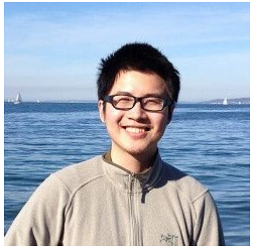
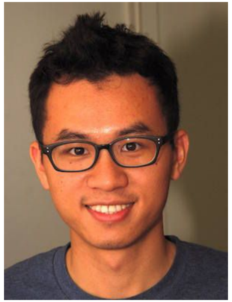
Lecture 1: Introduction to Deep Learning

CSE599W: Spring 2018

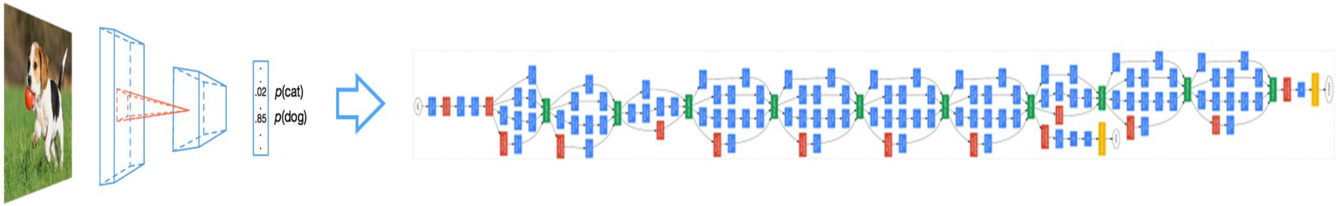
PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Lecturers






ML Applications need more than algorithms



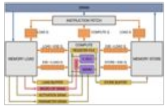
Learning Systems: this course

Hardware









What’s this course

- **Not** about Learning aspect of Deep Learning (except for the first two)

- System aspect of deep learning: faster training, efficient serving, lower memory consumption.

Logistics

- Location/Date: Tue/Thu 11:30 am - 12:50pm MUE 153
- Join slack: https://uw-cse.slack.com dlsys channel
- We may use other time and locations for invited speakers.
- Compute Resources: AWS Education, instruction sent via email.
- Office hour by appointment

Homeworks and Projects

- Two code assignments

- Group project
  - Two to three person team
  - Poster presentation and write-up

A Crash Course on Deep Learning

Elements of Machine Learning

Model

$$x_i = \begin{bmatrix} \text{feature}_0 \\ \text{feature}_1 \\ \cdots \\ \text{feaure}_m \end{bmatrix} \Rightarrow \hat{y}_i = \frac{1}{1 + \exp\left(-w^T x_i\right)}$$

Objective

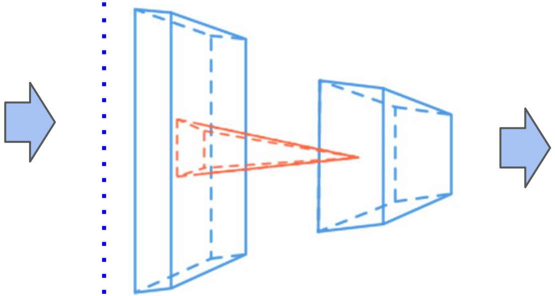
$$L(w) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \lambda \|w\|^2$$

Training

$$w \leftarrow w - \eta \nabla_w L(w)$$

What’s Special About Deep Learning

Compositional Model



predictor

$$\hat{y}_i = \frac{1}{1 + \exp\left(-w^T x_i\right)}$$

End to End Training

Ingredients in Deep Learning

- Model and architecture

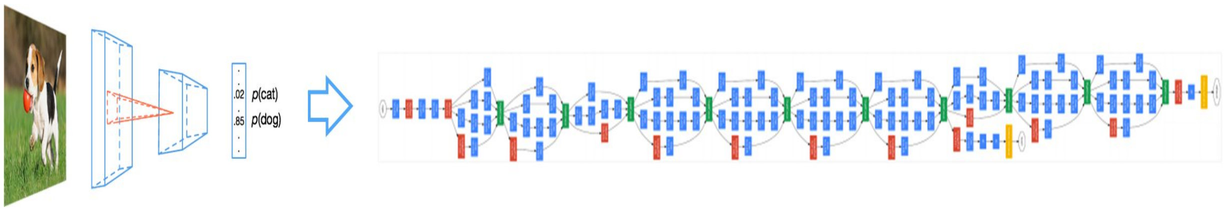
- Objective function, training techniques
  - Which feedback should we use to guide the algorithm?
  - Supervised, RL, adversarial training.

