

1 Introdução

O objetivo desse laboratório foi implementar (em *Python*) a rede neural *LeNet-5*. Tal rede é conhecida por fazer processamento de imagens e, tendo em vista essa utilidade, o objetivo final foi estudar o processamento de imagens manuscritas de algarismos.

2 Código

A implementação da *LeNet-5*, assim como de varias redes neurais, se torna muito simples, uma vez que estamos utilizando o framework *Keras* do *Google*. Com isso, bastou criar as camadas da rede, baseando-se na tabela 1 do roteiro. Tal implementação está descrita abaixo:

```
def make_lenet5():
    model = Sequential()

    model.add(layers.Conv2D(filters=6, kernel_size=(5,5), strides=(1,1), activation=activations.tanh,
                             input_shape=(32,32,1)))
    model.add(layers.AveragePooling2D(pool_size=(2,2), strides=(2,2)))
    model.add(layers.Conv2D(filters=16, kernel_size=(5,5), activation=activations.tanh))
    model.add(layers.AveragePooling2D(pool_size=(2,2), strides=(2,2)))
    model.add(layers.Conv2D(filters=120, kernel_size=(5, 5), activation=activations.tanh))
    model.add(layers.Flatten())
    model.add(layers.Dense(units=84, activation=activations.tanh))
    model.add(layers.Dense(units=10, activation=activations.softmax))

    return model
```

3 Resultados e discussão

3.1 Treinamento da LeNet-5

As figuras 1 e 2 mostram, respectivamente, a evolução da acurácia e da *loss function* com a época e número de iterações, respectivamente. Observe que o resultado foi esperado: a acurácia cresceu e aproximou-se de 1 (100%), enquanto a *loss function* caiu e se aproximou de zero.

3.2 Avaliação da LeNet-5

As imagens 3, 4, 5 e 6 são exemplos em que a rede neural conseguiu interpretar os algarismos de forma correta. Já nas figuras 8, 9, 10, 11 e 12, estão exemplos em que a rede neural falhou na interpretação dos algarismos. Note que a rede errou em exemplos que são difíceis até para um ser humano identificar.

epoch_accuracy

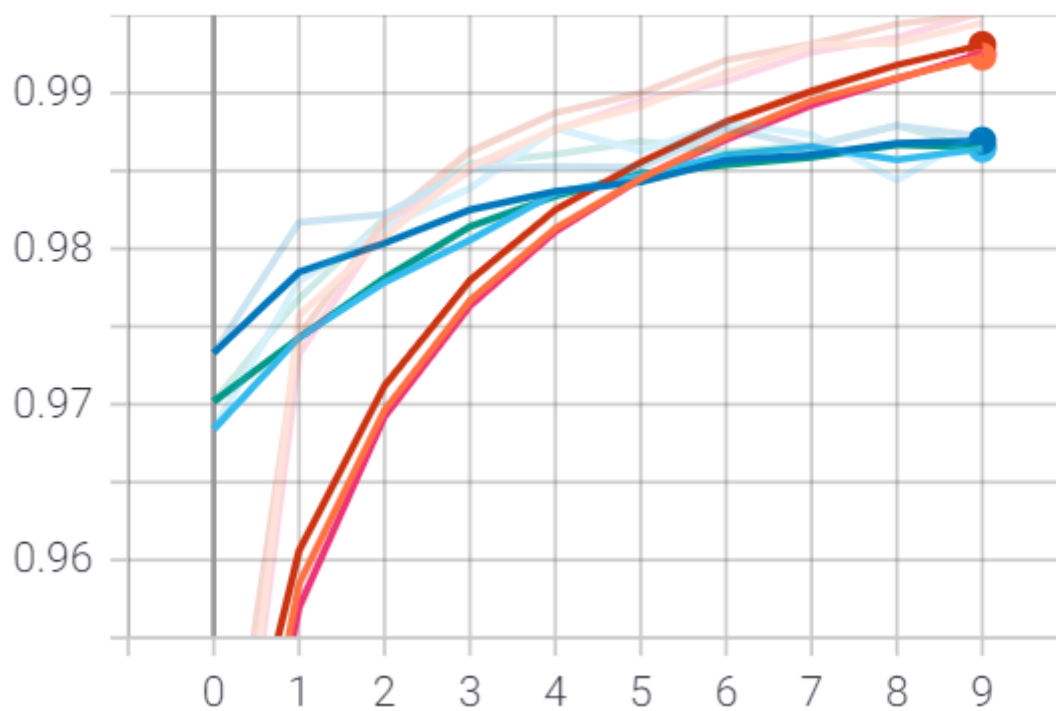


Figura 1: Evolução da precisão do método.

