

# Lecture Notes for **Machine Learning in Python**



Professor Eric Larson  
**Feature Spaces + 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

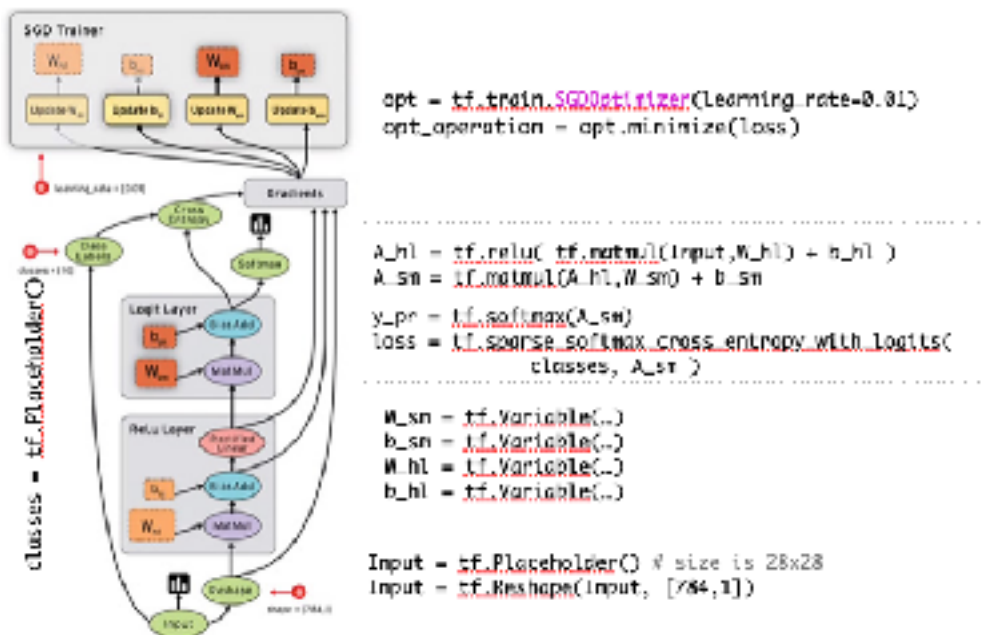
Recurrent  
Networks

Keras, Tensorflow  
Intuition, Detailed implement.

Ethics in  
Language Models

ConceptNet  
Case studies

# Last Time: Tensorflow and Keras

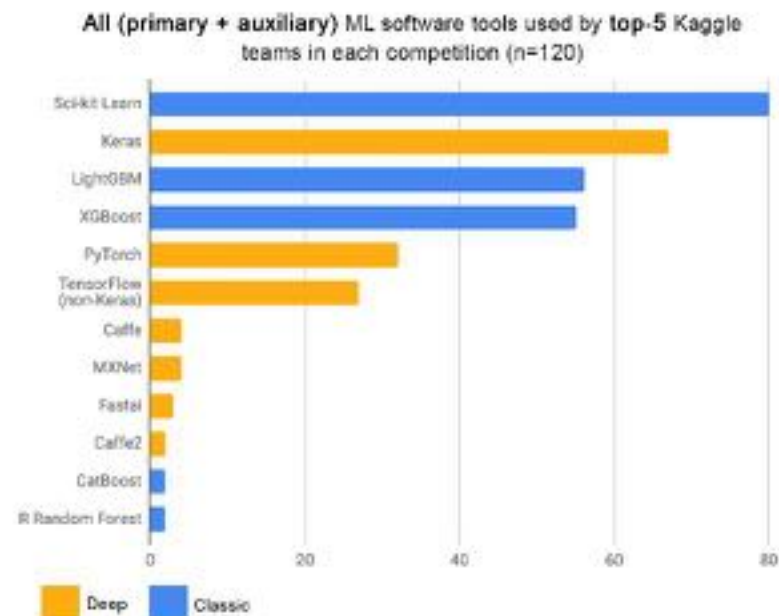
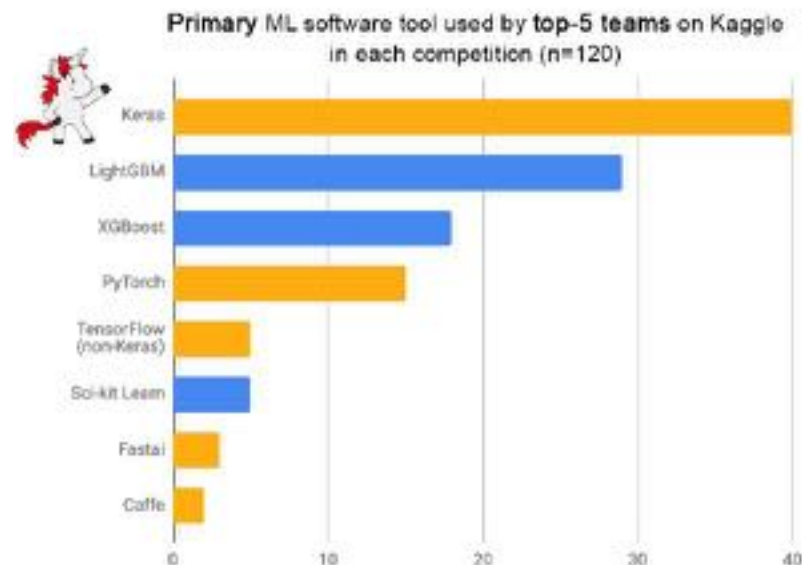


