Deep Neural Networks

Goals

- Show why deep learning is important
- Show what it is, and what it can do
- Show how to use it yourself

Why is deep learning important?

- It solves 'ai-hard' problems easily
- No feature engineering
- Little handtuning
- Human like performance on important tasks

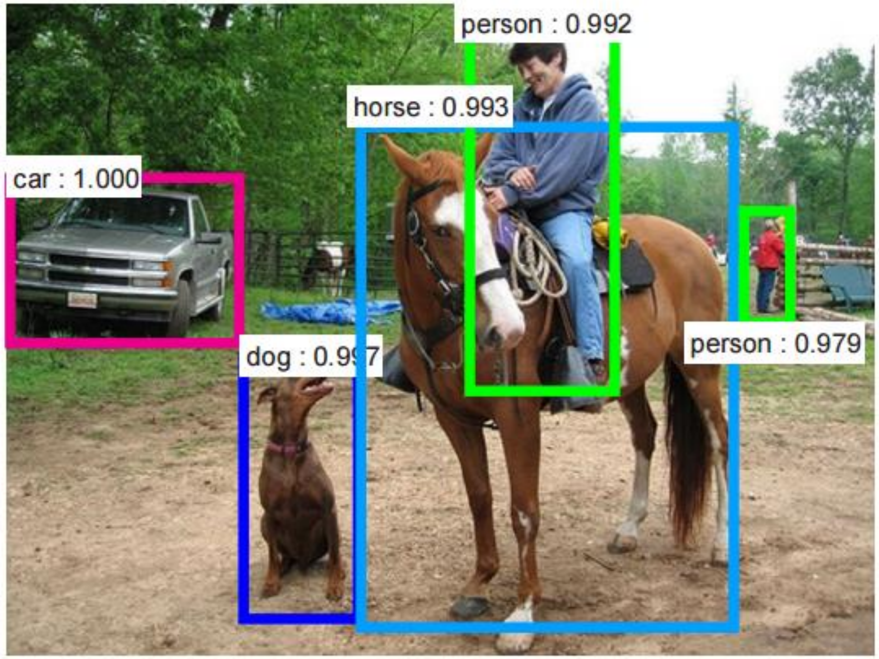
Deep Learning is Simple

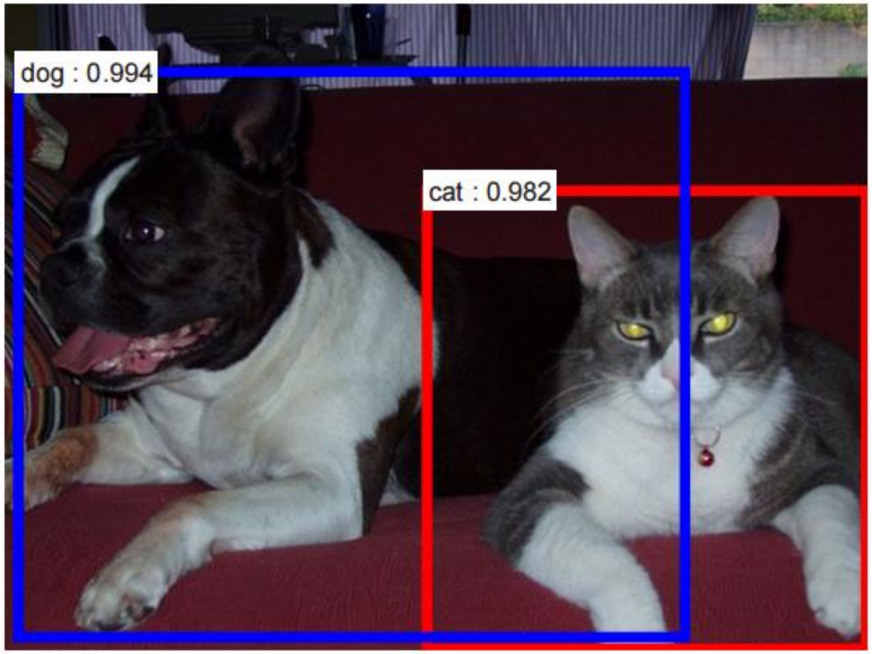
- Great, well documented libraries
- The math is straightforward
- Forgiving and very powerful
- However, it is fast moving

Motivating Examples

- Vision
- Natural Language
- Fun

Object Detection





Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN:
Towards Real-Time Object Detection with Region Proposal Networks, arXiv:1506.01497.

