Computer Vision, Convolutional Neural Networks, and Breast Cancer Classification

Pro Kumar

University of Delaware

May 21, 2020

Overview

1. Introduction (Very brief history, Current Applications)
2. What is a CNN? (Mathematical convolutions, benefits)
3. CNN Architecture Design Principles.
4. Classifying Breast Cancer (Problem & Dataset Description)
5. Implementing a few architectures (TensorFlow Tutorial, “ProNet" & AlexNet) and results.

Last 10 years or so

- 2009: Dr. Fei-Fei Li and her team lead an initiative called ImageNet.
  - Largest computer vision competition for object recognition.
- 2012: Krizhevsky et al, won the ImageNet Challenge that year with a CNN.
- Every year since, CNNs have won the ImageNet challenge.

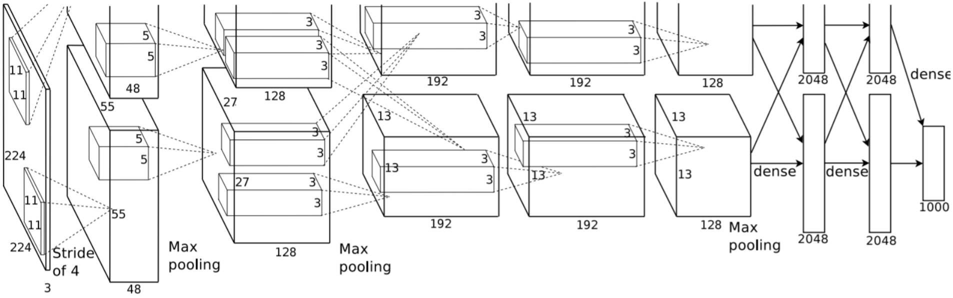
CNNs have been around for a while, LeCun (1998), so why the craze now?

- More efforts towards data curation, more computational power!

Based on Lecture 1 of Stanford's CS23n during Spring 2017.

AlexNet

CNN Architecture by Krizhevsky et al.



This will explained as we go through the presentation. Now, let's talk Mathematics!

Image credit: Krizhevsky et al. ImageNet Classification with Deep Convolutional Neural Networks. 2012.

Math: Def'n of Convolution

For convolutional neural networks, we need to talk about "convolutions".

Definition

Let $\mathbb{R}$ represent the space of all real numbers. Suppose we have two (Riemann) integrable functions $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$. The ***convolution*** *of f and g is a function*

$$(f * g)(x) := \int_{\mathbb{R}} f(t)g(x-t)dt. \tag{1}$$

How can we interpret this?

Math: Physical interpretation

Crudely, as amount of overlap between f and g as g is "swept" over f.

$$(f * g)(x) := \int_{\mathbb{R}} f(t)g(x-t)dt.$$

In application, f is a function that represents the input to our CNN and g will represent the **kernel** or **filter**.

In this form, we cannot implement the convolution on the computer. We must discretize!

Math: Discretization 1

Definition

Given two functions $f : \mathbb{Z} \to \mathbb{R}$ and $g : \mathbb{Z} \to \mathbb{R}$. The ***discrete convolution*** *of f and g is the function*

$$(f \bullet g)(x) := \sum_{n=-\infty}^{\infty} f(n)g(x-n) \tag{2}$$

We still have infinities!

- Assume f is zero everywhere except a finite number of (roughly localized) points.

Math: Tensors

Using more information about the input data type.

- Basic data structures for ML are tensors.
  - Scalars $\rightarrow$ 0-tensors.
  - Vectors (single row/column) $\rightarrow$ 1-tensors.
  - Matrices (2D arrays) $\rightarrow$ 2-tensors.

  This idea can be generalized to multi-dimensional arrays.

  We'll focus on 2- and 3-tensors for this project.

