Deep Learning Architectures

Giri Iyengar

Cornell University

gi43@cornell.edu

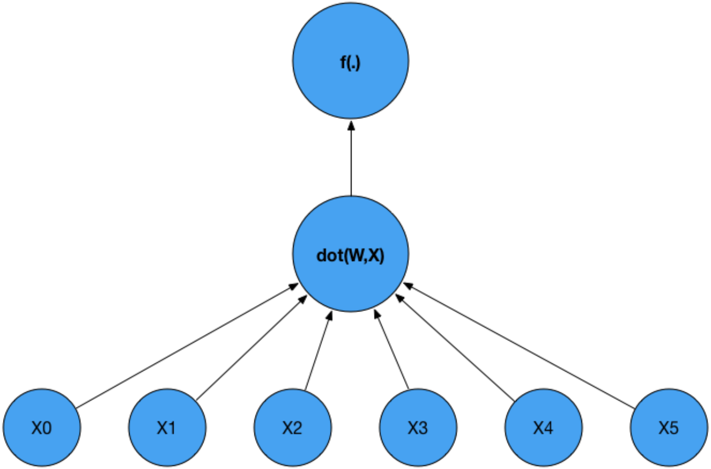
Feb 14, 2018

Agenda for the day

- Deep Learning Architectures
- Convolutional Neural Nets
- Stacked denoising Auto Encoders
- Recurrent Neural Networks
- Sequence to Sequence models

Overview

Quick Recap - Neuron



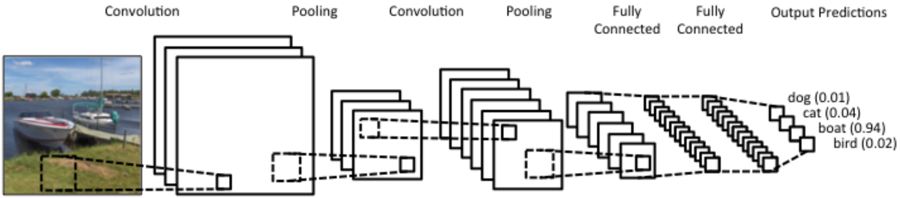
Some great tutorials

- Deep Learning Tutorials
- PyTorch Tutorials
- Plus many more on GitHub, Caffe, TensorFlow etc

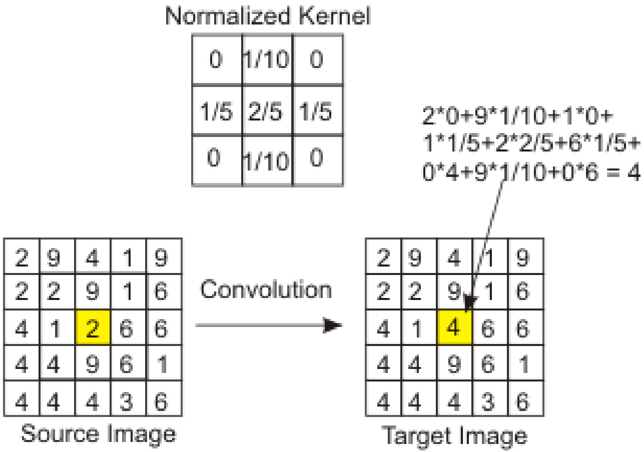
Convolutional Neural Networks

- Biologically-inspired variants of MLP
- Encode a notion of a **visual receptive field** in the network
- Exploit local correlations by enforcing connectivity between adjacent layers
- Shared weights / replicated units - greatly reduces the number of parameters

