Practical Deep Learning

Dan Becker

kaggle

#ODSC

Welcome

Do two things:

1. Download slides from https://github.com/dansbecker/odsc_2018

2. Ensure you have a **verified** Kaggle account
    - Verify by visiting kaggle.com/kernels, selecting “New Kernel” and then selecting Notebook.

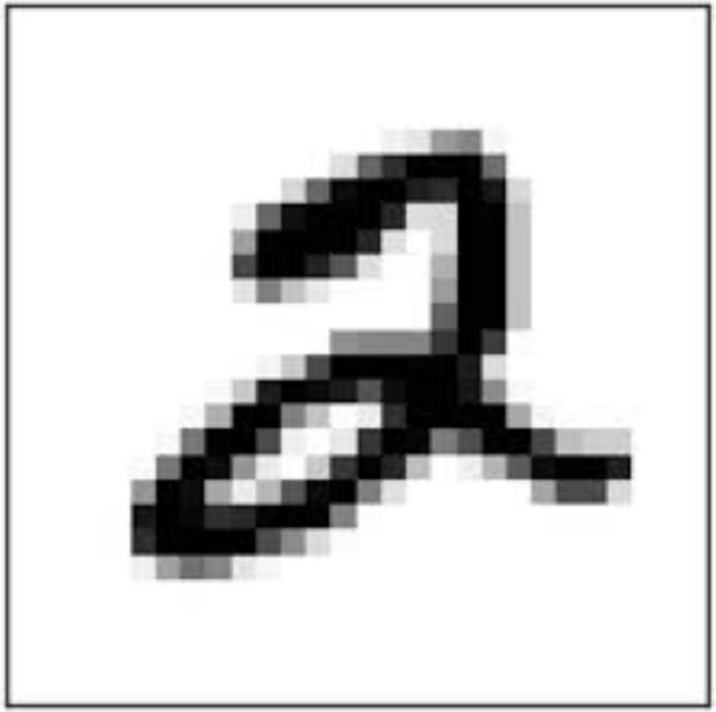
WORKSHOP PLAN

This Workshop

- Learning Approach
- Ideal Background
- Use Cases
- Tools Covered

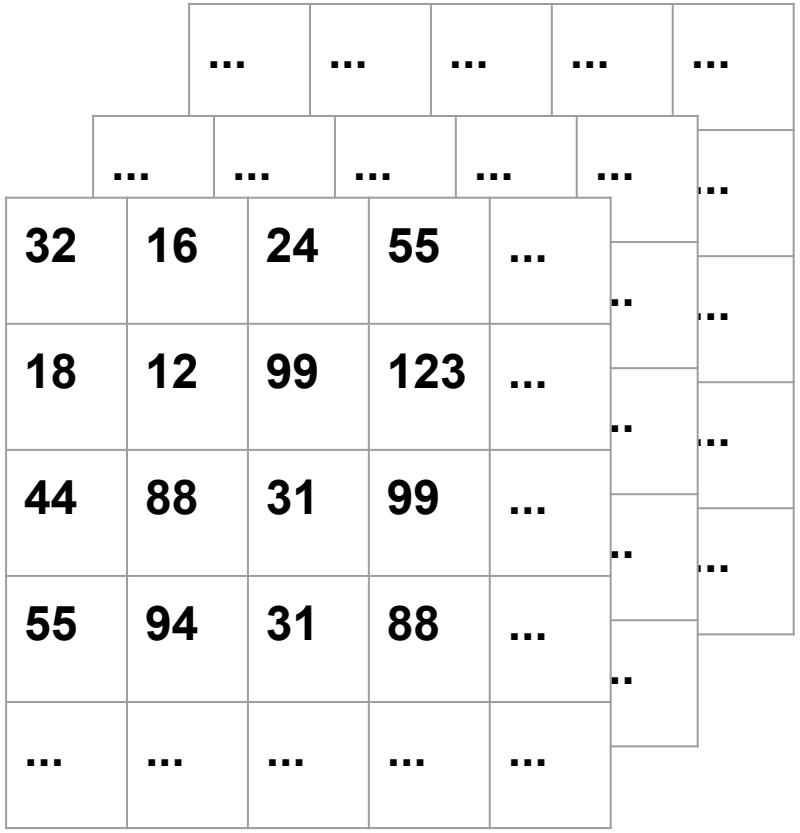
IMAGE PROCESSING BASICS

Image As Matrix

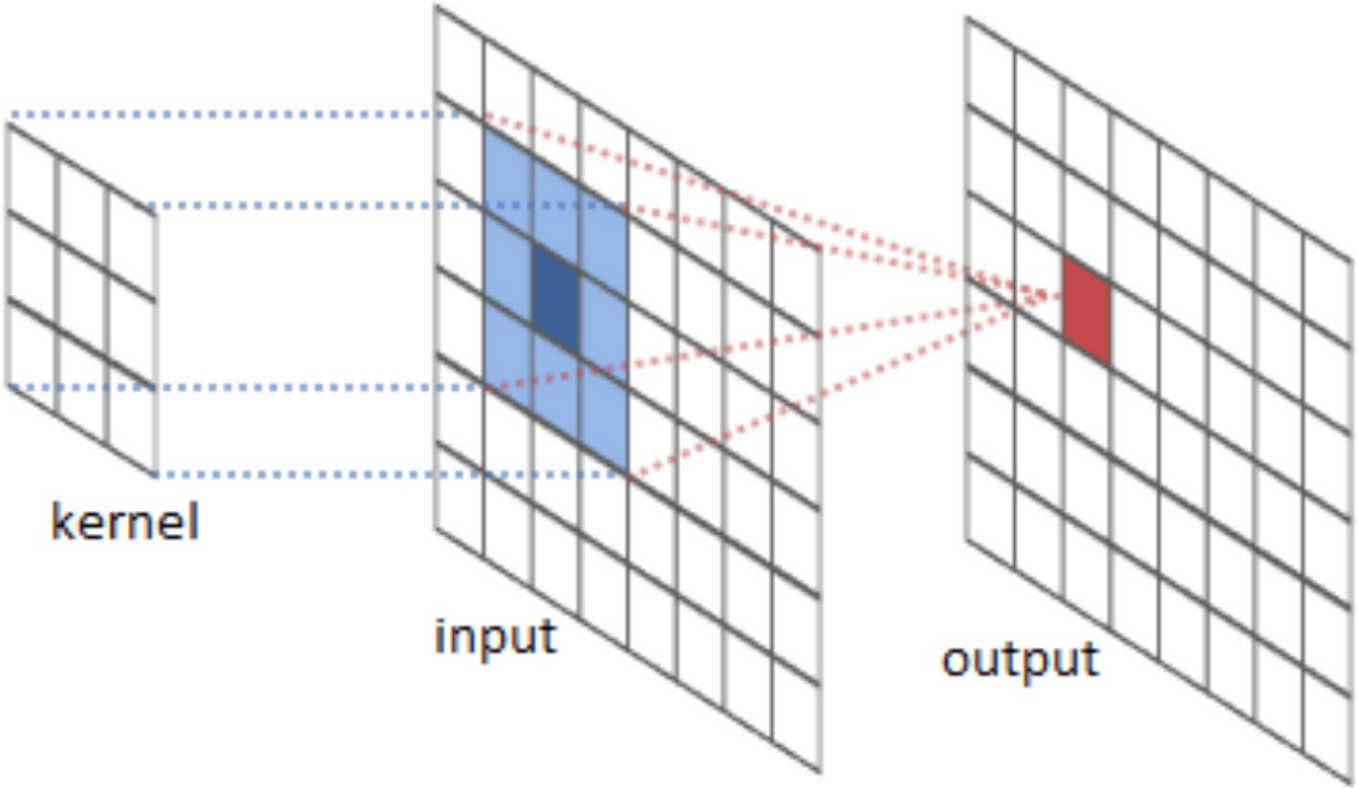


32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

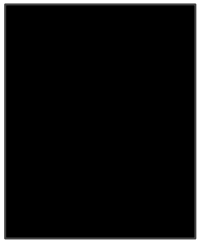
Color Images



Convolutions: The Building Block of Computer Vision



Applying a Convolution



Data

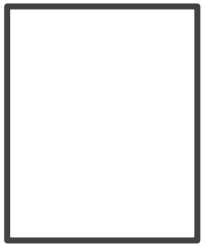
200	**200**	**...**	**...**	**...**
200	**200**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**

Convolution

1.5	1.5
-1.5	-1.5

= 200(1.5) + 200(1.5)
- 200(1.5) - 200(1.5)
= 0

Applying a Convolution: Example 2



Data

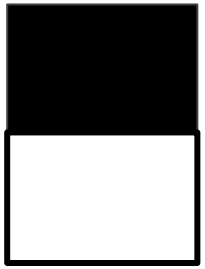
0	**0**	**…**	**…**	**…**
0	**0**	**…**	**…**	**…**
…	**…**	**…**	**…**	**…**
…	**…**	**…**	**…**	**…**
…	**…**	**…**	**…**	**…**

Convolution

1.5	1.5
-1.5	-1.5

= 4(0)(1.5)
= 0

Applying a Convolution: Example 3



Data

200	**200**	**...**	**...**	**...**
0	**0**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**
...	**...**	**...**	**...**	**...**

