Neural Networks

BOSTON UNIVERSITY
MACHINE INTELLIGENCE
COMMUNITY

Justin Chen, Charles Ma
Sept. 26, 2017

MIC Conference

Data: TBA

Project ideas:

Signup:

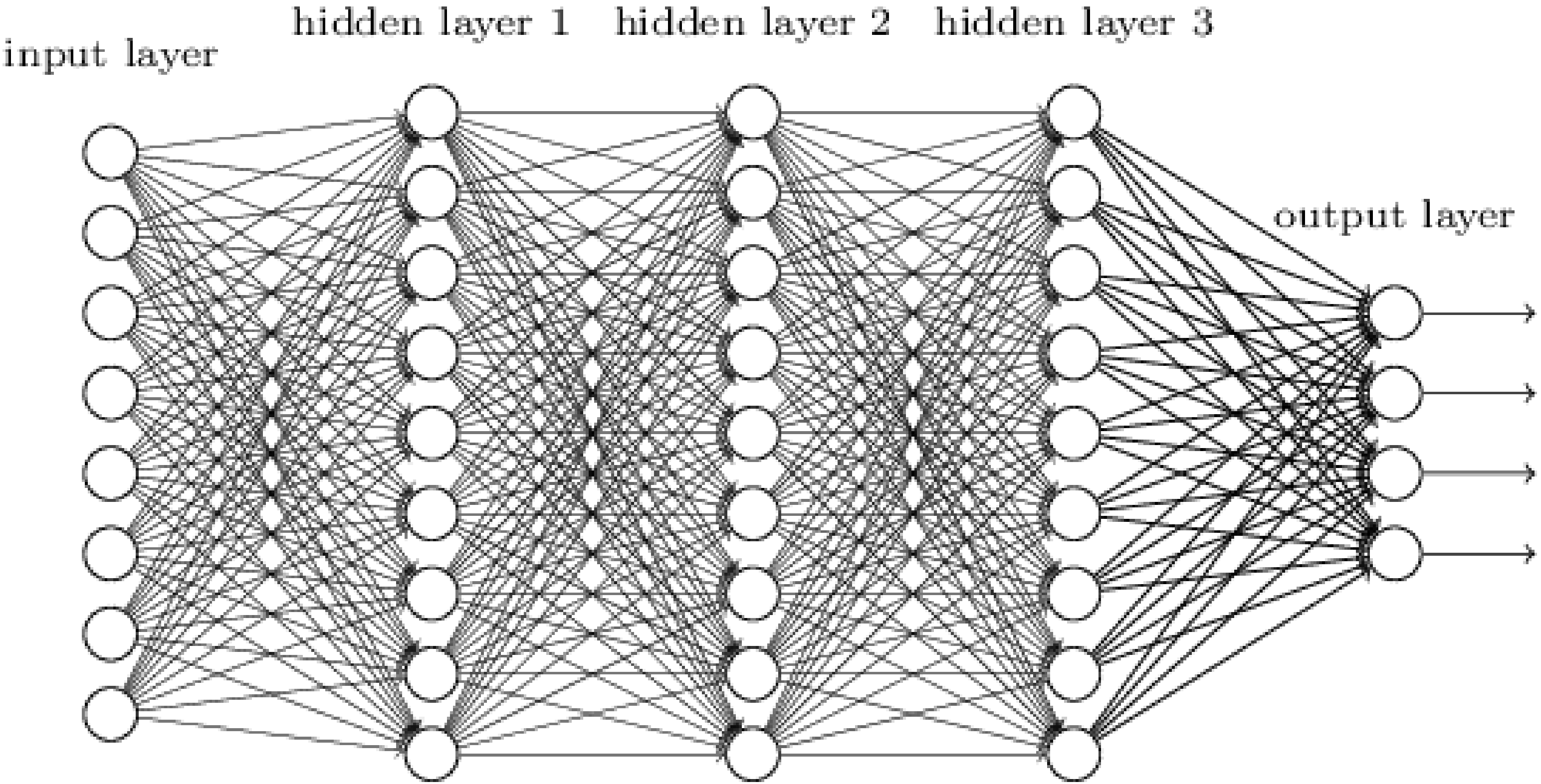
MACHINE INTELLIGENCE COMMUNITY

Brief Recap:

