Assignment 5: Deep k-Means: Jointly clustering with k-Means and learning representations

Juan Belieni e Juliana Carvalho

17 de Novembro de 2023

1 Conceitos e implementação

Esse assignment implementa o Deep K-means [1], que adapta o k-means para adicionar a clusterização conjunta (jointly clustering) e o aprendizado de representações.

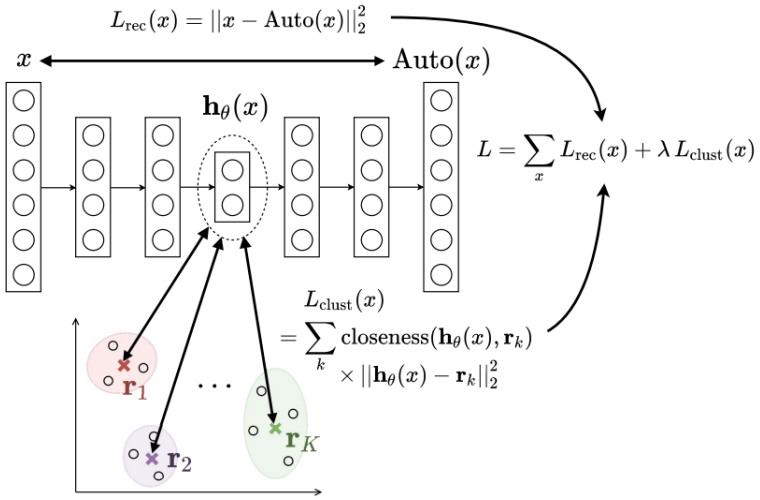

Figura 1: Visão geral do Deep K-means *Deep k-Means: Jointly clustering with k-Means and learning representations*, onde as funções de perda são baseadas na distância gaussiana.

O objetivo do algoritmo é aprender representações dos dados que sejam fiéis aos dados originais e adaptadas ao algoritmo de clusterização. O algoritmo onsiste em reparametrizar a função objetivo do k-Means de forma contínua, permitindo que se atualize os parâmetros da rede neural e os centróides dos grupos usando apenas gradientes.

Neste assignment implementamos o **DKMa**, que se utiliza a estratégia de **Annealing** para os valores de α e o **DKMp** que utiliza a estratégia de preteino.

Especificamente, o **DKMa** é independete do pretreino e os valores de α gerados são evoluidos da seguinte forma recursiva:

$\alpha_{n+1} = 2^{1/log(n)^2}\alpha_n$ com $m_\alpha = \alpha_1 = 0.1$

A variante **DKMp** é inicializada primeiro pré-treinando um autoencoder e, em seguida, aplicando a abordagem de clusterização conjunta com uma constante α tal que $m_\alpha = M_\alpha$.

1.1 Implementação

O notebook está disponível no Colab. Nesse código, introduzimos funções auxiliares, a saber, `get_alphas`, que fornece os valores possíveis de α e `train_pipeline`, que compila e treina os modelos de Deep k-means.

2 Dados

Os dados se tratam do MNIST, um dataset de imagens contendo 70000 imagens de dígitos manuscritos de 28×28 pixels divididos em 10 classes.

3 Experimentos

Nos experimentos, em relação aos hiperparâmetros utilizamos o Adam como otimizador, com uma learning rate de 0.001, `batch_size=256` no DataGenerator.

4 Resultados

4.1 Task 1

Tabela com acurácias:

Configuration	**α-Pretrained**	**α-Annealing**	**Accuracy**
1		α_0=0.01	0.28
2		α_0=0.1	0.6
3		α_0=0.5	0.31
4		α_0=1	0.21
5	α_n=3		0.11
6	α_n=30		0.56
7	α_n=150		0.65
8	α_n=300		0.76

