Week 3

Introduction to neural networks and deep learning

Topics you have covered in week 3 videos

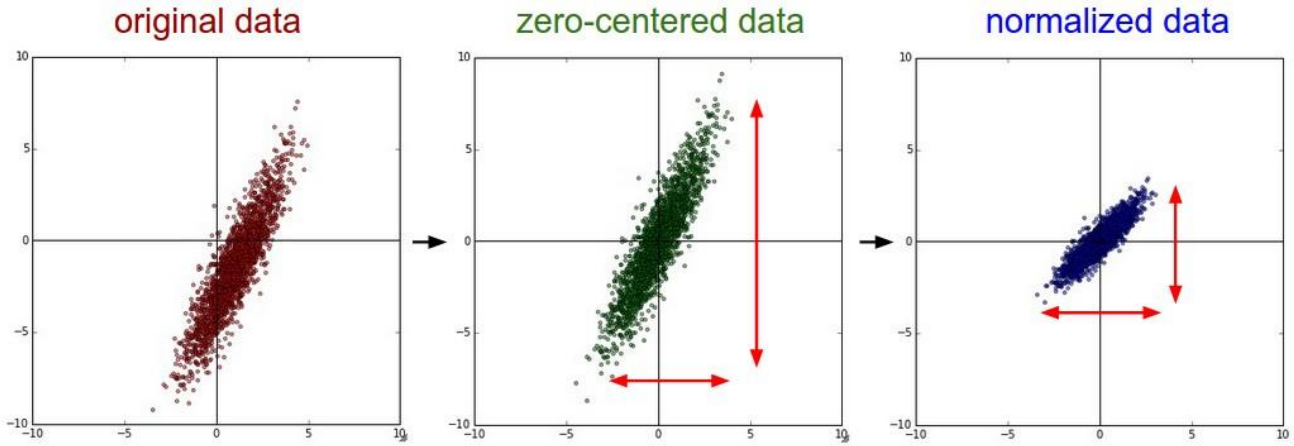
- Intro to babysitting and learning process
- Data preprocessing
- Data augmentation
- Weight initialization
- Regularization
  - Batch normalization
  - Dropout
- Hands on demo

Session agenda

- Data preprocessing
- Data augmentation
- Weight initialization
- Regularization
- Case study
- Questions

