Lecture Notes for

Machine Learning in Python

[👨‍🏫 , 👨‍💻 , 🐍 , 👨‍🔬]

Professor Eric Larson

Wide and Deep Networks

Lecture Agenda

- Logistics:
  - Grading Update
- Agenda:
  - Finish Keras Demo
  - Wide and Deep Networks
  - Wide and Deep Town Hall (if time)

Class Overview, by topic

Table Data Visualization

Numpy, Pandas, Seaborn
Overviews with some in-depth discussion

Dimension Reduction and Image Processing

Scikit-learn, Scikit Image,
Intuition only, Some mathematics

Linear and Logistic Regression

Numpy, Recreate API for Scikit-learn
Detailed mathematics for simple optimization
intuition for advanced optimization

Neural Networks and Back Prop.

Numpy
Detailed mathematics for NN operations

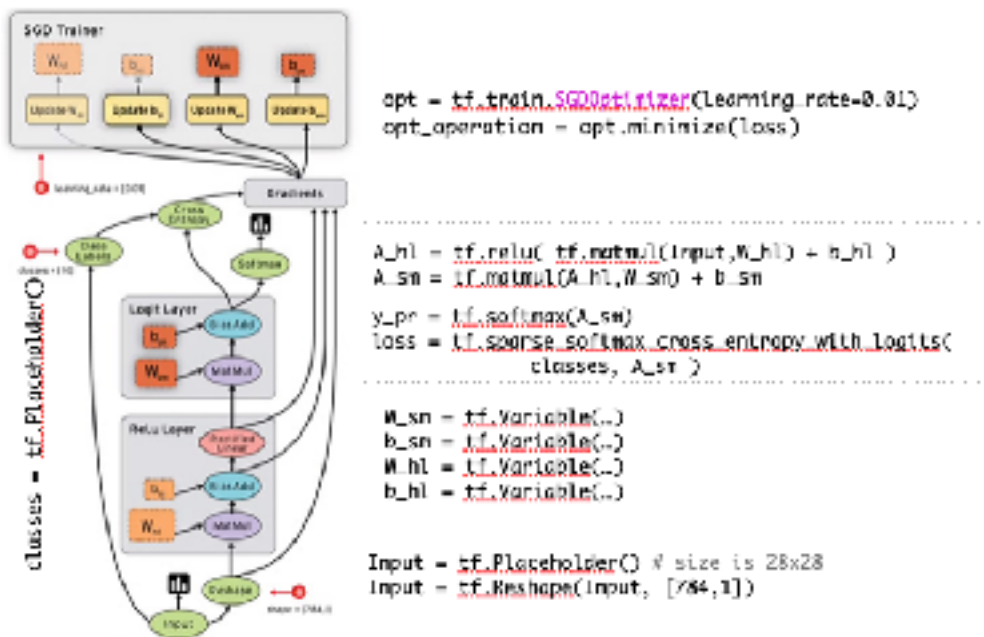**Wide and Deep Networks**

Convolutional Networks

Recurrent Networks

Keras, Tensorflow
Intuition, Detailed implement.