- Regularization, initialization (coupled with modeling)
  - Dropout, Xavier

- Get enough amount of data

Major Architectures

Image Modeling
Convolutional Nets



Language/Speech
Recurrent Nets

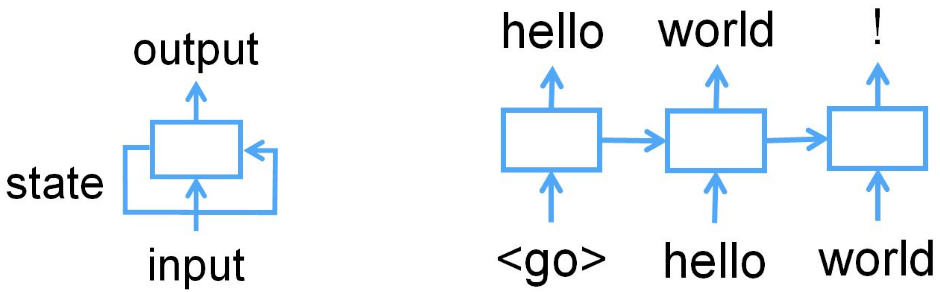
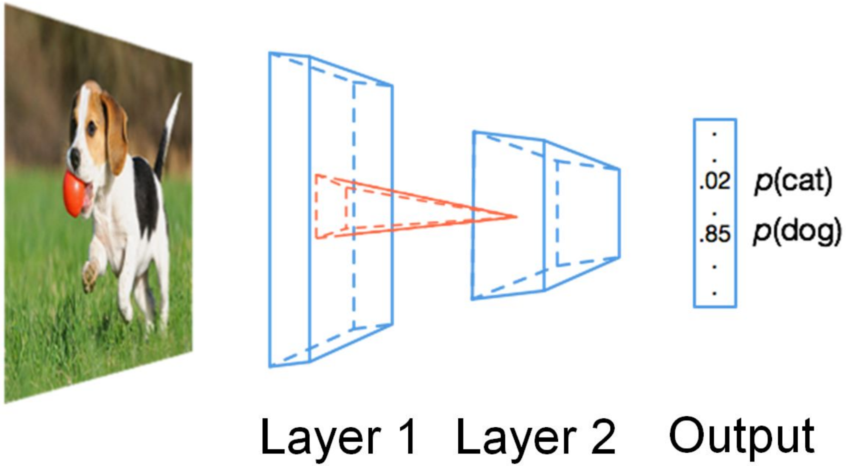
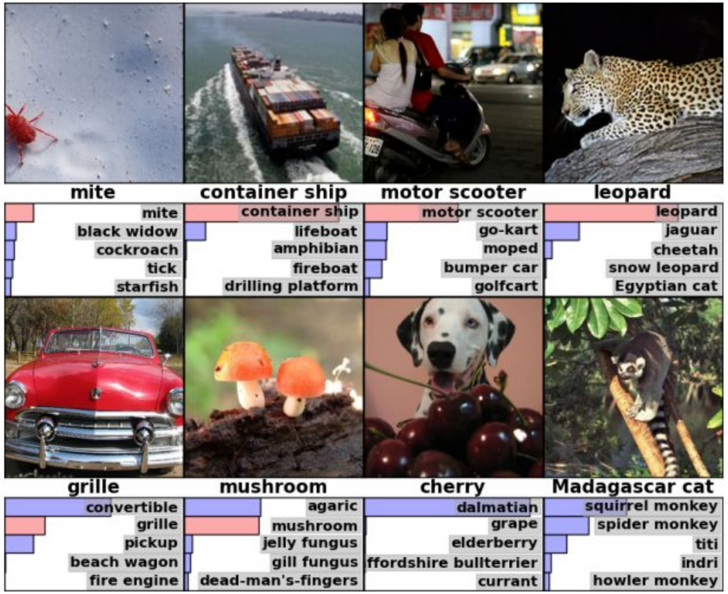


