Neural Networks:
Autoencoders

F. Burkholder

Objectives

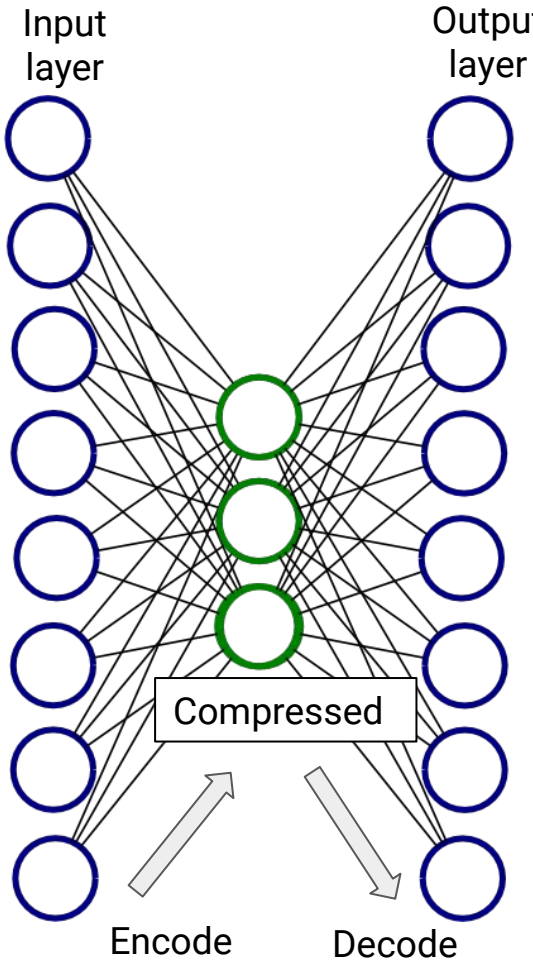
- Review multilayer perceptrons
- Be able to define an autoencoder
- List common use cases
- Show you what a denoising autoencoder looks like in Python
  - Tensorflow and Keras syntax
- Introduce the idea of an embedding
  - word2vec
- Public Service Announcement: Tensorflow callbacks
  - Tensorboard: a way to visualize the training of your Tensorflow model
  - Early stopping: why keep training when your loss has plateaued?

Review

- What's an epoch (in neural net training)?
- Where is the learning in neural networks stored?
- How do the weights get updated?
- Name some hyperparameters to tune in multilayer perceptrons.
- What activation function would you use in the final layer for:
  - a multi-class output
  - regression
  - a value representing a probability between 0 and 1
  - multiple values each with their own probability between 0 and 1
- What does supervised learning mean?
- What does unsupervised learning mean?
- What do you think "self-supervised" learning means?

Autoencoders

Original



Google's artificial brain learns to find cat videos.

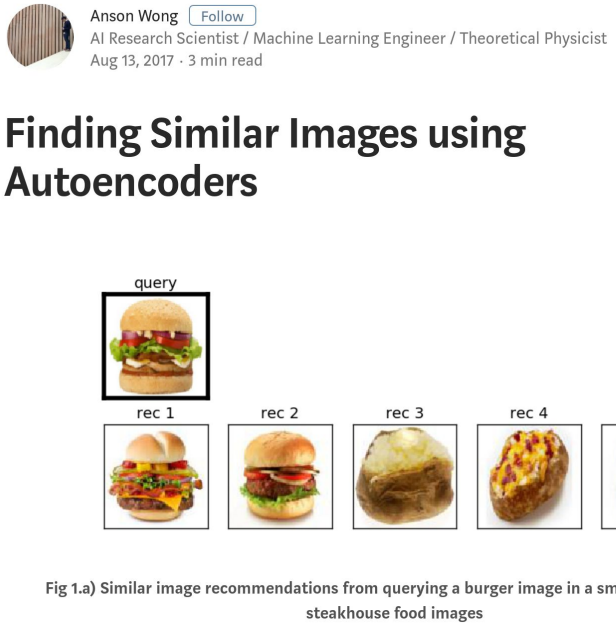
Reconstructed

arXiv abstract

Autoencoders

- ANN used to learn efficient data codings through an encoding/decoding process.
- Form of *self*-supervised learning
  - It's outputs are the inputs (or variations on the inputs)
- Autoencoders encode (embed) the input into a lower dimensional space
  - It's lossy!
- Can be used with many types of data: typical rows & columns, images, sequences.
- Common uses:
  - Reduction in dimensionality of data before sending to another algorithm
  - Denoising (noise added to input, output is input without noise) &
  - Recently, generative (learns parameters of distributions)
  - Face recognition
- Embeddings (similar idea, but not training on itself, exactly)

