Ch 10.3: Convolutional Neural Nets

Lecture 31 - CMSE 381

Prof. Elizabeth Munch

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Fri, Nov 22, 2024

Announcements

Last time:

- Multilayer NN
- pyTorch

This lecture:

- CNNs

Announcements:

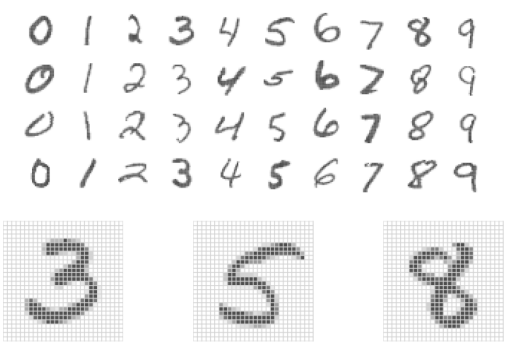
- Project due Friday
- HW #9 is posted
- Exam 3 is Dec 4th
- Project is due Dec 6th

Lec #	Date			Reading	HW
21	Mon	10/28	Polynomial & Step Functions	7.1,7.2	
22	Wed	10/30	Step Functions; Basis functions; Start Splines	7.2 - 7.4	
23	Fri	11/1	Regression Splines	7.4	HW #6 Due Sun 11/3
24	Mon	11/4	Decision Trees	8.1	
25	Wed	11/6	Class Cancelled (Dr Munch out of town)		
26	Fri	11/8	Random Forests	8.2.1, 8.2.2	HW #7 Due Sun 11/10
27	Mon	11/11	Maximal Margin Classifier	9.1	
28	Wed	11/13	SVC	9.2	
29	Fri	11/15	SVM	9.3, 9.4	HW #8 Due Sun 11/17
30	Mon	11/18	Single layer NN	10.1	
31	Wed	11/20	Multi Layer NN	10.2	
32	Fri	11/22	CNN	10.3	HW #11 Due Sun 11/24
33	Mon	11/25	TBD: Unsupervised learning/clustering	12.1, 12.4?	
	Wed	11/27	Virtual: Project office hours		
	Fri	11/29	No class - Thanksgiving		
	Mon	12/2	Review		
	Wed	12/4	**Midterm #3**		
	Fri	12/6	No class - EGR Design Day		Project due

Section 1

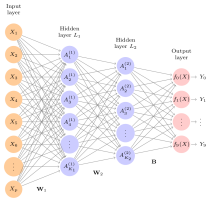
Last time: Neural Nets

MNIST



- Goal: Build a model to classify images into their correct digit class
- Each image has $p = 28 \cdot 28 = 784$ pixels
- Each pixel is grayscale value in [0,255]
- Data converted into column order
- Output represented by one-hot vector $Y = (Y_0, Y_1, \cdots, Y_9)$
- 60K training images, 10K test images

Neural network architecture for MNIST



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

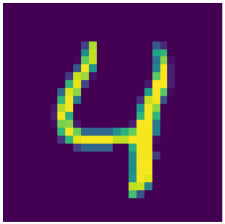
$$
\begin{aligned}
A_\ell^{(2)} &= h_\ell^{(2)}(X) \\
&= g(w_{\ell 0}^{(2)} + \textstyle\sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)})
\end{aligned}
$$

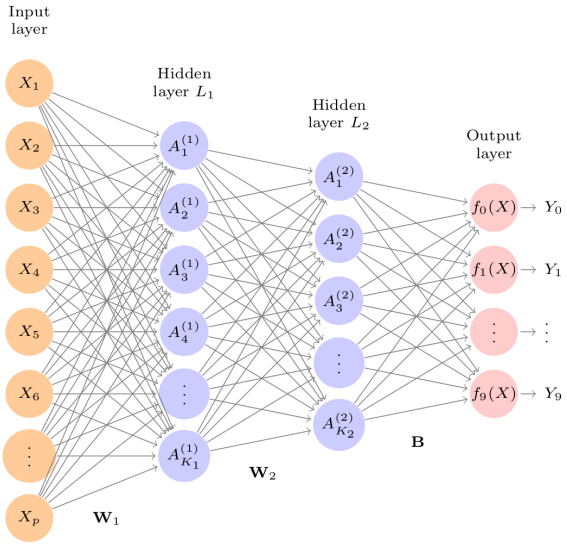
$$
\begin{aligned}
Z_m &= \beta_{m0} + \textstyle\sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X) \\
&= \beta_{m0} + \textstyle\sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)},
\end{aligned}
$$