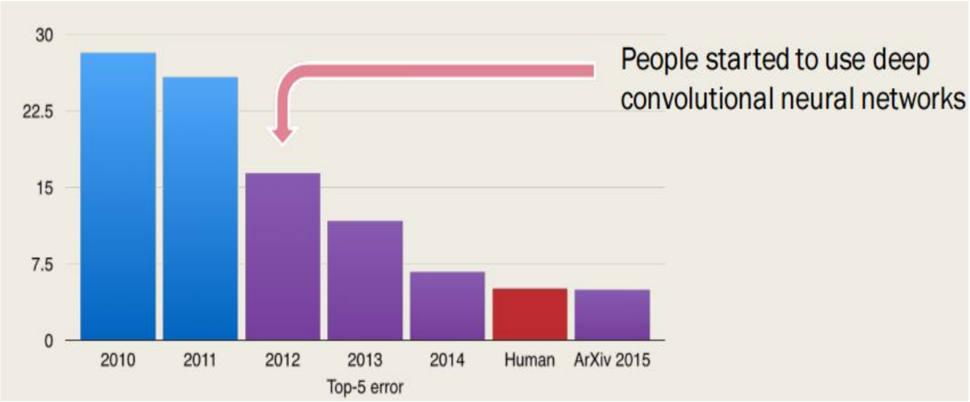
Image Modeling and Convolutional Nets



explore spatial information with convolution layers

Breakthrough of Image Classification

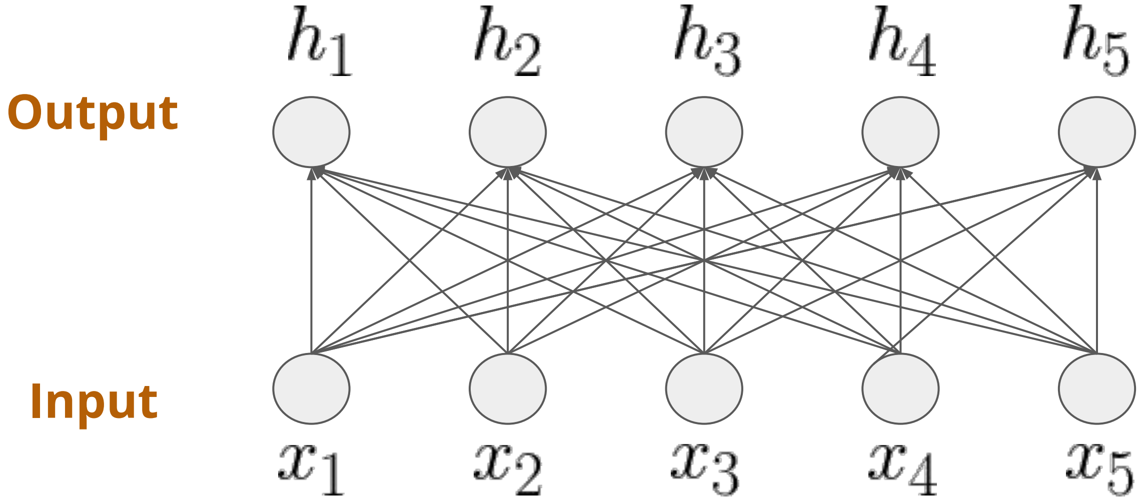




Evolution of ConvNets

- LeNet (LeCun, 1998)
  - Basic structures: convolution, max-pooling, softmax
- Alexnet (Krizhevsky et.al 2012)
  - ReLU, Dropout
- GoogLeNet (Szegedy et.al. 2014)
  - Multi-independent pass way (Sparse weight matrix)
- Inception BN (Ioffe et.al 2015)
  - Batch normalization
- Residual net (He et.al 2015)
  - Residual pass way

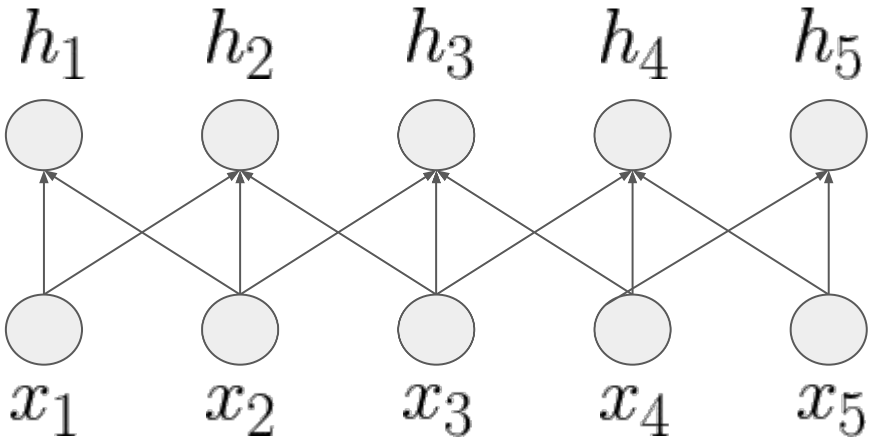
Fully Connected Layer



$$h_i = \sum_{j=1}^{5} W_{ij} x_i \qquad h_1 = W_{11}x_1 + W_{21}x_2 + W_{31}x_3 + W_{41}x_4 + W_{51}x_5$$

