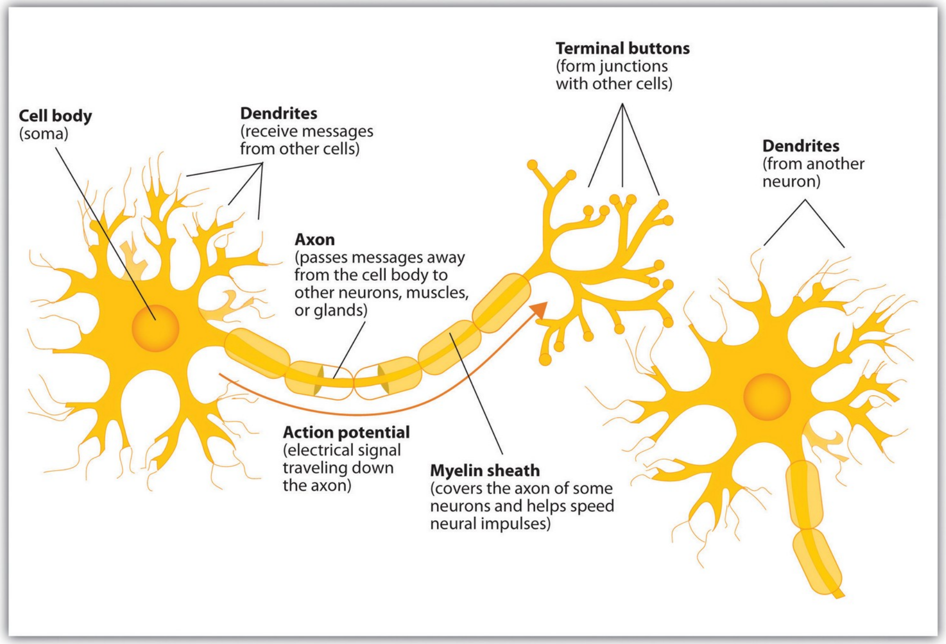
Convolutional Neural Networks

Nathan Sprague

Neurons

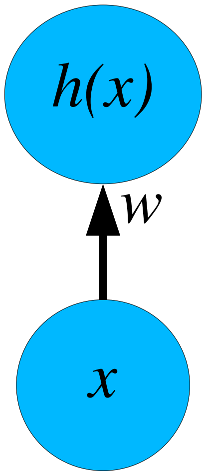
- Neurons communicate using discrete electrical signals called "spikes" (or action potentials).
  - Spikes travel along axons.
  - Reach axon terminals.
  - Terminals release neurotransmitters.
  - Postsynaptic neurons respond by allowing current to flow in (or out).
  - If voltage crosses a threshold a spike is created



Beginning Psychology (v. 1.0).
http://2012books.lardbucket.org/books/beginning-psychology/
Creative Commons by-nc-sa 3.0

Linear Regression – The Neural View

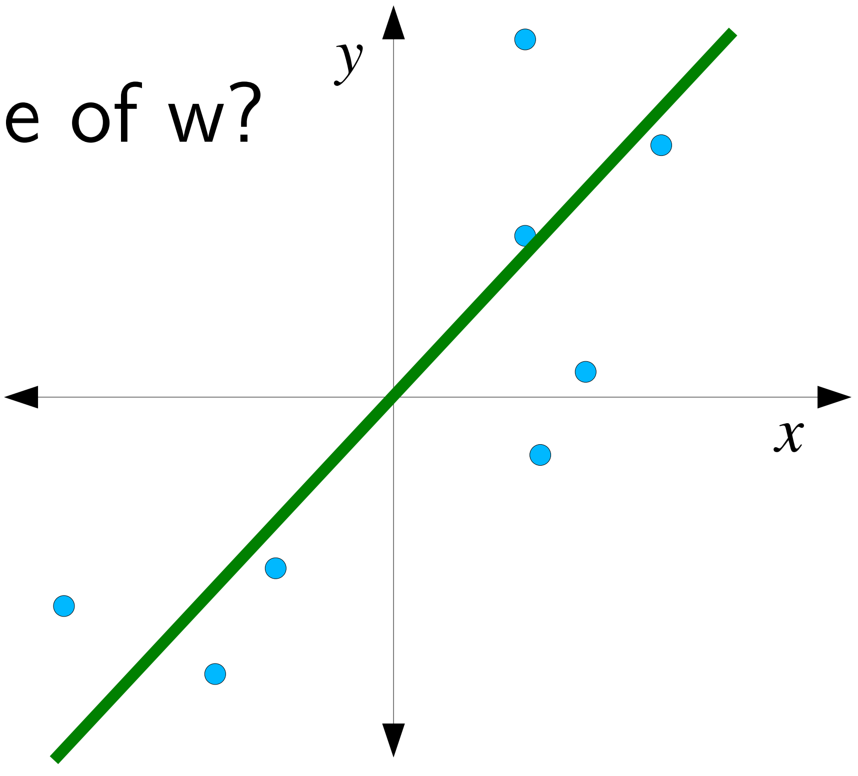
- input $= x$, desired output $= y$, weight $= w$.
- $h(x) = wx$



- We are given a set of inputs, and a corresponding set of outputs, and we need to choose w.
- What's going on geometrically?

Lines

- $h(x) = wx$ *is* the equation of a line with a y intercept of 0.
- What is the best value of w?
- How do we find it?