Ethics in Language Models

ConceptNet
Case studies

```
opt = tf.train.SGDOptimizer(Learning_rate=0.01)
opt_operation = opt.minimize(loss)
```

```
A_h1 = tf.relu( tf.matmul(Input,W_h1) + b_h1 )
A_sm = tf.matmul(A_h1,W_sm) + b_sm

y_pr = tf.softmax(A_sm)
loss = tf.sparse_softmax_cross_entropy_with_logits(
                classes, A_sm )
```

```
W_sm = tf.Variable(...)
b_sm = tf.Variable(...)
W_h1 = tf.Variable(...)
b_h1 = tf.Variable(...)
```

```
Input = tf.Placeholder()  # size is 28x28
Input = tf.Reshape(Input, [784,1])
```
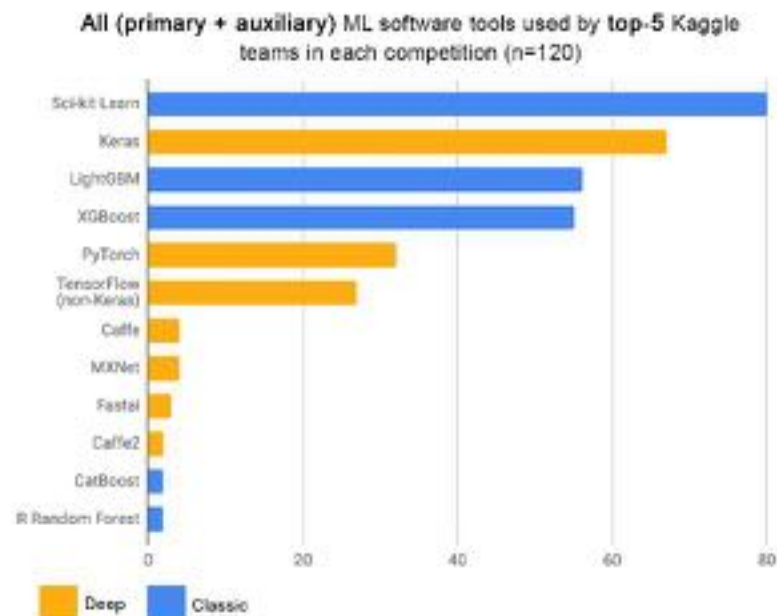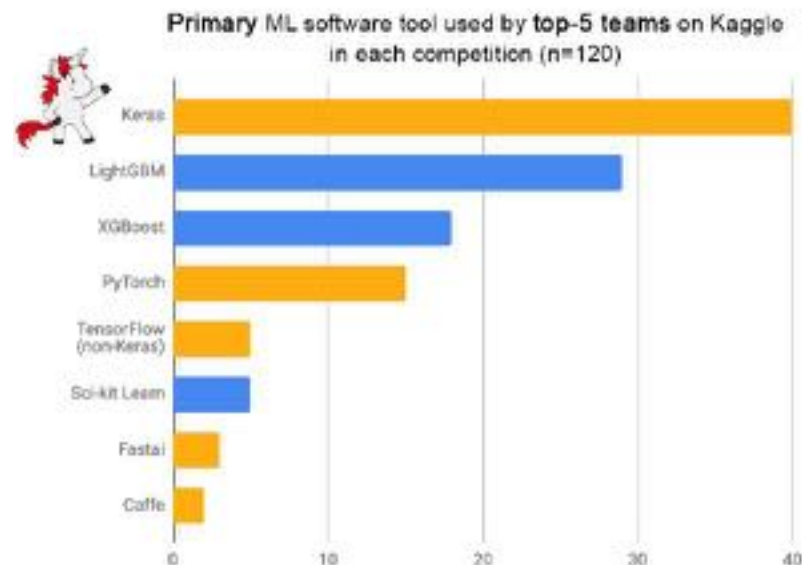
- **Keras Sequential API**
  - great for simple, feed forward models
- **Keras Functional API**
  - build models through series of nested functions
  - each "function" represents an operation in the NN

Primary ML software tool used by top-5 teams on Kaggle in each competition (n=120)

All (primary + auxiliary) ML software tools used by top-5 Kaggle teams in each competition (n=120)