- Data-driven learning
-

MACHINE INTELLIGENCE COMMUNITY

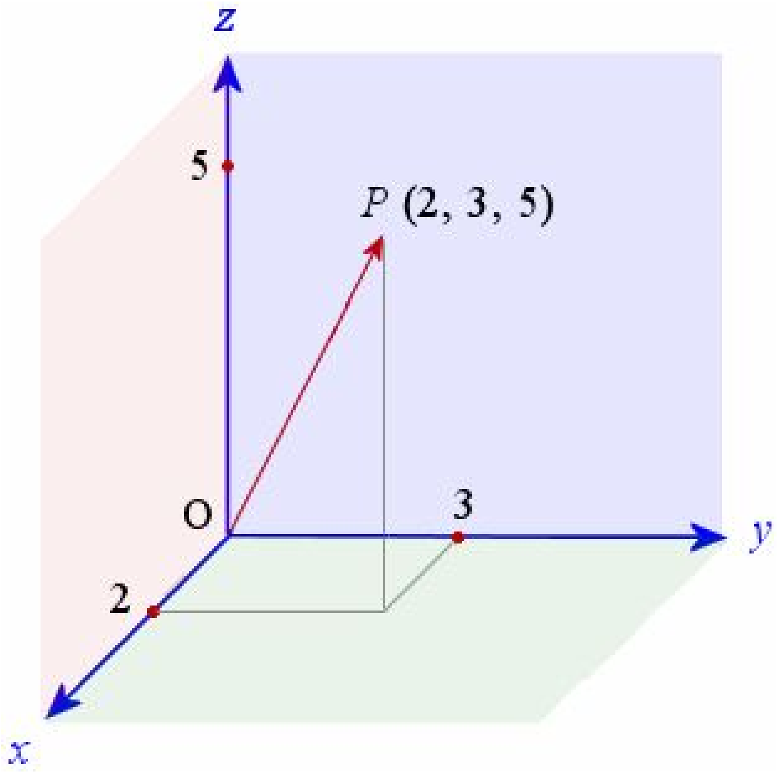
What’s a Neural Network?