Plots do F1 score e heatmaps de confusão

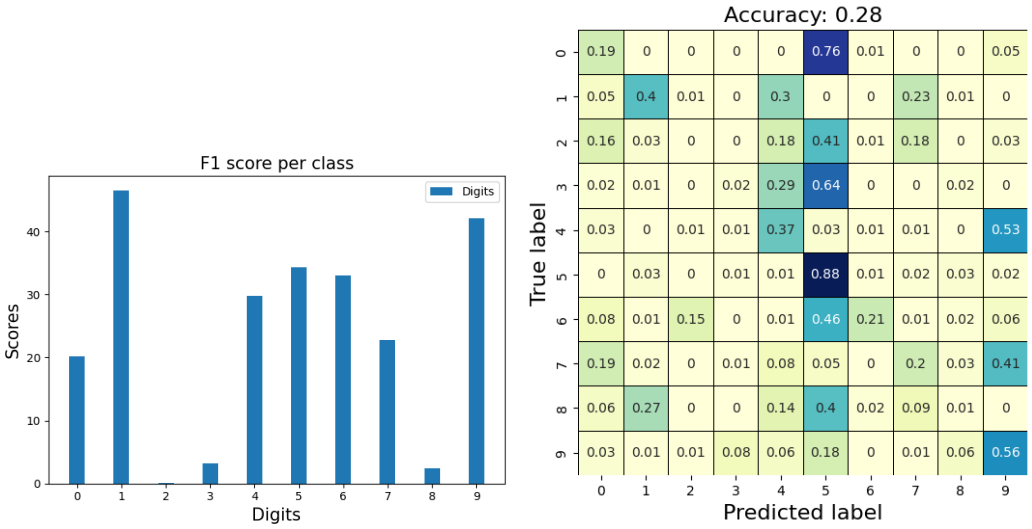

Figura 2: Configuração 1: F1 score máximo é um pouco acimam de 40%, e o heatmap ativa pouco nas diagonais, exceto no caso do 5.

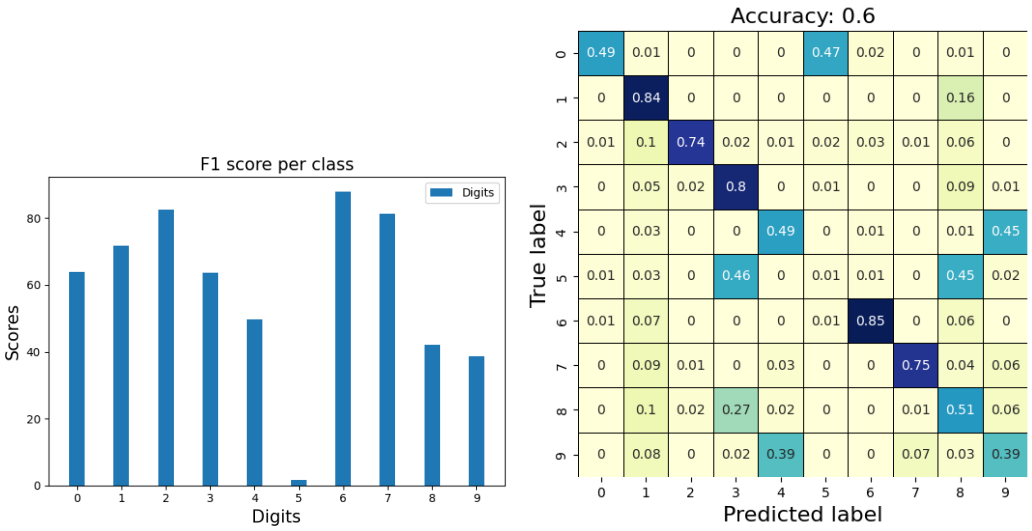

Figura 3: Configuração 2: f1 score maior do que 60% para várias classes e heatmap se ativa na diagonal.

