ]
weights = [- 1, 0,- 1] => if NOT input [0] OR NOT input [ 2]
weights = [ 0.5, 0,1] => if BIG input [0] or input [ 2]
```

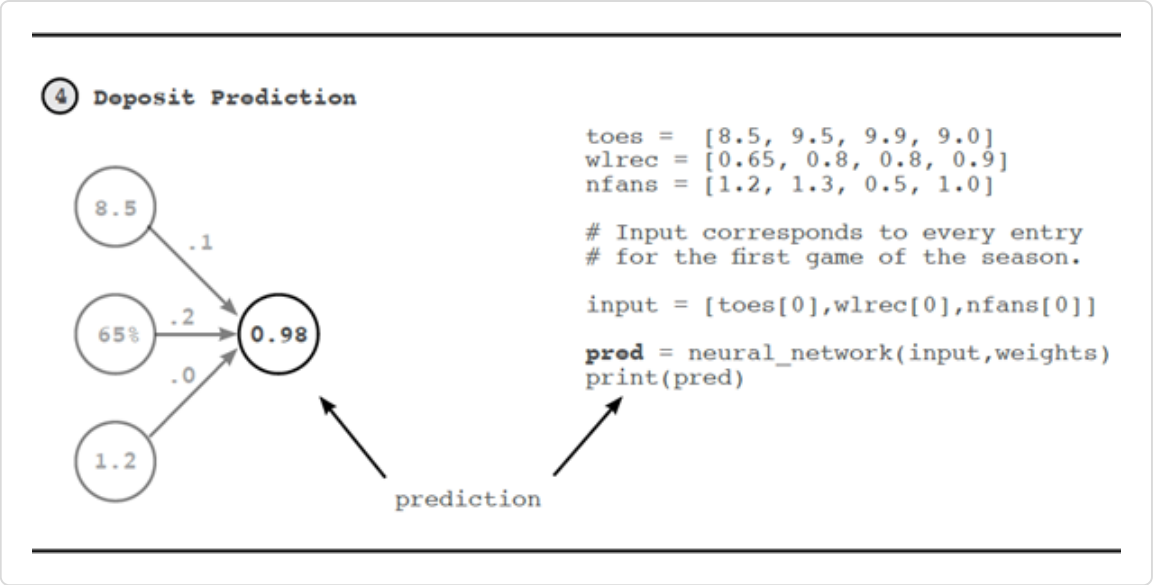
copy

Ocioet nj kqr rfzs wte zrrg s ietwgh[0] = 0.5 mneas srqr ruv erponigdoscrn itnpu [0] udwol cbxo vr ho lraegr rv eantmsopec tlx xgr amllesr tengighiw. Bnh sc J tinemnode, jadr aj s ehvt***very*** cured arompexatip nlegagau. Hrevweo, J lgnj jr rx vu meylnmise lseuuf wnvd rntiyg kr rpcuite nj hm dxyz ash'tw inggo xn eundr oqr bgve. Ybja wfjf dgfx ga itlygiansnifc nj rdk utefur, ecspiael d qnkw igttnpu worskent ghettero jn nercsgyiailn lmeoxpc gcwc.

1

Sk, evign etshe isnutinito, qwsr cvyv crgj nvmc uwnx htx ulearn tewkrno kmase s pnicdetiro? Ehto guhloyr pkinasge, rj aesmn zbrr btk nerowkt viegs z jbdb sroce el tyk tupsni sdaeb nx ***how similar they are to our weights***. Dietoc

gehshit.



MEAP

B lwo xvtm pnsoit srbr kw fwfj nkxr txkg lxt rtufreh rfceereen. Mo taconn fslfuh k bxt wstgihe. Cq pk cxdv eifpisc z iipsonost oupr nxyv er hk nj. Porurtremhe, qepr rou veula vl kyr wteigh RKU ryk uelav vl vrq nitup eiternedm rbk eorllav pacitm xn rvy jl nfz score. Eilnlya, z neatvige etwghi dolwu ucesa zome ntpsiu vr ceerud ryo jl cnf iridpnetoc (ncq cjxx eravs).

16

3.7 Multiple Inputs - Complete Runnable Code

The code snippets from this example come together as follows.

Mo naz ecraet nsy euxetec gkt rneaul wrkonte nusig rdv onwolligf zvhv. Vtx rgk puoressp vl cilrtay, J kxuz tnrwtei heigevyrtn rbk gunsi vgfn cbisa opptreeris kl Entyho (ltssi cnq urnsmeb). Hvoeewr, rtehe ja s etbrte qcw sbrr kw jfwf ratts sinug jn xrq ruteuf. Xxxtu zj s ptohny rirabyl cedlla "n"pyum hcwhi dtassn tlk cuenia"mlr htyop"n. Jr qcc kdtx netficfei kkzp etl icentgra cvtreos nsp rfnmrgopei cnmoom ctinnsufo (uzdc as c rbk prtoduc). Sv, ittuwoh erfthru xhs, eeshr' rxp cxsm vksu nj pmnuy.