MACHINE INTELLIGENCE COMMUNITY

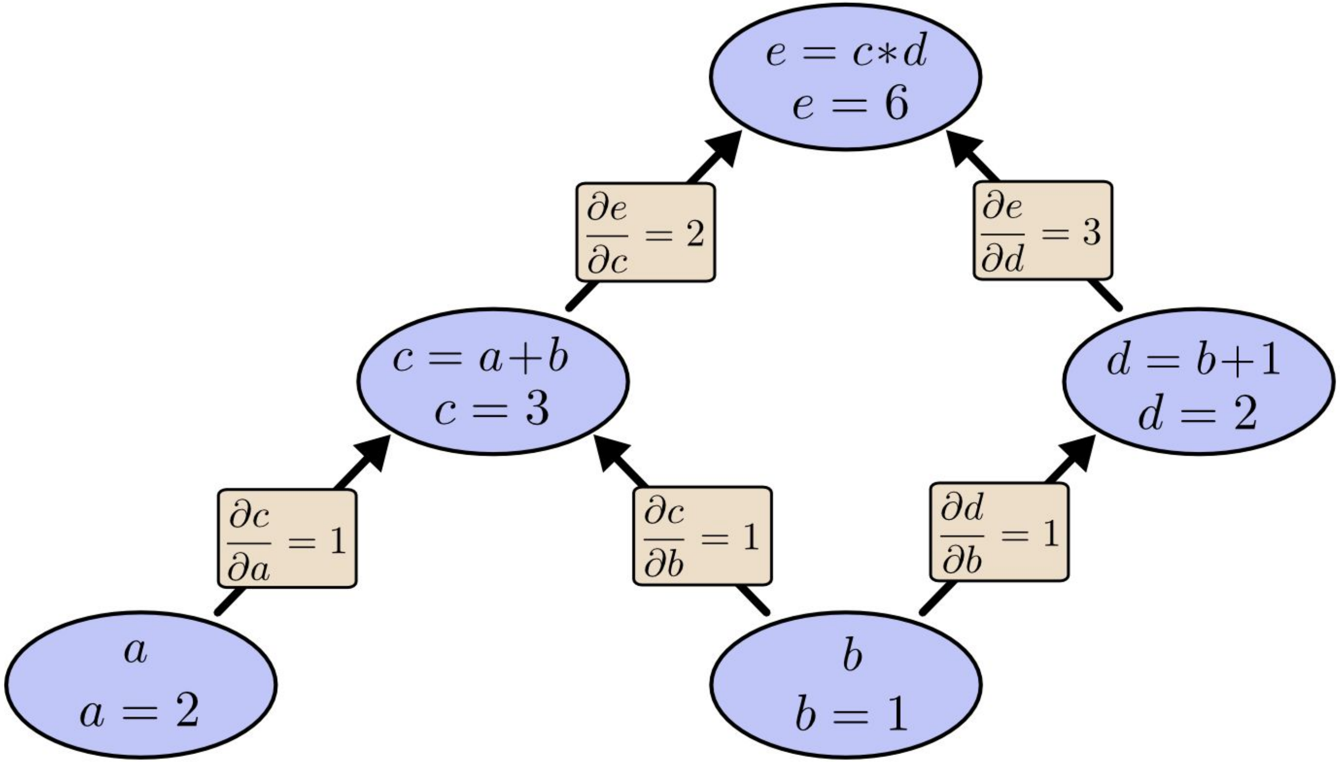
Vectors and Vectorization

- Data is represented in arrays, with each term representing one dimension
- Vectorization is essentially using built-in functions to take advantage of parallel computing capabilities (reduces training time)