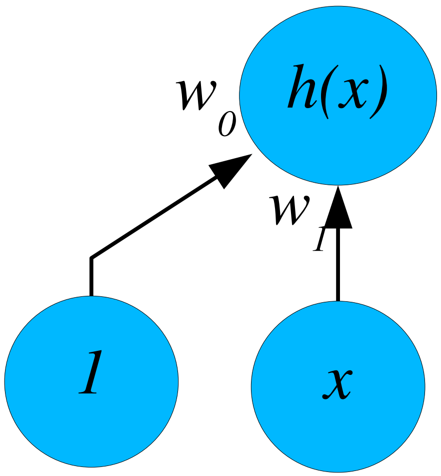


Bias Weights

- We need to use the general equation for a line:

  $$h(x) = w_1 x + w_0$$

- This corresponds to a new neural network with one additional weight, and an input fixed at 1.



Error Metric

- Sum squared error (y is the desired output):

$$Error_E = \sum_{e \in E} \frac{1}{2} \left(y_e - h(\boldsymbol{x}_e) \right)^2$$

- The goal is to find a w that minimizes E. How?

Gradient Descent

http://en.wikipedia.org/wiki/File:Glacier_park1.jpg
Attribution-Share Alike 3.0 Unported

Gradient Descent

- One possible approach (maximization):

  1)take the derivative of the function: $f'(w)$

  2)guess a value of w : $\hat{w}$

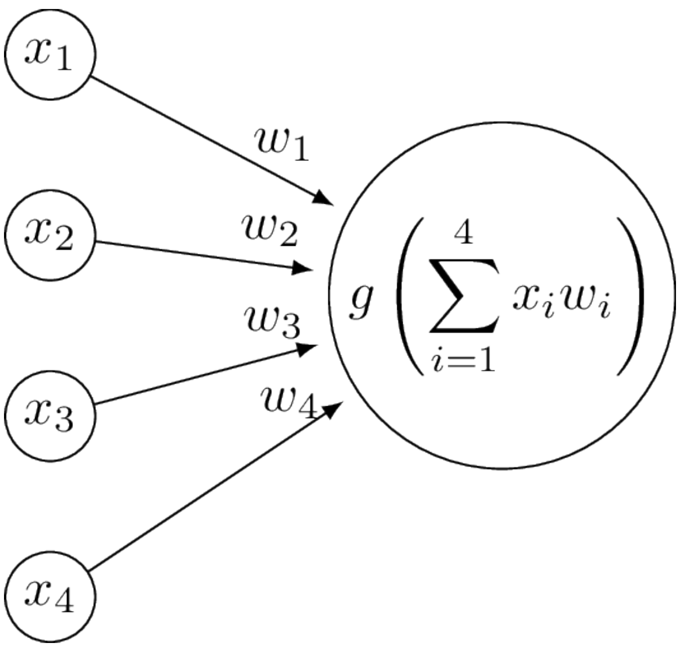
  3)move $\hat{w}$ a little bit according to the derivative:

  $$\hat{w} \leftarrow \hat{w} - \eta f'(\hat{w})$$

  4)goto 3, repeat.

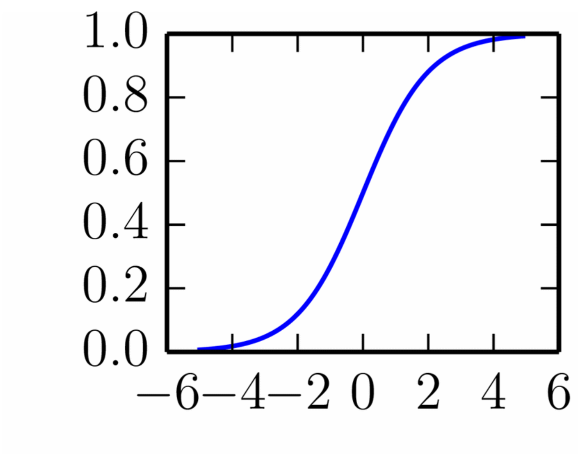
Full “Neuron”

