Exercícios Computacionais 1 – ECp1

Rodrigo Seiji Piubeli Hirao (186837)

16 de dezembro de 2021

Conteúdo

1 Questão 01 - Síntese de modelos lineares para classificação de padrões

Após rodar o script fornecido para o treinamento foi adquirido o coeficiente de regularização 64 para a procura grossa, que resultou numa taxa de acertos de 0.8555, como pode ser visto na figura 01. Visto isso, foi feito a procura refinada em torno do ponto 64, assim descobrindo o coeficiente 88, com taxa de acerto de 0.86, como pode ser visto na figura 2.

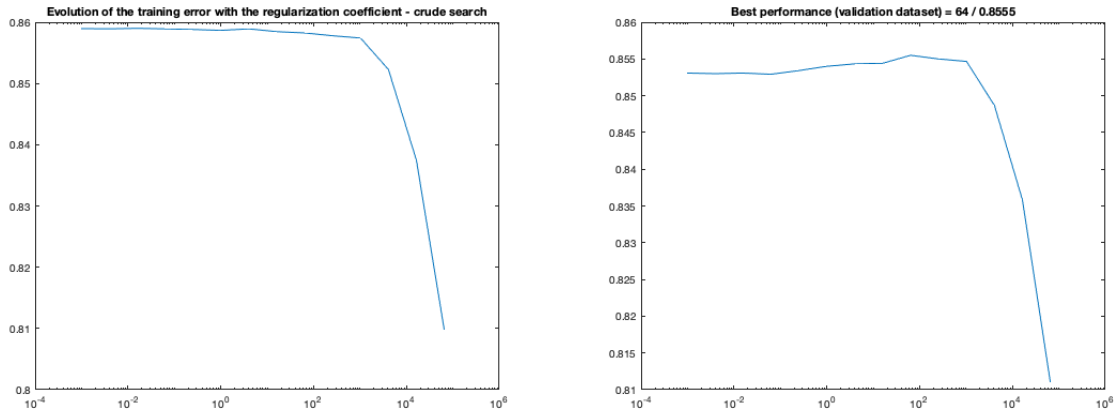

Figura 1: Resultados da procura grossa

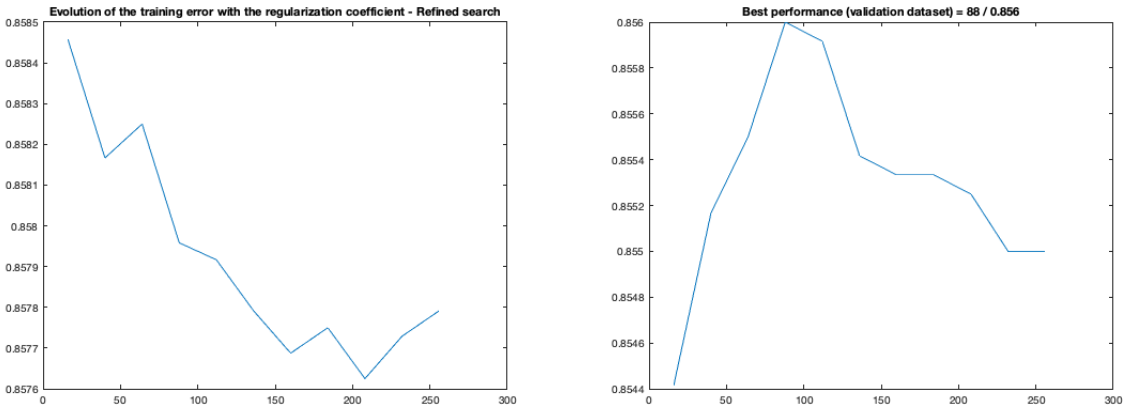

Figura 2: Resultados da procura refinada

Com tal coeficiente, foi então calaculada a matriz de transformação linear, que pode ser visualizada como os mapas de calor da figura 03.