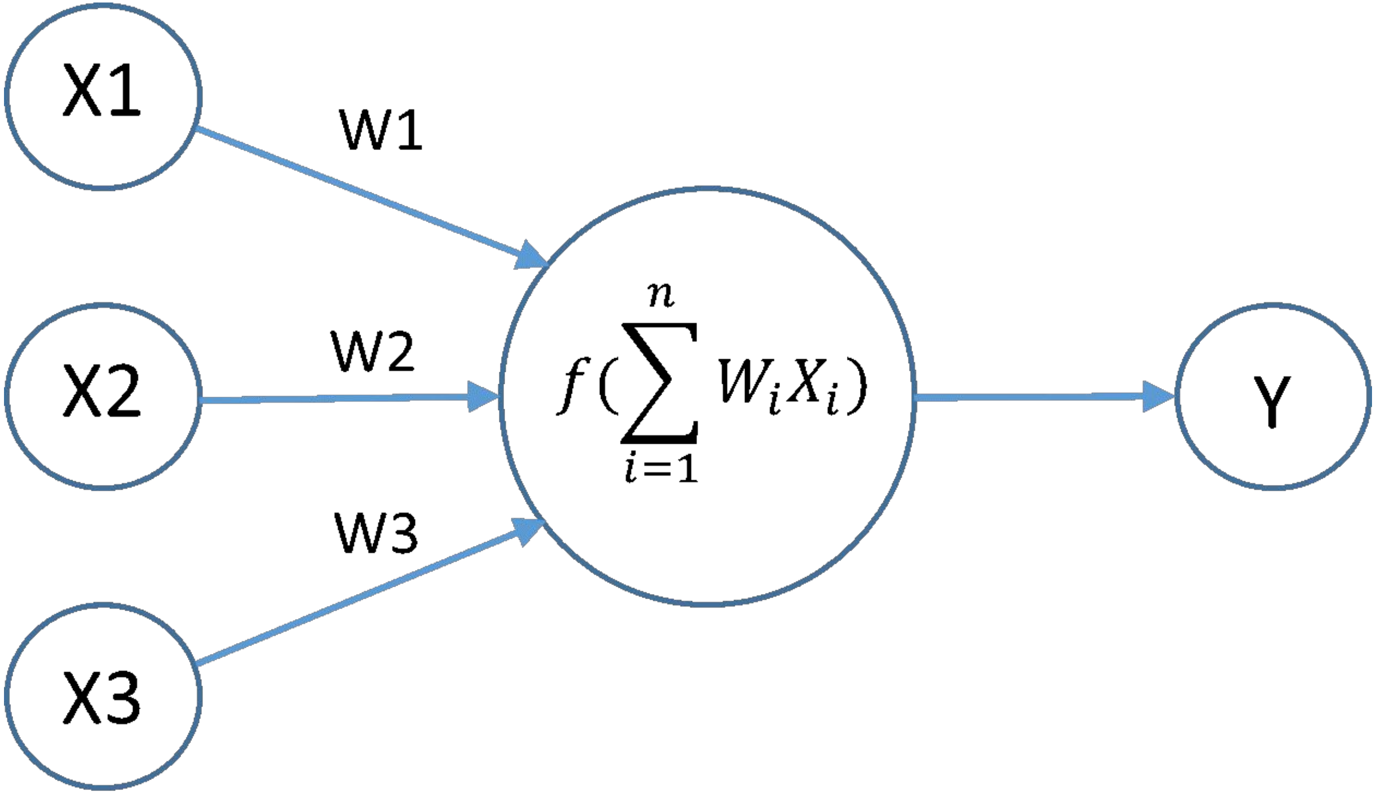
MACHINE INTELLIGENCE COMMUNITY

Computation Graph



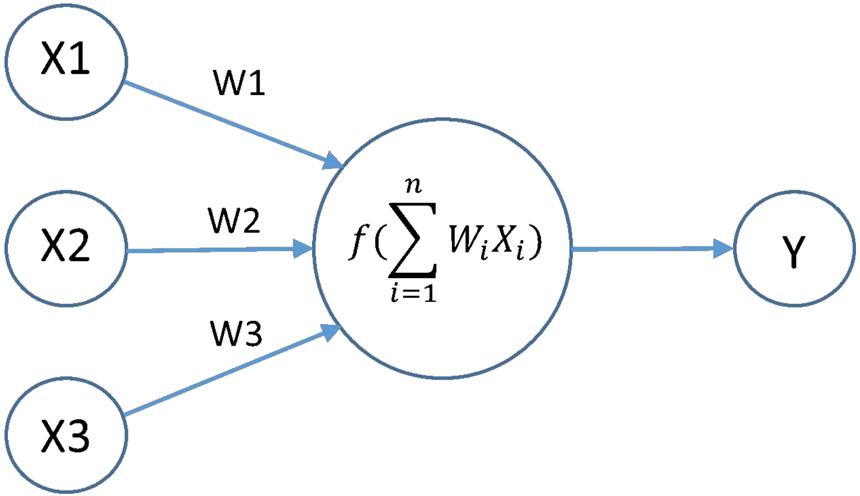
MACHINE INTELLIGENCE COMMUNITY

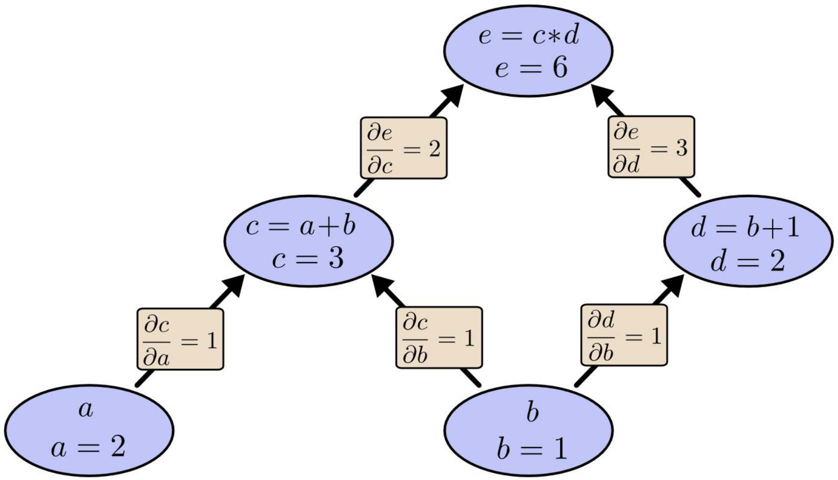
Neural Network (who decides function f)