evaluation_loss_vs_iterations

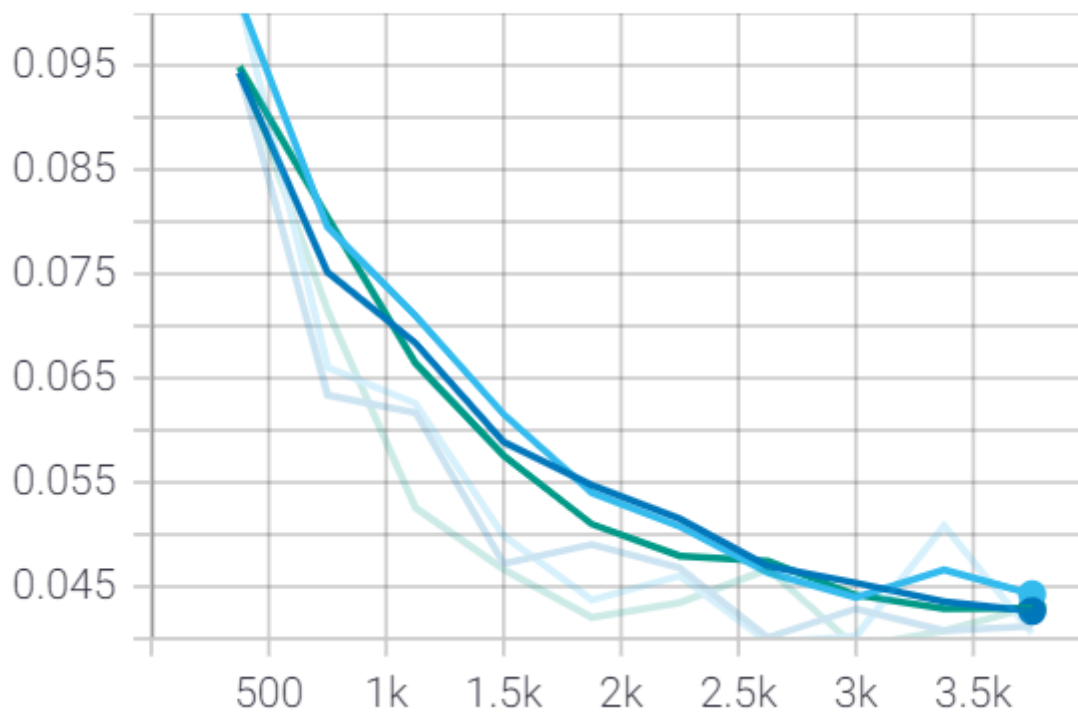


Figura 2: Evolução da *loss function* com o tempo.

Example: 3699. Expected Label: 1. Predicted Label: 1.

