



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Marcos Antonio Lommez Candido Ribeiro¹

Lista Extra #2 - CNN

Inteligência Artificial

¹Aluno de Graduação em Ciência da Computação – tonilommez@hotmail.com

Todo o código gerado pode ser encontrado no link a seguir

[Link para o código](#)

1) Acesse o código disponível em:

<https://www.deeplearningbook.com.br/reconhecimento-de-imagens-com-redes-neurais-convolucionais-em-python-parte-4/>

E faça o seguinte:

1. Teste a CNN para reconhecimento dos personagens Bart e Homer disponível em:

<https://www.kaggle.com/datasets/williamu32/dataset-bart-or-homer?resource=download>

2. Altere a estrutura da CNN, alterando número de camadas de convolução, pooling, kernels, etc

3. Veja o desempenho da CNN a partir destas alterações.

4. Gere um pequeno relatório explicando todas as decisões tomadas acima.

5. Mostre os resultados dos testes realizados.

6. Disponibilize o código

Estrutura Inicial da CNN

- **Camadas Convolucionais:** Uma camada convolucional com 32 kernels de tamanho (3, 3).
- **Camadas de Pooling:** Uma camada de MaxPooling para redução de dimensionalidade.
- **Duplicação das camadas de Convolução e Pooling:** Mais uma camada idêntica a anterior.
- **Ativação:** Função de ativação 'relu' nas camadas convolucionais e na primeira camada densa.
- **Camada Densa:** Uma camada densa com 128 unidades, seguida por uma camada de saída com ativação 'sigmoid'.

- **Compilação:** O modelo foi compilado com o otimizador 'adam' e a função de perda 'binary_crossentropy'.

Desempenho Inicial

- **Treinamento:** 73,47%.
- **Validação:** 82,19%.

Alterações Realizadas

1. Retirando a camada duplicada de Convolucao e Pooling
2. Aumentando para 3 camadas de Convolucao e Pooling
3. Aumentando o numero de kernels
4. Mudando o tamanho dos kernels para 5, 5
5. Duplicando o tamanho da camada densa
6. Mudando a funcao de ativacao para tahn

Desempenho Após Alterações

Mudança	Treinamento	Validação
1 Conv. + 1 Pool.	0.7347	0.7534
3 Conv. + 3 Pool.	0.7602	0.8082
64 kernels	0.7449	0.6575
kernel (5, 5)	0.7500	0.6712
camada densa 256	0.7194	0.7397
tahn	0.7704	0.8493

Tabela 1: Resultados de Treinamento e Validação para Diferentes Mudanças

Análise das Decisões

Em relação ao numero de camadas de pre-processamento (Convolução e Pooling) é percebido uma caída drastica de valores quando treinado utilizando apenas 1 camada, mas também caiu chegando em 3, levando ao valor ideal de 2 camadas nesse caso específico.

O aumento do numero de kernels assim como seu tamanho também gerou uma diminuição drástica da acurácia na validação mas ainda um valor "comum" no treinamento, o que indica que provavelmente existiu overfitting nos dados.

O aumento do numero de camadas também gerou um resultado ruim na rede

Por fim a alteração da função utilizada teve um aumento significativo na validação.

Além disso os algoritmos foram rodados diversas vezes e em todas elas a porcentagem da acurácia sempre seguiu um padrão muito próximo dos valores apresentados que pertencem a ultima iteração.

Embora não apresentado na tabela mas nas mudanças que tiveram desempenho pior a taxa de crescimento por epoca foi mais rápida porque o valor inicial de acurácia começou muito baixo, em contrapartida aqueles valores mais apropriados como o original e a mudança para o tahn que já começaram na primeira epoca com uma taxa de aprendizado relativamente alta.