

Guiliano Pinheiro Batista 173096

01

a) As GANs são compostas por duas redes que desempenham funções complementares no processo de geração de padrões sintéticos fidêis. Enquanto a rede geradora cria dados sintéticos a partir de um espaço latente, a rede discriminadora tenta descobrir se os padrões recebidos por ela são sintéticos, ou não. As duas redes são consideradas adversárias porque o aumento de desempenho de uma rede, mantendo a outra inalterada, acarreta uma perda de desempenho da outra. A rede geradora tenta criar padrões sintéticos cada vez mais verossímeis, enquanto a rede discriminadora busca classificá-los de maneira cada vez melhor se os padrões são sintéticos, ou não.

b) A função objetivo:

$$V = \min_{G} \max_{D} \mathbb{E}_{x \sim \text{dados}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

→ G representa a rede geradora e D a discriminadora. Primeiramente o objetivo é maximizar o desempenho de D(.)

D é um classificador binário: 1 se o padrão não for sintético e 0 se for. Em uma primeira etapa um batch de amostras reais e sintéticas é apresentado à rede e os parâmetros de D(.) são ajustados maximizando V.

Em seguida, apenas amostras sintéticas são apresentadas à rede. Como o objetivo de G(.) é enganar D(.), os parâmetros de G(.) serão ajustados de modo a minimizar V

D S T Q Q S S
D L M M J V S

Como somente amostras sintéticas serão apresentadas, isso equivale a minimizar somente o segundo termo de V . É necessário haver um equilíbrio entre o progresso de aprendizado das duas regras. Caso isso não ocorra o aprendizado de uma das regras pode ser prejudicado.

→ É interessante pontuar que G-1 aprende as geras padrões sintéticos fidelígnos sem a necessidade de ser apresentada às amostras reais. Em vez disso ocorre a seleção de características apropriadas do espaço latente ao longo do processo de treinamento.

Luciano Pinheiro Batista 173096
02

- Mapa de características é o que se obtém após a convolução entre os dados de entrada e um filtro. Trata-se de uma nova imagem que realça as áreas da imagem de entrada que mais foram ativadas pelo filtro. N filtros geram N mapas.
- Neurônios, no contexto de CNNs, são os "pixels" que compõe os mapas de características.
- Campo receptor de um neurônio é o subconjunto dos pixels da imagem de entrada que irão influenciar um determinado neurônio de um mapa de características.
- Nas CNNs os parâmetros de cada mapa de características são compartilhados, isto é, são iguais para todos os neurônios de um mapa, pois dependem do mesmo filtro. Isto contribui para uma redução do número de parâmetros treináveis em comparação com camadas totalmente conectadas, por exemplo.

Guiliano Pinheiro Batista 173096

03

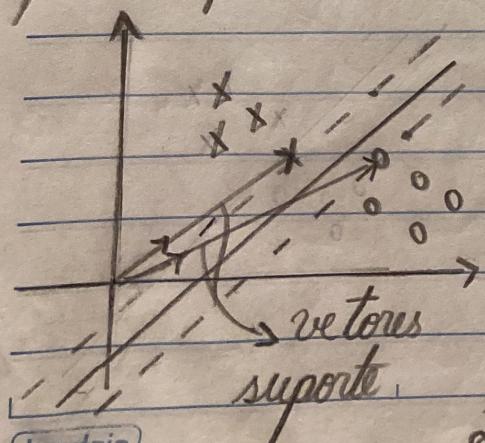
a)

$$\max L(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j d_i d_j \underbrace{\phi(x_i)^T \phi(x_j)}_{K(x_i, x_j)}$$

Sa $\sum \lambda_i d_i = 0$ e $\forall i \in \{1, \dots, N\}$, em que ϕ é um mapeamento, que pode ser não linear, dos padrões X num espaço de maior dimensão, possibilitando uma classificação não linear no espaço original, que separe melhor os dados.

b) O truque do Kernel consiste em substituir o produto escalar $\phi(x_i)^T \phi(x_j)$ na expressão de $L(\lambda)$ pelo Kernel $K(x_i, x_j)$, evitando a necessidade de realizar o mapeamento ϕ em todos os padrões X . Um Kernel é uma função capaz de computar o produto escalar $\phi(x_i)^T \phi(x_j)$ utilizando apenas os vetores originais x_i, x_j . Esse truque contribui para a eficiência computacional do processo. Os Kernels utilizados permitem realizar uma classificação não linear no espaço original, a depender do mapeamento escolhido.

c) Os vetores suporte são os vetores que são definidos pelos pontos que estão situados nas margens do classificador. Esses pontos são os únicos que desempenham algum papel na definição dos pesos ótimos do classificador. Na solução do problema dual:



$$\lambda_i [d_i (w_0^T x_i + b_0) - 1] = 0, i = 1, \dots, N$$

↳ $\lambda_i \neq 0$ se x_i estiver situado na margem

$$\frac{\partial L}{\partial w} = w - \sum \lambda_i d_i x_i = 0 \rightarrow w_0 = \sum \lambda_i d_i x_i$$