Neuron

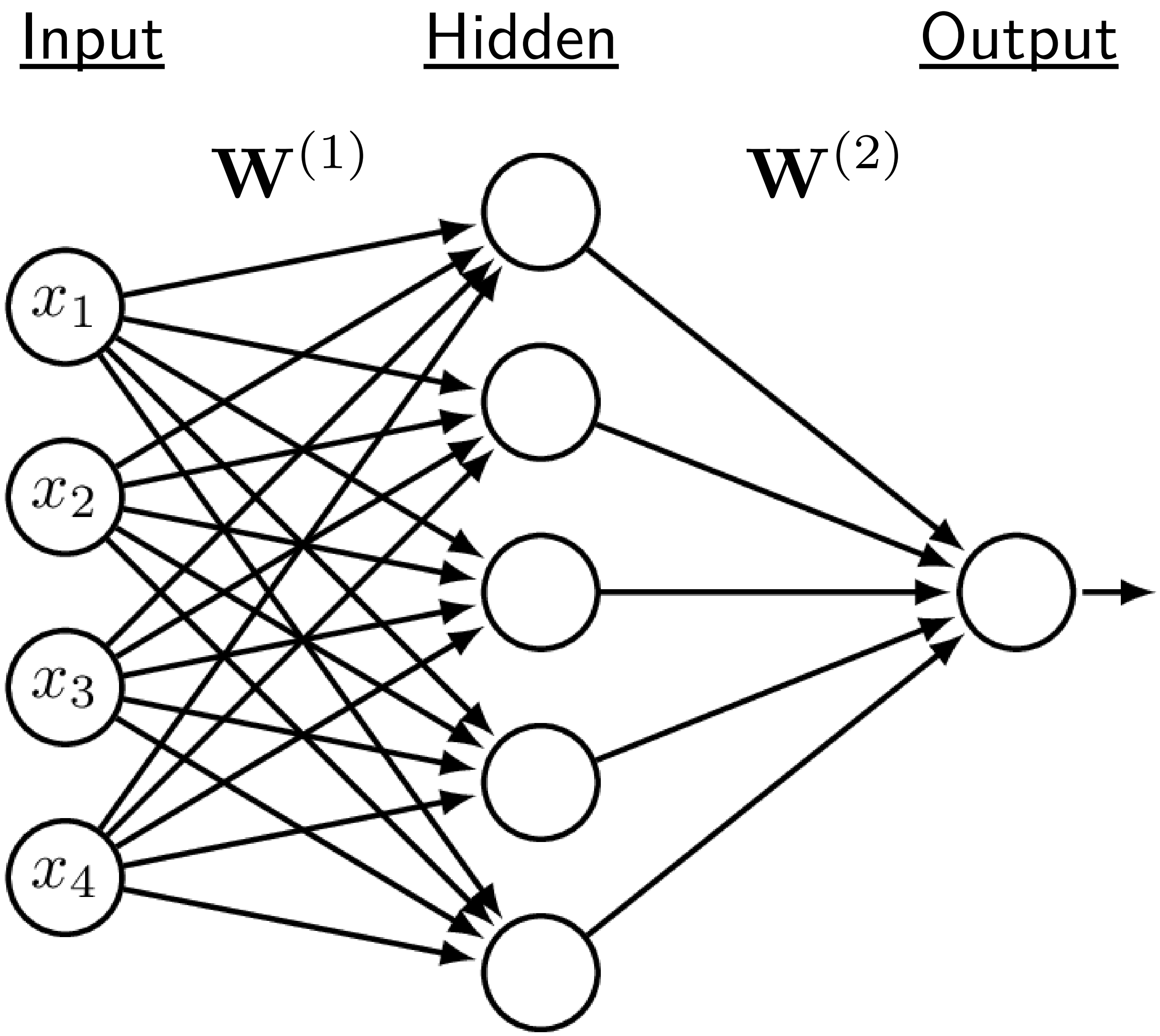


Non-linearity

$$g(a) = \frac{1}{1+e^{-a}}$$

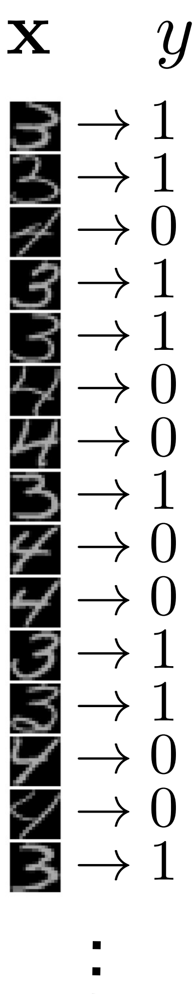


Multi-Layer Networks

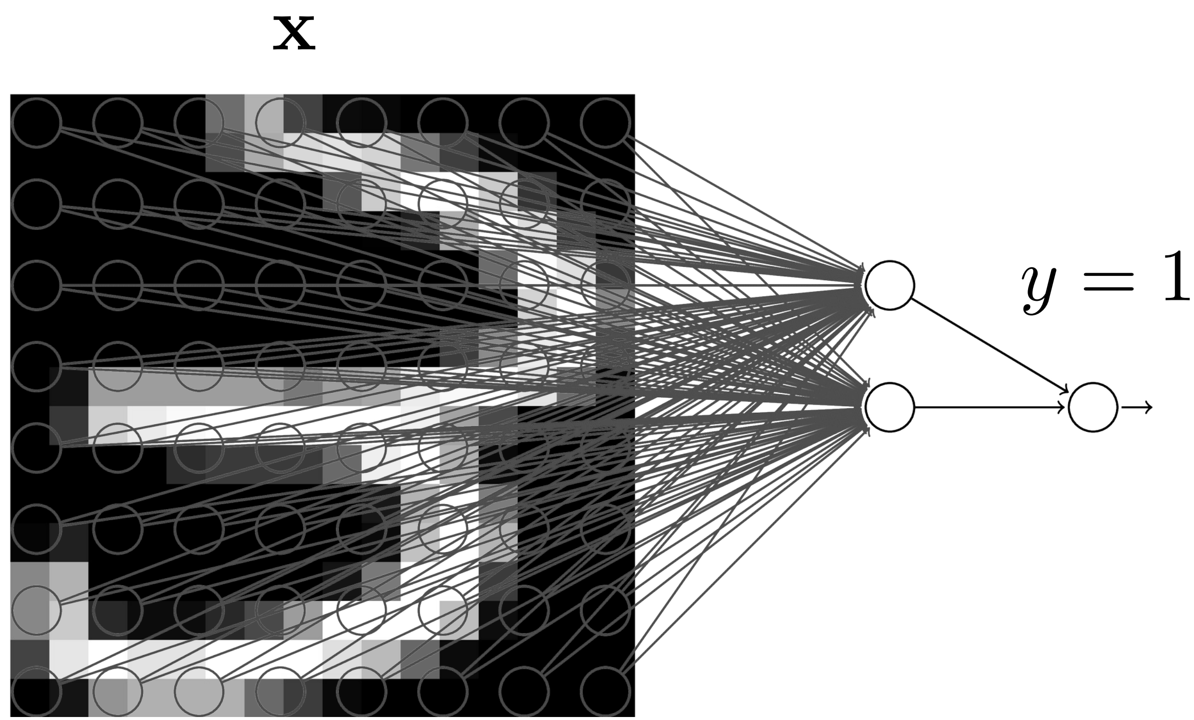


Neural Network Example

Training Data



Network



Backpropagation

- Activation at the output layer:

$$a_k = o\left(\sum_j w_{j,k}^{(2)}\, g\left(\sum_i w_{i,j}^{(1)} x_i\right)\right)$$

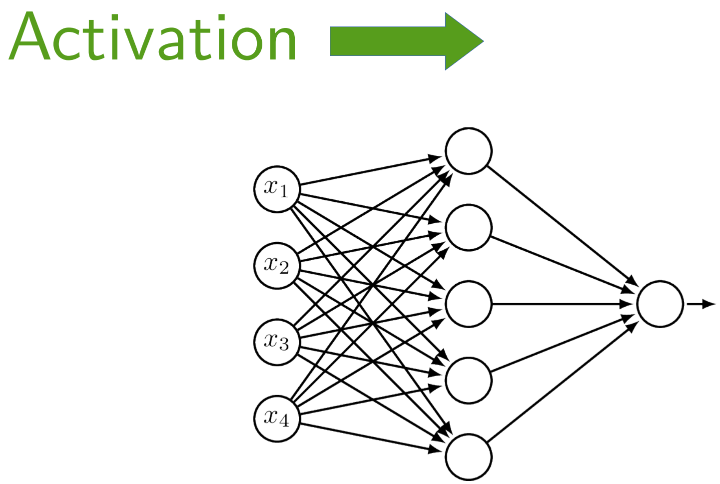