## • Keras Sequential API

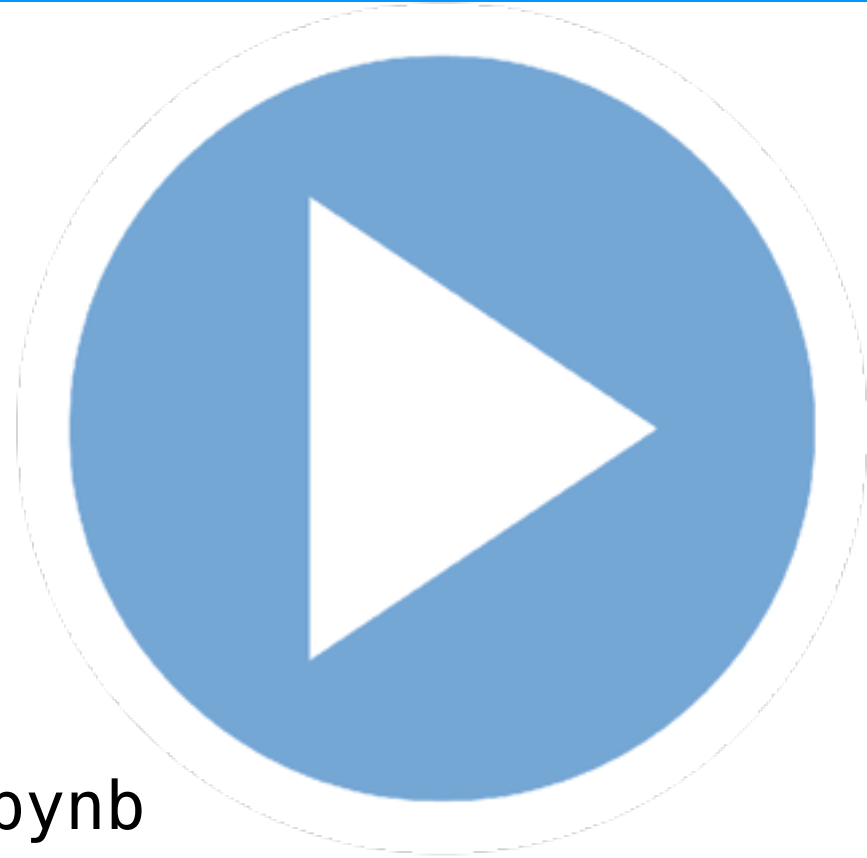
- great for simple, feed forward models

## • Keras Functional API

- build models through series of nested functions
- each “function” represents an operation in the NN



## Reinventing the MLP Wheel



10. Keras Wide and Deep.ipynb

10a. Keras Wide and Deep as TFData.ipynb

Make me slow down if I go too fast!!