Deep   Classic

Categorical Feature Embeddings Review

- One hot encoded data can be "embedded" through a matrix multiplication (column select)
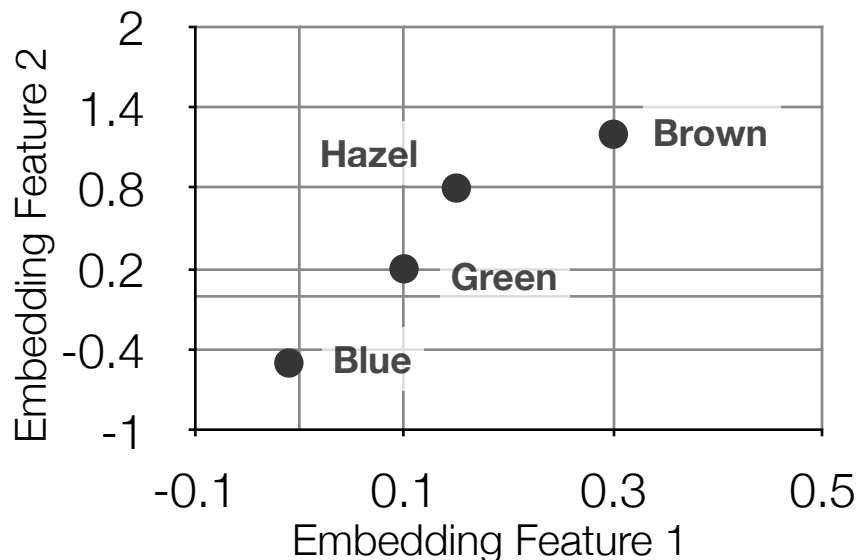
Trainable Matrix in Network

Eye Color

Reduced Dimensions

0.1	0.3	-0.01	0.15
0.2	1.2	-0.5	0.8

Num Categories

$$\cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \text{Green} & \rightarrow & 0 \\ \text{Brown} & \rightarrow & 1 \\ \text{Blue} & \rightarrow & 2 \\ \text{Hazel} & \rightarrow & 3 \end{matrix}$$

Embedding Feature 2 (y-axis): 2, 1.4, 0.8, 0.2, -0.4, -1
Embedding Feature 1 (x-axis): -0.1, 0.1, 0.3, 0.5
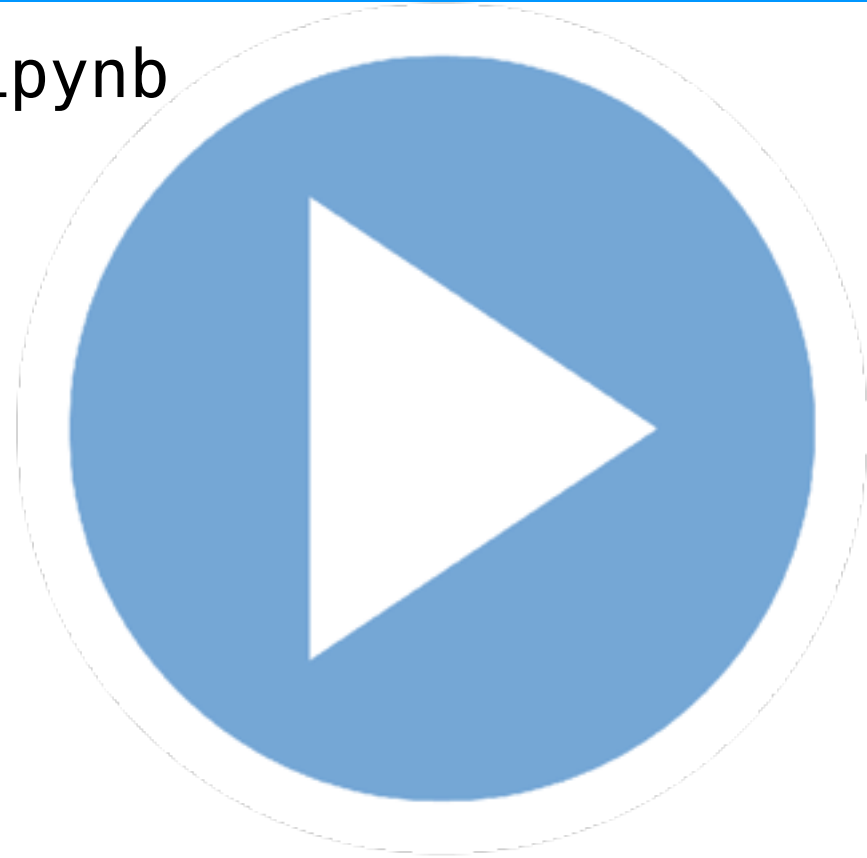
Brown, Hazel, Green, Blue