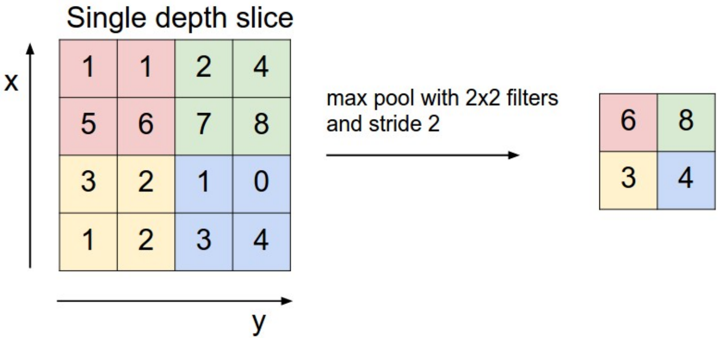
CNN Architecture – Typical



Convolutional Filter

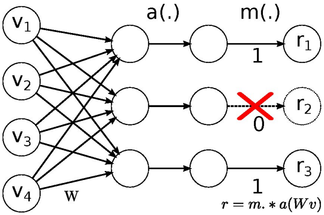


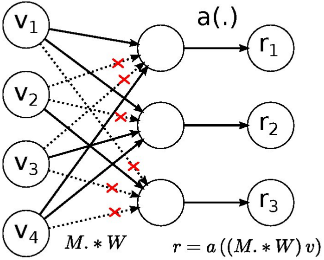
Pooling Layer



Dropouts/DropConnects

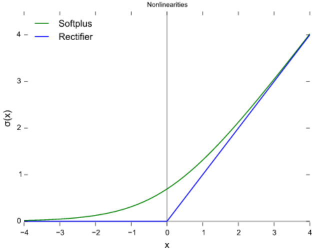
- Randomly sets a fraction of the inputs to zero during training time
- Forces the weights in the network to not **rely** on neighboring nodes
- A type of **Regularization**



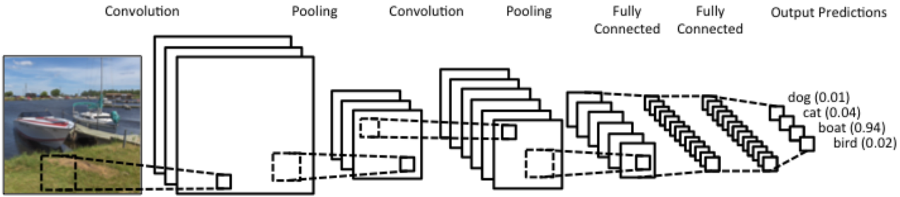


Nonlinearities that are commonly used

- ReLU - Rectified Linear Unit
- Softplus
- Tanh, sigmoid, softmax, leaky and noisy variants of ReLU



CNN Architecture – Typical



denoising Autoencoders

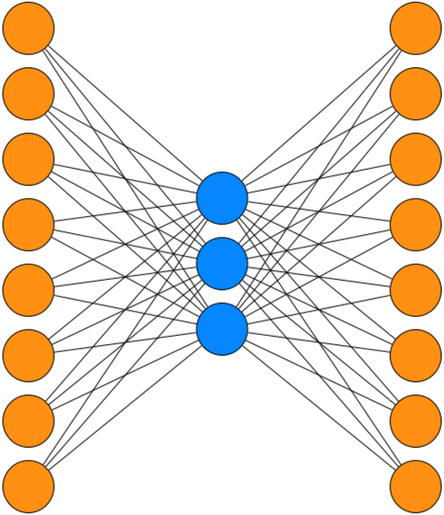
Autoencoders

- $\mathbf{X} \in [0,1]^d$ and *encodes* using hidden representation $y = s(WX + b)$, s is some non-linearity
- y or the code can then be decoded back $\mathbf{Z} = s(W^{'}y + b^{'})$
- If we constrain $W^{'}$ to be W^T, it is called *tied weights*

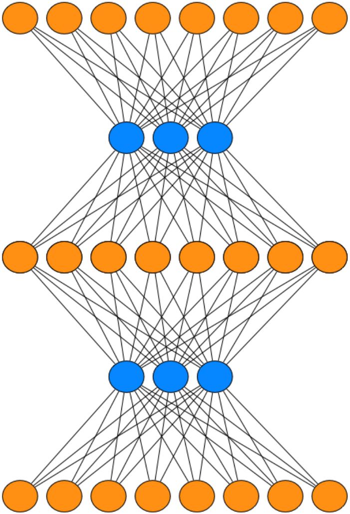
denoising Autoencoders

In order to force the autoencoder to become robust to noise and learn good representations of $\mathbf{X}$, train the autoencoder with *corrupted* versions of $\mathbf{X}$. denoising Autoencoder is a stochastic version of regular autoencoder.