# Categorical Feature Embeddings Review

- One hot encoded data can be “embedded” through a matrix multiplication (column select)

Trainable Matrix in Network

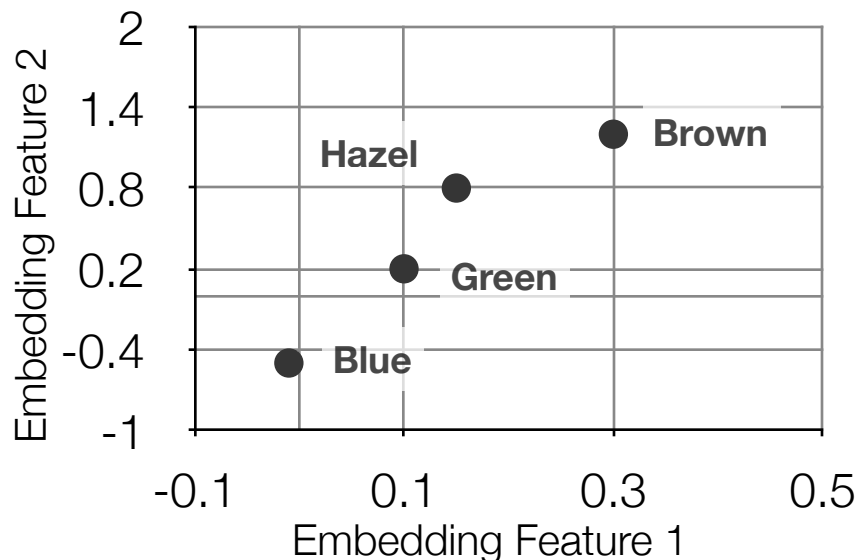
Reduced Dimensions

0.1	0.3	-0.01	0.15
0.2	1.2	-0.5	0.8

Num Categories