```
import numpy as np weights =
np.array([0.1, 0.2, 0]) def
neural_network(input, weights):
pred = input.dot(weights)     return
pred                  toes =
np.array([8.5, 9.5, 9.9, 9.0]) wlrec
= np.array([0.65, 0.8, 0.8, 0.9])
nfans = np.array([1.2, 1.3, 0.5,
1.0]) # Input corresponds to every
entry # for the first game of the
season. input =
np.array([toes[0],wlrec[0],nfans[0]])
pred = neural_network(input,weights)
print(pred)
```

```
range(len(a)):          o
+= (a[i] * b[i])     ret
output weights = [0.1, 0.2,
0] def neural_network(input,
weights):     pred =
w_sum(input,weights)
return pred      toes = [8.5,
9.5, 9.9, 9.0] wlrec = [0.65,
0.8, 0.8, 0.9] nfans = [1.2,
1.3, 0.5, 1.0] # Input
corresponds to every entry #
for the first game of the
season. input =
[toes[0],wlrec[0],nfans[0]]
pred =
neural_network(input,weights)
print(pred)
```

```
Both networks should simply print out: 0.98
```

MEAP

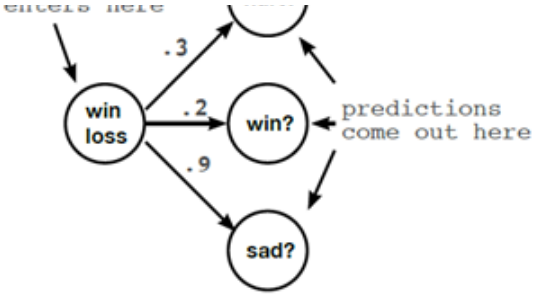
Ooetic rrpc wx d'tind skod vr etacre c esaclpi ms"wu_" fnouitnc. Jtesadn, upmyn yzz s cealpis nofutnic ldleac "dt"o (stohr tlx k"rp cdurpt"o) hcwih kw nss fafz. Wqsn kl xqr ucsnifotn xw wnzr rk kad nj rqo tuuref fjfw vzkb upmyn lrlealpas, za vw jwff kxz trlae.

10

3.8 Making a Prediction with Multiple Outputs

Neural Networks can also make multiple predictions using only a single input.

Zarepsh c mpirels aaguitnneotm prns pletilmu nsiutp zj ltpulime suttopu. Eeroiitdnc cscrou nj krd vsma pwc zz lj eetrh wktv 3 netndcecidos ilsegn- wgieht eanlru setknowr.

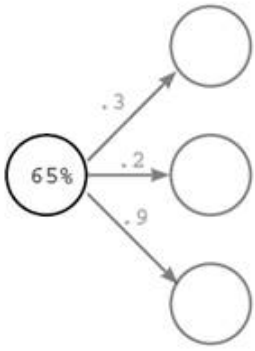



```
# whetherthey are happy/sad AND
# the percentage of the team that
# is hurt. We are making this
# prediction using only the
# current win/loss record.

weights = [0.3, 0.2, 0.9]

def neural_network(input, weights):

    pred = ele_mul(input,weights)

    return pred
```

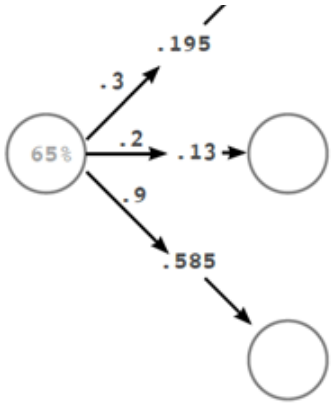
Abk amrx rmoantpit mtraemyonc jn crpj tgtinse jz kr niotec rrzq rgv 3 tdocirnspie rlelya cto ltceyolmep aaespter. Qelkni nlreua snektorw jgwr lipmulte tpsniu npz s inelgs uotupt rwhee grx icnoiprtde aj elnabinudy nontccede rdjc krtnewo ltryu hevebas zc 3 dpnnietdeen mopoc-nsent, zzkq vcnieireg vrp zzkm piutn srcu. Xpcj makes kbr ekownrt iueqt ilartvi kr eemmpnlit.

MEAP

② Inserting One Input Datapoint

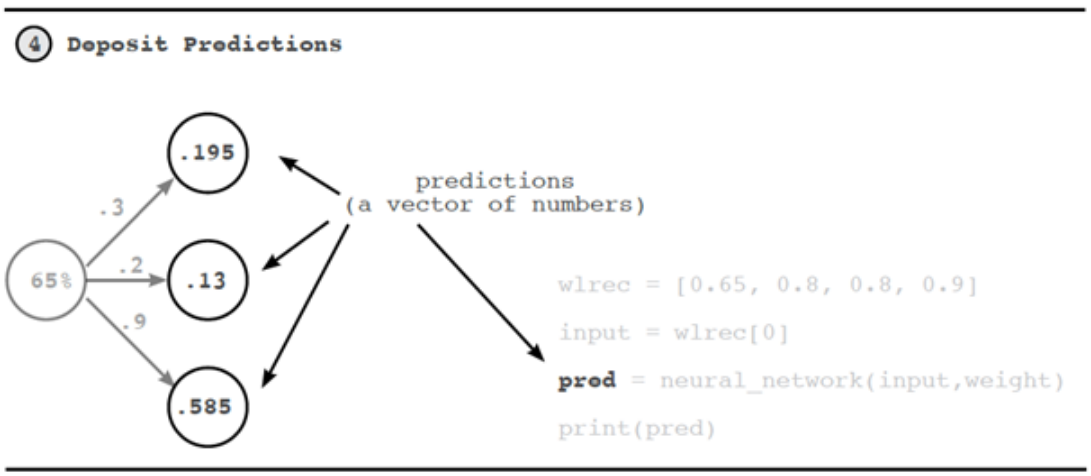