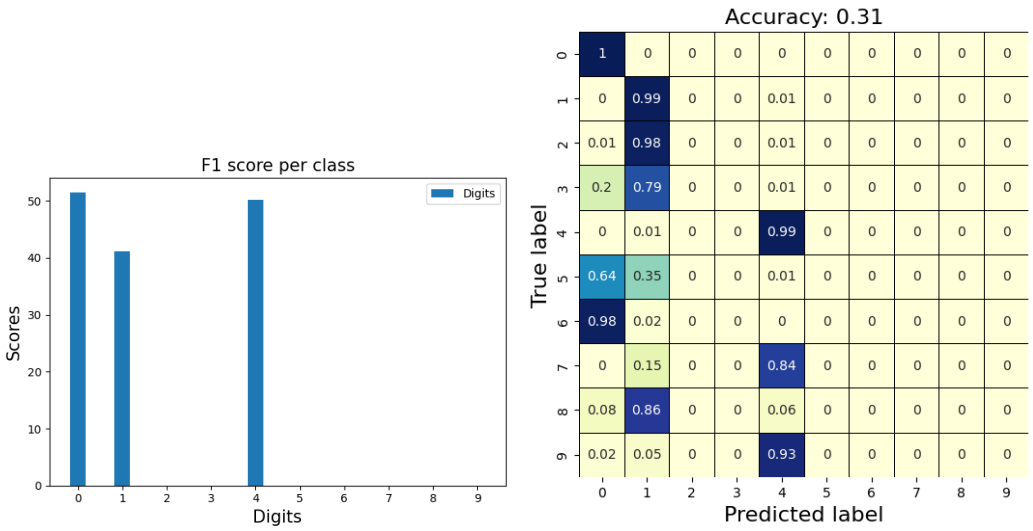

Figura 4: Configuração 3: apenas alguams classes têm F1-score acima de 40% e os demais não possuem. No heatmap há pouca ativação na diagonal, de forma gereal, apenas os dígitos 0,1,4 são classificados corretamente na grande maioria das vezes.

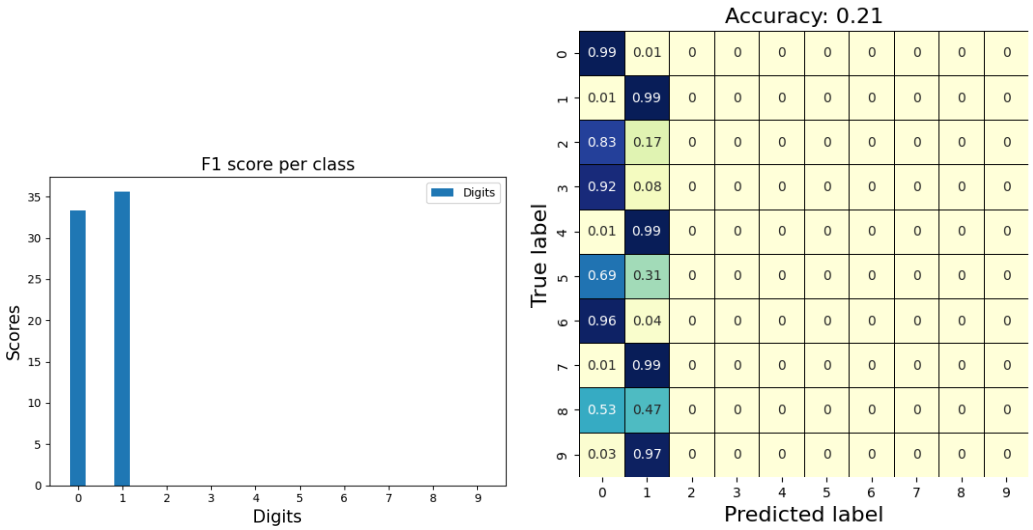

Figura 5: Configuração 4: apenas os dígitos 0 e 1 são classificados, pelo heatmap.

