PyTorch Tutorial
CSE455

Kiana Ehsani

What is PyTorch?

PYTORCH

Caffe

TensorFlow

K

PyTorch takes care of everything

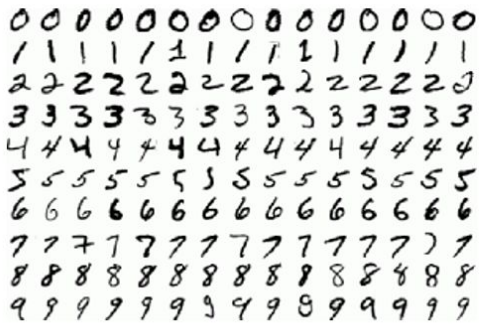
Installation

https://pytorch.org/

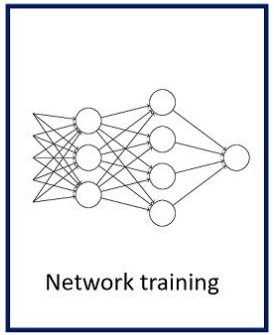
https://github.com/ehsanik/pytorch_installation

An example

MNIST classifier

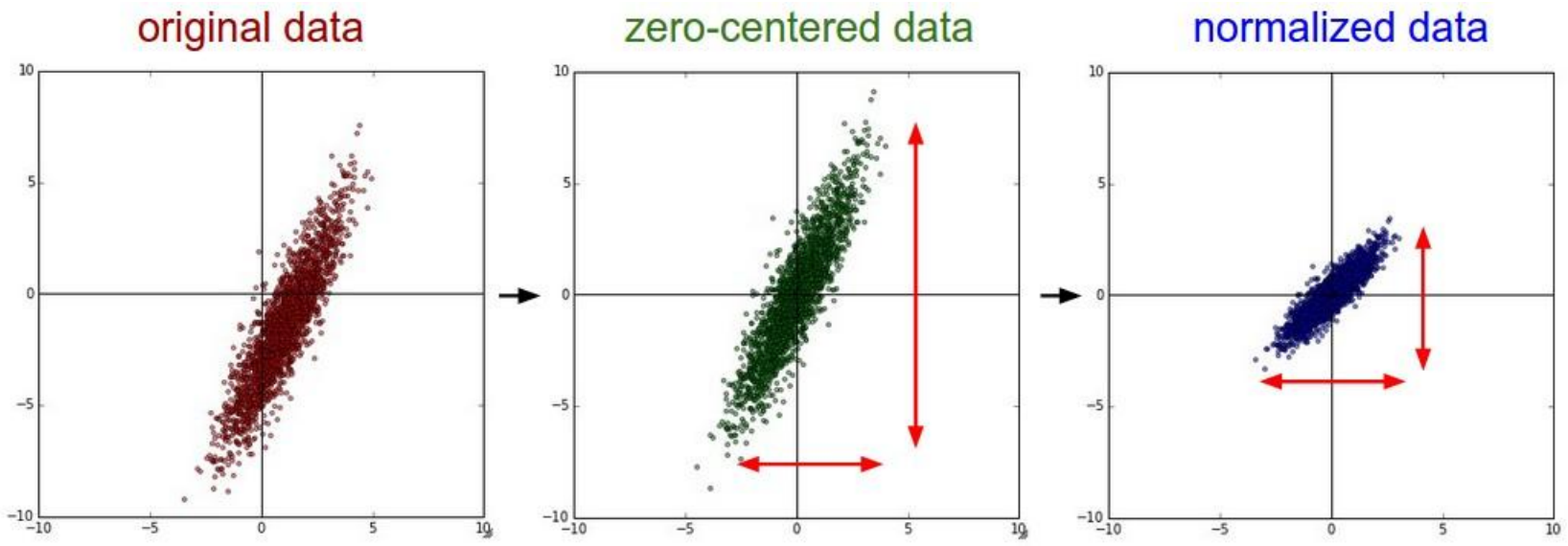


Data & Labels



0
1
2
3
4
5
6
7
8
9

Data normalization



Modules, Layers and more :)

