

Relatório 4

Convolutional Neural Networks usando *tensorflow*

1th Pedro Vidal Sales
Universidade Federal da Bahia
Tópicos em Computação Visual 3
Professor: Maurício Pamplona

I. INTRODUÇÃO

Esse relatório explica os parâmetros utilizados para treinar o modelo de rede convolucional para a tarefa de classificar dígitos, utilizando a biblioteca *tensorflow*.

II. DIVISÃO ENTRE TREINO E VALIDAÇÃO

A base de dados utilizada possui 5000 imagens disponíveis para treino, divididas igualmente em 10 classes, uma para cada dígito. A base foi ordenada e foi fixada uma *seed* com valor 1, para que fosse possível recuperar os conjuntos de treino e validação. Após carregar a base, os dados foram permutados aleatoriamente (imagens e labels correspondentes), e depois divididos entre treino e validação. As primeiras 4000 imagens (depois da permutação) foram utilizadas no conjunto de treino, e as outras 1000 imagens foram utilizadas para validação.

III. ARQUITETURA

Todos os modelos possuíam a mesma arquitetura, 3 camadas de convolução, com 32, 64 e 128 filtros, respectivamente e tamanho de kernel 5x5, seguidas de duas camadas densas, com 128 e 64 nós, respectivamente. A diferença entre cada um dos modelos treinados foi o *augmentation* utilizado.

IV. TREINO

Os modelos A, B, C, D e E foram treinados por 25 épocas. A cada época (uma passada por todas os exemplos) o conjunto de treino foi permutado aleatoriamente, para que os mini-batches fossem diferentes. Cada mini-batch possui 8 exemplos. O número de passos utilizado foi o número de exemplos do conjunto de treino dividido pelo tamanho do mini-batch, para garantir que cada exemplo da base só seria utilizado uma vez por época. Os valores dos pesos e bias foram atualizados com base no otimizador Adam e na taxa de aprendizado. A taxa de aprendizado que obteve melhores resultados foi 0.0005, ela foi escolhida com base nos trabalhos anteriores. A função de ativação utilizada nas camadas convolucionais e nas camadas *fully connected* foi a função ReLU. A função de ativação utilizada para calcular as probabilidades de cada classe foi a função sigmoid. Os pesos e bias foram inicializados utilizando o inicializador *global_variables_initializer* da própria biblioteca.

V. AUGMENTATION

Para cada imagem de treino, era sorteada uma probabilidade de realizar ou não alguma das transformações utilizadas. As probabilidades de cada modelo estão descritas na subseção correspondente.

A. Modelo A

Para o treino do modelo A, não foi utilizada nenhuma técnica de *augmentation*, para que o resultado obtido pudesse ser usado para comparação.

B. Modelo B

As técnicas utilizadas para modelo B foram translação e rotação. A imagem poderia ser rotacionado em -5.0, -2.5, 2.5 e 5.0 graus, e havia 25% de probabilidade de não haver rotação. Feita a rotação, a imagem poderia ser transladada, em até 3 pixels em qualquer direção.

C. Modelo C

A única técnica utilizada para o modelo C foi a técnica de translação. Cada imagem poderia ser transladada em até 3 pixels em qualquer direção.

D. Modelos D, E e F

As técnicas utilizadas para os modelos D, E e F foram rotação, como explicado no modelo B a translação. Para o modelo D, a probabilidade de sortear o número 0 era de 50%, e nos modelos E e F a probabilidade era de 37.3%. A diferença entre o modelo E e F é que o modelo F foi treinado por mais épocas.