$\hookrightarrow w_{\text{ótimo}}$

Guiliano Pinheiro Batista 173096

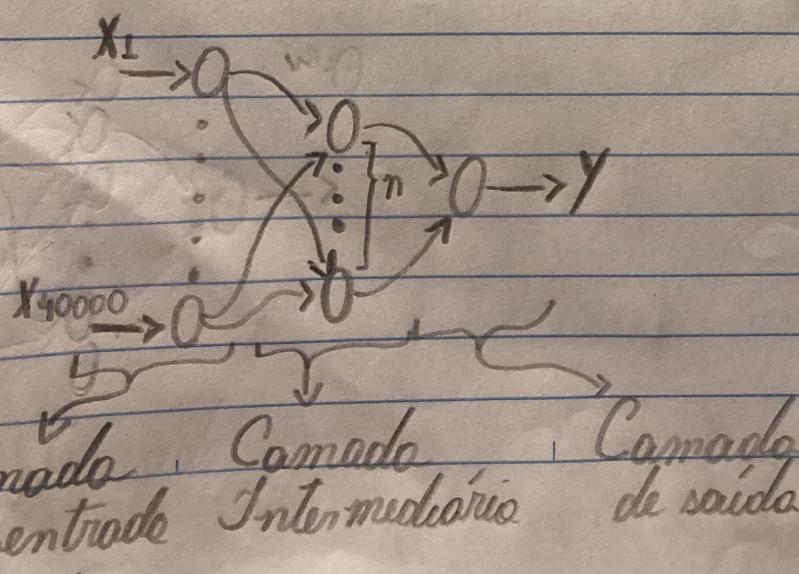
04

O primeiro passo é separar as imagens em três conjuntos: treinamento, validação e teste. Uma divisão coerente poderia ser 700 imagens para treinamento, 300 para validação e 500 para teste.

Após isso é conveniente, por motivos computacionais, normalizar os dados para que todos tenham um valor entre 0 e 1. Isto pode ser feito dividindo-os pelo valor máximo presente nas amostras, no caso de imagens comumente 255.

Para que os dados estejam adequados para servirem como entrada para uma MLP, é preciso que as imagens, que são matrizes de dimensão 200×200 , tenham os seus elementos empilhados em vetores de tamanho $200^2 = 40000$.

Em seguida é preciso definir alguns aspectos da rede MLP que será utilizada. Como o número de camadas intermédias está fixo, as características mais relevantes a serem determinadas são o número de neurônios presentes na camada e a função de ativação a ser utilizada. É preciso lembrar que, em uma MLP os neurônios de uma camada estão conectados a todos os neurônios da camada seguinte. Sendo assim a estrutura da rede seria assim:



Em que cada "O" consiste em um neurônio do tipo perceptron. A saída da rede é a classe a qual imagem de entrada deve pertencer - 0, 1, 2, ou 3 -. A saída de cada neurônio da camada intermediária é dada por:

$$y_i = \phi\left(\sum_{j=0}^{n-1} w_j x_j\right) \quad p/i = 1, \dots, n$$

em que ϕ é a função de ativação - por exemplo ReLU, tanh - e n é o número de neurônios da camada intermediária. Essas duas características serão hiperparâmetros que serão variados na etapa de treinamento a fim de se obter o melhor modelo possível na etapa de validação.

No etapa de treinamento as imagens deste conjunto são fornecidas à rede e com base nas suas classes os parâmetros da rede neural são ajustados. Isso é feito utilizando algum método de treinamento, como o gradiente descendente! Ele objetivo minimizar uma função custo que expressa uma medida do erro entre as saídas fornecidas pela rede e as classes das imagens. Para isso o gradiente da função pode ser calculado com o auxílio da técnica de retropropagação do erro, e um "passo" no vetor de parâmetros é dado na direção contrária a do gradiente. Esse processo é feito de maneira iterativa, apresentando um subconjunto dos padrões, em cada iteração. Quando todos os padrões são apresentados diz-se que passou uma época. O número de épocas e o tamanho do passo são escolhas do projetista da rede.

No etapa de validação as imagens deste conjunto são fornecidas à rede utilizando parâmetros obtidos na etapa anterior. A melhor configuração da rede é escolhida com base em alguma métrica da qualidade de aproximação obtida, como

Por exemplo a acurácia da rede. É importante que a rede possua uma boa capacidade de generalização, ou seja, que classifique bem imagens que não foram utilizadas na etapa de treinamento. Assim, para evitar o sobreajuste, pode ser usada alguma técnica de regularização como early stopping.

Na etapa de teste o melhor modelo, obtido após o ajuste e a validação, pode ser posto à prova com as imagens deste conjunto.