All networks need to be a child class of nn.Module

1. Conv1d/2d
2. MaxPool1d/2d
3. MaxUnpool1d/2d
    a. MaxUnpool1d takes in as input the output of MaxPool1d including the indices of the maximal values and computes a partial inverse in which all non-maximal values are set to zero.
4. Linear
5. Upsample
6. And tons of others

Initialization

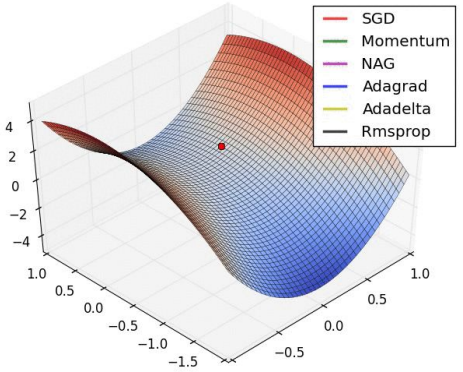
1. All Zero -> `constant_`
   a. PLEASE DON'T!
2. Identity -> `eye_`
3. Random
   a. Normal distribution -> `normal_`
   b. Uniform distribution -> `uniform_`
   c. Xavier
   d. Kaiming

Published as a conference paper at ICLR 2016

ALL YOU NEED IS A GOOD INIT

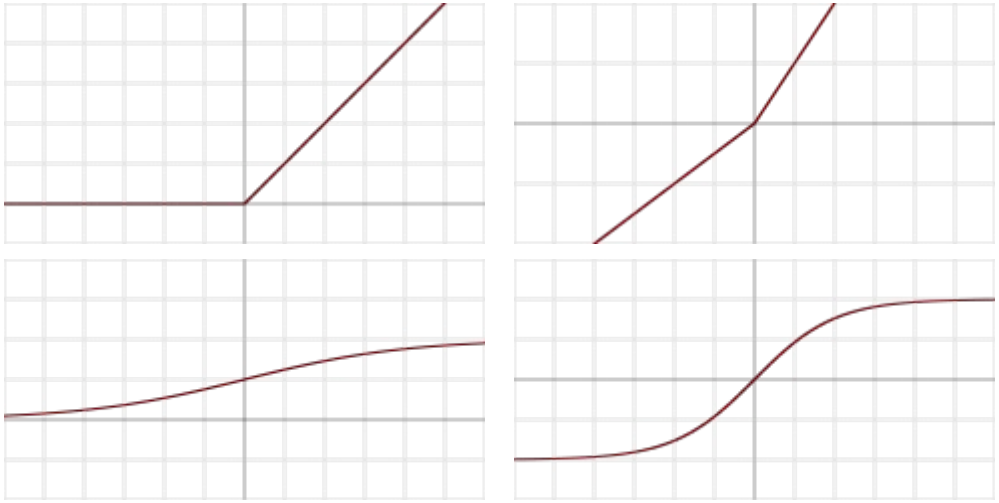
Dmytro Mishkin, Jiri Matas

Center for Machine Perception
Czech Technical University in Prague
Czech Republic {mishkdmy,matas}@cmp.felk.cvut.cz



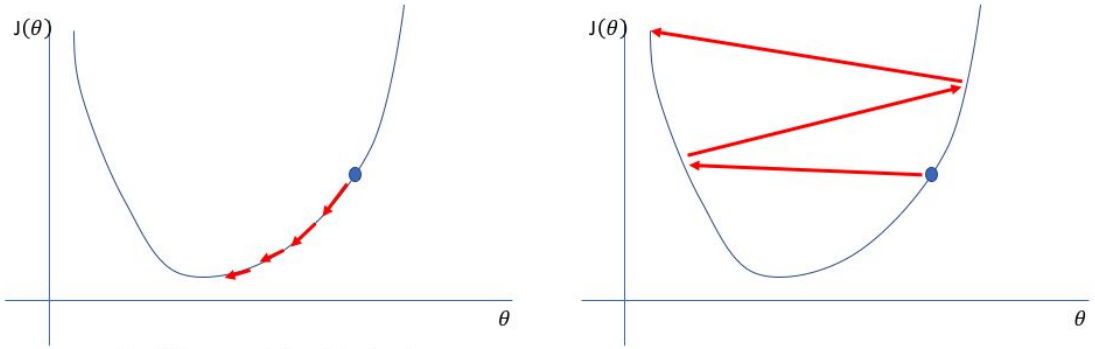
Non-linear activations

1. What does it mean?
    a. Neuron is firing!
    b. Binary! Either 0 or 1
2. What do we have?
    a. ReLU
    b. LeakyReLU
    c. Sigmoid
    d. Tanh
    e. Softmax
    f. And tons of others