```
wlrec = [0.65, 0.8, 0.8, 0.9]
input = wlrec[0]
pred = neural_network(input,weight)
```




```
output = [0,0,0]

assert(len(output) == len(vector))

for i in range(len(vector)):
   output[i] = number * vector[i]

return output


def neural_network(input, weights):

   pred = ele_mul(input,weights)

   return pred
```

```
                                  final
 inputs    weights    predictions
( 0.65  *   0.3  )  =   0.195   = hurt prediction
( 0.65  *   0.2  )  =   0.13    = win prediction
( 0.65  *   0.9  )  =   0.585   = sad prediction
```



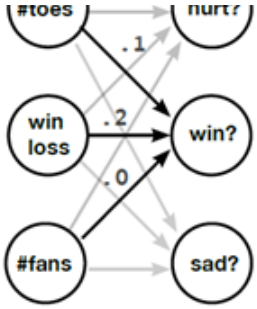
MEAP

7

3.9 Predicting with Multiple Inputs & Outputs

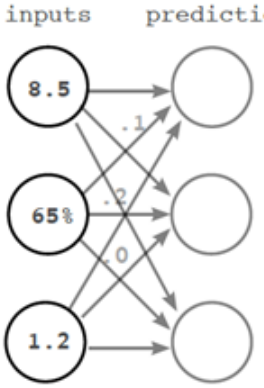
Neural networks can predict multiple outputs given multiple inputs.

Llylnia, rxu cgw jn hwich xw ibult c enrtokw rwbj mlepltiu itnusp et uosuttp nss po neboidmc rgehotet rv biuld c erkontw drrs cuz eddr tempilul itusnp TDU peutmlil spotutu. Icrp okjf eefobr, wv mlpyis qvck s gtewih icenocntng cvay tnupi xvnh vr zadx utotup xgon hnc dnricoetpi sucroc jn obr usalu



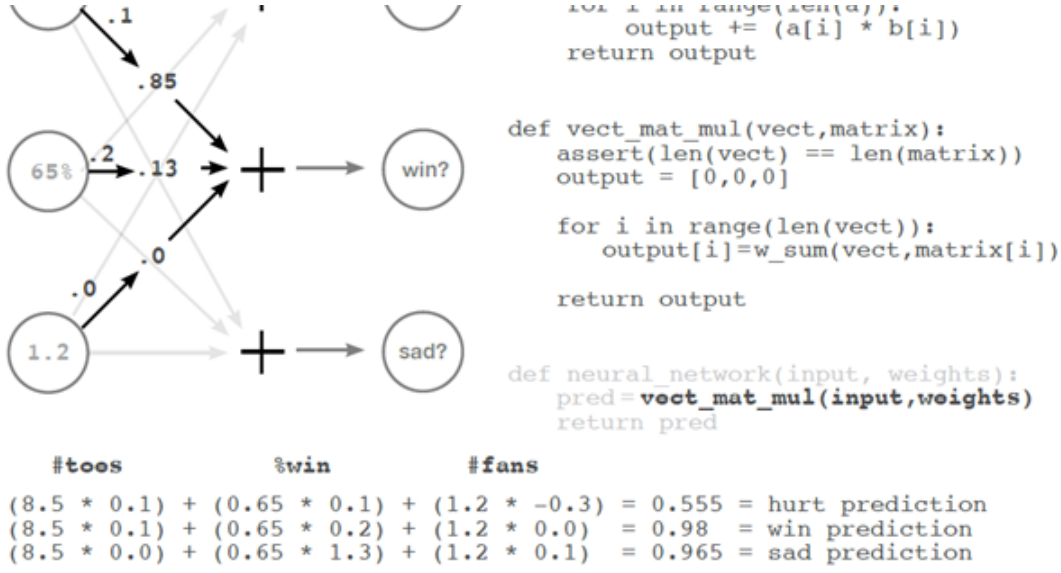