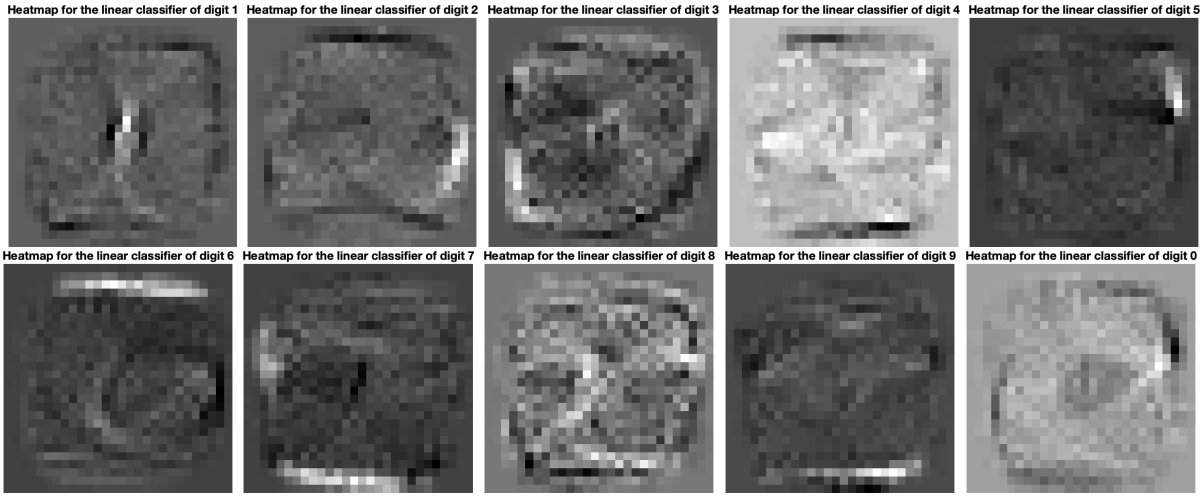

Figura 3: Mapas de calor de cada número

1.1

Então, usando o script a seguir foi executado, produzindo a matriz de confusão da figura 04,

```
confusion = zeros(10,10);

[Nt,mt] = size(St);
for i=1:Nt
    [expected_value,expected] = max(S_est_t(i,:));
    [predicted_value,predicted] = max(St(i,:));

    confusion(expected, predicted) = confusion(expected, predicted) + 1;
end
```

	1	2	3	4	5	6	7	8	9	0
1	1106	55	17	21	17	10	44	47	10	0
2	2	819	21	6	4	10	15	11	3	2
3	2	24	889	0	77	0	6	31	17	2
4	3	15	6	885	24	21	25	25	79	2
5	1	0	14	3	650	17	0	39	0	8
6	5	38	10	9	25	877	1	17	1	12
7	1	18	20	1	15	0	885	13	76	2
8	15	39	19	10	40	6	0	758	4	7
9	0	6	10	47	17	0	47	19	803	1
0	0	18	4	0	23	17	5	14	16	944

Figura 4: Tabela de confusão, com números previstos nas linhas e números esperados nas colunas

E executando o script a seguir percebemos que os elementos 1 e 0 tiveram a melhor percormance

```
diag(confusion)./sum(confusion)'
```

1	0.9744
2	0.8004
3	0.8802
4	0.9002
5	0.7231
6	0.9248
7	0.8716
8	0.7782
9	0.8077
0	0.9612

Figura 5: Tabela de confusão, com números previstos nas linhas e números esperados nas colunas

1.2

Esses resultados poderiam ser melhorados utilizando coeficientes de regularização distintos para cada classe, assim seria penalizado o crescimento de cada classe separadamente, o que poderia produzir minimizações mais fiéis.

1.3

Pelo mesmo motivo (ser usado apenas um coeficiente geral) os gráficos de performance podem produzir resultados não côncavos, pelo fato de uma minimização ser mais difícil.

1.4

Como pode ser visto nos mapas de calor da figura 03, os tons mais claros representam números maiores, enquando tons mais escuros representam números menores. Como cada pixel é o peso que será multiplicado pela imagem original há de perceber que os tons claros acompanham o esperado de cada número, mostrando que a regressão tentou criar um contorno em volta da letra, para dar mais importância para os píxels esperados.