Convolution = Spatial Locality + Sharing

Spatial Locality



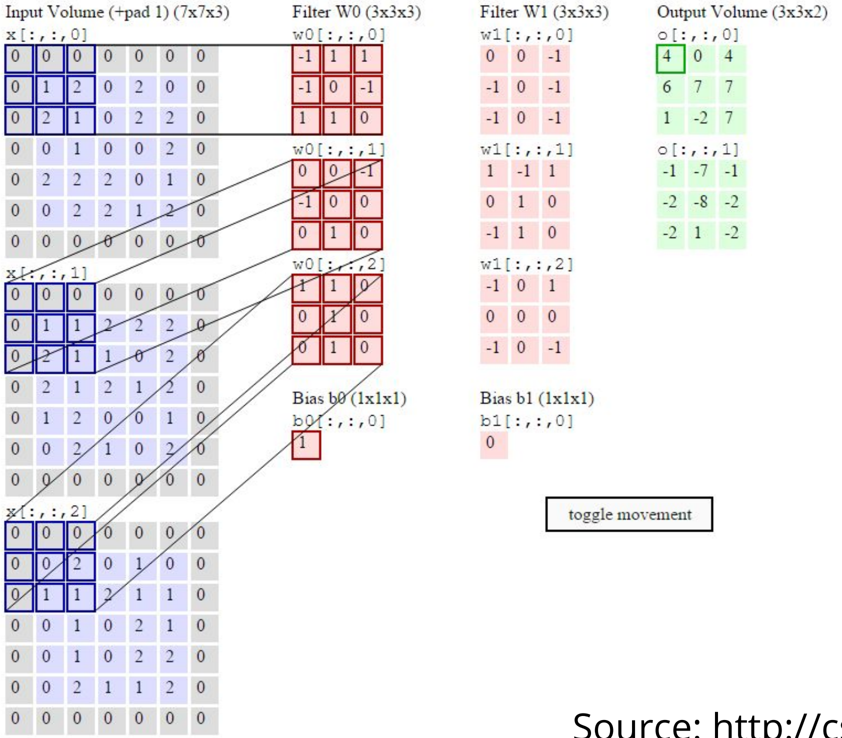
Without Sharing

$$h_i = W_{1,i} x_{i-1} + W_{2,i} x_i + W_{3,i} x_{i+1}$$

With Sharing

$$h_i = W_1 x_{i-1} + W_2 x_i + W_3 x_{i+1}$$

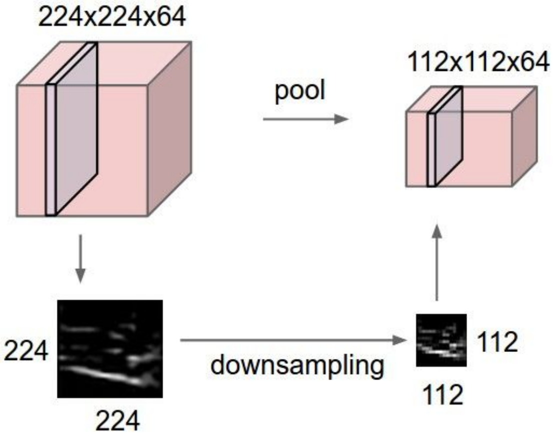
Convolution with Multiple Channels

