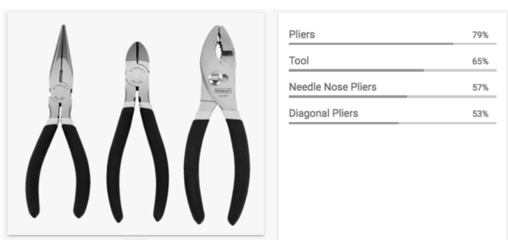
ECS 189
Web Programming

6/5

Exams

- If you are satisfied with your scores on the two midterms, you can skip the final
- As soon as your Photobooth and midterm are graded, I can give you your course grade (so far) so you can decide
- The material in this lecture will not be on the final. It covers the "interesting part" of our project.

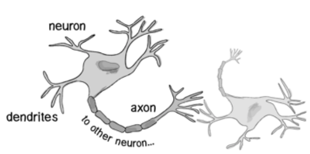
Cloud Vision API – the magic



How does it work?

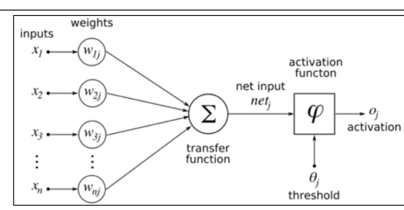
- Sadly, it really is magic…or at least not well understood!
- But we have a word for it…Convolutional Neural Networks!
- References:
  - "Computer eyesight gets a lot more accurate", NY Times
  - Stanford CS 231n
  - Christopher Olah's blog
- Take ECS 174!

Neural Networks

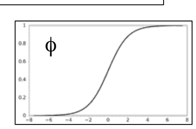


- Program based on metaphor of how brains work
- Highly interconnected collection of neurons
- Input comes in on dendrites
- When input is "enough", neuron "fires" and sends signal through axon to other neurons

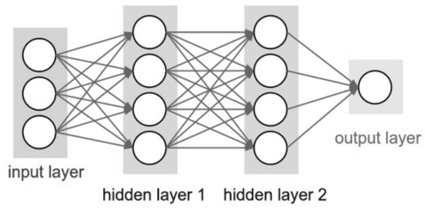
Artificial Neuron



- Possibly many inputs
- Weights
- Continuous φ function

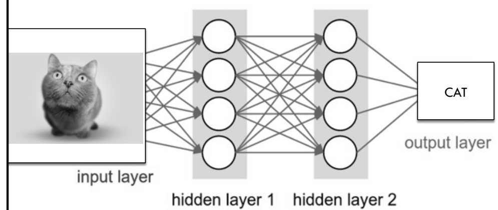


Neural Network



- Much bigger; billions of connections
- Different configurations
- "Deep learning" -> many layers

Image Labeling

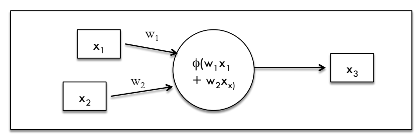


- Input is image pixels
- Output is words

Training

- Large training set of image-label pairs
- Start with random weights
- For each training image:
  - Adjust weights at every node to move the activation of the training label up and the other labels down
  - Do this over and over again until cycling through the whole training set does not change the weights any more
  - This takes a long, long time

Weight adjustment – the chain rule!

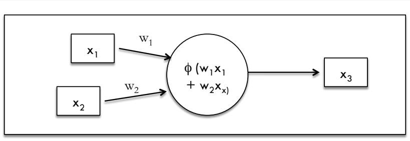


O = output of whole network

$dO/dw_1 = dx_3/dw_1 \; dO/dx_3$

$dO/dx_1 = dx_3/dx_1 \; dO/dx_3$

Weight adjustment – the chain rule!



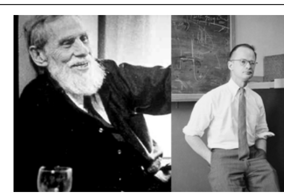
O = output of whole network

$dO/dw_1 = dx_3/dw_1 \; dO/dx_3$

$dO/dx_1 = dx_3/dx_1 \; dO/dx_3$

Start at the output end, work your way back taking derivatives – **back propagation**

Old idea



- McCulloch and Pitts, seminal paper, 1943
- Rosenblatt, weights contain memory, 1958
- Minsky and Pappert, critique, 1969
- Rumelhart, Hinton and Willams, backprop, 1986

So why is it a big thing now?

- Computational power – we can get really big networks to converge
  - Use GPUs
  - Do it at Google in a massive distributed parallel environment (Tensorflow)

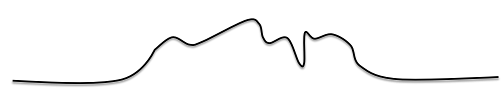
So why is it a big thing now?

- Computational power – we can get really big networks to converge
- Big data! We have enough training data to get really big networks to converge

So why is it a big thing now?