1.5

Após rodar o script fornecido para a classificar objetos mais complexos e coloridos, foi obtido os resultados das figuras 5 e 6.

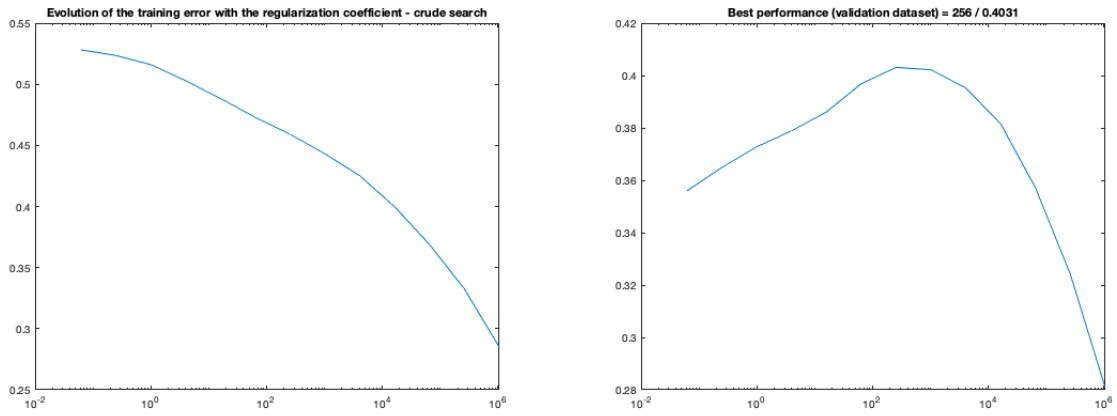

Figura 6: Resultados da procura grossa

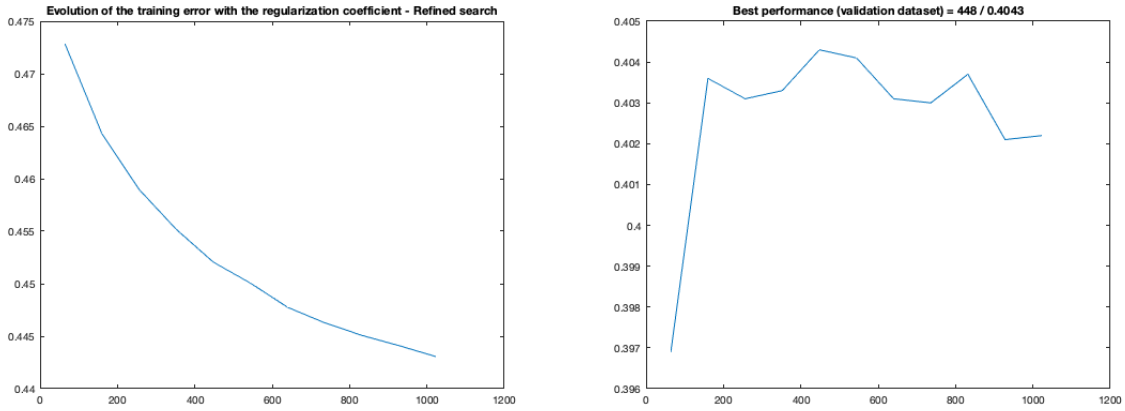

Figura 7: Resultados da procura refinada

Que não foram muito satisfatórios. A razão fica evidenciada quando olhamos os mapas de calor gerados da figura 07, que não conseguiram lidar bem com os objetos complexos, pois estes possuiam muitas características não lineares, como transição de cores, iluminação, etc, que não conseguiram ser bem modelados usando nossa regressão puramente linear.