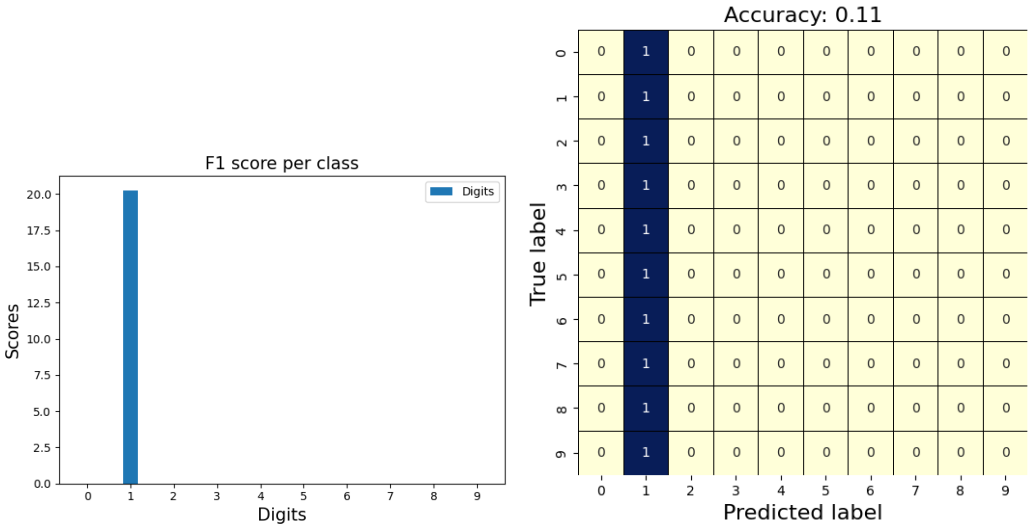

Figura 6: Configuração 5: apenas o dígitos 1 é identificado, sendo todos oss outros também classificados em 1, como se vê no f1-score e no heatmap.

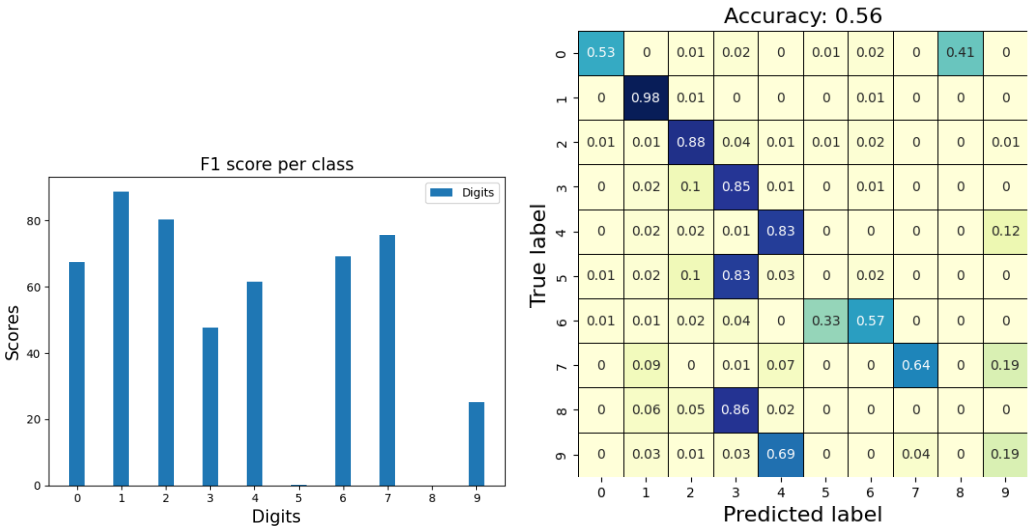

Figura 7: Configuração 6: o $f1$-score aparece em vários dígitos (de médio a alto), e as diagonais do heatmap de confusão se acendem.

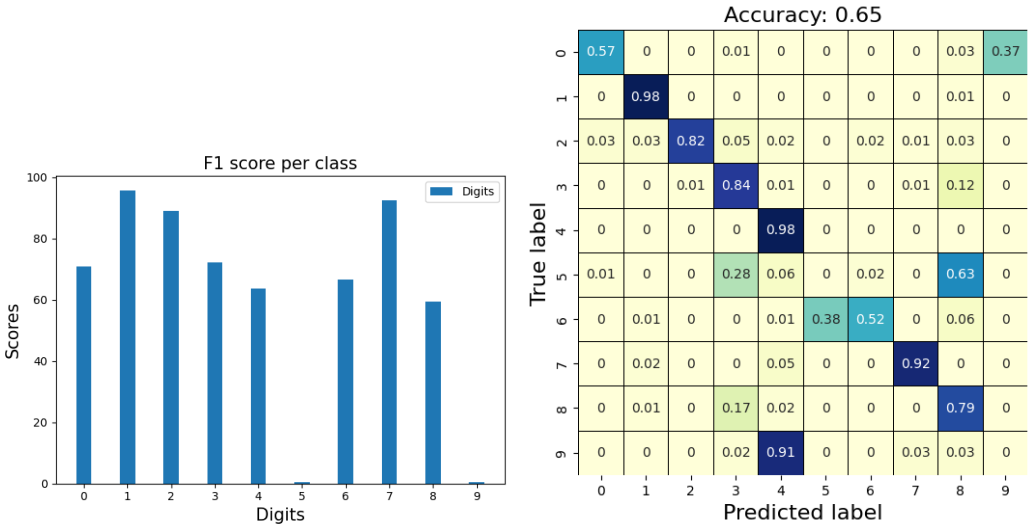

Figura 8: Configuração 7: O $f1$-score é alto para a maioria dos dígitos e a diagonal do heatmap se ativa na maioria dos dígitos.

