

Ch 10.3: Convolutional Neural Nets

Lecture 31 - CMSE 381

Prof. Mengsen Zhang

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Mon, 11/17, 2025

Announcements

Last time:

- Multilayer NN
- pyTorch

This lecture:

- CNNs

Final countdown:

- HW #9 is due Sunday 11/23
- Exam 3 is 11/24
- Project is due 12/5

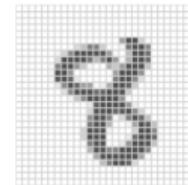
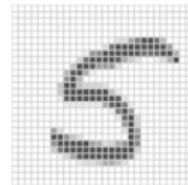
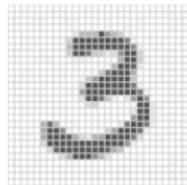
21	F	10/24	Polynomial & Step Functions	7.1-7.2	HW #5 Due Sun 10/26
22	M	10/27	Step Functions; Basis functions; Start Splines	7.2-7.4	
23	W	10/29	Regression Splines	7.4	
24	F	10/31	Decision Trees	8.1	HW #6 Due Sun 11/2
25	M	11/3	Random Forests	8.2.1, 8.2.2	
26	W	11/5	Maximal Margin Classifier	9.1	
27	F	11/7	SVC	9.2	HW #7 Due Sun 11/9
28	M	11/10	SVM	9.3, 9.4	
29	W	11/12	Single Layer NN	10.1	
30	F	11/13	Multi Layer NN	10.2	HW #8 Due Sun 11/16
31	M	11/17	CNN	10.3	
32	W	11/19	Unsupervised learning / clustering	12.1, 12.4	
33	F	11/21	Review		HW #9 Due Sun 11/23
	M	11/24	Midterm #3		
	W	11/26	Virtual: Project Office Hours		
	F	11/28	Thanksgiving		
	M	12/1	Virtual: Project Office Hours		
	W	12/3	Virtual: Project Office Hours		
	F	12/5			Project Due
	M	12/8			
	W	12/10			
	F	12/12	No final exam		Honors Project Due

Section 1

Last time: Neural Nets

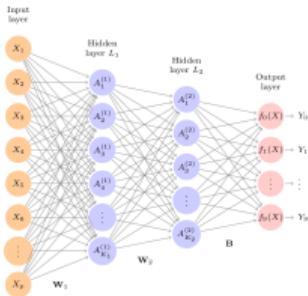
MNIST

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



- 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 $Y = (Y_0, Y_1, \dots, Y_9)$
- 60K training images, 10K test images

Neural network architecture for MNIST



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

$$\begin{aligned} A_\ell^{(2)} &= h_\ell^{(2)}(X) \\ &= g(w_{\ell0}^{(2)} + \sum_{k=1}^{K_1} w_{\ell k}^{(2)} A_k^{(1)}) \end{aligned}$$

$$Z_m = \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} h_\ell^{(2)}(X)$$

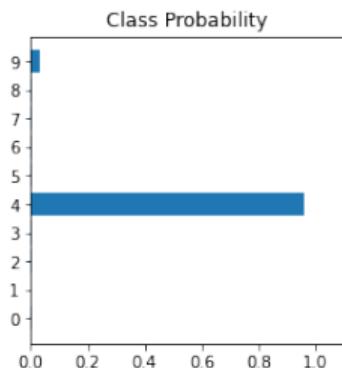
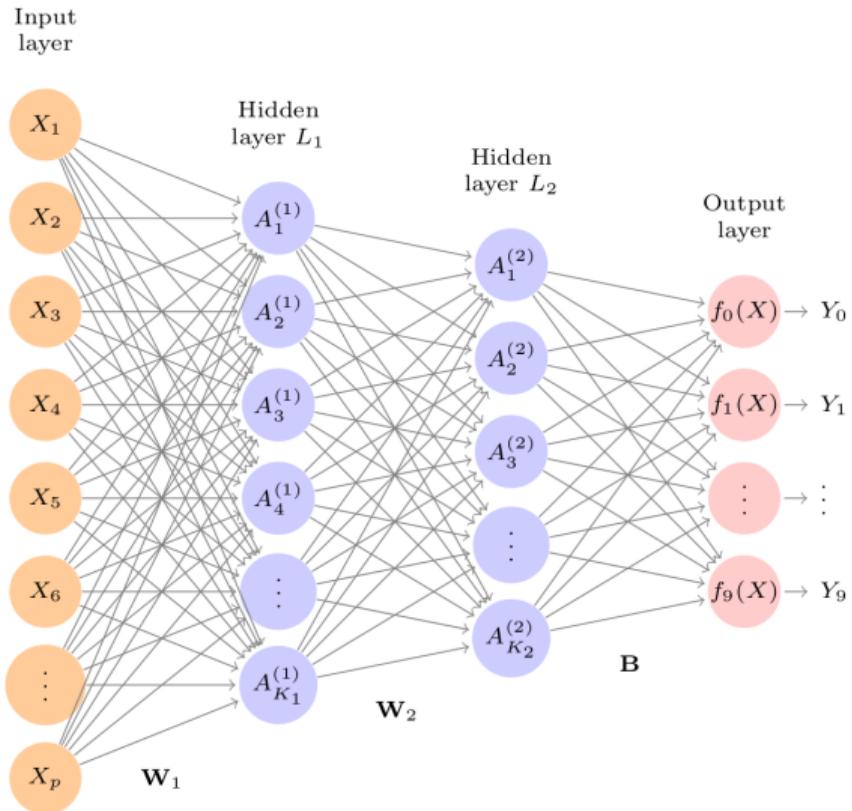
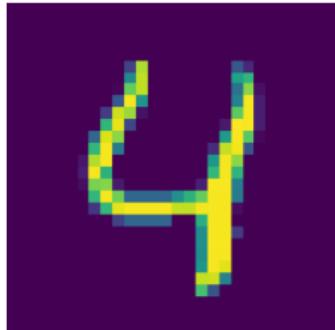
$$= \beta_{m0} + \sum_{\ell=1}^{K_2} \beta_{m\ell} A_\ell^{(2)},$$