Normalizations

1. More stable
2. Reduces the amount by which the hidden unit values shift around

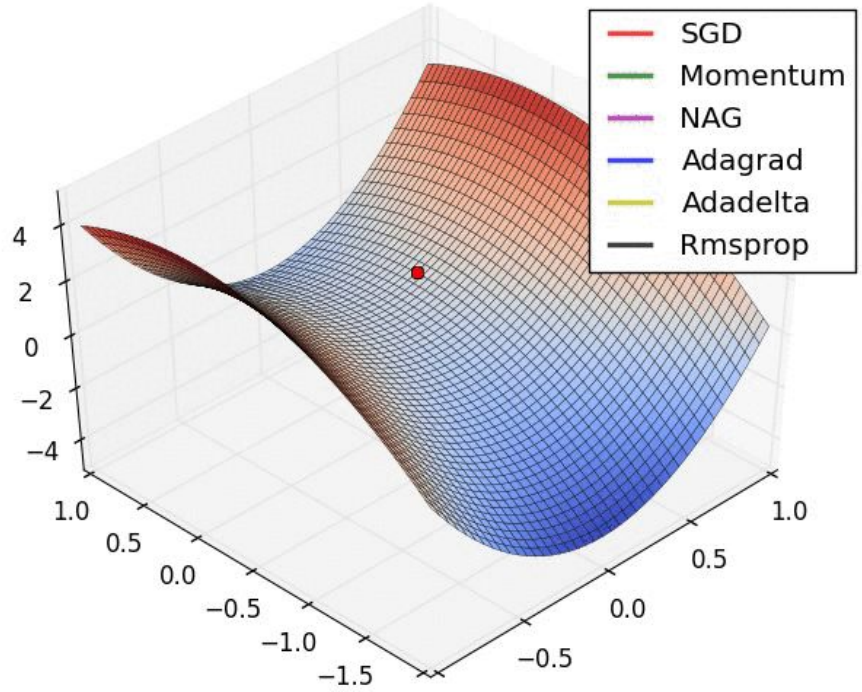


Loss functions

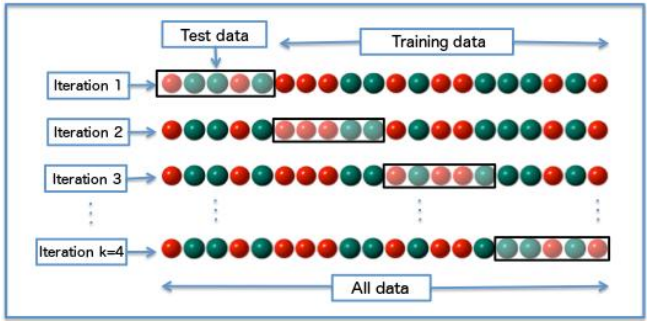
1. L1Loss
2. MSELoss
3. CrossEntropyLoss
4. BCELoss
5. KL Divergence Loss
    a. Useful when direct regression over a distribution
6. And tons of others
7. Implement your own loss function

Optimizers

1. SGD
2. Adagrad
3. Adam
4. Adadelta
5. And tons of others



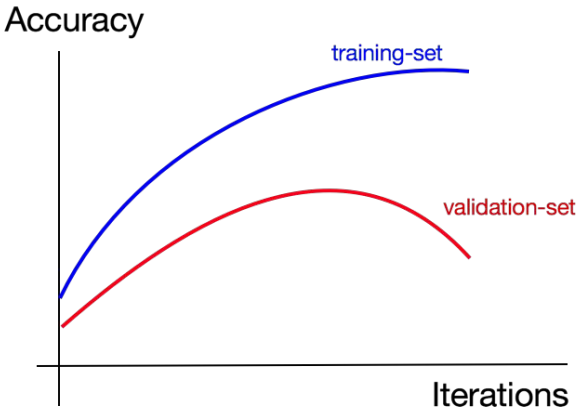
Train, Test, Val (Dev)