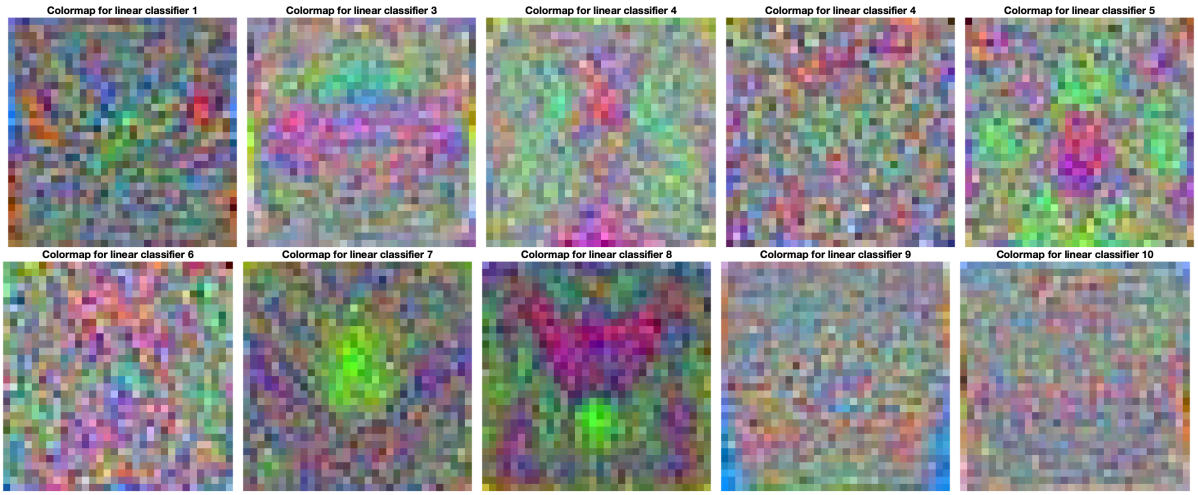

Figura 8: Mapas de calor de cada objeto

2 Questão 02 - Síntese de modelos não-lineares, mas lineares nos parâmetros ajustáveis

Para adicionar uma layer intermediária de 1000 neurônios com distribuição normal aleatória foi adicionado o seguinte código

```
24 % hidden layer
25 H = zeros(m, 1000);
26 for i=1:m
27     for j = 1:1000
28         H(i, j) = normrnd(0,0.2);
29     end
30 end
31 Xtr = Xtr*H;
32 Xv = Xv*H;
33 [N,m] = size(Xtr);
34 [Nv,mv] = size(Xv);
```

E o novo resultado estimado é

```
148 S_est_t = Xt*H*w;
```

Que obteve resultados muito parecidos com o anterior, como pode ser visto nas figuras 9 e 11, que mostram as taxas de acerto de, respectivamente, 0.8573 e 0.4043 para o MNIST e o CIFRA10, para os coeficientes de 4096 e 40. E é esperado que esses coeficientes sejam os mesmos mesmo que o treinamente seja realizado de novo com outros valores aleatórios, pois os valores são uma distribuição normal de forma que os neorônios sejam similares sempre.

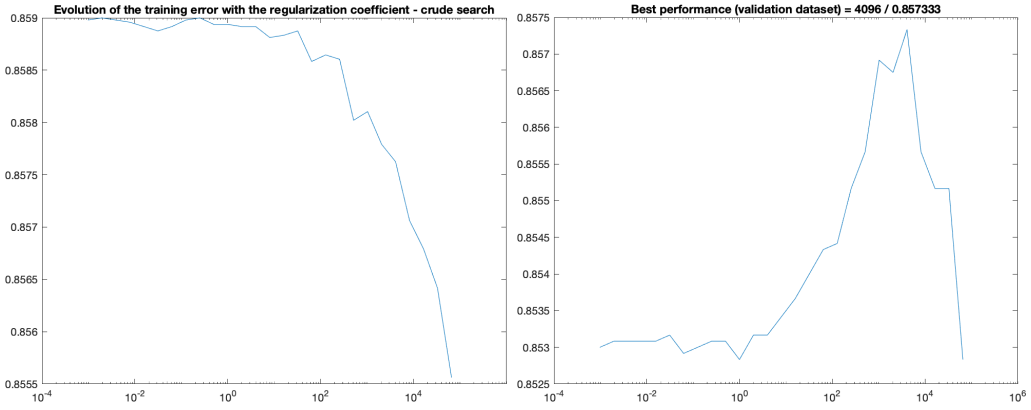

Figura 9: Resultados da procura grossa MNIST (ELM)