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

- 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

Test your understanding: [PollEv](#)

MNIST learning



What will you learn today?

- What is the architecture of a convolutional neural network?
 - ▶ You should be able to describe the types of layers, how they are arranged, and the purpose of each of them.
- How does convolution work?
 - ▶ You should be able to compute the result of the convolution between a simple matrix and a simple filter by hand.
- What information does Convolutional Neural Network capture through convolution?

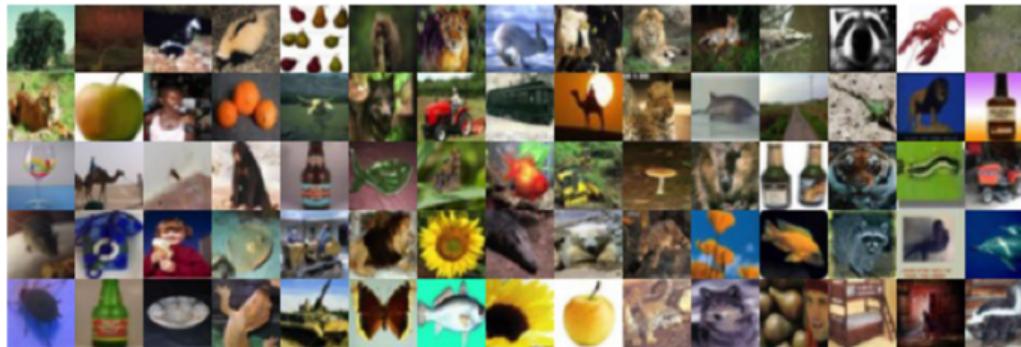
Section 2

Convolutional Neural Network

Last time: Flattening the image – why is it not great?

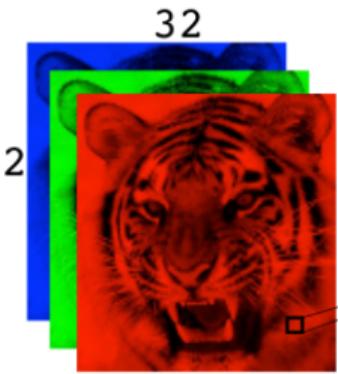
$$\begin{pmatrix} 1 & 1 & 0 \\ 4 & 2 & 1 \\ 0 & 2 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 4 \\ 2 \\ 1 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