- Here o is the activation function at the output layer. Units at the input layer are indexed with *i*, hidden with *j* and output with *k.*
- Error metric, assuming multiple output units:

$$Error = \frac{1}{k}\sum_k (y_k - a_k)^2$$

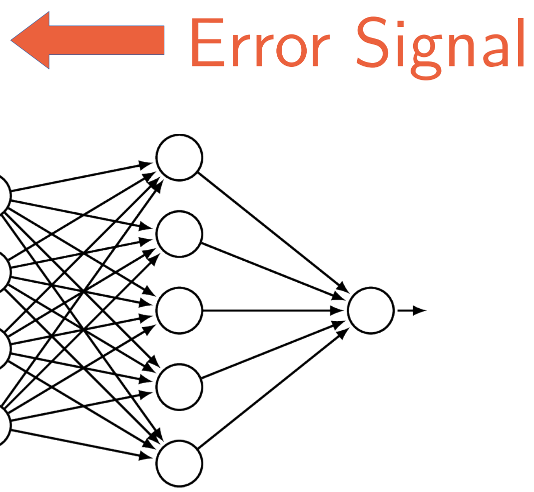
- Now just compute $\frac{\partial\, Error}{\partial\, w_{j,k}^{(2)}}$ and $\frac{\partial\, Error}{\partial\, w_{i,j}^{(1)}}$.

Backpropagation Algorithm

- Forward Pass:



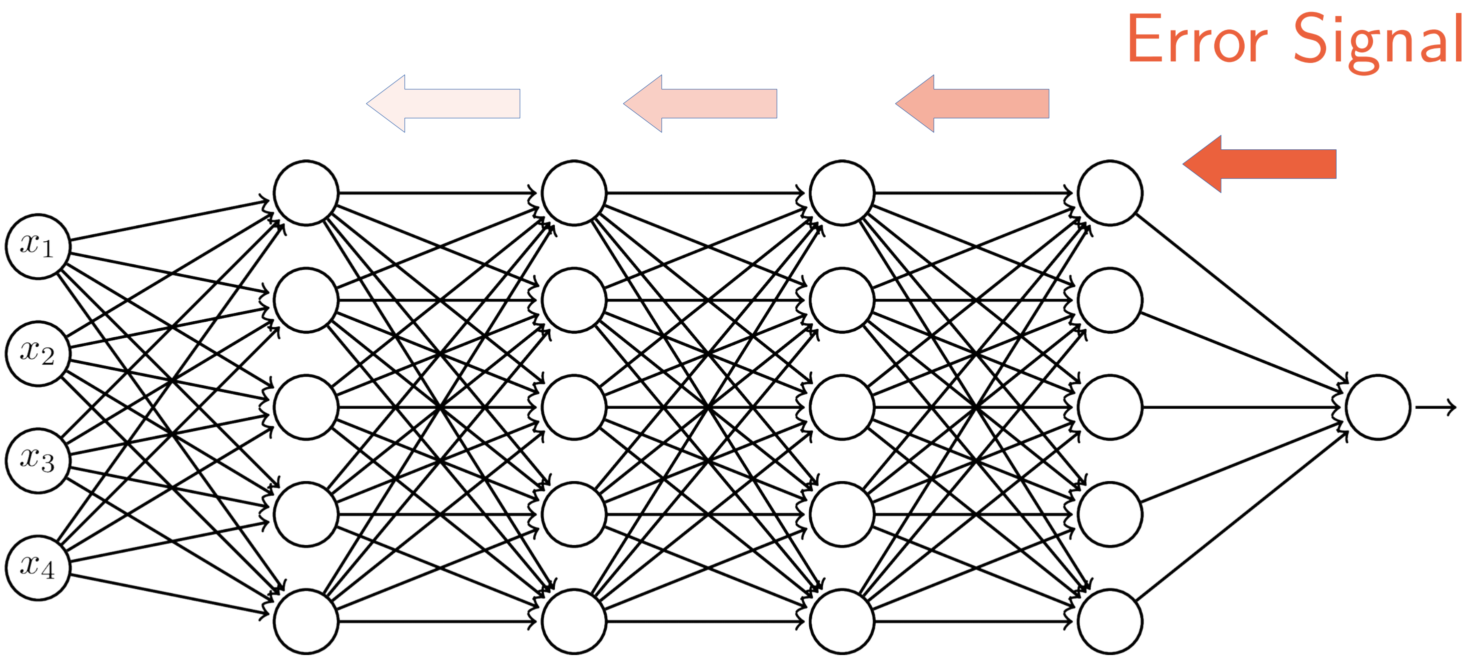
- Backward Pass:



