

Iniciado em quinta, 29 out 2020, 19:01

Estado Finalizada

Concluída em quinta, 29 out 2020, 19:39

Tempo empregado 37 minutos 49 segundos

Questão **1**

Completo

Vale 2,00 ponto(s).

Autoencoders do tipo Overcomplete possuem dimensão latente superior a da entrada. Como garantir o aprendizado sem que haja uma cópia simples dos dados?

Escolha uma:

- ☒ a. Impondo uma restrição na projeção do código que penalize o uso de todas as dimensões do espaço latente e privilegie projeções esparsas para uma dada instância.
- ☐ b. Utilizando normalização do tipo Batch ou Layer para que os dados sejam modificados ao longo da rede neural, assim evitando que haja uma cópia direta da entrada para o código
- ☐ c. Substituindo a função de custo de perda ou erro quadrático pela função de custo de entropia cruzada
- ☐ d. Utilizando mais camadas, projetando um Autoencoder Overcomplete Profundo que permita obter uma camada latente com maior qualidade

Questão **2**

Completo

Vale 2,00 ponto(s).

Considere uma arquitetura com dois elementos, organizada na forma encoder-decoder. O primeiro elemento, encoder, aprende a gerar uma camada latente com os parâmetros de distribuições de probabilidade, a partir da qual o segundo componente, decoder amostra exemplos a serem reconstruídos. Esse método é conhecido por:

Escolha uma:

- ☒ a. Variational Autoencoder
- ☐ b. Continuous Generalized Autoencoder
- ☐ c. Generative Adversarial Network
- ☐ d. Denoising Overcomplete Autoencoder

Questão **3**

Completo

Vale 2,00 ponto(s).

Nas redes geradoras adversariais, todos os dados de treinamento pertencem a classe positiva, enquanto que todos os dados gerados a partir de uma amostragem aleatória de uma distribuição pertence à classe negativa. É correto afirmar sobre o método de aprendizado de redes geradoras adversariais (em sua formulação original):

Escolha uma:

- ☐ a. O modelo discriminador produz a perda utilizada para treinar a rede como um todo. O aprendizado é semi-supervisionado já que sabemos o rótulo das classes positivas.
- ☐ b. O modelo discriminador é um classificador, que produz uma única perda utilizada para treinar a rede como um todo. Sendo um classificador o responsável pelo ajuste dos parâmetros, consideramos o aprendizado como supervisionado.
- ☒ c. O modelo discriminador (classificador) produz as perdas utilizadas para treinar a rede em associação com o gerador. Os rótulos são calculados não necessitando de anotação manual, sendo portanto não supervisionado.
- ☐ d. O modelo gerador produz a perda utilizada para treinar a rede como um todo. O aprendizado é não-supervisionado pelo fato de que os dados do gerador são obtidos a partir de um vetor amostrado a partir de uma distribuição aleatória.

Questão **4**

Completo

Vale 2,00 ponto(s).

Carregue a base de dados `smarphone_activity_dataset.csv`, conforme abaixo, com uma divisão hold-out utilizando os 80% exemplos iniciais para treinamento e os restantes para teste.

Projete um Autoencoder para produzir uma projeção em 2 dimensões, com as seguintes camadas:

- Entrada (com as dimensões da base de dados)
- Dropout de 0.3
- Camada densa de 2 neurônios (código) e ativação linear
- Camada densa de saída (com as dimensões da base de dados) e ativação tangente hiperbólica

Inicialize as sementes `seed(1)` e `set_seed(2)` antes de instanciar o modelo, compilar e treinar.

Utilize a função de custo mean absolute error (mae), otimizador Adam com taxa 0.01, batchsize 16 e treine por 30 épocas.

Após o treinamento, obtenha um scatterplot do código de treinamento, e analise visualmente a distribuição das classes utilizando os 500 primeiros exemplos :500.

Podemos identificar grupos com quais misturas de classes?

Escolha uma:

- ☒ a. 3 grupos, classes: (1 e 3); (2); (4, 5 e 6)
- ☐ b. 4 grupos, classes: (1 e 2); (3); (4); (5 e 6)
- ☐ c. 4 grupos, classes: (1 e 3); (2); (4 e 5); (6)
- ☐ d. 2 grupos, classes: (1, 3 e 6); (2, 4 e 5)

Ainda utilizando a base de dados `smartphone_activity_dataset.csv` com a mesma divisão entre treinamento e teste, projete um Autoencoder profundo para produzir uma projeção em 50 dimensões, com as seguintes camadas:

- Entrada (com as dimensões da base de dados)
- 2 Camadas densas com 128 neurônios, ativação tanh
- Dropout de 0.25
- 1 Camada densa com 50 neurônios, ativação tahn, name='code'
- 2 Camadas densas com 128 neurônios, ativação tanh
- Camada densa de saída (com as dimensões da base de dados), ativação tanh

Inicialize as sementes `seed(1)` e `set_seed(2)` antes de instanciar o modelo, compilar e treinar.

Utilize a função de custo `mean squared error (mse)`, otimizador Adam com taxa 0.001, `batchsize` 16 e treine por 50 épocas.

Após o treinamento, calcule o valor final da função de custo para os dados de treinamento.

Obtenha o código de 50 dimensões para o conjunto de treinamento e de teste. Treine um classificador svm (utilizando o pacote SVC do sklearn), utilizando parâmetro `C=0.5` e `random_state=1` , utilizando o código de treinamento, e calcule a acurácia no código do conjunto de teste.

Os valores observados de MSE de treinamento, e acurácia de classificação SVM no teste estão no intervalo:

Escolha uma:

- ☒ a. MSE =[0.00, 0.04]; Acurácia = [0.88, 0.95]
- ☐ b. MSE =[0.10, 0.25]; Acurácia = [0.70, 0.75]
- ☐ c. MSE =[0.10, 0.25]; Acurácia = [0.75, 0.77]
- ☐ d. MSE =[0.01, 0.05]; Acurácia = [0.80, 0.86]

◀ Apoio exercícios - `smartphone_activity_dataset.csv`

Seguir para...



Avaliação ▶