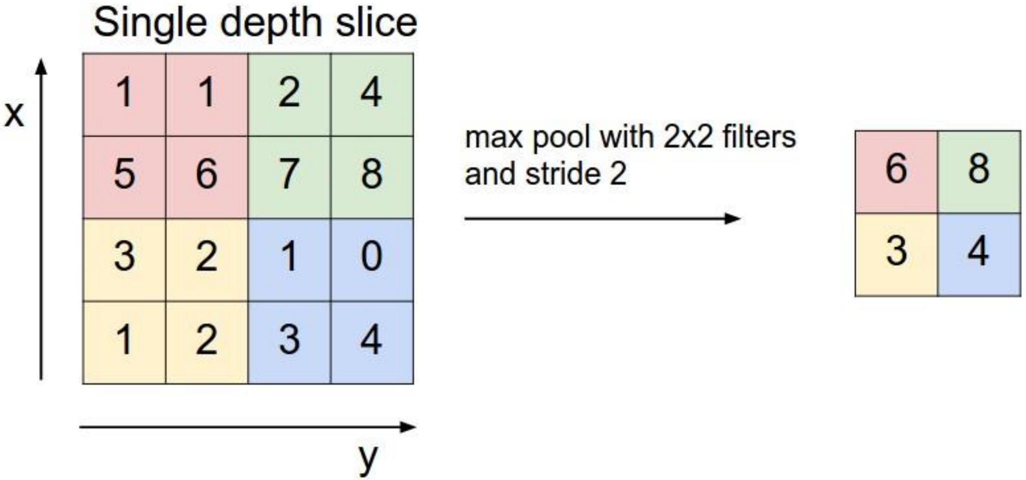


Source: http://cs231n.github.io/convolutional-networks/

Pooling Layer

Can be replaced by strided convolution



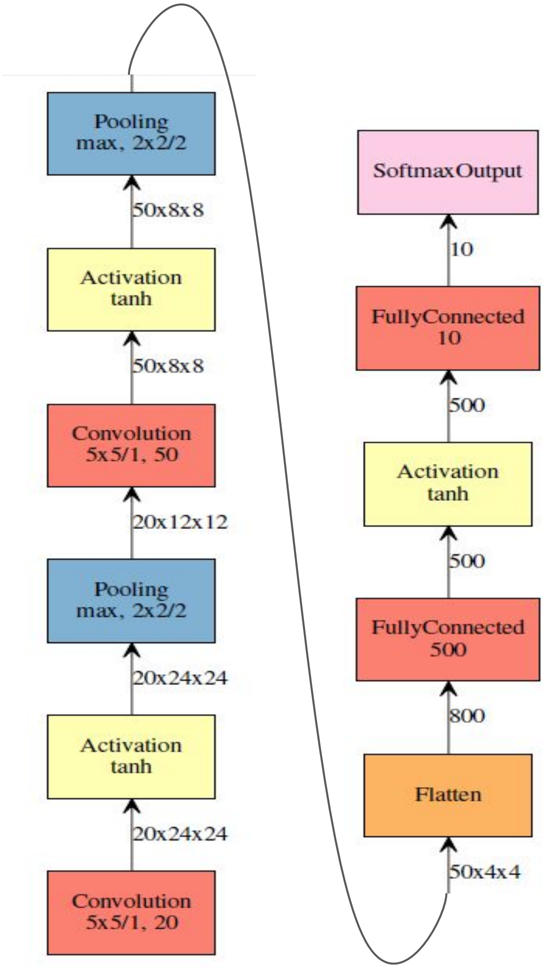


Source: http://cs231n.github.io/convolutional-networks/

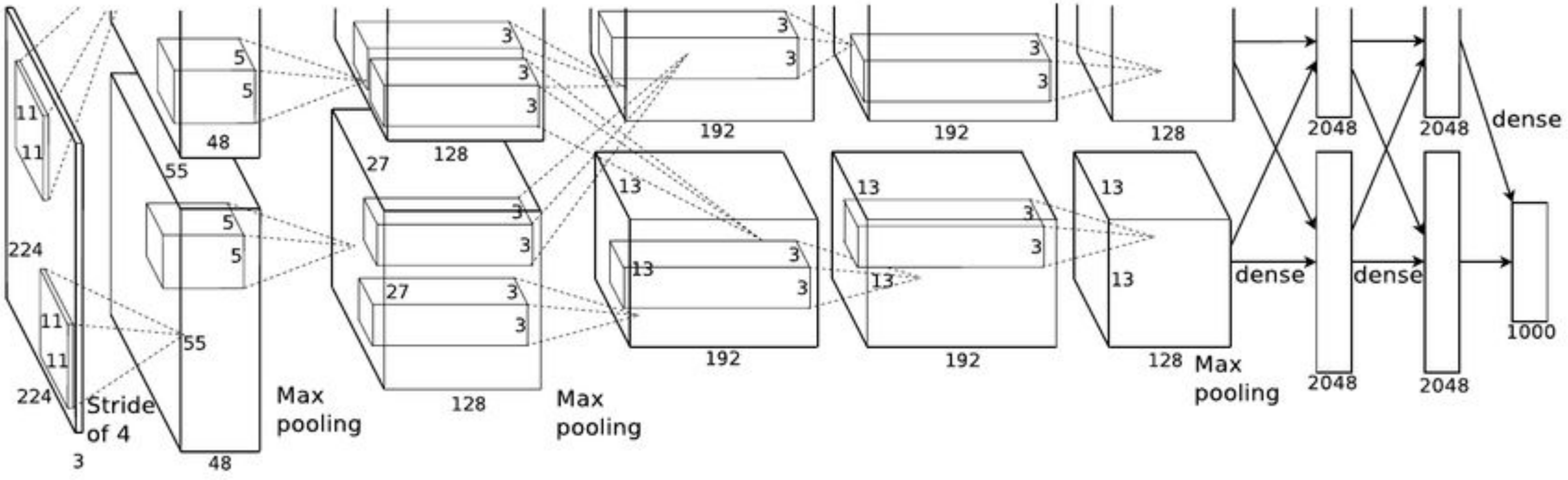
LeNet (LeCun 1998)

