Data Science Academy

www.datascienceacademy.com.br

Deep Learning I

Singular Value Decomposition

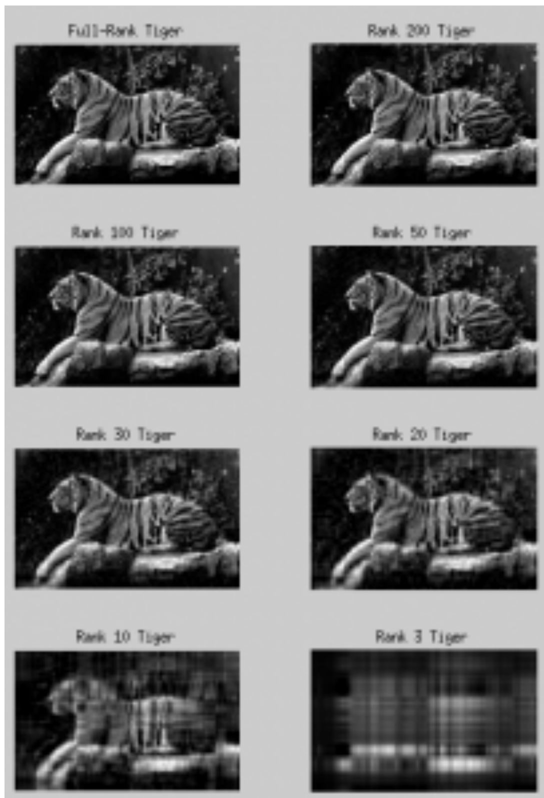
Um dos resultados mais bonitos e úteis da álgebra linear é uma decomposição de matriz conhecida como a Decomposição do Valor Singular (ou Singular Value Decomposition). Vamos examinar a teoria por trás dessa decomposição de matriz e mostrar alguns exemplos de porque o SVD é uma das ferramentas matemáticas mais úteis, principalmente no tratamento de imagens. Em anexo você encontra um exemplo de aplicação do SVD com Numpy, para compressão e imagens.

Grande parte da álgebra linear é sobre operadores lineares, ou seja, transformações lineares de um espaço. Um resultado típico é que, escolhendo uma base adequada para o espaço, o operador pode ser expresso de forma simples, por exemplo, diagonal. No entanto, isso não se aplica a todos os operadores.

A decomposição do valor singular é o único resultado principal sobre as transformações lineares entre dois espaços diferentes. Ao escolher bases adequadas para os espaços, a transformação pode ser expressa em uma matriz simples, uma matriz diagonal. E isso funciona para todas as transformações lineares. Além disso, as bases são muito agradáveis: bases ortogonais.

O SVD é usado para remover os recursos redundantes em um conjunto de dados. Suponha que você tenha um conjunto de dados composto por 1000 recursos. Definitivamente, qualquer conjunto de dados reais com uma grande quantidade de recursos vai conter recursos redundantes. Os recursos redundantes causam muitos problemas na execução de algoritmos de aprendizado da máquina. Além disso, executar um algoritmo no conjunto de dados original será ineficiente em termos de tempo de processamento e exigirá muita memória. Então, como resolvemos esse problema? Nós temos uma escolha? Podemos omitir alguns recursos? Isso levará a uma quantidade significativa de perda de informação? Poderemos obter um modelo suficientemente eficiente mesmo depois de omitir as linhas? Vou responder a essas perguntas com a ajuda de uma ilustração.

Podemos converter este tigre em preto e branco e podemos pensar nisso como uma matriz cujos elementos representam a intensidade do pixel como local relevante. Em palavras mais simples, a matriz contém informações sobre a intensidade dos pixels da imagem na forma de linhas e colunas. Mas, é necessário ter todas as colunas na matriz de intensidade? Poderemos representar o tigre com uma quantidade menor de informações? A próxima imagem esclarecerá o meu ponto. Abaixo, imagens diferentes são mostradas correspondentes a diferentes classificações com resolução diferente. Por enquanto, basta assumir que uma classificação mais alta implica a maior quantidade de informações sobre a intensidade de pixels.



É claro que podemos alcançar uma imagem muito boa com 20 ou 30 fileiras em vez de 100 ou 200 fileiras e é isso que queremos fazer em um caso de dados altamente redundantes. O que eu quero transmitir é que, para obter uma hipótese razoável, não precisamos reter todas as informações presentes no conjunto de dados original. Mesmo alguns dos recursos causam um problema ao alcançar uma solução para o melhor modelo. Para o exemplo, a presença de recursos redundantes provoca multi-linearidade na regressão linear. Além disso, alguns recursos não são significativos para o nosso modelo. Omitindo esses recursos ajuda a encontrar um ajuste melhor do modelo, juntamente com a eficiência do tempo e menor espaço em disco. A decomposição do valor singular é usada para eliminar os recursos redundantes presentes em nossos dados.

Diagonalização

Vamos passar rapidamente pela diagonalização. Uma matriz A é "diagonalizável" se pudermos reescrevê-la (decompor) como um produto:

$$A = PDP^{-1}$$

Onde P é uma matriz reversível (e, portanto, P^{-1} existe) e D é uma matriz diagonal (onde todos os elementos fora da diagonal são iguais a zero).

Apenas a partir desta definição, podemos deduzir algumas coisas importantes sobre a diagonalização. Antes de tudo, uma vez que o P é reversível, deve ser quadrado. Portanto, essa definição realmente só faz sentido para matrizes quadradas. Uma matriz que não é quadrada não é diagonalizável, simplesmente porque o conceito de diagonalização não faz sentido para matrizes não-quadradas.

Em certo sentido, a decomposição do valor singular é essencialmente a diagonalização em um sentido mais geral. A decomposição do valor singular desempenha um papel semelhante à diagonalização. Ou seja, o SVD aplica-se a matrizes de qualquer forma. Não só isso, mas o SVD aplica-se a todas as matrizes, o que torna muito mais aplicável e útil que a diagonalização!

Singular Value Decomposition

O SVD é uma decomposição muito útil. Gostaria de fornecer rapidamente alguns exemplos, apenas para mostrar um pequeno vislumbre de como isso pode ser usado em ciência da computação, matemática e outras disciplinas.

Uma aplicação do SVD é a compressão de dados. Considere alguma matriz A com o rank de 500. Ou seja, as colunas desta matriz abrangem um espaço de 500 dimensões. Codificar esta matriz em um computador vai demorar bastante e requerer muita memória! Podemos estar interessados em aproximar esta matriz com uma de menor classificação - quão perto podemos chegar a essa matriz se apenas a aproximarmos como uma matriz com classificação cem, de modo que só precisamos armazenar centenas de colunas? E se usarmos uma matriz de rank vinte? Podemos resumir toda a informação da matriz de rank 500 com apenas uma matriz de 20? Sim, podemos. Usando SVD.

Para uma explicação matemática completa, acesse o link abaixo no blog de Andrew Gibiansky: http://andrew.gibiansky.com/blog/mathematics/cool-linear-algebra-singular-value-decomposition/