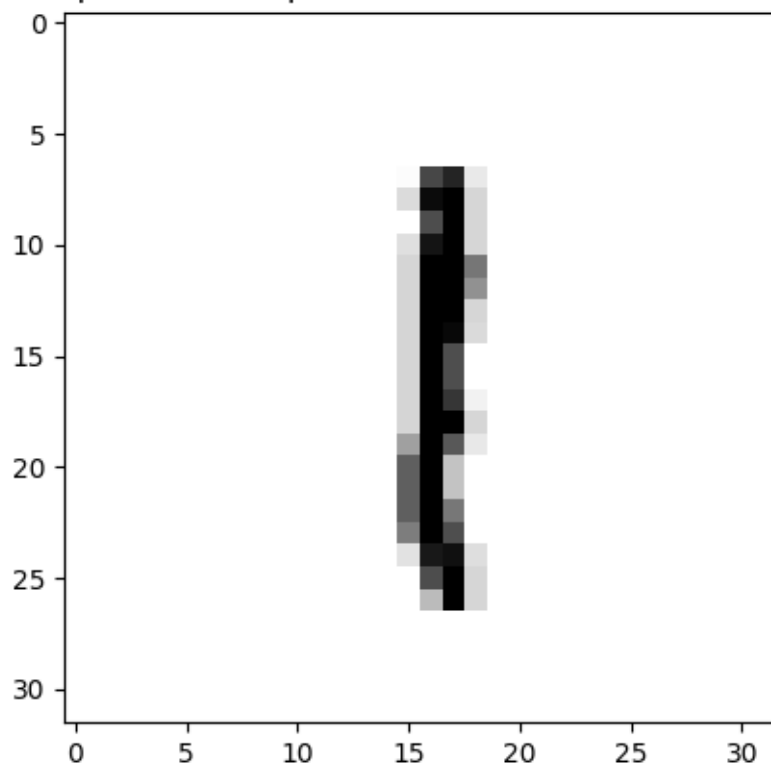


Figura 3: Exemplo 3699. Note que a rede neural conseguiu prever o valor do número esperado.

Example: 4302. Expected Label: 5. Predicted Label: 5.

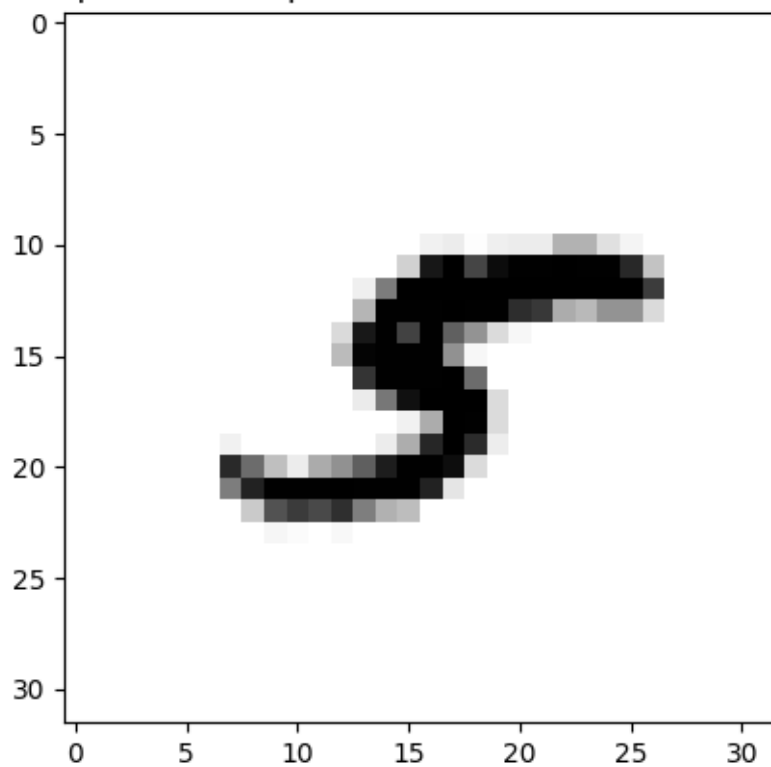


Figura 4: Exemplo 4302. Note que a rede neural conseguiu prever o valor do número esperado.

Example: 5663. Expected Label: 7. Predicted Label: 7.