Convolution

1.5	1.5
-1.5	-1.5

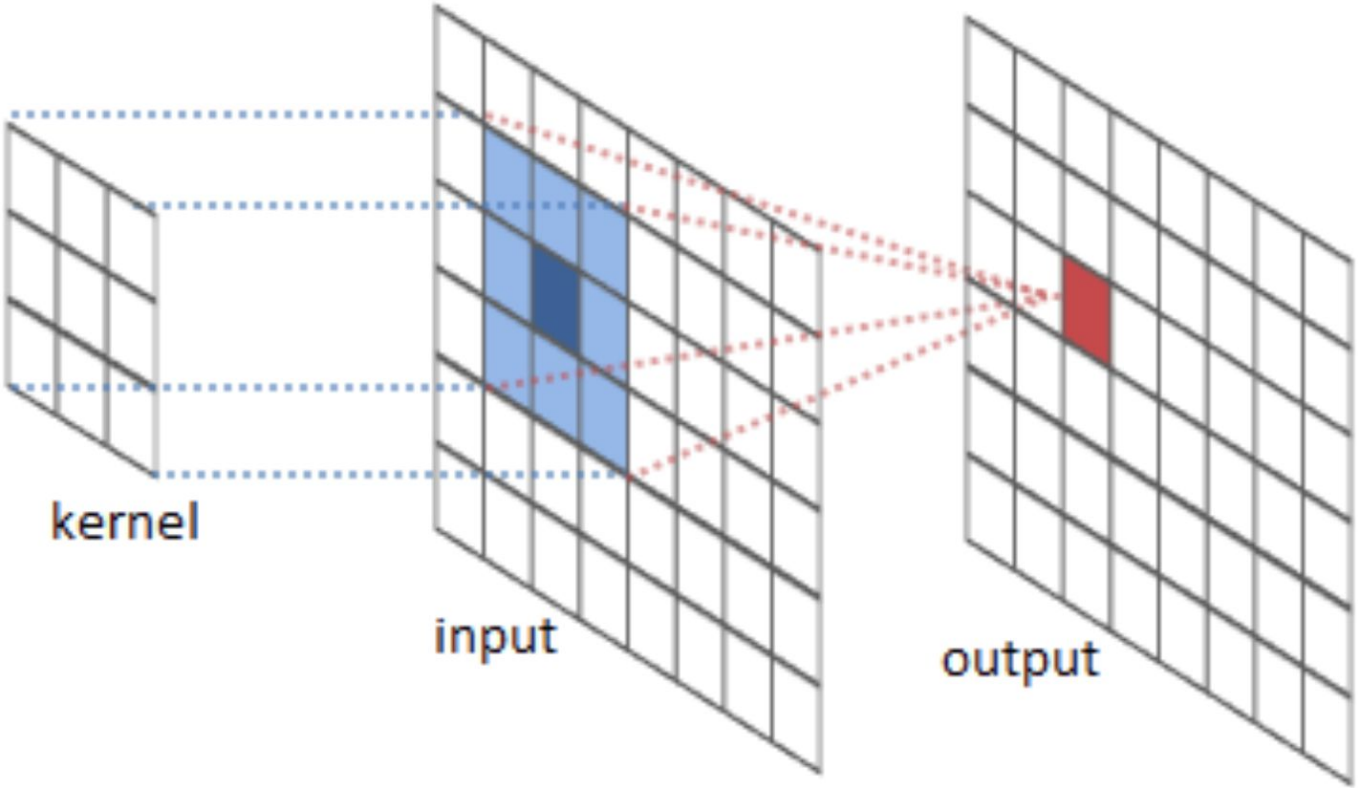
= 2 * 1.5 * 200
= 600

First Exercise

- https://www.kaggle.com/dansbecker/exercise-convolutions-for-computer-vision
- Need a verified kaggle account.

FROM CONVOLUTIONS TO MODELS

Convolutions: The Building Block of Computer Vision



Multiple Convolutions

Image

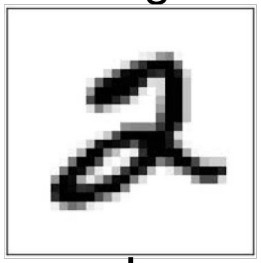


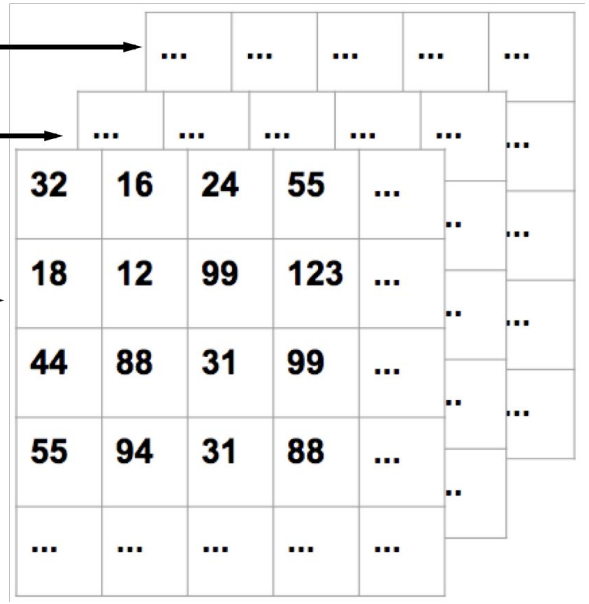
Image Data

0	0	200	150	0	0
0	143	55	99	222	0
0	188	0	0	181	0
0	0	0	200	0	0
0	0	149	0	0	0
0	245	202	140	225	0
0	0	0	0	0	0

Filter 3

Filter 2

Filter 1



#ODSC

HOW A DEEP NEURAL NETWORK SEES

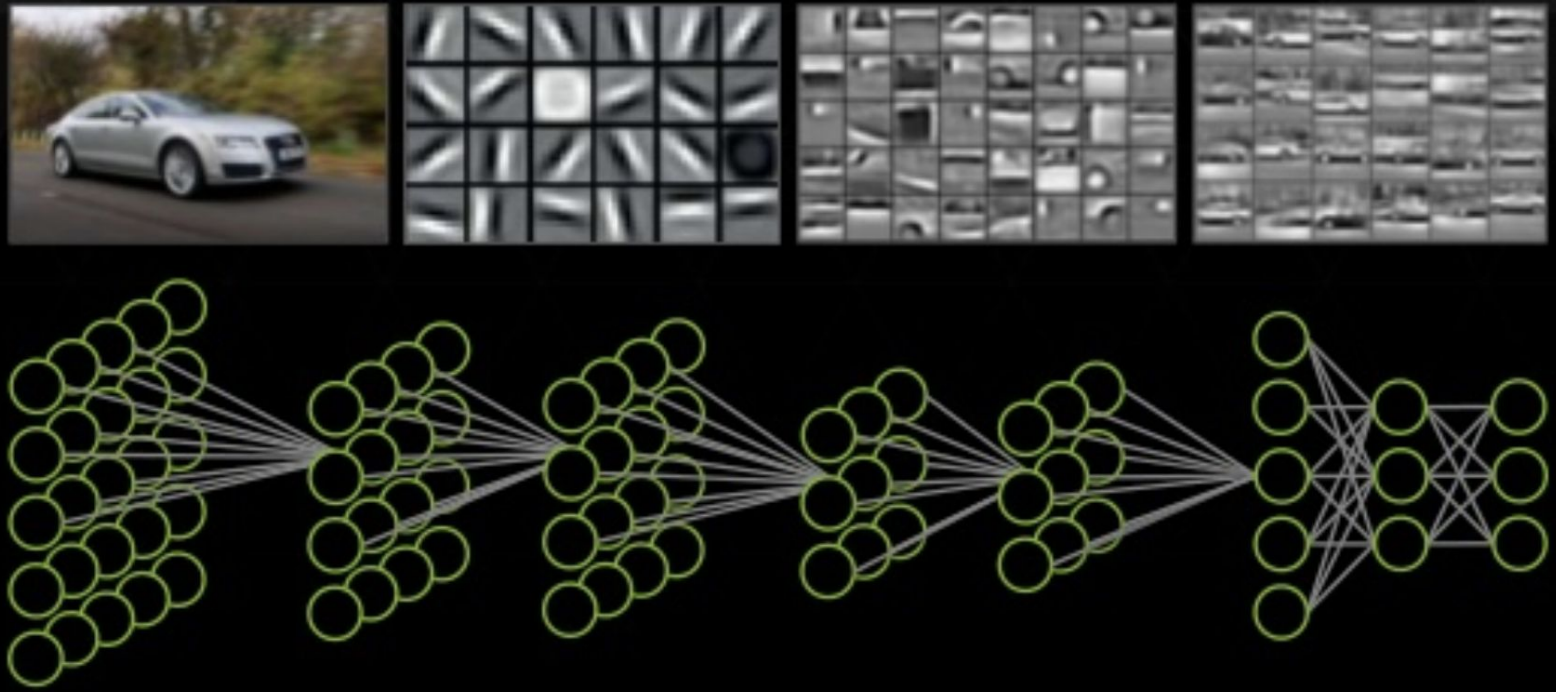
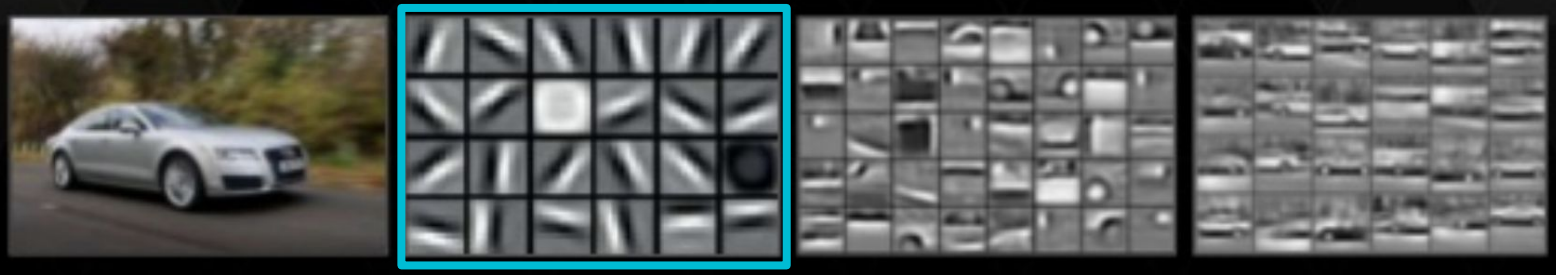


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse , Rajesh Ranganath, and Andrew Ng.