Data preprocessing

Data preprocessing

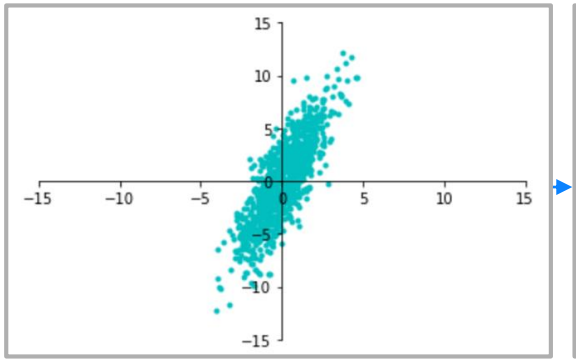


Mean subtract

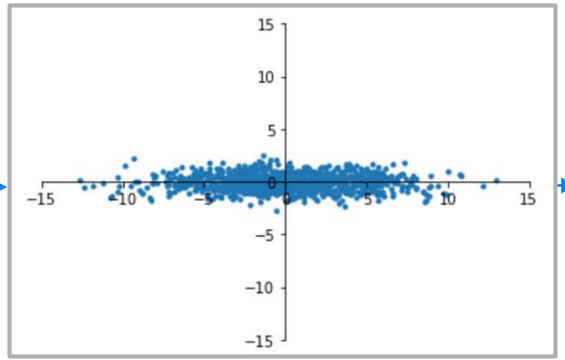
Divide by standard deviation

Data preprocessing

Original Data

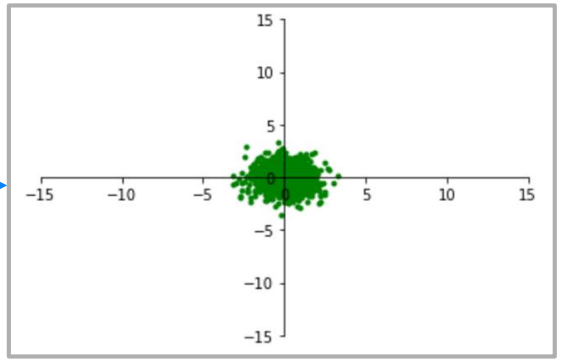


Decorrelated Data

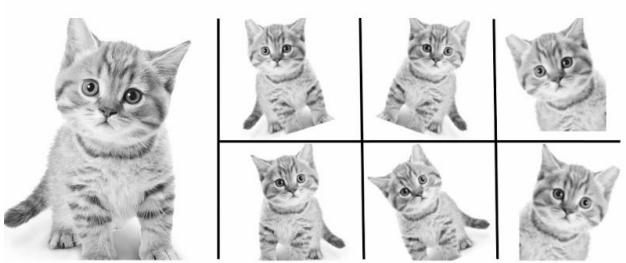


data has diagonal covariance matrix

Whitened Data



covariance matrix is the identity matrix



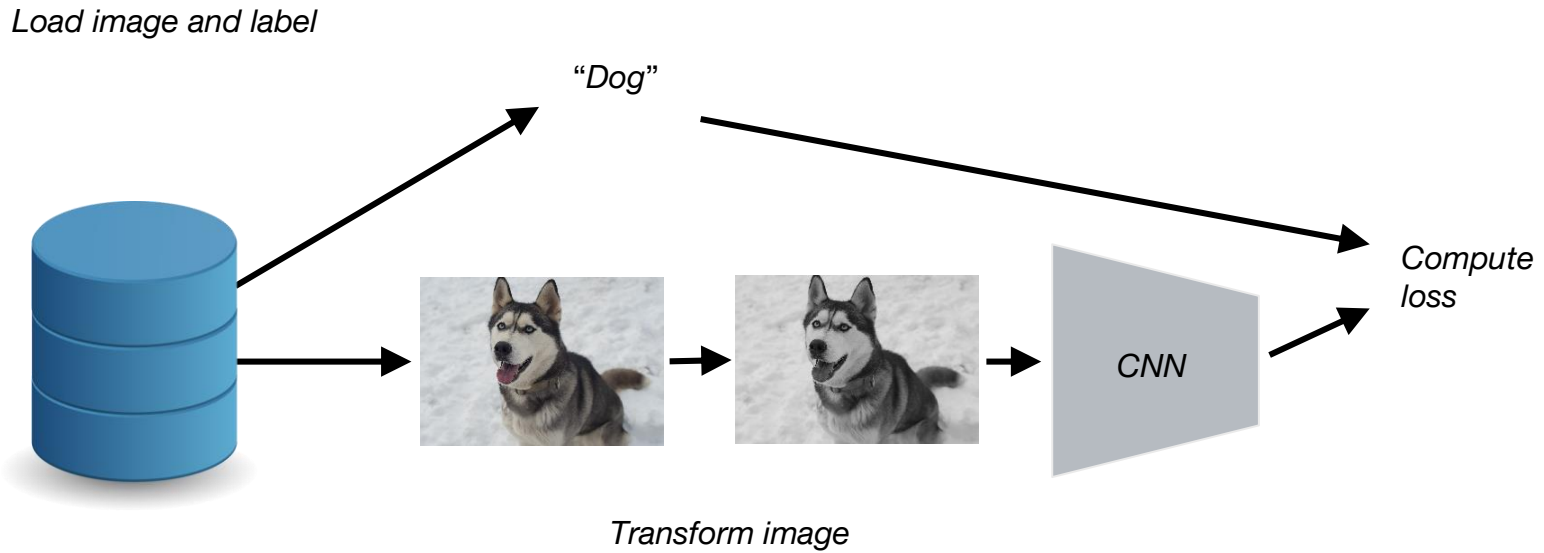
Data augmentation

Image source: medium

Why we should do data augmentation?

- We may not have a big dataset, so create more data.
- It helps in regularizing the network.