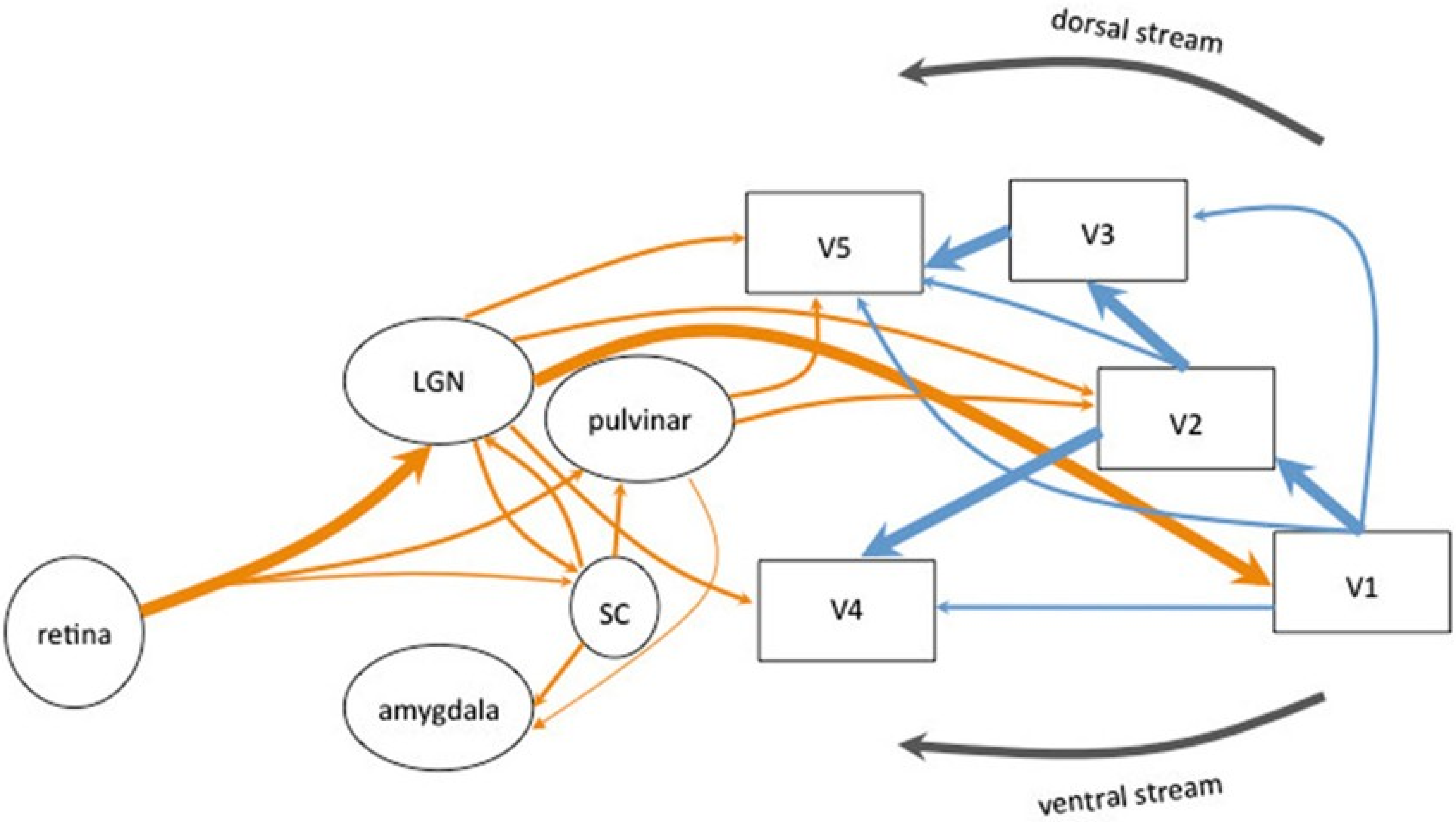
Backpropagation: Some Good News

- Calculating partial derivatives is tedious, but mechanical
- Modern neural network libraries perform automatic differentiation
  - Tensorflow
  - PyTorch
- The programmer just needs to specify the network structure and the loss function – No need to explicitly write code for performing weight updates
- The computational cost for the backward pass is not much more than the cost for the forward pass

Vanishing Gradients



Human Visual System



Urbanski, Marika, Olivier A. Coubard, and Clémence Bourlon. "Visualizing the blind brain: brain imaging of visual field defects from early recovery to rehabilitation techniques." Neurovision: Neural bases of binocular vision and coordination and their implications in visual training programs (2014).