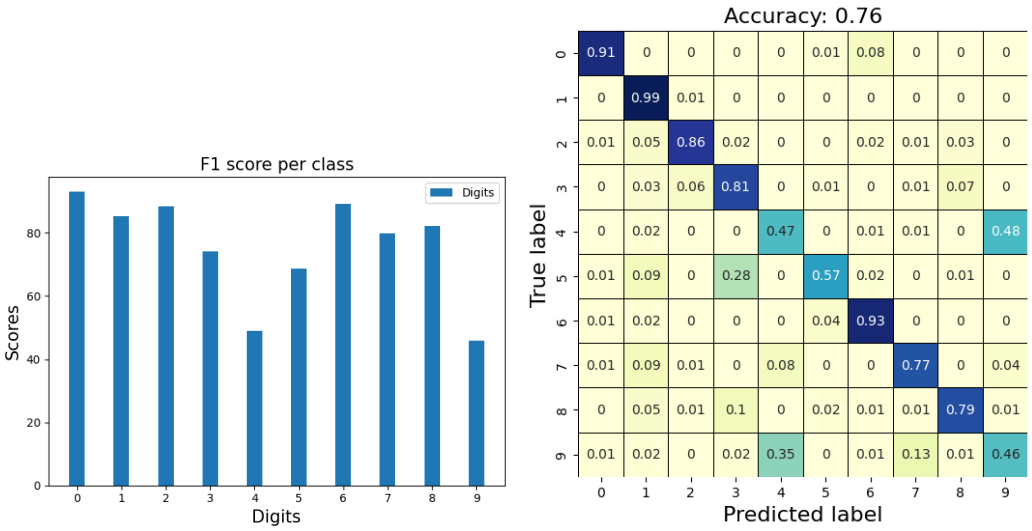

Figura 9: Configuração 8: vê-se que todas as classes apreesentam um valor de médio a muito alto no $f1$-score e no heatmap a diagonal é bem colorida, indicando uma boa performance.

4.1.1 Comentários:

As melhores acurácias ($\geq 50\%$) e resultados significativos de $f1$-score e heatmaps de confusão foram indentificados nas configurações 2, 6, 7, 8, com destaque para a última configuração.

De forma geral, o modelo DKMp, que utiliza a estratégia de α-Pretained com α fixo (constante), apresentou, em média, uma maior acurácia que a estratégia de α-Annealing do DKMa, onde a evolução do α é recursiva. Isso também foi identificado na tabela 2 do paper [1], que mostra uma acurácia superior para o DKMp treinado no MNIST.

Model	MNIST		USPS		20NEWS		RCV1	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Deep clustering approaches without pretraining								
IDEC^{np}	61.8±3.0	62.4±1.6	53.9±5.1	50.0±3.8	22.3±1.5	22.3±1.5	**56.7±5.3**	**31.4±2.8**
DKM^{a}	82.3±3.2	78.0±1.9	**75.5±6.8**	73.0±2.3	44.8±2.4	42.8±1.1	53.8±5.5	28.0±5.8
Deep clustering approaches with pretraining								
IDEC^{p}	**85.7±2.4**	**86.4±1.0**	**75.2±0.5**	74.9±0.6	40.5±1.3	38.2±1.0	**59.5±5.7**	**34.7±5.0**
DKM^{p}	**84.0±2.2**	79.6±0.9	**75.7±1.3**	**77.6±1.1**	**51.2±2.8**	**46.7±1.2**	**58.3±3.8**	**33.1±4.9**

Figura 10: Resultados superiores do **DKMp** em relação a **DKMa**, comparado também com outros métodos de Deep Cluster