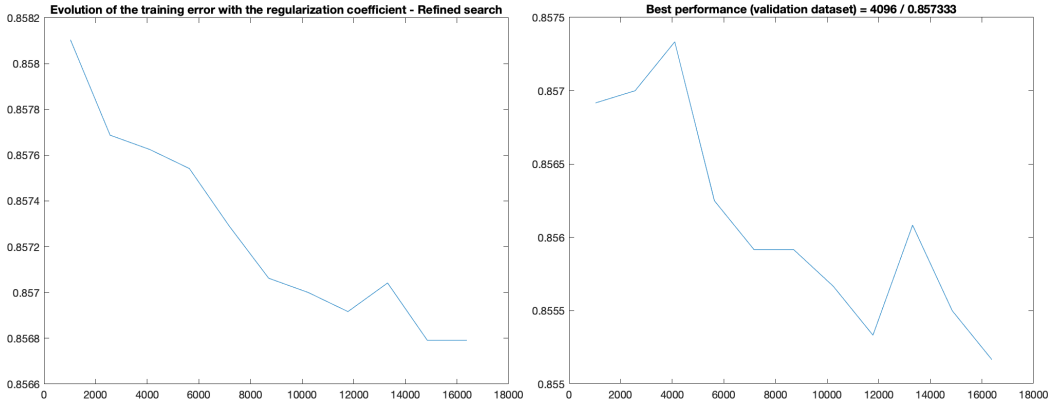

Figura 10: Resultados da procura refinada MNIST (ELM)

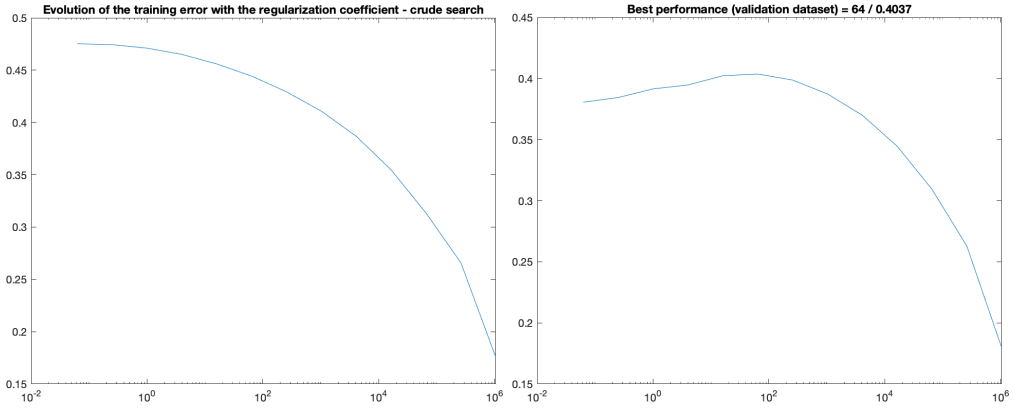

Figura 11: Resultados da procura grossa CIFAR10 (ELM)