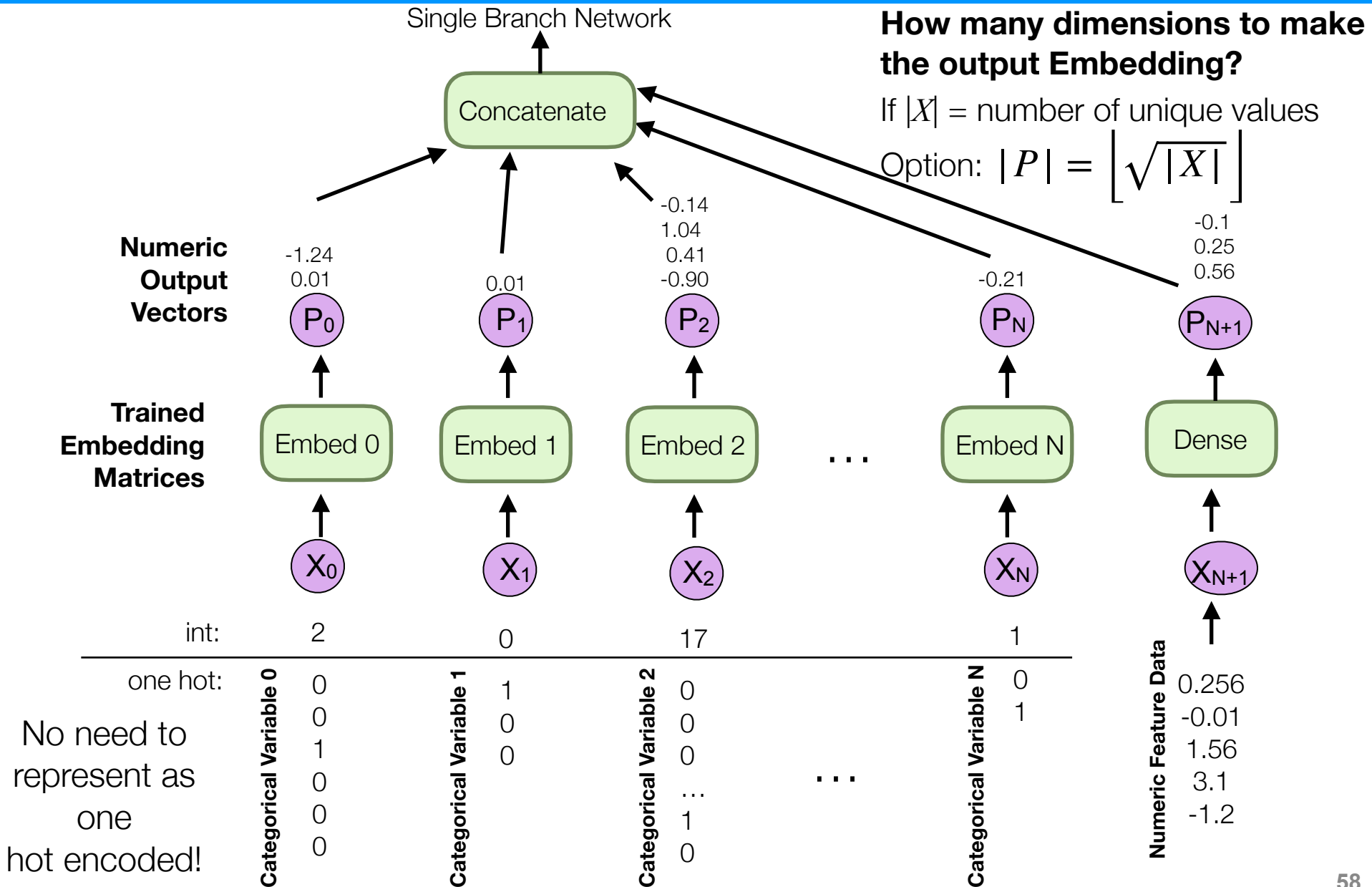
**Eye Color**

0	Green	→ 0
1	Brown	→ 1
0	Blue	→ 2
0	Hazel	→ 3

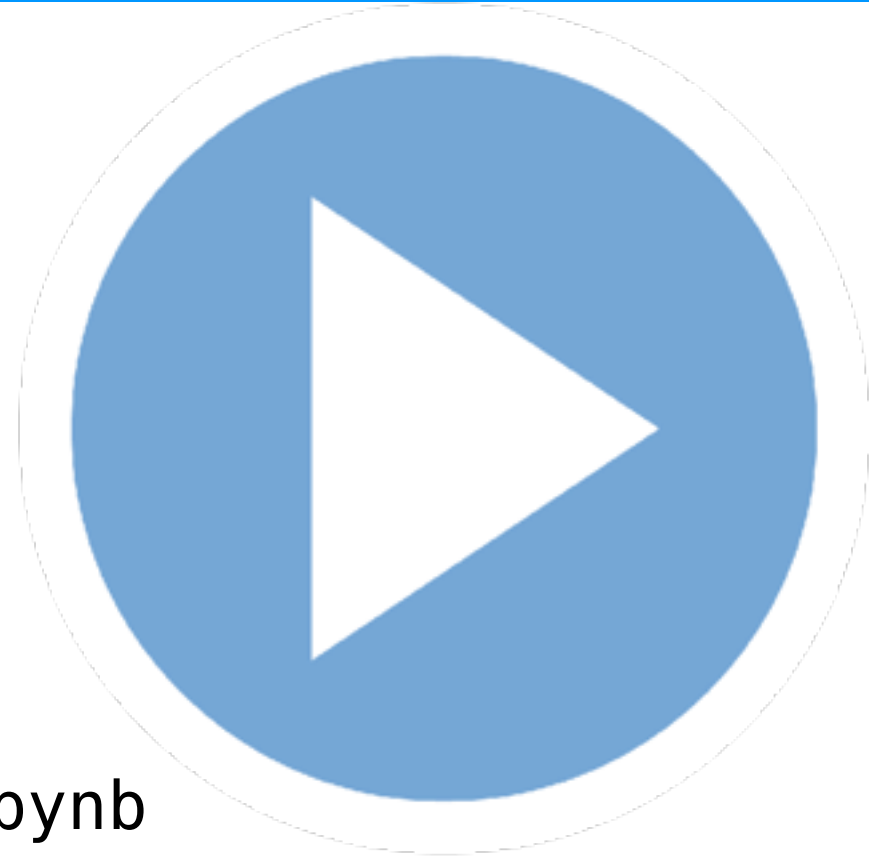


**Computationally:** there is no need to one hot encode eye color, we can just use the integer to index into column of **embedding matrix**