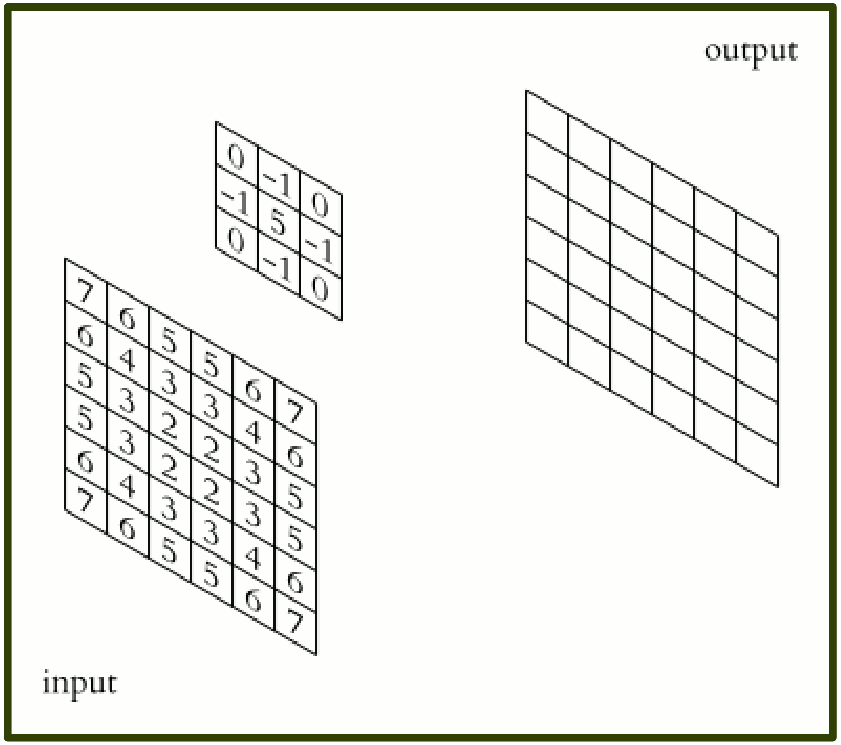
Convolutional Neural Networks

- Convolutional neural networks use the same trick of learning layers of localized features...
- CNN's were actually being used by Yann Lecun at Bell Labs around 1990

Convolutions

Grayscale Image

1 convolutional filter

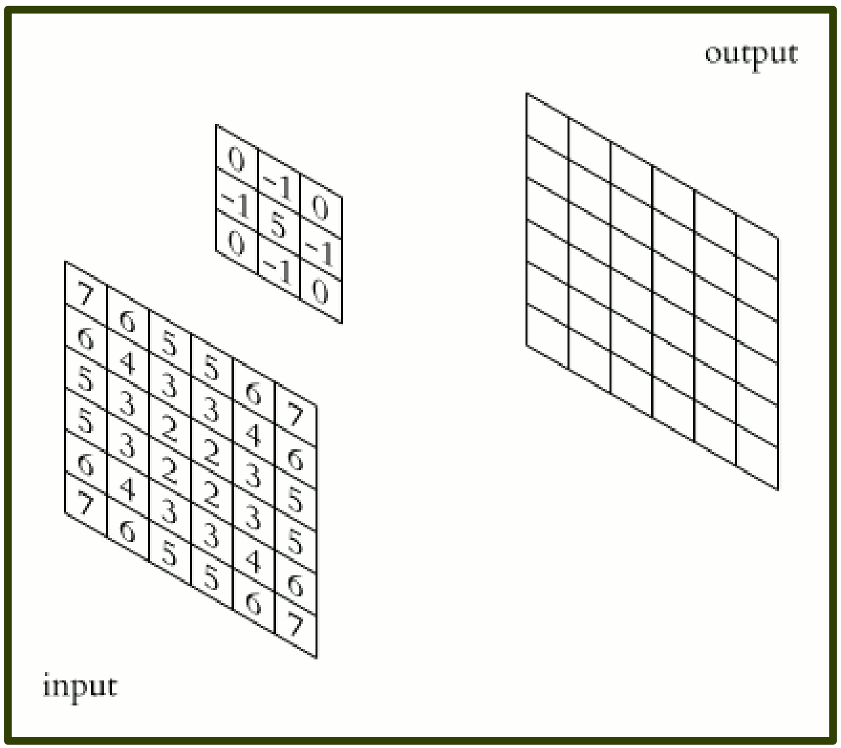


http://upload.wikimedia.org/wikipedia/commons/4/4f/3D_Convolution_Animation.gif

By Michael Plotke [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)