Use cases: reduce dimensionality then clustering



from [2]

Examples:

- [1] https://sefiks.com/2018/03/23/convolutional-autoencoder-clustering-images-with-neural-networks/
- [2] https://towardsdatascience.com/find-similar-images-using-autoencoders-315f374029ea

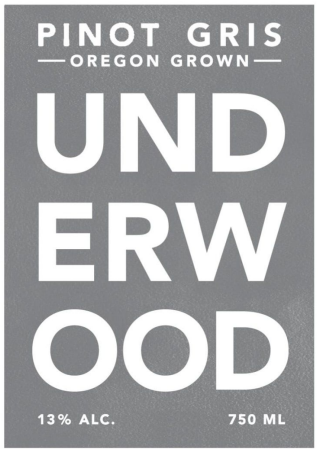
Use cases: reduce dim. -> feature(s) for content-based rec.

Wine Recommender

Project Objective

Do you pick your wine based on region? Price? Label? Varietal?

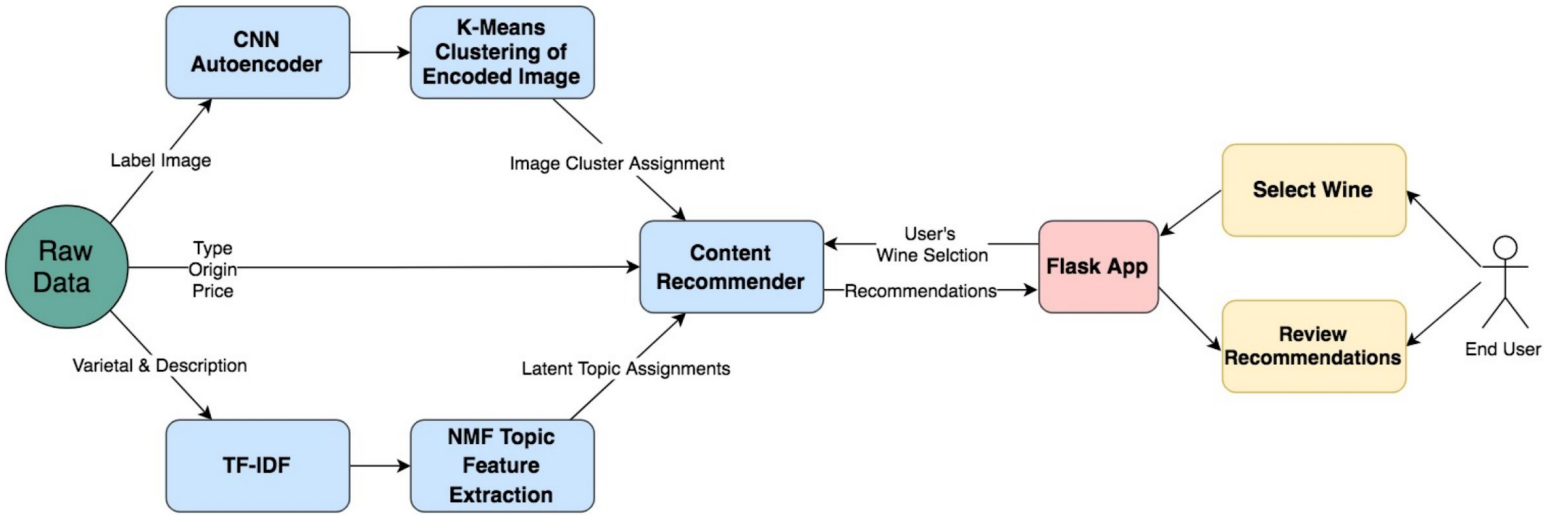
Image from The Daily Meal in Psychology of a Wine Label: Why We Buy the Bottles We Do



Like the label -> buy the wine….