Math: Image Tensor Example

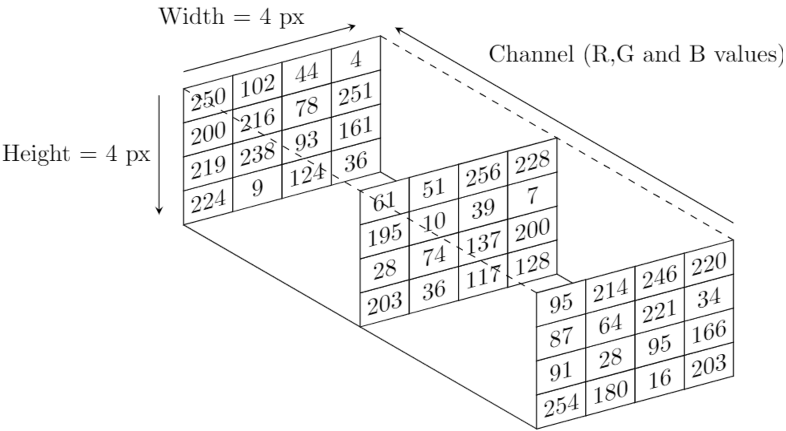


Image code based on response by Mark Wibrow at https://tex.stackexchange.com/questions/295006/drawing-multidimensional-array-using-tikz

Math: Discretization 2

Extend our definition of discrete convolution to multiple dimensions.

- Consider input to be a grayscale image, a 2-tensor, i.e. entries of representing matrix is a two-dimensional function of its input coordinates, $f(x, y)$.

Then, the convolution of f with the kernel g becomes

$$(f * g) = \sum_{n,m=-\infty}^{\infty} f(x, y) g(x - n, y - m). \tag{3}$$

Recall, how a CNN works...

Convolution on the Computer

Sweeping a kernel matrix pixel by pixel over the image and computing a sort of dot product in each place. Think 3D now!

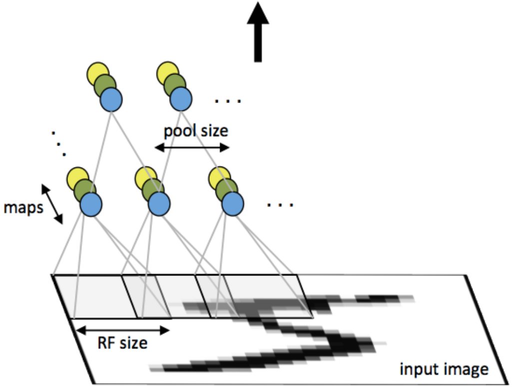


Image credit: A. Ng. UFLDL Tutorial. Stanford University. http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/

General CNN Architecture Principles

The convolution action described accounts for just the first layer of a CNN.

- Next, is the Pooling layer.
  - Usually Max/Mean Pool: Split matrix into uniformly sized submatrices, take max/mean of elements of each submatrix.
  - Kind of aggregate the findings of the previous convolution.
  - Reduce the spatial size of our image representation.
    - Reduces number of parameters to learn, and reduces overfitting.
- Fully connected (Dense) network.
  - Flatten representation.
  - Feed to dense NN to determine classification.

Benefits of CNNs

Three main attractions of CNNs, which are more or less tied to each other.

- Sparse connectivity. For example, instead of each input unit being connected to every hidden layer unit, they are just connected to a smaller convolutional kernel. (Usual kernel size is $(3, 3)$ or $(5, 5)$.)
- Parameter sharing. For example, instead of learning a weight for each connection location in a usual NN, a CNN only learns one set of weights and those weights are used everywhere.
- Equivariance. CNN's are equivariant to translation. Usually a kernel learns one feature across whole image.

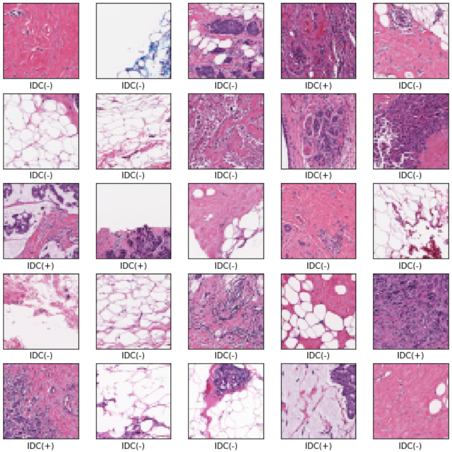
Breast Cancer Classification

- The type of breast cancer is invasive ductal carcinoma (IDC), which according to Johns Hopkins Medicine is "growing in a milk duct and [that] has invaded the fibrous or fatty tissue of the breast outside of the duct. IDC is the most common form of breast cancer, representing 80 percent of all breast cancer diagnoses."
- Downloaded dataset from Kaggle, Breast Histopathology Images (Paul Mooney).
- Originally from Janowczyk and Madabhushi's Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases (2016)
  - Survey on various digital pathology techniques.