Semantic Segmentation

Jifeng Dai, Kaiming He, Jian Sun, BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation, arXiv:1503.01640

Image Captioning

https://github.com/karpathy/neuraltalk2

Image Captioning

a man riding a bike on a beach with a dog in the water

a man sitting at a table with a laptop

a street sign on a pole in front of a building

a plate with a sandwich and a salad

a black and white cat sitting in a bathroom sink

a little boy standing in a field with a kite

Visual Attention

Visual Attention Kelvin Xu*, Jimmy Lei Ba†, Ryan Kiros†, Kyunghyun Cho*, Aaron Courville*, Ruslan Salakhutdinov†, Richard Zemel†, Yoshua Bengio*

Question Answering

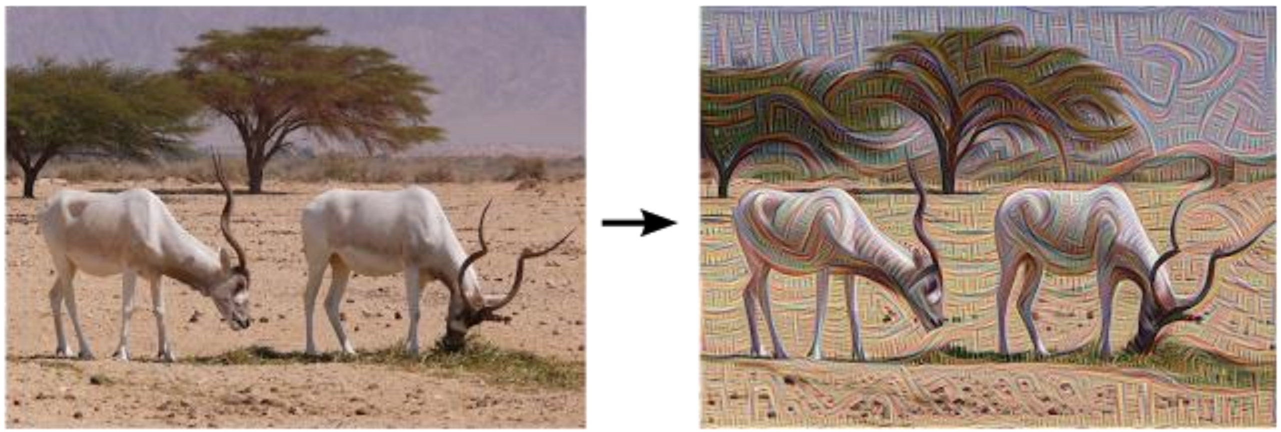
What kind of store is this?	bakery bakery pastry	art supplies grocery grocery
Is the display case as full as it could be?	no no no	no yes yes

How many bikes are there?	2 2 2	3 4 12
What number is the bus?	48 48 48	4 46 number 6

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, VQA: Visual Question Answering, CVPR, 2015 SUNw:Scene Understanding

Deep Dreams

Google over sampling a Convnet

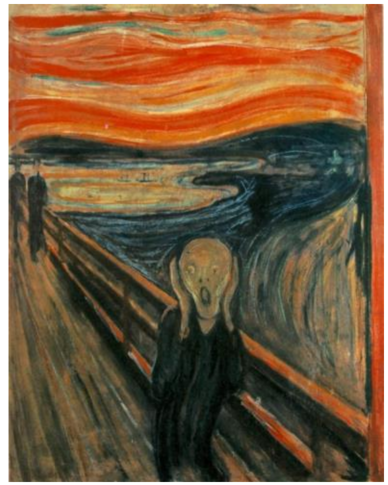


http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html

Video Captioning

https://vimeo.com/146492001

Style Shifting Images




Source Style Result

https://github.com/jcjohnson/neural-style

Style Shifting Images



What if we do it in reverse?



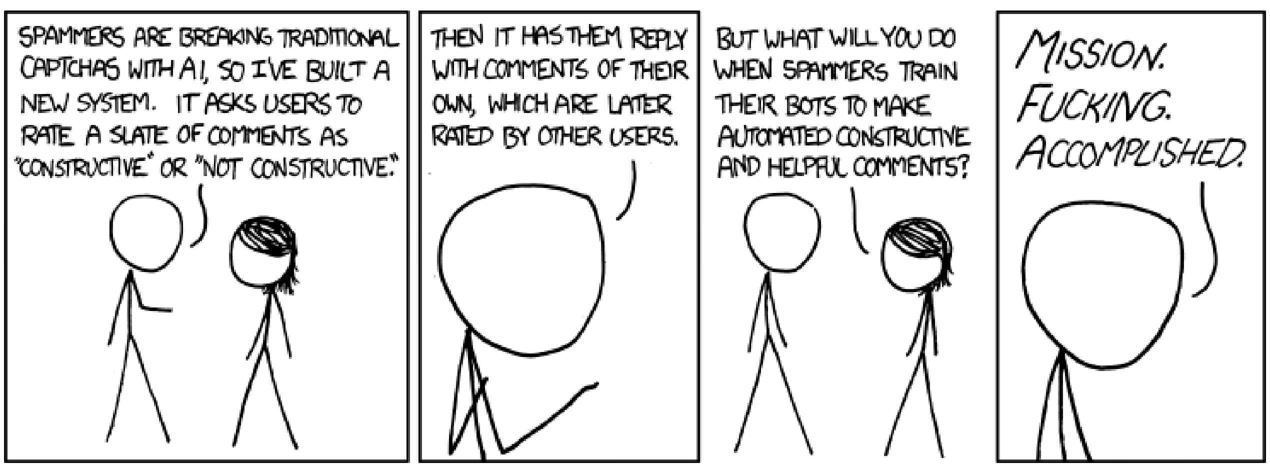
Source

Style

Result

Handwriting Recognition

- Notice how recaptcha no longer shows pictures of text?
- Deep neural networks first performed better than humans in 2011