Computationally: there is no need to one hot encode eye color, we can just use the integer to index into column of **embedding matrix**

Using Embeddings in Keras Review

Single Branch Network

Concatenate

How many dimensions to make the output Embedding?

If $|X|$ = number of unique values

Option: $|P| = \left\lfloor \sqrt{|X|} \right\rfloor$

Numeric Output Vectors

-1.24
0.01

P_0

0.01

P_1

-0.14
1.04
0.41
-0.90

P_2

-0.21

P_N

-0.1
0.25
0.56

P_{N+1}

Trained Embedding Matrices

Embed 0   Embed 1   Embed 2   . . .   Embed N   Dense

X_0 X_1 X_2 X_N X_{N+1}

	Categorical Variable 0	Categorical Variable 1	Categorical Variable 2	Categorical Variable N	Numeric Feature Data
int:	2	0	17	1	
one hot:	0	1	0	0	0.256
	0	0	0	1	-0.01
	1	0	0		1.56
	0		…		3.1
	0		1		-1.2
	0		0		

No need to represent as one hot encoded!

. . .

63

10. Keras Wide and Deep.ipynb

Adding Embedding Branches

Make me slow down if I go too fast!!
By asking questions!

Wide and Deep Networks

When $p < 0.05$
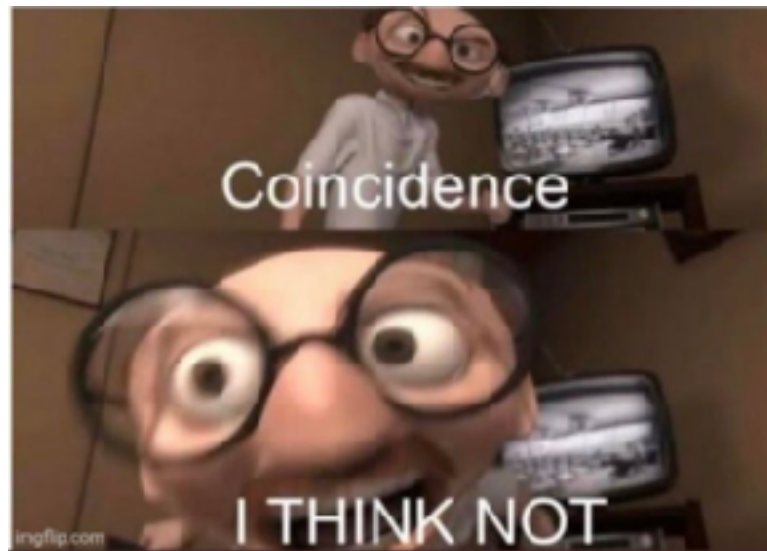
Wide & Deep Learning for Recommender Systems

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra,
Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil,
Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, Hemal Shah

Google Inc.[*]