denoising Autoencoders



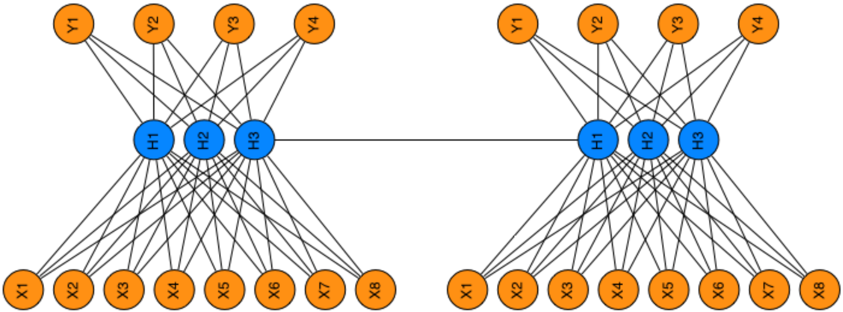
Stacked denoising Autoencoders



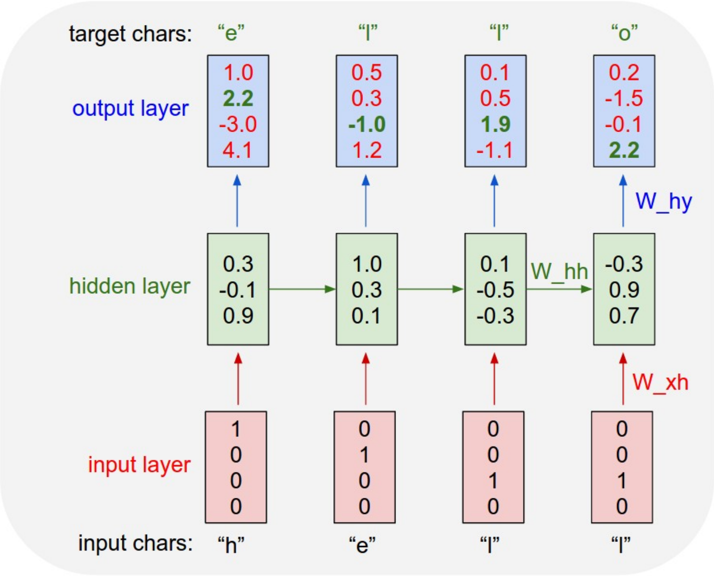
Recurrent Neural Networks

RNN

Recurrent because Hidden Layer is connected onto itself

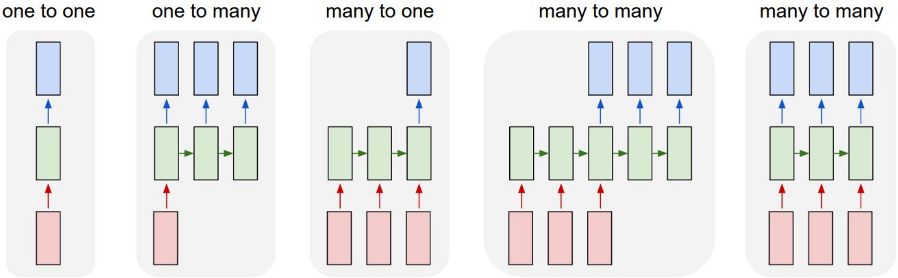


RNN



target chars:
"e"
"l"
"l"
"o"
output layer
1.0
2.2
-3.0
4.1
0.5
0.3
-1.0
1.2
0.1
0.5
1.9
-1.1
0.2
-1.5
-0.1
2.2
W_hy
hidden layer
0.3
-0.1
0.9
1.0
0.3
0.1
0.1
-0.5
-0.3
W_hh
-0.3
0.9
0.7
W_xh
input layer
1
0
0
0
0
1
0
0
0
0
1
0
0
0
1
0
input chars:
"h"
"e"
"l"
"l"