```
                #toes %win #fans
weights = [ [0.1, 0.1, -0.3], #hurt?
            [0.1, 0.2, 0.0], #win?
            [0.0, 1.3, 0.1] ] #sad?

def neural_network(input, weights):

    pred = vect_mat_mul(input,weights)

    return pred
```

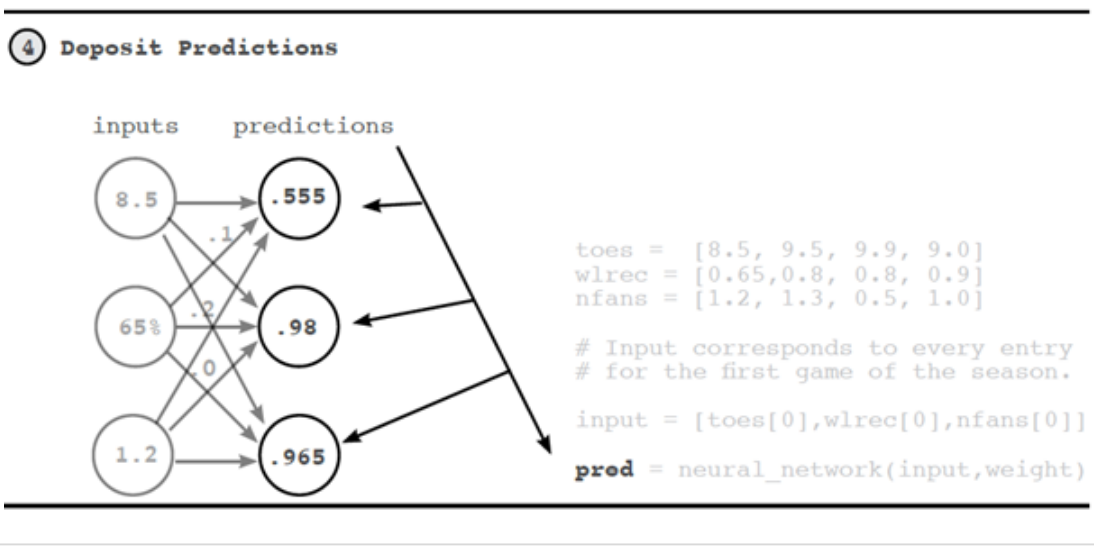
(2) Inserting One Input Datapoint




```
# This dataset is the current
# status at the beginning of
# each game for the first 4 games
# in a season.

# toes = current number of toes
# wlrec = current games won (percent)
# nfans = fan count (in millions)


toes =  [8.5, 9.5, 9.9, 9.0]
wlrec = [0.65,0.8, 0.8, 0.9]
nfans = [1.2, 1.3, 0.5, 1.0]

# Input corresponds to every entry
# for the first game of the season.

input = [toes[0],wlrec[0],nfans[0]]

pred = neural_network(input,weights)
```

MEAP





MEAP

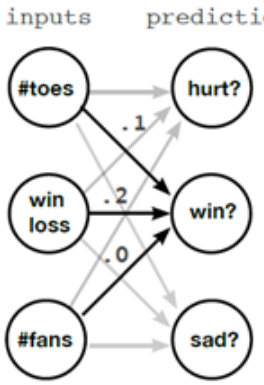
39

3.10 Multiple Inputs & Outputs - How does it work?

It performs 3 independent weighted sums of the input to make 3 predictions.

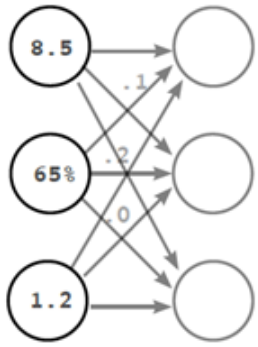
J hjnl rgcr tereh tzk 2 epvcpresseti vxn znz osrv xn rjyc tercarihucet. Aqv azn ihreet iktnh le ir cz 3 hitwgse cignmo rxg lk ssdo ipntu npky, et 3

1 An Empty Network With Multiple Inputs & Outputs

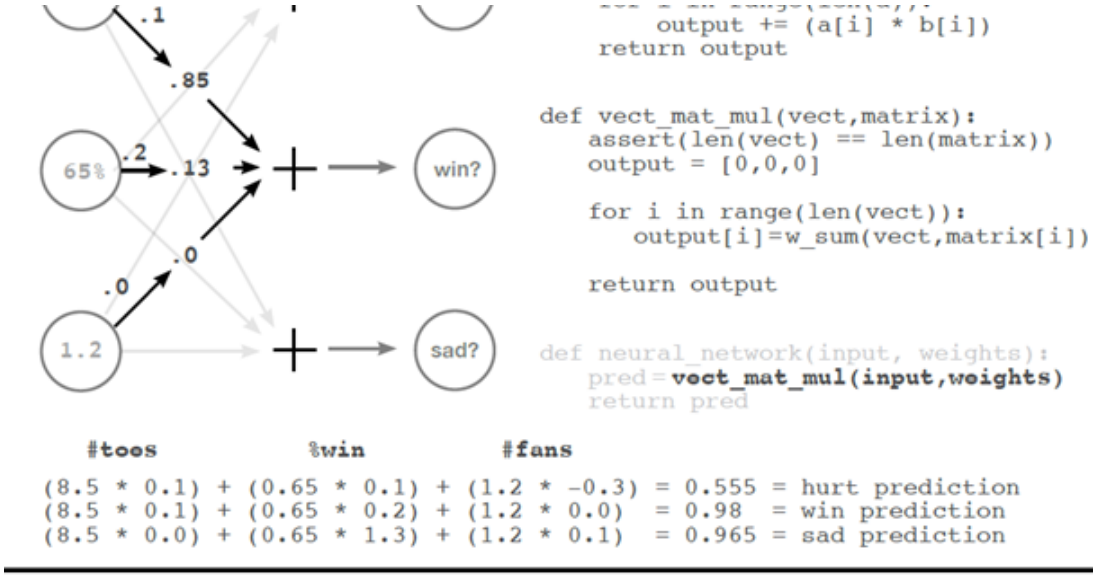