ABSTRACT

Generalized linear models with nonlinear feature transfor-

have never or rarely occurred in the past. Recommenda-
tions based on memorization are usually more topical and
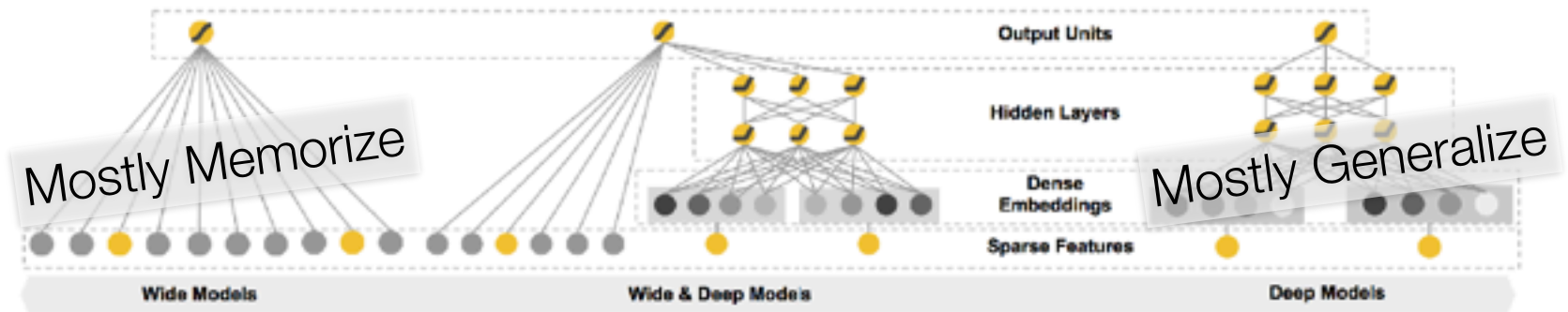
Figure 1: The spectrum of Wide & Deep models.

linear model with feature transformations for generic
recommender systems with sparse inputs.

- The implementation and evaluation of the Wide &
Deep recommender system productionized on Google

Why wide and deep?

Figure 1: The spectrum of Wide & Deep models.

Mostly Memorize **Mostly Generalize**
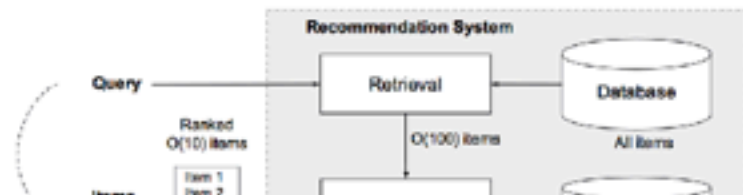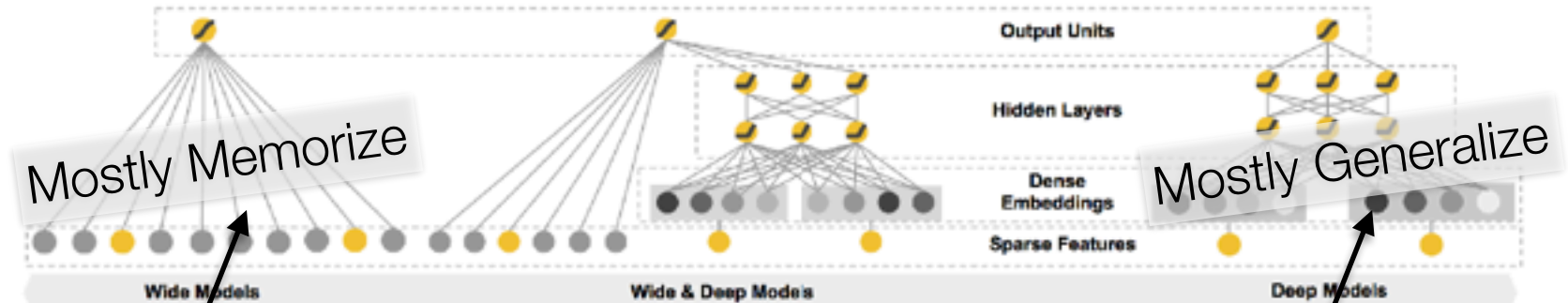
But why memorize?   **Obvious!**

- Categorical values have combinations that repeat!
  - so memorizing these values is not necessarily a bad strategy
  - let's make memorizing easy on one network

Wide networks (Memorize?)