# Using Embeddings in Keras Review



Adding Embedding Branches



10. Keras Wide and Deep.ipynb

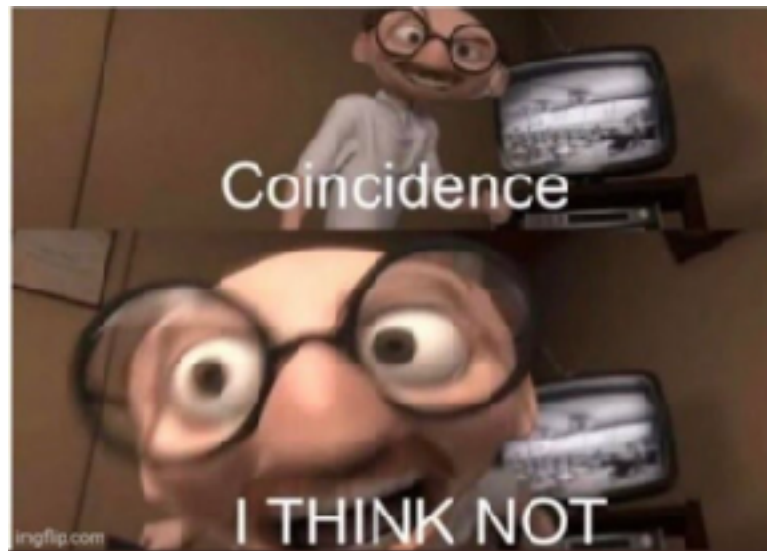
10a. Keras Wide and Deep as TFData.ipynb

Make me slow down if I go too fast!!



# Wide and Deep Networks

When  $p < 0.05$



# Wide and Deep

## 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. Recommendations based on memorization are usually more topical and

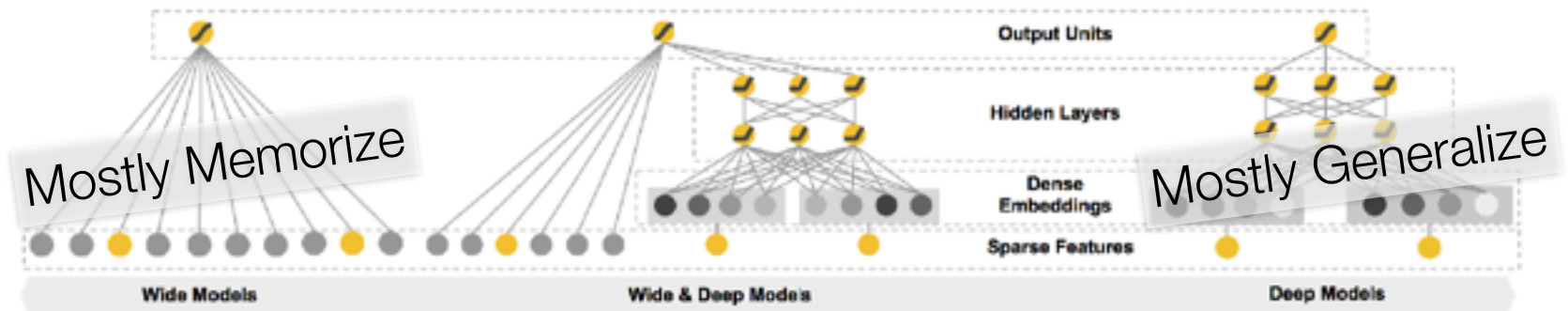
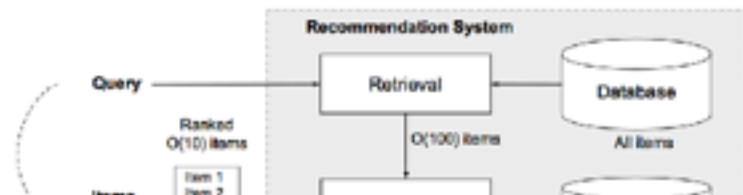


Figure 1: The spectrum of Wide & Deep models.

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?

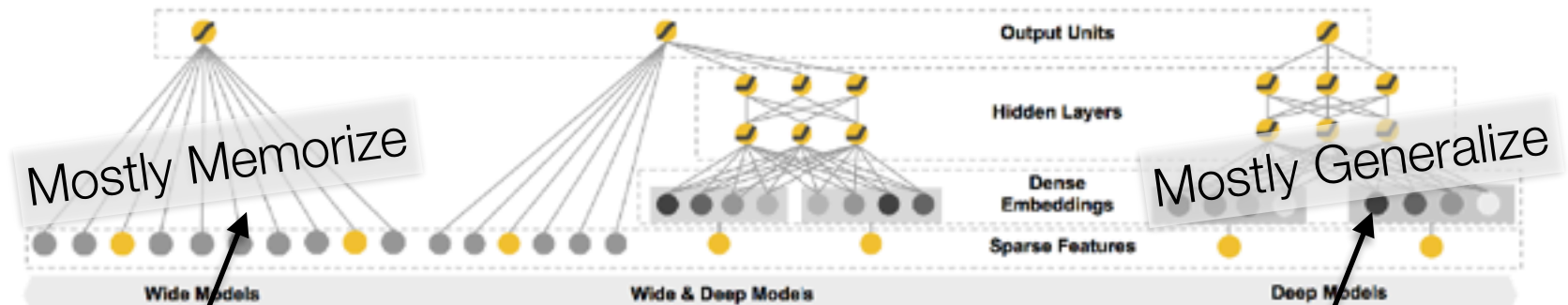


Figure 1: The spectrum of Wide & Deep models.