MACHINE INTELLIGENCE COMMUNITY

Neural Networks as Computational Graphs





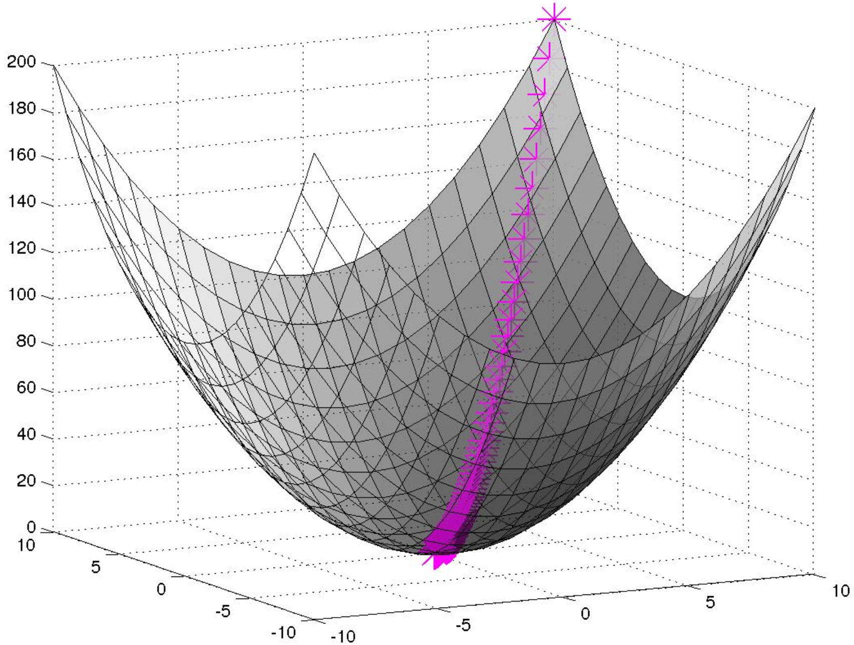
MACHINE INTELLIGENCE COMMUNITY

Gradient Descent

- The process by which your neural net is trained to be accurate
- Through numerous

  Iterations of logistic

  regression
- Reach the global minima!
-



MACHINE INTELLIGENCE COMMUNITY

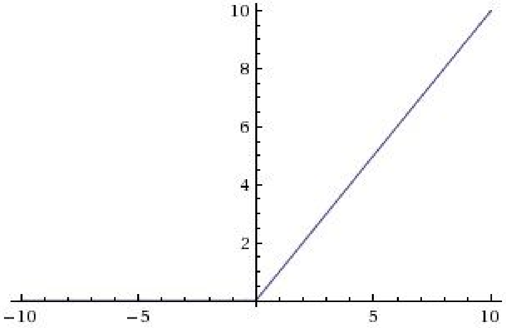
Logistic Regression Cost Function

- Used when output of neural network is 0 (false) or 1 (true)

Activation Functions

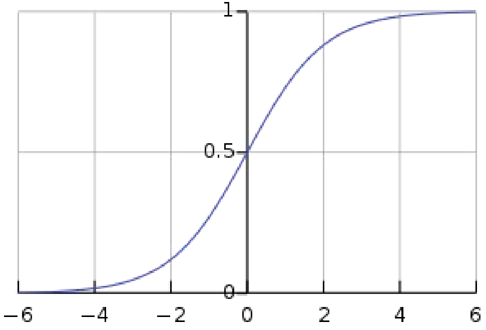
ReLU

(Rectified Linear Unit):