- Wide refers to the expansion of features set
- Crossed feature columns of categorical features

- Movie Rating
  - G
  - PG
  - PG-13
  - R
  - Else
- Movie Genre
  - Action
  - Drama
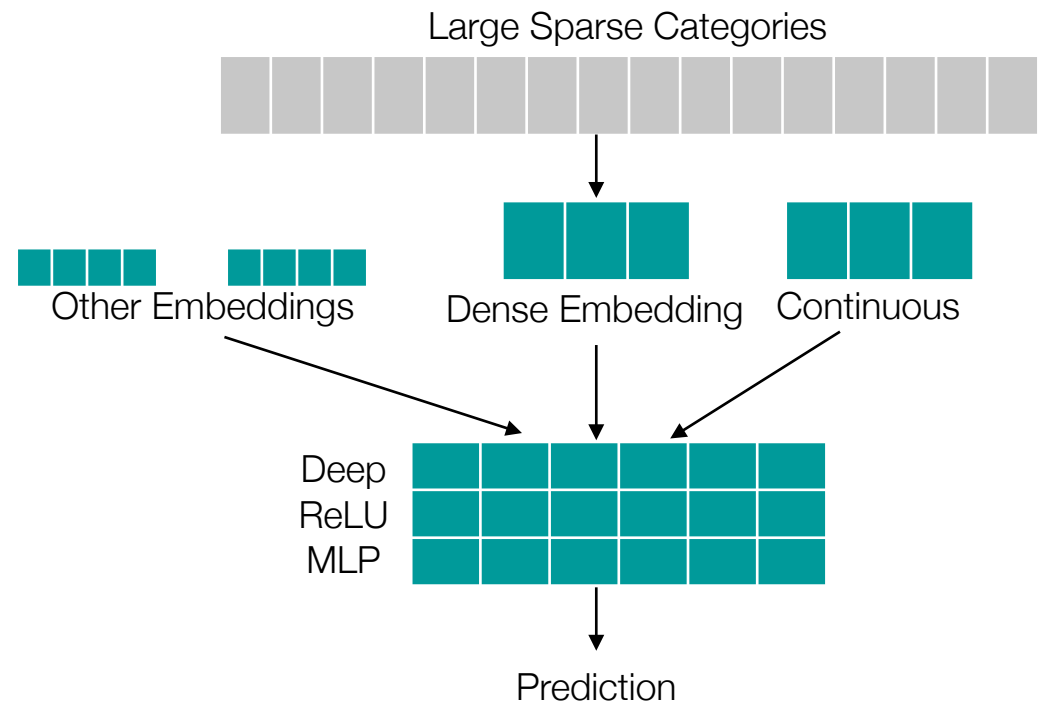 - Comedy
  - Horror
  - Else

Var 1   Var 2

Crossed feature values
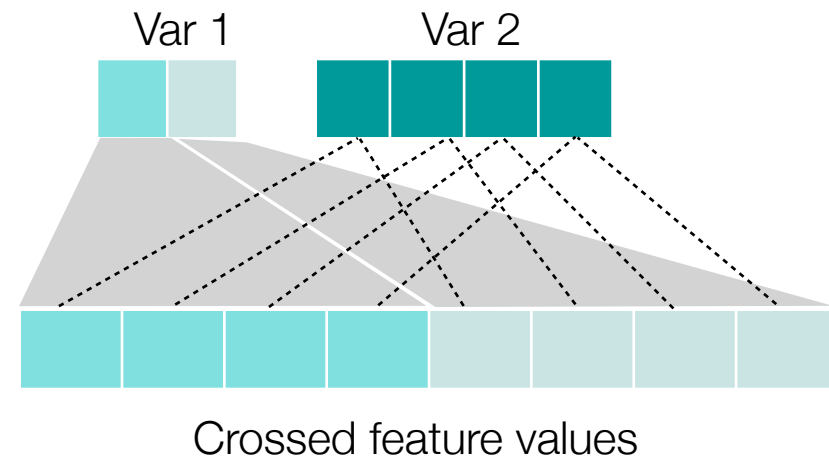
- Crossed feature "Rating-Genre"
  - G-Action, G-Drama, G-Comedy, G-Horror, G-else
  - PG-Action, PG-Drama, PG-Comedy, PG-Horror, G-else
  - and so on … one hot encoded

