

Traffic Sign Classifier

An approach to solve German Traffic Classification Problem

Manuel Scurti - Mingjie Ye



Problem

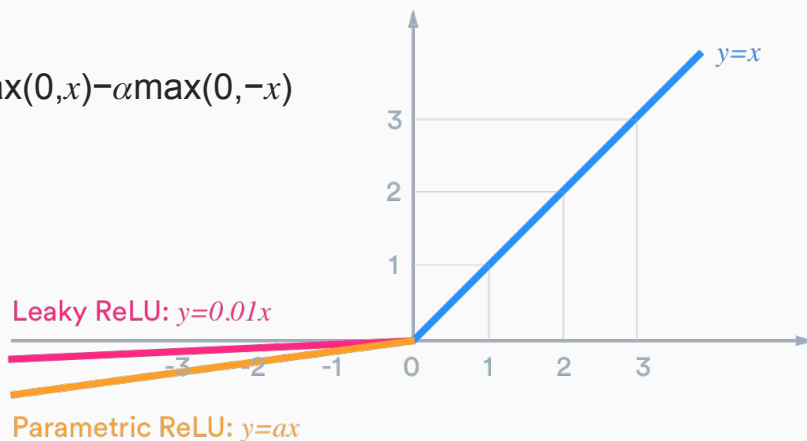
The german traffic sign benchmark (GTSRB) is a multi-class classification challenge. The benchmark has the following properties:

- Multi-class classification problem
- More than 40 classes
- More than 50000 images in total

Network Architecture

structure of the net

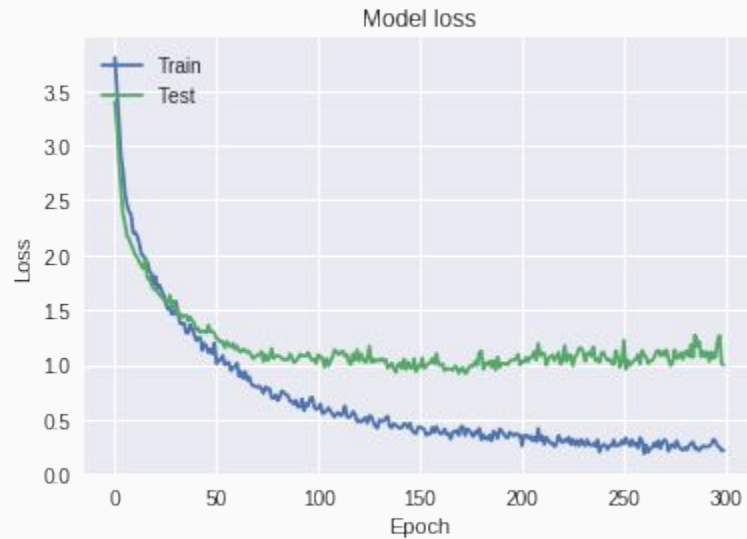
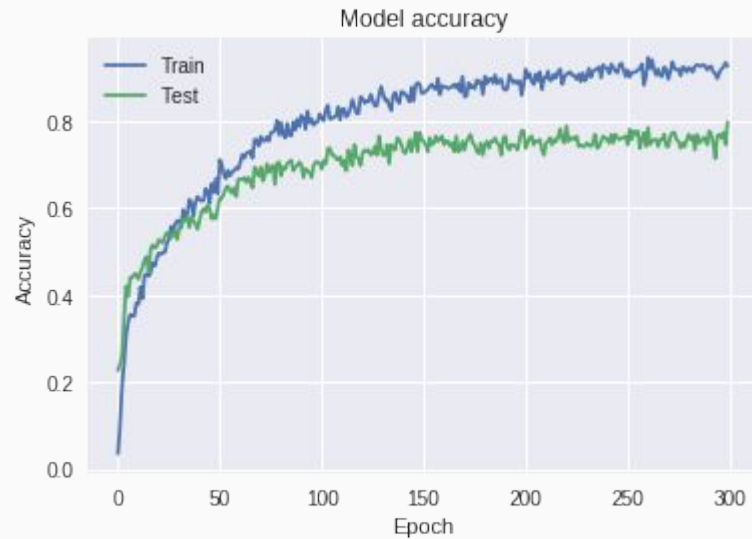
- 6 hidden layer
- Activation function: Leaky ReLU $f_1(x) = \max(0, x) - \alpha \max(0, -x)$
- Dropout as 0.15 or 0.3



Network Architecture

Layer (type)	Output Shape	Param #			
dense_1 (Dense)	(None, 32, 32, 1)	4	dense_4 (Dense)	(None, 256)	262400
flatten_1 (Flatten)	(None, 1024)	0	activation_4 (Activation)	(None, 256)	0
activation_1 (Activation)	(None, 1024)	0	dropout_4 (Dropout)	(None, 256)	0
dropout_1 (Dropout)	(None, 1024)	0	dense_5 (Dense)	(None, 256)	65792
batch_normalization_1 (Batch Normalization)	(None, 1024)	4096	activation_5 (Activation)	(None, 256)	0
dense_2 (Dense)	(None, 2048)	2099200	dropout_5 (Dropout)	(None, 256)	0
activation_2 (Activation)	(None, 2048)	0	dense_6 (Dense)	(None, 100)	25700
dropout_2 (Dropout)	(None, 2048)	0	activation_6 (Activation)	(None, 100)	0
dense_3 (Dense)	(None, 1024)	2098176	dropout_6 (Dropout)	(None, 100)	0
activation_3 (Activation)	(None, 1024)	0	dense_7 (Dense)	(None, 43)	4343
dropout_3 (Dropout)	(None, 1024)	0	activation_7 (Activation)	(None, 43)	0

Performance



Process followed

From the original architecture what we noticed was:

- Overfitting issues
- Very slow process, impossibility to add more layers
- It was a wide net, not a deep net

What we changed to speed-up

- Scale images size and thus, number of input neurons
- Added more layers but with fewer neurons

Result: MLP took from 0.23 seconds to 0.053 to achieve also a better accuracy score

Additional attempts made

During the process, once we resolved initial overfitting issues, we tried different strategies

- Image size (from 224x224 to 32x32) [WINNING STRATEGY]
- data augmentation (by applying zooming, rotations and other image transformations)
- early stopping
- other optimizers (Adam, Adagrad, Adadelata, Adamax)

Thank you!