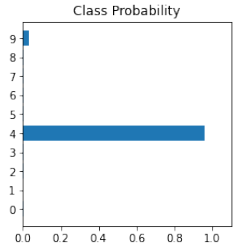
$$
f_m(X) = \Pr(Y = m|X) = \frac{e^{Z_m}}{\sum_{\ell=0}^{9} e^{Z_\ell}},
$$

- Two hidden layers.
- Softmax for classification output
- We used L_1 has 128 units; L_2 has 64
- 10 output variables due to class labeling
- Result is we are training approx 110K weights

MNIST learning







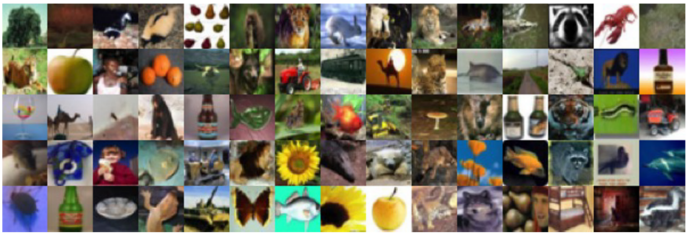
Section 2

Convolutional Neural Network

Last time: Flattening the image

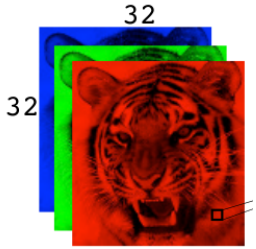
$$\begin{pmatrix} 1 & 1 & 0 \\ 4 & 2 & 1 \\ 0 & 2 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 4 \\ 2 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

Example data set: CIFAR100 Data

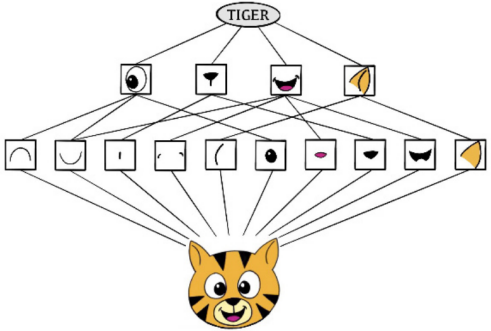


- 60,000 images: 50K training, 10K test
- Labels with 20 super classes (e.g. aquatic mammals)
- 5 classes per super class (beaver, dolphin, otter, seal, whale)
- Images are 32x32

Image channel data



CNNs



Convolution layer

Convolution Filter

Original Image:

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

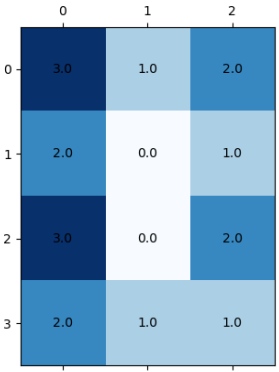
Convolution filter:

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

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:

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

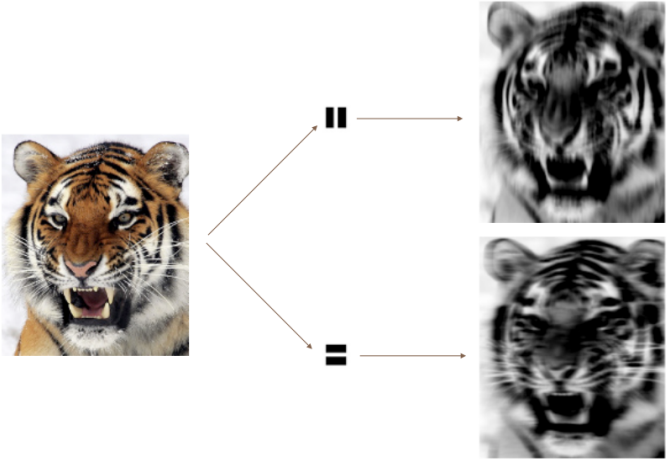
Same example, different filter

What is the convolved image?