4.2 Task 2:

4.2.1 Acurácia dos quatro modelos

Tabela 1: Tabela de Acurácia por Número de Clusters

Num Clusters	Acurácia
2	0.21
5	0.11
10	0.68
20	0.51

4.2.2 Imagens de uma amostra dos quatro modelos

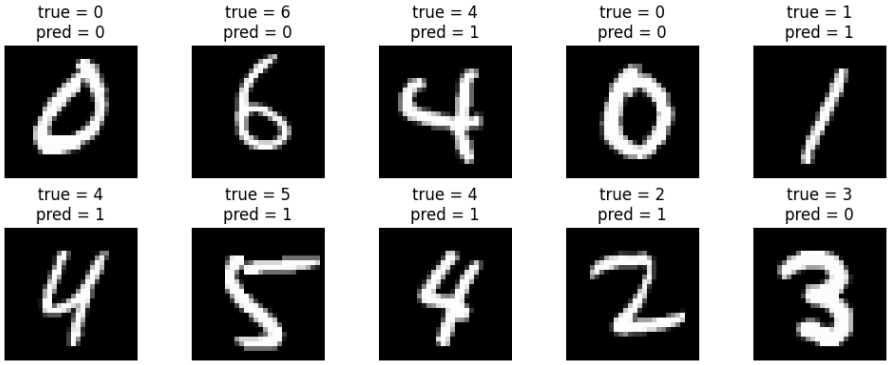

Figura 11: Amostras do modelo com 2 clusters: nesse conjunto de amostras, houveram apenas 3 acertos. Aqui todos os dados são preditos como 1 ou 0 (parece que os dígitos com alguma curva, como o 3 e o 6, sempre são preditos como 0)

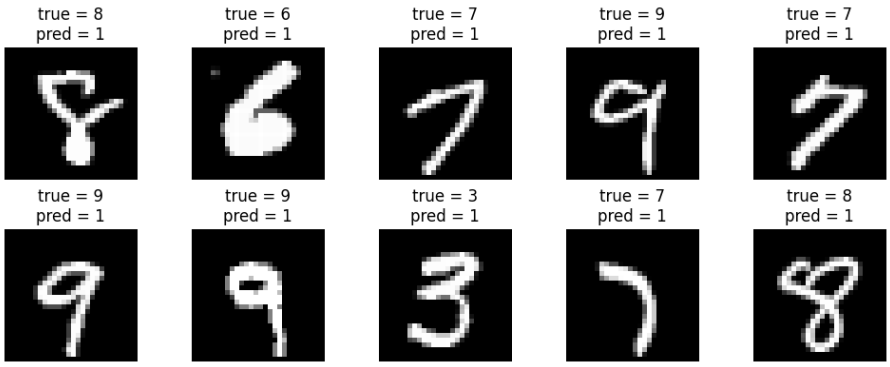

Figura 12: Amostras do modelo com 5 clusters: nesse conjunto amostral, não houve nenhum acerto: todos os dados são preditos como 1 e de fato nenhum label é 1.

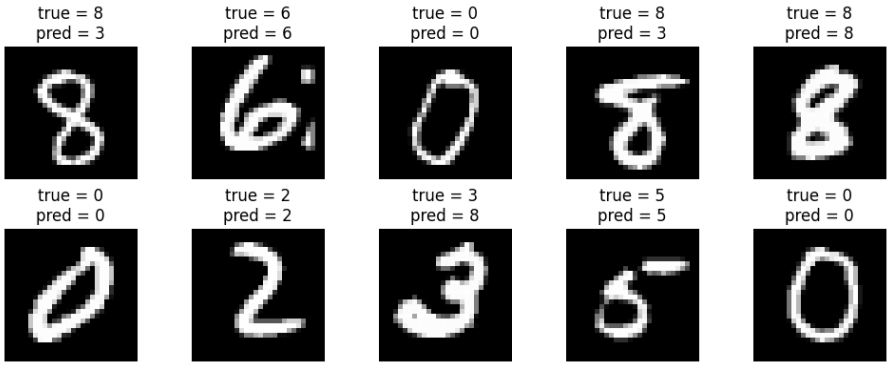

Figura 13: Amostras do modelo com 10 clusters: 7 acertos – três amostras foram incorretamente preditas, pois é possível ver que o 8 e o 3 são frequentemente confundidos.

