

# Relatório 2

## Regressão Logística e Multilayer Perceptron usando *tensorflow*

1<sup>th</sup> 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 regressão logística e o multi-layer perceptron para a tarefa de classificar dígitos, utilizando a biblioteca *tensorflow*.

### II. REGRESSÃO LOGÍSTICA

#### A. 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.

#### B. Treino

O modelo foi treinado por 50 épocas. Cada época corresponde a uma passada por todos os exemplos da base. A cada época o conjunto de validação foi permutado aleatoriamente, para que os mini-batches de cada época 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. Para a taxa de aprendizado, foram testados valores entre 0.007 e 0.0001, e o a taxa que obteve melhores resultados foi 0.0005, por isso ela foi a utilizada. Taxas menores demoravam para alcançar bons resultados, e taxas maiores não alcançavam bons resultados. 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.

#### C. Resultados

O melhor resultado obtido para o conjunto de validação foi 85.2% de acurácia, utilizando os parâmetros descritos na seção anterior.

### III. MULTILAYER PERCEPTRON

#### A. Divisão entre treino e validação

A divisão da base para o multilayer perceptron foi feita da mesma forma que a divisão para a regressão logística, tomando o cuidado de ordenar os exemplos da base antes da permutação, para que os conjuntos de treino e validação fossem os mesmos que os utilizados na regressão logística.

#### B. Treino

O modelo foi treinado por 10 épocas para definir os melhores parâmetros. O tamanho do mini-batch e número de passos foram iguais aos utilizados na regressão logística, pelos mesmos motivos explicados na sessão anterior. O otimizador utilizado também foi o Adam, porém, na camada *fully connected*, foi utilizada a função ReLU como função de ativação, e para a camada de *output* foi utilizada a função softmax cross entropy. Além disso, foi utilizada a técnica de dropout, com taxa igual a 50%. A learning rate utilizada foi 0.0005. Os pesos e bias foram inicializados da mesma forma que a regressão logística. Uma vez que estes parâmetros foram definidos, o modelo foi treinado utilizando 512 e 1024 nós na camada intermediária, com e sem dropout. Ao utilizar 1024, a acurácia aumentou mais rapidamente, por isso, foram treinados por 50 épocas um modelo com e sem dropout. O modelo sem dropout alcançou a acurácia de 93.3%, e o modelo com dropout alcançou uma acurácia máxima de 93.0%, por isso, foi utilizado o modelo com 1024 nós e sem dropout para o teste.

#### C. Resultados

O melhor resultado obtido para o conjunto de validação foi 93.3% de acurácia, utilizando os parâmetros descritos na seção anterior.