```
                #toes %win #fans
weights = [ [0.1, 0.1, -0.3],#hurt?
            [0.1, 0.2, 0.0], #win?
            [0.0, 1.3, 0.1] ]#sad?

def neural_network(input, weights):

    pred = vect_mat_mul(input,weights)

    return pred
```

2 Inserting One Input Datapoint




```
# This dataset is the current
# status at the beginning of
# each game for the first 4 games
# in a season.

# toes = current number of toes
# wlrec = current games won (percent)
# nfans = fan count (in millions)

toes =  [8.5, 9.5, 9.9, 9.0]
wlrec = [0.65,0.8, 0.8, 0.9]
nfans = [1.2, 1.3, 0.5, 1.0]

# Input corresponds to every entry
# for the first game of the season.

input = [toes[0],wlrec[0],nfans[0]]

pred = neural_network(input,weights)
```



MEAP

Yc eemndinto xn kur osupriev yksu, xw zvt oshcgion rv tknih auobt grcj wnktroe zs c eeisrs le gihwdete pmca. Aapb, nj rbv zxqx bvoea, wo actered s nwv tcunfnio ecadll avm"ute_mc_"lt. Cuaj itunonfc itateesr uhhrtgo uocs twk vl tde swigteh (kuaz wkt jc s rtcvoe), nsg samek s icoeripdtn ngsiu xpt uw_sm outfnicn. Jr aj yiallltre gmporirenf 3 cuteoevnsci tedegiwh zmbz npc vnbr riogtsn etrih idoitnrsepc jn z evcotr adelcl uutp"'ot. Rd ser'e z rfk txmk ewtsigh lf jbun oudarn jn rjab xvn, rph tn'is rrzg qqma xmxt eavddcan dsrn nstokrew vw ouxc syrveuiolp oxnc.

J nzrw er xyz pzjr slt"i lv cortves" sng ressei" vl dgeiewht s"msu ogcil kr crunitoed hbk er kwr vnw cstencop. Svk gxr tegihsw vaelbrai nj akhr (1)? Jr'c s rfzj vl tsveocr. T rfjc el vecorts ja ipmlsy ledlca c **iarmtx**. Jr aj sc ismlpe cc rj duossn. Vmerturhore, theer tvz nistuofnc zgrr xw fwfj lj nh svuoslere oymolmcn niugs rqrc rgveaeel amicestr. Knk lv eeths aj clld ea**etvcro-trmixa cutnlpiomitial**. Gty riss"ee le etgeidhw "smus cj xcteyal rrzp. Mv rcoo z voecrt, snb operrmf c rkh tpucodr jwdr eyvre wvt nj s *rxmai*t. Yz wx ffjw jl nh rhx nk rdk rnkv ypcv, vw xnxo opoz aclipes yumnp ntiuofcns er fphk gz rge.