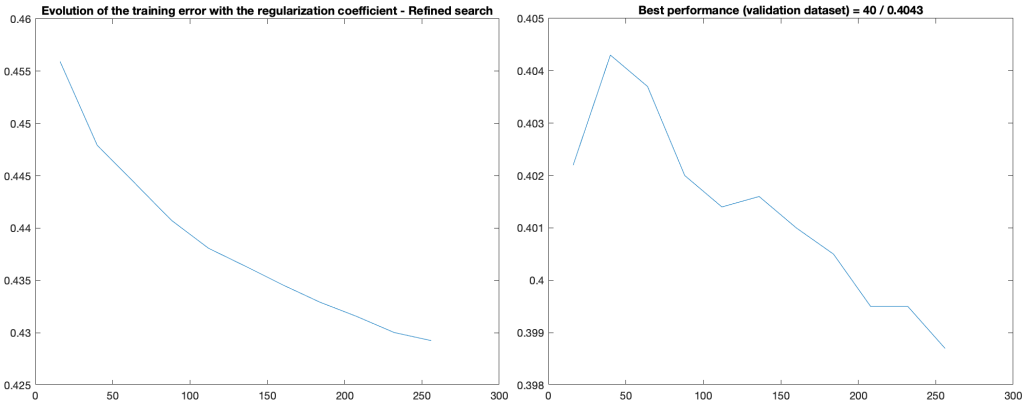

Figura 12: Resultados da procura refinada CIFAR10 (ELM)

Então novamente geramos a tabela de confusão e descobrimos que os valores com maior taxa de acerto são o 1 e o 0.

	1	2	3	4	5	6	7	8	9	0
1	1106	59	17	22	15	11	45	54	12	0
2	2	826	23	6	5	5	17	9	3	2
3	2	23	889	0	84	0	6	32	13	1
4	3	14	5	884	25	16	18	27	72	2
5	1	0	12	1	645	17	0	31	2	8
6	5	30	9	11	25	886	1	19	1	15
7	1	22	23	1	16	0	896	13	64	2
8	15	36	17	11	37	6	0	758	7	7
9	0	4	11	45	17	0	41	19	815	1
0	0	18	4	1	23	17	4	12	20	942

Figura 13: Tabela de confusão do ELM, com números previstos nas linhas e números esperados nas colunas

1	0.9744
2	0.8004
3	0.8802
4	0.9002
5	0.7231
6	0.9248
7	0.8716
8	0.7782
9	0.8077
0	0.9612

Figura 14: Tabela de confusão do ELM, com números previstos nas linhas e números esperados nas colunas

O sistema ELM poderia ser aperfeiçoado com o treinamento dos pesos da camada intermediária, assim como uma MLP. Também poderíamos usar funções não lineares de ativação para poder compreender melhor as imagens não lineares.

3 Questão 03 - MLP

Foram experimentados maiores quantidades de neurônios, de camadas e de épocas, e foi percebido que a partir de uma certa quantidade a acurácia da do treinamento aumentava mas o de teste diminuia, o que mostra que o modelo está especializado demais nos casos de treinamento (talvez um aumento do dropout melhorasse a performance nesses casos). Também foi variada a taxa de dropout mas se percebeu que 50% foi o ponto de melhor desempenho.

Assim o ponto de melhor desempenho encontrado foi o com 10 épocas, 2 camadas, 1024 neurônios em cada camada e 50% de dropout.

Layer 1	Layer 2	Layer 3	Épocas	Acurácia Média dos Testes
Dense/512/50%	-	-	5	0.9804999828338623
Dense/1024/50%	-	-	5	0.9815999865531921
Dense/1024/60%	-	-	5	0.9787814390587604
Dense/1024/40%	-	-	5	0.9767801348514384
Dense/1024/50%	Dense/1024/50%	-	10	**0.9829000234603882**
Dense/1024/25%	Dense/1024/50%	-	10	0.9779000282287598
Dense/1024/50%	Dense/1024/50%	-	15	0.9765999913215637
Dense/1024/50%	Dense/1024/50%	Dense/1024/50%	10	0.9757999777793884

Figura 15: MLP - Acurácia por configuração testada no MNIST (Cada layer no formato: Tipo de Layer / Quantidade de Neurônios / Dropout)

Já para o CIFAR10 tivemos performances muito piores, onde o ponto de melhor performance encontrado foi o de 2 camadas, 1024 neurônios em cada e 25%/50% de dropout.