HOW A DEEP NEURAL NETWORK SEES



Image

"Audi A7"

Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse , Rajesh Ranganath, and Andrew Ng.

HOW A DEEP NEURAL NETWORK SEES

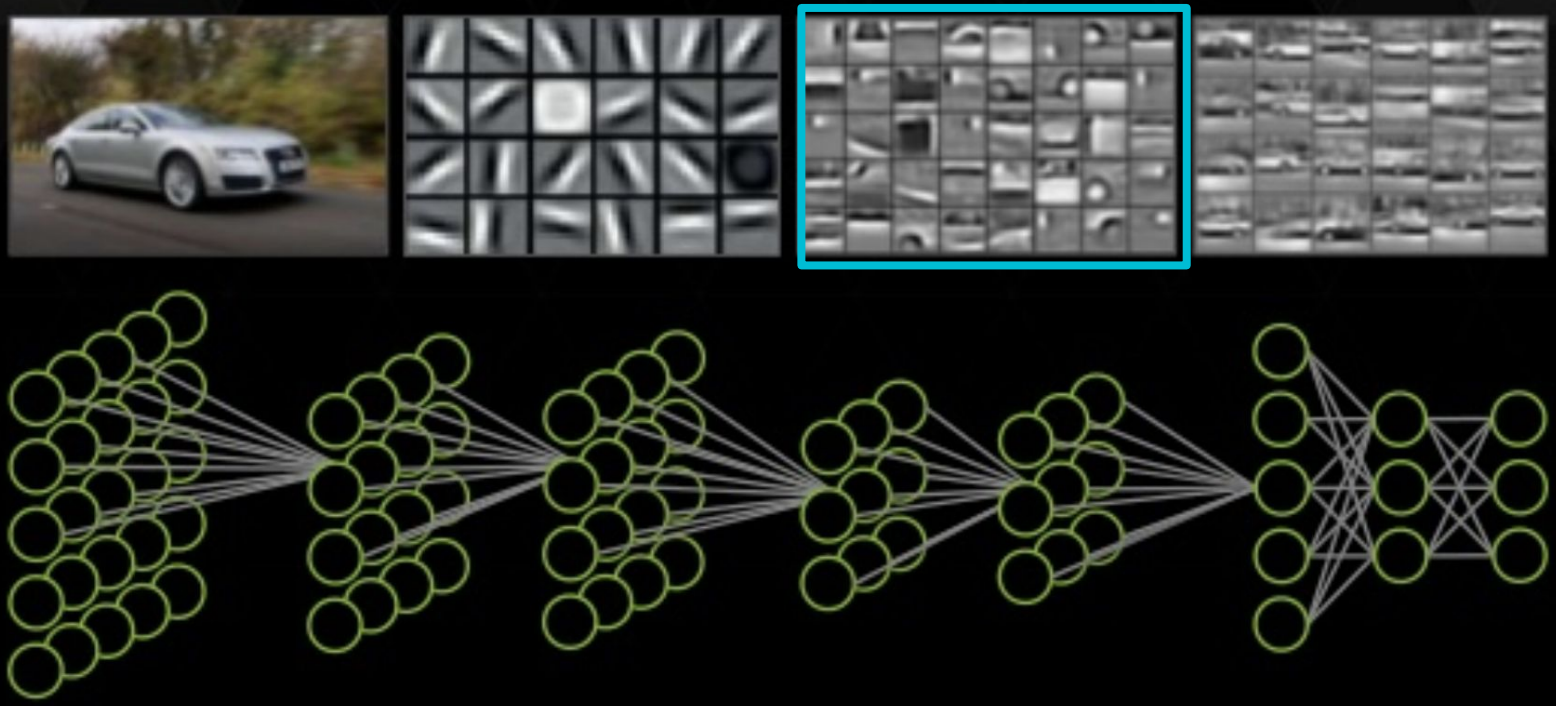


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse , Rajesh Ranganath, and Andrew Ng.

HOW A DEEP NEURAL NETWORK SEES

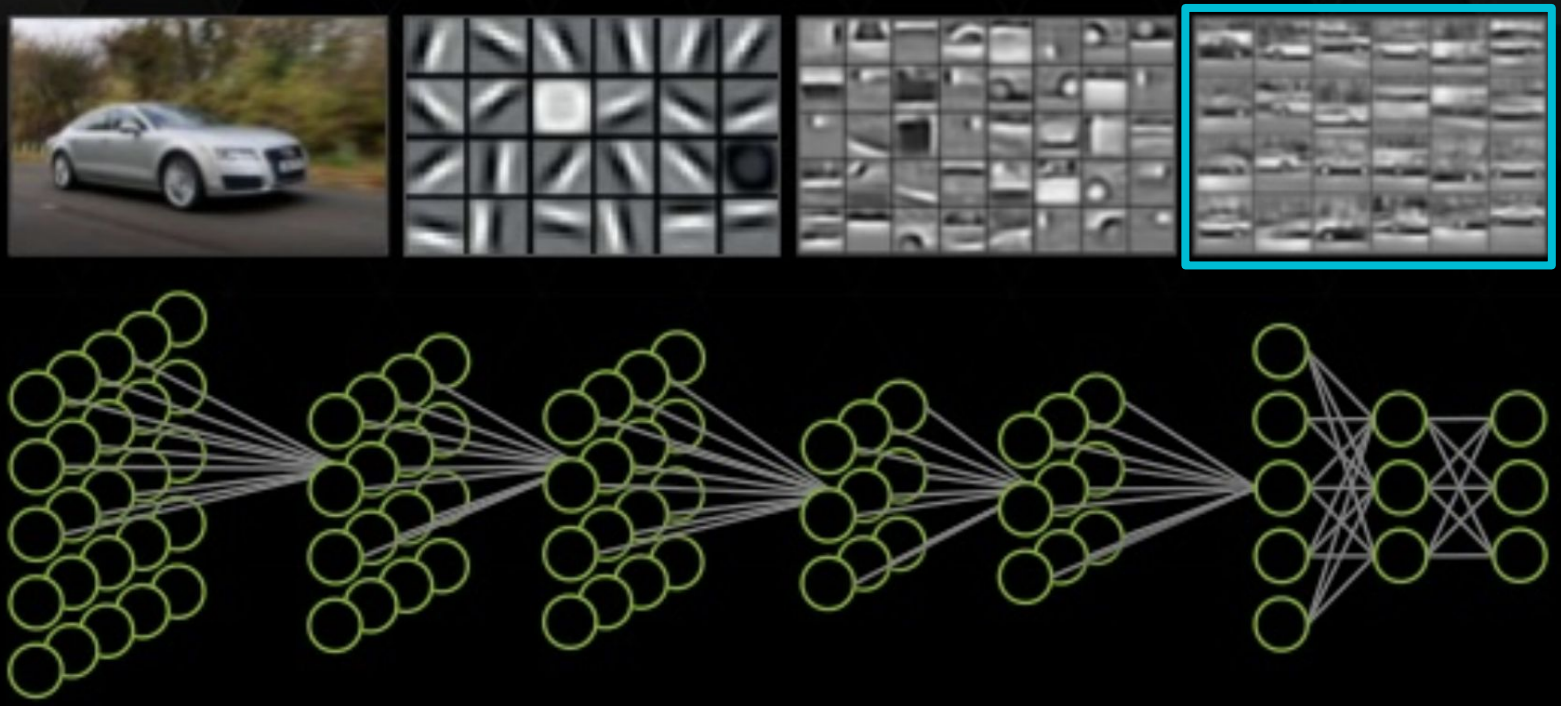


Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse , Rajesh Ranganath, and Andrew Ng.

CODING AND IMPROVING MODELS

Coding In TensorFlow and Keras

- 

Coding In TensorFlow and Keras

- 

- Sample code
  - https://www.kaggle.com/dansbecker/programming-in-tensorflow-and-keras