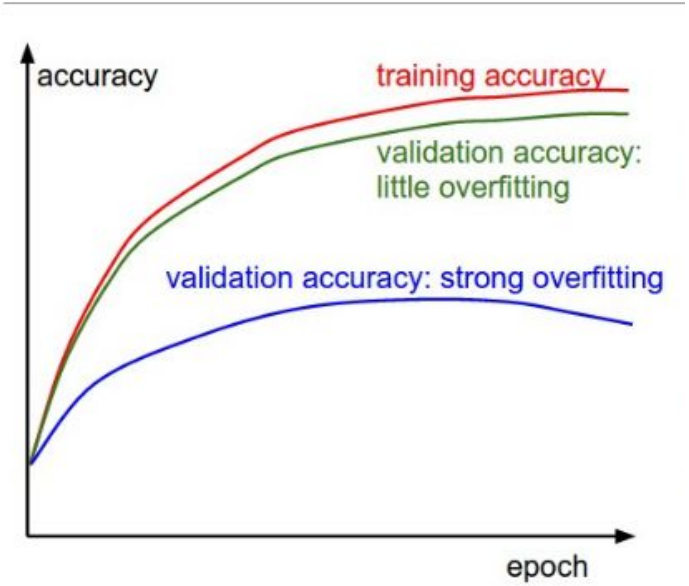


Issues you might face

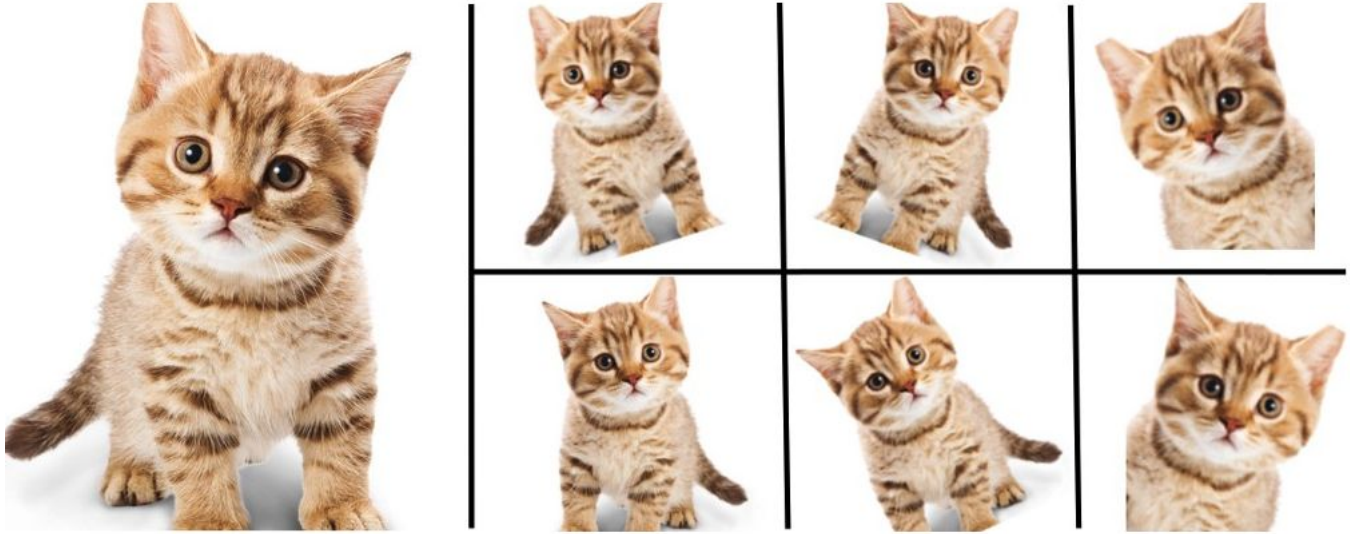
1. Overfitting
2. Getting stuck in local minima
3. Diverging