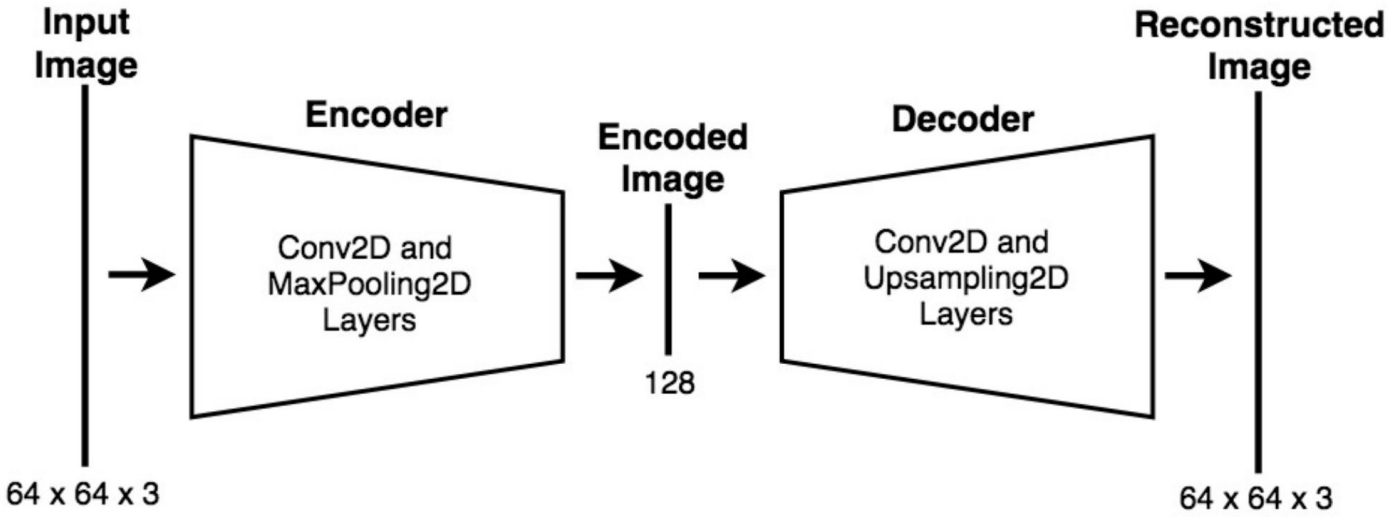
https://github.com/alyserecord/wine

Use cases: reduce dim. -> feature(s) for content-based rec.



https://github.com/alyserecord/wine

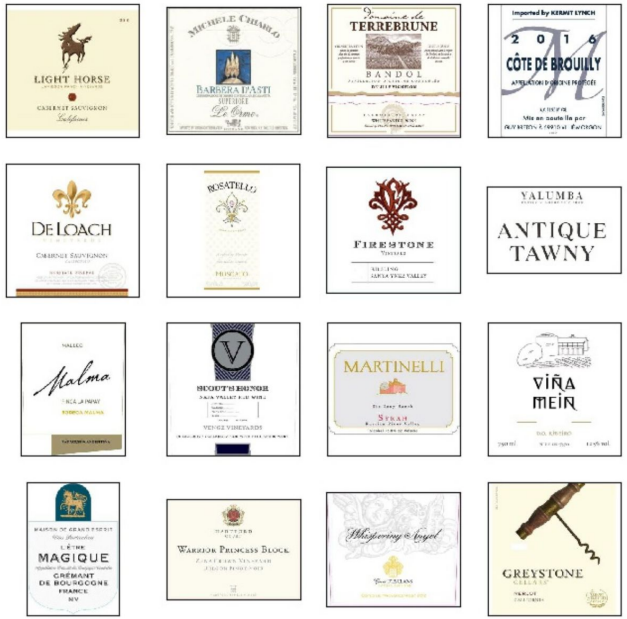
Use cases: reduce dim. -> feature(s) for content-based rec.



https://github.com/alyserecord/wine

Use cases: reduce dim. -> feature(s) for content-based rec.