https://xkcd.com/810/

Natural Language Generation

Shakespeare after .03 epochs
----
fiHXAnhnoeepwdnie bcaloTe? ndiol e hnhelNseooiaodUlaa::Lylo pd
e e e Steotnueeelh OSednst: eey:feio erriFhea
OadnnstDeemEislFmgs Sot: !eh, ft ltny
eiShe I on Senth.catssnac .Nachnhy
alo i
nt Mdo;aate
mh : tUem sLhrshbnRras ds aehfhhfO
tylta,hLefHtIyhCnhfraclos iysaws e
'teth!
Nrt;h e oeoe dnBsb rprInNymd metende e oihE tiirea t o haoe ns'ritt   e nc eshe e eraetrs
lhethhd,LnEbnms
ile
mocen vh   liyoofht Onelw Tr eOearnd er iTlso hd ye: tntg keTeeSr ; e. rOtd
aOawrhgkh
 epyea   efueYSr ri,AeinOdt ivnlnsu

Natural Language Generation

Shakespeare after .17 epochs

----

fiHXAnhnoeepwdnie bcaloTe? ndiol e hnhelNseooiaodUlaa::Lylo pd
e e e Steotnueeelh OSednst: eey:feio erriFhea
OadnnstDeemEislFmgs Sot: !eh, ft ltny
eiShe I on Senth.catssnac .Nachnhy
alo i
nt Mdo;aate
mh : tUem sLhrshbnRras ds aehfhhfO
tylta,hLefHtIyhCnhfraclos iysaws e
'teth!
Nrt;h e oeoe dnBsb rprInNymd metende e oihE tiirea t o haoe ns'ritt   e nc eshe e eraetrs
lhethhd,LnEbnms
ile
mocen vh   liyoofht Onelw Tr eOearnd er iTlso hd ye: tntg keTeeSr ; e. rOtd
aOawrhgkh
 epyea   efueYSr ri,AeinOdt ivnlnsu

Natural Language Generation

Shakespeare after .20 epochs
----
fod:
Thit, tullee hheenand
Yhepf le monem: ipeaml;
The dey;, than lee gpeoupteeiph feegitt,
Onw mheochith. Pf hhellss, in osairer Thiu chee ans otiy
In, me cir hhiug hhquosd I feelley
Dos o sraleige! hare daugee therpeepases ho hheange,
pyculce in why foons,
Whod owley,

K IVITH:
:euud:
I thir hiis,
eot, fyeice;
phiod i the asd, in was, fom srathot a mare is iOd loimtlr
fintein:
Thenfl on Inimee whaN riy, kho ginly
Dras inein lo ngitk mhay choss;
and mho be bhe thon hhey, mle

Natural Language Generation

Shakespeare after .37 epochs
CURTIO:
Now he vave the mensed
putinen's dont for the praces.
You kall we bairs of I love should selfur;
And more impore thee get the reasulf, we are in the keep my sarges,
I do is tent;
The pave a poy, her for yimg me his ane offinece
what stiff heart be be are ill how min ander a ding,
For refire looks, out we le? you have not her her
what be me shall If loved eyes.

Tifer:
My dytis that dast Eo, and hell, at him alme whe
would Beatter and hers it should shall fnim-day, mine:
I did it with meliol, see one and some father's prays,
Belive but us eel when it cinfoble
Acquawed for dang hit be when,
And me's rame with blood what me women to yher of eake that baw
And lot it. Ge but the gay us is ouch
'naghting his son discard our plandaziat.

Natural Language Generation

Shakespeare after 11.5 epochs:

Provost:
Hence!

GONZALO:
Sir, there is no false bride laid upon you.

BIRON:
What were you that hath proved so soldier mount?

DEMETRIUS:
Nor I, no more, good night.

KING HENRY VI:
Ill health and honourable friends be merry
Would make us nothing now another board.

CADE:
Even so.

GONZALO:
How? happy! read for germans, though indifferencies in
the king's, ten times do his answer.

NORFOLK:
You must know the heat for sleep in your own means, I
cannot drink it, he admits each vertueer in the court.
Have you yet forth him. The raven is no honest Lucius
on your griefs.