Overfitting





Data transformations

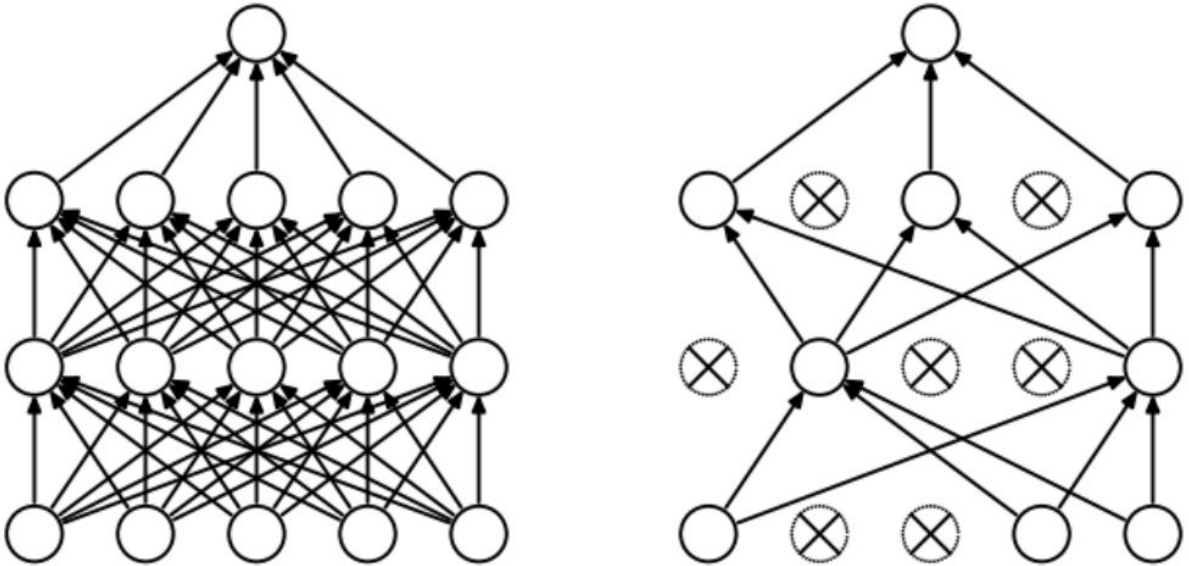


Data transformations

1. Grayscale
2. ColorJitter
3. LinearTransformation
4. Pad
5. RandomAffine
6. RandomCrop
7. RandomHorizontalFlip
8. RandomRotation

Be careful! It depends on the dataset.

Regularization: Dropout



(a) Standard Neural Net

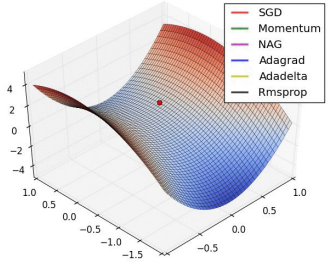
(b) After applying dropout.