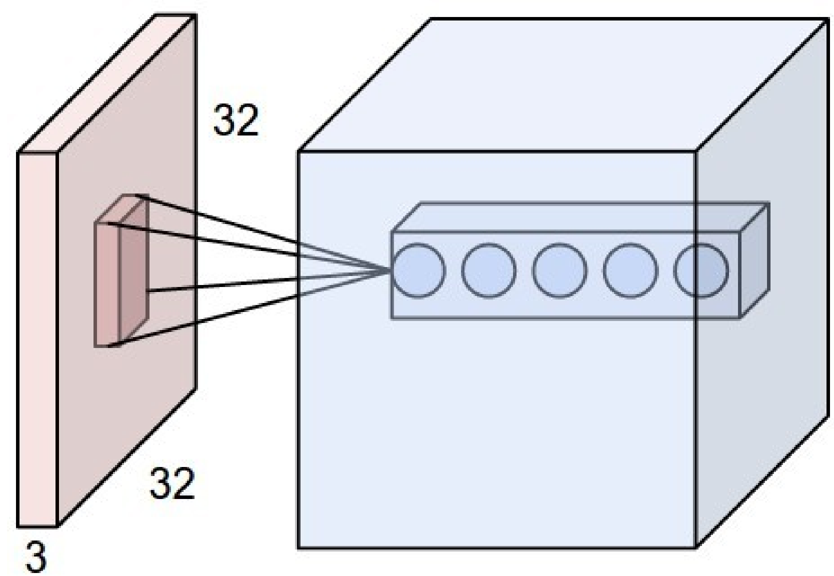
Convolutions

Grayscale Image
1 convolutional filter



http://upload.wikimedia.org/wikipedia/commons/4/4f/3D_Convolution_Animation.gif
By Michael Plotke [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)

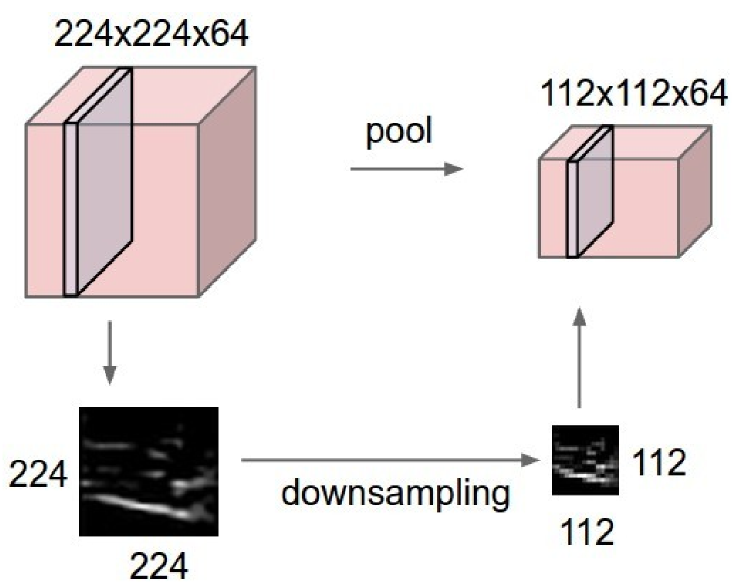
Color Image
5 convolutional filters

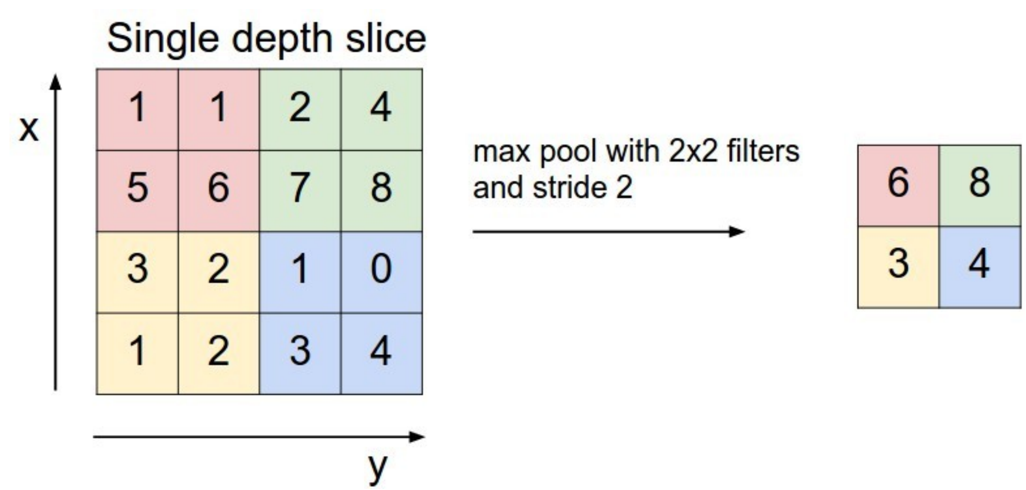


http://cs231n.github.io/convolutional-networks/
The MIT License (MIT)
Copyright (c) 2015 Andrej Karpathy

Pooling Layers

- Pooling layers down-sample the filter outputs to
  - Reduce dimensionality and computational requirements
  - Increase the spatial extent of subsequent filters

