

Ch 10.3: Convolutional Neural Nets

Lecture 32 - CMSE 381

Prof. Elizabeth Munch

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Weds, Dec 6, 2022

Last time:

- Multilayer
- pyTorch

This lecture:

- CNNs

Announcements:

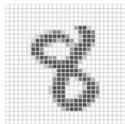
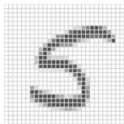
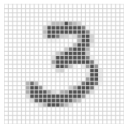
- Project due Friday

Section 1

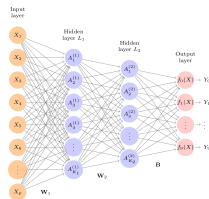
Last time: Neural Nets

MNIST

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9



Neural network architecture for MNIST



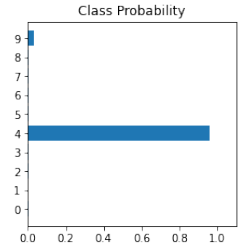
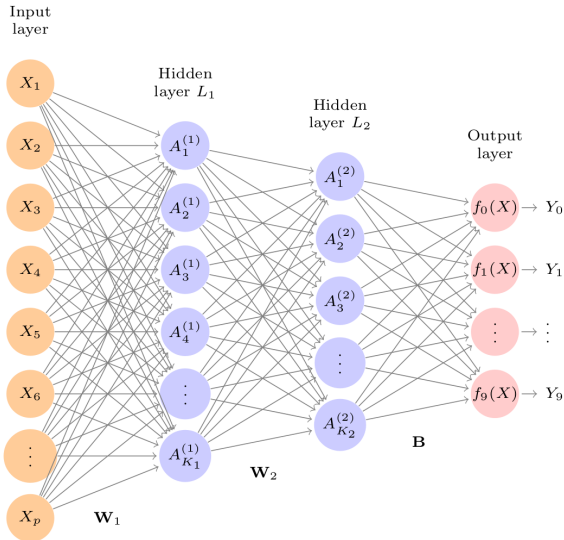
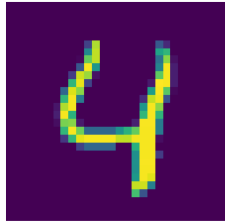
$$\begin{aligned} A_k^{(1)} &= h_k^{(1)}(X) \\ &= g(w_{k0}^{(1)} + \sum_{j=1}^p w_{kj}^{(1)} X_j) \end{aligned}$$

$$\begin{aligned} A_\ell^{(2)} &= h_\ell^{(2)}(X) \\ &= g(w_{\ell 0}^{(2)} + \sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)}) \end{aligned}$$

$$\begin{aligned} Z_m &= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X) \\ &= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)}, \end{aligned}$$

$$f_m(X) = \Pr(Y = m|X) = \frac{e^{Z_m}}{\sum_{\ell=0}^9 e^{Z_\ell}},$$

MNIST learning



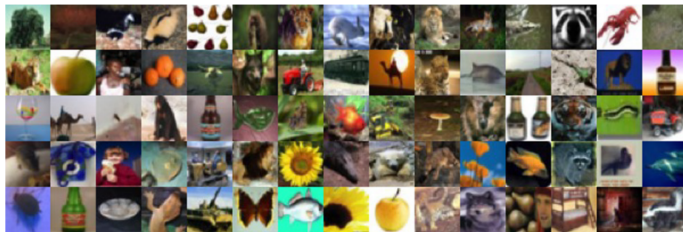
Section 2

Convolutional Neural Network

Last time: Flattening the image

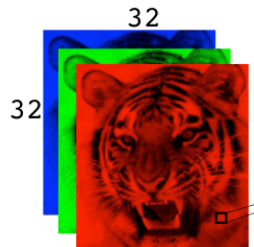
$$\begin{pmatrix} 1 & 1 & 0 \\ 4 & 2 & 1 \\ 0 & 2 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 4 \\ 2 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

Example data set: CIFAR100 Data

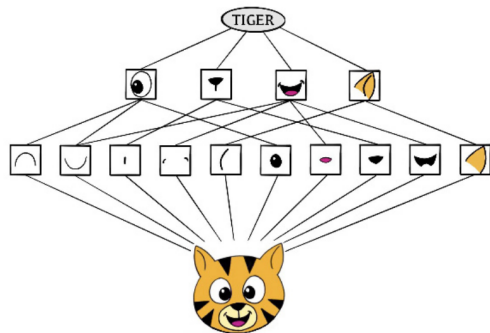


- 60,000 images: 50K training, 10K test
- Labels with 20 super classes (e.g. aquatic mammals)
- 5 classes per super class (beaver, dolphin, otter, seal, whale)
- Images are 32x32