Luciano Pinheiro Batista 173096

05

Muito provavelmente houve sobreajuste, isto é, a classificação gerada pelo modelo se "contorceu" de forma excessiva a fim de reduzir ao máximo o erro junto aos dados de treinamento. Quando nenhuma amostra é inédita, no entanto, o modelo comete erros significativos. Para contornar esta baixa capacidade de generalização é possível recorrer a técnicas como:

1) Dropout - Consiste em atribuir aos neurônios uma probabilidade " p " de serem desligados a cada passo de treinamento. A cada passo a rede atuará com uma arquitetura distinta, mas com pesos em comum. Assim cada neurônio será estimulado a desempenhar um papel útil para a rede, diminuindo a ocorrência de co-adaptações complexas e aumentando a sua capacidade de generalização.

2) Early Stopping - Consiste em interromper o treinamento quando o erro de validação começar a crescer de forma sistemática. O conjunto de validação pode ser um subconjunto dos dados disponibilizados pela competição e que não serão usados para o treinamento. Uma estratégia é interromper o treinamento quando o valor do erro de validação aumentar por ϵ iterações sucessivas.

1) e 2) se adequam bem ao contexto de CNNs. 1) inclusive foi adotado pelos criadores da alexnet.

Kleiciano Pinheiro Batista 173096

06

a) $\lambda = \text{eig}(E\{XX^T\}) \in \mathbb{R}^{8 \times 1}$

logo $XX^T \in \mathbb{R}^{8 \times 8}$, assim $X \in \mathbb{R}^{8 \times 1}$

b) Para encontrar a resposta é preciso calcular a porcentagem de preservação da variância original para cada número de componentes principais.

$$\frac{V_M}{V_m} = \frac{\sum_{u=1}^M E\{y_u^2\}}{\sum_{u=1}^K E\{y_K^2\}} = \frac{\sum_{u=1}^M \lambda_u}{\sum_{u=1}^K \lambda_u} = \frac{\sum_{u=1}^M \lambda_u}{8}$$

M é o número
de componentes principais
selecionados

N é a dimensão
original dos dados

y_u são as projeções
obtidas para
cada componente
principal

$$Y = W^T X$$

↑
 $E\{Y\} = E\{W^T X\}$
↑
 $E\{X\}$

$[w_1 \dots w_N]^T$

colunas são os
autovetores da matriz
de autocorrelação de
dados

Como os componentes principais seguem uma ordem decrescente em relação aos autovalores da matriz de autocorrelação, temos:

Nº de componentes principais exibidos	% da variância original
1	$3/18 = 37,5\%$
2	$(3+2,3)/18 = 66,25\%$
3	$(3+2,3+1,5)/18 = 85\%$
4	$(3+2,3+1,5+0,5)/18 = 91,25\%$

Logo p/ se obter uma preservação de ao menos 90%, é preciso utilizar no mínimo 4 componentes principais.

Guiliano Dinheiro Batista 173096

07

→ Fuzzy K-means incorpora o conceito de pertinência, permitindo que cada padrão possua um nível de pertinência gradual em relação a cada cluster. Isso contrasta com a estratégia K-means convencional em que o padrão pertence somente e totalmente a um cluster. Isto pode ser um impedimento em certas situações do mundo real. Por exemplo, qual o limite de altura para considerar uma pessoa alta ou baixa? Neste caso pode ser interessante se valer das ideias fuzzy, permitindo que cada pessoa pertença gradualmente ao cluster alto, ou baixo.

→ Os mapas auto-organizáveis de Kohonen são capazes de representar um conjunto de dados com um grande número de dimensões em um conjunto de dimensão menor, preservando porém a estrutura topológica dos dados. Cada padrão apresentado ao mapa provoca um ajuste dos pesos dos neurônios mais próximos. Este ajuste é feito justamente na direção do padrão que está sendo apresentado. A distância entre os agrupamentos pode ser feita com uma matriz-U. Esta estratégia pode facilitar a visualização e análise de dados com um número grande de dimensões. O K-means não realiza essa redução dimensional.

Luciano Pinheiro Batista 173096

08

a) TP → O que é da classe Glaucoma e foi classificado como desta classe. $TP = 100$

• FP → O que não é da classe Glaucoma, mas foi classificado como desta classe. F

$$\hookrightarrow FP = 15 + 25 = 40$$

• TN → O que não é da classe Glaucoma e não foi classificado como desta classe

$$\hookrightarrow TN = 410 + 25 + 55 + 100 = 590$$

• FN → O que é da classe Glaucoma, mas não foi classificado como desta classe.

$$\hookrightarrow FN = 10 + 40 = 50$$

b) No cenário multi classe a curva balançada é a média do recall obtido para cada classe. É uma métrica competente em cenários com um significativo desbalanceamento entre classes.

	TP	FP	TN	FN	BA
Normal	410	90			$\left(\frac{410}{410+90} \right) + \left(\frac{100}{100+50} \right) + \left(\frac{100}{100+80} \right)$
Glaucoma	100	40	590	50	3
Catarata	100		80		

$BA = \frac{410 + 100 + 100}{410 + 90 + 100 + 50 + 80} \approx 71,11\%$

$recall = \frac{TP}{TP+FN}$