Optimzing Neural Networks

Luis Moran and Diego Palomero

18 de diciembre de 2018

1. Introduction

In this task the neural network is going to use other architectures and training algorithms to compare the results obtained with the ones in Task 2. In this case the Adam Algorithm will be used instead of the Gradient Descent. Additionaly, a deeper network and residual blocks will to be used too.

2. Implementation

Two type of architectures will to try different activation functions and learning rates on them. These architectures are:

Deep neural network

For the implementation of the deep neural network with 9 hidden layers the hidden layers can be added just like in Task 2. For the Adam Optimization the optimizer Gradient Descent is changed with the class AdamOptimizer from TensorFlow.

ResNet architecture

To create the 9 layer ResNet architecture four residual blocks are being inserted following the architecture of figure 1 Each residual block follows this function:

$$reLU(x + W_2 ReLU(W_1 x + b_1) + b_2)$$

referencing 1 and 2 the latest 2 layers before the residual block.

Consdering the figure 2 shown in the task and the previous equation $W_1 x + b_1$ is the output of the first layer, which can be called y_1 and in consequence each residual block output is:

$$reLU(x + W_2 y_1 + b_2)$$

.

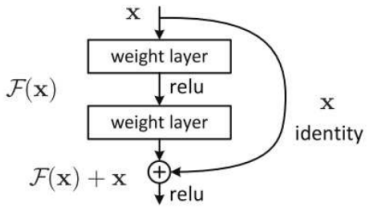

Figura 2: A residual block

3. Results

First the 9 hidden layers architecture is tested with ReLu and tanh activation functions. Early stopping is used although the execution is not stopped (only the results saved) so it can be visualized in the graphics the entropy and accuracy error after the early stopping is triggered. the results represent the maximum value before the early stopping is triggered

Activation function	Learning rate	Training error	Test error
ReLu	0.001	0.694	0.555
ReLu	0.002	0.744	0.607
ReLu	0.005	0.824	0.682
Tanh	0.001	0.590	0.448
Tanh	0.002	0.789	0.632
Tanh	0.005	0.077	0.077

It is noticeable how although the architecture has more layers the results are worse than in the Task 2 and they fluctuate much more in this case probably because of overfitting.

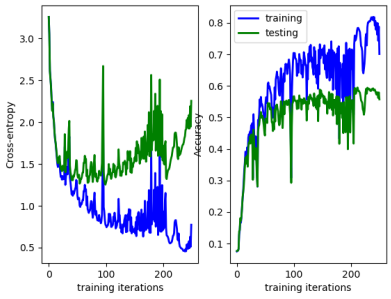

Figura 3: 0,001 learning rate ReLu

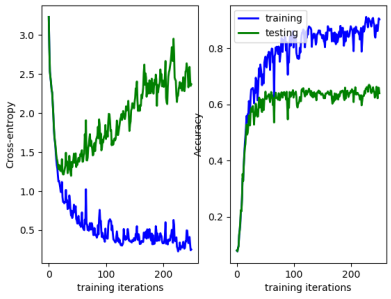

Figura 4: Relu 0,002 learning rate

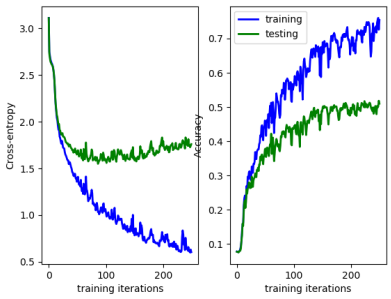

Figura 5: 0,001 learning rate tanh

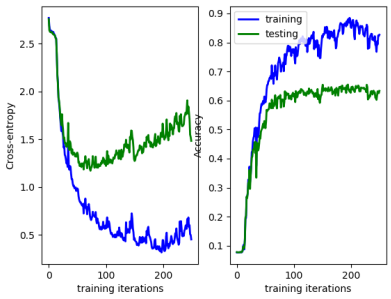

Figura 6: tanh 0,002 learning rate

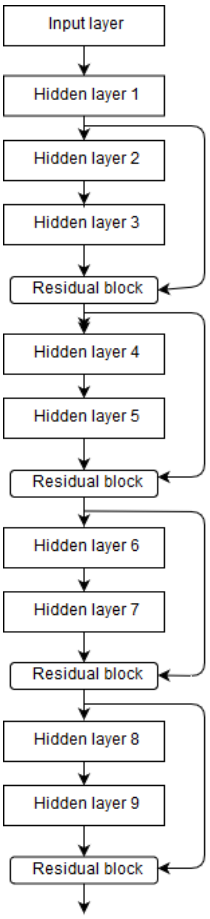

Figura 1: ResNet Architecture