Coding In TensorFlow and Keras

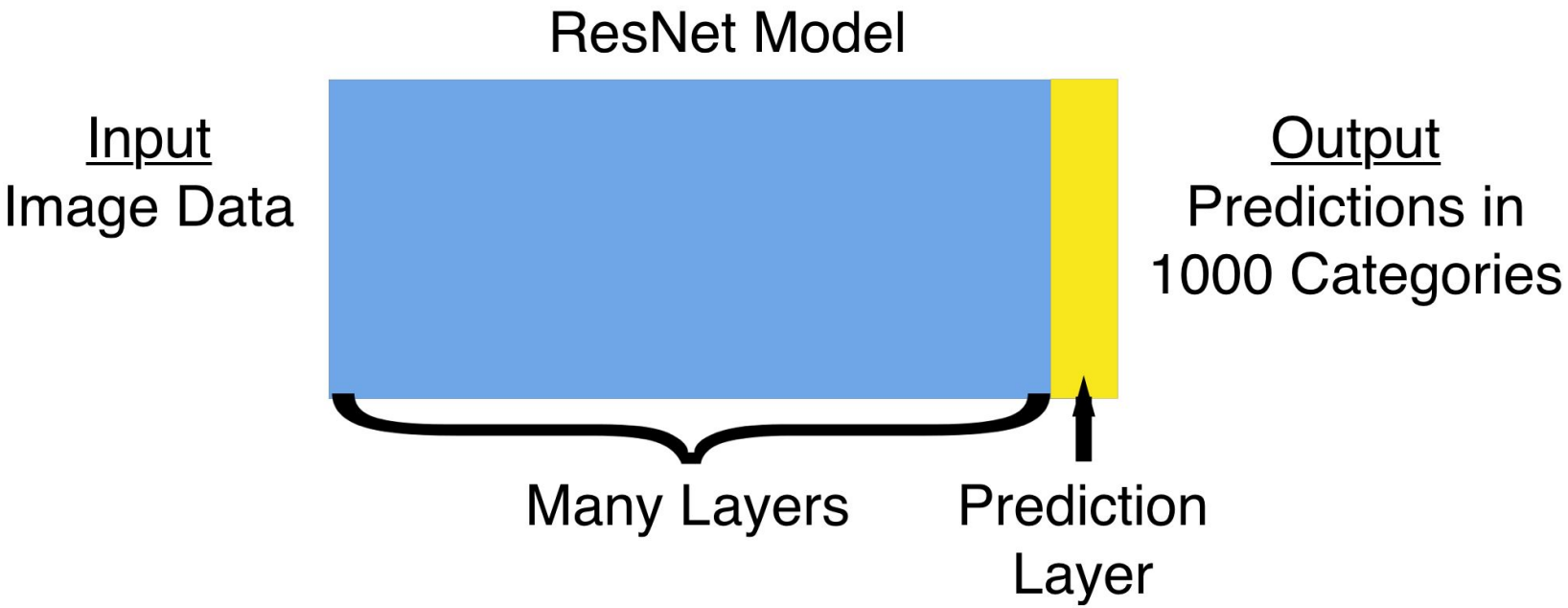
- 

- Sample code
  - https://www.kaggle.com/dansbecker/programming-in-tensorflow-and-keras

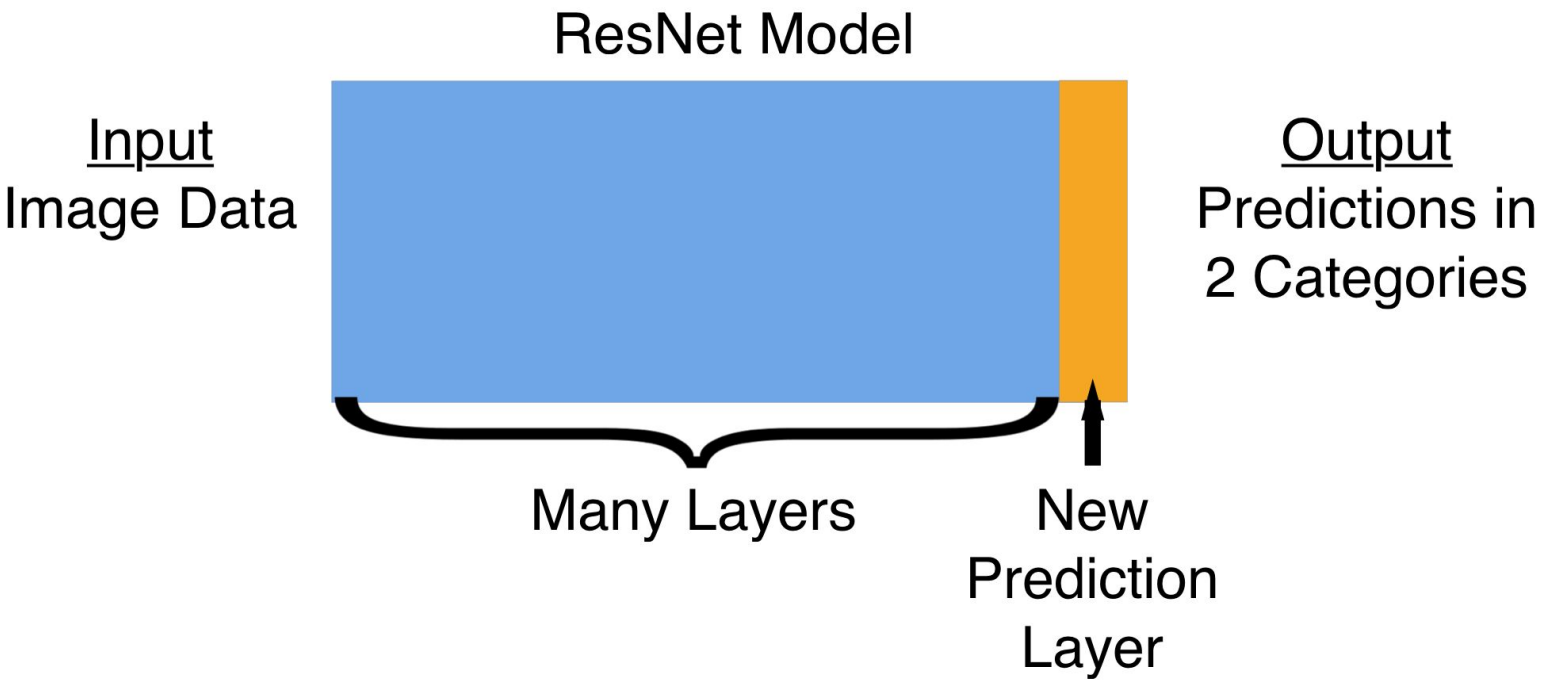
- Exercise
  - https://www.kaggle.com/dansbecker/exercise-coding-in-tensorflow-and-keras/

TRANSFER LEARNING

Transfer Learning



Replace Last Layer



#ODSC

Transfer Learning

- Sample code
  - https://www.kaggle.com/dansbecker/transfer-learning

- Exercise
  - https://www.kaggle.com/dansbecker/exercise-using-transfer-learning

DATA AUGMENTATION

Data Augmentation

Urban

Data Augmentation

Urban

Urban

#ODSC

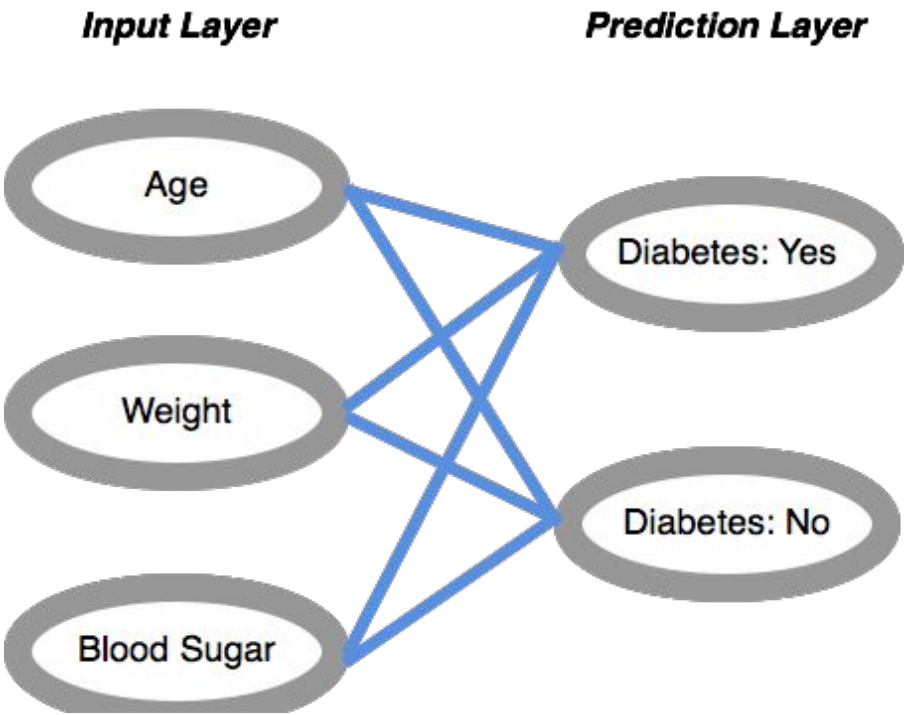
Data Augmentation

- Sample code
  - https://www.kaggle.com/dansbecker/data-augmentation

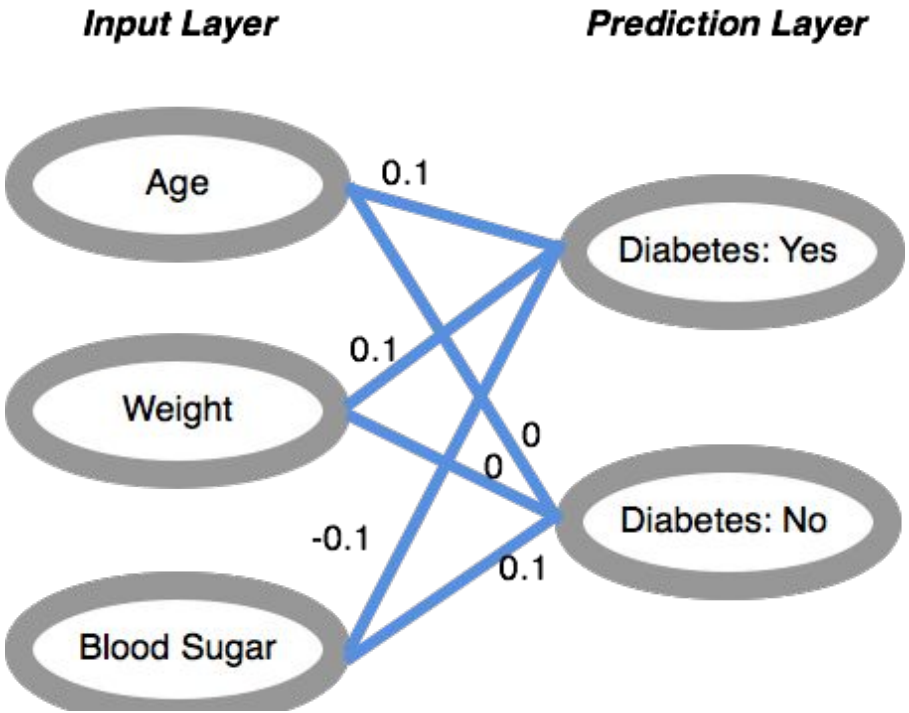
- Exercise
  - https://www.kaggle.com/dansbecker/exercise-data-augmentation

