

Relatório de Atividades – Exercício 4

Nesse exercício foram criadas quatro redes neurais convolucionais com configurações diferentes. Todas as redes foram treinadas com a mesma base de treino e avaliadas nas mesmas bases de teste.

O treino é baseado na base de dados MNIST que consiste em uma base com 55 mil imagens 28x28 de dígitos de 0 a 9. A rede foi avaliada de duas formas: 1) com a base de testes MNIST de 10 mil imagens; 2) com uma base de dados gerada manualmente que possui 4 imagens.

Criação da base de dados de teste

Uma das bases de dados de testes usada foi criada manualmente. Para tal, gerou-se um template 28x28 no Gimp e desenhou-se 4 números diferentes: 3, 4, 5 e 9. Esses dígitos foram escolhidos aleatoriamente. Após serem desenhados, no próprio Gimp foi aplicado um threshold para destacar os pixels brancos.

As imagens foram salvas em RGB no formato JPG e importadas para o Python com a biblioteca Pillow. A função `read_imgs` no script é responsável por ler essas imagens, convertê-las para a escala de cinza (`img.convert('L')`) e normaliza-las para valores entre 0 e 1. A Figura 1 mostra cada uma das imagens criadas.



Figura 1: Imagens RGB criadas no Gimp para testar os modelos de CNN gerados.

Diferentes modelos criados

Como dito anteriormente, quatro arquiteturas de CNN diferentes foram testadas. Todas as arquiteturas possuem os seguintes pontos em comum: 1) foram treinadas com 20 mil steps e o log é extraído a cada 50 steps; 2) o pooling é feito da mesma forma - possui tamanho 2 e stride 2. Decidiu-se não alterar esses parâmetros pois não era um requisito do exercício e porque as imagens são pequenas. Com o primeiro pooling elas diminuem de 28x28 para 14x14 e com o segundo de 14x14 para 7x7. Logo, não seria muito efetivo alterar esses valores; 3) para as redes que possuem apenas uma camada densa, seu tamanho foi mantido em 1024 com uma probabilidade de dropout de 40%.

A primeira arquitetura de testes, CNN1, é a arquitetura sugerida pelo tutorial do TensorFlow. Além das características mencionadas anteriormente, ela

possui 32 mapas na primeira camada convolucional e 64 mapas na segunda. Em ambas os filtros têm tamanho 5x5.

O segundo modelo, CNN2, possui 16 mapas na primeira camada convolucional e 32 na segunda. Em ambas os filtros têm tamanho 3x3.

A terceira arquitetura, CNN3, possui 64 mapas na primeira camada convolucional e 128 na segunda. Em ambas os filtros têm tamanho 7x7.

Por fim, o modelo 4 possui as mesmas configurações do modelo 1 (CNN1), porém com uma camada densa a mais. Nesse caso, a primeira camada densa possui 1024 neurônios com dropout de 20% e a segunda possui 512 com dropout de 40%.

Esses valores foram escolhidos com a intenção de verificar a eficiência da rede caso ela fosse “cortada pela metade”, “dobrada” e mantida da mesma forma, mas com uma camada densa a mais.

Cada modelo está descrito em um script. Como as redes demoram muito para treinar, os scripts são executados em paralelo para obter resultados mais rapidamente.

Resultados e Conclusões

A Tabela 1 mostra o resultado da avaliação do modelo nos conjuntos E1 e E2. E1 corresponde ao conjunto de teste MNIST e E2 ao conjunto criado manualmente.

A partir desses dados pode-se observar que a CNN4 possui uma performance melhor para o conjunto de dados E1, mas uma performance um pouco menor no caso do erro calculado pela entropia cruzada quando comparada à CNN1 no conjunto de testes E2.

Como a CNN4 possui mais uma camada densa, era de se esperar que sua performance fosse melhor que as outras. Conclui-se então que a CNN1 e CNN4 possuem uma convergência similar, mas a CNN4 generaliza melhor.

Um fator interessante é que para todos os casos a acurácia do conjunto de testes E2 é a mesma. Isso sugere que há uma amostra que não consegue ser devidamente classificada. Uma investigação mais aprofundada deve ser feita para descobrir qual dos exemplos é classificado incorretamente e se é o mesmo para todos os modelos.

Infelizmente os resultados da CNN3 não ficaram prontos para serem adicionados a tabela. Neste momento, metade dos steps foram executados e a outra metade deve demorar cerca de 2h para terminar.

Por fim, os gráficos das Figura 2, 3 e 4 mostram a convergência de cada rede durante o treinamento. Pode-se observar que a CNN1 começa a convergir após 6 mil steps, a CNN2 após 4 mil e a CNN4 após 8 mil steps. Aparentemente, quanto mais simples a estrutura mais

rapidamente a rede consegue convergir. Outro ponto interessante é que o valor do erro da CNN2 gira em torno de 0.2 enquanto nas outras, ele gira em torno de 0.3. Porém, baseado na Tabela 1, apesar da convergência da CNN2 ser mais rápida e possuir um valor, ela não consegue generalizar tão bem quanto as outras. Logo, conclui-se que as redes mais complexas, apesar de treinarem com mais dificuldade, são melhores para generalizar a classificação.

	Loss E1	Loss E2	Acc E1	Acc E2
CNN1	0.1042	1.0336	0.9681	0.750
CNN2	0.1267	1.1425	0.9613	0.750
CNN3				
CNN4	0.9017	1.2549	0.9715	0.750

Tabela 1: Resultado da função de erro e da acurácia para os conjuntos de teste. Onde E1 corresponde ao conjunto MNIST e E2 ao conjunto criado manualmente.



Figura 4: Convergência da função de erro para a CNN4



Figura 2: Convergência da função de erro para a CNN1

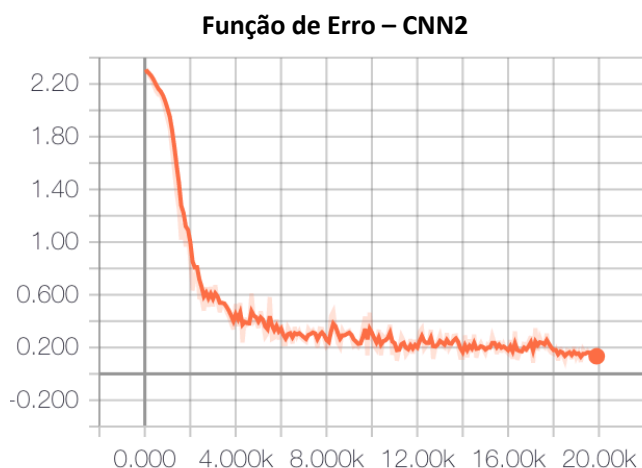


Figura 3: Convergência da função de erro para a CNN2