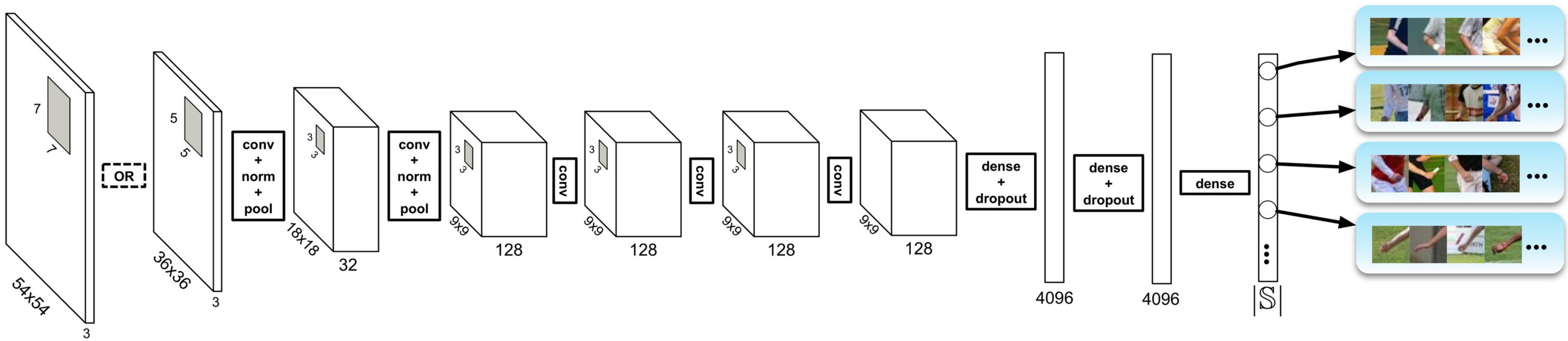




http://cs231n.github.io/convolutional-networks/
The MIT License (MIT)
Copyright (c) 2015 Andrej Karpathy

Complete Network

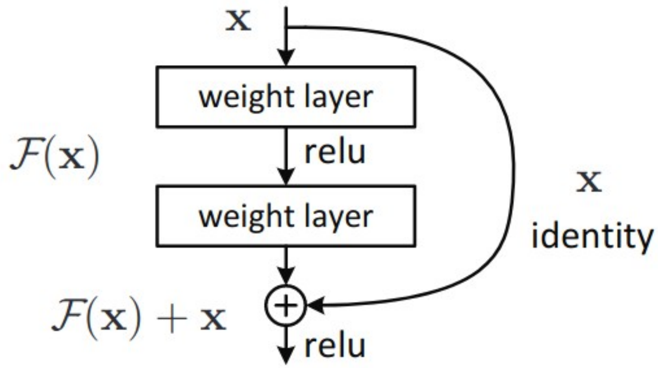
- A "traditional" CNN is composed of convolutional layers, each followed by non-linearities, followed by pooling layers, with a dense (non-convolutional) layer at the end:



Chen, Xianjie, and Alan L. Yuille. "Articulated pose estimation by a graphical model with image dependent pairwise relations." Advances in Neural Information Processing Systems. 2014.

Residual Networks

- How deep can we make these networks? Simply stacking more convolutional layers eventually degrades performance.
- One solution is to introduce "skip connections":



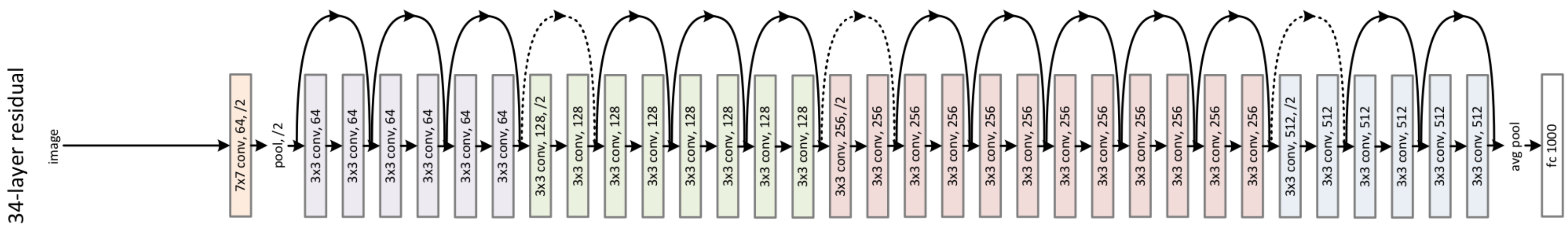
- "Residual learning"

He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

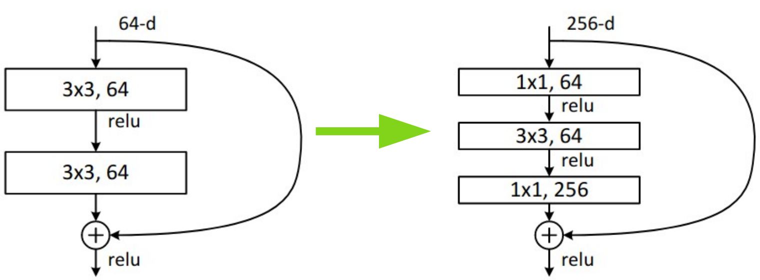
> 70,000 citations!

Residual Networks

- ResNet-34:



- Get ResNet-50 by introducing “bottleneck” blocks:



- The 1x1 convolutions can be used to increase or decrease the number of channels