BACK-PROPAGATION AND GRADIENT DESCENT

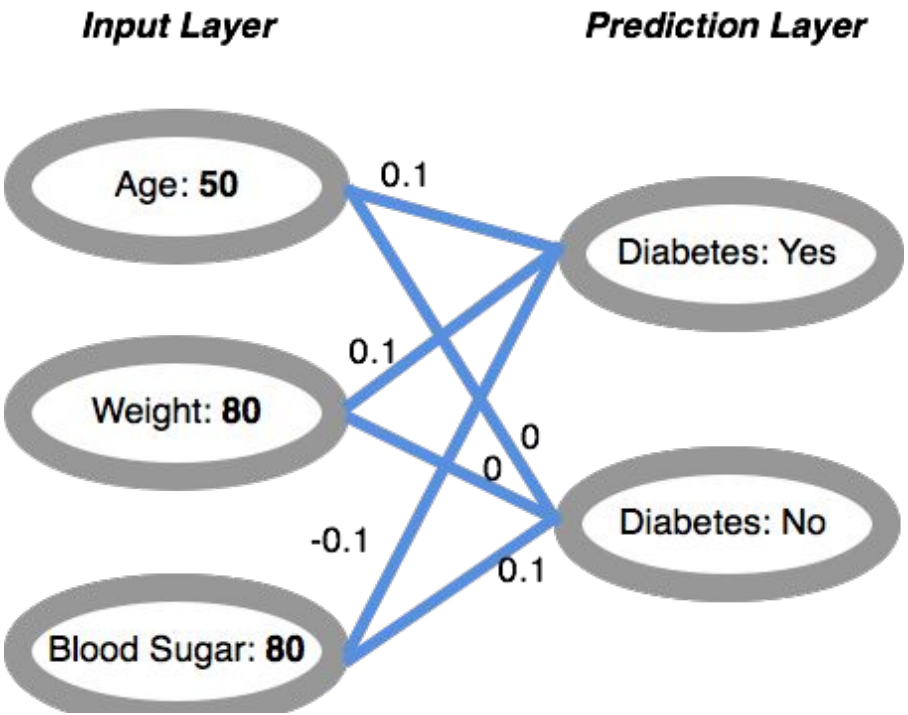
Simple Model Based on Tabular Data



Simple Model Based on Tabular Data

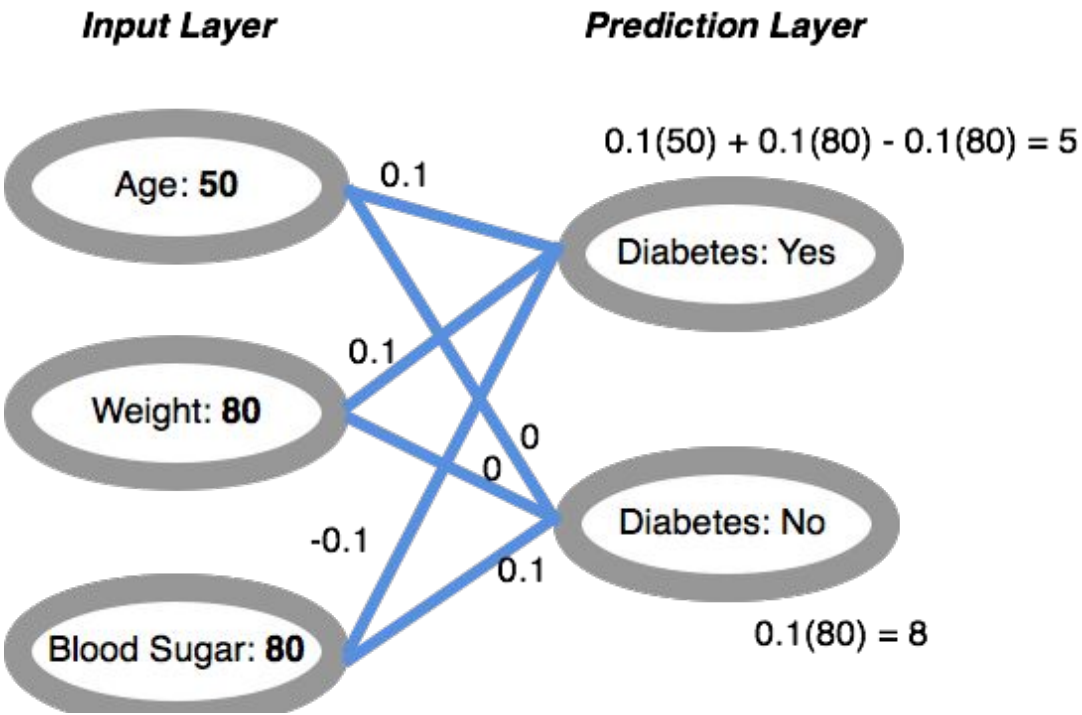


Simple Model Based on Tabular Data

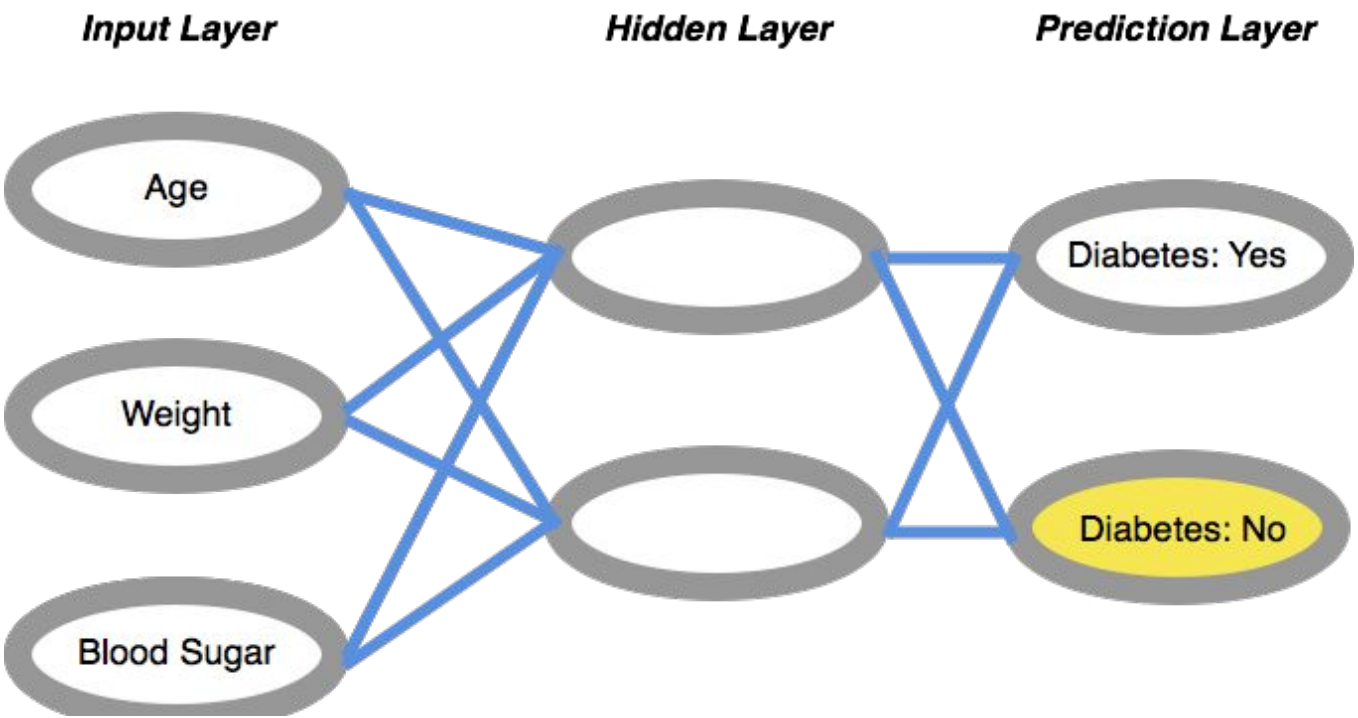


#ODSC

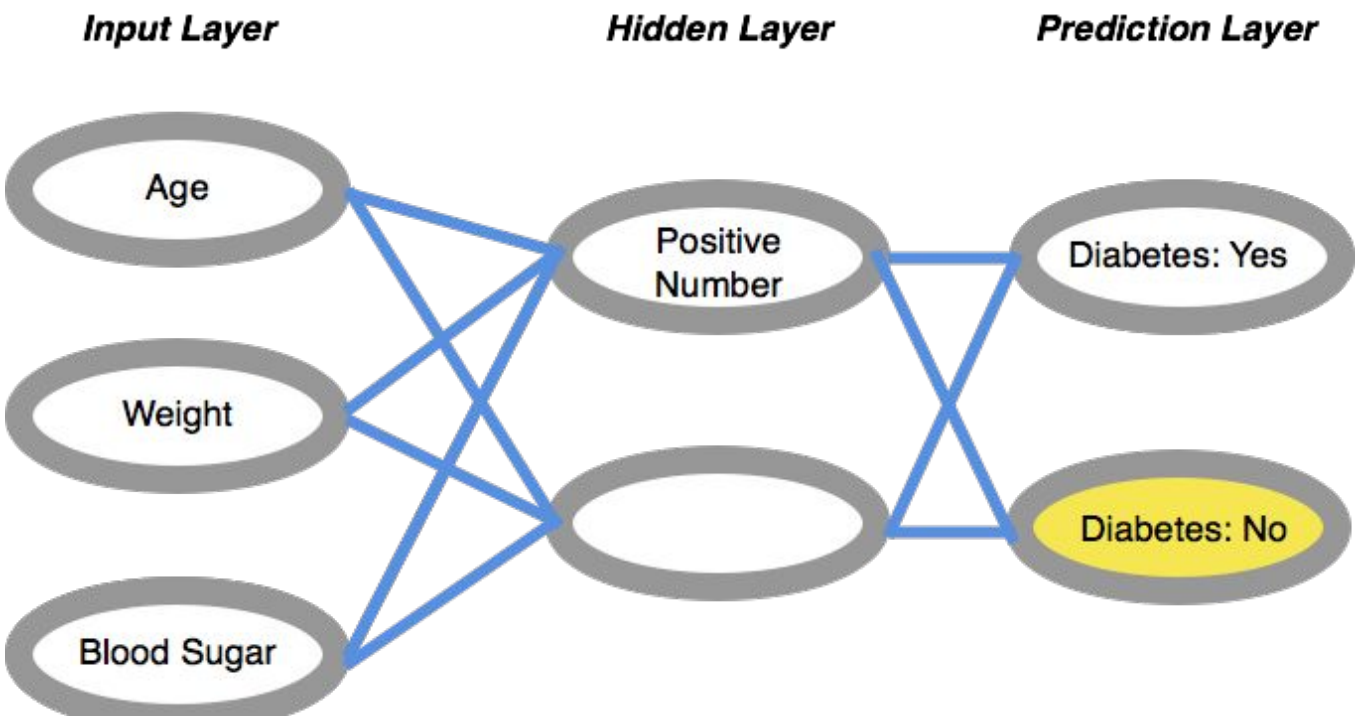
Simple Model Based on Tabular Data



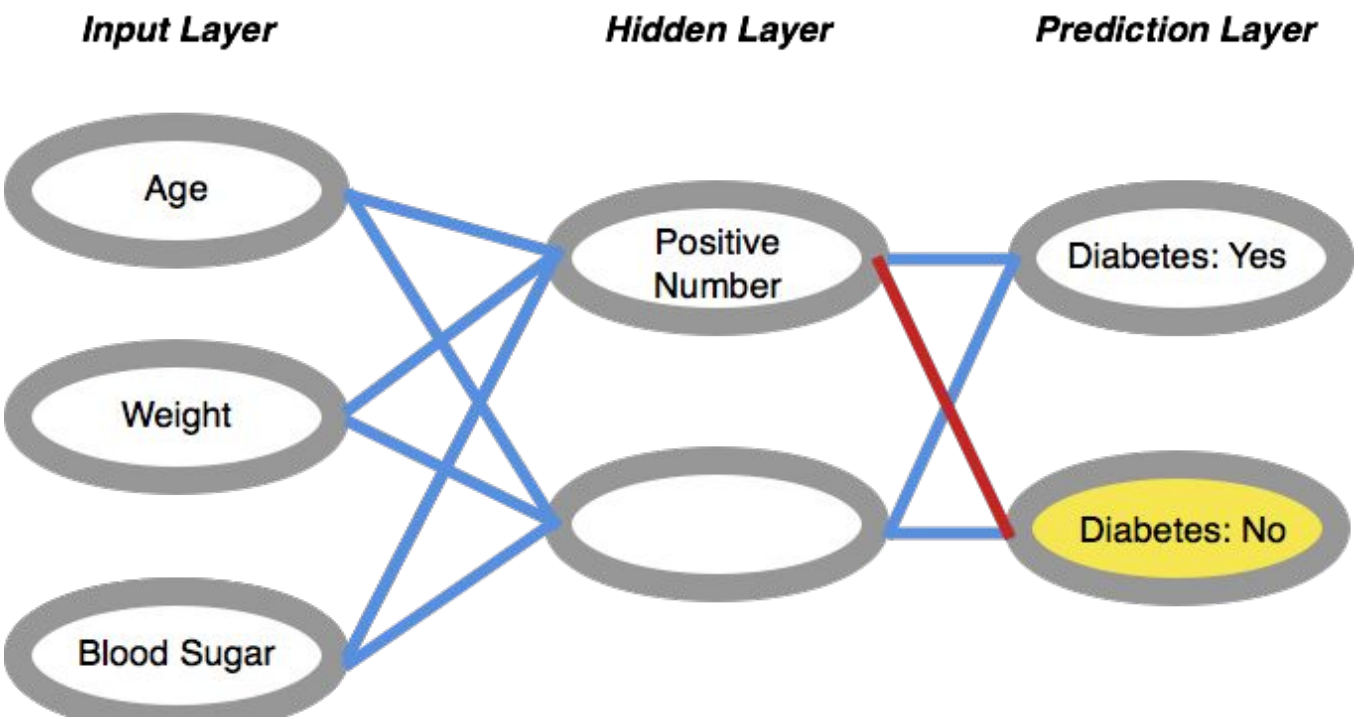