The Dataset

- Original data set was 162 whole mount slide images of breast cancer specimens scanned at 40x.
- From this 277,524 patches of size 50 x 50 were extracted
  - 198,738 IDC negative, IDC(-) and 78,786 IDC positive, IDC(+).
- 3.3 GB of data!!
- Based on data set labels, this data set comes from 280 patients.
- From this, 8 patients' data was used. (3198 image patches, after equalizing number of IDC(+) and IDC(-) cases.)
- Train/test split: 75/25. Validation: 25 % of Train set.

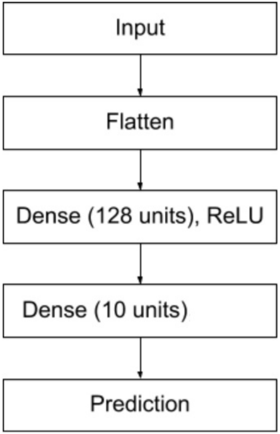
Data Example



Images downloaded from P. Mooney, Breast Histopathology Images. Kaggle Dataset. Originally from Janoczyk & Madabhushi, Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. 2016.

TensorFlow (TF) Tutorial: Network

Recall, the architecture of the TensorFlow Tutorial NN was the following:



TensorFlow (TF) Tutorial: Results

The original TF Tutorial NN was meant for grayscaled images.

- Using grayscaled images yields a test accuracy of 57%.
- Using RGB images yields test accuracy of 77%.

No other work done to improve the TF Tutorial models.

"ProNet", Network Design Approach

(Guided by Chollet, Goodfellow, CS231n (Stanford))

- Last layer activation: Sigmoid (vs ReLU?)
- Selected loss function: binary crossentropy (binary classification)
- Created a non-regularized, overfitting model.
  - Added layers, padding really helped out.
  - Lots of units
  - Trained for many epochs, (60-100).
- Regularized: Dropouts, L1/L2
- LEARNING RATE! Most time consuming & most important part of the process. Settled for "Adam" optimizer with lr = 1e-4.

Learning Rate Curve

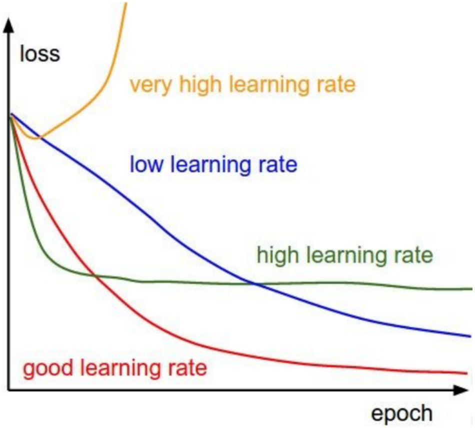
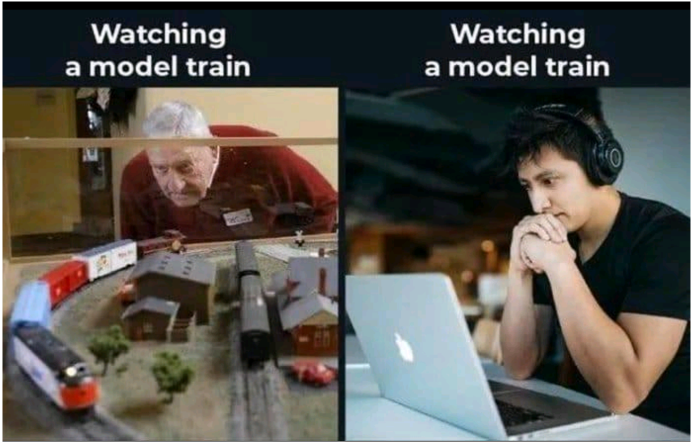
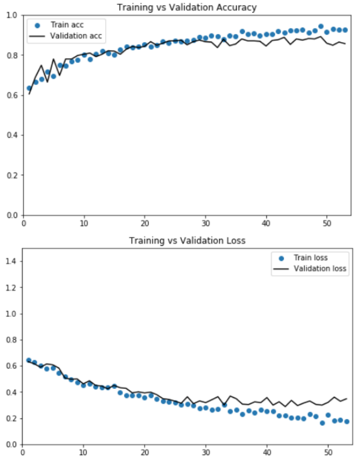