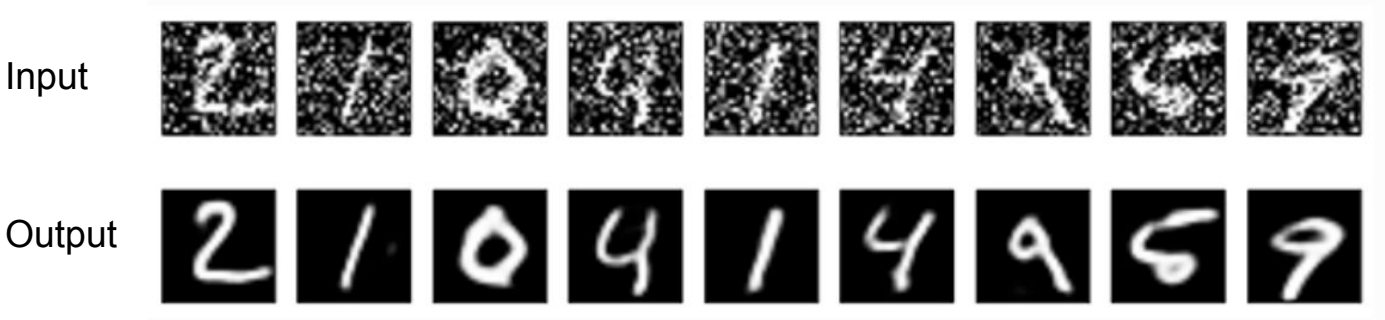
Cluster 3



Cluster 7

https://github.com/alyserecord/wine

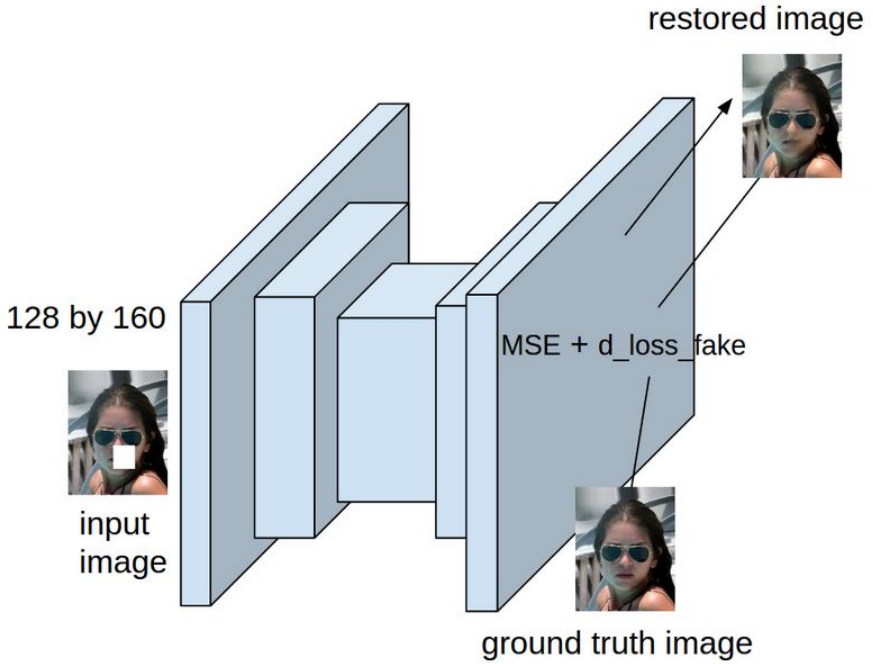
Use cases: reduce dim. then denoising



`mnist_denoising_autoencoder.ipynb`

Use cases: Generative Adversarial Denoising Autoencoder for Face Completion

Goal: Create a pipeline for photo restoration of portrait images.