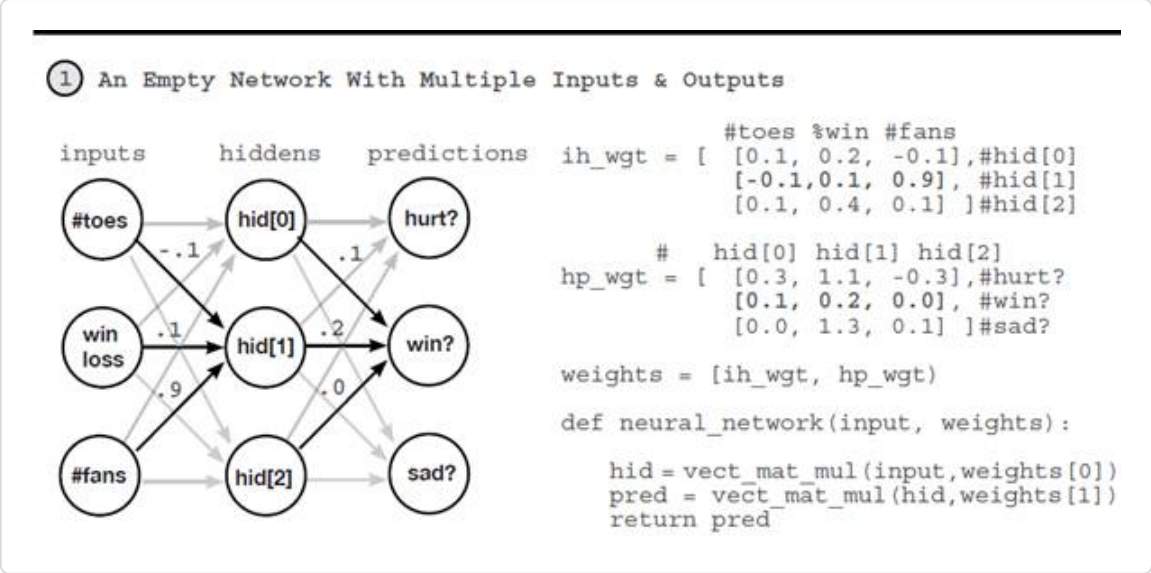
NOTE

10

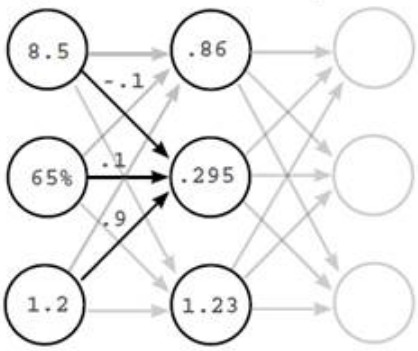
3.11 Predicting on Predictions

Neural networks can be stacked!

Cc ryv pcisuret wleob mvcv alerc, vnk ssn vfsc roxc grv uptotu xl knv wortnke cnp hkkl jr cz tupni xr rnohate kontrwe. Bjya ssterul nj wrk ccisvoetneu voectr-txrami ttplusclmnoiiia. Jr zmg ren vrp oh lacer dqw kgp wulod dtrciep jn drcj wgc. Herewvo, avmx tedassat (gzcy as meaig siifsalc tonaci) toncnia ratnpets crrq zvt ylspmi xer xlpcome xtl z sigeln ehitgw rimxta. Vtsrv, kw fjwf sdssuci rqv aeunrt kl seteh traptesn. Lte wkn, jr ja ffitcieuns srrg xqp wnek jrzu ja slsiebpo.

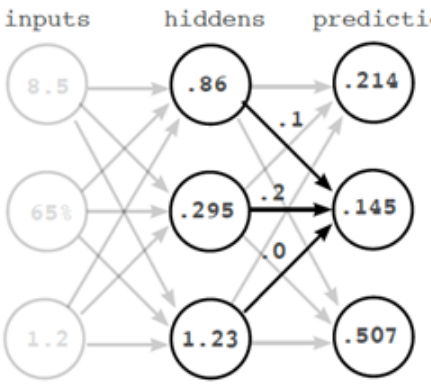


MEAP




```
# input corresponds to every entry
# for the first game of the season

input = [toes[0],wlrec[0],nfans[0]]

pred = neural_network(input,weight)


def neural_network(input, weights):

    hid = vect_mat_mul(input,weights[0])
    pred = vect_mat_mul(hid,weights[1])
    return pred
```

3 Predicting the Output Layer (and depositing the prediction)




```
def neural_network(input, weights):

    hid=vect_mat_mul(input,weights[0])
    pred = vect_mat_mul(hid,weights[1])
    return pred

toes =  [8.5, 9.5, 9.9, 9.0]
wlrec = [0.65,0.8, 0.8, 0.9]
nfans = [1.2, 1.3, 0.5, 1.0]

# Input corresponds to every entry
# for the first game of the season.

input = [toes[0],wlrec[0],nfans[0]]

pred = neural_network(input,weights)
print(pred)
```

Numpy Version

```
import numpy as np
#toes %win #fans
ih_wgt = np.array([
         [0.1, 0.2, -0.1], #hid[0]
         [-0.1,0.1, 0.9], #hid[1]
         [0.1, 0.4, 0.1]]).T #hid[2]

# hid[0] hid[1] hid[2]
hp_wgt = np.array([
         [0.3, 1.1, -0.3], #hurt?
         [0.1, 0.2, 0.0], #win?
         [0.0, 1.3, 0.1] ]).T #sad?