ML basics

- Regression
- Classification
- Supervised, Unsupervised
- Overfitting

History

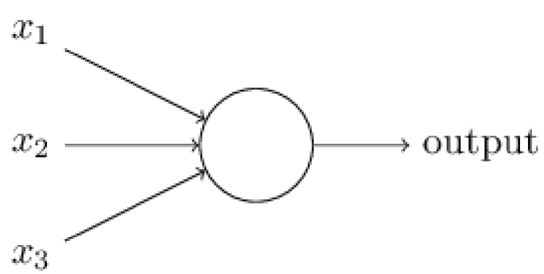
- Basic history
- Neural networks – progenitor 1943
- Backprop published in 1974
- 1989 first serious deep network
- 90s began to be used
- 2010 large scale speech recognition

History

- Single layer neural networks are just as general as deep neural-networks
- Deep networks are more difficult to train
- But, with algorithms and hardware they are better for many tasks

Basics

- Perceptron



- Formula

$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \le 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases} \tag{1}$$

http://neuralnetworksanddeeplearning.com/chap1.html

Perceptron Example

Do I mow the lawn today?

Bias b adjusts how much you want to mow the lawn

Weight w1 measures 'good weather.' this is important, so you assign it higher weight

Weight w2 measures 'grass too long.'

Weight w3 measures 'lawnmower will start easily.'

$$b = -5, w_1 = 4, w_2 = 2, w_3 = 1$$

- Scenario 1: It's raining, the grass is too long, and the mower is easy to start
- Scenario 2: It's clear, the grass is too long, and the mower takes a while to start

Perceptron Example

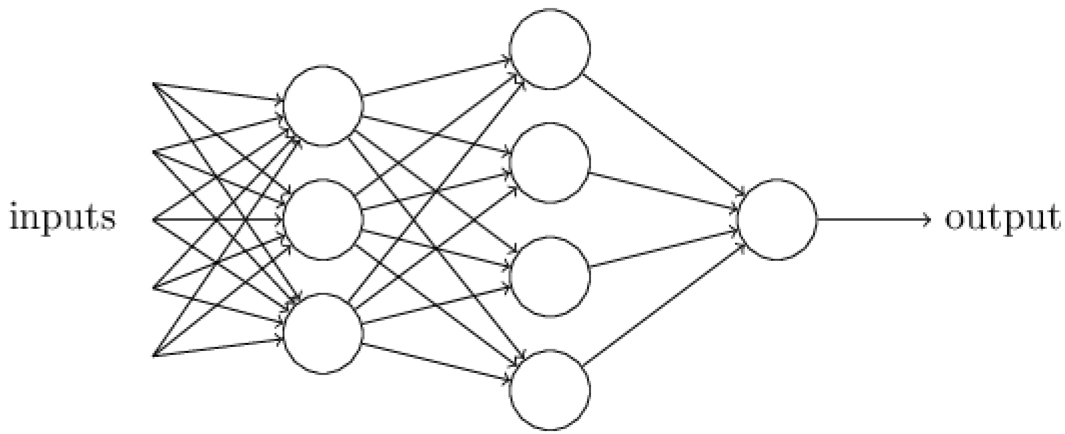
What if we really like mowing the lawn? Decrease the Bias.

$$b = -3, w_1 = 4, w_2 = 2, w_3 = 1$$

- Scenario 1: It's raining, the grass is too long, and the maintenance is done

Perceptron Net

Combine perceptrons into a network

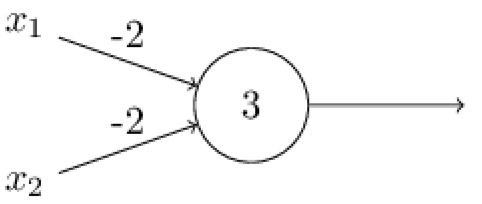


- The first layer can sum up several simple inputs into more complex considerations
- The second layer can act on these summaries

For instance: what if we broke down our earlier input 'lawnmower starts easily.'
Into the output from a perceptron consuming inputs: w1 = 'It started easily last time.'
w2 = 'Maintenance done in last 6 months.', w3= 'It's warm out'.

Perceptron NAND

- Perceptrons can implement NAND



$$b = 3, w_1 = -2, w_2 = -2$$

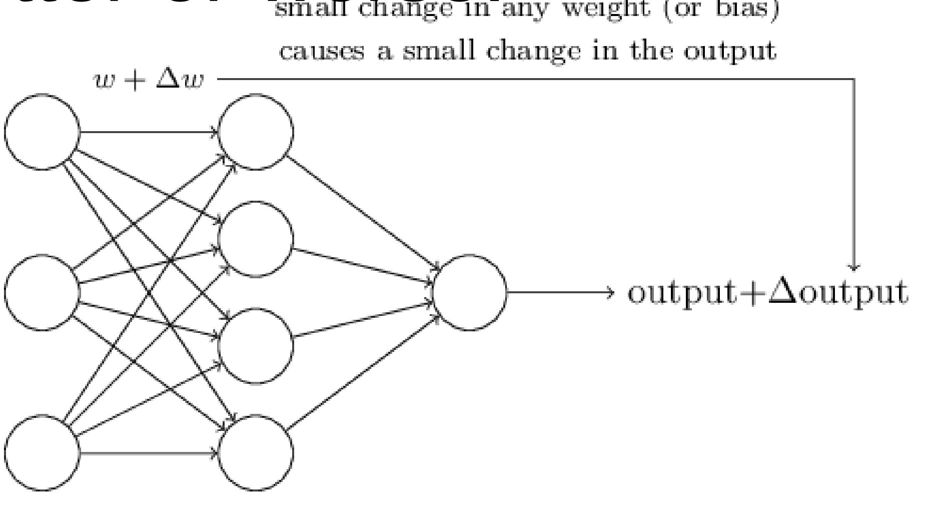
- Perceptron outputs 0 iff both inputs are 1
- Perceptrons are capable of general computation