link to video

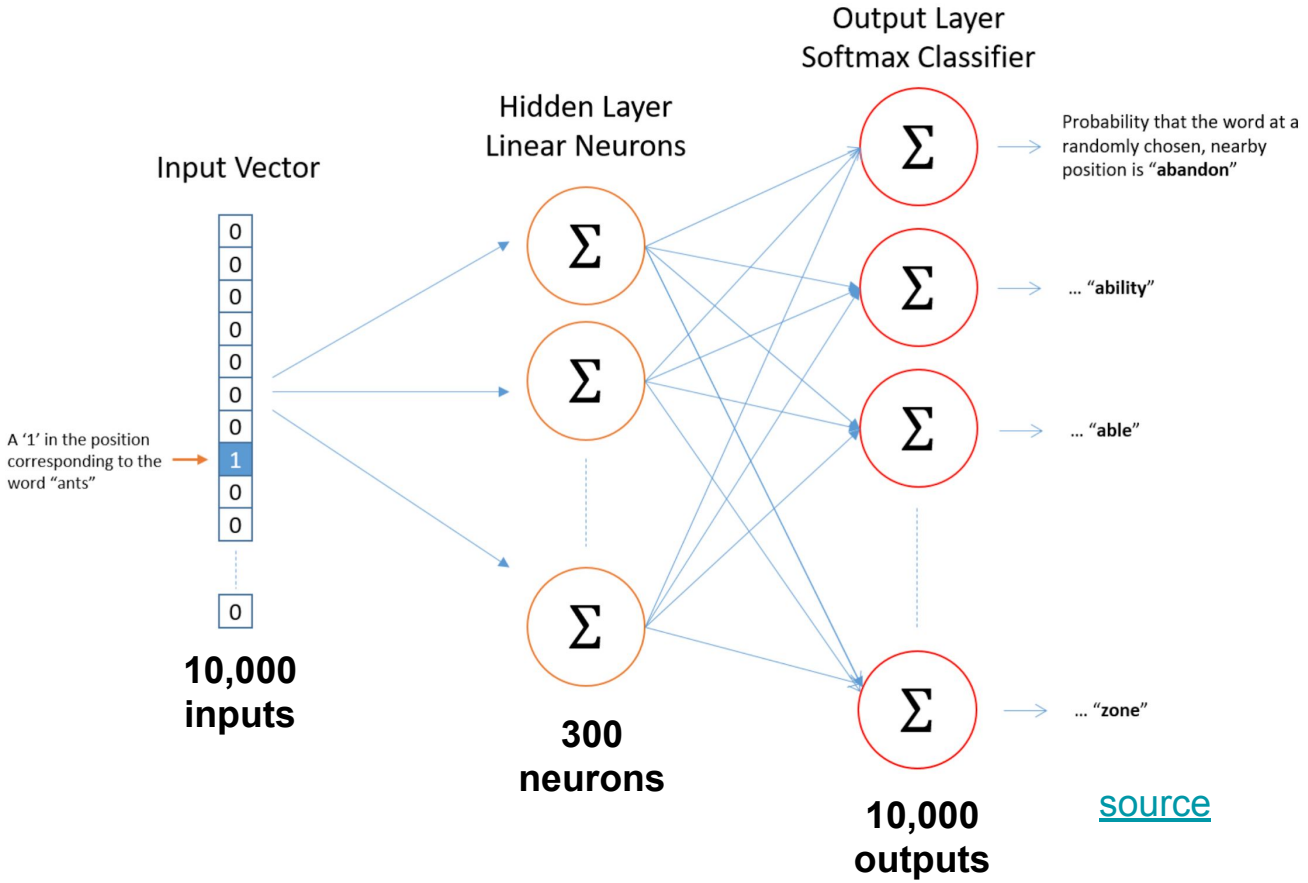
https://www.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen/
https://www.tensorflow.org/tutorials/generative/dcgan

Related topic to autoencoder - embeddings (Word2vec)

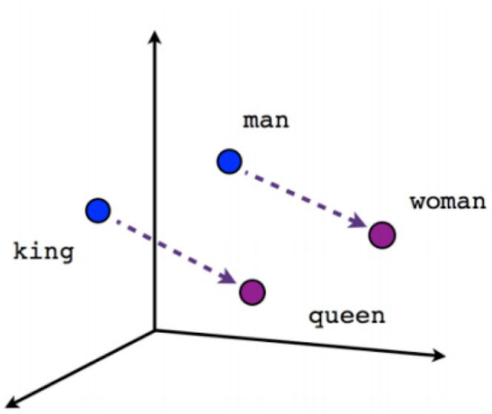
Word embedding

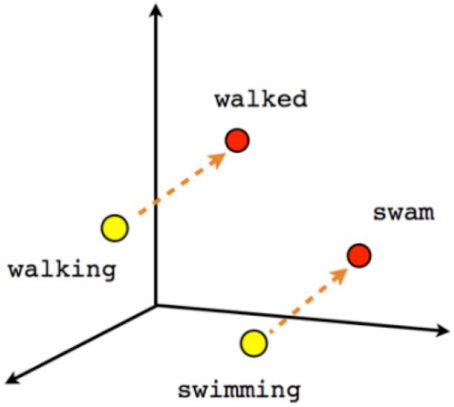
- where words or phrases from the vocabulary are mapped to vectors of real numbers.
- conceptually it involves a mathematical embedding from a space with many dimensions per word to a continuous vector space with a much lower dimension.