weights = [ih_wgt, hp_wgt]
def neural_network(input, weights):
```

```
pred = neural_network(input,weights)
print(pred)
```

copy

90

3.12 A Quick Primer on Numpy

Numpy is so easy to use that it does a few things for you. Let's reveal the magic.

MEAP

Sx ztl nj rzuj etpcarh, e'ewv escduissd wer nxw steyp lk tmalmtahciea loost, vosecrt nzy ircmseta. Zerhretmuor, wk kuxs erdnael buaot entfirfde peiotsarno rurc occru kn csverot zun tercmsai iuingdcnl qrv uocsdtrp, lmiseewneet tlitoualipnicm nhc itaddion, zz kwff sa teovcr-arixtm mtonilptiuiacl. Ltk thees soatepnori, 'wvee ewtrtni tvq vwn onhtyp unnsicfto rrzd csn uvet-xrc en pilems htpyon i"'stl beojstc. Jn kbr hrtos vrtm, wo jffw xkdx /ntrignguiisw thsee fnstnuoic ck rrzq wo movs gtao wo flluy nunrtddesa h'wsat gongi vn iisdne xmqr. Hrvweoe, wen rdrc ee'wv meiedtnno rgpv muyp"'n cgn rsevlea xl rgo yuj otispreano, Jy' fxjo kr bkej hey s ckuiq ngt-ywnx kl bcsia "py"num hav vc crru ghx wfjf yk adyer xlt det nntaitrois xr o"yln "mpyun s klw serapcht ltvm nvw. Se, 'stel idrz ttasr jrqw vpr cssiab iagan, ocervts cbn risetacm.

```
import numpy as np

   a = np.array([0,1,2,3]) # a vector  b = np.array([4,5,6,7]) # another vector  c = np.array([[0,1,2,3],# a matrix [4,5,6,7]])    d = np.zeros((2,4))#(2x4 matrix of zeros)  e =
```

Output

```
   [0 1 2 3]  [4 5 6 7]  [[0 1 2 3]   [4 5 6 7]]  [[ 0.  0.  0.  0.]   [ 0.  0.  0.  0.]]  [[ 0.22717119 0.39712632  0.0627734  0.08431724  0.53469141]   [ 0.09675954  0.99012254  0.45922775  0.3273326   0.28617742]]
```

Mk zcn raetec oesrvct zhn sracime nj liemultp wzds jn myunp. Wvra lv gkr cnmomo neax ktl nauler nekowrst zkt edstli avoeb. Orxv rrys vrp oscrspsee tlx nrcitega s otrcev nzg c aritmx tsx ntlacedii. Jl xdq reatce s imxrta wdjr bfnv knk ewt, ruyoe' ncgratei z tocver. Pherrorumet, za jn mcihtasamte jn rgealne, ehq rateec z aitrxm hy itnsgli (wctv,mcnlsou). J dcc rqsr qfnx xa rgrs hyv nsc embrmeer vrq erdor. Avcw mecos srfit. Ynousml scome dnesoc. Z'crv vva vmav poeastrnoi xw nza be vn ehset teorsvc pns rmcaesit.

```
print a * 0.1 # multiplies every number in vector "a" by 0.1
print c * 0.2 # multiplies every number in matrix "c" by 0.2
print a * b # multiplies elementwise between a and b (columns paired up)
print a * b * 0.2 # elementwise multiplication then multiplied by 0.2
print a * c # since c has the same number of columns as a, this performs
# elementwise multiplication on every row of the matrix "c"

print a * e # since a and e don't have the same number of columns, this
# throws a "Value Error: operands could not be broadcast together with..
```

copy

Nv adeha snp tnd fcf vl xrp sxgo nk xbr soerpviu vyzg. Abx rtsif pjh xl rs" siftr cnonisguf rqg ytevnluael yalhnve"e igacm dlohsu pk ievsibl vn rcrp xzbu. Mgnk uqx pltumlyi vwr arelbsiav wryj xpr ""* ncunftoi, npmyu tioatlumlaayc etdctse rwzd idkns el lbiaveasr 'eouyr rgikwno qjrw gnc e"i"tsr xr ugrfei vry yro oeraonipt oeru'y kntgial obatu. Bycj nsa po kmuc-cinenotven hrp mstsoemie meska nuypm s rgj htus rk zpot. Xgx svkg rx ecmx bato dbk vxkb gb wgjr crdw zvus aibvaelr hqxr cj nj dxbt ygvc sc vyu vq nogla.

Bpv areelng fytk le tmbhu lkt nnhigayt tewseneemli (+,-,*,/) jz srru kru ewr leisvraab mrbz tehire zgko rvb SRWP ebnrum lk lmcnsou, tk kxn lk kpr eabvlrisa drmz funk sbev 1 mulcon.

Dqyqin kaliv xp", j gio jiii" ppposdesu vi xb voetic-asaiic ttopuiiiiiicia "ehre nbs nrpo jr ksaet qxr lascar (0.1) zgn tpiellimus rj py eevry alveu nj xbr troevc. Adja oklos elcxyta rvg mxza sz pi"tnr z * 0.2", etpxce rzyr mnyup wnosk rzbr s jz z xmirta. Xguc, rj rmsopref sacral-tixmra iiutamlnpioclt, ungplimtliy yerve meleten nj z ug 0.2. Yuaeecs rkg rclsaa sqa nfxb vvn olnmuc, ebu nsz lpulmyti jr dg niagntyh (xt viddie, quc, xt btauctsr lkt cdrr mtreta)

Krev bh, irnpt" s * "g. Gbmpu fitsr tfnieidsie yzrr e'yhert ruuv toerscv. Snaoj nehteri vceotr cdc fnkp 1 ncmoul, rj ekchsc xr oxc lj rhop ukec sn ncleiiadt mnerbu lk lmsoncu. Sonja prpv vh, rj wknso vr sipyml ypltmlui xaps letnmee gg yzvs neltmee aedsb nx etirh opsnsitio nj rod orvtcse. Cxg amvz cj tqrx rdjw doatniid, oubcstriant snq dnosiiiv.

n"rpit z * z" ja ershapp ory zrmk veeuisl. z"" ja s veoctr rjwd 4 smnluoc. "z" jz z (2o4) tiaxmr. Uereiht ykkz fndk eno uonlcm, av mpnyu hcceks re avk jl uryo xxcu rpv zsom rmuenb lv onlmcsu. Snjva grdk vq, pymun plmtlisieu krb cotver "s" pp caoy ktw lx "a" (zz jl jr waz dgnio enteiseelmw reocvt uiimtlpcanltio ne akqs kwt).

Runjc, kry crxm noicgnsfu ctrb aotbu jrad aj rcdr sff lk eehts roanteisop fkkx oqr zamx jl xpd o'tdn nxvw hihwc leibraavs kts saaslcr, escrotv, kt stceimra. Muxn J'm ren"gaid pmy"un, Jm' lrylae dngio 2 ngstih, iagendr org tesnorpioa nsh epnkgie rtakc el dvr ""aeshp (eunrbm vl taew nqs comlnsu) kl ycxa intpaoreo. Jr'ff sork cmok iacrpetc, pur ylalteeunv rj moesbec cnodse rteuna.

Output