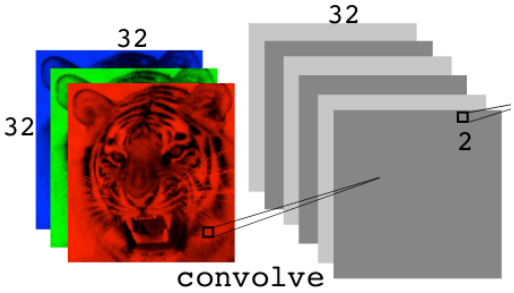
Filter:

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

Convolution filter: Bigger example



Convolution layer

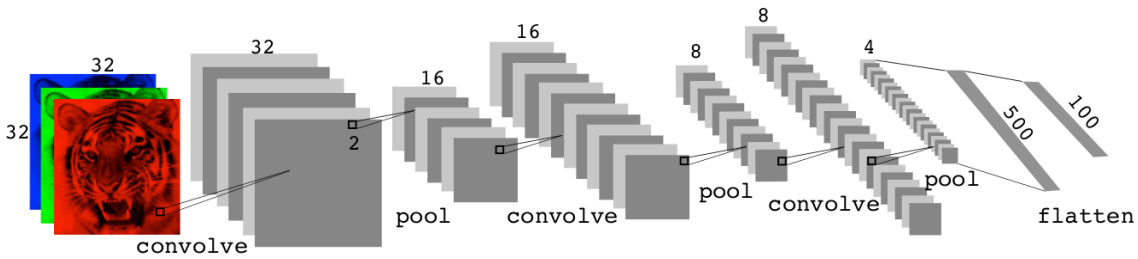


More notes on convolution

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

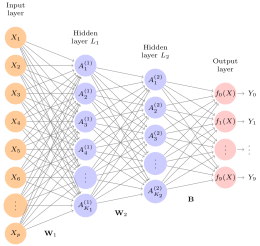


https://poloclub.github.io/cnn-explainer/

Coding

TL;DR

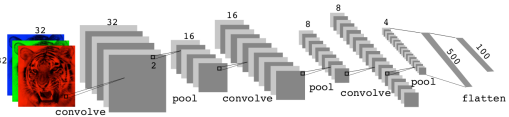
Feed Forward Neural Net



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

- 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	HW
21	Mon	10/28	Polynomial & Step Functions	7.1,7.2	
22	Wed	10/30	Step Functions; Basis functions; Start Splines	7.2 - 7.4	
23	Fri	11/1	Regression Splines	7.4	HW #6 Due Sun 11/3
24	Mon	11/4	Decision Trees	8.1	
25	Wed	11/6	Class Cancelled (Dr Munch out of town)		
26	Fri	11/8	Random Forests	8.2.1, 8.2.2	HW #7 Due Sun 11/10
27	Mon	11/11	Maximal Margin Classifier	9.1	
28	Wed	11/13	SVC	9.2	
29	Fri	11/15	SVM	9.3, 9.4	HW #8 Due Sun 11/17
30	Mon	11/18	Single layer NN	10.1	
31	Wed	11/20	Multi Layer NN	10.2	
32	Fri	11/22	CNN	10.3	HW #11 Due Sun 11/24
33	Mon	11/25	TBD: Unsupervised learning/clustering	12.1, 12.4?	
	Wed	11/27	Virtual: Project office hours		
	Fri	11/29	No class - Thanksgiving		
	Mon	12/2	Review		
	Wed	12/4	**Midterm #3**		
	Fri	12/6	No class - EGR Design Day		Project due