RNN



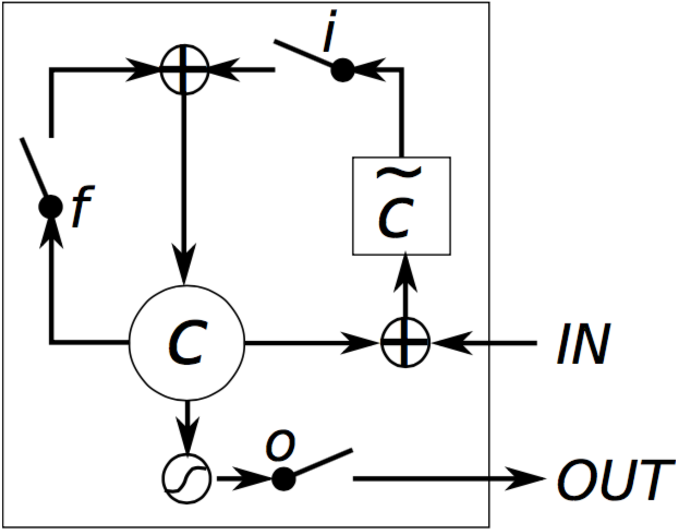
LSTM

Motivation for LSTM

In a regular RNN, the transition matrix weights (connecting hidden layer to itself) has a large impact on the learning rate – as that matrix is used in several gradient computations. Depending on the weight matrix, you either have **vanishing** gradients or **exploding** gradients. Vanishing gradients are the more common problem.

LSTM (Long Short Term Memory) cells try to mitigate this vanishing gradients with gating functions. Instead of $s_t = g(W_h s_{t-1} + W_i x_t)$, we have a more complex function connecting the output state at time t with current input and previous state.

LSTM



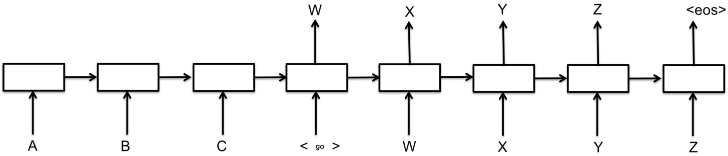
LSTM

- Input gate determines how much of an input does it allow to pass through. Input is $s_t = g(W_h s_{t-1} + W_i x_t)$.
- Forget gate is a new path that determines how much of the previous state is allowed to be carried through
- Output gate determines how much of the internal state does it expose to the external world

LSTM

If you set the input gates to all 1s, forget gates to all 0s, and output gates to all 1s, you get a regular RNN. RNN is a special case of LSTM

Sequence to Sequence Models



Sequence to Sequence Models

Just a RNN. Each rectangular block is a LSTM unit. You feed it sentences in one language and it produces sentences in another language.

Deep Learning

- Very active field
- New architectures / New applications emerging daily
- Very rapid progress in the last 2-3 years
- Some basic principles are understood. But is still a bit of the Wild West!

thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
bedankt
спасибо
Баярлалаа
рахмат
dank je
mochchakkeram
go raibh maith agat
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
мерси
감사합니다
ありがとう
ευχαριστώ
hvala
tapadh leat
sagolun
mersi
barka
welalin
tack
misaotra
matondo
paldies
grazzi
faafetai lava
kiitos
dankie
dhanyavad
vinaka
спасибі
blagodaram
enkosi
sobodi
dêkuji
mèsi
chnorakaloutioun
gratias ago
gràcies
sulpay
djiere dieuf
дякую
mamnun
хвала
asante
manana
obrigada
najis tuke
kam sah hamnida
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
trugarez
xiexie