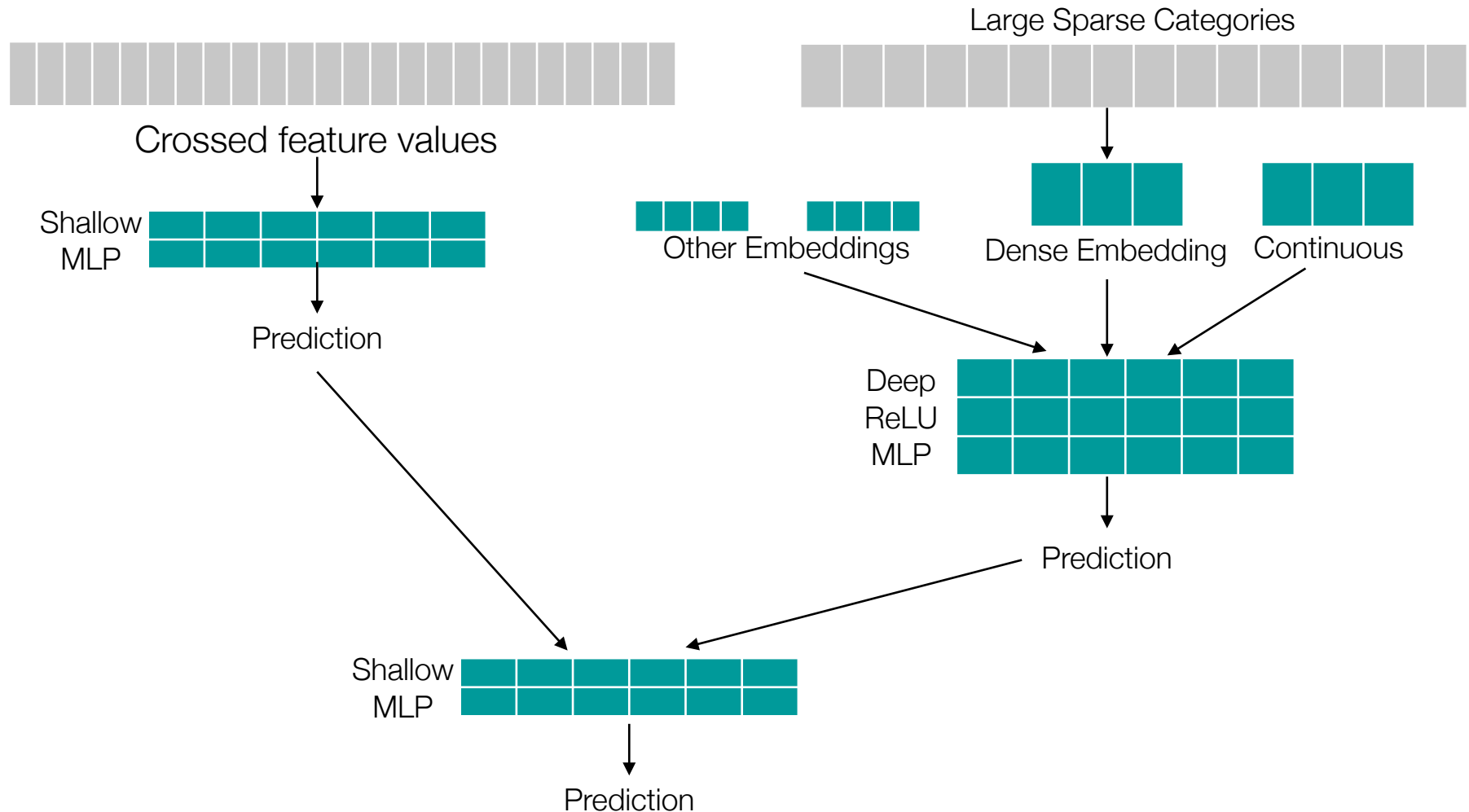# Sparse Embedding and Deep MLP (Generalize?)