Example data set: CIFAR100 Data

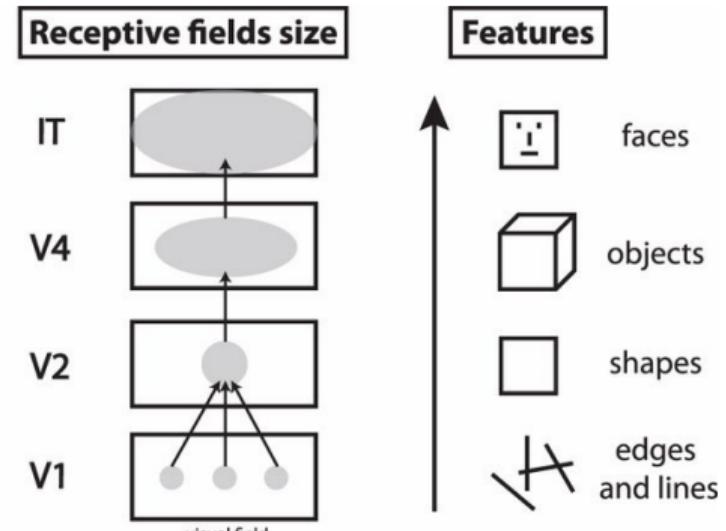
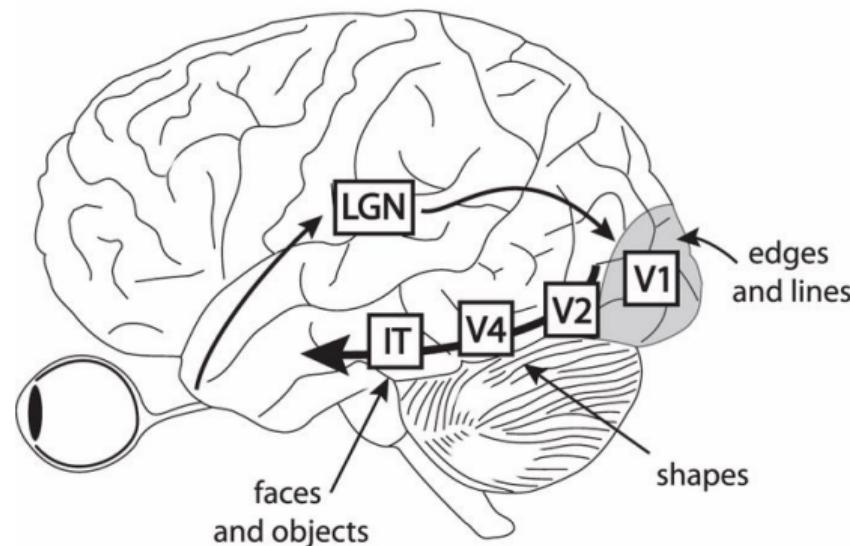


- 60,000 images: 50K training, 10K test
- Labels with 20 super classes (e.g. aquatic mammals)
- 5 classes per super class (beaver, dolphin, otter, seal, whale)
- Images are 32x32

Image channel data

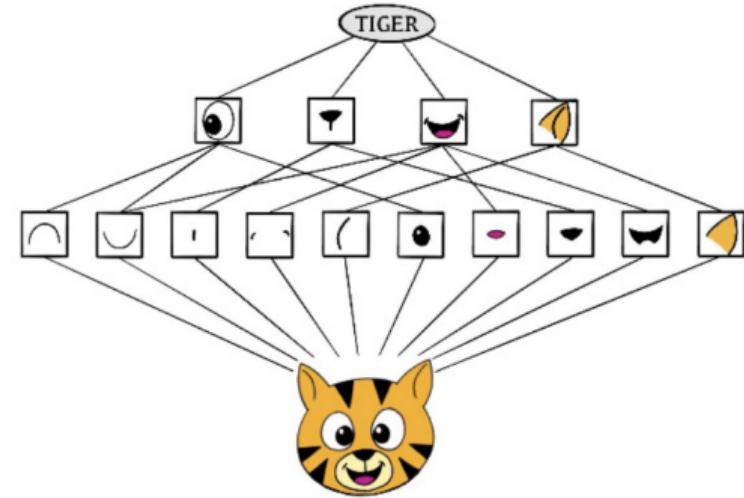


How does your brain do it? The visual hierarchy



Mauro Manassi, Bilge Sayim, Michael H. Herzog; When crowding of crowding leads to uncrowding. Journal of Vision 2013;13(13):10. <https://doi.org/10.1167/13.13.10>.