- Convolution
- Pooling
- Flatten
- Fully connected
- Softmax output

$$p(i) = \frac{e^{\frac{f(i)}{T}}}{\sum_j e^{\frac{f(j)}{T}}}$$



AlexNet (Krizhevsky et.al 2012)



PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Challenges: From LeNet to AlexNet

- Need much more data: ImageNet
- A lot more computation burdens: GPU

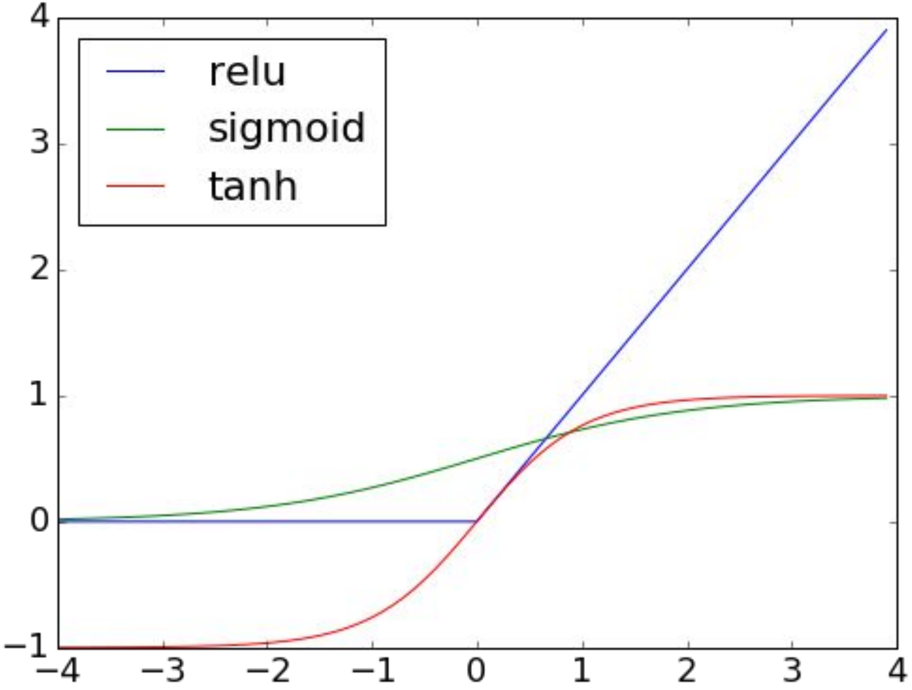
- Overfitting prevention
  - Dropout regularization

- Stable initialization and training
  - Explosive/vanishing gradient problems
  - Requires careful tuning of initialization and data normalization

ReLU Unit

- ReLU $y = max(x, 0)$

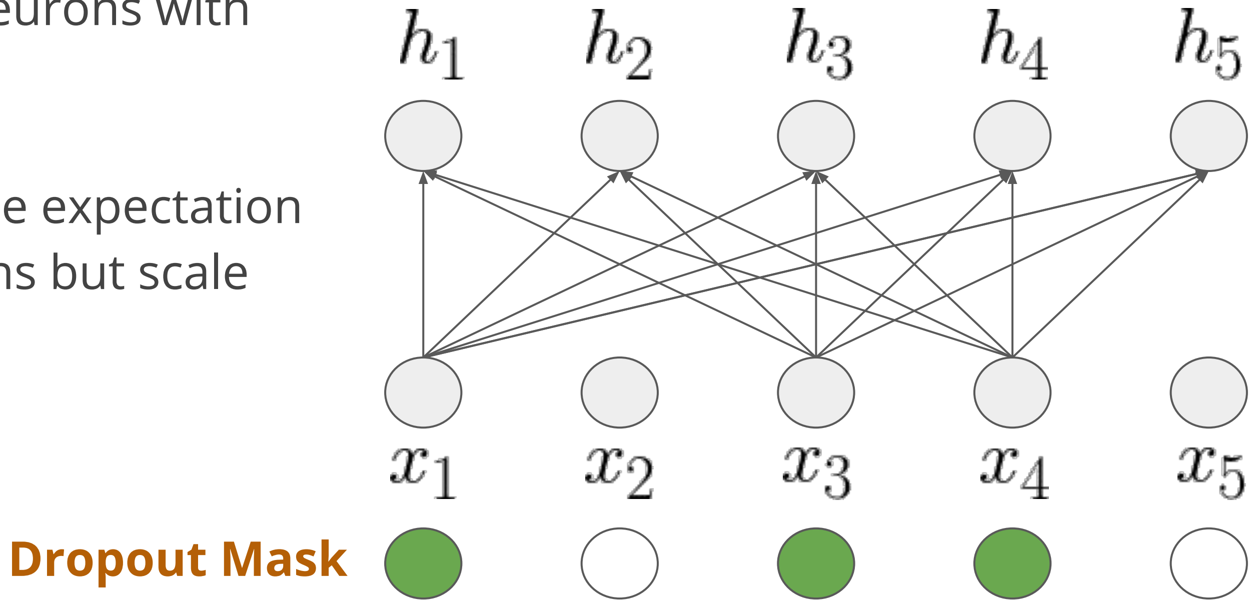
- Why ReLU?
  - Cheap to compute
  - It is roughly linear..