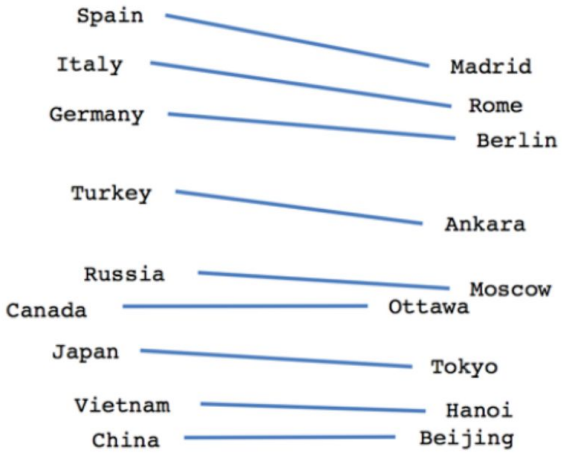
--Wikipedia



Related topic to autoencoder - embeddings (Word2vec)







Blog: [Clustering documents using Word2Vec]

source

PSA: Callbacks

Some useful Keras callbacks

A callback is a set of functions to be applied at given stages of the training procedure.
See full list at https://keras.io/callbacks/

To view your Tensorflow model after it has trained using Tensorboard

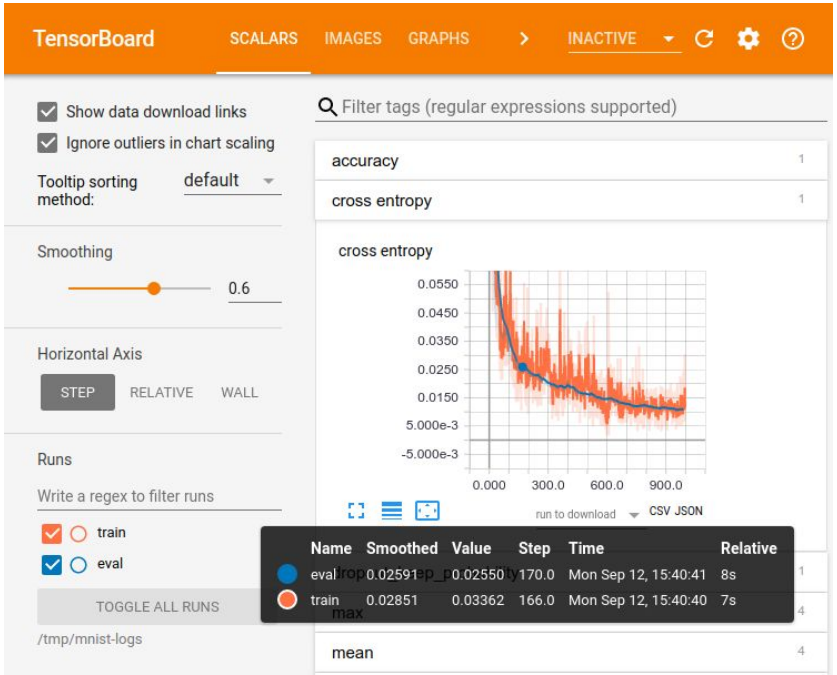
In your Python script:

```
from keras.callbacks import TensorBoard

...

tbCallBack = TensorBoard(log_dir='./logs', histogram_freq=0, write_graph=True, write_images=True)
model.fit(X_train, y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_split=0.2,
                        class_weight='auto', shuffle=True, callbacks=[tbCallBack])
```

After you have run your python script successfully, call Tensorboard from the terminal:

```
$ tensorboard --logdir ./logs
```

Cb: Tensorboard - use it to visualize loss after training



https://www.tensorflow.org/tensorboard/scalars_and_keras

Cb: EarlyStopping

If you want to stop at a convergence criteria:

In your Python script:

```
from keras.callbacks import EarlyStopping

...

earlyStopping = EarlyStopping(monitor='acc', min_delta=0.001, patience=0, verbose=0, mode='auto')
model.fit(X_train, y_train, batch_size=batch_size, epochs=nb_epoch, verbose=1, validation_split=0.2,
                        class_weight='auto', shuffle=True, callbacks=[earlyStopping])
```

Objectives

- Review multilayer perceptrons
- Be able to define an autoencoder
- List common use cases
- Show you what a denoising autoencoder looks like in Python
  - Tensorflow and Keras syntax
- Introduce the idea of an embedding
  - word2vec
- Public Service Announcement: Tensorflow callbacks
  - Tensorboard: a way to visualize the training of your Tensorflow model
  - Early stopping: why keep training when your loss has plateaued?