Layer 1	Layer 2	Layer 3	Épocas	Acurácia Média dos Testes
Dense/1024/25%	Dense/1024/50%	-	5	0.346700012683868
Dense/1024/25%	Dense/1024/50%	-	10	**0.353100001811981**
Dense/1024/50%	Dense/1024/50%	-	10	0.226500004529953
Dense/1024/25%	Dense/1024/50%	Dense/1024/75%	10	0.240700006484985

Figura 16: MLP - Acurácia por configuração testada no MNIST (Cada layer no formato: Tipo de Layer / Quantidade de Neurônios / Dropout)

O código pode ser acessado no google collab
https://colab.research.google.com/drive/1acYu05nwq0oCwfaHLH3pc0DuleU99vwA?usp=sharing

4 Questão 04 - CNN

No caso da rede convolucional foi notado uma performance melhor (acima de 99%), que foi melhorado mais ainda com um matriz de convolução maior ainda, porém com o contraponto do processo de treinamento ficar muito mais demorado.

Layer 1	Layer 2	Layer 3	Épocas	Acurácia Média dos Testes
Conv2D(3x3)/64/25%	Dense/128/50%	-	5	0.9909999966621399
Conv2D(9x9)/64/25%	Dense/128/50%	-	5	**0.9936000108718872**

Figura 17: CNN - Acurácia por configuração testada no MNIST (Cada layer no formato: Tipo de Layer / Quantidade de Neurônios / Dropout)

Já no CIFRAR10 foi natado uma melhora significativa de performance, chegando a quase 70% de acurácia, o algoritmo foi melhorado usando uma convolução 4x4.

Layer 1	Layer 2	Layer 3	Épocas	Acurácia Média dos Testes
Conv2D(3x3)/64/25%	Dense/128/50%	-	5	0.6875000000000000
Conv2D(4x4)/64/25%	Dense/128/50%	-	5	**0.6895000257492065**
Conv2D(9x9)/64/25%	Dense/128/50%	-	5	0.6144000291824341

Figura 18: CNN - Acurácia por configuração testada no CIFAR10 (Cada layer no formato: Tipo de Layer / Quantidade de Neurônios / Dropout)

O código pode ser acessado no google collab
https://colab.research.google.com/drive/1JeT6vEyIuWF5z98bealpG5Ozzt-HbP9F?usp=sharing

5 Questão 05 - encoder/decoder

Foi criado um encoder com 3 camadas (512, 128 e 32 neurônios) que resultou no manifold da Figura 20, que conseguiu as reconstruções da figura 21. Pode se ver que há caracteres que não estão bem definidos, pois pegam a área de 2 números diferentes, pode ser visto aina melhor isso na figura 22, que é o Manifold sendo percorrido por um círculo e por uma elipse, respectivamente, assim gerando números intermediários.