Dropout Regularization

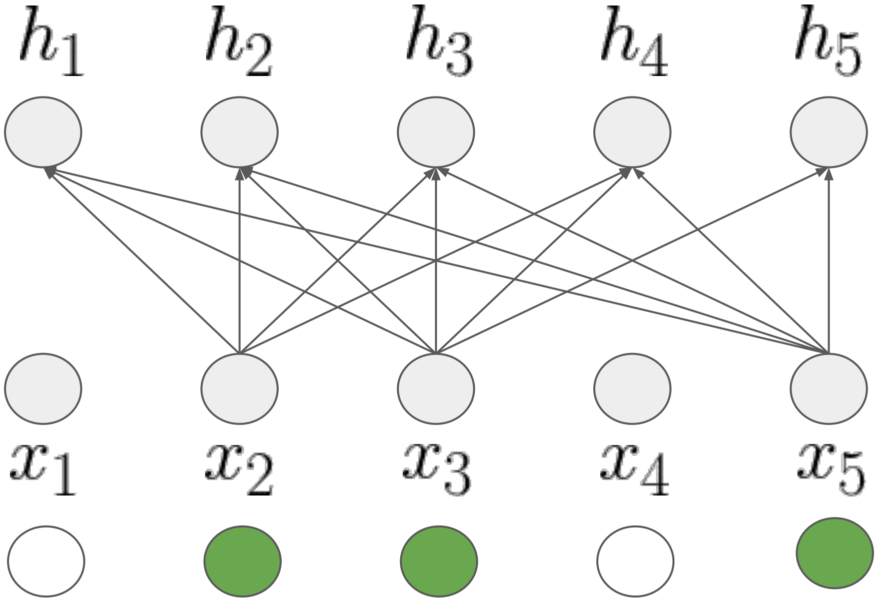
- Randomly zero out neurons with probability 0.5

- During prediction, use expectation value (keep all neurons but scale output by 0.5)



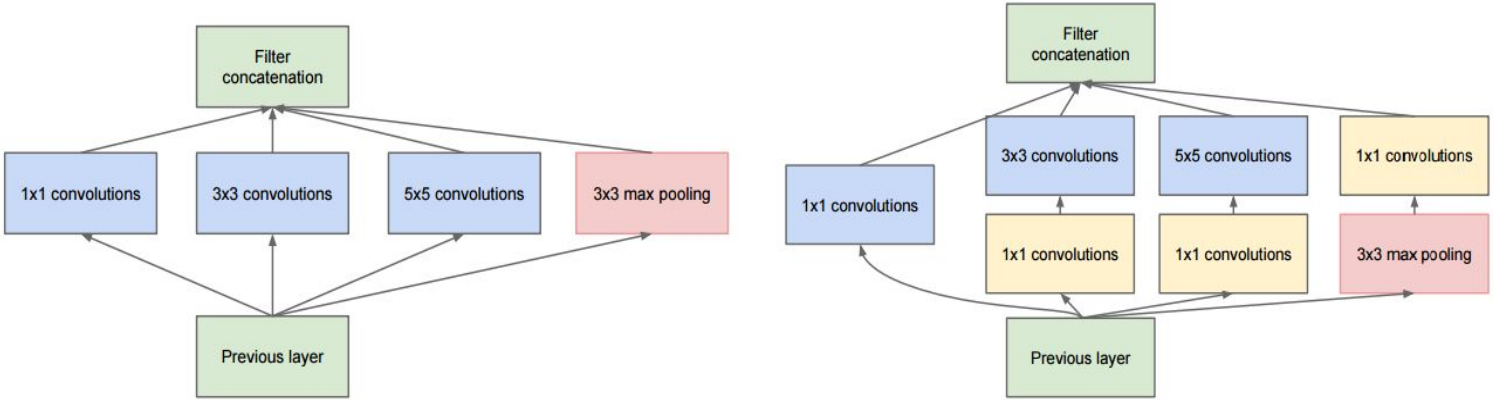
Dropout Regularization

- Randomly zero out neurons with probability 0.5
- During prediction, use expectation value (keep all neurons but scale output by 0.5)