**But why memorize?**

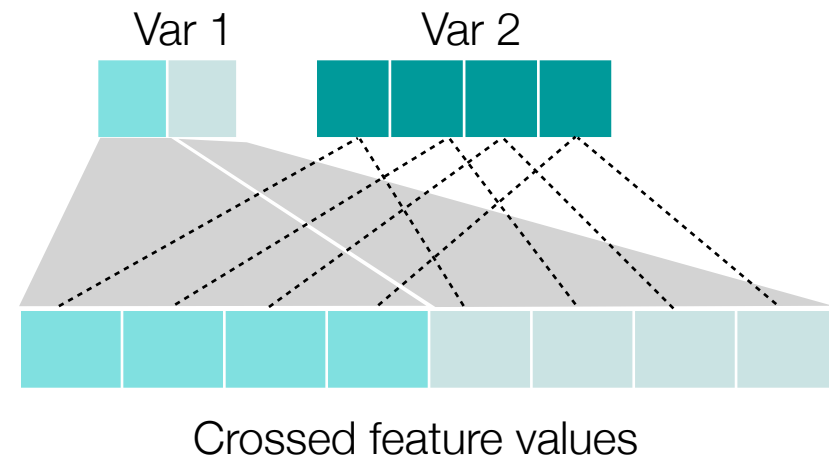
**Obvious!**

- 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

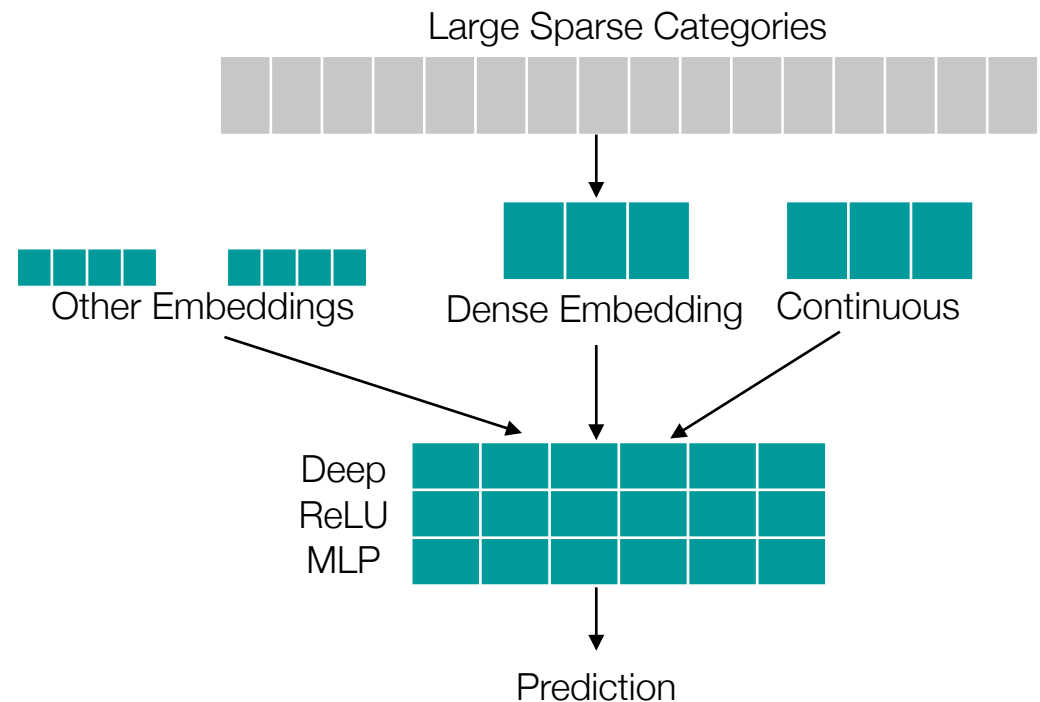


- Crossed feature “Rating-Genre”
  - G-Action, G-Drama, G-Comedy, G-Horror, G-else
  - PG-Action, PG-Drama, PG-Comedy, PG-Horror, G-else
  - and so on ... one hot encoded