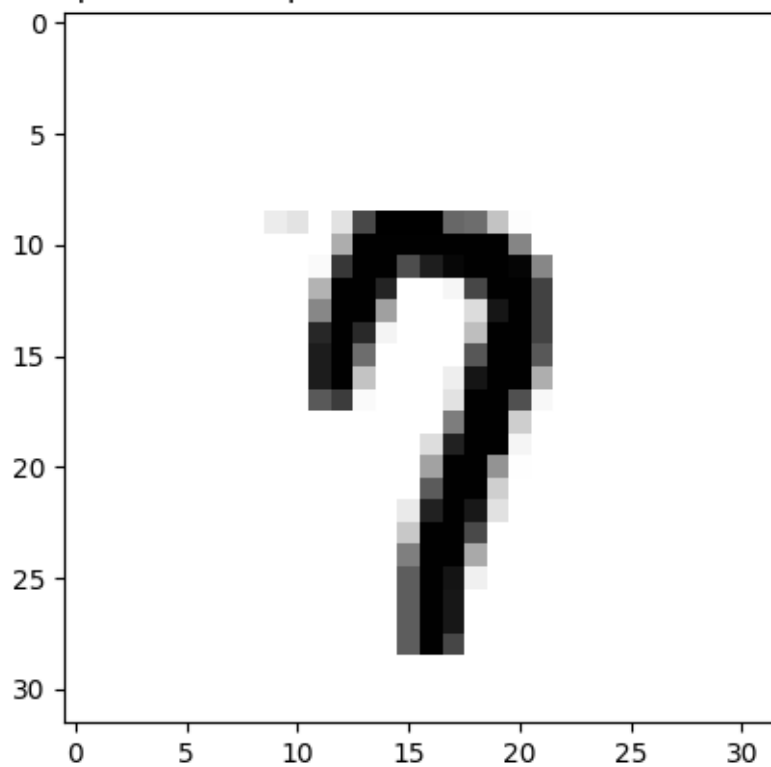


Figura 5: Exemplo 5663. Note que a rede neural conseguiu prever o valor do número esperado.

Example: 5904. Expected Label: 3. Predicted Label: 3.

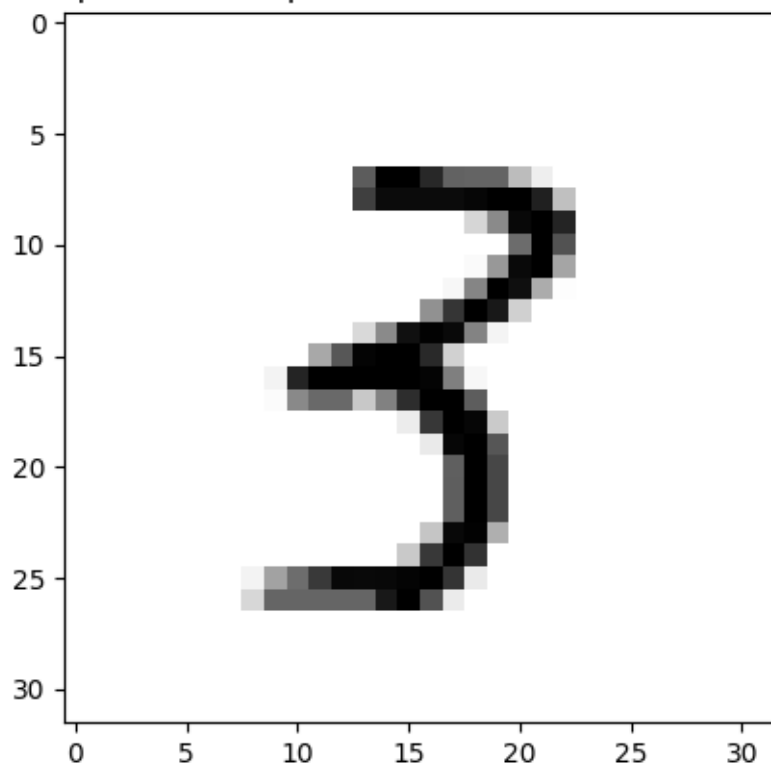


Figura 6: Exemplo 5904. Note que a rede neural conseguiu prever o valor do número esperado.

Example: 6416. Expected Label: 9. Predicted Label: 9.