Image channel data



CNNs



Convolution layer

Convolution Filter

Original Image:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

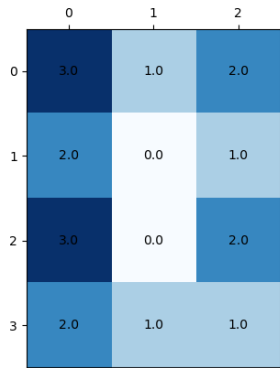
Convolution filter:

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

Convolved Image

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

Convolution Filter Example

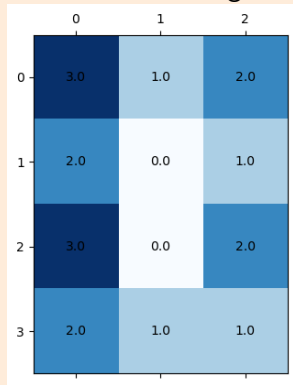


Filter:

$$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Same example, different filter

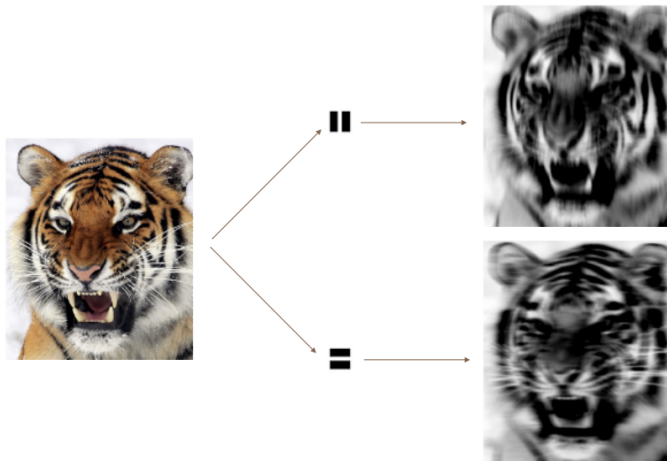
What is the convolved image?



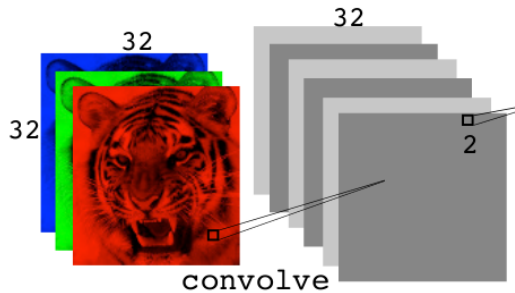
Filter:

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

Convolution filter: Bigger example



Convolution layer

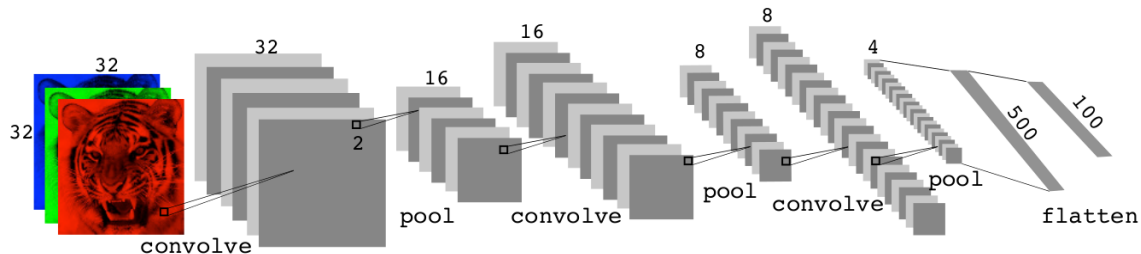


More notes on convolution

Pooling layers

$$\text{Max pool} \begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 5 \\ 2 & 4 \end{bmatrix}$$

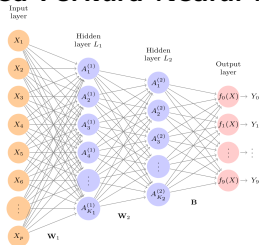
Putting it together to make a CNN



<https://poloclub.github.io/cnn-explainer/>

Coding

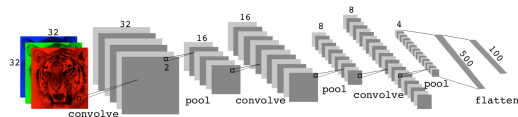
Feed Forward Neural Net



$$A_k = h_k(X) = g(w_{k0} + \sum_{j=1}^p w_{kj} X_j),$$

- Combines input data using learned weights
- Linear combo of those to get output
- Sometimes softmax to get probability of classification

CNN



- Specialized NN
- Gets next layer via
 - ▶ Convolution layer
 - ▶ Pooling Layer
 - ▶ Fully connected layer

Next time

Lec #	Date			Reading	Homeworks
	Wed	Nov 29	Midterm #3		
32	Fri	Dec 1	Multi Layer NN	10.2	
33	Mon	Dec 4	CNN	10.3	
34	Wed	Dec 6	Unsupervised Learning & Clustering	12.1, 12.4	
35	Fri	Dec 8	Virtual: Project office hours		Project due