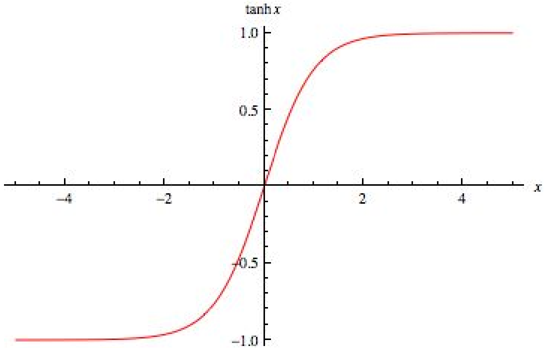
max(0,x)

Sigmoid ($\boldsymbol{\sigma}$):



1/(1+e^-x)

Tanh:



2*$\boldsymbol{\sigma}$(2x)-1

Levels of Supervision During Learning

Supervised Learning - you *have* labeled data; your data comes in pairs of input and desired output.

Unsupervised Learning - you *do not have* labeled data; your neural net looks to group different inputs based on similarities i.e. *clustering.*

Semi-supervised Learning - you have *some* labeled data; labeled data is expensive to generate, so you use a mix of labeled and unlabelled.

MACHINE INTELLIGENCE COMMUNITY

A Tidbit on the Universal Approximation Theorem

A multilayer perceptron can approximate *continuous functions* on a compact subset of real numbers with mild assumptions on the activation function. (Thanks Wikipedia)

MACHINE INTELLIGENCE COMMUNITY

A Basic Multilayer Perceptron with PyTorch

```python
import torch
import torch.nn as nn

class MLP(nn.Module):

    # one hidden layer
    def __init__(self, num_inputs, num_hiddens):
        super(MLP, self).__init__()
        self.fc1 = nn.Linear(num_inputs, num_hiddens)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(num_hiddens, 1)

    def forward(self, x):
        output = self.fc1(x)
        output = self.relu(output)
        output = self.fc2(output)
        return output

net = MLP(5, 3)
print(net)

parameters = list(net.parameters())
print(parameters)
print(len(parameters))
print(parameters[0].size())
```

Output:

```
MLP (
  (fc1): Linear (5 -> 3)
  (relu): ReLU ()
  (fc2): Linear (3 -> 1)
)
[Parameter containing:
-0.1721  0.0375 -0.4098  0.0769 -0.4214
 0.0392 -0.1720  0.3798  0.1420 -0.4239
 0.3698  0.3336 -0.4313  0.3198  0.2039
[torch.FloatTensor of size 3x5]
, Parameter containing:
-0.2104
 0.0990
 0.1316
[torch.FloatTensor of size 3]
, Parameter containing:
-0.2340 -0.2816  0.4150
[torch.FloatTensor of size 1x3]
, Parameter containing:
 0.1908
[torch.FloatTensor of size 1]
]
4
torch.Size([3, 5])
```

MACHINE INTELLIGENCE COMMUNITY

Paper Discussion Time!

http://cs.stanford.edu/~quocle/tutorial1.pdf

Read through, and we will reconvene in ~5minutes

**Feel free to start discussing with your peers if you finish beforehand!

MACHINE INTELLIGENCE COMMUNITY

Real World Applications

Image Identification (CNNs)

Natural Language Processing (RNNs)

MACHINE INTELLIGENCE COMMUNITY

Additional Resources

Andrew Ng's deeplearning.ai course (highly recommend!):

https://www.coursera.org/learn/neural-networks-deep-learning/lecture/Cuf2f/welcome

Segway for Next Workshop: PyTorch 60minute Blitz (with CNNs!):

http://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

Challenge Yourself: Efficient Backprop by Yann LeCun

http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf

MACHINE INTELLIGENCE COMMUNITY