Simple Model Based on Tabular Data



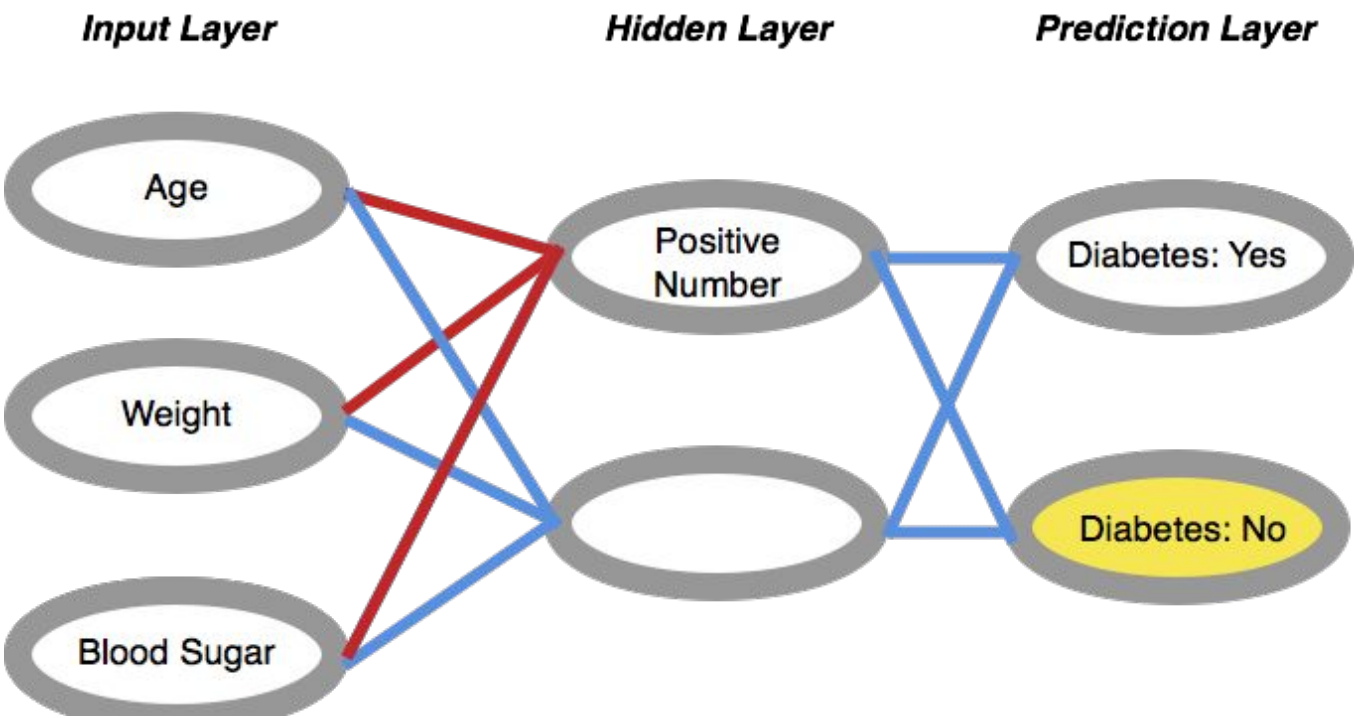
Simple Model Based on Tabular Data



Simple Model Based on Tabular Data



Simple Model Based on Tabular Data

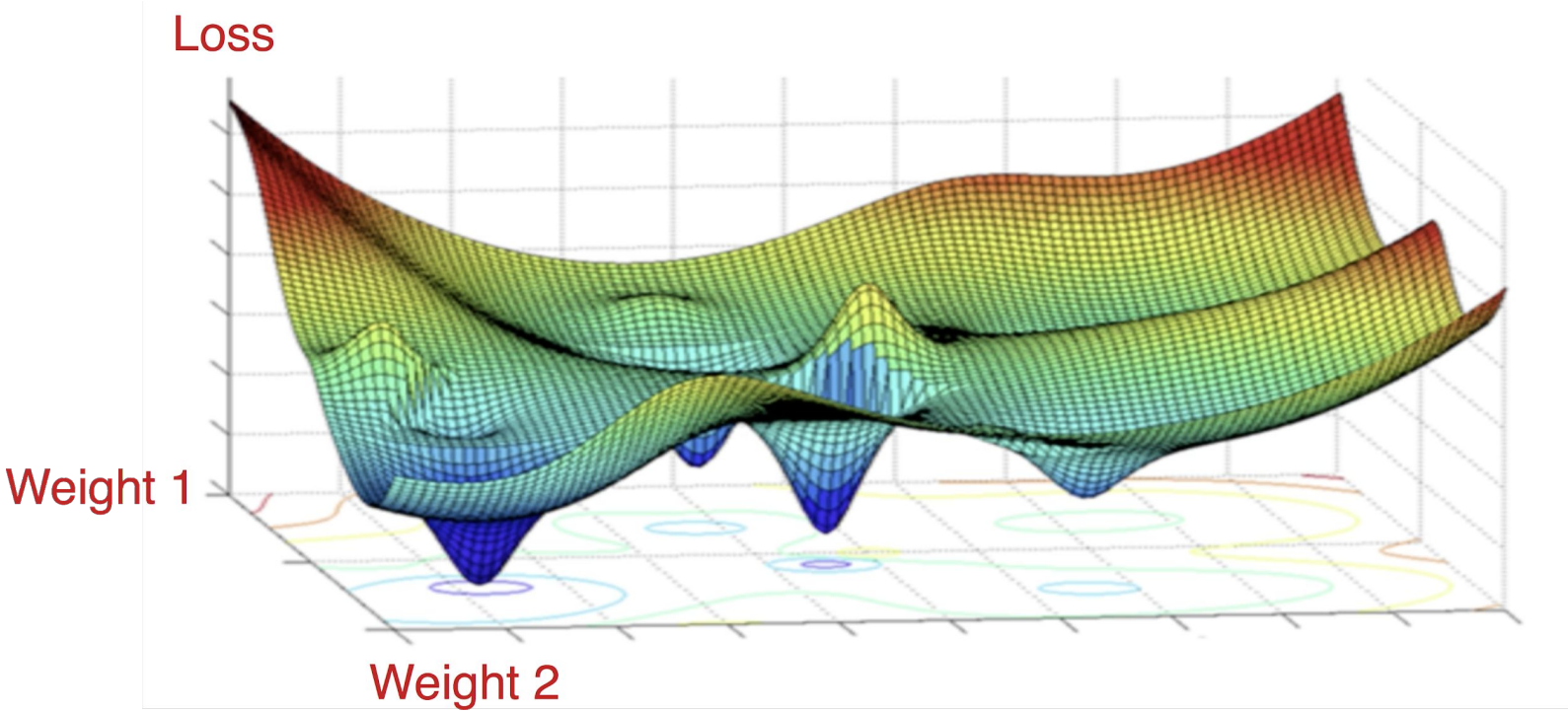


Loss Functions

Loss = f(actual, predicted)

Loss Functions

Loss = f(actual, predicted)



MODELING FROM SCRATCH

Modeling From Scratch