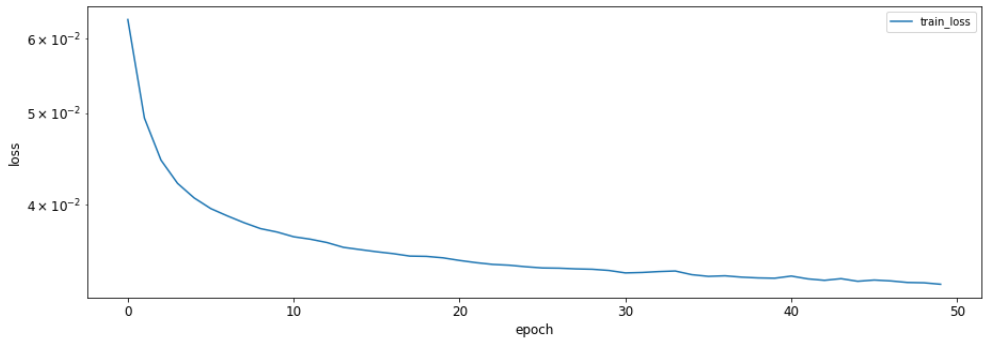

Figura 19: Treinamento do encoder e decoder

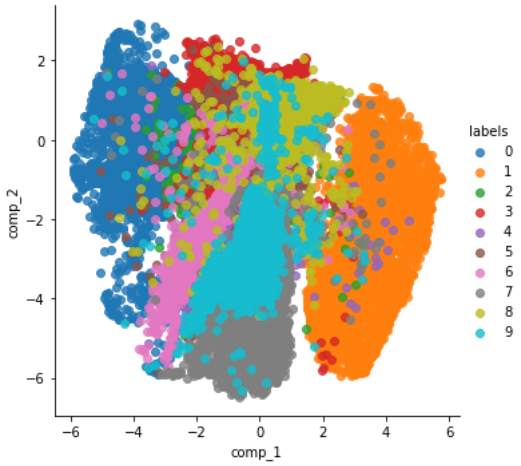

Figura 20: Manifold gerado do autoencoder

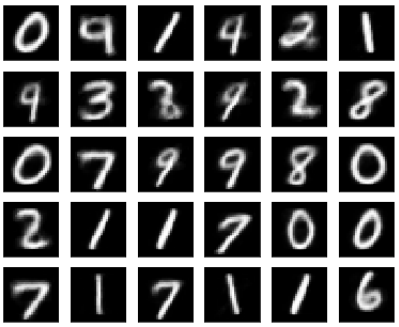

Figura 21: Reconstrução do decoder a partir do Manifold

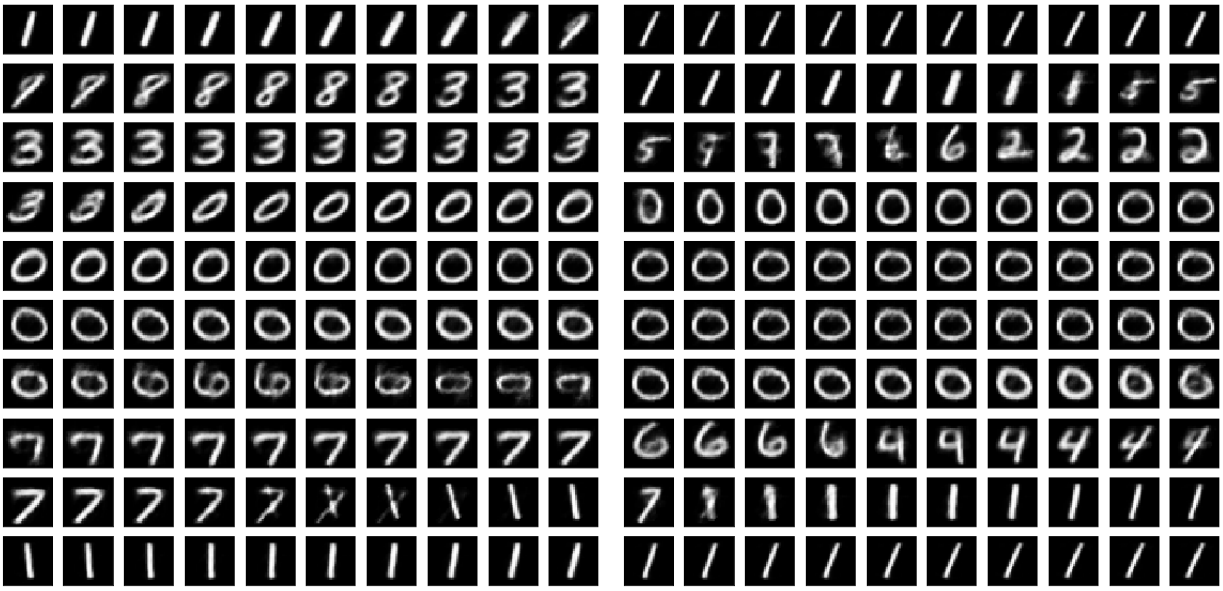

Figura 22: Manifold sendo percorrido por um círculo (esquerda) e uma elípse (direita)

O código pode ser acessado no google collab
https://colab.research.google.com/drive/1YfgTp4jxVHFF5q8k7sKpcMfRPeEu1ysp?usp=sharing