Perceptron NAND

- So why not just use NAND gates?
- We can automatically adjust weights and biases
- This allows us to train the network to output what we want, instead of designing it

Sigmoid Neurons

- Training perceptrons is difficult since all input and output are binary
- We want to be able to make a small change to a weight or bias, and see if that makes the result better or worse.



Sigmoid Neurons

- Instead of perceptrons define sigmoid neurons:

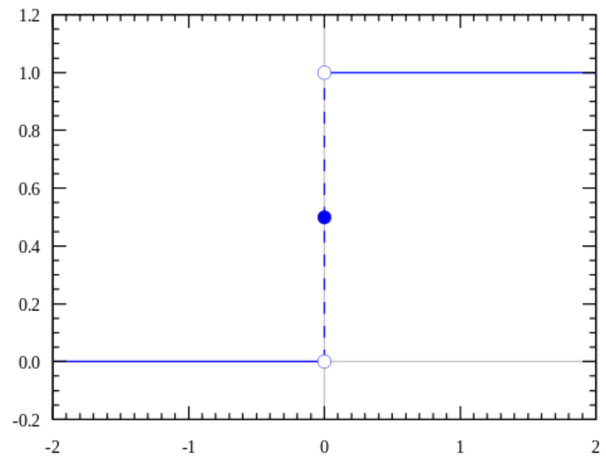
$$\sigma(z) \equiv \frac{1}{1+e^{-z}}. \tag{1}$$

$$\frac{1}{1+\exp(-\sum_j w_j x_j - b)}. \tag{2}$$

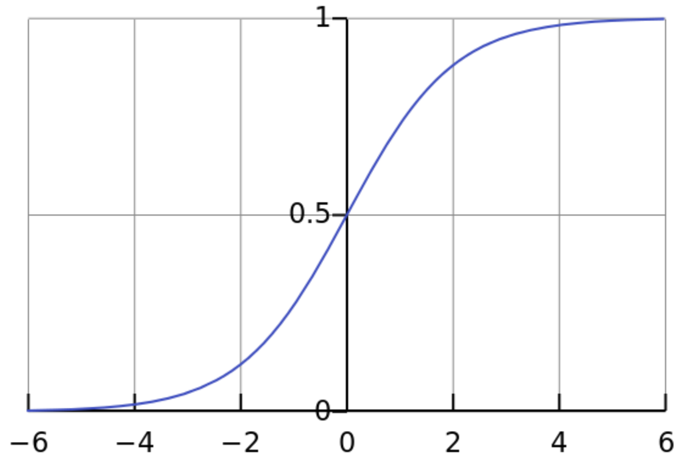
Inputs and outputs are somewhere between 0 and 1

Sigmoids

Step function:



Sigmoid:



https://en.wikipedia.org/wiki/Sigmoid_function, https://en.wikipedia.org/wiki/Step_function

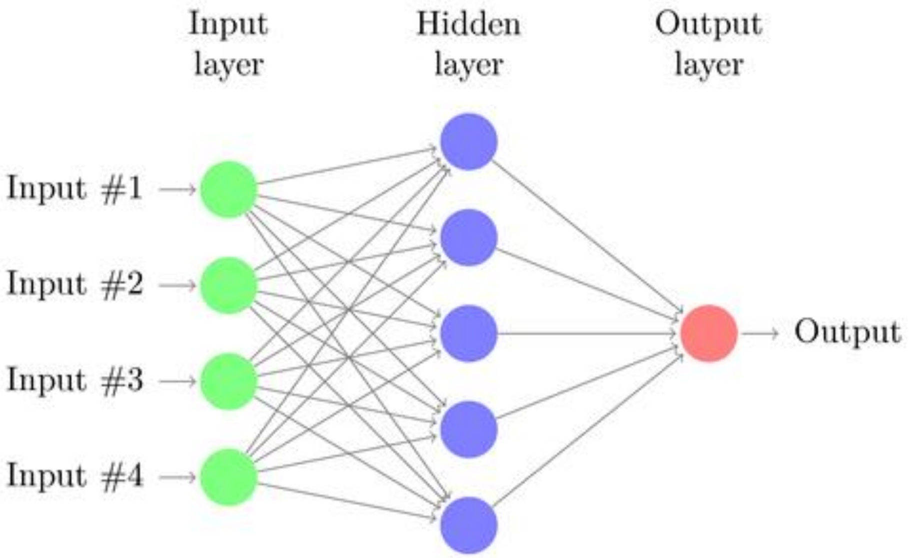
Smoothness

- This allows us to differentiate output with respect to weight and biases.
- This will be very important later