- Deep refers to increasingly narrow hidden layers
- Embed into sparse representations via ReLU

- Movie Actors
  - Armand Assante
  - Meryl Streep
  - Danny Trejo
  - Kevin Bacon
  - Audrey Hepburn
  - …
-

Large Sparse Categories

Other Embeddings   Dense Embedding   Continuous

Deep
ReLU
MLP

Prediction

Combining Memorization and Generalization

- Deep refers to increasingly narrow hidden layers
- Embed into sparse representations via ReLU

Large Sparse Categories

Crossed feature values

Shallow MLP

Prediction

Other Embeddings   Dense Embedding   Continuous

Deep ReLU MLP

Prediction
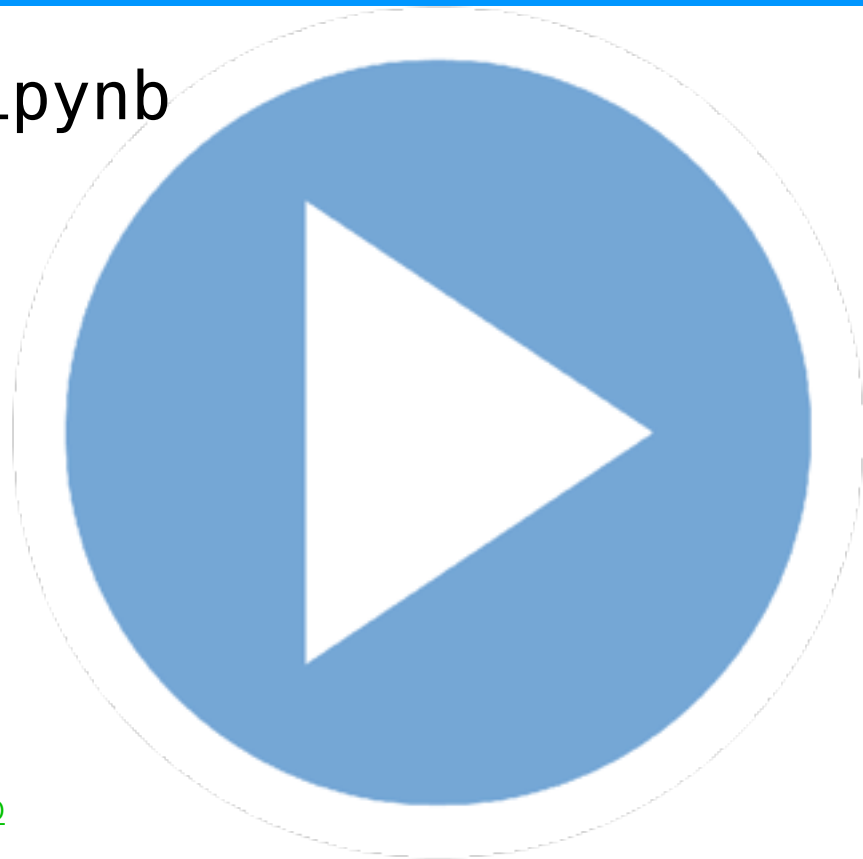
Shallow MLP

Prediction

`10. Keras Wide and Deep.ipynb`

The awful dataset:
Toy Census Data Example

Other tutorials:

https://www.tensorflow.org/tutorials/wide_and_deep

Town Hall, Wide and Deep Networks

End of Session

- Next Time:
  - Convolutional Neural Networks