Data augmentation pipeline



Data augmentation techniques

- Horizontal flips
- Rotation
- Crop/scale
- Color jitter
- Other creative techniques
  - Random mix/combinations of :
    - translation (what about a pure ConvNet?)
    - Rotation
    - Stretching
    - Shearing
    - lens distortions, … (go crazy)

Data augmentation

- Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips
- Very easy to implement, give good results especially on small datasets
- Easily fits into framework of noise / marginalization

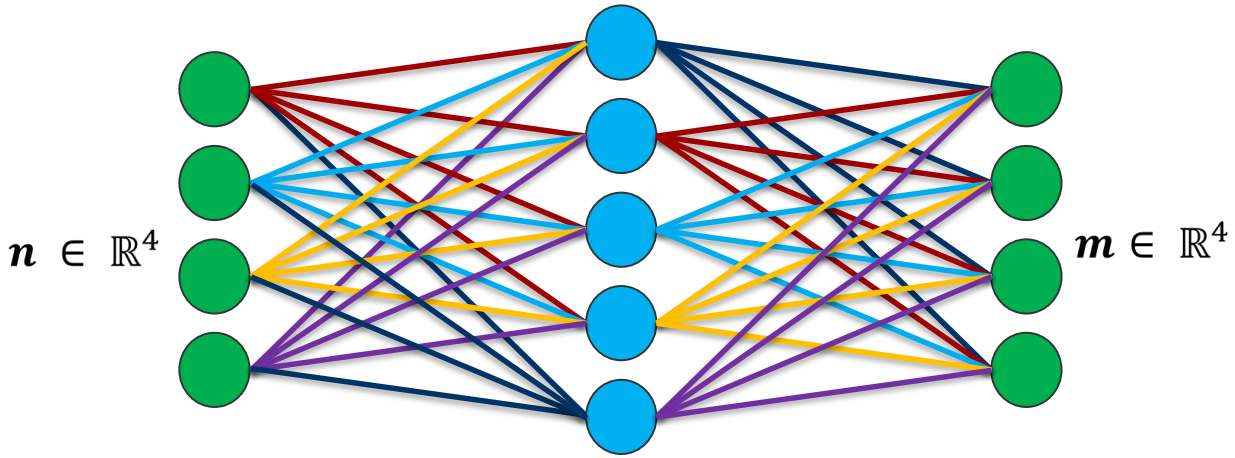
Weight initialization

Why initialize weights?

The aim of weight initialization is to prevent layer activation outputs from exploding or vanishing during the course of a forward pass through a deep neural network. If either occurs, loss gradients will either be too large or too small to flow backwards beneficially, and the network will take longer to converge, if it is even able to do so at all.

Text source: medium

What happens when W=0 init is used?



Initialization techniques

- Zero initialization
- Random initialization
- Xavier initialization
- He initialization
- Kaiming initialization
- And many more

Regularization

Batch normalization

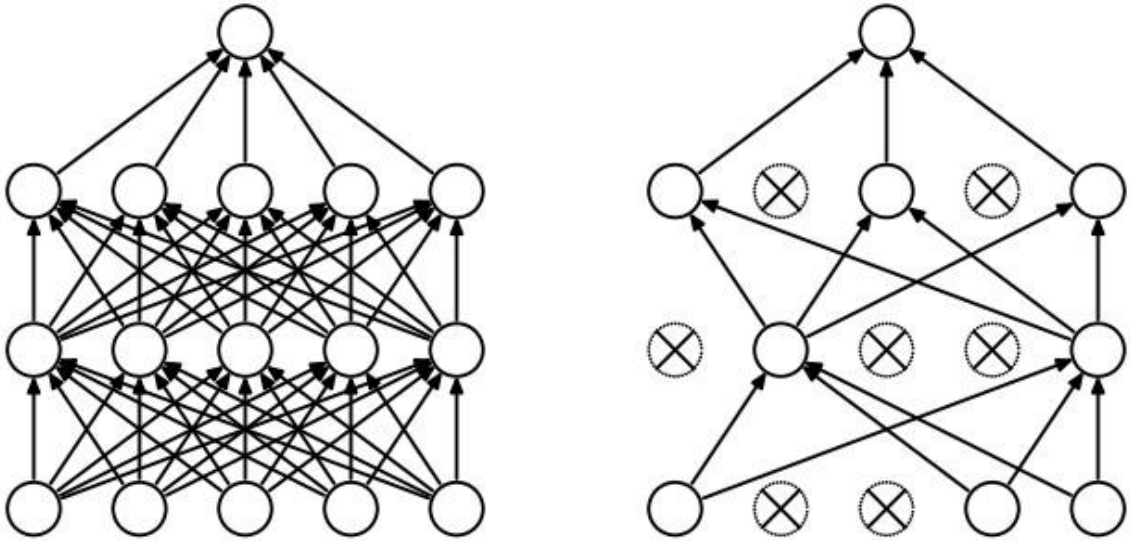
- Improves gradient flow through the network
- Allows higher learning rates
- Reduces the strong dependence on initialization
- Acts as a form of regularization in a funny way, and slightly reduces the need for dropout, maybe