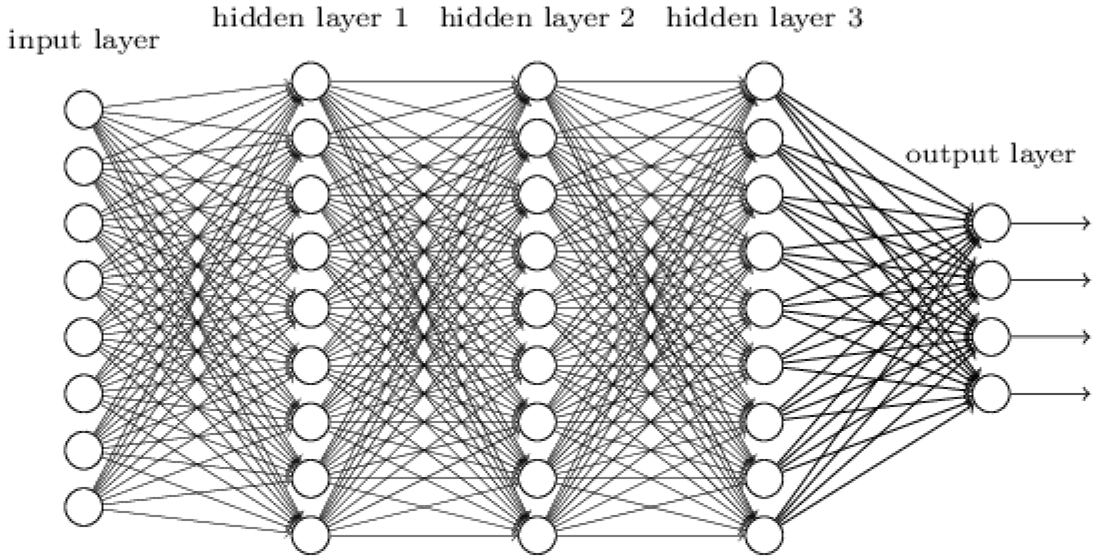
$$\Delta \mathrm{output} \approx \sum_j \frac{\partial \, \mathrm{output}}{\partial w_j} \Delta w_j + \frac{\partial \, \mathrm{output}}{\partial b} \Delta b, \qquad (1)$$

Neural Networks

Basic Neural Network:



Deep Neural Network:



http://www.texample.net/tikz/examples/neural-network/ http://neuralnetworksanddeeplearning.com/chap1.html

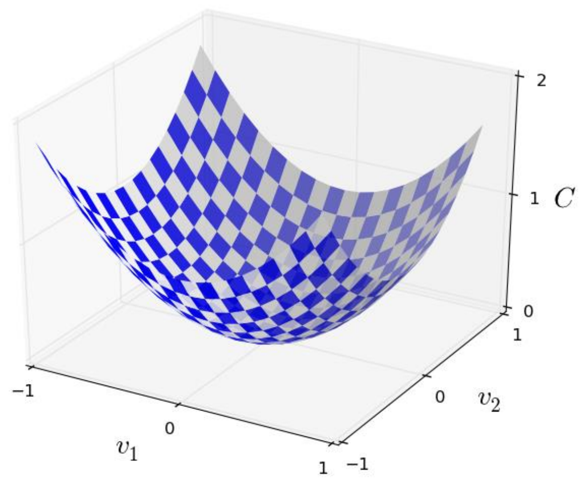
Training

- First we need data
- Then we need an objective
- Then we need to change the weights and biases to get closer to the objective
-

Cost Function

- First define a cost function (loss, objective)
- Our goal is to minimize this:

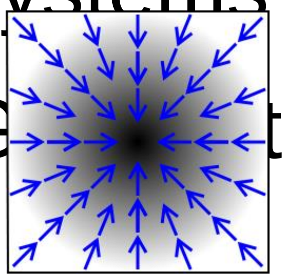
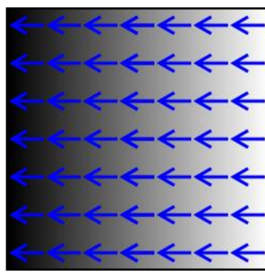
$$C(w,b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2. \qquad (1)$$



http://neuralnetworksanddeeplearning.com/chap1.html

Gradient Decent

- Goal: find the minimum
- Analytically solving breaks down for complex systems
- Gradient: generalization of derivative

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k}$$

- Transpose: row to column

$$\begin{vmatrix} a & b \\ c & d \\ e & f \end{vmatrix}^{T} = \begin{vmatrix} a & c & e \\ b & d & f \end{vmatrix} \qquad (1)$$

https://en.wikipedia.org/wiki/Gradient

Gradient Decent

- Delta of Cost function:

$$\Delta C \approx \frac{\partial C}{\partial v_1}\Delta v_1 + \frac{\partial C}{\partial v_2}\Delta v_2.$$