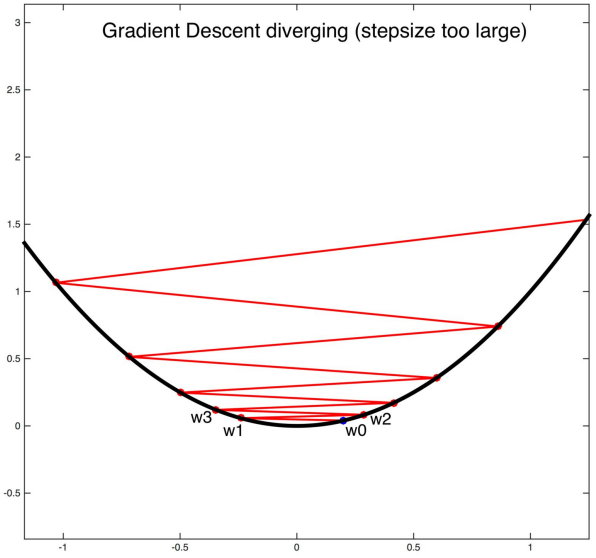
Dropout vs Batchnorm

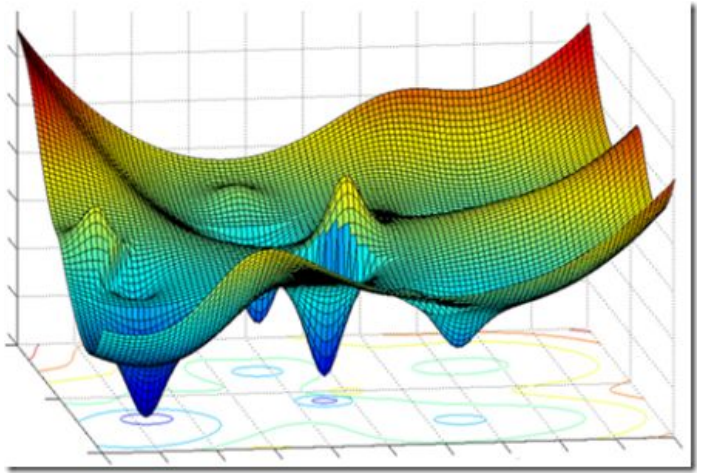
Batch normalization

Dropout

Diverging and local minima







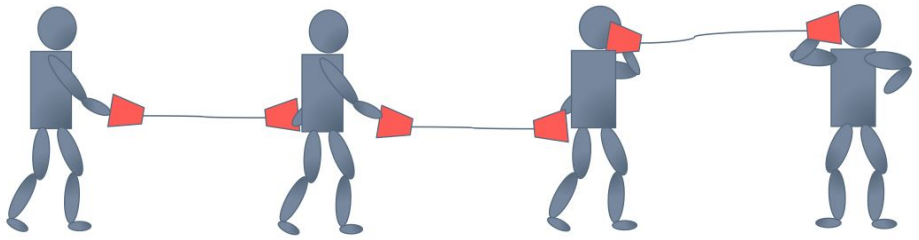
Project Ideas

Datasets:

https://docs.google.com/spreadsheets/d/1pXCrtVdPYOJthsBaMyRLQz6jLCor55vbcoKgetSvexU/edit#gid=0

Ideas? Brain Storming!