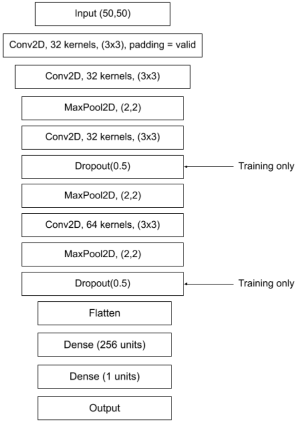
Image credit: CS231n, Lecture 6 (slides), Spring 2017. http://cs231n.stanford.edu/2017/syllabus.html



https://devhumor.com/media/watching-a-model-train
(Originally from Reddit (removed): https://www.reddit.com/r/ProgrammerHumor/comments/eftdzi/time_flies/)

"ProNet", Final Network & Training Curve





$\sim$ 89% Test Accuracy, F1-score = 0.92!

AlexNet (Janoczyck & Madabhushi)

padding = same

No Dropouts!

Sigmoid

Layer	Type	Num Kernels	Kernel size	Stride	Activation
0	Input	3	32×32	-	-
1	Convolution	32	5×5	1	-
2	Max pool	-	3×3	2	ReLu
3	Convolution	32	5×5	1	ReLu
4	Mean pool	-	3×3	2	
5	Convolution	64	5×5	1	ReLu
6	Mean pool	-	3×3	2	
7	Fully connected	64	-	-	Dropout+ ReLu
8	Fully connected	2	-	-	Dropout+ ReLu
9	SoftMax	-	-	-	

- Yields 85-86% test accuracy.
- F1-score = 0.86 (compared with 0.7648 in the paper, but they trained on about 324,000 patches (w/ data augmentation))
- Note: Input size = (32,32) and Only trained for 35 epochs.

Table credit: Janoczyk & Madabhushi, Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. 2016.

Conclusion

- ProNet performed the best, followed by Janoczyck and Madabhushi's AlexNet, then the TensorFlow Tutorial Network.
  - Probably needs a bit more a statisitical analysis (e.g. error bars)
- No generalization to entire data set by any means.
- Making these from scratch is an extreme pain!
  - Too many things to adjust!
  - Adjust one hyperparameter, another goes out of whack, etc.
  - If possible for the problem at hand, use a pre-trained model (available through Keras API.)
  - Don't try re-training any of the ImageNet winning architectures! (e.g. $\sim$65 million trainable parameters for 2012 AlexNet!)

Thank you and Stay safe!

*Image credit: https://twitter.com/deeplearningai_*

References

R. Bracewell. The Fourier Transform and its Applications. Third Edition. International Editions. McGraw Hill. 2000.

F. Chollet. Deep Learning with Python. Manning Publications Co. 2018.

DeepLearning.ai. Twitter post. May 4, 2020. https://twitter.com/deeplearningai_

I. Goodfellow, Y. Bengio, A. Courville. Deep Learning. The MIT Press. 2016. http://www.deeplearningbook.org

DevHumor (post). Watching a model train. 2019. https://devhumor.com/media/watching-a-model-train

A. Janowczyk, A. Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. J Pathol Inform. 2016; 7: 29. 2016 Jul 26. doi: 10.4103/2153-3539.186902.

A. Krizhevsky et al. ImageNet Classification with Deep Convolutional Neural Networks. 2012. http://www.cs.toronto.edu/ hinton/absps/imagenet.pdf

Y. LeCun. Gradient-based learning applied to document recognition. 1998. http://www.dengfanxin.cn/wp-content/uploads/2016/03/1998Lecun.pdf

F. Li, J. Johnson, S. Yeung. CS231n: Convolutional Neural Networks for Visual Recognition. Spring 2017. http://cs231n.stanford.edu/2017/

P. Mooney. Breast Histopathology Images. Kaggle Dataset. 2018.

A. Ng. UFLDL Tutorial. Stanford Univ. deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/

M. Wibrow. Drawing Multidimension Array using TikZ. StackExchange (forum). https://tex.stackexchange.com/questions/295006/drawing-multidimensional-array-using-tikz