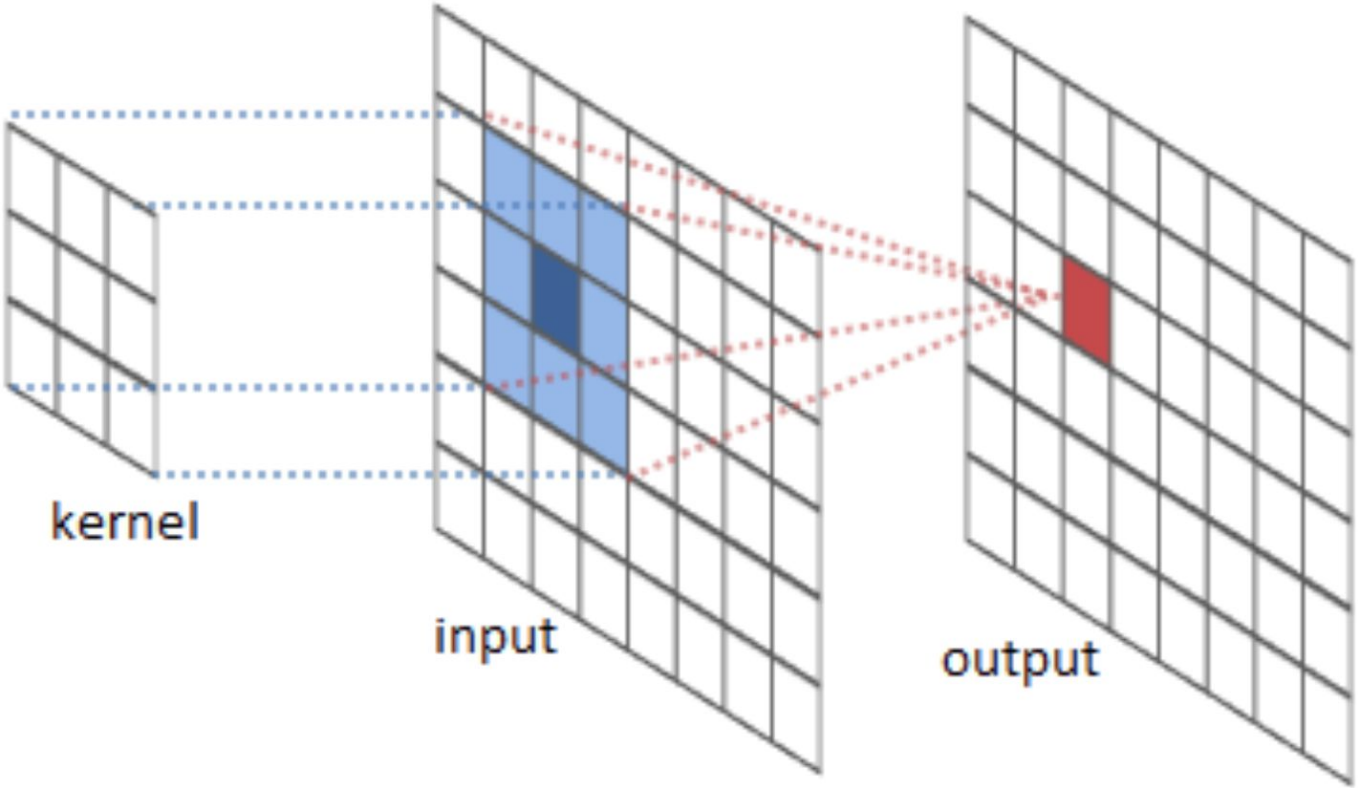
- Sample code
  - https://www.kaggle.com/dansbecker/deep-learning-from-scratch

- Exercise
  - https://www.kaggle.com/dansbecker/exercise-modeling-from-scratch

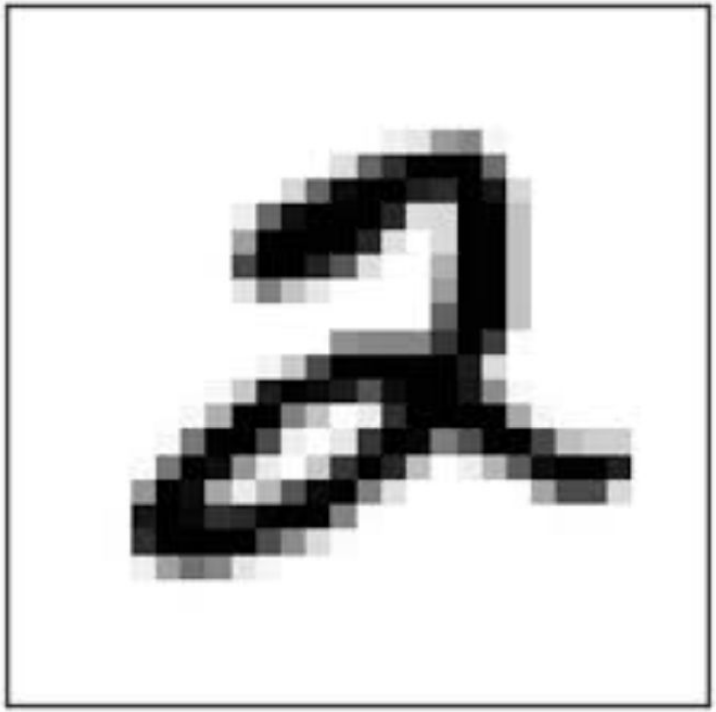
Bigger Models Usually Better

- Increase number of layers
- More convolutions per layer
- Control overfitting
  - Dropout
  - Stride length