- Depict v1, v2 as vector v, then:

$$\Delta v \equiv (\Delta v_1, \Delta v_2)^T \tag{1}$$

$$\nabla C \equiv \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2}\right)^T \tag{2}$$

$$\Delta C \approx \nabla C \cdot \Delta v \tag{3}$$

Update Rule

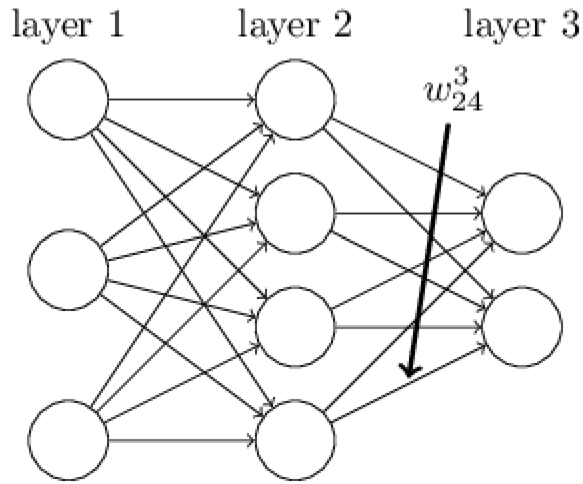
$$v \rightarrow v' = v - \eta \nabla C. \tag{1}$$

- What learning rate to use?
- Stochastic Gradient Decent

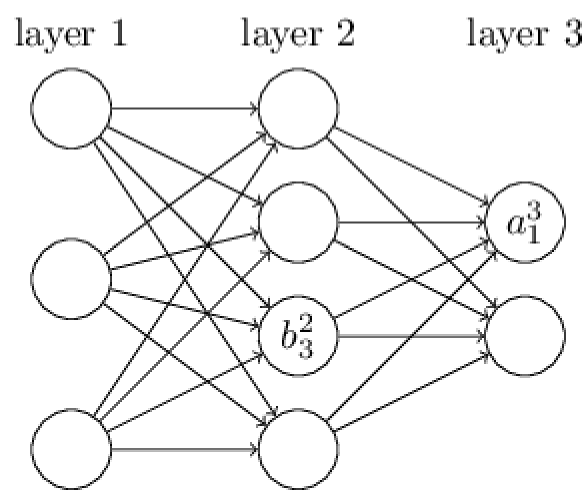
Backprogagation

- Goal: calculate error and gradient of cost for each neuron
- Assumption: Cost average of training examples
- Assumption: Function of output

Backpropagation



w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer



Backpropagation

- Changing a single neuron:

$$\frac{\partial C}{\partial z_j^l} \Delta z_j^l$$

- Error definition

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}.$$

Backpropagation Formulas

The first backpropagation formula

$$\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L).$$

The four backpropagation formulas in matrix form

$$\begin{aligned}
\delta^L &= \nabla_a C \odot \sigma'(z^L). && (1)\\
\delta^l &= ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l), && (2)\\
\frac{\partial C}{\partial b_j^l} &= \delta_j^l. && (3)\\
\frac{\partial C}{\partial w_{jk}^l} &= a_k^{l-1}\delta_j^l. && (4)
\end{aligned}$$

Training

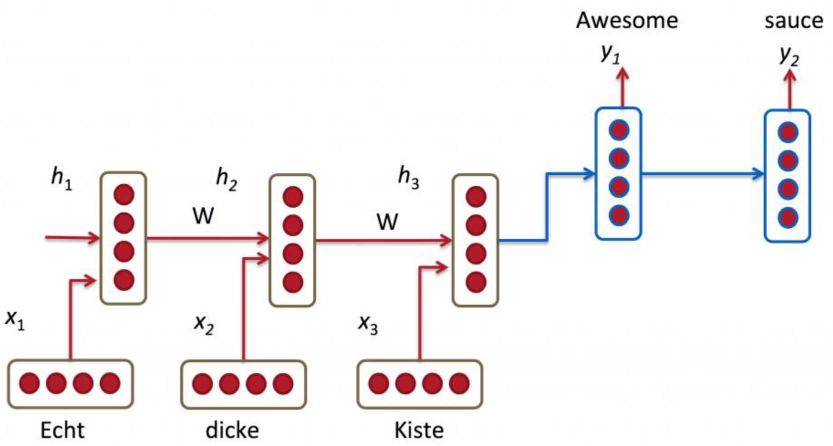
- Data
- Split data into train and val (and test)
- Define Input and Output
- Define Cost (loss, objective)
- Propagate forwards
- Propagate loss backwards
- Update
- Repeat
- Validate