Dropout

Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel.

During training, some number of layer outputs are randomly ignored or “dropped out.” This has the effect of making the layer look-like and be treated-like a layer with a different number of nodes and connectivity to the prior layer. In effect, each update to a layer during training is performed with a different “view” of the configured layer.

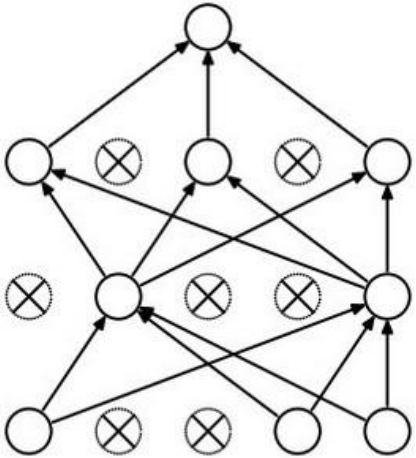
Dropout



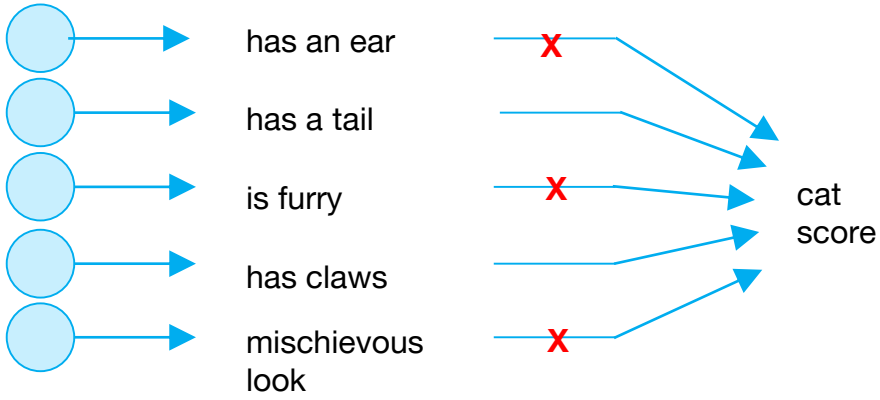
(a) Standard Neural Net

(b) After applying dropout.

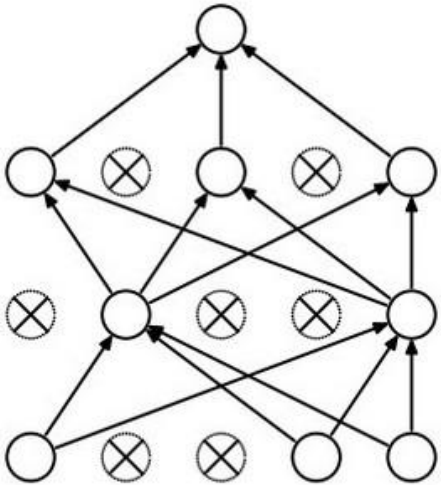
Dropout: how it works?



Forces the network to have a redundant representation.



Dropout



Another interpretation:

Dropout is training a large ensemble of models (that share parameters).

Each binary mask is one model, gets trained on only ~one batch.

Summary

We have learned about...

- Data preprocessing
- Data augmentation
- Weight initialization
- Batch normalization
- Dropout

Thank you! :)

Questions are always welcome