Dropout Mask

GoogleNet: Multiple Pathways, Less Parameters

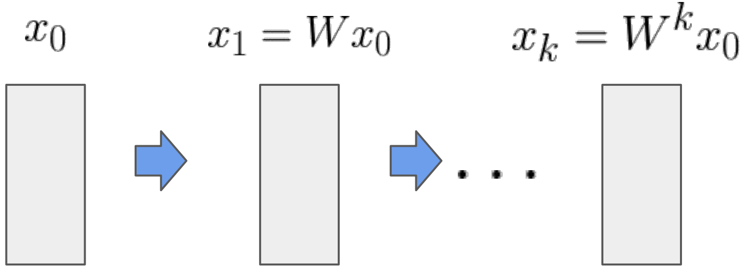


(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

Vanishing and Explosive Value Problem

- Imagine each layer multiplies
  Its input by same weight matrix
  - W > 1: exponential explosion
  - W < 1: exponential vanishing



- In ConvNets, the weight are not tied, but their magnitude matters
  - Deep nets training **was** initialization sensitive

Batch Normalization: Stabilize the Magnitude

- Subtract mean
- Divide by standard deviation
- Output is invariant to input scale!
  - Scale input by a constant
  - Output of BN remains the same

- Impact
  - Easy to tune learning rate
  - Less sensitive initialization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

(Ioffe et.al 2015)

The Scale Normalization (Assumes zero mean)

Scale Normalization

$$BN(x)_i = \frac{x_i}{\sqrt{\sum_{j=1}^{m} x_j^2}}$$

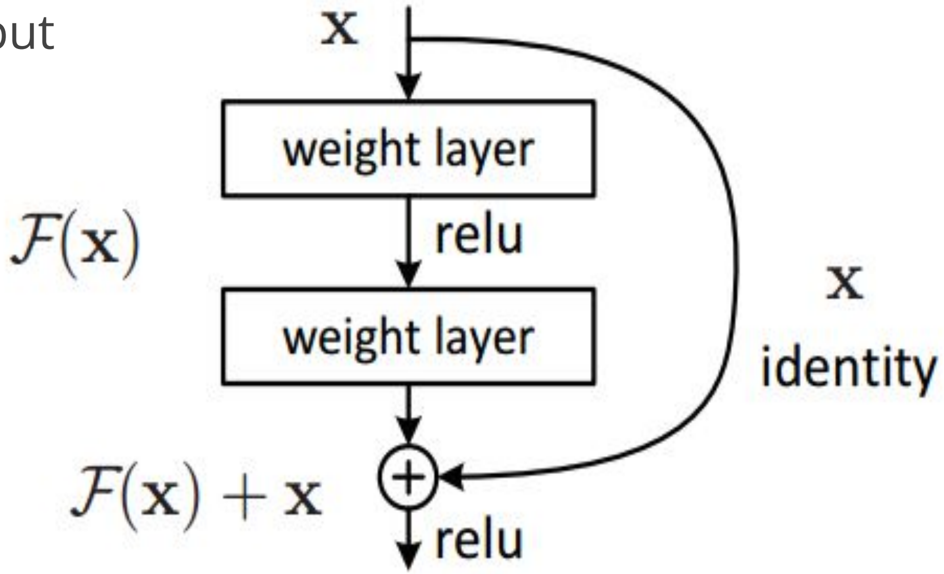
Invariance to Magnitude!

$$BN(\alpha x)_i = \frac{\alpha x_i}{\sqrt{\sum_{j=1}^{m} (\alpha x_j)^2}} = BN(x)_i$$

Residual Net (He et.al 2015)

- Instead of doing transformation add transformation result to input

- Partly solve vanishing/explosive value problem



Evolution of ConvNets

- LeNet (LeCun, 1998)
  - Basic structures: convolution, max-pooling, softmax
- Alexnet (Krizhevsky et.al 2012)
  - ReLU, Dropout
- GoogLeNet (Szegedy et.al. 2014)
  - Multi-independent pass way (Sparse weight matrix)
- Inception BN (Ioffe et.al 2015)
  - Batch normalization
- Residual net (He et.al 2015)
  - Residual pass way

More Resources

- Deep learning book (Goodfellow et. al)
- Stanford CS231n: Convolutional Neural Networks for Visual Recognition
- http://dlsys.cs.washington.edu/materials

Lab1 on Thursday

- Walk through how to implement a simple model for digit recognition using MXNet Gluon
- Focus is on data I/O, model definition and typical training loop
- Familiarize with typical framework APIs for vision tasks

- **Before class**: sign up for AWS educate credits
- https://aws.amazon.com/education/awseducate/apply/
- Create AWS Educate Starter Account to avoid getting charged
- Will email out instructions, but very simple to DIY, so do it today!