More types of Neural Networks

- Recurrent
- Convolutional
- Autoencoders
- Adversarial

Recurrent Neural Networks

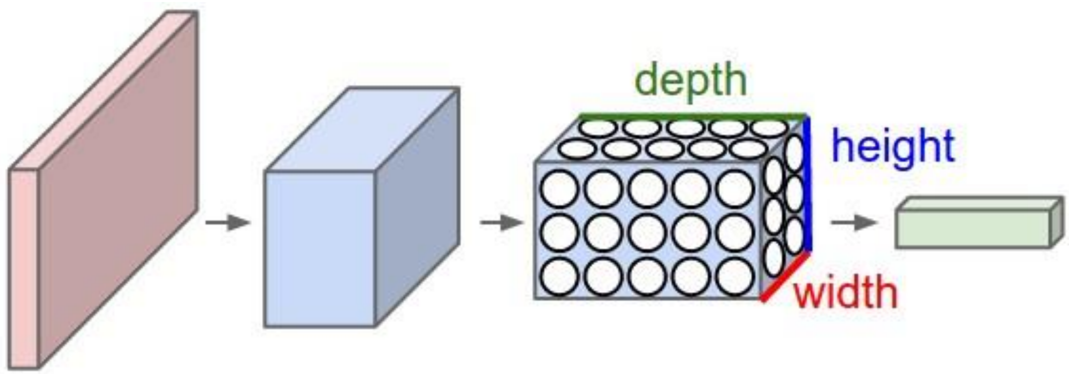
- All these networks only go forward (feedforward).
- Sequences might be better modeled with recurrence
- Hidden state
- Bidirectional, hierarchical
- LSTM



http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf

Convolutional Neural Networks

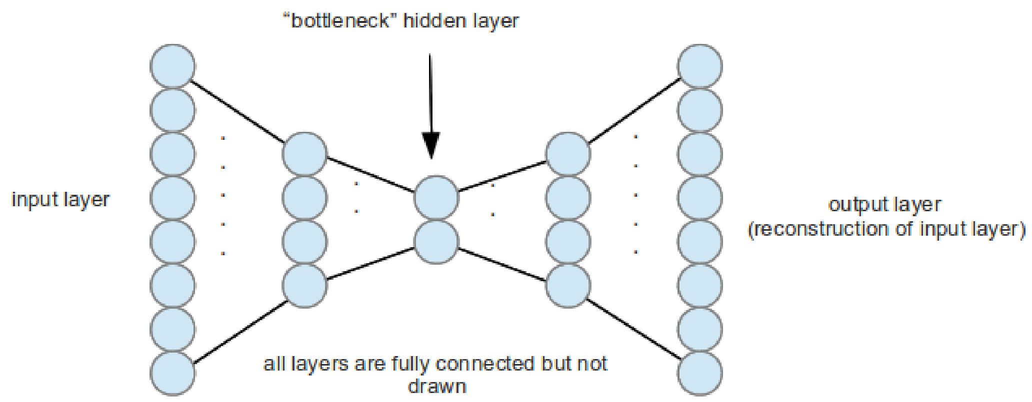
- Used in vision
- Based on biology (loosely)
- S connected.
  R



http://cs231n.github.io/convolutional-networks/

Autoencoders

- Input = Output
- Forces it to learn how to 'summarize'
- Unsupervised learning



http://nghiaho.com/?p=1765

Adversarial Generative Networks

- Neural nets tend to average.
- Generator and Discriminator
- They compete, so don't average as much

Summation

- What are deep neural nets
- What are they used for
- How do they work
- How do I use them

Neural-Network Libraries

- Torch
- Theano
- Tensorflow
- CNTK
- Caffe
- Lasagna, etc

- Torch
  - lua
  - speedy
  - facebook, deepmind
  - better for high end research

- Theano
  - python
  - symbolic
  - lots of libraries
  - slowest

- Tensorflow
  - python
  - more distributed
  - fairly new
  - google
  - good docs
  - a little slower

- CNTK
  - config files, c++ base
  - microsoft
  - only option for multi machine, fastest on distributed

- Caffe
  - c++, but several good front ends
  - very good for image processing
  - more restrictive, especially to make completely new models, but very easy to use existing.

What Next

- Read this next:

convnet tutorial

wildml

neuralnetworksanddeeplearning

- Fun projects to look at

char-rnn

neural-storyteller

neuraltalk2

Questions, and Possibly Answers

Run your own example

git clone https://github.com/reidsanders/dl-talk.git

A quick overview of how I run my models:

Small models on my laptop with an nvidia discrete chip
For larger: Amazon gpu instance
Aquired via spot bid (aws cli, set your security group outbound rule to your ip, and recheck re
Running ubuntu 14.04 (possibly use a ml ami, but many of these are out of date)
Use Cuda or opencl (probably cuda)
Install your libraries (use virtualenv for python please)
Deploy with fabric
When using ssh, use tmux
Train, keeping an eye on validation and training loss
Download results and checkpoints with fabric, save a snapshot in ec2 console or cli