Convolutions: The Building Block of Computer Vision

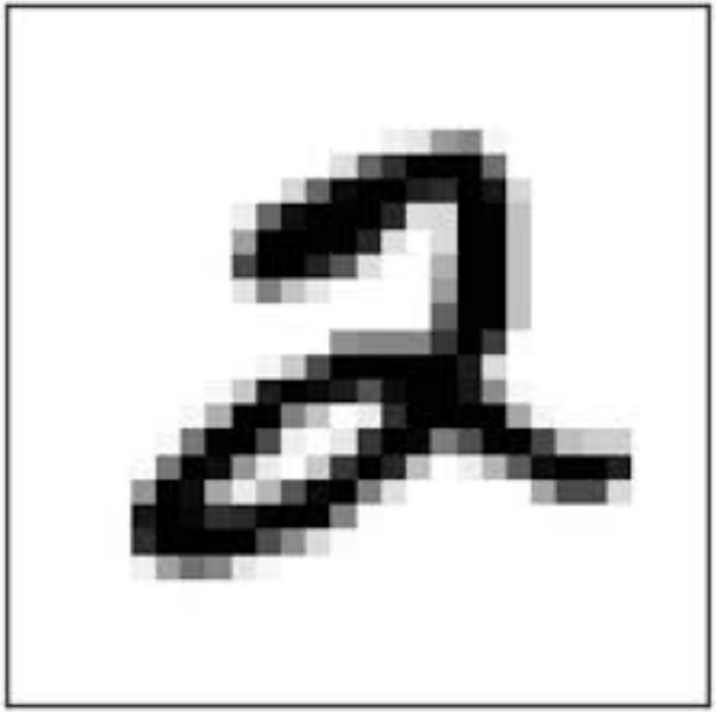


Stride Length 1



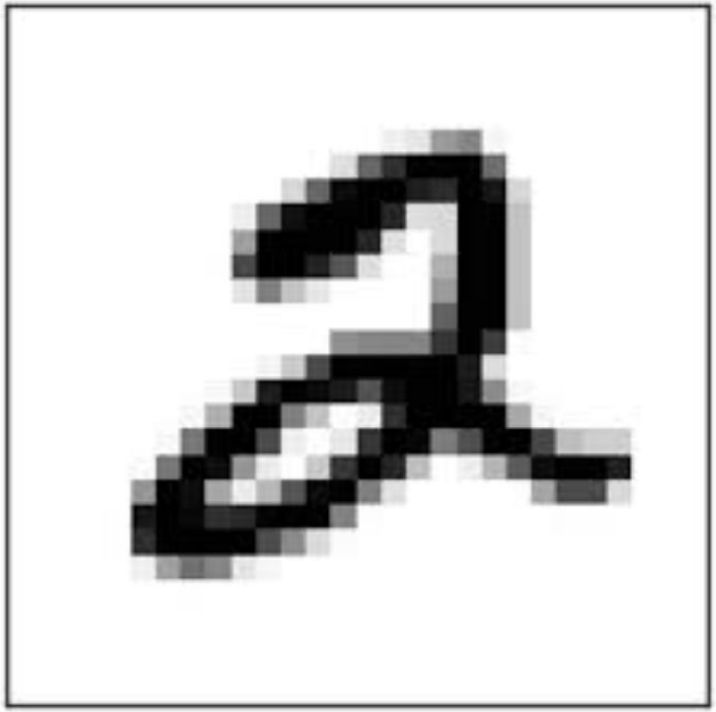
32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 1



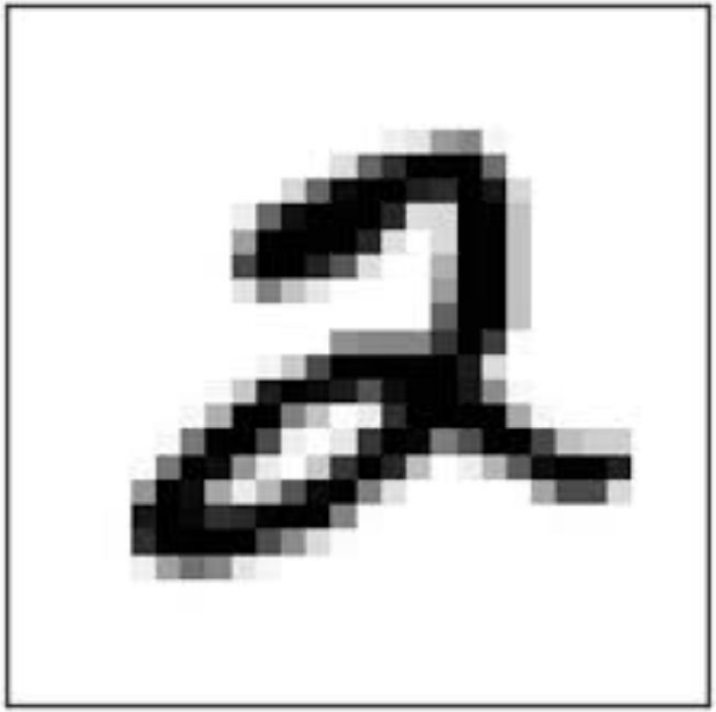
32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 1



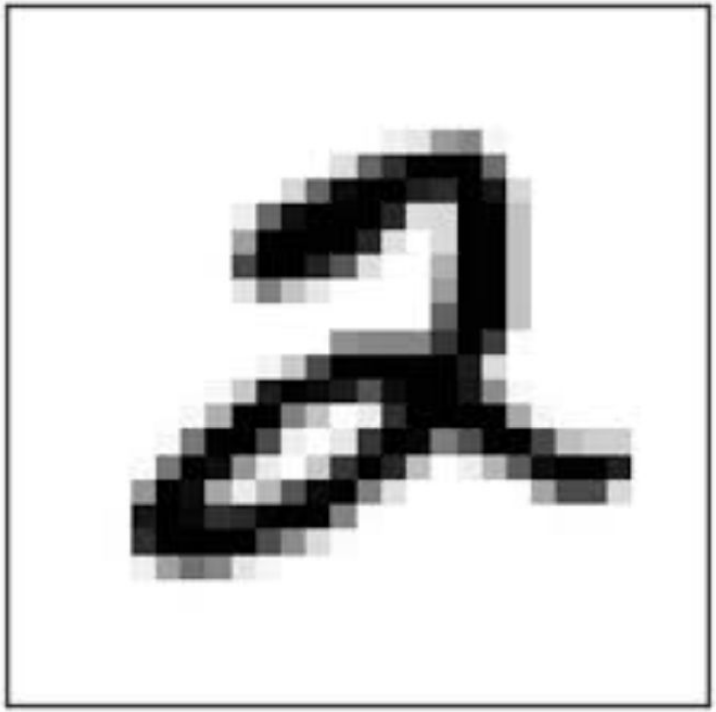
32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 1



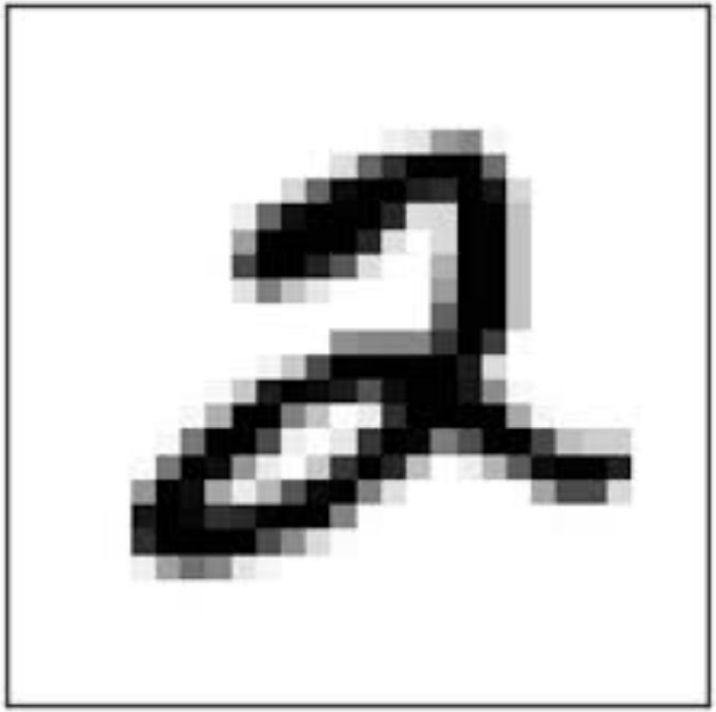
32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 2



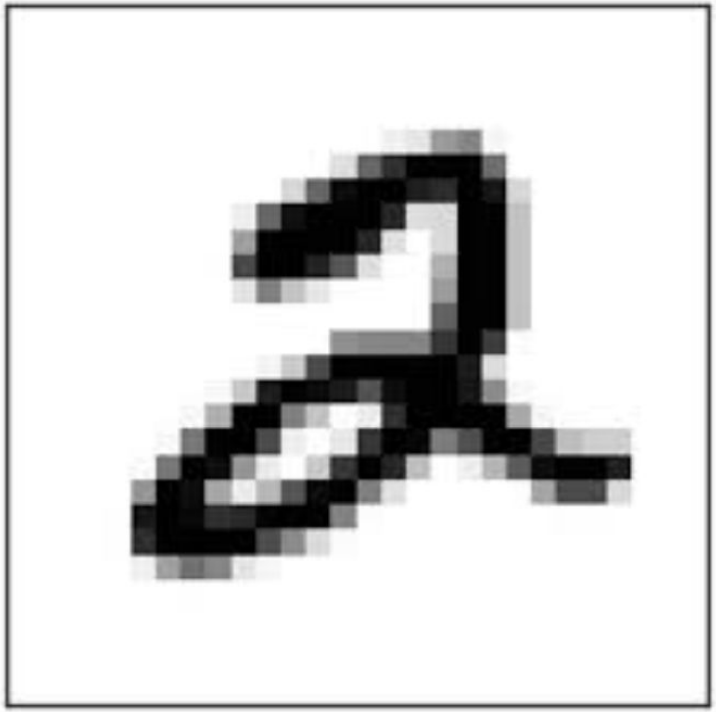
32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 2



32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Stride Length 2



32	16	24	55	...
18	12	99	123	...
44	88	31	99	...
55	94	31	88	...
...	...	...	...	...

Dropout and Stride Length

- Sample code
  - https://www.kaggle.com/dansbecker/dropout-and-strides-for-larger-models

- Exercise
  - https://www.kaggle.com/dansbecker/exercise-dropout-and-strides-for-larger-models

#ODSC

Next Steps

- Practice and experiment!
- Data on Kaggle
  - Digit Recognizer competition is a classic starter. Many others
- Documentation at keras.io is good
- Keras Functional API is gateway to even more sophisticated applications