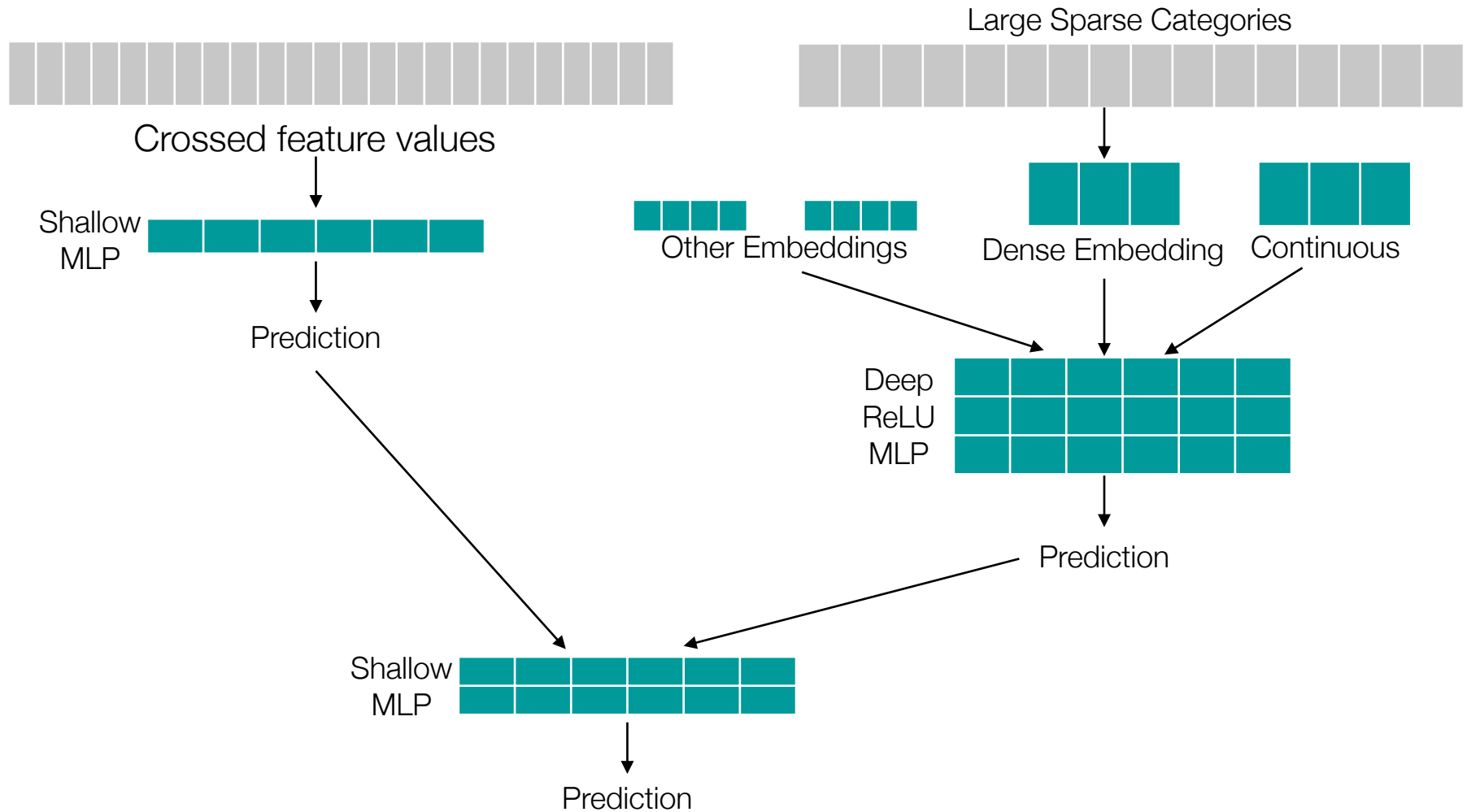
# Deep Features: What we have already done!

- Deep refers to increasingly narrow hidden layers
- Essentially the same as what we already did!

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



# Combining Memorization and Generalization



## 10. Keras Wide and Deep.ipynb

The awful dataset:  
Toy Census Data Example

Other tutorials:

[https://www.tensorflow.org/tutorials/wide\\_and\\_deep](https://www.tensorflow.org/tutorials/wide_and_deep)