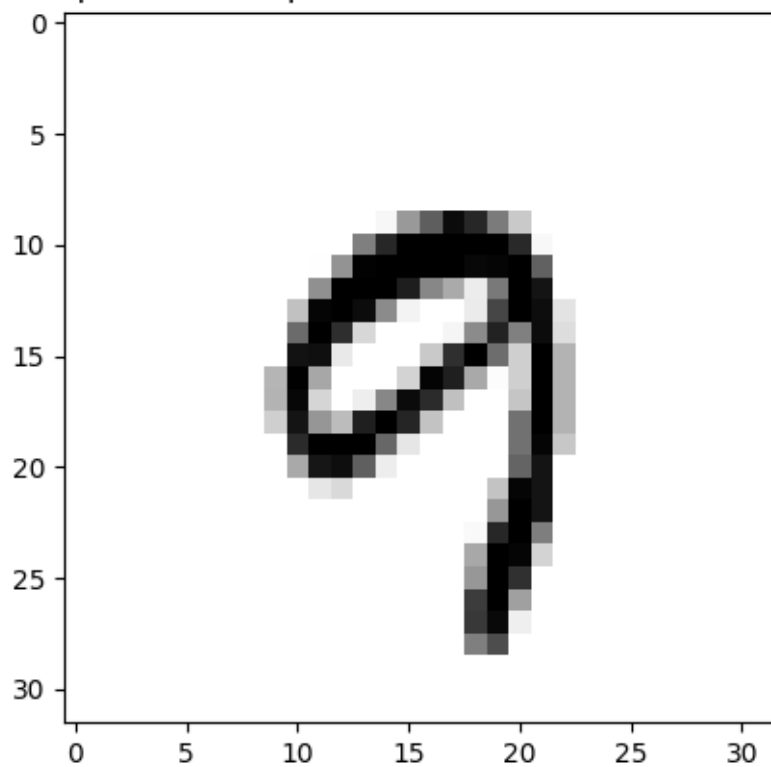


Figura 7: Exemplo 6416. Note que a rede neural conseguiu prever o valor do número esperado.

Example: 247. Expected Label: 4. Predicted Label: 6.

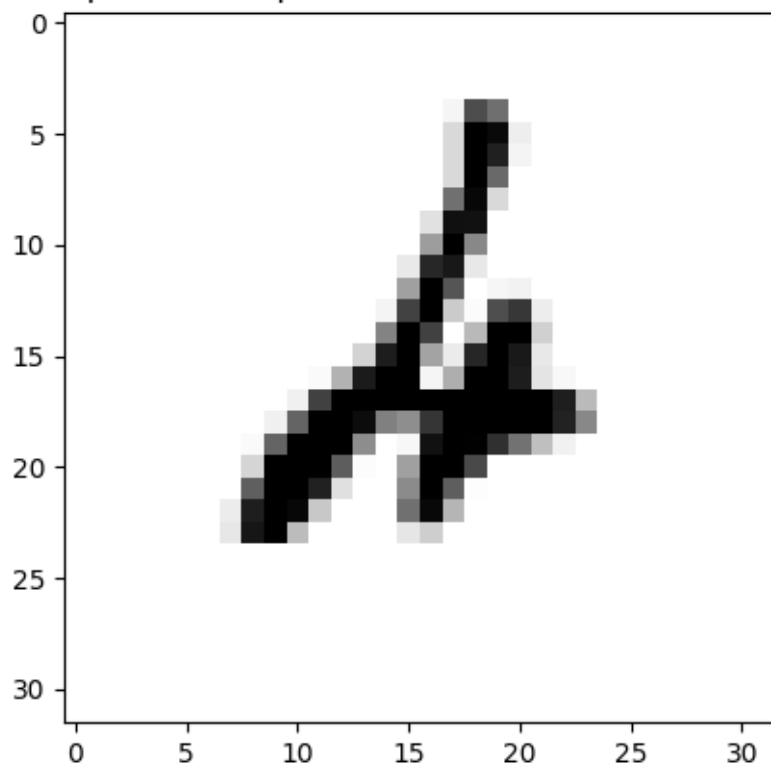


Figura 8: Exemplo de imagem mal-classificada pela rede neural.

Example: 321. Expected Label: 2. Predicted Label: 7.