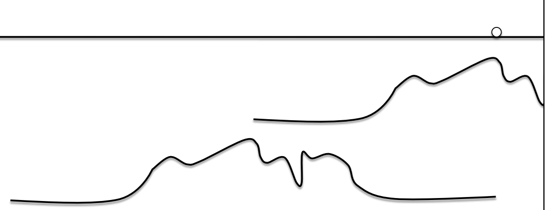
- Computational power – we can get really big networks to converge
- Big data! We have enough training data to get really big networks to converge
- Better organization of networks
  - Convolutional neural network

Convolution



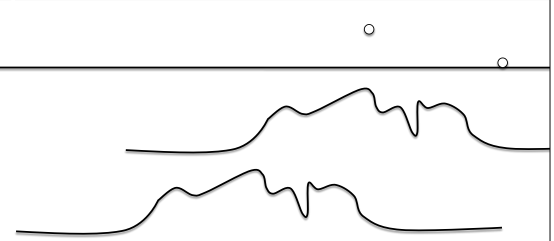
- Convolution is the mathematical operation for pattern matching
- Say we wanted to find a face in a 1D signal

Convolution



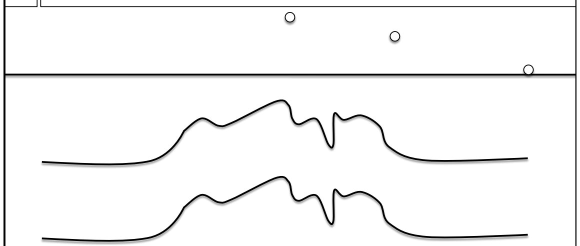
- Use a face-shaped pattern. Multiply pattern by face at every point to get a score for pattern position.

Convolution



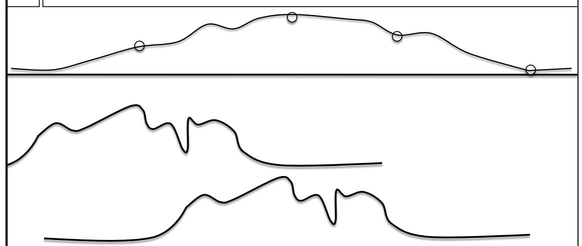
- Use a face-shaped pattern. Multiply pattern by face at every point to get a score for pattern position.

Convolution



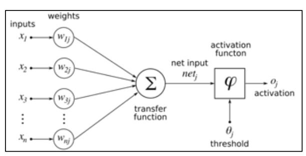
- Best score a position where pattern and signal match the best!

Convolution



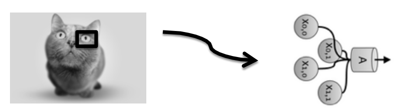
- Best score a position where pattern and signal match the best!

Convolution of original image



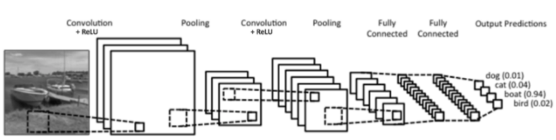
- Weights get multiplied by input x. So we will get strong signals (o_j) if weights form a pattern that matches part of the image (eg. kitty eyes)

Patterns for images



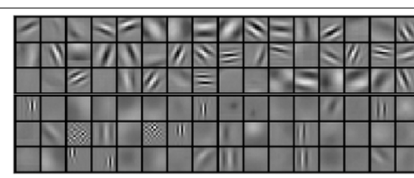
- Feed all small sub-images to a neuron, and hope we optimize weights so it makes an interesting max.
- Create hundreds or thousands of pattern neurons, optimize each over all possible positions, look for peak outputs

Convolutional Neural Network



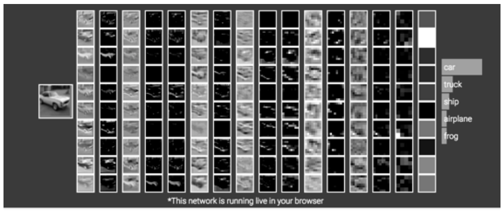
- Image as input
- Convolution consider features of small areas
- Pooling layers select the strongest features
- Fully connected layers connect features to outputs

Automatically generated features



- Two sets of convolution features trained on two different GPUs, feeding into the same fully connected layer
- One side looks for dark-to-light, edges, gradients, the other works on color and texture

Recognition in the browser



- Training is slow but recognition is pretty fast (we are seeing seconds, including the round-trip to Google

Big data for image labeling

- ImageNET training data
- Li Fei-Fei (Stanford) and Kai Li (Princeton)
  - A category for every English noun
  - Over a billion images
  - Massive data collection using Mechanical Turk over several years
  - Key factor that has enabled us to train CNNs for this task

Other big Deep NN applications

- Speech recognition
- Machine translation
  - "The great AI awakening" – NY Times
- Text understanding and question answering (grammar, meaning)