CNNs



Convolution layer

Convolution Filter

Original Image:

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

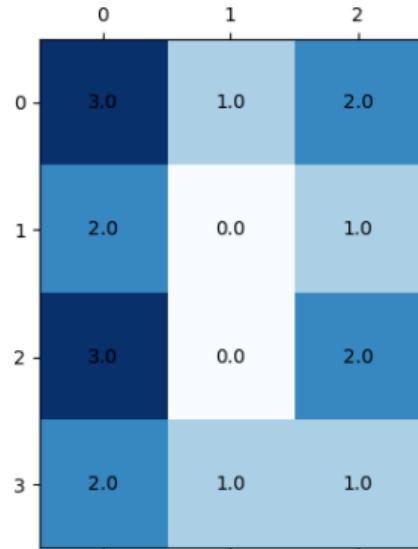
Convolution filter:

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

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

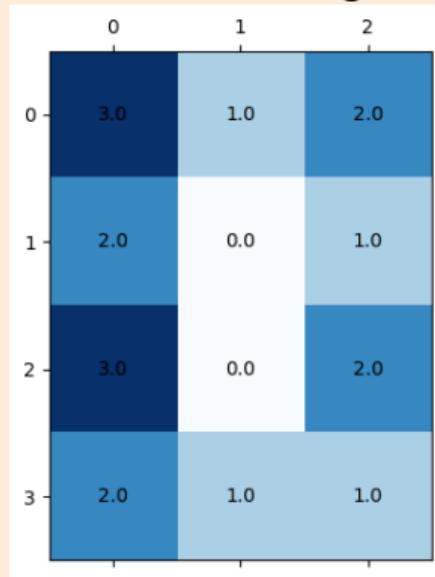


Filter:

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

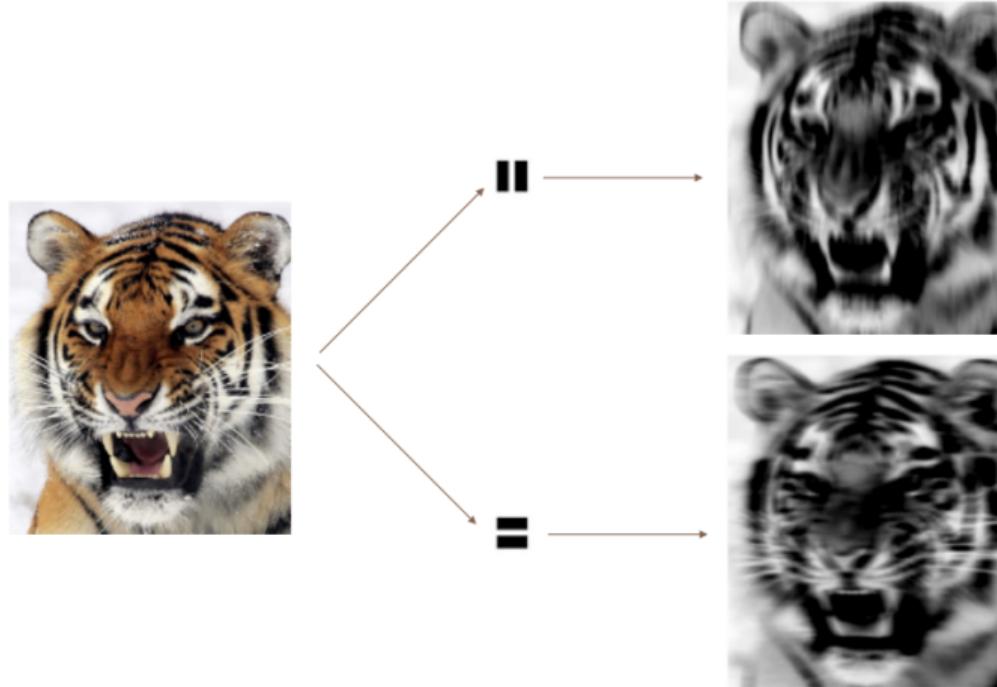
Same example, different filter

What is the convolved image?