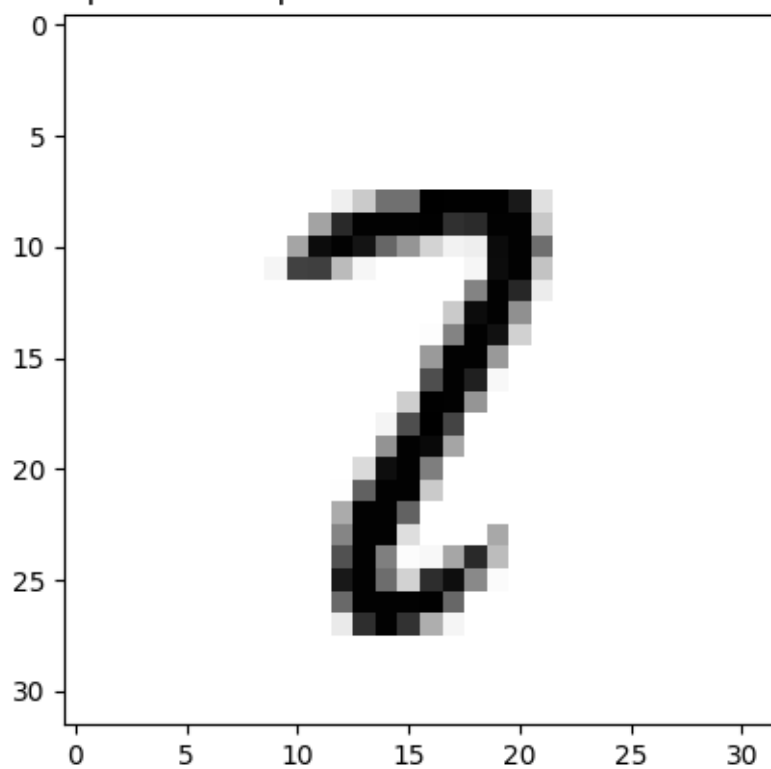


Figura 9: Exemplo de imagem mal-classificada pela rede neural.

Example: 340. Expected Label: 5. Predicted Label: 3.

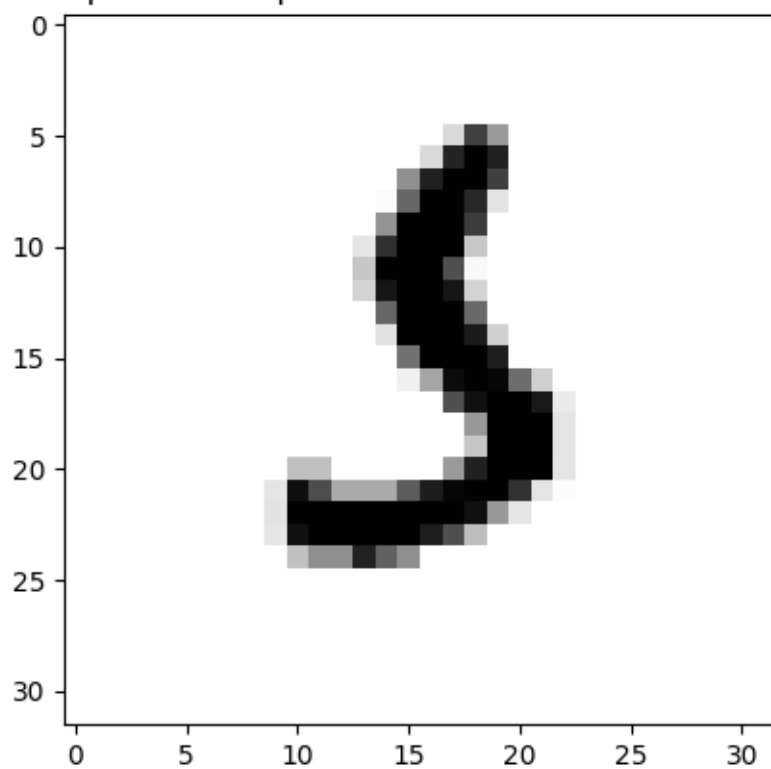


Figura 10: Exemplo de imagem mal-classificada pela rede neural.

Example: 445. Expected Label: 6. Predicted Label: 0.