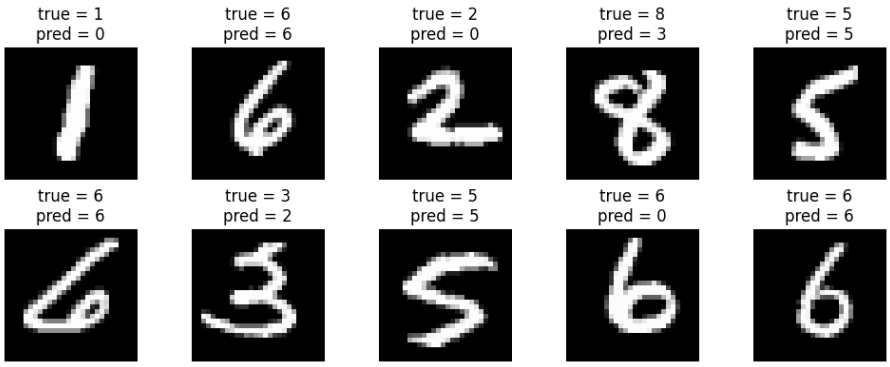

Figura 14: Amostras do modelo com 20 clusters: dese conjunto, o modelo errou a classe de 5 dados.

4.3 Comentários

Para cada conjunto de clusters, verificamos que o conjuntot amostral de 10 clusteres obteve o melhor resultado (7 acertos), seguido pelo de 20 clusteres (5 acertos), que por sua vez se saiu melhor que o de 2 clusteres (3 acertos), e que finalmente superou o de 5 clusters (0 acertos).

Isso reflete a acurácia dos modelos, que em ordem crescente de melhor performance, é a sequência 10, 20, 5, 2.

O fato de 10 clusteres apresentar o melhor resusltado está relacionado a existirem exatamente 10 classes de dígitos. Ao diminuir a quantidade de clusters, mais dados ficam agrupados em menos cluteres que

refletem apenas algumas características mais relevantes como curvas, levando a uma pior performance. Ao aumentar, algumas características menoso relevantes são capturadas pelo cluter, levando a uma pior performance.

5 Conclusão

Neste trabalho tivemos a chance de explorar o Deep k-means em algumas de suas implementaçoees. Verificamos que **DKMp** apresenta a melhor performance em realação ao **DKMa**, o que corrobora a conclusão do artigo:

> This places, to the best of our knowledge, DKMp as the current best deep k-Means clustering method.

Além disso, podemos verificar a performance variando diferentes tamanhos de clusteres. Verificamos que os clusteres de tamanho 10 levaram à uma melhor acurácia, já que existem exatamente 10 classes de digitos na base de dados.

Entretanto, uma das dificuldades encontradas foram os hiperparâmetros sensíveis, como a dependência dos valores de α e demais hiperparâmetros, como a random seed, fazem com que os resultados variem consideravelmente. Os autores do artigo [1] testaram o método em diferentes seeds e mediram o desempenho médio em 10 execuções, mostrando nos resultados a média e o desvio padrão. Segundo os autores, houve muita variância na inicialização aleatória dos hiperparâmetros:

> We observed in pilot experiments that the clustering performance of the different models is subject to non-negligible variance from one run to another. This variance is due to the randomness in the initialization and in the minibatch sampling for the stochastic optimizer.

6 Avaliação da parte prática do curso

6.1 Prós

Acreditamos que ter os assignments práticos auxiliou na compreensão da teoria. Tivemos a chance de explorar várias arquiteturas diferentes de modelos de Deep Neural Networks e muitos são a base dos modelos mais avançados que temos atualmente. Estaremos, dessa forma, mais preparados para ler trabalhos científicos que compreendem os conceitos abordados.

6.2 Contras

Como contras, acreditamos que ter uma referência para os códigos seria útil. Acredito também que os relatórios poderiam estar nos próprios notebooks, pois há um custo de trazer os resultados como imagens e tabelas para um segundo arquivo pdf. Tivemos também algumas dificuldades práticas com o Colab, como sua reinicialização e perda das variáveis, o que contribuiu para um maior tempo para executar os códigos.

6.3 Dicas do que poderia mudar ou ser adicionado

O curso é muito focado em CNN, poderia abordar outras arquiteturas, como GNN.

Referências

[1] Maziar Moradi Fard, Thibaut Thonet e Eric Gaussier. *Deep k-Means: Jointly clustering with k-Means and learning representations*. 2018. arXiv: 1806.10069 [cs.LG].