Advanced Institute for Artificial Intelligence

AutoEncoder

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Agenda

Agenda

- Aprendizagem não-supervisionada
- Redução de Dimensionalidade
- Autoencoders
- Autoencoders e Redes Convolucionais
- Autoencoders e Sequências
- Variotional Autoencoders

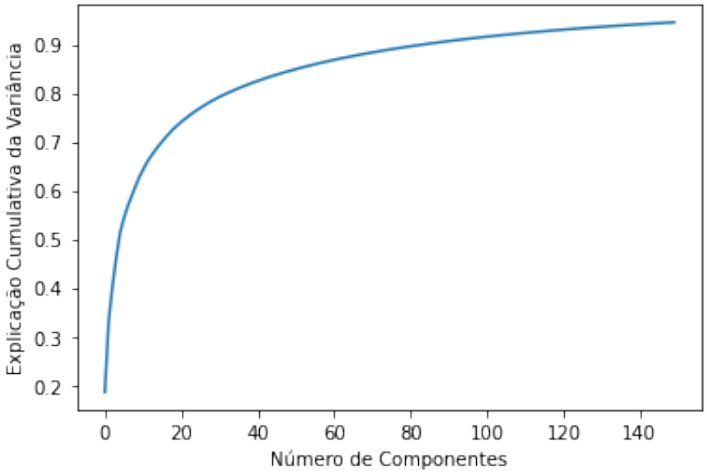
AutoEncoder

Aprendizagem não-supervisionada

- PCA - Principal Component Analysis
  - Monta uma representação de um domínio com um número menos de características
  - Em muitos domínios uma pequena parte das características é capaz de representar uma parte significativa da variância
  - Essa transformação é útil para manipular as características do problema

AutoEncoder

Uma forma de entender o poder do uso do PCA é avaliar a explicação da variância acumulada



AutoEncoder

Redução de Dimensionalidade

- Trabalhar com uma quantidade menor de informação
  - Melhoria de desempenho e uso de recursos
- Melhorar o desempenho do classificador
- A partir dos componentes gerados é possível gerar a entrada novamente
    - Em geral vai ocorrer perdas e imprecisões nesse processo

Reconhecimento Facial

- Um exemplo de aplicação de aprendizagem de máquina muito explorado é o reconhecimento facial
- A Base dados eigenfaces representa cada face usando 2914 pixel (42x67)
- Todos os pixels são representativos para identificar uma face unicamente?

AutoEncoder

Exemplo de Redução de dimensionalidade e depois retornando a representação original da base eigenfaces

AutoEncoder

Redes autocodificadoras (autoencoders)

- Aprendizagem não-supervisionada
- Representacao concisa da entrada
  - Similar ao modelo PCA
  - Representacao permite reconstruir entrada
- Características encontradas podem ser usadas para posterior aprendizagem supervisionada
- A idéia do Autoencoder é reproduzir na saída a própria entrada

AutoEncoder

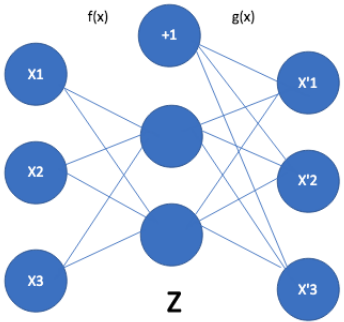
Ao reduzir a dimensionalidade da entrada é possível reconstruir a entrada com pouca quantidade de informação

- Alguns exemplos de problemas que podem ser investigados com autoencoders
  - Extração de características
  - Detecção de Outliers
  - Agrupamento
  - Compressão de dados
  - Recuperação de Informação faltante
    - Eliminar ruídos
    - Reconstruir parte faltante de uma sequência

AutoEncoder

- AutoEncoder não assume a premissa de linearidade
- AutoEncoder sem função de ativcação se comporta como PCA
- Transformacoes sao aplicadas na entrada de acordo com dois tipos de funcoes
  - Funcao de extracao de caracteristicas (encoder) mapeia o conjunto de treinamento para uma representacao latente.
  - Funcao de reconstrucao (decoder) mapeia a representacao do espaço latente de volta ao espaço original

AutoEncoder



X representa a entrada
f(x)=Z representa a transformação da entrada para espaço Latente
g(z)=x transforma do espaço latente para entrada novamente
O objetivo é aprender as duas funções minimizando o erro de reconstrução
A entrada e o valor alvo são os mesmos

AutoEncoder

Espaço Latente subcompleto e sobrecompleto

- Quando o tamanho do espaço latente é menor que a entrada é chamado de subcompleto
- Quando o tamanho do espaço latente é menor que a entrada é chamado de sobrecompleto

AutoEncoder

Classificação quanto ao tamanho do espaço latente

- Espaço subcompleto
  - Adequado para compressáo
  - Identificação de características relevantes na entrada (filtro de características
- Espaço sobrecompleto
  - Interpolação simples do espaço de busca
  - Dificuldade em aprender características importantes da entrada

AutoEncoder

Uma forma simples de criar um Autoencoder é usar dois modelos sequential combinados
Modelo encoder

```
1
2  latent_dim = 100
3
4  input_shape=784
5
6  encoder = Sequential(name="encoder")
7  encoder.add(Dense(128, input_shape=(input_shape,), activation="relu"))
8  encoder.add(Dense(latent_dim)) # Vetor Latente
```

AutoEncoder

Modelo decoder

```
1
2 decoder =  Sequential(name="decoder")
3 decoder.add(Dense(128, activation="relu", input_shape=(latent_dim,) ))
4 decoder.add(Dense(input_shape, activation="sigmoid"))
```

AutoEncoder

Combinando encoder e decoder

Modelo com base na entrada combina encoder e decoder

```
1
2  input_img = Input(shape=(input_shape,) )
3  z = encoder(input_img)
4  recons = decoder(z)
5  ae = Model(input_img, recons)
```