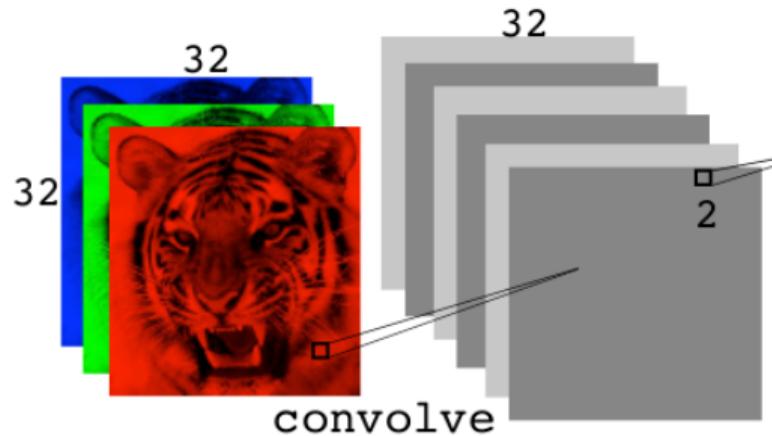


$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$$

Convolution filter: Bigger example



Convolution layer



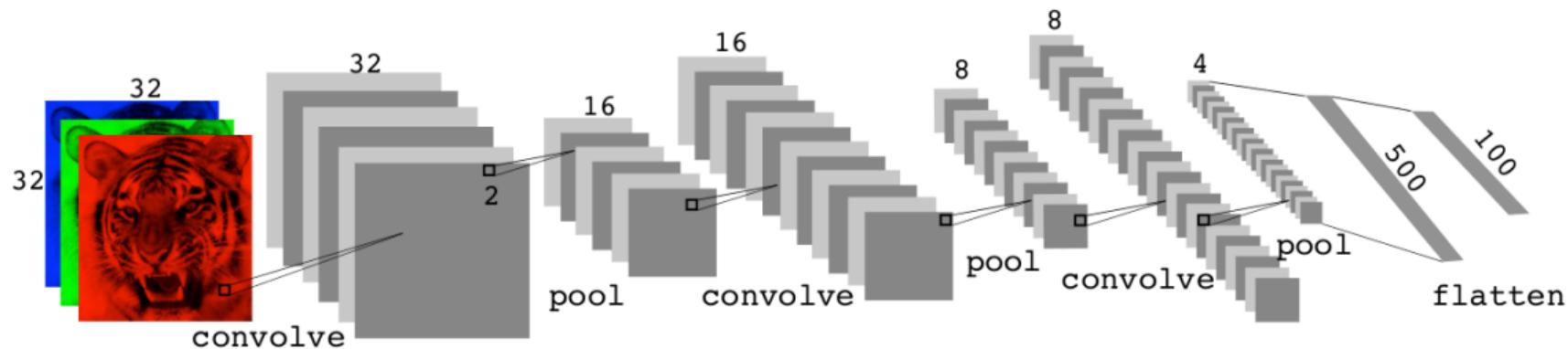
More notes on convolution

Pooling layers

Max pool

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

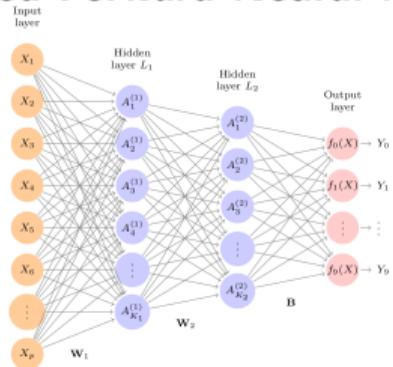
Putting it together to make a CNN



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

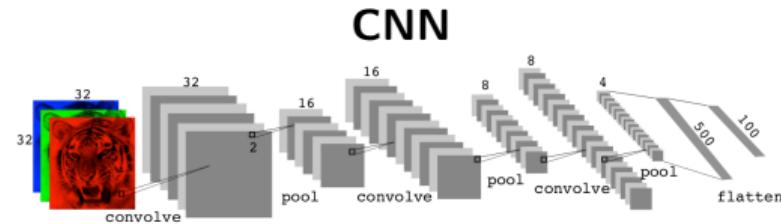
Coding

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



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

Next time

21	F	10/24	Polynomial & Step Functions	7.1-7.2	
22	M	10/27	Step Functions; Basis functions; Start Splines	7.2-7.4	HW #5 Due Sun 10/26
23	W	10/29	Regression Splines	7.4	
24	F	10/31	Decision Trees	8.1	HW #6 Due Sun 11/2
25	M	11/3	Random Forests	8.2.1, 8.2.2	
26	W	11/5	Maximal Margin Classifier	9.1	
27	F	11/7	SVC	9.2	HW #7 Due Sun 11/9
28	M	11/10	SVM	9.3, 9.4	
29	W	11/12	Single Layer NN	10.1	
30	F	11/13	Multi Layer NN	10.2	HW #8 Due Sun 11/16
31	M	11/17	CNN	10.3	
32	W	11/19	Unsupervised learning / clustering	12.1, 12.4	
33	F	11/21	Review		HW #9 Due Sun 11/23
	M	11/24	Midterm #3		
	W	11/26	Virtual: Project Office Hours		
	F	11/28	Thanksgiving		
	M	12/1	Virtual: Project Office Hours		
	W	12/3	Virtual: Project Office Hours		
	F	12/5			Project Due
	M	12/8			
	W	12/10			
	F	12/12	No final exam		Honors Project Due