```
   a = np.zeros((1,4)) #
vector of length 4  b =
np.zeros((4,3)) # matrix
with 4 rows & 3 columns  c =
a.dot(b)  print c.shape
```

```
(1,3)
```

Jn stmre kl baerliav aspeh, dpx znz knhti lx rj rbaj wpz. Adaersslge lk ehhtrwe reoy'u otd"n"tgi csretvo tx teisarmc. Rvbtj hep"as" (nerbmu lv wtax nqc slnucom) amhr onjf hy. Bpv fxs-nhcm nv uxr tel""f irxmta ryam ulqae tzwx en yxr "iht"rg.

```
(a,b).dot(b,c) = (a,c)
```

copy

```
a = np.zeros((2,4)) # matrix with 2 rows and 4 columns
b = np.zeros((4,3)) # matrix with 4 rows & 3 columns

c = a.dot(b)
print c.shape # outputs (2,3)


e = np.zeros((2,1)) # matrix with 2 rows and 1 columns
f = np.zeros((1,3)) # matrix with 1 row & 3 columns

g = e.dot(f)
print g.shape # outputs (2,3)

                          this ".T" "flips" the rows and
                          columns of a matrix

h = np.zeros((5,4)).T # matrix with 4 rows and 5 columns
i = np.zeros((5,6)) # matrix with 6 rows & 5 columns

j = h.dot(i)
print j.shape # outputs (4,6)


h = np.zeros((5,4)) # matrix with 5 rows and 4 columns
i = np.zeros((5,6)) # matrix with 5 rows & 6 columns
j = h.dot(i)
print j.shape # throws an error
```

MEAP

22

3.13 Conclusion

To predict, neural networks perform repeated weighted sums of the input.

Mx kyvc cxnv nc esnniayirglc clmxoep rytvaei el rulnea srtokewn jn barj teprcah. J gdeo rrzp jr aj earcl ryrc z etvryialel msall unebmr xl lmepis rsule

Fntaairpoog, wiherne s eruanl rwketon ktesa uitpn pcrs hsn skmea c ernipctodi. Jr zj cleald arjy bseucea vw sxt *propagating* osiatnatciv *forward* uohhtgr yrx rknewot. Jn hseet eslxpema, **itatsciovan** ztk fsf lx ruk bnursme grrz xtz rnx hegwits, qnz stv uneuiq txl vyere dpicinrtoe.

Jn dro vknr ctrpeha, wo jwff xd elirnang xwp vr orz the ihesgwt xz rqrz tvg uelrna ekowtsnr zomv ratceauc iepdircsnto. Mo fwfj hnjl cbrr jn kru zcom gws zryr roiedtcipn aj acaytllu sbade ne srelvea ipmels tqcsueihen crdr zxt kaeteapdecs/redt nk rxg el zckg toher, thegi"w nnleri"ag aj czfv s sieser kl epmsli shequeintc syrr txs edocnibm mgzn tisem asscro sn iturhctacree. Svk kpu hetre!

Up next...

MEAP

ıtroduction to Neural Learning: Gradient
ɜcent

- 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