Backstage



A graph is created on the fly

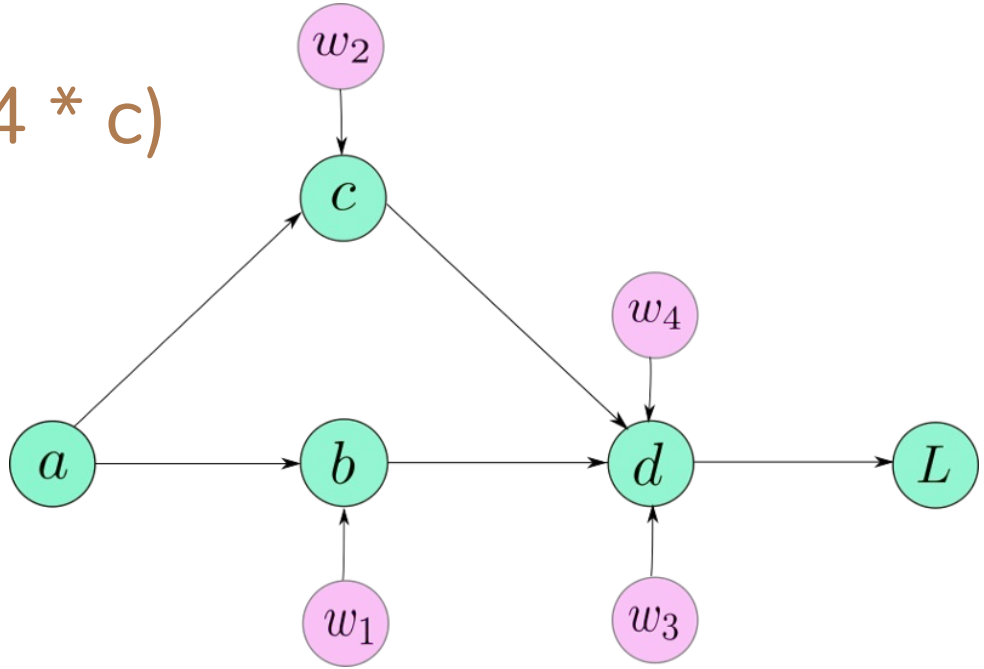

```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```

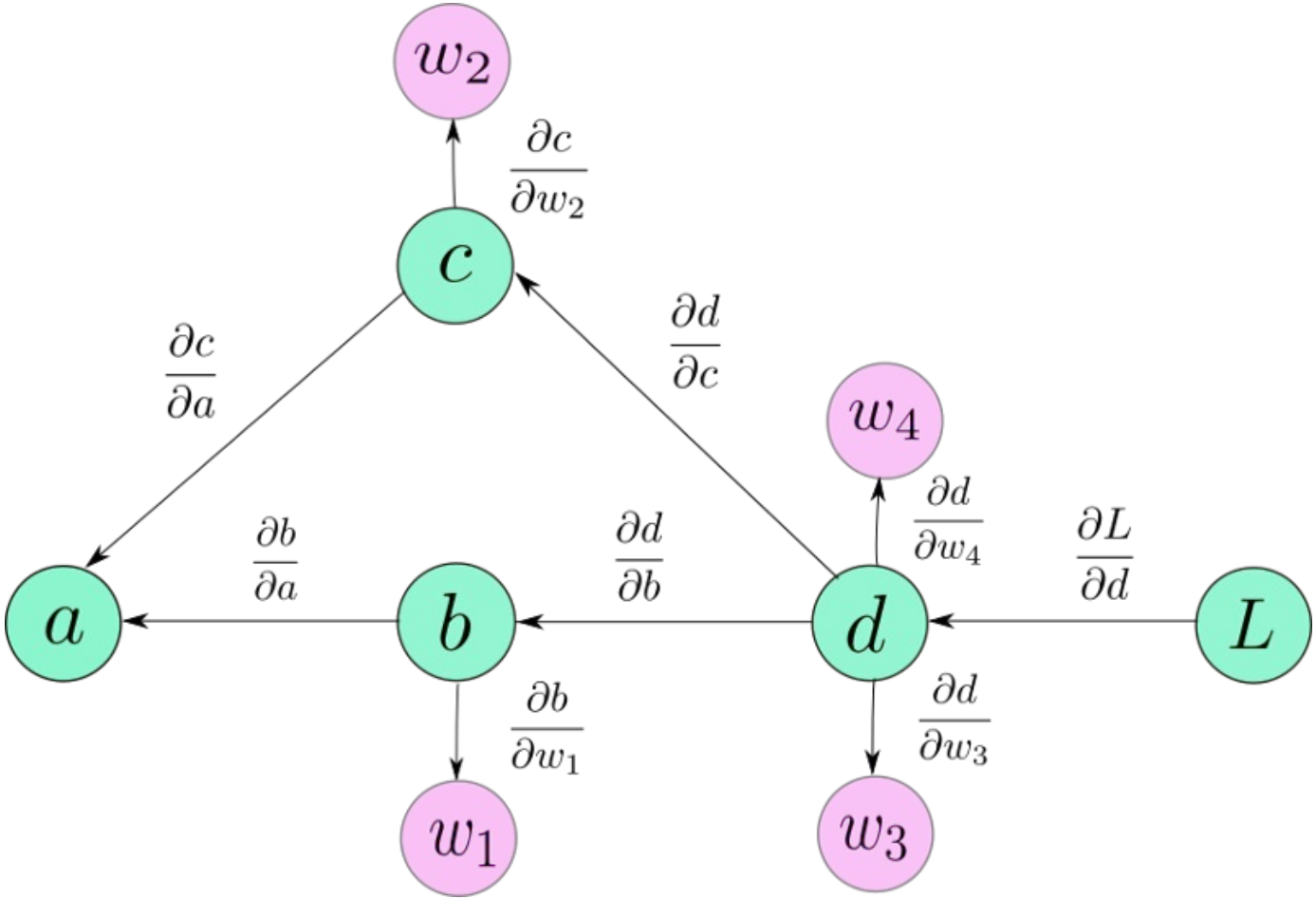
b = w1 * a

c = w2 * a

d = (w3 * b) + (w4 * c)

L = f(d)




w_2
$\frac{\partial c}{\partial w_2}$
c
$\frac{\partial c}{\partial a}$
$\frac{\partial d}{\partial c}$
w_4
$\frac{\partial d}{\partial w_4}$
$\frac{\partial L}{\partial d}$
$\frac{\partial b}{\partial a}$
$\frac{\partial d}{\partial b}$
a
b
d
L
$\frac{\partial b}{\partial w_1}$
$\frac{\partial d}{\partial w_3}$
w_1
w_3