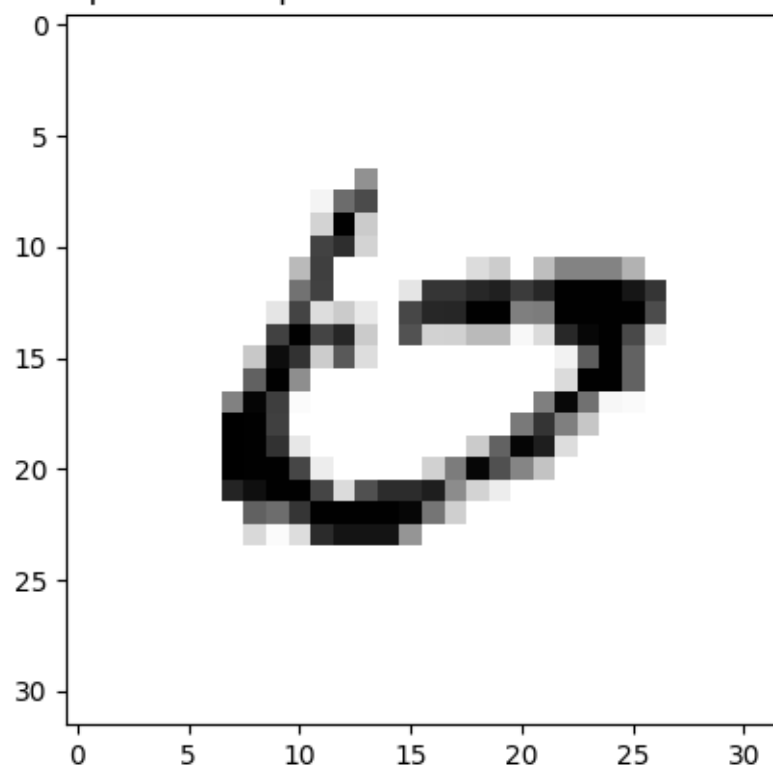


Figura 11: Exemplo de imagem mal-classificada pela rede neural.

Example: 582. Expected Label: 8. Predicted Label: 2.

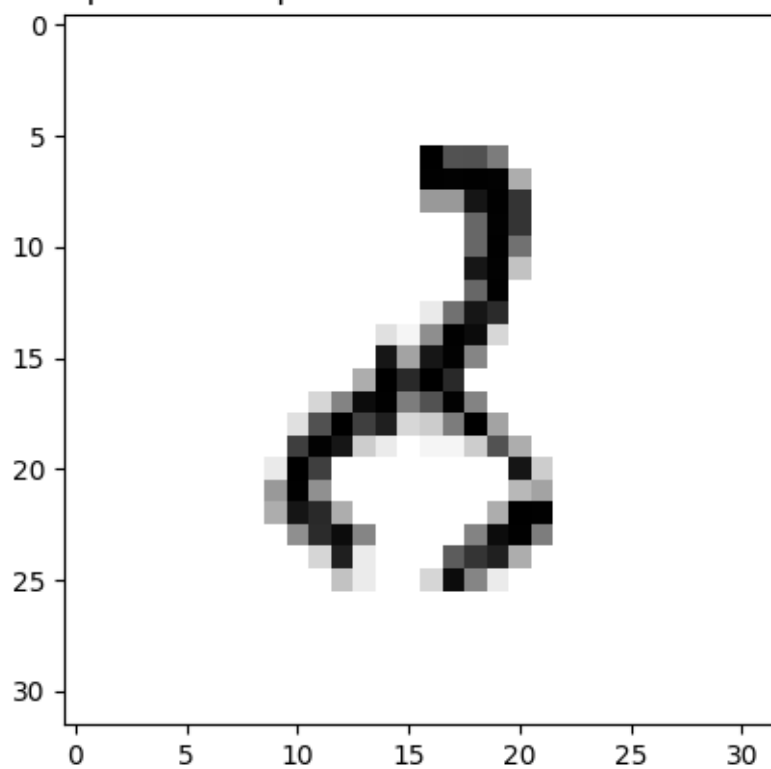


Figura 12: Exemplo de imagem mal-classificada pela rede neural.